Horizon Europe Programme

Research and Innovation Action

This project has received funding from European Union Horizon Europe programme, under Grant Agreement N°101080923

Start date of project: 1st May 2023

Duration: 42 months

D5.1 Design and architecture of the platforms

Deliverable details	
Work Package Title	Advanced tools and open knowledge platform
Task Number	T5.1
Deliverable Number	D5.1
Deliverable Title	Design and architecture of the platforms
Revision Number	1.3
Responsible Organization	FTK
Author(s)	Sebastian Bruchhaus (FTK), Alex Butean (WIZ), Dominic Heutelbeck (FTK), Izidor Mlakar (UoM), Dah DIARRA (RDIUP), Habib NASSER (RDIUP), Zouhair Haddi (NION)
Due Date	30 April 2024
Delivered Date	28 June 2024
Reviewed by	Shammi Seth (HWU) / Theodore Lim (HWU) & Vincenza Frisardi (IRCCS)
Dissemination level	Public

Contributing partners	
1.	FTK
2.	NION
3.	WIZ
4.	RDIUP
5.	UoM
6.	NURO

Disclaimer: SMILE is a project co-funded by the European Commission under the Horizon Europe Programme - Call: HORIZON-HLTH-2022-STAYHLTH-01-two-stage under Grant Agreement No. 101080923.

Funded by
the European Union

Glossary

Abbreviation	Description
AAI	Authentication and Authorization Infrastructure
ABAC	Attribute-Based Access Control
AI	Artificial Intelligence
API	Application Programming Interface
CB	Chatbot Framework
CBN	Causal Bayesian Network
CDA	Clinical Document Architecture
CRUD	Create, Read, Update, Delete
DCMM	Data Collection and Management Module
DDD	Data Driven Design
DSL	Domain-Specific Language
DSS	Decision Support System
ESM	Experience Sampling Method
FHIR	Fast Healthcare Interoperability Resources
GAD-7	Generalised Anxiety Disorder Assessment
GDPR	General Data Protection Regulation
IAM	Identity and Access Management
JRE	Java Runtime Environment
JWT	JSON Web Token
KM-EP	Knowledge Management Ecosystem Portal
LDAP	Lightweight Directory Access Protocol
LOINC	Logical Observation Identifiers, Names, and Codes
MCU	Main Control Unit
MRAST	Multimodal Risk Assessment Framework
ML	Machine Learning
NLU	Natural Language Understanding
OCI	Open Container Initiative
OKP	Open Knowledge Platform

PDP	Policy Decision Point
PEP	Policy Enforcement Point
PHQ-9	Patient Health Questionnaire
RBAC	Role-Based Access Control
REST	Representational State Transfer
RTPM	Real-time Processing Module
SAMF	Self-assessment and Monitoring Framework
SAML	Security Assertion Markup Language
SAPL	Streaming Attribute Policy Language
SDK	Software Development Kit
SNOMED CT	Systematized Nomenclature of Medicine Clinical Terms
SSE	HTTP Server-Sent Events
SSO	Sign-on and single Sign-Out
UI	User Interface
UX	User Experience
XACML	Xtensible Access Control Markup Language
XAI	Explainable Artificial Intelligence
XP	Experience Points

Table of Contents

1 Executive Summary

The SMILE project, funded by the European Union's Horizon Europe Programme, aims to enhance mental health among young people through a gamified platform. The initiative includes pilot studies in seven European countries to test the platform's functionalities and gather feedback, aiming to provide accessible, evidence-based mental health support to adolescents, parents, researchers, clinical psychologists and all the identified SMILE stakeholders across the EU. The project emphasizes a participatory approach, involving children, adolescents, young adults, their families, teachers, and health professionals in the development and evaluation of the platform. The platform adheres to stringent legal, ethical, and security standards, ensuring data protection and compliance with regulations like GDPR as framed in D3.1.

This deliverable brings together the outcomes of several Work Packages to outline the architecture of the cloud-based software system Open Knowledge Platform (OKP). The platform's architectural design includes several key components:

1. **The Serious Game:** This is a gamified mobile phone application for interactive mental health exercises. Its design engages users through gamification with interactive 3D game world scenarios, comprising the main point of interaction with the end-users. The SMILE serious game features integrate modules for identity and access management, healthcare data storage, and for gameplay data retrieval.

2. **The Awareness App:** This consists of a mobile application and a back-end server. Its purpose is data collection and user engagement through the Experience Sampling Method (ESM) among others. This app supports adolescent user's by teaching situational awareness and appropriate responses that can be integrated into their daily lives. It can deliver survey questions and educational video lessons to the participants. The Awareness App collects user data through semi structured interviews with a chatbot.

3. **The Self-Assessment and Monitoring Framework (SAMF)** for data analysis: This component collects data and outcomes from the two user applications. It has two main sub-components: A natural language chatbot service that can hold conversations with the end-user and a Multimodal Risk Assessment Framework (MRAST). The Users will report their weekly experiences in a semi-structured interview approach.

4. **Two Decision Support Systems (DSSs)** to investigate the viability of supporting clinical psychologists and Pilot Investigators in analysis of data coming from the other SMILE components and decision-making activities. The DSSs can provide cues, risk assessments, and warnings for Anxiety & Depression. It collects and then processes data in real-time using AI and ML

techniques. Its front-end application will provide explainable outputs that can be visualized via a Web UI.

The SMILE OKP has several supporting components for data transit and account management. The back-end server enables secure interactions between the components and data processing.

5. **A FHIR server**, based on the HAPI FHIR Framework, supports the latest FHIR specification for healthcare data exchange, providing structured resources for data transition and safe storage.
6. **The SMILE API** facilitates communication among components, ensuring compliance and security, particularly in healthcare contexts. This API builds the communication bridge between the game client and back-end infrastructure.
7. Attribute Based Access Management (ABAC) is supported through the **Streaming Attribute Policy Language** and Engine **(SAPL)**, providing a concise and user-friendly syntax for defining access control policies.
8. **A Keycloak server** to handle user authentication.

This document, derived from several outcomes of other work packages and provides technical description of the SMILE OKP components, is naturally intended for the project partners, but it also offers valuable insights that can inspire external developers of (mental) health digital tools and platforms, clinical psychologists, researchers, and policy makers.

2 Introduction

Research has demonstrated the potential of digital tools including gamification, Artificial Intelligence, mobile apps, and security technologies, especially as an effective approach to enhance digital mental health paradigm [1-2]. The SMILE project aims to leverage digitalization to deliver disruptive cognitive-behavioural interventions to young people. Its purpose is to improve mental well-being and resilience in youth. The project will develop an Open Knowledge Platform (OKP) that incorporates various digital and self-assessment tools designed to aggregate young's mental data and make it available to specific stakeholders in order to improve cognitive flexibility, self-efficacy, critical thinking, self-regulation, and self-confidence among young users.

The aim of Work Package 5 (WP5) is to consolidate concrete outputs, such as self-assessment tools, gamified interventions, training materials, psychoeducation resources, evidence-based knowledge, user-friendly interfaces (UIs) and decision support systems (DSS), into a unified and coherent package, the so-called OKP. In addition to the OKP, the Knowledge Management Ecosystem Portal (KM-EP) provides external users such as adolescents, parents, healthcare professionals, and policymakers with access to OKP resources, research results, and educational material.

Software architecture encompasses the set of significant decisions about the organization and structure of a software system. It involves the selection and arrangement of software components, their interactions, and the guidelines that govern their design and evolution over time. Essentially, software architecture acts as a blueprint for both the system and the project, facilitating the understanding, development, and maintenance of the software. It addresses both functional and non-functional requirements, aiming to optimize attributes such as performance, security, and scalability. By defining how components fit together and interact within a system, software architecture plays a crucial role in ensuring that a software system meets its intended goals and can adapt to future changes and requirements.

This deliverable D5.1 integrates together the outcomes from Work Packages 2, 3, 4, and 6 to outline the key elements and functionalities of a cloud-based software system, including how these components interact. A structured solution is presented to meet the technical, functional, and operational requirements of specific components: a DSS, an evidence-based knowledge base, and KM-EP. NION collaborates with technical participants to define various architectural views:

- Logical,
- Process,
- Deployment,
- Implementation, and
- Data.

For these components and their interactions with gamification (WP4), data operations (WP6), and the Awareness App (WP6), the objective is to identify needs impacting the solution's structure and minimize risks associated with aligning technical and functional requirements.

Supported by AI-tools, the SMILE system will include virtual self-assessments and self-monitoring all integrated into gamified scenarios. Additionally, SMILE will offer targeted tools for age and gender specific issues.

The platform will finally unlock access to evidence-based knowledge about mental stress to parents, teachers, decision-makers, clinical psychologists, all the relevant stakeholders identified, and increase awareness about young people's needs supporting their mental health. It will be accessible to EU citizens and will be available in the languages of the involved partner institutions.

3 Overview of the SMILE Open Knowledge Platform

The remainder of this document will lay out the following components:

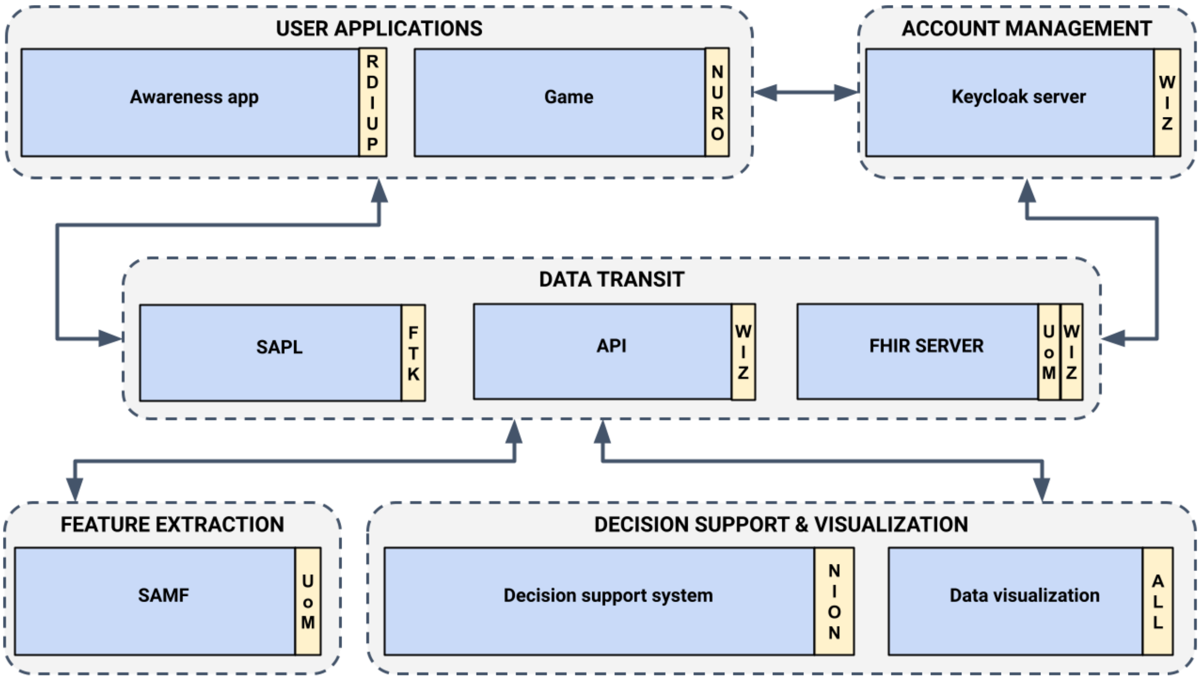

Figure 1: Overview of the SMILE OKP

The overview of the system is presented in the above diagram and contains the following components:

- User applications
    - Awareness App
    - Serious Game
- Account management
    - Keycloak server
- Data transit
    - SAPL
    - API
    - FHIR Server
- Feature Extraction
    - SAMF
- Decision support and visualisation
    - Decision support system
    - Data visualisation and Graphical User-friendly interfaces WebAPP (T5.3).

All components of the system are referred together as the SMILE system or SMILE OKP - Open Knowledge Platform. Each component will be detailed in the sections below.

4 Awareness App

The Awareness App is a mobile application, oriented to support situational awareness of users and collect users' data in safe way. This will enable the timely awareness and response of adolescents with the knowledge that is accessible and can be integrated into their SMILE routine and daily lives. The role of the awareness app is to collect data and information from participants, offer features for thematic discussions and communication, and to provide feedback on their progress and collected markers.

Regarding the data collection side, this application implements the following services:

- Collection of participants' information using standardised instruments,
- Semi-structured interviews for collection of weekly diary video recordings,
- Experience Sampling Method (ESM) for daily assessment for piloting.

On the feedback side, the application displays the conclusions of the collected data in a user-friendly dashboard system. It is also used to notify the users when an activity needs to be carried out. The Awareness App connects with the game via a common semantic data model built on top of the FHIR repository. Technical and implementation details will be found in the deliverables D2.2 and D6.4.

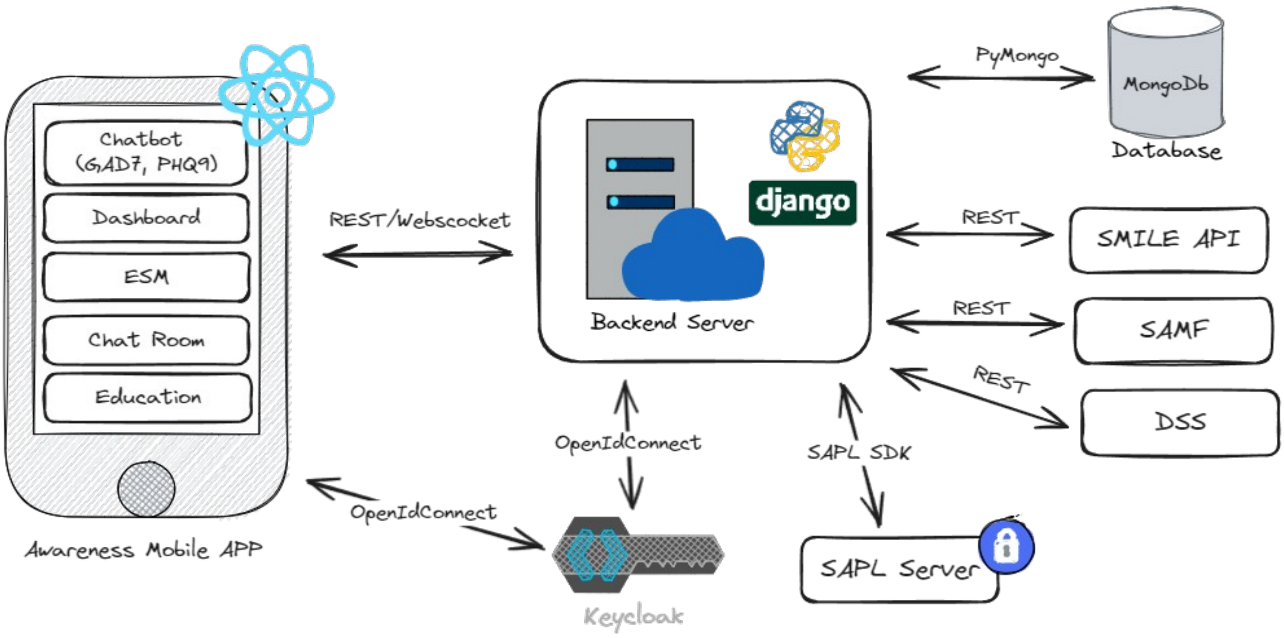

Figure 2: Awareness App architecture

As depicted in figure 2, the Awareness App is composed of two main systems, the mobile application running on end-user device and the back-end server running on the cloud. As part of the SMILE OKP, the Awareness APP will be connected to multiple SMILE components, as follows:

- SAMF: Awareness App will communicate with the SAMF mainly for chatbot questionnaires such as GAD-7, PHQ-9 and for weekly diary recording video.
- SMILE API: As this is the components on top of SMILE data store and also the middleware between components, the Awareness App will use it as mean for saving and retrieving data for the SMILE data store. All communication with these components will follows the REST Standard.
- DSS Module: Awareness App will generate insights and metadata for the DSS and store in the FHIR repository. When allowed, DSS can return feedback to Awareness App.
- SAPL Server: The Awareness APP will use the SAPL python SDK, provided by FTK, to protect sever endpoints against unauthorized access using.
- Keycloak: The Awareness App will use Keycloak for authentication of users in the frontend side and to retrieve user details in the back-end side. The back-end will also use the Keycloak to validate access token received in each request. The communication follows the standard OpenID Connect protocol.

4.1 Mobile APP

A cross-platform mobile application developed by RDIUP with React Native that can run on both IOS and Android. RDIUP will develop user-friendly interfaces based on the outcomes of T2.5 (UX analysis) to increase the acceptance and satisfaction of users. RDIUP will also coordinate the development of the user interfaces in T5.3 to define harmonized styling, colours, views and cards for all interfaces (user applications, web platform and DSS). In the same direction, the UIs will be designed to be also used by society at large.

Attention towards assistive requirements needs and continuous improvement will be maintained during the development process.

4.2 Back-end Server

Developed by RDIUP, it leverages the Django REST framework and adheres to REST API standards to facilitate secure interaction between the mobile app and other SMILE ecosystem components. Recognizing the complexity of the SMILE ecosystem's model, which incorporates rich medical concepts across various tools, we opted for a domain-driven design (DDD) approach. DDD prioritizes programming a domain model with a deep understanding of the domain's processes and rules from domain experts, rather than focusing solely on CRUD (Create, Read, Update, Delete) operations. The system supports HTTPS requests and responses over REST, along with WebSocket for real-time communication.

The back-end will provide multiple resources as follow:

- **Survey**: A survey represents a collection of questions to be asked to participants based on schedule or specific triggers. It can also include required data to be collected from the user when answering the survey. Surveys can be created for ESM, GAD-7, PHQ-9 etc.
- **Survey Answer:** Represent a response of participants to a single survey at a specific time. As surveys can be recurrent, participants can respond to the same survey multiple times, but a new survey answer resource will be created each time.
- **Reward**: Represent collection of SMILE rewards that a given participant earns either in the game or the mobile APP. Reward can be “smiles”, “strike”, “achievement”, “experience token aka XP”.
- **Chat Message**: Represent chat message sent by participant to a discussion room via the Mobile App.

- **Room**: A discussion room details and participants' information. Room can be between adolescent and expert or between adolescents themself with moderator for peer-to-peer support.
- **Lesson**: Represent lifestyle and nutritional education videos that participants can watch in the Mobile App.
- **Quiz**: A collection question used to assess adolescent understanding of the nutritional and lifestyle best practice. Adolescents can earn rewards when providing the correct answers.
- **User details**: Information of the user retrieved directly from the Keycloak server.

These resources objects are input and output of the Awareness app and will be shared by other tools via SMILE API or direct interaction. Input is mainly collected from the Mobile APP and output to the SMILE API mainly.

4.3 Awareness App Prototype

Below are some screenshots of the app user interfaces.

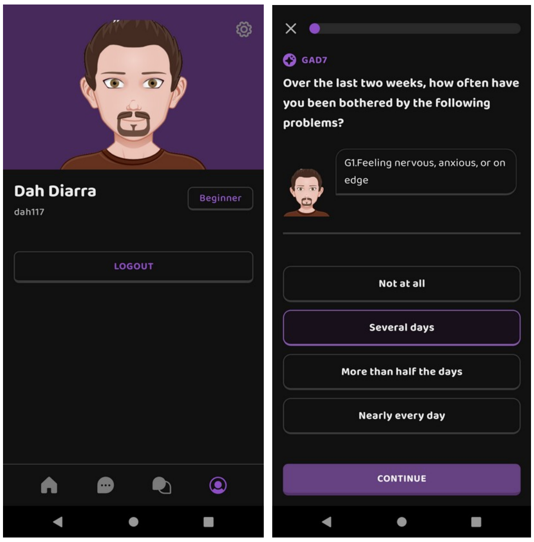

Figure 3: Preliminary Awareness App Interfaces

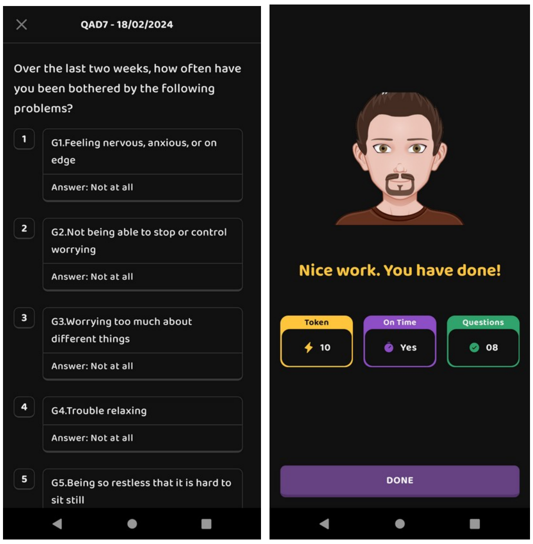

Figure 4: Preliminary Awareness App Interfaces, continued

5 SMILE Serious Game

The Serious Game is a mobile phone application that stands at the forefront of the interaction with the end-users (i.e., young players). The game is designed based on the interactive and engaging gamification scenario aligned with needs and requirements established in collaboration between the consortium technical and clinical teams. The game follows a unique way of combining a 3D world navigation with specific frames/ questions and situations that will guide the user in journey meant to strengthen his/her approach to difficult and awkward situations.

A significant aspect of the game's effectiveness lies in its data-driven back-end system. The middleware developed by WIZ acts as a critical communication bridge between the game client and back-end infrastructure, ensuring secure data exchange and processing. This middleware enables the game to leverage the Keycloak server for user authentication, providing a secure and scalable solution that handles user credentials and maintains session integrity.

Furthermore, the connection to the FHIR database allows for a seamless retrieval of crucial gameplay data. This data includes the player's latest actions, their progress through different game scenarios, and their current state within the game. This integration ensures continuity in gameplay, allowing players to pick up where they left off and resume their journey without losing progress.

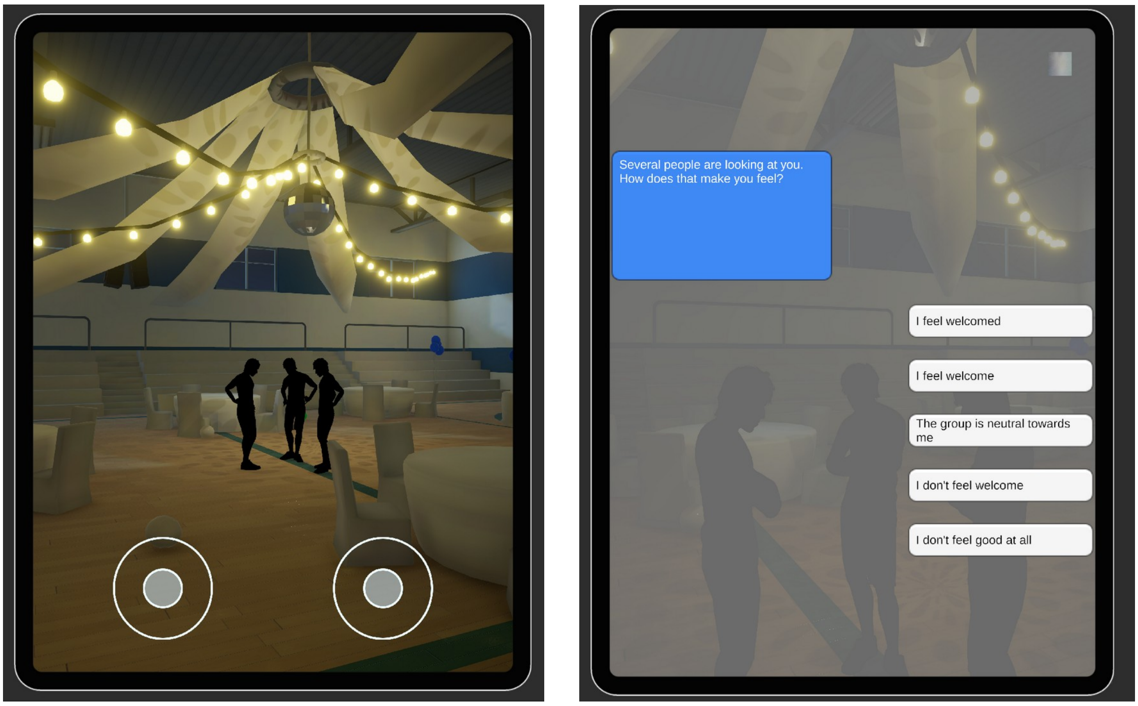

Figure 5: Serious Game Interface

SMILE Serious Game - Data Flow

As shown in Figure 6, the components involved in the Serious Game data flow are the following:

1. Serious Game Application (iOS/Android)

- Output:
  - o Player data and information: This includes data collected from the player during gameplay, such as player actions, scores, progress, and any other relevant player-specific information.
  - o Video snippets and other usage info: This includes video recording device during a gameplay session, screenshots, and other metadata about how the game is being used by the player.
- Recipient:
  - o Middleware

2. Middleware

- Input from Serious Game Application:
  - o Receives player data, video snippets, and other usage information from the Serious Game application.
- Processing:

- The Middleware processes this data, which might involve filtering, aggregating, or preparing the data for further analysis.

- Output:
  - Player checkpoints and progress info, game data: Processed data related to player progress and game checkpoints.
  - Data from the game for analysis: This includes any additional processed data that is required for analysis by the FHIR Back-end.
- Recipients:
  - Serious Game application (for video snippets and usage info), FHIR Back-end (for analysis data)

3. FHIR Back-end

- Input from Middleware:
  - Receives player checkpoints, progress information, game data, and other data required for analysis from the Middleware.
- Output:
  - Feedback or results of the analysis might be sent back to the Middleware for further use or stored for reporting purposes.

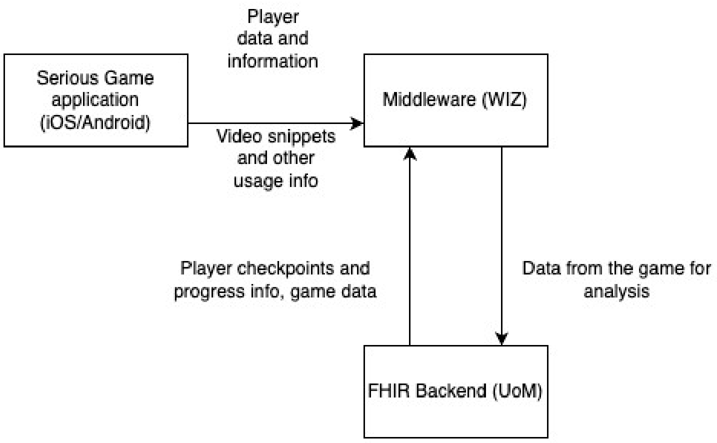

Figure 6: Data flow diagram

6 Keycloak Server

The Authentication and Authorization Infrastructure (AAI) of SMILE consists of SAPL and Keycloak. Keycloak is an open-source solution for Identity and Access Management (IAM), delivering a comprehensive suite of functionalities for user authentication through the exchange of JSON Web Tokens (JWTs) between the client applications and the Keycloak server. It centralizes user authentication and authorization for modern applications and services. Keycloak facilitates single sign-on and single sign-out (SSO), allowing users to access multiple applications with one login and log out simultaneously from all connected services. It supports standard protocols such as OpenID Connect, OAuth 2.0, and SAML, offering features like user federation, strong authentication, user management, and fine-grained authorization. Additionally, Keycloak can integrate with existing LDAP or Active Directory servers and allows customization through its plugin-based architecture. It serves as a centralized platform for managing authentication protocols, user attributes, and access logs across all applications integrated into the data flow.

A Keycloak server is configured through the configuration of realms. A Keycloak realm is a fundamental concept in the Keycloak authentication system, acting as a top-level container for managing a set of users, credentials, roles, and groups. Each realm is isolated from others, meaning it can only manage and authenticate the users that it controls.

The creation of a realm in Keycloak involves setting up general settings and managing users and groups. Keycloak will be mainly employed for configuring authentication to connect to the SAPL lightweight authorization server, to constitute the AAI of the SMILE ecosystem.

7 SAPL

An access control policy is a set of high-level requirements that lay down how access is managed and who is permitted to access information under specific conditions. The Streaming Attribute Policy Language (SAPL) provides a concise and user-friendly syntax for defining access control policies, minimizing the need for extensive and complex code that can obscure the policy's intent, a recurrent issue with other common policy description languages as the Xtensible Access Control Markup Language (XACML). SAPL Server CE and Server LT are software solutions that adhere to secure IT protocols and incorporate various levels of protection and are developed following the Secure by Design Principles, as outlined by CISA's Secure by Design framework [3].

SAPL assesses both requests for access and the entitlements to data resources, thereby offering authorization functionalities across digital platforms. Therefore, it is used to effectively regulate access to the entirety of data within the SMILE system.

SAPL defines an expressive domain-specific language (DSL) for the definition of access policies. Such policies describe circumstanced under which the system shall comply with access requests when they will be denied. These permissions are modelled as attribute-based access control (ABAC): SAPL authorization subscriptions are JSON objects with the attributes subject, action, resource and environment. However, a role-based access control (RBAC) system in which permissions are assigned to a certain role and roles can be assigned to users can be created with SAPL as well.

SAPL’s data model enables sophisticated querying of JSON objects through JSON path-like expressions. Furthermore, SAPL accommodates both conventional request-response policy decision processes and more dynamic, publish-subscribe attribute stream-driven decision-making through a reactive API. The Publish/ Subscribe (Pub/Sub) protocol is a messaging pattern used in software architecture that enables a decoupled, asynchronous communication model between components in a distributed system. In the Pub/Sub model, there are three main components: publishers, subscribers, and a broker (or message broker). The broker acts as an intermediary that manages the communication between publishers and subscribers. It receives messages from publishers, filters them based on the topic, and then routes them to the appropriate subscribers.

A SAPL PDP must expose a publish-subscribe API for subscribing via the subscription objects laid out above. SAPL defines two specific APIs for that. One is an HTTP Server-Sent Events (SSE) API for deploying a dedicated PDP Server; the other for using a PDP in reactive Java applications. The Java API may be implemented by an embedded PDP or by using the SSE API of a remote server.

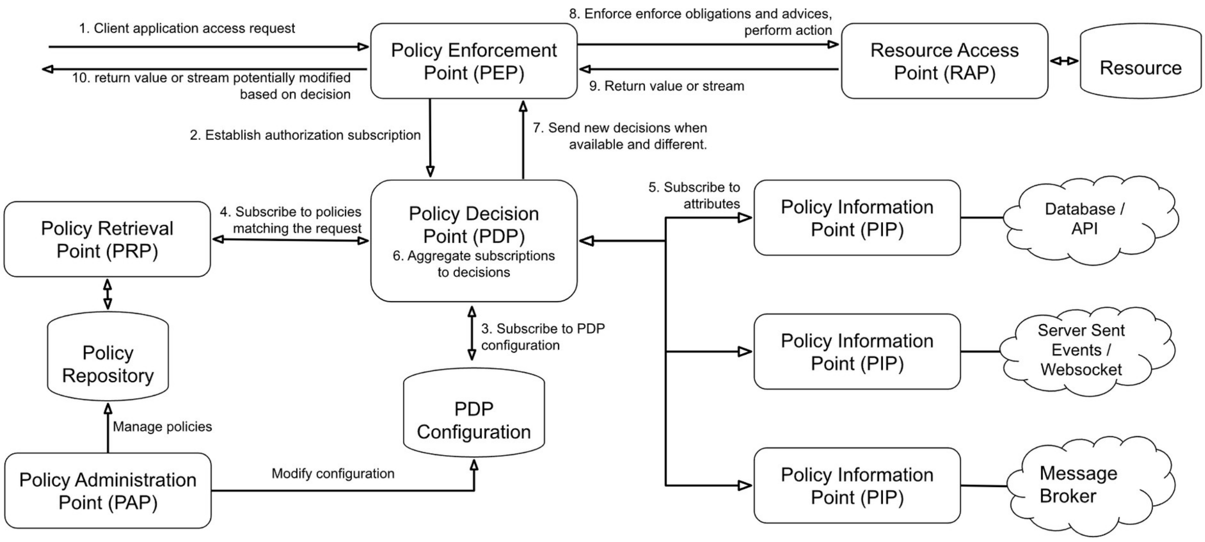

Figure 7: Elements and structure of the SAPL

A SAPL-based lightweight authorization server, functioning as a Policy Decision Point (PDP), is accessible for central policy management. It offers both request-response and subscription-based authorization services via an easy-to-use RESTful API. In ABAC systems, a Policy Enforcement Point (PEP) creates an authorization request, also known as an authorization subscription, and sends it to the PDP. The PDP evaluates this request against established policies and returns an authorization decision to the PEP. Both the request and the decision are structured as JSON objects, which are composed of key-value pairs, commonly referred to as attributes. These attributes have specific names that are predefined within the system's architecture. The PEP is responsible for generating the authorization request and subsequently interpreting the authorization decision provided by the PDP.

SAPL servers may run in a Java Runtime Environment (JRE) or as Open Container Initiative (OCI) images inside containerized virtualisation solutions like Kubernetes or Docker.

7.1 SAPL Server LT – Lightweight Authorization Server

The SAPL server LT keeps track of two folders for PDP configurations and SAPL documents and supports the dynamic updating of policies. These regulate the decision-making processes for current authorization subscriptions.

Full documentation is available at: https://github.com/heutelbeck/sapl-policy-engine/blob/master/sapl-server-lt/README.md

7.2 SAPL Server CE – Lightweight Authorization Server

The SAPL Server CE employs a database to store PDP configurations and SAPL documents. Through the WebUI, administrators have the capability to update policies in real-time, which are then immediately applied to current authorization subscriptions, influencing decision-making processes. SAPL Server CE requires a database like H2 or MariaDB.

Full documentation is available at: https://github.com/heutelbeck/sapl-server/blob/main/sapl-server-ce/README.md

8 API

An API, or Application Programming Interface, acts as a software intermediary that enables two or more software components to interact with each other. The SMILE API comprises a suite of technical resources that facilitate communication among its components. It offers a robust technical framework for interoperability, including the provision of mock data, endpoint examples, and data storage solutions. Additionally, the SMILE API processes each request in accordance with predefined data access policies by interfacing with SAPL.

To ensure compliance and security, particularly in healthcare contexts, the API server is hosted on infrastructure that meets stringent certification standards for healthcare applications. During the development stages, the API will reside on a classic cloud-based machine but once the API calls are created, identified and categorized based on processing needs, the plan is to decentralize the infrastructure behind key data endpoints using a container orchestration system to allow flexible allocation of resources based on demand.

From a technical response perspective, during the project development stages, the SMILE API will also offer a series of mock data endpoints, allowing quick response from components that are not yet developed. This will of course lead to hundreds of versions for the endpoints, but it will help speed up the development process in a project that has to function as a single entity but guided by multiple tech companies working in parallel.

The API will provide access to a Swagger interface (user interface) for testing of end points that will provide examples and validation for each endpoint. The interface will be available to all technical partners using their pre-existing allocated credentials from the Keycloak server.

A special component in the API will be the game middleware that technically will behave like all the other API collections but physically will be deployed on a separate machine in the same server infrastructure. This is needed because the requests from the game are more frequent and the size of the data transfers is bigger (video streams, more frequent data structure changes, etc.)

At this point in the project, we are sure that the API will have tens or hundreds of endpoints that will be further detailed in future technical deliverables. In the figure below, there is a representation on how the API is categorizing the endpoints and the requests.

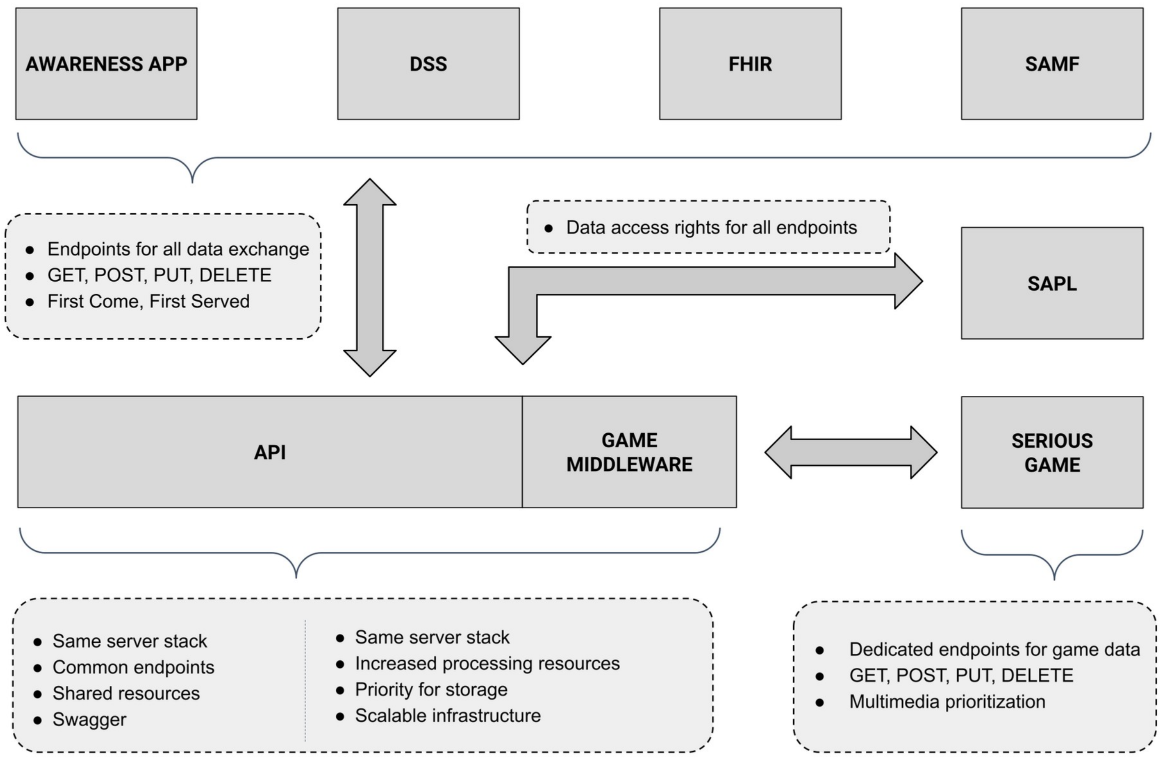

Figure 8: Structure of the SMILE API

Considering the ongoing process, below there are 2 screenshots from the SMILE API UI showing some of the developed endpoints.

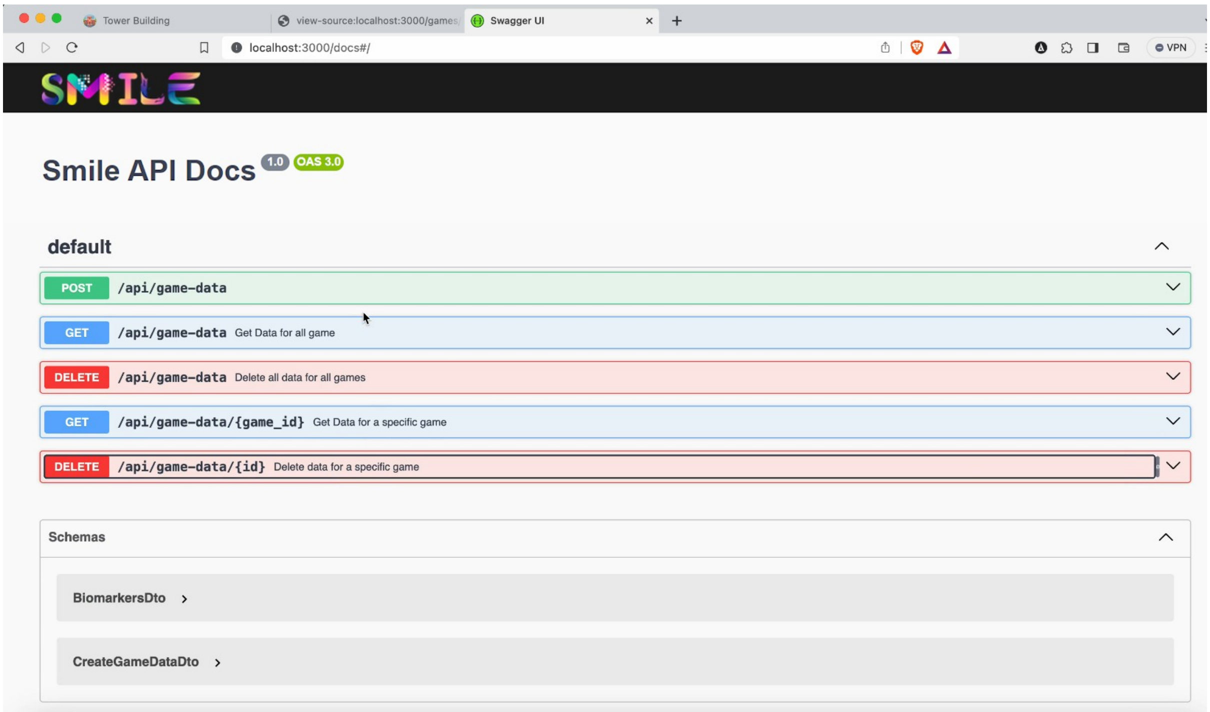

Figure 9: Web view of an API endpoint

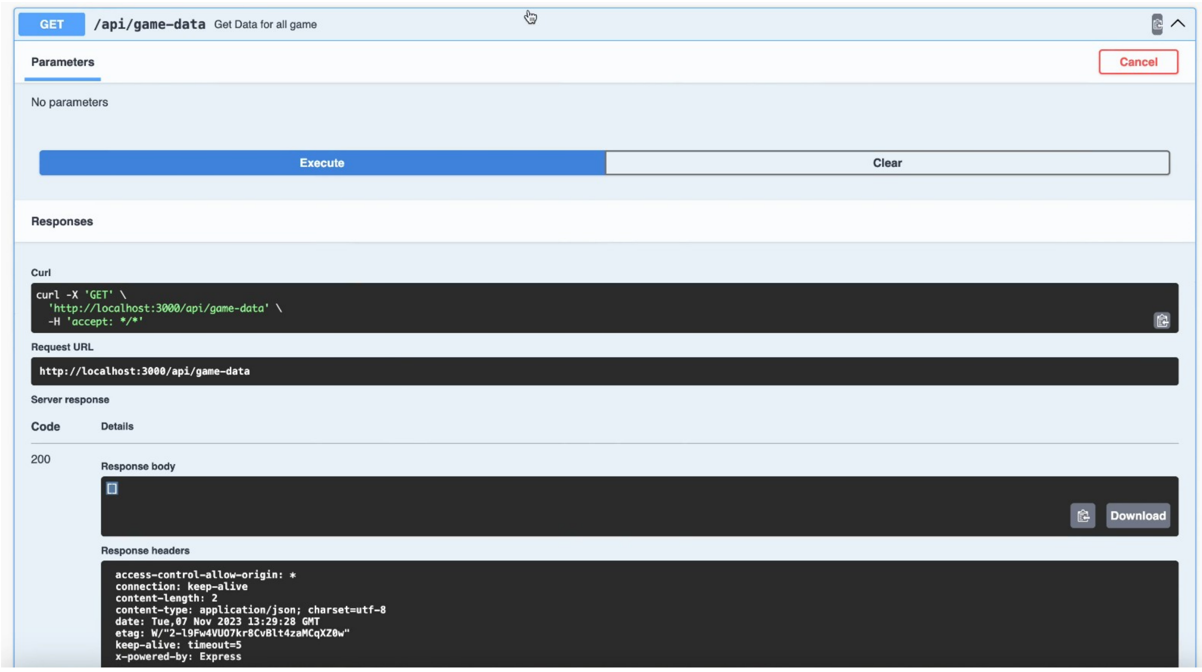

Figure 10: Interactive view of an endpoint's functionality

9 FHIR Server

HL7 FHIR, which stands for Health Level 7 Fast Healthcare Interoperability Resources, is an HL7 specification designed for healthcare interoperability. FHIR provides a standard for exchanging healthcare information electronically, aiming to streamline data exchange and integration across various healthcare systems. FHIR is built on modular components called "Resources," which can be easily assembled into systems to address real-world clinical and administrative challenges efficiently. This standard combines the strengths of HL7's version 2, version 3, and Clinical Document Architecture (CDA) product lines while incorporating the latest web standards to enhance implementability [4].

In SMILE, we deploy the FHIR Server based on the open source HAPI FHIR Framework. It provides a robust set of tools, libraries, and APIs that simplify the development of FHIR-based applications. HAPI FHIR supports the latest FHIR specification and offers compatibility with previous versions, ensuring smooth integration with existing systems. It includes server and client APIs, enabling developers to build FHIR-compliant applications that can either consume or provide healthcare data. HAPI FHIR also supports data persistence, allowing healthcare organizations to store, retrieve, and query FHIR resources efficiently. Additionally, it incorporates powerful terminology services, enabling seamless integration with standard code systems, such as Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) and Logical Observation Identifiers, Names, and Codes (LOINC). HAPI FHIR provides validation and conversion capabilities, ensuring the integrity and compatibility of FHIR resources [5].

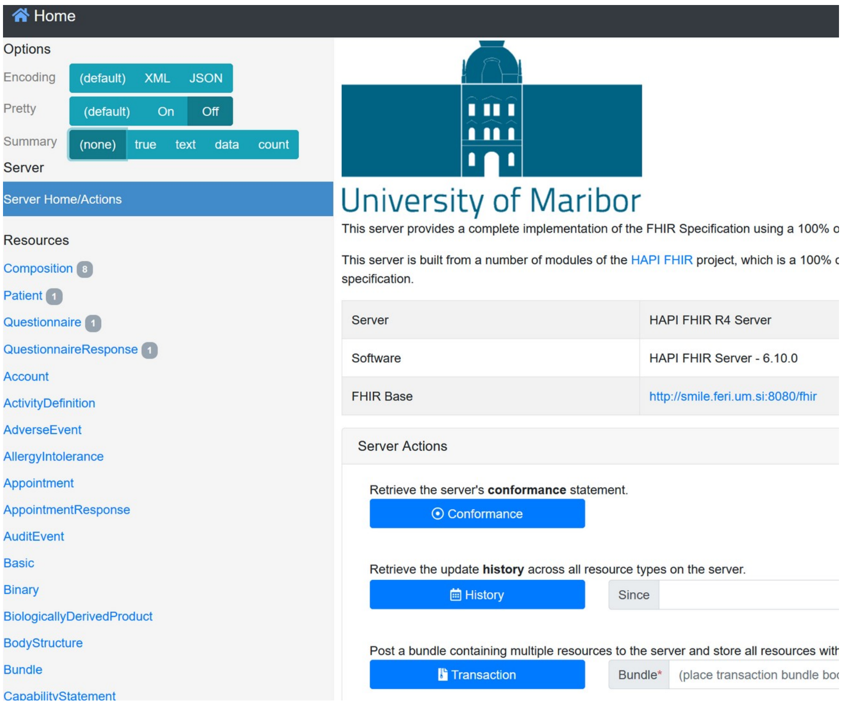

Figure 11: Implementation of HAPI FHIR Server for project SMILE

With HAPI FHIR, we enable a built-in RESTful client for connecting to FHIR REST servers. We use the Annotation client which relies on static binding to specific operations to give better compile-time checking against servers with a specific set of capabilities exposed. This client requires more effort to set up but can be particularly useful for complex operations and when the person defining the specific methods to invoke is not the same person who is using those methods.

Our HAPI FHIR client abstracts the underlying wire format JSON with which the resources are inserted and/or retrieved. The HAPI FHIR library handles the parsing and encoding of JSON messages, allowing the SMILE team to focus more on the logic of the application.

The inputs of the server are:

- FHIR Server URL: The URL of the FHIR server one is connecting to. This is used to establish communication between your application and the FHIR server.
- FHIR Resources: FHIR resources are the primary inputs when working with HAPI FHIR. These resources represent various healthcare-related concepts, such as patients, procedures, observations etc. Specific resources to be used will be defined under the Common Semantic Data Model delivered in D6.1.
- Search Parameters: When querying a FHIR server, one can specify search parameters to filter the results. These parameters are defined in the FHIR specification and can be used to retrieve specific subsets of data.
- FHIR Version: The version of the FHIR standard one is working with, such as R4, R5, etc. This is important as different versions may have different resource structures and capabilities. In SMILE, we utilize R4 version of the server.

The outputs of the server are:

- FHIR Resources: The primary output when working with HAPI FHIR is the FHIR resources, which can be in JSON or XML format. These resources contain the data you requested from the FHIR server.
- Search Results: When querying a FHIR server, the output will be a list of resources that match the specified search parameters.
- Validation Results: HAPI FHIR provides validation capabilities to ensure the integrity and compatibility of FHIR resources. The output of the validation process will be a report detailing any issues found with the resources.
- Conversion Results: HAPI FHIR also supports converting between different FHIR versions or data models. The output of the conversion process will be the converted FHIR resources.

10 SAMF

SAMF stands for Self-assessment and monitoring Framework. This is a framework of services to extract observable cues and user-reported outcomes. The framework does not provide a user interface, instead it provides a series of services that will be analysing the data that comes from the two user applications (Serious Game and Awareness App). It consists of multiple components dedicated to feature extraction and risk assessment. Similarly, as in with FHIR server, the framework provides a java based RESTful API (i.e. the UM's REST API), based on Spring Boot [6] to connect with the rest of SMILE components. For the development and integration processes, we also offer a Swagger-based [7] interface for seamless access to the hosted services.

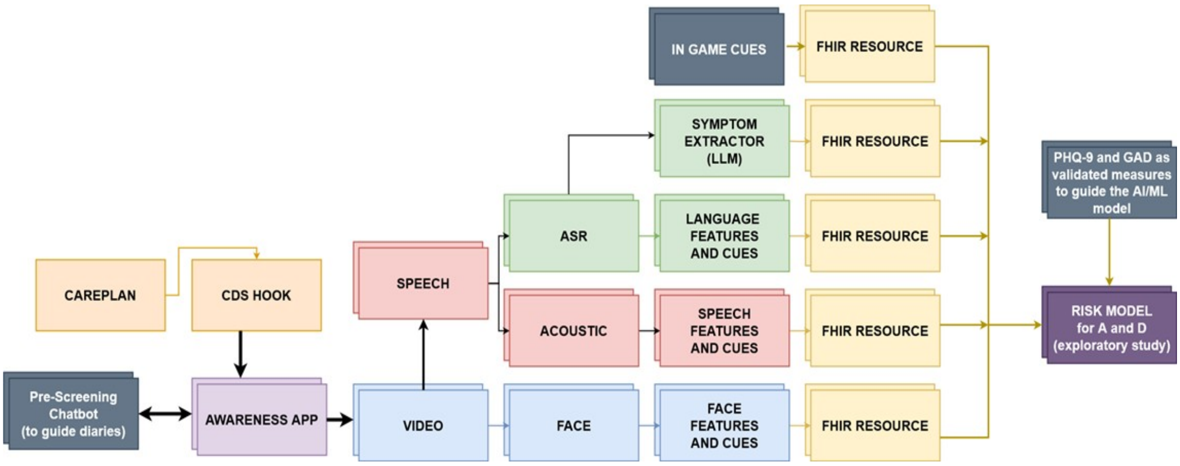

Figure 12: A Modular Architecture of the SAMF Framework.

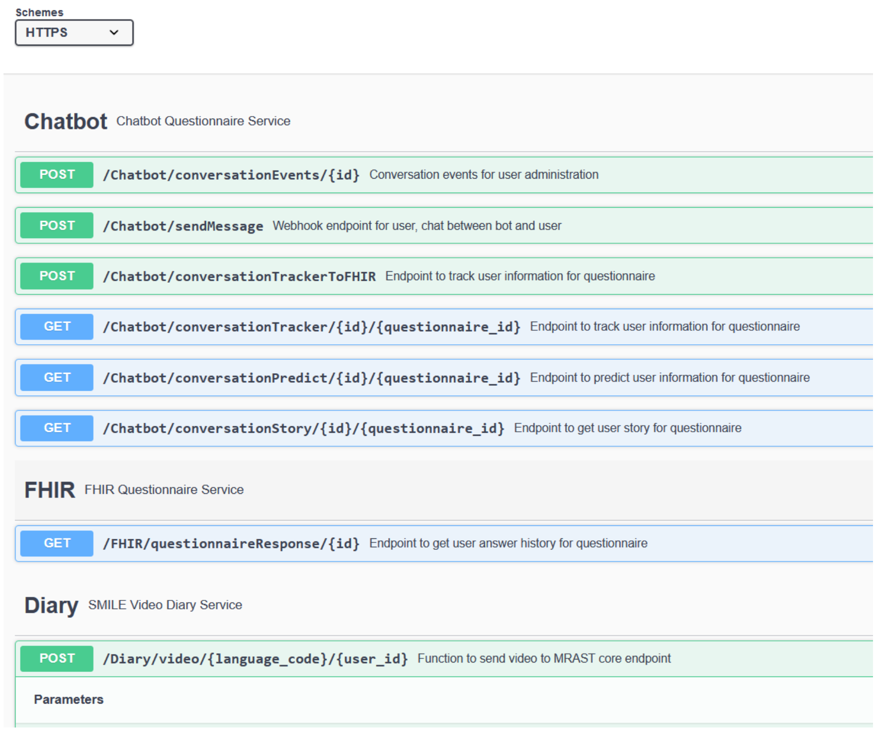

Figure 13: The UM's Swagger Interface for SMILE project

From an architectural point of view, it consists of two main frameworks, the Chatbot Framework (CB) and Multimodal Risk Assessment Framework (MRAST).

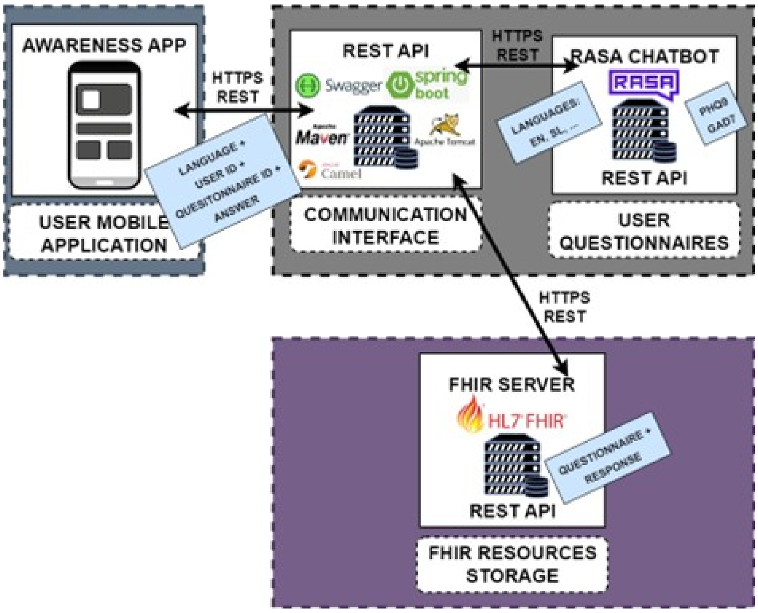

Figure 14: The implementation of the CB framework for standardized data collection using validated instruments

The main component of the CB is the Rasa-based chatbot. Rasa is an open-source conversational AI platform that enables developers to build chatbots and virtual assistants capable of understanding and holding conversations with users. It provides tools for natural language understanding (NLU) and dialogue management, allowing developers to create contextual assistants on different messaging channels and third-party systems through a set of APIs. In SMILE, we use the story-based approach for defining conversational workflows, where UM and pilot partners define a series of expected user inputs and system responses in a structured format. This allows the creation of contextual and personalized conversations, as well as gives us the ability to handle complex dialogue scenarios (e.g., the Diary recording).

Inputs to the framework:

- User inputs: Natural language utterances that the user types to the chatbot,
- Language: The targeted language of the end-user,
- Stories: Dialogue flows that represent a conversation between a user and the chatbot, used to train the dialogue management models.

Outputs of the framework:

- Chatbot responses: Natural language utterances generated by the chatbot based on the user's input and the learned dialogue models,

- Actions: Specific tasks or functions that the chatbot performs in response to the user's input, such as retrieving data, updating records, or invoking external services,
- Updated context: The chatbot's memory of the conversation, which is used to maintain context and personalize the conversation based on the user's history,
- FHIR Questionary Response: a structured set of questions and their answers. The questions are ordered and grouped into coherent subsets, corresponding to the structure of the grouping of the questionnaire being responded to.

Rasa also supports the integration of custom NLU models, allowing developers to tailor the chatbot's language understanding capabilities to their specific use case. In SMILE, we exploit the component to drive extraction of real-world data from diary recordings.

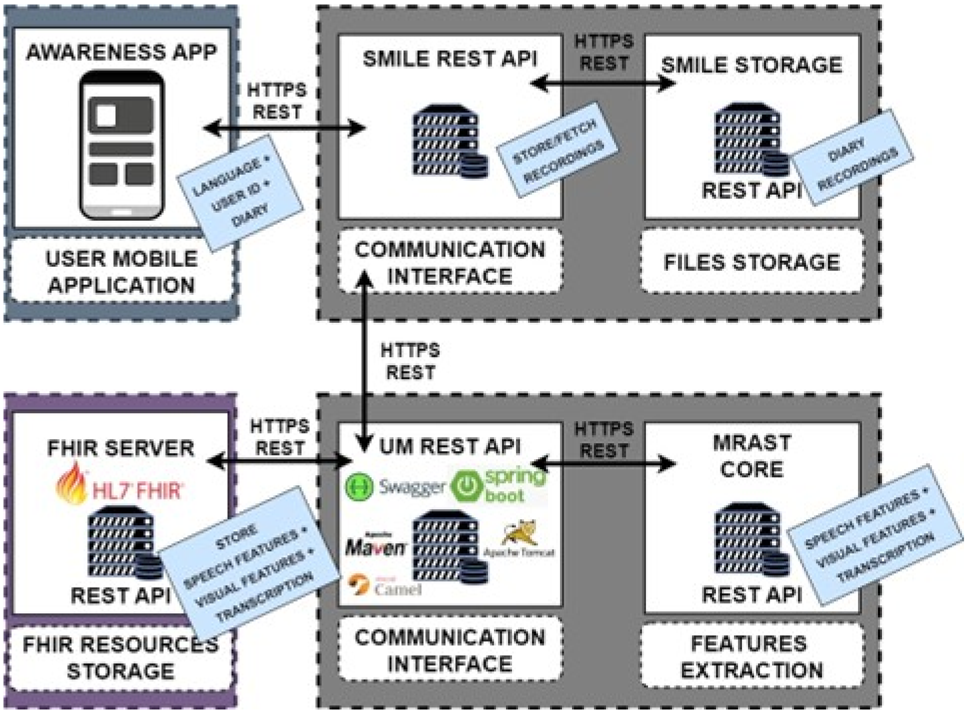

Figure 15: The implementation of the MRAST Framework for extraction of real-world data from diary recordings

In this case, the chatbot drives the Q&A session during which, end-users record their weekly experiences using a semi-structured interview approach, i.e. by answering 7 questions. The videorecording is encrypted and temporary stored on SMILE’s Secure Data Infrastructure and linked with a FHIR Media Resource. On demand (or based on allocated schedule), the MRAST framework, downloads, decrypts the video, extracts the observable cues and deletes the videorecording. The results are stored as FHIR Resources.

Inputs to the framework:

- Link to the FHIR video-resource (includes physical, location, date of recording, language, and subject's internal identifier).

Outputs of the framework (stored a set of FHIR Resources):

- Observable Cues (from speech, language and facial expressions),
- Extracted Symptoms,
- Link to the FHIR Resources generated.

Figure 16: The output generated by MRAST Framework and stored as FHIR Compositions.

11 Decision Support System and Data visualization

The DSS will be an informative and assistive tool designed to support clinical psychologists, Pilot Investigators, players and/or caregivers in taking informed decisions regarding children and adolescents' mental health situations. Its primary objective is to enhance individual care, streamline the research process, and evaluate the overall effectiveness of the SMILE OKP.

The DSS will operate by providing valuable outputs in various formats, including recommendations, warnings, cues, risks, and customized instructions for the Serious Game. To achieve this, the DSS will use data inputs sourced from the Awareness App, the SAMF, the FHIR Server, and the Serious Game, thereby leveraging real-time and assessed information to guide decision-making processes effectively. A general view depicting how the DSS will interface with other SMILE components is shown in Figure 17.

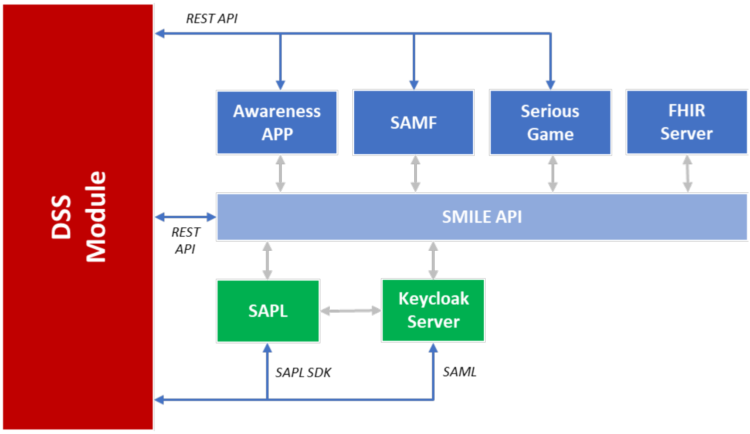

Figure 17: DSS interfaces with other SMILE components

The DSS module will utilize the SAML protocol to authenticate users via the Keycloak Server. It will also employ the SAPL module's SDK to enforce access control policies, ensuring the security of SMILE's functionalities. Additionally, data retrieval and storage will occur via endpoints that access the SMILE API, which interfaces with the general data storage infrastructure.

On the other side, the DSS module will provide a REST API to call the different existing DSS from other SMILE components. This approach enables the generation of decision-making output in response to events and data from the Awareness App, the SAMF, the Serious Game, the FHIR Server, or the SMILE API because of significant data changes within the database. Optionally (and additionally), DSS calls could be scheduled over time, but using endpoints a smoother coordination and better event and data synchronization will be ensured.

Looking now inward at its architecture, the SMILE DSS module is envisioned as a holistic system crafted to streamline decision-making procedures. It will achieve this by harmonizing diverse software components, each particularly tailored to align with the unique objectives of the project it supports. Below, we delve into a more elaborate depiction of the DSS.

- **Back-end Application**: The back-end part of the DSS is designed to encompass the system's DSS core functionality, primarily embedded within the Main Control Unit (MCU). Within this back-end layer, various modules are incorporated to fulfil specific roles and tasks:
    - *Main Control Unit (MCU):* This module will serve as the central module housing the "intelligence" of the system which can provide cues, risks, and symptomatology data, and/or tailored recommendations and warnings to clinicians. This functionality will be based on pre-configured knowledge (e.g., clinical evidence-based information) and will implement AI and ML techniques for enhanced decision-making support.

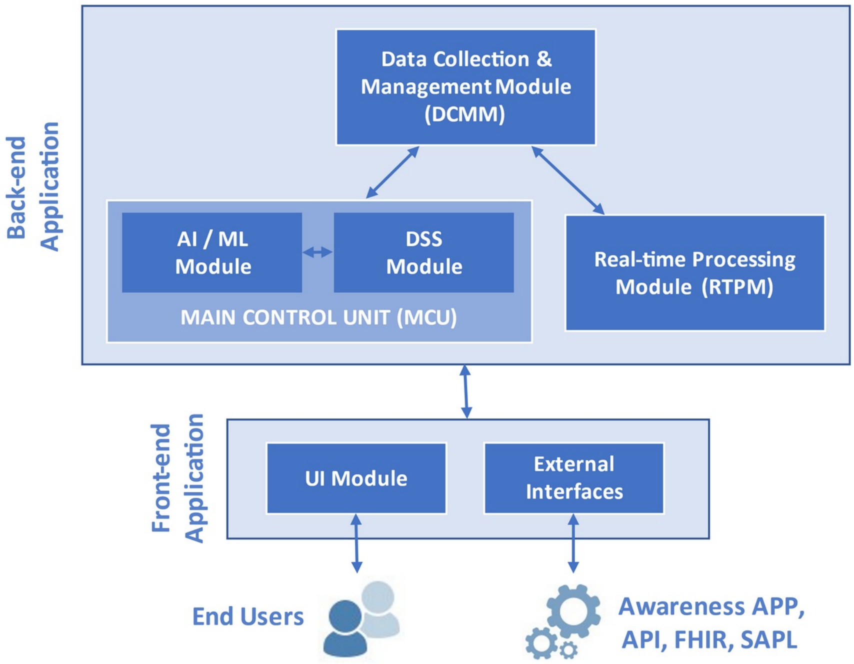

Figure 18: Back-end and front-end applications of the SMILE DSS

- *Real-time Processing Module (RTPM):* This module will be dedicated to enhancing the data provided by other components within SMILE. Its primary function will be transforming this data into more comprehensive information, facilitating rapid and straightforward processing of data streams to execute desired functionalities. Leveraging data and information fusion techniques, the RTPM will improve online processing, decrease response time, and enhance performance and management strategies.
- *Data Collection and Management Module (DCMM):* This module will play a pivotal role in gathering, organizing, and storing data within the DSS. It will manage both real-time and temporary data storage, facilitating reporting, analysis, and the development and execution of the DSS within the MCU. Employing a cloud-based database architecture, the DCMM will integrate optimized data management techniques, including automated data indexing and classification/regression algorithms to filter data before its insertion into the database.

- **Front-end Application**: The WebAPP (developed by RDIUP including the Knowledge Graph visualization provided by NVISION) will serve as the interface for PIs (Pilot Investigators), handling visualization, notifications, and event reporting. It will provide a functional interface to interact with the system, empowering clinicians with a smooth experience. Data models including knowledge graph will be carefully designed to facilitate the implementation of a human-machine interface, enhancing ergonomics and decision-making explainability by offering deeper insights. Moreover, it will orchestrate seamless interactions with the other SMILE components such as the Awareness App, the SMILE API, the FHIR server, and the SAPL, ensuring smooth data flow and system integration.

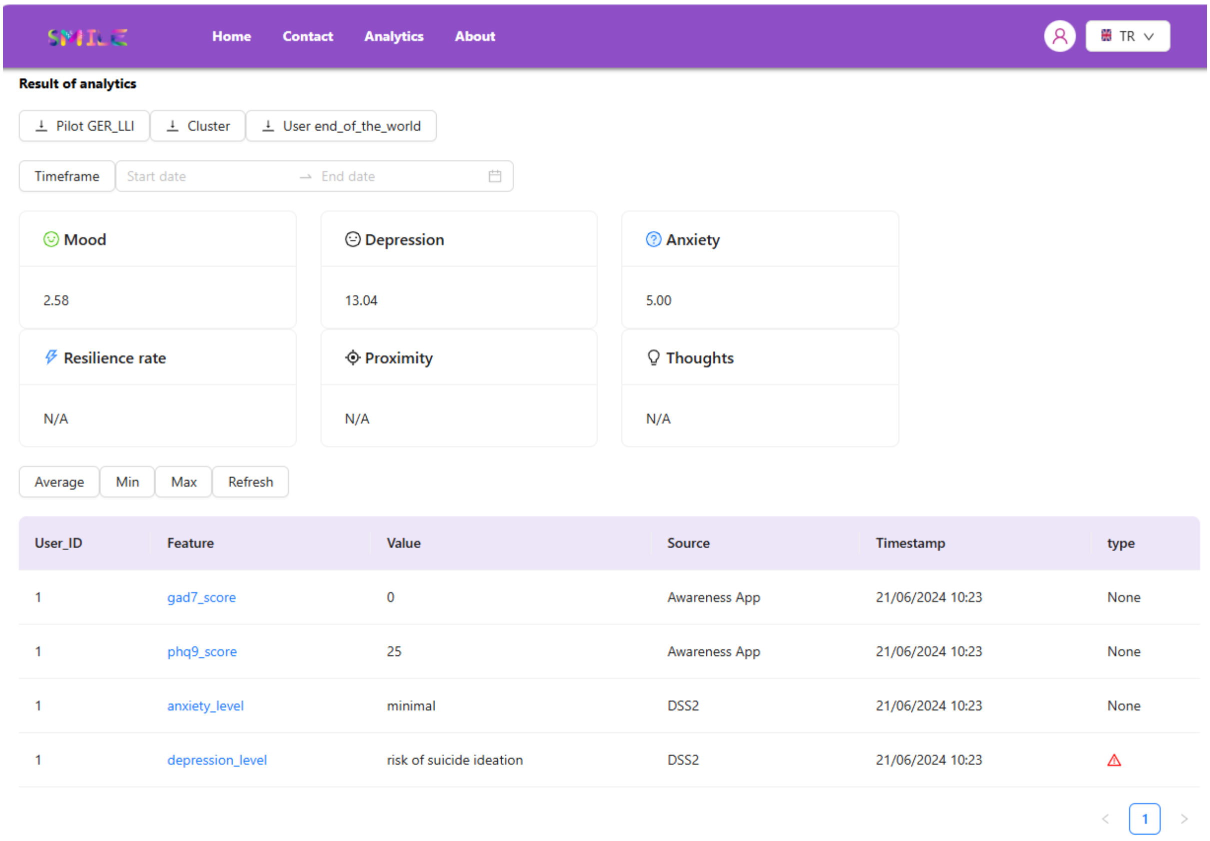

Figure 19: WebAPP co-designed in SMILE

Functionally speaking, within the SMILE project, the DSS will establish the connections and relationships between digital inputs, outputs and parameters to automatise the actions and generate additional insights (or evidence) to the clinicians. From the analysis of players' cues, risks, features and/or symptoms sourced from the Awareness App, the SAMF, the Serious Game, and/or the FHIR Server, two different DSS will be executed:

1) **DSS1** for managing Serious Game parameters and data. After the completion of the piloting phase and the processing of the generated games data (i.e., modules completed, score, competency points, meta-skills points, task completed, levels, number of attempts, choices, path taken, avoided, specific to scenarios responses or exercise, etc.) we will explore potential inferences and/or correlations and/or relationships and/or causality between the games outputs and other available data from the Awareness app and clinical knowledge. Linear regression and Causal Bayesian Network (CBN) will be targeted to develop DSS1. CBN, which is a graphical representation to highlight causal knowledge, and other graph analytics can be employed to intricate relationships and dependencies among entities such as cues, features, risks, and symptoms. By utilizing graph algorithms like centrality, community detection, and path analysis, the DSS would be able to pinpoint key influencers, uncover patterns, and facilitate more informed decision-making based on network insights. The outcome expected from DSS1 will be agreed with the clinicians and with NURO, the developers of the Serious Game. It will be mostly feedback for the game and for players, such as changes in the scenario, game level and difficulty, or characters interaction.

2) **DSS2** for players stratification based on the risk to develop anxiety and depression. DSS2 aims to classify the players into well distinct and defined categories (i.e., minimal/mild/moderate/severe anxiety; minimal/mild/moderate/moderately severe/severe depression). Its outcomes will include flagging players into various groups, issuing alerts based on identified risks, and furnishing warnings and recommendations. Suicide risk monitoring will be also investigated using standardised questionnaire (i.e., a specific item at the PHQ-9). While decision trees are an appropriate choice to deal with the Awareness App data inputs (ESM, GAD-7, PHQ-9 etc), other ML candidates such as Random Forest Support Vector Machine or Relevance Vector Machine could be also targeted to develop the first layer of DSS2. Given that the players will have the possibility to answer the questionnaire in the Awareness App on unlimited basis (Weekly, on-demand, 1-4 times a day, etc), DSS2 will have to generate an output every time the players answer the questionnaires. The second layer of DSS2, which will be developed based on a transparent and explainable AI framework utilizing a Belief Rule-Based Expert System (BRBES), will yield a comprehensive risk stratification output considering the sub-output of the first layer. The Clinicians/Pilot Investigators will receive these outputs, overseeing their credibility while supervising and monitoring their implementation.

The primary aim is to craft a DSS module that assists clinicians in making more informed decisions and enriches the influence of the Serious Game on players. This tool will harness advanced technologies and methodologies to provide robust support for navigating complex decision-making processes. Moreover, it will facilitate interaction with evidence-based knowledge resources, enhancing decision-making with reliable information and insights. Finally, a WebAPP user-friendly interface is under development for the visualization and analysis, essential for making DSS outputs comprehensive for stakeholders, especially clinical psychologists. When applied to DSS, the WebAPP UI for Explainable AI (XAI) becomes even more crucial in facilitating informed decision-making processes. Transparency and interpretability will be the cornerstones of the WebAPP design, providing clinicians with clear insights into the rationale behind DSS recommendations, warnings, and overall output, while ensuring a thorough understanding of its capabilities and limitations.

Tailored explanation visualizations within the UI will present complex data relationships and decision pathways in a visually intuitive manner, catering to the specific needs of DSS users. Integrating interpretability metrics into the UI will offer clinicians quantifiable measures of DSS reliability and transparency, enabling them to assess the trustworthiness of DSS outputs and make informed decisions confidently.

Moreover, the WebAPP will contextualize DSS outputs by offering explanations of relevant terms, concepts, and domain-specific knowledge. This would empower clinicians to interpret DSS outputs with confidence and discernment, enhancing their understanding of the information presented. By fostering a culture of user engagement and collaboration through feedback mechanisms, the UI will encourage clinicians to contribute insights and refine DSS performance iteratively. Through collaborative efforts, the UI for XAI in DSS will promote transparency, trust, and empowerment in decision-making processes, ultimately enhancing users' awareness, research outcomes and adolescent well-being.

12 Conclusions

The document integrates results from several work packages to present the architecture of the cloud-based software system, the Open Knowledge Platform (OKP) to support mental health related learning and guidance and data analysis.

The OKP includes several key components:

- **Serious Game**: A gamified mobile application for interactive mental health exercises.
- **Awareness App**: A mobile application for data collection and user engagement through methods like the Experience Sampling Method (ESM).
- **Self-Assessment and Monitoring Framework (SAMF)**: For data analysis, including a natural language chatbot service and a Multimodal Risk Assessment Framework (MRAST).
- **Decision Support Systems (DSS)**: To support healthcare professionals in decision-making activities, providing cues, risk assessments, and warnings.
- **Keycloak Server**: An open-source solution for identity and access management, providing user authentication and authorization through JSON Web Tokens (JWTs). It supports single sign-on (SSO) and integrates with LDAP or Active Directory servers.
- **Streaming Attribute Policy Language (SAPL)**: Provides a user-friendly syntax for defining access control policies, ensuring secure data access and management.
- **Application Programming Interface (API)**: Facilitates communication among components, ensuring compliance and security, particularly in healthcare contexts. It includes endpoints for data exchange and interaction with other components.
- **Fast Healthcare Interoperability Resources (FHIR) Server**: Based on the HAPI FHIR Framework, it supports the latest FHIR specification for healthcare data exchange, providing structured resources for data transition and safe storage.

Prototypes for the Serious Game and Awareness App have been developed and are being refined. Technical and clinical partners are collaboratively developing additional scenarios. The API and Self-Assessment and Monitoring Framework are under continuous development. Decision Support Systems are in the planning stage with early prototypes using synthetic data. Security features like SAPL and the Keycloak server are prototypically integrated for existing endpoints, and work on the FHIR database is nearly complete.

13 References

[1] L Aschentrup et al., Effectiveness of gamified digital interventions in mental health prevention and health promotion among adults: a scoping review. BMC Public Health. 2024 Jan 2;24(1):69.

[2] X Lin et al., Scope, Characteristics, Behavior Change Techniques, and Quality of Conversational Agents for Mental Health and Well-Being: Systematic Assessment of Apps. J Med Internet Res. 2023 Jul 18;25:e45984.

[3] Cybersecurity & Infrastructure Security Agency: Secure-by-Design. Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software. 25 October 2023. Online available: https://www.cisa.gov/resources-tools/resources/secure-by-design

[4] National institute of Standards and Technology: NIST SP 800-162. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. Information Technology Laboratory, 08 February 2019. Online available: https://csrc.nist.gov/pubs/sp/800/162/upd2/final HL7 FHIR: https://www.hl7.org/fhir/overview.html, last visited April 2024

[5] HAPI FHIR: https://hapifhir.io/, https://hapifhir.io/

[6] Spring Boot: https://spring.io/projects/spring-boot

[7] SWAGGER: https://swagger.io/

14 Contact

Project Coordinator: Dr. Dominic Heutelbeck

FTK e.V. - Forschungsinstitut für Telekommunikation und Kooperation

Wandweg 3, 44149 Dortmund, 0231 - 975056-0

Dortmund, 0231 - 975056-0

www.horizonsmile.eu

@HorizonSmile

www.linkedin.com/company/HorizonSmile/

@HESmile_project

Funded by the European Union under Grant Agreement No°101080923. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.