SELFSUSTAINED CROSS-BORDER CUSTOMIZED CYBERPHYSICAL SYSTEM EXPERIMENTS FOR CAPACITY BUILDING AMONG EUROPEAN STAKEHOLDERS

Research Innovation Action

Project Number: 872614 Start Date of Project: 01/01/2020 Duration: 48 months

DELIVERABLE 5.1

SMART4ALL Technology Portal

Dissemination Level	**Public**
Due Date of Deliverable	**April 2020, M04**
Actual Submission Date	**30/6/2020**
Work Package	**WP5, Technology Enablement and SMART4ALL Marketplace**
Task	**T5.2, Technology Portal Development and Maintenance**
Lead Beneficiary	**UoP**
Contributing beneficiaries	**BTU CS, AVN**
Type	**DEM**
Status	**Submitted**
Version	**03/E**

Co-funded by the Horizon 2020 programme of the European Union

Abstract

The deliverable is the SMART4ALL Technology Portal, which is prepared in the context of Task 5.2 entitled "Technology Portal Development and Maintenance". The main objective of this document is to present the Marketplace technological infrastructure, which comprises one of the main pillars of SMART4ALL project. Additionally, apart from the technological aspect of this endeavor the goal of this document it to present the design and development approaches adopted and exploited as well as the services that are offered through the Marketplace and that will evolve and grow as the project progresses. Furthermore, a critical aspect of the work presented here is the tools that will be offered to the winners of the Open Calls in the context of SMART4ALL, facilitating project management, and supported by SMART4ALL partners. Finally, the design and development of a dedicated communication and computation infrastructure for the SMART4ALL purposes required considerable effort thus is also included here.

History and Contributors

Ver	Date	Description	Contributors
00	01/06/2020	Document structure	UoP
01/D	23/06/2020	Circulate the first complete version of the document for internal review	UoP, BTU CS, AVN
02/R	28/06/2020	Comments received from internal reviewers	DLR, PUT
03/E	30/06/2020	Final version completed	UoP, BTU CS, AVN

Executive Summary

This is an eight-section document providing all required information concerning the design and development of the Marketplace, the respective services offered, and the Tools offered to the projects that will be funded through SMART4ALL open calls based on the following order.

After an introduction in Chapter 1, Chapter 2 offers a high-level presentation of the Marketplace design approach adopted. This includes a presentation of the state-of-the-art open source technologies exploited after a careful elicitation process, which will ensure flexibility, extensibility, maintainability, and efficiency of the Marketplace as an ICT infrastructure long after the end of the SMART4ALL project.

A critical design choice to ensure the aforementioned characteristics is to clearly separate the frontend graphical user interface from the rest of the infrastructure which is presented in Chapter 3. Here a presentation of the functionalities offered to the end-user in this first version of the GUI is presented as well as operation modes supported.

Following the same design concept Chapter 4 focuses on the Backend Infrastructure as a critical technical infrastructure enabling the efficient Marketplace data/artefacts storage, categorization, query processing, data model support, and interfacing with 3rd party solutions.

A critical aspect of the Marketplace as well as the SMART4ALL as a project is the provision of efficient MatchMaking. Therefore, a separate chapter (Chapter 5) is devoted to presenting both what is currently developed and offered at the time this document is prepared as well as the planes for the next periods.

Another aspect that is offered a distinct section in this document due to its importance, is the Data Model defined and implemented for the artefacts that will be hosted and offered through the Marketplace. In this context Chapter 6 offers a clear presentation of the data model design approach at the time of the document preparation. It is noted that extensions, adjustments, and modification may be made based on information and experience acquired as SMART4ALL progress.

Chapter 7 presents the tools and services that will be offered by SMART4ALL to the projects that will be funded by SMART4ALL to facilitate respective project management.

Chapter 8 presents the main design approaches, technologies involved and components comprising the SMART4ALL communication and computational infrastructure upon which all services current, as well as future, will be based. This is followed by a brief Chapter 9 conclusions section summarizing the main points.

Abbreviations and Acronyms

API	Application Programming Interface
DB	Database Management
JSON	JavaScript Object Notation
REST	Representational state transfer
NLP	Natural Language Processing
AI	Artificial Intelligence
IoT	Internet of Things
CF	Collaborative Filtering
GUI	Graphical User Interface
AI	Artificial Intelligence
MQTT	Message Queuing Telemetry Transport
API	Application Programming Interface

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Purpose and Scope

The purpose of this deliverable is to serve as an accompanying document to the technological infrastructure and services offered by the SMART4ALL highlighting all critical design and development aspects that are not clearly visible by the developments themselves as well as present the main intended functionalities.

1.2 Approach

To achieve the purpose and scope of this documentation, the main components of the Marketplace and the Tools are clearly segmented and correspond to a specific chapter. Thus, the reader can have a clear idea of the design of the infrastructure, the characteristics of each component, the functionalities, and roes of each component and the technologies involved in each component and therefore have an objective appreciation of the whole infrastructure developed as an infrastructure that will evolve alongside the project.

1.3 Relation to other Work Packages and Deliverables

The relation of the Marketplace as a central part of WP5 the respective work package has a central role and clear interaction with all the rest SMART4ALL WPs. Specifically, apart from itself and rest of the WP5 Tasks, it received input from WPs 3, 6, and 7 in the form of funded projects' information and artefacts provided through the Marketplace and offers output to WPs 2, 4 and 8 as a critical pillar facilitation further promotion, outreach and advertising, sustainable business modes and impact monitoring.

2 DESIGN APPROACH & TECHNOLOGIES INVOLVED

The SMART4ALL Marketplace platform is designed as dictated by the main principles of the microservice architectural paradigm. Microservices provide an architectural style that structures an application as a collection of well-defined services that are:

- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Agnostic to the implementation technology
- Organized around business capabilities

In that context, SMART4ALL enables the rapid, frequent and reliable delivery of a web platform that presents significant dynamics. In more details, the adopted approach benefits the platform from many aspects:

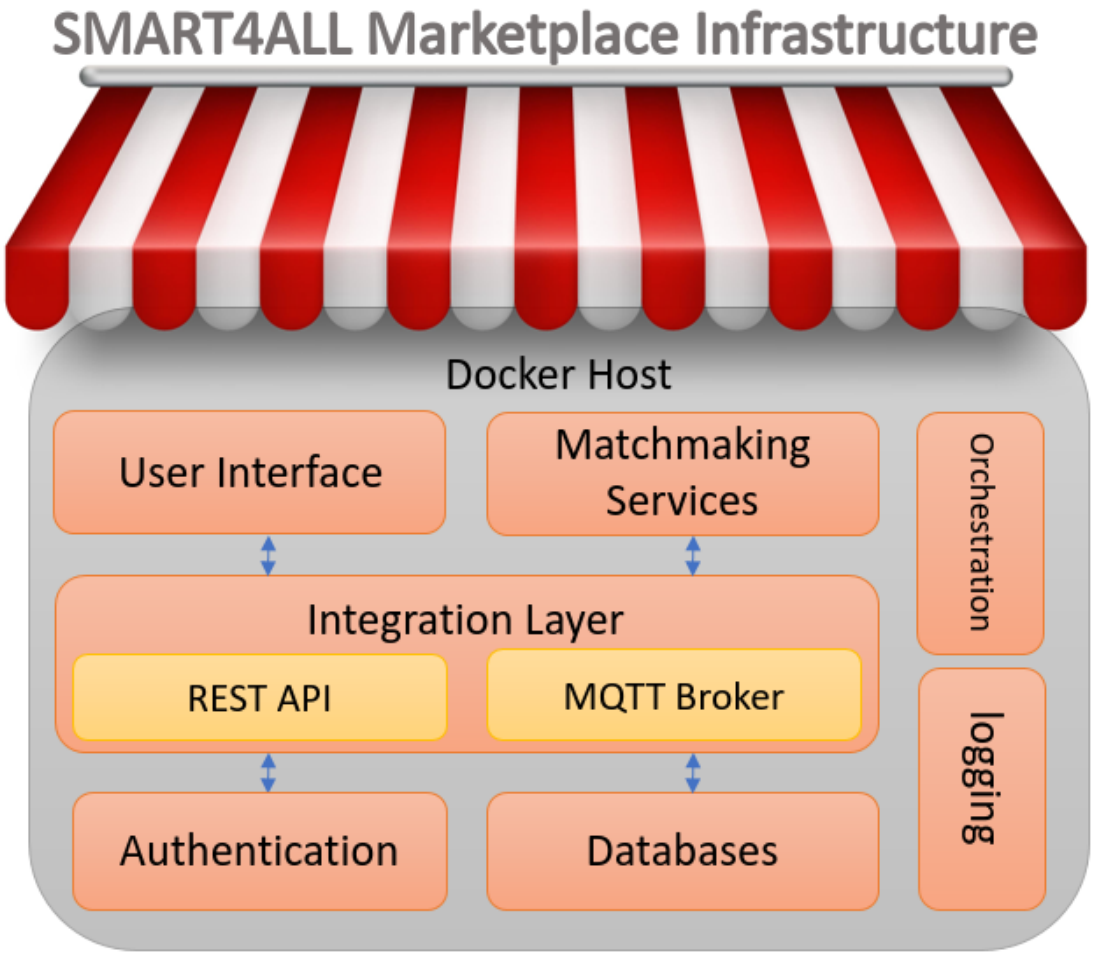

Figure 1: High Level Marketplace Infrastructure

- Enables the continuous delivery and deployment
- Improves maintainability - each service is relatively small and so is easier to understand and change
- Better testability - services are smaller and faster to test
- Better deployment strategies - services can be deployed independently

The small volume of each microservice (in terms of complexity and footprint) makes each microservice:

- Easier for a developer to understand
- Starts faster, which makes developers more productive, and speeds up deployments
- Improved fault isolation. For example, if there is a memory leak in one service then only that service will be affected. The other services will continue to handle requests. In comparison, one misbehaving component of a monolithic architecture can bring down the entire system.
- Eliminates any long-term commitment to a technology stack. When developing a new service, you can pick a new technology stack. Similarly, when making major changes to an existing service you can rewrite it using a new technology stack.

In that context, microservices are leveraged by the evolution of virtualization and particularly containerization technology. Docker [*1*] is the dominant containerization technology so far and has been adopted by SMART4ALL as the container supervisor to run all the Marketplace services. Each microservice is realized as a container service that all operate in the same docker network and communicate through well-defined interfaces (REST API, MQTT [*2*], database drivers, etc.). Docker pros can be summarized in the following:

- Continuous Integration (CI) Efficiency: Docker enables you to build a container image and use that same image across every step of the deployment process. A huge benefit of this is the ability to separate non-dependent steps and run them in parallel. The length of time it takes from build to production can be sped up notably.
- Compatibility and Maintainability: One of the benefits is parity. Parity, in terms of Docker, means that the images run the same no matter which server is running on.
- Simplicity and Faster Configurations: One of the key benefits of Docker is the way it simplifies matters. As Docker can be used in a wide variety of environments, the requirements of the infrastructure are no longer linked with the environment of the application.
- Rapid Deployment: Docker manages to reduce deployment to seconds. This is because it creates a container for every process and does not boot an OS. Data can be created and destroyed without worry that the cost to bring it up again would be higher than what is affordable.
- Continuous Deployment and Testing: Docker ensures consistent environments from development to production. Docker containers are configured to maintain all configurations and dependencies internally
- Multi-Cloud Platforms: One of Docker's greatest benefits is portability. Over the last few years, all major cloud computing providers, including Amazon Web Services (AWS)[*3*] and Google Compute Platform (GCP) [*4*], have embraced Docker's availability and added individual support. Docker containers can be run inside an Amazon EC2 instance, Google Compute Engine instance, Rackspace server, or VirtualBox, provided that the host OS supports Docker. If this is the case, a container running on an Amazon EC2 instance can easily be ported between environments, for example to VirtualBox, achieving similar consistency and functionality. Also, Docker works very well with other providers like Microsoft Azure [*5*], and OpenStack [*6*], and can be used with various configuration managers like Chef [*7*], Puppet [*8*], and Ansible [*9*], etc. While the Marketplace services run on the SMART4ALL infrastructure as described later, it was a design decision to make is easily ported to other cloud providers in favor of the sustainability of the platform.

- Isolation: Docker ensures your applications and resources are isolated and segregated. Docker makes sure each container has its own resources that are isolated from other containers. Docker helps you ensure that a clean application removal can be achieved since each application runs on its own container.
- Resource management: On top of these benefits, Docker also ensures that each application only uses resources that have been assigned to them. A particular application won't use all of your available resources, which would normally lead to performance degradation or complete downtime for other services.
- Security: The last of these benefits of using docker is security. From a security point of view, Docker ensures that applications that are running on containers are completely segregated and isolated from each other, granting you complete control over traffic flow and management. No Docker container can look into processes running inside another container. From an architectural point of view, each container gets its own set of resources ranging from processing to network stacks.

To further facilitate application deployment, scaling, and management, Kubernetes [*10*] has been chosen as the container orchestration solution. Kubernetes aims to provide a platform for automating deployment, scaling, and operations of application containers across clusters of hosts. It works with a range of container tools with Docker being one of them. Kubernetes is commonly used as a way to host a microservice-based implementation because it and its associated ecosystem of tools provide all the capabilities needed to address key concerns of the microservice architecture as already described. To this end, the platform offers several API primitives, deployment options, networking, container and storage interfaces, built-in security, and other useful features. The main advantages of Kubernetes are the following:

- Broad adoption in the cloud-native community
- A great ecosystem of supporting tools and interfaces
- Deep integration into the cloud-native ecosystem
- Broad support for containers runtimes
- Multiple workloads and deployment options
- Great support for stateful apps
- Flat networking model
- Efficient application updates
- Efficient resource management
- Built-in security
- Extensibility and pluggability
- Integration with major cloud providers

Towards the efficiency of the development and quality of the released software that constitutes the SMART4ALL Marketplace, modern Continuous Integration/Continuous Delivery (CI/CD) solutions have been also incorporated in the workflow. CI is a development methodology that involves frequent integration of code into a shared repository. The integration may occur several times a day, verified by automated test cases and a build sequence (not mandatory). Here are a few benefits that have made continuous integration essential to any application development lifecycle.

- Early Bug Detection: If there is an error in the local version of the code that has not been checked previously, a build failure occurs at an early stage. Before proceeding further, the developer will be required to fix the error. This also benefits the QA team since they will mostly work on builds that are stable and error-free.
- Reduces Bug Count: In any application development lifecycle, bugs are likely to occur. However, with Continuous Integration and Continuous Delivery is used, the number of bugs is

reduced a lot. Although it depends on the effectiveness of the automated testing scripts. Overall, the risk is reduced a lot since bugs are now easier to detect and fix early.

- Automating the Process: The Manual effort is reduced a lot since CI automates build, sanity, and a few other tests. This makes sure that the path is clear for a successful continuous delivery process.
- The Process Becomes Transparent: A great level of transparency is brought in the overall quality analysis and development process. Development gets a clear idea when a test fails, what is causing the failure, and whether there are any significant defects.
- Cost-Effective Process: Since the bug count is low, manual testing time is greatly reduced and the clarity increases on the overall system, it optimizes the budget of the project.

CD is the process of getting all kinds of changes to production. Changes may include configuration changes, new features, error fixes, etc. They are delivered to the user in a safe, quick, and sustainable manner.

The goal of the CD is to make deployment predictable and scheduled in a routine manger. It is achieved by ensuring that the code always remains in a state where it can be deployed whenever demanded, even when an entire team of developers is constantly making changes to it. Unlike CI, testing and integrating phases are eliminated and the traditional process of code freeze is followed.

If the best practices are followed, continuous delivery can help application development in quite a few ways.

- Reducing the Risk: The main goal of the CD is to make deployment easier and faster. Patterns like blue-green deployment make it possible to deploy the code at very low risk and almost no downtime, making deployment undetectable to the users.
- High-Quality Application: Most of the process is automated. Testing activities can now be continuously performed during the delivery process, ensuring a higher quality application.
- Reduced Cost: When an investment is made on testing, build, and deployment, the product evolves quite a lot throughout its lifetime. The cost of frequent bug fixes and enhancements are reduced since certain fixed costs that are associated with the release is eliminated because of continuous delivery.
- Better Product Releases: Since the aim of the CD is to make a product release painless, the team can work in a relaxing manner. Because of frequent release, the team works closely with users and learn what ideas work and what new can be implemented to delight the users. Continuous user feedback and new testing methodologies also increase the product's quality.

Towards this direction, SMART4ALL Marketplace development is supported by popular and open source automation tools such as Jenkins. Source code management is performed by the git versioning system and facilitated by the Gitlab platform that operates on the SMART4ALL infrastructure.

As already presented, containerization (through the Docker engine) allows the development of services in various technologies without jeopardizing their interaction and cooperation. In that context, the SMART4LL development team used their expertise on various domains and technologies to make the optimum selections for each service that is part of the SMART4ALL Marketplace ecosystem. The following table (Table 1) summarizes all the technologies that have been selected to serve particular objectives.

Table 1: Adopted technologies for the design, development, and operation of the SMART4ALL Marketplace

Objective	Technology
Service deployment	Docker

Service orchestration	Kubernetes
CI/CD	Jenkins [*11*]
Source code version control	git (Gitlab [*12*] platform)
Other automation	Python/Bash scripting
Frontend	ReactJS [*13*] framework (Javascript/JSX)
Backend/API	Django [*14*] (Python), Spring [*15*] (Java)
Match Making	Python

3 FRONDED GUI INTERFACE

The graphical user interface for the SMART4ALL Marketplace is based on popular web technologies and the design was performed to match the requirement for responsive layouts that are fully functional and friendly in every screen of modern electronic devices (from smartphones to large desktop screens).

The implementation technology selected for the UI was the ReactJS framework. ReactJS is an open-source JavaScript library that is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. The main advantages of ReactJS are listed below:

- It facilitates the overall process of writing components. JSX is an optional syntax extension to JavaScript that makes writing your own components much easier. It accepts HTML quoting and makes a subcomponent rendering easier.
- It boosts productivity and facilitates further maintenance through the ability to reuse system components
- It ensures faster rendering through the Virtual DOM - currently, one of the benefits of using React for heavy loaded and dynamic software solutions.
- It guarantees a stable code. To make sure that even small changes that take place in the child structures won't affect their parents, ReactJS uses only downward data flow.
- It is SEO friendly. Another ReactJS benefit is its ability to deal with a common search engine failure to read JavaScript-heavy apps.
- It comes with a helpful developer toolset - React Developer Tools
- It is backed by a strong community

The Layout of the SMART4ALL Marketplace consists of three main areas as presented in Figure 2:

- Header: Logo, navigation and user menu live here
- Footer: Horizon 2020 disclaimer, EU flag, terms of use, privacy policy, etc. are placed here
- Content: This the main area center of the Marketplace where the user will have access to the artefacts repository and matchmaking services.

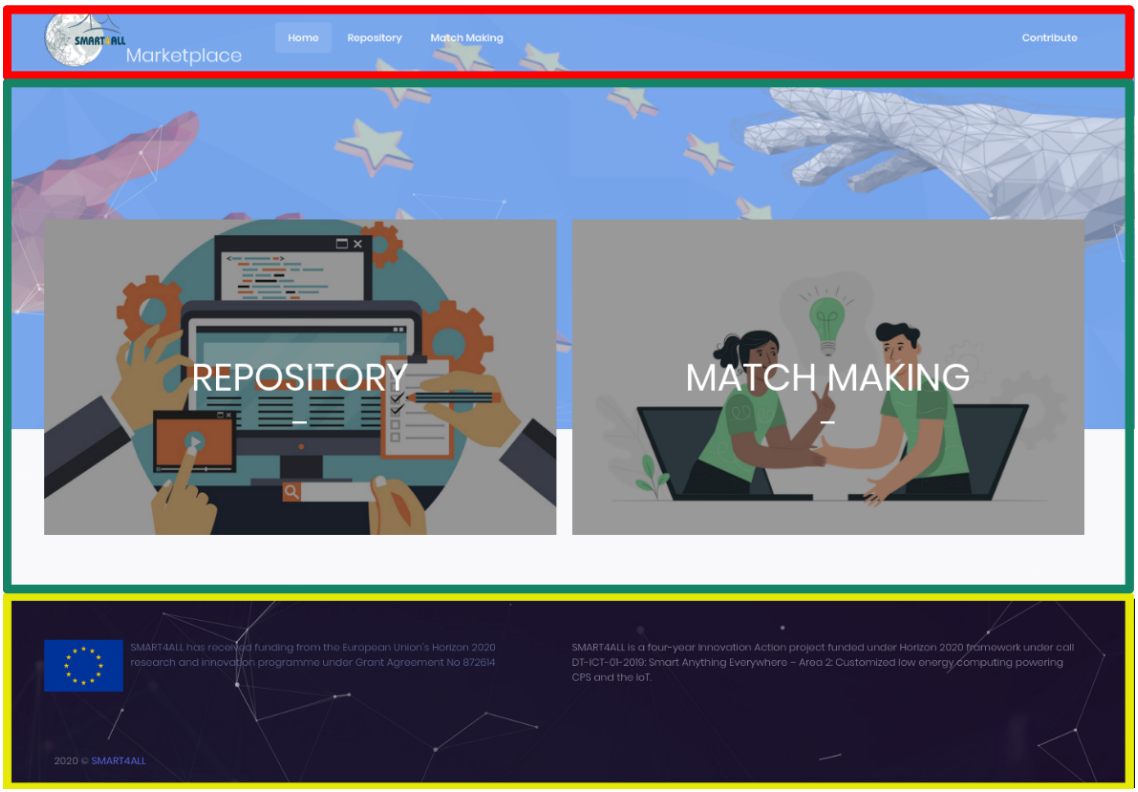

Figure 2: Marketplace Layout

The Web User Interface of the SMART4ALL Marketplace gets populated by a well-defined and secure REST API. The main content of the Marketplace is presented in 4 pages, particularly:

1. Landing page: The first page that welcomes the user to the Marketplace and gives him the options to navigate to the content of the Marketplace
2. Repository (Figure 3): Repository is the place where all the Marketplace artefacts are presented. The user is able to browse the artefacts and filter the artefact list based on its preferences. These preferences are denoted through a filters bar where the user can filter artefacts by:
    a. Domain (agriculture, transportation, anything, etc.)
    b. Category (tool, service, educational material)
    c. Type (document, VM, container, etc.)
    d. License (e.g. open-source, proprietary)
    e. Search term (free text field)

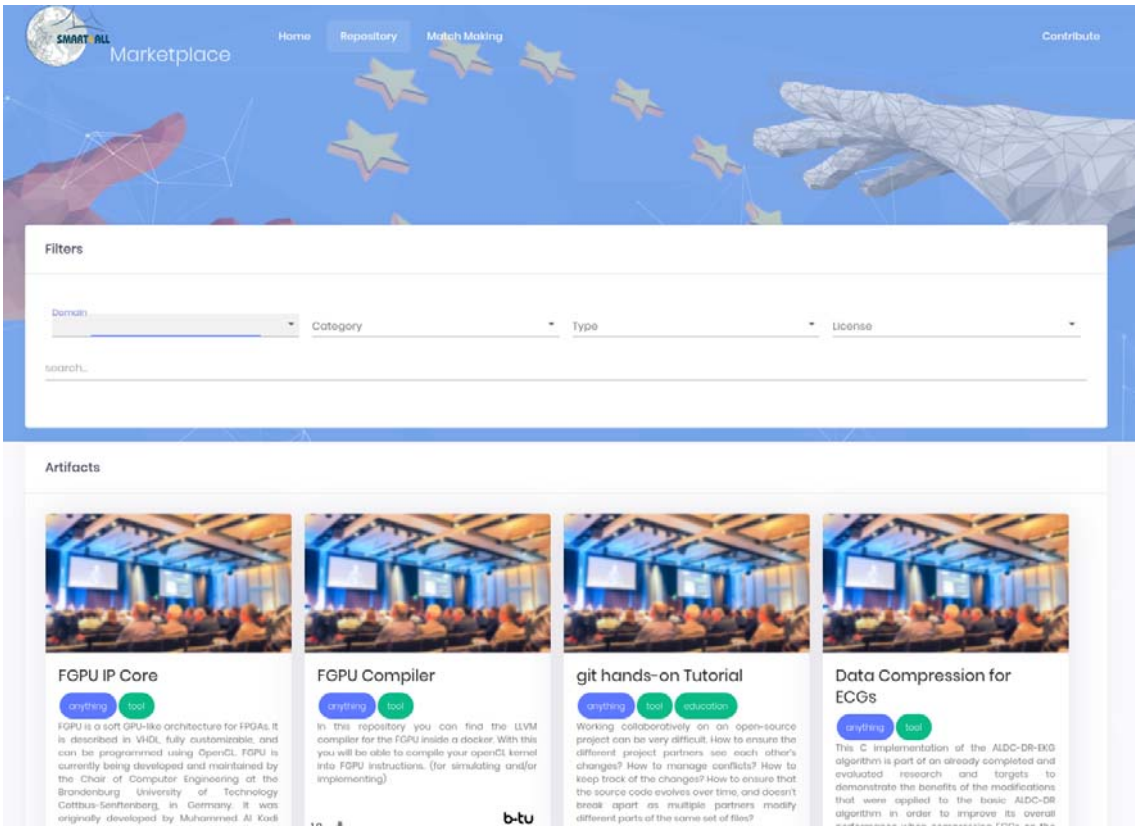

Figure 3: Artefacts repository

3. MatchMaking (Figure 4): The matchmaking page provides the user all the available controls in order to enter his preferences and get the results by the matchmaking tool. These results could be artefacts or other organizations that could settle cooperation.

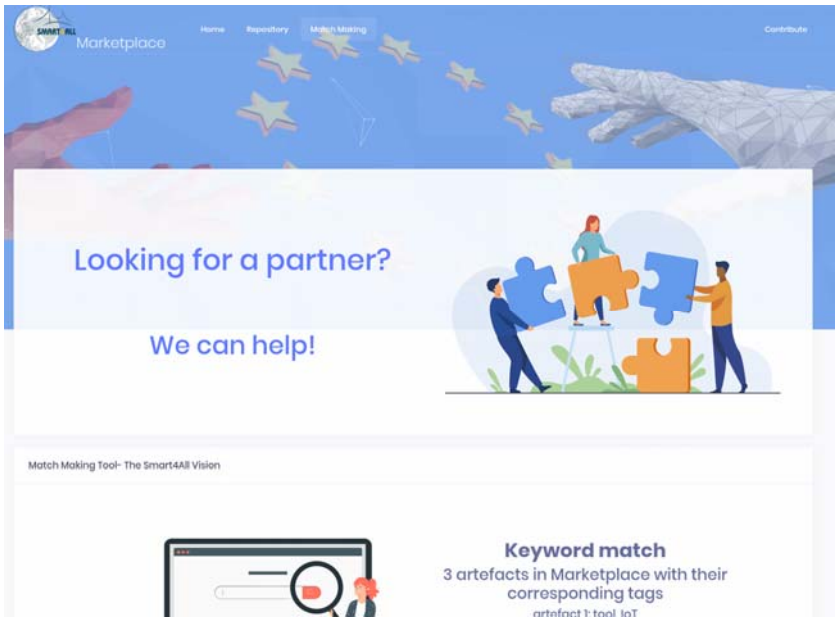

Figure 4: Match Making

4. Contribute (Figure 5). The contribute page is the form where an organization could register and offer one or more artefacts to the SMART4ALL ecosystem.

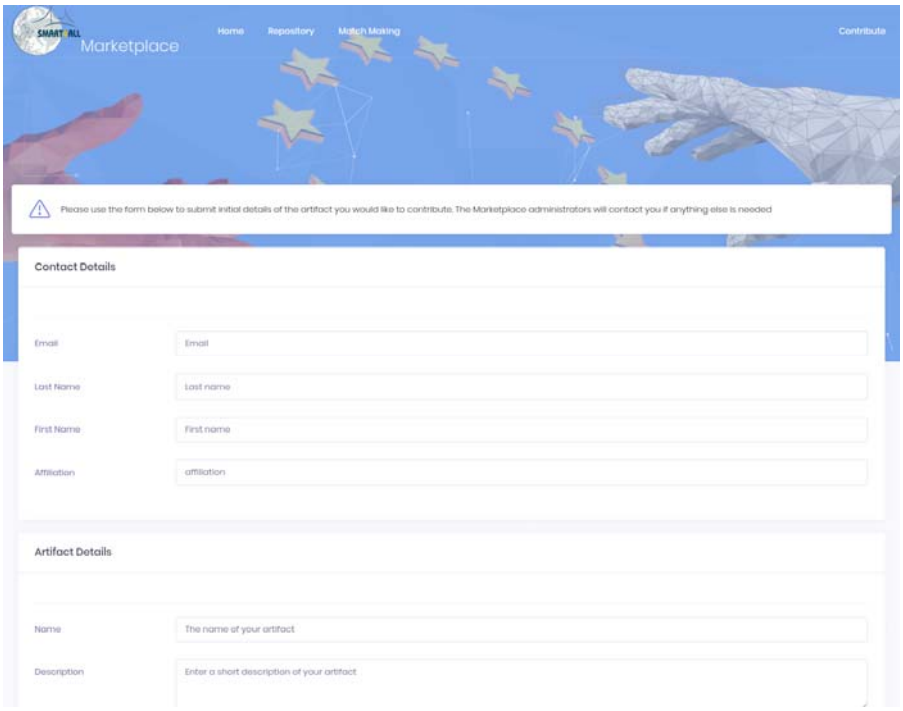

Figure 5: Artefacts submission form

4 BACKEND INFRASTRUCTURE

Aiming to support the functions described in the previous sections, a cloud-based back-end infrastructure has been established. This infrastructure, as Figure 6 depicts, comprises a compute engine and a non-relational database. In order to support the communication channels with the end-users, several API endpoints are exposed, as detailed in the following paragraphs.

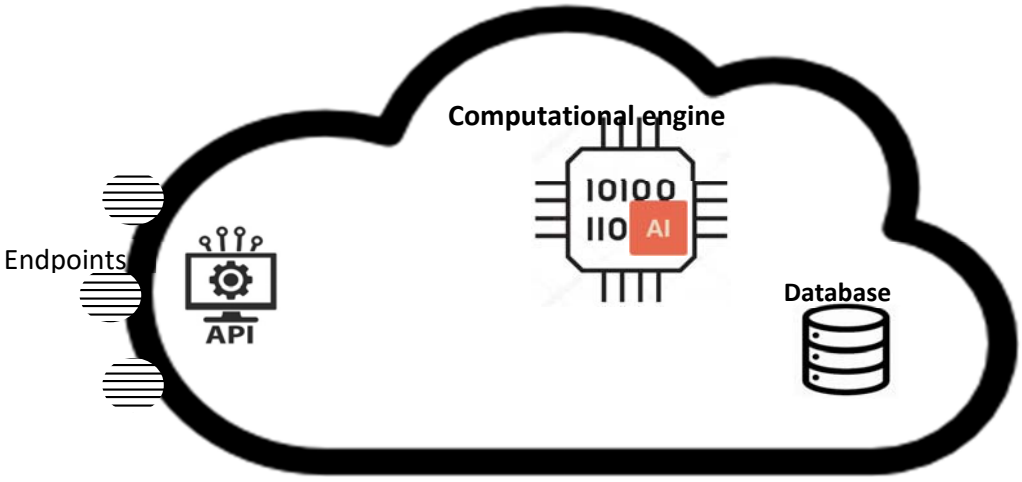

Figure 6: Main components of the cloud-based Matchmaking infrastructure

The main components of the back-end infrastructure:

1. Computational engine: The intelligence part of the Marketplace service is hosted in the computational engine. The trained models for the matchmaking service and the recommendation reporting functionalities are implemented in the engine.

2. Database: A non-relational database (Mongo-DB) has been selected to store the information related to the services. Additionally, the artefacts hosted by the service are also stored in the same database. Aiming to support potential scaling of the system, relative distributing technologies like Spark have been considered. If required, both the services and the data can be shipped to a distributed environment, supporting big-data analytics.

3. **API Endpoints:** Asynchronous communication is the main mean for exchange information between the end-users and the cloud-based service. Thus, a GET endpoint allows the users to query the service and retrieve the desired information, while a POST endpoint enables the upload of new artefacts to the relative database.

As far as the latest service is concerned (artefact upload API endpoint), a structural procedure has been formulated, in order to reassure that the uploaded artefacts fulfill all the relative requirements. More specifically, a session, for uploading a new artefact will require to formulate a JSON file, which describes the data-model. For this, a wizard-like procedure will be provided to the final user, which will guide her / him to provide all the necessary information. The produced JSON file will be used as input to a JSON validator. JSON validator is a data structure validation library based around JSON schema, and can be used directly to verify the correct structure of the data model file.

Aiming to provide more technical information related to the aforementioned modules, Figure 7 describes the main users of the database along with the related functions.

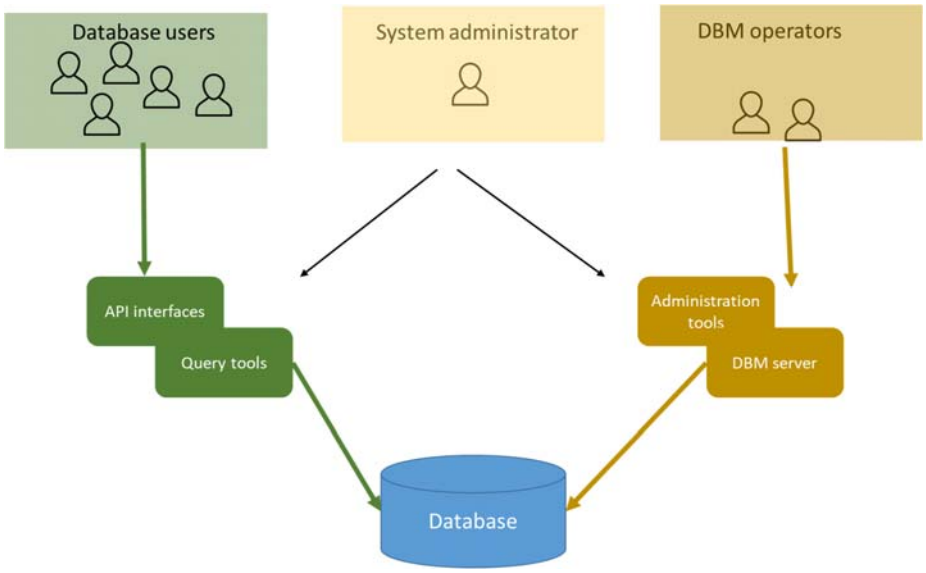

Figure 7: Database main users

- Database users refer to the end-users of the platform. The communication between the end-users and the platform is held using API Rest calls, which invoke certain query calls to the database.
- Database Management Operators (DBM operators) refer to the users with the permission to add/update/delete database users from the DBM and of course, add/update/delete artefacts
- The system administrator can edit all entities of the database.

The authentication of the end-users will be conducted using encrypted one-way authentication, utilizing the sha256 algorithm. All stored passwords will be encrypted, minimizing the possibility of data compromise.

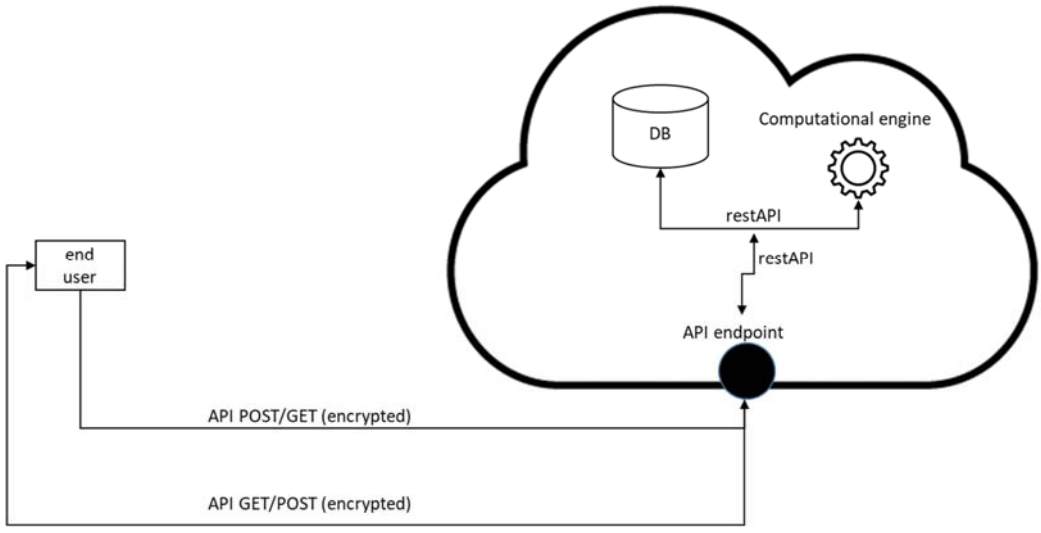

Figure 8: Technical specifications of the backend infrastructure.

Figure 8 details the interfaces among the basic entities of the backend infrastructure. It is important to mention at this point that the main communication pathways will be built upon restful APIs.

5 MATCHMAKING SERVICES AND FUNCTIONALITIES

5.1 Default Matchmaking Service

An online platform has been developed in order to enable the organizations' access to the SMART4ALL ecosystem. Its purpose is twofold: To assist the matchmaking process among technological parties based on specific requirements, offerings, and characteristics and to facilitate partner search for SMART4ALL Open Calls. The online "Matchmaking & Partner Search" service consists of two distinct subsystems. The first subsystem is publicly available to organizations at https://matchmaking.smart4all-project.eu, and the second one is privately available upon authentication at https://matchmaking.smart4all-project.eu/scp to SMART4ALL project partners who are responsible for the co-ordination of the Matchmaking and Partner Search process.

5.1.1 Organizations' Subsystem of "Matchmaking & Partner Search" platform

5.1.1.1 Initial Page

Through the initial page, an organization can either start a "New Search", by submitting a form containing all the required information or "Check Search Status" in order to gain access to a previously submitted form.

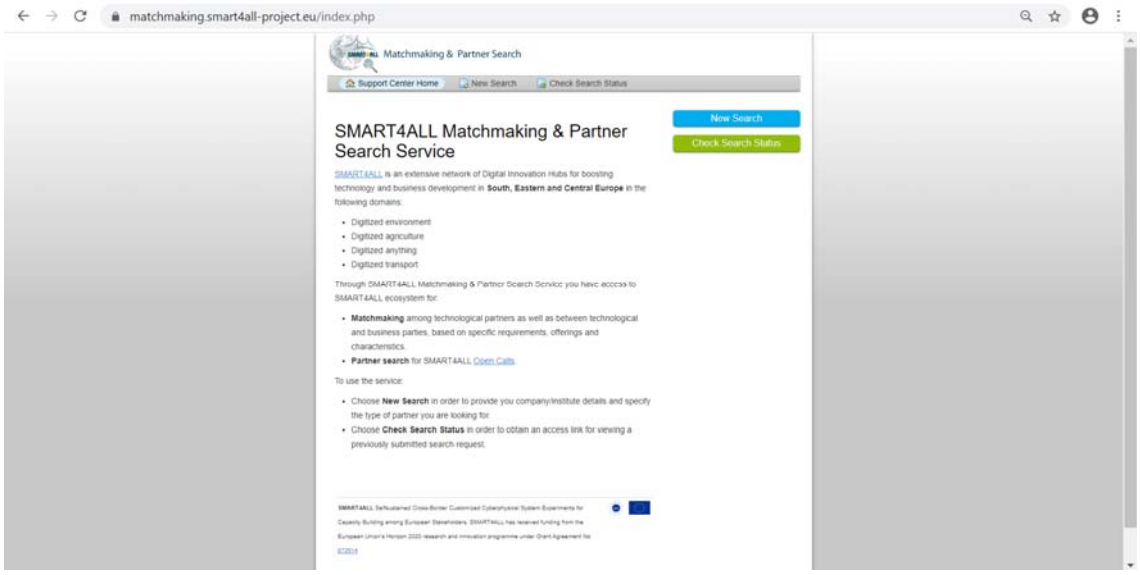

Figure 9: SMART4ALL MatchMaking Service Initial Page

5.1.1.2 "New Search" Page

In order to initiate a "New Search," an organization must fill in two distinct sections in the corresponding "SMART4LL Matchmaking Info" form. The first section requires information concerning the organization itself and the second one is related to information concerning the desired partner.

As far as the organization's information is concerned, the following must be entered:

- Name
- Expertise

- Country
  (single selection from a predefined list of eligible countries)
- Type
  (single selection from a predefined list: University, Research Institute, SME, Slightly Bigger Company)
- Contact Person Name
- Contact Person E-mail
- Interest in joining the SMART4ALL DIH Network
  (single selection: Yes/No)
- Interest in subscribing to SMART4ALL Newsletter
  (single selection: Yes/No)
- Proposal Targets
  (multiple selection from a predefined list: Digitized Environment, Digitized Agriculture, Digitized Transport, Digitized Anything)

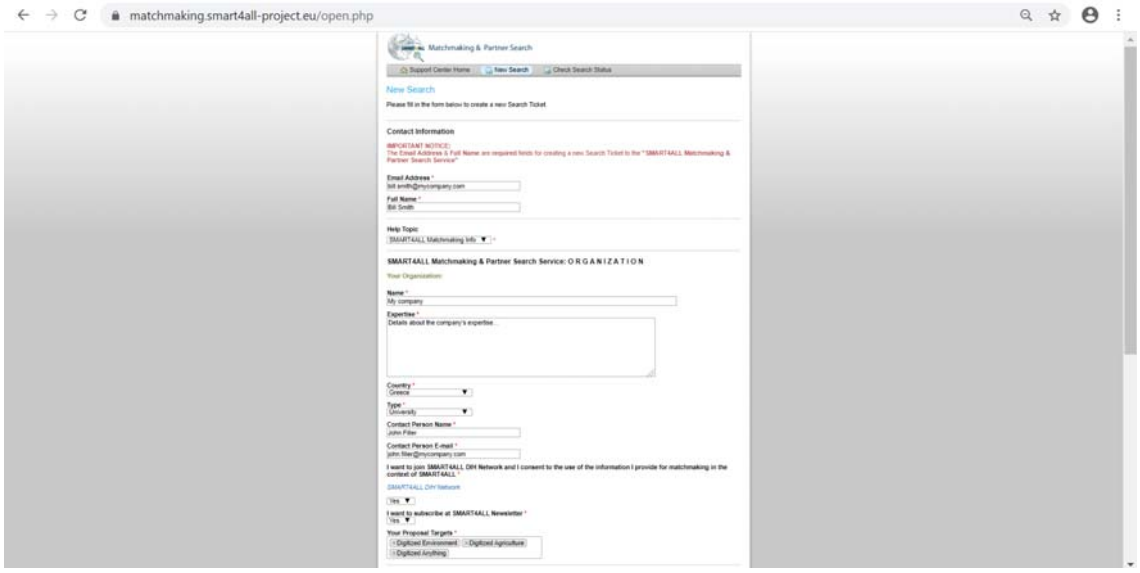

Figure 10: SMART4ALL MatchMaking Service New Search Page (1st)

As far as the desired partner's information is concerned, the following must be entered:

- Expertise
- Country
  (multiple selections from a predefined list of eligible countries)
- Geographical Area
  (multiple selections from a predefined list: Central Europe, Eastern Europe, South Europe, The Balkans)
- Type
  (multiple selections from a predefined list: University, Research Institute, SME, Mid Caps)
- Role

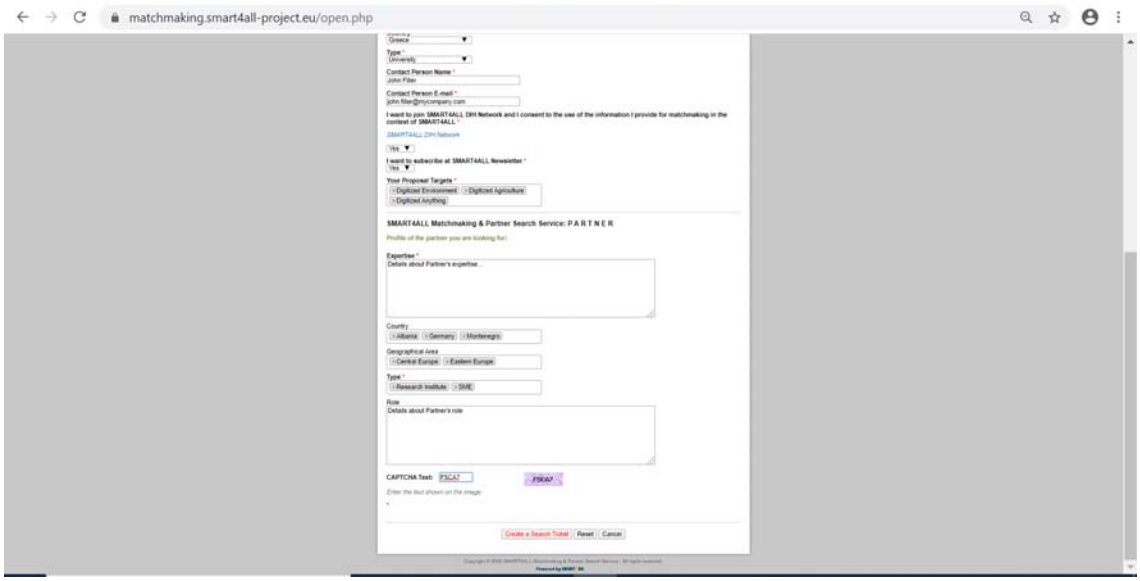

Figure 11: SMART4ALL MatchMaking Service New Search Page (2nd)

Upon successful submission of a "New Search" form, the organization receives an automated e-mail response for confirmation purposes from the "Matchmaking & Partner Search service" online platform. The confirmation e-mail also contains useful administrative information (the access link and a unique "new search" identification number), which can be used for future access to information related to that specific search.

5.1.1.3 Check Search Status

An organization may have access to the submitted information of its previous own search in two ways:

- Either employing the access link that was sent by the platform upon successful submission of a new search or
- By providing, through the "Check Search Status" page, the unique identification number of a previous search along with the organization's e-mail address. This can be useful in case the initial confirmation e-mail (which contains the access link) has been lost.

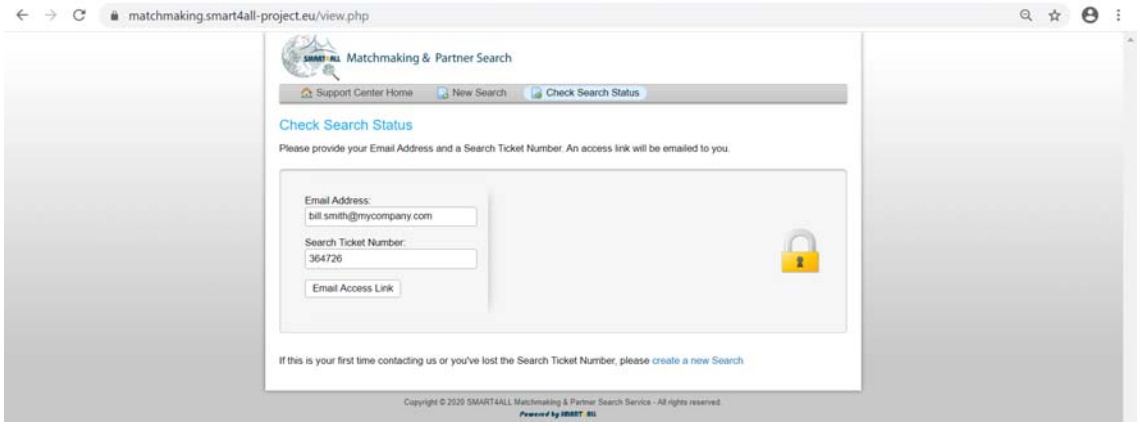

Figure 12: SMART4ALL MatchMaking Service Check Search Status Page

In any case, the organization obtains access to all the necessary information regarding its search ticket.

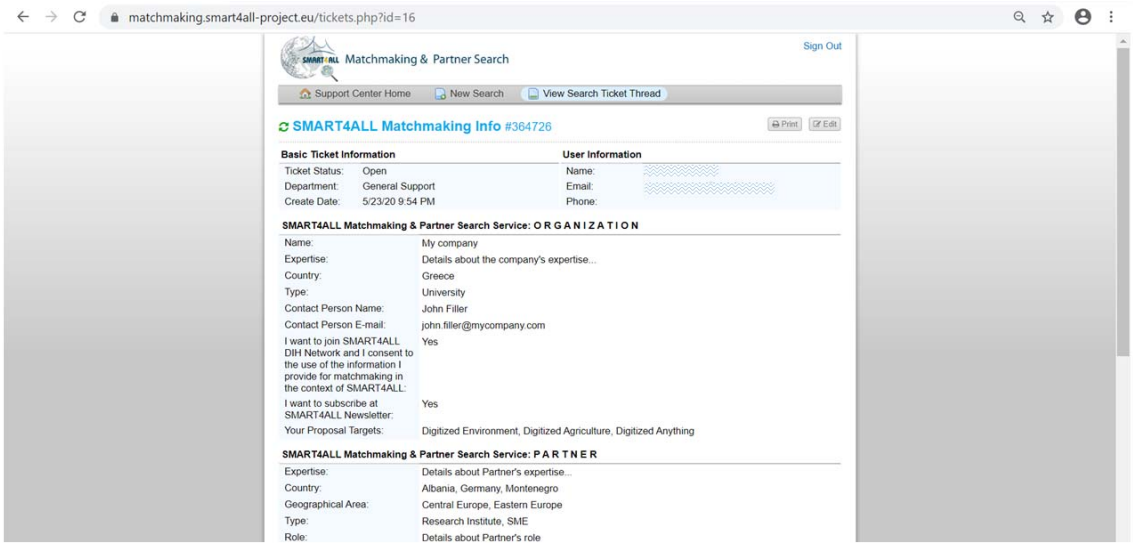

Figure 13: SMART4ALL MatchMaking Service Search Status Information Page

5.1.2 SMART4ALL project partners' Subsystem of "Matchmaking & Partner Search" platform

5.1.2.1 Initial Page

The SMART4ALL project partners have access to all the submitted organization forms for further processing through a private area, that is accessible via authentication.

Figure 14: SMART4ALL MatchMaking Service Private Area Login Page

5.1.2.2 New Search Ticket Management

The SMART4ALL project partners, who are responsible for assisting in the "Matchmaking and Partner Search" process, are divided into distinct support teams based on geographical and application criteria. For each submitted organization form, new useful metadata are added (APPLICATION DOMAIN GROUPING & GEOGRAPHICAL REGION GROUPING) by a first-level support team. After an initial evaluation at this first level, each Search Ticket Request is further assigned to the appropriate, more focused second level (GEO/APP) teams for further processing.

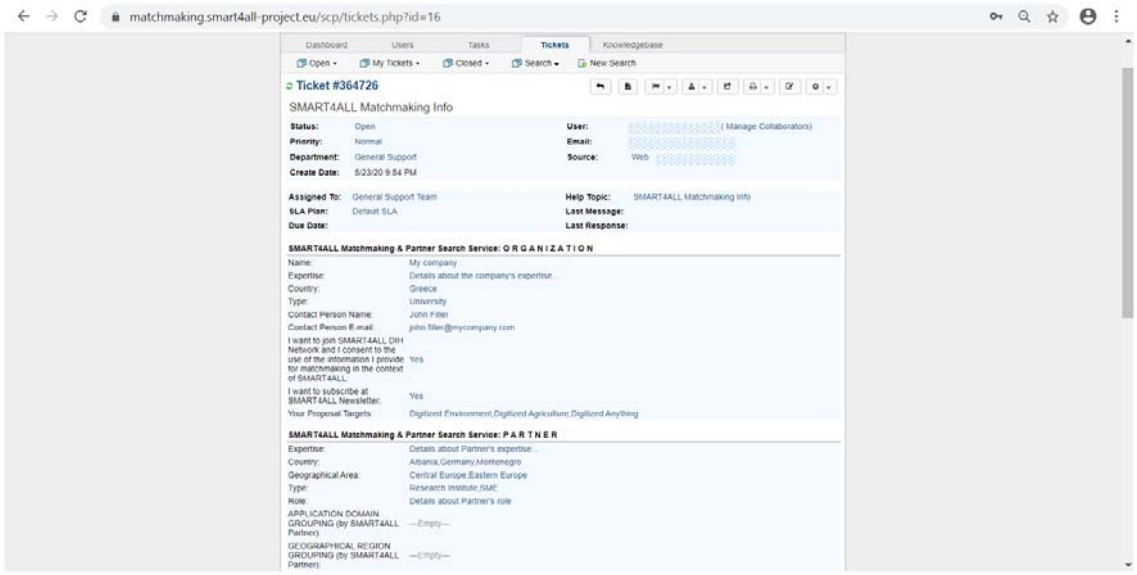

Figure 15: SMART4ALL MatchMaking Service Private Area New Search Ticket Management Page

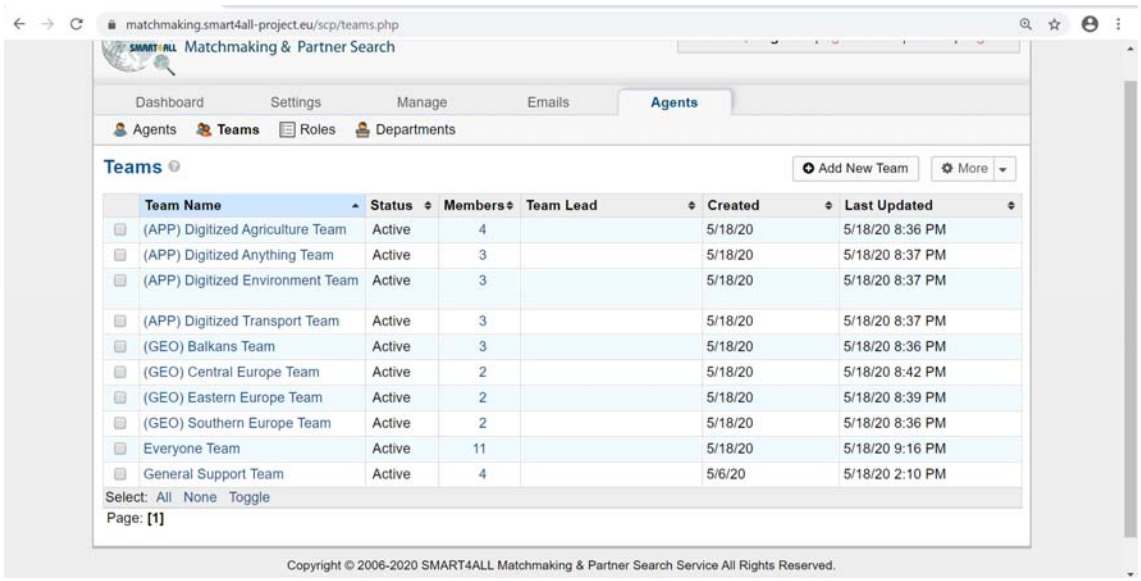

Figure 16: SMART4ALL MatchMaking Service Private Agents Management Page

5.2 Smart Matchmaking Service

5.2.1 NLP techniques- Tokenization, Stemming, Lemmatization

While matching only the exact words that the user has queried would be precise, it is not enough. We would miss out on many documents that the user would consider to be relevant. Instead, we need to also search for words that are not exactly the same as the original but are related. In order to improve the matching process we have incorporated natural language processing techniques (NLP) [*16*] and specifically tokenization and stemming [*17*]. When we deal with text, we need to break it down into smaller pieces for analysis. This process is termed tokenization and involves dividing the input text into a set of pieces called tokens. Stemming is another important text analysis process, which involves reducing a word to its word stem. Stem is affixed to suffixes and prefixes, or to the roots of words, known as a lemma. Stemming is important in natural language understanding and natural language processing. Stemming is a part of linguistic studies in morphology and artificial intelligence (AI) information retrieval and extraction. Stemming and AI techniques extract meaningful information from vast sources like big data or the Internet since additional forms of a word related to a subject may need to be searched to get the best results. Stemming is also a part of queries and Internet search engines. Recognizing, searching, and retrieving more forms of words returns more results. When a form of a word is recognized it can make it possible to return search results that otherwise might have been missed. That additional information retrieved is why stemming is integral to search queries and information retrieval.

Often, the best results can be attained by using the basic morphological form of the word: the lemma. The lemmatization process uses a vocabulary and morphological analysis of words. It obtains the base forms by removing the inflectional word endings.

Our Marketplace search engine utilizes all the above NLP techniques to gain a better understanding of a user's query and serve the most relevant result.

5.2.2 Semantics / Ontologies

Text similarity has to determine how 'close' two pieces of text are both in surface closeness (lexical similarity) and meaning (semantic similarity). Instead of doing a word for word comparison, we also need to pay attention to the context in order to capture more of the semantics. To consider semantic similarity we need to focus on phrase level (or lexical chain level) where a piece of text is broken into a relevant group of related words prior to computing similarity [*18*]. We know that while the words may significantly overlap, two phrases may actually have a different meaning. The big idea is that we represent search items as vectors of features, and compare them by measuring the distance between these features. There are multiple ways to compute features that capture the semantics of search phrases. Supervised training can help sentence embeddings learn the meaning of a sentence more directly.
The data exchange between searches and the Marketplace in a Semantic scenario is based on a set of vocabularies that provide shared terms to describe artefacts tags, descriptions, and industrial sectors. Semantic matching is a technique that combines annotations using controlled vocabularies with background knowledge about a certain application domain [*19*]. In our case, the domain-specific knowledge is represented by a taxonomy of artefacts' tags. Taxonomy is defined as a set of categories or terms organized into a hierarchy with parent-child relationships and implied inheritance, meaning that a child term (e.g Sensors) has all of the characteristics of its parent term (e.g. IoT). A taxonomy only contains broader and narrower relationships. A part of the implemented taxonomy is shown in Figure 17.

Figure 17: Part of the taxonomy implemented.

The implemented taxonomy serves a very important role: It matches the text of artefacts' tags and descriptions and identifies terms corresponding to search terms. It is important to clarify that we do not use a simple keyword search but a concept search. In most cases, a query does not ask for specialized technologies but resorts to more general categories, such as IoT hardware (as opposed to sensors or controllers). In this case, the proposed algorithm searches the hierarchy tree and identifies the leaves with the node of the technology required as their lowest (nearest) common ancestor (for instance, object-oriented programming). Next, the identified leaves are examined to determine if there is a match with the artefact tags. In the case that there is no match then the artefact is excluded from the ranking process.

5.2.3 Collaborative filtering

Like many machine learning techniques, a recommender system makes prediction based on users' past behaviors, to predict user preferences based on past experience. To build a recommender system, the most two popular approaches are Content-based and Collaborative Filtering. A content-based approach requires a good amount of information on items' own features, rather than using users' interactions and feedback. Collaborative Filtering, on the other hand, only relies on users' historical preference on a set of items. Because it's based on historical data, the core assumption here is that the users who made similar choices in the past will do so also in the future [*20*].

The standard method of Collaborative Filtering is known as the Nearest Neighborhood algorithm. There are user-based CF and item-based CF. Let's first look at User-based CF. We have an n × m matrix of ratings, with user u_i, i = 1, ...n and item p_j, j=1, …m. Now we want to predict the rating r_{ij} if target user

i did not watch/rate an item j. The process is to calculate the similarities between target user i and all other users, select the top X similar users, and take the weighted average of ratings from these X users with similarities as weights. Basically, the idea is to find the most similar users to your target user (nearest neighbors) and weight their ratings of an item as the prediction of the rating of this item for the target user. Collaborative Filtering provides strong predictive power for recommender systems, and requires the least information at the same time. However, it has a few limitations in some particular circumstances. Collaborative Filtering is faced with a cold start. When a new item coming in, until it has to be rated by a substantial number of users, the model is not able to make any personalized recommendations. Similarly, for items from the tail that didn't get too much data, the model tends to give less weight on them and have popularity bias by recommending more popular items.

6 MARKETPLACE ARTEFACTS DATA MODEL

A data model is a conceptual representation of data objects and the associations between different data objects. Data modeling helps in the visual representation of data and ensures consistency in naming conventions, default values, semantics, security while ensuring the quality of the data. Data modeling emphasizes on what data is needed and how it should be organized instead of what operations need to be performed on the data [*21*].

Data models represent information areas of interest. Two modeling methodologies stand out, top-down and bottom-up [*22*]:

- Bottom-up models usually start with existing data structures forms, fields on application screens, or reports. These models are usually physical, application-specific, and incomplete from an enterprise perspective.
- Top-down data models are created in an abstract way by getting information from people who know the subject area. A system may not implement all the entities in a logical model, but the model serves as a reference point or template.

Data models, following the bottom-up approach, have been used extensively in online Marketplaces to build product catalogs, organizing products and services in hierarchies. This helps sellers list their offerings into the appropriate category, which simplifies product search. For example, the following image (Figure 18) shows Skroutz's (the leading Comparison-Shopping Engine in Greece) product listing GUI, which is translated into an internal product model.

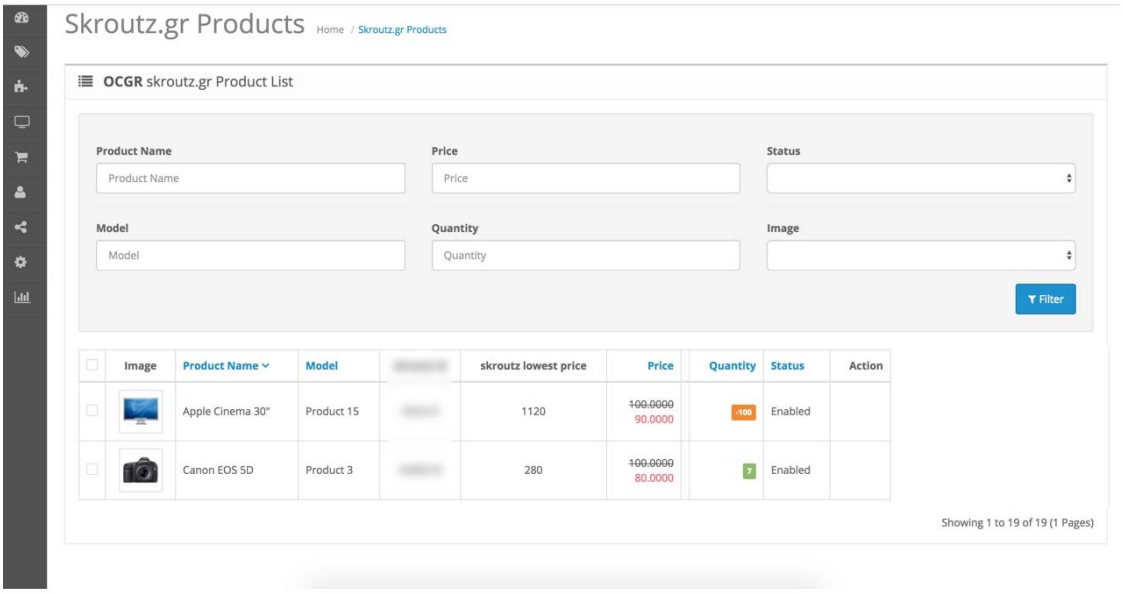

Figure 18: Skroutz's product listing

In Smart4All we followed the second approach, which is more appropriate for the much wider scope of products and services supported. We have implemented Smart4All's data model in json format in order to model and store all information needed for the Marketplace artefacts. First, we had to model all the information related to the artefact such as thematic area, description, version details, etc. As a second step we modeled the relationships between the artefacts via the "tags" field. Using the information included in the tag we will build a taxonomy with the artefacts' relationships, which will be used by the match-making tool of the Marketplace to implement smart matchings. Finally, we modeled the information of partners that have used the artefact and their corresponding thematic area.

This information will be used for building a recommendation system in the future, with collaborative filtering algorithms.

The data model includes the following main data categories. The fields and some indicative values are the following:

- Thematic_areas: environment (true/false), agriculture(true/false), transport(true/false), anything (true/false)
- Artefact_details:artefact_name, description, version,…, isTool, isService, isEducation
- Tool/service details: description, type(design|simulations| cloud computing| software| hardware|...), Mean (software|hardware|dataset|...), tags
- Education_details: description, type(Assessment Tool|Case Study|Software platforms|Drill and Practice|Online Course|Social Networking Tool|Instructional Video|Workshop and Training Material), mean (online|offline) , tags
- Used by: partner_name, partner_area: (environment|agriculture|transport|anything|...)

In a graphical form the initial basic data model is shown in the picture below which will be refined and extended as the project progresses and specific needs are identified. (Figure 19).

Figure 19: SMART4ALL data model

The data model in json format is shown below:

```
{
  "id": 1,
  "name": "Name of the artefact",
  "description": "Description of the artefact",
  "thematic_areas": [ //environment, agriculture, transportation, anything
    {
     "id": 1,
     "alias": "transportation",
     "label": "Smart Transportation"
```

```
    }
  ],
  "class": "artefact",
  "version": "0.0.1",
  "releaseDate": "03/03/2020",
  "license": "GPL3",
  "contributor": "esdaLab",
  "contributorUrl": "https://www.esda-lab.gr",
  "url_hosted": "https://markeplace.esda-lab.gr/artefact_url",
  "url_image_small": "/url_image_small.png",
  "url_image_large": "/url_image_large.png",
  "category": { //tool, service, education
    "id": 1,
    "alias": "tool",
    "label": "Tool",
    "description": "The artefact's category description"
  },
  "type": [ //design,simulations,cloud computing, software, hardware, Assessment Tool|Case Study|Software platforms|Drill and Practice|Online Course|Social Networking Tool|Instructional Video|Workshop and Training Material
    {
      "id": 1,
      "alias": "design",
      "label": "Design"
    }
  ],
  "means": [ //online, offline, software, manual
    {
      "id": 1,
      "alias": "on",
      "label": "online"
    }
  ],
  "tags": [
    {
      "id": 1,
      "alias": "tag1",
      "label": "Tag1"
    }
  ],
  "used_by": [
    {
      "orgId": "string"
    }
  ]
}
```

7 PROJECT MANAGEMENT SERVICES & TOOLS

As part of the software infrastructure of SMART4ALL, tools and services will be offered to the consortia of experiments funded through SMART4ALL open calls, that will facilitate the project management. Respective tools and services will be based on Nextcloud software suite comprising a quite popular open source solution characterized by its versatility, user friendly and intuitive interface, and the extended range of 3rd party components/tools that someone can easily integrate and use according to the specific needs of the project or the collaborating team.

SMART4ALL project will offer to each funded experiment a fully functional such infrastructure integrating some of the most useful project management and collaboration tools, although each consortium will have total freedom in configuring and adjusting it to their needs. Additionally, SMART4ALL will provide maintenance of these infrastructure and technical support by experts.

Finally, for most them, short tutorial videos are presented by the members of SMART4ALL consortium as to their intended use in the context of SMART4ALL which may be useful for funded experiments consortia. Links to these videos are indicated for specific tool/service in the following presentations.

An indicative example of tools/services that will be by default offered include the following but any next cloud related application can be activated and provisioned according to the needs of the experiments or project:

File sharing:

Nextcloud application functionally is similar to Dropbox but unlike Dropbox, Nextcloud does not offer off-premises file storage hosting. Nextcloud is free and open-source (GNU AGPLv3), which means that anyone is allowed to install and operate it on their own private server devices. In contrast to proprietary services like Dropbox, Office 365, or Google Drive, the open architecture enables users to have full control of their data. The SMART4ALL repository permits user and group administration and authenticates users through the respective login page where credentials are entered (username & password). The administrator can fully manage the users and groups of the platform. Apart from creating or de-activating users, the administrator can set a quota limit for each user in order to manage optimally the storage capacity of the repository.

Nextcloud files are stored in a conventional directory structure and are encrypted during transit and storage. The SMART4ALL administrator created the required file structure of the project as presented in Figure 20.

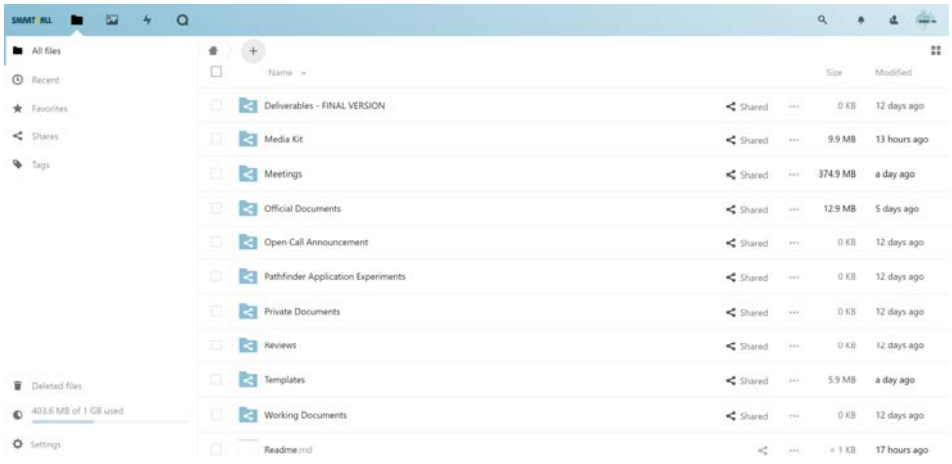

Figure 20: Repository file structure

Content can be shared by defining granular read/write permissions between users and groups (Figure 21). Alternatively, Nextcloud users can create public URLs when sharing files. Logging of file-related actions, as well as disallowing access based on file access rules is also available. Additionally, the allocation of cloud storage can be extended according to the needs of the experiment or project.

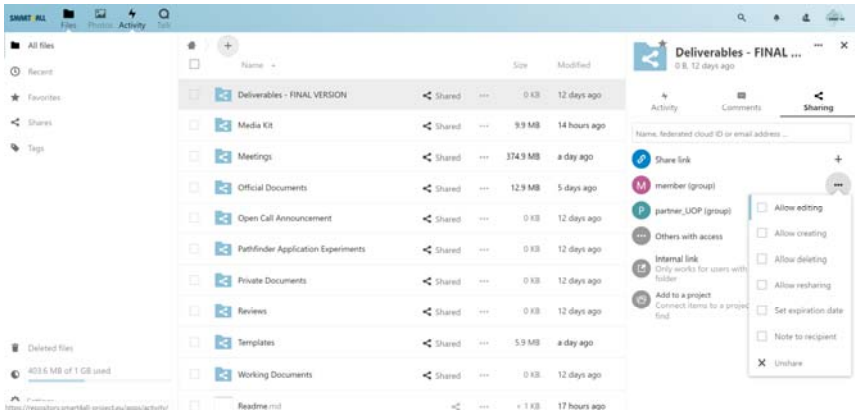

Figure 21: Sharing Capabilities

Tutorial video prepared in the context of SMART4ALL: https://repository.smart4all-project.eu/s/dxoJxgLDscag68G

Calendar:

Nextcloud calendar is a typical calendar application offering all the expected functionalities, features, and configuration options of any relative application. The Calendar app is a user interface for Nextcloud's CalDAV server. Easily sync events from various devices with your Nextcloud and edit them online. It offers reliable integration with other Nextcloud apps, WebCal Support, invitations can be issued to expected attendees, highly configurable reminders, and many more. A typical view of Nextcloud calendar is depicted in

More details can be found in this link: https://apps.nextcloud.com/apps/calendar

Tutorial video prepared in the context of SMART4ALL: https://repository.smart4all-project.eu/s/PsBNmSyZQXCYcxJ

Activities Organization and Monitoring:

In that respect, two different tools will be integrated and offered by default.

Nextcloud Tasks: This a very simple, easy to use, and flexible task management application. From there you can add and delete tasks/subtasks etc., edit their title, description, start and due dates, mark them as important, indicate the percentage of completeness, and tag them as required. A description can also be added to provide more detailed information. Tasks can be shared between users. Tasks can be synchronized using CalDav. Completed tasks can be hidden or shown.

More details can be found in this link: https://apps.nextcloud.com/apps/tasks

Tutorial video prepared in the context of SMART4ALL: https://repository.smart4all-project.eu/s/YAw9JrRRm5BM65F

Nextcloud Deck: This tool follows the very popular kanban style organization tools. As such, following a top-down hierarchical approach, the user can define Boards/Lists/Cards to organize its tasks. In each card the user can specify details such as Tags, assign to users, due dates, connection to project or another list/conversation, etc. Additionally, attachments can also be added increasing the manageability of the tasks while comments can be made by all assigned partners making it very easy to follow the progress of the task. Changes can be easily tracked through a detailed timeline as well as by issuing notifications.

More details can be found in this link: https://apps.nextcloud.com/apps/deck

Internal Communication:

For that purpose, the repository's efficient plugin system will be deployed in order to activate the Talk plugin as provided by the Nextcloud directory of apps Figure 22.

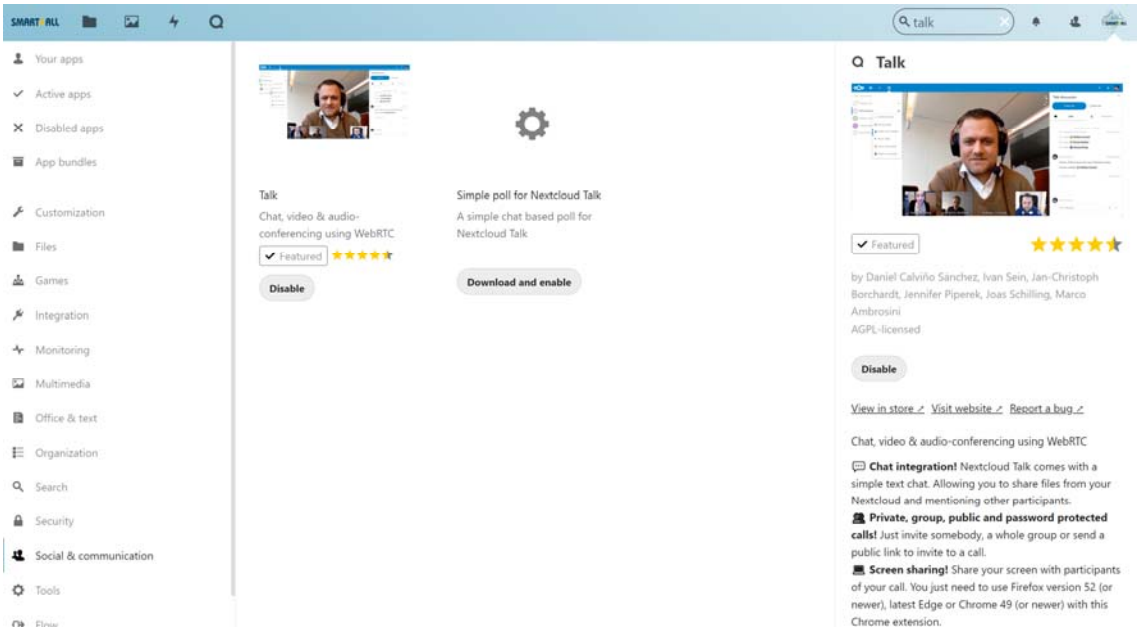

Figure 22: The Talk Plugin

Users of the repository platform can reach others through the repository web site and create topics with one or more users. Of course, private messages between the users is also supported through the Talk plugin. An indicative view of this app is given in Figure 23.

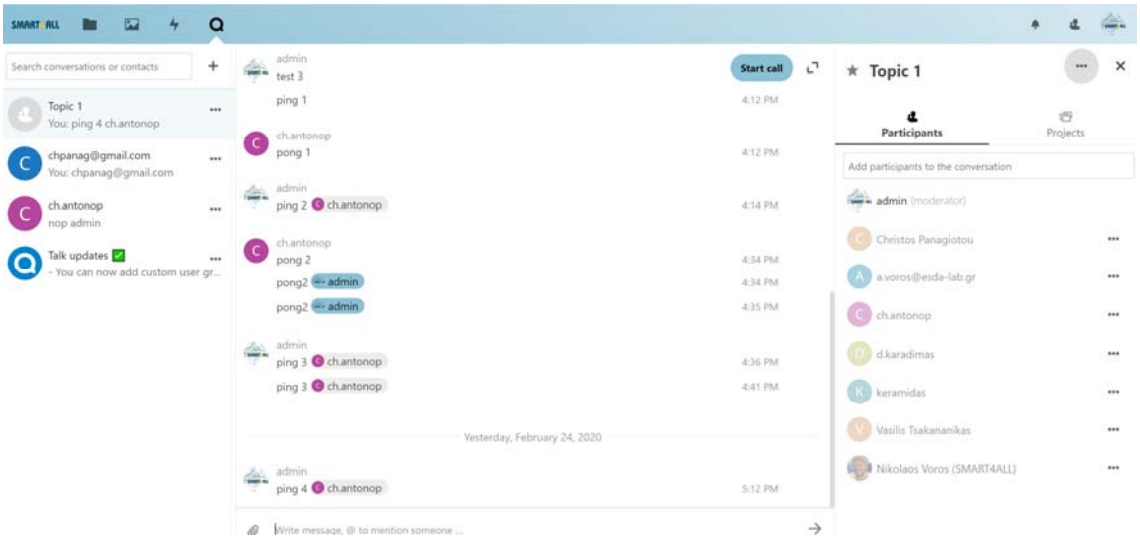

Figure 23: Conversation through talk

Additionally, apart from the traditional text messaging, Talk supports also speech/video conference along with screen sharing. Thus, Talk, as offered through the SMART4ALL repository platform, stands as the perfect alternative for ad-hoc meetings between the members of the consortium that facilitates their communication and collaboration in favor of the project's progress. Furthermore, a very good integration to other Nextcloud tools is worth noting which is continuously expanding.

More details can be found in this link: https://apps.nextcloud.com/apps/spreed

Tutorial video prepared in the context of SMART4ALL: https://repository.smart4all-project.eu/s/Ferb2fXAJbC7xpC

8 NETWORK INFRASTRUCTURE

Aiming to build a sustainable DIH and support and maintain all the services provided long after the end of the SMART4ALL project significant effort has been devoted into designing and developing an adequate Communication and Computation Network Infrastructure in the premises of ESDA Lab, representing the University of Peloponnese as the coordinating partner of SMART4ALL. This has been a critical endeavor since it required substantial investment both in terms of technological infrastructure as well as design and development effort.

Through this infrastructure SMART4ALL DIH will offer:

Ecosystem Services:

- Access to SMART4ALL network
- Identify funding opportunities
- SMART4ALL Open Calls
- Local and European funding frameworks
- Application-oriented ethics coaching
- Interconnection with other networks/ecosystems

Business Services:

- Application-oriented coaching for business development
- Partner search and business matchmaking

Technological Services:

- Application-oriented technological coaching
- Access to Computation and Communication resources through HaaS cloud infrastructures
- Technological matchmaking

SMART4ALL offers to the members of its network/ecosystem cloud computing services through state-of-the-art hardware & software infrastructure:

- Software-as-a-Service (SaaS)
- Hardware-as-a-Service (HaaS)
- Scalable architecture to meet workloads and provide 24/7 availability
- 24/7 support by technology experts
- High-speed network interconnection with GRNET backbone
- Open source SW employed from virtualization to application layer

SMART4ALL infrastructure is hosted and maintained by the Embedded System Design and Applications Laboratory member of the University of Peloponnese and acting as coordinating partner of SMART4ALL.

Here we present the main aspects of this infrastructure comprising ESDA Lab Data Center. The Data Center is comprised of a set of high-performance Rack Servers that are connected through high availability (Active/Active) networking equipment to a Storage Area Network Infrastructure and the ESDA Lab Gigabit Backbone Network. The connections to the Storage Area Network operate at FC 16 Gbps speed and to the ESDA Lab Backbone Network at 1 Gbps speed.

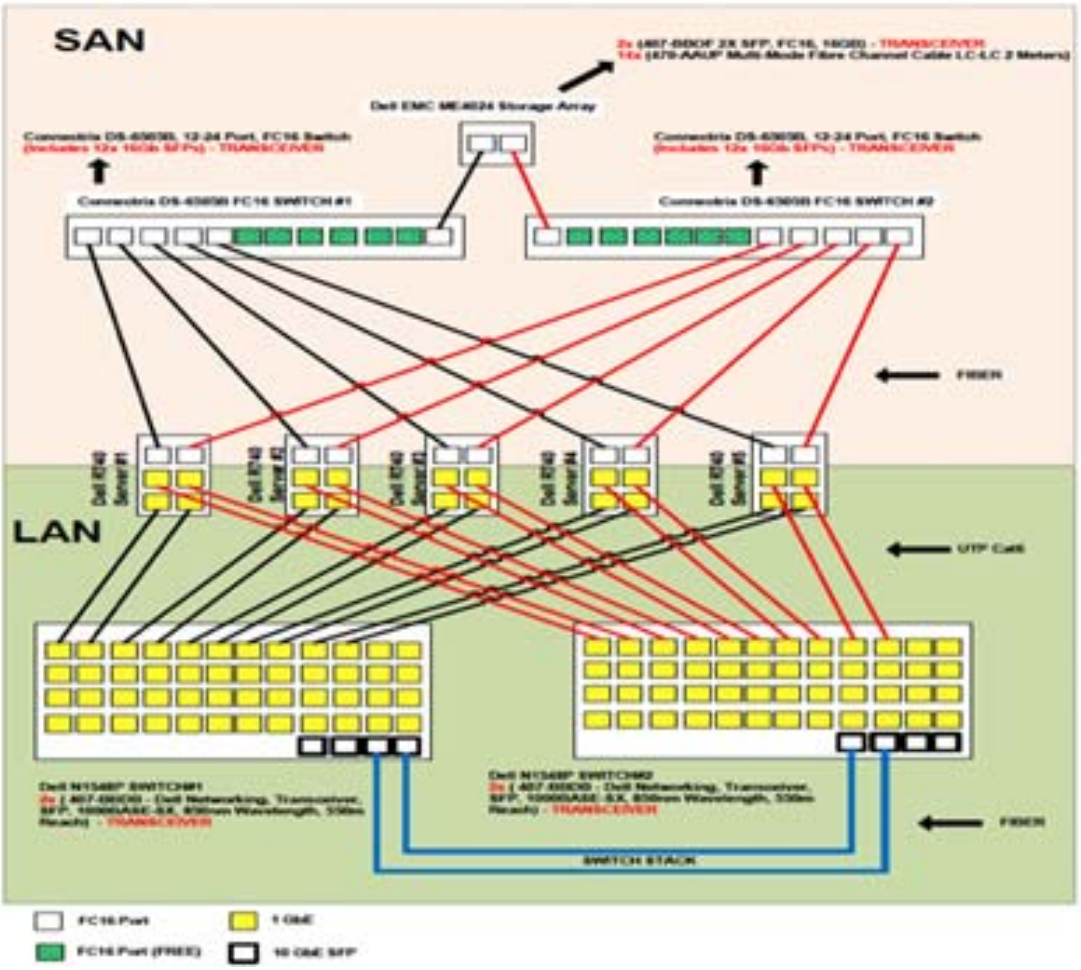

Figure 24: SMART4ALL Marketplace and Services Provisioning SAN Topology

The Data Center equipment is connected to the Internet via dual Active/Standby BGP peerings to the "GRNET – National Infrastructures for Research and Technology" Network at 1 Gbps speed. A cold standby BGP Router with identical backup configuration is also provided for use in case of possible hardware failure, so as to minimize the Internet connection downtime.

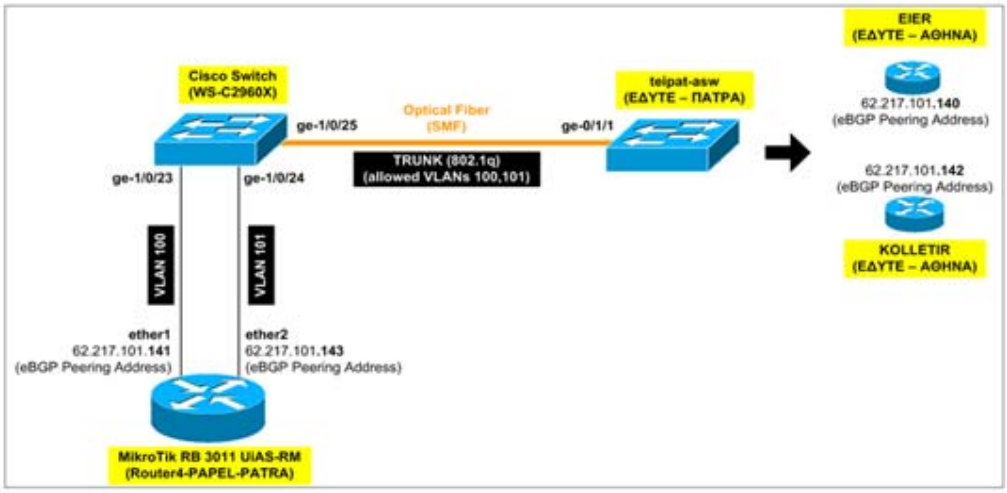

Figure 25: SMART4ALL infrastructure connection to the Internet

Computational wise the infrastructure is based on 8 Rack Mounted High-Performance Servers offering as a total number of 164 CPUs: 164 and 1.216 GBytes total memory while these resources are managed through Type 1 Hypervisors, specifically VMWARE ESXi / Proxmox Virtualization Technologies. The main components are depicted in the following figure. Furthermore, oversubscription can be supported i.e. statically assigned excess CPU, memory resources that totally available due to the nature of virtualization.

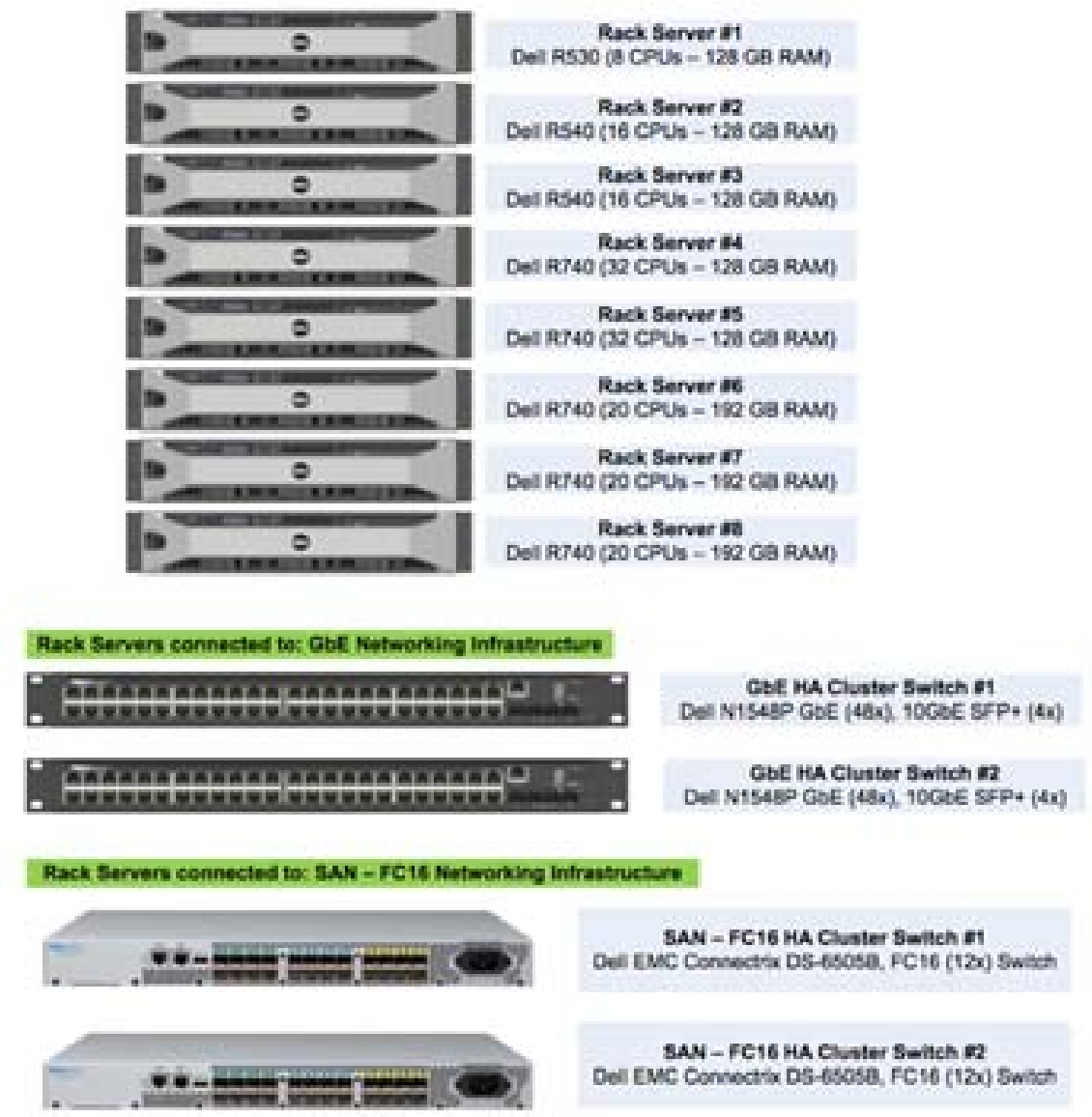

Figure 26: SMART4ALL computational infrastructure

Concerning the storage of data, a storage area network based on fiber channel (FC @ 16Gbps) infrastructure is developed offering a 20Tbytes secure and reliable capacity for all data handled (mirroring included). The main components are depicted in the following figures.

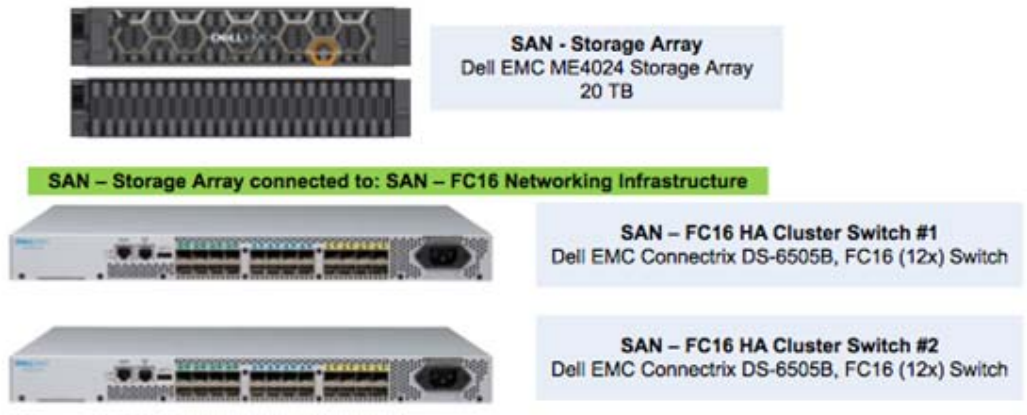

Figure 27: SMART4ALL infrastructure main network components (1)

Intercomponent communication is based on High Availability "Fiber Channel (FC)" Switches Cluster as depicted in the following figures. The FC Switch Infrastructure interconnects high-performance Rack Servers and the Storage Array. Each endpoint device has a redundant connection to the infrastructure.

Figure 28: SMART4ALL infrastructure main network components (2)

While a Network Attached Storage (NAS @ 1 Gbps) infrastructure offering a 30 Tbytes capacity is also provided based on the following components. The GbE Switch Infrastructure interconnects high-performance Rack Servers and Network Attached Storage Systems. It also provides the backbone network of the ESDA Lab. Each endpoint device has a redundant connection to the infrastructure.

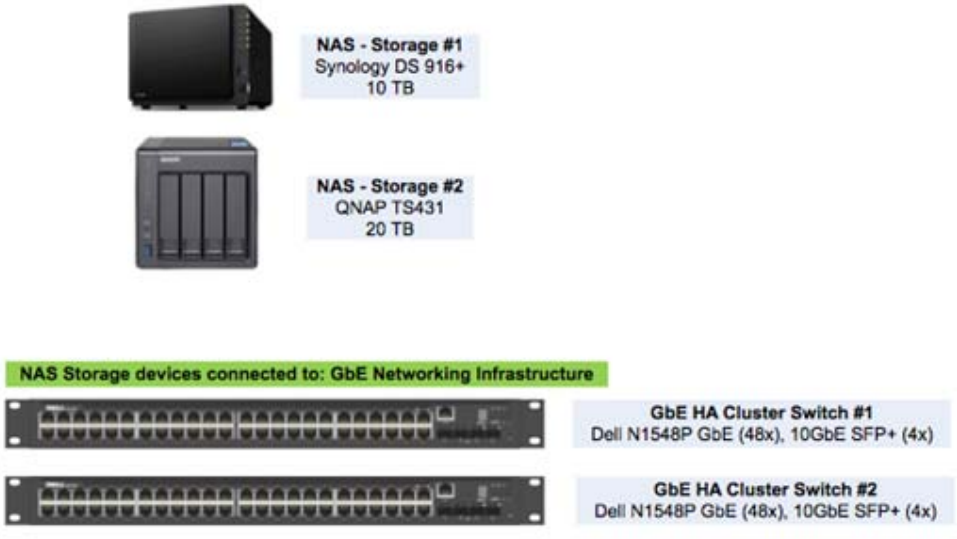

Figure 29: SMART4ALL infrastructure main storage components

Data and communication security are fully covered through next Generation Firewalls (NGFW)-Hardware / High Availability Cluster as depicted in the following figures. High availability is attained through an Active/Standby architecture. The NGFW infrastructure also provides VPN Services (Site-to-Site and Remote Access) to the Data Center. The VPN Services are used for remote administrative access to the available Data Center equipment and the implementation of backup/disaster recovery schemes.

Figure 30: SMART4ALL infrastructure security components

Last but not least in order to support High-Performance Computing experiments and respective research activities as well as collaborations, apart from general-purpose CPUs the infrastructure designed and developed offers state-of-the-art FPGA and GPU clusters as depicted in the following figures.

Figure 31: SMART4ALL infrastructure FPGA and GPU nodes

The allocation of resources (storage, computation, network) will be decided per case depending on the needs of the experiment or project. Relative guidelines will be prepared.

9 CONCLUSION

In the context of SMART4ALL, the technology portals, the services and tools provided through them and network and computational infrastructure upon which they are based are of cornerstone importance. Consequently, the main objective of this document is to present all the design and development aspects that came into play aiming to provide technological portals and services that are efficient, flexible, extensible and are developed το have a cycle far exceeding that of SMART4ALL project and thus continue to be useful for the purposed of the DIHs long after the end of the project itself.

Specifically, this document is accompanying the respective actual developments and reports all that is needed to have a clear understanding of their design, functionality, and features. The main pillar of this work is the Marketplace portal. In that respect, a distinct section is devoted to the technologies involved, the frontend interface (GUI), and the backend infrastructure to offer a clear presentation of the infrastructure. Then two concepts of Matchmaking services that are going to be supported and provided are presented. Both concerning the Marketplace as a repository and as an AI-based Matchmaking service support infrastructure the data model of the artefacts offered through the Marketplace is critical, thus a separate section is devoted to that cornerstone concept.

Then the tools facilitating project management and collaborative effort that are both used in the context of SMART4ALL and will be provided to the funded projects through the SMART4ALL open calls are presented.

Finally, representing a large percentage of devoted effort, the last chapter presents that high end, high-performance data center that is designed and developed specifically in the context and for the purposes of SMART4ALL activities. This is also of cornerstone important since upon this hardware infrastructure all the portals, tools, services, and artefacts are going to be based, maintained, supported, and offered. Finally, the computational and communication resources of this infrastructure will be offered to the funded projects through the SMART4ALL projects offering a critical advantage to this experiment.

As a last remark, the Marketplace, the tools, and the services will evolve and grow as the project progresses offering more and more resources and capabilities to the SMART4ALL ecosystem.

REFERENCES

[1] https://www.docker.com/

[2] http://mqtt.org/

[3] https://aws.amazon.com/

[4] https://cloud.google.com/compute

[5] https://azure.microsoft.com/en-us/

[6] https://www.openstack.org/

[7] https://www.chef.sh/

[8] https://puppet.com/

[9] https://www.ansible.com/

[10] https://kubernetes.io/

[11] https://www.jenkins.io/

[12] https://about.gitlab.com/

[13] https://reactjs.org/

[14] https://www.djangoproject.com/

[15] https://spring.io/

[16] Alexander Clark, Chris Fox, and Shalom Lappin, eds. The handbook of computational linguistics and natural language processing. John Wiley & Sons, 2013.

[17] Larkey, Leah S., Lisa Ballesteros, and Margaret E. Connell. "Light stemming for Arabic information retrieval." In Arabic computational morphology, pp. 221-243. Springer, Dordrecht, 2007.

[18] Liu, Tie-Yan. "Learning to rank for information retrieval. pp. 225-331. Found. Trends in Inf." Ret 3, no. 3, 2009

[19] Evanthia Faliagka et al. "On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed CV.", pp. : 515-528. Artificial Intelligence Review 42.3, 2014

[20] Schafer, J. Ben, Dan Frankowski, Jon Herlocker, and Shilad Sen. "Collaborative filtering recommender systems." In The adaptive web, pp. 291-324. Springer, Berlin, Heidelberg, 2007.

[21] Berkovich, Marina, Jan Marco Leimeister, Axel Hoffmann, and Helmut Krcmar. "A requirements data model for product service systems.", pp. 161-186, Requirements Engineering 19, no. 2 2014

[22] Silverston, Len, William H. Inmon, and Kent Graziano. The data model resource book: a library of logical data models and data warehouse designs. John Wiley & Sons, Inc., 1997.

ANNEX : SECURITY & PRIVACY REQUIREMENTS

SMART4ALL has identified the need and the paramount importance of security and privacy throughout the spectrum of the SMART4ALL services. In that context, the fundamental security and privacy policies and mechanisms have been investigated and applied horizontally to all the SMART4ALL services. In that context, the solutions that were adopted on the design and implementation phase of the SMART4ALL services were driven briefly by the following guidelines:

Secure Communication: Every communication of the services through the integration channels (REST, MQTT, etc) and web interfaces are secured through encrypted communication (TLS).

Authentication: Users and services interact each other through an authentication mechanism where needed.

Authorization: The SMART4ALL application and services implement authorized access to the resources.

Password storage: Another fundamental policy is that no passwords are stored in any of the SMART4ALL applications' databases as plain text. All passwords are encrypted and no one can have access and recover the original password phrase.

Respective attention was given on how the services handle data collected by the users of the SMART4ALL services. In that context the policies below were adopted.

Consent Forms: Any web resource that operates in the European territory requires mandatory consent of the users to store and process their data, which they share in the registration form. SMART4ALL provides detailed text regarding data protection. Apart from this, respective checkboxes help users to confirm all the information shared and agree to it.

Right to Access: Every user has the right to turn to the developers of the SMART4ALL applications and obtain any data or information they have shared about themselves in the app without any hassles. However, such GDPR requirements concerns projects that are both currently active, as well as closed.

Right to Modification: Every user of has the right to modify their personal details at any point of time apart from the access to change the standard 'login & password' combination.

More details on the security and privacy solutions of the SMART4ALL are given on the Data Management Plan Deliverable (D1.2) and subsequently version of that document.