CLOUD FOR DATA-DRIVEN POLICY MANAGEMENT

Project Number: 870675 Start Date of Project: 01/01/2020 Duration: 36 months

D7.4 DATA MARKETPLACE: DESIGN AND OPEN SPECIFICATION

Dissemination Level	PU
Due Date of Deliverable	(31/08/2021), Month 20
Actual Submission Date	02/08/2021
Work Package	WP7, Communication, Exploitation, Standardisation, Roadmapping & Business Development
Task	T7.2
Type	Report
Approval Status	
Version	v1.0
Number of Pages	p.1 – p.29

Abstract: The deliverable D7.4 Data Marketplace: Design and Open Specification describes the initial design and specifications of the PolicyCLOUD Data Marketplace. The latter will be a unified web-based platform consisting of two (2) core services, its front-end and back-end services, offering to its users various ready-to-use solutions, by supporting different kinds of assets.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/ her sole risk and liability.

PolicyCloud has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 870675.

Versioning and Contribution History

Version	Date	Reason	Author
0.1	14/05/2021	ToC	Argyro Mavrogiorgou, Thanos Kiourtis (UPRC)
0.2	28/05/2021	Contribution in Sections 1, 2	Argyro Mavrogiorgou, Thanos Kiourtis (UPRC)
0.3	03/06/2021	Contribution in Sections 3.1, 3.2	Argyro Mavrogiorgou, Thanos Kiourtis (UPRC)
0.4	14/06/2021	Contribution in Sections 3.3, 3.4	Argyro Mavrogiorgou, Thanos Kiourtis (UPRC)
0.5	25/06/2021	Contribution in Sections 4, 5	Argyro Mavrogiorgou, Thanos Kiourtis (UPRC)
0.6	06/07/2021	Check and revision of all Sections	Argyro Mavrogiorgou, Thanos Kiourtis, Fotis Karagiannis (UPRC)
0.7	16/07/2021	Vizualized material for Section 3.2	Marieke Willems (TRUST-IT)
0.8	20/07/2021	Internal review	Panayiotis Michael (ICCS), Giannis Ledakis (UBI)
0.9	22/07/2021	Updates based on received feedback	Argyro Mavrogiorgou, Thanos Kiourtis, Fotis Karagiannis (UPRC)
1.0	28/07/2021	Quality check	Argyro Mavrogiorgou (UPRC)

Author List

Organisation	Name
UPRC	Argyro Mavrogiorgou
UPRC	Thanos Kiourtis
UPRC	Fotis Karagiannis
TRUST-IT	Marieke Willems

Abbreviations and Acronyms

Abbreviation/Acronym	Definition
AI	Artificial Intelligence
API	Application Programming Interface
CRUD	Create Retrieve Update Delete
EC	European Commission
EOSC	European Open Science Cloud
EU	European Union
DL	Deep Leaning
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
ML	Machine Learning
NoSQL	Not only SQL
PDT	Policy Development Toolkit
REST	Representational State Transfer
UI	User Interface

Contents

List of Tables

List of Figures

Executive Summary

This deliverable (entitled “Data Marketplace: Design and Open Specification”) describes the initial design and specifications of the PolicyCLOUD Data Marketplace. The Data Marketplace will be a unified web-based platform consisting of two (2) core services, its front-end and back-end, offering its users various ready-to-use solutions. More specifically, it will provide to the wider research and innovation community various assets (i.e. objects/solutions) in different domains.

As a Marketplace, the current design provides information on how several users (and contributors) can interact with the Marketplace in order to ingest and retrieve the respective assets, through potential user journeys that are analyzed. Moreover, the deliverable introduces the different types of offered assets, since the Data Marketplace will go beyond policies and tools and will store and offer additional elements, including rich descriptions of the offered assets, algorithms, validation datasets, and in general different outcomes of the project.

To support the aforementioned functionalities, the PolicyCLOUD Data Marketplace specifications exemplify the respective layers of the Marketplace, the functionalities supported by each layer, the interaction points with different stakeholders as well as the technical details that will drive the implementation of the Marketplace. Finally, it reviews the current state of the art in order to identify the baseline technologies and approaches for the realization of the implemented Marketplace.

1 Introduction

PolicyCLOUD aims to be a cloud-based data-driven policy management platform, enabling its stakeholders to model, analyze, evaluate and optimize their policies using a variety of Big Data tools and services. Among the components that will be developed and offered from the consortium is the Data Marketplace.

The main idea of the Data Marketplace is to create a community of users who will be able, through the platform that will be developed, to provide and share various assets related to the scope of the PolicyCLOUD. It also aims at being one of the project's main ambassadors to different external communities (third parties), by disseminating and demonstrating the main outcomes of the project. To this end, the platform should provide the means for storing and retrieving (after searching) several types of assets. Examples of assets to be provided by the Data Marketplace are analyzed in Section 3.3 (Offered Assets).

In summary, the Data Marketplace will consist of a public web-based environment with many different APIs and functionalities, in order to cover all the different requirements of the project's stakeholders. In the following sections of this document, the design and the specifications of the PolicyCLOUD Data Marketplace will be analyzed in detail.

1.1 Objective of the Deliverable

The main objective of this deliverable is to provide the ground for the realization of the PolicyCLOUD Data Marketplace. It will analyze the architecture and the specifications of the Marketplace, and its development will be performed based on them. In addition, the deliverable will outline the potential users of the Marketplace (i.e. providers and consumers) and clarify the reason for its implementation and existence, describing diverse user journeys. Finally, it will list the types of assets (offered assets) that will be populated through the Marketplace and will be provided through it to the end-users.

1.2 Structure of the Deliverable

This document is structured as follows: Section 2 introduces relevant work and approaches that serve as market platforms in order to identify potential links and baseline technologies for the Data Marketplace. Section 3 describes the scope of the Data Marketplace and the users that will interact with it, along with some user journeys. It also lists the offered assets towards these users of the Marketplace. Section 4 is the core section of this report, presenting in deep detail the architecture and the specifications of the Data Marketplace, analyzing its functionalities/capabilities that will be supported. The report concludes with a reference to the presented material in Section 5.

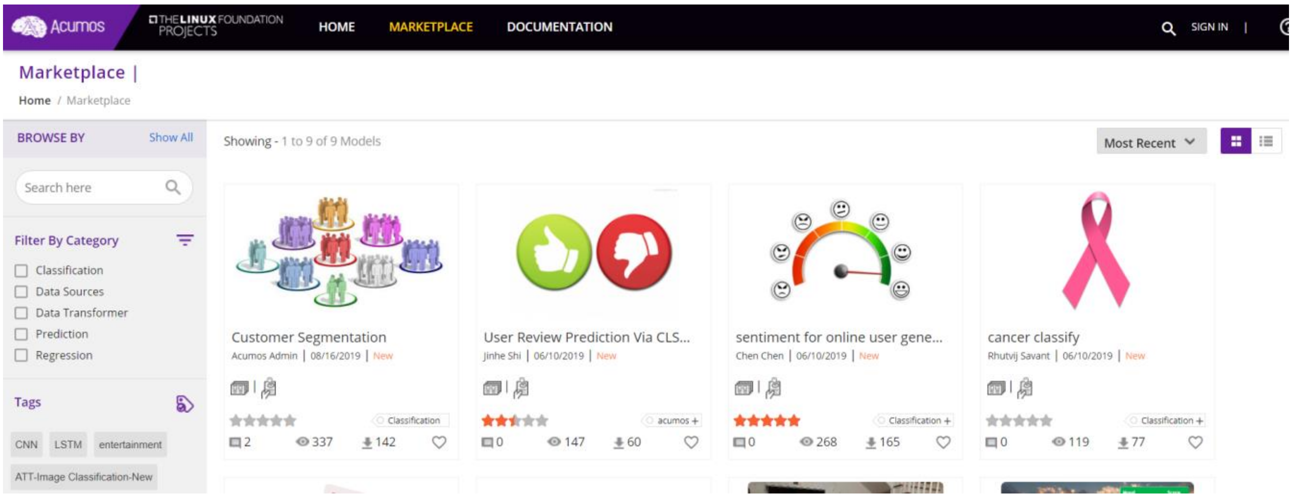
2 Relevant Work in Marketplaces

This section describes relevant works that have been identified and can be exploited as a baseline to realize the PolicyCLOUD Data Marketplace.

2.1 Acumos Marketplace

Acumos [1] is a platform for open-source code that concerns AI, ML, and DL models. It can be exploited by data scientists and developers, for the purpose of directing them to an AI-centric process for software production. Among others, it includes features and provisions for toolkits, chain models, marketplace, visual IDE for AI applications, that could be exported as Docker images to run in private environments or in cloud.

The Acumos Marketplace is a web-based platform designed to make it easy to discover, explore, and use AI models. It has both public and private catalogs, where the former are the only models displayed to non-logged-in users. In the “Marketplace” page (Figure 1) the users can find the offered models, being able to search them based on keywords, filter them based on various tags or their categories, and also sort them by their popularity, their number of downloads, rating and other metadata.



Customer Segmentation
Acumos Admin | 08/16/2019 | New
Classification
2 337 142

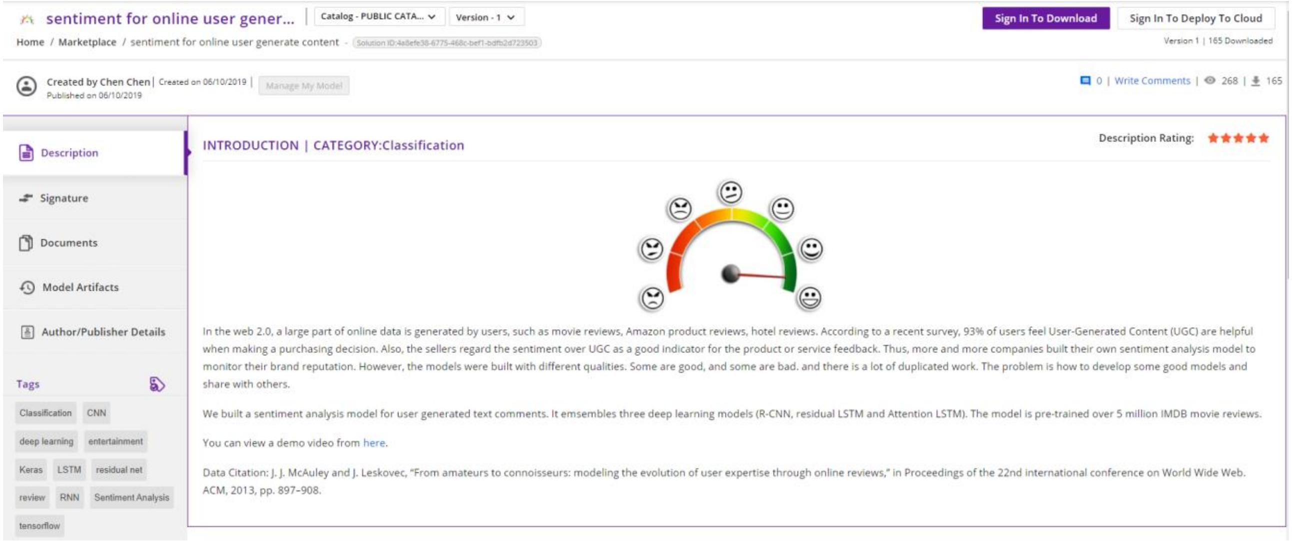
User Review Prediction Via CLS...
Jinhe Shi | 06/10/2019 | New
acumos +
0 147 60

sentiment for online user gene...
Chen Chen | 06/10/2019 | New
Classification +
0 268 165

cancer classify
Rhutvij Savant | 06/10/2019 | New
Classification +
0 119 77

FIGURE 1 - SNAPSHOT OF THE “MARKETPLACE” PAGE OF THE ACUMOS MARKETPLACE.

Moreover, there is a “Model Details” page (Figure 2) from which the users can find more information for a selected model. More specifically, they can find information for the model such as its description, signature, relative documents and model artifacts, and other details of the author/provider, including even some metadata such as the number of views, comments and downloads.


sentiment for online user gener...
Catalog - PUBLIC CATA...
Version - 1
Sign In To Download
Sign In To Deploy To Cloud
Home / Marketplace / sentiment for online user generate content - Solution ID:4a8efe38-6775-468c-bef1-bdfb2d723503
Version 1 | 165 Downloaded
Created by Chen Chen | Created on 06/10/2019 | Published on 06/10/2019
Manage My Model
0 | Write Comments | 268 | 165
Description
Signature
Documents
Model Artifacts
Author/Publisher Details
Tags
Classification
CNN
deep learning
entertainment
Keras
LSTM
residual net
review
RNN
Sentiment Analysis
tensorflow
INTRODUCTION | CATEGORY:Classification
Description Rating: ★★★★★

In the web 2.0, a large part of online data is generated by users, such as movie reviews, Amazon product reviews, hotel reviews. According to a recent survey, 93% of users feel User-Generated Content (UGC) are helpful when making a purchasing decision. Also, the sellers regard the sentiment over UGC as a good indicator for the product or service feedback. Thus, more and more companies built their own sentiment analysis model to monitor their brand reputation. However, the models were built with different qualities. Some are good, and some are bad. and there is a lot of duplicated work. The problem is how to develop some good models and share with others.

We built a sentiment analysis model for user generated text comments. It emsembles three deep learning models (R-CNN, residual LSTM and Attention LSTM). The model is pre-trained over 5 million IMDB movie reviews.

You can view a demo video from here.

Data Citation: J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews," in Proceedings of the 22nd international conference on World Wide Web. ACM, 2013, pp. 897–908.

FIGURE 2 - SNAPSHOT OF THE "MODEL DETAILS" PAGE OF THE ACUMOS MARKETPLACE.

Another important feature of the Acumos Marketplace is that it enables its users (who have been assigned the Publisher role) to publish their own models upon request. Before the publication of the models, the publish requests for new solutions go through content control and at the end they are approved or declined.

For the front-end of the PolicyCLOUD Data Marketplace, it is intended to develop and implement a website with similar functionalities in order to enable users both to search and retrieve the offered assets of the Marketplace and also to upload their own assets (with content control). As a result, a "Model Details" page (i.e. Asset's page) will be built, where the core details for each asset will be presented and its corresponding retrieval can occur.

2.2 5GTANGO Catalogue

5GTANGO [2] was an EU project whose purpose was to enable the flexible programmability of 5G networks and devise and realize a radical shift in the development of software for 5G-ready applications. In the context of the project, a market platform named as "Catalogue" was developed in order to host and offer different network-related services.

The Catalogue is an instrumental component of the 5GTANGO environment, presenting different parts of it, as it is depicted in Figure 3. Primarily, it hosts the different descriptors of the 5GTANGO packages. Since 5GTANGO aimed at a multi-platform environment, it enables the developers to orchestrate and deploy their services using different Service Platforms. In this context, the 5GTANGO Catalogue has been adapted to support the storing and retrieval of new packages.

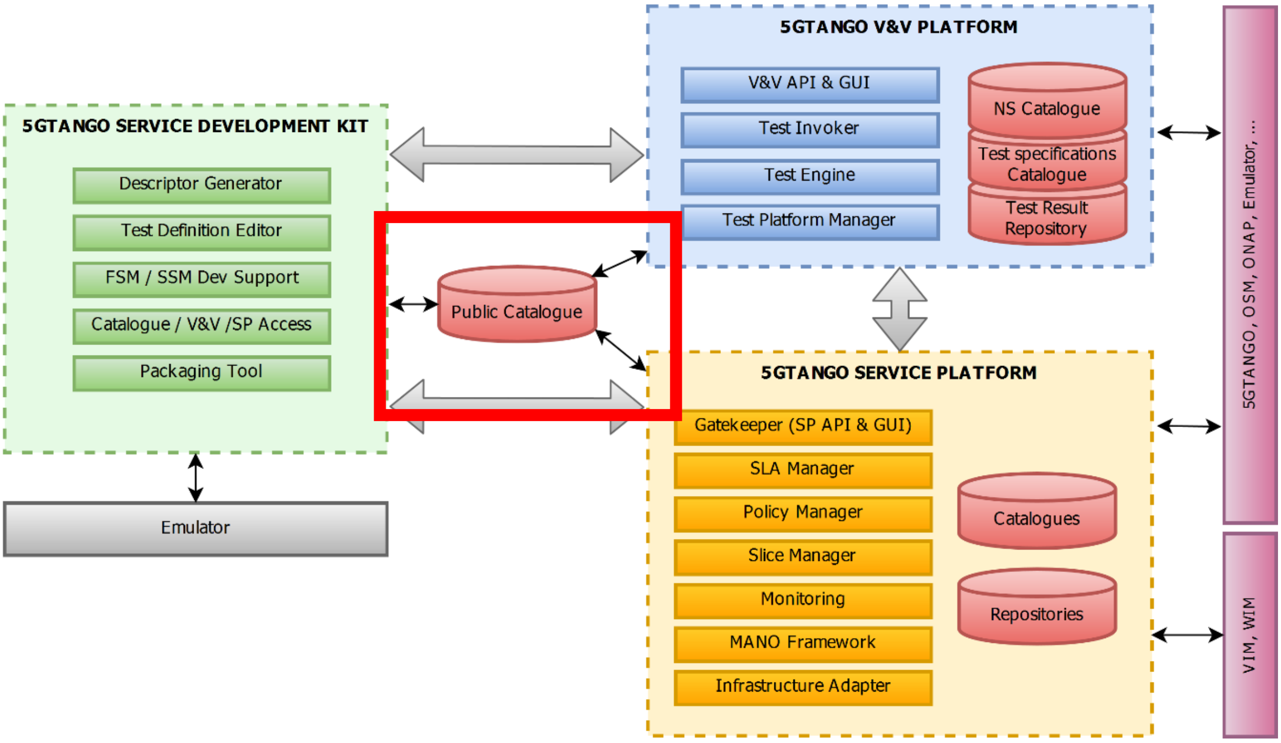


FIGURE 3 - 5GTANGO OVERALL ARCHITECTURE DESIGN AND THE ROLE OF THE CATALOGUE.

Moreover, the Catalogue is aligned with the principle of persistent storage by extending the hosted descriptors with valuable fields for successful data integration, accuracy in the format of the document, confirmed time of creation, etc. In this way, it enables the development of enhanced operations for Creating, Retrieving, Updating and Deleting (CRUD) descriptors inside it, while reassuring the correct data format of the stored documents.

Going beyond conventional data storage, the 5GTANGO Catalogue provided intelligent functionalities in this 5G environment. Since the types of information vary, one of the requirements satisfied by the Catalogue, was the full-text search capability in structure-agnostic documents. Since the schema of the diverse documents (i.e. descriptors of different types of assets) is variable, the Catalogue provided searching capabilities without the necessity of indexes. Thus, it provides seamless retrieval abilities in deep-hierarchical machine-readable document structures.

Furthermore, besides the plain NoSQL document store for the diverse descriptors, the Catalogue provided a scalable file system for hosting the artifact files, required for the instantiation lifecycle of the services. The Catalogue also provided a set of end-points where the CRUD methods were supported for the different descriptors of the project. Finally, as previously addressed, the Catalogue was responsible to store different objects with respect to the management of the services’ lifecycle and more.

The PolicyCLOUD Data Marketplace will follow a similar approach in the back-end side by utilizing the main functionalities that have been described above, such as:

- Support of different types of assets on the Storage layer (Section 4.1.1).
- Support for descriptions of the stored assets (Section 4.1.2).
- Support for advanced search and retrieval of assets (Section 4.1.2).
- Support for various interaction methods (Section 0).

2.3 FINSEC Marketplace

The FINSEC project has recently established and launched the Finsecurity.eu [3] market platform. FINSEC develops integrated (cyber/physical) security solutions for the critical infrastructures of the financial sector. Therefore, Finsecurity promotes and offers solutions and services for the security of the critical infrastructures of the finance sector. It serves as a promotional channel for the solutions and services that have been developed in the FINSEC project (Figure 4), while providing access to relevant knowledge assets such as whitepapers and training presentations about security and digital finance services.

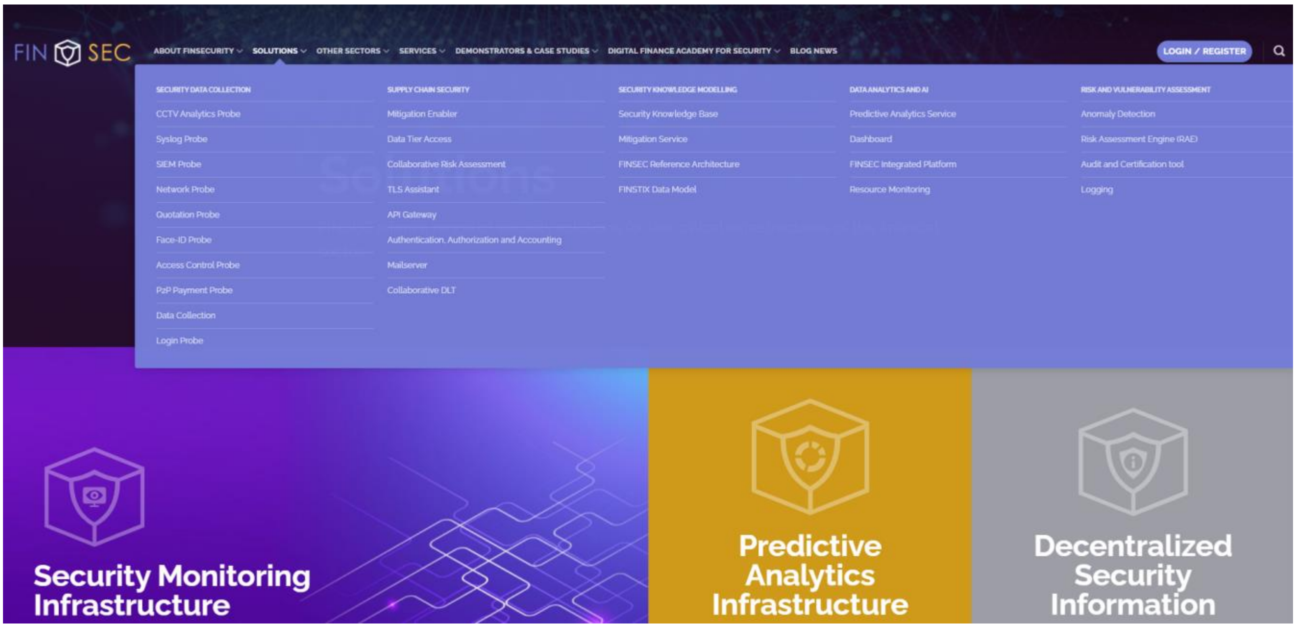


FIGURE 4 - SNAPSHOT OF THE FINSECURITY.EU PLATFORM, INCLUDING A LIST OF ITS SOLUTIONS.

Finsecurity offers selected contents to registered members only. It therefore provides a registration mechanism, along with user management. Registered users have access to premium content and updates. The Finsecurity community is gradually expanded based on members of other projects in critical infrastructures security. The latter members are also expected to contribute additional solutions.

From what has been described above, it is understood that PolicyCLOUD Data Marketplace and Finsecurity have common objectives, a common strategy for managing and promoting the offered solutions and possibly common communities.

3 Scope, Users & Assets of Data Marketplace

This section describes the scope of the PolicyCLOUD Data Marketplace and summarizes the main users that will interact with it along with the respective user journeys. Moreover, it lists the different types of assets that will be available through the platform and the types of its users.

3.1 Scope

The scope of the Data Marketplace in relation to the other components of PolicyCLOUD (like PDT, etc.) is different and unique. More specifically, the Marketplace is intended to offer through its platform ready-to-use sets of solutions related to various subjects. At the same time, it will enable its users to upload different types of assets (according to the list in Section 3.3) that will be provided to all the Marketplace's users in order to be able to use them to solve/handle any of their needs. Definitely, it will be up to the assets' providers to limit the end-users of the assets that they offer through the platform. However, in the current implemented version of the Data Marketplace, it is assumed that all the uploaded assets will be provided to each registered user.

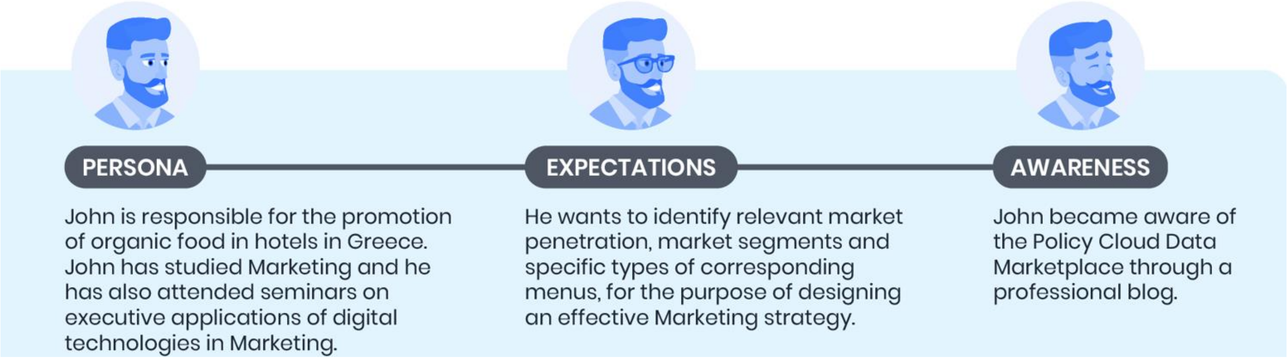
As analyzed in the next subsection through the user journeys that are presented, some reasons that will motivate the users to visit the Data Marketplace may be to search and retrieve assets that will help them either to resolve some of their business issues, or to achieve some of their educational goals, or even use them just for their personal interests.

Therefore, the Data Marketplace is considered as a smart user-based repository of assets. Its users will provide to the community of the platform ready-to-use solutions/tools to various subjects related to the areas of interest of the project.

3.2 User Journeys

The following subsections present potential user journeys of the Data Marketplace users, their interaction with the platform and the reasons why they may be interested in visiting it and taking advantage of its functionalities. The described examples include users looking to retrieve (download) assets for their benefit and developers interested in disseminating their assets by uploading them to the Marketplace. The journeys also describe what are the expectations of the users, explain possible difficulties that they may face in the context of their work, and finally how they resolve them by retrieving relative solutions from the Marketplace.

3.2.1 Organic Food in Hotel Menus



PERSONA

John is responsible for the promotion of organic food in hotels in Greece. John has studied Marketing and he has also attended seminars on executive applications of digital technologies in Marketing.

EXPECTATIONS

He wants to identify relevant market penetration, market segments and specific types of corresponding menus, for the purpose of designing an effective Marketing strategy.

AWARENESS

John became aware of the Policy Cloud Data Marketplace through a professional blog.

GOALS	ACTIONS	RESULTS
Recognize attitudes for the organic food in hotels.	He finds a publicly available dataset from the 'Discover' page that includes information from web contents, booking platforms and social media of hotel reviews for different hotels in Greece.	He extracts useful information regarding the product penetration based on the hotel type, location, and satisfaction with organic food.
Identify contribution of organic food products to the rating of hotels, based on the variety of offered organic product.	He finds and downloads a publicly available ML algorithm in order to apply Regression analysis in the hotel ratings based on the presence and satisfaction with organic food.	He finds out that there exists a tight relation between organic food and hotel rating, since almost 45% of the overall rating depends on the high-quality food.
Identify variety of offered organic product in current hotel menus.	He exploits a publicly available content analysis tool for finding online information for all the existing online hotel-restaurants menus retrieved by hotels' sites and promotional material.	He discovers that currently, the hotels in Greece offer a variety and mix of organic food with some ingredients being repeated in almost every recipe.
Detect specific organic product combinations.	He finds and uses a publicly available cluster analysis tool.	He extracts various combinations of organic food products from specific locations in Greece that he can promote.
Identify market segments of adoption of organic food products.	He exploits a publicly available statistical tool for applying statistical associations between clusters of hotels and hotels characteristics.	He realizes segmentation of hotels on the basis of users'/non-users' profiles of different offerings and hotel characteristics.
Find the profiles of non-adopting hotels as potential targets for conversion to adopters.	He exploits a publicly available analytical tool for the analysis of the patterns of proximities of characteristics of adopting and non-adopting hotels, on the basis of cluster analysis.	He identifies segments of non-adopters for targeted promotional actions for conversion to adopters, including hotels from specific regions with different ecological beliefs among them.

TABLE 1 - USER JOURNEY FOR MARKETING DIRECTOR.

3.2.2 Association of Terrorism with Geolocation

PERSONA

Isabella is a PhD student in the Psychology department of the Saint Louis University in Madrid.

EXPECTATIONS

Isabella participates in a research project in which she tries to draw conclusions about the association of terrorism with geolocation.

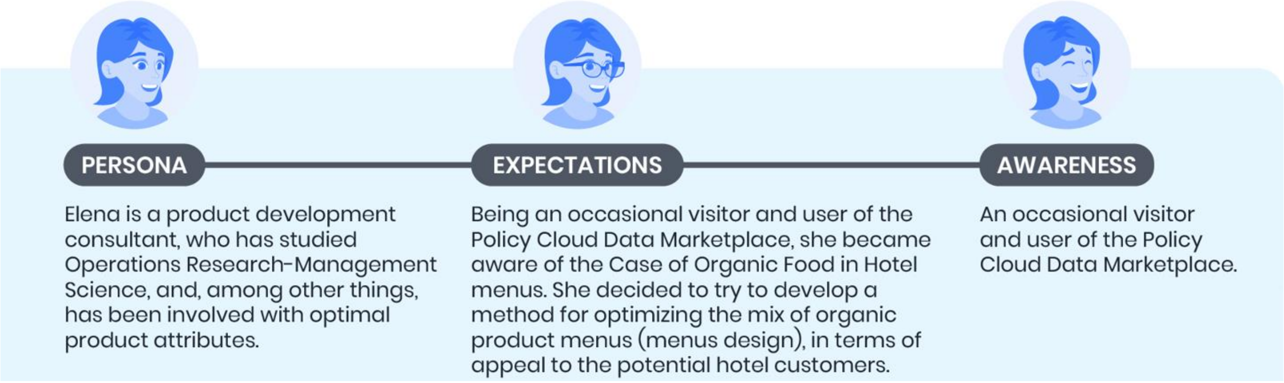
AWARENESS

Isabella became aware of the Policy Cloud Data Marketplace through a recommendation by one of her former university professors.

GOALS	ACTIONS	RESULTS
Extract knowledge for how terrorism is being affected by the geolocation.	She searches and finds various datasets that contain relevant information, but most of them are not available for the unregistered users.	She begins the procedure of creating a free account in the Data Marketplace.
Create an account.	She wants access to all Data Marketplace material that is not available to unregistered users. Thus, she fills in her personal data to create an account.	She confirms her email, she creates the account, and she navigates across the various assets of the Marketplace.
Identify the correlation between terrorism and geolocation characteristics.	She access a sample dataset that contains a list of terrorist attacks that placed in diverse geolocations across Europe. Since this dataset (i.e. asset) is of high-interest, she then downloads it.	She extracts useful information for her research from the dataset, by manually categorizing terrorist attacks based on the different geolocation characteristics.
Find additional tools to help her with her research.	She continues to explore the assets offered in the Data Marketplace, by reading the description of each asset.	She clicks in the suggestion button, which helps her to find relevant information.
Identify the common characteristics among the diverse geolocation that have been attacked.	In the suggestion list, she finds and reads a description of an algorithm that performs association rule mining in a research dataset including information by various geolocation attacks in Spain. To verify her selection, she reads the asset's reviews, and then downloads it.	She executes the asset by herself to outline the results in her research, performing various experiments upon it. She then applies the asset on top of the previously downloaded dataset, to compare the derived results.
Find statistical results about terrorism and geolocation.	She finds and downloads a statistical tool with data visualization results about different types of terrorism attacks and the corresponding geolocation characteristics of these attacks.	She discovers the impact of geolocation into the evolution of terrorism, by visualizing the extracted results on top of the previously downloaded datasets.
Express satisfaction of Data Marketplace functionalities.	She reviews and writes a comment in all the assets that she has downloaded.	She helps other users to decide whether these assets will be useful for them or not.

TABLE 2 - USER JOURNEY FOR PSYCHOLOGY RESEARCHER.

3.2.3 Optimization of Organic Food in Hotel Menus



PERSONA

Elena is a product development consultant, who has studied Operations Research-Management Science, and, among other things, has been involved with optimal product attributes.

EXPECTATIONS

Being an occasional visitor and user of the Policy Cloud Data Marketplace, she became aware of the Case of Organic Food in Hotel menus. She decided to try to develop a method for optimizing the mix of organic product menus (menus design), in terms of appeal to the potential hotel customers.

AWARENESS

An occasional visitor and user of the Policy Cloud Data Marketplace.

GOALS	ACTIONS	RESULTS
Develop and upload a relevant application to the Data Marketplace.	She is visiting the 'Discover' page of the Data Marketplace in order to find similar applications to the one she wants to make.	She does not find any relevant application.
Understand the asset form that she should upload.	She clicks the 'About us' page of the Data Marketplace and she reads the provided documentation and Q&A.	She understands the provided material, and she is able to upload her asset form. However, a mandatory step is to create an account in the Data Marketplace.
Create an account.	She fills in her personal information to create an account in order to be able to upload her application.	She confirms her email, she creates the account, she logins in, and from now on she can access the 'Upload Asset' page of the platform.
Review the databases of the topic of interest.	She searches the relevant databases from web, social media, portals in order to discover the relevant hotel reviews.	She selects the results of the applications that are more relevant to her goal.
Consider the relationships between the hotel ratings and the organic food menus.	She finds and downloads an optimized Regression analysis of hotel ratings on presence and satisfaction with organic food.	She extracts the optimal degree of presence of organic food on hotel ratings using her implemented application.
Upload the dataset that she created before.	She navigates to the 'Upload Asset' page again.	She writes a description and submits it.
Upload the optimize regression analysis tool.	She chooses the dataset, the regression algorithm that she created (results of the contents analysis of hotel reviews) and the Policy (Optimal degree of presence of organic food on hotel ratings).	She submits the files.

TABLE 3 - USER JOURNEY FOR DEVELOPMENT CONSULTANT.

3.3 Offered Assets

In the context of PolicyCLOUD, the Data Marketplace will offer several types of assets (objects) that may result from the separate procedures and mechanisms implemented in the PolicyCLOUD platform. The main assets' categories that will be exploited and added into the Data Marketplace are listed and described below:

- Algorithms and Tools in general, implementing specific solutions. By the term algorithm, we refer to the type of asset that includes code in any programming language (e.g. Java, Python, Python Notebooks, etc.), which once retrieved can be executed to produce a result (e.g. to validate/create a new policy) or even integrate it into an existing project of the end-user (depending on the provided asset and its author/provider). In terms of tools, there may be analytics tools, management tools and so on. The idea behind the tools is that instead of offering parts of code, there will be executable programs (i.e. tools) that integrate different solutions for a subject/a couple of subjects.
- Policies, which are the project's core outcome. There will be offered "ready-to-use" policies, which may have been derived by the PDT, whereas there can also be examples and templates of policies.
- Datasets that may "feed" the previous algorithms (e.g. training and tests datasets for the algorithms).
- Project's presentations/documents/deliverables could also extend the list of the offered assets. Maybe also tutorials in text format that can be considered as "how-to" guides regarding various project's subjects.
- Videos in general (e.g. webinars, tutorials, etc.).
- Links to other platforms/APIs:
  - In the case that there will be services (APIs) that cannot be added/stored to the Data Marketplace, but are still relevant to the project and therefore should be included in order to be "promoted" (e.g. PDT).
  - In order to offer assets that are already stored in public repositories (e.g. in Gitlab).

All these types of assets emerge from the needs identified by all the PolicyCLOUD partners, and have been extended with additional suggested assets as a fruitful way to exploit and disseminate the project's results to the community. More specifically, the list above has been formed based on the requirements analysis that has been accomplished over various inputs collected in the scope of Task 7.2.

3.4 Users

As described in the Introduction section of the deliverable, the Data Marketplace aims to create a community of users who will be able to provide and share various assets through the platform. Thus, in the Data Marketplace, it is envisioned that there will be many and different users - different by the meaning that they will have diverse backgrounds, specialties, interests, and objectives.

From the beginning, the users can be distinguished into human users and machine users. In deeper detail, in addition to the normal "human" users, it was considered as a nice extension to make the Data Marketplace more interoperable and able to offer its assets and services to other services, platforms, and systems, which, with the appropriate interfaces, will be able to provide or retrieve the offered assets in a more direct way. In general, the users of the Data Marketplace will be able to interact with it either through the website that will be created (Front-end - Section 4.2) or directly with the Marketplace's back-end API (Back-end - Section 4.1). Human users will be able to use both ways, but the machines/services will be able to retrieve or provide new assets in the Marketplace, with a more direct and automated process, interacting directly with the back-end.

In addition, users are divided into providers and end-users. Providers can upload and at the same time publish to Data Marketplace their assets. If the providers are the authors of the provided asset, they can also be called producers/owners. The end-users of the Data Marketplace, also named as consumers, are the users who will benefit from the platform, being able to retrieve the offered assets and/or services for which they are interested in, and use them by deploying and executing them, depending on their needs.

In general, the users that will form the community that will be created around the Data Marketplace, are expected to be:

- Policy makers, who can contribute and enhance the Data Marketplace with policies.
- Developers, who will be able to share their algorithms and datasets to the platform in order to provide the end-users with their complete solutions regarding a field of use.
- Scientists, who can contribute assets that would like to share with other scientists for the same reason as above.
- Services, that during or after their execution generate and store new assets on the Data Marketplace, for example policies/datasets. An example of a service – provider could be the PDT.
- Any other interested third parties, who want to achieve some personal, business, or educational objectives and will benefit from the Data Marketplace. Thus, the platform is intended to serve the needs of a wider community and it will allow everyone to join (through registration) the Marketplace and obtain the respective assets.

Additional examples to the abovementioned types of users are the PolicyCLOUD members (e.g. technical members, analysts, etc.) that fall in the classes described above, and other authorized users (e.g. members from other collaborating projects, etc.).

4 Architecture of Data Marketplace

As outlined in the Introduction of this deliverable, the Data Marketplace will be a public web-based environment with various different APIs. It will also be considered as a unified and standalone platform, able to store several types of assets that may derive/result from the separate procedures and mechanisms that are implemented in the scope of the project.

From the architecture perspective, the Data Marketplace will be structured around two (2) core services, the back-end and the front-end. This is basically the reason why the Data Marketplace is considered as a unified platform (i.e. unified platform of these two services). This separation contributes towards the platform's enhancement in terms of functionality (e.g. reduce maintenance costs, facilitate its management, etc.), also providing additional information and capabilities (e.g. enables direct access to the stored assets through the back-end).

Generally, the Data Marketplace will provide several functionalities that are mapped to different layers. The back-end includes three layers (i.e. Assets Storage Layer, Assets Management Layer, and Interaction Layer), while the front-end includes the fourth layer of the Data Marketplace (i.e. Presentation Layer). The Data Marketplace in full is consisted of all of these four different layers (as depicted in Figure 5 and Figure 6) that realize the expected capabilities as presented below, and further discussed in greater detail in the next sections.

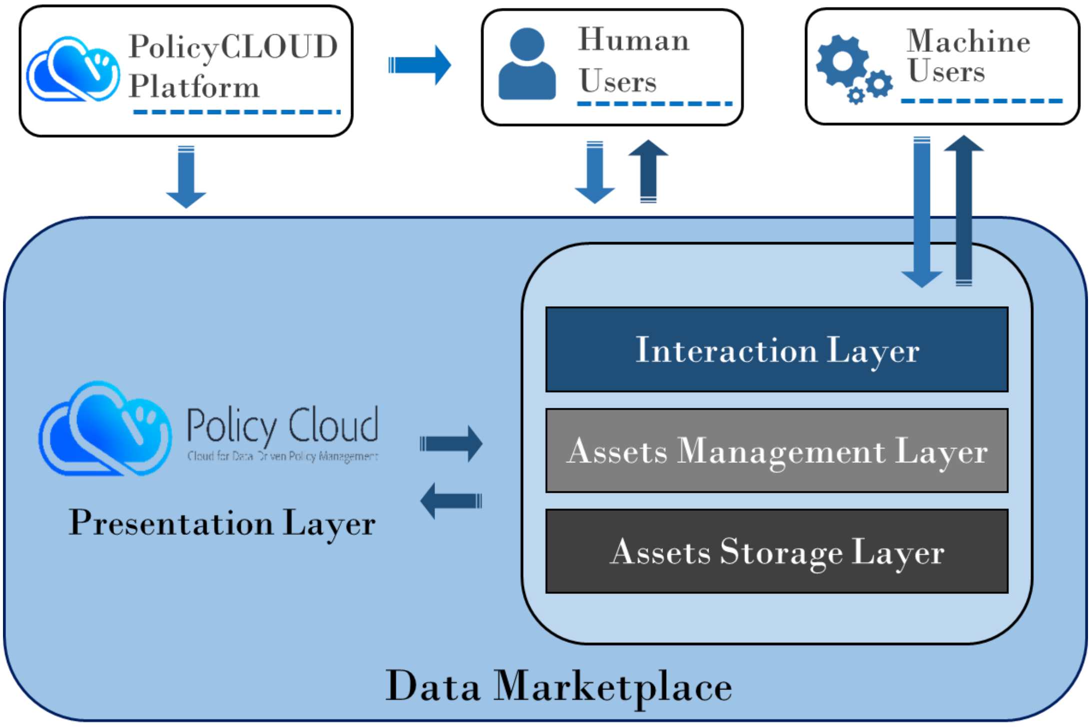


FIGURE 5 - DATA MARKETPLACE ARCHITECTURE.

- The Assets Storage Layer (part of the back-end) is the layer in which the platform's offered assets are stored.
- The Assets Management Layer (part of the back-end) delivers all the needed principles and techniques for the management of the Marketplace's assets.
- The Interaction Layer (part of the back-end) supports the communication between the Data Marketplace and its users (i.e. human users, and machine users), by providing discrete APIs for exploiting each different type of asset.
- The Presentation Layer (part of the front-end) provides the User Interface towards the different types of users that are willing to use the platform.

As described in Section 3.4, the Data Marketplace will support access to its offerings to both human users (i.e. policy makers, developers, scientists, interested third parties) and machine users (through the respective interfaces). In this context, the human users will be able to interact with the Marketplace through the front-end (through the Presentation layer) that reflects a user-friendly platform (providing the UI), while other additional services (e.g. project's services, 3rd parties) may interact directly with the back-end (through the Interaction layer). This separation is also represented in Figure 5.

In terms of storage, as described in Section 3.3, the Data Marketplace will store various types of assets (objects) in any format, and furthermore, relevant functionalities will be developed for the assets' management. In addition to these, the platform will consider issues related to the security of the assets it stores and offers. It will also consider, comply, and inform the end-users on privacy issues (i.e. GDPR). Hence, user registration is a required functionality to ensure that all the regulations are enforced. Thus, the providers of the assets, among other things, will be able to set rules on which end-users will be able to retrieve their solutions, etc.

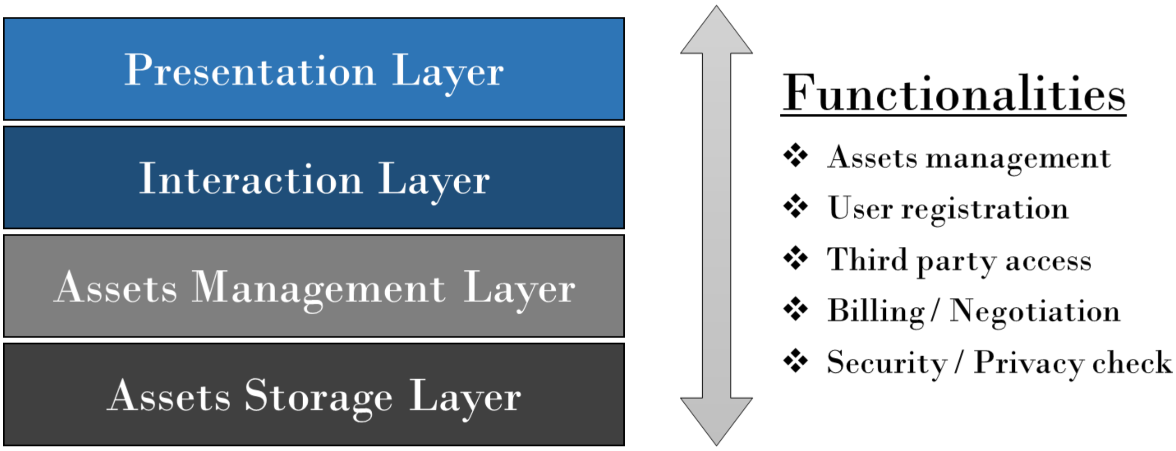


FIGURE 6 - DATA MARKETPLACE LAYERS AND MAIN FUNCTIONALITIES.

4.1 Back-end

The back-end is the core base of the Data Marketplace and it will be developed using a variety of components/tools. All of its sub-components will be containerized in Docker images [4] that, among others, offer more efficient management and maintenance, enabling continuous updates and integration. Python [5] will be used as the programming language that along with the Flask [6] framework, which is a Web Server Gateway Interface (WSGI) developed in Python, implements RESTful APIs to handle the respective HTTP requests.

The offered assets will be stored in a MongoDB [7] database that will be used in combination with GridFS [8] specification for storing and retrieving large files/objects, of any format. Moreover, Gunicorn [9], a Python WSGI HTTP Server for UNIX, will be utilized with NGINX [10], an open-source high-performance HTTP web server and reverse proxy, since Flask is not optimum for production mode, and thus, both tools will extend the Flask framework in order to enable access to multiple users at the same time.

The following subsections analyze and describe the functionalities of the back-end's three layers (i.e. Assets Storage Layer, Assets Management Layer, and Interaction Layer).

4.1.1 Assets Storage Layer

The Assets Storage Layer is responsible for storing the assets/objects that will be offered by the Data Marketplace (Figure 7). An essential component of it, is a database that can store files in any format as well as additional information about the files provided. For this purpose, the appropriate database type is NoSQL technology. NoSQL databases are non-tabular and store data in a different way than relational databases do. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph databases. They provide flexible schemas and scale easily with large amounts of data and high user loads.

Based on the above, a NoSQL database will be used for the Data Marketplace and in particular, a document-based database so that it can store files. One of the most performant and efficient document databases is MongoDB. MongoDB stores data in flexible, JSON-like documents, meaning fields that can vary from document to document and data structure that can be changed over time.

In addition, GridFS, a file system/specification for storing and retrieving files that exceed the maximum size limit that MongoDB sets, will be used to store files and assets divided into parts, called chunks, storing each chunk as a separate document. The documents will be stored with their binary data in two collections that contain their chunks and metadata, respectively.

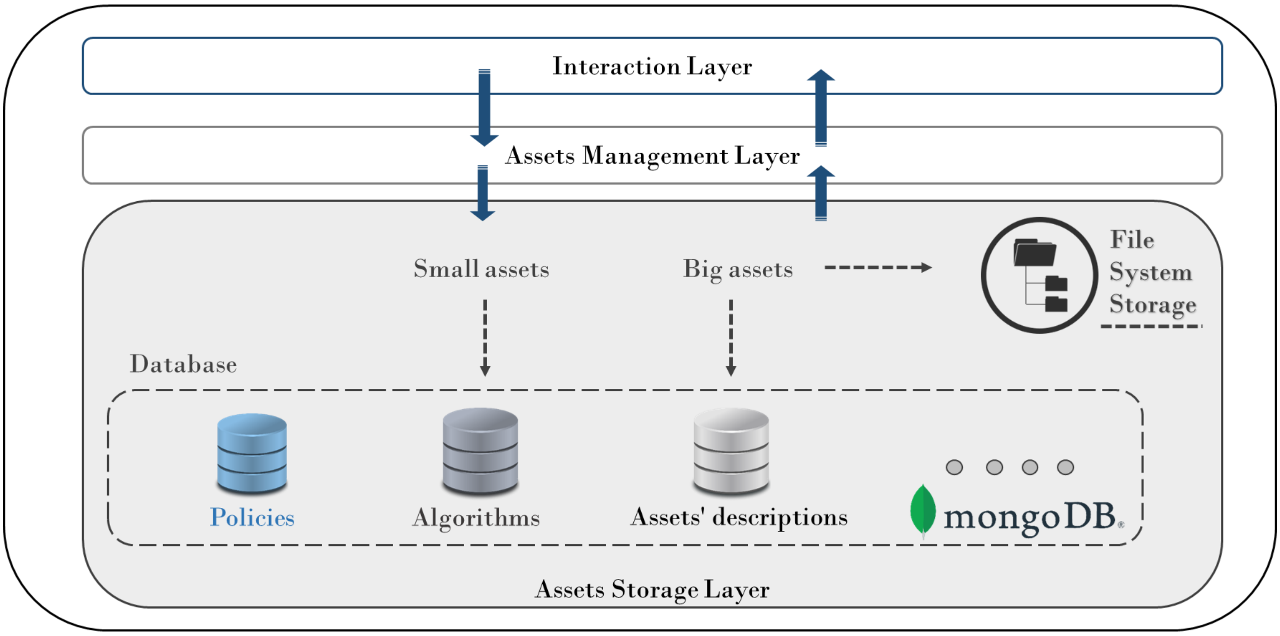


FIGURE 7 - ASSETS STORAGE LAYER.

Thus, for this layer of the Data Marketplace, the combination of MongoDB and GridFS will be used to store both small and large files. Appropriate collections will be created in MongoDB per destination branch/field of application of the files (i.e. a collection for tools, another collection for policies, etc.).

The information provided for the offered assets by their providers, called descriptions (described in Section 4.1.2), will also be stored in MongoDB in JSON format, which is fully supported by MongoDB, as mentioned above. Moreover, JSON files are easy to be created, retrieved and read even by a simple user, although in the Presentation Layer (front-end), the containing information will be presented to the end-users in an even more friendly way. JSON files also extend the interoperability of systems, especially in this case, the back-end and the front-end.

4.1.2 Assets Management Layer

The Assets Management Layer is responsible for the entire life cycle of the data and assets within the platform and offers all the principles and techniques for their management (Figure 8). More specifically, the Assets Management Layer is the layer that handles the assets from the moment they are entered to the platform through the APIs and then stored in the database (in the Assets Storage Layer), until their final deletion from the platform.

In general, through this layer, the Marketplace will support the CRUD operations. CRUD stands for "Create", "Read" ("Retrieve" in this case), "Update" and "Delete" that are the four basic functions of persistent storage. on top of these operations, the platform will also support the searching functionality, as described below:

- Create functionality is the functionality where new assets/objects are ingested in the Data Marketplace. This operation is triggered following the upload of a new asset by a provider and it results to the creation of appropriate entries in the database for the new document (the document in chunks, its metadata, etc.). An important information for each object refers to a unique alphanumeric identifier through which the indexing can be done as well as its subsequent retrieval.

- Search functionality is the functionality that will enable the users to search for assets, based on various parameters from metadata given by the providers or possibly other information generated by the system during its creation (e.g. unique identifier, number of downloads, etc.).
- Retrieve functionality is the functionality that will be executed when a user wants to download an asset from the Data Marketplace, after first seeing/finding it through the search functionality. The retrieval will result to obtaining the corresponding asset from the Storage Layer and delivering it to the appropriate API.
- Update functionality is the functionality that will be triggered when a user replaces an asset with a newer version of it or modifies some of its metadata.
- Delete functionality is the functionality that will handle the deletion of the assets from the platform when the owners, or other users with the appropriate rights, decide to delete them.

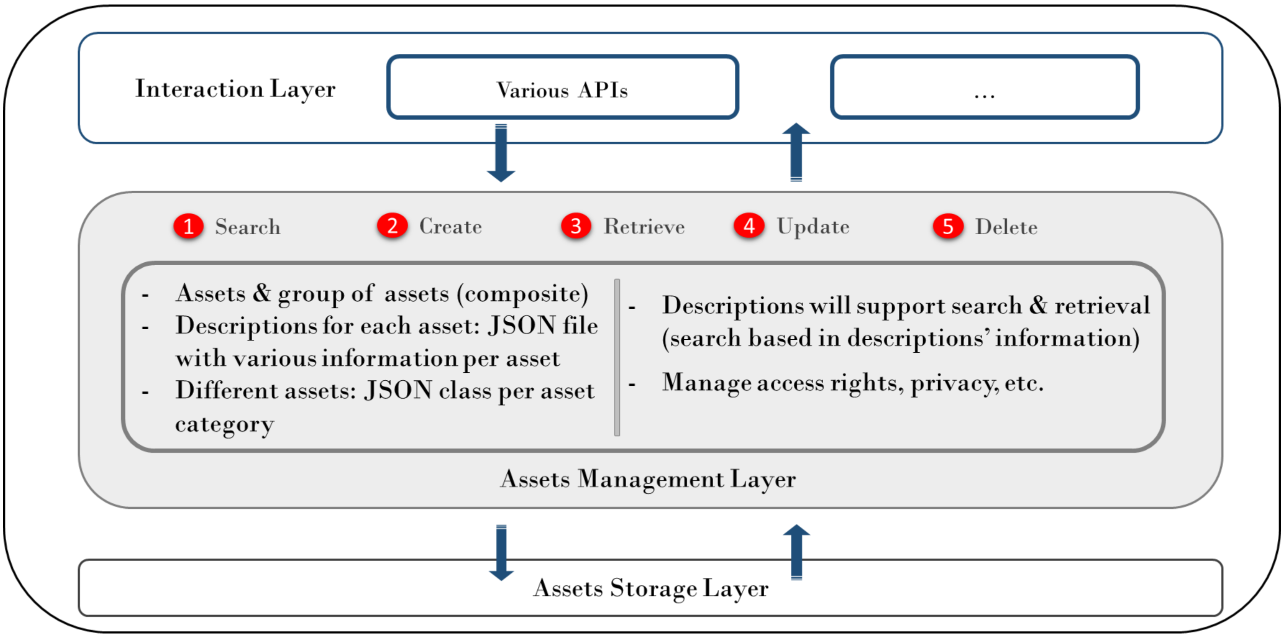

Interaction Layer
Various APIs
...
1 Search
2 Create
3 Retrieve
4 Update
5 Delete
- Assets & group of assets (composite)
- Descriptions for each asset: JSON file with various information per asset
- Different assets: JSON class per asset category
- Descriptions will support search & retrieval (search based in descriptions' information)
- Manage access rights, privacy, etc.
Assets Management Layer
Assets Storage Layer

FIGURE 8 - ASSETS MANAGEMENT LAYER.

The main part of the Marketplace, i.e. all the layers except the Presentation Layer (because of its separation from the back-end), will be managed by RESTful APIs. A RESTful API is an Application Program Interface that uses HTTP requests to GET, PUT, POST and DELETE data/assets. The latter indicate some widely used HTTP methods that are related to the previous Search and CRUD operations/functionalities. Table 4 matches these HTTP methods with the above supported functionalities and the corresponding statements in MongoDB (the database that will be used).

Operation	HTTP Method	MongoDB
Create	POST / PUT	Insert
Retrieve	GET	Find
Update	PUT / POST	Update
Delete	DELETE	Remove
Search	GET	Find

TABLE 4 - MATCHING (S)CRUD OPERATIONS WITH HTTP METHODS AND MONGODB STATEMENTS.

An important feature of the Data Marketplace is the search functionality. When a provider uploads a new asset to the Marketplace, he/she will be prompted to provide additional information about the asset (i.e. metadata). This information could be related to the usability of the file, the type of the file/asset, its input parameters, as well as the exported results if it is an algorithm, who has the right to retrieve it, other useful comments, etc.

This set of metadata related to an asset will be the content of a JSON file called "description file" of the asset. Each asset will have a description file so that it can be searched and retrieved by the end-users, who will be able to search for files according to this information (i.e. metadata). Therefore, the submission of these description files per asset is necessary because otherwise assets' existence on the platform will not make sense and they will probably not be able to be retrieved.

It will also be possible to obtain the assets' information and automatically convert them to JSON files through the front-end, but in cases where the users interact directly with the APIs of the Interaction Layer (e.g. the machine users/other services, etc.), they will need to create on their own an appropriate description file. Based on the above, as the offered assets vary and since different information may be needed for each type of offered asset, templates with minimum required fields per asset type/class will be created. A representative example of a description file for K-means algorithm in JSON format is cited below:

```
{
   "id": "afhi4df-hj6sf57-43fjd", owner": "sample user",
   "properties":{ type": "ML Algorithm", "name": "Kmeans", task": "Clustering" },
   "executable_details":{
        "filename": "kmeans.py", runtime_environment": "Python",
        "description": "environment and libraries required for execution"
   },
   "input_parameters": {…}, comments": […],
   "other_details":{ author":"sample user", created_at":"14.06.2021",
        "last_updated_by":"sample user", "updated_at":"25.06.2021"
   }
}
```

4.1.3 Interaction Layer

The Interaction Layer has to do with the different aspects and interfaces of the RESTful API that will handle the back-end's operations. More specifically, it will offer the envisioned interfaces that will connect the outside world with the platform. The Data Marketplace will have different APIs that will allow all the users of the system (i.e. providers, end-users, human and machine users) to interact with the platform.

In short, appropriate APIs will be developed that will receive HTTP requests and through them will trigger the platform's CRUD operations. There will be APIs for handling the descriptions of the assets and APIs that will handle the lifecycle of the assets. For example, there will be APIs that will enable users to search the description files and/or metadata stored in the database to find information that fits their needs and there will be APIs that will enable the uploading of the assets. These APIs will be common for all types of users of the Marketplace, but depending on the needs, there may be some additional APIs that will enable other platforms/systems to interoperate with the Marketplace with a different approach. One example of this is the interoperation of the front-end and the back-end. In general, the back-end will be able to respond to HTTP requests with exchanged information in JSON format and only (binary data for assets).

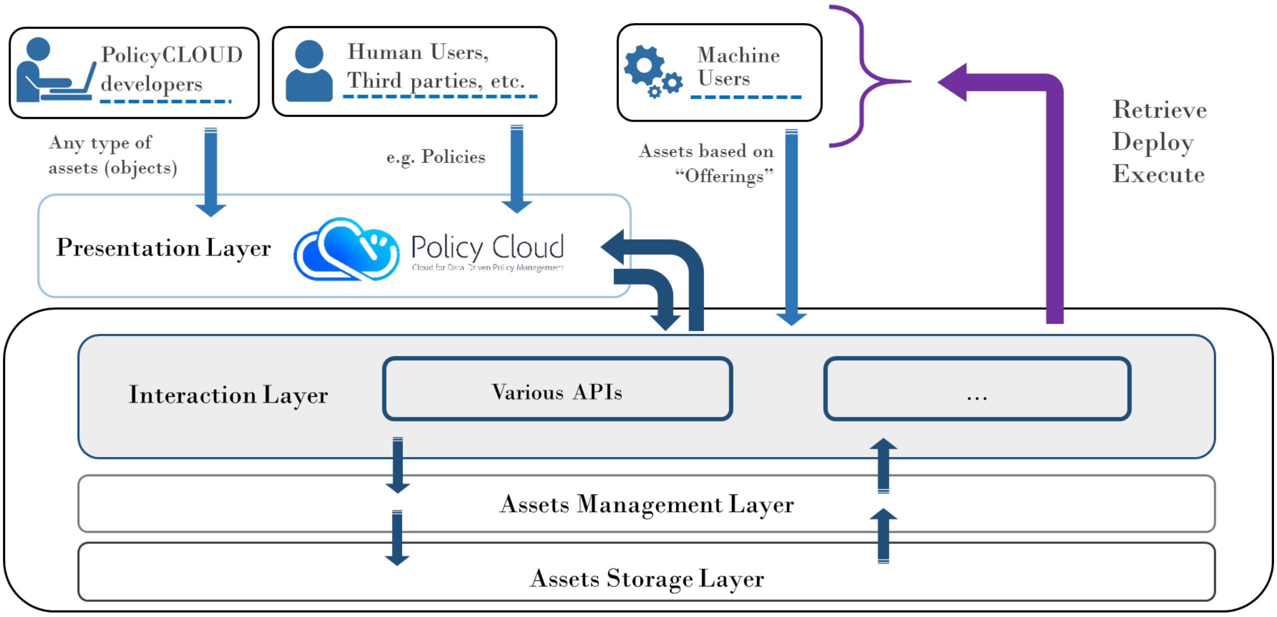


FIGURE 9 - INTERACTION LAYER.

Moreover, there will be different APIs per asset type, in the same way as with the collections of the database (as described in Section 4.1.1), and through appropriate APIs, the users of the Marketplace will be able to authorize themselves to the platform. Finally, as described above, the back-end will provide direct access to its interfaces for all types of users (especially for machine users/services). Based on that and the specifications of the back-end and the front-end, the Marketplace will have two domain names to be accessed, one for the back-end (e.g. https://db.policycloud.eu) and one for the front-end (e.g. https://marketplace.policycloud.eu).

4.2 Front-end

The front-end is the fourth layer of the Data Marketplace and it will be a web-based server that will present the offered assets to the users, with a friendly UI. It will be implemented using various web technologies (HTML, CSS, etc.) and it will become functional using PHP and JavaScript. It will also exploit WordPress [11] and various plugins of it, in order to manage the content that will be presented.

In general, the front-end will convert all interfaces of the back-end (RESTful API) into user friendly interfaces and provide automated forms and processes that will make it easier for users to interact with the back-end and benefit from its stored assets. Therefore, it will act as an intermediate among the users and the back-end that will send the respective HTTP requests to the latter and present its responses.

Figure 10 depicts the interactions between the front-end and the back-end and the additional components of the overall Data Marketplace.

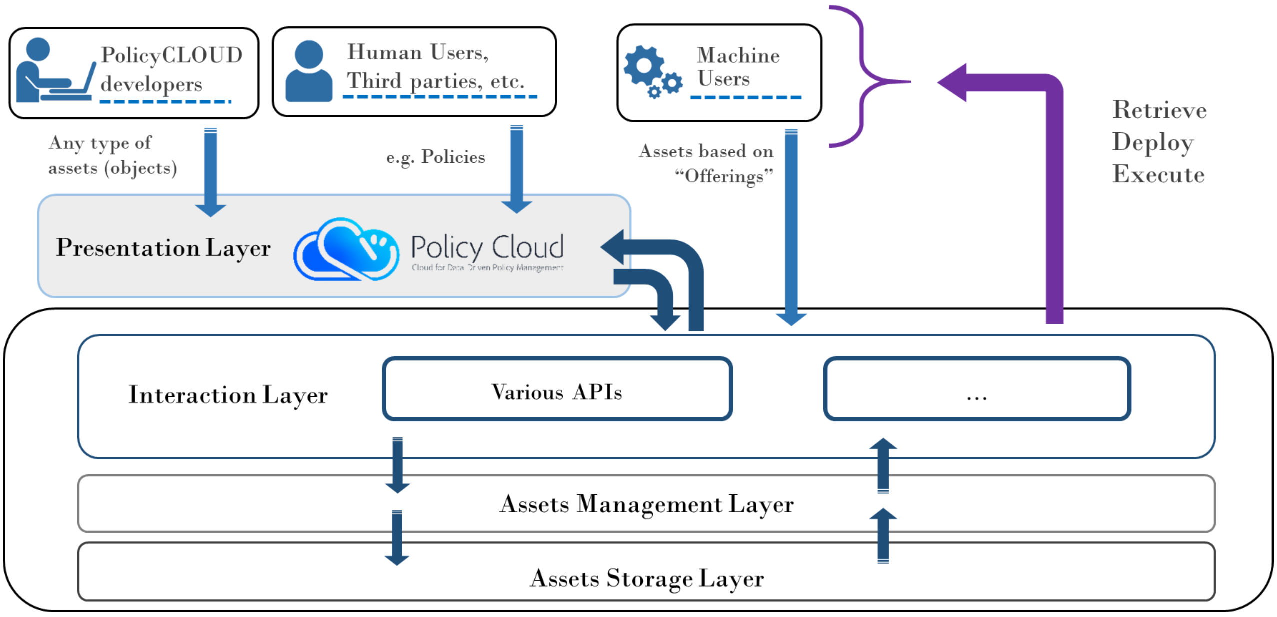


FIGURE 10 - PRESENTATION LAYER.

In short, the front-end will allow users to register and log-in to the Data Marketplace (user-based platform), upload their offered assets by filling out an appropriate form whose fields will be the content of the description files of the assets (as mentioned in Section 4.1.2); search for assets according to various fields (title, asset's type, fields of use, provider, other metadata, etc.) that can be further filtered or even sorted by the number of views or the date they were uploaded to the Marketplace, etc. Also, there will be a page that will present in detail the information of the assets, and through this page, the users will be able to retrieve the assets (if they have the right depending on the respective access rules set by the provider).

4.3 Overall Conceptual View

Following the description of the individual elements, the overall conceptual view of the Data Marketplace is presented in the Figure 11, which depicts all the platform's layers along with their key offered functionalities, the providers and end-users.

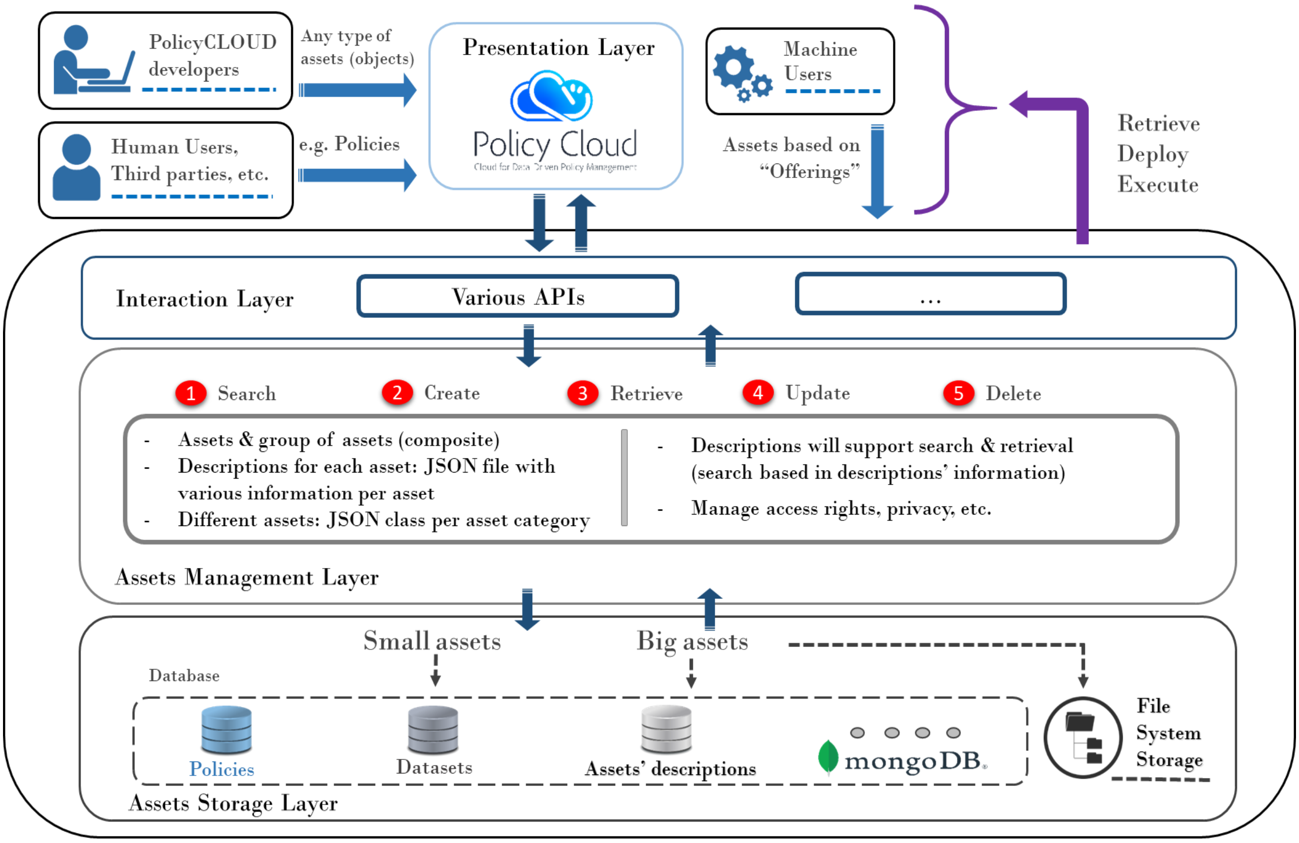

PolicyCLOUD developers
Any type of assets (objects)
Human Users, Third parties, etc.
e.g. Policies
Presentation Layer
Policy Cloud
Cloud for Data-Driven Policy Management
Machine Users
Assets based on "Offerings"
Retrieve
Deploy
Execute
Interaction Layer
Various APIs
...
1 Search
2 Create
3 Retrieve
4 Update
5 Delete
- Assets & group of assets (composite)
- Descriptions for each asset: JSON file with various information per asset
- Different assets: JSON class per asset category
- Descriptions will support search & retrieval (search based in descriptions' information)
- Manage access rights, privacy, etc.
Assets Management Layer
Small assets
Big assets
Database
Policies
Datasets
Assets' descriptions
mongoDB.
File System Storage
Assets Storage Layer

FIGURE 11 - OVERALL CONCEPTUAL VIEW.

The overall information flows are depicted in the figure through the respective arrows that represent the main interactions:

- The users of the Data Marketplace will interact with it through the front-end (Presentation Layer), from which HTTP requests will be sent to the back-end platform (requests depending on the case: search, upload / store new assets, retrieve, update or delete an existing asset).
- These HTTP requests will be received by the corresponding APIs of the Interaction Layer. Users will be able to interact directly with the APIs, sending the HTTP requests by themselves (using an appropriate tool).
- After receiving the requests, the Assets Management Layer will undertake the processing of the requests, using the developed functionalities. Specifically, it will interact with Assets Storage Layer to retrieve useful information that, after processing, will be sent to the users via APIs, in response to their HTTP requests.

- When a provider intends to upload a new asset, the provider should also submit the asset description file (via front-end it is generated automatically by filling in the appropriate fields), which will contain metadata for the asset. Both the asset and the description will be stored in theAssets Storage Layer, the descriptions in the MongoDB and the assets in the GridFS specification.
- In order to retrieve the assets, the Assets Management Layer, through the retrieve functionality, will find the requested asset from the database and will deliver it to the end-user via the platform’s APIs. The retrieval of the assets will be also done in a similar way by other systems / services that will be linked to the Data Marketplace.

5 Conclusion

This document reported the work that has been currently done in the scope of task T7.2 Data Marketplace, whose main objective is to provide the initial design and specifications of the project's Data Marketplace and develop it.

The specifications that have been introduced in this deliverable will be utilized for the realization and implementation of the platform, encompassing its main functionalities regarding the storage of assets, the advanced search and retrieval, and their combination into unique turn-key offerings. The latter will be demonstrated through the upcoming deliverable D7.5 - Data Marketplace: Software Prototype that will be realesed in M22 (October 2021) of the project.

Overall, the Data Marketplace will integrate ready-to-use solutions and assets of the project, including policies, tools, algorithms, datasets to feed them as well as various project's outcomes (e.g., presentations, tutorials, etc.). In general, the Data Marketplace will be open to everyone who wants to join its community and benefit from its offerings. The community created around the Marketplace will consist of experts (e.g., data scientists, developers, policymakers, etc.) who may offer such solutions and every interested third party who wants to achieve any type of objectives in respective fields of use.

In conclusion, the Data Marketplace will be developed based on these specifications and populated with such assets. In M32 (August 2022) is planned to be released an updated version of the specifications (D7.11 - Data Marketplace: Design and Open Specification) following the initial implementation and provision of the Marketplace.

References

[1] Acumos, Homepage, https://marketplace.acumos.org, retrieved 2021-05-28

[2] 5GTANGO, Homepage, https://www.5gtango.eu, retrieved 2021-05-28

[3] FINSEC, Solutions, https://finsecurity.eu/solutions, retrieved 2021-05-28

[4] Docker, Homepage, https://www.docker.com, retrieved 2021-06-25

[5] Python, Homepage, https://www.python.org, retrieved 2021-06-25

[6] The Pallets Projects, Flask, https://palletsprojects.com/p/flask, retrieved 2021-06-25

[7] MongoDB, Homepage, https://www.mongodb.com, retrieved 2021-06-25

[8] MongoDB, GridFS, https://docs.mongodb.com/manual/core/gridfs, retrieved 2021-06-25

[9] Gunicorn, Homepage, https://gunicorn.org, retrieved 2021-06-25

[10] NGINX, Homepage, https://www.nginx.com, retrieved 2021-06-25

[11] WordPress, Homepage, https://wordpress.com, retrieved 2021-06-25