SHared automation Operating models for Worldwide adoption

SHOW

Grant Agreement Number: 875530

D6.1: SHOW Marketplace and services – first version

This report is part of a project that has received funding by the European Union's Horizon 2020 research and innovation programme under Grant Agreement number 875530

Legal Disclaimer

This report is subject to a disclaimer and copyright. This report has been carried out under a contract awarded by the European Commission, contract number: 875530. The content of this publication is the sole responsibility of the SHOW project.

Executive Summary

The following work is related to ***Work Package 6: Services Marketplace*** for the SHOW project. Hence, this deliverable describes the work held in the five activities included in WP6, namely: *A6.1: SHOW Marketplace*, *A6.2: Metadata-based Value Added Services*, *A6.3: SHOW Operational services*, *A6.4: Energy Management services* and *A6.5 Dynamic Personalised Services*. The purpose of this deliverable is to provide details regarding the SHOW Marketplace in terms of requirements, architecture, infrastructure, design, and development along with the services that will be hosted by the platform. This deliverable is related to the first version of the SHOW Services Marketplace, whereas the second version will follow at month 30 and the final version at month 40.

The marketplace objective is to act as a one-stop place encouraging participation from the Connected, cooperative & Automated Mobility (CCAM) community in order to accelerate awareness and offering of innovative CCAM technologies and services.

The purpose of this document is to describe the Marketplace platform, the functionality it provides and the technologies used to support this functionality. It also includes the business logic with all the other subcomponents of SHOW pilot sites but also mock-ups to showcase the actual implementation. The result of the platform is the ICT tools developed to support the needs of the CCAM community and the consumers. This deliverable summarises the efforts accomplished so far and outlines insights gained throughout the process as well as conclusions from these insights. Thus, the document is structured in different sections to describe the main technical aspect and all the concepts for the marketplace platform. In detail, Chapter 1 includes the introduction to the document, and interrelations to the other SHOW components, while Chapter 2 gives an overview of the research which was conducted to understand the related work. The next chapter 3 describes the business logic of the marketplace including stakeholders and their identities, products items, the requirements to satisfy, and use cases on some marketplace activities. Chapters 4, 5 and 6 elaborate on the available marketplace open-source platforms and the selected architectural design with its overall components but also summarise the design and development services included in the first version of the marketplace. Then, Chapter 7, briefly describes the first list of product items-services that will be hosted in the marketplace. At the time being, considering also the technological maturity of the marketplace, it is obvious that only product items that are developed within SHOW (or developed by partners of SHOW) will be hosted. Finally, Chapter 8 gives information and recommendations on the next steps towards the next version of the marketplace.

Document Control Sheet

Start date of project:	01 January 2020
Duration:	48 months
SHOW Del. ID & Title:	D6.1: SHOW Marketplace and services – first version
Dissemination level:	PU
Relevant Activities:	A6.1: SHOW marketplace A6.2: Metadata based Value Added Services A6.3: SHOW Operational services A6.4: Energy Management services A6.5: Dynamic Personalised Services
Work package:	WP6: Services Marketplace
Lead authors:	Konstantinos Giapantzis, Evangelos Antypas, Alexandros Papadopoulos, Iordanis Papoutsoglou, Athanasios Sersemis, Georgios Spanos, Antonios Lalas, Konstantinos Votis (CERTH/ITI)
Other authors involved:	Thomas Wiesmueller (AVL/DE), Pavlos Spanidis (CERTH/HIT), Roberto Carroccia, Davide Shingo Usami, Sevket Oguz Kagan Capkin (CTLup), Alberto Bellini (SWARCO)
Internal Reviewers:	Coralie Renard, Nicolas Morael, Mihai Chirca (Transdev), Nikolaos Tsampieris (ERTICO), Léon CHAUVET (Keolis), Maria Gkemou (CERTH/HIT)
External Reviewers:	–
Actual submission date:	09/03/2022
Status:	Final
File Name:	SHOW_D6.1_SHOW Marketplace and services - first version_v2.0

Document Revision History

Version	Date	Reason	Editor
0.1	09/07/2021	ToC proposed	CERTH-ITI
0.2	09/23/2021	ToC changed	CERTH-ITI
0.3	10/08/2021	Added chapters 3rd to 8th	CERTH-ITI
0.4	11/12/2021	First draft completed	CERTH-ITI
0.5	11/22/2021	Second draft completed	CERTH-ITI
1.0	11/24/2021	Full version provided for peer review	CERTH-ITI
1.9	03/02/2022	Peer reviewed version sent for submission	CERTH-ITI
2.0	09/03/2022	Proof-check and submitted version	ERTICO, CERTH/HIT

Table of Contents

List of Tables

List of Figures

Abbreviation List

Abbreviation	Definition
API	Application Programming Interface
AV	Autonomous Vehicle
B2B	Business to Business
B2C	Business to Customers
GNN	Graph Neural Networks
CCAM	Connected Cooperative & Automated Mobility
CCAV	Centre for Connected and Autonomous Vehicles
C-ITS	Cooperative Intelligent Transport System
CVIM	Common Vehicle Information Model
DARP	Dial-A-Ride Problem
DBMS	Database Management System
DDM	Decentralised Data Marketplaces
DRT	Demand Responsive Transport
DRY	Don’t Repeat Yourself
EC	European Commission
ETA	Estimated Time of Arrival
I3	Intelligent IoT Integrator
IoT	Internet of Things
IPFS	Inter Planetary File System
LaaS	Logistics as a Service
MaaS	Mobility as a Service
MAE	Mean Absolute Error
MQTT	Message Queuing Telemetry Transport
MVC	Model View Controller
OCF	Open Connectivity Foundation
OEMs	Original Equipment Manufacturers
PCA	Principal Components Analysis
PTOs	Public Transport Operators
REST	REpresentational State Transfer
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
SDKs	Software Development Kits
SOA	Service Oriented Architecture
SSL	Secure Sockets Layer
TCR	Token Curated Registry
TTG	Time To Green
VRP	Vehicle Routing Problem
VRPTM	Vehicle Routing Problem With Time Window
WCAG	Web Content Accessibility Guidelines

1 Introduction

1.1 Purpose and structure of the document

The scope of this document is to provide information on the services offered in the SHOW Marketplace version 1.0, as released. So, the main purpose of this document is to present the work performed in the context of Work Package 6: Services Marketplaces and consists of the software and the infrastructure developed for the SHOW project marketplace, its business logic, including stakeholders of the marketplace (plethora of different kind of users from Public Transportation Operators and Service Providers to Research and Academia Institutions) and requirements as well as the content ("products") provided through it. The structure of this document is as follows.

In the next chapter, the methodological approach is presented, while in the third chapter the literature related to marketplaces is described. Moreover, the fourth chapter provides insights with respect to the business logic of the present marketplace such as the related stakeholders, the available products, the requirements and the use cases. Additionally, in the fifth chapter, there is the analysis of the available open-source platforms for marketplaces that concludes to the selection of the most appropriate for the SHOW marketplace. Furthermore, the sixth chapter constitutes the technical documentation of the marketplace presenting the design and development activities (A6.1 & A6.2) until now. Moreover, in the seventh chapter the available product items are described, emphasising the services (operational (A6.3), energy management (A6.4) and personalised dynamic (A6.5) of the marketplace. Finally, Chapter 8 summarises the conclusions of the deliverable and presents the corresponding outlook.

1.2 Intended Audience

- Technical team of SHOW demo sites, OEMs responsible for the CCAV deployment in demo sites, service designers and technical teams
- Evaluation team of WP9
- Business stakeholders, academic and research community outside SHOW in the domain of CCAV

1.3 Interrelations

WP6 has many interrelations with other items and outcomes of the SHOW project. First of all, from WP1, it takes the input related to the general understanding of the project and more specifically the system needs (A1.1) and the use cases (A1.3).

Moreover, WP6 consists of the activities of the SHOW Marketplace and services, which is, in turn, a software product and, as with all software products, it should comply with very specific ethical and legal issues. In this context, this work is interrelated with WP3.

Furthermore, due to the fact that WP4 describes the architectural concept and the required tools of the project, there is an inevitable connection between the two work packages and more specifically, WP4 defines the position of SHOW marketplace to the whole system.

Finally, WP6 is related to WP5, because the services and algorithms developed in WP5 (Activities A5.2 & A5.3) will be hosted also to the marketplace, but also datasets that constitute one of the available product items could be stored to the SHOW Data Management Portal (A5.1).

2 Methodological Approach

The current work describes the activities composing *WP6: SHOW Marketplace and services*, which are in essence a) the theoretical concept and related work analyses, b) the software engineering procedures (analysis, design and development), c) the development operations and d) the product items that will be hosted in the SHOW marketplace.

First of all, a State-of-the-Art analysis was performed to explore the research area of the marketplaces and find the optimal solutions dominating the respective field. Hence, this analysis focused on three main axes: in the services marketplaces, in the marketplaces existing in European projects and business applications as well as in the different architectural approaches.

Regarding software engineering procedures, a combination of traditional methodologies in the field [1] with agile methodologies [2] is followed. Agile software engineering [3] is characterised by the fact that changes are expected to appear very frequently and in many phases of the software development procedure. For this reason, a human-centric approach is adopted in agile software development, emphasising the user satisfaction instead of the typical software engineering procedures. Considering the aforementioned and with respect to the adopted approach from the project, although the software phases are in accordance with the principles of traditional software engineering, an iterative approach is also considered in all the procedures composing the software life-cycle.

Therefore, the requirements elicitation, the architecture and the development consist of many cycles in order to reachthe final version. As an instance, for the SHOW marketplace architecture, three versions were created for reviewing and incorporating comments, with the third one being the final. Similarly, the User Interface of the marketplace was redesigned to be more user-friendly and increase user satisfaction and experience.

Regarding the requirement engineering process, the protocol IETF RFC 2119 [4] has been taken into account in the definition of the terms. According to that protocol, the following terms are utilised in the requirements definitions to characterise importance.

- **Must** denotes that a requirement is critical for the implementation of the SHOW Marketplace platform success. If the respective requirement is not included, the entire functionality of SHOW Marketplace will deteriorate.
- **May** denotes that a requirement is optional for the implementation of SHOW Marketplace platform. However, the respective requirement remains highly desirable for the SHOW Marketplace to constitute a state-of-the-art component for the AV domain.

Furthermore, another aspect of the adopted methodological approach, in accordance with the agile software development, is the use of the user stories [5] and the Scrum methodology [6] for the user requirements description and definition. This strategy is an intuitive tool that follows the agile software development, in order the simple presentation of SHOW Marketplace requirements to be feasible and easily readable/understandable. During the procedure of the requirement definition, previous analyses has been studied, mainly relevant to the Volere model as it has been described in [7].

Furthermore, with respect to the selection of the software platform, which would be the basis for the development of SHOW Marketplace, a state of the art analysis of the

corresponding open source platforms has been conducted. In order to make the final choice, different factors were taken into consideration, such as the available features of each platform, the community that supports the product and the skills of the SHOW development team.

Moreover, for the development operations and generally, the infrastructure of the SHOW marketplace, the main target is to comply with the concept of agile methodology that supports the Continuous Integration & Continuous Delivery [8]. According to this strategic decision, the aim of the SHOW marketplace development is to perform small changes to the code that can be easily integrated and afterward, the delivery of the software will be performed with an automated procedure. Furthermore, according to this agile development philosophy, the different environments that constitute the SHOW infrastructure, are updated automatically. Finally, with regard to the architectural approach, a service-oriented architecture [9] has been adopted in order for the offered services of the marketplace to be modularised and communicate by a corresponding protocol. Thus, the core values from the adoption of this decision can be summarised in the respective SOA manifesto [10]. According to SOA manifesto, more importance is given i to: business value, strategic goals, intrinsic interoperability, shared services, flexibility and evolutionary refinement. It is obvious from the aforementioned that the SOA approach is also in accordance with the agile methodology followed in SHOW Marketplace.

Finally, a common approach is required to populate the SHOW marketplace with a variety of services (operational, energy management and personalised dynamic services) that either will be developed during the corresponding activities or will be only hosted, but also for the rest of the product items that will be hosted in the SHOW marketplace. Thus, a specific data scheme has been formulated describing the mandatory and optional details that the owners/providers should provide in order to allow the hosting of their products to the SHOW marketplace. That way, the integration of the services could be performed as easily as possible and the required communications for this integration could be the minimum possible.

3 Related Work

3.1 Service Marketplaces

During the last years, several service marketplaces have been introduced in the corresponding literature. As an instance, in the work by HP [11], a services marketplace was envisioned as the way to replace B2C and point-to-point B2B connections with an approach that accommodates the many-to-many B2B model. The primary roles along with their responsibilities for such a marketplace were set to be the service provider and the buyer. The work introduced the five life-cycle phases in a marketplace that are the creation, discovery, contract negotiation, monitor, and fulfilment phase. In other words, Durante et al. [11] have described the way that a marketplace for services was working.

The definition for a service marketplace was given by Cerbo et al. [12] as they referred to the notion as a software marketplace. The marketplace was defined as a virtual place that allows software providers to advertise their apps and services to potential users. Also, the marketplace was a centralised distribution entity that could be accessed by multiple users.

Service marketplaces are evolving along with the emergence and progression of technologies. For instance, cloud adoption and the IoT devices popularisation have impacted the service marketplace ecosystem. Eltoweissy et al. [13] have included four pillars for the smart communities that are employed by the marketplaces, which are: sustainability, resilience, empathy-driven proactive intelligence and emergent behaviour.

The technology with a future impact in the marketplaces, especially the ones that handle data as a commodity, can be proved to be Blockchain. Blockchain is a technology that is an append-only ledger of transactions using cryptography. Sherman et al. [14] has researched the origins of Blockchain by pointing out the works by Chaum [15] and Haber et al. [16]. Blockchain has become popular due to the cryptocurrencies and especially due to the ground-breaking work by Nakamoto [17] for a decentralised money exchange system, Bitcoin. The Blockchain appliance in multiple sectors has grown since the smart contract introduction in 2014, exhibiting that it can become a tool which will permit the monetarisation of data and control of the transactions.

Blockchain can be an option for providing identity and proof of ownership along with tracking of every transaction. Intriguing examples to be researched in that area are the OCEAN protocol and the data marketplace by IOTA foundation. The whitepaper for the OCEAN's data marketplace [18] illustrates the architecture and defines the tokens to be created following ERC-20 standard. The IOTA foundation data marketplace is currently in a proof-of-concept phase and there are published academic works on this conception. Specifically, Musso et al. [19] have introduced PEGASUS that is a decentralised marketplace for Smart Cities based on the IOTA Tangle. Similarly, Lamtzidis et al. [20] proposed a distributed sensor node system that utilises IOTA protocol to exchange data.

3.2 Marketplaces in projects and business applications

3.2.1 European projects with marketplaces

Initial desk research has been performed to pinpoint other projects that developed a marketplace. For the desk research, the materials offered by EC's database for Horizon projects between 2014 and 2020 were utilised [21]. The performed desk

research on such data aimed to uncover mature projects that have delivered marketplace applications. The research's objectives were to document initiatives in the automotive sector or a place for applications in any business sector.

One of the first initiatives in the automotive sector is the AutoMat [22] project that concluded in 2018. The project aimed to offer a marketplace for data gathered for the various OEMs for cross-sectorial access. The marketplace architectural overview engaged vehicle users, OEMs, and service providers in a single place [23]. The users are incentivised to engage in the marketplace to monetise the available data.

Another project that can accommodate data in the automotive industry is named Cross-CPP [24]. While the marketplace generally concerns data and related services, the automotive sector is a candidate sector that may benefit from the platform's use. The project considers the role of the platform users and their data activity to divide into categories. For instance, platform users can be developers or users depending on their involvement [25].

The majority of initiatives in the automotive industry are concerning the creation of a marketplace for data. For this reason, the desk research was expanded to seek out projects that delivered marketplaces for applications. An intriguing project is the CLARITY project [26] which aims in creating a marketplace for e-government services. The marketplace used the open content management system, Drupal [27], for its conception. Other technical specifications were the use of MySQL and MariaDB in the Apache server as described in the project's deliverables.

A prominent project that created a marketplace for services in the IoT ecosystem is ACTIVAGE [28]. The marketplace has divided its users into three distinct categories; Users, Developers, and Administrators. Each category of users has access rights to their corresponding actions. The deployed applications are built on AIOTES and the marketplace can act as a one stop for finding apps. The marketplace's architecture consists of a front-end, back-end, and bridges to third parties marketplaces. The technology stack is described with the underlying technologies of the project like Django, JQuery, and HTML5 [29].

A marketplace application was designed during the GOAL project [31] that was to foster social interactions among the project's community. The website promotes e-health with its application and divides users into three categories; Achievers, Supporters, Developers [32]. The marketplace uses a Java Spring Boot application as a task manager, MongoDB and exposes a REST API [33].

One final project that deployed an AI Marketplace is the Bonseyes project [34]. Four separate categories are in place to divide the platform's users into Innovators, Researchers, Developers, and Companies. Each category has its objectives and goals that put effort into achieving. The deployed data marketplace micro-components have made use of technologies like Gateway, Postgres, Django, Nginx, and RabbitMQ.

The desk research has indicated that European projects in the automotive industry focus on the creation of a data marketplace. Essentially, their aim is solely to deploy a data pipeline from the basis, like sensors, to users. For that reason, the desk research was extended to marketplaces with services from projects in other sectors. The prevailing technologies for marketplaces were some of the results of the research. While various options were included for the marketplaces' database, several marketplaces used the Django framework. Another point for the desk research was the users and their roles in the marketplace. While some marketplaces divide users into three general categories of users, developers, and administrators, others have allocated identities that correspond to their audience.

3.2.2 Business applications of marketplaces

Both marketplaces and distinct applications can be found in the automotive and public mobility sector. The concepts behind both marketplaces and applications are not far from the academic point of view. The desk research for business deployed applications can help identify the place of SHOW's marketplace and the needs that the market may have for an application.

Some of the most prominent use cases for marketplaces in the automotive sector are related to data. Data is one of the most valuable resources during the digital age and the new industrial revolution. Data marketplaces aim to incentivise users to provide their data via a reward system and enhance OEMs capabilities as their decisions will be based on data. A notable example of a data marketplace is coming from Otonomo [36]. Privacy is a key issue for data and the company deploys solutions alongside the marketplace to handle privacy. In more detail, a Consent Management Hub and a Dynamic Blurring Engine are safeguarding data privacy by controlling the consent for the data use and applying anonymisation techniques respectively. The platform accommodates more than 100 service providers. Another data platform in the automotive sector is Caruso [37] which aggregates the data from multiple manufacturers into one place and allows customers to benefit from building solutions based on data from connected vehicles.

Individual applications for public mobility are prevalent cases in the desk research and only a few are to be presented. CityMapper [38] is a platform that permits users to search for the most efficient way to publicly commute in real time. The platform allows users to create their own SDKs and implement the same APIs as the platform. The final examples are in the smart mobility sector and can be described as Mobility-as-a-Service. The examples are the Beeline Singapore [39] and Optimile [40] that deployed mobility platforms. The case of Beeline Singapore is included in the report despite the foreclosure of the team's operations. The team has open-sourced the deployed service for public use in a blog post [41]. Finally, Optimile has deployed a marketplace for mobility services that address preferably B2B customers. The various applications like charge points, taxi service, and bike rental can help businesses expand and enhance their services by plugging into the marketplace.

Other sectors accommodate service marketplaces for developers to deploy their applications and for the public to use them. The most notable case that has to be mentioned is coming from the FIWARE Foundation and is named FIWARE marketplace [42]. The marketplace accommodates applications that are generally related to smart cities. Mobility is covered in the smart cities' applications since a couple of related applications are available in the marketplace. The applications in the marketplace are described in a problem statement, a description, and followed by relative tags. Developers can include documentation and references for their applications. Other marketplaces that have been considered were universAAL IoT [43] and OpenIoT [44].

The desk research on the business applications indicates some points of interest. The initial point is that the marketplace is a centralised application deployed by a single entity. The main purpose is commercial exploitation and excludes members of CCAM communities. CCAM communities can only interact with these marketplaces either as customers or developers deploying applications. Thus, there is room for an open marketplace that differentiates in products and user identities.

3.3 Architectural approaches

3.3.1 General Contemporary Marketplaces

There are several research works on innovative marketplace architecture in general. The paradigm of I3 is researched in various works. The growth of technologies is impacting the marketplace architecture as blockchain implementations are being developed.

Krishnamachari et al. [45] are proposing an innovative IoT marketplace to tackle the creation of separate silos developed by one entity involved in each phase of the creation and to alleviate the dependence on significant governmental capital expenditures. The suggested platform is called the Intelligent IoT Integrator (I3) and facilitates the movement of real-time streaming data between owners and third-party applications. I3 places a data-exchange middleware layer above the transport layer to allow device owners to contribute with their data. The data ownership of the data is mediated through an end-to-end agreements between the device owners and service developers. The marketplace would bestow the data owners with the right to decide a set of usage conditions on the data. The marketplace gives incentives to both device owners and developers to participate in the market. A simple proof of concept of an I3 domain was created by combining MQTT pub-sub broker on the backend with a Python Django based web marketplace front-end using MySQL database.

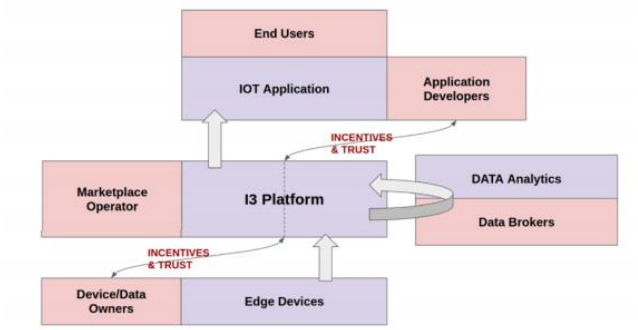


Figure 1: I3 Concept (Krishnamachari et al. [45])

An innovative architecture on a data marketplace for smart cities has been proposed by Ramachandran et al. [46], who explore the creation of a decentralised data marketplace using blockchain and distributed ledger technologies. In their work, the decentralised marketplace is composed of ten different key components. The basis for the marketplace is the sellers of data in a city and the buyers interested in the data. The next critical component is the creation of a method that connects buyers and sellers. A proposed method for that discovery component is the use of smart contract that the seller posting is a hashed pointer on blockchain to a description on an off-chain distributed file storage system. The third component of the marketplace is the meta-data organisation, where standards like SensorML or OCF could be applied to standardise the data format. The next component is focused on the data transfer and payments and is implementing a designated protocol to collect the data in exchange for payment such as the streaming data payment protocol (SDPP). Another component that could be integrated is a rating system that records the seller's quality of data and buyer's reliability based on their actual interaction on a particular transaction in a smart contract. A curation and recommendation component could provide value where a token curated registry (TCR) would be applied to preserve the quality of the data. The previous components, the TCR and ratings, would enhance the trust in buyers. Furthermore, access control and identity management are important as they can be a

component to guarantee the identity in a decentralised secure identity platform such as Civic. The security properties for the product, ratings and curation are inherited from blockchain as the systems are decentralised. Finally, the privacy of sensitive data is vital and could be achieved by two methods. The first method is to not log the data on an immutable ledger and the data transfer between buyers and sellers to be handled via encrypted data channels. The second method is that sensitive metadata that are stored on the blockchain, to be hashed pointers to actual data stored offchain. All in all, the conceived proof-of-concept is implementing the Ethereum blockchain, IPFS as a distributed file storage framework and SDPP as a streaming data protocol.

Figure 2: Decentralised data marketplace on smart cities (Ramachandran et al. [46])

Another blockchain based data marketplace proposal was BLockchainENabled Secure Microservices for Decentralised Data Marketplaces (BlendSM-DDM) by Xu et al. [47]. BlendSM-DDM is built as a decentralised security services architecture, which was inspired by blockchain and containerised micrcoservices. An oracle could be implemented for new participants' enrollment on the permissioned blockchain. The permissioned blockchain could be more advantageous than a public blockchain network as it could act as a lightweight consensus mechanism. The security services are blockchain-enabled and handle key elements such as Data Pub-sub Payment, Identity Verification, Data Integrity. Furthermore, the microservices are designed to be exposed to accept service requests by REST APIs. To sum up, BlendSM-DDM conception could ensure data exchanges and payments in DDM systems like the I3.

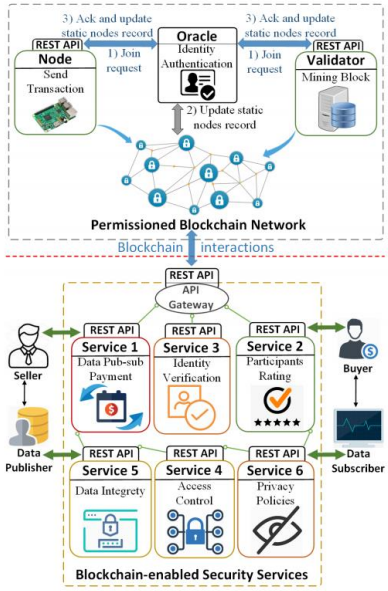


Figure 3: BlendSM-DDM architecture (Xu et al. [47])

3.3.2 Marketplace in CCAM

Extensive research is conducted on proposing a holistic service based on the transfer of data and availability in the CCAM sector. In the documented solutions, the architecture focuses on the data as one of the most valuable resources in the Big Data era. The architecture takes into account various aspects such as privacy and ownership of data that should be guaranteed and incorporates cyber security techniques to ensure. The marketplace is the final piece of the architecture, acting as a place to attract various stakeholders such as application developers, users, and manufacturers.

The project Automat has proposed a data model for accommodating data from various automotive manufacturers, as described in the works of Pillmann et al. [[23] and [48]]. The model, Common Vehicle Information Model (CVIM), is applied in the OEM backend to transform the data from brand-dependent into the standardised format. The marketplace is a mediator platform and offers access to the data allowing the flow of data from vehicle owners to service providers. The marketplace key features are the data exploration; the service development; the service execution; and the service termination. The features are exposed via REST interface that specialises in a machine-readable format for seamless integration with services, and a responsive cross-platform front-end for human interaction. The proposed architecture system is designed to enable the providers of services to access vehicle data via a single point of access. All in all, the marketplace is responsible for handling the vehicle fleet accumulation and preserving the privacy of stakeholders' data.

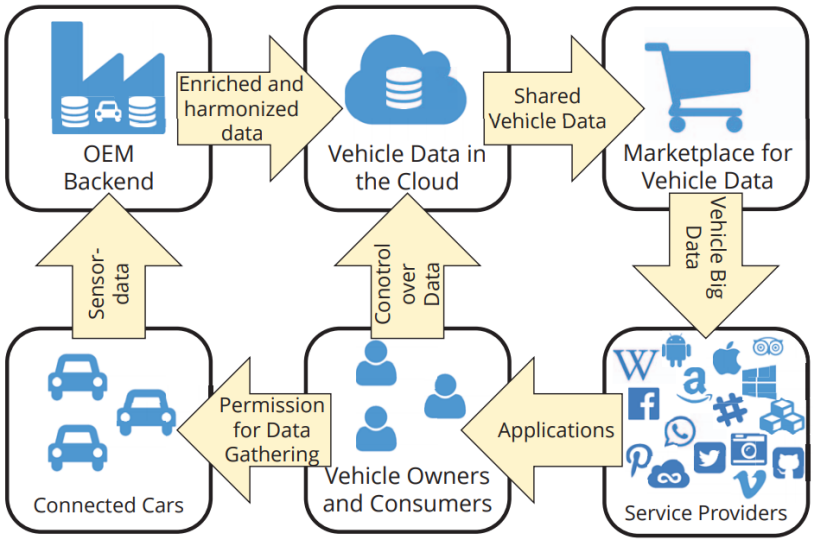


Figure 4: Vehicle big data architecture (Pillmann et al. [23])

3.3.3 Service Oriented Architecture (SOA)

The Service Oriented Architecture (SOA) has been researched to be applied in IoT applications. Butzin et al. [48] presented a brief overview of patterns and best practices for the microservice approach. Microservices are self-contained, and this is a core property defining those services that contain every element to run successfully. In other words, the business logic, front- and back-end, and libraries should be enclosed leading to services that could be scaled individually and evolve independently. The monitoring and prevention of fault cascading in SOA call for an interface's ability to

hand over monitoring information. The health status is a piece of vital information to be provided and could be dealt with by the “Circuit Breaker” pattern. Moreover, numerous services could be patched together by applying various concepts such as choreography and orchestration. It is noted that the choreography concept should be favoured in practice as it has a higher degree of freedom. Furthermore, containers could be implemented for better testability, ease of service deployment and better scalability. Finally, services are bound to be deployed in different versions that should be handled. Different methods could be applied such as the immutable serve pattern, blue-green deployment and the canary release pattern. All the aforementioned methods have yet to be applied in the IoT domain at the time of work from Butzin et al. [48]. Finally, it is noted that the microservice approach comes from another direction than the IoT, but the architectural goals are the same.

3.4 Conclusion

The research on the relevant work has indicated that marketplaces in the automotive industry predominantly focus on data. Marketplaces for data offer services around creating channels for safe data exchange. A new trend is in applying blockchain for incentivising and monetarising the data availability by users. This kind of marketplace seems confined for use in CCAM community, as there is a range of products for a marketplace to include.

The service marketplaces in the market are generally private and target specific audiences. These are limitations to the open participation for members of the CCAM community that SHOW Marketplace aims to achieve. A marketplace that appeals to a broader audience can aid the community's growth. Another worth-noting point is that collaborative spaces for the CCAM community are hard to find. Considering all the above, the SHOW Marketplace sets to build an open place where the CCAM community will actively participate via a wide range of products. The inclusion of a wide range of products differentiates the SHOW Marketplace from the rest of the marketplaces. For example, researchers should find their place in the marketplace, since specific products are included in the marketplace tailored to fit their needs. Details on the hosted product items are presented in the following chapters.

The desk research for service marketplaces in other sectors indicates the roles included in these applications. The main roles can be summarised in three general categories: users searching for applications, developers supplying the applications, and administrators curating the marketplace. Marketplaces extend the general roles by allocating users with identities closely related to the target community. Examples are the project GOAL which includes achievers and supporters and the project Bonseyes which includes innovators and researchers. Hence, the SHOW Marketplace draws inspiration from other projects and clearly defines identities for the different members of the CCAM community considering a wide range of active identities. The SHOW Marketplace identities are described in the following chapter in efforts of promoting active participation.

The marketplace considers the SOA as an approach to enhance the service delivery performance and retain their vital features. For this reason, the SHOW Marketplace will consider the philosophy behind SOA not only for its application. Potential benefits for the marketplace in modularizing the services are business value, strategic goals, intrinsic interoperability, shared services, flexibility, and evolutionary refinement. The SOA blends well together with the agile methodology as it can support the continuous improvement of the application.

European projects openly publish material on structures and technologies. The published material is a source to identify technologies for adopting in the marketplace

implementation. The adopted technologies can differ from other projects, but similarities in their delivered functionalities are expected. The research indicates the adoption of different solutions for databases like MariaDB, Postgres, and MongoDB. A framework standing out of the rest is Django for web development that abstracts procedures in the development and allows developers to focus on the functionality. The technologies and tools to deliver this initial marketplace concept are presented in the following chapters.

Finally, the definition of users' identities and the hosted product items facilitate the requirements definition. The requirements should satisfy the user needs and smoothly transition in hosting the product items. The requirements will consider registered and unregistered users to achieve a user-friendly application that encourages them to use it again. The following chapters include details on the requirements.

4 Business Logic

The ambition of the SHOW Marketplace is to be a one stop-shop, similar to the intention of all marketplaces existing in the business world, specialised in the CCAM domain. Hence, the key value propositions of SHOW Marketplace are:

- **One-stop-shop** for automated mobility services – within & beyond SHOW
- **Non commercialised services pool** for the research enhancement in the CCAM field
- **Accelerator** for adoption of CCAM services in Europe
- **Stimulator** for passenger familiarisation to new mobility paradigm

Considering the SHOW marketplace customers' perspective, the proposed solution has the ambition to serve as a place that offers multiple services and products in a unified manner constituting thus, the suggested solution unique in the research field of Marketplace for CCAM. The whole approach of the SHOW marketplace is customer-centric and for this reason, main attributes of the solution are the openness, making accessible the platform to everyone interested in the CCAM field, the user satisfaction, independently of the user experience and the easiness to find the appropriate solution that offers by combining different product items (from the simple one to a complete solution for operators) as will be analysed extensively in the next subchapters.

Undoubtedly, the pricing policy in any marketplace plays a crucial role in the business logic, since this is the main source of profit for marketplaces and the definition of this policy is a mandatory step for the financial analysis. However, due to: a) the fact that the present marketplace is part of an EU funded project having as main aim the stimulation and acceleration of the CCAV paradigm, and b) the fact that there will be many product items (as will be analysed extensively in the deliverable) free of charge, for the time being, the charging policy is not fully specified. The main pillars in which this policy will be based are: the free charging of the product items that are not commercial (thus, not having any price but only empowering the adoption of CCAV paradigm) and the proportional pricing for the commercial ones. In the next versions of the deliverable this pricing policy will be more specific and more details will be provided. SHOW Marketplace constitutes in essence a software product and as with all the existing software, there is a specific software life-cycle. Hence, the initial action in the SHOW marketplace, which precedes the implementation of the platform, is the mapping of the whole ecosystem, which is an activity included in the idea initiation and system concept. The SHOW Marketplace is a vital component in the project's architecture since, as mentioned above in the key value propositions, it can be mainly a place of promotion for partners with innovative services and discovery of software solutions to existing problems for others. As a fundamental step to build upon, the stakeholders of the SHOW Marketplace should be defined. Following the specification of stakeholders, the requirements should be the following step. The requirements could be distinguished into two categories; the functional and the non-functional ones.

This procedure is essential since well-defined and detailed requirements are one of the critical keys to project success. They allow the technical team and any part of the SHOW consortium to:

- Obtain a more profound and accurate understanding of the final SHOW Marketplace.
- Reveal less obvious and suitable for the special domain requirements.
- Distinguish the main roles of the users and define which actions each role could execute.
- Clearly define the functionalities that SHOW Marketplace should include.

- Check the process of the SHOW Marketplace's implementation from a deliverable to another.
- Map the needs of the stakeholders' category for the SHOW Marketplace to be rendered as a prominent but easy-to-use platform.

4.1 Stakeholders

The procedure of defining the entities that will interact with the SHOW Marketplace is not straightforward. The nature of the SHOW ecosystem and the targeted impact of the project in the CCAM domain in Europe render extra challenges and features that should be taken into account. In this chapter, apart from the current definition of the stakeholders, the challenges are presented to provide an achievable understanding of the whole flow and the criteria of the final choice.

4.1.1 SHOW's Stakeholders

The stakeholders consist of a guest user that could be an unregistered user, the main user of SHOW Marketplace that could obtain access to the functionalities of the SHOW Marketplace and the administrator. The entity that is a subject of further and in-depth investigation is undoubtedly, the main user.

As agreed with the consortium, each partner who has developed a service within another project or in SHOW, will be able to upload it in the Marketplace under the conditions that will be decided. Additionally, the autonomous vehicle operators participating in the project should have access to the platform to identify and utilise the services that meet their expectations. Therefore, the user of SHOW Marketplace could be either a Product provider or Product consumer. The SHOW Marketplace's stakeholders are depicted in Figure 5**.**

Figure 5: SHOW Marketplace's Stakeholders

Specifically, the roles and the corresponding description of them are following:

- **Guest user**: Every unregistered actor of the marketplace is considered as a guest user. The guest users could apply for their registration if the site's context

is compatible with their needs (consume or sell a service) or to leave without any further action.

- **SHOW Marketplace Product Consumer**: This is the basic level for registered users of the application that have access to the marketplace's material. This category includes end-users who would obtain, install and use applications, tools or services hosted in the Marketplace. SHOW Marketplace Operators could register, manage their profile, maintain a wish list or installed list, review and rate the product items. At the point that they would also like to upload a product item, they should create a corresponding service provider profile separately from their user profile, as most online marketplaces currently.
- **SHOW Marketplace Product Provider**: This category of end-users is the consortium partners and third-parties that have the knowledge and purpose to develop applications, tools and services that are compatible with the needs of SHOW project and autonomous vehicles domain in general. The SHOW Marketplace, first of all, offers guidelines that should be followed by any product provider. SHOW Marketplace product providers have the ability to upload and host applications, tools or services in the SHOW Marketplace, monetising them and expanding their reach, while also contributing to the ecosystem of SHOW. They can also keep track of their performance, receive ratings and review insights.
- **SHOW Marketplace Administrator/Aggregator**: This category of end-users has complete control over the SHOW Marketplace, including the ability to add and remove products or users at will. SHOW Marketplace Administrators attribute users with access privileges and rights corresponding to their role and can revoke these rights if a part acts against the rules of the platform. Their main task for the SHOW Marketplace maintenance is the validation of applications, tools or services for compliance. To obtain these privileges, consortium members should be in communication with WP6 leader CERTH/ITI, who is in charge of SHOW Marketplace development, at least for the duration of the project.

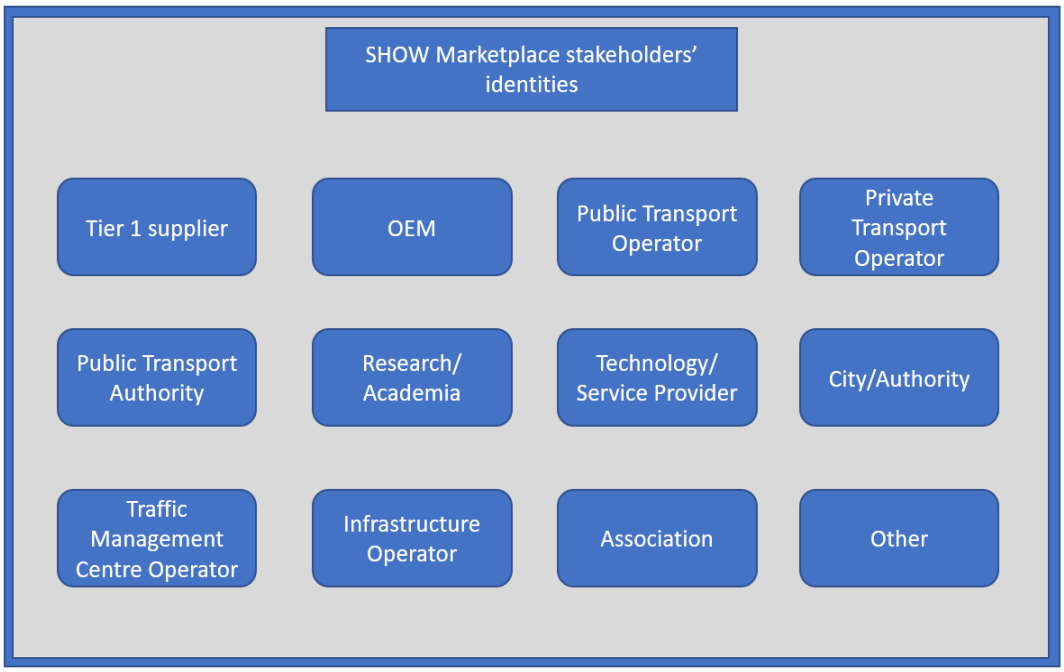


Figure 6: SHOW Marketplace's stakeholders' identities

4.1.2 Stakeholders' Identities of SHOW Marketplace

Apart from the definition of the SHOW Marketplace's stakeholders, the identities of the users that could potentially utilise the platform to circulate, search, upload or download any product item which is relevant to autonomous vehicles' domain should be also determined. From the mapping of the possible users, the main identities that have already been distinguished are shown in Figure 6.

Explaining the previous figure, Tier 1 supplier is the part that provides products and devices, which are almost close to the final product. Moreover, Technology or Service providers could provide already existing solutions in plenty of domains. Usually, the aforementioned parts support the actions of the Original Equipment Manufacturers (OEMs), who obtain an innovative idea and intend to give it for production. In the automated vehicle domains, OEMs are in contact with a plethora of institutes such as Research and Academia Centres to enrich existing and develop new ideas accounting for the different use cases. Furthermore, the CCAM domain encompasses by default Public or Private Transport Operators, Public Transport Authorities and, of course, the authorities of the local area (City, Region etc.). The automated transportation ecosystem could also include Association or undefined parts that are classified in "Other".

It is crucial to underline that each identity will combine the privileges of SHOW Marketplace Product Consumer and SHOW Marketplace Product Provider, as described above, once their registration is completed without further action from the SHOW Marketplace Administrators.

The definition of the current stakeholders' identities is only the initial step for the SHOW Marketplace to exceed the limitation of being a platform solely for developers to sell their applications and operators to purchase them and become a reference point for the autonomous vehicle domain in Europe.

4.2 Product Items

The SHOW Marketplace consists of a platform in which different items that are relevant to automated driving would be available. The material that is to be circulated among the parts of the SHOW Marketplace platform could expand and include solutions, software components, datasets, business plans, material from European projects in the autonomous vehicles domain, an innovative idea or architecture of the network, the topology of the cities in which autonomous fleets operate, code that is under construction, a user interface, a dashboard etc. From this point onward, any material that could be uploaded or downloaded from the SHOW Marketplace is referenced as a product item.

Figure 7 presents the definition of the product items. Each product item is unique and has its properties. As a next step, a separate data scheme for each product has been created to specify the format and the relevant information for every product item. These details are included in Appendix I. Additionally, a mock-up example for each product item is also attached.

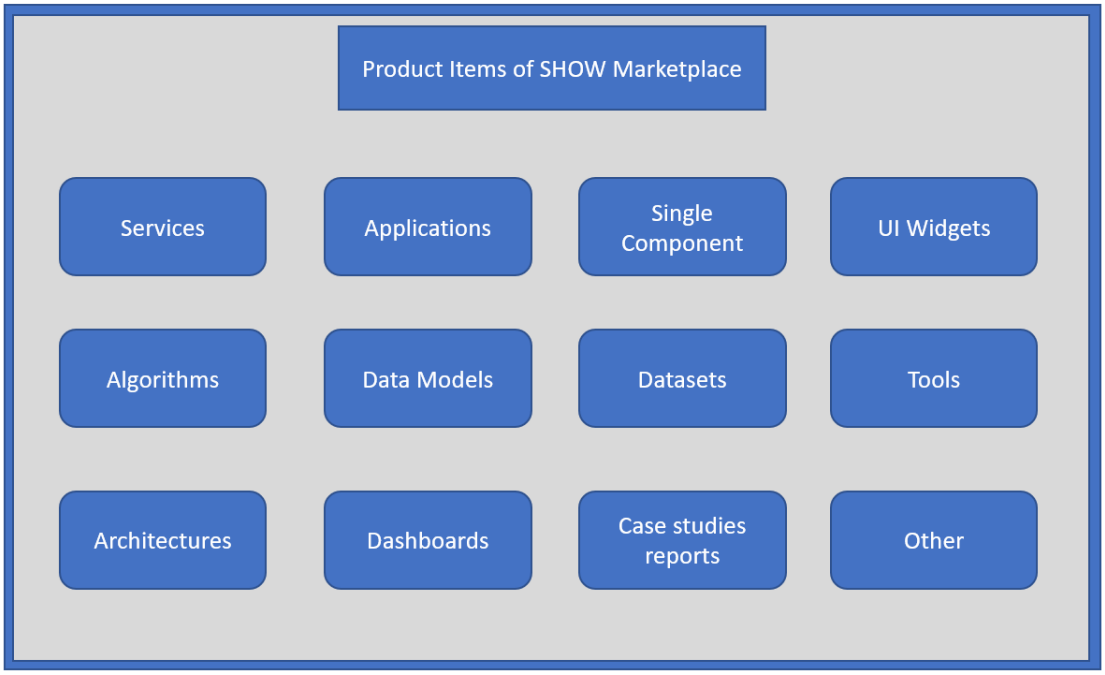


Figure 7: SHOW Marketplace Product Items

4.3 Requirements

This chapter aims to define the requirements coherently and profoundly. Specifically, the requirements define all functionalities that the marketplace has to fulfil and analyse all the constraints that the system should respect. The requirements specification is a mandatory and prerequisite procedure for the actions related to the application's implementation (architecture, design). As previously mentioned, the requirements consist of two categories, functional and non-functional. The functional requirements are composed to 8 modules and the non-functional ones are related to 8 categories, as depicted in Figure 8. Obviously, the most prominent task remains the promotion of the applications and services via the SHOW Marketplace platform. Therefore, some requirements are referred to only in these types of product items.

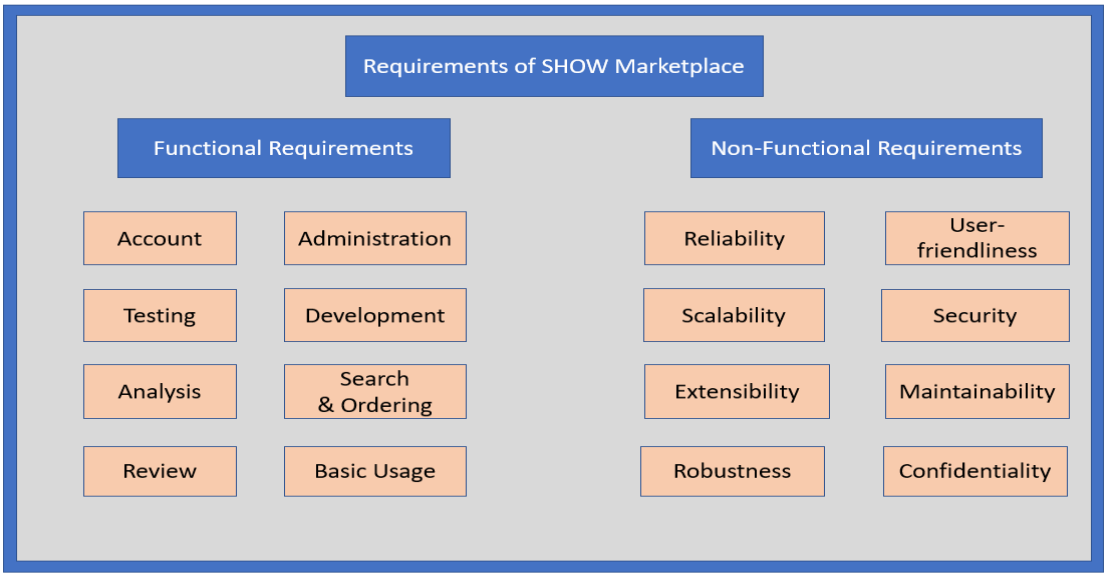


Figure 8: Functional and Non-functional requirements of SHOW Marketplace

4.3.1 Functional Requirements

A model of **8 modules** is used to describe the main structure. These modules are **Account, Search & Ordering, Basic Usage, Review, Analysis, Development, Testing** and **Administration**.

Table 1: Functional Requirements of SHOW Marketplace

Module	Title	Req_Id	Description
Account	**Registration**	**Req_001**	As a **guest user**, I **must** be able to create an account by using an email address and filling in personal details
	Login	**Req_002**	As a **registered user**, I **must** be able to log into the SHOW Marketplace platform.
	Delete	**Req_003**	As a **registered user**, I **must** be able to delete my account.
	Edit	**Req_004**	As a **registered user**, I **must** be able to edit my profile details.
	Reset password	**Req_005**	As a **registered user**, I **must** be able to reset my password for security or in case of forgetting it.
Search & Ordering	**Search for Product Items**	**Req_006**	As a **guest/registered user**, I **must** be able to search for product items of my interest.
	Order by	**Req_007**	As a **guest/registered user**, I **must** be able to order the results by a specific criterion for product items of my interest.
	Save Product Item Metadata	**Req_008**	For saving the metadata of a product item to the database, as a **registered user**, I **may** be able to search for product item data. Using a microservice, the product item data of a chosen product item can then get saved to the database
	View of the available product items	**Req_009**	As a **guest/registered user**, I **must** be able to view the list of available products along with some information about the products.
Basic Usage	**Featured Product items**	**Req_010**	As a **registered user**, I **may** be able to see the suggested product items.
	New Product items	**Req_011**	As a **guest/registered user**, I **may** be able to see the newly added product items.
	Product item Description	**Req_012**	As a **guest/registered user**, I **may** be able to see the detailed description for each product item.
	Show my orders	**Req_013**	As **a registered user**, I **may** have the ability to see the transaction history (installed applications/ tools/ services). For this, a microservice loads the data of all saved applications from the database.

Module	Title	Req_Id	Description
	Recommendation per Applications	**Req_014**	As a **guest/registered user**, I **may** be able to have at my disposal a list of relative-recommended product items when I visit the application description page.
	Recommendation per User	**Req_015**	As a **registered user**, I **may** be able to see a list of relative-recommended product items when I am on the installed applications page.
	Get free tools Product items	**Req_016**	As a **registered user,** I **must** have the ability to download product items if they are freely available.
	Buy Product items	**Req_017**	As a **registered user**, I **must** have the ability to purchase product items in the marketplace if payment is necessary.
	Show my product items	**Req_018**	As a **registered user,** I **must** have the ability to see and update all the product items that I have uploaded.
Review	**Review Application**	**Req_019**	As a **registered user**, I **may** have the ability to review my installed applications by triggering this process from the application page in the user interface. Completed feedback can be sent to a microservice that is responsible for storing it in the database.
	Evaluate registered Product items	**Req_020**	As a **registered user,** I **must** have the ability to rate my installed product items.
	Show All Surveys per Product item	**Req_021**	For accessing all the stored reviews relevant to a specific product item, as a **guest/registered user,** I **may** have this ability on the product description page. By doing this, the data for the product is retrieved from the database.
Analysis	**Analyse Product Metadata**	**Req_022**	As a **guest/registered user,** I **may** retrieve the available metadata from the database. Additionally, I **may** have the ability to analyse this data in a form that offers additional insights.
	Analyse Product Survey Data	**Req_023**	As a **guest/**registered user, I **may** have the ability to get the analysed survey data when I am on the product page.
Development	**Develop new Product items**	**Req_02**	As a **registered user**, I **must** have the ability to create a new application or, generally, a product item.
	Register new Product items	**Req_025**	As a **registered user**, I **must** have the ability to upload in the

Module	Title	Req_Id	Description
			marketplace a new application or, generally, a product item.
	Benchmark existing Product items	**Req_026**	As a **registered user**, in the progress of preparing a new service or a product item, I **may** have the ability to use the existing source code of the services, the applications and generally the product items which are available in the marketplace.
	Development tools	**Req_027**	As a **registered user**, I **may** have the ability to access the developer's tools.
Testing	**Test products**	**Req_028**	As a **registered user**, I **may** have the ability to test services, algorithms and applications on demo data collection.
	Development Testing	**Req_029**	As a **registered user**, I **may** have the ability to validate technically the registered services and applications.
Administration	**Monitor online services activity**	**Req_030**	As an **administrator**, I **must** have the ability to check the functionality of web services in the marketplace.
	Monitor users	**Req_031**	As an **administrator**, I **must** have the ability to check the activity of the users.
	Report current status	**Req_032**	As an **administrator**, I **may** have the ability to characterise the system according to its consumption condition.
	Apply security policy and secure transactions	**Req_033**	As an **administrator**, I **must** have the ability to check that the system and personal data remain safe, according to the existing standards.

4.3.2 Non-Functional Requirements

Non-Functional Requirements may be more critical than functional since the system may be useless in case of not fulfilling them. These requirements are related to more abstract features and describe all the characteristics that render the platform appealing to the users.

The main non-functional requirements are:

- **User-friendliness**: The system should characterise with ease of use to be used by all the users considering a minimum level of experience based on the corresponding users' roles. A major part of the user-friendliness is the provision of the Graphical User Interface personalised in the historical activity of each dedicated user. The fulfilment of the respective functional requirements could lead to a high personalisation level.
- **Security:** The system must be protected from any hazard or threat. All the security measures should be in place to verify the identity validity of each

operator and service provider. Especially for service providers, the quality of the uploaded service is a subject of an investigation to reveal cases of malicious applications. **Furthermore, sensitive information should be protected by any hostile part.** The security requirement may generate several relative functional requirements.

- **Reliability**: The reliability of a platform is referring to the probability of a system operating without failing over a defined duration. The best-case scenario dictates that the final product is defect-free with no delays and operates consistently and correctly under any given circumstance. The application has to be reliable in any situation. Therefore, the application must be accessible in all possible situations.
- **Scalability**: Scalability is the capacity of the platform to handle a growing number of operators and service providers. It portrays the ability of future registered users to maintain the marketplace and its data consistent. The scalability of SHOW Marketplace ensures that, during the project and after the completion of the implementation, the platform will be sustainable. For this reason, all parts of the platform ought to be replicable, and the software utilised should have been executed considering scalability.
- **Maintainability**: Maintainability alludes to the ability of administrators to fix, improve and comprehend software code. Software maintenance is a critical part of the software development life-cycle that begins after the launch of the SHOW Marketplace platform. Maintainability renders the marketplace's users feedback a priority. Both operators and service providers are to provide their feedback, as they could recognise any problem in the utilisation of the platform. This includes repairing bugs, optimising the already existing functionality and adjusting code to anticipate future problems and challenges. The life duration of the platform depends on the administrator's ability to keep up with maintenance prerequisites. Extended and detailed documentation during the development could be crucial in the maintainability of the SHOW Marketplace platform.
- **Extensibility**: In the implementation of a marketplace, extensibility is defined as the quality of being designed to permit the expansion of new features, capabilities or functionality. It could be a degree of the capacity to expand a system and the level of exertion demanded to implement the desirable extensions. The characteristics that guarantee the extensibility of the platform are flexibility, configuration, customise-ability, upgrade-ability, accessibility, and collaboration. For extensibility, documentation is more than essential.
- **Robustness**: The robustness or health of the SHOW Marketplace platform has a primary importance in its success. Stability of SHOW should be guaranteed before the launch of the marketplace among the consortium of SHOW which could be characterised as a high availability environment. Additionally, the platform should be able to process or intercept incorrect user input. Therefore, the application should react in a predefined way.
- **Confidentiality**: The SHOW Marketplace should ensure that the personal data of the registered user will be protected through dedicated security mechanisms. The Policy Privacy of the SHOW Marketplace platform should be aligned with the respective GDPR rules.

4.4 Use Case scenarios

This subchapter presents the two primary use case scenarios on users' interaction with the platform. The use case scenarios elaborate on the actions taken to ***upload*** an object on the marketplace and ***buy*** an uploaded object, also including the prerequisite

actions of ***login/register*** and the complementary action of ***filtering/searching***. These are the fundamental actions for users regardless of being a provider or a consumer of a product item. Below, the sequence diagrams and the corresponding flowcharts are presented.

4.4.1 Sequence diagrams

The sequence diagrams depict the actions in a timely ordered fashion for readers to get an overview. The actor in all the sequence diagrams for each use case scenario is the marketplace user responsible for commencing the actions.

4.4.1.1 Upload product items

Uploading products is essential for the operation of a marketplace. Readers should not consider the term product as solely a physical object or application. Products are broader as they follow requirements and the specified user identities. A vivid marketplace that aims for operational longevity requires the participation of the community to share its products. The users are to share their products by uploading objects, filling text fields or setting tags to facilitate consumers to discover and consume these products.

The uploading process begins from a user's own intention to share a product. Users interact with the underlying automated components of the marketplace via a frontend application. The frontend application is titled SHOW Marketplace UI and guides users to upload their product successfully. Basic architectural components in the marketplace are the user management and validation process that safeguard the uploading procedure.

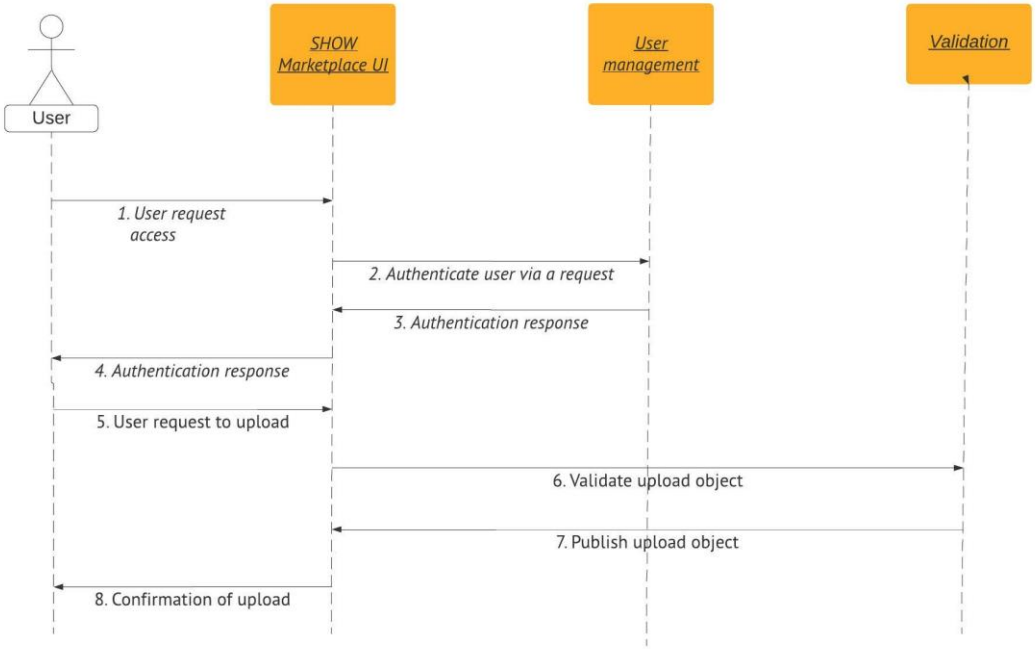


Figure 9: Sequence diagram for uploading a product

4.4.1.2 Consume product items

This subchapter concludes with the presentation of a complementary procedure to a product's upload. This procedure is the consumption of an uploaded product. The term consumption is relevant to using such a product and has no association with the product's price or quantity. It becomes apparent that the consumption takes place once

an item is uploaded and discovered by users browsing through the marketplace. Monetisation of products is possible and depends on the product owner's volition to set a price or not. A monetary incentive can maintain owners' active participation in the marketplace.

The consumption of an object begins with an actors' action to acquire the discovered product that suits their requirements. Actors are the subscribed users in the marketplace who browse through the uploaded products accommodated on the marketplace. Once the marketplace authentication mechanism validates a user, the user interacts with a frontend UI, called SHOW Marketplace UI. This UI presents in a convenient way to the user the uploaded products that he/she can choose from. Once the user identifies an appropriate product, an order request is transmitted to the payment component. The payment component is responsible for making a decision on the order, upholding payments procedures, monitoring transactions and transmitting the decision back to the user. The user is informed of the decision, either approval or denial of the transaction, via the SHOW Marketplace UI.

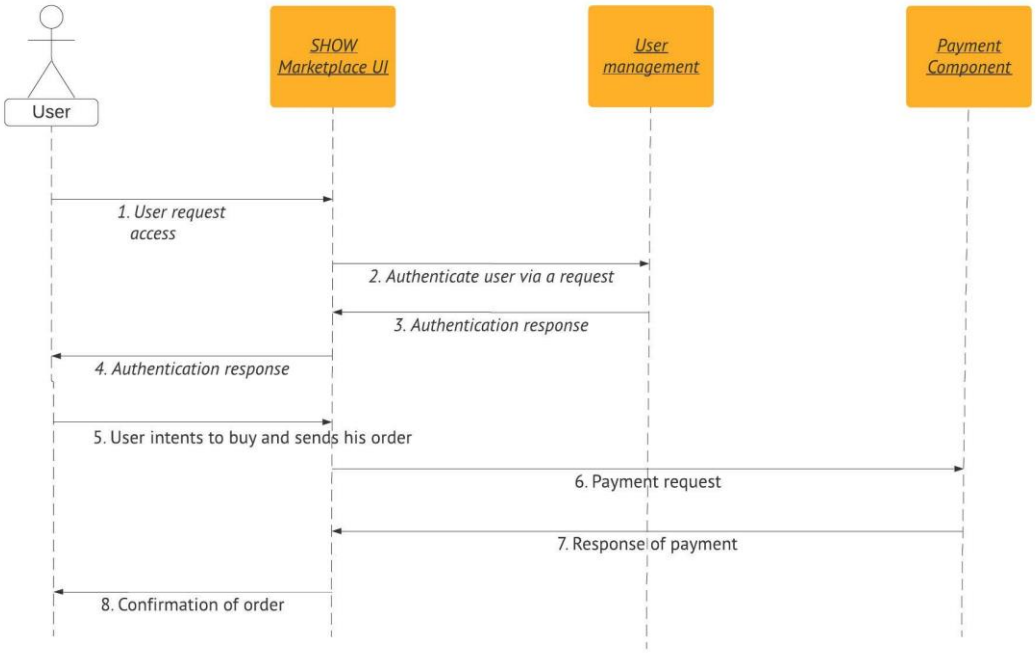


Figure 10: Sequence diagram for consuming a product

4.4.2 Flowcharts

Flowcharts constitute a type of diagram that depicts the interaction between the users and the corresponding components. The separate actions are presented in sequential order and account for different situations based on users' actions.

4.4.2.1 User Registration

The first flowchart displays the registration procedure. Unregistered users that wish to join the marketplace, request their registration. The Application Client prompts the user to provide SHOW Marketplace with the information deemed mandatory for the registration. Unregistered users fill in the corresponding information and set their preferable username and password. Several requirements are in place for username and password, as the uniqueness of the username. The Authorisation Server checks

if the credentials are indeed unique. In the case that the user already exists, the login page becomes available for the corresponding procedure to be performed. In the case that the credentials are unique, they are saved in the database. The Application Client presents the platform policy rules to the candidate user. After the user's consent to these rules, the user is saved in the database and the procedure is successfully completed.

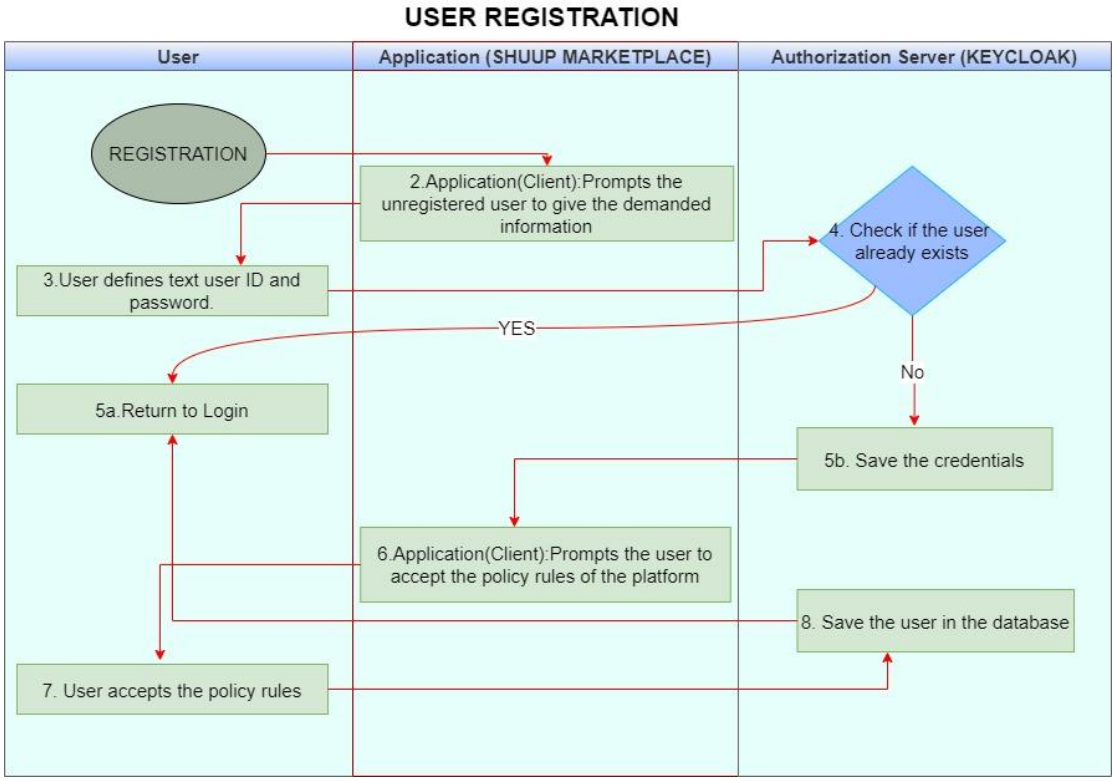


Figure 11: User Registration flowchart

4.4.2.2 User Login

The next flowchart depicts the procedure of login. When a user wants to have access to a specific resource or service from a specific service provider using Oauth2.0 protocol, there are sequential steps for the login process to be completed. As presented in the corresponding depicted image when a user asks to gain access to the service provider's resources, the service provider asks for credentials and redirects the user to the authorisation server. The user gives his/her credentials and after the validation has been finalised the authorisation server first generates an authorisation code that has to be signed from the service provider and second the authorisation server generates a token for the user that describes which resources the specific user can access in the web app (in this case, in the marketplace). The token is then sent to the user and using that token the user can access the desired resource in the service provider web application (marketplace).

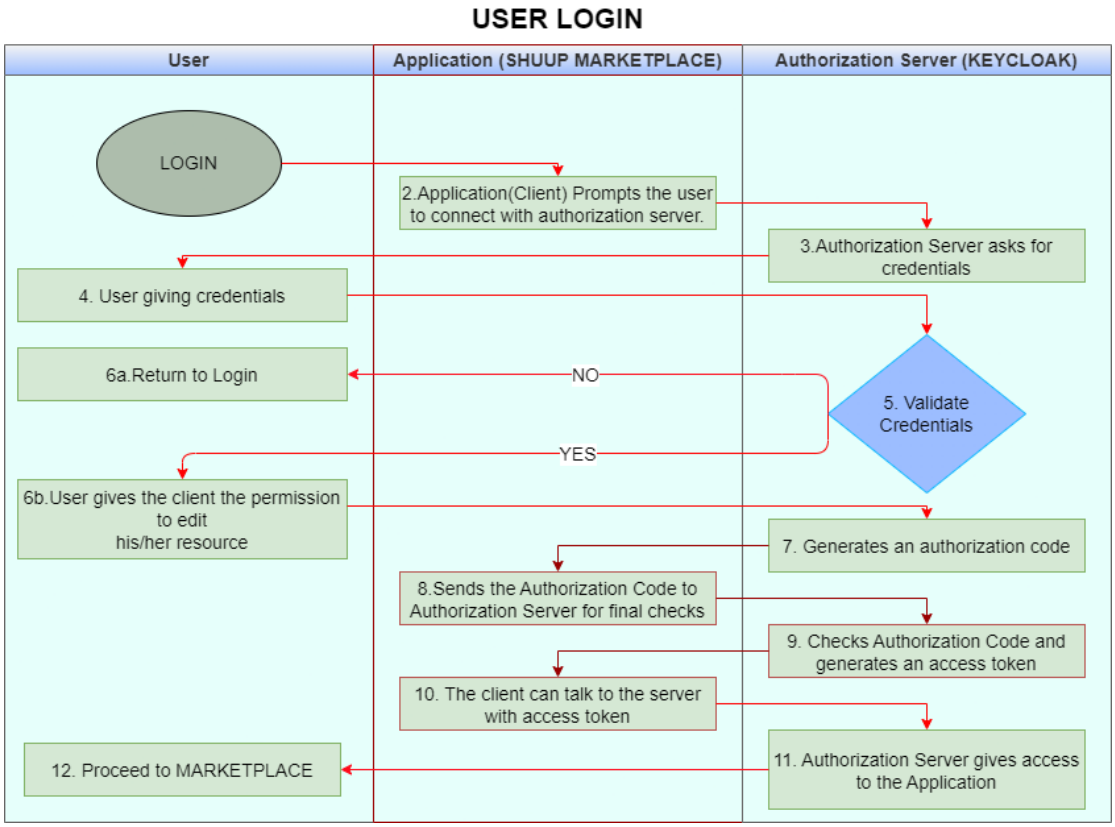


Figure 12: User Login flowchart

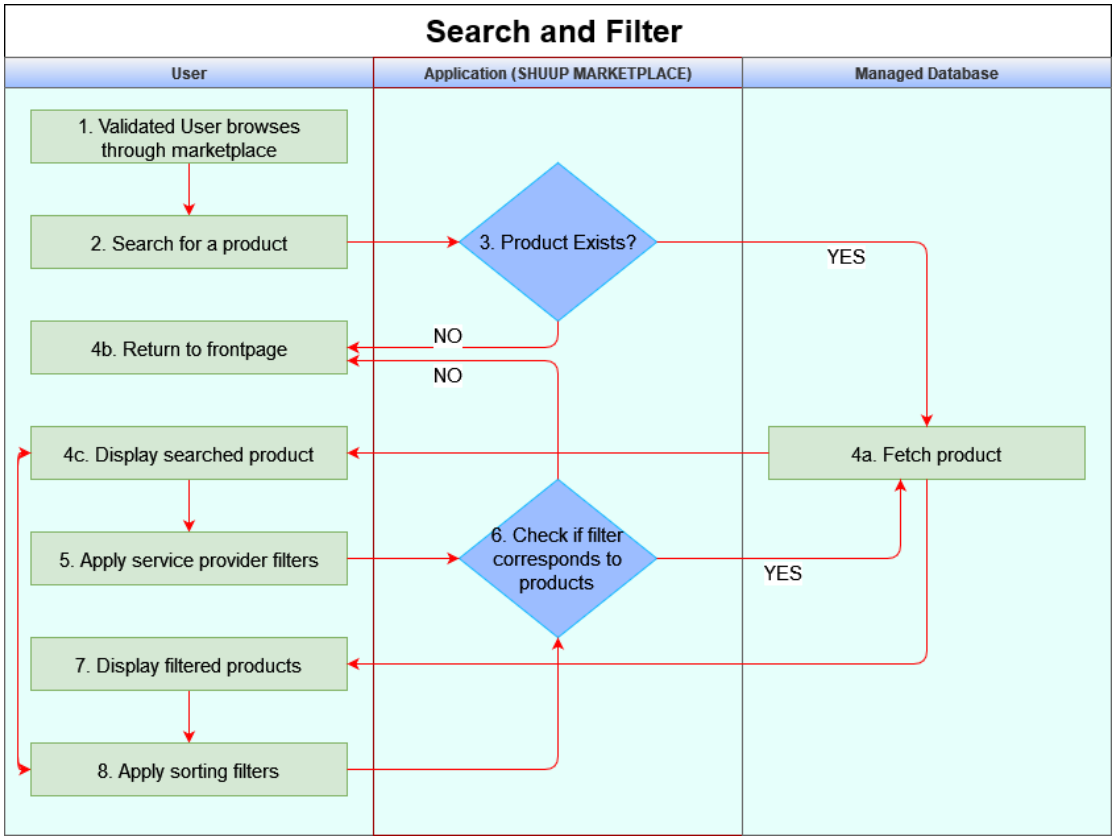


Figure 13: Search and filter flowchart

4.4.2.3 Filtering/Searching

The third flowchart shows the complementary procedures of filtering and searching. The main functionality of this flowchart is explained as follows. When a validated user searches for a product, a “daemon” checks if this product exists in the Marketplace, communicates with the storage/database to fetch the product and then the product is displayed to the user. Afterwards, the user has the ability to filter his/her searches, by applying “sorting” and “service provider” filters. Then a similar procedure follows for the filtering, which is to search whether a product item covering the filters exists, fetch it from the database and display it to the user. In case the search or filtering is invalid, the user is returned to the front page.

4.4.2.4 Purchase a product

One of the expected users' actions is the ability to purchase a product accommodated in the marketplace. The following flowchart depicts the procedure in sequential actions. The procedure kicks off with a registered user, since guest users do not possess the right to purchase products. The registered users will browse through the marketplace interface to pinpoint products that interest them. There are cases when users hesitate to complete their initial purchase, so they should return to the front-end marketplace UI with no actions. On the other hand, users committed to purchasing their selected products should trigger a system's validation for the availability of the products. While the products are available for purchase, the user should be guided to an external page that the bank operates to pay for the products safely. The bank's external page should check if the user has the monetary amount to conclude his purchase. The bank's external site should return its verdict on the payment to the marketplace. Only the successful payments should be stored in the marketplace database to create a purchase history. In the opposite case, a message in the marketplace should indicate the purchase rejection.

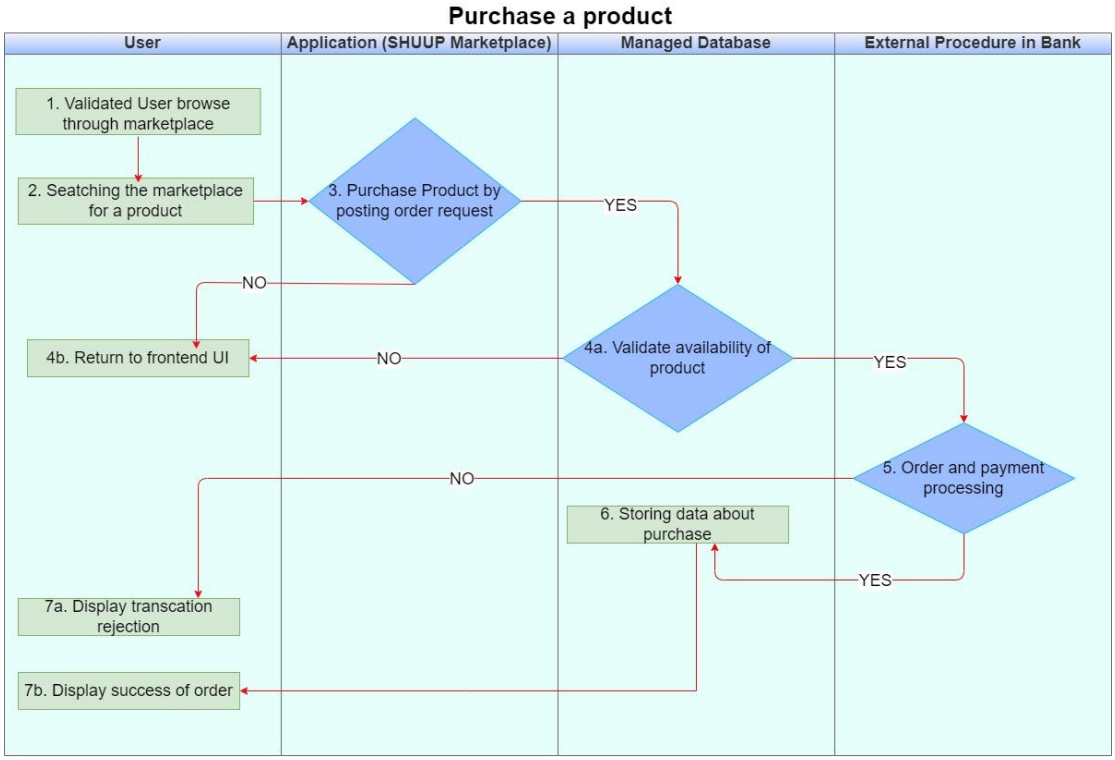


Figure 14: Purchase a product flowchart

4.4.2.5 Register a product

The final flowchart shows the register of a product procedure. Initially, the user will have to fill in the mandatory fields defined by the forms of the website. The first form concerns the completion of the title that the product will have, while the second is related to the choice of the category. In the third form the provider completes the description of the product he will register, in the fourth form he must enter the technical prerequisites of the product to be functional during use by the consumer and finally, he must upload at least a representative image. These 5 fields are not stored directly in the database but in the cache of the session, thus entering a pending process. For the process of registering a product to be completed, the system administrator will check for its validity and then, if approved, all the remaining information that has been given before is stored in the database. If the administrator rejects the product then an appropriate message appears on the provider's screen and asks him to upload the product again having made the required changes.

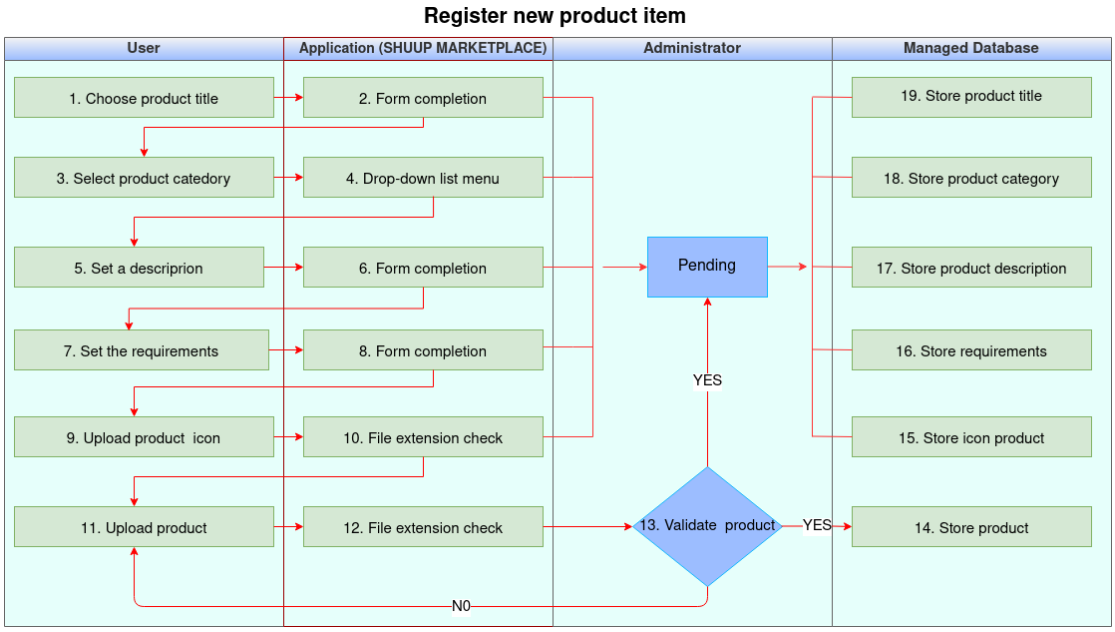


Figure 15: Register a product flowchart

4.5 Personalisation of the SHOW Marketplace

The information of the marketplace has been structured in a way to support personalisation. The personalisation functionality on the marketplace will include content and graphical user interface (GUI) personalisation.

The marketplace provides personalised content to boost the users' engagement with the platform. In other words, the marketplace content serves the user need according to specific attributes. These attributes are 1) the location of the user, 2) the profile of the user (type of stakeholder: developer, PTO, PTA, OEM, Tier 1 provider, service provider, research/commercial, etc.) and 3) the user's marketplace behaviour (historical data). A prerequisite on the above process is the advanced completion of the marketplace user's profile and their consent to share their web browser's historical data (using browser cookies).

Based on the profile information, the marketplace categorises its content to better fit the user needs. For example, the top items displayed in the marketplace should be in a convenient (geographical) location to the user (in CCAM for PT especially, the strengthening of the local ecosystems makes sense) and order lower the rest of the

choices. The marketplace will return the results that best fit this specific stakeholder's type in advance. Finally, the user's past behaviour (historical data) on the marketplace helps in the results prioritisation as previously visited items indicate the user's interest and should be ordered higher in the results.

In more detail, the prioritisation process provides results directly related to the stakeholder's type. Furthermore, these items are prioritised by country, then by continent and last, the rest items. Finally, the user's historical data – the visited items of the marketplace give higher prioritisation to the results.

Additionally, the marketplace provides graphical user interface personalisation. This is a specific functionality that supports better experience on the browsing of the marketplace. Although today's web browsers support some accessibility functionality to assist impaired users, like screen reading and content magnification, they do not support contrast change. The accessibility guideline that meets contrast requirements is WCAG 2.0 level AA. This guideline requires a contrast ratio of at least 4.5:1 for normal text and 3:1 for large text. The marketplace fills the accessibility contrast gap by letting visually impaired users select between three different colouring combinations with specific contrast. The user is able to select the one which fits better their needs.

5 Open-Source Platform Selection

There are several tools to choose from for Marketplace development. The most popular and industry standards are included in this chapter. This chapter presents basic features, advantages and disadvantages of each platform so that the reader can understand the reasons behind the SHOW open-source platform selection, which was chosen to implement the SHOW Marketplace.

5.1 Shuup

Shuup [50] is an open-source e-commerce platform and appeals to any business that seeks to allow vendors to sell services or physical products. The Shuup platform is written in Python and Django that are well-known technologies with extensive support from the community. Shuup offers a variety of features such as:

- Cataloguing/Categorisation
- Order Management
- Payment Processing
- SEO management
- Shopping Cart
- Categorisation/ Grouping

Shuup is extensible, free, fast to deliver, simple to use and ready to serve the demands of the global market.

Figure 16: Shuup Logo

5.2 Magento

Magento is an open-source E-Commerce platform and an experienced solution among other Marketplace platforms. It is written in PHP and offers a large variety of extensions that can help developers to deliver a high-quality Marketplace platform. Some of the features which are included in the open-source version are:

- Marketing and Conversion tools
- SEO Features
- Site Management
- Catalogue Management
- Checkout, Payment and Shipping
- Order Management
- Customer Accounts and Customer Service
- Analytics and Reporting
- International Support
- Mobile Commerce

Magento [51] was a successful application that was bought by Adobe in May 2018 for $1.68 billion. More than 100k online stores have been created using Magento and the platform has been downloaded more than 2.5 million times.

Figure 17: Magento Logo

5.3 Sellacious

Sellacious [52] is an open-source e-commerce software that can be installed with Joomla. It has been downloaded more than 100k times and offers more than 45 payment options. The most important features in Sellacious are:

- Detailed Rule Filter
- Multivendor Architecture
- Compare Products
- Multiple Variants
- Sell Software
- Template like Amazon, Ebay, Olx, Etsy
- Shippable Product Store

Figure 18: Sellacious Logo

5.4 Mailchimp Open Commerce (Reaction Commerce)

Mailchimp [53] (formerly Reaction Commerce) is an open-source, API-first made for technical retailers. Mailchimp is built in Node.js, React and GraphQL and it can be easily developed with Docker and Kubernetes. Mailchimp core is flexible at scale and can be extended to suit every business model and its needs without limit. A summary of features is included below:

- User, Role and Access Management
- Internationalisation
- Integration APIs
- Data import and export tools
- Building and Personalising emails
- Basic Reporting

Figure 19: Mailchimp Logo

5.5 Dokan – Wordpress

Dokan [54] is a WordPress plugin. Dokan has a free version with the following features:

- Unlimited Vendors
- Frontend Vendor Dashboard
- Order Management
- Vendor Withdraw system
- Store Widgets
- WordPress support
- Order Email Notifications
- Individual Vendor Stores

Dokan supports two types of payment methods and also offers a cryptocurrency payment gateway. Dokan is easy to use, very fast in deployment and is compatible with a variety of WordPress plugins such as: shipping and auditing plugins. Dokan also offers templates similar to eBay, Shopify, Amazon and Magento.

Figure 20: Dokan Logo

5.6 Aimeos Laravel

The Aimeos E-Commerce platform [55] is an open-source and free platform that scales from one to one million items. Aimeos is built in Laravel framework, is PHP-based and is used for multi-vendor setup. Aimeos is easy to install and it is available in more than 30 languages. Some of the most highlighted features are:

- Multi-vendor
- Bundles, vouchers, virtual configure
- 100 + payment methods
- JSON REST API
- Every product can have a subscription with recurring payment
- Full RTL language support

Aimeos is fast, ultra-flexible and scalable.

Figure 21: Aimeos Logo

5.7 Ever Demand Platform

Ever platform [56] is an open E-Commerce platform for digital marketplaces. Ever is built with Typescript and some of the basic features are:

- Single/Multi store
- Multi-Vendor
- Peer-to-Peer marketplaces
- Admin website
- Multi-language
- Products catalogues
- Real Time product Management
- Gateway and Payment processing
- Plugins / Extensions / Custom Fields

Figure 22: Ever Logo

5.8 Comparison Table

After a meticulous search for the available open-source platforms presented above, the Shuup platform was selected for the development of the SHOW marketplace. The reasons that led to this choice concern both the prerequisite features and the way the Django framework works. Initially, the choice of the database, and the compatibility with the Keycloak authentication mechanism, played a crucial role in the marketplace open-source platform selection. Shuup is based on the Django framework providing flexibility for the database selection and with minor changes, the result can be easily changed and adapted to the required operation of the marketplace. The Python language is used for the backend functionalities and can also connect to the Keycloak so that users can register and log in to the marketplace.

By selecting Shuup as a reference, the marketplace can be adapted with very small changes in the code to any requirement, as it uses dynamic objects and not repetitive functions. It also follows the Model View Controller (MVC) architecture which ensures the efficiency of software development but also its scalability. Finally, the mechanisms for security against possible cyber-attacks that can take place in real-time on the

marketplace are much easier to implement than any other application for both client-side attacks and server-side attacks.

Table 2 summarises the prerequisite features of each tool that was described in this chapter and highlights the reason for choosing the Shuup as a reference and basis for the development of SHOW marketplace. Shuup covers most of the needed requirements. For these reasons, Shuup open source platform was selected for the implementation of the SHOW marketplace.

Table 2: Open source platforms comparison table

		Marketplaces								
		Aimeos Laravel	I3-Core	SHUUP	Reaction	Ever	Sellacious – Joomla	Wordpress – Dokan	WMar-ket	Ma-gento2
Technical Characteris-tics	Programming language / framework	PHP	Python Django & eclipse mosquitto	Python Django	Node JS	Node JS	PHP	PHP	Java Web Applica-tion	PHP
	Database	MySQL	MySQL	Mongodb	Mongodb	Mongodb	MySQL	MySQL	MySQL	MySQL
	Development Team Experi-ence (1-5)	2	2	**4**	2	2	2	2	2	2
	Availability of Keycloak	**YES**	NO	**YES**	**YES**	NO	**YES**	**YES**	NO	**YES**
Require-ments	Registration	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**
	Login	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**
	Delete Account	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**
	Edit Account	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**
	Reset Pass-word	**YES**	NO	**YES**	**YES**	**YES**	**YES**	**YES**	NO	**YES**
	Professional account	NO	**YES**	**YES**	**YES**	NO	**YES**	**YES**	NO	**YES**
	Search for Apps	YES	NO	**YES**	**YES**	NO	**YES**	**YES**	**YES**	**YES**
	Order by	YES	NO	**YES**	**YES**	**YES**	**YES**	**YES**	NO	**YES**
	Save App Metadata to Database	YES	NO	NO	**YES**	NO	NO	**YES**	NO	NO

		Marketplaces								
		Aimeos Laravel	I3-Core	SHUUP	Reaction	Ever	Sellacious – Joomla	Wordpress – Dokan	WMar-ket	Ma-gento2
	Featured Applications	**YES**	NO	**YES**	**YES**	NO	**YES**	**YES**	NO	**YES**
	New Applicatons	**YES**	NO	**YES**	**YES**	NO	**YES**	**YES**	**YES**	**YES**
	App Description	**YES**	**YES**	**YES**	**YES**	NO	**YES**	**YES**	**YES**	**YES**
	Show All Installed Apps	**YES**	NO	**YES**	**YES**	NO	**YES**	**YES**	NO	NO
	Recommendation per App	**YES**	NO	NO	**YES**	NO	**YES**	**YES**	NO	NO
	Recommendation per User	**YES**	NO	NO	**YES**	NO	NO	NO	NO	NO
	Get free tools, applications and services	**YES**	**YES**	**YES**	**YES**	NO	**YES**	**YES**	**YES**	**YES**
	Buy tools, applications and services	**YES**	**YES**	**YES**	**YES**	NO	**YES**	**YES**	NO	**YES**
	Review App	**YES**	NO	NO	**YES**	NO	**YES**	**YES**	NO	**YES**
	Evaluate registered tools, applications or services	NO	NO	NO	**YES**	NO	**YES**	**YES**	NO	**YES**
	Show All Surveys per App	NO	NO	**YES**	NO	NO	NO	NO	NO	NO
	Show Survey Data	NO	NO	NO	NO	NO	**YES**	NO	NO	NO

		Marketplaces								
		Aimeos Laravel	I3-Core	SHUUP	Reaction	Ever	Sellacious – Joomla	Wordpress – Dokan	WMar-ket	Ma-gento2
	Analyse App Metadata	NO	NO	NO	NO	NO	NO	NO	NO	NO
	Analyse App Survey Data	NO	NO	NO	NO	NO	NO	NO	NO	NO
	Develop new services	NO	NO	**YES**	NO	NO	**YES**	**YES**	NO	**YES**
	Register new services	**YES**	NO	**YES**	**YES**	NO	**YES**	**YES**	NO	**YES**
	Benchmark ex-isting services	**YES**	NO	**YES**	**YES**	NO	NO	NO	NO	NO
	Development tools	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**
	Test services	NO	NO	NO	NO	NO	NO	**YES**	NO	**YES**
	Development testing	NO	NO	NO	**YES**	**YES**	NO	**YES**	NO	**YES**
	Monitor online services activ-ity	**YES**	NO	**YES**	**YES**	**YES**	**YES**	**YES**	NO	**YES**
	Monitor users	**YES**	NO	**YES**	**YES**	**YES**	**YES**	**YES**	NO	**YES**
	Report current status	**YES**	NO	**YES**	**YES**	**YES**	**YES**	**YES**	NO	**YES**
	Apply security policy and se-cure transac-tions	**YES**	NO	**YES**	**YES**	**YES**	**YES**	**YES**	NO	**YES**
	Total	**24**	**9**	**23**	**27**	**12**	**25**	**27**	**9**	**24**

		Marketplaces								
		Aimeos Laravel	I3-Core	SHUUP	Reaction	Ever	Sellacious – Joomla	Wordpress – Dokan	WMar-ket	Ma-gento2
Program-ming Com-munity	Stars in GitHub	4.2K	5	1.5K	11.1K	973	204K down-loads/No github	141	6	9.3 k
	Watches in GitHub	191	7	117	479	68		22	9	1.4K
	Latest update	2/23/2021	2/15/2021	2/26/2021	2/10/2021	2/26/2021		04/03/21	08/01/16	02/04/21

6 Design & Development

This chapter presents the implementation of the SHOW marketplace via the architectural framework and the underlying technologies used in its deployment. Firstly, a subchapter presents the initial concept for the architectural design along with the fundamental technologies used in its deployment. The framework Django is the initial reference in the chapter, as the marketplace is based on that framework. Other marketplace's elements that are elaborated are the database and the security mechanisms that are in place. The registration and the credential validation process during login are described, as they are processes that are obligatory for operators and service providers. Also, how the data are stored in the database is described along with the procedure that the administrator will follow to give or remove permissions to a user. The chapter concludes with the presentation of two distinct protocols. The first concerns product providers who are to register their service. The latter protocol is about consuming a service that is available to the marketplace by the registered users. Consequently, the following chapters present the framework on which the SHOW marketplace was based, the database that stores all the information and data of users and the way the platform operates.

6.1 Architectural framework

This subchapter describes the marketplace architecture that defines the different components existing in the software, the components' interactions and the different technologies selected for implementing the marketplace. As described in the methodological approach chapter, an iterative procedure was implemented to conclude the final version of the architecture.

Figure 23 displays the final current architecture of the marketplace. First of all, SHOW Marketplace Services use valuable data from the database to provide access to the marketplace's users with respect to the SHOW product items. Moreover, the way to host services and products in the SHOW Marketplace is analysed in detail in the following subchapter, 7.1. Furthermore, the requirements elicitation procedure profoundly dictates the modular SHOW Marketplace Services application logic (essentially the main functionalities). More specifically, the functionalities are composed of: *Account*, *Basic Usage*, *Review*, *Development*, *Search & Ordering*, *Testing*, *Administration* and *Analysis*.

Additionally, this final version presents the SHOW marketplace users and technologies. More specifically, the finalisation of the marketplace users, having been performed as part of the requirements engineering phase, is also depicted in SHOW marketplace architecture. Hence, the possible main users interacting with the marketplace are *Guest Users*, *Product Providers*, *Product Consumers* and *Administrators/Aggregators*. Moreover, the tools and platform selection are also highlighted in the architecture. Indeed, the *programming languages*, the *Shuup* open-source platform, the *architectural pattern*, the *communication protocols*, the *data transferring,* and the *security aspects* are displayed in this final version. In chapter 6, all the aforementioned technical details will be analysed extensively.

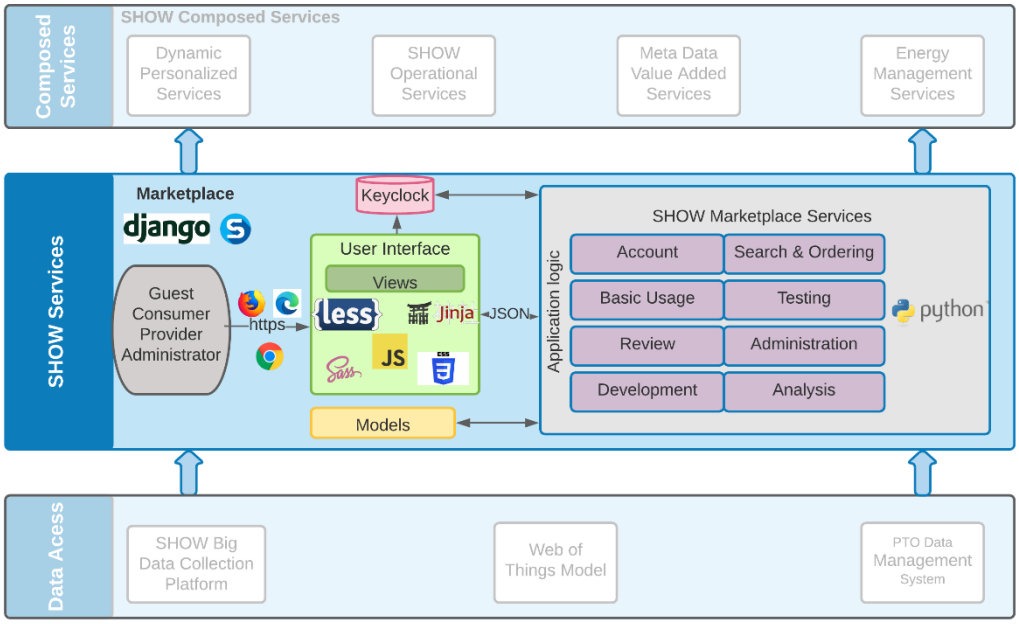


Figure 23: Final version of SHOW marketplace architecture

6.2 Django framework

Django was first introduced in 2003 to Lawrence news agency in Kansas [57]. It is a web framework that uses the Python [61] programming language to create web pages. The goal of the Django web framework is to create dynamic websites with great flexibility in the way of implementation, but also the ease of application development. Initially, the organisation proceeded to publish the Django source code in the BSD license in 2005. Passing three years forward in 2008, the Django Software Foundation was established to support and promote Django [58]. Few months later, the Version 1.00 of the framework was released, while the version that uses the marketplace of SHOW project is the latest 3.2.7 [59].

Django is a web framework since it is considered a collection of Python modules and is designed to create web applications. It is a free, open-source program and can be accessed by anyone [61]. A framework is defined as a set of software that organises the architecture of an application [57], simplifies the developers work, and minimises the allocated time to create the final result. Essentially, the utilisation of frameworks can accelerate the development procedure for the SHOW marketplace. A framework is adaptable to fulfil the various requirements and easily add functions, such as authenticating a user or deleting it if the administrator wishes. Thus, some features that are used repeatedly on a site can be automated, such as managing databases, managing users, changing user rights, and more.

The basic principle that the Django framework follows is the DRY (Don't Repeat Yourself) [65]. Therefore, a good design should include dynamic objects [66]. Dynamic objects are flexible and can be used for different reasons depending on the parameters accepted as input. This way the programmer avoids using repetitive code, thus making it easier to modify it when needed. Figure 24 shows the various constructions that the programmer can create from the same models without repeating his code.

Figure 24: Django models create separate constructs based on DRY principle [65]

As depicted in Figure 24 entities representing Django models can create HTML forms without the need for the programmer to define them elsewhere. Validation logic for enforcement entity values, administrative interface for managing entities, as well as DDL to create database tables representing the entities [65].

Django web framework can function as a database manager, but it is mandatory to apply two main commands. The first command is "*python manage.py makemigrations*" which will deduct the changes in models.py file and transmit data into MongoDB [63]. The second command is "*python manage.py migrate*" which is responsible for storing all changes in the database. To start the marketplace, the "*python manage.py runserver*" command must be given by the system administrator and then the local IP address of the running project is automatically provided. After this, the marketplace is ready to accept users' requests and respond to them as provided, without any additional intervention by the administrator.

Django framework uses Model View Controller (MVC) design structure [62]. Django was selected out of other web frameworks due to the required time for the marketplace deployment. Once compared to the rest of the frameworks, the development time is less due to the simplicity in Django's use. In the MVC design pattern, the framework will function based upon the model as a database, the view as a controlling functionality and the template will work as a user side for communication interaction [62]. Also, the Django MVC design structure [58] allows the developer to have complete control over the architecture, without applying repetitive patterns. The next subchapter describes in more detail the MVC design pattern.

6.3 Model View Controller (MVC)

The purpose of the MVC design is to separate the model representing data from the interaction between programmer and computer, to improve software development efficiency, ensuring the stability of the software development process and the scalability of the software in the later period, thus improving sustainability [64]. MVC consists of 3 features: the model which can be the data and the business plan, the presentation, which usually consists of HTML and is called View and finally, the Controller which is the layer that combines the model with the View [67]. In MVC each layer is composed of classes [68].

The Model is mainly used to encapsulate business logic and data model as mentioned above, to abstract the objects of applications, provide access from program functions to controllers, to encapsulate the structure and operation of an application function, accept data query requests from views and also to notify interested views when data change [64]. In a model, the database can also be represented, defining it as objects. Each object constitutes a table in MongoDB, while each object method constitutes the attributes of the table.

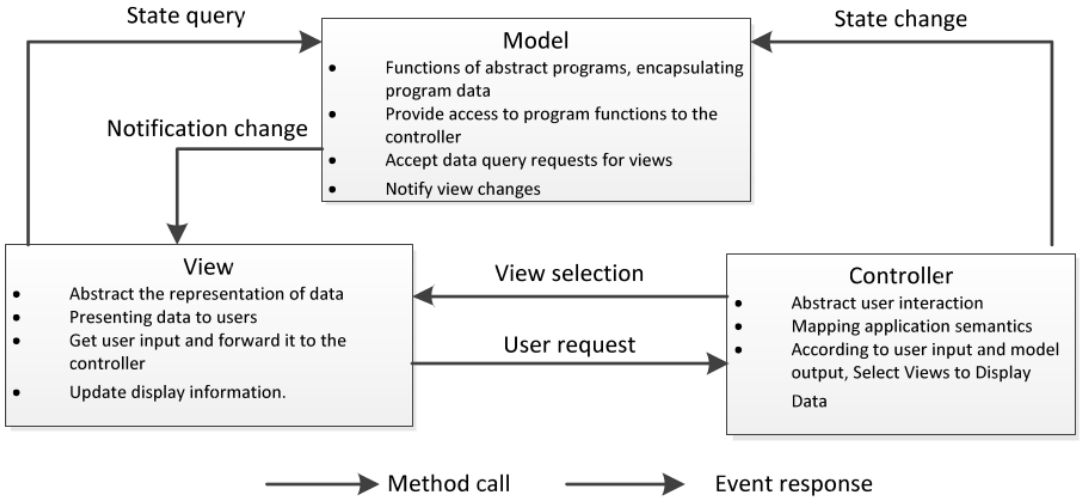


Figure 25: MVC design pattern [65]

The View is responsible for managing the graphical user interface. This level controls both the way of data presentation and the way of user interaction with the graphical interface [68]. At the same time, it receives as input the requests conducted by the user and forwards them to the controller. Finally, it updates the display information when the data received from the model is updated [64].

The controller contains all actions performed by the server in the background and are not visible in the client browser [58]. The controller converts the output drawn by the View into typical business events by recording external information. It then analyses the output of the View into actions that the model must perform (such as activating business logic and changing the state of the model). The View will also be notified by the controller in case of the model's update or modification. In this way, consistency between the View and the model is achievable [64]. The block diagram of the MVC is shown in Figure 25.

As shown in Figure 25, the three levels of MVC are separated from each other and have their own separate role. MVC was utilised for the SHOW's marketplace due to the fact of having many advantages such as its features, high reusability and ease of

application. The separation of views from the models can be changed without affecting each other, which significantly improves the software scalability, flexibility and portability from one server to another. Separating the models from the views means that the models can be expanded independently and modified without impact. Models, Views and controllers are isolated so that developers can design independently [64].

6.4 Database

The database selected for the SHOW marketplace is MongoDB [63]. MongoDB is a Non-relational Database Management System (DBMS) [69] and introduces an alternative way of scaling compared to other databases. Storing structured data has been a major issue since the beginning of the computer age. They were originally stored in restricted files, eventually stored in databases that the programmer first had to define one by one, but more recently a new methodology has emerged regarding the operation of databases that record data in key/value form, graphs and complex documents that have more than one level. The SHOW database was designed in such a way to allow the database administrator to represent relationships with structured documents but in a much more intuitive and flexible way [70].

The other advantage of the marketplace database is that it is built to be efficient, scalable and can easily modify the relationships between collections. Operating with MongoDB, the marketplace offers powerful solutions for storing and managing data such as operator's personal information, service provider's services, and related information, while the database scales automatically and easily as the marketplace grows and evolves.

Nowadays, storing high capacity data with low storage delay is relatively easy at low cost since the whole example of database design has changed. SQL database design focuses on keeping databases as small as possible through normalisation [70]. This fact creates a huge problem that is addressed within the design of the marketplace database. The computational cost for reading data is an issue, since it cannot be manageable, when normalisation has not been used on the data. Normalisation is necessary for merging the data into logical blocks, such as when service providers have to relate to the services they upload.

SHOW's marketplace has successfully addressed this issue, since it is designed to have high availability with its distributed architecture, while maintaining durable and stable changes. Specifically, as the marketplace is active, there are many nodes in the database, where each has a complete copy of the data kept in sync through replication [70]. Marketplace's database uses a variant of the master/slave template. In that template, a member of a copy set is designated as the primary member and is responsible for identifying any write operations, while the other members of the copy set are secondary and solely maintain a copy of the data. Secondary members are a kind of queue for recovery in case of data alterations and are ready to take on the role of primary if this is not available. In comparison with the MySQL [71] Database, it was observed that the response time of the MongoDB was 82.3% less, while when the collections increased the response time compared to the MySQL database decreased even more by 97.7% [69].

The marketplace database consists of four levels, as shown in Figure 26. The first level is called the application layer which is responsible for the data model schema and the data access logic. The second level is called MongoDB database server which contains the source code that process MongoDB commands, manages the distributed cluster and finally maintains indexes. The next level is the storage engine. The storage engine is part of the database but is also a distinct layer of code. In MongoDB, there are multiple options for storage engines, such as in-memory, RocksDB, and MMAP.

However, whichever storage mechanism is used is usually represented by the WiredTiger storage engine. In the case of the SHOW marketplace, the storage engine that was used is the in-memory, which is responsible for caching data in memory among other things. Finally, the last level is the storage subsystem. The storage subsystem is implemented in the operating system or storage hardware and not in the MongoDB software. On a simple single-server configuration, it is represented by the filesystem and the disk device's firmware [72].

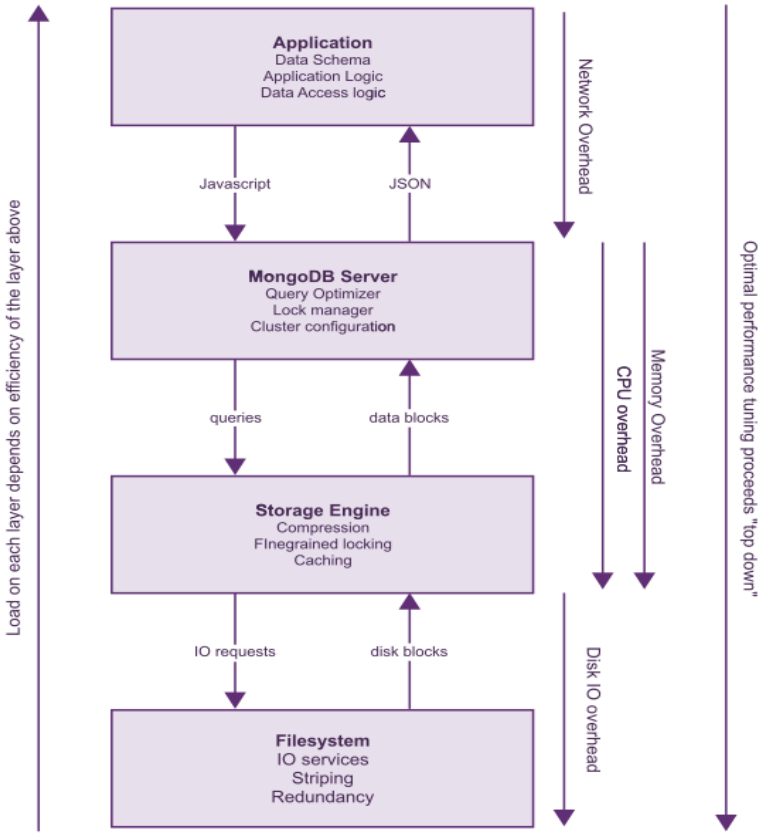


Figure 26: The four levels of the marketplace database [72]

6.5 User authentication

The primary mechanism used by the marketplace to determine whether a user is a service provider or an operator is the Keycloak [73]. Keycloak is an authentication server published in 2014. It is open-source software publicly available in JAVA and licensed under the Apache 2 permit for use by administrators and access controllers for web applications [74]. In this protocol there are three different entities: a) the user, b) the service provider and c) the authorisation server which holds and manages the identities.

Marketplace's Keycloak authenticates a user through OpenID Connect authorisation code flow. Specifically, for authentication of a user with this flow, during the authentication process on the login or in the register page, the marketplace sends the credentials via API created in python to the Keycloak server for user authentication. After authenticating the user, the application receives an identity token, which contains information about the user [75] on the occasion that the credentials provided by the user are valid. Otherwise, the Keycloak mechanism returned an error message via the API to the marketplace server, which contained detailed information about the user

request and an explanation about the inaccurate information provided by the user. Note that the above procedure takes place in the background of the two servers, i.e. on the marketplace server and the Keycloak server, and so the user does not need to intervene but inform about the result of the credentials provided. The following two subchapters describe how a user can register in the marketplace and how s/he can connect to it.

6.6 User Interface

6.6.1 Registration

This paragraph describes the process by which the user can register in the marketplace. As shown in Figure 27, the user to register in the marketplace of the SHOW project will first have to fill in some fields, some of which are mandatory and some optional. Required fields are the username, password, and password confirmation, while optional ones are the first name and surname, which can be filled in on the profile management page as presented in detail in Figure 29. Required fields are distinguishable by an asterisk inside the placeholder, while optional fields have no indication.

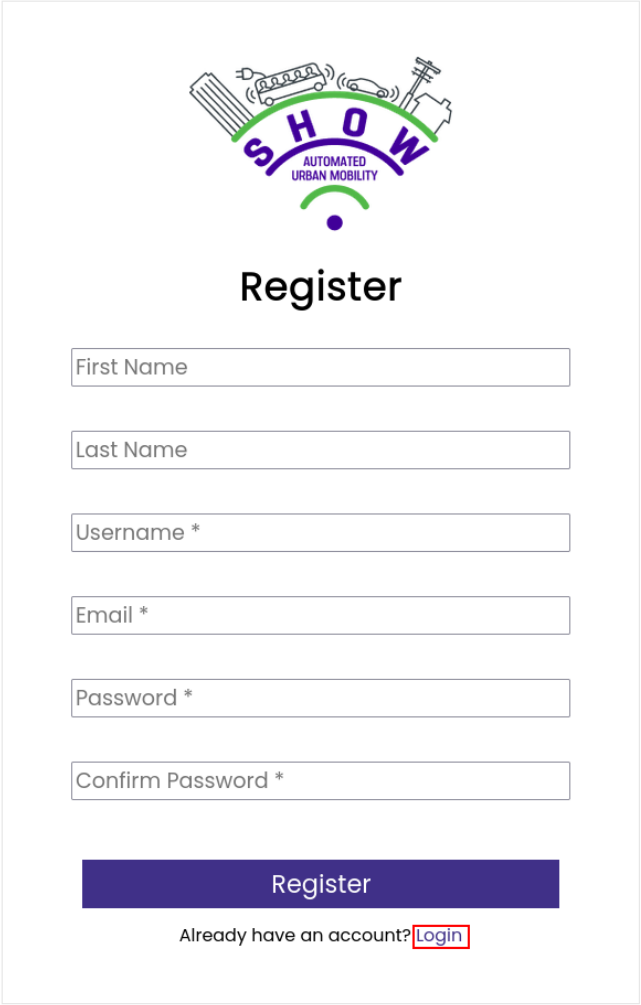


Figure 27: Marketplace registration page

The next two fields requested by the user concern personal information, such as his first and last name, which is protected according to the ISO/IEC 27000 [76] Protocol. Then, he must define the desired username. Note that the username that the user chooses cannot match the username of an already registered user in the database. In the next step, he will need to provide a valid email, to which an automatic request for confirmation will be sent. If the email is not valid, the registration process cannot proceed, and so the user will have to re-enter his email address. Finally, the user provides the password consisting of at least 8 letters, including alphanumeric characters, uppercase letters and numeric values. After confirming that the password provided is the desired one, the registration process is completed, and the user can now proceed to the login by clicking on the "*login*" button inside the red box as shown in Figure 27.

6.6.2 Login

The login procedure is as simple as shown in Figure 28. Initially, the user is asked to provide his credentials.

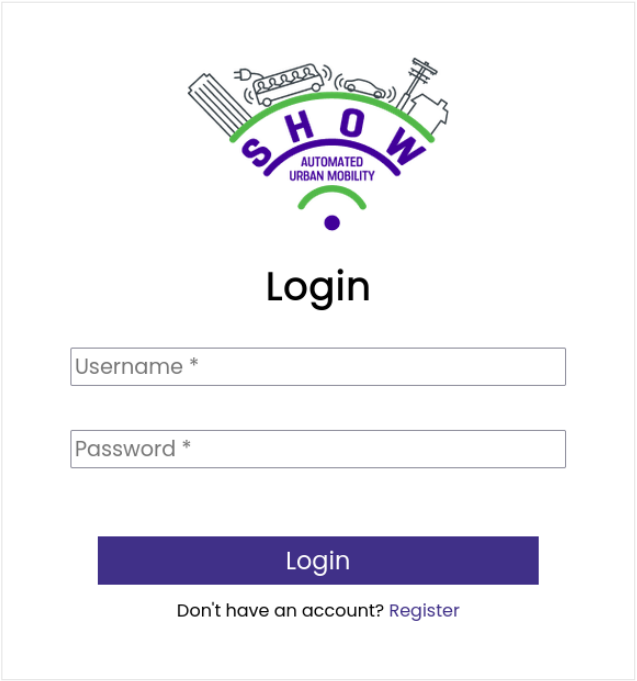


Figure 28: Marketplace login page

The first field concerns the e-mail address provided during the registration process. Alternatively, in case of the user's unwillingness to use their e-mail, the chosen username can be the identifiable element for the login. Finally, after providing the password, the page will redirect him to the profile panel to possibly fill in his personal information. There is also the option "*Forgot password*" which concerns the recovery of a password in case of lost or forgotten. Pressing this option, the system automatically sends a temporary password to the e-mail address registered for the user to gain access. After gaining access to the marketplace, they will be asked to change the temporary password to one of their choice.

6.6.3 User profile

After the successful user's login, the page shown in Figure 29 is visible to the user who can opt to fill in their personal information. The navigation bar is present at the top to allow the site's seamless navigation. The SHOW logo is clickable and redirects the

user to the main website of the project once pressed. It also includes the "*Market*" option in which all the available products contained in the marketplace are presented. Finally, a drop-down permits the user to change his personal information or log out of the marketplace by clicking on username button.

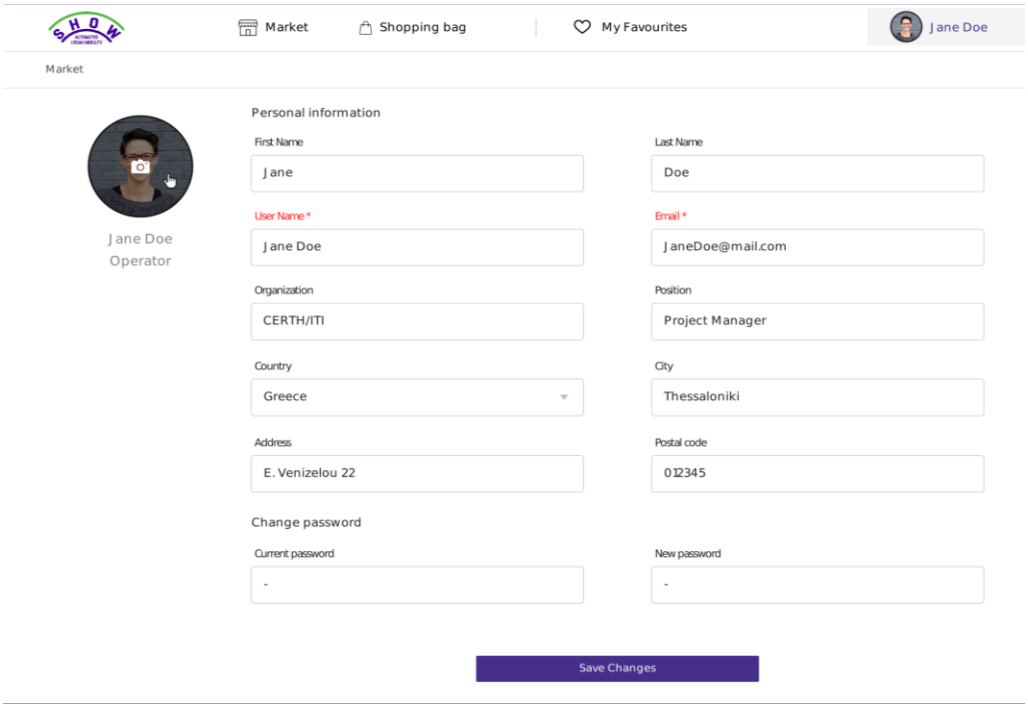


Figure 29: Marketplace user profile

Under the navigation bar, the user can select the photo that will represent him by clicking on the circular button. In the right part of Figure 29 two columns are presented concerning the: first name, last name, username, e-mail address, organisation, the position holding in the organisation to which belongs to, the country, the city, the address and the postal code. Additionally, the registered user has the ability to change the password after first completing the current password. In the current marketplace's version, the user chooses the role which is identical with the position held in the organisation to which it belongs to. Finally, the personalisation function can be based on fields such as "Country" and "Position" to define how the available products will be displayed.

6.6.4 Available products

The main function of the page shown in Figure 30 is to present all available products, serving also as the home page of the marketplace. The user can filter the products and representative examples are: the price range and the product providers, which determine which products appear. A panel with the filters is visible on the left side of the page to discover products based on the users' selected filters.

In the right part of the image, the available products are presented according to the filters selected by the user. The title and the price are presented together with the image of each product. At the top, the "*Sort by*" option appears providing the option to sort the products according to the ones that were added recently, the ones that have the lowest price, but also the ones that have the highest price.

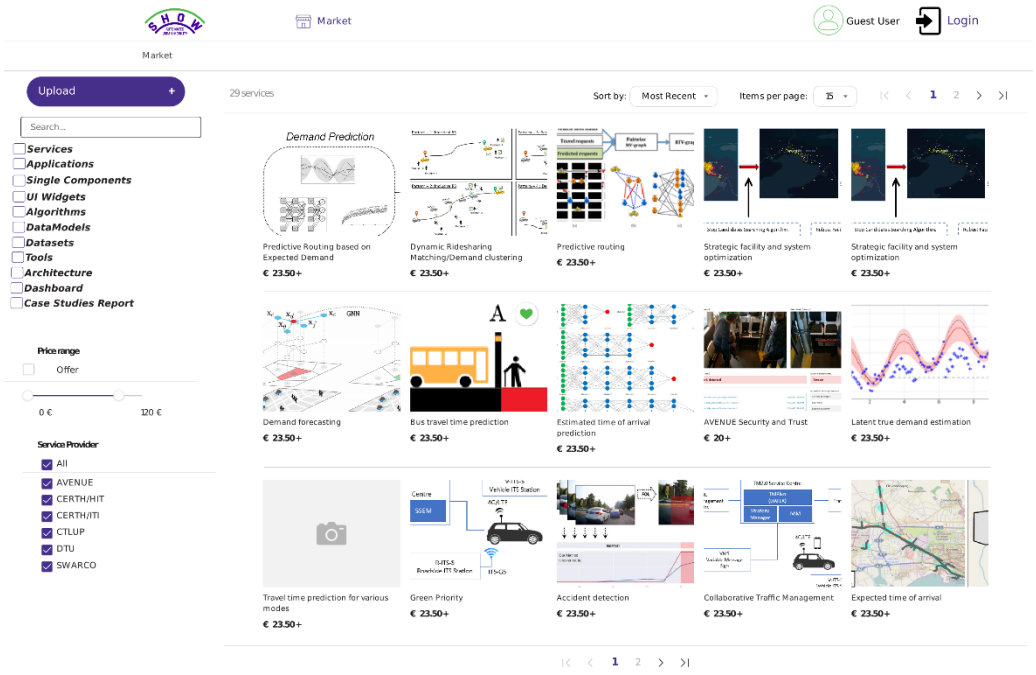


Figure 30: Marketplace product selection for a guest user

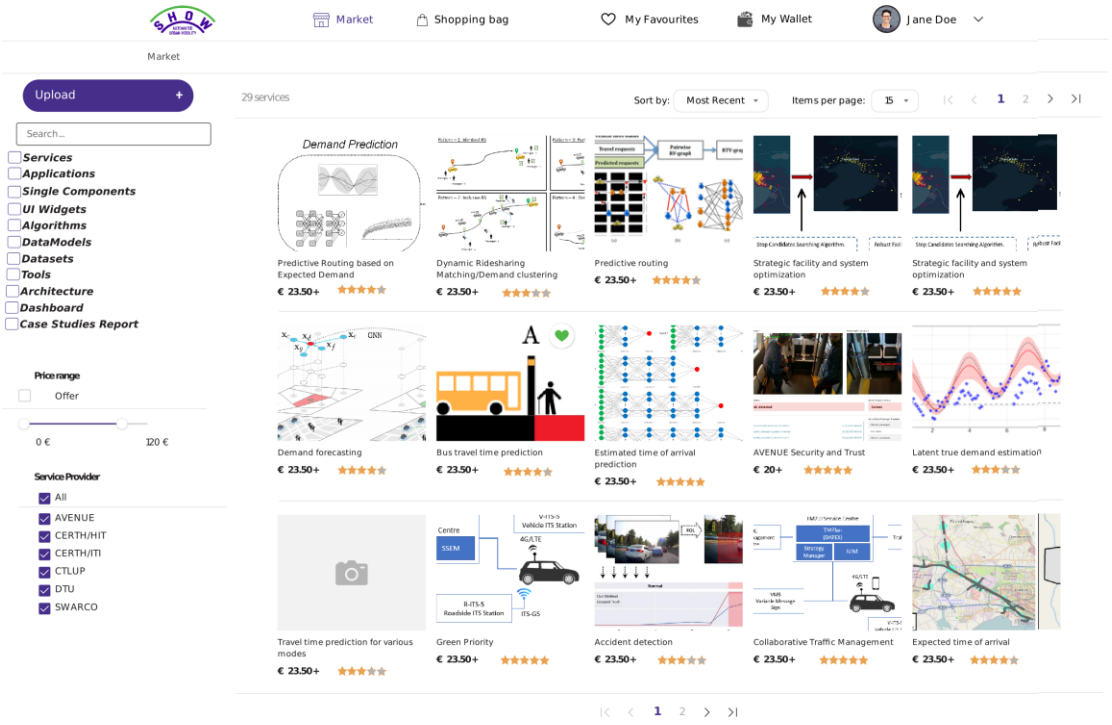


Figure 31: Marketplace product selection for a registered user

As can be seen from Figure 31, the main difference between a guest user and a registered one is that the latter has the ability to purchase a product, while the other has to register to render this feature available. As shown at the top of the image, a

registered user can put his favourite products on the list at any time and refer to the shopping list, a functionality that is unavailable to an unregistered user. Finally, registered users have at their disposal the "Upload" button on the left of the screen for uploading a product whenever it is necessary.

6.6.5 Upload a product

This paragraph presents how the users can upload a product. First, on the right side of Figure 32 they can select the category to which the desired product belongs, as described in Chapter 4.

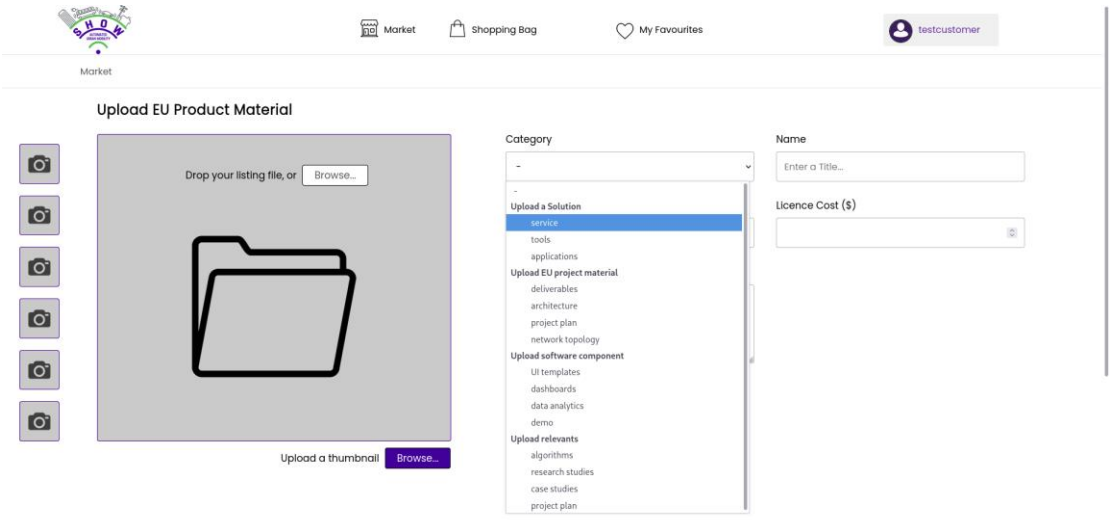


Figure 32: Marketplace category selection

The following fields in Figure 33 can change according to the Appendix 1 table depending on the category to be selected. In the “Browse” button up left, the users can choose the product they want to upload. It is also possible to upload more than one image representing the product on the left side.

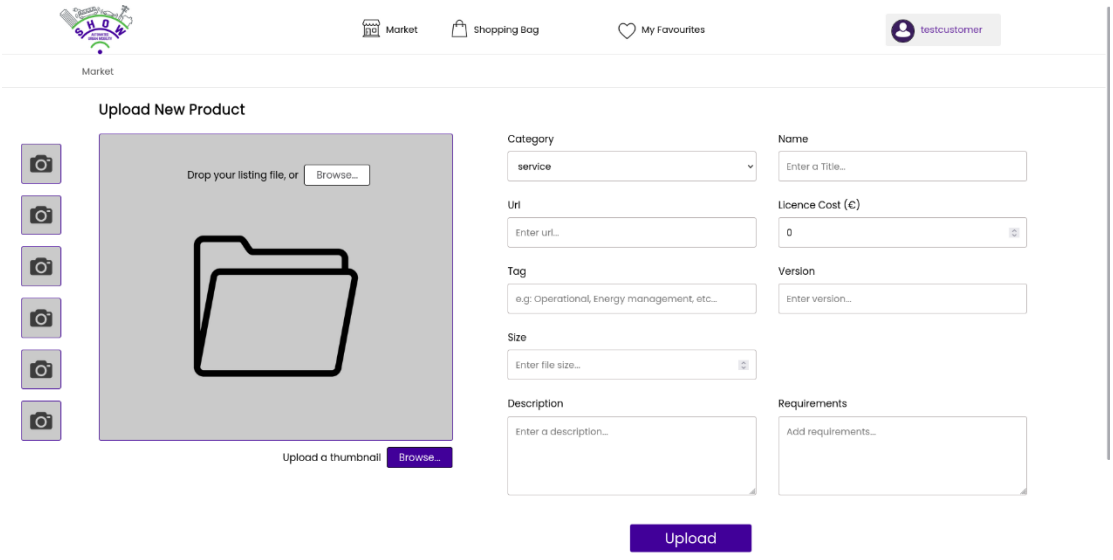


Figure 33: Marketplace upload a product example

The above figure is a page template that users will view prior to the uploading and once they select one of the products. In Figure 33, a user chooses to upload a service and according to this choice, eight different fields must be filled. The first field is related to the name of the specific service. The next fields concern the URL through which the service can be purchased in case of unwillingness to host it in the marketplace. The "Category" field defines the domain to which the proposed service belongs to, the "Version" field defines the current version of the service, and the “Size” field defines the total size of the service in MB. They can also give a description by entering it in the field labelled "Description". Finally, users can set in the "*Requirements*" field the prerequisites that customers have to fulfil in order to become the offered service functional.

6.6.6 Purchase a product

After the product selection, the users would like to download the product as described in the previous paragraph, hence they can now buy the product by clicking on the "*Buy*" button or adding it to the wish list as shown in Figure 34. They can also see the comments made by other users and their ratings on a scale of 1 to 5. It is important to mention that the user can only add their comment and rating only when the product is downloaded, otherwise this feature is not available.

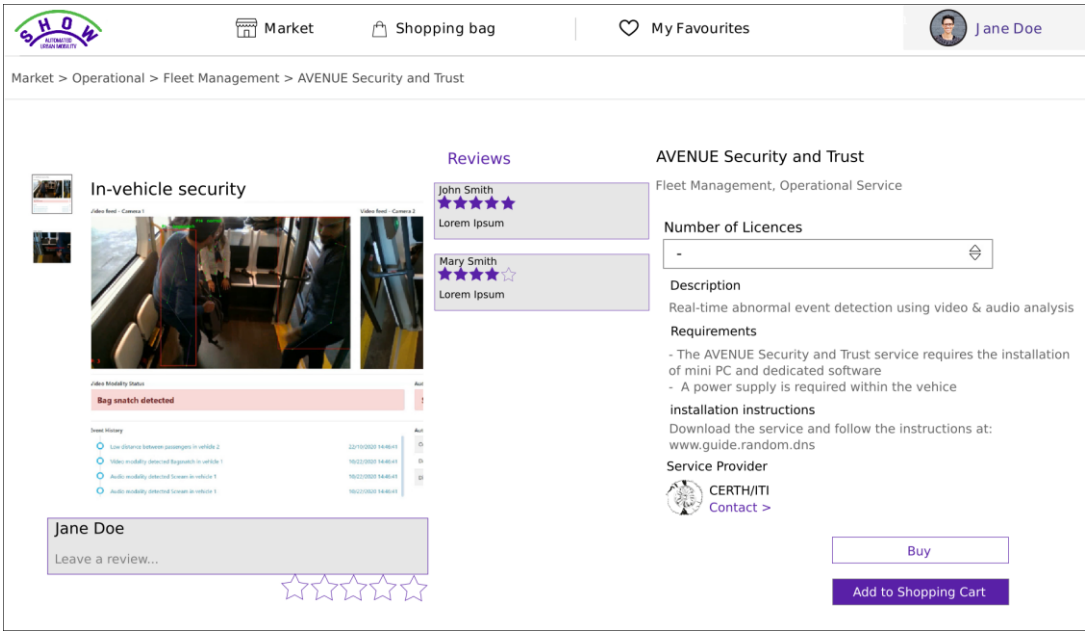


Figure 34: Marketplace purchase product

6.6.7 Checkout

This page presents the wish list in which the users can select the number of products they want to buy as shown on the left side of Figure 35 in the field "*Number of Licences*". At this point, it should be clarified that the checkout option is available as long as the product provider chooses to distribute the product through the market and not through another website via a declared URL. On the right side, the user selects a payment method to purchase the products from one out of the two separate options below the field "*Choose your payment method*". The price configuration is done dynamically depending on the price set by the product provider and the number of

products selected by the user. The process is completed by clicking on the "*Proceed to checkout*" button or the "*Keep Shopping*" option.

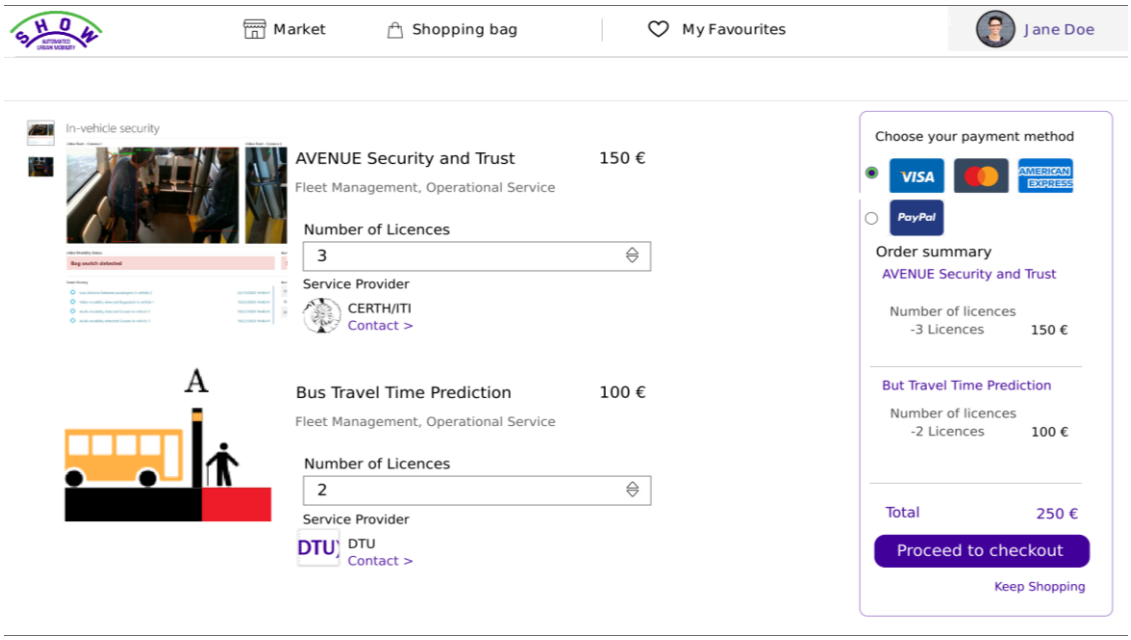


Figure 35: Marketplace checkout

7 Product Items

This chapter introduces the available and possible product items that will or could be hosted to SHOW marketplace. More specifically, this chapter consists of 4 sections, where in the first section the services that will be hosted -from three different sources- to SHOW Marketplace are described. Since this product item constitutes the most complete, and probably the most important type of product, for the marketplace ecosystem, the corresponding section occupies the most of this chapter. In the second section, the personalised dynamic services, which are not individual, but already developed services enhanced with the personalisation feature, are described. Finally, the third section and the fourth section describe briefly the available product items from the pilot sites, as well as the additional product items, that constitute an available pool for potential integrated services to the SHOW Marketplace.

7.1 Services

This paragraph describes the services to be hosted in the SHOW Marketplace. The variety of identities (chapter 4.1.2) allows numerous stakeholders to accommodate their services in the marketplace. The primary services sources are the pilot sites and service providers that participate in the project. Other European (or not) projects can have the opportunity to make their services available via the marketplace. The same is valid for any local, commercial or public provider of services.

The **SHOW project's WP5** activities will deploy operational services, and the SHOW Marketplace can act as the platform to publish and increase the adherence to these services. The operational services belong to the following groups: ***Fleet Management***; ***Planning***, ***Mobility Patterns***, ***Expected Time of Arrival***, and ***C-ITS and Traffic Management***. Moreover, the second source of services that will be hosted to the SHOW Marketplace is the services developed under **WP6** and more specifically, under ***Activities A6.3*** (Operational services), and ***A6.4*** (Energy management services). Additionally, the European projects in the CCAM sector constitute a valuable source, since they develop various and important services that can be published in the marketplace to significantly leverage the new mobility paradigm establishing high impact for this sector. Thus, the last source of hosted services for the marketplace is the **AVENUE project** [78], offering a variety of services from security-oriented approaches to environmental monitoring. Totally, 24 services that are the first to be integrated to the SHOW Marketplace are described.

It is worth noting that depending on the main source of the service and the maturity of the development, the extent of the service description differs considerably. Hence, services derived from WP5 (D5.2 [77]), and the AVENUE project are described briefly and the corresponding reference for further details is provided. On the other hand, services deployed within the context of WP6, are analysed and described to a greater extent, since these services constitute a prominent and core part of this deliverable.

7.1.1 Predictive Routing based on Expected Demand

7.1.1.1 Description

This service is envisioned by DTU and pertains to Public Transport routes' dynamic planning according to passenger demand forecasting. For example, autonomous shuttle fleets can benefit from the service to facilitate roaming to universities, amusement, parks, hospitals, etc. more efficiently, As another example, bus lines can operate based on this service to dynamically respond to sudden surges in passenger demand, e.g., around exceptional events that draw big crowds.

7.1.1.2 Technical Requirements & Operations

The service consists of two main components: 1) Artificial Intelligence methods for predicting the dynamics of passenger demand, 2) algorithms for demand-based route optimisation. Advanced Machine Learning is implemented for the first component to model and predict the dynamics of demand via probabilistic distributions. Sequentially, the second component uses the predictive distributions to choose between several routing strategies. A hypothetical scenario is the possible lower demand prediction than usual, in which the routes may be optimised to use lower frequency and save energy and reduce costs. Conversely, if the demand is predicted to be significantly increased in some areas, the optimisation procedure may temporarily allocate more resources to these areas.

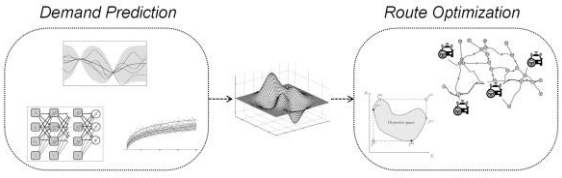
The service is amenable to a broad scope of input data from various sources. Such data include: 1) historical and real-time traffic data collected from road sensors and data aggregators, 2) shared transportation data, 3) (optional) travel surveys. The service then produces the following output: 1) a passenger demand's complete distribution, 2) a list of optimised routing options.

7.1.1.3 Development & Deployment

The service and its methodology are developed and evaluated via several empirical case studies. Additionally, the methodology is now applied in a real-world project and upon finalization will be integrated into the marketplace.

7.1.1.4 Integration to Marketplace

Table 3: Predictive Routing based on Expected Demand

Characteristic	Description
Name	Predictive Routing based on Expected Demand
Description	The service pertains to the dynamic planning of Public Transport routes based on the passenger demand forecasts.
Tags	Predictive Routing; Demand Prediction; Public Transport; Artificial Intelligence; Machine Learning
Technical requirements	Data on traffic, shared transportation, and travel surveys
Version	0.1
Price	TBD
Service/URL	TBD
Image	
Size	Under development
Publisher	DTU

7.1.2 Dynamic Ridesharing Matching (Demand clustering)

7.1.2.1 Description

Contemporary urban trends are overcrowded cities and a constant rise in congestion events. Sharing transport services emerge as a sustainable alternative to reduce cars and mitigate costs and trip delays. Ridesharing (RS) is one of the best alternatives, as it promotes the sharing spirit between commuters sharing a trip that satisfies their needs in a flexible, comfortable, and affordable way with the use of cars. Those services can aid in matching agencies or service providers, but it is necessary to have an available fleet for commuters to use as a transport mode.

7.1.2.2 Technical Requirements & Operations

The module uses the commuters' requests and data of the available fleet, consisting of both private drivers and service fleets, to perform matching and routing operations. Regarding the technical aspect of the service a major issue generally, in RS services is the ridesharing pattern the system follows. The patterns can be divided into four categories that impact the service in different ways, as each pattern requires a distinct algorithmic approach. Namely, the four categories are: *Pattern-1* (identical ridesharing), the trip's origin and destination are identical for the participants, *Pattern-2* (inclusive ridesharing): this case refers to passengers with origins and destination accommodated in a driver's route, resulting in at least two stops during the trip, *Pattern-3* (partial ridesharing): the participants have identical origin or destination, and the problem is handled as one-to-many or many-to-one RS, *Pattern-4* (detour ridesharing): this is the case where the driver shifts his route to match the path to serve all passengers in the best possible way.

7.1.2.3 Development & Deployment

The underlying algorithm of the service was developed and evaluated under a real-world project. Under the same project, tests in real-world cases were conducted with the objective being for companies' employees to have an alternative transportation method to public transport in fear of COVID-19 infection. The aforementioned service will be integrated also into the marketplace.

7.1.2.4 Integration to Marketplace

Table 4: Dynamic Ridesharing Matching

Characteristic	Description
Name	Dynamic Ridesharing Matching
Description	The service implements advanced machine learning to dynamically allocate commuters with similar profiles to shared trips.
Tags	Ridesharing; Taxi; Matching
Technical requirements	Python and corresponding libraries, Historical data
Version	1.0
Price	TBD
Service/URL	TBD
Image	-
Size	-
Publisher	CERTH/HIT

7.1.3 Predictive Routing

7.1.3.1 Description

This service can benefit various stakeholders like OEMs, drivers, and travellers, as it provides routes for transportation. For example, automated vehicle fleets can have greater flexibility for their business operations. Predictive routing is a service providing the optimal routes for a vehicle to follow to transport commuters. The optimal route takes into account different variables prior to producing the suggestion.

7.1.3.2 Technical Requirements & Operations

Initial research in the literature indicated numerous methods that could be applied for computing the optimal route. Examples of the encountered methods included graph theory methods and deep learning algorithms. Historical and real-time data on the trips were part of the state-of-the-art solutions. A prevalent dataset is the Manhattan taxi trips for basing the prediction. Furthermore, dimension reduction methods are applied to the data.

7.1.3.3 Development & Deployment

The service is under development, and experimentation, to determine the most appropriate method to be employed. Inspired by the literature, public open datasets are candidates for the experimentation. This service will be developed under the SHOW project and will be integrated into the marketplace upon finalisation.

7.1.3.4 Integration to Marketplace

Table 5: Predictive Routing

Characteristic	Description
Name	Predictive Routing
Description	The service will provide the optimal route for transportation.
Tags	Prediction; Optimal route
Technical requirements	Historical and real-time data
Version	0.1
Price	TBD
Service/URL	TBD
Image	TBD
Size	Under development
Publisher	CERTH/ITI

7.1.4 Strategic Facility and System Optimisation

7.1.4.1 Description

There is a recent surge in the deployment of shared transportation services to alleviate congestion issues in crowded cities. Demand Responsive Transport (DRT) and sharing services such as Bike Sharing (BS) strive to provide an alternative to the fixed-route public transport. Both systems share some strategic decisions that should be made during the planning procedure. The most vital is to allocate the stops in places inside the operating network to satisfy most of the demand. Finally, the fleet size, the

routing algorithm, and the vehicle's capacity are vital input parameters to calculate the planning process.

7.1.4.2 Technical Requirements & Operations

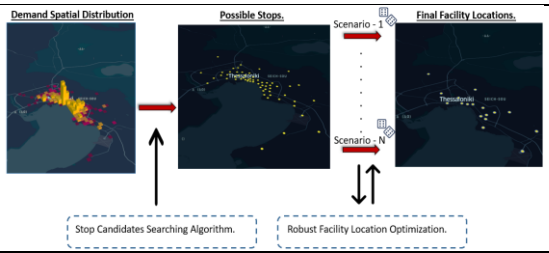
The service will have a number of options like stops' optimal location determination for DRT or BS services; optimal fleet size for DRT systems; DRT vehicle's optimal capacity size or rebalancing vehicles in case of BS; and more. Firstly, locations will be calculated from disaggregated trip data and used as stop candidates. With stop candidates determined, the methodology relies on different demand scenarios to robustly solve facility location mixed-integer programs. While the algorithm solves VRP instances to determine the optimal number of vehicles, the algorithm tests the different capacity parameters. Finally, different operational programs and constraints are applied to approximate the optimal solution during simulations.

7.1.4.3 Development & Deployment

This service is applied under European funded project and in different EU cities to help authorities decide the optimal service and the strategic features that must have. The service will be hosted in the marketplace after the development.

7.1.4.4 Integration to Marketplace

Table 6: Strategic Facility and System Optimisation

Characteristic	Description
Name	Strategic facility and system optimisation
Description	This service is an alternative to traditional fixed-route transportation and offers functionalities in DRT and BS.
Tags	Operational services, Planning
Technical requirements	Historical traffic data
Version	0.1
Price	TBD
Service/URL	TBD
Image	
Size	Under development
Publisher	CERTH/HIT

7.1.5 Bike Sharing Demand Forecasting

7.1.5.1 Description

Predicting the demand of the bike-sharing systems is one of the important elements of the operations management procedure. Indeed, a strong prediction algorithm can lead to better inventory management and a more accurate rebalancing. Thus, accurate

predictions of bike trips allow services to deliver as many users as possible and increase their profits and quality.

7.1.5.2 Technical Requirements & Operations

The objective of the service is the accurate prediction for bike-sharing. Trips can be treated as time series of m previous steps, and the objective is to determine each station's demand for the next k-steps. Numerous machine learning methods are applicable to the problem. In the past few years, architectures like Graph Convolutional Networks (GCNs) have been applied with a significant impact on the system's overall performance. The service will provide the option of selecting between GCNs and a classical ML approach depending on problem requirements and data availability.

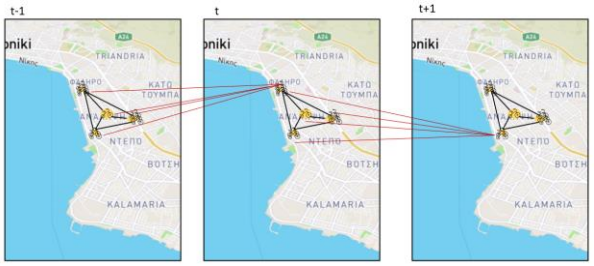
The service needs data of arrivals and departures in bike-sharing systems in a time-series format. For data aggregated in specific time intervals (e.g. every 30 minutes), the output corresponds to that same time interval. The time interval can be regarded as a tuning parameter, as intervals smaller in duration have increased noise compared to larger timesteps that miss information due to the aggregation.

7.1.5.3 Development & Deployment

This algorithm has been implemented during ibikeshare project where CERTH/HIT developed a complete dashboard for monitoring and decision making for the bike sharing company of Thessaloniki. The respective service will be integrated into the marketplace after the appropriate configuration.

7.1.5.4 Integration to Marketplace

Table 7: Bike Sharing Demand Forecasting

Characteristic	Description
Name	Bike Sharing Demand Forecasting
Description	This service will predict the demand for bikes to aid in the allocation of bikes in the network.
Tags	Bike Sharing; Demand Prediction; Machine Learning; Rebalancing
Technical requirements	Bike positions and timestamp, historical data
Version	1.0
Price	TBD
Service/URL	TBD
Image	
Size	-
Publisher	CERTH/HIT

7.1.6 Demand Forecasting

7.1.6.1 Description

This service is trying to assist the problem of urban transportation by predicting future demand using state of the art deep learning techniques and external factors such as weather and event data.

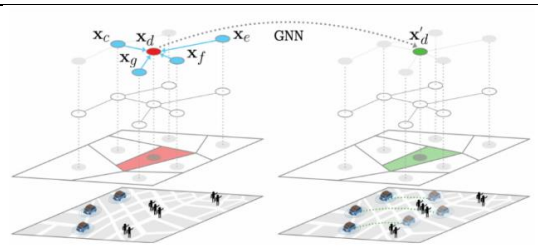
7.1.6.2 Technical Requirements & Operations

The proposed model uses historical data of the transport service along with possible external factors that could ameliorate the prediction ability of the model. The suggested algorithm is based on Graph Neural Networks (GNNs) and Recurrent Neural Networks (RNNs). By using GNNs it is possible to model complex spatial correlations and by applying RNNs the modelling of temporal correlations is becoming available. Finally, the output of the model is the demand prediction at the desired level (e.g. hourly, daily etc.).

7.1.6.3 Development & Deployment

The algorithm has been validated in a public open dataset and the respective service will be hosted to SHOW Marketplace.

7.1.6.4 Integration to Marketplace

Characteristic	Description
Name	Demand Forecasting
Description	A service assisting in urban transportation by predicting future demand at different levels.
Tags	Operational Services, Forecasting, Mobility Patterns
Technical requirements	Historical traffic demand, weather data
Version	0.1
Price	TBD
Service/URL	TBD
Image	
Size	Under development
Publisher	DTU

7.1.7 Latent True Demand Estimation

7.1.7.1 Description

Transport demand is highly correlated with the supply, particularly for shared transport services with limited availability. As the observed demand cannot be larger than the available supply, historical data tend to be a biased version of the latent. Classical

predictive models of demand may not account for this inherent distinction resulting in biased true demand and therefore decreasing the prediction's effectiveness. This service will predict the latent demand, as it is vital for planning and decision-making processes in transport services.

7.1.7.2 Technical Requirements & Operations

A broad method for building models is included in the service. The method accounts for the bias in observational data that are more reliable in reflecting user behaviour. Specifically, the method builds a censored likelihood function applicable to any pattern of user demand. The latent demand depends on a Gaussian process (GP) that permits the structured quantification of confidence due to the implicit adoption of a fully probabilistic approach from the GP.

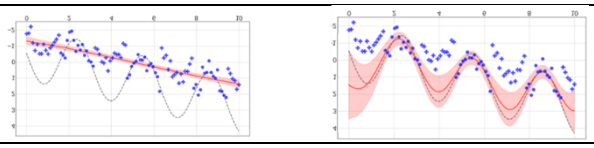
The service is applicable to input data from various sources, publicly available or offered by municipalities. The models for predicting latent demand usually require: 1) historical patterns of user demand, and 2) historical records of supply. Specifically, latent demand models will use the information on when the service historically ran out of supply. As a result, latent demand models will reasonably predict the unobservable process of service demand.

7.1.7.3 Development & Deployment

This methodology has been tested and evaluated on several experimental case studies. Results have recently been published in several scientific venues. The corresponding service will be hosted in the marketplace upon finalisation.

7.1.7.4 Integration to Marketplace

Table 8: Latent True Demand Estimation

Characteristic	Description
Name	Latent True Demand Estimation
Description	Estimated true mobility demand for a service based on censored historical data (e.g. cases where real demand could be a lot higher, but the observed demand is limited by the available supply in the area at the time)
Tags	Operational Services, Demand Estimation
Technical requirements	Historical and real-time data of requests (origin, destination, timestamp)
Version	0.1
Price	TBD
Service/URL	TBD
Image	
Size	Under development
Publisher	DTU

7.1.8 Public Transportation Demand Forecasting

7.1.8.1 Description

This service deals with the issue of public transportation contributing to the planning of itineraries and the optimal fleet management by offering the mandatory tools performing demand forecasting.

7.1.8.2 Technical Requirements & Operations

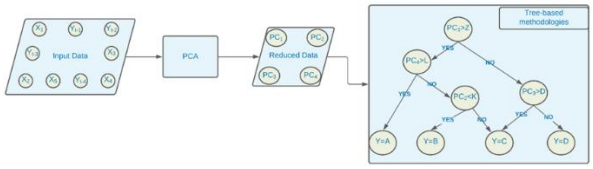
Public transportation demand forecasting receives as input historical demand of passengers and predicts the demand for the next period using statistical (Principal Components Analysis) and machine learning methodologies (Regression Trees, Random Forests and Boosting).

7.1.8.3 Development & Deployment

The service has been validated until now in a public dataset and the results have shown quite particular accuracy of the methodology. Regarding the next steps, these include the tuning of the machine learning parameters, the validation of the algorithm and the development of the service.

7.1.8.4 Integration to Marketplace

Table 9: Public Transportation Demand Forecasting

Characteristic	Description
Name	Public Transportation Demand Forecasting
Description	The goal is to provide a service that will perform demand forecasting of public transportation to enhance fleet management.
Tags	Operational Services, Mobility patterns, Machine Learning, Forecasting
Technical requirements	Historical data of passengers demand
Version	0.1
Price	TBD
Service/URL	TBD
Image	
Size	Under development
Publisher	CERTH/ITI

7.1.9 Expected time of Arrival (ETA)

7.1.9.1 Description

The scope of this service is to calculate the Expected Time of Arrival (ETA) in the transportation sector. By providing information to commuters (or other system's users), the service facilitates the task of providing an accurate vehicle arrival to a certain location.

7.1.9.2 Technical Requirements & Operations

A machine learning algorithm utilizes a Neural Network and predicts the vehicle's location by predicting arrival points, represented by polygons. The data needed include historical spatial data (i.e. location and timestamp), as well as real time location data to detect the vehicles' current location.

7.1.9.3 Development & Deployment

This service was developed for an EU funded project and tested in real life conditions, to compute the ETA of trains and inform taxi cabs accordingly. After the appropriate configuration will be hosted to the marketplace.

7.1.9.4 Integration to Marketplace

Table 10: Expected Time of Arrival

Characteristic	Description
Name	Expected Time of Arrival.
Description	The goal is to provide a service that will perform arrival time and location prediction.
Tags	Estimated Time of Arrival, Operational Services, Machine Learning, Forecasting.
Technical requirements	Historical data of vehicle's location and time.
Version	1.0
Price	TBD
Service/URL	TBD
Image	
Size	-
Publisher	CERTH/HIT

7.1.10 Bus Travel Time Prediction

7.1.10.1 Description

This service aims to provide accurate and reliable travel time predictions in PT networks. The goal is to create an attractive service, competitive with other modes of urban transportation.

7.1.10.2 Technical Requirements & Operations

This service's approach consists of a multi-output deep learning NN for bus time travel prediction, enabling the discovery of patterns with higher complexity. This is achieved by a combination of convolutional and Long Short Term Memory (LSTM) layers, being able to capture non-static spatio-temporal correlation and allowing in this way the model to generalise patterns in predictions across both time and space. To produce confidence prediction intervals, Bayesian and quantile regression methods are used.

The data used in this service include historical observations of bus travel time, information about weather conditions, incidents and temporary changes in the network.

7.1.10.3 Development & Deployment

The evaluation of the aforementioned method was done empirically and compared to other popular approaches. The deployment was concluded and funded by a Copenhagen regional PT authority.

7.1.10.4 Integration to Marketplace

Table 11: Bus Travel Time Prediction

Characteristic	Description
Name	Bus Travel Time Prediction
Description	The goal is to create an accurate, reliable, attractive and competitive service for travel time prediction in Public Transport networks.
Tags	Expected Time of Arrival, Bus Travel Time, Operational Services, Machine Learning, Forecasting.
Technical requirements	Historical data of vehicle's travel time, weather conditions, incidents and temporary changes.
Version	1.0
Price	TBD
Service/URL	TBD
Image	TBD
Size	-
Publisher	DTU

7.1.11 Travel Time Prediction for Various Modes

7.1.11.1 Description

This service provides estimations of travel times for a number of transportation modes, specifically railways, buses and trams, as well as underground transportation. The service takes into consideration factors such as different times and rush hours, along with internal (e.g. temporal door blocking) and external (e.g. weather) contingent factors.

7.1.11.2 Technical Requirements & Operations

This service operates as follows. For long-term forecasting, it identifies point-to-time routes, homogenous periods by day, time slice within the day and vehicle types, in order to calculate mean and variance of the data, while by averaging the (historical) data it forecasts the link travel time for a route, within a time slice of a specific day for a vehicle. The real-time forecasting uses a one step ahead approach, meaning for a specific time-slice, it forecasts the variable's time for the next time slice.

7.1.11.3 Development & Deployment

This service is developed as a part of an Italian national project and has not been tested yet. The required data for this service include the type of vehicle, day and time-slice, the road section and historical data for travel time.

7.1.11.4 Integration to Marketplace

Table 12: Travel Time Prediction for Various Modes

Characteristic	Description
Name	Bus Travel Time Prediction
Description	The goal is to create an accurate, reliable, attractive and competitive service for travel time prediction in Public Transport networks.
Tags	Expected Time of Arrival, Bus Travel Time, Operational Services, Machine Learning, Forecasting.
Technical requirements	Historical data of vehicle's travel time, weather conditions, incidents and temporary changes.
Version	1.0
Price	TBD
Service/URL	TBD
Image	TBD
Size	-
Publisher	CTLUP

7.1.12 Estimated Time of Arrival Prediction

7.1.12.1 Description

This service aims to provide a solution to a critical problem in smart cities, such as the prediction of the estimated time of arrival of vehicles. Applied in both public transport and demand-responsive transport, the service provides real-time information to commuters about the status of their request, taking Mobility-as-a-Service into consideration.

7.1.12.2 Technical Requirements & Operations

The current implementation of the proposed methodology uses sophisticated machine learning algorithms belonging to the Gradient Boosting Trees family, using different algorithms for the approach. The required data for this service are vehicle's location and timestamp, while also taking into consideration other vehicle data (speed, acceleration) and external (weather, type of day) factors.

7.1.12.3 Development & Deployment

This service is developed within the SHOW ecosystem. Therefore, it is still in development stage and will be tested with data from pilot sites. After the development completion, the service will be hosted on the marketplace.

7.1.12.4 Integration to Marketplace

Table 13: Estimated Time of Arrival Prediction

Characteristic	Description
Name	Estimated Time of Arrival
Description	The goal is to provide a service that will perform prediction of the time of arrival, while providing constant updates about the time remaining for the vehicle to arrive to a stop.

Characteristic	Description
Tags	Estimated Time of Arrival, Operational Services, Machine Learning, Gradient Boosting, Forecasting
Technical requirements	Historical data of vehicles location and timestamp.
Version	0.1
Price	TBD
Service/URL	TBD
Image	TBD
Size	Under development
Publisher	CERTH/ITI

7.1.13 Collaborative Traffic Management

7.1.13.1 Description

The specific service is able to foresee the activation of collaborative actions upon the occurrence of certain pre-defined conditions on the current status of the road network. This service could lead to improved traffic flows and quality of the network, reduced probability of an accident and increased driving comfort.

7.1.13.2 Technical Requirements & Operations

A road operator defines a Traffic Management Plan (TMPlan). When the pre-defined conditions are met, the road operator initiates the TMPlan. Moreover, the management in such conditions may imply the need for one or more operations to be executed by several Traffic Management or Service Provider Centres. Information regarding the implementation of the required actions/services are shared, among relevant Traffic Management Centres and the vehicles, to restore a high Level of Service.

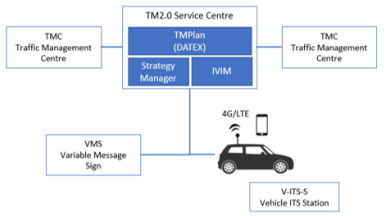
7.1.13.3 Development & Deployment

The service is under development and has been tested only in controlled environment. The evaluation is based on the respective performance metrics and when the service is ready will be integrated into SHOW Marketplace.

7.1.13.4 Integration to Marketplace

Table 14: Collaborative Traffic Management

Characteristic	Description
Name	Collaborative Traffic Management
Description	The goal of this service is to foresee collaborative actions, thus improving traffic flows and network quality
Tags	Operational Services, Traffic Management.
Technical requirements	Historical data in DATEX II, ETSI CAM 1.4.1, ETSI DENM V1.3.1
Version	0.1
Price	TBD
Service/URL	TBD

Characteristic	Description
Image	
Size	Under development
Publisher	SWARCO

7.1.14 Traffic Light Assistance

7.1.14.1 Description

This service delivers Green Light Optimal Speed Advisory (GLOSA) and a Time-To-Green (TTG) service to the connected vehicles. Connected and automated vehicles that approach a connected intersection will receive information on the optimal speed for the given intersection. The service also provides information of the phase state of the traffic light combined with the remaining phase duration.

7.1.14.2 Technical Requirements & Operations

The Central ITS Station is intended to be connected with the traffic lights of an intersection and receives feedback from the controllers about the current status of the traffic light. The prediction algorithm is running and produces a Traffic Light Forecast. The forecast is used as a message for the TLA Application. Based on this information, the TLA Application is able to calculate the GLOSA and TTG countdown taking also into account the current positions of the vehicles.

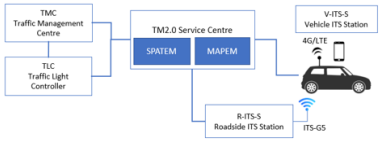
7.1.14.3 Development & Deployment

The Traffic Light Assistance service has been tested in several cities through plenty EU projects. The testing procedure has been implemented by the installation of the service on smartphones and tablets.

7.1.14.4 Integration to Marketplace

Table 15: Traffic Light Assistance

Characteristic	Description
Name	Traffic Light Assistance
Description	The service aggregates data regarding the geometry of intersection, the real time data of the vehicle and the light signal phase in order to assist in traffic management.
Tags	Operational Services, Traffic Management, Traffic Light.
Technical requirements	MAP (topology) Extended Message, ETSI TS 103 301, Signal feedback from the connected traffic lights.
Version	0.1
Price	TBD
Service/URL	TBD

Characteristic	Description
Image	
Size	Under development
Publisher	SWARCO

7.1.15 Green Wave Activation

7.1.15.1 Description

This service is utilised in order to ensure the prioritisation of the public transportation and safety vehicles (e.g., ambulance, fire brigade, etc.). These types of vehicles could obtain the ability to traverse a pre-defined intersection having the highest priority by the system.

7.1.15.2 Technical Requirements & Operations

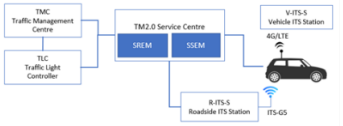
An authorised vehicle sends a message (SREM) to the traffic infrastructure (e.g., R-ITS-S, TMC) requesting the traffic light signal priority or the signal pre-emption. As a sequence, the infrastructure will judge and decide if the request has been approved or rejected. In case that the request has been granted, the required actions could be done by the infrastructure.

7.1.15.3 Development & Deployment

The specific service has been tested during European projects under different specifications (i.e., using CAM/SPAT messages). In terms of the SHOW project, it will be deployed utilizing another set of messages being compatible with the requirements of C-ROADS.

7.1.15.4 Integration to Marketplace

Table 16: Green Wave Activation

Characteristic	Description
Name	Green Wave Activation
Description	This service supports prioritisation of public transport and public safety vehicles to traverse a signalised intersection as fast as possible or using a higher priority than ordinary traffic participants.
Tags	Operational Services, Traffic Management, Traffic Light, Safety Vehicles
Technical requirements	ETSI SREM V1.3.1
Version	0.1
Price	TBD
Service/URL	TBD
Image	

Characteristic	Description
Size	Under development
Publisher	SWARCO

7.1.16 Accident Detection

7.1.16.1 Description

This service calculates the deviation between the actual and predicted position of the vehicles taking into account that an unexpected event affects, mainly, the location and their speed. The recognition of accidents and, generally, the non-normal events equips the system with the significant ability to face up with incidents existing in the current traffic status.

7.1.16.2 Technical Requirements & Operations

The model consists of the Future Object Localisation that includes the Bounding Box Prediction, Ego-Motion Cue and Missed Objects. The proposed strategies of accident detection are the Predicted Bounding Boxes - Accuracy, the Predicted Box Mask and the Predicted Bounding Boxes – Consistency. The majority of the already known models are based on the classification of all possible events on the road. On the other hand, the proposed service detects the statistical deviation between the predicted and actual position of the vehicle. Therefore, the model will be trained only with a large-scale dataset of normal driving videos.

7.1.16.3 Development & Deployment

The model is still under development. This service will be implemented during the SHOW project.

7.1.16.4 Integration to Marketplace

Table 17: Accident Detection

Characteristic	Description
Name	Accident Detection
Description	This service detects the deviation between the actual and predicted position of the vehicles taking into account that an unexpected event affects, mainly, the location and their speed of them.
Tags	Operational Services, Traffic Management, Anomaly Detection.
Technical requirements	Dashboard camera, Routing Data
Version	0.1
Price	TBD
Service/URL	TBD
Image	TBD
Size	Under development
Publisher	CERTH/ITI

7.1.17 Logistics Service for Freight Replenishment and Final Delivery in the Supply Chain

7.1.17.1 Description

The supply chain's vital operations are the freight replenishment of the delivery hub's warehouse and the last-mile delivery. Two distinct operations will deliver the above functionalities in the proposed service. While the traditional approach is composed of human interaction between the different supply chain stakeholders, the digitalisation of processes in the delivery services is possible with the deployment of a Logistics as a Service (LaaS) platform facilitating the digital interaction via the platform.

7.1.17.2 Technical Requirements & Operations

The scenario is a last-mile distribution service concerning the main depot and some peripheral distribution hubs in an urban area. The platform will deliver functionalities such as online purchases through an online platform; scheduling vehicles journeys from the cloud platform; the number of vehicle optimisation with a DARP algorithm; VRPTM algorithm for delivery hubs provisioning. The use case of the cargo service, which CTLup will implement as LaaS for publication to the SHOW Marketplace, is intended to be used only by 3PL operators, for the part of the supply chain relative to the distribution hub replenishment and last-mile final delivery.

The Main Depot has a big Warehouse, where the stock consists of the products to be delivered. The provision of products is made from the main depot to the delivery hubs. The main depot is located outside the urban (suburban) area, while the delivery hubs are located in an internal urban zone. The final part of the materials delivery process is the trip from the delivery hub to the destination (final customer), commonly referred to as last-mile delivery. Traditional cargo vehicles and electric vehicles can bridge the last-mile delivery. The frequency of journeys for last-mile deliveries is more than one per day, dependent on the number of shipments per route, the capacity of vehicles, and time windows requested.

7.1.17.3 Development & Deployment

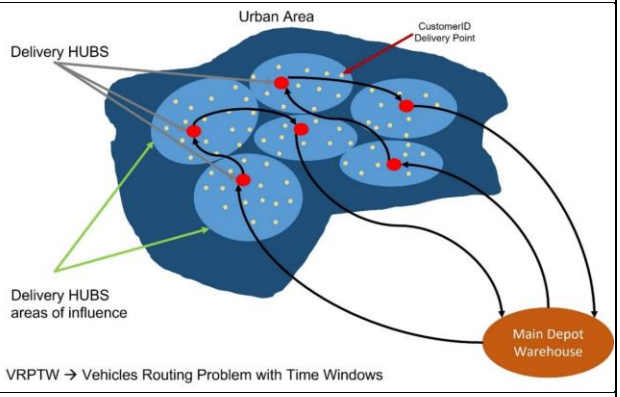
The LaaS cloud platform will be implemented as a web-based application with reserved access and a secure channel (SSL). It will be accessible by an APP on the Marketplace and will operate for the users (3PL) to handle the last-mile distribution service for its e-commerce customers.

The Master Data that will be set in the application are the delivery hub's storage capacity, the vehicles' quantity and types used in transportation, the capacity of these last-types of vehicles in terms of volume and weight, the number and the types of vehicles used for the journeys from the delivery hubs to the final customers, and the complete products data (Product ID, Volume (m3) with the three linear dimensions: Length (m), Width (m), Height (m), Net weight (without packaging) and gross weight of the products).

Generally, the application by the system of machine learning techniques on historical data will provide advanced functionalities concerning the following procedures: Products demand prediction and Products' stock-out prediction of the delivery hubs. Finally, this service will be tested during SHOW project and also, will be hosted at SHOW Marketplace after the completion of the validation process.

7.1.17.4 Integration to Marketplace

Table 18: Logistics Service for Freight Replenishment and Final Delivery in the Supply Chain

Characteristic	Description
Name	Logistics Service for Freight Replenishment and Final Delivery in the Supply Chain
Description	This service will aid the supply chain with delivery hub management and last-mile delivery.
Tags	Operational services, Logistics, LaaS, DARP, VRPTM, Delivery hubs
Technical requirements	Historical data of passengers demand
Version	0.1
Price	TBD
Service/URL	TBD
Image	
Size	Under development
Publisher	CTLUP

7.1.18 Predictive Energy Management

7.1.18.1 Description

Predicting the forthcoming energy demand of vehicles is pivotal in setting up vehicles' smart energy management. This work's objective is to provide a service that reads in-vehicle and environmental data. Two distinct machine learning models utilise the data for training, one for vehicle control and one for velocity prediction and benchmark.

7.1.18.2 Technical Requirements & Operations

The learning framework has been applied to a Plug-in hybrid electric vehicle. The vehicle powertrain has been equipped with a conventional combustion engine on the front axle, two different electric machines, one per axle, a 6-speed gearbox to couple the engine to the wheels, a 2-speed gearbox to connect the rear e-machine to the wheels and a large battery to drive long enough to cover at least 5 times the most common route for a passenger car use case (5-6km). The measured vehicle performance metrics used in practice are:

- the total energy consumption, as the sum of the battery and fuel energy contents consumed over a given driving scenario;

- the total energy cost, as the sum of the cost to charge the battery as much as it has been depleted during the trip and to refill the tank;
- The total CO_2 emissions, from Wheel to Wheel, as considered a good standard for a proper and fair assessment of a PHEV.

In this study, the energy cost has been used as the primary objective to define the vehicle control policy. The two major challenges in benefiting from this kind of architecture are:

- Battery State-of-Charge (SOC) management.
- Large state-action space to optimise

The electric power can usually be conveniently utilised as per the higher overall efficiency than the combustion engine, despite the greater price of the energy unit. However, this advantage is not always clear. The energy policy's goal is to detect the right spots to use the battery energy and prevent vehicles from entering a very inefficient area for the engine to work in due to the battery's complete depletion.

The second obstacle is how to handle the state-action space of the problem. A simplified model has been developed to make the problem more tractable, by neglecting engine/battery thermal effects on the overall powertrain efficiency. Nevertheless, the space to explore is still the combination of the system states and actions. States are the discretised SOC levels, the engine state (on/off), the e-machine states and the gearbox gears. Actions are the number of power-split between the front and rear axles, torque-split between the engine and the e-machine on the front axle and the rear transmission gear.

The creation of an appropriate dataset for training the vehicle policy control necessitates data from numerous trips. The set of these trips needs to contain sufficiently diverse driving scenarios that should be statistically representative of the target use case. Each trip is a log file of the vehicle velocity at fixed time steps. In the case of GPS coordinates availability, the dataset augmentation is available by looking up to an open-source geographical service, such as Open Street Map (OSM), to gather speed limits or road types. There are occasions that measurements include the vehicle altitude value. This information is valuable in estimating the slope profile after a proper filtering phase. The full dataset will be then fed to both the vehicle control and velocity prediction blocks.

The full powertrain (transmissions, e-machines, engine, battery) has been modelled to estimate the energy consumption due to both fuel and electric energy. The model has been validated in previous studies. The assumption here is that the selected vehicle is similar enough to the actual vehicles used to gather the velocity profiles. It is planned to use the vehicle data of the SHOW Linköping test site for training purposes.

The vehicle control unit is responsible for determining the power distributions across the different components and the gear speed ratios that can optimise the objective function of the problem (e.g., energy consumption, energy cost). This problem is described as constrained multi-stage optimisation and belongs to the optimal control theory. The main challenge is to derive an optimal control policy to as many conditions as possible.

The study follows an imitation learning structure, where the control model learns to map specific states to ground truth actions provided by the teacher. The teacher feedback is computed via a Dynamic Programming (DP) algorithm over the full set of driving profiles. DP searches for the optimal control policy over a specific driving profile by minimising the objective function and ensuring that each hard system constraint is met at any time. The DP technique is powerful as it guarantees the optimal solution.

The core idea is to develop a lightweight model that can only rely on a fraction of the information available to the DP without significantly jeopardising the overall performance.

Over the years, many rule-based controllers have been developed and introduced, but they require high human effort to be developed and tuned. A machine learning-driven model that exploits the DP outcomes to find the mapping between states and actions in an automatic fashion could help to achieve a higher complexity of the controller in a shorter span of time.

The control model is built as a multi-class classification problem, where the system states and previous actions are used as features to determine the class of action to take at the present time step. The model output size, i.e. the number of classes, is the Cartesian product of the two discrete sets of possible power-splits and gear speed ratios. The control model employs an increasing number of features to predict the correct action. These features are categorised to easily analyse afterwards the impact of each category on the overall model performance. The features' categories are:

- Past actions: torque-split and gear speed ratio values over the last n time steps
- Actual state: vehicle velocity and power demand, battery SOC
- Short-term future states: median and variance velocity values over the next n time steps
- Long-term future states: distance to the end of the route, distance share over urban, rural and highway segments

The model architecture (random forest, support vector machine, shallow neural networks) has been kept as a hyperparameter to investigate. An extensive analysis for the model selection will be performed at the end.

The second block aims at vehicle velocity forecasting over a finite horizon, given the velocity history and the meta-data gathered from the map data. The model outcome will be solely used as an additional feature to enhance the control model. Due to the complex nature of any forecasting problem where the signal to study is affected by several factors (traffic, pedestrians, driver unpredictable behaviour, traffic lights, weather conditions), the framework was defined in a flexible way. The model output can be either the sequence of speed values over the next n time steps or a set of aggregated speed values as median, variance and 25-75 percentiles.

The model employs a sequence-to-sequence structure, where each sequential block is a recurrent neural network that reads in a time series of features to generate the output. The second block uses this representation and the available information over the next time steps to generate the representation of the future. If the model output is the sequence of future speed values, the final layer would use the future representations to return the velocity at each time step, otherwise, an aggregated representation is transformed to give the statistical data, as previously described.

As many studies have shown, the velocity forecast based on the past velocity only performs poorly over medium-term horizons. The pre-processing phase is fundamental to augment the sequential model input content and eventually improve the model performance.

7.1.18.3 Development & Deployment

The performance of the ML-based control model strategy has been assessed against four different benchmarks. In addition, three cases of the ML control strategy have been defined and compared to estimate the real benefit of further functionalities, as

the velocity forecast. The different benchmarks and models are compared with respect to the available metrics in this study. The four benchmarks are:

- Engine-only mode
- Charge depleting/Charge sustaining, where the battery is fully exploited till the lower bound is reached and the hybrid mode is then activated to complete the trip.
- Charge blending, where the battery is gradually discharged over the full trip.
- DP output

The first and last cases represent the low and high bounds over the performance scale. The three ML model variants are:

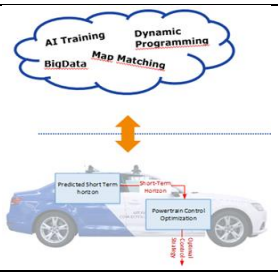
- No velocity prediction outcome is used as feature
- The actual model velocity predictions are used as features
- The ground-truth velocity values are fed as features.

This set of different benchmark results would help to fully understand the potential of the proposed approach and eventually where to mostly focus further efforts on. This ranking system will be also used to get insights of the effects of additional features, model architectures and the dataset size.

Finally, there is close cooperation and alignments of the partners participating in the development of energy management services with regard to a) the availability of vehicle data (FZI), b) the test case related to optimisation of a bus stop manoeuvre (FEV) and c) mapping solution developed by ARTIN.

7.1.18.4 Integration to Marketplace

Table 19: Predictive Energy Management Service

Characteristic	Description
Name	Predictive Energy Management Service
Description	The goal is to provide a service that reads in-vehicle and environmental data. Two distinct machine learning models utilise the data for training, one for vehicle control and one for velocity prediction and benchmark.
Tags	Energy Management, Machine Learning, Data Analytics, AI Prediction, Hybrid Energy, Distribution, Dynamic Programming.
Technical requirements	Vehicle Performance Data, Map with topography
Version	0.1
Price	TBD
Service/URL	zip
Image	

Characteristic	Description
Size	Under development
Publisher	AVL Software and Functions GmbH

7.1.19 Battery Optimisation

7.1.19.1 Description

Battery optimisation service will be based on the inductive charging technology and on the prediction of energy consumption. The inductive charging use case is studied under SHOW project, since it appears to have many advantages in comparison with the utilisation of the classical one wired charging. Through this procedure the power transportation could be optimized. Moreover, inductive charging could ensure a longer lifetime and smaller size of the battery. Furthermore, this type of charging is proved to be easier than the traditional charging. Finally, in addition to the aforementioned advantages, inductive charging is safer and can be performed more frequently.

7.1.19.2 Technical Requirements & Operations

There are three main groups for inductive charging in terms of the transmitted distance: near-field, medium-field and far-field. According to the literature, inductive charging is feasible only in the near and middle fields, since the far-field charging is characterized by extra challenges in the electromagnetic domain. In more detail, the major physic principles that are valid for these two feasible groups (the near and middle field) are magnetic inductance and resonance.

Battery Optimisation service could ensure the required conditions in order for the lifetime of the battery to be maximized. Succeeding in the optimal usage of battery, except for the economic benefits (also mentioned previously), a more eco-friendly approach is also established. Taking into account the results of the research community, the battery should not be charged in the time that is entirely empty. Additionally, the charging of the battery to the maximum could lead to reduced lifetime. Hence, the optimized range in which the charging status should be is between 30-80%. For this reason, a respective prediction and optimisation service is more than useful.

As mentioned previously, the service will be based on the prediction of energy consumption and more specifically, this problem could be characterised as a time series problem that will take into account the location of the vehicle, the mean energy consumption and the positions of energy stations. The current status of the battery will be an input for this service and an alert will be created once an internal threshold is achieved which means charging is required. The time of arrival for the next two or more energy stations along with the respective required energy will be predicted from the service. After the completion of this procedure, the service will decide which station is optimal for charging considering the estimated level of the battery for all the available choices.

7.1.19.3 Development & Deployment

The service is still under the design phase, when the technical requirements are ready the development phase will initiate. After the validation of the service during the SHOW project, this will be integrated into SHOW Marketplace.

7.1.19.4 Integration to Marketplace

Table 20: Battery Optimisation

Characteristic	Description
Name	Battery Optimisation
Description	The service takes into account the status of the battery and predicts when the optimal charging should be done.
Tags	Energy Management, Battery Optimisation, Inductive Charging
Technical requirements	Inductive Charger, Battery Data, Routing Data
Version	0.1
Price	TBD
Service/URL	TBD
Image	TBD
Size	Under development
Publisher	CERTH/ITI

7.1.20 Enhance the Sense of Security and Trust

7.1.20.1 Description

Given the absence of the bus driver in AVs and the increased threat of terrorism in European cities, this service aims to address the new reality formed in autonomous shuttles mobility infrastructures. This service will handle incidents of passengers' abnormal behaviour, like petty crimes and more, according to standard procedures. The main concerns include first aid if required, feeling of discomfort being all alone in the bus at night, vandalism, bag snatching, indoor fighting, unaccompanied luggage and lack of authority figure to keep passengers (e.g. school kids) calm. This service provides a timely, accurate, robust and automatic detection of petty crimes or misdemeanours and the assistance of authorised end-users towards the identification of any involved offenders.

7.1.20.2 Technical Requirements & Operations

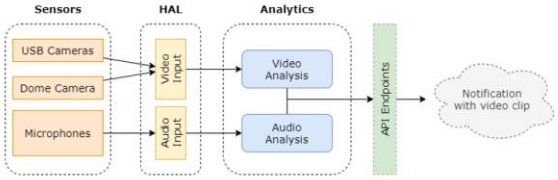
Petty crimes targeted for identification by the sensors include petty theft like bag snatching and pickpocketing, vandalism, aggression, illegal consumption of cigarettes, public intoxication, simple assault and disorderly conduct. This service involves camera and microphone sensor support, specifically embedded USB and IP cameras, along with USB and IP microphones, so no specialised protocols and/or ports are required. During this service, the sensors connect to the Hardware Abstract Layer and then the data is converted and formatted to be passed to the video and audio analytics algorithms. The algorithms used for video analysis are mainly LSTM Classification via Pose Estimation and for audio analysis mainly consist of Convolutional Neural Networks (CNN).

7.1.20.3 Development & Deployment

This service is developed and tested for the needs of the AVENUE project and after the finalisation will be hosted in the SHOW Marketplace.

7.1.20.4 Integration to Marketplace

Table 21: Enhance the Sense of Security and Trust

Characteristic	Description
Name	Enhance the Sense of Security and Trust
Description	The goal is to handle incidents of passengers' abnormal behaviour, like petty crimes and more, according to standard procedures.
Tags	Security, Abnormal Behaviour, Bag Snatching, CNN, LSTM
Technical requirements	USB and IP cameras and microphones
Version	1.0
Price	TBD
Service/URL	TBD
Image	
Size	-
Publisher	CERTH/ITI

7.1.21 Automated Passenger Presence

7.1.21.1 Description

This service aims to address a fundamental problem of operators' services, which is the routes' scheduling according to vehicles' occupation and knowledge of the number of people currently on board. Furthermore, enhancing passengers' in-advance knowledge can assist them in planning as relevant information are the available seats and the free space on the shuttle. The introduction of an automated system aims to achieve real-time and accurate passenger detection, consisting of passenger count and vehicle occupancy calculation.

7.1.21.2 Technical Requirements & Operations

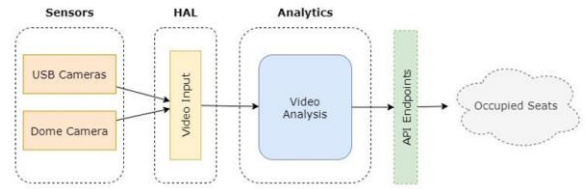
This service provides a video analysis of the vehicle interior using the onboard camera to specify the vehicle occupation, the vehicle's free space, and counting people on-board. The on-board camera enables the automatic assessment of space occupation. The passengers can also determine whether to board a vehicle or not with the aid of space occupation assessment for different cases. This service only involves video analysis, using the same camera sensors as the "Enhance the sense of Security and Trust" service, i.e. embedded cameras and USB/IP cameras. This service is based on YOLO, a state-of-the-art real-time object detection system that applies a single neural network to the full image.

7.1.21.3 Development & Deployment

This service is developed and tested for the needs of the AVENUE project and after the finalisation will be hosted in the SHOW Marketplace.

7.1.21.4 Integration to Marketplace

Table 22: Automated Passenger Presence

Characteristic	Description
Name	Automated Passenger Presence
Description	The goal of this service is the routes' scheduling according to vehicles' occupation and knowledge of the number of people currently on board.
Tags	Operational services, Passenger Presence, Passenger Counting, YOLO
Technical requirements	USB and IP cameras
Version	1.0
Price	TBD
Service/URL	TBD
Image	
Size	-
Publisher	CERTH/ITI

7.1.22 Follow my Kid/Grandparent

7.1.22.1 Description

This service increases the autonomy of non-fully autonomous people (Kids, Grandparent(s), disabled people etc.), by ensuring family members and/or caretakers for their family member's safety while commuting using public transportation. Moreover, the service can boost the non-fully autonomous people's confidence in using public transport. Surveillance using sensors such as cameras, microphones, and bus-specific smart software maximises the passenger's feeling of security and the actual level of security. This service's main beneficiaries include passengers feeling discomfort travelling alone at night-time, parents ensuring their children reach their destination safely, and caretakers tracking people with health issues.

7.1.22.2 Technical Requirements & Operations

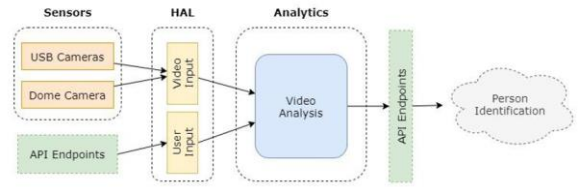
The service proposes a successful solution that allows guardians to follow the journeys of more vulnerable people since the guardians can check the trip using a dashboard or mobile app, receive notifications via mobile app, add people to their "guarded" list, and share trips/position, and ETA with others. This service utilises the same equipment mentioned above, which is embedded and USB/IP cameras. The algorithm used for the implementation of this service is Siamese Neural Networks to score similarities.

7.1.22.3 Development & Deployment

This service is developed and tested for the needs of the AVENUE project. After the finalisation of the service, it will be hosted in the SHOW Marketplace.

7.1.22.4 Integration to Marketplace

Table 23: Follow my Kid/ Grandparent

Characteristic	Description
Name	Follow my Kid/ Grandparent
Description	The service is dedicated to automated public transport. It provides the operator with the ability to detect if a kid/grandparent remains standalone in the vehicle.
Tags	Operational services, In-Cabin Monitoring
Technical requirements	A camera sensor (RGB) and an IoT Gateway are mandatory.
Version	0.1
Price	TBD
Service/URL	TBD
Image	
Size	-
Publisher	CERTH/ITI

7.1.23 Shuttle Environment Assessment

7.1.23.1 Description

This service automatically preserves the environmental conditions in the vehicle interior at satisfactory levels as there is no other option due to the driver's absence. Minimum acceptable conditions and comfort necessary for safe transportation include good air quality, acceptable odours, absence of smoke, and the viability of the whole service since lacking these standards could significantly discourage potential passengers. Monitoring the environmental conditions could enhance the user experience and safety during the trips.

7.1.23.2 Technical Requirements & Operations

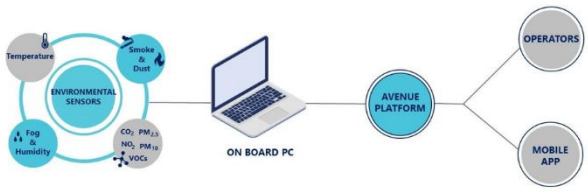
This service's main functionalities are the passengers' discouragement in lighting a cigarette, regulating the air conditioning system in high and low temperatures, and regulating potentially harmful gases such as CO, CO_2, and NO_2. The main goal is to preserve the shuttles' environment comfortable for passengers' majority. For this service, environmental sensors can be deployed (CO_2, NO_2, temperature, humidity fogging prevention and dust particles concentration sensors), along with smoke sensors and smoking detection sensors.

7.1.23.3 Development & Deployment

The aforementioned service constitutes an operational service of the AVENUE project in which it is developed and tested. After the completion of the service, it will be hosted in the SHOW Marketplace.

7.1.23.4 Integration to Marketplace

Table 24: Shuttle Environment Assessment

Characteristic	Description
Name	Shuttle Environment Assessment
Description	This service automatically preserves the environmental conditions in the vehicle interior at satisfactory levels.
Tags	Vehicle environment, comfort, smoking, smoke sensors.
Technical requirements	Environmental sensors (CO2, NO2, temperature, humidity fogging prevention and dust particles concentration sensors), smoke sensors and smoking detection sensors.
Version	1.0
Price	TBD
Service/URL	TBD
Image	
Size	-
Publisher	CERTH/ITI

7.1.24 Smart Feedback System

7.1.24.1 Description

The objective of this service is to allow the passengers to give feedback in a promptly and effortlessly manner to the shuttle operators, as it is crucial to monitor passenger satisfaction considering the services and transportation. In the case of a safety driver, they can communicate with the commuters, and their presence becomes the communication channel between the operators and the passengers. However, in a case where the safety driver is removed, the knowledge of the satisfaction levels of the end-users becomes the paramount priority, hence enhancing traveller assistance. This service aims at allowing passengers to give their feedback as easily as possible about liking/disliking the service experience. This can be acquired by suggesting the passengers to give a hand gesture to one of the cameras inside the shuttle.

7.1.24.2 Technical Requirements & Operations

This service utilises the cameras inside the shuttle, which are embedded, namely USB and IP cameras. The approach adopted uses a Single Shot Detection (SSD) and CNN using VGG16 for feature extraction.

7.1.24.3 Development & Deployment

This service is developed and validated experimentally within the AVENUE project and will be hosted in the SHOW Marketplace.

7.1.24.4 Integration to Marketplace

Table 25: Smart Feedback System

Characteristic	Description
Name	Smart Feedback System
Description	This service aims at allowing passengers to give their feedback as easily as possible about liking/disliking the service experience.
Tags	User Feedback, Single Shot Detection, VGG16, CNN
Technical requirements	USB/IP cameras.
Version	1.0
Price	TBD
Service/URL	TBD
Image	TBD
Size	-
Publisher	CERTH/ITI

7.2 Personalised Dynamic Services

Personalised Dynamic Services do not constitute individual services that will be integrated to the marketplace as the 24 afore-described services, but they are services enhancing some existing services with the personalisation feature. As described in the respective deliverable (D5.2 [77]), the list of AI services that will be deployed on the trials of the test sites are the following:

- Fleet management (at operational level)
- Service design (at planning level)
- Mobility patterns identification
- Prediction of travel time/time of arrival for public transport modes
- C-ITS and traffic management

Personalised Dynamic Services may call the services that are tailored to each specific user based on their wants and needs. Personalisation functionality will be developed on the:

- Fleet management & service design
- Mobility patterns identification and
- Prediction of travel time/ time of arrival for public transport modes.

In the following subsections, the Personalisation functionality that will be added to each of the above services is introduced and analysed.

7.2.1.1 Fleet management and service design – Personalised

Fleet management constitutes a key procedure for transport companies. It helps these companies to optimise costs, enhance efficiency, and maintain a high level of service for the travellers, while making optimum use of the available resources.

A fleet management service with personalised functionality will be more effective for the end user of the system, who will be the fleet operator in the SHOW cases. The

personalisation functionality will reside on both the design of planning and the operation modes as well as the operation of the fleet management service itself.

With regards to the service design, the personalisation will be related to:

- the kind and format of the input data to be provided, which include among other the type of service to be offered, the expected demand and the available resources
- the visualisation and format of the outputs provided by the service design service, including the location of the stops, the frequency of the service and the way it will operate among many others

From the fleet management point of view, the user may decide:

- when the routes for the next day will be calculated
- how these will be presented to the fleet manager and the drivers
- if the service will estimate the expected pick-up time to declare it to the travelers
- what is the flexibility in terms of this pick-up time in order to satisfy the requests of the travelers
- if the system will be able to accommodate real-time requests during the execution of the routes

7.2.1.2 Mobility patterns identification including demand prediction – Personalised

Being able to predict the demand for urban transport is crucial for properly managing fleets and increasing user satisfaction. In the traffic modelling domain this is a non-trivial task due to the dynamic and interconnected nature of the urban environment. It should be mentioned that this service is a prerequisite of the previously mentioned Fleet management and service design services.

The personalised mobility patterns allow the operator to select the presentation structure of the actual demand. The user may select between:

- heat map - a graphical representation of the demand where values are depicted by colour
- list of bus stops, categorised by their potential demand (from the most to the least used stop)
- groups per hour per day – different groups according to the demand of the users.

Furthermore, the user may personalise the update timing of the results. For example, the user may ask for a night update to have the time to crosscheck the results or for an update just before the start of the route to have the most up-to-date values. The user may select the structuring of the results between the following demand categorisation on:

- a specific bus
- a specific bus stop
- a specific bus route

7.2.1.3 Prediction of travel time/ time of arrival for public transport modes – Personalised

Estimated Time of Arrival (ETA) is a critical procedure for smart cities, as provides information to passengers and consumers about the status of their request. It can be

utilised in both Public Transport (PT) and Demand Responsive Transport (DRT) for both passenger and cargo transport. This service provides information that can be proved very useful to every kind of urban travel planning. Therefore, it is crucial for providers in MaaS with intermodal trips to have at their disposal the arrival time at the transfer points to better plan the route and recalculate in the case of delays.

This service may be personalised based on

- the type of outputs the operator wants to receive (e.g., travel time to the next stop, arrival time to the next stop, expected delay at the stop…)
- the frequency of their calculation (e.g., every few seconds, at every depart from the previous stop…).

Of course, in addition to all the above, user-interface issues on how operators will interact with the system will be personalised. All the above personalisation functionality is available to the operator via the graphical user interface of each service.

7.3 Product Items from Pilot sites

Many Pilot Site Operators have expressed their interest in sharing services to the marketplace. In the following months, extensive interviews will be held among Pilot Sites and the Marketplace technical team, to decide whether these services will be hosted at the SHOW Marketplace or it will be utilised as a platform for service promotion that will redirect the interested buyers to the respective URL/platform. Additionally, many pilot sites have expressed interest in applying services developed by other pilot sites and stakeholders within the SHOW ecosystem and are briefly presented below.

Linköping provides a service for PT, DRT and MaaS mode, where passengers will be able to view real-time information of the status of the trip as a multimodal trip planner, drivers are provided with a pick-up and drop-off map, while fleet managers are provided with vehicle and event tracking.

Brno provides to passengers real-time vehicle location and schedule information, to drivers the ability to check seat occupancy and capacity in addition to being able to view pickup and dropoff locations on the map, while to fleet managers the ability to ensure vehicle tracking and Passenger Security and Trust assurance.

Before closing this chapter, it is important to note that many pilot sites have expressed their interest in acquiring services that are not yet implemented and will be provided through this platform. Specifically **for passengers, real-time trip tracking** is requested by *Tampere*, *Graz* and *Trikala* pilot sites, **real-time vehicle location** and **notification and alerts** are requested by *Graz* and *Trikala* sites, **seat booking** is requested by *Gothenburg, Brno, Graz, Trikala* and *Saltzburg* pilot sites, **scheduled booking** is requested by *Gothenburg* and *Graz*, **e-ticket service** is requested by *Gothenburg, Brno* and *Graz*, **multimodal trip planner** is requested by *Brno, Graz* and *Trikala* sites, **on-demand DRT** and **schedule information** services are requested by *Graz* pilot site. **For drivers**, *Saltzburg, Graz* and *Gothenburg* have expressed interest in the **seat occupancy** service. *Gothenburg, Brno* and *Graz* have expressed their interest in the **QR ticket scanner** service and *Graz* has shown interest in the **pickup and dropoff map** service. **For fleet managers**, **vehicle tracking** is requested by *Tampere* and *Graz* pilot sites, **event tracking** is requested by *Brno* and *Graz* pilot sites, while *Gothenburg* and *Brno* have expressed interest in **cybersecurity assurance**. **Passenger details** service is requested by *Gothenburg* and *Graz* sites. *Graz* pilot site has finally expressed interest considering the **alerts and notification** and **report extraction** services, while Gothenburg has also shown interest in the **route optimisation** service.

7.4 Additional Product Items of SHOW Marketplace

Subchapter 4.2 describes the product items that the SHOW Marketplace aims to host, as depicted in Figure 7. Through the SHOW Marketplace platform, the SHOW consortium obtains the ability to host and promote different and significant parts of the project in the platform. WP5 has a major contribution to the new ***algorithms'*** deployment that are valuable for both operators and passengers. The SHOW marketplace can be the place to publish related material for public use. This action would foster cooperation with teams involved in relevant projects.

In contrast with the services that are considered to be oriented for the operators' needs, the ***applications*** that will be hosted in SHOW Marketplace target to the end-users of the automated vehicles providing useful components for the utilisation of them. The passengers would be able to download the corresponding applications in order for the optimum exploitation of automated transportation's benefits to be achieved. The applications would improve the quality of CCAV services and increase the passengers' trustworthiness in automated transportation.

Moreover, indicative objects composing SHOW Marketplace's ***Data Model*** are the SHOW Data Registry, the defined JSON format for the data provision, and the raw data processing in the SHOW Data Management Platform. In this procedure of creating SHOW Data Registry, CERTH/ITI's team has drawn inspiration from existing Public Transport models such as Transmodel and NeTEx. The corresponding material would also be uploaded for a clear review to be feasible.

A significant achievement of the SHOW project is the data concentration from countries and cities across Europe. The produced ***datasets*** could be utilised in the development of new services and applications. Therefore, there is a need to be circulated among a community that would deploy the datasets. The uploading of the datasets in the SHOW Marketplace demands the permission of the data providers. Furthermore, public data or data from other European projects would be available through the SHOW Marketplace to create a valuable data repository for future usage.

The SHOW ***Dashboard*** could also be hosted in the marketplace. Since the autonomous fleet will be utilised more and more in the next years, SHOW Dashboard and similar technologies would be an irreplaceable part of the operators' system. The architecture, the source code and the description of its functionalities would assist in this direction.

The contribution of the deliverable D4.1 of SHOW [79] would be located in the architecture module. The analysis of the SHOW ***architecture*** combined with other relevant ones could be utilised to realise better autonomous public or private ecosystems that would be dominant in the next years. The demanded interoperability among the different systems, the adaption of a common approach and the overcome of the existing the respective documentation available through the SHOW Marketplace.

Furthermore, companies will be able to promote their hardware equipment via the ***tool*** cluster. Sensors, radars, antennas for 5G transmissions, cameras, Lidars, etc. will be gathered for the appropriate materials to be selected by the contributors of future projects. The SHOW Marketplace will combine each component with an already existing system giving major feedback about its properties and integration with the rest parts. The co-existence of a tool with the implementation of them in real analysis would assist the trust on the efficiency of it.

The contributions of the European and other projects are, by all means, more than useful in the upcoming domination of autonomous fleets. Their ***use cases***, the

exported technologies and main conclusions could be the ground on which the future projects should be supported. Some of the materials are publicly available. However, the concentration in one dedicated to autonomous vehicles domain, such expected to be the SHOW Marketplace, would make it easier for the public concerned. Moreover, relevant papers could be accessible and concentrated also. Finally, ***UI widgets*** and single components that could be utilised in different parts of designing an autonomous vehicle system could be uploaded, rendering the SHOW Marketplace a unique platform in which each interested site would have the opportunity to find the ideal solution.

8 Conclusions and outlook

The deliverable objective is to present the logic behind the marketplace and describe the underlying infrastructure and the products that will host. This deliverable introduces the marketplace's initial version, while two updates are expected in months 30 and 40 respectively.

A state-of-the-art analysis of the marketplaces and the architectural approaches set the ground for understanding the available solutions. The state-of-the-art analysis generally defined the marketplace and presented marketplaces from other European projects and business applications. The business logic was presented to clarify the different components. Specifically, the stakeholders and product clusters are the core components of a marketplace that dictate the marketplace's technical side with requirements and use case scenarios. A detailed review of open-source platforms presented the range of choices used to design and develop the marketplace. The selected tools along with the first version of the marketplace were presented. The deliverable concludes with the product items that users can upload to the marketplace emphasizing in the services.

The project sets ambitious goals for the marketplace, as there is the opportunity for a lasting impact on the automotive sector. Traditionally, marketplaces are convenient one-stop places for users to discover applications for their needs. The SHOW Marketplace can impact the CCAM sector with the inclusion of numerous stakeholders and multi-layered products in the sector. The requirements and objectives for each stakeholder differ but SHOW accounts for all of them in the defined identities, product items, requirements, and use case scenarios. The result of such actions is to encourage the active participation of the stakeholders and influence the community.

Regarding the next steps of the SHOW Marketplace development activities, various meetings with all the interested partners will be organised to define potential integration/hosting of the partners' offered services or applications to the marketplace depending on the maturity of implementation. Moreover, a prominent action that will be considered in the future is the technical review of the marketplace by a committee of SHOW partners specialising in software engineering. Similarly, the SHOW Marketplace's quality assurance will be realised in a mature phase. Essentially, user acceptance testing will follow to receive constructive comments from the end-users concerning user experience and satisfaction. Finally, according to the results of the aforementioned meetings with Marketplace stakeholders, it is possible to modify, enhance and enrich the requirements (for instance, set different visibility constraints for some product items) towards the next version of the Marketplace, since this is also the major advantage of adopting an agile software development approach.

References

[1] Sommerville, I. (2011). Software engineering 9th Edition. *ISBN-10*, *137035152*, 18.

[2] Agile Manifesto. Retrieved 19 October , 2021, from: https://agilemanifesto.org

[3] Hazzan, O., & Dubinsky, Y. (2009). Agile software engineering. Springer Science & Business Media.

[4] Bradner, S. (1997). RFC2119: Key words for use in RFCs to Indicate Requirement Levels.

[5] Cohn, M. (2004). User stories applied: For agile software development. Addison-Wesley Professional.

[6] Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017, May). SCRUM model for agile methodology. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 864-869). IEEE.

[7] Robertson, S., & Robertson, J. (2006). Mastering the Requirements Process Second Edition. Addison Wesley Professional.

[8] Arachchi, S. A. I. B. S., & Perera, I. (2018, May). Continuous integration and continuous delivery pipeline automation for agile software project management. In 2018 Moratuwa Engineering Research Conference (MERCon) (pp. 156-161). IEEE.

[9] Perrey, R., & Lycett, M. (2003, January). Service-oriented architecture. In 2003 Symposium on Applications and the Internet Workshops, 2003. Proceedings. (pp. 116-119). IEEE.

[10] SOA Manifesto. Retrieved October 1, 2021, from: http://www.soa-manifesto.org/

[11] Durante, A., Bell, D., Goldstein, L., Gustafson, J., & Kuno, H. (2000). A model for the E-service marketplace. HP LABORATORIES TECHNICAL REPORT HPL, 17.

[12] Di Cerbo, F., Bezzi, M., Kaluvuri, S. P., Sabetta, A., Trabelsi, S., & Lotz, V. (2012, May). Towards a trustworthy service marketplace for the future internet. In The Future Internet Assembly (pp. 105-116). Springer, Berlin, Heidelberg.

[13] Eltoweissy, M., Azab, M., Olariu, S., & Gračanin, D. (2019, September). A new paradigm for a marketplace of services: Smart communities in the IoT era. In 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) (pp. 1-6). IEEE.

[14] Sherman, A. T., Javani, F., Zhang, H., & Golaszewski, E. (2019). On the origins and variations of blockchain technologies. IEEE Security & Privacy, 17(1), 72-77.

[15] Chaum, D. L. (1979). Computer Systems established, maintained and trusted by mutually suspicious groups. Electronics Research Laboratory, University of California.

[16] Haber, S., & Stornetta, W. S. (1990, August). How to time-stamp a digital document. In Conference on the Theory and Application of Cryptography (pp. 437-455). Springer, Berlin, Heidelberg.

[17] Nakamoto, S. (2008). Re: Bitcoin P2P e-cash paper. The Cryptography Mailing List.

[18] Ocean Protocol Foundation. (December 2020). Ocean Protocol: Tools for the Web3 Data Economy. https://oceanprotocol.com/tech-whitepaper.pdf

[19] Musso, Stefano, et al. "A Decentralized Marketplace for M2M Economy for Smart Cities." 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE). IEEE, 2019.

[20] Lamtzidis, Odysseas, and John Gialelis. "An IOTA based distributed sensor node system." 2018 IEEE Globecom Workshops (GC Wkshps). IEEE, 2018.

[21] Data Europa EU. (10 December 2018). CORDIS - EU research projects under Horizon 2020 (2014-2020). https://cordis.europa.eu/data/cordis-h2020projectDeliverables.xlsx

[22] EC Cordis. (27 September 2021). AUTOMAT project. https://cordis.europa.eu/project/id/644657
[23] Pillmann, J., Sliwa, B., & Wietfeld, C. (2018, June). The automat CVIM-a scalable data model for automotive big data marketplaces. In 2018 19th IEEE International Conference on Mobile Data Management (MDM) (pp. 284-285). IEEE.
[24] EC Cordis. (27 September 2021). Cross-CPP project. https://cordis.europa.eu/project/id/780167
[25] Cross-CPP Library. (May 2021). Cross-CPP Methodology. https://14012e06-adc0-4f51-8df3-b845f2929cf7.filesusr.com/ugd/6c1158_d47c0f8578bc41aa973cead2cb5cea55.pdf
[26] EC Cordis. (27 September 2021). CLARITY project. https://cordis.europa.eu/project/id/693881
[27] Drupal Software. (27 September 2021). https://www.drupal.org/
[28] EC Cordis. (27 September 2021). ACTIVAGE project. https://cordis.europa.eu/project/id/732679
[29] EC Cordis. (27 September 2021). ACTIVAGE: Marketplace components. https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5bd46ca49&appId=PPGMS
[30] EC Cordis. (27 September 2021). The Open Government Marketplace, electronic version of the CLARITY blueprint and an evaluation of take-up, case studies and testimonials. https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5b8ee2265&appId=PPGMS
[31] EC Cordis. (27 September 2021). GOAL project. https://cordis.europa.eu/project/id/731656
[32] EC Cordis. (27 September 2021). GOAL: website. https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5bc4c8e54&appId=PPGMS
[33] EC Cordis. (27 September 2021). GOAL: social marketplace application. https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5c4786a9f&appId=PPGMS
[34] EC Cordis. (27 September 2021). BONSEYES project. https://cordis.europa.eu/project/id/732204
[35] EC Cordis. (27 September 2021). BONSEYES: Data Marketplace Report. https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5d212b1b6&appId=PPGMS
[36] Otonomo. (27 September 2021). One-stop Shop for Vehicle Data. Retrieved October 1, 2021, from: https://otonomo.io/
[37] Caruso Dataplace. (27 September 2021). From Connected Cars to Connected Business. https://www.caruso-dataplace.com/
[38] CityMapper. (27 September 2021). Making Cities Usable. Retrieved October 1, 2021, from: https://citymapper.com/
[39] Beeline Singapore. (27 September 2021). A smart mobility platform that enables demand-driven shared transport. Retrieved October 1, 2021, from: https://www.beeline.sg/
[40] Optimile. (27 September2021). Offer the ultimate mobility experience. Retrieved October 1, 2021, from: https://www.optimile.eu/
[41] Medium. (10 October 2017). Open sourcing Beeline. Retrieved October 1, 2021, from: https://medium.com/open-government-products/open-sourcing-beeline-8da9cfde9936
[42] FIWARE Marketplace. (27 September 2021). FIWARE Marketplace. Retrieved October 1, 2021, from: https://marketplace.fiware.org/
[43] UniversAAL IoT. (27 September 2021). Retrieved October 1, 2021, from: https://www.universaal.info/

[44] OpenIoT. (27 September 2021). Open Source cloud solution for the Internet of Things. Retrieved October 1, 2021, from: http://www.openiot.eu/
[45] Krishnamachari, B., Power, J., Kim, S. H., & Shahabi, C. (2018, June). I3: An iot marketplace for smart communities. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services (pp. 498-499).
[46] Ramachandran, G. S., Radhakrishnan, R., & Krishnamachari, B. (2018, September). Towards a decentralized data marketplace for smart cities. In 2018 IEEE International Smart Cities Conference (ISC2) (pp. 1-8). IEEE.
[47] Xu, Ronghua, et al. "Blendsm-ddm: Blockchain-enabled secure microservices for decentralized data marketplaces." 2019 IEEE International Smart Cities Conference (ISC2). IEEE, 2019.
[48] Pillmann, Johannes, et al. "Novel common vehicle information model (cvim) for future automotive vehicle big data marketplaces." 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2017.
[49] Butzin, Björn, Frank Golatowski, and Dirk Timmermann. "Microservices approach for the internet of things." 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2016.
[50] S. (2021, July 22). Multivendor Marketplace Platform - Enterprise Commerce Software. Shuup - Multivendor Marketplace Platform. Retrieved October 1, 2021, from: https://www.shuup.com/
[51] Magento Extensions & Themes for Your Online Store | Marketplace. (n.d.). Magento. Retrieved October 1, 2021, from https://marketplace.magento.com/
[52] Open Source eCommerce Marketplace Software - free Joomla Marketplace Platform. Sellacious. (n.d.). Retrieved October 1, 2021, from https://www.sellacious.com/.
[53] All-In-One Marketing Platform for Small Business. (n.d.). Mailchimp. Retrieved October 1, 2021, from https://mailchimp.com/
[54] W. (n.d.). Best Multi Vendor Marketplace Plugin & eCommerce Solution. WeDevs. Retrieved October 1, 2021, from https://wedevs.com/dokan/
[55] Best Laravel E-Commerce package. (2020, February 10). AIMEOS. Retrieved October 1, 2021, from: https://aimeos.org/laravel-ecommerce-package
[56] Ever Co. LTD. (2021, September 21). *Home - Ever*. Retrieved October 1, 2021, from: https://ever.co/
[57] Forcier, J., Bissex, P., & Chun, W. J. (2008). Python web development with Django. Addison-Wesley Professional.
[58] Dauzon, S., Bendoraitis, A., & Ravindran, A. (2016). Django: Web Development with Python. Packt Publishing Ltd.
[59] Django version 3.2.7. Retrieved October 1, 2021, from: https://docs.djangoproject.com/en/3.2/releases/3.2.7/
[60] Python, S. F. (1991). Python programming language–official website. Python is a.
[61] Yudin, A. (2020) Building Versatile Mobile Apps with Python and REST.
[62] Gore, H., Singh, R. K., Singh, A., Singh, A. P., Shabaz, M., Singh, B. K., & Jagota, V. (2021). Django: Web Development Simple & Fast. Annals of the Romanian Society for Cell Biology, 25(6), 4576-4585.
[63] MongoDB. https://www.mongodb.com/
[64] Hao, L., Zhang, J., & Ma, X. (2019, July). Design and Implementation of Simulation Training System Based on MVC Architecture. In IOP Conference Series: Materials Science and Engineering (Vol. 563, No. 5, p. 052043). IOP Publishing.
[65] Rubio, D. (2017). Beginning Django. Apress.
[66] Yu, Y., Leite, J. C., & Mylopoulos, J. (2004, September). From goals to aspects: discovering aspects from requirements goal models. In Proceedings. 12th IEEE International Requirements Engineering Conference, 2004. (pp. 38-47). IEEE.

[67] Pitt, C. (2021). Publishing Your Code. In Pro PHP 8 MVC (pp. 351-357). Apress, Berkeley, CA.
[68] Paolone, G., Marinelli, M., Paesani, R., & Di Felice, P. (2020). Automatic Code Generation of MVC Web Applications. Computers, 9(3), 56.
[69] Eyada, M. M., Saber, W., El Genidy, M. M., & Amer, F. (2020). Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments. IEEE Access, 8, 110656-110668.
[70] Cottrell, N. (2020). Fault-Tolerant Design. In MongoDB Topology Design (pp. 31-53). Apress, Berkeley, CA.
[71] MySQL, A. B. (2001). MySQL.
[72] Harrison, G., & Harrison, M. (2021). MongoDB Architecture and Concepts. In MongoDB Performance Tuning (pp. 13-32). Apress, Berkeley, CA.
[73] Keyckoak. https://www.keycloak.org/
[74] Divyabharathi, D. N., & Cholli, N. G. (2020). A review on identity and access management server (KeyCloak). International Journal of Security and Privacy in Pervasive Computing (IJSPPC), 12(3), 46-53.
[75] Keycloak, O. S. I. (2021). Access Management For Modern Applications and Services.
[76] ISO/IEC 27001. Retrieved October 1, 2021, from: https://www.iso.org/isoiec-27001-information-security.html
[77] SHOW (2021). D5.2: Big Data and Artificial Intelligence Toolboxes. Deliverable of the Horizon-2020 SHOW project, Grant Agreement No. 875530.
[78] AVENUE (2020). D4.5: Second iteration In-vehicle services. Deliverable of the Horizon-2020 AVENUE project, Grant agreement ID: 769033
[79] SHOW (2021). D4.1: Open modular system architecture and tools. Deliverable of the Horizon-2020 SHOW project, Grant Agreement No. 875530.
[80] Ng, V., & Kim, H. M. (2021). Autonomous vehicles and smart cities: A case study of Singapore. In *Smart Cities for Technological and Social Innovation* (pp. 265-287). Academic Press.
[81] Wu, H. H., Gilchrist, A., Sealy, K., Israelsen, P., & Muhs, J. (2011, May). A review on inductive charging for electric vehicles. In *2011 IEEE international electric machines & drives conference (IEMDC)* (pp. 143-147). IEEE.

Appendix I

This appendix presents the excepted characteristics for the products that will be accommodated in the marketplace. Each table presents the characteristics for each different product type. Apart from these characteristics, a short description and the level of necessity (mandatory/optional) for this characteristic accompany the characteristics name. Finally, for each different product type there is also a respective example (in a table format) right after the corresponding table of the product item.

Table 26: Data scheme for services

Service Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the service	mandatory
Description	A description that explains the functionality of the service	mandatory
Tags	Provide the specific tags with which the service will be reachable (Operational, Energy Management etc.)	mandatory
Technical requirements	A list with all the technical requirements (with respect to the software and hardware) in order for the service to run flawlessly	mandatory
Version	The current service version	optional
Price	The price of the service (can be free)	mandatory
Service/URL	The software needed for the service or the corresponding URL	mandatory
Image	One representative image of the service	optional
Size	The size of the service	mandatory
Publisher	The service provider	mandatory

Table 27: Mockup for services upload

Characteristic	Description
Name	Follow my kid
Description	The service is dedicated for automated public transport. It provides the operator with the ability to detect if a kid remains standalone in the vehicle.
Tags	Operational, in-cabin monitoring, camera based, CERTH/ITI
Technical requirements	A camera sensor (RGB) and an IoT Gateway are mandatory.
Version	1.0
Price	25.99€
Service/URL	zip/www.folowmykid.com
Image	
Size	55.5 MB
Publisher	CERTH/ITI

Table 28: Data scheme for applications

Application Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the application	mandatory
Description	A description that summarizes the application	mandatory
Tags	Provide the specific tags with which the application will be reachable (optimal routing, estimated time of arrival etc.)	optional
Technical Requirements	A list with all the technical requirements in order for the application to run flawlessly	mandatory
Version	Current application version	optional
Price	The price of the application (can be free)	mandatory
Application/URL	The application as executable or the corresponding URL	mandatory
Image	One representative image of the application	optional
Size	The size of the application	mandatory
Development Team	The technical team providing the application	mandatory

Table 29: Mockup for applications upload

Characteristic	Description
Name	Just in time
Description	The application suggests to the passenger the optimal route for their destination.
Tags	Optimal routing
Technical Requirements	Android 6 or newer
Version	2.1
Price	Free
Application/URL	exe/www.justintime.com
Image	
Size	10.2MB
Development Team	Just in time team

Table 30: Data scheme for single components

Single Component Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the component	mandatory
Description	A description that summarises the component	mandatory
Tags	Provide the specific tags with which the component will be reachable	optional
Technical Requirements	Provide the technical prerequisites for the component to be integrated	mandatory
Version	Current component version	optional
Price	The price of the component (can be free)	mandatory
Component/URL	The component as source code or the corresponding URL	mandatory
Size	The size of the component	mandatory
Development Team	The technical team providing the component	mandatory

Table 31: Mockup for single components upload

Characteristic	Description
Name	SHOW Marketplace
Description	The SHOW Marketplace is a dedicated to the automated transportation platform.
Tags	Marketplace, automated transportation, CERTH/ITI
Technical Requirements	Windows 7/10, Ubuntu 18.04/20.04, macOS
Version	1.0
Price	Free
Component/URL	Zip/www.showmarketplace.eu
Size	87.8MB
Development Team	CERTH/ITI

Table 32: Data scheme for UI widgets

UI Widget Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the UI Widget	mandatory
Description	A description that should present the features of the Widget.	mandatory
Tags	Provide the specific tags with which the UI Widget will be reachable	optional
Technical Requirements	Provide the technical prerequisites that a system and underlying infrastructure should follow for the optimal use of the UI Widget.	mandatory
Version	Current UI widget version	optional
Price	The price of the UI widget (can be free)	mandatory
UI Widget/URL	The UI Widget as source code or the corresponding URL	mandatory
Mockups	Present with pictures the design of the Widget.	mandatory
Size	Provide the component size	mandatory
Development Team	The technical team providing the UI widget	mandatory

Table 33: Mockup for UI widgets upload

Characteristic	Description
Name	Coordinates Finder.
Description	A simple tracker extension for AVs.
Tags	Software components/UI templates, coordinates, location, localisation.
Technical Require-ments	1) npm version 3.5.2 and higher 2) npm install --save coordinates.
Version	0.5.7.
Price	5€ per licence.
UI Widget/URL	coordinates_package.js coordinates_package.apk
Mockups	
Size	45 kB.
Development Team	MockupWidgetTeamINC.

Table 34: Data scheme for algorithms

Algorithm Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the algorithm	mandatory
Description	A description that explains the functionality of the algorithm. The description can include the problem statement, how the algorithm works, and other necessary components for users to understand the algorithm.	mandatory
Tags	Provide the specific tags with which the algo-rithm will be reachable	optional
Technical re-quirements	Set prerequisites for the algorithm. The pre-requisites can vary, and examples are the pro-gramming language, dependencies, and the required computational power.	mandatory
Version	The current algorithm version	optional
Price	The price of the algorithm (can be free)	mandatory
Algorithm/URL	The file(s) needed for the algorithm to run or the corresponding URL	mandatory
Developer(s)	The developer(s) of the algorithm	mandatory

Table 35: Mockup scheme for algorithms

Characteristic	Description
Name	Estimated Time of Arrival Calculation
Description	Calculation of estimated time of arrival using cutting edge gradient boosting techniques
Tags	Relevants/Algorithms, ETA, ETA calculation, time elapsed, timestamp, location, distance, time
Technical require-ments	1) Python3 3.6.9 or higher 2) pip3 21.3.1 or higher 3) python3 libraries: pandas, numpy, sklearn, xgboost, haversine
Version	0.6.2
Price	30 € per licence
Algorithm/URL	git clone github.com/mockup/repo/eta_SHOW.git
Developer(s)	CERTH/ITI SHOW technical team

Table 36: Data scheme for data models

Data Model Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the data model	mandatory
Description	The description is a field of text that intends to allow owners to inform the public on their data model. Subjects to the description can be the problem definition, and an overview of use cases.	mandatory
Tags	Provide the specific tags with which the data model will be reachable	optional
Version	The current version of the data model	optional
Data Model/URL	The file(s) needed for the specification of the data model or the corresponding URL	Mandatory
Publisher(s)	The people and team that put the effort in publishing the data model.	mandatory

Table 37: Mockup scheme for data models

Characteristic	Description
Name	Data Relation standard
Description	The data structure for standardisation of AV componential rela-tionship.
Tags	Software Components/Data Analytics, public automated transport
Version	1.0.15.
Data Model/URL	mockupModel.erwin
Publisher(s)	DataModelCo ltd.

Table 38: Data scheme for datasets

Dataset Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the dataset	mandatory
Description	A description of the purpose that the dataset serves	mandatory
Tags	Provide the specific tags with which the dataset will be reachable	optional
Technical requirements	Technical requirements of the dataset such as for example the training-test ratio or the length of the dataset.	mandatory
Dataset/URL	The file(s) of the dataset or the corresponding URL.	mandatory
Size	The size of the dataset	mandatory
Publisher	The dataset provider	mandatory

Table 39: Mockup scheme for datasets

Characteristic	**Description**
Name	CCAM dataset
Description	This dataset is the result of the coordinated efforts of Research Centre to produce a dataset for the CCAM for Thessaloniki. 100 devices used for gathering the data for a duration of 01/01/2021 to 01/07/2021.
Tags	dataset; CCAM; [Research Centre on datasets]
Technical requirements	The dataset size is 1GB and is split into 5 files for convenience. 1,000,000 samples are present in the files.
Data Model/URL	File.zip, description.txt, data.csv
Size	1GB
Publisher	Research Centre on datasets

Table 40: Data scheme for tools

Tool Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the tool	mandatory
Description	A description for the purpose that the tool serves.	mandatory
Tags	Provide the specific tags with which the tool will be reachable	optional
Price	The price of the tool	mandatory
URL	A URL to lead users to a source for purchasing the tool.	optional
Image(s)	One or more representative images of the tool	mandatory
Provider	Company/organisation that provides the specific tool	mandatory

Table 41: Mockup scheme for tools

Characteristic	Description
Name	CCAM tool for measuring pollution
Description	This is a device that can help anyone who wants to measure the pollution that is produced by their vehicle. The pollution monitoring can help mitigate the environmental concerns of the public. Bluetooth can be used to send the data. The device's manufacturer is Device Manufacturer.
Tags	device; pollution; Bluetooth
Price	50,00€
URL	www.pollutiondevice.com/CCAM_tool
Image(s)	
Provider	Device Manufacturer

Table 42: Data scheme for architectures

Architecture Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the Architecture	mandatory
Description	A description for the purpose that the architecture serves.	mandatory
Architecture/URL	The file(s) representing the architecture or the corresponding URL.	mandatory
Image(s)	One or more representative images of the architecture	mandatory
Author(s)	The people and team that put the effort in formulating and publishing the architecture.	optional

Table 43: Mockup Data scheme for architectures

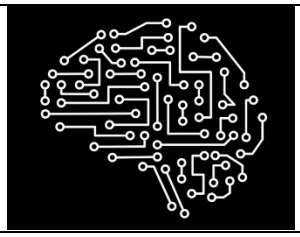
Characteristic	Description
Name	Pretrained Transformer Neural Network architecture
Description	A new simple network architecture is proposed, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely.
Architecture/URL	https://www.Trasnformer-Architecture.eu
Image(s)	
Author(s)	Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit

Table 44: Data scheme for dashboards

Dashboard Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the dashboard.	mandatory
Description	The description should present the features that the Dashboard possess.	Mandatory
Tags	Provide the specific tags with which the dashboard will be reachable	Optional
Technical requirements	The prerequisites are the characteristics that a system and underlying infrastructure should follow for the optimal use of the Dashboard. The operating system and RAM requirements can be examples.	Mandatory
Price	The owner of the Dashboard could set a price for users to acquire.	Mandatory
Dashboard/URL	The source files of the dashboard or the corresponding URL	Mandatory
Image(s)	One or more representative images depicting the design of the dashboard.	Mandatory
Size	The size of the dashboard	Mandatory
Development Team	The technical team providing the dashboard	mandatory

Table 45: Mockup Data scheme for dashboard

Characteristic	**Description**
Name	Autonomous Vehicle Dashboard
Description	Presents statistics on the performance of autonomous vehicles, as well as graphs for graphical representation
Tags	Autonomous vehicles performance
Technical requirements	Debian based Linux server Nginx version 1.20.2 and later Python version 3.8 and later Django 4.0 Vue.js version 3.2.20 and later MySQL version 8.0 and later
Price	100 €
Dashboard/URL	AV-dashboard.zip
Image(s)	
Size	20 MB
Development Team	CERTH/ITI

Table 46: Data scheme for case studies

Case Study Characteristics		
Characteristic	**Description**	**Necessity**
Name	The name of the case study report.	mandatory
Abstract	The abstract includes the main components of the report, its structure and conclusions of the analysis.	mandatory
Tags	Provide the specific tags with which the case study will be reachable	optional
Project	It provides details for the project which the report is a part of.	optional
Report/URL	The file with the case study report or the corresponding URL	mandatory
Author(s)	The name of the author(s) of the case study	optional

Table 47: Mockup Data scheme for case studies

Characteristic	**Description**
Name	Autonomous Vehicles and Smart Cities: a case study of Singapore [80]
Abstract	The development of Information Communication Technology (ICT) is reflected in a transport revolution that presents one of the greatest changes since the adoption of automobiles. Investors, planners, ICT experts, and policymakers are trying to make sense of the implications of disruptive technology on urban transport; autonomous vehicles (AVs) demand attention. There are both potential benefits and complicated risks in safety, livability, productivity, sustainability, and governance.
Tags	Smart City
Project	Describes information on the contribution of autonomous vehicles to smart cities
Report/URL	AutonomousVehicles_SmartCities.pdf
Author(s)	Vincent Ng, Hyung MinKim