SHared automation Operating models for Worldwide adoption

SHOW

Grant Agreement Number: 875530

D6.2: SHOW Marketplace and services – second 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

Undoubtedly, marketplaces in the contemporary society constitute a useful tool for the corresponding stakeholders since they are a common place for providers and consumers in order to perform selling and buying of products and services respectively. Moreover, in some circumstances, a marketplace could stimulate a whole community, and this is the vision of the proposed marketplace with respect to the Connected, Cooperative & Automated Mobility (CCAM) community. Therefore, the marketplace objective is to act as a one-stop place for the community that could also accelerate CCAM adoption and awareness by offering innovative CCAM technologies and services.

Considering the importance of such a vision, SHOW project includes a whole work package dedicated to SHOW Marketplace and Services. Hence, the present deliverable is the second subsequent version of the previously released D6.1, "SHOW Marketplace and services – first version", and is the outcome of work conducted in ***Work Package 6: Services Marketplace***, which encompasses five related activities: *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 Personalized Services*. More specifically, this deliverable summarizes the work performed, since D6.1 release and until M30 of the project.

The main outcomes of the work described in this deliverable are related to the business strategy definition of the SHOW Marketplace, to the new SHOW Marketplace features, to the development operations and infrastructure updates and to the integration of product items into the first public version of the SHOW marketplace, which was launched in June 2022.

The SHOW marketplace is continuously upgraded, enriched in terms of content and living at: https://show-project.eu/ccam-marketplace/.

Document Control Sheet

Start date of project:	01 January 2020
Duration:	48 months
SHOW Del. ID & Title:	D6.2: SHOW Marketplace and services – second 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:	Andreas Laskaris (CERTH-ITI), Georgios Nikoletos (CERTH-ITI), Evangelos Antypas (CERTH-ITI), Konstantinos Giapantzis (CERTH-ITI), Dimitrios Tsiktsiris (CERTH-ITI), Eleni Diamantidou (CERTH-ITI), Theoktisti Marinopoulou (CERTH-ITI), Athanasios Sersemis, Alexandros Papadopoulos (CERTH-ITI), Iordanis Papoutsoglou (CERTH-ITI), Georgios Spanos (CERTH-ITI), Antonios Lalas (CERTH-ITI), Konstantinos Votis (CERTH-ITI)
Other authors involved:	Roberto Carroccia (CTLup), Davide Shingo Usami (CTLup), Sevket Oguz Kagan Capkin (CTLup), Tristan Mueller (AVL), Mattia Venditti (AVL), Wolfram Ratzke (AVL)
Internal Reviewers:	Nikolaos Tsampieris (ERTICO), Jaâfar Berrada (VEDECOM), Alberto Bellini (SWARCO), Maria Gkemou (CERTH/HIT)
External Reviewers:	–
Actual submission date:	07/11/2022
Status:	Final
File Name:	SHOW_D6.2_Marketplace_2nd version_Final

Document Revision History

Version	Date	Reason	Editor
0.1	19/05/2022	ToC	CERTH-ITI
0.5	03/06/2022	Added Sections 1-4	CERTH-ITI
0.8	24/06/2022	Added Section 5	CERTH-ITI
0.9	05/07/2022	Draft completed	CERTH-ITI
1.0	08/07/2022	Full version provided for peer review	CERTH-ITI
1.9	07/10/2022	1st peer reviewed version shared	CERTH-ITI
2.0	07/11/2022	2nd peer reviewed version shared (upon CERTH/HIT comments) for submission	CERTH-ITI

Table of Contents

List of Tables

List of Figures

Abbreviation List

Abbreviation	Definition
3PL	Third Party Logistics
API	Application Programming Interface
AV	Autonomous Vehicle
BEV	Battery Electrical Vehicle
BMC	Business Model Canvas
CBA	Cost-Benefit Analysis
CCAM	Connected, Cooperative & Automated Mobility
CCAV	Connected, Cooperative & Automated Vehicles
CD	Continuous Delivery
CDe	Continuous Deployment
CF	Continuous Feedback
CI	Continuous Integration
C-ITS	Cooperative Intelligent Transport System
CPS	Cost Per Sale
CRUD	Create Read Update Delete
CSRF	Cross Site Request Forgery
DevOps	Development Operations
ETA	Estimated Time of Arrival
IaC	Infrastructure as Code
ICEV	Internal Combustion Engine Vehicle
IT	Information Technology
IoV	Internet of Vehicles
ML	Machine Learning
OEM	Original Equipment Manufacturer
PCI	Payment Card Industry
PHEV	plugin hybrid vehicle
PT	Public Transport
R&D	Research and Development
R&I	Research and Innovation
ROI	Return On Investment
SaaS	Software as a Service
SDK	Software Development Kit
SEO	Search Engine Optimisation
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UI	User Interface
UX	User experience

1 Introduction

1.1 Purpose and structure of the document

This deliverable is the intermediary version for the SHOW Marketplace and services and constitutes the follow-up of D6.1 [1]. The deliverable is part of WP6 titled *"Services Marketplace"*, and all the underlying activities of the WP (A6.1, A6.2, A6.3, A6.4, A6.5) have contributed to its effort in the activities covered by the deliverable. The WP's objective is the development of a CCAM services marketplace for promoting various types of products and services related to CCAM, for equally internal and external (to the project) promotion and deployment. The third and final version of the deliverable is due M40 and will wrap up the activities of the whole WP.

The second chapter of the deliverable presents business models for the online marketplaces and provides suggestions on the model to follow. The chapter is a commitment as of the previous version of Marketplace, where it was stressed that an elaboration on pricing policies is considered to be necessary for the Marketplace. The third chapter updates the development status of the previous version, considering the requirements set in D6.1 [1]. In detail, the Marketplace has been updated with views and functions for procedures like payment, rating, and favourites for improving the user experience. The fourth chapter elaborates on the development and infrastructure philosophy that the developing team follows for developing the Marketplace platform, by adopting DevOps tools, towards cultural shift for the SHOW Marketplace implementation. The fifth chapter details the services and products hosted on the Marketplace as part of WP6 efforts that are allocated to enriching the platform. The previous (first) version of the SHOW Marketplace Deliverable (D6.1) included the description of 24 draft product items (that were also uploaded provisionally in the actual marketplace infrastructure). In the current public version of the SHOW Marketplace there are only 13 product items out of them described/uploaded that are in a mature development state, thus this deliverable displays only the uploaded product items (either services or architectures). The SHOW Hackathon produced solutions are also included in the 13 aforementioned products of the Marketplace. Finally, the deliverable concludes with remarks and future steps on the actions for WP6.

1.2 Intended Audience

The intended audience of this Deliverable includes:

- Technical team of SHOW demo sites
- OEMs responsible for the CCAV deployment in demo sites
- Service designers
- Evaluation team of WP9
- Business stakeholders
- Academic and research community outside SHOW in the domain of CCAV

1.3 Interrelations

The interrelated activities and WPs remain as in D6.1 and are as follows (see more details in D6.1 [1]):

- A1.1 – System needs

- A1.3 – Use cases
- WP3 - Ethical and legal issues
- WP4 – System architecture & tools
- WP5 – Big Data collection, processing, & analytics

2 Business model

Business modeling is a concept that gives an overview of the technological, organizational, and strategic approaches that many businesses follow to discuss and assess innovation and create value for their organization and for their respective customers [2]. Accordingly, the economists Osterwalder and Pigneur define the term of business models as follows: "*A business model describes the rationale of how an organization creates, delivers, and captures value*."[3]. Thus, a business model is a simplified illustration of a business project.

Section 2 describes the business model for the SHOW Marketplace. More specifically, it explores the business goals and value propositions, the main challenges and proposed solutions, the customer segmentation, the main distribution channels and key activities, the business revenue model, the key partners and the cost structure. Special emphasis is given to the Business Revenue Model that is proposed for the SHOW Marketplace. The revenue model is one of the hardest choices that marketplace owners need to address, before launching it to the public, and should be reviewed (and revised if necessary) continuously. Concerning the SHOW Marketplace, the presentation of its business model will be demonstrated via the Business Model Canvas (BMC) [4]-with a few modifications compared to the traditional BMC-, preceded by a presentation of a thorough business analysis of each section of the BMC.

As all existing marketplaces, the SHOW Marketplace aims to become an online platform that starts, facilitates, organizes, and completes the exchange of services between service providers and service seekers. It intends to promote the implementation of shared, connected, and electrified automation in urban transport in order to advance sustainable urban mobility by serving as a place of promotion for innovative services and software products (both prototypes and commercial mature solutions)[1].

2.1 Business Goals/Value Propositions

In today's overcrowded and oversized cities, demands for efficient and organized public transport are constantly rising. Autonomous vehicles plan to revolutionize and drastically change the way people commute and make use of mass transit vehicles [5]. The special needs of the commuters and the time constraints are two important factors to make this future into a reality, as the existing solutions of prescheduled bus itineraries will become obsolete. Moreover, AVs stand as an alternative for commuters who do not own a vehicle, or they are not willing to commute by using their own personal car, something that can strengthen the necessity of the offered services.

The SHOW project works towards this effort by scanning and promoting mature solutions of its beneficiaries and beyond, but, also, by developing state-of-the-art technologies and services which will be eventually integrated into the SHOW Marketplace. As a Pan-European effort, the SHOW project aims to promote seamless and safe sustainable mobility [6]. Although the SHOW Marketplace can be considered to be a side product of the project (since its development does not affect the rest of the activities), it is however a vital part of the whole SHOW ecosystem in terms of its exploitation. Indeed, it will be at the forefront of the outcomes of each partner work and will be the main host of the developed product items and services. As a cloud-based software platform, it will assist distributors and operators to buy, sell, manage, and promote products and innovative services.

The vision of the SHOW Marketplace is to be a one stop-shop (the same is valid for the whole set of existing business marketplaces), dedicated in the CCAM domain. The key value propositions of the SHOW Marketplace were presented in the previous version of the deliverable [1]. However, one more is added in this version and is shown below:

- Easy-to-consume overview of available product and data items.

As public transport becomes increasingly a data-driven sector, and due to the deployment of on-demand services and the emergence of micro-mobility, the SHOW Marketplace will make evident that data sharing and data refinement can benefit the communities by assisting in the effort of safe and sustainable mobility. This can be achieved through the digitization of mobility, the optimization of existing services and the creation of new ones, while also encouraging the adoption of driverless automated vehicles [7].

Finally, through a plethora of advantages (as analyzed extensively in D6.1 [1]), the SHOW Marketplace will be unique in the research field of marketplaces for CCAM, by providing a variety of services and products in a more cohesive way.

2.2 Challenges and Solutions

Digital technologies have the potential to transform the way we commute, by making it smarter, more efficient, and more environmentally friendly. Especially for the transportation industry, such technologies can be both an opportunity and a challenge [8]. Since, the SHOW Marketplace will be based on all existing, innovative, and newly developed digital technologies, there are extra challenges and features that should be considered. In addition, data sharing and open data policies affect business models, while the challenges presented should be taken into consideration [9]. Some of the key challenges and the potential way to solve them are presented in Table 1 [1].

Table 1: Key challenges and solution

Challenge	Solution
Openness of the Marketplace	This can be achieved by relying on open standards.
Ensure user acceptance	Creating a customer-centric strategy (analyzed also in Chapter 4) will empower users and guarantee their satisfaction and acceptance.
Pricing, affordability, and methods for payment	The distinction between free and commercialized services will render them an affordable and low-cost solution.
Personalization of the SHOW Marketplace	By serving the user need according to specific attributes, thus enhancing the user engagement with the platform.
Lack of business and financing models for each product item (service) respectively	Specific business models, accompanied by detailed financial analysis approaches (e.g., CBA analysis), will give a detailed overview over the economic impact of the SHOW Marketplace and its products.
Providing maximum value to all stakeholders (WIN-WIN)	By creating a collaborative context for synergies among all stakeholders of the CCAM value chain, often allowing them to have a dual role as a client and/or a provider.

2.3 Customer Segmentation

As part of the development phase of the Marketplace, certain target groups have been identified that are potential business customers for the SHOW Marketplace and of the products hosted within, which have been already discussed in D6.1 [1]. Their key goals and interests are presented in Table 2 below. This information could also be proven useful for similar EU projects in the automotive sector as well.

Table 2: Customer segmentation

Role	Goals/Interests
Passenger	Transportation from A to B, comfort, awareness about new solutions coming.
Tier 1 Supplier	Provides solutions for OEMs; revenue and clientele expansion.
Original Equipment Manufacturers (OEMs)	Get benefited of new solutions and services that they can integrate in their new products.
Public or Private Transport Operator	Get aware of SoA solutions in the domain and create synergies with technology providers aiming at sustainable mobility promotion.
Public Transport Authority	Boost the appeal of public transportation through new CCAM solutions of varying readiness level.
Research/Academia	Reinforce R&I in the domain through new elements (get benefited mostly by open-source knowledge shared).
Technology/Service Provider	Offer their solutions to be exploited by different vendors in varying contexts; revenue increase; business synergies.
City/Authority	Increase traffic safety and efficiency, inclusion of all demographic groups, awareness about automation.
Traffic Management Centre Operator	Providers of new CCAM specific traffic management services, as implemented and deployed in SHOW. Get benefited from supporting services provided by technology providers.
Infrastructure Operator	Get benefited from services and solutions that can leverage their infrastructure operation (Digital Twins, Simulation suites, etc.).
Associations (e.g., Mobility Association)	Awareness about new research based and commercial CCAM services and solutions, that after they validate, they can promote to their members and, finally, Cities transport system passengers.

2.4 Distribution Channels/Key Activities

As the SHOW Marketplace is under a continuous implementation phase, not all of its full envisioned capabilities are available to potential customers (but also providers/contributors). Since the SHOW project is still ongoing, we only present a preliminary marketing plan that will be updated and specified in the final WP6 Deliverable (D6.3, M40).

The promotion of the SHOW Marketplace to the broad CCAM community will be enabled through a series of dissemination activities as follows, that, apart from contributing to the

general awareness about its existence, will reinforce its use and progressive enrichment beyond SHOW:

- Marketplace access through the SHOW project web page (https://show-marketplace.eu/).
- Conferences, events, workshops; organized by SHOW or those SHOW participates in.
- Social media campaigns.
- Paid advertising.
- Email marketing.
- Publications.
- Search engines: by optimizing the Marketplace for SEO, in order to reach people and businesses when they are actively looking for related (CCAM or SHOW specific) products or services.
- Offline channels: like word-of-mouth recommendations, radio, billboard, or TV campaigns.
- Public affairs activities by providing information to stakeholders directly.
- Printed materials (case studies, one-pagers, etc.).

2.5 Business Revenue Model

The revenue model is a plan that determines how businesses and organizations make money and is also an essential part of any business model. A well-designed model aids businesses in making decisions on the prioritization of different revenue streams, the pricing policies for products and services, and, in general, to better understand the target market. When it comes to online marketplaces, they act as a one-stop-shop that connects buyers and sellers and facilitates a transaction between them.

A number of revenue models used in software marketplaces exist. However, only few could be utilized for the SHOW Marketplace pricing policies, including:

- **Commission Model**: The most popular revenue model used for online marketplaces. It is a model where a certain fee is charges to the service/product provider (usually a percentage or a fixed price) for every sale that occurs through the marketplace.
- **Subscription Model**: In this revenue model, the user is charged a regular fee for gaining access to the platform.
- **Licensing Model**: A revenue model mostly used by SaaS platforms, where the client is charged a certain fee for acquiring a software license. Some examples of licensing models are the SaaS License model (also called subscription software model) and the Perpetual License model.
- **Freemium Model**: A revenue model where a marketplace has both free and premium features.
- **Listing Fee Model**: According to this revenue model, the marketplace charges its clients for posting products, services, or ads on the platform.
- **Featured Listing and Ads Model**: It is more commonly used as a supplementary revenue model for online marketplaces. It allows sellers to buy advertising privileges or even have their listing displayed before others, thus enhancing visibility on the platform.
- **Mixed Model**: Combining several models and having several revenue sources.

Some factors that need to be considered while selecting a suitable revenue model are the customers' needs and behavior, the market potential and the existing (if any) competition, the value propositions, and the revenue streams [10]. By addressing these factors, a marketplace can be unique and successful. Concerning the SHOW Marketplace, a Mixed Revenue Model will be used, as presented in the next section.

2.5.1 Proposed Revenue Model for Marketplace: Mixed Revenue Model

The proposed Revenue Model for the SHOW Marketplace is a Mixed Revenue Model. The Marketplace will be a free-to-enter and to register platform, but will have both free and premium features, since it is divided between free and commercial product items. Therefore, the first business revenue model it is based upon is the Freemium model, where the customer will have the option (concerning the free of charge services) to download and use the software for free with no loss of functionality. The Freemium model exhibits a variety of advantages, since it can make easier the engagement of the customers with the platform, and the required costs for customer acquisition are quite lower. The main challenge of the Freemium model is to find the right balance between the free and the premium features, so that potential customers do not undervalue what the platform is offering.

Additionally, since the SHOW Marketplace will host commercialized product items (services, etc.) that are developed as prototypes in the SHOW project and will be sold directly from the platform but will also host product items developed and offered from other service providers, different pricing policies must be combined. For the first case, a SaaS License model (also called Subscription pricing model -either Tiered or Flat-Rated- with payments processed monthly or annually over a multiple year contract) or a Perpetual License model (either Tiered or Flat-Rated pricing model) will be used. For the second case, the Commission Model will be used. At the initial business stages of the Marketplace, the SaaS License or the Perpetual License pricing model will serve as the sole revenue stream for the Marketplace.

The SaaS license model (also known as the subscription pricing model) is mostly used from SaaS platforms. The licenses (monthly or annual) are usually per user (or per vehicle that is an often case for the specific marketplace products) and include the license to use the software, technical support, maintenance, and the hosting environment. The main benefits of the SaaS License pricing model are as follows:

- Low upfront costs.
- Payments spread evenly over a long-term contract (e.g., 3-5 years).
- Non-performance risk usually weighs the vendor.
- The application of all software updates, upgrades, and maintenance of the hosting environment is the responsibility of the service providers.

On the other hand, the main disadvantages are as follows:

- Lack of ownership concerning the purchased products.
- The longer the commitment, the higher the total cost of ownership.
- Higher churn rate, thus higher revenue churn rate. SaaS churn rate measures the number of customers who cancel their subscriptions for a given period.

The perpetual license-pricing model is the opposite of the SaaS license model. In this model, the customers usually own the licenses forever (in perpetuity), by making an

upfront and one-time purchase. Perpetual licenses might also include an “Annual Support and Maintenance” fee, which is a percentage (10%-20%) of the perpetual license purchase and usually starts in the first year. The main benefits of the Perpetual License pricing model are:

- Better return on investment (ROI) over a multiple year horizon.
- Cost of ownership is lower after 3-4 years compared to the SaaS license model.
- If a large one-time investment occurs, the vendor might offer a significant discount.

On the other hand, its main disadvantages are:

- Higher upfront investment, thus higher risk from a business perspective.
- Unless an “Annual Support and Maintenance” fee has been set, the customer is responsible for implementing updates and maintenance.

The Commission Model is one of the most popular business models used for online marketplaces. In this model, the marketplace charges a percentage or a fixed fee for its services. This billing fee is called Cost per Sale (CPS) and is calculated as a percentage (discussed in the Appendix I) of the price of the product item. Finding the correct percentage is important, as it depicts the added cost for the provider. The main advantages of the Commission model are:

- A marketplace can capture value by taking a percentage of every sale made through the marketplace.
- It offers higher scalability, since the higher the value of the products sold, the more the income it generates.
- Less risks for new vendors to list their products because they only pay when a sale is made.

On the other hand, its disadvantages are:

- During the early stages of a marketplace, value is usually low, especially for marketplaces that only act as the intermediary.

At a later stage, the Listing Fee Model might also serve as a supplementary revenue stream for the SHOW Marketplace. In this model, sellers are charged a fee for listing a product on the marketplace, which can ensure a greater level of product quality on the marketplace. This might be risky for marketplaces and should not be used as the sole source of revenue, but in combination with other revenue models.

The main advantages of the Listing Fee model are:

- The marketplace can generate revenue, regardless of how well the products perform.

On the other hand, its disadvantages are:

- Dissuade new vendors from using the marketplace as their primary outlet and deter existing vendors from listing more of their products.

2.5.2 Pricing Plan for SHOW Marketplace

The proposed Pricing Plan for the SHOW Marketplace is explained in Table 3. Since the Marketplace will be a free-to-enter online platform, the relevant pricing policies used

appertain to product items and services that will be offered either directly from SHOW (beneficiaries) or from other service providers. The pricing plan of the Marketplace will be flexible, as different pricing policies will be used for different product items to better suit the needs of the service providers and the clients as well. Moreover, a prospective pricing plan for the Marketplace product items is presented in Table 7 and in Appendix I. Note that the pricing plan will be shared with potential customers, although it will not be realized during the course and duration of the SHOW project. Consequently, the proposed business models exists as part of the SHOW Marketplace research and innovation.

Table 3: Pricing Plan for Marketplace

CPS Commission (%) - will not be used at the early business stages and definitely not in the course of the SHOW project.	**5%-15%** per sold product item/service (offered from various service providers).
SaaS License model or Perpetual License model.	For the commercial services that are being developed and will be sold directly from the SHOW marketplace, a certain price will be set. The pricing policy for each service has not been finalized yet.
Listing Fee (will not be used at the early business stages and definitely not in the course of the SHOW project) for the services and product items developed and sold from other service providers.	**0.10€-0.20€** per added product item/service (listings will remain active for 4 months). The new listing can be renewed automatically, or the provider must renew it manually.
Payment processing (not yet defined if it will be utilized).	**0.30€** (or a percentage fee) per successful transaction.

Calculation for revenue commission is done with the following equation:

- **Total Service Value x Commission Percentage = Total Commission**.

For example, if a service provider sells 500 licenses of a certain service hosted on the marketplace and the service value is 30€, then the total revenue amount the marketplace will receive is: (30€x500) x 0.15%=2,250€.

At its early business stages after its commercial launch, the Marketplace could only generate revenue through the License pricing model of the services developed within SHOW project. Since the pricing model for each service has not yet been defined and while some of the services will be free of charge, it is hard to give a certain billing plan for each service respectively. At a later stage and after attracting a sufficient number of external to SHOW service providers that will sell their product items through the Marketplace, the Commission Revenue Model will also be one of the main revenue streams for the SHOW Marketplace.

In addition, the Listing Fee model will not be utilized at first as it might be discouraging for new service providers to upload and sell their products through the marketplace and for existing service providers to add new products and services. At a later stage also, a Cost per Click billing fee can be used whenever a customer is redirected through the Marketplace to the service provider's website.

2.5.3 Other Revenue Streams

A revenue stream depicts a certain source of income for a company or organization, and it consists of different types (e.g., transaction-based revenue, project revenue, service revenue etc.). The SHOW Marketplace will mostly generate revenue at first through the Licensing pricing models for each service and at a later stage from the Commission revenue model. Except for the above sources of revenue, several other exploitable streams to generate income are the following:

- Shareholder contributions
- Sponsoring
- Donations
- Advertising: Revenue generated from charging fees for product and service advertising
- Licensing: Revenue from charging for the use of a protected intellectual property
- Marketplace promotions (sponsored products, stores, etc.)
- Revenue generated through affiliate or referral marketing

2.5.4 Revenue Model Framework

As described in the previous sections, a revenue model is a well-thought-out monetization strategy that is designed in a specific way to assist companies and organizations to generate sufficient income. A revenue model framework can assist to better demonstrate the revenue generated from a certain product or service. The revenue model framework (Figure 1) outlines the five categories of choices that together make up a revenue model, where each category is formulated as a question around payment [11].

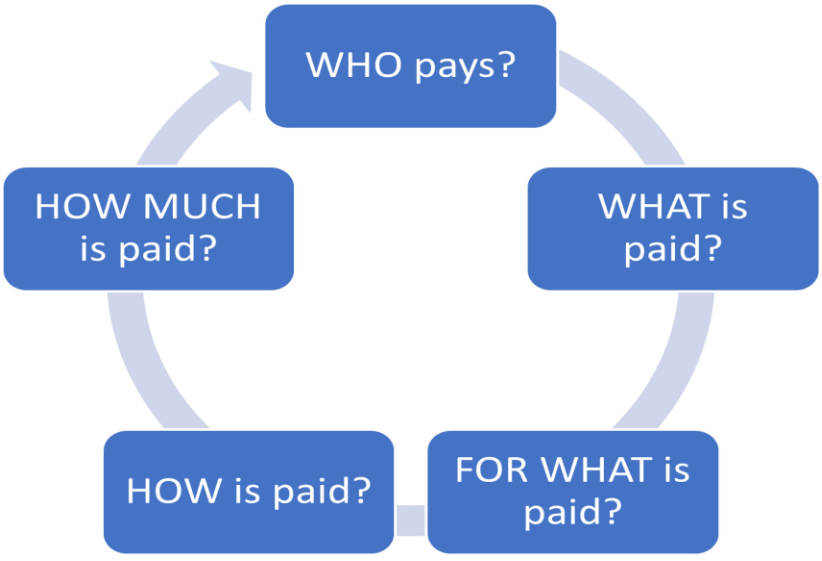

Figure 1: Revenue Model Framework [11].

Since the pricing policy for each service has not yet been finalized, a certain revenue model framework cannot be presented.

2.5.5 Business Metrics for Model Validation

A business model depicts the different methods, ideas, and assumptions, on how a product or a service, or even a whole organization, can provide a viable solution and deliver strong value propositions to its respective customers. An important step towards the success of this solution is the validation of the business model and especially the value propositions and the revenue model. A critical task of this validation process is to engage with customers and to keep track of their feedback, which can happen through different ways, such as [12]:

- Market research
- Demonstrating how the product works under real conditions and by communicating directly to customers, can help find the user needs and pain points that need to be evaluated and addressed
- Conducting surveys, queries, and interviews to receive and keep track of the customer feedback

The next important step towards validating the business model is to measure the different revenue options and the cost structure, by calculating different core metrics that should revolve around the profitability, the growth, and the scalability of the revenue model. These objectives will give an overview of how income can be generated and how costs can be minimised. The following metrics can be used to further validate the revenue generated and the business model in general [12][13].

- **Sales revenue**: tracking the overall revenue can help measure the overall performance of the Marketplace and its product items.
- **Customer Acquisitions Costs**: this metric provides information regarding the necessary expenses for attracting and acquiring new customers (e.g., advertising costs).
- **Lifetime Value**: this metric can only be useful if the SaaS license model for the services will be used, since it provides information regarding the duration customers remain active and continue to contribute to the Marketplace.
- **Net worth**: it can be a valuable metric to measure the return on investment (ROI), since it is the ratio of lifetime value to the customer acquisition costs.
- **Churn rate**: this metric helps to calculate the number of customers who cancel their subscriptions for a given period (this is applicable on the SaaS model).
- **Customer Engagement:** the sales revenue and the churn rate metrics can provide useful information regarding the user's engagement to the platform and can help measure if what the Marketplace is offering and how serves the respective customers, need improvement. A powerful technique to assess the user needs is also the customer feedback.
- **Customer Feedback and Satisfaction:** keeping track of the customer feedback is essential for the success of the whole Marketplace. An important feature that the Marketplace already offers is the review and rating system of every product item, and the "My favourites" feature. These features can provide valuable information regarding the quality of the offered services and the satisfaction levels of the respective customers.

The main methods utilized in this scope will be presented thoroughly in D6.3 (M40). However, some of the methods that can be further exploited to assess the impact of the offered services and to evaluate the performance of the above metrics, are the following:

- Development of a detailed and well-structured business plan.
- Cost-Benefit analysis and/or Cost-Effectiveness analysis.
- Willingness to pay analysis.

2.6 Key Partners

The relevant partners and stakeholders involved in WP6 are presented along with their participating effort in the SHOW Grant Agreement [6]. In addition, the partners that are involved in the development of the Marketplace infrastructure but also in the implementation of the prototype solutions developed within the SHOW project ecosystem and hosted in the Marketplace, have been thoroughly presented in D6.1 [1]. Thus, there is no need to mention them once again.

2.7 Cost Structure

Cost structure is the combination of various types of costs (fixed and variable), that give an overview of a business' expenses [14]. The cost structure together with revenue streams, help assess the sustainability and the operational scalability of an organization [15][16]. Since the SHOW Marketplace is still under development, not all fixed and variable costs have been identified yet. In addition, it is hard to estimate the revenue flow at this stage, but only a few examples are given of how the cost structure would be. Indicatively:

- Initial investment required.
- Investment in infrastructure (physical and digital).
- Recurring cost of physical and digital infrastructure.
- Technical installation.
- Personnel costs (operation, R&D, education, etc.).
- Marketing and Communication.
- Other costs (Business Licenses/Permits/Fees, travel, etc.).

2.8 Business Model Canvas

To better depict and understand in a more straightforward and structured way the business model for the SHOW Marketplace, a Business Model Canvas [3] was created, which is presented below in Table 4**.**

Table 4: Business Model Canvas

<table>
<tr><th colspan="3">Value Proposition</th><th colspan="2">Main Challenges</th></tr>
<tr><td colspan="3">The SHOW Services Marketplace creates value by serving as a:
▪ 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.
▪ Easy-to-consume overview of available product and data items.</td><td colspan="2">▪ Openness.
▪ User acceptance.
▪ Pricing, affordability, and methods for payment.
▪ Personalization of the SHOW Marketplace.
▪ Lack of business and financing models for each service.
▪ Provide maximum value to each stakeholder.</td></tr>
<tr><th>Customer Segmentation</th><th colspan="2">Distribution channels/Key Activities</th><th>Key Partners</th></tr>
<tr><td>▪ Passenger.
▪ Tier 1 Supplier.
▪ OEMs.
▪ Public or Private Transport Operator.
▪ Public Transport Authority.
▪ Research/Academia.
▪ Technology/Service Provider.
▪ City/Authority.
▪ Traffic Management Centre Operator.
▪ Infrastructure Operator.
▪ Association.</td><td colspan="2">▪ SHOW Marketplace.
▪ SHOW project web page.
▪ National and international conferences.
▪ Events, workshops, and webinars.
▪ Social media campaigns.
▪ Paid advertising.
▪ Email marketing.
▪ Publications.
▪ Search engines (SEO optimization).
▪ Offline channels.
▪ Public affairs activities.
▪ Case studies, one-pagers, etc.</td><td>The relevant partners and stakeholders have been thoroughly presented in D6.1 and the SHOW Grant Agreement.</td></tr>
<tr><th colspan="3">Business Revenue Model</th><th colspan="2">Cost Structure</th></tr>
<tr><td colspan="3">▪ Commission-based model (will not be used at the early business stages): Cost per Sale percentage fee.</td><td colspan="2">▪ Initial investment required.
▪ Investment in infrastructure (physical and digital).
▪ Recurring cost of physical and digital infrastructure.</td></tr>
</table>

▪ Revenue from the Licensing pricing models (SaaS or Perpetual license model will be adopted) for the commercialized services. ▪ Listing Fee model (cannot be used for the duration of the SHOW project and for SHOW stakeholders): a fixed fee for adding a service. ▪ Payment processing fee (not yet defined if it will be utilized): charge of a certain amount for every successful transaction.	▪ Technical installation. ▪ Personnel costs (operation, R&D, education, etc.). ▪ Marketing and Communication. ▪ Other costs (Business Licenses/Permits/Fees, travel, etc.).

3 New development features

In this section of the deliverable, the new features that have been developed during the previous period (M22-M30) are described. These new features include payment, rating and the “favorites” page, in accordance to the requirements of the SHOW Marketplace as presented in D6.1 [1].

3.1 Payment

3.1.1. Payment options

This subsection describes the process by which the user can purchase one or a bundle of products in the Marketplace. As shown in Figure 2, a box in the right side contains a dropdown menu, the total price, and a “Proceed to Checkout” button. It is important to note that only authenticated users (Logged in) are eligible to purchase a product from the Marketplace.

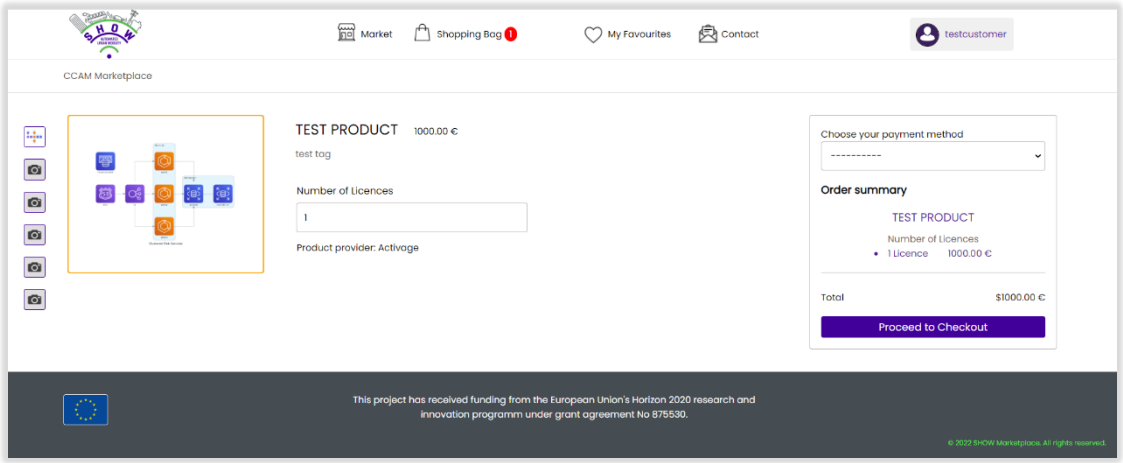

Figure 2: The “purchase” page of the show marketplace.

The user in this screen can choose from three available dropdown menu options to complete the financial transaction. Figure 3 depicts the available options, which are the following:

- Credit card.
- PayPal.
- Bank transfer.

These three options are the most common and reliable ways to complete transactions in any marketplace, and the public is accustomed to having them at their disposal. Furthermore, the option use is conducted in almost every payment system integration, ensuring the system confidence without excluding any user from buying a product from the Marketplace.

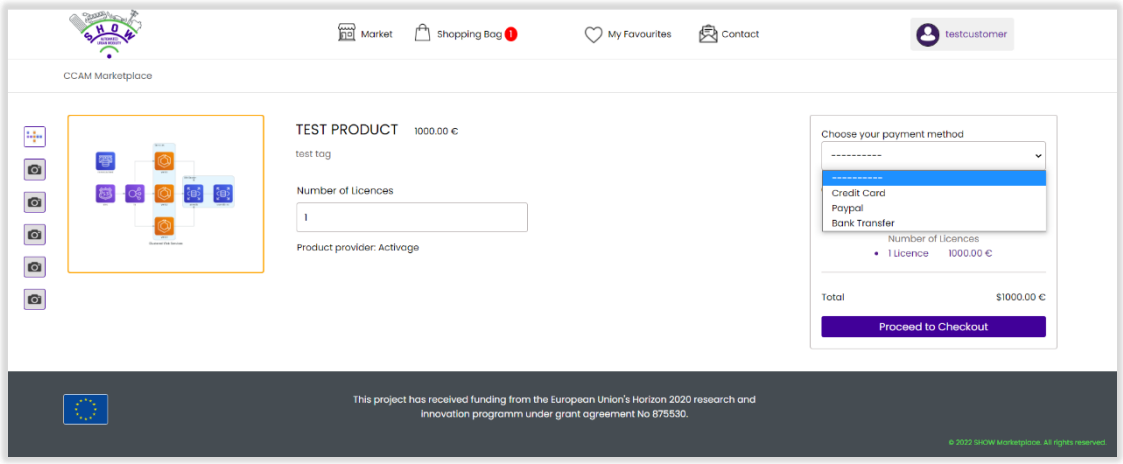

Figure 3: The selection of the different payment options.

The following paragraphs present an extensive analysis of each payment option. For the purposes of the SHOW Marketplace demonstrating and testing, a nominal account was used to replicate the users' actions. The account is visible in the top right corner, in the "username" box.

3.1.1.1. Credit Card

This paragraph describes the transaction by using the credit card option. By selecting the credit card option and pressing the "Proceed to Checkout" button, the user is redirected to the "/creditcard" section of the Marketplace, which can be observed in Figure 4.

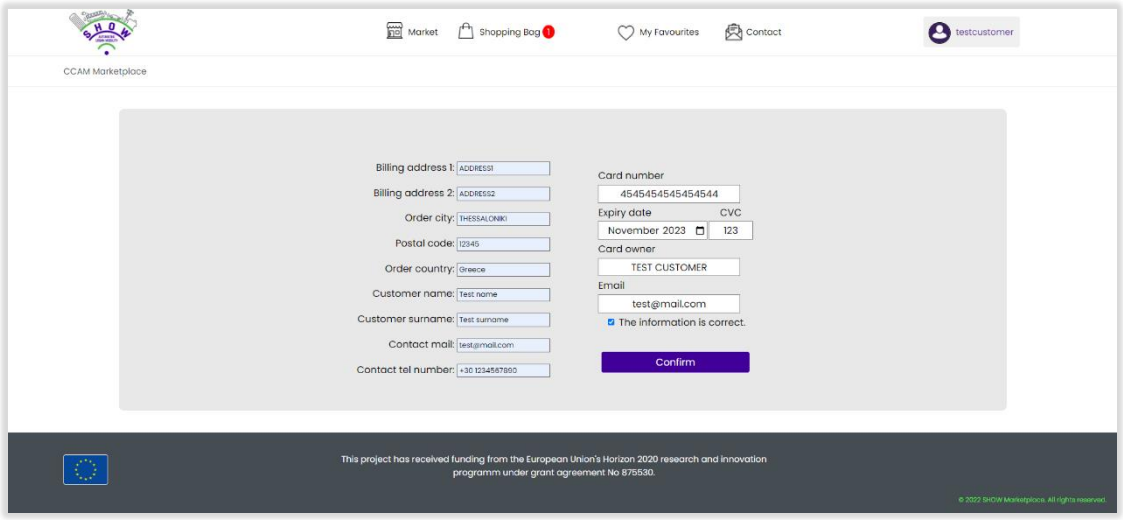

Figure 4: The section '/CreditCard' of the SHOW Marketplace with filled input fields.

The completion of the transaction mandates users to fill in all the required fields and to confirm the validity of the information inside each input field, by selecting the checkbox above the confirm button. The only optional field is the "Billing Address 2" field. After filling out all the required fields, the user must press the confirmation button. By doing that, a sequence of events occurs. The final integration of the credit card payment system is to be completed by the next iteration by setting a bank account for online payments and the provision of a wide range of data (final integration of the marketplace, tax statements, etc.) to the bank.

There are various ways to interact with a bank's server in order to support online payments. Two of the most well-established are the simple interaction and the direct API call interaction.

- The "simple interaction" option is a button provided by the bank that can be added to the User Interface of the application. The button redirects the users to the bank's safe environment for committing the payment securely. This option supports instalments and is the most common way to integrate a transaction using a credit card.
- The second option is to use direct API calls, where the provider must process the payment on the local server before sending the data to the bank for transferring the monetary amount of the transaction. This implementation is not optimal for the SHOW Marketplace integration, unless the provider (product seller) wants to provide special escalated instalments. This option is more complex and can pose complexities as the server administrator must be fully compliant with the PCI standards for hosting and processing bank transactions and shoulder responsibility for any errors. This option has future overheads as the system must be updated all the time in order to comply with regulatory changes to the EU's personal data law or PCI standards. Hence, the first option is favored.

By completing the transaction, the bank sends a request to the server with information regarding the transaction. This information is stored locally as a database model.

3.1.1.2. PayPal

This paragraph describes the transaction using the PayPal option. PayPal is widely accepted and common for facilitating online markets. The popularity of this payment method relies on features such as ease of use, safety and transaction speed. Users choosing this payment method via the payment button are redirected from the merchant site to PayPal and then back to the merchant site.

According to the implementation of the SHOW Marketplace, the user can transact using PayPal, as depicted in Figure 5.

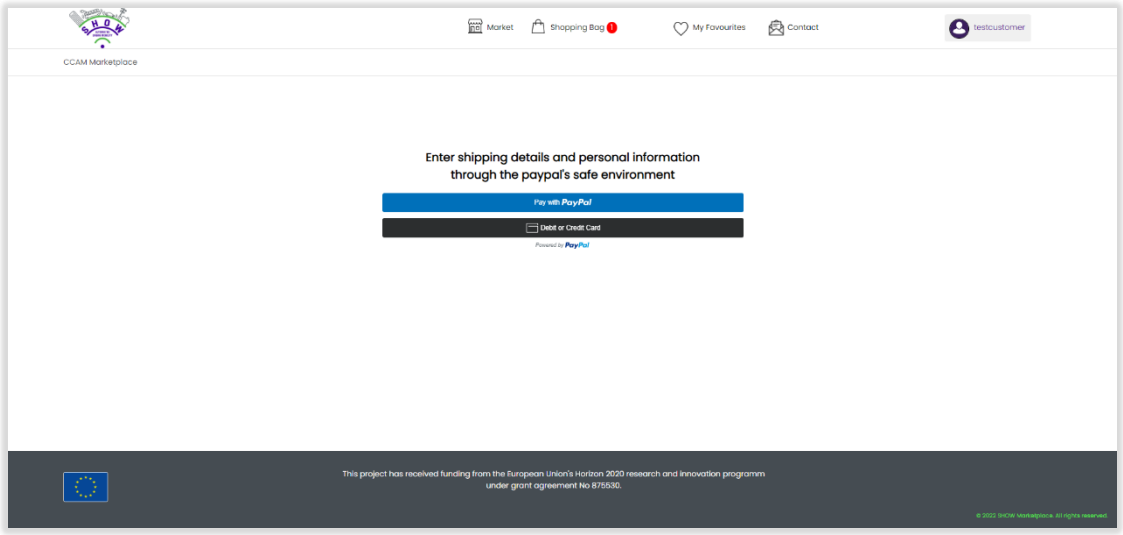

Figure 5: The '/PayPal' section of the SHOW Marketplace.

By clicking the blue button titled "Pay with PayPal", a new window appears. This new window is generated within PayPal's safe environment (Figure 6). Depending on PayPal's server load, it is possible to have a small delay (in seconds) in the appearance

of the pop-up window. Additionally, the user's internet connection is another reason that could cause delays in the window response time.

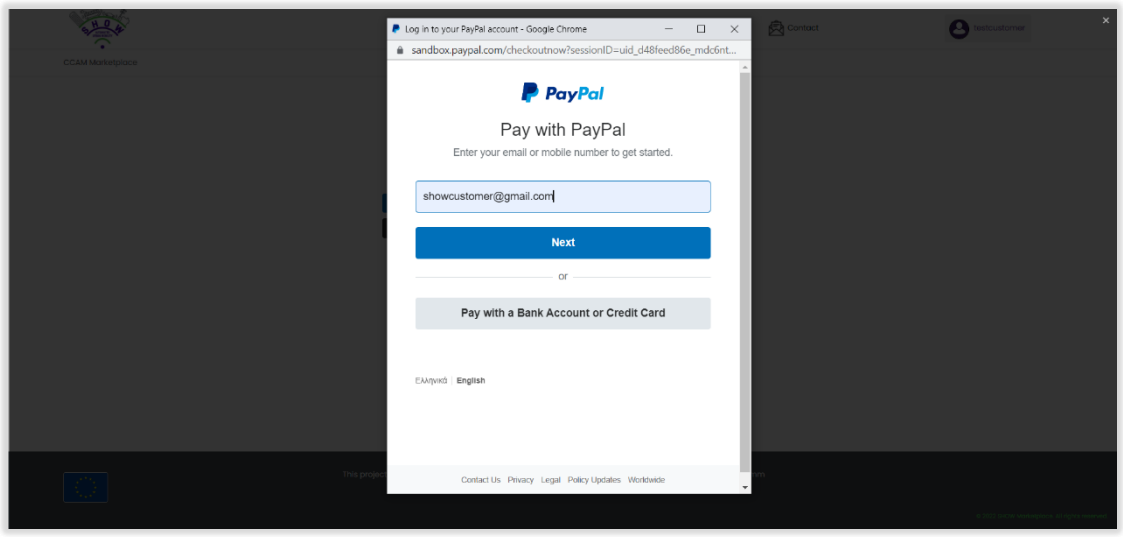

Figure 6: Logging in to the PayPal personal account to commit a payment.

The next step of the procedure is presented in Figure 7. It is important to elaborate more about the "Send to" option. This option contains the input fields, such as the shipping details, contact information, and more, that the user must fill in the transaction completion.

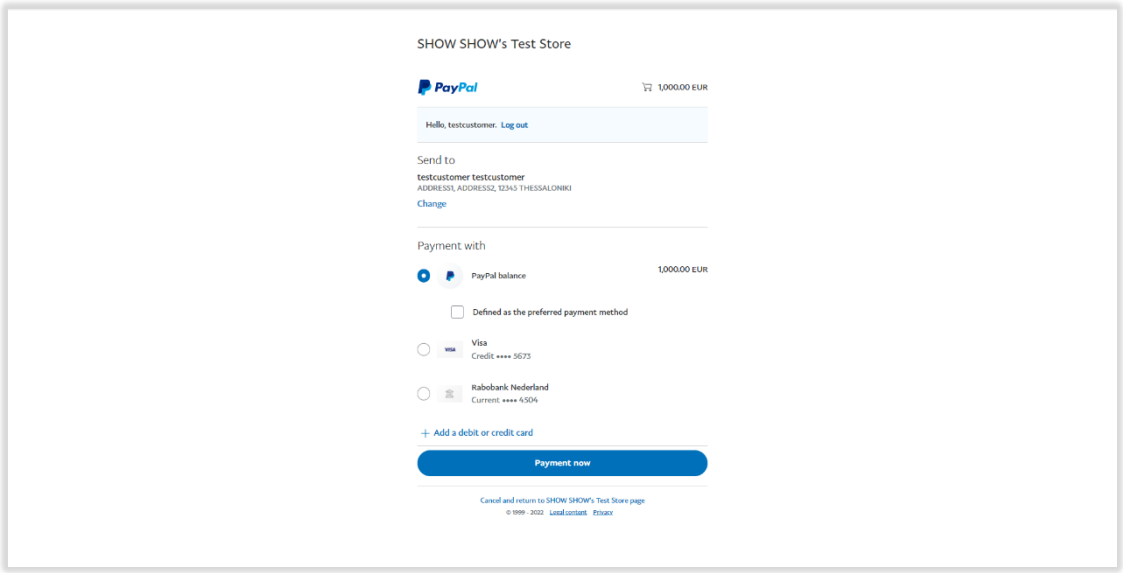

Figure 7: Committing the payment using PayPal.

In the same screen, the user receives information about the total amount of the transaction and the remaining balance of their account. The transaction is completed by clicking the "Payment now" button. After that, the "/Paypal" section is displayed in Figure 8.

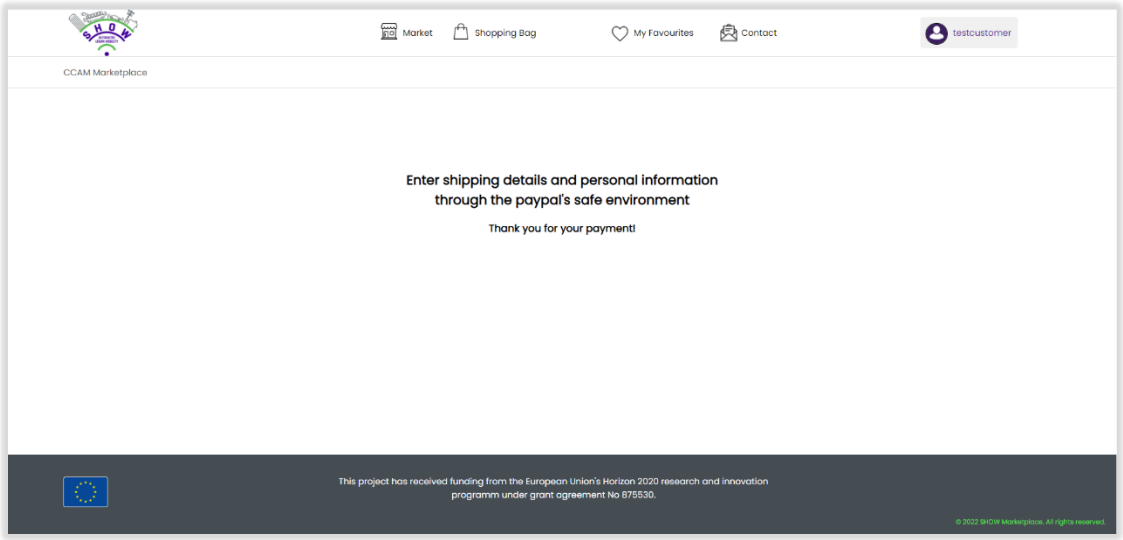

Figure 8: The 'Thank you' message displayed after a successful transaction using PayPal.

In case the transaction fails to occur, the user is redirected to the previous page, and the console presents an alert message notifying "*Order cancelled by user*". By completing the transaction, PayPal sends a request to the server containing information regarding the transaction. Similar to the previously described credit card option, this information is stored locally as a database model.

Regarding the technical aspect of the integration, PayPal offers many options and implementations. First, PayPal's website provides the methods for a safe environment redirection. Moreover, a business account is mandatory in order to use these buttons, which is also responsible for the payment receipt. All PayPal Payments Standard buttons are created using the "Create a PayPal" payment button page and are protected automatically. There are two available options regarding security: a) payment data are saved by PayPal and only a single button ID is shared between the merchant website and the PayPal and b) the payment button code is encrypted before displayed on the merchant website. However, the aforementioned applied protection involves only the PayPal server; therefore, the SHOW Marketplace shall apply its own security mechanisms.

The whole integration of the Marketplace strongly depends on secure transactions for the users. PayPal may provide encrypted buttons, but the encryption is about PayPal's safe environment and not the transaction itself, as mentioned earlier. Hence, a two-step validation is set on the SHOW Marketplace's server by applying security measurements from the Marketplace side. In the first step, the server recalculates the total price synchronously, and the result is placed on PayPal's integration. This ensures that in case of price changes, there will not be a false charging in the system. The second step occurs once the transaction is performed (considered valid), when the server checks again for any changes in the total price. In case of a false price, the transaction will not be completed successfully. This technique is widely applicable among e-commerce providers for securing their platform.

When a transaction takes place, PayPal sends an HTTP "POST" request to the SHOW marketplace server, for the payment details to be saved and processed by the server. A vulnerability with respect to this communication method is that anyone can send a POST request to the server. Thus, a secure way of posting to the server must be established so that only PayPal can POST this kind of data. For this reason, the CSRF

(Cross Site Request Forgery) protection middleware was used to protect the system against CSRF.

3.1.1.3. Bank Transfer

This paragraph describes the transaction using the Bank Transfer option. The user gets the banking information of the provider, and after filling out a form, a transaction object is created. Afterwards, the buyer will have to deposit the required amount to the provider's (Marketplace) bank account and the system administrator will allow the transaction to continue in the same pattern as if it was submitted via PayPal or Credit card. The process is presented in Figure 9.

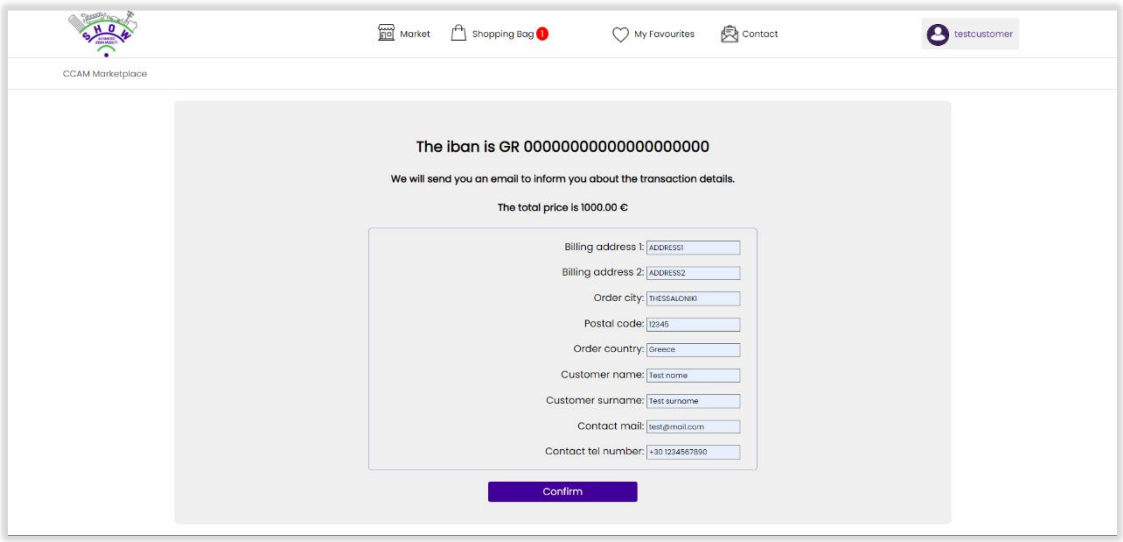

Figure 9: The '/banktransfer' section of the SHOW Marketplace.

Once the user fills all the required input fields and presses the confirm button, is redirected to the "Thank you!" page (Figure 10), similar to the other options.

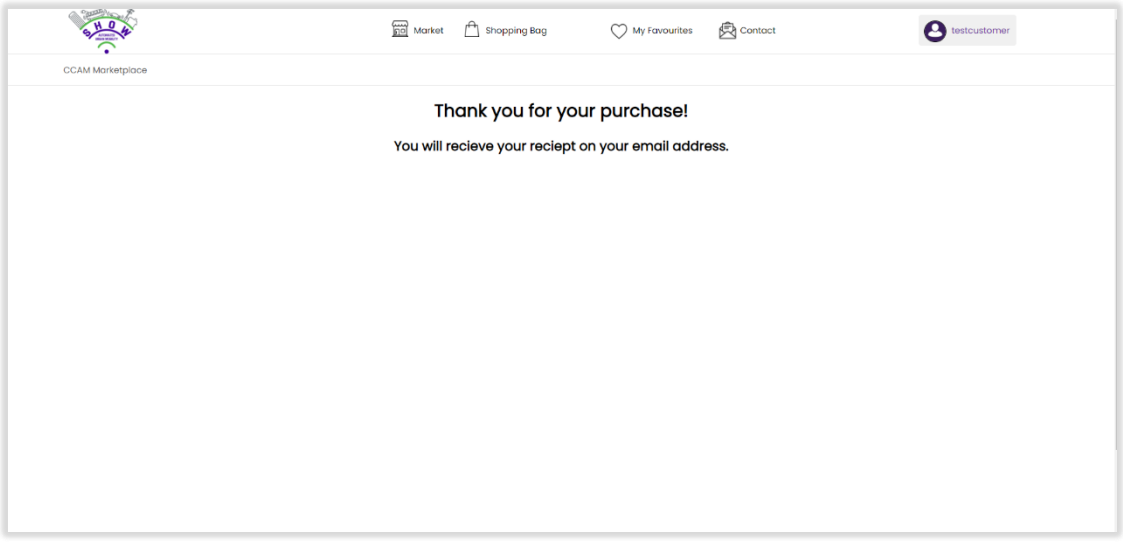

Figure 10: Redirection to the "/ThanksPage" after a successful transaction.

3.1.2. Email confirmation

After each successful purchase, the buyer receives a confirmation email. This approach is the most common way to send the transaction details to the customer,

along with other information, such as invoices or files. This process is highly reliant on admin privileges, on the provider's local server, on the internet connection and on the email provider. Hence, local computing environment highly affects this process and may encounter security issues that the IT department can easily solve on the server provider's side.

For ease of use and a better understanding of the email confirmation, a Google account is used for the mailing system. This account can be tied to a corporate mailing account for management and administrative purposes. However, this method provides better security and more flexibility in its use. Creating an email confirmation system on a corporate server can lead to many security vulnerabilities making the use of an intermediary Google account a reliable option. Therefore, using a programming language to complete the email confirmation contains the following steps:

- Connecting to an email server.
- Confirming connection.
- Setting a protocol.
- Logging in.
- Sending the message.

For this functionality, Python's built-in "smtplib" library was used, that simplifies these steps with function calls. Each major provider has its own SMTP (Simple Mail Transfer Protocol) server that is worth mentioning in Table 5.

Table 5: Email providers list and their respectful SMPT server domain names

Provider	SMTP server domain name
Gmail (will need App Password)	Smtp.gmail.com
Yahoo Mail	Smtp.mail.yahoo.com
Outlook.com/Hotmail.com	Smtp-mail.outlook.com
AT&T	Smtp.mail.att.net (Use port 465)
Verizon	Smtp.verizon.net (Use port 465)
Comcast	Smtp.comcast.net

The SMTP server is a domain name to which the user connects when they try to access the email programmatically. Gmail is currently one of the most popular email-hosting servers. For using a Gmail account, a specialised app password needs to be generated in place of the normal password. This prompts Gmail to know that the right user (owner of the Gmail account) authorizes the Python script attempting to access the account.

The email system is also used for products that do not present their prices. When uploading a product, the option "Contact the provider" may be added. In this case, any potential buyer will have to ask the price or any additional product information. The user interaction with this feature is shown in Figure 11 below.

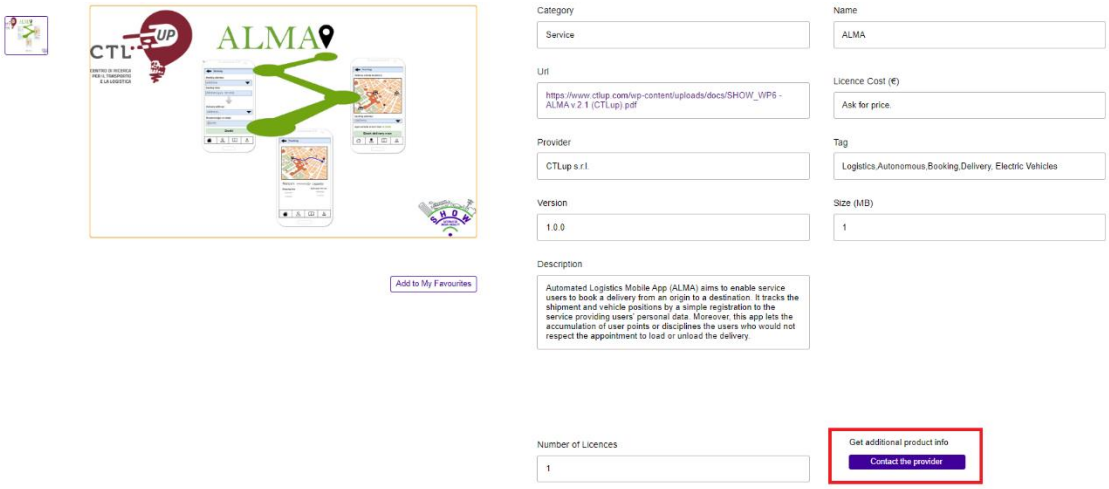

Figure 11: The 'Contact the provider input field in the '/product_info' section of the SHOW Marketplace.

Clicking on the 'Contact the provider' button, the user is redirected to the page shown in Figure 12.

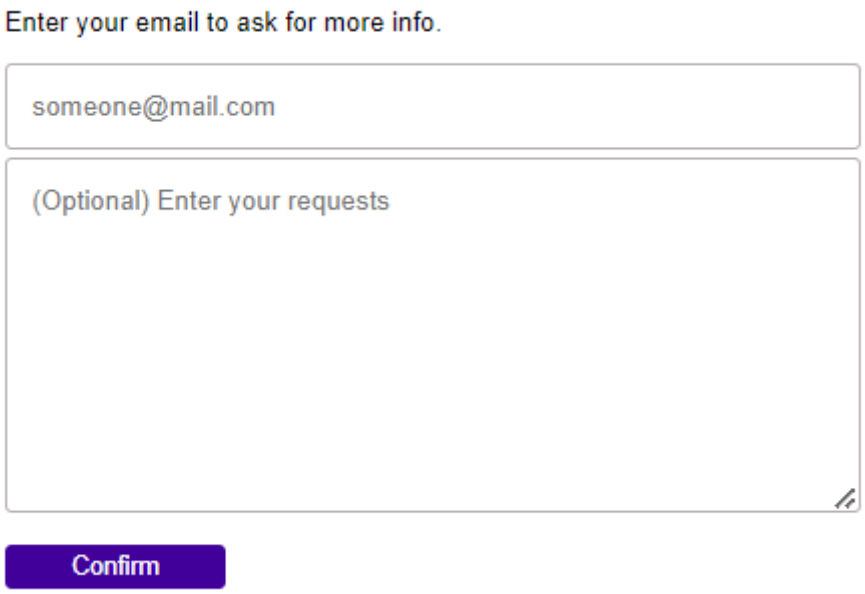

Figure 12: The 'Contact the provider' user interface.

In the form presented in Figure 12, the email field is a mandatory filling option. The text area below is a non-mandatory field where potential buyers may ask questions about the product or any additional information needed, not covered by the product description field.

After submitting the email address (and potentially some further questions or information), the buyer can be assured that the provider has been informed and they will receive timely feedback. The notification functionality takes place on the server side of the SHOW Marketplace. In this particular case, the server acts as an intermediary between the potential client and the service/product provider. Specifically, the client's email address and possible questions are composed into a new email that is sent to the provider. This email has the following pattern:

<< The customer with the {submitted email} has send the following questions: {questions} about the product {name of the product} >>

In this manner, the provider can answer any possible questions by contacting the potential client who submitted an email in the aforementioned form directly.

3.1.3. Products free/open source

In many cases, some products are provided without any charge. In this case, the 'Buy', 'Add to card' and 'Keep shopping' buttons are not present. Instead, there is a 'Download' button available that redirects to the download page of the respective provider. Figure 13 displays the respective screen page.

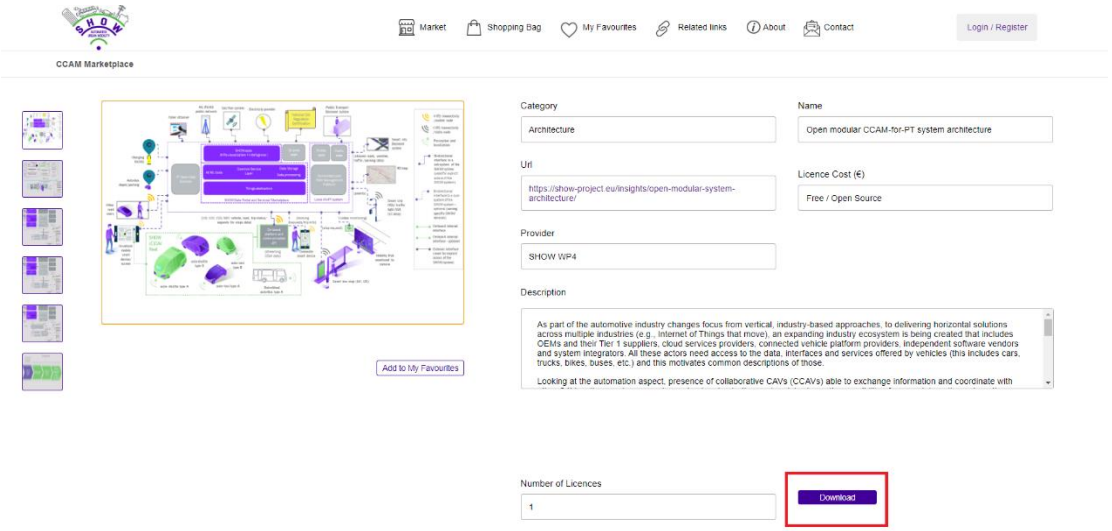

Figure 13: Download option for the free/open-source products.

3.2 Transaction handling

3.2.1 The backend system

When the transaction is complete, the server saves the transaction data to the database. The database preserves the order transaction model and creates a new model for each individual product in the transaction's shopping cart. Since the shopping cart has multiple product functionality (the user can buy different products by different quantities), the modelling structure follows a one-to-many implementation.

For better understanding, the unique transaction model will be referred as order-transaction and the multiple models that are connected to the unique will be referred as order-product. The following chapters include an explanation regarding the fields of those two models.

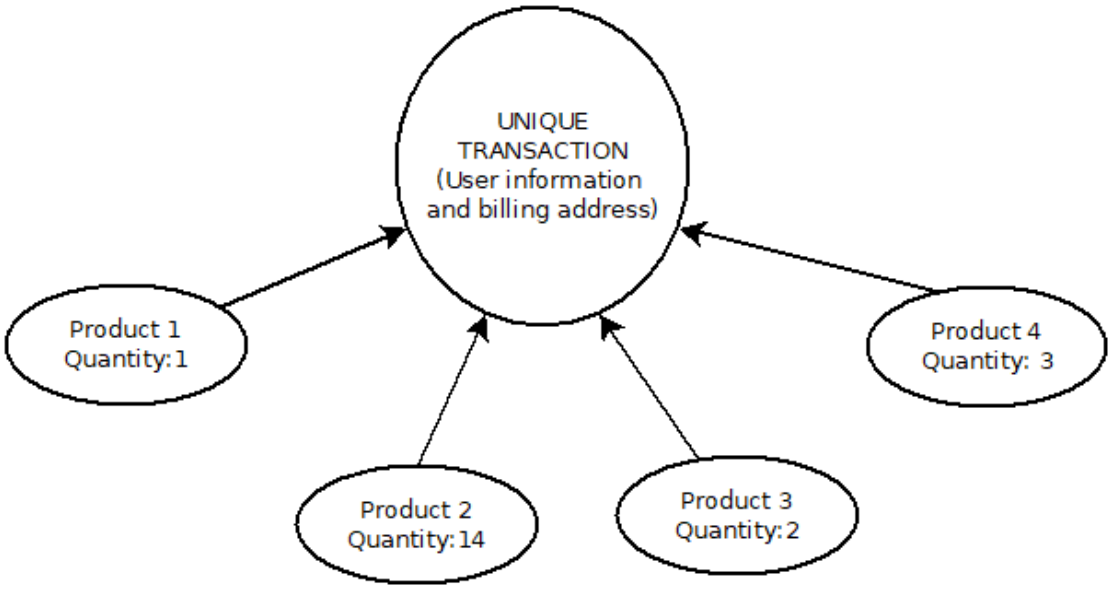

Figure 14: One to many modellings structure for the transactions.

3.2.2 The transaction model (Order-Transaction)

Once the transaction is complete, the server saves the transaction data to the database. The saved transaction data are characterized by two types, as follows:

1. User oriented data, such as the user's profile, billing name, surname, telephone number and email (in case the billing name is different from the profile's name), along with all the address information (country, city, primary and secondary addresses, postal code, etc.).
2. Transaction information, including transaction ID (Primary key and unique transaction ID), transaction creation index, total price, payment method (automatically configured field) and the order status (this field depends on the final billing configuration and may be complete/declined/processing/aborted, etc.).

3.2.3 The transaction product (Order-product)

Given the registration of the order transaction model in the database, a model is created for each different product on the shopping cart that was included in the transaction. In this case, the fields include:

- The unique order-product ID (typical primary key).
- The unique product ID.
- The unique order-transaction ID (connection to the main transaction)
- The quantity of the product purchased by the user (eg number of licenses).
- The current product price (prices fluctuate in time).
- The current taxes included at the time being.

There is also a 'final price' field that is calculated by the current price, quantity, and taxing rates. A visualisation of the parameters for each model can be seen in Table 6.

Table 6: Transaction models. The order-transaction and the order-product models.

Data models in SHOW Marketplace	
Order transaction	**Order product**
• Unique ID (Primary key)	• Unique ID (Primary key)

Data models in SHOW Marketplace	
• User profile • Customer's contact information • Billing address information • Order details (price, date) • Order progress	• Product ID (Foreign key) • Transaction ID (Foreign key) • Pricing info (quantity, price, taxes, final price)

It is important to note that this implementation may have alterations in the next final Deliverable. The vast variety of existing and future products along with the billing system might facilitate the transaction functionality to increase in scale containing more information, billing plans, subscriptions etc. and thus changing the model fields.

3.2.4 Facilitating transactions

There is a variety of methods that can be used to arrange revenue in an online marketplace, in order to better allocate earnings to each provider selling through the SHOW Marketplace. Reportedly, two of the most common methods are the following:

1. Customer's money goes to the sellers. This way, the marketplace allocates revenues to each service/product provider directly, after withholding the predefined fees. This process usually takes one business day for the money to appear in the seller's account.
2. Customers' money goes to the marketplace. Thereafter, the marketplace allocates revenues to each service/product provider at a moderated and predefined date and time (after withholding the predefined fees as well).

In the case of the SHOW Marketplace, both methods have been examined resulting in the choice of the first method during the current implementation although the final decision will be performed in the final phase of the SHOW Marketplace implementation. According to the first method, the payment functionality is applied in the following manner.

- The process consists of the following steps:
    - First, each individual provider must provide a personally owned commercial key to the Marketplace administration (for example PayPal's public key or a bank account's public key).
    - Next, with this key, the server may be able to switch the revenue receiving address, setting the value based on the desired provider's key.

The SHOW Marketplace is being developed as part of the EU-funded SHOW project, which is an Innovation Action. Therefore, in the course of the SHOW project, revenue cannot be generated. In addition, it cannot operate as a revenue administrator for the relevant partners that aspire to sell through the Marketplace. After the completion of the SHOW project, all the legal, financial, and technical regulations are to be considered, in order to outline a comprehensive and strategic plan on the revenue allocation to the prospective service and product providers. This will most probably be performed through the final conversations related to the SHOW project exploitation plans.

3.3 Rating

3.3.1 The rating system

The rating system is a common feature that many marketplaces possess globally, in order to facilitate product review and rating. Online reviews can transform the way consumers choose products and services. A well-managed review system creates

value for buyers and sellers. Trustworthy systems can give consumers the confidence they need to buy a relatively unknown product. Hence, a well-established reviewing system could work in a very beneficial manner for the proposed SHOW Marketplace.

The Rating system in the SHOW Marketplace fulfils all the criteria of the CRUD (Create, Read, Update, Delete) operations, which compose the cornerstone of every database system. Only registered users are able to interact with the reviews section of the SHOW Marketplace, for enhanced security. Otherwise, the reviews section is a read-only field and the write, edit and delete options are unavailable.

3.3.2 Writing a Review

This paragraph describes the functionality of writing a review for a specific product in the SHOW Marketplace. The reviews are located on the bottom side of the '/product_info' section on the Marketplace, as shown in Figure 15.

Figure 15: The review section.

By clicking the 'Write a review' button, the window resizes and adds the review input fields, as shown in Figure 16.

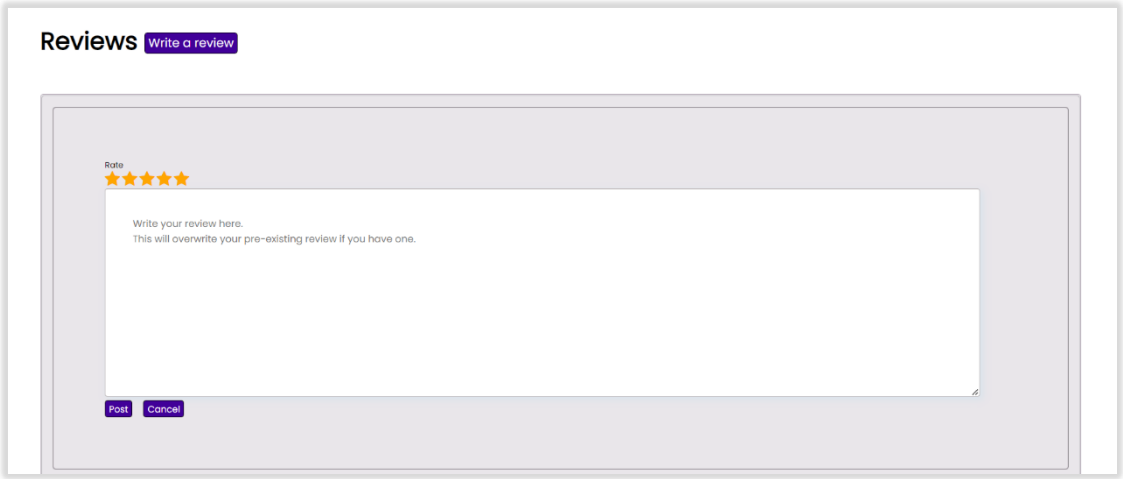

Figure 16: Input fields for writing a review.

The reviews are comprised of two elements:

- The product score, based on the submitted reviews, is indicated by yellow stars. The user selects the number of yellow stars, each representing one review point, with a maximum of five.
- The potential review text describes the product in a more detailed manner. The user is encouraged to write a maximum of 2000-character text to review the product and its use.

It should be noted that the review score is a required field, but the review text is optional. Writing a review is intended to be simple, as shown in Figure 17.

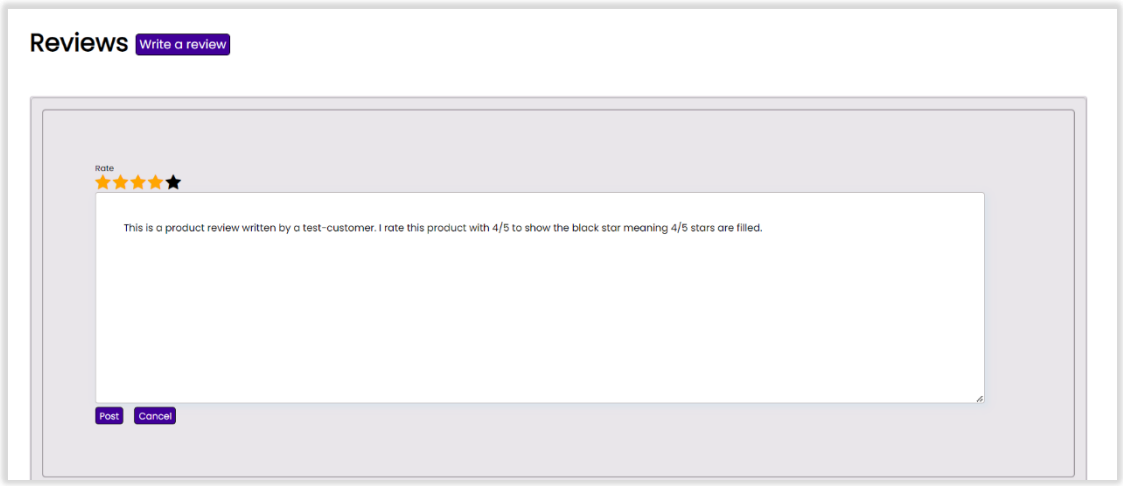

Figure 17: Writing a review for a specific product.

By pressing the “Post” button in the bottom left of the review field, the user is able to post the current review. Then, the user is redirected to a page with the message, “Thank you for your feedback”, as shown in Figure 18.

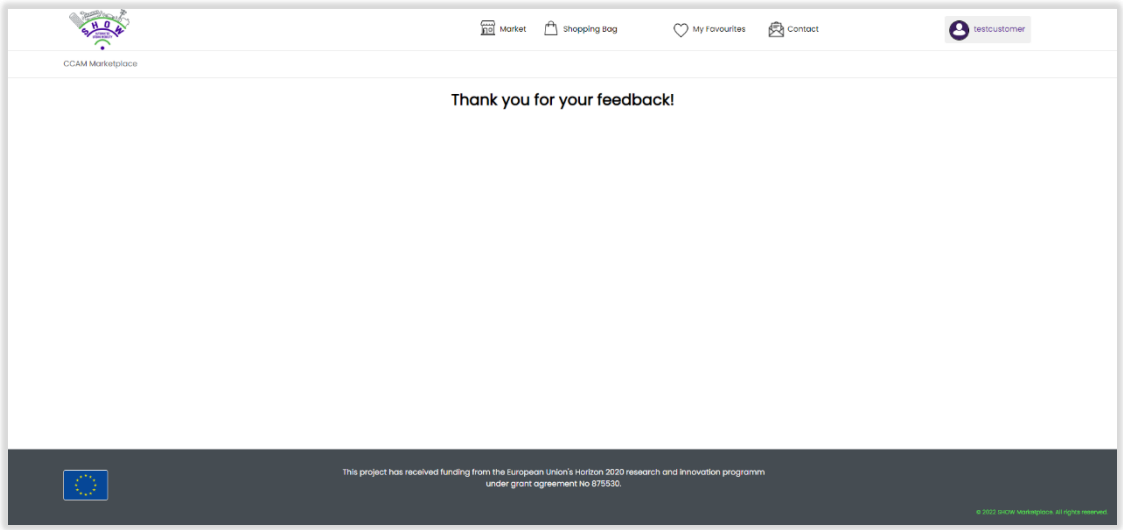

Figure 18: The 'ThanksForFeedback' page of the show marketplace.

There is the capability for a user to change or delete the rating and review for a product when revisiting it. As shown in Figure 19, there are two additional buttons at the user’s disposal for updating or deleting the review.

Figure 19: The user's recent review.

At this point, it is important to notice that a user is able to review a product only once. If a user selects the ‘Write a review’ option again, the input fields will be shown again,

and the process will be the same. However, the new review will replace the existing one (of the same unique user).

After adding a review, the rating score applied to a product is considered for the calculation of the updated total score. The update rating score is calculated dynamically and in real time.

3.3.3 Deleting a Review

This paragraph describes the functionality of deleting a review for a specific product in the SHOW Marketplace. As mentioned above, security restrictions are applied to the reviewing system. The user is eligible to delete only their own review, while the other user reviews are a read-only field. By clicking the 'delete' button shown in Figure 19, users are able to delete the review they have written.

3.3.4 Editing a Review

Editing a review from the user perspective is similar to writing and deleting review, since it has the same constraints with respect to security restrictions. By clicking the edit button shown in Figure 19, the review input fields appear, while the operation follows the same procedure with the 'Write a review' option mentioned above.

An example of two reviews, written by different profiles, is presented in Figure 20. In this instance, the currently logged-in user is the "testcustomer". Since the second comment is linked to the current account, the user is able to interact with all the CRUD operations, in contrast with the first review that has been written by another user and only read permissions are available.

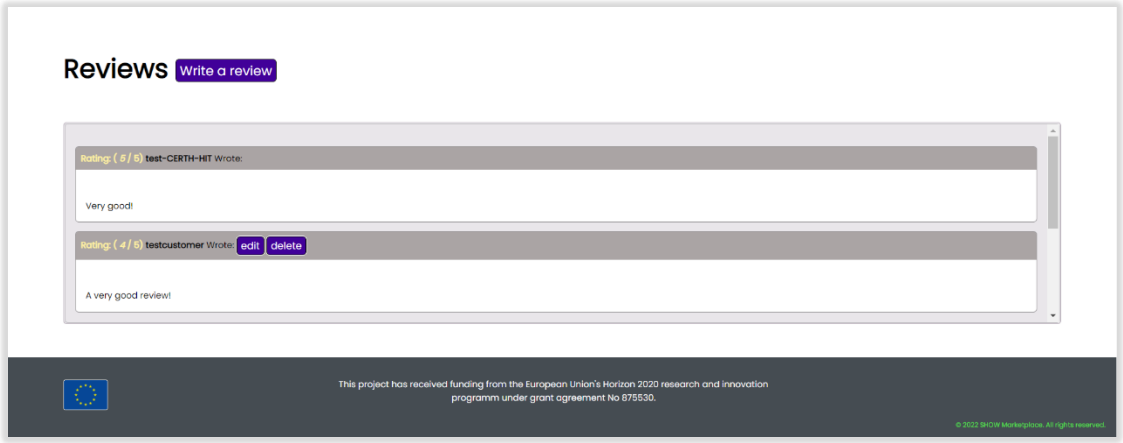

Figure 20: Reviews for a specific product.

3.4 My favourites

3.4.1 Introduction

This paragraph describes the concept and functionality of adding a product to the "My favourites" option. Wish lists and favourites allow customers to recall their chosen product items quickly and easily, without the hassle of navigating through a website. Customers are more likely to return to a store or site when they have used a functionality such as "my favourites" to save items for later inspection. This feature can result in sales improvement and shopping cart abandonment reduction. Moreover, automatic marketing emails could remind visitors of products in their "favourites" list. The addition of a product to the favourites is only available to registered users. Every user has their own favourite product item list when navigating the SHOW Marketplace.

3.4.2 Adding a favourite product

A user may add a favourite product to their page by simply clicking the “Add to favourites” button, highlighted in red in Figure 21.

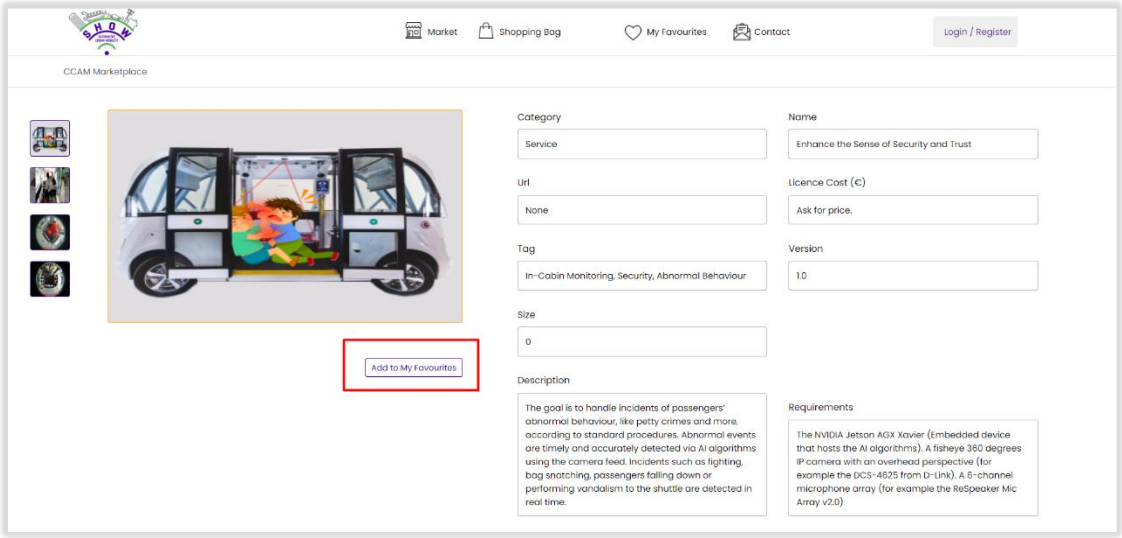

Figure 21: The 'Add to favourites' button in the '/product_info' section of the SHOW Marketplace.

This action sends a request to the server to add the selected product to the “My favourites” list of the current user. Then the user is notified with a confirmation message saying that the selected product is added as their favourite.

Whether an unregistered user presses the “add to my Favourites” button, an error message appears to inform the user that something went wrong. The error message dialogue is displayed in Figure 22.

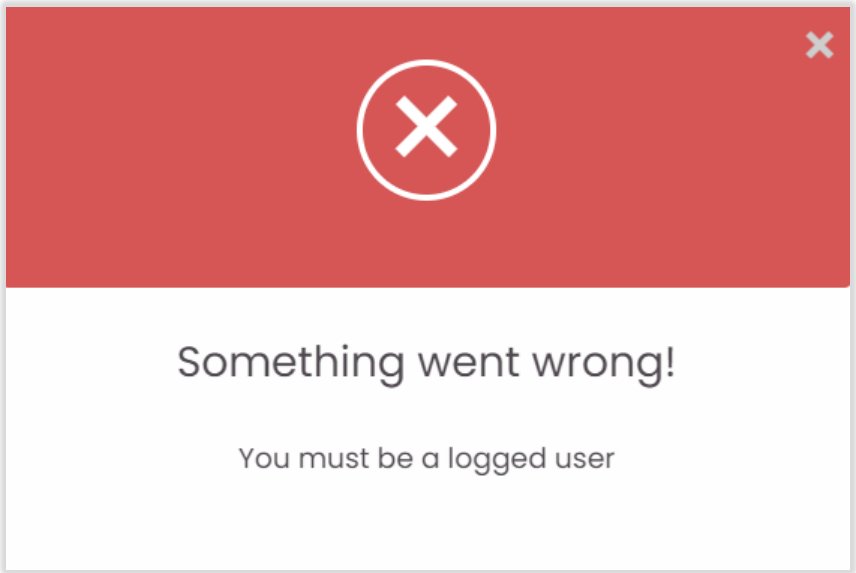

Figure 22: Error message.

After adding product(s) to “my favourites” list, the user can see all the favourite products by clicking the ‘My favourites’ button in the menu bar. This action redirects the user to the ‘/my_favourites’ section of the marketplace, as displayed in Figure 23.

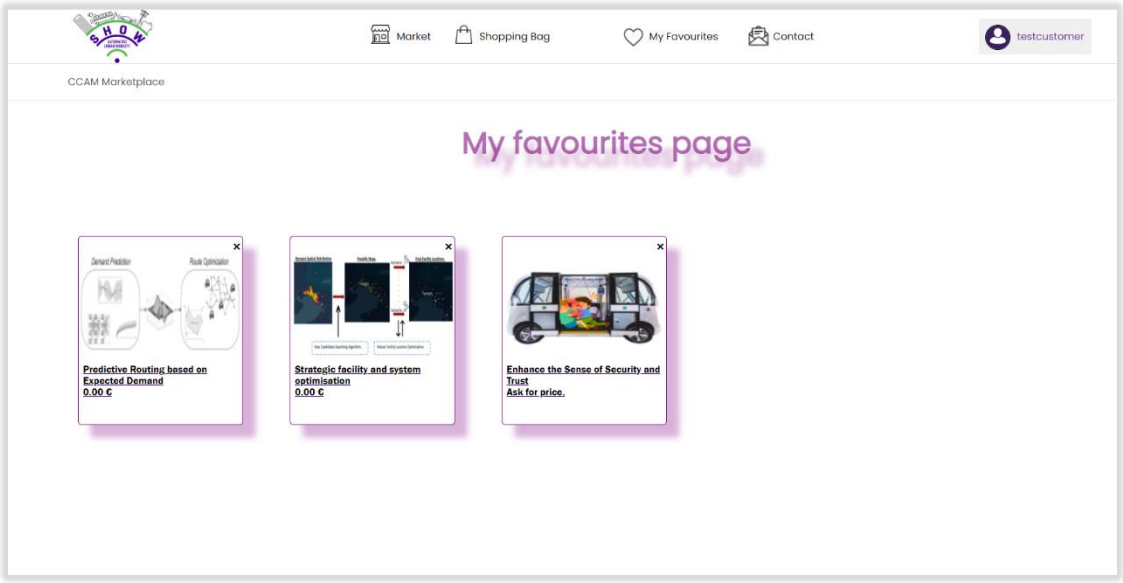

Figure 23: The '/my_favourites' page of the SHOW Marketplace.

The process of removing a favourite product is also simple and intuitive procedure. The user is able to remove a product from its list by clicking the ‘x’ symbol in the top-right of a favourite product, visible in Figure 23. This will update the page automatically, and the product will be removed from the user’s list.

3.5 Other changes to the Marketplace User Interface (UI)

Considering the changes in the Marketplace layout, an “About” page was added in the Marketplace, which contains information about the scope of the SHOW Marketplace and its role as a one-stop-shop for CCAM. Meanwhile, the implemented filters were updated to suit the needs of the hosted products. The UI navigation is developed to be as simple as possible. For the users to find all available services, they need no clicks, as the products are presented in the home screen. The filters help to sort out user preferences, aspiring to provide the desired products with just a few clicks.

4 Development Operations & Infrastructure

This section presents the current trends of best practices, regarding Development Operations (DevOps), that also affects issues related to the infrastructure (in terms of databases and computational resources) of every software application. After a thorough analysis of the benefits and guidelines in order to follow such a strategy, the required interconnection with the SHOW Marketplace is performed.

4.1 DevOps Overview

Development Operations (DevOps) is a practice that promotes collaboration in order to build and deliver secure software as quickly as possible [17][18]. Although DevOps has been thoroughly discussed for a decade, a widely accepted and agreed definition was missing [19] until IEEE released a DevOps standard in 2021 [20]. Dyck et al. [21] described DevOps as a "*collaborative and multidisciplinary effort within an organization, to automate continuous delivery of new software versions, while guaranteeing their correctness and reliability*". However, different organizations have varying definitions, as they expect different things from DevOps practices [22]. Nevertheless, DevOps practices are universally expected to break down the wall between Development (Dev) and Operations (Ops) teams, while automation and collaboration, along with continuous improvement and feedback, accelerate software delivery. Along with human collaboration and reliability, the main principles of DevOps include continuous integration, continuous delivery and continuous deployment through automation, focusing on user needs and iterative improvement.

4.2 DevOps Principles

DevOps was introduced as a complementary to the Agile methodology for software development, as many aspects were inspired by it [19]. DevOps principles focus on customers, collaboration and communications among different teams and departments, and automation of the lifecycle of the software development in order to achieve a swifter time to market.

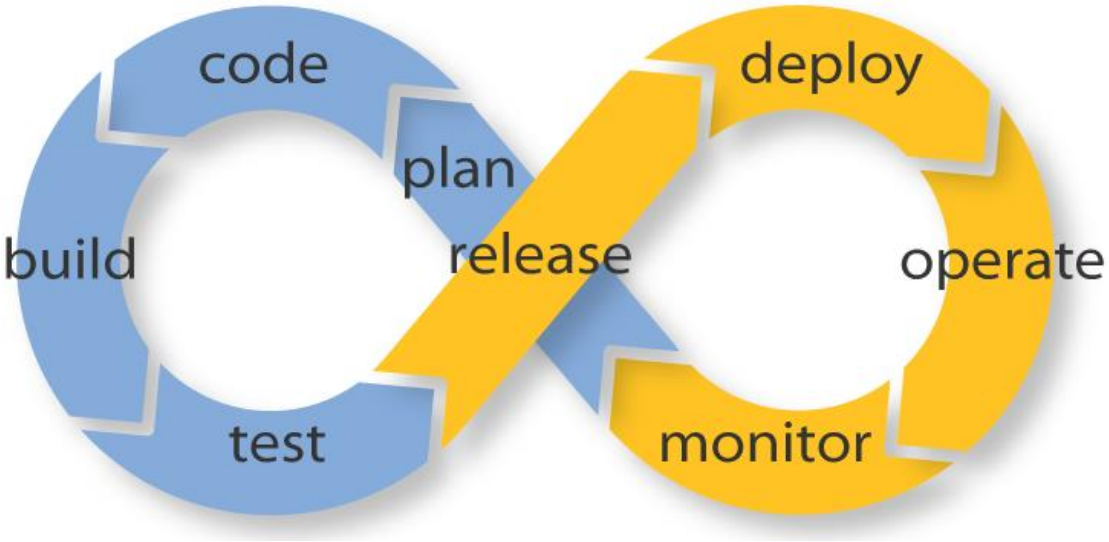

Figure 24: Eight Stages of DevOps Life Cycle [24].

The DevOps lifecycle consists of eight stages of development and deployment, along with two stages that are stable in every phase of the lifecycle [17][23][24]. The two supervising stages are Manage and Secure, while the other eight are Plan, Code, Build, Test, Release, Deploy, Operate and Monitor. Figure 24 presents the eight stages

of the DevOps life cycle and the basic characteristics of these eight stages, along with the two supervising stages, are as follows.

- **Manage**: Every time a DevOps loop is closed, feedback from the entire lifecycle is incorporated within the ongoing iteration.
- **Secure**: Vulnerabilities in the software are detected via static and dynamic tests along with fuzzy testing and dependency scanning.
- **Plan**: The first development stage summarizes the work that needs to be carried out, defines the priorities, and monitors the software completion.
- **Code**: The code is written and discussed collaboratively with other teams and co-workers.
- **Build**: In this stage of development, the code is tested and its functionality is ensured.
- **Test**: The developed software is deployed to a staging environment. The practice of providing a new environment at the development stage is called Infrastructure as Code (IaC) and constitutes a core element of the DevOps pipeline.
- **Release**: The software is released to end users. The code has passed a number of tests, and breaking issues are improbable.
- **Deploy**: The same IaC that built the test environment is configured to build the production environment. In this stage, infrastructure and software platforms are managed.
- **Operate**: The well-tempered software operation is ensured, while customers are able to provide feedback about the service.
- **Monitor**: This consists of the final phase of the DevOps life cycle. The feedback in the “Operate” stage is taken into consideration, along with bugs, errors etc. The DevOps pipeline is also monitored by introspecting bottlenecks that impact productivity. It seems that this is the stage where the loop starts again. However, the whole DevOps process is continuous, with no definitive beginning and ending.

4.2.1 Mission first

The first step in developing each software is to create a mission for the DevOps teams to follow in all the stages. The whole team constructs a common vision to identify the labour that needs to be carried out while ensuring work priorities. The Value Streams are identifiable once the mission has been stated [25]. Value Streams can be defined as the series of stages an organization uses to create value from an idea and deliver it to its customers. Operational Value Streams deliver the products to the corresponding customers and generate income. Development Value Streams create the products to be sold to end users.

4.2.2 Customer focus

DevOps is not only about timing the market by continuously providing new features. The adoption of DevOps methods allows organisations to generate business value and revenue, while concurrently optimising customer experience [26]. In traditional software development, customer experience receives significantly less attention in delivery chains, and their needs are not always fulfilled. As high as 40 percent of customers leave a page that fails to load within five seconds [26], while security and privacy are also of paramount importance. By adopting innovative approaches, DevOps methods enable organisations to create a ***customer-centric delivery chain***.

Several issues should come under consideration for creating a customer-centric environment. Initially, quality should be everyone’s responsibility. Furthermore, every team member should consider software improvements. Data Monitoring measures a platform’s trend and influence. By monitoring underperforming microservices, the

DevOps team can improve them with the inclusion of updates. Identifying performance issues can address suboptimal performance quickly before a negative end user impact. Understanding customer experience leads to contextual monitoring according to different needs, personalities, activities, expectations and use cases of the end users of the software product [26][29]. Therefore, these steps assess customer satisfaction and improve user experience.

4.2.3 Left-shift

Left-shift, also referred to as shift left or left-shifting, is an approach to promote collaboration and communication by involving teams earlier in the DevOps life-cycle [27]. The method aims to reduce the amount of manual labour within the pipeline [28]. Left shifting is a practice in software development that ensures focus on quality, shorter test cycles (by testing earlier than expected) and problem prevention. By shifting left, the “test early, test often” goal is achieved [27].

During the software development routine, security testing occurs at the final steps of the development. By left-shifting, a complementary security method is implemented, as every new software is tested and monitored. Therefore, every change in the code undergoes a series of security tests before its deployment in the production system. Moreover, the compatibility of every new service is ensured by virtualisation testing. Finally, early testing in Left-shift discovers bugs and issues and needs smaller changes, therefore diminishing the risks and improving the overall software quality [27].

4.2.4 Continuous Everything

The definition of Continuous Everything is the systematic repetition of the same pattern [25]. Continuity resides at the core of DevOps operation. The following methods, namely Continuous Integration (CI), Continuous Delivery (CD), Continuous Deployment (CDe or CD) and Continuous Feedback (CF), compose a solution in order to accelerate production and feature releases while avoiding product quality reduction. This circle aims to bridge the gap between development and operation teams while applying high levels of automation.

4.2.5 Continuous Integration

CI is defined as a practice that enables developers to integrate code seamlessly and run tests quickly and automatically [25]. The main goal is to promote the collaboration of developers by using a single coding infrastructure. Merging is facilitated, while regularly merging code changes ensures fewer bugs and effort, improving productivity. CI is a method of detecting software bugs as soon as possible within the development lifecycle, while guaranteeing proper integration of all platform components. “Code” and “Build” phases correspond to this method.

4.2.6 Continuous Delivery

CD is a delivery method of software that functions properly, has been tested and can be used as a production platform [25]. CD works as an extension of CI that automates the deployment process. The main aims of CD are automated testing and build environment verification for release, testing these environments for stability, performance, and security, while also releasing a new feature into production when approved by the organisation. Manual integration is common in this method and aligns with the “Test” and “Release” phases.

4.2.7 Continuous Deployment

Continuous Deployment is an enhanced version of Continuous Delivery, thus can be found in the literature as both CDe and CD. The goals of CDe remain the same as CD, but automation is increased. Manual approval of new releases is omitted, as each build that successfully passes the pipeline checks, is automatically deployed in the production. CDe is an implementation where deployments are carried out automatically without human intervention and align with the “Test”, “Release” and “Deploy” phases.

4.2.8 Continuous Feedback

Although CF is usually outshined by the CI/CD cycle [20][24], it constitutes a major factor in DevOps operations. As mentioned before, DevOps aims to release versions, builds and fixes as soon as possible. Moreover, feedback obtained by end-users, customers and stakeholders needs to be incorporated within the new releases, in order to make better development decisions. By creating a functional feedback loop, the final product obtains better quality. Delivering smaller packets of working code and establishing a robust feedback loop enables the developers to refine the product. Concisely, CF corresponds to “Operate” and “Monitor”, ties to the DevOps lifecycle together and leads to the “Plan” phase once again.

4.3 Container Orchestration Infrastructure

IaC is both a process and a managing tool. Cloud infrastructure is essential to abstract the effort and requirements in configuring environments, Virtual Machines, networks, and storage components. In the traditional approach, Standard Operation Procedure builds the environment manually. In this manner, predicting human or operational error is difficult. On the other hand, using a configuration management tool allows the environment to be built in the cloud seamlessly [30].

Orchestration is a part of the configuration management tool set. Its function is to manage a number of servers and networks and automatically configure an application in an available server. Therefore, new software delivery is accomplished anywhere and anytime. Modern systems are built as stacks of multiple components, with each component being a self-contained system. The complexity skyrockets due to the entangled relationships among different systems. Although containerization gives a degree of freedom in deployment tasks, component coordination is still challenging. Tools that help in this direction include Kubernetes and Docker Swarm, among other orchestration tools [30]. An orchestration tool is very important for a containerized platform. Sections 4.3.2 and 4.3.3 present an overview of the most popular tools.

4.3.1 Docker

Docker is a platform functionality that helps in application development, shipment and run. By using Docker, applications are separated from infrastructure, enabling swift software delivery. Furthermore, the Docker platform allows packaging and running an application in a container, i.e., a relatively isolated environment. Isolation and security allow running multiple containers on a given host simultaneously. Docker tools include application development using containers, seamless distribution and testing within the container, as well as application deployment into the production environment [31].

Containers work great within the CI/CD workflow. Docker’s platform enables highly portable workloads, and its portability and lightweight nature allow deployment and scalability according to the organization’s needs. By running multiple workloads on the same machine, Docker interweaves with high-density environments and deployments where fewer resources must be used [31].

A Docker image is a method to create a Docker container by acting as a template with a set of instructions. Images can be considered as a starting point when using Docker. In order to create applications, a Docker image includes code, libraries, and other files for an application to run. The main goal is to increase reusability while simultaneously decreasing disk usage. A Docker container is a runnable image instance. A container can be operated either with the Docker API or by the command line. It helps in creating, running and deploying applications in a network environment that are relatively isolated from the hardware and other containers [31].

4.3.2 Kubernetes

Kubernetes [32], commonly encountered as K8s, is a portable and extensible open-source platform used for container management, developed by Google. Kubernetes enables both configuration and automation. The main benefits of Kubernetes are as follows. In case of high network traffic to a container, load balancing helps stabilise the deployment. Storage orchestration includes automatically mounting a desired storage system, either locally or in the cloud. Kubernetes can be automated to create and remove containers and adopt resources to new containers. In order to make the best use of resources, the volume of CPU and RAM can be specified. Moreover, Kubernetes restarts, replaces, and kills containers to perform health checks while allowing passwords, OAuth tokens and SSH keys for secure storage.

Every Kubernetes deployment creates a cluster. Each cluster consists of nodes (worker machines) that run containerised applications. The worker nodes host the Pods (i.e., a set of running containers), which are the components of the workload. A container orchestration layer is created, which is called control plane, managing the containers' lifecycle. The control plane provides high availability and fault-tolerance by running on multiple computers and clusters running multiple nodes.

Kubernetes is able to connect applications with services, and pods can communicate with one another, regardless of the host. Each Pod obtains its own IP address, so Pods reach each other on localhost. Therefore, a Kubernetes service is created by a logical set of Pods that provide the same functionality.

4.3.3 Docker Swarm

Docker Swarm [33] constitutes a viable Kubernetes alternative. A Swarm consists of multiple Docker hosts. Each time a service is created, its optimal state is defined, and Docker maintains the desired state. Swarm deployments include Docker nodes distributed across multiple machines. A manager node is created that determines the tasks of the worker nodes. Therefore, manager nodes perform orchestration and cluster management. The tasks run the commands inside a container, constituting the nuclear unit of a Swarm. Thus, services are created as the central Swarm structure and dictate the interaction of the user with the Swarm.

Docker Swarm offers several features described as follows. Initially, a Swarm can be created without the need for any additional orchestration software. Docker command line can deploy application services. Swarm provides decentralized design and a declarative service model to define the desired state of every service, as well as desired state reconciliation. A swarm manager adds or removes tasks to maintain the desired state by scaling up or down. The swarm manager automatically assigns network addresses for multi-host networking along with a unique DNS to each node while being able to perform load-balancing distribution. Swarm mode is secure by default, with TLS mutual authentication, and service application updates are performed incrementally [34].

In comparison to Kubernetes, Docker Swarm provides a simple installation and setup, automated load balancing and fast container deployment. Moreover, a container's migration to different platforms is possible, and data volumes can be shared with any container. However, it is limited to Docker API, and its automation capabilities are not as robust as are of its competitor. Kubernetes includes built-in monitoring and supports multiple security protocols. Every platform has its advantages and disadvantages, and the choice depends on the needs of each organization.

4.4 DevOps impact on business culture

DevOps culture embodies a number of business values and benefits. Its precedence is to improve the production environment for faster software delivery and continuous improvement. The synchronous ability of anticipation and response to industry disruptors is enabled within a software development process that empowers teams to be autonomous, work smarter and deliver faster, reducing the required labour. In this manner, DevOps teams are able to respond to market demands [17].

The main advantage of the DevOps culture is the removal of institutionalized silos that usually lead to constraints. This is especially important when all teams work to succeed in the same key performance indicators. Moreover, a unified tool chain allows multiple teams to collaborate, accelerating delivery and feedback [17].

DevOps microservices are used to manage complex architecture [35]. Microservices aim to break down systems and architectures to a molecular level. Fragmenting complex applications and on-demand service delivery aim to improve software performance. DevOps supplies a developing and deploying framework for managing the microservices container ecosystem. The major DevOps asset is fast delivery cycles. Since it is a cloud-based approach, microservices are innately distributed and operated across networks.

In [36], Erich et al. interviewed six organizations that shifted to DevOps practices. The main targets of this adoption are as follows. Lead-time reduction, problem solving, feedback improvement, improved automation, release time reduction, and velocity increase are some of the main benefits these organizations reaped. Tech giants such as Google, Amazon, Netflix, LinkedIn, Meta/Facebook, and Spotify have shifted towards DevOps implementations for almost a decade [37], constituting them able to dominate the market.

Figure 25 summarizes the business outcomes of DevOps. Risk and cost reduction, quality improvement and customer satisfaction are of paramount importance. DevOps process unveils the business outcomes achieved through continuous delivery and shorter life cycles that aim to improve product quality and customer satisfaction.

In conclusion, DevOps cultural shift provides a number of benefits to organizations. It is much more than just software delivery, as aligns with business goals and objectives. DevOps improves collaboration and communication, while it increases efficiency and time to market. The main business focus is process and infrastructure automation, efficient delivery, along with freedom for experimentation and innovation, creating a sense of shared responsibility and prioritizing user experience. DevOps, therefore, is not just a specific engineering team but about people and cultural shifts.

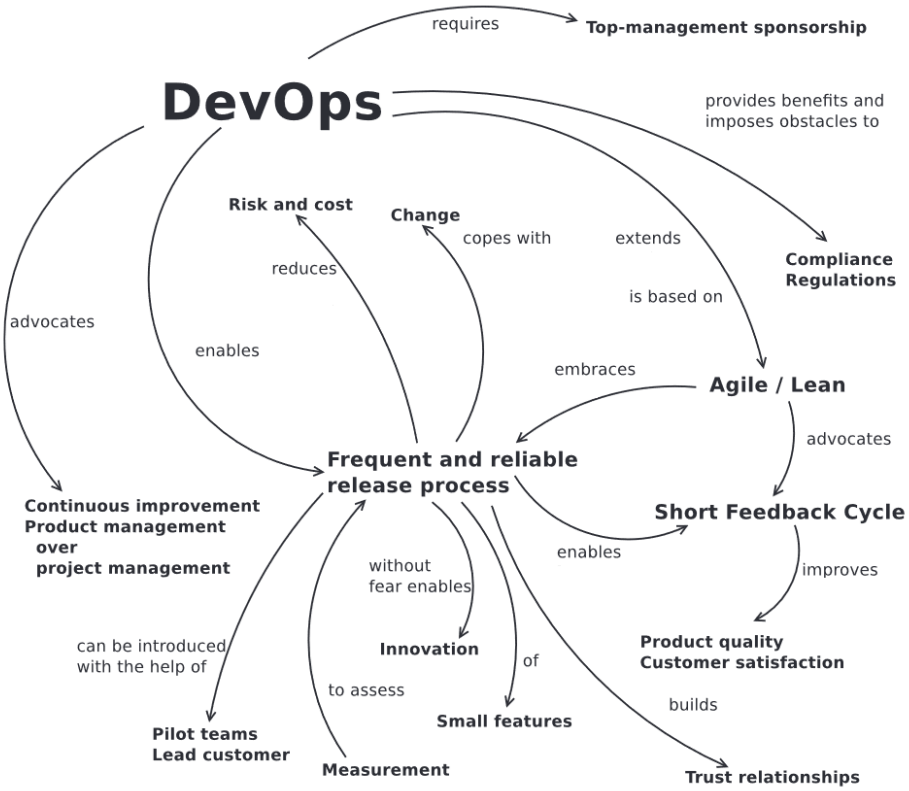

Figure 25: DevOps process map [25].

4.5 DevOps & Infrastructure in SHOW Marketplace

DevOps practices are also implemented within the SHOW Marketplace ecosystem. Docker containers contain the Marketplace microservices, since web services (gunicorn and Django), MongoDB and nginx all have their dedicated container. Moreover, services that have been deployed in a later stage, i.e., payment methods and rating system, are also to be constructed in separate containers. GitLab [18] offers the capability for Continuous Integration and Continuous Delivery, therefore facilitating DevOps practices even in the earlier development stages. Other DevOps practices are applied as follows. Mission First and Customer Focus play a pivotal role within the whole Marketplace ecosystem development, by defining work priorities and reducing the necessary manual labor, while also creating customer-centric delivery chain. Left-shifting is accomplished with early testing, while also implementing security measures within the Marketplace code in early development stages. Security measures are also materialized on the server that hosts the Marketplace. Following the major impact of DevOps practices in business culture, microservices deployment and management are embedded within the SHOW Marketplace lifecycle. Finally, considering the Orchestration infrastructure, based on the analysis of subsection 4.2 and the needs of the SHOW Marketplace, one of either Kubernetes or Docker Swarm will be selected as the desired orchestration tool for the next Marketplace version delivery.

5 Product items

In this section the product items that have been uploaded until now to the first public version of the SHOW Marketplace (accessed at: https://show-marketplace.eu/) are presented. These products are either complete solutions (some implemented within WP6 of the SHOW project and others deriving from other external EU funded projects (i.e., Avenue project)), or architectures, which are clustered under the "SHOW specific material" category in the SHOW marketplace data model. It is worth mentioning that in comparison with the previous deliverables [1], although there are descriptions of new product items that are present in the SHOW Marketplace as well, there are some missing product items, since there is not any update for them and cannot be uploaded to the SHOW marketplace at the time being. Until the next and final version of the deliverable and progressively until then (as the content enrichment process is continuous) all the product items will be present and will have been integrated to the SHOW Marketplace.

5.1 Solutions

This subsection describes the complete solutions that have been integrated into the first public version of the SHOW Marketplace. Since this is the second version of deliverable related to SHOW Marketplace and services, the solutions having been already described in the first version [1] are described very briefly in this version, including the corresponding reference for further details, and only the new information is described more extensively. If there are not any changes in the description of a product compared to the previous version (this is valid in case of the Avenue services), only the integration into the SHOW marketplace subsection is included.

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

5.1.1.1 Description

Logistics Service for Freight Replenishment and Final Delivery in the Supply Chain has been described in the previous version of the deliverable. For the deployment and illustration of a Logistics as a Service (LaaS) platform, an automated logistics mobile app has been developed by CTLup. Hence, the new app titled Automated Logistics Mobile App (ALMA) aims to enable service users to book a delivery transfer from an origin to destination and track the delivery and vehicle positions. Moreover, this app correspondingly enhances the accumulation of user points to utilize the mobile app functionalities, or it disciplines the users who do not respect the appointment for loading or unloading the delivery. In the subsequent subsections, the parts of the Automated Logistics Mobile App (ALMA) are described and specified.

Figure 26: ALMA product for automated logistics services.

5.1.1.2 Technical Requirements & Operations

New information about operations and technical requirements of the scenario of a last-mile distribution service is presented in this subsection. Based on the objectives having been described in D6.1 [1], the following functionalities will be provided by an automated logistics web-app/mobile-app to the users in terms of the following areas:

- **Registration:** The act or process of entering information about potential users-customers to sign up to an automated logistics mobile-app or website by entering:
    - name/surname.
    - phone number (to reach the users if there in case of unexpected difficulties related to delivery).
    - address (to send the delivery back if needed).
    - e-mail address (for validation purposes).
    - login password (authentication).

  Automated Logistics website/mobile-app may offer free information, news, and other services to unregistered users, but requires registration for the validation and authentication of user reliability to avoid any fake-order. The registration will make logging faster, more trustful ordering, and enables the users to earn credits/points to utilize the logistics service.
- **Delivery Tracking:** Tracking the delivery aims to track the product for a user to follow and monitor the progress of deliveries. This procedure of “package tracking” assists to localize shipped products, during sorting, transfer, and delivery, but also helps to verify provenance and predict delivery. Delivery Tracking is able to provide users with information about the route of delivery and the anticipated delivery date according to the vehicle route and the chosen timeslot. This is also very crucial in order to offer a reliable service that will respect timelines and is resilient against extreme weather situations that could cause the delivery to get lost.

  When the delivery is loaded to the shuttle, the automated logistics app will allow the customers to take a photo and send it to the platform. After the arrival to the final destination, the final users could also take a photo in order to support the certification of the correct delivery.

- **Vehicle Tracking:** The vehicle tracking aims to monitor the vehicle's position by combining the automated location of individual vehicles with software that collects the fleet data. Based on this purpose, the modern approach for tracking a vehicle is generally based on the vehicle location data from a system such as GPS, Galileo, etc. This vehicle location can be also visible with a separated sub-function of the app/website via the Internet.
- **Booking:** This function is related to the booking service in order to pick up or send materials by selecting a timeslot and the place where the delivery will be transferred.

5.1.1.3 Development & Deployment

As mentioned in the SHOW Deliverable D6.1 [1], the LaaS cloud platform is realized by an automated logistics mobile app as a web-based application with reserved access and a secure channel (SSL). It is planned to operate for specific users (3PL) to handle the last-mile distribution service for its e-commerce customers.

The aim of the automated logistics mobile app (ALMA) is to serve potential customers while having some material-product transfers from an origin to a destination during logistics pilot sites. The pilot scenarios will be supported in terms of booking the delivery slot to load materials, booking the pick-up time for the final customers, tracking the deliveries to be sure they are loaded and received without any harm, and continuously tracking the vehicle positions by GPS, EU satellite system, or any location providers.

Automated Logistics Mobile App functionalities are described below:

- The first functionality is related to the registration of the users-customers to the platform. User Profile section on the automated logistics mobile-app provides user characteristics information regarding basic identification information and user points that will be utilized to use functionalities.
- After that, when users have been logged-in to the platform will be able to review user profile details and earned-utilized "user points".
- Furthermore, the automated logistics mobile-app will provide the ability to book delivery service, namely, to choose picking address and pick-up time and also to insert shipping address and the e-mail address of receivers.

Apart from the aforementioned functionalities, there will be two more abilities in the second version of the automated logistics mobile app. These functionalities are the delivery and the vehicle tracking. Delivery Tracking aims to visualize the position and location of the delivery with status updates, shipping/picking addresses, estimated time of arrival etc. Vehicle Tracking aims to visualize the exact locations of the vehicles by using GPS data or EU-Satellite data. It aims to display information about the closest vehicle to the users.

5.1.1.4 Integration into the SHOW Marketplace

In Figure 27, it is possible to see how ALMA product has been represented on the SHOW Marketplace platform.

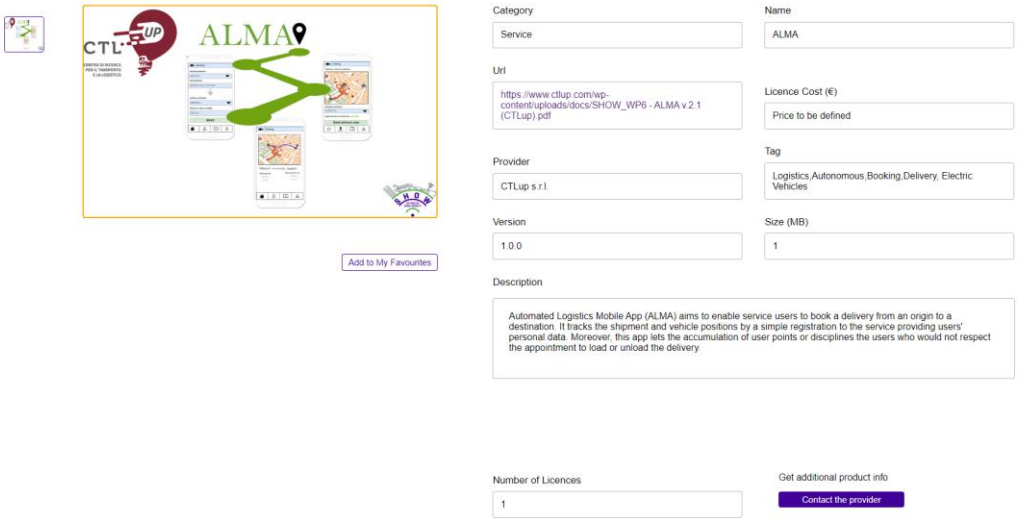

Figure 27: ALMA product on SHOW Marketplace platform.

5.1.2 Analysis of vehicle/powertrain performance

5.1.2.1 Description

This service provides a substantial decision basis for the fleet owner before a vehicle is actually bought or dispatched. Depending on the most relevant indicator of interest (will be explained below), a simulation of different vehicle types is performed using simultaneously real data of the route as additional input data.

The following vehicle types can be analyzed:

- plugin hybrid vehicle (PHEV).
- battery electrical vehicle (BEV).
- internal combustion engine vehicle (ICEV).

For the assessment, the performance will be analyzed with respect to:

- energy consumption.
- well-to-wheel CO_2 emission.
- total energy costs.

Moreover, as the model is adaptable, an optimization with regard to a dedicated performance indicator can be performed by varying vehicle dimensions (e.g., battery pack energy, motor power, etc.)

The result of such an analysis will be showcased by a case study, which will be available in future on the marketplace as well. For that study, data from the Linköping site are used, which are accessible via the SHOW CKAN portal. In near future, also data from other sites, e.g. Graz, will be considered for the case study once they are available.

5.1.2.2 Technical Requirements & Operations

An adaptive virtual plug-in hybrid vehicle model is used to simulate the vehicle performance. The full powertrain (transmissions, e-machines, engine, battery) has been modelled to estimate the energy consumption with both fuel and electricity. The model has been validated in previous studies, and the results have shown that by changing the parameters, the model can be transformed into an electrical and combustion engine model and thus, a comparison between different vehicle types is possible. As a result, the following performance indicators are computed:

- The total energy consumption.
- The total energy cost.
- The total CO_2 emissions.

The velocity and acceleration of the vehicle have a major impact to the consumption of energy and accordingly to CO_2 emission and energy costs. Hence, the simulation requires real vehicle or traffic data of the planned route. The assumption here is that the selected vehicle for performing the simulations is similar enough to the actual vehicles used to gather the velocity profiles.

5.1.2.3 Development & Deployment

The service has been validated by performing a sensitivity analysis of the key indicators across the driving dataset provided by the project partners. The hardware performance data for the three types of vehicles have been retrieved from open-source projects and the models to scale the performance according to their size came from the literature review. Estimating vehicle indicators requires some parameter assumptions. The software lists the parameters that need to be defined before the running of the simulation.

5.1.2.4 Integration into the SHOW Marketplace

The integration of the Analysis of vehicle/powertrain performance service into the SHOW marketplace is depicted in Figure 28.

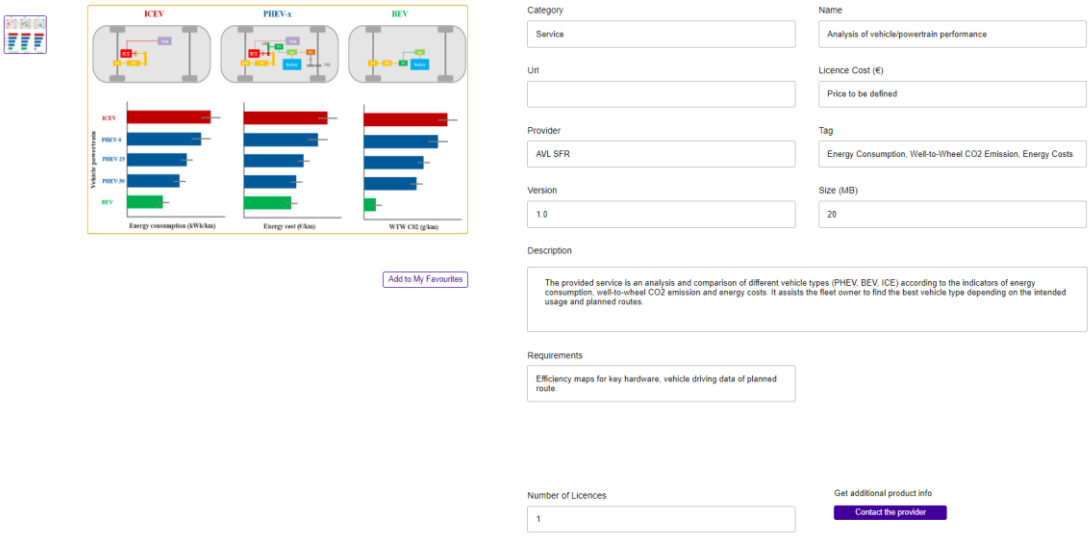

Figure 28: Analysis of vehicle/powertrain performance product on SHOW Marketplace.

5.1.3 Advanced PHEV vehicle control software

5.1.3.1 Description

By using real data from vehicles in use, an in-depth analysis applying advanced strategies can provide an insight whether the vehicle control strategy can be further improved. This may be of special relevance if the routes do not change over time. If the simulation leads to a significant benefit, the process of retraining the control software can be initiated. Hence, in cooperation with the OEM, an update is performed. Fleet owners benefit from a reduction of energy consumption and consequently from a reduction of operation expenses.

As previously with the *Analysis of vehicle/powertrain performance service* the result of analysis and optimization will be presented in the future as a corresponding case study using data for that study from the Linköping and the Graz pilot sites.

5.1.3.2 Technical Requirements & Operations

The aforementioned learning framework has been applied to a PHEV-x. The vehicle powertrain has been equipped with a conventional combustion engine, two different electric machines, one per axle and a battery large enough to drive “x” km. The measured vehicle performance metrics used in practice are the total energy consumption, the energy cost (used also to define the optimal policy) and the well-to-wheel CO_2 emissions.

The model is essentially a multi-class classification problem, having as features the system states and the previous actions. The control model employs two sets of features to predict the correct action: 1) past actions and actual states and 2) the first set along with short/long-term future inputs.

5.1.3.3 Development & Deployment

As mentioned also in D6.1 [1], the performance of the ML-based control model strategy has been assessed against four different benchmarks. Moreover, three ML cases for control strategy have been determined and compared in order to estimate the benefit of functionalities such as the forecasting of the velocity forecast.

5.1.3.4 Integration into the SHOW Marketplace

The aforementioned service is available into the SHOW Marketplace as can be observed in Figure 29.

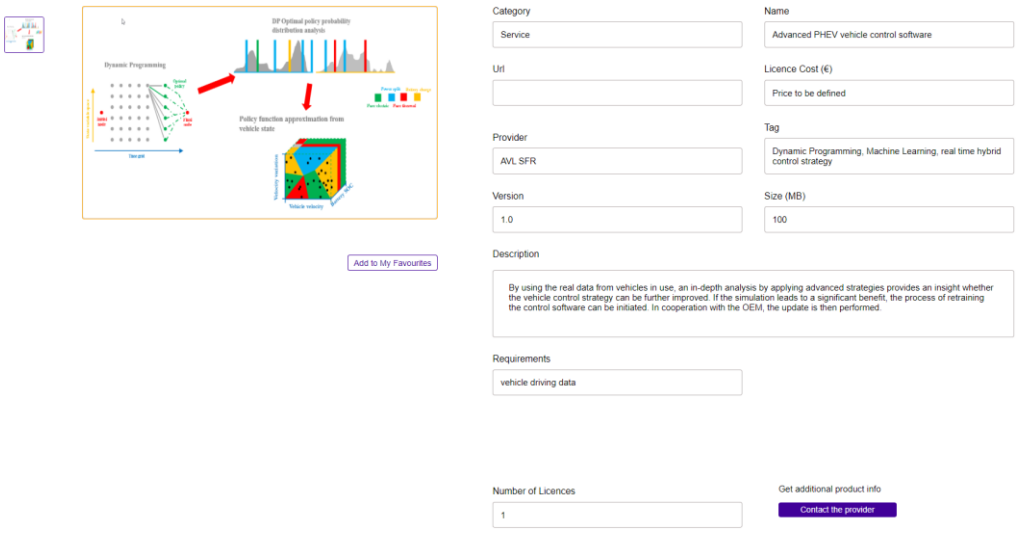

Figure 29: Advanced PHEV vehicle control software on SHOW Marketplace.

5.1.4 Dawn – Autonomous driving software platform

5.1.4.1 Description

Sensible 4’s Dawn is a software platform enabling SAE Level 4 autonomy for vehicles. It is aimed for last-mile passenger and goods transportation and industrial use cases. Dawn’s key features include the ability to drive autonomously in all weather conditions, on the roads without the lane markings and without a good GNSS coverage. The solution provides safe and smooth autonomous operations in all weather conditions and environments. The core of autonomous driving is the full-stack autonomous driving

software with four modules – Positioning, Control, Obstacle Detection and Remote Control, each with a specific task. The software components enable the vehicle to locate itself, observe its surroundings, and maneuver in traffic. In addition, when the robot vehicle needs human assistance, remote operations enable smooth help, with fleet operations and APIs to third party services.

5.1.4.2 Technical Requirements & Operations

While Dawn is platform agnostic and can be deployed to most vehicles, there are certain requirements for the vehicle. Dawn requires a compatible sensor stack, especially LiDAR sensors providing a 360 view around the vehicle. Other requirements include Drive-by-Wire functionality, compatible computing hardware and connectivity features for vehicle platform. All required hardware can be included in the service as additional purchases. Once installed to a vehicle, the use will require a license. Sensible 4 can also provide a more extensive SLA.

5.1.4.3 Development & Deployment

Dawn is under constant development and the client will always get the latest stable release. The baseline product can be integrated to vehicles as a service, and adjustments to the specific Operational Design Domain can be accommodated within operating margins. Deployment support is available from the service provider.

5.1.4.4 Integration into the SHOW Marketplace

Dawn product has been uploaded to the SHOW marketplace as can be displayed in the Figure 30.

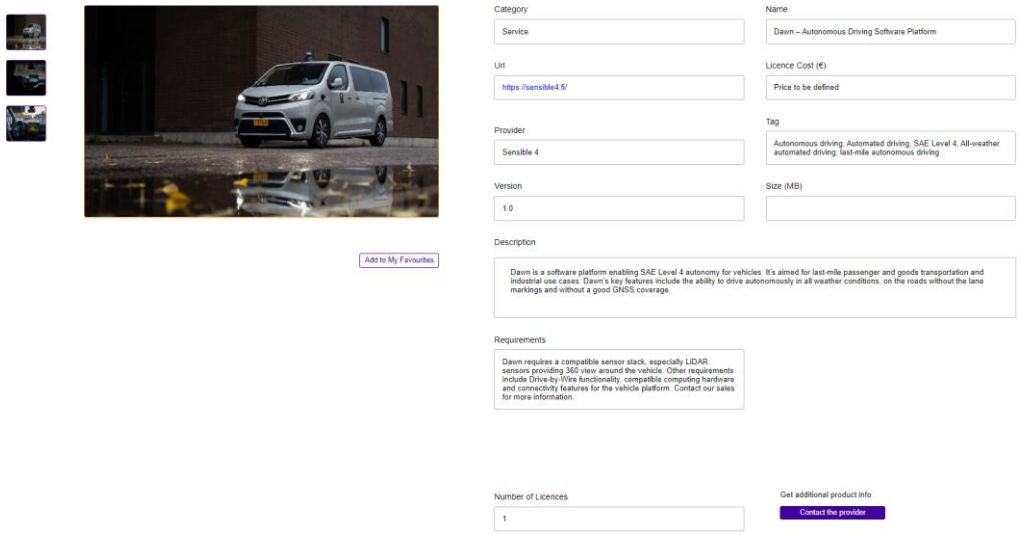

Figure 30: Dawn product on the SHOW Marketplace.

5.1.5 Enhance the Sense of Security and Trust

5.1.5.1 Integration into the SHOW Marketplace

The “Enhance the Sense of Security and Trust” solution derives from Avenue project, has been described in [1] and the integration into the SHOW Marketplace is shown in Figure 31.

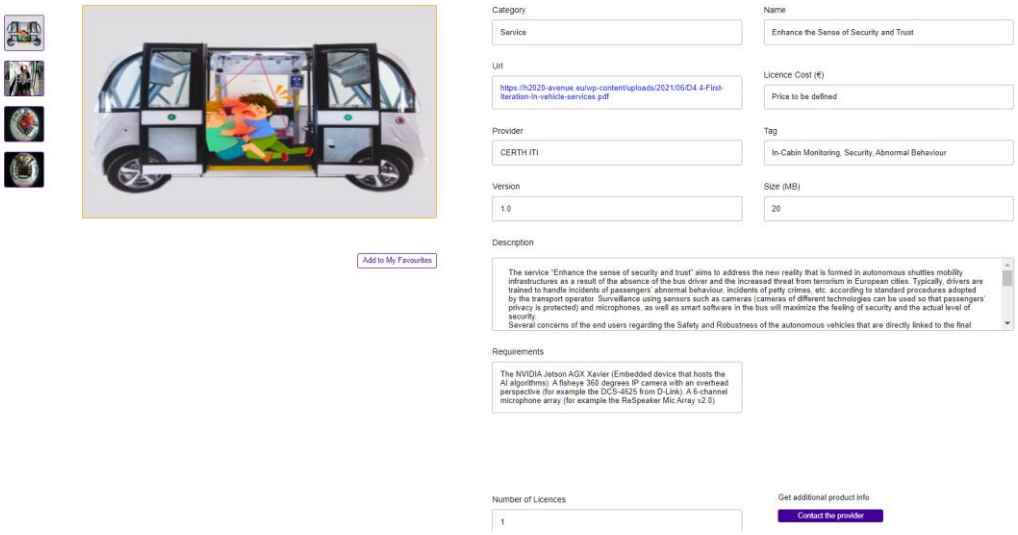

Figure 31: The "Enhance the Sense of Security and Trust" solution on the SHOW Marketplace.

5.1.6 Automated Passenger Presence

5.1.6.1 Integration into the SHOW Marketplace

"Automated Passenger Presence" is another solution coming from Avenue project and has also been described in [1]. The service is presented in the SHOW Marketplace as depicted in Figure 32.

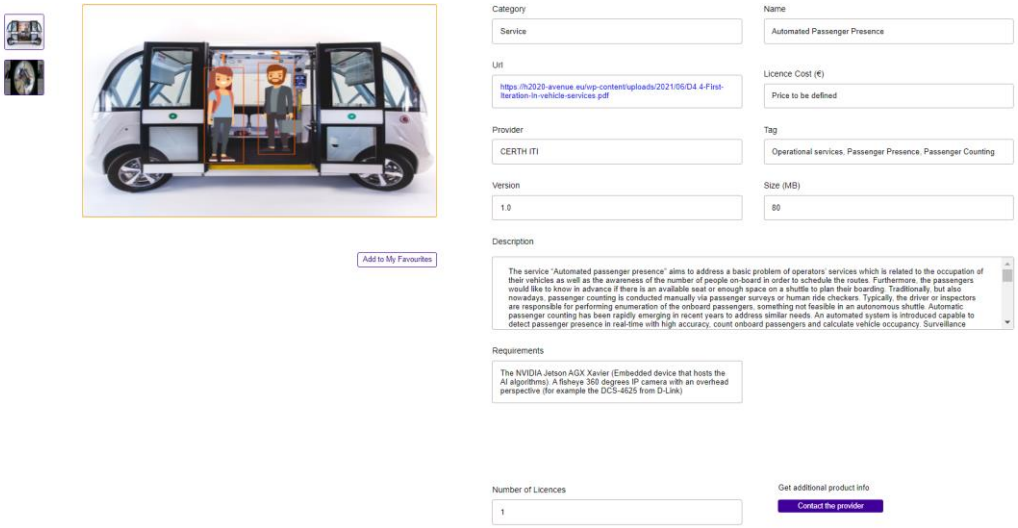

Figure 32: The "Automated Passenger Presence" solution on the SHOW Marketplace.

5.1.7 Follow my Kid/ Grandparent

5.1.7.1 Integration into the SHOW Marketplace

From the Avenue project, the "Follow my Kid/ Grandparent" service has also been uploaded to the SHOW Marketplace. An instance of this integration is displayed in Figure 33.

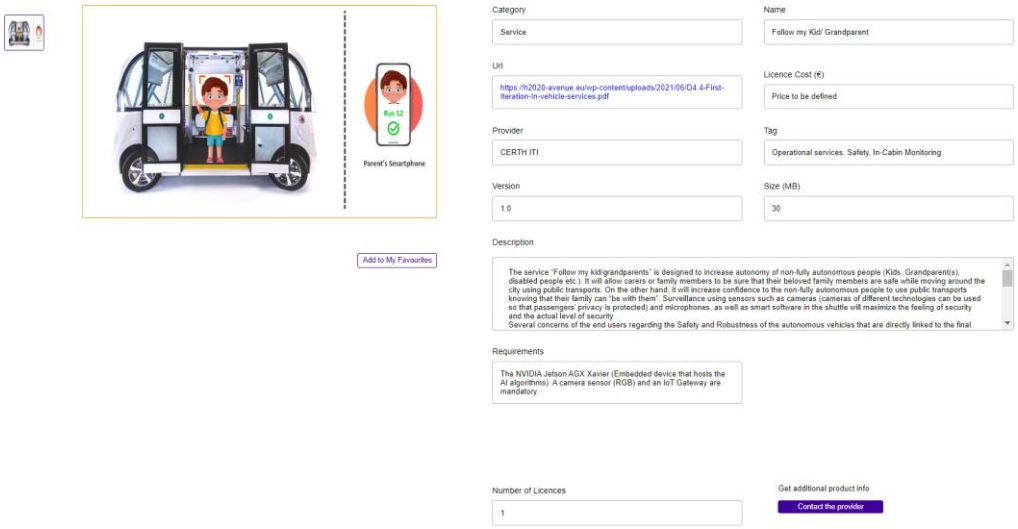

Figure 33: The "Follow my Kid/ Grandparent" solution on the SHOW Marketplace.

5.1.8 Shuttle Environment Assessment

5.1.8.1 Integration into the SHOW Marketplace

The fourth service coming from the Avenue project is titled "Shuttle Environment Assessment" and has been uploaded recently to the first public version of the SHOW Marketplace. The aforementioned integration can be observed in Figure 34.

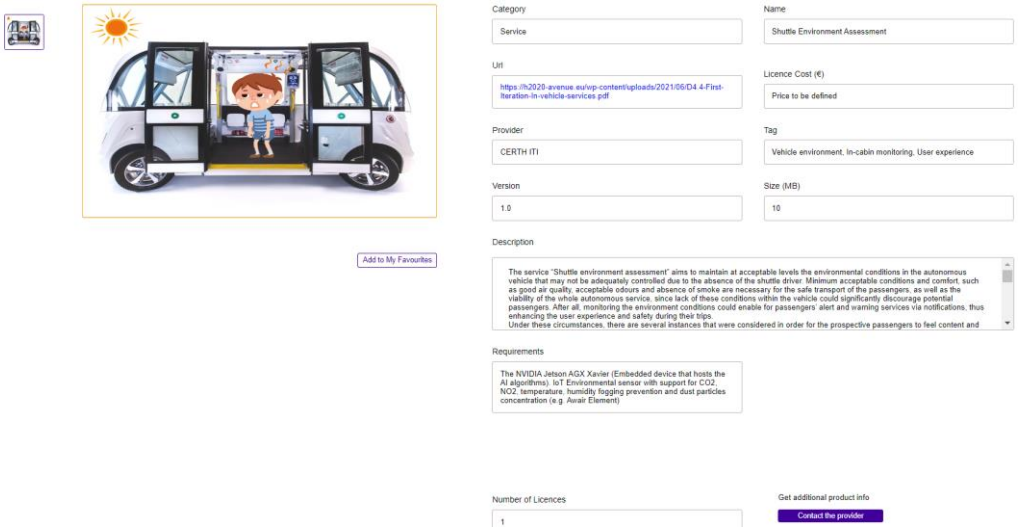

Figure 34: The "Shuttle Environment Assessment" product on the SHOW Marketplace.

5.1.9 Smart Feedback System

5.1.9.1 Integration into the SHOW Marketplace

The last service from Avenue that has been integrated into the SHOW Marketplace is the "Smart Feedback System". The depiction of this service into the SHOW Marketplace is illustrated in Figure 35.

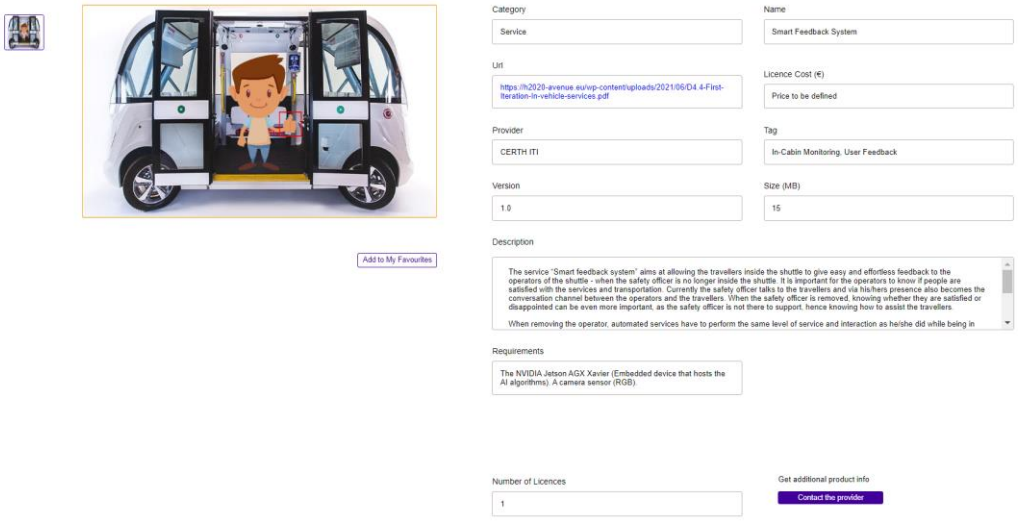

Figure 35: The "Smart Feedback System" on the SHOW Marketplace.

5.2 Architectures

In this subsection, the architectures originating from the SHOW project and have been integrated into the SHOW Marketplace are presented. Specifically, the first product item is the architecture of the SHOW project, which was the outcome of the activities of WP4, whereas the other 3 product items are derived from a Hackathon within SHOW context that took place on 21-23 March 2022 in Thessaloniki, Greece. The challenges for which the three finalist teams proposed solutions were [38]:

- Human assistance stand-by in case of problems (emergencies, security).
- Adapting and optimizing capacity to handle demand in a flexible way.
- Accessibility and assistance for persons with reduced mobility.

5.2.1 Open modular CCAM-for-PT system architecture

5.2.1.1 Description

The proposed architecture derived from SHOW project, particularly from WP4, constitutes a multi-tier architecture for PT containing different variations, according to the needs of the stakeholders. Specifically, in the IoV paradigm, there are multiple stakeholders including among other OEMs and Service Providers and all these actors need access to data, interfaces, and services. Moreover, CCAV cooperate by exchanging information among them, but also receive data by other road users and cloud systems and therefore require safety and security. Hence, the whole architecture of such systems is challenging. The SHOW reference architecture that has been integrated to the SHOW Marketplace as well, is based on web-of-things layered model and expects to provide support for the CCAM services of the future as these are envisioned within SHOW. The core output of this work was the SHOW reference architecture, inspired by the web-of-things layered model, supporting modularity and expects to be a useful tool for existing fleet management, PT systems and CCAM services of the future. Thus, three architecture variations were proposed with respect to the interoperability, whereas specific cyber security mechanisms to ensure communication have been also included. Finally, in this approach the well-established C4 model was applied, in order to support different architectures with different levels of detail.

5.2.1.2 Integration into the SHOW Marketplace

The Open modular CCAM-for-PT system architecture has been uploaded to the SHOW Marketplace and is shown in Figure 36.

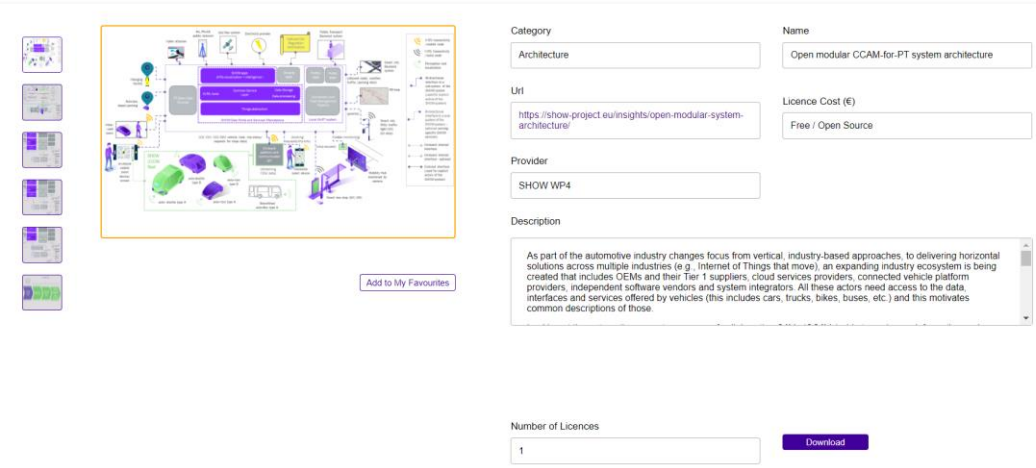

Figure 36: Open modular CCAM-for-PT system architecture.

5.2.2 Gustav

5.2.2.1 Description

The first team in the SHOW Hackathon tackled the issue of securing and creating a safe environment for commuters inside AVs, creating the “Gustav” (Global Utilities for Security and Trust in Automated Vehicles) architecture for a service. Gustav (Global Utilities for Security and Trust in Automated Vehicles) expects to be a complete solution proposes an AI-based lightweight proposal for detecting abnormal activities in AVs. Interest in this service can be manifested by Safety Operators, as automation in security and trust can increase the effectiveness while minimizing manual labour, as well as AV stakeholders for enhanced services and reliability in AVs. The main beneficiaries are AV commuters by creating a secure environment and, therefore, the overall community by reducing criminal activities and ensuring safe local spaces. AVs can be considered the future of public transport. However, the absence of a driver could lead to uncomfortable situations for the commuters (e.g., if a fight or another abnormal activity breaks out). The main goal of this service is to instil a sense of commuting certitude in passengers and robustness in AV itineraries.

5.2.2.2 Integration into the SHOW Marketplace

The integration of Gustav’s architecture into the SHOW Marketplace is reflected in Figure 37.

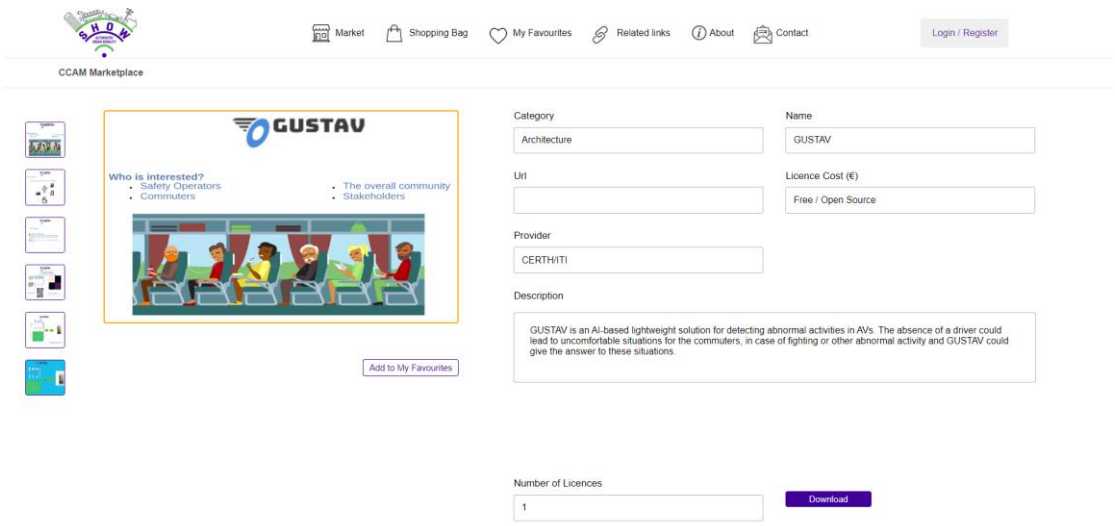

Figure 37: The Gustav architecture on the SHOW Marketplace.

5.2.3 DeForus

5.2.3.1 Description

The second team in the SHOW Hackathon, in order to answer to the second challenge, proposed DeForus (Demand Forecasting and Optimal routing in Unscheduled Stops). DeForus (Demand Forecasting and Optimal routing in Unscheduled Stops) is a proposed novel design dedicated to forecast demand and considering dynamic routing based on both historical and real-time data. The contemporary solutions in this field are based on fixed schedules without benefitting from real-time data. DeForus proposes dynamic fleet management and route selection and prioritisation based on real-time data. DeForus aims to be an excellent solution for road and fleet operators, stakeholders, and municipalities. The historical data used in this scope include traffic data by road operators, weather data from external sources, historic demand using passenger counting, as well as scheduled event data utilising text mining by sources such as social media and websites. The real-time data include on-board passenger counting collected by vehicle equipment, on-station passenger counting collected by bus station equipment, along with information coming from on-board units and roadside units (e.g., traffic jam, roadworks, ice on the road etc.) retrieved by vehicle and infrastructure equipment. Both historical and real-time data are used as inputs for fine-tuning and forecasting, having customer service and energy efficiency improvement as optimisation factors, in order to conduct fleet management.

5.2.3.2 Integration into the SHOW Marketplace

Figure 38 shows the representation of the DeForus architecture into the SHOW marketplace.

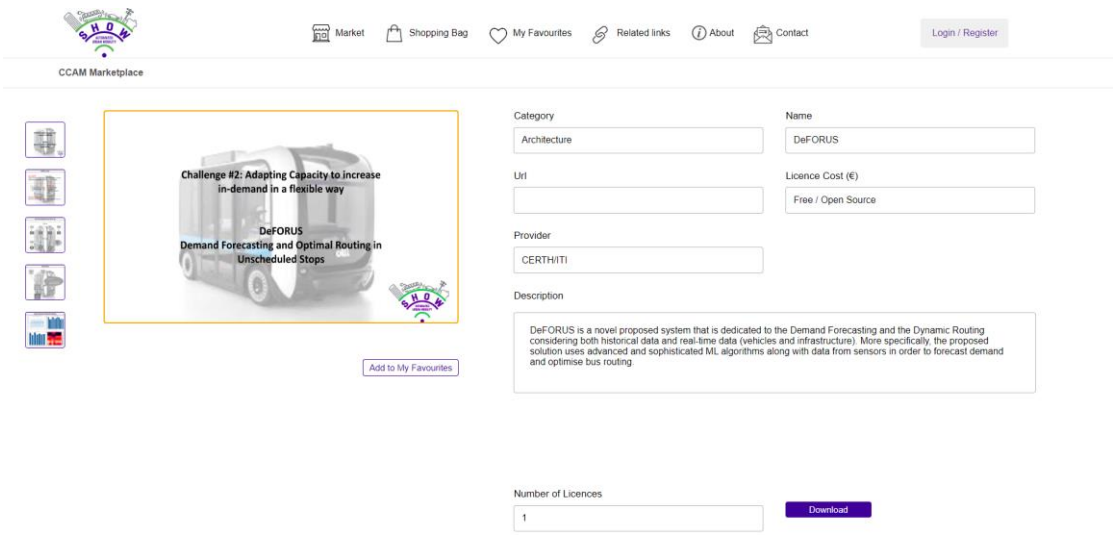

Figure 38: The DeForus architecture on the SHOW Marketplace.

5.2.4 AssistIO

5.2.4.1 Description

The final proposal, AssistIO, was introduced by the third team of the SHOW Hackathon as an architecture to address the issue of assisting people with reduced mobility [39]. The priority of the AssistIO architecture is to provide services to visually impaired people. Specifically, this application helps people to use AVs offering simplicity in both design and implementation. Hence, a mobile application is proposed, easy to use and understandable for the visually and cognitive impaired user, that communicates to a server with a secure socket connection and AI training. AssistIO application includes pre-trip navigation, in/out cabin navigation, stop and emergency button, utilizing accessible design with text-to-speech and module for notifications. Moreover, the server functionalities include authentication and location tracking for both the commuter and the shuttle, along with AI training for image recognition.

5.2.4.2 Integration into the SHOW Marketplace

As with all the aforementioned product items, AssistIO was integrated into the SHOW Marketplace and the corresponding depiction of this architecture follows in Figure 39.

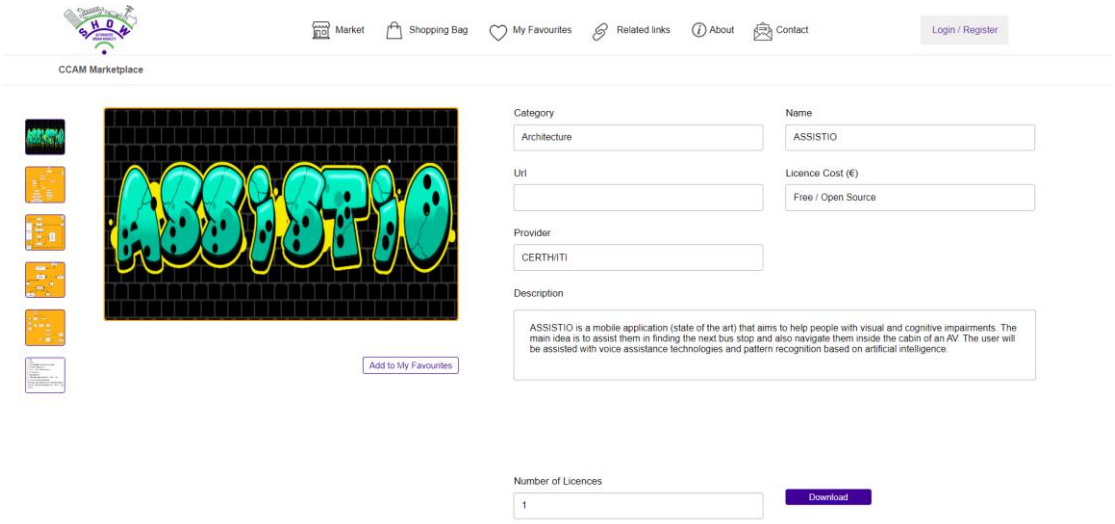

Figure 39: The AssistIO architecture on the SHOW Marketplace.

6 Conclusions and outlook

The present deliverable summarizes the work performed in WP6 during the previous months (M22-M30). Specifically, the outcome of the five activities (A6.1-A6.5) composing WP6 have been presented. The major points raised by the present deliverable are: a) the development of the business strategy governing the SHOW Marketplace, in order to exploit it in the most efficient way, b) the partial fulfilment of the requirements having been set in the previous deliverable and are to be fully addressed in the coming period, c) the strategy followed for the infrastructure and the development operations, and finally, d) the integration of product items (either complete solutions or architectures) to the first public version of the SHOW Marketplace.

All the aforementioned points that constitute the outcome of the work performed are very prominent, since they are trying to ensure that the SHOW Marketplace is capable to meet its main objective, which is to act as a one-stop place for the whole CCAM community offering corresponding services and stimulating the CCAM adoption, successfully. Moreover, regarding the business model presented in the deliverable is of utmost importance since it determines the sustainability of the SHOW Marketplace and its respective efficient exploitation. This will be further analysed in the next deliverable D6.3, where a detailed business plan for the Marketplace will be developed, and additionally a more detailed financial and benefits analysis will be performed (through Cost-Benefit Analysis, Willingness to pay analysis, and other similar financial methods).

Regarding future work related to the SHOW Marketplace and services, there is one more related deliverable, which constitutes the final version, D6.3: “SHOW Marketplace and services - final version”, scheduled for M40. This will summarize all the work encompassed by WP6 and the corresponding related activities. Hence, the following deliverable will describe the final version of the software (A6.1) and the respective infrastructure (A6.2). Moreover, it will contain the whole set of product items (in addition to the ones described already in this and the former Deliverable version and existent in the online one-stop-shop), including among others operational services (A6.3), energy management services (A6.4), along with the personalization of the SHOW Marketplace and specific services (A6.5).

Finally, the valuable comments proposed from the Advisory Board should not be neglected, since these will initiate the inclusion of new features regarding the SHOW Marketplace (filtering, searching, enhancement of description page, UI/UX improvements, demo video). Hence, by the time of D6.3, the SHOW Marketplace development team aspires to provide new functionalities in order to offer a more user-friendly experience with the SHOW marketplace and also, an extensive report for all contained products, including the benefits in using each of them.

References

[1]. SHOW (2022). D6.1 SHOW Marketplace and services. Deliverable of the Horizon-2020 SHOW project, Grant Agreement No. 875530.

[2]. Casadesus-Masanell, R. and J. E. Ricart. 2011. How to design a winning business model. Harvard Business Review (January/February): 100-107. Retrieved May 10, 2022, from: https://hbr.org

[3]. Alexander Osterwalder, Yves Pigneur (2011). Business Model Generation: A Handbook for Visionaries, Game Changers and Challengers. Wiley USA 2010-2011

[4]. Hong, Y. C., & Fauvel, C. (2013). Criticisms, variations and experiences with business model canvas.

[5]. AVENUE project, https://h2020-avenue.eu/

[6]. SHOW Grant Agreement No. 875530 (2019).

[7]. AVENUE (2020). D4.5 Second iteration In-vehicle services. Deliverable of the Horizon-2020 AVENUE project.

[8]. European Commission (2020). A fundamental transport transformation: Commission presents its plan for green, smart, and affordable mobility. Retrieved May 10, 2022, from: https://transport.ec.europa.eu

[9]. UITP (2020). SUSTAINABLE AND SMART MOBILITY: UITP input to the European strategy (2022). Retrieved 10 May 2022, from https://cms.uitp.org/wp/wp-content/uploads/2020/09/20200924_UITP_contribution_EUSSMS_final.pdf

[10]. Novak, Thomas & Hoffman, Donna. (2001). Profitability on the Web: Business Models and Revenue Streams.

[11]. Meyer, R. (2022, March 22). Revenue model framework. TIAS. Retrieved June 2, 2022, from https://www.tias.edu/en/item/revenue-model-typology-revenue-model-framework

[12]. Gilsing, R., Wilbik, A., Grefen, P. et al. Defining business model key performance indicators using intentional linguistic summaries. Softw Syst Model 20, 965–996 (2021). https://doi.org/10.1007/s10270-021-00894-x

[13]. van de Ven, Montijn & Machado, Paola & Athanasopoulou, Alexia & Aysolmaz, Banu & Turetken, Oktay. (2022). Key Performance Indicators for Business Models: A Review of Literature. Retrieved July 27, 2022, from https://www.researchgate.net/publication/360109925_Key_Performance_Indicators_for_Business_Models_A_Review_of_Literature

[14]. Gartner_Inc. Definition of cost structure - gartner finance glossary. Gartner. Retrieved May 29, 2022, from https://www.gartner.com/en/finance/glossary/cost-structure

[15]. 50MINUTES (2017). The Business Model Canvas: Let your business thrive with this simple model. 50Minutes.com

[16]. AVENUE (2020). D8.3 First Iteration Economic impact. Deliverable of the Horizon-2020 AVENUE project.

[17]. GitLab team, “What is DevOps?”. Last Accessed: 01 June 2022. Available online: https://about.gitlab.com/topics/devops/

[18]. GitLab. Last Accessed: 07 July 2022. Available online: https://about.gitlab.com/

[19]. Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. ACM Computing Surveys (CSUR), 52(6), 1-35.

[20]. IEEE Standards Committee. (2021). IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment: IEEE Standard 2675-2021.

[21]. Dyck, A., Penners, R., & Lichter, H. (2015). Towards definitions for release engineering and DevOps. In 2015 IEEE/ACM 3rd International Workshop on Release Engineering. IEEE. https://doi. org/10.1109/releng.

[22]. Erich, F. M., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. Journal of Software: Evolution and Process, 29(6), e1885.
[23]. Gokarna, M., & Singh, R. (2021, February). DevOps: a historical review and future works. In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 366-371). IEEE.
[24]. Farroha, B. S., & Farroha, D. L. (2014, October). A framework for managing mission needs, compliance, and trust in the DevOps environment. In 2014 IEEE Military Communications Conference (pp. 288-293). IEEE.
[25]. Utami, E., & Al Fatta, H. (2021, August). Analysis on the Use of Declarative and Pull-based Deployment Models on GitOps Using Argo CD. In 2021 4th International Conference on Information and Communications Technology (ICOIACT) (pp. 186-191). IEEE.
[26]. Sharma, S. (2017). The DevOps adoption playbook: a guide to adopting DevOps in a multi-speed IT enterprise. John Wiley & Sons.
[27]. Dr. Darrell R. Schrag "DevOps: Shift Left to Reduce Failure". Last Accessed: 02 June 2022. Available online: https://devops.com/devops-shift-left-avoid-failure/
[28]. Arman Karman. "DevOps is all about facilitating the Flow of Value to your customers." Last Accessed: 02 June 2022. Available online: https://medium.com/@armankamran/devops-insights-part-1-its-all-about-the-flow-777f2c71e843
[29]. CA technologies team. "Taking DevOps to the next level. Putting customer first". Last accessed: 02 June 2022. Available online: https://docs.broadcom.com/doc/taking-devops-to-the-next-level-putting-the-customer-first
[30]. Saito, H., Lee, H. C. C., & Wu, C. Y. (2019). DevOps with Kubernetes: accelerating software delivery with container orchestrators. Packt Publishing Ltd.
[31]. Docker documentation. "Docker Overview". Last accessed: 02 June 2022. Available online: https://docs.docker.com/get-started/overview/
[32]. Kubernetes documentation. "Overview". Last accessed: 02 June 2022. Available online: https://kubernetes.io/docs/concepts/overview/
[33]. Docker documentation. "Swarm mode key concepts". Last accessed: 03 June 2022. Available online: https://docs.docker.com/engine/swarm/key-concepts/
[34]. Docker documentation. "Swarm mode overview". Last accessed: 03 June 2022. Available online: https://docs.docker.com/engine/swarm/
[35]. Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. Ieee Software, 33(3), 94-100.
[36]. Erich, F. M., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. Journal of Software: Evolution and Process, 29(6), e1885.
[37]. Díaz, J., Almaraz, R., Pérez, J., & Garbajosa, J. (2018, May). DevOps in practice: an exploratory case study. In Proceedings of the 19th International Conference on Agile Software Development: Companion (pp. 1-3).
[38]. SHOW Hackathon. Last accessed: 06 June 2022. Available online: https://show-project.eu/2022/03/07/applications-open-show-hackathon-2022/
[39]. SHOW Hackathon. "We have a winner". Last accessed: 06 June 2022. Available online: https://show-project.eu/2022/03/28/the-show-hackathon-we-have-a-winner/

Appendix I

Table 7: Proposed Pricelist for the SHOW Marketplace (as a business product).

For product items sold from us directly, a SaaS (monthly or annual) or a Perpetual Licensing Model will be used.					
For mixed categories (if there any), the pricing policy might differ.					
A Listing Fee might be added at a later stage.					
For every successful payment processed through the Marketplace a fee of 0.30€ might be charged (not yet defined if it will be utilized).					
A Cost per Click value might be added at a later stage.					
Pricing group	**Category**	**Listing Fee**	**CPC (€)**	**CPS Commission (%)**	**Pricing Model**
Solution	Service	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS
Solution	Tool	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS
Solution	Application	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS
SHOW specific material	Architecture	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS
Software Component	UI widgets	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS
Software Component	Dashboards	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS

Software Component	Data Model	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS
Other	Algorithm	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS
Other	Single Component	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS
Other	Case study	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS
Other	Dataset	None at the early business stages.	None at the early business stages.	5-15	SaaS Licensing or Perpetual Licensing or CPS