D5.3-Documentation (white goods demonstrator)

Project acronym:	**ReCiPSS**
Project full title:	**Resource-efficient Circular Product-Service Systems — ReCiPSS**
Grant agreement no.:	**776577-2**
Author/s:	Ruud de Bruijckere, Pontus Blomberg, Aleš Mihelič, Simon Kotnik , Mitja Štimulak, Iulia Stefan
Reviewed:	**PDS, SIG,MU, GOR & KTH**
Approved:	**Amir Rashid**
Document Reference:	**D5.3**
Dissemination Level:	**PU**
Version:	**0.7 DRAFT/FINAL**
Date:	**10.06.2020**

This is a draft document and subject to approval for final version. Therefore the information contained herein may change.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

History of Changes

Version	*Date*	*Modification reason*	*Modified by*
0.1	17.02.2020	Initial Draft	Dana Oniga
0.2	17.04.2020	Architectural overview and Backend Server	Ruud de Bruijckere
0.2.1	20.04.2020	Service instructions – Augmented Reality	Pontus Blomberg
0.2.2	22.04.2020	RecipSS Webstore	Iulia Stefan
0.3	28.04.2020	Use cases and Components of IoT platform	Mitja Štimulak Aleš Mihelič Simon Kotnik
0.4	06.05.2020	Peer Review	Pontus Blomberg
0.5	11.05.2020	Peer Review	Ruud de Bruijckere
0.6	22.05.2020	Peer Review	Alena Klapalova
0.7	08.06.2020	Final draft review	SIMAVI, GOR, PDS, SIG & KTH

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

Table of contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

List of figures

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

List of tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

List of abbreviations

Abbreviation	*Explanation*
3CX	Call Center system
API	Application Programming Interface
AR	Augmented Reality
AT / UAT	Acceptance test / User acceptance test
AuId	Appliance unique identifier, unique number for the washing machine
BC	Business customer
BR	Business requirements
BUC	Business use case
CAD	Computer-aided design
CRUD	Create, Read, Update, Delete functionality
CT	Component tests
DD	Design document
ERP	Enterprise resource planning
FR	Functional requirements
FT	Functional tests
GDPR	General Data Protection Regulation
GSD	Gorenje Service Documentation
ICT	Information and communications technology
IoT	Internet of things
IT	Information technology
JUConnect	Gorenje's cloud platform for connecting washing machines
NFR	Non-functional requirements
PLM	Product Lifecycle Management
PPU	Pay per use
PPU Flag	Status of WM dedicated to Pay per use
PVZ	ProductView Packaged Structure And Drawing
PTC	American Software Company
ReCiPSS	Resource-efficient Circular Product-Service Systems
SAD	Software architectural design
SAG / SAHG+	Gorenje's service application
SAP	Enterprise resource planning software from the company SAP
SBU	Service Business Unit
SIT	System integration tests
SW	Software
UAT	*See AT*
USP	Unique selling point
WM	Washing machine

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

1. Executive Summary

Deliverable D5.3 Documentation (white goods demonstrator) is a result of Task 5.3: Integrating, testing and documenting the IoT platform for the white goods demonstrator. The report summarizes activities performed between M7 to M24 of ReCiPSS project, within tasks T5.1: Implementing an IoT platform for white goods for monitoring asset usage and condition and T5.3: Integrating, testing, and documenting the IoT platform for the white goods demonstrator.

The main activities in this period were to finalize and test the four main components of the Whitegoods ICT platform: the **ReCiPSS backend server,** the **Pay Per Use web store,** the **JUConnect cloud platform** and the **Connected washing machines**. This includes tests of each component separately and tests of end-to-end uses cases to test the main use cases of the ICT platform: **create** a Pay Per Use contract, **deliver an appliance** for the Pay Per Use contract, **create monthly bills** for the Pay Per Use contract, **terminate** a Pay Per Use contract, **replace** a Pay Per Use appliance within the contract terms. Additionally, use cases include: **provide feedback** to encourage more cost-effective and sustainable use of the appliance and **intelligent service instructions** based on augmented reality technology.

The four components have been developed and tested separately until M24 and integration testing will continue until M31. The platform will gradually increase its functionalities throughout the demonstration where feedback from users and stakeholders will be taken into account. For providing feedback intelligent service instructions will be introduced in a later iteration.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

2. Introduction

WP5 - ICT Platforms for Multiple Life cycles is one of the most important work packages in ReCiPSS project, representing the foundation for both WP6 - White Goods Demonstrator and WP7 - Automotive Parts Demonstrator.

One the main tasks of WP5 is the development of the cloud-based IoT platform for white goods (T5.1), that will support circular manufacturing systems in the context of the white goods demonstrators, including smart sensing and communication technologies which can monitor customer behaviour, asset usage and conditions and communicate that information for predictive maintenance. Deliverable D5.1-IoT platform supporting smart washing machines, submitted at the end of November 2019 (M18) gave more details about the scope of the IT solution, the stakeholders with their concerns and needs, the business use cases, the chosen solution architecture and the development activities performed at that time.

Deliverable D5.3 Documentation (white goods demonstrator) is a result of Task 5.3: Integrating, testing and documenting the IoT platform for the white goods demonstrator, and is focused on the integration of the IoT module in a washing machine enabling the gathering of sensor information and communication with the cloud.

2.1. Document Scope

The scope of this document is documenting development, integrating and testing of the Whitegoods ICT platform, as of M24 of the ReCiPSS project.

2.2. Document Structure

This document will discuss the implementation of the Whitegoods ICT platform in the following order. In chapter 3, the business requirements are discussed that specify the needs and goals of the Pay per Use (PPU) business model. In chapter 4, the use cases are described that fulfil these business requirements. Chapter 5 discusses the overall solution design for the Whitegoods ICT platform and in chapter 6 each component will be discussed in detail. Chapter 7 discusses the testing of the separate components and the integration and acceptance testing of the combined components. Chapter 8 concludes the document.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

3. Business requirements

3.1. Description of the Business context

GOR business **model is nowadays linear** and is oriented only in production and sales of domestic appliances. Currently, the company makes money through sales transactions and after-sales trade.

Project ReCiPSS brings into the company revolution in ways of doing business. During the project, Gorenje is going to implement circular thinking into newly developed “pay per use” service business model. In this case, the company will bring to the market a new service and will keep owning the WM. On the other side, the user will only pay for the usage of WM.

The project ReCiPSS tries to solve the contradiction between the overproduction of cheap short-lived products and sustainable production with high-quality long-lasting products.

The idea explored in this project is simple. The manufacturer should sell the service of quality washed clothes, rather than a washing machine appliance. The customer is not worried about the price of a washing machine and related repair costs. The manufacturer, therefore, gains the incentive to build "build to last" products that will be making money for years to come. This will reduce the consumption of natural resources and at the same time provide high-quality products to the market.

The ReCiPSS project for Gorenje is not only to develop pay per use model for the washing machines, but also to develop a new business model that can include any connected appliance produced by Gorenje or Hisense. Professional washing machine ASKO WM75.C and WM85.C were chosen for the demo because the nature of this product is already very much aligned with the principles of the eco-friendly production and refurbishment process and has smart machine functionalities. This product is built to last, with regular checkups or refurbishments it should be able to be in operation for at least ten times longer than current domestic washing machine in the market.

The main customers of this appliance are typically shared laundry rooms in the multi-apartment buildings and small and medium enterprises, such as laundry saloons or small hotels. These appliances can be a heavy financial burden for these customers and it makes sense for them to rather pay for what they are actually using with included service costs than buying an appliance. There is an additional benefit of this model in the scenarios of the shared usage, as each customer is billed for what is washing and not paying a flat rate regardless of the usage.

There is no reason to limit the offer only to above-mentioned business users, as the product can equally well serve the private users, even families. Our calculations showed that our product in the second life cycle gets very competitive comparing to the new domestic machines on the market, with the added value of using a well build and high-quality machine produced from steel rather than plastic and optimized for energy-efficient washing reducing the total cost of ownership of the product.

This demo project will be performed in Slovenia, Austria, Netherlands, and Denmark. At first ASKO Professional washing machines will be deployed, with possible expansion to other WM brands.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 776577-2

SBUs in demo countries are responsible for marketing, sales, organization of local logistic support to and from the customer and repair/maintenance/regular check-ups. The SBU service can include delivery and installation, depending on the contract.

3.2. Key features of the Whitegoods ICT platform

The architecture of the Whitegoods ICT platform has the following key characteristics:

- Maintainable using standard proven technologies
  The platform is based on well-known frameworks and technologies like C#, ASP.Net, SQL server, SAP Hybris. This ensures that support and developers will be available for a long time from now.
- Flexible using a loosely coupled architecture
  The platform is open for additional appliance types, service options and markets. The platform is designed with loosely coupled components that communicate through standard REST services which decreases the dependency between the components and makes it easier to integrate new components.
- Cost-effective by reusing existing systems and processes
  The platform is not built from scratch to provide all the services and features needed to provide a Pay Per Use business model. The design carefully reuses already existing functionality and complements it with the needed information, functionality and services.
- Preserving Privacy and comply with GDPR
  The platform only stores data needed for Pay Per Use billing, e.g. the usage information that is needed to create the monthly Pay Per Use bill. All other personal data is stored in existing user profiles in SAP-PI for which processes for ensuring GDPR compliance already are in place. The usage data will be purged after the bill is created or, after written agreement from the customer, the information will be anonymised and used for improving the Pay Per Use offer.

The platform provides the following key functionality:

- Create and manage Pay Per Use contracts
- Plan and execute Pay Per Use appliance deliveries
- Create Pay Per Use bills
- Support maintenance and repair of Pay Per Use appliances
- Plan and execute pickup and replacements for Pay Per Use contracts
- Monitor and control connected appliances
- Provide feedback on sustainable use of Pay Per Use appliances
- Empower users and owners to perform maintenance and repair with augmented reality

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

4. Use cases

4.1. Pay per use contract creation

One of the most important use cases is Pay per use contract creation. **Error! Reference source not found.** below shows the flow of activities needed to create a pay per use contract:

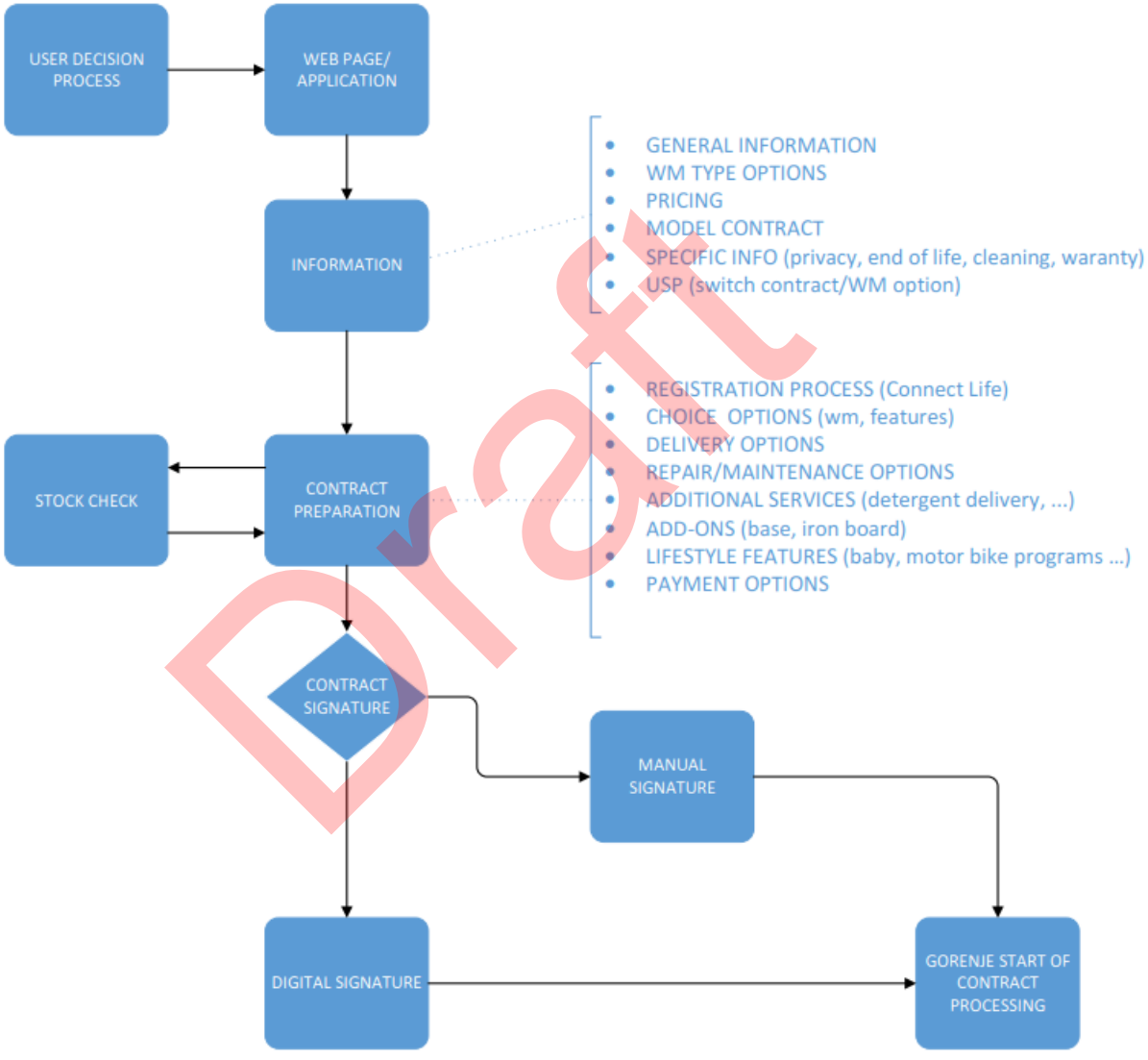

Figure 1: Pay per Use flow

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

When a customer decides that he/she wants to sign a contract and start using service he/she must first register with the online registration process by filling the registration form. In the registration form data that is needed for contract preparation and appliance delivery is inserted in the information system.

The user enters the data in the registration form

1. User ID (email)
2. User Name and Surname
3. Social accounts – if a user selects to use supported social service as credentials providers
    a. Account provider name (Facebook, Google, …)
    b. User profile ID (user ID at account provider)
4. Email
5. Password – only if no external credential provider is used
6. Delivery address
    a. Other name and surname if not the same as for billing
    b. Street
    c. House number
    d. Floor
    e. Postal code
    f. City
    g. Country
7. Billing address
    a. Street
    b. House number
    c. Postal code
    d. City
    e. Country
8. Contact phone:
    a. Type (fixed, mobile)
    b. Country and country-code
    c. Number

After filling the registration data on the register button, the user is presented with a "registration successful" screen with the announcement that registration confirmation email was sent then the user selects the washing machine on the webpage. Depending on the stock, only available washing machine types according to the chosen country of delivery are displayed to the customer.

After this customer selects fixed service packages and desired flexibility of contract, duration, service and maintenance options, etc. These options may be gathered in the pre-selections, such as Gold, Silver and Bronze packages. Price for the chosen options is displayed and adjusted according to the selection. A customer makes obligatory choices about payment options and communication channels i.e. email. In the end, the customer makes choices about optional additional features/services. The extra costs of the personal choices are added onto the price per wash or monthly fee etc.
By submitting the order, the contract becomes valid and binding for both parties. At the end of order process users receive a pdf version of the contract, the service provider also stores the contract to its databases.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

Customer than pays initial fee, if this is the chosen option. User provider starts appliance delivery process and contract starts to be executed.

4.2. Usage tracking

Connected washing machines can continuously check the status of vital parts of the washing machine. It records around 140 life cycle parameters and reports their values in real-time intervals to the cloud. These parameters track the usage of every major electrical and mechanical washing machine component.

When the machine is in the pay per use status (flag PPU FLAG is set to TRUE), then the time of wash is logged in a separate table in the washing machine. This table is called credit table.

Credit table has three properties (Program ID - PID, TIME, and Wash finished).

The time in this table is updated each minute and it must represent the actual time the specific program is running as accurately as possible.

The customer starts the program with specific settings on WM.

- WM checks PPU_flag (pay per use) in internal memory if it is set to TRUE.

The washing machine starts the program:

- The WM checks that the selected program is stored in the credit table under PID.
- Minutes of the program in program run state is stored in credit table (this is updated every minute)
- Program finished successfully:
    - Time is updated
    - Wash finished status is set to TRUE
    - The ReCiPSS backend is pulling these parameters after wash finished parameter is set to true for individual wash billing is performed and wash finished parameter set to FALSE.
- When wash finished parameter is set to from true to false on the appliance, wash entry is deleted.

In PPU mode there is additional information available to the customer on the UI about the cost of the program and eco sign.

- At pairing the data about the cost of the program is gathered from the cloud
- Based on this data €/$ signs are presented on UI
- ECO sign is displayed based on information in the WM. (It is not synced via the cloud)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

Price determination and cost calculation:

- Sales agent can create »Program price plan (for each program Cotton, Synthetics, Wool, ….)« where he/she defines »Program cost table« for each available program and connects this cost table with a contract
- Cost of the program is based on “Program cost table”
  - WM sends data PID, CHARGED TIME to the RECIPSS backend
  - RECIPSS backend calculates the costs
  - There is a minimum price defined for each program independent on the above calculation that you have to pay when you start the program.

4.3. PPU billing

There are several options supported, including:

- Customer is post-billed: At the end of the month RECIPSS backend sums all the costs and sends them to the payment module, where bill is generated.
- Customer billing pre-paid: The application send data to backend continuously. Costs are deducted from user credit.

4.4. Feedback

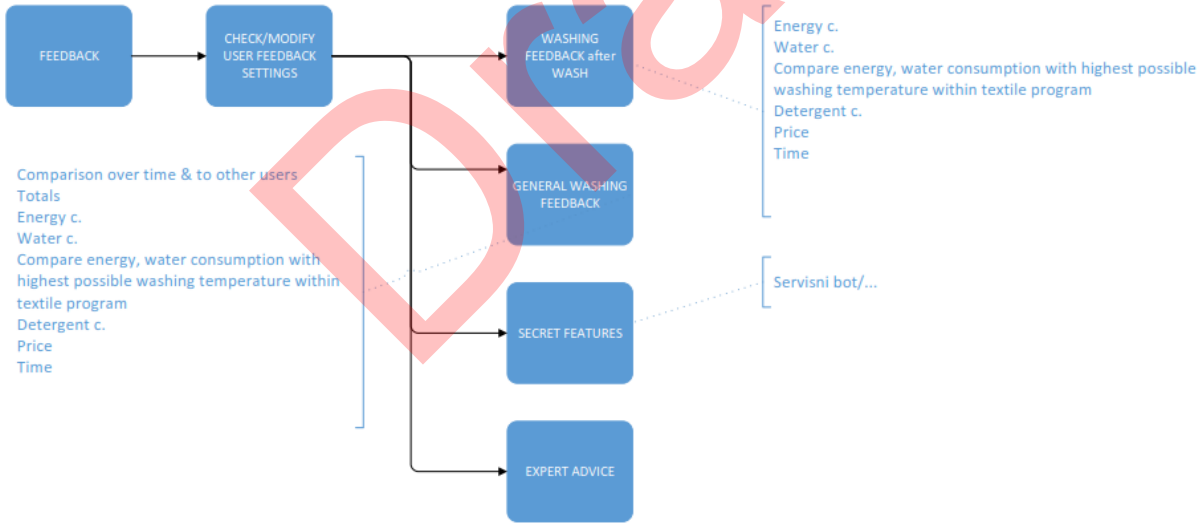

Figure 2: Feedback flow

On the mobile application user of pay-per-use washing machine service can also receive feedback related to the previous washing machine usage.

One can choose

- get feedback on the cost structure of all my laundry costs
- get feedback about the resource consumption so that user can see how much water, energy, and money, detergent was used

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 776577-2

- get feedback on his/her user behaviour and money I saved with it over a period of time
- comparison to others

Good ECO performance Rewards

i. Virtual rewards and statuses (possible to share on Facebook,Twitter…) --> Acknowledgment: Gorenje issues a virtual certificate

ii. Material rewards (free washes, discounts, other products, planting real trees)

The app shows different types of awards/rewards systems: green (growing a tree) or monetary (saving money, collecting discounts/coupons) or competitive (beating high scores/other users).

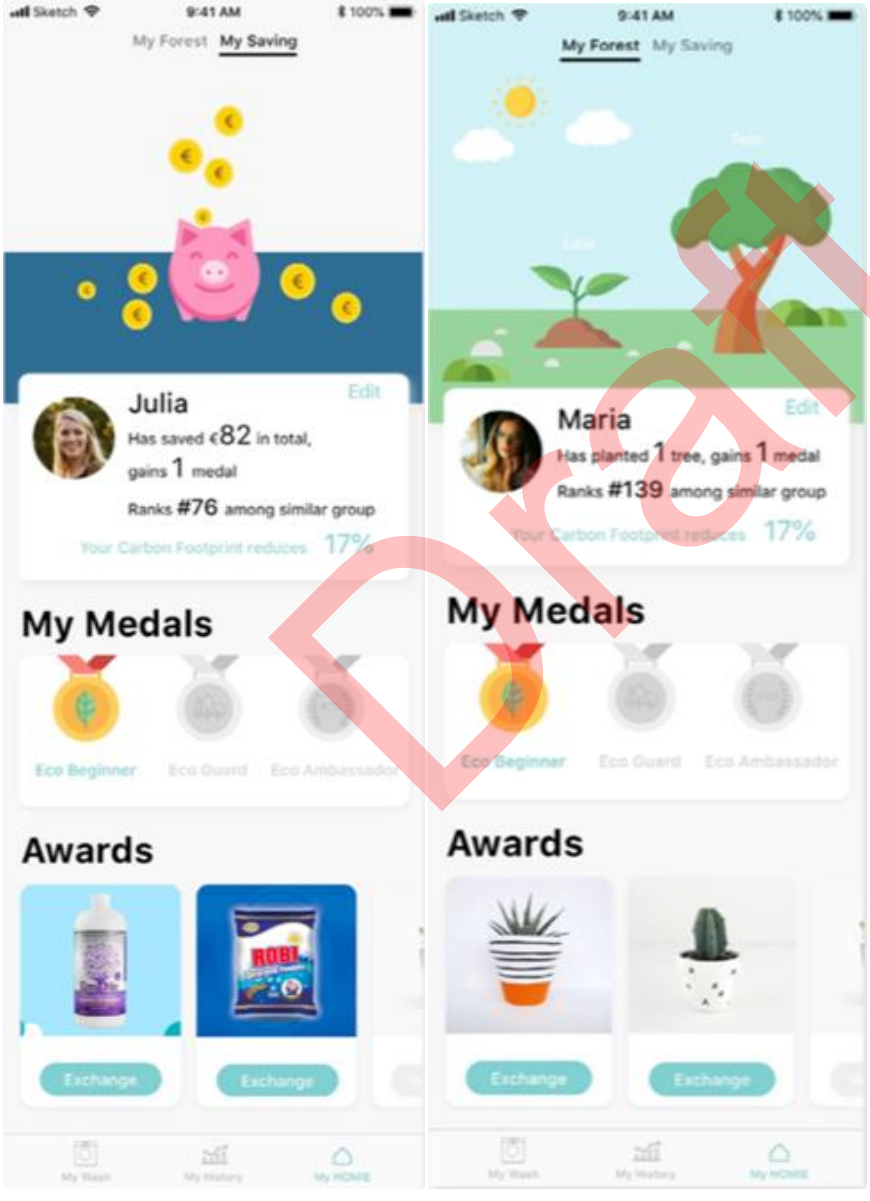

Figure 3: Awards samples

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

4.5. Replacement and termination of contract

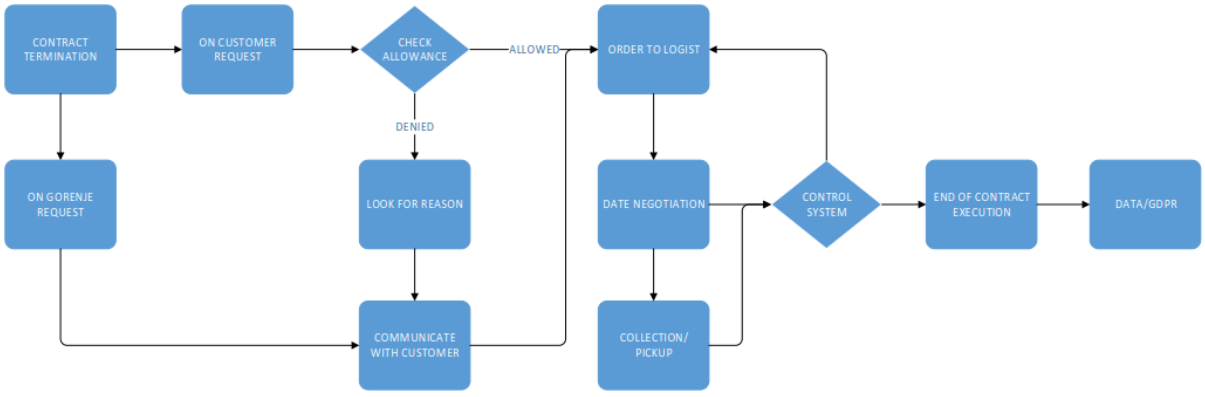

Figure 4: Replacement and termination flow

When the contractor or business customer terminates the contract or at the end of the regular contracts period the WM needs to be picked up and delivered back to the warehouse. The same conditions are valid for replacement in case of WM breakdown.

If allowed:

1. Gorenje issues an order for transport to logistics
2. Logistics negotiate a date of picking up the WM
3. Logistics collect the WM
4. Logistics deliver WM to a local warehouse
5. Gorenje ends the contract period with the date of pick-up

If denied:

1. Gorenje looks for the reason
2. Gorenje contacts customer
3. Gorenje issues an order for transport to logistics
4. Logistics negotiate a date of picking up the WM
5. Logistics collect the WM
6. Logistics deliver WM to a local warehouse
7. Gorenje ends the contract period with the date of pick-up

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

Below are the steps to be followed when Gorenje requests the contract termination:

1. Gorenje requests contract termination (due to non-paying …)
2. Gorenje informs the user about contract termination
3. Gorenje issues an order for transport to logistic
4. Logistics negotiate a date of picking up the WM with costumer
5. Logistics collect the WM
6. Logistics deliver WM to a local warehouse
7. Gorenje ends the contract period with the date of pick-up.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

5. Solution design

5.1. IoT Platform for WhiteGoods Architecture

The Whitegoods demonstrator ICT platform enhances Gorenje's existing ICT platform so that Gorenje can support new business models for the Circular Manufacturing and Circular Economy.
The business model at hand is a Pay Per Use business model for Gorenje washing machines. By offering washing machines through a Pay Per Use model, the washing machines can go through multiple life cycles. They will be owned by Gorenje, who will take care of logistics, replacement, refurbishment and maintenance. In this way the total lifetime of a washing machine can be extended significantly, contributing to a more sustainable society.
One of the challenges is that the Whitegoods demonstrator ICT platform is not a new platform from scratch but must be implemented within the boundaries of a platform already in place. It needs to coexist with the existing infrastructure, enhance it but not interfere with the current functionality and processes.
The Pay Per Use ICT platform architectural overview is illustrated in **Error! Reference source not found.**.
The components of the ICT platform that already exist today are shown in dark grey: the SAHG+ service application, the SAP-PI Master data profiles and the SAP ERP economical system. The Webstore frontend, Webstore backend and the ReCiPSS backend server are developed especially for the Pay Per Use business model. The washing machine and JUConnect IoT platform are enhanced with smart connectivity to support the Pay Per Use monitoring and control needs.

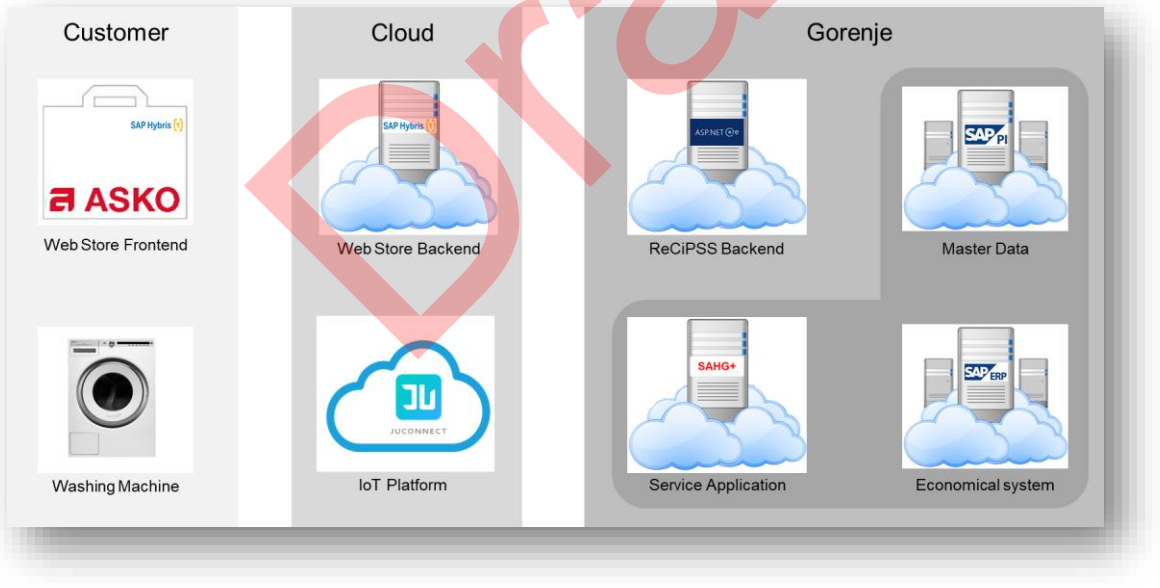

Figure 5: Solution design pay per use ICT platform

Figure 6 describes the cooperation between the components during the use cases for creating a Pay Per Use contract, delivering the appliance and set up the system for future payments. Contract creation is initiated in the ReCiPSS Web store that is based on the SAP Hybris framework. The contract is managed and stored in the ReCiPSS backend server that communicates with the service application SAG and the enterprise resource planning application SAP to ensure delivery and billing respectively.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

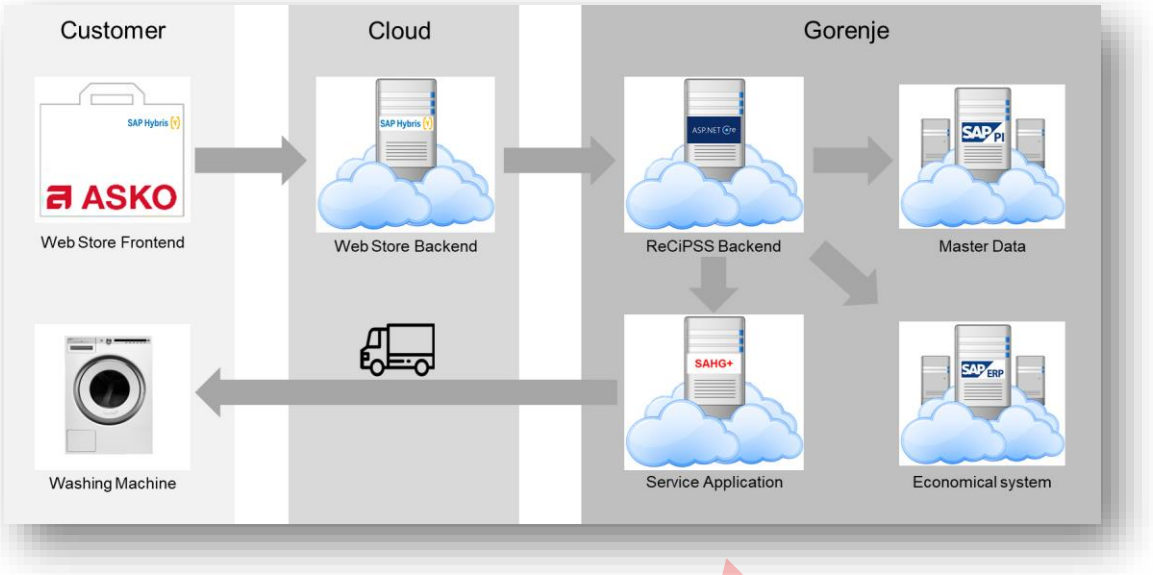

Figure 6: Create contract and delivery appliance use cases

Figure 7 describes the cooperation between the components during the use cases for usage, monitoring, control, feedback and creating the Pay Per Use bills. The washing machine connects to the JUConnect IoT platform to report the usage of the machine. This information is received by the ReCiPSS backend server. User and appliance profiles are retrieved from the Enterprise Server Bus (ESB) that is implemented by SAP PI. The ReCiPSS mobile application connects to the backend server through the JUConnect API.

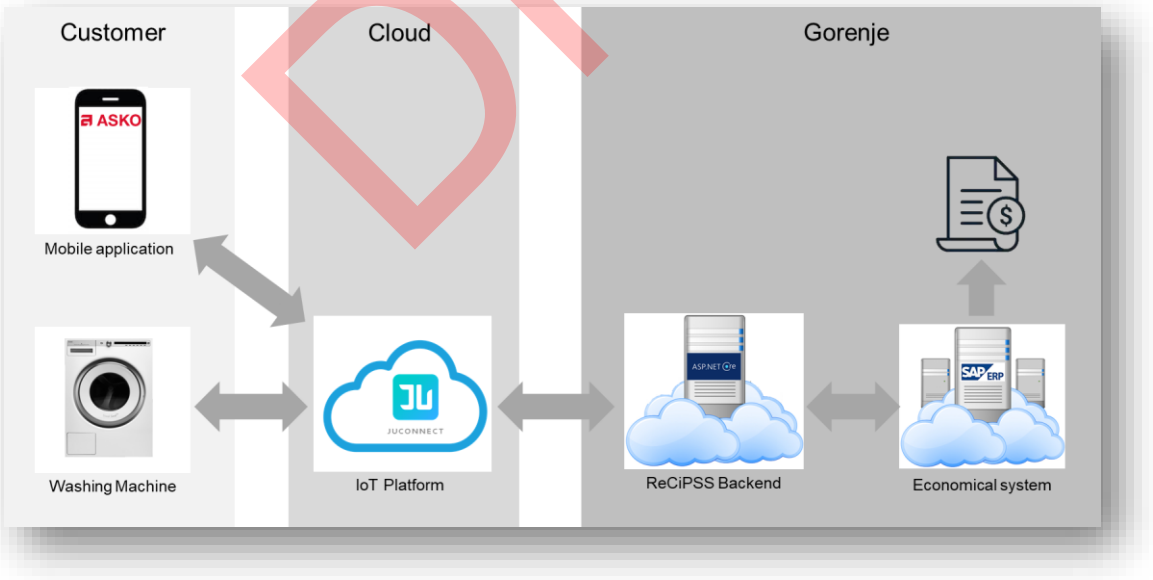

Figure 7: Usage, monitoring, control, feedback and Pay Per Use billing scenarios

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

5.2. Development Methodology

For the project management process, we choose Agile Methodology, which is one of the adaptive software development approaches, close to Rapid Application Development (RAD) methodology. [1]

“Agile software development comprises various approaches to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s). It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change.” [2]

In this way, we ensure changes can be made quicker and throughout the development process by having consistent evaluations and keeps the project transparent too by having regular consistent meetings that allow everyone involved to access the project data and progress.

The process involves breaking down the project into prioritized requirements and delivering each individually within an iterative cycle. Each iteration is reviewed and assessed by the development team and everyone involved. The insights gained from the assessment are used to determine the next step in development.

5.3. Technologies used

The ReCiPSS backend server is a REST web service, written in the C# programming language and deployed on Microsoft’s ASP.Net Core platform for web applications. The code is structured according to the Model-View-Controller (MVC) pattern and data is formatted as json-files. Microsoft’s Entity Framework is used for persisting information into a Microsoft SQL Server database.

The WebStore front- and backend are based on SAP Hybris eCommerce platform. This is a proprietary platform from the SAP company and uses the following underlying technologies: Spring MVC, JSF (Frontend), REST Web services, JSP, JavaScript, CSS, bootstrap.

The Augmented Reality (AR) content is mechanically designed in 3D CAD. The CAD is then imported into Creo Illustrate where the model is prepared and re-structured and animation sequences are applied. All engineering excellence including product structure, hierarchy and metadata is transferred throughout this process. When the Illustrate step is finalized the file is saved to a PVZ extension file (ProductView Packaged Structure And Drawing).

The AR application is then built in PTC’s AR content and experience creation tool, Vuforia Studio. Here the User interface as well as the related 3D content is defined. The PVZ file from Creo Illustrate is imported and further optimised to allow for easy consumption in a mobile or head warn device. The AR experience is uploaded to the Vuforia Experience Service and is then ready to be consumed through Vuforia View. Vuforia View is a free App available for iOS, Android and some dedicated head warn devices like the Realwear HMT devices (hands free remote collaborative tools) and the Microsoft HoloLens devices. Later on, the AR experience can be populated with connected business system data and live IoT data.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 776577-2

6. Components of IoT platform- development

6.1. ReCiPSS Web Store

The ReCiPSS web store is the front-end of the IoT platform, where user can get the information about the pay per use model and immediately create the contract/ buy the service. ReCiPSS Webstore is developed using SAP Hybris. Hybris core is an e-commerce platform that has several stand-out features which allow developing a full omni-channel solution. This means you can sell online, mobile, in-store and via a contact centre. The development was made using the B2C accelerator. Each of the Commerce Accelerator applications include extensions that offer ready-to-use storefronts and use state-of-the-art web technologies such as Spring MVC 3, the Blueprint CSS framework , jQuery JavaScript library, HTML and Bootstrap, all integrated into the modular SAP Hybris Commerce platform.

6.1.1. Login/Register

The customer can register into the platform by clicking the Login/Register button in the top right corner of the page, displaying a page with two forms – one for returning customers, the ones that already have an account, and one for new customers, those that are registering for the first time. The form for creating an account will prompt users to input their name and surname, the title that they use and their contact information (e-mail address and telephone number).

For the login functionality, the user is prompted to provide his e-mail and the password that he has chosen during the registering process. He will then be redirected to the homepage, where he can begin his shopping.

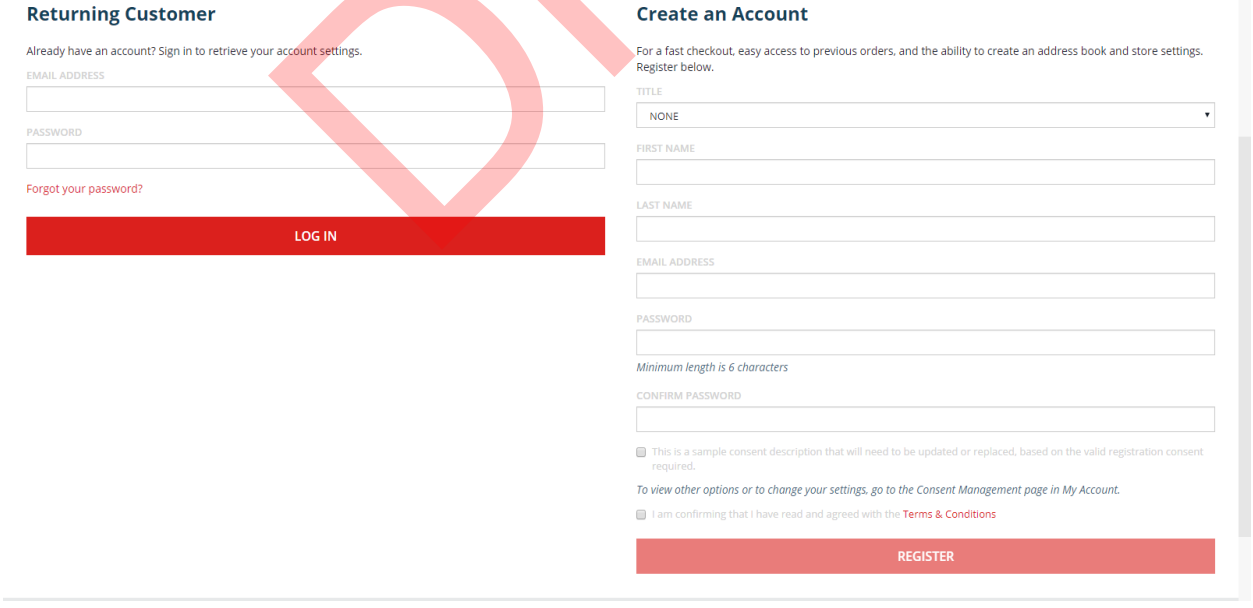

Figure 8: Login/Register page

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

6.1.2. Homepage

All pages are divided into three main parts – header, body and footer. The header and the footer are mostly the same on every page with only the page body contents being different for each individual page.

The header consists of a section where the Login button that was previously used by the customer is being replaced by a hello message, along with a sign out button and a dropdown menu consisting of account settings.

There is also a section that displays a search bar, where a customer may input some details about a product that he wishes to see, and three buttons: one that opens a store-finder page, one that allows the user to import a previously created shopping cart and one that displays the current cart information.

The final section of the header consists of a bar with multiple buttons, which allows the user to navigate the web store.

The page body content shows the main information about the products and the philosophy behind it.

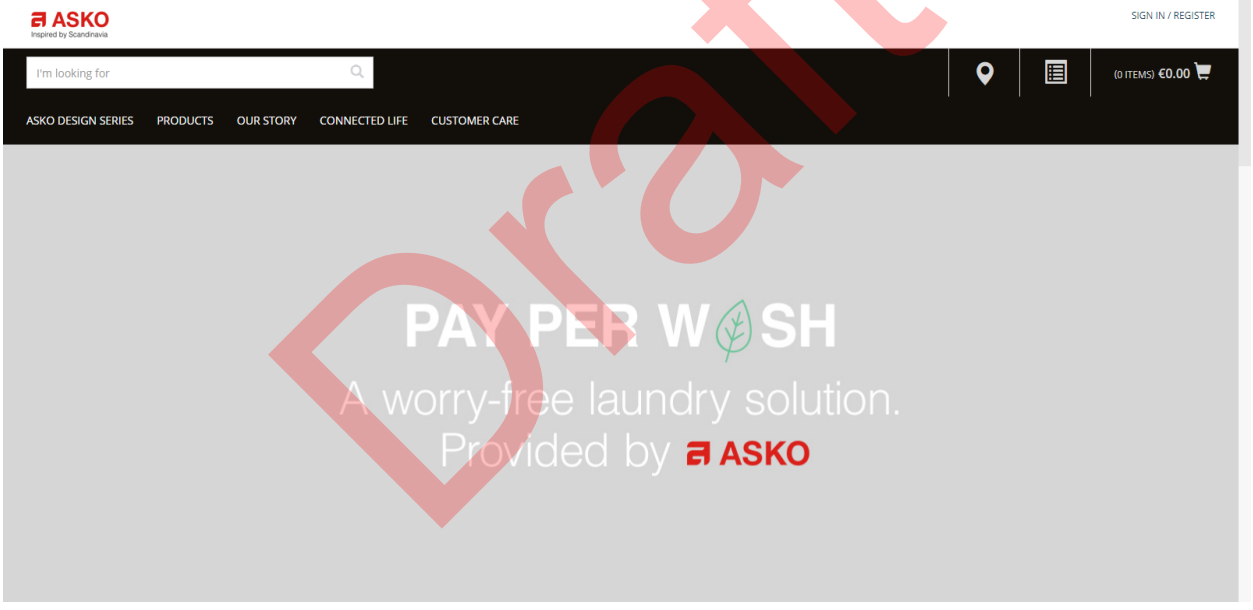

Figure 9: Homepage first part

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

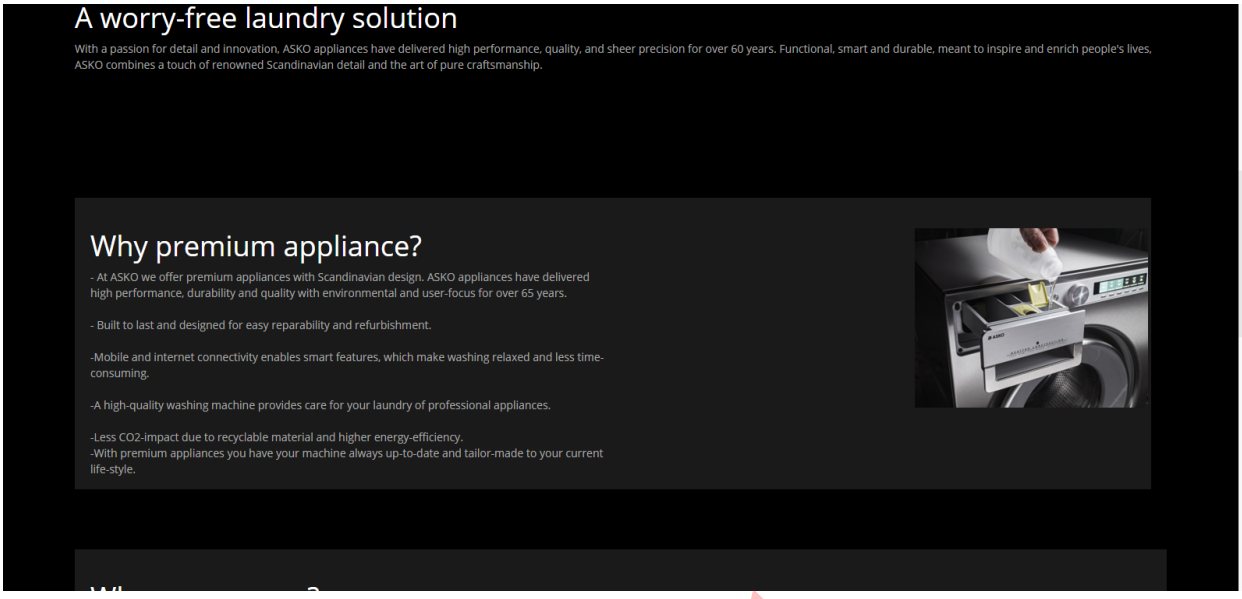

Figure 10: Homepage second part

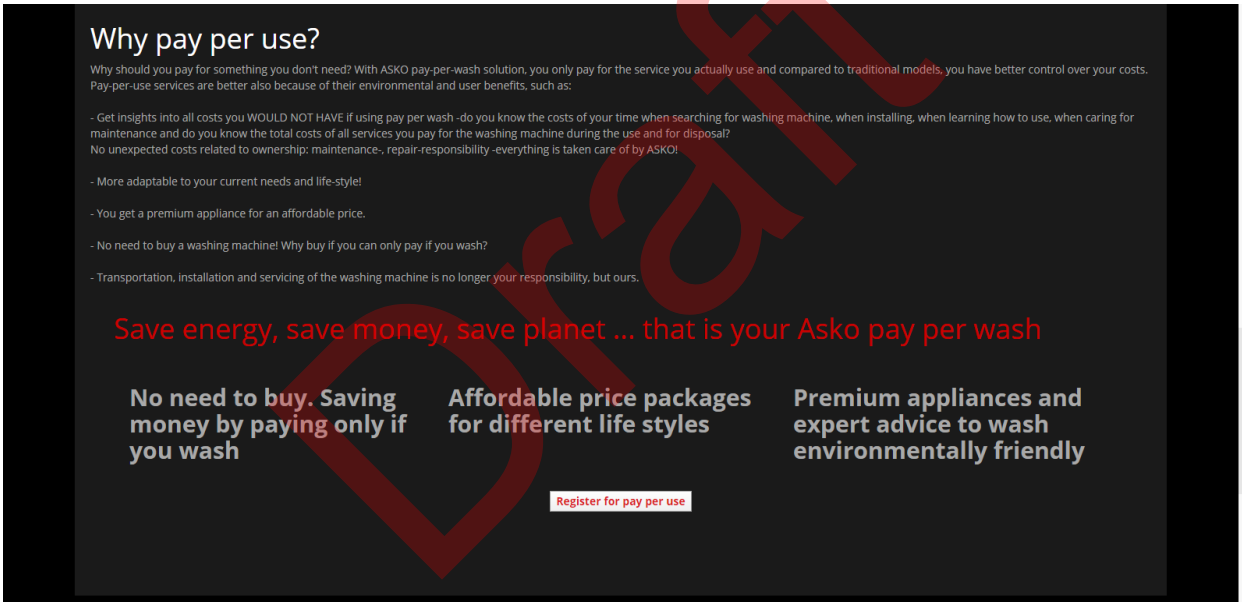

Figure 11: Homepage third part

6.1.3. Products page

This page can be accessed by different means. Customer can access this page through the search bar or through the “Products” buttons displayed in the category navigation section. The products page displays all the products that match the customers search if it was accessed by inputting some information in the search bar, or all the products available. By default, the products are displayed by their relevance to the customer, taking into account the information requested in the search bar.

The items are displayed paginated, with buttons that help the user navigate through the multiple pages of products.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 776577-2

Any product can be added to cart by using the specific button or the user may click on the product image in order to display more information about the selected item.

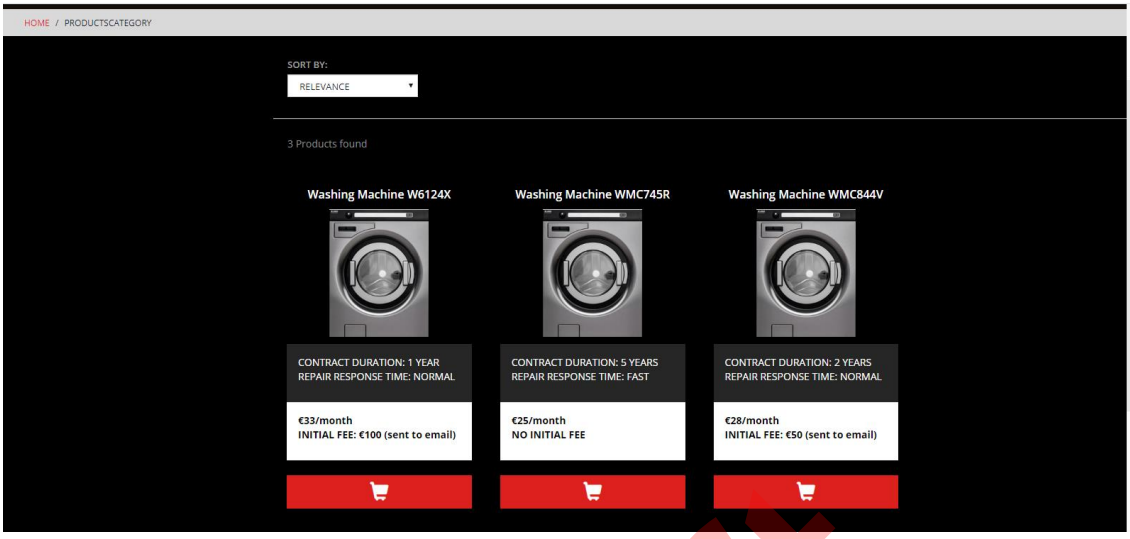

Figure 12: Products page

6.1.4. Product details page

The product details page displays all the information available on the selected product. On top of the page are written the name and the product code. User has the possibility to write a review for the product. The product's picture is displayed underneath those.

The price is displayed on the right side of the picture and there is also a short description of the product, as shown in Figure 13.

On the right side of the page, there is a section where stock information is displayed. This is where the user can see if the product is available, how many pieces are in stock and the customer may add the product to the cart by clicking a button.

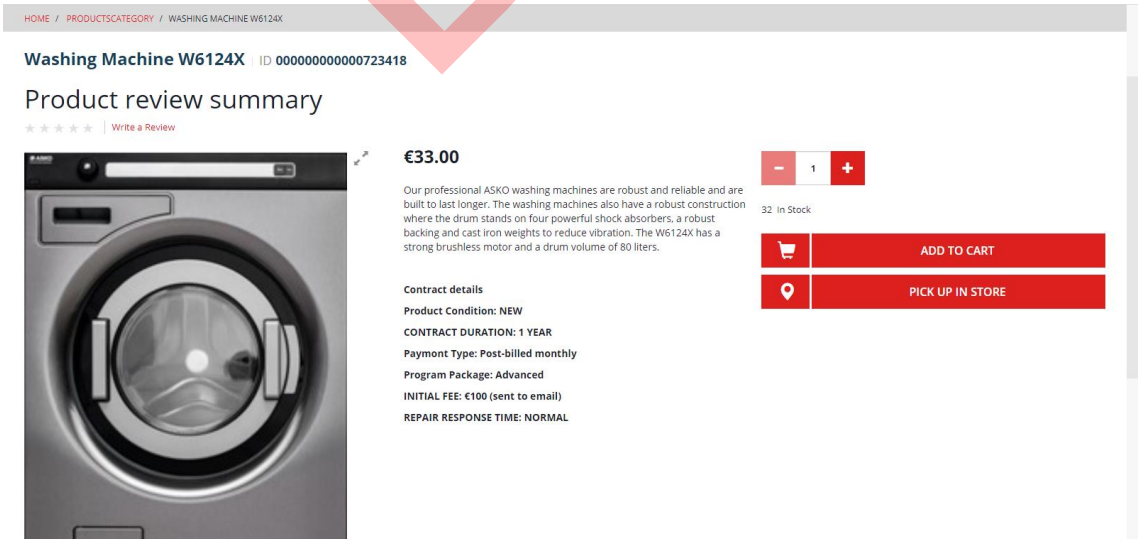

Figure 13: Product details page

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

6.1.5. Shopping cart page

In the shopping cart page the customer may see information about how many pieces have been added into the cart, the price and information about shipping and delivery. The user may choose to either continue his shopping or checking out the order. In case a wrong product was added, the user can delete the item by clicking on the three red dots on the right side of the page.

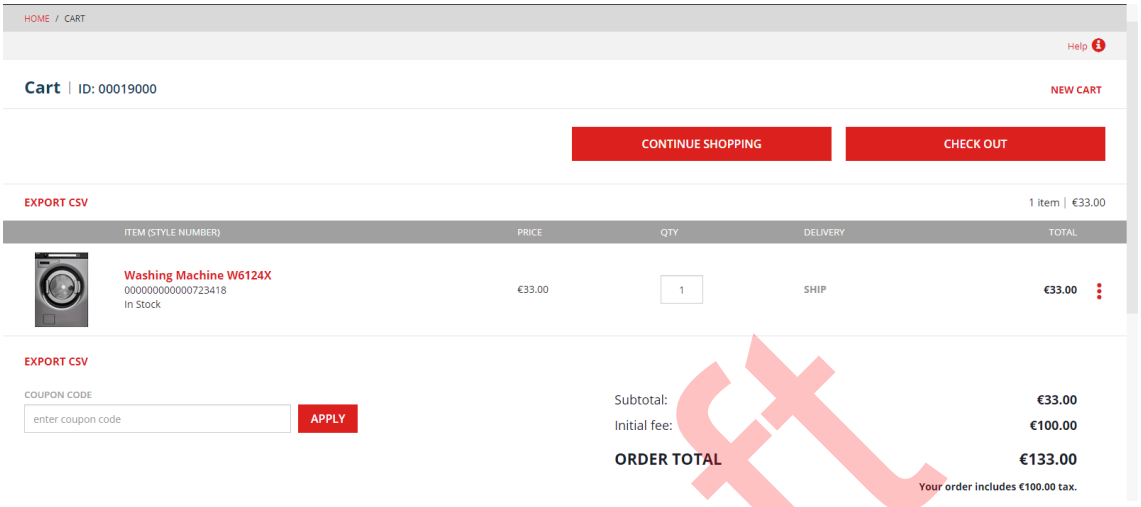

Figure 14: Shopping cart page

6.1.6. Checkout Page

After clicking the check out button, the customer is redirected to the checkout page. On the right side of the page the order summary is displayed. On the left side of the page the user is prompted to insert all the shipping, delivery, payment and billing details.

The first step is to insert the shipping details. The address can be inputted manually or chosen from a previous saved address book.

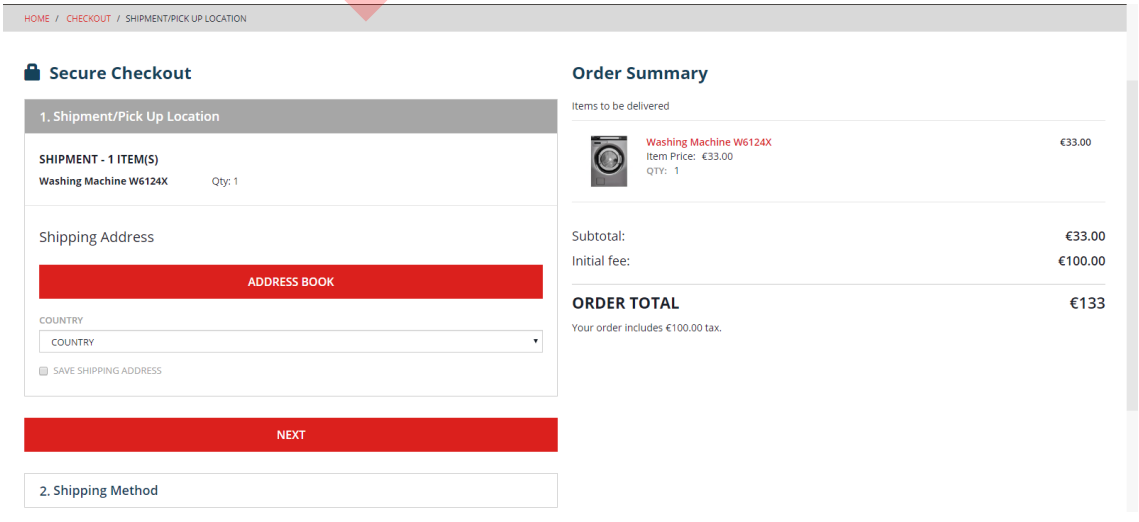

Figure 15: Checkout page

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

For inputting a new address, the country must be chosen first.

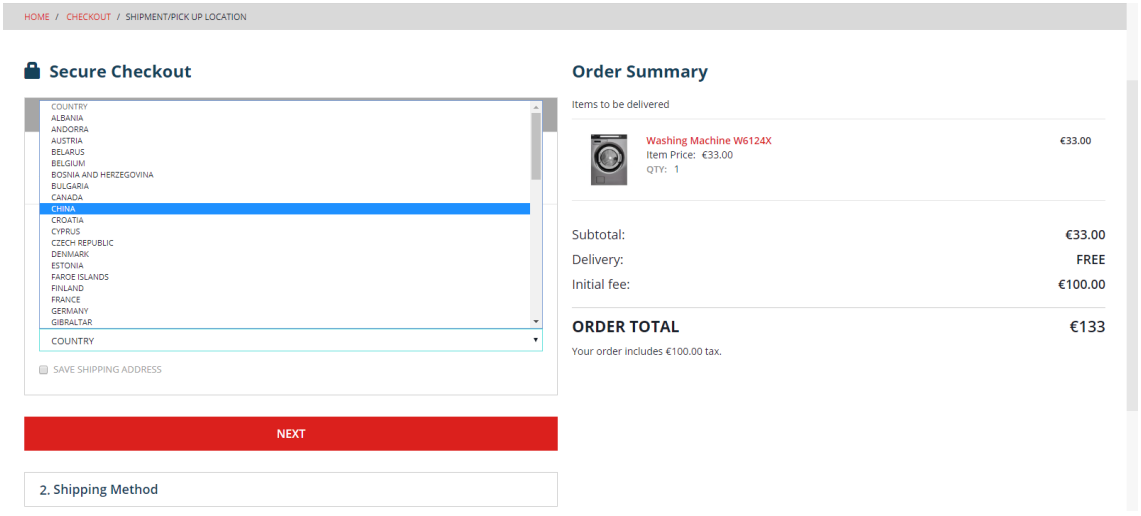

Figure 16: Shipping address country

After choosing the country, the shipping address form is displayed where the user should provide a title, first name, last name, address, city, post code and optionally the phone number.

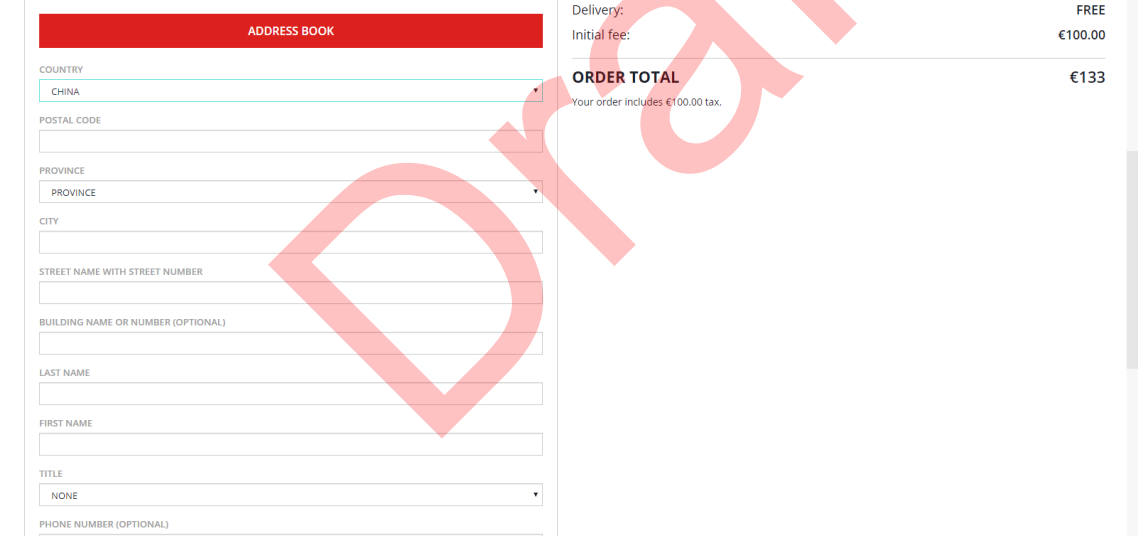

Figure 17: Shipping address form

The other option is choosing an address from the address book. A pop-up with the saved address is displayed, letting the customer choose the address he wants to use.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

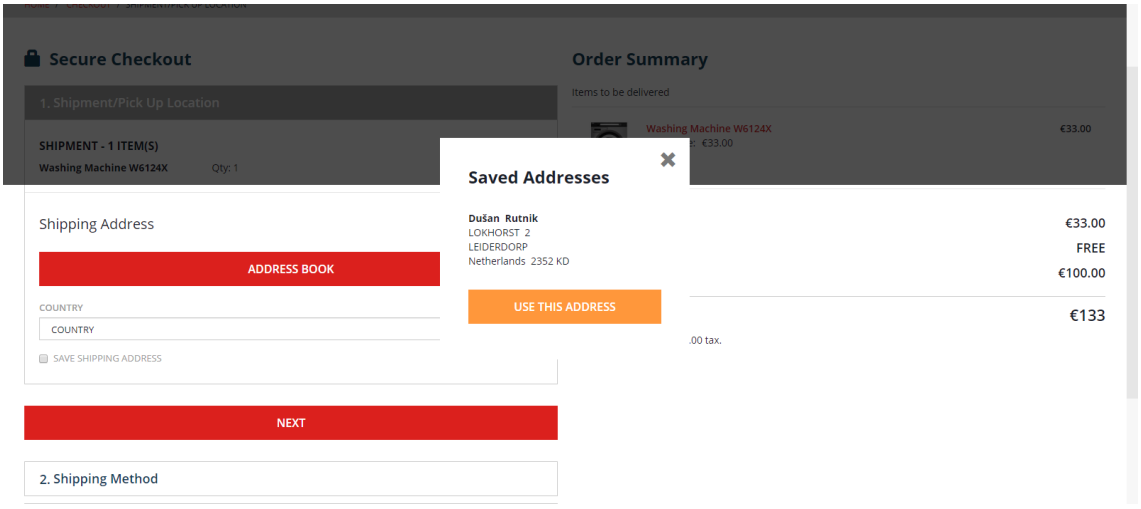

Figure 18: Address book

On the summary, the shipping address previously inputted is displayed, as well as the product. The second step is choosing the shipping method, from the dropdown displayed on the page.

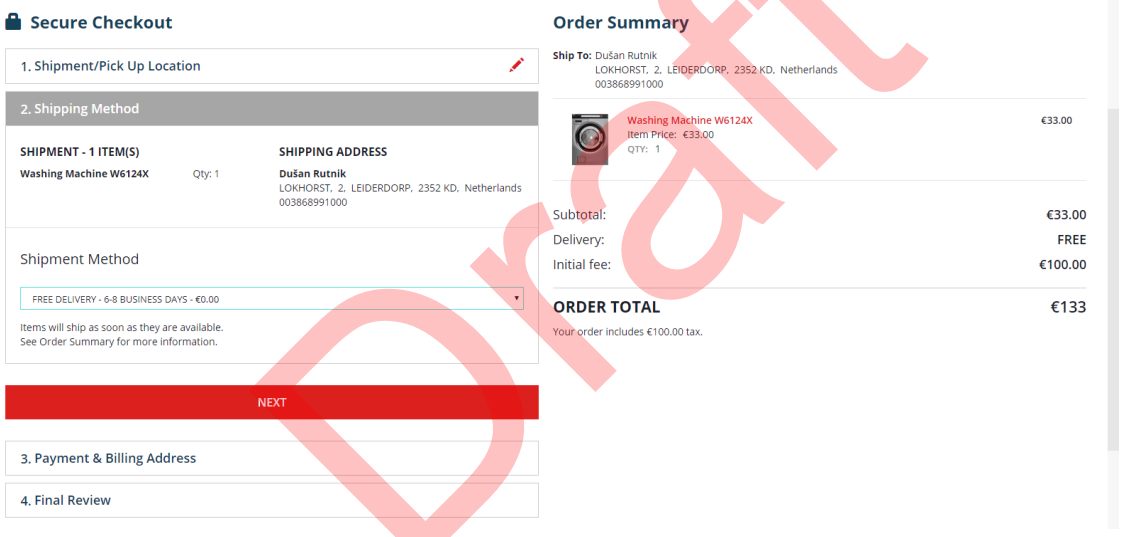

Figure 19: Shipping method

Third step consists in providing the payment method and billing address. The user may choose his card type and input all the payment data. The billing address may be provided, similarly with the shipping one. The customer may also choose to use the same address for both shipping and billing.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

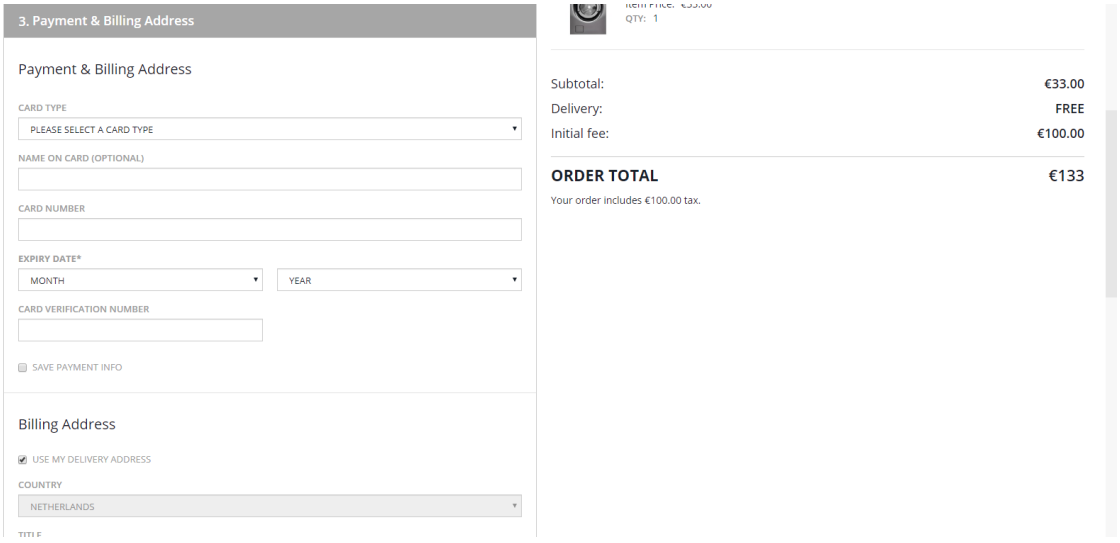

Figure 20: Payment & billing address

The last step is reviewing the order one final time. Here all the information provided is displayed: the products ordered, how many products have been ordered, their total price, shipping and billing information, initial tax, along with the final total. The initial tax consists in an installation fee that should be paid before the delivery is done.

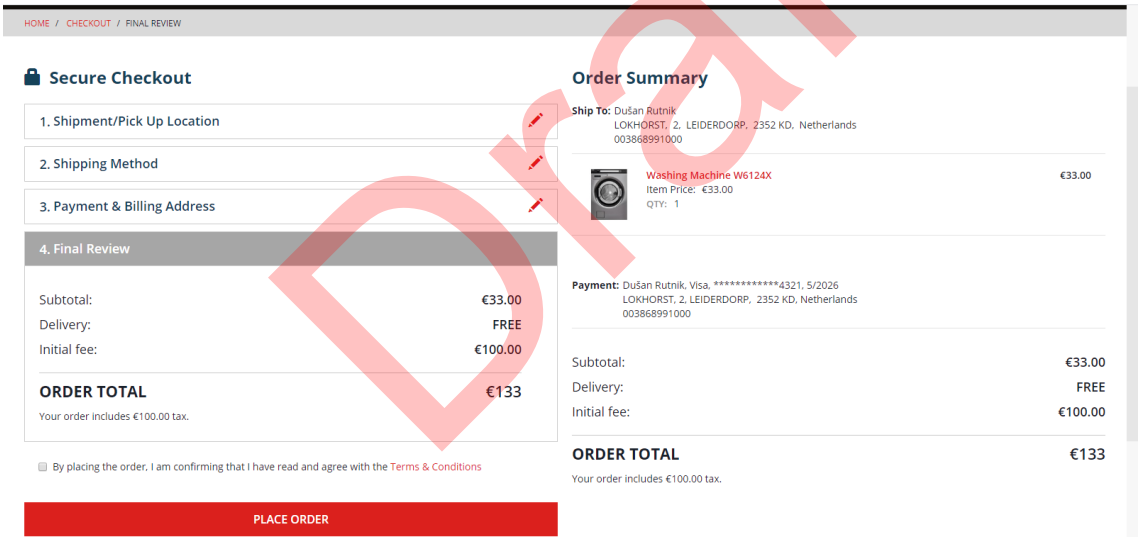

Figure 21: Final review

In order to place the order, the user must agree to the terms and conditions. After placing the order a spinner is displayed while the order is processed.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

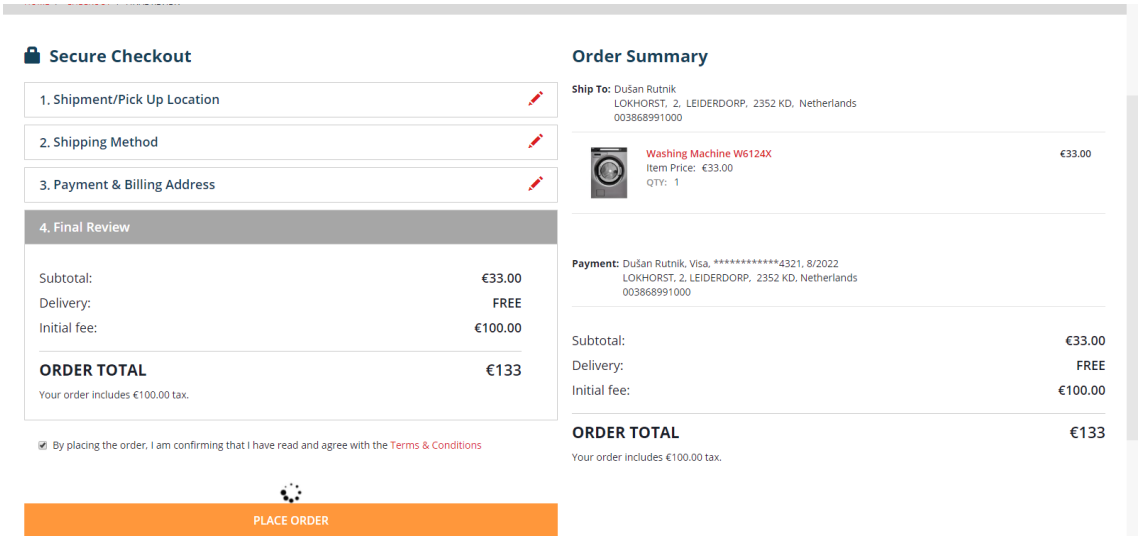

Figure 22: Processing the order

The final step of the checkout is the order confirmation. On this page a thank you message is displayed, along with the order number and a message saying that a copy of the order has been sent to the user e-mail. The summary is displayed again, confirming once more the order details. Once the order is placed and confirmed, the customer will receive two e-mails: one with the order details and one to inform him specifically about the fact that he has to pay an initial installation fee before he can have his product delivered.

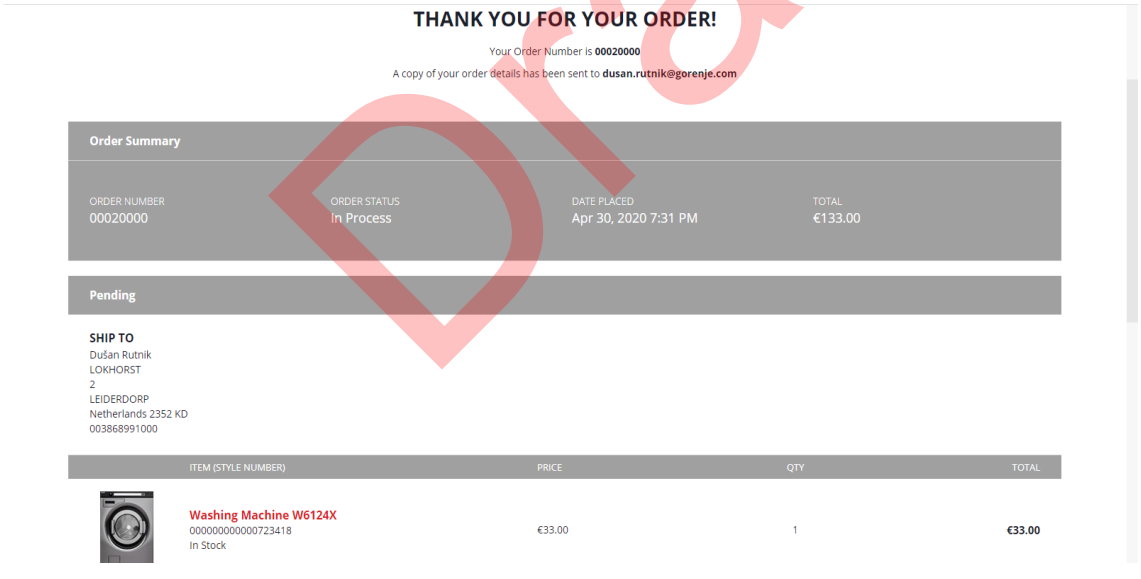

Figure 23: Order confirmation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

6.2. ReCiPSS backend server

The ReCiPSS backend service keeps track of all Pay Per Use contracts and how each contract relates to users and appliances (washing machines). For each washing machine it continuously records the usage and at the end of the month the ReCiPSS backend server calculates the Pay Per Use bill based on the usage and the cost table that is linked to the contract. The ReCiPSS backend server functionality is exposed through a REST interface. The code is built on ASP.Net core technology and data stored in an SQL Server database. The ReCiPSS backend server integrates with the WebStore, the JUConnect platform and the SAHG+ service application.

6.2.1. ReCiPSS backend server information model

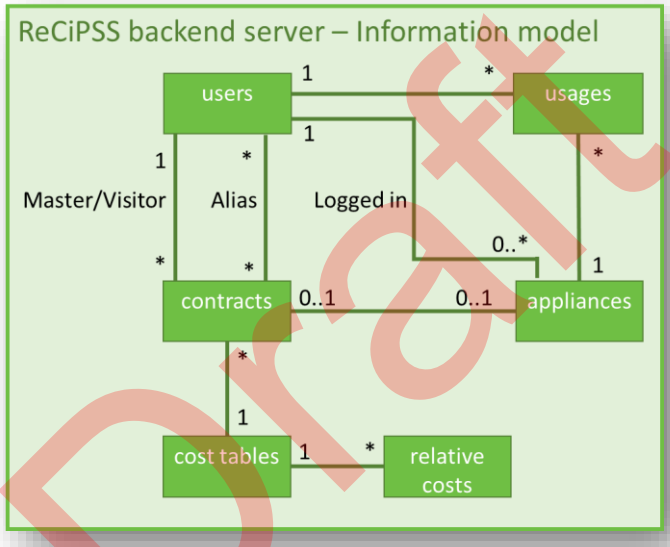

Figure 24: ReCiPSS backend server information model

Figure 24 illustrates the information model of the ReCiPSS backend server. The main element is the contract (Pay Per Use contract), which can either be a Master or a Visitor contract. Both types of contracts are linked to a user (the signee) that will be billed for the usage on the contract. A Master contract will be linked to a washing machine when it is delivered. By default, all usage of the machine will be billed to the Master's contract unless a Visitor is logged in into the machine, then the usage is billed on the Visitor's contract. Alias users do not have their own contract, but a Master can allow an Alias user to use the appliance and then the usage will be billed on the Master's contract. The initial delivery of the Whitegoods ICT platform in M24 will only implement the Master user. Visitors and Alias users will be added to the platform during the duration of the demonstrator. At the end of the month, the bill for each contract will be calculated based on the usage and a cost table with fixed prices (price per minute e.g.) and relative prices based on the use of special features (prewash e.g.).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

6.2.2. ReCiPSS backend server REST Interface

Table 1 lists the main methods exposed by the REST interface. Some less important methods for maintenance and administration have been left out (e.g. basic CRUD methods).

Method name	Purpose	Request	Response
POST /contracts/create	Create a PPU contract, SAP contract and delivery order	Customer Id Article number Address (optional)	SAP contract Id
POST /contracts/linkAppliance POST /contracts/unLinkAppliance	Link and unlink a washing machine to a contract	SAP Contract Id AuId Delivery date	SAP contract Id AuId
GET /appliances/	Retrieve appliance configuration (incl. PPU flag)	AuId	Appliance fields
POST /usages/add	Push usage data to ReCiPSS backend server	AuId ProgramId Wash parameters	Usage Id
POST /usages/addByAuId	Pull usage data from appliance	AuId	ProgramId Wash parameters
GET /contracts/getSumOfChargedCosts	Return the pay per use costs for a given time period	AuId StartDate, EndDate	Cost

Table 1: Main methods of ReCiPSS Backend Server REST interface

6.3. SAP and SAG

6.3.1. Gorenje backend system connector (Gorenje SAP PI)

Gorenje SAP PI takes on the role of an Enterprise Service Bus. An enterprise service bus (ESB) is implementing a communication system between mutually interacting software applications in a service-oriented architecture (SOA).

6.3.2. Gorenje SAP Commerce Cloud (Hybris) system

Gorenje Hybris takes on the role of Customer relationship management (CRM) and a content management system (CMS).

CRM is an approach to managing a company's interaction with current and potential future customers which tries to analyze data about customers' history with a company, to improve business relationships with customers, specifically focusing on customer retention, and ultimately to drive sales growth.

CMS is a computer application that supports the creation and modification of digital content.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

6.3.3. Service Application Gorenje (SAG) system

The main idea of the SAG web application is to combine in one application all the functionality that needs a Gorenje service network in one country for its normal work.

SAG is a system that is strongly linked to SAP since all master data comes from SAP – via interfaces:

- Material data
- Spare parts data
- Pricelist
- Customers (partners in SAP)

Documents from SAG are sent to SAP

- Material purchase order
- Sales order
- Material movement on storage location
- CS orders
- Etc.

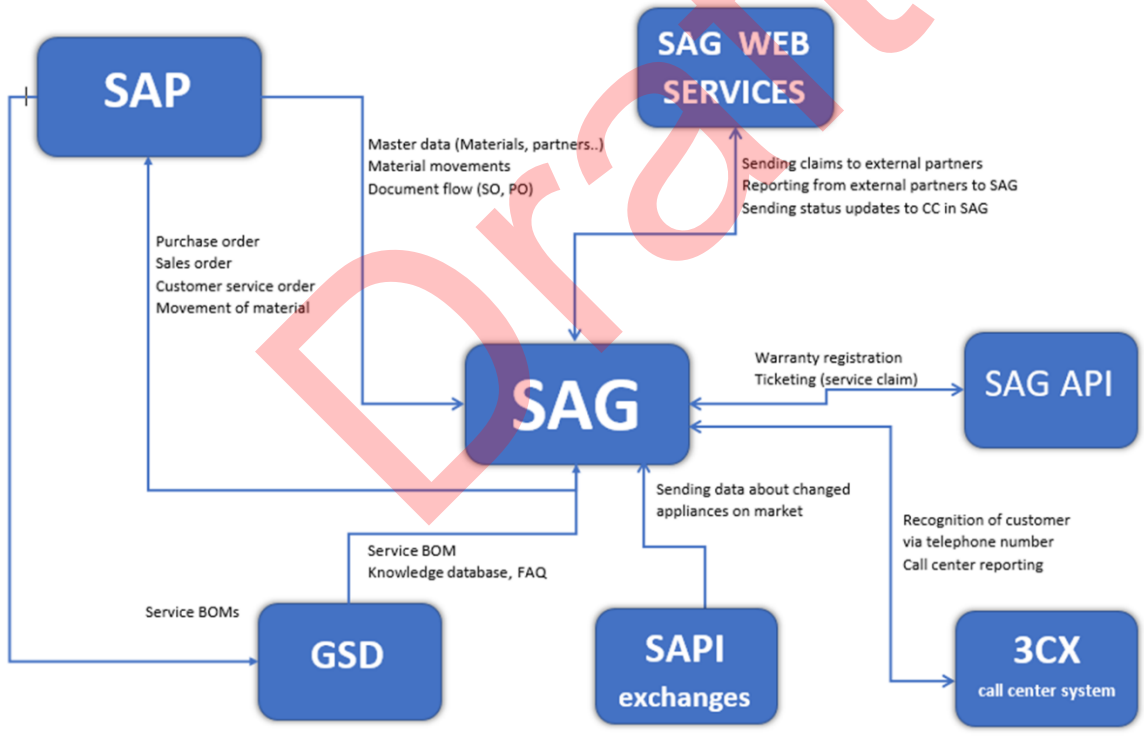

Figure 25: SAG Connections

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

6.4. JUConnect platform

JuConnect is a cloud platform that is used for communication between appliances, ASKO Professional Laundry applications and Gorenje Backend Systems. The goal of the JuConnect is to connect the smart home industry chain by establishing an open ecosystem for IOT devices. It supports all kinds of hardware interfaces and has already integrated more than 40 manufacturers with 90 categories and 300 different models of smart home products. The main functionality of the platform is to provide access to the IOT device, monitor the status of the device, support firmware upgrades and to offer the device fault detection through the analysis of the big data. The data from WMs sensors are sent to the cloud and saved there.

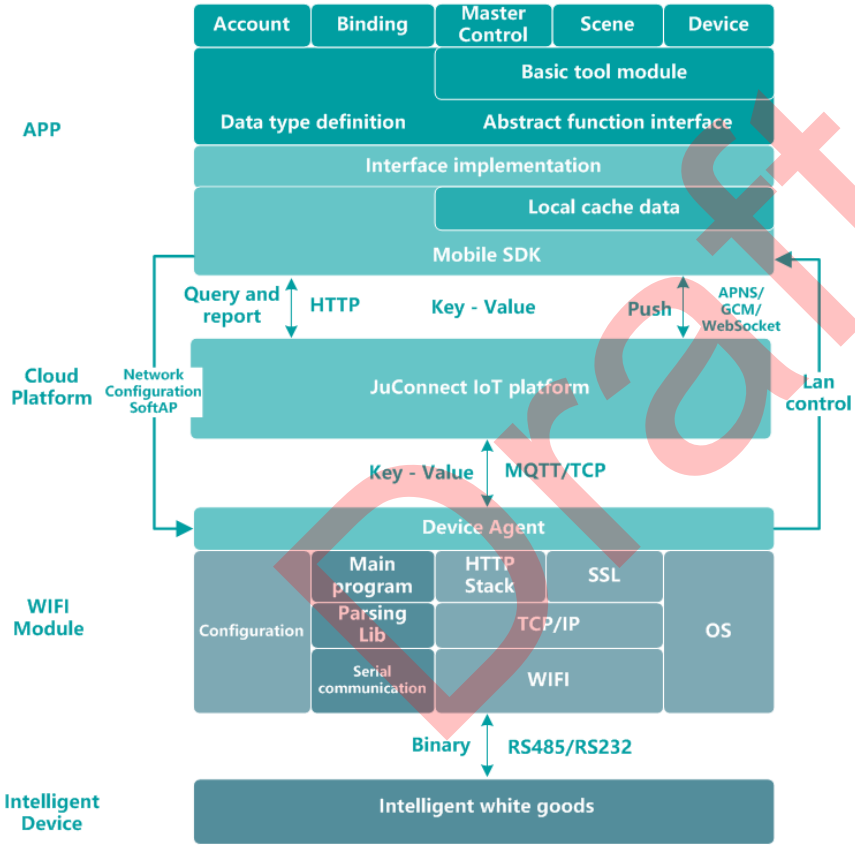

Figure 26: Overall structure of the JuConnect Overseas Cloud Platform

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

6.5. ConnectLife API

Secure façade for public access for information otherwise stored on the backend. All applications (Hisense, Gorenje and 3rd party) can connect only to ConnectLife API in order to communicate with the system. ConnectLife API provides flexible and secure connection point (through OAuth 2.0) for all types of client applications.

6.6. Demo Appliance – ASKO Washing machine

ASKO washing machines are ideal where domestic appliances are not powerful enough or do not fulfill prevailing hygiene requirements. They consist of two main categories:

- ASKO PRO Washing Machines
- ASKO PRO Tumble Dryers

They are robust machines that are built to be used many times a day. They all have a long durability, are easy to use and are equipped with special and shorter programs to fulfil the demands of the customers. They provide the interfaces and SW support to all major producers of the external booking and payment systems.

In the ReCiPSS Whitegoods demonstrator only the Washing machine will be deployed.

To support the pay per use model we will use the washing machine with integrated Wi-Fi module. We are planning to use this appliance primarily in the shared spaces (basements, professional laundry saloons, etc), typically with more than ten appliances of the same kind in the same room.

Typically, there will be a lot of different users using the same machine. But they can also be used in a private home where only one person/family has access to the appliance.

The appliance is connected to the internet via WIFI module through a WIFI access point. Which can be a router (or a similar device) or a hotspot on the user's mobile phone.

The connection settings are done by the person who has access to the service menu.
Typically, this is an owner of the appliance in a private business (dentist, hairdresser), owner of the appliance in the laundry saloon; it might be a service provider for a whole neighborhood or a private person using the appliance at home.

Connected machine support:

- Remote access, diagnostics and service,
- Auto registration,
- Remote control & monitoring of the appliance,
- Start/stop the program cycle (with/without delay),
- See what program is running, remaining cycle time, get ready message,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

- Push notifications (get failure messages),
- Safety functions (children lock, open door),
- Synchronized settings (ex. water hardness, auto-dosing),
- Diagnostics (number of cycles, programs, consumption values),
- Get service & periodic maintenance messages,
- User wizard (access to user care guides, tutorial, online advice troubleshooting,
- Possibility to download customized programs via WiFi,
- Reporting of the ReCiPSS parameters to the cloud

6.7. Cloud appliance data

The User can access the appliance data on the cloud with two interfaces:

1. Web application
2. Mobile application

Web application is intended for the users who also have access to the service menu on the appliance. In the web application will be all the data about the machine that an owner of more washing machines needs. It provides the overall status of the device, product information and possibility to call a repair service. Additionally, the statistics of most used programs will also be gathered in the web application.

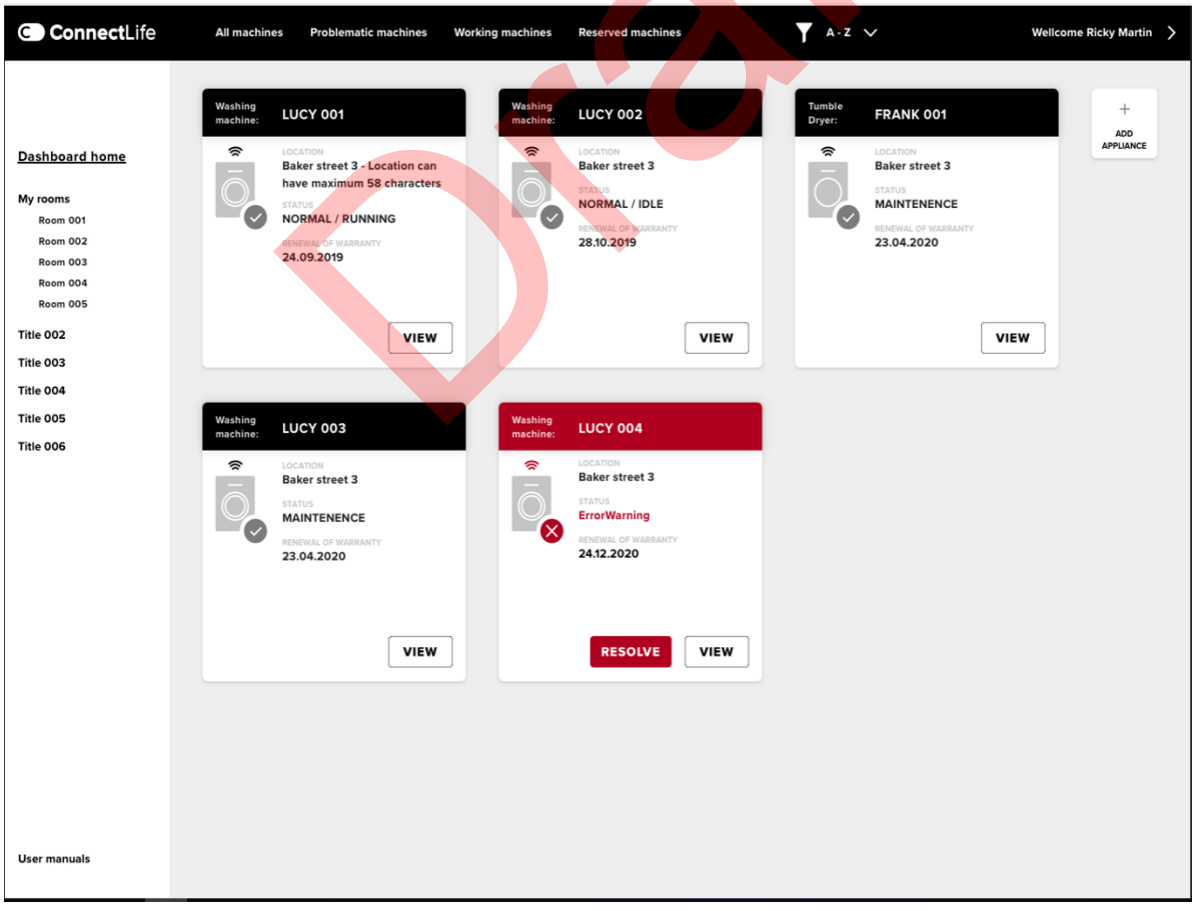

Figure 27: Cloud web application

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

obile application is intended for use by end customers in common laundry rooms and for home and small businesses users. Users can install the application through public App Store (Apple Store or Google Play). The application enables a user to easily identify the location of the nearest laundry room, and if the room is bookable the user can make a reservation for laundry room and check availability. When in Laundry user can connect to selected laundry appliance through QR code visible on the appliance and then she/he can remotely monitor the operation of the appliance.

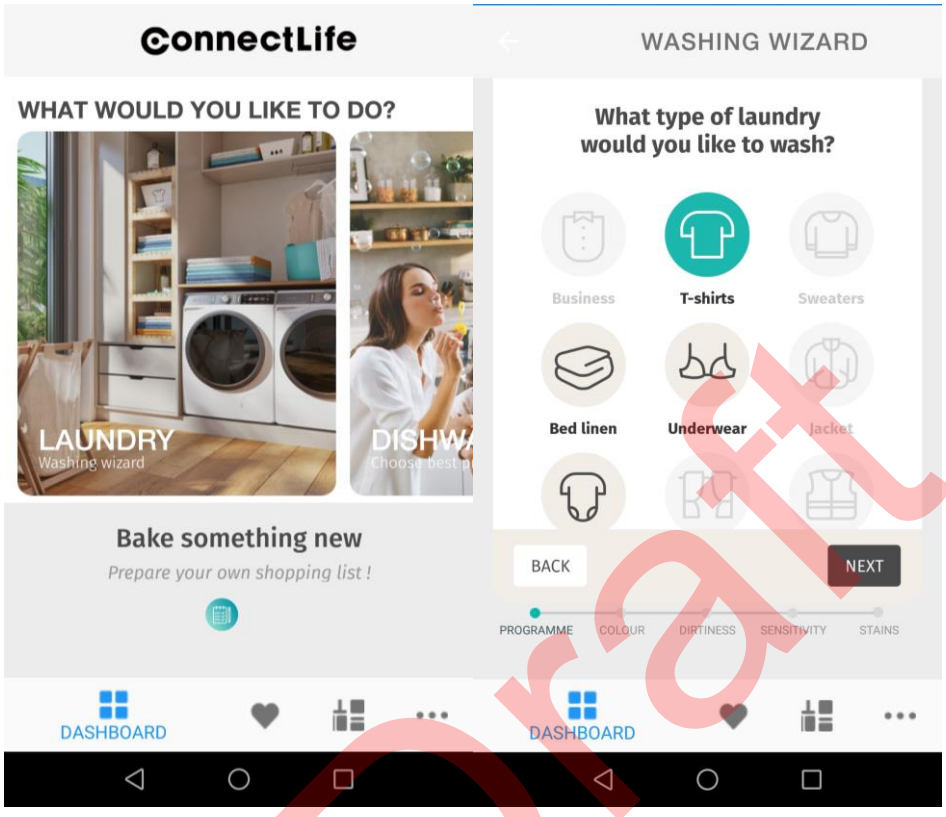

Figure 28: Cloud mobile application

6.8. Service instructions – Augmented reality

This application is aimed at supporting Service, Maintenance and Training with visual Augmented Reality (AR) instructions that can be consumed internally and externally. But, in order to meet certain enterprise needs to enable the success of visual-data strategies such as AR, several constraints and limitations must be overcome. Some of these are listed here:

6.8.1. Lack of re-use of existing CAD assets where available

In many cases, 3D CAD models are available (either as assemblies, individual parts, or configurations of sub-assemblies/parts) as engineering assets. However, these are often not re-purposed for downstream enterprise usage due to:

- Concern about Intellectual Property leakage
- Lack of access to CAD data by non-engineering staff
- Size and complexity of native CAD data

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

- Heterogeneous (*multi-CAD*) assemblies that don't 'exist' within a single CAD environment
- A lack of process to automate the conversion from heavyweight CAD to lightweight (and ultra-lightweight) 3D derivatives

6.8.2. Inability to generate derivative 3D data that meets size and performance goals

For mobile-led usage of 3D data, care must be taken to ensure that data is as small and simple (but still visually 'good enough') as possible. Smaller data is able to be transported more quickly (and therefore be usable on lower bandwidth connections), requires less memory on the target device, is able to be displayed more quickly and therefore provide a more pleasing and interactive experience.

However, the process to do this is not well understood by many within the industry.

6.8.3. Inability to automate at enterprise-scale

Ideally, a company should have, at its disposal, an automation pipeline that can perform Automated Content Transformation from the canonical 3D CAD content (following a principal rule, typically stored within a PLM system) all the way through to delivery of applications and experiences which harness ultra-lightweight 3D content.

Such automation is achievable but is typically lacking in many enterprises.

As a consequence of these *perceived* barriers to adoption, many companies wishing to harness the power of 3D technologies result in pursuing manual and therefore *costly* and *time-consuming* techniques to re-model 3D content, rather than harnessing the availability of existing 3D CAD assets.

Moreover, without addressing these challenges, it becomes difficult to develop a longer-term strategic vision with a high level of confidence, often resulting in difficulties in securing funding from management.

6.8.4. Initial AR demonstrator

The first (current) AR demonstrator is manually prepared through the following steps:

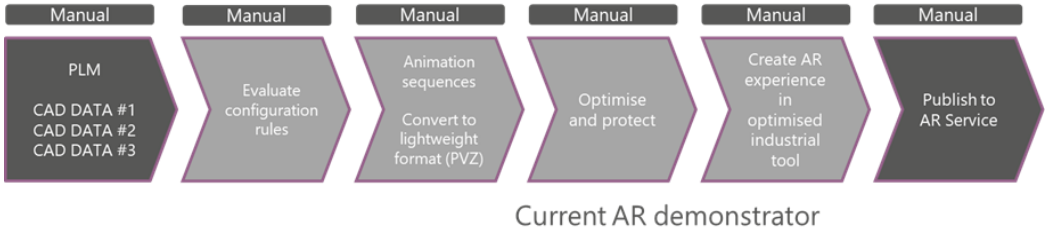

Figure 29: Current AR Demonstrator

The following screen captures (from Android mobile) shows the current AR demonstrator indicating optimized and protected 3D CAD data displayed within the real environment at scale. The 3D CAD data has animation sequences connected displaying removal of the lock and

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

removal of the hinge. The UI and functions can be treated as 'templates'. The developed functions can be re-used for further AR experiences and as base for further development.

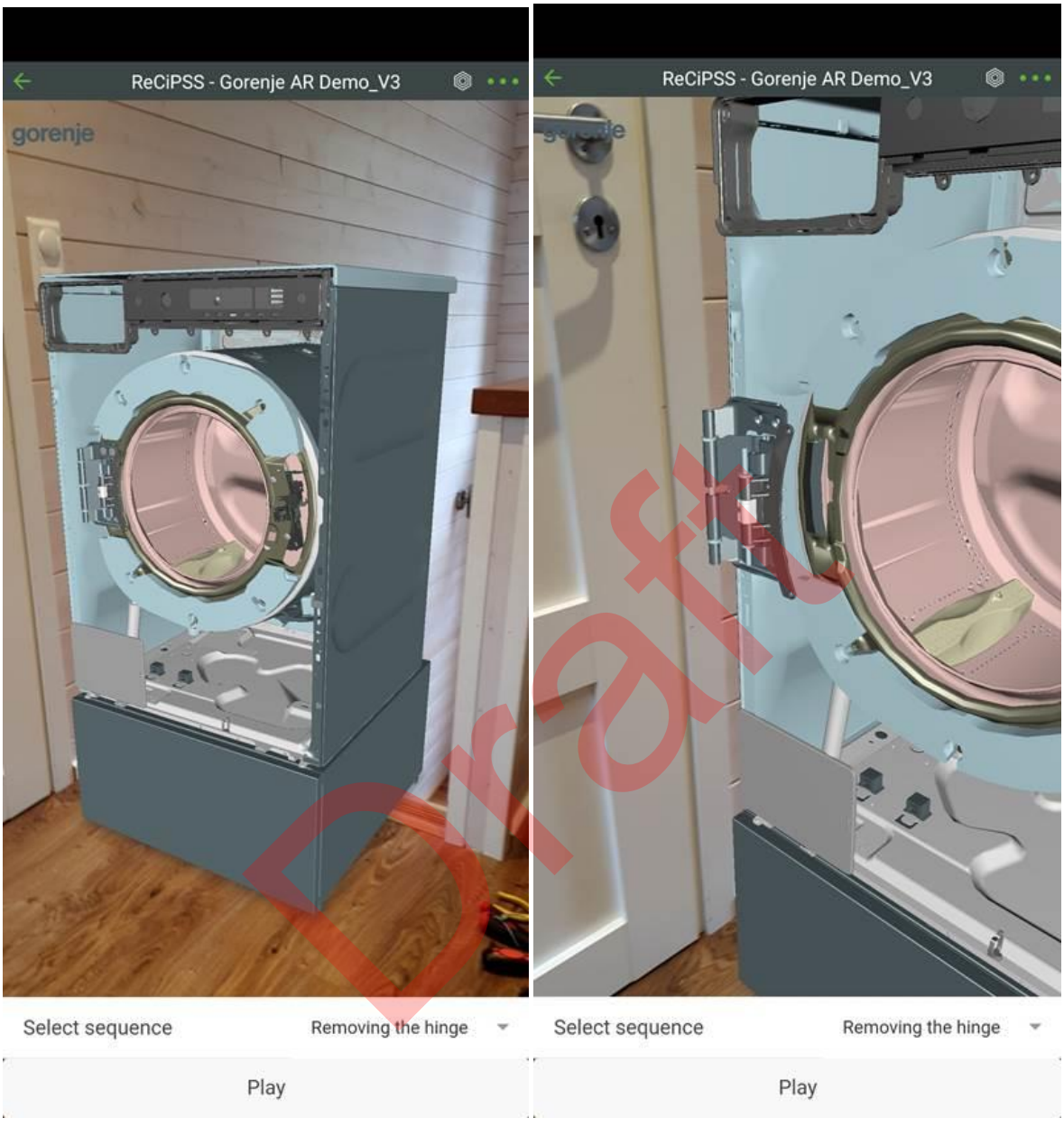

Figure 30: Android captures

Further steps could be considered in order to meet the industrial need of cost and scalability according to the following illustrations (Phase 1, 2, 3):

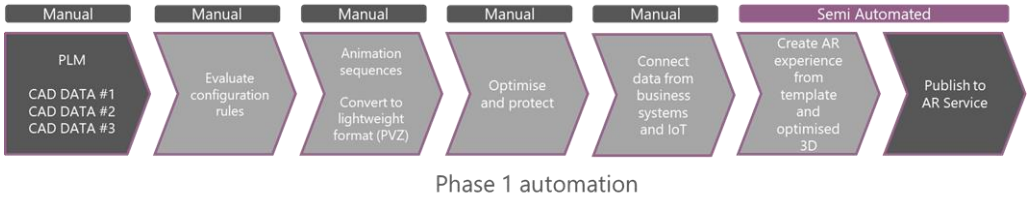

Figure 31: Phase 1 automation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

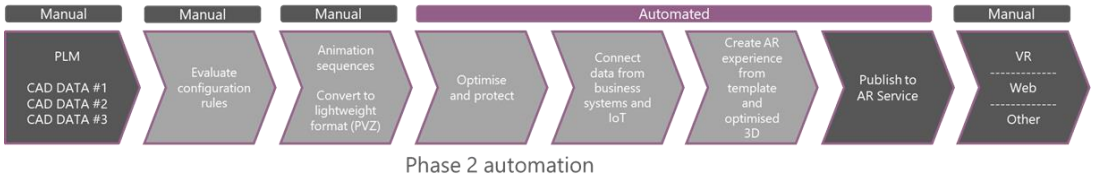

Figure 32: Phase 2 automation

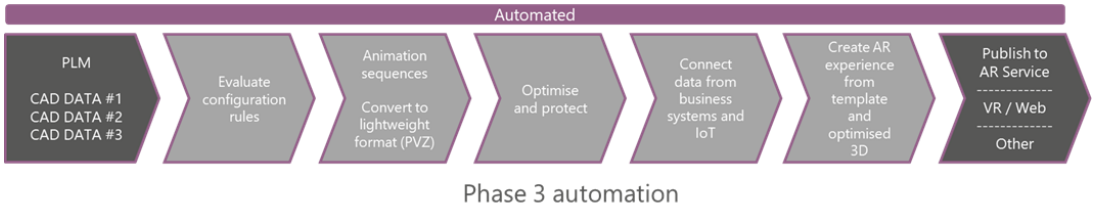

Figure 33: Phase 3 automation

Figures 31, 32 and 33 above describe the conceptual maturity model of AR automation connected to product and process master data. In order to meet the industrial needs, the process needs to be automated as much as possible including all steps described.

PLM and CAD data

This is product and lifecycle data of products that have been released and are in use.

Evaluate configuration use

Products are configured and maintained; the system needs to know the products current state in order to provide the correct visual appearance and instructions.

Animation sequences

In order to achieve visual instructions based on product CAD data, animation sequences need to be in place. In this case Creo Illustrate is used from where the PVZ data is achieved containing the following data sets: Product structure and hierarchy (PVS), Geometry (OL), Product meta-data (PVS), Sequences and animations (PVI). These data sets will drive the visual appearance and automation further downstream.

Optimise and protect

In order for data to be compatible for the use case and for IP to be removed, data optimisation needs to be managed. Manual optimisation and re-modelling for visualisation purposes is not a viable method. Solutions for automatically performing this step needs to be in place to allow for scalability and process control.

Connect data from business systems and IoT + Create AR experience + Publish the data

The visual product representation, from PLM, reflects the products current state and need to be enhanced with data from other business systems like ERP, MES, CRM and others. Use case driven templates will define how data is connected and from where, to drive relevant user-controlled experiences and make them available for consumption.

Current state and maturity

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

A detailed architectural maturity model describing current toolchain and data formats used will be worked out based on the findings and ddiscussions with the participating companies. This will include the bigger picture and business system landscape in order to drive and direct further developments. The architecture needs to be flexible enough to support the import and re-use of data, using converters and tools from various software vendors.

A solution based on the architecture maturity model will be piloted according to the steps described in Figure 31 – 33.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

7. Testing

7.1. Test strategy

The test strategy is illustrated in Figure 34: and is strongly related to the requirement strategy. The business requirements are formulated as business use cases. Based on the use cases an overall architecture was created that divides the functionality into components: the web store, the backend server, the appliance, the cloud platform and the existing backend systems. For each component, the responsible technology provider has developed requirements and design using their own processes. Interface specifications have been developed to ensure the operability between the components. Testing follows the same hierarchy:

- FT and CT: Functionality and components are tested in their development environment using Functional and Component Tests. Responsible: component developers.
- SIT: Interoperability between the components is tested in the testing environment against the functional and non-functional requirements of each component. Responsible: component owners (SIG, SIV, PDS, GOR).
- UAT: Compliance with the business requirements is tested in the test environment against the business use cases. Responsible: product owner (GOR).

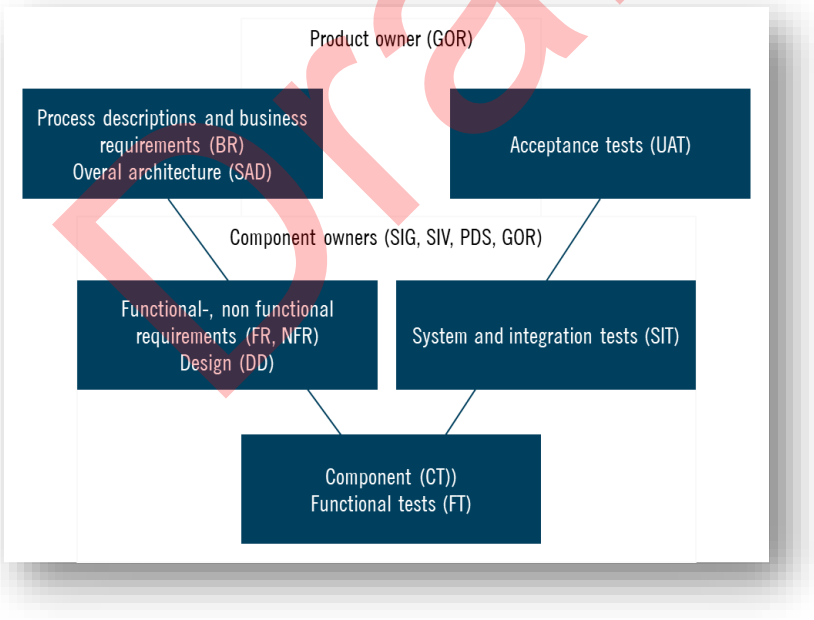

Figure 34: Whitegoods ICT platform test strategy

7.2. Test planning

Due to the agile development methodology Functional, Component and Integration tests are being done continuously during the development of the components. :

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

The testing will continue as the development will continue. For the first version of the platform the following steps were planned:

1. All functional and component tests ready for version 1 of the Whitegoods ICT platform
2. System integration tests (two or three rounds)
   Bug fixes and retest.
   Setup of deployment in the production environment.
3. Acceptance tests (two rounds)
   Bug fixes and retest.
4. Deployment in the Production environment
   Final tests.

The test plan for each component will follow similar paths:

1. Test preparation - Preparation of test specifications
2. Test configuration - Configuring the test environment
3. Test execution - Execution of tests and detailing of non-conformities
4. Test evaluation - Structuring and analyzing test results.

7.2.1. Functional tests ReCiPSS Backend Server

Error! Reference source not found. contains a list of the methods that have been tested functionally. During the functional tests, all these methods have been tested under the following circumstances:

1. Correctly formatted data with correct values (GET, POST, UPDATE)
2. Correctly formatted data with incorrect values (GET, POST, UPDATE)
3. Correctly formatted data with missing mandatory values (GET, POST, UPDATE)
4. Incorrectly formatted data (GET, POST, UPDATE)
5. Searching for items that do exist (GET, UPDATE, DELETE)
6. Searching for items that do not exist (GET, UPDATE, DELETE)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

Figure 35: ReCiPSS Backend Server tested methods

7.2.2. Component test ReCiPSS Backend Server

After the functional tests for each method were approved, the ReCiPSS Backend Server has been tested as a component. **Error! Reference source not found.** shows the setup for the component tests. The WebStore requests and responses were provided by the testing tool SoapUI. The appliance information was simulated by a test method in the ReCiPSS backend server. The SAG Test system could be used during the component test, so these requests and responses did not need to be simulated. The following use cases were tested:

1. WebStore creates a PPU contract; contract is created in ReCiPSS backend server and in SAP. Delivery order is created in SAG.
2. Appliance is delivered; contract is linked to an appliance
3. Appliance is replaced; contract is unlined from old appliance and linked to new appliance
4. Appliance is removed or contract is terminated; appliance is removed from contract.

Monthly bill is created; usage data is retrieved from the appliance and cost for each wash is calculated; total cost for the month is returned.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

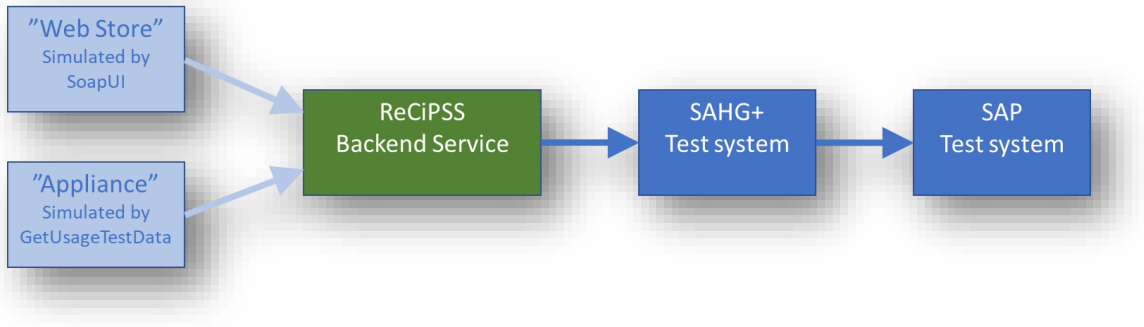

Figure 36: ReCiPSS Backend Server component test architecture

8. Conclusions

IoT platform that supports smart washing machines represents the most critical part and one of the most important steps of ReCiPSS project implementation for the white goods demonstrator.

As seen in the previous chapters, the IT platform is very complex and involves coordination and contribution of different IT specialists and experts from the legal department, accounting, tax department, R&D, logistic, repair, international trade and GDPR. Furthermore, this development also requires coordination within Gorenje services, sales and business units in countries where the white goods demonstrator will be implemented.

The complexity, technical obstacles and difficulties that we could not foresee at the preparation phase may result in a slight deviation of the final IoT platform from the described one.

The consortium will put out most effort to fulfil the set development task within the time and content frame.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2

9. References

[1] https://en.wikipedia.org/wiki/Rapid_application_development

[2] https://en.wikipedia.org/wiki/Agile_software_development

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776577-2