

EFPF: European Connected Factory Platform for Agile Manufacturing



European Factory
Platform

WP5: EFPF Add-ons

D5.13: EFPF Interfacing, Evolution and Extension Vs: 1.0

Deliverable Lead and Editor: Norman Wessel, ASC

Contributing Partners: ASC, CETH, C2K, FIT, ICE, ISMB, SRDC, SRFG, VLC

Date: 2020-06-30

Dissemination: Public

Status: <Draft | Consortium Approved | EU Approved>

Short Abstract

This deliverable describes the activities that took place in three interrelated tasks during the first 18 months of the EFPF project. These tasks relate to Ecosystem, Evolution and Extension Requirements; Integrated Marketplace and the EFPF Portal. The deliverable presents key rationale of these tasks, along with the overview of the technical progress and the current status.

Grant Agreement:
825075



Document Status

Deliverable Lead	Norman Wessel, ASC
Internal Reviewer 1	Simon Osborne, C2K
Internal Reviewer 2	Martin Lorenz, ASI
Type	Deliverable
Work Package	WP2: Requirements Elicitation and Pilot Scenarios WP5: Add-ons
ID	D5.13: Interfacing, Evolution and Extension
Due Date	2020-06-30
Delivery Date	2020-06-30-30
Status	<Draft Consortium Approved EU Approved>

History

See Annex A.

Status

This deliverable is subject to final acceptance by the European Commission.

Further Information

www.efpf.org

Disclaimer

The views represented in this document only reflect the views of the authors and not the views of the European Union. The European Union is not liable for any use that may be made of the information contained in this document.

Furthermore, the information is provided “as is” and no guarantee or warranty is given that the information is fit for any particular purpose. The user of the information uses it at its sole risk and liability.

Project Partners:



Executive Summary

This deliverable describes the activities that took place in the following three interrelated tasks during the first 18 months of the EFPF project.

T2.5: Ecosystem, Evolution and Extension Requirements

T3.3: Integrated Marketplace Framework and Realisation

T5.2: EFPF Portal

These tasks correspond to the thematic topic of EFPF Interfacing, Evolution and Extension. The deliverable presents key rational of these tasks, along with the overview of the technical progress and the current status.

The EFPF federation has established an ecosystem of multiple stakeholders, where connectivity between different tools, platforms and stakeholders has been established through the Data Spine in the first 18 months of the project. This federated ecosystem is expected to grow and evolve overtime with the inclusion of new entrants (such as those coming from the open funding call). This deliverable lays out the initial requirements that allow the project partners to plan ahead and put in place adequate means (technologies, processes, protocols etc) that can support the evolution and extension of the federation; and underlying mechanisms.

The EFPF Marketplace framework provides an integration mechanism that allows the representation of multiple marketplace offerings through a unified interface. The framework has been enhanced to support connections to additional external marketplaces and with an automated agent-based marketplace, which supports online bidding. Additionally, to support the business case of the EFPF platform, an Accountancy Service provides the functionality of event logging and invoice data retrieval and generation from connected marketplaces.

The EFPF Portal has been set up early in the project to enable tool, marketplace, and platform providers to make first connections to the EFPF platform. Since then the EFPF Portal has grown in terms of tools, platforms and required functions like registration.

In this respect, this deliverable describes the different interface (such as marketplace and portal) offered by the EFPF platform and the reports on the ongoing work that is being carried out to make EFPF an open and adaptive ecosystem.

Table of Contents

0	Introduction	6
1	Ecosystem, Evolution and Extensions Requirements	8
1.1	Requirements for a Federated Ecosystem	8
1.2	Requirements for the Evolution of a Federated Ecosystem	9
1.3	Requirements for the Extension of a Federated Ecosystem	10
1.4	Governance for the Growth of the EFPF Platform Ecosystem	11
2	EFPF Marketplace	14
2.1	Integrated Marketplace Framework	14
2.1.1	Scope and Relationship with Other EFPF Components	15
2.1.2	Requirements and Realisation	19
2.1.3	Deployment	19
2.1.4	Execution and Usage	20
2.1.5	Limitations and Further Development	20
2.2	Automated Agent-based Marketplace and Online Bidding	21
2.2.1	Scope and Relationship with Other EFPF Components	21
2.2.2	Requirements and Realisation	22
2.2.3	Deployment	24
2.2.4	Execution and Usage	24
2.2.5	Limitations and Further Development	28
2.3	Accountancy Service	28
2.3.1	Scope and Relationship with Other EFPF Components	28
2.3.2	Requirements and Realisation	29
2.3.3	Deployment	30
2.3.4	Execution and Usage	30
2.3.5	Limitations and Further Development	33
3	EFPF Portal	34
3.1	Current Status	34
3.1.1	Register	35
3.1.2	Login	35
3.1.3	Dashboard	36
3.1.4	Value Propositions	37
3.1.5	Federated Search	38
3.2	Scope and Relationship with Other Components	39
3.3	Requirements and Realisation	40
3.4	Deployment in the EFPF Platform (Installation)	41
3.5	Execution and Usage	41
3.6	Limitations and Further Developments	41
4	Conclusion and Outlook	43
	Annex A: History	44
	Annex B: References	44
	Annex C: Data Models and Interfaces	44

0 Introduction

0.1 EFPF Project Overview

EFPF - European Connected Factory Platform for Agile Manufacturing - is a project funded by the H2020 Framework Programme of the European Commission under Grant Agreement 825075 and conducted from January 2019 until December 2022. It engages 30 partners (Users, Technology Providers, Consultants and Research Institutes) from 11 countries with a total budget of circa 16M€. Further information: efpf.org

In order to foster the growth of a pan-European platform ecosystem that enables the transition from “analogue-first” mass production, to “digital twins” and lot-size-one manufacturing, the EFPF project will design, build and operate a federated digital manufacturing platform. The Platform will be bootstrapped by interlinking the four base platforms from FoF-11-2016 cluster funded by the European Commission, early on. This will set the foundation for the development of EFPF Data Spine and the associated toolsets to fully connect the existing platforms, toolsets and user communities of the 4 base platforms. The federated EFPF platform will also be offered to new users through a unified Portal with value-added features such as single sign-on (SSO), user access management functionalities to hide the complexity of dealing with different platform and solution providers.

0.2 Deliverable Purpose and Scope

The purpose of this document is to provide an iteration of the requirements about the ecosystem creation, evolution and extensions of the EFPF platform (T2.5), and to provide a technical overview of the EFPF components Marketplace (T3.3.) and Portal (T5.2). As this deliverable is the first of two iterations, the provided content is subject to change at the last iteration.

0.3 Target Audience

This deliverable aims primarily at external platform and marketplace providers that are interested in offering their services and tools through the EFPF platform.

0.4 Deliverable Context

This document provides information to multiple software components of the EFPF platform. Its relationship to other documents is as follows:

- **D2.4: EFPF Platform Requirements:** Provides detailed information about the EFPF platform and its requirements.
- **D5.12: EFPF Security and Governance:** Provides detailed information about security, privacy, and governance in the context of EFPF ecosystem

0.5 Document Structure

This deliverable is broken down into the following sections:

- **Section 1: Ecosystem, Evolution and Extensions Requirements:** Provides information about the requirements needed for ecosystem creation, evolution and extension of the EFPF platform.
- **Section 2: EFPF Marketplace:** Provides information about the software component produced by Task 3.3: Integrated Marketplace Framework and Realisation.
- **Section 3: EFPF Portal:** Provides information about the software component produced by Task 5.2: EFPF Portal.
- **Section 4: Conclusion and Outlook:** Concludes this deliverable and provide a brief outlook for each component.
- **Annexes:**
 - **Annex A:** Document History
 - **Annex B:** References
 - **Annex C:** Data Model and Interfaces

0.6 Document Status

This document is listed in the Description of Action as public. Therefore, details which may temper the security of the platform or its components are not content of this deliverable.

0.7 Document Dependencies

This document is part of an iteration of living deliverables. This is the first iteration and will only cover specific subsections for Section 1, 2 and 3. This second final iteration (D5.16: EFPF Interfacing, Evolution and Extension - Final Report) completes the document.

0.8 Glossary and Abbreviations

A definition of common terms related to EFPF.

0.9 External Annexes and Supporting Documents

Annexes and Supporting Documents:

- None

0.10 Reading Notes

- None

1 Ecosystem, Evolution and Extensions Requirements

1.1 Requirements for a Federated Ecosystem

A goal of the EFPF project is to develop a digital manufacturing ecosystem that demonstrates a federated approach to functionality, processes, and users in a bid to ensure the initiative is sustainable. This is motivated by the shift in architecture design in recent times in favour of microservices enabling enhanced flexibility, rather than the proprietary digital platform approach that has been attributed with vendor lock in, proprietary standards, inhibiting evolution.

A federated approach needs to bring together different stakeholders into a level playing field where they can perform collaborative activities and develop relationships such as those between providers and suppliers of products and services. The federation model also supports the creation of competitiveness networks, which can be composed of multi-industry partners; and facilitates knowledge exchange between different stakeholders through collaborative processes. Requirements to guide these processes need to be understood e.g. what is the source of the requirements, their type (whether user-oriented, non-functional, technical), who are they applicable to etc. In this respect, a dedicated task in the EFPF Project (T2.5: EFPF Interfacing, Evolution and Extension) gathers requirements from a range of ecosystem stakeholders starting with the partners involved in this task, which come from a variety of domains including Industrial, Research and Technical. Further requirements are collected from partner interactions, interactions and collaborations with different projects, from the open calls experimenters and other activities and stakeholders engaged in the project duration. This task (T2.5) runs throughout the duration of the project with the aim to continuously feed new/emerging requirements into the project (development, business modelling, ecosystem creation etc) activities.

The requirements for a federated ecosystem gathered so far in the project are relevant to the following roles:

- **Developers:** Technical users who may develop and wish to deploy Tools and Services in the EFPF ecosystem
- **Systems Integrators:** Technical users who want to leverage the latest smart factory and digital manufacturing toolsets, in order to be able to offer them to customers
- **Platform managers:** Interested in extending the scope of an existing platform or to start a new platform
- **Manufacturing Managers:** Business focussed users who would like to access, reliable mature and productivity generating tools and services

The guiding requirements for a federated ecosystem to be established through the EFPF project are described below. These requirements are defined through partner consultations and study of existing literature on federated ecosystems and the digital manufacturing platforms. Naturally, these requirements will evolve overtime as the project develops links with other stakeholders and explores new relationships (e.g. through open-call mechanism).

- A federated ecosystem should have multiple systems and platform virtually connected through software interfaces for seamless authorised access. Ideally, the connectivity should extend beyond the software accessibility and interoperability aspects to also support interactions between tools/service providers and user communities

- The federation should provide necessary trust, security, and privacy mechanisms to ensure that the partners and their interactions in the ecosystem are safe from potential threats like cyber-attacks, session riding, hijacking etc
- The federation should support the heterogeneity of the ecosystem and provide necessary governance mechanisms that allow the relevant control (data, access, business models etc) to remain within the partners
- The federation should implement governance procedures and technology mechanism that act as a deterrent for any potential monopoly
- The platforms in the federation should be conducive to integration and interoperability. The federation should allow users to connect and operate through APIs, applications, and third-party service libraries. This would further eliminate the possibilities of monopoly and vendor lock-in
- The federated ecosystem should provide an easy to use service integration and interoperability platform/mechanism to make the provision and consumption of services as easy as possible, enhance developer productivity and enable collaboration amongst them.
- The service integration and interoperability platform/mechanism offered by the federated ecosystem should make use of standards and its design should be flexible enough to have the choice of multiple open source technologies to realise its conceptual components. This would also help in avoiding vendor lock-in.
- The service integration and interoperability platform/mechanism offered by the federated ecosystem should have inbuilt support for standard communication protocols and for data transformation tools/languages, such as XSLT (Extensible Stylesheet Language Transformations), that are widely used in the industry.
- The service integration and interoperability platform/mechanism offered by the federated ecosystem should be scalable and should support high availability and high throughput.
- The federation should promote the use of standards for the design and implementation of components of the federated ecosystem to make the integration of new services, tools, and platforms effortless.
- The federation should provide adequate documentation in the form of HowTos, Tutorials and API documentation of provided services in order to ease the development of applications using distributed services from the various platforms and service providers within the EFPF ecosystem. The federation should promote the use of standard deployment practices such as containerization to make the deployments of components of the federated ecosystem easy to manage and maintain.
- The federated ecosystem should inherently provide services, tools, utilities that are widely used to realise the common use cases in the domain.
- The legislative regulation across different regions may be different so it is important to have a fair geographical distribution of vendors or datacentres to sustain the federated ecosystem

1.2 Requirements for the Evolution of a Federated Ecosystem

The process of evolution of a federated ecosystem should be self-governing in nature and encompass change across all aspects of the characteristics that make up a federated system. Naturally, evolution happens over time based on many different factors that are often not controllable or even predictable (such as behaviours of different actors, environmental changes, interactions between different entities etc). As an ecosystem grows

(e.g. with the number of actors, the interactions, different types of activities etc) the need for supporting new actors, new roles, new interactions as well as adaptation of existing systems and processes also grows. Therefore, the requirements for the evolution of a federated ecosystem need to highlight the need for certain level of flexibility, openness, and adaptability in the underlying federation (mechanisms) to ensure its evolution overtime.

While collecting requirements for the evolution of a federated ecosystem, the following guiding principles are:

- **Diverse user base:** This will ensure that the market vulnerability isn't a risk by naturally evolving to target users in new domains.
- **Heterogeneity:** Use of microservice architecture should support a natural evolution of functionality and process to support in an adaptive robust way.
- **Geographical distribution:** This is also an essential characteristic, where evolution can be naturally supported through distributed deployment of microservices and interoperability, but ultimately driven by human processes.
- **Interoperability:** Interoperability mechanisms can support these evolutionary step changes

Based on the above recognition and consideration for the following principles the following requirements for the evolution of federated ecosystem are gathered so far in the project:

- A federated ecosystem in essence should constantly evolve to include new partners, tools and services; in order cater to the users need in present and in the future
- There should be feedback loop to capture user experiences and proper user interfaces should be defined to allow the multi-part interactions to take place and keep the ecosystem moving
- The interoperability feature of the platform should attract new software integration to develop opportunities for different types of interactions and business
- Overtime the ecosystem should adapt and accommodate new stakeholders and usage scenarios
- In a federated ecosystem, the involved parties rely on the agreed API Contracts or Interface Contracts for communicating with each other. However, as the participant services evolve, the upgradation of APIs becomes necessary and inevitable. Therefore, the federated ecosystem should define Interface Contract policies that allow the Service Providers to convey plans to deprecate/upgrade their APIs to the Service Consumers in advance allowing a smoother transition/collaboration.

1.3 Requirements for the Extension of a Federated Ecosystem

To support the goals of evolution step changes, a framework of processes will be needed to support the concrete development steps and allow the extension to happen in a managed manner that is disseminated to all stakeholders. The microservices architecture and REST communications, rather than close coupled systems, naturally supports extension of the ecosystem. However, topics including deployment, version management, service lifecycles and protocol support will all need to be planned for and delegated to ensure the process of extending the ecosystem is handled particularly as the it grows.

As this is an ongoing task, the EFPF federation is currently being development and therefore it is too early to formulate the requirements for the extension of the federation. However, high-level requirements to guide the ongoing development activities are listed below:

- The federated platform should be extensible and there should be interfaces for the inclusion of new services and gateways available for data exchange
- There should be openly accessible APIs for the integration of external tools
- The federation should provide necessary support, not only in terms of interfaces and connectors, but also in terms of human resources (experts) who can facilitate the onboarding of new partners and technologies
- The federated ecosystem should promote cross domain and cross border interactions and therefore the extension mechanisms should be able to capture the differences in languages, approaches and techniques
- Documentation should be available to describe the scenarios in which different extensions should be done
- The service integration and interoperability platform/mechanism offered by the federated ecosystem should follow a modular and extensible architecture, thereby making it possible to add support for new communication protocols, data transformation tools, etc

The above requirements will be continuously refined and updated throughout the project to provide more specific guidelines to the technical activities in the project. The updates will be published in the next iteration of this deliverable at M48 of the EFPF project.

1.4 Governance for the Growth of the EFPF Platform Ecosystem

The complexity of digital platform ecosystems comes with the decision-making twist between different sectors and different actors operating at multiple levels, including organisational, sectoral, local, regional, national, international, etc. The EFPF is an emerging ecosystem of multi-sided digital platforms and requires governance mechanisms to be in place, to effectively reach its goals and create sustainable outcomes. The governance mechanisms for digital platform ecosystems need to reflect on the lawful interactions of key stakeholders, be they owners of the platforms, companies using the platform, or developers, users, advertisers, economists, computer scientists, governments or regulators. To stimulate positive interaction payoffs within the platform ecosystem, both platform stakeholders and platform technology enablers must be regulated and governed.

The D5.12 "EFPF Security and Governance" describes the EFPF Governance Framework (eFGF) that incorporates platform organizational standards, strategic planning, business rules and norms of behaviour within the ecosystem, software standards, regulatory requirements, and other aspects which need to be continuously monitored and assessed from the perspective of various EFPF users (stakeholders).

The eFGF covers five functional areas:

- AREA 1: Terms and conditions ("Terms of Use", "User Agreement", or "Terms of Service Agreement"): it outlines the terms and conditions the user must agree to in order to interact with the EFPF platform. Well-established terms and condition-related rules prevent misunderstanding between the EFPF platform owner and the users, by defining e.g. Intellectual Property Rights (IPRs) protection, limiting responsibilities towards third parties, setting platform rules and the consequences for violating these rules, etc.
- AREA 2: Architecture and IT governance: it includes a policy-based control of information to meet all legal, regulatory, risk, and business demands. The focus in on

the actual software development and maintenance activities of the IT that are aligned with the business objectives of the platform, e.g. with the European Factory Foundation.

- AREA 3: Data governance, service policies, APIs policies and SDKs include those rules that define i.e., processes and controls to ensure that information at the data level is true, accurate, and unique (not redundant). Data policies address policies related to personal data (data belonging to the user of the EFPF platform), corporate data (data belonging to the company that user of the EFPF platform represents), community data (data belonging to the group of users that get together over the EFPF platform for sharing specific interests, e.g. forum, working teams, etc.), sectoral data (data that constitute a specific sector, e.g. Smart Manufacturing sector, Automotive Driving sector, etc.) and potential infrastructural data lakes (data that constitute a larger body of data, e.g. Common European Data Space [EC-DATA20]).
- AREA 4: Marketplace rules, trust and reputation that define the expected behaviour in the platform ecosystem, in order to create expected positive effects.
- AREA 5: The existing laws and regulations at the international and national levels, including the EU Network and Information Security (NIS) Directive with the goal to enhance cybersecurity across the EU; Incident notification for DSPs in the context of the NIS Directive; GDPR [GDPR18] that adopts the core principles required for personal data processing; Ethics Guidelines for Trustworthy AI, by Independent High-Level Expert Group (HLEG) on AI set by the EC [HLEG19]; Ethically Aligned Design (EAD) [EAD19] which is an initiative created by the IEEE Standards Association and covers many topics of interest to EFPF development, including e.g. general (ethical) principles; how to embed values into autonomous intelligent systems; methods to guide ethical design; safety and beneficence of artificial general intelligence and artificial superintelligence; personal data and individual access control; reframing autonomous weapons systems; economics and humanitarian issues; law; affective computing; classical ethics in AI; policy; mixed-reality, and well-being.

One of the greatest business challenges is how to get competitors to cooperate with each other in the platform ecosystem. Here, efficiency gains, Return of Investments (ROIs) and other incentives need to be supported through platform ecosystems by selecting the adequate governance mechanisms. To support the further growth of the EFPF ecosystem, we focus on these attributes that promote the common good and limit negative impact and conflicts either via the platform or in the platform ecosystem. Some examples of such attributes are: Inclusion, participation, accountability, etc.

The authors in [GRIN07][GISS12] emphasize good governance as an essential driver of sustainable development. Implementing good governance practices requires knowledge about existing actors (stakeholders) and their business- and government-driven rules, as well as knowledge about possible benefits and losses, and other external effects that can cause both positive and negative effects (conflicts).

In EFPF, we adopt the concept of "positive platform governance" as discussed in [MATR17], that emphasizes the importance of shared decision making with the platforms' contributors, through the following principles (see Figure 1 below):

- Inclusion - which is about defining the roles and power for actors with diverse levels of participations;
- Participation - which is based on principles of fairness, simplicity, transparency and trust integrated in all decision-making processes of the platform;

- Autonomy - which helps assuring that all contributors affected by those decisions will be able to participate;
- Recognition of the generated value - ensuring reward system for all contributors in the ecosystem.

Principle	Mechanism
Inclusion of the max number of stakeholders in decision-making, ensuring that contributors have decision-making power that's proportionate to their level of activity and engagement	1) Binary power system (power defined by the owner) 2) Smooth power system (power defined by the level of contribution and commitment)
Fostering participation by embedding the principles of fairness, simplicity, transparency, and trust into decision-making process of the platform	1) Online participation in the decision-making (comm. channels/tools) 2) Participatory mechanism (consensus, majority, holacracy) 3) Collab. decision-making
Embedding autonomy in the decision-making to keep agility while scaling, but assuring that all contributors affected by those decisions will be able to participate	
Recognition of the generated value: Ensure a fair reward systems for all value contributors on the platform	Adding value accounting system (algorithms, peer feedback, etc.) that measures and accounts for contributors, with well defined rules.
Welfare: Inclusion of protection and security for contributors at the core of the governance model	Policies; Insurance; etc.

Figure 1: Adopting Positive Governance Mechanisms in EFPF

2 EFPP Marketplace

This section provides a general view on digital marketplaces in the EFPP context and information about the EFPP Integrated Marketplace Framework and associated components, as listed below:

- **Integrated Marketplace Framework:** This provides mechanism to enable access to external marketplace offerings and the necessary GUI to present a unified interface to the EFPP users
- **Automated Agent-based Marketplace and Online Bidding:** This component provides an agent-based marketplace with automated bidding functionality
- **Accountancy Service:** This component provides features to track & trace and credit users of connected marketplaces

Marketplaces are common components in digital manufacturing platforms. They provide collaboration and interaction between a platform and its service and tool provider and its customers via a catalogue framework. This allows products to be listed and therefore browsed based on categorisation and searched according to search terms and filter conditions. An outcome of this is the natural knowledge transfer and technology exchange between users. Marketplaces may list only products provided by the respective platform or provide products coming from third party platforms.

In this context marketplaces can be divided into two types:

- **Hosting marketplace:** Products listed are provided by the hosting platform of the marketplace, which can offer standard ecommerce functionalities such as listing new products, which are uploaded directly to the marketplace, and checkout functionalities, which will handle customer transactions within the marketplace
- **Integrated marketplace:** Products listed are provided by marketplaces from external platforms. This type does not offer the above-mentioned ecommerce functionalities. In this scenario products are listed on the external marketplace, which also provides the transaction and checkout functionalities

2.1 Integrated Marketplace Framework

An integrated marketplace framework is implemented within the EFPP project. The integrated marketplace framework connects with external marketplaces in the EFPP ecosystem through API interfaces. This allows the integrated marketplace framework to list the products (from external marketplaces) and provide typical filter and sorting mechanisms, with the ecommerce features (e.g. product listing and checkout) provided by the base/external marketplace.

To expand the product offering, the integrated marketplace framework allows users to publish new products and services by providing redirects to the relevant external marketplaces. The newly published products/services become available on the unified GUI provided by the integrated framework at the EFPP Platform. As connected external marketplaces may provide different feature sets and may have different focus e.g. a marketplace may focus on software apps and the other marketplace may provide features that are more suited to physical products. In this respect, the EFPP integrated marketplace framework is able to bring different types of offerings to the users in the EFPP ecosystem and provides a unified interface that allows users to search for different types of offerings through a an intuitive and unified interface – available through the EFPP Portal. Therefore,

it may be possible that future external marketplace may be added and offer additional opportunities to the EFPF users.

The marketplace integration approach adopted in the EFPF project allows a fast integration of external marketplaces and their products into the EFPF ecosystem and therefore is more suited for a federated environment, where multiple marketplaces co-exist. As each marketplace may have a different data model and also a different design of the marketplace, the integrated marketplace framework performs the necessary transformation at the level of data model and interface – through custom configurations and integrations that are also supported by the Data Spine.

Additionally, the EFPF Marketplace provides federated services such as single sign-on (SSO), which allows customers to use connected marketplaces without registering on each single marketplace again. This all leads to a unified user experience and similar user journeys.

2.1.1 Scope and Relationship with Other EFPF Components

The Marketplace Framework provides access to different external marketplace.

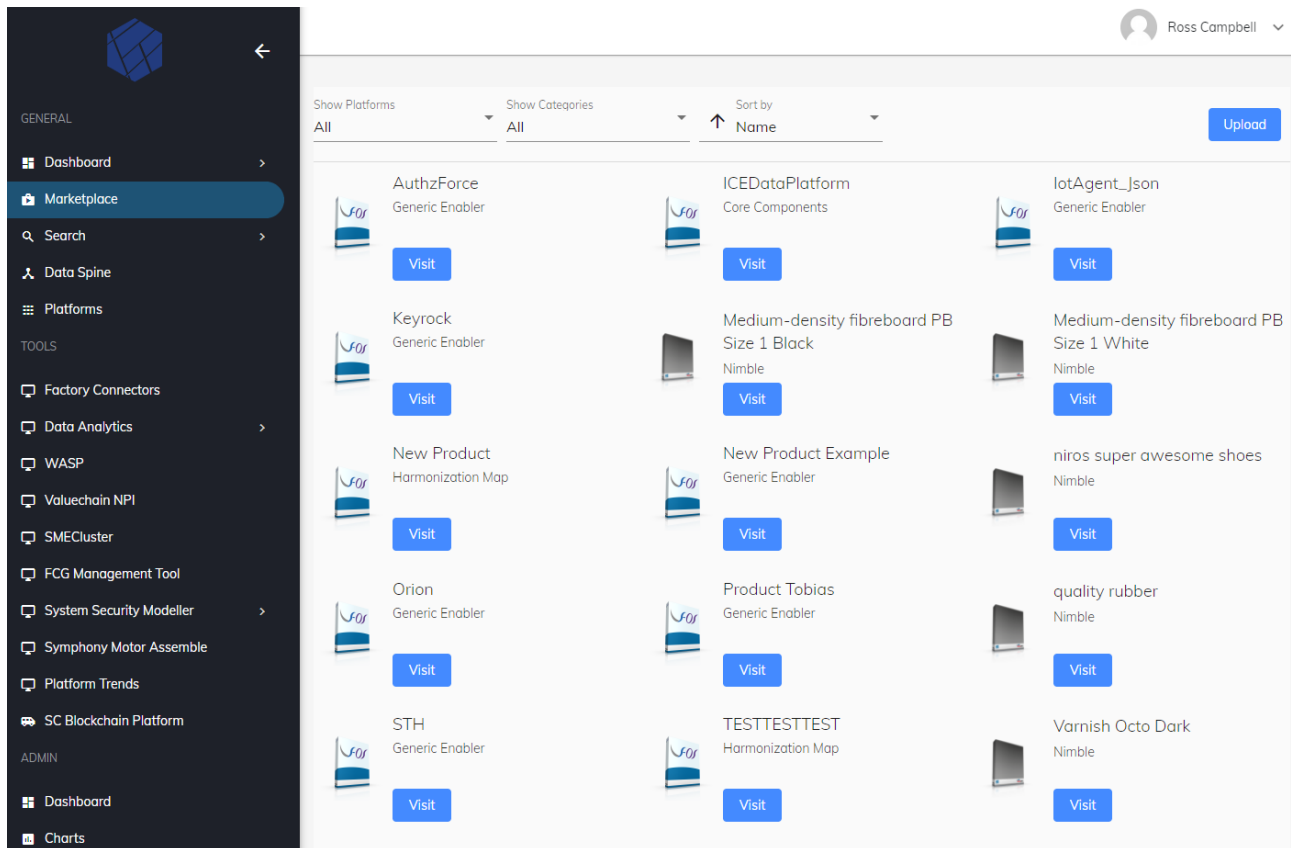


Figure 2: Marketplace Framework - User Interface Screenshot from EFPF Portal

Currently the following external marketplaces are connected and their products are listed through the unified user interface of the EFPF marketplace framework:

- vf-OS¹: A marketplace of software applications that are mostly plug and play in nature

¹ <https://www.vf-os.eu/>

- Nimble²: A marketplace of physical products
- SMECluster³: A marketplace of systems that require consultancy and configurations
- WASP: A marketplace of software/web services

Each external marketplace has been adapted to enable Single Sign-On (SSO), so EFPF platform user can access the marketplace without having to register again. Nevertheless, a marketplace may ask for providing additional information as the EFPF portal may not gathered all information at registration time.

The following list provides an overview of the relationship regarding connected components and a description of the functionality it enables:

- **EFPF Portal:** This component acts as a container for the Marketplace Framework UI as can be seen in Figure 2. The portal at this point also provide access control as only registered members have access to this section.
- **Integration Flow Engine:** This component provides REST interfaces to retrieve product data from external marketplaces.
- **Identity and access management component EFS:** This component secures access and communication with both components mentioned before.

2.1.2 Interlinked External Marketplaces

This section provides an overview and description of each marketplace that is connected with the integrated marketplace framework and therefore their offerings are available through the unified Marketplace interface on the EFPF Portal. Each marketplace exposes its product catalogue via REST interfaces. The internal marketplace framework can retrieve the products and list them including a link to the details page of a product inside each of the external marketplaces.

vf-OS Store

The vf-OS Store has been developed in the scope of the European research project vf-OS. It provides a one-stop-shop for the extension of the vf-OS platform, which are called vApps. Users can browse listed applications, filtering by categories, providing feedback and view dependencies. It provides an easy checkout through various payment methods.

Key features:

- Browser applications as a customer
- Easy checkout
- Manage uploaded applications

The vf-OS Store also provides an upload feature which will be used to upload any downloadable software applications using the integrated marketplace framework inside the EFPF portal.

² <https://www.nimble-project.org/>

³ <https://www.smecluster.com/>

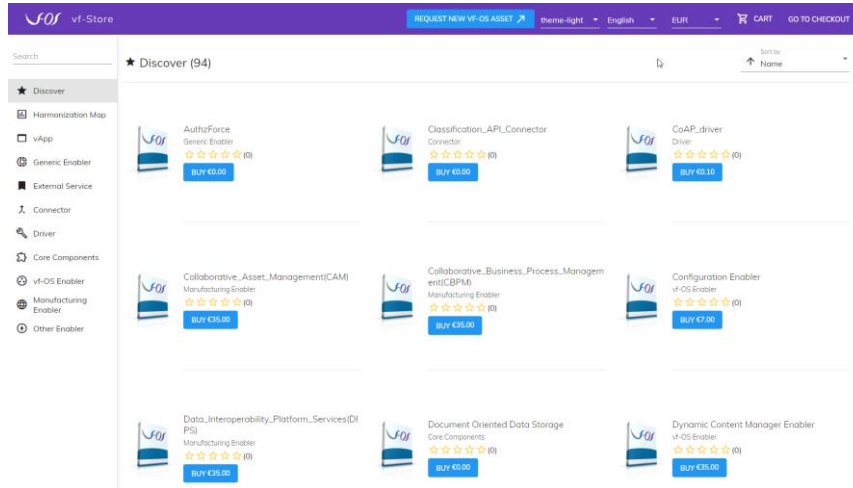


Figure 3: Marketplace Framework - vf-OS Marketplace Screenshot

SMECluster Marketplace

The SMECluster Marketplace is an integrated part of the SMECluster website providing an overview of available tools and services. These tools and services may consist both of software and hardware and additional set-up work. Therefore, many listings cannot be bought right away but the customer provides its contact data for more information and a personal contact.

Key features:

- Browse products by categories
- Provide rich descriptions
- Support products including hardware setup
- Checkout and payment feature

The SMECluster Marketplace will be used to provide tools and services which require a manual configuration or adaptations to be used at the customers premises.

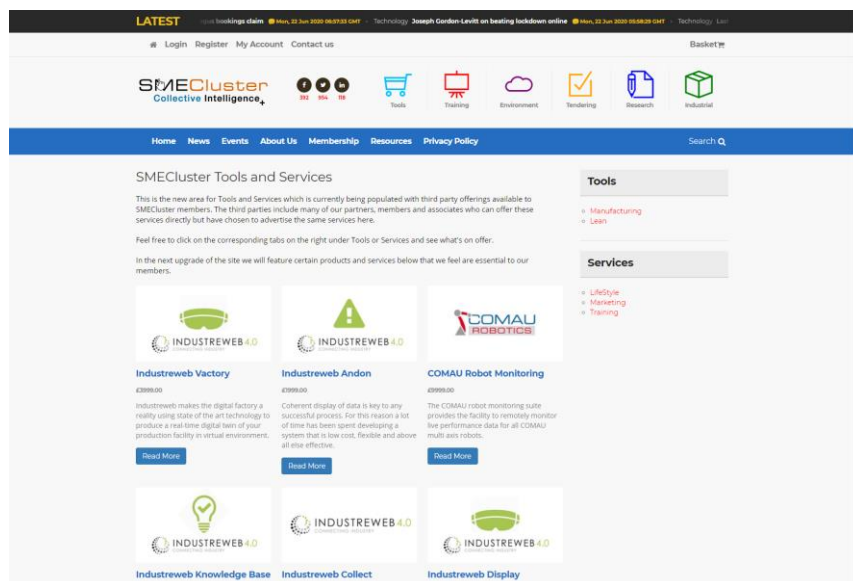


Figure 4: Marketplace Framework - SMECluster Marketplace Screenshot

Nimble Marketplace

The Nimble Marketplace has been developed in the scope of the European research project Nimble and provided a federated, multi-sided and cloud services-based business ecosystem. The Nimble Marketplace is the front-end of this ecosystem and enables users to buy, offer and negotiate.

Key features:

- Company and product search
- Product catalogues
- Advanced checkout incl. negotiations

The Nimble Marketplace will be used to provide physical products.

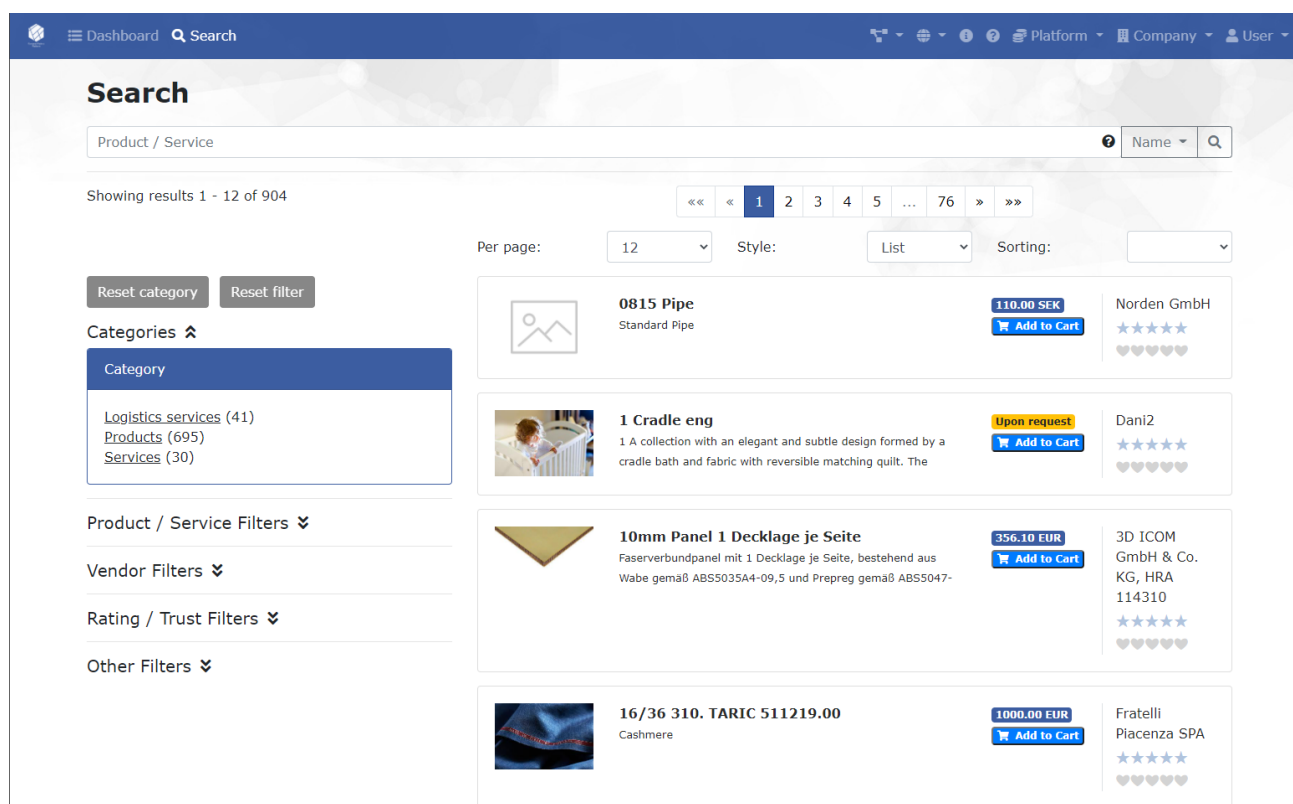


Figure 5: Marketplace Framework - Nimble Marketplace Screenshot

WASP Marketplace

The WASP Marketplace is part of the Workflow and Service Automation Platform. This marketplace provides software services ready to use in websites and other components. Services are sorted by categories and detailed information are provided.

Key features:

- Web services search
- Service listing
- Bubble navigation

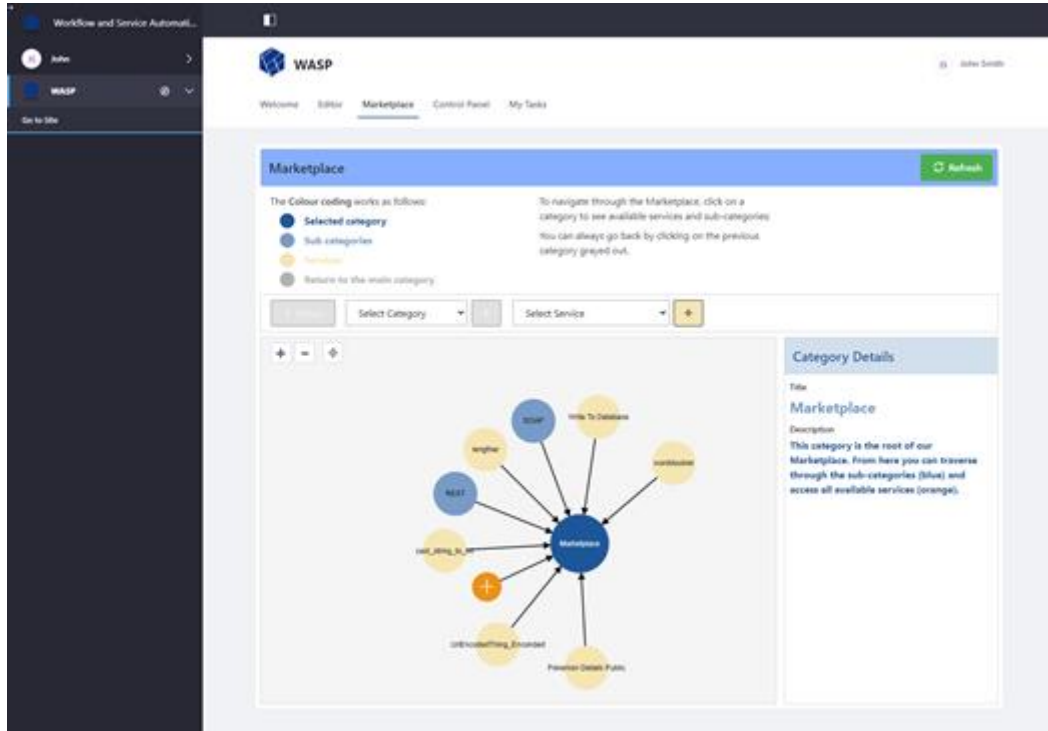


Figure 6: Marketplace Framework - WASP Marketplace Screenshot

2.1.3 Requirements and Realisation

The Marketplace Framework has the following requirements:

- Display items provided by external marketplaces
- Track, trace and credit users
- Upload and checkout of items

The first requirement is already implemented by displaying items from external marketplaces mentioned in Section 2.1.1. The second requirement (which is a set of requirements) are being provided by the Accountancy Service (in Section 2.3). The third requirement is the focus of on-going work at it requires upgradation of certain functionalities at the level of base/external marketplaces that are responsible for these functionalities.

The Marketplace Framework is currently realized as a custom Web Component based on the Angular web application framework. As a web component, it has to be integrated in a website to use it. To retrieve products from external marketplaces this component connects to the Data Spine and makes use of the provided product routes. Future versions will make use of the Service Registry and will use a backend subcomponent.

2.1.4 Deployment

The Integrated Marketplace Framework is offered as an open component in the EFPF ecosystem. This means that any other platform in the EFPF ecosystem can also take advantage of these features to enhance their functionalities. In its current form the framework can be integrated as a JavaScript library in a website. After importing the library into the website, which may differ based on the used framework, the developer can use the “EFPF-marketplace” HTML element in their source code as shown in Figure 7. At this point

the element requires a valid Java Web Token (JWT) to retrieve products from the EFPP platform.

```

1
2
3 <efpf-marketplace [IdToken]="idToken"></efpf-marketplace>
4
5

```

Figure 7: Marketplace Framework - Source Code Deployment

As Section 2.2.5 describes this component is subject of change as platform communication will be moved to a separate backend component.

2.1.5 Execution and Usage

By default, the Marketplace Framework lists all available items sorted by the name attribute. Additionally, the user can sort the products based on the item category, which is provided by the source marketplace. The user has also the option to negate the sorting.

The user can also use the provided filters, which allows to filter the list based on the source marketplaces and the categories. Both filters can be reset to their default values.

Each item provides basic information, which are the name, categories, item image and an external link, which leads the user to the detail page of the source marketplace.

2.1.6 Limitations and Further Development

The marketplace framework is currently provided as Angular web component embedded in the EFPP portal. This approach provides an easy integration into other websites, but it also limits adaptation to future external marketplaces. Therefore, it is planned to create an additional backend component, which handles the communication with the Data Spine and its subcomponents and therefore also support the communication with the Service Registry.

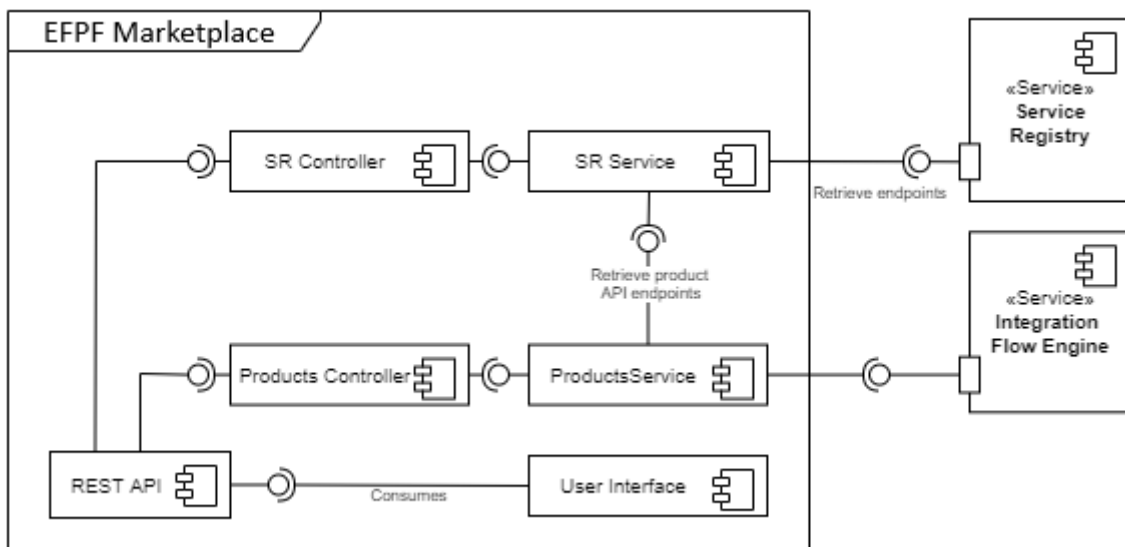


Figure 8: EFPP Marketplace - Planned Architecture Snapshot

2.2 Automated Agent-based Marketplace and Online Bidding

2.2.1 Scope and Relationship with Other EFPF Components

Besides the overall marketplace framework of the project, an agent-based marketplace will be also available to users. This type of marketplace aims to provide mechanisms that enable the automated negotiations within the participants. In this marketplace, each company sets up an agent which represents the company in an Online Bidding Process in which the agents negotiate for specific goods. This provides automation of existing manual procedures, and reduce time costs for traditional negotiations carried out by phone calls and email. The Online Bidding Process component is composed of three main components:

- Agent ecosystem
- Matchmaker
- User Interface

The platform is derived from the COMPOSITION project and will be integrated with the EFPF ecosystem to provide automated bidding process functionalities to EFPF users. This is done to propose a faster way to manage supply/demand matching to speed up the process of reaching agreement between two parties.

The Agent ecosystem is composed of a set of agents built to communicate with each other and with other platform's components to perform an automated negotiation. To achieve this an Agent includes modules to support different communication protocols using their own communication language called CXL (Composition eXchange Language) derived from the FIPA ACL communication standard. Agents are differentiated by the following roles:

- **Requester:** Represents a company who is requesting for a service or/and a related good
- **Supplier:** Represents a company who provides a service

The Matchmaker component provides a full semantic framework for the agents with CRUD operations for agents/companies and two type of matchmaking functionalities. It matches requesters with suppliers for a service/good and in a second level matches the request with the best available offer (coming from supplier agents' bids) based on different evaluation criteria such as price, payment and delivery methods, reliability etc.

The UI of the bidding process is web-based and enables a user to carry out the following functions:

- Register/set up and agent
- Initialize a bidding process in order to request for a service/good (requester)
- Add information and priorities for the request (requester)
- Online monitoring of the process for both suppliers and requester
- Opportunity for supplier to bid for a request
- Suggestion for best offer and details of all submitted offers in the case the user want to ignore system suggestion and evaluate by himself (requester)
- Visual notification for win/lose in bidding process (supplier)
- Bids history available to user (requester)

The system will be made available to the user through the EFPF portal where an authenticated user will be able to configure and set up a new agent and to control the bidding processes by the means of UI.

2.2.2 Requirements and Realisation

Design Requirements

The Online Bidding Process is a stand-alone component that is also made of agents representing companies to enable them to perform automated negotiation following predefined protocols and logics. The platform coming from the COMPOSITION project expects that each agent should be manually built and deployed together with an ad-hoc designed UI for each company taking into account the needs and the use cases. The EFPP platform has a wider scope with respect to the COMPOSITION project from which this tool derives from.

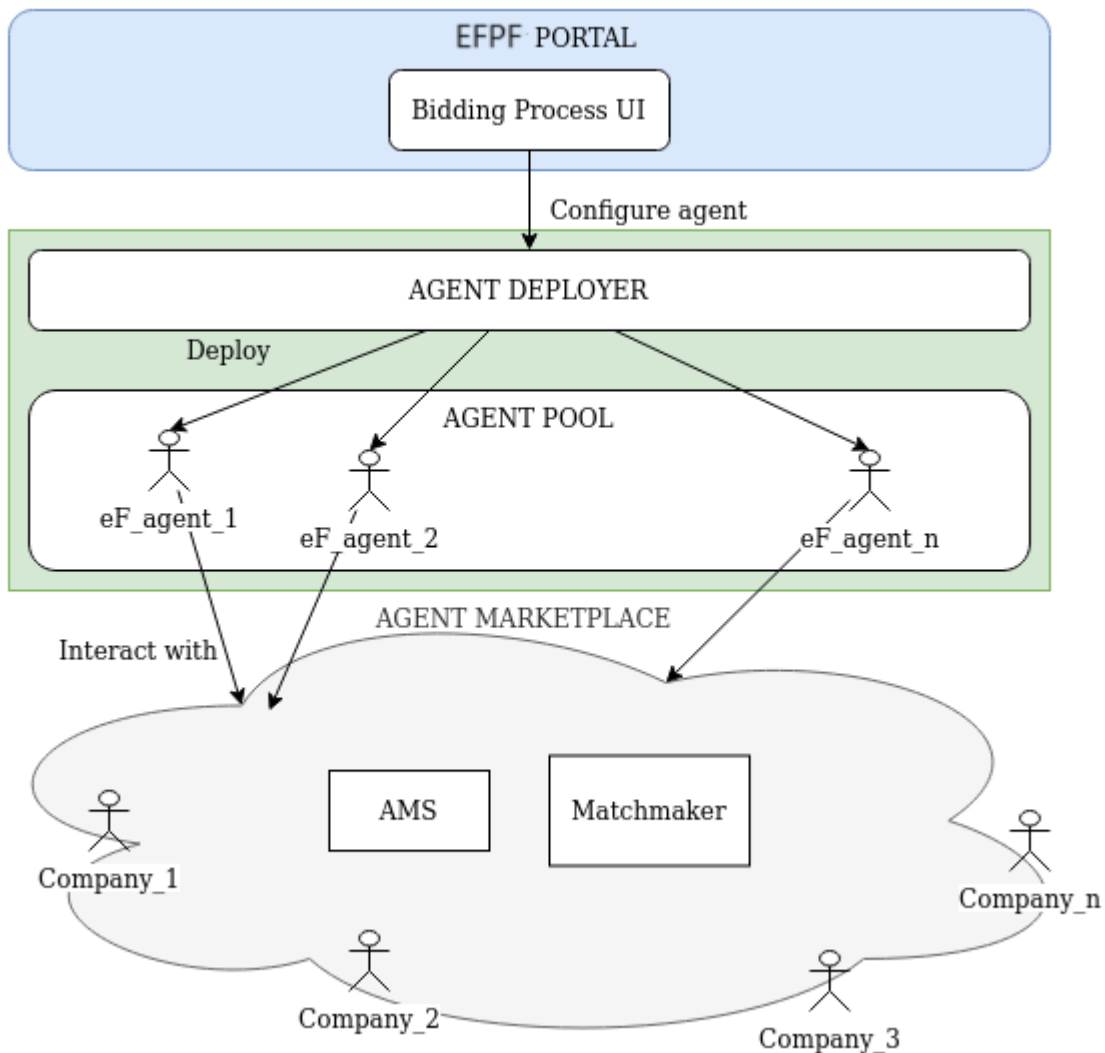


Figure 9: Agent Marketplace - Deploying Process

To overcome these issues and to integrate the solution with EFPP platform, new components have been developed:

- The Agent Deployer
- New UI
- Extensions on matchmaking functionalities

The Agent Deployer component provides a REST API interface to enable the user to configure a new agent that will be deployed in an Agents Pool. This pool will collect all the custom agents deployed by the EFPF platform making them communicate with the existing agent ecosystem.

The Agent Deployer API is used by the new Online Bidding Process UI which was developed within EFPF as a web form that enables users to create their own agents, enabling a more generic but flexible approach to Agent creation. In addition, the UI is rendered dynamically from the semantic framework rather than being designed explicitly for two or three partners only. The general design approach for EFPF was to move from a custom solution to a more generic one in order to satisfy user requirements.

For the Matchmaker side that provides CRUD operations to agents, new web services have been developed to meet the Agent Deployer and UI needs.

In addition to the components that have been designed for EFPF needs, developments to existing modules have been applied or scheduled for future developments. All the updates to both front end UI and back end services are driven by the need for generic services throughout the platform.

User Requirements

Besides the requirements that have been extracted by the project design approach, the agent-based bidding process is a core component for Circular Economy pilot of the project.

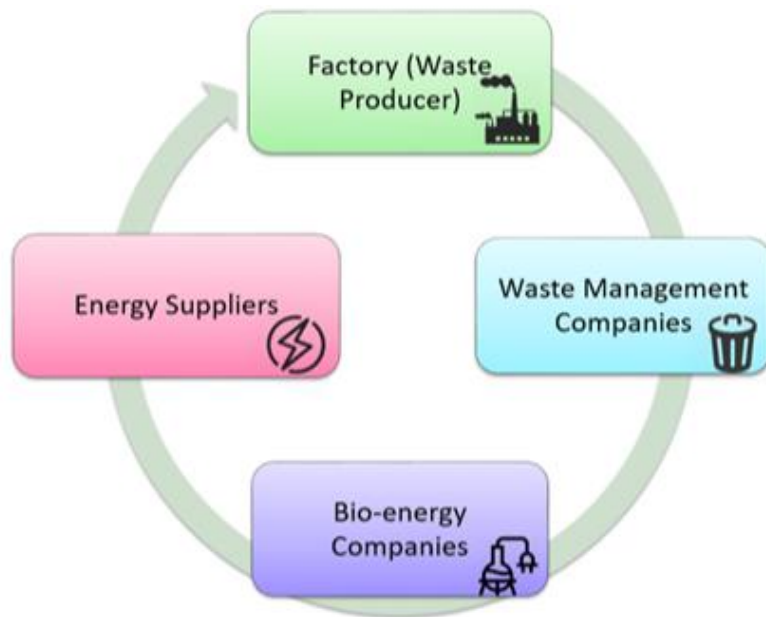


Figure 10: Agent Marketplace - Process

In this pilot there is the major requirement for the users to be able to negotiate automatically for some goods/services in this closed loop. The pilot partner that manufacture lifts wants to negotiate its scraps automatically through by using its agent and the bidding process. The waste management company that wins on bidding process wants to sell the processed waste to a bio-energy company by using the same system as well. Finally, the produced energy by the scraps returns to lift manufacturer in this circular scenario.

Therefore, a Purchasing manager or purchasing specialist wants to negotiate prices and contracts so to obtain high-quality services/products at reasonable prices in an automatic way. This need led to design this automated online bidding mechanism that provides suggestions of best available supplier for a service and enables the automation of negotiation procedures in a close loop. More detailed pilot requirements are available on the corresponding deliverable.

2.2.3 Deployment

The components that needs to be used to make the Online Bidding Process working are:

Agent Deployer and Agents Pool: The Agent Deployer service in his actual implementation should be deployed on a server to provide access to APIs and to handle the Agents Pool, being able to control the agent lifecycle. The Agents Pool is based on Docker technology and a Docker container represents each agent. Now, both components are deployed in a server at LINKS premises.

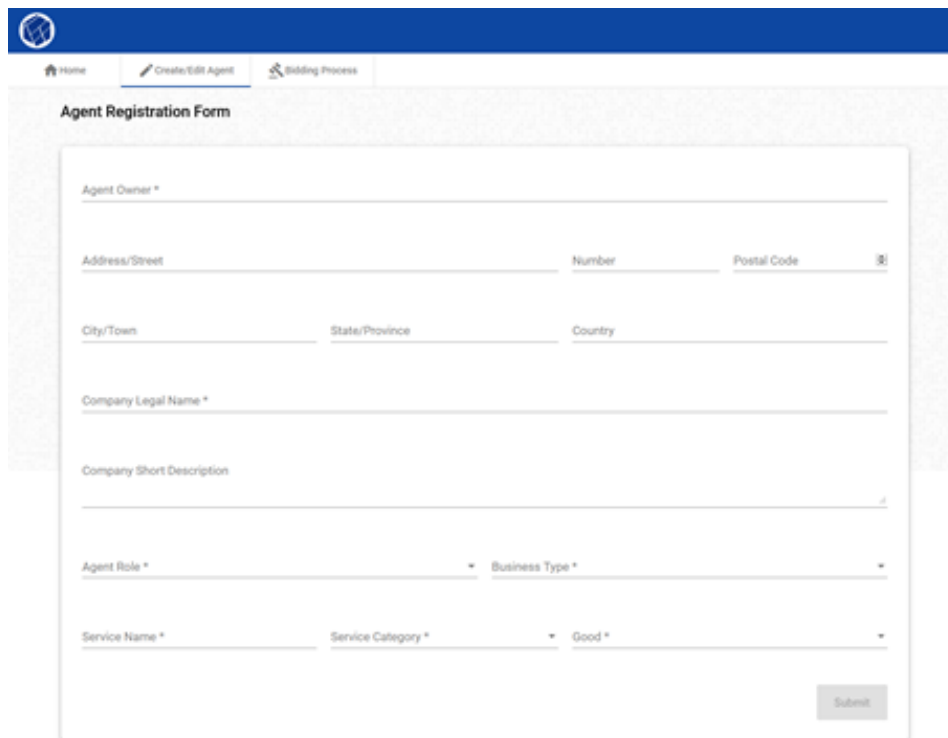
User Interface: Currently the UI is deployed on a CETH's server.

Matchmaker and Agent Management Service: Already existing components ported by the COMPOSITION project (i.e. Matchmaker and Agent Management Service) are currently deployed as Docker containers in an EFPF dedicated space hosted by FIT. All, the updated Docker images for these components are updated to these containers, in the environment that has been set up for EFPF purposes. The AMQP broker used to enable agent's communication should be mentioned also. At the moment it is deployed with other containers in EFPF space hosted by FIT but, as further developments, will probably be moved according to EFPF platform integrations needs.

2.2.4 Execution and Usage

The features provided by this tool will be available through the User Interface integrated into the EFPF portal. The basic UI that has been implemented in Angular 9 by keeping the design direction adopted by COMPOSITION and CNET. However, all the static attributes replaced with dynamic ones that enable the interfaces usage by different partners. The UIs are available to the EFPF user are:

- A registration form – as shown below

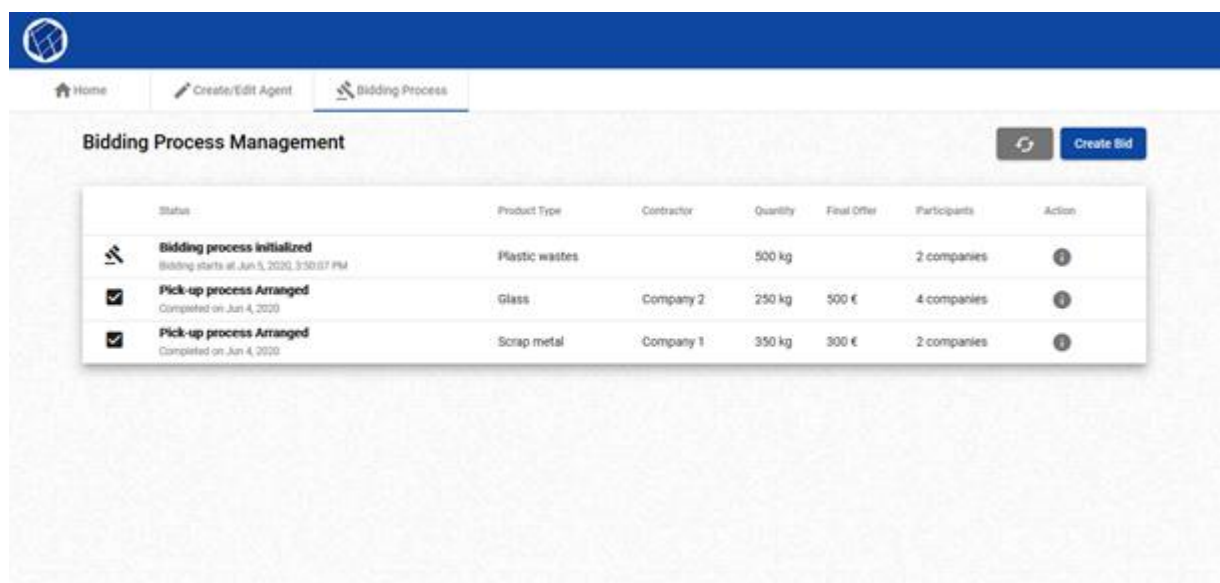


The image shows a web application interface for the Agent Marketplace. At the top, there is a blue header bar with a logo on the left and three navigation tabs: 'Home', 'Create/Edit Agent', and 'Bidding Process'. The 'Create/Edit Agent' tab is currently selected. Below the header, the page title is 'Agent Registration Form'. The form itself is a white box with a light gray border. It contains several input fields: 'Agent Owner *' (a single-line text field), 'Address/Street' (a single-line text field), 'Number' (a single-line text field), 'Postal Code' (a single-line text field with a small icon on the right), 'City/Town' (a single-line text field), 'State/Province' (a single-line text field), 'Country' (a single-line text field), 'Company Legal Name *' (a single-line text field), 'Company Short Description' (a multi-line text area), 'Agent Role *' (a dropdown menu), 'Business Type *' (a dropdown menu), 'Service Name *' (a single-line text field), 'Service Category *' (a dropdown menu), and 'Good *' (a dropdown menu). A 'Submit' button is located at the bottom right of the form.

Figure 11: Agent Marketplace - Agent Registration Form

By using this form, a company register its information to the semantic framework that consists the knowledge base of the agent ecosystem. After that, the agent for this company is automatically created. This agent can be used to represent the company in online bidding processes for specific services and goods.

- Bidding Process Management Dashboard



The image shows a web application interface for the Bidding Process Management Dashboard. At the top, there is a blue header bar with a logo on the left and three navigation tabs: 'Home', 'Create/Edit Agent', and 'Bidding Process'. The 'Bidding Process' tab is currently selected. Below the header, the page title is 'Bidding Process Management'. On the right side of the header, there are two buttons: a 'Refresh' button (a circular arrow icon) and a 'Create Bid' button. The main content area contains a table with the following columns: 'Status', 'Product Type', 'Contractor', 'Quantity', 'Final Offer', 'Participants', and 'Action'. The table has three rows of data:







Status	Product Type	Contractor	Quantity	Final Offer	Participants	Action
 Bidding process initialized Bidding starts at Jun 5, 2020, 3:50:07 PM	Plastic wastes		500 kg		2 companies	
 Pick-up process Arranged Completed on Jun 4, 2020	Glass	Company 2	250 kg	500 €	4 companies	
 Pick-up process Arranged Completed on Jun 4, 2020	Scrap metal	Company 1	350 kg	300 €	2 companies	

Figure 12: Agent Marketplace - Bidding Process Management Dashboard

In this interface, the user can explore details about previous bidding process that are closed (bidding history) or watch the bidding processes that are active right now. In this interface, the user can select the button 'Create Bid' in order to initialize a new online bidding process. Furthermore, the status bar for each bidding process is automatically updated during the process. The states are: Initialize -> Supplier selection -> Pick-up arrangement

- Initialize/create new bidding process

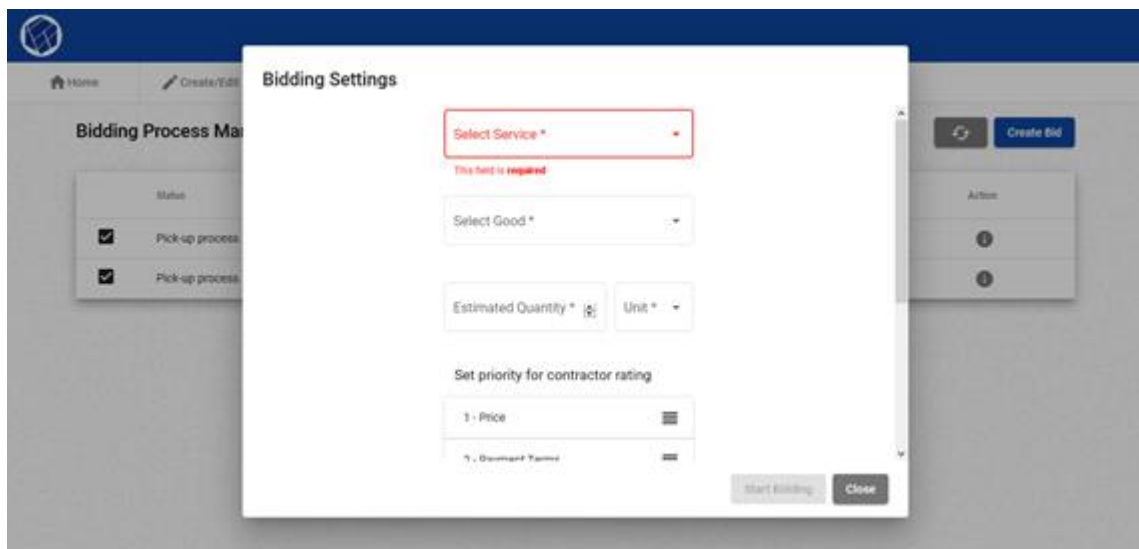


Figure 13: Agent Marketplace -Bidding Settings Popup I

In this menu the user can select a requested service, the corresponding good, to add the quantity and the unit of measurement, to set the priorities for his request and finally to start the bidding process. Next figures depict the available options for services, goods and supported priorities by the matchmaking/evaluation engine:

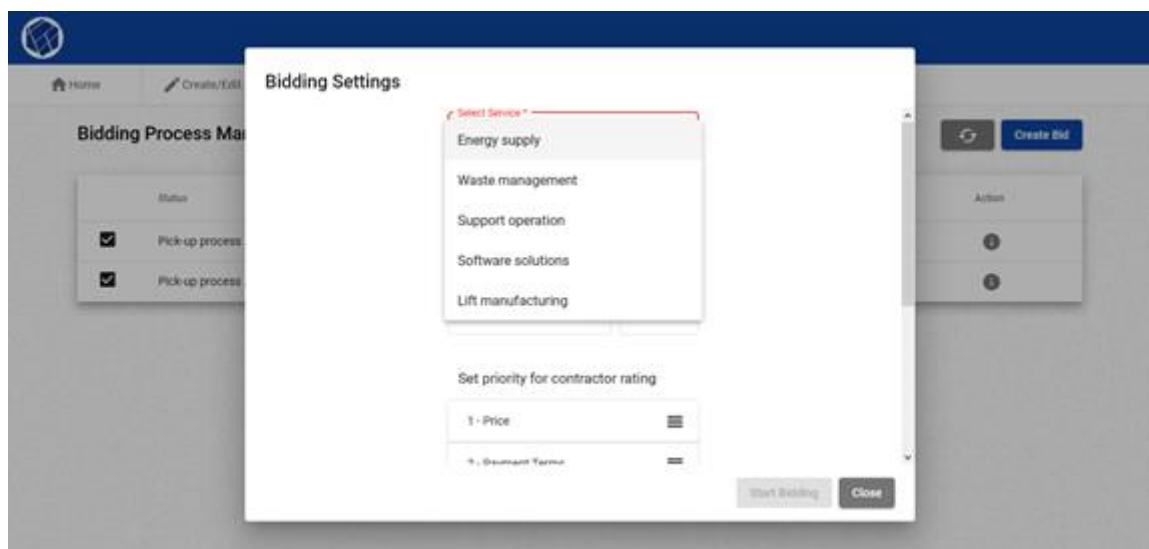


Figure 14: Agent Marketplace - Bidding Settings Popup II

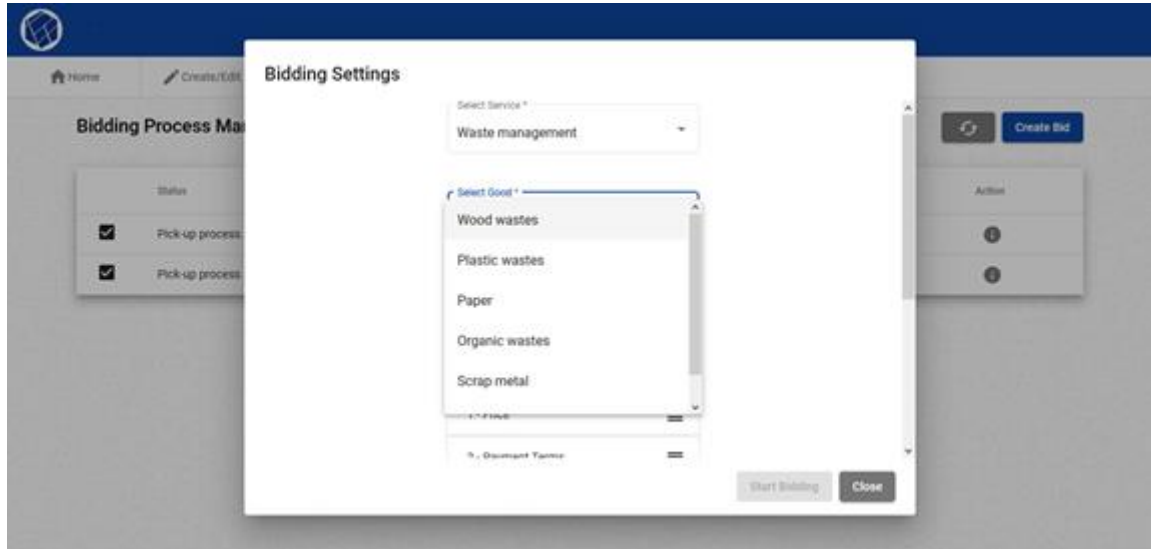


Figure 15: Agent Marketplace - Bidding Settings Popup III

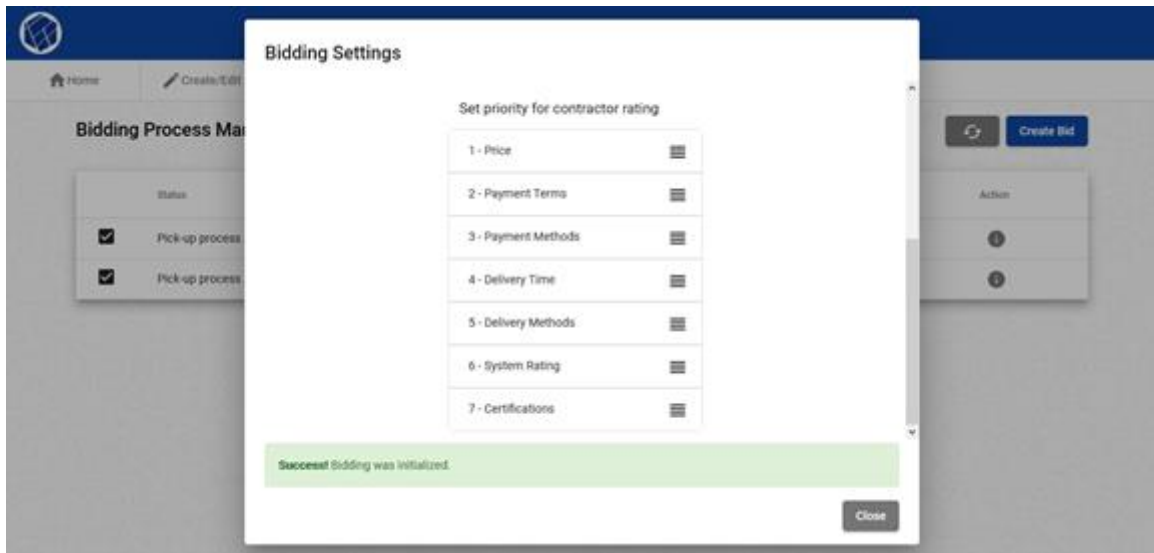


Figure 16: Agent Marketplace - Bidding Settings Popup IV

The Matchmaker component is available as an API. The component is an application for automated online bidding through agent-level and offer-level matchmaking. It is an Ontology based framework which applies semantic rules and SPARQL queries to the dedicated Ontology for requesters and suppliers straightforward matching and implements weighted criteria assessment for offer evaluation and best offer suggestion. The Matchmaker is connected with the Marketplace agents and stakeholders through RESTful web services and HTTP protocol. Figure 17 presents the information flow during the bidding process as it is originally designed during COMPOSITION and adopted for EFPF as well.

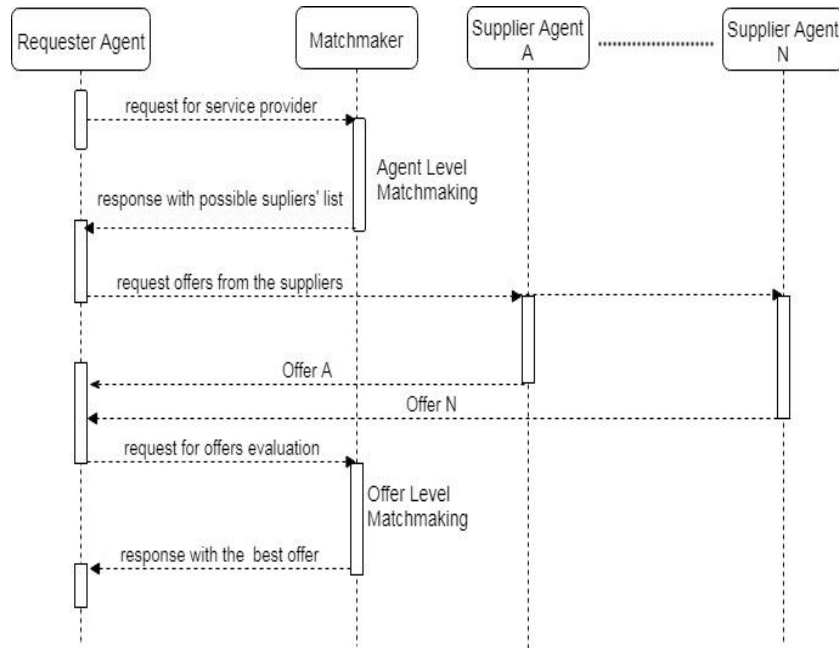


Figure 17: Agent Marketplace - Information Flow

More details about Matchmaker component are available on D5.11: Matchmaking and Intelligence Gathering.

The Agent Deployer provides a REST endpoint to be used by the UI in order to provide the customisation agent and deploying service. The actual endpoint implementation is described in Annex C: Data Models and Interfaces. The data model used by the API has been designed according to the needs of the integration process. It is composed by two main sections: one dedicated to new agent configuration and one to the matchmaker input data.

2.2.5 Limitations and Further Development

Further research and development related to agent framework and the bidding process will be focused on the delivery of the solution as a service through the EFPF portal to be fully customized and available to different users.

The agent communication mechanism will be aligned with other components communication behaviours, passing through a common Data Spine according to integration needs.

Moreover, the bidding process will be connected with the Blockchain app for asset and delivery process tracking that is under development in T5.4 Blockchain and Smart Contracting.

2.3 Accountancy Service

2.3.1 Scope and Relationship with Other EFPF Components

The Accountancy Service has been developed within the project as an integral part of the EFPF Marketplace Framework and provides insight into users' interactions with the EFPF Platform as well as its connected marketplaces. This includes transactions that EFPF users make on different marketplaces, which are linked with the EFPF Marketplace Framework.

A taxonomy has been setup to identify the trackable user actions in which action items are listed in 'subject, verb, object' manner and these actions include users' basic interactions with various parts of the EFPF Platform such as login, register, inviting other users as well as payments realized on external marketplaces if the user has initiated his/her journey from EFPF Marketplace. In this way, when a user performs a certain action on either EFPF Portal or a connected marketplace, corresponding information is sent to Accountancy Service to be persisted so that it can later be visualized to extract valuable information.

The Accountancy Service has been developed based on Elastic Stack which comprises the following components:

- **Elasticsearch:** Stores, indexes, provides and manages user logs to be later analysed. Since relational databases are not well-suited for managing log data, a NoSQL database like Elasticsearch is preferred due to their flexible and schema-free document structures, enabling analytics of the log data.
- **Logstash:** Gathers user behaviour data from various components of the EFPF platform, executes different transformations and filters the content, before sending the data to the Elasticsearch component
- **Kibana:** Enables interactive dashboards, filters and advanced data analysis and exploration of user logs.

In addition, the following custom modules listed below were developed to provide additional functionality:

- **Reporting Component:** Creates periodic (i.e. monthly) reports for each dashboard at the end of each month in PDF format and sends it as an email
- **Invoicing Component:** Processes all the payment data accumulated within each month, sums all the amounts from successful transactions realized on each marketplace, calculates a corresponding cashback amount and creates a detailed invoice with the information including purchased products, dates of transactions as well as the calculated commission for each product. The invoice will then be used to charge marketplaces.

2.3.2 Requirements and Realisation

Tracking the user behaviour enables businesses to make productive decisions and develop effective business strategies. This is a valuable feature of a federated digital platforms and the Accountancy Service provides this functionality to support the long-term sustainability of the EFPF platform, beyond the span of the project. The Accountancy Service also aims to track and trace a user's journey across the EFPF ecosystem and collect data about the transactions they make on different marketplaces. The collected data logs are then used by a cashback mechanism that enable a commission charge or a referral fee to be made to the marketplace where a EFPF user carries out a business transaction (Figure 18). In addition to the log collection, customizable dashboards are also needed for better and easier tracking of user interactions. Therefore, in order to process the accumulated log data and address all the above requirements, the Accountancy Service uses Elastic Stack (Elasticsearch, Logstash, Kibana) as an advanced log persistence, monitoring, processing, and visualization framework.

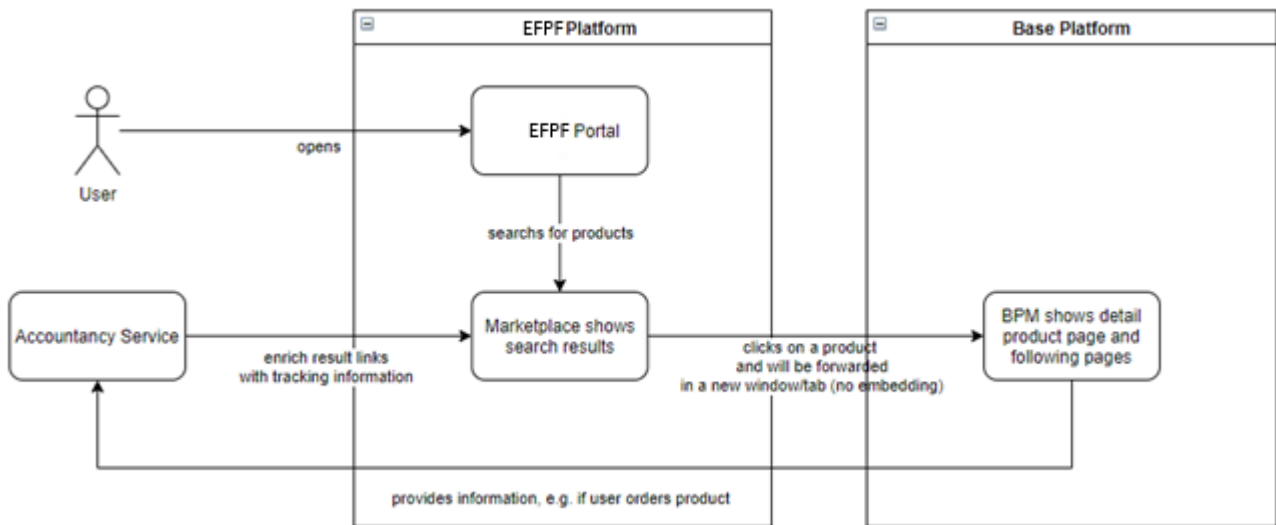


Figure 18: Accountancy Service - Cashback Process

2.3.3 Deployment

Accountancy Service is a standalone component that runs independently of existing EFPF tools and services and can be integrated with unlimited number of external marketplaces. Each of the 5 components of the Accountancy Service has its own Docker image and runs on a corresponding Docker container. Moreover, a production-ready configuration allowing Elasticsearch running on a multi-node cluster is also available. Currently, all the components of the Accountancy Service run on a production server hosted by SRDC.

2.3.4 Execution and Usage

Accountancy Service uses Logstash as a data ingestion and server-side data processing pipeline. The logs sent to Logstash are forwarded to Elasticsearch for persistence after executing certain ingestion pipelines; and then Kibana dashboards are automatically updated based on the certain fields stored on Elasticsearch. In other words, Accountancy Service uses a basic data model for visualization and logs must conform to a basic data model so that Kibana dashboards can be updated automatically.

Currently, there is a running instance of Logstash component that is publicly available through a public endpoint so that events from EFPF Portal and external marketplaces can be sent using HTTP POST method. Events are modelled as a JSON message related with the action conforming to the data model. This will be enough for the Accountancy Service to capture the data and update the dashboards.

Accountancy Service uses Kibana to visualize the data ingested through Logstash and persisted on Elasticsearch. Kibana provides mechanisms to create interactive dashboards with filtering as well as advanced data analysis capabilities and Accountancy Service provides 4 different dashboards for log visualization. Furthermore, all these dashboards can either be accessed on the Kibana instance hosted by SRDC and EFPF Portal Admin pages. Details of the 4 dashboards can be seen below:

- **Payments Dashboard:** Displays all payments realized on marketplaces as well as the corresponding cashback (commission) amounts calculated for each transaction - Figure 19

- **Marketplace Usage Dashboard:** Visualizes marketplaces usages in terms of most frequently used search keywords, queried platforms, and their distribution - Figure 20
- **Platform Engagement Dashboard:** Displays base platform visits and tool/service usages and tracks the frequency of these usages - Figure 21
- **User Activities:** Visualizes user actions such as login and register - Figure 22

In addition to the central functionalities offered by the Elastic Stack, Accountancy Service also provides extra features such as preparing monthly reports based on the accumulated data and generate invoices in accordance with the commissions calculated for successful transactions that users perform on the external marketplaces connected to the EFPF Platform. Since these functionalities require custom implementations, these add-on modules are written in JavaScript and provided as a Docker image. At the end of each month, both modules processes relevant data of the target month and generates 2 PDF documents: One for the monthly report including all charts updated for the target month and one for the invoice document containing all the transaction details (e.g. purchased products, their prices, transaction dates, calculated commission, etc.) to charge each external marketplace.

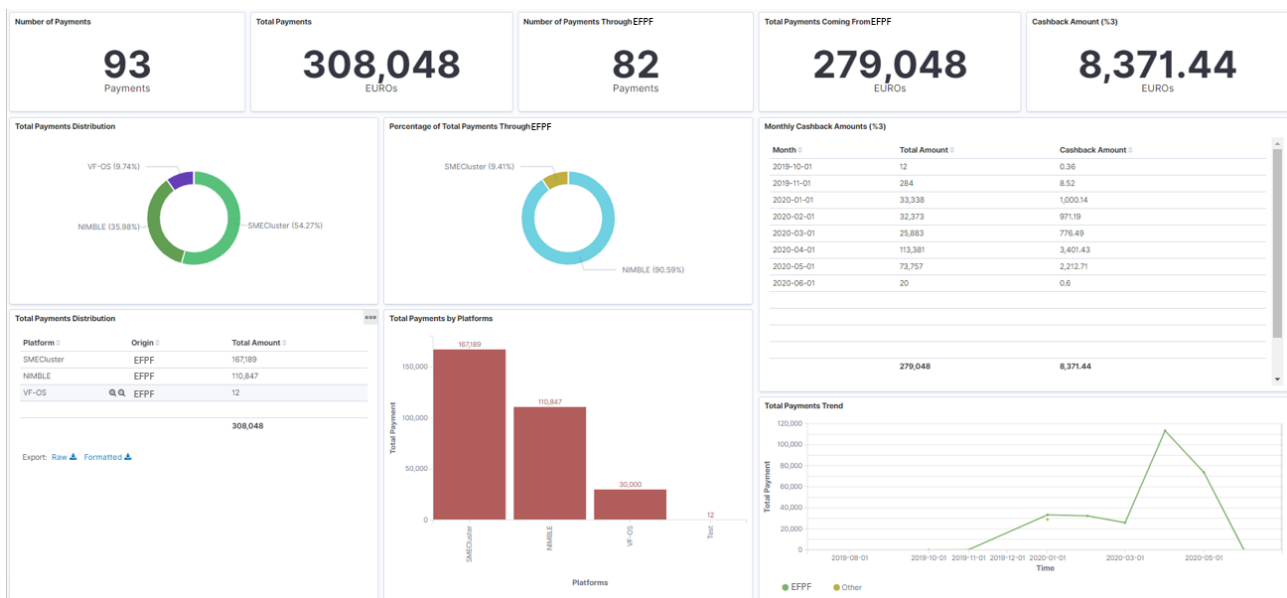


Figure 19: Accountancy Service - Payments Dashboard

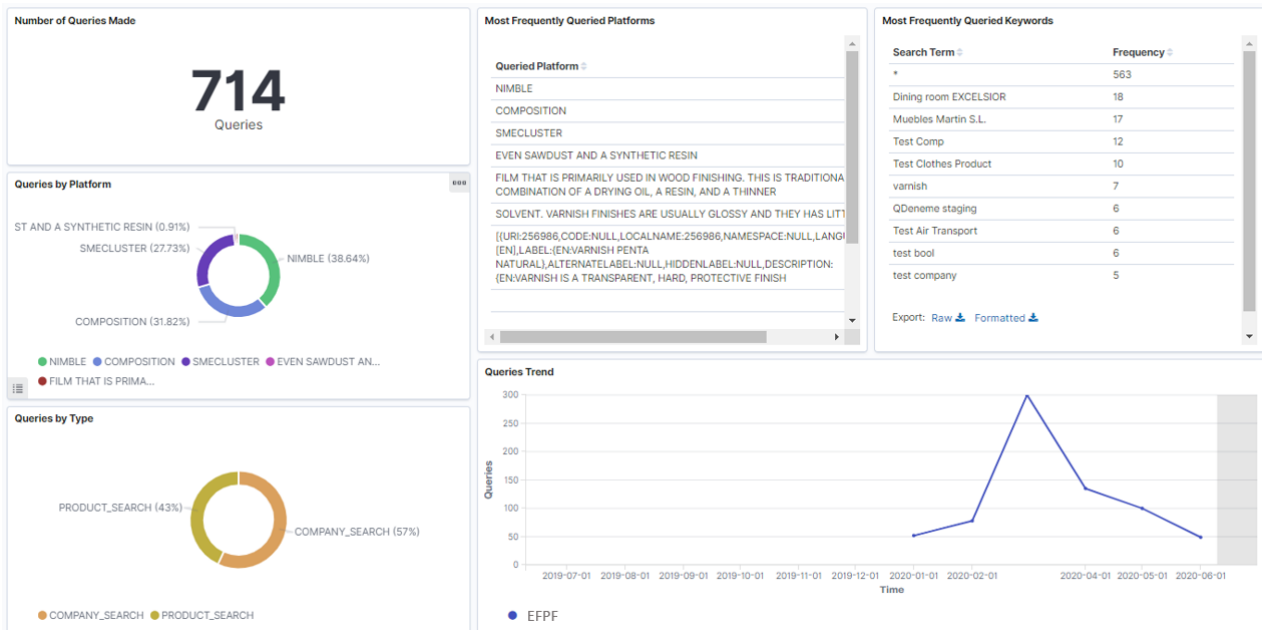


Figure 20. Accountancy Service Marketplace Usage Dashboard

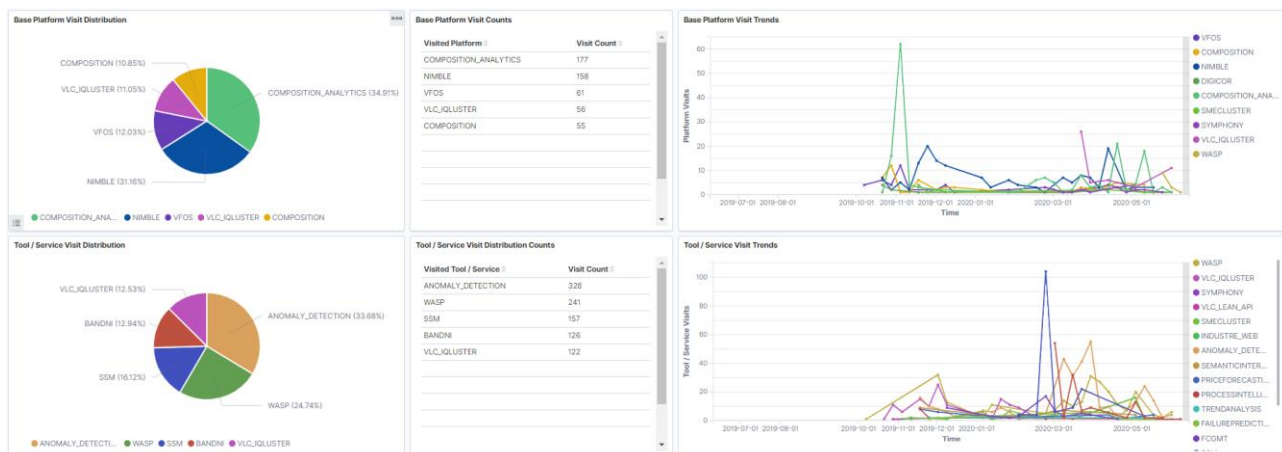


Figure 21: Accountancy Service - Platform Engagement Dashboard



Figure 22: Accountancy Service - User Activities Dashboard

2.3.5 Limitations and Further Development

In the future releases of the Accountancy Service, its integration with the Service Registry will be realised. In this way, Logstash endpoint of the Accountancy Service will be discoverable by external marketplaces through the Service Registry and new marketplaces will be integrated easily. Moreover, new dashboards will be created upon requests to track different types of user interactions.

3 EFPP Portal

The EFPP Portal is the unification point of distributed tools and platforms in the EFPP ecosystem. It allows the users to access connected tools, base platforms, marketplaces, experiments, and pilots through a unified interface.

3.1 Current Status

This section provides an account of the current status of EFPP portal and describes a typical EFPP user journey through the portal.

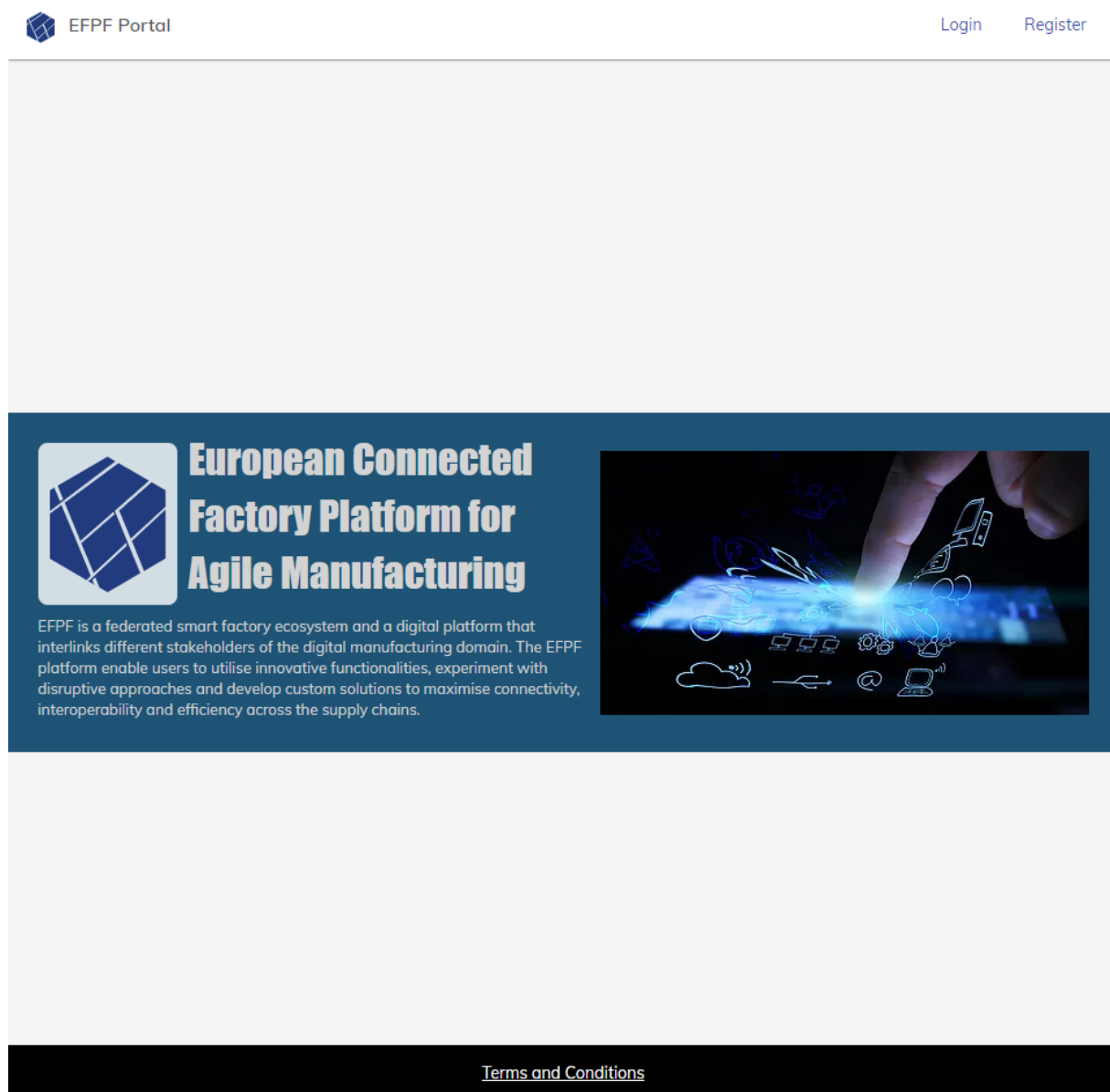


Figure 23: EFPP Portal - Landing Page

The EFPP landing page can be accessed by following this link - <https://efactory-portal-dev.ascora.eu/>. The landing page provides a brief description of this project and explains the breadth of possibilities to a perspective user. The page allows a user to Login to EFPP portal or Register, if it is their first time on the platform. The landing page also provides the user with EFPP Terms & Conditions, so that they can read through them and ensure that it

aligns with their expectations before joining. The rest of the user journey is explained through separate sections below.

3.1.1 Register

First time visitors can click on ‘Register’ button to sign up to the platform. Upon clicking on Register, a simple and quick registration form appears on the screen as shown in Figure 24.

EFPP Portal

Login Register

Create an Account for EFPF

Note: Non-EFPF members will be authorized to access the portal starting from the beginning of November.

Firstname Lastname

Company VAT Identification Nu...

Postal Code City

Street Country

Sector E-Mail

Password Password Confirmation

☐ I read and accept [terms and conditions](#)

☐ I accept storage in [EFPF Blockchain](#)

Figure 24: EFPF Portal - Registration Form

Once this form is filled in, a user account will be set up and a welcome email will be sent to the new user providing general information about the EFPF portal and the next steps. As the new user account is being disabled by default, an admin will have to check the user account and the provided information first. If the user account is valid, the admin enables it and the user will be able to log in. On the first login an email validation will take place.

3.1.2 Login

A returning user can straight away click on ‘Login’ button from the landing page. The user is then redirected to the login credentials window (Figure 25) where they can enter their login credentials and get access to the platform.

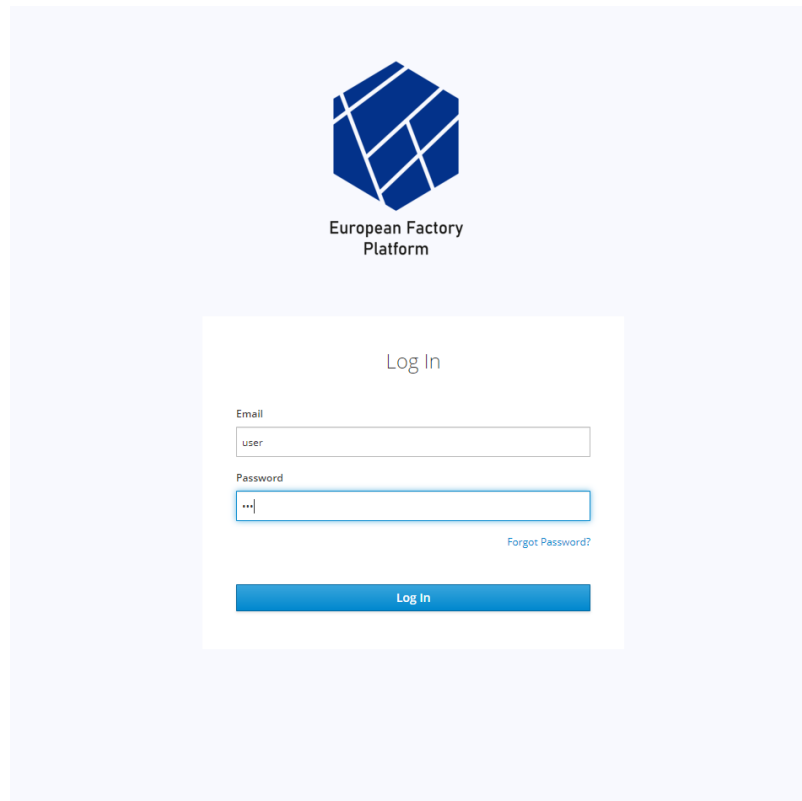


Figure 25: EFPF Portal - Login Screen

3.1.3 Dashboard

After the user enters the portal, they are greeted by the dashboard. The dashboard has been designed with a view to showcase the breadth of capabilities available in the federation. The presentation of the dashboard has been adapted to bring out customer benefits, so that they can understand the application and usefulness instantly, without going into the details of where this functionality is coming from in the EFPF federation.

A direct link to access the marketplace is provided for those specifically looking to explore the integrated marketplace framework.

Further an array of tiles each pointing to a different value propositions of the EFPF federation are shown in the centre of the dashboard. The aim is to provide the user ease of navigation by bundling supplementary tools and services under high level value propositions. Figure 26 shows the overall layout of the dashboard.

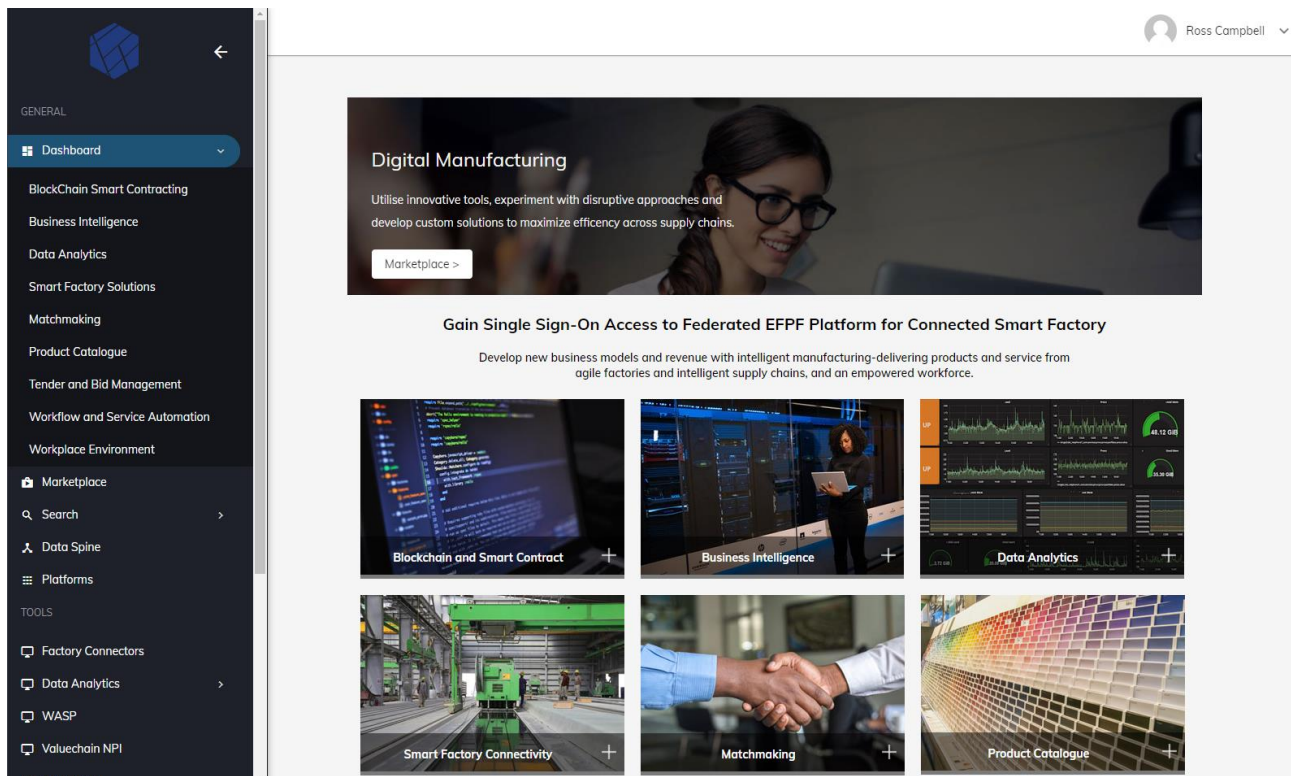


Figure 26: EFPP Portal - Dashboard

There is a side panel available to the users throughout the portal which provides general information about the user and then a series of shortcuts to a few key tools and platforms. The user can go straight to those tools by using the left-hand pane at any moment while on the platform.

Finally, at the bottom of the dashboard page 5 key links are provided, the use and application of which are described below:

- **Contact Us:** Intended to provide users with contact details of the governing body (in this case, European Factory Foundation) including contactable email, phone number and other relevant details.
- **Find a Partner:** This links the user to the company matchmaking search, where users filter and search for companies based on multiple attributes like sectors and business types.
- **Support:** This link will be forwarding the user to a support page, which enables users and developers to get in contact with the platform support team.
- **EFPP Ecosystem:** This links the user to European Factory Foundation website. This foundation will be the ultimate owner of the EFPP platform after the EFPP project. The foundation website can also be reached by following this link - <http://ef-foundation.com/>.
- **Public Portal:** This links the user to the public website for the EFPP project. This website can also be reached by following this link - <https://www.efpf.org/>. The project website aims to disseminate project's efforts through regular updates on the technical progress, shows attended and constant blog posts.

In order to describe the unique value propositions (VP) offers to the users, the dashboard provides direct links to the pages that provide details and access to the VP in the EFPP

ecosystem. Each VP page describes what's on offer and presents actual solutions to bring out the range of possibilities. The list of VPs has been designed and refined in consultation with technical partners to provide detailed information about the possibilities of the platform and how users can engage with the different solutions. Following this approach, individual VP pages were created and organised in a simple to understand fashion in the portal.

Initially the VP webpage explains overall ambition of the solution in a few lines. It is then followed by a concise description of the key benefits as well as implementation scenarios by project pilot partners and their experience.

Blockchain and Smart Contracting
Seamless connectivity and secure data exchange for a new level of digital transformation in dynamic industrial supply chains

Reduce cost and establish trust in the delivery process
A flexible and open ecosystem that uses the blockchain based distributed ledger technology to provide a transparent, secure and reliable log of the distributed activities. The blockchain technology allows new actors to easily join the existing supply chains.

Provides transparency support for the Circular Economy
Blockchain provides a secure platform for transparent information exchange among different stakeholders in the circular economy. The blockchain platform also provides a trusted and cost-effective way to exchange business documents where the origin and integrity of data is preserved.

Manufacturing companies like KLEEMANN, ELDA and MIL Oil are using the blockchain technology to track and trace the flow of materials in closed-loop supply chains. The use of blockchain technology allows the companies to determine the status of their shipments in real-time e.g. where is a specific shipment at any moment and who is handling it in the supply chain. The eFactory blockchain platform provide mobile app and web-based user interfaces that allow users to make use of the rather complex blockchain technology with much ease. Using the mobile app, the shipment handlers can record the handover processes (with relevant information, including multimedia-based evidence) in the distributed ledger for future reference. The use of blockchain technology in the circular economy offers new possibilities for data-driven business models in the industrial sectors.

Blockchain-based Solutions for Industrial Supply Chains

Delivery process support
The blockchain-based delivery tracking solution uses a distributed ledger for recording transactions and events. Smart contract process logic and a DApp framework is developed to provide a configurable system for monitoring delivery processes and verifying events in the supply chain. Users can define a delivery process based on the pre-defined handover procedures and verify identity of actors and products using mobile units, QR codes and NFC. The web interface and DApps can be used to manage and monitor the status of active processes. This solution provides a flexible open way to ensure trust, lower handling costs and reduce errors in the delivery chain.
[Get Started >](#)

Circular economy support

Figure 27: EFPF Portal - Value Proposition Page Example

3.1.4 Federated Search

The federated search component, which has been developed and provided by T4.5 - Matchmaking and Agile Networks Creation, allows the user to search for two different types of entities:

- Products (See Figure 28)
- Companies (See Figure 29)

More information regarding this component can be found in deliverable D5.11: EFPF Matchmaking and Intelligence Gathering.

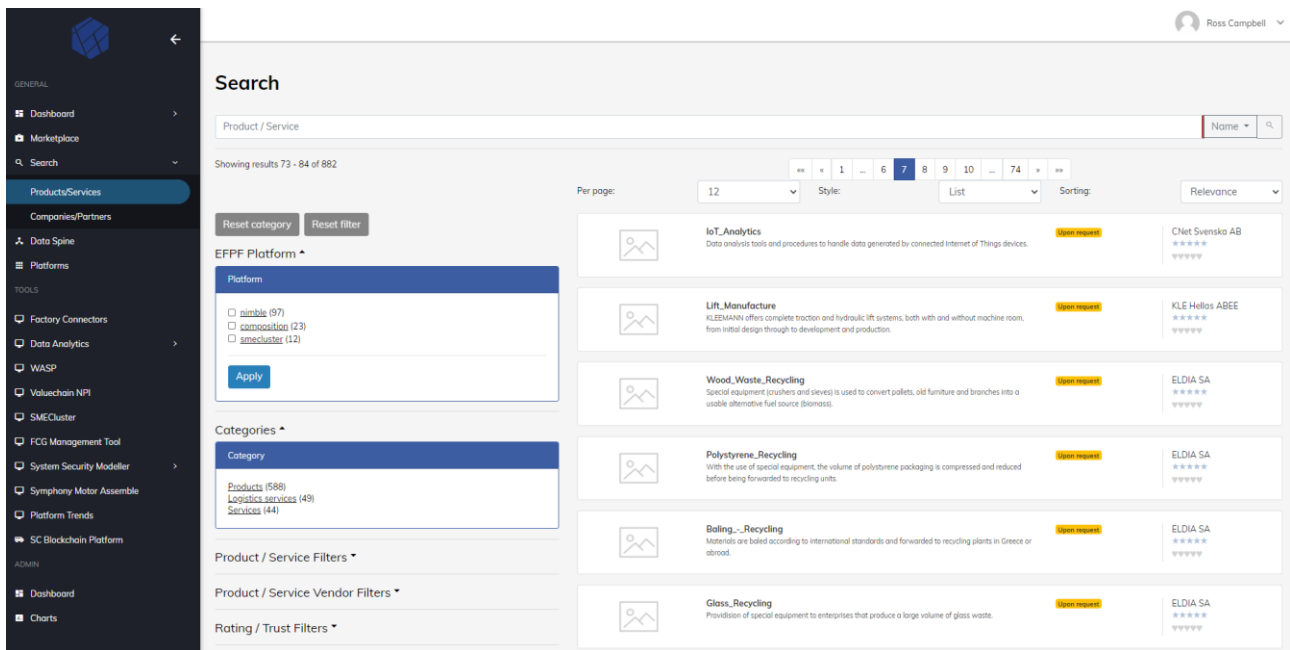


Figure 28: EFPF Portal - Federated Product Search

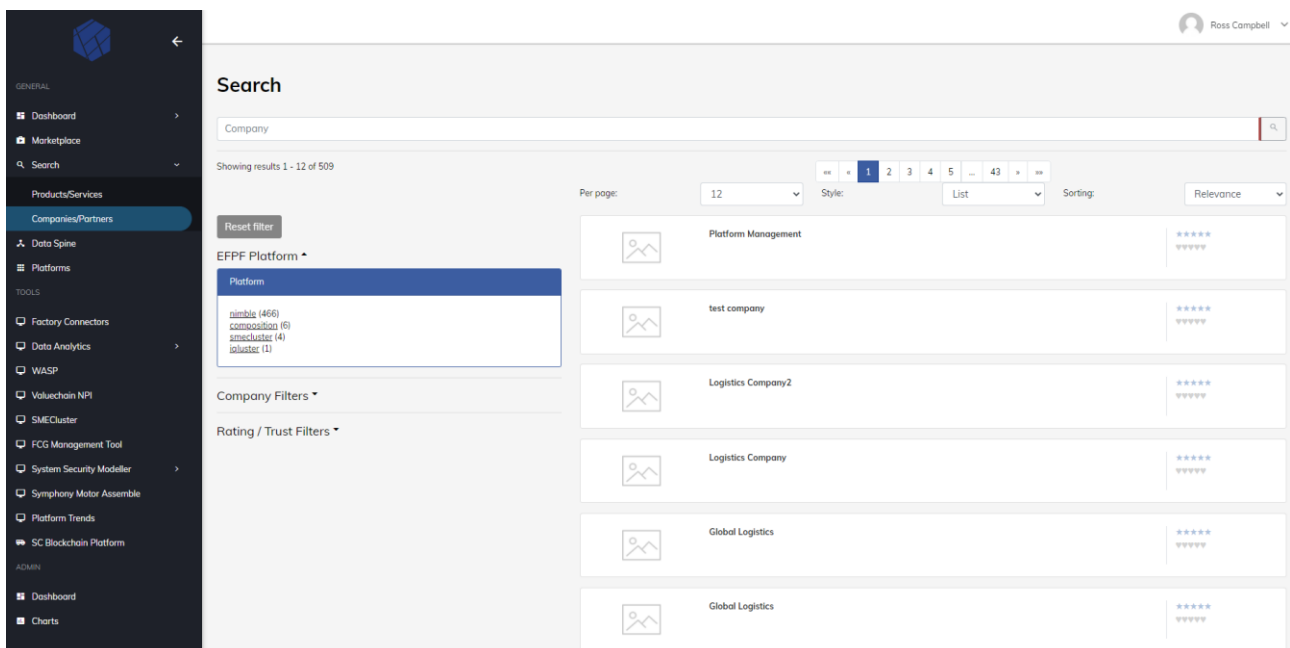


Figure 29: EFPF Portal - Federated Company Search / Matchmaking

3.2 Scope and Relationship with Other Components

The EFPF Portal component provides the main user interface of the EFPF platform. It is being connected to other components in the EFPF platform. The following list provides an overview of the relationship regarding connected components and a description of the functionality it enables:

- **Identity and access management component EFS:** This component secures access and communication of the EFPF portal.
- **Smart contracting component:** This component is being used at user registration time and enables the storage of registration information, i.e. does the user agreed to the EFPF Terms and Condition.
- **Accountancy Service component:** This component is being used to track user events like visiting an external platform or conducting a product search.
- **External mail service:** At this time the EFPF Portal backend uses an external mail service to send out notification or confirmation emails, i.e. when a user registered successfully a welcome email will be sent out with the next steps.

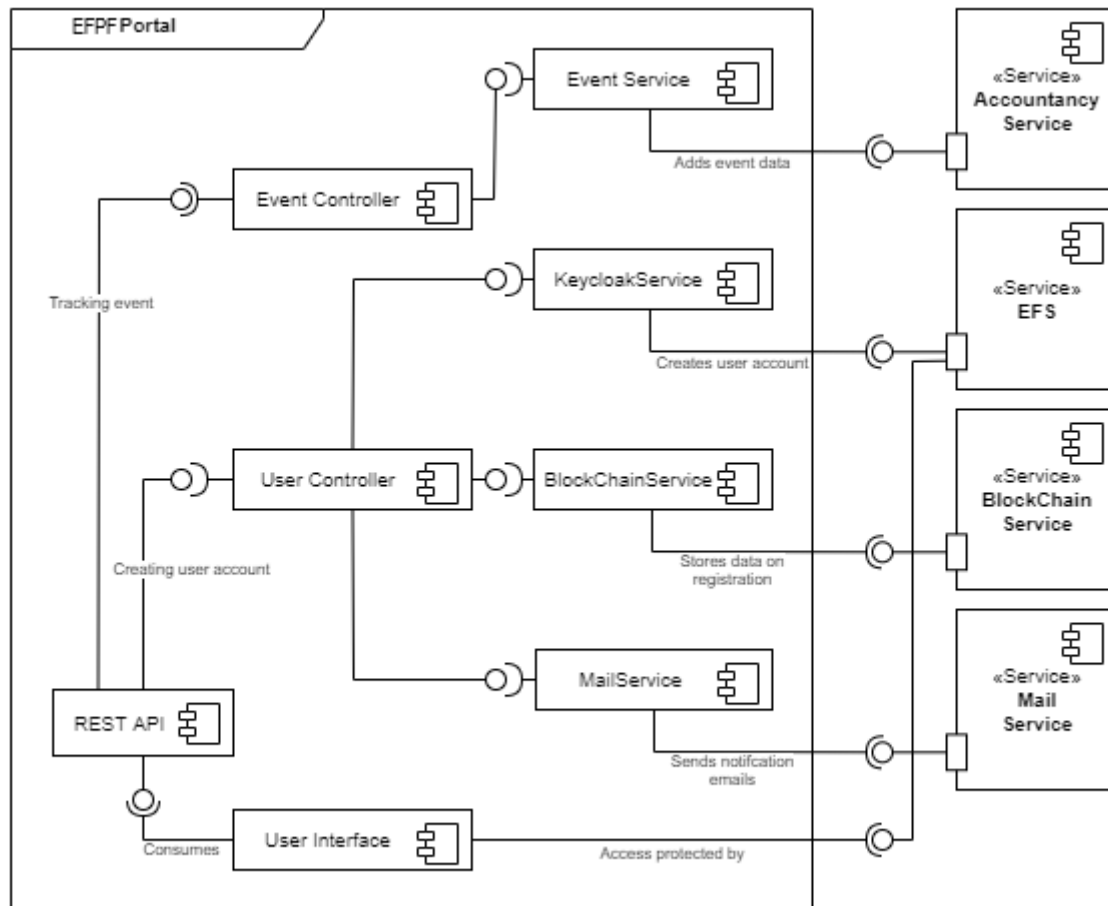


Figure 30: EFPF Portal - Architecture Diagram

3.3 Requirements and Realisation

The EFPF Portal component is divided in two components. The first component is the EFPF Portal frontend, which is a single-page application (SPA) based on the Angular web application framework⁴. The second component is the EFPF Portal backend, which is responsible for providing the required data for the frontend. It handles also different tasks like registration and event logging.

⁴ <https://angular.io/>

Both portal components are provided as Docker images, which enables a fast and uncomplicated way to deploy them. It is recommended to use the Linux distribution Ubuntu 18.04 LTS as the operating system with Docker Engine v19.03 installed. Any other operating system can be used too, but it cannot be guaranteed that it will work successfully with the Docker version.

3.4 Deployment in the EFPF Platform (Installation)

This section provides information on how the EFPF Portal components can be deployed.

Note: Due to the current server situation, this section provides a preliminary manual on how to deploy the Docker images. Once the different server environments for the EFPF platform have been set up, this section will contain step-by-step information for deploying these specific components in the providing server environments.

Both components (Frontend and backend) are provided as Docker images build by the CI/CD feature of GitLab⁵. GitLab stores Docker images in an integrated Docker registry, to which a Docker Engine can connect and retrieve the required Docker images. As the Docker registry is not public, the Docker Engine installation first must be connected to the Docker registry. This can be done by the Docker login⁶ command.

\$ docker login --username foo --password-stdin registry:8080

As the server environment may use portainer⁷ for Docker container management, a connection to the GitLab registry does not require a manual login but will be able through UI configuration.

3.5 Execution and Usage

Both the backend and frontend component will run directly after the deployment. The REST APIs of the backend component are documented using Swagger⁸ and provide information on how the frontend is communicating with the backend.

The frontend can be access via browser. The landing page (Figure 23) will be shown by default and currently provides basic functionality which has been described in Section 3.1. After a successful login, the user will be forwarded to the Dashboard and can use the provided tools and services of the portal.

As access can change due to systems like CloudFlare and other routing components, so information about how to access the EFPF Portal frontend will be updated after the final deployment environment has been set up.

3.6 Limitations and Further Developments

The following limitations for the EFPF Portal exist:

- Users cannot change profile and company details after the registration.
- Registration process requires manual steps

⁵ <https://about.gitlab.com/>

⁶ <https://docs.docker.com/engine/reference/commandline/login/>

⁷ <https://www.portainer.io/>

⁸ <https://swagger.io/>

Future development will include the following functionalities:

- Limit and manage access to sections based on roles
- Improve user experience for new members
- Provide user profile management
- Provide use case pages
- Working towards a more integrated portal

4 Conclusion and Outlook

This deliverable presents the requirements for a federated ecosystem, its evolution and extension. These requirements are supported by the required governance mechanism that are being developed in the project – more details of Governance mechanisms are available in D5.12. The requirements discussed in this deliverable provide necessary conditions that should be considered for creating a sustainable federated digital platform and a participatory ecosystem that can grow over time to address the changing needs of different stakeholders in the digital manufacturing domain.

The deliverable also introduces an integrated marketplace framework that has been implemented in the EFPF project to enable the interlinking of different marketplaces in the EFPF federation and providing access to their offerings through a unified interface. The integrated marketplace interface, accessible through the EFPF portal, allows users to explore the offerings of multiple external marketplaces. Since the external marketplaces are connected via SSO, the users do not have to create additional accounts and can continue in their user journey seamlessly from EFPF portal to other marketplace and platforms in the federation. In addition, an automated agent-based marketplace has been implemented and will be integrated into the EFPF portal in the near future.

To support a sustainable business model, an Accountancy Service offers track and trace of user journeys across the EFPF federation. If an EFPF user makes a transaction on any of the linked marketplaces, the Accountancy Service records that activity and generates monthly invoices based on the predefined commission or sales-fee agreements between the linked marketplaces and EFPF platform.

Finally, the EFPF portal provides login and registration so new users can access the portal. Tools and services provided by project partner are listed inside the portal. The value proposition pages provide information for solutions, which the EFPF platform offers. Future tasks are to provide a more unified and seamless design.

Annex A: History

Annex B: References

Annex C: Data Models and Interfaces



**European Factory
Platform**

www.efpf.org