Master thesis

Developing a graphical user interface to a furniture digital product passport in data spaces

carried out for the purpose of obtaining the academic degree of a

Diplom-Ingenieur

under the supervision of

Univ.-Prof. Dr.-Ing. Sebastian Schlund
(E330 Institute of Management Sciences, Department: Human-Machine Interaction)

Univ.Lektor Dipl.-Ing. Dr.techn. Arko Steinwender
(Fraunhofer Austria Research GmbH, Department: Factory planning and production management)

Viola Gallina, Phd
(Fraunhofer Austria Research GmbH, Department: Factory planning and production management)

submitted to the Vienna University of Technology

Faculty of Mechanical Engineering and Management Sciences

by

Baris Tekin BSc

[redacted]

Vienna, September 2025

Baris, Tekin BSc

I have taken note that I am authorised to submit my thesis for printing under the title

Diploma thesis

only with the authorisation of the examination board.

I further declare in lieu of oath that I have completed my thesis independently in accordance with the recognised principles of academic writing and that I have cited all the resources used, the literature on which it is based.

Furthermore, I declare that I have not previously submitted this thesis topic in any form as an examination paper either in Austria or abroad (to an assessor for assessment) and that this thesis corresponds to the thesis assessed by the assessor.

Vienna, September 2025

Baris, Tekin BSc

Acknowledgements

I would like to warmly thank all who supported me during the completion of this thesis.

My special thanks go to my principal supervisor Univ.-Prof. Dr.-Ing. Sebastian Schlund for his academic guidance and valuable support during this work. I would also like to thank Univ. Lektor Dipl.-Ing. Dr.techn. Arko Steinwender and Viola Gallina, PhD for their dedicated supervision, constructive feedback and insightful discussions.

I would also like to thank Nexyo GmbH for their technical support and for supplying the resources vital for this thesis.

I am deeply grateful to my family, especially my parents, my grandparents and my brother, who has been by my side throughout the entire journey. From the beginning of our bachelor studies to the completion of the master's degree, he has supported me in every step.

Thank you all for your contributions, encouragement, patience and continuous support.

Kurzfassung

Die Kreislaufwirtschaft verfolgt das Ziel, Ressourcenverbrauch, Abfälle, Schadstoffe und Treibhausgasemissionen zu reduzieren, um eine nachhaltige Entwicklung zu fördern. Für eine erfolgreiche Umsetzung sind sektorübergreifende Ansätze und der gezielte Einsatz verschiedener Strategien, Maßnahmen und digitaler Technologien erforderlich. Die Digitalisierung spielt dabei eine entscheidende Rolle.

Ein wichtiges digitales Werkzeug ist der digitale Produktpass, der transparente Produktinformationen über den gesamten Lebenszyklus bereitstellt. Eine effektive Visualisierung ist dabei unverzichtbar, um komplexe Daten verständlich darzustellen.

Diese Arbeit beschäftigt sich mit der Gestaltung und Entwicklung einer modularen, flexibel einsetzbaren grafischen Benutzeroberfläche sowie einer Datenschnittstelle, die den anwendungsorientierten und unternehmensübergreifenden Datenaustausch im Rahmen des digitalen Produktpasses ermöglichen. Als Anwendungsbeispiel wird die Möbelindustrie betrachtet, wobei insbesondere die Rückverfolgbarkeit von Holzmaterialien im Fokus steht.

Nach umfassender Analyse bestehender Architekturen, Identifikationssysteme, Datenträger und Datenmanagementkonzepte wurde eine universell übertragbare Benutzeroberfläche entworfen.

Der Zugang zum digitalen Produktpass erfolgt über das Scannen eines QR-Codes, der direkt zur Benutzeroberfläche führt. Das Backend optimiert die Datenstruktur für eine effiziente Speicherung und Abfrage, sichert die Interoperabilität verschiedener Systeme und bietet eine skalierbare Systemarchitektur. Für den Datenaustausch wurde ein dezentraler Datenraum implementiert, der hohe Datenintegrität und die ständige Verfügbarkeit aktueller Informationen gewährleistet.

Die Benutzeroberfläche liefert detaillierte Produktdaten, Leistungskennzahlen zur Bewertung der Zirkularität und ermöglicht die Nachverfolgung der Materialien über den gesamten Produktlebenszyklus. Das Frontend sorgt für eine übersichtliche und benutzerfreundliche Darstellung der Informationen und vermeidet eine Überladung der Oberfläche durch eine klare und intuitive Gestaltung. Zur Bewertung der Benutzerfreundlichkeit wurden standardisierte Fragebögen eingesetzt, darunter die System Usability Scale und der User Experience Questionnaire.

Nach Abschluss der laufenden Standardisierungsaktivitäten werden die offiziellen Datenformate für den digitalen Produktpass veröffentlicht. Durch eine entsprechende Anpassung des Backends an diese Vorgaben kann die Lösung anschließend branchenübergreifend für alle Produktgruppen eingesetzt werden.

Abstract

The circular economy aims to significantly reduce resource consumption, avoid waste, decrease pollutants and lower greenhouse gas emissions in order to promote sustainable development. Successful implementation requires cross-sectoral approaches and the targeted use of various strategies, measures and digital technologies. Digitalisation plays a crucial role in this process.

An important digital tool is the Digital Product Passport, which provides transparent product information throughout the entire life cycle. Effective visualization is essential to present complex data in an understandable way.

This thesis focuses on the design and development of a modular, flexibly deployable graphical user interface and a data interface that enable application-oriented and cross-company data exchange within the framework of the Digital Product Passport. The furniture industry is used as a case study, with a particular focus on the traceability of wood materials.

After a comprehensive analysis of existing architectures, identification systems, data carriers and data management concepts, a universally transferable user interface was designed.

Access to the Digital Product Passport is granted by scanning a QR-Code that leads directly to the user interface. The backend optimizes the data structure for efficient storage and querying, ensures interoperability among different systems and offers a scalable system architecture. For data exchange, a decentralized data space was implemented to guarantee high data integrity and continuous availability of up-to-date information.

The user interface provides detailed product data, key performance indicators for assessing circularity and enables tracking of materials throughout the entire product life cycle. The frontend ensures a clear and user-friendly presentation of information and avoids interface overload through a straightforward and intuitive design. Standardized questionnaires, including the System Usability Scale and the User Experience Questionnaire, were used to evaluate usability.

Upon completion of ongoing standardization activities, official data formats for the Digital Product Passport will be published. By adapting the backend accordingly, the solution can then be deployed across industries, covering all product groups.

Table of Contents

1 Introduction

1.1 Initial Situation

In recent years, the concept of the Circular Economy (CE) has emerged as a transformative model for sustainable development. The linear economy operates on a "take, make, use, dispose" approach. Raw materials are extracted from the environment ("take") to create various products and generate energy ("make"). After a relatively brief period of use ("use"), most of these products eventually become waste and emissions, contaminating air, water and soil and reentering our ecosystems ("waste"). Minimizing resource consumption, waste and emissions is crucial for long-term ecological balance and environmental preservation. This goal calls for a fundamental shift, from the traditional "take-make-use-dispose" linear economy to a CE. In a CE, raw materials are sourced sustainably and products are designed to minimize resource consumption and waste. The aim is to extend the lifespan of products and optimize their use. At the end of their life, products are reintegrated into the production cycle to maintain their value. The focus is on reducing resource use, prolonging product life and encourage reuse. Recycling should be the final option, pursued only when repair or reuse is no longer possible (Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology, 2022a).

The Figure 1 “Guiding principle of the CE” illustrates the key processes of a sustainable CE and emphasizes the transition from linear to circular economic models. It depicts the flow of primary resources through the phases of production, trade and consumption, highlighting the critical role of return systems to reintegrate materials into the cycle. Secondary resources are generated through recycling. The lower part of the Figure 1 presents the waste hierarchy, with waste prevention as the top priority, followed by preparation for reuse, recycling, other recovery and finally disposal. The closed material flows and the prioritization of measures underline the objective of sustainable resource use and the minimization of waste in alignment with the principles of the CE (Federal Ministry Republic of Germany for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection, n.d.).

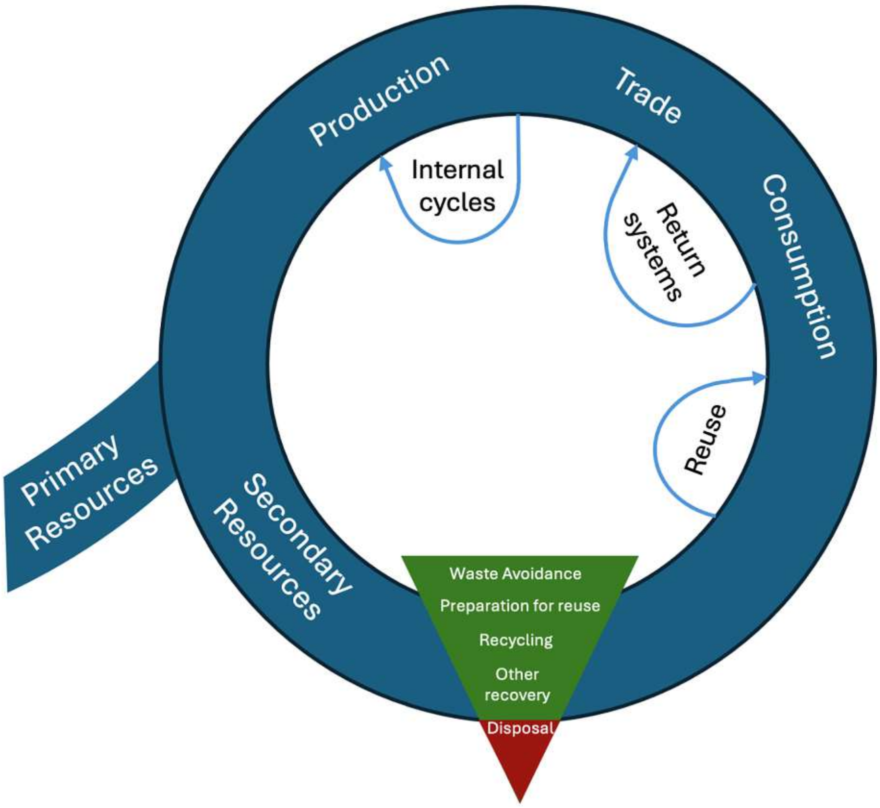


Figure 1: Principle of the CE (own illustration based on (Federal Ministry Republic of Germany for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection, n.d.))

Designing systems, business models, services and products to be fully circular from the outset ("circular by design") is essential for minimizing resource consumption and fostering long-term sustainability. To achieve this goal of reducing resource consumption and promoting sustainability, the CE relies on a set of guiding principles that provide a framework for designing and implementing circular systems. These principles, commonly referred to as the R-principles, outline the key strategies for maximizing resource efficiency, minimizing waste and ensuring products and materials remain in use for as long as possible. The R-principles are (Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology, 2022a):

- Refuse: Products are replaced by alternative solutions that offer the same benefits, thereby avoiding the consumption of unnecessary resources
- Rethink: Products are reimagined and designed in a circular way, with a focus on their more intensive use
- Reduce: Resource consumption in the production is reduced by increasing efficiency and reducing the use of natural resources and materials
- Reuse: Functional products are reused without major adjustments
- Repair: Products are repaired to extend their lifespan
- Refurbish: Old products are refurbished and updated to the latest status

- Remanufacture: Parts from defective products are used in new products that perform the same functions
- Repurpose: Parts from defective products are used in new products for different purposes
- Recycle: Materials are processed to ensure high quality so they can reenter the material loop
- Recover: Non-reusable materials are thermally processed for energy recovery

The R-principles approach provides a comprehensive foundation for transitioning to a more sustainable and resource-efficient economy.

The objectives of the CE strategy are (Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology, 2022b):

- Significantly reducing resource consumption and usage
- Eliminating waste (Zero Waste)
- Preventing pollution from harmful substances (Zero Pollution)
- Lowering Greenhouse Gas Emissions (GHGs)

A successful transition to a CE requires a multi-sectoral approach and the implementation of a strategic mix of various tools and measures. The key intervention areas in Austria's CE strategy are as follows (Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology, 2022a):

- Legal and regulatory framework conditions
- Smart market incentives
- Financing and funding
- Research, Technology Development and Innovation (RTDI)
- Digitalisation
- Information, knowledge and cooperation

Digitalisation plays a key role in this transformation, aiming to position nations as leading digital hubs while securing long-term prosperity, employment and quality of life. Digital technologies and solutions also serve as crucial enablers of the CE (Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology, 2022a).

One of the major challenges in advancing the CE is the lack of consistent and precise data as well as limited information sharing among stakeholders, which hampers transparency and collaboration across sectors. Closing this gap through digitalization across the entire product lifecycle is essential to bridge the divide between the concept of the CE and its practical implementation (Walden et al., 2021).

A crucial component in this digital transformation is the Digital Product Passport (DPP), which provides transparent, accessible and comprehensive product information. Through the DPP, stakeholders can access detailed data that supports informed decision-making, including the evaluation of environmental and social impacts. This transparency is expected to increase the visibility and credibility of sustainable products (European Commission, n.d.-a).

The DPP will build on existing databases, creating a decentralized system that ensures data remains at its source, improving efficiency and reducing complexity. In addition to the decentralized DPP data repository, data spaces also play a key role. A data space is a trusted and secure environment where stakeholders and organizations establish common standards and rules for the storage, processing and use of their data. A key feature of data spaces is that they allow participants to retain control over and manage their own data (Data Intelligence Offensive, n.d.).

1.2 Problem Statement and Problem Definition

The implementation of a DPP faces the challenge that, while initial draft standards are already emerging, there are currently no fully established and widely adopted guidelines or standardized processes for the provision and management of the required data. At present, there is no unified architecture that enables seamless integration and interoperability among stakeholders throughout the value chain. These uncertainties lead to inefficiencies and complicate the practical implementation of the DPP. Nevertheless, several draft standards addressing essential elements of a DPP including data exchange protocols, unique identifiers, data storage and archiving, Application Programming Interfaces (APIs), system interoperability and security measures are already under development at the national DIN level and are expected to provide a clearer and more consistent framework in the near future (Deutsches Institut für Normung, n.d.).

Therefore, the underlying IT architecture of the DPP is crucial for its functionality. As a decentralized system, the DPP ensures that data is not stored centrally, which adds complexity to data access and management. Technologies such as blockchain and data spaces play a key role in enabling secure, distributed data storage and access, further highlighting the need for efficient management and interoperability. Additionally, the DPP must be globally unique for each product, meaning that each product has its own distinct DPP, linked through a Unique Product Identifier (UPI), such as a Uniform Resource Identifier (URI). The CIRRPASS project proposes two main approaches for structuring the DPP system, one using Hypertext Transfer Protocol-Uniform Resource Identifiers (HTTP-URIs) and another utilizing Decentralized Identifiers (DIDs). With HTTP-URIs, a Responsible Economic Operator (REO)-Resolver (a web service) receives product identification requests and forwards them to the appropriate target, where the DPP data is stored. In contrast, DIDs are essentially a form of URI. Product identification using DIDs is based on a DID-Subject, which is linked to a DID-Document. This document contains services that provide the appropriate functionality for the stakeholders (Wenning et al., 2024).

Furthermore, each stakeholder has access to DPP data with different access rights, making efficient access rights management essential. For example, consumers may only have read access, while manufacturers might be authorized to modify data. Proper data classification is therefore essential to ensure that the right information is available to the appropriate stakeholder (Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology, 2023).

One of the key challenges is designing the user interface for the DPP. Currently, there is insufficient focus on developing user-friendly, open-source visualization solutions. The DPP should be universally applicable across different products (horizontal scalability) and adaptable to varying levels of data granularity (vertical scalability). Horizontal scalability means that the DPP should be capable of providing product-related data to stakeholders across a broad spectrum of products, whether, for example, it involves furniture or a laptop. Vertical scalability refers to the DPP's ability to manage product information at varying levels of granularity, ensuring it can handle both simple and highly detailed datasets. Furthermore, the DPP must provide both machine-readable data to enable automated processing and integration with other systems, as well as human-readable data to ensure clarity and accessibility for users. Designing an open-source, user-friendly Graphical User Interface (GUI) that is both intuitive and functional remains a critical technical challenge.

One major concern is the risk of information overload for end users. The DPP should be designed in such a way that, for example, when a consumer scans a product’s data carrier, they are directed to a simple, intuitive interface displaying only the essential product details they need. In contrast, manufacturers may require a more complex interface that enables them to modify and manage data, ensuring the system can accommodate the diverse needs of all stakeholders while displaying the appropriate amount of data.

1.3 Research Questions and Research Objectives

Based on the previously mentioned challenges, the following research questions arise:

1. Which data requirements are imposed on a Digital Product Passport regarding different stakeholders such as end user, repair service, recycler and authorities?
2. What solutions exist for efficient access rights management in data spaces based on different roles in the context of a Digital Product Passport?
3. How can a graphical user interface for a (furniture) Digital Product Passport be designed to accommodate the different needs of various stakeholders?

The aim of this thesis is to design and develop a GUI and data interface for application oriented cross company data exchange in the context of a DPP. The resulting prototype will serve as a foundation for future developments and extensions. This prototype will demonstrate the core functionalities of the DPP and provide a model for further refinement. In addition, the thesis seeks to reduce the effort involved in creating multiple DPP demos by addressing various technical challenges. These include optimizing data structure for efficient information storage and retrieval, ensuring interoperability between different systems and platforms and designing a robust system architecture that can support scalability. By focusing on these technical aspects, the goal is to streamline the development process and create a flexible framework that can easily adapt to future requirements and technological advancements.

1.4 Methodological Approach

The foundation of this thesis is based on a literature review, focusing on key aspects of the DPP system. The review will examine topics such as the concept and architecture of the DPP, identification systems and data carriers used for the DPP and the principles of data management and exchange. The purpose of the literature review is to gather, evaluate and analyse relevant contributions on these topics, enabling the researcher to determine the current state of research and identify key findings. This process not only serves to establish a solid theoretical framework for the thesis but also helps to identify gaps in the existing body of knowledge that this research aims to address.

A selected database was explored to find relevant articles, books, conference papers and other scholarly sources. In addition, ongoing projects and grey literature, such as reports from CIRPASS, were considered to ensure a comprehensive overview of the topic. The review was structured to prioritize high-quality, peer-reviewed documents, ensuring that only reliable sources were included. In line with the research questions and the objectives of the thesis, a targeted search strategy was applied, defining key search terms and specifying inclusion and exclusion criteria. The goal was to gather academic contributions and relevant project insights that align with the thesis objectives and provide a comprehensive understanding of the topic.

Based on the insights gained from the literature review, the next step in this research involves programming a GUI. The GUI is developed to manage and display data for three distinct products and is structured around three key building blocks:

- product data, which includes detailed information about each product
- circularity key performance indicators (KPIs), which allow the assessment of the products sustainability
- and material tracking, enabling the monitoring of materials used throughout the lifecycle of the products

The development of the GUI is accompanied by comprehensive documentation, outlining both the technical details of the system and its intended use cases, to ensure clarity and ease of use for stakeholders. To evaluate the effectiveness and usability of the GUI, several assessment methods are employed. Standardized questionnaires such as the System Usability Scale (SUS) and the User Experience Questionnaire (UEQ) are used to quantitatively evaluate usability and user experience.

The GUI is assessed based on key criteria such as usability, adaptability, understandability, user satisfaction, efficiency and data visualization. These criteria ensure that the interface meets the practical needs of the wood industry while providing a clear and efficient user experience.

1.5 Structure of the Thesis

The structure of this master's thesis is organized as follows:

- **Theoretical Principles**
  Presentation of standards and regulations, fundamentals of the DPP, technical foundations and core technologies of the DPP
- **State of the Art**
  Overview of current architecture and data exchange principles for DPP systems, including existing solutions and initiatives related to DPP GUIs, identification systems and data carriers, a as well as concepts for data management in DPPs
- **Development and Implementation of the GUI**
  Requirements of the DPP GUI, development of a universally applicable modular DPP GUI (software architecture documentation based on the C4 model, code implementation – DPP backend, code implementation – DPP frontend), evaluation of the DPP GUI
- **Discussion and Outlook**
  Discussion of results and outlook

2 Theoretical Principles

In light of the growing global threat posed by climate change, decisive action at the international level is essential to secure a sustainable future. Clear, coordinated political measures are needed to limit global warming and reduce CO_2 emissions. A significant step in this direction was the adoption of the Paris Agreement in 2015, which secured a historic consensus among nations to combat climate change and set binding climate targets. At the European level, groundbreaking initiatives have also been launched to accelerate the transition to a climate-neutral economy. In this context, the European Green Deal plays a central role, as it commits European Union (EU) member states to concrete actions aimed at achieving climate neutrality by 2050 and promoting a sustainable, resource-efficient economy (European Commission, n.d.-b). The following sections will examine key international and European climate policy frameworks and regulatory initiatives, including the Kyoto Protocol, the Paris Agreement, the Katowice Outcome, the European Green Deal and the Ecodesign for Sustainable Products Regulation (ESPR). These will be followed by an introduction to the fundamentals of the DPP and its technical foundations and core technologies.

2.1 Standards and Regulations

2.1.1 European Green Deal

The European Green Deal builds upon the foundations laid by international climate frameworks such as the Kyoto Protocol, the Paris Agreement and the Katowice outcome. While the Kyoto Protocol adopted in 1997 established binding emission reduction targets for industrialized countries and emerging economies (United Nations Climate Change, n.d.-a), the Paris Agreement adopted in 2015 expanded the approach to a global scale, urging all countries to limit global warming to well below 2°C and pursue efforts to limit it to 1.5°C (United Nations Climate Change, n.d.-b). The Katowice outcome reached in 2018 further strengthened the Paris Agreement by defining technical guidelines for monitoring, reporting and transparency of climate commitments (United Nations Climate Change, n.d.-c).

Building on these international frameworks, the European Green Deal represents the European Union’s comprehensive strategy to achieve climate neutrality by 2050 while simultaneously transforming its economy toward sustainability (Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology, n.d.).

The European Green Deal, presented by the European Commission on December 11, 2019, pursues the ambitious goal of making Europe the first climate-neutral continent by 2050. As part of this deal, the “Fit for 55" package was introduced, which commits the EU to reducing its GHGs by at least 55% by 2030 compared to 1990 levels. This target, set by the European Council, forms a foundation of European climate policy (European Commission, 2023a).

The eleven elements of the Green Deal include (European Commission, n.d.-c):

- Increasing the EU’s climate ambition for 2030 and 2050
- Supplying clean, affordable and secure energy
- Mobilising industry for a clean and CE
- Building and renovating in an energy and resource efficient way
- Accelerating the shift to sustainable and smart mobility
- From “Farm to Fork”: a fair, healthy and environmentally friendly food system
- Preserving and restoring ecosystems and biodiversity
- A zero pollution ambition for a toxic-free environment
- Mobilising research and fostering innovation
- Financing the transition
- Leave no one behind (Just Transition)

The green transition driven by the Green Deal aims to achieve a range of positive changes. By reducing emissions, it not only combats climate change but also secures a clean and sustainable future. At the same time, this transition will unlock significant economic potential by creating growth and jobs, particularly in innovative and environmentally friendly sectors. Additionally, the reliance on energy imports will be reduced, strengthening energy security and making the EU more independent from external energy sources. In the long run, the Green Deal will also enhance public health and well-being (European Commission, 2021).

A key component of the Green Deal is the EU's Industrial Strategy, which lays the foundation for industrial policy that promotes the transition to a green and digital economy. The aim is to strengthen the competitiveness of European industry in the global market (European Commission, 2023a).

With the Green Deal Industrial Plan, the European Commission aims to foster a more supportive environment for the development and deployment of clean technology manufacturing capacity, which is essential to meet Europe’s ambitious green targets. This plan ensures that Europe will have access to the technologies, products and solutions necessary to achieve a transition to a net-zero energy system. Not only will this drive climate action, but it will also stimulate economic growth and create high-quality jobs across the continent (European Commission, 2023b).

The Green Deal Industrial Plan is based on four key pillars (European Commission, 2023b):

- A predictable and simplified regulatory environment
- Speeding up access to finance
- Enhancing skills
- Trade and resilient supply chains

A predictable and simplified regulatory environment

As part of the Green Deal Industrial Plan, the European Commission proposes several measures to create a predictable and simplified regulatory environment that supports EU industry in achieving climate goals. These measures include (European Commission, n.d.-d):

- The Net-Zero Industry Act
- Promote Regulatory Sandboxes
- Critical Raw Materials Supply
- Affordable and Sustainable Energy
- Infrastructure Expansion
- Harmonization of Sustainability Requirements in Public Procurement

Speeding up access to finance

One of the key pillars of the Green Deal Industrial Plan is ensuring faster access to funding for green projects and technologies. The European Commission proposes several measures to streamline financial support and help achieve climate goals. These measures include (European Commission, n.d.-d):

- National funding
- EU funding (REPowerEU, InvestEU Programme and Innovation Fund)
- Private funding

Enhancing skills

The third pillar of the Green Deal Industrial Plan centers on enhancing skills. As the green transition accelerates, there will be a growing demand for new qualifications at all levels, requiring extensive upskilling and reskilling of the workforce. The European Pact for Skills fosters partnerships within European industrial ecosystems to equip workers with the necessary competencies for the shift to a carbon-neutral and digital economy. The Clean Energy Industrial Forum is also committed to increasing efforts and investments in skill development (European Commission, n.d.-d).

The Digital Education Action Plan aims to speed up education reforms, ensuring the widespread provision of both basic and advanced digital skills across all sectors of the economy and for all age groups. This will enable individuals and businesses to harness digital skills, driving a more sustainable and positive environmental impact (European Commission, n.d.-d).

Trade and resilient supply chains

The fourth pillar of the Green Deal Industrial Plan focuses on trade and resilient supply chains. Open trade supports the EU's net-zero leadership by enabling exports and securing access to key raw materials and components. Through the European Union-United States Task Force on the Inflation Reduction Act, both partners aim to strengthen transatlantic value chains. Additionally, climate and energy partnerships within the Global Gateway initiative support global raw material supply and cooperation in clean technologies (European Commission, n.d.-d).

In conclusion, the European Green Deal Industrial Plan, with its four key pillars, provides a solid foundation to support Europe's industry in transitioning towards a green, competitive and sustainable future.

2.1.2 Ecodesign for Sustainable Products Regulations (ESPR)

The European Commission has established a clear policy framework, including regulations like the ESPR, to accelerate the industrial shift toward a net-zero balance. The ESPR seek to significantly improve the sustainability of products on the EU market by enhancing their circularity, energy performance, recyclability and durability. By broadening the range of ecodesign requirements, the ESPR introduce new standards focused on product durability, circularity and reducing the overall ecological and climate impact. This approach not only helps protect the planet but also encourages the adoption of more sustainable business models, thereby boosting the competitiveness and resilience of the EU economy (European Commission, n.d.-e).

The ecodesign requirements established by the ESPR are applicable to nearly all categories of products and include (European Commission, n.d.-e):

- Enhancing product durability, reusability, upgradability and reparability
- Facilitating easier maintenance and refurbishment of products
- Increasing energy and resource efficiency
- Addressing substances that hinder circularity
- Boosting the use of recycled materials
- Simplifying the remanufacturing and recycling processes
- Defining standards for carbon and environmental footprints
- Minimizing waste generation
- Improving access to information on product sustainability

Furthermore, the ESPR introduces a range of additional measures, including (European Commission, n.d.-e):

- Green Public Procurement: The ESPR establishes mandatory rules for environmentally responsible public procurement of products. According to these regulations, authorities purchasing the relevant products must ensure that they meet the highest standards of sustainability and CE.
- Rules to address destruction of unsold consumer products: The ESPR implements measures to ban the destruction of unsold textiles and footwear, while also obligating companies in all product sectors to annually publish details on their websites about unsold consumer goods they dispose of, along with the reasons for their disposal.
- Digital Product Passport: The ESPR will introduce a DPP, a digital Identifier (ID) for products, components and materials, that provides essential information to support sustainability, promote circularity and ensure legal compliance (European Commission, n.d.-e). The DPP increases transparency for both supply chain companies and the public, enhancing the efficiency of information exchange. It facilitates the exchange of product-related information between supply chain businesses, authorities and consumers (CIRPASS, 2022).

Through these measures, the regulations promote a shift towards a CE, ensuring that products are designed with their entire lifecycle in mind, from production to disposal.

The following provides an overview of the fundamentals of the DPP, along with its technical foundations and core technologies.

2.2 Fundamentals of the DPP

The foundation of the DPP is based on a key focus of the European Commission, the so called "twin ecological and digital transitions." This focus not only addresses emissions and energy, but also highlights resource use, which plays a crucial role (Iakovidis & Ebert, 2023). As previously emphasized, it is essential to design systems, business models, services and products from the outset with full circularity ("circular by design") in order to reduce resource consumption and achieve long-term environmental benefits through sustainable product design.

To effectively implement the Ecodesign requirements, it is crucial that companies and stakeholders across the entire value chain have access to the necessary data and information. Therefore, comprehensive data and information from the entire value chain are required. This is where the DPP comes into play. The DPP facilitates access to relevant data across the value chain, thereby promoting transparent, sustainable product development that supports the objectives of the Ecodesign Regulation (Halmschlager, 2023).

The DPP is a structured dataset containing product and material related information, serving several key functions. From a technical perspective, the DPP facilitates the digital storage and collection of data throughout a product's entire lifecycle. One of its key advantages is the easy accessibility of this data, for example, via data carriers like Quick Response (QR)-Codes, Barcodes, Radio Frequency Identification (RFID)-Tags or Near Field Communication (NFC)-Tags (Verein Industrie 4.0 Österreich, 2024).
From a regulatory standpoint, the DPP serves as a central instrument within the ESPR. This regulation defines overarching requirements for product performance and information, which are fulfilled through the implementation of the DPP. By providing the essential data needed to assess a product's sustainability, the DPP ensures compliance with these requirements. Its implementation will be gradual, initially starting with batteries (from February 2027) and subsequently focusing on prioritized product groups such as textiles (particularly clothing and footwear), iron and steel, aluminum, furniture, tires, detergents, coatings, lubricants, chemicals, energy-related products, and ICT and electronic devices, with expected entry into force for textiles and iron & steel as early as 2027, while excluding food, feed, medicines, live plants and animals, microorganisms, products of human origin, plant- and animal-derived products and vehicles (Verein Industrie 4.0 Österreich, 2024).

The DPP essentially consists of two key components, the DPP Data and the DPP System. The DPP Data represents the information provided by various stakeholders along the product's lifecycle. This dataset includes product-specific information that will vary by product group and will be defined in the delegated acts (Halmschlager, 2023). The data elements required for each product group will be identified when the relevant secondary legislation is developed (Iakovidis & Ebert, 2023).

The DPP Data may contain potential Track & Trace identifiers, such as (Iakovidis & Ebert, 2023):

- Global Trade Identification Number (GTIN) or its equivalent
- Global Location Number
- Name and registered trade name of the economic operator
- Authorized representative
- Integrated Tariff of the European Communities (TARIC)-Code
- Reference to the backup data repository
- etc.

The DPP System refers to the technical infrastructure designed to facilitate the implementation of the DPP concepts outlined in the Ecodesign Regulation

(Halmschlager, 2023). This system is to be developed prior to the DPP's deployment and encompasses the IT and software framework required for the secure storage, retrieval and exchange of DPP data. The DPP System will (Iakovidis & Ebert, 2023 and Halmschlager, 2023):

- Facilitate seamless interaction among stakeholders across the value chain
- Ensure the unique identification of physical products via a Unique Identifier (UID)
- Comply with relevant standards and protocols for IT architecture including:
  - Standards for data carriers and UIDs
  - Interoperability ensuring compatibility on technical, semantic and organizational levels encompassing data exchange protocols and formats
  - Data processing
  - Data authentication, reliability and integrity
  - Data security and privacy
  - Data storage capabilities
  - Access rights management

Thus, the DPP System provides the necessary infrastructure to manage and handle the product-related data, ensuring that the system is both efficient and secure.

The DPP aims to achieve the following key objectives (Halmschlager, 2023):

- Ensure compliance with legal requirements through effective regulatory oversight
- Support consumers in making informed, sustainable decisions
- Foster new business models for stakeholders by enhancing circular value and improving repair, maintenance, refurbishment and recycling through data access
- Speeding up the transition to a CE by optimizing product design, manufacturing, usage and end-of-life management, while enhancing material and energy efficiency and extending product lifespans
- Facilitate the exchange of critical product related information, which is essential for ensuring the sustainability and circularity of products

The expected benefits of the DPP are a direct result of achieving these objectives, leading to a significant impact on product sustainability and circularity. These benefits include (Iakovidis & Ebert, 2023):

- Provide authorities with the necessary information to carry out market surveillance and customs duties

- Allow manufacturers to link digital twins to their physical products, integrating all necessary information throughout the process
- Monitor raw material extraction and production, thereby aiding in due diligence efforts
- Support the monitoring of a product’s full lifecycle, enabling services such as remanufacturing, repair, reuse, resale, second-life applications, recyclability and the creation of new business models
- Offer reliable data to authorities and policymakers, enabling the alignment of market incentives with sustainability targets
- Provide consumers with easy access to verified and relevant product information

2.3 Technical Foundations and Core Technologies of DPP

The successful implementation of DPP´s depends on leveraging innovative technologies that facilitate secure data exchange and enhance resource optimization. Two key technologies in this regard are (Nowacki et al., 2023 and Sousa et al., 2024):

- Data spaces: Data spaces are crucial for the effective exchange of valuable information, creating trusted environments where data can be exchanged, accessed and used while adhering to privacy, security, ownership and legal requirements. International Data Spaces (IDS) provide a framework that establishes common standards, protocols and regulatory frameworks ensuring interoperability and trust between different data spaces. This framework fosters collaboration among stakeholders while protecting the integrity and confidentiality of sensitive data.
- Blockchain: Blockchain technology is essential for the secure and transparent management of DPP´s. As a decentralized and tamper-proof ledger, it ensures the safe storage and verification of product data. A subcategory of Distributed Ledger Technology (DLT), blockchain guarantees the immutability of data meaning that once information is recorded, it cannot be modified or deleted. This feature allows multiple stakeholders to access and contribute to the DPP´s information without compromising data integrity. Additionally, the combination of Ethereum and smart contracts enables data processing within blockchains without the need for central authority coordination. Ethereum’s decentralized platform facilitates process automation enhancing both efficiency and flexibility in product passport management. Smart contracts on Ethereum allow for the automation of transactions and processes directly within the blockchain. Moreover, IOTA, a distributed ledger technology designed for Internet of Things (IoT) architectures, offers innovative features such as zero-cost and efficient

messaging. These capabilities make IOTA especially suited for DPP systems that require efficient, real-time communication across interconnected devices.

While data spaces and blockchain technology provide the foundational infrastructure for secure data exchange and management in DPP´s, the effective tracking and management of physical objects within these systems is equally important. In this regard, automatic identification systems play a crucial role in enabling accurate, real-time tracking and integration of objects within complex environments.

Automatic identification systems are essential in modern industries for the efficient tracking and management of objects. The process begins with identifying an object, which requires an identifier that assigns a unique identity to each object. Identifiers are represented as character strings and their global uniqueness must be guaranteed to avoid any confusion between objects. In addition, a marker, a physical element attached to the object, must store at least the character string, enabling the object to be recognized and identified within the system (Palm et al., 2024). Common examples of markers include barcodes, RFID-Tags, NFC-Tags and QR-Codes (Gligoric et al., 2019). To ensure the markers can be read automatically, sensor technology is essential making sure the system is compatible with sensors that can easily capture the markers. Finally, a database is crucial for storing the relevant information about the identified objects and their lifecycle. This database facilitates traceability throughout supply chains providing valuable insights into the status and movement of the objects (Palm et al., 2024).

There are several architectures for the digital securing of identifiers, as shown in Figure 2.

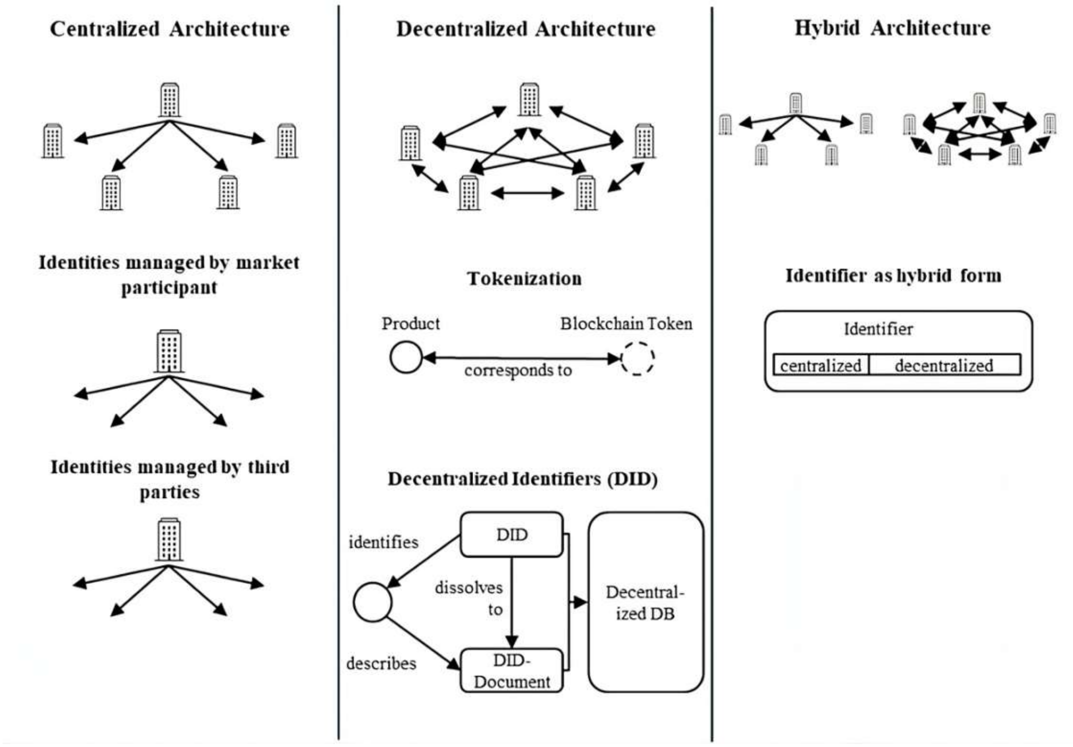


Figure 2: Overview of Identity Management (Palm et al., 2024)

A central architecture relies on a single entity to create and maintain unique identities. In some cases, a market participant acts as the central authority managing the system, known as a centrally managed system by a market participant. Alternatively, an independent third party not directly involved in the market may oversee and manage the identifiers, forming a centrally managed system by a third party (Palm et al., 2024).

In contrast, a decentralized architecture enables identities to be created independently, ensuring their uniqueness through DLT. One approach is decentralization via tokenization, where unique identifiers, like Non-Fungible Tokens (NFTs), are generated on blockchains. These tokens can be linked to real-world products and incorporated into the DPP data model, maintaining decentralized and consistent identities (Palm et al., 2024).

Another approach in decentralized systems is the use of DIDs, defined by the World Wide Web Consortium (W3C). DIDs serve as verifiable, decentralized digital identities described through DID-Documents. These documents provide detailed information about an object's identity and can be retrieved by resolving the DID. Typically stored on a distributed ledger, DIDs and their corresponding documents ensure both integrity and security (Palm et al., 2024).

Lastly, hybrid architectures integrate both centralized and decentralized elements, striking a balance between control and flexibility. This approach makes them well-suited for a variety of identification needs (Palm et al., 2024).

For the DPP, the choice of identity management architecture directly affects scalability, interoperability and trust. While centralized systems may offer simpler governance and faster implementation, decentralized and hybrid approaches better align with the DPP’s objectives of transparency, tamper-resistance and cross-border applicability, making them particularly relevant for ensuring reliable and verifiable product information across the entire value chain.

3 State of the Art

This chapter provides an overview of the current state of the art regarding the DPP, combining insights from scientific literature, grey literature, as well as existing initiatives. The focus is on both backend and frontend.

The backend forms the structural foundation of a DPP system. Key elements include the architecture and data exchange principles of the DPP system, identification systems and data carriers and concepts for data management in DPPs. Building on this backend foundation, the frontend provides accessible and intuitive interfaces for users. Existing solutions and initiatives for DPP GUIs are presented, highlighting how different approaches enable stakeholders to interact with the DPP.

3.1 Backend

3.1.1 Architecture and Data Exchange Principles of the DPP System

This section provides an overview of the fundamental concepts, architectural design and data exchange principles of the DPP system. The following key elements illustrate the structural foundation of a DPP system:

- DPP System architecture using decentralized identifiers and data spaces
- DPP System architecture using asset administration shell and cloud technologies
- DPP System architecture using distributed ledger technology

Some initiatives, such as the CIRPASS project, propose specific implementations and structuring options for the DPP system. In the CIRPASS project, there are fundamentally two different approaches to structuring the DPP system, each with distinct technical foundations and applications. These approaches offer various ways in which the DPP system can be structured and utilized, depending on the specific requirements and planned implementation. According to (Wenning et al., 2024), the two main approaches can be differentiated as follows:

- Structure of the DPP system architecture using HTTP-URIs
- Structure of the DPP system architecture using DIDs

<u>DPP System architecture using decentralized identifiers and data spaces</u>

In the article by (Sousa et al., 2024), a concept for a DPP system is introduced, specifically designed for the footwear industry. Figure 3 presents the proposed concept and architecture of the DPP system. In the proposed concept, a DID-Document is used to access datasets provided via service endpoints.

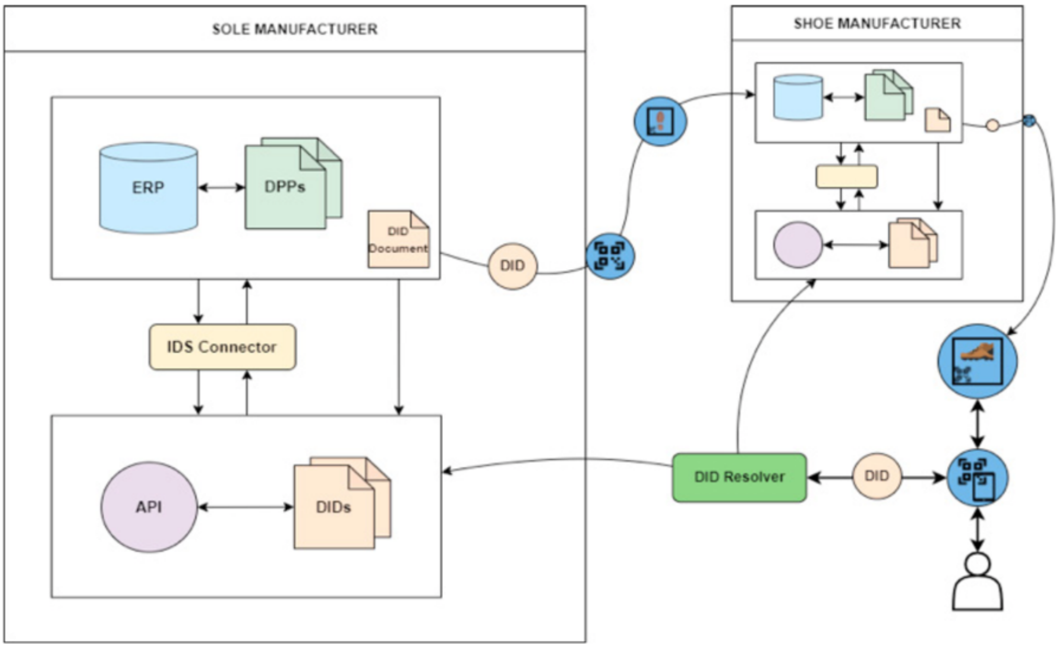


Figure 3: Architecture of the DPP System (Sousa et al., 2024)

The identification of the correct DID-Document for a product is carried out using a unique DID-Code. This code takes the form of a URI and is linked to a data carrier. When a stakeholder scans the data carrier, the corresponding DID-Code can be retrieved, which then allows access to the product's associated DID-Document. Through the service endpoint, the relevant data for the stakeholder is made accessible and displayed (Sousa et al., 2024).

The DID-Resolver plays a crucial role in this process. It is responsible for resolving the DID-Code to the corresponding DID-Document. The DID-Resolver queries the decentralized network or registry where the DID-Document is stored, ensuring that the correct document is retrieved (Sousa et al., 2024).

In the concept proposed by (Sousa et al., 2024), the relevant data for creating the DPP is stored in a dataspace, which serves as a shared data environment or repository. This data space enables secure storage, access and exchange of product-related data among authorized stakeholders. An API provides an endpoint to allow access to the DPP metadata. To access specific private data, the API uses the IDS Connector to query the data space for the required information. If the requested data is available and the consumer has the appropriate access rights, the data is retrieved and made accessible to the stakeholder via the API (Sousa et al., 2024).

DPP System architecture using asset administration shell and cloud technologies

In the article by (Kunz et al., 2024) a concept is introduced for implementing the DPP based on the Asset Administration Shell (AAS), which enhances the DPP with additional functionalities.

The AAS is a standardized digital representation of a physical or virtual asset. It consolidates all relevant information and data about the asset, including technical specifications, condition data and lifecycle information and makes this information accessible in a uniform, machine-readable format (Weber et al., 2024).

During the manufacturing phase, product data is gathered and stored within a product AAS, creating a digital twin that covers the entire product data across the supply chain. Throughout production, various data points, including part-specific details such as weight and geometry, quality parameters and CO_2 footprint are collected within the product AAS, giving each product a UID. These product AAS components are then aggregated into distinct product AASs at different production stages through references. This process creates a continuous information framework that encompasses all product components and processes. Additionally, data from the product's usage and end-of-life phases, as gathered by stakeholders, is also captured and stored. When the product reaches the end of its lifecycle, it can be dismantled and its components may be reprocessed or resold. If a part is reused in a new product, reference to the original product AAS is made again during the assembly process (Kunz et al., 2024).

The proposed concept allows customers to choose between new or reused parts when configuring a product. By employing standardized communication protocols and participant authentication within the data space, this architecture ensures secure, interoperable and data-controlled information exchange maximizing the potential of the DPP throughout the product lifecycle. The used part is placed in reusable parts storage, ready for integration into a newly configured product using reused components. When configuring a product, customers can choose offers from the reuse service provider and opt for used parts instead of newly produced ones, prioritizing the environmental impact of their orders. Figure 4 illustrates the integration of the DPP concept with the data exchanged via the marketplace platform and the information collected throughout the manufacturing process (Kunz et al., 2024).

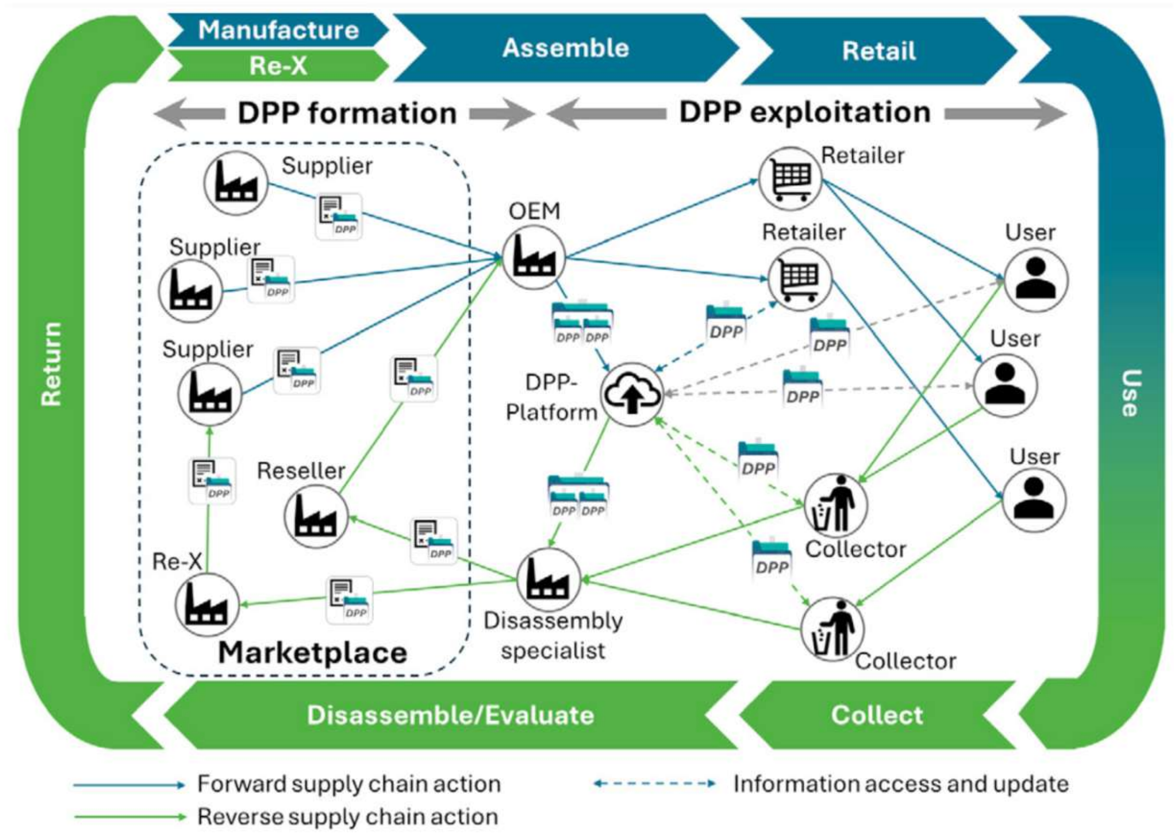


Figure 4: Integration of the DPP Concept with Marketplace Data Exchange and Manufacturing Process Information (Kunz et al., 2024)

Overall, this approach highlights the potential of the DPP system to support CE practices by enabling transparency, traceability, and informed decision-making throughout the product lifecycle, ultimately contributing to more sustainable production and consumption patterns.

In the article by (Bär et al., 2023), a detailed proposal for the implementation of AASs and their use within the supply chain is presented, with a specific focus on their integration into the DPP. The proposal suggests that each asset is associated with one or more AAS which are utilized by the different actors within the supply chain. These AAS exchange data and information using various services and data formats, with options for storage in formats like JavaScript Object Notation (JSON), Extensible Markup Language (XML) or Asset Administration Shell Extended (AASX).
Stakeholders can choose from various methods for storing information, utilizing different protocols. In the case of a cloud solution, a viable option is to create a single AAS that stores all relevant information from the various checkpoints of the DPP. Conversely, with an embedded solution it is more practical to divide the AAS and store only the relevant information (Bär et al., 2023).

To ensure product traceability, all stakeholders must be able to represent, store and interpret both their own data and the data shared by others. Each checkpoint is recorded as a new submodel within the AAS, forming part of the AAS body. Depending on the strategy, these submodels may reside within the same or different AASs of the same asset. As a result, the product's lifecycle can be tracked through the data collected at each checkpoint (Bär et al., 2023).

Two approaches are proposed for storing and accessing data at various DPP checkpoints (Bär et al., 2023):

- Embedded Solution: The product passport is integrated into a device physically attached to the product, storing information on a low-cost storage medium.
- External Cloud Solution: Data is stored in a cloud managed by an external partner. In this case, checkpoints utilize interfaces such as barcodes, QR-Codes or RFID-Tags to facilitate data integration and exchange between AAS instances.

(Bär et al., 2023) propose developing a unified communication platform for interconnected assets, enabling seamless data and information sharing. By consolidating this information within the DPP, a digital thread can be established and made accessible to all stakeholders. This digital thread consists of digitized data linked to various phases of the product lifecycle ensuring comprehensive traceability. Figure 5 illustrates a hybrid structure that integrates both cloud-based and embedded solutions (Bär et al., 2023).

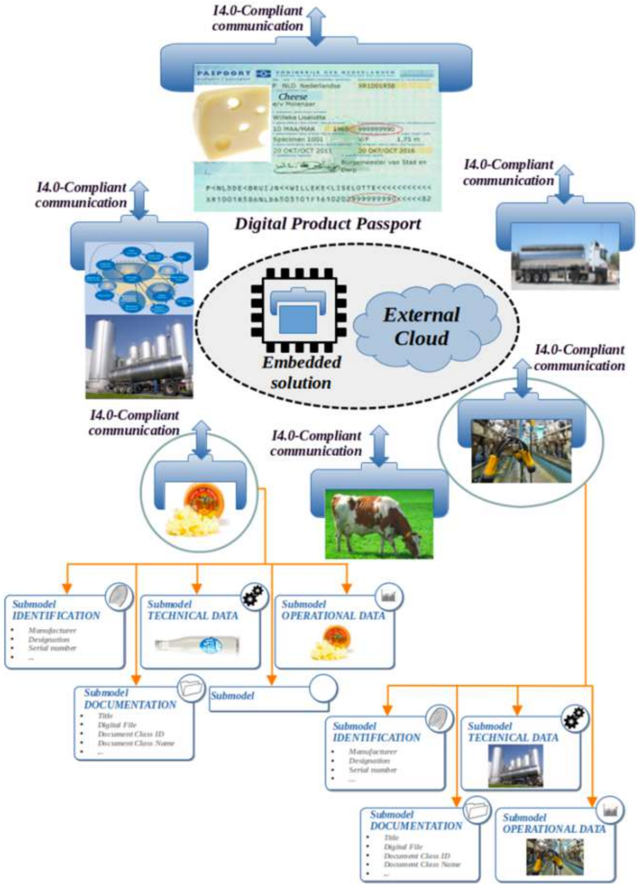


Figure 5: Hybrid Storage Solutions for DPP Information using AAS and its Submodels External Cloud and Embedded Approaches (Bär et al., 2023)

In the article by (Pohlmeyer et al., 2024) the system architecture of a DPP based on AAS is presented, illustrating how participant authentication within a data space enables secure access to specific DPPs, and how DPPs are identified and located using unique asset IDs on the AAS server. Figure 6 depicts this architecture and shows the interactions between the raw material supplier, third-party provider and production system within the value chain.

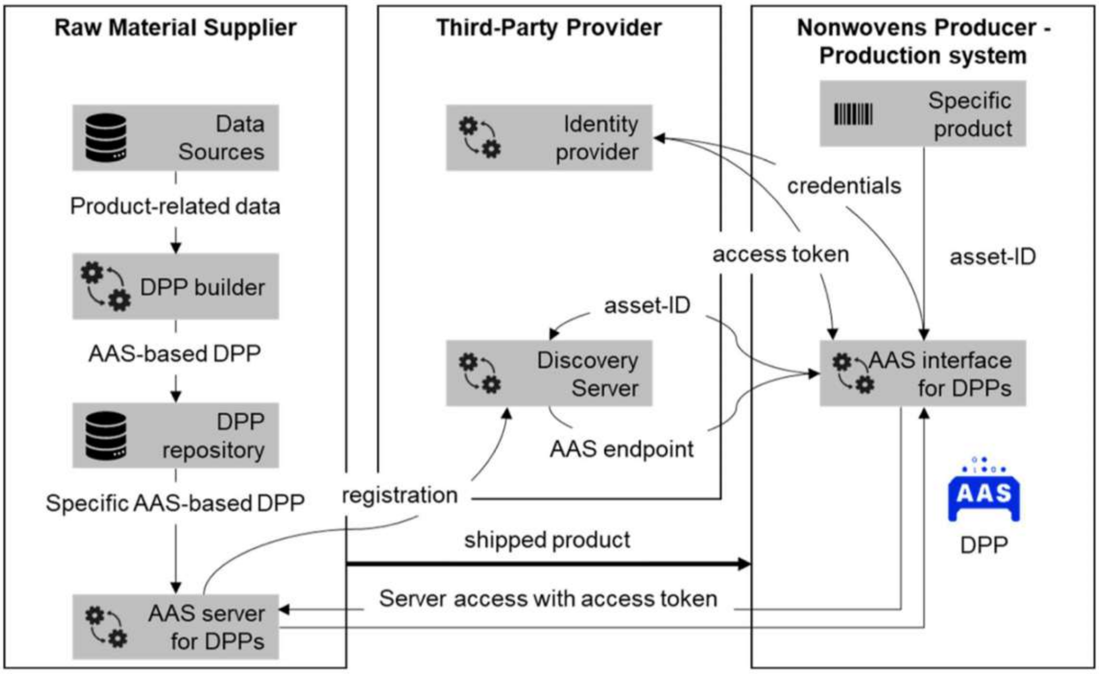


Figure 6: The Architectural Framework of the DPP System (Pohlmeyer et al., 2024)

The process begins with the raw material supplier who manages multiple data sources that include product-related information. This data is handled by a DPP builder which creates an AAS-based DPP. The created DPPs are then stored in a DPP repository functioning as an organized database for product information. To allow standardized and interoperable access to this data, the supplier provides certain AAS-based DPPs through an AAS server (Pohlmeyer et al., 2024).

A key element of the architecture is the third-party provider section, which includes an identity provider and a discovery server. The identity provider handles the authentication of participants within the data space. When a participant from the production system wishes to access a DPP, they must log in through the AAS interface for DPPs using their credentials. Once authenticated, the identity provider generates an access token that grants access to the specific DPP data stored on the supplier's AAS server. The discovery server facilitates the identification and location of DPPs based on a unique asset ID. It does this by accessing a registry where DPP providers store their DPPs. Once the corresponding asset ID is entered or scanned (for example, via a barcode or RFID-Tag), the discovery server retrieves the AAS endpoint address of the requested DPP (Pohlmeyer et al., 2024). This architecture ensures secure and standardized data exchange across the value chain.

The article by (Siska, 2023) describes a system architecture for a DPP based on Eclipse Dataspace Connector (EDC). This architecture is divided into various services and connectors to ensure secure and efficient data management.
The architecture, as shown in Figure 7, consists of several key components including (Siska, 2023):

- A portal enables participants to search, supply and access data while adhering to predefined access and usage policies.
- The semantic hub stores standardized data definitions and makes them available to facilitate automatic data interpretation.
- The registration service handles membership requests and upon approval, issues Verifiable Credentials (VC) to confirm membership.
- Additionally, a catalog improves the system by enabling participants to search for and select datasets. It utilizes a federated architecture, with nodes in each connector and a crawler within the data space. The crawler queries the nodes of each registered participant's connector, which return the current policies and assets. The portal depends on the crawler to enable interaction with the data space content.

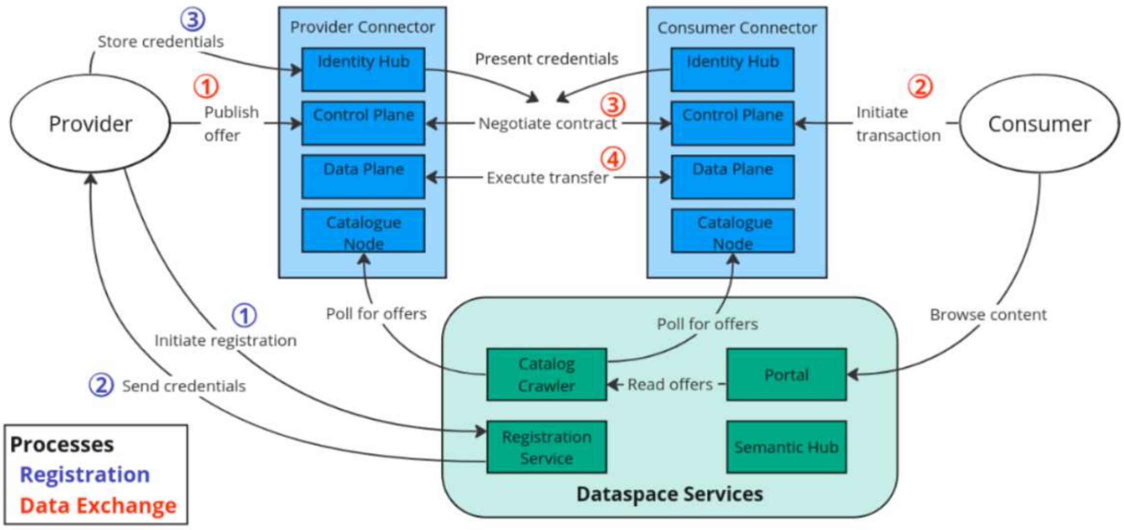


Figure 7: Overview of the Battery Passport Architecture and Process Flow (Siska, 2023)

Each participant is equipped with a connector, a secure gateway for data exchange. In EDC, processes are divided into two planes (Siska, 2023):

- The control plane manages transaction preparation and monitoring, data publication and contract negotiations.
- The data plane handles the actual data transfer, ensuring efficiency and security. Each connector also includes an identity hub that stores and manages VCs for identity verification.

To register, a participant submits a request to the registration service along with the necessary metadata. Upon successful verification, the service issues a signed VC which is then stored in the participant's identity hub. This credential serves as authentication within the data space (Siska, 2023).

The data exchange follows several steps. First, the producer publishes a dataset offer along with associated policies. The consumer initiates contract negotiations through their connector, where both connectors verify each other's VCs, validate policies and ensure compliance with the agreed conditions. Once approved, the data transfer takes place via secure protocols (Siska, 2023).

DPP System architecture using distributed ledger technology

The article by (Nowacki et al., 2023) presents a proposal for the technical architecture of a DPP infrastructure, which leverages IoT sensors, blockchain technology and backend services to ensure secure and efficient data management. As shown in Figure 8, the process begins with data collection from IoT sensors, which monitor and capture essential product-related information. These IoT sensors are strategically deployed throughout the product's lifecycle, from manufacturing and logistics to usage by end consumers. This continuous data collection ensures a comprehensive and transparent digital record of the product.

After the data is gathered, it is sent to a gateway device, which is responsible for generating a digitally signed transaction that encapsulates the processed data and prepares it for secure storage on the IOTA blockchain. The signed transaction is then transmitted to an IOTA node, which is responsible for disseminating it across the distributed ledger. Moreover, smart contracts can be integrated into the system, enabling the automated execution of predefined rules and logic, which further enhances its functionality (Nowacki et al., 2023).

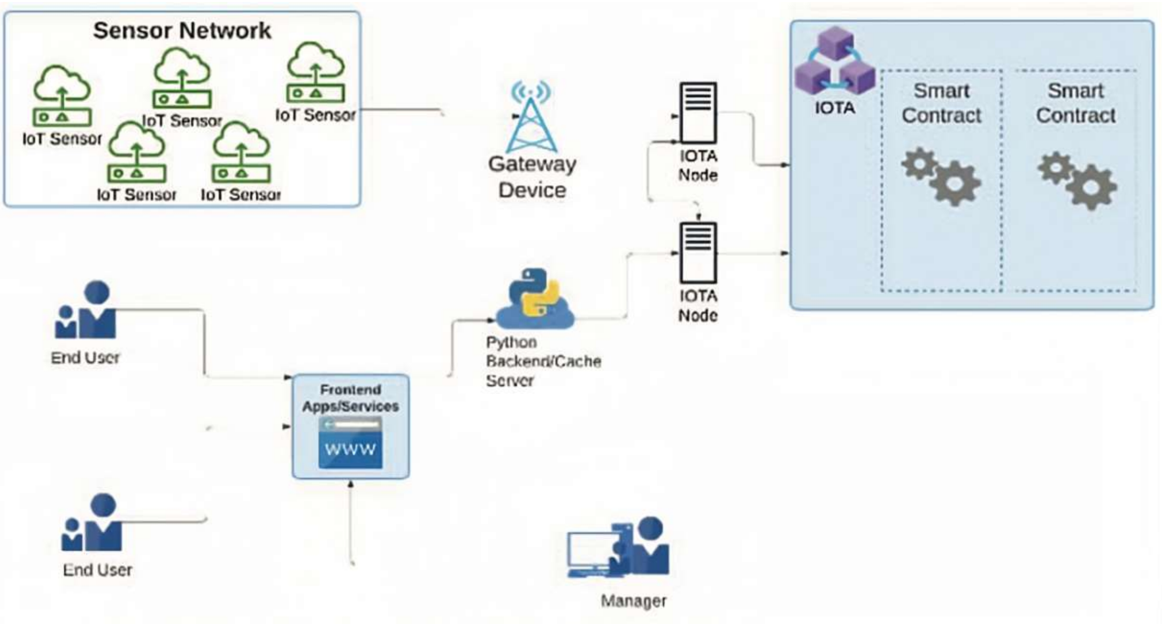


Figure 8: Proposed Architecture of the DPP Infrastructure (Nowacki et al., 2023)

Once the data has been securely stored on the IOTA network, it becomes accessible to users through frontend applications. To facilitate seamless interaction with the stored data, a python-based backend system and cache server are implemented. The backend performs several essential functions, such as authorization management, additional data processing and acting as a central access point for authorized personnel who need to update or expand the functionality of the DPPs (Nowacki et al., 2023).

To optimize performance and ensure fast and efficient data retrieval, cache services are utilized. These services minimize the number of direct requests made to IOTA nodes, thereby reducing system load and improving response times for frontend applications. This results in a more scalable and responsive infrastructure for users accessing DPP information (Nowacki et al., 2023).

Each newly created DPP is assigned a UID, ensuring that all associated data from the IoT sensors is securely linked and consistently stored within the IOTA network (Nowacki et al., 2023). By combining IoT technology, DLT systems and smart contracts, this architecture offers a highly secure, transparent and decentralized approach to DPP.

In the article by (Miron & Hulea, 2024), an innovative approach to the architecture of a DPP system is presented, which combines two separate blockchain technologies to enable effective management and verification of product data.

The proposed approach is based on a multi-blockchain architecture that comprises two main components, the DID-Management Blockchain (DID-MB) and the DPP Data Blockchain (DPP-DB), as illustrated in Figure 9. This architecture uses DIDs to create a unique and verifiable ID for each product. This ID is stored in a distributed and secure data registry that is linked to a DPP (Miron & Hulea, 2024).

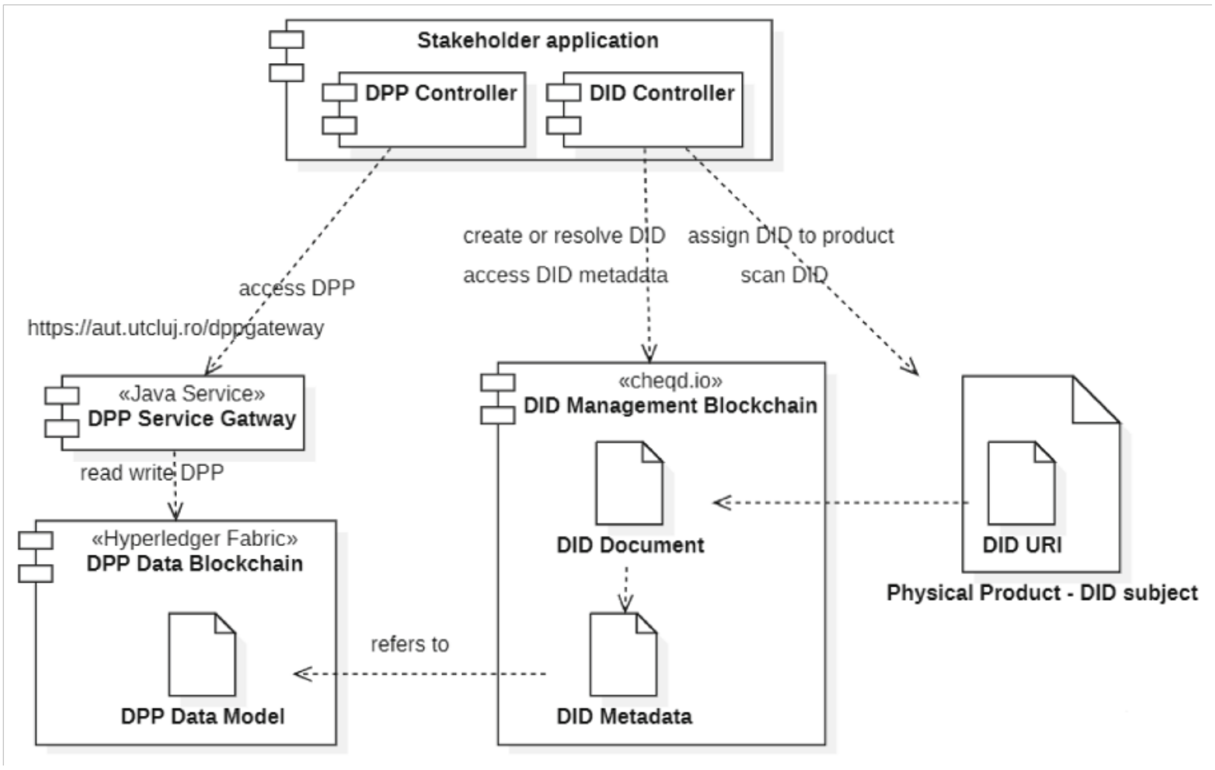


Figure 9: Innovative Architectural Approach of a DPP System (Miron & Hulea, 2024)

The DID-MB is responsible for managing the DIDs and enables secure and decentralized identity data management. On the other hand, the DPP-DB is specifically responsible for storing and managing the product data contained in a DPP. The DPP-DB ensures that all product-related information, such as manufacturing data, specifications and lifecycle details, is stored permanently and immutably. In addition, there are two key components, the DPP-Controller and the DID-Controller. The DPP-Controller is responsible for accessing the DPP and allows both reading and writing of DPP data. The DID-Controller, by contrast, is responsible for creating and resolving DIDs, accessing DID metadata and assigning a DID to a product (Miron & Hulea, 2024).

Structure of the DPP system architecture using HTTP-URIs

Figure 10 shows the structure of the DPP system, with the involved stakeholders and essential components of the system.

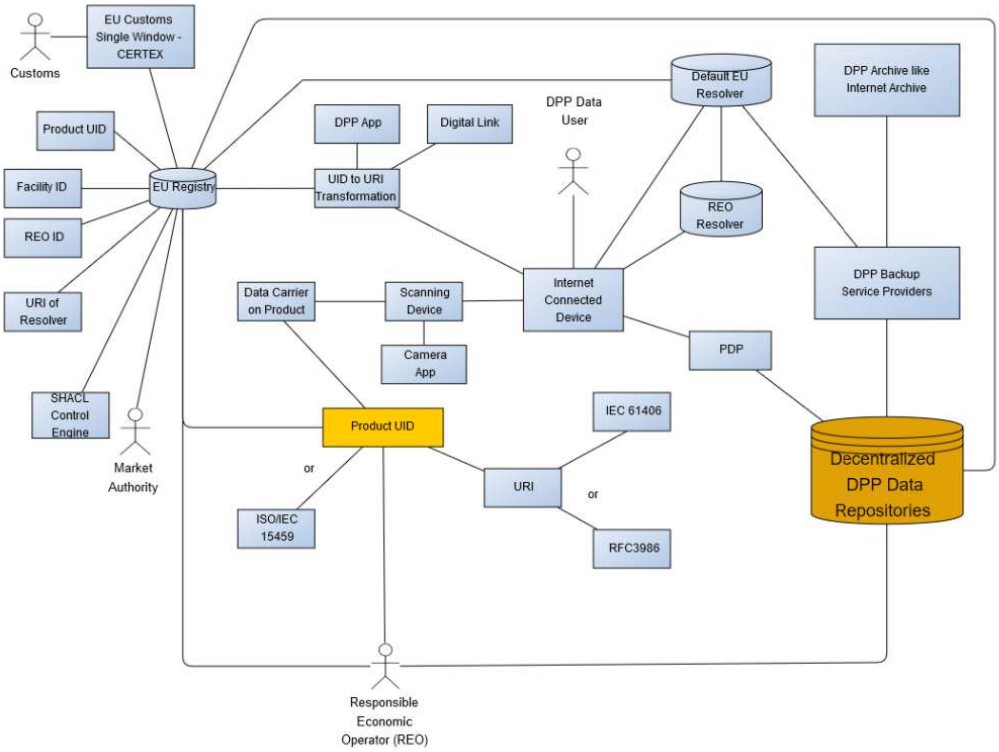


Figure 10: Structural view of the DPP System with HTTP URIs (Wenning et al., 2024)

This figure depicts four stakeholders, namely the DPP data users, the REO, the market authority and customs. The first stakeholder is the REO, who can be a manufacturer, importer or dealer responsible for putting a product on the market or commissioning it. The REO is responsible for creating the product UID, linking it to the product and ensuring that the relevant information is stored in the DPP and made accessible. The product UID and data carrier are immutably attached to the product (Wenning et al., 2024).

The product UID may be in the form of a URI, but it is not necessarily predefined as such. If the product UID is not in URI form, it can be transformed into a URI through a transformation process performed by an Internet Connected Device (ICD) (Wenning et al., 2024).

With a URI, a REO resolver (web service) can receive requests and forward them to the appropriate target. Since the DPP system is decentralized, each REO can operate its own resolver. Product information is stored in the decentralized DPP data repository. If a request fails due to an HTTP 404 error (Page not found), the default EU resolver is activated. This resolver is connected to the EU registry and provides information to the backup resolver to ensure that the DPP information for the product UID is accessible (Wenning et al., 2024).

The EU registry contains key data, such as the unique product identifier (Product UID), unique facility identifier (Facility ID), unique REO identifier (REO ID) and information on public authorities, including regulators, customs and market surveillance authorities. The EU registry also includes a Shapes Constraint Language (SHACL) control engine that validates Resource Description Framework (RDF) graphs against predefined conditions. Customs checks whether products undergoing customs procedures have a valid DPP. This check is performed electronically and automatically through the connection between the EU customs single window certificates exchange and the EU registry. Customs can also use the SHACL validation templates provided by the EU registry to verify the accuracy of the DPP (Wenning et al., 2024).

Another important stakeholder is the market authority, which is responsible for ensuring that the products comply with eco-design regulations (Wenning et al., 2024).

Moreover, the access rights of the stakeholders are managed within the Policy Decision Point (PDP). Not every stakeholder should have access to all data, but only to the information that is relevant and useful to them. The link type allows for the identification of the requestor's role and the provision of the corresponding data. Furthermore, data spaces are in place, designed for secure data exchange. These data spaces must meet specific security, certification and trust requirements. The DPP system can accommodate these requirements, enabling easy integration into a data space. The DPP can be retrieved either from a database or a data space (Wenning et al., 2024).

Structure of the DPP system architecture using DIDs

Figure 11 shows the structure of the DPP system, with the involved stakeholders and essential components of the system.

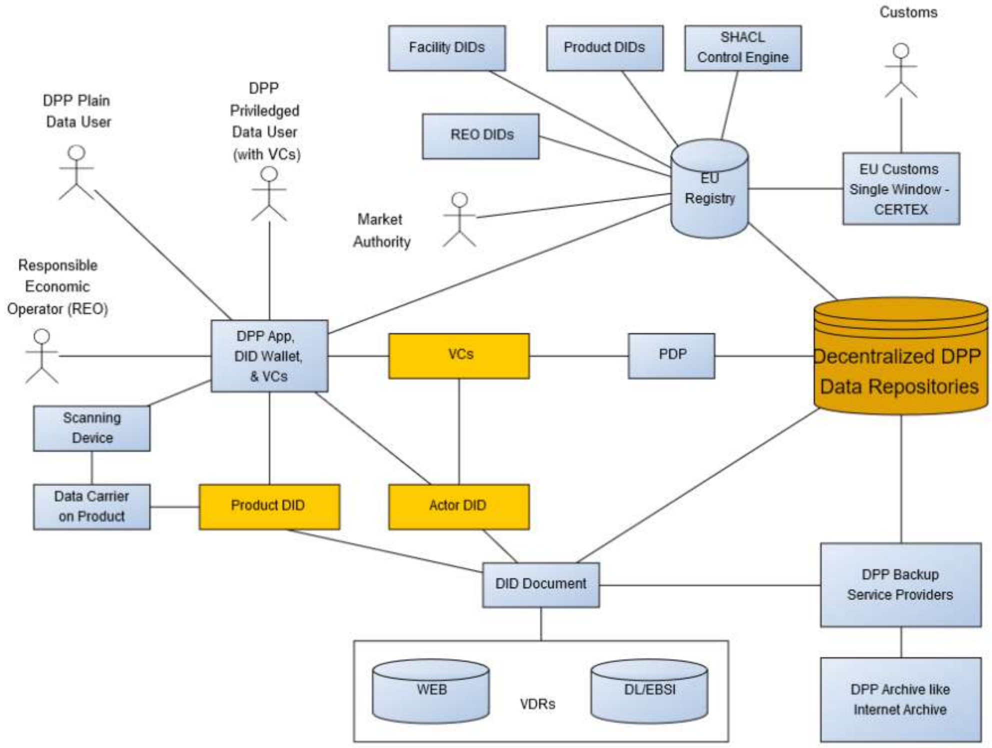


Figure 11: Structural view of the DPP System with DIDs (Wenning et al., 2024)

The structure of the DPP system with DIDs bears similarities to the structure of the DPP system using HTTP-URIs. The following section outlines the key differences between the two systems.

Product identification with DIDs is performed through decentralized identifiers linked to an object. DIDs are URIs that connect a DID-Subject to an associated DID-Document, enabling trusted interactions with this object (CIRPASS, 2024). Along with the use of VCs, DIDs allow for the cryptographic verification of ownership of the identifier and the access control rights for various privileged users. VCs contain information about the identified object, the issuing authority, the type of attestation and the object's claimed properties. Additionally, it documents how the attestation was derived and whether there are restrictions such as an expiration date. Each DID-Document may contain cryptographic material, verification methods or services that allow the DID-Controller to prove control over the DID. These services enable trusted interactions related to the DID-Subject. The Verifiable Data Registry (VDR) is the repository where DID-Documents are stored (Wenning et al., 2024).

A DID-Document, retrieved by looking up a DID on the internet, contains numerous pieces of information. A common application of DID-Documents is to list Uniform Resource Locators (URLs) of services linked to the identified object. Thus, the DID-

Document acts as a resolver, allowing the retrieval of relevant data via the provided links (Wenning et al., 2024).

Various methods exist for generating, storing and resolving DIDs. Some of these methods are based on blockchain technology, while others rely on the direct exchange of cryptographic keys or the control of an internet domain (Wenning et al., 2024).

As in the DPP system with HTTP-URIs, there is an EU registry that contains key data such as the unique product decentralized identifier (Product DID), the unique facility decentralized identifier (Facility DID), the unique REO decentralized identifier (REO DID) and information on public authorities, including regulators, customs and market surveillance authorities. The EU registry also includes a SHACL validation engine (Wenning et al., 2024).

The market authority is also responsible for ensuring that the products comply with eco-design regulations (Wenning et al., 2024).

In the DPP system with DIDs, access rights are also managed within the PDP. By specifying the service parameter in the DID-Identifier (Figure 12), the requestor's role can be identified and relevant data can be provided accordingly. The decentralized DPP data repository also contains product information (Wenning et al., 2024).

Figure 12 shows an example of a DID-URI. A DID-URI starts with the DID-Scheme, followed by the DID-Method. Then comes the DID method-specific ID and finally, a service parameter that selects the appropriate service to retrieve the DPP data (Wenning et al., 2024).

did:example:123456789abcdefghi?service=dpp

Figure 12: Example of a DID-URI (Wenning et al., 2024)

3.1.2 Overview and Comparative Analysis of the Architecture and Data Exchange Principles of the DPP System

The following table provides an overview and comparative analysis of key architectural concepts and data exchange principles within DPP systems.

Table 1: Comparative Overview of DPP System Architectures and Data Exchange Principles

Architecture and Data Exchange Principles	Key Characteristics	Limitations / Challenges
Decentralized identifiers and data spaces	• DID-Code in URI format linked to a QR-Code, enabling stakeholders to retrieve the associated DID-Document upon scanning • DID-Resolver enables access to the correct DID-Document • DID-Document provides a service endpoint for stakeholders • DPP-related data is stored in a secure, shared data space accessible to authorized stakeholders • An API serves as the access point for DPP metadata and connects to the data space via an IDS connector	• Complex implementation • Dependency on decentralized networks • Ensuring security and privacy in data sharing
Asset administration shell and cloud technologies	• Architecture of a DPP built on the standardized AAS framework	• Requires standardized submodels • High implementation effort

	• AAS as a digital twin, storing complete product data
	• Lifecycle tracking via unique IDs and submodels
	• Secure data transfer using EDC, standardized protocols and VC validation
	• Flexible storage: cloud or embedded
	  ○ Embedded solution: data stored directly on the product using local storage
	  ○ External cloud solution: data accessed through data carrier via cloud services
	• Data sharing occurs through contracts, VC validation and secure data transfer protocols
		• Dependence on cloud and network connectivity

Distributed ledger technology	• Unique product identification via DIDs • Use of IoT sensors for lifecycle data • Each DPP assigned a UID to ensure consistent linkage of sensor data • Generation of digitally signed transactions and submission to the IOTA blockchain • Smart contracts enable automated rule and logic enforcement	• High computational and storage requirements • Complexity of blockchain integration • Transaction latency, especially under high simultaneous transaction loads

CIRPASS project: Hypertext Transfer Protocol-Uniform Resource Identifiers	• Product UID may be in URI form or transformed into a URI via ICD • REO creates UID, links it to the product and ensures storage in the DPP • REO resolver forwards requests to appropriate targets • Product information stored in decentralized DPP repository • EU registry contains key identifiers (Product UID, Facility ID, REO ID) and public authority info • PDP manages access rights	• Dependency on centralized EU registry as fallback • Security and privacy must be carefully managed • Potential latency in data retrieval if resolvers fail or network issues occur
CIRPASS project: Decentralized identifiers	• Product identification via DIDs linked to DID-Documents • DID-Documents stored in VDR • VCs enable cryptographic proof of ownership and access rights • EU registry still maintains key decentralized identifiers (Product DID, Facility DID, REO DID) and SHACL validation engine • PDP manages access rights	• Complex implementation due to cryptography and DID resolution • Dependency on decentralized networks and VDR availability • Need for proper governance and secure management of DIDs and VCs to ensure trustworthiness and prevent unauthorized access

3.1.3 Identification Systems and Data Carriers

Identification systems assign a UID to a product, enabling its recognition and tracking across supply chains. In General, UID is assigned by the REO. The UID is permanently and immutably attached to the product, either directly or via a data carrier. It may be expressed in the form of a URI, which enables the UID to be resolved over the internet. If not originally formatted as a URI, the identifier can be transformed by an ICD to enable resource resolution (Wenning et al., 2024).

In the article by (Miron & Hulea, 2024) DIDs are employed to create a unique and verifiable identifier for each product, which is then stored in a VDR and linked to the corresponding DPP. This identifier resides in a distributed and secure data environment, ensuring persistent and tamper-proof association with the product. The creation and assignment of the DID is managed through a DID-Controller. When the DID is resolved via a DID-Resolver, it retrieves the corresponding DID-Document and metadata, which in turn references the DPP.

The use of DIDs offers a significant advancement in the identification process by providing a decentralized, cryptographically secure method of identity verification and data linkage. Unlike conventional identifiers, DIDs are not issued by a central authority, instead any actor within the supply chain can generate and manage their own DIDs autonomously. This promotes self-sovereignty and ensures that control over identity and associated data remains with the rightful owner (Miron & Hulea, 2024).

DIDs are designed to be persistent, globally resolvable and interoperable, making them particularly suitable for tracking products across their lifecycle. They also enable the use of VCs, which support secure and trustworthy communication of information between parties in the value chain. This integration enhances the trustworthiness, consistency and accessibility of product information throughout the product's lifecycle and contributes significantly to the robustness of DPP systems (Wenning et al., 2024).

Furthermore, a DID is a URI-based identifier that is not dependent on a centralized authority. Each DID refer to a corresponding DID-Document. This document provide access to external resources, such as a DPP. DID-Documents can be resolved using DID-Resolvers, enabling secure, verifiable access to product information across distributed systems (CIRPASS, 2024).

The GTIN is a globally standardized identifier used for the unique identification of products. It forms the basis for common barcodes and follows the specifications defined by the ISO/IEC 15459 standard. The international organization GS1 manages this standard and has developed the GS1 Digital Link standard, which enables the transformation of a GTIN into a URI (Wenning et al., 2024).

This transformation allows the GTIN to be used not only for physical product labeling but also for digital linking to a DPP. Through a resolver, the URI can be resolved to automatically access the DPP. Therefore, the GTIN acts as a key identifier that connects physical products to their digital information, which is especially advantageous in the context of DPP (Wenning et al., 2024).

In order to link physical products with their digital identities, physical data carriers are attached directly to the product. In principle, various data carriers are used, such as:

- RFID-Tags
- NFC-Tags
- QR-Codes
- Barcodes
- and other alternative product identifiers

These are employed to uniquely identify products and enable their identification (Gligoric et al., 2019).

In the article by (Yang et al., 2024), RFID-Tags are used as data carriers. A significant advantage of RFID-Tags, highlighted in the article, is that they do not need to be touched directly to be read remotely, enabling efficient data retrieval. They also have greater storage capacity and can hold more information.

Another benefit is that multiple RFID-Tags can be read simultaneously, accelerating large-scale traceability. Compared to barcodes and QR-Codes, RFID-Tags are also more durable and tamper-resistant (Yang et al., 2024).

A disadvantage of RFID-Tags is that their production is significantly more expensive than that of barcodes and QR-Codes, especially in the case of active RFID-Tags. Additionally, reading performance can be affected by materials such as metal or liquids (MECALUX GmbH, n.d.).

In the article by (Narayanan et al., 2024), an innovative approach to product identification is presented, in which an RFID-Tag is used in combination with a NFT linked to the product UID, which is stored on a blockchain.

The NFT, which represents the digital identity of the product, contains all relevant product information. Creating the NFT ensures that the physical product and its digital representation are inseparably linked (Narayanan et al., 2024).

The data stored on the RFID-Tag includes (Narayanan et al., 2024):

- the product UID
- the batch or lot number
- the manufacturing date
- the expiration date for perishable goods or limited-shelf-life products
- the product category or type
- the manufacturer ID
- a URL for access
- the NFT reference ID or a link to the associated NFT

Linking RFID data with NFTs enhances traceability and enables verification of product authenticity. The NFT acts as a certificate of authenticity, ensuring transparency and trust throughout the product lifecycle (Narayanan et al., 2024).

A simple scan of the RFID-Tag allows immediate access to the associated NFT and provides all product history and relevant product data. This makes the process of verifying a product's history and authenticity more transparent for all parties involved (Narayanan et al., 2024).

After examining the use of RFID-Tags as a data carrier for product identification, attention turns to the NFC-Tag.

In the article by (Gegeckienė et al., 2022), the authors discuss various approaches for retrieving data, with NFC being identified as one of the most promising technologies. This technology offers numerous advantages for both businesses and consumers, particularly in the packaging industry. It presents an innovative way for companies to bring their products and brands to life and strengthen customer relationships.

For NFC-Tags to be used effectively in the packaging industry, they must meet certain requirements. These include resistance to cold and heat, as NFC-Tags are used in various environments. Additionally, they must be moisture-resistant to function reliably in wet or humid conditions. Mechanical abrasion resistance is also important, as packaging may be exposed to physical stress during transport or handling. Depending on the NFC-Tag, both the storage capacity and scan range can vary. The scan range of NFC-Tags depends on the surface. On metallic surfaces, the range is typically limited to about 6-20 mm due to interference, while on non-metallic surfaces, ranges of 21–25 mm can be achieved. NFC is specifically designed for short-range communication to ensure security and avoid interference. Compared to other wireless technologies like RFID, which can cover greater distances, NFC remains a technology for close-proximity communication (Gegeckienė et al., 2022).

While NFC allows direct and contactless communication over short distances, QR-Codes represent another common technology for transmitting data but they rely on visual recognition and the scanning of codes.

QR-Codes are a widely used technology applied in various fields for product identification. They can be divided into two main types (QR TIGER LTD, n.d.):

- Static QR-Codes
- Dynamic QR-Codes

Static QR-Codes contain fixed information that is created once and then printed onto the product. These codes are suitable for product details or URLs that do not change. Dynamic QR-Codes, on the other hand, allow the information embedded within them to be updated without replacing the code itself. This makes them especially useful for product tracking, where associated data needs to be regularly updated (QR TIGER LTD, n.d.).

In the article by (Dayma et al., 2023), a system for secure product identification and traceability is introduced, focusing on the use of QR-Codes in combination with blockchain technology. The proposed solution utilizes the Ethereum blockchain to ensure transparency, security and data integrity across the supply chain.

A core element of the system is the generation of QR-Codes linked to product data stored on the blockchain. These QR-Codes allow consumers to verify the authenticity of a product simply by scanning them. The product information is stored and managed via smart contracts, which automate key processes such as product registration, transaction logging, and verification.

Similarly, the article by (Chiacchio et al., 2020) presents an innovative method that also uses QR-Codes and blockchain for product tracking and authentication.

Their approach involves using QR-Codes to track and authenticate products within a blockchain-based system. The data for this application is generated by an automated software routine in the so-called blockchain server. This server interacts with a smart contract to create a digital identity for the product, which is securely stored and encrypted on the blockchain. This digital identity includes all relevant product information such as name, classification, production date, expiration date and a UID. A UID, assigned by the smart contract in the form of a QR-Code, is attached to the product and serves as a key to access all relevant information (Chiacchio et al., 2020).

The system ensures transparency and traceability of products through the combination of blockchain and QR-Codes, while providing a secure and decentralized method for managing and authenticating products throughout the supply chain (Chiacchio et al., 2020).

In the article by (Gligoric et al., 2019), the Smart-Tag technology is introduced. It combines printed sensors and QR-Codes with UIDs and can detect predefined environmental changes.

This technology uses printed sensors made of functional ink, which react to environmental influences such as temperature and brightness. These sensors can measure various physical or chemical parameters and detect changes in the surroundings, offering wide applications in product labeling and tracking (Gligoric et al., 2019).

A Smart-Tag must encode at least two types of information, the identity of the object and the sensor status. The reading device must be able to decode both types of data to fully benefit from the technology (Gligoric et al., 2019).

A specific example from the article involves the use of thermochromic ink (Figure 13) in a Smart-Tag. The ink changes color based on temperature changes and can be reversible or irreversible. Reversible thermochromic ink returns to its original color, while irreversible ink changes color permanently. This feature can be used to monitor temperature ranges during transport or storage to determine if a product exceeded a critical temperature. Smart-Tags can also contain additional data such as URLs or specific product information (Gligoric et al., 2019).

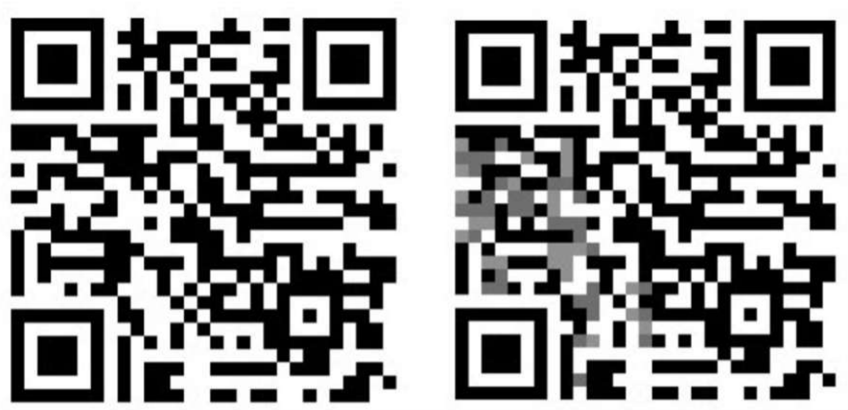


Figure 13: Smart-Tag with Irreversible Thermochromic Sensors; left: QR-Code before the State Change, right: QR-Code after the State Change of the Sensor (Gligoric et al., 2019)

Overall, Smart-Tag technology provides an innovative solution for product verification and monitoring, combining printed sensors with QR-Codes. It enables not only the confirmation of a product’s identity but also the collection of crucial environmental data, which is vital for quality assurance.

While Smart-Tag technology offers a promising approach, the method proposed by (Ishiyama et al., 2016) introduces an innovative alternative for objects where conventional product identifiers are difficult to apply.

The article describes the use of micro ink droplets, called micro Identifier Dot on Things (mIDoT). These droplets act as UIDs that are mixed with standard ink and form random, irregular patterns on surfaces. Since each droplet is unique, every mIDoT can be identified by matching its image characteristics (Ishiyama et al., 2016).

Manufacturers apply mIDoTs to individual parts and store their images in a cloud database, along with serial numbers and product details. A microscope camera connected to a smartphone is used to identify the parts and request product information by communicating with the cloud server (Ishiyama et al., 2016).

The Fingerprint-based Image Recognition (FIBAR) method is used to capture surface irregularities as robust image features. These allow for reliable image matching across large-scale identification tasks. In this method, the flat area of the ink droplet appears dark in images, while all other areas appear bright (Ishiyama et al., 2016).

The matching process occurs in three steps, as shown in Figure 14:

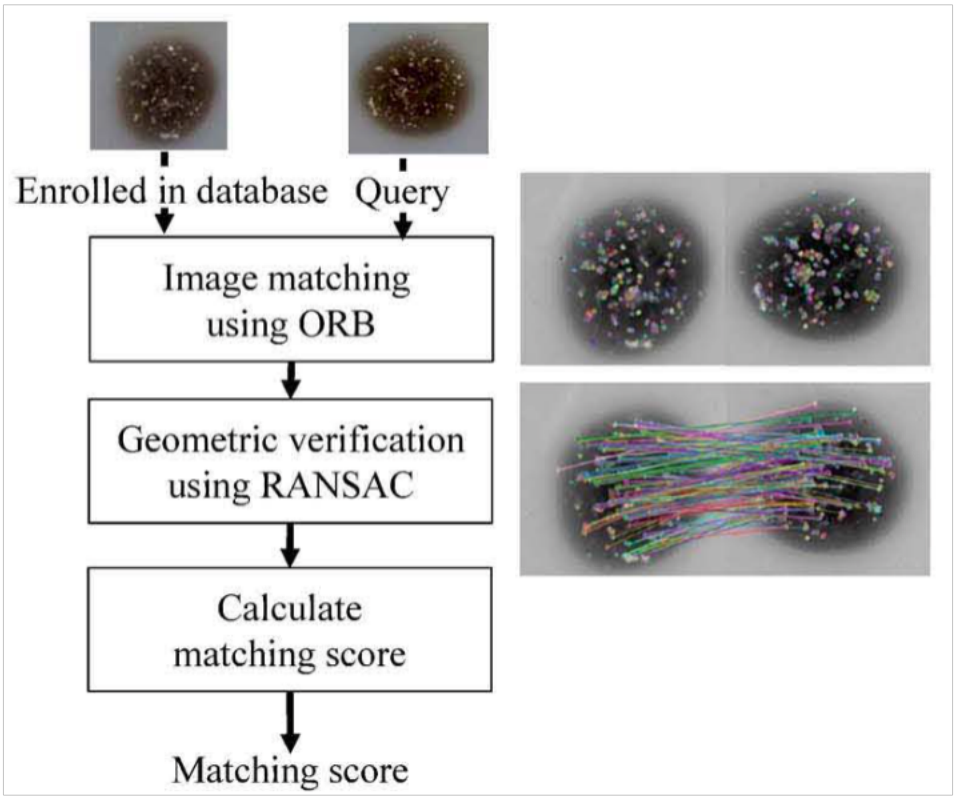


Figure 14: Illustration of the Matching Process for mIDoT Images (Ishiyama et al., 2016)

First, local feature matching is performed, followed by geometric verification and the calculation of a matching score. The Oriented Fast and Rotated Brief (ORB) algorithm is used to identify matching points between images. Random Sample Consensus (RANSAC) is then used for geometric verification and the match score is calculated by dividing the number of correct matches by the total number of point pairs. Once the highest match score is determined, the image can be identified as the same mIDoT (Ishiyama et al., 2016).

In the article by (Sai et al., 2022), identifiers are embedded directly into products using 3D printing technology, as shown in Figure 15. In this approach, a barcode is utilized as the UPI.

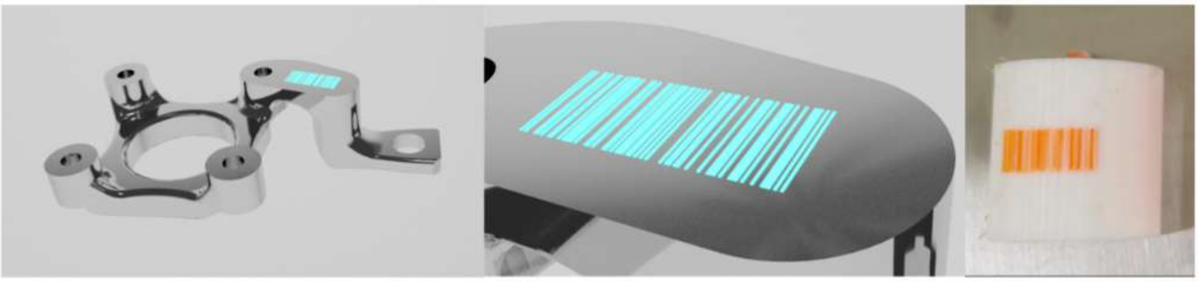


Figure 15: Embedding Identifiers directly into Products using 3D Printing Technology (Sai et al., 2022)

The barcode is inseparably embedded within the product's structure, ensuring that any attempt to remove or alter it would result in visible damage, thereby safeguarding the integrity of the identifier. To verify the identifier, a standard smartphone is used as a reader, equipped with a mobile application that scans the embedded barcode. This application retrieves the corresponding certificate from a blockchain and compares the scanned code with the stored data to confirm product authenticity (Sai et al., 2022).

3.1.4 Summary and Comparative Analysis of Identification Systems and Data Carriers

Table 2 provides a comprehensive overview of various identification systems and data carriers used for DPPs, highlighting their key characteristics and functionalities. It begins with identification systems and then presents a range of data carriers.

Table 2: Overview of Identification Systems and Data Carriers with Advantages and Disadvantages

Identification System / Data Carrier	Advantages	Disadvantages
Decentralized Identifier	• Decentralized and cryptographically secure • Enables self-sovereignty: stakeholders generate and control their own DIDs • Persistent, globally resolvable and interoperable (enables the use of VCs for trusted information exchange) • Each DID resolves to a DID-Document that links to external resources	• Requires infrastructure for DID-Resolvers and VC ecosystem • Challenging integration into existing systems
Global Trade Item Number	• Globally standardized identifier defined by the ISO/IEC 15459 standard • Foundation of barcodes, widely adopted worldwide • Can be converted into machine-readable URIs via GS1 Digital Link	• Centrally managed by GS1 • Static identifier, dynamic data accessible only via external infrastructure (e.g., GS1 Digital Link standard)
RFID-Tags	• Contactless scanning possible over longer distances	• Higher cost compared to barcodes and QR-Codes, especially for active RFID-Tags

	• Higher storage capacity than barcodes and QR-Codes • Multiple tags can be read simultaneously • Resistant to dust, dirt and moisture • Can be linked to NFTs for digital identity	• Signal interference with metals and liquids • Requires specialized readers and infrastructure
NFC-Tags	• Contactless and highly secure • Less prone to interference compared to other wireless technologies • Resistant to environmental factors such as dust, dirt and moisture	• Very short reading range (6–25 mm) • More expensive than QR-Codes or barcodes
QR-Codes	• Easy to generate, scan and use • Can be scanned with smartphones • Enables end-to-end product traceability • Couples QR-Codes with printed sensors (Smart-Tags) that respond to environmental conditions such as temperature and light • Dynamic QR-Codes allow data updates without reprinting	• Limited storage capacity compared to other technologies like NFC or RFID • Vulnerable to physical damage • Static QR-Codes unsuitable for changing information
Barcodes	• Simple, low-cost, widely used • Scannable with smartphones • Immutable and tamper-proof • Can be applied as stickers or directly embedded	• Very limited storage capacity • Susceptible to wear and physical damage

Alternative product identifiers	• Unique and hard to counterfeit • Space-efficient, suitable for small products • Provides precise identification through surface irregularities	• Requires special hardware (microscopic camera) • Dependent on cloud connectivity and image quality • High computational demand for image processing

3.1.5 Concepts for Data Management in DPPs

Effective data management is a key prerequisite for ensuring reliable and consistent handling of information. Especially within the context of the CE, innovative concepts such as DPPs are gaining importance by enabling comprehensive and transparent tracking of product-related information throughout the entire lifecycle.

The article by (Spiß et al., 2024) explores the bidirectional integration of DPPs within Production Planning and Control (PPC) systems. This integration enables real-time transfer of data from DPPs to planning systems, while production-related information such as manufacturing events is simultaneously fed back into the DPPs. This ongoing data exchange establishes a foundation for an integrated, data-driven production planning process that aligns with CE objectives.

To facilitate this interaction, the authors propose an event-driven integration model, as illustrated in Figure 16. Product-relevant data is generated at specific stages throughout the product lifecycle by various stakeholders. These events, which represent changes in the status of objects are stored in an event repository, ensuring consistent data synchronization across different systems. When the DPP is queried, these recorded events are retrieved and reused within PPC systems. The temporal context of each event guarantees traceability and precise assignment throughout the entire lifecycle (Spiß et al., 2024).

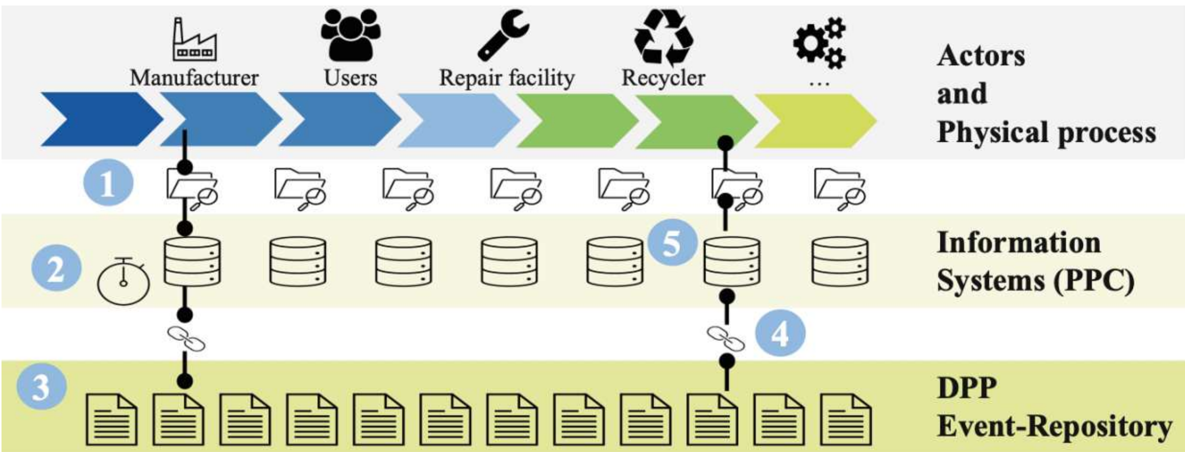


Figure 16: Event-driven Data Lifecycle Management (Spiß et al., 2024)

The integration approach relies on the Electronic Product Code Information Services (EPCIS) standard developed by GS1, which structures events along five key dimensions (Spiß et al., 2024):

- What (object)
- When (time)
- Where (location)
- Why (business context)

- and How (state)

Several standard event types capture various lifecycle activities and status changes (Spiß et al., 2024):

- ObjectEvent records status changes of individual objects, such as shipping, receiving or inspection.
- AggregationEvent documents the physical grouping or separation of objects, for example assembling multiple components into a new unit or disassembling them.
- TransformationEvent describes irreversible transformations where input objects are converted into output objects, such as processing raw materials into finished goods.

These event types support comprehensive documentation and traceability of product lifecycle processes. Additionally, the EPCIS standard includes a Core Business Vocabulary that standardizes common business processes, statuses and documents, thereby ensuring semantic consistency and interoperability (Spiß et al., 2024).

From a data management perspective, this model requires clearly defined rules regarding event logging, data ownership and data processing. Data must be accessible at the appropriate time, in the correct quality and format. A unified data model is critical to satisfy the needs of both DPPs and PPC systems, with an emphasis on semantic interoperability and standardization to facilitate seamless data exchange across organizational boundaries.

(Greiner et al., 2024) highlight that the operation of blockchain consortia for DPPs brings specific challenges and requirements for effective data management. Within these decentralized ecosystems, governing data flows and defining the roles of participating actors are crucial to ensuring reliability, transparency and trust.

In terms of data management, responsibilities are clearly distributed among consortium members, covering technical, organizational, compliance and supply chain aspects. Each participant contributes to maintaining the security and availability of the blockchain infrastructure, guaranteeing that product certifications and sustainability data are tamper-proof and verifiable. Governance frameworks assign roles for overseeing data quality, access control and validation processes, protecting data sovereignty and preventing misuse (Greiner et al., 2024).

Moreover, the consortium must implement monitoring mechanisms and enforce rules with graduated sanctions that address violations proportionally. This regulatory layer is essential for preserving a trusted data environment, especially when dealing with sensitive or immutable blockchain data. Stakeholders must also accept shared data ownership and adapt organizationally and technically to the digital transformation these systems require (Greiner et al., 2024).

In summary, according to (Greiner et al., 2024), effective data management within blockchain-based DPP consortia depend on robust governance structures, reliable technology and shared agreements among participants. Together, these elements enable secure, transparent and efficient handling of product-related data, fostering sustainability and trust across complex value chains.

(Thunyaluck & Valilai, 2024) demonstrate in their article that effective data management requires the structured collection, administration and provision of product-related data throughout the entire product lifecycle. The authors highlight the importance of technical measures such as data validation, quality assurance, version control and audit trails, which are essential for ensuring the integrity and traceability of information. Additionally, they emphasize the necessity of developing standardized data models and semantically harmonized ontologies, as only uniform data structures enable consistent processing and interpretation of information across different systems.

Effective data management also includes the lifecycle management of the data itself. To minimize risks such as data leaks or unauthorized disclosure, binding policies for the storage, archiving and deletion of DPP data are necessary. A well-designed data lifecycle management system thus contributes significantly to ensuring data protection throughout a product's entire usage period and to fulfilling regulatory requirements (Thunyaluck & Valilai, 2024).

Permissioned blockchains, such as those implemented with the Hyperledger Fabric framework, provide a reliable technical foundation for the secure and privacy-compliant exchange of product-related information within known actor networks (Canciani et al., 2024).

(Canciani et al., 2024) demonstrate in their work that Hyperledger Fabric enables the creation of tailored, private blockchain environments featuring defined access controls, private communication channels and support for smart contract development in common programming languages. Due to its permissioned structure, this model is particularly suitable for enterprise networks where data access must be limited to authorized participants. This reduces security requirements compared to public blockchains while allowing for a precise balance between transparency and data privacy. This balance is essential for DPPs, as both internal data integrity within companies and external traceability must be ensured.

(Canciani et al., 2024) propose a hybrid blockchain architecture, as illustrated in Figure 17. This approach combines the advantages of private blockchains such as controlled access and high configurability with the strengths of public blockchains which provide full auditability and immutability. Product-related information is managed in lightweight, portable data structures (ledgers) within the private blockchain. All changes to these ledgers are cryptographically signed and coordinated via a central notary node. The notary consolidates the ledger history and regularly publishes hash-based proofs and integrity verifications on a public blockchain such as Algorand.

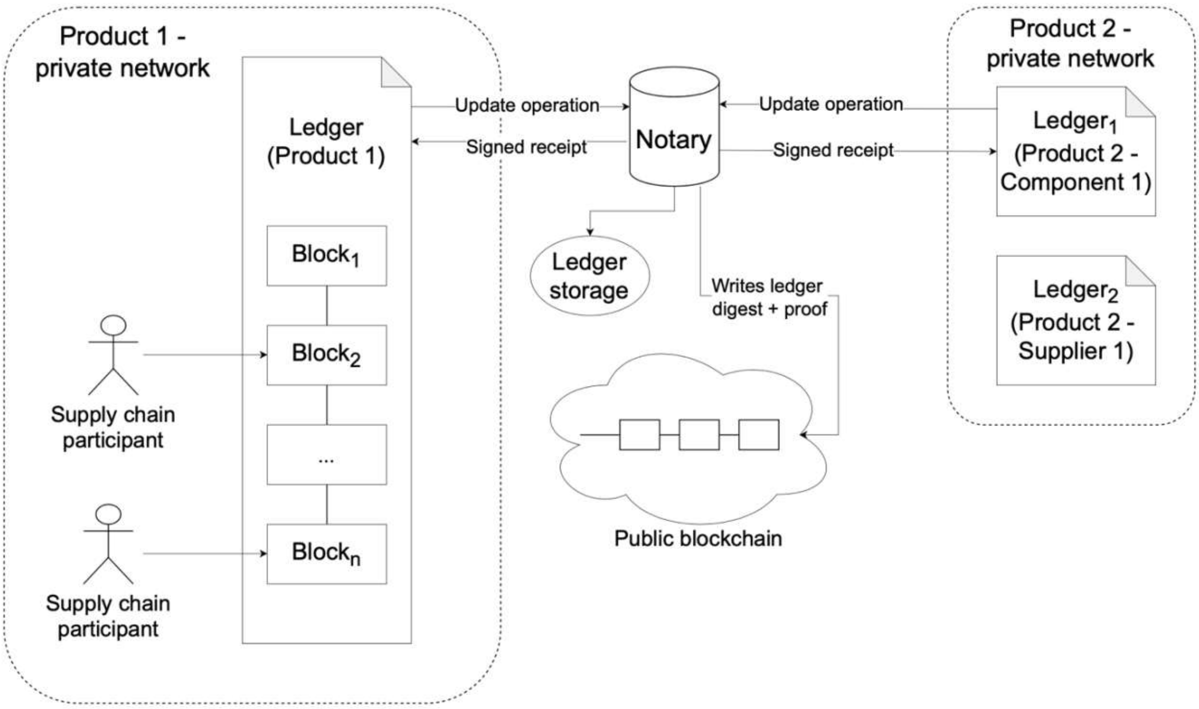


Figure 17: Hybrid Blockchain Architecture for Privacy-Compliant and Verifiable Data Exchange in DPPs (Canciani et al., 2024)

This creates a verifiable external reference structure that allows external auditors to check the immutability and authenticity of a DPP without requiring direct access to the private network. Moreover, this architecture supports granular data sharing tailored to the relevance and role of the data recipient and ensures that every published version of a DPP remains uniquely verifiable and tamper-proof (Canciani et al., 2024).

In conclusion, the hybrid blockchain model presented by (Canciani et al., 2024) offers a forward-looking approach to data management in the context of DPPs. By combining secure data handling within private networks, cryptographic integrity verification and transparent traceability, the model ensures a high level of trust, data protection and interoperability.

3.1.6 Overview and Comparative Analysis of DPP Data Management Concepts

The following Table 3 offers a synthesized comparison of key data management concepts proposed in recent literature for DPPs. Each approach is assessed based on its core features, demonstrating how different architectures fulfill critical data management requirements.

Table 3: Comparison of Data Management Concepts and Key Considerations for DPPs

Concepts	Key Characteristics	Key Considerations
Event-driven integration model	• Clearly defined rules for event logging, data ownership and data processing • Data accessibility at the right time, with correct quality and format • Unified data model • Emphasis on semantic interoperability	• Ensures timely and standardized access to accurate data • Supports consistent communication across systems
Proposal of a blockchain consortia for DPPs	• Clear distribution of responsibilities among consortium members • Governance frameworks for data quality, access control and validation • Secure and verifiable data • Monitoring mechanisms and enforcement of proportional sanctions	• Protects data sovereignty and prevents misuse • Enables trust and accountability among consortium members
Considering lifecycle management of data	• Standardized data models • Policies for storage, archiving and deletion of data • Compliance with regulatory and legal requirements	• Minimizes risks such as data leaks or unauthorized disclosure • Ensures compliance over the full product lifecycle

	• Support for data protection throughout product usage	
Proposal of a hybrid blockchain architecture	• Permissioned access for authorized participants • Cryptographically signed updates coordinated via central notary node • Notary publishes integrity proofs • Granular data sharing based on role and relevance	• Enables external verification without exposing internal data • Ensures authenticity, immutability and controlled data sharing

3.2 Frontend

The development and implementation of DPPs have gained significant traction in recent years, leading to several innovative platforms. Among the most prominent solutions are those from CIRPASS, Circularise and Keepelectronics. Each platform employs distinct technical architectures and technologies tailored to meet specific requirements such as scalability, security, data interoperability and user experience.

3.2.1 CIRPASS

The CIRPASS project has developed a frontend demonstrator to visualize and manage DPP data across various stakeholders in the textiles and electronics value chains. This demonstrator serves as a practical tool to assess data availability, identify information gaps and facilitate role-based access to product information. The development process ensures that the demonstrator aligns with the specific needs and expectations of its intended users. The frontend demonstrator also emphasizes user-friendly interaction and intuitive navigation, allowing stakeholders to efficiently explore and analyze complex product information (Osterwalder, 2024).

A key feature of the demonstrator is its role-based access system, which categorizes product information into various levels of confidentiality and availability, ensuring that users such as product designers, material producers, consumers, recyclers and certifiers can access data relevant to their specific role and information needs across the value chain. For example, upstream stakeholders in the textiles industry may have limited access to commercial information, warranty details and care instructions, while downstream stakeholders, such as recyclers and certifiers, may require detailed information on material composition, hazardous substances and instructions for disposal. This disparity underscores the importance of a flexible and user-centric frontend design that can accommodate diverse informational needs. In addition, the CIRPASS frontend provides interactive visualization features, including tables, hierarchical data structures and filtering options, which allow users to quickly navigate and analyze complex product information. Furthermore, it includes visual indicators for sustainability compliance (Osterwalder, 2024).

In summary, the CIRPASS frontend demonstrator exemplifies a user-centered approach to DPP visualization, emphasizing the importance of role-based access, data transparency and stakeholder-specific information needs. By aligning the frontend design with the varied requirements of its users, the demonstrator contributes to the broader goal of enhancing product transparency and supporting CE initiatives.

3.2.2 Circularise

Circularise offers a user-centric frontend solution for DPPs, enabling stakeholders to intuitively visualize and manage product information throughout the supply chain. The platform is designed to facilitate transparency and sustainability by granting users easy access to product data through a user-friendly interface. This frontend allows stakeholders to explore critical product information efficiently, supporting informed decision-making along the product lifecycle (Circularise, n.d.-a).

The frontend emphasizes user-centered design, ensuring high usability and interactivity. Stakeholders can view product data in structured formats, including material composition, sustainability metrics and compliance information. This approach allows users to make informed choices regarding product design, production, usage and recycling. Access is streamlined through simple mechanisms such as QR-Code scanning or secure login, making the platform accessible even to non-technical users (Circularise, n.d.-b).

A key feature of the Circularise frontend is its secure data-sharing system, which enables selective disclosure of information. Businesses can share only the specific data required for regulatory compliance while keeping confidential business information protected. This ensures both adherence to privacy and security standards and the safeguarding of commercial secrets (Circularise, n.d.-b).

In addition, the Circularise frontend supports the principles of the CE by providing full traceability of products. This functionality is particularly relevant for meeting EU regulatory requirements, such as the ESPR, which mandates the implementation of DPPs for certain product categories. By integrating these traceability features directly into the user interface, the platform facilitates compliance while promoting transparency and sustainability throughout the supply chain (Circularise, n.d.-a).

Overall, the Circularise frontend exemplifies a modern, user-focused approach to DPP visualization. It balances accessibility, security and detailed data presentation, allowing stakeholders to interact with digital product information efficiently. By providing interactive data visualization, selective disclosure of information and full traceability of products, the frontend contributes to broader objectives of enhancing product transparency and supporting CE initiatives (Circularise, n.d.-a and Circularise, n.d.-b).

3.2.3 Keepelectronics

Another initiative for implementing a GUI for the DPP is provided by (Keepelectronics, n.d.). Their platform offers a comprehensive and transparent representation of product-related information, exemplified by a laptop. Users are given structured access to various data categories, such as product origin and usage information including "Manufactured," "Purchased" and "Certification." In addition, the interface presents detailed technical specifications about the product.

A central feature of the interface is the so-called Keep Score, as shown on the right-hand side of Figure 18, which aggregates sustainability-related criteria including durability, repairability, energy efficiency, social conditions and recyclability. This score provides consumers with a transparent overview of the ecological and social impact of the product (Keepelectronics, n.d.).

Furthermore, the platform includes information on the digital receipt, warranty conditions and complete product history. This history encompasses stages such as manufacturing, shipping, import, distribution and sale, along with detailed information about the actors involved. The platform also integrates data on social impact, such as worker satisfaction and average working time of assembly workers in manufacturing facilities. Under the section "Environmental Impact" users can find additional environmental data, including carbon footprint, recycled content and energy efficiency (Keepelectronics, n.d.).

Particularly helpful for users is the support section, which not only displays the current device status but also offers repair guides. Moreover, the platform provides information on end-of-life options for the product, such as the nearest collection points for used devices. An overview of relevant certifications further enhances transparency (Keepelectronics, n.d.).

The platform is role-based, as illustrated in the bottom-left corner of the Figure 18, meaning that the displayed information varies depending on the user's role. Supported roles include Producer, Certifier, Consumer, Support, Remanufacturer and Material Recycler. For example, producers are granted access to detailed data on materials and substances used, while recyclers receive relevant information for recovery and reuse processes.

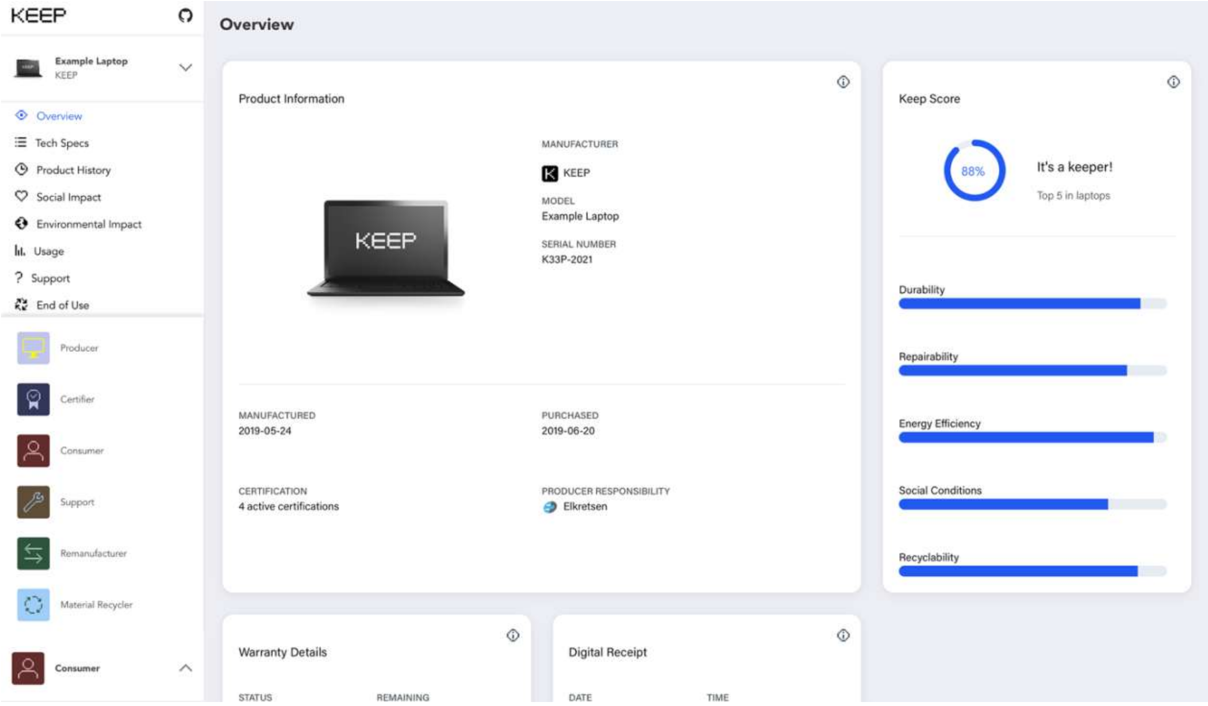


Figure 18: Structure and Layout of the DPP GUI by (Keepelectronics, n.d.)

The initiative by Keepelectronics exemplifies how a DPP can be implemented through a user-friendly and role-specific interface, offering comprehensive data transparency across the entire product lifecycle. This initiative was used as a reference model and served as an orientation for the development of the DPP GUI within the scope of this thesis.

3.3 Consolidated Findings on Backend and Frontend Approaches and Implications for GUI Design

The analysis of the technical implementation of DPPs demonstrates that successful deployment relies on an integrated interplay of backend and frontend. In the backend various approaches exhibit distinct strengths and challenges. DIDs and dataspaces provide high interoperability and security but require stable decentralized networks and involve complex implementation efforts. AAS combined with cloud technologies enable standardized and continuous data frameworks, yet require considerable implementation effort and adherence to strict standards. Distributed Ledger Technologies ensure data transparency, immutability and automation through smart contracts, but encounter limitations in scalability and performance. HTTP-URI-based architectures, as applied in the CIRPASS project, facilitate accessibility through established web mechanisms. DID-based approaches merge the advantages of cryptographically secured decentralized identities with regulatory requirements, although governance and practical implementation remain challenging.

The analysis of product identification systems further emphasizes that no single solution comprehensively fulfills all requirements. Decentralized Identifiers provide security, flexibility and user sovereignty but necessitate infrastructure. Established standards such as the GTIN offer global recognition and standardization yet are comparatively inflexible. RFID-Tags and NFC-Tags enable automation and robustness but incur higher costs. QR-Codes and barcodes are low-cost and widely adopted but limited in storage capacity and security. Emerging approaches, such as Smart-Tags, demonstrate significant potential for innovation, although they are still in early stages of development.

In the frontend, user-centered design proves critical for adoption and effectiveness. Platforms such as CIRPASS, Circularise and Keepelectronics prioritize intuitive usability, interactive visualization and role-based access controls, allowing diverse stakeholders along the value chain to access relevant information. CIRPASS emphasizes flexible, role-based data sharing and interactive visualizations, Circularise provides selective data disclosure, traceability and regulatory compliance, while Keepelectronics provides extensive user support and guidance on end-of-life options, enabling consumers and stakeholders to make informed decisions regarding repair, recycling and product disposal. Collectively, these frontend solutions illustrate that combining accessible visualization, role-specific information control and sustainability measures is essential for the successful deployment of DPPs.

Based on these insights, it becomes evident that the effective implementation of DPPs requires careful alignment of backend and frontend components. The user interface serves as the primary point of interaction, directly influencing usability, transparency and adoption of product-related information. Chapter 4 focuses on the development of the DPP GUI, outlining its design and key implementation considerations. It further illustrates how backend and frontend components work together to deliver an intuitive, secure and interoperable solution, enabling stakeholders to access product-related information throughout the entire product lifecycle.

4 Development and Implementation of the GUI

The development of a user-friendly and functional GUI is a central component in the implementation of the DPP. As the primary interface between users and the underlying product data, the GUI must meet both functional requirements and high usability standards. The goal is to deliver a modular and interoperable solution that enables access to product-specific information throughout the entire product lifecycle.

This chapter provides a comprehensive overview of the development and technical implementation of the DPP GUI. It begins by outlining the core requirements identified during this work, such as the needs of different stakeholders, data accessibility and the integration of external data sources. A detailed description of the system architecture and the software components follows. Special emphasis is placed on the modular system structure, which allows for future scaling and flexible adaptation.

The implementation is divided into two main technical components, the backend and the frontend. The backend handles system logic, data exchange with decentralized sources and the secure and efficient storage of information. The frontend provides an intuitive and responsive interface that visualizes product data for end users and stakeholders. It ensures that all relevant information is clearly presented and easily accessible.

4.1 Requirements of the DPP GUI

The DPP GUI must satisfy a wide range of functional and non-functional requirements to support its intended role across the product life cycle. It should enable intuitive access to product-related data, facilitate seamless integration with upstream data providers, allow for stakeholder interaction and remain adaptable for future extensions. The following section outlines the key requirements identified in the context of this master thesis, based on the ESPR (European Union, n.d.) as well as the defined research questions and research objectives.

<u>User-Friendly and Intuitive Design</u>
One of the primary goals in the development of the GUI is to create an interface that is clear, well-structured and easy to use. The system should be designed so that users do not need technical or programming knowledge to operate it. Every user should be able to understand and use the interface intuitively and without extensive training.

Furthermore, the interface should be responsive and optimized for use on various devices such as desktop computers, tablets or smartphones. This ensures that the DPP can be accessed reliably in different environments and usage contexts.

Simple Access via Data Carrier

To ensure low-barrier access to the DPP, the system supports the use of data carriers, with the QR-Code chosen as the implementation (European Union, n.d.). This QR-Code is physically attached to a product and can be scanned using any standard mobile device.

Upon scanning, the user is automatically redirected to a web page that displays the DPP of the corresponding product. In this master thesis, the QR-Code shown in Figure 19 leads to the webpage dpp.fraunhofer.at/show-data.

Figure 19: QR-Code used as Data Carrier for the DPP

This mechanism ensures that end users can access relevant product data directly without needing to register, log in or install additional software.

Flexible, Modular and Extensible Architecture

The DPP including its GUI and underlying system architecture should be designed to be flexible, modular and easily extensible in order to meet current and future requirements.

This means that the DPP should not be implemented as a rigid system but rather as a dynamic and adaptable solution capable of evolving over time. Key characteristics of this approach include:

- Adding new components or views to the user interface
- Integrating additional data sources
- Supporting new user roles or permissions

This extensibility ensures that the system can evolve over time without requiring a complete redesign.

Data Classification and Access Control

One of the current major challenges in implementing the DPP is ensuring that not all data is visible to all users (European Union, n.d.). Therefore, the system must implement a robust classification and access control mechanism. This includes:

- Public data: Basic product information visible to everyone, such as product details, circularity KPIs and material tracking
- Stakeholder-specific data: Information accessible only to authenticated users, such as additional data or the ability to upload documents (e.g., repair reports)

Integration with Data Spaces

In this master thesis, a data space is used as a decentralized solution that enables various stakeholders along the supply chain to provide product-related data in a standardized and controlled manner. Key requirements include:

- The ability to connect to and retrieve data from data spaces
- Handling data offers provided by stakeholders
- Ensuring that data used in the DPP is up-to-date

This approach supports a data ecosystem where each stakeholder retains control over their data while enabling cross-organizational collaboration.

Automated Data Updates

To ensure the DPP remains accurate and reliable in line with ESPR requirements, the system must support automatic background updates whenever upstream data changes (European Union, n.d.). Once a DPP has been created, any modifications to the product-specific data should:

- Trigger an automatic update of the corresponding DPP
- Be processed without manual intervention from the REO

In this master thesis, the implementation ensures that each time the DPP GUI is accessed, a real-time HTTP POST request is sent to the data space API to retrieve the most up-to-date data available. This guarantees that the information displayed is always current, establishing the data space as the single source of truth.

Authentication and Stakeholder Interaction

In addition to simply viewing product data, the GUI is designed to support authorised stakeholder interaction throughout the entire product lifecycle, as foreseen by the ESPR (European Union, n.d.). After authentication, which in this prototype is implemented through a classic login mechanism using a username and password, stakeholders gain access to enhanced features, such as:

- Viewing extended, role-specific product information

- Uploading new documents

This interactive functionality allows stakeholders to document product-related changes and events over time, directly within the DPP. As a result, all lifecycle events are clearly and transparently visible at a glance.

Maintainability and Governance

In this master thesis, the REO also assumes responsibility for creating, maintaining and extending the DPP. This is feasible for demonstration purposes and small-scale deployment. However, as the system grows in scope and complexity, it becomes necessary to clarify:

- Who oversees long-term system maintenance?
- Who manages user roles and access permissions?
- How are updates, extensions or error handling carried out?

Defining these responsibilities and governance structures is a natural requirement for ensuring that the DPP remains reliable, consistent and accountable over time. Clear role definitions and structured processes support smooth operation, prevent errors and enable efficient scaling of the system.

4.2 Developing a universal applicable modular DPP GUI

The development of a universal applicable modular DPP GUI is crucial for ensuring the flexibility, scalability and interoperability required in modern industrial applications. Such a GUI enables seamless interaction with various data sources, systems and services, all of which are essential for managing product lifecycle information across different sectors and industries. To achieve these goals, several key software tools and technologies are employed in the development process.

An essential part of developing such a GUI is the integration of various software components, each of which serves a specific function within the overall system. The components that are particularly important in this context include:

- Nexyo DataHub for data management and exchange
- Render for web hosting and deployment
- MongoDB for data storage
- Webspace for document storage
- and GitHub for version control

The Nexyo DataHub enables secure and decentralized data management and exchange. By utilizing the EDC framework, it ensures sovereign data management and sharing between different stakeholders while maintaining control over their own datasets. Nexyo allows the GUI to interact with external systems and integrate data securely, ensuring that product-related information can be shared across various partners while maintaining strict data sovereignty principles (Nexyo, n.d.).

Render is a cloud-based platform that simplifies the deployment and scaling of web applications, APIs and databases. It offers automated deployment, seamless database integration and scalability features, ensuring efficient hosting and data management for Nexyo's data transfer (Render, n.d.). In this context, Render acts as the destination endpoint for securely transferring data. Once data is handled by Nexyo DataHub, it is transferred to Render, where further processing and storage occur.

MongoDB, as a Not only Structured Query Language (NoSQL) database, serves as the primary storage solution for product data within the DPP system. MongoDB offers a document-based storage format, which is particularly useful for handling unstructured or semi-structured product data. This architecture ensures that the DPP GUI can manage and retrieve large volumes of product data efficiently, which is especially important for real-time processing of product information throughout the entire product lifecycle (MongoDB, Inc, n.d.).

Webspace refers to online storage that allows users to store and access data, such as documents, images or other files, via the internet. It is commonly used for hosting websites and file sharing. Webspace is typically provided by hosting services, enabling file access through File Transfer Protocol (FTP) or Hypertext Transfer Protocol (HTTP) (Hosttech, n.d.) and (Kinsta, n.d.). In the context of DPP development, the webspace serves as a central repository for important product-related documents. These include repair reports, maintenance instructions, assembly and disassembly guides, warranty and return policies, as well as environmental guidelines. Such documents are essential for managing product information throughout its entire lifecycle. Users can retrieve and upload documents e.g. via FTP, ensuring continuous updates and management of the product’s documentation. Webspace thus plays a critical role in ensuring the efficient storage, retrieval and updating of product-related documents.

For version control, GitHub is employed. GitHub provides a robust platform for managing source code, tracking changes and supporting parallel development. It also offers features such as issues, pull requests and continuous integration, which are essential for maintaining code quality and ensuring smooth collaboration (Github, Inc, n.d.). In the GitHub repository, all code files are stored and managed.

The integration of these components enables the creation of a powerful and scalable GUI, capable of interacting with various external systems, while also offering the flexibility to adapt to the ever-evolving demands of the industry. To effectively document the architecture of such software solutions, the C4 model offers an ideal framework.

4.2.1 DPP Software Architecture documentation based on the C4 Model

The C4 model is a framework aimed at assisting software development teams in defining and communicating the architecture of a system. It is useful during both the initial design process and when documenting an already existing codebase. This model allows teams to build "maps" of the system's architecture at different levels of abstraction, providing a way to explore the components of the system in increasing detail as you zoom in (C4 Model, n.d.-a).

Starting with the System Context Diagram, this diagram offers a high-level view of the system by placing it at the center and surrounding it with its key users and interacting systems. It provides a broad understanding of the system's environment, focusing on the main external entities and their interactions with the system. The diagram intentionally omits low-level technical details such as protocols, emphasizing the relationships between the system and its surroundings. Its goal is to give both technical and non-technical stakeholders a clear understanding of the system's role and connections in its broader ecosystem (C4 Model, n.d.-b).

After gaining an understanding of the system's context, the next step is to explore its internal structure through the Container Diagram. This diagram zooms in on the system's architecture, highlighting the main building blocks, referred to as containers, that form the system. Containers can include elements like web applications, mobile apps, databases or any self-contained, deployable unit. The Container Diagram provides an overview of how the system is organized, showcasing the distribution of tasks across containers and illustrating how they interact (C4 Model, n.d.-c).

The last stage in this process involves examining each container in more detail through the Component Diagram. This diagram dissects each container into its fundamental components, illustrating their roles and how they interact. It offers an in-depth look at the system's architecture, shedding light on the technologies and design decisions behind each component. This diagram is particularly useful for development teams, as it enables them to gain a clearer understanding of how each part contributes to the overall functionality of the container and how they work together seamlessly (C4 Model, n.d.-d).

Starting with the System Context Diagram, shown in Figure 20, the DPP-Software System is positioned in the centre. Surrounding entities include the REO, the End-User, upstream stakeholders and other stakeholders in the Product Lifecycle (PLC). These entities interact with the DPP-Software System in various ways. For example, the REO is responsible for placing the product on the market, creating the DPP and furnishing the product with a data carrier such as a QR-Code. In this context, the REO plays a key role in establishing the DPP-Software System.

Simultaneously, upstream stakeholders are crucial in this process, as they provide raw materials, components and essential data for product manufacturing and compliance. By supplying product-related data, they enable the REO to access this information (via a data space through the Nexyo DataHub) when creating the DPP.

Next, the end user, the product owner with access to the DPP, scans the QR-Code to be directed to a webpage that grants access to product-related data. Here, they can view detailed information such as technical specifications, dimensions, order details (e.g., purchase and delivery dates), material data (e.g., the source of the raw material), circularity KPIs (e.g., circularity scorecard) as well as documents and repair reports.

Furthermore, additional stakeholders in the PLC, such as recyclers, authorities and repair services, have extended access to the DPP. After authenticating themselves, they can interact more extensively with the system, retrieve additional product-related data and generate reports within the DPP-Software System.

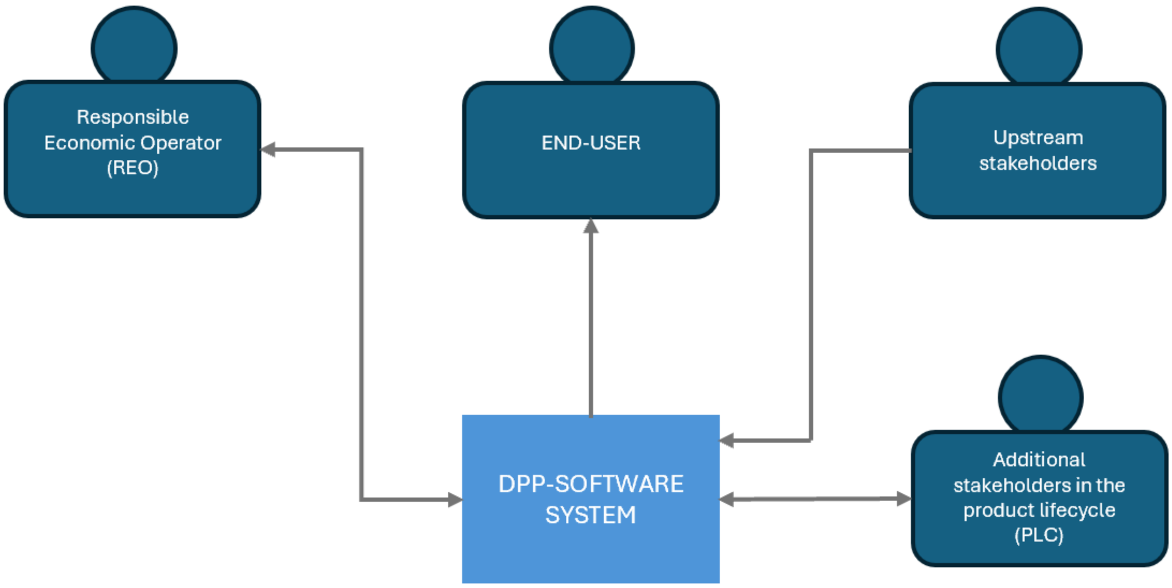


Figure 20: System Context Diagram of the DPP

After understanding the system's context, the next step is to explore its internal structure using the Container Diagram, which is shown in Figure 21.

The DPP-Software System consists of a GUI that provides web access to the DPP. The system also includes an API application, which implements HTTP methods to handle web requests, perform database operations for data management and execute FTP commands for file transfers. To retrieve the necessary data for the GUI, the API application reads from and writes to both a NoSQL database via an API or library and a data storage system via FTP. The database (MongoDB) stores product-related data, while instructions and reports are stored in the data storage (Webspace).

The roles of the various stakeholders are crucial to the functionality of the system. For instance, the end user scans the QR-Code, is redirected to the GUI and accesses product-related data using the interface. Similarly, the upstream stakeholders interact with the system in the same way, accessing data without making any changes.

The arrows in the diagram indicate the flow of data. The arrow pointing outward from the GUI to the end user reflects that they consume the information without modifying it. In contrast, the arrow pointing toward the GUI from upstream stakeholders indicates that they provide the necessary data to the interface. In contrast, the arrows for the REO and additional stakeholders in the PLC point both inwards and outwards, indicating that these users have the ability to interact with the system in a more dynamic way, including making changes to the data.

For example, the REO, who is responsible for creating the DPP, can add new modules or make other modifications to the GUI. On the other hand, additional stakeholders in the PLC, such as repair services and recyclers, gain extended access after authentication. These stakeholders can view additional product-related information and document actions, such as repairs in a detailed report, which can then be updated through the GUI.

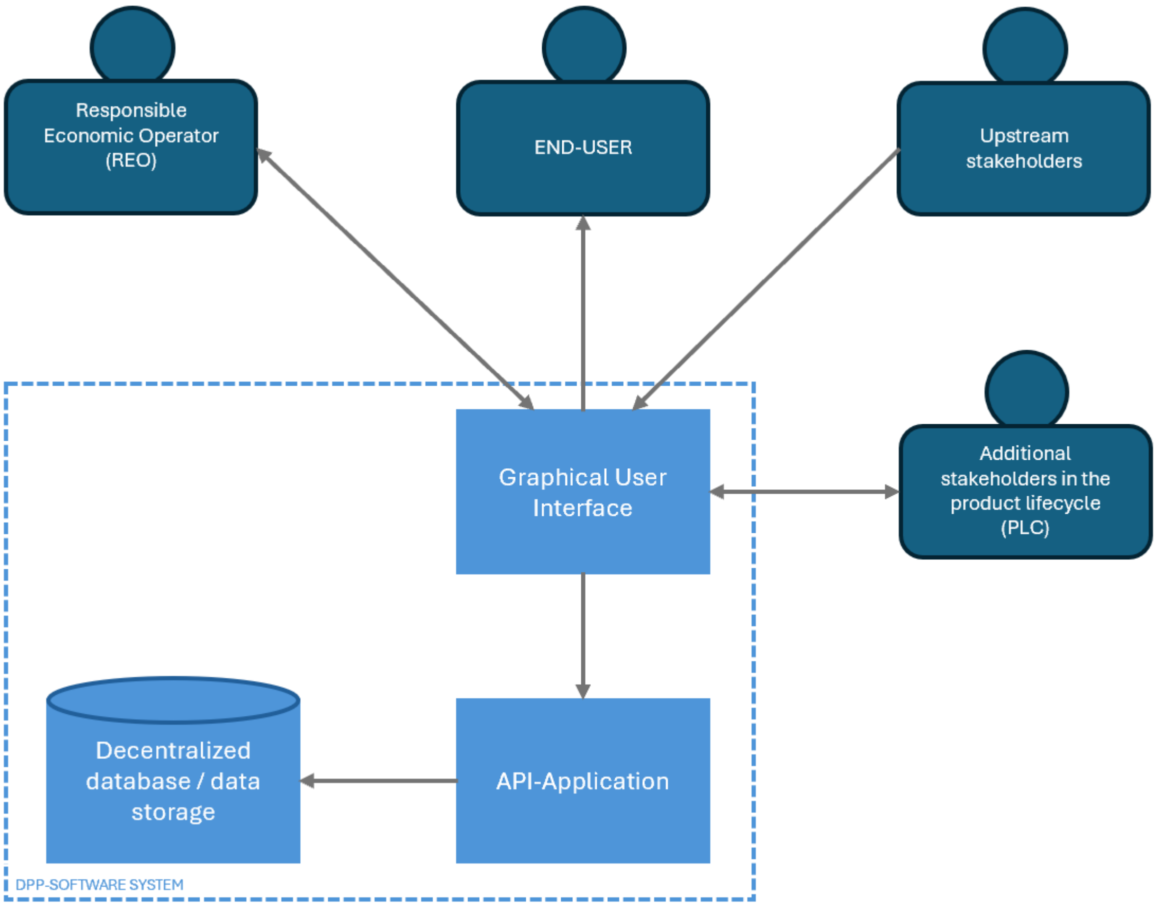


Figure 21: Container Diagram of the DPP

The Component Diagram provides a more detailed view of each container by breaking it down into its fundamental components. Figure 22 illustrates two main controllers the Sign-In Controller and the Data Display Controller, each composed of specific components that define their respective functionalities.

The Sign-In Controller is responsible for authenticating users and enabling access to extended information and document management features. This functionality is implemented through the Dynamic Login Form Component which handles the submission of form data, validates user inputs and displays feedback messages based on the authentication results. This component interacts with a NoSQL database through an API or library, both reading and writing data. It reads data to display additional product-related information to stakeholders after a successful login and writes data back to the database each time a login occurs, as reflected by updated timestamps.

The Data Display Controller consolidates and presents data and documents from various sources to the user. It includes three integrated components that ensure smooth data flow and user interaction. The Data Integration Service Component manages bidirectional data exchange by sending and receiving POST requests to and from external systems. It processes incoming data and stores it in the NoSQL database, specifically MongoDB. It reads product-related data for display and updates the database with only the latest information via POST requests each time the webpage is accessed, ensuring that users always see the most up-to-date content.

The Report Generation Component is responsible for creating and uploading Portable Document Format (PDF) based repair reports that may include tasks, comments and images. These reports are transferred via FTP and stored in a designated webspace. In contrast, the Document Retrieval Component ensures secure access to documents such as instructions and reports. It retrieves this content via FTP from the same webspace allowing users to view essential information.

All components interact either with the database or with the data storage system working together to ensure a seamless, up-to-date and user-friendly experience throughout the application.

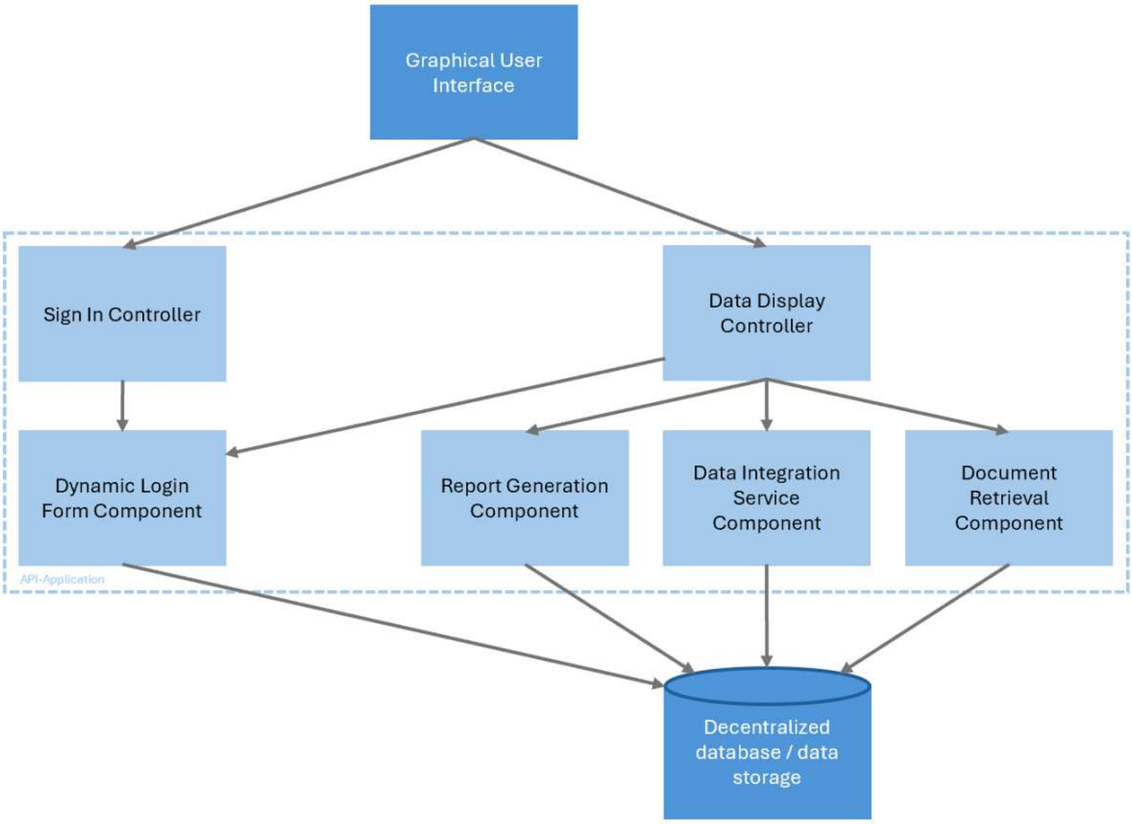


Figure 22: Component Diagram of the DPP

4.2.2 Code Implementation – DPP Backend

To illustrate the implementation of each component, this section first focuses on the backend of the DPP software system, before proceeding to the frontend GUI. Particular emphasis is placed on the backend, as previously outlined, since the application relies on several key software tools (see Section 4.2 for an overview).

The backend is developed using Python, specifically as a Flask project. Flask is a lightweight Web Server Gateway Interface (WSGI) web application framework (Pallets Projects, n.d.-a). It is particularly well-suited for creating Representational State Transfer (RESTful) APIs and backend services, offering clear structure and modularity. RESTful APIs are web interfaces based on an architectural style for distributed systems. They enable data exchange between a client (e.g., the frontend) and a server (e.g., a Flask backend) via the HTTP protocol using defined HTTP methods such as (Mindtwo GmbH, n.d.):

- GET: Requests data from a server
- POST: Sends new data to the server to create a resource
- PUT: Fully updates an existing resource on the server
- DELETE: Removes an existing resource from the server

The development is carried out using Visual Studio Code (VS Code) as the integrated development environment (IDE), which provides extensive support for Python and web technologies. VS Code offers features such as code completion, syntax highlighting, Git integration and debugging tools, making it suitable for both backend and frontend development (Microsoft, n.d.). Python is used as the main programming language due to its strong ecosystem, readability and simplicity (Python Software Foundation, n.d.). In particular, the Flask framework is used to build the backend, enabling the creation of RESTful APIs and dynamic content delivery. Together, VS Code and Python offer a flexible and efficient environment for building modern web applications.

Required Libraries for Backend Implementation

Before describing the individual components of the DPP-Software System, it is important to first introduce the libraries and packages required for the backend code implementation. These libraries provide the necessary functionality to build a secure, modular and efficient web application. The libraries used in the backend are:

- Flask: enables routing, handling HTTP requests (request), rendering Hypertext Markup Language (HTML) templates (render_template), sending JSON responses (jsonify), managing redirects (redirect, url_for) and displaying messages to users (flash)
- Flask-Login: handles user authentication and session management, including login, logout and accessing the current user (LoginManager, UserMixin, login_user, login_required, logout_user, current_user)
- FPDF: allows dynamic generation of PDF documents, such as repair reports
- Werkzeug: provides secure utilities for file handling (secure_filename) and integrates with Flask’s WSGI architecture
- Requests: supports HTTP communication with external systems
- Python-dotenv: loads environment variables from a .env file for secure configuration
- PyMongo (MongoClient, ServerApi, ObjectId): Provides the necessary tools to interact with MongoDB for storing, querying and managing data efficiently. MongoClient establishes the connection, ServerApi ensures compatibility with the server version and ObjectId is used to uniquely identify documents
- Standard Python modules:
    - os: file and directory operations
    - uuid: generation of unique identifiers
    - base64: encoding and decoding of data
    - datetime: timestamping as well as date and time operations
    - json: JSON serialization and parsing
    - re: pattern matching
- Flask utilities: send_file allows sending files (e.g., PDFs) to clients, while abort handles HTTP errors

- io.BytesIO: in-memory streams for creating and transmitting files without writing them to disk
- ftplib.FTP_TLS: secure file transfers between the application and external storage via FTP over TLS

<u>Custom python classes and their role in the system architecture</u>

In the implementation of the backend application, several custom python classes are introduced to encapsulate specific responsibilities and to promote modular, reusable code. These classes are essential for managing file transfers, user sessions and database serialization. Figure 23 illustrates the corresponding UML diagram outlining the structure and interaction of these components.

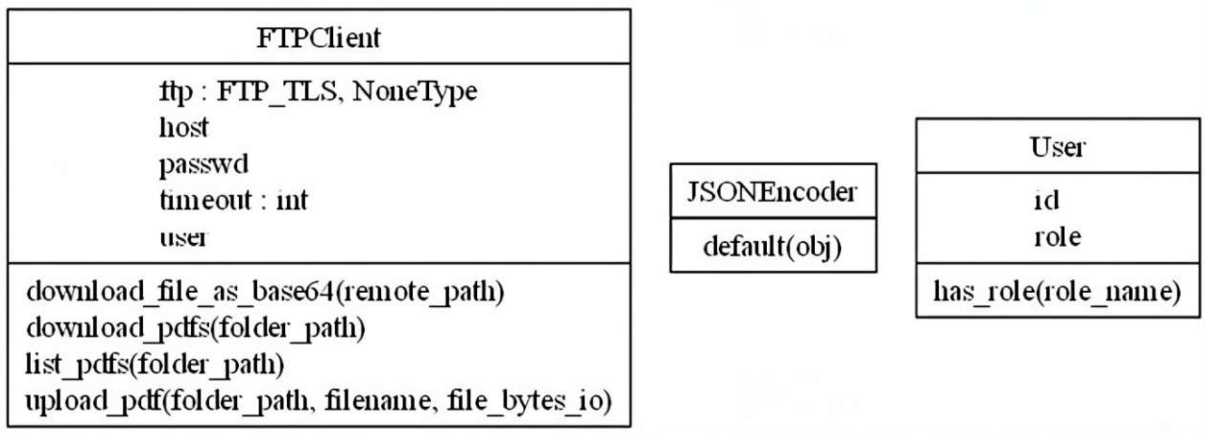


Figure 23: UML Class Diagram illustrating FTP Handling, User Management and JSON Serialization

The FTPClient class is designed to handle secure communication with a remote FTP server using the FTP over Transport Layer Security (TLS) (FTPS) protocol. It is implemented as a context manager to ensure that connections are properly opened and closed, even in the event of exceptions. The primary responsibilities of this class include establishing a secure connection to the FTP server using the FTP_TLS protocol, downloading individual files as base64-encoded strings (which is particularly useful for embedding images directly into HTML), listing all available PDF documents in a given server directory, downloading all PDFs from a folder and storing them as binary data in memory and uploading new PDF files from the application to the server. These functionalities are illustrated in appendix 6.1, specifically in the code sections from lines 39 to 87. By abstracting all FTP-related logic into a single class, the application benefits from a clean separation of concerns and simplifies error handling and debugging related to file transfers.

MongoDB uses special data types, such as ObjectId, which are not natively supported by Python's standard JSON encoder. To address this, the application defines a custom encoder class named JSONEncoder that extends the standard json.JSONEncoder. This class overrides the default method to detect ObjectId instances and convert them into string representations. As a result, complex objects retrieved from MongoDB can be safely serialized and returned as JSON responses via Flask. This is particularly important for enabling interoperability with client-side applications and API consumers that expect valid JSON format. This functionality is demonstrated in appendix 6.1, in the code section from lines 96 to 100.

The User class extends the UserMixin provided by Flask-Login which supplies default methods for session management such as authentication state and user identification. In addition to inheriting this functionality, the User class introduces a custom role management mechanism. Each user is associated with a specific role, stored in the role attribute. The has_role(role_name) method provides a simple way to verify access permissions throughout the application. This enables the implementation of Role-Based Access Control (RBAC), where certain features (e.g., additional information access or creation of repair reports) are restricted to users with specific roles. This approach offers both flexibility and security as it ensures that access to sensitive operations is granted only to authorized users. These functionalities are illustrated in appendix 6.1, in the code section from lines 117 to 123.

The following sections demonstrate how the described components are implemented within this python-based backend system.

The login process begins with the Dynamic Login Form Component, which is activated when the user selects the “Log-in” option from the GUI, as shown in Figure 24. This triggers a GET request to the /login route. The server responds by calling and rendering the login.html template (see appendix 6.3), which displays the login page where the user can enter their username and password. This functionality is demonstrated in appendix 6.1, in the code section from lines 144 to 158.

Once the user submits their credentials, a POST request is sent to the server. The system then attempts to retrieve the submitted username from a predefined users dictionary. If the username exists and the corresponding password matches the stored value, the user is authenticated using Flask-Login’s login_user() method. A confirmation message “Successfully logged in!” is shown and the user is redirected to a new interface view with extended functionality. This functionality is demonstrated in appendix 6.1, in the code section from lines 151 to 154.

If the login credentials are invalid, the login process fails. In this case, an error message “Login failed. Please check your credentials.” is displayed and the login page is reloaded to allow another attempt. This functionality is demonstrated in appendix 6.1, in the code section from lines 155 to 156.

The load_user() function plays a key role in session management. It is used to reload the user object based on the user ID stored in the session, thereby maintaining the user's login state across requests. This functionality is demonstrated in appendix 6.1, in the code section from lines 137 to 142.

When the user chooses to log out, a request is sent to the logout route, which is protected to ensure that only authenticated users can access it. Upon logging out, the user is securely signed out using Flask-Login's logout_user() method, a success message is displayed and the user is redirected back to the main interface. This functionality is demonstrated in appendix 6.1, in the code section from lines 160 to 165.

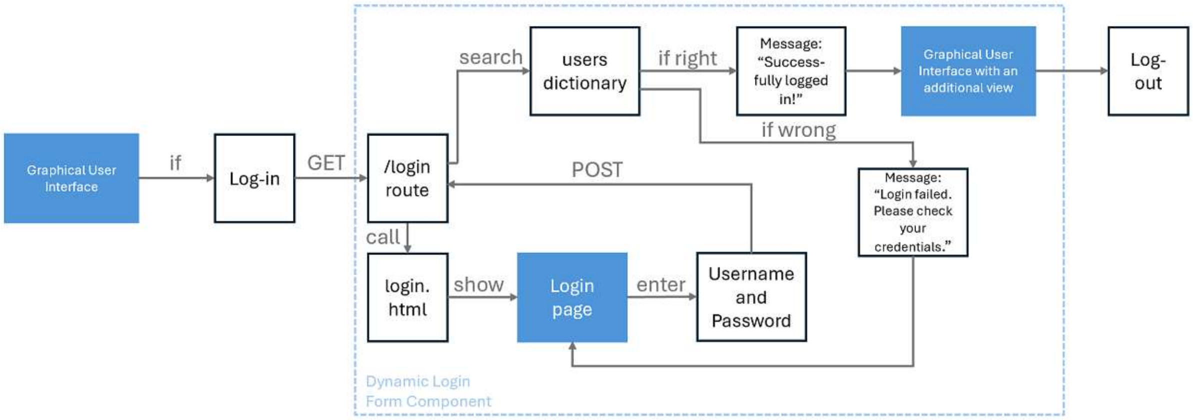


Figure 24: Dynamic Login Form Component

Next, the Report Generation Component can be seen in Figure 25. When a user logs in as a repair service via the GUI and is successfully authenticated, the interface expands to include an additional view. This new view provides access to a form for filling out a repair report. Once the form is completed and submitted, a POST request is sent to the /repair_report route. This functionality is demonstrated in appendix 6.1, in the code section from lines 457 to 459.

In the route handler, the submitted tasks, any additional notes, comments and optional image uploads are collected from the form data. A PDF report is then generated using the FPDF library. The report includes the timestamp, a report Universally Unique Identifier (UUID), the list of selected tasks, a description of the repair and any uploaded photos. This functionality is demonstrated in appendix 6.1, in the code section from lines 460 to 520.

Before saving, the PDF is written to memory using BytesIO, which enables in-memory manipulation without saving to disk. This functionality is demonstrated in appendix 6.1, in the code section from lines 523 to 526.

Next, the PDF is uploaded via FTP to a predefined directory on a webspace, using the upload_pdf_to_ftp() function. This functionality is demonstrated in appendix 6.1, in the code section from lines 529 to 530.

If the upload is successful, the system displays the message "Repair report successfully uploaded to FTP" and redirects the user to the main interface. This functionality is demonstrated in appendix 6.1, in the code section from lines 532 to 533.

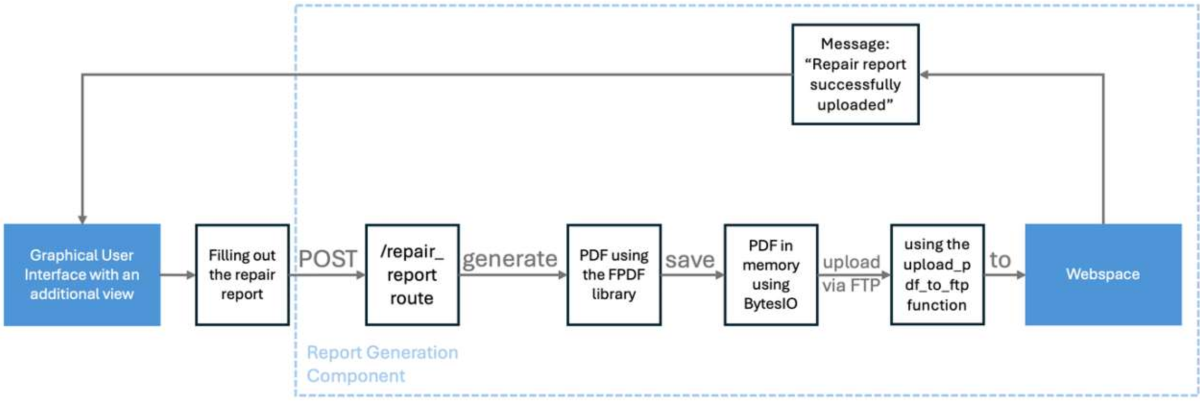


Figure 25: Report Generation Component

Next, the system integrates the Data Integration Service Component, as illustrated in Figure 26. The Data Integration Service Component is triggered through a GET request to the /show-data route. When this route is called, two transfer functions, send_transfer() and send_transfer_joinery(), send data via HTTP POST to the external Nexyo API. Nexyo then forwards the data to another external API, namely the Render Endpoint. This external API, hosted outside the local system, processes the data and returns logs containing relevant information. Based on the presence of the attribute Label_of_boards in the received data, the system classifies the data as joinery data if the attribute is present, otherwise, it is classified as forest data. The data is stored in a MongoDB cluster in a specific database, with joinery and forest data placed into separate collections. This is handled using the PyMongo library.

This functionality is demonstrated in appendix 6.1, in the code sections from lines 169 to 318 and 365 to 373.

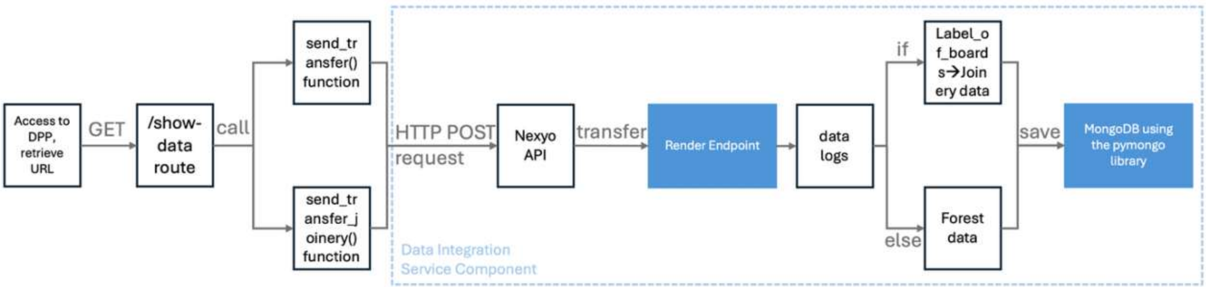


Figure 26: Data Integration Service Component

In parallel, the Document Retrieval Component fetches and matches the documents required for visualization, as illustrated in Figure 27. A joinery document with ID 2 is first retrieved from the MongoDB collection. From this document, board labels are extracted and parsed into searchable IDs. These IDs are then used to perform regex-based queries against forest documents to find all relevant matches. This functionality is demonstrated in appendix 6.1, in the code section from lines 377 to 400.

At the same time, PDF files from two specific folders, /static/repairs and /documents, are loaded from an external webspace using an FTP-based function. Repair PDFs are retrieved using the list_ftp_pdfs() function, which is also used to load document PDFs. The relevant code for list_ftp_pdfs() is shown in appendix 6.1, in the code section from lines 343 to 349.

Additionally, base64-encoded images are retrieved from the webspace for frontend display using load_image_base64(), load_image_base64_logo() and load_image_base64_logo2(). The code for these functions can be found in appendix 6.1, in the code section from lines 320 to 333.

The overall functionality for retrieving repair and document PDFs, as well as loading images, is demonstrated in appendix 6.1, in the code section from lines 410 to 428.

Finally, all retrieved and matched data including forest and joinery information, PDFs and images are rendered in a GUI. This allows users to access and interact with the structured information visually. This functionality is demonstrated in appendix 6.1, in the code section from lines 430 to 455.

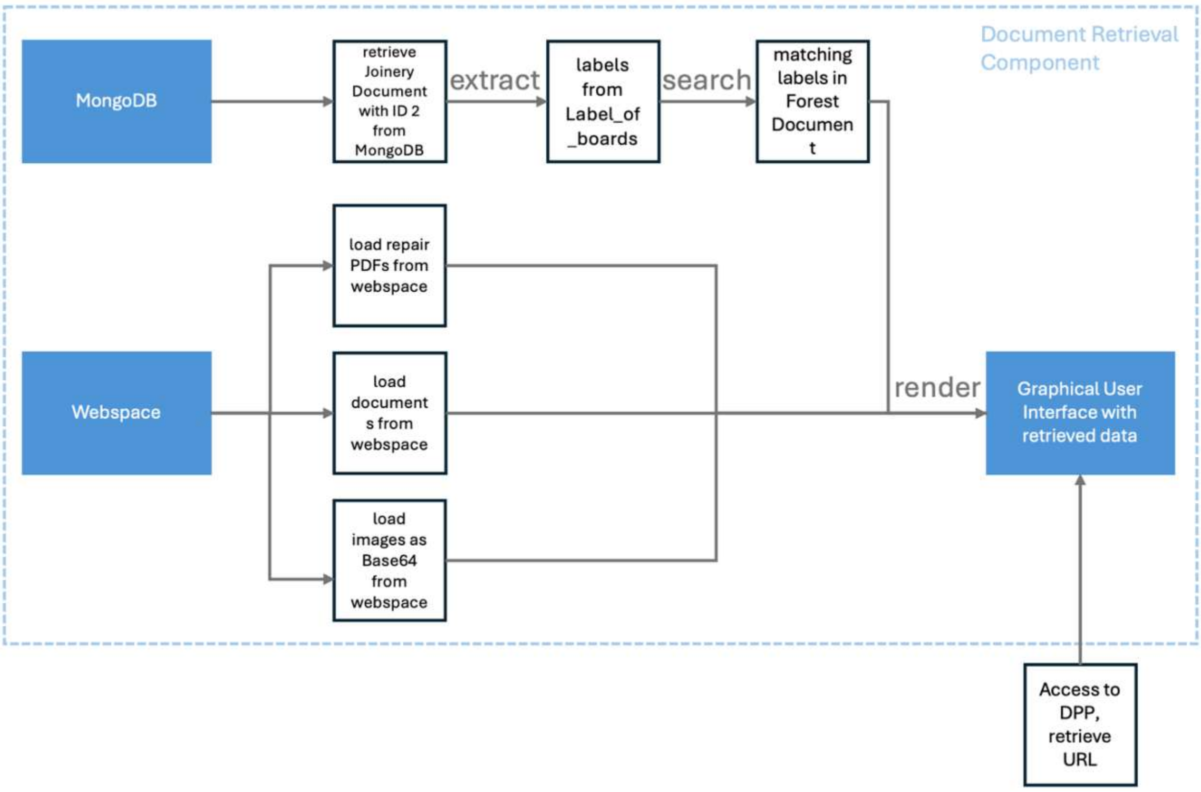


Figure 27: Document Retrieval Component

4.2.3 Code Implementation – DPP Frontend

After the backend handles the core data processing and business logic, the next step is the frontend, which is responsible for presenting the content to the user. The GUI was developed entirely in python and is based on the Flask web framework and the Jinja2 templating engine.

Flask is particularly well suited because it is a lightweight and flexible framework that operates without significant overhead. This means it does not include unnecessary features that would slow down or complicate the application. Flask allows for a quick start while still providing enough flexibility to implement more complex requirements. Thanks to its modular structure, which allows adding specific functions as needed, Flask can be precisely tailored to the requirements of this master thesis (Pallets Projects, n.d.-a).

For the design of the user interface Jinja2 is used. This template engine is tightly integrated with Flask and enables dynamic generation of web pages using HTML. It allows data processed in the backend or retrieved from a database or data storage to be seamlessly integrated into the frontend. By supporting control structures such as loops and conditions within the templates, Jinja2 also makes it easy to create more complex views. Moreover, Jinja2 ensures a clear separation of logic and presentation, the application logic remains in the python code, while HTML is used solely for rendering the content. This separation improves code clarity and significantly simplifies maintenance and future development (Pallets Projects, n.d.-a and Pallets Projects, n.d.-b).

<u>Role-based classification and presentation of data</u>
The presentation of the required data on the website is based on the specific role of the user. A careful classification is necessary to clearly define which data is relevant and important for each user group. This differentiation ensures that each user receives only the information pertinent to them, thereby improving the usability and clarity of the application.

As shown in Figure 28, there are various data providers supplying different types of data. Examples include Forest, Sawmill Arrival, Sawmill Product, Joinery Arrival and Joinery Product. These data providers deliver the information required, serving as the foundation for the application.

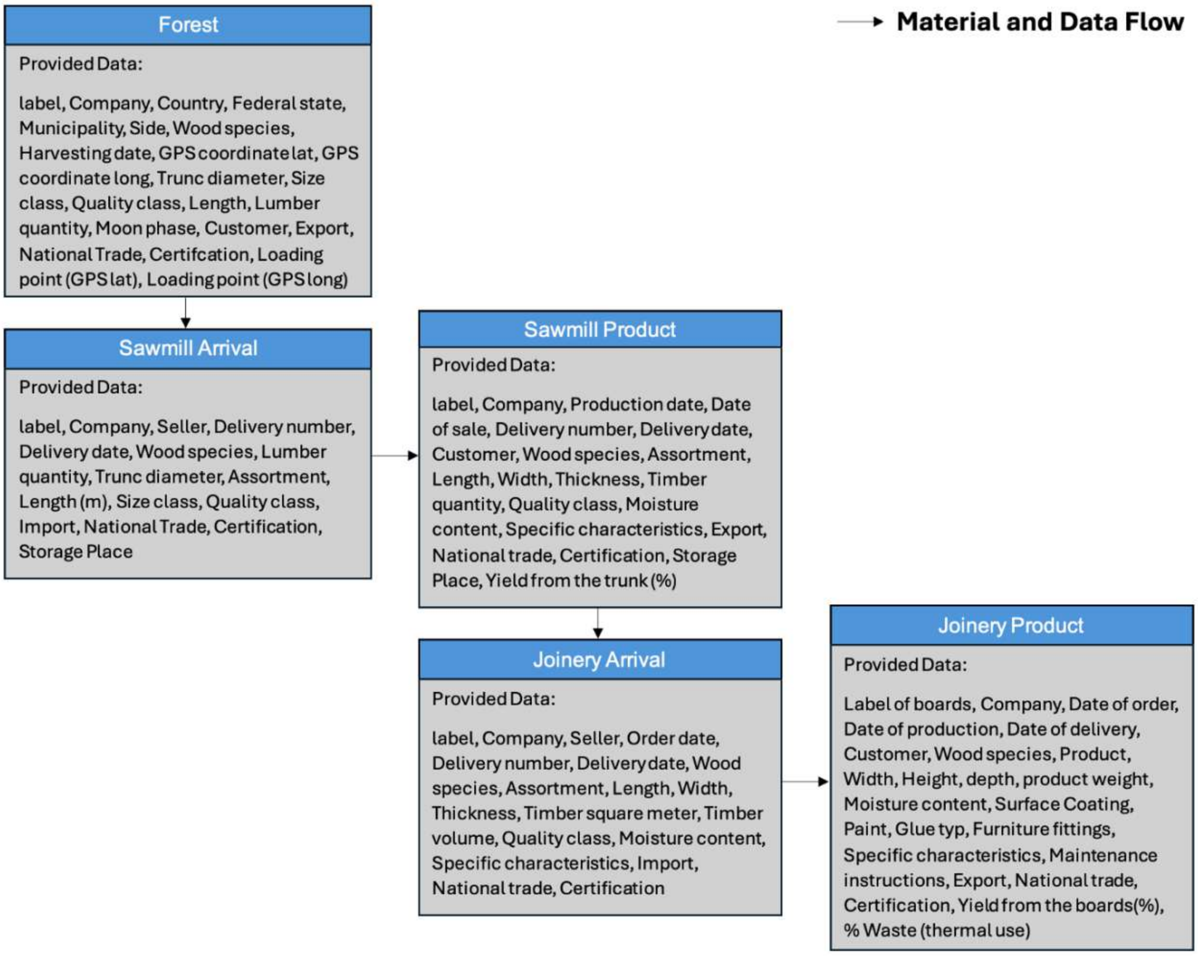


Figure 28: Data provided by upstream stakeholders

The provided data is synthetic, meaning it is based on interviews and expert knowledge from relevant fields. Despite the diverse origins of the data, overlaps occur where different data sources provide essentially the same information. It is particularly important that no changes occur during data transmission to maintain data integrity and reliability.

To accommodate the varying needs of users a comprehensive table was created that clearly shows the relevance of each data type for different user groups. The users are categorized in four main roles which include End User, Repair Service, Recycler and Authority. Table 4 provides a detailed overview of which data is especially relevant for each user group and how the data is allocated accordingly. This clear classification enables targeted and needs-based data provision that optimally meets the individual requirements of the users.

In the GUI, not all relevant data for each user group is displayed. On the one hand, some data was not available. On the other hand, to maintain a user-friendly interface and avoid overloading the page, certain data was intentionally left out. Furthermore, as previously mentioned, the GUI is based on a flexible, modular and extensible architecture, which allows additional data to be integrated with minimal effort.

Table 4: Overview of Data Relevance and Allocation for Different User Groups

Stakeholder	Forest	Sawmillproduct	Joineryproduct
End User	Country, Federal state, Municipality, Wood species, GPS coordinate lat, GPS coordinate long, Quality class, Moon phase, Certification		Date of order, Date of production, Date of delivery, Wood species, Product, Width, Height, depth, product weight, Moisture content, Surface Coating, Paint, Furniture fittings, Specific characteristics, Maintenance instructions, Export, National trade, Certification, Yield from the boards(%), % Waste (thermal use)
Repair Service	label, Company, Country, Federal state, Municipality, Wood species, Harvesting date, GPS coordinate lat, GPS coordinate long, Quality class, Moon phase, Export, National Trade, Certification	label, Company, Production date, Delivery number, Wood species, Length, Width, Thickness, Quality class, Moisture content, Specific characteristics, Export, National Trade, Certification	Label of boards, Company, Date of production, Date of delivery, Customer, Wood species, Product, Width, Height, depth, product weight, Moisture content, Surface Coating, Paint, Glue typ, Furniture fittings, Specific characteristics, Maintenance instructions, Export, National trade, Certification, Yield from the boards(%), % Waste (thermal use)
Recycler	label, Company, Country, Federal state, Municipality, Side, Wood species, Harvesting date, Quality class, Moon phase, Certification	label, Company, Production date, Delivery number, Wood species, Length, Width, Thickness, Quality class, Moisture content, Specific characteristics, Certification, Yield from the trunk (%)	Label of boards, Company, Date of production, Date of delivery, Customer, Wood species, Product, Width, Height, depth, product weight, Moisture content, Surface Coating, Paint, Glue typ, Furniture fittings, Maintenance instructions,

			Certification, Yield from the boards(%), % Waste (thermal use)
Authority	label, Company, Country, Federal state, Municipality, Wood species, Harvesting date, Customer, Export, National Trade, Certification		Company, Wood species, Product, Width, Height, depth, product weight, Moisture content, Surface Coating, Paint, Specific characteristics, Export, National trade, Certification, Yield from the boards(%), % Waste (thermal use)

Creation of the GUI

The GUI was developed using python, the flask web framework and the Jinja2 templating system, as mentioned earlier. Technically, the user interface is divided into two main areas:

- Style Section: Defines the visual appearance, including colours, fonts, layout structure and the sidebar
- Body Section: Contains the content part of the page, which is dynamically generated based on provided data

Style Section

The HTML document begins with a declaration that defines it as an HTML5 document. The root <html> element specifies that the language of the content is English. This is shown in appendix 6.2, in the code section from lines 1 to 2. Inside the <head> section, several important configurations are made.

First, the character encoding is set to UTF-8 which ensures that all characters including special symbols and non-English letters are displayed correctly. Next, the viewport is configured to make the page responsive on different screen sizes. This means the layout will adapt properly on smartphones, tablets and desktops. The title of the page is set to "Chair" which will appear in the browser tab. A stylesheet for the Leaflet library is also linked. Leaflet is used to display interactive maps and the Cascading Style Sheets (CSS) file ensures that the map will be styled appropriately when it is added to the page. This is demonstrated in appendix 6.2, in the code sections from lines 4 to 7.

Figure 29 presents the structural layout of the DPP GUI including the main content area and a collapsible sidebar. It demonstrates how CSS media queries adjust the layout for both large screens (e.g., desktops) and small screens (e.g., smartphones) ensuring usability and a consistent design across different device types.

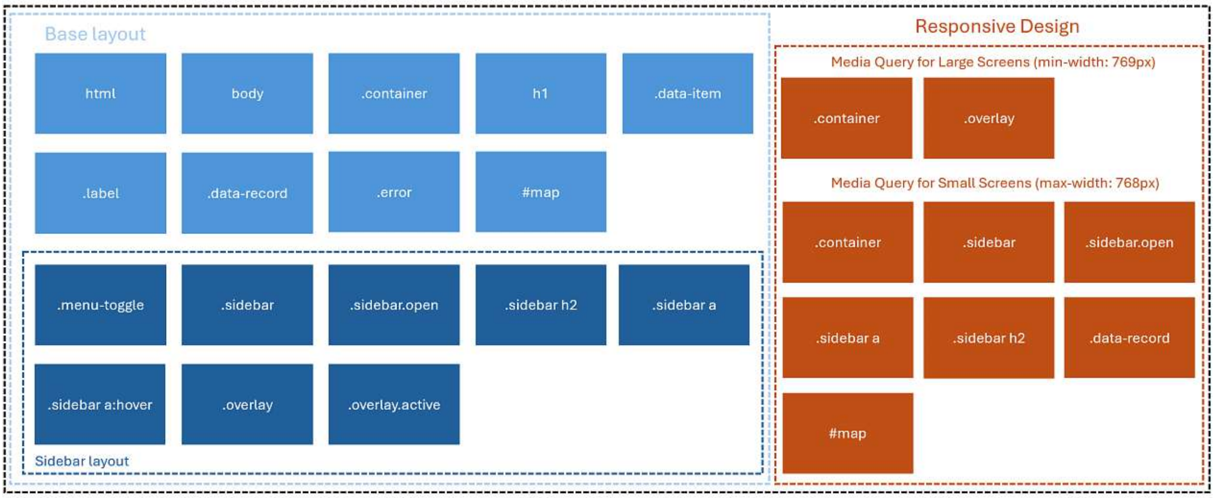


Figure 29: Base and Sidebar Layout with Responsive Media Queries for Large and Small Screens

Starting with the basic layout and sidebar layout, it consists of the following elements:

- html: The html selector applies the CSS property “scroll-behavior: smooth” enabling smooth animated scrolling when navigating through the webpage. This improves user experience by providing fluid transitions instead of abrupt jumps. The corresponding code can be found in appendix 6.2, lines 9 to 11.
- body: The body element is styled with a clean sans-serif font (Arial), zero margin for full-width layout and a light gray background colour (#f4f4f4) to create a neutral and modern page base. The corresponding code can be found in appendix 6.2, lines 13 to 17.
- .container: The .container class defines a white background area with padding for content spacing. It also includes a smooth transition effect on the left margin to allow for animated shifts, for example when a sidebar is toggled. The corresponding code can be found in appendix 6.2, lines 19 to 23.
- h1: Headings of level one (h1) are centered horizontally to emphasize the main page title. The corresponding code can be found in appendix 6.2, lines 25 to 27.
- .data-item: Each .data-item has vertical margin spacing (10px 0), meaning 10 pixels of space above and below and no horizontal spacing (left and right). This visually separates individual data sections from one another. The corresponding code can be found in appendix 6.2, lines 29 to 31.
- .label: The .label class uses bold font weight to highlight text labels. The corresponding code can be found in appendix 6.2, lines 33 to 35.
- .data-record: Containers with the .data-record class have spacing below, padding, a light border and rounded corners to visually group related information blocks with a clean and accessible style. The corresponding code can be found in appendix 6.2, lines 37 to 42.
- .error: The .error class displays text in red, centers it horizontally and makes it bold to clearly indicate error messages to users. The corresponding code can be found in appendix 6.2, lines 44 to 48.
- #map: The map container has a fixed height of 500 pixels, top margin for spacing from preceding content and rounded corners for a polished look. The corresponding code can be found in appendix 6.2, lines 50 to 54.
- .menu-toggle: The .menu-toggle button is fixed in the top-left corner with a large font size for visibility, a dark background, white text, padding and rounded edges. Its opacity changes smoothly to provide interactive feedback. The CSS property opacity controls an element’s transparency with values from 0 (fully transparent) to 1 (fully opaque). A high z-index ensures the button stays above other elements. The corresponding code can be found in appendix 6.2, lines 57 to 69.
- .sidebar: The sidebar, as shown in Figure 30 with the three-line (hamburger) icon on the left, is fixed to the left side of the viewport but initially hidden by

being positioned off-screen (left: -260px). It has a fixed width, full height, dark background, white text and padding. The sidebar uses a smooth transition to slide in and out and is layered above other content with a high z-index. The corresponding code can be found in appendix 6.2, lines 71 to 82.

- .sidebar.open: When the .open class is applied, the sidebar shifts its left position to zero, making it visible within the viewport. The open sidebar is illustrated in Figure 31. The corresponding code can be found in appendix 6.2, lines 84 to 86.
- .sidebar h2: The sidebar's secondary heading has no top margin, a medium font size and a subtle bottom border for separation from navigation links. The corresponding code can be found in appendix 6.2, lines 88 to 93.
- .sidebar a: Sidebar links are block elements with white text, no underline and vertical padding for easy clicking. They span the sidebar width. The corresponding code can be found in appendix 6.2, lines 95 to 100.
- .sidebar a:hover: On hover, sidebar links change background color and add left padding for a subtle sliding effect, indicating interactivity. The corresponding code can be found in appendix 6.2, lines 102 to 105.
- .overlay: The .overlay is a full-screen fixed layer with semi-transparent black background, initially hidden. It covers the page content to focus user attention on active elements like the sidebar. The corresponding code can be found in appendix 6.2, lines 107 to 116.
- .overlay.active: When active, the overlay is displayed (display: block), dimming the background and preventing interaction with underlying content. The corresponding code can be found in appendix 6.2, lines 118 to 120.

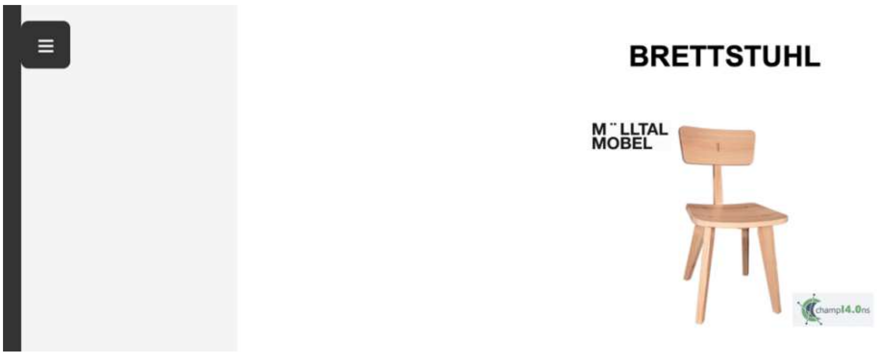


Figure 30: Sidebar in its hidden state, indicated by the hamburger icon on the left

Figure 31: Sidebar in its open state

To ensure optimal usability across a variety of devices, the base layout is enhanced with responsive design techniques. This is achieved through CSS media queries that adapt the page structure and styling for both large and small screen sizes, maintaining functionality and visual consistency.

Media query for large screens (min-width: 769px):

- .container: The container element receives a left margin of 260 pixels to make space for the visible sidebar on larger screens, ensuring the content is not overlapped. The corresponding code can be found in appendix 6.2, lines 123 to 125.
- .overlay: The overlay element is forcibly hidden (display: none !important) to allow unobstructed interaction with the main content, as the sidebar is always visible on large screens. The corresponding code can be found in appendix 6.2, lines 127 to 129.

Media query for small screens (max-width: 768px):

- .container: On smaller screens, the .container element removes any left margin by setting margin-left: 0, allowing it to use the full width of the viewport. Additionally, padding: 15px is added to create sufficient spacing between the content and the container’s edges, improving readability on compact displays. The corresponding code can be found in appendix 6.2, lines 133 to 136.
- .sidebar: The .sidebar is resized to occupy 80% of the screen width, making it more suitable for smaller devices. It is initially hidden by positioning it completely off-screen to the left using left: -100%, preventing it from covering the main content until needed. The corresponding code can be found in appendix 6.2, lines 138 to 141.
- .sidebar.open: When the sidebar receives the .open class, its left position is reset to 0 which slides it into the visible area of the screen. This enables responsive toggling of the navigation menu on mobile devices. The corresponding code can be found in appendix 6.2, lines 143 to 145.

- .sidebar a: On small screens sidebar links are styled with an increased font size of 16px and vertical padding of 15px. This enhances readability and ensures easier interaction on touch devices. The corresponding code can be found in appendix 6.2, lines 147 to 150.
- .sidebar h2: The heading level 2 (h2) inside the sidebar is resized to 18px to ensure that section titles remain legible and visually distinct, even on small displays. The corresponding code can be found in appendix 6.2, lines 152 to 154.
- .data-record: The .data-record element groups related data entries. It receives 12px of padding and a reduced margin-bottom of 15px to maintain compact but clear spacing between items. The corresponding code can be found in appendix 6.2, lines 156 to 159.
- #map: The height of the #map container is reduced to 300px and the top margin is adjusted to 20px. This ensures that the map remains functional and visually balanced without taking up excessive space on smaller screens. The corresponding code can be found in appendix 6.2, lines 161 to 164.

Body Section

The body section of the GUI is modularly structured into individual visual containers. Each container represents a specific functional area of the DPP. This clear separation into independent, logically organized modules ensures clarity, enhances user navigation and simplifies code maintenance. The overall layout and structure of the GUI are illustrated in Figure 32.

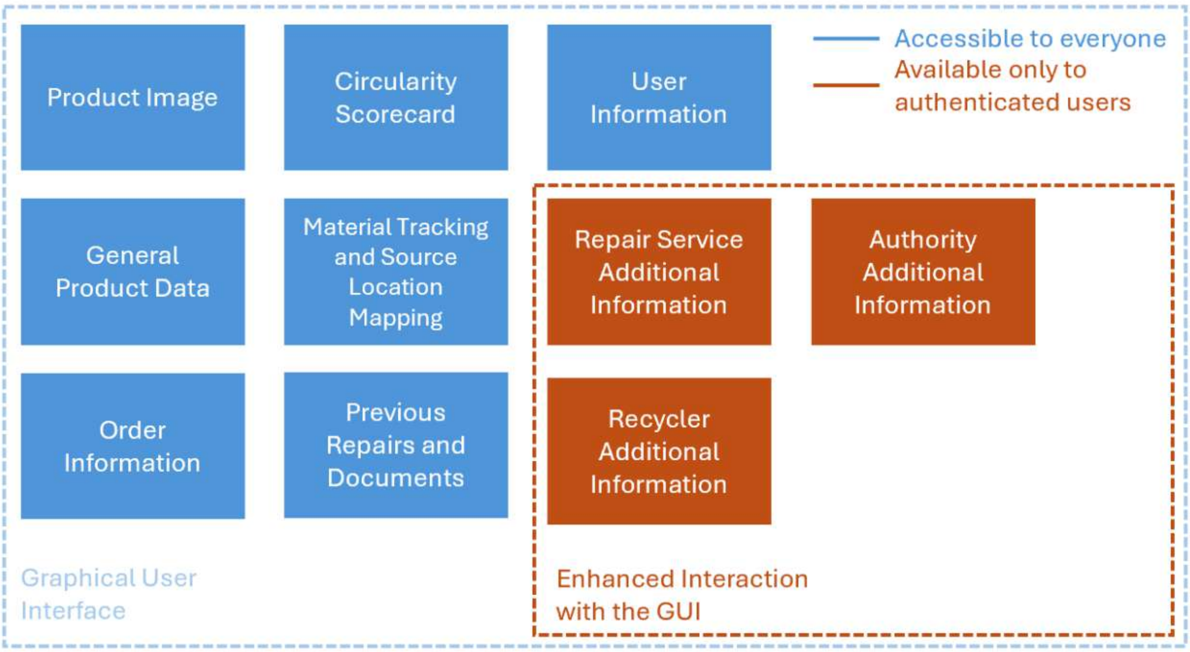


Figure 32: Modular Structure of the GUI

The GUI is divided into distinct visual containers, each representing a functional section of the DPP. Blue containers denote universally accessible elements, while orange containers represent extended interaction modules available only to authenticated users (e.g., repair services, recyclers and authorities).

After scanning the QR-Code and being redirected to the page, the following content is displayed to everyone.

Product image: A photo of the “Brettstuhl” serves as the central visual element, as shown in Figure 33. The image is displayed within a container that also includes two clickable logos linking to related partner websites. The main product image is centered, while the partner logos are positioned at the top-left and bottom-right corners of the container. This functionality is demonstrated in appendix 6.2, in the code section from lines 197 to 209.

General product data: This section provides key specifications and attributes of the product, including the product name, dimensions (width, height, depth), weight, surface coating, paint, specific characteristics and certifications, which are displayed in Figure 33. The information is dynamically retrieved from the database (MongoDB). Each product entry displays its respective properties alongside descriptive labels. If a value is missing, a placeholder such as “no” is shown. In cases where no product data is available, the user is informed accordingly. This functionality is demonstrated in appendix 6.2, in the code section from lines 211 to 231.

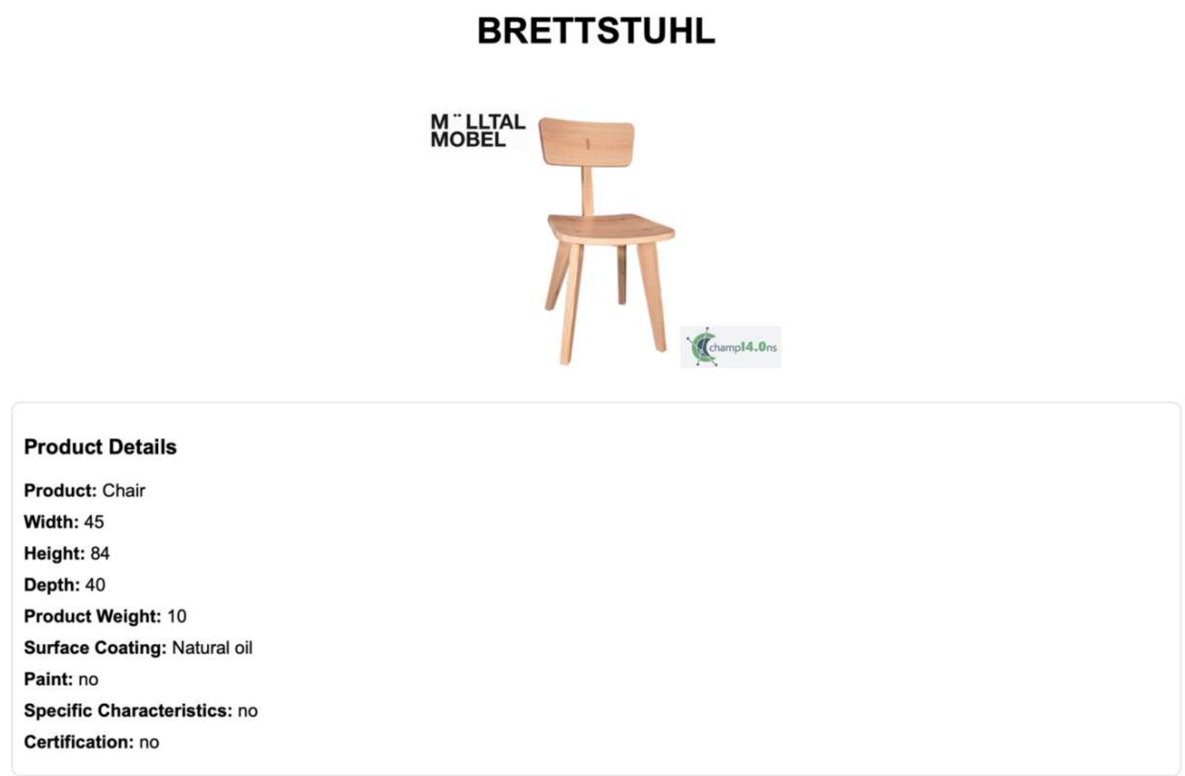


Figure 33: Central product image featuring two clickable logos and displaying the general product data

Order information: This section presents essential order-related details such as the date of order and the date of delivery, as shown in Figure 34. The data is dynamically loaded from the database (MongoDB), with each entry displaying the relevant fields accompanied by descriptive labels. If certain information is missing, a placeholder like “no” is used. If no order data is available, a corresponding message is displayed to the user. The implementation is shown in appendix 6.2, in the code section from lines 233 to 246.

Circularity scorecard as KPI: The circularity scorecard represents a KPI within the CE domain, as shown in Figure 34. The displayed "Repairability Score" quantifies how easily the product can be maintained or repaired, thereby supporting sustainable design and lifecycle management. While no official repairability index exists for furniture, the score conceptually follows criteria from the French Repairability Index for electronics, which evaluates factors such as documentation availability, ease of disassembly, spare parts accessibility and pricing (German Environment Agency, n.d.). Applying similar principles to furniture promotes transparency and encourages product designs aligned with circularity goals. This implementation is detailed in appendix 6.2, in the code section from lines 249 to 266.

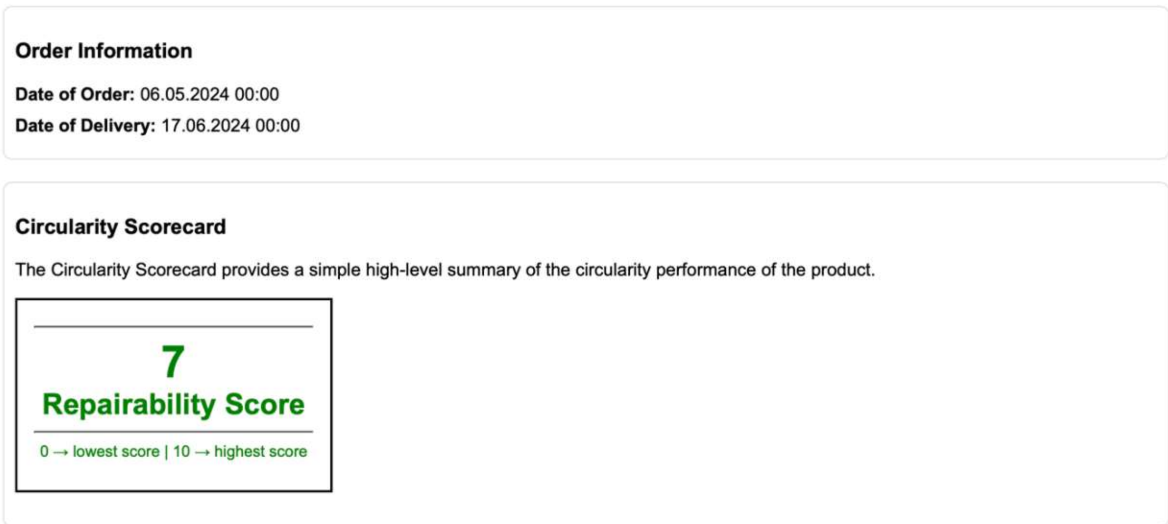


Figure 34: Order information and circularity scorecard

Material tracking and source location mapping: This section displays the origin and composition of the materials used in the product. For each tree, details such as country, federal state, municipality, wood species, quality class and certification are displayed. The data is dynamically loaded from the MongoDB database. If certain values are missing, a placeholder like “no” is displayed. If no data is available, the user is informed accordingly. In addition to the data records (code lines 268 to 284 in appendix 6.2), the Global Positioning System (GPS) coordinates of each tree’s source location are visualized on an interactive map using Leaflet.js (code lines 286 to 293 and 468 to 506 in appendix 6.2). This map provides a clear geographic representation of the material origins, enhancing supply chain transparency and traceability. Both the tree data records and the interactive map are shown in Figure 35.

Tree 1 Data Record

Country: austria
Federal State: carinthia
Municipality: 9833 Rangersdorf
Wood Species: fir
Quality Class: B
Certification: no

Tree 2 Data Record

Country: austria
Federal State: carinthia
Municipality: 9833 Rangersdorf
Wood Species: fir
Quality Class: B
Certification: no

GPS coordinates of trees in the forest

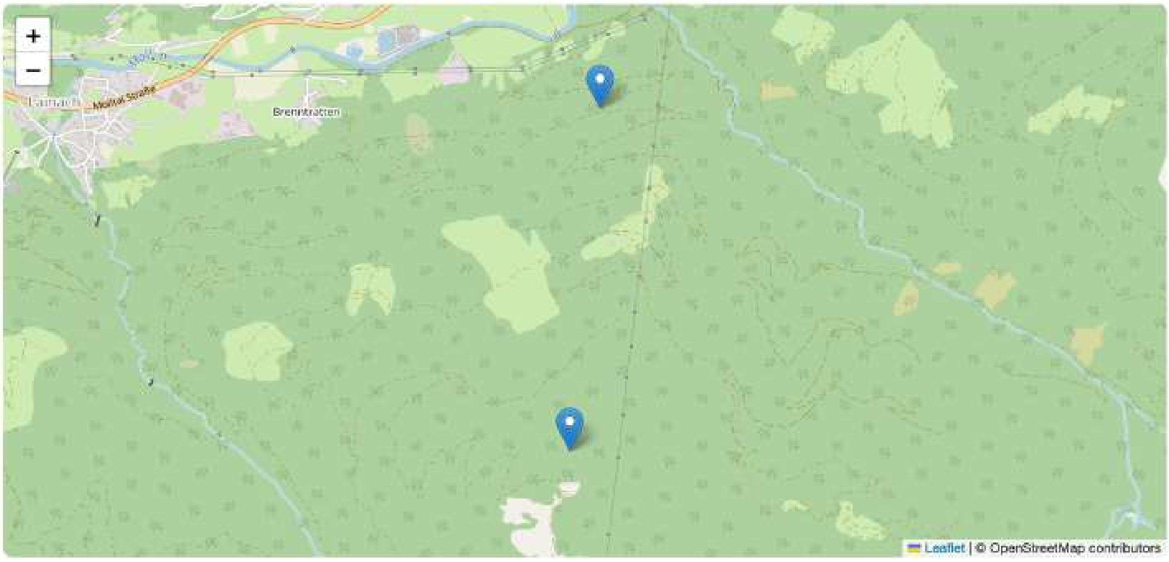


Figure 35: Tree data records and the GPS coordinates of each tree's source location visualized on an interactive map

Previous repairs and documents: This section lists previous repair reports related to the product, providing downloadable links when available. If no reports are present, a corresponding message notifies the user. Additionally, key documents such as maintenance instructions, assembly and disassembly guides, warranty information and environmental guidelines are linked for easy access. Figure 36 illustrates this part of the interface, showing both the repair reports and associated documents. This functionality is demonstrated in appendix 6.2, in the code section from lines 295 to 316.

User information: This section invites users such as repair services, recyclers or authorities to log in for accessing more detailed information or to add and edit documents related to the product. Figure 36 displays this part of the interface, highlighting the option for stakeholders to log in. This functionality is demonstrated in appendix 6.2, in the code section from lines 318 to 326.

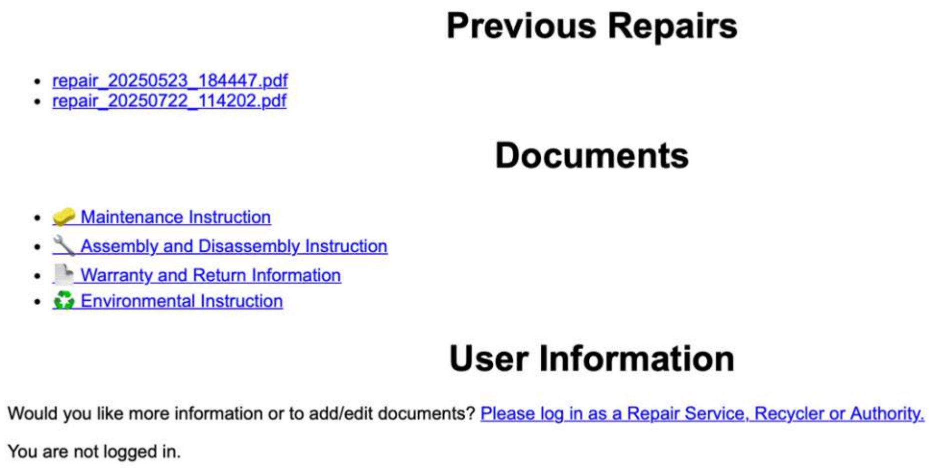
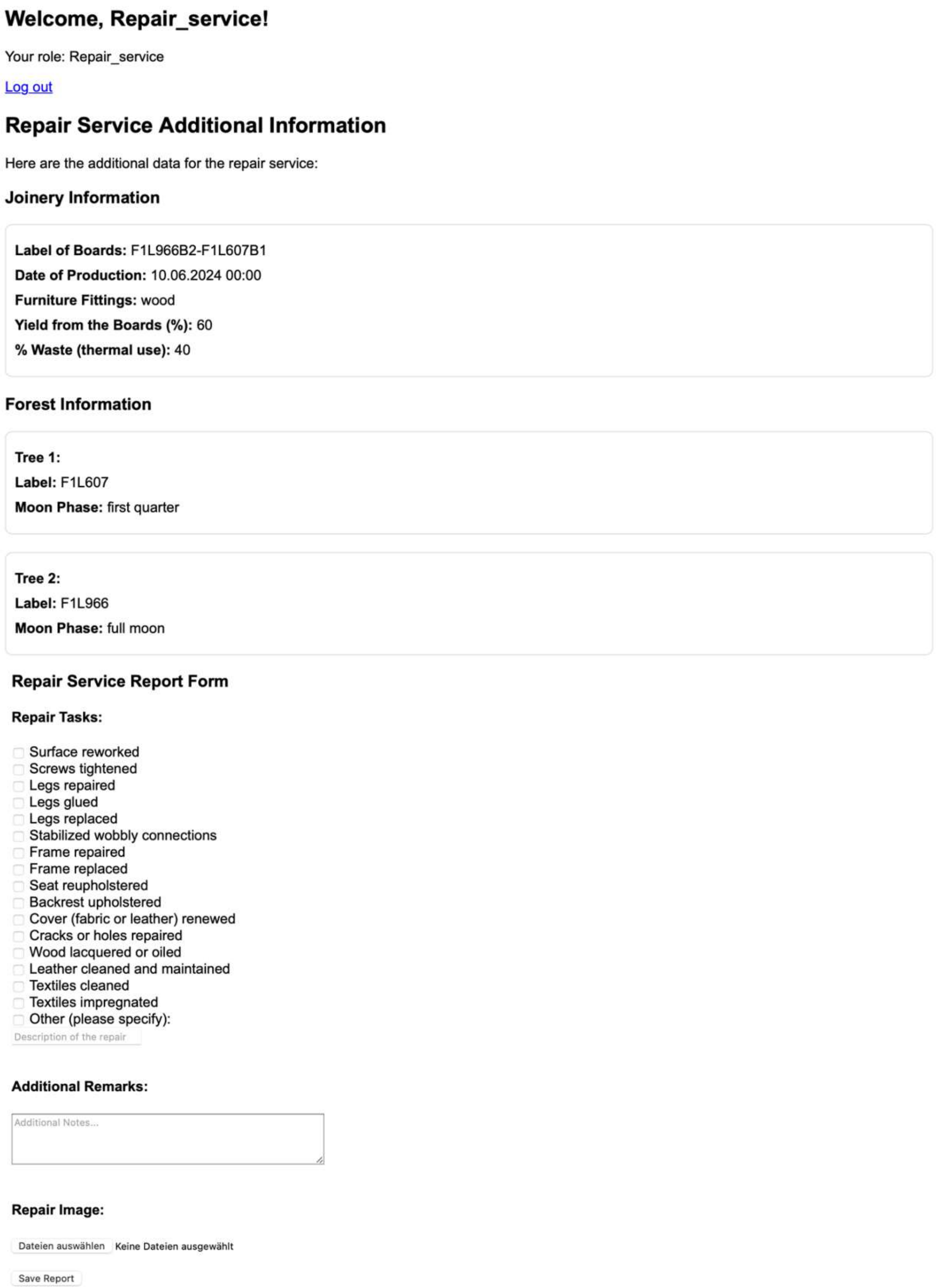


Figure 36: Previous repairs and documents along with user information for logging in

After successful login authentication by entering a username and password, as shown in Figure 37, users experience enhanced interaction with the GUI.

Figure 37: Login process via username and password

<u>Repair service additional information:</u> This section is accessible only to authenticated users with the role of “repair service”. Figure 38 displays detailed joinery and forest data related to the product, such as board labels, production dates, furniture fittings, waste percentages and specific tree information including labels and moon phases. If no data is available, corresponding messages are shown. Additionally, repair services can submit detailed repair reports through an interactive form featuring predefined repair tasks, a field for additional comments and the option to upload repair images, as shown in Figure 38. This functionality is demonstrated in appendix 6.2, in the code section from lines 333 to 396.



Welcome, Repair_service!

Your role: Repair_service

Log out

Repair Service Additional Information

Here are the additional data for the repair service:

Joinery Information

Label of Boards: F1L966B2-F1L607B1
Date of Production: 10.06.2024 00:00
Furniture Fittings: wood
Yield from the Boards (%): 60
% Waste (thermal use): 40

Forest Information

Tree 1:
Label: F1L607
Moon Phase: first quarter

Tree 2:
Label: F1L966
Moon Phase: full moon

Repair Service Report Form

Repair Tasks:

- Surface reworked
- Screws tightened
- Legs repaired
- Legs glued
- Legs replaced
- Stabilized wobbly connections
- Frame repaired
- Frame replaced
- Seat reupholstered
- Backrest upholstered
- Cover (fabric or leather) renewed
- Cracks or holes repaired
- Wood lacquered or oiled
- Leather cleaned and maintained
- Textiles cleaned
- Textiles impregnated
- Other (please specify):

Description of the repair

Additional Remarks:

Additional Notes...

Repair Image:

Dateien auswählen Keine Dateien ausgewählt

Save Report

Figure 38: Repair service interface showing additional data and an interactive form for submitting repair reports

<u>Recycler additional information:</u> This section provides recyclers with access to detailed joinery and forest data relevant to the product, as shown in Figure 39. For joinery, information such as board labels, production dates, moisture content, yield percentages and waste used for thermal purposes are displayed. Forest data includes labels and moon phase information for each recorded tree. If data is unavailable in either category, a corresponding message informs the user. This functionality ensures recyclers have the necessary information to support material recovery and sustainable recycling processes. This functionality is demonstrated in appendix 6.2, in the code section from lines 398 to 429.

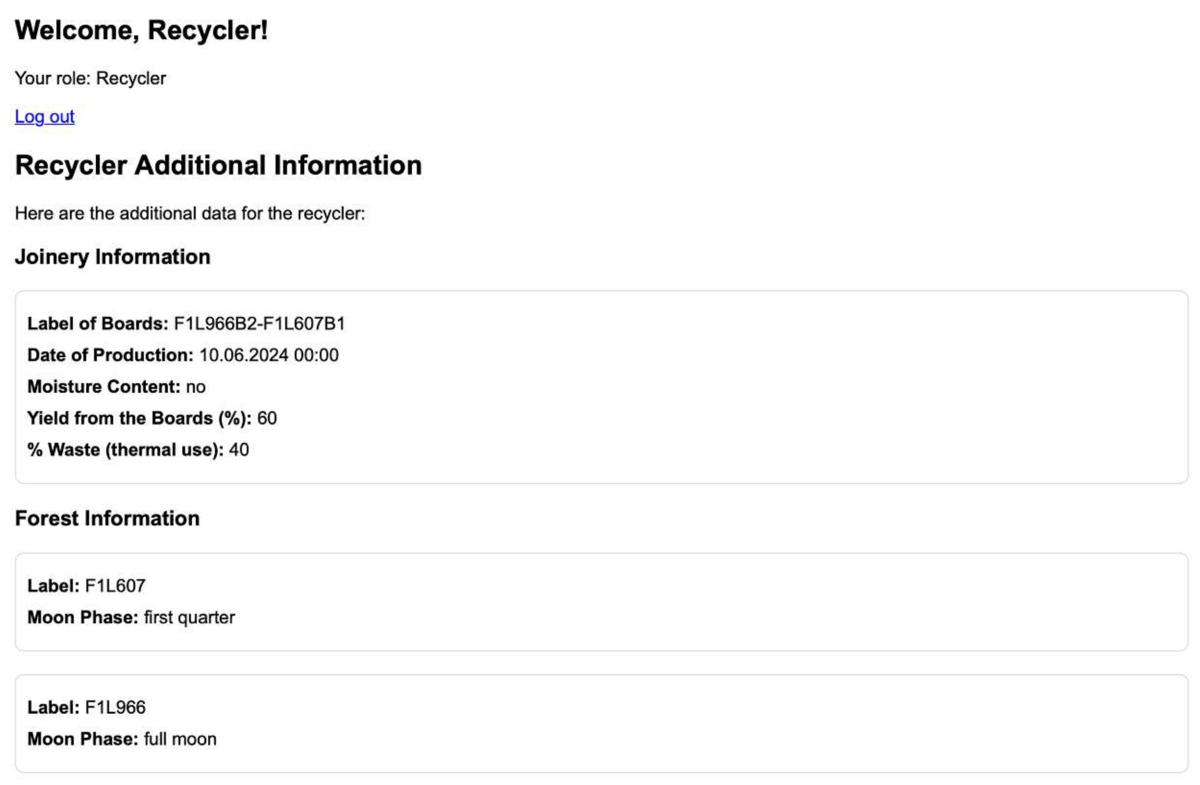


Figure 39: Recycler interface displaying additional information

<u>Authority additional information:</u> This section provides authorities with detailed joinery and forest data related to the product, as shown in Figure 40. Joinery data includes board labels, export information and national trade details. Forest data presents tree labels along with precise GPS coordinates (latitude and longitude) for each recorded tree. If data is missing in either category an appropriate message is displayed. This ensures authorities have comprehensive information to support regulatory oversight and traceability. This functionality is demonstrated in appendix 6.2, in the code section from lines 431 to 466.

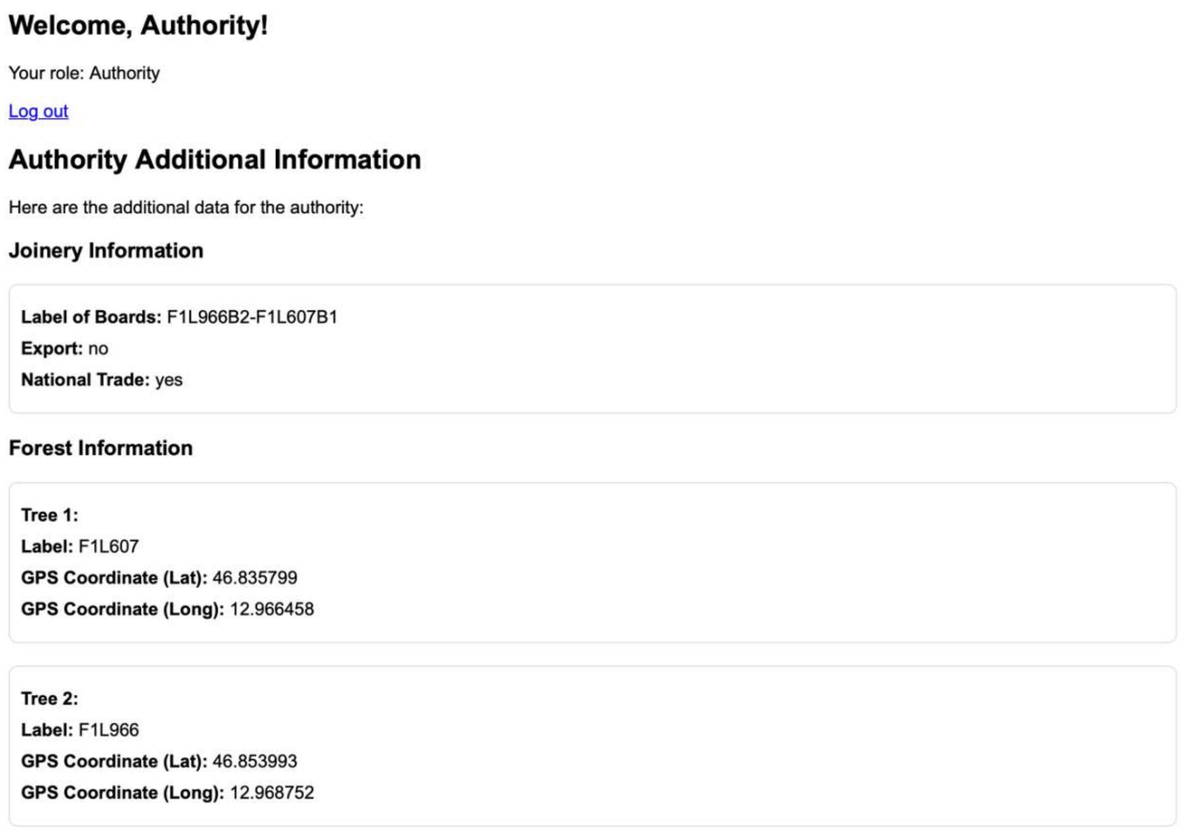


Figure 40: Authority interface displaying additional information

4.3 Evaluation of the DPP GUI

As part of the development process of the DPP, a structured user evaluation was conducted to assess the usability and user experience of the GUI. The goal was to gather feedback from a relevant group of stakeholders, particularly from research institutions, companies along the value chain, IT or service companies and other affiliated users. The evaluation was carried out using a combination of the System Usability Scale and the User Experience Questionnaire, as outlined in Appendix 6.4. Both methods were selected to capture complementary aspects of usability and user experience. In total, 22 individuals completed the survey.

Participants were asked to indicate their institutional affiliation at the beginning of the questionnaire. Figure 41 shows that the majority (59,1%) were affiliated with research institutions, followed by 27,3% who selected other, 9,1% from companies within a value chain and 4,5% from IT or service companies. Although the sample is predominantly from research institutions, it still reflects a range of perspectives from both academia and industry.

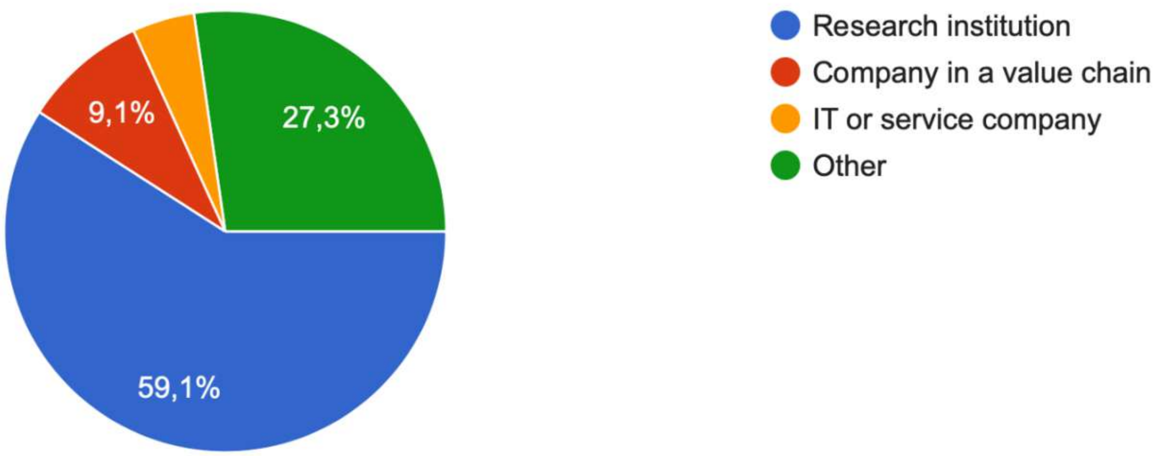


Figure 41: Distribution of respondents by organizational affiliation

4.3.1 Results of the System Usability Scale

The SUS is a standardized and widely used instrument for assessing perceived usability. It consists of 10 statements, alternating between positive and negative wording, rated on a five-point scale from "Strongly disagree" (1) to "Strongly agree" (5).

Rather than interpreting each item individually, the responses were grouped thematically to provide a more holistic understanding of how users experienced the system. The analysis focuses on three central aspects:

- Usability

- Learnability
- Intent of use

<u>Usability</u>

The users rated the DPP GUI predominantly positively in terms of usability, as shown in Figure 42. “Easy to use” received the highest average rating of 4,45, indicating that users found the interface straightforward and intuitive. “Well integrated functions” scored 3,91, reflecting a satisfactory integration of features within the GUI, while “confident use” achieved a rating of 4,41, demonstrating users’ confidence in operating the DPP GUI.

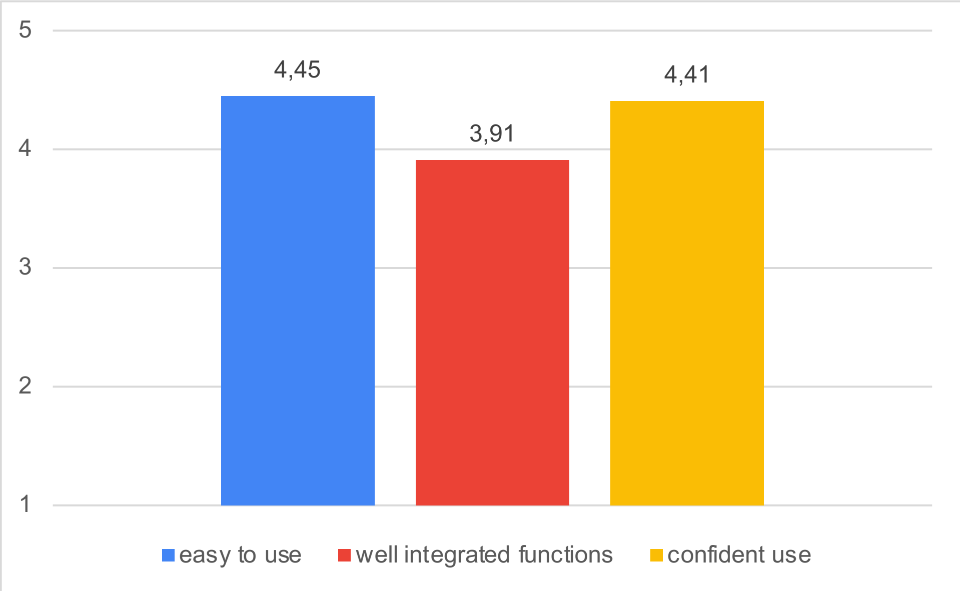


Figure 42: Average user ratings of positive usability aspects

In contrast, ratings for potential negative usability factors were consistently low, as illustrated in Figure 43. “Unnecessary complexity” was rated at 1,45, suggesting that users rarely perceived the interface as overly complex. “Inconsistency” received the lowest average rating of 1,36, indicating a high level of design consistency and “cumbersome to use” also scored 1,45, showing that users did not consider the interface difficult to operate.

Overall, these results demonstrate that the DPP GUI is predominantly perceived as user-friendly and well-structured, with usability strengths significantly outweighing any identified weaknesses. However, the score for “Well integrated functions” is slightly lower than other positive aspects, indicating potential areas for improvement in feature integration and user guidance. Addressing these minor shortcomings could further enhance the overall usability and effectiveness of the GUI.

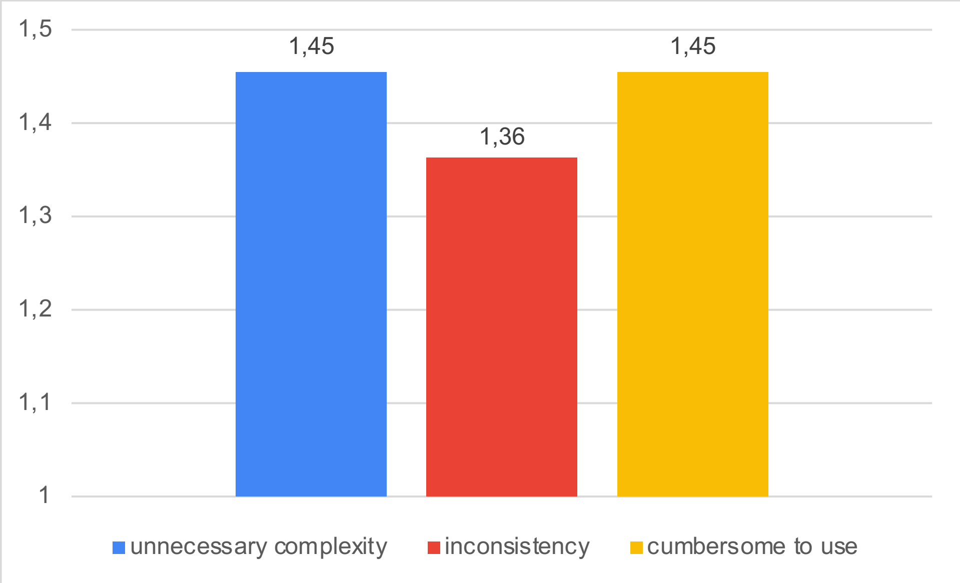


Figure 43: Average user ratings of negative usability aspects

Learnability

The learnability results for the DPP GUI are presented in Figure 44 (positive aspects) and Figure 45 (negative aspects). The interface was rated very positively in terms of ease of learning. “Quick to learn” achieved an exceptionally high average rating of 4,64, indicating that users were able to become familiar with the DPP GUI quickly and without major difficulty.

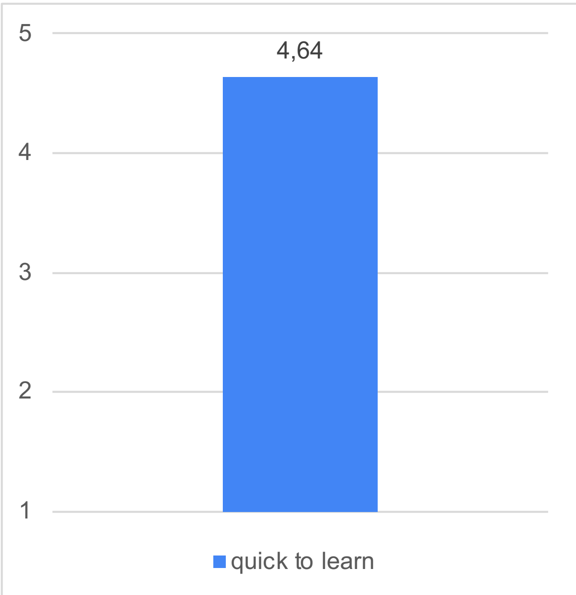


Figure 44: Average user ratings of positive aspects related to learnability

In contrast, the negative aspects of learnability were rated very low, pointing to minimal barriers during initial use, as shown in Figure 45. “Need for technical support” received a low average rating of 1,14, suggesting that users generally did not require external assistance to operate the interface. Similarly, high learning effort was rated at only 1,27, further confirming that users perceived the GUI as intuitive and easy to understand.

Overall, the results demonstrate that the DPP GUI offers a high level of learnability, making it accessible and easy to adopt without the need for extensive training or support.

Figure 45: Average user ratings of negative aspects related to learnability

<u>Intent of use</u>

Figure 46 presents the user feedback regarding their “willingness to use the DPP GUI frequently”. With an average rating of 3,86, users generally showed a positive inclination toward regular use of the interface. However, since the score is below 4, there is potential for improvement in terms of visual appeal, engagement and overall user experience. Possible limiting factors may include the current functional focus of the GUI, relatively modest visual design or lack of interactive incentives. Targeted enhancements to increase attractiveness, intuitive interaction and user guidance could further encourage more frequent use.

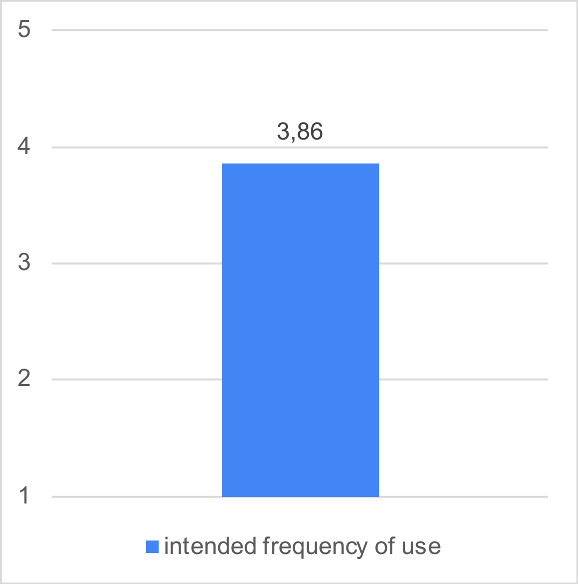


Figure 46: Average user ratings of the willingness to use the DPP GUI frequently

4.3.2 Results of the User Experience Questionnaire

In addition to the SUS, the UEQ was employed to evaluate the subjective impression of the user experience. The UEQ measures emotional and functional quality aspects using bipolar adjective pairs rated on a seven-point semantic differential scale, ranging from 1 (very left) to 7 (very right). The following eight attribute pairs were included:

- boring - exciting
- not understandable - understandable
- inefficient - efficient
- confusing - clear
- impractical - practical
- conventional - innovative
- discouraging - attractive
- unpredictable - predictable

In line with the SUS analysis, the UEQ responses were also grouped thematically to provide a more comprehensive understanding of how users perceived the DPP GUI. The evaluation focuses on four key dimensions:

- Stimulation
  - boring - exciting
  - conventional - innovative
  - discouraging - attractive
- Perspicuity
  - not understandable - understandable
  - confusing - clear

- Efficiency
  - inefficient - efficient
  - impractical - practical
- Dependability
  - unpredictable - predictable

Stimulation

The stimulation dimension reflects how exciting, innovative and appealing users perceived the DPP GUI. As shown in Figure 47, the ratings were overall positive, indicating that the system was experienced as engaging and motivating. On the scale from “boring to exciting,” the interface received an average rating of 4,55, suggesting that users found the GUI more stimulating than dull. The “conventional to innovative” scale was rated at 4,27, indicating a moderate tendency toward perceiving the system as innovative. The highest score was achieved on the “discouraging to attractive” scale, with an average rating of 4,59, showing that users generally considered the GUI visually and functionally appealing. While these values are clearly on the positive side, the DPP GUI already provides a strong level of stimulation but still holds potential for further improvement, particularly in enhancing its innovative qualities and maximizing its attractiveness to users.

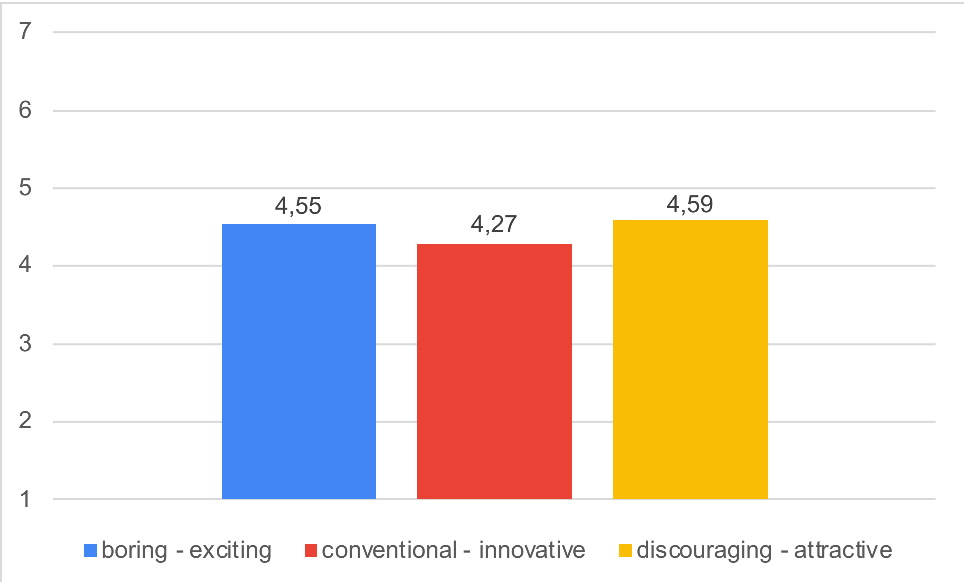


Figure 47: Average user ratings for the stimulation dimension of the DPP GUI

Perspicuity

The perspicuity dimension reflects how clear and understandable users found the DPP GUI. As shown in Figure 48, the ratings were very positive, indicating that users generally perceived the system as easy to understand and navigate. The scale from “not understandable to understandable” received an exceptionally high average rating of 6,32, demonstrating that users found the interface highly comprehensible. Similarly, the scale from “confusing to clear” was rated at 5,95, further confirming that the GUI was seen as clear and straightforward to use. These results highlight the DPP GUI’s strength in providing a transparent and user-friendly experience, facilitating efficient interaction without confusion.

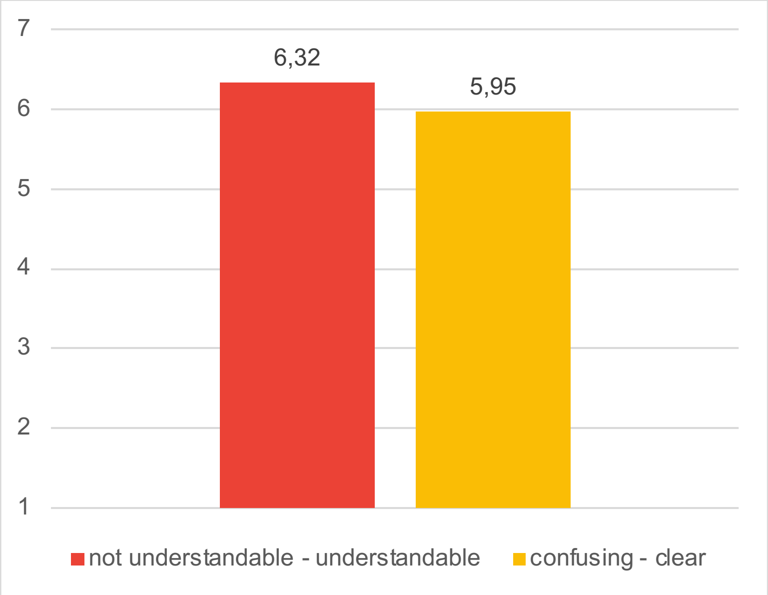


Figure 48: Average user ratings for the perspicuity dimension of the DPP GUI

<u>Efficiency</u>

The efficiency dimension assesses how effectively and practically users were able to interact with the DPP GUI. As shown in Figure 49, the ratings indicate a strong positive perception of the system's functional usability. The scale from "inefficient to efficient" received an average rating of 5,64, suggesting that users found the interface to support efficient task completion. Additionally, the scale from "impractical to practical" achieved an even higher rating of 5,91, indicating that the system was perceived as functionally appropriate and well-suited to user needs. These results demonstrate that the DPP GUI provides an effective and practical environment for accomplishing tasks, reinforcing its suitability for everyday use in the context of the DPP.

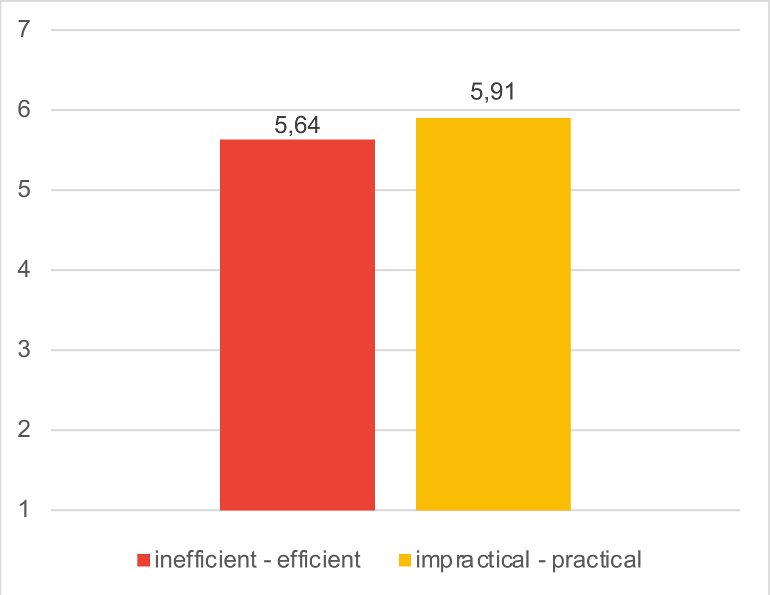


Figure 49: Average user ratings for the efficiency dimension of the DPP GUI

Dependability

The dependability dimension reflects how reliable and predictable users found the behavior of the DPP GUI. As shown in Figure 50, the average rating on the scale from “unpredictable to predictable” was 5,59, indicating that users generally perceived the interface as consistent and trustworthy. This result suggests that the DPP GUI fosters user confidence by providing stable interactions and expected outcomes, which is essential for maintaining a reliable user experience within the DPP context.

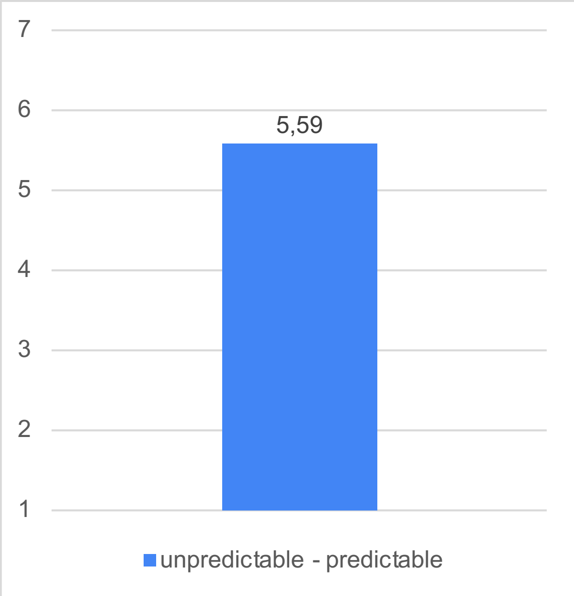


Figure 50: Average user ratings for the dependability dimension of the DPP GUI

4.3.3 Conclusion of the Evaluation Results

The evaluation of the DPP GUI revealed an overall positive perception among users, highlighting its strong usability and favorable user experience. Usability aspects such as ease of use and user confidence were consistently rated highly. This indicates that the interface is designed in a way that users find intuitive and straightforward to operate. The learnability of the GUI was also rated very positively, demonstrating that users were able to familiarize themselves with the system quickly and with minimal effort. Such ease of onboarding not only enhances initial acceptance but also reduces the need for extensive training or support, which is a crucial factor for practical implementation.

From a subjective user experience perspective, the DPP GUI was perceived as stimulating and appealing. Users particularly appreciated the clarity and comprehensibility of the interface, which facilitated efficient navigation and task completion. Functional aspects including efficiency and practicality were also rated favorably. Additionally, the dependability of the system, reflected in its predictability and stability, contributed to building trust and satisfaction among users, ensuring reliable performance during interaction.

However, some areas for improvement were identified. The average score for willingness to use the interface frequently was 3,86, indicating only a moderate level of user engagement. While the GUI is functional and user-friendly, usage frequency could be increased through targeted measures. For example, a brief interactive introduction to the main functions could be offered at the beginning to familiarize users with the interface more quickly. Additionally, a dynamic 3D representation of the product could be displayed while scrolling through the page, providing a more engaging visual presentation. Progressive information presentation, such as stepwise disclosure of complex data through tabs, could prevent users from being overwhelmed and guide them efficiently through detailed product-related information. Such interactive and visually appealing elements could enhance users emotional engagement, improve navigation and increase motivation for regular use.

In summary, the results indicate that the DPP GUI provides a robust foundation for effective and satisfying user interaction. However, further efforts to increase emotional engagement and innovation in the interface design are likely to enhance user appeal and encourage sustained usage.

5 Discussion and Outlook

The theoretical fundamentals highlight the growing regulatory momentum and heightened expectations for sustainable practices driving the transition towards a CE. Key initiatives such as the Kyoto Protocol, the Paris Agreement, the Katowice Outcome, the European Green Deal and the Ecodesign for Sustainable Products Regulation emphasize the critical need for resource efficiency, transparency and product lifecycle extension. Within this framework, the DPP emerges as a vital tool to facilitate sustainability and circularity across the entire product lifecycle.

The DPP requires comprehensive, structured and interoperable data across the entire value chain. A well-designed DPP system architecture is essential, as it represents the underlying structure enabling the integration of technologies. Among the core technologies enabling the DPP are data spaces and blockchain. Data spaces allow for decentralized, trusted data exchange, giving stakeholders full control over their data (Data Intelligence Offensive, n.d.), while blockchain provides a tamper-proof, immutable ledger for transparent verification of product information (Sousa et al., 2024).

In light of this context, the following section presents the answers to the research questions defined in Chapter 1.3.

For research question 1, concerning the data requirements imposed on a DPP for different stakeholders, the analysis reveals that, as emphasized by (Spiß et al., 2024), clearly defined rules regarding event logging, data ownership and data processing are critical to addressing the needs of diverse stakeholders. A unified data model is indispensable to meet the requirements of DPP systems. In particular, semantic interoperability and standardization enable smooth data exchange across organizational boundaries. (Thunyaluck & Valilai, 2024) further highlight that effective data management necessitates the structured collection, administration and provision of product-related data throughout the entire product lifecycle. Technical measures such as data validation, quality assurance, version control and audit trails ensure data integrity and traceability. Moreover, they stress the importance of a well-designed data lifecycle management strategy, including binding policies for storage, archiving and deletion of DPP data. This significantly contributes to complying with data protection requirements and reducing risks such as data breaches or unauthorized access. To effectively address the data needs of different stakeholders, a systematic data classification must be performed individually for each user group. This classification allows targeted provision of relevant data. For example, as shown in Table 4, the allocation of data to respective user groups supports demand-oriented data provisioning and ensures that the specific requirements of the various stakeholders are optimally met.

Building upon the understanding of data requirements, research question 2 focuses on solutions for efficient access rights management in data spaces based on different roles in the context of a DPP. Current research presents multiple approaches to implementing role-based access rights management within the context of DPPs. (Canciani et al., 2024) propose private blockchain environments featuring defined access controls that combine the advantages of controlled access and high configurability with the strengths of public blockchains, such as full auditability and immutability. Product-related information is managed in lightweight, portable data structures (ledgers) within the private blockchain. All changes to these ledgers are cryptographically signed and coordinated via a central notary node, which regularly publishes hash-based proofs and integrity checks on a public blockchain. This approach enables secure and verifiable access management tailored to the needs of diverse stakeholders. (Greiner et al., 2024) emphasize that operating blockchain consortia for DPPs involves specific challenges and requirements regarding data management. Within these decentralized ecosystems, governing data flows and defining participant roles are critical to ensuring reliability, transparency and trust. Responsibilities are clearly distributed among consortium members, covering technical, organizational, compliance and supply chain aspects. Governance frameworks define roles for access control, monitoring data quality and validation processes, thereby protecting data sovereignty and preventing misuse. Overall, robust governance structures, reliable technology and shared agreements among participants are essential to enable efficient access rights management. In the developed DPP GUI, access to the required data is facilitated via a data space, where information is stored in a decentralized manner. Access rights for different stakeholders are controlled through a login system, enabling an extended and role-specific presentation of the relevant information.

Finally, turning to research question 3, which explores how a GUI for a (furniture) DPP can be designed to accommodate the different needs of various stakeholders, the developed GUI implements a clear separation of accessible data to address diverse stakeholder requirements while avoiding interface overload. Data is fundamentally divided into public and stakeholder-specific information. Public data, such as basic product information, circularity KPIs and material tracking, is accessible to all users. Stakeholder-specific data, however, is only accessible to authenticated users and may include extended product details or the ability to upload documents such as repair reports. This differentiated access structure ensures that each user only receives information relevant to their role. The role-based concept enhances usability, prevents information overload and simultaneously protects sensitive data within the system.

As a data carrier, a QR-Code attached to the product serves as the access point to the DPP GUI, enabling users to retrieve up-to-date product information. Thanks to the modular structure of the GUI, future work can build upon this foundation to implement additional functionalities. The system’s flexible design allows for seamless integration of new features and modules without the need for major architectural changes. This provides a solid basis for further development and long-term scalability of the DPP solution.

Several opportunities for improvement and further development emerged from the evaluation questionnaire and testing:

- GUI Enhancement: Future iterations of the DPP GUI should focus on optimizing the layout and overall user experience. Based on the evaluation questionnaire, several targeted enhancements were identified. To improve navigation and orientation, a visible breadcrumb navigation should be integrated, clearly indicating the user's current position within the page hierarchy. In terms of layout optimization, the implementation of collapsible sections or interactive tables would help manage long lists more efficiently, reducing visual clutter and streamlining the interface. To further support users in understanding complex content, tooltips or brief explanations for technical terminology should be added. Additionally, replacing generic PDF links with descriptive, clearly labelled buttons would improve intuitiveness and interaction quality. The visual presentation of data can also be enhanced by displaying the circularity scorecard using progress indicators such as circular charts, which would improve visual clarity and make the data more engaging and accessible to users.
- Access and Authentication: The current login mechanism relies on manually issued credentials. A next step would be the integration of a secure, self-service password reset system with role-based access control, managed via a relational database (e.g., SQL).
- Governance and Responsibilities: Currently, full responsibility for DPP management lies with the REO. While sufficient for the prototype, a more detailed and distributed responsibility model will be required in future implementations to ensure scalability, compliance and accountability.

6 Appendix

6.1 Backend – Code

```
from flask import Flask, render_template, request, jsonify, redirect, url_for, flash
from flask_login import LoginManager, UserMixin, login_user, login_required, logout_user, current_user
from fpdf import FPDF
from werkzeug.utils import secure_filename
import json
import os
import requests
import uuid
import base64
import datetime
from dotenv import load_dotenv
from pymongo.mongo_client import MongoClient
from pymongo.server_api import ServerApi
from bson import ObjectId
import re
from flask import send_file, abort
from io import BytesIO
from ftplib import FTP_TLS

load_dotenv()

ftp_host = os.getenv("FTP_HOST")
ftp_user = os.getenv("FTP_USER")
ftp_passwd = os.getenv("FTP_PASSWD")

try:
    ftp = FTP_TLS()
    ftp.connect(ftp_host, 21, timeout=10)
    ftp.auth()              # Initiate TLS (explicitly)
    ftp.prot_p()            # Initiate TLS (explicitly)
    ftp.login(ftp_user, ftp_passwd)
    ftp.set_pasv(True)      # Enable passive mode
    print("Connected and logged into the webspace!")
    ftp.quit()

except Exception as e:
    print("Error connecting or logging in:", e)

class FTPClient:
```

```
def __init__(self, host, user, passwd, timeout=10):
    self.host = host
    self.user = user
    self.passwd = passwd
    self.timeout = timeout
    self.ftp = None

def __enter__(self):
    self.ftp = FTP_TLS()
    self.ftp.connect(self.host, 21, timeout=self.timeout)
    self.ftp.auth()
    self.ftp.prot_p()
    self.ftp.login(self.user, self.passwd)
    self.ftp.set_pasv(True)
    return self

def __exit__(self, exc_type, exc_val, exc_tb):
    if self.ftp:
        self.ftp.quit()

def download_file_as_base64(self, remote_path):
    buffer = BytesIO()
    self.ftp.retrbinary(f"RETR {remote_path}", buffer.write)
    buffer.seek(0)
    ext = remote_path.split('.')[-1].lower()
    encoded = base64.b64encode(buffer.read()).decode('utf-8')
    return f"data:image/{ext};base64,{encoded}"

def download_pdfs(self, folder_path):
    self.ftp.cwd(folder_path)
    files = self.ftp.nlst()
    pdf_files = {}
    for file in files:
        if file.lower().endswith('.pdf'):
            bio = BytesIO()
            self.ftp.retrbinary(f"RETR {file}", bio.write)
            bio.seek(0)
            pdf_files[file] = bio.read()
    return pdf_files

def list_pdfs(self, folder_path):
    self.ftp.cwd(folder_path)
    files = self.ftp.nlst()
    return sorted([f for f in files if f.lower().endswith('.pdf')])
```

```
    def upload_pdf(self, folder_path, filename, file_bytes_io):
        self.ftp.cwd(folder_path)
        self.ftp.storbinary(f"STOR {filename}", file_bytes_io)

# === Helper Functions ===

def objectid_to_str(obj):
    if isinstance(obj, ObjectId):
        return str(obj)
    raise TypeError(f"Type {type(obj)} not serializable")

class JSONEncoder(json.JSONEncoder):
    def default(self, obj):
        if isinstance(obj, ObjectId):
            return str(obj)
        return super(JSONEncoder, self).default(obj)

# === Flask Configuration ===

app = Flask(__name__)
app.json_encoder = JSONEncoder
app.secret_key = os.getenv("SECRET_KEY")

if not os.path.exists('logs'):
    os.makedirs('logs')
if not os.path.exists('requests'):
    os.makedirs('requests')

print("Current working directory:", os.getcwd())

# Example User

class User(UserMixin):
    def __init__(self, id, role):
        self.id = id
        self.role = role

    def has_role(self, role_name):
        return self.role == role_name

# User Data

users = {
```

```
    "repair_service": {"password": os.getenv("REPAIR_PASSWORD"), "role": "repair_service"},
    "recycler": {"password": os.getenv("RECYCLER_PASSWORD"), "role": "recycler"},
    "authority": {"password": os.getenv("AUTHORITY_PASSWORD"), "role": "authority"},
}

login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = "login"

@login_manager.user_loader
def load_user(user_id):
    for username, user_info in users.items():
        if username == user_id:
            return User(id=username, role=user_info["role"])
    return None

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        user = users.get(username)

        if user and user['password'] == password:
            login_user(User(id=username, role=user['role']))
            flash('Successfully logged in!', 'success')
            return redirect(url_for('show_forest_data'))
        else:
            flash('Login failed. Please check your credentials.', 'danger')

    return render_template('login.html')

@app.route('/logout')
@login_required
def logout():
    logout_user()
    flash('Successfully logged out.', 'success')
    return redirect(url_for('show_forest_data'))

# === Nexyo API Configuration ===

url = os.getenv("NEXYO_URL")
api_key = os.getenv("NEXYO_API_KEY")
```

```
headers = {"X-Api-Key": api_key, "Content-Type": "application/json"}
connection_uuid = os.getenv("CONNECTION_UUID")
joinery_uuid = os.getenv("JOINERY_UUID")

# POST request to Forest

def send_transfer():
  try:
    response = requests.post(
      url,
      json={"connectionUUID": connection_uuid, "dataRequestParameters": {}},
      headers=headers
    )
    print("API Status Code:", response.status_code)
    print("API Response:", response.text)

    if response.status_code == 200:
      return response.json().get("data", [])
    else:
      print("Error with the API request:", response.status_code)
      return None
  except Exception as e:
    print(f"API connection error: {e}")
    return None

def send_transfer_joinery():
  try:
    response = requests.post(
      url,
      json={"connectionUUID": joinery_uuid, "dataRequestParameters": {}},
      headers=headers
    )
    print("Joinery API Status Code:", response.status_code)
    print("Joinery Response:", response.text)

    if response.status_code == 200:
      return response.json().get("data", [])
    else:
      print("Error with the Joinery API:", response.status_code)
      return None
  except Exception as e:
    print(f"Joinery API connection error: {e}")
    return None
```

```
# === MongoDB Connection ===

uri = os.getenv("MONGODB_URI")
client = MongoClient(uri, server_api=ServerApi('1'))
db = client["nexyodata"]
collection = db["Forest_data_dynamic"]
collection_joinery = db["Joinery_Product_c"]

try:
    client.admin.command('ping')
    print("Successfully connected to MongoDB!")
except Exception as e:
    print(f"MongoDB connection error: {e}")

@app.route('/receive-from-nexyo', methods=['POST'])
def receive_from_nexyo():
    try:
        timestamp = datetime.datetime.now().strftime('%Y%m%d_%H%M%S')
        log_entry = f"""
        Timestamp: {timestamp}
        Request Headers: {dict(request.headers)}
        Content-Type: {request.content_type}
        Remote Address: {request.remote_addr}
        """
        with open(f'logs/request_{timestamp}.log', 'w', encoding='utf-8') as log_file:
            log_file.write(log_entry)

        if not request.content_type:
            return jsonify({"error": "No Content-Type header sent"}), 400

        if request.content_type == 'application/json':
            data = request.get_json()

            if not data:
                return jsonify({"error": "No JSON data received"}), 400

            # Log data

            json_filename = f'requests/request_{timestamp}.json'
            with open(json_filename, 'w', encoding='utf-8') as json_file:
                json.dump(data, json_file, indent=2, ensure_ascii=False)
            app.logger.info(f"JSON data saved to file: {json_filename}")

            # Decide which collection based on "Label_of_boards"
```

```
    if isinstance(data, dict):
        if data.get("Label_of_boards"):
            target_collection = collection_joinery
            print("Joinery data detected based on 'Label_of_boards'.")
        else:
            target_collection = collection
            print("Forest data detected (no 'Label_of_boards').")
    elif isinstance(data, list):
        # If all objects have the field "Label_of_boards" → Joinery
        if all(isinstance(item, dict) and item.get("Label_of_boards") for item in data):
            target_collection = collection_joinery
            print("Joinery data (list) detected based on 'Label_of_boards'.")
        else:
            target_collection = collection
            print("Forest data (list) detected, 'Label_of_boards' missing in at least one object.")
    else:
        return jsonify({"error": "Unexpected format. Expected dict or list of dicts."}), 400

    # Save to MongoDB

    inserted_ids = []

    if isinstance(data, dict):
        result = target_collection.update_one(
            {"id": data["id"]},
            {"$set": data, "$currentDate": {"updated_at": True}},
            upsert=True
        )
        inserted_ids = [str(result.upserted_id)] if result.upserted_id else []
    elif isinstance(data, list) and all(isinstance(item, dict) for item in data):
        for item in data:
            result = target_collection.update_one(
                {"id": item["id"]},
                {"$set": item, "$currentDate": {"updated_at": True}},
                upsert=True
            )
            if result.upserted_id:
                inserted_ids.append(str(result.upserted_id))
    else:
        return jsonify({"error": "Unexpected format. Expected dict or list of dicts."}), 400
```

```
            app.logger.info(f"Data saved in collection: {target_collection.name}, IDs: {inserted_ids}")

            return jsonify({
                "message": "Data successfully received and saved.",
                "filename": json_filename,
                "inserted_ids": inserted_ids,
                "data": data
            }), 200

        else:
            return jsonify({"error": "Unknown Content-Type"}), 400

    except Exception as e:
        error_message = f"Error processing the request: {str(e)}"
        app.logger.error(error_message)
        with open(f'logs/error_{timestamp}.log', 'w', encoding='utf-8') as error_file:
            error_file.write(f"{error_message}\n")
        return jsonify({"error": error_message}), 500

def load_image_base64():
    remote_path = "/Tekin - Files/static/Chair.png"
    with FTPClient(ftp_host, ftp_user, ftp_passwd) as client:
        return client.download_file_as_base64(remote_path)

def load_image_base64_logo():
    remote_path = "/Tekin - Files/static/Mölltal_Möbel.png"
    with FTPClient(ftp_host, ftp_user, ftp_passwd) as client:
        return client.download_file_as_base64(remote_path)

def load_image_base64_logo2():
    remote_path = "/Tekin - Files/static/Champions.png"
    with FTPClient(ftp_host, ftp_user, ftp_passwd) as client:
        return client.download_file_as_base64(remote_path)

def download_pdfs_from_ftps_folder(folder_path):
    try:
        with FTPClient(ftp_host, ftp_user, ftp_passwd) as client:
            return client.download_pdfs(folder_path)
    except Exception as e:
        print("FTP error:", e)
        return {}

def list_ftp_pdfs(ftp_host, ftp_user, ftp_passwd, ftp_dir):
```

```
    try:
        with FTPClient(ftp_host, ftp_user, ftp_passwd) as client:
            return client.list_pdfs(ftp_dir)
    except Exception as e:
        print(f"FTP error while listing files: {e}")
        return []

def upload_pdf_to_ftp(ftp_host, ftp_user, ftp_passwd, ftp_dir, filename, file_bytes_io):
    try:
        with FTPClient(ftp_host, ftp_user, ftp_passwd) as client:
            client.upload_pdf(ftp_dir, filename, file_bytes_io)
            return True
    except Exception as e:
        print(f"FTP upload error: {e}")
        return False

def extract_labels(label_of_boards):
    # Extract labels like F1L966B8-F1L1032B1 here and return only F1L966 and F1L1032
    matches = re.findall(r'F1L\d{3,4}', label_of_boards)
    return matches

@app.route('/show-data', methods=['GET'])
def show_forest_data():
    try:

        # Send transfer to Nexyo

        send_transfer()
        send_transfer_joinery()
        print("Transfer sent to Nexyo.")

        #  Load Joinery document with id: 2

        joinery_document = collection_joinery.find_one({"id": 2})
        if not joinery_document:
            raise Exception("No Joinery document found with id 2.")

        # Extract labels from the Joinery document

        label_of_boards = joinery_document.get("Label_of_boards")
        if not label_of_boards:
            raise Exception("No Label_of_boards found in the Joinery document.")

        # Extracting labels (e.g., F1L966B8-F1L1032B1 -> F1L966, F1L1032)
```

```
label_candidates = extract_labels(label_of_boards)
if not label_candidates:
   raise Exception("No valid labels found in Label_of_boards.")

# If multiple labels are present, search for all of them

regex_queries = [{"label": {"$regex": f"^{label}$"}} for label in label_candidates]
forest_documents_cursor = collection.find({"$or": regex_queries})
forest_documents = list(forest_documents_cursor)

if not forest_documents:
   raise Exception("No matching Forest documents found for the Joinery document.")

# Convert ObjectIds

for doc in forest_documents:
   doc["_id"] = str(doc["_id"])
joinery_document["_id"] = str(joinery_document["_id"])

# Retrieve PDF files from the static/repairs folder

repair_files = list_ftp_pdfs(
ftp_host,
ftp_user,
ftp_passwd,
'/Tekin - Files/static/repairs'
)

# Retrieve PDF files from the documents folder

document_files = list_ftp_pdfs(
ftp_host,
ftp_user,
ftp_passwd,
'/Tekin - Files/documents'
)

image_data = load_image_base64()
image_data_logo = load_image_base64_logo()
image_data_logo2 = load_image_base64_logo2()

# Render template

```

```
        return render_template(
            'display_data.html',
            forest_info=forest_documents or [],
            joinery_info=[joinery_document] if joinery_document else [],
            repair_files=repair_files,
            document_files=document_files,
            image_data=image_data,
            image_data_logo=image_data_logo,
            image_data_logo2=image_data_logo2,
            error=None
        )

    except Exception as e:
        return render_template(
            'display_data.html',
            forest_info=[],
            joinery_info=[],
            repair_files=[],
            document_files=[],
            image_data=[],
            image_data_logo=[],
            image_data_logo2=[],
            error=f"Fehler: {str(e)}"
        )

@app.route('/repair_report', methods=['POST'])
def repair_report():
    if request.method == 'POST' and current_user.is_authenticated and current_user.has_role('repair_service'):
        tasks = request.form.getlist('tasks')
        tasks_other = request.form.get('tasks_other', '')
        comment = request.form.get('comment', '')

        if tasks_other:
            tasks.append(f"Other: {tasks_other}")

        timestamp = datetime.datetime.now().strftime('%Y%m%d_%H%M%S')
        report_id = uuid.uuid4().hex  # UUID remains fixed for the report content
        pdf_filename = f"repair_{timestamp}.pdf"  # Filename includes only the timestamp

        # Prepare the PDF
        pdf = FPDF()
        pdf.add_page()
        pdf.set_font("Arial", style="BU", size=18)
```

```
    pdf.cell(200, 10, txt="Repair Report", ln=True, align="C")

    # Display UUID in the report
    pdf.set_font("Arial", style="I", size=10)
    pdf.cell(200, 10, txt=f"Report ID: {report_id}", ln=True, align="C")

    pdf.set_font("Arial", style="B", size=12)
    pdf.cell(200, 10, txt=f"Date: {datetime.datetime.now().strftime('%Y-%m-%d %H:%M')}", ln=True)
    pdf.ln(10)

    pdf.set_font("Arial", style="U", size=12)
    pdf.cell(200, 10, txt="Repair tasks", ln=True)

    pdf.set_font("Arial", size=12)
    for task in tasks:
        pdf.cell(200, 10, txt=f"- {task.replace('_', ' ').capitalize()}", ln=True)
        pdf.ln(5)

    pdf.set_font("Arial", style="U", size=12)
    pdf.cell(200, 10, txt="Description of the repair", ln=True)
    pdf.set_font("Arial", size=12)
    pdf.multi_cell(0, 10, txt=comment)

    # Save and insert images if available (multiple images possible)
    images = request.files.getlist('repair_image')  # Multiple files with input name="repair_image"
    image_counter = 1
    if images:
        pdf.set_font("Arial", style="U", size=12)
        pdf.cell(200, 10, txt="Photos of the repair", ln=True)

        for image in images:
            if image and image.filename != "":
                image_filename = secure_filename(f"{timestamp}_{image_counter}_{image.filename}")
                image_path = os.path.join("static", "repairs", "images", image_filename)
                os.makedirs(os.path.dirname(image_path), exist_ok=True)
                image.save(image_path)

                # Check if there is enough space on the page; if not, add a new page
                if pdf.get_y() + 70 > 270:  # Rough check of the bottom margin
                    pdf.add_page()

                pdf.set_font("Arial", size=10)
```

```
                    pdf.cell(200, 10, txt=f"Photo {image_counter}", ln=True)
                    pdf.image(image_path, x=10, y=pdf.get_y(), w=100)
                    pdf.ln(60)  # Add spacing after the image
                    image_counter += 1

        # Save PDF in memory
        pdf_buffer = BytesIO()
        pdf_data = pdf.output(dest='S').encode('latin-1')
        pdf_buffer.write(pdf_data)
        pdf_buffer.seek(0)

        # FTP Upload
        ftp_dir = '/Tekin - Files/static/repairs'
        success = upload_pdf_to_ftp(ftp_host, ftp_user, ftp_passwd, ftp_dir, pdf_filename, pdf_buffer)

        if success:
            flash("Repair report successfully uploaded to FTP.", "success")
        else:
            flash("Failed to upload repair report to FTP.", "danger")

        return redirect(url_for('show_forest_data'))

@app.route('/download_repair/<filename>')
def download_repair(filename):
    # Security check: allow only PDF files
    if not filename.endswith('.pdf'):
        abort(403)

    try:
        ftp = FTP_TLS()
        ftp.connect(ftp_host, 21, timeout=10)
        ftp.auth()
        ftp.prot_p()
        ftp.login(ftp_user, ftp_passwd)
        ftp.set_pasv(True)
        ftp.cwd('/Tekin - Files/static/repairs')

        # Save file in BytesIO
        bio = BytesIO()
        ftp.retrbinary(f"RETR {filename}", bio.write)
        ftp.quit()

        bio.seek(0)
```

```
        return send_file(
            bio,
            mimetype='application/pdf',
            download_name=filename,
            as_attachment=False
        )
    except Exception as e:
        return f"Error loading the file: {str(e)}", 500

@app.route('/dokumente/<filename>')
def dokumente(filename):
    if not filename.endswith('.pdf'):
        abort(403)

    try:
        ftp_dir = '/Tekin - Files/documents'
        ftp = FTP_TLS()
        ftp.connect(ftp_host, 21, timeout=10)
        ftp.auth()
        ftp.prot_p()
        ftp.login(ftp_user, ftp_passwd)
        ftp.set_pasv(True)
        ftp.cwd(ftp_dir)

        bio = BytesIO()
        ftp.retrbinary(f"RETR {filename}", bio.write)
        ftp.quit()

        bio.seek(0)
        return send_file(
            bio,
            mimetype='application/pdf',
            download_name=filename,
            as_attachment=False
        )
    except Exception as e:
        return f"Error loading the file: {str(e)}", 500

# === Starting Point ===

if __name__ == '__main__':
    port = int(os.environ.get("PORT", 10000))  # Fallback to 10000 locally
    app.run(host="0.0.0.0", port=port)
```

6.2 Frontend – Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Chair</title>
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.3/dist/leaflet.css" />
  <style>
    html {
      scroll-behavior: smooth;
    }

    body {
      font-family: Arial, sans-serif;
      margin: 0;
      background-color: #f4f4f4;
    }

    .container {
      background-color: #ffffff;
      padding: 20px;
      transition: margin-left 0.3s ease;
    }

    h1 {
      text-align: center;
    }

    .data-item {
      margin: 10px 0;
    }

    .label {
      font-weight: bold;
    }

    .data-record {
      margin-bottom: 20px;
      padding: 10px;
      border: 1px solid #ddd;
      border-radius: 8px;
    }
```

```

.error {
  color: red;
  text-align: center;
  font-weight: bold;
}

#map {
  height: 500px;
  margin-top: 40px;
  border-radius: 8px;
}

/* Sidebar styles */
.menu-toggle {
  position: fixed;
  top: 20px;
  left: 20px;
  font-size: 28px;
  cursor: pointer;
  z-index: 10001;
  background-color: #333;
  color: white;
  padding: 10px 15px;
  border-radius: 8px;
  transition: opacity 0.3s ease;
}

.sidebar {
  position: fixed;
  top: 0;
  left: -260px;
  width: 240px;
  height: 100%;
  background-color: #333;
  padding: 20px;
  color: white;
  transition: left 0.3s ease;
  z-index: 10001;
}

.sidebar.open {
  left: 0;
}
```

```
.sidebar h2 {
  margin-top: 0;
  font-size: 20px;
  border-bottom: 1px solid #555;
  padding-bottom: 10px;
}

.sidebar a {
  display: block;
  color: white;
  text-decoration: none;
  padding: 10px 0;
}

.sidebar a:hover {
  background-color: #444;
  padding-left: 10px;
}

.overlay {
  display: none;
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0, 0.3);
  z-index: 9999;
}

.overlay.active {
  display: block;
}

@media (min-width: 769px) {
  .container {
    margin-left: 260px;
  }

  .overlay {
    display: none !important;
  }
}
```

```

    @media (max-width: 768px) {
      .container {
        margin-left: 0;
        padding: 15px;
      }

      .sidebar {
        width: 80%;
        left: -100%;
      }

      .sidebar.open {
        left: 0;
      }

      .sidebar a {
        font-size: 16px;
        padding: 15px 0;
      }

      .sidebar h2 {
        font-size: 18px;
      }

      .data-record {
        padding: 12px;
        margin-bottom: 15px;
      }

      #map {
        height: 300px;
        margin-top: 20px;
      }
    }
  </style>
</head>
<body>

  <!-- Hamburger Icon -->
  <div class="menu-toggle" onclick="toggleSidebar()">☰</div>

  <!-- Sidebar Navigation -->
  <div class="sidebar" id="sidebar">
```

```
    <h2>Navigation</h2>
    <a href="#general" onclick="closeSidebar()">📦 General</a>
    <a href="#map" onclick="closeSidebar()">📍 Map</a>
    <a href="#documents" onclick="closeSidebar()">📄 Documents</a>
    <a href="#users" onclick="closeSidebar()">👥 Users</a>
  </div>

  <!-- Overlay for mobile -->
  <div class="overlay" id="overlay" onclick="closeSidebar()"></div>

  <!-- Main Content -->

  <div class="container">

    <!-- General -->
    <div id="general" style="text-align: left;">
      <h1>BRETTSTUHL</h1>

      {% if error %}
        <p class="error">{{ error }}</p>
      {% endif %}

    <div style="display: flex; flex-direction: column; align-items: center; padding: 20px;">
      <div style="position: relative; width: 250px; height: 250px;">
        <img src="{{ image_data }}" alt="Chair" style="width:200px; height:auto; display: block; margin: auto;">

        <a href="https://www.moelltal-moebel.at/" target="_blank">
          <img src="{{ image_data_logo }}" alt="Mölltal Möbel" style="width:90px; height:auto; position: absolute; top: 10px; left: -25px; z-index: 1;">
        </a>

        <a href="https://www.champi40ns.eu/" target="_blank">
          <img src="{{ image_data_logo2 }}" alt="Champions" style="width:90px; height:auto; position: absolute; bottom: 10px; right: -40px; z-index: 1;">
        </a>
      </div>
    </div>

    <div id="product-details" >
      <h1 style="display: none;">Product Data</h1>
      {% if joinery_info %}
        {% for item in joinery_info %}
          <div class="data-record">
```

```
            <h3>Product Details</h3>
            <div class="data-item"><span class="label">Product:</span> {{ item.get('Product') if item.get('Product') else 'no' }}</div>
            <div class="data-item"><span class="label">Width:</span> {{ item.get('Width') if item.get('Width') else 'no' }}</div>
            <div class="data-item"><span class="label">Height:</span> {{ item.get('Height') if item.get('Height') else 'no' }}</div>
            <div class="data-item"><span class="label">Depth:</span> {{ item.get('depth') if item.get('depth') else 'no' }}</div>
            <div class="data-item"><span class="label">Product Weight:</span> {{ item.get('product_weight') if item.get('product_weight') else 'no' }}</div>
            <div class="data-item"><span class="label">Surface Coating:</span> {{ item.get('Surface_Coating') if item.get('Surface_Coating') else 'no' }}</div>
            <div class="data-item"><span class="label">Paint:</span> {{ item.get('Paint') if item.get('Paint') else 'no' }}</div>
            <div class="data-item"><span class="label">Specific Characteristics:</span> {{ item.get('Specific_characteristics') if item.get('Specific_characteristics') else 'no' }}</div>
            <div class="data-item"><span class="label">Certification:</span> {{ item.get('Certification') if item.get('Certification') else 'no' }}</div>
          </div>
        {% endfor %}
      {% else %}
        <p>No product data available.</p>
      {% endif %}
    </div>

    <div id="order-information" >
      <h1 style="display: none;">Order Information</h1>
      {% if joinery_info %}
        {% for item in joinery_info %}
          <div class="data-record">
            <h3>Order Information</h3>
            <div class="data-item"><span class="label">Date of Order:</span> {{ item.get('Date_of_order') if item.get('Date_of_order') else 'no' }}</div>
            <div class="data-item"><span class="label">Date of Delivery:</span> {{ item.get('Date_of_delivery') if item.get('Date_of_delivery') else 'no' }}</div>
          </div>
        {% endfor %}
      {% else %}
        <p>No order data available.</p>
      {% endif %}
    </div>

    <!-- Circularity Scorecard -->
    <div id="circularity-scorecard">
```

```
        <h1 style="display: none;">Circularity Scorecard</h1>

        <!-- Data container with gray border -->
        <div class="data-record">
            <h3>Circularity Scorecard</h3>
            <p>The Circularity Scorecard provides a simple high-level summary of the circularity performance of the product.</p>

            <!-- Score box within the data record -->
            <div style="border: 2px solid #000; padding: 15px; width: 250px; height: 140px; text-align: center; margin-bottom: 20px;">
                <hr>
                <p style="font-size: 40px; font-weight: bold; color: green; margin: 0;">7</p>
                <p style="font-size: 26px; font-weight: bold; color: green; margin: 0;">Repairability Score</p>
                <hr>
                <p style="font-size: 14px; color: green; margin: 0;">0 → lowest score | 10 → highest score</p>
            </div>
        </div>
    </div>

    <div id="forest-data">
        <h1 style="display: none;">Forest Data</h1>
        {% if forest_info %}
            {% for item in forest_info %}
                <div class="data-record">
                    <h3>Tree {{ loop.index }} Data Record</h3>
                    <div class="data-item"><span class="label">Country:</span> {{ item.get('Country') if item.get('Country') else 'no' }}</div>
                    <div class="data-item"><span class="label">Federal State:</span> {{ item.get('Federal_state') if item.get('Federal_state') else 'no' }}</div>
                    <div class="data-item"><span class="label">Municipality:</span> {{ item.get('Municipality') if item.get('Municipality') else 'no' }}</div>
                    <div class="data-item"><span class="label">Wood Species:</span> {{ item.get('Wood_species') if item.get('Wood_species') else 'no' }}</div>
                    <div class="data-item"><span class="label">Quality Class:</span> {{ item.get('Quality_class') if item.get('Quality_class') else 'no' }}</div>
                    <div class="data-item"><span class="label">Certification:</span> {{ item.get('Certification') if item.get('Certification') else 'no' }}</div>
                </div>
            {% endfor %}
        {% else %}
            <p>No Tree data available.</p>
        {% endif %}

```

```
        <!-- Map -->
      {% if forest_info %}
        <h1>GPS coordinates of trees in the forest</h1>
        <div id="map"></div>
      {% else %}
        <p>No forest data available.</p>
      {% endif %}
    </div>

    <h1>Previous Repairs</h1>
    {% if repair_files %}
    <ul>
      {% for file in repair_files %}
      <li>
        <a href="{{ url_for('download_repair', filename=file) }}" target="_blank">{{ file }}</a>
      </li>
      {% endfor %}
    </ul>
    {% else %}
    <p>No repair reports found.</p>
    {% endif %}

    <div id="documents">
      <h1>Documents</h1>
      <ul>
        <li><a href="{{ url_for('dokumente', filename='Pflegeanleitung.pdf') }}" target="_blank">🧽 Maintenance Instruction</a></li>
        <li><a href="{{ url_for('dokumente', filename='Montage- und Demontageanleitung.pdf') }}" target="_blank">🔧 Assembly and Disassembly Instruction</a></li>
        <li><a href="{{ url_for('dokumente', filename='Garantie.pdf') }}" target="_blank">📄 Warranty and Return Information</a></li>
        <li><a href="{{ url_for('dokumente', filename='Umwelthinweise.pdf') }}" target="_blank">♻️ Environmental Instruction</a></li>
      </ul>
    </div>

    <!-- User Information -->

    <div id="users">
      <h1>User Information</h1>
      <p>
        Would you like more information or to add/edit documents?
```

```
                <a href="{{ url_for('login') }}">Please log in as a Repair Service, Recycler or Authority.</a>
            </p>
        </div>

{% if current_user.is_authenticated %}
    <h2>Welcome, {{ current_user.id | capitalize }}!</h2>
    <p>Your role: {{ current_user.role | capitalize }}</p>
    <a href="{{ url_for('logout') }}">Log out</a>

    {% if current_user.role == 'repair_service' %}
        <h2>Repair Service Additional Information</h2>
        <p>Here are the additional data for the repair service:</p>

        {% if joinery_info %}
            <h3>Joinery Information</h3>
            {% for item in joinery_info %}
                <div class="data-record">
                    <h2 style="display: none;">Data Record: {{ item.get('id', '-') }}</h2>
                    <div class="data-item"><span class="label">Label of Boards:</span> {{ item.get('Label_of_boards') if item.get('Label_of_boards') else 'no' }}</div>
                    <div class="data-item"><span class="label">Date of Production:</span> {{ item.get('Date_of_production') if item.get('Date_of_production') else 'no' }}</div>
                    <div class="data-item"><span class="label">Furniture Fittings:</span> {{ item.get('Furniture_fittings') if item.get('Furniture_fittings') else 'no' }}</div>
                    <div class="data-item"><span class="label">Yield from the Boards (%):</span> {{ item.get('Yield_from_the_boards') if item.get('Yield_from_the_boards') else 'no' }}</div>
                    <div class="data-item"><span class="label">% Waste (thermal use):</span> {{ item.get('_Waste_thermal_use') if item.get('_Waste_thermal_use') else 'no' }}</div>
                </div>
            {% endfor %}
        {% else %}
            <p>No Joinery data available.</p>
        {% endif %}

        {% if forest_info %}
            <h3>Forest Information</h3>
            {% for item in forest_info %}
                <div class="data-record">
                    <h2 style="display: none;">Data Record: {{ item.get('id', '-') }}</h2>
                    <div class="data-item"><span class="label">Tree {{ loop.index }}:</span></div>
                    <div class="data-item"><span class="label">Label:</span> {{ item.get('label') if item.get('label') else 'no' }}</div>
```

```
                <div class="data-item"><span class="label">Moon Phase:</span> {{ item.get('Moon_phase') if item.get('Moon_phase') else 'no' }}</div>
            </div>
        {% endfor %}
    {% else %}
        <p>No Forest data available.</p>
    {% endif %}

        <h3>Repair Service Report Form</h3>
        <form method="POST" action="{{ url_for('repair_report') }}" enctype="multipart/form-data">
            <h4>Repair Tasks:</h4>
            <input type="checkbox" name="tasks" value="surface_reworked"> Surface reworked<br>
            <input type="checkbox" name="tasks" value="screws_tightened"> Screws tightened<br>
            <input type="checkbox" name="tasks" value="legs_repaired"> Legs repaired<br>
            <input type="checkbox" name="tasks" value="legs_glued"> Legs glued<br>
            <input type="checkbox" name="tasks" value="legs_replaced"> Legs replaced<br>
            <input type="checkbox" name="tasks" value="connections_stabilized"> Stabilized wobbly connections<br>
            <input type="checkbox" name="tasks" value="frame_repaired"> Frame repaired<br>
            <input type="checkbox" name="tasks" value="frame_replaced"> Frame replaced<br>
            <input type="checkbox" name="tasks" value="seat_reupholstered"> Seat reupholstered<br>
            <input type="checkbox" name="tasks" value="backrest_upholstered"> Backrest upholstered<br>
            <input type="checkbox" name="tasks" value="cover_renewed"> Cover (fabric or leather) renewed<br>
            <input type="checkbox" name="tasks" value="cracks_or_holes_repaired"> Cracks or holes repaired<br>
            <input type="checkbox" name="tasks" value="wood_lacquered_or_oiled"> Wood lacquered or oiled<br>
            <input type="checkbox" name="tasks" value="leather_cared_for"> Leather cleaned and maintained<br>
            <input type="checkbox" name="tasks" value="textiles_cleaned"> Textiles cleaned<br>
            <input type="checkbox" name="tasks" value="textiles_impregnated"> Textiles impregnated<br>
            <input type="checkbox" name="tasks" value="other"> Other (please specify):<br>
            <input type="text" name="tasks_other" placeholder="Description of the repair"><br><br>
```

```

            <h4>Additional Remarks:</h4>
            <textarea name="comment" rows="4" cols="50" placeholder="Additional Notes..."></textarea><br><br>

            <h4>Repair Image:</h4>
            <input type="file" name="repair_image" accept="image/*" multiple><br><br>

            <button type="submit">Save Report</button>
          </form>

    {% elif current_user.role == 'recycler' %}
      <h2>Recycler Additional Information</h2>
      <p>Here are the additional data for the recycler:</p>

      {% if joinery_info %}
        <h3>Joinery Information</h3>
        {% for item in joinery_info %}
          <div class="data-record">
            <h2 style="display: none;">Data Record: {{ item.get('id', '-') }}</h2>
            <div class="data-item"><span class="label">Label of Boards:</span> {{ item.get('Label_of_boards') if item.get('Label_of_boards') else 'no' }}</div>
            <div class="data-item"><span class="label">Date of Production:</span> {{ item.get('Date_of_production') if item.get('Date_of_production') else 'no' }}</div>
            <div class="data-item"><span class="label">Moisture Content:</span> {{ item.get('Moisture_content') if item.get('Moisture_content') else 'no' }}</div>
            <div class="data-item"><span class="label">Yield from the Boards (%):</span> {{ item.get('Yield_from_the_boards') if item.get('Yield_from_the_boards') else 'no' }}</div>
            <div class="data-item"><span class="label">% Waste (thermal use):</span> {{ item.get('_Waste_thermal_use') if item.get('_Waste_thermal_use') else 'no' }}</div>
          </div>
        {% endfor %}
      {% else %}
        <p>No Joinery data available.</p>
      {% endif %}

      {% if forest_info %}
        <h3>Forest Information</h3>
        {% for item in forest_info %}
          <div class="data-record">
            <h2 style="display: none;">Data Record: {{ item.get('id', '-') }}</h2>
            <div class="data-item"><span class="label">Label:</span> {{ item.get('label') if item.get('label') else 'no' }}</div>
            <div class="data-item"><span class="label">Moon Phase:</span> {{ item.get('Moon_phase') if item.get('Moon_phase') else 'no' }}</div>
```

```
                </div>
            {% endfor %}
        {% else %}
            <p>No Forest data available.</p>
        {% endif %}

    {% elif current_user.role == 'authority' %}
        <h2>Authority Additional Information</h2>
        <p>Here are the additional data for the authority:</p>

        {% if joinery_info %}
            <h3>Joinery Information</h3>
            {% for item in joinery_info %}
                <div class="data-record">
                    <h2 style="display: none;">Data Record: {{ item.get('id', '-') }}</h2>
                    <div class="data-item"><span class="label">Label of Boards:</span> {{ item.get('Label_of_boards') if item.get('Label_of_boards') else 'no' }}</div>
                    <div class="data-item"><span class="label">Export:</span> {{ item.get('Export') if item.get('Export') else 'no' }}</div>
                    <div class="data-item"><span class="label">National Trade:</span> {{ item.get('National_trade') if item.get('National_trade') else 'no' }}</div>
                </div>
            {% endfor %}
        {% else %}
            <p>No Joinery data available.</p>
        {% endif %}

        {% if forest_info %}
            <h3>Forest Information</h3>
            {% for item in forest_info %}
                <div class="data-record">
                    <h2 style="display: none;">Data Record: {{ item.get('id', '-') }}</h2>
                    <div class="data-item"><span class="label">Tree {{ loop.index }}:</span> </div>
                    <div class="data-item"><span class="label">Label:</span> {{ item.get('label') if item.get('label') else 'no' }}</div>
                    <div class="data-item"><span class="label">GPS Coordinate (Lat):</span> {{ item.get('GPS_coordinate_lat') if item.get('GPS_coordinate_lat') else 'no' }}</div>
                    <div class="data-item"><span class="label">GPS Coordinate (Long):</span> {{ item.get('GPS_coordinate_long') if item.get('GPS_coordinate_long') else 'no' }}</div>
                </div>
            {% endfor %}
        {% else %}
            <p>No Forest data available.</p>
        {% endif %}
```

```
  {% endif %}
{% else %}
  <p>You are not logged in.</p>
{% endif %}

  <!-- Leaflet JS -->

  <script src="https://unpkg.com/leaflet@1.9.3/dist/leaflet.js"></script>
  <script>
    function toggleSidebar() {
      document.getElementById('sidebar').classList.toggle('open');
    }

    function closeSidebar() {
      document.getElementById('sidebar').classList.remove('open');
    }

    const forestData = {{ forest_info | tojson }};
    if (forestData.length > 0) {
      const map = L.map('map').setView([forestData[0].GPS_coordinate_lat, forestData[0].GPS_coordinate_long], 13);

      L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
        attribution: '© OpenStreetMap contributors'
      }).addTo(map);

      forestData.forEach((item, index) => {
        const lat = parseFloat(item.GPS_coordinate_lat);
        const lng = parseFloat(item.GPS_coordinate_long);
        if (!isNaN(lat) && !isNaN(lng)) {
          // Dynamically set the label based on index (1, 2, 3, etc.)
          const forestLabel = `Forest tree ${index + 1}`;

          L.marker([lat, lng])
            .addTo(map)
            .bindPopup(forestLabel);  // Show Forest tree X in popup (Forest tree 1, 2, 3, etc.)
        }
      });

      const bounds = L.latLngBounds(forestData.map(item => [
        parseFloat(item.GPS_coordinate_lat),
        parseFloat(item.GPS_coordinate_long)
      ]));
```

```
        map.fitBounds(bounds);
      }
    </script>
  </body>
  </html>
```

6.3 Frontend – Code Login

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>Login</title>
</head>
<body>
  <h1>Login</h1>

  {% with messages = get_flashed_messages(with_categories=true) %}
   {% if messages %}
    {% for category, message in messages %}
     <div class="alert alert-{{ category }}">{{ message }}</div>
    {% endfor %}
   {% endif %}
  {% endwith %}

  <form method="POST">
    <label for="username">Username:</label>
    <input type="text" name="username" required><br>

    <label for="password">Password:</label>
    <input type="password" name="password" required><br>

    <button type="submit">Login</button>
  </form>
</body>
</html>
```

6.4 User Evaluation Questionnaire – Digital Product Passport GUI

This questionnaire aims to evaluate the usability and user experience of the developed graphical user interface (GUI) for the Digital Product Passport. The feedback collected will be used to improve the system and is part of the evaluation process in a research context.

1. Which of the following best describes your affiliation?
   - Research institution
   - Company in a value chain
   - IT or service company
   - Other

Section 1: System Usability Scale (SUS)

Please rate the following statements based on your experience using the GUI. Use a scale from **1 (Strongly disagree) to 5 (Strongly agree).**

1. I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think I would need support from a technical person to use this system.
5. I found the functions well integrated.
6. I thought there was too much inconsistency in the system.
7. I imagine most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot before I could get going with the system.

Section 2: User Experience Questionnaire

Please indicate your impression of the interface by selecting a value between each pair of opposite attributes **(1 = very left, 7 = very right)**.

1. boring – exciting
2. not understandable – understandable
3. inefficient – efficient
4. confusing – clear
5. impractical – practical
6. conventional – innovative
7. discouraging – attractive
8. unpredictable – predictable

Optional: Open Feedback

What did you like about the GUI? What could be improved?

7 Bibliography

Bär, M. A., Colombo, A. W., Torres, J. L., Fernandez, E., Rico, M., & Caliusco, M. L. (2023). An Industry 4.0-Compliant Digital Product Passport Approach for Realising Dairy Product Traceability. IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society, 1–9. https://doi.org/10.1109/IECON51785.2023.10312481

C4 Model. (n.d.-a). Retrieved 7 June 2025, from https://c4model.com/introduction

C4 Model. (n.d.-b). Retrieved 7 June 2025, from https://c4model.com/diagrams/system-context

C4 Model. (n.d.-c). Retrieved 7 June 2025, from https://c4model.com/diagrams/container

C4 Model. (n.d.-d). Retrieved 7 June 2025, from https://c4model.com/diagrams/component

Canciani, A., Felicioli, C., Severino, F., & Tortola, D. (2024). Enhancing Supply Chain Transparency through Blockchain Product Passports. 2024 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops), 751–756. https://doi.org/10.1109/PerComWorkshops59983.2024.10502429

Chiacchio, F., Compagno, L., D'Urso, D., Velardita, L., & Sandner, P. (2020). A decentralized application for the traceability process in the pharma industry. Procedia Manufacturing, 42, 362–369. https://doi.org/10.1016/j.promfg.2020.02.063

Circularise. (n.d.-a). Retrieved 17 August 2025, from https://www.circularise.com/blogs/digital-product-passports-dpp-what-how-and-why

Circularise. (n.d.-b). Retrieved 17 August 2025, from https://www.circularise.com/dpp-for-espr

CIRPASS. (2022). The Digital Product Passport as defined in the Proposal for Eco-design for Sustainable Product Regulation (ESPR).

CIRPASS. (2024). D3.3_Identification Schemes-v1.3.pdf.

Data Intelligence Offensive. (n.d.). Retrieved 28 January 2025, from https://dataintelligence.at/dataspaces/

Dayma, N., Badwane, S., Bhole, C., & Mahajan, M. (2023). Fake Product Detection using Blockchain. 2023 6th International Conference on Advances in Science and Technology (ICAST), 223–228. https://doi.org/10.1109/ICAST59062.2023.10454928

Deutsches Institut für Normung. (n.d.). Retrieved 30 August 2025, from https://www.din.de/de/mitwirken/entwuerfe/ne-stellung/108664!search-na?query=digitaler+produktpass&gremselect=0&submit-btn=Submit

European Commission. (n.d.-a). Retrieved 28 January 2025, from https://eur-lex.europa.eu/resource.html?uri=cellar:ccd71fda-b1b5-11ec-9d96-01aa75ed71a1.0001.02/DOC_4&format=PDF

European Commission. (n.d.-b). Retrieved 26 December 2024, from https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal/climate-action-and-green-deal_en

European Commission. (n.d.-c). Retrieved 8 January 2025, from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52019DC0640

European Commission. (n.d.-d). Retrieved 6 January 2025, from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52023DC0062

European Commission. (n.d.-e). Retrieved 10 January 2025, from https://commission.europa.eu/energy-climate-change-environment/standards-tools-and-labels/products-labelling-rules-and-requirements/ecodesign-sustainable-products-regulation_en

European Commission. (2021). EUROPÄISCHER GRÜNER DEAL: DIE VERWIRKLICHUNG UNSERER ZIELE.

European Commission. (2023a). Questions and Answers: Green Deal Industrial Plan for the Net-Zero Age.

European Commission. (2023b). THE GREEN DEAL INDUSTRIAL PLAN: Speeding up the contribution of Europe’s innovative clean tech industries to net-zero [Dataset]. https://doi.org/10.1163/9789004322714_cclc_2023-0189-0914

European Union. (n.d.). Retrieved 13 August 2025, from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1781

Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology. (n.d.). Retrieved 5 January 2025, from https://www.bmk.gv.at/themen/klima_umwelt/eu_international/euop_greendeal.html

Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology. (2022a). Austria on the path to a sustainable and circular society.

Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology. (2022b). Austria‘s Circular Economy Strategy.

Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology. (2023). DPP4ALL – A Digital Product Passport for All.

Federal Ministry Republic of Germany for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection. (n.d.). Retrieved 22 January 2025, from https://www.bmuv.de/media/leitbild-der-kreislaufwirtschaft

Gegeckienė, L., Venytė, I., Kaunas University of Technology, Faculty of Mechanical Engineering and Design, Kaunas, Lithuania, Karpavičė, J., Aarhus University, Department of Business Development and Technology, Aarhus BSS, Herning, Denmark, Tambo, T., Aarhus University, Department of Business Development and Technology, Aarhus BSS, Herning, Denmark, Vaitasius, K., Kaunas University of Technology, Faculty of Mechanical Engineering and Design, Kaunas, Lithuania, Pauliukaitis, D., & Kaunas University of Technology, Faculty of Mechanical Engineering and Design, Kaunas, Lithuania. (2022). Near field communication (NFC) technology in the packaging industry. Proceedings - The Eleventh International Symposium GRID 2022, 495–501. https://doi.org/10.24867/GRID-2022-p54

German Environment Agency. (n.d.). Retrieved 23 June 2025, from https://www.umweltbundesamt.de/sites/default/files/medien/11850/publikationen/58_2025_te xte.pdf

Github, Inc. (n.d.). Retrieved 7 June 2025, from https://docs.github.com/en/get-started/start-your-journey/about-github-and-git

Gligoric, N., Krco, S., Hakola, L., Vehmas, K., De, S., Moessner, K., Jansson, K., Polenz, I., & Van Kranenburg, R. (2019). SmartTags: IoT Product Passport for Circular Economy Based on Printed Sensors and Unique Item-Level Identifiers. Sensors, 19(3), 586. https://doi.org/10.3390/s19030586

Greiner, M., Seidenfad, K., Langewisch, C., Hofmann, A., & Lechner, U. (2024). The Digital Product Passport: Enabling Interoperable Information Flows Through Blockchain Consortia for Sustainability. In F. Phillipson, G. Eichler, C. Erfurth, & G. Fahrnberger (Eds.), Innovations for Community Services (Vol. 2109, pp. 377–396). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-60433-1_21

Halmschlager, V. (2023). Der Digitale Produktpass—Kreislauforientierte Produktion.

Hosttech. (n.d.). Retrieved 7 June 2025, from https://www.hosttech.at/blog/hosting/webspace/

Iakovidis, I., & Ebert, T. (2023). Digital Product Passport – Perspective of the European Commission.

Ishiyama, R., Kudo, Y., & Takahashi, T. (2016). mIDoT: Micro identifier dot on things — A tiny, efficient alternative to barcodes, tags, or marking for industrial parts traceability. 2016 IEEE International Conference on Industrial Technology (ICIT), 781–786. https://doi.org/10.1109/ICIT.2016.7474850

Keepelectronics. (n.d.). Retrieved 4 July 2025, from https://keepelectronics.com/#/product/PF0H268N

Kinsta. (n.d.). Retrieved 7 June 2025, from https://kinsta.com/de/wissensdatenbank/was-ist-ftp/#was-ist-der-unterschied-zwischen-ftp-und-http

Kunz, L., Reif, M., Petzsche, T., Schmallenbach, A., Plociennik, C., & Ruskowski, M. (2024). Realizing Closed-Loop Supply Chain Networks based on Dataspaces and Manufacturing Marketplaces. IFAC-PapersOnLine, 58(19), 235–240. https://doi.org/10.1016/j.ifacol.2024.09.177

MECALUX GmbH. (n.d.). Retrieved 30 November 2024, from https://www.mecalux.de/handbuch-lagerlogistik/lager/rfid

Microsoft. (n.d.). Retrieved 9 June 2025, from https://visualstudio.microsoft.com/de/

Mindtwo GmbH. (n.d.). Retrieved 9 June 2025, from https://www.mindtwo.de/lexicon/rest-representational-state-transfer

Miron, R., & Hulea, M. (2024). Digital Product Passport Implementation Based on Hyperledger Fabric Technology. In J. Machado, F. Soares, J. Trojanowska, V. Ivanov, K. Antosz, D. Cagáňová, V. K. Manupati, & A. Pereira (Eds.), Innovations in Industrial Engineering III (pp. 51–62). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-61582-5_5

MongoDB, Inc. (n.d.). Retrieved 7 June 2025, from https://www.mongodb.com/de-

de/resources/basics/databases/nosql-explained

Narayanan, G., Cvitić, I., Peraković, D., & Raja, S. P. (2024). Role of Blockchain Technology in Supplychain Management. IEEE Access, 12, 19021–19034. https://doi.org/10.1109/ACCESS.2024.3361310

Nexyo. (n.d.). Retrieved 7 June 2025, from https://nexyo.io/technologie

Nowacki, S., Sisik, G. M., & Angelopoulos, C. M. (2023). Digital Product Passports: Use Cases Framework and Technical Architecture Using DLT and Smart Contracts. 2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), 373–380. https://doi.org/10.1109/DCOSS-IoT58021.2023.00067

Osterwalder, M. (2024). Product Information 4.0 – Front-end Demonstrator.

Pallets Projects. (n.d.-a). Retrieved 9 June 2025, from https://flask.palletsprojects.com/en/stable/

Pallets Projects. (n.d.-b). Retrieved 23 June 2025, from https://flask.palletsprojects.com/en/stable/templating/#jinja-setup

Palm, D., Kuenster, N., Dorka, F., & Buecheler, M. (2024). Architectures And Systems For Identifying Assets In Circular Supply Chains Using Digital Product Passports And The Asset Administration Shell. https://doi.org/10.15488/17776

Pohlmeyer, F., Möbitz, C., & Gries, T. (2024). A Sovereign and Interoperable Data Ecosystem for an Eco-Efficient Nonwovens Industry. Sustainability, 16(23), 10735. https://doi.org/10.3390/su162310735

Python Software Foundation. (n.d.). Retrieved 9 June 2025, from https://www.python.org/doc/essays/blurb/

QR TIGER LTD. (n.d.). Retrieved 4 July 2025, from https://www.qrcode-tiger.com/de/what-is-the-difference-between-static-and-dynamic-qr-code#toc_static_qr_code

Render. (n.d.). Retrieved 7 June 2025, from https://render.com/docs

Sai, Y., Chu, C., Trinchi, A., Sola, A., Shen, S., & Chen, S. (2022). UIT - A Universal Identifier of Things to Bridge Cyber and Physical Worlds. 2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 1–3. https://doi.org/10.1109/ICBC54727.2022.9805494

Siska, V. (2023). Building a Sustainable Battery Supply Chain with Digital Battery Passports. https://doi.org/10.35011/IDIMT-2023-347

Sousa, C., Ferreira, R., Pinto, P., Pereira, C., & Rebelo, R. (2024). Digital Product Passport Architecture for Boosting Circularity in Footwear Industry. Procedia Computer Science, 239, 1560–1567. https://doi.org/10.1016/j.procs.2024.06.331

Spiß, M., Berninger, S., Perau, M., Janßen, J., Boos, W., & Schröer, T. (2024). Bidirectional Integration of Digital Product Passports into Information Systems of Production Planning and Control. In L. M. Camarinha-Matos, A. Ortiz, X. Boucher, & A.-M. Barthe-Delanoë (Eds.), Navigating Unpredictability: Collaborative Networks in Non-linear Worlds (Vol. 727, pp. 179–

193). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-71743-7_12

Thunyaluck, M., & Valilai, O. F. (2024). Integrating Digital Product Passports in Multi-Level Supply Chain for enabling Horizontal and Vertical Integration in the Circular Economy. 2024 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 36–42. https://doi.org/10.1109/IEEM62345.2024.10857137

United Nations Climate Change. (n.d.-a). Retrieved 26 December 2024, from https://unfccc.int/kyoto_protocol

United Nations Climate Change. (n.d.-b). Retrieved 28 December 2024, from https://unfccc.int/process-and-meetings/the-paris-agreement

United Nations Climate Change. (n.d.-c). Retrieved 31 December 2024, from https://unfccc.int/process-and-meetings/the-paris-agreement/katowice-climate-package#Limiting-and-reducing-greenhouse-gas-emissions

Verein Industrie 4.0 Österreich. (2024). Digitaler Produktpass—Fact Sheet- Basics.

Walden, J., Steinbrecher, A., & Marinkovic, M. (2021). Digital Product Passports as Enabler of the Circular Economy. Chemie Ingenieur Technik, 93(11), 1717–1727. https://doi.org/10.1002/cite.202100121

Weber, S., Forte, S., Savarino, P., Dickopf, T., & Apostolov, C. (2024). Asset Administration Shell Driven (Product) Carbon Footprint Exchange with Data Management Systems. 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA), 1–6. https://doi.org/10.1109/ETFA61755.2024.10710683

Wenning, R., Papadakos, P., & Bernier, C. (2024). DPP System Architecture. https://doi.org/10.5281/ZENODO.12206138

Yang, Y., Liu, P., Lv, Z., Yao, Y., & Zhou, M. (2024). Development of an Anti-counterfeiting and Traceability System for Goods Based on RFID Tags. In H. Song, M. Xu, L. Yang, L. Zhang, & S. Yan (Eds.), Innovative Technologies for Printing, Packaging and Digital Media (Vol. 1144, pp. 198–204). Springer Nature Singapore. https://doi.org/10.1007/978-981-99-9955-2_25

8 List of illustrations

9 List of tables

10 List of abbreviations

AAS	Asset Administration Shell
AASX	Asset Administration Shell Extended
API	Application Programming Interface
CE	Circular Economy
CSS	Cascading Style Sheets
DID	Decentralized Identifier
DID-MB	Decentralized Identifier-Management Blockchain
DLT	Distributed Ledger Technology
DPP	Digital Product Passport
DPP-DB	Digital Product Passport-Data Blockchain
EDC	Eclipse Dataspace Component
EPCIS	Electronic Product Code Information Services
ESPR	Ecodesign for Sustainable Products Regulations
EU	European Union
FTP	File Transfer Protocol
FTPS	File Transfer Protocol over Transport Layer Security
GHG	Greenhouse Gas Emission
GPS	Global Positioning System
GTIN	Global Trade Identification Number
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ICD	Internet Connected Device
ID	Identifier
IDE	Integrated Development Environment
IDS	International Data Spaces
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
mIDoT	micro Identifier Dot on Things
NFC	Near Field Communication
NFT	Non-Fungible Token
NoSQL	Not only Structured Query Language
ORB	Oriented Fast and Rotated Brief
PDF	Portable Document Format
PDP	Policy Decision Point

PLC	Product Lifecycle
PPC	Production Planning and Control
QR	Quick Response
RANSAC	Random Sample Consensus
RBAC	Role-Based Access Control
RDF	Resource Description Framework
REO	Responsible Economic Operator
REST	Representational State Transfer
RFID	Radio Frequency Identification
RTDI	Research, Technology Development and Innovation
SHACL	Shapes Constraint Language
SQL	Structured Query Language
SUS	System Usability Scale
TARIC	Integrated Tariff of the European Communities
TLS	Transport Layer Security
UEQ	User Experience Questionnaire
UID	Unique Identifier
UPI	Unique Product Identifier
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universally Unique Identifier
VC	Verifiable Credentials
VDR	Verifiable Data Registry
VS	Visual Studio
W3C	World Wide Web Consortium
WSGI	Web Server Gateway Interface
XML	Extensible Markup Language