CIRPASS
Digital Product Passport

D3.2
DPP System Architecture

(Version 1.9)

May 2024

Funded by
the European Union

Lead Beneficiary	ERCIM/W3C
Author(s)/Organisation(s)	Rigo Wenning/ERCIM, Panagiotis Papadakos/ERCIM, Carolynn Bernier/CEA
Contact Email	rigo@w3.org , panagiotis.papadakos@ercim.eu, carolynn.bernier@cea.fr
Contributor(s)	Phil Archer, Staffan Olsson, Adrien Jousse
Work Package	WP3
Due Date:	2023-12-31
Actual Delivery Date	2024-02-08
Abstract:	This document describes the CIRPASS information system architecture centred around the product identifier. It takes into account feedback from WP2, the information architecture, the user stories, the standards landscape, and from the requirements document coming out of task 3.2.
Citation:	Wenning, R., Papadakos, P., & Bernier, C. (2024). DPP System Architecture (V1.9). CIRPASS Consortium. https://doi.org/10.5281/zenodo.12206138

Document Revision History			
Date	**Version**	**Author/Contributor/Reviewer**	**Summary of Main Changes**
02/12/2023	1.2	Rigo Wenning	Initial draft
19/01/2024	1.3	Panagiotis Papadakos	Data flows & diagrams
11/02/2024	1.3	Rigo Wenning	Addressing comments
11/02/2024	1.3.1	Carolynn Bernier	Re-write of Section 2
13/02/2024	1.4	Carolynn Bernier	Section numbering fixed
22/02/2024	1.5	Panagiotis Papadakos & Rigo Wenning, Adrien Jousse	Addressing comments Validation of DPP Architecture
28/02/2024	1.6	Panagiotis Papadakos	Formatting
18/03/2024	1.7	Rigo Wenning, Carolynn Bernier	Addressing comments from EC
26/03/2024	1.8	Rigo Wenning, Carolynn Bernier	Additional clarifications
28/05/2024	1.9	Adrien Jousse	Update of Figures 2 and 3.

Dissemination Level and Nature of the Deliverable		
PU	Public	x
SEN	Sensitive, limited under the conditions of the Grant Agreement	
Nature	**R** = Report, **E** = Ethics or, **O** = Other	R

CIRPASS Consortium			
#	**Participant Organisation Name**	**Short Name**	**Country**
1	COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES	CEA	FR
2	SLR ENVIRONMENTAL CONSULTING(IRELAND)LIMITED	SLR Consulting	IE
3	FRAUNHOFER GESELLSCHAFT ZUR FORDERUNG DER ANGEWANDTEN FORSCHUNG EV	Fraunhofer	DE
4	WUPPERTAL INSTITUT FUR KLIMA, UMWELT, ENERGIE GGMBH	WUPPERTALINSTIT	DE
5	STIFTELSEN CHALMERS INDUSTRITEKNIK	CIT	SE
6	VDE VERBAND DER ELEKTROTECHNIK ELEKTRONIK INFORMATIONSTECHNIK EV	VDE	DE
7	GLOBAL TEXTILE SCHEME GMBH	GTS	DE
8	+IMPAKT LUXEMBOURG SARL	+IMPAKT	LU
9	F6S NETWORK IRELAND LIMITED	F6S	IE
10	GEIE ERCIM	ERCIM	FR
11	E CIRCULAR APS	CEI Society ApS	DK
12	GS1 IN EUROPE	GS1 in Europe	BE
13	POLITECNICO DI MILANO	POLIMI	IT
14	CIRCULAR.FASHION UG (HAFTUNGSBESCHRANKT)	circularfashion	DE
15	DIGITALEUROPE AISBL*	DIGITALEUROPE	BE
16	KIC INNOENERGY SE	KIC SE	NL
17	TECHNISCHE UNIVERSITEIT DELFT	TU Delft	NL
18	TALLINNA TEHNIKAÜLIKOOL	TalTech	EE
19	VELTHA IVZW	VELTHA	BE
20	Energy Web Stiftung (Energy Web Foundation)	EWF	CH
21	BUNDESANSTALT FUER MATERIALFORSCHUNG UND -PRUEFUNG	BAM	DE
22	SyncForce BV	SyncForce	NL
23	ASOCIACION DE EMPRESAS TECNOLOGICAS ES INNOVALIA	INNOVALIA	ES
24	Textile Exchange	TextileExchange	US
25	Responsible Business Alliance	RBA	US
26	WORLDLINE FRANCE	WORLDLINE	FR
27	RISE RESEARCH INSTITUTES OF SWEDEN AB	RISE	SE
28	IPOINT-SYSTEMS GMBH	iPoint	DE
29	Global Electronics Council	GEC	US/NL
30	Avery Dennison Atma GmbH	atma.io	AT
31	Global Battery Alliance	GBA	BE

LEGAL NOTICE

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor the granting authority can be held responsible for them.

Preparing the ground for the gradual piloting and deployment of DPPs from 2023 onwards, focusing on developing a roadmap for prototypes in three value chains: electronics, batteries and textiles.

Grant Agreement: 101083432

Theme: DIGITAL-2021-TRUST-01

Start Date of Project: 01 October 2022

Duration: 18 months

© CIRPASS Consortium, 2024

Reproduction is authorised provided the source is acknowledged.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms	
AAS	Asset Administration Shell
CEOP	Circular Economy Operator
DDR	Decentralized DPP Data Repository
DFD	Data Flow Diagrams
DID	Decentralized Identifier
DNS	Domain Name System
DPP	Digital Product Passport
DPP KG	DPP Knowledge Graph, a graph containing all information about a specific product or model
ERP	Enterprise Resource Planning (Software & System)
ESPR	Ecodesign for Sustainable Products Regulation
GLN	Global Location Number
GTIN	Global Trade Identification Number
ICD	Internet Connected Device
IRI	Internationalized Resource Identifiers
PDP	Policy Decision Point
REO	Responsible Economic Operator: Actor responsible for placing the product on the market and for issuing the DPP
URI	Universal Resource Identifier
VC	Verifiable Credentials
VDR	Verifiable Data Registry A VDR is essentially the place where the [DID Document]s are stored.

About CIRPASS

The European Commission has strong interest and ambition in relation to emerging technologies to support the 'twin', green and digital, transitions and specifically in the development of a **Digital Product Passport (DPP).** The DPP is defined by the European Commission as a structured collection of product related data with pre-defined scope and agreed data management and access rights conveyed through a unique identifier, and that is accessible via electronic means through a data carrier. The intended scope of the DPP is information related to sustainability, circularity, value retention for re-use, remanufacturing and recycling.

The aim of CIRPASS is to prepare the ground for a gradual deployment of DPPs, with an initial focus on the electronics, batteries and textile sectors. Spurred by the need to accelerate the transition to a more circular and sustainable economy, combined with new opportunities offered by a burgeoning data market, a large number of European and international initiatives have emerged recently. CIRPASS's methodology consists in uniting representatives from a large number of these early DPP pilots in order to build a balanced, open and transparent community dedicated to the design and roll-out of the upcoming European DPP.

To ensure a neutral and technology agnostic stance, CIRPASS relies heavily on the involvement of leading European Research and Technology organisations, supported by three standardization organisations, an experienced pool of circular economy and sustainability consultancies, several large European industrial associations, digital technologies and web experts, and digital solution providers. The CIRPASS consortium is made up of 31 partners in total.

By bringing together this community of expertise, the project will build consensus and momentum around the DPP concept and contribute to the development of common principles, prototypes and roadmaps to secure the interoperability of DPPs across value chains, sectors and market participants. Enhanced stakeholder dialogue will be achieved through extensive consultations addressing key DPP aspects such as ontologies, technical requirements and standardization needs.

1 Introduction

Digital Products Passports (DPP) and related information system infrastructure are an integral part of the various strategies that have been embedded in recent European legislation:

- The European Parliament and the Presidency of the Council of the European Union have suggested harmonized and cross-sectoral digital passports for products and intermediate products as decisive ingredients for the introduction of a circular economy, sustainable consumer decisions, as well as resource and energy efficiency. The need for such passports is expressed in the **European Green Deal** and the new **Circular Economy Action Plan (CEAP)**.
- Simultaneously, the **European Data Strategy** described the vision of a common European data space, a Single Market for data in which data could be used irrespective of its physical location of storage in the Union in compliance with applicable law, thus allowing all stakeholders to have access to the data relevant to them. It also called for the free and safe flow of data with third countries. To turn that vision into reality, it proposes to establish domain-specific common European data spaces to improve the access to and interoperability of data, and the concrete arrangements in which data sharing and data pooling can happen. Two of these data spaces, the Industrial (Manufacturing) data space and the Green Deal data space, are closely related to the DPP.

The CIRPASS vision of a Digital Product Passport is fully integrated in the aforementioned strategies. In this vision, the DPP is at the crossroads between the coming data economy, industry digitisation, and the quest for a circular economy. Simply put, a physical good is accompanied by digital information that describe its sustainability, repairability, reusability and recyclability properties and so on, but also events like repair for high value goods. In other words, the CIRPASS system is product centric. Everything starts from the simple case that a person or machine reads an identifier on a tangible good.

Realising this concept, however, immediately poses several challenges. How is the data to be provided? Is it primarily for humans or primarily for machines? Is the data primarily for consumers or for business partners? Should the data be public or access-controlled? Should the data be centralised or distributed? If distributed, how can it be discovered? And where in the lifecycle of a product should data collection begin and where should it end? If a used good is refurbished, how much of the data describing the original production process must be retained? How can the physical and digital worlds be linked in a way that is persistent and robust enough to survive for the long term, including the manufacturer ceasing to trade?

This document describes two parallel, yet interoperable architectures for the provision of Digital Product Passports designed to address these questions:

- An architecture based on HTTP URIs
- An architecture based on the use of decentralized identifiers (DIDs)

These architectures are presented from both a structural viewpoint and a data flow viewpoint and are further validated against the requirements of the DPP system. This report aims no less than to provide a crucial brick to build the information system for the circular economy.

1.1 Scope and limitations of this Deliverable

The architecture proposed in this Deliverable depicts a network of services that can interact with each other, which will meet DPP-related regulatory goals (ESPR, and Battery Regulation) and that maximises flexibility and opportunity for commercial businesses. This deliverable however has the following limitations:

Exhaustiveness: The suggested system makes use of web-technology. Given that the implementation remains within the boundaries of a HTTP-based or a DID-based system, the protocols used allow for interoperability despite a high variety of concrete implementation options. In light of this, this document avoids going into deep details of implementation and depicts a platform for information exchange where every node or service can be realized with a variety of implementations. If specific solutions are mentioned in this deliverable, this should not preclude the use of alternative solutions. Often legacy systems can be connected via transforms. As a consequence, examples are provided for convenience and illustration purposes only.

Focus: The provision of a DPP should be seen as an opportunity for the creation of an information system that helps all players share and pool data in a circular economy. But given this insight, it becomes immediately clear, that the present document must be carefully scoped to avoid digressing into a system that tries to explain the digitising of the entire industry. This is why there will be only hints on where data comes from, where data goes to, how existing data can be reused in a DPP, and how DPP data could be re-used in an Industrie 4.0 scenario.

The current architecture is suggested, because it allows for the use of a large variety of available tools and open-source modules. But it is certainly not the only possible architecture, it is the architecture CIRPASS was able to come up to make the system resilient and decentralized while allowing for easy deployment on top of a well understood and mostly standardized level playing field. Not every deviation from this architecture will break it, but some criteria are essential, especially the use of URIs and the use of a graph data model.

1.2 Definitions used in this report

A **Digital Product Passport** (DPP) is a structured collection of mandatory, machine-readable (where appropriate), product-related data with pre-defined scope and agreed data management and access rights extracted from a standardized product dataspace thanks to a unique product identifier and that is accessible via electronic means through a Data Carrier. The intended scope of the DPP is information related to sustainability, circularity, value retention for reuse, remanufacturing, recycling and legal compliance.

DPP system: In general terms, a DPP- IT- System can be understood as a set of networked and interconnected hardware/software or a collection of components and elements using widely implemented Common Technical Specifications. Note that the Standardisation Request - DPP defines 'DPP system' as the "set of IT standards (and/or protocols, since protocols are IT-Standards) required to ensure the interoperability of cross-sectoral digital product passports ('product passports') and compliance with essential requirements defined in Articles 9 and 10 of COM (2022) 142 final or defined in Article 78 of Regulation (EU) 2023/…."

DPP-as-a-Service: DPP data storing, processing and back-up services provided by certified independent third-party product passport service providers.

A **Data Space** is a secure and standardized digital infrastructure that enables trusted data exchange and data-based services among various stakeholders. [IDSA] defines it as a virtual space that provides a standardized framework for data exchange, based on common protocols and formats, as well as secure and trusted data sharing mechanisms.

2 Methodology

2.1 Requirements for the DPP system

2.1.1 Essential requirements from regulations

Essential requirements for the DPP system are defined in Articles 8 to 13 of the draft text of the Ecodesign for Sustainable Products Regulation [ESPR]). The main characteristics of the DPP system are:

- A persistent unique product identifier (Art.9 (1a))
- A machine-readable data carrier (Art.9 (1b) & (1c)) based on standards
- Use of open standards (Art.9 (1d))
- An open interoperable data exchange network without vendor lock-in (Art.9 (1d))
- Technical, semantic and organisational aspects of end-to-end communication and data transfer
- Interoperable and machine-readable data formats (Art.10 (1a))
- Free of charge and easy access, based on defined access rights (Art.10 (b))
- No secondary use without consent (data usage control) (Art.9 (1da))
- Decentralized data storage, meaning information stored by the REOs or a certified independent third-party product passport service providers authorised to act on their behalf (Art.10 (c) & (d))
- Archiving: Availability of a back-up copy through a certified independent third-party DPP service provider (Art. 9 (3a))
- DPP information points may be either static or dynamic (updatable)
- DPP information points may be either public or have restricted access conditions.

2.1.2 Technical requirements from “DPP User Stories”

Based on draft versions of the ESPR and the Standardisation Request - DPP, CIRPASS consortium partners have formulated several “DPP user stories” based on a conceptual DPP system. For a number of selected ESPR use cases, the processes by which DPP data would be exposed, accessed and managed between stakeholders along the circular supply chain are described. The aim, therefore, is to explain how the DPP system is implemented, operated and maintained. Each user story is described in a textual step-by-step manner detailing the interactions between the parties involved, but without going into detail of the internal process for each participant.

2.1.3 Requirements from the EU Data Strategy

The requirements for the DPP system are fully aligned with the goals of the European Data Strategy which aims to make the EU a leader in a data-driven society and create a single market for data to enable innovative processes, products and services. The European Data Strategy also aims to make more data available for use in the economy and society, while keeping those who generate the data in control and ensuring that European rules, in particular privacy and data protection, as well as competition law, are fully respected. The DPP and the DPP system were designed with these objectives in mind, by including, for example, provisions for the protection of personal data, and by clearly defining responsibility for DPP data and access rights to sensitive information.

Even more importantly for the DPP and the DPP system, the European Data Strategy announced the creation of data spaces in several strategic fields. Common European data spaces bring together relevant data infrastructures and governance frameworks to facilitate data pooling and sharing. The aim of data spaces is to improve the **access to and interoperability of data**. The European Strategy for data refers under the European Green Deal data space to an action for establishing a common European data space for smart circular applications to make available the most relevant data for enabling circular value creation along supply chains. The DPP is a key emanation of these efforts. For these reasons, syntactical and semantic interoperability of DPP data is one of the primary design drivers for the DPP system described in this document.

2.1.4 Additional business requirements

In addition to the above requirements, the deployment of the DPP system will be most easily accepted by business communities if the following aspects are considered:

- **Support of legacy**: The DPP system should facilitate the reuse of legacy IT systems and legacy data, vocabularies, dictionaries, ontologies and data models;
- **Flexibility**: The DPP system should be flexible to easily accommodate both regulatory and non-mandatory (business-model-specific) evolving information requirements;
- **Ease of deployment**: The DPP system should be built using mature technological ecosystems to support the mandatory issuing of DPPs starting in 2027;
- **Future-proof:** The DPP system should be based on state-of-the-art technologies with demonstrated capabilities to support emerging technologies to deliver additional services in the future (e.g., blockchains, verifiable credentials, AI, etc.);
- **Low-cost**: The DPP system should allow for the issuing of DPPs at the lowest cost possible.

2.2 Basic design principles

The CIRPASS proposal for the DPP system architecture is based on several principles set out below. Having a decentralized approach is essential and suggested by consideration 32 of the [ESPR]. The product centric vision stems from the requirements in Art. 8 [ESPR]. The other principles explain the design choices in [section 3] of this document.

2.2.1 A decentralised approach

Principle: The DPP system architecture is decentralised.

Justification: The decentralised approach maximises robustness, resilience and security of data provision and maximises opportunities for a diverse commercial market to evolve for DPP provisioning (DPP-as-a-Service). It allows to locate responsibility with the relevant stakeholders, distributing load at the same time.

Discussion: A decentralised architecture means that data is held and managed by the data's creator (or their appointee) and is not aggregated in a single, centralised, location. Art. 10 [ESPR] has opted for this solution when stating that: "*the data included in the product passport shall be stored by the economic operator responsible for its creation or by certified independent third-party product passport service providers authorised to act on their behalf*".

The disadvantage of a decentralized system is that there is no single known and authoritative place that has all DPP information in the single market. While this is true, it does not mean that a web portal and search engine as required by Art. 12a [ESPR] is impossible or even hard to realize. It does not mean that there is no way to know all DPP Information.

At a first glance, it looks like the DPP system could just be some large-scale central database on a grid- or cloud computing infrastructure spanning over many machines and data centres. This would be a distributed approach. Such an infrastructure could technically balance and bear all the load concentrated on a central system, although it would have to be extremely robust and expensive. A decentralised approach is different from a distributed approach. A distributed approach still has a central service but uses distributed computing resources. The difference between a decentralised and a distributed approach mainly lies in the control structure. A distributed approach has a central point of control, which in the case of the DPP could be a central Commission service that controls what is going and what can be retrieved from such a central service. As can be seen from the big search engines on the web, a distributed system can scale up well with the appropriate investment. But it remains a central system with central control. Participants in such a system will have their relation always mediated by the central authority, who can ban them and lock them out. And the central authority remains responsible for all operations like adding additional infrastructure, defining APIs, selection of service providers and monetary compensation of all service providers.

A decentralised system has many participants and no central point of control. It works because participants in that system follow certain rules of interoperability and data management. In a decentralised system, however, disadvantages are outweighed by the very substantial benefit that there is no single point of failure, no monolithic controlling authority to which all data must be surrendered, and no vendor lock-in. Additionally, a centralised system, once established, would be hard to replace with an alternative supplier. Inertia means that the first contract to provide a centralised system is likely to be the only one. As there are many services and actors in a decentralized system, changing just some actor or component is not as disruptive. An interoperable decentralised system also allows for a real ecosystem of information to develop as every participant can freely contract with every other participant in the system. The public authority takes the role of a market authority with watch dog function instead of being the point of central control. This means controls have to be done ex-post, upon complaint or test, and not ex-ante. But it also means that some instances participating in the overall ecosystem could require ex-ante controls. A push notification to market authorities or custom authorities could be required and are implementable in a decentralized system.

A decentralised approach means that each manufacturer (or their appointee) can operate a variety of innovative services for both consumers and business partners, all the time remaining in control of their own data and being responsible for meeting their own legal obligations. The fact that the responsibility for the DPP system component sits with those responsible for the product has a large variety of positive effects that also create a healthy eco-system of responsibilities and duties.

Although the data architecture is decentralised, there *is* a single starting point from which the DPP and other data can be discovered: the product itself. Whoever holds the product in hand will have a very easy access to information about the tangible thing in their hands.

But some uses, like customs, need an information-centric access to the system. They are not interested in the actual product, but only in the information about the product. The web portal as required by Art. 12a [ESPR] is one way to realize such an information centric access, but there can be others. The web portal is just an element in a decentralized system that helps with the discovery of decentralized resources. But it does not contain the content. A search engine would just crawl the DPP system to set up an index. The web portal with its search engine is an application within the decentralized system that is substantially different from a centralized system.

Another option is that the creation of a new DPP is notified to a central point. In Art.12, the [ESPR] has opted for such a solution by tasking the Commission to set up ("*the registry*"). This document calls that component [EU-Registry]. At minima, this central point then could serve as a backup in case information is lost. But it is understood that the distributed data point at the location of the producer or importer creating the DPP remains the authoritative answer and will be first used. This means that a decentralised system can be used to create an imperfect centralised point of information. The imperfect central entity must be good enough for the purposes intended. This way, the security, agility, elastic reaction to load and the resilience of a decentralised architecture is preserved while still catering to regulatory needs. According to Art. 12a [ESPR], the European Commission will have to set up a web portal for DPP information. Given that the European Commission also has the [EU-Registry], it knows about all resolvers and all Product IDs in the market. In this case, the search engine or portal can be near to complete. In the architecture suggested here, the portal does not have to carry the information. Instead, like a real search engine, it has indexes that allow to find information and then point to the information at its source via the [REO Resolver].

2.2.2 The DPP system is Product Centric

Principle: The entire information system is rooted in the Product-ID. Requirements for the Product-ID are laid down in [D3.3].

Justification: In a circular economy, the tangible good is in the centre of interest. This goal is reflected in this architectural principle. Most DPP use cases depart from a tangible good and from the challenge to discover information about *that* tangible good.

Discussion: There are many existing systems that contain product information. There are ERP systems[1], Product Information systems (PIM) and Industrial Documentation systems[2], there are track

[1] e.g. SAP Systems

[2] e.g. CIDOC-CRM, https://cidoc-crm.org

and trace systems in the retail industry and there are large scale industrial information systems like International Material Data System (IMDS[3]) that contain information about products. Typically, they contain information about products and are used to find the right spare parts for replacement. They typically do not contain information on item level other than the number of items currently in stock at a given place. ERP[4] systems typically are used to do planning and resource management, such as stocks. The information is used to design a desired situation and then, means are engaged to make that desired state a reality. All those systems can be used to help create and manage a DPP system, they are sources of information. But they are fundamentally different, because they are information centric. All research starts from the information system. At best, a tangible good is evaluated to find the type that corresponds to the spare part. Most systems existing today are such information centric systems.

The DPP has a different goal, although it may not be excluded that it could also be used to complement ERP goals. Although a DPP system could theoretically also carry track and trace information, this is not the main goal of the present effort. In the DPP scenario, a natural person or a machine detains a tangible good and wants to find information about the thing detained. This is a nice way to scope the information, but also the information requirements around the utility of information for the goal pursued. The Data Carrier is the immutable link between the physical world and its informational twin that is, on top, linked to further useful information. To centre the information system around the [Product UID] seems evident at this high-level view. But once detailed requirements, scenarios and use cases are explored, it is very easy to fall for an information centric view and to disregard the relation to the actual tangible good. At the same time, the product-centricity has limitations. Those limitations are particularly clear for industries that work with bulk retail. When trading wheat, we could theoretically give serial numbers to all the grains, but the nonsensical nature of such endeavour may only serve the amusement of the reader. In this case, helper constructs are needed in the form of labelling, of classes of goods, of information on wrappings and a combination of model, time and lot. In those scenarios, the DPP system is rather information centric. This can be seen as opposed to high value goods that have a history of repairs and even of remanufacturing.

2.2.3 The DPP is (conceptually) a Knowledge Graph

Life has infinite variety. A DPP system will have to cope with the variety of life. Therefore, the information system must be instantly extensible without additional roll-out requirements. The information system should allow society to start with easy, lightweight information requirements while not creating obstacles for a further sophistication after both the needs of the real world and the requirements from regulation have changed. This leads to the conclusion that the DPP should be a knowledge graph [KGBook].

2.2.3.1 The Arguments in favour of knowledge graphs

In a knowledge graph, information is stored in the form of semantic triples. A semantic triple is an assertion made in predicate logic: {Subject, Predicate, Object}. For example, we can make simple assertions such as {Marie, lives in, Paris} or {The sky, has colour, blue}. As any Object can become the

[3] International Material Data System https://www.mdsystem.com/imdsnt/startpage/index.jsp visited 2023-12-06

[4] Enterprise Resource Planning https://en.wikipedia.org/wiki/Enterprise_resource_planning

Subject of a new assertion, e.g., {Paris, is a city of, France}, graphs can grow as new information is added. 'Subjects' and 'Objects' are referred to as "nodes" of the graph, whereas 'Predicates' are referred to as "edges" of the graph.

As knowledge graphs make it easy to attach new nodes to the graph, there is no need to know everything, i.e., define all nodes of the graph and their relations, in advance. The system can develop as it goes forward in very easy ways. To frame this evolution, ontologies and vocabulary standards will determine the boundaries of possible expressions of the graph.

In the CIRPASS proposal for the DPP system architecture, the DPP is a knowledge graph whose root node is the unique product identifier [Product UID]. The knowledge graph receives all relevant data points and thus holds the DPP data. While mandatory DPP information requirements will be defined by regulation, the complete DPP data itself does not have to be defined a priori because using a graph, the data can evolve incrementally. New sources and new fields and data points can be added, and data can be easily combined with other data. This has wide ranging consequences. The use of a graph model makes the DPP System future proof. A change in an ESPR delegated act will only require marginal changes in the system. An integration into data value chains will be easy. The integration of DPP data into a circular economy dataspace will be painless.

Constructing the DPP as a knowledge graph has many advantages. The knowledge graph can put data points in relation to each other. The semantic information allows for much better analytics and a higher level of interoperability. There are a wide range of tools, commercial ones and open source, available to deal with knowledge graphs. If data is already expressed using a graph representation, this means that this available tooling can be used without bigger transformations. If data is not already expressed using a graph representation, e.g., it is contained in a relational database, transforms will be necessary to expose DPP data. As relational databases are the most common form of data storage representation, such transforms are already widely available. Indeed, the current technology was made for data integration and serves that purpose well. This means that these transforms can be easily implemented on top of the existing IT landscape of a given enterprise.

A very specific knowledge graph is a [Named Graph]. A Named Graph is a graph which is assigned a name in the form of a URI. Because the [Product UID] is transformed into a URI and that URI determines the data elements that belong to this Product, the DPP is conceptually a Named Graph. This has a variety of consequences. Named Graphs were invented in 2005 to respond to requirements of security, trust and provenance. As all the elements of a Named Graph are known, the Graph can be easily signed cryptographically. It can be the object of assertions on provenance and trust, of track and trace information and other metadata. It turns the DPP into an object that can be part of another knowledge graph, and that can be annotated further. This is a very powerful feature which anticipates on the increasing use of linked data in international transparency schemes or by national authorities' credential issuing schemes.

2.2.3.2 Linked Data

When each 'Subject', 'Predicate' and 'Object' of each assertion is defined by an URI, the knowledge graph is expressed in the form of Linked Data.

"Linked Data refers to a method of publishing structured data, so that it can be interlinked and become more useful through semantic queries, founded on HTTP, RDF and URIs" [Bizer2009]. The main ideas of Semantic Web are to represent data in RDF format, using ontologies that enable the creation of

inference rules, for achieving the emerging need for semantic data integration [Mountantonakis2019]. The major principles of Linked Data, for achieving the target of the "Web of Data" (or Semantic Web), were officially proposed in July 2006 by Tim Berners-Lee: "(1) use URIs as names for things, (2) use HTTP URIs so that people can look up those names, (3) when someone looks up a URI, provide useful information, using the standards (RDF, SPARQL), and (4) include links to other URIs, so that they can discover more things." Below, we provide some key definitions about the relevant technologies and standards.

- **URI (Uniform Resource Identifier)** provides a compact sequence of characters that identifies an abstract or physical resource. URI resolving is a common operation performed on URIs. It involves determining the proper data access method and parameters needed to locate and retrieve the resource that the URI represents.
- **RDF (Resource Description Framework)** is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas (database structures) differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. RDF extends the linking structure of the Web to use URIs to name the relationships between things as well as the two ends of the link (this is usually referred to as a "triple"). Indeed, it allows expressing content in form of triplets (subject, property, object), and sets of triples actually form a semantic network /graph. RDF provides a variety of syntax notations and data serialization formats, including RDF/XML, Turtle, N-Triples, JSON-LD, and others.
- **RDFS (RDF Schema)** is used for describing specific kinds of resources and will use specific properties in describing those resources. The basic RDFS concepts (i.e., classes and properties) are provided in the form of an RDFS vocabulary.
- **ODRL (Open Digital Rights Language)** is a standard policy expression language using semantic web technologies for representing permitted and prohibited actions over a certain asset, along with the obligations that are required by parties for exchanging this asset.
- **OWL (Web Ontology Language)** is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things.
- **SHACL (Shapes Constraint Language)** is a language for validating RDF graphs against a set of conditions. These conditions are provided as shapes and other constructs expressed in the form of an RDF graph.
- **SPARQL** is a semantic query language based on graph patterns that can retrieve and manipulate data stored in RDF format across diverse data sources. The queried data can be stored natively in RDF or can be viewed as RDF through mappings.

2.2.3.3 Ontologies and Metadata

An ***ontology*** is a formal description of a conceptualization, which is described as a set of concepts within a domain and the relationships that are held between them. Concerning some formal definitions of an ontology, Nicola Guarino has stated that "*an ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models*" [Guarino1998]. Moreover,

Martin Doerr, has mentioned that "*a formal ontology is a specification of kinds of things and their relations in terms of logic, approximating reality. It is formalised knowledge*"[5].

The DPP KG is data organized as a graph. Vocabularies describe what the semantics in that graph mean. When we talk about "weight" in the DPP, we need a formal definition of what we mean by weight and which scale or measure is used, pounds or kilogramme. Once the vocabularies are defined, ontologies allow to describe the relation between certain objects in the graph or between objects of distinct graphs. This can be very handy when merging data from several sources. But it is not limited to merging and fuels reasoners that can automatically draw conclusions from semantics and relations as encoded in vocabularies and ontologies.

The graph can contain data or, via its use of URIs, point to data on the network. But once we can point to things, we can also describe the things we point to. This is called the "*about principle*". In this case, the term metadata is used to refer to any secondary piece of information that is separate in some way from the primary data. A **metadata schema** can be used for describing metadata information about resources in general. Such resources also include ontologies, where a metadata schema can be used as a bridge between the publishers and the users of the data. There are already many common technical specifications defining metadata schemas, including general resources metadata schemas such as Dublin Core (ISO 15836) and Simple Knowledge Organization Schema (SKOS). They contain a set of "core" elements (properties) for describing resources. They can talk about the origin and quality of data via provenance-based metadata schemas such as PROV-O, VoID, and many others.

To frame the evolution of the DPP knowledge graph, standardised ontologies will determine the boundaries of possible expressions of the graph. First, it is likely that a minimal upper-level cross-sectoral DPP system ontology expressing the generic DPP data model with relations and hierarchies will be required. Then, once the information requirements of each ESPR delegated act are known, sector-specific regulatory ontologies will be needed to define sector-specific terminology. These regulatory ontologies may or may not be aligned with existing sectoral or branch ontologies, and thus might require an alignment step. The regulatory ontologies may also extend or reuse elements from existing sectoral or branch ontologies.

The creation of such ontologies is technically well understood but represents a social challenge as these ontologies are designed based on the agreement on terminology of a large number of actors, possibly including the digital services department of the European Commission (DIGIT), standardisation committees, and branch and sector actors. To relieve this challenge, it can be said that such ontologies can evolve in the same way as the system, given there is a social framework for the evolution. This social framework can be provided by standardisation efforts but could also be done within the industry association of certain sectors or branches.

The Steps of Expressing and Integrating Data/Metadata with one or more Ontologies

This section describes the expression of data through ontologies and the semantic data integration process. In the DPP ecosystem, it is a requirement to link and integrate data from various sources, to facilitate the data discovery process, to perform data analysis and validation, and to offer integrated query services. The integration process requires the execution of various steps that have to tackle

[5] Introduction to CIDOC-CRM Conceptual Reference Model (ISO 21127) https://projects.ics.forth.gr/isl/cci/training/CCI-Hmerida-23-09-2020/files/4-CIDOC-CRM-GR-2020_09_21_%CE%95%CE%9D.pdf

issues like: (a) datasets are produced, kept, or managed by different actors using different models, schemas, or formats, (b) the same real-world entities or relationships are referred with different URIs or names (and in various languages), (c) datasets usually contain complementary information, (d) datasets can contain data that are erroneous, out-of-date, or conflicting, (e) datasets may follow different conceptualizations of the same domain, and (f) everything can change (e.g., schemas, data) as time passes. In the following, we analyse the main approaches and steps of the integration process.

- **Data Collection & Digitization.** This step includes the collection of data in any format, from their original sources in various ways, e.g., through sensors, databases, or even from printed paper. The final output of this step is a collection of heterogeneous datasets expressed in various formats (i.e., usually in their original format).
- **Data Curation.** This step aims at revising the collected data. Usually the process includes: a) normalizing their values (e.g., capitalization, empty values, normalization of dates, multiple values, etc.), b) cleaning them (e.g., fixing spelling errors, removing anomalies and duplicates, etc.), c) enriching them (e.g., add particular information not existing in the original data) and d) normalize their format, so that they are less ambiguous and their structure is simplified and more convenient for next steps. Here, it is also important (if possible) to assign a unique ID for the same real-world object for avoiding ambiguities. As a result, local IDs can be transformed into global IDs like UUIDs or URIs.
- **Data Modelling.** It aims at adopting one or more proper ontologies based on the semantics of the curated collected data and metadata, and extending it, if needed, so that it can be properly used for modelling the collected data. Part of this step includes the creation of the schema mappings that will describe how the values of the curated collected data will be mapped to particular classes and properties of the adopted ontology or ontologies. Schema mappings between different schema concepts can also be defined. For the DPP system, at minima, this will consist of the sector-specific regulatory ontologies and the corresponding schema mappings. However, each sector, branch or even REO may opt for the adoption of one or more ontologies, that are richer and more expressive and fit better their needs. In the latter case schema mappings between these ontologies and the regulatory ones will have to be created.
- **Data Transformation.** It aims at transforming the collected and curated data/metadata with respect to the adopted ontologies. This step uses the schema mappings that have been identified in the previous step. Part of this step is the generation of the identifiers (e.g., URIs) and literal values for the semantic resources. It also includes the process of instance matching, which identifies different instances referring to the same entity across datasets. This step can also include policies followed for specific data types (e.g., using the same measurement unit when measuring time). The output of this step is the integrated knowledge graph that consists of the ontology-based descriptions. In the context of the DPP system, this step creates the integrated named knowledge graph for the DPP data that will be exposed through the DPP Data Repositories.
- **Data Exploration & Exploitation.** This step includes all the operations that can be delivered on top of each knowledge graph (i.e., DPP), such as access services (e.g., browsing, querying, updating, searching, question answering, etc.), validation services, conflict detection and information extraction services (e.g., extracting particular entities from text).

2.2.3.4 Indicative Tools for the key Steps

Here, we provide an indicative list of tools for the key steps, which are also presented in Table 1.

- OpenRefine is an open-source tool that handles messy data. It runs as a Java-based web application that supports loading a dataset, cleaning and reconciling it, as well as transforming it from one format to another.
- Protege is a free, open-source platform that provides a suite of tools to construct domain models and knowledge-based applications with ontologies.
- CSV2RDF is a streaming/transforming CSV to RDF converter, which can build resource URIs on the fly, can fix and remap datatypes and can map different groups of values to different RDF structures.
- R2RML is a mapping language expressing the transition from relational databases to RDF datasets. R2RML mappings refer to logical tables to retrieve data from a source database. Those logical tables are then mapped to RDF using a triples map, a set of rules that maps rows of the logical table into RDF triples. The R2RML mappings are themselves expressed as RDF graphs.
- X3ML Framework is a suite of tools that is able to support the data aggregation process by providing mechanisms of data transformation and URI generation. Mappings are specified using the X3ML mapping definition language, which is a declarative, human readable language that supports the cognitive process of a mapping. The X3ML Engine is responsible for the transformations.
- Ontop provides a platform and a mapping language that can describe how to generate RDF data from relational databases. Ontop relies on the construction of a virtual knowledge graph, using a virtual integration approach. This means that the original data reside in their original data sources and are not transformed or replicated anywhere. They are rather accessed at query time. The mapping definitions rely on R2RML language.
- KARMA is an information integration tool that enables users to integrate data from a variety of data sources in various formats, such as relational databases, JSON, XML, CSV and others. Users describe their mappings based on a target ontology using a user interface that automates much of the process. The tool also supports the transformation of the data and their publishing as RDF data.
- EasyRDF is a library designed to make it easy to consume and produce RDF. It was designed for use in mixed teams of experienced and inexperienced RDF developers. It is written in Object Oriented PHP.
- RDF serializer is a web service for parsing RDF data and transforming it into other RDF serialisation format, including Turtle, RDF/XML, RDF/JSON, N-Triples, and N-Quads.

Table 1: Examples of Tools/Services for data curation, modelling and transformation

Tool/ Services	Input format	Output format	Data Curation	Data Modelling	Data Transformation
OpenRefine	Tabular	RDF	Yes	Yes	Yes
Protege	RDF	RDF	No	Yes	Yes
CSV2RDF	Tabular	RDF	Yes	Yes	Yes

R2RML	Relational Databases	RDF	No	Yes	No
X3ML Framework	RDF Schema	RDF	No	Yes	Yes
Ontop	Relational Databases	RDF mappings	No	Yes	No
KARMA	Variety of formats	RDF	Yes	Yes	Yes
EasyRDF	RDF	RDF	No	No	Yes
RDF serializer	RDF	RDF	No	No	Yes

It is important to note that Linked data, Ontologies, Metadata and Vocabularies are a conceptual step for interoperability. This means that the tools described are there to help, but the system itself does not have to be built up exclusively with Linked data. There are lot of connectors and APIs that connect to widely deployed legacy data systems. But it remains important to have this interoperability layer in mind when designing the production systems.

2.2.4 An interoperable level playing field

This document intends to explain the technological options stemming from the principles above. But instead of describing a single solution, the document will describe options and suggestions on how to implement them using well known and widely implemented technology. In some cases, recommendations will complement the information for decision makers and implementers. A DPP can be potentially thought of as carrying all B2B, B2C and B2G information of products. Therefore, the architecture needs to be precisely scoped. While a special mention on Dataspaces is included, the connection to other developments will be just mentioned without diving deeper into the question on how to integrate those and the DPP system.

The CIRPASS proposal for the architecture of the DPP system has the function to create an interoperability layer, not to prescribe very specific framework rules or very specific tooling. The basic interoperability layer is provided by web technology. Web technology is known and was designed in CERN to allow for a high variety of information sources and information consumers to communicate. On top of the web we know, a web of data for machines has been constructed and standardised. This basis has very few indispensable requirements. Using URIs throughout the information system is one of those requirements. But as can be seen in [section 3.1.9], even for this requirement, there are ways to accommodate situations where there is not enough space on the product to put a data carrier with a full URI. In this case, transforms are needed. [section 3.1.9] shows possible transformations without being exhaustive. The other central element is a target interoperable data format. But this again, does not mean that there is "the DPP data format". A target interoperable format means a format that is known to be syntactically and semantically interoperable. This document assumes to use [RDF] for that without precluding other solutions. Additionally, the format should have easy, well known, widespread ways to transform from a legacy data format into the target format and back, respectively. The currently standardised graph data formats have those properties. They even translate easily into each other via upper ontologies. Those upper ontologies can then determine which elements in a given vocabulary A are the same element in vocabulary B. It is therefore expected that many industry branch agreements or standards on vocabularies will coexist and that not one single unifying data vocabulary or syntactical format will be imposed.

But in order to be useful, the document will contain hints on how to achieve a certain required functionality. This hint is in no way prescriptive. The authors of this document are aware that there will be many ways to achieve a certain required functionality. In case other means are used, a transformation will be required when talking to others in the network. Note that such transformations consume energy and should be factored into the overall green balance of the entire undertaking. It should be noted that nearly every element in the DPP System can be either implemented as a networked service or as an application feature.

It was equally important that this architecture considers existing legacy systems and gives hints on how to leverage them to produce and manage a DPP. The DPP system described serves in those cases as an interoperability layer between legacy systems. To connect these systems to the DPP system, the use of [IDSA] Connector technology is encouraged.

2.2.5 Use of standardised technology

Unfortunately, technology has not yet evolved to a point where interoperability between services and applications is natural. To achieve interoperability, certain constraints in the design of the system were necessary. The system is therefore based on Internet and Web technology. This technology is well understood and standardized with free and open specifications available to everyone free of charge. Using Internet and Web technologies means that there is a large variety of open source, but also proprietary software, available to easily implement the features required by the DPP System. This includes software for the implementation of the [REO Resolver] and the [Decentralized DPP data repository].

But having the basic components of the level playing field standardised does not mean that any combination of those basic components is already interoperable without additional considerations and further technical specifications. There are still many things to be specified while integrating all those constitutive elements into a working DPP System.

In Art. 8 (2)a [ESPR] refers to Annex III that refers itself to ISO 15459:2016 for the use of Global Trade Identification Numbers or equivalent. Meanwhile, the [ESPR] asks to register the [REO ID] and the [Facility ID] into the [EU Registry], which has no standardised format so far. It is expected that those formats will be made available following the current mandated standardisation effort by CEN/CENELEC.

Even though the data transport and the DPP data use standardized technologies to get from a product to the information, not all protocols and identifiers needed to go from a product to its information are necessarily interoperable to each other. Thus, there may be more than one way to get from the product to the information. **The choice of an identifier will determine the set of systems and protocols to get to the DPP or product information (see figure below).** The information itself, the DPP, has efficient data interoperability up to the semantic layer. To exemplify this situation, this document offers a HTTP scenario following mainly the ideas laid down in the [GS1 Digital Link 1.1.2 Specification] and another scenario where the same functionality is provided in a System using [DIDs] and [Verifiable Credentials]. Both are widely standardised in Common Technical Specifications in the sense of Regulation (EU) 1025/2012 but cannot be mixed without a transformation step.

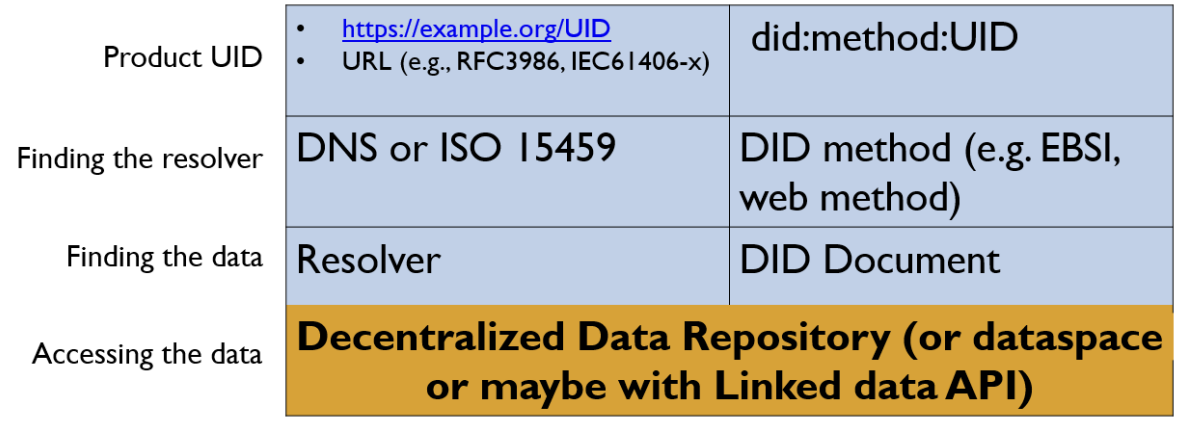

Figure 1. Different systems and protocols deployed for different Product UIDs

2.3 DPP system architecture validation

Based on the requirements and design principles defined above, this document first presents an architecture and associated data flows for the DPP system assuming the use of either HTTP URIs or DIDs to connect tangible products with their associated data. Validation of both architectures is performed by ensuring that all technical requirements defined in each DPP user story are met.

3 Structure of the DPP System

As was already discussed in [section 2.2.4], there are many ways to design and implement a DPP system. This can be a central database, a framework or just a way to use existing infrastructure. We depart from the assumptions and principles made in [section 2.2]. The structure presented herein is not the only one possible to fulfil the requirements set forth in [section 2]. Because the DPP System is product centric, it is a way to discover information while holding a tangible good in some way. The DPP data itself is supposed to be interoperable on the syntactical and semantic layers. Because a graph and ontologies are used, it is easy to merge information from various sources. And because graph data uses standardised formats that are widely used, there are many tools already available for easy transformation of DPP Data into something that can be extracted from or used by the existing systems. It is anticipated that a substantial part of the DPP data will be extracted from Enterprise Resource Planning (ERP), Product Information Management (PIM) and Product Lifecycle Management (PLM) Software, transformed to fit the DPP system's need. And that DPP data can be easily imported into such ERP systems. The system architecture also considers the easy integration of existing systems, like the widespread systems used in retail.

This document presents two alternative architectures to access the DPP data, knowing that there are more alternatives. This only concerns the way to find the DPP data starting from discovering a Product UID. The DPP Data itself remains the same whatever route to access is chosen. The two ways were selected because they are very commonly used and integrate well into the existing and available Internet infrastructure. This will allow for a maximum reuse of a technology stack and network infrastructure commonly available. Presenting only those two ways does not preclude industry branches with an obligation to provide a DPP to come up with their own way to go from a [Product

UID] to the DPP. Whatever means are used to access the DPP information, the DPP information itself uses a highly interoperable standardised format that allows for easy merging, splitting and analysis of the data contained in the DPP. As pointed out in [section 2.2.3], this interoperability includes transformation from and to legacy formats, integration into dataspaces and more.

The two ways of getting from the product to its DPP have similarities and differences. The document will first explore a structure that relies heavily on the HTTP-world of protocols, including [TLS] and web servers. The basic ideas behind this structure are inspired by the GS1 Digital Link (DL) Standard in its [version 1.1.2]. In this scenario, the [Product UID] is either a URI or can be transformed to a URI. But instead of pointing to the DPP information, it points to a resolver that can react on roles and does redirects. In the second scenario, the [Product UID] is a Decentralized Identifier [DID] and uses a predefined method to discover information about this DID. The DID method allows to discover a [DID Document]. This DID Document then serves the same purpose as the resolver in the HTTP or DL scenario. But the description of the DID scenario adds additional functionality that is enabled using this way to link a product to its DPP, including identity management, access control and more. The non-congruence of both scenarios is, thus, intended to show the potential of the suggested structure. A dedicated application to discover DPPs may choose to implement both suggested structures in one application.

Both scenarios start with a diagram depicting the structure and the structural elements of the intended architecture. This section only describes the structural elements. The relation between the elements and the data flows is detailed in [section 4].

3.1 Structure of the DPP System using HTTP URIs

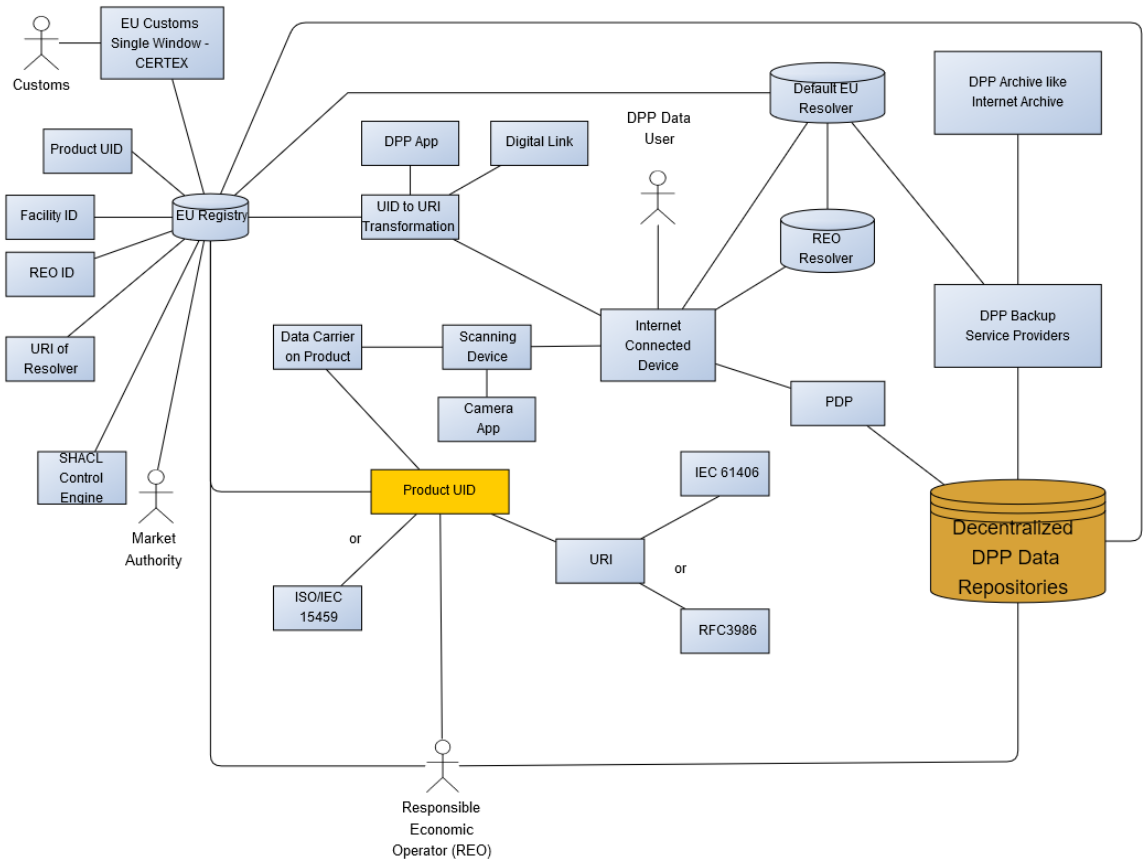

Figure 2. Structural view of the DPP system showing the structure, actors and components of the system without showing the data flows for the HTTP architecture.

The above structure describes the HTTP centric architecture that is inspired by the [GS1 Digital Link Specification 1.1.2]. The detailed diagram for the DID scenario is in [section 3.2]. The diagram does not contain any data flows. The lines indicate a relation of any type between elements of the diagram. Section 4 will add details about the relations and how they lead to specific data flows. The points below will therefore only describe the structural elements or actors and their respective role within the system. For information on data flows between those structural elements, see [section 4].

Below, all elements of the system as shown in the diagram are detailed. Some of the elements represent actors in the system. Actors are initiating actions in the system because they have the obligation to provide information or because they need information. Other elements are just system components that respond to requests or that have an API to allow them to receive or to send information.

3.1.1 Responsible Economic Operator (REO)

Art. 2 (46) of the [ESPR] contains a legal definition of the REO: an " *'economic operator' means the manufacturer, the authorised representative, the importer, the distributor, the dealer and the fulfilment service provider;*" that is responsible for placing a product on the market or putting it into service. This definition remains unaffected by the recent changes proposed in the European Parliament in Document TA-9-2023-0272.

The REO is a central actor in the system. The Draft Regulation assumes that the economic operator in this wider sense is responsible to make sure the [Product UID] is created and attached to the tangible product and that at least the mandatory information is put into the DPP and made available.

An important question is the creation of an European identifier for REOs that need to register the [Product UID], a [REO ID] and a [Facility ID] into the [EU Registry] according to Art. 11 [ESPR]. This question will be treated there.

There are many remaining questions regarding that central role of the REO. For new products, the assumptions made are not complex. An economic operator who introduces a product into the single market will have to create all necessary information and will be responsible for the DPP. It is understood and not very complex that, in a product centric system, the [Product UID] remains immutable on the product with the data carrier immutably attached to the tangible good it identifies.

But once the basic actions and responsibilities are clarified, there are difficult questions appearing. The REO could remain responsible for the DPP information for the lifetime of the product. But what is the lifetime of the product in a circular economy? While the legal boundaries between a refurbished and a remanufactured product are clear (a refurbished product IS NOT a new product, a re-manufactured product is a new product), the boundary between the uses of the DPP for an existing product and a potential new product begins to blur in case of the remanufacturing or refurbishment. Indeed, for many practical reasons related to refurbishing and remanufacturing processes, product UIDs and existing data carriers might continue to be used or new ones issued independently of the above legal boundaries. If the [Product UID] on the tangible good remains the same, who will now be responsible to bear the cost of the information system? The [ESPR] gives a first hint in Recital 14a by stating that "*products that are remanufactured are considered as new products and they are subject to ecodesign requirements if they fall within the scope of a delegated act*." The remanufacturer is a new downstream REO and the initial REO is released from their duties. The system can thus not assume that the REO remains the same legal entity for the lifetime of the product and that the first REO must bear responsibility for the DPP as long as the product exists.

There are many solutions to these issues. A delegated act as foreseen in Art. 4 [ESPR] can determine the boundary between repair, refurbishing and issuing a new, remanufactured, product. In case of remanufacturing, the remanufacturer must have access to the entire old DPP Data. Up to and including refurbishing, the DPP data is updated. But in case of remanufacturing, a new DPP must be issued, and a new data carrier must be associated with a new [Product UID]. It is a non-trivial question whether the remanufacturer could be enabled to re-use the existing data carrier on a remanufactured product. It is trivial to import relevant data from the previous DPP into the new DPP, if the remanufacturer has access. It would help if this access could be secured by law via the delegated acts for industry branches and product categories, where this makes sense, especially for higher complexity products that are assembled from independent parts like machines or cars.

The REO also must bear the responsibility to manage access for read and write events to the DPP. This way, a repairer can add all relevant information about a repair event into the DPP. To do so, the repairer would need permission and it is the assumption here, that the REO is responsible for the identity management that allows to enrol arbitrary repairers. But given the decentralized nature of the DPP graph structure, it is not excluded that a system could be designed that separates the [Decentralized DPP Data Repository] of the REO and the repairer. In this case repairers would write into their own repository that is queried on the fly in case a DPP Data user is querying for repair events.

But even in this case, the REO remains the first contact and thus in control of all further information delivery. A repairer, despite having hypothetically their own repository, would have to enrol with the REO. It is expected that delegated acts pursuant Art. 4 will also determine the access level to certain data points.

Looking into the future, it is expected that a REO of a higher-level product creates the DPP by assembling DPP information they received from suppliers. If components have DPP information, the DPP can not only be used for the creation of the higher-level DPP. A system could also preserve the provenance of such data in the DPP following the standardized [Provenance Vocabulary]. The assembly of DPP data that is then stored into the [Decentralized DPP Data Repository] requires merging of data from a variety of sources and network interfaces to upload that information to the relevant network services. The considerations around the merging of existing information into a DPP is done in [section 4.1.1.2]

3.1.1.1 REO ID

The [ESPR] remains enigmatic in this respect when noting in Art. 11 that a REO "*creating or updating the product passport shall request a unique operator identifier on behalf of the relevant actor*". As the ESPR itself mentions the use of Global Trade Identification Numbers as provided by the ISO/IEC standard 15459:2015 (or equivalent ones), one could speculate that "*relevant actor*" could mean the relevant registry in this standard. In this case, GLNs (Global Location Number) from GS1 could be used and would identify legal entities and locations. But the enigmatic wording could also open the path to REO IDs that are linked to the Commercial Registers of the EU Member state. Another possibility is the use of DIDs to identify the REO. This is explored in [Section 3.2].

3.1.1.2 Facility ID

The REO will also have to provide a [Facility ID] to the [EU Registry]. Art. 11 [ESPR] just refers to Annex III. But Annex III (i) just mentions the Facility ID without defining it. A first hint is that Annex III talks about a "*unique*" facility identifier. It remains unclear whether a facility producing for many different brands will have several "unique" facility identifiers related to the brand or whether the identification scheme is really facility centric. In the latter case, a DPP crawl in the EU Registry will reveal all customers of a certain factory, e.g., a fashion factory producing for many brands. The notes on the REO ID apply. The default will probably be an implementation of the ISO/IEC standard 15459:2015 using GLNs or equivalent such as the IEC 61406-x [IEC 61406-1, IEC 61406-2], with other options taken up from the DID scenario or while re-using the identifiers from Track & Trace systems.

3.1.2 DPP Data user

The DPP Data user is a generic placeholder for a large variety of roles in the context of the information system for an eco-responsible and sustainable economy. DPP Data users are actors wanting to access the DPP data for a large variety of reasons and according to certain roles (e.g., section 3.3 of Deliverable D4.1 proposes a non-exhaustive list of such possible roles). The system can bear an arbitrary number of roles without suffering additional overhead for interoperability. From the use cases specified in D4.1, we will describe additional roles to give examples for such roles on top of the default consumer role. As can be seen in [section 4], the roles can be used directly to affect the type, quality and quantity of data or information delivered to the DPP Data user. This may depend on needs and access rights of the DPP Data user in question.

Not all DPP Data users will start with the [Product UID] described in [section 3.1.4]. The scanning step with a scanning device as described in [section 3.1.7] is only necessary, if the Product UID is not already known to the DPP Data user. This means that a DPP Data user, such as a consumer, can act as a product centric actor, but other actors can use this architecture in an information centric manner. Detailed options and data flows will be described in [section 4].

This should detail the various roles of people scanning the Data Carrier. Those roles will then directly be translated into a Typed Link request that is described later. The next subsections describe the most common actor roles expected to be present in the DPP system.

3.1.2.1 Consumer

The default DPP Data user is the consumer. A consumer is expected to scan a [Product UID] with some intelligent device like a smartphone, and expects to get human readable information in return as described in the user stories in D4.1. If no other additional information is provided by the requester, the system will always assume to serve a Consumer and will return human readable information. It is expected that this information will be in HTML and CSS to allow for a correct display across a high variety of devices that can display information in a responsive way considering all accessibility requirements, even on devices with constraints.

3.1.2.2 Circular Economy Operator

The Circular Economic Operator role (CEOP) is a very generic role that abstracts all stakeholders involved in the circular economy, including but not limited to following roles: Sorters, Recyclers, Remanufacturers, and Repairers. The CEOP is expected to use dedicated software to consume DPP Data and to interact with the system. CEOPs could use human readable information generated from DPP Data, but it is highly likely that CEOPs will consume DPP Data in a machine-readable format that eases further processing in their own production chain. Depending on their needs, they will only consume data points in a syntactically interoperable format. Or they will consume the full DPP knowledge graph to allow for logical operations within more complex machines or robots.

While the Consumer is expected to have a purely passive reading role, some CEOPs are expected to write data back into the system. If an actor is supposed to consume privileged information or if an actor has the privilege to also write data or information into the system, this actor needs to be identified at least in a pseudonymous way.

Note : the needed identity management system is not fully described in this document. A beginning of a solution is detailed further down in [section 3.2] describing the structure while using DIDs and within [section 4.3] on DID and VC Data flows.

With the different roles of the DPP Data users or CEOPs come different requirements concerning access to data, the data format and the various data flows and operations needed.

If the product is repaired, this can be done by the initial REO. It is worthwhile for the circularity of the product in question to be able to follow the replacement of components or modules of a higher level or complex good e.g., tools, machines or cars. If this is done by the REO, this latter can just update the DPP knowledge graph in very easy ways. But the average case will involve some independent repair shop. This repair shop is also expected to write information back into the DPP. But to avoid DPP mutilation and Graphiti, the repair shop must enrol into an identity management system that gives write access to the DPP. Fortunately, the evolving eID system of the EU will facilitate the creation of

such ecosystem and make it easy to enrol online. There is a risk of anti-competitive behaviour though when excluding repair shops. The Competition authorities should be invited to take notice of the DPP System.

Refurbishers and Remanufacturers can be seen as an extensive repair. There is a borderline where Remanufacturers become themselves REOs. In this case, the control over the DPP information changes. As a DPP Data user, the access control system of the DPP system determines whether access is given to all DPP information. For Refurbishers and especially for Remanufacturers who will take over responsibility for the DPP, it is therefore essential that they have access and can duplicate the full DPP into their own systems. This will then facilitate the creation of a new DPP for the remanufactured good.

A recycler dismantling a product may also note into a DPP that a product was dismantled and recycled. It may be interesting for public authorities to know how much of a product was recycled or how many times an item has been recycled, and exploit the DPP system for extended statistics.

It is expected that CEOPs will indicate their role in the Link-type variable when doing a HTTP GET request on the URI resulting from the Product UID. This will allow to filter down an extensive amount of DPP Data to the CEOPs precise need. It also allows for very specific requests, e.g., the transformation of the DPP Knowledge graph data into the Administration Asset Shell (AAS) [IEC 63278] format that can then automatically be consumed by the machines that made that request. Standardisation for Link-types for typical roles of DPP Data users is expected to help the adoption of the DPP system.

3.1.3 Public Authorities

Public Authorities can be divided into three main categories: regulators, customs, and market surveillance authorities. The requirements known so far indicate that Public Authorities are mostly information centric, and very few DPP access requests will start by scanning the ID of a product. The authorities are rather looking whether required information is available and compliant. Another angle is the need for large scale precise statistics that help steering the economy. This becomes possible with an ubiquitously present DPP system in a given branch.

Regulatory authorities will benefit from a privileged access to the [EU-Registry] further described in [section 3.1.5]. The EU-Registry serves as a way to gain overview over the market concerning facts that are of interest to Public Authorities.

3.1.3.1 Regulators

Regulators are expected to set certain rules within the DPP System. The Draft ESPR sets forth requirements for a DPP in Art. 7 – 13. Additionally, in Art4, it allows the European Commission to adopt delegated acts with further ecodesign requirements, including rules for the DPP. While the ESPR already establishes guidelines for the information content of future delegated acts, these delegated acts will define the mandatory data points to be included in the DPP.

One way to automatically check whether a given DPP complies with the rules is for the Public Authority to provide a formal description (i.e., non-ambiguous) of the rules. To achieve that purpose, the rules will be denoted into a [SHACL] template. All REOs will hence be able to self-check compliance with the requirements for the product category of their product subject to DPP requirements. But not only REOs are able to check DPPs against that SHACL template, but also other Public Authorities. The benefit of using a formal format like SHACL is that it removes most of the ambiguities compared to

natural language, even if used as legal language. The disadvantage is that the burden of resolving those ambiguities is now on regulators when defining the rules. But a little higher investment on the top will yield economies of scale downstream in the system. Regulators may also need to access aggregated data and statistics as mentioned above to refine the regulations, release important burdens or otherwise adapt existing or create new rules.

3.1.3.2 Customs

Customs at Union borders control more than 350 sectoral legislations. In addition, the tasks handled by Customs are not uniform across EU Member states.

In accordance with ESPR, customs will verify that the products placed under the customs procedure release for free circulation (import) have a valid DPP. This verification will take place electronically and automatically through the interconnection to be created between EU Customs Single Window Certificates Exchange (EU CSW-CERTEX) and EU Registry. But customs may also use the SHACL validation templates which will likely be made available by the EU – Registry to validate the correctness of the DPP.

Moreover, Customs and the Commission are privileged users of DPP and the [EU-Registry] as they may retrieve and use the information included in the DPP and the EU Registry for carrying out their tasks including in the future Customs EU Data Hub. Customs can thus benefit from the DPP system in an information centric way, but also in a product centric way.

3.1.3.3 Market Surveillance Authorities

Market surveillance authorities check product compliance against ecodesign rules. These public authorities may use the DPP in either a product-centric or informationcentric way. They can access the DPP Information via the DPP [Data Carrier] or via the information contained in the [EU-Registry]. They can consume either human readable data or the machine-readable data formats. The data selection displayed will be dependent on the context of the situation leading to the access to DPP Data. The [ESPR] describes a web portal in Art. 12a but its implementation will likely behave similarly to a specialised search engine, especially as the obligation to register all Product UIDs with the [EU-Registry] will allow the search engine to cover nearly 100% of the available information. Such a specialized search engine could make automatic conformity assessments using [SHACL]. The current work of DG IT in the area of data catalogues, the so-called core vocabularies, will prove to be immensely useful. Market surveillance authorities may require privileged access queries to establish statistics about products as is done e.g., in multiparty computation to generate non-identifying statistics without revealing any commercial secrets.

3.1.4 Product UID

The DPP system architecture only works if a Product UID is globally unique or can be made globally unique when scanning the [Data Carrier], the first option being mandatory for the EU DPP. The Product UID is essential to link up the tangible product with the information about that tangible product. The Product UID itself does not have to be in the form of a URI. Often, constraints on space and memory for the [Data Carrier] will mean that a number on a product must be rather short. But those short numbers can be transformed into URIs by the [ICD]. The DPP System proposed in Figure 1 provides modules to transform those shorter numbers into globally unique URIs that are resolvable in the

system. Which means that the short Product UID does not have to be a URI or URL in the sense of [RFC3986] or IRI in the sense of [RFC3987]. But there must be an automatic and standardised way to transform the Product UID into a string that is globally unique and fulfils all the requirements from RFC3986 or RFC3987, respectively. In case Decentralized Identifiers [DID]s are used, they are URIs in the sense of RFC3986. But they will have their own transformation mechanism, explained in the DID [section 3.2]. The details on the transformation of numbers into URIs will be given in [section 3.1.9].

This section assumes a Product UID but does not detail how such a UID is formed, designed or constructed. Requirements for unique identifiers suitable for use with this system are exposed in [CIRPASS Deliverable D3.3.] There are many options, ranging from pure dereferenceable URIs according to [RFC3986] over the use of [Digital Links] up to IEC 61406-x specifications that add principles and restrictions to the formation of a URI [IEC 61406-1, IEC 61406-2].

To be machine readable, the Product UID will be transformed to be recorded in a Data Carrier. For example, in the case of a QR code, the alphanumeric string will be transformed into a machine-readable graphics that is then printed onto the product. The Data Carrier will be immutably attached to the product subject to a DPP obligation.

DIDs are also taking the form of a URI, in case the REO uses DIDs for identification. However, although DIDs are URIs, they need different steps to resolve them and to obtain the corresponding DPP. Those will be detailed in [section 3.2.1].

Art. 9 (3) [ESPR] orders the REO to provide the Product UID also for online marketplaces. However, the online marketplace does not give access to the tangible good which would allow direct access to that good's DPP. But Art. 9 ESPR requires a DPP user to be able to discover DPP information from that online offer. There are several ways this could be realized. First, a simple link, e.g. under the photo of the good, can be created using HTML link elements. That link then points to the resolver which either delivers information or redirects to the DDR.

Online sales will often only offer information related to fungible goods. In this case, the good is not individualized and no individual item or batch level DPP information can be given. In this case, the link described has to point to model level information. At a later stage in the purchase, the good will be individualized before it is delivered to the purchaser. At this point, an individualized Product UID can be given, if required e.g. by a delegated act. This individualized Product UID will then point to DPP information that contains at least the model level information already visible when looking at the good online.

3.1.5 EU-Registry

Art 12 (1) [ESPR] orders the European Commission to set up a European Product Passport registry called *the registry*. To achieve disambiguation with other registries, we call it the EU-Registry. According to Art. 12, the registry must at least include the following elements:

1. The unique product identifier, see [Product UID]
2. The unique facility identifier, see [Facility ID]
3. The unique operator identifier, see [REO ID]
4. The unique registration number
5. The product commodity code (in case of products intended to be placed under the customs procedure 'release for free circulation')

6. The batteries unique identifiers according to Art. 77 (3) of [Regulation (EU) 2023/1542] in case the Product is an industrial battery with a capacity greater than 2 kWh or electric vehicle battery.

The Commission can mandate any other information to be added in delegated acts according to Art. 4 [ESPR]. This could include, for example, the link to the currently active resolver.

While, according to this architecture, the unique registration number and the [Product UID] could be the same number without any negative effects on the system, Art. 12 (4a) of the [ESPR] states that the EU-registry shall automatically generate and communicate to the economic operator this identifier upon upload of the information associated to the three identifiers for a specific product. Via the DPP knowledge graph, the [Product UID] is linked to all other information like the REO and other information. As the [Product UID] is supposed to be globally unique it can also link to registration information. But depending on implementation details, mainly, if the EU-Registry is implemented as a SQL database, a lead table is needed that would carry the unique registration number according to Art. 12 (4a) [ESPR]. This means the additional registration number has no influence on the overall system here and is just an additional data point that is carried only in the EU-Registry.

In case, the [Product UID] stored in the [Data Carrier] is not a dereferenceable URI, a [UID to URI transformation] from that non-URI [Product UID] to a URI must be done in some way for the system to function properly. It can be assumed that a normative interpretation will make clear that [Data Carrier] and [Product UID] are linked via a standardised and well-formed procedure. If only the UID is stored in the EU-Registry, the [UID to URI transformation] must be known to the EU-Registry. It is therefore recommended, that the EU-Registry rather stores the full URI. This can be done at registration time by submitting a URI or within the EU-Registry by applying the known transformation to the number submitted. A special attention should be given to current works on ISO/IEC JTC1 [DIS 18975].

Many products regulated by product-specific legislation under the [ESPR] framework legislation and put on the market will have mandatory DPP issuing requirements and therefore will have to be registered in the EU-Registry. This creates a central point of information in an otherwise decentralized system. While this has significant advantages for market surveillance, a significant load on such a system can be expected as all decentralized actors will have to interact with it, which can undo some of the advantages of the decentralized nature of the system. If further opened, the EU-Registry would allow a DPP search engine to easily collect and index data. This would allow to build a Web portal according to Art. 12a [ESPR] that does not hold the information itself, but only indexes the information available in the system and points or links to the otherwise decentralized information. This can work for normal consumers via some web browser, but also for robots doing data science and reporting results in a machine-readable way.

After having minted a [Product UID] and after having associated this [Product UID] immutably with the product, the REO is expected to register the information mentioned above to the EU-Registry via a standardised API. Allowing REOs to store information about the REO resolver in the EU registry would be advantageous, as this would allow to explore all possible ways to access the DPP. The exact data flows and details on how the REO submits the information will be further detailed in [section 4].

The central EU-Registry can be constructed as a simple and fast key–value store that can hold information, depending on the requirements for the product class given. According to Art. 12 ESPR,

the REO may be obliged to send the [Product UID] and required additional information to the EU-Registry. In particular, it would be a good idea to oblige the REOs to also provide information on how to reach the long-term archive or EU archive in case the REO's server is not available. Indeed, additional sustainability of the system could be achieved if the information in the EU-Registry would also contain a link to the resolver of the DPP service provider which holds the DPP backup for a given REO. A "resolver" is a commonly used web service that receives incoming requests, formulated in the form of a URI, and then redirects the request, in the sense of [RFC9110], to the appropriate target (another URI) or targets (a list of URIs). This additional feature is exemplified in this document via the DID data flow section.

As currently constructed, Art 12 [ESPR] requires the EU-Registry to contain all unique product identifiers of products subjected to mandatory DPP requirements. As already indicated, having all [Product UID]s of the European single market in a database can add up to a lot of information. Therefore, in Annex III, the [ESPR] allows delegated acts to only require DPPs with batch or model-level granularity. However, with model or batch level identifiers, the DPP of an individually repaired, customized or refurbished product cannot be found. In this case, the EU-Registry can only point to a model or batch DPP which in turn makes the EU-Registry an information-centric system that can harvest information on certain models or batches of fungible goods. But repairs or lifetime information about individual products cannot be accessed. This will greatly reduce the utility of the DPP system and the aforementioned potential utility of the EU-Registry. A careful evaluation of the product class should be done before using model or batch level [Product UID].

Another very nice functionality could potentially be carried by the EU-Registry: If, for a given [REO-ID], the EU-Registry also registered the URI of the [REO Resolver], the EU-Registry could act as a Resolver of Resolvers[6]. This can be compared to the Domain Name System (DNS) where the EU-Registry would play the same role for DPPs as the Root – DNS servers play for resolving hostnames. All nodes in the decentralized DPP system would then have a central point to find information in case of changes. The EU-Registry would only return the then current REO-Resolver and allow a requester to then further explore the contents of the REO's systems. Such information about a backup resolver within the EU Registry, would provide an easy resilience against REO failures, e.g., if a REO goes out of business. If receiving an HTTP status 404 (not found) upon a request to a Product UID, all EU DPP applications could ask the EU-Registry for the currently relevant resolver. At the same time, the full list of REO resolvers will allow Market Authorities to create search engines that can batch-crawl all relevant Resolvers and then access the relevant DPP Knowledge graph with a privileged access to the product information. This way, targeted statistics or compliance controls are very easy to implement. And it could be the backbone for a working archiving system for DPP information in case the REO goes out of business. It is understood that the EU-Registry will not hold that archive itself, but that it would be the central point of information pointing to where each information is stored.

3.1.5.1 The Validation & Control Engine

The DPP is modelled as a knowledge graph as explained in [section 2.2.3]. To make the knowledge graph interoperable, it should follow the relevant W3C standards, especially [RDF 1.2]. The DPP graph in RDF, whatever the serialization, can now be checked for correctness. The way to do this with graphs is to use the Shapes Constraint Language [SHACL]. The [SHACL] Control Engine has two functions. It

[6] See also Section 3.1.10.2 on the [Default EU – Resolver]

will hand out templates to REOs to validate their DPP. And it will be able to help Market Authorities to validate DPP information found while searching the DPP dataspaces.

Looking at this in detail, it has to be noted that the DPP knowledge graph contains information about a product including semantics. And because it is linked data, it also contains information about relations between certain of its data points. [XML Schema] can be used to validate and constrain information expressed in XML. This way, a system can constrain a field to only contain numeric and not alphanumeric characters. Trying to enter "A" or "B" into that field yields an error. This way, things can be validated on a syntactical level. But this is not sufficient to validate things on a semantical level, let alone on a relational level.

[SHACL] is a language for describing and validating RDF graphs. [SHACL] allows to construct a so-called *shape*. The shape is in fact a template description of elements and relations in an RDF graph that expresses constraints over the values of those elements and relations. With SHACL, it is possible to construct a model graph that must be at least present. The presence of all elements, relations and values in the actual graph is then checked against that template. Failures can be reported precisely.

This is a very generic and powerful mechanism to test a DPP and to validate its content. Simple regulations like parts of Delegated Acts according to Art. 4 [ESPR] could be translated to [SHACL] shapes by the regulator as already mentioned in [section 3.1.3] on Public Authorities. All actors involved, the REO, Customs, Market Authorities and other service providers can then check the validity of the DPP automatically, without any bureaucratic overhead. REOs could test automatically the DPP they are about to submit to the [EU-Registry] or to the Decentralized DPP data repository. Thus, this quick validation check could be done before DPP registration in the EU-registry, during the registration process (automatically), and at any other time after that.

The [EU-Registry] is an ideal place to carry that functionality for REOs and Public Authorities alike. It allows for the verification mentioned in Art. 12 (2)a [ESPR].

3.1.6 Data Carrier

The Data Carrier is either some QR-code, RFID chip or other Data Carrier that is ideally immutably attached to the product. In Art. 9 (1) a of the latest iteration of the [ESPR], it is stated that “the data carrier shall be physically present on the product, its packaging or on documentation accompanying the product, as specified in the applicable delegated act adopted pursuant to Article 4”. In the DPP system, the Data Carrier contains a physical instantiation of the [Product UID]. The only condition is that the number must be readable by a scanning device that can extract the number from the Data Carrier. Data carriers often have technical limitations in terms of memory or characters they can represent. The more information is encoded, the bigger the QR-codes become. On a small item, it will be difficult to encode much information in the Data Carrier. The only thing the Data Carrier needs to contain is the [Product UID] as it is the necessary condition for the system to find the corresponding named graph or DPP. This is why it is recommended to only carry the [Product UID]-related URI. In [section 3.1.4] there is an additional option where identifiers are even shorter (and are therefore not URIs) but there is a known transform to construct a URI from that number or ID. This option is explained further down in [section 3.1.9].

Art. 9 (3) of the [ESPR] covers a situation where a product is sold online and no physical access to the physical good is given. In this case, the online marketplace is still obliged to provide an easy access to the DPP information. To do this, the REO can either provide a copy of the Data Carrier, most probably

a QR-Code (this can be helpful when browsing an online shop or a flyer with a mobile phone) or the REO can provide a link that can be clicked. Both can be easily implemented in the System presented here.

However, whenever the Data Carrier holds a link, it needs to be a canonical URI as defined in [RFC 6596]. In more details, "*The canonical link relation specifies the preferred IRI (Internationalized Resource Identifiers) from resources with duplicative content. Common implementations of the canonical link relation are to specify the preferred version of an IRI from duplicate pages created with the addition of IRI parameters (e.g., session IDs) or to specify the single-page version as preferred over the same content separated on multiple component pages*". Canonical GS1 Digital Link URIs are defined in [GS1 Digital Link 1.1.2 Specification]. ID Links as defined by [IEC 61406-1, IEC 61406-2] are URIs, but also need to be made canonical here according to the [RFC 3986] section 3.1.

3.1.7 Scanning Device

A scanning device is a device capable to extract the [Product UID] from the [Data Carrier] by some technical process. This process can be optical, like in the case of QR-codes and Bar-codes. But it also can use radio waves to extract a number from an RFID tag. The scanning device can be integrated in the Internet connected device or be totally separate from it. The scanning device just needs to communicate to the Internet connected device in some way, not necessarily via internet. A mobile phone e.g., is a scanning device and an [Internet Connected Device] at the same time. A cashier scanner that just scans things and communicates via a proprietary protocol to the check-out system is also a scanning device, despite not having any logic to decode or transform the [Product UID]. The scanning device reads the [Data Carrier] and pushes the gained information onwards into the system. However, ongoing development in the retail and scanner industry [2D-Retail] will add the capacity for point-of-sale scanners to parse GS1 Digital Link compliant URIs and extract GTIN and any available supplementary data from the URI. [IEC 61406-1, IEC 61406-2] defines additional constraints on top.

3.1.8 Internet Connected Device (ICD)

An Internet Connected Device is a computational device capable of receiving the information of the scanning device and to further treat that information. The ICD is a module or application that has several tasks.

First may transform [Product UID]s into something useable by the system. If a product is small and has not much space, the QR code may be very small and only carry a number. The scanning device serves that number to the ICD. The ICD can now do the transforms from [Product UID] to [Resolver] URI itself or it can ask a service online to do the task and to return the result.

In its most basic instantiation, the ICD is an application on a mobile phone that scans a QR code and returns product information on the screen, which can be done by the phone's camera app or with a dedicated Application. But ICDs can also be highly sophisticated operational modules in a recycling machine that reads and caches Product category data and does automatic sorting while products pass by on a belt. If it hasn't cached the information, it can go looking for the information it needs.

The ICD is under the control of the DPP Data consumer and serves the DPP Data consumers by delivering information to them to be displayed or otherwise computed in order to make decisions based on DPP information.

3.1.9 UID to URI transformation

If the DPP is supposed to work also in the retail chains without much change, bar codes must be considered. Very small products have limited space for QR-codes. Small RFID tags have limited storage capacity. Many products in the market have already numbers. Cars, for example, have a Vehicle Identification Number that is linked to information systems. But those are not in the form of a URI. How to integrate them easily into a DPP system? If not yet a canonical URI in the sense of [RFC 6596], the challenge is to transform a globally unique number into a canonical URI. This way the number is transformed into a form that allows to get to the DPP data about that [Product UID]. This means, the number, directly or indirectly, must enable the information discovery, the path to the DPP information, in some way. This document assumes that information discovery is either done via a URI or via a DID (see [section 3.2]). The UID to URI transformation module represents a step necessary if the required information is not already fully encoded in the right format into the [Data Carrier]. This will currently often be the case. An application reading that number needs to proceed with a transformation step or call a transformation service before giving a usable URI or DID back.

3.1.9.1 Camera App

The Camera App is just the normal camera app on the average smartphone. Most of those are capable to decode a URI from a QR-code. In case the [Product UID] is a URI, most smartphones can send a [HTTP 1.1] GET request to some server. This is the default scenario. The data flows for this case will be detailed in [section 4].

3.1.9.2 GS1 Digital Link

Bar codes on products have a long tradition. Their widespread use started already in the seventies of the last century. Bar codes represent a Global Trade Identification Number, a GTIN. They follow the rules set forth in [ISO/IEC 15 459]. GS1 has issued a new standard specifying how to transform a given GTIN into a URI. A Bar code reader is a scanning device in the sense of this document. But neither the Bar Code nor the resulting GTIN are a URI that allows an application to find a resolver that will tell it where to find the DPP data. But if the application has implemented the Digital Link transforms of GS1, the GTIN found on the product and serving as [Product UID] can be easily transformed into a URI. This URI will point to a resolver and the resolver itself will return the place where to find the appropriate DPP data.

3.1.9.3 Web link & ID-link

Identifiers can be implemented on a granularity level that is useful for the product under consideration: either on item level for products that bear an individual serial number, or on product model or batch level where serialization is not applied. In case of ID-link [IEC 61406-1, IEC 61406-2] specify the relevant principles and restrictions for the identifier. Those are additional constraints compared to a free use of [RFC3986] to serialize URIs as identifiers to identify products. [RFC 1738] is an obsoleted specification for URLs that simplifies the resolution of the identifier to the final source of DPP data. By using the Internet Domain Name System from the host part of the URL, Manufacturers can define and take responsibility of their own domain name, and codify product identification (on model, batch, or item level), the reason being:

- That way the identifiers can be assigned without a central registration authority. Only the domain name must be registered in the DNS system, like for every URI according to [RFC3986]

- no incremental cost for each created identifier.
- Weakness of the Domain Name System, hence the forward-looking suggestion to use DIDs in [section 3.2]

3.1.9.4 Other methods

It is not excluded that a branch of industry invents their own numbering scheme that fulfils the requirements set forth in Deliverable D3.3 and offers their own standard telling how to transform the numbers into URIs.

The caveat here is that, except for using DIDs as explained below, inventing more numbering schemes will require some intelligence in a DPP application or some service capable of transforming them into a URI in the sense of [RFC3986], and even better into HTTP URIs. Those naming schemes, before transformation, are not interoperable as such. If they are done in open specifications, it allows everyone to produce their own software to decode them. It is therefore encouraged to only use HTTP URIs or DIDs with widely known methods. It is recommended that the transformation of [Product UID]s into URIs should be standardised in a common technical specification referred to in Directives 2004/17/EC, 2004/18/EC and 2009/81/EC, and Regulation (EC, Euratom) No 2342/2002.

3.1.10 Resolver

Printing a URI into a QR code was invented in Japan in the late nineties. The QR code is decoded, a URI is discovered, and a web browser will do a HTTP GET request on that URI. This will return an HTTP object, normally a web page that is then rendered within a web browser. For many reasons, just returning a web object from a URI constructed from the [Product UID] is not good enough for the DPP. It would reduce the DPP to a consumer information tool, a kind of dedicated and regulated consumer information page. Such a convenient simplified manual would be a step backwards. Instead, CIRPASS proposes an information system for the DPP system optimized towards data reuse by all actors of the circular economy that also includes imperatives of protection in a competitive market of circular economy actors.

A DPP system will have to cope with a complexity that is significantly higher. For a consumer, of course, it should just return a very relevant web page. Therefore, the default redirection is towards a webserver that will itself draw information from the data repository and send back a web page. For other actors of the circular economy and for market authorities, it should return machine readable information adapted and aligned to their needs. DPP information is expected to grow, and the architecture must respond to this perspective by offering highly sophisticated mechanism of role-based information filtering. To address both syntactical interoperability challenges and challenges around information filtering, the resolver is a key node in the CIRPASS proposal for the DPP system.

[GS1 Digital Link 1.1.2], again, serves as a trail blazer. Instead of requesting a URI to a web server, the expectation is that the URI related to the [Product UID] will point to a "Resolver". This resolver will then redirect the request to the appropriate target. Redirect means redirect in the sense of [RFC9110]. To express the roles and relations, GS1 Digital Link 1.1.2 uses [RFC8288] to define those relations and GS1 has published a list of possible link types in [GS1 Link Types] within their GS1 Digital Link standardisation effort. A system not using the GS1 Digital Link standard will have to undertake similar efforts in order to provide roles and relations that can be considered when asking a Resolver for specific information from a given DPP Data Repository. In essence, in a DPP system, we will need

standards around DPP Link-types, for example based on roles such as economic actors like consumers or recyclers.

The fact of having a resolver not only allows to filter information according to actors and roles, but it also allows to cater to interoperability needs. If an industrial DPP Data user's IT-Systems consume Asset Administration Shell information (AAS) [IEC 63278-series] and if there is a transform from the knowledge graph to AAS, requesting the URI constructed from the [Product UID] with Link-type AAS could return the DPP in an AAS-specific format, or any other format if the link type and the appropriate process are defined and implemented by the REO Resolver and the DDR..

This initial redirect function of the resolver is thus a central element for the interoperability and manageability of the DPP system. A system using DIDs as explained in [section 3.2] will use the DID document to provide the same functionality.

3.1.10.1 REO Resolver

The URI constructed from the [Product UID] will be used by the [Internet Connected Device] to issue a HTTP GET request according to [RFC9110]. For DIDs the process is different and is described in [section 3.2]. For the default behaviour concerning consumers, see [section 4.1].

As explained above, a "resolver" is a commonly used web service that receives incoming requests, formulated in the form of a URI, and then redirects the request, in the sense of [RFC9110], to the appropriate target (another URI) or targets (a list of URIs). As the DPP system is decentralized, this means that every REO can have their own resolver controlled by them. This is very close to the commercial reality where companies fear information and idea leakage that could constitute a competitive disadvantage. This is why companies want to have control over what information they share. The DPP System takes this into account by its addressing scheme in the HTTP context, but also in the DID context. By controlling the first server (i.e., the REO resolver) to respond to the request for the DPP, REOs also control how such requests will be answered. Resolvers can exist per REO, or many REOs can pool together to share the burden and reduce cost. A REO resolver is a concept that can be implemented in many ways. The central requirement is that the resolver operates under the responsibility of the REO. But a REO can outsource that service. It is expected that industry branches use their associations or representations to pool resources for the delivery of DPPs and/or DPP resolvers. This has the advantage of stabilizing the existence of the resolver against the risks of a market system, where companies can go out of business. If an industry organization hosts a resolver service for all its members, it can maintain the relevant redirection information despite one member that ceases to exist. It is further expected that industry organizations will also propose DPP hosting and backup services to their members.

The HTTP GET request can be a plain default GET request. In this case, the resolver shall send back the URI that allows to load the Consumer DPP information in HTML and CSS to allow for full mobile integration, internationalization and accessibility features of the web. By default, requesting the URI constructed from the [Product UID] with HTTP GET must return the DPP for consumers, exclusively with public data and serialized in standard HTML and CSS. Note that the architecture does not exclude that the REO Resolver directly returns that HTML/CSS information in the HTTP response without redirecting first. If there is fear of too much advertisement and misinformation, delegated acts can limit the information that can be contained in that server response. A request from a Circular Economy Operator (CEOP) may also include a specific type in the HTTP GET request. This GET request will insert

a specific type into the HTTP Link header field when issuing the request. It is expected that the system will follow [RFC8288] in this case. Upon such a request, the REO resolver will return a URI specific to that link type. This can be a file, but also a query into a [SPARQL] endpoint.

An additional very useful feature is specified in [RFC9264] and used in [GS1 Digital Link 1.1.2]. An application can request a set of links from the resolver. This will return the list of all available link types for that resolver. This way, an application can discover all the options available and a configuration or an algorithm may choose which one to use.

Resolvers can be chained. This means a resolver may redirect to another resolver in case a content, link type or information is unknown or unavailable. But that redirect can also work to help with load balancing. This way, a manufacturer also can redirect to the resolver of the supplier having delivered important components of the final product. The decentralization is thus not limited to REOs only. While REOs may remain legally responsible, they can delegate obligations from the DPP to their suppliers via that feature.

3.1.10.2 Default EU Resolver

REOs can go out of business. When they cease to exist, their infrastructure will cease to exist as well. But there is a high probability that products from that REO are still circulating in the market. And when those products are in their end-of-life phase, they will be funnelled to a Circular Economy Operator for recycling. That means in the most crucial phase, the information would not be available anymore.

The [Product UID] is still on the tangible good to-be-recycled. The construction of the URI according to a standardised method is still feasible. But a request to the URI would just end in a HTTP status 404 – Not Found. This is a general problem of the web. Content disappears and links point to nowhere. For some content, this is not problematic. But already very early, people started to think about remedies to this unfortunate fate. The socially interesting one is the [Internet-Archive]. People can point to interesting content and the archive will fetch the page and conserve it in the[Internet-Archive]. The preserved pages can be accessed via the Wayback-machine. What is interesting is that the Internet Archive can survive with donations. This hints at a very high social utility that pushes people to donate. Another approach to ensuring content continuity on the Internet is the use of so-called Digital Object Identifiers [DOI]. A unique ID is created with a certain scheme or algorithm and assigned to, mostly scientific, articles. Calling the DOI server then allows to redirect to copies of that article. There is a shift in paradigm as the same object can have one DOI but can be found in various locations under a variety of URLs. It is important to recognize this shift as the Linked data world assumes that the URIs identify an object and thus, for the Linked data world, those are different objects of the same type. The [Product UID] can act in the same way: One ID can, via some known root resolver, point to several other URLs where DPP Data can be found. The DPP system with its [Product UID] thus uses a similar concept than the DOI system, but has additional requirements set forth below.

An application receiving a "no-host" or HTTP 404 messages will need to react in different ways than a normal web browser. The [Product UID] is there and allows to construct a valid DPP URI, but the next step, the request to the resolver, is broken. If the [REO Resolver] is gone, there needs to be another service that can answer questions. This is where the EU-Resolver comes into play. There is a very close relation to the EU-Registry described in [section 3.1.5]. But the EU resolver (or root resolver) and the [EU-Registry] are conceptually two distinct services, yet they serve nearly the same purpose.

According to the [ESPR], the REO must deliver certain information into the [EU-Registry]. As already noted in section 3.1.5, the [EU-Registry] could potentially include information about the current resolver for a given [Product UID]. In case a resolver is going offline because the REO goes out of business or for any other reason, the [EU-Registry] could be informed via the same API that is used to submit the [Product UID] of the new current (backup) resolver. Where the industry is organized in branches and have their own industrywide organization, one could imagine that they provide a fallback resolver for their entire industry branch instead of using the [EU-Registry] for that task.

A DPP application though cannot know about all eventual backup resolvers. The application thus needs a way to discover where to find the relevant currently active resolver in case the REO Resolver is not responding. This is an issue very similar to the issue solved by the Domain Name System. So, if the application fails to connect to the resolver from the Data carrier information it will have to ask a default resolver that is hardcoded into the application itself. This is the EU Resolver or root resolver. This root resolver is a very efficient key-value store and just knows which resolver is responsible to respond to a given [Product UID]. As hinted already, the [EU-Registry] could potentially take that role of root resolver on top of its other tasks. The EU-Registry would then also be the EU Resolver. In this case, the application having not being able to contact the [REO Resolver], will ask the root resolver for more information. The root resolver can now return information about where to find the relevant DPP knowledge graph. Or it just redirects to another resolver that knows more about the [Product UID]. The application can now continue to work with this new resolver it discovered thanks to the root resolver. As a root resolver contains mostly the same information as the [EU-Registry], it may be efficient to combine both concepts. But they do not necessarily have to be combined.

But the information can't be fetched from a REO that ran out of business as there is no DDR anymore. The information must have been archived before the REO's infrastructure ceases to exist. The information still needs to be somewhere, hence the concept of an accessible archive. To sustain the information, some system of archiving must be included in the DPP Architecture as is further detailed in [section 3.1.13].

3.1.11 PDP – The Policy Decision Point

Policy controls and data processing constraints can be very important in a context of commercial data exchange. While consumer information is expected to be publicly available to everyone, information about the composition of materials, supply chains and other details can be highly sensitive and are expected to be only available under certain conditions.

One difficulty is that such policy controls can be implemented in a variety of ways. As the REO controls the resolver, either directly or through a delegated service arrangement, the REO can already control access to the resolver. As the resolver is, in essence, a web server doing redirects, normal web identity management can be used here. Because this is just normal web technology, things like Web Authentication and FIDO can be used as well as the entire toolchain around the eID implementing the eIDAS Regulation 2014/910EU. Even more advanced systems can use DIDs as described in [section 3.2].

When thinking about a DPP, one may be tempted to think that only the REO provides information, and that access control is limited to read access. For a pure consumer information system that may be true, but as soon as supply chains and repair come into the picture, it becomes clear that read access is not sufficient. If the DPP is more than yet another label, the repairer must be able to log his

repair into the DPP information system. This means REOs need to enrol repairers and other value chain actors into their system to allow for write access.

With the link type being able to announce the role of a requester, the system can react to that role including a specific policy per role. A recycler will need sensitive information about the product for better sorting and for the application of specific recycling methods. But REOs fear that the recycler could amass data and information to a point where this recycler can draw commercially relevant conclusions that may be detrimental to the REO's market position. This is where *usage control* comes into the picture. Because the DPP is a graph, policy information can be easily added as an annotation to the product ID. Using [ODRL], this allows to mirror the role in the permissions and rights to process sensitive information from the DPP. Such systems have been proven to work already in several industrial use cases and projects. Because the DPP system knows about roles, permissions can be tied to those roles. Beyond access control, we can now talk about usage control. Instead of complicated legal paperwork, the data carries the permissions with it.

This concept of rules that determine usage limitations for data is at the heart of dataspaces. [IDSA] defines a dataspace as data exchange plus governance. For [IDSA] this governance has certain conditions, namely security, certification and trust requirements between the parties participating in a given dataspace. As the DPP system can carry information about those requirements, a DPP system can be easily integrated in and delivered from a dataspace. An additional beneficial aspect is the concept of dataspace connectors. Those can be used to connect legacy systems to dataspaces. Conceptually, the DPP System is more than the [IDSA] dataspace Reference Architecture Model as it has several aspects concerning finding information starting from the product. But whether the DPP is served from a database or from a dataspace is a technical implementation detail. This means that all DPP Systems implemented as dataspaces are valid DPP systems.

While this is nice, conceptually, it also helps to fill the most important gap in the system: Semantics, Vocabularies and Ontologies that will be specific to certain branches. As those are also used in the dataspaces, there is a stronger incentive to create those missing components and use them for the DPP and in dataspaces exchanges in the market and between parties of a supply chain. A DPP can then be a side product of those digitized information exchanges along a value chain.

Remark: it should be noted that eIDSA was made for contracts, not for products. Applicability for Market surveillance and customs authorities needs to be verified.

3.1.12 Decentralized DPP Data Repositories (DDR)

The [Resolver] only redirects. The information about the tangible good carrying a [Product UID] is stored in some data repository. It is assumed that the system operating the data repository will be a state-of-the-art data system with backups, load balancing and other features needed to be fail-safe. Specific commitments for uptime and connectivity will be described in the Service Level Agreements (SLA). However, the notion of [DDR] is an abstract one. It describes the point in the system where the DPP information is stored. The resolver knows about those data repositories and can link to them via the redirection as described in [section 3.1.10] above and later in [section 4] on data flows.

The CIRPASS proposal for the DPP System assumes that there will be many DPP Data Repositories:

- A REO can build and maintain its own data repositories.

- It is not excluded that several REOs jointly operate a data repository or have a data repository by branch.
- A REO can delegate its responsibility to a DPP service provider which will operate a data repository on its behalf.

There is no hindering on concentration as there is no technical obstacle to distribute the repositories even more. It is expected that the operation of a data repository will follow operational structures in REOs or branch-associations of the REO's industry.

3.1.12.1 An interoperability layer built using linked data

The CIRPASS proposal for the DPP system includes a conceptual interoperability layer that comes in the form of a data graph built using linked data. This interoperability layer is not limited to the DPP data itself, but also encompasses metadata like access control, usage control and other commercially important constraints**.** This allows DPP data to be easily integrated into value chains. It also allows DPP data to integrate well with track and trace solutions. Constructing the DPP as a knowledge graph has many advantages, as discussed in [section 2.2.3]. The knowledge graph can put data points in relation to each other. The semantic information allows for much better analytics and a higher level of interoperability.

While there are a wide range of tools, commercial ones and open-source, available to deal with knowledge graphs, the interoperability layer described in this document is implementable without a strict limitation to those tools. If data is already expressed using a graph representation, this means on the one hand that this available tooling can be used without bigger transformations. On the other hand, if data is not already expressed using a graph representation, transforms will be necessary to expose data from the Decentralized DPP Data Repository as RDF models. The current technology was made for data integration and serves that purpose well. This means that these transforms can be easily implemented on top of the existing IT landscape of a given enterprise.

Looking into dictionaries from the relevant industry branch can be helpful. There are for example concept dictionaries based on [IEC 61360] which can be used to clarify the semantic meaning of sub-models and sub-model elements.

3.1.12.2 Knowledge Graphs – A very short introduction

There are many definitions of the term Knowledge Graph. [KGBook] defines it as: "*a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities*." The graph is constructed when applying a graph abstraction to data resulting in an initial data graph.

There are several ways to create a graph abstraction to the DPP data. The CIRPASS proposal for the DPP system described in this document uses directed edge-labelled graphs ([KGBook], Chapter 2). With the Resource Description Framework ([RDF]), these models benefit from a high number of standard specifications and a very advanced degree of syntactic and semantic interoperability. The other well-known model is called property graph, supported by popular software. The W3C RDF-star Working Group is currently tasked to overcome the dichotomy and to allow for a data format that can represent both models, directed edge-labelled graphs and property graphs, in an interoperable way. [RDF 1.2] will help to overcome a perceived dichotomy. Thus, the DPP system could be easily extended to allow the use of property graphs while remaining compatible. It has to be noted again, that this

only concerns the interoperability layer of the system. The DPP system does not prescribe the use of a specific tool or database. But the reduction of transforms is encouraged in an overall assessment of the ecological implementation cost.

3.1.12.3 Implementation Considerations

The data repository is more of a concept than it is the requirement for a specific data base. In fact, nothing excludes that product information could be served from a company's ERP system on the fly and transformed into a format that is corresponding to the interoperable formats that can be easily consumed by consumers or some Circular Economy Operator.

As the DPP is conceptually a knowledge graph, it may be easier and more promising to use so-called triple-stores to store the DPP information if it is natively expressed as RDF data. But it is not excluded to use ERP systems, normal relational databases (e.g., SQL) or other tools to store the DPP information, provided it can be transformed into the interoperable data format (RDF) that is usable within the circular economy by Circular Economy Operators. In this context, the notion of Dataspace appears. A dataspace connector in the sense of the [IDSA] framework could be used to connect arbitrary legacy systems to a DPP Dataspace.

It has to be noted though that every transformation of data from one format to another introduces risks for data quality, is costly in terms of calculation and thus in terms of energy. Too many transforms thus counter the very goal of the DPP to help create a more environmentally friendly economy. While transformation from one standardised format to another one is burdensome, but possible, it is not desirable. The more people use the same data format and the same semantics, the less transformation is needed. In order to facilitate this some social agreements and standards around data formats will be needed.

In order to supply or retrieve information, the [DDR] has to be queried. In the architecture, the [DDR] is not proscribing a specific software. The architecture is general enough to allow for more than one option. The choice of D3.2 is to have a REST-API (ideally described using OpenAPI) that connects to the DDR. This allows to have specific declarative requests that are in the API transformed into preformulated [SPARQL] queries that retrieve or update the DPP Data in the DDR. Which in turn fits well with the [RFC8288] Link-type submission to the Resolver. In this case, a specific link type in the HTTP request to the Resolver can return a URI to the REST API described. This has two advantages:

1. The REST API can transform requests to preformulated SPARQL queries that retrieve or update the data.
2. If actors interact with the triplestore through the REST API, this can be the place for transforms needed for interoperability
3. A REO can now use their own preferred database given the appropriate transform is done in the API implementation.

In case of privileged access, the requests to the endpoints are submitted with the corresponding credentials (e.g., a JWT token [RFC7519]) when needed, and the REST services are responsible for transforming the request to the appropriate [SPARQL] query. Endpoints that accept adhoc [SPARQL] queries for actors with the appropriate access rights (e.g., REO) can also be made available. The REO has to update the [REO Resolver] with the Typed Links created for this specific [Product UID], and the [Default EU-Resolver] with the backup Typed Links.

3.1.13 Archives

3.1.13.1 The Need for Archives

It should be reminded that the goal of a DPP is to further the circular economy. A DPP is not only a consumer information system or a control tool for the administration. This means, DPP information will be especially precious once the tangible good carrying the [Product UID] reaches its end of life. In our economic system, at this point in time, there is a non-negligeable risk that the legal entity initially responsible for the DPP has ceased to exist or has gone out of business. There is no initial REO anymore. But a recycler, refurbisher or remanufacturer still needs DPP information more than ever for products of this REO that are still on the market. This is a known issue for large scale decentralized systems, like the web. For the web, the [Internet-Archive] provides access to information once found on the web, where the websites have been discontinued. If this is possible for the Web with its billions of pages, this can be seen as a proof that an archive for the DPP data is also doable. While the [Internet-Archive] is a US non-profit association that is financed by donations, providing the DPP Archive can either be a public service financed by taxes. Or it could be a cost of business that could be organized by branches and their association like an insurance model where the community covers this social cost via contributions from the members of the branch. In this case, the fact that all businesses of a branch contribute will also ensure that the provision of DPP information is more sustainable and not dependent on the survival of one single company or even a dedicated archival company.

3.1.13.2 Archiving and Backup

As described until here, the DPP System follows an Internet approach. This means the usual practices on the Internet and the Web apply. This means professional hosting provider guarantee a minimum online availability of the system and normally also include some failsafe backup solution in case things go wrong, including a professional backup system in a separate location. But those systems normally do not come for free. As long as there is a legal entity in business to pay for the services of those hosting providers or as long as some legal entity provides their own service in a failsafe way, there is no need for a second structure providing essentially the same service.

On the other hand, there is a need for an archive of DPP data as already argued, this archive does not necessarily implement an exact copy of the initial DPP system implemented by the REO. The Archive must provide data that is otherwise not available anymore. Following again the example of the [Internet-Archive] the archive does not need to react with the same latency. Toning down the requirements for responsiveness will make the Archives cheaper. The DPP System as suggested also has a tremendous advantage compared to the normal Internet archiving. Because there is an [EU Registry] that feeds its knowledge into an EU Root resolver, the system can have a very smooth reaction in case the initial DPP is not available anymore. In this case, a DPP Application will scan the [Product UID] and issue a GET request as shown in [section 4.1]. Because there is no REO anymore, there is no REO-Resolver, and the network layer will return a "host not found" response. The application now can turn to a root resolver who knows itself where to find the archive or who knows another resolver that has information about the archive. Now the application addresses this resolver, and that resolver will return a URI that serves to access the information. All it needs is a root resolver that will have very little traffic. This can be compared to the ROOT DNS servers that have a similar role in the DNS system. And this resolving system allows the archives to be as distributed as the DPP system, provided there is a root DPP resolver that is called [Default EU-Resolver] in this document.

3.1.13.3 Long term archives

For longer term archives, DPP data may be stripped again to the strictly necessary. This can take the form of an aggregation or a limitation to certain data points. But given today's capacities for storage, it may even be more efficient to store things just the way they are in some long-term archival systems. Those long-term archival systems typically trade the efficiency of storage against a higher burden accessing the data. The long-term archive is thus not expected to generate much traffic and only provides information once it can't be found anywhere else. This role of service of last resort can be enshrined into the protocols used for resolving as described in [section 3.1.10].

3.2 Structure of the DPP System using DIDs

As shown in [section 3.1], the DPP System can be implemented short term with the HTTP approach as all technical preconditions for a quick implementation are available but need to be adapted. The HTTP approach uses URIs. Those rely on the Internet Domain Name System DNS. But a domain name can be lost, and the new domain owner can now redefine all identifiers made under that domain name and point to different information. There are several ways to counter this risk. There is the parallel registration of the number. GTINs are not only expressed as URIs, but they are also registered as numbers, so there can be a second lookup system that allows to make sure that the loss of a domain name is not making all identifiers invalid. But there is new forward-looking technology that allows a variety of methods to secure decentralized identifiers. A REO could still create and manage their own identifiers, but those can be secured in many ways. One current way is to use blockchain technology. But there are more ways than only the use of blockchain. This is why [Gaia-X] is using this way to create identifiers. Not everything is ready at a production readiness level. This is why it is a view into the future on how a DPP system could work.

Many of the new initiatives in the EU's Data Strategy, like [Gaia-X] suggest using Distributed Identifiers [DID]s. The keyword is self-sovereignty for the identity management. This means that everyone can create their own identity in their own sphere while this identity can be used by all others to address the creators of those self-sovereign identities. This nicely distributes the responsibility for the very important identity management system that is needed if the DPP data is not read-only but allows actors like repairers to add information to the DPP. As [DID] systems are very present in the DPP discussion, this document shows how to create the DPP System with all of its necessary elements, services and properties using [DID]s.

Below we describe the [DID]-based approach, commenting on the differentiating parts to the HTTP-based architecture for DPPs. [DID]s along with the deployment of Verifiable Credentials [VC]s can showcase more advanced features that a DPP system could implement to further the digitization of the industry towards major improvements in decentralization, efficiency, and verification. For example, it can be noted that the HTTP-based DPP system requires the transformation of [Product UIDs] to resolvable URIs and Typed Links using centralized resolvers, while in the [DID] case the Typed Links are offered in a decentralized manner. Additionally, the [DID]s along with the deployment of [VC]s offer the ability to cryptographically verify ownership of the identifier and access control rights to the various privileged users inherently.

[DID]s represents a shift from traditional, centralized models of identity management. Most of the [DID] Systems embrace blockchain technology as their main verification method, but there are also

ways to implement the [DID] system without any blockchain involved. However, in this case one loses the immutability features of that blockchain technology. So instead of relying to a central authority for issuing, storing and validating a digital identity, any individual, company, or organization can own, control and have the responsibility of the digital identities of their own and of their products, proving the ownership of the [DID] and offering trustworthy communication services. Key benefits of [DID]s include:

- **Self-Sovereignty and Privacy**: Owners of [DID]s are in control of their [DID]s, and what data they share with whom;
- **Security and Availability**: Distributed Ledgers like blockchains employ encryption, immutable data records, and decentralization, avoiding central point of failures and data breaches;
- **Portability and Interoperability**: [DID]s are portable and interoperable across various contexts and use cases;
- **Cost Savings**: Reduce administration cost by eliminating duplication and manual verification.

[DID]s are URIs that offer a verifiable decentralized digital identify that is decoupled from centralized authorities, leaving the sovereignty of the ID to the controller of the [DID]. Specifically, [DID]s allow for the decentralization, persistence, global resolvability, and cryptographic verifiability of identifiers. [DID]s can refer to various subjects, including organizations, businesses and products. The controller of the [DID] can verify the control of this specific [DID] without requiring any other intermediate service or authority, due to its decentralized nature, allowing trustable interactions in untrusted networks.

[DID]s are associated with a [DID Document] that contains the information associated with this [DID], its control verification, and the provided service endpoints. This [DID Document] is accessible through a specific [DID] method that is included in the URI of the [DID]. The method is associated with this specific [DID] for creating, reading, updating and deactivating operations over a Verifiable Data Registry (VDR). A VDR is essentially the place where the [DID Document]s are stored.

Figure 2 depicts the structural view of the architecture. Below we provide more details regarding the components of the structural view of the architecture and discuss the differences with the HTTP URIs approach described in [section 3.1]. Two major differences with the HTTP approach are that the resolution of a [DID] is done by the [DID] approach itself, and the inherent support of authentication and authorization mechanisms using [DID]s and [VC]s. The data flows of the [DID] approach and its advanced features that utilize [VC]s are described in [section 4.2] and [section 4.3] respectively.

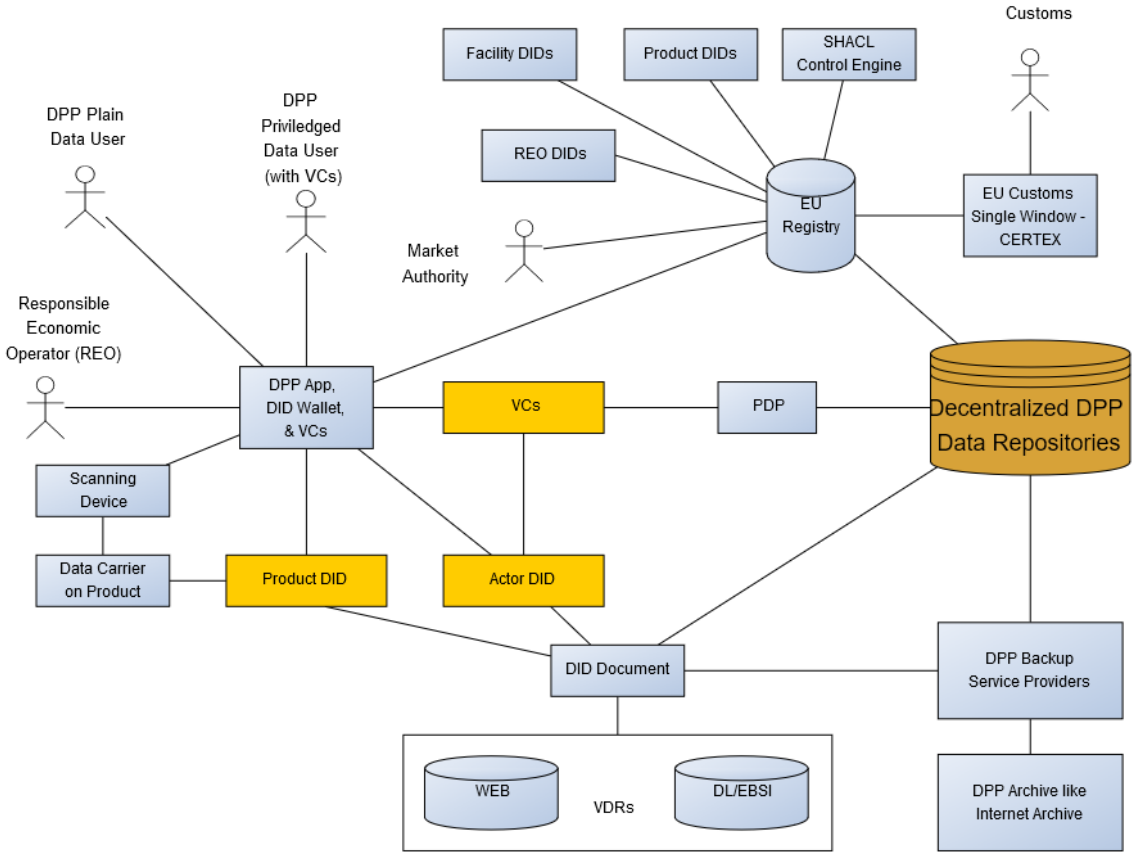

Figure 3. Structural view of the DPP system showing the structure, actors and components of the system without showing the data flows for the DID architecture.

Some parts of the [DID] System are already standardized, including the basic functionality. However, the space is still evolving. For example, there are numerous [DID] methods under implementation or specification with vastly different functionalities and properties. As a result, an implementor of the proposed architecture will have to get an overview of the available [DID] methods to select the most appropriate ones (e.g., [Fdhila2021], [Hoops2023]), across the evaluation criteria set in the [DID-Rubric]. Furthermore, notice that currently there is no large-scale deployment of [DID]s/[VC]s.

Currently cameras on consumer mobile phones don't know yet about DIDs. All operations with DIDs therefore require the use of a dedicated DPP App that has access to the camera and handles the DIDs resulting from the scanning of the Data Carrier.

3.2.1 Decentralized IDs (DIDs)

Decentralized identifiers [DID]s are globally unique[7] persistent identifiers that offer a decentralized digital identity in the form of a URI, for any kind of entities like people, organizations, and products.

[7] DID uniqueness comes from algorithms capable of generating globally unambiguous identifiers producing random strings of characters. For example, in the case of ethr DID method, that uses Ethereum blockchain addresses

DIDs are decoupled from centralized authority and registry, and the controller of the DID is able to prove that it is the real controller of the DID without the need for any other intermediate party. DIDs are URIs associating DID subjects with a [DID Document] that holds the verification methods and service endpoints (e.g., data retrieval) for this specific DID. Usually, the controller of the DID is the DID subject, but it can potentially be any other party that can act on behalf of the DID subject.

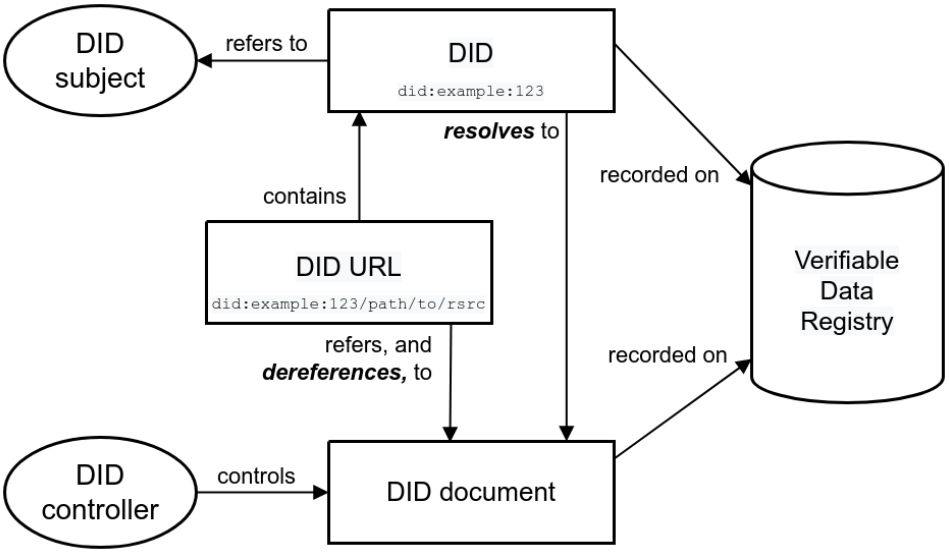

Figure 4. An overview and the relationship of the basic components of DIDs

Each DID URI starts with the DID scheme and is followed by the DID method (discussed below) and the corresponding identifier which depends on the deployed DID method. Since the DIDs are valid URIs, they can also include the path, query, and fragment parts of URIs (based on RFC 3986). Of particular interest for the DPPs is the service parameter that selects the appropriate service for fetching the DPP data. In that case a DID URL is dereferenced to a service endpoint. Below, is a valid DID URL. "did" is the *scheme*, "example" is the *DID method* used for the resolution of the DID and the management of the corresponding [DID Document], "123456789abcdefghi" is the *DID method-specific ID*. The query part "service=dpp" is the *service ID*, that describes the default endpoint that is responsible for dereferencing the DPP in the [DID Document]. A backup service pointing to a backup endpoint can also be provided in the parameters of the DID.

did:example:123456789abcdefghi?service=dpp

The DID method is the mechanism by which a particular DID is resolved and leads to the [DID Document] that contains information associated with a DID (see below for a description of a DID

or the btcr DID method that uses bitcoin blockchain addresses, the probability of a collision occurrence is 1 in 2^160. In the case of the web DID method the uniqueness from the URI. Persistence is guaranteed through the use of DLTs, even if a DID is deactivated or the REO is out-of-business.

document). There are numerous DID methods[8] (currently over one hundred) that use various Verifiable Data Registries (VDRs) implemented using numerous technologies with different properties regarding security, performance and privacy aspects and trade-offs. A VDR is essentially the place where the [DID Document]s are stored. Such technologies include Distributed Ledger Technologies (DLT)s like blockchains, which are immutable and decentralized (examples include the Bitcoin, the Ethereum, and the European Blockchain Services Infrastructure (EBSI) blockchains), Distributed Hash Tables (DHT) like IPFS, or even plain web domains, which however are not as resilient. For example, the Web DID method assumes a trusted environment that resolves to the web host that the domain name described by the DID resolves to using the Domain Name System (DNS) (e.g., did:web:dpp.eu:ProductBranch:ProductUUID). However, in case the web service is not online, the corresponding [DID Document] will not be accessible. This is an important reason why DIDs are usually registered using DLTs that offer a non-centralized and non-trusted environment, that can also benefit from the immutable, decentralised and tamper-free nature of blockchains.

In any case, all entities in the DID ecosystem, trust the VDR to be tamper-evident and the valid record of which data are controlled by which entities. For DIDs to be resolvable they need an application that knows how to access the [DID Document] associated with each DID using the DID method present in the DID URI. The interaction with the VDR, including the DID resolution process based on the DID method, can be provided through the DPP App. Another option is to decouple the client from interacting with the VDR, a role that could be provided by various external services, like the Universal DID Resolver. In our architecture we assume the deployment of web and mobile apps with integrated wallets that know how to resolve a DID based on the method of the DID imprinted on the [Data Carrier]. If there is a need for a generic app (e.g., Google Lens app or a camera app on a mobile) that is currently unaware of how to resolve DIDs, the [Data Carrier] could support two URIs, a DID one and a HTTP one. The HTTP URI will lead to an authoritative service supporting the resolution of the corresponding DID, encoded in the HTTP URI, that will be responsible for retrieving the associated [DID Document], identify the public service endpoint and retrieve an HTML page of the corresponding DPP. Application-level protocols that allow a secure communication on the DIDs are already available (e.g., [DIDComm]).

As already noted, many of the available methods are not implemented, so any DID-based implementation of the proposed DPP architecture should consider a state-of-the-art comparison of the DID methods (e.g., [Fdhila2021], [Hoops2023]) along the evaluation criteria set in the [DID-Rubric]. In the proposed architecture we assume that the deployed DID method supports [DID Document]s with *full key management* and *service descriptions* and that the resolution of DIDs is *permissionless* and can be done by anyone.

In our design we assume two kinds of DIDs. The first one refers to the various actors that participate in the DPP environment (not including plain users) and the second one to the products.

3.2.1.1 Actor DID

Each actor of the DPP environment should be associated with a DID. This is essential for all participants ([REO]s, [CEOP]s, and [Public Authorities]) except [Plain Consumers]. For example, [REO]s should have

[8] For a list of available DID methods check https://decentralized-id.com/web-standards/w3c/decentralized-identifier/did-methods/

a DID to mint product DIDs that are controlled by them. In addition, Actor DIDs are important for letting the various services authenticate that any requests on behalf of an Actor DID indeed originate by the actors themselves, without the need of any central authority. This can be done through the use of a handshake and the public keys and verification methods described in their [DID Document]s. Actor DIDs are also a prerequisite for privileged users that can have access to services that need authorization. In this case, the owner of the service using the corresponding Actor-DID can provide Verifiable Credentials ([VC]s) to the DIDs of the interested parties and sign them, allowing those Actor DIDs to have access to their services after authentication and presentation of the corresponding [VC]s. In addition, Actor DIDs allow third parties to provide verifiable information in the DPPs. More details regarding VCs are in [section 4.3]. Plain users (i.e., consumers) are not required to control a DID. Actor DIDs should deploy the same DID method for consistency reasons, satisfying the needs of the EU regulators. REO's DIDs can be stored in the [EU-Registry] as unique operator identifiers. Similarly, it is very easy for any [REO]s to additionally mint or create their own [Facility ID] as required for the [EU-Registry]. Additionally, this [Facility ID] can be verified in the DID scenario and can thus participate in track & trace schemes.

3.2.1.2 Product DID

Regarding Product DIDs we can follow two approaches. The first one allows a REO to mint DIDs that are controlled by the [REO DID]. In that case the deployed method holds the [DID Document] of the corresponding product. Each REO could deploy its own DID method. However, the preference is towards decentralized and untrusted approaches. The EU could also propose specific DID method(s) that fulfil the requirements and specifications needed for supporting the European market. Another approach is to take advantage of the [REO DID] for the Product DID, by augmenting the [REO's DID] URI with a query part that will hold a UUID for each specific product (e.g., did:method:REO-DID?UUID=9c5b94b1-35ad-49bb-b118-8e8fc24abf8). Resolving the [REO DID] will provide the associated REO DID Document with the corresponding querying service endpoints, where the UUID of the product can be used for retrieving the relevant information. In this case there is no need to explicitly create a DID for each product, reducing the storage and communication cost. It also reduces even further the probability of a Product DID collision, since UUIDs are associated with a specific [REO DID]. However, it needs the deployment of a dedicated app that knows how to use the UUID in the query part of the DID for augmenting the service endpoints provided in the [DID Document].

3.2.2 DID Document

A DID Document can express cryptographic material, verification methods, and services (e.g., querying endpoints) that allow the [DID] controller to prove control of the [DID]. Among others, it offers the service of the resolvers of the HTTP architecture. The provided services enable trusted interactions associated with the [DID] subject. Usually, they are represented using the [JSON-LD] format. Some properties of the DID Document include the [DID] subject associated with this DID Document, the controller of the DID Document (e.g., in the case of [Product DID] the [REO's DID]), the verification methods (including cryptographic public keys for authentication and authorized interactions with the [DID] subject), and the services for advertising how an interested party can communicate with the [DID] subject or the associated entities, including authentication, authorization, discovery and interaction (i.e., querying) services end-points. The DID Document can be updated whenever needed (e.g., update the services endpoints) by the corresponding [DID] controller (e.g., the REO's [DID]). The

deployed [DID] method should support full key management and service descriptions for the DID Documents.

3.2.3 Verifiable Credentials (VCs)

Verifiable Credentials [VC]s are cryptographically secured identity attributes and assertions about a specific subject issued by an issuing authority. They represent all the information related with identifying the subject of the credential, the issuing authority, the type of the credential, asserted properties by the issuing authority about the subject, evidence about how the credentials were derived and any constraints (e.g., expiration date). The validity of the VCs can be verified by any third-party without the need to interact with the issuer.

VCs can be issued from and for Actors with a [DID] (i.e., [Plain Consumers] can't have VCs issued to them as they are not identified). In the DID-enabled DPP architecture, VCs are either issued by the [REO]s or by a trustworthy public organization to any actors that need to have privileged access to the provided DPP Data Repositories. The VC issuing service endpoint should be available in the corresponding [DID Document] of the issuer. The subject and the holder of the VC is usually the same (i.e., the [DID] of the privileged actor), although a holder could potentially transfer a VC to another holder. The issuer can also forbid transferring the VCs. Actors ask for VCs from [REO]s or trusted public organization using a corresponding service documented in their corresponding [DID Document]. Each actor is responsible for storing its corresponding VCs in its own VC wallet (see below). In addition, third parties can supply any verifiable information related to the DPP in the form of VCs (e.g., GHG emissions and test reports) that can be integrated to the DPP or associated with DPP information. VCs can expire and actors can ask for a VC reissuing. Privileged actors must provide the VC to be granted privileged access (e.g., update operations, access to sensitive data, etc.).

VCs are provided in the form of verifiable presentations, which is the recommended format for sharing the verifiable credentials. Verifiers always have to send a unique challenge and domain when requesting a verifiable presentation, which is then incorporated in the proof section of the presentation during the signing process to avoid replay attacks. Usually, VCs are represented using [JSON-LD]. Regarding the invalidation of VCs various approaches can be deployed (like the ones described for EBSI). For example, VCs could be short-lived and reissued or the status information could be obtained from the trusted issuer.

The data flows regarding the VCs are described in [section 4.3].

3.2.4 DPP Apps

DPP apps are mobile- or web-based apps that support the DID-enabled DPP ecosystem described here and require Internet access. We identify two different kinds of apps, one focusing on the needs of the REO and one for the rest actors. These apps support the storage and interaction with the [DID]s and the supported VDRs through the corresponding [DID] methods to retrieve/update [DID Document]s, the interaction with the provided REO endpoints including the submission of [SPARQL] queries and [SHACL] templates, and the management and storage of [VC]s. At their core lies a [DID] wallet, responsible for issuing [DID]s, holding the private keys, interacting with the VDRs and implementing the communication protocols. In addition, it also implements a [VC] wallet, that holds all [VC]s issued to this specific holder as well as functionalities for issuing and requesting [VC]s. Finally, these apps

offer functionality for scanning [Data Carrier]s of a [Product DID] and the input of sets of [Product DID]s (for bulk querying).

3.2.4.1 DPP minting App, DID & VC Issuer Wallet (REO-App)

This app is responsible for holding the [DID] related data of the REO and the minting of [Product DID]s and [DID Document]s using the corresponding [DID] methods of the supported VDRs, which will then be imprinted in the data carrier. In addition, it can receive and review requests from other [Actor DID]s that ask for privileged access to the REO's DPP [DDR] and can issue the corresponding VCs. The [VC] issuing endpoint should be available in the corresponding service endpoint in the Actor's [DID Document]. This app should also be used from any other actor that mints [DID]s (e.g., Remanufacturer).

3.2.4.2 DPP App, DID & VC Issuer Wallet

This app is a restricted version of the previous app, responsible for holding the [DID] related data of the actors/products and the associated [DID Document]s using the corresponding [DID] methods of the supported VDRs. In addition, it can request VCs from REOs and can hold all VCs issued by various REOs to this actor for privileged access to specific DPP [DDR]s. Finally, it is responsible for the interaction with the endpoints of the REO's DPP [DDR], for any [Product DID] that was scanned from the data-carrier or was given as input in a set of DIDs, the inclusion of the corresponding [VC]s (if they exist) to the HTTP requests, and finally the rendering of the retrieved data. For third parties like Recyclers that can potentially provide verifiable information to a REO's DPPs, the wallet can issue [VC]s of this actor to the REO for inclusion in the DPP so that any DPP consumer can verify the provided data.

4 DPP System Data Flows

In this section we describe the core DPP data flows for both the HTTP and DID architectures. In addition, we describe advanced DPP data flows for the DID architecture that use the Verifiable Credentials (VCs) for authentication/authorization and verification of DPP data.

4.1 HTTP Data Flows

The information flows in the HTTP are not very constraint. They follow normal web technologies. But the concept presented here remains high level. A DPP System can be implemented in a variety of ways while remaining interoperable to a certain extend. As already mentioned in [section 3.1.9], the DPP system can be implemented as a [GS1 Digital Link System] or be based on the principles of the IEC 61406-x [IEC 61406-1, IEC 61406-2] for example. But if additional constraints are matched, it can also be seen as an [IDSA] dataspace. Or someone can just make up a new system with the web components of their choice.

Below we describe the HTTP data flows regarding the initiation of a DPP from a REO and its use from either a plain user or a privileged actor like a Recycler, Repairer, Remanufacturer, or a Market Authority. The data flows are provided in the form of Data Flow Diagrams (DFDs).

4.1.1 Creating a DPP

Initiating a DPP consists of three main processes: a) the creation of the product ID by the REO and the attachment of the Data Carrier to the product, b) the compilation of the DPP information with data gathered from the REO or any other third-party suppliers and its storage to the DPP Data Repository, and finally, c) registering any relevant information with the authorities.

4.1.1.1 Minting a Product UID

Before placing a product in the market, the REO must create a [Product UID]. This can also be named "minting". The requirements of this [Product UID] have been described in [Deliverable D3.3]. In our architecture we do not assume a specific kind of a UID. The only assumption is that the UID can be mapped to a URI that will point to the REO's resolver which will return where the DPP data for this product will be available. After creating the [Product UID] the REO must produce a Data Carrier that will hold this UID and attach the Data Carrier to the product. The Data Carrier can be a QR, an RFID or other Data Carrier.

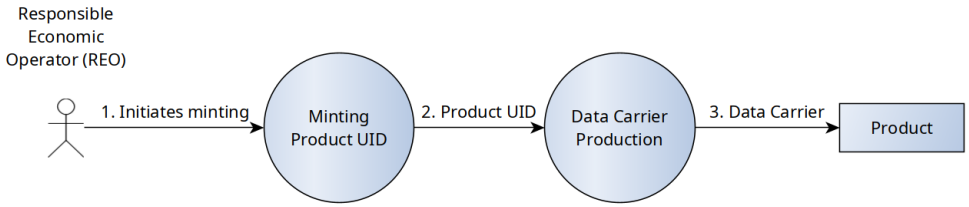

Figure 5. HTTP DFD – Minting a Product UID

4.1.1.2 Assembling and Issuing the DPP Data

Each [Product UID] is associated with the DPP data for this specific product. If there is only model level granularity in the system, the [Product UID] is associated with model level information. The REO is responsible for gathering all the data, cleaning, normalizing and transforming them to the required format and vocabulary to assemble the Knowledge Graph (DPP KG), whatever that format will be. The DPP KG will be stored in the DPP [DDR]. This document assumes that the DPP KG is using Linked data serialized in the [JSON-LD] format. But this is a design choice that isn't compulsory. Other choices would not break the system.

The source of the data can come from REO's information systems (e.g., ERP) and/or third-party systems (e.g., suppliers ERPs). At industrial scale, it is clear that it will not be possible to fill all the DPP Data into a form by hand that is then registered with the DDR. Instead, we assume that the DPP Data will be assembled by a fusion of data from a variety of existing data sources. This includes ERP systems, but also documentation systems and production systems. The more the DPP can be created via easy data re-use from internal and external sources, the more compelling the DPP system will be. The Linked data paradigm explained in [section 2.2.3] helps in that respect as the technology platform has developed many bridges to legacy data over time. Nearly everything can be transformed into Linked data. Once transformed into Linked data, this greatly facilitates the merging of data from a multitude of sources and systems.

In the DPP system architecture, the REO initiates the creation of a [Product UID], transforms the UID into a [Data Carrier], and fixes that carrier onto the tangible good. Data from various sources is merged into the DPP Data KG. Because the DPP data should adhere to the [ESPR] regulation for setting ecodesign requirements based on the sustainability and circularity aspects, it must be compliant. To

check this, the DPP Data KG is matched against the corresponding [SHACL] template, hopefully provided by the Public Authorities. This validation check is performed at minima during the registration process into the EU registry (see below). Finally, the DPP Data KG is stored in the [DDR].

Some of the DPP data must be made available publicly and an appropriate access needs to be provided. As described, some data is only accessible to privileged users. For those with a specific role appropriate access control rights have to be set in the [DDR] or in a [PDP] that manages access to the [DDR]. The access to the various resources and services of the DPP can be described through [ODRL] policies within the [DDR]. But there are other implementations possible with a very classical [PDP], e.g., using [XACML] and [SAML]. Conceptionally, the [PDP] is responsible for evaluating the access requests against the authorization policies either within the DDR or as a separate service. Finally, the REO has to notify the [EU Registry] with the data points required by Art. 12 [ESPR], which includes the [Product UID], [REO ID] and [Facility ID]. More details regarding registering with the authorities are provided in the below DFD and in the next section.

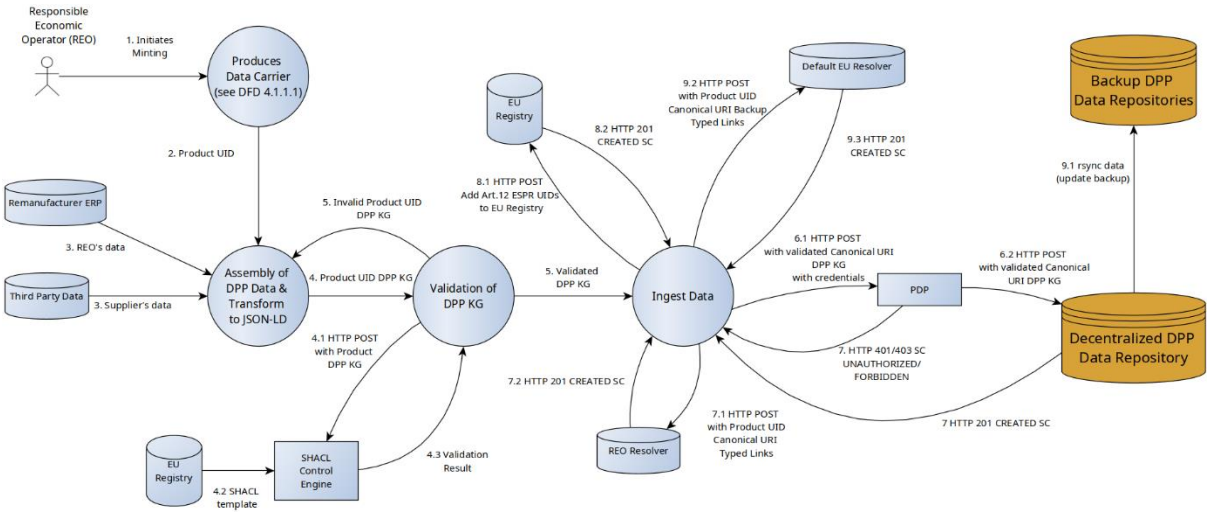

Figure 6. HTTP DFD – Assembling and Issuing the DPP data

4.1.1.3 Registering the DPP with Authorities

After the REO has inserted the DPP KG data for the corresponding [Product UID] into the [DDR] it sends the data points required by the [ESPR] to the API of the [EU Registry]. Once the Authorities have the [Product UID], they can request the DPP KG data from the [DDR]. For discovering the service endpoint, any Authority wanting to access DPP information must retrieve the DPP service endpoint through the [REO Resolver], as discussed in the Role-based DFD for the Market Authorities described later (see section 4.1.2.5). A [Market Authority] wanting to access non-public data may need privileged access to the repository and must provide the corresponding credentials to the REO to get access to that information. The Authority might optionally want to validate the DPP using the [SHACL] template for this specific product or branch. If the validation fails, the Authority validating can optionally store the status regarding this specific REO and DPP in their own system, or elsewhere and recommend to the DPP Registration Authority to revoke the registration. It is not excluded that more information can flow into the systems held by Authorities beyond the strict limit of the data points required by Art. 12 [ESPR], namely the [Product UID], [REO ID] and [Facility ID]. This is an implementation question. The last step for the REO is to update the [Default EU Resolver] with the backup Typed Links created for this specific [Product UID].

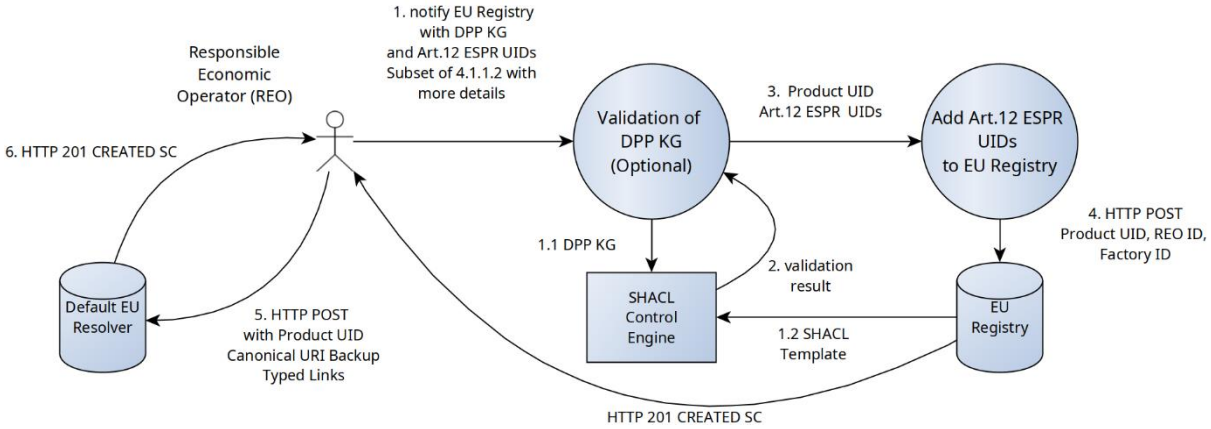

Figure 7. HTTP DFD – Registering the DPP with Authorities

4.1.2 Using a DPP

Once the DPP is created as described in [section 4.1.1], the DPP is ready for use. As the DPP system architecture is based on Internet and web technology, there are many ways to access the DPP Data KG. And there are as many reasons to access that data. The very reason to make this architecture is a situation where a tangible product is evaluated in presence and there is a need to access DPP Data from an identifier on the tangible product. We call this the product-centric view. This view itself has many colours, including consumer views, recyclers, repairers. It is not so obvious that, because of the use of web technologies, we can also have an information-centric view. This way, the actor does not start with the [Product UID] and the [Data Carrier], but they start by exploring the [EU Registry] or the [DDR]. All those other uses are as legitimate as the initially intended use for the circular economy. This means the CIRPASS proposal for the DPP system architecture contributes to the circular economy, but also to the digitization of the industry.

In [section 3], the structure, actors and components were presented. [Section 4] shows how they interact. On the DPP Data user side, the many possible roles and interests are condensed into conceptual roles. Those roles are then played through with a focus on the data flows between the entities given by [section 3]. The results are data flow diagrams (DFDs) that show how certain tasks can be accomplished within the architecture. The concrete implementation into software of those tasks can take a variety of ways without losing interoperability.

On the DPP Data user side, we showcase the transformation of the [Data Carrier] attached to the product to a usable URI which is then dereferenced to get the corresponding public DPP information. We also showcase the data flows of other privileged actors (e.g., Recycler, Repairer, Remanufacturer, and Market Authority) that can get more refined information for a Product DPP or can possibly update the DPP information.

4.1.2.1 From Data Carrier to a Usable URI

This data flow transforms any [Product UID] assigned to a [Data Carrier] attached to a product into a resolvable Link. The DPP Data user uses an [ICD] and through its scanning device scans the [Data Carrier] attached to a Product. The scanning device can be the camera of a mobile device, or any other scanner device connected to the [ICD]. The scanner device delivers a [Product UID], either encoded in a resolvable link or as a simple Number, like in Barcodes. In case of a Barcode that is not an URI, additional transforms are needed. The reasons to integrate that option are detailed in [section 3.1.6].

Those transforms can be provided as a part of the DPP user application, but they can also be implemented as an online service, where the number is submitted, and a URI is returned. In fact, even if the number of the [Product UID] is only locally unique, this mechanism allows to de-Silo the number and make it globally unique. If combined with making the number a URI, we speak about semantification. It is a decisive advantage if such transforms are standardised or have at least an open technical specification attached to them. An example for such a standard can be found in [GS1 Digital Link 1.4.1] that specifies how to transform flat GTINS into URIs. The relevant principles and restrictions link the data are also specified in IEC 61406-x [IEC 61406-1, IEC 61406-2].

Now the application has a URI and can proceed with the normal procedure further detailed below. The default response is to return a link that provides access to the public DPP data. Privileged actors have to provide more information regarding the kind of Typed Link they want to retrieve or can select the appropriate Typed Link from the Resolver's response to a HEAD request as shown in the privileged actors DFDs later on.

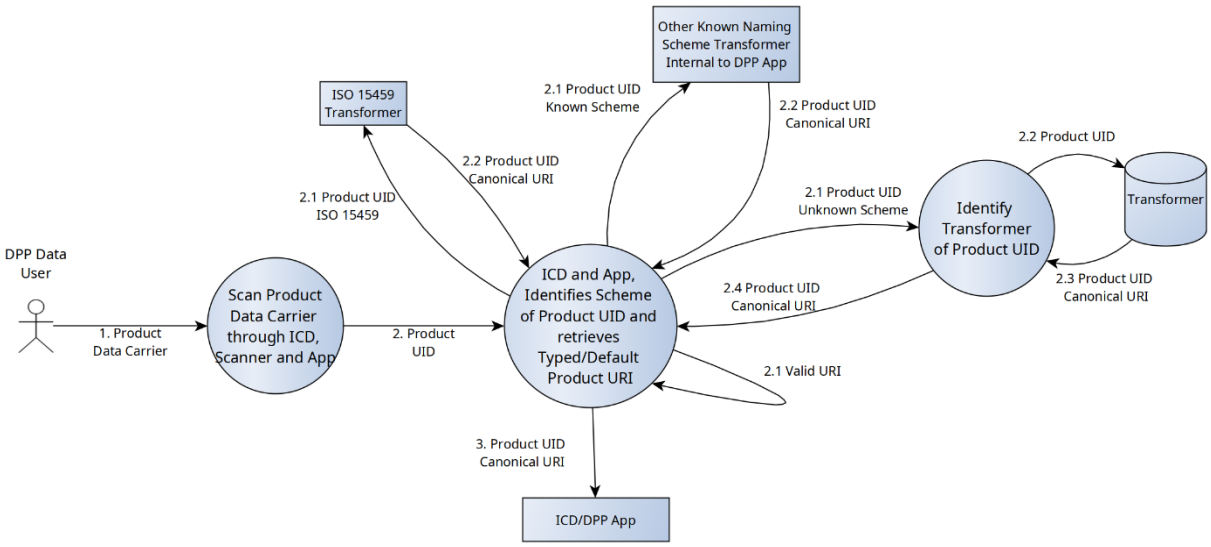

Figure 8. HTTP DFD – Using a DPP – From Data Carrier to a Usable URI

4.1.2.2 The Default (Consumer) Data Flow

Here we assume a default DPP Data User (i.e., consumer). Default DPP Data Users are defined by the fact that they send a HTTP GET request according to [RFC9110] without any Link type information to the resolvable URI of the [Product UID] or constructed therefrom. A REST-API can transform this GET request to a SPARQL query that retrieves all the public information of a DPP from the triplestore and returns a plain HTML/CSS web page. No access control functionality is required. The big advantage is that this scenario works out of the box with every Smartphone camera application. In this case, the Smartphone's camera application serves as a scanning device, deciphering the URI from a QR code and handing the URI to the smartphone's browser to do the GET request. The system will return a web page that can be displayed in the smartphone's browser.

If the HTTP status code of the response is a 404 according to [RFC9110], meaning that the DPP is not available anymore from the REO's Data Repository, the [Default EU Resolver] is used to retrieve the DPP from a the [Archive] in the same manner. The distinction between the resolvers and the data repositories is only conceptual in order to underline the flow of data. Any implementor can potentially

opt to integrate them so that the Resolver will be able to return the DPP data without a redirection in a seamless way.

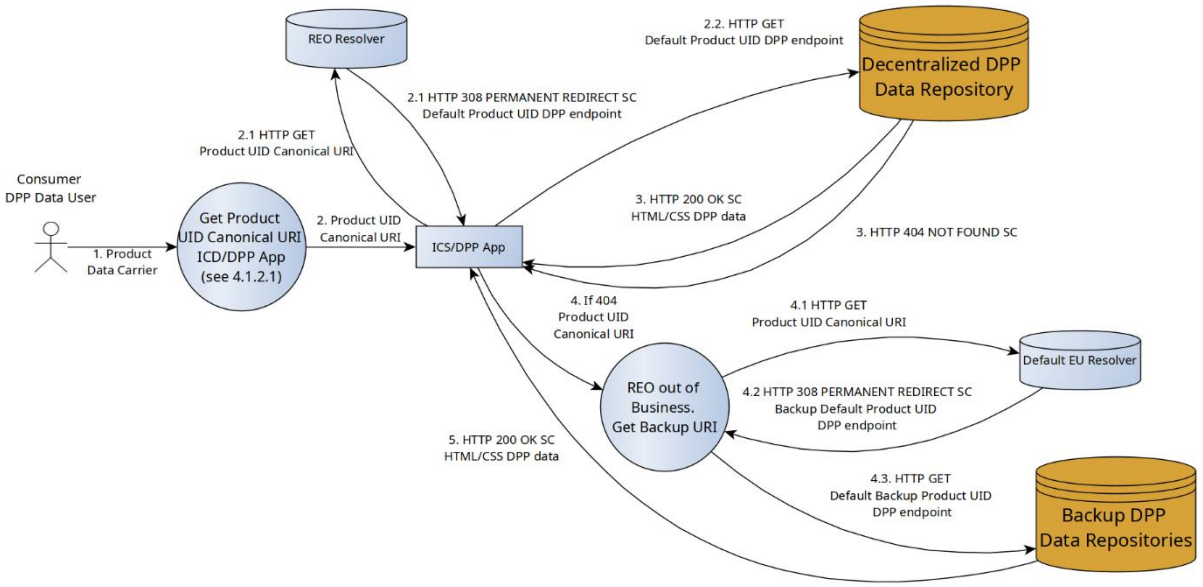

Figure 9. HTTP DFD – Using a DPP – The Default (Consumer) Data Flow

4.1.2.3 Role-based Data Flow – Recycler

This data flow introduces the [PDP], needed to access privileged, non-public, information. The 'Recycler' represents any circular economy actor needing privileged access to DPP Information.

There are many reasons why the access needs to be limited. This ranges from data protection reasons over security and trade secrets to business process imperatives. The Recycler is a privileged user that has access to more refined, sensitive and restricted DPP information than the default consumer. But constraints in the system are not limited to access control. All policy can be modelled here. Using [ODRL], DPP Linked data can be annotated with usage limitations. This way it is e.g., possible to limit data to the use for recycling while excluding market research purposes. Such usage limitation can reflect business agreements between a REO and his recycling network[9]. In this case, the [DDR] will not only contain the DPP data, but also policy data that is queried together using [SPARQL].

It is obvious that access control needs an established Identity Management. Before the Recycler can have access to these data it must request credentials like login and password or some bearer token from the REO. Another option is to allow a trusted actor like a Market Authority, alliances, etc. provide bearer tokens or Verifiable Credentials ([VC]s) to the various actors of the ecosystem (e.g., recyclers) based on their role and business branch during their registration to the EU registry. The REOs will have to value those Market Authority bearer tokens and allow access to their non-public data based on the corresponding roles. The data flow is the same as in the default consumer case. The only difference is that the Recycler must discover the available [RFC8288] Typed Links with the needed roles to make the corresponding [RFC9110] HTTP GET requests. In order to discover the Typed Links available from

[9] This has been demonstrated in the SPECIAL project (https://specialprivacy.ercim.eu/) and the Mosaicrown project (https://mosaicrown.eu) (visited 2024-01-31)

a given Resolver, the Recycler issues a HTTP HEAD request. According to [RFC9264], the Resolver will now return a list of all available Link-types. This HEAD request is usually not subject to access control, but could be. In this case, the Recycler would have to submit the required credentials (e.g., a JWT token [RFC7519]) in the corresponding HTTP HEAD request which will be validated by the [PDP] of the backends.

The Link-type feature can serve many purposes. First, it can determine the role of the requester and redirect them to different, e.g., non-public, resources. The Link-type redirects into another workflow that includes PDP. Second, the Link-type can cause the Resolver to return a completely different URI. In this case, the Recycler issues a HTTP GET request on the URI given by the [Product UID] and gets back a totally different URI that may even point to another domain. Third, announcing a certain role may imply specific information needs. As a URI can also include a query, the Resolver can return a query-URI for certain Link-types. In this case, the Link-type serves as a trigger for a prefabricated query into the [DDR]. Link-types can also offer endpoints for getting the DPPs of a set of products for supporting batch querying for privileged actors like Market Authorities as described later on. Furthermore, specific needs of the Recycler, like the delivery of a specific machine-readable data format can be taken into account for delivery. It is expected that the response to the typical Recycler request will deliver DPP Data in a machine-readable format including semantics, e.g. [JSON-LD].

If recyclers shred a product that has a DPP, they may be asked by the authorities to note that into the DPP as e.g., "destroyed". In this case, the recyclers need write access to the DPP. In this case, the repairer data flow below applies.

But the variety of possible reactions does not end there. In case the data are to be consumed by machines the HTTP request can be set to accept any other MIME-Type in the HTTP header to indicate the preferred response type.

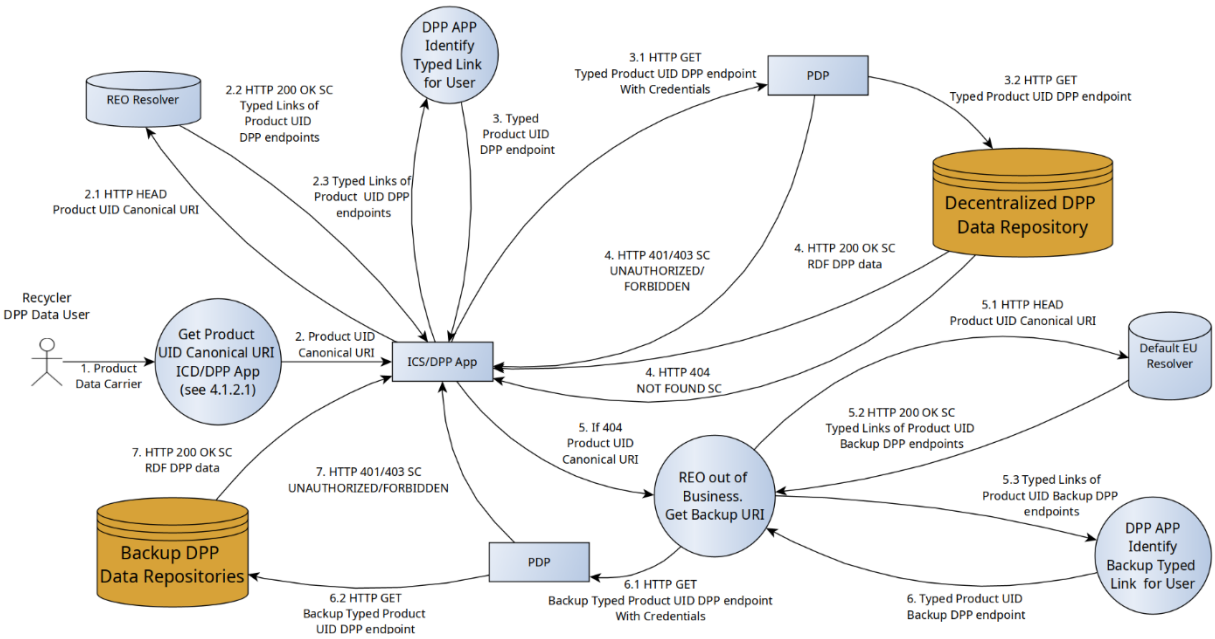

Figure 10. HTTP DFD – Using a DPP – Role-based Data Flow - Recycler

4.1.2.4 Role-based Data Flow – Repairer & Update of the DPP

On top of having privileged access, this data flow introduces the concept of writing back information into the [DDR]. A typical role in this respect is the Repairer. The 'Repairer' role is defined as a privileged user that is authorized by the REO to write into the DPP. But the role is not limited to Repairers. One can imagine all kinds of situations where there is a legitimate interest and benefit to write additional data into the [DDR] or where data in the [DDR] needs to be corrected.

The role of repairer (or, e.g., maintenance) is a crucial one in the circular economy. If a product is repaired, it stays longer in the market, producing less waste and carbon impact. The goal of circularity is greatly helped if it is known what *new* parts are in an instance of a product. Some things could be recovered, and statistics can be made over the question of how often a product is repaired or recycled.

But the Repairer is also a very challenging role for the system. Recyclers have privileged access to DPP data but can't alter data, they have only read access. The risk for the Recycler case is therefore limited to unwanted information disclosure. The Repairer can write into the [DDR]. On top of unwanted disclosure, the risk now encompasses loss of data by unwanted erasure and loss of data quality by the introduction of poisoned false information. Or worse, introduction of true data that is embarrassing for the REO. Many REOs will therefore be very reluctant to allow write access to their DDR.

Especially as for higher investment goods, there can be many repair shops that are totally independent from each other. To allow for that, a REO, who remains the owner and maintainer of the DPP, needs to enrol the repair shops into the system. Car manufacturers do that already. The repair role can be implemented in the same Decentralized Data Repository. It is technically possible to separate the DPP Data KG into REO information and Repairer information running on two different [DDR]s. The repair repository is then queried only in case of need. The latter solution has significant advantages in terms of security as only repair information could be affected by security weaknesses in the identity management with the repairers. With [Provenance] Vocabularies the origin of write events and repair information can be written back into the [DDR] in machine readable ways to allow for automatic assessment of data quality.

The [ESPR] reminds in Art. 9 that GDPR applies. But legally, this seems more than a reminder. It looks rather like a clarification that the [ESPR] itself is not a legal ground for the processing of personal data in the sense of Art 6 (1) c GDPR. The system as described here has no technical difficulty in carrying the personal data consent information together with the DPP data. Several systems already work using Linked data to express not only data, but also consent and to which data it applies. This means that if there is an interface to collect consent, e.g., physically in the repair shop, this fact can be easily stored within the repair information. Corresponding compliance can be demonstrated easily as demonstrated by [SPECIAL].

In the same manner as in any privileged role, the Repairer must request credentials from the REO and should make an HTTP HEAD request to the Resolver to identify the Typed Link for the update. Through these credentials, the Repairer will be able to update the corresponding part of the DPP data stored in the DPP Data Repository, if the [PDP] validates the credentials. For accountability, provenance and spamming reasons, the Repairers should sign the repair data. Another more complicated option for this dataflow, is to allow the REO to retrieve data from the Repairer's Data Repositories. In this case the REO has to be notified about the repair process, along with the endpoint of the Repairer's Data Repository to fetch the data and provide them to the DPP. In any case, the data ingested/retrieved by the Repairer are included in the DPP after their validation using the [SHACL] template.

In the following Data Flow Diagram, we focus on the first case, where the Repairer is able to update the DPP data repository. Before updating the DPP data the Repairer has to retrieve the latest DPP data from the data repository. The Repairer can then proceed with assembling the updated DPP data, which might be limited to a specific part of the DPP that corresponds to the repairs. Before updating the data in the DPP repository, the Repairer has to validate the updated DPP using the [SHACL] Control Engine provided by the [EU Registry] and the corresponding [SHACL] template for the specific product branch. If valid the Repairer submits an HTTP PUT request to update the corresponding DPP at the DPP Data Repository. Reporting documents themselves (e.g., pdfs) or links to the reporting documents can be provided in the HTTP PUT payload for REOs to review. The update may trigger a new validation of the DPP from the REO’s side before storing the data.

A more sophisticated approach to the write access with additional components for trust and security is provided in the DID Data flows described in [section 4.2.2.4].

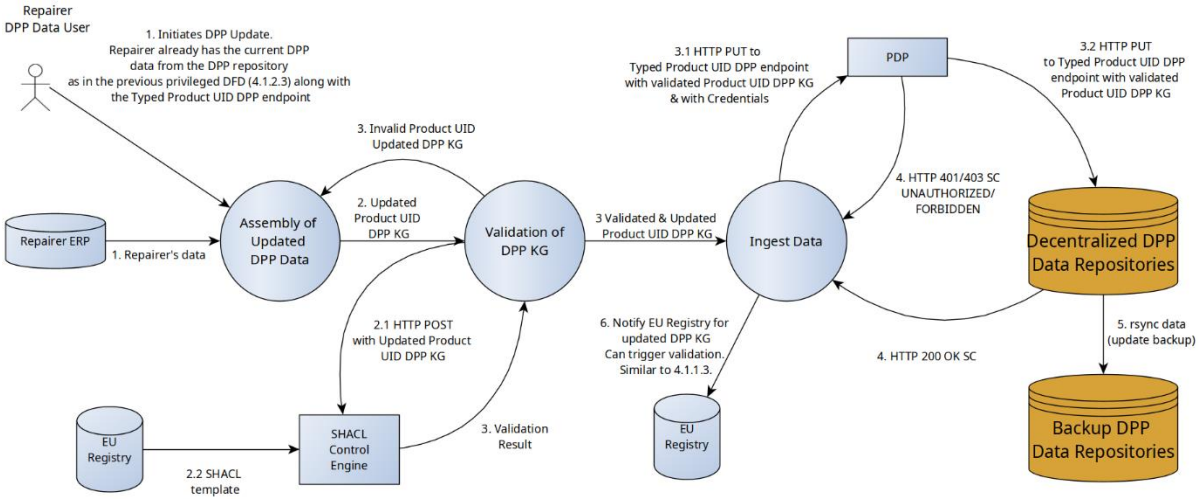

Figure 11. HTTP DFD – Using a DPP – Role-based Data Flow – Repairer & Update of the DPP

4.1.2.5 Role-based Data Flow – Authorities

The Market Authority role is fundamentally different from the previous roles. This role is information centric as it does not start with the tangible good, but it starts with the URI that leads to the DPP Data. The [Product UID]-URI is used to send the GET request to the resolver. This URI will not always come from a scanning device but might also include sets of [Product UID]s retrieved from registries or other information systems of the Authorities. As a privileged user the Market Authority has a right to also access the non-public information that may e.g contain trade secrets. In order to do so, the Authority needs to request the appropriate credentials from the REO in order get access to protected DPP information. This need for access credentials distinguishes this role from the Default Data Consumer. Such access control credentials have to be made available by the REO on demand or they could potentially already be recorded in the EU registery when registering the [Product UID] In addition, a Market Authority might also want to validate a single or a set of DPPs, resolving a single or a set of Product UIDs with Typed Links. As with any privileged role, the request to the services of the DPP Data Repository is done with the provided credentials (e.g., a JWT token [RFC7519]) to assign and verify the corresponding role to the request. If authorized, the request returns the Market Authority related DPP data for this specific Product UID in the requested format (e.g., [RDF]). The DPP data could then be validated using the corresponding [SHACL] template. In the case of a set of Product UIDs, the services should provide endpoints that allow the validation and the retrieval of numerous (i.e., millions) DPPs in an efficient way through a single HTTP GET request.

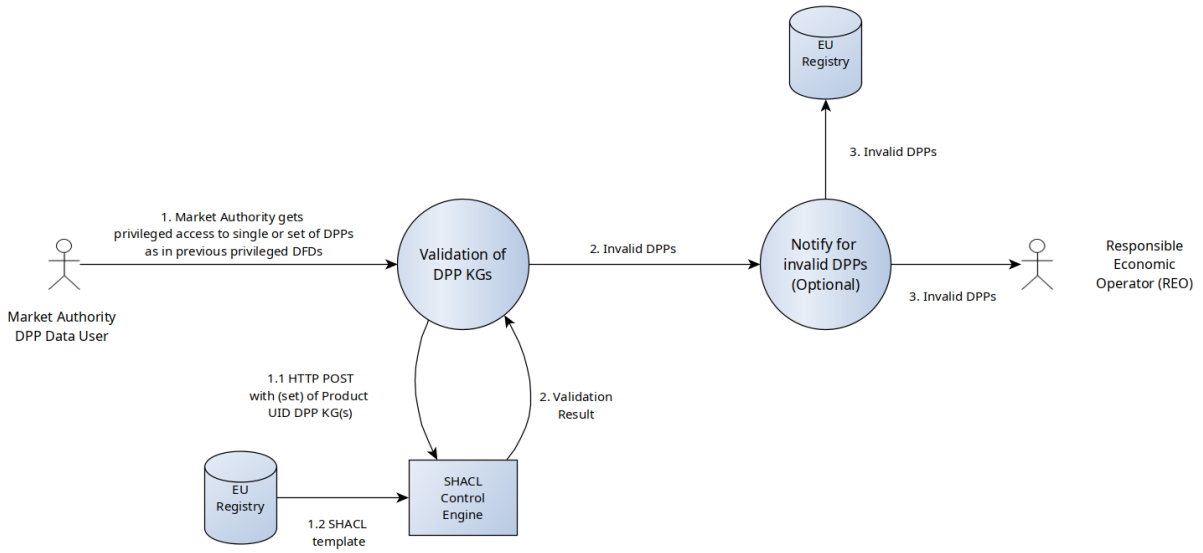

Figure 12. HTTP DFD – Using a DPP – Role-based Data Flow - Authorities

4.1.2.6 Role-based Data Flow – Remanufacturer

A remanufacturer is a company or industry that engages in making an existing product new again. Remanufacturing is the rebuilding of a product to the specifications of the original product matching the same customer expectations as a new product. The process requires a combination of repair and replacement of components and modules with new and/or recycled components, including parts subject to degradation affecting the performance or the expected life of the product. It is a costly process and the products that are remanufactured are considered new products that are subject to ecodesign requirements if they fall within the scope of a delegated act. As a result, they require a *new* DPP. The role has many commonalities with the role of the DFD described in [section 4.1.1.2] because they Remanufacturers must assemble data and issue a DPP for the remanufactured product, with the critical difference that the Remanufacturer has to consume and update the original product DPP so that any consumer can be notified that the DPP of the original product(s) has been invalidated and/or should point to the new DPP. Other approaches for DPP invalidation could also be considered, such as removing the original Product UID from the [Resolvers] and the [DDR]s. Finally, the Remanufacturer may need to include links to the previous DPP(s) into the new product's DPP.

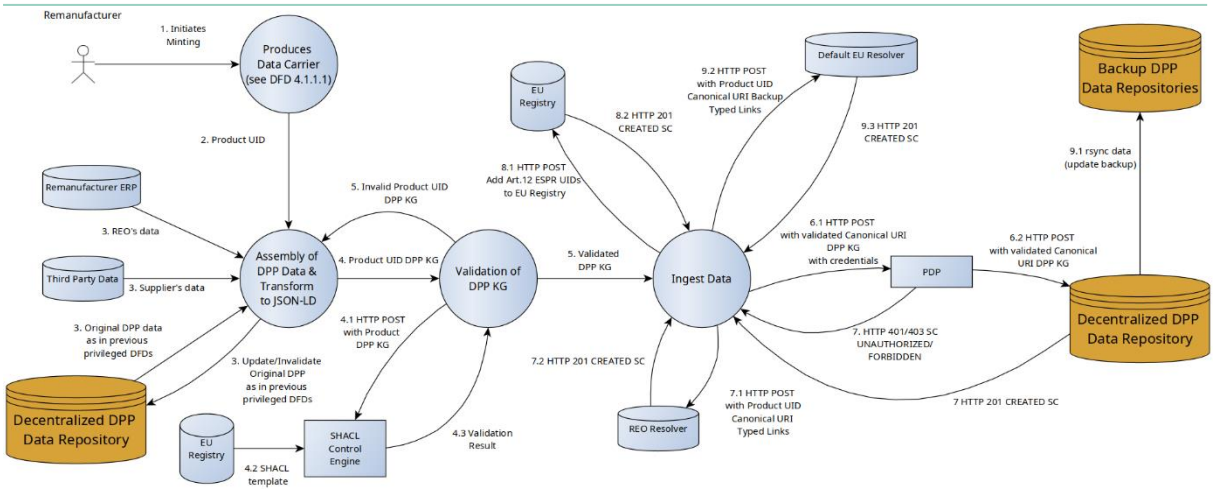

Figure 13. HTTP DFD – Using a DPP – Role-based Data Flow - Remanufacturer

4.2 DID Data flows

In the following, we describe the DFDs of the alternative architecture that is based on [DID]s. All data flows for non-consumer actors, presuppose that the actors own an [Actor DID] created through the [DPP App], associated with the corresponding [DID Document] that is stored in the Verifiable Data Registry (VDR). Regarding the resolution of [DID]s and any [DID] method-based operations, we consider that the [DPP App] wallet is responsible for the resolution of the corresponding [DID] using the appropriate [DID] method. External [DID] resolvers like the Universal DID Resolver could also be used. For REOs, the [DID Document] will contain the querying service endpoints that will be used for retrieving the DPP data. The endpoint ID (e.g., service=dpp) that provides the default public DPP data, along with the backup endpoint ID (e.g., service=backup) must be provided as a parameter in the [DID] URL that the [Data Carrier] holds, for the Default Consumer use case. The [Actor DID]s could be stored in addition to the [Product DID]s in the [EU-Registry] or some other service. This would allow for advanced features in the identity management for repairers and recyclers e.g. The use of [VC]s for authorization and verification are described in the next section (see [section 4.3]).

4.2.1 Creating a DPP

Initiating a DPP for the [DID] case consists of the same three main processes as in the HTTP case: a) the minting of the [Product DID] by the REO and the creation and attachment of the [Data Carrier] to the product, b) compiling the DPP information with data gathered from the REO or any other third-party suppliers and its storage to the DPP Data Repository, and finally, c) registering any relevant information with the authorities.

4.2.1.1 Minting a Product DID

During the minting process, the REO creates a [Product DID] that is controlled by its [Actor DID]. Notice that we assume that the REO has already created an [Actor DID] through the corresponding [DPP App]. The process of the [Product DID] minting is the following. The REO creates the [DID Document] where the [REO's DID] is set as the controller so it can make any modifications to the [DID Document] in the future. The REO also adds any other relevant metadata (e.g., alsoKnownAs) and the corresponding service IDs and endpoints in the service property of the document that will be used in the [DID] URL (e.g., dpp) for requesting the DPP data. However, if the endpoint is not known at the time of the [DID

Document] creation (e.g., it depends on the [Product DID] that will be minted), it can be updated after the [Product DID] has been created or while adding the DPP information to the [DDR] (see [section 4.2.1.2]). A backup service ID can also be added to the [DID] URL along with the corresponding service section in the [DID Document]. Then the REO submits the [DID Document] to the corresponding Create [DID] method operation. The wallet is responsible for setting the verification methods and key types. When the Create [DID] method operation is invoked the VDR is responsible for creating the [Product DID] and storing the corresponding Product [DID Document]. The [Product DID] can also be stored in a registry of minted [Product DID]s for the corresponding REO in the [DPP Minting App]. The [Product DID] is then used to create the [Product DID] URL. This URL is constructed using the [Product DID] along with the service ID parameter. The service parameter is appended to the URL and will be used for identifying the corresponding default DPP consumer service endpoint from the [Product DID] document during the resolution of the [DID] URL. After minting, the REO must produce a [Data Carrier] that will hold this UID and attach the [Data Carrier] to the product. The [Data Carrier] can be a QR, an RFID or other [Data Carrier]. If the service endpoint is based on the [Product DID], the REO can update the Product [DID Document] with the service endpoint for the service ID encoded in the [Product DID] URL.

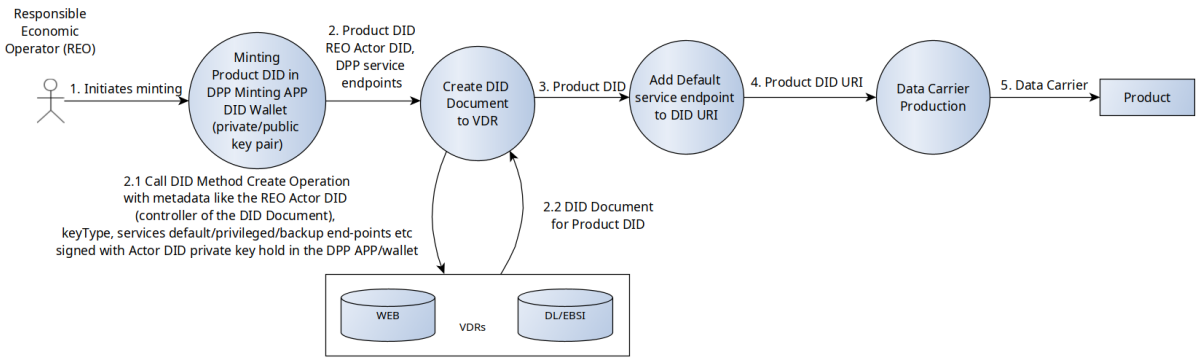

Figure 14. DID DFD – Minting a Product DID

4.2.1.2 Assembling and Issuing the DPP Data

Similarly to the HTTP case, each [Product DID] is associated with the DPP data for this specific product. The REO is responsible for gathering all the data, cleaning, normalizing and transforming them to the required format and vocabulary to assemble the DPP. We assume that the DPP KG is serialized in the [JSON-LD] format. The source of the data can be the REO's databases and information systems (e.g., ERP), and/or third-party systems (e.g., suppliers ERPs). The DPP data should adhere to the [ESPR] regulation for setting ecodesign requirements based on the sustainability and circularity aspects and should be compliant. One advanced feature would be to then check the resulting DPP Data against the [SHACL] template that would ideally be provided by the Public Authorities. Finally, the data are stored in the DPP [DDR] of the REO or operated by a DPP service provider under REO's control. The DPP data must be accessible from any role that will have access to this [Product DID] with the appropriate access control rights. The access to the various resources and services of the DPP can be described through [ODRL] policies. The Policy Decision Point ([PDP]) is responsible for evaluating the access requests against the authorization policies. The REO has to update the service IDs and endpoints in the [DID Document], if they were not known during the minting. The services include the default consumer and the privileged endpoints, along with the backup ones. The architecture is

general enough, so an implementation is not limited to what technologies will be deployed for the DPP [DDR]. For example, an implementation could be based on a triplestore for storing the corresponding information, while actors interact to the triplestore through a REST-API, that associates the HTTP endpoints to [SPARQL] queries that are then submitted to the triplestore. REOs can use their own databases, ERPs, etc., but they must implement the appropriate transformations for interoperability reasons. The requests to the endpoints are submitted with the corresponding credentials (e.g., a JWT token [RFC7519]) when needed, and the REST services are responsible for transforming the request to the appropriate [SPARQL] query. Endpoints that inject DPPs to the Data Repository or accept adhoc [SPARQL] queries for actors with the appropriate access rights (e.g., REO, Market Authority) are also available. The REO can sign the DPP data with its corresponding [Actor DID] private key before injecting the DPP data to the DPP [DDR]. As a result, any actor consuming the DPP data can verify that the data were assembled and provided by the REO. The verifier will have to retrieve the public key of the REO [Actor DID] from the VDR in order to do the verification.

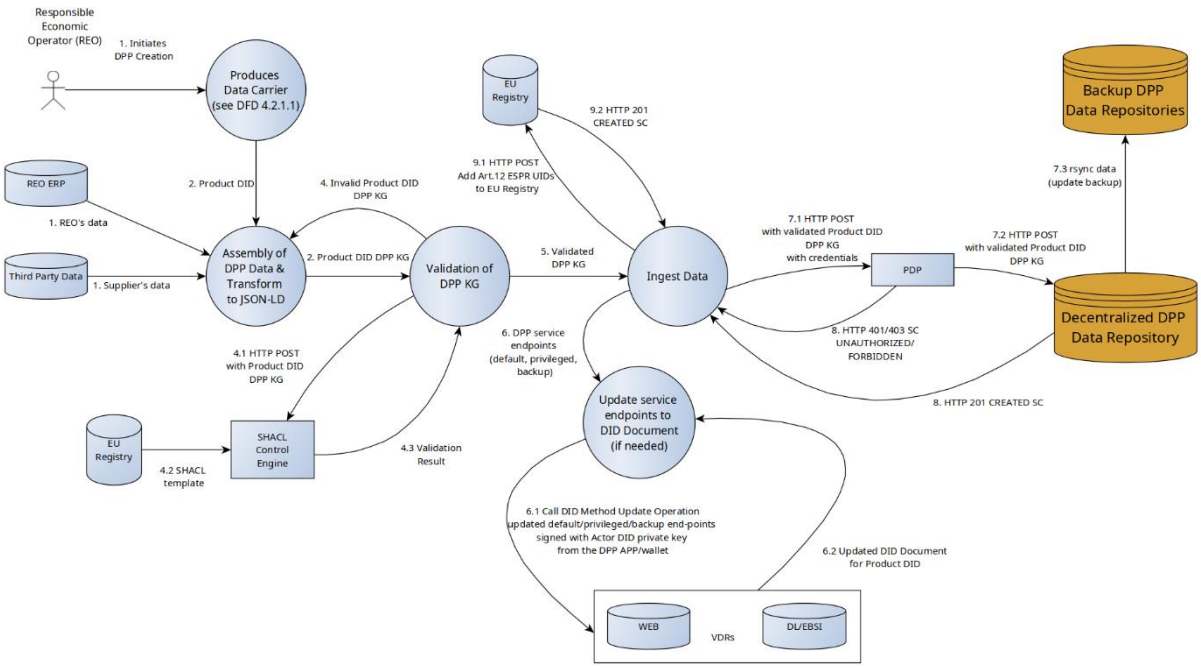

Figure 15. DID DFD – Assembling and Issuing the DPP Data

4.2.1.3 Registering the DPP with Authorities

This step is almost the same as in the HTTP version described earlier. After the REO has inserted the DPP of the [Product DID] it informs the [EU-Registry] of the required data elements and identifiers. Authorities can request the DPP KG data from REO DPP [DDR]. First, they need to discover the service endpoint. To do so, the Authorities can now retrieve the [DID Document] of each [Product DID] from the VDR, as discussed in the Role-based Data Flow for the Market Authorities described later (see [section 4.2.2.5]). For access to non-public information, the Authorities needs privileged access to the REO's DPP [DDR] and the REO must provide the corresponding credentials. As an additional feature and service, Authorities could validate the DPP using the [SHACL] template stored in the [EU Registry] for this specific product and submitting the appropriate request to the corresponding DPP [DDR]. If

the validation fails, the Authority could store the status regarding this specific REO and DPP in the [EU Registry] or into their own database or DDR, or alternatively reject the registration of a DPP.

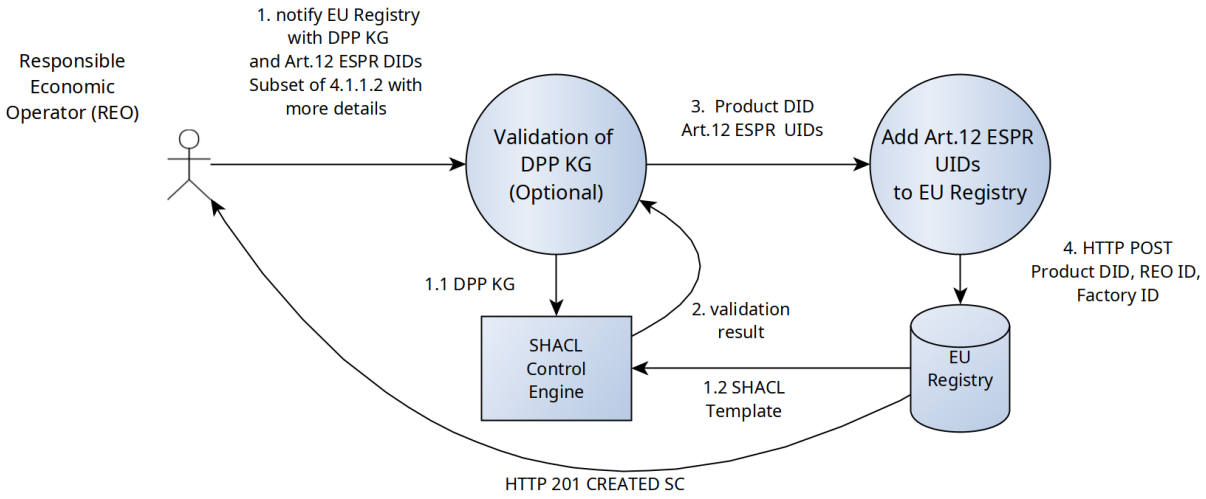

Figure 16. DID DFD – Registering the DPP with Authorities

4.2.2 Using a DPP

In this data flow we showcase the use cases of transforming the [Product DID] URL on the [Data Carrier] attached to the product to a usable URI which is then dereferenced to get the corresponding DPP information. The difference to the HTTP approach is that there is no [REO Resolver], but the URI that will be used for dereferencing is available in the corresponding [DID Document] stored on the VDR. We also showcase the data flows of other privileged actors (e.g., Recycler, Market Authority, Remanufacturer) that can get more refined information for a Product DPP or can possibly update the DPP information.

4.2.2.1 From Data Carrier to a Usable URI

This data flow transforms any [Product DID] URL assigned to a [Data Carrier] attached to a product into a resolvable URI. The DPP Data user uses the [DPP App] and through a scanning device scans the [Data Carrier] attached to a Product. The scanning device can be the camera of a mobile device, or any other scanner device connected to the [ICD] that the [DPP App] can have access to. The [Product DID] URL and the corresponding [Product DID] is then resolved by the [DPP App], retrieving the [DID Document] associated with the [Product DID]. Using the service parameter of the [Product DID] URL, the [DPP App] parses the [DID Document] to get the REO's DPP service endpoint.

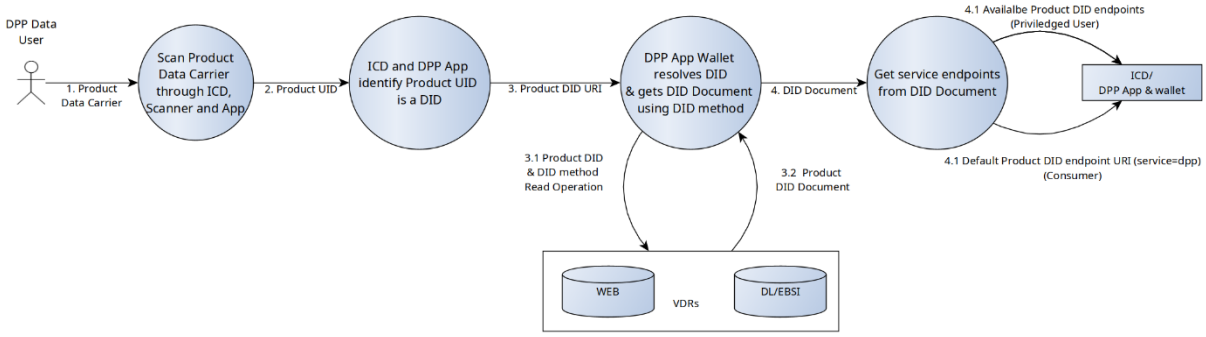

Figure 17. DID DFD – Using a DPP – From Data Carrier to a Usable URI

4.2.2.2 The Default (Consumer) Data Flow

Here we assume a plain DPP Data User (i.e., consumer). The process is the same as in the HTTP case. After transforming the [Data Carrier] to the REO's service endpoint for this [Product DID], the [DPP App] sends a GET request with no credentials to the corresponding endpoint. The REST-API transforms this GET request to a SPARQL query that retrieves all the public information of a DPP and returns a plain HTML/CSS page. No access control functionality is required. In case the data are to be consumed by machines the request can accept RDF data. If the default service returns a 404 status code then the [DPP App] will send a GET request to the corresponding backup endpoint.

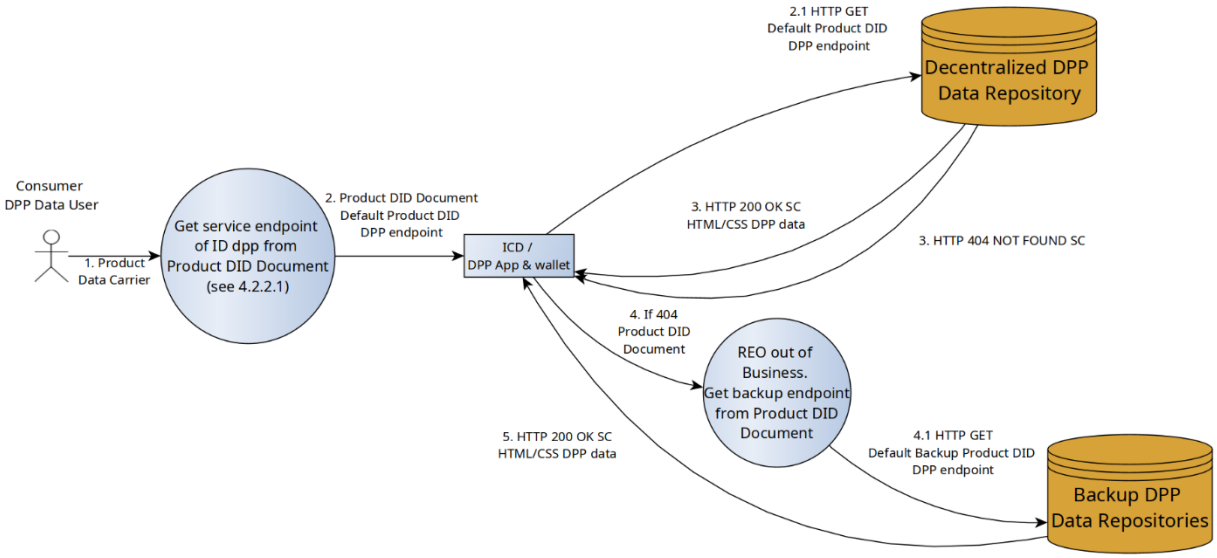

Figure 18. DID DFD – Using a DPP – The Default (Consumer) Data Flow

4.2.2.3 Role-based Data Flow – Recycler

As in the HTTP data flow, the Recycler needs to request credentials from the REO to get access to more refined DPP information than the default plain [Data Consumer] role. Else the data flow is the same as in the Default Data Consumer role. The only difference is that after scanning the [Data Carrier], the request to the REST-API must be done with credentials (e.g., a JWT token [RFC7519]) to assign and verify the access of the request to privileged resources and actions. A two-way authentication approach that also uses a challenge message encrypted with the public key of the [Actor DID] can be

deployed. The [Actor DID] public key and verification methods can be retrieved using the appropriate [DID] method. If authorized the request returns the Recycler related DPP data for this specific [Product DID] in the requested format (e.g., [RDF]).

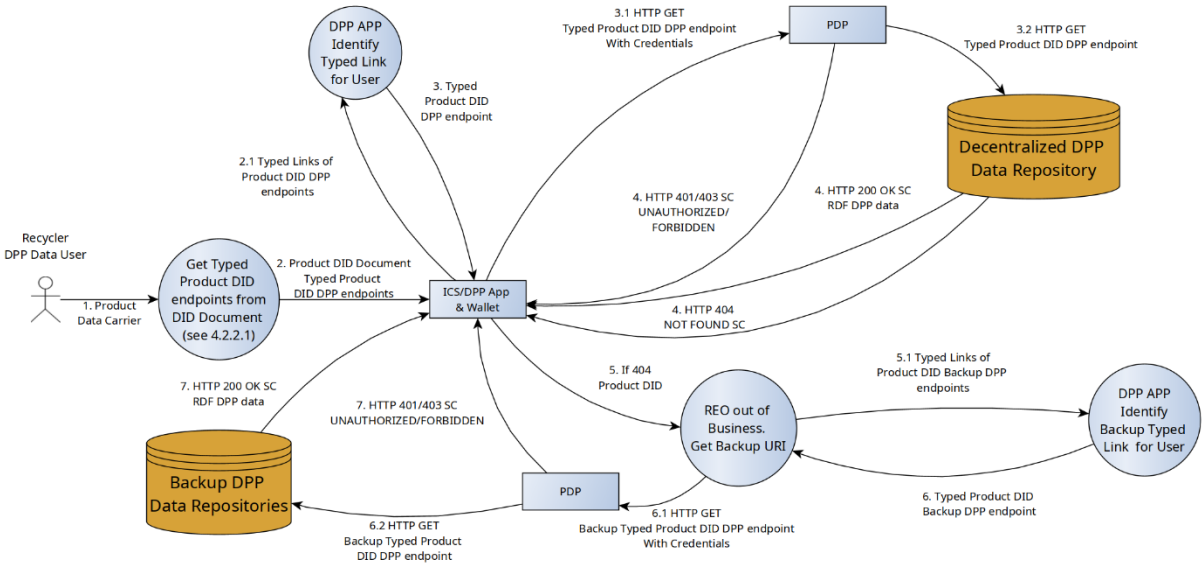

Figure 19. DID DFD – Using a DPP – Role-based Data Flow - Recycler

4.2.2.4 Role-based Data Flow – Repairer & Update of the DPP

Similarly to the case of the HTTP data flow, the Repairer needs to request credentials from the REO to have access to the corresponding REST update endpoints and be able to update the DPP data. Another option is that the Repairers can store the corresponding data to their own [DDR]s and notify the REOs to link to their data in the DPP. Here we showcase the DFD for the first case. The data flow is the same as in the Default Data Consumer role. The only difference is that after scanning the data carrier, the request to the REST-API is done with the provided credentials (e.g., a JWT token [RFC7519]) to assign and verify the role. A two-way authentication approach that also uses a challenge message encrypted with the public key of the [Actor DID] can be deployed. The [Actor DID] public key and verification methods can be retrieved using the appropriate [DID] method. If authorized the request updates the Repairer related DPP data for this specific [Product DID]. The Repairer can sign the data using its private key and its [Actor DID] so that any actor can validate them by retrieving the Repairer public key and verification method from the VDR.

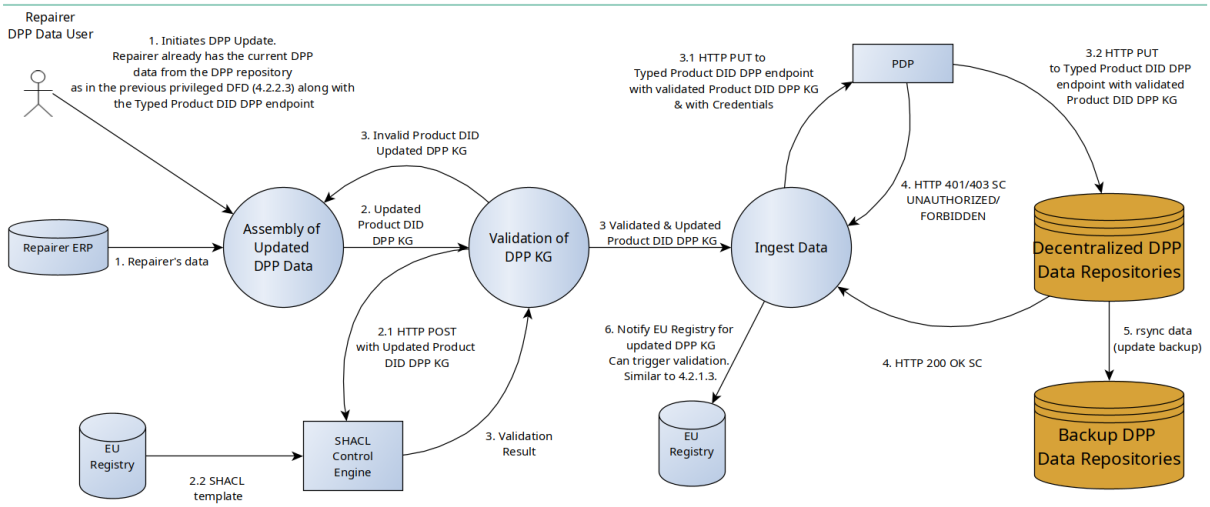

Figure 20. DID DFD – Using a DPP – Role-based Data Flow – Repairer & Update of the DPP

4.2.2.5 Role-based Data Flow – Authorities

As in the HTTP data flow, the Market Authority needs to request credentials from the REO in order to get access to more refined DPP information than the Default Data Consumer role as in the previous privileged roles. The Market Authority can resolve a single or a set of [Product DID]s. As with any privileged role, the request to the REST-API is done with the provided credentials (e.g., a JWT token [RFC7519]) to assign and verify the corresponding role to the request. A two-way authentication approach that also uses a challenge message encrypted with the public key of the [Actor DID] can be deployed. The [Actor DID] public key and verification methods can be retrieved using the appropriate [DID] method. If authorized the request returns the Market Authority related DPP data for this specific [Product DID] in the requested format (e.g., [RDF]). The DPP data is validated using the corresponding [SHACL] template. In the case of a set of [Product UID]s, the Typed Links and the corresponding services should provide endpoints that allow the validation and the retrieval of numerous (i.e., millions) DPPs in an efficient way through a single HTTP GET request. The Market Authority can optionally verify the signature of the DPP using the REO's public key retrieved from the REO's [DID Document].

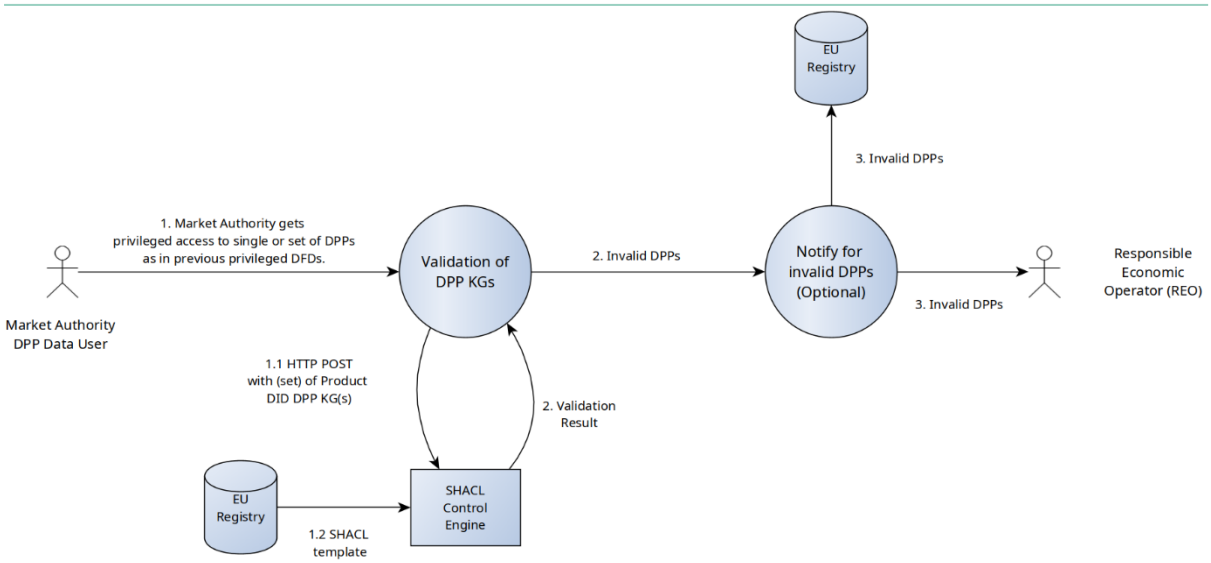

Figure 21. DID DFD – Using a DPP – Role-based Data Flow – Authorities

4.2.2.6 Role-based Data Flow – Remanufacturer

The [DID] DFD is similar to the HTTP DFD for the Remanufacturer, with the exception that the Remanufacturer has to provide a new [DID Document] for the new [Product DID]. Additionally, in this DFD we showcase a scenario where the REO is notified by the Remanufacturer to invalidate the original [DID]. In the corresponding HTTP DFD (see [section 4.1.2.6]) we depicted a data flow where the Remanufacturer was granted access to update the DPP of the original product and invalidate it.

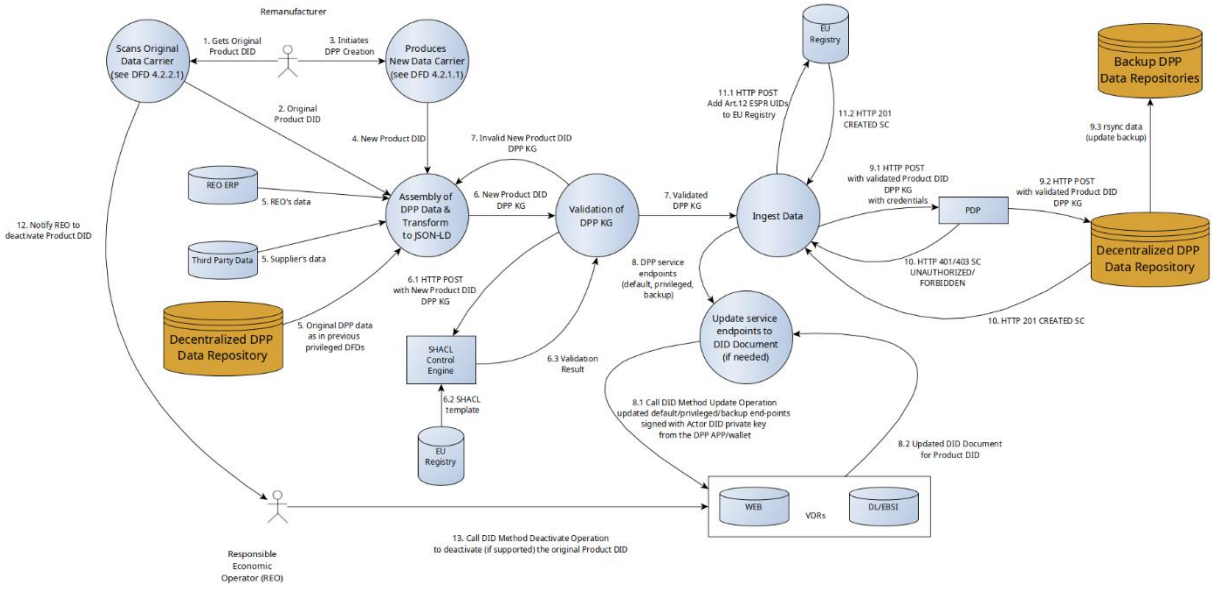

Figure 22. DID DFD – Using a DPP – Role-based Data Flow – Remanufacturer

4.3 Advanced Features

In this section we discuss advanced features for the [DID] approach that take advantage of the Verifiable Credentials ([VC]s). However, [VC]s do not depend on [DID]s and [DID]s do not depend on [VC]s. However, they complement each other. [DID]-based URIs can be used for expressing global, persistent, self-sovereign and decentralized identifiers associated with subjects, issuers, holders,

credential status lists, cryptographic keys, and other machine-readable information associated with a [VC]. In the following DFDs, we focus on [VC]s that deploy [DID]s.

4.3.1 VC Issuance

The [VC]s are issued by actors like the REO/Remanufacturer for the various privileged actors (e.g., Recycler, Repairer, Market Authority) describing their authorized and privileged role. The actor asking the [VC] issuance needs to know the [DID] of the issuer to resolve it and fetch the corresponding [DID Document]. The [DID Document] should provide in the reported services the [VC] issuance endpoint to which the asking actor should make an HTTP POST request with all the relevant metadata such as the asking [Actor DID] and the requesting role and privileges. The REO/Remanufacturer has to audit the request of the asking actor and decide whether the [VC] should be issued or not. In case of success, the [VC] is issued, usually in the form of a [JSON-LD], where the issuer is the [DID] of the REO/Remanufacturer and the [VC] contains the credential subjects for the corresponding [DID] of the asking actor. The [VC] also contains a digital proof that makes the credential tamper-evident. The verification method uses the public key of the issuing [Actor DID]. Finally, the asking actor stores the issued [VC] in the [VC] wallet of the [DPP App].

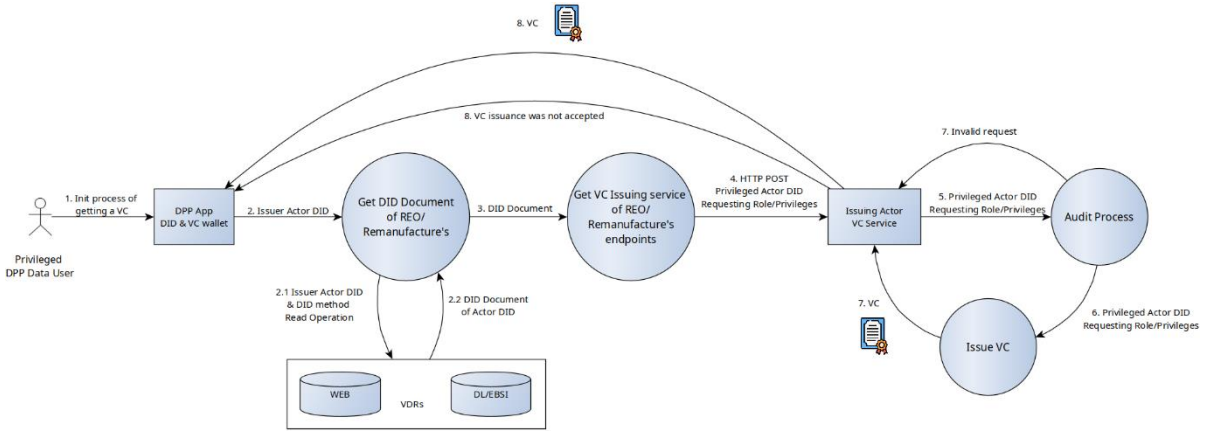

Figure 23.VC Issuance

4.3.2 VC-based Authorization

[VC]s can be used for authentication/authorization purposes on the REO/Remanufacturer DPP endpoints. The actors must provide their [VC]s in their requests to get privileged access to the corresponding DPP data (e.g., retrieving sensitive data or updating part of the DPP data). Specifically, the verifier (i.e., the REOs endpoints), request a verifiable presentation of the [VC]. A unique challenge/domain is provided when requesting a verifiable presentation from a privileged actor, which is used for the authentication phase. These properties are then included in the proof section of the corresponding verifiable presentation of the [VC]s to avoid a replay attack. The challenge should be random to avoid attackers been able to predict it. The domain is just used in the rare case that two verifiers generate the same random challenge. A verifiable presentation without the matching verifier challenge and domain in its proof is considered invalid by the verifiers. The verifier is able to check the authenticity of the verifiable presentation using the public key of the holder [Actor DID]. Subsequently, the REOs endpoints validate the [VC]s using their corresponding [Actor DID] public key from their wallet and if they are invalid, they deny access to the data. If the [VC] was not issued by the REO/Remanufacturer, or the [VC] has been revoked or has been invalidated, or the request was not

submitted by the subject of the [VC], it denies access. Else the REO/Remanufacture grants access for reading or updating the corresponding sensitive DPP data.

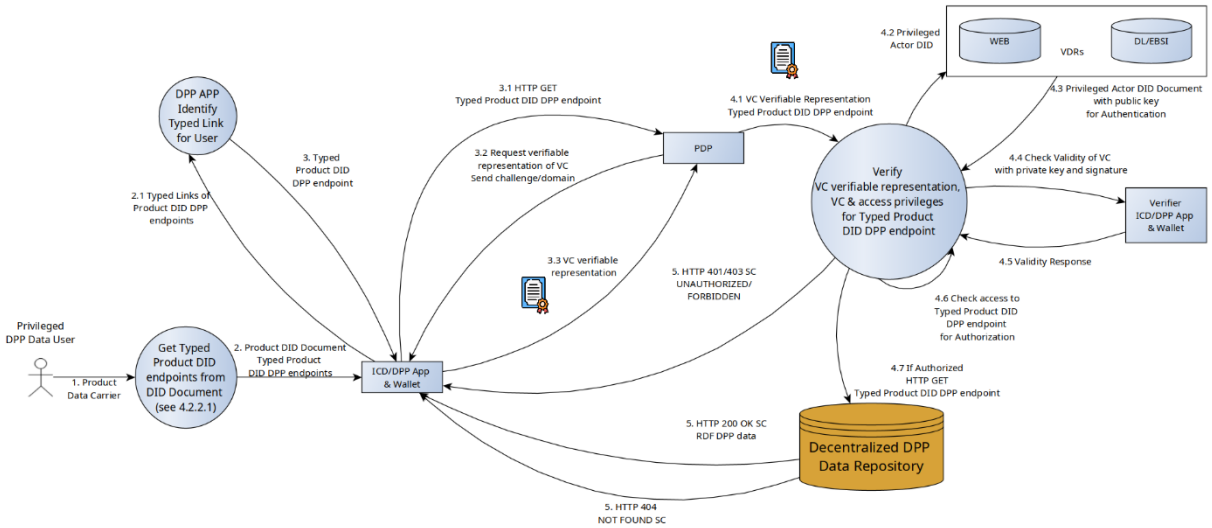

Figure 24. VC-based Authorization

4.3.3 VC-based Verification of Third-parties Information in DPPs

One way to allow third-party information to be integrated into a DPP is to allow write-access to the REO DPP data repository for these parties and let them sign the corresponding part of the DPP, that includes their [Actor DID], with their [Actor DID] private key for verification reasons. The [VC] issued by the REO could also be appended in those data from the REO as proof that the REO has provided access to this third-party to write to the DPP for privileged users like Market Authorities. In that case, any DPP consumer can verify the provided information by retrieving the public key of the third-party through the corresponding [DID] method, while the Market Authorities can verify the provided [VC] through the public key of the REO. In case a REO does not want to allow third parties to have write access to the DPP data repository for security or other reasons (e.g., spamming the DPP data, control over the DPP data), the suppliers can provide [VC]s to the corresponding REO for accessing and publishing information from their repositories. Those data should include the third-party [Actor DID] and should be signed by its private key, and should include the issued [VC], allowing clients to verify that the data were derived from the third-party, and authorized consumers that the REO had access to them.

Below we adapt the Repairer's ([section 4.2.2.4]) and Market Authorities DFDs ([section 4.2.2.5]) to this dataflow.

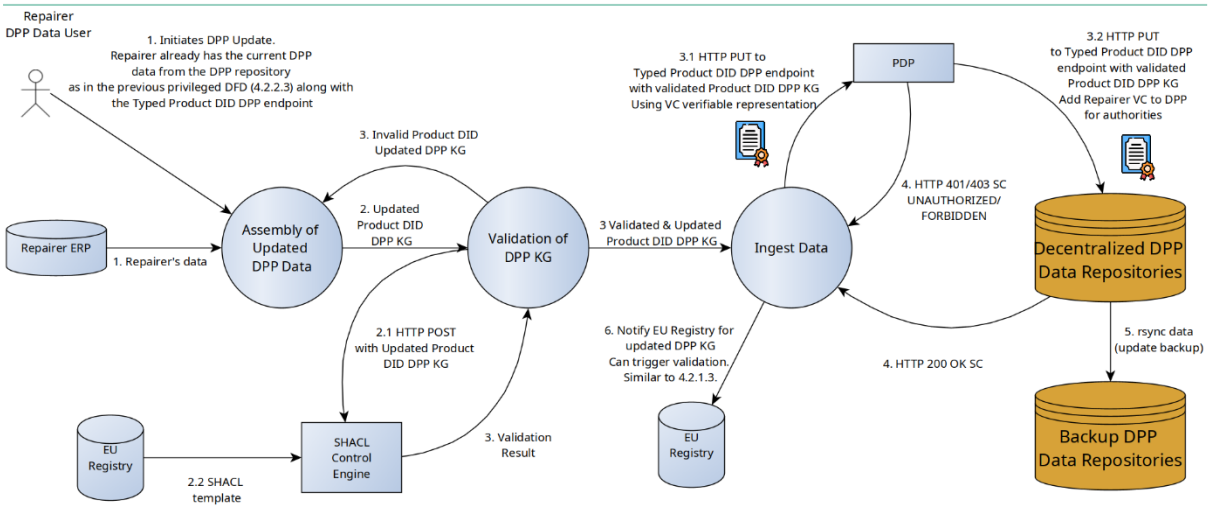

Figure 25. VC-based Verification of Third-parties information – Repairer

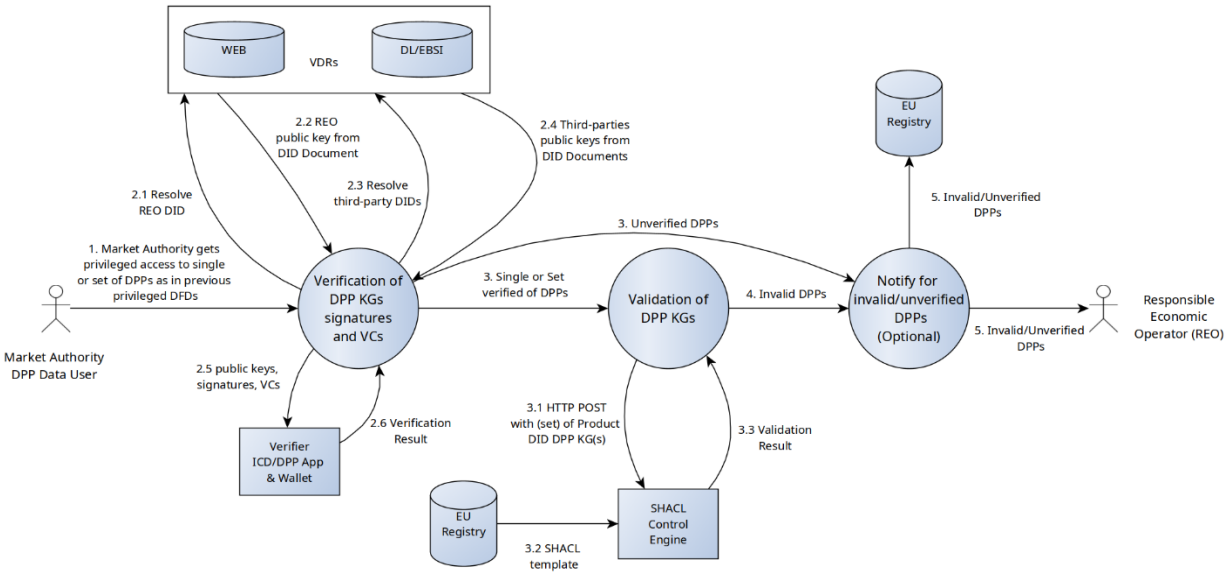

Figure 26. VC-based Verification of Third-parties information –Market Authority

5 Validation of the DPP System Architecture

The validation step is essential to ensure that the proposed DPP system architectures can support the user-stories that have been described in D4.1 (DPP user stories V1.0). As there are two different architectures to implement the DPP system, each step will be validated for both architectures. This is described in the two right most columns of the following tables.

If steps of the user stories cannot be validated using the proposed system architectures, developments will be required and will be described in the road mapping document D3.4.

5.1 Validation of the DPP System Architectures

5.1.1 User story 1: A stakeholder (e.g., economic operator) places a product on the market

Step	*User Story Step*	*Validation using HTTP URIs*	*Validation using DIDs*
1	The economic operator's IT system sends a request for a DPP data model for the relevant product category to assess the current regulatory DPP data requirements. Assumption: these DPP data requirements could be obtained via a query on a server.	This flow starts from the economic operator's IT system which sends a HTTP request to a dedicated server that stores the data model for product categories. This specific flow is described in the section [4.1.1.2] Assembling and Issuing the DPP Data.	This flow is described in section [4.2.1.2] Assembling and Issuing the DPP Data.
2	The economic operator creates/assesses on the supply side all data needed to populate the DPP (and to meet other external data requirements) in their IT system and creates the Product Identifier. This data can originate from individual (today mainly manual) data generation from suppliers into PLM or ERP systems, upstream traceability systems and more.	This step is described in [4.1.1.1] Minting a Product UID and [4.1.1.2] Assembling and Issuing the DPP Data. However, this step describes that the economic operator can create data on the supply side to populate the DPP.	This flow is described in [4.2.1.1] Minting a Product UID and [4.2.1.2] Assembling and Issuing the DPP Data.
3	The economic operator generates machine-readable DPP content and makes it available at one or more digital locations. Some data may be generated by service providers (e.g. safety data sheet, traceability, image banks) or external parties such as certification bodies. Appropriate access levels are assigned to each data point.	This step is described in [4.1.1.2] Assembling and Issuing the DPP Data.	This step is described in [4.2.1.2] Assembling and Issuing the DPP Data.

4	The economic operator registers the product's links in a resolver service component (self-managed or external), pointing to multiple data sources as mentioned above	This step is described in [4.1.1.2] Assembling and Issuing the DPP Data.	This step is described in [4.2.1.2] Assembling and Issuing the DPP Data.
5	The economic operator registers the product identifiers and other relevant content encoded in the data carrier (i.e. the web URL, depending on ID scheme) at the product ID granularity level specified in the relevant delegated act in the appropriate central registry managed by a competent market surveillance body. Remark: It is assumed that the registration in the central EU registry is at product model level, i.e., not at batch or item level. It will still be possible for economic operators placing product on the EU market to identify products at more granular levels, such as batch or item in the data carrier and corresponding DPPs when needed.	This step is described in [4.1.1.3] Registering the DPP with Authorities.	This step is described in [4.2.1.3] Registering the DPP with Authorities.
6	The economic operator places the product on the market (retail store, online etc...)	This step does not describe a data flow related to the DPP system architecture.	This step does not describe a data flow related to the DPP system architecture.

5.1.2 User story 2: A stakeholder (e.g., professional buyer) with a list of product identifiers gets DPP data about all the products on the list

Step	***User Story Step***	***Validation using HTTP URIs***	***Validation using DIDs***
1	A user has received a list of product identifiers (internally for a product review or from a supplier as part of a tender) encoded as web URLs or with a known resolver endpoint. This will be at model ID level, not at batch or item level. Remark on "level": Depending on industry, different terms are used for	This step describes a flow that can be outside of the DPP system architecture. The list of product identifiers could be printed or exchanged by email. Regarding digital exchange mechanisms,	This step describes a flow that can be outside of the DPP system architecture. The list of product identifiers could be printed or exchanged by email. Regarding digital exchange mechanisms,

	determining what is a model, product, style. The relevant distinction here is that it is not at batch or instance level, but at the level purchase decisions are made.	the list of identifiers should be exchanged in a known format.	the list of identifiers should be exchanged in a known format.
2	The data is entered in the user's internal IT system.	This is a local flow on the economic operator's IT system. If the data is received through a digital medium, then it is already in the economic operator's IT system.	This is a local flow on the economic operator's IT system. If the data is received through a digital medium, then it is already in the economic operator's IT system.
3	The IT system uses the web URLs from the list, or constructs web URLs using the known resolver endpoint URL to request all available links associated with the product identifiers from the resolution service component.	Local flow (optional, to construct the URL to which the requests will be sent). If the URLs must be constructed, the flow is described in section [4.1.2.1] From Data carrier to a usable URI. When the system user has all the URLs, he sends request to all of them to obtain the available links linked to the product identifiers. This is also described in section [4.1.2.1].	Using DID, the flow is described in [4.2.2.1] From Data carrier to a usable URI.
4	The resolver service endpoint responds back to the IT system with a list of links and their associated link types for the product identifier.	This step is the response that will receive the economic operator's IT system. It takes place in the same data flow as the previous step and is described in sections [4.1.2.3], [4.1.2.4] or [4.1.2.5] and [4.1.2.6].	Depending on the type of actor performing the request, different flows can occur, see [4.2.2.3], [4.2.2.4], [4.2.2.5] and [4.2.2.6].

		The resolvers send back all the available links for each product identifier. The IT system must be able to associate the list of received links to each product identifier.	
5	The IT system selects the relevant links and sends queries to them. The data sources receiving the request determines the appropriate access level of the querying party.	To obtain the relevant information for the economic operator, credentials must be provided. These credentials will be evaluated on the endpoint receiving the request. Depending on the result of the evaluation (credentials valid or credentials invalid), the system will reply with a specific set of data (see step 6). This request will most likely embed credentials, as described in sections in sections [4.1.2.3], [4.1.2.4], [4.1.2.5] and [4.1.2.6]. Depending on the provided credentials, different information will be provided.	To obtain the relevant information for the economic operator, credentials must be provided. These credentials will be evaluated on the endpoint receiving the request. Depending on the result of the evaluation (credentials valid or credentials invalid), the system will reply with a specific set of data (see step 6). This request will most likely embed credentials, as described in sections [4.2.2.3], [4.2.2.4], [4.2.2.5] and [4.2.2.6]. Depending on the provided credentials, different information will be provided.
6	The IT system receives machine-readable data from multiple data sources identified by the links.	This step is the response to the request of the previous step and is described in sections [4.1.2.3], [4.1.2.4], [4.1.2.5], and [4.1.2.6]. depending on the credentials provided.	This step is the response to the request of the previous step and is described in sections [4.2.2.3], [4.2.2.4], [4.2.2.5], and [4.2.2.6]. depending on the credentials provided.
7	The IT system processes the data received and presents the relevant data to the user.	This local step is not related to the DPP system architecture. It relies on the economic operator's IT system.	This local step is not related to the DPP system architecture. It relies on the economic operator's IT system.

5.1.3 User story 3: A stakeholder (e.g. the end user or someone who wants to access the data e.g. end customer, data consumer etc.) gets DPP data by scanning a QR code with their mobile phone

Step	*User Story Step*	*Validation using HTTP URIs*	*Validation using DIDs*
1	The user reads the DPP data. It is assumed that: - The user starts a DPP-capable app on their mobile phone. (e.g., In case of a server-based app, or that - The user starts the QR-enabled camera on their mobile phone). - The user uses other technologies Remark: Ideally, no vendor specific app on the phone should be needed to access basic information. The user scans a QR code containing a product identifier encoded into a web link with their mobile phone.	This step describes the action of scanning a data carrier to obtain the unique product identifier. It is part of the data flow described in section [4.1.2.1] From Data carrier to a usable URI. The app can obtain a usable URL or must construct one with the same flow as [4.1.2.1].	Described in [4.2.2.1].
2	The app uses the URL from the data carrier (e.g., the QR code) to request all available links from the resolution service component. The service component can be managed by the economic operator or a service provider acting on their behalf. The app can run locally on the mobile phone or be a server-based web app.	In this step, once the app obtained a usable link, a request will be sent to the Resolver to get all available links for the scanned product identifier. Described in section [4.1.2.1] From Data carrier to a usable URI.	Described in [4.2.2.1].
3	The resolution service component responds back to the app with a list of links and their associated link types.	The app receives all the typed links known by the resolver for the scanned unique identifier. This step is in the same data flow as step 2: section [4.1.2.1] From Data carrier to a usable URI.	Described in [4.2.2.1].
4	If the product is identified through serial number, and the user is interested in all data related to the product (including downstream activities) the data on the	**This step requires clarification.**	**This step requires clarification.**

	item related digital link contains the data, the app shows such data (provided by repairers/refurbishers, etc.).		
5	The app selects the relevant links and sends queries to them.	If the user scanning the DPP is a consumer, then no credentials are required, then flow is described in section [4.1.2.2] The default (consumer) data flow.	If consumer (no credentials) [4.2.2.2].
6	The data sources receiving the request determines the appropriate access level of the querying party and the app receives machine-readable data from multiple data sources identified by the links.	This step is the reply to the request sent in the previous step. It is therefore described in section [4.1.2.2] The default (consumer) data flow (this step is performed as many times as there is data sources).	If consumer (no credentials) [4.2.2.2].
7	The app processes the data received and presents the relevant data to the user.	If the user is using a specific app, then it is the responsibility of the app to handle the data format received. The endpoints providing the data should use a known data format to send back the results.	If the user is using a specific app, then it is the responsibility of the app to handle the data format received. The endpoints providing the data should use a known data format to send back the results.

5.1.4 User story 4: A component of the product (with instance level ID) is replaced by the original economic operator

Step	*User Story Step*	*Validation using HTTP URIs*	*Validation using DIDs*

1	The supplier adds information about the replacement to the instance level data about the product.	In this step, a supplier, mandated by the responsible economic operator to repair the product, adds new information in the DPP, at the instance/item level. This addition can be due to a part replacement, or a mandatory maintenance (same as with a vehicle). In this case, the repairer must be able to prove its identity, as a mandated repairer, to the REO IT systems, in order to add new data to the DPP KG. This flow is described in section [4.1.2.4] A role-based data flow – Repairer.	This flow is described in section [4.2.2.4] A role-based data flow – Repairer.

5.1.5 User story 5: A component of the product is replaced by the another (independent) stakeholder (e.g. repair company) acting on its behalf

Step	*User Story Step*	*Validation using HTTP URIs*	*Validation using DIDs*
1	The repair shop adds information about the replacement to the instance level data about the product in the original economic operator's DPP publication system.	This data flow is described in section [4.1.2.4] A role-based data flow – Repairer. Same as US4, but it is an external actor.	This data flow is described in section [4.2.2.4] A role-based data flow – Repairer. Same as US4, but it is an external actor. Details about authentication in data flow [4.3.2] and [4.3.3].

5.1.6 User story 6: A used product is collected and sorted. In a sorting process, it is evaluated if the product item is suitable for re-commerce, repair, upcycling, refurbishment or recycling. Dynamic data is added to prepare for one these next steps

Step	*User Story Step*	*Validation using HTTP URIs*	*Validation using DIDs*
1	Read product identifier: The sorter scans the product's data carrier (e.g., QR code or more likely the RFID tag) and downloads all relevant data needed for recommerce, repair, upcycling, refurbishment or recycling. The data sources recognise that the request comes from an authorised sorter and makes the relevant product data available. Details of this process is described in user story 3, steps 1-6. The product may be identified at model, batch or item level, depending on the requirements specified in the relevant delegated act.	This step is described in two sections [4.1.2.1] From Data carrier to a usable URI and [4.1.2.4] A role-based data flow – Repairer & Update of the DPP.	This step is described in two sections [4.2.2.1] From Data carrier to a usable URI and [4.2.2.4] A role-based data flow – Repairer & Update of the DPP. Authorization: [4.3.2].
2	The sorter uses the data combined with a condition evaluation of the product to decide the most appropriate action (refurbishment, remanufacturing, upcycling or recycling). If recycling is the most appropriate, the data is used to manage their recycling process. In other cases, one of use cases 4, 7 and 8 is activated.	This is a local flow on the recycler IT system. The decision is based on the DPP available data, but the DPP system may not provide this functionality.	This is a local flow on the recycler IT system. The decision is based on the DPP available data, but the DPP system may not provide this functionality.
3	The sorter optionally adds dynamic data about the product such as: 1) The condition of the product and its components (e.g., like new), 2) Current photos for recommerce, 3) Changes made such as repair, cleaning etc., 4) Take-back place and time.	This step is described in section [4.1.2.4] A role-based data flow – Repairer & Update of the DPP.	This step is described in section [4.2.2.4] A role-based data flow – Repairer & Update of the DPP.

4	Retrieve recycling information from current DPP holder. Potentially market surveillance and/or the economic operator receive data as described in user story 3 (A stakeholder gets DPP data e.g., by scanning a QR code with their mobile phone) and user story 9 (market surveillance and customs consume DPP data).	**This step requires clarification.**	**This step requires clarification.**

5.1.7 User story 7: An economic operator other than the original one takes over responsibility for the product, for example after refurbishment or remanufacturing

Step	***User Story Step***	***Validation using HTTP URIs***	***Validation using DIDs***
1	The independent refurbisher scans the product's QR code and downloads all data related to the product. The data sources receiving the request determines the appropriate access level of the querying party. Please refer to use story 2 and 3 for details. Remark: It is assumed that for product categories for which this user story may apply, there are requirements by delegated acts that all products are identified at item (instance) level when placed on the market, thus avoiding the addition of serialized identification by the downstream economic actor (for example refurbisher).	[4.1.2.4] A role-based data flow – Repairer & Update of the DPP.	[4.2.2.4] Role-based Data Flow – Repairer & Update of the DPP.
2	The independent refurbisher creates new product information using the data from step 1 and adds/changes data reflecting the refurbishment (new parts). Option 1 for Step 2: If the	Option 2 is described in [4.1.2.4] A role-based data flow – Repairer & Update of the DPP. Options 1 and 3 are described as part of [4.1.2.6] Role-based	Option 2 is described in [4.2.2.4] A role-based data flow – Repairer & Update of the DPP. Options 1 and 3 are described as part of [4.2.2.6] Role-based

	refurbishment would lead to need of a new product ID, then a new data carrier must be on it. Option 2: If the product is already identified at instance level by the original economic operator (e.g., product identifier + serial number), the product may keep the same identity, but the refurbisher will register one or more links in a resolution service component under their control enabling the link to the new data set. The QR code on the product remains unchanged. Option 3: If the product is only identified at product or batch level, the refurbisher can: a) Create a new serial number to be used in combination with the original product identity to provide an instance identifier, b) Create a new product and serial number. In this case the original product identifier becomes part of the DPP content for the refurbished product. The viability of Option 3b would be influenced by whether or not products subjected to recording of refurbishment need instance-level identification upon market introduction. Both option 3 a&b requires a new data carrier (QR code) on the product.	Data Flow – Remanufacturer.	Data Flow – Remanufacturer.

3	The independent refurbisher makes new entries in their resolution service components containing links to the DPP of the original product (at instance level).	Flow described in [4.1.2.6] Role-based Data Flow – Remanufacturer.	Flow described in [4.2.2.6]. Role-based Data Flow – Remanufacturer.
4	The independent refurbisher registers the instance level product identifier including the web URL for the resolution service component under their control in the Government centralised registry.	[4.1.2.4] Role-based Data Flow – Repairer & Update of the DPP.	[4.2.2.4] Role-based Data Flow – Repairer & Update of the DPP.
5	The independent refurbisher optionally reports to the original economic operator that they have refurbished the product and taken over responsibility for the product.	When sending the updated information in the DPP KG, the refurbisher can link a report with the data. It is up to the OEO to check the report. Receiving the updated data can also trigger a verification of the DPP data on the OEO side before performing the update. [4.1.2.4] Role-based Data Flow – Repairer & Update of the DPP.	[4.2.2.4] Role-based Data Flow – Repairer & Update of the DPP.
6	The independent refurbisher places the refurbished product on the market.	This flow is outside of the DPP system.	This flow is outside of the DPP system.

5.1.8 User story 8: A product is disassembled, and the material is recycled. An economic operator uses information in the DPP to change the design of their products

Step	***User Story Step***	***Validation using HTTP URIs***	***Validation using DIDs***

1	The recycler scans the combined product data through a Data Carrier (e.g., QR code or RFID tag) and downloads all relevant data needed for recycling management. The data sources recognise that the request comes from an authorised recycler and makes the relevant material and disassembly data available. Technically, this is the same as User story 2 above.	Global flow for this user story is described in [4.1.2.3]. See [4.1.2.1] for more details	Global flow for this user story is described in [4.2.2.3]. See [4.2.2.1] for more details
2	The recycler uses the data combined with a condition evaluation of the product to decide the most appropriate recycling activity and the data is used to manage their recycling process. In other cases, one of use cases 4, 7 and 8 is activated. Remark: a more detailed description is in user story 6.	This is a local flow on the recycler IT system. The decision is based on the DPP available data, but the DPP system may not provide this functionality.	This is a local flow on the recycler IT system. The decision is based on the DPP available data, but the DPP system may not provide this functionality.
3	Once the old product is no longer valid, the data can be deleted.	The deletion of the DPP is a subject that must be clarified. When raw materials are recycled, should the old DPP be preserved, and linked to the new one, for traceability purposes? How long a model level DPP should be available? The old DPP can be invalidated, this is described in [4.1.2.6] Role-based Data Flow – Remanufacturer	The deletion of the DPP is a subject that must be clarified. When raw materials are recycled, should the old DPP be preserved, and linked to the new one, for traceability purposes? How long a model level DPP should be available? The old DPP can be invalidated, this is described in [4.2.2.6]. Role-based Data Flow – Remanufacturer
4	The recycler optionally reports to the economic operator that placed the product on the market. a) The product ID. b) The condition of the product and its components (for example in order to improve the design of the product model).	When sending the updated information in the DPP KG, the refurbisher can link a report with the data. It is up to the OEO to check the report. Receiving the updated data can also trigger a verification of the DPP data on the OEO side before performing the	When sending the updated information in the DPP KG, the refurbisher can link a report with the data. It is up to the OEO to check the report. Receiving the updated data can also trigger a verification of the DPP data on the OEO side before performing the

	c) An offer to buy back the components and recycled materials. Remark: In the battery regulation, Article 65 (6b) mentions: Should the "passport" then cease to exist, that means there should be some mechanism to delete / remove the passport out of circulation.	update. [4.1.2.6] Role-based Data Flow – Remanufacturer	update. [4.2.2.6]. Role-based Data Flow – Remanufacturer

5.1.8.1 User story 9: A stakeholder (e.g., market surveillance) and stakeholder (e.g. Customs) consume DPP data

Step	*User Story Step*	*Validation using HTTP URIs*	*Validation using DIDs*
1	In their IT system, the market surveyor selects product identifiers that they want to get data about from their product ID registry.	Not specifically described in [4.1.2.5] A role-based data flow – Authorities.	Not specifically described in [4.2.2.5] A role-based data flow – Authorities.
2	The IT system gets all product identifiers from product ID registry embedded in web URLs	If the ID registry is the EU registry of D3.2, then this step is not described. In D3.2 Authorities already have the list of IDs.	If the ID registry is the EU registry of D3.2, then this step is not described. In D3.2 Authorities already have the list of IDs.
3	The IT system uses the web URLs from the list to request all available links associated with the product identifiers from the resolution service component, including the links registered by downstream actors such as repairers and refurbishers.	This flow is described in [4.1.2.3] Role-based Data flow – Recycler. Batch requests are supported.	Not described for the DID scheme but should be supported to. The alternative for the DID case is to provide a list of DIDs.
4	The resolution service component(s) responds back to the IT system with a list of links and their associated link types for each product identifier.	This step is described in flow [4.1.2.3] Role-based Data flow – Recycler.	This step is described in flow [4.2.2.3] Role-based Data flow – Recycler.
5	The IT system selects the relevant links and sends queries to them.	This step is described in flow [4.1.2.3] Role-based Data flow – Recycler.	This step is described in flow [4.2.2.3] Role-based Data flow – Recycler.
6	The data sources receiving the request determines the appropriate access level	[4.1.2.3] Role-based Data flow – Recycler,	[4.2.2.3] Role-based Data flow – Recycler,

	of the querying party and the IT system receives machine-readable data from multiple data sources identified by the links.	[4.1.2.4] Role-based Data Flow – Repairer & Update of the DPP and [4.1.2.6] Role-based Data Flow – Remanufacturer. This only difference between these data flows is the data that is received, which depends on the provided credentials. For Authorities, they should have access to every DPP mandatory data, and potentially more.	[4.2.2.4] Role-based Data Flow – Repairer & Update of the DPP and [4.2.2.6] Role-based Data Flow – Remanufacturer. This only difference between these data flows is the data that is received, which depends on the provided credentials. For Authorities, they should have access to every DPP mandatory data, and potentially more.
7	The IT system processes the data received and presents it to the user.	Not described in D3.2, application dependant.	Not described in D3.2, application dependant.

5.1.8.2 User story 10: An economic operator that has placed products on the market goes out of business

Step	*User Story Step*	*Validation using HTTP URIs*	*Validation using DIDs*
1	Before closing the company, the economic operator that has placed products on the market transfers all their DPP data to a DPP backup service provider. Comment 1: Optionally this step can be done continuously. Comment 2: If the economic operator placing the products on the market uses a DPP service provider, the same company may operate as a DPP backup service provider, too.	No data flow describing how the transfer of data is handled	No data flow describing how the transfer of data is handled

2	The DPP backup service provider takes control over the internet domain name used by the economic operator to direct DPP queries to their DPP data. Alternative Option: In case of 404, a new prefix is applied, and the URI is transformed in a standardized way to point into the archive of the resolver of the archive (as for an internet archive).	This step is (partly) outside of the DPP system (DNS transfer). For the prefix, more explanation is needed. Data flow is not represented in D3.2.	The DID document owner must be changed.
3	The DPP backup service provider restores the received DPP data or a subset containing only the mandatory DPP data. Remark: In case that the backup service provider goes out of business, there may be need of an additional layer of "ultimate" backup (e.g. by private company or official office).	Data flow is not represented in D3.2.	Data flow is not represented in D3.2.
4	The DPP backup service provider activates the correct internet domain name with appropriate links to the DPP data.	Data flow is not represented in D3.2.	Data flow is not represented in D3.2.

5.2 Results of the validation of the DPP System Architectures

For both architectures, this section presents what is covered, and what is not. In their current description, both architecture support essential features for the DPP system: creation, read (single DPP or batch), update and deletion of the DPP data are described. However, some operations and components of the architecture are not detailed in the current propositions.

The backup/archive of the DPPs is not described in the proposed architectures. This choice was made to give more details about the functions that must support the architectures to deliver DPPs in 2027. However, it will be critical in the upcoming DPP project to address this point to avoid losing data in a real-world scenario. Data transfer between REO and backup service provider is a process that depends on many points that can lead to different transfer processes. The use of domain names as part of the unique identifier calls for a great caution and foresight to avoid losing a domain name during a transfer. The backup process also depends on the implemented architecture (updating the DID document vs. updating the resolver).

The deletion of a DPP must be clarified. Currently, an open question remains: should DPP data be deleted? If the focus of the DPP is market use, when a product with an item level DPP is destroyed (e.g., recycled), its DPP data could be deleted. This leaves open the question of the deletion of DPP

data for products with a model level DPP. However, by taking into account traceability (UNECE Recommendation No. 49), linking the DPP data of the recycled product to the obtained raw materials could have an importance to obtain a whole chain of traceability. Currently, both architectures can support deletion of DPP data, but they also propose an invalidation of a product's DPP to avoid modifications while keeping the data accessible.

5.2.1 Specific points for the DPP System Architecture using HTTP URIs

Using the HTTP URIs based architecture, access rights are one the missing points that must be clarified to use this architecture in a real-world scenario. Different technical solutions can be proposed to cover this aspect (e.g., VCs and X.509) but an assessment must be performed to choose the most appropriate one regarding the readiness of the technologies, the ease of integration in the architecture, and the ease of use by REO and service provider.

5.2.2 Specific points for the DPP System Architecture using DIDs

With the DIDs based architecture, access rights are described, and already built in the technologies used to setup such a DPP system. Verifiable credentials can be used in a centralised manner (issuance by a trusted authority) or a decentralised way (issuance by the REO). This gives more flexibility to the REO to adapt to new market authorities, e.g., new business wanting to access privileged DPP data.

6 References

[2DRetail] Next generation barcodes, https://www.gs1.org/industries/retail/2d-barcodes/explorer, [accessed 2024-01-26]

[ActivityPub] W3C ActivityPub https://www.w3.org/TR/activitypub/, (2018).

[AAS] Specifications define the software structure, interface and semantics of the Asset Administration Shell and thus create the basis for the standardized Digital Twin. AAS Specifications, https://industrialdigitaltwin.org/en/content-hub/aasspecifications

[Batteries Regulation] Regulation (EU) 2023/1542 of the European Parliament and of the Council of 12 July 2023 concerning batteries and waste batteries, amending Directive 2008/98/EC and Regulation (EU) 2019/1020 and repealing Directive 2006/66/EC, ELI: https://eur-lex.europa.eu/eli/reg/2023/1542/oj

[BDVA] The Big Data Value Association (BDVA) is an industry-driven organisation witha mission to develop an innovation ecosystem that enables the data-driven digital transformation of the economy and society in Europe. https://bdva.eu/

[Bizer2009] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. International Journal on Semantic Web and Information Systems, Tom Heath, Martin Hepp, and Christian Bizer (Eds.), 5, 3 (2009), 1–22. 2009.

[BDVA Sharing] Data sharing spaces and interoperability", BDVA discussion paper, December 2023. https://bdva.eu/downloads

[Deliverable 3.3] CIRPASS Deliverable D3.3 Identification Schemes v1.3, February 2024, https://cirpassproject.eu/wp-content/uploads/2024/02/D3.3_IdentificationSchemes-v1.3.pdf

[DID] Decentralized Identifiers (DIDs) v1.0, Core architecture, data model, and representations, W3C Recommendation 19 July 2022, https://www.w3.org/TR/did-core/

DIDComm] S. Curren, T. Looker, O. Terbu. DIDComm Messaging v2.1 Editor's Draft. https://identity.foundation/didcomm-messaging/spec/v2.1/

[DID-Rubric] https://www.w3.org/TR/did-rubric/

[DOI] https://www.doi.org/

[EBSI] European Blockchain Services Infrastructure https://hub.ebsi.eu/vc-framework/did

[eIDAS] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, ELI: http://data.europa.eu/eli/reg/2014/910/oj

[ESPR] Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL establishing a framework for setting ecodesign requirements for sustainable products and repealing Directive 2009/125/EC - COM(2022) 142 final https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52022PC0142 with its latest available version of 2023-12-19 after the Trilog on the Council's website at https://www.consilium.europa.eu/media/69109/st16723-en23.pdf

[Fdhila2021] Fdhila, Walid, et al. "Methods for decentralized identities: Evaluation and insights." Business Process Management: Blockchain and Robotic Process Automation Forum: BPM 2021 Blockchain and RPA Forum, Rome, Italy, September 6–10, 2021, Proceedings 19. Springer International Publishing, 2021.

[Gaia-X] Gaia-X is an initiative that develops, based on European values, a digital governance that can be applied to any existing cloud/ edge technology stack to obtain transparency, controllability, portability and interoperability across data and services. https://gaia-x.eu

[GDSO] Established in January 2022 the « Global Data Service Organisation for Tyres and Automotive Components », abbreviated to « GDSO », is an international non-profit association. https://gdso.org/

[GS1 link types] All link types defined in the GS1 Web Vocabulary https://gs1.org/voc/?show=linktypes

[GS1 Digital Link 1.1.2] GS1 Digital Link Standard, Release 1.1.2, Ratified, Nov 2022 https://ref.gs1.org/standards/digital-link/

[GS1 Digital Link 1.4.1] GS1 Digital Link Standard, Release 1.4.1, Ratified, July 2023 https://ref.gs1.org/standards/digital-link/

[Guarino1998] Guarino, N. ed. Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy (Vol. 46). IOS press. 1998.

[Heinz2019] Heinz ketchup sent buyers to a porn site (forgot to renew the domain), Sudonull.com 2019. https://sudonull.com/post/59405-Heinz-ketchup-sent-buyers-to-a-porn-site-forgot-to-renew-the-domain Retrieved 2022-11-25

[JSONLD] https://json-ld.org/

[Hoops2023] Hoops, Felix, et al. "A taxonomy of decentralized identifier methods for practitioners." IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), 2023

[IDSA] Dataspaces comprise relationships between trusted partners that are governed by the IDSA standard for secure and sovereign data exchange, certification and governance for business and industry across Europe and around the world. https://internationaldataspaces.org/

[IEC 61406-1] IEC 61406-1:2022 Identification Link - Part 1: General requirements. IEC, 2022-09-15 https://webstore.iec.ch/publication/67673

[IEC 61406-2] IEC 61406-2 ED1 Identification Link - Part 2: Types/models, lots/batches, items and characteristics. IEC, expected publication 2024 https://www.iec.ch/dyn/www/f?p=103:38:8271442881253::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:1452,23,112292 (website last accessed 21.02.2024)

[IEC 61360-Series]:

- [IEC 61360-1] IEC 61360-1:2017 Standard data element types with associated classification scheme - Part 1: Definitions - Principles and methods. IEC 2017-07-27. https://webstore.iec.ch/publication/28560
- [IEC 61360-2] IEC 61360-2:2012 Standard data element types with associated classification scheme for electric components - Part 2: EXPRESS dictionary schema. IEC 2012-10-02 https://webstore.iec.ch/publication/5381
- [IEC 61360-6] IEC 61360-6:2016 Standard data element types with associated classification scheme for electric components - Part 6: IEC Common Data Dictionary (IEC CDD) quality guidelines. IEC 2016-10-04 https://webstore.iec.ch/publication/25984#additionalinfo
- [IEC 61360-7] IEC 61360-7:2024 Standard data element types with associated classification scheme - Part 7: Data dictionary of cross-domain concepts. IEC 2024-01-25 https://webstore.iec.ch/publication/72956
- [IEC 62656-1] IEC 62656-1 ED2 Standardized product ontology register and transfer by spreadsheets - Part 1: Logical structure for data parcels. expected publication 2025. https://www.iec.ch/dyn/www/f?p=103:38:3476541954914::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:1345,23,112372 (website last accessed 21.02.2024)

[IEC TS 62443-Series]:

- IEC TS 62443-1-1:2009 Industrial communication networks - Network and system security - Part 1-1: Terminology, concepts and models. IEC 2009-07-30 https://webstore.iec.ch/publication/7029

- IEC TS 62443-1-5:2023 Security for industrial automation and control systems - Part 1-5: Scheme for IEC 62443 security profiles. IEC 2023-09-15 https://webstore.iec.ch/publication/67461
- IEC 62443-2-1:2010 Industrial communication networks - Network and system security - Part 2-1: Establishing an industrial automation and control system security program. IEC 2010-11-10 https://webstore.iec.ch/publication/7030
- IEC TR 62443-2-3:2015 Security for industrial automation and control systems - Part 2-3: Patch management in the IACS environment. IEC 2015-06-30 https://webstore.iec.ch/publication/22811
- IEC 62443-2-4:2023 Security for industrial automation and control systems - Part 2-4: Security program requirements for IACS service providers. IEC 2023-12-15 https://webstore.iec.ch/publication/67631
- IEC TR 62443-3-1:2009 Industrial communication networks - Network and system security - Part 3-1: Security technologies for industrial automation and control systems. IEC 2009-07-30 https://webstore.iec.ch/publication/7031
- IEC 62443-3-2:2020 Security for industrial automation and control systems - Part 3-2: Security risk assessment for system design. IEC 2020-06-24 https://webstore.iec.ch/publication/30727
- IEC 62443-3-3:2013 Industrial communication networks - Network and system security - Part 3-3: System security requirements and security levels. IEC 2013-08-07 https://webstore.iec.ch/publication/7033
- IEC 62443-4-1:2018 Security for industrial automation and control systems - Part 4-1: Secure product development lifecycle requirements. IEC 2018-01-15 https://webstore.iec.ch/publication/33615
- IEC 62443-4-2:2019 Security for industrial automation and control systems - Part 4-2: Technical security requirements for IACS components. IEC 2019-02-27 https://webstore.iec.ch/publication/34421

[IEC 63278 Series:]

- [IEC 63278-1] IEC 63278-1:2023 Asset Administration Shell for industrial applications - Part 1: Asset Administration Shell structure. IEC, 2023-12-14 https://webstore.iec.ch/publication/65628
- [IEC 63278-2] IEC 63278-2 ED1 Asset Administration Shell for Industrial Applications – Part 2: Information meta model. IEC, expected publication 2024 https://www.iec.ch/dyn/www/f?p=103:38:3476541954914::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:1250,23,109017 (website last accessed 21.02.2024)
- [IEC 63278-3] IEC 63278-3 ED1 Asset Administration Shell for Industrial Applications – Part 3: Security provisions for Asset Administration Shells. IEC, expected publication 2024 https://www.iec.ch/dyn/www/f?p=103:38:3476541954914::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:1250,23,109075 (website last accessed 21.02.2024)
- [IEC 63278-4] IEC 63278-4 ED1 Asset administration shell for industrial applications - Part 4: Use cases and modelling examples. IEC, expected publication 2025 https://www.iec.ch/dyn/www/f?p=103:38:3476541954914::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:1250,23,116472 (website last accessed 21.02.2024)
- [IEC 63278-5] PNW 65-1032 ED1 Asset Administration Shell for industrial applications – Part 5: Interfaces. IEC, expected publication 2026 https://www.iec.ch/dyn/www/f?p=103:38:3476541954914::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:1250,23,121674 (website last accessed 21.02.2024)

[IEC 63489] IEC 63489 ED1 DB - Common data concepts for smart manufacturing. expected publication 2025. https://www.iec.ch/ords/f?p=103:38:214986186657521::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:1452,23,111532 (website last accessed 21.02.2024)

[IEC 63538] IEC 63538 ED1 Lifecycle-events: information models and services. expected publication 2026. https://www.iec.ch/dyn/www/f?p=103:38:7217878881923::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:1452,23,119296 (website last accessed 21.02.2024)

[IMDS] The International Material Data System https://www.mdsystem.com/imdsnt/startpage/index.jsp

[Internet Archive] The Internet Archive, a 501(c)(3) non-profit, is building a digital library of Internet sites and other cultural artifacts in digital form. https://archive.org/about/

[ISO/IEC 15459] The ISO/IEC 15459 standard comprises several parts. Of these, the most relevant to this document is Information technology — Automatic identification and data capture techniques — Unique identification — Part 3: Common rules. ISO/IEC JTC 1/SC31 2014-11 https://www.iso.org/standard/54781.html

[ISO/IEC DIS 18975] Automatic identification and data capture techniques, Encoding and resolving identifiers over HTTP, https://www.iso.org/standard/85540.html

[ISO/IEC 29500-2] ISO/IEC 29500-2:2021 Document description and processing languages Office Open XML file formats Part 2: Open packaging conventions. ISO 2021-08 https://www.iso.org/standard/77818.html

[JSON-LD] JSON-LD 1.1, A JSON-based Serialization for Linked Data, https://www.w3.org/TR/json-ld11/

[KGBook] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, Antoine Zimmermann (2021) Knowledge Graphs, Synthesis Lectures on Data, Semantics, and Knowledge, No. 22, 1–237, DOI: 10.2200/S01125ED1V01Y202109DSK022, Springer, online available at https://kgbook.org/

[Linked-Data] Linked Data, Tim Berners-Lee 27 June 2006 https://www.w3.org/DesignIssues/LinkedData

[Mountantonakis2019] Mountantonakis, Michalis, and Yannis Tzitzikas. "Large-scale semantic integration of linked data: A survey." ACM Computing Surveys (CSUR) 52.5 (2019): 1-40.

[Named-Graph] Jeremy J. Carroll; Chris Bizer; Pat Hayes; Patrick Stickler. Named Graphs, Provenance and Trust. The Semantic Web — ISWC2004, Yokohama, Springer–Verlag, 2005, https://dx.doi.org/10.2139/ssrn.3199260

[ODRL] ODRL Information Model 2.2, https://www.w3.org/TR/odrl-model/ But also further work of the ODRL CG, https://www.w3.org/community/odrl/

[Otto2022] Boris Otto, Michael ten Hompel, Stefan Wrobel, Designing Data Spaces, https://doi.org/10.1007/978-3-030-93975-5

[Provenance] PROV-N: The Provenance Notation, W3C Recommendation 30 April 2013, http://www.w3.org/TR/prov-n/

[RDF 1.2] RDF 1.2 Concepts and Abstract Syntax, W3C Working Draft 21 January 2024, https://www.w3.org/TR/rdf12-concepts/

[RFC1738] Uniform Resource Locators (URL), T. Berners-Lee, L. Masinter, December 1994, obsoleted by RFC4266 & RFC4248 and updated by [RFC3986], https://datatracker.ietf.org/doc/html/rfc1738

[RFC 2231] MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations, N. Freed, K. Moore, November 1997, https://www.rfc-editor.org/rfc/rfc2231

[RFC 3986] Uniform Resource Identifier (URI): Generic Syntax, T. Berners-Lee, R. Fielding, L. Masinter, RFC3986 January 2005, https://www.rfc-editor.org/rfc/rfc3986

[RFC 3987] Internationalized Resource Identifiers (IRIs), M. Duerst, M. Suignard, RFC3987 January 2005, https://www.rfc-editor.org/rfc/rfc3987

[RFC 6596] The Canonical Link Relation, M.Ohye, J. Kupke, RFC6596, April 2012, https://www.ietf.org/rfc/rfc6596.txt

[RFC 6761] Special-use domain names. S. Cheshire, M. Krochmal. IETF RFC 6761 February 2013 https://www.rfc-editor.org/rfc/rfc6761

[RFC 7519] JSON Web Token (JWT). M. Jones, J. Bradley, N. Sakimura. IETF RFC 7519 May 2015 https://www.rfc-editor.org/rfc/rfc7519

[RFC 8288] Web Linking, IETF RFC 8288 October 2017, M. Nottingham, https://datatracker.ietf.org/doc/html/rfc8288

[RFC 9110] The Hypertext Transfer Protocol (HTTP): HTTP Semantics. IETF RFC 9110 R. Fielding, M. Nottingham, J. Reschke, https://www.rfc-editor.org/rfc/rfc9110

[RFC 9264] Linkset: Media Types and a Link Relation Type for Link Sets. E. Wilde, H. Van de Sompel. IETF RFC 9264 July 2022 https://www.rfc-editor.org/rfc/rfc9264

[SAML] Security Assertion Markup Language (SAML) V2.0 Technical Overview, https://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

[SHACL] Shapes Constraint Language (SHACL), Holger Knublauch, Dimitris Kontokostas. W3C Recommendation 20 July 2017 https://www.w3.org/TR/shacl/

[SPARQL] SPARQL 1.1 Overview, W3C Recommendation 21 March 2013, The SPARQL WG, http://www.w3.org/TR/sparql11-overview/

[SPECIAL] Scalable Policy-aware Linked Data Architecture For Privacy, Transparency and Compliance, EC project N°: 731601, from January 2017 to December 2019, https://specialprivacy.ercim.eu/

[TA-9-2023-0272] Amendments adopted by the European Parliament on 12 July 2023 on the proposal for a regulation of the European Parliament and of the Council establishing a framework for setting eco-design requirements for sustainable products and repealing Directive 2009/125/EC (COM(2022)0142 – C9-0132/2022 – 2022/0095(COD)), https://www.europarl.europa.eu/doceo/document/TA-9-2023-0272_EN.html

[TLS] RFC8446 The Transport Layer Security (TLS) Protocol Version 1.3, https://datatracker.ietf.org/doc/rfc8446

[W3C-VC] Verifiable Credentials Data Model v1.1, https://www.w3.org/TR/vc-data-model-1.1/

[XACML] eXtensible Access Control Markup Language (XACML) Version 3.0, https://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

[XML-Schema] W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation 5 April 2012, https://www.w3.org/TR/xmlschema/

[ZVEI] ZVEI Discussion Paper DPP 4.0: An Architecture Proposal for a DPPSystem to implement the EU Digital Product Passport for Industrial Products. ZVEI 2023 https://www.zvei.org/fileadmin/user_upload/Themen/Industrie/PCF%40ControlCabinet/20230411_Discussion_Paper_DPP40.pdf