A multi-DLT verifiable registry for the digital product passport of circular ICT devices

Leandro Navarro, Javier Cano Esteban, Marc Font Miralles David Franquesa Griso
Universitat Politècnica de Catalunya, Barcelona

Abstract

Products in the market, specifically ICT devices, have a significant environmental footprint that needs to be reduced to improve sustainability. Furthermore, the availability of digital product information and its quality, in terms of reliability, integrity, and verifiability, helps different stakeholders make informed decisions to choose and increase the circularity of these products.

The digital product passport is a digital twin for device management that provides detailed and trusted information throughout the product's lifespan that can comprise multiple use phases and changes. ICT devices have a long lifespan and usually keep a unique chassis hardware identifier while adding or replacing parts over the lifespan as part of product usage and maintenance. Recording proofs about key information and actions on a device (such as registration, reconfiguration, data wipe, repair, transfer, and recycling), supported by attestations and documents, brings accountability and verifiability.

We aim to validate a multi-DLT registry to record verifiable, document-supported proofs for the hardware configuration of ICT devices over a circular lifespan with multiple owners. This registry can rely on different distributed ledgers to record these proofs. The registry has been integrated with the open-source DeviceHub inventory system, which calls the registry API to record proofs. As a result, device identifiers can be looked up to find the digital product passport for each device configuration.

To validate our design and development, we have developed two DLT drivers, one for an Ethereum permissioned PoA network and another for the IOTA DLT with channels. We have run tests to verify the verifiable registry API can be implemented and works correctly in both cases.

We confirm that our DLT-agnostic registry API can complement device inventory services to record verifiable proofs about device milestones. Decentralised identifiers allow cross-checking the proofs retrieved from the registry with the product information details collected from the lookup of inventories from manufacturers, owners, repairers, recyclers, etc. That results in digital product passports that represent a trusted digital twin for each device over its complete lifespan.

Introduction

Digital technology can be part of the environmental solution by bringing efficiencies to human activities, we call it digitalisation and digital transformation, but it is also part of the environmental problem. As the e-waste monitor [EWM20] reports, digital devices become e-waste after a shorter or longer lifespan. In 2019, the world generated 53.6 million metric tons (Mt) of e-waste. Only 17% were officially documented as properly collected and recycled, with only 7% returning to factories as secondary/recycled materials. Unfortunately, most e-waste (83%) disappears from data sources and statistics, but it is somewhere. We need a digital transformation of the management of digital devices, supported by good quality digital data, to improve and make digital devices more sustainable.

The circular economy (CE), and the term circularity, is about "designing out waste and pollution, keeping products and materials in use, and regenerating natural systems"

[EMF21]. For ICT goods, circularity translates into the need to design more circular electronics, and digital products in our scope. That means using more recycled and recyclable content, supporting circular business and ownership models that include reuse, repair, refurbishment or remanufacturing, end-of-life collection and high-quality recycling, and more circular e-waste or value chain management.

Having good quality information related to sustainability and specifically about circularity in digital and standardised format can bring qualities, facilitate and improve many processes, and allow citizens, organisations, and governments to assess their environmental footprints and other statistics about the digital/ICT sector. That is a central topic of the idea behind the “product information sheets”, also called “digital product passport” (DPP) or “digital twin”, to provide access to product information as part of a sustainable digital transformation of society [ITUT21][EC21].

The digital counterpart of a material product requires good-quality information. It is not only about details but also about validity: correctness, up-to-date and verifiability. As with the formal economy, formal sustainability requires relying on accurate and verifiable information from well-kept records and documents.

Digital devices, such as computers, phones, or any networking device, change over their lifespan as a result of hardware changes due to maintenance, reconfiguration or repair. In consequence, their DPP twins have to reflect that.

There is environmental information and sustainability-related information that helps make decisions about a product during its lifespan. However, this information should protect personal data privacy and business data confidentiality while ensuring credibility and usefulness.

In summary, we need standardised ways to share linked data about product items related to participants related to (traceable) specifications (design), materials, parts, products, flows (as business processes), decisions with outcomes (e.g., production, sale/purchase, transfer, disposition). That data has to be in digital form, accessible to the relevant actors. It has to be trustable (integrity, verifiability) and comprehensive (composable, traceable). The information and its properties facilitate informed and efficient decision-making and the assessment of impacts, all that being scalable to global markets.

These digital devices not only have associated digital data. They can generate data from internal sensors, including listing and checking the status of internal parts, and process and communicate information.

We have designed a DPP system combined with a verifiability service for the circular management of ICT devices over their lifespan that allows recording proofs about actions by actors with references to supporting documents that allow for further verification [DPP-DLT22].

In this paper, we propose and evaluate the design of a multi-driver verifiable registry for that DPP system, provided through an API, mapped to two distributed ledger technologies: an Ethereum permissioned blockchain and an IOTA ledger.

In Section II we analyse related work on the twinning of digital devices with digital data for circularity. Then, in Section III we present the system model. Section IV describes the implementation, followed by the validation in Section V. Finally, we discuss the results in Section VI with concluding remarks about the usefulness and generality of this multi-driver API in Section VII.

Related work

Several related works can be classified into four main categories: the circular economy of ICT devices, decentralised identifiers and the concept of a verifiable registry, distributed ledger technology for accountability, the digital product passport and digital twins.

Regarding the **circular economy**, we align with ITU-T L.1410 recommendation [L.1410] that defines the interlinked processes or stages that products, ICT goods, can follow during their lifecycle. These lifecycle processes or stages are raw material acquisition, production, use, and end-of-life, with subprocesses that we follow. On that, ITU-T has a series (L) of recommendations (standards) about the circular economy of digital devices that inspire this work, specifically Q7 in SG5, which focuses on “E-waste, circular economy and sustainable supply chain management”. The first author is a co-rapporteur of Q7.

eReuse is an initiative involving several social enterprises that collect and refurbish used computers and mobile phones donated by public and private organisations. After refurbishment, these devices are given to vulnerable citizens, supported by sponsors that cover the refurbishment cost and assist them in their use for social inclusion. eReuse has developed software tools that allow for more efficient (time, quality) processing of ICT devices: less refurbishment time per device, higher efficiency and quality of refurbishment, more digital data to manage these devices over their complete lifespan, and the ability to quantify and certify social benefits and environmental impacts [CAPC21].

The DeviceHub inventory service and Workbench tool developed by eReuse.org provide a comprehensive solution for managing and tracking information and communication technology (ICT) devices. These tools allow organisations to keep track of the location, status, and usage of their ICT devices, as well as manage repair and maintenance processes. The DeviceHub service utilises a centralised database to store information about ICT devices, and the Workbench tool run on devices produces self-generated hardware descriptions and identifiers, a device fingerprint. Several organisations have implemented these tools, including schools, government agencies, and non-profit organisations [ERE23]. They are similar to other inventory management systems, such as Asset Panda [AP23] or Freshservice [FS23], but with a specific focus on ICT devices and the needs of organisations that use and manage them. Other related work includes the open-source software AssetCloud [AC23], which also provides a platform for managing and tracking ICT assets. Overall, the DeviceHub inventory service adds to the growing field of inventory management systems with its unique focus on reusing and recycling ICT devices.

The World Wide Web Consortium (W3C) has been researching **decentralised identifier** (DID) technology to improve online identity management and enable secure, verifiable interactions. DIDs are self-sovereign identifiers that allow individuals and organisations to authenticate themselves online without relying on a central authority. W3C has published several technical specifications and recommendations related to DIDs, including the DID Core and the DID Resolver specifications [W3CD22].

In addition to DIDs, W3C has been researching verifiable registries to store and manage digital records securely. Verifiable registries use cryptographic techniques to ensure the authenticity and integrity of records, making them suitable for use in various contexts, including supply chain management, government services, and financial transactions. W3C has published several technical specifications related to verifiable registries, including the Verifiable Credentials Data Model and the Verifiable Claims Data Model specifications [W3CC22].

Other researchers have also explored the potential applications of DIDs and verifiable registries in various contexts. For example, Halpin [HAL20] explores different uses of DIDs and verifiable credentials in healthcare, identifying potential benefits and shortcomings in terms of data security and privacy protection. Similarly, Lauinger et al. [LAU21] explored the use of DIDs and verifiable credentials in the context of Self-Sovereign Identity Management, finding that they could improve the integrity and confidentiality against hostile network participants. Overall, the research on DIDs and verifiable registries suggest that these technologies can revolutionise how we manage digital identities and records online. More on identifiers, data model, system architecture in our work are described in [DPP-DLT22].

There has been significant research on applying **distributed ledger technology**, specifically Ethereum and IOTA, **to improve accountability** in various industries. For example, a study by Meeradevi et al. [MEE20] discusses the potential of using Ethereum to enhance supply chain transparency and traceability. The authors argue that by using smart contracts, Ethereum allows for real-time tracking of products and verification of authenticity, which can improve accountability in the supply chain. Additionally, a paper by Park et al. [PAR19] discusses the use of IOTA in the energy sector to provide accountability and transparency in renewable energy trading. The authors propose a system that uses IOTA's distributed ledger to track the generation and consumption of renewable energy, enabling consumers to track their energy consumption and incentivising renewable sources accurately. These studies demonstrate the potential of distributed ledger technology to improve accountability in various industries.

The **Digital Product Passport** (DPP) is a structured collection of product-related data with a predefined scope and agreed data ownership and access rights conveyed through a unique identifier. [GALA21]. Digital product passports are digital records that provide detailed information about a product's characteristics, performance, and lifecycle, including its telemetry data. Digital twins, on the other hand, are virtual representations of physical products that allow to keep track of relevant indicators about the physical product and therefore optimise their environmental performance, among other applications. For example, in a study by Walden et al. [WAL21], the authors explored the potential of using digital product passports for batteries in the ICT industry. They conclude suggesting that the digital product passport could be a central element of the digital circular economy and as such it needs to be developed further, ideally through a multi-stakeholder collaboration across the entire industry value chain. In addition, [GOTZ22] discusses how a well-designed DPP could have both short- and longer-term benefits, improving access to reliable and comparable product sustainability information for businesses, consumers and policymakers. According to Guth-Orlowski [GUTH21], digital product passports are a technical tool that can provide valuable information about the environmental and social impact of products and can be implemented through the use of blockchain technology and standardised data formats.

The International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) has been actively working on defining the requirements [GDS21] and data model [D4PI22] for a global digital sustainable product passport for ICT devices. This effort is aimed at providing a standardised approach for documenting the sustainability aspects of ICT products throughout their lifecycle. The first author of this paper is the editor of that standard.

In the realm of **digital twins**, several researchers have explored the use of sensing and updating dynamic information (telemetry) to improve the accuracy and utility of digital twins. For instance, Wang et al. [WAN19] proposed a system where a digital twin is linked and

updated with details from the physical product. The authors demonstrated the effectiveness of this approach in WEEE recycling, recovery and remanufacturing in the background of Industry 4.0.

Overall, the research on digital product passports and digital twins highlights the potential benefits of using these approaches to improve the transparency, efficiency, and sustainability of ICT devices.

The verifiable service combined with an inventory service allows twinning each device and concrete hardware configuration with their unique decentralised identifiers (DID). These DID allow access to digital product passports with details that reflect as a digital twin the changes in hardware configuration over the circular lifespan of an ICT device (chassis). In addition, DID allows the verification of claims by cross checking the details between the inventory and the registry service supported by DLTs.

Model

These processes that digital devices follow can be grouped as pre-use, use, post-use, and value chain (life cycle) management:

The **pre-use phase** is about supply-chain manufacturing and goods production. L.1410 processes include design, raw material acquisition, and ICT goods production. Design and manufacturing decisions determine the use of primary materials extracted from nature, and secondary materials, captured from e-waste. All this information can only come from the product supplier.

In the **use phase**, devices can change owner, location, and usage and get modified by a repair or reconfiguration. L.1410 processes include ICT goods procurement, sale, use, reuse, repair, modification and other support activities. During use, they consume energy, parts can be added or replaced, and they suffer from wear and tear and change during an expected long lifespan. In addition, ICT devices can generate data from internal or external sources (sensors) or connected to events. Finally, in the use phase, devices can be used and transferred for reuse until they are disposed of as no longer valid.

In the **post-use phase**, L.1410 processes include end-of-life treatment starts with the collection, and transport of de-installed ICT goods or support goods to storage, disassembly, parts reuse, dismantling, and shredding facilities. It ends with the recovery of materials, recycling of raw materials and final disposal of treatment of waste ICT and support goods [L.1410].

Circularity implies **life cycle management**, considering the value chain system, which facilitates knowledge generation across the value chain over a product life cycle to help keep resources at their highest value, preventing value and information leaks (such as documentation, traceability, and history).

All these data items associated with or generated during the lifespan of a device can be related/linked to **additional data and documents**. These miscellaneous details (multimedia content) can give credit and help in the accountability and verifiability of these processes, even motivating and rewarding human participants. These documents with details can be grouped as a “portfolio” per device (about all events and data in its lifespan), and per organisation (all devices owned and managed).

Details about the devices (data, documents) can be stored and updated as digital data in an organisational inventory system, but a device may have multiple owner organisations over an extended lifespan.

However we may expect to keep track of devices along their full lifespan to assess their circularity. In addition, to twin devices to their digital counterparts, devices need to have **unique identifiers**, either based on individual serial numbers (usually) or allocated by the device owner. We use name-based UUID (version 5 RFC4122).

The manufacturer, reseller, or first owner can register the chassis of a device on a ledger using a unique Chassis ID (CHID). It also publishes its first DPP, which refers to the initial detailed hardware configuration of the product that includes a unique Product Hardware ID (PHID) for that configuration. All DPPs can be located from the CHID, and each DPPs for a specific configuration can be located by an ID composed of CHID:PHID.

Devices can have **physical tags** with digital identifier codes. These tags act as data carriers to facilitate identification for tracking these assets during their lifecycle and twinning the material and digital counterparts. Commonly, these physical tags include a written identifier and a machine-readable, optical (e.g., QR code), or electromagnetic (e.g., RFID, NFC) element to facilitate reading.

Digital devices can be computers, mobiles, networking equipment, and sensor/IoT devices. These digital devices correspond to products with some capability of communication and processing: data input, output, and processing, therefore they can execute code to check their internals and report on their current internal configuration and status (**fingerprinting**).

In summary, devices generate repeatable fingerprints for the same device, resulting in unique identifiers that can be printed in physical tags attached to devices and point to a history of details in an inventory service as well as entries in a verifiable registry. This allows logging details that remain during the whole lifespan (chassis) and others that change as the chassis changes over use with observations about internal or external (environmental) sensors. Access to these logs is restricted to authorised participants and permissions. All together (inventory, verifiable registry, related documents) facilitate auditing (verification).

This model is generalisable to any ICT device, including IoT or network devices such as routers.

Which actors are involved: device operators that register devices and issue DPPs, witnesses that record observations with documents that provide testimony and verifiability, and verifiers that can audit/verify claims about details with facts.

We store details (metadata about devices, data produced by devices on several events along their lifespan, participants involved, supporting files that document event-related data) in the inventory cloud service, and the verifiable registry stores accounting and audit details with summaries of the documents and data involved as proofs, and identifiers for devices and participants involved, following agreed procedures (as smart contracts) and stored in an irreversible log. An audit can be performed by looking up the DLT entries associated with a device DID in the verifiable registry, and the details about that device DID in the inventory system, and confirming the data retrieved matches and matches the entries in the DLT, so the information is verified.

We record details about devices and data from the devices. This data comes from internal sensors (like a tachometer to record milestones such as usage counters, data wipe, and power cycles) or from external sensors, like readings from environmental sensors.

Our inventory system stores that data and the verifiable registry provides the proofs about when, who, what information and related events took place with integrity guarantees.

Therefore our verifiable registry acts as a globally unique, tamper-evident, immutable log according to the W3C model [W3CI21]. Figure 1 illustrates the main concepts and how they relate to each other.

In fact, a device may appear in multiple organisational inventories over multiple use phases in its lifespan, but its DID (chassis ID) will be the same for the same chassis, even if some components are added or replaced.

We want to separate the DLT concerns in implementing a verifiable registry, from a DLT agnostic interface and client who can interact with the verifiable registry in the domain of data verifiability. We also want to explore how generic our verifiable registry API is by developing multiple drivers for different DLTs and exploring the different design and implementation challenges raised by different DLT models.

Multi-driver does not imply federation (integration of data in different DLTs), only that a client can choose which DLT backend, by passing a context string selecting one and operating with it through the same API. (same abstractions/different implementation model).

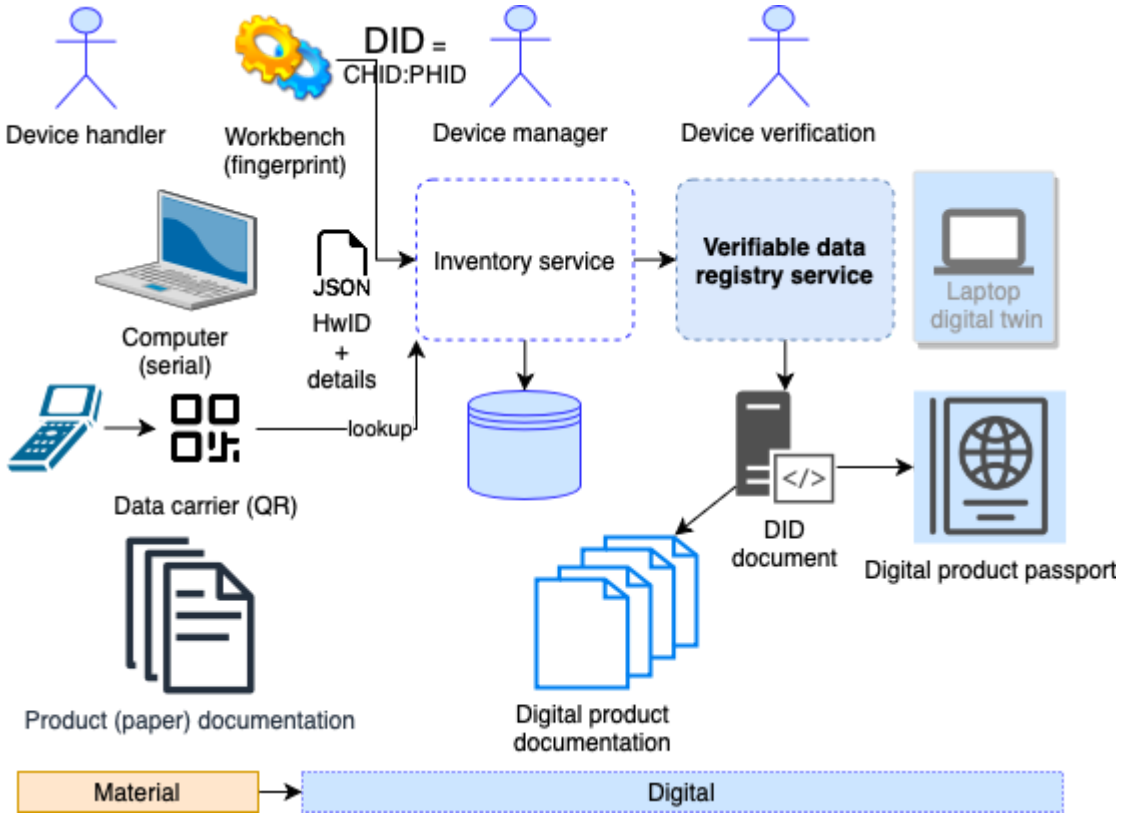

Figure 1. Conceptual diagram of the material (computer, data carrier, fingerprinting software) and the digital twin (inventory + verifiability information) offering a digital product passport collecting details (including digital product information).

Data in the verifiable registry is not stored in a reversible form. User identifiers are stored as numeric IDs (public keys), device identifiers are stored as version 5 UUIDs, and the documents are stored as summaries (hashes). Data is irreversibly recorded in the ledger (append-only).

The rules to record operations (proofs) are ideally implemented and run as smart contracts, running on a set of replicated servers, and therefore running inexorably and according to

agreed-upon rules. However, IOTA didn't yet have support for smart contracts, so that part of the driver was implemented as normal server code.

We aim to develop a verifiable registry that is W3C DID compatible, at least architecturally. For that we describe the implementation and then the validation to confirm that in the case where the registry has been integrated and called from the DeviceHub inventory service, the test and experiments confirm the registry operated as expected, satisfying the system requirements.

Implementation

The implementation of the verifiable registry is divided into two parts, the verifiable registry and the REST API. The verifiable registry can be implemented in any DLT capable of storing arbitrary data. The REST API, through different drivers that adapt to the specific DLT interfaces, can use any of them, providing the client with a common interface. In our case, we have used Ethereum and IOTA. This design doesn't necessarily imply federation, as clients can choose which backend DLT to use to record verifiability proofs.

The reason for supporting differently implemented verifiable registries may be that each technology used to implement them has different properties that may be of more or less interest to the client. For example, our verifiable registry implemented in Ethereum can use smart contracts, which enable us to enforce the procedures and data structures to be written (through code that runs on the DLT). IOTA doesn't yet have that capability, making the writing of unstandardized data a possibility.

In any case, data written in the verifiable registry must be associable with a particular device. This is easily done as devices must have unique identifiers. However, how this association is done varies between verifiable registries.

In Ethereum, we use a unique smart contract instance for every registered device. Any data associated with a particular device must be written through a call with that device's smart contract. We keep a factory smart contract to look for a device's smart contract through its unique identifier.

In IOTA, we use the IOTA streams functionality [STR23]. This allows us to create channels, which are data structures containing messages holding arbitrary data. Each registered device has its channel, and any data associated with that device must be written as a message to that particular device's channel. Similarly to Ethereum's factory contract, we keep an index channel to look for a device's channel through its unique identifier.

Because data written to a DLT can be read by any user with access to a node of the DLT, any sensitive data must not be written. In place, we propose that the written data acts as a proof. A proof consists mainly of a summary, a hash or signature, of other existing data stored elsewhere, including supporting documents, and any other valuable information. For example, additional information could help the user locate the original data.

By the properties of both systems, the author of any write can be easily checked, so any data auditor can decide which data can be trusted by checking its author and deciding if it's a trusted source. A credential and permissioning system support this decision. The credential system may be implemented and maintained inside the particular verifiable registry, making the own registry capable of verifying the credentials. In both driver cases, a root authority exists that emits the first credentials to other users, which then can form a tree-like structure.

For Ethereum, we use a smart contract to keep track of the credentials held by a particular user. This smart contract can be read by a device's smart contract, giving us the capability of permissioning the writes and reads to the latter. In this way, these permissions are enforced by the own verifiable registry when attempting to write or read.

For IOTA, we use their built-in verifiable credentials system. A user can hold a credential's data, which, when presented, can be verified by a method call to the registry. Since a call from outside the registry must invoke this verification, the permissioning then can only be enforced outside the verifiable registry. In other words, with IOTA this verification is not performed by a smart contract as trusted code, but instead done locally by the API provider code, and therefore more vulnerable to discretionary changes in behaviour, less trusted.

As said before, when used by a client, all these particularities of the different registries are leveraged to the REST API, which implements them through different drivers. The API then presents a common interface to the client. This interface consists mainly of methods to write and read data regarding a particular device and manage users' credentials.

To make the API as DLT-agnostic as possible, because each user must have a different identity (pair of keys) in each DLT, we have decided to manage those identities in place of the user. The identities are generated and given to the user when they register and are stored symmetrically encrypted by the API, with an API key also given to the user. The API does not store this API key. Still, the user is verifiable by presenting it (it is the full responsibility of the user to preserve it, and losing it means losing that identity). The API also holds any IOTA credential that the user should hold, in encrypted form using the client API key.

Overall, everything described above makes using the API very simple for a client. The client must register a user through a single call and then can invoke any other call by providing their API key and the identifier of which available verifiable registry they want to act on. Because they may not be able to do what they intend without a credential, they must first obtain the corresponding credential from another user. This user must have a credential to issue credentials to other users. These credentials can be issued by a single API call, indicating the target user.

Finally, since a verifiable registry is used, this system can comply with the W3C DID standard. A DID method can be and has been developed. The method-specific ID is mapped to the unique identifier of a device. This, in turn, is presented to a resolver that retrieves information about a device from a verifiable registry to construct a DID document using the developed API. This DID document includes, by default, references to other DID methods. For instance, the DID controllers use registry-specific DID methods like "did:ethr" and "did:iota". The DID document also includes a service to reference the location in the registry where proof data of that device may be read or written.

Implementation details

We describe the main implementation details about the API and the two developed DLT drivers, and the DID support.

API

The API stores necessary data about each user and provides a common interface to interact with the DLT.

Data storage

For each user, the API stores the following data:

- Salt
- Hash (of the user's API key + salt)
- Ethereum keys (encrypted)
- IOTA keys (encrypted)
- IOTA verifiable credentials (encrypted)

The encrypted data is symmetrically decrypted by the user's provided api key.

Interface

Composed of three types of calls.

- User management: registering, unregistering, issuing and revoking credentials.
- Device management: registering, writing proofs, reading proofs.
- DID support: writing and reading data related to building a DID document.

DLT drivers

Because each DLT works differently, specific code has to be written for each to implement the API calls. These drivers can be separated into two parts:

1. **Data management in the DLT**. The driver should be able to store and read data on the DLT. This data should consist mostly of information that acts as a proof to verify data stored elsewhere (hashes of the actual data).
2. **Management of permissions** to read and write data. The driver should be able to assign roles to the different users that dictate what they can and can't do and verify these roles with the DLT.

Both current driver implementations use pretty different techniques to satisfy these requirements.

Ethereum

For Ethereum, as Figure 2 shows, we use three different smart contracts. The first two only have a single instance deployed, while the third is deployed once per device:

- The first contract provides the **assignment of roles to Ethereum addresses** (users). An administrator user can give any other user the "issuer" role. An issuer can give any other role to any other user. This contract also provides a simple interface to check the role of any given user.
- The second contract is responsible for **deploying an instance of the third kind of contract for each device**. It is also responsible for storing an index that matches DIDs to their corresponding deployed contract's address. In addition, this contract can read the first contract to limit who can call its methods.

- The third contract is deployed for each device. This contract **stores information about the device**. Methods are provided to store information as proofs and retrieve them. This contract can also read the first contract to limit who can call its methods.

All in all, given a device's unique identifier:

1. A contract will be deployed to store its information.
2. An entry will be written in the second contract's index to map the identifier to its contract's Ethereum address for easy lookup.
3. Every operation will be limited to the role of the user that invoked it.

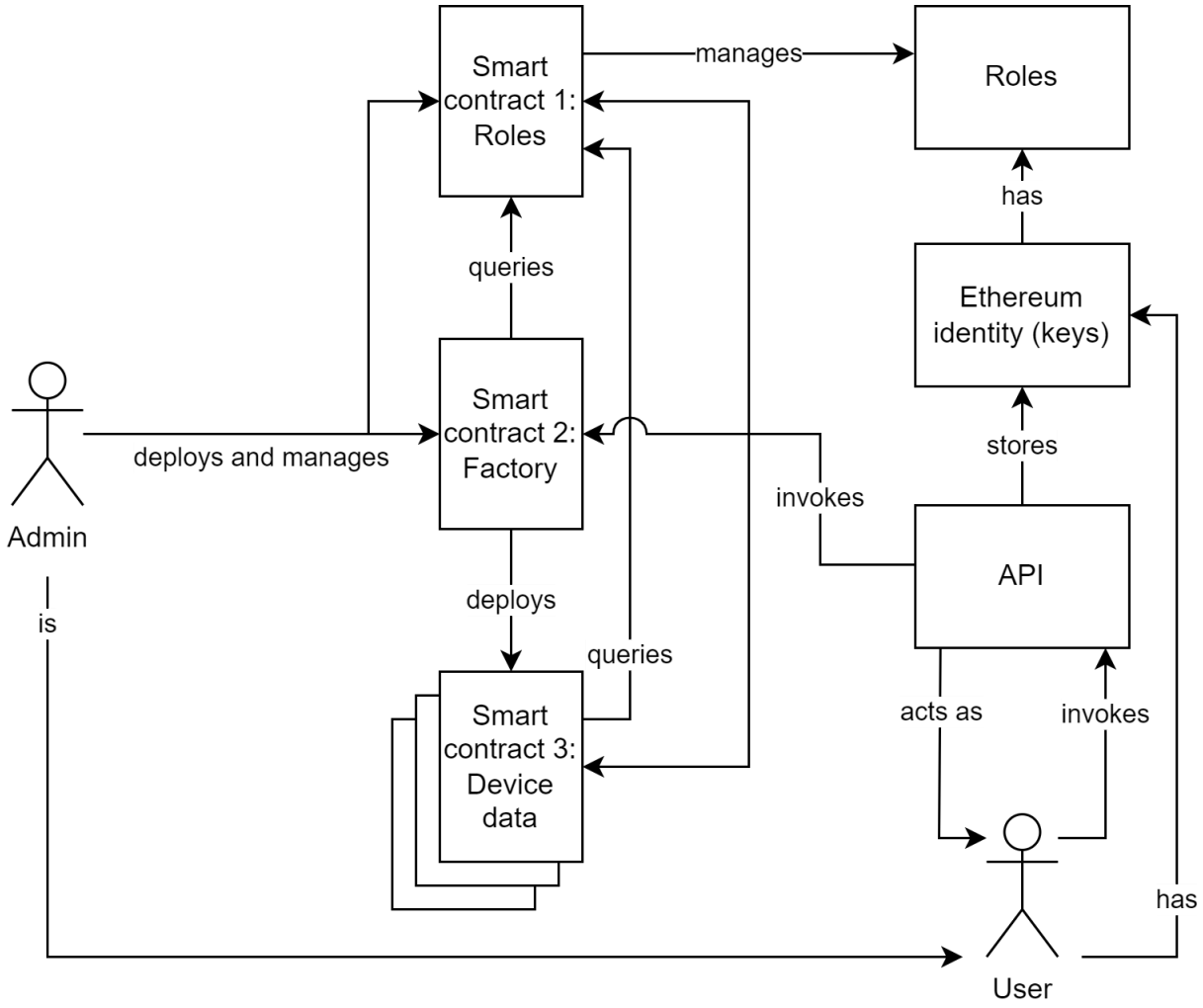

Figure 2. Conceptual diagram of the data model for roles and actions in the Ethereum driver.

IOTA

For IOTA, we use their IOTA Streams functionality [STR23]. This gives us access to all the tools we need for the implementation: identities, verifiable credentials and channels:

- **Identities** are the mechanism by which we identify the users. Equal to Ethereum, they have a public and private key pair, but they also add a DID to the mix. This means that an IOTA user can be identified by default with a DID, unlike Ethereum, where a user is just identified by their public address.
- A **verifiable credential** is some data structure that one user can release to another user, and the issuer of it can verify it. These credentials can be linked to a previous

credential, creating a chain of trust by default. They can contain arbitrary data, which we use to define roles.

- **Channels** are data structures that contain messages from users. The user that creates the channel can manage read and write permissions over it. We use a channel for each device to store information about it. We also use a unique channel as an index to store the relation between every device's unique identifier and its channel address. These channels allow us, similarly to Ethereum's smart contracts, to store data as proofs, but make it unfeasible to enforce the writing of specific data structures or to limit the user's capability to perform reads and writes automatically. This is because, unlike smart contracts, they lack the logic to check for the user's role or that the data structure given is valid.

Thus, the main difference from Ethereum is that the API itself has to:

- Validate the data structure to be written.
- Manually check (by calling the DLT) a user's role before allowing data to be written or read.

We have worked with the IOTA Foundation to develop a library that uses the IOTA integration services to care for it [INTS22]. This library can issue credentials in the same hierarchic way as roles are issued in the Ethereum smart contract. These credentials can be linked to a role through the previously mentioned arbitrary data. The library can also manage the index channel and every channel related to a device. All in all, as Figure 3 shows, given a device's unique identifier:

1. A corresponding channel is created to store its data.
2. An entry will be written to the index channel for lookup purposes.
3. Every action is limited by the role and verifiability of the presented verifiable credentials by the user starting it.

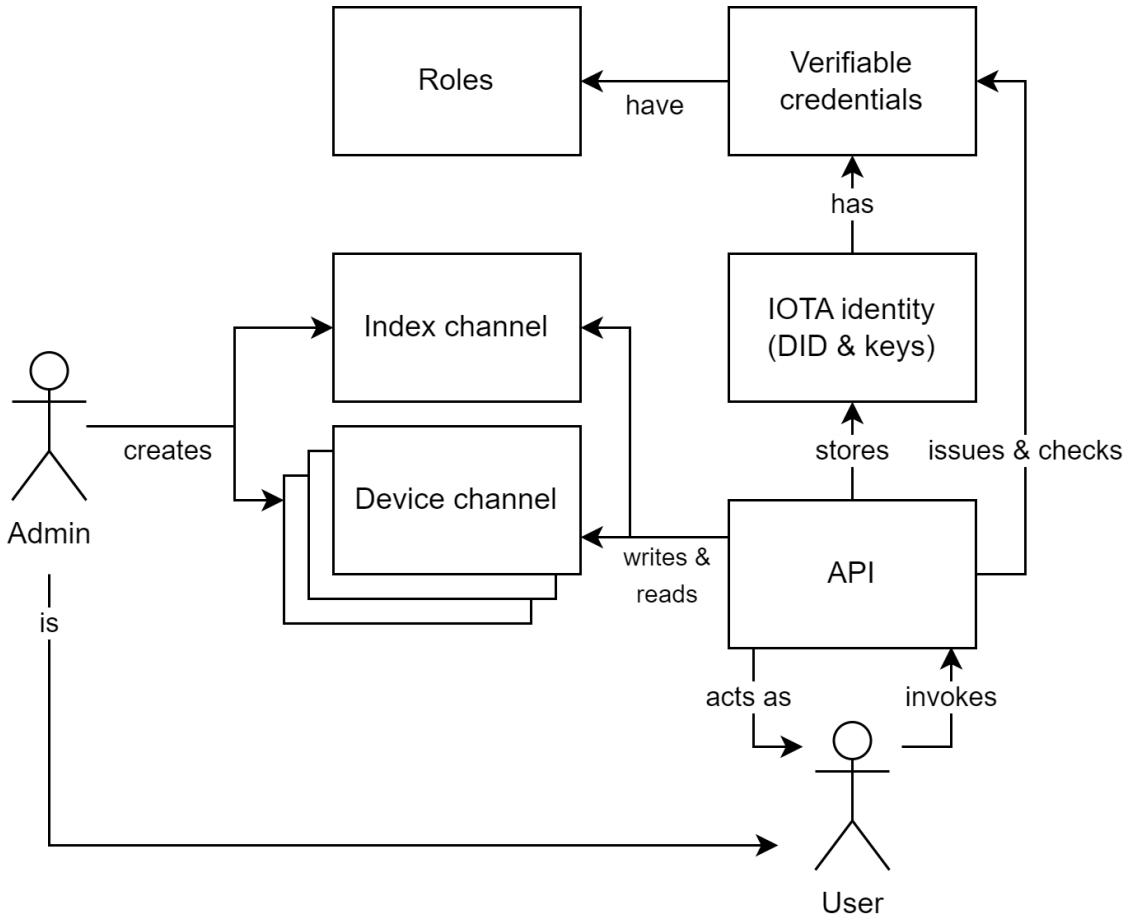

Figure 3. Conceptual diagram of the data model for roles and actions in the IOTA driver.

DID support

We have developed a DID method that maps unique device identifiers to DIDs. The DID format is the following:

- did:ereuse:<device's unique identifier>

By providing this DID string to our own developed DID resolver, a DID document is retrieved. This document contains the following:

- The document's controller. Usually, the DID of the device's owner. This DID uses the ethr-did method [EDID22] if the device is stored in Ethereum or the iota method [IDID23] if the device is stored in IOTA.
- A service with the document's location. Smart contract address if stored in Ethereum and channel address if stored in IOTA.

The Ethereum and IOTA DIDs can be resolved by their respective resolver. The structure of the DID system is shown in Figure 4.

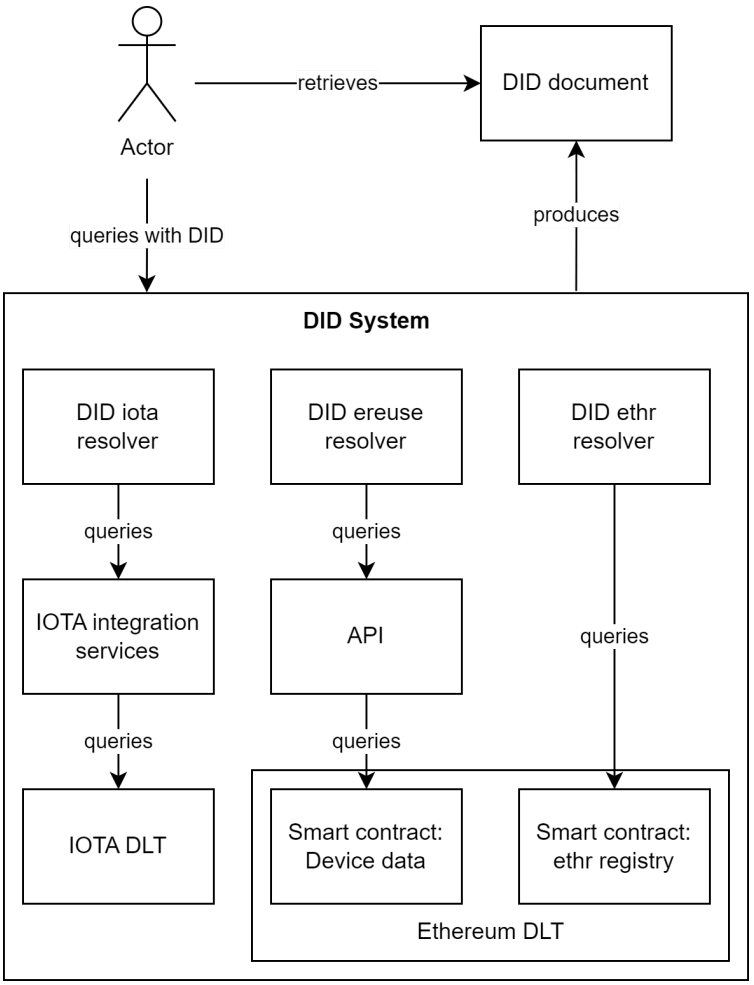

Figure 4. Conceptual diagram of the DID system.

Validation

The design of the registry API to satisfy the requirements to act as a verifiable registry for ICT devices is described in detail in [NAV22]. Accordingly, we have developed a set of software tests to validate our implementation of the drivers according to the API design.

While our tests do not pretend to be completely exhaustive, they aim to provide certainty that all the API calls and Smart Contract methods work as expected under a wide range of common scenarios or use cases, as well as validate the integration between the API and the Ethereum Verifiable Registry. We did similar tests initially with the IOTA driver with successful results, but not with the latest version due to the unavailability of the IOTA integration services in the last validation phase.

This validation has been achieved by executing each API call several times under different conditions and parameters and monitoring its behaviour and return data. Since all API calls interact with all Smart Contract methods, these methods have also been tested.

To illustrate a sample of some existing API calls validated, a user who registers a new device into the verifiable registry will be used as an example. The user (given the operator credential by another user) registers, issues a new Device Passport and generates new proofs about a device. The Verifier later reads all the proofs generated. This is represented in Figure 5.

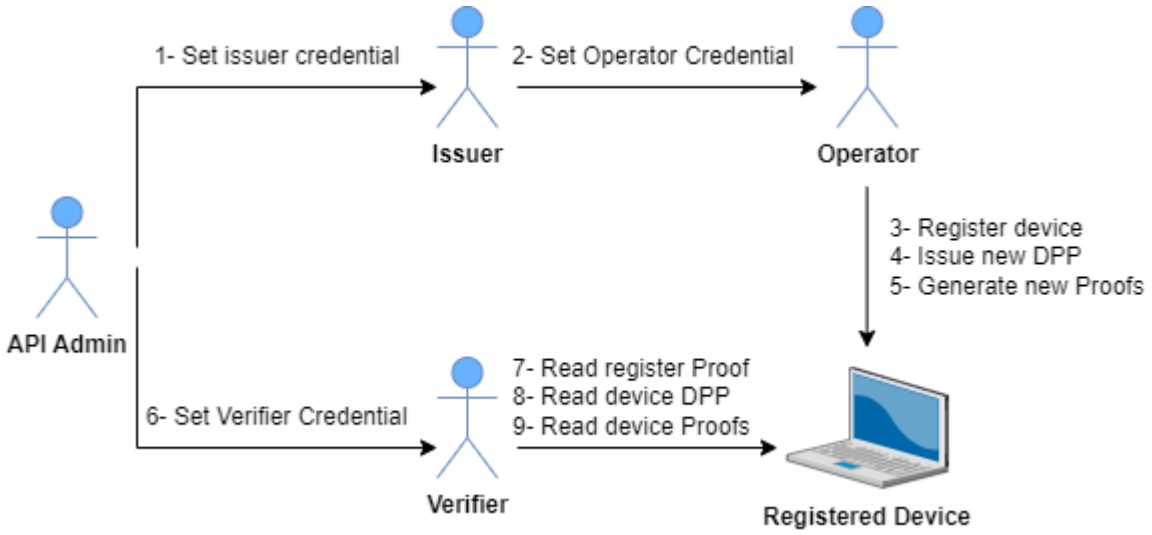

Figure 5. Conceptual diagram of a sequence of API calls done by multiple API users.

In our testing environment, tests are grouped by the API calls they validate. Two tests that validate the *"Set Operator Credential"* API call execute it using two different users:

- Test 1 executes *"Set Operator Credential"* with a user that owns an "Issuer" credential.
- Test 2 executes *"Set Operator Credential"* with a user that does not own an "Issuer" credential.

While "Test 1" is expected to return a successful "200" code from the API, "Test 2" is expected to return an error code and a message, since it executed the API call without the necessary credentials.

All the testing is done with Cucumber.js, a tool that lets us test our implementation as behaviour-driven development (BDD). All tests can be found in our repository [VRTST22]. These tests are:

- Register a new user to the API.
- Set user as Issuer.
- Issue credentials to an API user.
- Check API user credentials.
- Register a new device into a DLT.
- Issue a new DPP into a DLT.
- Store a new generic proof into a DLT.
- Deregister a device of a DLT.
- Transfer a device (change owner).
- Get all register proofs of a given device.
- Get all issue proofs of a given device.
- Get all generic proofs of a given device.
- Get all deregister device proofs of a given device.
- Get all transfer proofs of a given device.

Tests are grouped by functionality. Each functionality contains a group of “use cases” that execute a set of methods of the API and Smart Contracts to verify its behaviour. To implement this, we have used Cucumber, a testing tool that supports Behaviour Driven Development (BDD).

In our Cucumber framework, a functionality like “*Register a new user to the API”* is known as features, each with one or more scenarios (use cases). Each scenario will execute a list of steps (API calls and Ethereum Smart Contract methods). If one or more steps of any scenario do not return the desired data, we consider the scenario as failed. More details are in the repository documentation [VRAPI22].

Discussion

The system has been integrated with the DeviceHub[1] device inventory system. Hence, its backend calls the verifiable registry API to record proofs about new devices (so-called chassis with a unique hardware ID), about each detailed hardware configuration detected with a specific set of components or parts (a DPP), and proofs for certain actions on the device (such as data wipe, repair, transfer, recycling). As a result, DeviceHub delivers a device-centric (chassis) view and a given configuration (DPP) view that contains not only details about serials and components but also verifiability information about these hardware details, in terms of supporting documents, timestamps, and participants.

The validation of the verifiable registry with its API, and its integration with the DeviceHub system, allows us to confirm that the API calls to our Ethereum smart contracts with the Ethereum driver, as well as for the IOTA driver, work as expected, and can keep track of device identifiers and all related proofs. Therefore the API and each of the drivers can deliver the required verifiability to DPP documents and proofs.

We have explored the needs and role of digital product information, a DPP or circular management digital twin, for refurbished computers, assuming original manufacturers provide the information they have from the supply chain. Still, our design combines both sets of details, those produced by a manufacturer in the pre-use phase with those for circularity processes in the use phase.

Performance, response time or overhead is not an issue here. We already reported that in [NGIA21], with an evaluation of our PoA-permissioned Ethereum DLT. IOTA has measured performance as part of an experimental testbed for the EBSI pre-procurement competition in phase 2a [EBSI22].

From our experience with these two DLT drivers, there are three abilities as main requirements: the ability to record data irreversibly (a form of immutability, append only), the ability to identify participants to award permissions, the ability to run trusted code to guarantee that checks and processing according to stable code working according to agreed governance rules, a form of inexorability. That can be satisfied by diverse DLTs or blockchains, whether public, private, permissionless or permissioned. Furthermore, there is the design requirement of privacy and security by design, keeping sensitive information away from the verifiable registry, by recording decentralised identifiers and summaries of detailed data, that can only make sense in the light of information details to be looked up with via DID, accessed by authorised through a DID subject, that can be checked by

[1] https://github.com/eReuse/devicehub-teal

comparison of a summary stored in the verifiable registry with the summary of the detailed retrieved data.

Multiple DLTs (drivers) imply the choice to record verifiability proofs in one, but not a federation or replication. API clients can decide and express a preference for one driver to record their proofs.

Transparency and accountability are linked to the ability to verify data, which means performing an audit. That is related to the verifiability need in the EU ecodesign directive for products, the need for due diligence in green public procurement, the “non-financial” reporting required by the EC to certain organisations, the growing need for environmental impact assessment of organisational activities, or the need for validation of open datasets. That follows the pattern of looking up DIDs to retrieve DID documents that point to two types of sources, one for informative details and the other for verifiability proofs. Finding the DID associated with an ICT device can be achieved in multiple ways through diverse data carriers, such as scanning a QR code on a label, reading a NFC or RFID tag, or running code in the device that recreates the DID from internal hardware identification information. After that, a lookup can lead to finding the right service endpoint to retrieve details either on a service instance of a details (inventory) service or through an instance of a verifiable registry API. Ths lookup requires an identifier lookup such as those provided by hierarchical schemes, such as DOI, or decentralised schemes such as a DHT in IPFS or name-based lookup in NDN.

Governance (social sustainability) requires agreements on procedures to manage verifiability information, that can be later implemented in code and run inexorably as smart contracts. Economic sustainability depends on ways to contribute to covering the infrastructure and service costs of the registry, as well as providing support and motivation to participants in circular business models to contribute with fees or acting driven by economic incentives set by governing bodies.

The model of a registry to record verifiability information about ICT devices is quite general. To ensure information and devices are not lost, the major problem nowadays with e-waste, is keeping devices accountable. Any device capable of introspection for monitoring (checking its own hardware, reading its identifiers, sensing its status, diagnostics or configuration, including its battery) is equivalent to an IoT device with environmental sensors, that can also sense internal information. This is a quite generic model for any ICT device (e.g., computers, routers, phones, IoT), a device-centric model assuming that ICT devices preserve a lifetime identity (chassis) but are reconfigurable due to modification during their lifespan. These operations and configurations can be recorded as proofs, performed by actors, and associated with documents, to keep a track record for accountable circular management.

Devices with a unique identifier that can be looked up through a data carrier to find details, links and verifiability information that allows following (twin) a physical device along its lifespan to inform, record and prove relevant product-related information about a physical device constitutes the basis for a digital product passport. Beyond that, harmonisation about which information is relevant and how to access and verify it, is part of future work that has started but will require reaching a consensus across all stakeholders involved with devices. Something that will take time and probably an evolutionary approach of successive refinements. The standardisation work in ITU-T and ETSI on the DPP topic are examples.

Verifiability information, as well as user keys, assume a degree of reliability, riskier as systems grow in size and volume of operations. Fragile hardware and software, or users with

fragile memory may require to add recovery mechanisms to prevent catastrophic situations and recover from failures.

Conclusions

In conclusion, ICT devices have a significant environmental impact that needs to be reduced for improved sustainability. The digital product passport, a digital twin of a device, provides detailed and trusted information about the device throughout its lifespan, including multiple use phases and changes. A multi-DLT registry, which can record verifiable, document-supported proofs about milestones of the hardware configuration of ICT devices over a circular lifespan with multiple owners and uses, has been developed and tested using Ethereum and IOTA DLTs. Our results demonstrate that the DLT-agnostic registry API can effectively complement device inventory services and produce a trusted digital product passport for devices throughout their lifespan.

Future work is needed to harmonise the information that is relevant for a DPP and the methods for accessing digital twins and verifying that information. That requires the involvement of all stakeholders involved with devices, including industry, governments and the public. It will likely be a process of successive refinements that requires reaching a consensus among all stakeholders. The standardisation work being conducted by ITU-T and ETSI on the DPP topic is an example of this effort.

Acknowledgements

This work was partially funded by the NGI Trublo project, IOTA Foundation, Spanish Government under contracts PID2019-106774RB-C21, and PCI2019-111851-2 (LeadingEdge Chistera).

References

[AC23] AssetCloud (2023). AssetCloud. Retrieved from https://www.assetcloud.com/

[AP23] Asset Panda (2023). Asset Panda. Retrieved from https://www.assetpanda.com/

[CAPC21] Association for Progressive Communications (2021), A guide to the circular economy of digital devices, https://circulartech.apc.org/books/a-guide-to-the-circular-economy-of-digital-devices/

[D4PI22] ITU-T Q7/SG5 L.D4PI (2022), An information model for digital product information on sustainability and circularity, https://www.itu.int/ITU-T/workprog/wp_item.aspx?isn=18559

[EBSI22] European Commission (2022), European Blockchain Pre-Commercial Procurement (EBSI), https://digital-strategy.ec.europa.eu/en/news/european-blockchain-pre-commercial-procurement

[EC21] European Commission (2021), EU countries commit to leading the green digital transformation,

https://digital-strategy.ec.europa.eu/en/news/eu-countries-commit-leading-green-digital-transformation

[EMF21] Ellen McArthur Foundation (2021), What is the circular economy? https://www.ellenmacarthurfoundation.org/circular-economy/what-is-the-circular-economy

[ERE23] eReuse.org (2023). DeviceHub. Retrieved from https://www.ereuse.org/tool/devicehub/

[EDID22] Veramo.io Uport Project (2022), Ethr-DID Library, https://github.com/uport-project/ethr-did

[EWM20] Forti, V., Baldé, C., Kuehr, R., Bel, G. (2020). The Global E-waste Monitor 2020. UNU/ UNITAR and ITU. https://www.itu.int/en/ITU-D/Environment/Documents/Toolbox/GEM_2020_def.pdf

[FS23] Freshservice (2023). Freshservice. Retrieved from https://www.freshservice.com/

[GALA21] Michele Galatola, DG GROW, Green and Circular Economy Unit, European Commission (2021), The Sustainable Products Initiative, https://www.lifeeffige.eu/wp-content/uploads/2021/06/20210609_presentazione_Michele_Galatola.pdf

[GDS21] ITU-T Q7/SG5 L.GDSPP (2021), Requirements for a global digital sustainable product passport to achieve a circular economy, https://www.itu.int/ITU-T/workprog/wp_item.aspx?isn=17712

[GOTZ22] Thomas Götz, Holger Berg, Maike Jansen, Thomas Adisorn, David Cembrero, Sanna Markkanen, Tahmid Chowdhury (2022), Digital product passport: the ticket to achieving a climate neutral and circular European economy? https://nbn-resolving.org/urn:nbn:de:bsz:wup4-opus-80497

[GUTH21] Susanne Guth-Orlowski (2021), The digital product passport and its technical implementation, https://medium.com/@susi.guth/the-digital-product-passport-and-its-technical-implementation-efdd09a4ed75

[HAL20] Halpin, H. (2020). Vision: A Critique of Immunity Passports and W3C Decentralized Identifiers. In: van der Merwe, T., Mitchell, C., Mehrnezhad, M. (eds) Security Standardisation Research. SSR 2020. Lecture Notes in Computer Science, vol 12529. Springer, Cham.

[IDID23] IOTA Foundation (2023), IOTA DID Method Specification, https://wiki.iota.org/identity.rs/specs/did/iota_did_method_spec

[INTS22] IOTA Foundation (2022), IOTA Integration services, https://www.iota.org/solutions/secure-digital-infrastructure

[ITUT21] UN ITU-T (2021), Sustainable Digital Transformation Dialogues, https://www.itu.int/en/ITU-T/Workshops-and-Seminars/sg05rg/sdtd/Pages/default.aspx

[LAU21] J. Lauinger, J. Ernstberger, E. Regnath, M. Hamad and S. Steinhorst (2021), "A-PoA: Anonymous Proof of Authorization for Decentralized Identity Management," 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC).

[MEE20] P. S, Meeradevi and M. R. Mundada (2020), Analysis of Agricultural Supply Chain Management for Traceability of Food Products using Blockchain-Ethereum Technology, 2020 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), pp. 127-132.

[NAV22] Navarro, L., Esteban, J. C., Miralles, M. F., & Griso, D. F. (2022). Digital transformation of the circular economy: digital product passports for transparency, verifiability, accountability. Technical report (under journal revision) https://people.ac.upc.edu/leandro/docs/DPP_DLT_ACMJ22.pdf

[NGIA21] UPC, NGI-Atlantic (2021), D3: Decentralized data ecosystem for the Open Blockchain for Asset Disposition Alliance Deliverable 3: Experiment Results and Final Report, https://people.ac.upc.edu/leandro/p/D3-final-pub.pdf

[PAR19] Park, J., Chitchyan, R., Angelopoulou, A., Murkin, J. (2019). A Block-Free Distributed Ledger for P2P Energy Trading: Case with IOTA?. In: Giorgini, P., Weber, B. (eds) Advanced Information Systems Engineering. CAiSE 2019. Lecture Notes in Computer Science(), vol 11483. Springer, Cham.

[STR23] IOTA Foundation (2023), IOTA Streams, https://www.iota.org/solutions/streams

[VRAPI22] Distributed Systems Group at UPC (2022), Trublo-eReuse verifiable registry, contracts API, https://gitlab.com/dsg-upc/trublo_contracts_api and documentation: https://gitlab.com/dsg-upc/trublo_contracts_api/-/wikis/Overview.

[VRTST22] Distributed Systems Group at UPC (2022), Verifiable registry API testing features BDD with Cucumber, https://gitlab.com/dsg-upc/trublo_contracts_api/-/tree/testing_fixes/features/api

[W3CC22] W3C (2022), Verifiable Credentials Data Model 1.1 - W3C Recommendation. Retrieved from https://www.w3.org/TR/vc-data-model/

[W3CD22] W3C (2022), Decentralized Identifiers (DIDs) - W3C Recommendation. Retrieved from https://www.w3.org/TR/did-core/

[W3CI21] W3C (2022), DID Implementation Guide v1.0 Retrieved from https://www.w3.org/TR/did-imp-guide/

[WAL21] Walden, J., Steinbrecher, A. and Marinkovic, M. (2021), Digital Product Passports as Enabler of the Circular Economy. Chemie Ingenieur Technik, 93: 1717-1727.

[WAN19] X. V. Wang & L. Wang (2019) Digital twin-based WEEE recycling, recovery and remanufacturing in the background of Industry 4.0, International Journal of Production Research, 57:12, 3892-3902.