IoT-NGIN

D5.3
Enhancing IoT Data Privacy & Trust

WORKPACKAGE	WP5	**PROGRAMME IDENTIFIER**	H2020-ICT-2020-1
DOCUMENT	D5.3	**GRANT AGREEMENT ID**	957246
REVISION	V1.0	**START DATE OF THE PROJECT**	01/10/2020
DELIVERY DATE	30/11/2021	**DURATION**	3 YEARS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 957246

DISCLAIMER

ACKNOWLEDGEMENT

This document is a deliverable of the IoT-NGIN project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 957246.

The opinions expressed in this document reflect only the author's view and in no way reflect the European Commission's opinions. The European Commission is not responsible for any use that may be made of the information it contains.

PROJECT ACRONYM	IoT-NGIN
PROJECT TITLE	Next Generation IoT as part of Next Generation Internet
CALL ID	H2020-ICT-2020-1
CALL NAME	Information and Communication Technologies
TOPIC	ICT-56-2020 - Next Generation Internet of Things
TYPE OF ACTION	Research and Innovation Action
COORDINATOR	Capgemini Technology Services (CAP)
PRINCIPAL CONTRACTORS	Atos Spain S.A. (ATOS), ERICSSON GmbH (EDD), ABB Oy (ABB), INTRASOFT International S.A. (INTRA), Engineering-Ingegneria Informatica SPA (ENG), Bosch Sistemas de Frenado S.L.U. (BOSCH), ASM Terni SpA (ASM), Forum Virium Helsinki (FVH), Optimum Technologies Pilroforikis S.A. (OPT), eBOS Technologies Ltd (EBOS), Privanova SAS (PRI), Synelixis Solutions S.A. (SYN), CUMUCORE Oy (CMC), Emotion s.r.l. (EMOT), AALTO-Korkeakoulusäätiö (AALTO), i2CAT Foundation (I2CAT), Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), Sorbonne Université (SU)
WORKPACKAGE	WP5
DELIVERABLE TYPE	REPORT
DISSEMINATION LEVEL	PUBLIC
DELIVERABLE STATE	FINAL
CONTRACTUAL DATE OF DELIVERY	30/11/2021
ACTUAL DATE OF DELIVERY	30/11/2021
DOCUMENT TITLE	Enhancing IoT Data Privacy & Trust
AUTHOR(S)	Lei Wu, Dmitrij Lagutin, Teemu Kääriäinen, Juuso Autiosalo, Yki Kortesniemi (AALTO), Kimmo Ojala [ABB], Antonello Corsi [ENG], Jonathan Klimt, Maliheh Haghgoo [RWTH]
REVIEWER(S)	Kimmo Ojala [ABB], Dimitrios Skias [INTRA]
ABSTRACT	SEE EXECUTIVE SUMMARY
HISTORY	SEE DOCUMENT HISTORY
KEYWORDS	cyber security, privacy, distributed ledger technology, interledger, self-sovereign identity, decentralised identifier, verifiable credential, digital twin, interoperability, ontology, semantic twin, digital twin

Document History

Version	Date	Contributor(s)	Description
V0.1	04/05/2021	AALTO, ENG	State of the art report for interledger
v0.2	30/06/2021	AALTO	State of the art report for SSI technology
v0.3	23/09/2021	AALTO	Design of solution for decentralised interledger
v0.4	29/09/2021	AALTO	Added description of SSI design
v0.7	25/11/2021	AALTO, ABB, RWTH	Ready for internal review
v1.0	30/11/2021	AALTO, ABB, RWTH	Finalised based on review feedback

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AML	Anti Money Laundering
CCC	Cross-Chain Communication
CIM	Context Information Management
CSP	Credential Service Provider
DIB	Decentralised Interledger Bridge
DID	Decentralised IDentifier
DLTs	Distributed Ledger Technologies
DSM	Decentralised State Management
DT	Digital Twin
DTDL	Digital Twins Definition Language
FIB	Flexible Interledger Bridge
GDPR	General Data Protection Regulation
HTLC	Hash Time Locked Contract
IAA	Identity, Authentication, and Authorization
ILP	Interledger Protocol
IoT	Internet of Things
JWT	JSON Web Token
KSI	Keyless Signatures Infrastructure
KYC	Know Your Customer
MLDT	Meta-Level Digital Twin [deprecated]
PDS	Privacy and Data Sovereignty
SAREF	Smart Applications REFerence ontology
SSI	Self-Sovereign Identities
SSN	Semantic Sensor Network
ST	Semantic Twin
UUID	Universal Unique Identifier
VC	Verifiable Credentials
WoT TD	WoT Thing Description
ZKP	Zero-Knowledge Proof

Executive Summary

This document focuses on the problems of *privacy preservation and trust improvement* in the domain of IoT systems and explores select technical approaches to tackle the problems.

The requirements from the various use cases in the IoT-NGIN project are identified and analysed to determine the best features and properties for the technical solutions to be developed within the IoT-NGIN project. The state-of-the-art in the field of *multi-ledger operations*, *Self-Sovereign Identities*, and *semantic interoperability* practices for *Digital Twins* are then studied in order for the technical solutions to apply the latest and best practices in the field.

Based on the requirements and best practices, the document then describes the high-level approach and/or architecture that will be applied in the solutions being developed for the project pilots.

The detailed descriptions of the developed solution will follow in the upcoming deliverable D5.4.

1 Introduction

The expanding use of IoT solutions has enabled many new services, but has also raised new privacy and trust challenges. Ubiquitous IoT makes it possible to have a much more accurate and up-to-date situational awareness, but this can pose major privacy issues to the individuals, whose actions are being observed with this technology. Also, individuals themselves are deploying more IoT devices and are in some cases even making the collected data available to a wider audience to enable new services, but at the same time also potentially raising privacy issues. Finally, for the audience utilising the data, a key question is, which IoT devices and data to trust in this abundance of options.

This document addresses these problems in the context of the IoT-NGIN project using a few select technologies: multi-ledger operations, Self-Sovereign Identities, and semantic interoperability practices for Digital Twins. For each of the technologies, the project use cases are first analysed to extract requirements for the technical solution to be developed. Next, the State-of-the-Art of the technology in question is analysed to identify the best approach to utilise in the solution. Finally, the solution design is described on a high level. The detailed solution description will then follow in the upcoming deliverable D5.4.

1.1 Intended Audience

This document is intended for the following groups of people:

- Technical people that are interested in IoT systems, decentralised applications, Digital identity management, and Digital twin interactions can find state-of-the-art reviews of these fields and examples of applying the latest technical solutions to select application scenarios.
- Solution designers and policy makers may find the document helpful to understand what kind of services the different technical solutions enable, which level of trust and privacy protection can be provided, and what standard ways for semantic interoperability are possible.
- Internal users within the IoT-NGIN project can find useful resources on the components or architecture solutions that are being made available in WP5, so that use of developed modules is made easier.

1.2 Relations to other activities

This document reviews the state-of-the-art of various technical solutions involving interledger, Self-Sovereign Identities, ontologies and semantic twins, and can, thus, provide guidelines to other work packages in the project on best practices in the respective fields.

Also, the corresponding solutions described in this document and the proof-of-concept implementations that will follow can serve as examples on how to tackle similar problems in IoT systems.

1.3 Document overview

The document is organised around the 4 solutions covered.

Section 2 covers the motivation for multi-ledger operations, identifies the requirements of interledger solutions for the use cases in IoT-NGIN, analyzes the state-of-the-art of interledger operations, especially for general-purpose applications, and finally highlights the approach of the decentralised interledger bridge architecture being developed.

Section 3 looks back to the concept and evolution of digital identities in the past, lists the relevant SSI techniques that can help with IoT systems, and presents the PoC architecture solution based on the use case requirements within the project.

Section 4 describes the motivation for ontologies to formally model the structure of digital systems, reviews the existing SAREF ontology, and details how it can be extended to meet the requirements of the IoT-NGIN project.

Section 5 discusses the need for semantic interoperability for digital twins, defines the concept of Semantic Twin (called a Meta-Level Digital Twin in the IoT-NGIN proposal) to address the interoperability issue, reviews the related techniques, and describes how those can be applied to the use cases in IoT-NGIN project.

Finally, Section 6 concludes the report.

2 A decentralised Interledger solution

This section discusses the need for multi-ledger transactions and how they can be met with a suitable interledger solution. The section then summarises the IoT-NGIN requirements for the interledger and reviews the existing interledger approaches. Based on the requirements, the Flexible Interledger Bridge (FIB) [Wu2021] developed in the EU Horizon 2020 project SOFIE [SOF2021] is then chosen as the basis for developing a decentralised solution, the *Decentralised Interledger Bridge (DIB)*. The section then concludes with an overview of the DIB architecture.

2.1 Motivation for Interledger

Interledger technologies enable transactions that span two or more distributed ledgers. This subsection details why a separate technical solution is required for linking the ledgers, what benefits this approach enables, and what requirements a good interledger solution has to meet to be able to address the needs of IoT-NGIN

2.1.1 Diversity of DLTs

Distributed Ledger Technologies (DLTs) have been developed for over a decade, and they have been widely adopted due to the immutability and transparency provided by the decentralised secure storage, the distributed trust ensured by sophisticated consensus algorithms, and the automatic execution within the system enabled by features such as smart contracts [Zha19].

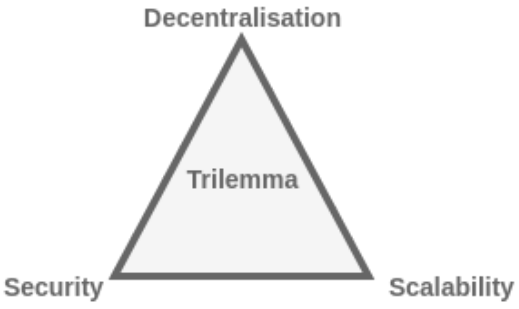

Figure 2.1 - Blockchain trilemma.

According to their individual design goals, different DLTs have varying emphasis, including the accessibility of data on the ledger, i.e., who is allowed to read or write on the ledger, the consensus mechanism adopted to reach agreement on ledger status, and the range of supported functionalities. The initial focus in the area was on distributed finance and cryptocurrencies, such as Bitcoin [Nak08], Litecoin [Lee2011] and Ripple [Sch14]. Later, DLTs with highly different capabilities have been developed: Ethereum [Woo2014]aims to enable a world computer, Hyperledger Fabric [And2018] and Corda [Bro2016] support enterprise-focused ledgers, while the Sovrin network [Kho2017] enables Self-Sovereign

Identities (SSIs) [Muh18]. The key design trade-offs for all DLTs have been summarised in the so-called *blockchain trilemma* illustrated in Figure 2.1, namely that decentralization, security, and scalability cannot all be maximised at the same time [Zho20]. Further design choices for DLTs revolve around data privacy, openness, performance, cost, and the supported functionality.

2.1.2 Need for multi-ledger transactions

As DLTs have been deployed to more application areas, it has become clear that no single DLT is suitable for all the use cases, and sometimes even the requirements of a single complex use case can easily exceed the strengths and capabilities of any single DLT. In such situations, combining multiple DLTs with different strengths and features can be a beneficial approach as it enables new functionality [But16]. For instance, it might help improve the data integrity by utilizing a highly trustworthy public ledger, while reducing the cost and latency of a system by keeping most of the heavy-lifting business logic in private ledgers.

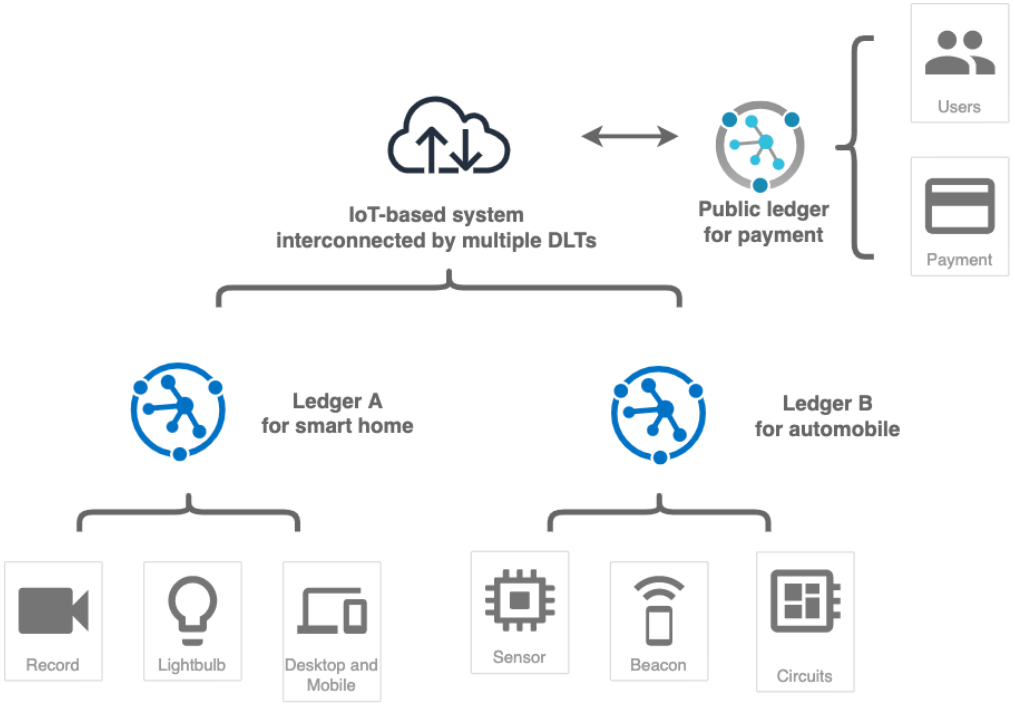

Figure 2.2 - An IoT-based system combining multiple DLTs.

A typical example are Internet of Things (IoT) systems, where an information sharing mechanism across multiple DLTs could help resolve the security, maintenance, and authentication issues in an automated manner [Has2019]. As illustrated in Figure 2.2, it is typical that IoT devices and services are connected to and backed by private distributed ledgers of individual vendors so that, e.g., the devices and equipment for a smart home interact with Ledger A, and the automobile sensors and circuits work together with ledger B. Then, a public ledger could be used for providing services for authentication and

payment, and interlinking these three DLTs would enable a more complex (eco)system with additional functionality, e.g. payment services could be used with automobile ledgers at electricity charging stations.

Also some applications rely on multiple ledgers to utilise assets in different ways, but the asset is allowed to be active in only one ledger at a time. An example is an online game [Cam2021], where one ledger is used to maintain all the player's assets (weapons, clothing etc.) available in the game while another ledger is used as the marketplace to trade the assets between the gamers, and an asset being traded cannot be used in the game and vice versa. As shown in Figure 2.3, a suitable interledger can be used to build a protocol to ensure the asset (represented as ERC721 tokens [ERC721] on Ethereum) follows the above rule. The same process then takes place in the reverse direction when the asset is returned to the Game Asset Ledger.

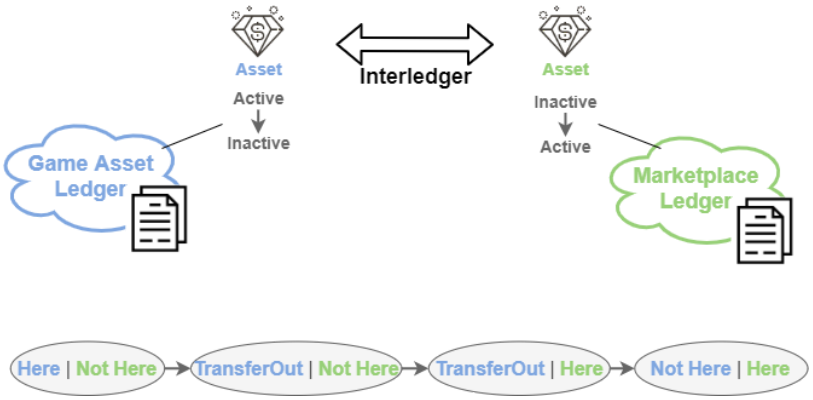

Figure 2.3 – Game asset state transfer enabled by Interledger.

So, combining DLTs together via information exchange or transfer functionalities can be beneficial in multiple ways. First, the strengths of different types of ledgers can be utilised at the same time, such as the performance of private ledger and the security of public ledger. Second, the limitation of the individual ledgers can often be overcome by using them in connection with other types of ledgers in the system. Finally, often new functionalities are enabled when different ledgers are applied together. However, to gain all these advantages, a general-purpose, flexible, and highly efficient interledger solution is required. In practice, the diversity of design goals, consensus mechanisms, and supported functionalities of the various DLTs described above have become a major obstacle for the interoperability between them, and the ledgers themselves don't normally provide any functionality for communicating with other ledgers, so all transactions between ledgers require a separate interledger solution.

2.1.3 Requirements of IoT-NGIN

The Interledger solution being developed (from here on: interledger) will be used in the IoT-NGIN project in several ways including the Smart agriculture Living lab from

WP7(specifically the disease prediction and irrigation precision UC 3.1), the IoT intelligence empowered by federated machine learning from WP3, and also the meta-level digital twins use case from WP5. Further, the IoT-NGIN architecture is expected to introduce many other uses for the interledger beyond the IoT-NGIN project itself.

Specifically, the Smart agriculture Living lab in WP7 wants to store state data related to disease findings and volume of irrigation water etc., while in WP3 trusted AI is targeted for federated machine learning: Zero Knowledge Proofs (ZKPs) of training datasets and trained federated machine learning model together with its parameters can be automatically stored on DLTs in form of hash values and later utilised for verification by third party to ensure those are not tampered with, while no actual data is released on the DLTs. Finally, meta-level digital twins in WP5 utilize DLTs in a similar pattern, to ensure the integrity of relevant objects.

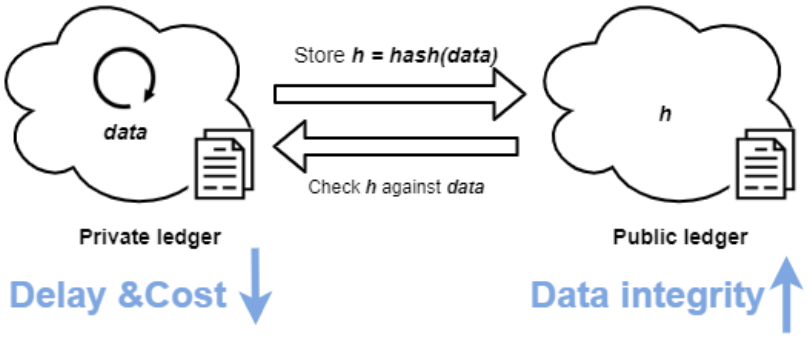

Figure 2.4 – Hash storage for data integrity.

All the above use cases require auditability for logged data, but storing everything in a highly trustworthy public ledger would result in high costs and expose all data to potentially prying eyes. The low throughput of public ledgers can also become a problem in some cases. Storing everything in a private ledger would protect privacy, provide better throughput, and slash costs, but would also lack the high level of trust. A solution is to store the data in the private ledger and then leverage an interledger to automatically store a hash of the data at suitable intervals to the public ledger, as shown in Figure 2.4. This way, it is easy to verify whether the data in the private ledger has been tampered with while the overall costs are kept significantly lower as the usage of the expensive public ledger is reduced drastically.

Based on these different uses discussed above, 7 key requirements for the interledger solution can be identified as listed in Table 2.1 and detailed in the following text.

REQ_IL_NF01: The interledger must be able to support the transfer of different types of data (so this excludes e.g. interledger solutions that focus exclusively on value transfers). Also, depending on the use, different types of DLTs may be utilised as part of the system, so the interledger solution has to be adaptable to different DLTs with relative ease.

Table 2.1 – Requirements for the interledger solution.

ID	Requirement	Description
REQ_IL_NF01	Generality	Must support general-purpose data transfers and be easily adaptable to different types of distributed ledgers.
REQ_IL_NF02	Atomicity	Must guarantee atomicity of transactions across the ledgers.
REQ_IL_NF03	Transparency	Must be transparent enough that the correct operation of all transactions can be verified based on the data on the ledgers.
REQ_IL_NF04	Non-repudiation	Must support non-repudiation so that the participants to a transaction cannot later deny their actions.
REQ_IL_NF05	Scalability	Must support a large number of transactions per second.
REQ_IL_NF06	Efficiency	Should keep the application overhead low
REQ_IL_NF07	Decentralization	Must support decentralisation, where the interledger is run by a consortium of parties

REQ_IL_NF02: The interledger must guarantee that the transactions across the ledgers are atomic, i.e. they happen completely on all the involved ledgers or not at all.

REQ_IL_NF03: The interledger must provide transparency to the operations so that the correct operations of the interledger can be verified based on the data on the ledgers.

REQ_IL_NF04: The interledger must operate so that non-repudiation for all parties of each individual transaction is guaranteed.

REQ_IL_NF05: The interledger must be designed so that it can support a large number of transactions per second.

REQ_IL_NF06: The interledger should minimize the overhead (cost, performance, storage etc.) for the application utilizing the component for cross-ledger communication.

REQ_IL_NF07: The interledger itself must support decentralisation, i.e. that the functionality is provided by a consortium of parties so that none of them can misbehave in any data transfer (e.g. change data payload, report invalid ledger transaction, or reject the transfer) without being detected by others. As a contrast, an interledger run by a single party has several limitations: the party has to be trusted by all users and it forms a single point of failure that can also pose problems for the resiliency and performance of the solution; a decentralised interledger helps address these limitations.

2.2 Review of the State-of-the-Art

Over the years, different interledger technologies have already received significant attention as the original "one chain to rule them all" model was found too limited [Sir2019] [Bel2020] [Zam2019]. The following subsections first summarise two classifications for the existing interledger approaches and then discuss an effort to formalise the cross-chain communication problem in standard protocols. Finally, the existing solutions are analysed in view of the requirements listed in the previous section.

2.2.1 Interledger by different techniques

Siris et al [Sir2019] survey a wide range of interledger approaches, which differ in whether they support the transfer or exchange of value, their trust mechanism during connection, complexity, scalability, and transaction cost. The approaches are divided in six categories:

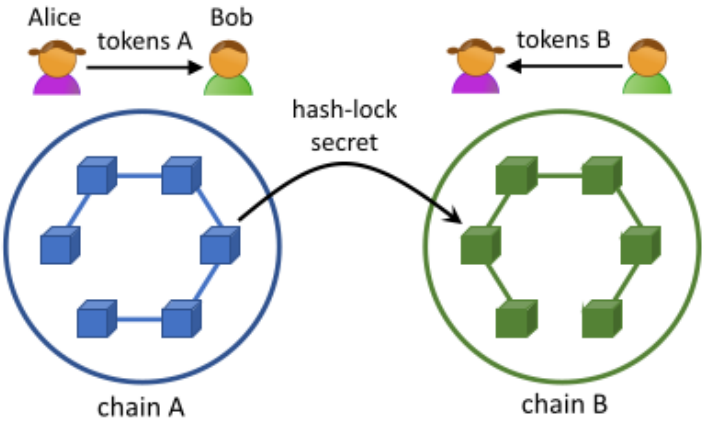

Figure 2.5 - Atomic cross-chain trading, from [Sir2019].

Category 1: Atomic cross-chain transactions focus on trading assets on two blockchains while avoiding the risk that a participant is not obeying the agreement [But2016] [Her2018], thus enabling low-complexity peer-to-peer trading based on simple mechanism like Hash Time Locked Contract (HTLC) [Htl2021], as illustrated in Figure 2.5, where a HTLC ensures that either both Alice and Bob receive all their intended token(s) or the whole token swap is cancelled.

Category 2: Transactions across a network of payment channels is a decentralised system for routing micropayments, realising off-chain exchange of value [Dec2015]. It improves scalability using off-chain processing thus reducing the total transaction cost, but it also introduces privacy concerns at the middle nodes [Poo2016] [Bur2018].

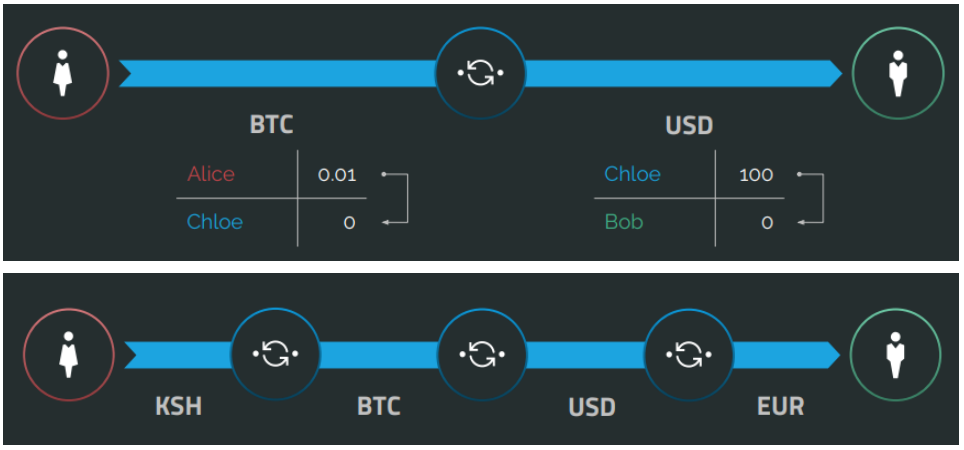

Figure 2.6 - W3C Interledger Protocol.

Category 3: The W3C Interledger Protocol (ILP) is a protocol [Ilp2021] for secure atomic transfer of funds across any two ledgers that are compatible with ILP [Hop2016]. Despite its high scalability and openness, ILP only focuses on exchange of value just like the first two approaches. The ILP protocol has undergone a number of design changes, with its main implementations being version 1 (ILPv1) and 4 (ILPv4). The overarching goal of ILP is to allow atomic transfer of funds from a ledger A to a ledger B in such a way that none of the involved parties incurs any risks and the sender gains an indisputable proof that the final receiver redeemed the funds. This is achieved by means of a third user, referred to as the connector, who maintains accounts in both ledgers A and B, so the transfer involves two distinct transactions. So, if the sender makes a transfer to the connector first, he/she has to trust that the connector will also do its part by then transferring the value to the recipient. Likewise, if the connector transfers the amount to the recipient first, it has to trust that the sender will pay it accordingly. This requires the entire transfer to be atomic, that is, either both or neither transactions are executed, which defines a so-called *escrow transaction*, whose redemption requires the satisfaction of a condition. The Ledger-based escrow payments used in ILPv1 are inherently slow and expensive, thus they are no longer a requirement for value transfers, and with the new ILPv4 version is it possible to allow connectors to set up bilateral trust relations of arbitrary type not bound to ledger typologies. Although this appears to call for high risks compared to escrow payments, all risks are confined exclusively between directly interacting connectors and typically concern only small values. Importantly, regardless of how the connectors interact, the senders and receivers enjoy a completely risk-free operation.

Category 4: Bridging approaches are one or two-way transfers of value or information between two or more distributed ledgers. They are usually implemented with modules that connect to the ledgers for monitoring transactions and exchanging information, so they can also be designed to support non-value-based uses and to link multiple different types of ledgers. However, many existing bridging solutions including Blocknet [Blo2021], ARK [Ark2021], and BTC Relay [Btr2021] focus only on the value exchange, and other works like

POA network [POA] and Wanchain [Wan2021] work only with Ethereum-based blockchains, or are limited to certain consensuses as is the case for e.g. Aion [Aio2021].

One of the latest bridging approaches is the *Flexible Interledge Bridge (FIB)* design that addresses many of these shortcomings. FIB has been designed to support many different types of applications by enabling a simple atomic data transfer between a source and a destination ledger with the following steps: 1) the source sends the data; 2) the destination either accepts or rejects the data; 3) Source commits or aborts the data transfer according to that feedback in step 2, thus ensuring a consistent state for both parties. Despite its simplicity, this basic functionality can be the building block for many more complicated business logic and protocols.

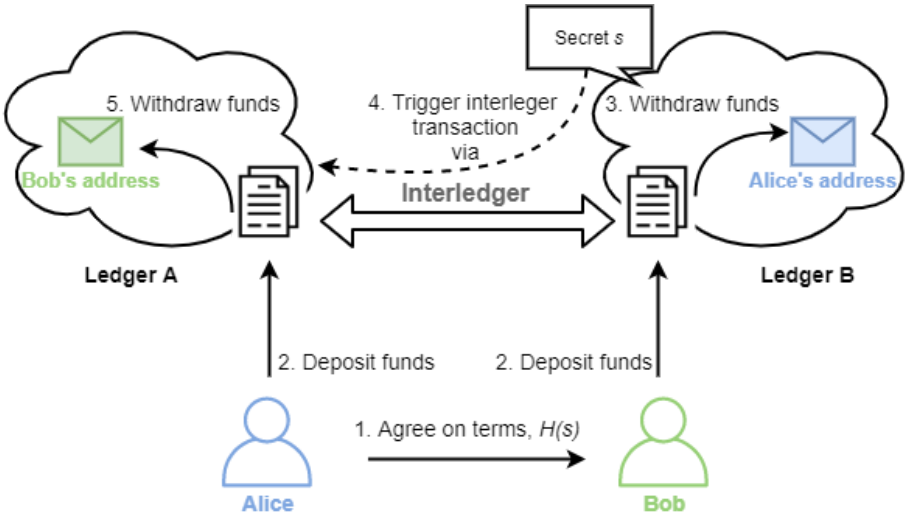

Figure 2.7 – Interledger-based HTLC for asset exchange across ledgers.

As a concrete example, HTLCs are utilised to ensure that linked transactions, such as an exchange of assets between two parties, happen atomically, i.e. that either both complete successfully or both fail. In HTLCs, the same hash lock protects execution of the linked transactions and after one party reveals the secret to claim its assets, the second party can do the same with the revealed secret. While HTLCs are usually used within a single ledger, the FIB allows automation of the HTLC process across multiple distributed ledgers: after the secret is revealed on one ledger, the Interledger node conveys the secret to another ledger, triggering the corresponding transaction and concluding the HTLC process.

Category 5: Sidechain solutions move assets from a main chain to a side chain, which is used as auxiliary worker, and later return the asset back to the main chain thus allowing more features or higher efficiency than the main chain alone could provide [Bac2014] [Dil2016] [Ler2016]. Many sidechains focus only on asset state transfer, while others serve intra-ledger transactions [Poo2017]. However, all such frameworks have difficulty with interactions between heterogeneous independent ledgers.

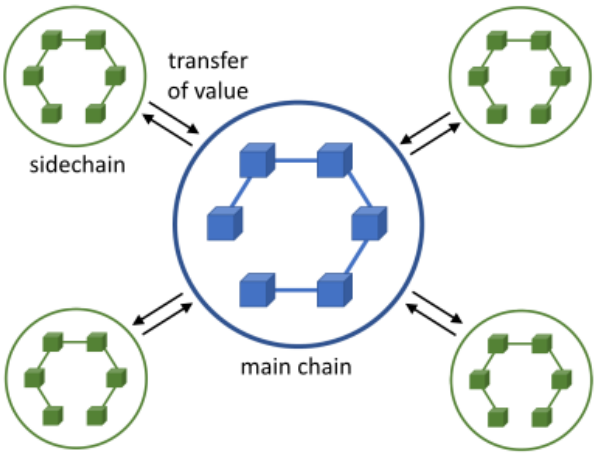

Figure 2.8 - Sidechains.

Category 6: Ledger of ledgers introduces a separate "super-ledger" with which multiple sub-chains can synchronise [Eng2016]. While high scalability can be achieved with this approach, it also results in high complexity due to the additional ledger(s). Two representatives of this approach are Cosmos [Kwo2019] and Polkadot [Woo2016].

2.2.2 Interledger for different purposes

Belchior et al [Bel2020] presents an alternative categorisation based on the *purpose of solutions*, which divides the existing solutions into three major types.

1) Chain interoperability solutions [But2016] within the domain of cryptocurrencies are categorised as *cryptocurrency-directed approaches* regardless of the underlying techniques, e.g. sidechains [Kok2018] [Btr2021] [Blo2021], notary schemes [Li2019], HTLC [WCW2021], and their hybrids.

2) *Blockchain Engines* [Kwo2019] [Woo2016] [Ark2021] [Aio2021] are frameworks that provide reusable data, network, consensus, incentive, and contract layers for the creation of customised blockchains to power decentralised applications that interoperate between each other.

3) *Blockchain connectors* includes all other efforts to connect distributed ledgers and it is subdivided into four subcategories. *Trusted relays* such as Hyperledger Cactus [Hyp2021] and Abebe et al's work [Abe2019] are mechanisms utilising trusted parties that redirect transactions from a source blockchain to a target one. Trusted relay is often used in private blockchain environments with a blockchain registry facilitating the discovery of the target blockchains. *Blockchain Agnostic Protocols* [Ilp2021] [Kan2018] try to provide technology-agnostic protocols for interoperability between distributed ledgers that are compatible with the protocols. *Blockchain of blockchains* [Ver2018] [Liu2019] is a system, where a consensus protocol organises blocks containing transactions belonging to cross-chain applications. Finally, *Blockchain migrators* perform runtime state migration across blockchains with a mechanism that resembles the notary scheme.

Intersections or hybrid approaches in such classifications are not only possible but also the focus in some research and industry. For instance, the cryptocurrency-directed Xclaim framework [Zam2018] makes use of the atomic cross-chain transfer techniques. Also, blockchain engines such as Cosmos [Kwo2019] and Polkadot [Woo2016] make use of the ledger-of-ledgers approach as mentioned above. Despite the popularity of both, there are obvious limitations, i.e. the risk of potential integrity breaches and the complexity of additional architecture (peg zones in Cosmos or bridge chains in Polkadot) when connecting ledgers with probabilistic and deterministic consensus mechanisms. Likewise, sidechains remain a major topic in sharding schemes [Al2017] [Kok2018] of cryptocurrency-based chains, both for processing capability and scalability concerns.

2.2.3 Cross-Chain Communication (CCC) problem

Finally, Zamyatin et al. [Zam2019] formalised the *Cross-Chain Communication (CCC)* problem and showed the impossibility of achieving fair exchange in such a process without a trusted third party. The CCC process in the provided design and evaluation framework contains four major steps, i.e. 1) *Setup* of a protocol with the parameters of the ledgers, parties, and transactions involved; 2) *Pre-commit* the transfer on the source ledger to be publicly verifiable; 3) *Verify* by node of the destination ledger the correctness of commit made on the source ledger; 4) *Commit or Abort* the transfer on the destination ledger. A similar scheme can be utilised in those general purpose cross-chain communication or interledger solutions, and is the approach used by FIB.

2.2.4 The best interledger approach for IoT-NGIN

Though there are a large number of existing interledger approaches, none are perfectly suitable for the needs of IoT-NGIN in view of requirements from Section 2.1 as summarised in Table 2. In particular, REQ_IL_NF01 (support for different types of data and ledgers) already excludes most approaches. First, categories 1, 2 and 3 in Section 2.2.1 and the cryptocurrency-directed approaches in Section 2.2.2 all focus exclusively on value exchange. Similarly, Blockchain Engines in Section 2.2.2 are not for general application either, and techniques such as sidechains (category 5) and some of the blockchain migrators have restrictive assumptions about the consensus or functionalities the ledgers should support, again a violation of REQ_IL_NF01. Finally, the high complexity of the ledger-of-ledgers (Category 6) is another major obstacle for lightweight general-purpose cross-chain communication, a problem in view of REQ_IL_NF05 (scalability). This means that of all the approaches, the simplicity of bridging approaches (category 4) and blockchain connectors are suitable for the purposes of IoT-NGIN. However, most blockchain connectors work for only certain types of ledgers conflicting with REQ_IL_NF01, while others incur high complexity, leading to scalability and efficiency issues (REQ_IL_NF05 and REQ_IL_NF06, respectively).

Table 2.2 – Comparison of interledger solutions for IoT-NGIN.

Approaches		**Major limitation**	**Requirement confliction**
Category by techniques	Atomic cross-chain trading	Focus on value exchange only	REQ_IL_NF01 Generality
	Transactions across a network of payment channels		
	W3C Interledger Protocol (ILP)		
	Bridging approaches: most	Focus on value exchange only	REQ_IL_NF01 Generality
	Bridging approaches: Flexible Interledger Bridge (FIB)	*Is centralised*	*REQ_IL_NF07 Decentralisation*
	Sidechains	Restrictive assumptions about the consensus or functionalities	REQ_IL_NF01 Generality
	Ledger of ledgers	High complexity of solution	REQ_IL_NF05 Scalability, REQ_IL_NF06 Efficiency
Category by purposes	Chain interoperability for cryptocurrencies	Focus on value exchange only	REQ_IL_NF01 Generality
	Blockchain Engines	For creation of ledgers, not for general-purpose applications	REQ_IL_NF01 Generality
	Blockchain connectors	Most come with restrictive assumptions about the consensus or functionalities, others with high complexity of solution	REQ_IL_NF01 Generality, REQ_IL_NF05 Scalability, REQ_IL_NF06 Efficiency

As can be observed from the comparison, the Flexible Interledger Bridge (FIB) [Wu2021], which has addressed many of the shortcomings of other bridging solutions, provides a lightweight solution that enables a general-purpose atomic data transfer across heterogeneous distributed ledgers without requiring any changes to ledgers themselves. It meets requirements 1-6 and the only requirement it does not satisfy is REQ_IL_NF07 (decentralization) as FIB is a centralised solution running on a single node.

However, the extendable design of FIB and the open-source implementation of FIB provided by the SOFIE project [SOF2021] makes it possible to extend FIB into a *Decentralised Interledger Bridge (DIB)* with a reasonable effort by designing a suitable coordination solution for the interledger nodes - while still retaining full application compatibility with FIB through identical APIs.

2.3 Decentralised Interledger Bridge (DIB)

This section introduces the high-level design of the Decentralised Interledger Bridge (DIB). First, the section describes the design of the Flexible Interledger Bridge (FIB), which functions as the basis for the DIB design. Then, the key missing property, decentralisation, is analysed, and finally, the high-level design of DIB is described. The detailed DIB design will be described in D5.4.

2.3.1 The foundation: Flexible Interledger Bridge (FIB)

There were two key design choices in the Flexible Interledger Bridge (FIB) design: 1) how is the functionality divided into on- and off-ledger parts, and 2) how are multiple ledger types supported? As to the first choice, in a bridging approach [Sir2019] the solution typically contains a software component running off-ledger and smart contracts on-ledgers. This way the solution benefits from the flexible off-ledger logic while still providing direct on-ledger access to the functionality. The FIB design chose to keep the off-ledger part as simple as possible, which means it simply forwards the data payload to the correct destination and signals the acknowledgements back but performs no other processing. This way, application specific logic is left to be implemented by on-ledger transactions, which include the triggering of an interledger transaction, the accepting/rejecting it at the destination, and committing/aborting the transfer at the source. Specifically, interledger transactions can be triggered by pre-defined changes on the source ledger, typically events.

This choice of no processing on the data payload off-ledger leads to following key advantages: transparency, auditability, extensibility, and scalability, all properties necessary for the DIB, as well. First, since the triggering of an interledger transaction and corresponding data processing both happen on the connected ledgers, an external party monitoring both ledgers can easily verify that the FIB node has performed its task correctly, thus ensuring the transparency and auditability. Second, support for different application logic can be implemented via smart contracts without any modifications to the off-ledger design, improving the extensibility of the FIB design for different types of ledgers. Finally, by minimising the processing in the off-chain part, it is easy to achieve high throughput already with a single node and multiple independent nodes can be used in parallel to increase the throughput.

The other design choice is about the support for heterogeneous ecosystems with a single design. Since there is no processing of incoming data off-ledger, no assumption is held on the type of the data. The FIB takes a type-agnostic byte array as the data payload, making it usable by any type of distributed ledgers (as long as the application logic at both ends are aware of the encoding scheme). Multiple data items can be also packed into a byte array (and unpacked at the destination) when multiple data items need to be passed in a single interledger transaction.

However, different ledgers provide different types of functionalities, and some do not even support smart contracts. Working via *adapters* for handling different ledgers and their native APIs is, therefore, a flexible choice to satisfy the Extensibility requirement. To that end, the interfaces of adapters with the Source ledger relate to the operations of 1) triggering interledger transaction from predefined changes (typically by listening for events) and 2) commiting the transaction when successfully accepted by the other end, or aborting otherwise. On the other hand, the interfaces for the Destination ledger involve the operation to accept or reject the data transfer.

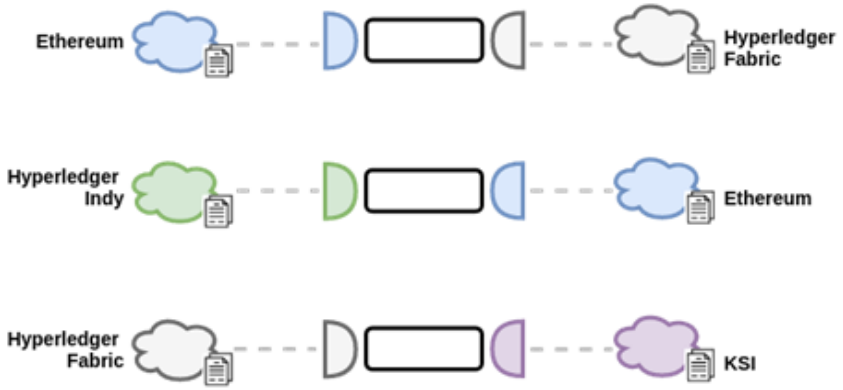

Figure 2.9 – Flexible Interledger Bridge connecting various DLTs.

The design choices discussed above led to the high-level design of the FIB shown in Figure 2.9 that consists of a *Core* and *adapters* for *Source/Initiator* and *Destination/Responder* ledgers, respectively. The *Core* (illustrated as the white boxes in the middle) is in charge of passing the data payload of interledger transactions from the Source ledger to the configured Destination ledger. The *Initiator* adapter (illustrated as the semicircles on the left) is used for interacting with the source ledger, including listening for subscribed changes that trigger the transfers and committing those when finished, while the *Responder* adapter (illustrated as the semicircles on the right) is used for destination ledger interaction, specifically accepting/rejecting incoming data transfer according to application logic. This way a uniform way of interacting with the *Core* was ensured by abstracting the diversity of all ledgers away with the two types of adapters.

The design described above has been implemented and released as open-source software by the SOFIE project [Spi2021]. Currently, there are adapters for Ethereum, Hyperledger Fabric, Hyperledger Indy, and Guardtime KSI [Bul2013]. On ledgers such as Ethereum and Hyperledger Fabric, the Interledger node communicates with a smart

contract via the interfaces defined for data transfer protocol. The interfaces in Solidity for Ethereum are shown below in Figure 2.10.

```
contract InterledgerSenderInterface {
    event InterledgerEventSending(uint256 id,
        bytes data);
    function interledgerCommit(uint256 id) public;
    function interledgerAbort(uint256 id, uint256
        reason) public;
}
```

```
contract InterledgerReceiverInterface {
    event InterledgerEventAccepted(uint256 nonce);
    event InterledgerEventRejected(uint256 nonce);
    function interledgerReceive(uint256 nonce, bytes
        memory data) public;
}
```

Figure 2.10 – Interledger sender and receiver interfaces for Ethereum.

However, on some ledgers such as Hyperledger Indy and KSI, where smart contracts are not available, the adapter communicates with the ledger directly. Both the *Initiator* and *Responder* for Ethereum and Hyperledger Fabric are included, while only the *Initiator* for Hyperledger Indy and *Responder* for KSI have been implemented. Finally, any other type of DLTs can be supported as long as similar interfaces are defined and implemented on the ledger together with corresponding adapters for interaction.

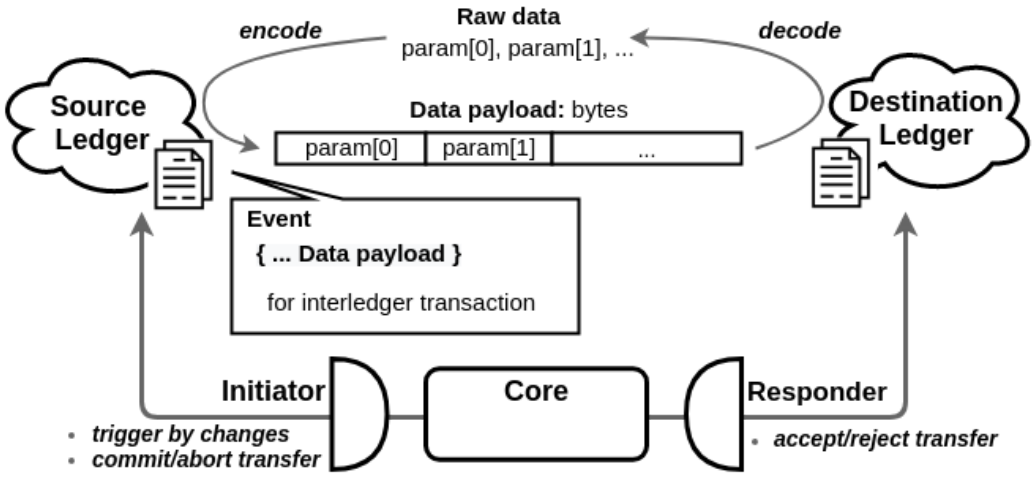

Figure 2.11 – Data flow of interledger transaction utilizing FIB.

The process of executing a complete interledger transaction via FIB is a two-phase commit protocol that consists of the following steps as illustrated in Figure 2.11.

1. The source ledger encodes the application data before the transfer. It is assumed that the encoding schema is known to both sides of the communication, so that both have the same understanding of the data.
2. The source then triggers the transfer by a predefined change on the source ledger, typically by emitting an event that carries the encoded data.
3. The transfer is passed via the FIB component, where the data payload is treated as a bytes array so that a uniform flow is adopted no matter what types of distributed ledgers are connected.
4. The destination side (smart contract) decodes the incoming data.
5. Destination decides to accept/reject based on the content of the data and business logic in the smart contract and calls the accept/reject transfer function of the FIB component
6. Finally, the source commits/aborts the whole data transfer according to feedback from the destination side.

2.3.2 The Issues of centralization

The key limitation of FIB is the centralization of its design, which causes two major issues. First, FIB only provides post-hoc auditability of the interledger transactions. This means that in case anything goes wrong and a transaction is not properly concluded (either intentionally or unintentionally), it is only possible to spot it by checking the transaction history on the related ledgers, but there is no support for real-time monitoring. Moreover, FIB can be a single point of failure as there is no redundancy in the design.

As described above, the FIB design consists of the Interledger Core, the Initiator and Responder adapters for multiple different ledgers. For simplicity, an Interledger Core, together with the adapters used for enabling a particular *connection* (=flow of data) between endpoints (i.e. specific smart contracts on the source and destination ledgers) can be defined as an *interledger instance*, which is illustrated in Figure 2.13. A computer that runs such interledger instances is called an *interledger node*.

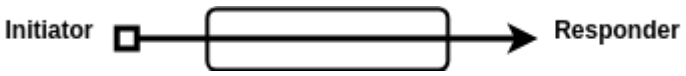

Figure 2.12 – Interledger instance.

The FIB design assumes that 1) the running Interledger node has to be trusted by the parties using it; 2) it will not be corrupted by an external (malicious) party; 3) the node always behaves correctly. However, these are not necessarily the case in many practical applications. Specifically, a single node bridge controller could misbehave eg. in the following ways:

1. ignore an incoming interledger transaction
2. modify the incoming data in an interledger transaction
3. report wrong ledger transaction result within the bridge

A possible improvement to mitigate the trust upon the component is to share the status of all the transfers within a consortium that jointly provides the *interledger service*. As long as consensus can be reached on the transfer status, no single party can forge or conduct invalid data transfers.

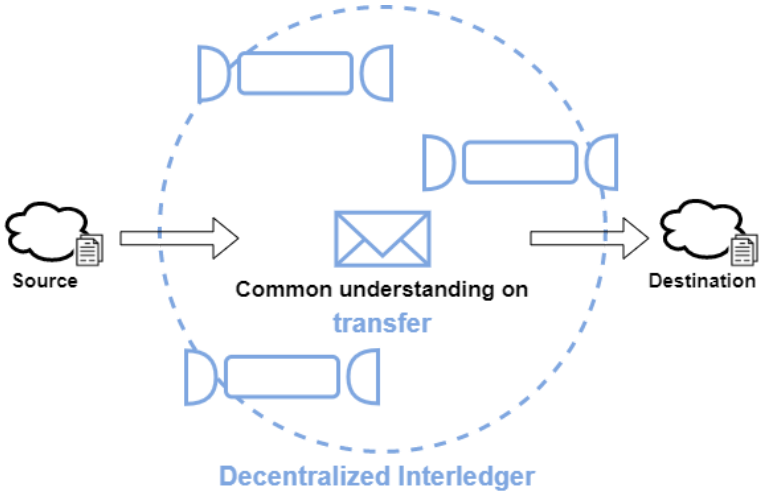

Figure 2.13 – Share transfer status among a consortium of partners.

This naturally leads to the decentralised architecture shown in Figure 2.13. Compared with FIB, the decentralised architecture provides *distributed trust* by relying on a consortium of parties (the exact same approach used by the DLTs), while also improving the robustness of the interledger data transfer via redundancy.

Technically, such a major architecture change is actually quite feasible as the FIB design has already anticipated such a development: the change only affects the Interledger Core, which manages the transactions, so all the Adapters remain unaffected. This also means that the change is completely invisible to the applications utilising FIB: they can simply switch to using a DIB-based interledger service without changing anything in the application and benefit from the reduced trust requirement and increased robustness.

2.3.3 The DIB architecture

The architecture of the DIB is based on the following assumptions:

1. DIB is run by a consortium of parties, which do not necessarily trust each other.
2. Each party runs one or more interledger node(s) that participate in the interledger transactions.
3. All nodes are equal; there are no super-nodes with more functionality or rights.

4. Endpoints of the transactions, which typically are smart contracts on the source and destination ledgers, are able to implement the interfaces required by the Interledger.
5. All nodes in a consortium have the same full access to the endpoints, including both read and write operations.

The key concepts of the DIB architecture have been described in Table 2.2.

Table 2.3 – Concepts relevant to DIB architecture.

Concept	Symbol	Type	Description
Endpoint	E_s/E_d	Smart contract (account able to make ledger transactions in general)	An endpoint E_s/E_d on either source or destination ledger is the application logic that initiates or receives the interledger data transfer, typically it is in the form of a smart contract on these ledgers
Connection	C_i	N/A	A connection is a communication configured between endpoints E_s and E_d located at source and destination ledgers
Interledger instance	I_i	Python module	An interledger instance I_i is a software module that conducts communication between source and destination distributed ledgers of a connection C_i to fulfil a general-purpose data transfer
Interledger node	N_x	Server	An interledger node N_x is a (server) computer controlled by party x that has access to the DSM layer of a consortium and hosts interledger instances for connections
DSM	DSM	Ledger	A Decentralised State Management (DSM) layer is an Ethereum consortium ledger that is set up to manage the states of interledger transactions of common interest by consortium partners
Smart contract	SC_i	Smart contract	A smart contract SC_i on DSM is the code to manage and synchronize the status of interledger transactions among bridges for a particular connection C_i
Transfer entry	t	N/A	A transfer entry is an instance of a interledger transaction on the DSM layer

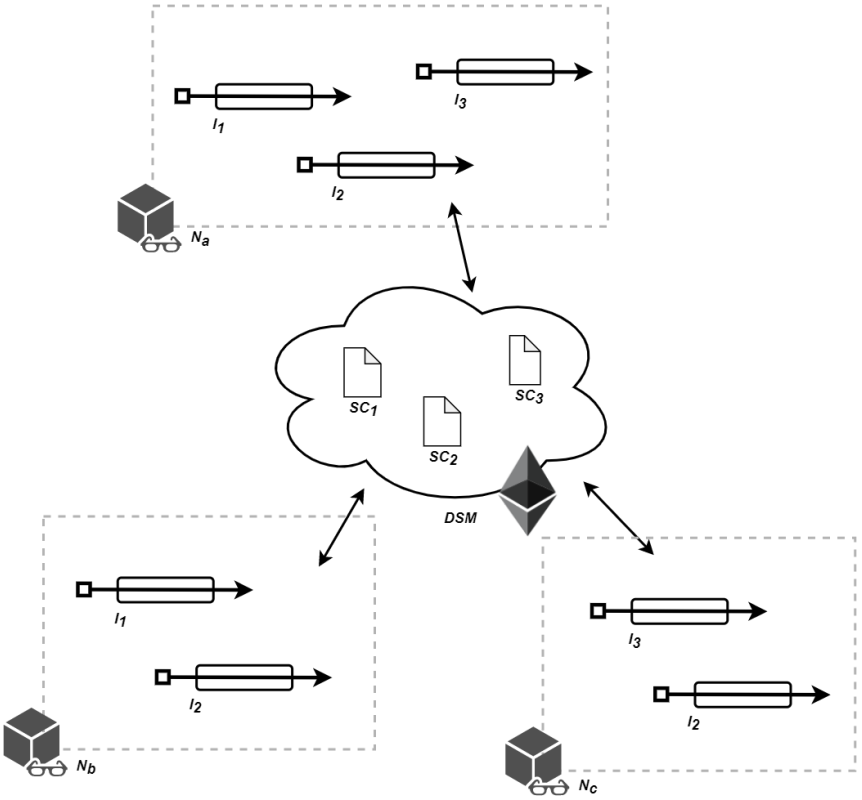

Figure 2.14 – High-level design of decentralised interledger.

The high-level structure of the DIB design is illustrated in Figure 2.14. The key new element in the architecture is the Decentralised State Management (DSM) layer in the center for synchronizing the state of individual interledger transactions between the nodes. In this design, one of the nodes is chosen to carry out the interledger transaction while the other nodes monitor that everything proceeds correctly, and should the chosen node fail to perform, the spares will automatically take over the duties for that transaction.

Currently, the implementation of DIB architecture is under active development, and will be released later as an open source software, together with the corresponding unit tests and usage documentation. The key design choices and implementation details will be reported in the upcoming deliverable D5.4.

3 Self-Sovereign Identity Technologies

This section discusses self-sovereign identities (SSI), provides their state-of-the-art, and describes how the IoT-NGIN pilots will be supported with privacy-preserving SSIs.

3.1 Evolution of digital identities: the way to SSI

The importance and role of digital identities as an enabler in digital transactions has been discussed in multiple studies. Ferdous et al [Fer2019] argue that "with the proliferation of online services ... the management of identities of users and services has taken a central stage in many ways" and that digital identities have become "foundation upon which different online services are built". Similarly, Wagner et al [Wag2018] argue that digital identity is "at the core of all transactions and interactions between natural persons, legal entities, and other things". As discussed in ISO24760-1 [ISO2020], the "proper management of identity information is crucial to maintain security of the organizational processes" and for individuals "correct identity management is important to protect privacy".

3.1.1 Digital identities

According to Cameron [Cam2005], digital identities can be used to (1) convey an identifier to be used to uniquely identify an individual entity, (2) assert that a subject knows a given key, (3) convey personally identifiable information, such as name, address, date of birth, or citizenship, (4) convey information that the subject is part of a group, or to (5) state that a subject has certain capability. Wagner et al [Wag2018] discuss different kinds of identities that individuals may have, such as identities related to the personal life (e.g. father, husband), social life (e.g. employee, football player), or state-issued identities (e.g. citizenship). Examples of use cases where digital identities may be used are e.g. opening a bank account, performing Know Your Customer (KYC) and Anti Money Laundering (AML) checks, reusing bank customership identity for other services, digital education transcript, and registering for an online course or other e-learning.

Formal definitions of digital identities vary significantly in existing literature. Cameron [Cam2005] defines *digital identity* as "*set of claims made by one digital subject about itself or another digital subject*", whereas IDPro Lifecycle [IDPro2020] argues digital identity as being "defined as a unique identifier together with relevant attributes required to enable a digital transaction to generate value". NIST Digital Identity Guidelines [Gra2017] defined digital identity in the following way: "digital identity is the unique representation of a subject engaged in an online transaction. A digital identity is always unique in the context of a digital service". Common understanding between definitions is that the subject itself does not need to be a natural person, but instead it can be "a person, organization, device, hardware, network, software, or service" as defined in NIST guidelines or "a person or thing

represented or existing in the digital realm which is being described or dealt with" as defined by Cameron [Cam2005].

The general term for *all subjects* is that they *are different types of entities*. ISO24760-1 [ISO2020] defines an entity as an "item relevant for the purpose of operation of a domain that has recognizably distinct existence" whereas Ferdous et al [Fer2019] discuss an entity as being "a physical or logical object which has a separate distinctive existence either in a physical or logical sense". ISO24760-1 [ISO2020] provides examples of different kinds of entities, such as "a person, an organization, a device, a group of such items, a human subscriber to a telecom service, a SIM card, a passport, a network interface card, a software application, a service or a website".

In current literature, the technical representation of digital identities often decomposes them into *identifiers* and *claims*. E.g. Allen defines digital identities as "A representation of an entity. It can include claims and identifiers." [Allen 2016]. Claim is often used interchangeably with the term *attribute* so that e.g. NIST guidelines use the following definition for digital identities: "An attribute or set of attributes that uniquely describe a subject within a given context". Definition of identifiers and claims is often complemented with credentials which define how identifiers and claims are grouped together to be used in digital transactions. Allen defines credentials as incorporating "one or more identifiers and numerous claims about a single entity, all authenticated with some sort of digital signature" [All2016]. Credentials are also often used in the context of *authentication* which according to ISO24760-1 [ISO2020] refers to the "formalized process of verification that, if successful, results in an authenticated identity for an entity". In case of authentication, credential is often referred to as "representation of an identity for use in authentication" ISO24760-1[ISO2020].

ISO24760-1 [ISO2020] defines identifiers as an "attribute or set of attributes that uniquely characterizes an identity in a domain". Similar specification is provided e.g. by Allen [All2016] in the form "A name or other label that uniquely identifies an identity." or by Wagner et al [Wag2018]: "An Identifier is something that enables an individual, organization, or thing to be discovered and identified in a given context". Examples of identifiers provided in ISO24760-1 [ISO2020] include e.g. "A name of a club with a club-membership number, a health insurance card number together with a name of the insurance company, an email address, or a Universal Unique Identifier (UUID)". CWA17525 [Kal2020] distinguishes between static identifiers and dynamic identifiers. Static identifiers include e.g. date of birth that never changes, whereas dynamic identifiers change over time, and they can include e.g. your passport number. Another distinction provided by CWA17525 is between self-issued (e.g. DID - decentralized identifier) and automatically given (e.g. national security number issued by government) identifiers. Additionally, the IDPro “Identifiers and Usernames” article [IDProID] discusses internal and external identifiers. Internal identifiers offer the way an identity management system refers to a digital identity whereas external identifiers offer the means by which a person in control of a digital identity refers to that identity when interacting with a system.

Cameron [Cam2005] defines a claim as "an assertion of the truth of something, typically one which is disputed or in doubt". Distinction between claims and attributes is not always clear, but Wagner et al [Wag2018] provides the definition between these two in the following way: "A Claim is a statement or assertion that one DID Subject, such as a person or organization, makes about itself or another DID Subject. The Claim will relate to one or more attributes about a DID Subject". Allen groups different types of claims into facts, opinions, and something in between [All2016]. Facts are e.g. a person's age, opinions are e.g. rating of the person's trustworthiness, and an example of a claim that is between facts and opinions is an assessment of a skill. ISO24760-1 [ISO2020] groups different types of attributes into the following categories: information about physical existence, information describing the entity's evolution over time, information intrinsic to the physical existence of the entity, information assigned to the entity, or reference to an object that represents identity information for the entity.

Current literature often defines credentials as a way to combine claims with identifiers. This definition is used e.g. in Sovrin Glossary [Sg-v2] by describing credential as "A digital assertion containing a set of Claims made by an Entity about itself or another Entity." or by Wagner et al [Wag2018] in the form "A Credential is a set of one or more Claims about a Subject". Sovrin Glossary provides some examples of credentials, including college transcripts, driver licenses, health insurance cards, and building permits. To this end, credentials are often tied to their counterparts in the physical world, as defined by Allen [All2016]: "credentials refer to the state-issued plastic and paper IDs that grant people access in the modern world". A special case of credential usage is authentication, in which case the credential is issued by a Credential Service Provider (CSP) and according to NIST guidelines: "a credential binds an authenticator to the subscriber, via an identifier, as part of the issuance process. A credential is stored and maintained by the CSP, though the claimant may possess it". An authenticator in the case of authentication can be seen as a special case of a claim.

3.1.2 Evolution of digital identities

Key problem with digital identities discussed by Cameron is related to the way the Internet has been built "without a way to know who you are and who you are connecting to". This means that an essential capability of digital identification has been missing and therefore "Identity one-offs" have been the model for digital identities in the past [Cam2005]. However, as defined by Ferdous et al [Fer2019]: "The landscape of identity management has gone through an evolutionary path". Their study, and the studies by Allen [All2016] and Avellaneda [Ave2019] indicate that an evolution in digital identity models has occurred through the following four phases: centralised identity, federated identity, user-centric identity, and self-sovereign identity.

As defined by Allen [All2016], *centralised identities* were the identity model of the early days of the Internet where "centralized authorities became the issuers and authenticators of

digital identity". These were the "Identity one-offs" mentioned by Cameron [Cam2005] where, as defined by ISO24760-1 [ISO2020], "a fully centralized system has a single identity register and a single point of control over enrolment and access to the stored identity information". According to Allen [All2016], this led to problems such as "granting control of digital identity to centralized authorities of the online world suffers from the same problems caused by the state authorities of the physical world: users are locked in to a single authority who can deny their identity or even confirm a false identity". Another aspect related to scalability is raised [Ave2019]: "This works fine as long as users only need to deal with a small number of centralized systems. But once the Internet took off, users were faced with dozens, then hundreds of accounts, all demanding their own usernames, passwords, security and privacy policies, and account maintenance". Other names for this identity model are e.g. "Isolated User Identity (SILO) Model" used by Ferdous et al [Fer2019] and "Siloed / Traditional" model used by [Ruff].

Next phase of digital identity model evolution is discussed by Allen [All2016]: "The next major advancement for digital identity occurred at the turn of the century when a variety of commercial organizations moved beyond hierarchy to debalkanize online identity in a new manner". This led to the introduction of *federated identities* where according Allen [All2016], "Federation improved on the problem of balkanization: users could wander from site to site under the system. However, each individual site remained an authority". NIST guidelines define federation as "a process that allows the conveyance of identity and authentication information across a set of networked systems.", whereas IDPRo Lifecycle [IDPro2020] uses the definition "An organization relies on the digital identity and lifecycle management processes of another organization". According to NIST guidelines, "Federated architectures have many significant benefits, including, but not limited to: enhanced user experience, cost reduction, data minimization, pseudonymous attribute assertions". There is discussion about federated identity protocol standards that were used to implement the concepts of federated architectures [Ave2019]. These included e.g. "the Security Assertion Markup Language (SAML), OpenID and OpenID 2.0 (URL-based identity), and OpenID Connect (based on the OAuth authorization framework)". Problems with federated identities have been raised e.g. by Wagner et al [Wag2018]: "solutions create single points of failure and correlate the user's activity over time and across contexts, requiring the user to trade their privacy for convenience". Additional names for the federated identity model include e.g. the "Third-Party IDP" model used by Ruff [Ruf2018].

As defined [Ave2019], the main concern related to federated identity models has been "the fundamental limitation ... that the identity provider(s) is at the center". This along with the need raised by Allen [All2016] for individuals "to have the right to control his or her own online identity" were the main drivers for *user-centric identities*. ISO24760-1 [ISO2020] defines an identity management system as being user-centric "when it allows the entities to play an active role in the management of the identity information stored in the identity register". Additionally Allen defines user-centric identity model as turning "centralised identities into interoperable federated identities with centralised control, while also respecting some level

of user consent about how to share an identity (and with whom)" [All2016]. Even though the user-centric identity model provides improvements compared to federated identities in the form of improved control, it is still similar to federated model through the reliance on centralised identity providers.

Finally, Allen defines *self-sovereign identities (SSI)* as the "next step beyond user-centric identity" where "the user must be central to the administration of identity" and "rather than just advocating that users be at the center of the identity process, self-sovereign identity requires that users be the rulers of their own identity" [All2016]. The notion of control with SSIs is defined as "the means for users to control personal information flow during digital interactions [Ave2019]. In effect, SSI adds security controls to enable entities to agree on the nature and context of shared information for an online interaction". The main benefit compared to the reliance on centralised identity providers is described by Naik et al [Nai2020]: "it removes the third-party IDP and offers a direct connectivity between a user and organisation". Ruff [Ruf2018] further argues that SSI "is a two-party relationship model, with no third party coming between you and the organization, now considered your "peer"". This is why Ruff [Ruf2018] also uses the term "Peer-to-Peer" to describe the SSI model.

3.1.3 Self-Sovereign Identities

Formal definitions of SSIs are provided e.g. in CWA17525 where a SSI is described as being "created and maintained by an individual themselves but also verified by decentralised issuers for validity in different contexts and use cases". Sovrin Glossary uses the definition "An identity system architecture based on the core principle that Identity Owners have the right to permanently control one or more Identifiers together with the usage of the associated Identity Data". According to Laatikainen [Laa2021], SSI can thus be seen as "an emerging concept that can be viewed as (1) an identity management system, (2) a human-centric data management paradigm or (3) an identity protocol".

Main source of requirements for SSIs originates from Allen [All2016] who has created "a group of principles specific to self-sovereign identity" whose main purpose is to offer "a departure point to provoke a discussion about what's truly important". Allen's principles [All2016] for SSIs include: (1) Existence, (2) Control, (3) Access, (4) Transparency, (5) Persistence, (6) Portability, (7) Interoperability, (8) Consent, (9) Minimisation, (10) Protection. These have been built using Cameron's essential laws [Cam2005] for managing digital identities as the basis which include: (1) user control and consent, (2) minimal disclosure for a constrained use, (3) justifiable parties, (4) directed identity, (5) pluralism of operators and technologies, (6) human integration, and (7) consistent experience across context. [Kal2020] has extended the requirements with "universal capabilities ... considered for the next generation of digital identities". These include e.g. persistent digital unique identifier for a person, capability to link various digital identities online and in various systems to the universal identity, possibility to represent other digital identities, e.g for children, people with

special needs or under legal custody; detachment from the hardware devices and user interfaces, possibility to support ethical and user-centric models for fair data use, support for both centralised and decentralised identity and data architectures, capability to enable the use of digital rights, consents, permits, semantics and the linking of data based on the authenticated digital identities; independency from the issuers' own customership and solutions, and semantic classification of identities and their reference data needed to promote the use of AI and automation of trust for the machines. Additional classifications of SSI requirements are made e.g. by Andrieu [And2016] who emphasises control, acceptance, and zero cost, and by Ferdous et al [Fer2019] who categorises SSI requirements under high-level properties of: foundational property (existence, autonomy, ownership, access, single source), security property (protection, availability, persistence), controllability property (choosability, disclosure, consent), flexibility property (portability, interoperability, minimisation), and sustainability property (transparency, standard, cost).

3.1.4 Privacy-preserving identities

There are many issues with the current digital identity solutions. First, a separate identifier is usually required by different services, requiring maintaining a large amount of separate identifiers, which then often leads to reusing email addresses or phone numbers as identifiers. While some social network providers support single sign-on [social-login], they have multiple issues including lack of privacy and control by the user, and vendor lock-ins, since only large companies could provide these services, and it is difficult for the user to move away from them if the user is already using their single sign-on solution. There exist some federated identity solutions like eduroam [eduroam], however they can only be used in limited environments. Finally, it is difficult and complicated to provide proofs digitally in a privacy-preserving way that does not reveal any unnecessary information.

In practise, the same centralised identifier such as email address or the phone number is often used in multiple services, which allows easy tracking of user activities by both the various services used and by identifier provider (phone operator, email hosting service).

The privacy-preserving identity solution should allow users to avoid tracking by third parties as much as possible and disclose only a minimal amount of information necessary. In practise, this requires self-sovereignty, which means that identifiers can be created and modified directly by the owner, without reliance on any third party.

3.1.5 Requirements of IoT-NGIN

The work in the IoT-NGIN project focuses on using the identities for all the different types of entities in IoT solutions. Of the different entities, the Internet of Things (IoT) devices require special attention due to their central role and sometimes more constrained resources. First, IoT devices are often used for access control and other interaction with the real world (controlling machinery, etc.), which means that security breaches can cause physical

damage and risks. Furthermore, identities of the IoT devices can allow other parties, such as the manufacturer, to track activities of the owner of the device. Finally, many IoT devices are personal (smartwatches, sensors at home, etc.) and their identities can also be used to track and collect personal data about individuals.

An identity solution that relies on public key cryptography is the most convenient one from the security and management point of view. While the solution based on symmetric cryptography offers good security, its key management is problematic. The IoT devices may be constrained in several ways, and these constraints must be taken into account when implementing an identity solution [IoT-SSI1], the devices may be constrained in terms of: 1) computation power, 2) amount of energy available, 3) non-volatile storage space for storing program code and cryptographic keys, 4) entropy for generating random cryptographic keys, and 5) user interface.

As analysed in [IoT-SSI1], most of the modern IoT devices have sufficient computational power and energy for performing public key cryptographic operations, making them suitable for public key-based identity solutions such as SSI. However, in many cases the actual devices may not be able to use public key cryptography, since even though the hardware could be capable of it, the software may be not and manufacturers often do not have interest to provide software updates to the existing devices. Furthermore, devices may be extremely constrained in the terms of computational power, and may also lack the user interface, sufficient entropy sources, or space to store cryptographic keys. In such situations, the SSIs can still be used with a proxy-based approach [IoT-SSI2]: a more capable device handles the SSI-related communication with clients while the communication with the actual constrained device is handled using more lightweight symmetric keys.

There is also some older research (preceding SSIs) proposing the use of Idemix Anonymous Credential Service [idemix] to allow actors to disclose only a subset of their attributes, in a similar manner as the selective disclosure [iot-idm1][iot-idm2].

The main use cases and scenarios where privacy-preserving SSI solutions are planned to be used are related to the Human-Centred Twin Smart Cities Living Lab, Smart Agriculture IoT Living Lab, Industry 4.0 Use Cases & Living Lab trials described in IoT-NGIN document "D1.1 Definition and Analysis of Use Cases and GDPR Compliance". Key capabilities offered by the privacy-preserving SSI solutions to fulfill the functional and non-functional requirements of the trials are related to following:

- Ability to identify and discover different entities (natural persons, organisations, things, services).
- Ability for natural persons to give informed consent.
- Ability for end-users to conduct data minimisation through selective disclosure when sharing their personal data.
- Ability for entities to manage their digital identities and to delegate identity management to trusted parties.

- Ability to authorise other entities to perform operations and access data on behalf of the entities.

Table 3.1 lists the functional requirements set for the developed SSI solution with references to the functional and non-functional requirements from use cases listed in "D1.1 Definition analysis of use cases and GDPR Compliance".

Table 3.1 – Requirements of SSI solutions.

ID	Description	References
REQ_SSI_F01	**Entity Identification** Entities (end-users, organisations, things, services) need to be identifiable using persistent identifiers, one-time transient identifiers, or pairwise pseudonymous identifiers when identifying themselves in interactions with other entities. In case of end-users, using transient or pairwise-pseudonymous identifiers ensures privacy preservation through non-correlatability of interactions between distinct entities. It must be possible to perform entity identification through machine-to-machine communications e.g. with RFID, QR codes, or other similar mechanisms.	REQ_SC1_F02 REQ_SC1_NF02 REQ_SC1_NF03 REQ_SC2_F01 REQ_SC2_NF01 REQ_SC2_NF02 REQ_SC3_F01 REQ_SC3_F03 REQ_SC3_F04 REQ_SC3_F05 REQ_SC3_NF02 REQ_SA1_F01 REQ_SA1_F04 REQ_SA1_F08 REQ_SA1_F09 REQ_SA1_F11 REQ_SA1_F13 REQ_SA1_F14 REQ_SA1_F15 REQ_SA1_F16 REQ_SA1_NF01 REQ_SA2_F04 REQ_SA2_F07 REQ_SA2_F13 REQ_SA2_NF01 REQ_IN1_F02 REQ_IN1_F03 REQ_IN1_F06 REQ_IN1_F08 REQ_IN1_F10 REQ_IN1_NF08 REQ_IN2_NF05

REQ_SSI_F02	**Informed Consent** End-users need to be able to provide an informed consent for a service provider to access their social networking and other personal data (e.g. micro-climate measurements). The consent may be issued by a consent provider as a verifiable credential to the service provider that can then generate a proof (verifiable presentation) of the consent when requesting user's data from the social network.	REQ_SC1_NF03 REQ_SC2_NF02 REQ_SC3_F01 REQ_SC3_F03 REQ_SA1_F09 REQ_SA1_NF01 REQ_SA2_F06 REQ_SA2_NF01
REQ_SSI_F03	**Data Minimisation** End-users need to be able to perform data minimisation through selective disclosure so that they only share the information with service providers that the service provider needs.	REQ_SC1_NF03 REQ_SC2_NF02 REQ_SC3_F01 REQ_SC3_F03 REQ_SA1_F09 REQ_SA1_NF01 REQ_SA2_NF01 REQ_IN1_NF08 REQ_IN2_NF05
REQ_SSI_F04	**Entity Discovery** It should be possible to discover sensors / devices / other non-personal things in a decentralised manner. Discovery may be initiated e.g. by scanning a QR code with a reference to a resolver component to discover additional information about the entity. Example of achieving entity discovery is by issuing a DID to each sensor / device / other non-personal thing and building a mechanism for DID document retrieval with DID methods.	REQ_SA1_F02 REQ_SA1_F03 REQ_SA1_NF05 REQ_SA2_F04 REQ_SA2_F08 REQ_SA2_NF04 REQ_IN1_F06
REQ_SSI_F05	**Identity Management** Entities (end-users, organisations, things, services) should be able to manage their digital identities, and associated identifiers and credentials. Additionally it should be possible to delegate the identity management of an entity to a trusted party (e.g. guardian) that can perform identity management operations on behalf of the entity.	REQ_SC1_NF03 REQ_SC2_NF02 REQ_SA1_F04 REQ_SA1_NF01 REQ_SA2_F06 REQ_SA2_F09 REQ_SA2_NF01
REQ_SSI_F06	**Entity Authorisation**	REQ_SA1_F02

	It should be possible to affiliate end-users to organisations (e.g. through employment contracts) and sensors / devices / other things (e.g. ownership of the device) so that the affiliation information can be used to authorise access to e.g. organisational resources or devices themselves.	REQ_SA1_F03 REQ_SA1_F04 REQ_SA1_F13 REQ_SA1_F14 REQ_SA1_F15 REQ_SA1_F16 REQ_SA2_F09 REQ_SA2_F13 REQ_SA2_F14 REQ_SA2_F15 REQ_IN1_F04 REQ_IN2_F04 REQ_IN3_F01 REQ_IN3_F02

3.2 SSI State-of-the-Art

Laatikainen [Laa2021] raises several concerns over SSI ecosystem development, such as "the fragmentation of the SSI market, lack of standards and regulations, the immaturity of the technology, legal uncertainty, and challenges related to decentralised governance". This raises the need for SSI technology maturation and protocol standardisation. According to Decentralised Identity [Ave2019], "the fundamental elements of this emerging technology stack are already coming into place" with references to W3C Verifiable Credentials (VC) specification [Spo2019] and W3C Decentralized Identifier (DID) specification [DID]. Additional standardisation efforts according to Decentralised Identity [Ave2019] are incubating in "the W3C Credentials Community Group, the Decentralized Identity Foundation, the Hyperledger project, the Rebooting the Web of Trust conference, and other communities".

SSI protocols are defined in Sovrin Glossary to "work interoperably across any number of SSI ledgers, blockchains, or networks". An often referred approach to SSI protocol standardisation is the Trust Over IP Stack [ToIP] that defines "a four-layer architectural stack called the ToIP stack for establishing trust between peers over the Internet and other digital networks". The layer model of the Trust Over IP Stack is shown in Figure 3.1.

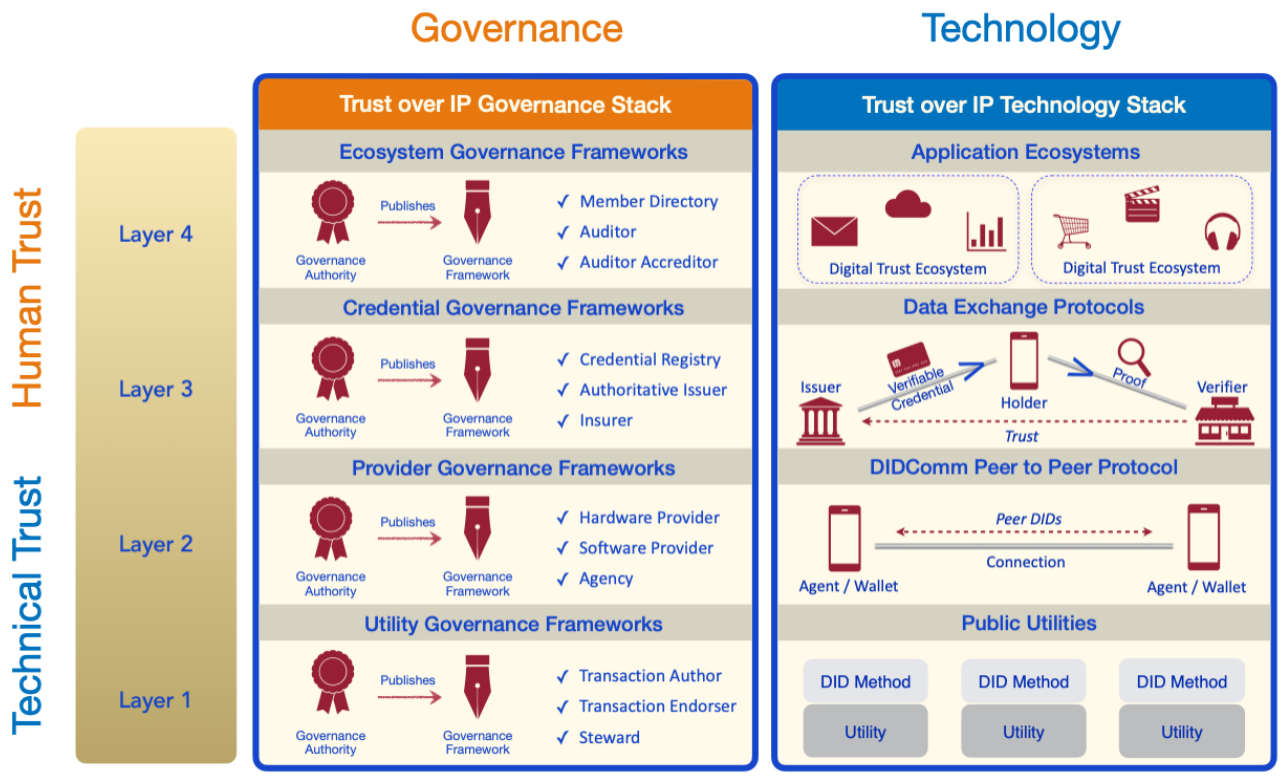

Figure 3.1 – Trust Over IP stack design [ToIPF].

This standardisation effort brings together the previously mentioned DID and VC specifications with the ability to define the interfaces and interoperability between these standards. Layer one of the ToIP "establishes decentralised trust roots using decentralised identifiers (DIDs)" and Layer two defines the DIDComm protocol that offers "a transport-independent protocol that uses DIDs to form and communicate over a cryptographically secure connection". Layer three defines the use of VCs with "a suite of credential exchange protocols based on the W3C Verifiable Credentials standard for cryptographically verifiable digital credentials", and finally layer four "adds cryptographically verifiable governance frameworks".

In order to leverage privacy-preserving SSI capabilities in IoT-NGIN, the following SSI technologies were found most suitable after analysis of the requirements.

From the EU H2020 SOFIE project, following components will be utilised:

- Privacy and Data Sovereignty (PDS) Component (https://github.com/SOFIE-project/Privacy-and-Data-Sovereignty): Allows issuance of access tokens (JWT) based on Hyperledger Indy DID/VCs or pre-shared keys, also support privacy preserving surveys.
- Identity, Authentication, and Authorization (IAA) Component (https://github.com/SOFIE-project/identity-authentication-authorization): acts a

forward proxy to the IoT device (which is assumed to support OAuth2 bearer tokens) and performs user authorization based on DIDs, VCs, and JWTs.

- More examples on how the PDS and IAA components can be used can be found here: https://github.com/SOFIE-project/PDS-IAA/.

Additionally following SSI community contributed components will be utilised:

- Findy Agency (https://findy-network.github.io/docs/) is a Hyperledger Aries compatible identity agent service supporting did:sov DID method. It includes a web wallet for individuals and an API for organizations to utilize functionality related to verified data exchange: issuing, holding, verifying, and proving credentials.

Following sections discuss the DID and VC technologies in more detail, focusing on the technical details of these protocols and how they can be used to offer interoperability across SSI implementations. Additional details are also provided about the specific DID and VC implementations that will be used in the IoT-NGIN solution.

3.2.1 DIDs

DIDs are defined as a "new type of identifier that is globally unique, resolvable with high availability, and cryptographically verifiable" [Hug2019]. As defined by Sovrin Glossary, its main goal is to "enable interoperable decentralised Self-Sovereign Identity management" and according to the W3C DID specification [DID] to be "decoupled from centralized registries, identity providers, and certificate authorities". To that end, the W3C DID specification defines the DID's main design goals to be "decentralization, control, privacy, security, proof-based, discoverability, interoperability, portability, simplicity, and extensibility" [DID]. DIDs are assumed to be most beneficial for "any application that benefits from self-administered, cryptographically verifiable identifiers such as personal identifiers, organizational identifiers, and identifiers for Internet of Things scenarios" [DID].

DID Infrastructure is defined as "as a global key-value database in which the database is all DID-compatible blockchains, distributed ledgers, or decentralized networks" [Hug2019]. In this global key-value database, according to Sovrin Glossary, a DID is "associated with exactly one DID Document" and the DID document "describes the public keys, service endpoints, and other metadata associated with a DID". As such, DID documents are defined as consisting of six components: the DID itself, a set of cryptographic material, a set of cryptographic protocols, a set of service endpoints, timestamps, and an optional digital signature to verify the integrity of the DID document [Hug2019].

Two additional definitions are important to be defined when discussing DIDs: DID Methods and DID Resolution. According to Sovrin Glossary, *DID Method* is a "specification that defines a particular type of DID conforming to the W3C DID specification" and "specifies both the format of the particular type of DID as well as the set of operations for creating, reading, updating, and deleting (revoking) it". DID-Resolution [Sab2021] defines *DID Resolution* as the "process of obtaining a DID document for a given DID". DID Methods and

DID Resolution are important to address the interoperability aspects of DIDs and to address the criticism e.g. from Brunner [Bru2020] arguing that "the interoperability between the currently evolving DID methods is limited and usability issues (such as storing and managing cryptographic key material) are evident". As defined in Trust Over IP Stack, a growing number of "DID methods have already been registered in the informal DID Method Registry" which include methods for "permissionless blockchains, permissioned ledgers, distributed file systems, and ledgerless P2P networks".

Main privacy related risk related to DID usage is discussed in DID specification [DID]: "like any type of globally unambiguous identifier, DIDs might be used for correlation". The specification, therefore, recommends to mitigate this privacy risk "by using pairwise DIDs that are unique to each relationship". Sovrin Glossary defines a pairwise relationship as "a direct relationship between exactly two entities". Pairwise DIDs are an example of Pairwise Pseudonymous Identifiers that are defined by NIST guidelines as "an opaque unguessable subscriber identifier generated by a Credential Service Provider (CSP) for use at a specific individual Relying Party (RP). This identifier is only known to and only used by one CSP-RP pair". The benefits of this approach have been discussed by Ruff [Ruf2018]: "SSI can prevent unwanted correlation by third parties, and even among colluding second parties, by incorporating pairwise identifiers". The technical details of pairwise DIDs are offered e.g. in Peer-DID [Pee2021] with the main use case being to offer a method that "can be used independent of any central source of truth, and is intended to be cheap, fast, scalable, and secure". The main application areas according to Peer-DID [Pee2021] are to establish "private relationships between people, organizations, and things".

Due to the decentralisation characteristics of SSIs, many DID implementations aim to take advantage of blockchains and DLTs as solutions for decentralisation. This may however be problematic when considering personal data. As discussed [Hug2019], "storing any type of PII on a public blockchain, even encrypted or hashed, is dangerous for two reasons: 1) the encrypted or hashed data is a global correlation point when the data is shared with multiple parties, and 2) if the encryption is eventually broken ..., the data will be forever accessible on an immutable public ledger". To this end, the recommendation is to "store all private data off-chain and exchange it only over encrypted, private, peer-to-peer connections" [Hug2019]. This same approach has been discussed in several studies, such as Muhle et al [Muh2018] ("most claims are stored off-chain not publicly available") and Wagner et al [Wag2018] ("Underlying data, including Claims and Credentials, should not be written to or stored on a public blockchain"). Additionally as discussed [Ave2019], "the solutions needed here are not just technical ... but also legal and regulatory".

3.2.2 Agent Communications

In order to allow identity owners to use their DIDs in SSI processes and interactions, it is mandatory to define the secure and privacy-preserving communications protocols that are needed. These are called the DIDComm protocols that Laatikainen describes as specifying

"the communication between DIDs such as connecting and maintaining relationships and issuing credentials, providing proof" [Laa2021]. DIDComm protocols take advantage of peer-to-peer Connections that are according to Sovrin Glossary "cryptographically verifiable communications channels established using an Agent-to-Agent Protocol between two DIDs representing two Entities and their associated Agents". According to Naik et al [Nai2020], an Agent is "a program required for an identity owner or any other participating entity to interact with each other in the SSI process" or as defined in Security Analysis [Kim2021], "an agent is the delegated entity by a DID subject, which is responsible for agent-to-agent DID communication, operation of cryptographic functions of DID identity wallet, and use of credentials with authorized identity sovereignty according to each relationship". According to Aries RFC-0025 [Cur2019], the "Agent Messaging is designed to be transport independent" with existing definitions in place e.g. for HTTP(S), Websocket, XMPP, Message Routing, and other transports.

In addition to agents, an additional component of equal importance related to DID-enabled secure communications is the digital wallet that according to Sovrin Glossary is "a software module, and optionally an associated hardware module, for securely storing and accessing private keys, link secrets, other sensitive cryptographic key material, and other private Data used by an Entity". According to Trust Over IP Stack, the main purpose of digital wallet is to "safeguard sensitive data such as key pairs, zero-knowledge proof blinded secrets, verifiable credentials, and other cryptographic material needed to establish and maintain technical trust" and as defined by Wagner et al [Wag2018] to "provide security and encryption for the personal information and for the actual transaction". An exhaustive list of capabilities of digital wallets has been provided by O'Donnell [Odo2019] including capabilities such as receiving, offering, presenting, organising, and rendering credentials; and managing personas, private connections, emergency access, and offline operations. Guiding principles of digital wallets according to O'Donnell [Odo2019] should be that they are consent-driven, portable, and that they are designed with privacy-by-design, security-by-design, open-by-default principles.

3.2.3 Verifiable Credentials

"DIDs are only the base layer of decentralized identity infrastructure. The next higher layer ... is verifiable credentials (VC)" [Hug2019]. According to Naik et al [Nai2020], "a VC is used to represent similar information on the Web to that of a physical credential in the real world". The formal definition in specification describes VC as a "standard way to express credentials on the Web in a way that is cryptographically secure, privacy respecting, and machine-verifiable" with benefits achieved through VCs being "tamper-evident and more trustworthy than their physical counterparts". As defined by Ruff [Ruf2018], "Verifiable credentials can be issued and digitally signed by any person, organization, or thing and used anywhere they are trusted" and additionally according to Wagner et al [Wag2018], VCs can "express virtually any kind of Claim about an individual or entity, and given the

adequate verification processes and legal acceptance, these Claims can represent anything about the individual or entity who is the subject of that Credential".

As defined by Muhle et al [Mul2018], the verifiability of VCs originates from the "signature of an attestation issuer that has either issued the claim himself or can attest the correctness of it". VC specification defines three main mechanisms for these proofs, including JSON Web Tokens secured using JSON Web Signatures, Linked Data Signatures, and Camenisch-Lysyanskaya Zero-Knowledge Proofs. Additionally Trust Over IP Stack divides currently supported credential exchange protocols into two types based on the types of credentials that they support "those that do not use zero-knowledge proof (ZKP) cryptography, which are easily correlatable, and ZKP credentials that enable credential holders to selectively disclose claims to verifiers without correlation".

The main use case related to VCs, the credential exchange and the associated roles are defined by Trust Over IP Stack as "enabling any issuer to assert any set of claims to any holder who can then prove them to any verifier". As defined by VC specification, this "describes the roles of the core actors (issuer, holder, verifier) and the relationships between them in an ecosystem where verifiable credentials are expected to be useful". The roles and the relationships between them are described below in Figure 3.2:

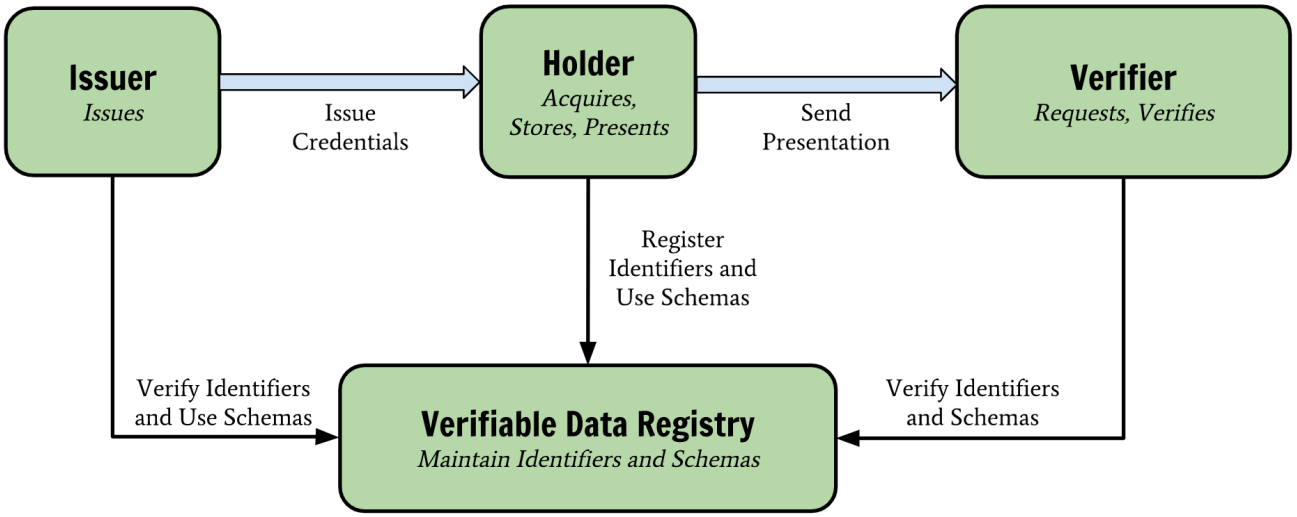

Figure 3.2 – Verifiable Credential related roles and relationships [Spo2019].

In addition to the roles of issuer, holder, and verifier, it is important to define the role of subject, that according to VC specification is the "entity about which claims are made" or according to Wagner et al [Wag2018] is "the individual, entity, or thing that a given Credential is about or relates to". Examples of subjects provided include e.g. human beings, animals, and things. According to VC specification, "in many cases the holder of a verifiable credential is the subject, but in certain cases it is not. For example, a parent (the holder) might hold the verifiable credentials of a child (the subject)".

Wagner et al [Wag2018] shortly define *issuer* as "the individual or entity who issues a given credential". More formal definition is provided in VC specification as "a role an entity performs by asserting claims about one or more subjects, creating a verifiable credential from these claims, and transmitting the verifiable credential to a holder". As discussed by Sorokin, "issuers are typically a government entity or corporation, but an issuer can also be a person or device" [Sor2020]. Examples of issuers defined in VC specification include "corporations, non-profit organizations, trade associations, governments, and individuals". CWA17525 additionally discusses the role of issuer compared to the role of Credential Service Provider (CSP) defined e.g. by NIST guidelines as "a trusted entity that issues or registers subscriber authenticators and issues electronic credentials to subscribers" or by ISO24760-1 [ISO2020] as an "entity responsible for provisioning of a credential to a principal". CWA17525 argues that the "role of the Credential Service Provider (CSP) in the centralized / federated model is equivalent to the role of issuer in the decentralized model".

VC specification defines *holder* as "a role an entity might perform by possessing one or more verifiable credentials and generating verifiable presentations from them". Additional term used e.g. in Sovrin Glossary is called the prover that is defined as "a role played by an Entity when it generates a Zero Knowledge Proof from a Credential". VC specification provides some examples of holders as students, employees, and customers. Sorokin additionally argues that "holders are typically users but can also be organizations or devices" [Sor2020].

Finally, the *verifier* as defined by VC specification is "a role an entity performs by receiving one or more verifiable credentials, optionally inside a verifiable presentation" or as defined by Wagner et al [Wag2018], "the individual or entity who verifies or relies upon a given Credential". Example verifiers discussed in VC specification include employers, security personnel, and websites. As defined in Sovrin Glossary, a verifier is "typically an organization, but it may also be an individual or even a thing—seeking trust assurance of some kind". The term verifier is discussed also in NIST guidelines as "knowing the claimant's public key through some credential (typically a public key certificate), can use an authentication protocol to verify the claimant's identity by proving that the claimant has possession and control of the associated private key authenticator". In this terminology, the claimant is analogous to a holder.

NIST guidelines define that CSP (analogous to issuer) "maintains status information about the credentials it issues". This means, as defined in VC implementation guidelines [Cha2019], that "verifiable credentials may need to be revocable" and that "if an issuer can revoke a credential, verifiers must be able to determine a credential's revocation status". One example mechanism to achieve VC revocation is the Revocation Registry discussed in Sovrin Glossary: "privacy-respecting cryptographic data structure maintained ... by an Issuer in order to support Revocation of a Credential".

Additionally, to support the interoperability aspects of VCs, there is the need to manage data models of the VCs. Sovrin Glossary defines the *data models* as *Schemas* that are "a

machine-readable definition of the semantics of a data structure". Schemas are further used to define *Credential Definitions* that according to Sovrin Glossary are "a machine-readable definition of the semantic structure of a Credential based on one or more Schemas". These mechanisms offer the possibility to manage VC data models in a decentralised setup.

The verifiable presentation passed from the holder to the verifier is according to VC specification "a tamper-evident presentation encoded in such a way that authorship of the data can be trusted after a process of cryptographic verification". Further as defined by Sorokin [Sor2020], "verifiable presentations are digitally signed by the holder and can encapsulate all the information that a verifier is requesting in a single package". Verifiable presentations also support the privacy-preserving mechanism of Selective Disclosure which as defined by Brunner [Bru2020] allows the holder to "select a subset of claims attached to a VC and create a verifiable presentation of those selected attributes".

The two main privacy-preserving capabilities of VCs discussed earlier in this paper include Selective Disclosure and Zero-Knowledge Proofs (ZKPs). ISO24760-1 [ISO2020] defines Selective Disclosure as a "principle of identity management that gives a person a measure of control over the identity information that can be transferred to a third party" or as defined by Sovrin Glossary as "a Privacy by Design principle of revealing only the subset of the data described in a Claim, Credential, or other set of Private Data that is required by a Verifier". VC specification defines ZKPs as "a cryptographic method where an entity can prove to another entity that they know a certain value without disclosing the actual value". As defined [Hug2019], these techniques allow the identity owners to gain "greater control over their personal data" through data minimisation by e.g. only disclosing "that you are over a certain age without disclosing your exact birthdate". An additional example discussed by Wagner et al [Wag2018] includes allowing "to prove the ownership of a credential to the verifier, such as a driving licence without revealing the identifier it has been initially issued to".

Examples of verifiable credentials that can be used in the IoT-NGIN SSI solution to implement the use cases listed in “D1.1 Definition and analysis of use cases and GDPR compliance" include:

- Verifiable credential issued by a certification agency to a traffic or weather data producer that the traffic or weather data that they produce can be trusted.
- Verifiable credential issued by an end-user (possibly through a consent provider) to a service provider that represents the consent that the end-user has given to the service provider to access their personal data in some 3rd party service (resource server). The service provider may generate a proof (verifiable presentation) of the verifiable credential to be passed to the resource server for additional authorisation.
- Verifiable credential issued by an organisation to an end-user representing the affiliation between the organisation and the end-user (e.g. an employment contract). The end-user may create a proof (verifiable presentation) from the

verifiable credential to authorise access to organisational resources made available by a resource server.

- Verifiable credential issued by an organisation to an end-user representing the ownership of a device. The end-user may create a proof (verifiable presentation) from the verifiable credential to authorise access to management of the device.

3.2.4 Alternatives for Resolving DID Documents

A Decentralized Identifier (DID) must be resolvable to the DID Document, which can either be a private or public registry [Tob2018]. Publicly available DID documents are useful for verifying credentials, since in that case the verifier can easily fetch the DID document of the credential issuer containing, for example, the public key of the issuer, credential definition, information about the revocation registry, etc. Public DID registries can also be used for service discovery purposes, to discover DIDs of the public entities.

There are several options for making DID documents available:

- The DID document can be directly disseminated by the user of the DID to relevant parties without a separate registry, *did:self* method uses this approach [did-self].
- The DID document can be stored in distributed ledgers which can be either public (such as Ethereum in case of *uPort* [uport]) or private (such as in the case of *Sovrin* [sovrin]).
- The DID document can be stored on any webserver. Examples of such an approach include *Github DID* [github-did] method and a more general did:web method [did-web].
- The DID document can also be stored in the DNS records along with the DID associated with the domain. *DID DNS* draft [did-dns] proposes storing associated DID within the DNS records, and the similar principle could also be used to store the associated DID document. Using DNS for this purpose is natural, since most online services use hostnames and therefore the user must perform (and trust in) DNS resolution in any case.

European Union's General Data Protection Regulation (GDPR) [gdpr] applies to all personal data and includes among other things "right to be forgotten". The term personal data is defined broadly and it includes "all data which are or can be assigned to a person in any kind of way". In the context of SSIs this basically means that DIDs related to the person, including DIDs of the personal IoT devices, should not be stored on immutable distributed ledgers. Thus, SSI solutions utilizing DLTs such as Sovrin already have a policy to store only DIDs related to public entities (organisations, companies, etc.) to the ledger.

The relevant DID methods that will be utilised in the IoT-NGIN SSI solution are summarised as the following.

- *did:self* (https://github.com/mmlab-aueb/did-self) method does not use DID registry, the DID is an encoded public key and the method supports updating the DID

document and delegating the DID to a third party. An implementation of the method is available for Python.

- *did:key* (https://w3c-ccg.github.io/did-method-key/) is a simple method where the DID is the encoded public key, the method does not use DID registry and it does not support updating or deactivating the related DID document. Implementations of this method are available for Javascript (https://github.com/digitalbazaar/did-method-key-js) and Rust (https://github.com/decentralized-identity/did-key.rs).
- *did:sov* (https://sovrin-foundation.github.io/sovrin/spec/did-method-spec-template.html) is a method developed by the Sovrin Foundation, it uses a Hyperledger Indy ledger as a DID registry. The ledger is used to store public DID documents including public keys and service endpoints, credential schemas and definitions, revocation information, etc. The DID itself is composed of 16-byte uuid which may or may not be derived from the owner's public key.

Additional DID Document resolution technologies that will be utilised in the IoT-NGIN SSI solution are summarised as the following.

- GS1 Digital Link (https://www.gs1.org/standards/gs1-digital-link) allows encoding relevant information about an entity (such as an IoT device) into a GS1 URL, which in turn can be resolved to any kind of document providing more information about the entity. In the scope of IoT-NGIN GS1 Digital Links can be used to discover a Digital Twin document or DID document associated with an IoT device. There exists an open-source implementation of the GS1 Digital Link resolver service (https://github.com/gs1/GS1_DigitalLink_Resolver_CE).

3.3 Privacy-preserving SSI solution for IoT-NGIN

The privacy-preserving SSI solution for IoT-NGIN assumes that the OAuth2 protocol is used for authentication with IoT devices and the IAA proxy supports OAuth2 bearer tokens for accessing the Resource server.

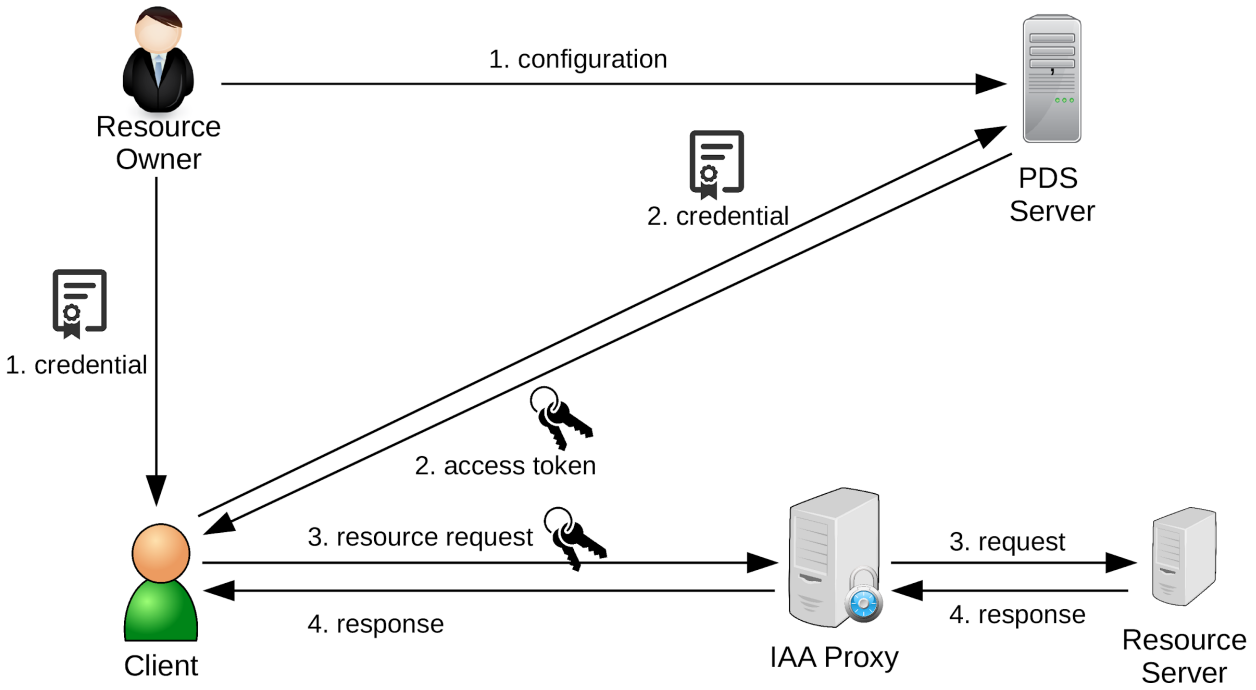

Figure 3.3 – Example of authorisation and authentication flow.

The IoT-NGIN SSI solution is based on the existing PDS and IAA components. The basic flow is described in Figure 3.3 and goes as follows:

1. The Client receives credential or a shared secret from the Owner or party trusted by the owner, denoting that the Client has a right to access the Resource. The Owner also configures the PDS server if necessary
2. (optional) The Client contacts the PDS server, presents the credential received in step 1, and receives the access token (JSON Web Token (JWT)) to access the Resource.
3. The Client contacts the IAA proxy to access the Resource, and presents the credential or access token received in steps 1 or 2.
4. If the credential presented in step 3 is valid, the IAA proxy forwards the request to the Resource server and relays the reply (containing the information that Client wants to retrieve) back to the Client.

Users are free to choose the used DID methods, *did:self* method offers good features and simplicity, *did:key* method is even simpler but it lacks features such as delegation and key rotation, while Hyperledger Indy and *did:sov* method offer most of features (including revocation lists) with added complexity (need to run a ledger).

For expressing consent JWTs or VCs can be used, they express that the user has given a consent to disclose certain information to a certain party. The issued consent should contain at least following information:

- Identification of the consent issuer, i.e. who has issued the consent. This may be the end-user itself or e.g. a dedicated consent management system.

- Identification of the entity who is giving the consent, i.e., the end-user. In case the consent is given by the end-user itself, this will be the same value as the consent issuer.
- Identification of the entity for whom the consent is given.
- The scope of the consent, i.e., the resources for which the consent is applicable.

Additionally, the consent may contain the following information:

- The jurisdiction in which the consent is applicable.
- The validity period of the consent (start and end timestamps).
- The purposes for which the consent is applicable.
- Information about the collection method of the consent.

Support for ACE-OAuth authentication is needed for very constrained devices that do not support public key cryptographic operations in the implementation.

Besides the secure access approach that is described, QR codes can be used to allow users to discover additional information about the IoT device and its services in a straightforward way. GS1 Digital Link standard is used here to provide a permanent identifier for the device, which can then be encoded into a QR code. GS1 Digital Links are then resolved to DID documents, Semantic Twin (as discussed in Section 5.3), or other kinds of documents, such as Web of Things (WoT, https://www.w3.org/WoT/) descriptions of the device.

The detailed solutions for the use cases will be presented in the upcoming deliverable D5.4.

4 Ontologies for IoT

This section discusses the role of ontologies in enabling interoperability for IoT systems, how IoT-NGIN intends to utilize ontologies, and the SAREF ontology chosen as the primary ontology.

4.1 The challenges of interoperability

Today's IoT systems often integrate many heterogeneous devices from different vendors using different networking standards, communication protocols, and data formats, which presents many challenges for the interoperability within and between systems. These interoperability issues can be broadly separated into three layers: Network interoperability, message protocol interoperability, and data annotation/information layer interoperability [DSA15]. Providing interoperability on the network layer is a problem that can be largely solved by means of network gateways. Interoperability between communication protocols is already harder to address, as different protocols possess different characteristics, which can be incompatible. Nevertheless, it could be argued that the overall system architecture should be designed to be independent of the protocol standards [DSA15].

However, a major challenge lies in interoperability on the data level. For example, many IoT devices simply transmit raw sensor data and expect the endpoint to know how to interpret it, which prevents the use of other data endpoints in the system. A better approach in terms of interoperability would be to also transmit information about the meaning of the data to allow true *semantic interoperability*.

This, however, still leaves the issue of finding a way to encode this meaning in a universal and understandable way. Providing interoperability in the information layer requires a common understanding of the domain in question and this can be provided by a common data model, information model, and ontologies.[1]

4.2 Ontologies facilitate interoperability

This subsection introduces ontologies in the context of computer sciences, how the IoT-NGIN wants to utilize ontologies, and finally, identifies the most suitable ontology for the project: SAREF.

[1] It should be noted that the problem of information interoperability is not limited to the IoT domain, but applies to all kinds of complex systems with data exchange. This leads to the conclusion from Ganzha et al. in [Gan2017]: "To be able to apply semantic technologies, one has to have ontologies available".

4.2.1 Ontologies in computer sciences

Different fields of study have a different understanding of the term ontology. In the field of computer science, a number of slightly varying definitions exist [GOS2009], but this document will use the widely adopted definition by Guarino et al. from [GOS2009]:

> *Computational ontologies are a means to formally model the structure of a system, i.e., the relevant entities and relations that emerge from its observation, and which are useful to our purposes.*

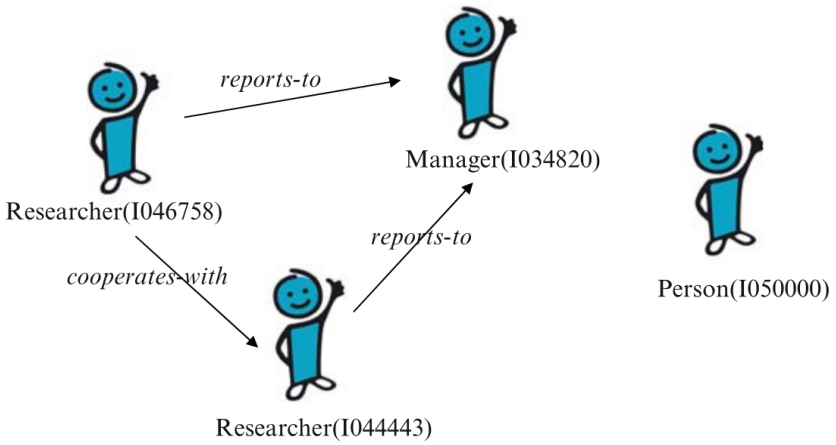

Figure 4.1 – Company relations from [GOS2009].

A simple example by Guarino, Oberle, and Staab, an ontology for business relations in a fictional business domain, from the same publication illustrates the definition (Figure 4.1). The domain is modeled with a list of its basic entities, in this case: *Persons*. *Researchers* and *Managers* are a special subset of the Person-type entities, and all Persons have an ID number. The only relevant relations between Persons in this domain are *cooperates-with* and *reports-to* with the limitations that only Researchers can cooperate with each other and only Researchers can report to Managers.

Even though this example is simple and limited to a very specific domain, it does show most of the general building blocks used in ontologies:

- Concepts (or classes): Here Researcher and Manager are concepts and Person is a super concept of these.
- Instances: There are 4 instances of Persons.
- Relations: reports-to and cooperaties-with are relations between entities.
- Attributes: The ID number of a Person

The study of ontology development processes, ontology life cycles, and tools for these matters is called Ontology Engineering [Gal2009]. Currently, there are many ontologies focused on a specific domain, so a large variety of ontologies with different granularity and

generality can be observed. However, finding a universal ontology for all domains can be considered an almost impossible task, and if possible, it is likely that such a comprehensive ontology would suffer in usability for special domains.

Yet, attempts to ensure some level of universal semantic interoperability exist and have resulted in so-called upper ontologies such as SUMO [NP01]. These ontologies contain terms that are common across multiple or even all domains.

Ontologies can also include subsets or even whole other ontologies into their domain-specific ontology, thus improving interoperability across domains. An example is the OGC GeoSPARQL ontology, which is used in many ontologies to represent geospatial properties.

4.2.2 Use of ontologies in IoT-NGIN

Whilst not being a new technology developed in IoT-NGIN, the project has two important applications for Ontologies. One is the structured analysis of the living labs and the use cases to enhance semantic interoperability in general, the other are the Semantic Twins.

The aim of the first is to classify the new and existing solutions in the living labs by identifying the requirements and specifications of use-cases and then providing recommendations on how suitable semantic technologies can enhance the solutions. This is an ongoing process throughout the project, for which a document with guidelines for semantic technologies in general and specific recommendations for the living labs will be created.

The second application for ontologies are the Semantic Twins discussed in Section 5. These are being developed to provide a uniform semantic description for Digital Twins. Many of the required information in such a description are semantic by nature, and thus, should be formalised as ontology elements. The use of ontological information therefore is a key technology to provide the semantics and especially to fulfil the extensibility requirement for the Semantic Twins (***REQ_TWIN_NF12*** - section 5.1.3).

4.2.3 Review of the State-of-the-Art

The first argument for choosing an ontology in the project context is *standardization,* as using a standardised ontology almost invariably promises better interoperability for applications. In the field of IoT, a multitude of ontologies exist and Bajaj et al. provided a comprehensive study of ontologies [Baj2017] of which more than 25 could be categorised as IoT ontologies.

SAREF (introduced in detail in section 4.3) was identified as the most suitable ontology for the IoT-NGIN already in the project proposal, and the following analysis confirms the finding. Therefore, it will be used as the primary ontology in IoT-NGIN, though it can be complemented with other ontologies, when relevant.

Besides SAREF, numerous other ontologies on the subject of IoT have been standardised as well. Li et al. highlight the most important [19]:

- oneM2M Base Ontology - ETSI standard TS-0012 [One2019]. A base ontology for syntactic and semantic interoperability between oneM2M projects and external systems and devices.
- WoT Thing Description (WoT TD)- W3C Recommendation [Kae2020]. A description of the metadata and interfaces of things in the Web of Things.
- Semantic Sensor Network (SSN) and Sensor, Observation, Sample, and Actuator (SOSA) - a joint W3C-OGC implementation standard [Arm2017], which is intended to model sensors, observations as well as devices and systems.
- Context Information Management (CIM) Information Model by ETSI [Abb2019]. Data centric cross domain ontology for the CIM APIs.

The oneM2M ontology is not meant to be used outside the oneM2M projects, but rather describes a minimal ontology for which other ontologies have to provide a mapping to be compatible with the organization's projects. In fact, SAREF is the default example for such an external ontology and the mapping is provided in the SAREF standard, as well as in the oneM2M standard.

WoT TD was not chosen as primary ontology as it mainly focuses on the data view of the IoT world, and is not suited to describe e.g., physical properties of an object. These are especially relevant in the context of DTs, which are important in this project. However, WoT TD can be extended with SAREF ontologies, making it potentially useful in the IoT-NGIN project.

SSN and the contained SOSA ontology are also suitable ontologies for most IoT applications, however the difference in features to SAREF is low, especially since SAREF extensions like SAREF4AGRI and SAREF4SYST extension already bring in several of these features. Besides this, SAREF development has been more active in the past years with the latest SSN release being from 2017 whilst SAREF was updated last in 2020.

Finally, the CIM Information Model is a simpler base ontology which is also intended to be extended with domain-specific ontologies. However, as of now, no such extension has been standardised. Because a mapping to SAREF is provided by the standard, the SAREF extensions could be used as extensions for CIM to a certain degree, as well. As IoT-NGIN doesn't use the CIM API, there is no advantage in focusing on this ontology. However, parts of all these ontologies could be brought into SAREF by a mapping or by extensions.

4.3 SAREF, the primary ontology in IoT-NGIN

The *Smart Applications REFerence ontology (SAREF)* is an ontology "intended to enable interoperability between solutions from different providers and among various activity sectors in the Internet of Things" [Lau2020], created in close interaction with the industry [DHR2015] and standardised by the ETSI as TS 103 264. Based on the results of a European Commission Study Group, SAREF is designed as a reference ontology and contains recurring concepts for several IoT domains.

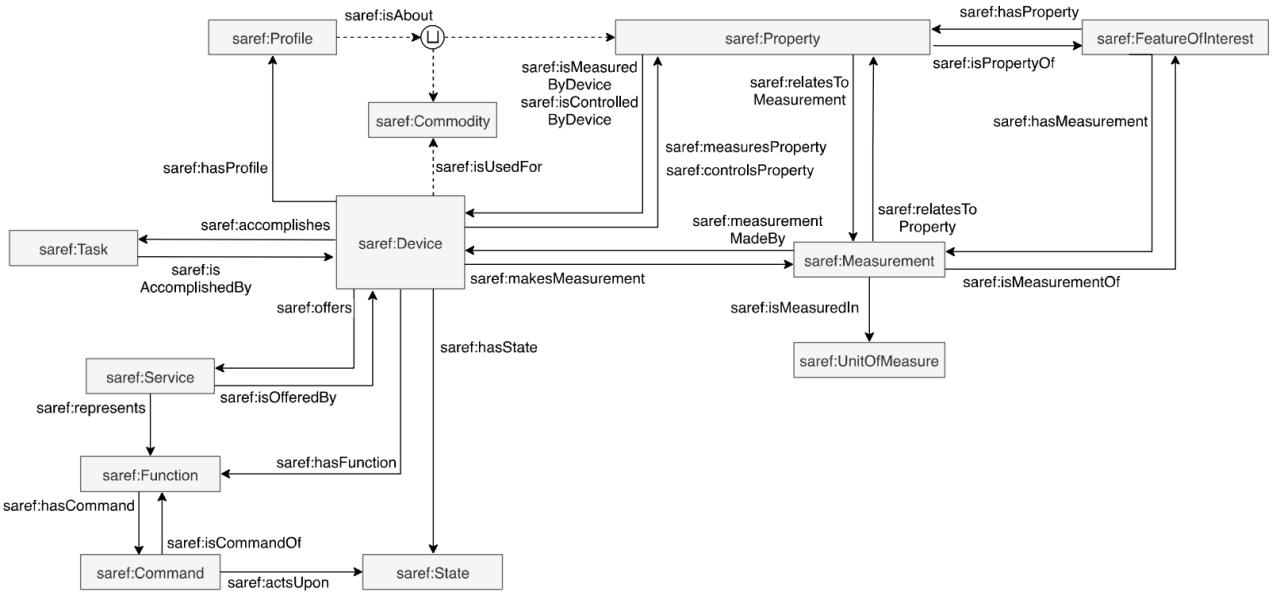

Figure 4.2 – Overview of the SAREF ontology [Lau2020].

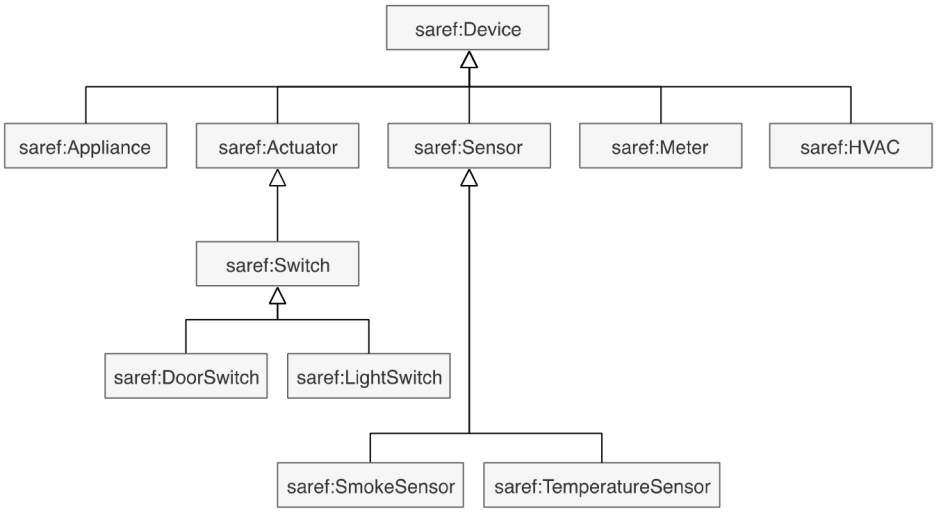

Figure 4.3 – Inheriting properties of the parent classes [Lau2020].

SAREF is centered around the concept of a "Device". It is defined as "a tangible object designed to accomplish a particular task in households, common public buildings or offices". Examples are light switches, temperature sensors or washing machines. Most of the other concepts in the ontology are used to describe the interactions and properties of these Devices. For instance, a device could accomplish a "Task" such as "washing", provide a "Service" or make "Measurements" as shown in Figure 4.2. Devices can be defined further as shown in Figure 4.3 to provide more specialised categories, whereas the specific device types inherit all properties of their superclass.

The SAREF specification claims the following four design goals:

- **Reuse and alignment**: The core concepts for SAREF were collected in a study of 23 so-called "assets" related to energy management or home appliances to harmonize with existing solutions [HDR2015]. Furthermore, a mapping to the oneM2M base ontology and the W3C Semantic Sensor Network ontology exist.

  SAREF also reuses parts of other ontologies, such as the W3C Time ontology or the W3C geo positioning vocabulary
- **Modularity**: Different parts of the ontology can be separated and recombined, depending on the specific needs.
- **Extensibility**: SAREF provides basic ontology building blocks which often need to be refined with custom derivations of existing classes to provide a precise view of a domain. This topic is covered in more detail in subsection 4.3.1.
- **Maintainability**: The authors claim that the modular and extensible design improves the maintainability, as each extension can be maintained individually or could even be replaced by others which provides higher flexibility.

4.3.1 SAREF Extensions

The core part of SAREF was designed with a focus on home appliances, such as light switches or temperature sensors. However, due to its extensibility, a number of standard and non-standard extensions have been developed, which extend the ontology to multiple other domains. At the moment of writing, 10 domain specific standard extension exist:

- SAREF4ENER: Energy domain
- SAREF4ENVI: Environment domain
- SAREF4BLDG: Building domain
- SAREF4CITY: Smart Cities domain
- SAREF4INMA: Industry and Manufacturing domains
- SAREF4AGRI: Smart Agriculture and Food Chain domains
- SAREF4AUTO: Automotive domain (under development)
- SAREF4EHAW: eHealth/Ageing-well domain
- SAREF4WEAR: Wearables domain
- SAREF4WATR: Water domain

In addition, SAREF4SYST is a standardised extension for connected systems and is intended to be combined with the domain-specific extensions where applicable.

These extensions can define domain-specific subclasses for devices, measurements, services etc. such as the s4ener:PowerProfile which is a subclass of saref:Profile, as well as new concepts and classes parallel to the existing ones, for example the SAREF4SYST defines a s4syst:System and a s4syst:Connection class.

Besides the standardised extensions, other custom extensions exist as well with the aim of filling gaps of the standardised extensions or providing support for domains which have not been covered yet. An example of such an extension is SARGON [Hag2020], which extends SAREF to the smart building and electrical grid automation domains by including standards such as IEC 61850 and the Common Information Model (CIM).

5 Semantic Twins

Semantic Twins are being developed in the IoT-NGIN project as a general solution for adding metadata to digital twins. This section describes the motivation for semantic twins, the state-of-the-art in digital twin metadata, and the high-level design of the Semantic Twin solution.

5.1 Motivation

This section describes the motivation for adding metadata to digital twins and the needs of IoT-NGIN Living Labs, and formalizes them into requirements for the Semantic Twins.

5.1.1 Issues with digital twins

Digital twins are virtual entities linked to real-world entities. Twins consist of features and building blocks that are selected to serve the underlying use cases. [Aut2020] For at least a decade, digital twins have been used to monitor, simulate, and predict the behavior of different types of machines. However, the last few years have witnessed digital twins for almost anything, for example, cities, people, and organizations. The most recent trend is to start building a global network of digital twins [Aut2021a].

Use cases for digital twins are difficult to describe concisely due to their ability to absorb pretty much any other digital technology, which is why it is similarly difficult to describe concisely what tangible problems digital twins solve. Digital twins draw their relevance from the fact that "information is a replacement for wasted physical resources" [Gri2017]. One could say that all digital twins create their added value by optimizing real-world functions with information.

Despite digital twins being used for untangling almost any information-related problems tied to the real world, there is no single commonly accepted standard way to *describe* digital twins. Due to the lack of standardization, twins need to be assembled manually for every use case, most of them are not discoverable over the internet, and they rarely enable scalable collaboration with other systems. The necessity of manual labour and lack of interconnectedness significantly limits the scalability of digital twins.

Digital twins typically consist of highly specialised, usually proprietary software that solve a specific problem. Standardizing the internal workings of such purpose-built solutions has proven to be too great of a challenge. Simultaneously, use cases for digital twins are getting so complex that they cannot be solved with a single software solution. Instead, digital twins are being assembled from several software blocks that each serve their specific purposes, resembling the microservice and SOA (service-oriented architecture) architecture styles commonly used in the software industry [Ala2021]. To be able to connect the software blocks, they need to have standard interfaces.

In addition to twins being assembled from blocks that communicate with each other, some use cases will also benefit from twin-to-twin communications, leading to a *network of digital twins*. For example, digital twins of intersections could communicate directly with twins of road users to optimize the smoothness of traffic. The twins need to have machine-readable descriptions of themselves that indicate e.g. what types of road users they represent, what commands the intersection gives, and where they are located.

Networking of digital twins was identified as one of three major ongoing shifts for digital twins by Kaivo-Oja *et al.* [Kai2020]. Autiosalo reviewed academic publications for networks of digital twins, finding relatively little research on the topic and no proper methods for implementation even though the majority of the publications seemed to assume that a network of digital twins would appear in the future [Aut2021a]. Autiosalo then developed an initial approach for building networks of digital twins, recognizing the standardization of identifiers and semantics for digital twins as an important topic for future research [Aut2021a]. To be properly scalable, networks of digital twins should support semantic interoperability, which is the ability of computer systems to exchange data with each other while preserving the meaning of the data. Developers should also be able to organize twins into a network that mirrors the relations of the real-world counterparts of the twins.

The IoT-NGIN project aims to solve the lack of standardization of software interfaces and machine-readable descriptions with one general approach: the Semantic Twin, a structured description of the digital twin and its capabilities.

5.1.2 Need for Semantic Twins in IoT-NGIN

The IoT-NGIN project has a significant number of use cases and technological components. The Semantic Twin acts as an integrating element between these technological components by enabling semantic-level interoperability, and thus provides a common approach to IoT-application development. A common approach avoids the problem of producing one-off solutions, where a considerable amount of effort is used to produce a use case specific solution, that is hard to extend, maintain, or reuse. [WoT-doc] The open and technology agnostic nature of Semantic Twins also supports the distributed and federated approach utilised in the IoT-NGIN-project.

In the following sections, the use of Semantic Twins is examined by exploring two IoT-NGIN use cases in more detail. Detailed descriptions of the use cases are available in the IoT-NGIN deliverable 1.1 [D1.1].

5.1.2.1 UC#8: Digital powertrain and condition monitoring

In the context of this use case, the term powertrain is used to describe the equipment involved in transforming energy provided by a power source into useful work done by some machine. In industrial applications, such equipment typically include an AC motor and a variable speed drive responsible for its control [ABB]. A general description of the powertrain setups used in UC#8 is given in Figure 5.1. Such powertrain setups can be

implemented using different protocols and data models depending on the application domain, or due to differences in site specific setups. Thus, the creation of e.g. condition monitoring applications for such powertrains are typically manufacturer and site specific, resulting in siloed solutions.

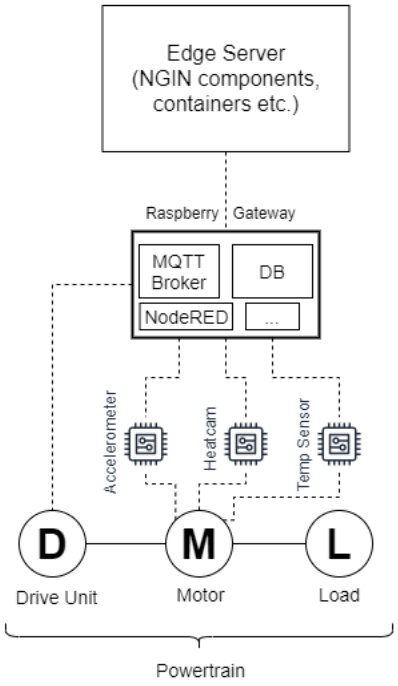

Figure 5.1 - UC#8 Powertrain setup.

For UC#8, the following scenario is considered. A company has two industrial sites, A & B, which both contain 3 powertrain setups each. The goal is to create a condition monitoring application that is common for all sites and may leverage data produced by all powertrains for analytics and machine learning.

5.1.2.2 UC#1: Traffic Flow Prediction & Parking prediction

The Jätkäsaari region in Helsinki is a bottleneck for traffic because the harbor operations in the peninsula bring high peaks of traffic to streets with several traffic lights. To address the issue, different types of sensors are being installed according to Figure 5.2. Traffic flows can then be optimised by leveraging the sensor data in predictive simulations that control traffic lights.

However, developing the simulations is laborious in itself, and connecting sensors to them can be overwhelming if the simulation expert is not also an expert in IoT sensors integration. Development of the predictions could be made more efficient by providing metadata about the sensors in an easily accessible and structured way, potentially supporting no-code integrations to the simulation software. Hence, the main goal of Semantic Twins in the Smart City Living Lab is helping developers make traffic and crowd-related simulation

models and other analysis applications more efficiently while enabling the necessary security and trust features.

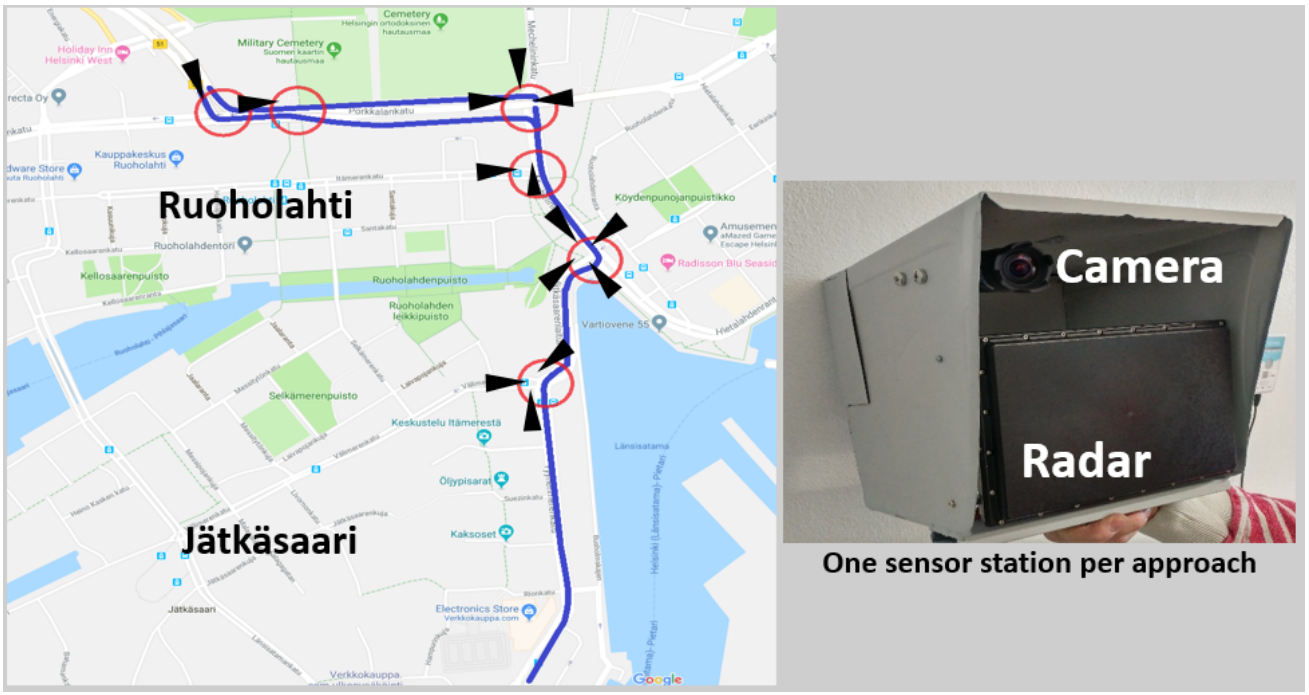

Figure 5.2 – Sensor installations for the Jätkäsaari Smart Junction. Image from: https://mobilitylab.hel.fi/materials/

5.1.3 Requirements for Semantic Twins

The IoT-NGIN project uses three sources of information to form requirements for the Semantic Twins solution:

1. The issues with digital twins described in section 5.1.1.
2. Previously identified requirements from literature. (E.g. the doctoral dissertations of Autiosalo [Aut2021a] and Ala-Laurinaho [Ala2021].)
3. The needs for the usage of Semantic Twins in IoT-NGIN Living Labs.

The requirements are summarised in Table 5.1 and detailed below.

REQ_TWIN_NF01: Supported real-world counterparts need to include anything that someone wants to make a digital twin for. This can mean both physical entities such as IoT devices and non-physical entities such as organizations, projects, and planned physical things.

REQ_TWIN_NF02: The solution must enable the creation of internet-wide networks of digital twins in distinction to twins that only need to work in a limited environment, such as inside a company intranet.

Table 5.1 – Requirements for the semantic twin solution.

ID	Requirement	Description
REQ_TWIN_NF01	General	Must support digital twins for all types of real-world entities.
REQ_TWIN_NF02	Global	Must support a global network of digital twins.
REQ_TWIN_NF03	Trustworthy	Must enable verification of data and metadata.
REQ_TWIN_NF04	Machine readable	The twins must follow a machine-readable structure to facilitate automated processing.
REQ_TWIN_NF05	Human readable	Semantic Twins must be human readable to the extent appropriate for each person's user group.
REQ_TWIN_NF06	Public	Must enable distributing selected parts of Semantic Twin instances publicly.
REQ_TWIN_NF07	Private	Must enable keeping selected parts of Semantic Twin instances private.
REQ_TWIN_NF08	Discoverable	Must be discoverable via appropriate methods.
REQ_TWIN_NF09	Developer friendly	Must support a developer-friendly workflow for creating and updating twins.
REQ_TWIN_NF10	Technology agnostic	Must be able to be built on-top of existing digital twin implementations.
REQ_TWIN_NF11	Decentralised	Must support decentralised implementations.
REQ_TWIN_NF12	Extendable	Must enable developers to introduce custom-defined semantics to provide enough development freedom.

REQ_TWIN_NF03: Users of Semantic Twins must be able to verify the source and integrity of the data and metadata of the corresponding digital twins.

REQ_TWIN_NF04: The twins must be described in a standardised machine-readable format and structure that enable automated processes, such as parsing and crawling.

REQ_TWIN_NF05: Developers of Semantic Twins should be able to read the twins in their raw standardised format to promote development efficiency and adoption. End users of twins should be able to access relevant information via specialised browser applications.

REQ_TWIN_NF06: The overall solution must enable twin owners to distribute whole twins or their parts publicly on the internet.

REQ_TWIN_NF07: It must be possible to keep some twins or their parts private, in distinction to offering them publicly on the internet. This can mean extending a public twin with private additions, or keeping the whole twin undiscoverable for outside users. The solution

should support sharing information with a group of users in addition to keeping it accessible only to the owner.

REQ_TWIN_NF08: Machines and human users should be able to discover Semantic Twins with methods appropriate to each use case, such as following the semantical structure of the real world.

REQ_TWIN_NF09: Semantic Twins must support digital twin builders in their development efforts. For example, an HTML equivalent should be introduced for digital twins.

REQ_TWIN_NF10: Semantic Twins must support all types of digital twins, e.g. by allowing extendable ontologies.

REQ_TWIN_NF11: The Semantic Twins must support decentralised implementations, such as the SSI technologies.

REQ_TWIN_NF12: The basic format to describe a digital twin may not be enough for some use cases. To tackle these use cases, developers must be able to introduce new semantic definitions, i.e. ontologies, to describe the various types of information related to digital twins and the things they represent and for building semantic connections between the twins. Extendability can be achieved by including existing ontologies, e.g. SAREF, or by creating new custom ontologies.

5.2 State-of-the-art of Semantic Twins

While the exact concept of Semantic Twins developed by the IoT-NGIN project appears to be novel, there are several existing solutions that can be leveraged during the development. The following subsections review the overall directions of the semantic interoperability of digital twins as well as existing solutions for twin description documents and other supporting tools and technologies.

5.2.1 Semantic interoperability of digital twins

In this report, semantic interoperability of digital twins refers to an approach where digital twins understand the basic structure of themselves and how they are related to each other. For example, a digital twin of a sensor knows what type of data it produces and can tell it to other twins so they can judge if they can utilize the data. I.e. the information about the interoperability of the twins is embedded in the twins.

The interoperability can be described in human-understandable terms using a Linked Data format. The descriptions of the interoperability are then structured using ontologies as described in Section 4. The Semantic Web is an example of this approach: it describes WWW resources with ontologies using Linked Data.

However, the approaches to create semantically interoperable digital twins are fragmented, and a digital twin developer does not have a clear answer to what format they should use. (as a contrasting example, web page developers have a clear answer to

what format they should be using to describe their web pages: HTML.) IoT-NGIN aims to empower digital twin developers by working towards a situation where the format of digital twin descriptions is either obvious or trivial.

5.2.2 Twin description document

Twin description document is a phrase used for the text-format file that actually describes a digital twin [Aut2021a]. The term is being established and this report uses it to refer to the concept rather than to any of its implementations. Several standards or specifications have been proposed and used for describing twins or things [Jac2020]. There is no consensus on which is the best, but all approaches seem to be at least compatible with Linked Data formats such as JSON-LD. In this section, four existing approaches utilising twin description documents are reviewed:

1. Web of Things Thing Description (WoT TD) [WoT-TD]
2. Asset Administration Shell (AAS) [AAS]
3. Digital Twins Definition Language (DTDL) [DTDL]
4. Aalto Digital Twin document (ADTD) [DTd2021]

Web of Things is a set of standards for IoT by the World Wide Web consortium. Web of Things that started forming in the late 2000's, and Thing Description is their later specification for describing IoT devices. WoT TD seems to be utilised mainly by consumer IoT devices rather than industrial users.

Asset Administration Shell (AAS) is a container format and an information model for describing information about an industrial machine, published by Plattform Industrie 4.0, which is a Germany-based organization coordinating the digital structural shift of German industry. Several large industrial companies have been involved in the development of the AAS approach.

Digital Twins Definition Language (DTDL) is a format developed and hosted by Microsoft for the use of the Azure Digital Twins platform [ADT]. It seems to be the only approach originally developed specifically for digital twins, but the tie to a single commercial provider may hinder developing it towards a vendor-neutral standard. However, thanks to the technological maturity of the platform, DTDL already has some working commercial solutions.

Aalto Digital Twin document is an initial approach for a developer-friendly digital twin document format [Ala2020]. An early version of the draft specification defines some exact terms for a digital twin document, but the whole approach should be merged with other specifications capable of describing digital twins. However, in particular the basic principles of Aalto's DT document approach, such as developer-friendliness, should be preserved in the merged format in order to promote the adoption of digital twin description documents.

A key problem with the proposed formats is that currently users have to choose between specifications that mostly try to solve the same tasks, but have some differences in details

and are not compatible with each other, thus limiting interoperability. IoT-NGIN's aim is to merge or translate between the specifications so that users can just start writing DT description documents and not think about the format. Also, the developer-friendliness of the formats likely needs to be enhanced as JSON-LD is tedious to write and does not allow comments. A good overall solution will allow the use of convenient editors that abstract the raw format of the document, but also enable developers to use their favourite text editor efficiently.

All of the existing methods to create twin documents support JSON-LD at least to some extent, so it seems promising that a common approach for implementing semantic interoperability of digital twins can be found soon. However, agreeing to use only JSON-LD-compatible formats is only one layer of standardization, and further layers need to be standardised too. For example, the global network of digital twins requires a digital twin information model that enables full semantic interoperability in twin-to-twin communications. The documents also need further supporting tools and technologies to flourish.

5.2.3 Supporting tools and technologies

The semantic network of digital twin documents requires several technical solutions, out of which three seem to be essential for achieving a functioning ecosystem:

1. server,
2. browser, and
3. identifier system.

Server implementations are needed to make digital twin description documents available for users across the internet. In this case, 'server' refers to any device capable of distributing the documents to clients. Each digital twin description document has some server implementations, such as Eclipse Thingweb node-wot [node-wot] for WoT TD, Eclipse BaSyx [BaSyx] for AAS, Azure Digital Twins [ADT] for DTDL, and Twinbase [Twinbase] for Aalto DT documents.

Browsers for digital twin description documents show the contents of the document in a use-case-specific and user-friendly manner. There seems to be no comprehensive implementations of twin browsers, but e.g. the network formed by DTDL twins can be visualised with Azure Digital Twins Explorer [ADT-expl] and AAS packages can be viewed with AASX Package Explorer [AASX-expl].

Finally, identifiers are needed to facilitate the communication of digital twins [El2018], but concrete implementations seem to be lacking. As a general trend, it seems that most simulation-focused twins are given identifiers that are unique only locally and do not enable creating a global network of digital twins, whereas data-focused twins get globally unique and discoverable identifiers more often. However, the approaches are fragmented and there is no consensus on the format of global identifiers for digital twins. An initial concept for a digital twin identifier registry was introduced by Autiosalo *et al.* [Aut2021b].

The self-sovereign identity technologies described in chapter 3 can be leveraged for sustainable solutions.

Also other kinds of tools are needed, for example editors that help writing the documents, such as Eclipse EdiTDor [EdiTDor]. Ontologies, such as SAREF, are essential for the scalability of the Semantic Twin solution. Finally, DLTs could be used for use cases that require enhanced trust in the immutability of the contents of the digital twin description documents.

5.3 The Semantic Twin solution

IoT-NGIN formalizes the concept of Semantic Twins into a solution that enables digital twin developers to build their applications more efficiently. The following subsections describe the overall solution, its basic use case, and an initial detailed description of a semantic twin implementation for the Smart Industry Living Lab Use Case #8. A more detailed description of the solution will follow in the upcoming deliverable D5.4.

5.3.1 Semantic Twin

Semantic Twin is the new name for the concept that was previously referred to as the "meta-level digital twin" in the IoT-NGIN proposal. A Semantic Twin consists of a *Twin ID* and a *twin description document* as shown in Figure 5.3.

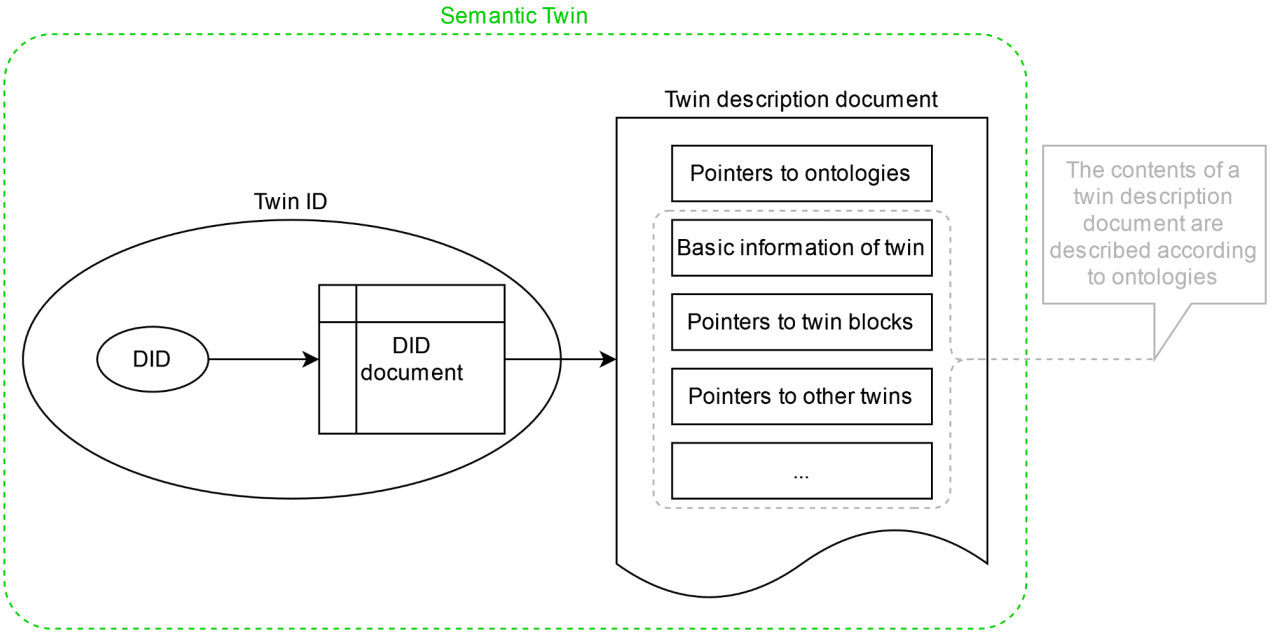

Figure 5.3 – The detailed contents of Semantic Twin.

A *Twin ID* represents the identifier and/or identity solutions of Semantic Twin. These solutions are implemented with SSI technologies described in Section 3. More detailed requirements

for the twin ID have been described by Autiosalo *et al.* [Aut2021b], who used the term "Twin ID registry" to refer to a system that manages Twin IDs.

The Twin ID then points to a *twin description document* (TDD) that provides a semantic description of a twin. As stated in Section 5.2.1, TDD is a Linked Data document that can be written e.g. in JSON-LD format. However, the format needs to be specified more exactly, having standardised ways to describe the contents so that the semantic understanding of the document can be achieved. The contents of a twin description document are described according to ontologies to ensure that they are machine readable. SAREF is the main ontology used in the IoT-NGIN project, but also other Linked Data compatible ontologies can be used when appropriate, and new ontologies can be created if a suitable ontology does not yet exist.

The IoT-NGIN proposal states that the developed Semantic Twin solution will be DLT-enabled. The Semantic Twin uses SSI technologies that can either be built with DLTs or allow the use of DLTs as an additional feature. However, it should be noted that DLTs excel in providing trustworthiness through immutability, but the higher trustworthiness comes with a higher cost. IoT-NGIN explores when the added trustworthiness provided by the DTLs is required and only deploys them accordingly to minimise unnecessary costs. Also, interledger will be used to achieve almost the same level of trust with significantly reduced costs by combining highly trustworthy ledgers with less expensive ones.

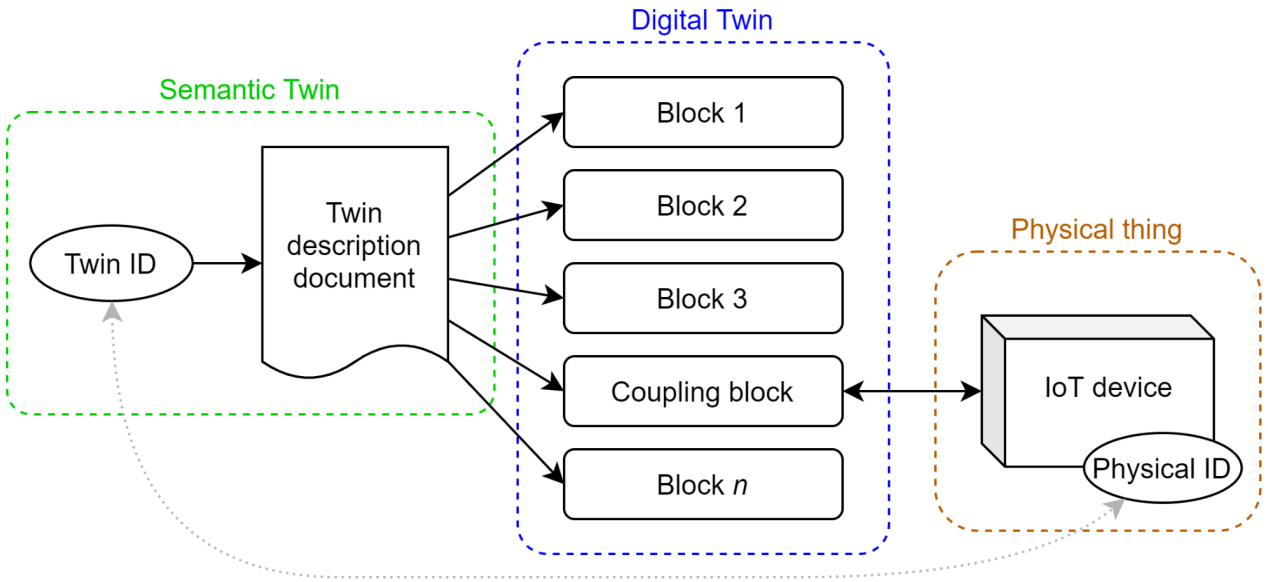

Figure 5.4 – Overview of a twin-thing system. The system includes a semantic twin, a digital twin, and a physical thing. The work in IoT-NGIN is focused on developing the details of the semantic twin concept and its connections.

Figure 5.4 illustrates how the Semantic Twin is connected to other concepts. A semantic twin is connected to a digital twin via a twin description document. The twin description

document includes links to digital twin building blocks that provide the actual services of the twin, such as a simulation model, data storage, and a connection to the corresponding physical thing. The physical thing, such as an IoT device, can contain a Physical ID that is linked to the Twin ID. The Physical ID can be e.g. a QR code, a microchip capable of writing cryptographic signatures, or both.

5.3.2 Basic discovery flow for Semantic Twins

This section describes the basic discovery flow for semantic twins. It involves a mobile application that scans a QR code with a GS1 Digital Link to access a twin application. Figure 5.5 omits some intermediary steps that need to be defined and validated through experimentation.

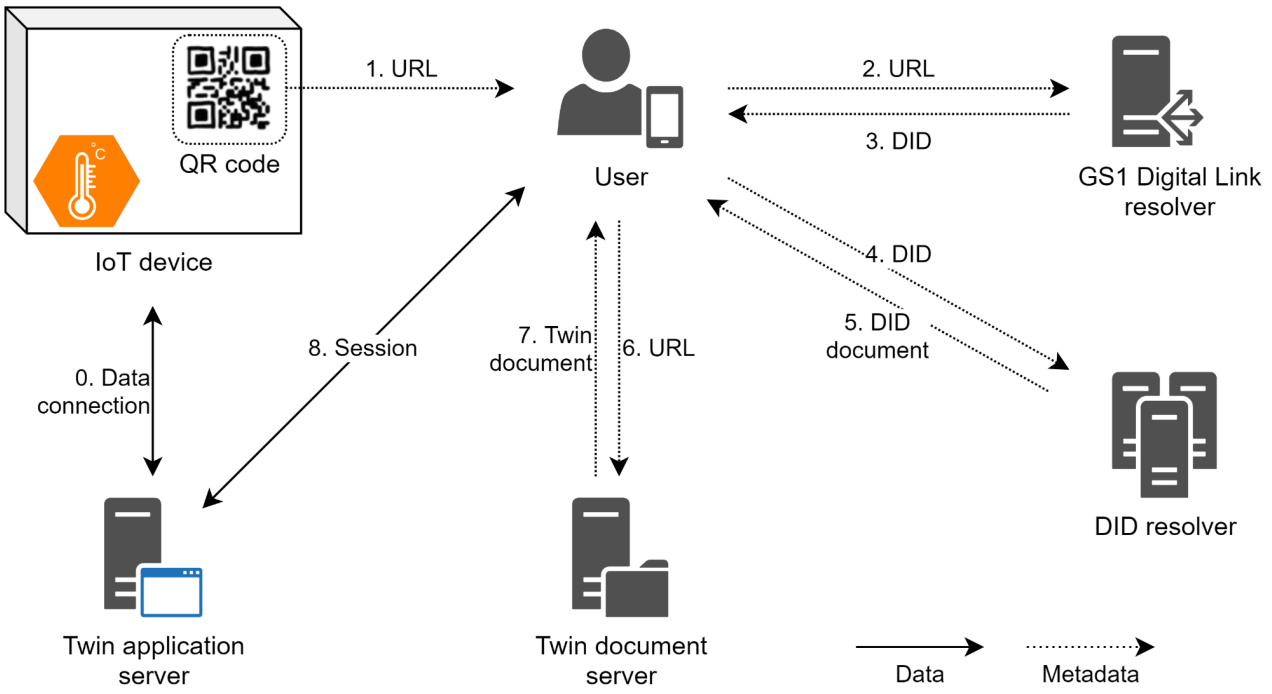

Figure 5.5 – Basic discovery flow for a semantic twin using a QR code.

The steps 0-8 shown in the figure are as follows:

0. Data connection between an IoT device and a twin application is established as a preparatory step by the solution provider.
1. A mobile application is used to scan a QR code attached to the IoT device.
2. The mobile application requests a GS1 Digital Link resolver to resolve the URL.
3. GS1 Digital Link Resolver sends a DID back to the application
4. The application uses a DID resolver according to the used DID method.

5. DID resolver provides a DID document to the application.

6. The application requests the twin document using a URL specified in the DID document.

7. The application receives the twin document. The application can validate the integrity of the document according to the contents of the DID document.

8. The application establishes a session with the twin application. The session initialization procedure may include features enabled by SSI technologies, such as ensuring the user's right to access the twin application with verifiable credentials.

In addition to this basic flow, IoT-NGIN develops Semantic Twins by experimenting with the technology through PoC implementations in Living Labs, aiming to reach a technology readiness level (5) that enables the use of semantic twins in production environments of the Living Labs, and explores using semantic twins as tools for the other technical work packages.

5.3.3 Initial solution for UC#8

Twin description documents are created for the powertrains and sensors of the use case. The documents are used in application development, abstracting the underlying protocols used for a specific powertrain setup. The resulting twin description view of the use case is depicted in Figure 5.6.

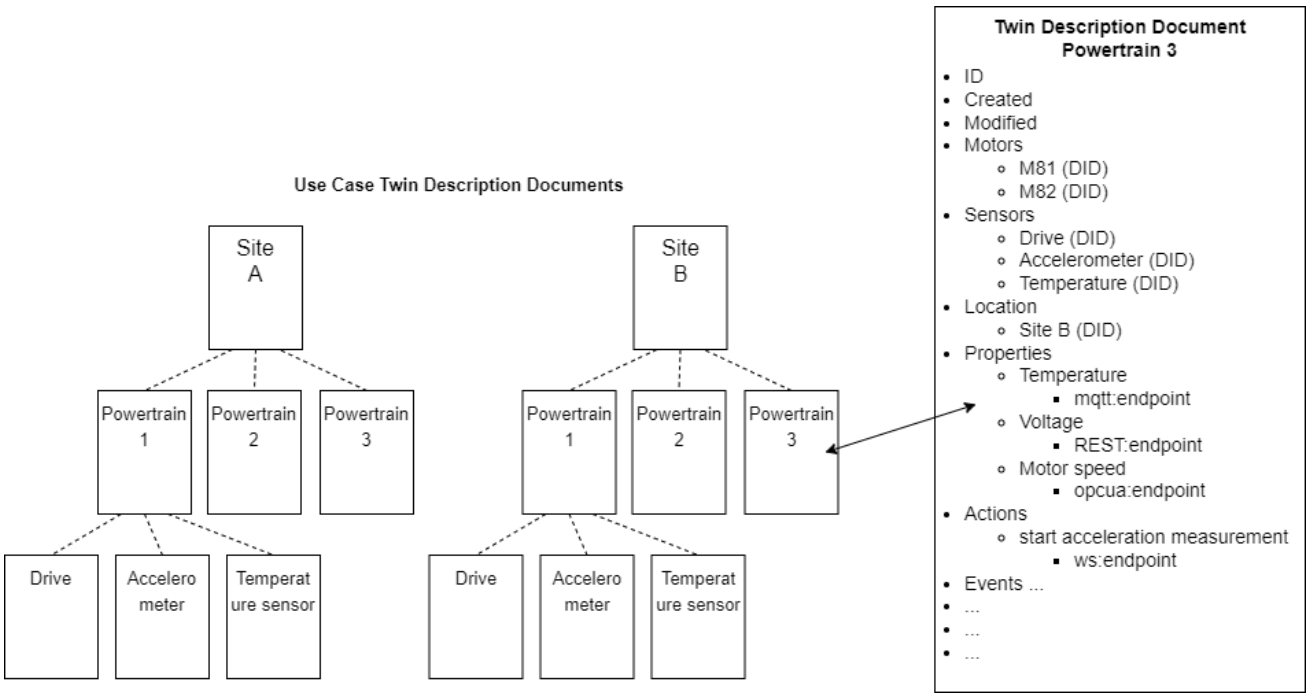

Figure 5.6 - UC#8 twin description view.

The twin descriptions follow a general structure and utilise ontologies, resulting in machine readable documents that can be parsed and crawled for information. A twin description

example utilising the W3C WoT model is given in Figure 5.7. The example shows how a UC#8 powertrain websocket endpoint generating json data could be described using the WoT Thing Description (TD) specification.

```
"drive_signals":{
    "data":{
        "type": "object",
        "properties":{
            "payload": {
                "type": "array",
                "items":{
                    "type": "object",
                    "properties":{
                        "objectId":{
                            "type": "string"
                        },
                        "model":{
                            "type": "string"
                        },
                        "variable":{
                            "type": "string"
                        },
                        "timestamp":{
                            "type": "string",
                            "format": "date-time"
                        },
                        "value":{
                            "type": "number"
                        },
                        "unit":{
                            "type": "string"
                        }
                    }
                }
            }
        }
    },
    "forms": [
      {
        "href": "ws://172.21.0.11:1880/ws/drive_panel",
        "contentType": "application/json",
        "op": "observeproperty"
      }
    ]
}
```

Figure 5.7 - UC#8 WoT TD Websocket example.

Similarly, any interface can be described using the basic structure defined by the WoT model: "WoT introduces a simple interaction abstraction based on properties, events, and actions. Any IoT network interface can be described in terms of this abstraction. By using this abstraction, applications have a common anchor for retrieving an IoT service's metadata as well as a way to understand what and how the data and an IoT services'

functions can be accessed. [Wot-doc]" Applications process the information from the TD document to create working instances, e.g. client or server instances, based on the data and endpoint descriptions given in the document. Frameworks for processing these types of twin description documents exist for the AAS, WoT and DTDL specifications. For example, Thingweb [ThingWeb] implements the WoT model, providing a node.js based toolkit for processing TDs and various protocols via additional protocol binding packages.

Thingweb also provides implementations for hosting and browsing the TD documents. In the NGIN-project, DLTs will also be explored as a means to store and serve parts of the powertrain meta-level data e.g. DID documents. However, storing an entire twin description document using DLTs is not ideal due to the privacy issues outlined in section 3.3.1.

The benefit of using twin descriptions for UC#8 is the common structure provided by the twin description document: instead of handling the underlying protocols of each powertrain individually, the application programmer works with the meta-level descriptions of these interfaces. The twin description document describes the available data interfaces in a structured way, possibly including semantic information as well. For example, the drive unit gathers measurements of the same physical quantities from all powertrains, e.g. motor speed, torque, and current. However, the underlying protocols and data structures used to collect this data may vary. The twin description allows the application programmer to handle all powertrains in a consistent structured manner. For example, the addition of new powertrains, or changes to existing powertrain implementations, can be handled with less effort, as the overall structure of the twin description remains the same, even when the underlying implementation changes. This becomes more apparent in larger and more complex use cases, which may consist of multiple parties and hundreds of devices.

6 Conclusions

This document observes the problems of privacy preserving and trust improvement in the domain of IoT systems, and discusses the technical solutions including multi-ledger transactions, Self-Sovereign Identities, ontologies and Semantic Twins that can be utilised to tackle the problems.

Based upon various needs in use cases within the IoT-NGIN project, requirements were analysed and summarised, in order to specify the targeted features and property of each technical solution respectively. The state-of-the-art of each technology were then reviewed to identify the best approaches for the solutions to be developed in IoT-NGIN.

Finally, technical solutions for each area were then outlined. With them, the analysed technologies can successfully be deployed to address the identified problems. The detailed solutions will be described in the upcoming Deliverable 5.4.

7 References

[AAS] Asset Administration Shell Specifications, Plattform Industrie 4.0. 23/11/2020 https://www.plattform-i40.de/IP/Redaktion/EN/Standardartikel/specification-administrationshell.html

[AASX-expl] https://github.com/admin-shell-io/aasx-package-explorer

[Abb2019] A. Abbas, G. Privat et al., "ETSI GS CIM 006 V1.1.1 Context Information Management (CIM); Information Model (MOD0)". Group Specification CIM 006, Jul. 2019 url: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/006/01.01.01_60/gs_CIM006v010101p.pdf

[ABBdri] ABB drives. Technical guide No. 4. Guide to variable speed drives. https://library.e.abb.com/public/d3c711ec2acddb18c125788f002cf5da/ABB_Technical_guide_No_4_REVC.pdf

[Abe2019] E. Abebe, D. Behl, C. Govindarajan, Y. Hu, D. Karunamoorthy, P. Novotny, V. Pandit, V. Ramakrishna, and C. Vecchiola, "Enabling enterprise blockchain interoperability with trusted data transfer (industry track)," in Proceedings of the 20th International Middleware Conference Industrial Track, 2019, pp. 29–35.

[ADT] https://docs.microsoft.com/en-us/azure/digital-twins/concepts-models

[ADT-expl] https://docs.microsoft.com/en-us/samples/azure-samples/digital-twins-explorer/digital-twins-explorer/

[Aio2021] "AION whitepaper." [Online]. Available: https://whitepaper.io/document/31/aion-whitepaper, accessed on 03/12/2021.

[Al2017] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, "Chainspace: A sharded smart contracts platform," arXiv preprint arXiv:1708.03778, 2017.

[Ala2020] R. Ala-Laurinaho, J. Autiosalo, A. Nikander, J. Mattila and K. Tammi, "Data Link for the Creation of Digital Twins," in *IEEE Access*, vol. 8, pp. 228675-228684, 2020. https://doi.org/10.1109/ACCESS.2020.3045856.

[Ala2021] R. Ala-Laurinaho, "API-based Digital Twins - Architecture for Building Modular Digital Twins Following Microservices Architectural Style," Doctoral dissertation, Aalto University, 2021. http://urn.fi/URN:ISBN:978-952-64-0594-0

[All2016] Allen, Christopher. "The Path to Self-Sovereign Identity," 2016. http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html.

[And2018] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevichet al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in Proceedings of the thirteenth EuroSys conference, 2018,pp. 1–15.

[And2016] Andrieu, Joe. "A Technology-Free Definition of Self-Sovereign Identity." Rebooting the Web of Trust III, 2016. https://github.com/WebOfTrustInfo/rwot3-sf/blob/master/topics-and-advance-readings/a-technology-free-definition-of-self-sovereign-identity.pdf.

[Aries-0005] Hardman, Daniel. "Aries RFC 0005: DID Communication." Hyperledger, 2019. https://github.com/hyperledger/aries-rfcs/blob/master/concepts/0005-didcomm/README.md.

[Aries-0050] https://github.com/hyperledger/aries-rfcs/blob/master/concepts/0050-wallets/README.md

[Ark2021] "ARK Releases All New Whitepaper v2." [Online]. Available: https://ark.io/blog/ark-releases-all-new-whitepaper-v2-1915ad56dd33, accessed on 03/12/2021

[Arm2017] Armin Haller et al. Semantic Sensor Network Ontology. Technical Specification OGC 16-079. W3C & OGC, Oct. 17, 2017. url: https://www.w3.org/TR/vocab-ssn/.

[Aut2020] [1]J. Autiosalo, J. Vepsäläinen, R. Viitala, and K. Tammi, "A Feature-Based Framework for Structuring Industrial Digital Twins," IEEE Access, vol. 8, pp. 1193–1208, 2020, doi: https://doi.org/10.1109/ACCESS.2019.2950507

[Aut2021a] J. Autiosalo, "Discovering the Digital Twin Web - From singular applications to a scalable network," Doctoral dissertation, Aalto University, 2021. http://urn.fi/URN:ISBN:978-952-64-0621-3

[Aut2021b] J. Autiosalo, J. Siegel and K. Tammi, "Twinbase: Open-Source Server Software for the Digital Twin Web," in *IEEE Access*, vol. 9, pp. 140779-140798, 2021, https://doi.org/10.1109/ACCESS.2021.3119487

[Ave2019] Avellaneda, Oscar, Alan Bachmann, Abbie Barbir, Joni Brenan, Pamela Dingle, Kim Hamilton Duffy, Eve Maler, Drummond Reed, and Manu Sporny.

"Decentralized Identity: Where Did It Come From and Where Is It Going?" IEEE Communications Standards Magazine 3, no. 4 (2019): 10–13.

[Bac2014] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Tim´on, and P. Wuille, "Enabling blockchain innovations with pegged sidechains," URL: http://www.opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains, vol. 72, 2014.

[Baj2017] Garvita Bajaj et al. "A study of existing Ontologies in the IoTdomain". In: *arXiv:1707.00112 [cs]* (July 1, 2017). arXiv: 1707.

00112. url: http://arxiv.org/abs/1707.00112 (visited on 06/25/2021).

[BaSyx] https://www.eclipse.org/basyx/

[Bel2020] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," arXiv preprint arXiv:2005.14282, 2020.

[Blo2021] "The Blocknet Design Specification." [Online]. Available: https://blocknet.co/whitepaper/Blocknet\Whitepaper.pdf, accessed on 03/12/2021.

[Bro2016] R. G. Brown, J. Carlyle, I. Grigg, and M. Hearn, "Corda: an introduc-tion," R3 CEV, August, vol. 1, p. 15, 2016.

[Bru2020] Brunner, Clemens, Ulrich Gallersdörfer, Fabian Knirsch, Dominik Engel, and Florian Matthes. "DID and VC: Untangling Decentralized Identifiers and Verifiable Credentials for the Web of Trust." In 2020 the 3rd International Conference on Blockchain Technology and Applications, pp. 61-66. 2020.

[Btr2021] "BTC Relay," http://btcrelay.org/, accessed on 03/12/2021.

[Bul2013] A. Buldas, A. Kroonmaa, and R. Laanoja, "Keyless signatures' infrastructure: How to build global distributed hash-trees," in Nordic Conference on Secure IT Systems. Springer, 2013, pp. 313–320.

[Bur2018] C. Burchert, C. Decker, and R. Wattenhofer, "Scalable funding of bitcoin micropayment channel networks," Royal Society open science, vol. 5, no. 8, p. 180089, 2018.

[But2016] V. Buterin, "Chain interoperability," R3 Research Paper, 2016

[Cam2005] Cameron, Kim. "The Laws of Identity." Microsoft Corporation, 2005. https://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf.

[Cam2021] "SOFIE - Context-Aware Mobile Gaming Pilot," https://media.voog.com/0000/0042/0957/files/sofie-onepager-gaming-noScreens.pdf, accessed on 03/12/2021.

[Cha2019] Chadwick, David, et al. "Verifiable Credentials Implementation Guidelines 1.0." W3C Working Group Note, Sep (2019).

[Cur2019] Curren, Sam. "Aries RFC 0025: DIDComm Transports." Hyperledger, 2019. https://github.com/hyperledger/aries-rfcs/blob/master/features/0025-didcomm-transports/README.md.

[D1.1] IoT-NGIN D1.1 - Definition analysis of use cases and GDPR Compliance https://iot-ngin.eu/index.php/deliverable/

[Dec2015] C. Decker and R. Wattenhofer, "A fast and scalable payment network with bitcoin duplex micropayment channels," in Symposium on Self-Stabilizing Systems. Springer, 2015, pp. 3–18.

[DHR2015] Laura Daniele, Frank den Hartog, and Jasper Roes. "Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology". In: Formal Ontologies Meet Industry. Ed. by Roberta Cuel and Robert Young. Lecture Notes in Business Information Processing. Cham: Springer International Publishing, 2015, pp. 100–112. isbn: 978-3-319-21545-7. doi: 10.1007/978-3319-21545-7_9.

[DID] Reed, Drummond, Manu Sporny, Dave Longley, Christopher Allen, Ryan Grant, and Markus Sabadello. "Decentralized Identifiers (DIDs) v1.0." W3C, 2020. https://www.w3.org/TR/did-core/.

[did-dns] Alexander Mayrhofer, Dimitrij Klesev, and Markus Sabadello, "The Decentralized Identifier (DID) in the DNS, draft-mayrhofer-did-dns-05," IETF draft, June 2021, https://datatracker.ietf.org/doc/draft-mayrhofer-did-dns/05/.

[DID-Self] https://github.com/mmlab-aueb/did-self

[did-self] Nikos Fotiou, "did:self method specification," 2021, https://github.com/mmlab-aueb/did-self.

[did-web] "did:web Method Specification", 2021, https://w3c-ccg.github.io/did-method-web/.

[Dil2016] J. Dilley, A. Poelstra, J. Wilkins, M. Piekarska, B. Gorlick, and M. Friedenbach, "Strong federations: An interoperable blockchain solution to centralized third-party risks," arXiv preprint arXiv:1612.05491, 2016.

[DTDL] Digital Twins Definition Language https://github.com/Azure/opendigitaltwins-dtdl

[EdiTDor] https://github.com/eclipse/editdor/

[eduroam] https://eduroam.org/

[El2018] A. El Saddik, "Digital Twins: The Convergence of Multimedia Technologies," in *IEEE MultiMedia*, vol. 25, no. 2, pp. 87-92, Apr.-Jun. 2018. https://doi.org/10.1109/MMUL.2018.023121167

[Eng2016] S. M. English, F. Orlandi, and S. Auer, "Disintermediation of inter-blockchain transactions," arXiv preprint arXiv:1609.02598, 2016.

[ERC721] "ERC-721 Non-Fungible Token Standard," https://eips.ethereum.org/EIPS/eip-721, accessed on 03/12/2021.

[Fer2019] Ferdous, Md Sadek, Farida Chowdhury, and Madini O Alassafi. "In Search of Self-Sovereign Identity Leveraging Blockchain Technology." IEEE Access 7 (2019): 103059–103079.

[Gal2009] Avigdor Gal. "Ontology Engineering". In: Encyclopedia of Database Systems. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 1972–1973. isbn: 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_1315. url: https://doi.org/10.1007/978-0-387-39940-9_1315.

[Gan2017] M. Ganzha et al. "Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective". In: undefined (2017). url: /paper / Semantic interoperability-in-the-Internet-of-An-the-Ganzha-

[GDPR] Regulation (Eu) 2016/679 of the European Parliament and of the Council–on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016, https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[github-did] Orie Steele, "GitHub DID," 2021, https://github.com/decentralized-identity/github-did.

[GOS2009] Nicola Guarino, Daniel Oberle, and Steffen Staab. "What Is an Ontology?" In: Handbook on Ontologies. Ed. by Steffen Staab and Rudi Studer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1– 17. isbn: 978-3-540-70999-2 978-3-540-92673-3. doi: 10.1007/978-3-540-92673-3_0. url: http://link.springer.com/10.1007/ 978-3-540-92673-3_0 (visited on 05/27/2021).

[Gra2017] Grassi, P., Michael E. Garcia, and James L. Fenton. "Digital identity guidelines." NIST Special Publication 800 (2017): 63-3.

[Gri2017] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," In: Kahlen FJ., Flumerfelt S., Alves A. (eds) Transdisciplinary Perspectives on Complex Systems. (2017) Springer, Cham. https://doi.org/10.1007/978-3-319-38756-7_4

[Hag2020] Maliheh Haghgoo et al. "SARGON – Smart energy domain ontology". In: IET Smart Cities 2.4 (Dec. 2020), pp. 191–198. issn: 26317680, 2631-7680. doi: 10.1049/iet-smc.2020.0049. url: https:

//onlinelibrary.wiley.com/doi/10.1049/iet-smc.2020.0049 (visited on 05/27/2021).

[Has2019] M. U. Hassan, M. H. Rehmani, and J. Chen, "Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions," Future Generation Computer Systems, vol. 97, pp. 512–529, 2019.

[HDR2015] Frank den Hartog, Laura Daniele, and Jasper Roes. "Toward semantic interoperability of energy using and producing appliances in residential environments". In: 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). Las Vegas, NV, USA: IEEE, Jan. 2015, pp. 170–175. isbn: 978-1-4799-6390-4. doi: 10.1109/CCNC.2015.7157972. url: https://ieeexplore.ieee.org/document/7157972/ (visited on 06/08/2021).

[Her2018] M. Herlihy, "Atomic cross-chain swaps," in Proceedings of the 2018 ACM symposium on principles of distributed computing, 2018, pp. 245–254.

[Hop2016] A. Hope-Bailie and S. Thomas, "Interledger: Creating a standard for payments," in Proceedings of the 25th International Conference Companion on World Wide Web, 2016, pp. 281–282.

[Htl2021] Hash Time Locked Contracts - Bitcoin Wiki," https://en.bitcoin.it/wiki/HashTimeLockedContracts, accessed on 03/12/2021.

[Hug2019] Hughes, A, M Sporny, and D Reed. "A Primer for Decentralized Identifiers." Draft Community Group Report, W3C, 2019.

[Hyp2021] "Hyperledger Cactus: Hyperledger Cactus is a new approach to the blockchain interoperability problem," https://github.com/hyperledger/cactus, (Accessed on 03/12/2021).

[idemix] J. Camenisch and E. V. Herreweghen, "Design and implementa- tion of the idemix anonymous credential system," in Proceedings of the 9th ACM Conference on Computer and Communications Security CCS '02, pp. 21–30, ACM, New York, NY, USA, November 2002.

[IDProID] Glazer, Ian. "Identifiers and Usernames." IDPro Body of Knowledge 1, no. 1 (2020).

[IDPro2020] Cameron, Andrew, and O. Grewe. "An Overview of the Digital Identity Lifecycle." IDPro Body of Knowledge 1, no. 3 (2020).

[Ilp2021] Interledger protocol v4." [Online]. Available: https://interledger.org/rfcs/0027-interledger-protocol-4, accessed on 03/12/2021.

[iot-idm1] "Integration of anonymous credential systems in IoT constrained environments" -> https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8242656

[iot-idm2] "Holistic Privacy-Preserving Identity Management System for the Internet of Things" -> https://www.hindawi.com/journals/misy/2017/6384186/

[IoT-SSI1] "Improving the Privacy of Iot with Decentralised Identifiers (Dids)." https://www.hindawi.com/journals/jcnc/2019/8706760/

[IoT-SSI2] "Enabling Decentralised Identifiers and Verifiable Credentials for Constrained Internet-of-Things Devices Using OAuth-Based Delegation." https://www.ndss-symposium.org/wp-content/uploads/diss2019_05_Lagutin_paper.pdf

[ISO2020] "ISO/IEC 24760-1:2019 IT Security and Privacy — A Framework for Identity Management." ISO/IEC, 2019. https://www.iso.org/standard/77582.html.

[Jac2020] M. Jacoby, and T. Usländer, "Digital Twin and Internet of Things—Current Standards Landscape," *Applied Sciences* 2020, *10*, 6519. https://doi.org/10.3390/app10186519

[Kae2020] Sebastian Kaebisch et al. Web of Things (WoT) Thing Description. Technical Specification. World Wide Web Consortium (W3C), Apr. 9, 2020. url: https://www.w3.org/TR/wot-thingdescription/ (visited on 06/25/2021).

[Kai2020] J. Kaivo-oja, O. Kuusi, M. S. Knudsen, and I. T. Lauraéus, "Digital twin: current shifts and their future implications in the conditions of technological disruption," International Journal of Web Engineering and Technology, vol. 15, no. 2, pp. 170–188, Jan. 2020, https://doi.org/10.1504/IJWET.2020.109730

[Kal2020] Kalliola, Markus, Katri Korhonen, Juhani Luoma-Kyyny, Pirkka Frosti, Paul Knowles, Antti Kettunen, Robert Mitwicki, et al. "Elements of Fair and Functioning Data Economy: Identity, Consent and Logging." European Committee for Standardization, 2020. ftp://ftp.cencenelec.eu/CWA/CEN/IHAN/CWA17525.pdf.

[Kan2018] L. Kan, Y. Wei, A. H. Muhammad, W. Siyuan, G. Linchao, and H. Kai, "A multiple blockchains architecture on inter-blockchain communica-tion," in 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, 2018, pp. 139–145.

[Kho2017] D. Khovratovich and J. Law, "Sovrin: digital identities in the blockchain era,"Github Commit by jasonalaw October, vol. 17, 2017.

[Kim2021] Kim, Bong Gon, Young-Seob Cho, Seok-Hyun Kim, Hyoungshick Kim, and Simon S. Woo. "A Security Analysis of Blockchain-Based Did Services." IEEE Access 9 (2021): 22894-22913.

[Kok2018] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in 2018 IEEE Symposium on Security and Privacy (SP).IEEE, 2018, pp. 583–598.

[Kwo2019] J. Kwon and E. Buchman, "Cosmos whitepaper: A network of distributed ledgers, 2019," URL: https://cosmos.network/cosmos-whitepaper.pdf, accessed on 03/12/2021.

[Laa2021] Laatikainen, Gabriella, Taija Kolehmainen, Mengcheng Li, Markus Hautala, Antti Kettunen, and Pekka Abrahamsson. "Towards a trustful digital world: exploring self-sovereign identity ecosystems." arXiv preprint arXiv:2105.15131 (2021).

[Lau2020] Laura Daniele et al. *ETSI TS 103 264 V3.1.1 - SmartM2M; Smart Applications; Reference Ontology and oneM2M Mapping*. Technical Specification TS 103 264. ETSI, Feb. 11, 2020. url: https:// saref.etsi.org/core/v3.1.1/.

[Lee2011] C. Lee, "Litecoin," 2011

[Ler2016] S. D. Lerner, "Drivechains, sidechains and hybrid 2-way peg designs," 2016.

[Li2019] D. Li, J. Liu, Z. Tang, Q. Wu, and Z. Guan, "Agentchain: A decen-tralized cross-chain exchange system," in2019 18th IEEE InternationalConference On Trust, Security And Privacy In Computing And Com-munications/13th IEEE International Conference On Big Data ScienceAnd Engineering (TrustCom/BigDataSE). IEEE, 2019, pp. 491–498.

[Liu2019] Z. Liu, Y. Xiang, J. Shi, P. Gao, H. Wang, X. Xiao, B. Wen, and Y.-C. Hu, "Hyperservice: Interoperability and programmability across heterogeneous blockchains," in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 549–566.

[Muh2018] Mühle, Alexander, Andreas Grüner, Tatiana Gayvoronskaya, and Christoph Meinel. "A Survey on Essential Components of a Self-Sovereign Identity." Computer Science Review 30 (2018): 80–86.

[Nai2020] Naik, Nitin, and Paul Jenkins. "Self-Sovereign Identity Specifications: Govern your identity through your digital wallet using blockchain technology." In 2020 8th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pp. 90-95. IEEE, 2020.

[Nak2008] S. Nakamotoet al., "Bitcoin: a peer-to-peer electronic cash system(2008)," 2008.

[node-wot] https://github.com/eclipse/thingweb.node-wot

[NP01] Ian Niles and A. Pease. "Towards a standard upper ontology". In: undefined (2001). url: https://dl.acm.org/doi/10.1145/ 505168.505170 (visited on 05/28/2021).

[Odo2019] O'Donnell, Darrell. "The Current and Future State of Digital Wallets." Continuum Loop Inc, 2019, 83.

[One2019] oneM2M Partners Type 1. oneM2M TS-0012-V3.7.3 - Base Ontology. Technical Specification TS 103 264. oneM2M, Feb. 28, 2019. url: https://www.onem2m.org/images/pdf/TS-0012-Base_ Ontology-V3_7_3.pdf.

[Pee2021] https://identity.foundation/peer-did-method-spec/

[POA] "POA Network," https://www.poa.network/, accessed on 03/12/2021.

[Poo2016] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016.

[Poo2017] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," White paper, pp. 1–47, 2017.

[Ruf2018] https://medium.com/evernym/the-three-models-of-digital-identity-relationships-ca0727cb5186

[Sab2021] Sabadello, Markus, and Dmitri Zagidulin. "Decentralized Identifier Resolution (DID Resolution)." W3C, 2021. https://w3c-ccg.github.io/did-resolution/.

[Sch2014] D. Schwartz, N. Youngs, A. Brittoet al., "The ripple protocol consensus algorithm," Ripple Labs Inc White Paper, vol. 5, no. 8, p. 151, 2014.

[Sg-v2] https://sovrin.org/wp-content/uploads/Sovrin-Glossary-V2.pdf

[Sir2019] V. A. Siris, P. Nikander, S. Voulgaris, N. Fotiou, D. Lagutin, and G. C.Polyzos, "Interledger approaches," IEEE Access, vol. 7, pp. 89 948–89 966, 2019.

[social-login] https://auth0.com/learn/social-login/

[SOF2021] SOFIE project: Secure Open Federation for Internet Everywhere,"https://www.sofie-iot.eu/, (Accessed on 03/12/2021).

[Sor2020] Sorokin, Leo. "A Peek into the Future of Decentralized Identity." IDPro Body of Knowledge 1, no. 3 (2020).

[sovrin] Sovrin Foundation, Identity For All, Sovrin Foundation, Northampton, MA, USA, 2021, https://sovrin.org/.

[Spi2021] "SOFIE project - Interledger," https://github.com/SOFIE-project/Interledger, accessed on 03/12/2021

[Spo2019] Sporny, Manu, Dave Longley, and David Chadwick. "Verifiable Credentials Data Model 1.0." W3C, 2019. https://www.w3.org/TR/vc-data-model/.

[ThingWeb] https://www.thingweb.io/

[Tob2018] "Sovrin: What Goes on the Ledger?" -> https://www.evernym.com/wp-content/uploads/2018/10/What-Goes-On-The-Ledger.pdf

[ToIP] Davie, Matthew, Dan Gisolfi, Daniel Hardman, John Jordan, Darrell O'Donnell, and Drummond Reed. "The Trust over IP Stack." IEEE Communications Standards Magazine 3, no. 4 (2019): 46–51.

[TolPF] "Introducing the Trust Over IP Foundation." Trust Over IP Foundation,2021. https://trustoverip.org/wp-content/uploads/2020/05/toip_introduction_050520.pdf.

[Twinbase] https://github.com/twinbase/twinbase

[uport] Uport, "Open identity system for the decentralized web," https://www.uport.me/.

[Ver2018] G. Verdian, P. Tasca, C. Paterson, and G. Mondelli, "Quant overledger whitepaper," 2018.

[Wag2018] Wagner, K, B Némethi, E Renieris, P Lang, E Brunet, and E Holst. "Self-Sovereign Identity: A Position Paper on Blockchain Enabled Identity and the Road Ahead." Identity Working Group of the German Blockchain Association (Https://Jolocom. Io/Wp-Content/Uploads/2018/10/Self-Sovereign-Identity-_-Blockchain-Bundesverband-2018. Pdf, 2018.

[Wan2021] Wanchain - Decentralized Finance Interoperability," https://www.wanchain.org/, accessed on 03/12/2021.

[WCW2021] "WANCHAIN Building Super Financial Markets for the New Digital Economy." [Online]. Available: https://wanchain.org/files/Wanchain-Whitepaper-EN-version.pdf, accessed on 03/12/2021.

[Wei2021] Weingaertner, Tim, and Oskar Camenzind. "Identity of Things: Applying concepts from Self Sovereign Identity to IoT devices." The Journal of The British Blockchain Association (2021): 21244.

[Woo2014] G. Woodet al., "Ethereum: A secure decentralised generalised trans-action ledger,"Ethereum project yellow paper, vol. 151, no. 2014, pp.1–32, 2014.

[Woo2016] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," White Paper, 2016.

[WoT-doc] https://www.w3.org/WoT/documentation/

[WoT-TD] Web of Things (WoT) Thing Description. W3C Recommendation 9 April 2020 https://www.w3.org/TR/wot-thing-description/

[Wu2021] Wu, L., Kortesniemi, Y., Lagutin, D., & Pahlevan, M. (2021, September). The Flexible Interledger Bridge Design. In 2021 3rd Conference on Blockchain

Research & Applications for Innovative Networks and Services (BRAINS) (pp. 69-72). IEEE.

[Zam2018] A. Zamyatin, D. Harz, J. Lind, P. Panayiotou, A. Gervais, and W. J. Knottenbelt, "Xclaim: A framework for blockchain interoperability," in 40th IEEE Symposium on Security and Privacy, no. July 2018, 2018.

[Zam2019] A. Zamyatin, M. Al-Bassam, D. Zindros, E. Kokoris-Kogias, P. Moreno-Sanchez, A. Kiayias, and W. J. Knottenbelt, "SoK: communication across distributed ledgers." 2019.