IHAN BLUEPRINT 2.5

22 January 2020

0 About IHAN Blueprint versions

0.1 Versions

The first versions of IHAN Blueprint were created within the Technical Workstream as part of Sitra's IHAN Project during summer and autumn 2018. The first official version (2.0) was released closer to the year end and was written by Antti Larsio, Juhani Luoma-Kyyny, Teemu Karvonen and Jyrki Suokas.

After that, further development of the Blueprint has been channeled to workshops and respective workstreams each being responsible for their own subject areas.

This version (2.5) is based on feedback from IHAN Pilot projects and the work from IHAN workstreams. Three chapters of this document (about Identity, Consent and Logging) will be published during spring 2020 as a CEN Workgroup Agreement.

0.2 Release notes

There are some major changes in the document. First, the end user/data provider/service provider point-of-view chapters have been complemented with a layer approach – identity, consent, services, logging and data transportation.

This release contains the first versions of the cross-cutting concepts in the layer approach, final versions will be released later.

Original version had "empty" chapters that have been removed.

Also, the chapters describing the future development organization have been omitted.

1 Introduction and Goals

1.1 Background

Data economy is the fastest growing part of the overall economy. Companies like Amazon, Facebook and Google have grown to be among the largest companies in the world when measured by market capitalization. The related field of study is called data economics. A very good definition by Aalto University professor Pekka Nikander and Université Paris 13 professor Bruno Carballa Smichowski define data economics as follows:

Today, data and information are two major factors of production. They may affect a firm's production efficiency and competitiveness more than the other factors of production combined. However, from the structural point of view, data is completely different from the traditional factors of production, since data can be efficiently used by multiple actors at the same time, while (most of) the other factors of production cannot. That is, if I have a hammer and some nails, you cannot use the same hammer at the same time with me, and if you use some of the nails, I can no longer use the very same nails. However, if I have a computer programme and a dataset, you can use the same programme and dataset without my ability to use them being diminished.
Because of this structurally different nature of data and the rising importance of data as a factor of production, some scholars have argued that the current market structures are insufficient to efficiently clear the markets. Hence, in order to create a sustainable economy for data, it may be necessary to develop new forms of asset governance (i.e., new forms of "ownership") and new forms of compensation (i.e., new, structurally different forms of "money.")

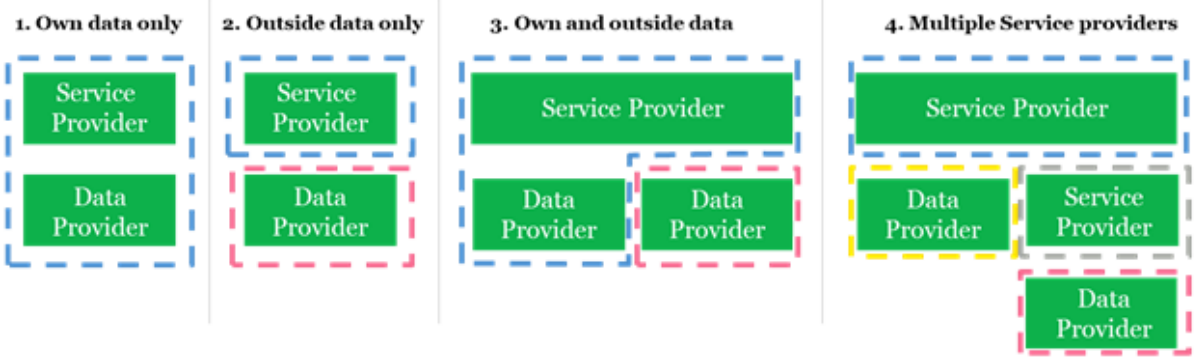
In today's world, the processing rights and privacy terms of personal data are based on a contract between a user and a service provider. In most cases, contracts are based on service provider proprietary terms. GDPR regulation, which came into effect 25.5.2018, gives multiple rights to EU citizens concerning their personal data. Article 20 of the regulation grants the right of data portability, which dictates that EU citizens can order the transfer of their personal data from one data controller to another. Current regulation is a good start, but even with Article 29 Working party clarifications, it does not define the format, governance nor method for personal data sharing in our real-time, many-to-many world.

1.2 About IHAN

Sitra – the Finnish Innovation Fund - has started a project called IHAN that intends to build a governance framework, architectural definitions and requirements for essential components to build a data-driven world. A world where data flows in a seamless but secure fashion enabling new services to be created to create value for all parties: end users, service providers and data providers. The main benefits for all participants are listed below:

- **End Users:** Receive value through relevant services and an ability to control the usage of their personal data
- **Service Provider:** Create new innovative services combining information from multiple sources generating value for customers
- **Data Providers:** Standardized consent management enables sharing end user-connected personal data and creating new innovative business cases around data

These participants should be treated as **roles** rather than individual players or organizations performing a limited set of activities. These roles also overlap, a service provider can also act as a data provider. End users can be individuals or other identified participants in the ecosystem.

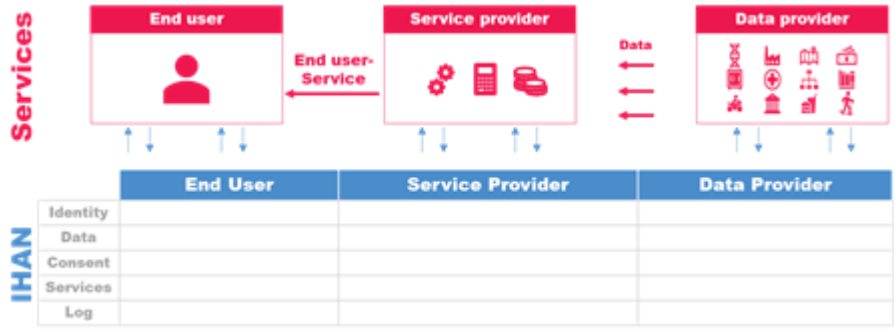


NOTE: IHAN Blueprint is not something that you take as a complete specification and start developing. Blueprint is a collection of requirements that can be used to design IHAN-compatible components or solutions and an overall description of how the components are arranged and how they interact with each other and the surrounding infrastructure. For instance, we describe **what** the Wallet should do but **not how** it should be done. In another example, we describe what the IHAN Identifier is and what it consists of but not how it should be generated in detail or managed.

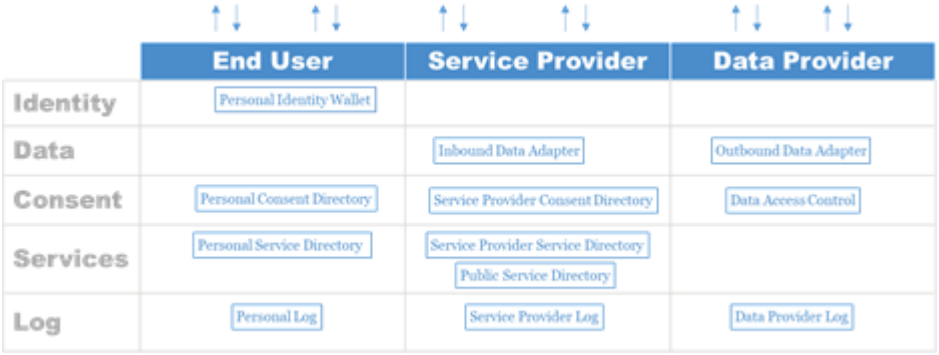
When released at the end of the IHAN project, this Blueprint, together with a reference architecture collection from various business projects, will form the basic tool kit for implementing fair data ecosystem solutions.

2 System scope and context

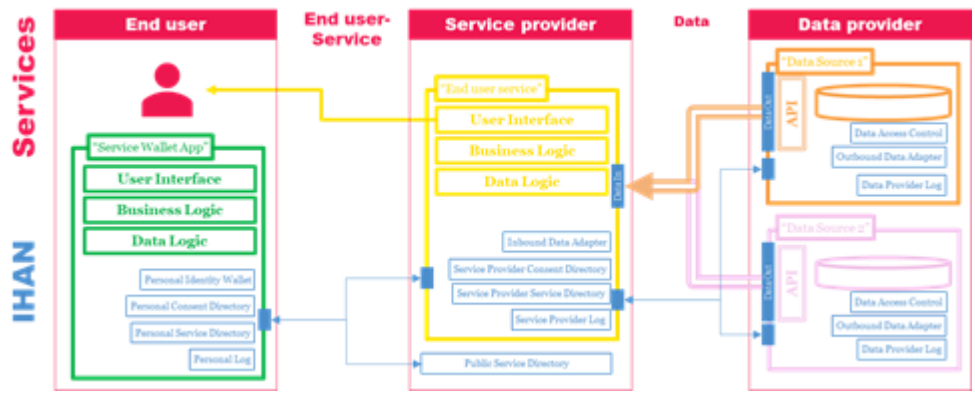
The End User consumes the Business Service “outside” of IHAN. IHAN exposes its functionality in the form of services which the Business Service level uses to build the actual End User services.



This also means that IHAN components do not have built-in user interfaces but provide services so that user interfaces can be built. We have identified the following components that are the initial components of the ecosystem. New components can be added, and old ones modified – even discarded if needed.



Below is an illustrative example of a Service ecosystem sourcing data from two data sources where different developers can concentrate on **their** applications services, user experience, data structures and business logic (green, yellow, orange, pink) and do not need to worry about plumbing (blue IHAN components):



2.1 Business context

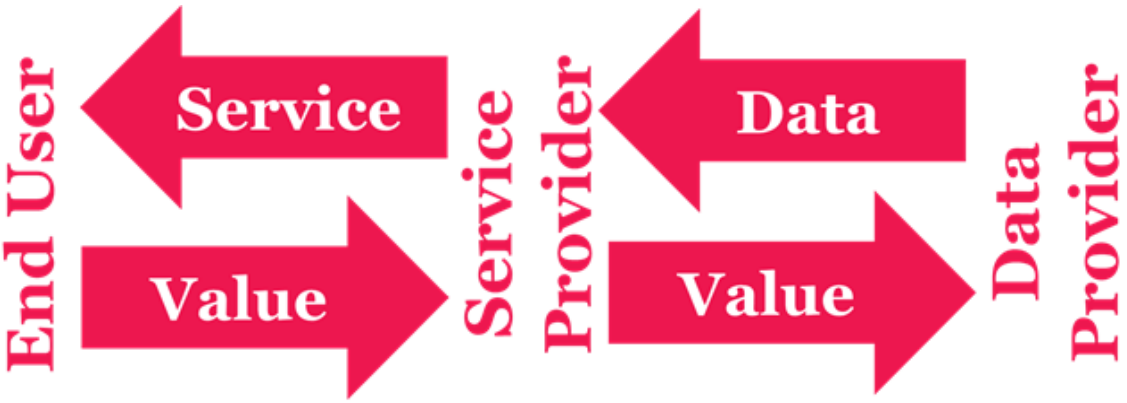
Current data economy was gravitating towards a world where the rights of individuals were overrun by business and revenue-increasing business models:

- On one hand, US-based GAFA companies are hoarding data and using their massive sizes to their own advantage to take over markets.
- On the other hand, centrally controlled Chinese BAT companies have a monopolistic hold in their markets.

Both GAFAs and BATs are also serving European customers in increasing numbers. GDPR introduction and EU initiatives to increase the importance of data economy in Europe have levelled the playing field somewhat.

At the same time, data economy-related regulations like PSD2, where banks are not just forced to build expensive APIs, opening access to accounts and transactions but are also forced to let external parties' initiate payments at no cost. The good intention to open possibilities for new players to start offering new and more innovative services that banks have been able to offer is overshadowed by the unfortunate fact that banks are making a halfhearted attempt to just comply with the regulation instead of embracing being a player in data economy as a new operating model.

For this reason, the IHAN project is giving the business model for data economy considerable attention. Fair value exchange is at the heart of the whole IHAN ecosystem. Not only must Service Providers be compensated for the creation of the Services but, equally importantly, the Data Providers must be compensated for storing data and making that data available. Value can be money or any other form of value exchange that both sides transparently consider to be fair:



2.2 Technical context

There are no further limitations for technical solutions from this documentation. Each of the three functionality levels - End User, Service Provider and Data Provider - may be developed with multiple different architectures and technologies. Having said that, there are some requirements that should be considered during development and met in the end product.

- Several solutions or applications may be developed for the same component by different parties. Although these solutions may be competing, they should be technically compatible to avoid creating software silos that prevent open data exchange.

- Functionalities and software interfaces between components should be standardized to a point where interoperability is straightforward to implement. For example, the interfaces between End User applications and Service Providers as well as interfaces between Service Providers and Data Providers should be sufficiently similar both functionally and technically.

- All three functionality levels should have a standard way to support metadata exchange and consent management as well as usage and actual data exchange. A set of standards and/or best practices should be defined for the mentioned purposes, especially between Service Providers and Data Providers. The set of used standards and practices will increase as the solutions mature.

Even if the Service Providers have data, they have only their own data. For this reason, it should be recognized that the role of Data Providers in this project is vital. No viable solutions can be developed without a vast amount of data provided by Data Providers. That is why it is essential to make sure that providing data to be used to create Services is straightforward and as easy as possible. Also, the concept should be beneficial and profitable for both Service and Data Providers.

It is emphasized that decisions related to technical design and implementation of service components are to be made by the developing organization (and development team). It should be noticed that implementation requires more precise technical design. This document is not a technical or architectural specification that provides full details for implementation purposes. Having said that, it is in the interest of all involved parties that created solutions are generic and based on standards and/or best practices.

3 Important cross-cutting concepts

The purpose of this chapter is to introduce the most important concepts of the IHAN ecosystem. As these concepts span multiple layers and components, it is important to first understand the overall concept of IHAN and its component structure.

The key concepts of IHAN are presented below and in more detail after the table.

Identity	In IHAN context, an identity is represented by a universally unique identifier. A digital identity is a digital representation of person's identity which he or she has decided to use. There can be an unlimited amount of different digital identities for one person, each of which is used for none to many services and data sources. Identities may be verified by a third party.
Consent	For a Service Provider to be able to provide Services, the Service Provider and End User enter into an agreement together – this agreement is the Consent. A consent is the key component that implements the authority of the usage of a data element. Without consent, there can be no interchange of data between Service and Data Providers.
Logging	Logging is a core principle of IHAN. To provide reliable and secure services for End Users, it is essential to collect comprehensive logs of every operation where End User information is involved. Logs need to be immutable and accessible to authorized roles only.
Service	Services are integral part of the data economy. Without services is hard to imagine value exchange and economical flow. To be able to create user-centric services which respect user's privacy and preferences we need to build it around user context and purpose.
Data transfer	In IHAN, the technical data transportation mechanisms are outside of the scope – data and service providers can use whatever technical solutions are fit for their purposes. So, data transfer within IHAN is about different models (direct, aggregator, broker) and requirements.

3.1 Identity

General

Human centric data economy runs around us individuals. Human biology, needs, ambitions, limitations, capabilities and the understanding of our existence defines us. The way we behave and interact with others and the physical world either limit or empower our everyday life. Global economy and societies are built on the system of connected individuals.

Identity is the foundation where all the other data economy capabilities are built on. The way we understand the digitalization of a natural person today is narrow and driven by limited industries, technologies of the past and the interests of a few. In this document we aim to gather a new perspective on how to approach the digitalization of a natural person and thus create the first broader definition of a true global digital identity.

Identity within digital world is hard due to the nature of data which can be in multiple places at the same time, can be easily duplicated, transferred or compromised. Due to that we need to have mechanisms which allow us to create our digital representation of our identity in a way that serve us in digital world without compromising our privacy or losing control over it. Based on the nature of the identifiers we could distinguish between identifiers which are strongly coupled with human being like biometrics and those which are pure digital identifiers which have no connection to analog world.

Identifiers

Natural person identity can be defined through perspective of different identifiers, set of the attributes which allow uniquely identifies a person. Those identifiers can be static like e.g. data of birth which never change but as well as dynamic changing over time like passport number. We could as well distinguish between self-issued like DID (decentralized identifiers) as well as those which are automatically given to us e.g. by government like National Security Number.

We defined identifier as follows:

“Identifier is an attribute which allows, alone or together with another attribute, uniquely identify a person.”

Those identifiers are commonly known as PII’s (Personal Identifiable Information).
This definition is generic enough to do not limit anyone what attributes can be used but specific enough to serve the purpose of protecting a person.

Biometric

Digital representation of human properties, retina, fingerprint, voice, face, heartbeat etc. Any identifier which is tied to human body or physical world provide strong assurance that this person is who he claims but big disadvantage of such identifiers is that once compromised you cannot change. Due to the progress with technology it is easier and easier to steal such identifiers and use them against user. They are important component of set of digital identifiers, but they cannot be the only once.

- Characteristic:
    - hard to lose
    - once compromise hard or impossible to change
    - can't revoke without losing access
    - uniquely bind to the human being
- Examples:
    - DNA Matching
    - Ear
    - Fingerprint recognition
    - heartbeat
    - face recognition
    - Eyes - iris recognition
    - Eyes - retina recognition
    - fingerprint geometry recognition
    - gait
    - hand geometry recognition
    - odor
    - typing recognition
    - vein recognition
    - voice - speaker identification/verification/authentication
    - signature recognition

Non-biometric

Cryptographic material - keys, password, tokens, etc.
Identifiers based on any type of the data not tied to the physical world can be easily changed and randomly generated, use can use different identifiers against different systems, ergo user can have different personas which can be his digital avatars without need to reveal too much information.

It is easier to maintain and create such identifiers user can easily create new once lost

- Characteristic:

  - easy to create new
  - possibility to revoke
  - can have more then one
  - huge variation of types
- Example:
  - DID
  - Public/Private key
  - password

Identity management

As defined in ISO/IEC 24760-1:2019, identity management generally refers to a mechanism comprising of policies, procedures, technology and other resources for maintaining identity information including associated metadata. An identity management system is typically used for identification or authentication of entities. It can be deployed to support other automated decisions based on identity information for an entity recognized in the domain for the identity management system.

Identity information associated to metadata specifies for instance, its origin, scope of use, and period of validity. Identity information metadata can itself be identity information and can be included in the identity it relates to.
Identity information and its associated metadata can be changed. As defined in ISO/IEC 24760-1:2019, the procedures and conditions for changing, updating, and creating identity information can include e.g. following activities

- Requesting and receiving information from external sources
- Verifying and validating the identity information
- Qualifying and categorizing
- Recording
- Provisioning
- Archiving
- Deleting

Identity Management covers the lifecycle of identity information from initial enrolment to archiving or deletion. It includes the governance, policies, processes, data, technology, and standards, which can include:

- An Identity Register
- Authentication of the identity
- Establishing provenance of the identity information

- Establishing the link between identity information and an entity
- Maintaining the identity information
- Ensuring integrity of the identity information
- Providing credentials and services to facilitate authentication
- Mitigating the risk of identity information theft

Trust

Trust in the context of digital identities is built around three central concepts: identity proofing, digital authentication and federation. For centralized and federated identity solutions these core concepts are mainly by defined by NIST SP 800-63-3. It defines identity proofing as the process of establishing that a subject is who they claim to be. Digital authentication establishes that a subject attempting to access a digital service is in control of one or more valid authenticators associated to the subject's digital identity. Successful authentication provides reasonable risk-based assurance that the subject accessing the service is the same as the one who has accessed the service previously. Federation is the process to convey results of authentication and relevant identity information to Relying Parties.

3.2 Consent

Legal view of Consent

As IHAN is based in the European Union environment, the General Data Protection Regulation should be our legal framework for establishing the Fair data economy. Several Articles of GDPR are touching upon consent and have to be taken into account, listed below for an overview:

- Article 4 (11) of GDPR defines *consent* as:

> *'consent' of the data subject means any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her;*

- Article 6 (1a) of GDPR defines having consent of the data subject for processing of his personal data (for one or more purposes) as one of the possible six legal bases for lawful processing of personal data.
- Article 7 of GDPR defines the conditions for consent:
    - It is up to the controller to demonstrate that the data subject has consented to processing of his personal data.
    - If in written form, the request for consent must be clearly distinguishable from other other matters and clearly understandable.
    - The data subject has the right to withdraw consent at any time.

    - Consent must be freely given.
- When processing personal data of children under certain age consent must be given by the holder of parental responsibility (GDPR Article 8).
- When processing special categories of personal data (more sensitive in their nature - e.g. health data) an explicit consent has to be given unless other exceptions apply (GDPR Article 9).
- Article 13 of GDPR defines the information that has to be provided to the data subject when personal data are collected. This applies also to cases where given consent is the legal basis for processing of personal data, but not exclusively to them. The data subject has to be informed of any further processing outside of the initially communicated purposes.
- In the case that personal data are not obtained directly from the data subject, Article 14 defines similar information about terms of usage of his personal data has to be provided to the data subject.
- Article 41 of GDPR describes monitoring of compliance with codes of conduct by supervisory authorities. Consents should therefore be recorded in such a way to facilitate the monitoring process including tasks described in Article 57 of GDPR (e.g. (a, f)).

Technical view of consent

IHAN Blueprint requires that with the act of giving consent, the technical means to access the data to be processed are also given to the Data controller. This means that with consent, authorization details to get to the data residing with the Data provider have to be given. The authorization details have to be properly protected to be only accessible by required parties. The metadata about the naming / structure of the data have to be passed in the consent, for semantic interoperability. The Consent receipt therefore has to have additional data, that enables this.
Structure of Consent:

- Human readable part describing the relationship between End user and Service provider (what kind of data is used, purposes, etc. - allowing for GDPR compliance). Kantara specification should be used to generate a Consent receipt. Publicly available vocabularies can be used for describing the fields (e.g. W3C Data Protection Vocabulary).
- Data specification part describing how to get to the data, the APIs, metadata about the data. This should include descriptions of which fields might contain personal identifiers, sensitive data, etc. and implying how the data should be processed.
- Authorization part with authorization details to access the data at a Data provider.

Business view of consent

With properly recorded consents, the level of regulatory compliance for companies should increase. Not only that, but the trust the individuals assign to those companies should increase as well - as now individuals have greater control and overview of how their personal data is used.
To make interoperability as seamless as possible, the structure of the Consent receipt in the IHAN ecosystem must be common. This makes it possible for companies to more easily join the IHAN Fair data economy and get to the data they need. It also makes it easier for individuals to have greater control over the data they shared with a Service provider or are keeping at a Data source. With a standardized Consent receipt, an individual could be offered services on top of his “shoebox” of consent receipts, to see which companies have which kind

of data about him and exercise different actions on them (updating, revocation, GDPR portability right, etc.).
If at every kind of data usage by a Data controller (company), one would issue a Consent receipt to the individual, the individual saving his Consent receipts in his “shoebox” would in time get a total overview of what kind of data companies keep about him, where his data resides, what it is being used for. (Note that not every usage of data is based on consent, so a more appropriate term for such a receipt might be a Data receipt.) Assuming also that the information in the Consent receipt is semantically equipped, this would open the possibility for him to leverage services that are based on that data, even if they are not under his direct control (e.g. are stored somewhere else). This, for example, opens opportunities for various analytics services to offer him insights based on that data, as well as enabling more complex scenarios.

Summarizing consent

Consent in the IHAN sense could be viewed from the three facets described - legal, technical and business. It enables data exchange between Data provider and Service provider, while giving the individual End user control over how their data is used. It can and should be used even if no data exchange is required - e.g. if the Service provider is also the Data provider. The Consent must be recorded in digital form and available to both the individual (End user) and the Service provider, possibly in part also to the Data provider(s). The individual must also have the option to view (relevant parts) of the contents of the Consent in human readable form. A digital receipt recording the details about the act is called a Consent receipt.

3.3 Logging

Logging in IHAN

IHAN is about Human Centric Fair Data Ecosystem where giving and changing the state of consent related to data is an essential enabler. Contracts and agreements between the End User and IHAN service and data providers must be reflected in the logging so that the End User can verify that these arrangements have been respected.
In order to gain End Users’ confidence in the system they must have proof that their will about data has been honored. In practice this means that reliable and chronological track record of events has to be available for inspection at all times. Without immutability and proof of respecting End Users given consents there is a low trust level into data economy.
Identity and consent management combined with undisputed and immutable logs provide the fundamentals for transparent trust in circumstances where trust is not self-evident.
Logging ihan will build the framework to answer questions for fair data economy implementors and users.

- How can I as a service provider prove I had the end-user’s explicit consent for processing the personal data?
- How can I as an end-user see, who has accessed my personal data?
- How can I as a data provider prove that I have shared personal data with the end-user’s explicit wishes.

Logging in IHAN context is not focused on technical nor application / service internal logging. The focus is to log user data related events: What data of a user is transferred between different stakeholders? How can the trace of events be tracked in a meaningful way? How is the immutability of the logs ensured?

Architectural models

In case there are multiple logs upheld of the same event (e.g. data source and data subject's operator both hold their own logs), there should be either a syncing or verification method between the logs, to resolve potential non-consensus situations.
Logging can be done centrally inside a single role, in case which that party needs to provide all the logging needs for that use case. A decentralized approach would allow distribution of logs between roles, while maintaining logical connections between them. Another approach to distributed logging holds the log in a decentralized storage and makes the immutable log accessible to all parties, making only one version of the log needed. Swarm presents an example of this kind of architecture.

Centralized model
In a centralized model, each role keeps their own log, and is centrally responsible for its integrity.

Decentralized model
In a decentralized model, a globally accessible distributed database, such as a distributed ledger or blockchain, is used to maintain a single source of truth for logging.

Data operator

A Data Operator manages some aspects of the exchange of data, on behalf of other entities (companies, organisations, or individuals). A Data Operator may concentrate on managing the technical transfer of data, on operating the consent management process, on negotiating deals for secondary use of data, on calculating and distributing the income from licensed data between entities, or perhaps on all of these aspects.

Distributed identity agent

A distributed identity agent is a software that can be used for creating secure messaging channels with other similar agents. An agent can be used on behalf of an End User or an organization for exchanging digitally signed messages, such as verifiable credentials. An agent can receive loggable events, such as reception or revocation of a consent.

Data provider

A data provider holds information about the End User and offers interfaces for retrieving data. The data provider concentrates on verifying that the data requester has proper consent and/or authorization to retrieve the data. A data provider can be for example a simple API service.
The data provider is responsible to the End User's and authorities to uphold personal data with utmost care. Data provider's interests are in proving that data is shared and processed

according to one of the six legal bases (see the table below) defined in GDPR, and that they have done their due diligence when somebody requests access to End User's data.
The following information should to be logged:

Aspect	Notes / Example
What operation was performed?	Catalog of possible operations for data modelling and distributed logging purposes
Which component/system performed the operation?	Catalog of components in the system for data modelling purposes
Who initiated the operation?	Identifier of the authenticated individual or organization, if such identification can be performed, and is in scope of the service.
Which component/system received information about the End User?	Catalog of components in the system for data modelling purposes
What End User information was handed over?	Metadata structure / elements transferred
When was the operation performed?	Timestamp synchronization in a distributed system
Did the operation succeed?	Verification method
Which consent was used?	Catalog of consents used in the system.

In short: data provider should log data access and what basis that access has had.
NOTE: IHAN application or service may require the use of non-IHAN compliant data providers. Due to regulatory or other reasons, access to the logs of these components may be restricted or prohibited by data operator.

End user

The End User is interested in having transparency regarding his/her personal data. Who has accessed, and by what grounds. What kind of consents of delegations of authority one has given, and how they are used.
At least the following data is logged in the End User's personal log:

Aspect	Notes / Example
Identity changes (Operation?)	• For example, a new Identity Record is created, I.e. a new identity provider (3rd party) and credentials are linked to the Personal Identity Wallet, or an existing one is removed or modified • Catalog of allowed transitions for identity in the system
Definition of identity in the context	• Identity definitions provided in the section describing the End User identity in the IHAN context
Service changes	• For example, a new Service Provider is added to Personal Service Directory • Catalog of allowed transitions for service in the system
Service usage	• For example, the End User uses a service provided by a Service Provider • Catalog of services in the system
Consent issuance	• When End User provides a new consent for accessing or fetching personal data
Data usage	• For example, a new Data Access Record is created to be used with a selected Data Provider or a Service Provider uses a Consent to access data for a Data Provider • Metadata structure / elements accessed

Aspect	Notes / Example
Revocation of consent	• a previously issued consent is revoked for data users

3.5 Services

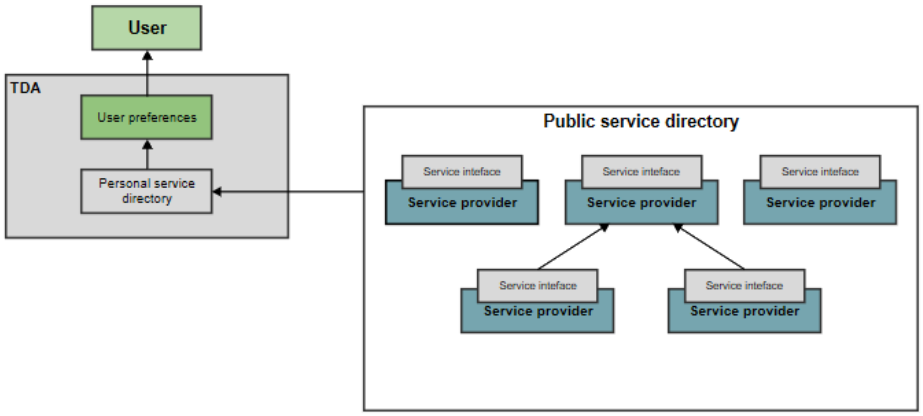
Services are important part of our economy and definitely one of the major components of emerging decentralized data economy. To be able to create user-centric services which respect users privacy and preferences we need to build it around user context and purpose. This eliminate need for consenting to each service independently which in practice help user to maneuver within vast amount and not understandable privacy policy and terms and conditions, protecting their privacy and giving him chance to protect their data.

Without services is hard to imagine value exchange and economical flow. To help with transformation of existing services into new decentralized data economy we defined major components which are required for decentralized data economy services to operate properly.

We can distinguish

- personal directory
- public directory
- service interfaces
- user preferences

Below diagram presents structure of above components.



Use cases / requirements

- End user is able to filter Services based on Personal Service Directory preferences
- End user is able to add new Services to the Personal Service Directory from the Public Service Directory
- End user is able to remove Services from the Personal Service Directory
- End user is able to discover new Public Service Directories
- End user is able to unsubscribe from Public Service Directories
- End user is able to enlist their own user data as a data source where applicable

- Changes in service provider services must be updated automatically to the end user's Personal Service Directory
- A Public Service Directory may recommend a service which requests rating of services from users which it may maintain for ranking services
- The Personal Service Directory may suggest new uses for user's data as a data source (data source applicability discovery)
- The Personal Service Directory can be connected to a "Trusted Digital Assistant" or similar automation tool which acts based on the user's preferences
- Service Provider is able to add new Services
- Service Provider is able to modify existing Services
- Service Provider is able to delete existing Services.
- Service provider is able to list all Data Sources providing specific data elements
- Data Provider is able to add new Data Sources
- Data Provider is able to modify existing Data Sources
- Data Provider is able to delete existing Data Sources

Personal directory

Personal directory acts as the user's personal repository of all services which are relevant for the user. The sources of those services could be from multiple public or private directories. The personal directory is connected directly with Trusted Digital Assistant (TDA) within which user sets their preferences and control access of the services to particular data sources. The personal directory aims to convey the user's purpose and explicit purpose in an automated way.

The TDA serves a user, through automation of components within and possibly without the Personal Directory which may benefit from the implementation or use of it.

Public directory

The public directory should be an open and decentralized allowing anyone to join and submit their own services. Some public directories could require special conditions to join their registry, e.g. following specific regulation or fulfilling specific purpose. It is up to the operator of the public directory to deiced about those rules and manage their participators as well as a level of trust which would be provided to the user.

The public directory provides a set of open, standardized interfaces which may be called for example by the TDA to pull the list of the services and synchronize it with private directory.

More than one public directory may exist, personal directories may choose to connect to any number of them.

The public directory may provide the user with the appropriate consent recommendation for a purpose (ie. what is considered the norm.) this information may update over time based on public consensus of the norm and evolution of services and purposes.

The public directory may manage inter-dependencies between different service providers.

Finding new Public Directories can occur through "out of bound" discovery, such as word of mouth. However, it is expected that should there be a need for a service which offers searching capabilities within this space, such services will be created and added to the most popular Public Directories, which then facilitate searching of other Public Directories.

Service interface

To be able to interact with any services within decentralized data economy we have to define common set of the interfaces. Those interfaces expose all functionality of the services as well as required metadata, data, terms and conditions, service agreement, dependency on other services and more.

User preferences

User preferences are set of metadata stored with TDA. Preferences are set of the attributes describing user values and their will. Through the preferences TDA can automatically match user with specific services which match to their profile or automatize managing consent for the user.

User preferences are strongly linked to the Consent construct described within Consent Chapter which describe how consent look like and explain difference between public and private purpose-based consent which allow user to automate his consent base on the context and the purpose.

Data source

The data which is shared by data sources for the use of services shall be made available using the standard method described by the Data Source components. The standardized format should be aligned with other streams.

The data source offers data based on some terms, which may include SLA’s, Consent, Pricing, Logging protocols etc. This should all be standardized in such a way that Service providers can shop for data providers easily, and that it can be abstracted away by service providers, so users don’t need to see the complexity.

Any information relevant to the users consent should be communicated via the user consent channels. The user can act as a data source.

3.6 Data transport

Data transfer models

We divide data transfer models into three categories that are Direct model, Aggregator model and Broker model. Different models naturally fit for different purposes or for different stakeholders.

Direct

Direct models are used by companies that have strong business connection or partnerships. In direct model companies create point-to-point APIs with each other or based on the direction data is planned to flow. For example, if Company B wants to utilize Company A's data then Company B only needs to integrate to Company A's APIs. Direct model is suitable for connection between partnered parties.

Aggregator

Aggregator models are used by companies that either represent individual persons or otherwise collect and aggregate individuals' data for different purposes.

Broker

Broker does not store or use the data content. Best use of a broker is to act as a hub which can deliver same message to multiple receivers and change the protocols and data structures. This enables the receiver to use protocols that are not supported by the original sender.
Broker is not a data handler but more like technical operator.

Data transfer requirements

Independent of the chosen data transfer model the data transfer quality must be based on the agreement. In many cases it is crucial that all data arrives to destination and in right order, but in some cases, it might not matter if some packages are lost, or received in wrong order. The technical layers of data transmission might take care of the integrity, but it depends on the technology. But it is always possible that the connection breaks in situation where the transmitter cannot know if the previous message was received or not. It can detect that the ACK does not come but not if was sent by the receiver or not. To ensure integrity a rolling data packet id is recommended. Based on that the receiver is able to request for restart sending starting from certain packet.

Data transfer model does not solve the issues above but might change the control points. Basically, there are two different possibilities in the transfer mode: polling or streaming / broadcasting.

- In polling the receiver asks for data. There it is easy to control that the received packet has the end element

- In streaming the receiver picks data whenever it comes. A long pause in the data flow does not indicate a fault until the next comes. Streaming would be efficient for the sender especially if the same data is sent to multiple receivers. It is not recommended if data loss is not allowed.

3.7 Architecture constraints

Any decision that takes the ecosystem away from the quality goals – fair value exchange, decentralized instead of centralized and secure handling of personal data – must be avoided at all costs.

Here are some of the basic architecture principles concerning IHAN

- Architecture must support implementation of services that enable ”an easier, smarter and happier life for individuals”
- Architecture must provide solutions that enable individuals to gain control over their own data
- Centralized, single-point-of-control solutions are not recommended
- All solutions, programs and applications must be based on a recognized requirement or a set of requirements
- Reusability is recommended
- Solutions should be inter-operational
- Application design must be user centric and ease-of-use based
- All solutions should be technology independent
- New technology experiments must ensure performance and scalability
- Architecture must enable implementations that comply with data regulations concerning person-level data (GDPR, PSD2)
- All solutions must enable secure data management through the entire life cycle of data.
- Architecture must support the management of the individual’s multiple virtual identities
- Architecture must support logging, auditing and trust
- Architecture must enable secure data transfer, management and storing

3.8 Other guidelines

Although GDPR is one of the central and fresh regulations around IHAN, existing old regulation cannot be ignored. This means that all IHAN compatible solutions and services should comply with relevant regulation. For example, an individual might give consent to service providers to access clinical healthcare data – which is not possible due to existing regulation concerning sensitive health information management.

As IHAN Blueprint is a requirements document, quite many pervasive issues (security, quality, standards etc.) are not so widely covered. For example, when deciding the levels of authentication required for a service, IHAN states that there needs to be authentication but does not define what the level is or what mechanisms should be used. Another example from healthcare – the decision to use HL7 standard is something that needs to be agreed between data and service providers, regardless of IHAN.

In general, IHAN is or will be a new way to do things – not a platform, program, solution or even a standard. Companies that decide to create new services according to IHAN and fair data economy principles, do not necessarily need to renew or rebuild their existing technical solutions.

4 Requirements overview

In the following chapters, the IHAN functionality is divided into three sections

1. **Setup** functionality – related to creation and updates of primary components like Wallets, Services and Data Sources

2. **Management** functionality – related to changes to status of main components like Consent creation, Consent management and Service changes

3. **Usage** functionality – related to delivery of services like using Consent to access Data Sources and providing Service to End User

End user Setup	Service Provider Setup	Data Provider Setup
End User Manage	Service Provider Manage	Data Provider Manage
End User Usage	Service Provider Usage	Data Provider Usage

4.1 End User Point of View

The primary functionalities of the End User level are related to Identities and Services.

4.1.1 Setup Functionality

End Users can:

- create new **Personal Service Wallets**
- delete existing Personal Service Wallets
- Modify an existing Personal Service Wallet – for example, suspend access for a set period (in case of - for example – if a device gets stolen)
- recreate Personal Service Wallet (in case of - for example - a missing device)

A Personal Service Wallet always contains **Personal Identity Wallet**, **Personal Service Directory** and **Personal Log** components.

4.1.2 Management Functionality

End Users manage their identities and access to personal data (located in several systems) in **Personal Identity Wallet.** Personal Identity Wallet allows the End User to manage multiple identities and services for which the user gives the data access to. Depending on the identity and the source of the identity, it might or might not require a third (trusted) party verification.

Personal Service Directory contains a record of all current and past Services of the End User. If a service provider wants to use data located in external sources (Data Providers), it needs to ask for a Consent to use it. When the End User discovers a service that he decides to begin using, a new service subscription in the form of a Consent is created. Service Provider is provided with Consents containing all needed Data Access Records to retrieve data from all related Data Providers. Consent is stored in End User's **Personal Consent Directory.**

All changes to identities, data access, consents and service subscriptions are logged and stored in **Personal Log**.

4.1.3 Usage Functionality

When a Service Provider invokes a service, the consent validity for that Service is checked by the Service Provider.

4.2 Service Provider Point of View

When providing Services, a Service Provider needs consent to use data located in external sources (Data Providers). End User creates a consent form for a Service Provider and specifies details for this data usage. A Service Provider will then use the consent form to get data from the specified Data Provider.

A Data Provider will receive consent and provide the Service Provider access to related data. There may be multiple ways to provide data access depending on the interacting systems used. Service Provider will use the data (or allowed data access) to create Services for an End User.

All instances of data access, data transfers and other relevant data actions will create a log entry for every involved party (End User, Service Provider and Data Provider). NOTE: Actual data contents are not written in log entries.

4.2.1 Setup Functionality

Service Providers can create new Services and publish them in a **Public Service Directory**. There can be multiple (physical) instances of directories, but logically,

from an End User's point of view, they all appear as one centralized Public Service Directory.

Service Providers must register their Services on the Public Service Directory. Service Description contains both technical and human-readable documentation of the service, most importantly describing the needed Data Sources in detail. The Data Sources list can contain both mandatory and optional data elements and it is the Service Provider's responsibility to ensure that the Service Description clearly outlines what value the service provides with mandatory data and what additional value comes from optional data / data clusters.

Published Services can be modified and deleted by Service Providers. A Service describes in detail - through metadata specifications - what kind of data elements are needed to produce the service. The best analogies for Public Service Directory are today's app stores.

4.2.2 Management Functionality

Service Provider Service Directory contains detailed description of each Service the Service Provider is offering.

End user can browse all Services in the **Public Service Directory.** End user can subscribe to Services that match the active data elements provided by data sources the End User has in his Personal Service Wallet and that he hasn't subscribed to already. While subscribing, if some data elements are missing, the Service Directory shows potential sources to those elements. This can prompt the user to connect more Identities and Data Access Records (=IHAN Identifiers) to his Personal Service Wallet and lead to new Services being taken into use. Service Providers can also promote Services for End Users that have opted in for the category of Services that the Service Provider is offering. If there is a match between the data elements that the End User possesses, the End User is notified of this new Service.

Service Provider stores up to date instance of End Users' Consents that have subscribed its service in its own **Service Provider Consent Directory.** If the End User modifies or revokes consent, then this information is automatically passed along to the Service Provider.

All changes to Services will be created a log entry to the public section of **Service Provider Log.**

4.2.3 Usage Functionality

When a Service is called, the Service Provider fetches the End User's Consent from its own **Service Provider Consent Directory**. The consent must be secured in a way that it cannot be tampered with. Service Provider then uses the Consent form(s) to request the data from (one or more) Data Providers.

If data retrieval through **Inbound Data Adapter** is successful, the Service Provider creates its Service using its own business logic and the retrieved data. Finally, the End User consumes the Service. Consent contains the criteria for data usage purposes and may contain rules for what must happen to the data at Service Provider after the service provision. In the consent form, the End User can specify what the service provider must do with the data after the service provision: should it be kept, archived or deleted.

All Service evocations are immutably logged in **End User Personal Log** and the private section of **Service Provider Log.** Part of the log can be used for Value Exchange information collection – billing itemisation.

4.3 Data Provider Point of view

Data Provider has data and gets data from business activities – for example banks that store credit card transactions or retailers that connect purchases to customers when they use loyalty cards. Pure storage vendors are not included in this scenario.

4.3.1 Setup Functionality

Data Providers can create new **Data Sources** and publish them in **Public Service Directory.** By browsing the Public Service Directory, the Service Providers know which Data Elements are available at which Data Provider. Data Sources can be modified and deleted by Data Providers. The **Public Service Directory** can be used as a marketplace for available data, rather than just being a place to offer the minimum interface required by regulation (PSD2 for example).

Data Sources represent available data sets which can be files, databases, documents, etc. To perform data transfer, Access Mechanisms need to be assigned to **Data Sources**. Data Providers present their assortment/selection of available data sets via availability services. In addition to availability itself, services provide a view to data properties based on data source metadata. Properties may include data descriptions, basic statistical information and certain quality aspects of data source contents.

All changes to **Data Sources** will create a log entry in the public section of the **Data Provider Log**.

4.3.2 Management Functionality

Public Service Directory contains a record of all Data Provider **Data Sources** that provide data elements for Services (provided by Service Providers).

The Service Provider may register a Service as a Data Source - thus enabling a model where a Service Provider can act as a subcontractor for other Service Providers.

During this use case, the End User Consent Form is provided to the sub-contracting Service Provider to access data from a Data Provider.

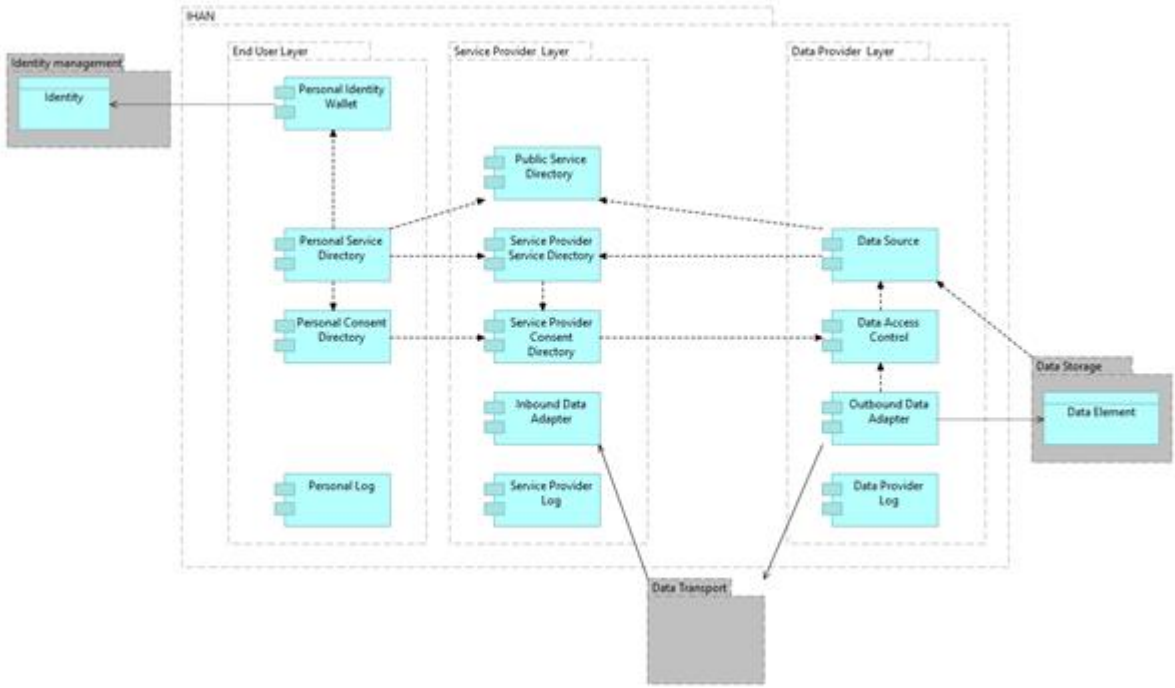
4.3.3 Usage Functionality

When a Service Provider wants to access data using a Consent Form, the **Data Access Control** on the Data Provider's side uses the credentials within the Consent to retrieve needed data elements (which **Outbound Data Adapter** sends to Service Provider)**.** The actual sending process depends on the Data Routing method.

All data requests will create a log entry in the **End User Personal Log**, private section of **Data Provider Log** and the private section of **Service Provider Log** of the Service Provider that requested the data. Part of the log can be used for Value Exchange information collection – billing itemisation.

6 Building Block View

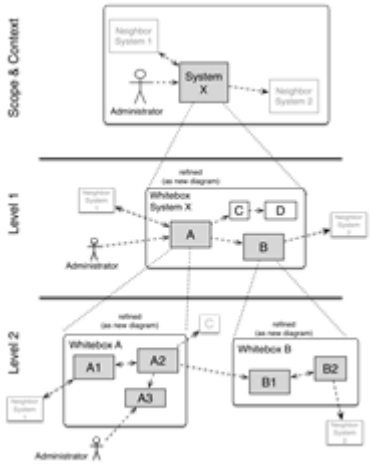
The building block view shows the static decomposition of the system into components as well as their dependencies. In analogy to a house, this is the *floor plan*. In the diagram below, the IHAN ecosystem Level 1 components and their relationships are described.



The building block view shows the static decomposition of the system into components as well as their dependencies. In analogy to a house, this is the *floor plan*.

6.1 Whitebox Overall System

In this chapter, all IHAN components are described in general. The chapter contains a Level 1 white box description of the overall system together with black box descriptions of all contained building blocks. Further elaboration work by Technical pilot projects will create Level 2 descriptions with further details (if needed).



Level 1 is the white box description of the overall system together with black box descriptions of all contained building blocks.
Level 2 zooms into some building blocks of level 1. Thus, it contains the white box description of selected building blocks of level 1, together with black box descriptions of their internal building blocks. At this point, no Component is yet described on Level 2 – as the first Technical Pilot Projects begin producing deliverables, the Level 2 descriptions for those components the Technical Pilot Project is working on will be added.

6.1.1 Personal Service Wallet

Personal Service Wallet (PSW) is a sub-system name for all functionalities at the End User level.

Requirements

Minimum requirements for Personal Identity Wallet are described below:

- End User must be able to create a service wallet that contains his/her identities, his/her services and logs of usage of thereof
- End User must be able to open Personal Service Wallet and access functional entities contained therein
- End User must be able to permanently delete a Personal Service wallet
- End User must be able to restore Personal Service wallet that End User has lost control to. All Identities, Data Sources, Services and Consents are also restored.

6.1.2 Personal Identity Wallet

Purpose and Responsibilities

Personal Identity Wallet (PIW) is a component for storing Identity Records and Data Access Records, the latter of them containing access credentials used to access specific data sources using identity.

1. An Identity Record describes the identity – i.e., needed access credentials like username and password.

2. Zero or more Data Access Records use the identity with individual access credentials for each data source to access data.

A combination of Identity Record and Data Access Record forms the IHAN Identifier, which is used by Data Provider Access Control to provide data.

Requirements

Minimum requirements for Personal Identity Wallet are described below:

- End User must be able to add new identities
- End User must be able to modify existing identities
- End User must be able to delete existing identities
- Wallet must support several types of identities with several authentication levels: from strongly authenticated identities to anonymous identities managed by the user
- Identity must be separated from Data Access. An Identity must be able to have zero or more Data Access Records, each of which must use the Identity combined with Data Source-specific credentials for data access. The identity alone must not be used to provide data access - making the role of Data Access Records (and the IHAN identifier) essential.
    - for this to work, the identity must be connected to the appropriate data set at the Data Source. (See Chapter 5.1.11 for Data Source requirements)
- All actions in the Personal Service Wallet must be logged in Personal Log
- It should be possible to link IHAN wallets to strong electrical identity management systems and related identifiers, such as social security number, electrical ID number, passport or any other data.
    - in this case, the IHAN wallet creates a link between the digital and real world
- Presentation of an identity should depend on the identity type. Some identities - like electronic passports - render a representation of the data in a predetermined format that can allow for a document to be used as an identification mechanism in the real world.
- When the Wallet shares a Data Access Record with a Service Provider, the Access Record must not reveal security critical information – for example, End User credentials – to the Service Provider

Sample Functional Flow

Sample functional flow is presented below:

.

1. End User can connect data access to an identity by providing valid credentials, so data access can be tested.

2. If verification is successful, the Data Provider Entry in the Personal Service Directory is populated with metadata information provided by the Data Access Control Management subsystem at Data Provider.

3. A successful verification creates a Data Access Record in Personal Identity Wallet– a combination of identity, access credentials and data source address.
4. The record can be shared with Service Providers, so they can access data at Data Provider without storing any End User data locally.

5. Data Providers always verify the Consent that Service Provider is uses against the Data Access Record. Access credentials can be stored in any form and Identity Wallet does not contain clear text versions of the credentials.

Restrictions
The following restrictions should be considered in implementation:

- IHAN does not depend on any specific Personal Identity Wallet systems implemented as it manages identity as part of the ecosystem it is working in

Interfaces including Data Streams

Inbound data:
- Identities from 3rd parties (when applicable)
  - for creating Identity Records
  - A standard API must be provided
- Data access verification from Personal Service Directory
  - for creating Data Access Records
  - A standard API must be provided

Outbound data:
- Sharing Data Access Records with Service Providers
  - to access data with Data Providers

Technical standards
API and methods for providing identities in the Wallet from 3rd parties must be standardized using current best practices and standards – SAML, oAuth, OpenID Connect or another widely used standard approach.

API's provided by Personal Identity Wallet should be RESTful. Data should be structured using JSON or XML. All communication between decentralized components should be secured using HTTPS-connections.

Apart from those mentioned, there are no technology constraints that limit component implementation, for example, using a specific programming language.

Quality and Performance

- There may be several Personal Identity Wallet systems that should be interoperable

6.1.3 Personal Service Directory

Purpose and Responsibilities

The Personal Service Directory (PSD) manages End User's Services. The Personal Service Directory contains descriptions of all the Service Provider Services that the End User has subscribed to. The Personal Service Directory is used to grant service providers access to data providers so that service providers can produce the service for the End User. As there can be more than one physical Public Service Directory, the End User's Personal Service Directory creates a logical unified view of all services for the End User**.**

Over the course of time, the Personal Service Directory will begin forming into a GDPR dashboard, showing the different services that an End User has access to and what data is behind each identity.

Requirements

Minimum requirements for the Personal Service Directory are described below:

- An End User must be able to list all Services and constrain the list based on filters in the Personal Service Directory
- The End User must be able to add new Services from the Public Service Directory. If an End User is willing to begin using a new service, the Personal Service Directory uses the Personal Consent Directory to automatically grant the necessary consent forms for the Service Provider that are required to access data from all necessary Data Sources. This information is stored in the Service Provider's Consent Directory. Service is also stored as activated in the End User's Personal Service Directory and consent forms are linked to it.
- End Users must be able to modify existing Services
- End Users must be able to delete (unsubscribe from) existing Services. If a Service is deleted, the corresponding consent form's validity must be terminated.
- The Personal Service Directory could also actively propose new services through the "Data Sources Available" Service discovery process, which requires opt-in from the End User for specific and narrow kinds of Services, which are available for the End User based on the Data Sources the End User has in his/her Personal Identity Wallet.
- Personal Service Directory could also actively propose new services through the "Data Sources Missing" Service discovery process available for the End

User based on the Data Sources that an End User does not have Personal Identity Wallet, but these Data Sources are common to the user profile that an End User has. For example, even if an End User has not connected his/her bank as a data source - where account transactions would be available - it is reasonable to assume that the End User could have this Data Source from any bank available. Hence this prompts the End User to connect more Data Sources to his Identity Wallet.

- An End User could rank a Service and this information could be stored in the Public Service Directory

Sample Functional Flow

1. The Personal Service Catalog contains a record of all current and past Services of the End User.
2. To find new ones, the End user browses the Services in his Personal Service Directory and sees them in the "Available new services" section.
3. When the user finds a Service that he wants to begin using, he signs up for it. If a Service Provider wants to use data located in external sources (Data Providers), it needs to ask for consent to use it. Consent is stored in an End User's Personal Consent Directory.
4. The Service Provider is provided with consent forms containing all necessary Data Access Records to retrieve data from all related Data Providers.

Restrictions

Interfaces including Data Streams

Inbound data:

- Services from Public Service Directory
  - for subscribing to new Services
- Data Sources from Personal Identity Wallet
  - for subscribing to new Services

Outbound data:

- Consent forms for Personal Consent Directory
  - Personal Service Directory uses Data Source information to create the necessary Consent forms

6.1.4 Personal Consent Directory

Purpose and Responsibilities

Personal Consent Directory (PCD) stores all an End User's consent forms submitted to Service Providers. Service Providers will use this information to access data from Data Providers.

Personal Consent Directory contains Consent information for each service. Each Consent form defines all Data Access Records that will be used to request data from Data Providers. There will be at least one Data Access Record for each Data Provider from which the Service Provider will request data.

Requirements

Minimum requirements for consent forms and the Personal Consent Directory:

- End Users will create consent forms which must be stored in the Personal Consent Directory
- These consent forms must have at least two parts:
    - Part 1 must be readable only to the Service Provider and must contain information about the Data Providers (interfaces for metadata, and data request)
    - Part 2 must be encrypted for the Data Provider in a way that only the End User and the Data Provider can understand it. The Service Provider will send this part "blindly" to the Data Provider based on the information in Part 1.
    - There could be multiple Part 2–type of elements, one for each Data Provider

Restrictions

The encryption/decryption mechanism as well the needed Secured Key Exchange mechanism are outside of the scope of this IHAN Blueprint.

Interfaces including Data Streams

Consent creation will have interfaces to Personal Service and Personal Identity Wallets and use information from these components.

Consent forms will be sent to a relevant Service Provider based on information from the Service Wallet.

6.1.5 Personal Log

Purpose and Responsibilities

Personal Log (PL) is the End User's private log that stores log entries created by the following processes:

1. Identity changes
2. Service changes and usage
3. Data usage

All actions in the Personal Service Wallet are logged in Personal Log.

Requirements

- The following processes must create a log entry for the Personal Log
    - Identity changes – for example, a new Identity Record is created, i.e., a new identity provider (3rd party) and credentials are linked to the Personal Identity Wallet, or an existing one is removed or modified
    - Service changes – for example a new Service Provider is added to Personal Service Directory
    - Service usage – for example, the End User, uses a service provided by a Service Provider
    - Data usage – for example, a new Data Access Record - is created to be used with a selected Data Provider or a Service Provider uses a consent form to access data for a Data Provider
- A Personal Log must contain all personal log entries associated with the End User regardless of the system or actor that performs the operation
    - This leads to the requirement that Personal Log, Service Provider Log and Data Provider Log must be connected to each other in a standard way. Service Providers and Data Providers must either have access to End User Personal Log API or the logs must be controlled as a shared ledger.
- Personal Log must not contain entries apart from personal ones, i.e., entries of operations concerning the End User who owns the current Personal Log
- Log entries must be created in standard format containing at least the following information
    - What operation was performed?
    - Which component/system performed the operation?
    - Which component/system received information about the End User?
    - What End User information was handed over?
    - When was the operation performed? (timestamp)
    - Did the operation succeed?
    - Which consent was used?
- Personal Log must provide standard APIs for creating and retrieving log entries
    - Personal Log APIs must be secured on a personal and a system level. Access to writing log entries must be restricted to authorised systems only. Access to Personal Log entries must be restricted to End Users only.
- The Personal Log must comply with GDPR, so personal information - like identifiers, personal information and credentials – must not be logged.
- Personal Log entries must be accessed only by the End User and by using Personal Service Wallet functionalities to do so. Personal Log entries must not be accessed from outside the Personal Service Wallet.

Restrictions

- A Personal Log is a storage for log entries. It provides APIs for creating and retrieving log entries does but does not provide a user interface. A user interface may be built separately.

Interfaces including Data Streams

Inbound data:

- Log entries from End User components
  - for creating log entries
  - A standard API must be provided

Outbound data:

- Log entries for End Users
  - For End Users to access log entries
  - A standard API must be provided
- Log entry synchronization between service layers
  - to synchronize log entries between End User, Service Provider and Data Provider Logs
  - Only if shared ledger approach is used

6.1.6 Public Service Directory

Purpose and Responsibilities

Public Service Directory (PSD) contains records of all connected Service Providers' Services and Data Provider's data sources

Requirements

Minimum requirements for Public Service Directory are described below:

- Service Provider must be able to add new Services
- Service Provider must be able to modify existing Services
- Service Provider must be able to delete existing Services.
- Service provider must be able to list all Data Sources providing specific data elements
- Data Provider must be able to add new Data Sources
- Data Provider must be able to modify existing Data Sources
- Data Provider must be able to delete existing Data Sources
- An End User must be able to list all Services and constrain the list based on filters

Sample Functional Flow

1. Data providers register Data Sources.
2. Service Providers build Services that use these Data Sources and possibly their am own data
3. Service Providers register Services
4. End users discover Services
5. End Users subscribe to Services

Restrictions

There can be more than one physical Public Service Directories (End Users Service Directory creates a logical unified view of all services for the End User).

Public Service Directory must contain entries for all Services and all Data Sources.

Service Providers must register their Services in the Public Service Directory with an entry that contains needed information about the Service so End Users can use this information when discovering Services.

Data Providers must register their Data Sources into the Public Service Directory with an entry that contains needed information about the Data Source, so Service Providers can use this information when creating their Services.

Interfaces including Data Streams

Inbound data:

- Services from Service Providers
  - For creating Services
- Data Sources from Data Providers
  - For creating Data Sources

Outbound data:

- Public Service Directory offers a list of Services, so End Users can discover new services in their Personal Service Directory.
- Public Service Directory offers a list of Data Sources, so Service Provider can discover new data sources to be used in their Service.
- Public Service Directory offers a list of Data Sources for a Service so End Users can connect new Data Sources so that more Services would become available.

Technical Standards

There are no technology constraints that limit component implementation - for example, to use a specific programming language.

Quality and Performance
Public Service Directory services need to always be available.

6.1.7 Service Provider Service Directory

Purpose and Responsibilities

Service Provider Service Directory (SPSD) contains records of a Service Provider's Services in more detail.

Requirements

Minimum requirements for Data Source are described below:

- Service Provider must be able to add new Services
- Service Provider must be able to modify existing Services
- Service Provider must be able to delete existing Services
- Service can be started
    - When an End User requests the Service or
    - an End User has greenlit the Service Provider to begin the service based on any combination of following
        - some triggered event,
        - schedule or
        - Service Provider's own service-related process /automated process

Sample Functional Flow

1. Service Provider creates a Service and attaches the necessary metadata descriptions to it
2. Service is automatically listed in Public Service Directory when a Service Provider promotes Service to be a production version

Restrictions

Public Service Directories must contain entries for all Services.

Any change to a Services is automatically conveyed to the Public Service Directory.

6.1.8 Service Provider Consent Directory

Purpose and Responsibilities

Service Provider Consent Directory (SPCD) contains records of all received consent for from all End User using Service Provider's Services
Service Provider Consent Directory contains consents from End Users. There will be two identical version of a Consent – End User's and Service Providers. Part of this

Consent will be further sent to Data Providers which will also store this part of the Consent to their Data Access Control component (Data Providers Consent Directory). There might be a solution where Consents are stored also into trusted 3rd party – either in same format or as has been created from the original Consent. So, both parties have a possibility to proof the content of the Consent.

The Personal Consent Directory contains Consent information for each service. Each Consent defines all Data Access Records which will be used to request data from Data Providers. There will be at least one Data Access Record for each Data Provider from which the Service Provider will request data.

In these consent forms, there will be information for the Service Provider about the Data Providers, but the actual Data Access Record which will be further sent to the Data Providers will be encrypted in a way that only the Data Provider can read it. This is the mechanism for how the Data Provider will trust that the origin for the data request is coming from this exact End User. There isn't any need to check this request online from the End User.

Requirements

Minimum requirements for Consents and Service Provider Consent Directory:

- End Users will create Consents which will be sent to Service Provider and then stored in the Service Provider Consent Directory
- These Consent forms will have at least two parts:
    - Part 1 will be readable only to the Service Provider and contains information about the Data Providers (interfaces for metadata, and data request)
    - Part 2 will be an encrypted message to the Data Provider in a way that only the End User and the Data Provider can understand it. The Service Provider will send this part "blindly" to the Data Provider based on the information in Part 1.
    - There can be multiple Part 2–type of elements, one for each Data Provider

Interfaces including Data Streams

Service Provider will receive Consent from End Users.

The Consent will be divided for several messages, one for each Data Provider. These messages will contain information from the Service Provider to the Data Provider as well as the encrypted message from the End User.

6.1.9 Inbound Data Adapter

Purpose and Responsibilities

Inbound Data Adapter (IDA) is the inbound data transfer point for the Service Provider's Service to receive data from Data Providers. IDA is an interface that isolates incoming data from Service Providers' operational systems (service production).

Requirements

IDA must

- receive incoming data
    - decrypt the data
    - decompose the data
    - deliver the data to the service production

Interfaces including Data Streams

IDA communicates with the Data Providers' Outbound Data Adaptor and Service Providers operational systems (service production).

Incoming data

- encrypted data package
    - metadata
    - identifier
    - data

Outbound data

- decrypted and decomposed data

6.1.10 Service Provider Log

Purpose and Responsibilities

Service Provider Log (SPL) is the Service Provider's internal log that stores all log entries created on the Service Provider Layer. The Service Provider Log contains both public and private sections of the Service Provider's log entries. All changes to services are logged as public entries. Invocations of services including usage of consent forms and data access are logged as private entries. Data itself – whether provided by a Service or Data Providers - is not logged.

Requirements

- The following processes must create a log entry to the Service Provider Log

    - Service changes – for example a new Service is added, or an existing Service is modified or removed from the Service Provider. Service change logs must be public.
    - Service usage – for example, the End User, uses a service. Service usage logs must be private.
    - Data usage – for example a Service Provider uses a Consent to access data for a Data Provider. Data usage logs must be private.
- Service Provider Log must contain all log entries associated with the Service Provider
- Service Provider must provide me logs concerning End Users - i.e., service usage and data usage logs - also for End User Personal Log (since End User Personal Log must contain all information about operations concerning the End User)
    - This leads to the requirement that Personal Log, Service Provider Log and Data Provider Log must be connected to each other in a uniform manner. See requirements for "1.1.5 Personal Log".
- Log entries must be created in standard format containing at least the following information
    - What operation was performed?
    - Which component/system performed the operation?
    - Which component/system received information about the End User?
    - What End User information was handed over?
    - When was the operation performed? (timestamp)
    - Did the operation succeed?
    - Which consent form was used?
- The Service Provider Log must provide standard APIs for creating and retrieving log entries
    - Service Provider Log APIs must be secured on the system level. Access to write log entries must be restricted to authorised systems only. Access to retrieve log entries must be restricted for Service Provider administration only.
- The Service Provider Log must comply with GDPR, so personal information - like identifiers, personal information and credentials – must not be logged.

Sample Functional Flow

See "1.1.5 Personal Log - Sample Functional Flow".

Restrictions

- Service Provider Log is a storage for log entries. It provides APIs for creating and retrieving log entries but does not provide a user interface. A user interface may be built separately.

Interfaces including Data Streams

Inbound data:

- Log entries from Service Provider Layer components
    - for creating log entries in the component's database/ledger
    - A standard API must be provided

Outbound data:

- Log entries for Service Provider administration
    - for administration to access log entries
    - A standard API must be provided
- Log entry synchronization between service layers
    - to synchronize log entries between End User, Service Provider and Data Provider Logs
    - Only if a shared ledger approach is used

Technical standards

See "1.1.5 Personal Log - Technical Standards".

6.1.11 Data Source

Purpose and Responsibilities

Data Source (DS) is a Data Provider's detailed description of its outbound interface Data Providers can register their Data Sources in the Public Service Directory. Data Source Description contains both technical and human-readable documentation of the data source.

Requirements

Minimum requirements for Data Source are described below:

- Data Provider must be able to add new Data Sources
- Data Provider must be able to modify existing Data Sources
- Data Provider must be able to delete existing Data Sources

Sample Functional Flow

1. Data Provider creates a Data Source and attaches needed metadata descriptions to it
2. Data Source is automatically listed in the Public Service Directory when Data Provider sets it to be so.

Public Service Directory must contain entries for all Data Sources

Any change to a Data Source is automatically conveyed to Public Service Directory

Interfaces including Data Streams

Outbound data:

- Data Sources from Data Provider to Public Service Directory

6.1.12 Outbound Data Adapter

Purpose and Responsibilities:

Outbound Data Adapter (ODA) is the transfer point for Data Providers to send data to a Service Provider. Outbound Data Adapter is an interface that separates outgoing data from Service Providers' operational systems, i.e., the actual data sources.

Requirements

ODA must

- receive the request from the Data Access Control
- choose the appropriate data transfer mechanism
- send the data to Service Providers Inbound Data Adaptor
- verify the delivery of the data

Interfaces

ODA communicates with the Data Access Control and the Service Providers Inbound Data adaptor.

Incoming

- decrypted data package

Outbound

- decrypted data package

Technical standards

API's provided by Data Access Control should be RESTful. Data should be structured using JSON or XML. All communication between decentralized components should be secured using HTTPS-connections.

Other than mentioned, there are no technology constraints that limit component implementation - for example, to use a specific programming language.

6.1.13 Data Access Control

Purpose and Responsibilities

Data Access Control (DAC) is a component that orchestrates the process of receiving data requests, identifying individuals and associated data, accessing the data and delivering it to Service Provider(s).

Interfaces including Data Streams

DAC communicates with Service Provider's Data Request, Data Provider's Outbound Data Adapter and internal data sources.

Technical standards

There are no technology constraints that limit component implementation - for example, to use a specific programming language.

6.1.14 Data Provider Log

Purpose and Responsibilities

Data Provider Log (DPL) is the Data Provider's internal log that stores all log entries created on the Data Provider Layer. Like the Service Provider Log, the Data Provider Log also contains both public and private sections of log entries. All changes to data sources are logged as public entries. Access to data are logged as private entries. Data provider contents and data itself – whether provided by a Service Provider in data service invocation or Data Providers - is not logged.

Requirements

- The following processes must create a log entry for a Data Provider Log
  - Data source changes – for example a new Data Source is added, or an existing Data Source is modified or removed from the Data Provider. Data Source change logs must be public.
  - Data access – for example a Service Provider uses a Consent to access data from a Data Provider. Data usage logs must be private.
- Data Provider Log must contain all log entries associated with the Data Provider

- Data Provider must provide logs concerning End Users - i.e., data access logs - also for End User Personal Log (since End User Personal Log must contain all information about operations concerning the End User)
  - This leads to the requirement that Personal Log, Service Provider Log and Data Provider Log must be connected to each other in a standard way. See requirements for "1.1.5 Personal Log".
- Log entries must be created in standard format containing at least the following information
  - What operation was performed?
  - Which component/system performed the operation?
  - Which component/system received information about the End User?
  - What End User information was handed over?
  - When was the operation performed? (timestamp)
  - Did the operation succeed?
  - Which consent was used?
- Data Provider Log must provide standard APIs for creating and retrieving log entries
  - Data Provider Log APIs must be secured on a system level. Access to writing log entries must be restricted to authorised systems only.

Access to retrieve log entries must be restricted for Data Provider administration only.

- Data Provider Logs must comply with GDPR, so personal information - like identifiers, personal information and credentials – must not be logged.

Sample Functional Flow

See “1.1.5 Personal Log - Sample Functional Flow”.

Restrictions

- Data Provider Log is a storage for log entries. It provides APIs for creating and retrieving log entries but does not provide a user interface. A user interface may be built separately.

Interfaces including Data Streams

Inbound data:

- Log entries from Data Provider Layer components
  - for creating log entries in the component’s database/ledger
  - A standard API must be provided

Outbound data:

- Log entries for Data Provider administration
  - for administration to access log entries
  - A standard API must be provided
- Log entry synchronization between service layers
  - to synchronize log entries between End User, Service Provider and Data Provider Logs
  - Only if a shared ledger approach is used

Technical standards

See “1.1.5 Personal Log - Technical Standards”.