This document contains detailed functional requirements for all IHAN ecosystem functional components in End User, Service Provider and Data Provider levels. It also contains all non-functional requirements for IHAN ecosystem. This document can be used as a cookbook for projects either implementing new IHAN functional components or utilizing services provided by IHAN ecosystem to create new services

IHAN Blueprint

2.0 v261018

Antti Larsio, Juhani Luoma-Kyyny, Jyrki Suokas and Teemu Karvonen

Table of Contents

1 Introduction and Goals

Data economy is the fastest growing part of 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 program and a dataset, you can use the same program and dataset without my ability to use them being diminished.*
>
> *As a consequence 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.")*

Different B2C businesses, such as commerce with loyalty cards, health providers, insurance and banking, have accumulated a wealth of consumer data associated to individual customers. Another rising trend is where sensors and other connected devices are gathering data – currently quite often so that the device manufacturers and other service providers collect lots of personal data - related for example to health, home surveillance and vehicle usage - that can be associated to an individual user. All this leads to a situation where there are massive amounts of data about an individual – but data is somewhere out there.

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:

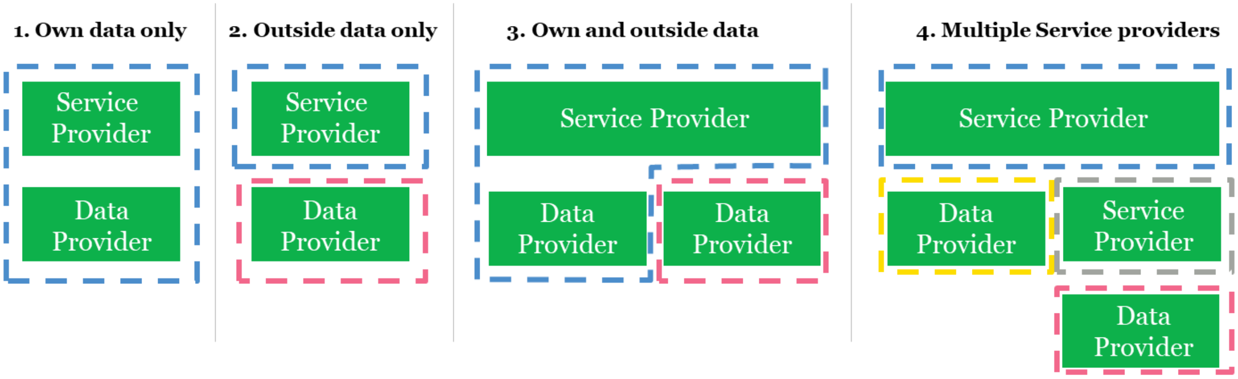
- **Format:** *"structured, commonly used and machine-readable format"* needs to be unambiguously and explicitly defined for personal data across all industries so automated data interchange is possible
- **Governance:** *"the data subject has given consent to the processing of his or her personal data for one or more specific purposes"*. As individuals currently do not have a way to manage their consents across data processors, the processes and tools need to be defined.

- **Method:** *"… personal data transmitted directly from one controller to another, where technically feasible."* Current regulation gives data controllers a maximum of two months to transfer the data to the requested party. The data transfer window being so long, it doesn't encourage automatic data transfer between parties - a big problem by itself. In addition, protocols and use-case specific standards for real-time transfer of personal information between systems do not exist and need to be created.

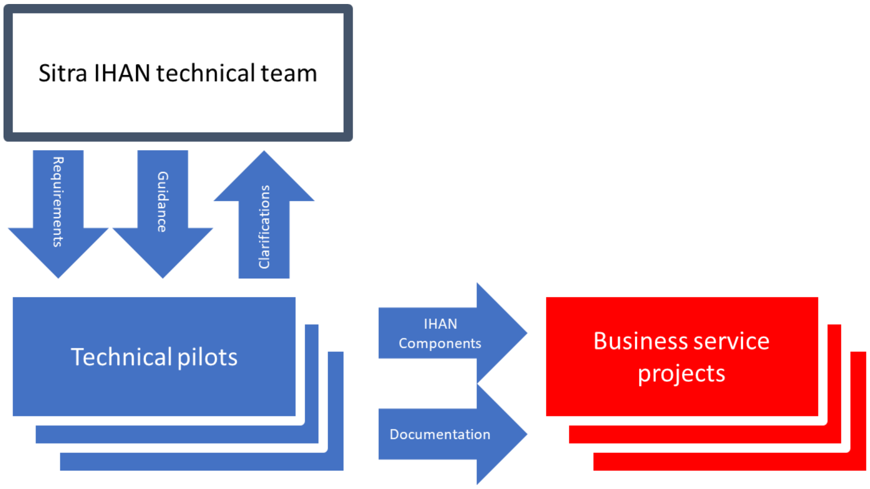
Sitra – the Finnish Innovation Fund - has started a project called IHAN that intends to build a governance framework, architectural definitions and requirements for essential components to build a data-driven world. A world where data flows in real-time in a seamless but secure fashion enabling new services 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-used-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 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.



At this stage in our project Sitra IHAN technical team concentrates on describing IHAN internal services and core components for other parties to implement. Results of this work is this document – IHAN Blueprint - describing functional and non-functional business and technical requirements to use in technical pilot projects. The technical pilots are all about creating as-generic-as-possible technical components that implement the IHAN ideology in practice. After the mandatory set of these well-interworking components have been developed, business service projects can start utilizing them and a functional ecosystem can be built:



After the initial creation phase a permanent community-based governance model will be put into place.

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 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 reference architecture collection from various business projects, will form the basic tool kit for implementing fair data ecosystem solutions.

2 System Scope and Context

2.1 Business Context

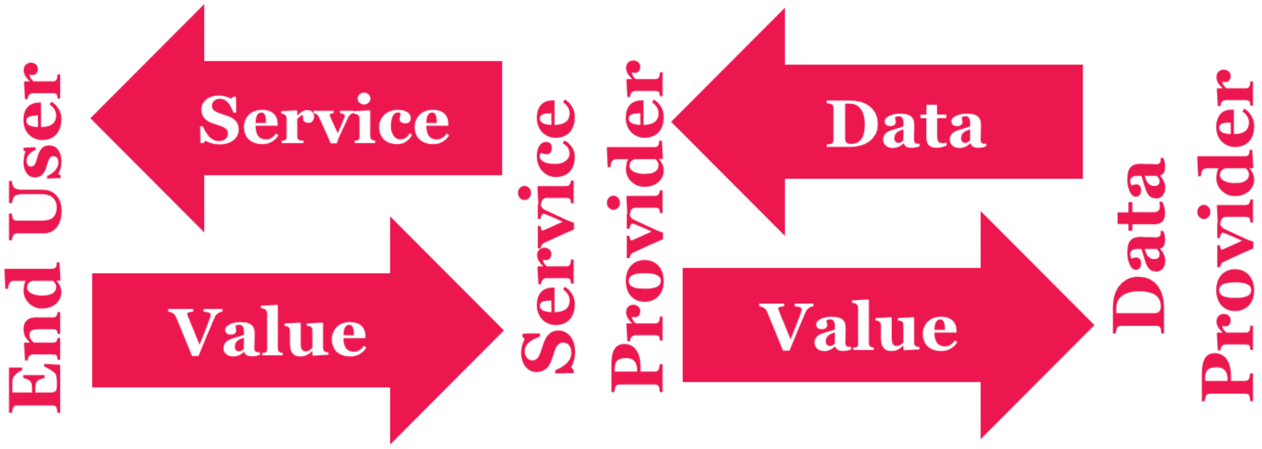
Current data economy was gravitating towards a world where the rights of individuals where overrun by business and revenue increasing business models:

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

Both GAFA's and BAT's 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

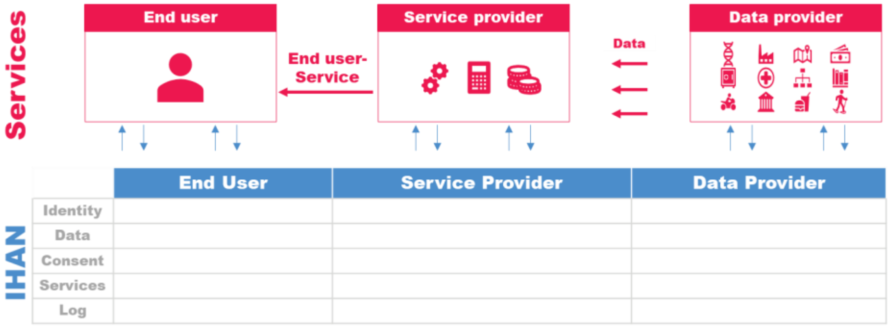
At the same time data economy related regulations like PSD2 where banks are not just forced to build expensive API's opening access to accounts and transactions but are also forced to let external parties to initiate payments at no cost. The good intention to open up possibilities for new players to start offering new and more innovative services that banks have been able to offer is shadowed by the unfortunate fact that banks are making a halfhearted attempt to just comply with the regulation instead of embracing to be 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 Service Providers must 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 that both sides of value exchange transparently consider to be fair:

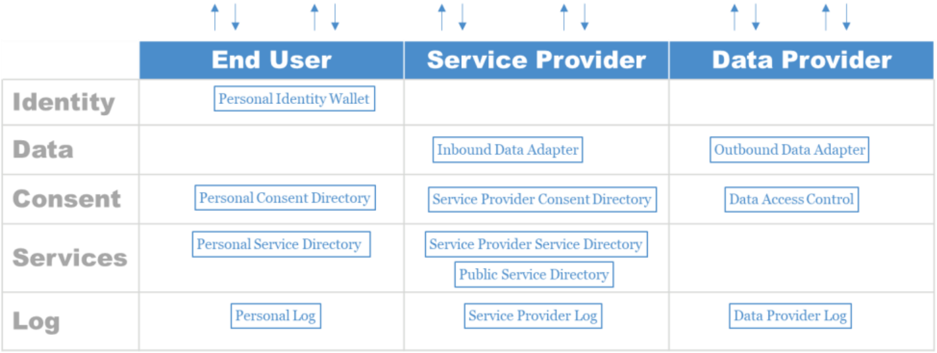


2.2 System Scope

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.

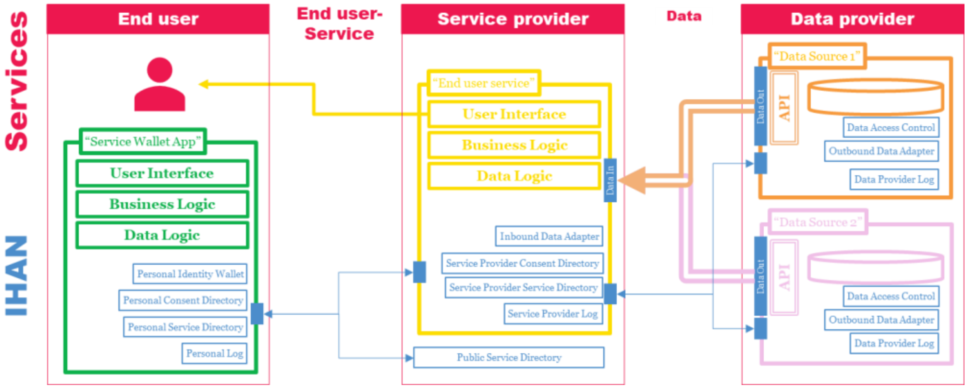


This also means that IHAN components do not have built-in user interfaces but provide services so that user interfaces can be built. We have identified following components that are the initial components for the ecosystem. New components can be added, and old ones modified – even discarded if needed.



	End User	Service Provider	Data Provider
Identity	Personal Identity Wallet		
Data		Inbound Data Adapter	Outbound Data Adapter
Consent	Personal Consent Directory	Service Provider Consent Directory	Data Access Control
Services	Personal Service Directory	Service Provider Service Directory Public Service Directory	
Log	Personal Log	Service Provider Log	Data Provider Log

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):



2.3 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 similar enough both functionally and technically.
- All three functionality levels should have a standard way to support metadata exchange, consent management and 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 seen as 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 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.

3.1 Key Concepts

The key concepts of IHAN are presented below.

End User Represents the individual or member of an organization to whom Services are created for

Service Provider An organization that provides Services to End Users and other Service Providers.

Data Provider An organization that provides data for Service Providers and/or End Users

Service A Service is what Service Providers deliver to End Users or other Service Providers.

Consent For a Service Provider to be able to provide Services, the Service Provider and End User enter into an agreement between each other – this agreement is the Consent.

Consent is one of the three IHAN base components (the other two being IHAN Identifier and Logging) and it is always connected to an IHAN Identifier. 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.

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 might be verified by a third party. These digital representations are called as **Identifiers**. These identifiers are used to form IHAN Identifiers.

It should be noted that an IHAN solution itself does not act as an identity management solution.

Attributes

Related to a digital identity or a group of identities, there can be Attributes defining certain features on a digital identity. Attributes might be verified by a third party.

IHAN Identifier

This is the fundamental component of IHAN functionality. This universally unique number is a combination of identities of a person (or an individual) and a data set. So, the IHAN Identifier identifies a connection between a person and a data entity. A Consent is linked to an IHAN Identifier and enables the interchange of data between Service and Data Providers.

Data Access Record

A record containing End User access credentials used to request data from Data Providers using a specific End User identity.

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.

Metadata and semantic interoperability

Semantic interoperability with using several data sources is on Service Provider’s responsibility. To have this semantic interoperability Data Providers must create ways to get the metadata of the data they have. In these definitions, there might be further links to other definitions like codes, definitions in detail, vocabularies, nomenclatures, document structures, used standards etc. Creating these definitions are outside of the scope of the IHAN project.

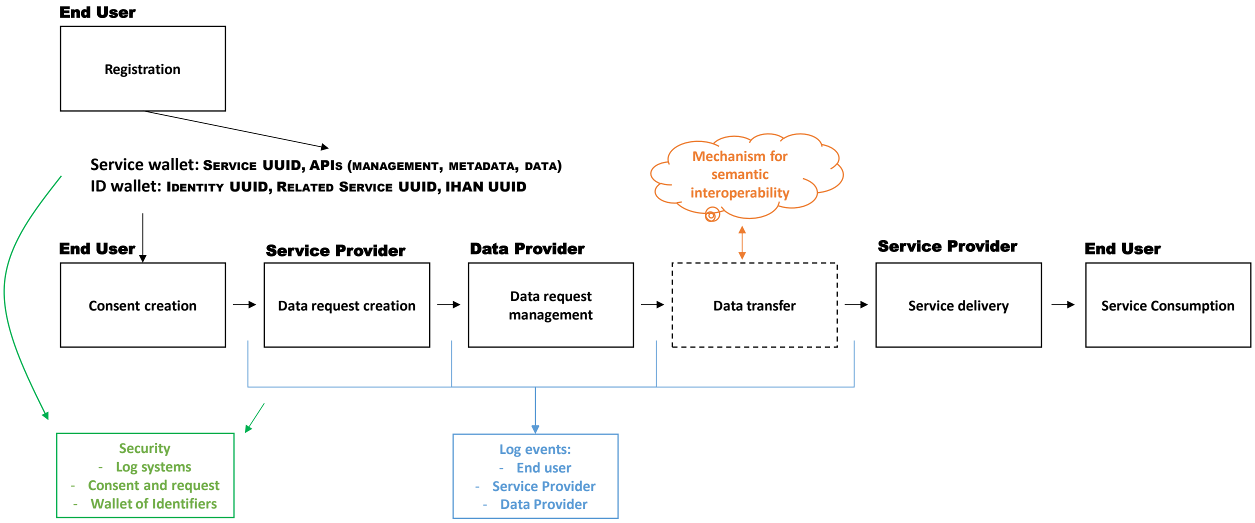
3.2 Functional Flow Between IHAN Components

So that Service Providers and Data Providers could fully concentrate on their own product development, IHAN project is building a set of reusable service components. These components provide basic functionalities and support services for real-life business application (created by Service and Data Providers). The main flow is quite simple:

1. End User completes registration - thus creating a Personal Service Wallet - and links an Identity to the Wallet
2. End User discovers a Service he/she wants to use and adds it to his/her Personal Service Directory
3. End user grants a Consent to the Service Provider to access data from one or more Data Providers

4. When service is invoked, the Service Provider uses the Consent to access data at Data Provider. Service Provider uses the data to create the Service for End User

5. All needed actions are logged immutably



4 Requirements Overview

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

1. **Setup** functionality – related to creation and updates of main 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

Main functionalities of the End user level are related to Identities and Services.

4.1.1 Setup Functionality

End Users are able to:

- create new **Personal Service Wallets**
- delete existing Personal Service Wallets
- Modify existing Personal Service Wallet – for example restrain access to for a set period of time (in case of - for example – if a device gets stolen). Full list of possible modification activities will be defined later.
- 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 start 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 a Consent to use data located in external sources (Data Providers). End User creates a Consent to a Service Provider and specifies details for this data usage. A Service Provider will then use the Consent to get data from the specified Data Provider.

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

All data accesses, data transfers and other relevant data actions will be created 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 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. Data Sources list can contain both mandatory and optional data elements and it is 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. Best analogues to 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, 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 taken into use. Service Providers can also promote Services for End Users that have opted in for the category of Services the Service Provider is offering. If there is a match between the data elements that 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 End User modifies or revokes a Consent, then this information is automatically passed 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 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(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, 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 private section of **Service Provider Log.** Part of the log can be used for Value Exchange information collection – billing itemization.

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 market place 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 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 be created a log entry to the public section of **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).

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 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, the **Data Access Control** on Data Provider 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 be created a log entry in **End User Personal Log**, private section of **Data Provider Log** and 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 itemization.

5 Solution Strategy

5.1 Quality Goals

There are three main quality goals for the IHAN ecosystem. Measurements will be added later.

1. Firstly, the ecosystem must allow for **value exchange**. Service Providers must be compensated for the creation of the Services and Data Providers must be compensated for storing data and making that data available. Value can be money or any other form that both sides of value exchange transparently consider to be fair. This goal will be measured by the amount of service providers and data providers joining the ecosystem and by the value exchanged between the parties

2. Secondly the ecosystem must remain **distributed** and contains no design decisions that create centralized solutions.

3. Thirdly the ecosystem must be **secure** as it is handling personal data which is governed by GDPR and other acts.

5.2 Architecture Constraints

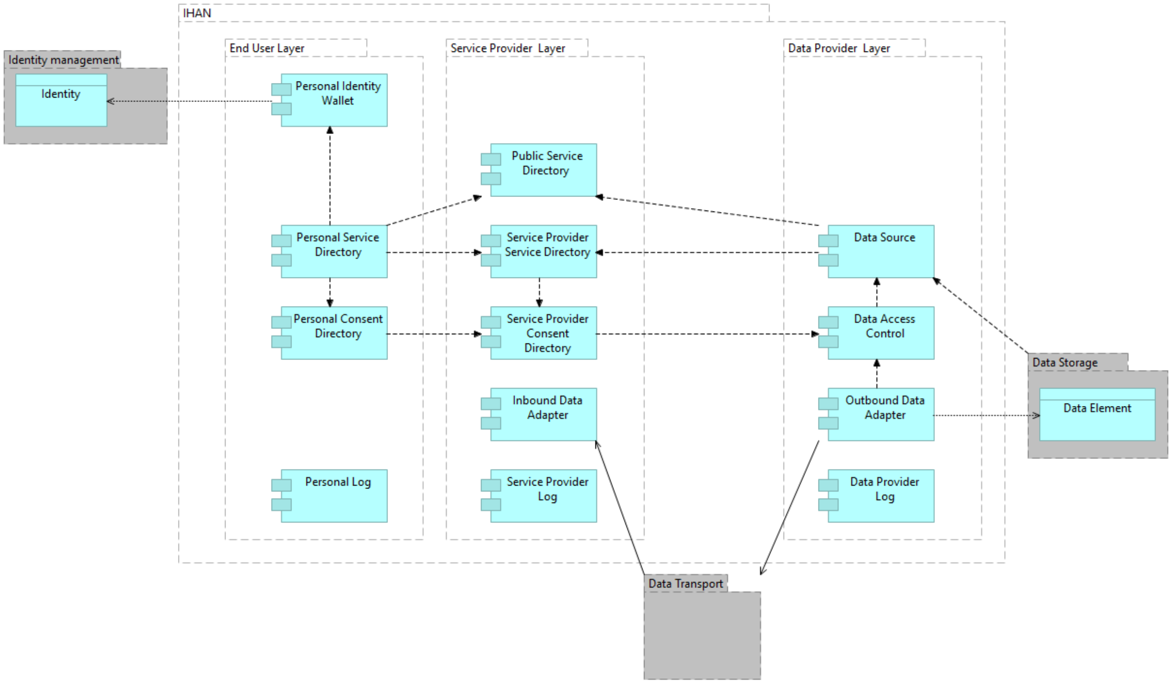
Any decision that takes the ecosystem away from the quality goals – fair value exchange, distributed instead of centralized and secure handling of personal data – must be avoided at all costs.

Here are some of the basic architecture principles concerning IHAN

- Architecture must support implementation of services that enable" 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
- Solutions must not be overlapping
- Reusability is recommended
- Solutions should be inter-operational
- Application design must be user centric and ease-of-use based
- All solutions must 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 management of individual's multiple virtual identities
- Architecture must support logging, auditing, trust
- Architecture must enable secure data transfer, management and storing

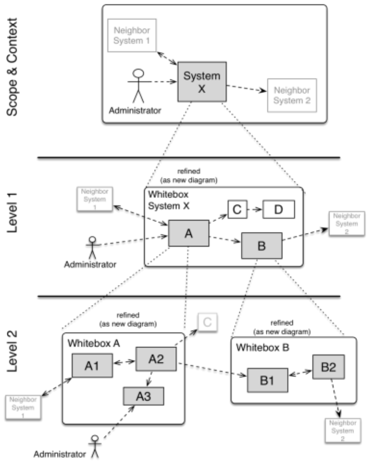
6 Building Block View

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.



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 first Technical Pilot Projects start 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 there-in
- 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.

Open issues/problems/risks

- Restore mechanism and needed functionality. What constitutes a "lost wallet"?

6.1.2 Personal Identity Wallet

Purpose and Responsibilities

Personal Identity Wallet (PIW) is a component for storing Identity Records and Data Access Records, 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

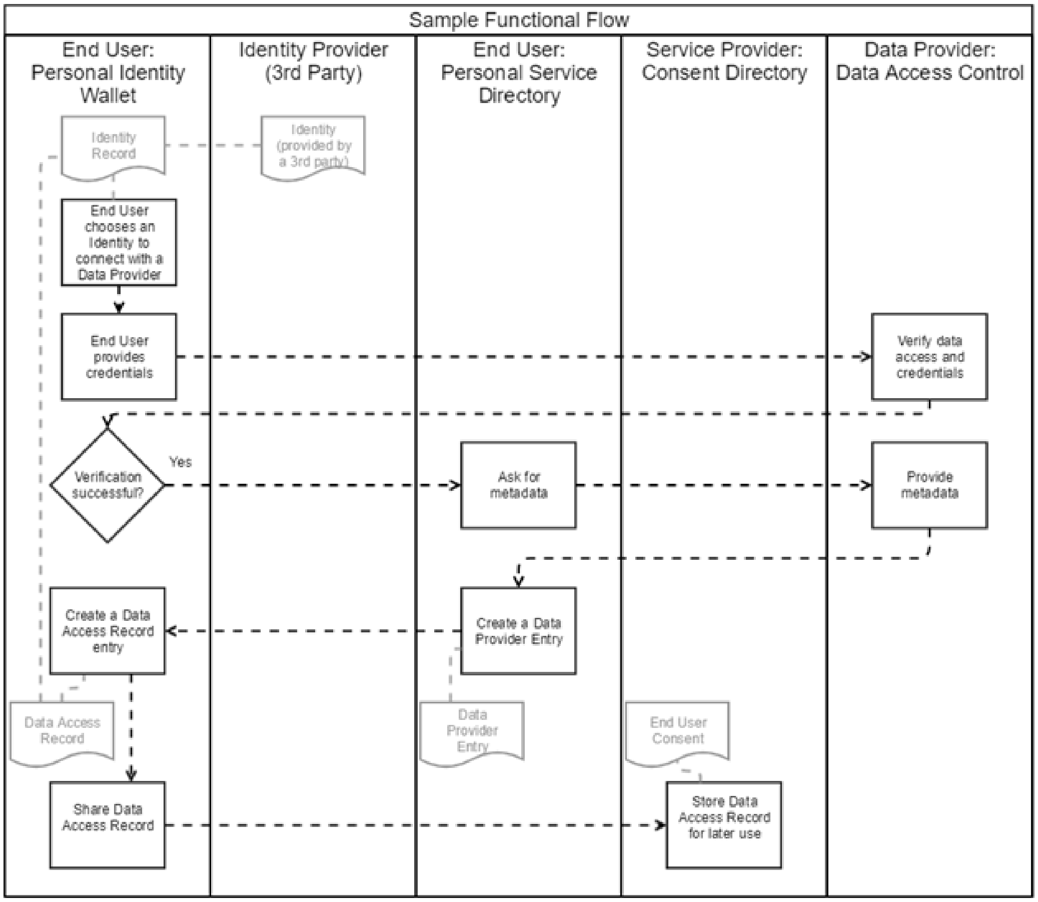
Minimum requirements for Personal Identity Wallet are described below:

- 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 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:



Sample Functional Flow
End User: Personal Identity Wallet
Identity Provider (3rd Party)
End User: Personal Service Directory
Service Provider: Consent Directory
Data Provider: Data Access Control
Identity Record
Identity (provided by a 3rd party)
End User chooses an Identity to connect with a Data Provider
End User provides credentials
Verify data access and credentials
Verification successful?
Yes
Ask for metadata
Provide metadata
Create a Data Access Record entry
Create a Data Provider Entry
Data Access Record
Data Provider Entry
End User Consent
Share Data Access Record
Store Data Access Record for later use

.

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 Personal Service Directory is populated with metadata information provided by 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 using 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 other 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 distributed 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.

Quality and Performance

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

Open issues/problems/risks

- The role of the IHAN identifier regarding data exchange and Data Access Records must be defined in more detail
- Should UI implementation for Person Identity Wallet be considered?
- Processes between components and component responsibilities in functional flow should be described in more detail

6.1.3 Personal Service Directory

Purpose and Responsibilities

Personal Service Directory (PSD) manages End User's Services. Personal Service Directory contains descriptions of all Service Provider Services that End User has subscribed to. Personal Service Directory is used to grant service providers access to data providers so service providers can produce the service for the End User. As there can be more than one physical Public Service Directories, 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 start forming into a GDPR dashboard, showing different service End User has access to and what data is behind each identity.

Requirements

Minimum requirements for Personal Service Directory are described below:

- End User must be able to list all Services and constrain the list based on filters in Personal Service Directory
- End User must be able to add new Services from Public Service Directory. If End User is willing to start using a new service, the Personal Service Directory uses Personal Consent Directory to automatically grant the needed Consents for the Service Provider that are required to access data from all needed Data Sources. This information is stored in Service Provider's Consent Directory. Service is also stored as activated in End User's Personal Service Directory and Consents are linked to it.
- End User must be able to modify existing Services
- End User must be able to delete (unsubscribe) existing Services. If a Service is deleted, the corresponding Consent's validity must be terminated.
- Changes to Service Provider Services need to be automatically updated in all End Users Personal Service Directories

- Personal Service Directory could also actively propose new services through "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 "Data Sources Missing" Service discovery process available for the End User based on the Data Sources that End User does not have Personal Identity Wallet, but these Data Sources are common to the user profile End User has. For example, even if 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.
- End User could rank a Service and this information could be stored in Public Service Directory

Sample Functional Flow

1. Personal Service Catalog contains a record of all current and past Services of the End User.
2. To find new ones End user browses the Services in his Personal Service Directory and sees them in "Available new services" section.
3. When user finds a Service that he wants to start 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 a Consent to use it. Consent is stored in End User's Personal Consent Directory.
4. Service Provider is provided with Consents containing all needed 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:

- Consents to Personal Consent Directory
  - Personal Service Directory uses Data Source information to create needed Consents

6.1.4 Personal Consent Directory

Purpose and Responsibilities

Personal Consent Directory (PCD) stores all End User's Consents given 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 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 Service Provider will ask data.

In these Consents 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 how the Data Provider will trust that the origin for the data request is coming of this exact End User. There isn't any need to check this request online from the End User.

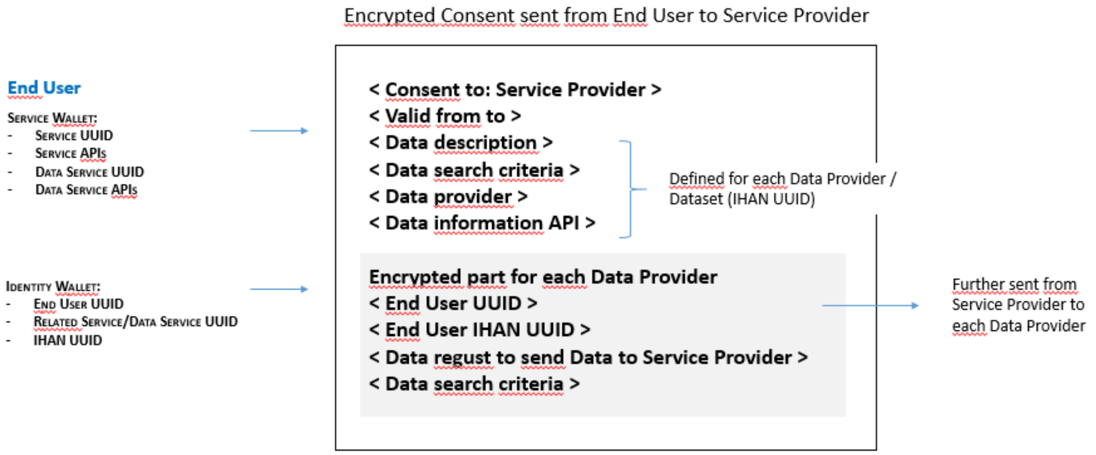
Requirements

Minimum requirements for Consents and Personal Consent Directory:

- End User will create Consents which must be stored in the Personal Consent Directory
- These Consents 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. 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

Sample Functional Flow

CONSENT CREATION



Encrypted Consent sent from End User to Service Provider

End User

Service Wallet:
- Service UUID
- Service APIs
- Data Service UUID
- Data Service APIs

Identity Wallet:
- End User UUID
- Related Service/Data Service UUID
- IHAN UUID

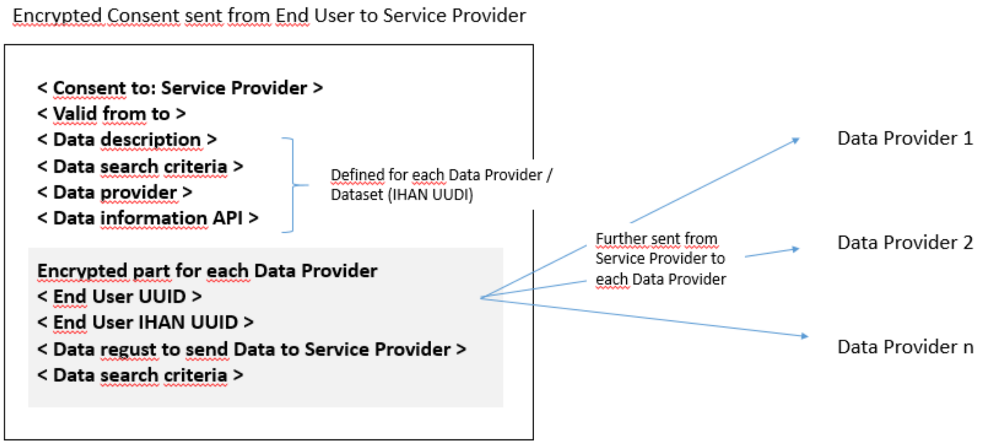
< Consent to: Service Provider >
< Valid from to >
< Data description >
< Data search criteria >
< Data provider >
< Data information API >

Defined for each Data Provider / Dataset (IHAN UUID)

Encrypted part for each Data Provider
< End User UUID >
< End User IHAN UUID >
< Data regust to send Data to Service Provider >
< Data search criteria >

Further sent from Service Provider to each Data Provider

DATA REQUEST MANAGEMENT



Encrypted Consent sent from End User to Service Provider

< Consent to: Service Provider >
< Valid from to >
< Data description >
< Data search criteria >
< Data provider >
< Data information API >

Defined for each Data Provider / Dataset (IHAN UUDI)

Encrypted part for each Data Provider
< End User UUID >
< End User IHAN UUID >
< Data regust to send Data to Service Provider >
< Data search criteria >

Further sent from Service Provider to each Data Provider

Data Provider 1

Data Provider 2

Data Provider n

Restrictions

The encryption/decryption mechanism as well the needed Secured Key Exchange mechanism are outside of the scope of this IHAN Blueprint. However, these mechanisms are mandatory for each implementation.

Interfaces including Data Streams

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

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

Open issues/problems/risks

- Consent structure
- Security mechanisms (encryption/decryption, key exchange)

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.

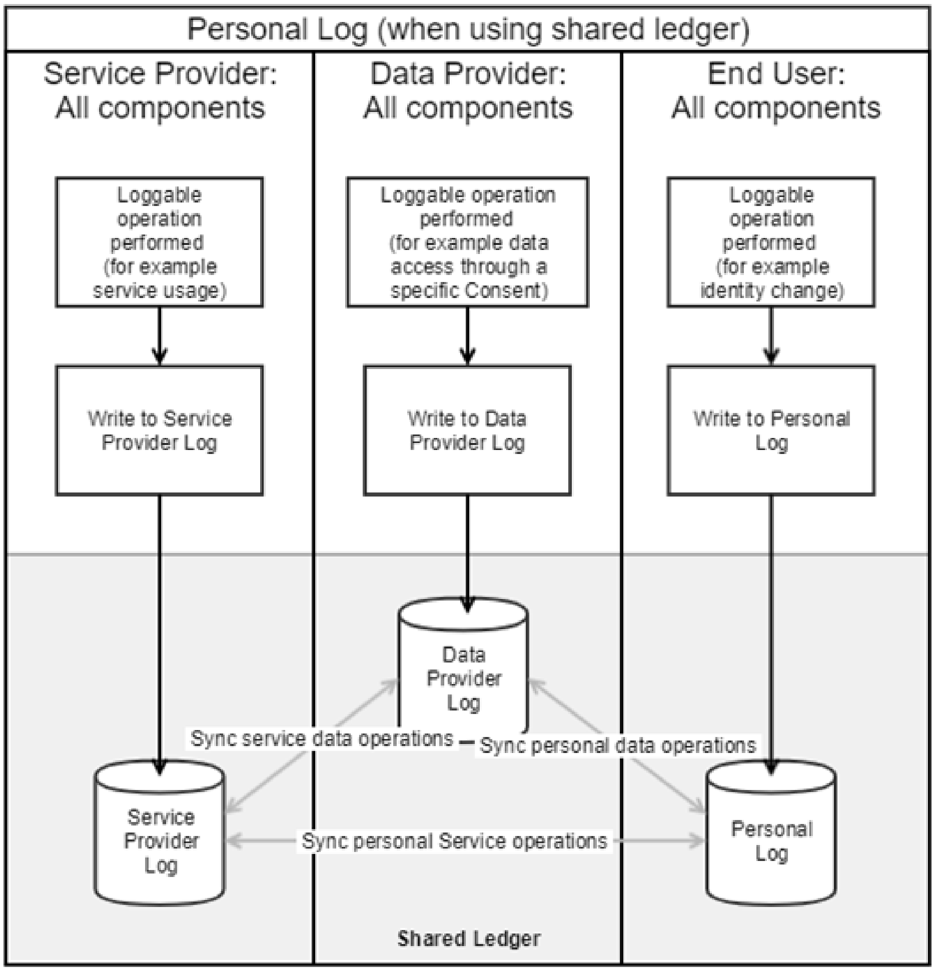
Requirements

- The following processes must create a log entry to 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 to access data for a Data Provider
- 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 other than personal log entries, 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 personal and system level. Access to write log entries must be restricted to authorized systems only. Access to retrieve Personal Log entries must be restricted for End User only.
- 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 Personal Service Wallet.

Sample Functional Flow

Sample functional flow is presented below:


Personal Log (when using shared ledger)
Service Provider: All components
Data Provider: All components
End User: All components
Loggable operation performed (for example service usage)
Loggable operation performed (for example data access through a specific Consent)
Loggable operation performed (for example identity change)
Write to Service Provider Log
Write to Data Provider Log
Write to Personal Log
Data Provider Log
Sync service data operations
Sync personal data operations
Service Provider Log
Sync personal Service operations
Personal Log
Shared Ledger

It should be noted that the diagram above illustrates the functionalities of logging when a shared ledger system is used. Another option is to provide access to logging APIs across layers and components.

Restrictions

- 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 User
    - For End User 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

Technical standards

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

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.

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

Open issues/problems/risks

- API vs. Shared ledger approach must be discussed further
    - The approach defines how to implement APIs and interoperability between logs on different layers

  - There must be a way to control the number of logs created and stored in each layer - which depends on the selected approach

6.1.6 Public Service Directory

Purpose and Responsibilities

Public Service Directory (PUSD) 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
- 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 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

Any change to a Service is automatically conveyed to Personal Service Directory of those End Users that are subscribing to the service

Any change to a Data Source is automatically conveyed to those Services that have it subscribed so regression testing need due to this change be assessed by the Service Providers

Service Providers must register their Services into 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
    - A standard API must be provided *(TODO: add this to requirements)*
- Data Sources from Data Providers
    - For creating Data Sources
    - A standard API must be provided *(TODO: add this to requirements)*

Outbound data:

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

Technical Standards

API's provided by Public Service Directory should be RESTful. Data should be structured using JSON or XML. All communication between distributed 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.

Quality and Performance

Public Service Directory services need to always available

Open issues/problems/risks

- Service Provider / Data Provider identity needs to be managed

- Can we have a world where anybody can freely create a Public Service Directory or does adding or removing one needs to be approved by some kind IHAN governing body?

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 Service
- Service Provider must be able to modify existing Service
- Service Provider must be able to delete existing Service
- Service can be started
  - When End User requests the Service or
  - End User has granted the Service Provider to start 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 needed metadata descriptions to it

2. Service is automatically listed in Public Service Directory when Service Provider promotes Service to be a production version

Restrictions

Public Service Directory must contain entries for all Services

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

Open issues/problems/risks

How to handle production, pilot and development instances of Services

6.1.8 Service Provider Consent Directory

Purpose and Responsibilities

Service Provider Consent Directory (SPCD) contains records of all received Consents 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 a has created from the original Consent. So, both parties have a possibility to proof the content of the Consent.

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 Service Provider will ask data.

In these Consents 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 how the Data Provider will trust that the origin for the data request is coming of this exact End User. There isn't any need to check this request online from the End User.

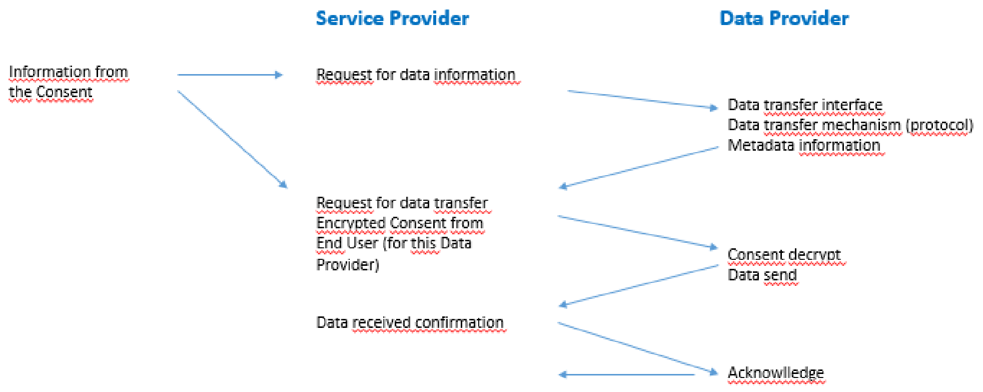
Requirements

Minimum requirements for Consents and Service Provider Consent Directory:

- End User will create Consents which will be sent to Service Provider and then stored in the Service Provider Consent Directory
- These Consents 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 encrypted message to the Data Provider in a way that only the End User and the Data Provider can understand it. Service Provider will send this part "blindly" to the Data Provider based on the information in Part 1.
    - These can be multiple Part 2 –type of elements, one for each Data Provider

Sample Functional Flow

DATA TRANSFER


Service Provider
Data Provider
Information from the Consent
Request for data information
Data transfer interface
Data transfer mechanism (protocol)
Metadata information
Request for data transfer
Encrypted Consent from End User (for this Data Provider)
Consent decrypt
Data send
Data received confirmation
Acknowlledge

Interfaces including Data Streams

Service Provider will receive Consents 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.

Open issues/problems/risks

- Consent structure
- APIs
- Security mechanisms

6.1.9 Inbound Data Adapter

Purpose and Responsibilities

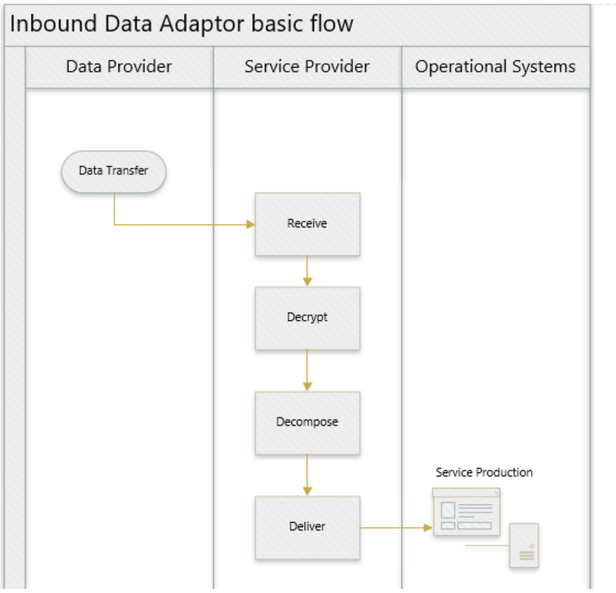
Inbound Data Adapter (IDA) is the inbound data transfer point for Service Provider’s Service to receive data from Data Providers. IDA is an interface that isolates incoming data from Service Providers operational systems (service production).

Requirements

IDA must

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

Sample Functional Flow



Restrictions

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

Technical standards

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

Quality and Performance

Open issues/problems/risks

- is the decomposition of the data (interpretation according to metadata) in scope of this component or the receiving service production component?

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 Service Provider Layer. Service Provider Log contains both public and private sections of Service Provider's log entries. All changes to services are logged as public entries. Invocations of services including usage of Consents 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 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 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 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?
- Service Provider Log must provide standard APIs for creating and retrieving log entries
  - Service Provider Log APIs must be secured on system level. Access to write log entries must be restricted to authorized systems only. Access to retrieve log entries must be restricted for Service Provider administration only.
- 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 shared ledger approach is used

Technical standards

See “1.1.5 Personal Log - Technical Standards”.

Open issues/problems/risks

See “1.1.5 Personal Log - Open issues/problems/risks”.

6.1.11 Data Source

Purpose and Responsibilities

Data Source (DS) is Data Provider's detailed description of its outbound interface

Data Providers can register their Data Sources on 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 Source
- Data Provider must be able to modify existing Data Source
- Data Provider must be able to delete existing Data Source

Sample Functional Flow

1. Data Provider creates a Data Source and attaches needed metadata descriptions to it

2. Data Source is automatically listed in Public Service Directory when Data Provider promotes Data Source to be a production version

Restrictions

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

Open issues/problems/risks

How to handle production, pilot and development instances of Data Sources

6.1.12 Outbound Data Adapter

Purpose and Responsibilities:

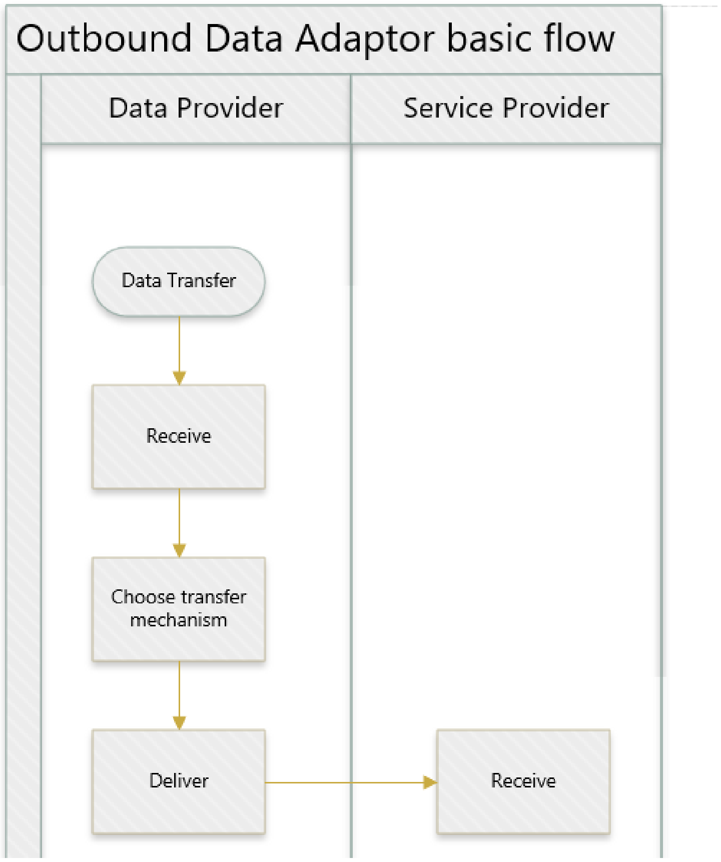
Outbound Data Adapter (ODA) is the transfer point for Data Providers to send data to Service Provider. Outbound Data Adapter is an interface that separates outgoing data from Service Providers operational systems i.e. the actual data sources.

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

Sample Functional Flow



Restrictions

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 distributed 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.

Quality and Performance

Open issues/problems/risks

- should the verification be described here or is it part of the infrastructure or delivery mechanism?

6.1.13 Data Access Control

Purpose and Responsibilities

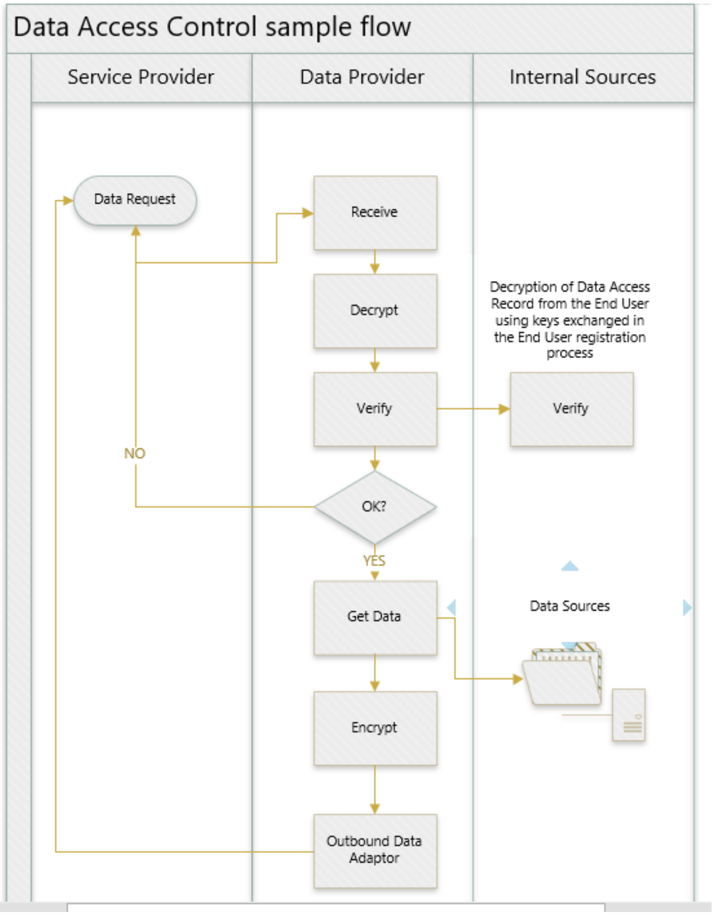
Data Access Control (DAC) is a component that orchestrates the process of receiving data requests, identifying individuals and associated data, accessing the data and delivering it to Service Provider(s).

Requirements

DAC must

- receive the Consent
- decrypt the Consent
- verify the Consent
- accesses end user’s data identified in the Consent
- encrypt the data
- deliver the data to the Outbound Data Adapter

Sample Functional Flow


Data Access Control sample flow
Service Provider
Data Provider
Internal Sources
Data Request
Receive
Decrypt
Decryption of Data Access Record from the End User using keys exchanged in the End User registration process
Verify
Verify
NO
OK?
YES
Get Data
Data Sources
Encrypt
Outbound Data Adaptor

Restrictions

Interfaces including Data Streams

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

Technical standards

API's provided by Data Access Control should be RESTful. Data should be structured using JSON or XML. All communication between distributed 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.

Quality and Performance

Open issues/problems/risks

- Does the DAC communicate directly with the actual data source?

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 Data Provider Layer. Like Service Provider Log, also Data Provider Log 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 to 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 system level. Access to write log entries must be restricted to authorized systems only. Access to retrieve log entries must be restricted for Data Provider administration only.
- Data 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

- 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 shared ledger approach is used

Technical standards

See "1.1.5 Personal Log - Technical Standards".

Open issues/problems/risks

See "1.1.5 Personal Log - Open issues/problems/risks".

7 Runtime View

This section is intentionally blank

8 Deployment View

This section is intentionally blank

9 Governance and External Stakeholders

As the IHAN project at Sitra is time bound we are already now at the beginning setting up the permanent governance structure to ensure seamless transition from project mode to steady state mode. The initial documentation is created by IHAN project itself, but as technical pilot projects start to create components they will elaborate on the documentation of specific components. To ensure a well working ecosystem through interworking components the IHAN Blueprint itself – document containing all documentation - and any other IHAN related documentation is to be governed by following workgroup structure:

BSG IHAN IHAN Business Models		
TSG ISA IHAN Services & System Aspects	**TSG ICSI** IHAN Core System & Internetworking	**TSG IAM** IHAN Access Mechanism
ISA WG1 Services and features	**ICSI WG1** Technical specifications per component	**IAM WG1** Protocols
ISA WG2 Architecture including component definitions as a part of the architecture	**ICSI WG2** Interworking with external systems	**IAM WG2** Smart contracts
ISA WG3 Privacy and Security	**ICSI WG3** Data transport and routing	**IAM WG3** Performance and Conformance aspects and testing
ISA WG4 Maintenance and Billing	**ICSI WG4** Identity management	

Each Working Group can create specific sub-working groups to work on more detailed items that report to the parent working group. Note: Not all working groups are established in the beginning and currently the only one up and running is ISA WG1, where IHAN Tech Team is writing this document.

9.1 IHAN Business Steering Group

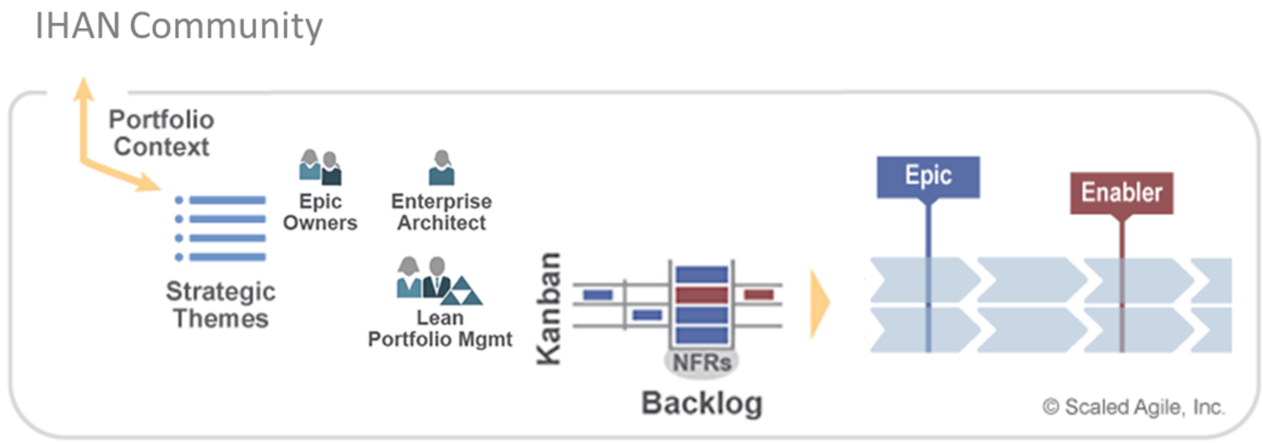
IHAN Business Steering Group is the overall governing body of IHAN Blueprint stating all business requirements. There are no working groups as of this moment under it, but these can be formed and dissolved later

IHAN BSG Business Models	
Terms of reference • BSG Business Models is the overall governing body of IHAN Blueprint making decisions about document structure	**Scope** • Business models and earning log for IHAN ecosystem
Responsibilities	**Outputs**

• Specification of business requirements and definition of business models within the IHAN ecosystem.	• The outputs of this steering group will be Business requirements, or changes to these. Once approved, they shall form the basis for the work for the IHAN ecosystem
Current participants	

On highest level IHAN will be run as single simplified SAFe portfolio which is managed by IHAN BSG as Lean Portfolio Management – IHAN BSG has the highest level of decision-making and financial accountability for a IHAN portfolio. IHAN BSG will also appoint

- Epic Owners – They take responsibility to break down the requirements into Epics and Enablers that will be managed as IHAN Backlog using IHAN Kanban
- Enterprise Architect – This person works across value streams and programs to help provide the strategic technical direction that can optimize portfolio outcome



The following IHAN PSG managed portfolio-level artefacts help describe the strategic intent:

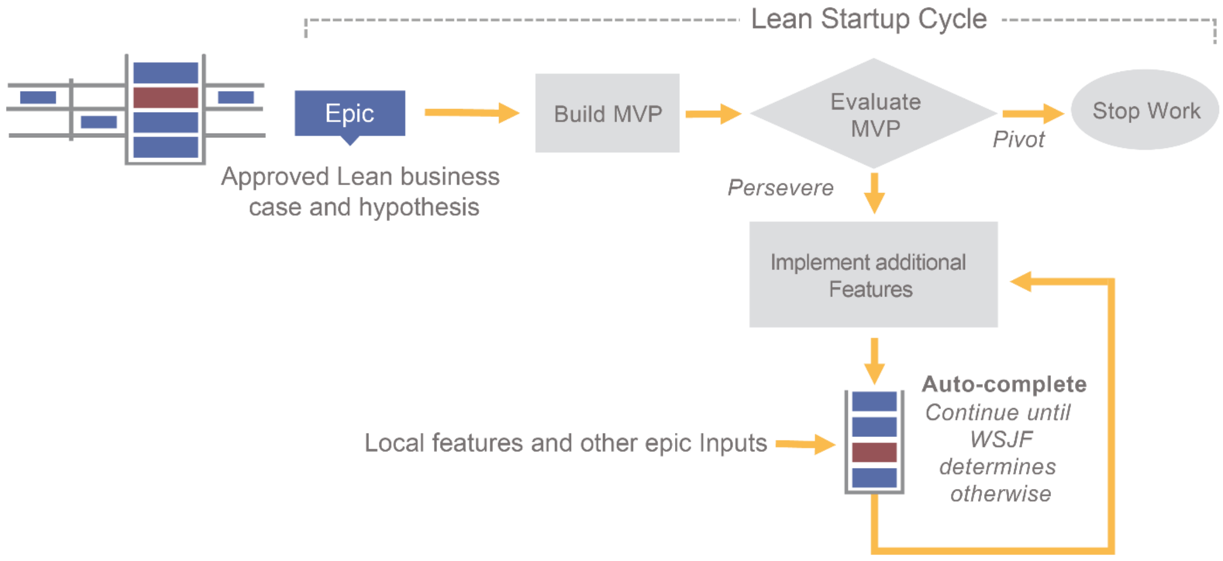
- Business Epics – Capture and reflect the new business capabilities that can only be provided through cooperation among value streams.

- Enabler epics – Reflect the architectural and other technology initiatives that are necessary to enable new Features and Capabilities.

- Strategic themes – Provide specific, itemized business objectives that connect the portfolio to the evolving enterprise business strategy.

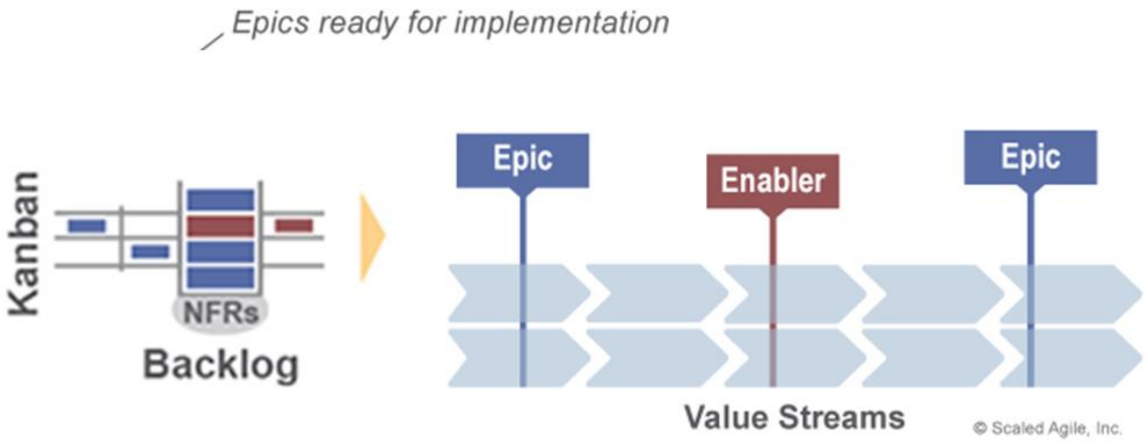
- Portfolio Backlog – Is the highest-level backlog in SAFe. It holds approved business and enabler epics that are required to create a portfolio solution set. This provides the competitive differentiation and/or operational efficiencies necessary to address the strategic themes and facilitate business success.

Lean Startup strategy recommends a highly iterative ‘build-measure-learn’ cycle for product innovation and strategic investments. Applying this model to epics provides the economic and strategic advantages of a Lean startup—managing investment and risk incrementally—while leveraging the flow and visibility constructs that SAFe provides

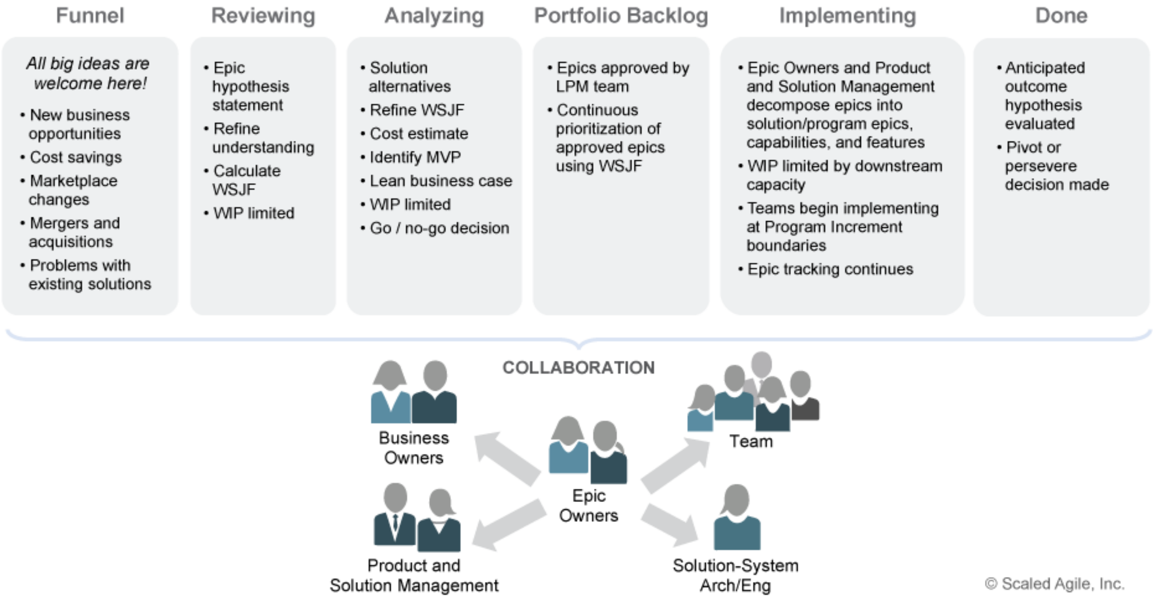

Lean Startup Cycle
Epic
Approved Lean business case and hypothesis
Build MVP
Evaluate MVP
Pivot
Stop Work
Persevere
Implement additional Features
Auto-complete
Continue until WSJF determines otherwise
Local features and other epic Inputs

© Scaled Agile, Inc.

IHAN Backlog is the highest-level backlog and pilot project backlogs will be subordinate to it. It provides a holding area for upcoming business and enabler Epics intended to create a comprehensive set of Solutions, which provides the competitive differentiation and operational improvements needed to address the Strategic Themes and facilitate business success.

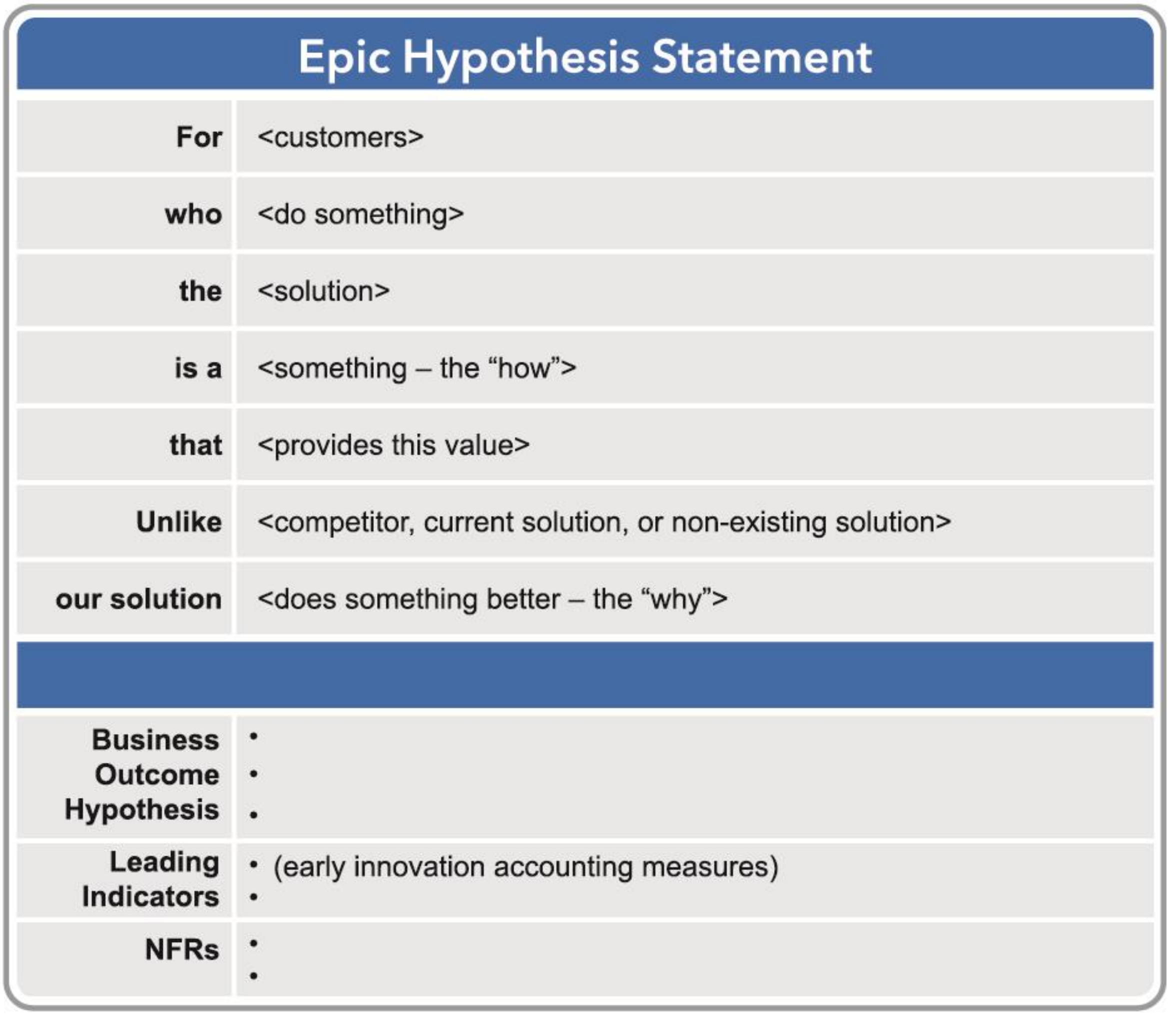


IHAN Kanban makes the work visible and creates Work-in-Process (WIP) limits to help assure that demand is matched to the pilot project capacities.


Funnel
All big ideas are welcome here!
• New business opportunities
• Cost savings
• Marketplace changes
• Mergers and acquisitions
• Problems with existing solutions
Reviewing
• Epic hypothesis statement
• Refine understanding
• Calculate WSJF
• WIP limited
Analyzing
• Solution alternatives
• Refine WSJF
• Cost estimate
• Identify MVP
• Lean business case
• WIP limited
• Go / no-go decision
Portfolio Backlog
• Epics approved by LPM team
• Continuous prioritization of approved epics using WSJF
Implementing
• Epic Owners and Product and Solution Management decompose epics into solution/program epics, capabilities, and features
• WIP limited by downstream capacity
• Teams begin implementing at Program Increment boundaries
• Epic tracking continues
Done
• Anticipated outcome hypothesis evaluated
• Pivot or persevere decision made
COLLABORATION
Business Owners
Epic Owners
Team
Product and Solution Management
Solution-System Arch/Eng
© Scaled Agile, Inc.

Reasoning about a potential epic must be based on a definition and intent that stakeholders can agree to. Epic hypothesis statement template is used to capture, organize, and communicate key information about an epic:

- Value statement – This is the structured 'for-who-the …' portion that describes the epic in general terms
- Business outcomes hypothesis – states the economic or other benefit outcomes the business can anticipate if the hypothesis is proven to be correct
- Leading indicators – describe the early measures that will help predict the business outcomes (For more on this topic, see the Innovation Accounting advanced topic article.)
- NFRS – identifies any Nonfunctional requirements associated with the epic



Epic Hypothesis Statement	
For	<customers>
who	<do something>
the	<solution>
is a	<something – the "how">
that	<provides this value>
Unlike	<competitor, current solution, or non-existing solution>
our solution	<does something better – the "why">
Business Outcome Hypothesis	• • •
Leading Indicators	• (early innovation accounting measures) •
NFRs	• •

@Scaled Agile, Inc.

9.2 Technical Steering Group IHAN Services & System Aspects

IHAN Technical Steering Group IHAN Services & System Aspects – TSG ISA is the approving body for IHAN ecosystem feature and service documentation. New components can only be created in this group and if components are deemed irrelevant it is by decision of this group

IHAN TSG ISA	
Terms of reference • TSG ISA reports to BSG BM	**Scope** • IHAN services and features high level documentation
Responsibilities • Approval of features and service specification	**Outputs** • The outputs of this steering group will be approved functional and non-functional requirements for IHAN services and features.
Current participants	

9.2.1 IHAN ISA WG1 Services

IHAN ISA Working Group 1 - Services is responsible to create specifications for IHAN services and features.

IHAN ISA WG1 Services	
Terms of reference • ISA WG1 reports to IHAN TSG ISA.	**Scope** • Service and feature requirements applicable to IHAN ecosystem for: • Data unit connected to identity • Access rights related to data unit - consents • Management of these access rights • Internal messaging and logging • Data management • Interworking with other systems
Responsibilities • Specification of features and services • Specification of service capabilities • Identification of requirements to support service operation. • Identification of requirements for service interworking. • Identification of requirements for service interoperability between networks. • Billing and accounting requirements	**Outputs** • The outputs of this working group will be Technical Specifications and Reports, or changes to these, which are all submitted to TSG ISA for approval. Once approved, they shall form the basis for the work for the whole of IHAN
Current participants	

9.2.2 IHAN ISA WG2 Architecture

IHAN ISA Working Group 2 - Architecture is responsible for the architecture of IHAN ecosystem

IHAN ISA WG2 Architecture	
Terms of reference • ISA WG2 reports to IHAN TSG ISA.	**Scope** • To have a system-wide view, and decides on how new functions integrate with the existing system entities • Definition, evolution and maintenance of the overall architecture including the assignment of functions to particular subsystems and associated high level functional interactions • In co-operation with the other TSGs, define required services, service capabilities and capabilities offered by the different subsystems, including Quality of Service requirements • In addition, the Architecture Working Group will consider how to carry out the technical co-ordination and overview role with the other TSGs

Responsibilities • Based on the services requirements elaborated by ISA WG1, ISA WG2 Architecture identifies the main functions and entities of the system, how these entities are linked to each other and the information they exchange	**Outputs** • The output of ISA WG2 is used as architectural input by ICSI and IAM working groups
Current participants	

9.2.3 IHAN ISA WG3 Privacy and Security

IHAN ISA Working Group 3 - Privacy and Security is responsible for the privacy and security aspects within the IHAN ecosystem

IHAN ISA WG3 Privacy and Security	
Terms of reference • ISA WG3 reports to IHAN TSG ISA.	**Scope** • The WG will perform analysis of potential threats to IHAN ecosystem, sub-systems and building blocks. • Based on the threat analysis, the WG will determine the security and privacy requirements for IHAN ecosystems and specify the security architectures and protocols. • The WG will ensure the availability of any cryptographic algorithms which need to be part of the specifications. • The WG will accommodate, as far as is practicable, any regional regulatory variations in security objectives and priorities • The WG will further accommodate, as far as is practicable, regional regulatory requirements that are related to the processing of personal data and privacy.
Responsibilities • ISA WG3 is responsible for security and privacy in IHAN ecosystems, determining the security and privacy requirements, and specifying the security architectures and protocols. • The WG also ensures the availability of cryptographic algorithms which need to be part of the specifications.	**Outputs** • The output of ISA WG3 is used as security and privacy requirement input by ICSI and IAM working groups
Current participants	

9.2.4 IHAN ISA WG4 Maintenance and Billing

IHAN ISA Working Group 4 - Maintenance and Billingis responsible for the value capture and transfer aspects within the IHAN ecosystem

IHAN ISA WG4 Maintenance and Billing

Terms of reference	**Scope**
ISA WG4 reports to IHAN TSG ISA	• Specifying the requirements, architecture, solutions and protocols for IHAN ecosystem Maintenance, Management, and Upgrading/Updating • Specifying the principles, architecture, servers and protocols for creating the IHAN ecosystem billing solutions.
Responsibilities • Defining IHAN Network management and maintenance solutions • Defining the Billing solutions and principles in IHAN Ecosystem	**Outputs** • The output of ISA WG4 is used as requirement input by ICSI and IAM working groups
Current participants	

9.3 Technical Steering Group IHAN Core System & Internetworking

IHAN Technical Steering Group IHAN Core System & Internetworking– TSG ICSI is the approving body for detailed IHAN ecosystem component and internetworking specifications.

IHAN TSG ICSI	
Terms of reference • TSG ICSI reports to BSG BM	**Scope** • IHAN components • IHAN interfaces • Data transfer from Data Providers to Service Providers • Identity management
Responsibilities • Approval of technical specifications for IHAN components • Approval of technical specifications for external interfaces between IHAN and Business Service layers • Coordination of specification work to allow Data transfer from Data Providers to Service Providers • Coordination of specification work within Identity management domain	**Outputs** • The outputs of this steering group will be approved technical specifications for IHAN components and interfaces • Approved mechanisms to transfer data and manage identities
Current participants	

9.3.1 IHAN ICSI WG1 Component Technical Specifications

IHAN ICSI Working Group 1 - Component Technical Specifications is responsible to create technical specifications for IHAN components.

IHAN ICSI WG1 Component Technical Specifications

Terms of reference • ICSI WG1 reports to IHAN TSG ICSI.	**Scope** • Functional descriptions and requirements of all IHAN components and modules e.g. data wallet, adapter nodes, servers, logging subsystems etc. • Specification of Interfaces and API's that modules make available to interact and interface within IHAN ecosystem
Responsibilities • Responsible for the IHAN specifications and requirements that define the IHAN ecosystem modules in details as described and identified by ISA WG2.	**Outputs** • The output of ICSI WG1 is used to implement and develop functional IHAN ecosystem components. IAM WG3 creates test specifications based on ICSI WG1 deliverables.
Current participants	

9.3.2 IHAN ICSI WG2 Interworking

IHAN ICSI Working Group 2 - Interworking is responsible for the architecture of IHAN ecosystem

IHAN ICSI WG2 Interworking	
Terms of reference • ISA WG2 reports to IHAN TSG ICSI.	**Scope** • Service interworking specifications e.g. between different IHAN ecosystem implementations. • Gateway specifications and functionalities to connect external networks • QoS specifications for Interworked networks
Responsibilities • Specifies the capabilities and scope for data services, and the necessary interworking functions between IHAN ecosystem and any external network identified by ISA WG2	**Outputs** • The output of ICSI WG2 are used to implement and develop the IHAN Interworking capabilities.
Current participants	

9.3.3 IHAN ICSI WG3 Data Transport

IHAN ICSI Working Group 3 - Data Transport is responsible for the data transportation aspects within the IHAN ecosystem. Even though Data Transport is not part of IHAN services and features a special working group is needed to liaise with Data Management related efforts. This Working group does not define the industry specific canonical data formats but co-operates efforts within those industries and adopts mature standards

IHAN ICSI WG3 Data Transport	
Terms of reference • ISA WG3 reports to IHAN TSG ISA.	**Scope** • Specifications for linking data routing information to IHAN components.

	- IHAN ecosystem requirements for User Data transport from different Data Providers to Service Providers - Data transport related IHAN messaging specifications
Responsibilities - Specifies requirements for transferring End Users' Data from Data Providers to Service Providers - .	**Outputs** The output of ICSI WG3 is used by the Data Providers and Service Providers to establish data transport and transfer gateways, interfaces and tunnels and link that to IHAN Ecosystem components and messaging
Current participants	

9.3.4 IHAN ICSI WG4 Identity Management

IHAN ICSI Working Group 4 - Identity Management is responsible for the identity management aspects within the IHAN ecosystem. Even though Identity Management is not part of IHAN services and features a special working group is needed to liaise with Identity Management related efforts

IHAN ICSI WG3 Privacy and Security	
Terms of reference ISA WG4 reports to IHAN TSG ISA	**Scope** - Specifying the requirements, architecture, solutions and protocols for IHAN ecosystem Maintenance, Management, and Upgrading/Updating - Specifying the principles, architecture, servers and protocols for creating the IHAN ecosystem billing solutions.
Responsibilities - Responsible for development and maintenance of specifications for Identity Management inside IHAN ecosystem.	**Outputs** - The output of ICSI WG4 is used to implement Identity structure and hierarchy in IHAN ecosystem and to link external Identity Management systems to IHAN identities.
Current participants	

9.4 Technical Steering Group IHAN Access Mechanism

IHAN Technical Steering Group IHAN Access Mechanism– TSG IAM is the approving body for detailed IHAN ecosystem protocol and contract specifications in addition to overall testing concept

IHAN TSG IAM	
Terms of reference - TSG IAM reports to BSG BM	**Scope** - IHAN protocols - IHAN smart contracts - Performance and Conformance aspects and testing of whole IHAN ecosystem
Responsibilities	**Outputs**

• Approval of technical specifications for IHAN protocols and smart contracts • Approval of testing	• The outputs of this steering group will be approved technical specifications for IHAN protocols and smart contracts • Approved mechanisms to transfer data and manage identities
Current participants	

9.4.1 IHAN IAM WG1 Protocols

IHAN IAM Working Group 1 - Protocols is responsible to create technical specifications for IHAN components.

IHAN IAM WG1 Component Technical Specifications	
Terms of reference • IAM WG1 reports to IHAN TSG IAM.	**Scope** • Specifications of protocols implementing Identity management profiles as defined by ICSI WG4 • Specifications of permission control and logging protocols. • Specifications of IHAN ecosystem Messaging and Metadata routing protocols. • Specifications of Smart Contract protocols as defined by IAM WG2 • Other relevant protocol specifications
Responsibilities • Responsible of the IHAN ecosystem Core protocols selection, definition and development that also provide scalability and security of the system.	**Outputs** • The output of IAM WG1 is used to create software that will be deployed while developing IHAN ecosystem components.
Current participants	

9.4.2 IHAN IAM WG2 Smart Contracts

IHAN IAM Working Group 2 - Smart Contracts is responsible for the detailed specifications of Smart contracts within IHAN ecosystem

IHAN IAM WG2 Interworking	
Terms of reference IAM WG2 reports to IHAN TSG IAM	**Scope** • Specification of roles of individuals and Service / Data providers in IHAN ecosystem. • Specification of permission statuses relating to data processing rights in IHAN ecosystem. • Linking the roles and permission statuses to specify flow chart descriptions of state diagrams of different allowed contractual situations in IHAN ecosystem.
Responsibilities • Responsible of specifying the Contract Structures that define the	**Outputs** • The output of IAM WG2 is used as a requirement input by other ICSI and IAM working groups.

Data Processing permission states within IHAN ecosystem	
Current participants	

9.4.3 IHAN IAM WG3 Data Transport

IHAN ICSI Working Group 3 - Data Transport is responsible for the data transportation aspects within the IHAN ecosystem. Even though Data Transport is not part of IHAN services and features a special working group is needed to liaise with Data Management related efforts. This Working group does not define the industry specific canonical data formats but co-operates efforts within those industries and adopts mature standards

IHAN IAM WG3 Data Transport	
Terms of reference • IAM WG3 reports to IHAN TSG IAM.	**Scope** • Conformance test case and setup descriptions and specifications for IHAN ecosystem and components and modules • Performance test case and setup descriptions and specifications for IHAN ecosystem and components and modules. • Specifications describing Test Network parameters and configuration that can be run parallel without interfering the main IHAN ecosystem.
Responsibilities • Responsible of creating Performance and Conformance testing specifications. • Responsible of creating Test System definitions and specifications.	**Outputs** • The output of ISA WG3 is used to create a test environment for IHAN ecosystem components development and new IHAN ecosystem features testing and development before entering the main IHAN ecosystem environment
Current participants	

9.5 Role based external stakeholder view

In the following table the external stakeholders that are involved in creation of IHAN ecosystem are identified

Role	**Expectations**
End User	*Understand how he/she can benefit from new kinds of services*
Service Provider Business Development	*Understands the benefits that offering services within IHAN ecosystem will bring to the organization*
Service Provider IT development	*Can easier build business services when using IHAN ecosystem components*

Data Provider Business Development	*Understands the benefits that opening data for IHAN ecosystem will bring to the organization*
Data Provider IT development	*Can easier build interfaces so data confined within the organization can be opened for IHAN ecosystem Service Providers*
Regulator (GDPR)	*Can see an ecosystem being built that abides with all GDPR regulation articles*

10 Design Decisions

This section is intentionally blank

11 Quality Requirements

This section is intentionally blank

12 Risks and Technical Debts

This section is intentionally blank

13 Glossary

This section is intentionally blank

Template

2017-05-03

About arc42

arc42, the Template for documentation of software and system architecture.

By Dr. Gernot Starke, Dr. Peter Hruschka and contributors.

Template Revision: 7.0 EN (based on asciidoc), January 2017

Note

This version of the template contains some help and explanations. It is used for familiarization with arc42 and the understanding of the concepts. For documentation of your own system you use better the *plain* version.