Health Information Exchange using Health Data Consent Managers

Technical Standards

Sequence #:
Version: 0.9

Author: iSPIRT, ReBIT

Track: Standard
Maturity Level: Draft
Type: Technical Specifications
Status: Under Internal Review

Feb 01, 2020

Abstract: This document presents the architecture for consented sharing of Health Information between Health Information Providers and Health Information Users via a new type of entity called Health Data Consent Manager. It outlines the use cases that are covered by this architecture, the responsibilities of the various stakeholders in these use cases and their interactions.

The architecture is closely based on the technical standards for Account Aggregators and the related API specifications, both of which are authored by Reserve Bank Information Technology Pvt. Ltd (ReBIT). iSPIRT was a key contributor to that work. Part of the ReBIT document (the API specifications) can be found on this link: https://api.rebit.org.in/group

Version History

Version	Date	Comments
0.5	26/10/2018	Preliminary Draft
0.9	26/11/2018	Draft
0.91	21/01/2019	Revised architecture with Health Data Access Fiduciary renamed to Health Data Consent Manager, direct data transfer with pull-based flows, HIP push-based linkage and applications section

Table of Contents

1 Introduction **6**
1.1 System Design Guidelines 6
1.2 Conventions used in the Document 7
1.3 Scope of the Document 7
1.4 About Health Specifications 7
1.5 References 7

2 Technical Specification **8**
2.1 Terms and Definitions 8
2.2 Entities Roles and Responsibilities 9
2.3 High Level Architecture 11
2.4 Linking Accounts of a Customer 13
2.4.1 Sign Up 13
2.4.2 Customer Address 13
2.5 Electronic Consent 13
2.6 Health Information Types 16
2.7 Purpose of Data Sharing 18

3 Use Cases **18**
3.1 Discover Accounts 19
3.1.1 Account and Identity Validation 21
3.2 Link HIP Accounts at HDCM 21
3.2.1.1 Link Account (Pull-based Discovery) 22
3.2.1.2 Link Account (Push-based Discovery) 23
3.2.2 Delink Account 25
3.3 Create Consent Artefact(s) 26
3.4 Request Health Information 28
3.4.1 HDCM-mediated one-time transfer 30
3.4.2 Direct one-time transfer 32
3.4.3 Periodic transfer 34
3.5 Consent History 35
3.6 Revoke Consent 36

4 Health Information **38**
4.1 Data Schemas for specific HI 38
4.2 Document Specifications for Health Information 38
4.3 Unsupported HI types 39
4.4 Deep Linking Format 39

5 Non Functional Requirements **40**

5.1 Reliability Considerations 40
5.1.1 Failure Scenarios 40
5.2 Security Considerations 41
5.2.1 Digital Identifiers 41
5.2.2 Customer Authentication between HDCM and HIP 42
5.2.3 Using a secondary PIN by customer in the Consent flow 42
5.2.4 Guidelines for API Security 42
5.2.5 Customer Management and Customer Protection 44
5.2.6 Fraud Detection and Analysis 44
5.2.7 Anonymity of HIUs when requesting information from HIPs 44
5.2.8 Encryption of Health Information 44
5.2.8.1 Data In-Flight Encryption 44
5.2.8.2 Data At-Rest Encryption 47
5.2.8.3 Controlling access to health information by Health Data Access Fiduciaries 47
5.3 Audit Considerations 47
5.4 Privacy Considerations 47
5.4.1 Data Masking and Identity Protection Guidelines 47
5.4.2 Data Portability Guidelines 48
5.5 Consumer Experience Considerations 48
5.6 Developer Experience Considerations 49
5.7 Grievance Redressal 49

6 Applications of Health Information Exchange 50
Insurance 50
Second opinions 50
Health locker 50
Drug trials 50

7 Appendix 51
7.1 Notifications 51
7.1.1 Account Linking Lifecycle Events 51
7.1.2 Consent Lifecycle Events 51
7.1.3 Data Lifecycle Events 52
7.1.4 Data Availability Notifications 53
7.2 Summary of APIs by Entity 53
7.2.1 Central Registry - Health Sector Regulator Metadata 54
7.2.2 Health Data Consent Manager 54
7.2.3 Health Information Provider 55

1 Introduction

A **Health Data Consent Manager** (HDCM) is a new type of entity proposed here whose task is to provide health information aggregation services to customers of healthcare services. It enables customers to fetch their health information from one or more **Health Information Providers** (e.g., Hospitals, Diagnostic Labs, Medical Device Companies), based on a Customer's explicit **Consent** and to share such aggregated information with **Health Information Users** i.e. entities in need of such data (e.g., Insurers, Doctors, Medical Researchers, Personal Health Record Applications).

This document contains the technical standards for implementing HDCMs and explains how health information exchange between health information providers, health information users and customers is enabled by HDCMs.

1.1 System Design Guidelines

Our technical standards adhere to the following guidelines:

- **Technology Agnostic**: The proposed design in the document is agnostic to applications, programming languages, and platforms and aims at seamless and secure flow of electronic data across different stakeholders.
- **Reliability and Scalability**: Reliability and scalability of the system is given particular emphasis. In the future, entities like these should be able to handle requests of hundreds of millions of customers.
- **Privacy by Design**: Customer information needs to be protected from abuse and compromise. The HDCM concept gives customers control of their data and ensures the privacy of data ground-up.
- **Security by Design**: The framework is designed from the ground up to be secure. End-to-end security of data flows between information providers and users is given special attention.
- **Minimalist and Evolutionary Design**: The HDCM design is simple and minimalistic. We also attempt to ensure that it is evolutionary in nature: capabilities are built incrementally while allowing for rapid adoption.
- **Customer Centric**: Customer experience and ease of use are critical to successfully delivering the various services in the ecosystem. The design principles take into account the various stakeholder responsibilities and mechanisms to simplify interactions and minimise friction.
- **Open APIs for Interoperability and Layered Innovation**: Systems should have programmatic interfaces for sharing and accessing the information available to them. The specification defines standard APIs to promote interoperability and deliver services that are designed to work with any device, any form factor, and any network.
- **Transparency and Accountability through Data**: Attempt is made to ensure Public Open Data is available via APIs. Access to open data will ensure high-quality analytics, accurate fraud detection, shorter cycles for system improvement and high responsiveness to Customer needs.

1.2 Conventions used in the Document

- Key terminologies use dark cornflower blue 3 color encoding.
- The **lowerCamelCase** is used in attribute naming. For nouns, **UpperCamelCase** is used.
- The document uses XML to illustrate examples and structures of entities.

1.3 Scope of the Document

This document describes the architecture for HDCMs and the associated health information exchange ecosystem, the use cases that are enabled in this ecosystem and the responsibilities of the various stakeholders. It also provides examples of HI data schemas that may be queried from the HIP for the purpose of data aggregation. High level API definitions are provided in the Appendix.

Technological underpinnings such as digital signatures, automated auditing, logging and non-repudiation concepts shall result in verifiability of transactions thus ensuring integrity of the relevant systems and supporting better grievance redressal.

1.4 About Health Specifications

These specifications are developed via industry led collaborative efforts with NCG providing the logistic support as a standards development organization (**SDO**).

1.5 References

[1] https://tmc.gov.in/ncg/index.php/ncg-members/list-of-members

[2] What are the different classes of Digital Signature Certificates? http://cca.gov.in/cca/?q=node/45, (assessed 10 March 2018)

[3] XML Signature Syntax and Processing Version 1.1, https://www.w3.org/TR/xmldsig-core1/ (accessed 2 February, 2018)

[4] Leach, Paul J., Michael Mealling, and Rich Salz. "A Universally Unique Identifier (UUID) URN Namespace." (2005). https://tools.ietf.org/html/rfc4122 (accessed 2 February, 2018)

[5] "JSON Schema". http://json-schema.org/ (accessed 2 February, 2018)

[6] Jones, Michael, and Dick Hardt. "The oauth 2.0 authorization framework: Bearer token usage". No. RFC 6750. 2012. https://tools.ietf.org/html/rfc6750 (accessed 2 February,2018)

[7] Institute for Development and Research in Banking Technology (IDRBT). "Certificate Authority". http://www.idrbt.ac.in/idrbtca.html (accessed 2 February,2018)

[8] Department of Science & Technology, Government of India. "National Data Sharing and Accessibility Standards". https://data.gov.in/sites/default/files/NDSAP.pdf (accessed 2 February,2018)

[9] Jones, Michael, John Bradley, and Nat Sakimura. "JSON web signature (JWS)". No. RFC 7515. 2015. https://tools.ietf.org/html/rfc7515 (accessed 2 February,2018)

[10] T. Berners-Lee, R. Fielding, and L. Masinter, Uniform Resource Identifier (URI): Generic Syntax, (January 2005) https://tools.ietf.org/html/rfc3986 (accessed 10 March, 2018)

[11] Income Tax Department, Online PAN verification Options, https://www.incometaxindia.gov.in/Pages/tax-services/online-pan-verification.aspx (accessed 10 March, 2018)

[12] Daniel J. Bernstein and Tanja Lange. SafeCurves: choosing safe curves for elliptic-curve cryptography. https://safecurves.cr.yp.to , (accessed 10 March 2018).

[13] NIST Special Publication 800-63B, Digital Identity Guidelines, https://pages.nist.gov/800-63-3/sp800-63-3.html , (assessed 10 March 2018).

2 Technical Specification

2.1 Terms and Definitions

HDCM	The Health Data Consent Manager is an entity that acts as a consent collector for the Customer and enables the Health Information data flows from the HIP to the recipient HIU or the Customer. HDCMs must provide (a) a server implementation and (b) standardised APIs for a front-end that will enable customers to fetch aggregated health information on their personal devices. The health information of a customer does not pass through the HDCM and HDCMs cannot aggregate such information. It is not to be used in any manner other than as consented to by the Customer.
HDCM Client	The HDCM front-end is referred to as the **HDCM Client.**
Consent Artefact	A consent artefact is a machine-readable electronic document that specifies the parameters and scope of data sharing that a Customer consents to in any data sharing transaction.
HI	Health Information refers to information about a Customer's health records such as referrals, health test reports, health images and other health information as applicable. It is obtained from Health Information Providers.
HIP	“Health Information Provider” refers to clinical establishments which generate or store customer data in digital form. These include hospitals, primary or secondary health care centres, nursing homes, diagnostic centres, clinics, health workers (ASHAs, ANMs), medical device companies and other such entities as may be identified by regulatory authorities from time to time
HIU	“Health information User” refers to an entity that wishes to consume the services of the Health Data Consent Manager and obtain aggregated health information for providing services to the **Customer**.
Customer	Customer for the purpose of NCG-HDCM service means either a patient who is availing the services of an HIP or an HIU or someone who is using other health-related services provided by an HIP or HIU. Customers maintain accounts with HIPs. Customers must register with HDCMs in order to be able to give consent to HIUs to access their HI. This registration process also results in the creation of a customer account maintained by the HDCM.
Central Registry	The Central Registry provides the HIP, HIU and HDCM public key information so that ecosystem components can validate digital signatures of these entities.

2.2 Entities Roles and Responsibilities

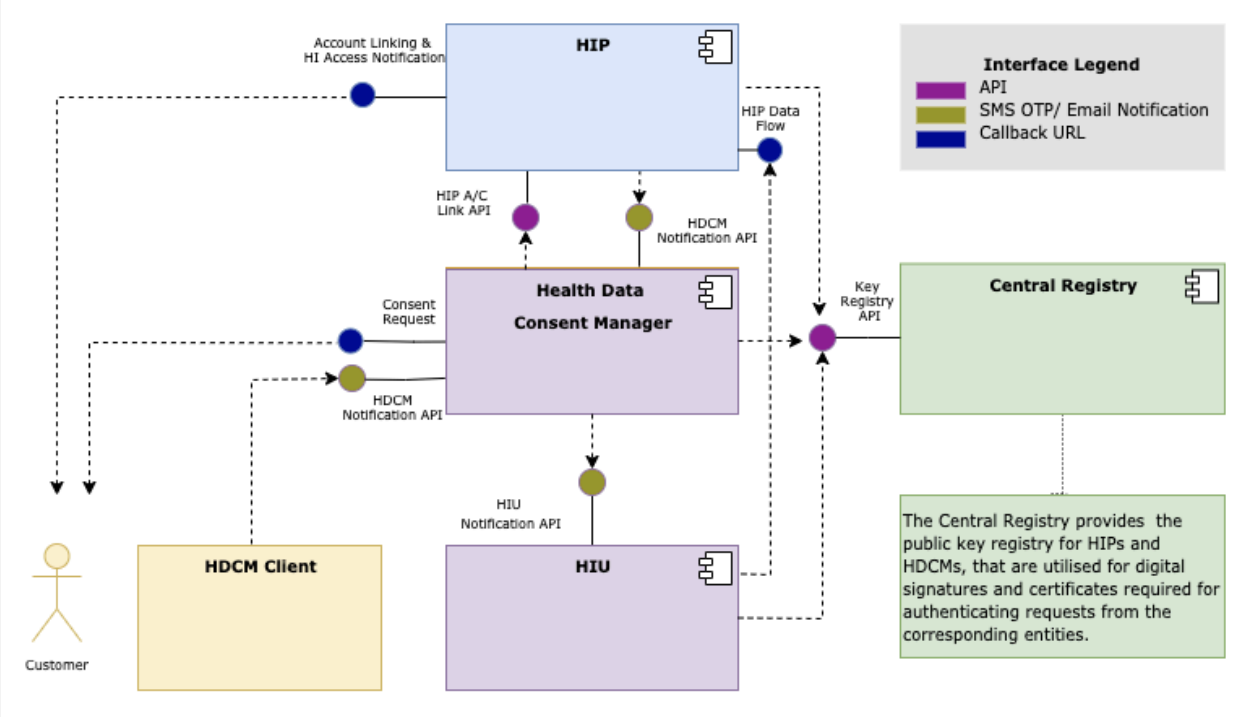
The following table defines the roles and responsibilities of the entities in the ecosystem.

HDCM	The HDCM is responsible for ensuring that information is requested from HIPs based on the customer's explicit consent. It collects consent from the customer for every information request and creates consent artefacts which contain all parameters of the consent granted by the customer. The HDCM manages the lifecycle of consent artefacts, including activities like revocation and pausing of consent. It may mediate the actual flow of information from HIPs to HIUs. The HDCM maintains logs of all requests for information from HIUs and all events associated with the flow of information. It also maintains logs about consent granted by the customers and any events associated with the consent lifecycle management. The HDCM must notify customers, HIUs and HIPs about key consent related events e.g., whenever a consent artefact is revoked by the customer, the associated HIU and HIP must be notified about it.
HDCM Client	The HDCM Client is an authorized client that interacts with the HDCM service. It may be implemented as library, SDK or might interact via direct authorized HDCM API calls. The HDCM Client provides its customers an interface using which they can view and manage consent artefacts associated with them and, optionally, an interface for the customers to view their aggregated health information.The HDCM client may operate in the customer's environment and provide the necessary customer interactions for requesting the health information based on the customer's consents. The HDCM client could be a web based application, a mobile based application offered by the HDCM or a SDK/library embedded in another application.
HIP	The HIP provides information to the HDCM about the nature of health information of customers held by it. It also provides the actual health information (in encrypted form) in response to consented requests from the HDCM. The HIP must maintain logs of all information request queries received from HDCMs. It must service queries only from authorized HDCMs. The HIP interacts with the Central Registry and keeps its public key information updated in that registry.
Central Registry	The Central Registry provides the public key certificates of HIPs, HIUs and HDCMs, that are utilized for digital signatures and authenticating requests from the corresponding entity.

Customer	The customer maintains and interacts with the HDCM to link accounts and provides electronic consent.
HIU	The customer may designate an HIU as the recipient of the Health Information when requesting consent artefact creation to HDCM. The HIU needs to maintain the customer's health information provided to it securely in compliance with the terms of the consent granted by the Customer. The HIU interacts with the Central Registry and keeps its public key information updated in that registry.

2.3 High Level Architecture

Our approach for health information exchange uses a layered approach that treats health information generation and sharing (done by the HIP) separately from health information consumption (done by the HIU). The HDCM acts as an intermediary in the information exchange process and helps connect the HIU to multiple HIPs through standardized APIs. The following diagram shows the various interfaces and system interactions in the health information exchange ecosystem.



The customer interacts with the HDCM Client or the HIU for requesting services. The HDCM Client interfaces with the HDCM via the API exposed by the HDCM. The customer interacts with the HDCM to link accounts and generate consent, all such interactions must happen directly between the customer and the HDCM. Aggregated Health Information is made available to the customer by the HDCM Client.

The central registry provides the necessary information about registered HIPs, HIUs and HDCMs in the ecosystem, their corresponding certificates for facilitating the cryptographic functions etc. Every registered HDCM and HIP/HIU must use the registry to verify digital signatures.

The architecture uses asynchronous mechanisms to enhance scalability and provides deterministic mechanisms to fetch consents and health information. The notification callbacks include handles for such deterministic calls enabling a decoupled operation from when the request is made, when recipients get notified about the success and failure status and when the recipient makes a call to retrieve the consent and health information data.

The following Interfaces are defined (as shown in the diagram):

Interface	Summary
HIP's Account Linking API	This API enables the HDCM to link HIP account(s) of the Customer with the Customer's HDCM account. Health information can be fetched only from accounts that are linked with an HDCM account.
HIP's DataFlow API	This API provides the interface to the HIU to fetch health information from the HIP. (Health information is subsequently forwarded to the HIU or the HDCM Client.)
HDCM's Consent Flow API	This interface enables the HIU to submit consent creation requests to the HDCM in response to which the HDCM collects consent from the Customer (via the HDCM client). Once consent is collected, the corresponding consent artefact can be shared with the HIU and can be used to fetch health information in the future.
HDCM's Data Flow API	The HIU (or the HDCM Client) uses this HDCM interface to fetch health information. In response to information fetch requests from HIUs, the HDCM itself issues information fetch requests to HIPs using the HIPs' data flow API.
HDCM's Notification API	The HDCM implements a notification interface to receive asynchronous status updates from HIPs in response to information Fetch requests
HIP's Notification API	This API enables the HIP to receive notifications from the HDCM for any changes to consent artefacts (e.g., creation or revocation of a consent artefact) involving information stored by the HIP.
HIU's Notification API	This is a notification interface hosted by the HIU to receive asynchronous status updates in response to its information fetch requests.
Key Registry API	This is an API hosted by the central registry which enables system entities (HIPs, HIUs or HDCMs) to discover other entities and their corresponding public key certificates.

Other than these APIs, the HDCMs and the HIPs must implement **notification services** which notify the customer about various events involving consent or the actual flow of information. For example, any changes in the status of a consent artefact (creation, revocation, pausing, etc) have to be communicated to the concerned customer. All HI access requests received by an HIP also have to be communicated to the customer.

2.4 Linking Accounts of a Customer

Before Customers can request health information, they need to specify the information that they want to make available for information exchange via the HDCM. Of all the accounts held by a customer at various HIPs, a subset of accounts are “linked” to the customer’s HDCM account. This linkage happens with explicit authorization from the customer. The HDCM is provided account numbers only for linked accounts. Each HIP must maintain a record of accounts that have been linked to an HDCM account of the customer, and must share information with the HIUs only from accounts that have been linked (upon presentation of an appropriate consent artefact). Linked accounts can also be delinked by the HDCM (upon request from the customer).

2.4.1 Sign Up

We recommend that the process of signing up with a HDCM be based on at least one strong identifier.

2.4.2 Customer Address

Each Customer who has an account with the HDCM is identified by a unique customer address. It represents a handle for searching for HDCM account details. All HDCM account addresses are denoted as “account@provider" form. Address should only contain a-z, A-Z, 0-9, .(dot), - (hyphen).

<Customer_identifier>@<HDCM_identifier>

The customer address becomes critical when an HIU wishes to collect consent from the customer for some information and it needs to communicate to the HDCM which customer’s consent is being sought. The construct of the customer identifier is similar to that of the UPI ID used in Unified Payments Interface (UPI) transactions.

2.5 Electronic Consent

As discussed, HDCMs must provide services to a customer based on her explicit consent. In our framework, consent is always electronically sought from the Customer and is captured using a digital object called the “Consent Artefact”. Consent artefacts comprise of the following components:

- **Identifiers**: This component specifies all entities that are involved in the information sharing transaction: the HIP issuing the information, the HIU accessing it, the HDCM, and the customer.
- **Permission section**: The Permission Section comprises fields describing the type of information (the HITypes) that is being exchanged and the access permissions. Permissions include the

duration for which information is requested (e.g., the past 6 months' of health records/information), the allowed duration of storage by Health Information (referred to as "datalife"), the frequency of access (in case information is allowed to be fetched repeated), along with a set of pre-processing "data filters" that can be used to filter out the information that is retrieved. Access permissions can be of four types:

- The requester can either get VIEW access to the information, which implies that it is not allowed to store the data or reuse it later.
- The HIU or HDCM client can STORE the information and use it within the period defined in datalife. All information must be exchanged between the HIP and the information requester in a secure fashion using either data and/or channel encryption. Information must be destroyed after the datalife.
- The STREAM permission facilitates in-point streaming of information to the HIU or the HDCM client.
- The QUERY permission allows additional filtering criteria to be included in the consent artefact. This allows the HIP to preprocess the data before responding to the request. The QUERY filter parameters may be defined by the HIU.

- **Purpose of access**: This component includes information about the application domain (e.g., health insurance) and the application within that domain that is enabled through the health information access (e.g., claims processing). These are specified using suitable codes (discussed later). A free-form textual description is also included.
- **Signature**: The Consent Artefact is digitally signed by HDCM as per the W3C recommendations [3] for XML format. The digital signature forms the final component of the artefact.

When an HIU or the customer (via the HDCM Client) requests for health information, the HDCM collects consent for this from the Customer. The HDCM creates two different consent artefacts for information requested from an HIP account:

1. The first Consent Artefact authorizes the HIU or the HDCM Client to request information from the HIP.
2. The second Consent Artefact authorizes the HDCM to obtain information from the HIP for subsequent transfer to the HIU or the Customer. This artefact does NOT mention the details of the HIU that is requesting the information. This is done in order to ensure anonymity of the HIU in information requests, which helps prevent differentiated service delivery by the HIPs.

Consent Artefact XML between HIU/Customer and HDCM

```
<?xml version="1.0" encoding="UTF-8"?>
<ConsentArtefact xmlns="http://standards.tmc.gov.in/HDCM" id="" createTime="YYYY-MM-DDThh:mm:ssZn.n"
expireTime="YYYY-MM-DDThh:mm:ssZn.n" revocable="true|false" >

  <!-- Identifiers -->
  <HIP id= "" />
  <HIU id= "" />
```

```
<HDCM id= "" />

<!-- Following element captures the customer ID (i.e. Customer Address) -->
<CustomerID idType="" id=""> </CustomerID>

<!-- Following element captures the HITypes to which access is given -->
<HITypes>
    <HIType> “DIAGNOSTIC-REPORT “ </HIType>
</HITypes>

<!-- following element repeats -->
<Permission name=””>
      <Access mode="STORE|QUERY|STREAM" />
      <!-- the duration for which information is requested -->
      <Duration unit="DAY|MONTH|YEAR|INF" value="" />
      <!-- how long can the information requester store data -->
      <Datalife unit="MONTH|YEAR|DATE|INF" value="" />
      <!-- frequency and number of repeats for periodic information access -->
      <Frequency unit="DAILY|MONTHLY|YEARLY" value="" repeats="" />
      <Data-filter>
          <!-- Data access filter, any encoded query string as per health information provider API needs -->
       </Data-filter>
</Permission>

<!-- Logging block -->
<ConsentUse logUri=""/>
<DataAccess logUri=""/>

<!-- Purpose block -->
<Purpose code="" refUri="">
  <!-- purpose text goes here -->
</Purpose>

<!-- Signature block -->
<Signature> Signature of HDCM as defined in W3C standards; Base64 encoded   </Signature>

</ConsentArtefact>
```

Consent Artefact XML between HDCM and HIP

```
<?xml version="1.0" encoding="UTF-8"?>
<ConsentArtefact xmlns="http://standards.tmc.gov.in/HDCM" id="" createTime="YYYY-MM-DDThh:mm:ssZn.n"
expireTime="YYYY-MM-DDThh:mm:ssZn.n"
revocable="true|false">
```

```
<!-- Identifiers -->
<HIP id= "" />
<HDCM id= "" />

 <!-- Following element captures the customer ID (i.e. Customer Address) -->
<CustomerID idType="" id=""> </CustomerID>

 <!-- Following element captures the HITypes to which access is given -->
<HITypes>
    <HIType> “DIAGNOSTIC-REPORT “ </HIType>
</HITypes>

<!-- Permission Block -->
<Permission name=””>
   <Access mode="VIEW|STORE|QUERY|STREAM" />
   <!-- the duration for which information is requested -->
   <Duration unit="DAY|MONTH|YEAR|INF" value="" />
   <!-- how long can consumer is allowed to store data -->
   <Datalife unit="DAY|MONTH|YEAR|INF" value="" />
   <!-- frequency and number of repeats for access repeats -->
   <Frequency unit="DAILY|MONTHLY|YEARLY" value="" repeats="" />
   <Data-filter>
        <!-- Data access filter, any encoded query string as per health information provider API needs -->
    </Data-filter>
</Permission>


<!-- Logging block -->
<ConsentUse logUri="" />
<DataAccess logUri="" />

<!-- Purpose block -->
<Purpose code="" refUri="">
  <!-- purpose text goes here -->
</Purpose>

<!-- Signature block -->
<Signature> Signature of HDCM as defined in W3C standards; Base64 encoded  </Signature>

</ConsentArtefact>
```

2.6 Health Information Types

The different types of **Health Information** (**HI**) applicable for consented information sharing between HIPs and HIUs are described in the following table along with the corresponding Health Information Type (**HIType**) identifiers. The table is mapped to FHIR Resources that define a common way to define and represent the corresponding data types. FHIR (Fast Health Interoperability Resources) is an internationally accepted data exchange standard which can be used in partnership with existing widely used standards.

#	Health Information	Health Information Type (HIType)	Data type sub-category
1	Demographics and other administrative information about an individual or animal receiving care or other health-related services	Patient	
2	A person who is directly or indirectly involved in the provisioning of healthcare.	Practitioner	
3	An interaction between a patient and healthcare provider(s) for the purpose of providing healthcare service(s) or assessing the health status of a patient	Encounter	
4	The findings and interpretation of diagnostic tests performed on patients, groups of patients, devices, and locations, and/or specimens derived from these. The report includes clinical context such as requesting and provider information, and some mix of atomic results, images, textual and coded interpretations, and formatted representation of diagnostic reports	DiagnosticReport	(FHIR Codes)
5	Outside scanned report	scannedReport	
6	Oncology Plan of care and summary note	Encounter	treatment summary
7	Hematology+Medical oncology Discharge summary	Encounter	discharge summary
8	A clinical condition, problem or diagnosis	Condition	
9	An action that is or was performed on a patient	Procedure	
10	Surgical procedure	Procedure	surgical

11	Clinical finding	ClinicalImpression	finding
12	A clinical assessment performed to determine what problem(s) may affect the patient	ClinicalImpression	
13	Measurements and simple assertions made about a patient, device or other subject. Including “social history”, “vital signs”, “imaging”, “laboratory”, “procedure”, “survey”, “exam”, “therapy”	Observation	
14	Representation of the content produced in a DICOM imaging study	ImagingStudy	
15	This resource is primarily used for the identification and definition of a medication. It covers the ingredients and the packaging for a medication	Medication	
16	Describes the intention of how one or more practitioners intend to deliver care for a particular patient, group or community for a period of time, possibly limited to care for a specific condition or set of conditions	CarePlan	
17	The Care Team includes all the people and organizations who plan to participate in the coordination and delivery of care for a patient	CareTeam	
18	A container for a collection of resources	Bundle	
19	A set of healthcare-related information that is assembled together into a single logical package that provides a single coherent statement of meaning, establishes its own context and that has clinical attestation with regard to who is making the statement. A Composition defines the structure and narrative content necessary for a document	Composition	

The health information provided by the HIPs, as shown in the table above, may be in varied different formats. A specific type of health information is represented by the health information type definitions (**HIType**). The architecture described in this standard is generic in nature and supports extension of the health information that can be aggregated based on new health information type definitions

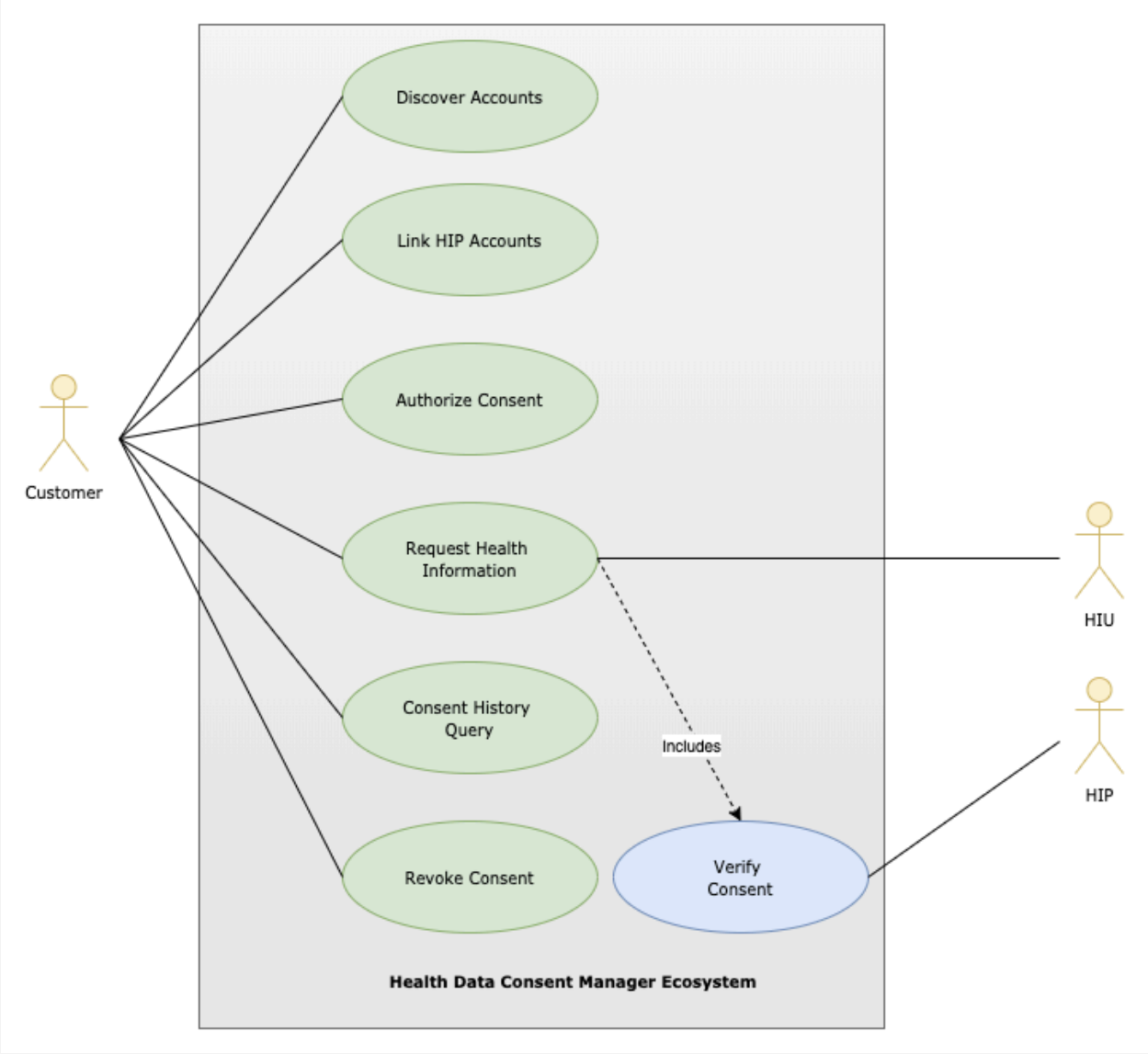
2.7 Purpose of Data Sharing

This section sets some guidelines for defining the purpose of data sharing. We refer to the FHIR standards for defining these in the context of healthcare:
https://www.hl7.org/fhir/v3/PurposeOfUse/vs.html

3 Use Cases

The Health Data Consent Manager ecosystem would support the following minimal use cases.

High Level Use Case Diagram

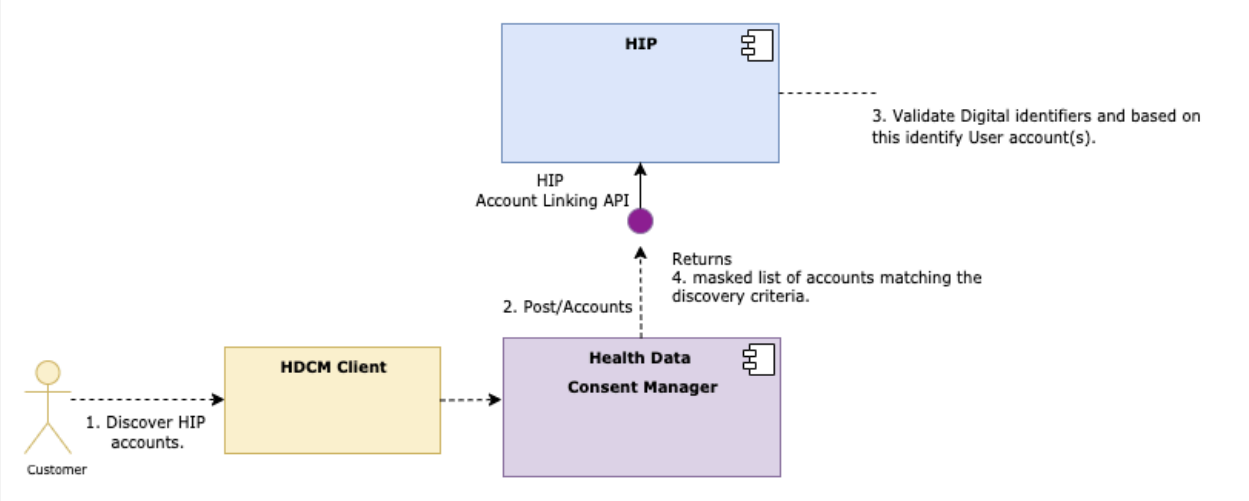


3.1 Discover Accounts

In this use case, a customer initiates linkage of the HIP accounts to their account at the HDCM. The HDCM, based on input from the customer, passes a verified digital identifier such as a Mobile Number, HIP Issued Patient ID, Aadhaar Number/Aadhaar Virtual ID to the "**POST /Accounts**" API of the HIP to obtain a list of masked account numbers. The Customer may choose to subsequently link these

discovered account(s) to their account at the HDCM. The linking of accounts, is a prerequisite for consents to be generated. Customer consent can only be generated for linked accounts.

The discovery relies on identifying a customer's account(s) based on a set of matching digital identifiers and their corresponding mapping with the customer accounts held by the HIP. Multiple identifiers may be passed in the discovery process but the request must include at least one verified strong identifier. Mobile Number and Aadhaar numbers are examples of verifiable strong digital identifiers. The following diagram illustrates the operation of the "Discover Accounts" use case:



Summary	This Use Case allows the Customer to discover their HIP accounts
Pre-condition	• The Customer has logged into the HDCM Client • The Customer has at least one strong verified digital identifier
Actor	Customer
Main Flow	1. The customer initiates an account discovery process across one or more HIPs. a. The HDCM must ensure that at least one strong identifier is present in the request. b. The HDCM Client may ask the customer to optionally select **HITypes** for further filtering in the account listing. 2. The HDCM uses the **POST /Accounts** API to receive the masked account details of the customer. 3. The HIP validates the passed verified identifiers and returns the list of accounts with (potentially masked) account numbers back to the HDCM. Masking of account numbers is performed based on the HIPs discretion. 4. The HDCM displays this information to the Customer for subsequent account linkage on the HDCM Client. 5. The use case ends.

Alternate Flows	**Alt.1**: Account at HIP not discovered In this case, the HDCM responds to the HDCM Client with an error code and explanation. For such HIPs the HDCM may provide a choice to the customer to enter additional identifiers such as Patient ID/Account Number and then retry the account discovery process.
Post-condition	A list of (potentially masked) HIP account numbers and a corresponding **AccountsRefNumber** is provided to the HDCM. The customer can then choose to link some or all of these accounts with her HDCM account. The customer can then authenticate himself to the HIP account to link the same with her HDCM account.
Security and Audit Requirement	In the Account Discovery request, the HDCM must mark all identifiers it has verified, that includes strong identifiers, weak identifiers and custom non-verifiable functional identifiers that the customer provides. The HIP must only respond if at least one of the strong identifiers match the record of the customer.
Comments	Aadhaar is considered a strong identifier. The HDCM may choose one of the available methods of verifying the customer Aadhaar number. Furthermore, the mobile number and emails can be verifier using OTP and code verification mechanisms. Patient ID can also be verified against the HIP system using either user credentials or OTP send to the registered email/mobile number.

3.1.1 Account and Identity Validation

The discovery happens in the HDCM domain. The customer must have an account in the HDCM domain. The HDCM must have provisions for the customer to verify, if they haven't already verified, during authentication, a strong identifier like their Mobile Number and the Aadhaar Number.

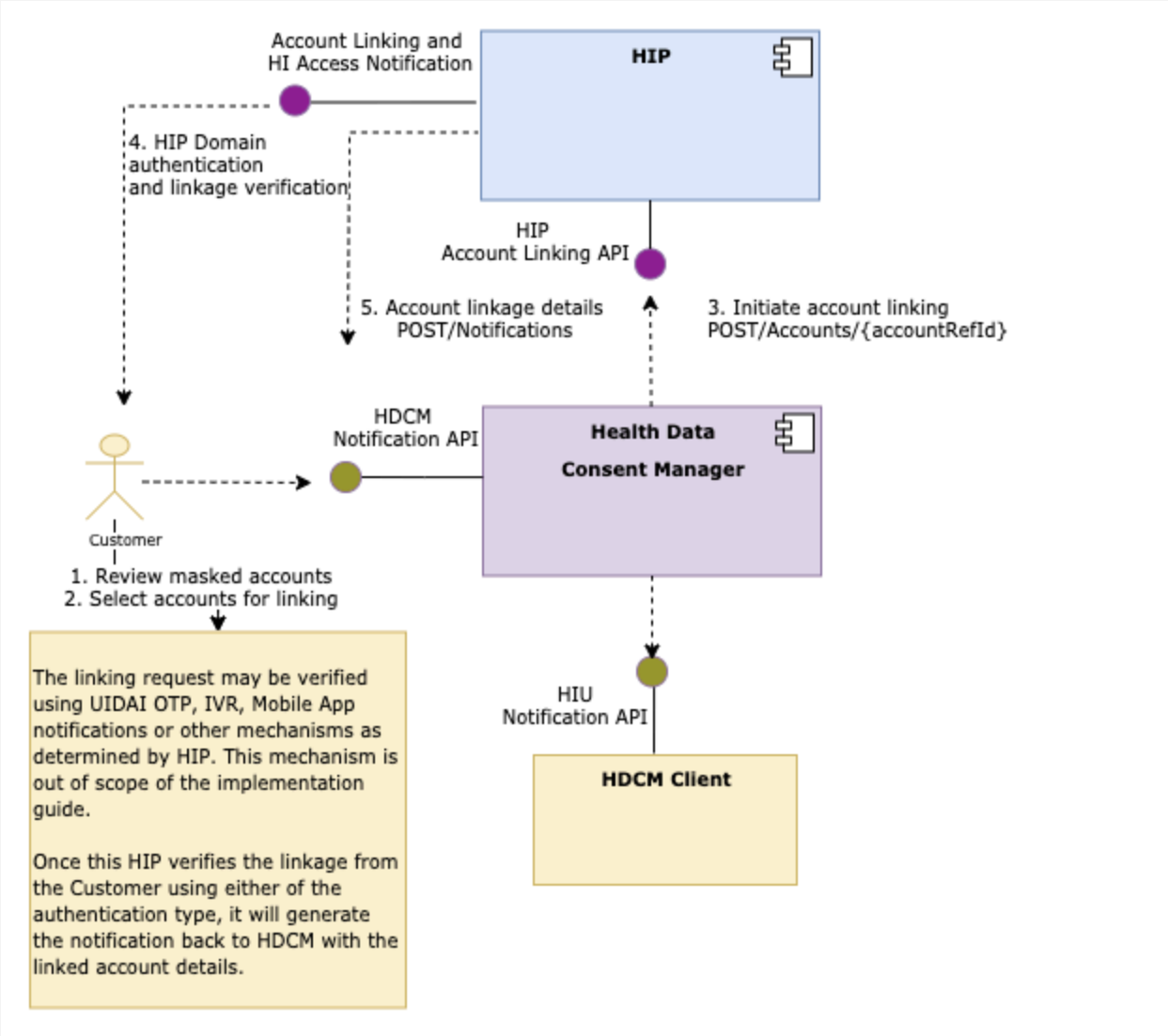
3.2 Link HIP Accounts at HDCM

This use case describes how a set of accounts are linked to the customer's HDCM account. Consent artefacts can only be created for accounts which the Customer has linked to their HDCM account. Customer authorization in the HIP Domain is required to complete the linking of the accounts.

There are two kinds of authenticators that the HIP may support:

1. **HIP Direct Authenticator**: The HIP authenticates the customer by directly interacting with him.
2. **Token-based Authenticator**: In this case, the HIP issues a token (e.g., an OTP) to the customer, which the customer then supplies to the HDCM for subsequent forwarding back to the HIP. This provides a confirmation to the HIP that the customer has approved the linking request to HDCM.

The diagram below illustrates this use case.



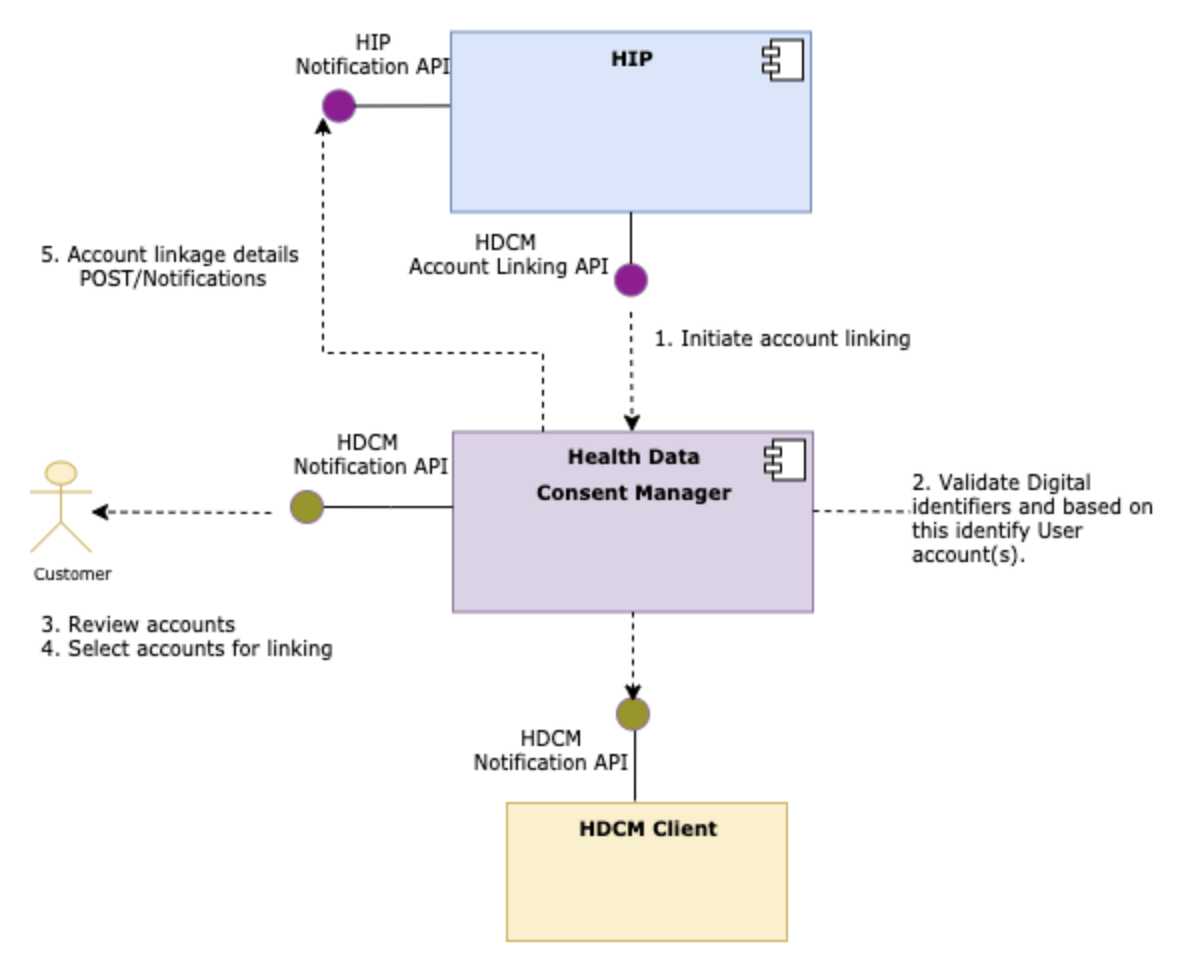
Once this linking is successfully completed (i.e. the User is authenticated by the HIP), the **accountNumber** and **accountType** are made available to the HDCM.

3.2.1.1 Link Account (Pull-based Discovery)

Summary	A customer initiates the account linking requests from the HDCM Client.
Pre-condition	• The customer has an account at the HDCM • A verified mobile and/or email address may be required for enabling OTP messaging from the HIP

	• The discovery of customer's accounts has already happened
Actor	Customer
Main Flow	**Token-based Authenticator** 1. The Customer has a masked list of accounts to initiate the account linking process in the HDCM Client 2. The Customer selects the masked accounts to be linked in her profile. 3. The HDCM requests the HIP for these masked accounts to be linked. The **AccountRefNumber** obtained from the HIP in response to account discovery request is also included. 4. The HIP sends OTP details corresponding to specific accounts of the customer 5. The Customer enters the OTP on the HDCM Client. The HDCM Client forwards this to the HDCM. 6. The HDCM forwards this OTP detail back to HIP. 7. Once the HIP confirms the OTP it sends a response to the HDCM with the details of the linked account. 8. The HDCM updates the linked accounts details. 9. The use case ends.
Alternate Flow	**Direct Authenticator** 1. The Customer has a masked list of accounts to initiate the account linking process in the HDCM Client 2. The Customer selects the masked accounts to be linked in her profile. 3. The HDCM requests the HIP for the masked account(s) to be linked. The **AccountRefNumber** obtained from the HIP in response to account discovery request is also included. 4. Customer authentication and verification happens in the HIP domain. 5. Once the HIP verifies the link account request with the Customer, the HIP sends a notification to the HDCM with the details of the linked account. 6. The HDCM updates the linked accounts details. 7. The customer is presented with the updated linked profiles. 8. The use case ends.
Post-condition	Once the HIP account linkages are established for the customer, the HDCM will enable consent artefacts creation for these linked accounts.
Notes	Customer's User Credentials based authentication MUST use Direct Authentication Flow to authenticate the Customer.
Security and Audit Requirements	All account linking activities should be logged. The linkages should be established based on explicit interaction between the HIP and the customer.

3.2.1.2 Link Account (Push-based Discovery)



Summary	An HIP initiates the account linking request using the customer details.
Pre-condition	• The customer has an account at the HDCM • A verified mobile and/or email address may be required for enabling deep link messaging from the HIP • The HIP has verified customer HDCM ID and/ or customer contact details
Actor	HIP
Main Flow	**HIP sends a linkage request directly to HDCM** 1. The HIP sends a link request containing a list of accounts to the Customer HDCM using the Customer HDCM ID. 2. The HDCM verifies the identity of the HIP sending the request. 3. The HDCM forwards this request to the HDCM Client. The HDCM

	Client notifies the Customer of the link request. 4. The Customer selects the accounts to be linked in her profile. 5. The HDCM updates the linked accounts details. 6. The customer is presented with the updated linked profiles. 7. The use case ends.
Alternate Flow	**HIP uses a deep linking method to send linkage request** 1. The HIP sends a deep link request containing a list of accounts to the Customer using the Customer's verified mobile number. 2. The Customer clicks on the link. It resolves in the HDCM Client domain. Alternatively, if the Customer does not have an HDCM Client account, she is asked to sign up with one. 3. The HDCM verifies the identity of the HIP sending the request. 4. If the link does not contain HDCM ID, HDCM also authenticates the Customer. 5. The HDCM forwards this request to the HDCM Client. The HDCM Client notifies the Customer of the link request. 6. The Customer selects the accounts to be linked in her profile. 7. The HDCM updates the linked accounts details. 8. The customer is presented with the updated linked profiles. 9. The use case ends.
Post-condition	Once the HIP account linkages are established for the customer, the HDCM will enable consent artefacts creation for these linked accounts.
Notes	Customer's User Credentials based authentication MUST use Direct Authentication Flow to authenticate the Customer.
Security and Audit Requirements	All account linking activities should be logged. The linkages should be established based on explicit interaction between the HIP and the customer.

3.2.2 Delink Account

Account delinking can happen in one of the two ways:

- Account delinking initiated at HDCM
- Account delinking at HIP, such as in case of customer request to HIP, account closure and other internal events in the HIP domain.

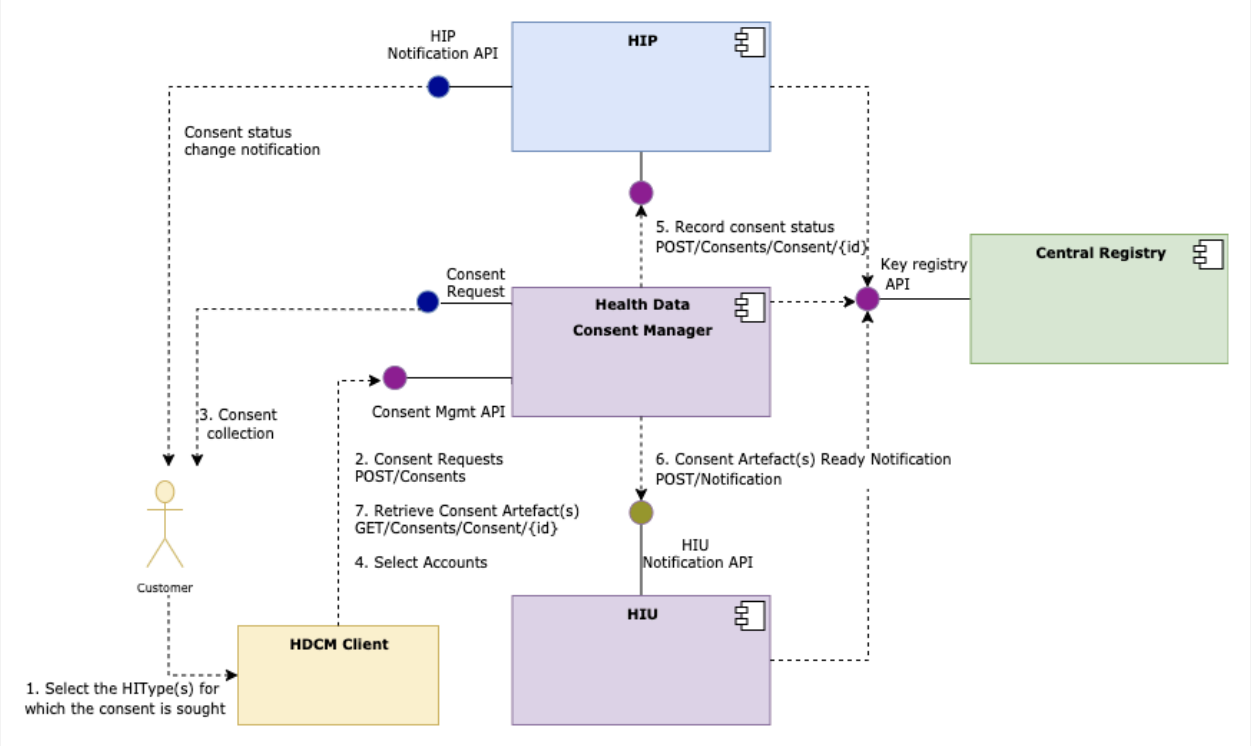
When the linked account is delinked, all the Consent Artefacts associated with this linked account are revoked and HIU and HIP are notified. The delinking action is performed on the HDCM Client.

Summary	Customer, using the HDCM Client, initiates the HIP delinking request.
Pre-condition	● The customer has an account at the HDCM

	• The customer's email and mobile are verified by the HDCM • The customer is able to provide an identifier(s) that can help the HIP identify the account associated with the customer. This could be Patient ID, Aadhaar, PAN or the unique account number. • The discovery of customer accounts has happened and some of the discovered accounts have been linked.
Actor	Customer
Main Flow	1. The customer initiates the account de-linking process in the HDCM Client 2. The customer selects the account(s) to be de-linked in her profile. 3. The HDCM Client notifies HIP for these delinked accounts by calling the **DELETE Accounts/link** API 4. The HDCM updates the de-linked accounts details. 5. The use case ends.
Alternate Flows	The customer requests account delinking at HIP In this case, once the HIP updates its record, the HIP sends a notification to the HDCM and HDCM proceeds with account delinking. HDCM will notify the customer of such a change.
Post-condition	The HDCM must revoke the consents previously created against this account. New consents can't be generated for this de-linked account.
Security and Audit Requirements	All account de-linking activities should be logged. The OTP must be sent to a verified address (verified mobile number or the email address). All records of prior consent artefacts generated against this linked account must be maintained in compliance with the IT Act.

3.3 Create Consent Artefact(s)

The customer's Consent Artefact is important for enabling request of HI from the HIPs. HDCM or HIU can only request for HI from HIP once they have obtained consent from the customer in the form of a consent artefact. HDCM is the Consent Manager and manages the entire lifecycle of the consents. This use case illustrates the mechanism for obtaining consent based on a customer interaction with the HDCM Client. Consent creation may also be initiated by the HIU (instead of the HDCM client) in a manner similar to what is described below.



The HDCM acts as the Consent Manager in the health information exchange architecture and interacts with the customer to obtain consent. The HIP is the consumer of the consent and provides information based on the Consent Artefact submitted to it. Furthermore, the HIP maintains the most up to date status of the pertinent Consent Artefacts.

The Consent Generation Request may include **HIType(s)** and other filtering criteria to improve the account selection on the HDCM for Consent Artefact(s) generation.

Summary	In order to perform the account aggregation for the customer, the HIU or the HDCM Client will request the HDCM Consent Flow API method "**POST /Consents**" to generate a digitally signed consent artefact. The entity requesting the information (HIU or HDCM Client) is referred to as the **Requester**. This use case defines the mechanism for obtaining consent from the customer and creation of the Consent Artefact(s). Valid Consent Artefact(s) are required to request for Health Information from the HIPs.
Pre-condition	• The customer has created her Customer Address with the HDCM • The customer's HIP account linkage has been performed on the HDCM
Actor	Customer

Main Flow	1. The customer selects the **HIType(s)** for which Consent is sought. The HDCM client either prefills a cached "**Customer Address**" or asks the customer to enter it. 2. Based on the selection of the **HIType(s)**, the HDCM client makes a request to the HDCM for the Consent Artefact(s) specifying the purpose for which the consent is sought using the "**POST /Consents**" API call on the HDCM Consent Management interface. 3. The HDCM initiates a Consent collection process 4. The customer selects the linked accounts for which the consents can be generated. 5. The HDCM generates the consent artefact(s) 6. The HDCM may generate a consent completed notification back to the requester (in the case of the HIU, this would involve using the "**POST /Notification**" API) 7. The HIU may then retrieve the consent artefacts from the HDCM using the "**GET /Consents/Consent/{id}**". 8. The use case ends.
Post-condition	Consent artefact(s) are generated.
Notes	The HDCM client or HIU can request the consent artefact(s) using the "**GET /Consents/Consent/{id}**" API on the HDCM. The HDCM must validate the HIU / HDCM client and only respond with consent artefact that corresponds to these entities. The "**GET /Consents/{consentRefNumber}**" method provides a mechanism for the HDCM client (or HIU) to obtain the status of the Consents Artefact(s) id generated in the "Consent Creation" request.
Security and Audit Requirements	1. The Consent Artefact(s) are digitally signed. 2. The HIP maintains the status (active, revoked, etc) of the Consent Artefacts. 3. All Consent creation requested must be logged.

3.4 Request Health Information

HIP must provide the requested information to the HDCM only after validating the consent artefact, which involves verifying the embedded signature and checking for timeliness of the request. Since the requested health information is time-dependent in nature, the request must specify the time instant for which information is being requested. This time instant must either be the same as the current time or an instant in the past. For example a request for information made at 12pm 31st October, 2018 may ask for information as it stands at that time, or it could ask for information as it existed at a past instant, say at 12pm, 30th October, 2018. This time instant for which information is requested is referred to as the ***capture time***. The HIP is expected to furnish information as it existed at the capture time, assuming that

the capture time specified in the request is the same as the time at which the request is received by the HIP or is prior to the time at which the request is received.

Information needs to be provided by the HIP to the HDCM in a timely manner in accordance with the SLAs set up at the time of the contractual agreement.

Consent Artefact Verification and Validation

Consent Artefact must be verified when HIP receives it from HDCM. Verification steps consist of:

1. Verify the credentials of the HDCM
2. Verify Consent Artefact is associated with the authorised customer
3. Verify that the Artefact is valid, is in active state and contains all mandatory information

Consent Artefact is verified if and only if all the verification steps are passed.

Consent Artefact must be validated every time information is requested based on the consent. The artefact is deemed valid if:

1. The capture time given in the request lies between the creation time (createtime) and the expiry time (expiretime) of the artefact and is either the same as the time at which the request is received or is prior to the time at which the request is received.
2. Consent has not been revoked based on the Revocation List maintained by the HIP (details of which have been specified in the section on Consent Revocation)

Types of Information Transfer

Information can be transferred from the HIP to the information requester (HIU or the HDCM client) in two ways:

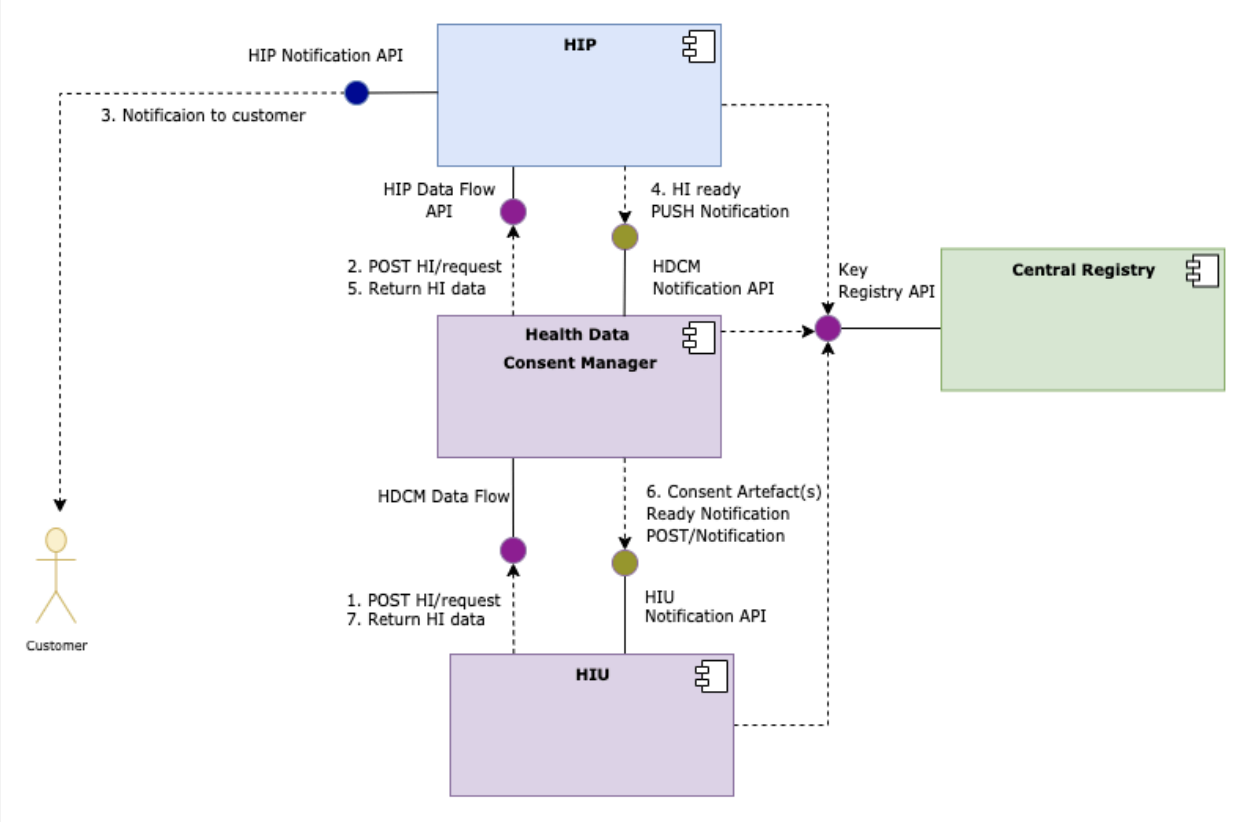
- **HDCM-mediated transfer**, where the information is first transferred from the HIP to the HDCM and then from the HDCM to the requester. In this mode, the HIP is not informed about the requestor, which ensures privacy of the requestor wrt the HIP.
- **Direct transfer**, where the information is transferred directly from the HIP to the requester. In this mode, the information request comes via the HDCM (e.g., from the HIU to HDCM and then from HDCM to HIP)

Furthermore, information flows can be categorized into two types based on the periodicity of the transfer. Two types of information flows are possible:

- **One Time**, where a customer requests for information to be shared exactly once.
- **Periodic**, where a customer requests for periodic sharing of information.

In all the above cases, the requester is allowed to make multiple information requests with the same capture time and the same consent artefact. This may be required to compensate for any loss of information previously transmitted to the requesting entity. As long as the consent artefact is valid, the HIP must respond to the information request.

3.4.1 HDCM-mediated one-time transfer

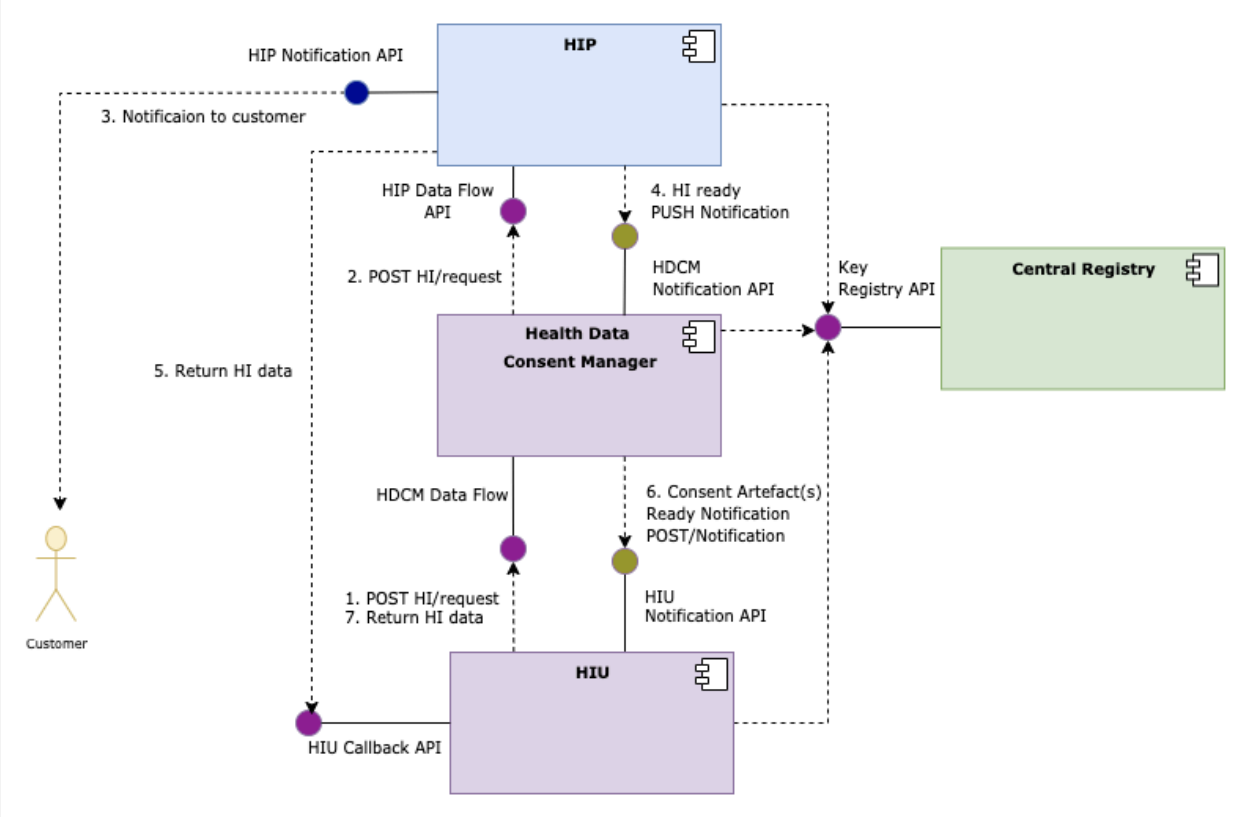


Summary	The HIU (or HDCM client) requests for health information once. The below description is for the case where HIU is the information requester. The flow for HDCM-Client requesting information is similar.
Pre-condition	• The customer has linked her respective HIP accounts as described in use case 3.2. • The consent artefacts for the information request has been created
Actor	Customer
Main Flow	1. The HIU makes a request for health information using identifiers of previously created Consent Artefact(s) maintained in the customer's profile. HIU includes the encryption parameters and the target recipient in the request using the "**POST /HI/request**" API. 2. The HDCM verifies the Consent Artefact(s) provided by the HIU: a. Verifies the status of the Consent Artefact(s) and their validity.

	b. Verifies that all accounts are still linked for the customer. For accounts that are no longer linked, revokes the Consent Artefact(s) and notifies the HIU and HIP. 3. For each HIP from which health information needs to be aggregated a. The HDCM constructs the health information request, including the Consent Artefact and the encryption parameters and makes a request to the HIP, with a callback notification URL. b. The HIP verifies the Consent Artefact: i. Verifies the artefact content as discussed above ii. Extracts and constructs the encryption key for responding to the health information request c. If the verification at HIP fails, goto **Alt.1** use case. d. The HIP constructs the health information, encrypts it (as discussed later in the security requirements section) and notifies the HDCM. 4. The HDCM fetches the encrypted health information from the HIP upon notification. 5. The HDCM tracks status from all HIP(s) from which it is requesting aggregation. 6. During the aggregation process, the HDCM notifies the HIU about the status update on the health information collection from various HIP(s). 7. Once requests from all HIP(s) is completed (either success or failure), it constructs a response notification back to the HIU with details to fetch the health information 8. The HIU upon receiving the aggregation completion request, fetches the health information from the HDCM. 9. The use case ends.
Alternate Flows	**Alt.1**: Verification at HIP fails 1. The HDCM is notified about the failure. 2. The HDCM records this in the audit log. 3. Depending upon the error conditions, the HDCM may revoke and archive the Consent Artefact. 4. The HIU is notified
Post-condition	● The HIU has obtained the aggregated health information. ● The HDCM purges the aggregated health information once the request is complete.
Notes	Real-time response from HIP is expected. The asynchronous responses will have timeouts.

In the situation where HDCM Client requests for the information (and not the HIU), the following audit requirement exists: The HDCM should not generate cryptographic keys on their environment for fetching the HI data; keys must be generated and stored by the HDCM client only. In essence, the HDCM client must run in an environment completely separate from that of the HDCM.

3.4.2 Direct one-time transfer



Summary	The HIU requests for health information directly from HIP
Pre-condition	• The customer has linked her respective HIP accounts as described in use case 3.2. • The consent artefacts for the request has been created
Actor	Customer
Main Flow	1. The HIU makes a request for health information to the HDCM using identifiers of previously created consent artefact(s) maintained in the customer's profile. The HIU includes the encryption parameters, and a **callback data-push URL** in the request using the "**POST**

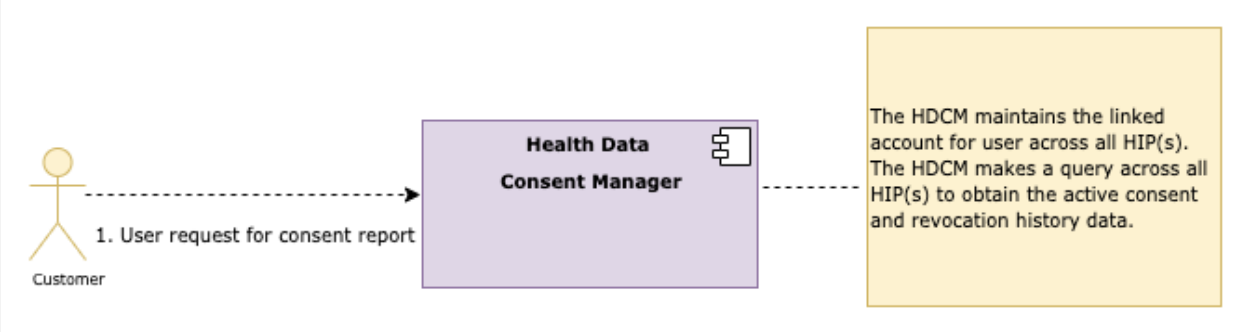
	/HI/request" API. This URL is where the information needs to be pushed by the HIP. 2. The HDCM verifies the artefact(s) provided by the HDCM client: a. Verifies the artefact(s) as before b. Verifies that all accounts are still linked for the customer. For accounts that are no longer linked, revokes the Consent Artefact(s) and notifies the HIU and HIP. 3. For each HIP from which health information needs to be aggregated a. The HDCM constructs the health information request, including the Consent Artefact and the encryption parameters and makes a request to the HIP, with a callback notification URL (meant to notify the HDCM about the availability of information) and a **callback data-push URL** (where the information needs to be pushed by the HIP). The data-transfer URL should be in the HIUs domain. b. The HIP verifies the Consent Artefact: i. Verifies the artefact content as discussed above ii. Extracts and constructs the encryption key for responding to the health information request c. If the verification at HIP fails, goto **Alt.1** use case. d. The HIP constructs the health information, encrypts it and pushes it to the data-push URL specified in the request. It also notifies the HDCM at the callback notification URL. e. Optionally, in case of high volume data, the HIP can also include a callback data-pull URL as part of the information push to the HIU. The HIU can then pull the information from this callback data-pull URL, which should be in the HIP domain. 4. The HDCM tracks status from all HIP(s) from which it is requesting aggregation. 5. The use case ends.
Alternate Flows	**Alt.1**: Verification at HIP fails 1. The HDCM is notified about the failure. 2. The HDCM records this in the audit log. 3. Depending upon the error conditions, the HDCM may revoke and archive the Consent Artefact. 4. The HIU is notified
Post-condition	The HIU has obtained the aggregated health Information.
Notes	Real-time response from HIP is expected. Asynchronous responses will have timeouts.

3.4.3 Periodic transfer

Summary	The HDCM will support periodic aggregation of the HI based on created consents and push an HI-READY notification to HIU or directly to an app installed in the Customer's environment. The HIU or the application can then fetch the HI. This use case defines the flow for this periodic fetch. This HDCM initiated push notification enhances the Customer experience and enables the customer to obtain the HI data based on a configured schedule.
Pre-condition	● Account linking profile is available on the HDCM ● Consents for periodic HI requests have been created ● Consent records status are maintained at HIP ● The Customer Address of the customer has been configured
Actor	HDCM
Main Flow	1. The HDCM initiates a pre-scheduled HI request based on set of keys received from the HDCM Client app or the HIP app and the corresponding consent artefacts 2. For each HIP from which health information needs to be aggregated a. The HDCM constructs the health information request, including the Consent Artefact and the encryption parameters previously sent by the HIU and makes a request to the HIP, with a callback notification URL. b. The HIP verifies the Consent Artefact: i. Verifies the Digital Signature of the Consent ii. Verifies the integrity of the Consent Artefact iii. Validates the content of the Consent Artefact against the linked account reference. iv. Extracts and construct the encryption key for responding to the health information request v. Verifies the validity time of the Consent Artefact vi. Verifies that Consent has not been revoked. c. If the verification at HIP fails, goto **Alt.1** use case. d. The HIP constructs the health information, encrypts it using the "encryption key" and notifies HDCM about HI data availability. 3. The HDCM tracks status from all HIP(s) from which it is requesting aggregation.

	4. During the aggregation process, the HDCM notifies HIU about the status update on the health information collection from various HIP(s). 5. Once requests from all HIP(s) are completed (either success or failure), it constructs a response notification back to the HIU with details to fetch the health information, so aggregated. 6. The HIU upon receiving the aggregation completion request, fetches the health information from HDCM. 7. HIU builds the aggregated health information and notifies the customer about the availability of the health information. 8. The use case ends.
Alternate Flows	**Alt.1**: Verification at HIP fails 1. The HDCM is notified about the failure. 2. The HDCM records this in the audit log. 3. Depending upon the error conditions, the HDCM may revoke or discard the consent. 4. The HIU is notified
Post-condition	After the data is fetched by User/HIU the data will be purged by HDCM
Note	HDCM is never supposed to generate the cryptographic primitives and the encryption keys in their environment although the cryptographic primitives for the HI encryption can be generated in the customer's environment by an HDCM Client.

3.5 Consent History

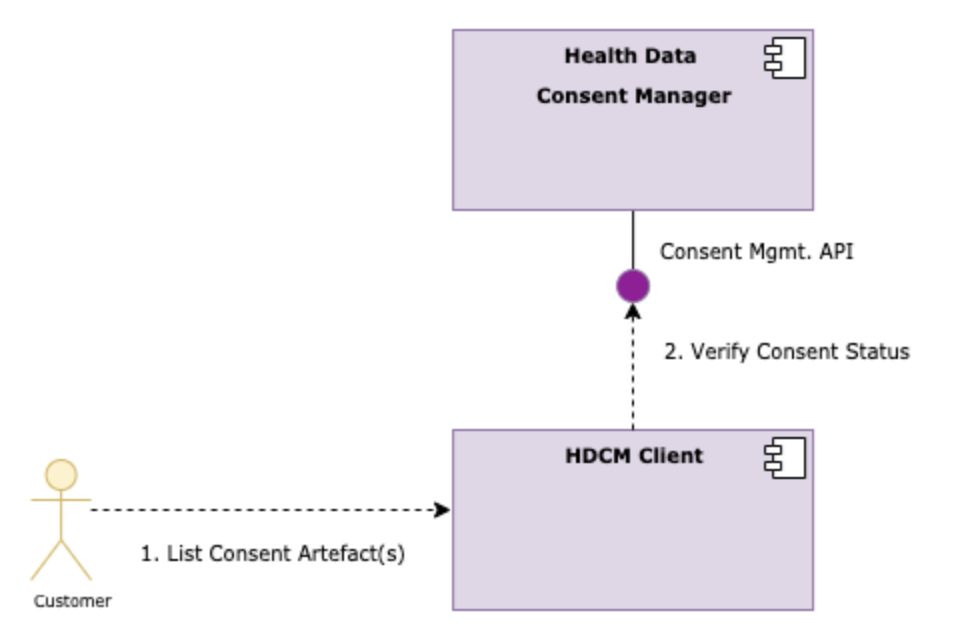


Summary	The customers may be interacting with multiple HIUs and HDCMs for Consent Artefact(s) creation. In this case it is important to provide customers with the ability to obtain a comprehensive list of all Consent Artefacts

	associated with her profile. This use case enables a comprehensive list of Consent Artefact(s) to be presented to the customer. Revoked and paused Consent History will also be made available to the customer.
Pre-condition	● Account linking profile is available on the HDCM ● The Customer Address of the customer has been configured
Actor	Customer
Main Flow	1. The customer logs into the HDCM client 2. The customer requests for the consent log from the HDCM 3. The HDCM retrieves the customer's profile from its datastore 4. The HDCM presents the list of artefacts to the customer. 5. The use case ends.
Alternate Flows	**Alt.1**: HIP consents log: Customer uses the HIP application to see the consent that it maintains for the Customer.
Post-condition	History of all consents that have been created for the customer is displayed.

3.5.1 Alt.1: HDCM client consents log

The HDCM client specific maintained Consent Artefacts are displayed to the customer when using the application that has HDCM client functionality. This may be a subset of all consent that may exist for the customer across multiple HDCM clients that the customer might be interacting with, but presents a comprehensive list of all Consent related to the specific HDCM client. The following diagram shows the use case operation.

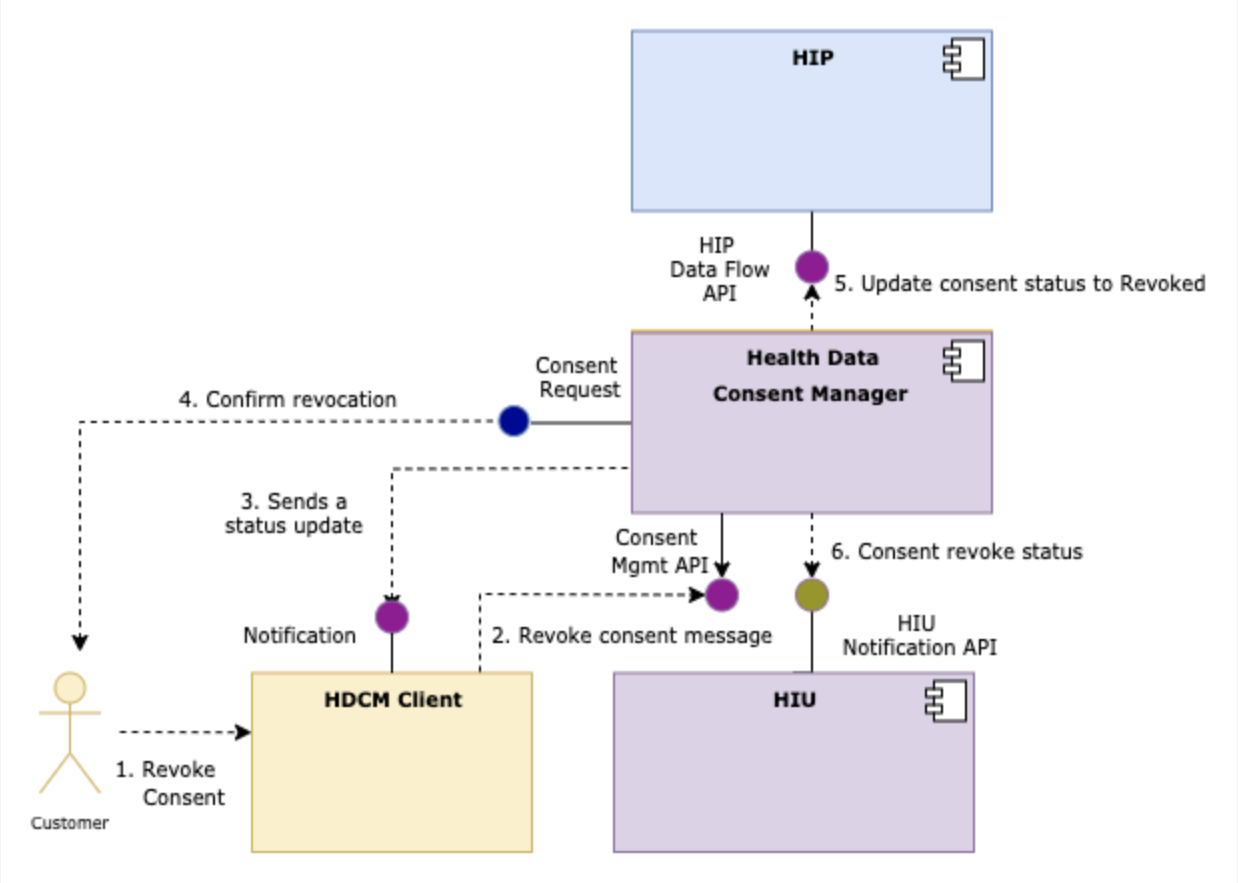


Summary	The HDCM client application may also provide the Consent History of the specific Consents that were generated from this application. This use case define the mechanisms for the HDCM client application to provide a listing of the Consents. Since the customer may use multiple HDCM client applications for different services, the HDCM client based “Consent History” will not be comprehensive list but may just be a subset. A comprehensive listing of the “Consents” may only be provided via the HDCM. This is an optional feature the HDCM client application may offer.
Pre-condition	● The customer is logged into the HDCM client application ● The “Consent Artefact(s)” for the customer have previously been generated on the HIU application
Actor	Customer
Main Flow	1. The customer requests consent history on the HIU application. 2. The HDCM client application loads the “Consent History” 3. It may verify the Consent Status for each of the Consent with the HDCM. 4. The HDCM client application displays the verified “Consent History” to the customer. 5. The use case ends.
Alternate Flows	None
Post-condition	Consent History of the Consent generated in the HDCM client domain is displayed to the customer within the HDCM client.

Note	

3.6 Revoke Consent

The HDCM enables the consent lifecycle management. An important aspect of the Consent Management is the Revoke, Pause and Resume the state of the consents. These state updates happen in the HDCM domain. In certain cases, for example, when a customer deletes their account with HIU, the HIU will initiate the Revocation of all consents associated with the customer's profile at HDCM.



Summary	The Consent Artefact(s) once created may be revoked by the customer. The revocation status is maintained by the HDCM and a notification sent to HIP.
Pre-condition	• The Consents have been created for the customer • The customer is using the HDCM client.
Actor	Customer
Main Flow	1. The customer selects option to revoke consent. 2. The customer selects the consents to be revoked.

	3. For each consent artefact that is revoked, the HDCM notifies the corresponding HIP about the Consent Revocation. 4. The HIP records the Consent's status as revoked and responds to HDCM. (It's important that HIPs maintain a list of all revoked artefact-IDs.) 5. The HDCM generates a notification to the HIU (as specified in the consent artefact) specifying that the Consent has been revoked. 6. The use case ends.
Alternate Flows	**Alt.1**: Revocation upon HDCM account deletion When a customer's HDCM account is deleted, all consents related to the customer must be revoked. The HDCM must trigger consent revocation. **Alt.2**: Revocation at HIP An HIP may enable revocation of consent artefacts linked to a customer that has that HIP listed. It can do this only upon customer request. In this case the HIP will send a revocation notification to the HDCM.
Post-condition	The HIPs have recorded the revoked consent artefacts as revoked in their records. These artefacts can't be used for requesting health information from the HIP in the future.
Security and Audit Requirement	Consent revocation calls will be logged for Audit trails.
Notes	The revocation notification is sent to both HIP and HIU.

4 Health Information

The API specifications deals with encrypted data. The cryptographic primitives required for encrypting the data and the structure within which the encrypted and signed data is exchanged is generic and is agnostic of the representation of the data. This makes the API agnostic to the HI document structure and enables addition and evolution of health information representation independent of the API versioning. It is however important that well defined data dictionary and standardized vocabulary is established for a common understanding of data representation between HIP and HIU/customer for an efficient exchange. This section provides primitives that will be used for exchanging customer's health information between HIP and HDCM/HIU. A standard list of health document formats that must be made available to the customer based on the customer's consent is defined separately.

4.1 Data Schemas for specific HI

The Health Information is delivered to HIU and customer as encrypted data. The HIU/HDCM client decrypts this data and processes it. This section defines a set of common data identifier constructs that the system will use. A list of supported HI schemas will be provided separately from this specification.

4.2 Document Specifications for Health Information

The response containing health information to a verified and valid consent request must contain HIP ID, Health Information Type, and Document ID as part of the response. Further attributes like Timestamp, Consent Artefact ID, etc must also be included in the header.

- **HIP ID (hipId):** Each HIP will be provided with a unique identification (HIP ID) by NCG so that all documents provided by them are accessible. It is likely that the list of unique codes will be derived via the HIP's domain URL whenever available and be published by NCG with the ability to add new HIPs on a need basis. When URL is not available for a department, a unique (alpha) code may be assigned. These codes must be unique across India.

- **Health Information Type (HIType):** NCG will decide and define the comprehensive list of Health Information Types using pure alpha case-insensitive strings.

- **Session ID** (**sessionId**): All HI documents are generated against a session identifier.

4.3 Unsupported HI types

The system will support new health information types which have not formally been defined in the specifications.

In a situation, where an HIP integrates into the system, while a formal schema for a new **HIType** is not specified or not possible to specify, the document can still be delivered into the data section, with a referenced to a schema.

The schema could be embedded in the document with reference to the namespace.

5 Non Functional Requirements

5.1 Reliability Considerations

The systems must be designed to be highly available, scalable using a distributed architecture for vertical and horizontal scale, and high on performance.

The APIs must have high uptime and a public API Status Page must be provided by the HDCMs and HIPs that reports the same for each of the endpoints (along with other open data like average response time, latency, etc). Furthermore, the HDCMs and HIPs must implement the Heartbeat API for reporting their system uptime in real-time.

5.1.1 Failure Scenarios

This section explains how the various failure scenarios must be handled:

- Failure to Notify customer
    - In this scenario, when the Health Data Consent Manager or HIP is not able to notify the customer on the status of the consent flow or data flow, a mechanism has to be put in place to notify the customer at a later stage. This can be achieved by reinitiating the notification message to the customer or by providing the customer an option to check the status through an application, or by providing a list of all consent flows and data flows (with status) in the application.

- Response from HDCM does not reach HIU
    - In this scenario, when the response sent by HDCM does not reach HIU, the HIUs should have a mechanism provided by HDCM to initiate a request to know the status of the consent flows and data flows.

- Response from HDCM does not reach HIP
    - In this scenario, when the response sent by HDCM does not reach HIP, the HIPs should have a mechanism provided by HDCM to initiate a request to know the status of the consent flows and data flows.

- Response from HIP does not reach HDCM
    - In this scenario, when the response sent by HIP does not reach HDCM, HDCM will wait for the response till the timeout period. HIP may have a mechanism to resend the response within the timeout period. If HDCM does not receive the response within the timeout period, HDCM will timeout the request and respond to customer or HDCM client with a timeout response.

- HIP is not available to HDCM

- In this scenario, when the HIP is not available to HDCM, HDCM may have a mechanism to re-initiate the request to HIP.

- HDCM is not available to HIP:
  - In this scenario, when HDCM is not available to HIP, HIP may have a mechanism to re-initiate the request to HDCM.

- HDCM is not available to HIU
  - In this scenario, when HDCM is not available to HIU, HIU may have a mechanism to re-initiate the request to HDCM.

5.2 Security Considerations

5.2.1 Digital Identifiers

During the registration process with HDCM, the customer will need to establish a set of validated identifiers. These validated identifiers enables the HIP with the discovery of the accounts that they may have for the customer. The NIST Digital Identity Guidelines [13] may be used to select an appropriate assurance level for enrollment and identity proofing.

There are two categories of digital identifiers, depending upon the verification mechanisms :

Type	Examples	Description
Strong Identifiers	Mobile, Email, Aadhaar	These identifiers can be verified by the means of OTP or biometric authentication and are thus considered strong identifiers.
Ancillary functional Identifiers	Patient reference number, Case ID, Care context Number	There does not exist a mechanism to verify this. These identifiers must be verified along with the Strong Identifiers, i.e., a combination of Case ID and mobile number together is only verifiable in the HIP domain. The HIP should not respond only on the basis of Case ID but must verify the other Strong Identifier along with the Case ID before responding with the masked account list.

The HIP should respond with its status of the identifiers as well. If the KYC process is performed by the HIP, then they may report these identifiers as “Strong Identifiers”.

Upon registration, the customer also creates a Customer Address [see section 2.3.1] and establishes a Consent Approval PIN.

5.2.2 Customer Authentication between HDCM and HIP

During the Account Linking process, the customer needs to be authenticated with the HIP. The purpose of this authentication is to identify the customer who will authorize the linking of the accounts between HDCM and HIP. The authentication mechanism thus ensures the right person is authorising the account linkage.

For the purpose of linking, the customer may be authenticated through any of the suggested mechanisms:

- **Approach 1**: Authenticate in HIP Domain
  The customer is redirected to the HIP and authenticates themself by entering their account credentials on the HIP interface. This redirection should happen via deep linking of the HDCM and HIP mobile applications and must NOT make use of Webview technology (for embedding the HIP Website in the HDCM app).
- **Approach 2**: Authenticate using OTP
  The customer provides Account ID of HIP and receives a one-time password (OTP) via SMS from the Health Information Provider. The OTP SMS could be replaced by In-App Notification from HIP authorized App, TOTP, or Email. This code is then used to authenticate the customer.

There are multiple approaches to authentication and the appropriate one should be chosen based on interoperability across Health Data Access Fiduciaries and the Health Information Providers, ease of implementation, security of User credentials and best possible User experience.

5.2.3 Using a secondary PIN by customer in the Consent flow

When using the Consent Approval process on the HDCM screen, the customer must enter the Consent Approval PIN. The HDCM should allow authenticated customers to change/reset the Consent Approval PIN.

5.2.4 Guidelines for API Security

Below we provide a list of security guidelines that must be followed for securing the Health Data Consent Manager APIs. Good references for learning about managing security in REST APIs are:

- The OWASP (Open Web Application Security Project) REST Security Cheat Sheet: https://www.owasp.org/index.php/REST_Security_Cheat_Sheet
- Top 5 REST API Security Guidelines by DZone.com: https://dzone.com/articles/top-5-rest-api-security-guidelines

Here are our guidelines:

1. **Enforce HTTPS only:** All Health Data Consent Manager APIs must enforce the use of TLS 1.1 or above in API calls and responses. This will ensure security and integrity of the health information that is shared by Health Data Access Fiduciaries with requesting entities.

2. **API Keys:** API Keys should be used as the first line of defense against unauthorized API calls. API keys need to be set up before HIUs or HIPs can start making API requests to HDCMs.
3. **Digitally Sign API Requests:** All API requests must be digitally signed using industry-grade signature algorithms. Here are some guidelines to use when generating digital signatures in API requests
    a. Use latest cryptographic algorithms: 2048-bit RSA based signatures (e.g., RSA-PSS) or the latest elliptic curve based digital signatures (e.g., ECDSA) in groups which provide at least 256 bits of security.
    b. Use SHA-256 for hashing the requests when generating signatures. Make sure to hash query parameters in the URL as well as in the body of the API request (for PUT/POST APIs). Make sure to also include API keys as part of the content to be hashed.
    c. Use JSON Web Tokens or the W3C Signature Syntax for XML signatures to embed signatures in API requests. The digital signature should be embedded as a query parameter in the URL.
    d. Use OCSP Stapling for including public-key certificate chain information in API calls. OCSP is described here: http://www.entrust.net/knowledge-base/technote.cfm?tn=70825
4. **Digitally Sign API Responses:** The use of TLS V1.1 and above will ensure that API responses are sent in an encrypted and signed manner. For added security, it is recommended that HDCMs use public-key based digital signatures to sign responses. The same guidelines as in the case of API requests will apply.
5. **Rate limiting API calls:** In order to prevent denial of service attacks, API requests should be rate-limited by HDCMs. Appropriate error codes should be provided in responses in situations of request overload.
6. **Use validation methods:** Standard practices around input validations should be applied in order to prevent injection and other attacks. These include the following.
    a. Input Validation techniques
    b. URL Validation techniques
    c. Validate incoming content-types
    d. Validate response types
    More information on validations is available in the OWASP REST Security Cheat Sheet.
7. **Use Output Encoding:**
    a. Security Headers
    b. JSON Encoding
    c. XML encoding
8. **Use HTTP response codes appropriately:** See examples in the OWASP REST Security Cheat Sheet for this.
9. **Maintain audit logs:** All API requests and responses should be logged locally by the HDCMs to ease detection of security threats and attacks.

Bug Bounty Programs may be organised to encourage the white-hat community to constantly monitor and proactively report possible threats to the HDCM APIs that could then be fixed leading to strengthened systems.

5.2.5 Customer Management and Customer Protection

Given the sensitivity of health information, HIUs, HDCMs, and HIPs must put in place appropriate customer management and customer protection mechanisms:

- Customers must have provisions to lock and unlock their HIP and HDCM accounts for consented data access.
- Two Factor Authentication for customers of the HDCM is mandatory for account linkage.

5.2.6 Fraud Detection and Analysis

The HDCMs and HIPs should put in appropriate monitoring mechanisms for Fraud Detection and Analysis. Logging is essential as that ensures traceability across systems. For example, unusual access patterns, too frequent access, over consenting, consent fatigue, and other such anomalies must be identified on a proactive basis.

5.2.7 Anonymity of HIUs when requesting information from HIPs

HDCM should not reveal the identity of the HIUs to HIPs in the process of requesting information from the HIPs. This is accomplished by the manner in which consent artefacts are generated in response to consent creation requests: for each request, two sets of artefacts are generated, one to support the flow of information from HIPs to HDCMs (these artefacts have no information about the HIUs that are requesting information) and the other to support the flow of information from HDCMs to HIUs (these artefacts enable *specific* HIUs to fetch information from the HDCMs, and thus have information about those HIUs). It is because of this separation in permissioning and data flows that HIPs are unable to learn the identities of HIUs who are requesting data from them.

5.2.8 Encryption of Health Information

5.2.8.1 Data In-Flight Encryption

Information shared as part of the data flow will be secured using an encryption mechanism that ensures perfect forward secrecy. This means that even if any of the key materials stored at HIPs, HIUs or HDCM Clients (either long-term private keys or session keys) are compromised at a given point in time, data that was exchanged in the past (i.e. before that point in time) would not be possible to decipher. This is a strong guarantee of secrecy which is necessary to ensure health data.

We describe the mechanism here; corresponding APIs are in the appendix. The mechanism uses Diffie-Hellman Key Exchange (DHE). DHE is used in many Internet protocols (like SSH and TLS) for establishing shared secret keys between remote parties.

Encryption for HDCM Client

1. When making the request for data, HIU picks a set of Diffie-Hellman (DH) parameters, generates a DH key pair (dhsk(U), dhpk(U)) (which is a short-term public-private key pair) and generates a

32-byte random value, rand(U). It sends these values (public key and random value) to HDCM, along with the data request via a digitally-signed API call.

2. HDCM ensures that the data request is in keeping with the terms of the artefact and, if so, it forwards the request to the HIP, again via a digitally-signed API call.
3. HIP checks that the consent artefact is valid (as above), that the data being requested is in keeping with the terms of the artefact and if so, it generates a fresh DH public-private key pair in the same group as specified by the HIU ((dhsk(P), dhpk(P)) and also a 32-byte random value rand(P). Using dhpk(U) and dhsk(P), it computes a DH shared key dhk(U,P) and using (dhk(U,P), rand(U), rand(P)) as key material, it computes a 256-bit session key sk(U,P) which is used to encrypt the data sent from HIP to HIU. HIP sends the public key dhpk(P), the nonce rand(P) and the encrypted data (encryption of the Health Information to be sent under sk(U,P)) to the HDCM. To ensure integrity of the encrypted data, HIP also signs the entire payload (the encrypted information as well as the key materials i.e., the public key dhpk(P) and the value rand(P)) using its long-term private key before sending it to HDCM.
4. In order to generate the shared secret key, the dhpk(P) would be used. The dhpk(p) should be used with no headers. We will follow the big endian (network byte order) so the keys used to derive the shared secret remain consistent. Also as the cryptography deals with large numbers, do not attempt to convert any data to binary format until its first converted to big integer. Sample sets of libraries that can be used are as follows.
    a. Openssl
    b. Bouncy castle (ensure the X dhpk is a big integer)
    c. NaCl
    d. Libgcrypt
    e. PyNaCl
    f. TweetNaCl
5. Most of the libraries allow exporting public keys but they are compressed and encoded. So use the libraries to extract the uncompressed form of the public key that will start with a hex 04. The 32 bytes after the 04 is the X.
6. HDCM forwards the encrypted information and the key materials received to the HIU, after ensuring that the signature on these values is valid.

The DHE mechanism ensures that the shared key dhk(U,P) can also be computed at the other end by the HIU using the values dhpk(P) (HIP's DH public key) and dhsk(U) (HIU's DH private key). For this reason, the HIP must accompany the encrypted data with dhpk(P) and rand(P) when sending it. All values must be digitally signed using HIP's long-term private key so that the HIU can verify the validity of the same.

At the end of the data flow, both HIU and HIP must delete all short-term key material that was generated in the process. This includes all DH key pairs, random nonces and the session key. This step is necessary for ensuring forward secrecy.

The session key is derived from the shared secret key as follows

i) XOR the nonces (rand(P) and rand(U))

ii) Take the first 20 bytes as the salt for HKDF
iii) Derive the key using HKDF (shared key, salt)

Now encrypt the data using AES GCM as follows

i) XOR the nonces (rand(P) and rand(U))
ii) Take the last 12 bytes as the IV for GCM
iii) Encrypt the data
iv) Generate the TAG (Some libraries would auto generate this)

Append 16 byte of TAG to the end of the encrypted data.

Encryption for HDCM application

The exact same flow would work except that all actions performed by the HIU would instead be performed by the HDCM client on the customer's phone. The following guidelines are to be followed in implementing the HDCM client, if the HDCM decides to offer the Health Information aggregation functionality:

1. DHE key pairs must be generated localling on the customer's phone and the private keys from the DHE key pairs must never be communicated to the HDCM server.
2. The decryption of the customer's data must also take place on the customer's phone and the decrypted data (customer's health information) must never be communicated to the HDCM server.
3. As stated above, the private keys must be deleted from the customer's phone at the end of the data flow.

The HDCM client should be implemented in a manner such that it can be easily verified (by an auditor) that the above three guidelines are followed.

Encryption for Periodic data access

For encrypting data in the case of periodic data access, HIUs and HIPs set up shared keys for a period of usage via a single exchange. A period is a series of multiple data accesses which is defined as follows: if the frequency of data access is WEEKLY or less frequent, the period should be set as 3 months; otherwise, the period should be set as frequency, multiplied by 12.

1. At the beginning of each period, HIU generates n key pairs (dhsk(U)[1], dhpk(U)[1]), … , (dhsk(U)[n], dhpk(U)[n]) where n is the number of data accesses in that period. It generates a random nonce rand(U) and sends the selected DH parameters, the value n, the n DH public keys dhpk(U)[1..n], the start-date and end-date of the period, the value rand(U) and the consent artefact to HDCM via a digitally-signed API call.
2. HDCM forwards the same, after validation, to HIP via a digitally-signed API call.
3. HIP stores the DH key pairs and rand(U) (and other fields).

For the “i”th instance of data access in a period (i ranging from 1 to n), HIP generates a fresh DH key pair (dhsk(P)[i], dhpk(P)[i]) and a random nonce rand(P)[i] and computes a DH shared key dhk(U,P)[i] and a session key sk(U,P)[i] using these values and the “i”th public key shared by HIU (i.e. dhpk(U)[i]) earlier. The data encryption and transmission then happens as usual, after which the DH keys dhsk(U)[i], dhsk(P)[i], and the session key sk(U,P) are deleted by the respective entities.

Choice of Diffie-Hellman parameters

DHE will be performed over Elliptic Curve Cryptography (ECC) groups. We recommend the use of Curve25519, which is used in DHE implementations in a lot of protocols like SSH and WhatsApp.

5.2.8.2 Data At-Rest Encryption

The customer data and metadata in HDCM must be encrypted using a symmetric key. These keys can be rotated from time to time and can be managed through a Key Management Service.

5.2.8.3 Controlling access to health information by Health Data Access Fiduciaries

Health information must only be accessible to the customer with whom the information is linked and Health Data Access Fiduciaries should not be allowed to view or store such information. Part of this objective is achieved via data in-flight encryption. Furthermore, if the Health Data Consent Manager provides an app (a mobile app or a desktop app) through which the customer requests for HI, certain measures must be in place when implementing this app:

- The app may receive health information from HIPs in encrypted form and decrypt such information, but it is not allowed to relay such information back over the network (e.g., it can communicate this information to a server maintained by the Health Data Consent Manager).
- The app must follow strong API security guidelines as outlined in this document

5.3 Audit Considerations

All events (consent created, consent revoked, data requested, data denied, data sent, etc) in the consent flow and data flow and corresponding system requests and responses between HIUs, HDCMs, and HIPs must be digitally signed and logged to ensure immutability, non tamperability, and non repudiability.

HDCMs, HIUs and HIPs need to persist the logs for certain period of time so that they can be retrieved when necessary and this audit trail must be made transparency available to the customer. To ensure integrity, logs cannot be edited - they can only be appended.

5.4 Privacy Considerations

5.4.1 Data Masking and Identity Protection Guidelines

There are three levels of information access:

- Standardised Lookups
- Standard Health Information Templates by Purpose

- Custom Health Information

There are three levels of identity visibility:

- Hidden (Not Visible)
- Masked (Partially Visible)
- Completely Visible

Here are a basic set of data masking and identity protection rules that need to be followed to ensure that privacy is maintained:

- Tokenization: Account numbers, card numbers, phone numbers, personal identifiers (PAN, Aadhaar, etc.) should be tokenized using Virtual IDs issued by the HIU, HDCM, and HIP (as defined in the Customer Identifier section of the Consent Artefact XML).
- Data Masking - Masking Out: For instance, only the last four digits of a credit card number involved in a transaction may be revealed - XXXX XXXX XXXX 1564. Data Masking may be static, on-the-fly, or dynamic.

Note: A comprehensive list of data masking and identity protection rules may be further decided by the respective FSR, after which a Data Masking and Identity Protection Guidelines Document for Health Information shall be published.

5.4.2 Data Portability Guidelines

The customer must be able to seamlessly transition from one HDCM to another by porting their HDCM account data and corresponding consent artefacts.

5.5 Consumer Experience Considerations

The user interfaces must be aesthetically designed, secure, honest, unobtrusive, understandable, useful, and an enabler of informed decisions.

The HDCM must provide mechanisms for customers to access the services via a desktop or laptop browser (web application), smartphone (mobile application), or feature phones. For ease of HIU integration, an SDK may be provided by the HDCM. New Customer Registration and subsequent linking of HIP accounts must take place in the environment of an HDCM (HDCM's mobile app, web app, etc).

The HDCM must provide a User-friendly interface for customers to access a record of the consents provided by them and the HIUs with whom the information has been shared, recurring consent artefacts, and the ability to revoke (permanent) or pause (temporary) a consent artefact.

5.6 Developer Experience Considerations

In order to ensure easy and seamless consumption of these APIs by developers for the purpose of development (integration), it's important that the developer experience for the APIs exposed by HIPs and HDCMs be given prime importance.

The APIs should be developer friendly and at the minimum the following considerations must be met:

- Developer Portal: It must be publicly accessible and must contain:
    - API Sandbox
    - API Documentation and Reference
    - Quickstart Guides
    - Open Source Libraries and SDKs

HIPs and HDCMs may further engage with developers via hackathons and other feedback channels like a Developer Forum or StackOverflow.

5.7 Grievance Redressal

There must be a dispute resolution mechanism in place by HIUs, HDCMs, and HIPs to route complaints digitally and redress grievances of the customer. This improves customer satisfaction and builds trust. Prompt and efficient service is essential to retaining existing customer relationships.

The customers may record their grievances / provide their feedback in writing, verbally, or digitally. SLAs must be put in place to ensure prompt response and necessary escalations in case of delays. Grievance Redressal can be further enabled through the use of detailed status and error messages in response to each request.

6 Applications of Health Information Exchange

Here are some examples of applications that can be realized using the consent manager framework for health information exchange:

6.1 Insurance

Data is critical to insurance-based healthcare delivery. Using the information exchange framework, insurance companies can get access to rich data in a much more seamless manner and use this to evaluate and adjudicate claims more rigorously, objectively and in an automated manner.

6.2 Medical consultations

Consider a patient who goes to Hospital A where he is diagnosed with cancer. The patient wants to get a second opinion from Hospital B, where the doctor could use the local EMR software to try to fetch patient data residing in Hospital A's lab information management system (LIMS). The latter may have already implemented APIs for enabling other hospitals to fetch data from its LIMS. The patient provides consent to Hospital B from within the EMR software being used there and the EMR uses the resulting consent artefact to request data from Hospital A's LIMS. Upon validating the consent, the patient's data is returned. The doctor at Hospital B can now recommend additional tests to the patient based on what is unavailable from the obtained data. This is a simple example which demonstrates how data can be exchanged between different care providers. This can also be extended to other applications where second opinions are sought online or through other AI-based technologies where patient history serves as the input to these applications.

6.3 Health locker

Entities that allow a patient to keep her copy of medical history can be yet another application of the consent framework. Imagine a health locker requesting STREAM-based consent from the customer for all HIPs at the time of linkage. In this way, any new data that is generated by the HIP for a particular customer can be streamed to the health locker, thus creating a single place where all health history of a particular customer can be found. This will be especially useful in cases where HIPs lose or remove storage of health information in their domains.

6.4 Drug trials

EHRs could also be useful to pharma companies in conducting drug trials. Suppose a pharma company wants to test the effectiveness of a new drug against a common disease. They could take a sample of patients suffering from the disease and collect their consent to be able to view their health records containing drug-specific information from *any* hospital or caregiver. This allows them to query their health data periodically and track the effectiveness of the new drug over time. Because of the nature of EHRs in our framework (dynamically generated, and from multiple sources), such information can be collected by the company from a wider range of providers and faster.

7 Appendix

7.1 Notifications

Appropriate Notifications have to be sent to all parties following any actions. Notification can be delivered through Email, SMS, Callback URLs, or In-app Notifications. Each notification can have different payloads and the payload of the event may vary based on the channel.

The following sections show the various account linking, consent and data lifecycle events for which the system will generate notification. The column M/O, specifies whether the customer should be sent a notification, where “M” means “Mandatory” and “O” means “Optional”. The other notifications to entities must be sent for auditing, logging and state management purposes.

7.1.1 Account Linking Lifecycle Events

Account Linking Lifecycle Events				
Notifications	**M/O**	**Sender**	**Receiver**	**Based on action performed by**
Account Discovery	O	HIP	customer	customer
Account Linked	M	HIP	customer	customer
Account Delinked	M	HIP	customer	customer

Static notification could be provided to the customers on their registered emails, mobile number or via in-app notifications.

7.1.2 Consent Lifecycle Events

Notifications	**M/O**	**Sender**	**Receiver**	**Based on actions performed by**	**Payload contains**
Consent Requested	M	HDCM	customer	HIU requested for consent	Time of request, HIU details
Consent Approved	M	HDCM	HIU	Customer approved the consent request by HIU	Time of request, customer’s details
Consent Revoked	M	HDCM	HIU, HIP	Customer revoked the consent	Time of revoke, Revoker details, Reason for Revoke

		HIP	HDCM	HIP revoked the consent	Time of Revoke, Revoker details, Reason for Revoke
		HDCM	HIU, Customer	HIP revoked the consent	Time of revoke, Revoker details, Reason for Revoke
Consent Paused	M	HDCM	HIU, HIP	Customer paused the consent	Time of pause, Customer details, Reason for pause
		HIP	HDCM	HIP paused the consent	Time of pause, HIP details, Reason for Pause
		HIP	HIU, Customer	HIP paused the consent	Time of pause, HIP details, Reason for Pause
Consent Expired	M	HDCM	HIU, HIP		Expiry date

The first time a consent is created it will go into an activated state.

7.1.3 Data Lifecycle Events

Data Lifecycle Events				
Notification	**M/O**	**Sender**	**Receiver**	**Based on action performed by**
HI Requested	O	HDCM	Customer	Customer
	O	HIP	Customer	HDCM
	O	HDCM	Customer	HIU
HI Ready	M	HDCM	Customer, HIU	HDCM
	M	HIP	Customer, HDCM	HIP
HI Denied	O	HDCM	Customer, HIU	HDCM
	O	HIP	Customer, HDCM	HIP
HI Sent	M	HIP	HDCM	HIU
HI Received	M	HIU	HDCM	HIU

HI Purged	M	HIU	HDCM	HIU

7.1.4 Data Availability Notifications

Data Availability Notifications		
From	**To**	**Description**
HIP	HDCM	The requested data is generated and HIP notifies the HDCM to retrieve the requested data.
HDCM	HIU	For every HIP that the HDCM receives data in an aggregation process, it sends a notification to HIU.
HDCM	Customer	HDCM pushes the data to the customer's data delivery destination and notifies the customer.

7.2 Summary of APIs by Entity

These API specifications cover all the entities within the Health Information Exchange ecosystem to facilitate seamless integration between ecosystem partners.

Technical Specifications for API Digital Signatures
All API requests and responses must be digitally signed by the respective organization initiating the APIs. The digital signature should follow W3C standards for XML and IETF standards for JSON. It should include information about the public key of the entity creating the signature and the associated certificates or certificate chains required for signature verification. Certificate chains can be included as part of the signature (e.g., the W3C standard has the option of including certificates in the “KeyInfo” element of the signature) or a URI for the certificate chain can be provided. This approach ensures that when a public key changes, the updated key and certificate are communicated to the party that needs to verify the signature. CA public keys should be managed as in any digital signature application. XML Format: Digital Signature XML element must adhere to the W3C Standards - https://www.w3.org/TR/xmldsig-core/

7.2.1 Central Registry - Health Sector Regulator Metadata

The registry is to be maintained centrally for ensuring all licensed HIPs and HDCMs are identified, discovered, and trusted by the ecosystem. HIPs and HDCMs will need be given an account on the application where the registry is maintained for managing their profile, public keys, and other details. The registry would be made available at a publicly accessible URL.

Entity		**Central Registry**
Method	API Path	Description
Registry Query		
GET	/Registry	It allows ecosystem partners to fetch details about all the HIUs, HIPs and HDCMs and obtain their respective certificates.

7.2.2 Health Data Consent Manager

Entity		**Health Data Consent Manager**
Method	API Path	Description
Consent Flow		
POST	/Consents	This API is intended for the HIU or HDCM Client to initiate the process of obtaining consent artefacts. Once the customer approves the consent request, the HDCM generates the digitally signed consent artefacts, stores them and notifies the HIU (who can later retrieve the artefacts at will).
GET	Consents/{ConsentsRefNumber}	This API is intended for checking the status of a previously submitted Consent Artefact creation request
GET	/Consents/Consent/{id}	This API is intended for fetching the consent artefact associated with a given consent ID. The method will return an artefact only if the requester initiated the consent creation process for it

Data Flow		
POST	/HI/request	This is the API for initiating the transfer of health information from the HIP to the HIU or HDCM Client (Requester). The requester submits the Consent IDs of the consents required for fetching health information from the HIP(s). A set of sessionIds are generated and returned. These SessionIDs enable the requester to fetch the information from the HDCM once available. The API provides an option to allow information to be sent directly from the HIP to a callback data-push URL specified by the HIU (“direct transfer” use case).
GET	/HI/fetch/{sessionId}	This API is used by the HIU/HDCM Client to fetch the health information from the HDCM against a given SessionID. It is invoked after the HIU/HDCM Client receives the <HI-Ready> notification from the HDCM.
Notifications		
POST	/Notification	This API is used by the HIPs and HIUs to submit notifications to the HDCM. In particular, this includes the notification (from HIPs) for availability of health information in the data flow and the notification for account linking in the linkage flow. It also includes the notification for purging of data from HIUs.
Monitoring		
GET	/Heartbeat	This API is used by HIUs to check availability of HDCMs

The details of these APIs are available from:
https://app.swaggerhub.com/apis/health_stack/hdcm_api/0.9#/

7.2.3 Health Information Provider

Entity	Health Information Provider

Account Discovery and Linking		
POST	/Accounts/discover	This API enables the HDCM to discover accounts belonging to the customer (and maintained by the HIP) based on the customer's identifiers.
POST	/Accounts/link	This API enables the HDCM to start the linkage of some of the customer's accounts (maintained by the HIP) to the customer's HDCM account.
GET	/Accounts/link/{LinkRefNumber}/{Token}:	This API is used only in the case of token-based authentication for linking accounts. The HDCM submits the token (received from the customer) to the HIP so that account linkage can be completed.
DELETE	/Accounts/link/{LinkRefNumber}	This API is used to delete a previously established account linkage i.e. to delink a set of HIP accounts of the customer that were previously linked to her HDCM account.
Data Flow		
POST	/HI/request	This is the API for initiating the transfer of health information from the HIP to the HDCM or directly to the HIU. The requester of the API (which is typically the HDCM, but could be the HIU as well) submits the consent artefacts required for fetching health information from the HIP. A set of sessionIds are generated and returned. These SessionIDs enable the requester to fetch information from the HIP once available. The API provides an option to allow information to be sent directly from the HIP to a callback URL specified by the HIU ("direct transfer").
GET	/HI/fetch/{SessionId}	This API is used to fetch health information from the HIP against a given SessionId. It is called after the HDCM has received the <HI-Ready> notification from the HIP. In the case of direct transfer, this API may not be

		called (i.e. the HIP may directly push the information to the HIU at the specified data-push URL)
Notification		
POST	/Notification	Notification API of the HIP
Monitoring		
GET	/Heartbeat	This API is used by the HDCMs to check availability of HIPs

The details of these APIs are available from: https://app.swaggerhub.com/apis/health_stack/hip_api/0.9