MyData Data Connection Specification

Notice

Implementation or use of certain elements of this document may require licenses under third party intellectual property rights, including without limitation, patent rights. The Participants of and any other contributors to the Specification are not and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights. This Specification is provided "AS IS”, and no Participant makes any warranty of any kind, expressed or implied, including any implied warranties of merchantability, non-infringement of third party intellectual property rights, and fitness for a particular purpose.

MyData Architecture defines the operations and APIs between the Operational Roles (Operator, Source, Sink etc.). Any descriptions or figures of the roles’ internal structure or operations are for illustrative purposes only.

1. Introduction

This document specifies MyData Data Connection.

This document is part of the MyData architecture release 1.1. It is assumed that the reader is familiar with MyData Architecture - Consent Based Approach for Personal Data Management document available at https://hiit.github.io/mydata-stack/.

1.1 Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2 Terminology

Key terminology used in this specification is defined in the Glossary of MyData Architecture - Consent Based Approach for Personal Data Management release 1.1 available at https://hiit.github.io/mydata-stack/.

1.2 Data Connection

A ***Data Connection*** is an authorised transfer of data from a specific Source to a specific Sink. The authorisation is given by Account Owner in the MyData Authorisation transaction, and multiple Data Connections may happen from Source to Sink as long as the authorisation is not deactivated or withdrawn.

For the Source, the authorisation consists of a Consent Receipt describing, what data can be provisioned to the requesting Sink. For the Sink, the authorisation consists of a Consent Receipt and an access token. The Consent receipt describes, how Sink can process the data. The token is used to authenticate to the Source as the authorised data requester. To this end, it is a Proof-of-Possession (holder-of-key) token, which contains Sink's public key. The data request must then be signed with the corresponding private key. The token is generated and signed by Operator.

2. Data Connection Transactions

This section describes MyData Data Connection transactions. There are 3 main transactions:

- Operator delivering a token to Sink
- Sink requesting a token from Operator
- Sink requesting data from Source

2.1 Operator Delivers Authorisation Token to Sink

Data request MUST always contain an authorisation token that proves Sink is authorised to make request to Source. Operator generates authorisation token and delivers it to the Sink. Transaction is depicted in Figure 2.1.

Prerequisites: MyData Authorisation has been completed.

Process:

- After Consent Records have been created in MyData Authorisation process, Operator generates an Authorisation token. Token is associated to Source's and Sink's Consent Record (CR) by Resource Set ID. Operator MUST sign this token with a private key indicated in the Source's CR. This signature is used to verify integrity of the token and that this token has been generated by Operator.
- Operator delivers token to Sink.
- Sink MUST store the Authorisation token.

Outcome: Sink receives the Authorisation token.

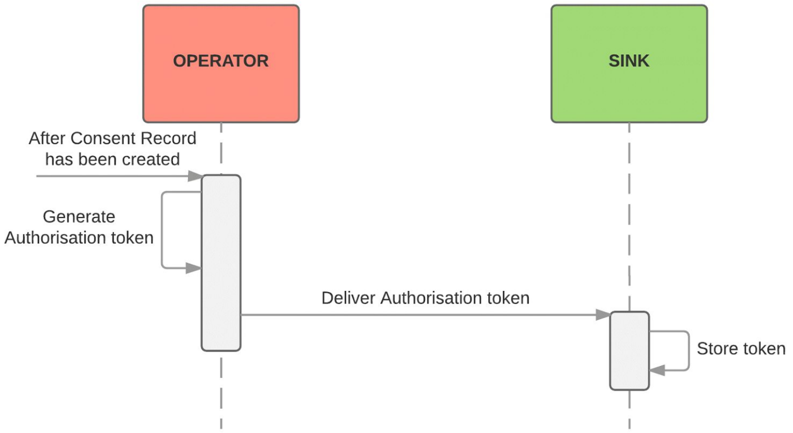

OPERATOR
SINK
After Consent Record has been created
Generate Authorisation token
Deliver Authorisation token
Store token

Figure 2.1: Operator generates and delivers Authorisation token to Sink

2.2 Sink Requests Authorisation Token

When Authorisation token has expired, Sink MUST request a new token before attempting a Data Connection. Sink MAY also request a new Authorisation token even if the previous token has not yet expired. Transaction is depicted in Figure 2.2.

Prerequisites: Sink has previously received an Authorisation token.

Process:

- Sink requests a new Authorisation token from Operator. As Authorisation tokens are always associated with a specific Consent Record (CR), the Sink indicates for which Consent Record it needs the token by using the corresponding Consent Record ID.
- When Operator receives a request for a new token, it fetches the indicated Consent Record
- Operator MUST check that the consent is active. If it is not, Operator MUST end the process and notify Sink about the inactive status of Consent Record.
- Operator generates a new authorisation token.
- Operator delivers token to Sink.

Outcome: Sink receives a new Authorisation token.

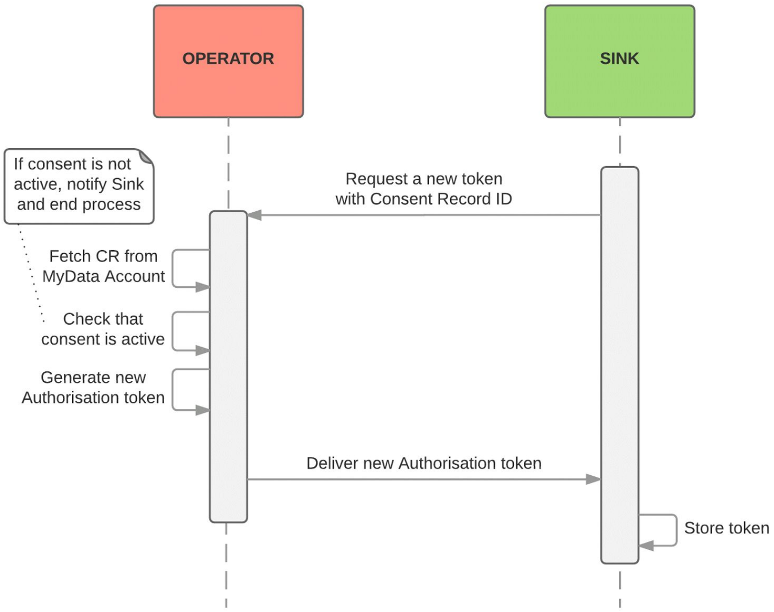

OPERATOR
SINK
If consent is not active, notify Sink and end process
Request a new token with Consent Record ID
Fetch CR from MyData Account
Check that consent is active
Generate new Authorisation token
Deliver new Authorisation token
Store token

Figure 2.2: Sink request a new Authorisation token from Operator

2.3 Sink Requests Data from Source

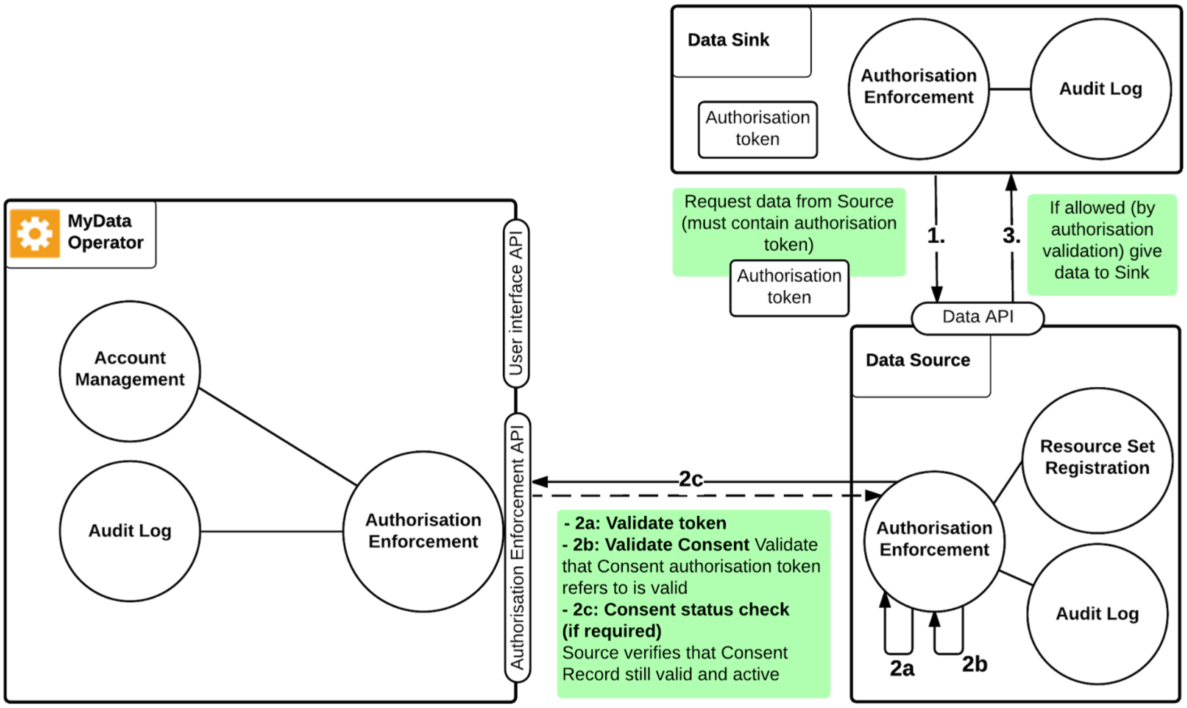
Sink requests data from the Source using an authorisation token. Transaction is depicted in Figure 2.3.

Prerequisites: Sink has a valid Authorisation token.

Process:

- Sink makes a data request to Source. The request MUST contain the Authorisation token Sink received from the Operator during the authorisation process and MUST be signed using the private key of the key pair specified in the Authorisation token.
- Source receiving the data request validates the provided token and the related Consent Record (see Section 2.3.1 for Request Validation).
- Source either denies or grants access to requested resources based on outcome of the validation. Source MUST also record the outcome in a Data Usage Log.

Outcome: Based on the validation process conducted by Source, Sink either receives the data it requested or receives an error message.



Data Sink
Authorisation token
Authorisation Enforcement
Audit Log
Request data from Source (must contain authorisation token)
Authorisation token
1.
3.
If allowed (by authorisation validation) give data to Sink
Data API
MyData Operator
User interface API
Authorisation Enforcement API
Account Management
Audit Log
Authorisation Enforcement
2c
Data Source
Resource Set Registration
Authorisation Enforcement
Audit Log
- 2a: Validate token
- 2b: Validate Consent Validate that Consent authorisation token refers to is valid
- 2c: Consent status check (if required) Source verifies that Consent Record still valid and active
2a
2b

Figure 2.3: Data connection transaction flow

2.3.1 Request Validation

For Source to grant access to the requested data, it must first validate the Sink's authorisation to access that data. This entails validating the Token and the related Consent Record as detailed in the following sections.

Prerequisites: Source has received a token as part of a data request.

Process:

- verify there exist a related Consent Record with the same Resource Set ID
- verify that token has been signed with a key indicated in the related Source's Consent Record
- verify token's integrity against the signature
- verify token's validity period includes the time of data request
- verify token's audience includes the URI to which the data request was made
- verify the request was signed with the key indicated in the token
- Validate the related Consent Record as defined in MyData Authorisation Specification.

Outcome:

- If all verifications pass and Consent Record is 'Active', grant access to requested data.
- If all verifications pass and Consent Record is 'Paused', reply with 'No new data'.
- Otherwise, deny the data request with appropriate error message.

3. Authorisation Token

This section defines the structure of the authorisation token, which is a *signed JSON Web Token*.

Table 3.1. Structure of a signed JWT

JOSE Header
JWS Payload
JWS Signature

Table 3.2. Structure of Authorisation token payload

FIELD_NAME	**DESCRIPTION**
iss	Public key of the token issuer (usually Operator). MUST be one of the keys listed in Source's Consent Record.
sub	Public key of the token user (Sink). Used to sign the data request as a Proof-of-Possession.
aud	Exhaustive list of URL patterns to which data requests are allowed with this token.
exp	The expiration time of the token on or after which the token MUST NOT be accepted.
nbf	The time before which the token MUST NOT be accepted.
iat	The time at which the JWT was issued.
jti	The "jti" (JWT ID) claim provides a unique identifier for the JWT.
rs_id	The Resource Set ID that was assigned in the linked Consent Record.

Token MUST be signed with the issuer's private key as defined in RFC 7515 (JWS) [RFC7515].

4. Data Connection APIs

Exact APIs to be used for data connection events will be released in upcoming MyData SDK, which is expected together with MyData architecture revision 1.2.

References

[RFC2119] Bradner, S, "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC7515] Jones, M., Bradley, J., Sakimura, N. “JSON Web Signature (JWS)”, RFC 7515, May 2015