MyData Architecture Framework

Notification Specification

v. 2.0

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 Notifications for MyData Architecture Framework.

This document is part of the MyData Architecture Framework release 2.0. The reader is assumed to be familiar with the 'MyData Architecture Framework' document available at https://github.com/mydata-sdk/mydata-docs/tree/master/architecture_specs .

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 [RFC2119].

1.2 Terminology

Key terminology used in this specification is defined in the Glossary of MyData Architecture Framework release 2.0 available at https://github.com/mydata-sdk/mydata-docs/tree/master/architecture_specs .

1.3 Formats

In MyData Architecture Framework, all data records and their respective digital signatures exchanged between actors are expressed using Javascript Object Notation (JSON). Digital signatures are expressed as JSON Web Signature (JWS)-structures and cryptographic keys as JSON Web Key (JWK)-structures.

In this document, JSON definitions of the data records are presented without JWS structures.
All Timestamps are in UTC in the NumericDate format as defined in [RFC7519].

2. Notification model

MyData Notifications allow Account Owner to subscribe to receive notifications from a linked Service that has chosen voluntarily or is required to support the framework's defined model. Actual contents of the notification are not defined in this specification as they are service specific. It is assumed that typical notification will contain information about data processing (type of data processed, by which organisation or individual, link to detailed information etc). Notifications can also be provided via multiple different delivery channels to the Account Owner (push notifications, email, etc.).

2.1 Discovery

Services supporting MyData Notifications MUST publish their list of supported notification types using the Notification Description defined in [MyData Architecture Framework - Service Descriptions].

3. Notification Transactions

3.1 Subscription

Motivation
Account Owner wants to receive notifications on data processing on a service.

Prerequisites:

- Service supports notifications and is linked to Account Owner's MyData Account

Process: (steps refer to figure 3.1)

- *Step 1:* Operator fetches information about notification types the service supports.
- *Step 2*: Account Owner selects what notifications she wants to receive
- *Step 3:* Operator saves the subscription request and sends it to the service.

Outcome:

- Account Owner is subscribed to receive notifications.

A simplified flow is shown in Figure 3.1 and a more detailed flow is shown in Figure 5.1.

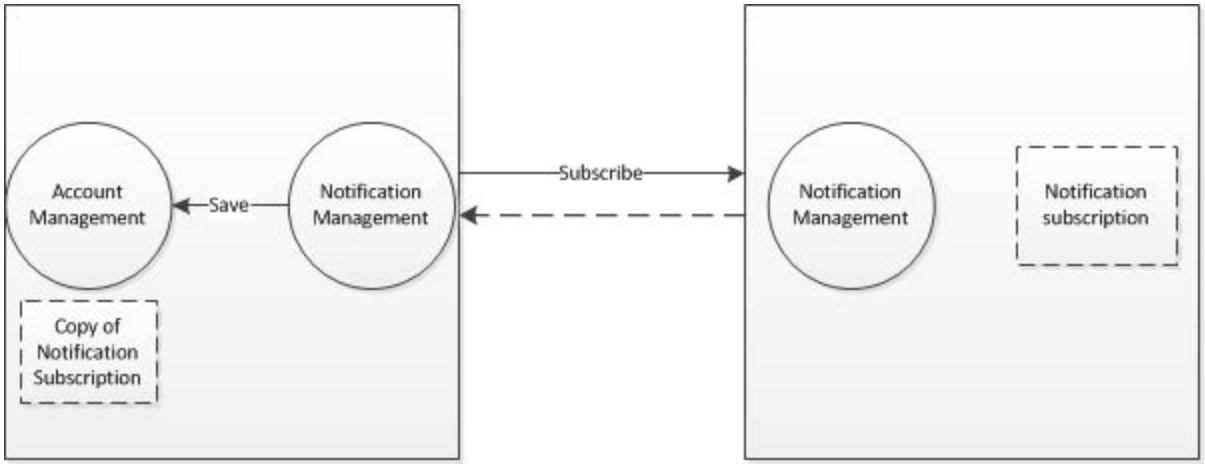

Figure 3.1: Notification subscription

3.2 Subscription Modification and Cancellation

Subscription may be modified by making a new subscription containing updated information. Cancellation uses same methods as subscription, the only difference being that the notification's id is prefixed with "-".

3.3 Notification Push

Motivation
Subscribed event happens at the service

Prerequisites:

- Account Owner has subscribed to receive notifications

Process: (steps refer to figure 3.2)

- *Step 1:* Service pushes notification to Operator.
- Step 2: Operator saves the received notification.

Outcome:

- Account Owner can see the notification when she logs in to the Operator, or
- Account Owner can receive the notification in real-time through Operator's own messaging service or push notification and app etc.

A simplified flow is shown in Figure 3.2 and a more detailed flow is shown in Figure 5.2.

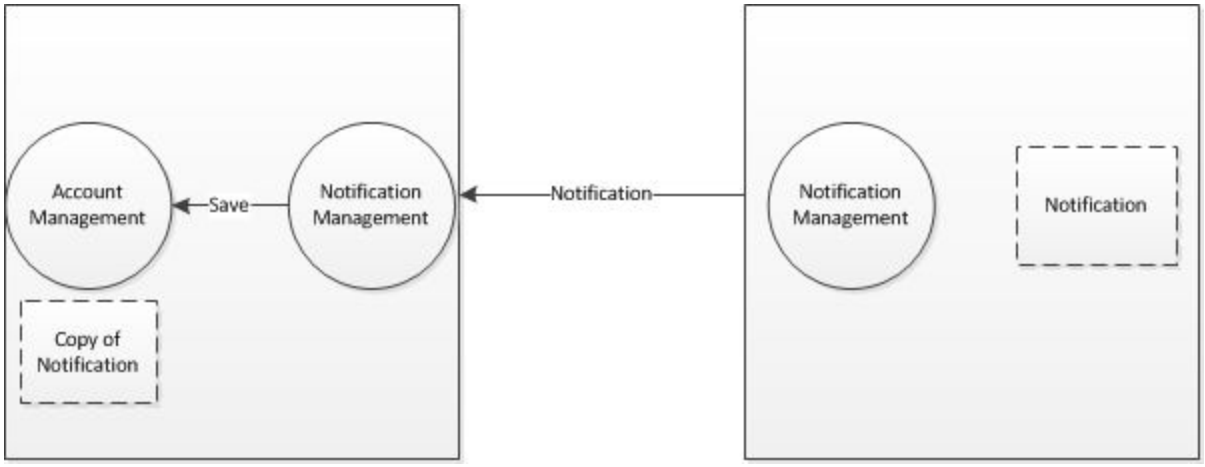

Figure 3.2: Notification subscription

4. Notification Data Structures

This section presents the structure of Notification payload and Notification Subscription.

4.1 Subscription request

Table 4.1 presents a detailed structure of Notification Subscription.

Table 4.1: Subscription request

KEY	TYPE	DESCRIPTION
id	string	Unique ID
iat	integer	Time when subscription was made
sub	string	surrogate_id of the user
notifications	Array of strings	Array of strings, each string referring to subscribed notification's ID. Subscription cancellation if ID is prefixed with "-"
encrypted	boolean	Whether notifications should be encrypted or not. Service MAY decide to encrypt the notification even if set to false.

Notification Request MUST be signed with the account owner's private key as defined in [RFC7515].

4.1.1 Subscription payload

```
{
  "iat": String,
  "sub: String,
  "notifications" : [String],
  "encrypted : Boolean
}
```

4.2 Notification

Table 4.2 presents a detailed structure of MyData Notification.

Table 4.2: Notification payload

<table>
<tr><th>KEY</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td>iat</td><td>string</td><td>Time when notification was created</td></tr>
<tr><td>service_id</td><td>string</td><td>ID of the service that issued the notification</td></tr>
<tr><td>sub</td><td>string</td><td>User surrogate_id</td></tr>
<tr><td>messages[]</td><td>Array of objects</td><td>
<table>
<tr><td>subject</td><td>string</td><td>Subject of the notification</td></tr>
<tr><td>id</td><td>string</td><td>Notification ID</td></tr>
<tr><td>message</td><td>string</td><td>The actual message</td></tr>
<tr><td>url</td><td>string</td><td>URL for additional information</td></tr>
</table>
</td></tr>
</table>

Notification MUST be signed with the Service's private key as defined in [RFC7515]. Furthermore notification MAY be encrypted using mechanisms defined in [RFC7516]. The actual encryption schema and parameters used are not defined in this specification.

4.2.1 Notification payload

```
{
   "iat" : String,
   "sub":  String,
   "Service_id" : String,
   "messages" : [
      {
         "subject" : String,
         "message" : String,
         "url" : String
      }
   ]
}
```

5. Notifications APIs

5.1 API Specification

API specification is available at: https://github.com/mydata-sdk/mydata-docs/tree/master/api_specs

6. Detailed Flow

Flow diagrams are available at: https://github.com/mydata-sdk/mydata-docs/tree/master/flow_diag

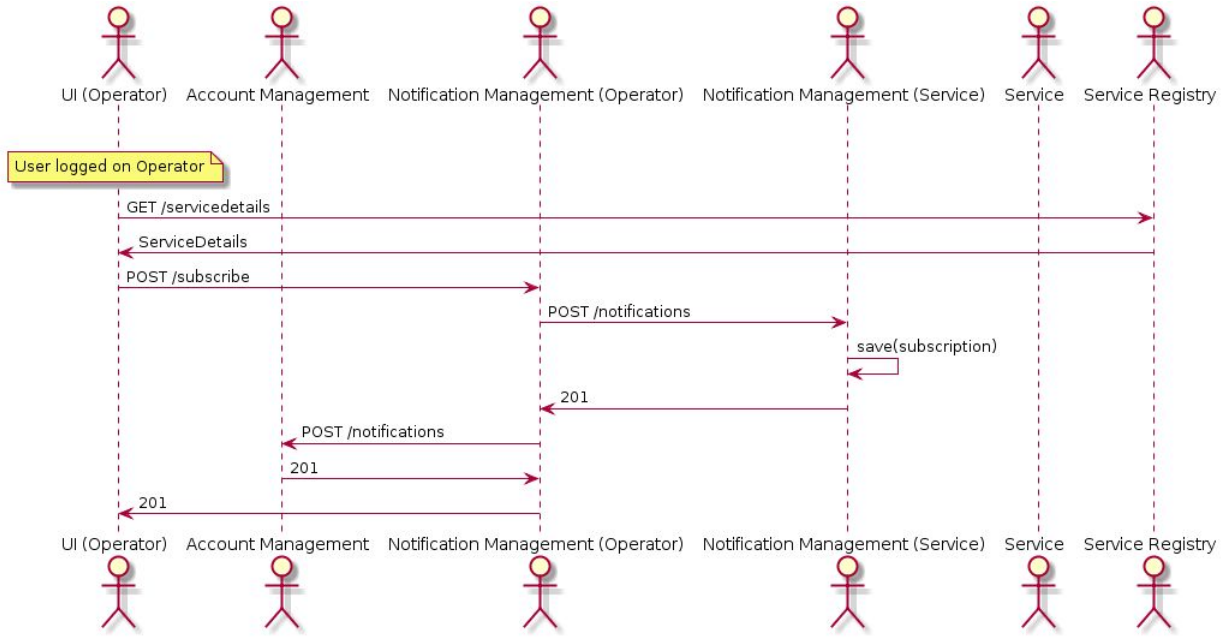

Figure 6.1: Notification subscription

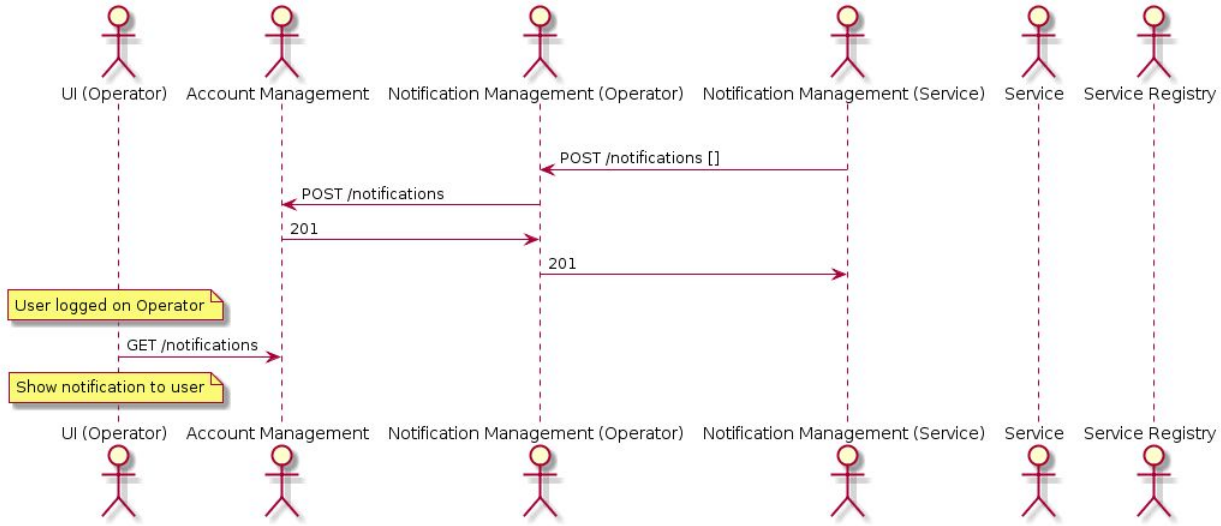

Figure 6.2: Push notification

7. 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”, RFC 7515, May 2015
[RFC7516] Jones, M, Hildebrand, J, “JSON Web Encryption” (JWE), RFC 7516, May 2015
[RFC7519] Jones, M., Bradley, J., Sakimura, N. “JSON Web Token (JWT)”, RFC 7519, May 2015