MyData Architecture Framework

Service Descriptions 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 the minimum service and data description structures for MyData Services and Operator configuration. It replaces the earlier ‘MyData Service Registry’ specification.

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 and with the parallel technical specification documents available at https://github.com/mydata-sdk/mydata-docs/tree/master/architecture_specs .

1.1 Notational Conventions

The keywords "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, 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. Service Description

Services MUST provide information on their configuration and MyData endpoint locations. Depending on the service type and supported use cases the service description consists of a common part, data description part, processing purposes part, processing part and notification part.

Depending on the service registration model this data is either provided from a central service registry or from the service itself. If the service is providing the information by itself, it MUST use specified .well-known/mydata/ endpoint - for using .well-known URIs see [RFC5785]. Structure of this endpoint is defined in https://github.com/mydata-sdk/mydata-docs/tree/master/api_specs. Discovery of self-declared services is out of scope of this specification.

2.1 Common Service Description

Each service MUST provide basic information, such as data controller details and access URLs. Service Description object consists of unique serviceId and serviceDescription object.

Table 1: Common Service Description

KEY	TYPE	DESCRIPTION
serviceId	String	Unique ID for the service
serviceDescription	Object	See table 2

Table 2: Service Description object

KEY	TYPE	DESCRIPTION
serviceProvider	Object	See table 3
serviceDescriptionTitle	String	Service's commercial/official title
serviceIconURL	String	URL pointing to service's icon
humanReadableDescription	String	Description of the service
keywords	Array of strings	List of keywords describing the service
issued	String	When entry was created (system log data)

<table>
<tr><td>serviceDescriptionVersion</td><td>String</td><td>Service description version number</td></tr>
<tr><td>supportedProfiles</td><td>Array of strings</td><td>Profiles supported: “contract, consenting, 3rd party re-use, notification, objection”</td></tr>
<tr><td>compatibleServices</td><td>Array of strings</td><td>Compatible service’s IDs</td></tr>
<tr><td>cert</td><td>Object</td><td><table><tr><th>KEY</th><th>TYPE</th><th>DESCRIPTION</th></tr><tr><td>x5u</td><td>String</td><td>URL pointing to service’s x.509 certificate</td></tr></table></td></tr>
<tr><td>serviceUris</td><td>Object</td><td>Object describing service URIs<table><tr><th>KEY</th><th>TYPE</th><th>DESCRIPTION</th></tr><tr><td>domain</td><td>String</td><td>Service’s domain (https://host:port)</td></tr><tr><td>libraryDomain</td><td>String</td><td>Services’s MyData integration library domain (https://host:port)</td></tr><tr><td>linkingUri</td><td>String</td><td>URI for service linking</td></tr><tr><td>linkingRedirectUri</td><td>String</td><td>URI user should be redirected after service has been linked. Used in service linking starting from the service.</td></tr><tr><td>objectionUri</td><td>String</td><td>URI for data processing objection</td></tr><tr><td>notificationUri</td><td>String</td><td>URI for subscribing notifications</td></tr></table></td></tr>
<tr><td>dataDescription</td><td>Array</td><td>Array containing Data Description objects (see 2.2)</td></tr>
<tr><td>processingBases</td><td>Array</td><td>Array containing Processing Description objects (see 2.3)</td></tr>
</table>

		<table><tr><th>KEY</th><th>DESCRIPTION</th></tr><tr><td>contract</td><td>Array containing Processing Description objects for contract based processing</td></tr><tr><td>consent</td><td>Array containing Processing Description objects for consent based processing</td></tr><tr><td>publicInterest</td><td>Array containing Processing Description objects for public interest based processing</td></tr><tr><td>legitimateInterest</td><td>Array containing Processing Description objects for legitimate interest based processing</td></tr><tr><td>vitalInterest</td><td>Array containing Processing Description objects for vital interest based processing</td></tr></table>

Table 3: Service Provider object

KEY	DESCRIPTION	VALUE
businessId	String	Service's business ID, e.g. VAT number
name	String	Name of the service
address1	String	Service's address
address2	String	Service's address
postalCode	String	Service's postal code
city	String	City where service is located
state	String	State where service is located
country	String	Country where service is located
email	String	Service's contact email

telephone	String	Service's contact telephone number
jurisdiction	String	Jurisdiction the service operates under

Service Description MAY be signed with the service's private key as defined in [RFC7515].
The JWS header MUST contain 'kid' field identifying services's key pair used to sign the Service Description.

2.2 Data Description

Each service processing or providing data (i.e. source) MUST describe the datasets it processes. Source service lists also datasets they can provide, and the API endpoint (distribution point). For clarification, a service does not need to describe data location if it uses data only internally i.e. it doesn't offer a related API for external use.

Table 4: Data description

<table>
<tr><th>KEY</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td>datasetId</td><td>String</td><td>Dataset's ID</td></tr>
<tr><td>title</td><td>String</td><td>Title</td></tr>
<tr><td>description</td><td>String</td><td>Description of the dataset</td></tr>
<tr><td>datasetSchema</td><td>Object</td><td>Object describing dataset's schema
<table>
<tr><th>KEY</th><th>DESCRIPTION</th></tr>
<tr><td>@context</td><td>JSON-LD context for the dataset (optional)</td></tr>
<tr><td>@type</td><td>JSON-LD type for the dataset (optional)</td></tr>
<tr><td>@id</td><td>JSON-LD id for the dataset</td></tr>
</table></td></tr>
<tr><td>url</td><td>String</td><td>URL pointing to further description of the data, e.g. to JSON schema (optional)</td></tr>
<tr><td>distribution</td><td>Array of objects</td><td>Objects describing dataset distribution point. NOT used in internal processing cases</td></tr>
</table>

		distributionId: ID for the distribution accessUrl: URL where data is available format: Format of the data

2.3 Processing Description

Each service processing data MUST describe the processing purposes and what data it uses for each purpose. Purposes refer to datasets using dataset's schema information. Each processing base ("contract", "consent", "publicInterest", "legitimateInterest", "vitalInterest") service uses is described using Processing Description described in table 3.

Table 5: Processing description

KEY	**TYPE**	**DESCRIPTION**
purposeId	String	Purpose's Id, MUST be unique within the service description
description	Array	Array of description objects containing localised information about the purpose **KEY**: **VALUE** language: Language used in information, ISO 639-1 coded descriptionUrl: URL pointing to information about the purpose title: Purpose's human-readable title description: Purpose's human-readable description iconUrl: URL pointing to icon for the purpose
requiredDatasets	Array of strings	IDs of the core datasets this processing purpose requires (to provide a meaningful outcome, i.e. without which the intended processing cannot fulfill its basic target)

optionalDatasets	Array of strings	IDs of the datasets this processing purpose may use when permitted by data subject. **Used only for consent based processing bases**

2.5 Notification Description

Services supporting Notifications MUST describe what notifications they support.

Table 6: Notification description

KEY	TYPE	DESCRIPTION
notifications	Array	Array of notification description objects
	Notification Object	Object describing notifications user can subscribe
id	String	Unique ID, used for subscribing notification
name	String	Notification's name (e.g. "Address Information", should pinpoint clearly what type of data has been processed)
shortDescription	String	Short description
longDescription	String	Long description
url	String	URL pointing to notification description

3. Operator Configuration

Operator MUST provide information on its configuration and MyData endpoint locations. Operator MUST implement .well-known/mydata/operator endpoint. Structure of this endpoint is defined in https://github.com/mydata-sdk/mydata-docs/tree/master/architecture_specs Operator discovery is out of scope of this specification.

3.1 Operator Description

Table 7. Operator description

<table>
<tr><th>KEY</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td>operatorId</td><td>String</td><td>Unique ID for the operator</td></tr>
<tr><td>serviceProvider</td><td>Object</td><td>See table 3</td></tr>
<tr><td>operatorservicedescriptionVersion</td><td>String</td><td>Operator service description version number</td></tr>
<tr><td>supportedProfiles</td><td>Array of strings</td><td>List of profiles supported: possible values are “contract, consenting, 3rd party re-use, notification, objection”</td></tr>
<tr><td>cert</td><td>Object</td><td><table><tr><th>KEY</th><th>TYPE</th><th>DESCRIPTION</th></tr><tr><td>x5u</td><td>String</td><td>URL pointing to operator’s X.509 certificate</td></tr></table></td></tr>
<tr><td>operatorUrls</td><td>Object</td><td>Object describing operator URls<table><tr><th>KEY</th><th>TYPE</th><th>DESCRIPTION</th></tr><tr><td>domain</td><td>String</td><td>Operator Service’s domain (https://host:port)</td></tr><tr><td>linkingRedirectUri</td><td>String</td><td>URL user should be redirected after service has been linked. Used in service linking starting from the operator</td></tr></table></td></tr>
<tr><td>createdOnDate</td><td>String</td><td>When the service description was created (system log)</td></tr>
<tr><td>createdByUserId</td><td>String</td><td>User who created the service description (system log)</td></tr>
</table>

Operator Description MAY be signed with the operator’s private key as defined in [RFC7515].

The JWS header MUST contain ‘kid’ field identifying operator’s key pair used to sign the Operator Description

4. Deployment Options

Service Descriptions can be provided from centralised source (Service Registry model, see earlier MyData Architecture releases) or the service can provide the information itself (decentralised model). Service must be registered to a service registry instance that serves the Operator's (or multiple Operators') registry services, if the Service Registry approach is used.

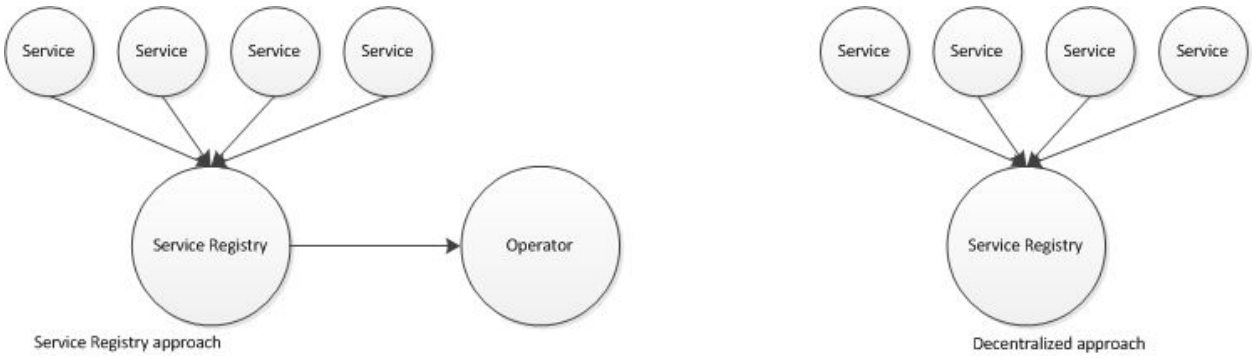

Figure 4.1 Centralised versus decentralised provisioning of service descriptions

5. References

[RFC2119] Bradner, S, "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC5785]: Nottingham, M, Hammer-Lahav, E, Defining Well-Known Uniform Resource Identifiers (URIs), 2010.
[RFC7515] Jones, M, Bradley, J, Sakimura, N, JSON Web Signature", RFC 7515, May 2015
[RFC7519] Jones, M., Bradley, J., Sakimura, N. "JSON Web Token (JWT)", RFC 7519, May 2015