

## MIWP-6: Registers - Discussion #2337

### Register Federation Architecture - DRAFT -

25 Feb 2015 12:14 pm - Daniele Francioli

<b>Status:</b>	New	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		

#### Description

## DRAFT ##

This discussion has been created in order to find the right architecture for the "Register Federation". The following paragraphs show a proposal for the different features related to the Register Federation.

The central tool of the federation is the Register of Registers (RoR), where **metadata** related to the registers and register items are indexed. The RoR does not contain the registers data but only metadata, reference to registers/register items and relations/links between them.

#### Definitions:

**Local registry:** a member state's registry or an european institution' registry. Also indicated as "local level".

**RoR:** register of registers - The central register handling the register federation.

## Metadata for the Register Federation

Below you can find a list of metadata elements for the federation.

- **Registry**
  - **Basic metadata:** URI, description, registry manager;
  - **API:** which operations are supported by this Registry (optional for the centralized approach). This is a list and a description of operations that can be performed by the Registry. A link to the Registry's API entry point is provided.
  - **Registers:** list of the registers contained in the registry (potentially only those that are included in the federation)
    - **Basic metadata:** URI, description, owner, register manager, control body, contact point;
    - **Extension information:** information on registers extending other registers and on register items extending other register items (e.g. RegB extends RegA --> RegB contains items in addition to those in RegA; RegY:CodeList2 extends RegX:CodeList1 --> CodeList2 contains items (values) in addition to the items (values) in CodeList1)

From an architectural point of view, there are 2 ways of setting the RoR: a centralized approach and a distributed approach.

## 1. Centralized approach

The centralized approach keeps all the federation information in the RoR at a centralized level. This means that all the stakeholders have to provide the metadata on registries and registers (see above) using the central API provided by the RoR. Below there are possible operations of the proposed RoR API for this scenario.

- Searching operations (to be used by register federation users):
  - GET RoRURI --> Collection - Gets a list of registries registered in the federation;
  - GET RegistryURI --> RegistryMD, including basic metadata, API, contained registers
  - GetRegistries() : Collection - Gets a list of registries registered in the federation;
  - GetRegisters(r : RegistryId) : Collection - Gets a list of federated registers contained in a registry (not all registers in a registry need to be part of the federation)
  - GetRegisterExtensions(r : RegisterId) : Collection - Gets all the extensions of a specific register
  - GetItemExtensions(i : ItemId) : Collection - Gets all the extensions of a specific register item (e.g. extended values of a code list item)
- Management operations (to be used by registry managers):
  - AddRegistry(r : RegistryDescription) : void - Adds a registry (incl. its metadata) to the federation
  - EditRegistry(rid : RegistryId, r : RegistryDescription) : void - Modifies registry metadata
  - RemoveRegistry(rid : RegistryId) : void - Removes a registry from the federation

- Adding/editing/removing registers, extension information etc. would be done through an update of the RegistryDescription. Alternatively, Add/Edit/Remove operations could also be provided for registers and extension information.

**PRO:** with this approach, the Local Registries (e.g. a member state registry) does not need to provide the API. Indeed, all the information will be registered in the federation by the local registries using the RoR API. Furthermore, no information related to the extension relations are stored at the local level (all centralized).

**CON:** any update to the Local Registers (e.g. addition or edit of an extension) shall be propagated to the RoR using its API.

## 2. Distributed approach

The distributed approach keeps only the information related to the local registries APIs entry in the RoR at a centralized level. This means that the RoR will automatically connect to the local registries API and get the information.

- Searching operations:
  - all of these operations are propagated to the local registries API
- Indexing operations:
  - AddRegistry(r : RegistryDescription) : void - Adds a registry (reference to the Registry's API) to the federation

**PRO:** with this approach, the local registries has no need to propagate the changes to the RoR because it automatically takes the information using the API provided by the local registries.

**CON:** the local registries has to provide the API with all the agreed functionalities. Furthermore, with this approach the extension relations are store at local level.

### History

#1 - 25 Feb 2015 12:15 pm - Daniele Francioli

- Description updated

#2 - 25 Feb 2015 01:48 pm - Daniele Francioli

- Description updated

#3 - 25 Feb 2015 01:48 pm - Daniele Francioli

- Subject changed from Register Federation Architecture to Register Federation Architecture - DRAFT -

#4 - 25 Feb 2015 05:05 pm - Daniele Francioli

- Description updated

#5 - 25 Feb 2015 05:12 pm - Daniele Francioli

- Description updated

#6 - 25 Feb 2015 05:21 pm - Daniele Francioli

- Description updated

#7 - 25 Feb 2015 05:22 pm - Daniele Francioli

- Description updated

#8 - 25 Feb 2015 05:23 pm - Daniele Francioli

- Description updated

#9 - 26 Feb 2015 09:16 am - Daniele Francioli

- Description updated

#10 - 26 Feb 2015 09:21 am - Daniele Francioli

- Description updated

**#11 - 26 Feb 2015 12:14 pm - Michael Lutz**

- Description updated

**#12 - 26 Feb 2015 03:18 pm - Daniele Francioli**

- Description updated

**#13 - 26 Feb 2015 03:25 pm - Daniele Francioli**

- Description updated

**#14 - 26 Feb 2015 03:31 pm - Daniele Francioli**

- Description updated

**#15 - 17 Mar 2015 03:13 pm - Michael Lutz**

- Description updated