

## ADOPT BBMRI-ERIC GRANT AGREEMENT NO. 676550

### DELIVERABLE REPORT

|                            |  |
|----------------------------|--|
| Deliverable no             | D3.6   |
| Deliverable Title          | Biobank connector specification and reference connector implementation |
| Contractual delivery month | M18  |
| Responsible Partner        | DKFZ   |
| Author(s)                  | Rumyana Proynova and the BBMRI CS-IT team                              |

## BIOBANK CONNECTOR SPECIFICATION AND REFERENCE CONNECTOR IMPLEMENTATION

### Executive Summary

The BBMRI-ERIC IT platform enables researchers to find samples and negotiate access to them. One of the components needed for that is the Connector, which governs the communication between the central search interface and the local data storage of a biobank. It is intended to be used by several BBMRI-ERIC IT components, most notably upcoming BBMRI-ERIC Locator, a tool to allow for retrieving availability statistics for structured data queries as a first step and to provide access to samples and data as a structured negotiation process between research and biobank as a second step. The Connector is also expected to provide automated updates of aggregated data on biobanks and their collections for future versions of the BBMRI-ERIC Directory.

This document is a Deliverable of the ADOPT BBMRI-ERIC project. It describes the technical interface between the Connector and the search interface accessible to the user, and reports the implementation of a proof-of-concept prototype of a Connector.



## Copyright notice



This work by Parties of the ADOPT BBMRI-ERIC Consortium is license under a Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

## Document log

| Issue     | Date (yyyy-mm-dd) | Comment  | Author/partner                  |
|-----------|-------------------|--|---------------------------------|
| D3.6_Rev1 | 2017-09-26        | Inclusion of results, next steps and future recommendations.   | Rumyana Proynova,<br>CS IT Team |
|           | 2017-10-04        | Inclusion of EU funding recognition to comply with Grant Agreement art. 29.4. Updates on Deliverable Template. | Outi Törnwall                   |



## Table of Contents

|   |    |
|---|----|
| 1. Background .....   | 4  |
| 2. Description of work and efforts .....  | 4  |
| 2.1 Role of the BBMRI Connector within the BBMRI-ERIC CS-IT architecture .....  | 4  |
| 2.2 The decentral search scenario .....   | 6  |
| 2.3 Query language and response language .....                                  | 7  |
| 2.4 Elements of the query language .....  | 7  |
| 2.5 Elements of the response language .....                                     | 8  |
| 3. Schedule .....   | 9  |
| 4. Results/ Outcome.....  | 9  |
| 5. Next Steps .....   | 10 |
| 6. Conclusions .....  | 10 |
| 7. Recommendations .....  | 10 |
| 8. References.....  | 11 |
| Appendix I - Sequence diagram for the confidential and the open connector ..... | 12 |
| Appendix II – Schemas for the Query language and Response language .....        | 18 |



## 1. Background

One of the goals of ADOPT BBMRI-ERIC is to create an IT platform in which researchers can submit sample requests to biobanks. The architecture for this platform was described in the ADOPT Deliverable D3.2, Architecture Notebook [1]. The platform is being developed in an incremental process, with new components added to it as they are released.

The BBMRI Connector is a component of the BBMRI-ERIC platform which will be released in 2017. It gets installed in the local biobank network (one instance per biobank) and has access to a *Local data silo*. When a query for samples arrives, the Connector searches for matching samples in the Local data silo, and if present, notifies the biobanker of the new query. The biobanker then decides whether to contact the requesting researcher and initiate a negotiation about the provision of samples and related data.

As a first step towards a full connector, we implemented a proof-of-concept system which can accept queries using a *query language* we defined for the Connector. This is a core functionality of the Connector which will later enable researchers to find samples via the BBMRI-ERIC hub infrastructure. It also defines the interface to the Sample Locator (search interface) component, making it possible for partners to develop their own connectors. This supports the idea of the BBMRI-ERIC infrastructure as an open, modular system, which allows the use of interchangeable components.

**In this report, we provide a description of the system implemented so far, as well as the definition of the query language used for communication between the Connector and Sample Locator.**

## 2. Description of work and efforts

### 2.1 Role of the BBMRI Connector within the BBMRI-ERIC CS-IT architecture

The BBMRI-ERIC IT platform has a complex structure (shown in Figure 1) which allows both centralized and decentralized search. The users can always create a search query using a central search interface. In the simple scenario, they can do a central search, which searches in a central data storage, which can be queried by the search interface directly. In BBMRI-ERIC, this storage component is called the BBMRI-ERIC Directory [2]. This is feasible for highly aggregated metadata about biobanks and their biological sample/data collections, with very low risks related to privacy of research participants (sample/data donors), which the biobanks are willing to supply to the BBMRI-ERIC maintainers in its entirety. The data available in Directories does not allow queries at the level of individual samples, as well as it does not support full “AND semantics” on the sample level (i.e., it can be searched for the AND queries on the level of collections, but that does not guarantee that there is corresponding combination available on the level of individual samples).



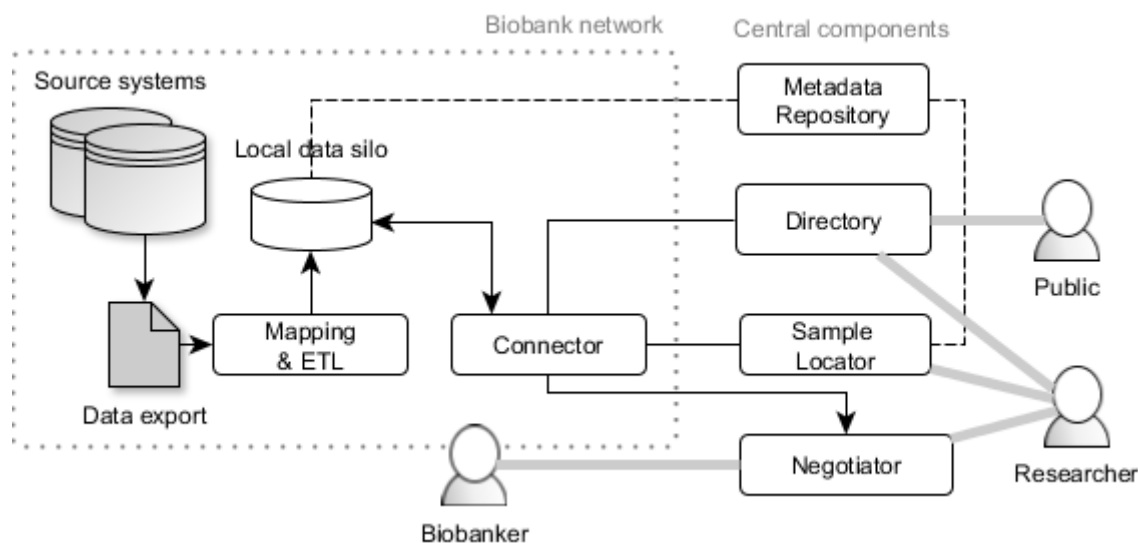


Figure 1 BBMRI-ERIC IT Platform

As many biobanks are not willing to list their detailed sample data in a system outside of their network, BBMRI-ERIC also aims at supporting decentralized search. For this search approach, the data is kept locally within the biobank network in a Local data silo (see Figure 1), and all queries are executed locally by a Connector component.

When a biobank joins the platform, it has to create a representation of its relevant data within the Local data silo. This is done in an ETL process, in which the biobank maps its data structure to an ontology created by BBMRI-ERIC. A Metadata Repository (MDR) describes the needed data elements.

The researcher can then enter a request using the central Sample Locator component. The Connector accesses the available queries and executes them against the Local data silo. If matching samples are found, the Connector notifies the biobanker, who can proceed to the Negotiator if desired.

As additional functionality, the Connector is able to extract the aggregated metadata about the sample/data collections from the Local data silo, allowing for automated or semi-automated (after approval) updates of the relevant data in the BBMRI-ERIC Directory. This makes the Connector a central part of the architecture, a component which acts as a bridge between the locally installed components in the biobank network and the central components hosted by BBMRI-ERIC. It also plays an important role in creating data protection and privacy, and creating trust among partner biobanks.

The data structure of the Local data silo is not encoded in the database model, but instead described in the Metadata Repository (MDR). This allows us to use a flexible and extensible data model, and ensures that all components (Local data silo, Connector, Sample Locator) work on the same data structure.

In the diagram in Figure 1, the biobank provides the source systems (which we assume are already present and used for the biobank's own purposes) and a data export. It is the biobanker who has to do the Mapping & ETL process using tools provided by BBMRI-ERIC. All other components are provided by BBMRI-ERIC.



## 2.2 The decentral search scenario

While gathering requirements for the Connector, we identified a number of scenarios that the stakeholders would like it to support. The main scenario, however, is the decentral search described above. This scenario represents the Connector's core functionality, and has to be supported under all circumstances.

### ***Open and confidential connector***

We recognized some division among stakeholders. A standard implementation of a decentral search architecture does not allow for the Connector to send any kind of automated response to the Sample Locator. We call this option the *confidential Connector*. It supports the highest level of data protection, and can be implemented in networks with a very stringent firewall.

The drawback of the confidential Connector is its low usability. With a confidential Connector, a researcher does not receive any kind of feedback before a human biobanker sees the query and takes the decision to reply (the Connector prepares a candidate reply for the biobanker, but the manual approval or possible manual modification of the reply is needed). This creates an uncertainty about the fate of a query and results possibly very lengthy gathering of the responses from the biobanks.

As an alternative, we also support *open Connector*, which responds to a query by sending to the Sample Locator the number of candidate samples in its biobank. The Sample Locator then sums up all results and displays them to the researcher without revealing the candidate biobanks from which the results came.

The open Connector option has the advantages of better usability, since it provides timely feedback. It avoids a situation in which a researcher creates a query without results, but is not notified and continues waiting to be contacted. It also allows the researcher to fine-tune the query until a reasonably high number of candidate samples is achieved, before committing to use it for a negotiation. It is however not well accepted by all biobanks, since the return of the number of samples may be a breach of their privacy policy. Also, it is more complex to implement, which increases the risk for meeting all goals of the ADOPT project in the projected timeline.

### ***Communication between components***

We represented the communication in this scenario as two sequence diagrams, one for the open Connector and one for the confidential Connector. The relevant components are the Connector, Negotiator and Sample Locator, with the two human actors Biobanker and Researcher.

The sequence diagrams are found in Appendix A of this document.



## 2.3 Query language and response language

In the decentral search, the researcher enters a query in the Sample Locator (running centrally), and it gets fetched and executed by the Connectors (running locally, one per biobank). The components are communicating over an Internet connection using REST. The query has to be formulated using a predefined *query language*, and the response is sent in a matching *response language*.

The messages in both queries are expressed as XML-documents, whose structure is aligned with the syntax of SQL queries, specifically SELECT-statements from SQL. The XSD schemas defining the languages are shown in Appendix B of this document.

This specification may yet be extended in later versions of the query and response language.

## 2.4 Elements of the query language

### **Attributes**

An attribute contains an MDR key and a value. A value has a given type, for example an integer, a float, a string or a date. The attributes are defined in Attribute.xsd.

### **MDR key**

A MDR key is the identifier of a data element saved in the MDR. It can occur together with a value for certain operators, or alone, e.g. when sorting. The MDR keys are defined in MdrKey.xsd.

### **View**

A virtual table, based on the results of a query. Supports the restriction to a number of attributes.

### **ViewFields**

Specifies the attributes, which are shown in a view, using a set of MdrKey elements. The default setting is to show all attributes.

### **Query**

Defines the query for a view. Its schema is defined in query.xsd.

### **GroupBy**

Used to group the results in a view.

### **Where**

Used to specify a filter within a query.

### **OrderBy**

Defines the sorting order for a query. The OrderBy element contains a list of Order elements.

### **Order**

Allows the ascending or descending sorting for MdrKeys.

### **And**

Used within a Where element to group filter options. The results have to fulfill all filter options.



### **Or**

Used within a Where element to group filter options. The results have to fulfill at least one filter option.

### **IsNotNull**

The record may not be empty.

### **IsNull**

The record must be empty.

### **Like**

Allows the use of wildcards, following the SQL LIKE syntax.

### **Neq**

The arithmetic operator “not equal”

### **Lt**

The arithmetic operator “less than”

### **Leq**

The arithmetic operator “less than or equal”

### **Eq**

The arithmetic operator “equal”

### **Geq**

The arithmetic operator “greater than or equal”

### **Gt**

The arithmetic operator “greater than”

## **2.5 Elements of the response language**

### [QueryResult](#)

This is an envelope element which contains either the number of donors or the number of samples, depending on what the query asked for

### [AmountDonors](#)

The number of donors within the Connector’s data silo which fit the search criteria

### [AmountSamples](#)

The number of samples within the Connector’s data silo which fit the search criteria

### **Error handling**

During the communication between the two components, it is possible that errors are encountered. For general errors, the system follows the error handling prescribed in RFC 2616 (Hypertext Transfer Protocol -- HTTP/1.1).





Some errors are not due to a technical problem, but occur on the semantic level. In this case, HTTP prescribes the code 422 Unprocessable entity. In the BBMRI platform, there is need for finer distinction of possible error states due to semantic problems. They are expressed using the definition in Error.xsd.

### 3. Schedule

The delivery was done on time

### 4. Results/ Outcome

Within the ADOPT project, we will deliver a Connector component which is capable of accepting queries from the Sample Locator and, if candidate samples are found, lead biobankers to contact the requesting researcher in the Negotiator. We have completed the **initial planning** phase, resulting in a specification for the first version of the Connector. This specification includes:

1. A definition of the scenario to be supported by the Connector. It has two variations, using either an Open connector which returns the number of candidate samples to the Sample locator, or a Confidential connector, which allows for a higher level of data privacy. It is the biobank's choice to configure their Connector as open or confidential.
2. An architecture of the CS-IT platform specifying the role of the Connector and its attached Data Silo in detail and clarifying that the needed interfaces. These are:
  - a. Sample Locator <-> Connector, for sample queries created by the researcher
  - b. Connector -> Negotiator, for allowing a biobanker with candidate samples to contact the researcher
  - c. ETL tools -> Data Silo, for filling the Data Silo with data on the samples available in the biobank

The connection between the Connector and the Data Silo is considered internal and not exposed as an interface. The Connector uses SQL to discover whether the Data Silo contains samples that match a researcher's query.

3. A description of the Core + Extensions strategy that will be used for the data model in the Data Silo.

After the planning phase, we implemented a **first prototype** of the Connector. This implementation can accept queries in the Query language, execute them against a temporary stand-in for the Data Silo, and return the number of found samples and donors in the Response Language. The prototype can be accessed at <https://bbmri-ccdc.mitro.dkfz.de/connector/>.



## 5. Next Steps

Development of a fully functional Connector as described above. This includes not only the search functionality, but also support for the full scenario of finding candidate samples and negotiating for them, as well as the supporting scenarios of installing a Connector in a biobank and preparing all necessary resources, e.g. harmonized data in the data silo, user management through the AAI component, and further prerequisites for successful operation.

## 6. Conclusions

With this deliverable, we provide a specification for the Connector and a first prototype which serves as a proof of concept of the basic principle of searching for samples. This provides the necessary basis for implementing a first functioning version of the Connector.

## 7. Recommendations

In the short term, ADOPT should continue the development of the Connector and also produce the Sample locator and the ETL tools, which are all needed for the search of biological resources on the sample level. This will allow for seamless communication between biobankers and researchers, and the search technology will allow for focused communication which includes only biobanks with candidate matches. User research shows that there is still a need for this type of solution on both the biobanker and researcher side.

The risks in this phase are related to properly implementing the interfaces between components created by different teams. These risks can be minimized with extensive communication between the teams.

After the end of ADOPT, BBMRI-ERIC should continue operation and maintenance of its CS-IT platform including the Connector. This requires sufficient resources for proper operation.

Our user research has shown that a significant problem in the communication between biobankers and researchers arises from the researchers not being able to formulate their request in detail, since they do not know which samples are best suited for a given research question. In their pre-BBMRI (non-automated) process, they get support from the biobankers themselves in an inefficient procedure. This inefficiency cannot be removed through purely technical solutions. Our recommendation for the long term is that BBMRI provides support by a human expert who provides consultation as a service to the researchers before they formulate a query for the Sample Locator. This will increase the value of the CS-IT platform as a whole, giving much better support to users' needs.

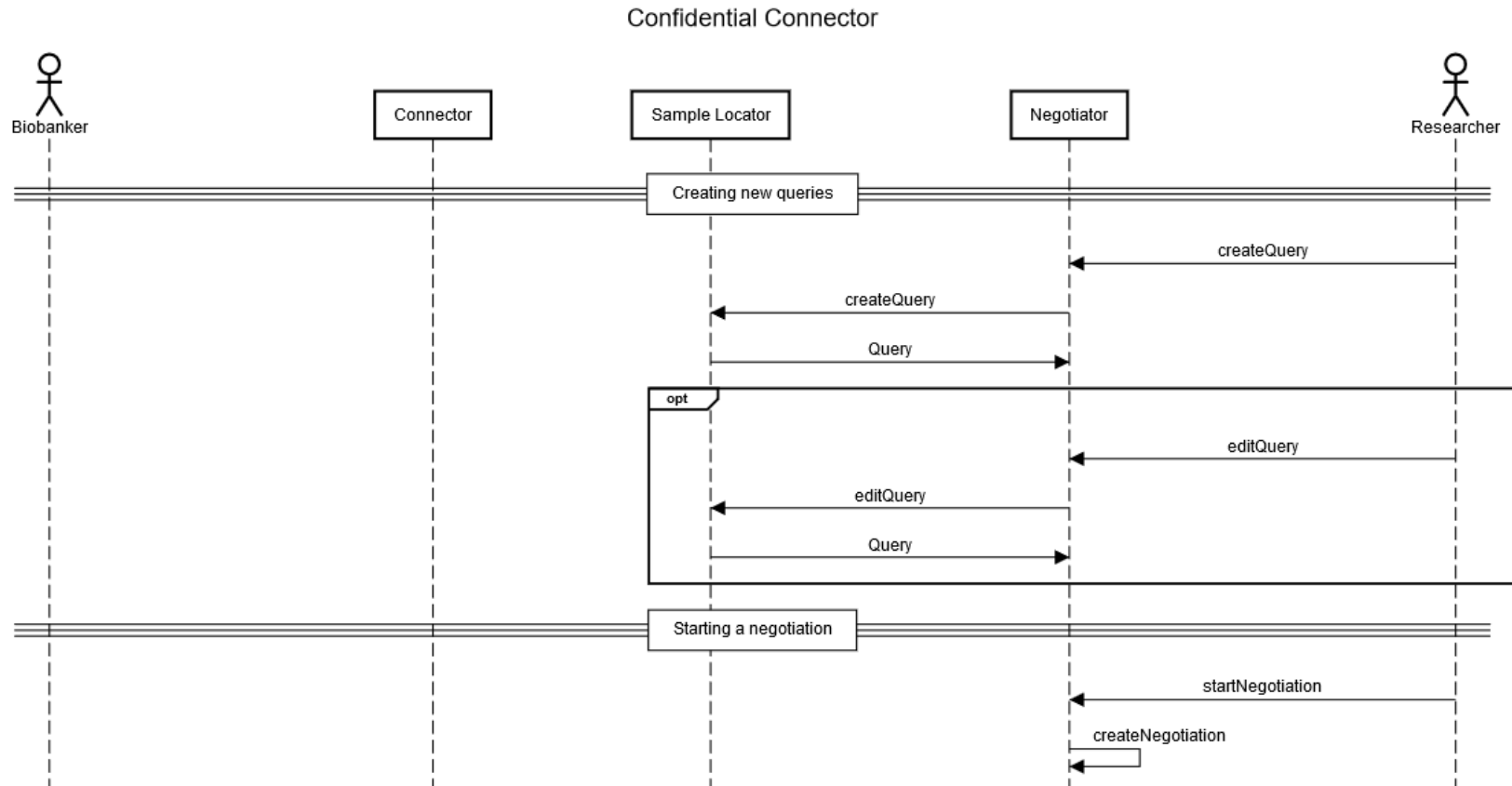


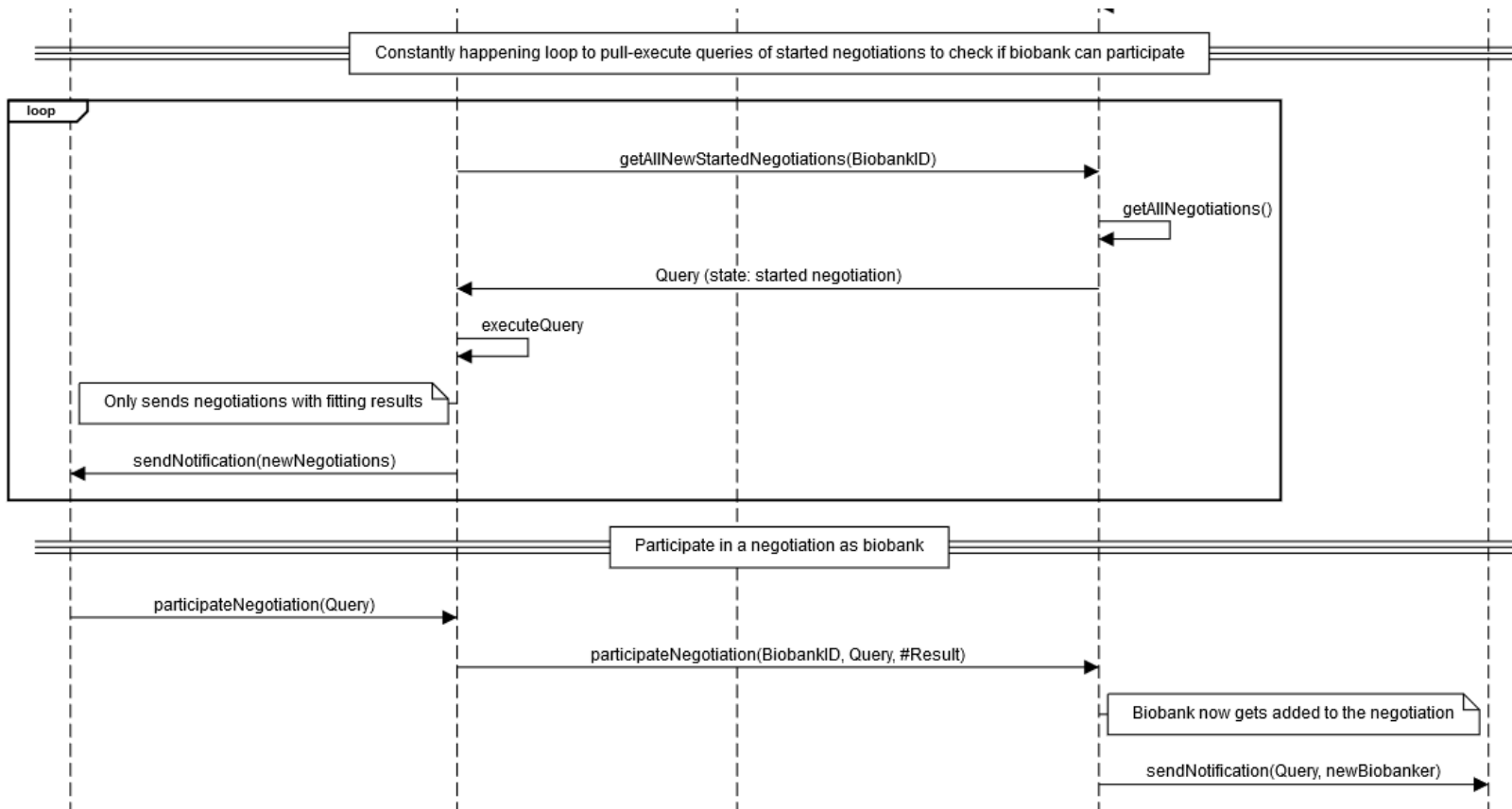
## 8. References

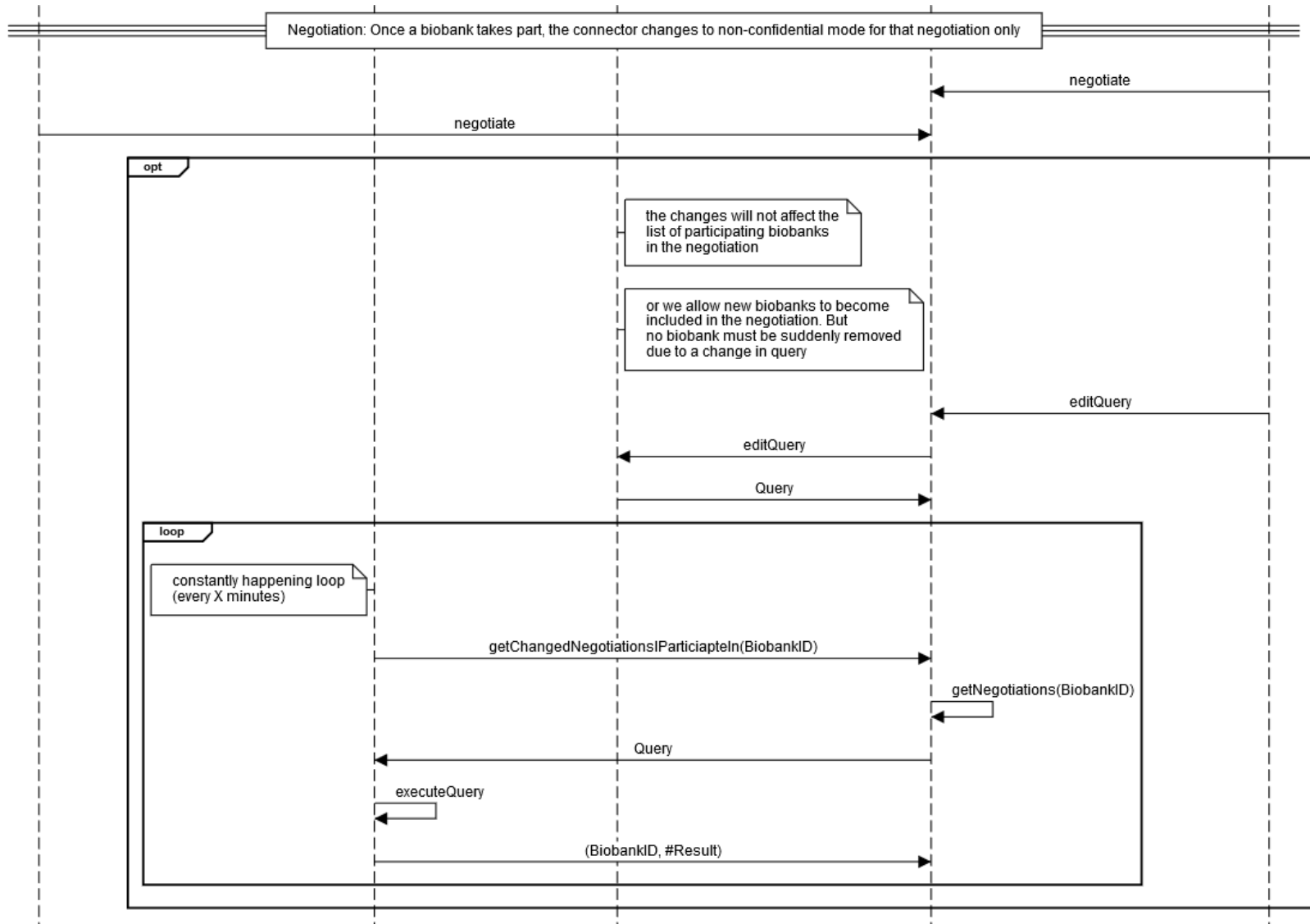
- [1] Diogo Alexandre, "Architecture Notebook: BBMRI-ERIC CS-IT Architecture," BBMRI-ERIC, 2016.
- [2] D. van Enkevort *et al.*, "BBMRI-ERIC directory: Metadata and aggregate data about biobanks and other bioresources," 2016.
- [3] Romyana Proynova: User interface for collection of the colon cancer cases and database. Report D 3.2 ADOPT. BBMRI-ERIC.



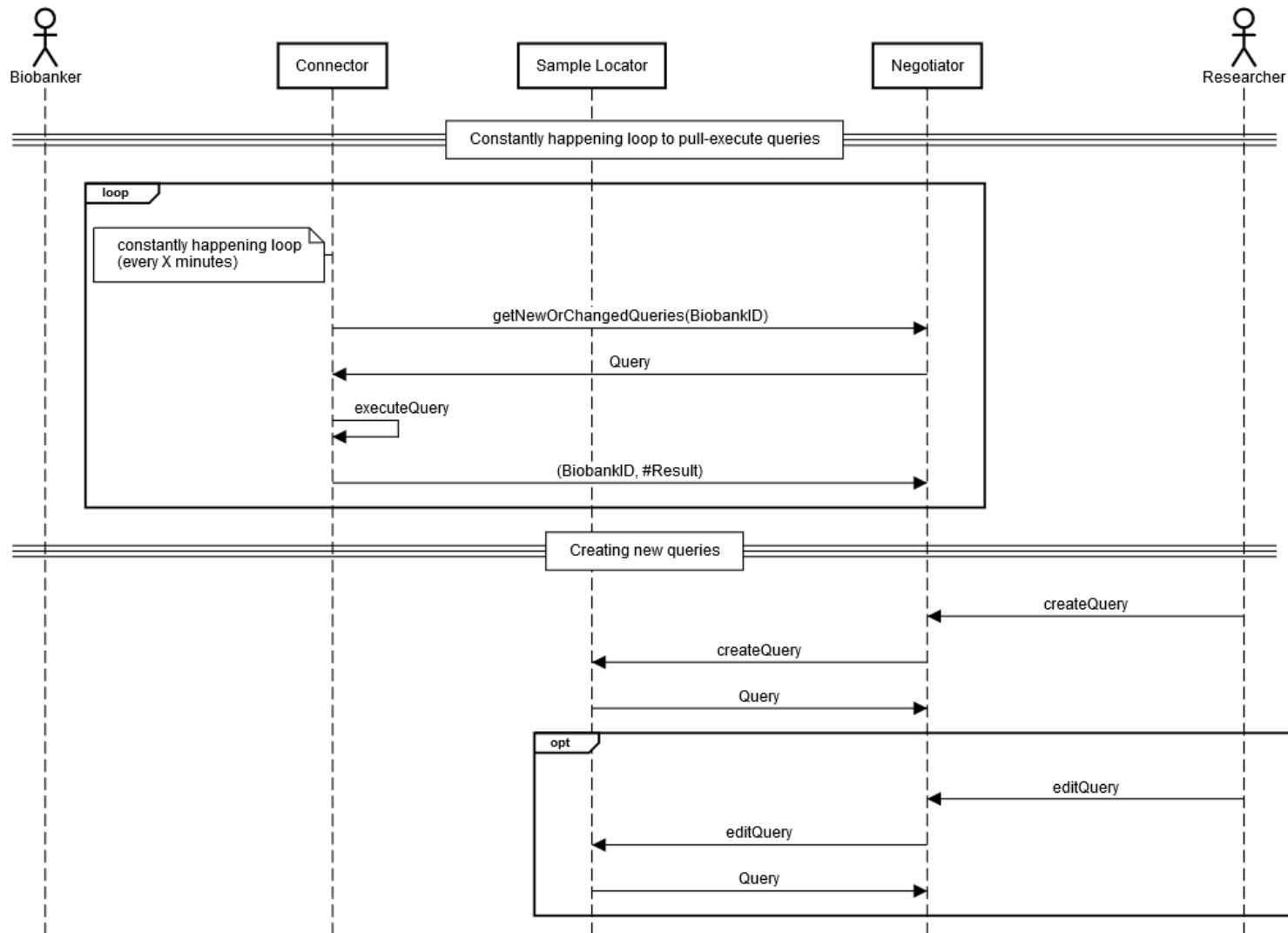
## Appendix I - Sequence diagram for the confidential and the open connector



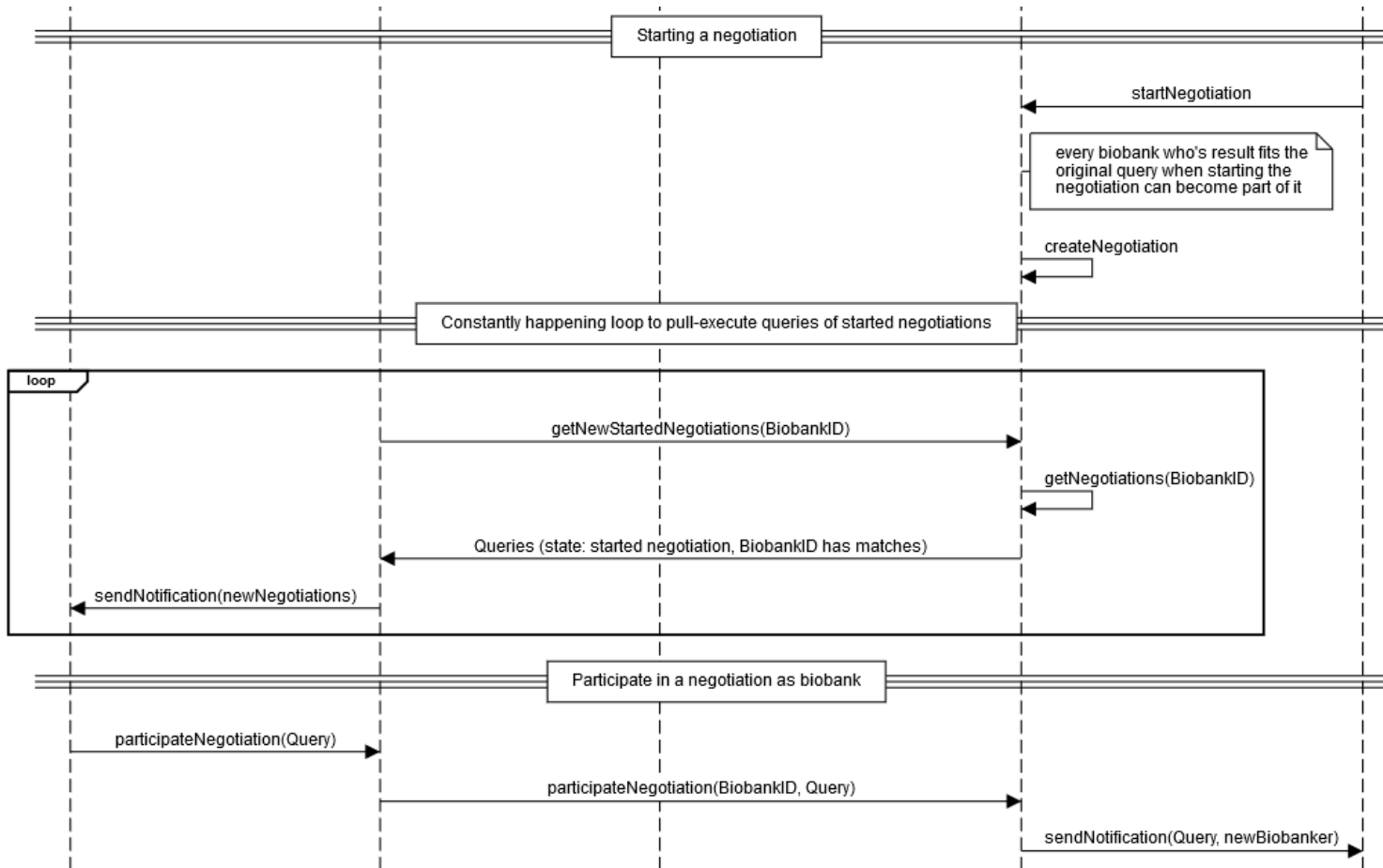




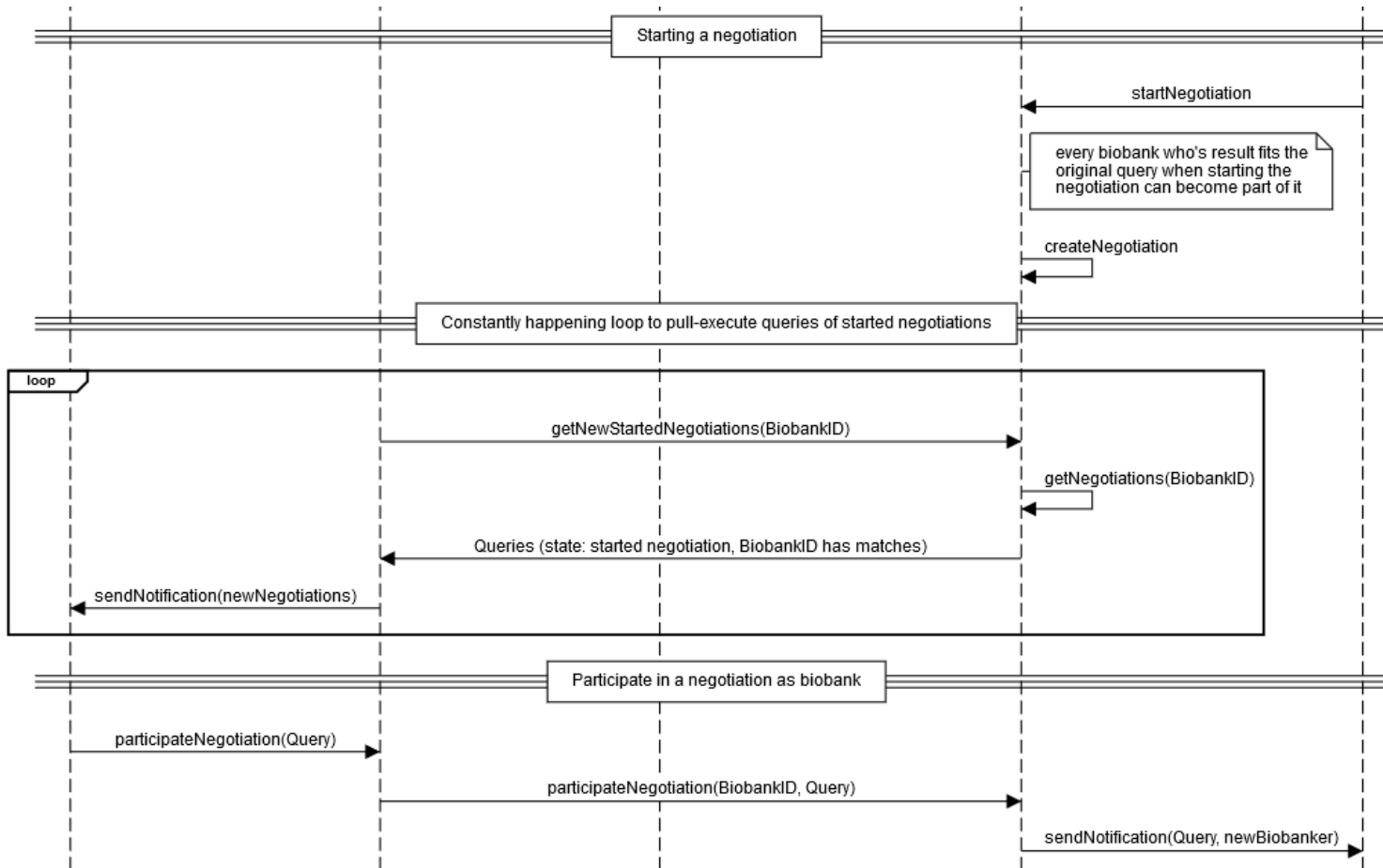
### Open Connector



This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 676550.







## Appendix II – Schemas for the Query language and Response language

### Attribute.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:attribute="http://schema.bbmri-eric.eu/bbmri/Attribute"
  xmlns:bbmri_mdrkey="http://schema.bbmri-eric.eu/bbmri/MdrKey"
  xmlns:bbmri_value="http://schema.bbmri-eric.eu/bbmri/Value" targetNamespace="http://schema.bbmri-eric.eu/bbmri/Attribute"
  version="1.0" xmlns:jxb="http://java.sun.com/xml/ns/jaxb" jxb:version="2.1">
  <xs:import namespace="http://schema.bbmri-eric.eu/bbmri/MdrKey"
    schemaLocation="http://schema.bbmri-eric.eu/bbmri/MdrKey.xsd" />
  <xs:import namespace="http://schema.bbmri-eric.eu/bbmri/Value"
    schemaLocation="http://schema.bbmri-eric.eu/bbmri/Value.xsd" />
  <xs:element name="Attribute">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="bbmri_mdrkey:MdrKey" />
        <xs:element ref="bbmri_value:Value" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:annotation>
    <xs:appinfo>
      <jxb:schemaBindings>
        <jxb:package name="de.samply.share.model.bbmri" />
      </jxb:schemaBindings>
    </xs:appinfo>
  </xs:annotation>
</xs:schema>
```



## MdrKey.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:bbmri_mdrkey="http://schema.bbmri-eric.eu/bbmri/MdrKey"
  targetNamespace="http://schema.bbmri-eric.eu/bbmri/MdrKey"
  version="1.0" xmlns:jxb="http://java.sun.com/xml/ns/jaxb" jxb:version="2.1">
  <xs:element name="MdrKey" type="xs:string"/>
  <xs:annotation>
    <xs:appinfo>
      <jxb:schemaBindings>
        <jxb:package name="de.samplly.share.model.bbmri" />
      </jxb:schemaBindings>
    </xs:appinfo>
  </xs:annotation>
</xs:schema>
```

## Value.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:bbmri_value="http://schema.bbmri-eric.eu/bbmri/Value" targetNamespace="http://schema.bbmri-eric.eu/bbmri/Value"
  version="1.0" xmlns:jxb="http://java.sun.com/xml/ns/jaxb" jxb:version="2.1">
  <xs:element name="Value" type="xs:string"/>
  <xs:element name="LowerBound" substitutionGroup="bbmri_value:Value" />
  <xs:element name="UpperBound" substitutionGroup="bbmri_value:Value" />
  <xs:annotation>
    <xs:appinfo>
      <jxb:schemaBindings>
        <jxb:package name="de.samplly.share.model.bbmri" />
      </jxb:schemaBindings>
    </xs:appinfo>
  </xs:annotation>
</xs:schema>
```



## Query.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema targetNamespace="http://schema.bbmri-eric.eu/bbmri/Query"
  xmlns="http://schema.bbmri-eric.eu/bbmri/Query"
  xmlns:bbmri_mdrkey="http://schema.bbmri-eric.eu/bbmri/MdrKey"
  xmlns:bbmri_attribute="http://schema.bbmri-eric.eu/bbmri/Attribute"
  xmlns:bbmri_range_attribute="http://schema.bbmri-eric.eu/bbmri/RangeAttribute"
  xmlns:bbmri_multivalued_attribute="http://schema.bbmri-eric.eu/bbmri/MultivaluedAttribute"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  version="1.1" xmlns:jxb="http://java.sun.com/xml/ns/jaxb" jxb:version="2.1">
  <xs:import namespace="http://schema.bbmri-eric.eu/bbmri/MdrKey"
schemaLocation="http://schema.bbmri-eric.eu/bbmri/MdrKey.xsd" />
  <xs:import namespace="http://schema.bbmri-eric.eu/bbmri/Attribute"
schemaLocation="http://schema.bbmri-eric.eu/bbmri/Attribute.xsd" />
  <xs:import namespace="http://schema.bbmri-eric.eu/bbmri/RangeAttribute"
schemaLocation="http://schema.bbmri-eric.eu/bbmri/RangeAttribute.xsd" />
  <xs:import namespace="http://schema.bbmri-eric.eu/bbmri/MultivaluedAttribute"
schemaLocation="http://schema.bbmri-eric.eu/bbmri/MultivaluedAttribute.xsd" />
  <xs:element name="Order">
    <xs:complexType>
      <xs:all>
        <xs:element name="Ascending" type="xs:boolean" minOccurs="1" maxOccurs="1"
default="true">
          </xs:element>
        <xs:element ref="bbmri_mdrkey:MdrKey" minOccurs="1" maxOccurs="1">
          </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="OrderBy">
    <xs:complexType>
      <xs:sequence>
```



This project has received funding from the *European Union's Horizon 2020* research and innovation programme under grant agreement No 676550.

```

        <xs:element ref="Order" minOccurs="1" maxOccurs="unbounded">
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Query">
    <xs:complexType>
        <xs:all>
            <xs:element ref="Where" minOccurs="1" maxOccurs="1">
            </xs:element>
            <xs:element ref="OrderBy" minOccurs="0" maxOccurs="1">
            </xs:element>
            <xs:element ref="GroupBy" minOccurs="0" maxOccurs="1">
            </xs:element>
        </xs:all>
        <xs:attribute name="id" type="xs:string" />
    </xs:complexType>
</xs:element>
<xs:element name="View">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Query" minOccurs="1" maxOccurs="1">
            </xs:element>
            <xs:element ref="ViewFields" minOccurs="0" maxOccurs="1">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ViewFields">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="bbmri_mdrkey:MdrKey" minOccurs="1" maxOccurs="unbounded">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```



```

    </xs:complexType>
</xs:element>
<xs:element name="IsNull">
  <xs:complexType>
    <xs:all>
      <xs:element ref="bbmri_mdrkey:MdrKey" minOccurs="1" maxOccurs="1">
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name="Where">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ConditionType" />
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="And">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ConditionType" />
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="Or">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ConditionType" />
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:complexType name="ConditionType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="And" />

```



```

    <xs:element ref="Eq" />
    <xs:element ref="Like" />
    <xs:element ref="Geq" />
    <xs:element ref="Gt" />
    <xs:element ref="IsNotNull" />
    <xs:element ref="IsNull" />
    <xs:element ref="Leq" />
    <xs:element ref="Lt" />
    <xs:element ref="Neq" />
    <xs:element ref="Or" />
    <xs:element ref="In" />
    <xs:element ref="Between" />
  </xs:choice>
</xs:complexType>
<xs:element name="Like">
  <xs:complexType>
    <xs:all>
      <xs:element ref="bbmri_attribute:Attribute" minOccurs="1" maxOccurs="1">
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name="Gt">
  <xs:complexType>
    <xs:all>
      <xs:element ref="bbmri_attribute:Attribute" minOccurs="1" maxOccurs="1">
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name="Geq">
  <xs:complexType>
    <xs:all>
      <xs:element ref="bbmri_attribute:Attribute" minOccurs="1" maxOccurs="1">

```



```

        </xs:element>
    </xs:all>
</xs:complexType>
</xs:element>
<xs:element name="Eq">
    <xs:complexType>
        <xs:all>
            <xs:element ref="bbmri_attribute:Attribute" minOccurs="1" maxOccurs="1">
            </xs:element>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="Leq">
    <xs:complexType>
        <xs:all>
            <xs:element ref="bbmri_attribute:Attribute" minOccurs="1" maxOccurs="1">
            </xs:element>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="Lt">
    <xs:complexType>
        <xs:all>
            <xs:element ref="bbmri_attribute:Attribute" minOccurs="1" maxOccurs="1">
            </xs:element>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="Neq">
    <xs:complexType>
        <xs:all>
            <xs:element ref="bbmri_attribute:Attribute" minOccurs="1" maxOccurs="1">
            </xs:element>
        </xs:all>
    </xs:complexType>
</xs:element>

```





```

    </xs:complexType>
  </xs:element>
  <xs:element name="IsNotNull">
    <xs:complexType>
      <xs:all>
        <xs:element ref="bbmri_mdrkey:MdrKey" minOccurs="1" maxOccurs="1">
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="In">
    <xs:complexType>
      <xs:all>
        <xs:element ref="bbmri_multivalued_attribute:MultivaluedAttribute" minOccurs="1"
maxOccurs="1">
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="Between">
    <xs:complexType>
      <xs:all>
        <xs:element ref="bbmri_range_attribute:RangeAttribute" minOccurs="1" maxOccurs="1">
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="GroupBy">
    <xs:complexType>
      <xs:all>
        <xs:element ref="bbmri_mdrkey:MdrKey" minOccurs="1" maxOccurs="1">
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>

```



```

</xs:element>
<xs:annotation>
  <xs:appinfo>
    <jxb:schemaBindings>
      <jxb:package name="de.samply.share.model.bbmri" />
    </jxb:schemaBindings>
  </xs:appinfo>
</xs:annotation>
</xs:schema>

```

## QueryResult.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://schema.bbmri-eric.eu/bbmri/QueryResult"
  targetNamespace="http://schema.bbmri-eric.eu/bbmri/QueryResult" version="1.0"
  xmlns:jxb="http://java.sun.com/xml/ns/jaxb" jxb:version="2.1">
  <xs:element name="QueryResult" type="QueryResult" />
  <xs:complexType name="QueryResult">
    <xs:choice>
      <xs:element name="AmountDonors" type="xs:integer" minOccurs="0" maxOccurs="1"/>
      <xs:element name="AmountSamples" type="xs:integer" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
    <xs:attribute name="id" type="xs:string" />
  </xs:complexType>
  <xs:annotation>
    <xs:appinfo>
      <jxb:schemaBindings>
        <jxb:package name="de.samply.share.model.bbmri" />
      </jxb:schemaBindings>
    </xs:appinfo>
  </xs:annotation>
</xs:schema>

```



## Error.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://schema.bbmri-eric.eu/bbmri/Error"
  xmlns:bbmri_mdrkey="http://schema.bbmri-eric.eu/bbmri/MdrKey"
  targetNamespace="http://schema.bbmri-eric.eu/bbmri/Error"
  elementFormDefault="qualified"
  version="1.0" xmlns:jxb="http://java.sun.com/xml/ns/jaxb" jxb:version="2.1">
  <xs:import namespace="http://schema.bbmri-eric.eu/bbmri/MdrKey"
    schemaLocation="http://schema.bbmri-eric.eu/bbmri/MdrKey.xsd" />
  <xs:element name="Error">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Description" />
        <xs:element ref="bbmri_mdrkey:MdrKey" minOccurs="0" maxOccurs="unbounded" />
        <xs:element name="extension" type="ErrorAttachment" minOccurs="0" />
      </xs:sequence>
      <xs:attribute name="errorCode" type="xs:int" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:complexType name="ErrorAttachment">
    <xs:sequence>
      <xs:any processContents="skip" minOccurs="0" maxOccurs="unbounded" namespace="##any"/>
    </xs:sequence>
  </xs:complexType>
  <xs:annotation>
    <xs:appinfo>
      <jxb:schemaBindings>
        <jxb:package name="de.samply.share.model.bbmri" />
      </jxb:schemaBindings>
    </xs:appinfo>
  </xs:annotation>
</xs:schema>
```

