RECAP Node Security Details

Version 1.2

INESC TEC Team
recap@lists.inesctec.pt
16/04/2019

Table of Contents

1. The RECAP Node

A RECAP Node is an isolated and decentralised software platform that provides data storage, harmonisation and analysis facilities, as well as user authentication and profile management and the creation of searchable study catalogs.

Data owners in RECAP Preterm consortium may choose to deploy one or more RECAP Nodes which they fully control.

1.1. Structure of a Node

A Node is composed of four main components: an Authentication Server, a Data Repository (which includes an R server for data analysis), a Study Manager and a Catalogue. Figure 1 provides an overview of the different components within each node.

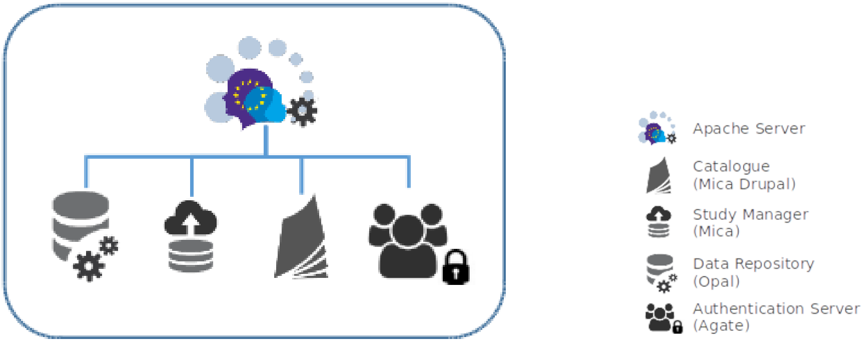


Figure 1: High-level view of the applications in a RECAP node

The four main applications belong to an open source software solution for data management, analysis and dissemination of epidemiological studies developed by OBiBa for the Maelstrom Research Consortium.

Each component serves its own purpose and interacts with the others to produce a functional interconnected system (more on intercommunication in Section 1.2). These components can be described as follows:

- The **Data Repository** (Opal) is the central data warehouse component at each RECAP Node. All operations of data import/export will take place here. Specific permissions can be granted to specific users on specific data (see Section 1.3.2).

  It also integrates an R server, which includes DataSHIELD[1], allowing controlled access to the data for non-disclosive statistical analysis.

[1] An infrastructure and series of R packages that enables the remote and non-disclosive data analysis

- The **Study Manager** (Mica) is the component where the metadata (study/population/collection events description) pertaining to epidemiological studies are defined. A study can be linked to data in the Data Repository.

- The **Catalogue** (Mica Drupal) is the top layer of the RECAP Node, providing browsing and querying capabilities over the published Study Manager metadata. Essentially, it displays all published content — with the exception of actual data, which never leaves the repository — in one place, allowing the end user to search for information in an accessible way.

- The **Authentication Server** (Agate) is the authentication component for each RECAP Node. Local users are able to access the other applications by using just one account registered in the local Authentication Server. It also provides an interface to manage users, groups and their correlations, as well as a role-based access control to the remaining components of the system.

The fifth component — the Apache Web server — sits on top of the others, allowing for HTTPS communication between the outside and the applications inside the Node (further described in Section 1.2).

1.2. Component Interactions

The applications within a node communicate with each other in the way illustrated in Figure 2 (further detailed in Figure 3). The Catalogue queries the Study Manager for published content and the Study Manager pulls variable metadata (data dictionaries and summary statistics) from the Data Repository.

All three of these applications use the Authentication Server to authenticate users.

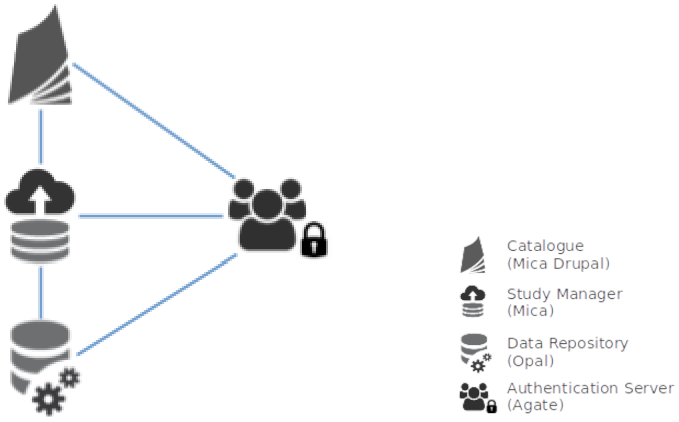


Figure 2: Communication between the applications in a RECAP Node

Note that the Catalogue does not directly communicate with the Data Repository. The only information from the repository that ever reaches the Catalogue is variable metadata that goes through the Study Manager. To run analyses on the data, a data analyst/researcher requires permission to directly log into the Data Repository via R or the web client. Analyses returning **aggregated** (i.e. anonymised) results can then be run using the DataSHIELD R packages. Only users with suitable permissions (normally only the local data manager) are able to upload or download individual patient data. Section 1.3.2 contains further details on permission levels.

Additionally, as shown in Figure 3, all outside requests directed at any of the four main applications must go through an Apache Web server, which is the only entrypoint into the node and therefore the only component exposed to the outside (see Section 1.3 for security details).

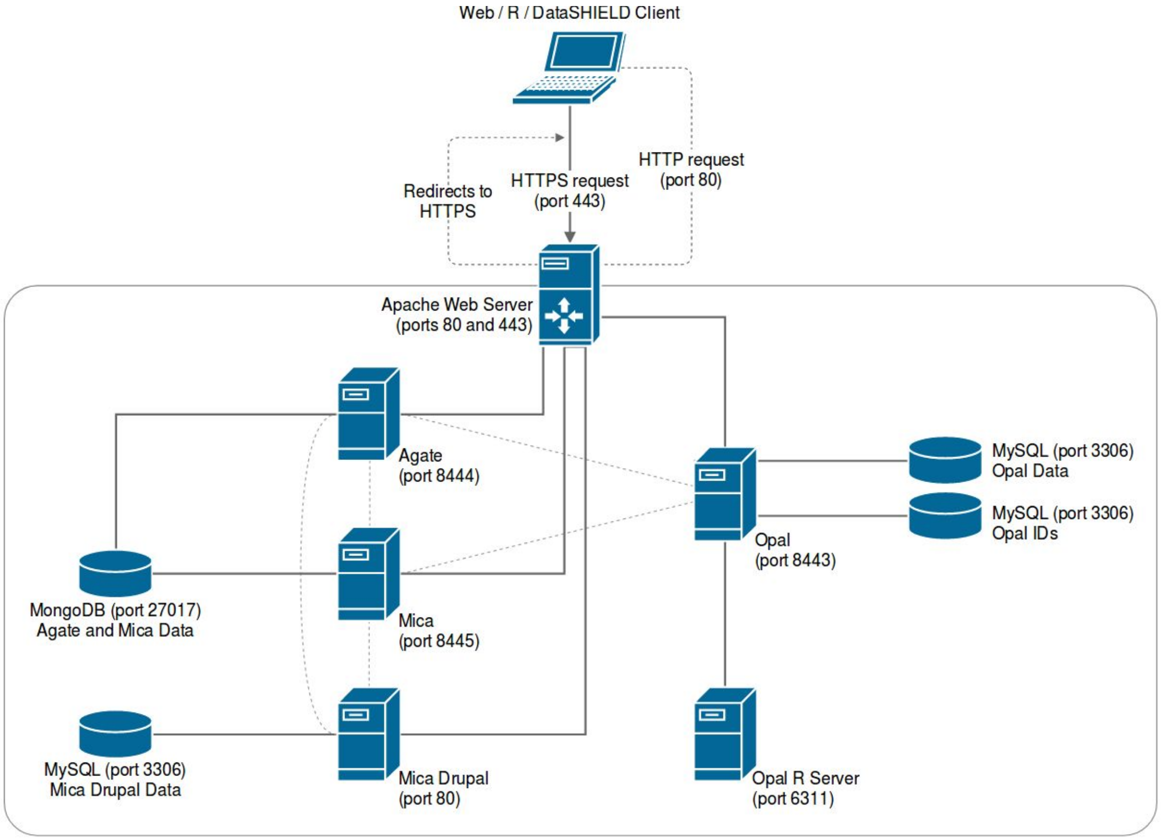

Web / R / DataSHIELD Client
Redirects to HTTPS
HTTPS request (port 443)
HTTP request (port 80)
Apache Web Server (ports 80 and 443)
Agate (port 8444)
Mica (port 8445)
Mica Drupal (port 80)
MongoDB (port 27017) Agate and Mica Data
MySQL (port 3306) Mica Drupal Data
Opal (port 8443)
MySQL (port 3306) Opal Data
MySQL (port 3306) Opal IDs
Opal R Server (port 6311)

Figure 3: Requests from the outside always go through the Apache Web Server (communication between the four main applications is represented by dashed lines, for clarity)

As referenced in the previous section, the applications are run inside Docker containers. Each component in Figure 3 (with the exception of the client) is a Docker container running inside the node.

1.3. Security Details

In order to comply with the different regulations that, in some cases, make it nearly impossible to host sensitive health-related data outside the country where it originates from, the RECAP network constitutes a distributed system, where **each partner hosts a RECAP Node and acts as a data provider (controller and processor)**.

The owner of a RECAP Node is autonomous in the management of its own research data, thus contributing to the long-term sustainability of the cohorts' data even after the temporal extent of RECAP.

1.3.1. External Access

As mentioned in Section 1.2, the Apache server is the only component that is directly accessible from outside of the node (on ports 443 and 80, which redirects to port 443 — as shown in Figure 3).

As the gateway into the node, it must be protected by a form of communication encryption. A well-known method to address this requirement is the use of digital certificates (TLS - Transport Layer Security), which are used by the Apache server at each node, in the transport layer, not only providing an encrypted connection (HTTPS), but also ensuring the authenticity of the server to which the client is connecting.

Additionally, Fail2ban and ModSecurity are also implemented in the Apache Web Server container to provide a Web Application Firewall (WAF) and to prevent brute-force attacks.

1.3.2. Role-based Access Control

Role-based access control is used to determine which resources (i.e. datasets, DataSHIELD functions, etc) each user is able to access. These role-based rules are guaranteed in each component of the RECAP Node, although, since the Study Manager and Catalogue do not handle any actual data (only metadata), they will not be fully detailed here.

In Agate, the **Authentication Server**, an account can belong to either an administrator or a regular user. An administrator is able to create regular users as well as other administrators.

Each user can be allowed — or not — to log into the other applications on the node.

An administrator is also able to organise users in groups. Each user group will be associated with different permissions defined inside each of the node's applications.

The **Data Repository** (Opal) provides very granular permission levels over stored data. The repository organises content as follows:

- Project

    - Table / View[2]
        - Dictionary (variables)
        - Values (the actual data)
        - Summary statistics

A project consists of a group of tables (or views) and each table contains values, a dictionary and summary statistics on each variable.

The repository allows three levels of permission:

- *Administrator*: enables the user to perform all available actions
- *Create Project*: enables the user to create its own projects (with full permissions on eventual tables inside the project)
- *Default*: the user is not allowed to perform any action, unless a second user explicitly grants him/her access to a project/table (provided, of course, that this second user already has permissions on the referred project/table).

Apart from these main permissions, there are also specific ones for:

Project

- *None*: no access to project (this is the default when the user is not the one who created the project)
- *Administrator*: enables the user to perform any action within a project (this is the default when the user is the one who created the project)
- *Project manager*: enables the user to perform any action within a project, except for changing the project's settings, which includes deletion of the project
- *View dictionary and values of all tables*: enables the user to read all project tables, including individual values.
- *Add tables*: the user is allowed to add tables to the project

Table / View

- *None*: no access to table/view (this is the default when the user is not the one who created the table/view or the project in which it resides)
- *Administrator*: the user has full access to the table, including edition of the dictionary and individual values (this is the default when the user is the one who created the table/view or the project in which it resides)
- *View dictionaries and summaries*: the user has no access to individual values
- *View dictionaries and individual values*: the user has access to individual values
- *Edit dictionaries and view values summaries*: the user has no access to individual values
- *Edit dictionaries and view individual values*: the user has access to individual values

[2] A view is a virtual table in which variables can be derived from other tables.

Variable

- *None*: no access to variable (this is the default when the user is not the one who created the variable, the table/view or the parent project)
- *Administrator*: the user has full access to the variable, including editing/modifying of its attributes and individual values (this is the default when the user is the one who created the variable, the table/view or the parent project)
- *View with summary*: view variable description and values summary (no access to individual values).

The Data Repository also allows the users to upload files either for a personal or a project folder. By default, a user is only allowed to access the files on the personal folder. If a user is granted access to a project, then they will also be able to access the project folder. As usual, an administrator is able to access all the personal and project folders.

1.3.3. Data Access and Analysis

In a RECAP Node, data is stored and managed at the Opal data warehouse. The access to the data is granted by the node's Data Manager, who can grant a specific user any of the permission levels described in 1.3.2 on any table or project.

After being granted permissions on some table on a node, a user has different options to access it. The appropriate option depends on the permission level that has been granted. As illustrated in Figure 4, if the permission level includes access to individual-level data, the user can either use the web interface or the R server to export the data. On the other hand, if the user does not have access to individual values, non-disclosive aggregated data can still be retrieved through DataSHIELD.

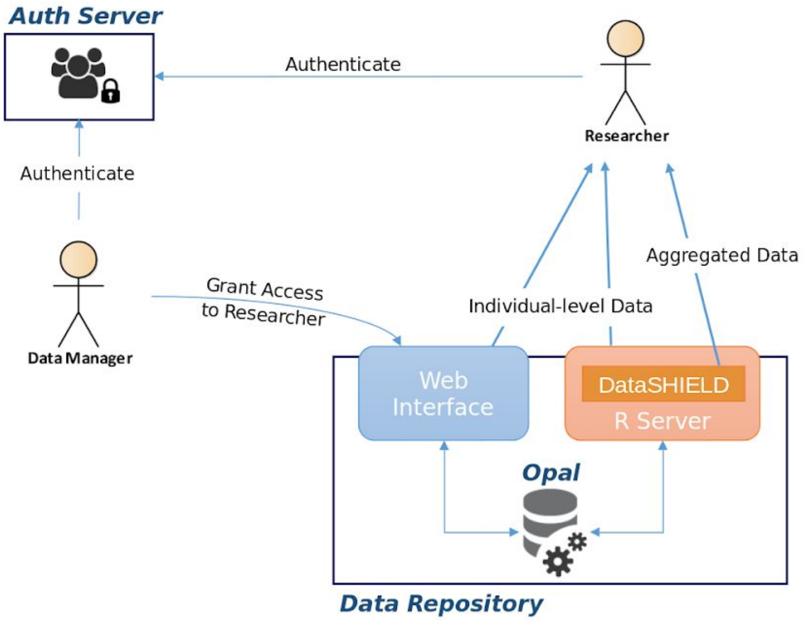


Figure 4: Extracting (aggregated) data from a RECAP node (Apache server is omitted for clarity)

In addition to table permissions, a user also needs specific permission to access the R server and DataSHIELD.

The R server permissions are divided into two options:

- *Default*: by default a user is not able to use the R server
- *Use*: allows a user to use the R server to access tables for which the user has permissions

DataSHIELD permissions include:

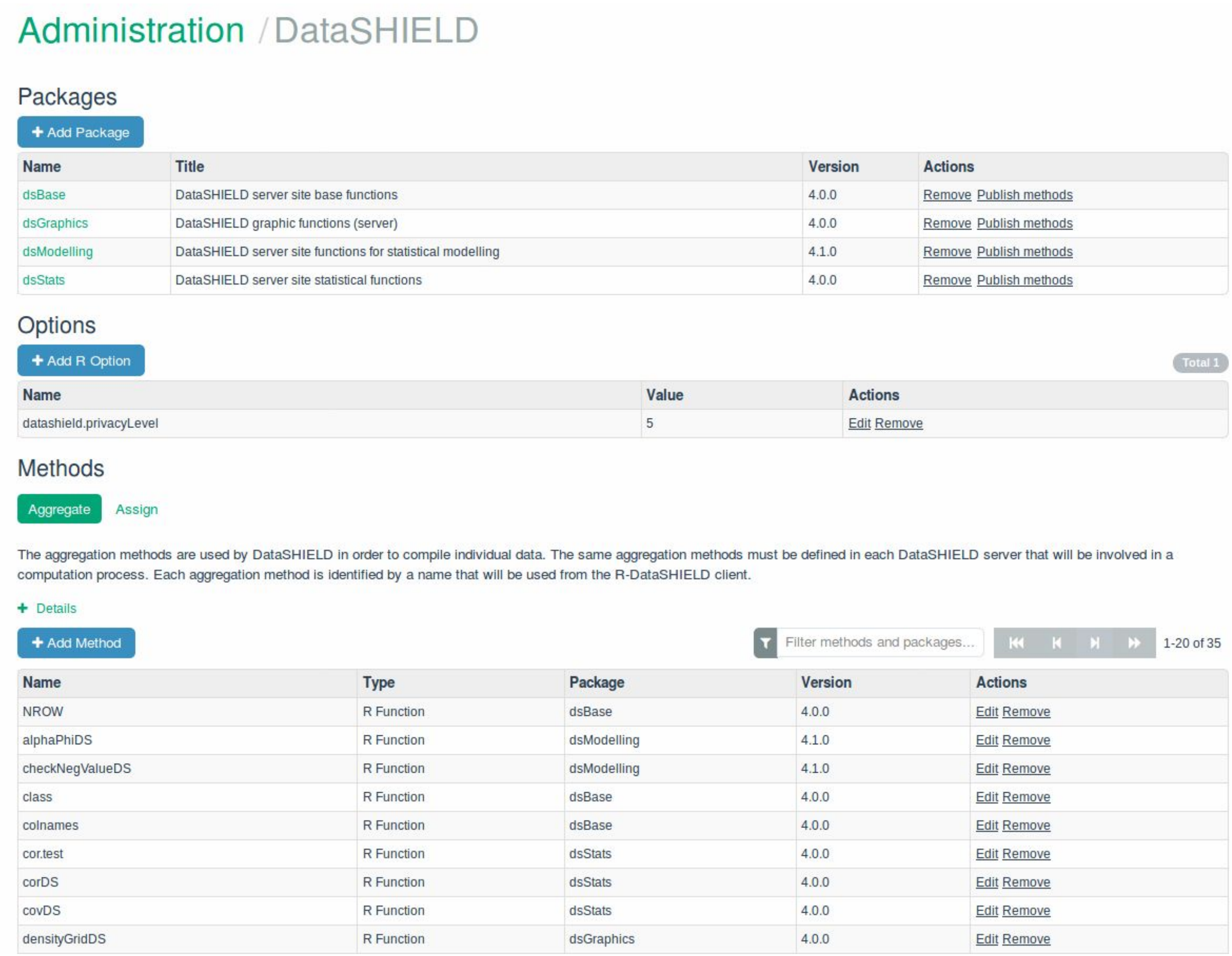
- *Default*: by default a user is not able to use DataSHIELD
- *Use*: enables the user to retrieve aggregated data through DataSHIELD. However, the user still needs at least “View dictionaries and summaries” permission on a table in order to access that table through DataSHIELD
- *Administrate*: user has ‘use’ permission and is also allowed to edit the DataSHIELD settings

DataSHIELD Privacy Settings

Figure 5 shows other available DataSHIELD settings. The Data Manager is responsible for configuring DataSHIELD server packages and functions on the node. Only the functions that have been published are allowed to be executed by users.

A further restriction the Data Manager can set is the DataSHIELD privacy level. By default, the privacy level is set to 5, meaning that published functions will not return any results if the calculation involves data from less than 5 participants, as the results may potentially be disclosive. Low privacy levels are not recommended as they can result in retrieving individual-level data. For example, for a setup with a privacy level of 1, using the *ds.mean* function on a variable for which only one participant has collected data, would result on an individual value being disclosed.

It is important to note that different studies can use different privacy-levels, but it can complicate the statistical analysis and is not recommended to build a statistical analysis process using studies with different data privacy levels (Wilson R C, 2017). Low privacy levels are not recommended as they can result in retrieving individual-level data.



Administration / DataSHIELD

Packages

Name	Title	Version	Actions
dsBase	DataSHIELD server site base functions	4.0.0	Remove Publish methods
dsGraphics	DataSHIELD graphic functions (server)	4.0.0	Remove Publish methods
dsModelling	DataSHIELD server site functions for statistical modelling	4.1.0	Remove Publish methods
dsStats	DataSHIELD server site statistical functions	4.0.0	Remove Publish methods

Options

Name	Value	Actions
datashield.privacyLevel	5	Edit Remove

Methods

The aggregation methods are used by DataSHIELD in order to compile individual data. The same aggregation methods must be defined in each DataSHIELD server that will be involved in a computation process. Each aggregation method is identified by a name that will be used from the R-DataSHIELD client.

Name	Type	Package	Version	Actions
NROW	R Function	dsBase	4.0.0	Edit Remove
alphaPhiDS	R Function	dsModelling	4.1.0	Edit Remove
checkNegValueDS	R Function	dsModelling	4.1.0	Edit Remove
class	R Function	dsBase	4.0.0	Edit Remove
colnames	R Function	dsBase	4.0.0	Edit Remove
cor.test	R Function	dsStats	4.0.0	Edit Remove
corDS	R Function	dsStats	4.0.0	Edit Remove
covDS	R Function	dsStats	4.0.0	Edit Remove
densityGridDS	R Function	dsGraphics	4.0.0	Edit Remove

Figure 5: DataSHIELD configuration

1.4. Auditability

System logs or equivalent mechanisms allow system administrators to audit each system component. For any given use case on a RECAP Node, there is a trigger that stores: the user who made the action, the action itself and a timestamp. Each component stores more specific details for each of its users' actions in a log.

These actions include:

- For the Authentication Server:
  - Creation and removal of user sessions, which are also associated with the application from which the authentication process was initialised;
  - CRUD[3] operations on Users, Groups or Applications.

- For the Data Repository:
  - User login attempts;
  - CRUD operations on Projects, Tables and Variables;
  - Import and export of data or data dictionaries;
  - Functions executed through R/DataSHIELD.

- For the Study Manager:
  - User login attempts;
  - CRUD operations on Networks, Studies and Datasets;
  - Changing the publication state of Networks, Studies and Datasets.

Since the Catalogue uses information from the Study Manager, every relevant request made is passed to the Study Manager, which means that any relevant activity in the Catalogue is registered in the Study Manager's logs.

[3] Create, Read, Update and Delete.

2. The RECAP preterm Network

The RECAP preterm network is composed by functional decentralised RECAP nodes - at least one node for each data-owning RECAP preterm partner. Each node is structured and works as described in Chapter 1.

2.1. High-level View

Figure 6 provides an overview of the RECAP preterm network, comprised of the several nodes deployed at the partner's own premises and of the Researcher's Portal hosted at INESC TEC.

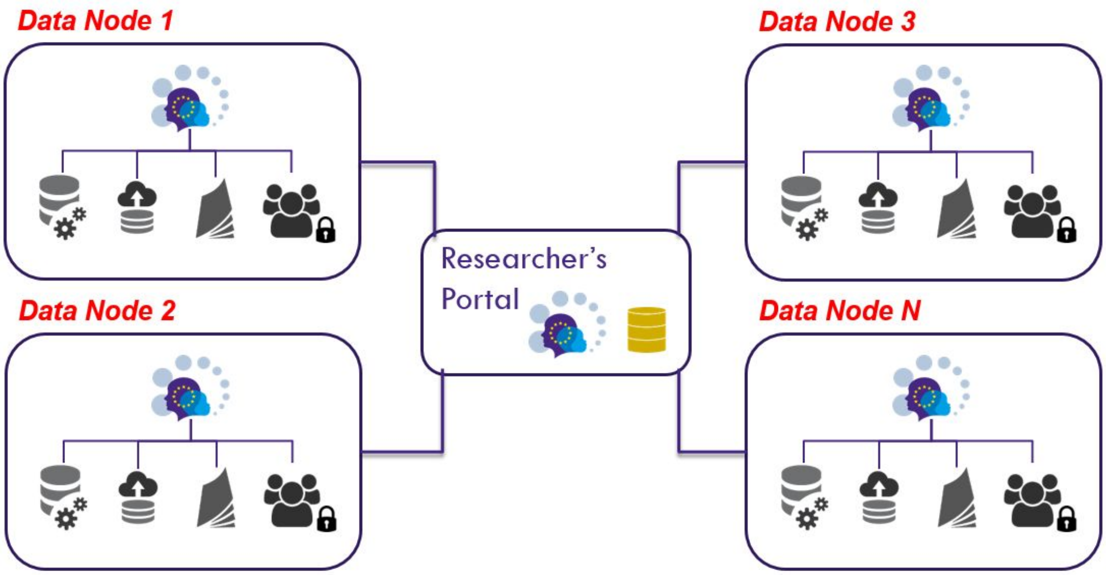


Figure 6: High-level view of the RECAP preterm network

The network uses a federated architecture in which **distributed** nodes have full control over their local users, data and the software that allows them to curate, describe and publish metadata about selected datasets and studies, and keep absolute control over local and remote access to the data.

Currently under development, the **Researcher's Portal** will serve as a common entry point for the whole network, providing global browsing of metadata and variable searching functionality in addition to supporting the steps to establish new collaborative research initiatives, from the definition of a hypothesis to the final decision on the approval of a new study. Furthermore, if an agreement between all parties in the study is reached, derived datasets and other outputs can reside here for the duration of the study.

2.2. Data, Data Dictionaries and Study Metadata

Currently, a central RECAP node hosted at INESC TEC is being used to facilitate the gathering of dictionaries and study metadata on the scope of WP3.
The central node can be described as a proxy for the nodes in the network and thus no actual data is kept there. It is kept instead by each partner at their own node. On the other hand, the location of the data dictionaries and study metadata will change along the run of the project.

In the future, all dictionaries and study metadata will be copied to their respective nodes. The dictionaries will then be associated with the actual data on the node.
Thus, only the study metadata will be kept at a central location and will contain links (without transferring data) to the respective nodes, allowing for regular synchronizations with the metadata at each of the nodes in the network.

The idea of imparting RECAP nodes with the ability to store not only data but also their own metadata, is to allow each node to function independently from the overall network, effectively granting partners complete autonomy and sovereignty over their (meta)data even after the temporal extent of RECAP.

The central catalogue will then operate as described in Figure 7.

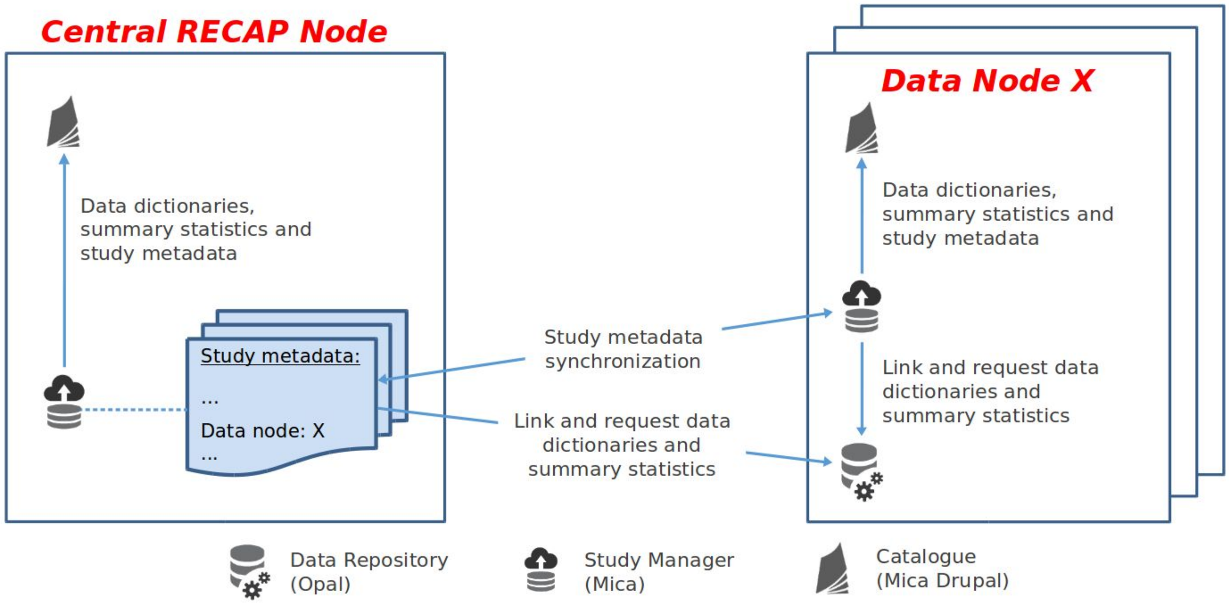

Central RECAP Node
Data dictionaries, summary statistics and study metadata
Study metadata:
...
Data node: X
...
Study metadata synchronization
Link and request data dictionaries and summary statistics
Data Node X
Data dictionaries, summary statistics and study metadata
Link and request data dictionaries and summary statistics
Data Repository (Opal)
Study Manager (Mica)
Catalogue (Mica Drupal)

Figure 7: How the central Catalogue sources information from the RECAP nodes