



D2.4

DAC Portal prototype, validated analytical workflows, analysis prototype, updated metadata standards and portal prototype

Project number	826121
Project acronym	iPC
Project title	individualizedPaediatricCure: Cloud-based virtual-patient models for precision paediatric oncology
Start date of the project	1 st January, 2019
Duration	53 months
Programme	H2020-SC1-DTH-2018-1

Deliverable type	Demonstrator
Deliverable reference number	SC1-DTH-07-826121 / D2.4 / 1.0
Work package contributing to the deliverable	WP2
Due date	31st May, 2022
Actual submission date	30th May, 2022

Responsible organisation	XLAB, BSC
Editor	Jolanda Modic (XLAB), Salvador Capella Gutierrez (BSC)
Dissemination level	PU
Revision	V1.0

Abstract	We report on the selection of the appropriate data models to handle the available data and metadata to the iPC Central Computational and Data platform. We also report on the current status of the development for the iPC Data portal.
Keywords	Metadata; Data models; Standards; Data Catalogue



Editor

Jolanda Modic (XLAB)

Salvador Capella-Gutiérrez (BSC)

Contributors (ordered according to beneficiary numbers)

Alejandro Canosa (BSC)

Dmitry Repchevsky (BSC)

Frederic Haziza (EGA/CRG)

Roberto Ariosa (EGA/CRG)

José María Fernández (BSC)

Laia Codó (BSC)

Alfonso Valencia (BSC)

Mihael Trajbarič (XLAB)

Dejan Štepec (XLAB)

Alvaro Garcia Faura (XLAB)

Disclaimer

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The users thereof use the information at their sole risk and liability.

This document has gone through the consortium's internal review process and is still subject to the review of the European Commission. Updates to the content may be made at a later stage.

Executive Summary

This document focuses on one of the core building blocks of the iPC platform, namely the Data Access Framework devoted to the management of access rights for sensitive datasets included in the iPC Data Catalogue (presented in D2.3).

First, we provide a high-level overview of the Data Access Framework by discussing its position and role within the iPC platform (**Chapter 2**). We provide a short description of the main tasks of the Data Access Framework, namely to manage Data Access Committees (DACs; data owners who grant/reject/revoke access to sensitive datasets to the iPC users) and to manage data access requests from the iPC users.

Afterwards, we provide implementation details for the Data Access Framework from the perspective of its components and interfaces (**Chapter 3**). Specifically, we provide details for the following components comprising the Data Access Framework:

- DAC-Portal: Enables the different members of a DAC to manage access rights for specific datasets under their control.
- Permissions-API: Allows the different types of DAC users (DAC-admin, DAC-member) to read, write, and delete user's file permissions under their control.
- DAC-Management-Portal: Enables system administrators to create brand new DACs, manage DAC-roles and DAC-resources, and also validate the public information from specific DACs.
- DAC-Notify: Is in charge of sending notifications to the different players of the iPC's Data Access Framework (e.g., DAC-admin, data requesters, etc.) which are triggered by events.
- Keycloak: Controls Authentication and Authorization (AuthN/Z) within the iPC Platform.

Interactions among these components are presented in **Chapter 4** through several use cases demonstrating the data submission and the generation of the DAC for the submitted dataset, the submission of a request to access this dataset and the review process that follows, and the management of the DAC roles and resources.

We conclude the report with a summary (**Chapter 5**) and a short elaboration on the future work (**Chapter 6**) that includes various development activities, as well as integration with external systems.

The work presented in this document builds on previous efforts and results presented in D2.2 *Initial infrastructure framework* and D2.3 *Recommended metadata standards and portal prototype*.

As this report D2.4 aligns with the completion of task T2.3 *Implement of a framework to facilitate communication with Data Access Committees*, we focus this report solely on the presentation of the Data Access Framework, whereas the analytical workflows and analysis prototypes, which have yet to be completed and validated, will be presented in the final WP2 deliverable D2.5 *Global Report on iPC Computational Infrastructure* due at the end of the project (at M53).

Table of Content

Chapter 1	Introduction	1
Chapter 2	Data Access Framework Overview	3
Chapter 3	Components	4
3.1	DAC-Portal	4
3.2	Permissions-API.....	4
3.3	DAC-Management-Portal.....	4
3.4	DAC-Notify	5
3.5	Keycloak (AuthN/Z)	5
Chapter 4	Uses Cases	7
4.1	Data Submission and DAC Generation.....	7
4.2	Data Access Requests and Review Process	10
4.3	DAC Roles and Resources Management	13
Chapter 5	Summary of Achievements	15
Chapter 6	Future Work	16
6.1	Development Tasks	16
6.2	Deployment in Production	16
6.3	Integration to External Systems	17
6.3.1	Integration of the EGA Data Access Framework.....	17
6.3.2	Integration of Cavatica.....	19

List of Figures

Figure 1: High-level architecture of the iPC Computational Platform.	3
Figure 2: Hybrid RBAC/ABAC authorization model for the iPC Data Access Framework.	6
Figure 3: A data submitter accesses her / his personal dashboard in Nextcloud.	7
Figure 4: The data submitter uploads files to Nextcloud.	8
Figure 5: A super administrator enters the DAC-Management-Portal.	8
Figure 6: The super administrator selects the resources and users that are going to administrate this specific DAC.	9
Figure 7: One of the users who have been granted a DAC-admin role accesses the DAC-Portal. ...	9
Figure 8: The DAC-admin reviews her / his DAC's information and submits relevant data about her / his DAC.	10
Figure 9: The DAC-admin applies a DUO code in one of the available datasets.	10
Figure 10: An iPC user finds an interesting dataset in the iPC Data Catalogue Portal.	11
Figure 11: The user requests access to the dataset.	11
Figure 12: The data requester describes why she / he needs access to this particular file (resource) and verifies the policies attached to this specific resource.	12
Figure 13: The DAC-admin accepts the submitted data access request.	12
Figure 14: The DAC-admin grants access to the dataset.	13
Figure 15: The "Manage resources" section in the DAC-Management-Portal panel.	13
Figure 16: The "Manage DAC roles" section from the DAC-Management-Portal panel where the super administrator changes the role of the "demo-user-3".	14
Figure 17: The personal dashboard of a DAC-member user in the iPC DAC-Portal.	14
Figure 18: iPC Computational Platform proposed architecture.	18

List of Abbreviations

Abbreviation	Translation
ABAC	Attribute-based access control
AMQP	Advanced Message Queuing Protocol
AuthN	Authentication
AuthZ	Authorization
CI	Continuous integration
CD	Continuous delivery/deployment
CHOP	Children's Hospital of Philadelphia
DAC	Data Access Committee
EGA	European Genome-phenome Archive
FEGA	Federated European Genome-phenome Archive
GA4GH	Global Alliance for Genomics and Health
iPC	individualized Paediatric Cure
JWT	JSON Web Token
NIH	National Institutes of Health
OIDC	OpenID Connect
RAS	Researcher Auth Service
REST	Representational State Transfer
SSO	Single-Sign-On
SSR	Server Side Rendering
VRE	Virtual Research Environment

Chapter 1 Introduction

One of the main objectives of the iPC project is to gather, integrate, harmonise, and share a wide list of existing data sources relevant to paediatric cancer research. To this end, we are developing a cloud-based platform to host newly produced datasets, link to other similar archives hosting paediatric cancer data (such as the European Genome-phenome Archive (EGA¹), the Kids First Data Portal (Kids First²), and the R2: Genomics Analysis and Visualisation Platform (R2³)), and then offer means to process them.

To make the heterogeneous datasets quickly findable and easily usable regardless of the data type, format, origin, the disease they describe or the profile of the patient they relate to, we have developed the iPC Data Catalogue introduced in deliverables D2.2 *Initial infrastructure framework* and D2.3 *Recommended metadata standards and portal prototype*. This report D2.4 presents our work on a system that allows the iPC users to **easily and securely access the datasets** included in the iPC Data Catalogue. Note that only authenticated users with an iPC account are allowed to perform data access requests via the iPC Data Catalogue, and that only such authenticated iPC users can process the datasets in the iPC Virtual Research Environment.

Some of the more relevant platforms that host and disseminate cancer research data are tending to adopt a login flow based on OpenID Connect (OIDC) for achieving interoperability between similar platforms across institutions and even borders. For example, USA-based platforms (such as Kids First) has implemented an authentication and authorization infrastructure based on the Researcher Auth Service Initiative (RAS⁴) system where several institutional identity providers are permitted, whereas EU-based platforms (such as EGA) are leveraging ELIXIR-AAI Life Sciences identity provider for enabling an institutional login across the different Federated EGA (FEGA⁵) nodes. Further, to get access to a controlled (i.e., not publicly open) dataset on these platforms, one needs to submit a data access request, which is (as shown in the guidelines for the EGA example⁶) reviewed and granted (or denied) by the affiliated **Data Access Committee (DAC)** - a body of one or more named individuals responsible for all data access decisions based on patients' consents and established legal/ethics terms.

One of the key challenges in improving data sharing and thereby contributing to more intense and more efficient collaboration in cancer research, is integrating and, in the eyes of the user, simplifying (private) data access control mechanisms across different platforms, while maintaining high levels of security and compliance with organisational security policies, ethical standards, and data protection laws. To this end, the long term goal of iPC is to interoperate with other systems, such as NIH (National Institutes of Health; the US medical research agency) or EGA, leveraging the OIDC protocol and use the Global Alliance for Genomics and Health (GA4GH) Passports/Visas specification (built on top of OIDC) for granting access to datasets.

The iPC Data Access Framework presented in this document is based on the grounds of GA4GH Passport specification⁷, which defines a standard way of representing data access rights on the basis of user's Passport and enclosed Passport Visas (GA4GH AAI OIDC Profile⁸) that contain human

¹ <https://ega-archive.org/>

² <https://portal.kidsfirstdrc.org/>

³ <https://hgserver1.amc.nl/cgi-bin/r2/main.cgi>

⁴ <https://datascience.nih.gov/researcher-auth-service-initiative>

⁵ <https://ega-archive.org/federated>

⁶ <https://ega-archive.org/access/data-access>

⁷ https://github.com/ga4gh-duri/ga4gh-duri.github.io/blob/master/researcher_ids/ga4gh_passport_v1.md

⁸ <https://github.com/ga4gh/data-security/blob/master/AAI/AAIConnectProfile.md>

readable and reliable information about the user's affiliations, roles, accepted terms, grants, etc. (see also GA4GH Passport and AAI specifications⁹).

In summary, this specification provides a federated (multilateral) authentication and authorisation infrastructure, enabling greater interoperability between different institutions and platforms in a manner specifically applicable to the access of restricted datasets.

By integrating various data access control infrastructures and following community-set standards, iPC will make a big step forward in simplifying the access of private data sources for cancer research, thereby supporting a wide community of experts in creating effective personalised therapies for kids with cancer.

⁹ https://docs.google.com/presentation/d/1HhOXWvF87fihWBY7wHmuAwS_ydII_gQWFopwHcCEpE8/edit

Chapter 2 Data Access Framework Overview

The iPC Platform hosts different (private) datasets that are listed in the **iPC Data Catalogue** and stored in **Nextcloud**¹⁰. Once an iPC user logs into the Platform through **Keycloak**, he/she can only access datasets for which he/she has appropriate permissions where all permissions are handled through the **Data Access Framework**.

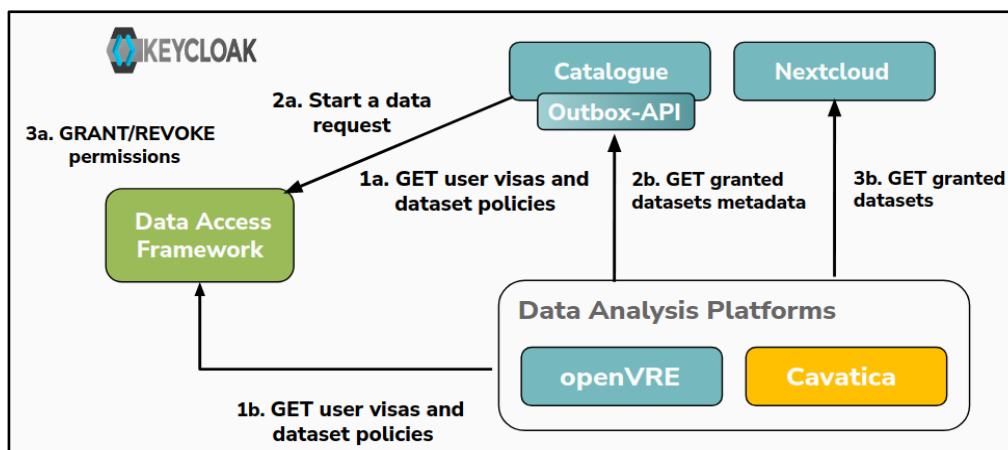


Figure 1: High-level architecture of the iPC Computational Platform.

The light blue colour represents elements included in the initial platform implementation (Deliverable 2.1), the green colour represents the Data Access Framework that includes a plethora of new components (shortly presented on Deliverable 2.3 and implemented within the work on Deliverable 2.4), and the yellow colour a tentative integration with an external analysis platform (see Chapter 6).

The Data Access Framework has the role of (1) **managing Data Access Committees** (DACs; i.e., from the creation of brand new DACs, to the roles, rights, and resources management for those users who own private datasets and who determine access rights and conditions for those private datasets) and (2) **managing data access requests** (approving/denying access requests, approving/revoking access rights). The details and interactions of the components comprising the Data Access Framework are presented in the following two Chapters, respectively.

Note that all components within the Data Access Framework are compatible with the rest of the iPC Platform components (i.e., can be deployed together with the rest of the components of the iPC Platform, even if they have their own development lifecycle), but are also independent in order to ensure maintainability and reusability of the components within different environments. Besides, the iPC Data Access Framework aims to be interoperable between different systems, including external Data Access Frameworks and/or Data Analysis platforms. This is important for eventually having additional data sources and analytical tools available to the end users through the iPC Platform (see Chapter 6.).

¹⁰ <https://data.ipc-project.bsc.es>

Chapter 3 Components

The iPC's Data Access Framework components are presented in this section. A general introduction for each of these components is provided, and also, more technical details with focus on the current implementation.

3.1 DAC-Portal

The DAC-Portal is a service where the different members of a DAC can give access to external users to specific files (resources) under their control. Two specific roles have been designed to perform all the DAC related operations, DAC-admins and DAC-members. On the one hand, DAC-admins are able to set up data policies for specific files, to provide information about their DAC (publicly accessible via REST), to invite others to join the DAC, and importantly, to manage user's permissions over time. On the other hand, either DAC-members or DAC-admins can accept/reject incoming data access requests started from the iPC's Data Catalogue Portal, and set up the requested permissions from the DAC-Portal user interface via Permissions-API.

The DAC-Portal is a full-stack web application where the integration between the user interface and the backend system is done via RESTful-API. The latter fuels the user interface with all the DAC-related relevant data, including requests, resources, memberships, users, and data policies.

The user interface is built with React Hooks and HTML5/SCSS (Single-Page Application), and the backend is based on an Express/NodeJS web-server with a MongoDB database. The implementation is available on GitHub¹¹.

3.2 Permissions-API

The Permissions-API service allows the different types of DAC users (DAC-admin, DAC-member) to read, write, and delete user's file permissions under their control. Besides, the external users, such as the iPC's Data Catalogue users, are only allowed to read their own files permissions. Since this service is compliant with GA4GH Passports/Visas specification, users will not only be able to consume files within the iPC services (e.g. open Virtual Research Environment), but they will potentially interoperate with external platforms and services acting as data consumers (e.g. Cavatica - CHOP).

The Permissions-API is a RESTful-API built with Express/NodeJS web-server with a MongoDB database, which serves user's files permissions in the form of visas according to the GA4GH specification.

The current implementation is available on GitHub¹².

3.3 DAC-Management-Portal

The DAC-Management-Portal is the place where system administrators can create brand new DACs, manage DAC-roles and DAC-resources, and also validate the public information from specific DACs. To achieve this, the DAC-Management-Portal has not only been integrated with the iPC's data submission system based on Nextcloud, but also with the DAC-Portal database. The former provides all the information about data submissions performed by specific user's groups (or institutions) via

¹¹ <https://github.com/inab/DAC-Portal.git>

¹² <https://github.com/inab/Permissions-API>

WebDAV protocol, and the latter, gives the information related to the different DACs, that can be modified and updated in an interactive way by the system administrators through the DAC-Management-Portal.

The DAC-Management-Portal is a full stack web application built on Next.js that provides SSR and RESTful capabilities, and which is linked to different databases. Given the nature of the different operations handled by this service, the general AuthN/Z system based on OIDC has been disabled, and instead, a restricted local AuthN/Z setup for system's administrators has been put in place, who are able to perform reading and writing operations at a database level.

The current implementation is available on GitHub¹³.

3.4 DAC-Notify

The DAC-Notify component is in charge of sending notifications to the different players of the iPC's Data Access Framework (e.g DAC-admin, data requesters, etc) which are triggered by events. This can be accomplished by offering a common communication system between the different applications of the Data Access Framework with end-users. To this end, a message broker based on RabbitMQ (AMQP protocol) and also an SMTP service have been deployed. The implementation follows the publish/subscribe pattern, where the different components (e.g. DAC-Portal, Permissions-API, ...) are able to send messages triggered by events (publishers), that are queued and delivered by the message broker to the proper consumer services (subscribers), that handle the different messages payload, and eventually, send formatted (HTML/CSS) emails with the essential information to specific users.

The current implementation is available on GitHub¹⁴.

3.5 Keycloak (AuthN/Z)

Keycloak controls Authentication and Authorization (AuthN/Z) within the iPC Computational Platform. In the first case, Keycloak supports OIDC and a SSO flow can be established through all the web-based platform's components with a standard username/password login mechanism. In the second case, since the OIDC protocol is built on top of the OAuth2, a model where protected resources can be accessed using JWT is employed.

The current implementation of Authorization in the iPC's Data Access Framework is based on a hybrid RBAC/ABAC authorization model, where the access to the different resources is decided by evaluating both user's roles (e.g. DAC-admin) and attributes (e.g. resources). To this end, Keycloak has been adapted for retrieving such information from the DAC-Portal database, and also, for including that information as claims within dedicated scopes into the final JWT. Additionally, the different services providing REST access (e.g. Permissions-API, DAC-Portal-API, ...) have been protected (against unauthorised requests made by malicious users) with middleware components that evaluate incoming JWT based on user roles and resource level access.

Figure 2 below illustrates the hybrid RBAC/ABAC authorisation model for the iPC Data Access Framework. After a user is authenticated (step 1) in a web-based service (e.g., the iPC Data Catalogue Portal or the DAC-Portal), the AuthN/Z server based on Keycloak retrieves the information about the user's roles and controlled resources from the DAC-Portal database (step 2) and appends these as claims in the JWT. Then, a client performs a request to protected resources (step 3), and the JWT is verified (step 4) by Keycloak (OAuth2). If the JWT is valid, user roles and resources claims are evaluated (authorization middleware) to determine whether the user is authorised to

¹³ <https://github.com/inab/DAC-Management-Portal.git>

¹⁴ <https://github.com/inab/DAC-Notify.git>

perform an action on the selected resources or not. If authorised, the user will be able to perform actions on the selected resources (e.g., read, create, and modify users permissions and/or specific DACs data).

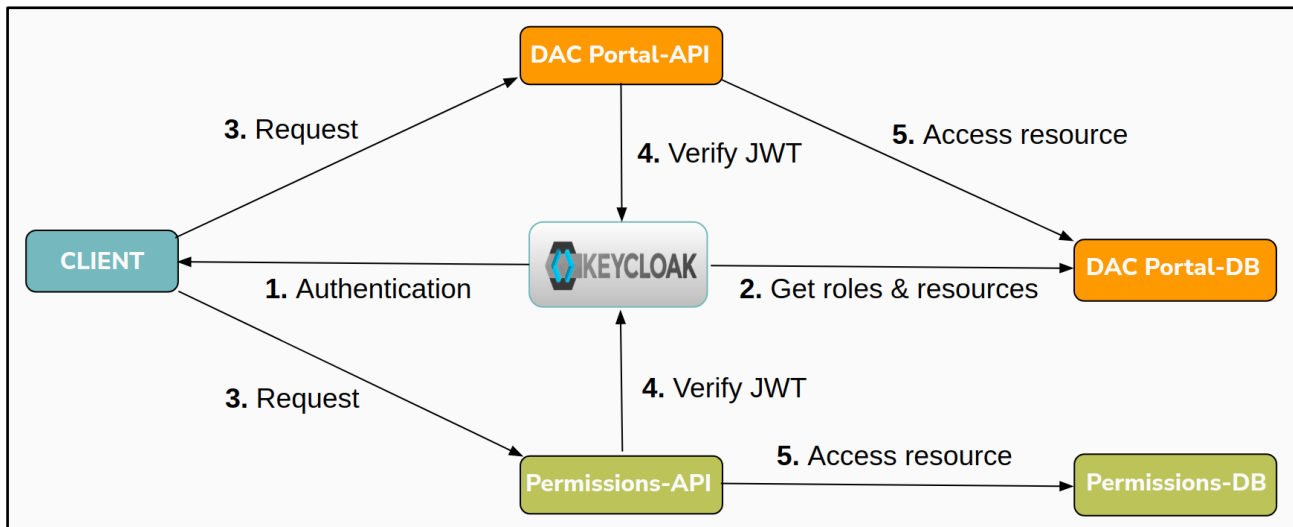


Figure 2: Hybrid RBAC/ABAC authorization model for the iPC Data Access Framework.

Chapter 4 Uses Cases

This Chapter presents the key use cases for the Data Access Framework, demonstrating the interactions among the components of the Data Access Framework.

4.1 Data Submission and DAC Generation

Description: A data owner who is an iPC user would like to add new datasets to the iPC Data Catalogue (Nextcloud), define its Data Access Committee, and set an access policy for the submitted dataset.

Actors: The data owner (data submitter) who becomes the DAC administrator (DAC-admin) for the submitted datasets, DAC-members (i.e., other users that are members of the DAC for the uploaded datasets), and the iPC super administrator who defines and manages initial DAC-admins.

Note that the **DAC-admin** and the **DAC-members** have the same rights in terms of reviewing / granting / revoking access rights for the datasets under the control of the DAC they belong to. The difference between these two roles is that the DAC-admin can configure rights of other DAC-members (can invite or remove DAC members), and also, set up both the datasets policies and/or the publicly available information about their DACs. The **iPC super administrator** is the role that is able to create DACs, to review DACs information, and also, to manage either the DAC roles or the assigned resources at any time.

Flow: The data owner submits new datasets to the iPC platform. Specifically, she / he accesses her/his personal dashboard in Nextcloud (i.e., the iPC Data Catalogue raw data storage system) and is able to see a specific submission folder for her / his institution named “DEMONSTRATOR” as illustrated in Figure 3. Afterwards, she/he accesses the institution’s submission folder and uploads five files (demo-file-X.txt) as shown in Figure 4.

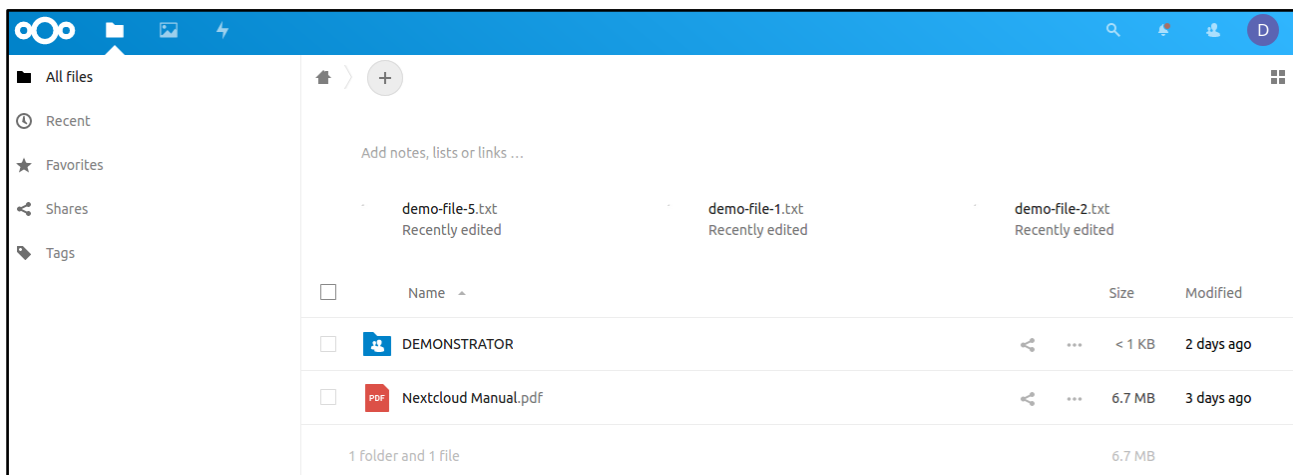


Figure 3: A data submitter accesses her / his personal dashboard in Nextcloud.

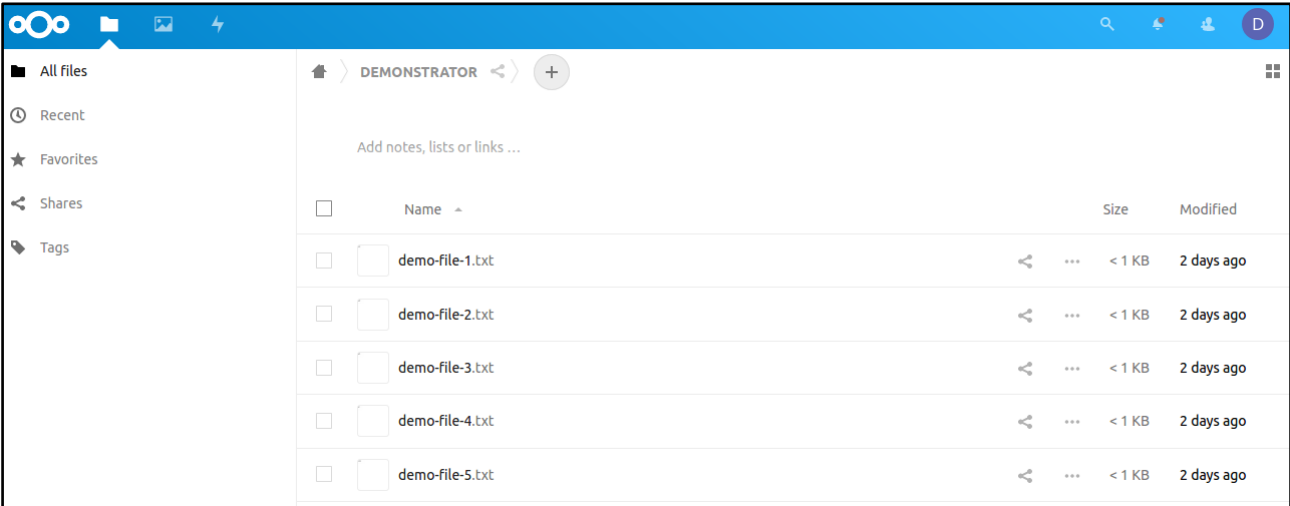


Figure 4: The data submitter uploads files to Nextcloud.

Next, the iPC’s super administrator creates a new DAC and defines administrators for this DAC and the resources under their control. Specifically, the super administrator enters the DAC-Management-Portal and accesses the “Create a DAC” section from the main administrator panel as seen in Figure 5. Afterwards, she/he is able to select both resources and users that are going to administrate (DAC-admin role) this specific DAC (for the data uploaded to the submission folder “DEMONSTRATOR”). As shown in Figure 6, the DAC-Management-Portal obtains metadata from the data submission system (Nextcloud), and shows users with access to the “DEMONSTRATOR” submission folder (same institution) and files that have been uploaded into it. The super administrator decides to give a DAC-admin role to some of these users (demo-user-1, demo-user-2, demo-user-3) and assigns all the uploaded files (resources) to this specific DAC (see Figure 3).

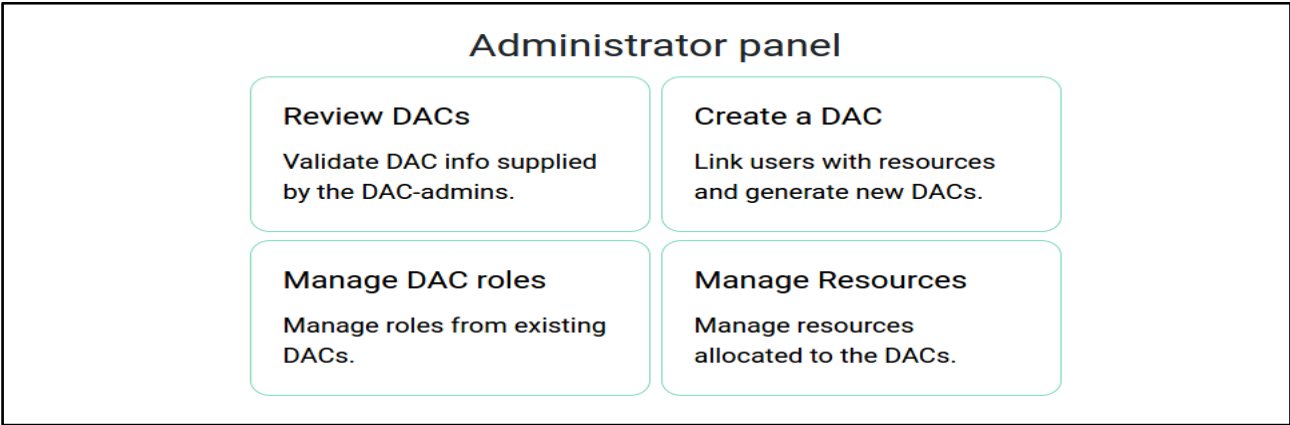


Figure 5: A super administrator enters the DAC-Management-Portal.

Create DAC - DEMONSTRATOR

Roles

Select administrators for this DAC

demo-user-1
demo-user-2
demo-user-3

Select

Resources

Select resources for this DAC

demo-file-1.txt
demo-file-2.txt
demo-file-3.txt

demo-file-4.txt
demo-file-5.txt

Select

Send

Figure 6: The super administrator selects the resources and users that are going to administrate this specific DAC.


Finally, one of the defined DAC-admins sets access policies for the submitted datasets by entering the DAC-Portal and using the dashboard shown in Figure 7. As seen in Figure 8, the DAC-admin navigates to the “CREATE A DAC” section to review her / his DAC’s information and verifies that she / he can select a DAC (IPC00000000005) called “DEMONSTRATOR”. The DAC-admin decides to fill in the form and submit relevant data about her / his DAC, that will be accessible via REST once a super administrator validates the data (DAC-Management-Portal). Finally, as illustrated in Figure 9, the DAC-admin navigates to “MY POLICIES” section and decides to apply a DUO code (DUO-3845) in one of the datasets available (total files or resources: 5 - See Figure 6) which are referenced with their file identifier (e.g.: demo-file-1.txt -> 323, demo-file-2.txt -> 324, etc. - See Figure 6).

DAC PORTAL


- USER PROFILE
- MY DACS
- CREATE A DAC
- MY POLICIES
- MANAGE PERMISSIONS
- MANAGE REQUESTS
- NEED HELP?

Dashboard
Logout


Welcome to the iPC DAC Portal



Registered DACs
5

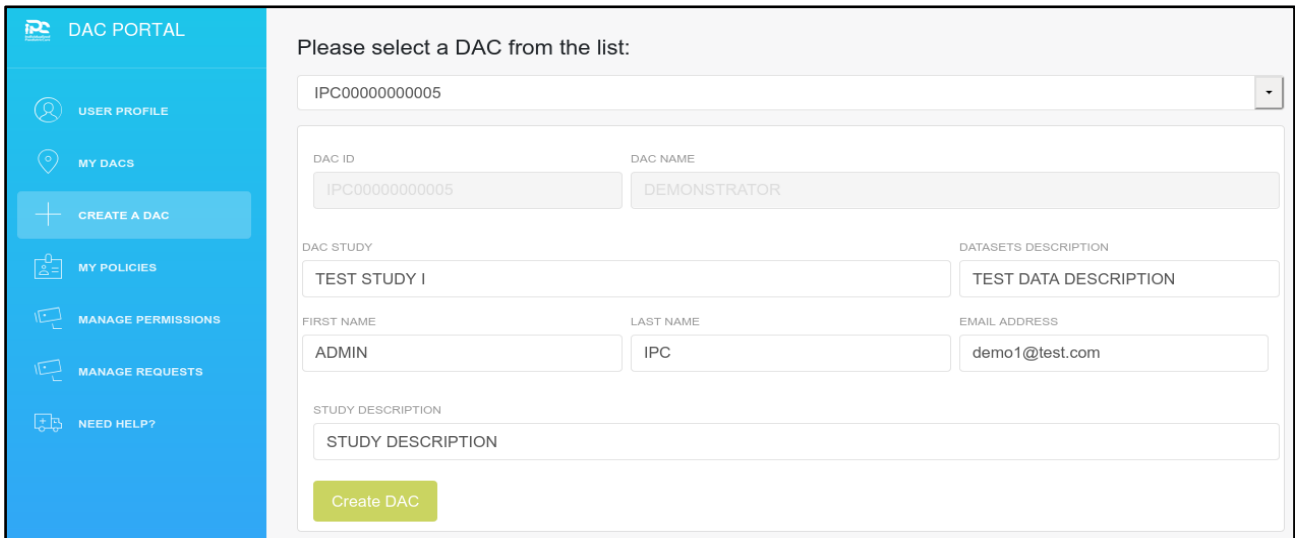


Registered Datasets
992



Total users
23

Figure 7: One of the users who have been granted a DAC-admin role accesses the DAC-Portal.



DAC PORTAL

USER PROFILE
MY DACS
+ CREATE A DAC
MY POLICIES
MANAGE PERMISSIONS
MANAGE REQUESTS
NEED HELP?

Please select a DAC from the list:

IPC000000000005

DAC ID: IPC000000000005 | DAC NAME: DEMONSTRATOR

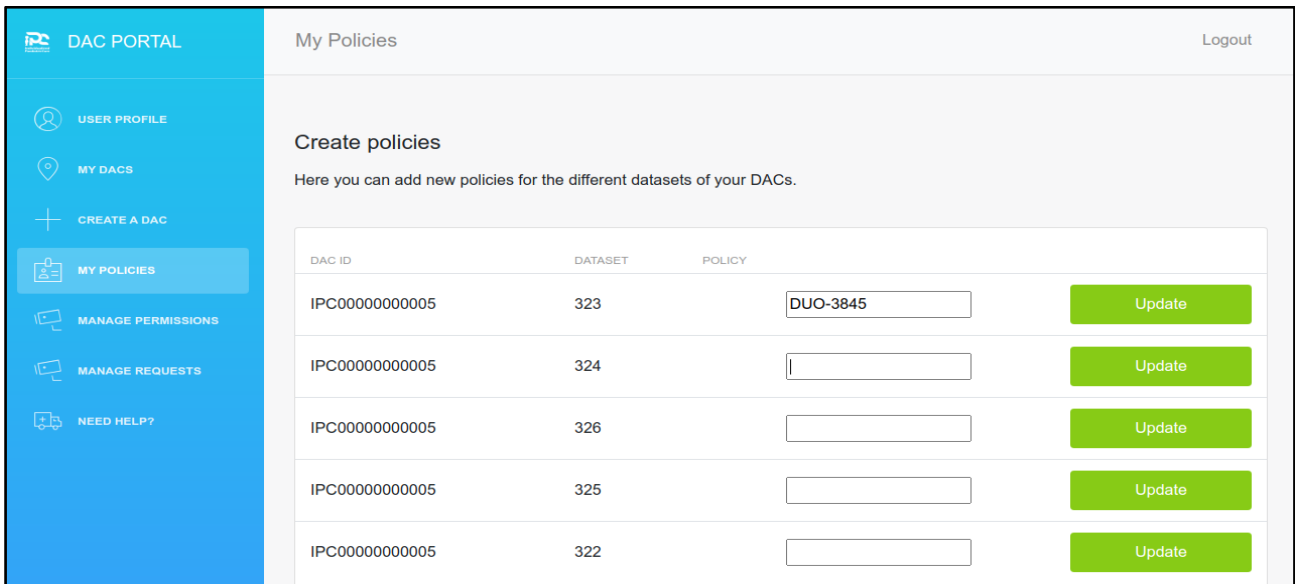
DAC STUDY: TEST STUDY I | DATASETS DESCRIPTION: TEST DATA DESCRIPTION

FIRST NAME: ADMIN | LAST NAME: IPC | EMAIL ADDRESS: demo1@test.com

STUDY DESCRIPTION: STUDY DESCRIPTION

Create DAC

Figure 8: The DAC-admin reviews her / his DAC's information and submits relevant data about her / his DAC.



DAC PORTAL

My Policies Logout

Create policies

Here you can add new policies for the different datasets of your DACs.

DAC ID	DATASET	POLICY	
IPC000000000005	323	DUO-3845	Update
IPC000000000005	324		Update
IPC000000000005	326		Update
IPC000000000005	325		Update
IPC000000000005	322		Update

Figure 9: The DAC-admin applies a DUO code in one of the available datasets.

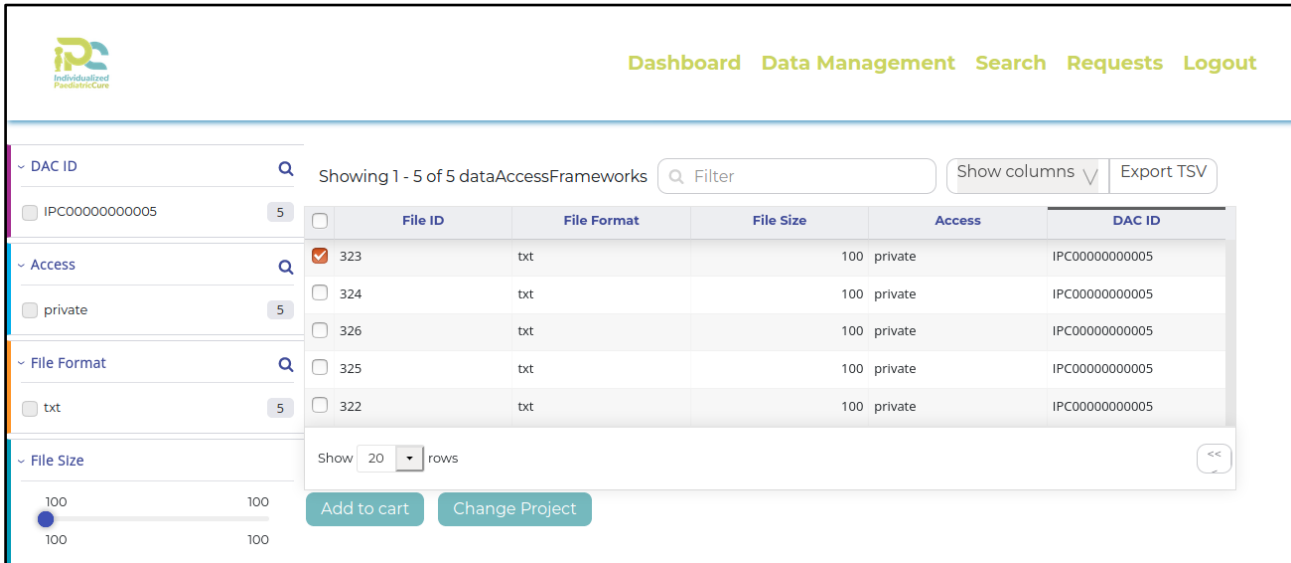
4.2 Data Access Requests and Review Process

Description: An iPC user would like to access a private dataset. She / he fills out the data access request form and submits it to the DAC portal for the relevant dataset. A DAC-admin reviews the request and approves it.

Actors: A data requester that submits an access request for a specific dataset and its DAC-admin.

Flow: A data requester submits an access request for a specific dataset found in the iPC Data Catalogue. Specifically, as illustrated in Figure 10, she / he accesses the iPC Data Catalogue Portal, navigates to the Search section, and then adds to the cart a dataset controlled by the DAC "IPC000000000005" with a file identifier "323". As seen in Figure 11, she / he navigates to the "Data Management" section in the iPC Data Catalogue and sees that she / he is not able to upload the selected dataset into the analysis platforms (e.g., Virtual Research Environment). Then, she / he decides to request access to this dataset by clicking the "Request Access" button. For the request

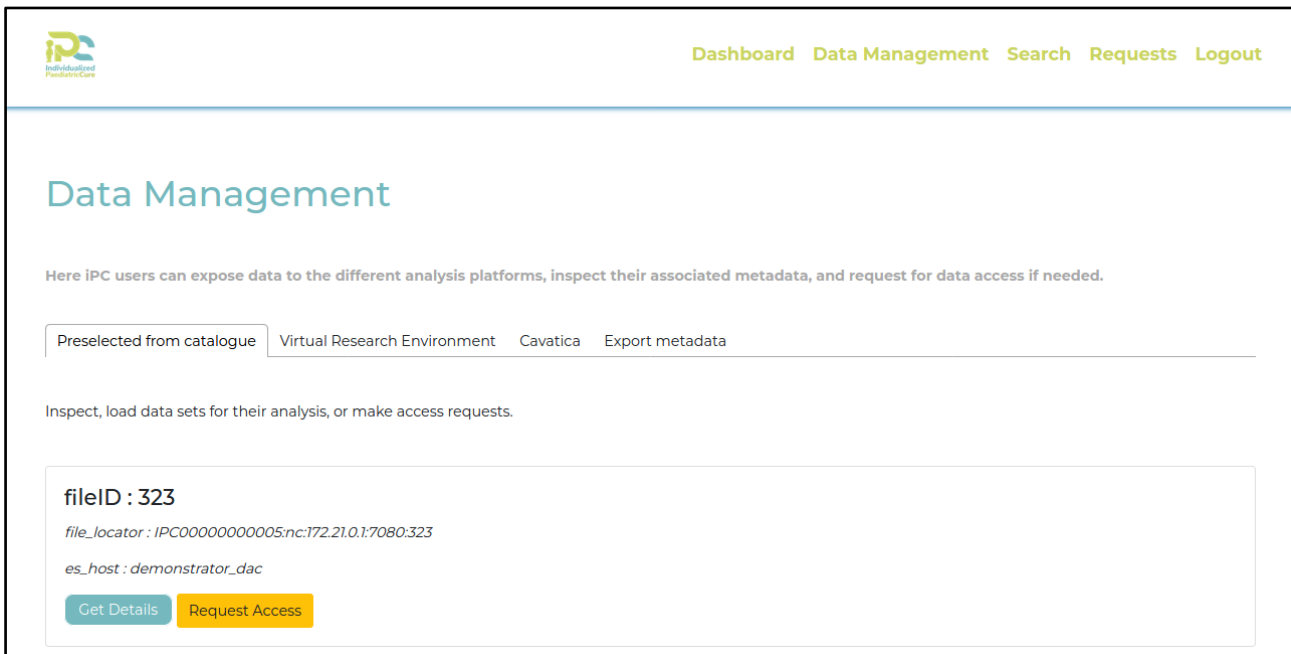
to be processed, she / he explains why she / he needs access to this particular file (resource) and verifies the policies attached to this specific resource (as seen in Figure 12).



The screenshot shows the iPC Data Catalogue Portal interface. The top navigation bar includes links for Dashboard, Data Management, Search, Requests, and Logout. The main content area displays a list of datasets with filters on the left and a table of results. The filters include DAC ID (set to IPC000000000005), Access (set to private), File Format (set to txt), and File Size (set to 100). The table shows 5 datasets, with the first one selected. The table columns are File ID, File Format, File Size, Access, and DAC ID.

File ID	File Format	File Size	Access	DAC ID
323	txt	100	private	IPC000000000005
324	txt	100	private	IPC000000000005
326	txt	100	private	IPC000000000005
325	txt	100	private	IPC000000000005
322	txt	100	private	IPC000000000005

Figure 10: An iPC user finds an interesting dataset in the iPC Data Catalogue Portal.



The screenshot shows the iPC Data Management page. The top navigation bar includes links for Dashboard, Data Management, Search, Requests, and Logout. The main content area is titled 'Data Management' and includes a description: 'Here iPC users can expose data to the different analysis platforms, inspect their associated metadata, and request for data access if needed.' Below this, there are tabs for 'Preselected from catalogue', 'Virtual Research Environment', 'Cavatica', and 'Export metadata'. The 'Preselected from catalogue' tab is active, showing details for fileID 323. The details include the file locator and the host. At the bottom, there are buttons for 'Get Details' and 'Request Access'.

fileID : 323

file_locator : IPC000000000005;nc:172.21.0.1:7080:323

es_host : demonstrator_dac

Get Details Request Access

Figure 11: The user requests access to the dataset.

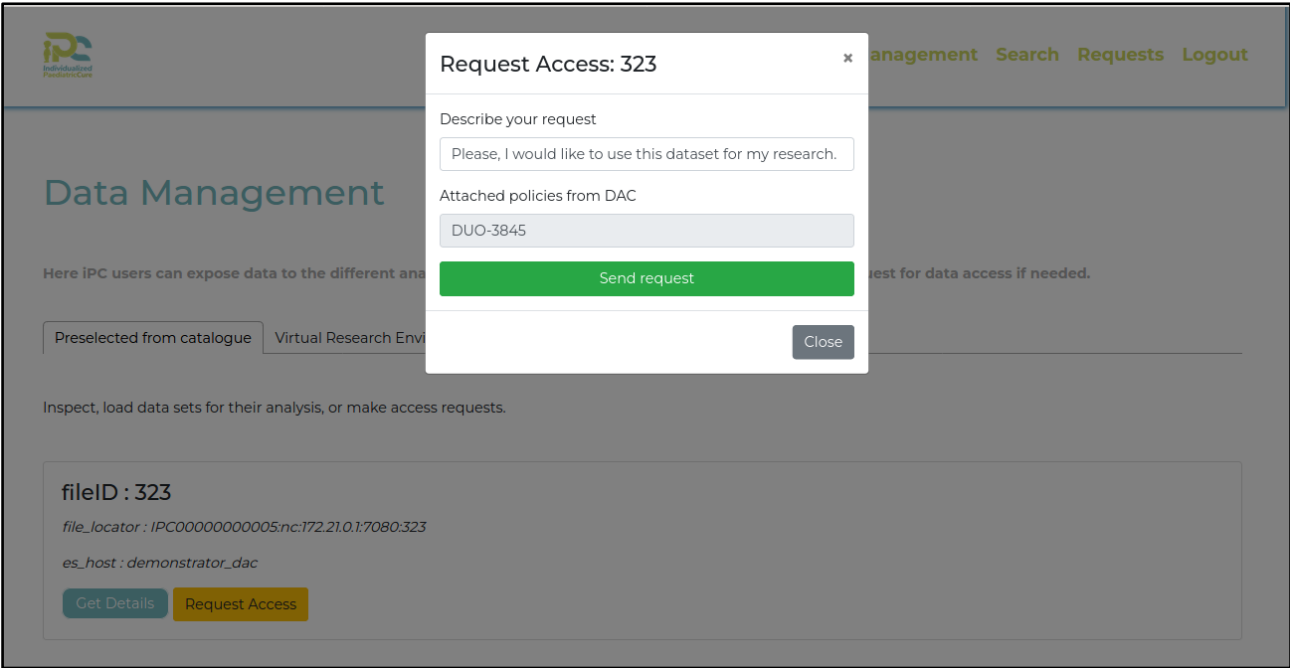


Figure 12: The data requester describes why she / he needs access to this particular file (resource) and verifies the policies attached to this specific resource.

After the data access request has been submitted, the relevant DAC-admin reviews the request and grants access to the requested dataset.

Specifically, as illustrated in Figure 13, the DAC-admin navigates to the “MANAGE REQUESTS” section and sees there is a new request corresponding to the file identifier “323” that is controlled by a DAC (See Figure 12). In this case, the DAC-admin decides to accept the request by clicking the “Grant” button (a new file permission is stored at the Permissions-API according to the GA4GH specification). Afterwards, as shown in Figure 14, the DAC-admin navigates to the “MANAGE PERMISSIONS” section and verifies that the granted permission¹⁵ has been properly assigned to the requester, who now will be able to load this file to the analysis platforms from the iPC’s Data Catalogue Portal.

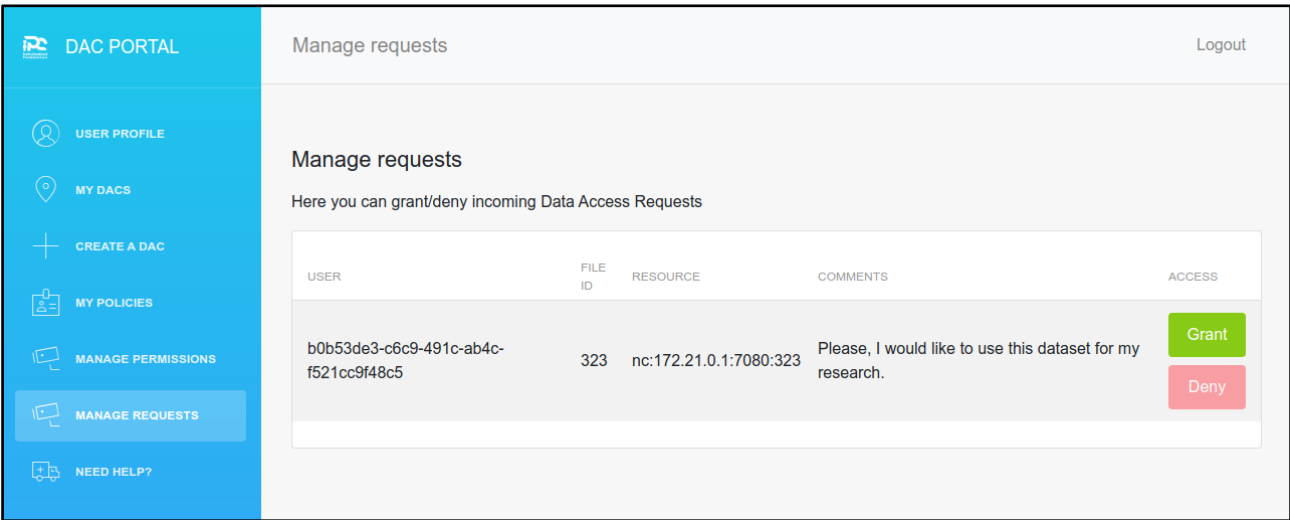


Figure 13: The DAC-admin accepts the submitted data access request.

¹⁵ DAC-admins can also revoke individual file permissions which are controlled by their DAC/s.

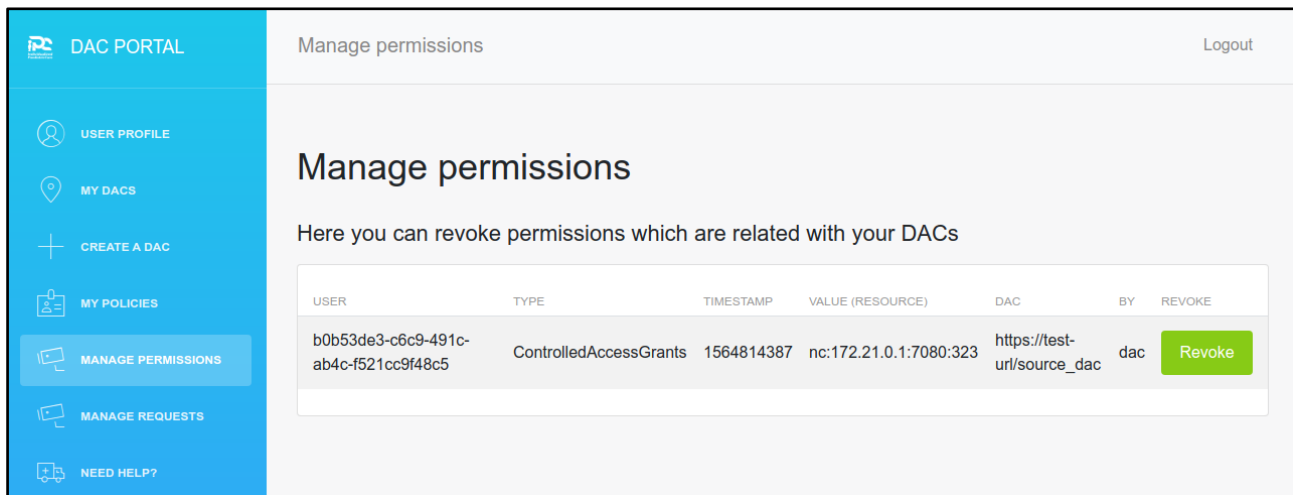


Figure 14: The DAC-admin grants access to the dataset.

4.3 DAC Roles and Resources Management

Description: The iPC's super administrator would like to manage / change DAC memberships and roles, and inspect datasets controlled by a specific DAC.

Actors: The iPC's super administrator that updates rights for a DAC-admin and the DAC-admin in question.

Flow: The super administrator removes the DAC-admin role of an iPC user and replaces this role with the role of a DAC-member. Specifically, as shown in Figure 16, the super administrator navigates to the "Manage resources" section from the DAC-Management-Portal panel. The super administrator can view all resources assigned to a particular DAC (DAC -> "DEMONSTRATOR") and decide to add/remove resources.. Afterwards, she /he navigates to the "Manage DAC roles" section from the DAC-Management-Portal panel (See Figure 16) where she / he can see the different roles assigned to the DAC users (DAC -> "DEMONSTRATOR"). In this case, the super administrator decides to remove the DAC-admin role to the "demo-user-3" and give her / him a DAC-member role instead.

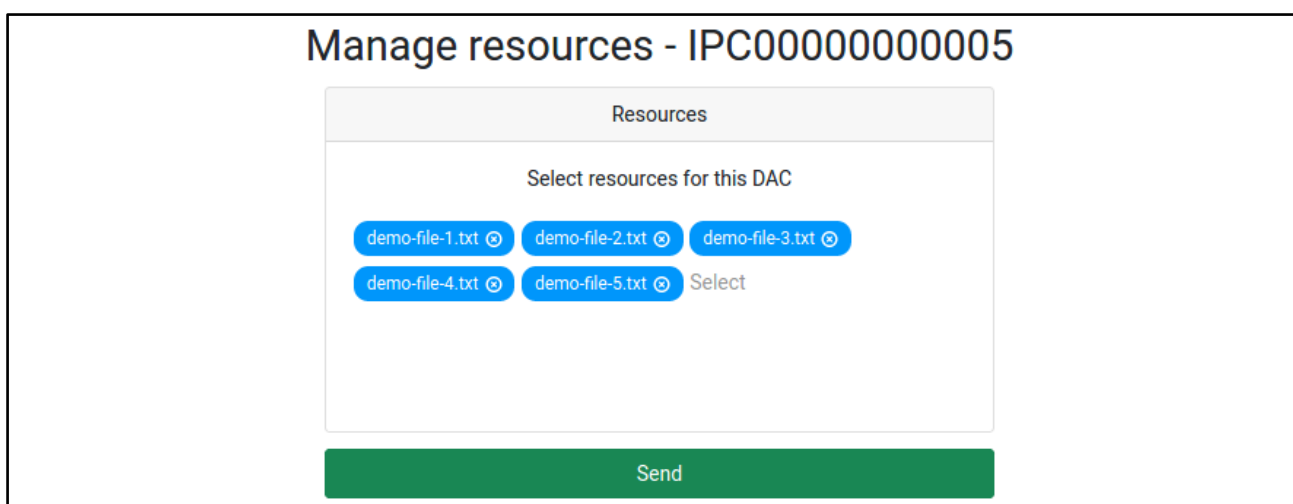


Figure 15: The "Manage resources" section in the DAC-Management-Portal panel.

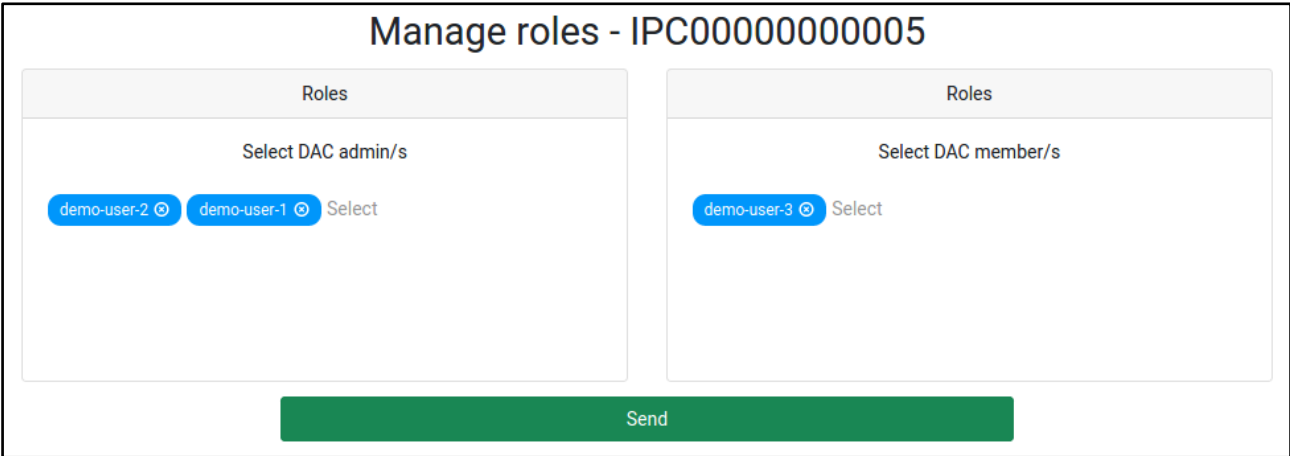


Figure 16: The “Manage DAC roles” section from the DAC-Management-Portal panel where the super administrator changes the role of the “demo-user-3”.

Finally, the DAC-member checks her / his dashboard, where she / he can manage requests made to her / his DAC. Specifically, as illustrated in Figure 17, the DAC-member user accesses the personal dashboard in the iPC DAC-Portal, where she / he can inspect incoming data access requests made to the DACs that she / he belongs to, and then accept/reject (grant / deny permissions) to them. Other functions such as submit DAC-information (“CREATE DAC”, see Figure 8), policies management (“MY POLICIES”, see Figure 9), and permissions management (“MANAGE PERMISSIONS”, see Figure 14), are exclusive of the users granted with a DAC-admin role.

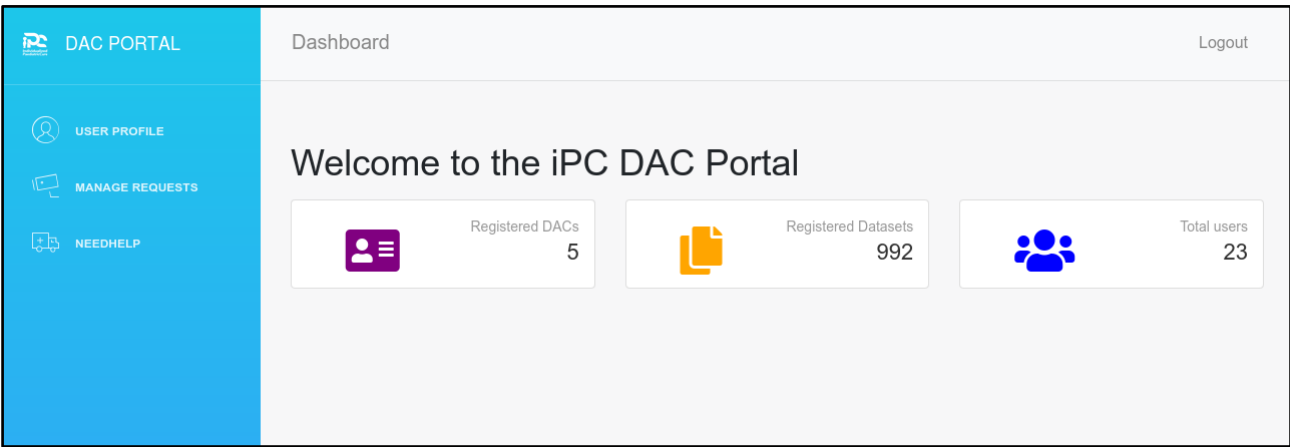


Figure 17: The personal dashboard of a DAC-member user in the iPC DAC-Portal.

Chapter 5 Summary of Achievements

This deliverable has described efforts made on the implementation of the iPC's Data Access Framework. The list of requirements for building such a system have been already described (see Chapter 2.) and the list of achievements are presented below:

- **Compatibility and integration of the Data Access Framework:** The Data Access Framework components have been described on this deliverable (Chapter 3. *Components*). These components are organised as microservices, and therefore, they are a good fit with the overall platform architecture which follows the same design principles. For the development of such a framework, the Data Access Playground¹⁶ was built, which consists on a dockerized sandbox that encapsulates independent git repositories (one per component) into a single repository, where a coordinated deployment is orchestrated (docker-compose) both on development and production environments. As components are isolated, and they are built under the microservices paradigm, they can also be developed and deployed independently. In other words, the components have their own application lifecycle. In fact, some of these components have been already added into the main project's repository^{17,18}, and also deployed into production.
- **Data Access Requests and Data Access Committees management:** A mechanism for performing data access requests from the iPC's Data Catalogue has been implemented, and also, the functionality for managing incoming requests and setting up specific data policies by the data controllers has been developed (See Chapters 3. *Components* and Chapter 4. *Use Cases*).
- **Data Access Committee management:** The functionality for creating new DACs has also been established (Chapter 3. *Components* and Chapter 4. *Use Cases*).
- **System usability:** All the operations can be completely done in an interactive and user-friendly way (See Chapter 4 *Use Cases*).

¹⁶ <https://github.com/inab/data-access-playground.git>

¹⁷ <https://github.com/inab/iPC-Platform-Deployment.git>

¹⁸ <https://github.com/acavalls/arranger.git>

Chapter 6 Future Work

Considering the list of requirements for setting up the iPC's Data Access Framework and the progress made so far, we can devise the following activities around its future development, as further described below:

- Development (improvements of the current prototype)
- Deployment in production
- Integration to external systems

6.1 Development Tasks

We plan several development tasks:

- Implementation of the “Invitations” feature in the iPC's DAC-Portal, that will allow users granted with a DAC-admin role to invite other members of the iPC's project to join their DAC/s. Nevertheless, super administrators are always able to modify memberships/roles for specific DACs.
- Improvements in the DAC-Notify module, by adding more appealing email formatting, better email contents, and connecting all the applications of the framework to the notifications module (e.g. DAC-Portal).
- Improvements in the general layouts for some of the components.
- Increase the testing coverage for the different applications. Implement CI/CD (continuous integration, continuous delivery/deployment) pipelines for automated testing, delivery, and deployment into production environments.

6.2 Deployment in Production

Some of the components of the Data Access Framework are still not deployed in production, and therefore, not available to the end-users. As it was shown in the previous chapters, a working prototype of the Data Access Framework has been developed and it is expected to be available to the iPC users after certain requirements are met. On one hand, the development of new features must continue (see Chapter 6.1) in order to complete the minimal set of features needed for a better inner working of the framework. On the other hand, at the operational level, the platform is embracing the concept of agility, where every component is tied to specific CI/CD workflows (e.g. iPC Data Catalogue¹⁹, Permissions-API²⁰, iPC Catalogue Outbox-API²¹) for their automated testing, delivery and deployment to the production environment. In order to make that possible, testing pipelines of the different components of the framework have to be properly designed (e.g., unit/integration testing (strategy), coverage (reach), etc.) for a secure and automatic deployment into production. Since these aspects are not considered exclusive of the Data Access Framework but methodologies of the general platform instead, they will be presented and extensively explained in the D2.5 *Global Report on iPC Computational Infrastructure* (at M53).

¹⁹ https://github.com/inab/iPC_Data_Portal

²⁰ <https://github.com/inab/Permissions-API>

²¹ <https://github.com/inab/iPC-Catalogue-Outbox-API-v2.git>

6.3 Integration to External Systems

There are two integrations that might be explored in the context of the iPC project that are strongly related with the iPC's Data Access Framework. On one hand, an integration with the EGA's Data Access Framework, that will require adapting the AuthN/Z server based on Keycloak to become a Passport broker, according to the GA4GH specification. On the other hand, an integration with the Cavatica platform, where the final mechanism is still yet to be decided.

6.3.1 Integration of the EGA Data Access Framework

The EGA Data Access Framework that has been built under the same principles as the iPC Data Access Framework is a good candidate to be integrated in the iPC Computational Platform. There are many reasons that make such an integration an important step forward for the iPC project, which are listed below:

1. EGA provides a permanent storage of files/datasets, which is very convenient for the data providers as their submitted data will last long even after the project is finished.
2. EGA provides mechanisms for a secure data transfer from their repositories to the final data consumer (e.g. Virtual Research Environment, Cavatica, etc). An example would be the Crypt4GH²² tool.
3. EGA AuthN/Z is fundamentally based on OIDC, and therefore, users can be linked from one system to another, and user's visas can be retrieved from their system and used in the context of the iPC Computational Platform for data access control.
4. EGA is both technically and legally prepared for hosting more sensitive data (e.g. pseudonymized data). The iPC Data Access Framework storage solution (Nextcloud) should only be used for storing anonymized data.

The advantages of integrating the EGA Data Access Framework into the iPC's Computational Platform are pretty obvious. However, this sort of integration comes with a series of technical challenges that must be addressed, some of which are discussed below.

BSC Keycloak as a Passport broker

In the current implementation, the iPC Data Catalogue retrieves the different Visas of authenticated users directly from the Permissions-API. Depending on these permissions, users can interactively load metadata from their selections in the catalogue to the Virtual Research Environment (VRE) via Catalogue Outbox-API²³, or instead, start a data access request. In the case of a Data Access Framework formed by different repositories and multiple Visa issuers (EGA & BSC Permissions-API), the main AuthN/Z OIDC compliant server (Keycloak) must act as a Passport broker, that is able to fetch Visas from different Passport Visa Issuers, and generate a Passport that links all the Passport Visas (Figure 18, step 1) to a specific identity (user) in the form of a JWT. Once the user is authenticated, Keycloak will provide this Passport to the different clients such as the iPC Data Catalogue or the Virtual Research Environment (Figure 18, steps 2a, 2b). After a dataset is selected in the iPC Data Catalogue (See Figure 10) the user will be either able to expose datasets for their analysis, or instead, start a data access request to the proper Data Access Framework (EGA DAC-Portal, BSC DAC-Portal) (Figure 18, steps 3a., 4a.) if the user does not have access to the selected dataset/s. In the former, data consumers (e.g., VRE) will fetch related metadata from the Outbox-API that is used to materialise files in the analysis platform (Figure 18, steps 3b., 4b). In the latter, if the data access request is approved by the specific DAC (DAC-Portal: Passport Visa Assertion

²² <https://crypt4gh.readthedocs.io/en/latest/>

²³ <https://github.com/inab/iPC-Catalogue-Outbox-API-v2.git>

Source), permissions will be granted and stored in the Permissions-API (Figure 18, step 5a), that will be shared in the form of Passport Visas (Permissions-API: Passport Visa Issuer) upon request (Keycloak: Passport broker). In both cases (EGA, BSC), Data Access Committees will be generated after data submissions are made in the specific repositories (Figure 18, steps 0a, 0b).

Note that the architecture presented in Figure 18 already includes the components related to the iPC Data Access Framework (DAC-Portal, DAC-Management-Portal, Permissions-API, and Keycloak AuthN/Z acting as a Passport broker). The EGA's Data Access Framework components (e.g., DAC-Portal, Permissions-API, ...) and the iPC's DAC-Notify module have been omitted for clarity. In yellow the representation of the missing interactions, in purple components deployed in the development environment, and in light blue the components already deployed in production.

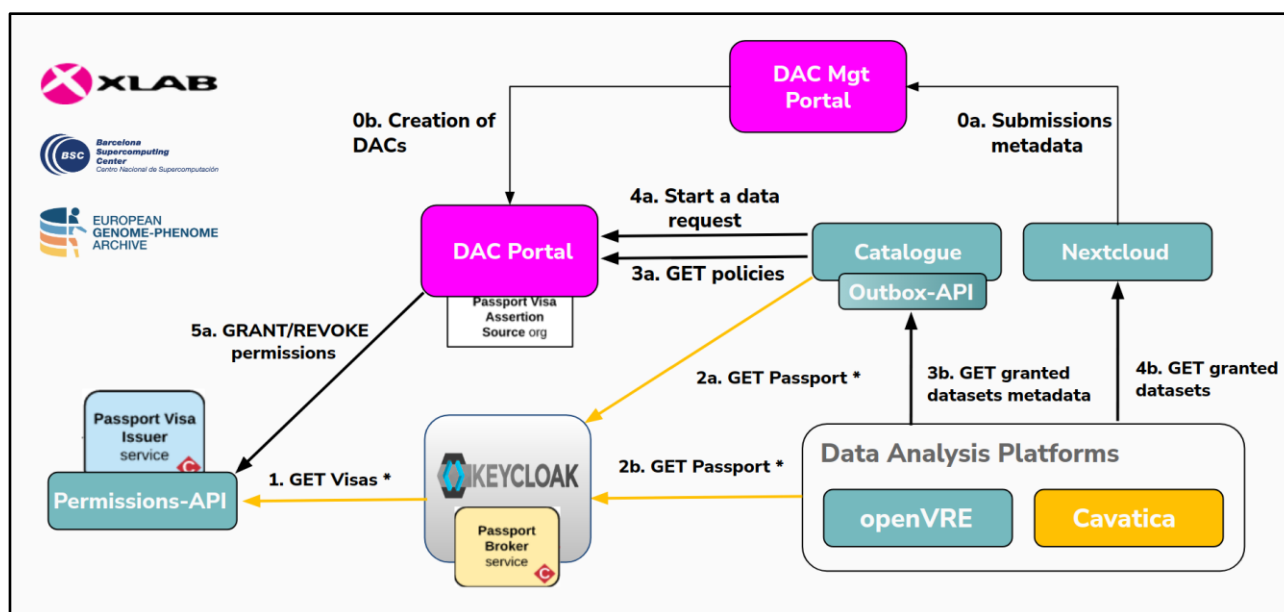


Figure 18: iPC Computational Platform proposed architecture.

Data sharing from external repositories into the Virtual Research Environment

Once permissions flow from one system to another in the form of Visas, data access control can be assured. However, files have to be shared in a secure way from the data repositories to the analysis platforms as well. In the case of the iPC Data Access Framework, this mechanism has been already described and implemented (Deliverable 2.1), where the Virtual Research Environment materialises files hosted on Nextcloud via WebDAV with a privileged account. In the case of bringing data from the EGA data repositories, the data sharing mechanism will be different as the technologies involved are also different. A good candidate for a secure data sharing between systems could be the Crypt4GH²⁴ tool provided by the EGA itself. However, discussions must be held and a final decision on this regard will be made accordingly.

²⁴ <https://crypt4gh.readthedocs.io/en/latest/>

6.3.2 Integration of Cavatica

Cavatica²⁵ is a cloud storage and analysis platform designed for the analysis of paediatric tumour data, which is integrated to the Kids-First Data Resource Portal²⁶ and produced in collaboration with Seven Bridges²⁷. Currently, this platform supports a login system with external identity providers (eRA commons). The Researcher Auth Service (RAS) promoted by the NIH promises to reduce the burden on researchers for accessing protected datasets by enabling a federated login system where different identity providers are allowed (OIDC) and that is aligned with the GA4GH standards. To our knowledge, Cavatica is planning an integration with RAS, and therefore, it would be worth, at least, to explore a potential integration with the iPC Computational Platform once the Keycloak server operates as a Passport broker.

²⁵ <https://docs.cavatica.org/docs/new-for-cavatica>

²⁶ <https://kidsfirstdrc.org/portal/>

²⁷ <https://docs.sevenbridges.com/>