

## **DELIVERABLE 3.2**

### **LAYERED SECURITY FRAMEWORK**

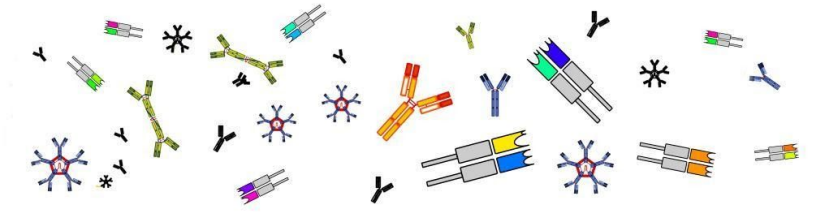
**WORK PACKAGE NUMBER: WP3**

**WORK PACKAGE TITLE: LAYERED DATA SECURITY**

**TYPE: PROTOTYPE**



This project is funded by the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 825821 and Canadian Institutes of Health Research (CIHR)

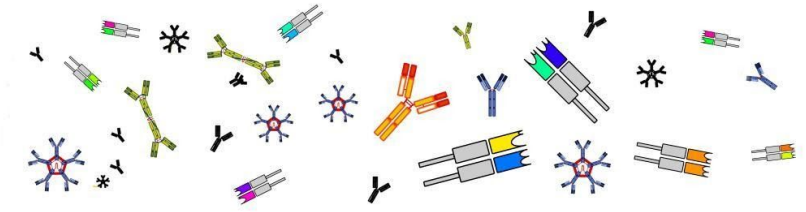


Document Information

<b>iReceptor Plus Project Information</b>	
<b>Project full title</b>	Architecture and Tools for the Query of Antibody and T-cell Receptor Sequencing Data Repositories for Enabling Improved Personalized Medicine and Immunotherapy
<b>Project acronym</b>	iReceptor Plus
<b>Grant agreement number</b>	825821
<b>Project coordinator</b>	Prof. Gur Yaari
<b>Project start date and duration</b>	1 <sup>st</sup> January, 2019, 48 months
<b>Project website</b>	<a href="http://www.ireceptor-plus.com">http://www.ireceptor-plus.com</a>

<b>Deliverable Information</b>	
<b>Work package number</b>	WP3
<b>Work package title</b>	Layered Data Security
<b>Deliverable number</b>	D3.2
<b>Deliverable title</b>	Layered Security Framework
<b>Description</b>	Establish a secure data sharing infrastructure that follows the security and privacy guidelines defined for the iReceptor Plus project, providing the FAIR principles of scientific data management and stewardship when dealing with protected AIRR-seq data.
<b>Lead beneficiary</b>	INESC TEC
<b>Lead Author(s)</b>	INESC TEC
<b>Contributor(s)</b>	INESC TEC, Ascora, SFU, BIU, Sorbonne, UTSW, Haifa, APHP, Mitmynid



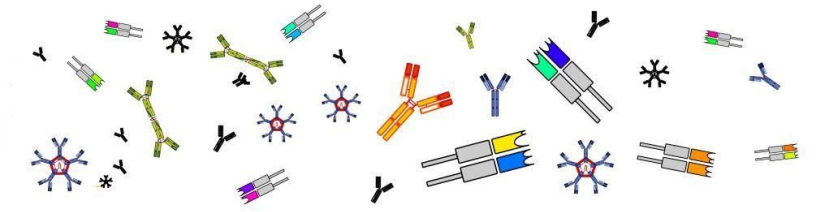


<b>Status (Final (F), Draft (D), Revised Draft (RV))</b>	D
<b>Dissemination level (Public (PU), Restricted to other program participants (PP), Restricted to a group specified by the consortium (RE), Confidential for consortium members only (CO))</b>	PU

<b>Document History</b>			
<b>Revision</b>	<b>Date</b>	<b>Modification</b>	<b>Author</b>
1	2019-12-11	Consolidated version circulated internally	Alexandre Costa, Artur Rocha, Ademar Aguiar
2	2019-12-12 to 2019-12-19	Document revision	Tobias Hinz, Brian Corrie, Felix Breden, Alexandre Costa, Artur Rocha, Ademar Aguiar
3	2019-12-23 to 2019-12-26	Document revision	Gur Yaari, Alexandre Costa

<b>Approvals</b>				
	<b>Name</b>	<b>Organisation</b>	<b>Date</b>	<b>Signature (initials)</b>
<b>Coordinator</b>	Gur Yaari	Bar Ilan University	2019-12-23	GY
<b>WP Leaders</b>	Artur Rocha	INESC TEC	2019-12-19	AR

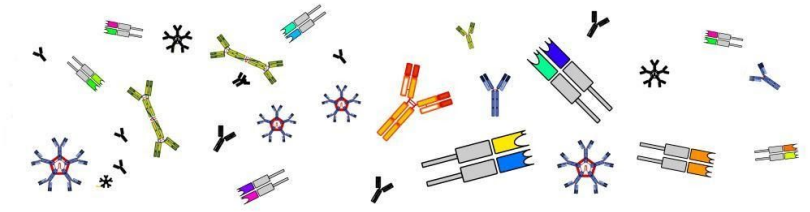




## Table of Contents

Executive Summary	6
Deliverable description	7
Introduction	8
Authentication: Identification and Identity Providers	8
Authorization	8
OAuth 2.0	9
User-Managed Access (UMA)	10
Exploratory Data Analysis	12
Goals and Security Requirements	12
Minimal authorization approach	12
Task Execution	13
Keycloak	13
Implementation	13
Identity Providers and Brokering	14
EGI Check-in	14
Integration with Keycloak	14
Prototype	18
Analysis API	18
Users	19
Resources	19
Frontend Prototype	22
Conclusions	24





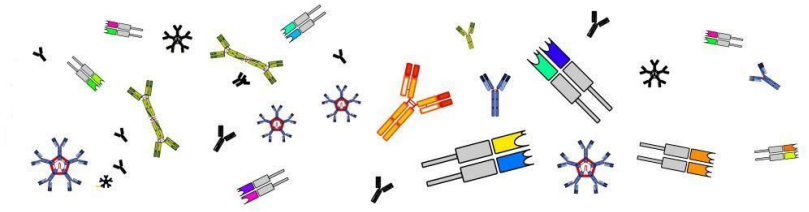
## Executive Summary

The purpose of this Deliverable is to provide a secure data sharing infrastructure that follows the security and privacy aspects defined in Deliverable 3.1 (Holistic Security and Privacy Concept) for iReceptor Plus. While the previous deliverable provided the conceptual framework and policies for this matter, this deliverable provides concepts for the software implementation that follows the approach of layered security, applying them to a prototype iReceptor Repository Service.

When it comes to the domain of health information and biostatistics, the concern for privacy is a subject of continuous discussion. This is vital when it comes to AIRR-seq data sets, which is subject to strict confidentiality and security constraints and this is especially true when these data sets are sampled from human subjects.

To address the main privacy and security objectives, the implemented framework prototype follows the standards and approaches necessary to provide accessibility between the different components of the iReceptor Plus infrastructure, providing multiple levels of authentication, authorization and the mechanisms for data stewards to control different levels of granularity.





## Deliverable description

Deliverable 3.2 is the result of the work done in WP3 Task 3.2 and its subtasks.

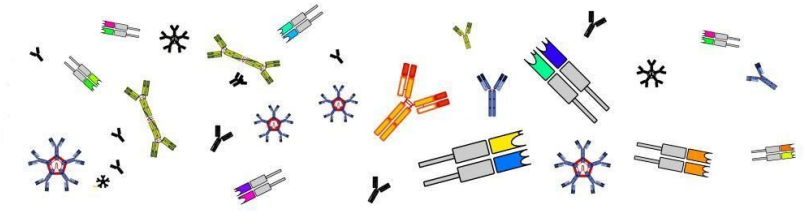
The goal of this deliverable is to provide a layered security framework that addresses the security and privacy aspects that surround AIRR-seq data sharing. While Deliverable 3.1 defined the holistic concepts and requirements for this subject, Deliverable 3.2 implements the actual software framework of layered security between the individual components across the iReceptor Platform.

The current development stage aims at providing a working prototype that demonstrates the capabilities of the proposed secure infrastructure, by providing levels of control at different levels of granularity, restricting access at multiple levels (through authentication), as well as restricting access to specific types of data (through role-based authorization).

Access may be restricted to Exploratory Data Analysis, where the user is limited to requesting summary characteristics of a certain data set, assuming the analysis process runs while protecting confidentiality and privacy by reducing the risk of disclosure of sensitive information.

Fine-grained authorization is also possible and will enable data stewards to control access to their confidential data while at the same time using the iReceptor Platform to enable the secure sharing with researchers who wish to directly access AIRR-seq data sets.





## 1. Introduction

The concern for privacy is a subject of continuous discussion within the health informatics community, especially when it comes to genetic data sets, which are subject to strict confidentiality, security constraints, rights and ownerships. At the same time, analyses done on these data sets may provide crucial research evidence, however, these analyses must be conducted in such a way as not to compromise standards of privacy, regulations and confidentiality for individuals, providers, facilities and data stewards.

*Identification* is the ability to uniquely identify a user of a system or an application that is running in the system. *Authentication* is the ability to prove that an end-user or application is genuinely who that person or what that application claims to be. *Authorization* protects critical resources in a system by limiting access only to authorized users and their applications. It prevents the unauthorized use of a resource or the use of a resource in an unauthorized manner.

### Authentication: Identification and Identity Providers

An Identity Provider (IdP) is defined as a system that can provide identity to an end-user through a set of login credentials and ensures the entity is who or what it says it is across the iReceptor Plus components. This makes identity consistent across services and repositories and asserts authentication to the user.

The identity provider either directly authenticates the user, such as by validating a username and password, or indirectly authenticates the user, such as by validating an assertion about the user's identity as presented by a separate identity provider.

The type of identity provider depends on the implementation of the service. The main standard used in the implementation of this prototype is OAuth 2.0 along with OpenID Connect, an open standard for token-based authentication and authorization over the different components and services to delegate secure and controlled access to a software client's resources<sup>1</sup>.

### Authorization

The authorization process comes after the end-user is properly identified and describes the method for providing access control to resources. Resource servers need to rely on information to decide if access should be granted to a protected resource. For web based resource servers, that information is usually obtained from a security token, usually sent as a bearer token on every request to the server as seen on

---

<sup>1</sup> <https://tools.ietf.org/html/rfc6749>



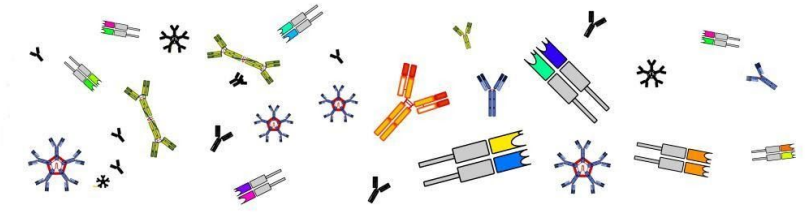


Figure 1. For web applications that rely on a session to authenticate users, that information is usually stored in a user's session, for example through a cookie or local storage, and retrieved from there on each request.

A resource server may perform authorization decisions based on role-based access control (RBAC), where the roles granted to the user trying to access protected resources are checked against the roles mapped to these same resources. But depending on the level of fine-grained authorization required, protections may be further refined through additional mechanisms, such as controlling access on a per-resource basis and by providing authorization policies and policy decision points.

The implemented prototype mainly uses an UMA-compliant (User-Managed Access) workflow for determining access and for processing policies.

## OAuth 2.0

OAuth 2.0 defines four roles that establish the interaction flow to access a protected resource. The terminology involved in this standard refers to its entities in the following way:

- **Resource Owner** is the entity capable of granting access to a protected resource by providing the necessary credentials (username and passphrase, for example). If this entity is a person, it is referred to as an *end-user*. This resource owner could be a researcher trying to access an AIRR-seq dataset, for example;
- **Resource Server** is the server that holds the *protected resource* and is capable of accepting and responding to requests for the protected resources, provided that the *end-user* is in possession of an access token.
- **Client** is the platform or application that can make protected resource requests on behalf of the *resource owner* with specific authorization.
- **Authorization Server** is the server capable of issuing access tokens to a *client* after a successful authentication by the *resource owner* and obtaining authorization.

The OAuth 2.0 standard allows a resource owner to access protected resources through the means of an access token. This token represents a string that denotes a specific scope, lifetime and other access attributes. On a basic authorization code grant flow, this token is normally stored by the client. When a query to a protected endpoint is made this token is traded between the resource server and the client itself to verify its validity.





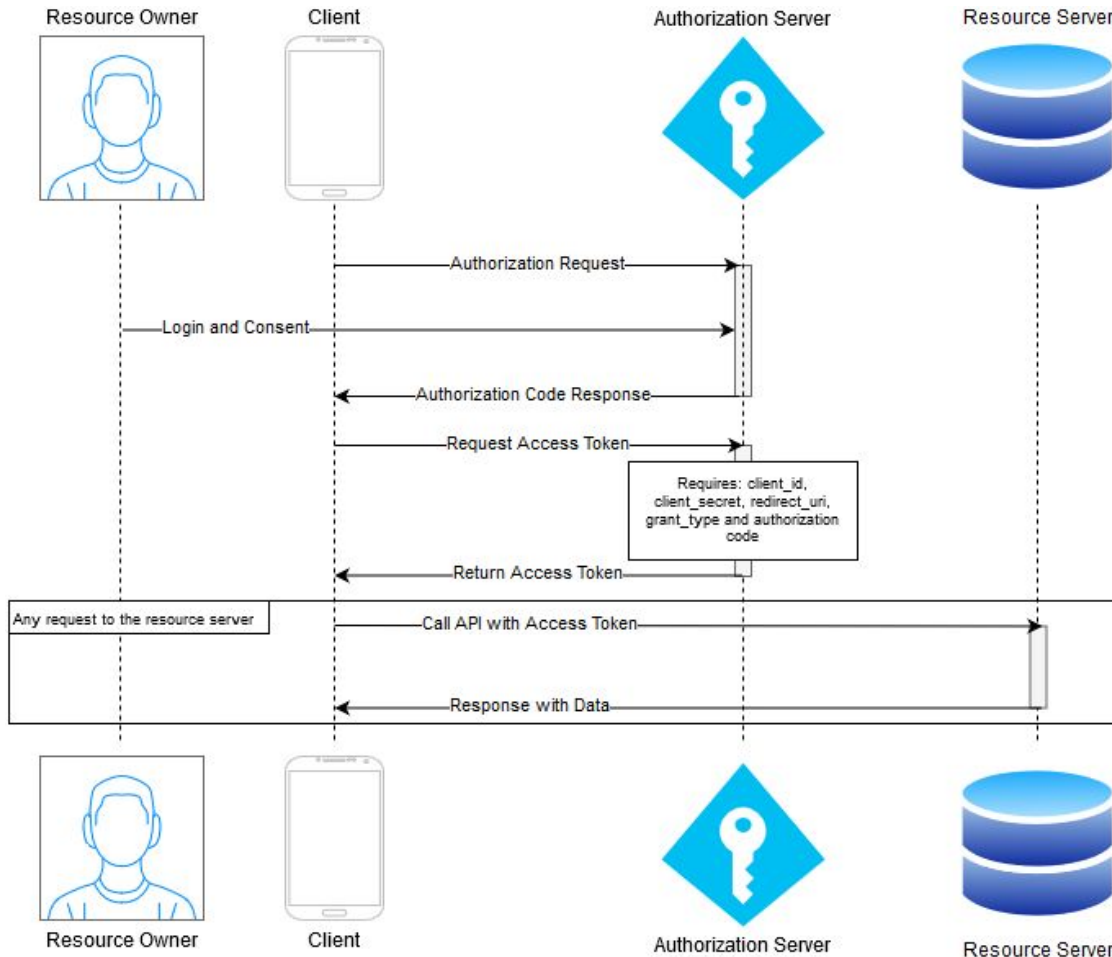
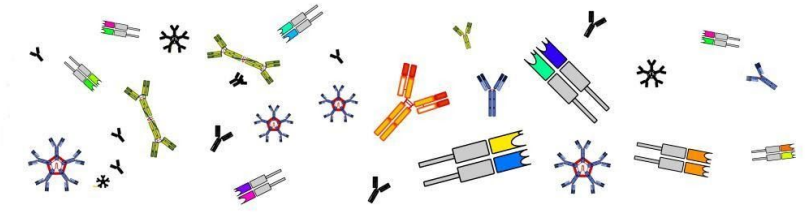


Figure 1: Sequence Diagram for the Authorization Grant Flow

## User-Managed Access (UMA)

User-Managed Access (UMA) is defined as an OAuth-based access management protocol standard<sup>2</sup>. Since the scope of the project requires policy enforced access to resources, access management is a crucial aspect to consider. It is important to properly identify *Data Producers* or *Stewards (Data Managers)* who can manage permissions to their resources and define who can access their resources (*Data Consumers*). UMA is a protocol that enables this kind of management, by giving users responsible for managing data the ability to regulate access. These resources may be anything from AIRR-seq data sets, a study, or even a whole iReceptor Plus node. From an implementation perspective, UMA allows for a centralized authorization server by creating authorization policies no matter where the resources reside, following the distributed nature of the iReceptor Plus repositories.

<sup>2</sup> <https://docs.kantarinitiative.org/uma/wg/rec-oauth-uma-grant-2.0.html>



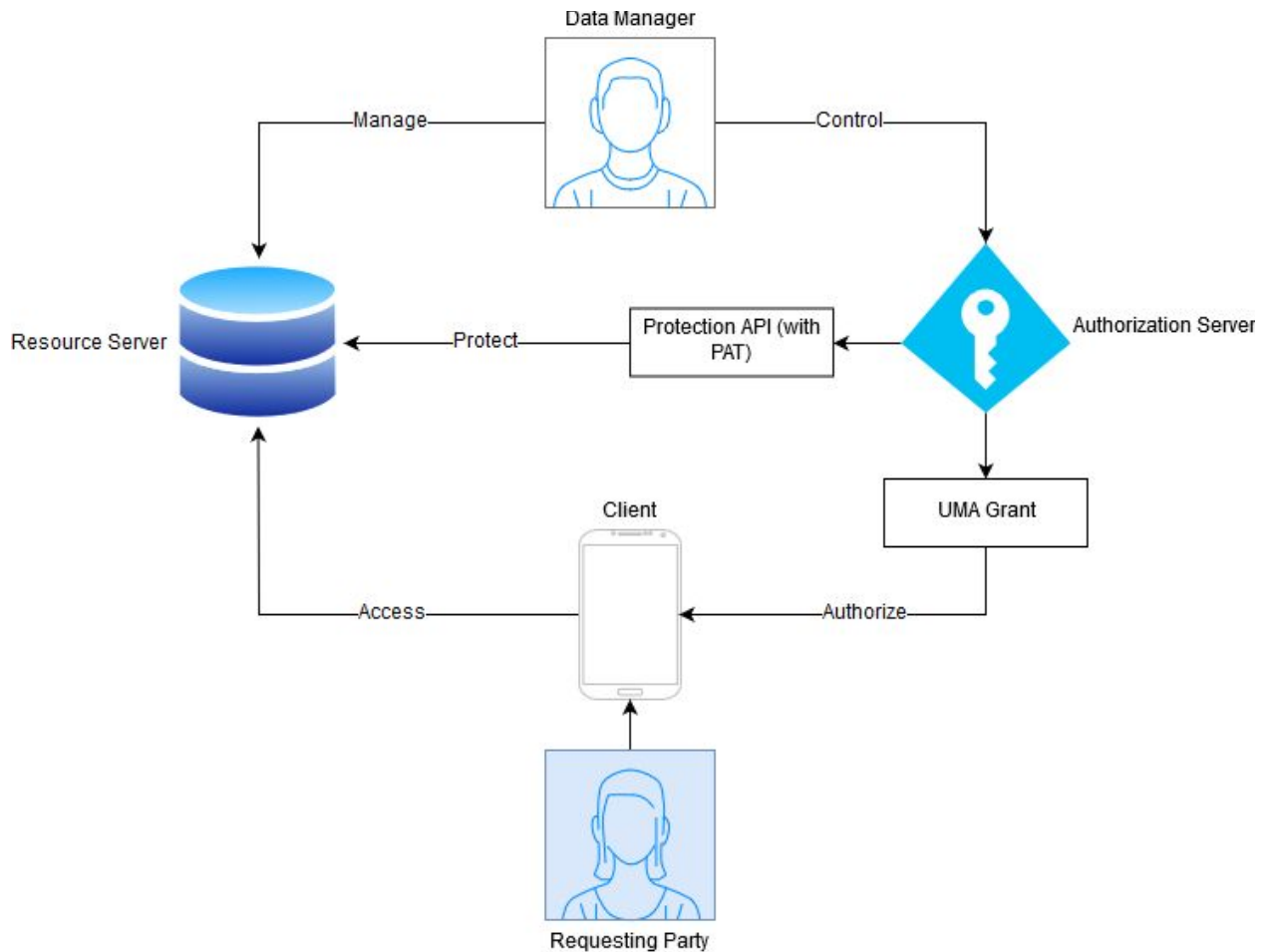
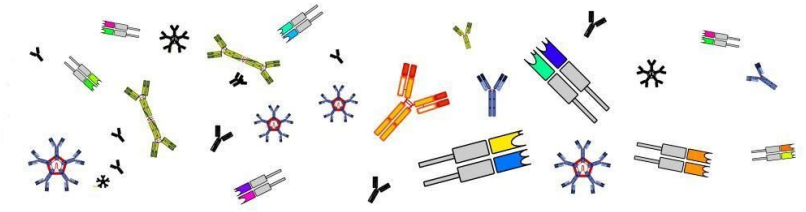
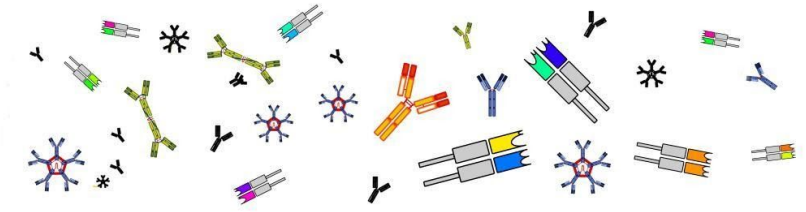


Figure 2: User-Managed Authorization flow

Figure 2 depicts a *Data Consumer (Requesting Party)* accessing a shared resource after proper authorization has been granted. This is possible when the *Data Manager* controls the Authorization Server and allows proper consent to access data sets. Access levels may depend on how fine-grained the authorization is desired to be. For example, a *Data Manager* shares an AIRR-seq data set with read-only permissions and the *Requesting Party* will have no permissions to modify any of its contents. Access to a data set at any time, forbidding the relay party from accessing data any further.

The Protection API is accessible through a PAT (Protection API Token) which is a special type of an OAuth 2.0 that represents the *Data Manager's* authorization to use the *Protection API* and with it they can manage any resources, policies, permissions and access UMA standard endpoints.





## Exploratory Data Analysis

The potential of Exploratory Data Analysis (EDA) and data mining tools to extract accurate and reliable material from confidential data set is an on-going challenge for research and development. Even if the right of direct access to the data has been prohibited, certain results from analyses in the hands of a sophisticated process could be enough to reveal or enable inference on private information. In other words, the process of Exploratory Data Analysis empowers a researcher to get insight into the data set in order to summarize its main characteristics without access to the raw AIRR-seq data sets. With the right types of analysis even with limited access, it may be possible to draw clear conclusions. Keeping this in mind, during the implementation of this prototype, by default, users that can access the platform are able to make EDA requests freely. Of course, restrictions may be expanded upon in the future and as such, this is controllable through fine-grained authorization.

## Goals and Security Requirements

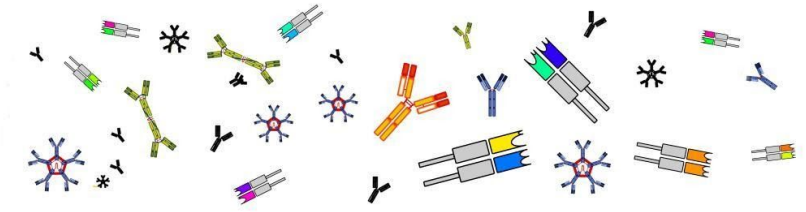
In a service-oriented architecture environment like iReceptor Plus, the platform is composed of services from different sources, such as iReceptor and VDJSerVer. The composition is technology-agnostic, as the applications are not based on similar technologies, languages or platforms. This distributed aspect of iReceptor Plus and the technology-heterogeneity of the integrated applications raise challenges for security. Therefore it is vital to determine an interoperable mechanism for managing resources, independent of the technology and programming languages used for the implementation. The goal of this task and with the implementation of this prototype is to build and demonstrate the feasibility of complementing the iReceptor Plus components with a working Authentication and Authorization Infrastructure (AAI).

### Minimal authorization approach

For the implementation of this security prototype a proposal for a minimal approach for private nodes was considered:

- If a *Data Manager* authorizes a researcher (*Data Consumer*) to access their node, then the researcher will be able to access all the statistics endpoints from all studies.
- A node manager can grant users the ability to access/download raw data on a study-level basis.





## 2. Task Execution

### Keycloak

The challenge for providing identity brokering, data access management and authorization will be met with Keycloak. It is an open source, free, identity and access management platform, providing multiple standards and protocols for client and end-user authentication and authorization. When it comes to identity brokering, it supports integration with external IdP services and has built-in support for user federation through LDAP and Active Directory. The main standard authentication and authorization protocols are OpenID Connect, OAuth 2.0 and SAML 2.0 and User-Managed Access (UMA) for authorization and the protection API. Keycloak is a UMA 2.0 compliant authorization server that provides most UMA capabilities and thus is possible to implement policy enforced access to resources. *Data Managers* can manage permissions to their resources and decide who can access a resource. Thus, Keycloak can also be used as a sharing management service from which ownership and access can be managed.

When it comes to implementing Keycloak support into the iReceptor Plus services, it offers support *adapters* made available in multiple languages such as Java, JavaScript and Node.JS. This means it can easily be integrated in any web-based application. If no adapters are available for a specific library or language, Keycloak is interoperable and implementation can follow the standard workflow and endpoints used for authentication and authorization.

### Implementation

The Keycloak service implemented for this prototype is available at:

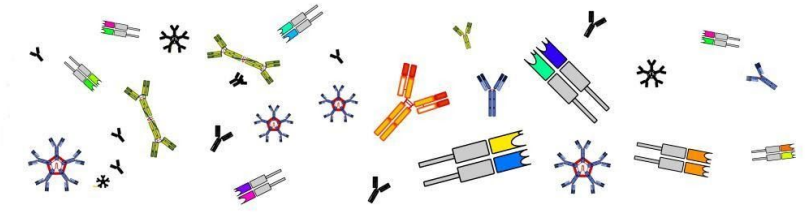
<https://ireceptorplus.inesctec.pt/auth> [last access: 2019-12-10]

Keycloak's OAuth 2.0 standard endpoints (OpenID Connect more specifically) are available through a standardized endpoint called *well-known*. This endpoint is used by OAuth Clients to determine the location of the various endpoints, for example, for obtaining *Access Tokens*, for obtaining *UserInfo*, for determining *Grant Types*:

<https://ireceptorplus.inesctec.pt/auth/realms/iReceptorPlus/.well-known/openid-configuration> [last access: 2019-12-10]

A starting realm named "iReceptorPlus" was created for the purpose of this prototype. The concept of a realm is modularity. It secures and manages security metadata for a set of users, applications, and registered OAuth clients. Users can be created within a specific realm within the Administration console.





Roles (permission types) can be defined at the realm level and admins can also set up user role mappings to assign these permissions to specific users.

## Identity Providers and Brokering

With the use of Keycloak it is possible to integrate external identity providers. An identity provider is usually based on a specific protocol that is used to communicate authentication and authorization information to their end-users. It can be a social provider such as ORCID or Github, it can be an internal service such as Tapis<sup>3</sup> or it can be a business level service like EGI Check-in or EOSC's ELIXIR<sup>4</sup>.

When using Keycloak as an identity broker, users are not forced to provide their credentials in order to authenticate in a specific realm. Instead, they are presented with a list of identity providers from which they can authenticate when they land on the login page.

## EGI Check-in

EGI is a federated e-Infrastructure dedicated to providing advanced computing services for research and innovation and is coordinated by the EGI Foundation. EGI Check-in, also known as EGI AAI, is a proxy service that operates as a central hub to connect Identity Providers with EGI's service providers and resources using federated authentication mechanisms. Through this service, users can authenticate with the credentials provided by the IdP of their home organisation (for example, through eduGAIN), as well as using social identity providers (for example, ORCID). To achieve this, the EGI AAI has built-in support for SAML, OpenID Connect and OAuth2. This means it can be directly integrated into iReceptor Plus' Keycloak instance.

## Integration with Keycloak

EGI Check-in documents the procedures for integration through a *wiki* platform available through their own service located at: [https://wiki.egi.eu/wiki/AAI\\_guide\\_for\\_SPs](https://wiki.egi.eu/wiki/AAI_guide_for_SPs) [last access: 2019-12-10]. These procedures were followed for the development of the prototype platform.

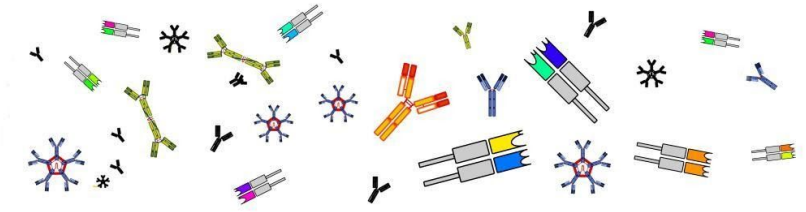
As mentioned previously, both EGI Check-in and Keycloak may communicate through the interoperability of the OAuth 2.0 standard (more specifically OpenID Connect). Following this workflow, it is first necessary to register Keycloak as a *Client* on EGI's side. EGI Check-in provides a development environment of their AAI that does not require formal registration (i.e. does not require a possible lengthy administration approval), available at: <https://aai-dev.egi.eu/oidc> [last access: 2019-12-10]. Figure 3 shows the Client configured for Keycloak, including the *redirect\_uri* that points to where

---

<sup>3</sup> <https://tacc-cloud.readthedocs.io/projects/agave/en/latest/>

<sup>4</sup> <https://elixir-europe.org/>





iReceptor Plus' Keycloak is located. While registering this client, it is necessary to provide *well-known* configurations such as the *authorization\_endpoints* and the *token\_endpoint*.

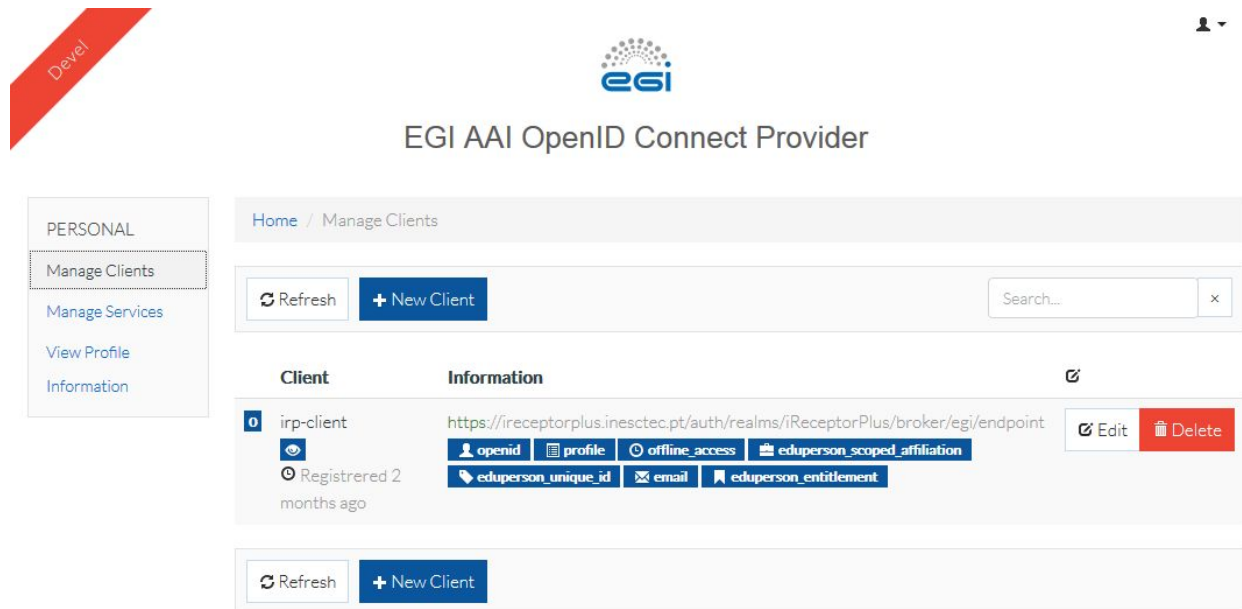


Figure 3: EGI Check-in All - OpenID Connect Client management, showing an entry created with keycloak's settings configured

When configured successfully this Client will be given a *client\_id* and a *client\_secret* that must be specified in Keycloak's configurations. Figure 4 shows how to initialize the registration of a new Identity Broker on Keycloak's side. The configuration that needs to be provided is like the previous step, being required to provide the *well-known* configurations, but this time for EGI Check-in: <https://aai-dev.egi.eu/oidc/.well-known/openid-configuration> [last access: 2019-12-10].



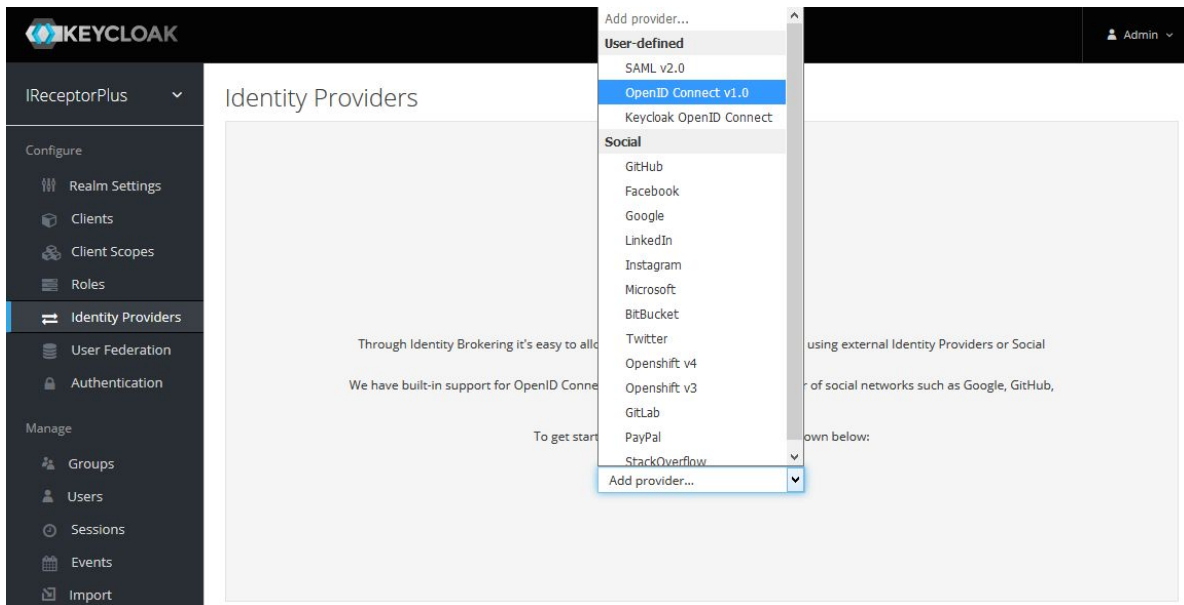
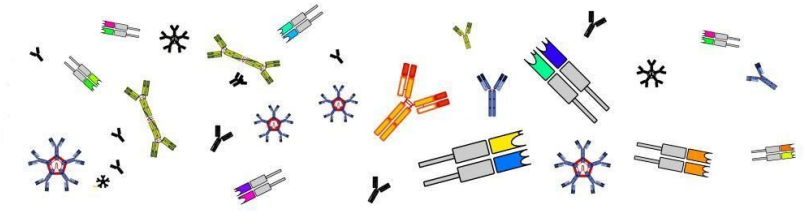


Figure 4: Keycloak Administrative Console - Configuring an external Identity Broker

On a successful configuration, Keycloak will automatically link the IdP's account with its own database. Any Client protected by Keycloak will now land on the Authentication page shown in Figure 5. Clicking EGI Check-in will take the user to Check-in's Authentication page as shown in Figure 6.

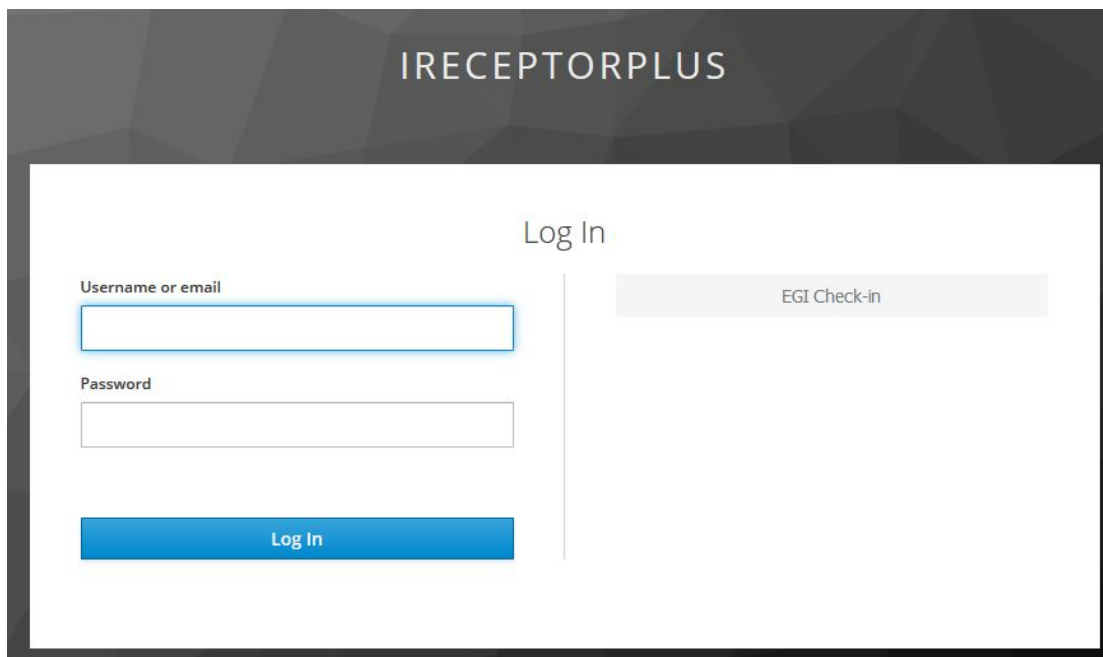
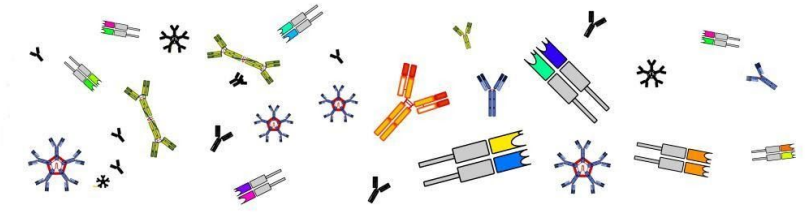


Figure 5: Keycloak's Client Authentication page - Configured with EGI Check-in as an Identity Broker





Check-in

Choose your academic/social account

Search...

- A. T. Still University
- AAF Virtual Home
- AAI@EduHr Single Sign-On Service
- Aalborg University
- Aalborg University
- Aalto University
- Aarhus School of Marine and Technical Engineering
- Aarhus School of Marine and Technical Engineering
- Aarhus University
- Aarhus University

or















				
				
				

Figure 6: EGI Check-in's Client Authentication page

## Prototype

### Analysis API

The Analysis API is the service responsible for providing analysis results on AIRR-seq data sets. The development of this prototype followed the usage of a mock-up Analysis API using the proposed implementation specifications available at:

<https://github.com/ireceptor-plus/WP4/blob/cdd3c0067d83308d9040838b206f2ccae4e7e826/specs/analysis-api.yaml> [last access: 2019-12-10]

The mock-up Analysis API works as a backend web service and will feed the frontend side to display and interact with its contents. In addition to the proposed specification. This version includes some additional endpoints to list repositories and repertoires.

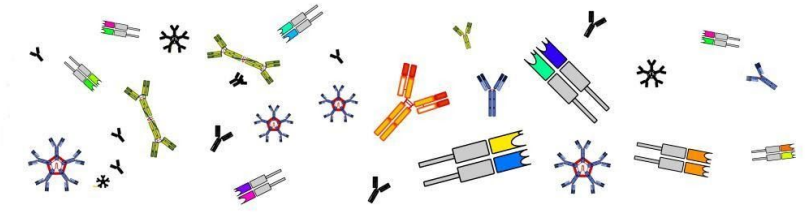
The implemented mockup Analysis API is online at:

<https://ireceptorplus.inesctec.pt/loop-airr/irplus/v1/analysis/explorer/> [last access: 2019-12-10]



This project is funded by the European Union's H2020 Research and Innovation Programme under Grant Agreement No. 825821 and Canadian Institutes of Health Research (CIHR)





## Users


To demonstrate the inner workings of the prototype we will be referring to two example users:




- *siramik vase* – *Data Manager* of the studies, may perform any operations, including access raw sequences directly.
- *emma nate* – *Data Consumer* and doesn't manage any studies but may freely request Exploratory Data Analysis. Has the "request-analysis" role.





## Resources

In Keycloak (and the UMA standard) *resources* represent the set of entities that will be managed by the authorization service. Protected *resources* may be defined by type (for example, a study, repository, sample), URI (an API endpoint), *owner* (the users responsible for managing permissions), scope and/or permissions.

Figure 8 shows an example of a study managed by *siramik vase* (PRJNA312319), configurable on Keycloak's interface. In addition to the study resource, it is also possible to control accessibility to specific endpoints. In Figure 8 both `/clonotypes` and `/rearrangement` endpoints from the Analysis API have been defined as resources, which will allow us to create custom policies and permissions for them if desired.






Loop-airr 


Settings Credentials Roles Client Scopes  Mappers  Scope  **Authorization** Revocation

Sessions  Offline Access  Clustering Installation  Service Account Roles 

Settings **Resources** Authorization Scopes Policies Permissions Evaluate Export Settings

Filter:

Name		Type		URI		Owner	
Scope							



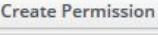
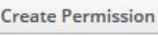
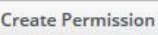
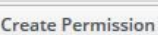
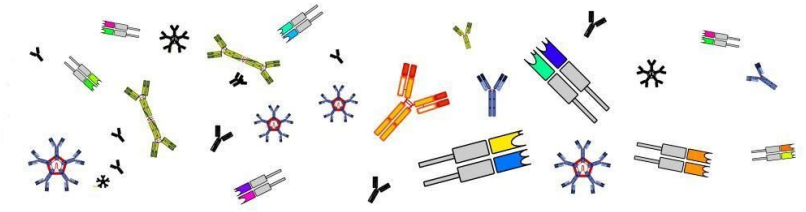
	Name	Type	URIS	Owner	Actions
>	<a href="#">Clonotypes</a>	No type defined.	<code>/clonotypes/*</code>	loop-airr	
>	<a href="#">Default Resource</a>	urn:loop-airr:resources:default	<code>/*</code>	loop-airr	
>	<a href="#">PRJNA312319</a>	No type defined.	urn:study:PRJNA312319	siramik vase	
>	<a href="#">Rearrangements</a>	No type defined.	<code>/rearrangements/*</code>	loop-airr	

Figure 8: Keycloak Authorization Manager, example of an AIRR-seq study ownership and API endpoints controlled as resources









As an example of a more fine-grained authorization, our goal is to allow Exploratory Data Analysis only to users who have the “request-analysis” role and let us assume this user is trying to access the /clonotypes endpoint. The next step is to create a policy that defines that a user must have this role, as seen in Figure 9. Policies define the logical conditions that must be satisfied before granting access to an object.

Clients > loop-airr > Authorization > Policies > Roles > Request Analysis



## Request Analysis



**Name \***  Request Analysis



**Description** 

**Realm Roles \***  Select a role... 

Name	Required	Actions
request-analysis	<input type="checkbox"/>	<button>Remove</button>

**Clients**  Select One..... 

**Client Roles \***  Select a role... 

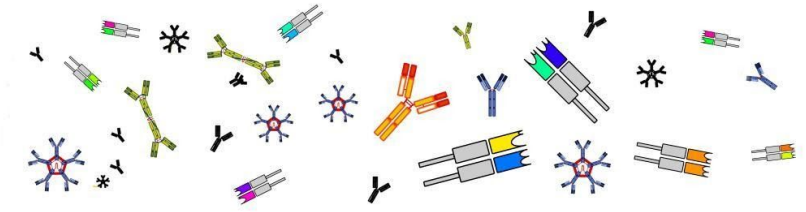
**Logic**  Positive 

Save Cancel

Figure 9: Keycloak Authorization Manager, defining a Policy where the user is required to have the role "request-analysis"

Policies are generally more generic and do not define restrictions for specific resources. That is where permissions come in. Permissions are more specific to the resource that is being protected and associate them to the policy that must be evaluated before deciding whether access should be granted to the user that is making the request. In Figure 10, the Permissions for the /clonotypes endpoint is being defined








where the Policy requirement of having the “request-analysis” role is defined as one of the evaluations that must pass before deciding access.




Clients > loop-airr > Authorization > Permissions > Request Analysis on Clonotypes


## Request Analysis On Clonotypes



Name \*  Request Analysis on Clonotypes

Description 



Apply to Resource Type  OFF

Resources \*  Clonotypes  

Apply Policy 

Select existing policy...  Create 

Name	Description	Actions
Request Analysis		Remove

Decision Strategy  Unanimous 

Save Cancel

Figure 10: Keycloak Authorization Manager, defining a Permission for the /clonotypes endpoint, with the Policy requirement of having the “request-analysis” role

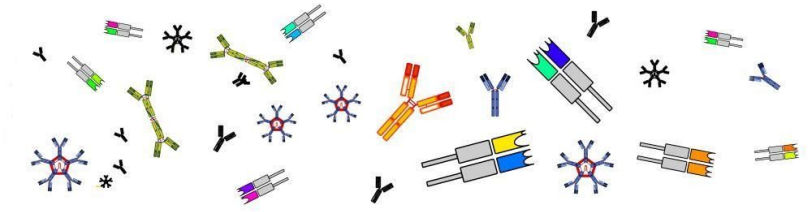
On the backend side of things, the server will rely on using the UMA standard protocol to check if the user has the required permissions for the resource it is trying to access. Assume the user is trying to request an analysis to the /clonotypes endpoint. A request of the following type will be made by the backend to determine if the user has permissions for this operation:

HTTP POST: `http://${host}:${port}/auth/realms/${realm}/protocol/openid-connect/token`

FORM DATA:

```
"grant_type": "urn:ietf:params:oauth:grant-type:uma-ticket",  
"audience": <backend-client-name>,  
"permission": Clonotypes,  
"response_mode": "decision"
```





## Frontend Prototype

The Frontend prototype is a mock-up web interface that serves as a medium to interact with the mock-up Analysis API. The first requirement to be able to access the web interface is to be signed-in. The authentication process follows the flow described in the Keycloak section. The user must be either registered in Keycloak's database directly or may sign-in through EGI Check-in's Identity Brokering service (or any other that may be integrated in the future). If the last option is desired, Keycloak will automatically link the Identity Broker's referring account with its own database, making it possible to tailor permissions and authorization options directly on iReceptor Plus' side.

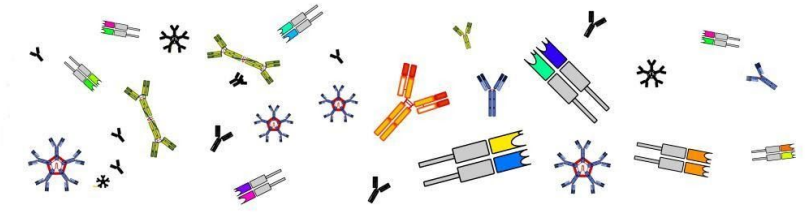
Figure 11 shows the homepage of the frontend prototype. The successfully authenticated user is displayed on the top-right corner and several options are available for navigation.



Figure 11: Frontend Prototype: homepage showing the various menus and links to relevant content

Following the proposal for the prototype, both users can see the available studies, as shown in Figure 12. However, access to Exploratory Data Analysis is limited by the "request-analysis" role and access to raw sequences is limited to *Data Managers* and users who the *manager* has shared data with.





**iReceptor+** Sharing AIRR-seq data

AIRR Studies | Analyses | EMMA NATE

**Studies**

Sample Name	Study ID	Study Title	Organism	Sex	Age	Rearrangement Tool	Actions
— repository: INESCTEC Repository ✕							
SRR3176830	PRJNA312319	Tracking T-cell immune reconstruction after TCR/CD19-depleted hematopoietic cells transplantation in children	Homo sapiens	Female	29 years	MIXCR	
SRR3176829	PRJNA312319	Tracking T-cell immune reconstruction after TCR/CD19-depleted hematopoietic cells transplantation in children	Homo sapiens	Male	3.54 years	MIXCR	
SRR3176828	PRJNA312319	Tracking T-cell immune reconstruction after TCR/CD19-depleted hematopoietic cells transplantation in children	Homo sapiens	Male	4.39 years	MIXCR	
SRR3176827	PRJNA312319	Tracking T-cell immune reconstruction after TCR/CD19-depleted hematopoietic cells transplantation in children	Homo sapiens	Male	3.54 years	MIXCR	
SRR3176826	PRJNA312319	Tracking T-cell immune reconstruction after TCR/CD19-depleted hematopoietic cells transplantation in children	Homo sapiens	Male	8.88 years	MIXCR	
SRR3176825	PRJNA312319	Tracking T-cell immune reconstruction after TCR/CD19-depleted hematopoietic cells transplantation in children	Homo sapiens	Female	3.04 years	MIXCR	

Figure 12: Frontend Prototype - A user lists accessible studies

While logged in as *emma\_nate*, since this is not the *manager* of the study, when trying to access a sample, the button “Request Raw Sequence Access” will be displayed on the screen, as shown in Figure 13.

Study PRJNA312319 / Repertoire SRR3176830

**Metadata**

Study Title: Tracking T-cell immune reconstruction after TCR/CD19-depleted hematopoietic cells transplantation in children

Description: Cancer Study

Institution: Chudakov Lab Shemyakin-Ovchinnikov Institute of Bioorganic Chemistry

Granted By: [Name] Russian Science Foundation

Submitter: [Name]

**Biological Data**

Organism	Sex	Age	Tissue
Homo sapiens	Female	29 years	Blood

**Acquisition Data**

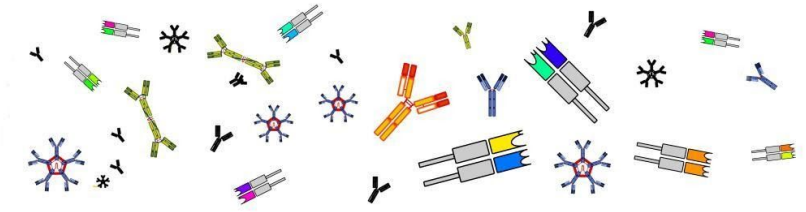
Sequencing Platform: Illumina HiSeq 2500

Processing Details: [Dropdown]

**REQUEST RAW SEQUENCE ACCESS**

Figure 13: Frontend Prototype - User with no management access to a Sample in the study is unable to view raw sequences





Through *siramik vase's* profile, it is possible to share the study with other users. By clicking the username on the top-right corner, the user can select "Account Management" (Figure 14), which will take them to the page where they can manage their resources. Here the *Data Manager* can enter the study and input the username/email of the person they wish to share the study with.



**My Resources**

My resources

Resource	Application	People sharing this resource
<a href="#">PRJNA312319</a>	loop-airr	This resource is not being shared.

Figure 14: Account Management and user managed resources

Figure 15 shows that the resource has been shared with *emma nate*. This access may be revoked at any time. *Emma nate* is now able to successfully access the raw sequences, as seen in Figure 16.

## My Resources > PRJNA312319

### People with access to this resource

User	Permission	Date	Action
emma nate	Any Permission	Dec 9, 2019, 4:00:38 PM	<a href="#">Revoke</a>

### Permissions granting access to this resource

Description	Permission	Action
No permissions granting access to this resource		

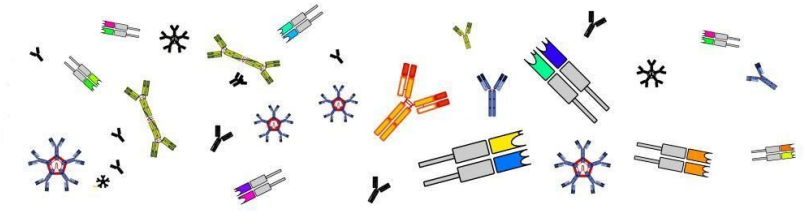
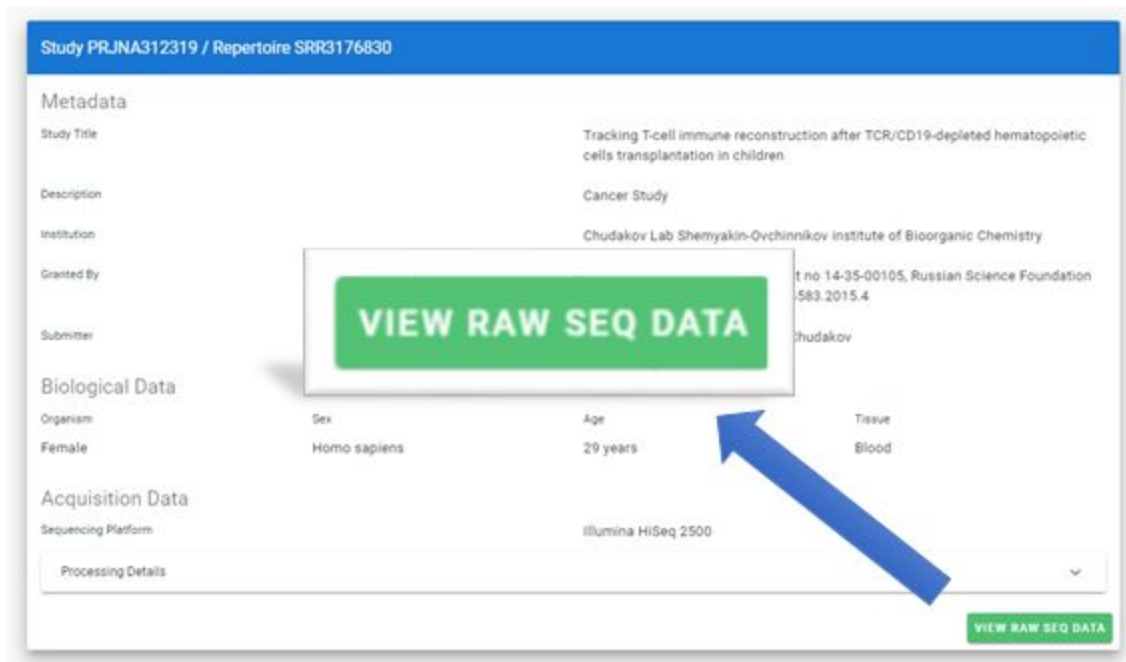
### Share with others

Username or Email \*

[Share](#)

Figure 15: Data Manager shares resource access with a user



Study PRJNA312319 / Repertoire SRR3176830

**Metadata**

Study Title: Tracking T-cell immune reconstruction after TCR/CD19-depleted hematopoietic cells transplantation in children.

Description: Cancer Study

Institution: Chudakov Lab Shemyakin-Ovchinnikov Institute of Bioorganic Chemistry

Granted By: Grant no 14-35-00105, Russian Science Foundation 583.2015.4

Submitter: Chudakov

**Biological Data**

Organism	Sex	Age	Tissue
Female	Homo sapiens	29 years	Blood

**Acquisition Data**

Sequencing Platform: illumina HiSeq 2500

Processing Details

[VIEW RAW SEQ DATA](#)

Figure 16: User received a shared resource and is now able to request raw data from the Sample

## 6. Conclusions and Future Actions

This deliverable is the first version of a prototype, implementing the Layered Security Framework in iReceptor Plus.

The main goal of the implemented software is to provide a technical framework to test and validate the approaches defined in D3.1 for authentication, authorization and auditing, in due respect of defined policies and applicable regulations, while assessing its impact across the global infrastructure.

A vertical prototype (architectural spike) has been developed for the M12 version of this deliverable so that integration issues resulting from the inclusion of security across the different components of the software ecosystem can be tackled from early on.

Further updates of this deliverable will be released in M24 and M36 as new use cases are incorporated, consolidating functional and non-functional requirements of the whole infrastructure and in particular of the implemented layered security mechanisms.

