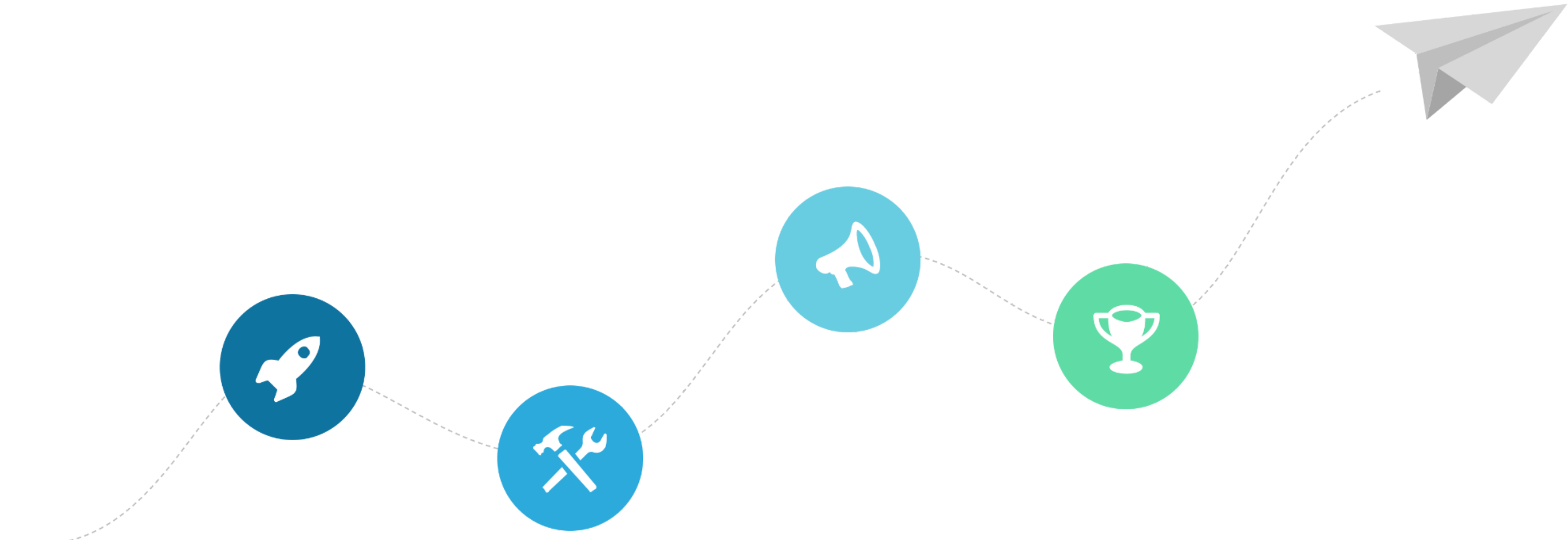
HUMAN GENOMES PLATFORM PROJECT

Data Access Committee Management Systems

CANDIDATE SOLUTIONS EVALUATION REPORT

Dec 2023



Authors

in alphabetical order by surname

Carnuccio, Patrick - AAF

Cowley, Mark - ZERO CCIA

Davies, Kylie - AAF

Green, Cherry - AAF

Hobbs, Matthew - Garvan

Holliday, Jess - BioCommons

Kummerfeld, Sarah - Garvam

Monro, David - NCI

Patterson, Andrew - UMCCR

Pearson, John - QIMRB

Pope, Bernard - BioCommons

Scullen, John - AAF

Shadbolt, Marion - BioCommons

Wong-Erasmus, Marie - ZERO CCIA

Wood, Scott - QIMRB

Acknowledgements

The HGPP formed part of Australian BioCommons’ Human Genome Informatics initiative and was funded by NCRIS via the Australian Research Data Commons (https://doi.org/10.47486/PL032) and Bioplatforms Australia. Contributions were also made by partner organisations: Australian Access Federation, Garvan Institute for Medical Research, National Computational Infrastructure, QIMR Berghofer Medical Research Institute, The University of Melbourne Centre for Cancer Research, the ZERO Childhood Cancer Program and Children’s Cancer Institute.

Table of Contents

Glossary

API	Application programming interface. A way for software applications to communicate with one another.
BAM	Binary Alignment Map. A standard compressed binary file format for describing genomic sequencing read data, usually aligned to a reference genome.
DAC	Data Access Committee. A committee that makes decisions on Data Access Requests based on a Data Access Policy.
DAR	Data Access Request. The suite of forms sent by the Principal Applicant to the DAC to obtain access to datasets.
DUOS	Data Use Oversight System. A software platform for managing the DAC access and request process.
FASTQ	A standard text-based file format for describing sequencing reads and their base quality scores.
HGPP	Human Genomes Platform Project.
OpenID	An open standard for federated identity management which allows users to authenticate for a service via trusted partners via a third-party identity provider[1].
ORCID	Open Researcher and Contributor ID. An alphanumeric code used to uniquely identify contributors to scientific literature.
PI (or CI)	Principal investigator (or Chief Investigator). The lead investigator on a grant/application for data.
REMS	Resource Entitlement Management Software. A software platform for managing the DAC access and request process.
ROR	Research Organization Registry. A database of unique and persistent identifiers for research organisations.
SAML	Security Assertion Markup Language. An open standard for exchanging authentication and authorisation data between parties, such as an identity provider and a service provider[2].
VCF	Variant Call Format. A standard text-based file format for describing genomic variants.

[1] https://en.wikipedia.org/wiki/OpenID

[2] https://en.wikipedia.org/wiki/Security_Assertion_Markup_Language

1. Introduction

The Human Genomes Platform Project (hereafter ‘HGPP’ or ‘the project’) is a collaborative research project aiming to enhance the secure and responsible sharing of human genomic data for research purposes. National and international connectivity is important to maximise the utility of these sensitive and valuable assets. HGPP partners represent many of the largest human genome sequencing and analysis organisations in Australia.

Navigating restrictions on data use is a major challenge in human genomics. Data Access Committees (DACs) spend significant effort assessing the merits of applications and determining whether to grant access to data sets. Historically, DAC activities have consisted of numerous manual steps, making the process slow, burdensome, and largely opaque to the applicant.

The DAC Automation sub-project within the HGPP explored a new data access request and approval paradigm driven by automation to streamline the process for the national human genomics research community. Our aim is to enable the DAC to quickly and easily determine whether access is permitted for the requested purpose when an individual, such as a researcher, applies for access to data.

Beginning with a knowledge discovery phase, the project team mapped the current state for data access requests and data sharing, and documented a set of problems, user stories, and requirements for further exploration. We prioritised requirements for our target solution and grouped the desired requirements into a number of themes. Outcomes and insights from this phase are documented in the preceding Discovery Phase report[3].

This report builds on the discovery phase to examine methods and software solutions for implementing a DAC management system, and evaluates them against our requirements. Our analysis and assessment was brief for patently unsuitable candidates but more credible alternatives were evaluated in detail. While no single system satisfied all our requirements, the Resource Entitlement Management System (REMS)[4] software proved to be the best fit. Despite some shortcomings, the conclusion from our pilot evaluation is that REMS is a workable production solution.

We adopted REMS as our software solution to support a DAC management system pilot in the project. This report details a comparison between REMS and requirements identified in the discovery phase of the project. We conclude with a technical discussion of the pilot solution, outlining the deployment process and other technical aspects examined during this project.

[3] Human Genomes Platform Project: DAC Automation Discovery Phase Report, https://zenodo.org/record/6644050

[4] https://github.com/CSCfi/rems

2. Data Access Committee Management Systems and Software

In this section we provide an overview of the main methods that exist for DAC management, including a brief description of how well they met our requirements.

Ad hoc / email

There are relatively few software systems for managing the operations of a DAC. Many organisations employ ad hoc solutions using group emails and paper forms.

Ad hoc systems are not designed for this kind of task and a key flaw is a lack of activity tracking, leading to reporting and auditing limitations. The inconsistent and unstructured content of email messages makes it impossible to parse for reporting purposes. Auditing is also hard since email does not provide an easy way to track how an application passes through the system, how long it took, who had issues or concerns with the application, and how those were addressed. Without effective tracking, it is difficult to produce meaningful reports. Such processes are opaque, making it difficult for the participant to understand the current status of a data request or to determine who is responsible for the next step.

Ad hoc systems do not satisfy many of the requirements for the Organisational and Person Roles we defined in tables 1a and 1b of our Discovery Phase report[5], such as Holding Organisation, Principal Applicant and DAC Member.

In conclusion, we deemed this default approach unsuitable for our DAC management system pilot implementation because it satisfied very few of our minimum viable product requirements.

Bespoke Form / Ticketing

The DAC process at its core involves completing and submitting application forms. There are various form-building tools that could be used to create bespoke forms to capture the information needed to assess a Data Access Request. Examples include REDCap[6] or Jotform[7]. REDCap is of particular interest because many medical research institutes already use it to collect survey data in research projects.

Similarly, the workflow for evaluating applications via a committee can be implemented using off-the-shelf request tracking systems traditionally used in software development and service desks. Examples include Jira[8], Trello[9] and Request Tracker[10].

[5] Human Genomes Platform Project: DAC Automation Discovery Phase Report, https://zenodo.org/record/6644050
[6] https://www.project-redcap.org
[7] https://www.jotform.com
[8] https://www.atlassian.com/software/jira
[9] https://trello.com
[10] https://en.wikipedia.org/wiki/Request_Tracker

Some organisations have successfully adopted such solutions, either separately or in combination, to conduct DAC processes[11]. Custom form-based workflows offer several advantages over email-based processes, including:

- Forms and templates help guide the applicant to provide the information needed for decision making, including specifying mandatory fields and input validation.
- Ticketing systems can support a variety of workflows of differing levels of complexity.
- Audit trails are centralised in a ticketing system rather than being spread across a variety of email threads.

However, there are downsides to bespoke systems. They do not fulfil many of our identified requirements, primarily because of inherent difficulties with integrating separate bespoke systems into a coherent whole. For instance, it may be hard to generate reports over the whole system if the form workflow and committee workflow are provided by two different components. Furthermore, each organisation is likely to name things differently, use a different data model, and adopt different technologies, making interoperability between organisations virtually impossible. For these reasons, we have not considered designing a bespoke form or ticketing system for our pilot solution in the project.

In conclusion, bespoke forms and ticketing system solutions were deemed unsuitable candidates for our DAC management system pilot implementation because they satisfied too few of our minimum viable product requirements.

Data Use Oversight System (DUOS)

The Data Use Oversight System (DUOS[12]) was created by the Broad Institute[13]. DUOS is a semi-automated data access management service governing compliant secondary use of human genomics data.

One of the attractions of DUOS is its implementation of the Data Use Ontology (DUO[14]), a GA4GH standard. DUO provides a common language to describe data use conditions, expressing appropriate uses and users of data based on the consent given by research participants. DUO was developed to enable machine-readable consent expression, which allows for the computational matching of data requests against data holdings while enforcing data use restrictions. This enables computer-guided access recommendations to help streamline decision-making by the DAC.

DUOS is currently free, designed as a service consisting of a single central instance managed by the Broad Institute. Users and organisations that wish to use DUOS must register for accounts on that central instance.

[11] e.g., Australian Genomics
[12] https://duos.broadinstitute.org/
[13] https://www.broadinstitute.org/
[14] https://www.ga4gh.org/product/data-use-ontology-duo/

The DAC sub-project team were given access to a Broad Institute test instance (see Appendix 1 for screenshots) for a short time during the pilot phase of the project to evaluate the data access request process within DUOS.

Our investigations suggested that DUOS had several favourable attributes, including:

- Source code is available.
- Committee voting is powerful and flexible.
- Automated consent matching algorithm is sophisticated.
- No functionality / stability issues with the features we used.
- Is in production at a major genomics institute (Broad Institute).
- Integrated directly with a major genomics compute platform (Terra[15]).

However, our exploration identified a number of disadvantages with the DUOS model which are detailed below.

Deployment

While the code is open source, deployment is complicated, even for experienced developers and seems to have embedded dependencies on Broad Institute environments. Understanding the nature of the dependencies is difficult and it is not clear if alternatives are possible. Several team members attempted local and/or cloud deployments of DUOS without success. To our knowledge there is no other deployment of DUOS outside of the central instance provided by the Broad Institute.

Uncertain Future Costs

DUOS use is currently free but may move to an annual subscription model in the future. The fee is expected to be tied to a DAC, under the assumption that each organisation will have a single DAC. This model would be challenging for organisations with multiple DACs. Several project partners are custodians of multiple datasets and multiple DACs per organisation are not uncommon.

Data Sovereignty

The Broad Institute DUOS environment is hosted on a commercial cloud platform (currently Google) in the USA, and thus operates in that jurisdiction. Australian laws and regulations, including the Australian Privacy Principles[16], are a more appropriate legal framework for an Australian genomic data platform. Donor consent may even prevent metadata about datasets from being stored outside of Australia. Australian organisations using DUOS would need to carefully control any information placed into DUOS so as to ensure that compliance with Australian laws and regulations is maintained, while also understanding and taking account of the broad data access powers afforded to US government agencies [17].

[15] https://terra.bio/

[16] Privacy Act 1988 (Cth) s 14

[17] https://en.wikipedia.org/wiki/CLOUD_Act

Committee Model

The DUOS model assumes that a single DAC exists for each organisation and handles all datasets for that organisation. This is unlikely to work for:

1. Highly collaborative research networks where external partners are part of the DAC process for particular datasets; and
2. Very large organisations such as Universities and Medical Research Institutes where different organisational sub-units are responsible for datasets they have created or which are under their custody.

Authentication

Because DUOS is a single global instance, authentication is centrally managed using an authentication system provided by the cloud vendor (currently Google). This model may create governance and operational management challenges as end users are unable to leverage existing enterprise identity management infrastructure to access DUOS, unless these are registered/integrated by the Broad Institute into their single global instance. Currently, the central DUOS relies on Google authentication, which may not be compatible with existing enterprise identity management deployments within institutions. Furthermore, subsequent actions within DUOS, including making a Data Access Request (DAR), require the user to provide an account for eRA Commons[18], a US authentication system focused on NIH-centred research grants, to which many Australian researchers do not have access.

Whilst there are Australian solutions allowing universities and medical research institutes to leverage their existing enterprise identity management, such as those provided by the Australian Access Federation (AAF[19]) and CILogon[20], there is no indication that these would be integrated into the single global instance of DUOS.

System improvements

The DUOS code[21] is open source but is heavily reliant on internal Broad Institute services. Project team developers have been unable to deploy a working instance in Australia. Without a local instance, the ability to make localisations and code changes is significantly reduced. The DUOS developers are open to change requests, but it is likely that requesting organisations would be required to fund development activities. Changes would also need to be evaluated against Broad Institute development roadmaps and priorities.

An Application Programming Interface (API) allows third party software to access the internal data and processes of a software system. APIs are frequently used to create software that adds functionality to a system. DUOS does not provide an API which further limits opportunities to add functionality to address requirements that it does not meet.

[18] https://www.era.nih.gov/register-accounts/understanding-era-commons-accounts.htm
[19] https://aaf.edu.au/
[20] https://www.cilogon.org/
[21] https://github.com/DataBiosphere/duos-ui

System upgrades

Because there is a single instance of DUOS, changes and the scheduling of change rollout are out of the control of DUOS users. Plans and roadmaps describing timelines and development priorities for DUOS were unclear to the project team at the time of writing this report.

Data access requests

As the Broad Institute's DUOS instance is multi-tenanted, users can explore all datasets managed by all organisations on the platform. A single data access request may include multiple datasets from the same or different DACs. This necessitates a single data access form and agreement for all datasets. There is no way to customise the data access process for individual organisations or datasets. The data access agreement is written subject to US laws, and data access conditions cannot be modified to cater for different legal jurisdictions or different organisational requirements. It is, however, possible to attach PDF documents to specific datasets, which could detail additional specific conditions on data use, such as compliance with Australian privacy laws. The single-agreement aspect could be overcome by having a separate system outside of DUOS for managing the legal agreements between users requesting data access and Australian organisations holding and managing data. However, moving agreements outside of DUOS adds complexity and reduces the utility of DUOS because it must be used in concert with another system to manage the legal agreements.

Reporting

DUOS has integrated reporting but there is no ability for users to create or modify reports outside of those provided. We did not examine the reporting system in detail but any additions or changes would need to be delivered by the DUOS development team.

In summary, while it was a capable and polished data access management system, DUOS is unlikely to satisfy the diverse operating models of the Australian genomic data community.

Resource Entitlement Management System (REMS)

REMS[22] is a software tool for managing access rights to resources, such as research datasets. It was created and is maintained by the CSC, IT Center for Science[23], a non-profit organisation jointly owned by Finland's government and universities, and administered by the Finnish Ministry for Education[24].

The REMS system is written in Clojure, which is a functional programming language that targets the Java Virtual Machine (JVM) as an execution platform. The open source REMS code[25] is available on GitHub[26] under the MIT License, meaning users can contribute improvements and changes to REMS. Updates can be submitted for review by the REMS developers for inclusion in the main REMS code repository. Alternatively, REMS deployers can maintain a fork of the REMS code where they have their own local

[22] Linden et al. 2013 Resource Entitlement Management System, *Trans-European Research and Education Networking Association (full paper)*

[23] https://www.csc.fi/en/home

[24] https://www.csc.fi/csc

[25] https://github.com/CSCfi/rems

[26] https://github.com/CSCfi/rems

code changes and extensions to the main REMS code. In this way, they can test and deploy the local changes in their local REMS instances on their own timetable, without needing approval from Central REMS developers.

Of all the identified options, REMS most closely matched our requirements for a DAC software system. We identified several production deployments of REMS including:

- The Garvan Institute of Medical Research, which uses REMS as the hub of an automated DAC process (Appendix 2).
- The Genomic Data Infrastructure (GDI) project[27], which employs REMS as a key underlying platform and is likely to continue maintaining and developing the solution.

The consensus by the project team was that REMS was the best choice for our DAC pilot solution. The remaining sections of the document explore REMS features and options for dealing with unmet needs in more detail.

[27] GDI Starter kit User Journey incorporating REMS for DAC management
https://gdi.onemilliongenomes.eu/gdi-starter-kit.html

3. REMS Pilot Solution

The project team piloted REMS to more deeply understand the capabilities and limitations of the system with respect to our requirements. Requirements had been developed in the preceding pilot phase of the project[28] and categorised with a priority rating using a MANDATORY / SHOULD HAVE / NICE TO HAVE scale, inspired by the MoSCoW prioritisation method[29]:

- MANDATORY (**M**ust have)
  - No point in delivering a solution without these requirements
- SHOULD HAVE (**S**hould have)
  - Important but not vital
- NICE TO HAVE (**C**ould have)
  - Wanted or desirable
- Unused (**W**on't have this time)
  - We did not bring these types of requirements through from the prior project

Once we had completed our review of software, we acknowledged that no system met all of our mandatory requirements. Our preferred solution, REMS, satisfied all but two of our mandatory requirements (Table 1).

Match against requirements

We deployed a test instance of REMS to understand and evaluate its capabilities and limitations. We conducted a number of exercises where different team members assumed different DAC roles and we went through the various processes in the operation of a DAC. We reviewed each requirement in the Requirements Traceability Matrix (Appendix 3) and determined whether it passed or failed in the context of our test environment. This process confirmed our initial impression that REMS fulfils many, but not all, of the requirements for a DAC system.

Table 1. The percentage of high priority requirements (MANDATORY or SHOULD HAVE) that were scored as a "pass" by REMS.

Requirement type	Total requirements	Total Passed	% Passed
Mandatory	13	11	85
Should have	16	06	37

Many high-priority requirements were met by REMS. In particular, REMS has excellent support for enabling a process of collaboration between applicants and DAC members regarding their applications for data. This collaborative space is secured through standard authentication and authorisation mechanisms. It is customisable at a dataset ("REMS resource") level, allowing custom forms, custom

[28] Human Genomes Platform Project: DAC Automation Discovery Phase Report, https://zenodo.org/record/6644050
[29] https://en.wikipedia.org/wiki/MoSCoW_method

committee processes, and extensive communication of status between all parties involved in the application process.

The user interface of REMS is clear and provides a useful dashboard for all participants allowing them to quickly see/action the status of all applications they are involved in. However, there are four broad groups of requirements that were not met by REMS.

- Reporting requirements. The ability of DAC members and users to extract summary information from the system for the purposes of their bespoke reporting. The REMS developers have acknowledged that this is a weakness of the current implementation and would support future developments with some clearer use cases and user stories. Using APIs or third-party reporting tools (see Section 4 for details of some of our technical exploration in this area) provides a partial workaround to the reporting limitations.
- The usage of REMS as a central location for DAC record keeping over the complete life span of a data application. The current implementation of REMS allows for a variety of attachments and comments to be made during the application process but does not allow any activity once an application has been approved. We have identified a set of requirements involving attachment of documentation (e.g., signed agreements, data sharing closure reports) after the application approval (indeed possibly years after the application approval). This set of requirements feels achievable within the current framework/design of REMS as all of the technical requirements already exist in the system (document attachment and commenting).
- The use of researcher identities such as ORCID. Various “should have” requirements centred on the ability for DAC members and applicants to be able to bring in other public information sources via researcher identifiers. This information could be used both to aid in filling in application forms and to help verify the bona fides of researchers.
- The final group of “should have” requirements would require a shift in the fundamental data model of the current REMS implementation and so are unlikely to be implemented in REMS. REMS has a very broad definition of a “resource”: the thing that is being shared. It can encompass many different types of datasets or artefacts. In human genomics, we have identified requirements that involve the nuanced control/release of items within the “resource”. For instance, a rare disease dataset (the REMS “resource”) may contain 1,000 patients. However, individual consent for those 1,000 patients may not be homogenous, and some individuals should be treated differently by the DAC depending on the nature of the proposed research activity. Since REMS has no concept of the individual items that might make up a larger “resource”, it is hard to envisage how these kinds of requirements can be satisfied. After further consideration by the project team, we determined that these requirements would be left to future community members to resolve by adding software that uses the REMS API to deliver the requirement. Alternatively, the requirement can be reclassified to NICE TO HAVE.

Where REMS does not fulfil a high priority requirement, plans were developed to address or work around that failure (Table 2). Where we could not identify a workaround, extending REMS directly or using the API to extend REMS via software add-ons are still options.

Table 2. REMS “failed” requirements and proposed plans to address or work around them to fulfil our needs.

Requirement Number & Short Description	**Priority**	**Plans to Address Gaps**
HGPPREQ-026 DAC administrators can record and track approved applications so as to efficiently report on what datasets have been shared and to whom.	Mandatory	This data is available but difficult to extract. Individual stakeholders can extract data and use external report tools if this requirement is mandatory for them. This requirement is unlikely to be needed immediately after adopting REMS but will become more important as the volume of applications increases.
HGPPREQ-039 Executed data sharing agreements/contracts must be able to be stored in the system and linked to the original data access request.	Mandatory	Attachments can only be made as a response to a field on a form. REMS locks approved applications and it is not possible to make any changes to the form once approved. The project team has submitted a feature request to the REMS team to introduce this feature.
HGPPREQ-011 Reporting information - who has been granted access to what dataset.	Should have	The information is available, but requires individual queries, there is no single report. Feedback was provided to the REMS development team, but the project stakeholders need to specify detailed reporting requirements to enable REMS to build it.
HGPPREQ-017 DAC members have access to an electronic archive of their historic approvals dating back for a reasonable period (such as 5 years) after the research project ends, so as to cross reference new applications with previous approvals made.	Should have	There is a work around. Data can be extracted via the API. However, this feature is of limited value until a history of applications has accumulated. To be addressed in a later phase.
HGPPREQ-020 The system should have the facility to capture an open researcher ID and project/activity ID (e.g. ORCID and ROR) and a placeholder for future RAID.	Should have	There are persistent identifiers for users (ORCID), research activities (RAiD), organisations (ROR) and data sets, but these are not fully supported within REMS. To be addressed in a later phase. Will likely require us to specify the requirements and for the REMS team to implement changes to support persistent identifiers within the application.
HGPPREQ-021 DAC members can know the verified identity and institution/workgroup details of the applicant so as to avoid the need for double checking with institutions and to be assured that they are only granting access to appropriate applicants.	*Should have*	Partial pass in the sense that primary identity does indicate home organisations and the affiliation the person has with the home organisation. There is no infrastructure available to perform real time checks to see if a person is employed. Downgraded to a SHOULD HAVE and to be addressed in a future iteration.

HGPPREQ-025 The system should allow users to add notes or log information into an approved application for the life of the project so future DAC members can derive insights as to how the data was used, what the outputs were, how the compliance was done etc.	Should have	REMS locks the DAR after approval. Future action to recommend REMS introduce this feature. While the core application can remain locked, adding supplementary notes and attachments (see HGPPREQ-039 in Appendix 3) would be extremely useful.
HGPPREQ-032 DAC administrators can record and follow-up on any periodical/publication/other reporting requirements via the platform. (e.g. Ability to track whether an annual report is due and has been done for data collection). Report attached to the DAC Sharing Agreement/Application would be nice to have.	Should have	REMS locks the DAR after approval. Future action to recommend REMS introduce this feature. See also HGPPREQ-039 and HGPPREQ-025 in Appendix 3.
HGPPREQ-047 The DAC members/administrators should be able to restrict the permissions of an applicant to a subset of the complete cohort population (for example DUO codes to be applied at the record level not dataset level).	Should have	REMS does not have this capability. This requirement cannot be fulfilled without major changes to the data model. Possible downgrade to NICE TO HAVE and to be addressed via API or third party solutions.
HGPPREQ-048 The DAC members/administrators should be able to restrict access of an application to a subset of data types in the cohort (BAMs only, VCFs only, not FASTQs etc).	Should have	
HGPPREQ-050 The DAC members/administrators should be able to enable access to the data cohort respecting the individual consent preferences of all the participants in the cohort (dynamic consent).	Should have	
HGPPREQ-111 The ability to partially populate a new application from other data sources such as ORCID and/or RAID, for example so you don't have to rekey your "life science profile" over multiple applications.	Should have	

4. Technical aspects of REMS

As part of the candidate solution evaluation of REMS, a number of the project partners deployed REMS instances for technical assessment. While not a formal assessment process, we have documented our impressions and findings in this section.

Deployment

Basic

REMS is deployable out of the box as a standard Java application requiring a PostgreSQL database backend, making deployment straightforward[30]. The REMS repository includes example configuration files that can be used as a starting point. This includes details of your connection to the database and the OIDC configuration which define the users who can log in.

Docker

It is possible to deploy as a simple Docker stack including the database, though the documentation for the Docker approach is targeted at developers and seems mainly designed for use in development. However, a perfectly usable production deployment could also be achieved using Docker[31].

Production AWS

A production-grade deployment of REMS can utilise native cloud container technologies as this provides access to an optimal mix of security, performance, cost, and ease of maintenance. In the case of AWS, we choose to use AWS Fargate. A REMS instance can be deployed as a container running in the Fargate managed compute service, using a managed Postgres instance in AWS. Under this architecture, there is no underlying machine or operating system to manage; the deployer only has to manage the version of REMS that is deployed. A detailed guide and scripts have been developed for REMS deployment using containers and AWS Fargate[32].

Alternatives

Google Cloud Platform (GCP) and Microsoft Azure have similar options for deploying containers in the cloud, including managed container infrastructure and managed PostgreSQL instances.

Application Programming Interface (API)

REMS has an extensive API that covers almost all operations available via the user interface. The API is central to REMS, and in fact, the user interface is implemented on top of the API. Figure 1 depicts the REMS architecture.

[30] https://github.com/CSCfi/rems/blob/master/docs/installing-upgrading.md#installing-rems

[31] https://github.com/CSCfi/rems/blob/master/docs/installing-upgrading.md#running-rems-with-docker

[32] https://github.com/AustralianBioCommons/rems-aws-deploy

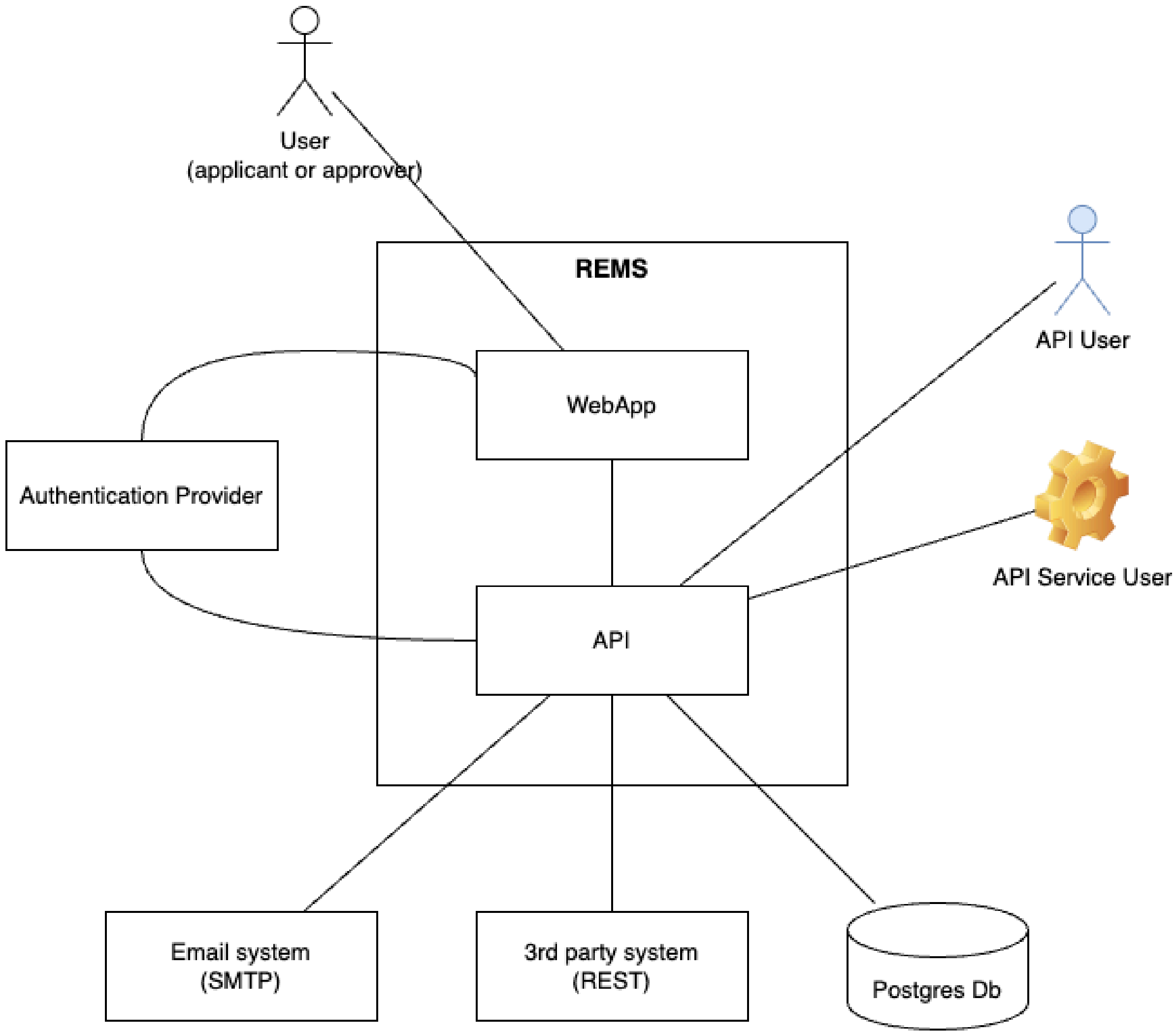
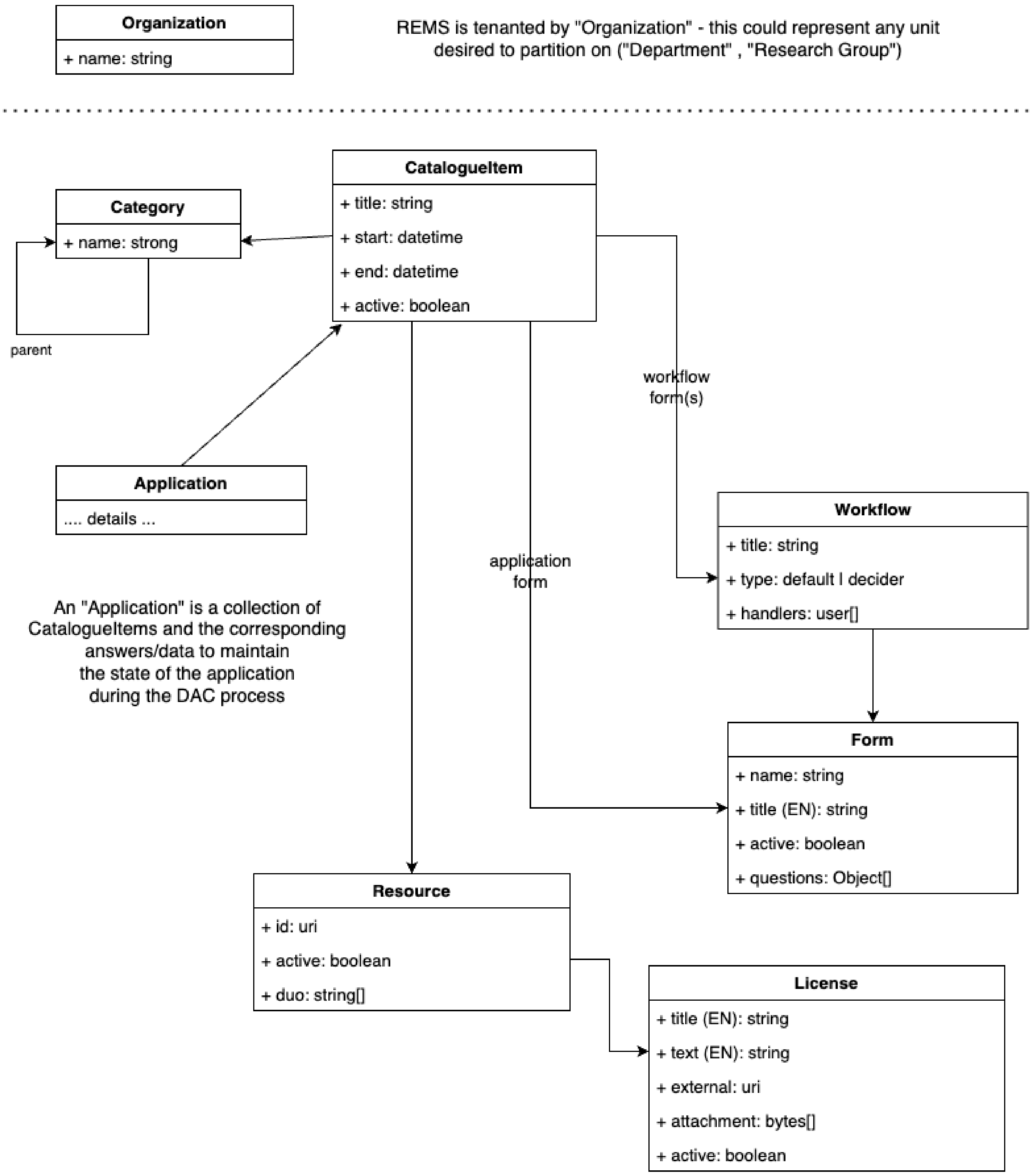


Figure 1. The central role of the API in REMS architecture (adapted from https://github.com/CSCfi/rems/blob/master/docs/architecture.md).

The APIs[33] main entities are “catalogue items” (a concrete instantiation of a resource), “forms” (the definition of the form to make an application), and “resources” (the items available for sharing). These are joined together in an “application” (a single application for resources, through the specification of catalogue items and their corresponding form answers etc). Figure 2 shows a simplified diagram of the main REMS entities and their relationship to each other.

[33] https://rems-demo.rahtiapp.fi/swagger-ui/index.html



REMS is tenanted by "Organization" - this could represent any unit desired to partition on ("Department" , "Research Group")

An "Application" is a collection of CatalogueItems and the corresponding answers/data to maintain the state of the application during the DAC process

Figure 2. Entities in the REMS system.

All API operations require a user and key, or the user must have already logged into REMS and have an active session. Most API work from an external system will require the creation of some form of service user and a corresponding key[34].

APIs can be called using any modern HTTP library or tool. The API can extend REMS functionality and conceptually can be divided into API calls which either push or pull data. In addition, REMS functionality can be expanded through the use of plugins and of hooks. Some example API scripts[35] may be useful.

Pull API calls

The API can be used to pull data from REMS, the obvious use case being generation of a custom REMS activity log. REMS offers basic reporting functionality (see below), but by using the API, more comprehensive information about any of the REMS entities (e.g. users, organisations, applications) or processes (e.g. invitations, workflows), is available.

Push API calls

An obvious use of a push API call is to send notifications upon approval of an application. For example, the REMS API could be used to contact a mail server and send an email notification to relevant people (in addition to the applicant and those involved in the REMS workflow) that an application has been approved.

Another scenario is an "entitlement push", in which REMS interacts with a third party system to automate access to a dataset for an approved applicant. REMS includes an implementation using this pattern. An optional application processing feature[36] allows the access rights of approved applications to be uploaded to the European Genome-phenome Archive (EGA).

Plugins and hooks

REMS is extensible through a plugin framework[37] that uses the API. Plugins can be used to transform, process or validate data[38]. REMS is also extensible through external scripts[39], although these are categorically different from plugins and are not tied directly to API endpoints.

Reporting

REMS currently has limited native reporting functionality available directly through the application user interface. The project team considered reporting via three possible avenues:

1. Extending the user interface.
2. Using the existing REMS API to perform reporting functions (see pull APIs above).

[34] https://github.com/CSCfi/rems/blob/master/docs/using-the-api.md#api-key-authentication
[35] https://github.com/AustralianBioCommons/rems-scripts
[36] https://github.com/CSCfi/rems/blob/master/manual/handler.md
[37] https://github.com/CSCfi/rems/blob/master/docs/plugins.md
[38] https://github.com/CSCfi/rems/tree/master/resources/addons/bona-fide-pusher
[39] https://github.com/CSCfi/rems/blob/master/docs/hooks.md

3. Using the existing REMS CSV export.

Extending the User Interface

The REMS user interface could be extended to add visualisation tools (e.g., charts and table generators) and slicing/dicing functionality (e.g., filters and sorting). However, this user interface would require significant effort from either the REMS development team or the user community. Any included functionality would be limited compared with specialist reporting and visualisation tools (e.g., Power BI and Tableau). Consequently, the project team did not pursue this option.

API and CSV Exports

The existing REMS API and CSV export can provide almost complete access to all elements stored in the REMS database. This data could be fetched nightly and moved into a data warehouse for reporting. Given the low volume of DAC applications, this seems the most viable route.

The CSV export API call is the most obvious technique for batch querying and data can be readily imported into other tools. This is the most promising area for further exploration.

4. Conclusions

The aims of the DAC Automation sub-project were to investigate Data Access Management Services appropriate for human genomics data from a technical, policy and operational requirements point of view, and to evaluate appropriate solutions at participating repositories to semi-automate DAC activities.

During the discovery phase of the project[40], we established the formal requirements of the system through community consultation, identified several candidate solutions, and assessed those solutions against the criteria. Through this process, REMS was identified as the leading candidate and was selected for a pilot implementation.

REMS is lightweight, open source, has flexible deployment models, and is already in production for DAC management at one of the partner organisations (Garvan Institute). Open source code and a comprehensive API minimises vendor lock-in, and we can extend REMS either via the main codebase or via external software that accesses the API.

While REMS fulfilled many requirements, there were some gaps. Most of the missing requirements can be met with further development to REMS or by using the extensive REMS API to process data in third-party tools. Most unmet requirements relate to more sophisticated use cases.

There were some requirements outside the capabilities of REMS, which could be addressed by:

1. Requesting the REMS developers to build these features.
2. Using project team developers to extend REMS and submitting that to the REMS developers for inclusion in the code base.
3. Using the API to create add-on features to deliver the capability outside of REMS.

REMS locks the forms once an application is completed, so no changes can be made once the form is submitted. This preserves the state of applications exactly as submitted and fits closely with the REMS worldview that sees application approval as that natural end state of its workflows. This resulted in a number of failed requirements relating to tracking and following up on applications. The ability to recall and edit supplementary information for a submitted application is a feature we will explore further with the REMS development team.

The following items will be recommended to the REMS development team for future development:

- Address the issue of recalling and editing submitted applications so supplementary information can be added to a finalised application.
- Improve the granularity of data access in order to respect individual consent and data types.

The project team found only a handful of systems designed to streamline the process of managing data access requests for research. It would be possible to construct a bespoke system using form-creation

[40] https://zenodo.org/record/6644050

tools or even to bend a related tool like REDCap for this purpose, but this will result in inconsistent approaches by different data custodians.

In summary, the DAC Automation sub-project achieved its main objective in confirming the technical viability of one or more solutions. However, significant work is required before launching a production-ready service. A short list of work packages that would be useful to address in future projects include:

- Creating policies to address operational issues such as privacy, acceptable use, and standardised data sharing agreements.
- Building and deploying infrastructure for a production-grade REMS service.
- Creating solutions to extend usage reporting directly into REMS or via API integration.

Appendices

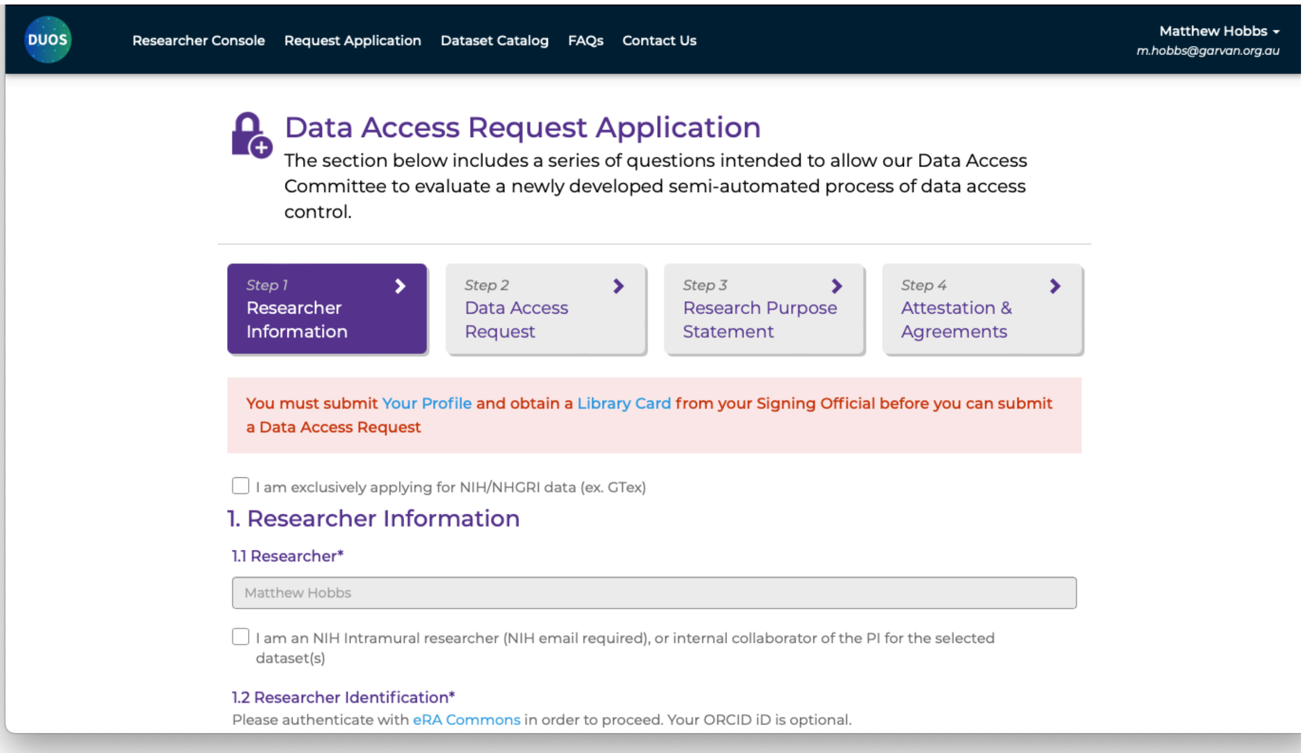
Appendix 1 - DUOS user interface

Figure A1 shows the front page of the DUOS user interface.

The page's top menu provides links to the main parts of DUOS:

- A Researcher Console where researchers can track their data requests;
- A Request Application which guides researchers through the process of requesting access to one or more data resources; and
- A Data Catalog which lists all resources that are managed through this instance of DUOS.

The page content outlines the four stages of the Request Application workflow. The first stage (shown here) begins by requesting information about the applicant. Note references to procedures or entities that are not of direct relevance to Australian researchers.


DUOS
Researcher Console
Request Application
Dataset Catalog
FAQs
Contact Us
Matthew Hobbs
m.hobbs@garvan.org.au
Step 1 Researcher Information
Step 2 Data Access Request
Step 3 Research Purpose Statement
Step 4 Attestation & Agreements

Data Access Request Application

The section below includes a series of questions intended to allow our Data Access Committee to evaluate a newly developed semi-automated process of data access control.

You must submit Your Profile and obtain a Library Card from your Signing Official before you can submit a Data Access Request

□ I am exclusively applying for NIH/NHGRI data (ex. GTex)

1. Researcher Information

1.1 Researcher*

Matthew Hobbs

□ I am an NIH Intramural researcher (NIH email required), or internal collaborator of the PI for the selected dataset(s)

1.2 Researcher Identification*

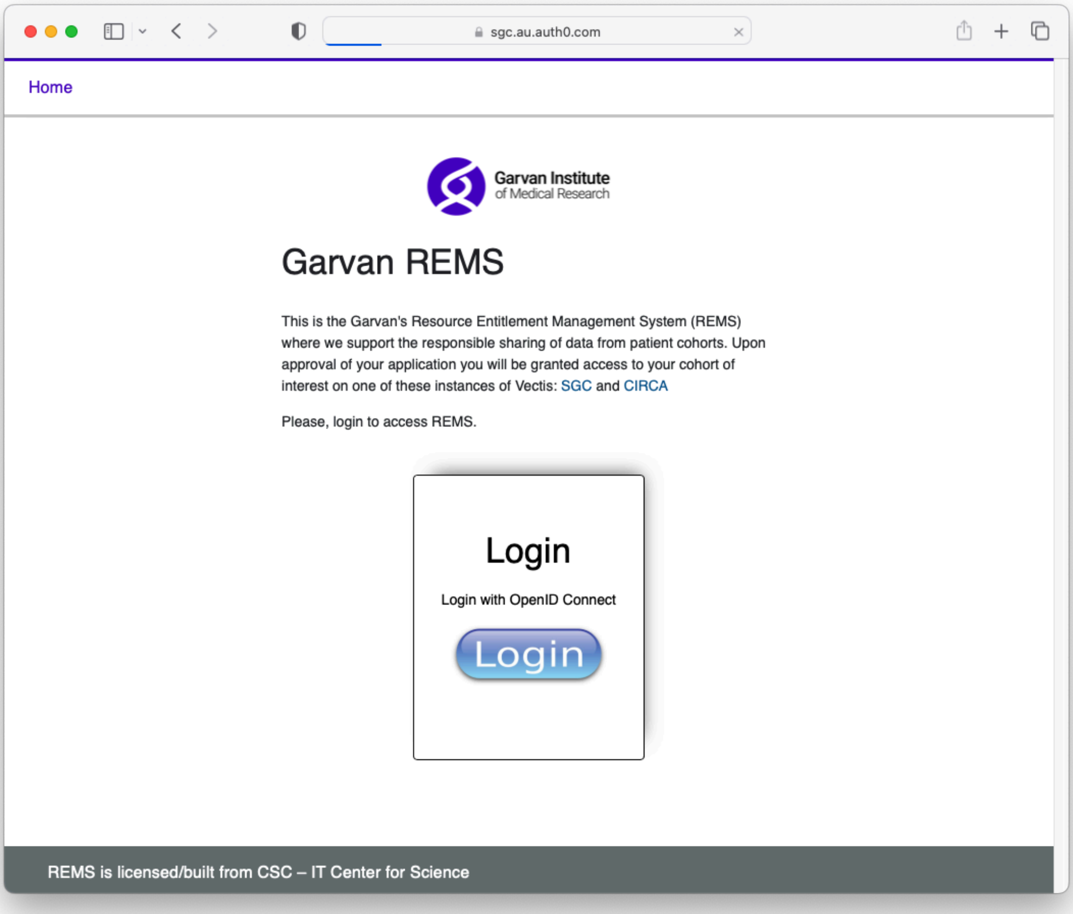
Please authenticate with eRA Commons in order to proceed. Your ORCID iD is optional.

Figure A1. Screenshot of the DUOS front page, taken during a session with a test instance of DUOS.

Appendix 2 - An Australian instance of REMS used in production

The Garvan Institute of Medical Research (Garvan) has used a publicly visible REMS instance[41] for several years. This instance manages applications to several biomedical datasets, most notably genomic data from the Medical Genome Reference Bank[42], for research purposes.

Figures A2-4 below are screenshots from various stages of the REMS application process.



Garvan REMS

This is the Garvan's Resource Entitlement Management System (REMS) where we support the responsible sharing of data from patient cohorts. Upon approval of your application you will be granted access to your cohort of interest on one of these instances of Vectis: SGC and CIRCA

Please, login to access REMS.

Login

Login with OpenID Connect

Login

REMS is licensed/built from CSC – IT Center for Science

Figure A2. Screenshot of Garvan's REMS instance: Home page, before authentication.

[41] https://rems.public.garvan.org.au/

[42] https://www.garvan.org.au/research/kinghorn-centre-for-clinical-genomics/research-programs/sydney-genomics-collaborative/mgrb

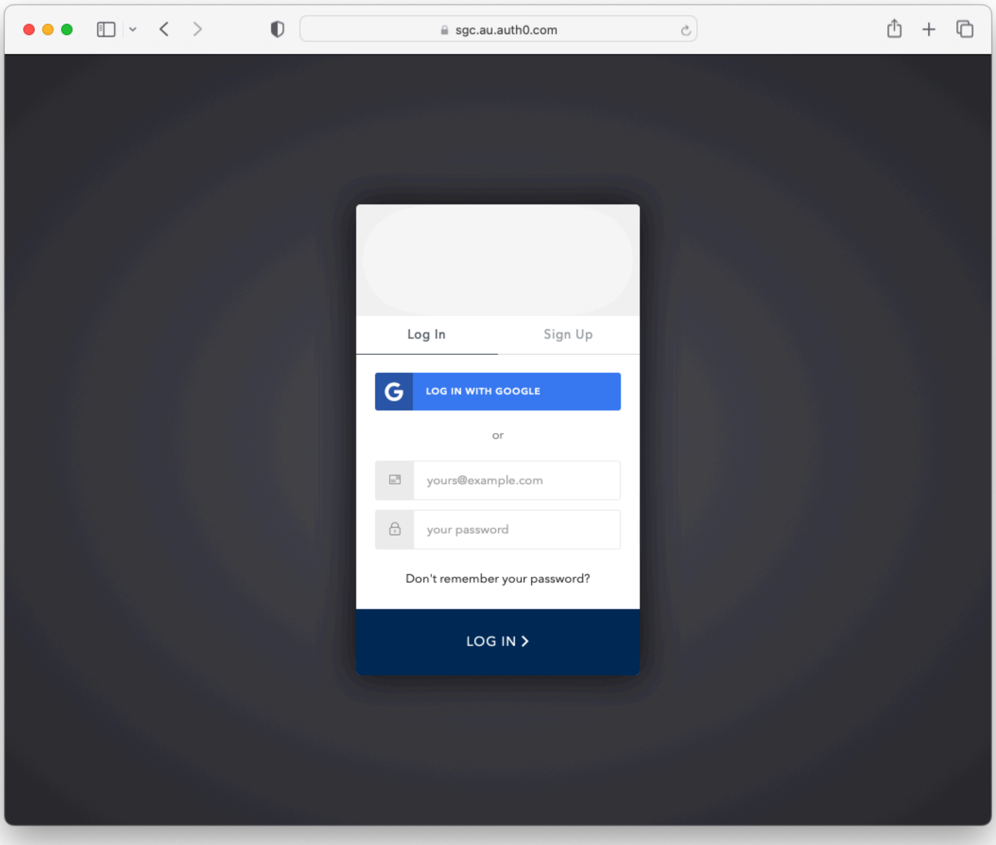
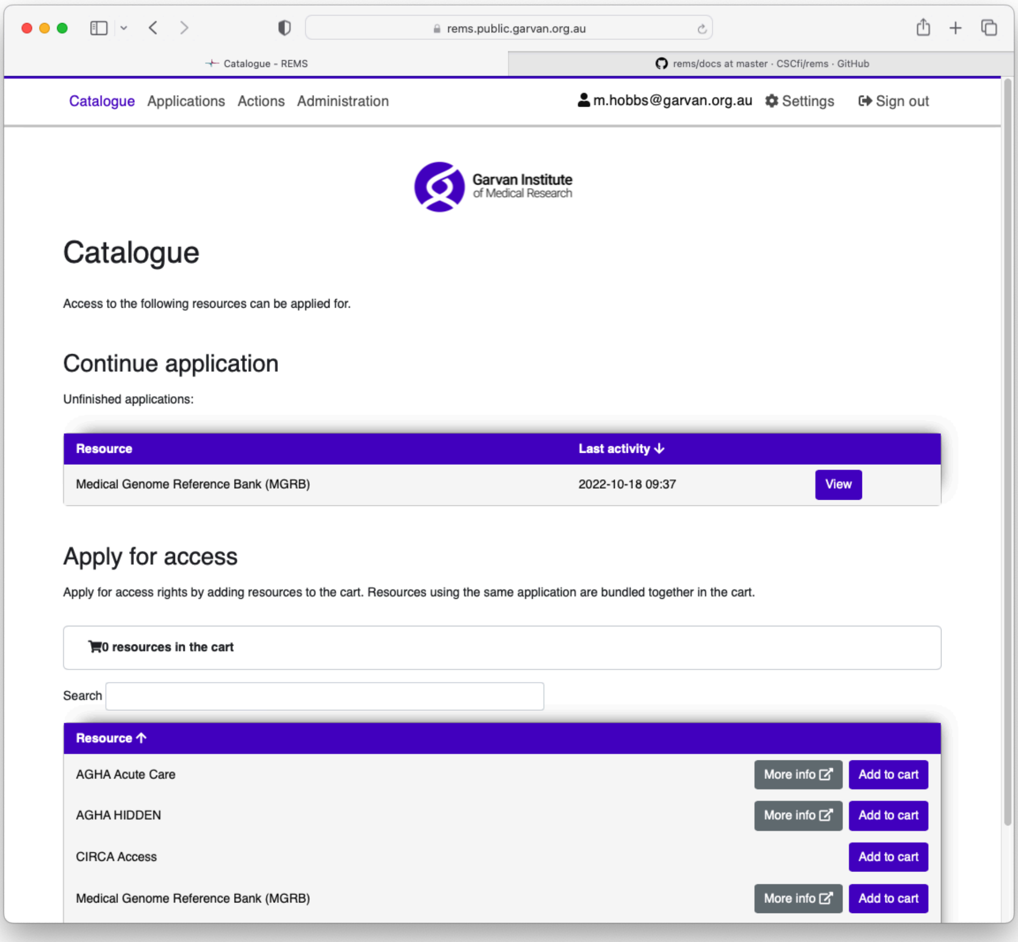


Figure A3. Screenshot of Garvan’s REMS instance: Login screen (using the Auth0 identity access management service).



rems.public.garvan.org.au
Catalogue - REMS
rems/docs at master · CSCfi/rems · GitHub
Catalogue
Applications
Actions
Administration
m.hobbs@garvan.org.au
Settings
Sign out
Garvan Institute of Medical Research

Catalogue

Access to the following resources can be applied for.

Continue application

Unfinished applications:

Resource	Last activity ↓	
Medical Genome Reference Bank (MGRB)	2022-10-18 09:37	View

Apply for access

Apply for access rights by adding resources to the cart. Resources using the same application are bundled together in the cart.

0 resources in the cart

Search

Resource ↑		
AGHA Acute Care	More info	Add to cart
AGHA HIDDEN	More info	Add to cart
CIRCA Access		Add to cart
Medical Genome Reference Bank (MGRB)	More info	Add to cart

Figure A4. Screenshot of Garvan's REMS instance: List of catalogue items and open applications.

Appendix 3 - Requirements Traceability Matrix

The requirements shown in the table below were presented as Tables 7-14 in the preceding Discovery Phase Report[43]. They are reproduced here in a consolidated form for convenience, with minor clarifications of language and the removal of extraneous comments.

ID	CATEGORY	REQUIREMENT
HGPPREQ-001	Mandatory	Dataset catalogue/summary details (metadata) should be able to be viewable by all authorised users of the system before data access application.
HGPPREQ-002	Mandatory	Researchers can use the system to apply for access to data.
HGPPREQ-003	Mandatory	The system presents the terms and conditions for access to data.
HGPPREQ-010	Should have	PIs can use the platform to add researchers in their group to a single data access application, and add and remove team members through the life of the project so as to easily establish seamless access for all members of the project.
HGPPREQ-011	Should have	The system should allow appropriately authorised users to find out what datasets people have requested and been granted access to, which people have been granted access to datasets, and to know the duration of the agreement so as to effectively manage the holdings and mitigate any risks from holding sensitive genomic datasets.
HGPPREQ-012	Nice to have	After approval is granted, the platform should allow a user to cross reference their data access application with their interface to the data, to know their proposed use of each dataset so as to only use the shared data in accordance with their proposed use original application and the terms of the data sharing agreement.
HGPPREQ-013	Nice to have	PIs can use the platform to know whether there are regular (annual?) reporting requirements linked to the granting of access to data so as to effectively deliver on the agreed responsibilities as a data recipient.
HGPPREQ-014	Nice to have	PI is provided with a simple way to provide reports on the usage of the datasets shared with them, when requested so as to effectively deliver on the agreed responsibilities as a data recipient.
HGPPREQ-015	Mandatory	DAC members have access to all components of data access applications together, electronically, so as to efficiently, quickly and confidently complete their review.
HGPPREQ-016	Mandatory	DAC members have an automated workflow for data access requests, so as to be able to easily add remarks for the DAC administrators to approve, reject or request more information from the applicant.
HGPPREQ-019	Mandatory	DAC members can electronically indicate the approval or rejection of an application, so as to eliminate the need for printing, signing and scanning. The user's credential as used in the system is reliable enough to support this feature.

[43] https://zenodo.org/record/6644050

HGPPREQ-022	Mandatory	DAC administrators can see all the details together to be able to review a data access request, and easily determine the current status of data requests at any time so as to efficiently process requests and not waste time going back to the researcher for further information.
HGPPREQ-023	Mandatory	DAC administrators should have a workflow and communication tool, so that they can easily communicate with the applicant, the DAC members, and data access team/data distributor: fielding questions, advising outcomes and notifying data teams to provision access.
HGPPREQ-026	Mandatory	DAC administrators can record and track approved applications so as to efficiently report on what datasets have been shared and to whom.
HGPPREQ-027	Mandatory	The system should support the DAC's requirement to compare documents such as (for example) the terms and conditions, the consent, the HREC and the ethics statement, by allowing upload of attachments.
HGPPREQ-039	Mandatory	Executed data sharing agreements/contracts must be able to be stored in the system and linked to the original data access request.
HGPPREQ-042	Mandatory	The system should allow for automated notifications to be sent to users when request status changes (such as progress, approval, rejection and request for more information, expiry).
HGPPREQ-051	Mandatory	The system should provide for the DAC members to be well informed enough to be able to complete the cohort sign off respecting the consent/rule preferences of the underlying dataset/cohort (of which I may have not been a part of formulating i.e the data committee may not be exactly the same people who collected the data) (for example dataset terms and conditions or DUO codes should be visible to the DAC so they are informed even if new).
HGPPREQ-017	Should have	DAC members have access to an electronic archive of the historic approvals dating back for a reasonable period (such as 5 years) after the research project ends, so as to cross reference new applications with previous approvals made.
HGPPREQ-018	Should have	All parties involved in an application can attach documents to that application so that all parties have access to external supplementary information if needed; and a log should be kept of when the attachment was added.
HGPPREQ-020	Should have	The system should have the facility to capture an open researcher ID and project/activity ID (e.g. ORCID and ROR) and a placeholder for future RAID.
HGPPREQ-021	Should have	DAC members can know the verified identity and institution/workgroup details of the applicant so as to avoid the need for double checking with institutions and to be assured that they are only granting access to appropriate applicants.
HGPPREQ-025	Should have	The system should allow users to add notes or log information into an approved application for the life of the project so future DAC members can derive insights as to how the data was used, what the outputs were, how the compliance was done etc.
HGPPREQ-030	Should have	The system should allow authorised people to edit/change the data access request forms. A record should be kept of the history of changes (for example like git does). There should be a revision management process.

HGPPREQ-032	Should have	DAC administrators can record and follow-up on any periodical/publication/other reporting requirements via the platform. (Ability to track whether an annual report is due and has been done for data collection). Report attached to the DAC Sharing Agreement/Application would be nice to have.
HGPPREQ-047	Should have	The DAC should be able to restrict the permissions of an applicant to a subset of the complete cohort population (for example DUO codes to be applied at the record level not dataset level).
HGPPREQ-048	Should have	The DAC should be able to restrict access of an application to a subset of data types in the cohort (e.g. BAMs only, VCFs only, not FASTQs etc).
HGPPREQ-050	Should have	The DAC should be able to enable access to the data cohort respecting the individual consent preferences of all the participants in the cohort (e.g. dynamic consent).
SUPPREQ-108	Should have	Scalability: able to accommodate growth; for example - thousands of data access requests.
SUPPREQ-109	Should have	Development pipeline potential: able to accommodate new additions and expand functionality.
HGPPREQ-110	Should have	The system should have the capacity to communicate with other components in the HGPP research lifecycle via automated interfaces such as an API.
HGPPREQ-111	Should have	The ability to partially populate a new application from other data sources such as ORCID and/or RAID, for example so you don't have to rekey your "life science profile" over multiple applications.
HGPPREQ-040	Nice to have	DAC and data providers have access to usage statistics/metrics for the datasets under their control such that they can be used for showing the relative usefulness of the data sets for funding (funding both of the dac infrastructure, but also of individual scientists who have contributed data sets).
HGPPREQ-043	Nice to have	The DAC system within the platform should accept an electronic application in a standardised form such that someone can apply quickly across multiple DACs and multiple cohorts in one application process - will be related to virtual cohorts - where those virtual cohorts span multiple data owners and cohorts (we are imagining being able to avoid having to copy and paste 50 identical ethics approvals, project description etc).
HGPPREQ-049	Nice to have	The system must be able to restrict access of an applicant to genomic regions that match the genomic regions of interest to the applicant (as opposed to the whole genome). This is an "avoid incidental findings" requirement.