

Fast Track Report

Walter Eliza Hall Institute of Medical Research

REDMANE Intake 14

Web Development Sub-Team

Nash Anuwar, Yashasvi Goswami, Soham Das, Ying Ying Lee, Manya Garg, Yi Lu

Supervised by Rowland Mosbergen

Summary

This document provides a concise overview of the REDMANE Data Registry system and its key components. The aim of the Fast Track Report is to help new interns or interested contributors familiarize themselves with the projects, its architecture, and development of workflow in rapid time. Estimated reading times are included for each section to help readers efficiently navigate the material and prioritize relevant parts.

Problem Statement (20 min)

Cross-organisational multi-omics projects generate large volumes of samples and datasets, but frequently face constraints in cloud storage, compute resources and data infrastructure. Because data is distributed across institutions, tracking relationships between donors, samples and datasets become fragmented and inconsistent; documentation alone is rarely sufficient.

Further challenges arise around secure result-sharing with non-technical collaborators, and around the diversity of data types in multi-omics research. Raw and processed data are too large to move cheaply and should remain stored in place, while summarised data is small enough for cloud-based sharing. A system must accommodate both dynamics without becoming rigid or project specific.

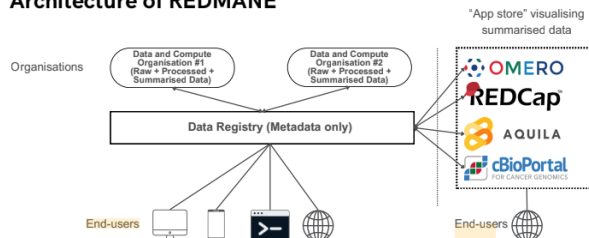
Many Research Data Management (RDM) initiatives prioritise governance and compliance

over usability, leaving data managers with few practical tools for ingestion and reuse at scale. This drives a core design tension: highly general systems risk low researcher adoption, while narrowly tailored ones struggle to scale across domains. The result is that many organisations—particularly smaller ones—find it difficult to maintain research infrastructure while keeping project data FAIR (Findable, Accessible Interoperable and Reuseable).

REDMANE (40 min)

Which stands for REsearch Data Management & ANalysis Environment, can be conceptualised as a library ecosystem. It provides a centralised data registry—analogous to a library catalogue—that tracks datasets held across distributed organisational locations. The system offers a suite of tools to streamline data management workflows for researchers and data stewards. REDMANE is designed primarily for use with de-identified datasets.

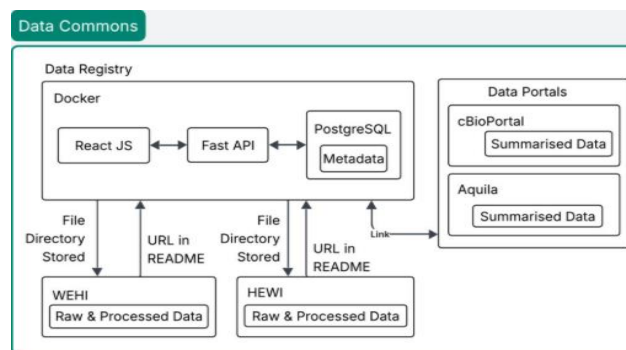
Architecture of REDMANE



Above is a diagram architecture of REDMANE. The Data Registry points to data storage (raw, processed, and summarised data) across different

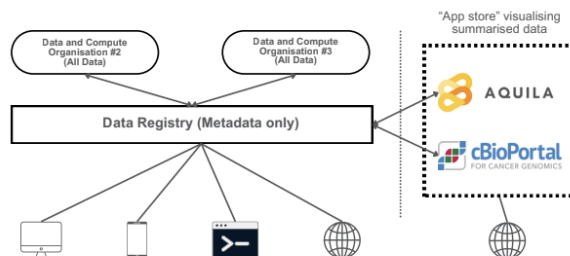
organisations and to the applications. The data storage and applications point back to the data registry.

The data registry stores metadata about refined datasets and sample information, and maintains references to raw, processed and summarised data locations, as well as associated data portals. These external resources in turn link back to the registry. The ecosystem also provides tooling that enables data scientists to quickly locate, access and load datasets into their preferred analysis software.

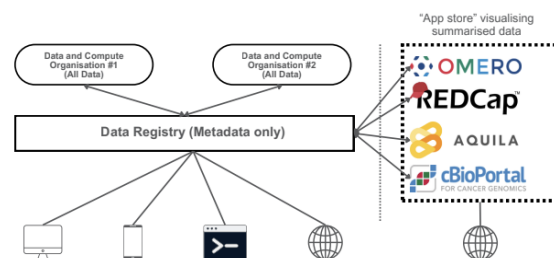


Data Portals

Summarised data should be easily visualisable for non-computational researchers through web-based portals, such as cBioPortal, Degust and OMERO. Given the heterogeneity of multi-omics data, different portals may require different portals, and the system must support custom portals such as Shiny/R applications. The diagrams below exemplifies that the data registry allows projects to select portals from an “App Store” model. Each portal is either shared as a Software-as-a-Service instance or deployed independently per project.



Project 2



Project 1

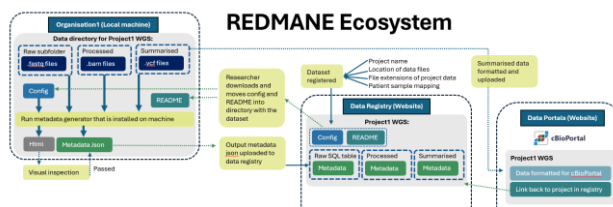
Dataset Population

How datasets are initialised from raw data to an initial summarised data. Check Slide 39-44 of ‘Introduction to REDMANE’ [here](#).

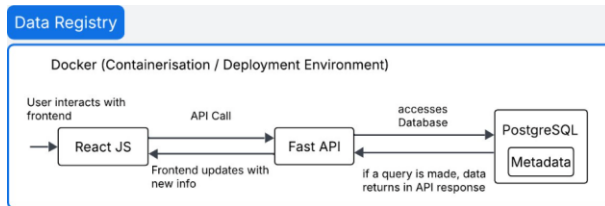
Examples of REDMANE Use-Cases

Check [here](#) for examples, (slide 7-8).

Additionally linked is another diagram of the data commons that considers Data Ingestion’s scope.



Web Development Scope (75 min)



Here is the scope of what primarily the web development team deals with. Restated, above is a diagram that shows tech stacks;

- + [ReactJS](#)
- + [Fast API](#)
- + [PostgreSQL](#)
- + [Docker](#)

The way to run the data registry is to first run PostgreSQL, then Fast API, then ReactJS. Running Docker runs it simpler altogether but currently doesn't compose up ReactJS (frontend).

Here is Rowland's technical diaries for running the data registry on Mac/Linux. To highlight, these are run on Home Brew, [ReactJS](#), [Fast API](#), [PostgreSQL](#).

The current GitHub Docker README has proficient instructions on how to compose Fast API and PostgreSQL, check it out [here](#).

At this point, this is a good opportunity to try running the data registry on your local device with the instructions above.

A barometer for a successful host is to see if it matches the publicly [hosted data registry](#).

Make sure to check:

- + Datasets
- + Projects
- + Patients

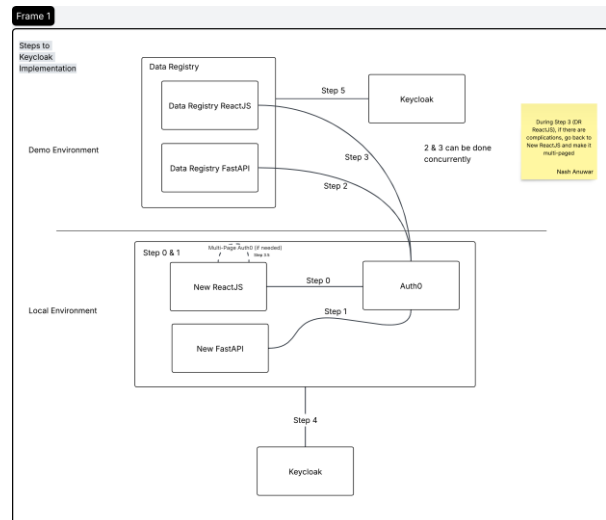
Regarding data population.

Authentication (1 Hour)

Authentication is important in REDMANE because it serves as a foundational layer of security. Without robust authentication, the data registry is highly vulnerable to attacks.

Next is Intake 14's road to authentication implementation.

You can also see the diagram [here](#).

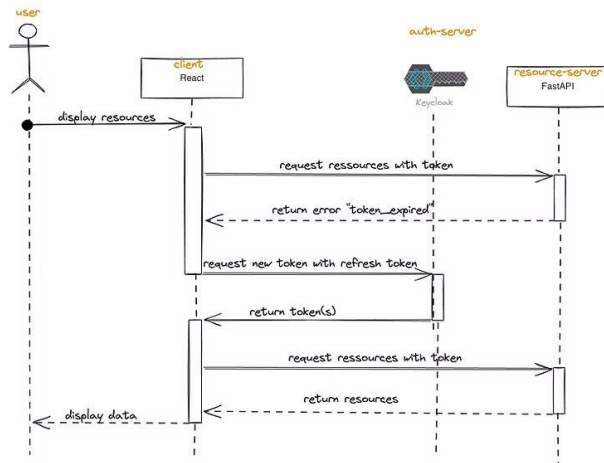


Our end goal is to implement Keycloak on the Data Registry. We implemented Auth0 on local apps and then on the data registry as it is a much easier IdP than Keycloak. Implementing Keycloak on the data registry straight away would cause difficulty in debugging.

The chosen library for authentication is [OIDC](#).

After implementing Auth0, we changed lines dealing with authentication to '[react-oidc-context](#)'. After this, it took under 5 file changes to change the IdP and have the local app or data registry run Keycloak.

Below is a diagram of JWT token handling from FastAPI to React JS.



This ensures that unauthorised access is prohibited. This [technical diary](#) shows how to run Keycloak using Docker and the three main repositories.

Future Extensibility

When the above is configured, the next steps would be to configure Keycloak for new interns/contributors. The simplest flow can potentially be:

1. Run Keycloak on a persistent server with a Postgres backend so config survives restarts
2. Connect Keycloak to the university's existing Active Directory/LDAP so staff can log in with their existing university credentials
3. Create one realm for REDMANE and configure the client once

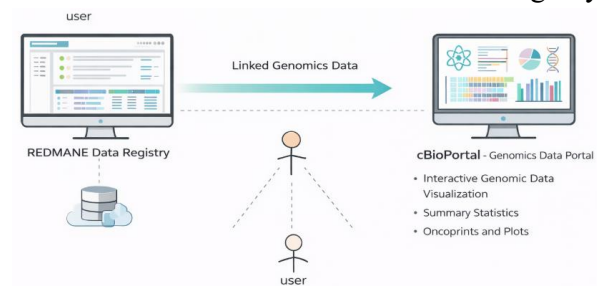
For onboarding new interns/users:

1. Admin creates the user in Keycloak (or the user already exists via LDAP)
2. Admin invites them to the relevant project via the existing invite system in the app
3. User logs in with their university credentials, no new account needed

The key recommendation is to connect Keycloak to WEHI's existing WEHI credentials. This can help password management, no separate passwords to manage and access is automatically revoked when their contract ends.

CBioPortal (30 min)

CBioPortal is a Data Portal in the REDMANE ecosystem to visualise and summarise genomics data linked from the data registry.



This intake integrated cBioPortal as an external Data Portal within REDMANE, dynamically linked from dataset metadata (via `url_cbioportal`) rather than hardcoded URLs. HTTPS is enforced for all portal links. Clicking a cBioPortal button in the Data Registry redirects users to the relevant cBioPortal endpoint, demonstrating the intended REDMANE-cBioPortal navigation flow.

Dev Notes

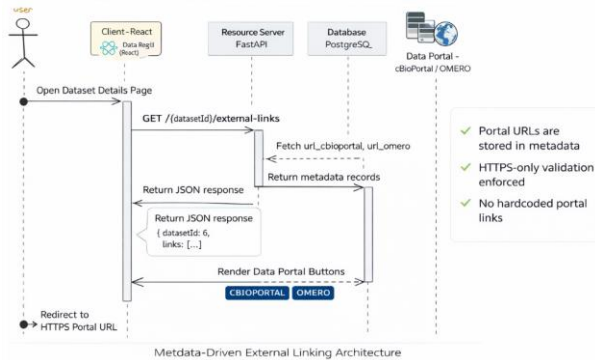
CBioPortal is hosted on the Data Portals behind HTTPS, routed via Nginx reverse proxy, and containerised using Docker Compose for consistent deployment.

Current Limitations

Dataset-specific URLs and real data are not yet attached; placeholder entry pages are used for the demo.

Summary

This fast-track report has outlined the current state of REDMANE's architecture, integrations and tooling. The work described here forms a foundation for future development as the system scales to support broader research collaboration across institutions.



Future Implementation

Once a dataset-specific cBioPortal page is available, no front-end changes are needed—simply update the metadata entry to point to the specific URL (e.g: <https://cbioportal.domain/study?id=TDE0001>). The frontend dynamically renders links from metadata, so the button will automatically redirect correctly.

Readme feature (10 min)

Users can download a README directly from the dataset detail page, containing a dynamic backlink to that dataset in the registry. This is generated client-side in 'SingleDataset.jsx' using `window.location.origin` and the dataset ID.

A "Download README" button was added below the Data Portals section. No backend changes were required—the generated README automatically adapts the deployment domain based on the frontend origin, making it environment-agnostic.

Docker

Docker containers to deploy the data registry was implemented in this intake in collaboration with the sysadmin team. See the DevOps Sysadmin Team's [fast track report](#) for more technical notes.