# Home

This is the home of the Portable Legal Consent - Common Genomic Research space.

Please navigate the Specification below for background on the project, REST API specifications, and other documentation as it evolves.  Feel free to comment on all aspects.



# PLC Specification

## Portable Legal Consent Specification

This document provides a dictionary of PLC concepts and specification of the operations of the individual system components with each other and with outside actors.

**Portable Legal Consent to Research is the full name of the project. It is currently referred to a CtR (Consent to Research) in most of the client conversation and at this point there are no grounds to disambiguate between PLC and CtR, they are the same. We may at some later point decide to separate the names for the specification and a particular implementation.**

### What is PLC all about

As formulated on the weconsent.us site:
"Portable Legal Consent (PLC) is being developed as a tool to allow patients to tell the doctors, researchers, and companies that are experimenting on them, that they, the patients own the rights to the data generated from their bodies. When a patient chooses PLC, it's a statement that what the patient desires is for the data to be shared broadly in the public domain, to serve scientific progress as a whole, regardless of the particular individual or institution that makes the breakthrough."

A Nature Genetics editorial (Nature Genetics 44, 357 (2012) doi:10.1038/ng.2244) on PLC can be found at http://www.nature.com/ng/journal/v44/n4/full/ng.2244.html.

Wilbanks on the fundamental problem to be addressed: "People are going to have their clinical data and they have their genotypes. How can they be made available so someone can make a model out of them?"
Wilbanks in reference to the Metcalfe's law: http://en.wikipedia.org/wiki/Metcalfe's_law "We don't have enough patients with compatibly communicating information, available to be connected so that the systemic value of these patient data goes up scientifically and economically."
Wilbanks on the goal: "port Creative Commons methodology into a world, where we don't just need the right to share and remix but to actually do assembly of patient data into networks"
Wilbanks on the modus operandi: "data get distributed in a patient driven manner ... all the researchers have to do is to download it"
Wilbanks on the uploaded personal information: "we are using this information not for redistribution but to generate a unique ID for you"
Wilbanks on the system design (paraphrased): The intent for the both the standard and the required technology is to be simple, weak and open so it "can be used in an unanticipated manner.

Summarizing this and the various communications we had I'd the key goals enabled by the PLC are:

*Enabling institutions and organizations in the patient's trust to solicit data sharing by providing the patient with means to get informed, consent and share data
*Enabling the informed patient to share data in a distributed environment
*Enabling the informed patient to share data unidentified (within limits as for example presented by genomic data)
*Enable the system to notify the patient in case shared data reveal actionable information if the patient chooses to be informed in such events
*Enabling the patient to stay in control of which data are shared, when and for how long
*Enabling the research community to discover, aggregate and use shared data
*Enabling the research community to identify and consolidate different (or redundant/repeated) pieced of data shared by the same patient via a globally unique patient identifier

### Actors and Systems

- **Systems:**
  - **CGRE - Common Genomics Research Entity** - any organization that implements CtR, e.g provide the capability for patients to consent and upload data, and share the data with Research Entities
  - **RoR - Registry of registries**, e.g. Registry of all CGREs that serves as a central discovery point for Research Entities who want to access data from one or more CGRE
- **Actors:**
  - **Patient** - person who wants/has consented to share data within the CtR framework
  - **CGRE Authorizer** - person affiliated with a particular CGRE who qualifies a RE to access data captured by that CGRE and

require such qualification (this is called a local qualification).
- **RoR Authorizer** - person affiliated with the RoR who qualifies a RE to access the data captured by any CGRE registered with the RoR (this is called a global qualification.
- **RoR Registrar** - person affiliated with the RoR that approves registration requests from CGRE's to the RoR
- **RE - Research Entity** - any researcher who wants or has to access data captured by a CGRE
- **QRE - Qualified Research Entity** - some data are only accessible to RE that have been approved by some authority. This authority is positioned either locally with a particular CGRE or globally with the RoR.

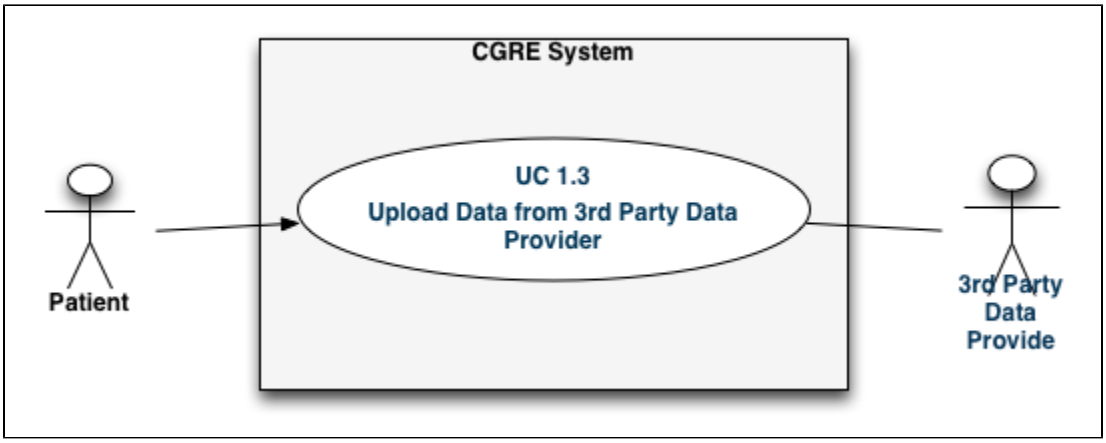
### Use Cases

Interactions between the Patient and the CGRE System
Interactions between the Research Entity and the CGRE System
Interactions of 3rd party data providers with CGRE System

# 3rd-party Vendor Integration

## Interactions of 3rd Party Data Provider System with CGRE System

*Specification on how vendors, typically providers that produce data that can be shared by patients via PLC, can integrate their services with the CrT platform.*




### Discussion

**There are 2 fundamental options of how a data provider can interact with the CrT platform:**

- Act as a CGRE by implementing the consent wizard and provide data sharing services for Research Entities
- Act as authorized CGRE client on behalf of the Patient

#### 3rd party data provider acts as CGRE System

Some providers of Whole Genome or Exome sequencing, or SNP genotyping are also storing the data in the context of customer accounts. The same is true for hospitals which store medical data for patients. It may make the most sense for some organizations like these to add the capability of sharing data from consenting patients with Research Entities by implementing the full CGRE System API.

**Benefits**: The data producer is the primary source of record, with the most up to date data. By alleviating the need to copy the data to other intermediate storage locations before being processed by a Research Entity sources of error are removed and additional transfer and duplicate storage of potentially large data files can be avoided.

**Caveats**: The 3rd party vendor would need to implement the CGRE API und also potentially subjected to CGRE and regulatory oversight. The biggest concern may come from legal reasons and liability concerns.

#### 3rd party data provider acts as authorized CGRE client

If the option of acting as CGRE is not viable then the key objective is to remove the need for a patient to download the data from the data provider just to upload them again to the CGRE System. We propose a workflow where the 3rd Party Data Provider System is authorized by the Patient to upload the data on its behalf directly to the CHRE System. There are 2 different scenarios we have considered so far:

- Patient is already customer of 3rd Party Data Provider System
- Patient is not yet customer of 3rd Party Data Provider System

For both cases it is our explicit goal to enable interoperation in a patient-controlled and temporary manner. The patient, who has credentials in

both the CGRE and 3rd party system, authorizes a trust relationship that only enables the desired data exchange and expires immediately thereafter.

**Patient is not yet customer of 3rd party data provider**
Patients who have no data of a particular moiety can choose from a list of providers that can provide this kind of data published by the CGRE System. In it's simplest form this is just a list of hyperlinks, each characterized by one or more tags, and optionally associated with a short vendor-provided description of the linked service. The hyperlink targets are determined by the vendor and could point either to a generic information or signup page. It is possible that future elaborations will result in a specification for a vendor provided registration service that would permit submission of a complete registration data bundle from a CGRE.

**Patient is already customer of 3rd party data provider**
If the patient has already data stored on the 3rd party vendor's system then this system should enable the Patient (customer of this vendor) to authorize the vendor system to directly upload the existing data to a particular CGRE system, chosen by the Patient. There are again two ways to invoke this kind of functionality:

- **PULL** - The Patient authorizes CGRE System to request (pull) the data from the 3rd party vendor's system.
- **PUSH** - The Patient authorizes 3rd party vendor's system to upload (push) to a selected CGRE system.

While the **PULL** modality appears to be preferable for the reasons listed below we are open to consider solutions involving the **PULL** mechanism:

- The 3rd party never has access to the Patients CGRE credentials and the PGUID
- Most vendors will already have mechanisms in place for their customers to download their data

#### UC 1.3 Upload Data from 3rd Party Data Provider System via CGRE System (PULL)

**User stories**

- UC 1.3 US 1 As a patient I want the CGRE System to retrieve my SNP genotype file directly from 23andme.

**Precondition**

- Patient has data and credentials to download the data from 3rd Party Data Provider System
- 3rd Party Data Provider System supports direct data exchange with a CGRE system (implement API) and is listed there (have published services endpoint to CGRE System)
- Patient is logged into the CGRE System, e.g. the CGRE System has identified PGUID linked to the Patient
- Patient has navigated to the data upload interface

**Basic Workflow**

1. Patient selects 3rd Party Data Provider System from a provided list
2. Patient is then redirected to the 3rd Party Data Provider System
3. Patient (Resource Owner) authenticates with and authorizes data access at the 3rd Party Data Provider System (Server) from the CGRE system (Client) via OAuth
4. CGRE System sends (GET) request for list of available files (potentially including metada and/or descriptions of some sort)
5. CGRE System displays 3rd party vendor system provided lists of available files to Patient
6. Patient selects the file(s) to be retrieved from 3rd Party Data Provider System by the CGRE
7. CGRE system sends (GET) request for selected file(s)
8. 3rd Party Data Provider System compresses all requested files + metadata into a single file
9. 3rd Party Data Provider System stores the compressed data file in a download location for predetermined period of time
10. 3rd Party Data Provider System sends a download URL back to the requesting CGRE
    a. this will allow the offload of large file transfer to a different more appropriate channel rather than perusing the messaging channel
    b. the download URL will be valid for a download to start successfully only within the specified expiration time
    c. access to the download URL will be secured by the same authentication used for the file request (OAuth)
11. CGRE System downloads the file from the specified URL within the time limit
12. CGRE System resolves the PGUID based on the authentication context
13. CGRE system stores the file associated with the PGUID

**Postcondition**

- CGRE System public data store contains the data file associated with the PGUID and the Data Type Tags
- Patient is informed that the data were successfully uploaded
- (FUTURE) Other actions maybe triggered notifying other actors or system components of the event

#### UC 1.3 Upload Data from 3rd Party Data Provider System (PULL)

**User stories**

- UC 1.3 US 1 As a patient I want 23andme to send my SNP genotype file directly to the CGRE System to be shared from there.

**Precondition**

- RoR publishes a list of registered CGRE (end points)
- Patient has data and credentials to access 3rd Data Provider System
- Patient has successfully completed the Consent Wizard, has a user account (and a PGUID) at a RoR registered CGRE

- Patient is logged into the 3rd Party Data Provider System
- Patient has navigated to the 'share via CGRE' interface

**Basic Workflow**

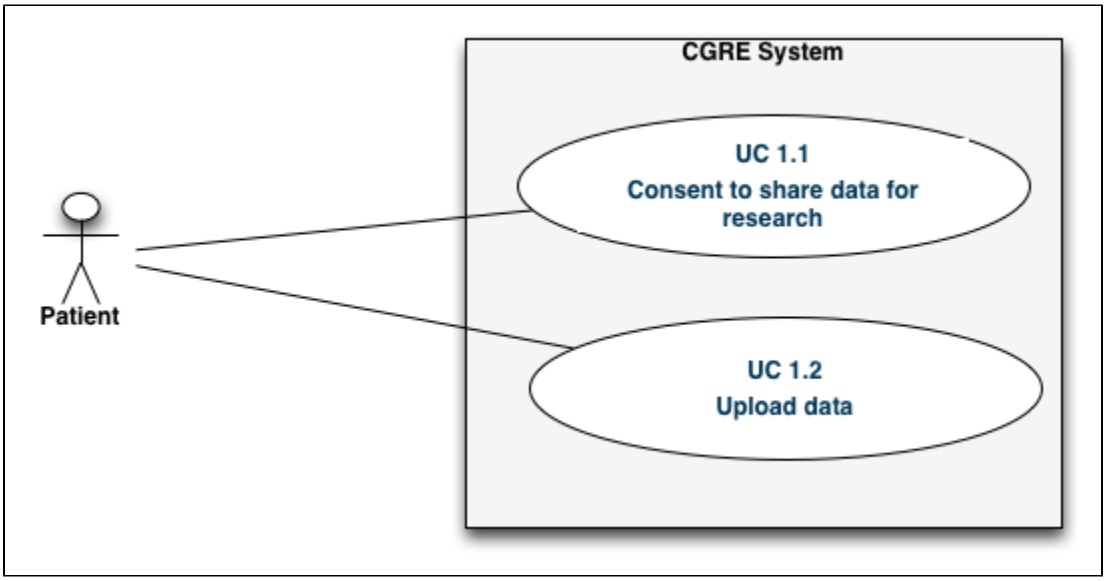
1. 3rd Party Data Provider System gets lists of CGRE endpoints from RoR and displays is
2. Patient (as a user of the 3rd Party Data Provider System) selects desired single CGRE endpoint
3. Patient authorizes data access from CGRE system at the 3rd party vendor system
4. 3rd Party Data Provider System displays available download options (files) to Patient
5. Patient selects the file(s) to be retrieved from 3rd Party Data Provider System by the CGRE
6. Patient (Resource Owner) authenticates with and authorizes data access at the CGRE system (Server) from the 3rd Party Data Provider System (Client) via OAuth
7. CGRE system sends (GET) request for selected file(s)
8. 3rd party vendor system compresses all requested files + metadata into a single file
9. 3rd party vendor system stores the compressed data file in a download location for predetermined period of time
10. 3rd party vendor system sends a download URL back to the requesting CGRE
    a. this will allow the offload of large file transfer to a different more appropriate channel rather than perusing the messaging channel
    b. the download URL will be valid for a download to start successfully only within the specified expiration time
    c. access to the download URL will be secured by the same authentication used for the file request (OAuth)
11. CGRE System downloads the file from the specified URL within the time limit
12. CGRE System resolves the PGUID based on the authentication context
13. CGRE system stores the file associated with the PGUID

**Postcondition**

- CGRE System public data store contains the data file associated with the PGUID and the Data Type Tags
- Patient is informed that the data were successfully uploaded
- (FUTURE) Other actions maybe triggered notifying other actors or system components of the event

# PLC Patient-CGRE Use Cases

## Use Cases involving the interaction of Patients with the CGRE System




**Glossary**

- Consent Wizard - a sequence of web pages that inform the Patient about CtR, and present forms for Consent Options, Patient Data and Patient Account Information leading to a account for the consenting Patient in the CGRE System
- Consent Options (specified by the Patient, binding conditions to be observed by data users (Research Entities)
    - Do not attempt to re-identify me (Patient)
    - Share new data with others as I have share with you
    - Share your research with the public under open access terms

- Patient Data (used to generate a globally unique patient identifier - PGUID - values as referenced on birth certificate if subsequent change occurred)
    - First Name (immutable)
    - Birth (Maiden) Last Name (immutable)
    - Date of Birth (immutable)

    - Place of Birth (immutable)
    - Country of Birth (immutable)

- Patient Account Information (Patient log in that maps one to one to PGUID) - can be updated
    - Full Name
    - Username (unique and immutable for a particular CGRE System)
    - Password (can be changed)
    - Email (can be changed)
    - One or more challenge questions (fav. teacher, movie, pet etc.) (can be changed)

- Data Type Tags - refers to both content, format and version of a Patient Data file (Lists are preliminary), the data model should allow mapping of (tag) values to 0 to * categories.
    - Content
        - SNP
        - Exome
        - Genome - FullSequence
        - EHR - Electronic Health Record
        - FHH - Family Health History
    - Format
        - 23andme
        - Knome
        - DecodeMe
        - Bluebutton
        - Hl7
        - Plain Text
        - PDF
        - Word
    - Version - free text (maybe superseded by controlled vocabulary later)

#### Nonfunctional Requirements

The CGRE System hosts both private patient information used to create a PGUID and a Patient user account and public , e.g. shareable data uploaded by the Patient to be used by Research Entities. The is an explicit requirement to completely separate the storage of the private data from the storage of the public data, hence the private Patient Data will be stored associated with the PGUID in the private data store and the shareable data will be stored associated with the PGUID in a different data store.

#### UC 1.1 Consent to share data for research

**User stories**

- UC 1.1 US 1 As a patient I want to learn about CtR and consent to share information such as genomic profiles, medical records or other data relevant to research.
- UC 1.1 US 2 As a patient I want the CGRE to recognize whether I am already registered (consented) at this CGRE or any other CGRE.

**Pre Condition**

- none

**Workflow**

1. Patient accesses and navigates through the Consent Wizard on the CGRE system web portal
    a. Patient agrees with all criteria that confirm his understanding of the consent
    b. Patient indicates his preferences for a small number (<5) of options concerning the sharing of her data
    c. Patient fills out a form with a small set of personal information, that in combination ensure global uniqueness among all patients registered with any CGRE system instance
    d. Patient fills out an CGRE account form to enter Patient Account Information
    e. Patient certifies consent via electronic signature and submits all data collected via the Consent Wizard

2. CGRE System receives Consent Wizard data

3. CGRE System validates Consent Wizard data
    a. checks for completeness (are all fields filled out, do the value make sense for the field)
    b. if checks fail
        i. the data together with actionable error information is sent back to Patient for correction and resubmission
    c. if checks pass
        i. proceed

4. CGRE System system checks the Patient Account Information whether the username already exists
    a. if no account with the same username is found
        i. a patient user account is created by the CGRE System from the Patient Account Information
    b. if an identical username already exists
        i. present the Patient with a choice change username or
            1. if login as the existing user redirect to login screen
            2. if change username resubmit with changed username

5. CGRE System checks whether there already is an existing record with identical values
    a. all Patient Data tokens get trimmed (removal of leading and trailing white spaces, and collapsing multiple white spaces into a single one)
    b. checking the local CGRE Patient records using the trimmed tokens
    c. (FUTURE) *checking patient records across all CGREs registered with the RoR*
    d. if no existing record matches the personal Patient Data:
        i. system stores the trimmed Patient Data tokens otherwise unaltered (keep the original case as entered)
        ii. a PGUID is created from the normalized Patient Data
    e. if a match with the personal Patient Data is found in the existing records:
        i. the PGUID of the existing record is returned to Patient
        ii. Patient confirms that this record by specifying additional information (correctly answering challenge question)

6. the CGRE System returns the generated PGUID see PLC SAD for algorithm

**Post Condition**

- The Patient has a patient user account in the (particular) CGRE System (instance)
- The Patient has a PGUID linked to the account above
- The Patient is logged into the CGRE System, e.g. the CGRE System has identified PGUID linked to the Patient

**Restful Service API**

URI: ${BASE-URI}/patient
Protocol: https
Operation: POST
Consumes: json
Input:

- PatientData
    - fullName - STRING
        - constraints - no white space only letters
        - length < 256 characters
    - email - STRING
        - email contraints
    - username - STRING
        - unique string
        - normalized to lowercase
        - length 6 -20 characters (including letters, hyphens, underscores, digits)
    - password - STRING
        - length 8 - 20 characters,
        - constraints: at least 1 digit, at least 1 capital letter, at least 1 special character (non alpha numeric)
    - recovery question - answer pairs - MAP<STRING, STRING>
        - 1 - 3 pairs
    - firstName - STRING
        - normalized to lowercase
        - length up to 50 characters
    - birthName - STRING
        - normalized to lowercase
        - length up to 50 characters
    - birthPlace - STRING
        - clarification - village, city - constant answer likely referencing what is noted on birth certificate
        - normalized to lowercase
        - length up to 100 characters
    - birthCountry - STRING
        - Will capture ISO 3166-1 alpha-2 country code (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2)
        - normalized to lowercase
        - length exactly 2 characters
    - birthDate - DATE

Produces: text/plain
Output: PGUID
Note: Since no identifying value (external id) was provided the client is responsible to maintain the relationship between real “patient” and the returned individual id.

#### UC 1.2 Upload/Share data

- UC 1.2 US 1 As a patient I want to share my data by uploading relevant data files via the CGRE web portal.
- UC 1.2 US 2 As a patient I want to be able to log into the CGRE web portal and have my data automatically associated with my PGUID.

**Pre Condition**

- Patient has successfully completed the Consent Wizard, has a user account and a PGUID
- Patient has to be logged into the CGRE System, e.g. the CGRE System has identified PGUID linked to the Patient
- Patient has data file of a Data Type that is recognized by the CGRE System

**Workflow**

1. Patient navigates to the data upload form
2. Patient selects the data file to be uploaded and shared
3. Patient selects the appropriate Data Type Tags for Content, Format and Version (all optional)
4. Patient optionally provides additional information via a text field (<256 characters)
5. Patient submits the data upload form containing the PGUID added automatically by the system
6. CGRE System stores the data file associated with the PGUID and the Data Type Tags
7. CGRE System sends a confirmation message back to the Patient

**Post Condition**

- CGRE System public data store contains the data file associated with the PGUID and the Data Type Tags
- Patient is informed that the data were successfully uploaded
- (FUTURE) Other actions maybe triggered notifying other actors or system components of the event

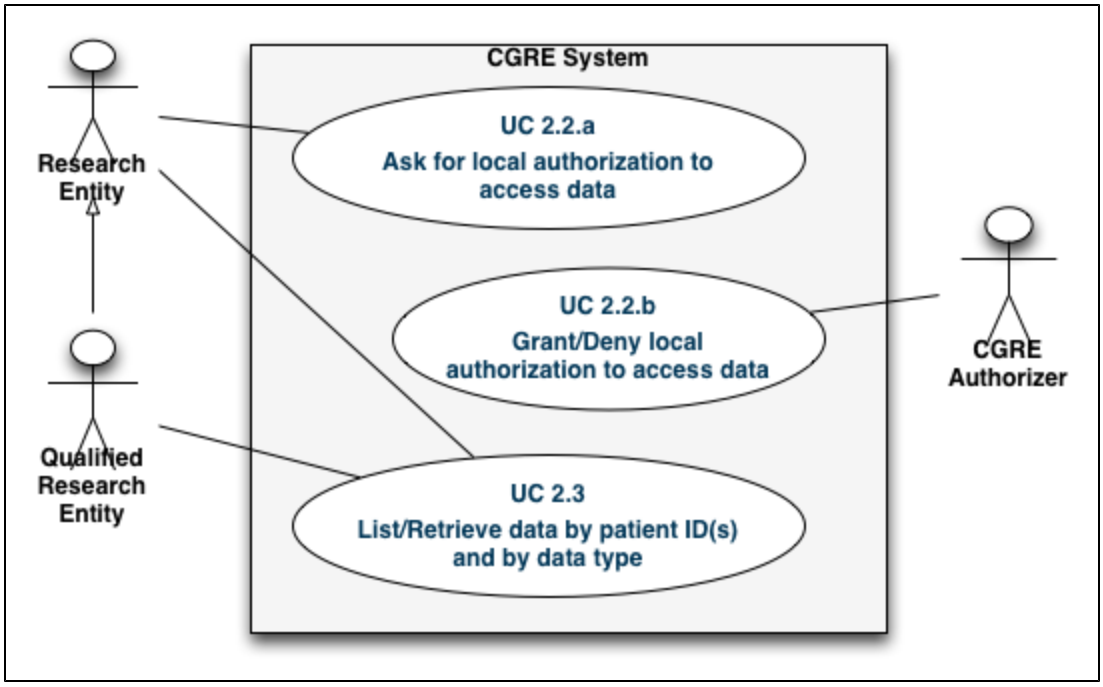
**Restful Service API**
URI: ${BASE-URI}/patient/${PGUID}/
Protocol: https
Operation: POST
Consumes: multipart/mixed
Input:

- Data - BLOB
  - (Compressed or plain) binary or text contents of a data file
- MetaData <MAP<STRING,STRING>
  - Key (category) - STRING
    - length <= 50 characters
  - Value - STRING
    - length < 256 characters

Produces: N/A
Output: none

# PLC Research Entity and CGRE Use Cases

## PLC Research Entity and CGRE Use Cases




### Nonfunctional Requirements

The CGRE will capture data with potentially distinct requirements on who they can be shared with. Patients may choose to share the data with everyone or limit to those who agree to certain terms. Hence there is a distinction between qualified researcher entities and the rest. Qualification occurs via an approval process that requires human intervention. This role is referred to as CGRE authorizer.
In the use cases below the assumption is that even the listing of summary data is limited to the shared data that the requesting RE is privileged (qualified) to access.

### UC 2.2.a Ask for local authorization to access data

#### User Stories

- As a research Entity I want to qualify to access all data, e.g. data that require me to be approved (qualified).

#### Pre Condition

- A CGRE ready for business.

#### Workflow

1. RE sends request containing contact information (including email) and further details TBD to CGRE System
2. System stores the request in the CGRE Authorizer processing queue
3. System sends confirmation of request received to email specified in the request

#### Post Condition

- RE entry is created in the CGRE system with a status indicating unprocessed qualification request
- qualification request is in CGRE Authorizer processing queue
- potentially a notification is sent to the CGRE Authorizer
- email is send to the requesting RE

#### Restful Service API

URI: ${BASE-URI}/qualification_request/
Protocol: https
Operation: POST
Consumes: json
Input:

- ResearchEntity
  - name - STRING
    - designation of the organisation, institution or person
    - length < 256 characters
  - telephone - STRING
    - includes country code, area code, phone nr and extension

        - length <= 40 characters
    - email - STRING
        - email constraints
        - length <= 40 characters
    - url - STRING
        - well formed URL constraints (http or https)
        - length < 256 characters
    - description - CLOB
        - Information about Research Entity and research intentions

Produces: plain/text
Output:

- confirmation - STRING
    - message containing conformation of request and further instructions, CGRE contact information
    - length < 256 characters

### UC 2.2.b Approve or deny qualification request

**User Stories**

- As a CGRE Authorizer I want to be notified of new requests from Research Entity for qualification, I want to be able to review the request and either approve or deny the request.
- As a researcher I want to be informed whether my my request for qualification has been approved or denied.

**Pre Condition**

- RE entry was created in the CGRE system with a status indicating unprocessed qualification request
- CGRE Authorizer was notified of the pending request

**Workflow**

1. CGRE Authorizer reviews request by recalling the entry of the requesting RE
2. CGRE Authorizer researches the provided information, verifies and communicated with RE to come to decision
3. CGRE Authorizer changes the status of the requesting RE in the CGRE System
    a. conceivable states are: denied, qualified, further information required, blocked for all requests
4. CGRE System generates an email notifying the requesting RE of the decision
    a. if RE is found to be qualified a certificate enabling qualified data access will be issued and attached

**Post Condition**

- RE entry status in CGRE system has been updated reflecting decision of CGRE Authorizer
- RE has been notified of decision of CGRE Authorizer via email
    - including the delivery of certificate token if qualified
    - including he details of request for further information if required by the CGRE Authorizer

**Restful Service API**

- There is no restful API yet. The state changes are considered CGRE System internal.

### UC 2.3.a List shared data

**User Stories**

- As a Research Entity I want to see a summary of all the shared data offered by the CGRE. The summary should include information on data file counts, types and sizes.
- As a Research Entity I want to see a summary of a keyword filtered subset of the shared data offered by the CGRE. The summary should include information on data file counts, types and sizes.
- As a Research Entity I want to see a summary of the shared data offered by the CGRE for a particular PGUID or a list of PGUIDs. The summary should include information on data file counts, types and sizes.

**Pre Condition**

- Summary information on protected data require RE to present certification token (certifying it to be a QRE)

**Workflow**

1. RE requests summary information on shared data
    a. request for qualified data have to contain the certificate token
    b. request is for summary information on **all** shared data
    c. request is for summary information on a subset of the shared data as specified by the filter parameters
2. CGRE System queries the stored shared data and calculates the summary information
3. CGRE System sends the summary information back to the requesting RE

**Post Condition**

- the summary information is sent back to the requesting RE

**Restful Service API**

URI: ${BASE-URI}/shared/summary
Protocol: https
Operation: GET

*Request for summary on all data*
Consumes: N/A
Input: none

*Request for summary on a subset of the data as determined by a filter*
Consumes: json
Input:

- Filter (all criteria are applied combined via AND semantics, NULL fields are ignored)
    - pguids - SET of STRINGs
        - PGUID STRING length <= 64 characters
        - limit summary to shared data on patients with the specified PGUIDs
    - tags - SET of STRINGs
        - tag STRING length < 256 characters
        - limit summary to shared data associated with the specified tags
    - lastChangeDate - DATETIME
        - limit summary to shared data on records that were added or updated on or after the specified

Produces: json
Output:

- Summary (for requests from unqualified RE only unrestricted data are summarized)
    - totalFileCount - INTEGER
        - number of all shared data files available
    - filteredFileCount - INTEGER
        - number of all shared data files available after application of the filter
    - totalPGUIDCount - INTEGER
        - number of all patients (PGUIDs) the repository has data files about
    - filteredPGUIDCount - INTEGER
        - number of all patients (PGUIDs) the repository has data files about - after application of the filter
    - totalFileSize - INTEGER
        - sum of all individual file sizes in the repository
    - totalFileSizeUnit - ENUM [PLC:B, KB, MB, GB, TB]
    - filteredFileSize - INTEGER
        - sum of all individual file sizes in the repository after application of the filter
    - filteredFileSizeUnit - ENUM [PLC:B, KB, MB, GB, TB]
    - fileCountPerTag - MAP<STRING, INTEGER>
        - key: tag | value: count of shared data files associated with the key tag
    - filteredFileCountPerTag - MAP<STRING, INTEGER>
        - key: tag | value: count of shared data files associated with the key tag for files left after application of the filter

### UC 2.3.b Retrieve shared data

**User Stories**

- As a Research Entity I want to retrieve shared data captured by the CGRE. I want to bei able to download either all shared data for a subset of the shared data depending on filters based on one or more PGUIDs and or keywords and or types.
- As a researcher I want to be able to download only the data that have changed (new or updated) since a particular data and time.

**Pre Condition**

- Request for and Download of protected data requires RE to present certification token (certifying it to be a QRE)

**Workflow**

1. RE requests download for shared data
    a. request for qualified data have to contain the certificate token
    b. request is for download of **all** shared data
    c. request is for download of a subset of the shared data as specified by the filter parameters
2. CGRE System queries the stored shared data
3. CGRE System compresses all requested files + metadata into a single file
4. CGRE System stores the compressed data file in a download location for predetermined period of time
    a. it is conceivable that for typical requests such compressed are already prepared and updated when needed
5. CGRE System sends a download URL back to the requesting RE
    a. this will allow the offload of large file transfer to a different more appropriate channel rather than perusing the messaging channel

b. the download URL will have an appropriate expiration time
c. access to the download URL will require the certificate token if the shared data is limited to qualified RE only

**Post Condition**

- the compressed data file is stored in a download location
- the download URL is sent back to the requesting RE
- the compressed data file is deleted after expiration of the storage time

**Restful Service API**

URI: ${BASE-URI}/shared/
Protocol: https
Operation: GET

*Request for download of all data*
Consumes: N/A
Input: none

*Request for download of a subset of the data as determined by a filter*
Consumes: json
Input:

- Filter (all criteria are applied combined via AND semantics, NULL fields are ignored)
  - pguids - SET of STRINGs
    - PGUID STRING length <= 64 characters
    - limit download to shared data on patients with the specified PGUIDs
  - tags - SET of STRINGs
    - tag STRING length < 256 characters
    - limit download to shared data associated with the specified tags
  - lastChangeDate - DATETIME
    - limit download to shared data on records that were added or updated on or after the specified

Produces: json
Output:

- DownloadDetails
  - size - INTEGER
    - size of the download
  - sizeUnit - ENUM [PLC:B, KB, MB, GB, TB]
  - url - STRING
    - URL providing access to the requested data
    - well-formed URL constraints
    - length < 256
  - expirationTime - DATETIME
    - Date and time when the download expires and the URL is no longer valid