Portal Mirror Specification

Introduction

ICGC's members submit clinical and downstream analysis files (e.g. mutation calls) to the DCC using its submission system. Raw data (BAM files) is submitted to the European Genome Archives EGA. The DCC is not storing raw data. The ICGC portal's main roles are to browse downstream analysis results and to index files for search.

A mirror site is an exact replica of the original site and is usually updated frequently to ensure that it reflects the content of the original site. Mirror sites are used to make access faster when the original site may be geographically distant (for example, a much-used Web site in Germany may arrange to have a mirror site in the United States). In some cases, the original site (for example, on a small university server) may not have a high-speed connection to the Internet and may arrange for a mirror site at a larger site with higher-speed connection and perhaps closer proximity to a large audience.

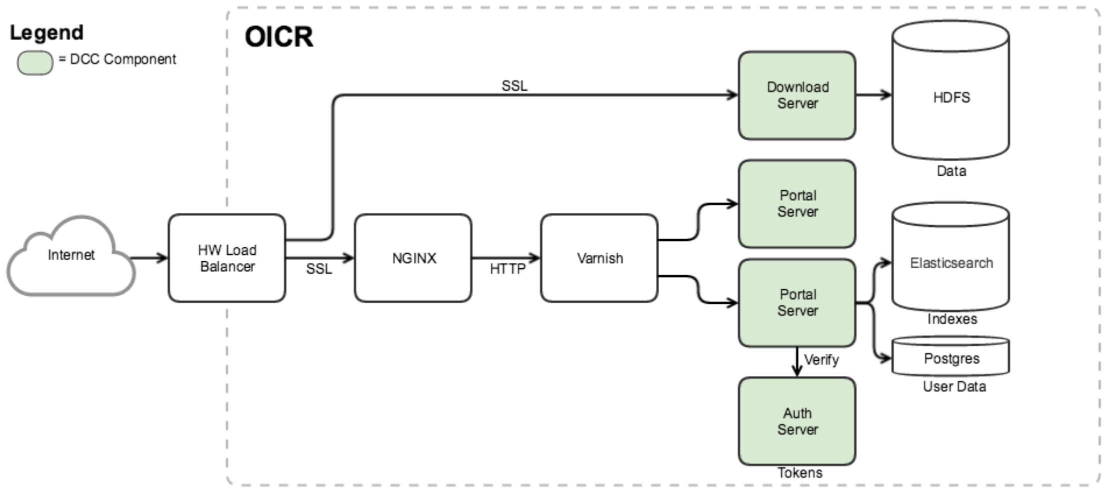
The purpose of this document is to describe the infrastructure and processes required to create a mirror of the ICGC DCC Portal.

Architecture

The portal architecture is comprised of 3 tiers:

- Web (HW Load Balancer, NGIX, Varnish)
- Application (Portal Server, Download Server)
- Data (Postgres, Elasticsearch, HDFS)

Traffic enters from the internet by way of a hardware load balancer which provides basic networking functions (VIP, banning, monitoring etc.). Traffic is then routed to a web server (NGINX) which provides SSL termination. Next, requests are sent in clear text to a cache server (Varnish) to accelerate serving previously requested pages. It is important to note that Varnish can only accept non-SSL traffic, hence the need for NGINX to provide the SSL termination. The cache server then round-robins the requests between two backend Portal Servers. These application servers are responsible for serving clinical / analysis information and repository indexes from Elasticsearch. A small amount of user data is stored in Postgres to allow users to save "entity sets" of interest. Large data archives are served from HDFS by the Download Server.



Data

Release

Every 3-4 months the DCC produces a data release (e.g. Release 21) which supplies the main data published by the Portal. It includes data provided by submitters during that release cycle as well as a number of reference data sources (e.g. GO, Gene model, Reactome, Cancer Gene Census, etc.). This data is captured in an "Release Index" which powers entity pages and Advanced Search and a set of "Release Data Archives" which backs the releases page and the download of individual donor data.

Repository

Every night the DCC indexes various ICGC data repositories in order to present a unified view of the available files for external download. This information is stored in a so-called "Repository Index" in Elasticsearch.

Mirroring

Hardware

The following table shows the minimum requirements needed to run a Portal environment

Minimum Requirements				
Component	**#**	**Resources**	**Required**	**Notes**
Portal Server	1	4G RAM	Yes	
Download Server	1	8G RAM, 500G Disk	Yes*	Needed for experimental files
Varnish Cache Server	1	16G RAM	No	
Elasticsearch Nodes	6*	32G RAM, 100G Disk	Yes	Data dependent (# ssms, donors)
Postgres	1	4G RAM	Yes	

The following table shows what is actually used by the ICGC DCC:

ICGC DCC			
Component	**#**	**Resources**	**Notes**
Portal Server	2	4G RAM	
Download Server	1	8G RAM	
Varnish Cache Server	1	128G RAM	
Elasticsearch Nodes	15	128G RAM, 4TB Disk	
Postgres	1	4G RAM	
HDFS Nodes	32	128G RAM, 4TB Disk	Stores experimental files
Nginx Server	1		
HW Load Balancer	1		

Strictly speaking a Varnish cache server is not required but it significantly helps in speeding up page load times and taking pressure off of the backend Elasticseach cluster. As the size of cluster decreases, the more important the cache's role becomes.

Additionally, HDFS is not required. It may be replaced with local disk (preferrable SSD) or NFS. Note however that HDFS is prefered due to its distributed nature and HA capabilities. It is also part of the ETL infrastructure (not required here) used to create the data artifacts.

Networking

In order to serve over HTTPS a valid CA SSL certificate is required. Furthermore, the certificate must be on a DNS subdomain of icgc.org for SSO login to function propertly (see below).

In order for data synchronization and authenication to function propertly, the target environment must be able to access the ICGC DCC Portal.

Software

Application

The portal software can either be built from source:

- https://github.com/icgc-dcc/dcc-portal
- https://github.com/icgc-dcc/dcc-download

or downloaded via Artifactory:

- https://artifacts.oicr.on.ca/artifactory/list/dcc-release/org/icgc/dcc/dcc-portal/
- https://artifacts.oicr.on.ca/artifactory/list/dcc-release/org/icgc/dcc/dcc-download/

Boths systems require Oracle Java 8

Third Party

The following is a list of 3rd party software required to run and mirror the Portal:

Name	Version
Elasticsearch	1.4.4
Knapsack Plugin	1.4.4.1
Postgres	9.4
HDFS	CDH 5.2.1

Data Synchronization

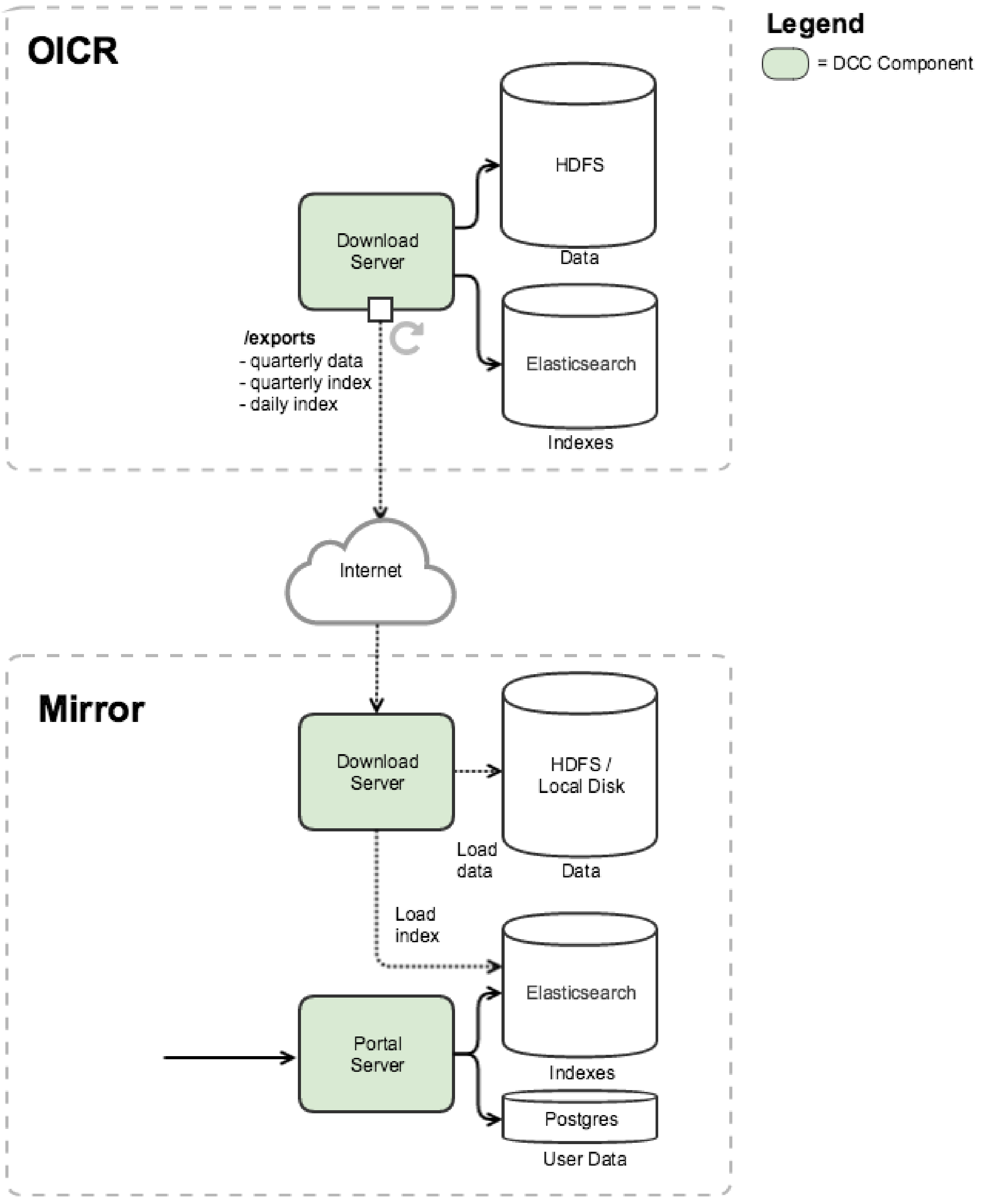
The DCC will create a collection of data exports exposed via our Download Server's /exports endpoint. These exports include:

- A quarterly "Release Index"
- A quarterly "Data Release Archive"
- A daily "Repository Index"

To effectively mirror the data, a mirror site must:

- Download the quarterly "Release Index" when it is made available (quarterly)
  - Use the Elasticsearch Knapsack plugin to load the "Release Index"
- Download the quarterly "Data Release Archive" when it is made available (same time as above)
  - Untar and make available to Download Server via configuration
- Download the daily "Repository Index" every night, after 1:00 AM EST.
  - Use the Elasticsearch Knapsack plugin to load the "Repository Index"
- Flush Varnish cache (if used) after every index load

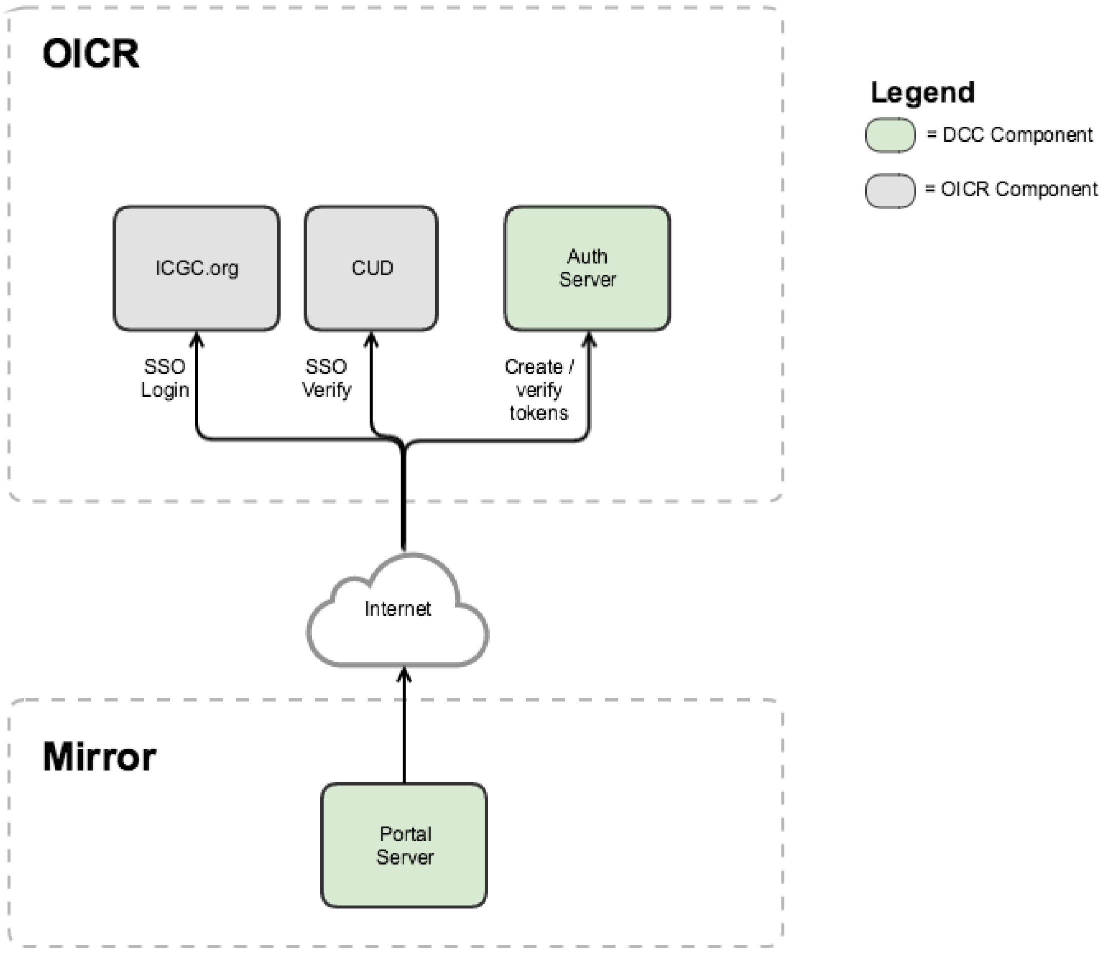
A diagram of the process is presented below:



Authentication and Authorization

In ICGC, the authentication is based on a successful login to ICGC.org or via OpenID. For ICGC.org, SSO is used (see networking section for requirements). When OpenID is used, the OpenID must be associated with an ICGC.org account. In either case, authorization is granted based on DACO access which is supplied by an internal API.

For a mirror site, the site would need access to these resources (see Networking section for requirements):


OICR
ICGC.org
CUD
Auth Server
SSO Login
SSO Verify
Create / verify tokens
Internet
Mirror
Portal Server
Legend
= DCC Component
= OICR Component