SOFTWARE DESIGN DOCUMENT

SciApps.org

DOCUMENT CHANGE HISTORY

Version Number	Date	Description
1.0	February 2020	Modified first version

SciApps.org

Software Design Document

Version 1.0

TABLE OF CONTENTS

SOFTWARE DESIGN DOCUMENT ..... **I**
1. INTRODUCTION **1**
1..1 Purpose *1*
1..2 Platform Overview *1*
1..3 Glossary *2*
1..4 References *2*
1..5 Document Overview *2*
2. PLATFORM OVERVIEW **3**
2..1 Motivations *3*
2..2 Technologies Used *3*
2..3 Additional constraints, programming language and tools being employed *3*
3. SYSTEM ARCHITECTURE **4**
3..1 User Interface *4*
3..2 Application Programming Interface (API) *5*
3..3 Database Architecture *6*
3..4 Workflow Engine *6*
3..5 User Jobs and Workflows *7*
4. USE SCIAPPS FOR BIOINFORMATICS ANALYSIS **8**
4..1 Authentication *8*
4..2 Doing Analysis *8*

Table 1: Glossary terms **2**
Table 2: SciApps release 1.0 RESTful API **5**

Figure 1: Overview of the SciApps architecture **1**
Figure 2: SciApps web interface **4**
Figure 3: Database schema **6**
Figure 4: SciApps Workflow Engine **7**

1. INTRODUCTION

1..1 Purpose

The purpose of this Software Design Document (SDD) is explaining the SciApps.org platform. SciApps.org is designed to provide a cloud-based 'ready-to-compute' platform with bioinformatics workflows.

1..2 Platform Overview

As shown in Figure 1, the SciApps platform is powered by the CyVerse Data Store (for cloud storage), the Texas Advanced Computing Center (TACC for cloud computing), and a local cluster at Cold Spring Harbor Laboratory (CSHL, for providing more computational power).

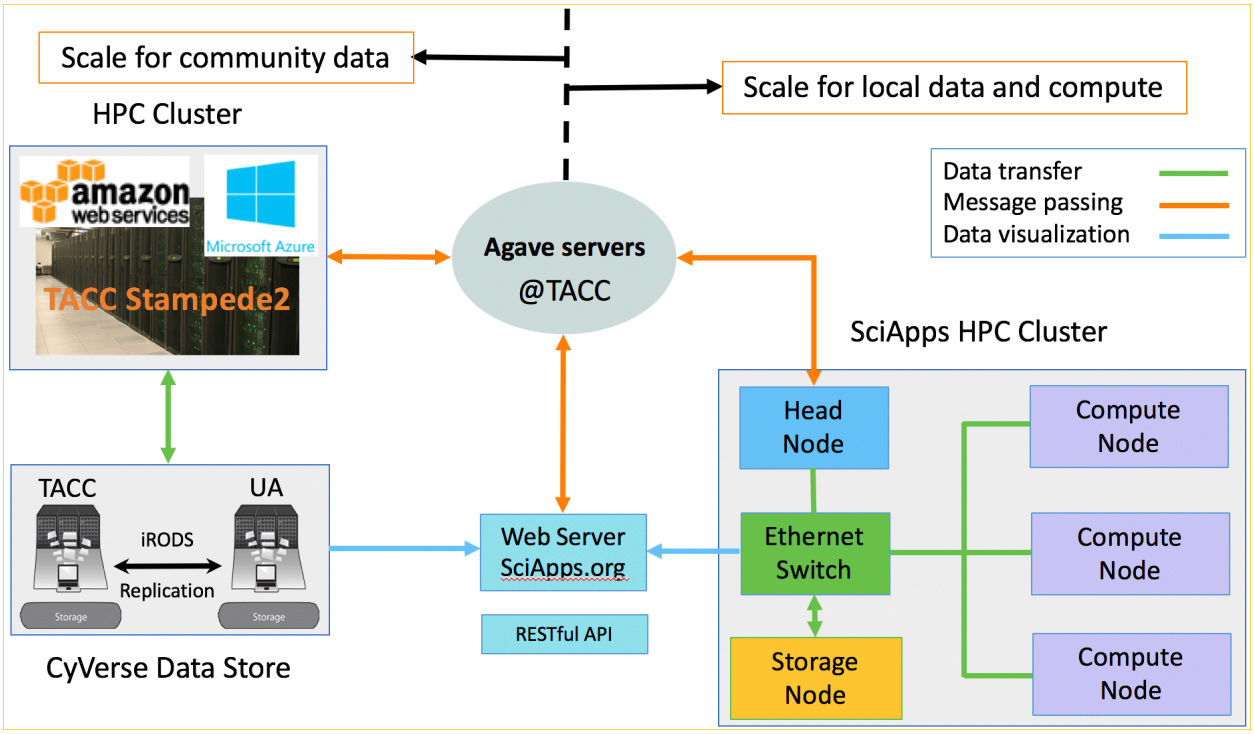

Figure 1: Overview of the SciApps architecture

SciApps uses the Agave Science API (http://agaveapi.co/) to manage the entire cycle of analysis jobs between TACC's HPC and the CyVerse Data Store. SciApps is hosted on a web server at CSHL for providing a RESTful API and a graphical user interface for job submission, workflow creation, and management of both jobs and workflows. SciApps can run on the TACC and CyVerse cloud without the optional local cluster if additional resources are not needed. Users need to have CyVerse accounts (which is free) to authenticate to the cloud system.

1..3 Glossary

Table 1: Glossary terms

Term	Definition
CyVerse Data Store	Based on a technology called iRODS, the Data Store gives you great flexibility and control over your data, from web services to mountable file systems to high-speed command-line transfers.
CyVerse	Funded by the NSF to provide life scientists with powerful computational infrastructure to handle huge datasets and complex analyses, thus enabling data-driven discovery.
Agave	An open-source, science-as-a-service API platform for bringing together your public, private, and shared high-performance computing (HPC), high throughput computing (HTC), Cloud, and Big Data resources under a single, web-friendly RESTful API.
User	The person who wants to perform data analysis on the SciApps platform and have a CyVerse account (currently with over **70,000** users).
RESTful API	An application program interface (**API**) that uses HTTP requests to GET, PUT, POST and DELETE data.
TACC	The Texas Advanced Computing Center (TACC) designs and operates some of the world's most powerful computing resources.

1..4 References

[1] Wang L, Lu Z, Van Buren P, Ware D. SciApps: a cloud-based platform for reproducible bioinformatics workflows. Bioinformatics. 2018 Nov 15;34(22):3917-20. Link.

[2] Wang L, Van Buren P, Ware D. Architecting a distributed bioinformatics platform with iRODS and iPlant Agave API. In2015 International Conference on Computational Science and Computational Intelligence (CSCI) 2015 Dec 7 (pp. 420-423). IEEE. Link.

[3] SciApps platform guide: Link.

1..5 Document Overview

The rest of this document is organized as follows: Section 2 is written to provide a high-level overview of the platform. Section 3 describes how the platform is implemented. Finally, section 4 describes how to use the platform for data analysis.

2. PLATFORM OVERVIEW

2..1 Motivations

SciApps is initially developed to provide a Graphical User Interface to the CyVerse federation system (the local cluster) hosted at CSHL. Later it is extended to support automatic analysis workflows over TACC and CyVerse Data Store, which, currently, is not supported by the CyVerse project. The SciApps platform has been further enhanced to support projects like MaizeCODE for data management, analysis, and distribution.

2..2 Technologies Used

The backend of SciApps was built using Perl and the MySQL, and the front-end was built with React, an open-source JavaScript library. The graphic workflow diagram is built with the mermaid package (https://knsv.github.io/mermaid/) and modified for interactivity on metadata and real-time job status. The latter is acquired through Agave API's Webhook notification for jobs, which is also used for automatic updating of the MySQL database and automated execution of a workflow. Regarding genome browsers, JBrowse is supported for visualizing alignments, variants, and genome annotation results, etc.

2..3 Additional constraints, programming language, and tools being employed

- JavaScript is the programming language being used to allow platform independence.
- A Perl Dancer web framework is used in the backend.
- A MySQL relational database backend is required for most functionalities.
- A CyVerse user account is needed
- Configuration of iRODS on the webserver is required for browsing CyVerse Data Store.
- Configuration of Agave on the webserver is required for accessing CyVerse Data Store and TACC HPCs.
- An SSL certificate is needed for the webserver to support authentication to the cloud systems.

3. SYSTEM ARCHITECTURE

3..1 User Interface

The interface, in which users perform data analyses and build workflows, is designed to have four areas: the navigation bar, containing project data (e.g. MaizeCODE), workflow functionalities (building, loading, public and private workflows), tools (JBrowse and the API's swagger interface), help (link to the platform guide), and login for CyVerse authentication; the app panel (left column) for categorized apps (with the **Mapping** category clicked and expanded); the main panel (middle column) for app form(s) or workflow builder form; and the history panel (right column) for job name followed by four icons: checkbox for building a workflow from executed jobs, job history (*i*), job re-launch, and visualization of results.

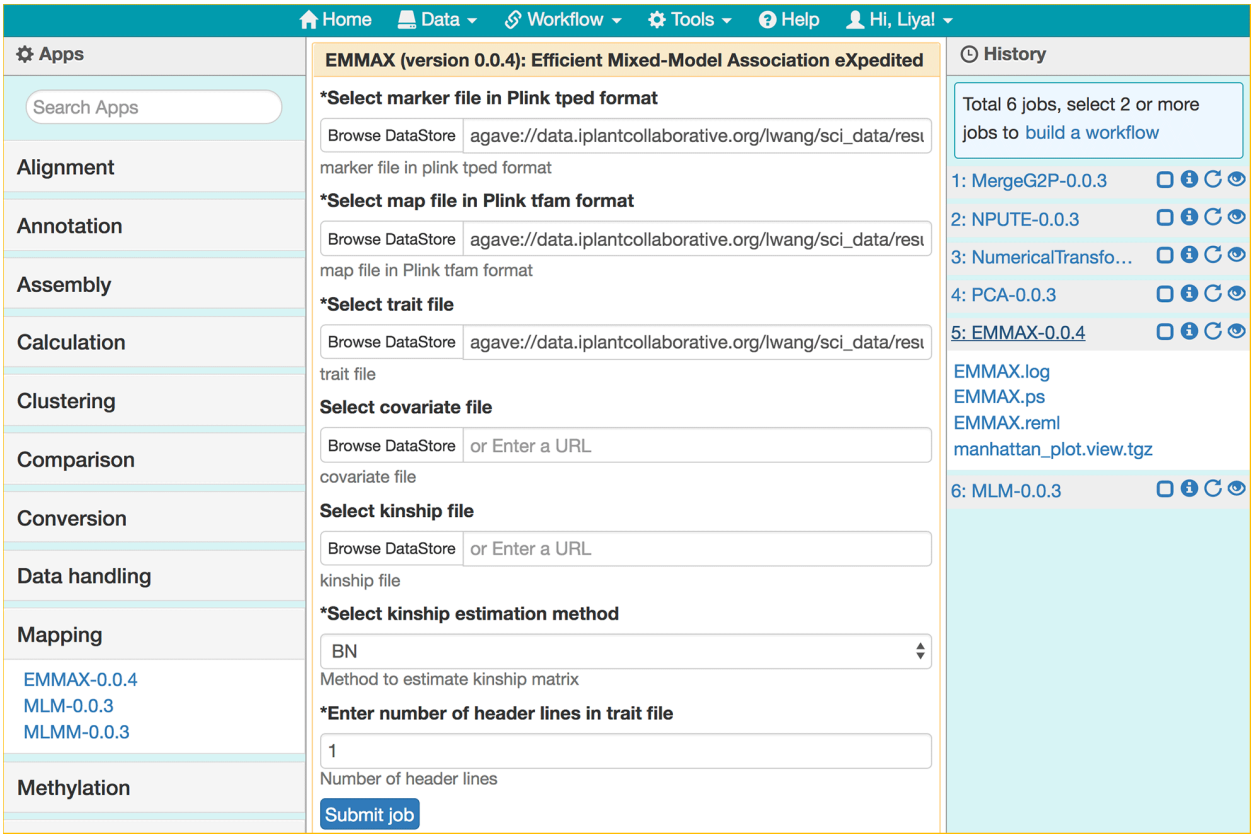

Figure 2: The SciApps web interface

Here, a six-step association workflow is loaded in the history panel, the app form of the fifth step is re-loaded in the main panel with inputs and parameters (used for the analysis), and the results from Step 5 are clicked and expanded in the history panel, in which the manhattan_plot.view.tgz file can be visualized with an interactive Shiny app (using the 'eye-shaped' visualization icon).

3..2 Application Programming Interface (API)

RESTful API is designed to automate the management of metadata and analysis for projects like MaizeCODE and provide access to data and results, with the currently available endpoints listed in Table 2.

Table 2: SciApps release 1.0 RESTful API

Endpoint	Method	Description
/job	GET	List all jobs
/job/new/{id}	POST	Run a new job
/workflow/build	POST	Build a workflow from jobs
/workflowJob/new	POST	Generate a workflow JSON
/workflow/new	POST	Save a new workflow
/job/{id}	GET	Return the job JSON
/job/{id}/delete	GET	Delete the job
/workflowJob/run/{id}	GET	Run a new workflow
/workflow/{id}/metadata	GET	Get the workflow metadata
/workflow/{id}/update	POST	Update the workflow with metadata etc
/workflow	GET	List all workflows
/apps/{id}	GET	Return the application JSON
/workflow/{id}/delete	GET	Delete the workflow
/workflow/{id}	GET	Return the workflow JSON
/apps	GET	List all integrated apps

Two examples are provided below for accessing metadata of a workflow/analysis through API.

- Retrieving computational metadata of a workflow
  - https://sciapps.org/workflow/a14ff622-7af9-4b1f-877a-2be926dc1059
- Retrieving experimental metadata of a workflow
  - https://sciapps.org/workflow/a14ff622-7af9-4b1f-877a-2be926dc1059/metadata

3..3 Database Architecture

A relational database backend is required for SciApps to function. Freely available open-source model relational database MySQL is being used by SciApps.

The relational database retains all data entered through the web interface and network paths to data and associated files. The major components (tables) of the SciApps database are shown in Figure 3, including the user-workflow table for connecting users with their analysis workflows, the user table for storing basic user information after authenticating with CyVerse, the workflow table for storing workflow information, the job table for recording each analysis job in the workflow, and the next-step table for powering the workflow engine (see next sub-section for more details).

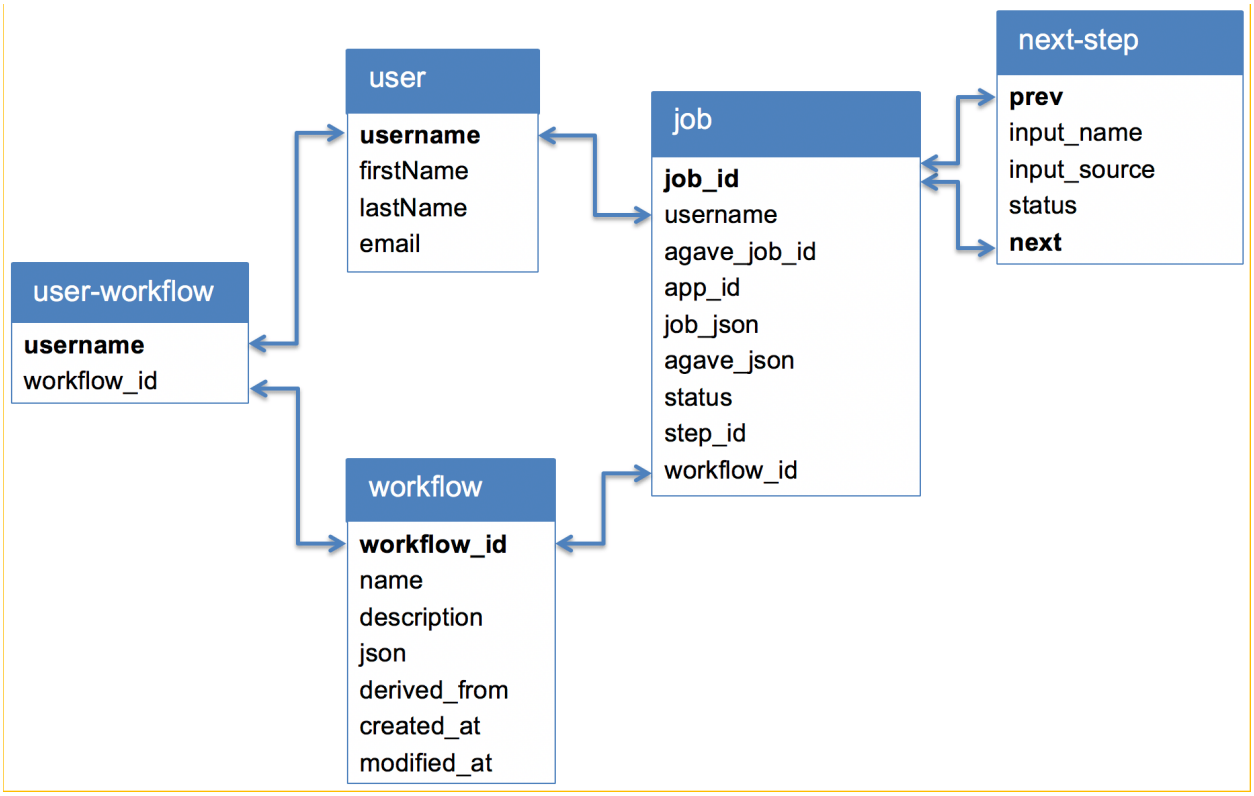

Figure 3: The database schema

3..4 Workflow Engine

The SciApps workflow engine uses the webhook notification service from the Agave API to automate the complex analysis, with the basic logic shown in Figure 4. Generally, when SciApps receives a notification that a job is completed, it checks all remaining jobs of all running workflows and makes decisions on whether any jobs are ready to be submitted (if dependency is clear) to run. If a new job is indeed submitted, both the web interface (e.g., the job status in the History panel and the workflow diagram) and the SciApps database will be then updated.

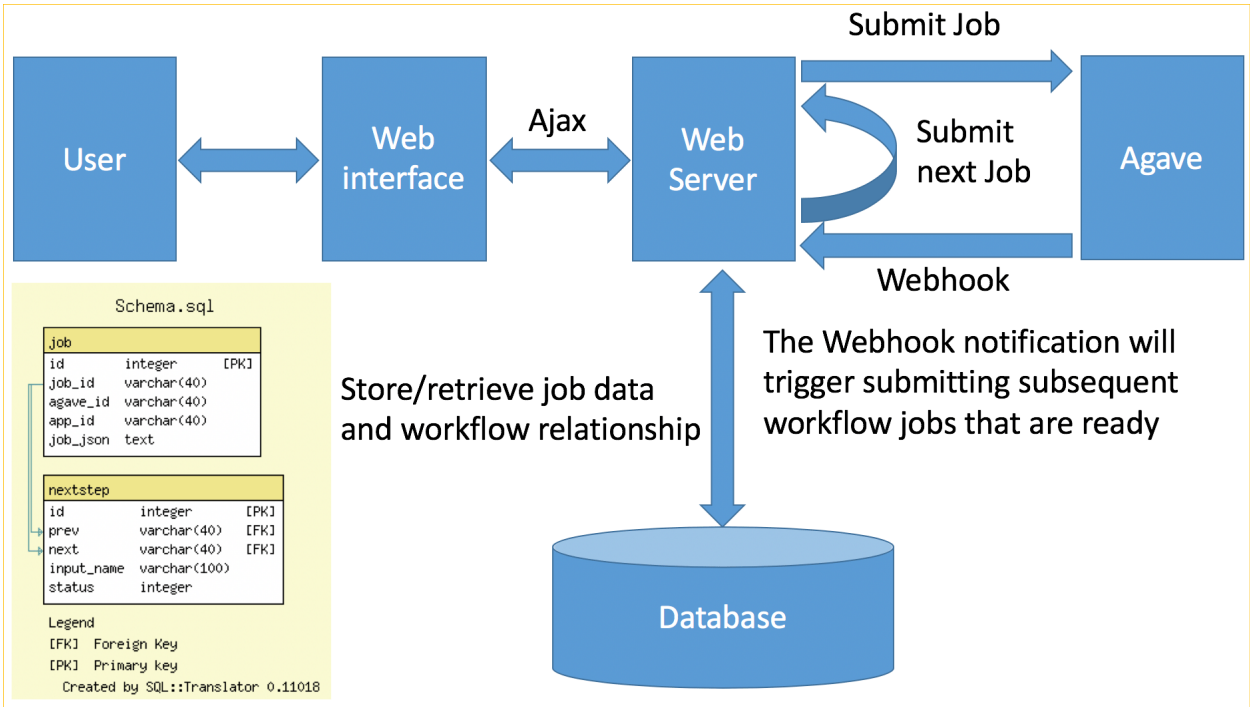

Figure 4: SciApps Workflow Engine

3..5 User Jobs and Workflows

As shown in Figure 3, workflows and jobs are associated with users. SciApps enforces that no job can be deleted by users from the database, itself, though the user may use the web interface "delete" function to delete jobs or workflows from the list of viewed jobs. In this way, all jobs are retrievable for shared users to maintain existing workflows or build new workflows.

As demonstrated in the platform guide, users can load jobs into the History panel for examining outputs or using these jobs to build new workflows. Users can also load all jobs of a workflow into the History panel to build new workflows with a subset of the jobs. All workflows associated with a user can be shared with others through a unique workflow ID.

4. USE SCIAPPS FOR BIOINFORMATICS ANALYSIS

4..1 Authentication

When users log into SciApps.org, they will be directed to CyVerse for authentication. Users must enable the 'SciApps' service in their CyVerse User Portal (https://user.cyverse.org/), as shown in Figure 5. A 'sci_data' folder will be created under the user's CyVerse root folder for SciApps to access all data in the folder and archive analysis results into it.

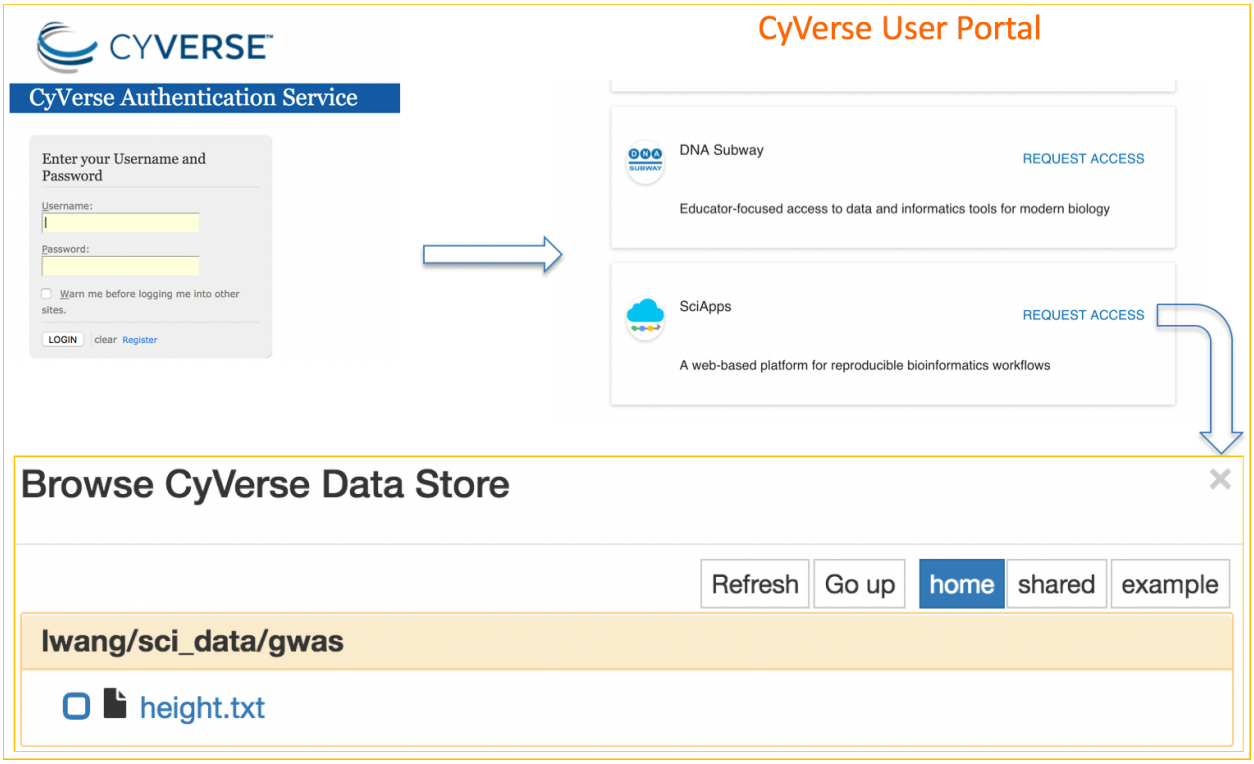

Figure 5: Authentication Flow

4..2 Doing Analysis

Before processing their data with SciApps, users need to use iCommands (command line), CyberDuck (GUI), or CyVerse Discovery Environment (GUI for small files < 2GB) to upload data files into their **sci_data** folder. A **results** folder will be created inside the **sci_data** folder when users run their first analysis job on SciApps.org. Detailed analysis examples can be found in the platform guide.