

## **Scoping Document for DynaFed project**

### **DynaFed background**

DynaFed was created as a way to present a distributed repository as one (federating the data) without needing to explicitly keep an index of all the files it contains (dynamic). It provides an easy access point for data contained in multiple locations across the world and it is already in use in places such as Belle II and the CMS@Home project.

DynaFed can provide a single namespace for a VO that allows the user to access data from the site geographically closest to them automatically. It also provides an authentication and authorisation layer that is flexible and can utilise Apache modules for authentication (thus giving easy access to authenticating with technologies such as LDAP) and a highly customisable Python plugin can handle authorisation.

DynaFed also allows for use of commercial storage solutions such as Amazon S3 to augment the existing storage network alongside dedicated storage systems run by the WLCG and related partners. DynaFed can manage multiple storage locations via the use of plugins, which means that as new technologies are developed they can easily be incorporated into an existing DynaFed instance by developing a new plugin.

### **Desired Outcomes**

Overall, this project aims to improve the usability of DynaFed such that it becomes a more attractive option to all users.

As it is, all instances of DynaFed have been set up using X509 as the authentication and authorisation mechanism. This works for CERN VOs, but others use more recent access control mechanisms and showing that DynaFed can support these mechanisms greatly increases the usability of DynaFed. By creating additional authentication/authorisation plugins for DynaFed, it becomes more attractive for users to adopt as it will give them more flexibility in how they allow users access to their data.

Additionally, since DynaFed was initially created as top layer to existing storage mechanisms, more focus has been put into reading from DynaFed than writing to it. Therefore, improving the write performance would make DynaFed a more complete storage solution, as currently it writes with undocumented behaviour. Initial investigations have shown that it tries to write to the closest location(s) geographically, but it is unknown how it determines whether to write to multiple locations, and what file paths are valid for each location.

### **Stakeholders e.g. can we identify local (Facilities) users we can involve in testing prototypes before engaging with external users (such as Ligo VO)**

The CCP4 group have been identified as a potential on-site interested party who could be utilised as a more local user group for quicker feedback before focusing on LIGO.

### **Technical challenges - what tools we have to work with**

DynaFed plugins are written in C, but they have created a Python layer on top of the authorisation plugin so the authorisation plugin would be written in Python. It will require knowledge of LDAP and related authentication technologies.

The location plugins would be written in C. We would be modifying existing location plugins (e.g. S3) so it would involve improving existing code.

**Resources Available**

**(e.g Graduate placement 6 mo. duration, people with experience of CVMFS and DynaFed)**

The project has been assigned to a Scientific Computing graduate for a 6 month duration. They have previous skills in both Python and C from their previous projects, and have some familiarity with LDAP.

We also have a working production DynaFed instance, which will be a useful reference resource, as is the previous knowledge of the person who set up this instance.

**Milestones**

As part of creating the plugins, a test set up of DynaFed will be required so that the plugin configuration can be changed without disrupting the production server.

There are two main code project outputs, two authorisation plugins for DynaFed and an improved data writing mechanism in the location plugins.

The authorisation plugin aims to provide an alternative to the existing X.509 based authentication system currently used. By aiming towards providing LDAP access, since this is a widely used technology, multiple sites would benefit from this being implemented. This would simplify the amount of set up required for end users to be able to access their data.

Initially, the project will try to set up authentication against Federal IDs, with the end user in mind being the CCP4 group.

After a plugin for Federal IDs has been created, a plugin for LIGO will be next, using expertise gained by implementing it for Federal IDs first. LIGO may require using Shibboleth as the authentication mechanism.

After the authentication/authorisation plugins have been completed, next is improving write performance on the location plugins.

As the project focuses on improving the usability of DynaFed, if the project has time then several potential improvements could be made to the web UI.

Additionally, we aim to submit a paper to the IEEE eScience 2018 conference.

**Learning and Development needed?**

There are no major learning requirements other than an allowance of on-the-job time to explore and learn the technologies involved (e.g. DynaFed, Shibboleth)

## **Appendix: Milestones, Preliminary Work Breakdown, Success criteria**

- Federal ID authentication/authorization plugin
  - Set up LDAP authentication on test DynaFed instance using Apache LDAP module
  - Write Python authorisation plugin for DynaFed that enforces access policies
  - Write test(s) to verify the correctness of the plugin
  - Get user feedback on functionality and usability, e.g. from CCP4 people
- LIGO authentication/authorization plugin
  - If required, set up Shibboleth authentication on DynaFed test instance
  - Write Python authorisation plugin for DynaFed that enforces access policies
  - Write test(s) to verify the correctness of the plugin
  - Identify LIGO beta-testers – or can we get LIGO credentials ourselves?
- Improved write performance on location plugins
  - Determine current performance and set ideal target for improved performance
  - Improve existing location plugins write performance. Perhaps use config options to be able to select original write behaviour vs new behaviour we implement. Focus on S3 location plugin first as it is easiest to test.
  - Write test(s) to verify the correctness of the plugin
- Web UI improvements
  - Fix file upload mechanism in the web UI
  - Write tests to check the upload and any other functional improvements
  - Improve aesthetics of web UI, primarily to improve usability
- Submit abstract and paper to the 2018 IEEE eScience conference