Final Report for DOE Award ER25756

Recipient: University of Southern California Information Sciences Institute (ISI)

Project Title: A Center for Enabling Distributed Petascale Science

ISI PI: Carl Kesselman

1. Executive Summary

The SciDAC-funded Center for Enabling Distributed Petascale Science (CEDPS) was established to address technical challenges that arise due to the frequent geographic distribution of data producers (in particular, supercomputers and scientific instruments) and data consumers (people and computers) within the DOE laboratory system. Its goal is to produce technical innovations that meet DOE end-user needs for (a) rapid and dependable placement of large quantities of data within a distributed high-performance environment, and (b) the convenient construction of scalable science services that provide for the reliable and high-performance processing of computation and data analysis requests from many remote clients. The Center is also addressing (c) the important problem of troubleshooting these and other related ultra-high-performance distributed activities from the perspective of both performance and functionality.

This report summarizes the work accomplished during the finding period of the project.

2. Overview and Project Structure

The Center for Enabling Distributed Petascale Science is a SciDAC Center for Enabling Technologies whose mission is to provide software and middleware support to DOE applications with petascale data management requirements. The CEDPS project consisted of four partner sites, including two national laboratories (Argonne National Laboratory, Lawrence Berkeley National Laboratory) and two university partners (University of Southern California Information Sciences Institute and University of Wisconsin).

The project was organized initially organized around three basic research topics: data management, services, and troubleshooting. While different institutions had areas of specialization, it was generally the case that an institution would be engaged in research and development in more then one area. Specifically, USC/ISI made project contributions to activities in data movement services, policy based placement, troubleshooting and monitoring, and policy placed data placement. During this initial phase of the project, the CEDPS team focused on developing enabling technologies for petascale data management and building research prototypes to explore key topics such as policy-based data placement.

On the basis of feedback during a mid-project review, the CEDPS project underwent a significant restructuring in October 2009. In this second phase of CEDPS, the project's emphasis shifted to develop an integrated approach to petascale data movement and caching. In particular, the team is developing a platform called Globus.org whose goal is to provide a hosted data service for the DOE community that manages the complexity of petascale data movement. The goal of this hosted service is to reduce or eliminate the complexity for DOE applications that need secure, high performance data movement, since the application community will not need to install or maintain complicated software; rather, the Globus.org service is deployed and maintained by the CEDPS team and accessed by the application community via a simple request interface. In addition, the use of a hosted service should also provide greater sustainability and maintainability for the service, since the CEDPS team can build upon and take advantage of the scaling and robustness of the underlying cloud infrastructure, such as Amazon Web Services, on which the Globus.org service is deployed.

In parallel with the restructuring of the CEDPS project, the teams at ISI and ANL adopted an agile development project for the Globus.org service. Developers working on Globus.org are members of a

Scrum Team. Together, these developers construct a Product Backlog, which is a prioritized list of all the product requirements. The Scrum team takes on tasks from the Product Backlog and works on those items during a short, specified period of development called a Sprint. Each Sprint produces new product functionality. A Scrum Master manages the Scrum process. The team has frequent phone calls to check on progress, and at the end of each Sprint, there is a review meeting that evaluates the development of the current spring and plans for the next sprint.

Under the reorganization of the CEDPS project, ISI developers are part of the Globus.org Scrum team. They participate in weekly Scrum calls and regular Sprint planning meetings. For the second phase of the CEDPS project, the ISI team has participated in three aspects of the development of the Globus.org system: back end development, testing and data mirroring.

3. Data Placement

In the first phase of the CEDPS project, ISI work focused on the issues associated with the impact of the location of data on performance of petascale applications. It is in general, desirable to have the data collocated with the computing resources being used to analyze or process that data. However, in practice, this is not always the case, due to factors such as the need to compute on data that was produced by another institution, limitations of storage space, or the need to replicate for purposes of reliability. As a result, one is put in a position of having to determine optimal locations for storing data and for moving data to appropriate locations in advance of computation.

In the area of data replication and placement services, the team evaluated the functionality that we should provide based on our interactions with DOE users. Our initial approach was to create high-level data placement services that were configured by a high-level rules driven policy. This approach would enable issues such as project priorities, available network bandwidth, storage management policies and performance and reliability goals to drive the creation of a data placement strategy that could be executed via the data movement mechanisms that were created as part of CDEPS.

The Data Placement Service offers a policy driven end-to-end service to orchestrate the placement of large scale data sets and decide where to stage and replicate data sets to improve the performance of applications and workflows. The ISI and Wisconsin teams collaborated on a study of placement service options that evaluated the performance of a data placement service in combination with the Pegasus workflow management system. The results of this study were published in a Grid 2007 Conference paper [5]. In addition, the ISI team modified an existing Grid simulator to experiment with different data placement algorithms in conjunction with a workflow manager. We prototyped an implementation of a policy based placement system based on DROOLS, a widely used rules engine. Building on this existing rules engine gave us the advantage of providing a powerful policy language and a robust implementation platform. Our work focused on developing appropriate policy formulations for data management purposes and integrating sensing and actuating mechanisms that required to execute policy actions.

Work in policy based placement investigated two potential areas of application. First, we looked at the requirements of DOE virtual organizations, such as the high-energy physics community, to disseminate data according to policies at the Virtual Organization level, such as the tiered distribution of data produced by the LHC at CERN. For example, a DOE science collaboration might have a policy that there should be three copies of every file stored on geographically distant storage systems to provide a high level of availability. Another policy might require that as new files are added to an existing dataset, those files should automatically be copied to sites that subscribe to that dataset. We looked at whether we could use a policy engine to enforce similar policies and had some initial success, as reflected in a poster at the SC2008 conference.

A second area of increasing interest to DOE communities is the use of workflow engines to manage complex scientific workflows. We have done research to characterize realistic scientific workflows (resulting in a paper at the Workshop on Workflows in Support of Large-Scale Science WORKS08). Based on those scientific workflows, we have simulated a variety of data placement strategies that could work in conjunction with a workflow management system to improve the efficiency of execution of scientific workflows. This work showed how combining automated policy based data management with large-scale workflows could result in superior workflow performance over workflows that execute in isolation of data placement policy. A paper on this work was submitted to the DADC 2009 Workshop.

4. Data Mirroring

Based on our experience with our policy based placement prototype we came to the conclusion that in practice, DoE petascale computing projects had a much simpler and more pressing data management concern. Specifically, data mirroring is a key requirement for several petascale, distributed DOE science collaborations. For example, the Earth System Grid project will mirror hundreds of terabytes of key climate simulation data sets at several international climate data centers. The high energy physics community replicates terabyte-scale (and eventually petabyte-scale) data sets at multiple tiers to make them accessible to physicists around the world.

ESG is working on the next-generation of replica management functionality for their application domain, and the CEDPS group is participating in the design of replication services that will be used by ESG. In addition, we had extensive discussions with the STAR and SNS applications. All three application communities (ESG, STAR, and SNS) identified a need for data mirroring functionality, and their requirements will drive the ongoing implementation of data mirroring tools. Our design and development work is focused on providing this functionality.

Based on this experience and observations, we decided to reorient our work toward providing simple replication utilities rather than higher-level data placement services, because the former better met the needs and requirements of DOE application communities. In particular, we are focused on supporting the needs of the Earth System Grid, which currently was using the Globus Replica Location Service to manage location information for data sets.

In the initial phase of the CEDPS project, the ISI team worked to develop a data mirroring tool called globus-url-sync that worked in conjunction with the globus-url-copy command line tool for the GridFTP data transfer service. This tool compares a source and destination directory and identifies which files need to be copied to bring the destination directory into synchronization with the source directory. This list of files to be copied is then handed of to the globus-url-copy tool, which performs the transfers. Since the reorganization of the CEDPS project, the development of globus-url-sync continued under other funding. The tool was released in the Globus Toolkit version 5.0.1 in early 2010.

In the second phase of the CEDPS project, emphasis switched to the creation of a unified software as service platform, called Globus Online. With the transition to Globus Online (Globus.org), the ISI team refocused their data mirroring activities to de-emphasize the command line tool and to assume responsibility for integrating data mirroring functionality into the Globus.org system.

In addition, the ISI team is worked with DOE scientists from the Advanced Photon Source at ANL and the Spallation Neutron Source at ORNL to mirror data sets between their sites. We have had discussions with these teams to define their data mirroring requirements.

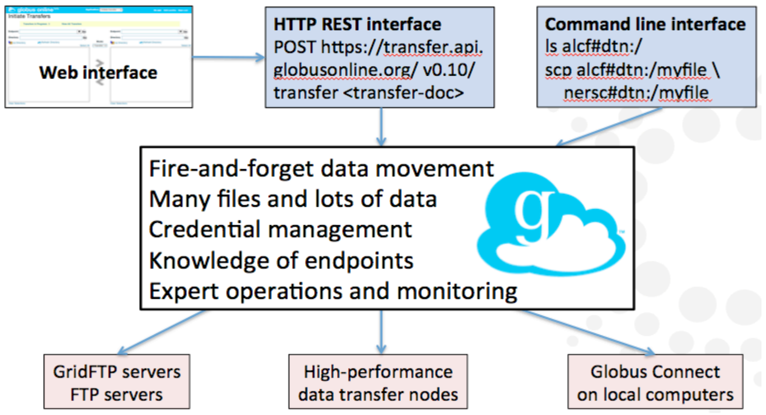
5. Globus.org

Effective use of high-speed networks for DOE research requires that we overcome the usability barriers that impede network use by non-expert users. To this end, the Center for Enabling Distributed Petascale Science (CEDPS) project launched the Globus Online project in 2009 to enable reliable high-performance research networking for the masses. The key idea is to outsource complex file movement tasks (e.g., “copy directory D from system A to system B” or “synchronize directory E and F”) to a specialized software as a service (SaaS) provider, Globus Online, that then takes responsibility for managing the end-to-end process. Data transfers are performed via GridFTP [3], which supports high-speed transport but has previously required tedious manual configuration. The Globus Online service incorporates knowledge about endpoint and network protocol configurations to reduce the expertise (and software installation) required of users. Intuitive Web 2.0 interfaces and a one-click install data movement client further simplify the user experience. The service also incorporates logic designed to optimize end-to-end performance and to recover from transient failures. The SaaS approach also permits rapid (and transparent) software upgrades and expert operator diagnosis of persistent failures.

The ISI team worked with the ANL team on the design of future capabilities of the Globus.org service. In particular, the two teams will collaborate to identify key additional functionality that should be provided in hosted services and to define the right interfaces for users. This included participation in biweekly management meetings for the project and regular scrum meetings of the development team. Under this process, developers at ISI implement functionality in the Globus.org service that corresponds to the priorities set during each Sprint of the agile development process.

6. Design, deployment, and application of Globus Online

CEDPS launched in late 2009 an experiment aimed at exploring the feasibility of using SaaS capabilities to achieve a radical simplification of research data transfer. Our goal was to construct a hosted service that would implement solutions to challenging data transfer tasks (e.g., transfer management, credential management, recovery from transient errors) while also providing modern Web 2.0 interfaces. The result is the Globus Online system.



```
POST https://transfer.api.globusonline.org/ v0.10/
transfer <transfer-doc>
```

```
ls alcf#dtn:/
scp alcf#dtn:/myfile \
   nersc#dtn:/myfile
```

Figure 1: A user view of Globus Online.

The first production-level Globus Online system, delivered in November 2010, implements methods for managing the transfer of single files, sets of files, and directories, as well as rsync-like directory synchronization. It can manage security credentials, including for transfers across multiple security domains; select transfer protocol parameters for high performance; monitor and retry transfers when there are faults; and allow users to monitor status. It provides REST, Web browser, and command line interfaces, so that the casual user can initiate and monitor a transfer from a Web browser while a frequent user can integrate Globus Online calls into their application via command line scripting or REST messages. Figure 1 presents a user view of the system. Note the scp command used to illustrate the command line interface; this command has the same syntax as the commonly used but slow secure copy, but invokes high-performance, Globus Online-optimized GridFTP transfers. The Globus Online implementation is hosted on Amazon Web Services cloud infrastructure to enable convenient state replication and elastic computing capacity. Note that data transfers do *not* proceed via Amazon; only the data transfer management logic executes there.

Fortuitously, our development of Globus Online coincided with the roll out of data transfer nodes (DTNs) [1] across DOE sites: dedicated Linux boxes, each with two 10 Gb/s network connections (to ESnet and an internal network), configured for high-speed GridFTP transfers to site parallel file systems. Globus Online allows users to refer to these systems (and any other endpoint) by simple symbolic names (e.g., alcf#dtn) rather than hostname, ports and URLs. Globus Online also dynamically load balances and fails over among multiple DTNs if a site has more than one—as is the case, for example, at DOE leadership computing facilities.

The first version of Globus Online did not solve the "last mile problem," of transferring data to and from anywhere—not just between the high-end facilities that had GridFTP servers installed. We addressed this shortcoming in April 2011 with the introduction of Globus Connect (Figure 2), a special packaging and wrapping of the GridFTP server binaries for Windows, Mac OS X, and Linux that can be trivially installed by anyone (without administrative privileges) on their own desktops and laptop, even if they are behind a firewall or Network Address Translation device that only allows out-bound connections. With a few easy steps, users can connect their computer to Globus Online, thus opening Globus Online's high-performance, easy-to-use file transfer capabilities to many more users and uses.

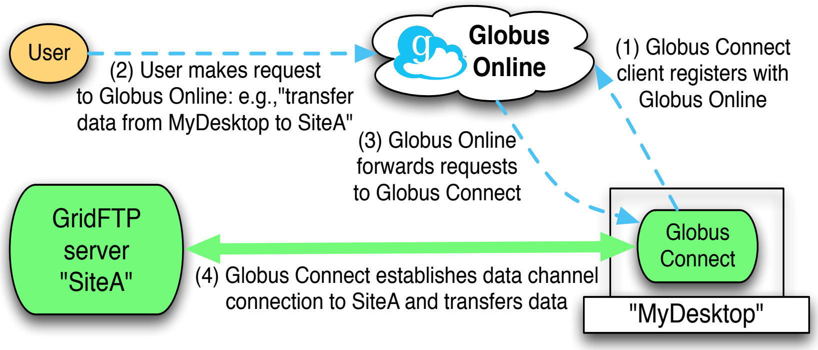

User
(2) User makes request to Globus Online: e.g.,"transfer data from MyDesktop to SiteA"
Globus Online
(1) Globus Connect client registers with Globus Online
(3) Globus Online forwards requests to Globus Connect
GridFTP server "SiteA"
(4) Globus Connect establishes data channel connection to SiteA and transfers data
Globus Connect
"MyDesktop"

Figure 2: Schematic of the Globus Connect client

Performance results were also encouraging. We present results in Figure 3 for transfers over ESNet between high-performance parallel storage systems at ALCF and NERSC [4]. Each data point represents the average performance achieved when transferring many files of a specified size. We give results for Globus Online in two configurations: running between a single data transfer node (DTN) at ALCF and NERSC ("go-single-ep"), and (the default configuration) using the two data transfer nodes that are supported by ALCF and NERSC ("go"). We also show results for scp and for the commonly used globus-url-copy (GUC) client, both in its default configuration and when tuned by an expert. Scp performs badly in all cases. GUC with its default configuration performs badly for all file sizes. (The default configuration needs improving.) Tuned-guc performs much better than GUC in almost all cases, but less well than Globus Online for smaller file sizes—probably because Globus Online drives GridFTP pipelining more aggressively, due to the improved pipelining support in Globus Online's fxp client to GridFTP. Tuned-guc does slightly better than Globus Online for large files; thus, it seems that there remain opportunities to tune Globus Online performance further. Note also that the Globus Online transfers to a two DTNs vs. a single DTN are not substantially different except for the largest transfer. We conclude the bottleneck is not the DTNs, but the network or local storage.

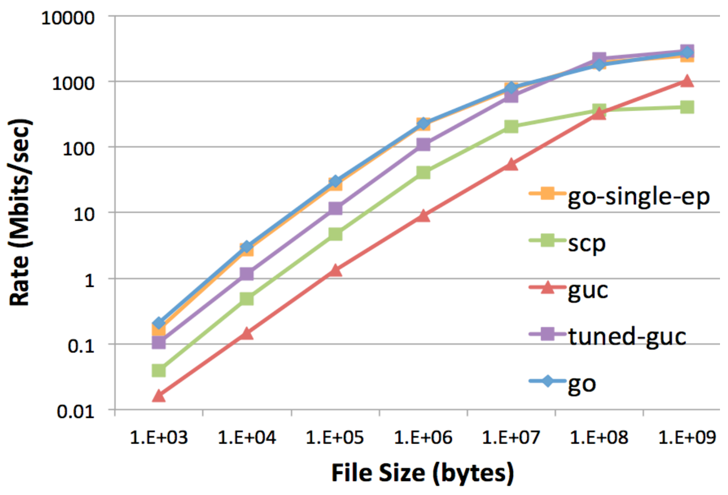


Figure 3: Globus Online performance ALCF-NERSC

Our initial experiences also suggest that SaaS can indeed have advantages as a delivery mechanism for research data management software. At various points during Globus Online's development, we variously discovered errors in the software and received urgent requests for new features. The resulting corrections and enhancements were delivered within hours to days, rather than the weeks or even months that sometimes ensue with traditional software distribution methods. In addition, our operations team demonstrated their ability to discover persistent errors associated with specific endpoints, and to notify sites of those errors in a manner that permitted their timely correction.

Inevitably we have also encountered some problems, particularly as we learned how to manage a production SaaS capability. For example, we had some brief service interruptions due to power system upgrades at the site (Argonne) used to host static (Web) content; this situation was resolved, but emphasized to us the benefits of hosting SaaS services on commercial hosting services. (Operation of the Globus Online service itself was not interrupted by this situation.) The May 2011 Amazon service outage, which received a lot of press, led to a less-than-one-hour interruption in Globus Online service.

7. Developing Back End Functionality for Globus.org

ISI worked on the Sprint tasks related to the development of back end functionality for Globus.org from October 2009 until April 2009. These developer resources were later redirected to testing, as described in the next section. At ISI, we implemented the interface between the Globus.org's RESTful service and the Globus.org's persistent backend (PostgreSQL database). The implementation used the Java Persistence Mechanism (JPA). The first version of the implementation used the JPA specification 1.0 and an implementation of the same by Apache called OpenJPA. This first version was used for the SuperComputing Conference 2009 (SC09) demo. The first implementation revealed several deficiencies in the JPA 1.0 specification and the Apache OpenJPA implementation. Two noteworthy limitations were: a) synchronization issues during caching in OpenJPA, especially when the backend database is modified outside of the JPA context and b) lack of support for filtering and sorting query results using JPQL (Java Persistence Query Language).

After SC09, we switched to a newer version of the JPA specification i.e. JPA 2.0 and EclipseLink, which was a reference implementation of the same. Although EclipseLink's caching mechanism was superior and faster than OpenJPA, implementing filtering in the queries was not an easy task. The Criteria Query API in JPA 2.0 that we used to implement filtering was complex and did not support pagination; a feature that we badly needed to limit the number of records displayed on each page. Also, Criteria Query API's support for sorting was limited and cumbersome. Since the JPA 2.0 specification did not support all the functionality that we needed, we switched to EclipseLink specific APIs for implementing filtering, sorting and pagination. This reduced the portability of the code.

Due to the limitations in the JPA 2.0 specification and its implementation, and concerns about its performance, it has been decided to abandon JPA and switch to AMQP (Advanced Message Queuing Protocol) and its implementation called RabbitMQ.

8. Testing for Globus.org

The ISI CEDPS team has took primary responsibility for developing the testing infrastructure for the Command Line Interface (CLI) of the Globus.org system and for setting up automated testing using the Bamboo continuous integration framework.

At ISI, we have done extensive testing of the CLI commands: a command-line interface to the Globus.org's back end database. We wrote a large number of tests to test the various options in the CLI commands and various combinations of those options. The tests fall under two broad categories a) basic tests, which test and validate the various options in the CLI commands and b) acceptance tests, which test and validate various combinations of the options and many common use case scenarios. There are 172 tests in 86 test files in the basic tests category. In the acceptance tests category, there are 242 tests in 43 test files. Each test can be run as a stand-alone test or part of a bigger test suite. The tests are written in Python and run using 'NoseXUnit'. 'NoseXUnit' extends unit tests to make testing easier. It produces JUnit like XML reports and is compatible with Nose and Junit testing frameworks.

All the tests (basic and acceptance) are run at ISI using Bamboo, a continuous integration framework from Atlassian. There are two different build plans configured in our Bamboo. The first build plan is run manually. This is used to test our tests before we commit them to the Globus.org's SVN. The second build plan is an automated build plan that runs nightly. It first creates three Amazon EC2 instances with two different Globus.org images. The first instance has the Globus.org's server image. The other two instances have the Globus.org's endpoint image. In other words, one Globus.org server and two Globus.org endpoints are created to run the tests. Once the instances are created, they are updated with the latest Globus.org code from the SVN. After this update, the tests are run against the Globus.org images (updated with the latest code from the SVN). It takes about 2 hours to run all the tests. After the tests are run, the Amazon EC2 instances are automatically shutdown. The results of the tests are available on our Bamboo for everyone to view using a browser.

9. Expanding Globus Online capabilities via virtual endpoints

ISI also contributed to the design and implementation of a Globus Online storage abstraction called a virtual endpoint. Let us define a physical endpoint to be a GridFTP server. We then define a *virtual endpoint* as a set of physical endpoints with an associated set of policies. When files are transferred to a virtual endpoint, its policies govern how and what files are transferred or replicated across its constituent physical endpoints. For example, a virtual endpoint can enforce a set of data replication policies to achieve a specified level of fault tolerance and performance, in a manner that is transparent to the user. More specifically, a policy might require that every file written to a virtual endpoint be automatically replicated on three geographically distinct physical endpoints to provide a high level of availability. A more sophisticated policy might enforce different replication strategies for files of different types or sizes.

A virtual endpoint also provides users with a unified view of a shared data space, as illustrated in Figure 6. Virtual endpoint VE1 (on the right) consists of two physical endpoints, GridFTP Server 1 and GridFTP Server 2 (on the left). Let us assume that policies associated with VE1 mean that the directory *user1_data_dir*, when written to VE1, is replicated at Endpoint1://home/user1/data_dir and Endpoint2://home/user1/data_dir. Any changes made to files within a virtual endpoint automatically get reflected in all replicas of that file within that virtual endpoint. When retrieving files from a virtual endpoint, Globus Online will choose the optimal replica to minimize the data transfer time.

We have designed and prototyped support for virtual endpoints for the Globus Online system. Our design gives users the ability to create and manage virtual endpoints and to associate policies with those endpoints. It also provides additional Globus Online commands to write, read, and edit files and to manage file transfers in a virtual endpoint. When a user writes files to a virtual endpoint, those files are written to one or more physical endpoints according to the specified policies. When a user reads files from a virtual endpoint, those files can be retrieved from any physical endpoint within the virtual endpoint that holds a valid copy of the files. The system also supports a check-out operation that allows users to copy files in a virtual endpoint to a local file system, manipulate those files, and then check them back in to the virtual endpoint, where they are automatically propagated to constituent physical endpoints according to the policies of the virtual endpoint. Users can also manage file transfers to/from a virtual endpoint. The system provides commands to query transfer status and to cancel active file transfers.

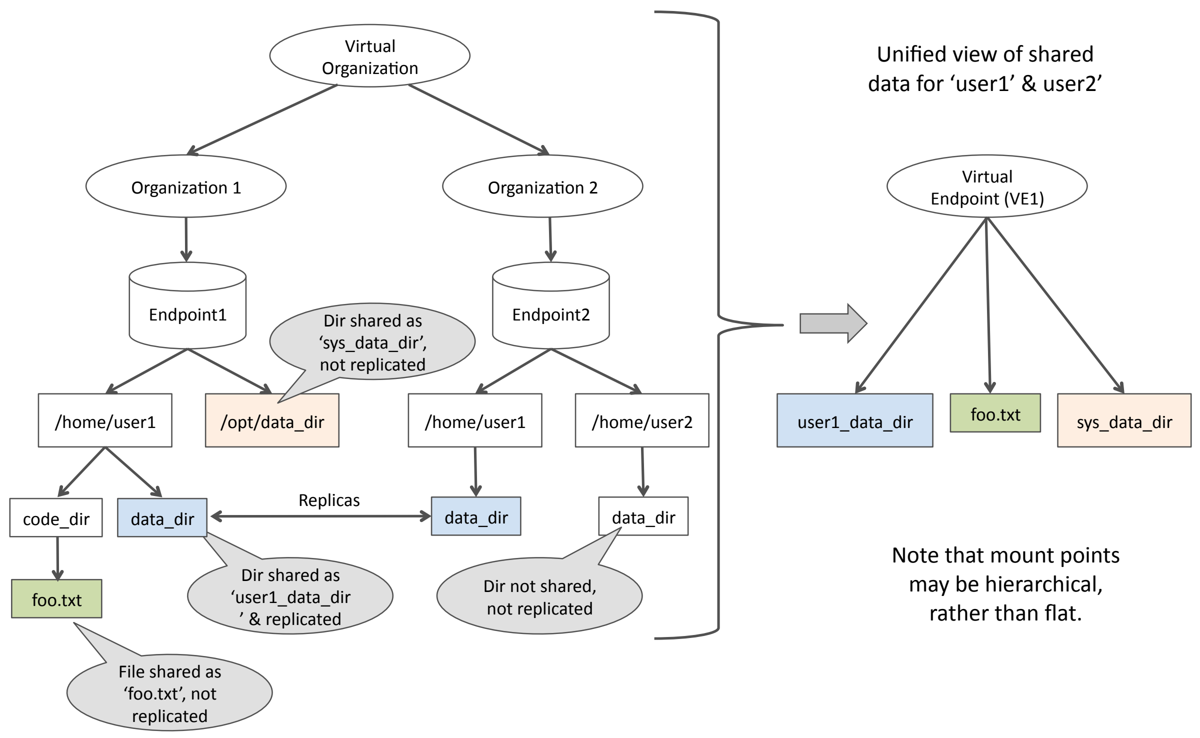

Virtual Organization
Organization 1
Organization 2
Endpoint1
Endpoint2
Dir shared as 'sys_data_dir', not replicated
/home/user1
/opt/data_dir
/home/user1
/home/user2
code_dir
data_dir
Replicas
data_dir
data_dir
foo.txt
Dir shared as 'user1_data_dir' & replicated
Dir not shared, not replicated
File shared as 'foo.txt', not replicated
Unified view of shared data for 'user1' & user2'
Virtual Endpoint (VE1)
user1_data_dir
foo.txt
sys_data_dir
Note that mount points may be hierarchical, rather than flat.

Figure 4: An illustration of the virtual endpoint design. See text for details