

## Kindura - Federating Data Clouds for Archiving

---

**Jens Jensen<sup>1</sup>, Roger Downing,**

*Science and Technology Facilities Council*

*Harwell Oxford Campus*

*Oxon OX11 0QX, UK*

*E-mail: jens.jensen@stfc.ac.uk, roger.downing@stfc.ac.uk*

**Gareth Knight, Simon Waddington, Jun Zhang, Mark Hedges**

*Centre for e-Research, King's College London*

*E-mail: gareth.knight@kcl.ac.uk, simon.waddington@kcl.ac.uk,*

*jun.zhang@kcl.ac.uk, mark.hedges@kcl.ac.uk*

We present the architecture and design of a "cloudy" data infrastructure for archiving and backup. By reducing the cloud elasticity, we are able to build a more cost effective service for archiving. DuraSpace and Fedora provide friendly front-ends for users. We have investigated several options for the back end, focusing currently on a federated iRODS infrastructure which will permit automatic replication and metadata extraction. All services will appear as "cloud-like", even internal ones – it is thus a hybrid approach that combines the advantages of the commercial/external (public) cloud with an institutional/consortium (private/community) cloud. This project will provide Infrastructure-as-a-Service (IaaS) components, via storage and compute services, but more importantly it will combine these, using DuraCloud and Fedora as enabling technologies, to provide an integrated Software-as-a-Service (SaaS) package of repository-centric services. While this is work in progress, we can already present results. In future work, DuraCloud will be extended to broker between the clouds.

*The International Symposium on Grids and Clouds and the Open Grid Forum  
Academia Sinica, Taipei, Taiwan  
March 19 - 25, 2011*

---

<sup>1</sup> Speaker

## 1. Introduction

As research in many disciplines increasingly turns digital, researchers need to process and archive the “data deluge” effectively. Cloud computing and storage, i.e., resources available and scalable on-demand, look like a good way to process data as researchers no longer have to worry about how resources are provided. However, a quick calculation shows that for longer term storage (say a year or more) for large data volumes (a terabyte, or more) commercial cloud storage providers quickly become more expensive than traditional data centres.

Funded by the Joint Information Systems Committee (JISC), the Kindura project aims to address this cost efficiency problem by providing cloud interfaces to traditional data centres: researchers will be able to use any cloud provider to analyse and process their data, while archiving the raw data and the results in the Kindura cloud for a fraction of the cost. (The cost effectiveness obviously depends on the volume of data and other factors; but as we shall see, Kindura has other benefits.)

The Kindura data cloud is built on the DuraCloud software [4], which provides a common interface to a variety of storage providers, including commercial cloud providers such as AWS, but also private clouds that expose a CDMI-compliant interface. Specifically, Kindura will be enabling iRODS (Integrated Rule-Oriented Data System), a grid storage system, to be used as a local cloud storage provider within the system.

The project includes work on a data migrator which knows about the costs of resources and is able to intelligently decide whether to move data or not. It can also be used to replicate data selectively across Kindura storage options, making it possible for institutions to have a local copy of their own data that is replicated remotely. Of course, this data can then be accessed locally.

Led by King’s College London (KCL), the Kindura project is a collaboration with the Science and Technology Facilities Council (STFC) and DuraSpace, with demonstrators of the service expected by July 2011.

## 2. Case Studies

To determine how cloud-based services may be applied in an academic working environment, the project team are in the process of interviewing researchers at King’s College London working in the academic disciplines of Environmental Science, Financial Mathematics, Biomedical Science and Humanities. The interviews examine their day-to-day activities when performing research and the potential value that cloud services may offer. The project is using the results of the interviews to develop a set of use cases for preservation of research data.

Although work is still ongoing on the case studies, a number of trends have begun to emerge. At a broad level, the perceived use and value of cloud services for storage and compute is influenced by the amount and type of data handled, research activities undertaken, the legal framework associated with the data and the speed of access. All researchers interviewed during the work were interested in the use of cloud services for large-scale storage of their research data, either as a trustworthy primary store, or as a backup store that supports existing storage facilities. It was recognized that contractual and legal obligations meant that personal data could not be stored outside the institution or outside the EU. There were also concerns that the payment model associated with cloud storage meant that it would be uneconomical to store data in the cloud in the medium to long-term. This concern is particularly relevant when considering storage of research data. It was questioned whether the ongoing cost of cloud storage fits with the short-term funding arrangement of many research projects. By comparison, interviewees found it more difficult to consider the use of cloud compute services, preferring to utilise local computing environments that were available in-house, rather than re-engineer work activities to fit around cloud infrastructure.

Of those interviewed, researchers working in the fields of Environmental Science and Financial Mathematics expressed most interest in the potential use of cloud compute services. Researchers working in these fields store and analyse large datasets comprised of tabular information, in order to address a research question. This processing can be performed using cloud services.

Despite interest in the field, they highlighted two key issues – degree of control over the remote environment and data transfer speeds. These key issues would need to be addressed prior to the adoption of cloud compute services for short processing activities. It was seen as important that the researcher had root access to the remote virtual machine, enabling them to install and configure the software to their own needs (i.e., IaaS). The time required to transfer data between local and remote storage, made worse by limited network infrastructure in particular academic buildings, was considered to be a factor that would inconvenience academic staff and PhD students analysing their data with short deadlines.

Researchers interviewed in the humanities and biomedical sciences were less engaged with cloud compute services, expressing uncertainty on how they could be used for the types of data analysis they perform, which typically require use of diverse tools and workflows. Greater interest was expressed in the potential use of cloud compute services to enable secondary use of data, e.g. access to and manipulation of research outputs through a web browser.

### **3. Architecture**

Figure 3.1 illustrates the architecture of the Kindura pilot system. Kindura uses a hybrid cloud approach that combines internal storage infrastructure developed with iRODS that is repurposed with a Representational State Transfer (REST) interface, with commercial cloud providers such as Amazon Web Services and Microsoft Azure. The storage and compute service

providers connect to the DuraCloud [3] application. Developed by DuraSpace, DuraCloud is open source middleware that provides a framework for addressing multiple commercial and non-commercial cloud storage resources via a common REST application programming interface (API). DuraCloud provides both basic create, read, update and delete (CRUD) operations for storage as well as a framework for deploying common services such as data replication, content format transformation and bit integrity checking.

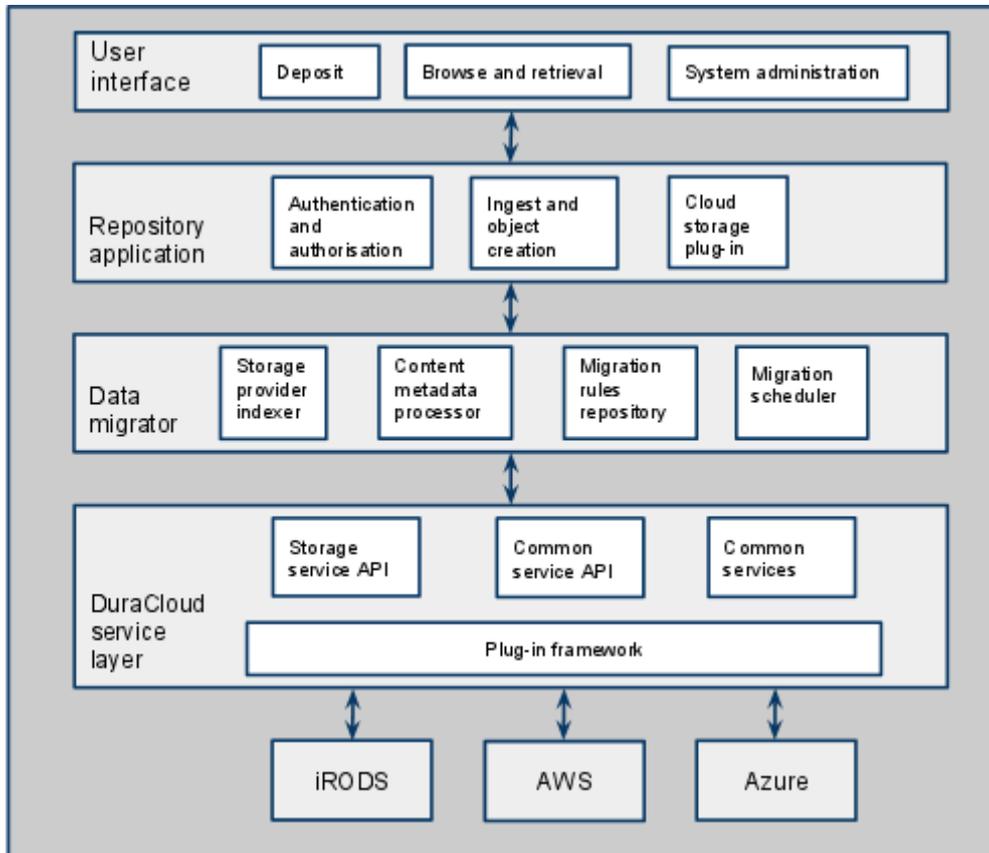


Figure 3.1 Kindura system architecture

The DuraCloud service layer is exposed to a Data Migrator, which handles allocation of data objects to storage providers, both for ingest of the content, as well as for migration during the lifecycle of the data. The Data Migrator stores an index of storage provider attributes such as cost, reliability, security and access latency, which are updated periodically to reflect changes in providers, costing models and infrastructure provision. Metadata associated to content collections and captured at ingest is parsed by a Content Metadata Processor in order to determine storage and access requirements of specific datasets. Migration of data is based on business rules that are applied by the Migration Scheduler and can be edited by the system administrator to reflect changing business requirements.

The Kindura Data Migrator exposes a REST interface that enables it to be integrated as a back-end storage provider with multiple external front-end applications. For the purpose of the

Kindura pilot, we are carrying out integration with the Fedora Commons repository. For the initial prototype, we are storing the binary data in Kindura storage, whilst the content metadata is retained in the repository's local storage. Since DuraCloud does not provide individual user accounts, user authentication and authorisation are managed by Fedora.

A web-browser based user interface is provided to enable users to deposit content items or collections as well as enter the required preservation metadata. The ultimate aim of such a cloud-based system is to provide resources that are transparent to the user. In this way, the system can make the most cost effective and efficient decisions regarding content storage based on the criteria entered by the user and the institutional policies. The user interface also features browse and retrieval functionality.

## 4. Implementation

### 4.1 DuraCloud

The Kindura team has integrated the open source DuraCloud middleware developed by DuraSpace. Kindura participated in the pilot programme for trialling and initial evaluation of DuraCloud. DuraCloud is released under Apache License, Version 2.0.

DuraCloud provides three web applications DuraStore, DuraService and DurAdmin. **DuraStore** provides a REST API for initialisation of *stores* (storage providers), creation and deletion of *spaces* (content containers), storage and retrieval of content and metadata, and initialisation for authorised users. **DuraService** provides a REST API to initialise common services, specify authorised users, and to configure, deploy and update the common services. DuraCloud is supplied with common services to support storage and preservation. These include services for data replication across storage providers, viewing of images files via the Djakota image server, media streaming, image format transformation and bit integrity checking. Services can be run in bulk and deployed in the cloud to enable ingest of large volumes of data exploiting elastic compute resources. **DurAdmin** is the user-facing application through which DuraCloud exposes DuraStore and DuraService functionality. It allows users to access all storage system capabilities and service administration capabilities via web browsers.

### 4.2 iRODS

iRODS is a data management and virtualisation middleware provided by the Data Intensive Computing Environments group at the University of North Carolina [5]. iRODS provides a unified namespace for files stored in a multitude of disparate storage technologies, and allows files stored within the system to be replicated or migrated between different storage resources, thus permitting the implementation of hierarchical data storage. A particular feature of iRODS is its ability to represent data management policies in terms of rules, by means of a rule engine that executes rules and allows pre-defined sequences of actions be executed in particular circumstances. These features of iRODS can be exploited to implement automated digital curation and preservation functionality [6]. iRODS also has the ability to federate

multiple geographically separate iRODS systems, and files can be copied and replicated between the members of a federation, aiding redundancy and accessibility of data.

The transfer is managed by embedding an iRODS client into the DuraCloud infrastructure, based on the Jargon interface. iRODS will make a decision based on the size of the file whether to transfer in a single stream or multiple parallel streams. Once it reaches iRODS, it will be stored on disk: at this point, we need to know from the file metadata whether the file is to be migrated to tape. If so, small files have to be bundled up before being migrated to Castor (described below.) In general, we prefer using the `iphybun` command to bundle the files, as they will still appear as individual files to the user. We can also use a Fedora object to describe a bundle of files (see below) – moreover, the object could then have a more complex structure which reflects the internal structure of the deposited collection, using RDF relations. This is something we are planning to do.

If any one of the files in the bundle is read, the whole bundle will have to be read from tape, but there may also still be a cached copy on disk, either on the iRODS disk or on Castor’s disk. We don’t currently have a scheme for “garbage collection” (i.e. deleting files from disk which have been migrated to tape), but implementing one should not be difficult.

There are two types of metadata, one to describe the content and format of the object, and another to describe the data placement policy, e.g. whether the file should be written to tape. The latter is described in section 5.1. For the former, we are using Fedora’s FOXML to describe the data, using MediaClient to generate the description. Using Fedora in “externally referenced” mode, the Fedora object contains pointers to the location of the data which can be used to access it using the DuraCloud REST API. The physical copies of the file will be in one or more of the clouds managed by DuraCloud (as we shall see in section 5.2, it will be useful to always have an archived copy in the Kindura iRODS cloud.) The disadvantage is the lack of abstraction: the FOXML object refers to the physical location of the files, and the client will have to select one of the sources even if they all refer to replicas of the same file, so Kindura is not able to optimise this step (e.g. by choosing the cheapest provider). However, this is also an advantage: it is often useful to know where the replicas of the file are located, e.g. when the client wishes to analyse the data with a commercial cloud provider, it is useful to know if there is a copy of the data in that cloud already.

At STFC the long-term data archival system in use is Castor, a large-scale tape storage system developed by CERN to meet the demanding data requirements of the Large Hadron Collider. Work within the Kindura project will provide a driver for iRODS which will permit the storage of data through iRODS and into Castor. However, at the level of the Kindura system (DuraCloud plus the data migrator), iRODS is regarded as a “black box”, essentially another storage solution exposing a cloud-like interface, and the replication and rule execution features of iRODS will not be visible. That said, as iRODS rules can be invoked explicitly via a supplied API, this offers the possibility of using this functionality to implement “cloud computation” services by mapping the API to the corresponding cloud API.

### 4.3 Interfacing to commercial cloud services

The DuraCloud middleware is supplied with a plug-in framework for integrating cloud storage providers. Adaptors for Amazon Web Services, Microsoft Azure and Rackspace are supplied out of the box. The adaptor is required to convert the provider specific web service calls to the format required by DuraCloud.

#### 4.3.1 CDMI

The Cloud Data Management Interface (CDMI) is the Storage Networking Industry Association's cloud storage interface, from the Cloud Storage Initiative . As the only standard for cloud storage (all the others are vendor-defined), it is our best hope for a standard interface to cloud storage. While it is “inspired” by existing cloud storage interfaces, it is quite a rich interface offering capability objects, notification, object serialization, queues, and more. Some of this is may be out of scope of what we can realistically aim for in this project, but it would certainly be interesting to have a CDMI interface to iRODS. Currently we know of the (Java-based) SNIA reference implementation (easily located from the SNIA website, [www.snia.org](http://www.snia.org)); and one in C++, by personal communication - investigating and interoperating is an obvious area for future work. See also “Related Work” below.

## 5. Data migration

### 5.1 Constraints

Kindura uses an XML content classification schema to determine the preservation requirements of content. The content classification can be performed at the content collection, sub-collection or item level, depending on the specific context. Four basic categories were identified based on the research carried out with users:

- Ownership: the ownership and usage rights of content collections can restrict where the content is stored and the operations that can be performed on it.
- Protective marking describes the sensitivity of data, whether the data contains personal data and export classifications. In this category, we also consider the business need for retaining the data.
- The likely or intended usage of the content will influence the choices of storage location. Content intended for frequent and real-time access will need to be stored on low latency storage devices. Network speeds also play a major role in determining the portability of data.
- The provenance of the data includes such factors as whether experimental data is a source dataset, an intermediate output of an experiment or a final dataset.

Much of the content classification needs to be carried out manually, although enabling users to enter metadata at the collection rather than item level considerably reduces the workload. An interface is provided in Kindura to enable researchers to enter such metadata. There are a number of mechanisms that can be used to reduce the data entry requirements when data is ingested into the repository that we are investigating.

- Many researchers use a systematic folder hierarchy and labelling to structure their experimental datasets, which assists in automatically determining the attributes of specific data items. This requires configuration of a custom upload tool by the researcher to suit their specific data format.
- Experimental tools may require entry of metadata to describe the datasets generated or may automatically generate metadata, for instance to describe intermediate datasets that are generated. Such metadata can be repurposed for data preservation purposes. This requires a specific plug-in for each experimental research tool to perform the metadata extraction and format conversion.

Constraints on where data items should be stored are captured in a storage provider XML schema, which is periodically updated. The main categories that we have identified are:

- Storage performance, which includes such factors as whether storage is primary or backup, the response times for retrieving data, and the network speed.
- Security factors describe the potential for loss, corruption or unauthorised access to data. For instance, private clouds are inherently more secure than public clouds due to the reduced risk of corruption between VMs running on the same physical machine. This category also includes potential legal constraints such as the geographic location in which data can be stored. This is particularly relevant for personal data that might be stored in medical research or information relating to intellectual property. Many commercial cloud providers enable users to determine approximate geographic areas in which will be stored, such as within the EU.
- Reliability of storage describes factors such as provider availability, upper thresholds on data errors, the number of replications held and the backup frequency. Typically commercial cloud providers will hold multiple copies of data, and in some cases, the user can change the number of replications held by a given provider.
- Cost is a key factor in determining the location of storage and the replication of data, and includes both storage and network costs. This factor is described in more detail in section 5.2.

The classification of storage provider attributes involves many complex factors, and is in some cases subjective. In order to make the assignment of content items to storage spaces tractable, we have reduced the storage provider attributes to a set of numbered levels for each of the categories above. In some cases, it may be difficult to achieve an accurate assessment of the capabilities of commercial cloud providers, as this would typically involve confidential proprietary information. In this case, we are reliant on the minimum performance standards provided by Service Level Agreements (SLA).

Cloud providers have traditionally provided flat rates for storage services. However, as many real-time online applications such as app stores are migrated into the cloud, tiered storage is becoming increasingly common. Since higher response speeds in commercial providers are

typically achieved by higher levels of data replication, there is a corresponding costing structure. Therefore, for individual providers, we have developed the notion of *storage profiles* to reflect the differing offerings of storage providers. This allows us to differentiate between spaces with different storage characteristics within the same store.

## 5.2 Cost Models

The role of the cost model is to minimise costs for the customer, or possibly keeping it below a certain level, while working under a given set of constraints. Examples of constraints could be:

- Ensure data is in a certain location before a given time, and/or until a given time (“location” meaning a site, or a cloud.)
- Ensure Kindura always holds copies of data at three geographically separate sites.
- Ensure data does not leave the country, or geographical region.
- Always have a copy on tape.
- Maintain low overall cost of both storage and the associated processing, e.g. by picking storage and compute from the same commercial provider.

While we cannot possibly satisfy all constraints in a small project like Kindura, we can develop a cost model which can guide the user in finding cost-effective storage for their data.

Before we provide a brief description of the cost model, let’s first look at the costing of storage services. Traditionally, STFC’s Atlas data store would charge users for their *allocated capacity*: if a customer wants a capacity of, say, 10TB, they would have this allocated to them (disk servers, tapes, a share of a tape drive, possibly licence costs depending on the type of service, special ingest services, etc), and would normally pay for the allocation, as it is available to no-one else, rather than how much of the allocation they use. Conversely, with a commercial cloud provider, the customer pays only for what they use, but then they pay for everything: use of network, data ingest, data access (unless from the same provider’s compute cloud), storage.

The traditional data centre is far more cost-effective for longer term storage for larger data volumes (>1TB), but you have to stay within your allocation (or have it extended by an administrator.) The main purpose of Kindura is that its costing is like a traditional data centre: it is more cost effective, but the shares are pre-allocated. Kindura looks like a cloud, but is not elastic, and is therefore cheaper for long term storage, as will be evident once numbers are fed into the cost model.

The basic premise is to look at *replicas*, *locations*, and *time*. The migrator, knowing how many replicas of the data are available and their locations, can speculatively try to reduce costs by contemplating the deletion of some of these replicas. An example will illustrate this best.

Let us consider a 2TB dataset, which has two constraints: it must always have a copy within Kindura, and it must be available for processing inside a commercial cloud two weeks from now, for a duration of three months. Assuming the data has already been ingested into

Kindura (e.g. locally, at the researcher’s institute), we can now consider the use of commercial cloud providers for analysis. The migrator should ideally know the bandwidth of the available providers, so it can start moving the data a week early, say. The cost model will then take into account the cost of moving the data across the network, as well as the “integral” of the accrued costs of intermediate storage of the dataset as it is moved, as well as the cost of storage for three months. Knowing that the data can be deleted, the migrator does not have to consider the costs of moving the data back out of the provider: it can be replicated back out of Kindura at no cost.

We can now do this calculation for each commercial provider based on advertised and/or measured data, and use the results to choose one of them. Moreover, if the commercial costing changes, we can revisit the dataset during its three months’ sojourn in the commercial cloud, to consider the options of switching it over to another provider. In this case it would still be cheaper to recreate the copy with the new provider out of Kindura - as Kindura will incur no network costs - and the copy with the first provider can then be deleted once the processing is running with the second.

Finally, it is of course possible that derived data is generated by processing it in the commercial cloud, data which itself needs archiving. A priori, the migrator has no way of knowing how much data, if any, is created by processing, but it will need to take this into account (i.e., copying the data from the commercial provider into Kindura, as well as its storage.) We have two options: either the researcher must inform the migrator about the size of derived data, or the migrator must watch the researcher’s account within the commercial cloud. The latter may of course incur unforeseen costs - the best approach here is probably for the migrator to watch the current costs, to use this data to predict the expected overall costs of the three months storage (and the migration out of the cloud), and to include the *growth* of derived data with time into the calculation.

For reasons of space, we have here provided only a description-by-example of the cost-based migration.

### **5.3 Data Migrator**

A framework for defining and implementing business rules is under development that will allow storage requirements for content collections to be determined based on the content classification schema and the storage provider characteristics. The rules will be applied by the Data Migrator, which will determine both storage requirements at ingest as well as performing periodic appraisal and migration as appropriate. The business rules additionally encapsulate the institutional data retention policies, as well as any explicit information entered by the user.

## **6.Related work**

There is currently considerable interest in the preservation and reuse of research data. Much of this activity is being driven by funding bodies that are increasingly mandating preservation of research data as part of the conditions of support. Journals are in some cases

requiring that experimental results be made available to support the findings in published papers.

The JISC Flexible Services delivery programme features two projects, Kindura and Flessr that are focusing on applications of cloud computing to support research. Flessr [1] is developing a pilot cloud platform based on a hybrid public-private cloud infrastructure to support two specific use cases. The software development management system demonstrator allows multiple cloud platforms to be used for large project development. The storage demonstrator provides a simple on-demand block storage system for research data. The eZeel [2] project at Belfast e-Science Centre has developed a framework for deployment and management of multiple heterogeneous cloud instances.

The Wellcome Trust Sanger Institute is using iRODS primarily as an institutional repository.

ARCS [7] is currently using iRODS as a national data management infrastructure. As the ARCS project is about to end, it remains to be seen who will take over the operations of the infrastructure. Both NeCTaR [8] and RDSI [9] are funding work in data management in Australia, some of which may be based on iRODS. As Australia have an access management (Shibboleth) federation [10], it may be useful to use that to manage access to the infrastructure.

Also related is the ASGC work on building an SRM [11] interface to iRODS [12], as SRM is, from a high level perspective [12], quite close to CDMI.

## 7. Future work

This paper has described work in progress. Future work will obviously include looking at how the cloud best interfaces to iRODS, the practical experiences with merging data grids (based on iRODS) with data clouds, and the usage models arising from this. The role of CDMI is still open. Other user communities across the world are in similar situations (e.g. the Australian research data infrastructures), and liaising with them will be interesting. On the other side of the architecture diagram, it will be interesting to see what the DuraSpace community make of this work. Also, Kindura provides no compute resources - in principle, we could combine Kindura with academic compute resources, either cloud ones or traditional grid or cluster. This would also be interesting from a cost model perspective. Apart from the geographic placement of replicas, there are likely to be other security aspects that we can identify but will not have time to address in the project. Finally, we need to follow up on the sustainability aspects of providing a Kindura cloud service in production.

### Acknowledgments

The authors are grateful to the Joint Information Systems Committee for funding this work. The authors would also like to thank Guy Coates, Peter Clapham, and Gen-Tao Chiang of the Wellcome Sanger Institute, Andrew Richards from UCL, and Glenn Moloney from NeCTAR for interesting discussions.

## 8. Conclusion

Clouds are increasingly being used as a base for collaborative research. The attraction of the cloud model is based on convenience: resources are available on demand, without the consumers having to worry about how they were provisioned. However, as we have seen, sometimes they do worry: they need to know where their data is, and how it is protected. We have also seen that for certain types of data usage, traditional data centres are more cost effective, e.g. for long term storage of large volumes of data. We have described in this paper a cloud interface to traditional data centres, providing a cost effective hybrid where data can be moved in and out of Kindura. In fact, for the most common use cases behind Kindura, it turns out to be the most effective model to have the a permanent copy in the Kindura cloud, and then copy the data into commercial cloud resources whenever this is required.

Additional benefits to Kindura include:

- Data is cloned to precise locations, based on federated iRODS services: so the locations of (replicas of) the data are known.
- The Kindura cloud can be configured to always have a copy at the users' home institution, where it can be accessed locally.
- Since data is copied over JANET, copying data into and out of Kindura incurs no extra cost. This can make Kindura a cost effective way to distribute data (again for certain volumes and usage patterns.)
- While Kindura does not provide compute resources, so must always replicate data into compute clouds, it does merge some of the advantages of data grids into the hybrid cloud model: the type of resources allocated via Kindura is known (e.g. bandwidth, types of storage), so a different type of SLA can be offered to the user.

## References

- [1] FleSSR: <http://www.jisc.ac.uk/whatwedo/programmes/flexibleservicedelivery/flessr.aspx>.
- [2] T Harmer, P Wright, C Cunningham, J Hawkins, R Perrott , An application-centric model for cloud management, services, pp.439-446, 2010 6th World Congress on Services, 2010.
- [3] DuraCloud: <http://www.duraspace.org/duracloud.php>
- [4] <https://wiki.duraspace.org/display/DURACLOUD/DuraCloud>
- [5] A. Rajasekar, M. Wan, R. Moore, W. Schroeder, A prototype rule-based distributed data management system, in: HPDC workshop on "Next Generation Distributed Data Management", Paris, France, May 2006.
- [6] Hedges, M., Blanke, T., Hasan, A. (2009). Rule-based curation and preservation of data: A data grid approach using iRODS, Future Generation Computer Systems, Vol. 25, 4, <http://portal.acm.org/citation.cfm?id=1486271.1486375>
- [7] Australian Research Collaboration Services (ARCS): <http://www.arcs.org.au/index.php/services/data-services>
- [8] National e-Research Collaboration Tools And Resources (NeCTAR): <http://www.nectar.unimelb.edu.au/>
- [9] Research Data Storage Infrastructure: <http://rdsi.uq.edu.au/>
- [10] Australian Access Federation (AAF): <http://www.aaf.edu.au/>

[11] <http://www.ogf.org/documents/GFD.129.pdf>

[12] Storage Resource Management at OGF31: [http://www.ogf.org/gf/event\\_schedule/index.php?id=2199](http://www.ogf.org/gf/event_schedule/index.php?id=2199)