Journal of Cloud Computing
a SpringerOpen Journal

RESEARCH Open Access

Cloud repositories for research data – addressing the needs of researchers

Simon Waddington[1*], Jun Zhang[1], Gareth Knight[1], Jens Jensen[2], Roger Downing[2] and Cheney Ketley[2]

* Correspondence:
simon.waddington@kcl.ac.uk
[1]Centre for e-Research, King's College London, 26-29 Drury Lane, London WC2B 5RL, UK
Full list of author information is available at the end of the article

Abstract

This paper describes the problems and explores potential solutions for providing long term storage and access to research outputs, focusing mainly on research data. The ready availability of cloud storage and compute services provides a potentially attractive option for curation and preservation of research information. In contrast to deploying infrastructure within an organisation, which normally requires long lead times and upfront capital investment, cloud infrastructure is available on demand and is highly scalable. However, use of commercial cloud services in particular raises issues of governance, cost-effectiveness, trust and quality of service. We describe a set of in-depth case studies conducted with researchers across the sciences and humanities performing data-intensive research, which demonstrate the issues that need to be considered when preserving data in the cloud. We then describe the design of a repository framework that addresses these requirements. The framework uses hybrid cloud, combining internal institutional storage, cloud storage and cloud-based preservation services into a single integrated repository infrastructure. Allocation of content to storage providers is performed using on a rules-based approach. The results of an evaluation of the proof-of-concept system are described.

Keywords: Hybrid cloud storage, Fedora repository, DuraCloud, Cost optimisation, Rules engine

Introduction

Data-driven research has become increasingly prevalent across a wide range of disciplines ranging through science, humanities and social sciences. The size and complexity of data are increasing at a rapid rate in parallel with advances in computing and storage technologies. Thus management of data is therefore becoming increasingly important.

The open access agenda [1] promotes free online access to the outputs of publicly funded research. Although initially targeting peer-reviewed journal articles and conference papers, recently attention has been focused on research data. Preservation of research data has become an increasingly critical issue for Higher Education institutions. UK funding bodies such as EPSRC (Engineering and Physical Sciences Research Council) and NERC (Natural Environment Research Council) mandate the retention of data for extended periods (e.g. 10 years or more) [2,3] beyond the end of the funded project work. Academic publishers are also increasingly requiring that research data should be retained to support academic publications. There is currently only a limited range of

infrastructure and tools for managing data over long timeframes. The Digital Curation Centre (DCC) [4] maintains a catalogue of the available tools and services for digital preservation, including preservation of research data.

There are considerable benefits to be gained in effective curation and preservation of research data. It enables datasets to be shared and reused, and provides a basis for validating research findings. There is also a wider economic benefit in enabling innovation, policy and practice to draw from academic research.

Many research organisations are under pressure to reduce costs whilst at the same time providing improved levels of service. Cloud computing is viewed as a method for achieving efficiency and cost savings through the use of commodity IT infrastructure.

Cloud storage provides a particularly attractive option for "small science" subjects, which cannot maintain their own dedicated infrastructure. Frequently, central IT organisations do not have the resources to support the requirements of researchers and research groups are required to find their own solutions. Trust, privacy, quality of service and service availability are all factors, which we explore in more detail in the paper. SLAs (Service Level Agreements) are difficult to interpret without specialist legal knowledge and often do not require the safeguards required for retention of critical data. There is a significant risk of catastrophic loss as well as breaches of data protection legislation.

Hybrid cloud storage combines multiple cloud storage providers into a single infrastructure [5]. It enables institutions to retain sensitive data in-house, whilst providing an elastic extension to their existing internal storage, at the expense of added complexity.

The main contributions of this work are:

(i) To design, implement and evaluate a repository framework for preservation of research outputs based on hybrid cloud.
(ii) To define and implement algorithms to manage the storage, replication and migration of content across multiple repository storage providers. This takes into account both policy-based data management and minimisation of storage costs.

The requirements for the framework are based on a set of case studies conducted with researchers at King's College London and STFC[a].

This work was funded by JISC[b] as part of the Kindura[c] project as part of the Flexible Service Delivery programme [6]. The authors would like to acknowledge the support of Michele Kimpton and the team at DuraSpace in the installation and configuration of the DuraCloud software.

Prior art

Integrating multiple cloud providers is not straightforward due to the differences between service APIs. Cloud storage standards working groups have been established by the Open Grid Forum (OGF) and the Storage Network Industry Association (SNIA). The CMDI (SNIA) [7] and OCCI (OGF) [8] standards consider both the storage and administrative APIs. Adoption of standards, particularly by major providers, is still limited.

OpenNebula [9] is an open-source cloud computing toolkit for managing heterogeneous distributed data centre infrastructures. The toolkit includes features for integration, management, scalability, security and accounting. OpenNebula is primarily a toolkit for building cloud infrastructures from the bottom up. In contrast, we are primarily concerned with managing distributed cloud infrastructures that expose a simple REST web service interface, but have their own independent management structures.

The EC FP7 Reservoir project [10] considered federation of cloud infrastructure to enable organisations to pool cloud resources to increase scalability. Such an approach goes beyond our approach of providing a service connector, requiring a much higher level of system integration. Another FP7 project, Contrail [11], plans to build federated hybrid clouds out of commercial public clouds and private clouds based on OpenNebula, via an adapter interface.

Various authors [12] have considered using the grid storage system iRODS [1] as a storage provider for the Fedora repository. For example Pcolar et al. [13] describe a use case where content is ingested into Fedora, then processed using iRODS microservices to extract metadata and store in iRODS. Hedges et al. [14] represent digital curation policies and strategies as iRODS rules in order to automate preservation processes.

Tessella Safety Deposit Box (SDB) is a commercial product that provides long term storage based on disc and magnetic tape suitable for high volume data storage, and has been evaluated for the storage of ISIS data [15]. It provides ingest workflows, period checksum verification of data, basic format conversion functions and reporting. SDB does not currently support preservation of data in the cloud.

The Cloud Adoption Toolkit [16] is a planning tool that enables estimation of costs of cloud computation and storage costs to be made. It is an offline tool rather than being integrated with a live repository or storage system.

The International Organisation of Standardisation (ISO) has developed a process of certification for digital archives as ISO 16363:2012 Audit and Certification of Trustworthy Digital Repositories [17]. On 14th February, 2012, this work reached stage 60:60, "International Standard published" [18]. ISO 16363 enables the assessment and certification of a repository as being a trustworthy digital repository. Since Kindura is not a production system, certification was outside the project scope. However, Kindura addresses some of the themes in Infrastructure and Security Risk Management area.

Issues relating to trust in cloud have been investigated extensively in the literature, particularly for the storage of digital assets. This research considers both the design of clouds themselves as well as the selection of cloud providers based on trust-related criteria (see [19] and the references therein). Kindura aims to combine existing storage providers, with their own management infrastructures, into a practical and trustworthy system rather than addressing lower level issues relating to trust.

Service Level Agreements (SLAs) record the agreement between cloud providers and consumers [20]. An SLA is typically a plain-text document. Providing a standardised machine readable SLA in a format such as XML would be an important advance to enable automated service comparison and brokerage. The design of the Kindura prototype was based on the available SLAs from several major cloud providers. Although this is an area of current research, none of these commercial vendors provided a public machine readable SLA.

Data protection and privacy are also major concerns for storing sensitive data in the cloud [21]. Legal and regulatory issues play a major role. User perception and confidence is critical to ensure the uptake of cloud related services. The Kindura approach was based on a pragmatic view of minimising risk to data assets by matching of data attributes to available storage providers. Provenance is also an important issue and models such as the Open Provenance Model can be used to express this information in a way that enables exchange between systems [22].

Case studies

Methodology

The Kindura project carried out a requirements study with stakeholders, including researchers, archivists and IT staff, to determine requirements and issues in the use of cloud storage. Researchers in environmental science, biomedical sciences, psychiatry, financial mathematics and digital humanities were interviewed, some of whom are already making use of commercial cloud to store their research data. A total of twelve researchers were interviewed including four each in environmental science and biomedical sciences, two in digital humanities and one each in financial mathematics and psychiatric research.

Interviews were conducted face-to-face, lasting between sixty and ninety minutes. They were based on a structured set of fifty questions covering issues such as current arrangements and future requirements for research data storage and preservation, existing and planned storage infrastructure, the volumes and types of data to be stored, requirements for reuse of legacy data, and usage and awareness of cloud computing.

A common set of questions were used for all the interviews, which were reviewed by staff at both project partner organisations to avoid bias. Where practicable, interviews were conducted by or included staff from a project partner from a different organisation to the interviewee.

Environmental science

Environmental researchers at King's College London and STFC work across disciplines exploring topics related to historical and contemporary environment. A high value is placed upon source data, which frequently contains content that is unique and cannot be reconstructed. Derived data are perceived to have less importance, since it can often be easily regenerated. Research teams perceived themselves to have data storage and processing requirements that were much greater than could be supported by IT services and had made significant investment in implementing, maintaining and upgrading their departmental IT infrastructure. This had the advantage of giving them direct control over the storage of their data, but placed a considerable administrative and financial burden on their team. Researchers typically stored their data in folder structures, using predefined naming conventions to identify source data, intermediate experiments and final results.

Financial mathematics

The financial mathematicians interviewed frequently collaborate with financial institutions to address quantitative problems, using proprietary data, as well as public

information derived from the finance market. The analysis performed by financial mathematicians is computationally-intensive, resulting in a need to invest in powerful processing hardware. In contrast, providing storage capacity to manage and preserve their datasets had not been considered in depth. The financial mathematicians interviewed for Kindura were aware of the potential value that cloud computing may offer for analysing and storing datasets without needing to maintain local infrastructure. However, source data made available under a commercial licence often stipulates storage within the institution. Thus, when using external storage, strict enforcement of storage policies would be required. There was also concern about the bandwidth for transfers to and from external storage.

Digital humanities

Digital humanities researchers at King's produce a broad range of content types, sizes and formats, and organised in diverse structures. Several digital humanities projects were found to be actively using commercial cloud providers as a storage and content delivery system. Amazon S3 Web Services is currently used to deliver database-driven sites, one of which contains streamed audio content, as well as for offsite backup data storage.

Psychiatric research

Psychiatric researchers at King's and the collocated NHS[d] Foundation Trust investigate topics related to psychiatry, psychology, basic and clinical neuroscience, making use of quantitative methods. Medical data containing information about individuals is subject to data protection and other government regulations that prohibit its storage in off-site locations. IT staff interviewed were aware of cloud services, perceiving it as potentially useful for performing common processing activities, such as format conversion and metadata de-duplication, as well as data backup. However, the complex legal compliance issues associated with clinical data, as well as the need for fast access to data, would necessitate that some critical data could only be stored on internal servers.

Biomedical sciences

Biomedicine comprises wide range of disciplines with almost as wide a range of data storage requirements. Biomedical researchers produce a range of digital material, including images, tabular datasets, 3D models, binary data files, and structural/description metadata in various file formats.

The scientists interviewed for Kindura at KCL were aware of the value that cloud services offered for data storage and considered the provision of an institution-wide service to offer financial economies of scale and resource saving (staff time and effort to select, purchase and setup a local infrastructure) over local systems. However, concern was expressed about potential escalation of data storage costs, as "unlimited" storage could result in staff adopting a less strict appraisal process for evaluating the data that should be deleted.

The invited interviewees at STFC focussed on small angle neutron scattering of biological samples (SANS). The important aspect for Kindura is that users either come on site with their samples or send them by post (or courier), and they often get their data

burned on a CD (or DVD) which they take back with them, or it is returned by courier. Users can of course also download their data, but competition for bandwidth at their home institution, or other time constraints, often prevent this. They would like to offer an approach where research data can be made available on the cloud, without having to be sent to the researcher's home institution. By enabling researchers to move all, or parts of it, to a public cloud, or to a national community cloud, for analysis, a model could be offered where the full research data are preserved, but can also be shared, analysed piecemeal, or explored.

While the data here contain no personal information, there is still a requirement for data security because of the often sensitive or competitive work. For publicly funded research, the important aspect is that data are preserved and protected until the data policy requires that it be published.

Discussion

The perceived use and value of cloud services to individual researchers and departments was influenced by several factors, including previous financial or time investments that had been made into local storage/processing infrastructure (IT staff were less willing to consider cloud alternatives, if they had an existing infrastructure in place); staff time and training requirements necessary to transition to cloud services; the costs of using cloud services (cloud services must be affordable and offer better value over storage and processing data on non-cloud systems), and awareness of "pressure points" in existing local infrastructure (e.g. insufficient storage capacity, limited processing capabilities). The case studies also identified a number of organisational and technological issues that required resolution before cloud services would be accepted as a valid alternative to existing storage and compute methods.

Payment for cloud storage

The 'pay per use' funding model adopted by many cloud-based services makes it an appealing option for researchers who wish to store and/or process data over a short time period and have limited or no existing infrastructure to perform the activity. However, maintaining the data in the cloud beyond the project lifecycle is more problematic, but is being increasingly mandated research funders. Researchers would need to be aware of changes in pricing of cloud and migrate their data accordingly. There also needs to be clear guidelines for removing data once the retention period has expired. Frequently, departments were not aware of the cost of running local storage infrastructure. Hence they were unable to make satisfactory comparisons with other forms of storage.

Handling IPR and legal compliance issues

Data assets created and used by researchers may be subject to Intellectual Property Rights, legal compliance, or contractual requirements that specify the location where data assets may be stored and the security arrangement that must be made to prevent unauthorised access. Very often, researchers made use of cloud storage purchased on personal credit cards. Researchers often lack the legal skills to understand and interpret Service Level Agreements (SLAs) of commercial cloud providers, resulting in possible

legal breaches. Providing centrally managed infrastructures would facilitate more effective governance and legal compliance.

Trust

Many researchers perceived that their data were more secure if stored on portable drives or on local server infrastructure than it would be in commercial clouds. In fact many were uncomfortable with the concept of storing data in commercial clouds due to the possibility of loss of service or data, the difficulty of interpreting agreements with providers at an individual level and data protection issues.

Design and architecture

Design principles

Two main issues are addressed by Kindura. The first is to manage the allocation and replication of data across multiple storage providers including both in-house and external clouds, in order to mitigate for loss of service or data loss by individual storage providers, and ensure legal compliance. The second is to minimise the storage costs by using costing information and predictions, based on the information available from storage providers.

A fundamental design choice was to hide details about specific storage providers and services from the end user. Thus users specify the characteristics of the data they wish to deposit into the framework, which is then used to determine the number of replications of the content to be stored and the storage locations. However, users have no direct control over the absolute storage locations. This gives system administrators the flexibility to add and remove storage providers, and mirrors the situation with typical non-cloud based storage infrastructure.

End users are able to view the costs of storage of each of their datasets. Since hybrid cloud involves use of both internal infrastructure and on demand cloud storage services, this enables users to monitor all their costs of accounting purposes. However, we do not give the users an option to reduce their storage costs by directly changing the storage decisions made by the system, such as to reduce the number of replications of a dataset.

We separate the logic that determines in which content providers content should be stored and migrated from the other system components. Thus an institution can change and adapt the management of content to suit its own policies, without the need to rebuild the software. Changes to the rules might be made in response to a change in institutional or research funder policies.

The primary application of the Kindura framework would be to preserve completed work, such as the results of one or more experiments or simulations, or source datasets, as opposed to providing working storage. The basic deposit unit, termed *collection*, is defined to be a hierarchical set of files and folders, reflecting the typical structures used by researchers to store their experimental data. A collection would typically a homogeneous and related group of files such as a set of source observations or the results of one experiment of a number of related experiments.

User roles

Figure 1 illustrates the user interactions with the Kindura. Kindura inherits the roles of Producer, Consumer and Management from the OAIS[e] definition of a preservation system [23].

The primary producers of research data are researchers themselves, although in some cases, data may be generated and deposited automatically from experimental equipment. Researchers can deposit personal or shared collections and perform entry of associated metadata required for curation. Researchers can also track the costs associated with storage of their data at the collection level. Research collaborators are consumers of the preserved content and can search and browse the collections and download files. They have no permissions to view financial information or preservation metadata.

There are two management functions. The Technical Systems Administrator role is responsible for operation of the Kindura system, including the adding, configuration and removal of storage providers, and the setup of user accounts, and the maintenance of the cost models. Archivists play an indirect role. Researchers themselves are responsible for appraising and depositing content collections. Archivists determine the retention and replication requirements for specific categories of content, which are managed through the system rules.

Ingest and migration scenarios

External stakeholders form an additional set of management roles. Kindura maintains information about project storage budgets, which link with external finance systems. Storage provider administrators provide updates on costing information. Kindura supports a number of content management scenarios based on the requirements research described in the Case studies section.

Ingest involves upload of content to the system by users, together with appropriate preservation metadata, a decision making process to determine the number of copies of a content collection to be retained and the storage locations, and execution of preservation services.

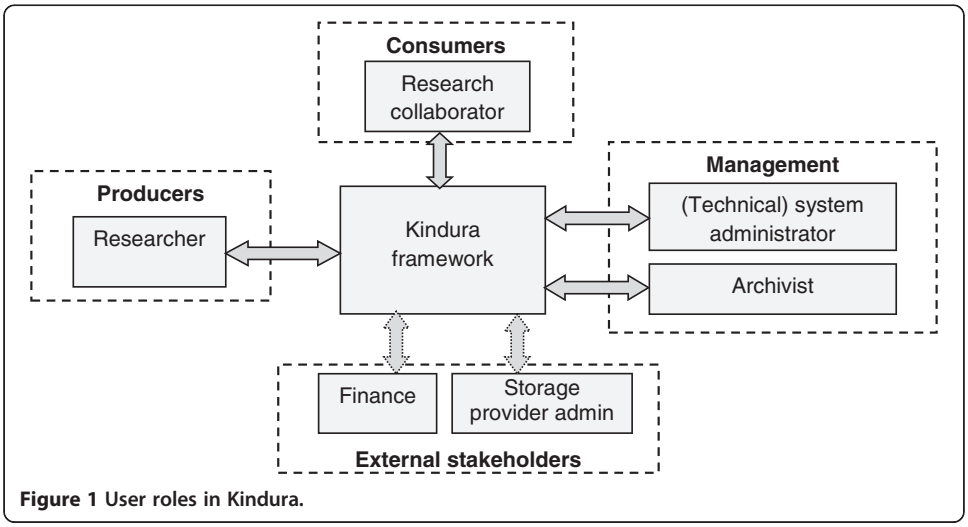

Figure 1 User roles in Kindura.

Data migration can be divided into three distinct scenarios. For this purpose storage providers are assumed to be grouped into tiers, reflecting their access speed, ranging from fast disc-based storage (high) to magnetic tape (low). *Down-migration* and *up-migration* are based on usage tracking of content collections and involve moving copies of content between storage providers in different tiers. *Cross-migration* involves moving content between storage providers in the same tier. This is to mitigate for loss of service by a given storage provider, restore corrupted content from a replica stored in another provider, move content from a storage provider that is to be deprecated, or move content to a new storage provider that has been added to Kindura, e.g. a more cost-efficient one.

Content retrieval involves selection of a copy of the collection from one of the available replications stored in Kindura. Retrieved content is transferred to the user via the Kindura server. A further scenario is to retrieve content to a separate cloud provider account owned by the user, for instance for the purpose of processing in the cloud.

Kindura architecture

Figure 2 shows the architecture of the Kindura, building on earlier work of the authors [24]. Kindura is based on a hybrid cloud combining internal storage and external clouds. Such an approach enables the flexibility to retain sensitive data in-house, whilst enabling less critical data to be stored externally. Connecting multiple storage providers

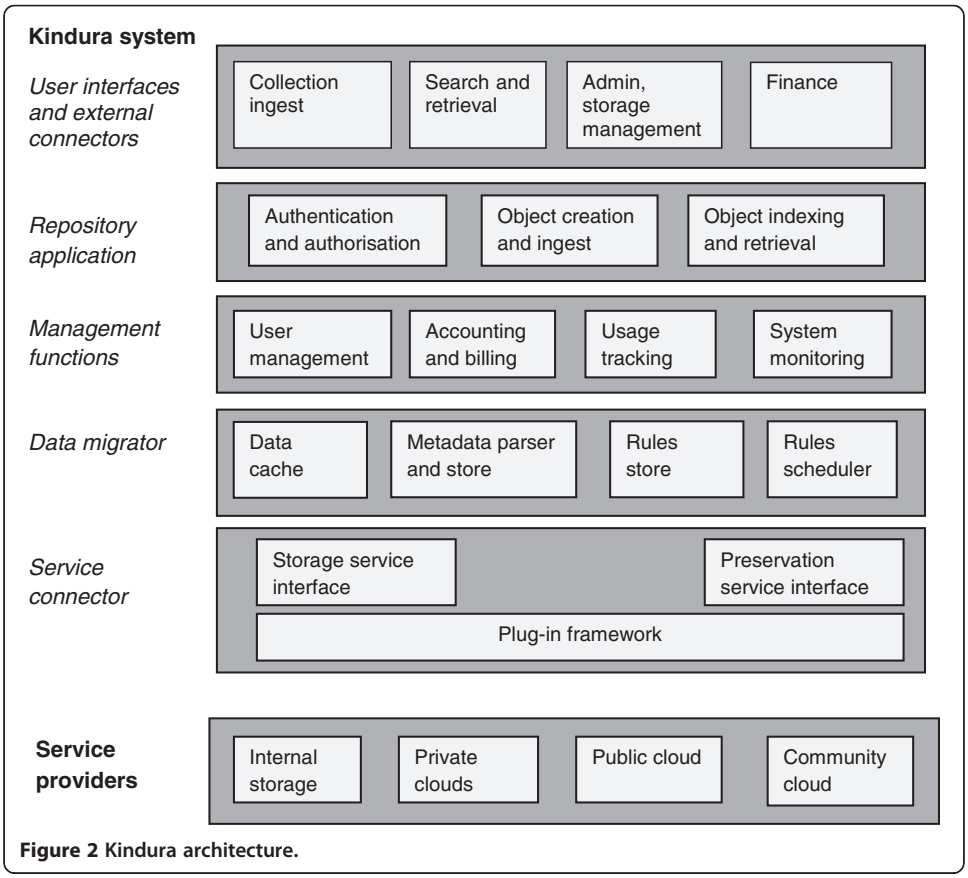

Figure 2 Kindura architecture.

to a single service connector enables centralised management, planning and billing for storage resources.

Service providers take the form of internal or external storage and compute services accessible through a CRUD (Create, Read, Update, Delete) REST API. Kindura provides a plug-in Service Connector layer to enable cloud service providers to provide a uniform interface to cloud resources including storage and preservation services. Such services include content replication and bit-integrity checking. Plug-ins adaptors essentially implement the uniform API using the provider API.

The Data Migrator layer manages the allocation of content collections to storage providers. A data cache is used for retaining data at ingest or temporarily storing data during migration. The management functions include user management, finance systems, usage tracking, both to determine migration requirements as well as for costing. System monitoring verifies the operational status of storage providers.

The Management Functions layer provides centralised user and account management, usage tracking and system monitoring.

The Repository Application layer provides tools for both ingest of data as well as indexing and retrieval functions. The User Interface layer provides a front end to allow users to deposit and retrieve content, and perform administrative functions. This layer also provides connectors to external systems, such as finance systems.

A key aspect of the overall framework is that the management of the data is handled by a Data Migrator layer that is abstracted from the Repository Application and associated Management Functions layers.

Implementation

Proof-of-concept system

A proof-of-concept system, based on the description in the Design and Architecture section was implemented and evaluated. This makes use of the Fedora Commons open source repository [5]. Fedora provides flexibility to define metadata schemas and content models to reflect the relationships between research data objects. Fedora enables three models of managing content, namely *managed content*, *externally managed content* and *redirect*. The latter two enable binary objects stored outside the local repository storage. In our implementation, we use the externally managed approach. Thus metadata was stored on the repository server and only binary content was stored in the cloud.

Upload uses a Java applet running in a web browser. The applet enables upload of folders containing multiple files and folders in a hierarchical structure. The file structure is modelled in the repository using Fedora objects linked by RELS-EXT[f] RDF relationships. Thus collections can be browsed as a file system, which was suitable for a large proportion of the researchers interviewed. Collection metadata is stored as a Fedora datastream[g] in the top level collection object. (One or more datastreams can be contained within a Fedora object and comprise containers for metadata). The top level collection object also contains datastreams with URIs linking to the copies of the associated binary content.

Kindura is piloting the DuraCloud [25] connector developed by DuraSpace. DuraCloud is an open source Java application[h]. (It is also available as a commercial

service from DuraSpace, connecting to a number of cloud providers). DuraCloud provides a CRUD[i] REST API, DuraStore, to enable interactions with cloud providers through a common interface. DuraCloud also provides common services interface, DuraService. This enables services such as content replication, content format transformation and bit-integrity services to be accessed, as well as the deployment of custom services.

Cloud providers are integrated into DuraCloud via a plug-in framework. Several plug-in adaptors are provided "out-of-the box" including Amazon S3[j], Rackspace[k] and Azure[l]. In the Kindura prototype system, internal storage is provided by the grid-based storage system iRODS [23]. A CASTOR datastore [26] has been interfaced to the iRODS server to provide tape storage at STFC.

DuraCloud currently provides the capability to create user accounts. However DuraCloud user accounts provide uniform access to all content in the system. Thus each storage provider is accessed via a single DuraCloud account with no direct access for end users. User authorisation in the Kindura system is therefore managed through Fedora. DuraCloud storage provider accounts are organised into containers, called *spaces*, analogous to buckets in Amazon S3.

The DuraCloud API currently does not support the entire provider-specific account configuration required by Kindura, such as selecting geographic regions for storage. Storage provider accounts are set up and configured manually. Limited configuration of storage provider accounts is possible through a properties file, which also holds the account authentication details.

Storage management

Figure 3 illustrates the implementation of the storage rules and cost optimisation process. Storage locations are determined at the collection level so that collections are not sub-divided across storage providers. Collection metadata comprises metadata entered by the user, described in more detail in the subsection Collection Metadata, and metadata automatically extracted from the collection such as the file sizes. For content that is already in the system, usage logs are available. For ingest, a usage estimate is entered by the user.

For each storage provider, a *Storage Provider Profile* file describes the capabilities of the provider and the usage costs. The storage provider profile is described in the subsection Storage Provider Profile. The rules engine has two purposes. Firstly it

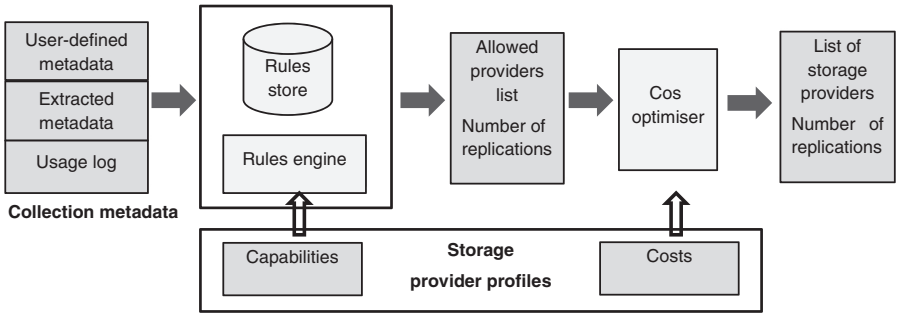

Figure 3 Selection of storage locations.

determines the number of replications required for a given content collection. Secondly it determines the allowed storage locations based on the collection metadata and the storage provider capabilities.

The second stage is the cost optimisation. If the length of the list of allowed storage locations is greater than the number of content replications required, the cost optimiser selects the most cost effective.

Physical architecture

Figure 4 shows the physical architecture of the pilot system. The DuraCloud server provides the Service Connector layer. The Repository Server provides basic repository functionality. The remaining components, including the Data Migrator and User Interface layer were implemented on the Kindura server.

The Kindura server, DuraCloud server, and repository server can all be run on separate virtual machines. The Kindura server provides a potential bottleneck, since it mediates all data transfers between the user and the data stores. Currently, complete data collections are cached on the server prior to being stored via DuraCloud. This is required to extract metadata on the size of the collection prior to running storage rules and cost optimisation. Thus the data cache needs to be a multiple of the size of the largest collection that is to be uploaded. A potential refinement would be to run tools on the user terminal in order that the collection could be uploaded in chunks.

In the prototype implementation, to simplify the development, uploaded binary content is transferred directly from the Kindura server to the DuraCloud server. The repository server handles only the metadata (Fedora objects), which contain references to the storage location of the content.

Data migration is currently mediated by the Kindura server. Since collection metadata is already available in the metadata store, large datasets can be transferred in smaller chunks.

Network bandwidth provides a limitation in moving large datasets, both between the user and the Kindura server and between the DuraCloud server and external storage providers, thus providing a limitation on the volume of content that can be effectively uploaded and managed.

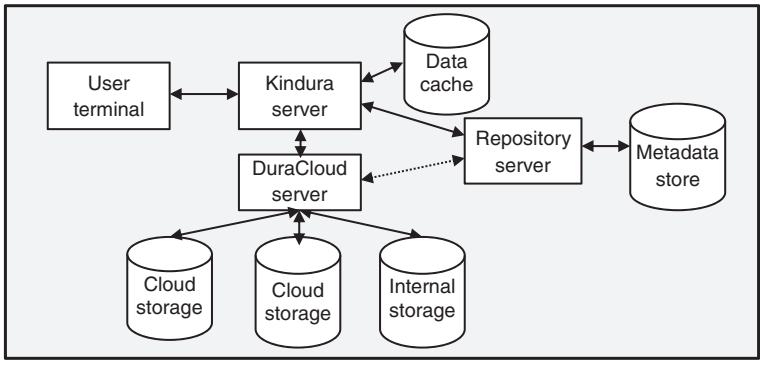

Figure 4 Physical architecture of the Kindura proof-of-concept system.

Metadata schemas

Collection metadata

The metadata entered by the user are entered in a web form and uploaded with the content collection. A set of relevant criteria was extracted from the data retention requirements described in the section Case Studies. These were then clustered into five categories of Ownership, Sensitivity, Usage, Provenance and Content Type, in order to consolidate them into as few criteria as possible.

In order to minimise the user workload in uploading content, all collections are associated with a project. Thus users must first create a project and enter the required metadata. In the proof-of-concept, we do not allow a collection to be associated with multiple projects, although this would be a desirable feature for users.

Project level metadata fields we abstracted implemented in the prototype are:

- Descriptive metadata. This includes the project name and a textual description of the project, which can be used to enable projects to be retrieved by free text searches.
- Ownership information, including the Principal Investigator and contact details.
- Administrative metadata, including the research funder and start and end dates of the project. These are required to determine the date until which the content must be retained.

Collection level metadata fields entered are:

- Descriptive metadata based on free text entry.
- The project to which the collection is associated.
- Protective marking, based on a classification of data and documents into predefined categories.
- An indication whether the collection contains personal information.
- Lifecycle information, which determines whether a collection contains source data, intermediate or published results.
- Access frequency, which is a figure estimated by the user based on the likely usage of the content. This figure is subsequently updated by the system based on actual usage statistics.

Additionally, users are able to select preservation services to be run at ingest such as format conversion. After upload to the server, further metadata such as file sizes are extracted.

Storage provider profile

We investigated various ways to represent the storage provider profile. Ruiz-Alvarez and Humphrey [27] propose a comprehensive XML schema for the description of storage provider metadata. Existing schemas already exist that could be adapted or extended for this purpose including the Resource Specification Language (RSL)[m] and the Job Submission Description Language (JSDL). The SoaML (Service oriented architecture Modelling Language) specification and the CloudML project[n] whose goal is to develop extensions to SoaML for the deployment of applications and services on cloud

for portability, interoperability and reuse, provide an alternative approach for describing storage resources.

Simplified storage provider profile – XML approach

The first approach we evaluated used the Ruiz-Alvarez and Humphrey XML schema [27]. The storage provider profile allows for a storage provider to offer multiple services, all described in a single XML file. In our cases, we used only one set of service configuration per storage provider account. An example would be for an XML file describing the Amazon S3 storage provider. S3 permits two levels of redundancy in storage of data, with selection being made at the point of bucket creation. The S3 interface for DuraCloud uses S3 buckets to implement the similar DuraCloud concept of spaces for data storage. Options for S3 bucket creation are held in a configuration file which is read at start-up of the service, meaning that creation-time selection of bucket options is not possible.

We use a simplified version of the schema to describe the storage providers installed into DuraCloud, thus permitting the programmatic determination of suitability of storage provider according to the collection metadata.

A single set of storage provider credentials can be shared by these different storage provider profiles with the only difference being in the parameters utilised in the access of that provider. These different services provided by the same underlying storage provider are represented as storage tiers in Kindura.

Furthermore, the XML schema permits the definition of geographical regions where data uploaded to a cloud storage provider will be physically held. Based on metadata provided by the user on upload of a data collection we constrain the selection of geographical regions in order to conform to legal limitations of where data may be kept.

Whilst XML profiles are suitable for machine processing, they are difficult for human users to edit. Since the costs and specifications of cloud providers are subject to constant change, it was essential that the profile could be easily edited by an administrator. Manually updating an XML profile was time-consuming, tedious and prone to error.

Simplified storage provider profile – spreadsheet approach

In order to facilitate more easily the manual editing of the storage provider profile, we developed a version of the storage provider profile based on the .xlsx spreadsheet format. The manual keying-in of the spreadsheet data is considerably easier and more accurate than that of manually entry in XML.

In Kindura's implementation the Xlrd[o] spreadsheet interpreter was used as a front-end to the rules interpreter used by the Drools[p] rules engine. The implementation of the rules is discussed further in the subsection Rules Implementation. An example of a Drools-formatted spreadsheet is shown in Figure 5. Conditions are contained in columns B to I. Note that the logical structure and interrelationship of the rules becomes readily visible by using the Excel feature to merge cells. This is turn encourages the user to make use of the cut and paste feature of Excel to improve the speed and accuracy of editing.

Collection of service costing information

The storage provider profile provides parameters relating to the costs of various interactions with the storage providers they describe. The use of these parameters permits

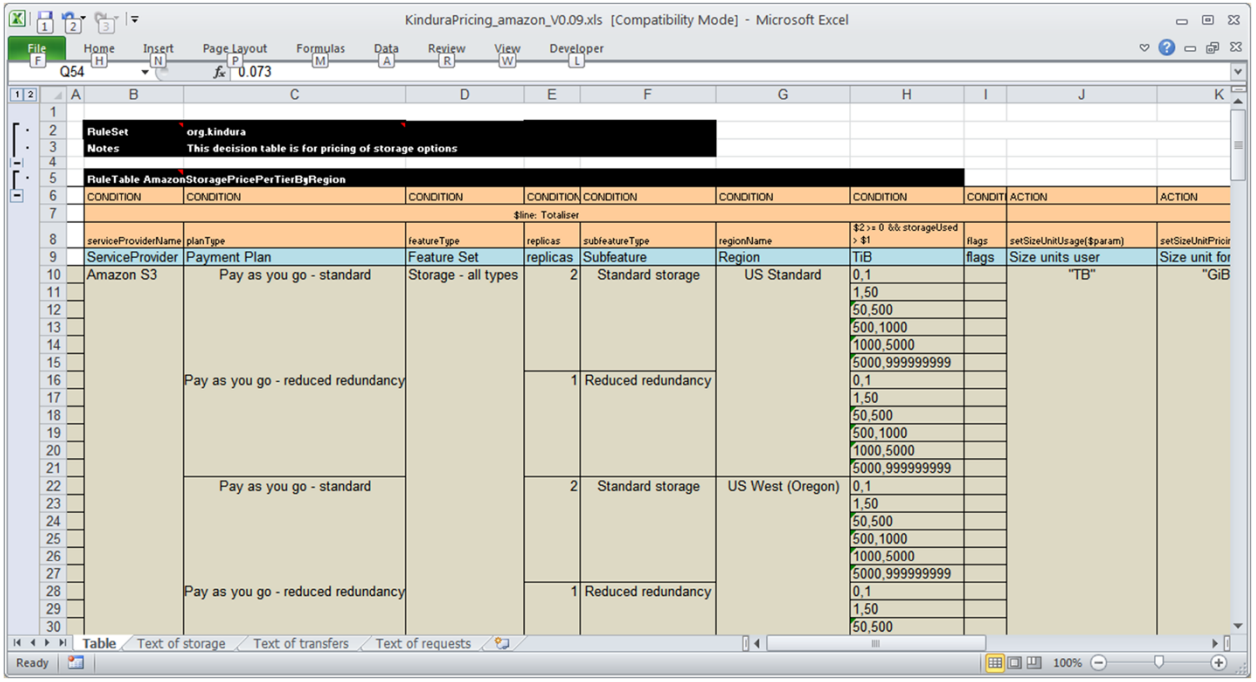

Figure 5 Conditions used in the spreadsheet interface for the Kindura implementation.

the calculation of the costs of moving the data into and out of storage providers, on the on-going cost of the static storage, both of which are of importance to the cost optimiser in Kindura.

A review of three major cloud storage providers (Amazon, Azure, Rackspace) showed no programmatic way to obtain costing information other than the error-prone extraction of data points from web pages. This presents a problem with respect to keeping costing accurate. It is quite possible that one service provider could alter their pricing and thus become more cost-effective than another when evaluating a dataset held within Kindura. We found no straightforward solution to this issue. Given the increased trend of reselling cloud resources by third party vendors, we would expect this issue to be addressed by cloud providers.

Kindura features a storage provider service implemented in iRODS. This provider is also described by a storage provider profile in the same fashion as with the cloud providers, meaning that direct comparison of relevant metadata is possible and cost-based optimisation can be used. Pricing figures were calculated based on internal estimates of infrastructure, power and staffing costs.

Business rules

Rule design

Based on the user, technical, funder and institutional requirements, a set of business rules were defined for managing the storage of the content. The rules-based approach can be used to implement policies such as

1. Collections containing personal information should not be stored in a storage provider based outside the EU (to comply with personal data protection legislation).
2. Collections containing intermediate or final results from EPSRC funded research projects should be retained for a minimum of 10 years.

The rules are run at content ingest, then periodically re-run to take account of potential changes in the available storage providers. This can be used to initiate data migration as described in the subsection Ingest and Migration Scenarios.

Rules implementation

We use a Java-based rule engine called Drools [28], originally developed as part of the JBoss application server system. Drools permits the definition of business rules in a file which can be loaded at runtime, thus allowing for changes to parameters and configuration without the need for recompilation and redeployment of application code. An example of a Drools rule is as follows:

```
// Define the appraisal periods for different funding situations
// First set of rules defines periods for project funding
rule "Institutional funder appraisal period"
salience 100
when
    collection: UploadCollection( project.funder == "Institutional" )
then
    collection.setAppraisalDate( 6 );
end
```

This rule specifies that if a project funder has been selected as being "Institutional" then there should be a six year appraisal period before the user should be prompted to decide whether the collection should be retained, removed or relocated to other storage.

The Drools rules operate directly on Java objects, which must be constructed and a reference to them inserted into an area of working memory instantiated by the Drools rule engine. Modified objects can then be used following the rule engine operation. Objects representing available storage providers are initialised using data points from the XML files described above, with specific data being located through the use of an XML stream event reader.

Cost model and implementation

Assumptions and definitions

The costing model assumes that complete cost information or cost estimates are available for each storage provider connected to the Kindura system. A storage service provider defined by parameters $(M; S; \alpha; \beta)$ where M is the bandwidth in bytes per unit time, S is the storage cost in monetary units per byte per unit time, α is the ingest cost in monetary units per byte, and β the cost in monetary units per byte in downloading data from the service provider.

Cost model

Let us say that the aim is to transfer B bytes to a given provider (where $B < 0$ means to download the data.). The cost of the transfer is αB for $B > 0$ and $-\beta B$ for $B < 0$ and the cost of persistently storing the full dataset is BST where T is the time interval during which the data are to be stored with this service provider. The time required for the transfer is $t_B = |B|/M + L$, where L is the latency of the communications network. The value of L may be a few milliseconds. Since in most cases the value of L will be negligible compared to $|B|/M$, we assume that $L=0$.

Discrete time intervals

The billing model that we have previously described by the parameter S is likely to be discrete in practice, i.e. it will involve a time unit Δt, such that the user is billed only at multiples[q] (or increments) of Δt. Thus, we can take the billing rate *as* $S_{\Delta t} := S\Delta t$, in units of monetary units per byte. We also define $B_{\Delta t} := B\Delta t$. If at time $t = k\Delta t$ the data volume stored with the provider is $B_{k\Delta t}$ then the user is charged $S_{\Delta t}B_{k\Delta t}$. At the next charging interval, $(k + 1)\Delta t$, the user is charged $S_{\Delta t}B_{(k+1)\Delta t}$, and the amounts are added together. In other words, the total sum becomes a step approximation to the integral $S\int B(t)dt$ or rather, the other way around, because the *actual* billing in this model is based on the step function. In the absence of knowledge about Δt, or for "small" values of Δt (compared to B/M), it becomes convenient to use the integral.

Consider a (small) time interval of $\delta t > 0$, during which a data volume of δb is transferred (i.e., if the full bandwidth is used, $\delta b = M\delta t$). Suppose that either δt is (much) shorter than the billing interval Δt, or in general, the time to move the data is not negligible (e.g. B/M is large).

In this case, the time $t_B := |B|/M$ required to transfer the file must considered; in particular we must consider also the cost of storing the data that are held by the provider while the rest of the data are transferred.

For *downloaded data*, i.e. $B < 0$, the model assumes that the data remains in the store until the downloading is completed. The cost associated with the transfer and storage during transfer is therefore

$$C_D(B) = St_B|B| - B\beta = \frac{SB^2}{M} - B\beta \tag{1}$$

For *uploaded data*, the data held by the storage provider at time t is $b(t) := Mt$ for $0 \le t \le B/M$. The storage cost incurred at time t over a time interval of length δt is therefore $Sb(t)\delta t$, and the total cost associated with the transfer is

$$C_U(B) := \alpha B + \int_0^{B/M} SMt\, dt = \alpha B + \frac{1}{2} S\frac{B^2}{M}. \tag{2}$$

In this simple model we assume that these parameters are constant, i.e. not only do they not depend on time, but they do not depend on the volume of data already held by the service provider. This model is, for most providers, valid up to a certain volume, $|B|$.

In the cost model above, it is assumed that the following operations do not incur any cost:

- Deleting the data from a provider;
- Using the data once they are stored with a provider.

If they do, this cost must be included in the overall costing of moving and using the data.

Using the cost model

Let $N > 0$ providers be given, and let $P_k = (M_k; S_k; \alpha_k; \beta_k)$, $k = 1,\ldots,N$. For convenience, we suppress the index k when the context makes it evident that the provider P_k is being referred to. Using the cost model, we can now calculate the cost of moving data to a provider, storing it, and moving it out (or deleting it) at the end. (It is assumed that the cost of deleting it is negligible; otherwise this cost must be included in the calculation of the overall cost).

We may consider a number of cases, including

1. Data are to be uploaded to the cheapest provider; the provider is chosen for which $C_U(B)$ is minimised.
2. Data are to be stored ($B > 0$) with a provider for a total time interval T, incurring a cost of $BST + C_U(B)$. (It is assumed that the data may be deleted at time $t = T$ and that the cost of deletion is nil).
3. Data are moved from one provider to another: if data is moved from P_m to P_n, the cost $C_{D_m} + C_{U_n}$ will be incurred for the transfer alone. Note that the transfer time (and hence cost) depends on the bandwidth M_{mn} from P_m to P_n.

Assume data are stored with a provider P_k and a new provider P_m is added, and it is known that the data must be stored for a length of time T (relative to the current time). For a data set of B bytes with $B>0$, the cost of download from P_k is $C_{D_k}(-B)$ and the cost of upload to P_m is $C_{U_m}(B)$. The cost of staying with the current provider is BS_kT.

Thus the total cost of moving it to the new provider will be (with $M = M_{km}$ and $B > 0$ throughout).

$$C_{k,m}(B) = C_{D_k}(-B) + C_{U_m}(B) + BS_m\left(T - \frac{B}{M}\right) \quad (3)$$

Here we may assume $T > B/M$ (otherwise there will not be time to transfer the data). Further, the data should be used from provider P_k when $0 \le t < B/M$ and with provider P_m when $B/M \le t < T$. (Here again we have assumed that the cost of using the data is nil. Otherwise a usage model will have to be combined with the storage model). If the overall cost in (3) is less than BS_kT, the data should be moved.

Content usage

We now update the model to incorporate usage of the content into the model. Let T denote the projected retention period of the content. Let λ denote the proportion of the content collection that is downloaded per unit time. For simplicity we assume that λ is constant, although in practice λ will vary with time. The parameter λ is estimated by users at ingest time, although at subsequent reappraisals of the content, it may be estimated from prior user transactions. The total volume of downloads of the lifecycle of the content storage is therefore λBT, and the content download cost $C_D(\lambda BT)$.

Hence when ingesting a content collection of size B bytes, it is necessary to evaluate the ingest cost $C_I(B)$ is the sum of the storage, upload and usage costs, given by

$$C_I(B) := BST + C_U(B) + C_D(\lambda BT) = BT(S + \lambda\beta) + B^2\frac{S}{M}\left(\frac{1}{2} + T^2\lambda^2\right). \quad (4)$$

Implementation

Three pieces of technology are used in the implementation: Drools[r], Xlrd[s] and Rete[t], all contained within the Drools package, which in turn is a callable routine of a Java application, the Cost Optimiser.

Drools builds a Rete tree of rules from the user input of price lists, features and service providers. Groups of "facts" are passed as Java objects to the Rete tree. If the group of facts match all the conditions that are relevant then an action is triggered and a set of values are passed to the calling Java program. The Java program builds its own tree of returned data but passes the data back to the Rete tree to prompt for further rule-set triggering that in turn creates new facts in a cascade four layers deep, to match the logical structure of the cost optimisation problem. Figure 6 shows the structure of the four layers.

Rules processing layer	Outputs of the rules processing layer
Layer 1	Length of time that the data must be held for Regulatory regions that the data may be stored in Frequency of access to the data Number of copies of data required
Layer 2	Cross-product translation of regulatory region to service provider datacentres
Layer 3	Constrained cross-product generation of available services and marketed products
Layer 4	Constrained cross-product lookup of multiple tiers of relevant prices

Figure 6 Outputs of the rules processing layers.

The cost is calculated using the tiered price per unit, versus length of time, versus amount of resources used (not only storage), versus pro-rata number of copies of data. The cost of entry into a service provider is typically priced at zero so has no impact on migration between service providers or on first-time ingestion. But the cost of exit from the old service provider can be significant and is included in the calculation for migration.

After all triggering of rule-sets within Drools has subsided the tree of data within Java is totalled by key-value, sorted and presented to the application. The application selects sufficient service providers from the sorted list to meet the target number of copies of data previously decided and returns a total cost to the user.

The architecture of Drools stands in contrast to that of name-value pairs in a database. A single rule-set will behave very similarly to a name-value lookup but the behaviour changes when multiple rule-sets are used. It is possible to easily add further rule-sets to the Rete tree that have very different input conditions to each other but the same structure to their output actions. This is comparable to the "polymorphism"[u] feature of modern programming languages and as a consequence Drools has a decision structure that is extensible.

A database name-value pair approach would require extensive re-programming work to achieve the same results as this behaviour. Re-programming becomes very costly in an environment of rapid change.

An example of this many-to-one "polymorphic" structure can be seen in use within the spreadsheet that carries the tiered prices (for example see Figure 5). Actual costs are calculated and accumulated from three dissimilar sources - storage, transfers of data and webserver protocol usage. Further sources could be added with trivial work.

Shown is an example in two parts. Figure 5 shows the conditions in the left hand part of the spreadsheet and Figure 7 the actions in the right-hand side of the same spreadsheet. The spreadsheet is read by Xlrd and translated to Drools rules format for input to the Drools rules engine. This spreadsheet is used to lookup exact prices given the facts shown in rows 10 onwards.

Each column J to T in Figure 7 is fired in sequence from left to right and calls a Java method in row 8 that sends data at the cell intersected by the triggered row and currently processing column to the Java program calling Drools. The action in column S is a dummy text used only to initiate an activity in the Java program, in this case the calculation and storage of a cost item using the data returned by the preceding cells. The action in column T is used as a debugging aid during development.

Evaluation

Methodology

The project carried out an evaluation of the Kindura proof-of-concept system with a sample group of four users. The main objectives of the evaluation were to assess the functionality and usability of the prototype, to answer a set of questions relating to technical issues addressed by the project (e.g. trust, user workload, payment models), to understand how useful the system would be within their working environment and determine how the prototype could be improved and extended.

The evaluation took the form of a structured interview with each subject, lasting approximately one hour. The subjects comprised a research project manager, a project data manager,

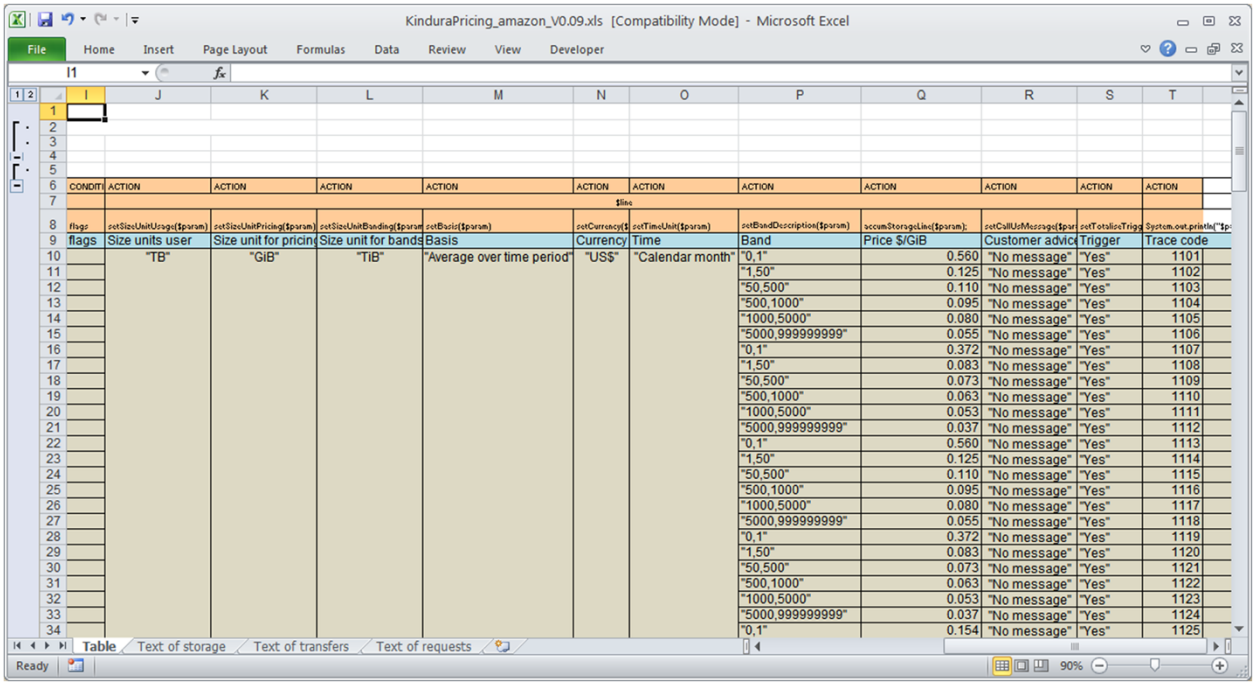

Figure 7 Actions resulting from the conditions in the spreadsheet interface for the Kindura implementation.

a researcher and an archive manager. Due to the time gap between gathering requirements and evaluation, it was not possible to use exactly the same sample group for evaluation as for the requirements gathering. The sample group included staff from both STFC and King's College London. The subjects were responsible for managing data at both large and small scales as well as across different academic disciplines.

Each interview began by explaining the purpose of the project, the types of data the user had to manage and their current provisions for preservation. The functionality of the system was described including upload of content and entry of associated metadata, retrieval of content and content migration. After this introduction, the interviewees were invited to perform a set of tasks with the prototype system covering the main implemented features. At each stage, their comments were recorded as well as final comments at the end of the set of tasks. Sample datasets were provided for the purposes of the evaluation. A further set of ten questions covering areas such as costs, security, metadata entry were then discussed. Users were invited to compare the Kindura approach to their existing solutions.

Findings

The research projects under the remit of the sample group created data at a rate of between tens of megabytes per day up to 10TB. The total volumes of data to be preserved ranged from several terabytes to multiple petabytes. Larger projects had in general made provision for long-term storage. Smaller projects tended to store data on a more ad hoc basis, on a variety of media such as local servers and portable devices.

Overall the Kindura system was seen to be most suitable for managing smaller projects due to the manual work required in entering the collection metadata, and the network bandwidth required to move data to external cloud providers. Public APIs for such a repository were thought to be an essential feature to enable reuse of the data. There is considerable interest in long-term storage of research data in the UK (and elsewhere) due to recent directives of funding councils discussed in the Introduction. One interviewee suggested the metadata entered regarding the data could be linked to a data management plan for the project itself.

The costing features were generally well received. Some users wanted the system to provide predictions of costs of storing the data in advance of it being uploaded, as well as billing for accrued costs. Detailed cost breakdowns including storage and network transfer costs were thought to be very useful for project budgeting. The system should also take into account currency fluctuations, since most cloud providers provide costs in US dollars, whilst Kindura converted costs to local currency. Providing a centralised repository resource with transparent costs was seen as attractive for monitoring storage costs at an institutional level.

There was some concern about the sustainability and maintenance overhead of the Kindura system, which comprises an additional layer of middleware between users and cloud storage providers. The Kindura architecture provides a modular approach, which can be assembled from industry standard components. Thus individual components could be exchanged and upgraded without compromising the overall functionality. The scalability of the framework in practical terms is determined by the capacities of the software components and the network bandwidth. Bandwidth constraints were seen as

the greatest concern, particularly around the data migration, and the latency of being able to download data.

The main motivation for placing data in a repository such as Kindura for most users was for backup or sharing. For sharing it was therefore essential to have transparent security that would enable users to control effectively with whom their data were shared. Support for data citation through a URI referencing the data was thought to be essential. This feature is supported by Kindura through the Fedora repository.

Many users already classify their data into different types such as source data and final results. This validated the Kindura approach. The main issue would be to align the terminology used by the Kindura system to a data retention policy within the hosting institution to ensure consistency of terminology.

The evaluation subjects were aware of the risk of storing data with external cloud providers over a long timeframe, particularly regarding loss of service. The replication of the data across multiple providers provided by Kindura was seen as an essential. The rules-based approach to managing the data was seen as a key value-add feature for Kindura over users interacting directly with cloud storage. It was viewed as more appropriate for administrators or archivists to deal with migration between storage providers over the longer term, since the researchers involved in creating the data may have left the institution.

Overall, Kindura was seen as a viable replacement to existing systems for managing smaller datasets of a large number of institutional users, rather than as a system for high-volume scientific work. Management and storage of high data volumes would require large bandwidth connections to the external cloud suppliers, which were seen as an issue. Due to the prototype nature of the Kindura software, system responses were quite slow, which may also have contributed to this feedback. The evaluation of the prototype against some of the key initial criteria determined in the initial requirements gathering are summarised in Table 1.

Conclusions and further development

The proof-of-concept system has demonstrated the feasibility of applying hybrid cloud storage to provide repository storage with policy-based data management across different providers and optimisation of storage costs.

The initial user study aimed to explore and capture the needs to researchers for repository storage of research outputs, with particular emphasis on research datasets in fields of data-intensive research across multiple disciplines. The study found a wide variation of current practices for management of research data, ranging from storage on portable media and local research group servers, to use of large scale infrastructures and in some cases cloud storage. Generally, smaller and highly distributed research communities ("small science") had made the least investment in data management infrastructure. These communities were also the most receptive to investigating the use of cloud storage.

The study identified and explored a number of issues in the long-term storage of research data relating to the use of cloud storage including trust, quality of service, cost-effectiveness, security and performance. As described the Prior Art in section, there is intensive on-going research in the design of cloud storage infrastructures,

Table 1 Summary of evaluation findings

Criteria	Evaluation
Trustworthiness (by end users)	Users were particularly keen that there data should be automatically replicated. This was seen as a major advantage over storing data on portable media and the perception of users was that their data would be safe.
Quality of service	The facility to have multiple copies of datasets dispersed across multiple physical locations and multiple commercial storage providers was positively received. Users were concerned that data stored in individual commercial cloud providers may be lost or become unavailable. Users were more concerned about knowing the hosting environment of their data rather than the exact physical locations.
Cost-effectiveness	The cost optimisation was seen to be useful, particularly if it provide to be more cost-effective than a rapid estimation of the storage costs by hand. Itemisation of the storage
Facilitate data reuse	The structured approach to entering metadata was seen as very useful to enable reuse, as well as the ability to share data with other users.
User workload	The classification of the data required by the system is already carried out by the users interviewed. The work required to enter metadata was not seen as a major burden, depending on the frequency and scale of data that is deposited.
Security	The security and access management of the system needs to be more transparent. Access logs were requested to monitor who had been downloading datasets. Sharing was seen as a key enabler of the system and users wanted to restrict access to certain subsets of their data.
Performance	In order to migrate and download large datasets, high bandwidth networks would be required. Hence the system was viewed as more suitable for smaller datasets, with potentially a large number of users.
Operational issues	System administrators were concerned that the Kindura system would represent an additional layer of mission-critical middleware between users and cloud storage. Thus the value-add features such as cost optimisation and management of replication would need to offset this cost.

particularly relating to trust and quality of service. The approach taken by Kindura was to design, build and evaluate a practical proof-of-concept based on existing cloud and grid storage technologies. Rather than considering the design and implementation of clouds themselves, we integrated existing cloud and grid technologies into a hybrid model, taking a "black box" approach. We then investigated how the issues raised around the use of existing infrastructures could be mitigated through the use of an additional management layer. This middleware firstly enabled rules to be created to determine where and with what degree of replication data should be stored within the hybrid cloud. Secondly, cost optimisation was implemented to enable minimisation of storage costs across the available storage providers. The rules and cost optimisation framework enable configuration to suit different operational requirements without the need for recompilation.

The evaluation was carried out with researchers, as well as IT staff and an archivist. Due to the automated management of replication, researchers expressed confidence that their data would remain accessible in the Kindura system. Although data replication was handled automatically by the management layer, an important principle was that transparency is important for building trust with researchers. Concerns were expressed by IT staff about the dependency on the Kindura management layer, which would be a critical for the operation of the system. However the value could be demonstrated through savings due to optimising the storage costs. Indeed, there is a trend of increasingly complex pricing models for commercial cloud storage [21], which makes comparison by hand increasingly impractical.

The combination of pay-per-use storage with existing institutional storage highlighted an important governance issue. Typically IT infrastructure at research organisations is provided through top-slicing of departmental budgets. The provision of storage costs for users suggests a model for charging for IT services more in line with usage, where users are charged according to the volume of data they store and transfer.

The Kindura system was a proof-of-concept and many potential improvements and extensions were identified.

The system could be improved by providing intuitive administrative and management interfaces. Editing of storage rules could be better integrated into the Kindura user interface, rather than requiring the editing of separate XML files or spreadsheets as at present. This would make the administration easier for archivists and other less technical staff.

The storage rules are currently applied on a case-by-case basis, without taking into account the availability of space in internal storage providers. Storage occupancy could be a weighting factor used by the cost optimiser that could assist in providing macro-storage management.

Further developments to the DuraCloud service connector, to enable storage provider account configuration to be entered through the Kindura user interface would simplify the setting up of storage provider accounts. Direct integration of the Fedora repository with DuraCloud, by means of providing an Akubra[v] plug-in would desirable, rather than accessing data indirectly through links (c.f. Figure 4).

A number of migration scenarios were identified during the requirements gathering that could not be fully implemented during the project due to time constraints. These include:

1. Migration of content with low access requirements to lower tiers of storage (e.g. transfer from disc to tape stores),
2. Automatic migration of content due to addition of removal of providers or changes in storage policies.
3. Manual migration of content by users to make a copy of content available in a specific storage provider, for instance for processing in the cloud.

For each of these options, the basic functionality for transferring the data within the hybrid cloud system is available within the proof-of-concept system. The main additional steps would be to add to the management and user interface layers. For point 1, usage tracking could be used as one mechanism to determine whether content was no longer required to be stored on high storage tiers. For point 3, creating a copy of the data within the host cloud provider is more cost-effective than exporting and reimporting the data, should a copy already be available. All data access is currently through Fedora and the only way to use S3 native protocols is to use a tool not part of Kindura. Direct access via S3 protocols would make the data difficult to navigate since the S3 names are UUIDs combined with our given names.

Intelligent migration scheduling could be incorporated into the rules framework, making use of network monitoring, so that large data migration tasks could be scheduled at times of low demand to reduce the impact on users of the network.

Endnotes

[a]Science and Technology Facilities Council: http://www.stfc.ac.uk

[b]http://www.jisc.ac.uk/.

[c]http://www.jisc.ac.uk/whatwedo/programmes/flexibleservicedelivery/kindura.aspx.

[d]National Health Service (UK)

[e]Open Archival Information System: http://en.wikipedia.org/wiki/Open_Archival_Information_System.

[f]https://wiki.duraspace.org/display/FEDORA36/Fedora+3.6+Documentation.

[g]See https://wiki.duraspace.org/display/FEDORA36/Fedora+Digital+Object+Model#FedoraDigitalObjectModel-Datastreamsdata.

[h]https://wiki.duraspace.org/display/DURACLOUD/DuraCloud+Downloads

[i]Create, Read, Update, Delete: http://en.wikipedia.org/wiki/Create,_read,_update_and_delete

[j]http://aws.amazon.com/s3/

[k]http://www.rackspace.co.uk/

[l]http://www.windowsazure.com/

[m]http://www.globus.org/toolkit/docs/5.0/5.0.0/execution/gram5/pi/

[n]http://aws.amazon.com/cloudformation/

[o]http://pypi.python.org/pypi/xlrd

[p]http://www.jboss.org/drools/

[q]If not known, or published by the provider, Δt can in principle be measured by transferring a known data volume and see what the incurred costs is. For example, if we transfer a data volume $B>0$ (with $B/M< \Delta t$), and delete it after time T', if the billed amount is an integral multiple of SB, say kSB (after subtracting the transfer costs αB of course), with k integer, then $k\ \Delta t<=T'<(k+1)\ \Delta t$. Experimenting with multiple values of T', we will be able to derive values for Δt.

[r]http://docs.jboss.org/drools/release/5.3.0.Final/drools-expert-docs/html/ch01.html

[s]http://pypi.python.org/pypi/xlrd and http://www.python-excel.org/

[t]http://en.wikipedia.org/wiki/Rete_algorithm

[u]http://en.wikipedia.org/wiki/Polymorphism_in_object-oriented_programming

[v]https://wiki.duraspace.org/display/AKUBRA/Akubra+Project

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
All the listed authors made substantive intellectual contributions to the research and manuscript. Specific details are as follows: Waddington: Kindura project lead. Responsible for the overall technical approach and architecture, editing and preparation of the paper. Contributed to requirements gathering, analysis and evaluation. Zhang: Performed design and development of the prototype system. Knight: Led the work on requirements gathering. Jensen: Lead of the STFC contribution, including contributions for the technical approach and architecture. Contributed to requirements gathering and evaluation. Carried out the theoretical work on cost modelling (section subsection Cost Model). Downing: Design of the rules engine. Contributed to requirements gathering Ketley: Contributed to the design and implementation of the rules engine and cost optimization. Contributed to evaluation. All authors read and approved the final manuscript.

Author details
[1]Centre for e-Research, King's College London, 26-29 Drury Lane, London WC2B 5RL, UK. [2]Science and Technology Facilities Council, Harwell Oxford Campus, Oxon OX11 0QX, UK.

Received: 23 January 2013 Accepted: 16 May 2013
Published: 15 June 2013

References

1. Open Access agenda. http://en.wikipedia.org/wiki/Open_access. Accessed 15 Oct 2012
2. (2011) EPSRC Policy Framework on Research Data. http://www.legislation.gov.uk/ukpga/2000/36/contents. Accessed 15 Oct 2012
3. (2011) NERC Data Policy. http://www.nerc.ac.uk/research/sites/data/policy.asp. Accessed 15 January 2013
4. Digital Curation Centre Resources for Digital Curators. http://www.dcc.ac.uk/resources/external/tools-services. Accessed 10 Apr, 2013
5. Mell P, Grance T The NIST Definition of Cloud Computing. http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf. Accessed 11 November 2012
6. (2011) Kindura JISC FSD Programme case study. http://jiscinfonetcasestudies.pbworks.com/w/page/45197715/Kindura. Accessed 12 Aug 2012
7. CMDI standard, SNIA CMDI standard, SNIA. http://www.snia.org/cdmi. Accessed 11 Nov 2012
8. OCCI working group Open Grid Forum (OGF). http://occi-wg.org/. Accessed 22 Jan 2013
9. OpenNebula. http://opennebula.org/. Accessed 15 Jan 2013
10. (2008) RESERVOIR - An ICT Infrastructure for Reliable and Effective Delivery of Services as Utilities, IBM Technical Library H-0262. http://domino.watson.ibm.com/library/CyberDig.nsf/papers/A44F6256BB697FCE852574E10052DDEE. Accessed 15 October 2012
11. (2010) EC FP7 Contrail project. http://contrail-project.eu/. Accessed 15 Jan 2013
12. Aschenbrenner A, Zhu B (2011) iRODS-Fedora Integration. http://www.irods.org/index.php/Fedora. Accessed 15 Jan 2013
13. Pcolar D, Davis DW, Zhu B, Chassanoff A, Hou CY, Marciano R Policy-Driven Repository Interoperability: Enabling Integration Patterns for iRODS and Fedora. http://www.researchgate.net/publication/228943451_Policy-Driven_Repository_Interoperability_Enabling_Integration_Patterns_for_iRODS_and_Fedora. Accessed 15 Jan 2013
14. Hedges M, Blanke T, Hasan A (2009) Rule-based curation and preservation of data. Future Generation Computer Systems 25:446–452, http://dx.doi.org/10.1016/j.future.2008.10.003
15. Corney D, Downing R, Folkes T, Griffin T, Hawker K, Hunter A, Jensen J, Moreton-Smith C, Norman K, Sharpe R, de Witt S (2011) Preserving ISIS Data, All Hands Meeting 2011, Poster Session. http://www.allhands.org.uk/
16. Greenwood D, Khajeh-Hosseini A, Smith J, Sommerville I (2010) The Cloud Adoption Toolkit: Addressing the Challenges of Cloud Adoption in Enterprise. http://arxiv.org/pdf/1008.1900. Accessed 15 Jan 2013
17. ISO 16363:2012 Audit and Certification of Trustworthy Digital Repositories (2012) International Organisation for Standardisation. http://public.ccsds.org/publications/archive/652x0m1.pdf. Accessed 10 Apr, 2013
18. ISO 16363:2012 Space data and information transfer systems -- Audit and certification of trustworthy digital repositories. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=56510. Accessed 10 Apr, 2013
19. Mahbub Habib SM, Ries S, Mühlhäuser M (2010) Cloud Computing Landscape and Research Challenges regarding Trust and Reputation. Proceedings IEEE 2010 Symposia and Workshops on Ubiquitous. In: Autonomic and Trusted Computing. pp 410–415. doi:10.1109/UIC-ATC.2010.48
20. Bianco P, Lewis GA, Merson P (2008) Service Level Agreements in Service-Oriented Architecture Environments (CMU/SEI-2008-TN-021). Software Engineering Institute, Carnegie Mellon University, http://www.sei.cmu.edu/library/abstracts/reports/08tn021.cfm. Accessed 10 Apr, 2013
21. Zhou M, Zhang R, Xie W, Qian W, Zhou A (2010) Security and Privacy in Cloud Computing: A Survey. Proceedings IEEE Sixth International Conference on Semantics, Knowledge and Grids 2010:106–112. doi:10.1109/SKG.2010.19
22. The Open Provenance Model (OPM). http://openprovenance.org. Accessed 15 Jan 2013
23. Reference Model for an Open Archival Information System (OAIS) (2009) Draft Recommended Standard, *CCSDS 650.0-P-1.1 (Pink Book)*, Issue 1.1. http://public.ccsds.org/sites/cwe/rids/Lists/CCSDS%206500P11/Attachments/650x0p11.pdf. Accessed 15 Jan 2013
24. Jensen J, Downing R, Waddington S, Hedges M, Zhang J, Knight G (2011) Kindura – Federating Data Clouds for Archiving Proc. Int'l Symp. on Grids and Clouds 2011. Academia Sinica, Taipei, Taiwan, http://pos.sissa.it/archive/conferences/133/039/ISGC%202011%20&%20OGF%2031_039.pdf. Accessed 13 June 2013
25. DuraCloud, DuraSpace. http://duracloud.org. Accessed 15 Oct 2012
26. CASTOR – CERN Advanced Storage System. http://castor.web.cern.ch/. Accessed 15 Jan 2013
27. Ruiz-Alvarez A, Humphrey M (2011) An Automated Approach to Cloud Storage Service Selection. ScienceCloud'11, San Jose, California, USA
28. Drools rules engine. http://www.jboss.org/drools. Accessed 15 Jan 2013

doi:10.1186/2192-113X-2-13
Cite this article as: Waddington *et al.*: **Cloud repositories for research data – addressing the needs of researchers.** *Journal of Cloud Computing: Advances, Systems and Applications* 2013 **2**:13.