Preservation Environments for Digital Entities

Reagan W. Moore (San Diego Supercomputer Center)
William Underwood (Georgia Tech)

Abstract

The long-term preservation of electronic records has many similarities with the processes used for paper records. However, an examination of the approaches needed to preserve electronic records identifies significant differences. The concepts that support the preservation processes for digital entities build upon the ideas used in data grids, which are software systems that allow for the organization of digital data distributed across multiple locations and storage systems. In particular, preservation of electronic records, in addition to ensuring integrity and minimizing risk due to catastrophes, requires the management of the evolution of the underlying storage systems. One approach is to convert electronic records to a preservable format, register the electronic records into a logical name space, replicate the records for security, manage the records with their associated metadata to protect their integrity and to maintain their accessibility, generate archival description and retrieval instruments, where retrieval instruments include archival indices and location registers, and manage technical metadata. The mechanisms provided by data grids for interoperability across data management systems encompass part of the required infrastructure. The ability to automate the application of archival processes for the generation of the preservable format is supported by data grids. This paper specifies the minimal data grid capabilities that are needed to enable the creation of persistent archives holding records for time periods longer than the life of the underlying software technologies.

Contents

1. Preservation Concepts

The archival community has developed standard approaches for accessioning, preserving, and accessing paper records. These approaches are expressed using terms that focus attention on the provenance of the records, the processes by which the records are created, the organizational structure imposed on the records by their creator, and the presumption of the authenticity of the records transferred to the archives by their creator. The preservation processes seek to maintain the organizational structure (i.e., classification) and the association of identity metadata (i.e., indexes) with the records, while providing descriptive instruments for the discovery, retrieval and access to the material. Paper records are organized in structures that break down the records of each creator, that is, each archives, or archival fonds (the term used varies from a culture to another) into record series, subseries and files, respecting the way the records are aggregated in the course of the affairs of which they are the byproduct. Descriptive metadata about these record aggregations are organized in an inventory to support browsing and discovery. With paper records, it is assumed that storage methods and access methods do not change in time. They are retrieved from a physical container, and used by reading the language in which they were written. Archival terms express concepts that are historically linked to the characteristics and methods of preservation of paper records.

Electronic records are created in a digital environment in which traditional archival terms and the concepts they express are no longer sufficient to describe their creation and management. In addition to attributes associated with the creation of the record, attributes are needed to describe the digital technology used to create the records, that are used to store them, and that are used to access them, because digital technology evolves over time. The mechanisms that are available today to read an electronic record may not be able to correctly interpret the encoding format used to write the record a few years from now, and the storage system that holds the bits comprising the digital entity may require by then an access protocol that is no longer supported by the current computer operating system. This implies that the preservation of electronic records requires new terms expressing concepts that can describe the management of technology evolution from the moment of creation to that of the transfer to the archives to the integration of new technology within the preservation environment. This is the most significant addition to the body of archival knowledge that is needed to enable archivists to preserve digital entities.

Digital environments do have some advantages over paper environments. Bit-for-bit identical copies may be made of a digital entity and stored at multiple locations. This greatly minimizes the risk of loss due to natural disasters, as long as the location of the duplicate is known. This means that the location of each electronic record and of each duplicate now must be maintained. The concept of a digital collection is used to describe the management of technical information about the storage of electronic records. The technical information includes attributes that were originally properties maintained by the storage system used by the record creator, such as size, owner, creation date. When the records are extracted from their creation environment, the associated technical information is also extracted and managed in the digital collection. The concept of integrity is used to assert that the bits in the copies have remained the same. In digital environments, the movement and storage of records may result in the introduction of errors. An integrity assertion mechanism is needed to validate the records.

In a digital world, anyone with sufficient access privileges can change a digital entity. For paper records, the authenticity is presumed to be established by the creator. For digital records the authenticity now depends upon evidence that persons with the ability to modify the records did not. The authenticity of a digital entity presumes that the set of persons who were given write access only executed approved preservation processes. The concept of audit trails (the tracking of all accesses to data) is used to identify all persons and the dates on which they manipulated a record. The concept of security (authentication of individuals) is needed to control access.

Distinguished user names are created for each archivist who is given access. The combination of distinguished user names and audit trails makes it possible to identify all records that were manipulated by a particular archivist.

Electronic records have an internal encoding format that is imposed by the application used to create the digital entity. The encoding format may be transformed into a different encoding standard through the use of transformative migrations. Conversion programs exist that transform text from proprietary encoding formats to published international standard encoding formats. The choice of encoding format is therefore at the discretion of the archivist. The selection of an encoding format for an electronic record by an archivist is equivalent to the specification of an archival format for electronic records.

In the digital world, the record as created may contain a virus or worm that changes the environment on which the digital entity is examined. The virus or worm consists of bits that are usually added without the knowledge of the creator. Each electronic record has the power to change the access environment. Electronic records can be examined to determine whether or not they contain a virus or worm, and the added bits can be removed. However this requires examining every electronic record as a unique digital entity. This requirement imposes the second most significant change in concepts for preserving electronic records. Digital preservation must be done at the digital entity level. Every electronic record must be examined to see if it will corrupt the environment in which it will be accessed. Every electronic record must be examined to see if it conforms to the desired archival format for preservation.

The data grid community has developed concepts and terms that characterize technology evolution, replication, security, and management of digital entities. This paper provides a bridge between the archivist terms that focus on the provenance of records, and the data grid terms that focus on the management of digital entities. Multiple examples are given below to identify the differences in terminology between the preservation and grid communities:

- Persistent archives. Archivists think of an archives as the whole of the records made or received by one person or organization (the creator) in the course of practical activities. A persistent archives is an archives the content of which is preserved over time. The data grid community thinks of an archive as the storage system that holds the digital entities comprising all the records that are being preserved. A persistent archive is the software infrastructure that manages the preservation of the digital entities while the underlying storage systems are replaced with more modern technology. In practice, a persistent archive is both the archives of a creator and the technology needed to support the preservation of digital records.
- Collections. Archivists think of a collection as an artificial accumulation of documents brought together on the basis of some common characteristic, without regard to the provenance of the documents. They think of an archives as a body of organizationally related records linked together by the purpose that they fulfill. They think of a record series as an aggregation of documents resulting from a function of the creator. The data grid community views a digital collection as a mechanism for imposing a logical organization on digital entities, regardless of their physical location. An attribute can be associated with each digital entity to characterize its actual storage location. A view can be imposed on the digital collection to present all of the digital entities composing an archives, even though the digital entities may be distributed across multiple sub-collections. In the digital world, the organizational structure used to manage the digital entities is decoupled from the characterization of the digital entities.
- Administrative metadata. Archivists use archival description to describe the organization that produced the electronic record. The data grid community uses technical administrative metadata to describe the management of the electronic record. The technical administrative metadata include the storage location, the size of the electronic record, the custodian of the electronic record, the date the electronic record was created, access control lists for who is allowed to manipulate the electronic record, and audit trails

for tracking accesses. The technical administrative metadata are stored as attributes in the digital collection that is used to organize the digital entities.

- Descriptive metadata. Digital collections impose a virtual context on digital entities that are registered into the collection hierarchy. The virtual context can include attributes that describe the provenance of electronic records, and attributes that describe characterizations of the digital entities that are registered by the archivist. Additional attributes can be added at any point in time. The attributes can be mined from each digital entity, or acquired by processing ancillary material.
- Logical name space. Archivists specify an identifier for a digital entity as a member of an archives and record series. In the digital world, a physical location identifier is created for each storage system on which the digital entity is stored. Unfortunately, the physical location identifier depends upon the type of storage system. To provide persistent identifiers for digital entities, a logical name is associated with each digital entity. The physical location identifier is mapped onto the logical name as an attribute that is managed in the digital collection that is used to organize the digital entities. The logical names remain invariant as the digital entity is migrated across storage systems.
- Folders. A folder is a container for a number of documents. The data grid community uses containers (aggregations of multiple files into a single file) to achieve the same purpose. Note that a container is a physical aggregation of digital entities for storage. The digital entities within a container may be members of different sub-collections within the digital collection hierarchy.
- Preservation processes. The processing steps that are applied to digital entities can be automated through the use of dataflow or workflow systems. It is possible to create procedures that examine each digital entity, assign a logical persistent name, validate the encoding format, create integrity information, extract descriptive metadata, aggregate the entities into containers for physical storage, store the containers, create the associated technical administrative metadata, and register the preservation metadata into a digital collection. These steps can be mapped to the traditional preservation processes of appraisal, accession, arrangement, description, preservation, and access.

The preservation of electronic records is of interest to all communities managing digital entities, from archivists to computer scientists, to librarians. The fundamental issue is the preservation of the authenticity of electronic records while the technology in the supporting infrastructure evolves. Whenever digital entities are preserved for time periods greater than the lifetime of technology, mechanisms are required to support migration to new technology. The challenge of technology evolution must be understood and managed for long-term preservation to be achievable.

2. InterPARES Preservation Model

The InterPARES Preservation Model is defined in the report “The Long-term Preservation of Authentic Electronic Records: Findings of the InterPARES Project” in Appendix 5, http://www.interpares.org/book/index.cfm .

The report specifies the activities associated with ensuring the preservation of authentic electronic records. A *preservation action plan* describes the preservation actions to be taken for the transfer of records to the archives, for accessioning the records, and for maintaining the records. Preservation actions are implemented using preservation methods. The methods rely on software for generic preservation procedures such as integrity checks, methods for packaging or archiving many files as one, for refreshing media, for database management, and for archival storage. They also include specific preservation methods, for example, for reproducing records, and for converting proprietary formats to standard formats. A digital entity that has been converted to the chosen archival encoding format can be said to be a persistent object.

In this paper, we examine the mapping from the operations required by the InterPARES Preservation Model to the capabilities provided by data grids. Data grids are software infrastructure that manage digital entities that are distributed across multiple storage systems. Data grids provide interoperability mechanisms for accessing multiple types of storage repositories, information repositories, and authentication environments. Data grids also provide persistent naming conventions that permit the migration and replication of digital entities onto new storage systems. The actions in a preservation action plan trigger methods associated with an activity. The actions can be implemented through processes executed at the storage repositories that are managed by a data grid.

An example of a data grid is the San Diego Supercomputer Center Storage Resource Broker (SRB). The SRB data grid uses a digital collection to register provenance and administrative metadata for each digital entity within an archival fonds or record series. Technical administrative metadata needed to describe the storage location, the existence of replicas, the assignment of access controls, and the integrity of each digital entity are also maintained. Technology evolution is managed through the specification of a set of standard operations for accessing digital entities. Software drivers are provided for mapping from the standard access operations to the access protocol required by a particular storage system. Access interfaces are provided to control the migration of digital entities between storage systems while maintaining the consistency of the technical administrative metadata that are managed in the digital collection.

The lowest-level processes in the InterPARES model specify explicit capabilities that can be supported with data grid technology. We present an overview of archival processes, the capabilities provided by data grids, an assessment of alternate preservation approaches, and a characterization of each of the principal data grid capabilities.

3. Persistent Archive Description

Preservation environments manage technology evolution. They provide the software abstractions needed to preserve an archival aggregation while the underlying software and hardware technologies evolve. Preservation environments also provide the mechanisms that enable automation of archival processes. Finally, preservation environments support multiple approaches towards management of archived material, including both emulation of software infrastructure, characterization of the structure, semantic labels, and behavior of digital entities through use of digital ontologies, and migration of digital entities to new encoding formats.

A preservation environment that is based upon data grid technology is called a *persistent archive*. We apply the term “persistent” to the concept of an archive to represent the management of the evolution of the software and hardware infrastructure over time. In this report, we describe the capabilities that are needed by preservation environments to automate both the management of technology evolution, and the application of archival processes by archivists. We note that the terminology used by the preservation community conflicts with the terminology from the data grid community. Throughout the paper, references are made to the management of digital entities in persistent archives. Archives, within the data grid, refer to the storage systems used for long-term storage. Archives, as used by archivists, are the organized non-current records of an institution. We use the term digital entity, to represent any sequence of bits that constitute an entity that will be preserved. Records, as the term is used by archivists, are a class of digital entities with unique attributes and constraints. Records, in addition to their data bits or content, also require metadata to describe their context (juridical-administrative, provenancial, procedural, documentary, and technological) and prove their authenticity (identity and integrity related metadata, and documentation that demonstrates the processes carried out by the archivists). This metadata defines the preservation context of the records. We require the ability to guarantee the integrity of the preservation context and the correct association of the preservation context with each electronic record. Since the technical administrative metadata changes over time, we need mechanisms to guarantee the consistency of the technical administrative metadata. All manipulations of the electronic records must be tracked and the associated technical administrative metadata updated. The preservation context is the result of the application of archival processes to the records, and is carried on through time by the continued application of archival processes.

A persistent archive provides the mechanisms needed to manage technology evolution while maintaining consistency between the preserved electronic records and their archival context. During the lifetime of the persistent archive, each software and hardware component may be upgraded multiple times. The challenge is creating an architecture that maintains the integrity of the archived documents while minimizing the effort needed to incorporate new technology. Fortunately, data grids provide the infrastructure needed to manage technology evolution. Persistent archives can be based on data grids. Both systems rely on interoperability mechanisms to support access to heterogeneous types of storage systems and information repositories, while supporting the re-creation of derived data products. A derived data product is the result of the application of a process to a set of digital entities. The process can be an archival process, such as the migration of a document to a new encoding format, or the wrapping of an application for execution on a new operating system, or the registration of digital entities into a digital collection. The persistent archive research group of the Global Grid Forum is examining how persistent archives can be built from data grids. Data grids are distributed systems that tie together data management systems and computer systems. Virtual data grids are differentiated from data grids by the ability either to access derived data products or to re-create the derived data products from a process description. Virtual data grids support the application of processes to create derived data products. In the context of persistent archives, virtual data grids are capable of applying the archival processes needed to manage the organized non-current records of an institution.

A persistent archive maintains not only the data bits comprising digital entities, but also the context that defines the provenance, integrity, and structure of the digital entities. The context, from the perspective of the data grid, is managed as attributes that are organized into an authoritative catalog. We use the term archival form of a digital entity to represent the data bits, a definition of the structure of the digital entity, and the associated preservation attributes. For example, the Open Archival Information System (OAIS) specifies an Archival Information Package (AIP) for defining the context of a digital entity. A digital collection is the registration of the archival forms of an aggregation of digital entities into an authoritative catalog.

Data grids provide a logical name space into which digital entities can be registered. The logical name space is used to support global, persistent identifiers for each digital entity within the context of each digital collection. The digital entities are represented by their logical name, a physical file name, and, if desired, an object identifier that is unique across digital collections. Data grids map distributed state information onto the logical name space for each grid service. Examples of state information are replica information for the location of each physical copy of a digital entity, descriptive metadata that is associated with each digital entity, integrity attributes associated with each digital entity, a globally unique object identifier or handle, etc. Archival processes create the state information that is mapped onto the logical name space. Thus systems for the execution of archival processes can be built out of data grid and virtual data grid technologies. We will show how the particular capabilities of data grids can be used to support archival processes.

The digital entities that are managed by persistent archives can be quite complex. A digital entity can be a single document, or a compound document with multiple components. A digital entity can also be the digital collection that is assembled by grouping multiple documents along with their descriptive, provenance, and integrity metadata. All of these forms of digital entities can be manipulated by archival processes.

The totality of the archival aggregation and the preservation information is treated as a unit and called the *derived archival data product*. The preservation information is registered in a digital collection as part of the description process, while the archival aggregation is written to a storage repository as part of the preservation process.

A persistent archive manages digital collections of digital entities. The digital collection itself can be thought of as a derived data product that results from the application of archival processes to a group of constituent documents. The archival processes partly capture and generate the descriptive and integrity metadata and register metadata for provenance. The digital collection is used to provide a context for the digital entities that are stored in the archive. Discovery of an individual digital entity within a digital collection is accomplished by querying the descriptive metadata. A description of the processing steps used to create the archival aggregation can also be archived. Thus the archival aggregation is itself a derived data product. A request for access to the archival aggregation can result in a query against an instantiated version of the digital collection metadata, residing in a database. If the archival aggregation resides in an archival form within a storage repository, the request can cause the execution of the processing steps that are needed to import the metadata for the digital collection into a database to support subsequent queries. A persistent archive can be treated as a virtual data grid that manages access to derived digital collections, and manages the re-creation of the digital collections if they are not already instantiated. Persistent archives also manage the migration of data from old storage systems to new storage systems and manage transformative migrations, in which the encoding standard used to describe a data entity is changed to a new standard. Management of the application of transformative migrations again is equivalent to management of derived data products in a virtual data grid.

3.1 Archival Processes

Archivists rely on persistent archives to support archival processes, including appraisal, accessioning, arrangement, description, preservation, and access. A virtual data grid provides support for digital entities and for digital collections, and for execution of processes. To demonstrate the advantage of using virtual data grids, we examine how the virtual data grid capabilities can be used to implement each of the standard archival processes. For each archival process, we list the corresponding processing steps, and the technologies that enable support for the processing steps. The characterization is done in terms of traditional archival processes applied for paper records. In section 4.2 we look at alternate choices for archival processes.

3.1.1 Appraisal

The process of determining the disposition of records and in particular which records need long-term preservation.

The data grid allows an archivist to get a quick overview of the other records of the institution that have already been accessioned into the archives. The metadata associated with those other digital collections would assist the archivist in assessing the relationship of the records being appraised to those other records. This metadata would also provide information that would help the archivist understand the relevance/importance/value of the records being appraised for documenting the activities, functions, etc. of the institution that created them

3.1.2 Accessioning

The formal acceptance into custody and recording of an acquisition: Accessioning requires the controlled import of data.

Controlled data import – Data grids provide a logical name space that supports the registration of digital entities as they are received. The logical name space is decoupled from the underlying storage systems, making it possible to reference digital entities without moving them. It is possible to represent a digital entity by a handle (such as a URL), register the pointers into the logical name space, and organize the pointers independently of the physical digital entities. Data grids put digital entities under management control, such that automated processing can be done across an entire digital collection. Data grids provide mechanisms that can be used to validate data models, extract metadata, and authenticate the identification of the submitter. Validation is also used to verify that the digital entity or archival aggregation ingested into the archive is unchanged from the entity or archival aggregation provided by the submitter. Data grid capabilities that are used to manage the ingestion of digital entities into the archive under process control are typically implemented as remote processes that can be executed at the storage repository that holds the digital entity.

3.1.3 Arrangement and Description

Arrangement is the process of identifying records as they belong to groups within an archives. Description is the written representation of archival material, the process of analyzing, organizing, and recording information that serves to identify, manage, locate, and explain archival materials and its context. Arrangement requires the management and definition of the context for the documents. In particular, it must be possible to preserve the structure (respect to original order) of the material, while making it possible to add a new structure when adding a new description. Description requires a logical name space and characterization of encoding formats. Additional requirements are maintenance of the meaning of the information in the digital collection, and the ability to add contextual information and keep it meaningful.

Structure management – Data grids decouple the definition of the structure that is imposed on digital entities by a classification system, from the physical location of the digital entities and the logical organization of the digital entities. It is possible to retain the association of a digital entity with the aggregation in which it belongs even when the digital entities within the aggregation are distributed across multiple sites. Data grids use digital collections to manage the context associated with data entities. The context includes provenance information describing the processes under which the data entities were created and their classification, attributes used to support information discovery to identify an individual data entity, technical administrative attributes for the location of the digital entity, and relationships that can be used to determine whether associated attribute values are consistent with implied knowledge about the digital collection, or represent anomalies and artifacts. An example of a knowledge relationship is the range of permissible values for a given attribute, or a list of permissible values for an attribute, or the presence of a worm/virus within the digital entity. If the range of values do not match the assertions provided by the submitter or a worm/virus is present, the archivist needs to note the discrepancy as a property of the digital collection. The context management also is used to control the level of granularity associated with the organization of the data entities into digital collections/sub-collections. Containers, the digital equivalent of a cardboard box, are used to physically aggregate data entities. Containers are important for minimizing the number of files that are used to store the digital entities, and are used to minimize the number of separate media used to hold large numbers of digital entities. Data grids also provide support for logical organization of digital entities into digital collection hierarchies.

Identification information for preservation, Global name space – The ability to identify derived data products is based on persistent logical identifiers that are independent of the local storage system file names. For persistent archives this includes the ability to provide persistent logical identifiers for the data entities stored within the digital collections. The logical name space may be organized into a digital collection/sub-collection hierarchy with each sub-collection supporting unique metadata. Each sub-collection is described by an extensible set of attributes that can be defined independently of other sub-collections.

Derived product characterization – Both the derived data products (transformative migrations of digital entities to new encoding formats) and the processes used to generate the derived data products can be characterized in virtual data grids. For persistent archives, the derived data product can be a digital collection or the transformative migration of a digital entity. Infrastructure independent representations are used to describe both the derived data product and the processes used to re-create the derived data product. Infrastructure independent representations are typically created by transforming from a proprietary encoding format to a published encoding standard such as the Rich Text Format for documents, eXtensible Markup Language (XML) for metadata, the Hierarchical Data Format (HDF) for binary array data, and an XML Schema and Data Definition Language table structure for digital collections. Published encoding standards exist for images (tiff), and audio and moving pictures (MPEG).

3.1.4 Preservation

The processes and operations involved in insuring the technical and intellectual survival of authentic records through time: Preservation requires the ability to maintain accessibility, instantiate a digital collection, a mechanism to interact with multiple types of storage repositories, support for disaster recovery, mechanisms to maintain the ability to display the records, and mechanisms for asserting authenticity.

Instantiation – A virtual data grid provides the ability to execute a process description. An example is the Chimera system that defines an abstract representation for the steps in a set of processes, and then instantiates the processes as applications running on grid resources. For a persistent archive, this is the ability to instantiate a digital collection from its infrastructure independent representation.

Storage repository abstraction – The ability to migrate digital entities between different types of storage systems is provided by data grids through a storage repository abstraction that defines the set of operations that can be performed on a storage system. The heterogeneous storage repositories can also represent different versions of storage systems and databases as they evolve over time. When a new infrastructure component is added to a persistent archive, both the old version and new version will be accessed simultaneously while the data and information content are migrated onto the new technology. Through use of replication, the migration can be done transparently to the users. For persistent archives, this includes the ability to migrate a digital collection from old database technology onto new database technology.

Disaster recovery – Data grids manage replicas of digital entities, replicas of digital collection attributes, and replicas of digital collections. The replicas can be located at geographically remote sites, ensuring safety from local disasters.

Persistency – Virtual data grids provide a consistent environment, which guarantees that the technical administrative attributes used to identify derived data products always remain consistent with migrations performed on the data entities. The consistent state is extended into a persistent state through management of the information encoding standards used to create platform independent representations. The ability to migrate from an old representation of an information encoding standard to a new representation leads to persistent management of derived data products. It is worth noting that a transformative migration can be characterized as the set of operations performed on the encoding syntax. The operations can be applied on the original digital entity at any point in the future. If a new encoding syntax standard emerges, the set of operations needed to map from the original encoding syntax to the new encoding syntax can be defined, without requiring any of the intermediate encoding representations. The operations needed to perform a transformative migration are characterized as a digital ontology. This idea is discussed further in section 6.

Integrity – Data grids provide the ability to track operations done on each digital entity. This capability can be used to track the provenance of digital entities, including the operations performed by archivists. Audit trails record the dates of all transactions and the names of the persons who performed the operations. Digital signatures and checksums are used to verify that between transformation events the digital entity has remained unchanged. The mechanisms used to accession records can be re-applied to validate the integrity of the digital entities between transformative migrations. Data grids also support versioning of digital entities, making it possible to store explicitly the multiple versions of a record that may be received. The version attribute can be mapped onto the logical name space as both a time-based snapshot of a changing record, and as an explicitly named version.

3.1.5 Access

The right or opportunity of finding, retrieving, and consulting records: Access requires the ability to identify relevant documents, interaction with storage systems for document retrieval, and the creation of a copy for the user.

Derived data product access – Virtual data grids provide direct access to the derived data product when it exists. This implies the ability to store information about the derived data products within a digital collection that can be queried. A similar capability, implemented as a finding aid, is used to characterize the multiple digital collections and contained data entities that are stored in a persistent archive. The finding aid can be used to decide which digital collection to instantiate if the digital collection is not already on-line.

Data transport – Data grids provide transport mechanisms for accessing data in a distributed environment that spans multiple administrative domains. This includes support for moving data and metadata in bulk, while authenticating the user across administrative domains. Data grids also provide multiple roles for characterizing the allowed operations on the stored data, independently of the underlying storage systems. Users can be assigned the capabilities of a curator, with the ability to create new sub-collections, or annotator with the ability to add comments about the digital entities, or submitter, with the ability to write data into a specified sub-collection, or public user, with the ability to read selected sub-collections. Annotations are an example of the execution of transactions on the original digital entities. The annotations are mapped onto the logical name space and are managed independently of the original digital entities.

Storage repository abstraction - Data grids provide the mechanisms needed to support distributed data access across heterogeneous data resources. Data grids implement servers that map from the protocols expected by each proprietary storage repository to the storage repository abstraction. This makes it possible to access digital entities through a standard interface, no matter where it is stored.

3.2 Re-arrangement of Digital Collections within data grids (re-creation done virtually or logically)

The re-arrangement of digital collections is the process of re-applying the archival processes to generate a new archival form of the digital entities, such as in the creation of a research collection.   From the perspective of researchers, the utility and usefulness of archived data is directly proportional to the ability to extract information and knowledge for application in new situations. This is in contrast to a study of access to Federal electronic records that indicates that some of the most frequent users of archival collections are government agencies seeking to use the records for their original purposes.   Data grid technology is able to support both user communities. The processes that are applied to create the archival collections can also be applied in support of research, to mine information and knowledge content.

Re-purposing corresponds to mapping the original context used to describe the digital entities to a new context. The mapping is intended to make the archived material relevant for new uses, beyond the original context under which the archival collection was formed. In one sense, the archival tasks of description and arrangement re-purpose an original collection, which was created for business purposes, to the archival purposes of preservation and access. The process of re-purposing of an archival collection by researchers corresponds to the execution of the data grid processes that support description, arrangement, and access to create a new context for the archived material. The new context can be expressed as additional descriptive metadata that is associated with the original digital entities.

4. Persistent Archive Functionality Requirements

The archival processes that have been described need to be mapped onto the functionalities that are provided by data grids. There are multiple challenges in doing the mapping, including the need to use grid concepts for describing the basic capabilities in data grids and the fact that data grid capabilities are used by more than one archival process. Each data grid capability has been implemented to address a particular data grid functionality requirement. There is no one-to-one mapping between the capabilities provided by data grids and the requirements needed by archival processes. We will describe the capabilities that data grids provide using data grid concepts, and then show how these capabilities can be mapped onto the archival processes.

The requirements for a persistent archive can be expressed in general as "transparencies" that hide virtual data grid implementation details. Examples include digital entity name transparency, data location transparency, platform implementation transparency, encoding standard transparency, and authentication transparency for single sign-on access environments. The capabilities of a persistent archive can be characterized as the set of "transparencies" needed to manage technology evolution. Implementations exist in data grids for at least four key functionalities or transparencies that simplify the complexity of accessing distributed heterogeneous systems:

- Name transparency – The ability to identify a desired digital entity without knowing its name can be accomplished by queries on descriptive attributes, organized as a digital collection. Persistent archives are inherently digital collections that support descriptive and location metadata and map from unique descriptive attribute values to a global, persistent, identifier, and then to the physical location of the digital entity.
- Location transparency – The ability to retrieve a digital entity without knowing where it is stored can be accomplished through use of a logical name space that maps from the global, persistent, identifier to a physical storage location and physical file name. If the data grid owns the digital entities (the digital entities are stored under a custodian user ID defined for the data grid), the administrative attributes for storage location and file name can be consistently updated every time the digital entity is moved.
- Platform implementation transparency – The ability to retrieve a digital entity from arbitrary types of storage systems can be accomplished through use of a data grid that provides a storage repository abstraction. The data grid maps from the protocols needed to talk to the storage systems to the operations defined by the storage repository abstraction. Every time a new type of storage system is added to the persistent archive, a new driver is added to the data grid to map from the new storage access protocol to the data grid data access protocol. Similar platform transparency is needed for the information repository in which the persistent archive stores the digital collection context. An information repository abstraction is defined for the set of operations needed to manipulate a catalog in an information repository, or database.
- Encoding standard transparency – The ability to display a digital entity requires understanding the associated data model and encoding standard for information. If infrastructure independent standards are used for the data model and encoding standard (non-proprietary, published formats), a persistent archive can use transformative migrations to maintain the ability to display the digital entities. The transformative migrations will need to be defined between the original encoding standard and the contemporary infrastructure independent data model standard.

The infrastructure that supports the above transparencies exists in multiple data grid implementations. When one examines the data grid implementations, it is possible to identify over 150 different capabilities that have been implemented to facilitate the management of data and information in distributed environments. The challenge is defining the minimal set of capabilities that should be provided by a data grid for implementing a viable persistent archive.

The fundamental capabilities can be categorized as:

- Logical name space
- Storage repository abstraction
- Information repository abstraction
- Distributed resilient scalable architecture
- Virtual data grid

A set of core capabilities has been defined in Table 1. The list includes the essential capabilities that simplify the management of digital collections of digital entities while the underlying technology evolves. The use of each capability by one of the six archival processes is indicated. We also list re-purposing. The columns are labeled by App. (Appraisal), Acc. (Accessioning), Arr. (Arrangement), Des. (Description), Pres. (Preservation), Ac. (Access), and Rep. (Re-purposing).

The decision to mark a capability as required by an archival process was inclusive. All proposed uses of a capability by an archival process were included. To illustrate the capability identification process, the rationales for inclusion of selected capabilities are given for boxes marked with numbers in Table 1. In Appendix A, the rationales are listed as examples of the reasoning behind the assessment.

Table 1 indicates one of the problems in defining a core set of capabilities, in that many of them are used by most of the archival processes. Thus the logical name space is used when referencing archived digital entities by all of the archival processes. This implies there may be a better decomposition of archival tasks that is more strongly aligned with the application of a logical name space versus the mechanisms used to manipulate digital entities. For this section, we choose to retain the traditional characterization of archival processes.

Core Capabilities	**App.**	**Acc.**	**Arr.**	**Des.**	**Pres.**	**Ac.**	**Rep.**
Storage repository abstraction		x	x		x	x	x
Storage interface to at least one repository		x	x	1	x	2	x
Standard data access mechanism		x	x	3	x	x	x
Standard data movement protocol support		x	x	4	x	x	x
Containers for data		x	x		x	5	x
Logical name space	x	x	x	x	x	x	x
Registration of files in logical name space	x	x	x	x	x		x
Retrieval by logical name		x	6		x	x	x
Logical name space structural independence from physical name space	x	x	x	x	x	7	x
Persistent handle		x	x	x	x	x	x
Information repository abstraction	x	x	x	x	x	x	x
Custodian owned data	x	x	x	x	x	x	x
Collection hierarchy for organizing logical name space	8	x	x	x			x
Standard metadata attributes (controlled vocabulary)	9	10	x	x	x	x	x
Attribute creation and deletion	11	x	x	x	x		x
Scalable metadata insertion		x	x	x	x		x
Access control lists for logical name space to control who can see, add, and change metadata	12	13	x	x	x	x	x
Attributes for mapping from logical file name to physical file names		14	x		x	x	x
Encoding format specification attributes	15	x		x	16	x	x
Data referenced by catalog query						x	x
Containers for metadata		x	x	x	x	17	x
Distributed resilient scalable architecture	x	x	x	x	x	x	x
Specification of system availability		x			x	x	x
Standard error messages		x	x	x	x	x	x
Status checking		x	x	x	x	x	x
Authentication mechanism	x	x	x	x	x	x	x
Specification of reliability against permanent data loss	18	19	20	21	x		
Specification of mechanism to validate integrity of data and metadata		22	x	23	x	x	x
Specification of mechanism to assure integrity of data and metadata	24	x	x	25	x	26	x
Virtual Data Grid		x	x	x	x	x	x
Knowledge repositories for managing collection properties	27	28	x	x	29	x	x
Application of transformative migration for encoding format		x	x	x	x	x	x
Application of archival processes		x	x	x	x	x	x

Table 1. Core data grid capabilities for implementing a persistent archive

Possibly the unique capability that must be present in a persistent archive is the ability to preserve integrity. This implies an environment in which only authorized actions can take place. Every operation within the persistent archive should be tracked, and the corresponding technical administrative metadata updated to guarantee consistency between the preservation context and the derived archival data products. This can be most easily implemented by having the digital entities stored under the control of the data grid. This forces access to be done through the data grid, making it possible to track all operations that are done on the digital entities, from transformative migrations, to media migrations, to replication, to accesses. Data grids implement restricted access through the use of collection-based ownership of the registered digital entities.

We note that the choice of core capabilities is an opportunistic definition of the mechanisms that are now available through data grids. We recognize that many of the core capabilities can be implemented as procedural policies on current file system based storage repositories, without using data grid technology. For example, integrity can be managed by defining a set of user IDs that are allowed to write to the archive. One can then require that the defined set of persons manually enter characterizations of all operations that they perform. In practice, this approach would be labor intensive. The list of core capabilities is intended to minimize the labor associated with organizing, managing, and evolving a persistent archive.

A question is whether the levels of abstraction associated with virtual data grids are consistent with operations in persistent archives. One can think of a data grid as the set of abstractions that manage differences across storage repositories, information repositories, knowledge repositories, and execution systems. Data grids also provide abstraction mechanisms for interacting with the objects that are manipulated within the grid, including digital entities (logical namespace), processes (service characterizations or application specifications), and interaction environments (portals). The data grid approach can be defined as a set of services, and the associated APIs and protocols used to implement the services. The data grid is augmented with portals that are used to assemble integrated work environments to support specific applications or disciplines. An example is an archivist workbench, which provides separate functions for each of the archival processes. A major question is whether a persistent archive is better implemented as a virtual data grid, incorporating the required functionality directly into the grid, or as a portal, with the required integrity and management control implemented as an application interface.

4.1 Persistent Archive versus Persistent Storage

There is a distinction between Persistent Archives and Persistent Storage. Persistent storage systems provide archival media that have a very long shelf life, such as heavy-ion beam encoded disk, film, etc. A standard encoding is chosen, such as ASCII that is assumed readable at an arbitrary date in the future. The technology to extract meaning from the archived material is based on the ability to parse ASCII. Persistent archives recognize that there is a cost benefit that can be obtained by migrating to new technology, including minimization of floor space through higher density media, lower cost storage media, elimination of obsolete equipment, and improved access.

Both systems need universal identifiers, the ability to manage descriptive, integrity, and preservation metadata for the archived material, policy management systems to control the archival workflow, and audit trails of transactional activity and user history. Grid technology provides the mechanisms that make it possible to migrate to new versions of technology, and to new archival services. The distributed state information that is managed by grid technology can be employed to support more extended applications for in-depth re-purposing beyond general catalog functions.

5. Data Grid Implementations

For a persistent archive implementation to be based upon existing data grids, we must demonstrate that the corresponding capabilities are actually present within current data grid environments. To better understand the current status of data grids, we present an analysis of the capabilities that are already provided by production systems. The Global Grid Forum is promoting the development of standards for the implementation of data grids. A survey has been conducted to identify the capabilities that are supported by most data grid implementations. We note that data grids are used to support distributed digital collections, digital libraries, and persistent archive projects.

In December 2001, a comparison was made between the Storage Resource Broker (SRB) data grid from the San Diego Supercomputer Center, the European DataGrid replication environment (based upon GDMP, a project in common between the European DataGrid and the Particle Physics Data Grid, and augmented with an additional product of the European DataGrid for storing and retrieving meta-data in relational databases called Spitfire and other components), the Scientific Data Management (SDM) data grid from Pacific Northwest Laboratory, the Globus toolkit, the Sequential Access using Metadata (SAM) data grid from Fermi National Accelerator Laboratory, the Magda data management system from Brookhaven National Laboratory, and the JASMine data grid from Jefferson National Laboratory. These systems have evolved as the result of input by user communities for the management of data across heterogeneous, distributed storage resources.

EGP, SAM, Magda, and JASMine data grids support high energy physics data. The SDM system provides a digital library interface to archived data for PNL and manages data from multiple scientific disciplines. The Globus toolkit provides services that can be composed to create a data grid. The SRB data handling system is used in projects for multiple US federal agencies, including the NASA Information Power Grid (digital library front end to archival storage), the DOE Particle Physics Data Grid (collection-based data management), the National Library of Medicine Visible Embryo project (distributed digital collection), the National Archives Records Administration (persistent archive prototype), the NSF National Partnership for Advanced Computational Infrastructure (distributed digital collections for astronomy, earth systems science, and neuroscience), the Joint Center for Structural Genomics (data grid), and the National Institute of Health Biomedical Informatics Research Network (data grid).

The systems we examine therefore include not only data grids, but also distributed digital collections, digital libraries and persistent archives. Since the core component of each system is a data grid, we can expect common capabilities to exist across the multiple implementations. The systems that provided the largest number of features tend to have the most diverse set of user requirements.

The comparison is an attempt at understanding what data grid architectures provide to meet existing application requirements. The capabilities are organized into functional categories, such that a given capability is listed only once. The categories have been chosen based on the need to manage a logical name space, the management of attributes in the logical name space, the storage abstraction for accessing remote storage systems, the types of data manipulation, and the data grid architecture. Since the listed data grids have been in use for multiple years, the features that have been developed represent a comprehensive cross-section of the features in actual use by production systems. The terms used in the comparison are explained in the Glossary in section 9. The results of the comparison are shown in Appendix B.

5.1 Common Data Grid Capabilities:

What is most striking is that common data grid capabilities are emerging across all of the data grids. Appendix B lists the common features organized by functional category. Each data grid implements a logical name space that supports the construction of a uniform naming convention across multiple storage systems. The logical name space is managed independently of the physical file names used at a particular site, and a mapping is maintained between the logical file name and the physical file name. Each data grid has added attributes to the name space to support location transparency, file manipulation, and file organization. Most of the grids provide support for hierarchical logical folders within the namespace, and support for ownership of the files by a community or collection ID.

The logical name space technical administrative attributes typically include the replica storage location, the local file name, and user-defined attributes. Mechanisms are provided to automate the generation of attributes such as file size and creation time. The attributes are created synchronously when the file is registered into the logical name space, but many of the grids also support asynchronous registration of attributes.

Most of the grids support synchronous replica creation, and provide data access through parallel I/O. The grids check transmission status and support data transport restart at the application level. Writes to the system are done synchronously, with standard error messages returned to the user. However, the error messages are different across each of the data grids. The grids have statically tuned the network parameters (window size and buffer size) for transmission over wide area networks. Most of the grids provide interfaces to the GridFTP transport protocol.

The most common access APIs to the data grids are a C++ I/O library, a command line interface and a Java interface. The grids are implemented as distributed client server architectures. Most of the grids support federation of the servers, enabling third party transfer. All of the grids provide access to storage systems located at remote sites including at least one archival storage system that can write data onto removable media such as tape. The grids also currently use a single catalog server to manage the logical name space attributes. All of the data grids provide some form of latency management, including caching of files on disk, streaming of data, and replication of files.

5.2 Prior Conceptual Models

Conceptual models can be used from prior research efforts to evaluate the completeness of the proposed approach to persistent archives based on data grid technology. The models are selected from the archives and computer science domains, and include: *traditional archive procedures*, a *reference model for BAC (business acceptable communications)* developed by the University of Pittsburgh, the *records continuum model* proposed by the Monash University in Australia, the *reference model for an open archival information system (OAIS)* designed by the CCSDS of NASA, the Preservation Model of the InterPARES Project, the ISO/IEC 11179 (metadata schema standard) for data element composition for inclusion in metadata registries, and the ISO records management standard (ISO 15489). A comparison with these models illustrates the multiple characterizations of archival processes that have been used. The comparison also demonstrates the importance of policy management issues.

The OAIS reference is now an ISO standard.

5.2.1 Traditional Archival Procedures

The capabilities of virtual data grids can be used to implement the traditional archival processes, as shown in the list in section 3.1.

5.2.2 Records Continuum Model

The records continuum model uses four processes for preservation, namely, create, capture, organize and pluralize. Frank Upward states: “The four continua I chose to represent as sets within a spacetime continuum model were identity, transactionality, evidentiality, and recordkeeping containers [which I more normally refer to these days as recordkeeping objects]” [Upward, 2000, p. 123]. The records continuum model was originally developed as a teaching tool to communicate evidence-based approaches to archives and records management. Upward [2000, p 128] states that: "It [the records continuum model] can never provide complete or satisfying views of detailed practice, but that is not what a worldview does. It provides an overview for re-organising our detailed knowledge and applying our skills in contexts framed by the task at hand." "As a view it presents a multi-layered and multi-faceted approach which can be used to re-organise knowledge and deploy skills. It is more in tune with electronic communications and technological change than a life cycle view [Upward, 2000, p. 128]."

Data management systems that provide mechanisms to manage technological change are consistent with the records continuum model.

5.2.3 BAC Reference Model

The Reference Model for BAC is also based on a distributed environment. Thus the data grid approach is consistent with the BAC model, except for three layers: terms and conditions layer, contextual layer, and user history layer. These layers comprise policy management (for terms and conditions), a knowledge layer (for defining the context), and an access layer (for describing user interaction history). These layers are expected to become grid mechanisms that in the future will be part of virtual data grids. The implication is that the proposed data grid model must continue to evolve to include future grid services that are appropriate for preservation.

5.2.4 OAIS Model

The OAIS system specifies a reference model for describing the processes associated with preservation from the viewpoint of submission information packages (SIPs), archival information packages (AIPs), and dissemination information packages (DIPs). The OAIS reference model can be implemented on top of data grid technology through the specification of the interaction and information packaging mechanisms. The OAIS reference model specifies the information that should be associated with each procedure. The information can be stored with the digital entities, or stored in a metadata repository that can be queried, or stored in both places. This implementation choice is left to the persistent archive.

From the OAIS point of view, "The OAIS model provides a theoretical framework for an archival system, and integrates its conceptual approach with a hierarchical structure for organizing information." The model does not specify an implementation strategy; instead it provides guidelines to address digital archiving concepts both from functional and information model. The OAIS model describes an Archival Information Package (AIP) as an aggregation of four types of Information Objects:

1. Content information object: includes the data object as well as representation information (structural and semantic information about the object).
2. Preservation description information object: comprised of reference information, provenance information, context information and fixity information.
3. Packaging information object: information that is stated as being used to bind and identify the components of an Information Package. An example is the ISO-9660 volume and directory information used on a CD-ROM to define the content of several files containing Content Information and Preservation Description Information.
4. Descriptive information object.

Besides the OAIS metadata, technical metadata (refers to the administrative, structural, and preservation metadata related to digital objects) is needed to facilitate management and access to archival objects. The technical metadata needs to be independent of the object itself to support interoperability and preservation. The OAIS reference model provides a very good reference not just for the design of a long-term digital archive, but also provides several real use cases in its appendix for verification.

Important questions are:

1. Whether all functional entities in OAIS are covered by Persistent Archives? Examples include the data management, administration, and preservation planning entities. As noted in the BAC Reference Model, the policy management and planning mechanisms have not yet been

implemented as grid services, and will need to be investigated for inclusion in the persistent archive when they become available.

2. Whether any difference or loss of functional requirements exists between the Persistent Archive Components and the Migration types (i.e. refreshment, replication, repacking and transformation) offered by OAIS? The assertion is that all of the OAIS Migration types are supported within the Data Grid implementation.

5.2.5 ISO/IEC 11179

ISO/IEC 11179 specifies basic aspects of data element composition for inclusion in metadata registries. The standard applies to the formulation of data element representations and meaning as shared among people and machines. Metadata registries attempt to be authoritative sources for metadata schemas for different communities that author semantic information (semantic maps with associated procedures for storing and registering detailed metadata from multiple sources and diverse organizations in a common structured form). (Extensions to the formats are recorded, as are agreed-upon mappings between diverse formats – Meta Object Facility of OMG). Use of the ISO /IEC standard and participation in metadata registries promotes access, understanding, and sharing of data across time and space, and use of this structure makes it easier to check the metadata for consistent application.

5.2.6 Other projects

There are many possible levels of granularity and different ways to categorize information. Working within the OAIS framework and ISO/IEC 11179 is a sound strategy because it makes possible improved communication among divergent digital applications. Other projects are also addressing the characterization of preservation systems.

- The National Library of the Netherlands Long Term Preservation Study distinguishes between Intellectual Preservation, Media Preservation and Technology Preservation.
- The Making of America project distinguishes between descriptive, administrative and structural metadata.
- The METS schema classifies administrative metadata into four types: technical metadata, intellectual property rights metadata, source metadata and digital provenance metadata.
- In "How to Preserve Authentic Electronic Records" InterPARES distinguishes between conservation actions and maintenance activities as part of preservation.
- The IEEE Learning Object Metadata draft standard includes metadata categories for general; lifecycle; meta-metadata; technical; educational; rights; relation; annotation; classification.
- The NDAP National Digital Archive Project proposes core capabilities for preservation, including linkage to the original object to keep complete information about how the digital entity was created (ways of digitization, equipment used, workflow, accuracy, data quality, and specifications for the digitization work); metrics to specify the quality and completeness (in terms of information loss ratio) of the digitized entity; support for knowledge level information discovery; analysis of metadata to provide structure and organization for the contents (metadata schema design); descriptions of each digital collection in the content space which is composed of space, time, and linguistics perspectives; flexible presentation with the content (representation) separated from presentation through a content management framework (CMF) constructed to manage the workflow from authoring to publishing; and authentication for data and owner.
- ISO TC 46 SC 11, committee responsible for records and archival management standards (issued 15489 standard and 23081 standard)

6. Persistent Archive Components

Given a consensus on the set of capabilities provided by a data grid, it is possible to identify those capabilities that are relevant to the creation of a persistent archive. In Appendix D, a description of each of the core capabilities is provided. We define the core capability, describe the functionality that would be provided by the capability, and provide a description of how the capability is implemented in current data grid environments. Note that the choice of implementation is arbitrary, with possibly multiple mechanisms used to implement a particular capability.

A major design issue for the creation of persistent archives is the development of an approach in which the digital entities can be preserved in an unchanged format, while still making it possible for future presentation applications to display the digital entity. The challenge is that the encoding format interpreted by future applications will not be the same as the encoding format used to create the original digital entity. Three approaches are being considered within the archival community to resolve this challenge.

1. Migration of digital entities applies an archival process to create an infrastructure independent representation of the digital entity by changing the encoding format to a non-proprietary standard. In the process, the bits of the digital entity must be changed to the standard encoding format. The expectation is that the transformative migration will need to be done at an infrequent interval.

2. Emulation preserves the original digital entity by migrating the presentation application onto new technology. Instead of migrating the digital entity to new encoding formats, the presentation application is migrated to new operating systems. This requires migrating onto new technology the applications that were used to create or view each digital entity. The result is a system that preserves the look and feel of the original software, but at the same time makes it very difficult to apply any new techniques to the interpretation of the digital entities. An emulator can be characterized as the set of operations that the original application must be able to perform through an operating system. This characterization is typically specified as a set of operating system calls. An emulator maps from the system calls used by the original application to the system calls provided by current operating systems. The Dyninst system is an example of software that supports the dynamic insertion of new system calls into existing code, and can be viewed as enabling infrastructure for the development of emulators.

3. Migration of digital ontologies, characterizations of the data structure and data model that specify how to manipulate a digital entity. Emulation and migration capabilities can be combined by creating a digital ontology that organizes the relationships present within a digital entity. A digital entity can be viewed as a sequence of bits onto which structural, procedural, and semantic relationships are applied. These relationships include the structural relationships that define how to turn the bits into binary arrays, or words, or tables. Logical relationships are used to apply semantic tags to the structures. Spatial relationships are used to map binary arrays to coordinate systems. Temporal relationships are used to apply time stamps to structures. The digital ontology specifies the order in which the relationships need to be applied to correctly interpret the information and knowledge content.

   The digital entity is kept in its original encoding format. Instead of changing the encoding format of the digital entity to a non-proprietary standard, a digital ontology is created that defines the relationships present within the digital entity. The digital ontology is migrated onto new encoding standards for relationships over time. For instance, a digital ontology can be represented using the Resource Description Framework syntax. In the future, when a new syntax is used to specify relationships, the digital ontology can be migrated from the old syntax to the new syntax, without modifying the original digital entity.

   The presentation application is emulated as the set of operations that can be performed on the defined relationships. The set of operations can be kept fixed on the original set, or they can be expanded over time as new capabilities are created (such as causal queries on time stamps). In effect, the presentation application is emulated as operations on a digital ontology, and the digital ontology is migrated forward in time onto new encoding formats.

All references to migration in this report can be interpreted as either migration of digital entities onto new encoding formats for display by future applications, or migration of digital ontologies onto new encoding formats for display through a standard set of operations.

6.1 Example Persistent Archive

An example persistent archive has been constructed using the San Diego Supercomputer Center Storage Resource Broker data grid. The persistent archive components based upon the SRB include:

- Logical name space implemented in the Metadata Catalog (MCAT). The logical names are chosen by the archivist. The archivist will use the original record names whenever possible. A mapping is maintained from the logical name to the physical file location. The logical names are infrastructure independent, and are organized in a digital collection hierarchy, allowing the specification of different descriptive metadata for each sub-collection. Soft links and shadow links are supported for the logical organization and registration of digital entities. Digital entities may include files, URLs, SQL command strings, directories, and database tables. Distributed state information is mapped onto the logical name space as attributes.
- Storage repository abstraction implemented in the SRB. The set of operations that are supported include Unix file system operations (create, open, close, unlink, read, write, seek, sync, stat, fstat, mkdir, rmdir, chmod, opendir, closedir, and readdir), latency management operations (aggregation of data, I/O commands, and metadata), and metadata manipulation (extraction, registration) through use of remote procedures. Containers are used to physically aggregate digital entities before storage into archives. Both digital entities and containers can be replicated. The storage repository abstraction is used to manage data within Unix file systems, archives, object-relational databases, object ring buffers, storage resource managers, FTP sites, GridFTP sites, and Windows file systems.
- Information repository abstraction implemented in the MCAT. Mechanisms are supported for schema extension through addition of new attributes, table restructuring, and metadata import and export through XML files. Soft links are supported for logical reorganization of digital entities within a digital collection hierarchy. Metadata attributes are maintained for provenance attributes (Dublin core), technical administrative metadata (file location), descriptive metadata (user-defined attributes), and integrity metadata (audit trails, digital signatures). The information repository abstraction is used to manage metadata in both proprietary and non-proprietary databases including DB2, Oracle, Sybase, Informix, Postgresql, and mySQL.
- Distributed resilient architecture implemented through a federated client server architecture. Servers are installed in front of each storage repository and in front of the information repository. Access to the system results in the creation of a service instance that manages further interactions for the request. The service instance retrieves all required distributed state information from the MCAT catalog that is needed to complete the request, and interacts with remote servers as needed to access the data. The system has been designed to minimize the number of message sent over wide area networks to improve performance and increase reliability. Data retrieval requests are automatically retried on a replica when a storage repository does not respond. All error messages generated by the network, storage repository, and information repository are returned to the user. Consistency constraints on distributed state information are explicitly integrated into the software through use of write locks and synchronization flags. This makes it possible to update a file that has been aggregated into a container and replicated into an archive, lock out competing activities to avoid over-writes, and then synchronize all replicas to the new state. When additional records for a record series are received, they can be appended to the container holding the records that have already been accessioned. Changes to digital entities within a container

are made by marking the original digital entity as deleted, and appending the new form of the digital entity to the end of the container. The addition of digital entities to an digital collection can also be done through soft links within the logical name space, making it possible to link digital entities into an existing digital collection, while simultaneously organizing the new digital entities in a separate sub-collection. All technical administrative metadata is automatically generated and updated by the SRB on each request.

- Virtual data grid implemented through use of remote proxies and external process management systems. The SRB provides a mechanism to process data remotely, before it is sent over a network. The Ohio State University DataCutter technology is used to filter data. External process management systems can control the generation of derived data products through application of remote proxies or the DataCutter filters. Interactions with databases can be expressed through SQL command strings that are registered into the logical name space. The SRB is able to apply simple transformative migrations such as unit conversion and reformatting of query results into HTML or XML. More complex transformations require the use of a process management system.

7. Summary

A proposed set of core capabilities can be defined for minimizing the labor required to implement, manage, and evolve a persistent archive. The capabilities are present within implementations of current data grids. Many of the capabilities are general properties that have been implemented across almost all existing data grids. A characterization of each capability has been defined. This characterization can be used as the set of requirements for defining a persistent archive architecture that supports the InterPARES preservation model.

The focus on preservation is intended to be illustrative of the power of data grids. In practice, the technology is applicable to both the records management environments and the preservation environments.

8. Acknowledgements

This paper is based upon a document submitted to the Global Grid Forum through the Persistent Archive Research Group. The authors were Reagan Moore, SDSC, and Andre Merzky, ZIB. The original paper was modified to reflect the viewpoint of the archivist community.

The results presented here were supported by the NSF NPACI ACI-9619020 (NARA supplement), the NSF NSDL/UCAR Subaward S02-36645, the DOE SciDAC/SDM DE-FC02-01ER25486 and DOE Particle Physics Data Grid, the NSF National Virtual Observatory, the NSF Grid Physics Network, and the NASA Information Power Grid. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, the National Archives and Records Administration, or the U.S. government.

The data grid and toolkit characterizations were only possible through the support of the following persons: Igor Terekhov (Fermi National Accelerator Laboratory), Torre Wenaus (Brookhaven National Laboratory), Scott Studham (Pacific Northwest Laboratory), Chip Watson (Jefferson Laboratory), Heinz Stockinger and Peter Kunszt (CERN), Ann Chervenak (Information Sciences Institute, University of Southern California), Arcot Rajasekar (San Diego Supercomputer Center).

The definition of terms used by archivists was provided by Mark Conrad of the National Historical Publications and Records Commission, and the comparison with OAIS technology was provided by Han-wei Yen , Simon C. Lin, Ya-ning Chen, and Shu-jiun Chen of the Computing Centre of Academia Sinica, Taiwan. Margaret Hedstrom provided valuable comments on preservation terminology. More information about the example persistent archive based on the San Diego

Supercomputer Center Storage Resource Broker can be found at http://www.npaci.edu/DICE/SRB/index.html and http://www.sdsc.edu/NARA/.

9. Glossary

Two sets of terminology are used in the report, one from the preservation community, and one from the grid community. In some cases, the same word is used in two different contexts.

9.1 Terms used within the preservation community

ACCESS

(n.) the right, opportunity, or means of finding, using, or approaching documents or information (SAA), http://rpm.lib.az.us/alert/thesaurus/terms.asp?letter=a

Access to archival documents is provided through an archival reference service. Key steps in an archival reference service are:

Querying the researcher to draw out the specific nature of the subject as well as secondary aspects of the subject that can serve as leads to documentation sources. Translating the terms and concepts of the inquiry into the terms and concepts of the archives' reference apparatus.

Explaining finding aids, archival methodology, and the nature of manuscripts and records documentation

Guiding the researcher to the appropriate finding aids and/or records.

Retrieving the records that appear to be relevant to the researcher's inquiry.

Informing the researcher of policies and practices for making copies and handling documents to ensure that the records are not damaged or disarranged.

Consulting with the researcher during and after the visit to determine how well the records answered the question or led to new questions. http://web.library.uiuc.edu/ahx/define.htm

ACCESSION

(v.) To transfer physical and legal custody of documentary materials to an archival institution.

(n.) Materials transferred to an archival institution in a single accessioning action.

http://www.archives.gov/research_room/alic/reference_desk/archives_resources/archival_terminology.html

ARCHIVAL DESCRIPTION

The hierarchy of archival description generally goes from the archival fonds to Series to sub-series to the file to the individual item.

ARCHIVAL FONDS

The whole of the records made or received by a person or organization in the course of activity.

APPRAISAL

n.) the process of determining the value and thus the disposition of records based upon their current administrative, legal, and fiscal use; their evidential and informational value; their arrangement and condition; their intrinsic value; and their relationships to other records (SAA), http://rpm.lib.az.us/alert/thesaurus/terms.asp?letter=a.

ARCHIVES

The organized non-current records of an institution or organization retained for their continuing value in providing a) evidence of the existence, functions, and operations of the institution or organization that generated them, or b) other information on activities or persons affected by the organization. Derived from the Greek word for "government house," the term "archives" also refers to the agency responsible for selecting, preserving, and making available non-current records with long-term value and to the building or part of the building housing them. http://web.library.uiuc.edu/ahx/define.htm

ARRANGEMENT

The body of principles and practices which archivists follow to group records in such a way as to reflect the manner in which they were held and used by the office or person creating the records. It involves the fundamental principles of respect des fonds, provenance, and sanctity of original order. The key units in archival arrangement are: fonds, series, file. http://web.library.uiuc.edu/ahx/define.htm

AUTHENTIC RECORD

A record that is what it purports to be and that is free from tampering or corruption. http://www.interpares.org/book/interpares_book_q_gloss.pdf

AUTHENTICATION

A declaration of a record’s authenticity at a specific point in time by a juridical person entrusted with the authority to make such a declaration. http://www.interpares.org/book/interpares_book_q_gloss.pdf

AUTHENTICATION CERTIFICATE OF TRUSTED THIRD PARTY

An attestation issued by a trusted third party for the purpose of authenticating the ownership and characteristics of a public key. It appears in conjunction with the digital signature of the author of a record, and is itself digitally signed by the trusted third party. http://www.interpares.org/book/interpares_book_q_gloss.pdf

AUTHENTICITY

The quality of being authentic, or entitled to acceptance. As being authoritative or duly authorized, as being what it professes in origin or authorship, as being genuine. http://www.interpares.org/book/interpares_book_q_gloss.pdf

COLLECTION

Collection is often used to refer to an artificial accumulation of documents brought together on the basis of some common characteristic (e.g. means of accumulation, creator, subject, language, medium, form, name of collector) without regard to the provenance of the documents (SAA Glossary).

CONTEXT

The circumstances of creation and history of ownership and usage of an archival collection, as well as the collection’s original arrangement or filing structure. A clear context gives a collection enhanced legal and research value as it indicates that the collection’s integrity was respected during a continuous chain of custody (ownership). The evidence in the collection remains intact. The collection was not rearranged or inappropriately added to or weeded. Historians may depend upon the inferences they draw from the collection’s authentic filing structure. See also original order and provenance http://crm.cr.nps.gov/archive/22-2/22-02-19.pdf

DESCRIPTION

The process of recording information about the nature and content of the records in archival custody. The description identifies such features as provenance, extent, arrangement, format, and contents, and presents them in a standardized form.
http://www.sfu.ca/archives/glossary.html

FONDS

The whole of the records, regardless of form or medium, automatically and organically created and/or accumulated and used by a particular individual, family, or corporate body in the course of that creator's activities or functions.
http://www.sfu.ca/archives/glossary.html

PRESERVATION

Preservation encompasses the activities that prolong the usable life of archival records. Preservation activities are designed to minimize the physical and chemical deterioration of records and to prevent the loss of informational content.
http://www.archives.gov/preservation/about_preservation.html

PROVENANCE

The organization or individual who created, accumulated and/or maintained and used records in the conduct of the business prior to their transfer to an Archives (SAA Glossary)

RECORDS

Documents, made or received by an individual or an organization in the course of a practical activity.

RECORD SUB-GROUPS

Bodies of organizationally related records placed within a fonds corresponding to the subordinate administrative units of the creating organization.

RECORD SERIES

A group of documents within an archival fonds, created to accomplish one function.

RESPECT DES FONDS

The principle of archival arrangement according to which each fonds should be maintained as a separate entity. Also called the Principle of Provenance.

PRINCIPLE OF ORIGINAL ORDER

The principle of archival arrangement according to which the order that the records had when last used by their creator should be maintained.

9.2 Data grid terms for a logical name space

A logical name space is a naming convention for labeling digital entities. The logical name space is used to create global, persistent identifiers that are independent of the storage location. Within the logical name space, information consists of semantic tags that are applied to digital entities.

Metadata consists of the semantic tags and the associated tagged data, and is typically managed as attributes in a database. Metadata is called data about data.

Digital collections organize the metadata attributes that are managed for each digital entity that is registered into the logical name space

Registration corresponds to adding an entry to the logical name space, creating a logical name and storing a pointer to the file name used on the storage system.

The logical name space can be organized as a digital collection hierarchy, making it possible to associate different metadata attributes with different sets of digital entities within the digital collection. This is particularly useful for accession, arrangement, and description.

Logical folders within a digital collection hierarchy represent sub-collections, and are equivalent to directories in a file system, but are used to manage different sets of metadata attributes.

Soft links represent the cross registration of a single physical data object into multiple folders or sub-collections in the logical name space

Shadow links represent pointers to objects owned by individuals. They are used to register individual owned data into the logical name space, without requiring creation of a copy of the object on storage systems managed by the logical name space.

Replicas are copies of a file registered into the logical name space that may be stored on either the same storage system or on different storage systems.

Collection-owned data is the storage of digital entities under a Unix user ID that corresponds to the digital collection. Access to the data is then restricted to a server running under the digital collection ID.

User access is accomplished by authentication to the data grid, checking of access controls for authorization, and then retrieval of the digital entity by the data grid from storage through the digital collection ID for transmission to the user.

9.3 Data grid terms for a storage repository abstraction

A storage repository is a storage system that holds digital entities. Examples are file systems, archives, object-relational databases, object-oriented databases, object ring buffers, FTP sites, etc.

A storage repository abstraction is the set of operations that can be performed on a storage repository for the manipulation of data.

A container is an aggregation of multiple digital entities into a single file, while retaining the ability to access and manipulate each digital entity within the container.

Load balancing within a logical name space consists of distributing digital objects across multiple storage systems

Storage completion at the end of a single write corresponds to synchronous data writes into storage.

Third party transfer is the ability of two remote servers to move data directly between themselves, without having to move the data back to the initiating client

Metadata about the I/O access pattern is used to characterize interactions with a digital entity, recording the types of partial file reads, writes, and seeks.

Synchronous updates correspond to finishing both the data manipulations and associated metadata updates before the request is completed.

Asynchronous updates correspond to completion of a request within the data handling system, after the return was given to a command.

Storage Resource Managers control the load on a Hierarchical Resource Manager or disk file system. They rearrange the submitted work load to optimize retrieval from tape, stage data from the HRM to a disk cache, and manage the number of allowed simultaneous I/O requests.

9.4 Data grid terms for an information repository abstraction

An information repository is a software system that is used to manage combinations of semantic tags (attribute names) and the associated attribute data values. Examples are relational databases, XML databases, Lightweight Directory Access Protocol servers, etc.

An information repository abstraction is the set of operations that can be performed on an information repository for the manipulation of a catalog or digital collection.

Template based metadata extraction applies a set of parsing rules to a document to identify relevant attributes, extracts the attributes, and loads the attribute values into the digital collection.

Bulk metadata load is the ability to import attribute values for multiple objects registered within the logical name space from a single input file.

Curation control corresponds to the administration tasks associated with creating and managing a digital collection

9.5 Data grid terms for a distributed resilient scalable architecture

Federated server architecture refers to the ability of distributed servers to talk among themselves without having to communicate through the initiating client.

GSI authentication is the use of the Grid Security Infrastructure to authenticate users to the logical name space, and to authenticate servers to other servers within the federated server architecture

Dynamic network tuning consists of adjusting the network transport protocol parameters for each data transmission to change the number of messages in flight before acknowledgements are required (window size) and the size of the system buffer that holds the copy of the messages until the acknowledgement is received.

SDLIP is the Simple Digital Library Interoperability Protocol. It is used to transmit information for the digital library community

9.6 Data grid terms for a virtual data grid

The automation of the execution of processes is managed in virtual data grids. References to the result of a process can result in the application of the process, or direct access to the result.

Knowledge corresponds to relationships between attributes, or to relationships that characterize properties of a digital collection as a whole. Relationships can be cast as inference rules that can be applied to digital entities. An example is the set of structural relationships used to parse metadata from a digital entity in metadata extraction.

The application of processes at remote storage systems is accomplished through systems such as the DataCutter, a data filtering service developed by Joel Saltz at the Ohio State University, which is executed directly on a remote storage system.

Transformative migrations correspond to the processing of a digital entity to change its encoding format. The processing steps required to implement the transformative migration can themselves be characterized and archived, and then applied later.

Digital ontologies organize the set of semantic, structural, spatial, temporal, procedural, and functional relationships that are present within a digital entity. The digital ontology specifies the order in which the relationships need to be applied in order to correctly display or manipulate the digital entity.

Derived data products are created by execution of processes under the control of a virtual data grid. For persistent archives, derived data products can be digital collections or transformative migrations of digital entities to new encoding formats. A digital collection can be thought of as a derived data product that results from the application of archival processes to a group of constituent documents.

10. Full Copyright Notice

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

11. References

1. BAC, Business Acceptable Communications, http://www.phila.gov/records/divisions/rm/units/perp/presentations/nagara/nagara96/sld005.html
2. Baru, C., R, Moore, A. Rajasekar, M. Wan, "The SDSC Storage Resource Broker," Proc. CASCON'98 Conference, Nov.30-Dec.3, 1998, Toronto, Canada.
3. Baru, C., R. Moore, A. Rajasekar, W. Schroeder, M. Wan, R. Klobuchar, D. Wade, R. Sharpe, J. Terstriep, (1998a) "A Data Handling Architecture for a Prototype Federal Application," Sixth Goddard Conference on Mass Storage Systems and Technologies, March, 1998.
4. Bellardo, L. J. and L. L. Bellardo, A Glossary for Archivists, Manuscript Curators, and Records Managers. Chicago: The Society of American Archivists, 1992.
5. Beynon, M.D., T. Kurc, U. Catalyurek, C. Chang, A. Sussman, and J. Saltz. ``Distributed Processing of Very Large Datasets with DataCutter''. Parallel Computing, Vol.27, No.11, pp. 1457--1478, 2001.
6. Chan, W.M. and S. E. Rogers. "Chimera Grid Tools Software" Gridpoints - Quarterly Publication of the NAS Division, NASA Ames Research Center, Spring, 2001. http://www.nas.nasa.gov/About/Gridpoints/PDF/gridpoints_spring2001.pdf

7. Dyninst – a machine independent interface to permit the creation of tools and applications that use runtime code patching, http://www.paradyn.org/release3.3/
8. EAD - Encoded Archival Description, http://www.loc.gov/ead/
9. EDG – European Data Grid, http://eu-datagrid.web.cern.ch/eu-datagrid/
10. EML – Ecological Metadata Language, http://knb.ecoinformatics.org/software/eml/.
11. Foster, I., J. Vockler, M. Wilde, Y. Zhao, "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation", Proceedings of the 14th Conference on Scientific and Statistical Database Management, Edinburgh, Scotland, July 2002.
12. Globus – The Globus Toolkit, http://www.globus.org/toolkit/
13. Grid Forum Remote Data Access Working Group. http://www.sdsc.edu/GridForum/RemoteData/.
14. GriPhyN – Grid Physics Network project, http://www.griphyn.org/index.php.
15. HDF – "Hierarchical Data Format", http://hdf.ncsa.uiuc.edu/
16. IEEE Learning Object Metadata, http://ltsc.ieee.org/doc/wg12/LOM_1484_12_1_v1_Final_Draft.pdf
17. InterPares Preservation Task Force, "How to Preserve Authentic Records", Oct. 2001, http://www.interpares.org/book/interpares_book_o_app06.pdf
18. ISO/IEC DIS 9660:1999(E), Information processing - Volume and file structure of CD-ROM for Information Interchange, http://www.y-adagio.com/public/standards/iso_cdromr/tocont.htm
19. ISO/IEC 11179, Specification and Standardization of Data Elements, http://www.diffuse.org/oii/en/meta.html#ISO11179
20. ISO 15489 Records Management Standard
21. Jasmine – Jefferson Laboratory Asynchronous Storage Manager, http://cc.jlab.org/scicomp/JASMine/
22. The Long-Term Preservation of Electronic Records: Findings of the InterPARES Project, Sept. 2001, http://www.interpares.gorg/book/index.html
23. Magda – Manager for distributed Grid-based Data, http://atlassw1.phy.bnl.gov/magda/info
24. Making of America II, http://sunsite.berkeley.edu/MOA2/
25. MARC (MA chine-Readable Cataloging) record, http://cweb.loc.gov/marc/index.html
26. MCAT - "The Metadata Catalog", http://www.npaci.edu/DICE/SRB/mcat.html
27. METS – "Metadata Encoding and Transmission Standard", http://www.loc.gov/standards/mets/
28. Moore, R., A. Rajasekar, "Common Consistency Requirements for Data Grids, Digital Libraries, and Persistent Archives", Grid Protocol Architecture Research Group draft, Global Grid Forum, April 2003
29. Moore, R., C. Baru, "Virtualization Services for Data Grids", Book chapter in "Grid Computing: Making the Global Infrastructure a Reality", John Wiley & Sons Ltd, 2003.
30. Moore, R., "The San Diego Project: Persistent Objects", Proceedings of the Workshop on XML as a Preservation Language, Urbino, Italy, October 2002.
31. Moore, R., C. Baru, A. Rajasekar, B. Ludascher, R. Marciano, M. Wan, W. Schroeder, and A. Gupta, "Collection-Based Persistent Digital Archives – Parts 1& 2", D-Lib Magazine, April/March 2000, http://www.dlib.org/
32. Moore, R. (2000a), "Knowledge-based Persistent Archives," Proceedings of La Conservazione Dei Documenti Informatici Aspetti Organizzativi E Tecnici, in Rome, Italy, October, 2000.
33. Moore, R., C. Baru, A. Rajasekar, R. Marciano, M. Wan: Data Intensive Computing, In ``The Grid: Blueprint for a New Computing Infrastructure'', eds. I. Foster and C. Kesselman. Morgan Kaufmann, San Francisco, 1999.
34. MPEG – Moving Picture Experts Group of ISO/IEC, http://mpeg.telecomitalialab.com/
35. National Library of the Netherlands, Koninklijke Bibliothee, http://www.konbib.nl/
36. NDAP, National Digital Archives Project, Taiwan
37. NPACI Data Intensive Computing Environment thrust area. http://www.npaci.edu/DICE/
38. OAIS - Reference Model for an Open Archival Information System (OAIS). submitted as ISO draft, http://www.ccsds.org/documents/pdf/CCSDS-650.0-R-1.pdf, 1999.
39. OAIS – "Preservation Metadata and the OAIS Information Model", the OCLC working group on Preservation Metadata, June 2002, http://www.oclc.org/research/pmwg/

40. RDF - Resource Description Framework (RDF). W3C Recommendation http://www.w3.org/TR/
41. Records Continuum Model, Records Continuum Research Group, http://rcrg.dstc.edu.au/
42. RTF – Rich Text Format, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnrtfspec/html/rtfspec.asp
43. SAM – Sequential data Access using Metadata, http://d0db.fnal.gov/sam/.
44. SDLIP – Simple Digital Library Interoperability Protocol, http://www-diglib.stanford.edu/~testbed/doc2/SDLIP/
45. SDM – Scientific Data Management in the Environmental Molecular Sciences Laboratory, http://www.computer.org/conferences/mss95/berard/berard.htm.
46. SRB - "The Storage Resource Broker Web Page, http://www.npaci.edu/DICE/SRB/.
47. Thibodeau, K., "Building the Archives of the Future: Advances in Preserving Electronic Records at the National Archives and Records Administration", U.S. National Archives and Records Administration, http://www.dlib.org/dlib/february01/thibodeau/02thibodeau.html
48. tiff – Tag Image File Format, http://www.libtiff.org/
49. Underwood. W. E., "As-Is IDEF0 Activity Model of the Archival Processing of Presidential Textual Records," TR CSITD 98-1, Information Technology and Telecommunications Laboratory, Georgia Tech Research Institute, December 1, 1998.
50. Underwood, W. E., "The InterPARES Preservation Model: A Framework for the Long-Term Preservation of Authentic Electronic Records". Choices and Strategies for Preservation of the Collective Memory, Toblach/Dobbiaco Italy 25-29 June 2002. To be published in Archivi per la Storia.
51. Upward, F., "Modeling the records continuum as a paradigm shift in record keeping and archiving processes, and beyond - a personal reflection', Records Management Journal, Vol. 10, No. 3, December 2000.
52. XML – Extensible Markup Language, http://www.w3/org/XML/
53. Yen, E., "Toward a Data Grid for National Archive", http://pnclink.org/annual/annual2002/pdf/0922/10/g221001.pdf

Appendix A. Example Rationales for Capability Assessment

The numbers listed in the table correspond to entries in Table 1.

No.	Explanatory rationale for inclusion of the capability within the archival process
1	Description process requires the ability to store metadata, and to extract metadata from digital entities in at least one storage repository.
2	Access process requires the ability to retrieve digital entities from at least one storage repository.
3	Description process may require interactive access by an archivist to digital entities residing in a storage repository in order to describe them.
4	Description process should use a standard data transfer protocol between the API and storage repository abstraction when an archivist retrieves digital entities for analysis.
5	Access process will need to manipulate data in containers such as an OAIS distribution information package.
6	Arrangement process is applied on digital entities by references to the logical name space.
7	Access process queries the directory or hierarchical collection structure of the logical name space.
8	Appraisal process can be initiated by a registration step for digital collections (scheduling them with regard to archival appraisal) with pointers to the digital collection- (community-) owned data that resides in the record-keeping system of the creator. This would support the validation of transfers to the archives at the time of accession.
9	Appraisal process relies on standard metadata attributes for deciding relevance of digital entities
10	Accession process uses the same standard metadata attributes as in the accession process.
11	Appraisal process may need to create new attributes for new types of digital entities.
12	Appraisal process uses access control lists to ensure decisions are by approved archivist
13	Accession process also uses access control lists to ensure processing is done by approved archivist
14	Accession process needs physical file location for document acquisition. At the time of accession, attributes will be associated with a logical file name of a transfer (acquisition), e.g., size, creation date, accession date.
15	Appraisal process maps encoding format specification attributes (data type, data model) to the logical name .
16	Preservation process requires the specification of the encoding format when there is a transformative migration, i.e., during preservation activity.
17	Access process may construct a distribution information package that contains metadata about the contained digital entities.
18	Appraisal process may use snapshots to protect against loss of state information.
19	Accession process may use snapshots to protect against metadata loss.
20	Arrangement process keeps a transaction log of arrangement actions to recover from system failure.
21	Description process takes snapshots of the metadata (Descriptive) catalog to protect against metadata loss.
22	Accession process requires a mechanism for validating the integrity of the transfer (acquisition).
23	Description process validates the integrity of the descriptive metadata about the digital entities. Otherwise, the description may be of inauthentic or incomplete digital collections of digital entities.
24	Appraisal process checks that the creator has specified mechanisms that are used to assure the integrity of the created digital entities and metadata. The appraiser should specify (in the terms and conditions for transfer) the mechanisms that will be used to assure the integrity of digital during transfer of the digital entities and metadata.
25	Description process creates metadata that is stored in the metadata catalog and/or in a container of the described digital entities, e.g., AIP. A mechanism must be specified to assure the integrity of the data in the metadata catalog or AIP. Access controls are needed to assure that only archives describe the digital entities.
26	Access process uses integrity mechanisms to assure the integrity of the digital entities and metadata distributed to users during the access activity, e.g., distribution information package.
27	Appraisal process may specify ontologies (data dictionaries, terminologies, database constraints) expressing the relationships used in the business process that created the digital entities, or that characterize the semantics of the digital entities.
28	Accession process may register in the knowledge repository any ontologies (including encoding formats) that are transferred with the digital entities.
29	Preservation process may use knowledge about the digital collection or digital entity to perform a transformative migration and to assure that the semantics of the transformed entity has not changed.

Appendix B. Data Grid Capability Summary:

A consensus on the approach towards building data grids can be gathered by examining which features are implemented by at least five of the seven surveyed data grids. Across the eleven categories of capabilities covered by the comparison, the following capabilities represent a standard approach. The number of grids that provided a given feature is listed in parentheses, with the default value being all of the grids.

Logical name space	
	Logical name space independence from physical name space
	Hierarchical logical folders (5)
	Management of attributes used for each capability (registration, deletion)
	Deletion of entities from logical name space
	Soft links between objects in logical folders (6)
	Support for custodian owned data (5)
	Registration of files into logical name space
Logical name space attributes	
	Replica storage location, local file name
	Group access control lists (5)
	Bulk asynchronous load of attributes (5)
	User defined attributes (5)
Attribute manipulation	
	Automated size, time stamp
	Synchronous attribute update
	Asynchronous annotation (6)
Data Manipulation	
	Synchronous replica creation (6)
Data Access	
	Parallel I/O support (6)
	Transmission status checking (6)
	Transmission restart at application level
	Synchronous storage write
	Standard error messages
	Thread safe client (5)
	Static network tuning (5)
	GridFTP support (5)
Access APIs	
	C++ I/O library API (5)
	Command line interface
	Java interface (6)
	Web service interface (5)
Architecture	
	Distributed client server
	Federated server (6)
	Distributed storage system access
	Third party transfer (5)
	GSI authentication (5)
Latency Management	
	Streaming (6)
	Caching
	Replication (6)
	Staging (5)
System Support	
	Storage Resource Manager interface (5)
	Archive interface to at least one system
	Single catalog server (6)
	Performance for import/export of files greater than 20 files per sec (5)
	Management of file transfer errors (5)

Appendix C. Comparison with InterPARES Preservation Model

A mapping of each of the virtual data grid core capabilities to the InterPARES preservation model is provided below. All mechanisms that are needed for each activity are listed. Note that a given data grid capability will be used multiple times across the InterPARES preservation activities. Processing steps bring electronic records into a processing workbench, create the archival context for each digital content component, and then manage consistency of the content and context. The content is written to storage repositories and the context is written to an authoritative catalog. Both content and context may be replicated to ensure the ability to recover from disasters. Most processing is done using authentication and authorization controls, and all operations are tracked using audit trails.

A.2 Bring in Electronic Records
A 2.1 Register Transfer
Register content from a remote site into the logical name space used by the archival processing environment (workbench), and copied onto a storage repository managed by the workbench. The content is stored under the control of the process custodian. The attributes that describe the source of the transfer are added to an authoritative catalog as part of the context for each content component, using standard archival metadata. Technical administrative metadata is updated to record the location of the content on the workbench. Status information is used to track completion of the transfer. Support for bulk registration and loading of content is provided through use of containers for both data and metadata.

A2.2, Verify that the Transfer is Authorized
Use authentication systems to identify the submitter, and control access to the workbench. The submitted material is examined to verify compliance with the accession schedule that is stored in the authoritative catalog or in a knowledge repository that describes the expected encoding formats, structural relationships, provenance metadata.

A2.3, Examine Electronic Records
A2.3.1, Map Records and Digital Components within Transferred Materials
Analyze the content to identify the digital components, and write the associated structural metadata into the authoritative catalog.

A2.3.2, Verify that the Records in the Transfer Can Be Preserved and Reproduced
Parse the submitted content and check that the desired archival form can be created. This requires reading the content, assigning metadata attributes to record the status of the analysis, updating the authoritative catalog, updating audit trails, and managing the parsing process.

A2.3.3, Take Action Needed to Preserve the Record
Create the archival form that associates preservation context with the material content. Update the context in the authoritative catalog, and update audit trails.

A2.4, Accession Electronic Records
Update the authoritative context to reflect the status of the processing, and set attributes asserting accession of the content.

Core Capabilities	A2	A2.1	A2.2	A2.3	A2.3.1	A2.3.2	A2.3.3	A2.4
Storage repository abstraction	x	x		x	x			
Storage interface to at least one repository	x	x		x	x			
Standard data access mechanism	x	x		x	x			
Standard data movement protocol support	x	x		x	x			
Containers for data	x	x						
Logical name space	x	x	x	x	x	x	x	x
Registration of files in logical name space	x	x		x	x		x	
Retrieval by logical name	x			x	x	x		
Logical name space structural independence from physical name space	x	x						
Persistent handle								
Information repository abstraction	x	x	x	x	x	x	x	x
Custodian owned data	x	x	x	x	x	x	x	x
Collection hierarchy for organizing logical name space	x	x		x	x			
Standard metadata attributes (controlled vocabulary)	x	x	x	x	x	x	x	x
Attribute creation and deletion	x			x			x	
Scalable metadata insertion	x	x		x	x	x	x	x
Access control lists for logical name space to control who can see, add, and change metadata	x	x	x	x	x	x	x	x
Attributes for mapping from logical file name to physical file names	x	x		x	x	x	x	
Encoding format specification attributes	x		x	x	x	x	x	
Data referenced by catalog query								
Containers for metadata	x	x						
Distributed resilient scalable architecture	x	x						
Specification of system availability	x	x						
Standard error messages	x	x						
Status checking	x	x		x	x	x	x	
Authentication mechanism	x	x	x	x	x	x	x	x
Specification of reliability against permanent data loss								
Specification of mechanism to validate integrity of data and metadata	x			x	x	x	x	
Specification of mechanism to assure integrity of data and metadata	x	x	x					
Virtual Data Grid	x			x		x	x	
Knowledge repositories for managing collection properties	x		x	x	x	x		
Application of transformative migration for encoding format	x			x			x	
Application of archival processes	x	x	x	x	x	x	x	

A3, Maintain Electronic Records
A3.1, Manage Information About Records
A3.1.1, Maintain Information About Records
Manage technical administrative metadata that tracks the location of the content in storage repositories, update audit trails when content is moved, and update information about relationships between digital components. Manage contextual information, manage all types of metadata. If context is replicated between multiple catalogs, a persistent handle may be used to assert equivalence between authoritative versions. Use scalable mechanisms to manage insertion of metadata.

A3.1.2, Retrieve Information About Records
Access the authoritative catalog to retrieve information about records. The records of interest may be identified through queries on descriptive metadata and contextual metadata. The person issuing the request is authenticated and checked for authorization.

A3.1.3, Retrieve Information About Digital Components

Access the authoritative catalog to retrieve technical administrative metadata. The records of interest may be identified through queries on descriptive metadata. The person issuing the request is authenticated and checked for authorization.

Core Capabilities	A3	A3.1	A3.1.1	A3.1.2	A3.1.3
Storage repository abstraction	x				
Storage interface to at least one repository	x				
Standard data access mechanism	x				
Standard data movement protocol support	x				
Containers for data	x				
Logical name space	x	x	x	x	x
Registration of files in logical name space	x	x	x		
Retrieval by logical name	x	x			x
Logical name space structural independence from physical name space	x	x	x		
Persistent handle	x	x	x	x	x
Information repository abstraction	x	x	x	x	x
Custodian owned data	x	x	x	x	x
Collection hierarchy for organizing logical name space	x	x	x	x	x
Standard metadata attributes (controlled vocabulary)	x	x	x	x	x
Attribute creation and deletion					
Scalable metadata insertion	x	x	x		
Access control lists for logical name space to control who can see, add, and change metadata	x	x	x	x	x
Attributes for mapping from logical file name to physical file names	x	x	x		x
Encoding format specification attributes	x	x	x		x
Data referenced by catalog query	x	x	x	x	x
Containers for metadata	x	x	x	x	x
Distributed resilient scalable architecture	x	x	x	x	x
Specification of system availability	x	x	x	x	x
Standard error messages	x	x	x	x	x
Status checking	x	x	x		
Authentication mechanism	x	x	x	x	x
Specification of reliability against permanent data loss	x				
Specification of mechanism to validate integrity of data and metadata	x	x		x	x
Specification of mechanism to assure integrity of data and metadata	x	x	x		
Virtual Data Grid	x				
Knowledge repositories for managing collection properties					
Application of transformative migration for encoding format	x				
Application of archival processes	x	x	x	x	x

A3, Maintain Electronic Records
A3.2, Manage Storage of Digital Components of Records
A3.2.1, Place Record Components in Storage
Store content using the storage repository abstraction for managing heterogeneous storage systems, update the technical administrative metadata in the authoritative catalog, and update the audit trails. Replicate the content and context as part of the data integrity mechanism. Store the content under custodian control. Use containers for managing interactions with archives, to avoid overloading the archive name space. Use scalable metadata insertion mechanisms to manage large numbers of digital entities.

A3.2.2, Refresh Storage
Support migration of content to new technology by accessing the original storage repository, reading the content, replicating the content to the new storage repository, and updating the technical administrative metadata for the location of the content. Update audit trails to track the change in location of the content.

A3.2.3, Monitor Storage
Track the status of the storage repositories, the specification of reliability against data loss, and analyze which content is at risk. Record the results as administrative attributes in the authoritative catalog.

A3.2.4, Correct Storage Problems
Support update of content of files in containers, with corresponding update of the audit trails in the authoritative catalog. Set integrity attributes to identify loss of data or uncorrected problems such as synchronization across replicas.

A3.2.5, Retrieve Components from Storage
Use the storage repository abstraction to interact with the storage system, the data movement protocol to transport the data, and the data access mechanism to deliver the data to the user. Authenticate the user and authorize the transaction. Use system availability to decide which replica to access.

Core Capabilities	**A3**	**A3.2**	**A3.2.1**	**A3.2.2**	**A3.2.3**	**A3.2.4**	**A3.2.5**
Storage repository abstraction	x	x	x	x	x	x	x
Storage interface to at least one repository	x	x	x	x	x	x	x
Standard data access mechanism	x	x	x	x		x	x
Standard data movement protocol support	x	x	x	x		x	x
Containers for data	x	x	x	x		x	x
Logical name space	x	x	x	x	x	x	x
Registration of files in logical name space	x	x	x	x		x	
Retrieval by logical name	x	x					x
Logical name space structural independence from physical name space	x	x	x	x		x	x
Persistent handle	x						
Information repository abstraction	x						
Custodian owned data	x						
Collection hierarchy for organizing logical name space	x						
Standard metadata attributes (controlled vocabulary)	x						
Attribute creation and deletion							
Scalable metadata insertion	x						
Access control lists for logical name space to control who can see, add, and change metadata	x						
Attributes for mapping from logical file name to physical file names	x						
Encoding format specification attributes	x						
Data referenced by catalog query	x						
Containers for metadata	x	x	x	x			x
Distributed resilient scalable architecture	x	x	x	x	x	x	X
Specification of system availability	x	x	x	x	x	x	x
Standard error messages	x	x	x	x	x	x	x
Status checking	x	x			x		
Authentication mechanism	x	x	x	x	x	x	x
Specification of reliability against permanent data loss	x	x	x	x	x	x	
Specification of mechanism to validate integrity of data and metadata	x						
Specification of mechanism to assure integrity of data and metadata	x	x	x	x	x	x	x

Virtual Data Grid	x						
Knowledge repositories for managing collection properties							
Application of transformative migration for encoding format	x						
Application of archival processes	x	x	x	x	x	x	x

A3, Maintain Electronic Records
A3.3, Update Digital Components
Transform content to new encoding formats, store the updated content, and update the authoritative catalog. Use scalable management mechanisms to enable the update of entire digital collections.

Core Capabilities	A3	A3.3
Storage repository abstraction	x	x
Storage interface to at least one repository	x	x
Standard data access mechanism	x	x
Standard data movement protocol support	x	x
Containers for data	x	x
Logical name space	x	x
Registration of files in logical name space	x	x
Retrieval by logical name	x	x
Logical name space structural independence from physical name space	x	x
Persistent handle	x	
Information repository abstraction	x	x
Custodian owned data	x	x
Collection hierarchy for organizing logical name space	x	
Standard metadata attributes (controlled vocabulary)	x	x
Attribute creation and deletion		
Scalable metadata insertion	x	x
Access control lists for logical name space to control who can see, add, and change metadata	x	x
Attributes for mapping from logical file name to physical file names	x	x
Encoding format specification attributes	x	x
Data referenced by catalog query	x	
Containers for metadata	x	x
Distributed resilient scalable architecture	x	x
Specification of system availability	x	
Standard error messages	x	x
Status checking	x	
Authentication mechanism	x	x
Specification of reliability against permanent data loss	x	x
Specification of mechanism to validate integrity of data and metadata	x	x
Specification of mechanism to assure integrity of data and metadata	x	x
Virtual Data Grid	x	x
Knowledge repositories for managing collection properties		
Application of transformative migration for encoding format	x	x
Application of archival processes	x	x

A4, Output Electronic Record
A4.1, Manage the Request
Track error returns when delivering output, access alternate replicas when data is unavailable, track authentication and authorization of the requestor, and report standard error messages to requestors.

A4.2, Review Retrieved Components and Information

Access the authoritative catalog to associate components with records. Determine the availability of the components by tracking the status of each storage repository.

A4.3, Reconstitute Record
Assemble the record by applying the structural metadata from the authoritative catalog.

A4.4, Present Record
Transmit the record to the requestor, and assert the integrity.

A4.5, Package Output
Provide mechanism for reconstituting a record from component parts through specification of the processes that should be applied in the virtual data grid, and package into a container.

Core Capabilities	**A4**	**A4.1**	**A4.2**	**A4.3**	**A4.4**	**A4.5**
Storage repository abstraction	x	x	x			
Storage interface to at least one repository						
Standard data access mechanism	x	x				
Standard data movement protocol support	x	x			x	
Containers for data	x					x
Logical name space	x	x	x	x	x	
Registration of files in logical name space						
Retrieval by logical name	x	x				
Logical name space structural independence from physical name space	x	x				
Persistent handle						
Information repository abstraction	x	x	x	x		x
Custodian owned data	x	x				
Collection hierarchy for organizing logical name space	x	x	x			
Standard metadata attributes (controlled vocabulary)	x	x	x	x		x
Attribute creation and deletion						
Scalable metadata insertion						
Access control lists for logical name space to control who can see, add, and change metadata	x	x			x	
Attributes for mapping from logical file name to physical file names	x	x	x			
Encoding format specification attributes	x		x	x		x
Data referenced by catalog query	x	x				
Containers for metadata	x		x			x
Distributed resilient scalable architecture	x	x				
Specification of system availability	x	x	x			
Standard error messages	x	x	x	x	x	x
Status checking	x	x	x			
Authentication mechanism	x	x			x	
Specification of reliability against permanent data loss						
Specification of mechanism to validate integrity of data and metadata	x		x		x	x
Specification of mechanism to assure integrity of data and metadata	x	x			x	x
Virtual Data Grid	x			x		x
Knowledge repositories for managing collection properties	x					x
Application of transformative migration for encoding format	x			x		
Application of archival processes	x			x		x

Appendix D. Definition of Core Capabilities for Persistent Archives

1. Storage repository abstraction

Core capability definition:
The set of operations that can be used to manipulate data within a storage repository.

Functionality provided by the capability
A storage repository holds digital entities. By mapping from the storage repository abstraction to the protocols required by a particular storage repository, it is possible to manage data in any type of storage system, including file systems, hierarchical storage managers, databases. To add a new type of storage system to the data grid, a new driver is written to map from the storage repository abstraction to the new access protocols.

Example grid implementation
A standard set of operations for management of distributed data may include Unix file system operations (create, open, close, unlink, read, write, seek, sync, stat, fstat, mkdir, rmdir, chmod, opendir, closedir, and readdir), latency management operations (aggregation of data, I/O commands, and metadata), and metadata manipulation (extraction, registration).

2. Storage interface to at least one repository

Core capability definition
Every persistent archive will contain at least one storage system for holding digital entities.

Functionality provided by the capability
The storage repository is intended to provide long term residency for data, the bits that comprise the digital entity. The information and knowledge content within the data may be annotated and encapsulated in Open Archival Information System (OAIS) packages. While this content may also reside in the storage repository, support for discovery based on the information content would be supported by an information repository.

Example grid implementation
Traditional long term residency systems for data are based on use of tape, managed by a hierarchical storage manager. However the capital cost of disk systems is starting to approach that of tape systems, with similar capacities. By using data grids to manage the digital entities, the user authentication, and the user authorization, the labor costs of disk systems can also be reduced to that of tape.

3. Standard data access mechanism

Core capability definition
The user interface used to access digital entities residing in a storage repository.

Functionality provided by the capability
A standard data access mechanism provides a uniform interface to digital entities residing in the storage repositories. The choice of standard access mechanism can be made separately from the choice of storage repository. The data access mechanism can be kept the same across multiple versions of storage repositories over time.

Example grid implementation
The choice for standard data access mechanism can be a web services interface based on the Open Grid Services Architecture, or a web browser interface. The standard data access mechanism specifies the user Application Programming Interface (API) that will be preserved over time.

4. Standard data movement protocol support

Core capability definition
The data transfer protocol that is used between the API and the storage repository abstraction.

Functionality provided by the capability
A standard data transport protocol minimizes the amount of effort needed to implement a data grid. The transport mechanism should provide parallel I/O support for bulk data transport, reliable and guaranteed delivery of data, and interoperate with a standard authentication protocol.

Example grid implementation
An emerging standard is the GridFTP protocol for data movement. Other protocols are in extensive use, including the SRB data transport and http. It is possible to build protocol conversion mechanisms to map between multiple data transport protocols.

5. Containers for data

Core capability definition
An aggregation mechanism to keep multiple digital entities in a single file.

Functionality provided by the capability
Containers make it possible to guarantee that multiple digital entities are stored on the same media. Containers also provide a needed management function for hierarchical resource managers, by minimizing the number of names that must be maintained in the HRM. Containers provide a latency management function, making it possible to move many digital entities as a single file.

Example grid implementation
The Storage Resource Broker data grid uses containers to aggregate data before storage on Hierarchical Resource Managers. References to a digital entity within a container causes the container to be cached on disk, off-loading I/O commands from the HRM. A containerization service is being developed as part of the Globus tool kit.

6. Logical name space

Core capability definition
Naming convention for labeling digital entities.

Functionality provided by the capability
The logical name space is used to create global, persistent identifiers that are independent of the storage location. This makes it possible to reference digital entities that reside on multiple storage systems using a common set of names.

Example grid implementation
Replica catalogs provide a mapping from the physical file name to a global identifier. The name space can be organized independently of the directory structures on the storage systems. Global names are created that can be used as persistent identifiers.

7. Registration of files in logical name space

Core capability definition
Mechanism to add digital entities to the name space.

Functionality provided by the capability

A logical name space can be used to register arbitrary digital entities. The most common digital entity used in preservation is a file. Registration corresponds to adding an entry to the logical name space, creating a logical name and storing a pointer to the physical file and its location.

Example grid implementation
Logical name spaces in data grids have been used to register files, URLs, SQL command strings, processing templates for metadata extraction, executables, etc. It is possible to register a link within the logical name space (soft link), which provides a way to re-purpose data without having to replicate the digital entity.

8. Retrieval by logical name

Core capability definition
Mechanism to retrieve a digital entity by mapping from the logical name to the physical location where the digital entity resides.

Functionality provided by the capability
Retrieval can include the physical transport of the digital entity to the client application that made the request. Retrieval can also include the invocation of a display or presentation application. Retrieval can also include the invocation of the digital entity in the case of a URL, or the execution of the digital entity in the case of a SQL command string.

Example grid implementation
Data grids invoke different retrieval mechanisms depending upon the type of user interface or API. Web browsers tend to invoke display applications on retrieval, while C library calls usually process the digital entity using Unix file operations. The digital entity can be partially returned, in the case of partial file reads, or may be returned in its entirety.

9. Logical name space independence from physical name space

Core capability definition
The organization of the logical name space has no dependence upon the organization of the physical name space.

Functionality provided by the capability
The logical name space can be organized as a directory hierarchy or a hierarchical digital collection. Pointers are used to identify the location of the digital entity within a storage system. The pointers can point to an arbitrary physical directory. For replicas of digital entities, the location of each replica in a storage system can be specified independently of all other replicas.

Example grid implementation
The ability to decouple the logical name space completely from physical names for digital entities makes it possible to manage a wide variety of digital entities. Logical name space independence is particularly important when registering URLs as replicas of each other. Each URL points to a different site, but is recorded as being equivalent. Replica catalogs also require logical name space independence when registering files as replicas of each other, when the files reside in different types of file systems (Windows NT versus Linux).

10. Persistent handle

Core capability definition
Infrastructure independent naming convention for a digital entity. The naming convention can be semantic free as well as location independent.

Functionality provided by the capability

Digital entity names can be persistent if their semantic meaning is decoupled from the physical location representing their storage location. Depending upon the management policies, the logical semantic tag can be kept invariant as the digital entity is migrated across multiple storage systems. Within the context of the digital collection for which the logical name is created, the naming convention can represent a globally unique identifier. A persistent handle can be implemented as a logical name. The choice of the syntax for the logical name is arbitrary. It can be defined via a handle system relative to an organization, or can be defined relative to a digital collection, or can be a user-defined name.

Example grid implementation
The implementation of persistent handles requires the ability to manage the consistency of the logical name space. While the persistent handle is held invariant over time, the archival state information mapped to the handle needs to remain consistent. Every operation on digital entities within the logical name space needs to be mirrored by appropriate changes to the location attributes associated with the logical name. Updates in distributed environments can be automated if the digital entities are owned by the digital collection in which the location attributes are maintained. Then it is possible to guarantee consistency of the persistent handles. The physical file name that represents a digital entity can be kept consistent with the persistent handle. Data grids have been implemented in which digital entities are owned by the digital collection (consistent environments), and in which digital entities are owned by individuals (consistency dependent upon user policies for updating the replica catalog references).

11. Information repository abstraction

Core capability definition
The set of operations that can be used to manipulate a catalog within an information repository such as a database.

Functionality provided by the capability
Technology evolution applies equally well to information repositories as it does to storage repositories. An abstraction for catalog manipulation operations is needed to make it possible to migrate the persistent archive metadata to new database technology.

Example grid implementation
Typical operations that are performed on information repositories include schema extension, bulk metadata loading, automated SQL generation, bulk metadata extraction and formatting, For a virtual data grid, in which the digital collection context is dynamically created by parsing the digital entities for information content, the information repository abstraction needs to include the ability to dynamically create a database instance.

12. Custodian owned data

Core capability definition
Ownership of digital entities within storage repositories by the organizing digital collection.

Functionality provided by the capability
To maintain integrity, all manipulations of digital entities within a persistent archive need to be audited. Tracking mechanisms can be built into the policy management specifications for a persistent archive, or they can be integrated into the data management system such that any change to the location or format of a digital entity is automatically recorded in the digital collection metadata. To minimize manual labor requirements, automation of metadata tracking is required. This can be accomplished by having the digital collection own the digital entities, requiring the involvement of the digital collection software before any operation can be performed on the digital entities.

Example grid implementation

Support for custodian owned data is becoming a standard capability within data grids. Of the seven grids surveyed, five grids supported custodian owned data. An implication is the need for the management of authorization mechanisms to restrict access. In the typical scenario, a user authenticates herself to the data grid. The data grid authenticates itself to the remote storage system, and checks its own access control lists to determine whether the user can manipulate the digital entity. Data grids decouple the management of the users and their access restrictions from the storage repositories. This simplifies administration of storage repositories that hold digital entities for the persistent archive.

13. Digital collection hierarchy for organizing logical name space

Core capability definition
Use of digital collection/sub-collection hierarchies for organizing the logical name space attributes used to control digital entities

Functionality provided by the capability
Logical name spaces inherently require the specification of attributes to manage information about the physical location of each digital entity. Additional attributes are used to manage soft links and sub-collection specific metadata. Since each sub-collection can have a different set of attributes, a digital collection/sub-collection hierarchy is used to organize the logical name space. A more general structure for organization of the logical name space would be a graph, in which relationships are used on each link to define a context for organizing metadata attributes. Such organization mechanisms will be required in the future when knowledge relationships are managed for persistent archives, in addition to the informational semantic tags.

Example grid implementation
Management of a digital collection hierarchy can be facilitated by the use of schema indirection. The attributes assigned to a digital collection can be specified by use of two arrays, one to record the attribute name, and one to record the attribute value. The use of a digital collection hierarchy can also be expressed as the use of schema indirection for organizing attributes. Data grids have been implemented that use explicitly defined tables for digital entity attributes, and that use schema indirection to manage the attributes. Explicitly defined tables are preferred for digital collections that manage millions of files.

14. Standard metadata attributes (controlled vocabulary)

Core capability definition
Use of standard metadata semantic names for describing digital collection specific attributes

Functionality provided by the capability
When a digital collection hierarchy is used to organize attributes for each sub-collection, it is very easy to create semantic terms for the information content that are unique to the sub-collection. By using standard metadata attributes, the utility of the information content can be extended to terms that are in common across multiple digital collections. An example is the use of Dublin Core. The two most widely used formats for describing digital collections are Encoded Archival Description (EAD) (http://www.loc.gov/ead/) and MARC (MA chine-Readable Cataloging) record. (http://lcweb.loc.gov/marc/index.html) attributes to specify provenance. The associated attribute values also may have embedded semantics. Use of controlled vocabularies is needed to provide a consistent interpretation to both the semantic meaning of an attribute name and the semantic meaning of the attribute value.

Example grid implementation
Data grid collection hierarchies have been organized by inheriting metadata attributes from standard metadata schema, by inheriting metadata from the parent collection, and by assigning unique metadata. When such hierarchies are queried at the top level, it is not uncommon to find

hundreds to thousands of metadata attributes across all sub-collections. The use of standard metadata attributes is essential to avoid semantic name explosion.

15. Attribute creation and deletion

Core capability definition
Both attribute names and attribute value assignments can be created and deleted relative to the logical name space.

Functionality provided by the capability
Schema extension is needed to allow descriptive metadata and technical administrative metadata to evolve over time. The state information that is generated by archival processes can evolve as new types of integrity metadata become available, such as digital signatures, access controls, and as new types of preservation environments become available. From the perspective of researchers, the choice of the appropriate context to use for discovery typically depends upon the user community. The semantic names used for discovery will change as the user community evolves. The infrastructure for managing technology evolution must also manage the evolution of the naming conventions for the archival collection. The evolution of naming conventions is usually associated with re-purposing of archival content by researchers.

Example grid implementation
Data grids support attribute creation through multiple mechanisms: synchronously when digital entities are registered, asynchronously after digital entities have been registered, and through bulk metadata registration. Attribute values are loaded from externally defined XML files, or through application of templates that apply parsing rules to annotate and extract semantic content. Attribute deletion is done either by setting a flag, or by actual deletion of the attribute from the collection.

16. Scalable metadata insertion

Core capability definition
Mechanisms to automate creation and loading of metadata attribute names and associated values.

Functionality provided by the capability
Scalability is achieved through the automation of metadata manipulation processes. The processes include metadata extraction from the digital entities, the aggregation of the metadata into XML files, and the bulk loading of attributes into metadata catalogs. Scalability is enhanced by the implementation of each of these processes as parallel I/O streams.

Example grid implementation
Scalable metadata insertion is typically achieved by working with parallel database technology that can handle multiple simultaneous insertion streams. This requires parallel technology to generate and manage the parallel I/O streams, either through creation of thread-safe clients, or by the spawning of multiple processes that simultaneously generate the I/O streams.

17. Access control lists for logical name space to control who can see, add, and change metadata

Core capability definition
Mechanisms for managing user authorization for access to persistent archive holdings

Functionality provided by the capability
For persistent archives that implement collection ownership of data held in storage repositories, a mechanism is needed to decide which digital entities can be accessed by each user. This

requires an authentication mechanism to identify users, and an authorization mechanism to define access controls. Each collection and each digital entity may need separate access controls. In addition, separate access roles are needed for the archivist to manage metadata and accretions (additions to a set of accessioned records) to a collection after they have gone through the accessioning process and for the public for access to the material.

Example grid implementation
Data grids address authorization through the use of access control lists on groups of users and on individual users. The authorization mechanism can be implemented as metadata that is managed about users and user groups. The metadata can be implemented directly within the collection as a collection-specific table, or can be implemented in a separate authorization server that is used to control access to multiple collections. The choice of access roles can include: archivist, owner, writer, annotator, and reader.

The Role-Based Access Control (RBAC Standard) by NIST is a reasonable approach for Grid security. Since the complexity of maintaining privileges scales non-linearly and can be labor intensive, simple data grid mechanisms are needed to manage security policy and enhance administrative efficiency, flexibility, scalability, and accuracy. The RBAC standard provides support for many-to-many relationships among individual users and privileges; support for a session that maps between a user and an activated subset of assigned roles; user/role relations that can be defined independently of role/privilege relations; privileges that are system/application dependent; and accommodation for traditional but robust group-based access control.

18. Attributes for mapping from logical file name to physical file names

Core capability definition
The logical name space manages attributes for mapping from the logical name for a digital entity to the physical name under which a digital entity is stored.

Functionality provided by the capability
The logical name space mapping to a physical name space can be one-to-one, with a single digital entity corresponding to the logical name. The mapping can be one-to-many, with multiple replicas associated with a single logical name. The mapping can be one-to-many with semantically equivalent, but syntactically different digital entities associated with the logical name. The mapping can be associative, with digital entities physically aggregated into a container and the container stored in a repository. The attributes in each case can contain not only the location of the digital entity and its name on the remote storage system, but also the name of the protocol required to talk to that storage system, the type of the digital entity, and Unix system semantics for information about the size, ownership, creation date, update date, etc.

Example grid implementation
The set of attributes associated with each digital entity can be unique in data grid implementations. Note that the attributes associated with a replica must be unique to that replica, since it resides at a different storage location or under a different path name on the same storage system. Similarly, it is possible to let the create and update times for each replica be unique, making it possible to track consistency across replicas. By allowing the data type of the digital entity to vary across replicas it is possible to manage syntactically different versions of the same logical digital entity.

19. Encoding format specification attributes

Core capability definition
Specification of the data type or data model associated with each digital entity, or the digital ontology that organizes the relationships present within a digital entity.

Functionality provided by the capability

Since a persistent archive must allow the evolution of the encoding format of each digital entity, an attribute that is managed by the persistent archive should be the data type or data model. When transformative migrations are performed on the digital entity, semantically equivalent replicas are made, that are differentiated by the syntax associated with the new encoding format. The ability to specify the encoding format needs to apply to replicas of digital entities. Alternatively, a digital ontology can be used to characterize the digital entity, with transformative migrations on the encoding format for relationships applied to the digital ontology. The set of operations that can be performed on the defined relationships will also need to be characterized, and emulated by future presentation applications.

Example grid implementation
Encoding format specification is used for both static transformative migrations and dynamic transformative migrations. The conversion from an old encoding format to a new encoding format can be thought of as a static transformation that is performed once. The conversion from an encoding format to an associated display can be thought of as a dynamic transformation that is invoked every time the digital entity is viewed. Graphical user interfaces to data grids typically access the data type associated with a digital entity to decide which display application should be invoked when the digital entity is retrieved and thus perform dynamic transformative migrations.

20. Data referenced by catalog query

Core capability definition
Mechanisms for attribute-based digital entity discovery

Functionality provided by the capability
Given the very large number of digital entities that are being archived, it is not possible to a priori know the logical names of all digital entities within a collection. A context is described for the digital entities by specifying semantic terms and associated attribute values. Discovery of a particular digital entity is then accomplished by querying on the attribute values. This requires that the user recognize the semantics inherent in the attribute names.

Example grid implementation
The mechanisms used to do discovery in data grids range from explicit creation of SQL commands, to specification of attribute names and values and automated generation of the required SQL. The latter approach requires the ability to characterize the table structure of the data grid metadata catalog, identify the foreign keys that are used between the tables, and generate the required joins. The result of the query generates logical names, which can then be queried to discover the associated physical replicas. Alternatively, the metadata catalog query can result in direct access to the "nearest" copy of the associated physical file.

21. Containers for metadata

Core capability definition
Mechanisms for manipulating metadata attributes that are aggregated into a single file

Functionality provided by the capability
The use of XML annotated files makes it possible to assemble metadata attributes for either bulk metadata ingestion or bulk metadata export. By structuring the XML annotated file through application of either an XML DTD or XML Schema, a characterization of the collection can be created.

Example grid implementation
In data grids, metadata containers are primarily used for latency management. When metadata is extracted from a digital entity by application of an extraction template at the storage system, the attribute values are aggregated before transmission over the network. When databases are registered as objects within the logical name space, SQL command strings can be generated that

extract metadata from the database. Again the metadata is aggregated into an XML file before it is moved over the network.

22. Distributed resilient scalable architecture

Core capability definition
Mechanisms to support fault tolerance and high access performance across distributed repositories

Functionality provided by the capability
The ability to scale requires automation of data management capabilities. Through use of a logical name space, storage repository abstractions, and information repository abstractions, it is possible to drive all data manipulation operations directly from an application. By designing the correct applications, any type of processing of a collection can be automated from metadata extraction, to encoding format transformation, to replication, etc. Resilience is accomplished through either replication for fault tolerance, replication for data assurance, or re-generation through application of the deriving process. Both the resilience and automation mechanisms need to operate in a distributed environment, primarily because migration onto new technologies requires the ability to simultaneously access both the old and new forms of the technology.

Example grid implementation
Most data grids are implemented as federated client server architectures. Servers are installed at each storage system where data will be held. The servers map from the protocol of the local storage system to the storage repository abstraction. The servers can exchange data directly between themselves through third party transfer, making it possible to issue commands for replication that only involve the source and destination sites. Replication is used as the primary resiliency mechanism, with data accesses automatically failing over to a replica location if the desired copy was not available. Distribution is handled by federation of the servers, such that servers can communicate between each other independently of the driving client.

23. Specification of system availability

Core capability definition
Mechanisms to specify the permanency of the digital holdings, the access limitations, and the access availability

Functionality provided by the capability
An approach to data assurance is to move data from storage systems in which the guaranteed residency period is shorter than the desired period, to storage systems that can meet the assurance requirements. By putting the burden on reliability on the underlying storage systems, it is possible to force the storage systems to manage replicas for data assurance. This is typically done in hierarchical storage managers, which keep multiple copies of data on tape.

Example grid implementation
Most data grids rely on assurance specifications through the data grid metadata catalog, typically by adding attributes to characterize the type of storage repository as permanent tape storage, permanent disk cache under the control of the collection, temporary disk cache under the control of a system administrator, ephemeral disk cache that is subject to policy based purging. Copies of data are kept to assure permanence, with the copies geographically distributed to protect against disasters.

24. Standard error messages

Core capability definition
Mechanism to report error messages generated by all components of the persistent archive, from storage systems, to networking, to presentation errors.

Functionality provided by the capability
There are over one thousand error messages that can be generated across storage, network, and information repository environments. The organization of error messages into classes or severity is done to minimize the necessity of learning the meaning of each error message. Severity classes can include unrecoverable (must try another resource), recoverable (try again against the current resource), and advisory (non-fatal problem).

Example grid implementation
Data grids currently use their own standards for error messages. The data grid forum is examining the development of event based reliability systems that will generate a consensus on error message types. Actual implementations of error messages within data grids either report every error message back to the user, or classify the errors into a small set of classes.

25. Status checking

Core capability definition
Mechanism to report on the status of a request

Functionality provided by the capability
Status checking can be managed by event monitoring systems when single requests are made. When bulk processing is attempted, such as in metadata extraction and the movement of thousands of files, database technology is used to track the status of each individual component of a request.

Example grid implementation
Many of the data grid status checking mechanisms are done synchronously through notification on completion of a task. Some systems support dynamic status checking, such as the transmission of markers in the data flow to support transmission restart, or the creation of process flows where each processing stage is described by a characterization of the processing step. Status then corresponds to identifying which processing step was last completed. Resiliency is implemented by restarting from the last complete stage of the processing pipeline.

26. Authentication mechanism

Core capability definition
Mechanism to identify both individuals as single persons, and individuals as members of groups

Functionality provided by the capability
To manage the multiple access roles required for collection building, authentication mechanisms are needed to identify each person. In particular, curatorial and creation roles must be restricted to the archivists to ensure permanency of the archival holdings. For proprietary data, identification as members of groups is usually sufficient to manage access. Group based identification is appropriate where anonymity of access is required.

Example grid implementation
Data grids differentiate between the authentication systems used between administration domains, and the authentication systems used within an administration domain. The Grid Security Infrastructure uses Public Key Infrastructure and certificates to identify individuals. The certificates are managed by certificate authorities that follow specified managerial practices for the assignment of certificates to individuals. Alternatively, encrypted passwords and challenge response mechanisms are used to identify not only individuals, but also servers within the federated client server architecture. Local authentication systems are either Unix based, Kerberos based, or DCE based. The Generic Security Service API is used to map between the different authentication environments.

27. Specification of reliability against permanent data loss

Core capability definition
Mechanism to ensure survival of collection holdings across all types of failure mechanisms

Functionality provided by the capability
The assurance of reliability against permanent data loss has two components, protection of the original bits that comprise the digital entities, and protection of the mechanisms that identify the context used to organize the digital entities. Protection against data loss can be done through replication. Protection against information loss is much harder. The context can change over time through the addition of new material. The information content therefore requires both replication and snapshot mechanisms to ensure digital entities can be both identified and retrieved.

Example grid implementation
Hierarchical storage managers have traditionally safeguarded both the digital holdings and the metadata describing where the holdings are stored. The digital holdings are replicated. The metadata is replicated. Snapshots are taken of the metadata state at periodic intervals, and transaction logging is used to record all changes to the metadata. The transaction logs are replicated and periodically applied to the snapshots to guarantee that the state of the metadata catalog can be recreated. Similar approaches are needed in persistent archives to ensure reliability of the collection.

28. Specification of mechanism to validate integrity of data and metadata

Core capability definition
Mechanism to validate the integrity of the digital holdings

Functionality provided by the capability
Integrity requires showing that all operations that have been performed on a digital entity can be identified and characterized, that the metadata that is used to define the context for the digital entity is consistent with the operations that have been performed, and that the bits of the digital entity have not changed between transformative migrations. The consistency that can be maintained between the data and metadata is one of the primary advantages of the use of data grid technology with collection owned holdings.

For example a grid can provide the capability to perform the same operations done under accessioning (“Data grids provide mechanisms that can be used to validate data models, extract metadata, and authenticate the identification of the submitter.”) on both the copy of the record before a transformative migration and the copy of that record after migration. The results of using these mechanisms could then be used to compare the before and after results and assert that the two records were equivalent. This capability is needed to further automate the process of maintaining the integrity of the record over time. At the very least it is helpful to have controls in place that would not allow copies of records made before a transformative migration to be deleted until the archivist had certified that the migration had successfully produced the ability to reproduce an authentic record. (See the InterPARES Preservation Model)

Example grid implementation
Audit trails are used to record all accesses and operations that are performed on the digital holdings. Digital signatures and checksums are used to show that a digital entity has not been corrupted by disk, transmission, or recording errors. By checking the audit trails and comparing the recorded checksum with the current checksum, one can validate the integrity of the data. By examining the operations performed upon the digital entity, and the person who initiated the operation, one can show that only archivists have applied archival processes to the digital entities. A virtual data grid should provide the capability to verify/validate that syntactically

different versions of the same record are equivalent to the point that they both transmit the same message.

29. Specification of mechanism to assure integrity of data and metadata

Core capability definition
Mechanism to uniquely characterize a digital entity

Functionality provided by the capability
Through the use of the OAIS technology for specifying archival information packages (AIPs), it is possible to aggregate metadata and data into a single file. By signing or check-summing the AIP, one can determine that the content has not changed over time. By comparing the metadata within the AIP to that organized into the data grid catalog and applying the audit trail transformations, one can show that the digital entity corresponds to all recorded operations, and thus still contains the expected information and knowledge content.

Example grid implementation
The assurance of integrity is primarily managed in data grids by restricting operations on the digital holdings to the persons fulfilling the archival roles. The specification of a signature or checksum is inadequate if the signature can be forged through an unauthorized operation. Data grids get around this problem by working with custodian owned data, and by auditing all operations done on a digital entity.

30. Virtual Data Grid

Core capability definition
Mechanism to create derived data products on demand

Functionality provided by the capability
The application of archival processes to digital entities can be characterized as a set of processing steps. The characterization can be stored as a process flow in the archive along with the digital entities, and organized in a logical name space sub-collection. A query against a collection for a digital entity can then be made against the collection attributes. If the query is not satisfied, a search can be done on the processing characterizations for the ability to generate the required derived data product. If the processing step is found, one then has to identify the required input files and input parameters. This requires knowledge about the relationships used to govern the archival process, and can be specified as part of the original query.

Example grid implementation
Virtual data grids use process characterizations based upon Directed Acyclic Graphs. These simple descriptions map output to input files for the multiple stages of a process pipeline. More sophisticated versions of process data flow are needed to incorporate knowledge about application of the processing steps; namely how to decide which digital entities are to be used as input to the processing stages. These process flow characterizations require the integration of concept spaces on top of the information catalogs managed by the data grids. The concept spaces specify the relationships that govern the application of the processing steps.

31. Knowledge repositories for managing collection properties

Core capability definition
Relationship management systems to describe the constraints used to form a collection of digital entities, or the properties of the resulting collection, or the organization of relationships within a digital ontology.

Functionality provided by the capability

The characterization, organization, and manipulation of relationships are managed by knowledge repositories. Given that multiple knowledge repositories can be created, a knowledge repository abstraction is needed to describe the set of operations that can be performed upon a concept space that is implemented within a knowledge repository.

Example grid implementation
The application of relationships that are organized in a knowledge repository requires the ability to generate logical inference rules or processing steps from the relationship description. Systems have been developed that generate logic rules for semantic relationships, spatial rules for manipulation of atlases, and procedural rules for applying processing steps. Each of these systems is typically implemented for a particular discipline. Generic mechanisms are needed for persistent archives. An example system is the mapping from the RDF relationship syntax to Common Logic rules that can then be evaluated across a collection.

32. Application of transformative migrations for encoding format

Core capability definition
Mechanism to migrate the encoding format to a new encoding standard

Functionality provided by the capability
The evolution of the encoding format of digital entities must be addressed by persistent archives along with the evolution of the supporting software and hardware infrastructure. A transformation of an encoding format to a new standard can be characterized as a processing step that is performed under persistent archive policy management control. The transformation would typically be applied when the collection holdings are migrated to a new media standard, as the entire collection must be read and processed. The transformative migrations can be applied to digital entities, digital ontologies, and even to encoding standards used to describe digital collections.

Example grid implementation
Grids provide the ability to execute procedures at the storage system where the digital entity resides, and to stream the digital entity through a sequence of filters. The procedures can be named and organized within the logical name space, and stored in the data grid along with the digital entities. This makes it possible to execute transformative migrations on digital entities as part of a media migration process.

33. Application of archival processes

Core capability definition
Mechanism to characterize and apply archival processes

Functionality provided by the capability
Many of the archival processes correspond to metadata extraction, digital collection formation, transformative migration, and data management. The abstractions needed to support these processes correspond to management of a logical name space, a storage repository abstraction for the operations that can be done on digital entities, and an information repository abstraction for the operations that can be done on catalogs in databases.

Example grid implementation
Data grids implement the abstraction levels needed to support the application of archival processes. The challenge is correctly characterizing the archival processes, and validating that the characterizations perform the desired functions.