On the Notion of Federated Archives Within Persistent Environments

Bruce R. Barkstrom

Atmospheric Sciences Data Center
NASA Langley Research Center

Introduction

This paper is intended to provide an architectural, or system design view of how federated archives may operate within a Grid-enabled, persistent environment. Under normal circumstances, Grid-enabled data centers have, of course, been concerned with storing, transferring, and performing computations on the data they contain. However, when we start thinking about storing data and making it accessible over long periods with respect to the life cycle of individual components of information technology, or with respect to the data curating institutions, it is apparent that we need to raise the importance of several items which normal data center operations may neglect.

The most obvious item is the evolution of the information technology that stores the data and makes it accessible. Moore's Law has operated with a vengeance during the last decade. Thus, the kinds of computers, networks, and storage devices used in data centers are very different today than they were a decade ago. These devices also cost much less for a given level of performance than they did then. Moreover, the software involved in any aspect of these centers is also very different. We do not design systems in the same way as we did a decade ago, and we do not test and deploy them in the same way. It is clear that we need to take into account the fact that the software and hardware of these systems will be obsolete before the media on which the centers store data has approached physical degradation. Still more important, a well-justified concern over the total cost of ownership with rising expectations regarding the ability of all user communities to access data from any time period means that we cannot regard data archives as passive repositories of “dead data”. Rather, we need to accommodate the fact that users need to access data that were obtained many years ago for purposes that may be quite different from those that now drive the current users.

In the material that follows, we start by considering the environment wherein persistent archives operate. We anticipate that these entities will use Web Services interfaces that allow exchanges of large amounts of data – both electronically and by media. It is true that most Grid computing environments prefer to concentrate on electronic data transfers.

However, for the time being, extremely large data transfers and off-line backup are still key functions that are unlikely to disappear. We also discuss the functions needed for persistent archives to benefit from federating with other archives. Because the largest challenges for knowledge persistence arise from sociological difficulties (or 'organizational politics'), we have strong inclinations to emphasize architectures and designs that accommodate local autonomy with strong security. After introducing the context wherein federated archives operate, we move through a discussion of the architectural and design considerations that appear to drive federated archives operating in a persistent environment.

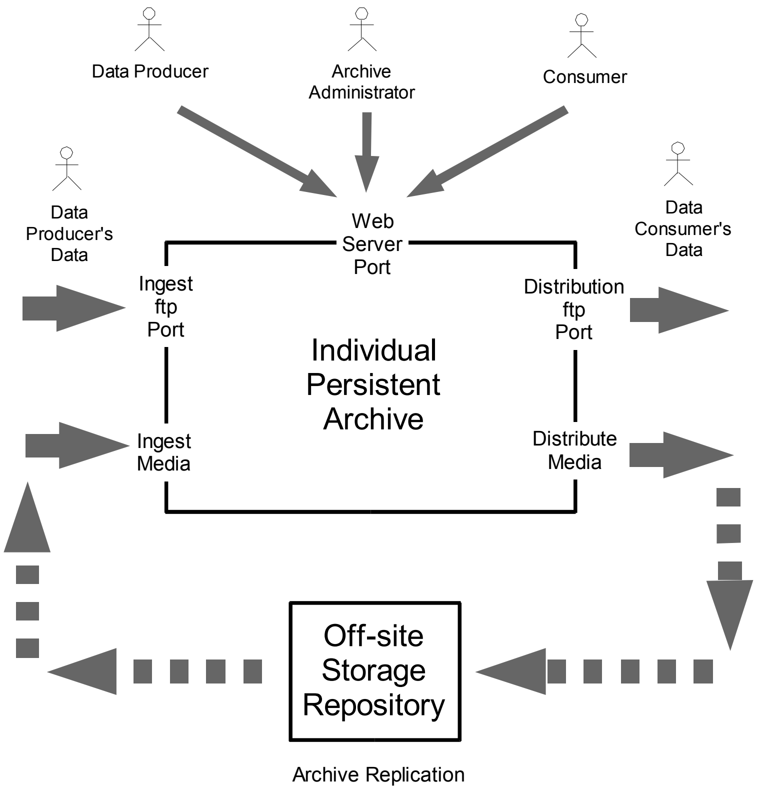


Figure 1. Context Diagram for a Persistent Archive, considered in isolation.

The Context of Federated Archives

If we consider a data center or archive in isolation, we can think of such an organization as having a Web Services port for most users and for archive administrators. However, we separate the ports through which data enters and leaves the system. Figure 1 shows this context.

At the top of this figure, we show separate interactions between Data Producers, Consumers, and Archive Administrators. We include machine-to-machine (or peer-to-peer, P2P) interactions in our discussion in other papers describing the proposed architecture. On the left, we allow data ingest through ftp and media. In accord with the Open Archive Information System (OAIS) Reference Model, we expect that such ingest will be in accord within the context of a Submission Agreement between a Data Producer and the Archive. On the right side of the diagram, we expect a Data Consumer to create an Order Agreement with the Archive. We also indicate large-scale data distribution to data users – in which we include other data archives. Thus, in a federation of data centers or archives, one data center may serve as a data producer with respect to a second.

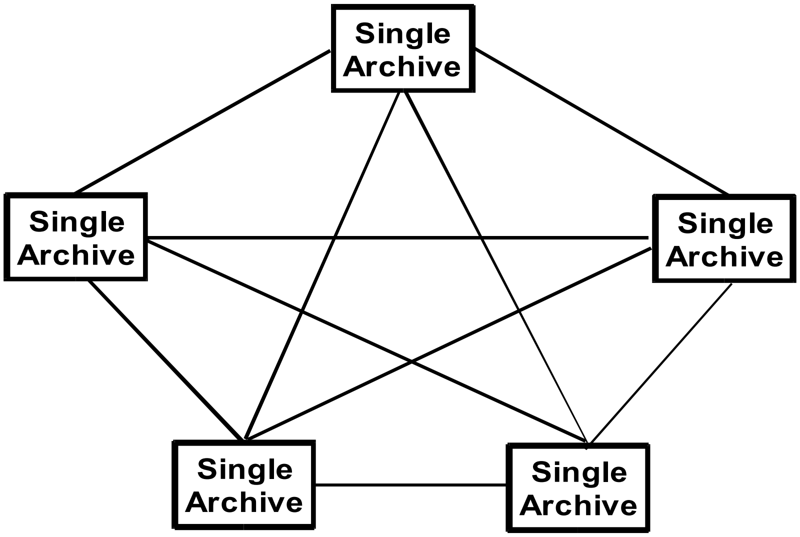
At the bottom of the figure, we show an Off-Site Storage Repository. Normally, we expect such a Repository to be used for backup, i.e., for providing copies of data to the archive to replace data damaged by various happenstances. On an infrequent basis, this Repository may be called on to help the archive recover from a disaster. In the design that we present in detail later, we believe it is critical to create a replicable archive. By this we mean that the archive (and not just its contents) should be put into a form that will allow the entire archive to be reconstructed automatically from the material stored in the Off-Site Repository. As we will see, creating Dissemination Information Packages that mirror Submission Information Packages will allow our design to perform this archive replication function.

In older contexts, an isolated archive might have a stable internal structure and a relatively static collection of material. However, in a modern context, archives that belong to federations can interchange data – and can add metadata and annotations that link together the data within several archives. For example, if one archive contains radar and microwave data that is particularly sensitive to raindrops, while another archive contains observations of reflected sunlight, the two together may develop a catalog of hurricanes. Such a catalog can provide a synergistic expansion in the value of the federated archive, beyond what would be available through the two archives considered separately. Additional synergism can result from providing peer-review and annotations on the collection elements that were not provided by the original data producers.

Figure 2 represents interactions between archives involved in a federation. Because these institutions will be tied together by the Internet and its advanced extensions, we anticipate that information may flow readily amongst any members of the federation. At the same time, we cannot remove the requirements for maintaining local autonomy over the content and connections of an individual archive.

The connections in Figure 2 do not distinguish between web services, ftp, or media. Rather, we simply indicate that the members of a federation are allowed to freely interconnect. The individual members will need to be able to accept or reject possible connections, data flows, and the use of archive services. In addition, the fact that the interconnections may be open to the entire WWW means that the members of a federation will need to build security into the architecture of the systems. Thus, we will have to take into account the fact that connections cannot be trusted to conform to the rules governing access and use of the data and services in the archive. This consideration suggests that the data and metadata in federated, persistent archives will need to encapsulate data, metadata, and provenance information in packages whose authenticity can be verified independently of the properties of the connection context (like the IP address).

In a sense, we anticipate that the single archives in figure 2 will need the equivalent to “guards” over incoming traffic – even if the other federation members are highly trusted. This notion suggests that when an archive receives a request for its resources (whether a request to store a file or database, or a request to perform computations on data within the archive), the archive will want to use information within the incoming data and within the request for services to be able to verify that the request is allowed.



Federated archive interconnections allowed
by either Web Services, ftp, or Media.
Connections allow local autonomy.

Figure 2. Archives linked into a federation.

As we show in figure 3, this security consideration suggests that we want to use a good security configuration. The ***notional*** firewall configuration, following fig. 6-8 in Zwicky, Cooper, and Chapman [2000], suggests that we want a strong firewall configuration with separate "bastion hosts" to handle the traffic on each of the three ports we identified in fig. 1.

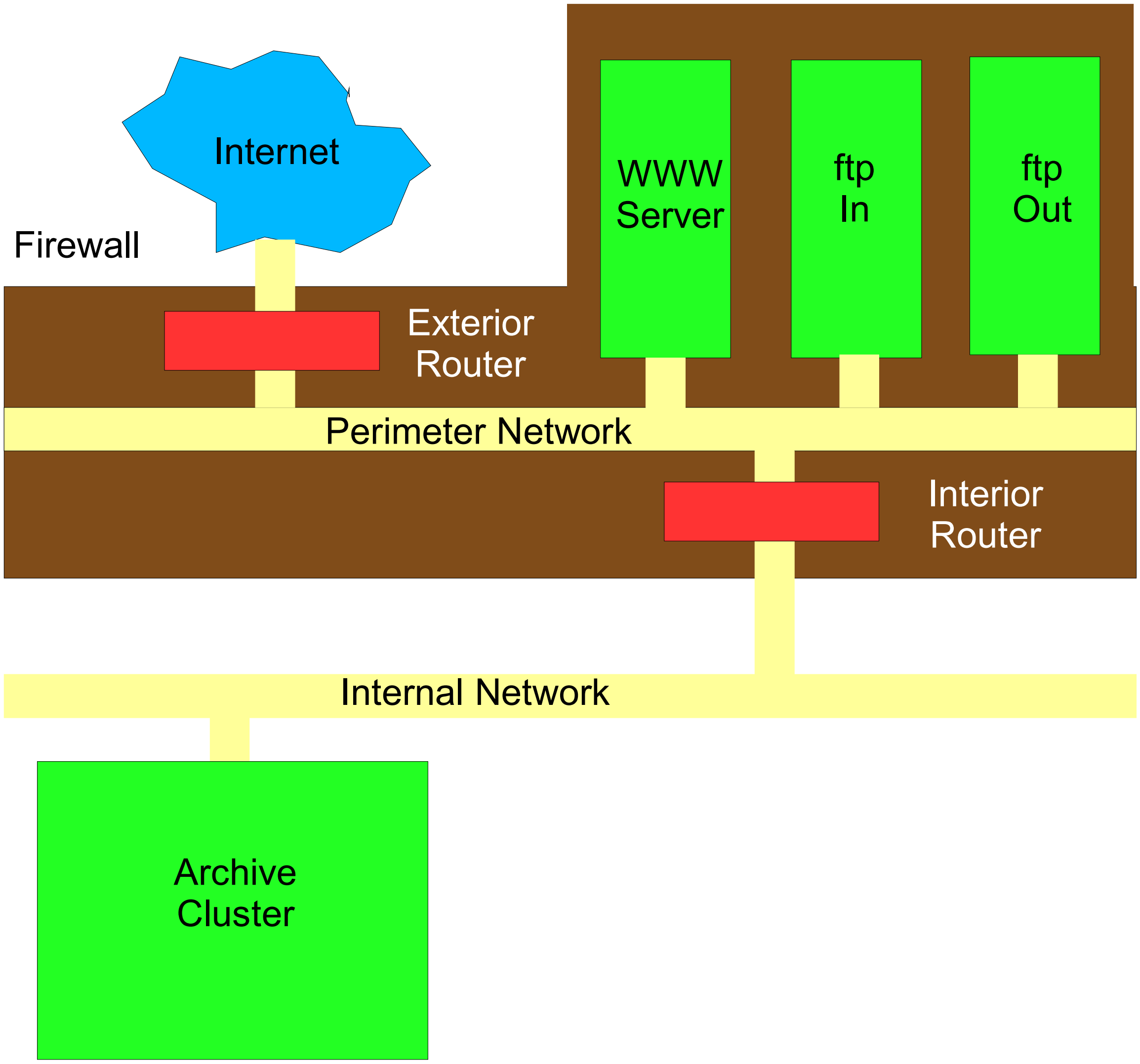


Figure 3. Notional Firewall Configuration for the Shadowfax Persistent Archive Architecture Moderate Security Configuration.

Thus, in this notional firewall configuration, HTTP packets are allowed through the external router into the WWW server and ftp packets are allowed into the ftp In server.

In the outbound direction, packets are allowed out of the WWW server and out of the ftp Out server. All other traffic would be disallowed.

Packets would be allowed from the WWW server and the ftp In server to the Archive Cluster. Outbound packets would be allowed to go from the Archive Cluster to the ftp Out host or to the WWW server.

Figure 4 shows a much higher security configuration with the same internal architecture. In this case, the entire archive is contained within a single physical space with no external network connections. In addition, the ftp input from and to the Internet is replaced by media input and output, with physical control around the media devices. The Web Server interface is also contained within the physically controlled access area – and is used solely by the Archive Administrators.

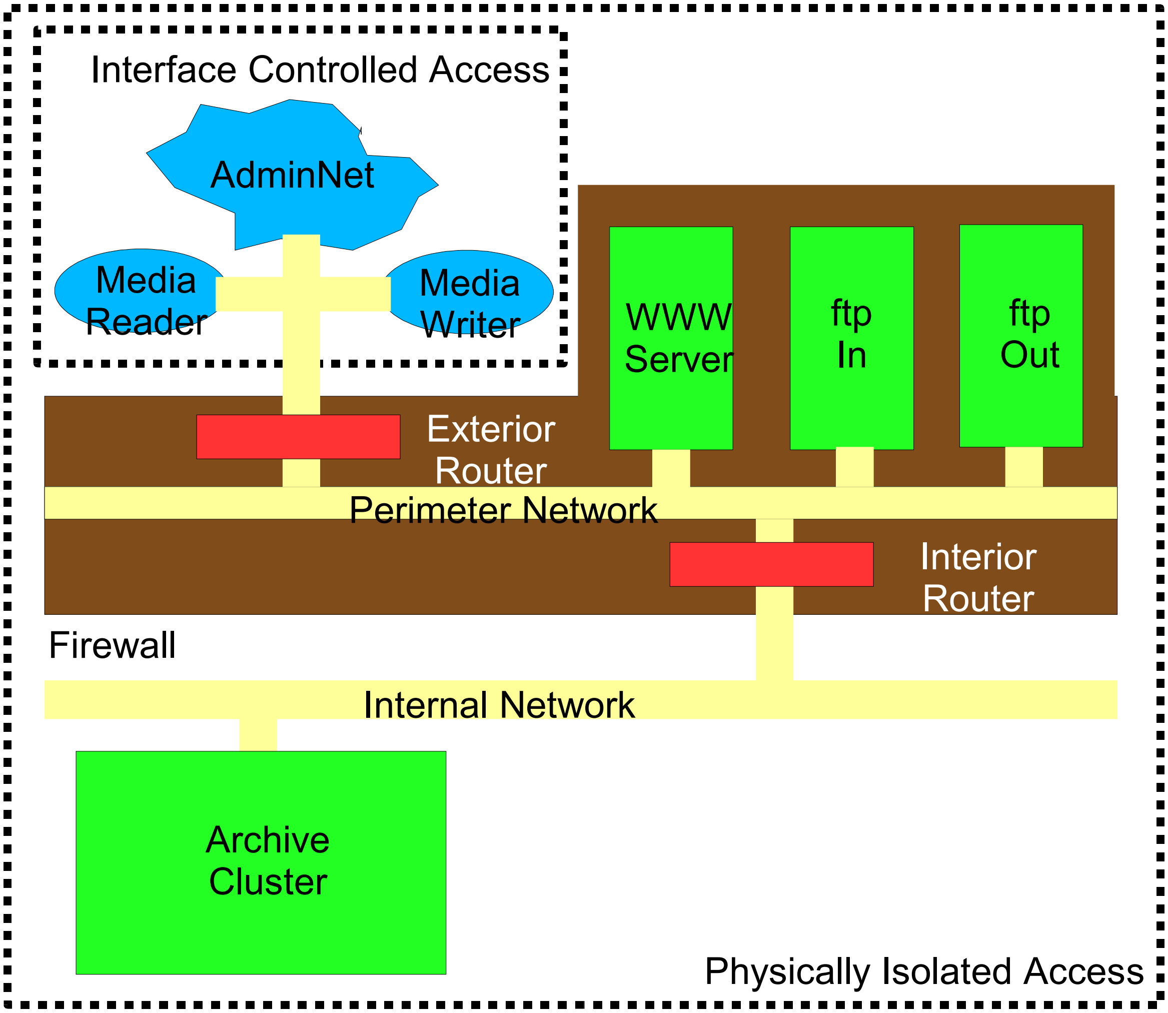


Figure 4. Notional Firewall Configuration for the Shadowfax Persistent Archive Architecture Higher Security Configuration.

From an architectural design standpoint, the issue of whether the archive is connected to the Internet or disconnected from it should make little difference – except for the software that must deal with the direct interface to the external universe. We also note that the internal design will not be affected by whether the internal storage is considered the long-term media by itself, or whether the architecture is simply serving as a large "filter" that checks the incoming data for reliability, perhaps organizes it to facilitate the off-site repository, and passes it on. From the standpoint of the system's logical structure, these are not different models. The throughput rate needs to be designed to accommodate the ingest and distribution rates – however, the logical structure inside the architecture does not appear to need to change to accommodate different throughput rates.

We do note that both the security model and the throughput rate affect the engineering choices of an implementation of the archive structure – and thereby the cost of building and maintaining the archive. For example, if there is a preference for using the archive as a "filter" with the real "long-term archive" being the off-site repository, then there is likely to be an increased staff cost that should be justified by the probable loss created by a realistic assessment of potential threats to the archive and its contents. The same quantitative justification is required for the engineering choices associated with data throughput requirements. As we will see in the documentation describing the Submission Agreement (which is another document), good practice, as well as design engineering strongly suggest that we would be prudent to develop a "Valuation and Risk Assessment" for proposed archive contents, to develop a preliminary quantification of a "User Access Model", and to conduct an "Engineering Analysis" to verify that the expectations of the Data Producer and the Archive are within acceptable and affordable limits.

We will discuss the security considerations in much more detail elsewhere in this design documentation (particularly in the design for the second phase of development identified below). For now, we want to emphasize that the firewall design in fig. 3 is purely notional – and not normative.

On the Contents and Structure of Archives in Persistent Environments

In this paper, we expect archives to store two fundamental kinds of data packages: files and databases.

A file, in this context, is a collection of individual data elements organized into records. The internal file organization may be quite complex. Within a file, the individual data elements may be text, but may equally be numerical values, images, or other kinds of digitized objects. The fundamental operations that an archive can supply for accessing the data in a file are reading and writing.

Again, in the context of this discussion, we regard a database as a collection of data, organized into (interconnected) tables that consist of rows and columns. At present, the Structured Query Language (SQL) provides the fundamental operations that provide access to the data in a database.

It is useful to consider some specific examples of archives in persistent environments. Table 1 provides some specifics for various systems.

Table 1. Specific Characteristics of Current Archives

Data Center	***Dominant Contents* [Text Files, Photographic Images, Binary Data, Other]**	***Number of Files or Database Tables***	***Total Data Volume* [TB]**	***User Access Rate* [User Page Hits/Dy]**	***Volume Distributed Per Year* [GB/Yr]**
ASDC	Binary Data	2,000,000	1,000	5,000	50,000

We will comment in more detail elsewhere on the structure of persistent archives [cf. documentation on the archive Logical Name Space or on the Information Packages, such as the Submission Information Package]. For now, we note that the files (or database tables) in such archives can have very different organizations and metadata structures. For text-based documents, several hundred years of library experience give a solid basis for structuring the archive collection and providing it with metadata. From the point of view of our work here, we can visualize the Library of Congress' classification as providing a shallow hierarchy that organizes the collection of books into a linear order. Each book can be described with a homogeneous set of metadata fields, such as those identified in the METS or Z39.50. User searches on this kind of collection then become equivalent to SQL queries that identify the desired books in the collection as result sets from the query.

However, this view of how collections should be organized and how the metadata should be structured is probably simplistic for collections that fall outside of the boundaries of the classic library experience. Many collections of material have complex structures that need to be reflected in their Logical Directory Namespace. For example, theoretical considerations suggest that the files in the Atmospheric Sciences Data Center have a “layered hierarchy” (Data Producer, Data Product, Data Set, Data Set Version, File), with each layer having distinctive metadata fields [Barkstrom, 2003]. Earthquake engineering archives may have complex collection structures based on the view that the files are most understandable when organized as “reports”. Similar complexity emerges for engineering projects in which the part of the collection associated with blueprints has one kind of hierarchy (devoted to describing the relationship between blueprints in an order that facilitates assembly of a machine), while another part has a very different organization (e.g., the chart of accounts in the financial records of the organization).

In the architectural description we provide below, we feel that it is necessary to provide "containers" into which we can slip a variety of Logical Directory Name Spaces and metadata organizations. By creating a flexible and readily implemented structure for holding this information, we can avoid painful and time-consuming arguments about details and concentrate on making sure the archive can be implemented in a reliable and cost-effective manner.

Views of Persistent Archives

For this paper, we believe it will be helpful to break down the material we present into three kinds of views, corresponding roughly to views we might take of a building as it is constructed. Indeed, we expect to develop prototype systems in the order in which we present these views. We base our views heavily upon the OAIS Reference Model, although we modify our description as this seems necessary. The three views are:

1. The **Architectural View**, which identifies the key data structures and functions that the archive needs for its basic operation – basically, getting data into and out of the archive.
2. The **Infrastructure View**, which shows data structures and functions that have to work properly in order to keep the archive operational – in particular security and reliability.
3. The **Administrative View**, which shows the organizational and managerial structures and operations that keep the archive viable – in particular, human operations and preservation planning.

The Architectural View

For the architectural view, there are four fundamental items from the OAIS RM:

- **Submission Agreement Between the Data Provider and the Archive**
- **The Information Packages that Encapsulate Data and Context**
    - **Submission Information Package (SIP)**
    - **Archive Information Package (AIP)**
    - **Dissemination Information Package (DIP)**
- **The Archive Logical Namespace and Metadata Structure**
- **The Archive, Metadata Database, and Off-Site Repository Structure**

As we move through the architecture and design documentation, we will flesh out each of these items with XML descriptions of data structures and object diagrams for more active objects.

The Infrastructure View

For the infrastructure view, the key items are

- **The Security Model**
- **Message Logging, Transactions, Activity Reconciliation, and Auditing**
- **Exception Handling**

The details of these items will be covered in appropriate subsections of the architecture and design material.

The Administrative View

For the administrative view, the key items are

- **Administration, notably Budgeting, Operations, and Capacity Planning**
- **Operations**
- **Information Management**

The details of these items will be covered in appropriate subsections of the architecture and design material.

References

Barkstrom, B. R., 2003: Data Product Configuration Management and Versioning in Large-Scale Production of Satellite Scientific Data, in B. Westfechtel and A. van der Hoek (Eds.), *SCM 2001/2003*, **LNCS 2649**, pp. 118-133.

Consultative Committee for Space Data Systems, 2001: *Reference Model for an Open Archival Information System (OAIS)*, **CCSDS 650.0-R-2**. CCSDS, Washington, DC.

Zwicky, E. D., S. Cooper, and D. B. Chapman, 2000: *Building Internet Firewalls*, 2nd ed. O'Reilly, Beijing, China, 869 pp.