*DRTC Workshop on
Digital Libraries: Theory and Practice
March, 2003
DRTC, Bangalore*

Paper: B

Digital Library Architecture

Richa Pandey
Documentation and Research Training Centre
Indian Statistical Institute
Bangalore-560 059
email: *richa_p5@yahoo.co.in*

Abstract

Digital library is a very complex system. A digital library can have multi-tier architecture. Different digital libraries follow different architectures and models. This paper discusses the basic concepts and principles involved in its design and architecture. It further presents an evaluative account of various digital library architectures.

1. INTRODUCTION

A digital library is an integrated set of services for capturing, cataloging, storing, searching, protecting, and retrieving information, which provide coherent organization and convenient access to typically large amounts of digital information. Digital libraries are realizations of architecture in a specific hardware, networking, and software situation, which emphasize organization, acquisition, preservation, and utilization of information.

2. CRITERIA FOR DIGITAL LIBRARY

To develop a digital library system, the following criteria can be considered:

1. *Low cost*, including all hardware and software components;
2. *Technically simple* to install and manage;
3. *Robust*
4. *Scalable*
5. *Open and inter-operable*
6. *Modular*
7. *User Friendly*;
8. *Multi-user* (including both searching and maintenance);
9. *Multimedia* digital object enabled; and
10. *Platform independent* (including both client and server components).

3. PRINCIPLES FOR DIGITAL LIBRARY DESIGN

The following principles guide the development of the architecture.

a. Service driven

The architecture for the DLs must be driven by the services it provides and tools required for delivering the service

b. Open architecture

The architecture must be open, extensible and support interoperability among heterogeneous, distributed systems

c. Scalability

The architecture must be robust, scalable and reliable in a high transaction rate production setting thousands of patrons with a wide variety of backgrounds and information needs

d. Preservation

The architecture must ensure persistent access to collection of the DL, addressing such issues as naming, digital archiving and digital preservation.

e. Privacy

The architecture must be sensitive to privacy issues and support both anonymous and customized access to resources

f. Practicality

The architecture should represent a flexible and practical approach to standards, recognizing the need to balance the level of information collection with economic constraints

g. Modularity

The architecture should represent a mix of new technology and legacy pieces, all of which must inter operate while involving at different rates.

h. Time frame

The time frame required to plan for system migrations in the next year as well as planning for a technology generation framework should be approximately 3 to 5 years.

i. Client support

The architecture should support a base line level of services, which can be accessed with common desktop configuration and software. Certain higher level services may require proprietary clients but the support of these clients should be determined by DL tool and services group.

4. COMPONENTS OF DIGITAL LIBRARY

Digital library framework permits many different computer systems to coexist. The key components are shown in the figure below. They run on a variety of computer systems connected by a computer network, such as the Internet. (1)(See figure 1)

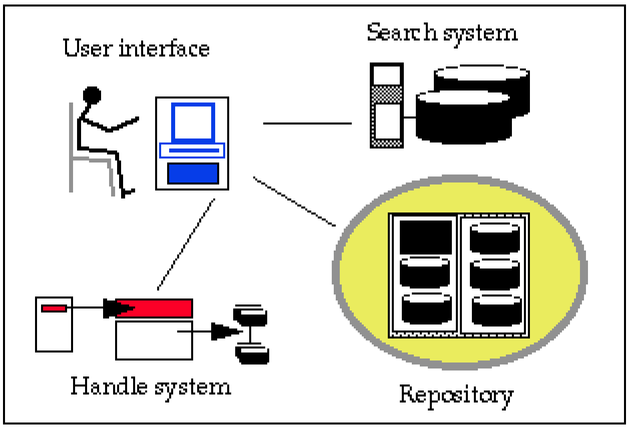

Fig. 1: Major system components

4.1. User Interfaces

We have to use two user interfaces: one for the end-users of the digital library, the other for digital librarians and system administrators who manage the collections. Each user interface is in two parts. A standard Internet **browser** is used for the actual interactions with the user. This can be Netscape Navigator, Microsoft's Internet Explorer. The browser connects to **client services**, which provide intermediary functions between the browser and the other parts of the system. The client services allow the user to decide where to search and what to retrieve; they interpret information structured as digital objects; they negotiate terms and conditions, manage relationships between digital objects, remember the state of the interaction, and convert among the protocols used by the various parts of the system.

4.2. Repository

Repositories store and manage digital objects and other information. A large digital library may have many repositories of various types, including modern repositories, legacy databases, and Web servers. The interface to this repository is called the *repository access protocol (RAP).* Features of RAP are explicit recognition of rights and permissions that need to be satisfied before a client can access a digital object, support for a very general range of dissemination of digital objects, and an open architecture with well defined interfaces**.**

4.3. Handle System

Handle*s* are general-purpose identifiers that can be used to identify Internet resources, such as digital objects, over long periods of time and to manage materials stored in any repository or database. When used with the repository, the handle system receives as input a handle for a digital object and returns the identifier of the repository where the object is stored.

4.4. Search System

The design of the digital library system assumes that there will be many indexes and catalogs that can be searched to discover information before retrieving it from a repository. These indexes may be independently managed and support a wide range of protocols

5. DIGITAL LIBRARY ARCHITECTURE

An architectural approach to the digital library can be discussed under following points. (2)

5.1 Notional Architecture

At notional level, data and metadata and meta-object are considered. Data are library materials in the traditional libraries where as digital library deals with digital information or data and metadata is data about object in the digital library. The traditional card record is an example of metadata for traditional library. A **meta-object** is an object that provides references to a set of digital objects. In its simplest form, a meta-object is a list of handles of other digital objects. For example, a meta-object for an anthology is a digital object that lists all the poems. An important example of a meta-object is a digital object that lists all converted versions of a specific physical item. Digital objects are kept for defining the metadata (Data about data). The designing of metadata is important for searching and retrieval of information. Most of the integrated library automation software takes care of the process of defining metadata. The metadata are entered in fields. This software indexes all the fields according to the requirement of users and the system administrators (See Figure 2)

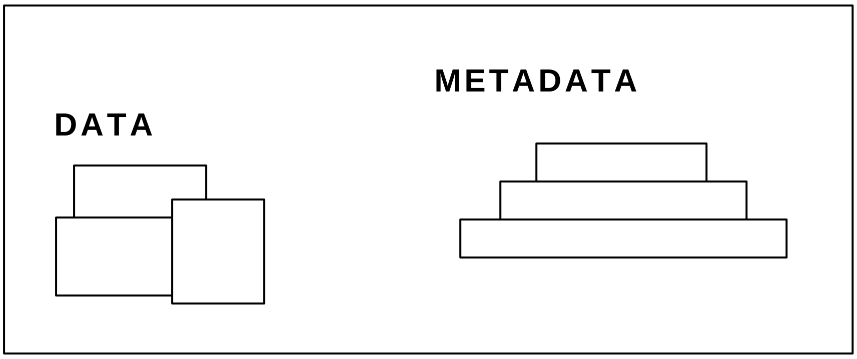

Fig. 2: Notional level

5.2 Operational System

At operational architecture level, it is important how information flow is manage through the system's components. Digital library will be a collection of disparate systems and resources connected through a network, and integrated within one interface, most likely a Web interface or one of its descendants. These resources may reside on different systems and in different databases, they would *appear* as though they were one single system to the users of a particular community. So for both contemporaneous and retrospective search and retrieval of information, the digital library service must provide information interoperability in middleware. And for this some common standards will be needed which will facilitate cross-domain searches and retrieval. (*See Fig. 3*)

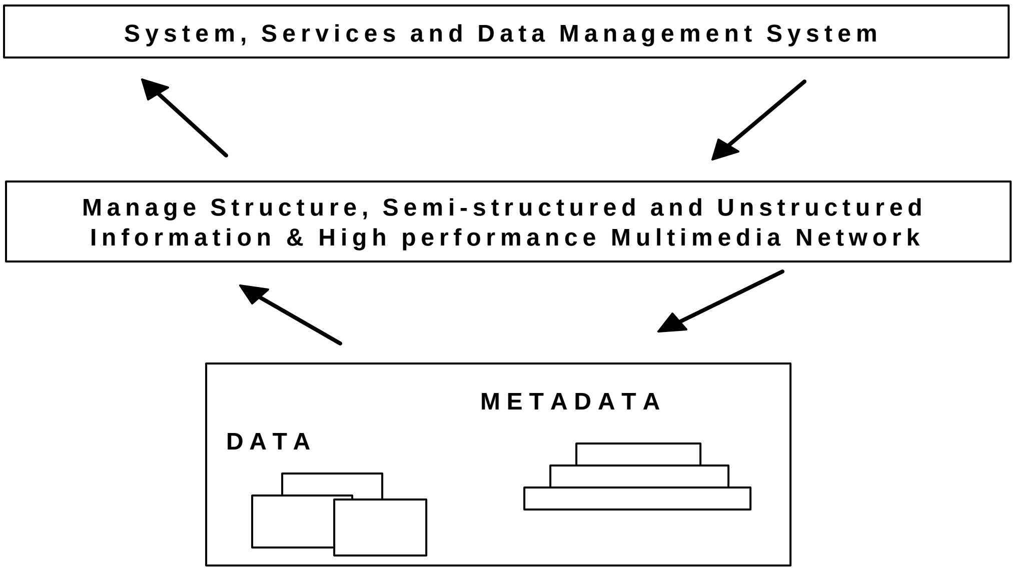
System, Services and Data Management System
Manage Structure, Semi-structured and Unstructured Information & High performance Multimedia Network
METADATA
DATA

Fig. 3: Operational Level

5.3. Technical System

At technical level we have to think about the functional component. The metadata is for content and is added to the digital library. It provides information about the content. So, Metadata and data must be bound together logically, and there must be a robust underlying technology to manage the logical connection through time, across platforms, and over geographical separations, all on a networked, distributed system. It describes major functional areas that taken together provide necessary components to build robust, scalable and interoperable digital library applications and services with the resulting digital objects. (*See Fig. 4*)

Functional Components:

- Hardware (Servers, PCs(Clients),Modems, Storage devices, Book Scanner, CD/DVD Writers and digital camera, Video digitizer, UPS backup etc)
- Software (OCR, Linux/Solaris, MS Windows (Windows NT, Windows 95, Windows 98 etc), ORACLE, publishing Software, Search Engines etc.)
- Digital Resources (CDs, E-journals, Scientific & Technical journals like, IEEE, ACM, ACS etc)
- Conversion of Materials to digital format with proper licensing agreements
- High Speed Internet connectivity to broadband backbone
- Miscellaneous expenditure

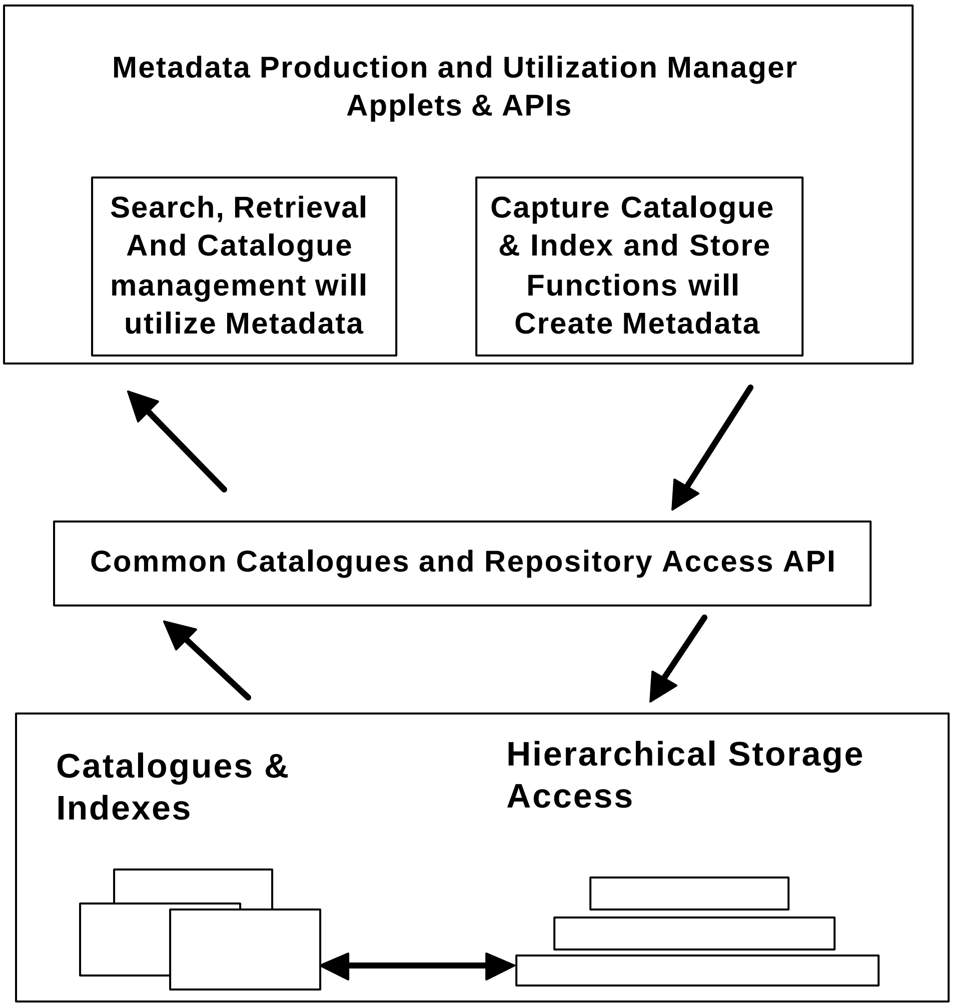
Metadata Production and Utilization Manager
Applets & APIs
Search, Retrieval And Catalogue management will utilize Metadata
Capture Catalogue & Index and Store Functions will Create Metadata
Common Catalogues and Repository Access API
Catalogues & Indexes
Hierarchical Storage Access

Fig. 4: Technical Architecture

5.4. System Architecture

The system architecture is rationalized relative to the operational and technical architecture. It is desirable, to concern, system properties such as scalability and extensibility can be taken into account at the system architecture level. At this level whole digital library system is kept in mind. It can be said that DL is a centralized subsystem that interacts with variety of data producers and customers within a complex distributed system

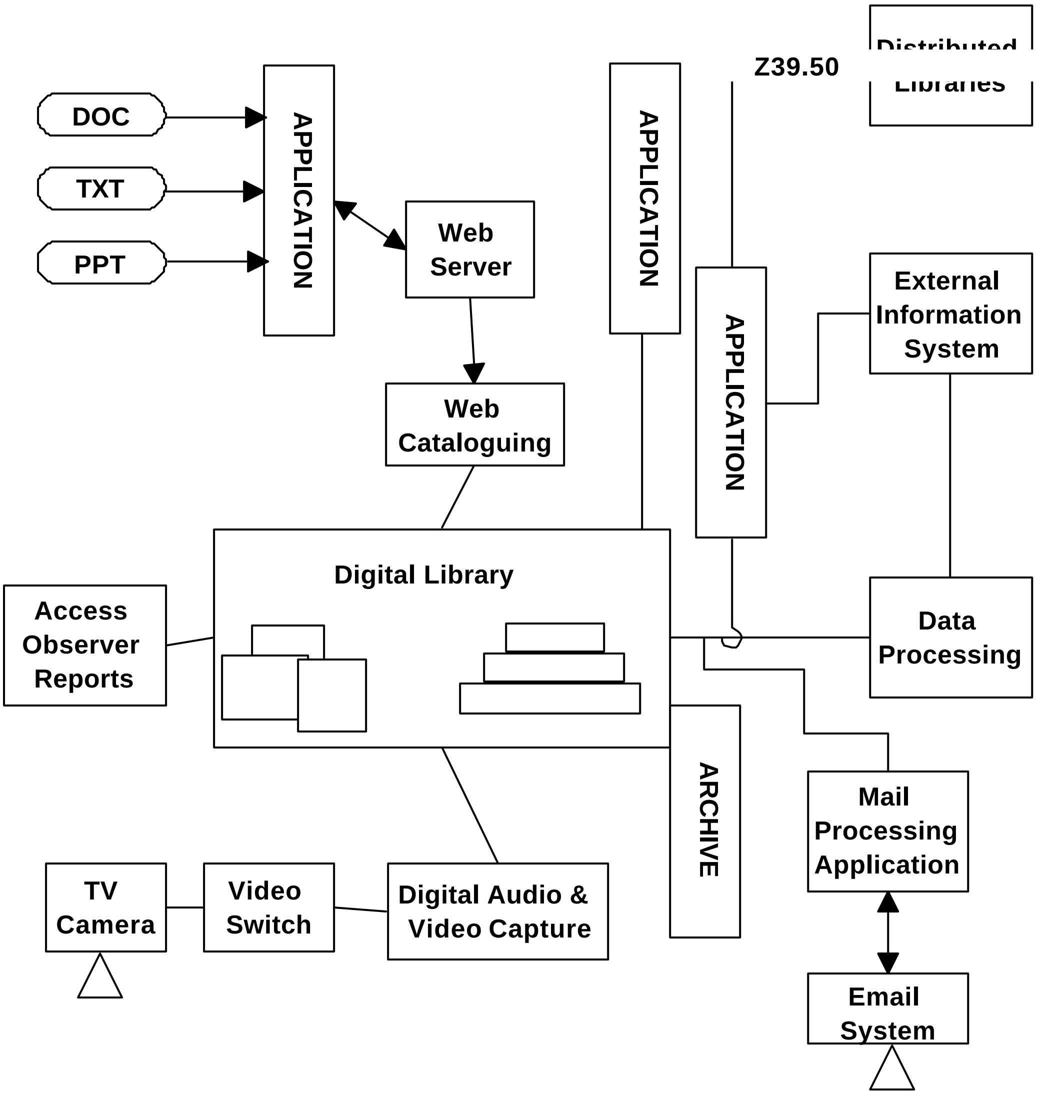

Fig. 6: System Architecture

7. GENERIC DIGITAL LIBRARY MODEL

This basic digital library architecture is derived from a traditional library's four major components:

- The catalog of holdings; the storage area, organized into collections
- The user interface, for access to library services
- The ingest facilities (which a digital library uses to add documents and information about document cataloging and access),
- For storing and processing data from new holdings. (3)(See Figure 7)

Fig. 7: Generic Digital Library Model

8. ARCHITECTURE OF DIFFERENT LIBRARIES

8.1. Architecture Of Alexandria Digital Library (ADL)

Alexandria Digital Library Project. The name *Alexandria* comes from the library of Alexandria, Egypt, which was considered the center of all knowledge/learning. The project began in 1995 with the development of the Alexandria Digital Library, a working digital library with collections of geographically referenced materials and services for accessing those collections. The Alexandria Digital Library Project is headquartered on the campus of the University of California at Santa Barbara. The Davidson Library hosted the Alexandria Digital Library. (4)

ADL follows three tiered structures (See Fig. 8)

- Client
- Middle ware
- Server

*KNF (Kevin's Normal Form)

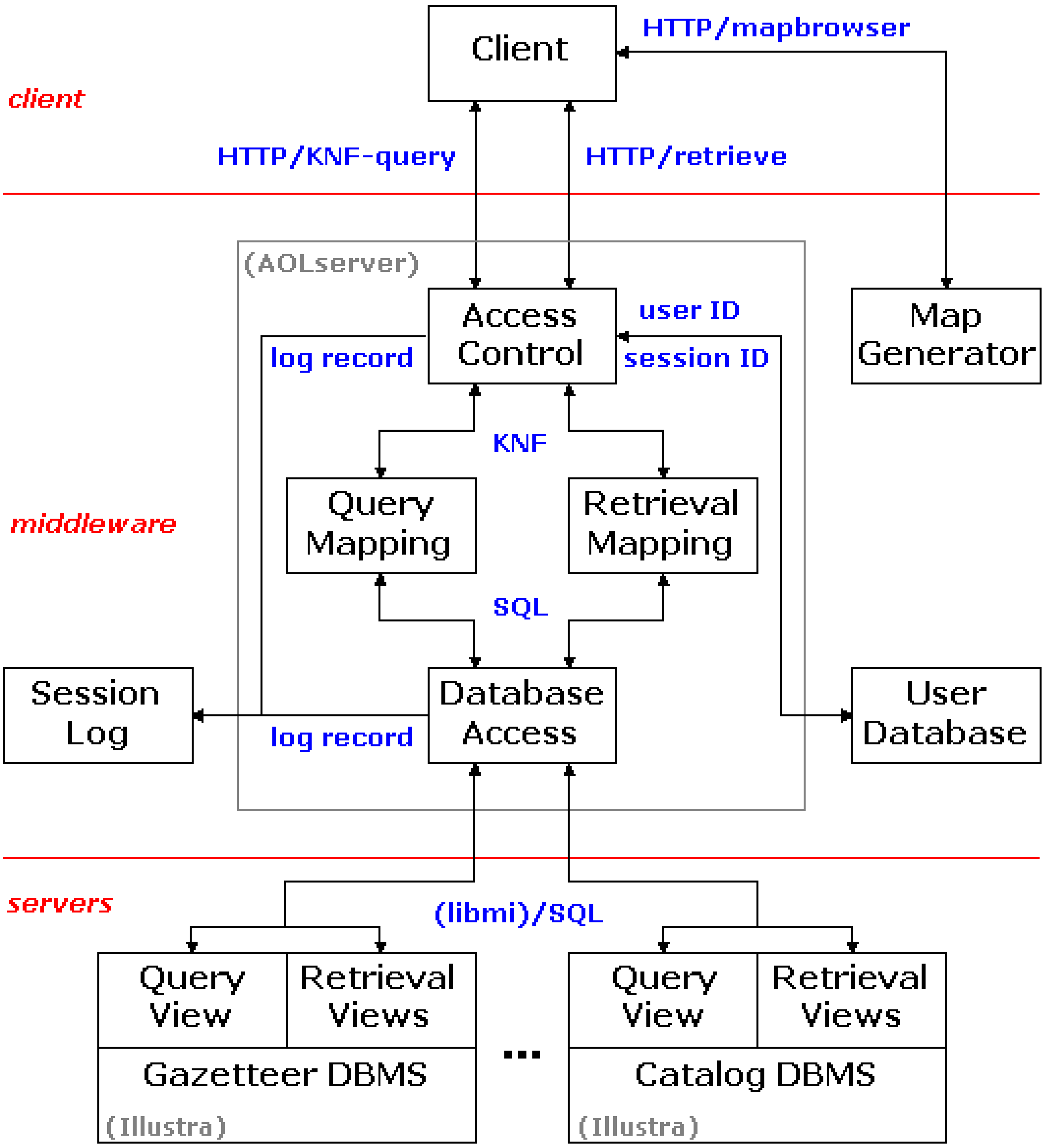

Fig. 8: Architecture of Alexandria Digital Library

8.2. Architecture of California Digital Library (CDL)

The California Digital Library was founded in 1997 by University of California President Richard Atkinson, who called it "a library without walls."

Major components of CDL are as follows:(5) (See Figure 9)

- Client Desktop
- Server Tools
- Object Metadata
- Digital Objects
- Storage
- Integrating Components
- Protocols
- Persistent identifier

Fig. 9: Architecture of California Digital Library (CDL)

8.3. An Agent Architecture for a Virtual Research Digital Library

Agents: (7)

- Helps in bridging the gap between information producer, information consumer and publisher
- Interacting software agents cooperate to provide digital library services

Different Types of Agents (See Figure 10)

- User interface agents (for consumer)
- User interface agents(for authors)
- Publisher agents
- Information retrieval agents

- Broker agents
- Directory agents
- Quantity agents
- Referee agents

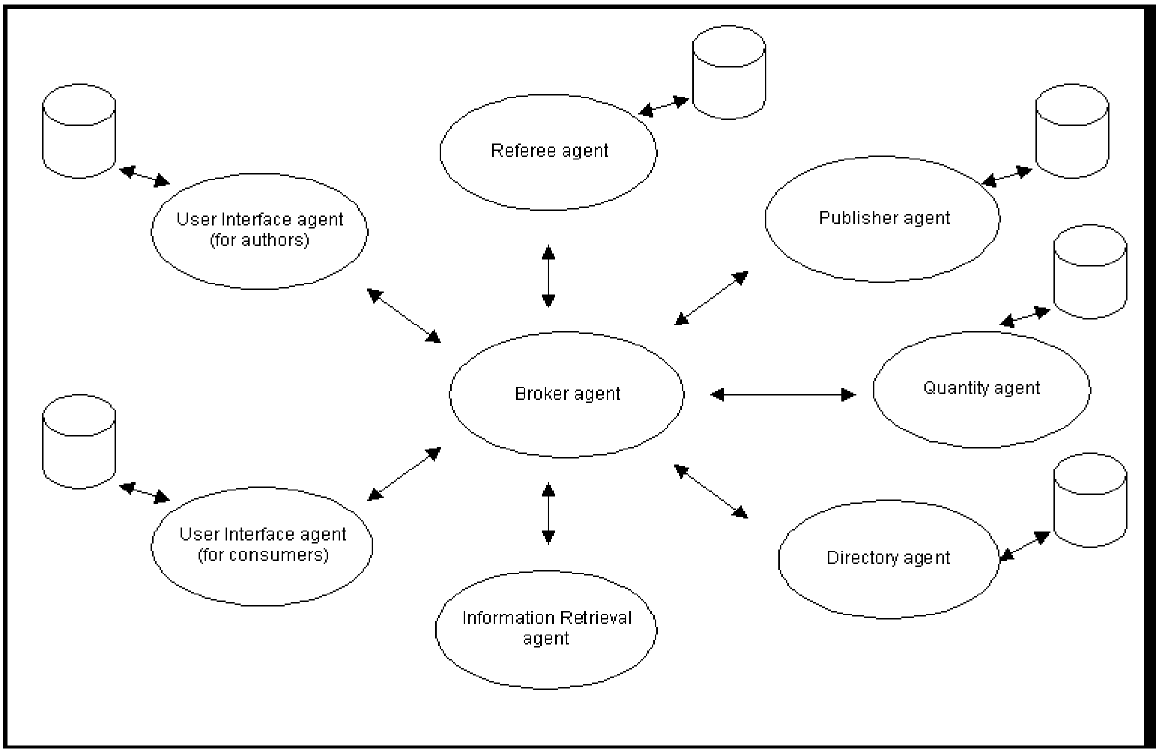
Referee agent
User Interface agent (for authors)
Publisher agent
Broker agent
Quantity agent
User Interface agent (for consumers)
Directory agent
Information Retrieval agent

Fig. 10: Architecture of Virtual Digital Library

8.4. Architecture of NCSTRL (Networked Computer Science Report Library)

NCSTRL is

- largest distributed digital library on the internet
- A collaborative effort by a number of universities
- Focuses on distribution of computer science technical reports
- A network of interoperability digital library server
- Underlying technology as foundation- DIENST

8.4.1. Services offered by NCSTRL

- User interface service
- Repository service
- Index service
- Collection service

8.4.2. Dienst

This is "a system for configuring a set of individual services running on distributed servers to cooperate in providing services of a digital library".

8.4.2.1 Features

- A conceptual architecture for distributed digital libraries
- A protocol for service communication in digital library architecture
- A software system that serves as a reference implementation of the protocol and architecture

- Specifies operational characteristics of core digital services
- Has structured document model
- Open, extensible protocol for communicating with digital library services and accessing these documents
- Provides a mechanism for definition and administration of distributed collection

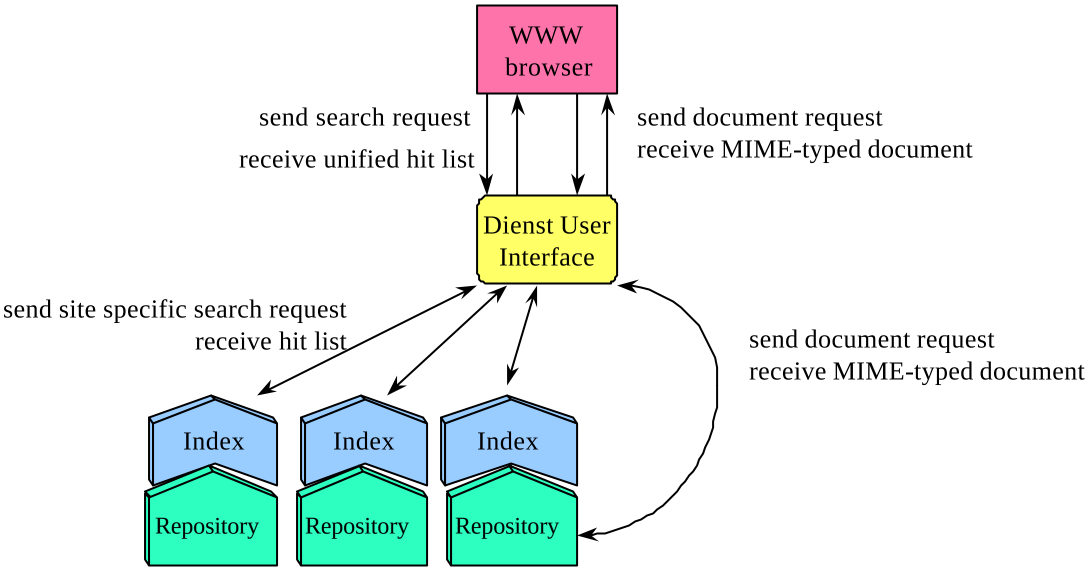

Fig. 11: Interaction of Dienst Services

8.5. Harvest

Harvest is a resource discovery tool, which leverages off of much of the functionality of the Web. Unlike the Web, it attempts to provide mechanisms to facilitate global resource discovery amongst heterogeneous Internet based services like FTP, Gopher, Usenet News, & the WWW. While Harvest does not introduce any new client technologies (WWW Browsers are the default Harvest client) and many components speak existing protocols (HTTP) and generate existing markup (HTML), Harvest does provide an architecture for global search and retrieval across existing network-based, information providers. The architecture of Harvest, which extends WWW, is based up the notion of information Gatherers and Brokers. A Gatherer resides either in an existing repository server (e.g., a WWW server or WAIS server). It interacts remotely via a network protocol (currently HTTP) and collects summaries of metadata for various Brokers to store. This collection is done by the recognizing types of information in the repository and then the Summarizer extracting useful content metadata. Gatherers and Brokers use a specialized protocol Summary Object Interchange Format (SOIF) for efficient communication and storage of metadata. Information from one Broker to another is replicated in a hierarchical manner, this organization being maintained by a Replication Manager. Periodically, the Collector component in the Broker will request updates of metadata from each Gatherer or Broker it services. Any updates are passed to the Registry, which records unique object identifiers and time-to-live values for the data. Identifiers are provided to the IR/Search Engine and the Store Manager is requested to archive a copy of summary metadata in the Metadata DB on the file system for later retrieval by the IR/Search Engine.The Broker responds to queries from clients (currently only WWW clients) by returning the location (URLs) of the items of interest. The query is accepted by the Query Manager, which then translates the request into the Harvest Query Language (HQL) and passes it onto the IR/Search Engine. The IR/Search Engine then provides a response to the query by accessing the Metadata DB according to its own searching strategy. (13) (See Figure 12)

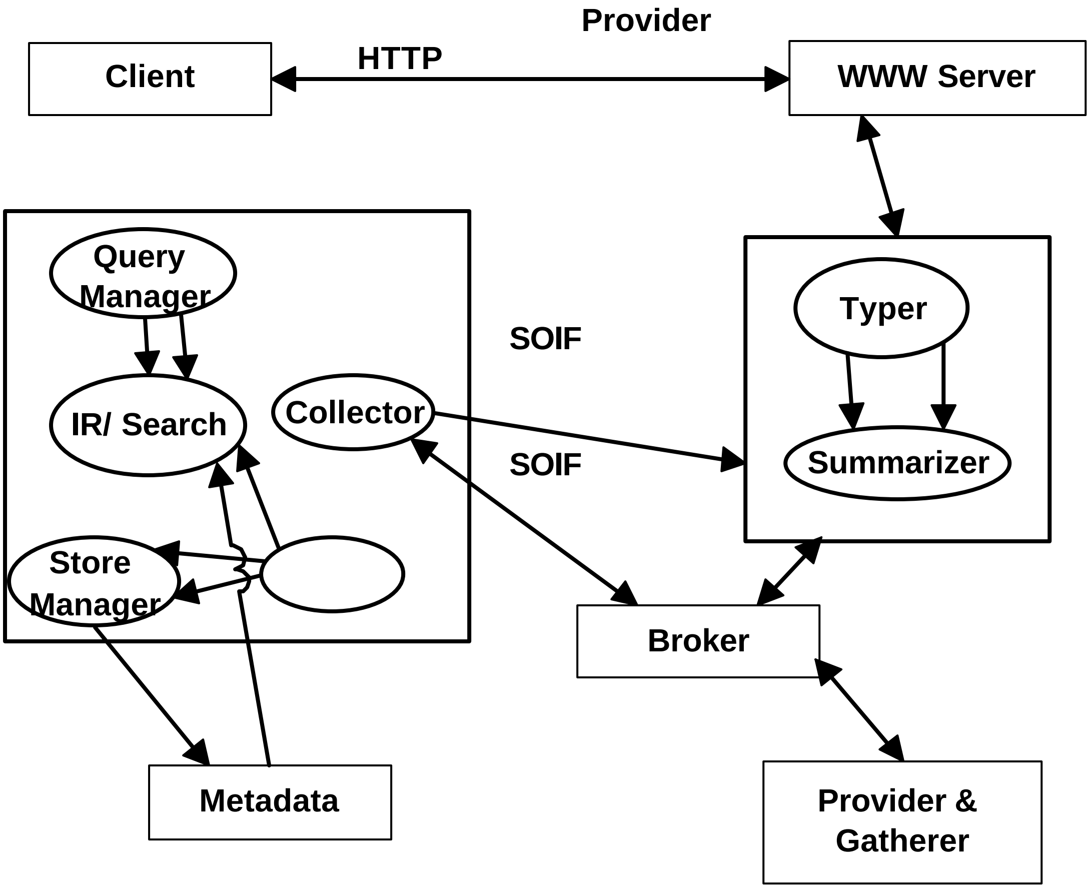

Fig. 12: Architecture of Harvest

8.6. Architecture of Green Stone Digital Library (GSDL) (12)

The GSDL was developed by the University of Waikato in collaboration with UNESCO and the Human Libraries Project to provide information to the developing world. Greenstone digital library retrieves requested information and generates HTML pages that contain the information. The prototype system contains following parts

- PDA(Portable Digital Assistant): A thin client that runs an HTML browser that displays the generated HTML pages. The browser is capable of sending HTTP requests and receiving HTTP responses. This is possible because we make use of the wireless TCP technology that allows HTTP to run over mobile networks without the need to translate to other protocols.
- Customization Tool: The tool can be used by users to customize Greenstone for small screen devices. It generates configuration files that are used by Greenstone to generate HTML pages that match user preferences and that are suited to small screen devices. The customization tool provides a complete abstraction from Greenstones low-level details such as macro or CGI arguments, and allows users to visualize what the generated HTML page would look like, while they customize. The major component of the system is the customization tool. The customization tool allows users to customize all four components of Greenstones interface.(See Figure 13)
- Home Page: Allows users to specify the structure of the home page and gives them the option to place a search for a single collection on the home page.
- Document Structure: The layout or structure of documents within each collection
- Search Preferences: Allows users to customize the search facility (e.g. they can choose to have a normal search or date search)
- Page Style: Allows users to customize the look and feel of every page (e.g. what appears in the header and footer of every page)

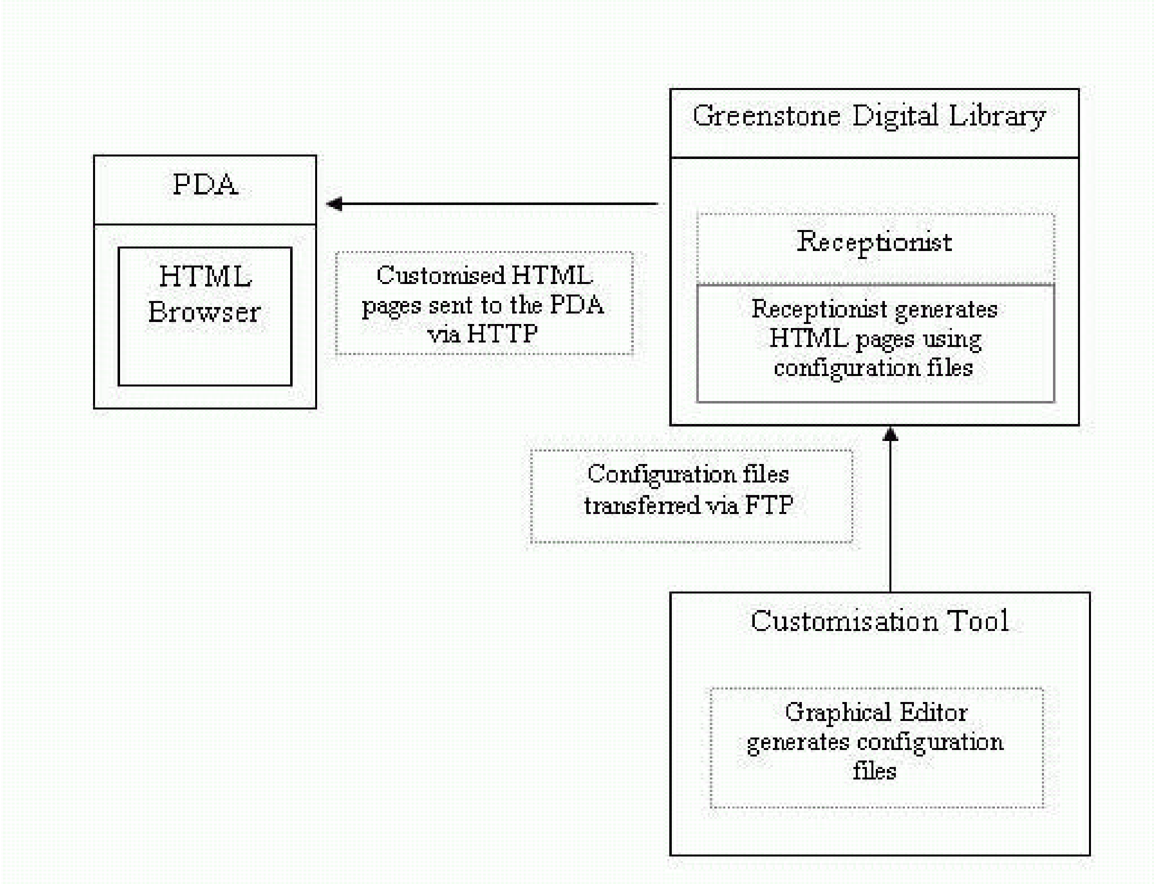
Greenstone Digital Library
PDA
HTML Browser
Customised HTML pages sent to the PDA via HTTP
Receptionist
Receptionist generates HTML pages using configuration files
Configuration files transferred via FTP
Customisation Tool
Graphical Editor generates configuration files

Fig. 13: Architecture of Green Stone Library

8.7. Open Digital Library Architecture

No digital library can claim that it can provide access to all kinds of the information, to cater the different kinds of needs of users by facilitating access among different libraries, In October of 1999 the Open Archives Initiative (OAI) was launched. This attempt to address interoperability issues among the many existing and independent DLs. The focus was on high-level communication among systems and simplicity of protocols. The OAI has since received much media attention in the DL community and, primarily because of the simplicity of its standards, has attracted many early adopters. The OAI Protocol for Metadata Harvesting in essence supports a system of interconnected components, where each component is a DL. Also, since the protocol is simple and is becoming widely accepted, it is far from being a custom solution of a single project. The OAI protocol can be thought of as the glue that binds together components of a larger DL. However, since DLs are themselves defined only loosely, this collaborative system could be composed of individual component DLs, each with different functionality. In the extreme case, each component DL could supply the functionality of exactly one (part of a) service expected by a user. This is the approach taken in this work, where *Digital Libraries are modeled as networks of extended Open Archives, with each extended Open Archive being a source of data and/or a provider of services*. (The "extensions" are necessary since Open Archives are optimized for the provision of data -- but are generalizable to other tasks with a few minor changes.) This network of extended Open Archives, can be called as an *Open Digital Library* (*ODL*). (10) (See Figure 14)

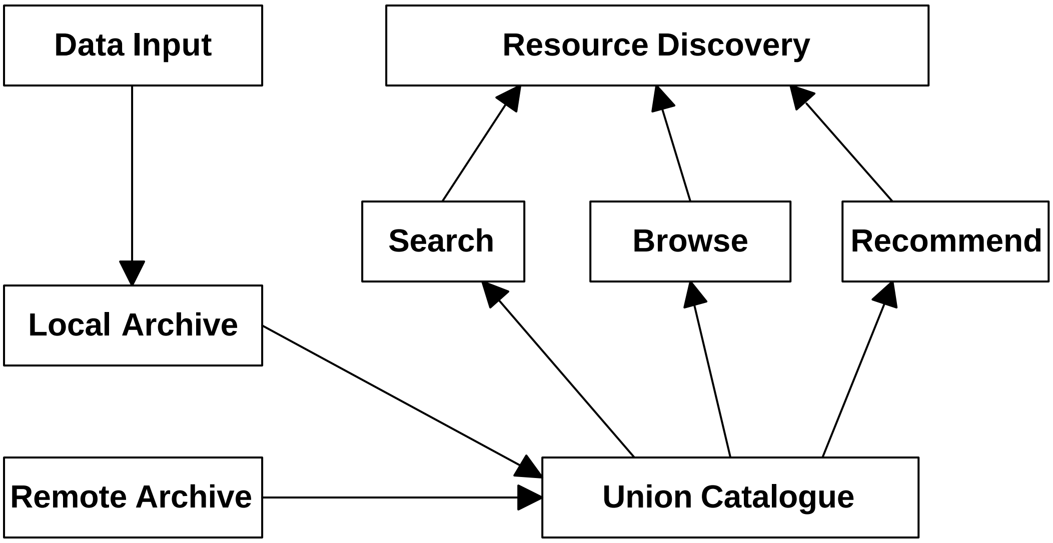

Fig. 14: Network Architecture of an Open Digital Library

9. CONCLUSION

Digital libraries are very important sources of structured well-organized and well-stored information. In fact, it is a collection of disparate systems and resources connected through a network, and integrated within one interface and are responsible for developing and providing access to shared collections. An ideal digital library has distributed structure as the distributed architecture promotes modularity, flexibility, and incremental development, and accommodates diversity in current and future library environments. At the same time, distribution presents difficult problems in interoperability, coordination, search, and resource allocation. So digital libraries must follow architecture guidelines and standards to ensure interoperability between system and consistent level of system performance and coherent searching and access methods.

10. REFERENCES

1. Arms, W. Y., Blanchi, C., & Overly, E. A. (1997). An Architecture for Information in Digital Libraries. *D-Lib Magazine*, *2*(1). from http://www.dlib.org/dlib/february97/cnri/02arms1.html
2. *Chapter 4:Digital Library Architecture*. from http://www.wtec.org/loyola/digilibs/04_03.htm
3. Smith, T. R. (1996). *A Digital Library for Geographically Referenced Materials*, from http://www.computer.org/computer/dli/r50054/r50054.htm
4. *Alexandria Digital Library*. from http://www.alexandria.ucsb.edu/
5. *California Digital Library*. from http://www.cdlib.org/
6. *California Digital Library Technical Architecture and Standard 2002*. from http://www.gseis.ucla.edu/~howard/Classes/208-s00/CDL/CDL-Arch-031000.pdf
7. Isaias, P. T. *A Virtual Research Digital Library: Architecture and Validation*, from http://www9.org/final-posters/65/poster65.html
8. *Networked Computer Science Technical Reference Library*. from http://www.ncstrl.org/
9. *Preservation and Transition of NCSTRL Using an OAI-Based Architecture*. from http://128.82.7.99/ncstrl/p183-anan.doc
10. Sulleman, H., & Fox, E. A. (2001). A Framework for Building Open Digital LibrariesD-Lib Magazine December. *D-Lib Magazine*, *7*(12). from http://www.dlib.org/dlib/december01/suleman/12suleman.html
11. *CDSware: About*. from http://cdsware.cern.ch/index.shtml

12. Patel, D., Ramsamy, N., & Marsden, G. (2002). *Customizing Digital Library for Small Screen Devices*, from http://www.cs.uct.ac.za/courses/CS400W/projects/project2002/libpda/paper/paper.htm
13. *Architecture of harvest*. from http://www.swen.uwaterloo.ca/~dasiewic/courses/ece452/local/slides/saam-Kazman.pdf