A Collaborative Image Database "Out of the Box"

F. Kruggel [1], A. Horsch [2], D.Y. von Cramon [1]

1 Max-Planck-Institute of Cognitive Neuroscience, Inselstraße 22-26, 04103 Leipzig, Germany
2 Department of Medical Statistics and Epidemiology, Ismaninger Straße 22, 81675 Munich, Germany

Abstract

This paper presents the design and implementation of a distributed (image) database on a TCP/IP network of heterogeneous machines. It is based on the tagged container idea: a dataset is treated as a black box and tagged with a description, which is stored in a freely searchable database. Using readily available public domain utilities with a minimum of „glue logic", this is a vendor-neutral, low-cost system. Furthermore, it supports the idea of collaboration by allowing multiple servers running under the policy of different departments.

1: Introduction

The ongoing specialization in clinical neuroscience makes shared image databases a necessity. Diagnosis and treatment of neurological diseases depend on the availability of results from various imaging modalities shared by neurologists, radiologists, neurosurgeons and radiation therapists. However, the implementation of a shared database in a clinical setting has to face a number of constraints. It should (a) be available on a heterogeneous network of computers, (b) leave the administration policy to the contributing party, (c) be scaleable at any time to allow participation of other departments (or even other institutions), and (d) be available at „no cost".

Collaboration has proven to be a successful concept when building scientific information sources: an example is the world wide web on the Internet where an enormous amount of information servers can be contacted by a common protocol. A more restricted example is given the by a joint project of university libraries in Germany. These libraries plan to make their catalogs available for searches over the Internet. A user looking for a specific book sends a request form to a local server which propagates this query to a number of connected servers on the network. Each server processes this query and sends results (a list of books available for this topic or author) back to the local server which compiles a report for the user. This information system consists of a loosely coupled network of multiple servers and clients. The common ground for collaboration is an agreement about the data maintenance policy and the data exchange protocol.

We translate this idea into the context of experimental databases in neuroscience. Treat a dataset (images, recordings, etc.) like a book and place it into a black box, a container. Add a description of the contents, a tag, to it and make this tag queryable in a database. Collaboration in this context refers to: leaving the datasets in the department where they were generated, placing the administration, access and security policy under the control of this source, and leaving the task of data retrieval, conversion and data analysis to the user at the client site. This concept also introduces a strict separation between storage and presentation of the data - it is nothing more than a data delivery system.

We would like to describe an image database, called Documentation System 2 (DS2), that has been implemented on these principles using readily available building blocks. It took four weeks to implement and readily served the needs of our neuroscience research unit.

2: Building Blocks

The Documentation System 2 is based on several widely available building blocks. We would like to introduce them briefly and give references for further information. The World Wide Web (WWW) is a distributed graphical information service on the Internet. This information is delivered in a tagging language called Hypertext Markup Language (HTML) which allows documents to contain text, images, sound files and links to other documents [1]. The WWW is a client/server environment. The control lies within the graphical WWW browser that is used to display the documents. The browser uses the document's reference, retrieves it from a WWW server, interprets the HTML, and presents the document to the user.

0-8186-7560-8/96 $5.00 © 1996 IEEE
Proceedings of IMAC '95

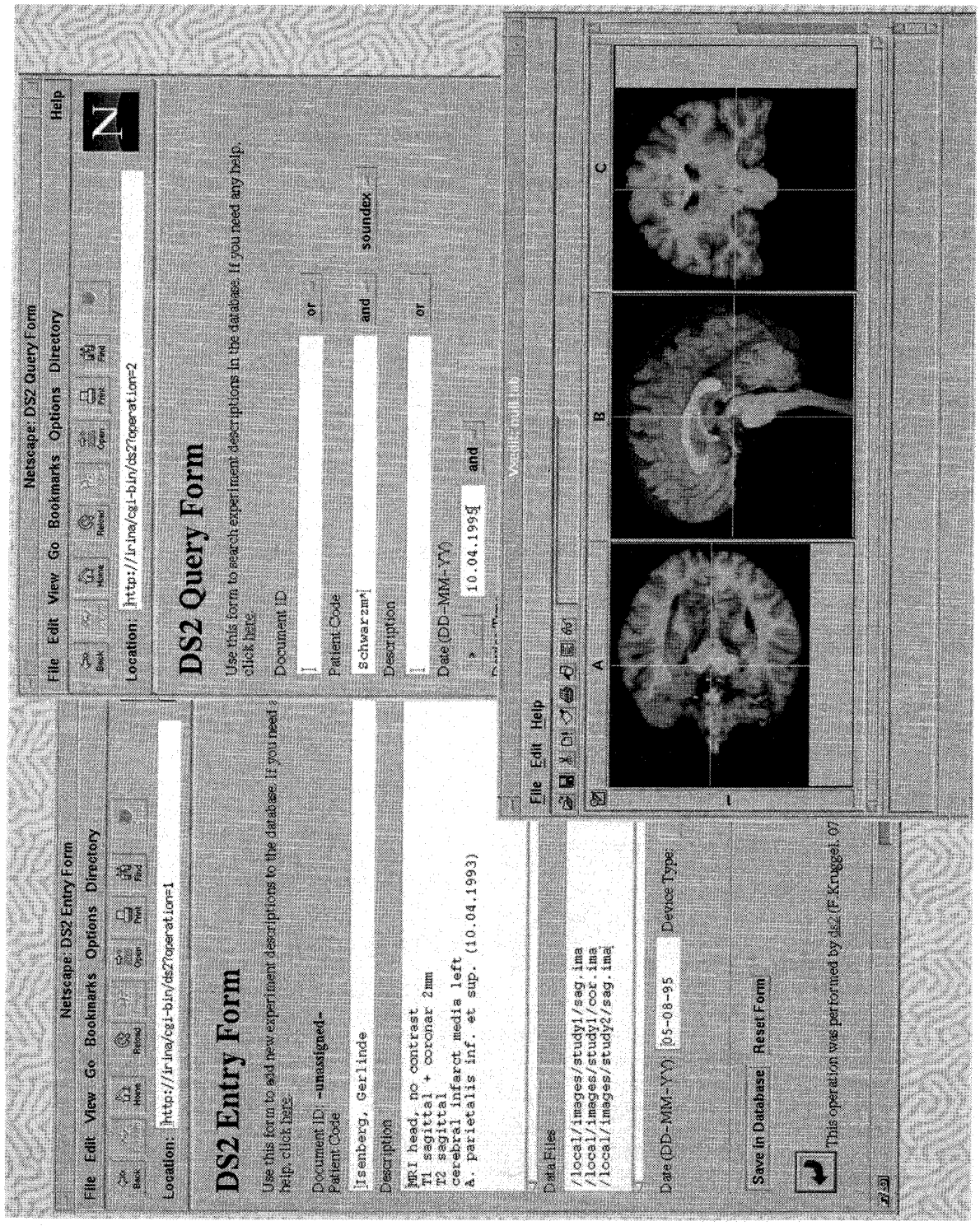

Figure 1: DS2 example screen with a data entry form on the left, a query form on the right and a popup window (bottom right) containing an image processing utility

When a user selects a hypertext link, the process starts all over again - the browser extracts the link, requests the document and displays it. Hypertext links may also cause the browser to spawn a specific viewer for an underlying document, i.e. a postscript viewer or an image processing program. WWW browsers and servers communicate using the Hypertext Transfer Protocol (HTTP) [2].

WWW servers can execute special scripts that enable them to serve as gateways to other information systems. These gateways translate other information sources (such as databases) into HTML documents and thus make them available on the WWW. A HTML document may contain input fields to collect data from the user. The WWW server passes this input to the gateway, which processes it and returns a HTML document to display with the browser. The communication mechanism between the server and the gateway is defined as the Common Gateway Interface (CGI), a term which is also often used for the gateway program itself.

Wide Area Information Server (WAIS) denotes an architecture for a distributed information retrieval system [3]. Structurally, WAIS is divided into three components:

- an indexer, used to create full-text indices of files fed to it,
- a server that can use those indices to search for keywords, and
- a client that build queries for the server in the appropriate format and displays search results to the user.

On the client side, queries are formulated as English language questions. The client application translates this query into a question description structure, and transmits it over a network to a server. The server receives the transmission, translates the received packet into searches for multiple words, conducts the searches involved in a boolean expression and decides which documents match the boolean criteria. The answer to a question is usually not a single document, but rather an ordered list of candidate documents. Those documents that share the most words with the question will come back at the top of the list (have the best score). The list is encoded in the protocol, and transmitted back to the client. The client decodes the response, and displays the results. The documents can then be retrieved from the server. Client and server communicate in the WAIS protocol, which is extension to the standard protocol Z39.50 [4].

FreeWais-SF [5] is an extension of the freeWAIS software provided by the Clearinghouse for Networked Information Discovery and Retrieval (CNIDR). The SF suffix in the software name stands for "structured fields," an indexing and search feature which includes major extensions to the original system:

- Introduction of text, date, and numeric field structures within a document, which allows a document to be indexed using potentially overlapping fields.
- Support for complex Boolean searches (using "and", "or", "not" phrases).
- Stemming and phonetic coding may be switched on and off for each individual field.
- Definition of document format and layout of the headlines are configurable by a specification language based on regular expressions.

There are several alternatives for WWW browsers to function as clients for WAIS databases. Mosaic, a widely available browser [6], includes an option to access WAIS databases via the Z39.50 protocol. However, a custom CGI program allows a more intimate control over how information is processed and thus is preferable for this project [7]. As a starting point, we have chosen WWWWAIS [8], which is a small ANSI C program that acts as gateway between WAIS indexed catalogs and a forms-capable World-Wide Web browser. It creates searchable databases of the information on a server site, allows users to search multiple databases via their WWW browser with customizable options, creates a custom pop-up menu of servers to search through and produces hypertext search results, with file information and links directly to the relevant HTML documents.

Image databases usually contain large files and tend to quickly fill up primary (hard disk) storage. Thus, a mechanism is needed to swap parts of the database out to secondary media (tapes or optical disks). Ideally, this mechanism should transfer files transparently between media by giving a file reference and a media identifier. We have selected Legato's Networker backup software [9] for this task, which performs all the necessary jukebox handling and file searching with a single command.

All these building blocks - except Networker - are available as source code from the specified sites.

3: Data Structure

The DS2 System was designed for storage and retrieval of big datasets. It is based on the tagged container idea. An actual dataset (f.ex. gained in an experiment) is put into a container which is tagged with a description. This description is stored in a WAIS database and can be retrieved by a query. The container is treated as a black box: the system does not know how to convert or present the information inside. This is left to any front end system of the database user. However, the tag serves as a flexible means for retrieving the containers and thus organizing the experiments. This tag consists of a minimum number of required fields, which are shown in Figure 2.

Document ID	A unique ID generated by the database and used for later reference.
Patient Name	A name or code for a patient examined in an experiment. When querying the database, available options include full word searches, partial searches ("Schmi*" looks for "Schmidt", "Schmit", ...) and phonetic searches (similar sounding words like "Mayer", "Meier", ...).
Description	This field is used to enter the description of the experiment and may contain a report, diagnoses or information about the application of contrast agents. Text of any length may be entered here, any word (or any combination of words) will be searchable by a query.
Data Files	Holds the reference to the datasets related to an experiment (i.e. the containers). On entry, this is a reference within the source filesystem. After inclusion in the database, this filename is replaced by a reference within the directory structure in the database. When a group of files is swapped out to secondary storage, it is replaced with a (tape) media I.D. and a filename.
Date	The date of this study in a format DD-MM-YY.
Device Type	The device type field gives an idea where the dataset is coming from. Currently allocated devices are: EEG (Electroencephalogram), MEG (Magnetoencephalogram), MRI (Magnetic Resonance Image) and Unspecified (Any other device).

Figure 2: Fields in the tag of a DS2 record.

The container includes a set of files, whose content is identified by their filename extension. WWW browsers allow the connection of specific viewers on the base of this extension (so-called multi-media extension (MIME) types). This mechanism provides a direct connection from a container to a specific data processing package (like AVS, Analyze).

The container file space is organized in a very simple manner: one directory is created per day for the data gained on that day.

4: Process Structure

Four HTML pages were designed as a front-end for user authentication, data entry, data retrieval, and user help. A small CGI program (Figure 3) sits between the WWW browser and the WAIS database. Its purpose is to (a) collect new data for inclusion within the database, (b) translate the user queries into WAIS queries, (c) generate answers in HTML format, (d) handle file transfers from and to the database, and (e) organize the data file space. An independent backup process watches the file space and swaps datasets to a secondary storage according to a retention policy.

Data entry. When a user has entered a new experiment description in the entry form (Figure 1, left), the tag is built by collecting the individual fields and generating a unique experiment I.D.. The connected datasets are transferred from the local filesystem to the server and renamed with a unique reference. These references are put into the tag, which is sent to the WAIS database. Finally, the tag is formatted as a HTML document and sent back to the WWW browser for review.

Data query. A query form (Figure 1, right) is provided to search experiment descriptions in the database. Only specified fields take place in a query. Search terms can be grouped locally (i.e. within any field) using boolean operations like:

media OR anterior
media AND anterior
(media AND anterior) OR posterior

as well as between fields using the selector on the right. The content of the query fields is translated into a WAIS query. The CGI program contains a list of other known WAIS databases to which a query is sent in parallel. The result of the query from the different servers is collected, translated into a HTML document and sent back to the WWW browser. If more than a single experiment matches the query, a list of experiment descriptions is compiled in a short form, from which the user may select an entry. This description is retrieved from the WAIS database in full detail and displayed in a result document. A click on the buttons provided for the connected containers spawns a viewer for the dataset inside the container, which is an image or signal processing package (Figure 1, in front).

Maintenance. Older directories are swapped out to secondary storage by an independent backup process. When a directory reaches the retention limit, this process queries the WAIS database and modifies the file references. Conversely, when a user requests a swapped-out file, the database process asks the backup server to retrieve the file from the secondary storage. This backup process also saves the WAIS database itself.

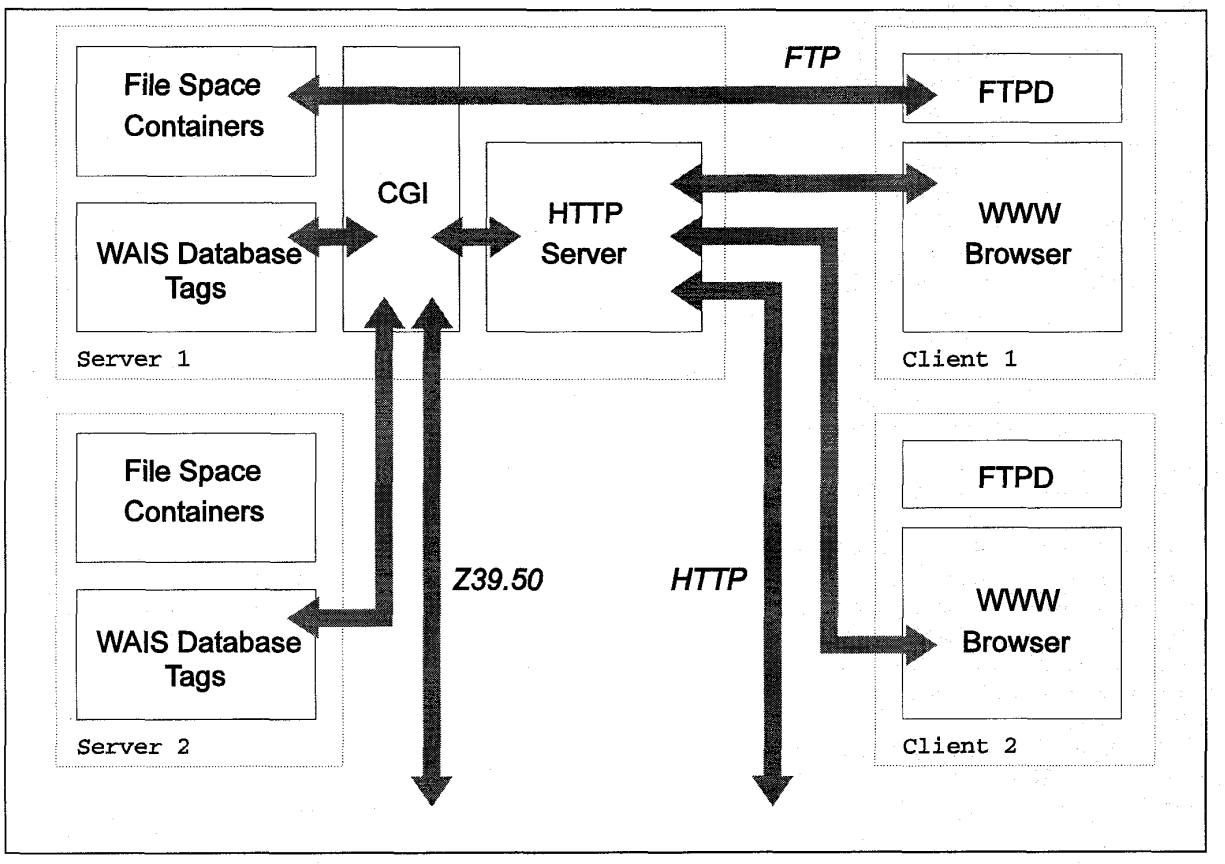

Figure 3: Process structure of DS2: browsers at clients talk to a http server via the hypertext protocol (*HTTP*), which sends the query to the CGI program for processing. This daemon contacts one or more WAIS databases via *Z39.50* protocol to find tags matching the query. The CGI daemon collects the answers, compiles them as a HTML document and sends it back to the browser. File transfers from containers are handled by *FTP*, which is mediated by the CGI process.

5: Implementation

The WWW browser and server and the WAIS database were installed and used as provided by the different sources, so the implementation of our database concept focused on the construction of the CGI process. Starting with WWWWAIS as a model this was a straight-forward task. We have rewritten this model to work as a simple state machine, which is set by a command line argument:

DS2_START	display start form
DS2_ENTRY	display entry form
DS2_ADD	add entry to database, return modified dataset
DS2_QUERY	display query form
DS2_SEARCH	find matching documents and display them in a list
DS2_RETRIEVE	get a specific experiment
DS2_TRANSFER	transfer a file to a client.

The only tricky part in the implementation was the file transfer mechanism. Current WWW browsers inherently allow downloading files, but do not provide any uploading mechanisms, which are nonetheless mentioned in the HTTP specification. We finally decided to incorporate transfers via the file transfer protocol (FTP) [10]. The daemon for handling FTP requests is called ftpd and common to any Unix system. For PCs, we use the shareware program wftpd [11], which requires MS Windows and a socket implementation at the client site. Since these are also requirements for using a graphical WWW browser, they cause no additional problems. However, the inclusion of a file upload mechanism in a WWW browser would be preferable.

Another design issue concerned database rebuilds. The authors of WAIS still report index problems when adding new datasets to an existing database and recommend a rebuild whenever possible [5]. We choose to add new ex-

periments during the daytime and provide a rebuild once a day just before the backup process starts.

We currently do not provide any user-level mechanism for removing experiments from the database. Because we consider this operation as potentially dangerous we accept to have some unnecessary files in our database.

This image database took only four weeks to implement. The CGI program consists of about 2000 lines of C code and has been ported to a range of common Unix workstations. The source is available upon request.

6: Case Study

This image database has been designed as a data delivery service for the research facilities in our institute (Figure 4). We currently have access to the following data sources:

- a 3 Tesla MRI system (Bruker)
- a 128 Channel magnetometer (BTI)
- a 128 Channel EEG system (Neuroscan)
- a 64 Channel EEG system (Neuroscan)

These sources together deliver between 15 and 30 datasets per day with a total volume between 500 MB and 2 GB. These sources send their data via a FDDI network to a DEC Alpha 400 workstation equipped with a Storage Works RAID Level 5 array of 40 GB formatted capacity. Transfer rates are typically between 5-6 MB/s, so transfer times vary between 3-100 seconds. Our retention policy on primary storage is four weeks. Datasets are then swapped out to a DLT jukebox with a compressed capacity of 7x40 GB. We planned a one year retention policy for the secondary storage. A second jukebox is the target for an independent backup scheme for all files on the server. Backup times are in the order of 10-40 minutes per day. Retrieval of a file from tape takes between 2 and 5 minutes. Connected to this server are 12 UNIX workstations (DEC, SGI and SUN) and 35 PCs as clients. We use Netscape on the PCs and Mosaic on the UNIX stations as WWW browsers. The database was set up four months ago and has been running ever since without major problems.

The investment in this implementation was the server at a price of $80k. With an expected lifetime of 5 years this translates to $40 per day. The storage material contributes $2 per day.

7: Future Enhancements

From our day-to-day experience with the database a few enhancements appear desirable:

Switching between related experiments makes a linking mechanism necessary. This could be implemented by embedded HTML links in the tag.

Addressablity by image content is a desirable database feature. Image features (i.e. the size, extent and texture of a brain lesion) may be represented as symbolic (textural) knowledge in a section of the tag and thus be available for

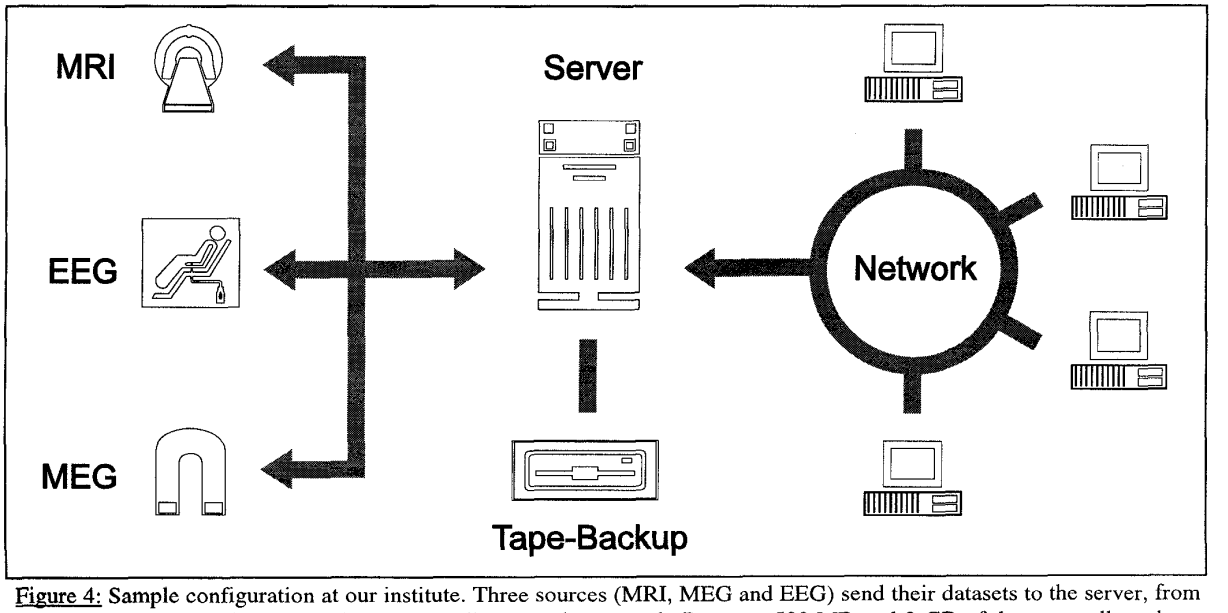

Figure 4: Sample configuration at our institute. Three sources (MRI, MEG and EEG) send their datasets to the server, from which they can be retrieved by PC and Unix clients on the network. Between 500 MB and 2 GB of data are collected per day. For long-term storage, datasets are transferred to digital linear tapes in a jukebox.

queries.

Because the connection between the data representation layer (the browser) and the data processing layer (the daemons) is plain HTML, the design of other interfaces is simple. We have already written a small utility that scans the DICOM files on an optical disk of our MRI scanner, extracts the patient information, packs the slices into a volume format and sends it to the database - without any user interaction. On the other hand we are developing more complex image processing packages, from which results may be fed back into the database - including informations about the processing and the results.

Having patient data on-line raises security issues. Since we currently use our data in a closed in-house network (as most clinical networks are), we do not currently address this problem. In an open network, we need to take care of (a) the access to the data, (b) the integrity and (c) the privacy of the data transferred.

We are currently testing the use of xinetd [7] as a replacement for the Internet super-daemon inetd, which allows us to restrict access to a known group of machines and users. Security mechanisms for the WWW (f.ex. pretty good privacy, PGP) are under discussion. Any method is expected to be transparent within this framework. However, the data transfer between WAIS servers remains a problem. Even the third revision of Z39.50 makes no remarks about transparent security mechanisms in this context. With respect to this application we could modify our WAIS servers and clients to exchange encrypted messages and so maintain data privacy.

References

1 David Raggett (1995) Hypertext markup language specification. Available via http://www.hpl.hp.co.uk/people/dsr/html/CoverPage.html

2 Berners-Lee T (1993) Hypertext transfer protocol (HTTP). Available via ftp://info.cern.ch/hypertext/WWW/Protocols/HTTP/HTTP2.html

3 Lincoln B (1991) WAIS Bibliography. Available via ftp://quake.think.com/pub/wais/wais-discussion/bibliography.txt

4 National Information Standards Organization (1988) Z39.50-1988: information retrieval service definition and protocol specification for library applications. National Information Standards Organization (Z39), P.O. Box 1056, Bethesda, MD 20817. (301) 975-2814.

5 Pfeifer U (1995) FreeWAIS-sf. Available via ftp://ls6-www.informatik.uni-dortmund.de/pub/wais/freeWAIS-sf-1.1/freeWAIS-sf-1.1.tar.gz

6 Dougherty D, Koman R, Ferguson P (1994) The Mosaic handbook for the X window system. O'Reilly and Associates, Sebastopol 1994.

7 Liu C, Peek J, Jones R, Buus B, Nye A (1994) Managing Internet information sources. O'Reilly and Associates, Sebastopol 1994.

8 Hughes K (1994) WWWWAIS. Available via http://www.eit.com/software/wwwwais/wwwwais.html

9 Legato NetWorker User's and Administrator's Guide. Legato Systems, Inc. 1994

10 Postel J, Reynolds J (1985) RFC959: File transfer protocol.

11 Jones A (1995) The Windows ftp daemon. wftpd196.zip