DATA HANDLING AT ECMWF

Baudouin Raoult
Meteorological Applications
ECMWF

1 INTRODUCTION

The MARS archive has been growing as much in size as in variety. It is now reaching its limits and needs to be redesigned in order to cope with the increase in generated data foreseen on the arrival of a new super computer and the never ending requirements to archive new kinds of data types.

2 GROWTH

In 1985 the operational archive was 70 MBytes/day. In 1995 the total archive was of 6 TBytes with a growth rate of 8 GBytes/day of operational data and 9 GBytes/day of research data.

The number of individual items archived is also very large: 120 000 operational fields/day, 140 000 reports/day and 200 000 research fields/day.

The meteorological data is not yet all under the control of MARS. New data types are first checked by the research. Then, when a new data type is being used by the operational suites, it is added to the archive. For that reason, the archive has grown in variety and complexity, as shown in Fig. 1.

This figure shows when each new data type was added to the archive.

In the new future, data from 3D variational analysis, seasonal forecasting, ocean models and ensembles from other centres will also be archived. In a more distant future, experimental analysis feedback and private datasets will have to be handled by MARS.

In order to accommodate all these future requirements, the internal architecture of the software has been changed while its user interface has remained untouched.

3 A BIT OF HISTORY

3.1 The Cyber

MARS is a system based on a file store from Los Alamos named *Common File System* (CFS) running on an IBM MVS mainframe. The first client was a CDC Cyber. At this time the client-server concept did not exist. The clients were called worker machines. The protocol was file based and the networking was done with RHF.

There was no communication between MARS and CFS. The requests would be passed in files to the MARS server that would translate them into a list of *mass* commands, the CFS native language. The list of commands would be passed back to the server that would issue the commands and get the data. See Fig. 2

3.2 The Cray

Later, the MARS client was ported to the Cray running COS. The MARS client would access the on-line field database (FDB) that contained the model output. Communication with the IBM was done via SUPERLINK and *ecfile* was used as a file transport mechanism. An interface between MARS and CFS was written. The MARS server would receive a request, translate it to a list of CFS files, offsets and lengths with the help of a basic database using a VSAM file. This information would be passed to CFS that would extract only the relevant fields, and pass it back to MARS. These results would be saved again in CFS to be fetched by the client, as shown in Fig. 3.

Post-processing facilities were added: sub-area extraction, interpolations and derived fields.

4 UNIX

With the coming of UNIX, new tools were available such as application level networking with TCP/IP, better control of the operating system resources with the language C, text parsing with parsing with *lex* and *yacc*.

So a new MARS client was developed. The previous system could access two sources of data: the FDB and CFS. The idea was to implement an architecture that could access any number of data sources or "*databases*".

5 THE "DATABASE" CONCEPT

The data access should be done in an object oriented fashion (data hiding and polymorphism). A database is a software package that supports four calls: *open*, *read*, *write* and *close* and understands a MARS request (Fig. 4).

The client will scan all available databases and call them one at a time. The MARS request would be passed in the *open* call. The *read* call would return one field at a time. Conversely, the *write* call would be called for archiving data.

Using this technique, any new data source can be added to the MARS system by simply implementing one of these packages.

Any such database can be installed on a network, with the help of a "*network glue*", a piece of software that uses TCP/IP to forward function parameters and return values across a network. Neither the database nor its client has to be changed to use this feature. None of the components involved has to be aware of the presence of the network (Fig. 5).

6 CACHING

Several of these database packages have been implemented: one to access the IBM archive, one to access the Cray FDB and one using the commercial SQL database "Empress". The latter is used to implement the MARS caching system (MCS) for fast access to the archived data, needed for interactive work done using Metview. Any data retrieved from one database can be written to another. This way, a field that is being accessed often is moved "closer" to the user (Fig. 6).

The following request is run twice with an interval of one minute interval between the first run:

```
MARS - INFO    - Request 1 is :

RETRIEVE,
          TYPE       = FC,
          LEVTYPE    = SFC,
          PARAM      = T,
          DATE       = 940601/TO/940605,
          TIME       = 1200,
          STEP       = 12/TO/240/BY/12,
          TARGET     = "foobar",
          GRID       = 1.5/1.5

MARS - INFO    - Calling IBM
```

```
MARS - INFO     - Got 5.55 Mbytes from IBM
MARS - INFO     - 100 fields retrieved from 'IBM'
MARS - INFO     - Request time: 44 min 21 sec
```

and the second run:

```
MARS - INFO     - Request 1 is :

RETRIEVE,
          TYPE       = FC,
          LEVTYPE    = SFC,
          PARAM      = T,
          DATE       = 940601/TO/940605,
          TIME       = 1200,
          STEP       = 12/TO/240/BY/12,
          TARGET     = "foobar",
          GRID       = 1.5/1.5

MARS - INFO     - 100 fields retrieved from 'MCS Research 2'
MARS - INFO     - Request time:  wall: 10 sec
MARS - INFO     - No errors reported
```

Note the difference in request times.

7 MEMBER STATES ACCESS TO MARS

The same scheme will be used for MARS access by the member states. A member state user will run a client MARS on his/her machine, possibly accessing some local database, then a site wide MCS and then the ECMWF MARS proxy, implemented as a database as far as the member state MARS client is concerned. The MCS database would cache any field retrieved from ECMWF, thus avoiding unnecessary network traffic if two users from the same site request the same field (Fig. 7).

8 THE DHS PROJECT

The present design of MARS has reached its limits. The maintenance of the MARS server on MVS is a nightmare. The Cray will be replaced by a more powerful Fujitsu that will produce more data. New data types will be added. So MARS needs a new design using newer technologies.

8.1 Hardware

The MVS mainframe will be replaced by 3 RS/6000 R30 from IBM, with 4 PowerPC 603 CPUs each. Each machine will have 1 GBytes of memory, 1 MByte of cache per processor, 3 GBytes of disk and will share 240 GBytes of disk with the other ones.

A robot with 8 drives will be able to mount two hundred 10 GBytes tapes per hour.

8.2 Software

CFS will be replaced with ADSM, and IBM file store. The MARS server will be written in C++, and its metadata will be stored in a objected oriented database (OODBMS)

To access both new and old systems, a new "database" will simply be added to the list of existing databases, as shown in Fig. 8.

The following figure shows the first analysis of the future system. A *Data Server* will sit on top of ADSM and handle files. Each file archived will be translated into a unique identifier called a *reference*. The MARS Server will translate requests into references and gets its data from the Data Server (Fig. 9).

By clearly isolating the role of the two servers, the Data Server will be able to organise the files in the most efficient way according to the underlaying software and hardware. On the other hand, MARS will have the meteorological understanding of the data and being relieved of the data organisation aspects, will be able to provide a better service than the present system.

8.3 Metadata

The success of the future MARS lies in the ability to represent its metadata, i.e. the "data representing the data". The main problem is that it is impossible to foresee the ideas coming from research. MARS must cope with data types not yet "invented", as well as a variety of very different "objects" such as fields, observations, images, or cross-secions.Using object-oriented technology will help: with *encapsulation* MARS will be able to handle new data types, each data type being "self aware". With *inheritance* it will easily support new attributes and help code reuse (e.g. an ensemble forecast is a forecast plus an ensemble number). With *polymorphism*, the MARS engine will be simple as it will have a single view of all the data types.

9 CONCLUSION

The present design has been successful for more than 10 years, and is nowadays one of the largest and most complete meteorological archives. We are now facing the challenge of redesigning MARS for the next decade. New technologies will be used to enhance the present system into more open and flexible software, that will cope with the endless growth of data, in quantity as well as in variety

Fig. 1: Introduction of new data types in MARS.

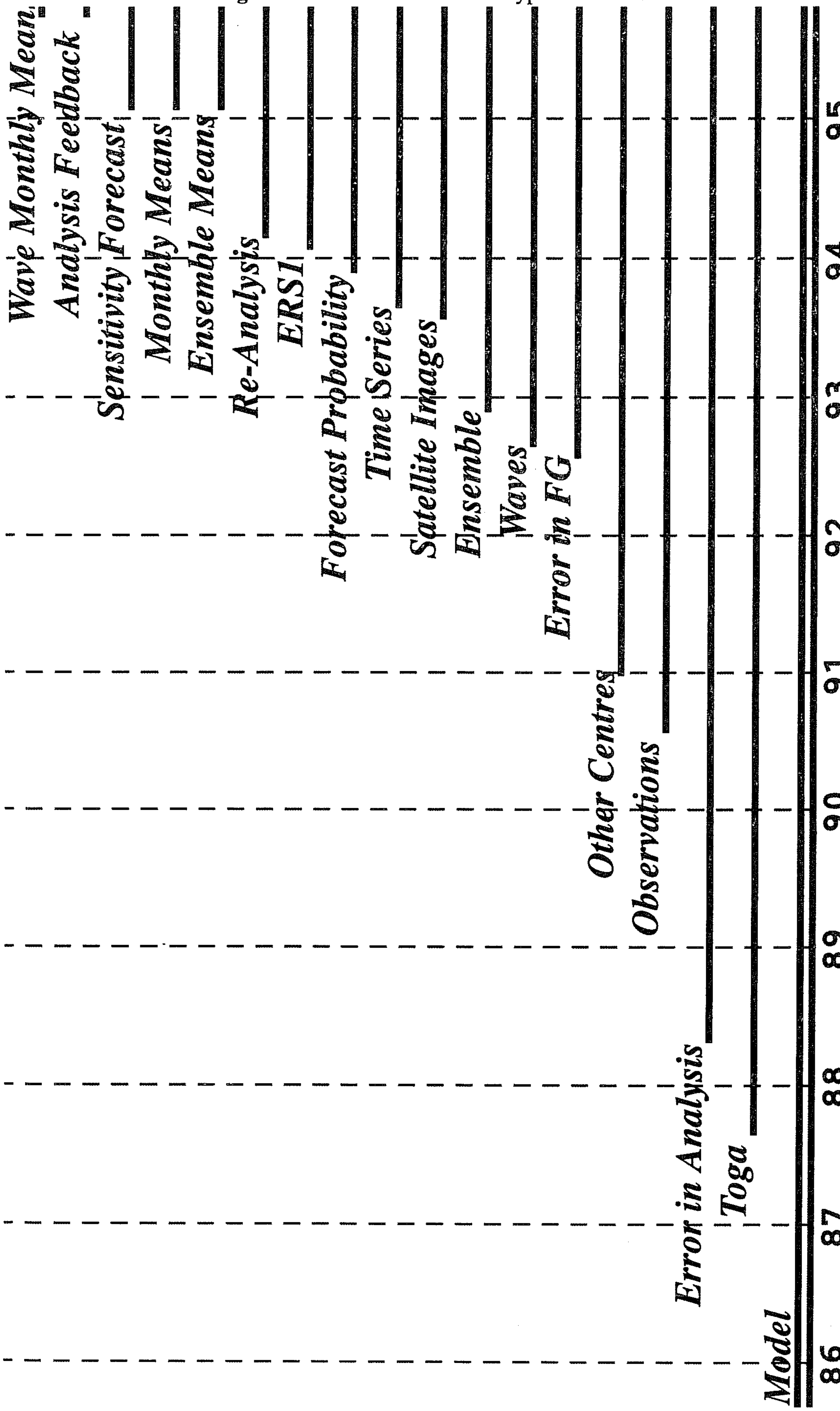

Fig. 2: MARS Architecture - Cyber

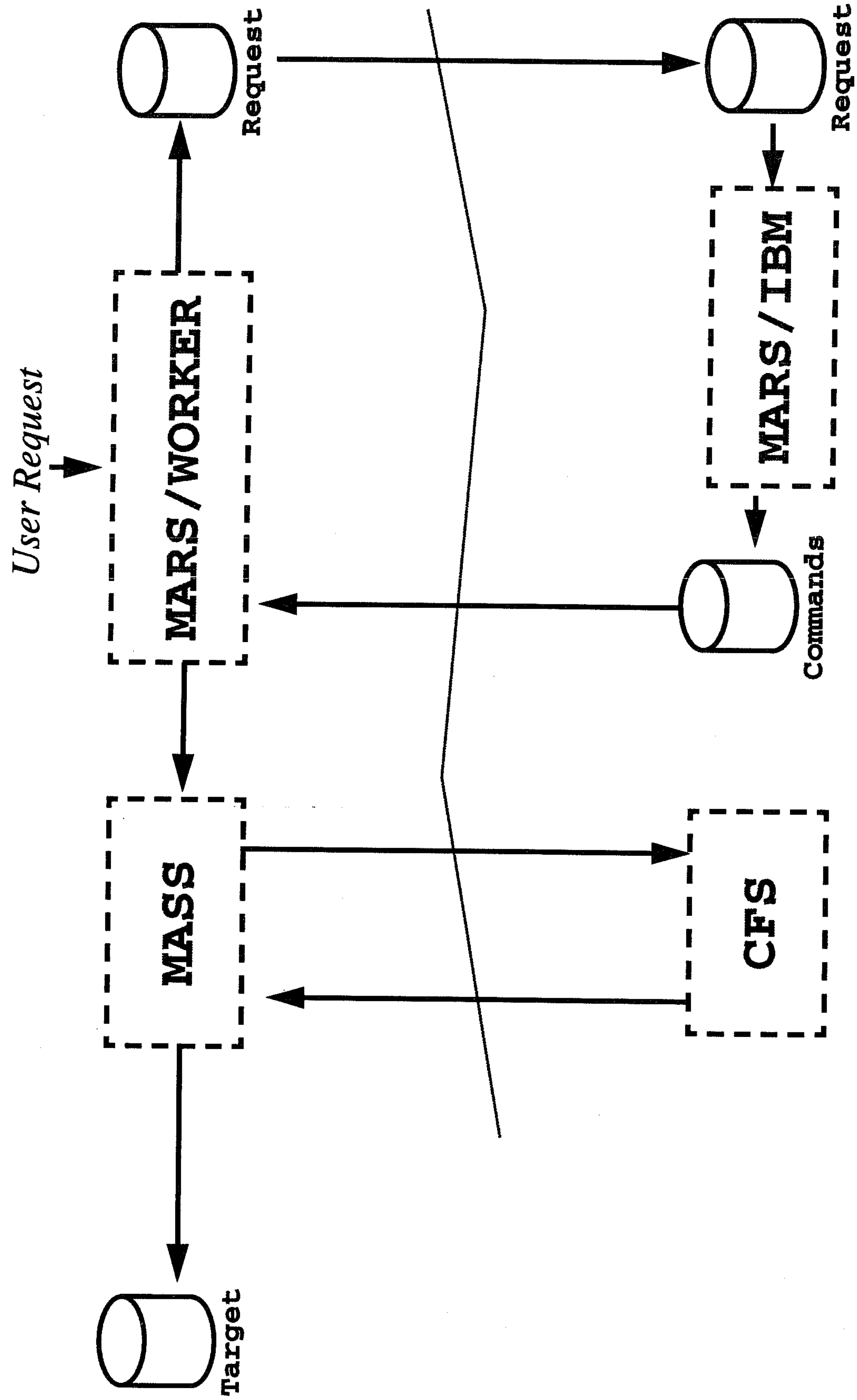

Fig. 3: MARS Architecture - Cray

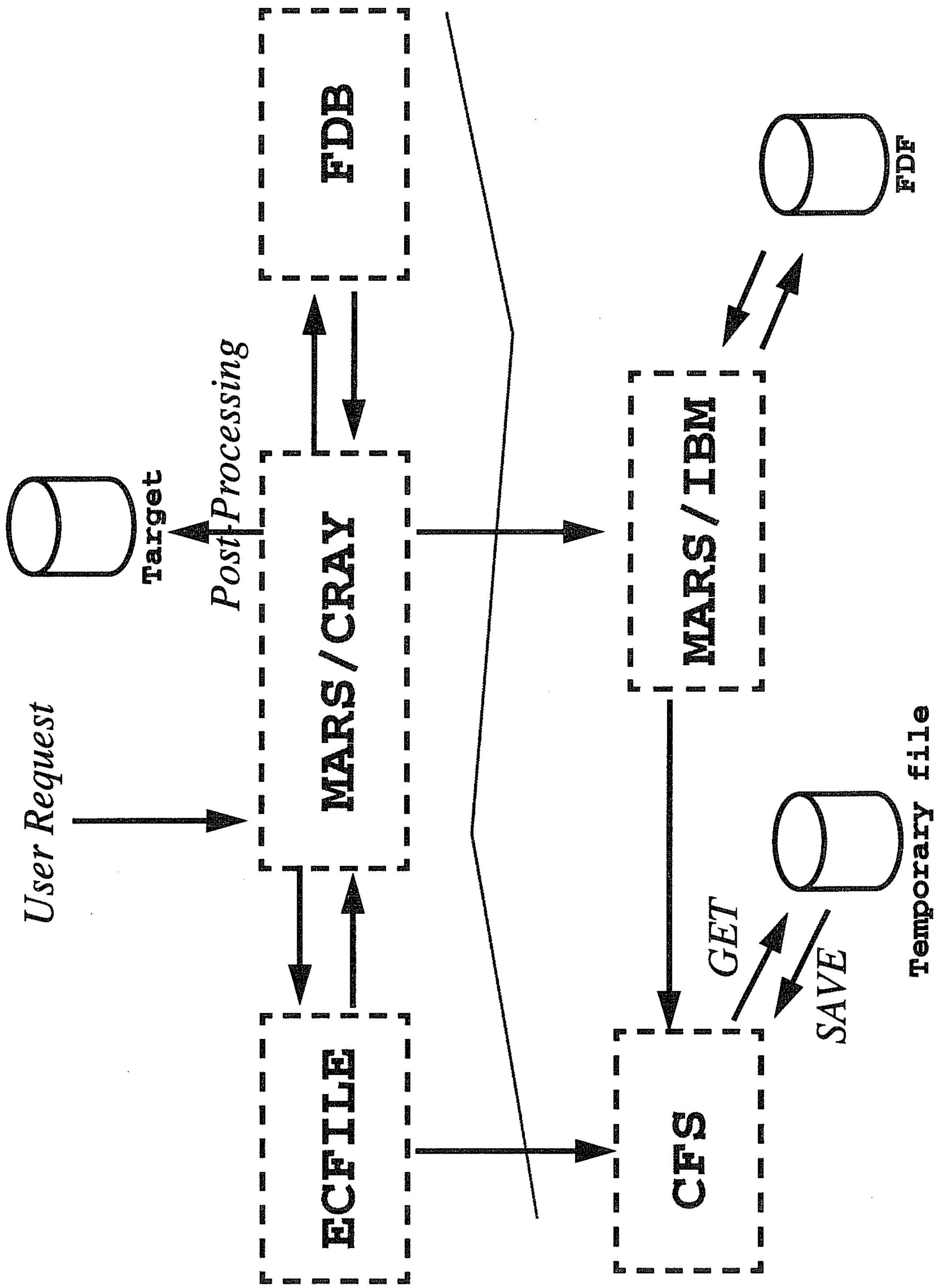

Fig. 4: The "Database" Interface.

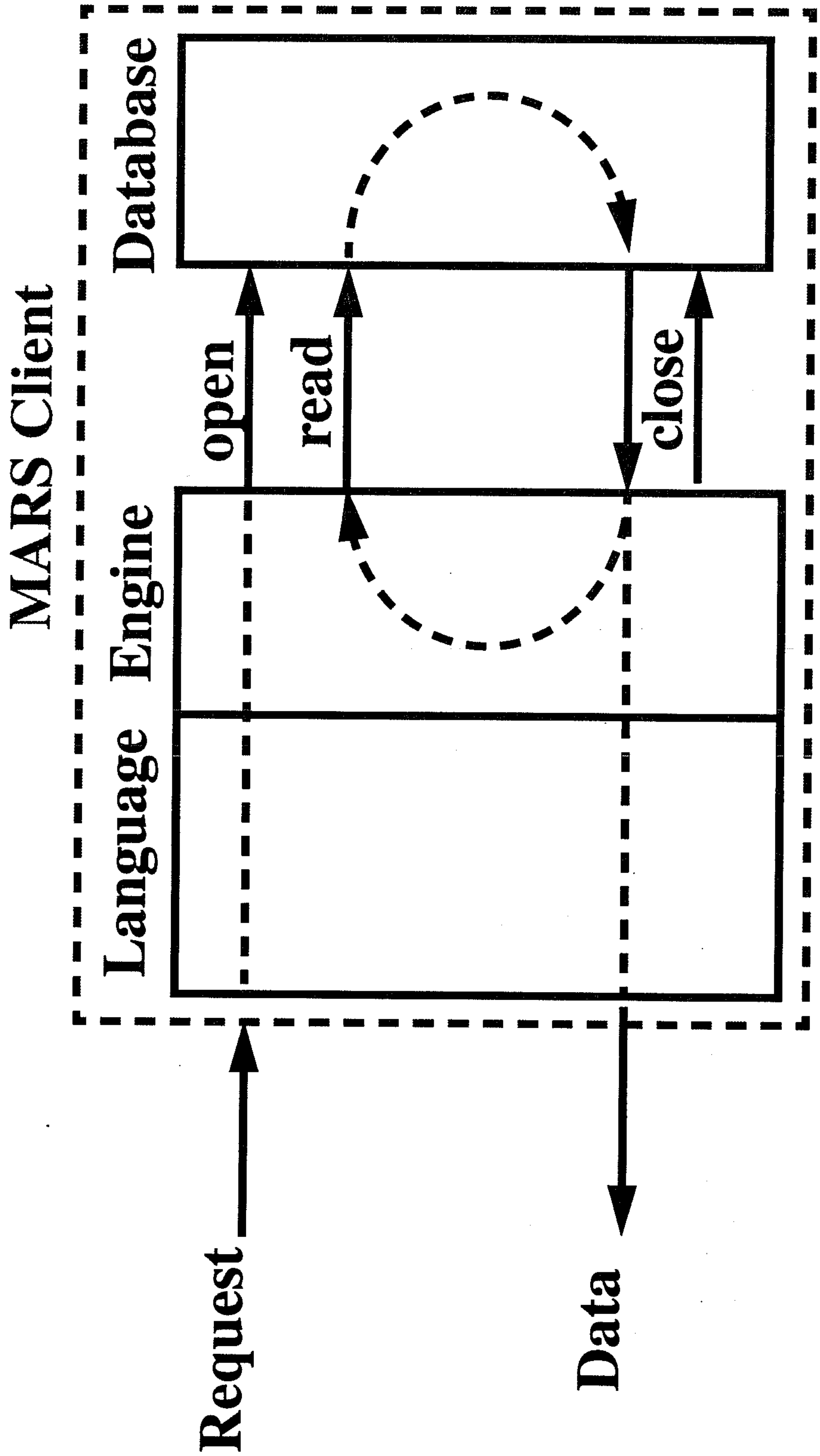

Fig. 5: Networked databases.

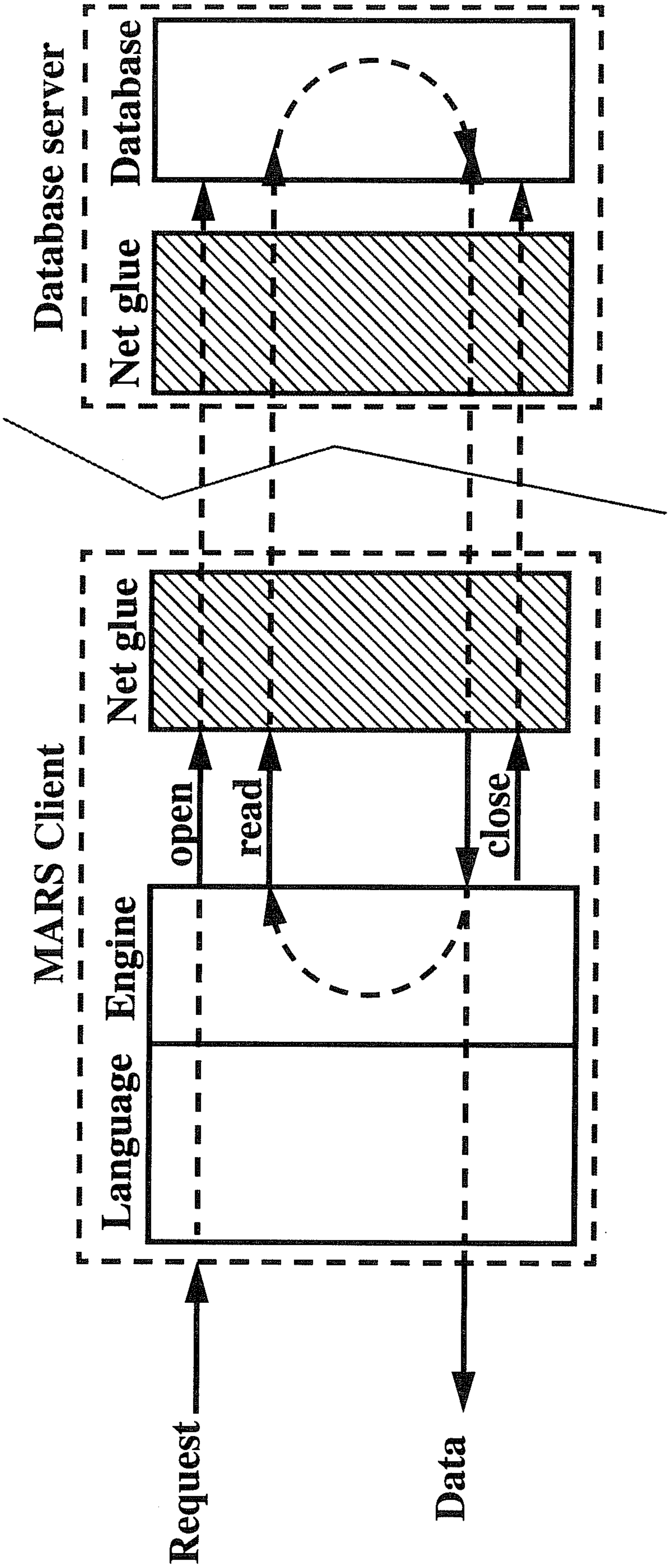

Fig. 6: Data Access and Caching.

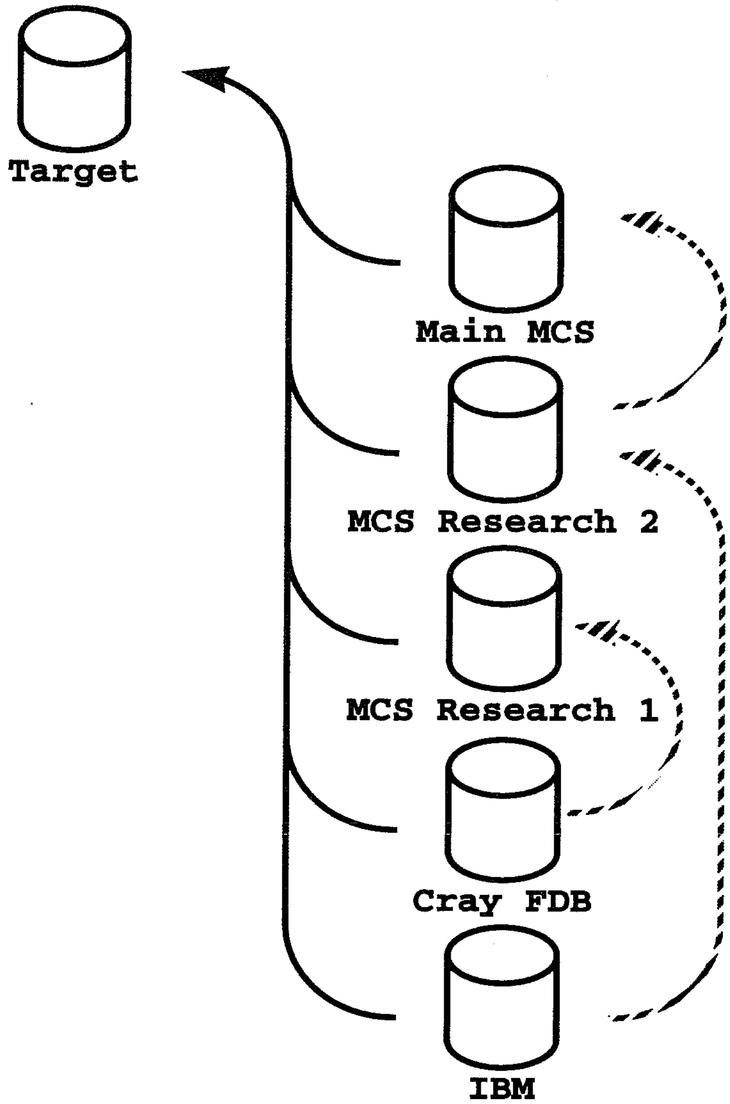

Fig. 7: Member States Access to MARS

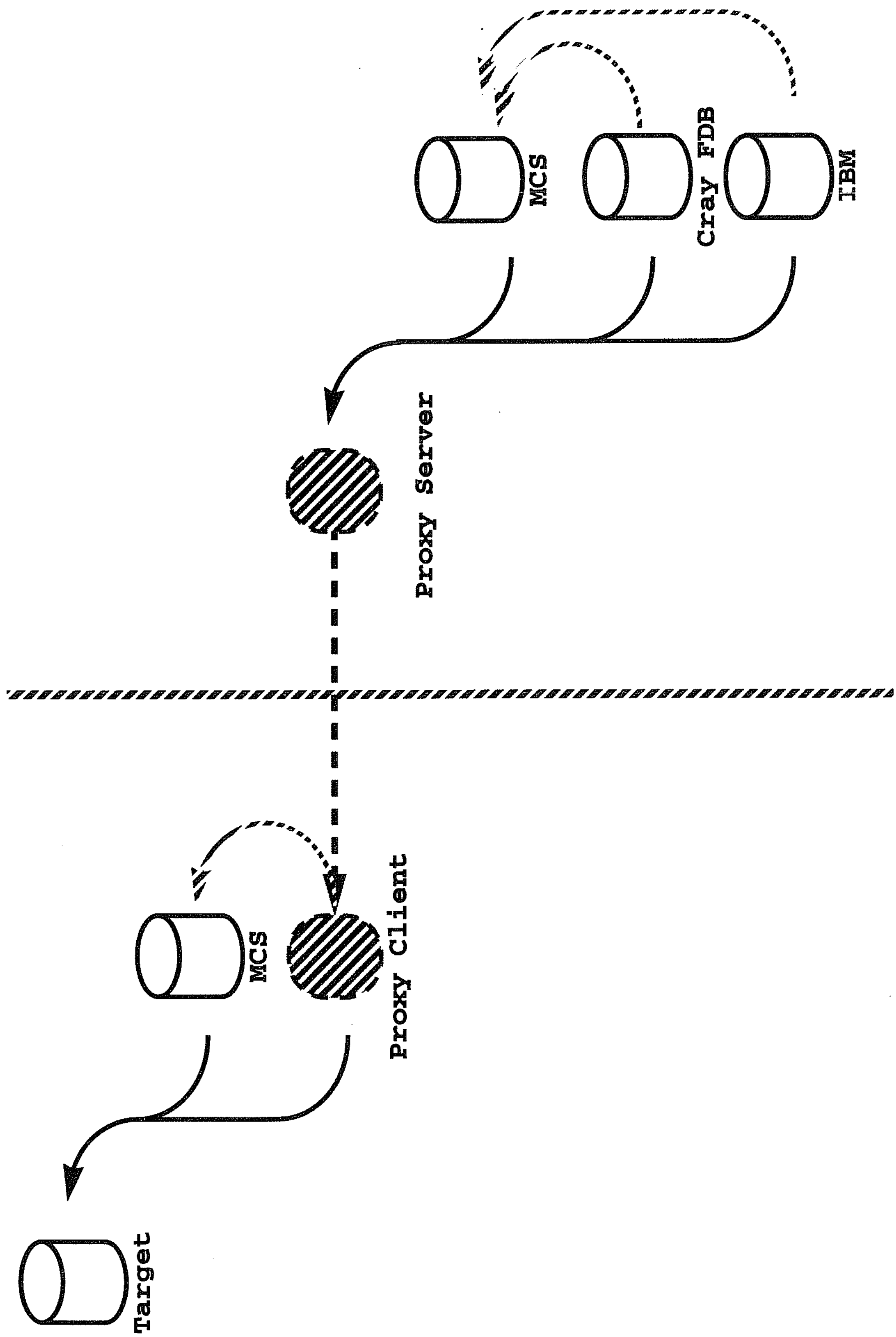

Fig. 8: Accessing Both Archives

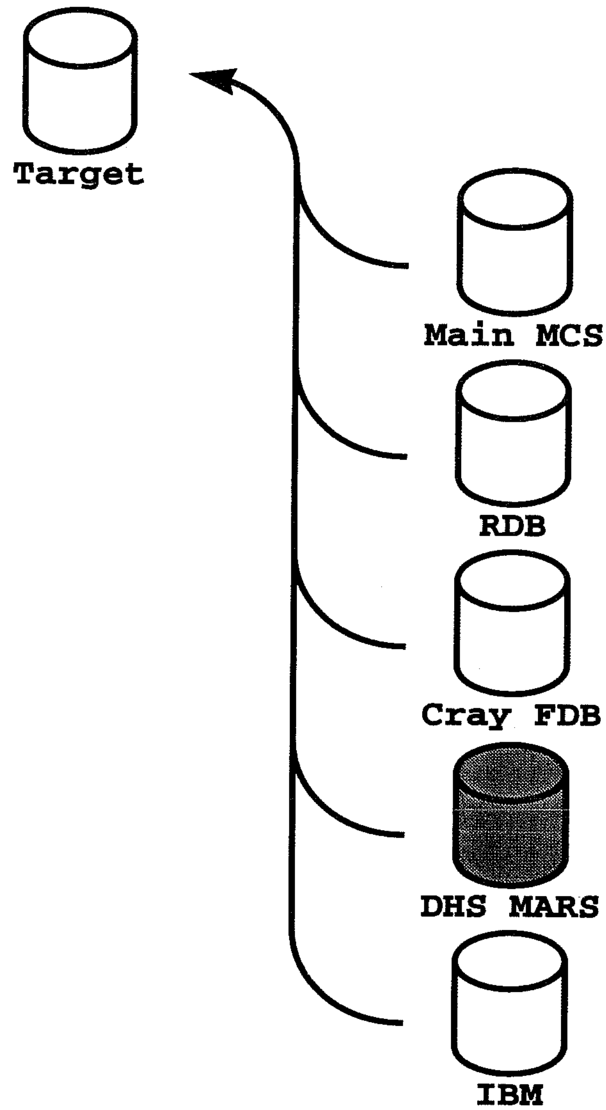

Fig. 9: MARS Architecture - DHS