GCOS 7

Overview

DPS7000/XTA
NOVASCALE 7000

Distributed Computing

REFERENCE
40 A4 61CF 01

DPS7000/XTA
NOVASCALE 7000
GCOS 7

Overview

Distributed Computing

August 1995

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
40 A4 61CF 01

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

To order additional copies of this book or other Bull Technical Publications, you are invited to use the Ordering Form also provided at the end of this book.

Trademarks and Acknowledgements

We acknowledge the right of proprietors of trademarks mentioned in this book.

Intel® and Itanium® are registered trademarks of Intel Corporation.

Windows® and Microsoft® software are registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark in the United States of America and other countries licensed exclusively through the Open Group.

Linux® is a registered trademark of Linus Torvalds.

The information in this document is subject to change without notice. Bull will not be liable for errors contained herein, or for incidental or consequential damages in connection with the use of this material.

Preface

Bull's product offering is evolving within the framework of the Distributed Computing Model (DCM). A key technology involved in this trend towards distributed client/server computing is OSF's Distributed Computing Environment (OSF DCE), implemented initially across UNIX® cells. The keystone of DCE is the RPC (Remote Procedure Call).

As one of the mechanisms for integrating GCOS enterprise systems within a distributed UNIX environment, Bull has implemented the RPC function on GCOS 7 and GCOS 8. This implementation relies heavily on another Bull component, the DCE RPC Ally which is implemented in the UNIX environment and which provides RPC support functions.

This document provides an overview of Bull's implementation of RPC onto the GCOS platforms - with the product known as GCOS DCE.

For an explanation of the terms and abbreviations used in this manual, refer to the Glossary at the end.

Typographic conventions

This manual uses the following typographic conventions. They are based on those used in the OSF DCE documentation:

Bold — Filenames , command names etc. which you must type in exactly as indicated.

Italic — Variable values which you must supply.

`Fixed pitch` — Examples of machine dialogue (e.g. file contents or screen display).

[] — Square brackets enclose optional items.

{} — Braces enclose a list from which you must choose one item.

| — A vertical bar separates the items in a list of choices.

<> — Angular brackets:
- either enclose the name of a key on the keyboard.
- or indicate the individual elements in command syntax, filenames, pathname, etc.

... — Ellipsis points indicate that you can repeat the preceding item one or several times.

Manual Directory

DCE Documentation (delivered on tape with the DCE code)

OSF Introduction to DCE[1]	*86 A2 30VA*
OSF DCE User's Guide and Reference[1]	*86 A2 31VA*
OSF DCE Application Development Guide[1]	*86 A2 32VA*
OSF DCE Application Development Reference[1]	*86 A2 33VA*
OSF DCE Administration Guide[1]	*86 A2 34VA*
OSF DCE Administration Reference[1]	*86 A2 35VA*
DPX20 DCE Message Reference Guide[2]	*86 A2 48WG*
DPX20 DCE GDS Administration guide and Reference[2]	*86 A2 86WG*
DPX20 DCE Administration Guide	*86 A2 98WE*
DPX20 DCE Overview[2]	*86 A2 00WG*

GCOS DCE Documentation

GCOS DCE - Overview	*40 A4 61CF*
DCE RPC on GCOS 7 Administration Guide	*40 A2 62CF*
DCE RPC on GCOS 7 Software Release Bulletin	*40 A2 65CF*
DCE RPC on GCOS 8 Administration Guide	*67 A2 LZ96*
DCE RPC on GCOS 8 Software Release Bulletin	*67 A2 SH24*
DCE RPC Ally Administrative Supplement	*40 A2 64CF*

GCOS 7 Documentation

GCOS 7 System Overview	*47 A2 04UG*
TDS Administrator's Guide	*47 A2 20UT*
GCOS 7 System Operator's Guide	*47 A2 60UU*
GCOS 7 Error Messages and Return Codes Directory	*47 A2 10UJ*
GCOS 7 Console Messages	*47 A2 61UU*

OPEN 7 Documentation

TCP/IP7 End User's guide	*47 A2 30US*
OPEN7 Administrator's Reference Manual	*47 A2 31US*
OPEN7 Administrator's Guide	*47 A2 32US*

[1] Manual also available from Prentice Hall

[2] Paper copy delivered with the software

GCOS 8 Documentation

Workstation Command Language Reference Manual	*67 A2 DH32*
TP8 Programmer's Guide	*67 A2 DH33*
TP8 Administrator's Guide	*67 A2 DH36*
Sockets Programmer's Guide	*67 A2 LC02*
Virtual C Language User's Guide	*67 A2 LC38*
GCOS 8 Software Installation Bulletin - rel. SS440204	*67 A2 SE24*
GCOS 8 Software Release Bulletin - rel. SS440204	*67 A2 SG61*
GCOS 8 Software Installation Bulletin - rel. SS45010	*67 A2 SH40*
GCOS 8 Software Release Bulletin - rel. SS45010	*67 A2 SH30*

Table of Contents

1. Introduction 1-1

1.1 WHY GCOS DCE? 1-1

1.2 ADVANTAGES 1-3

1.3 GLOBAL ARCHITECTURE 1-4

1.4 OTHER MECHANISMS AVAILABLE 1-5

1.4.1 Specific access methods 1-5
1.4.2 RPC and specific access methods 1-5
1.4.3 Network design considerations 1-6

1.5 DOCUMENTATION OVERVIEW 1-7

2. DCE Overview 2-1

2.1 WHAT IS DCE? 2-1

2.2 WHY DCE? 2-4

2.3 REMOTE PROCEDURE CALL 2-5

2.3.1 Operation 2-5
2.3.2 Generating stubs 2-6

2.4 THE CELL DIRECTORY SERVICE 2-7

2.4.1 Architecture 2-7
2.4.2 Binding 2-8
2.4.3 Updating the CDS 2-8

2.5 IMPLEMENTATION 2-9

2.5.1 Software components 2-9
2.5.2 Systems supported 2-10
2.5.3 Restrictions 2-10

3. GCOS 7 Overview 3-1

3.1 IMPLEMENTATION OUTLINE 3-1

3.1.1 Main features 3-1
3.1.2 Three working environments 3-2
3.1.3 Telecommunications facilities 3-2
3.1.3.1 ISO/DSA Telecommunications 3-3
3.1.3.2 Terminals and PCs 3-4
3.1.3.3 OPEN 7 Telecommunications 3-4

3.1.4 Remote operation facilities 3-6
3.1.5 File and catalog management 3-6
3.1.6 Database management 3-7

3.2 ADMINISTRATION 3-8

3.3 LIMITATIONS, RESTRICTIONS 3-9

4. GCOS 8 Overview 4-1

4.1 IMPLEMENTATION OUTLINE 4-1

4.1.1 File management 4-2
4.1.2 Time sharing 4-2
4.1.3 Transaction processing 4-3
4.1.4 Data management 4-3
4.1.4.1 INTEREL facility 4-4
4.1.4.2 INTEREL-SQL 4-4
4.1.4.3 FormsSQL 4-4

4.1.5 Networking 4-4
4.1.5.1 DSA communications 4-5
4.1.5.2 TCP/IP communications 4-6

4.2 THE OPEN 8 SYSTEM 4-8

4.2.1 Open 8 Architecture 4-8
4.2.2 Open 8 Terminal Connectability 4-10

4.3 CONNECTION WITH FCP8 4-12

4.4 ADMINISTRATION 4-13

5. GCOS and DCE 5-1

5.1 GCOS-UNIX COMMUNICATIONS 5-1

5.2 GCOS DCE ARCHITECTURE AND FUNCTIONS 5-3

5.3 THE DCE RPC ALLY 5-4

5.3.1 Functions 5-4
5.3.2 Location 5-5
5.3.3 Bindings 5-5
5.3.3.1 Bindings to the Ally 5-5
5.3.3.2 Bindings to other services 5-6

5.4 RPC ON GCOS 7 - ARCHITECTURE 5-6

5.5 RPC ON GCOS 8 - ARCHITECTURE 5-8

5.6 FUNCTIONS SUPPORTED AND RESTRICTIONS 5-8

Glossary g-1

Index i-1

Illustrations

Figures

1-1 GCOS DCE - Global Architecture 1-4
2-1 The Distributed Computing Environment 2-2
2-2 The Remote Procedure Call........ 2-5
2-3 DCE Implementation 2-9
3-1 GCOS 7 - Three working environments........ 3-2
3-2 GCOS 7 - ISO/DSA communications........ 3-3
3-3 OPEN 7 communications........ 3-5
4-1 GCOS 8 - DSA Communications 4-5
4-2 GCOS 8 - TCP/IP Communications........ 4-7
4-3 Open 8 Architecture 4-9
4-4 Open 8 Terminal connections 4-11
4-5 Connection with FCP8........ 4-12
5-1 Open 8 and OPEN 7 communications 5-2
5-2 GCOS DCE Architecture........ 5-3
5-3 The Ally Functions........ 5-4
5-4 RPC on GCOS 7 5-7
5-5 RPC on GCOS 8 5-9

1. Introduction

Market surveys show that GCOS users are globally satisfied with their mainframes, particularly for transactional applications and batch processing. However they are looking for new services such as electronic data interchange, security services and system management.

The integration of GCOS systems into DCE (Distributed Computing Environment) provides the user with an answer to this problem. It also ensures that any new applications can be developed on a unique, standard software platform.

In this initial phase of the project, Bull is introducing the RPC (Remote Procedure Call) mechanism, which is the keystone of DCE. This document provides an overview of the software components involved and of the implementation of RPC on GCOS platforms - the product known as DCE RPC on GCOS (7 or 8). This implementation uses TCP/IP communications and relies on a closely related component, the DCE RPC Ally which is implemented in the UNIX environment and which provides RPC support functions.

1.1 WHY GCOS DCE?

Bull is committed to assisting its customers in their evolution from a centralised organisation to a more distributed one. In a distributed organisation, the end-user gains access from a desktop system to any information or application in the network. In addition, a prime consideration is one of openness. A user must be able to access services on many different types of system using a standard user interface.

When considering new software investments, the most difficult decision for any customer to make is:

> "What do I do with my existing applications and volumes of data gathered over the years?"

Bull's answer to these and related problems is the Distributed Computing Model (DCM) and associated products. For example, BOS/TP provides access to TP applications on multiple platforms; DDA (Distributed Data Access) allows access to databases on diverse platforms. GCOS DCE provides interoperability between mainframe applications and distributed UNIX applications. It ensures that the investment a customer has made in existing systems and software is preserved while options are made available for future developments.

GCOS DCE is a key component of that phase of the Distributed Computing Model concerned with providing access from enterprise servers to distributed information resources. This is done by integrating GCOS within UNIX cells. The cell is a set of networks, systems and workgroups which is managed as a group.

Core components of these cells are drawn from the OSF's DCE (Distributed Computing Environment). The keystone of DCE is the RPC (Remote Procedure Call). RPC extends the traditional procedure call mechanism from a single system to a distributed computing environment.

Other mechanisms within DCE (such as the directory service or time service) make the development of distributed applications transparent to the underlying protocol or network topology considerations. This fact, coupled with the relative ease of program development under UNIX and the large number of third-party software products available, presents distinct advantages to the user of GCOS systems.

1.2 ADVANTAGES

GCOS DCE is an integral part of Bull's Distributed Computing Model (DCM). It permits distributed processing between the GCOS and UNIX worlds. DCM offers the following advantages to the user:

- enhancement of the GCOS environment with UNIX and desktop services (for example: electronic mail, directory services, distributed printing, EDI - Electronic Data Interchange, distributed application packages from third-party developers),
- maintaining current GCOS applications and current hardware investments,
- enhancement of the UNIX and desktop environment with GCOS functions (transaction processing, database access),
- enhancement of the working environment for the end-user (Affinity Visual), system administrator (ISM - Integrated System Management) and the software developer (CASE - Computer Aided Software Engineering or ISD - Integrated System Development),

Thus *GCOS DCE* opens the door to the UNIX world for GCOS and provides GCOS users with access to the services provided by Bull's DCM. Examples of services (present and future) are: distributed printing, electronic data exchange, customer applications and third party developed applications. *GCOS DCE* provides transparent access to these services and at the same time ensures that new applications can be developed on a unique software platform - UNIX. The advantages of the UNIX platform are well known:

- a state-of-the-art application, development and execution environment,
- relative ease of program development,
- a large number of third party software products available.

1.3 GLOBAL ARCHITECTURE

The implementation of OSF's RPC onto the GCOS platforms is called DCE RPC on GCOS (7 or 8). The global architecture is shown in figure 0-1. A client application on a GCOS system can invoke a procedure on a remote UNIX system using RPC over UDP/IP or TCP/IP. The RPC mechanism communicates the request to the application on the remote system. The server processes the request and calls the necessary procedure code. On completion of the procedure, the server transmits any return data back to the client using the RPC mechanism.

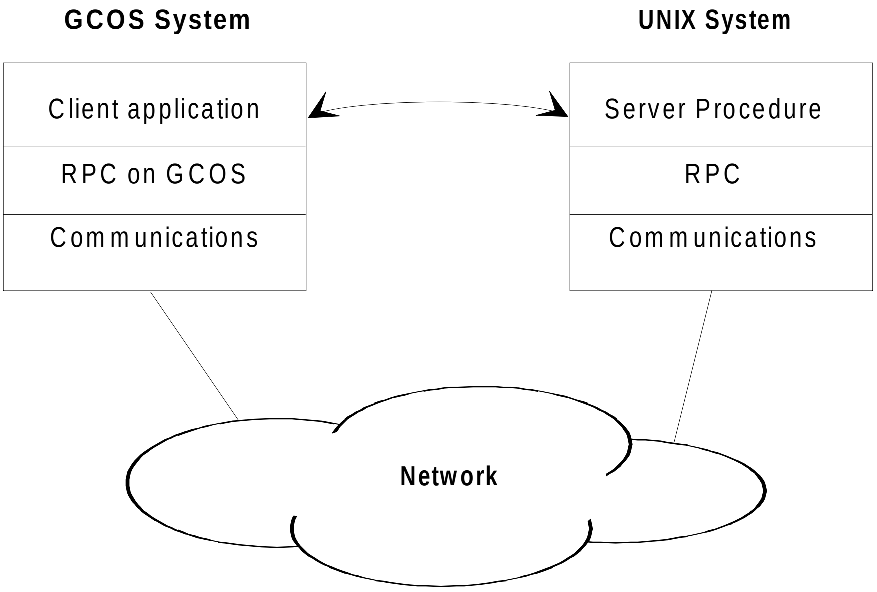

Figure 0-1. GCOS DCE - Global Architecture*Error! Reference source not found.*

The RPC mechanism is transparent to the client application. The client can access local or remote services without taking into account their location, so the location of a service in the network is totally invisible to the end user. The service executing on the remote system appears to be executing locally.

GCOS DCE relies closely on a second component to provide access to DCE basic services. This is the DCE Ally, implemented on a UNIX system. It is described in more detail in section 5.

1.4 OTHER MECHANISMS AVAILABLE

The successful distribution of applications in a network depends mainly on good communications access methods between the components of the application. Various methods exist.

1.4.1 Specific access methods

Certain applications have their own built-in access mechanisms. They are specific to a given application or type of application. SQL (Structured Query Language) is used particularly for information retrieval from databases. FTP (File Transfer Protocol) is used to transfer files across a TCP/IP link. FTAM (File Transfer Access Method) is used in the OSI environment and UFT (Unified File Transfer) is used with DSA.

Certain families of applications may use a common access method. For example CPI-C (Common Programming Interface - Communications) is a programming interface that is used in CTP (Co-operating Transaction Processing). It allows X-Open compliant systems to communicate in a peer to peer mode. These systems may implement the XCP2 protocol or the SNA LU6.2 protocol (IBM systems). It is specifically used for communications between transaction processing applications on GCOS and UNIX systems.

1.4.2 RPC and specific access methods

The mechanisms mentioned above were designed with specific applications in mind. They are therefore ideally suited to these applications. But when a client application requires access to a number of different types of service, the cost of building in all these different access mechanisms becomes prohibitive. For example, an accounting program may need to invoke a transaction with a remote system to update a data file, then transfer a file to another remote system and then print an inventory on the printer of a remote server.

RPC provides a convenient and standard method of communications between any two applications in a TCP/IP network which have an API (Application Programming Interface). Other methods of communication are generally more restricted in their use. The development of distributed applications with these specific mechanisms requires knowledge of many systems.

In addition, specific access methods require a fixed, non-dynamic network topology. Each application must know the location of each correspondent application. If the location of an application or a system changes, all the other applications must be modified to suit. DCE provides a directory service which can be accessed at the start of a connection, to locate the required service. Any changes in network topology are updated dynamically in the directory service by the concerned systems themselves.

Specific access methods tend to be fast and therefore cheaper to run than generalised methods. However they may be more costly at the development stage (one access method per application). Generalised access methods, such as RPC, are simpler to implement (one access method for all remote applications), and therefore cheaper to develop than specific methods. However the operational overheads (delays, communications costs) may be higher.

You must take into account these factors when designing your network.

1.4.3 Network design considerations

The initial conception and design of a network is a crucial phase of a project. It should consider not only the physical topology of the network (systems, communications lines and protocols), but also the projected use of the network (which applications will be used, where they will be located, how many exchanges, what type ...). In addition, a good understanding of DCE RPC will help to optimise its use in the network.

The success of a project depends on this initial design phase. Bull provides technical services in the areas of network design, configuration and training. If you have any questions or problems, contact your Bull representative who can guide you to the right choices.

1.5 DOCUMENTATION OVERVIEW

The following list indicates briefly the content of the GCOS DCE documentation set. Reference numbers for all these documents are given in the Manual Directory at the front of this book.

In addition to the GCOS DCE documentation, developers of distributed applications in the DCE environment will require access to the OSF DCE documentation set. This documentation is also listed in the Manual Directory. For an introduction to the DCE documentation set, refer to Appendix A of the document *Introduction to OSF DCE*.

GCOS DCE - Overview (the present document)

This manual presents the GCOS DCE offer at a general level. It describes the principles of Bull's integration of its GCOS products into the UNIX world, using OSF's DCE.

It has been written for all technical personnel who are involved in this type of evolution of the GCOS world within Bull's Distributed Computing Model.

DCE RPC on GCOS 7 - Administration Guide

This manual describes how to install and configure DCE RPC on GCOS 7 and how to develop a distributed application on GCOS 7.

It has been written for system administrators who will install the product and for application programmers who will use it.

DCE RPC on GCOS 7 - Software Release Bulletin

This manual describes how to install DCE RPC on GCOS 7 and gives information on maintenance and debugging of the product.

It has been written for system administrators who will install the product.

DCE RPC on GCOS 8 - Administration Guide

This manual describes how to configure DCE RPC on GCOS 8 and describes key elements that enable the transparent distribution of applications across multiple systems.

It has been written for system administrators who will administrate the product and for application programmers who will use it.

DCE RPC on GCOS 8 - Software Release Bulletin

This manual describes how to install DCE RPC on GCOS 8.

It has been written for system administrators who will install the product.

DCE RPC Ally - Administrative Supplement

This manual describes the differences between programming an RPC client in the OSF DCE environment and in the GCOS environment and explains how to install and configure the DCE RPC Ally.

It has been written for the system administrator who will install the Ally and for application programmers who will use it.

2. DCE Overview

The market trend is toward the definition and implementation of application programming interfaces (API) which provide simplified access to new services and applications developed on other systems. The user is faced with the problem of how to access all of these services. The OSF consortium, backed by the main world-wide vendors, has proposed DCE as a solution to this issue. It allows users to develop and execute applications in a multi-system, multi-vendor environment, via the APIs, as if they were running on a single system.

Today, DCE is the best, complete environment available that can be used to offer these new services to the GCOS user.

For a complete description of OSF's DCE, refer to the OSF documentation; specifically *Introduction to OSF DCE*. The present section gives a brief summary of the important points, based on that documentation.

2.1 WHAT IS DCE?

OSF's Distributed Computing Environment is a set of tools and services for the creation and execution of distributed applications in a heterogeneous environment. Development of distributed applications is made independent of the communications protocols, system hardware, operating system or data formats used.

Figure 0-1 illustrates the Distributed Computing Environment. The key component of this architecture is the Remote Procedure Call (RPC). RPC allows a program on one computer system to call a procedure on a remote system elsewhere in network. This allows a system (GCOS for example) to make use of services which are not implemented locally (on UNIX for example).

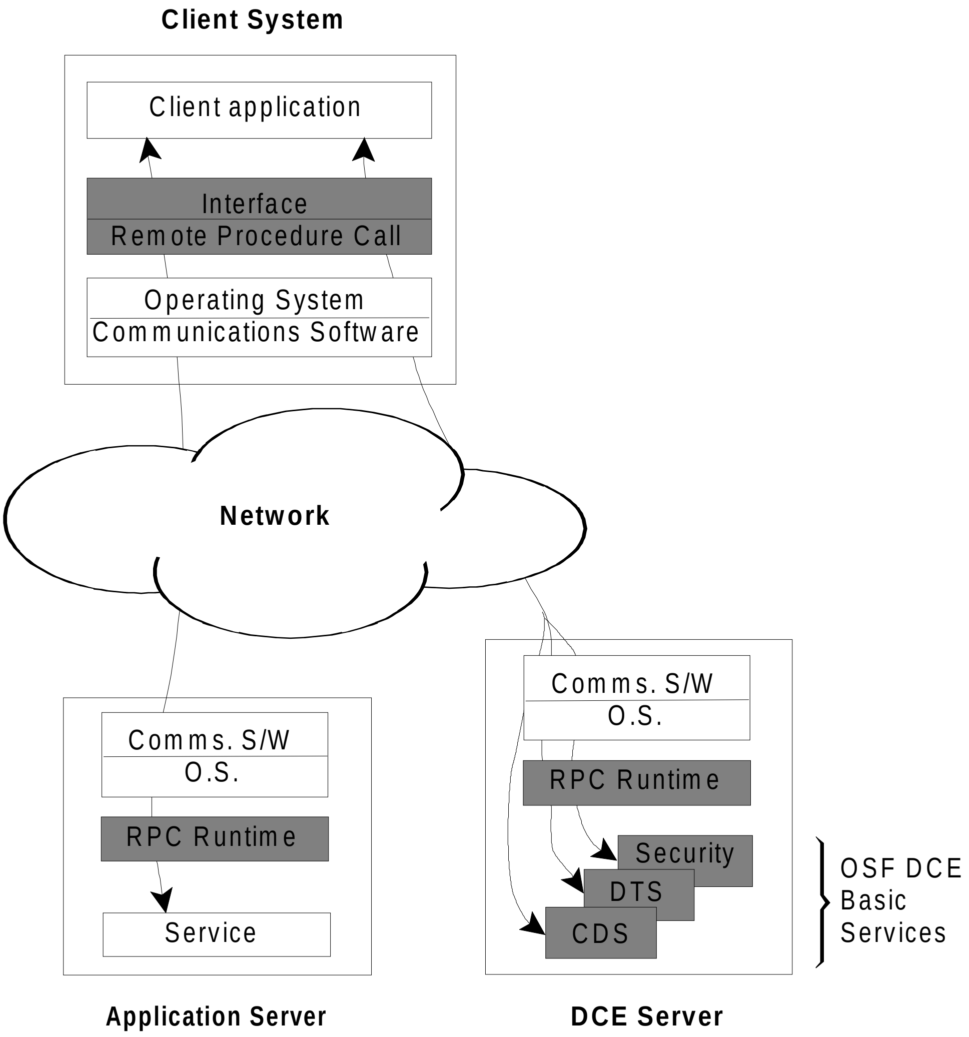

Figure 0-1. The Distributed Computing Environment*Error! Reference source not found.*

RPC is built on the Client/Server model. It allows a client program to call a procedure on a remote server elsewhere in the network.

As well as the RPC, DCE provides a number of basic services which use RPC. These include:

- time service DTS (Distributed Time Service which provides synchronisation of system clocks)

- security mechanism Kerberos (checking of passwords and access rights),

- directory service CDS (Cell Directory Service which provides names and access routes to remote systems within the OSF DCE cell),

In addition to these basic services, a DCE network also contains the UNIX servers providing the services required by the client applications. All of these services may be implemented anywhere in the network.

NOTE: Not all of the DCE basic services are supported by GCOS DCE. Refer to section "GCOS and DCE" for more details

2.2 WHY DCE?

Given that a company or other organisation requires distributed computing capabilities, DCE provides a number of advantages which Bull is currently exploiting. These benefits can be grouped under five main headings:

- Tools and services for distributed applications

DCE provides an environment for developing and running applications on a distributed system. Tools such as the RPC and Threads assist in the development of applications. Services such as CDS, DTS and the Security service provide the support necessary to run them.

- Integrated and comprehensive services

The DCE components are well integrated in that they use each other's services whenever possible (many of these services are themselves distributed applications). DCE services address some of the new problems created by distributed systems, such as data consistency and clock synchronisation. Finally DCE includes management mechanisms for administering the distributed environment.

- Interoperability and portablity

The DCE architecture is oriented towards heterogeneous rather than homogeneous systems. It allows for different operating systems and hardware platforms running at the various nodes of a network and it can accommodate a wide range of networks. Processes on totally different machines can interoperate and applications developed using DCE are portable to any other DCE system.

- Data sharing

DCE supports data sharing through its directory and file services. A user anywhere in the network can share data by placing it in a file. The file is then accessible by any authorised user.

- Global computing

DCE systems can cooperate with computing environments outside of DCE. For example the Directory Service can interoperate with X.500 and with Domain Name Service systems. In this way, users within DCE can access information about the outside world.

In the case of file services, DCE's distributed file system looks like one global file system. Users anywhere in the world can address the same file using a single global name.

2.3 REMOTE PROCEDURE CALL

2.3.1 Operation

Figure 0-2 illustrates the remote procedure call. A client application can invoke a procedure on a remote system using RPC. The interface to the RPC runtime software is standard on all systems but the interface with the client application depends on local implementation. The link between the client application and the RPC Runtime is made with a piece of software called a stub.

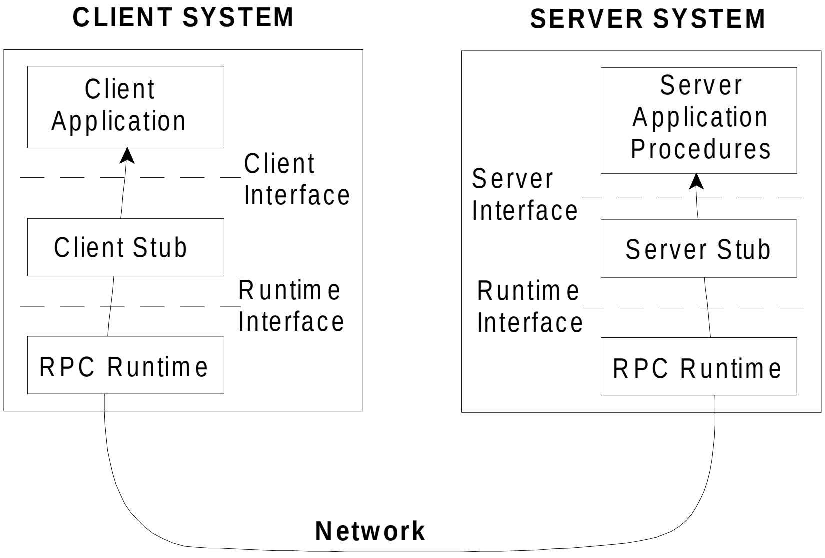

Figure 0-2. The Remote Procedure Call*Error! Reference source not found.*

The client application uses the client stub to request a service provided by the server system. The client stub acts as a local representative of the called remote procedure.

When the client application invokes the procedure, the stub organises any data into a format which can be transmitted to the server and it uses routines of the RPC Runtime library to communicate with the server. On completion of the remote procedure, it organises any result received into a format that can be processed by the client.

On the server system, when the RPC Runtime receives an RPC request it uses a server stub to request the service required by the client.

The server stub acts as a local representative of the calling remote client. It organises received data into a form which can be processed by the server and then calls the procedure code which executes the requested operation. On completion it organises any data into a form which can be transmitted to the client and uses routines of the RPC Runtime library to communicate with the client.

2.3.2 Generating stubs

The application programmer generates the stubs at the time that he writes the client application. The programmer first defines the interface of the remote service using the DCE Interface Definition Language (IDL). The interface defines the group of operations the remote service can perform, the parameters which must be provided and the return values expected.

The next step is to compile the IDL file with a DCE tool, the IDL compiler. The output of the compilation consists of the source code for the client stub, the server stub and a header file. The IDL compiler may additionally generate certain client and server auxiliary files.

The client stub represents the interface of the remote service. It is ported to the client system where the programmer writes application code which calls the operations represented by this interface. In this way DCE allows the logical "exporting" of APIs (Application Programmatic Interfaces) from a remote server system to the client system. The API of the remote service appears at the client system.

The server stub represents the interface of the client application. It is ported to the server system where the service is resident. This service will implement the operations defined in the IDL file. The server stub unpacks the arguments sent by the client and makes a call to the required service as if the client application had called it locally.

The last step is to link the client stub with the client application along with the RPC runtime code. Similarly, on the server system, the server stub must be linked with the remote service. The result is a distributed application comprising two parts, a client side and a server side. The stub code and the RPC Runtime convert what looks like a local procedure call from the client side into a network communication with the server side.

2.4 THE CELL DIRECTORY SERVICE

A distributed system may contain many users, machines, and other resources, along with large amounts of data, all geographically dispersed. The distributed system's attributes, such as number of users, location of servers, and contents of data, are continuously changing. It is difficult to keep track of this potentially large, geographically distributed, rapidly changing system. A dynamically updated directory service can help solve this problem.

2.4.1 Architecture

In the DCE environment, the network is split into cells for administrative purposes. The DCE cell is a group of systems running DCE which are administered as a unit. For example, consider an organisation comprised of several departments, each in a different building and each operating its own budget. The systems of each department could form a single cell. Users in one cell may be able to access resources in another cell but this access would be typically less frequent and more restricted than access to resources within the user's own cell.

The DCE Directory Service is a distributed directory containing information about resources in the distributed system. Resources may be users, systems, and RPC-based services. The directory service contains the name of the resource and its attributes (location of a server, a user's home directory, ...). It can be accessed by any RPC client in the network.

The directory service has a hierarchical structure and comprises three main components:

- The CDS (Cell Directory Service) is a distributed, replicated database. It contains information about the resources in a DCE cell. Every cell has at least one CDS.

- The GDS (Global Directory Service) is also a distributed, replicated database. It is an international standard directory service which connects the local CDSs into one world-wide hierarchy. It is based on the CCITT X.500 international standard.

- The GDA (Global Directory Agent) which acts as a go-between for the CDS and GDS.

The GDS and CDS are accessed via a common API, the XDS (X/Open Directory Service) API.

2.4.2 Binding

When a client sends an RPC to a server, it must first locate the server. This process is called binding. The client may have a direct way of locating the server (the server's address may be hard-coded into the program or it may be stored in a file). A more flexible way of locating the server is to use the Directory Service.

The client can use RPC to access the directory server and locate the required remote service.

2.4.3 Updating the CDS

In order for the directory service to contain information on resources, each server must first advertise itself in the directory service. The server adds itself to the directory service database and includes information about the interfaces it implements, the protocols it uses and where it is located.

In this way a server can move within the network or there can be multiple servers implementing a given interface, without affecting the client. The client continues to access the directory service, a well known central source of information, to find out where the server is located.

2.5 IMPLEMENTATION

2.5.1 Software components

Figure 0-3 shows the software implementation of OSF's DCE. It is a layer (comprising a number of sub-layers) between the operating system and communications software, on one hand, and the distributed applications, on the other. Several technology components work together to implement the DCE layer. A brief description is given below.

- Threads are used for the management of multiple threads of control within a single process. This component is mandatory if the host operating system does not support threads

- RPC is the mechanism which allows a program on one machine to call a procedure on another machine as if it were local. This component was described earlier.

- DTS (Distributed Time Service) provides synchronised time on all DCE systems.

- CDS (Cell Directory Service) is the name service for an OSF DCE cell. This component was described earlier.

- DFS (Distributed File Service) allows users to access and share files stored on a File Server anywhere in the network.

- Security Service provides secure communications and controlled access to resources in the distributed environment. DCE security covers three aspects: authentication of users, secure communications and authorisation of access.

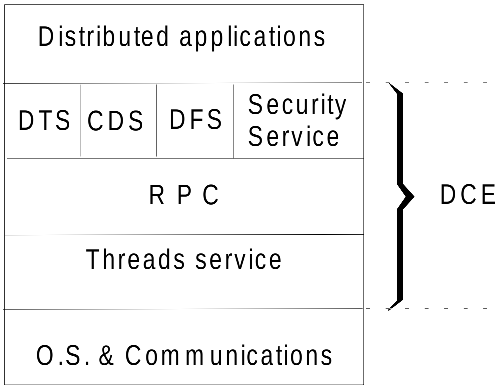

Figure 0-3. DCE Implementation*Error! Reference source not found.*

2.5.2 Systems supported

Bull supports OSF DCE on the Bull DPX20 range of UNIX systems running AIX 3.2.5 or later.

PC DCE is supported on PCs running Windows 3.1.

GCOS DCE is supported on GCOS 7 and GCOS 8 systems.

2.5.3 Restrictions

On GCOS systems, Bull is providing only the client-side RPC mechanism and the Cell Directory Service (CDS) in this initial release of the offer.

For full details on the standard DCE functions supported, refer to the Software Release Bulletin provided with your DCE software.

3. GCOS 7 Overview

3.1 IMPLEMENTATION OUTLINE

Within Bull's DCM model, GCOS 7 is classed as a high volume transactional and database oriented operating system. It is the operating system for Bull's DPS 7000 range of computers. This range extends from the compact DPS 7000 to the top of the range DPS 7000/870. This section provides a short reminder of what GCOS 7 does with specific emphasis on the DPS 700 hardware. For more information refer to the GCOS 7 documentation set, starting with *GCOS 7 System Overview*.

3.1.1 Main features

GCOS 7 offers the following main features:

- IOF (Interactive Operation Facility), a user-friendly operator interface with its own interactive command language, high level language processors and other tools,
- TDS (Transaction Driven Subsystem), an environment for creating and running transaction processing applications,
- a batch job management system,
- access to telecommunications networks,
- remote operation facilities,
- file and catalog management system,
- databases, database management systems and fourth generation access languages,
- system support facilities for system administration.

3.1.2 Three working environments

The GCOS 7 operating system can be accessed via one of three working environments. They are illustrated in figure 0-1.

TDS allows users to create and run transaction processing applications. It comprises a facility for generating the application and a set of tools for writing applications.

The batch job management system accepts jobs in the form of a stream containing one or more job descriptions. Users write job descriptions using JCL (Job Control Language) or possibly GCL. Each job description defines a job to be run for a particular user. Jobs are generally created on disk and then run using a command which is input from the IOF environment.

IOF is a time sharing environment which allows multiple users to access GCOS 7 processors simultaneously. Users access the system from a terminal or workstation using the GCL (GCOS 7 Command Language).

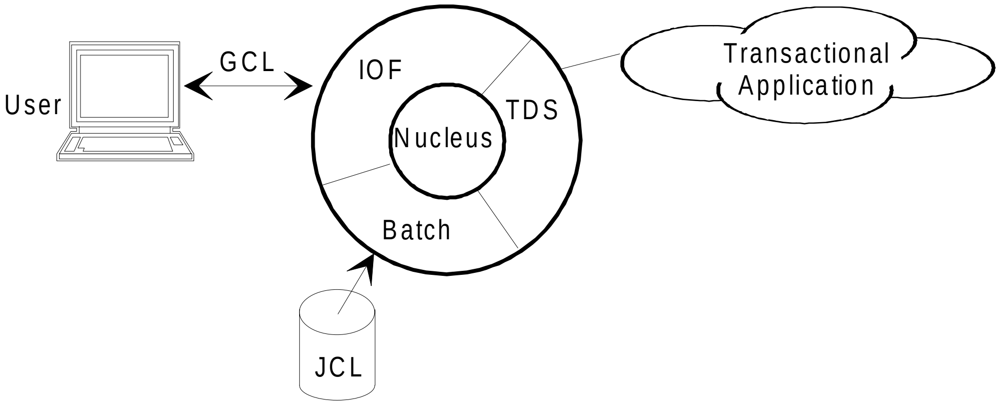

Figure 0-1. GCOS 7 - Three working environments*Error! Reference source not found.*

3.1.3 Telecommunications facilities

DPS 7000 systems are normally used in networks. In this way users can exploit the processing power and storage resources of a number of different systems. The distribution of processing resources over a network makes economic sense, reduces the impact of failures and avoids the duplication of effort.

3.1.3.1 ISO/DSA Telecommunications

GCOS 7 can connect to remote systems via LANs and public or private WANs using its ISO/DSA facilities. ISO/DSA is Bull's architecture for developing communications software and hardware. It provides access to other Bull systems (GCOS 6, GCOS 8, or UNIX) as well as to other GCOS 7 systems. These connections are called the "primary network". The ISO/DSA communications accesses are shown in figure 0-2.

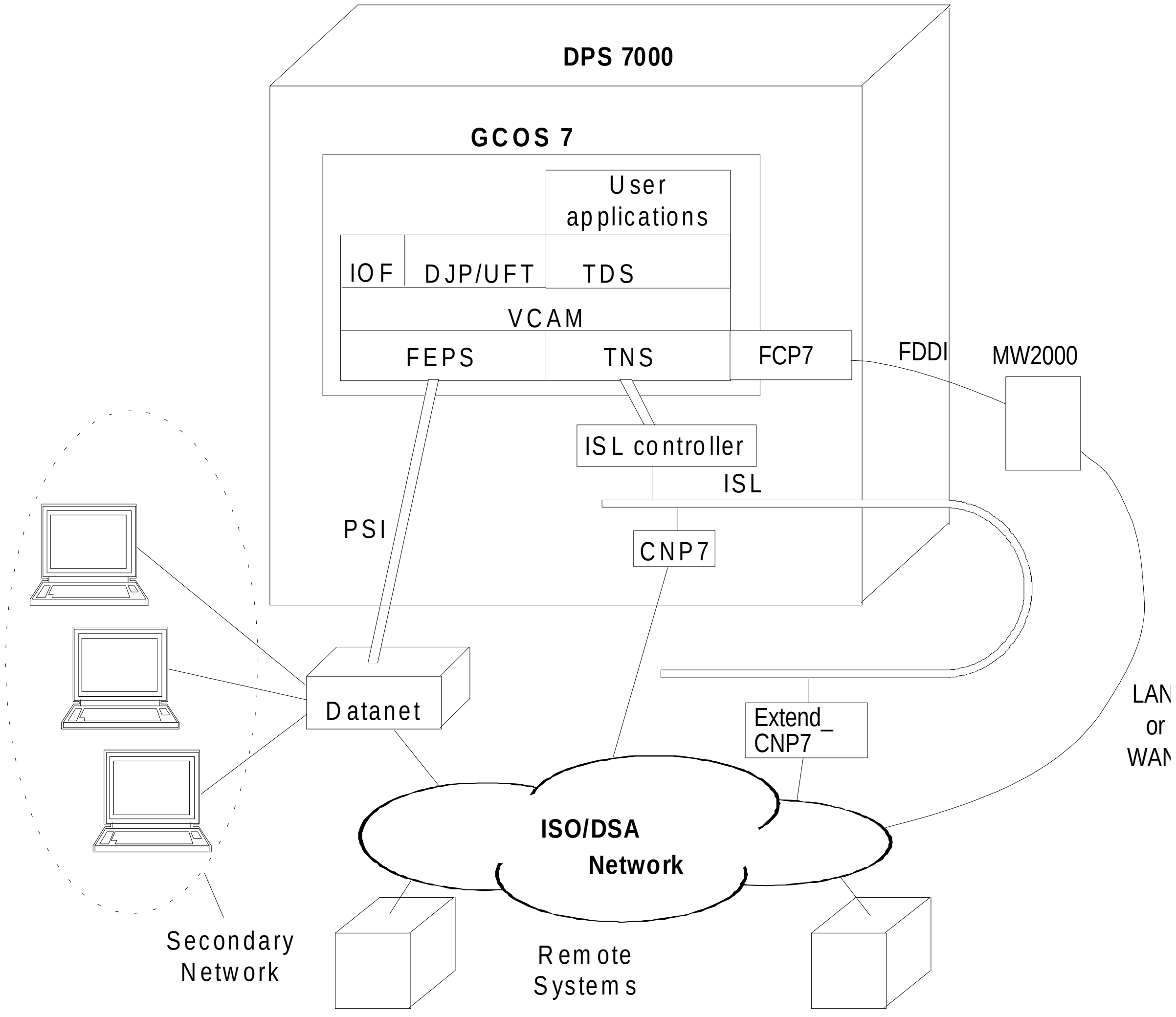

Figure 0-2. GCOS 7 - ISO/DSA communicationsError! Reference source not found.

The interface to ISO/DSA is provided by VCAM (Virtual Communications Access Method). VCAM manages connections and dialogs with remote systems, security checks for incoming calls and internal communications between applications on the same system.

Two channels are available for communication with the network:

- FEPS (Front End Processor Support) which provides the interface to a standalone network processor system (Bull Datanet) via the PSI channel (Peripheral Subsystem Interface).

- TNS (Transport and Network Subsystem) which provides the interface to an integrated network processor (CNP7) via the ISL controller and ISL cable (Inter-System Link).

3.1.3.2 Terminals and PCs

Terminals connected to GCOS 7 form part of the secondary network. The secondary network is a number of terminals connected to the same GCOS 7 system. The secondary network is connected to GCOS 7 via the network processor (Datanet or CNP7).

Terminals may be connected directly to the network processor (one terminal per line) or in clusters via a cluster controller. The first solution avoids possible bottle necks during peak hours, the second solution reduces installation and running costs.

GCOS 7 supports a number of terminal types for accessing applications. Also PC's running terminal emulators can access the software as though they were standard terminals.

Traditionally the Bull Questar 210 family of terminals (supporting the 7107 profile) is used to connect to GCOS 7 applications. IBM compatible PCs running emulators (Affinity for example) can be connected directly as terminals.

3.1.3.3 OPEN 7 Telecommunications

OPEN 7 is a UNIX system which runs as a GCOS 7 subsystem on the DPS 7000. It provides GCOS 7 with access to the UNIX world and, in particular, to its telecommunications facilities. It interfaces to GCOS 7 via the HSL (High Speed Link) which is an area of shared memory between OPEN 7 and GCOS 7.

It provides access to an Ethernet LAN or an X.25 WAN using TCP/IP or UDP/IP and provides standard UNIX file transfer facilities (**ftp**, **rcp**, **nfs**) as well as SQL*NET access between ORACLE databases. The OPEN 7 communications facilities are shown in figure 0-3.

TCP (Transmission Control Protocol) is a connection oriented protocol which runs over IP. It is the most commonly used protocol since it offers a number of user services.

UDP (User Datagram Protocol) is a connectionless protocol which runs over IP. It is faster than TCP but it offers fewer services. It is used specifically by the **nfs** protocol where speed is important.

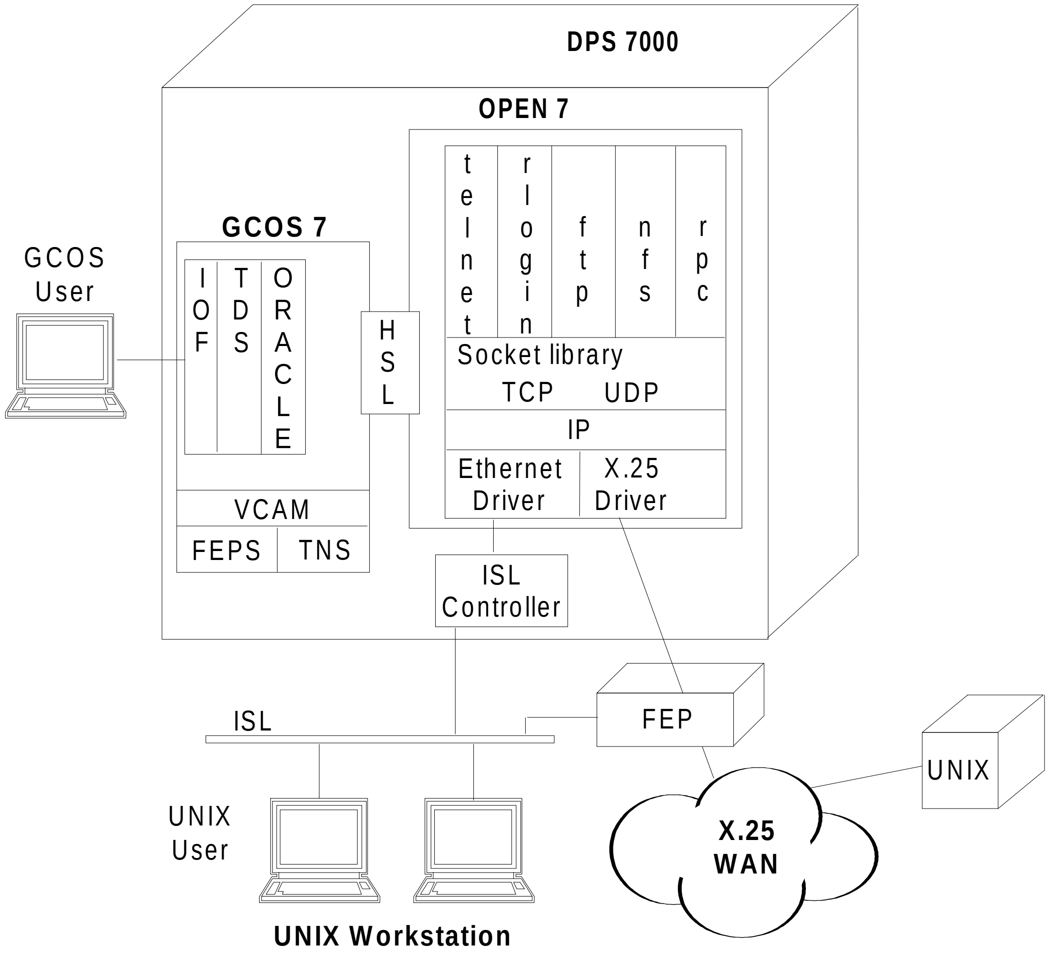

Figure 0-3. OPEN 7 communications*Error! Reference source not found.*

In a typical configuration using OPEN 7, a number of UNIX workstations connect to a GCOS 7 system via a LAN. The workstations use the large storage capacity of the DPS 7000 for storing user files. They access them with **ftp** or **nfs**.

A user at a UNIX system can logon to IOF or TDS under GCOS 7 using either **telnet** or **rlogin** or X-Window.

A user at a GCOS 7 system under IOF can start **ftp** file transfers between GCOS 7 and UNIX systems using GCL commands.

Access to OPEN 7 itself is reserved for the system administrator and concerns mainly file management under OPEN 7 or communications configuration for TCP/IP. The OPEN 7 administrator can use **rcp** to transfer files between UNIX systems and OPEN 7.

3.1.4 Remote operation facilities

Remote operation facilities allow a user at one system to use facilities provided by another remote system or to start jobs on a remote system. These facilities are of several types:

- IOF Pass Through allows a user working under IOF to logon to a remote DPS 7000 system and to use this system as if it were the local system.
- DJP (Distributed Job Processing) allows a user under IOF to request the execution of a job file on a remote system, using a GCL command.
- UFT (Unified File Transfer) allows a user under IOF to request the transfer of a file between any two ISO/DSA primary network systems (local-to-remote or remote-to-remote).
- RBF6 (Remote Batch Facility) allows a user on a DPS 6 to request the execution of a job file on a remote DPS 7000 system,
- FTAM7 is used for file transfers between systems in an OSI (Open Systems Interconnection) environment.
- DOF7 (Distributed Operator Facility 7) is a set of products which provide automated administration of a network of DPS 7000 systems.
- XCP1 and CPI-C/XCP2 (eXtended Co-operative Protocol) allow a user to implement distributed processing in a transactional environment.

3.1.5 File and catalog management

GCOS 7 offers a wide range of file management tools. The basic access method is UFAS (Unified File Access System) which handles sequential, relative and indexed sequential file organisations.

GAC (General Access Control) can be used in situations where many users access the same files, to prevent conflicts and inconsistency of data.

The catalog provides a flexible means of setting up file hierarchies and system and data access security. The Library Maintenance Processor provides program preparation and library facilities.

A Journal system provides both physical and logical file security and recovery in the event of system failures.

More sophisticated data structures can be created with databases (for example: IDS/II).

3.1.6 Database management

GCOS 7 supports both navigational and relational type databases.

The navigational database is IDS/II (Integrated Data Store II) which uses a classic hierarchical/network structure which conforms to the CODASYL standards. It is accessed and updated using batch and transactional programs and IQS (Integrated Query System - now called Relational Information System). IQS offers both fourth generation command language and a procedural language (Query) for creating stored procedures. IQS can also be used with UFAS files unrelated to IDS/II.

The relational database is ORACLE which has its own interactive facilities (SQL - Structured Query Language) as well as the pre compilers (PRO*COBOL and others) for procedural applications. GCOS 7 ORACLE can import and export IDS/II files.

An ORACLE database can be simultaneously accessed by IOF, batch jobs, transaction processing applications, and by remote ORACLE applications using SQL*NET.

3.2 ADMINISTRATION

The main responsibilities of a GCOS 7 system administrator are:

- system installation,
- file management,
- operator management,
- user management,
- system tuning,
- account management.

GCOS 7 provides tools for all these phases. They are grouped under the general term of system support facilities.

IUF (Installation and Updating Facility) is used to install the GCOS 7 software and update it when necessary.

VMM (Virtual Memory Management) and ARM (Automatic Resource Management) are used for allocating memory, CPU time and devices. These tools help to optimise system operation.

SBR (System Behaviour Reporter) is a tuning tool which gives a picture of how efficiently a system is operating. It indicates where there may be bottlenecks or problem areas and gives general statistical information.

TILS (Transactional and Interactive Load Simulator) allows the administrator to simulate the effects on the system of varying transactional and interactive loads.

DOF7 (Distributed Operator Facility 7) allows the automation of operator level administration of a network of DPS 7000 systems.

GCOS 7 also handles redundant, multi-processor configurations which ensure maximum availability and automatic reconfiguration, if necessary, with no interruption of normal operations.

3.3 LIMITATIONS, RESTRICTIONS

The following limitations apply to RPC running on a GCOS 7 system:

- The maximum number of sockets which may be opened simultaneously is 160.
- The maximum number of application servers with which a GCOS 7 process may interface is 10.
- RPC SIMU Regulation should be used with TDS applications to limit the use of sockets with this type of application. This product will be available with a later release of GCOS 7.
- The maximum "request for dynamic storage allocation" from a TDS application is 65487 bytes.
- The maximum number of vacant entries (refer to $H_SGCR of type 3 segments) is limited by **set** of the TDS "requests for dynamic storage allocation".
- The maximum file size handled by **rpctool** (see *DCE RPC Ally Administrative Supplement*) is defined as follows:
  - 256 source declaration lines (apart from functions),
  - 256 functions declared in a given source file,
  - 256 characters per source line.

4. GCOS 8 Overview

4.1 IMPLEMENTATION OUTLINE

GCOS 8 is the operating system for Bull's DPS 90, DPS 8000 and DPS 9000 computer systems (generically called the Bull DPS 8 family). It is a multidimensional software which controls the data processing activities within these machines. GCOS 8 can handle up to 2000 simultaneous user or system processes. It controls the allocation of system resources to optimise the use of memory space, CPU time, file access, etc.

The Bull DPS 9000 is a multiprocessor system. Up to four processors can operate simultaneously to efficiently handle heavy workloads. In a multiprocessor environment, each processor can execute different processes in parallel, and each processor can directly access memory. This facility is particularly effective when many processes in memory require time-sliced execution.

Within Bull's DCM, GCOS 8 is positioned as a mission critical, high volume TP/DB (Transaction Processor/Data Base) Server. It supports Bull's most powerful and sophisticated transaction processor, the TP8 and industry sources consider Bull's relational database controller as the "next generation" of data processing capability.

GCOS 8 is recognised by its users as the Enterprise TP/DB Server with long proved performance and robustness. The reliability of the system together with its security features make it the ideal system for storing and processing strategic corporate data.

GCOS 8 offers the following main features:

- file management,
- data management,
- time sharing system,
- transaction processing,
- local and remote batch operations,
- access to telecommunications networks.

- remote job entry,
- file transfer functions
- on-line tests and diagnostics

In addition, the different processing environments (batch, time sharing, transactional) can operate simultaneously.

4.1.1 File management

The File Management Supervisor (FMS) is a flexible file management system which responds to users needs while ensuring integrity of user data base files. It is responsible for cataloging files, allocating file space and controlling user access to files. The catalog structure includes chains of sub-catalogs with:

- one or more files under each catalog,
- files directly catalogued under a User Master Catalog (UMC), and
- access controls at all levels of the structure.

This organisation makes it easy to group files and sub-catalogs by user names, access restrictions, and resource control. Within this hierarchy, an individualised structure of catalogs, sub-catalogs, and files can be created to meet the particular needs of the organisation.

FMS also accommodates the wide variety of storage devices available in the GCOS 8 environment. Users may request file space on a specific device or FMS may assign it automatically. The user can define the files logically with a user name or physically in terms of a specific device or device name. In addition FMS permits the allocation of access rights by the creator of the file.

4.1.2 Time sharing

The Time Sharing System (TSS) is a multi-user, interactive facility which provides remote terminal users with data processing services. It meets the needs of the whole spectrum of possible users, ranging from the specialist software programmer through to the business manager, accountant and other non-expert users.

TSS provides the following features:

- access to functions such as programming, language processors, text processing, data base access and batch processing,
- a comprehensive, simple-to-use command language,
- access to the file system,
- data management facilities to access databases,

- a user friendly terminal interface,
- a time sharing library of application programs.

The TSS comprises the Time Sharing Executive (TSE) and a number of individual subsystems which support the various functions offered. The TSS subsystems and any user programs operate under the control of the TSE which allocates resources as required.

The TSE provides an interface between the user programs and the GCOS 8 system to obtain system services and resources required. Thus a batch job, submitted from a user's terminal, is passed to the GCOS 8 system by the Time Sharing Executive. The job is then processed in the same way as any other batch job except that the output may be returned to the user's terminal via the TSE.

4.1.3 Transaction processing

The Transaction Processor 8 (TP8) is a versatile, large volume transaction processor which can provide processed information when and where it is required and in the form specific to the needs of the user. It is the most powerful and sophisticated transaction processor offered by Bull.

The main purpose of TP8 is to update and manipulate data records dynamically within a user application. In this way, the user always has access to current data.

TP8 also processes off-line batch processing operations but its real strength lies in its ability to concurrently process large quantities of event-driven transactions initiated on-line at users' terminals.

4.1.4 Data management

The Data Management-IV (DM-IV) is responsible for concurrent access to common, shared network IDS/II data bases in TP and batch modes. It provides the user with data security, automatic restart/recovery (via FMS), a single Data description language (DDL) and numerous utility functions.

Associated with DM-IV are a number of facilities used for managing or retrieving data. These include INTEREL and the query facilities INTEREL-SQL and FormsSQL. The query facilities are interactive tools for retrieving data using a relational view from network IDS/II files. In addition they display data as relational tables, without restructuring or replicating the data base.

4.1.4.1 INTEREL facility

INTEREL is an integrated relational data management system. It provides query and update facilities through SQL-related (Structured Query Language) products, query facilities through DM-IV, and data definition directory services through the Operational Directory Interface (ODI). ODI is the relational file directory that INTEREL uses to maintain logical model definitions.

4.1.4.2 INTEREL-SQL

INTEREL-SQL is a complete relational language which can dynamically create and delete models, tables and indexes. It uses data created by the user or extracted from structured files. It can be used as an interactive interface with relational databases residing on Bull systems or on other relational database computers. It comprises a set of commands that tell the computer what processing actions should be performed on selected data.

4.1.4.3 FormsSQL

FormsSQL provides automatic construction of the SQL queries for managing data in a relational database. The user interface uses a form-oriented environment so it is better adapted to novice users than is INTEREL-SQL. Users do not need specific knowledge of SQL syntax or of database technology and in addition the forms can be designed by the system administrator to suit a given environment.

4.1.5 Networking

A distributed processing system comprises a number of separate computer systems all working in a co-operative way. In a distributed system, processing and storage facilities are geographically dispersed but connected by transmission media. A vital element in this type of environment is thus the communications network linking the different machines.

Any two systems can be interconnected as long as they use the same communications protocol.

GCOS 8 implements Bull's Distributed Systems Architecture (DSA) which is based on the OSI reference model. This architecture supports communications with any other DSA system.

GCOS 8 currently has access to the UNIX world via the Open 8 system. Open 8 is a gateway software running on a Bull DPX2 which also provides access to TCP/IP network. The users of GCOS 8 can also access the UNIX world via Mainway 2000 with FCP8 on the network FDDI.

4.1.5.1 DSA communications

DSA provides access to other DSA systems which include DPS 8, DPS 7 and DPS 6. GCOS 8 accesses a DSA network via the DN8 front end processor (FEP) running the DNS software (Distributed Network Supervisor) with CXI protocol or DNET/ROUT protocol. The DN8 is connected to the Direct Interface (DI) channel using the Common Exchange Interface (CXI) protocol. The Direct Interface Attachment board (DIA) on the DN8 provides the connection to the GCOS 8 DI channel. Also, access can be made via FCP8 (Rev01) on FDDI.

The DN8 supports public and private WANs (X.25, X.21 or HDLC) and LANs (Ethernet type) as shown in figure 0-1. These connections, which all use DSA, form the "primary network".

DSA communications (through both DNET/ROUT and CXI) are typically used by traditional GCOS 8 applications; TSS, TP8 etc. The DN8 also supports gateway software to provide access from the DSA world to SNA networks (IBM's network architecture) and OSI networks, thus providing access to IBM mainframes or systems supporting the OSI reference model.

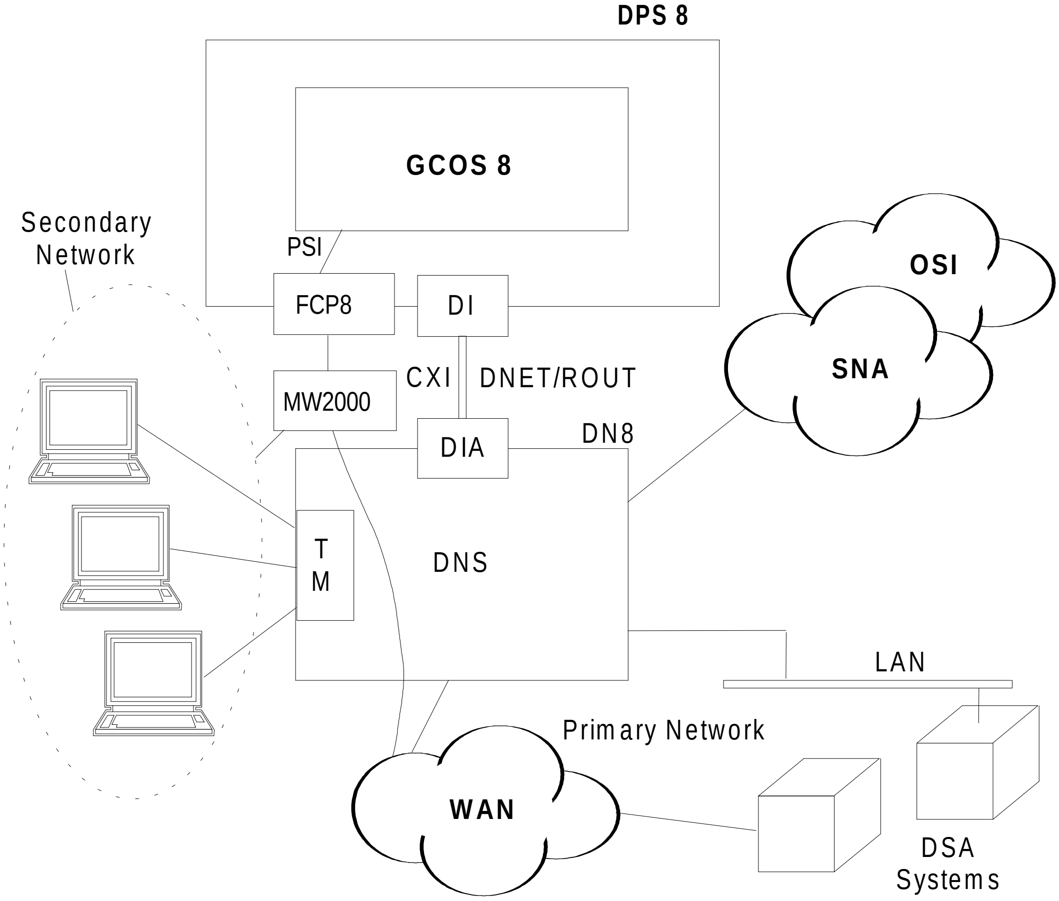

Figure 0-1. GCOS 8 - DSA Communications*Error! Reference source not found.*

Terminals connect to GCOS 8 through the DN8 or MW 2000 (Rev 0.1). They are managed by a software component called the Terminal Manager. The terminals connected to the same GCOS 8 machine are called the secondary network. These terminals can access applications on the GCOS 8 host connected directly to the DN8 or on any other DSA system in the network.

Traditionally VIP7800 terminals are used to connect to GCOS 8 applications. IBM compatible PCs running emulators (Affinity for example) can also be connected directly as terminals. See also section "Open 8 Terminal connections".

4.1.5.2 TCP/IP communications

TCP/IP provides access in general to TCP/IP workgroups and specifically to UNIX servers such as the Bull DPX2 and Bull DPX20. GCOS 8 accesses a TCP/IP network via an Open 8 system or an FCP8. Open 8 is a gateway software running on a Bull DPX2 (Rev 0.1).

TCP/IP access from GCOS 8 is shown in figure 0-2. Open 8 may be connected locally as a front end processor or remotely via an ISO/DSA network. In this latter case, GCOS 8 uses a DN8 as FEP. TCP/IP communications are typically used by UNIX oriented applications: FTP8, SMTP8, Oracle SQL*Net.

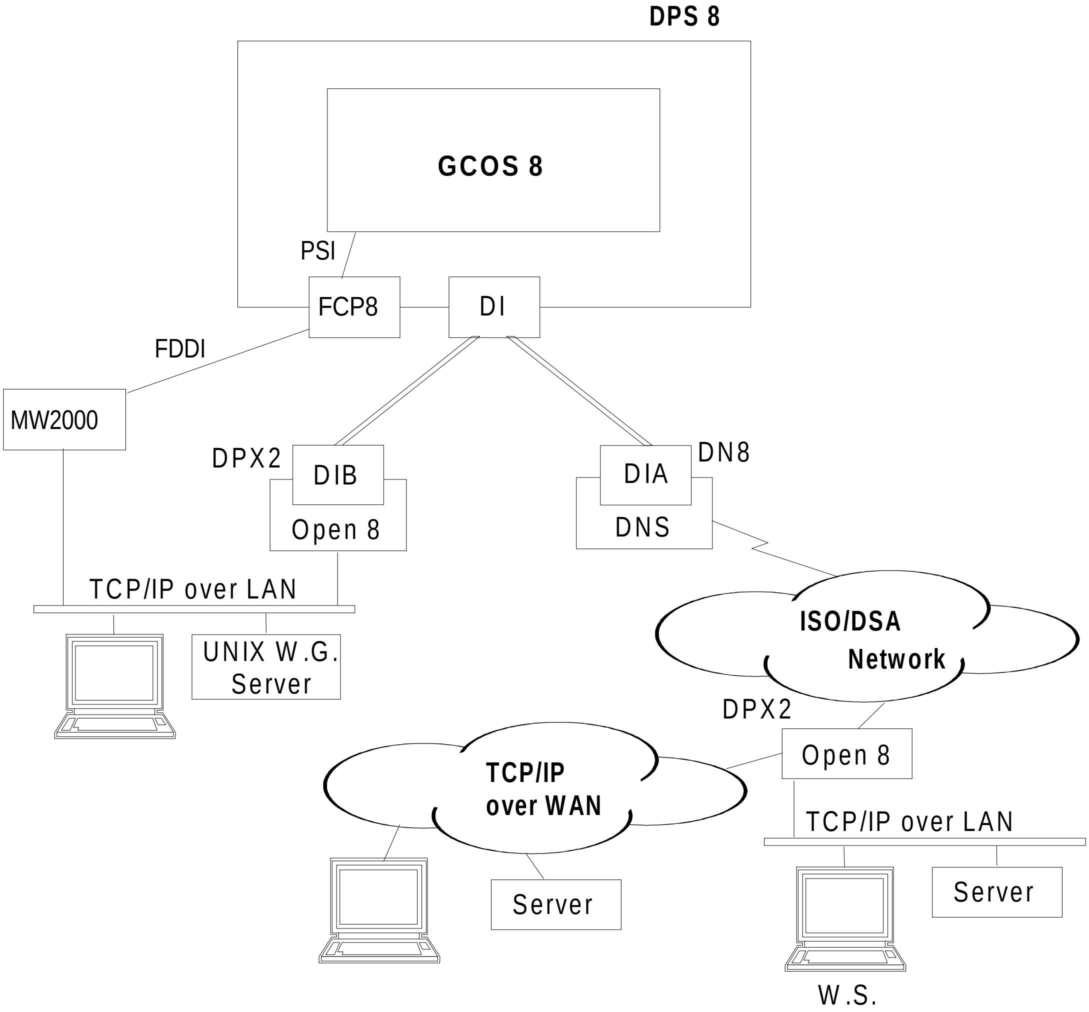

Figure 0-2. GCOS 8 - TCP/IP Communications*Error! Reference source not found.*

4.2 THE OPEN 8 SYSTEM

The Open 8 system is a UNIX system (Bull DPX2) which provides access to the TCP/IP world and in particular to other UNIX systems. The Bull DPX2/200 is used as a Communications Controller (Open 8/CC) which provides communications support only for GCOS 8.

4.2.1 Open 8 Architecture

Figure 0-3 shows the hardware and software architecture of the Open 8 system. Note that not all components are available on all models of the Bull DPX2 family.

Hardware

The Basic Ethernet board (BETH) provides connections to systems on the TCP/IP LAN. The Fast Extended Communications Processor board (FECP) provides primary network connections via LAN or WAN to the DN8. It can also be used for connections over the TCP/IP LAN. Note that the TCP/IP LAN and the OSI/LAN may use the same physical wires

The Direct Interface Board (DIB) provides the direct connection to the GCOS 8 DI channel. This interface uses the CXI protocol.

The DIB is the only hardware component which is specific to Open 8. The others are part of the standard DPX2 catalog.

Software

The GCOS 8-Link provides TCP/IP and UDP/IP communications services for GCOS 8. It contains the External TCP/IP Socket Agent (XTSA) which performs socket functions on behalf of GCOS 8 applications. It also provides administration and maintenance tools to support the Socket Agent.

The Socket Agent can be used to communicate with GCOS 8 either via the Direct Interface channel or via a primary network link and a DN8 relay system.

The GCOS 8-Link is the only software component which is specific to Open 8. The other software components are part of the standard DPX2 catalog. They are:

- the operating system: a runtime version of B.O.S. 2 which is a UNIX system based on AT&T's System V Release 3.1.
- INET Internet package: this contains the socket library with TCP/IP protocol, Telnet (see section "Open 8 Terminal Connections"), FTP the UNIX file transfer protocol,
- TPAD-HPAD software: provides asynchronous terminals with access to a WAN or a LAN (see section "Open 8 Terminal Connections").

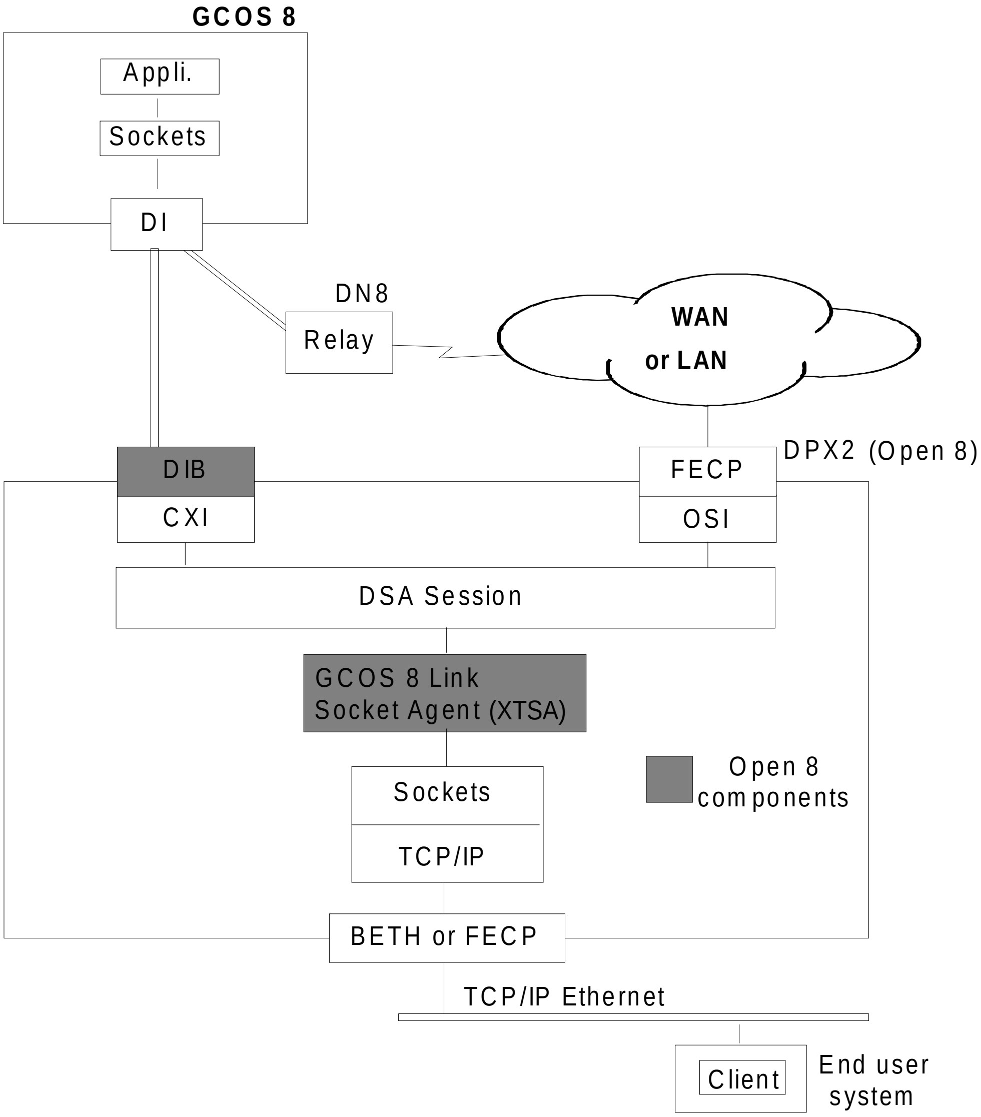

Figure 0-3. Open 8 Architecture*Error! Reference source not found.*

4.2.2 Open 8 Terminal Connectability

Figure 0-4 shows the connection of terminals though the Open 8 system. Terminal connections use Telnet. Telnet is a standard UNIX facility which provides virtual terminal management, allowing PCs and workstations to access any machine in the TCP/IP network. In the Open 8 environment, Telnet is used to provide access for DSA terminals to the TCP/IP world and from terminals in the TCP/IP network to DSA correspondents.

TCP/IP to DSA

End users in the TCP/IP network are connected via their local UNIX system to the Open 8 system over the TCP/IP LAN. The connections from these terminals are managed by the Server Telnet facility. The DPX2 uses the TPAD facility to connect over the X.25 network to the Terminal manager of a DN8. The TPAD is also used to connect terminals local to the DPX2.

Once connected to the DN8, users have access to any system in the DSA network, either the local DPS 8 or remote correspondents elsewhere in the network. Note that at each stage of the connection from the end user's terminal, the user must execute separate login procedures

Terminals connected in the TCP/IP network are generally managed as asynchronous terminals. VIP7800 terminal emulation can be used on Open 8/AP systems (for example the VIPEMU product) for connection to traditional applications on GCOS 8 (TSS, TP8, etc.).

DSA to TCP/IP

DSA terminals include terminals connected to any DSA terminal manager, such as exists on the DN8 and DN713x Datanets or on the CNP7 of GCOS 7 systems or on DSA6 on GCOS 6 systems.

Open 8 provides access for these terminals with the HPAD facility and the User telnet via an X.25 primary network connection. Connections to the TCP/IP network are then made with the standard sockets and TCP/IP protocols of the DPX2.

At each step of the connection process, the user must perform separate logon operations (at the DSA terminal manager, at User Telnet, and at the destination system). The terminal user can connect to any system in the TCP/IP network on which she has an account. She can also connect, via the HPAD, to local UNIX applications running on an Open 8/AP system.

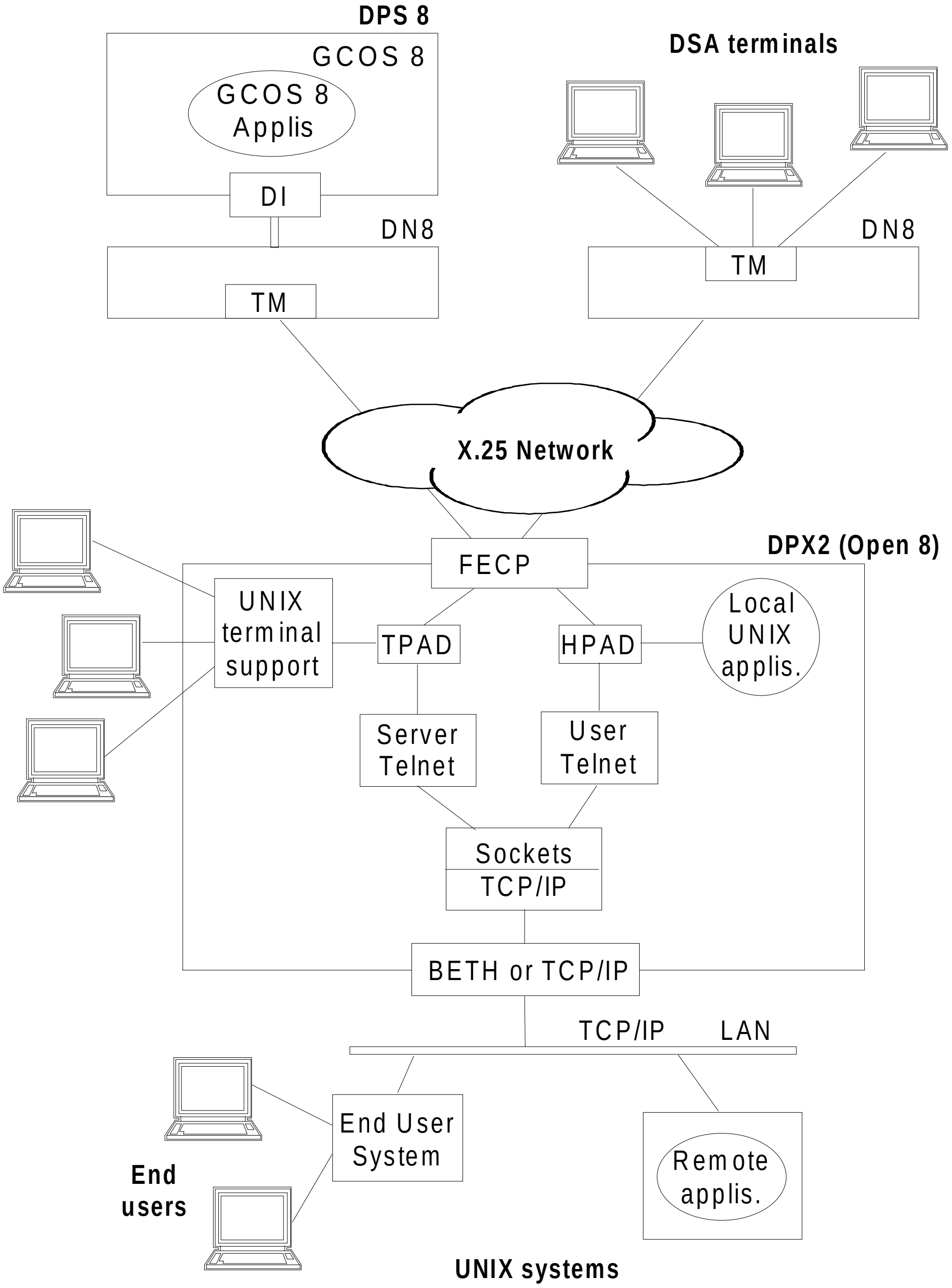

Figure 0-4. Open 8 Terminal connections*Error! Reference source not found.*

4.3 CONNECTION WITH FCP8

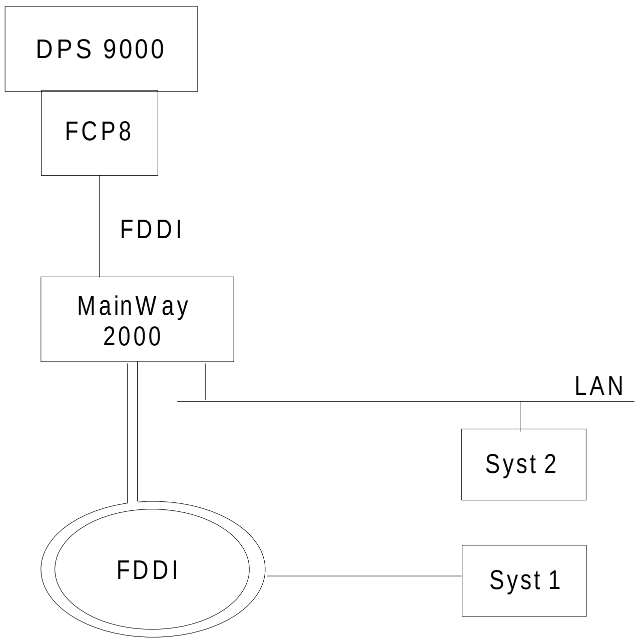

Figure 4-5. Connection with FCP8 *Error! Reference source not found.*

The users of Syst1 or Syst2 are able to access the central DPS 9000 with the communications networks TCP or ISO/DSA.

The link between the DPS 9000 and the Mainway 2000 is crossed by an FDDI cable of 30m and a simple attachment. This function is implemented on the DPS 9000 series server from GCOS 8 40204 onwards.

4.4 ADMINISTRATION

For information on GCOS 8 administration, refer to the *GCOS 8 Software Installation Bulletin* and the *GCOS 8 Software Release Bulletin*.

5. GCOS and DCE

On the one hand, the GCOS 7 and GCOS 8 software platforms perform high volume server functions extremely well. They are well suited to handling data retrieval functions for today's applications and also for future requirements.

On the other hand, the availability of new applications on standard UNIX platforms makes these platforms attractive for new system implementation.

GCOS DCE offers a mechanism to join these two disparate environments in a seamless manner.

5.1 GCOS-UNIX COMMUNICATIONS

The term "GCOS-UNIX communications" refers to the hardware connections and the lower-layer protocols used to connect GCOS 7 and GCOS 8 to UNIX systems. These connections are currently achieved with the OPEN 7 and Open 8 products. The communications protocols (TCP/IP and UDP/IP) are implemented in OPEN 7 and Open 8. Figure 0-1 shows the architecture.

GCOS 7 communicates with the UNIX world via OPEN 7. This is a UNIX system implemented as a subsystem of GCOS 7 on the DPS 7000. GCOS 7 communicates with OPEN 7 via the HSL (High Speed Link) shared memory. An ISL controller card is used to connect the DPS 7000 to the LAN cable.

GCOS 8 uses Open 8 to communicate with the UNIX environment. Open 8 is implemented on an independent front-end UNIX system. It connects to GCOS 8 via a DIB board connected to the DI channel on the DPS 8.

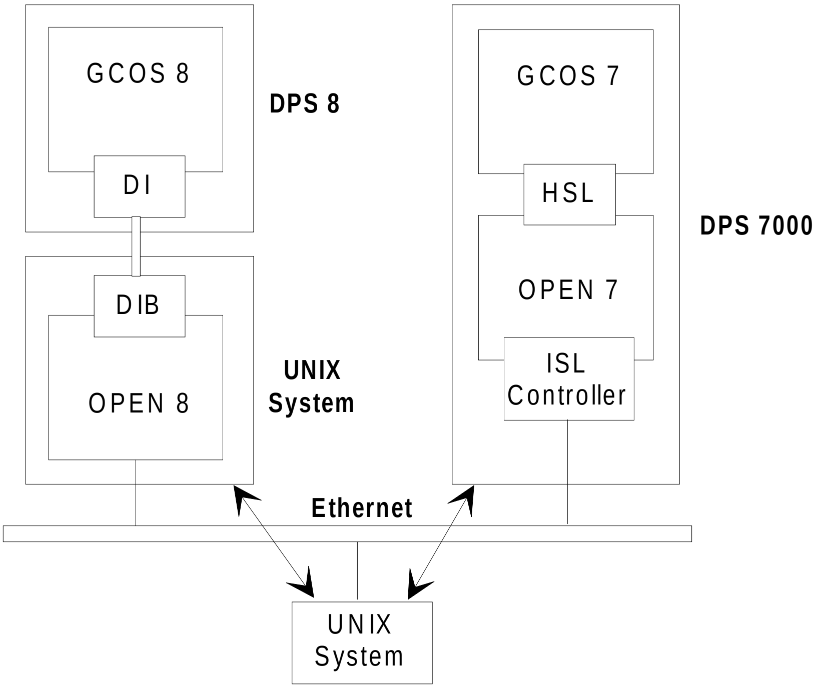

Figure 0-1. Open 8 and OPEN 7 communications*Error! Reference source not found.*

5.2 GCOS DCE ARCHITECTURE AND FUNCTIONS

GCOS DCE comprises two software components:

- DCE RPC on GCOS (7 or 8) installed on GCOS ,
- DCE RPC Ally installed on a UNIX system.

Figure 0-2 illustrates the architecture of the *GCOS DCE* offer. In order to reduce the work in the GCOS environment, only the minimum necessary of the DCE functions are ported onto the GCOS system (specifically the RPC Runtime library). This is *RPC on GCOS*.

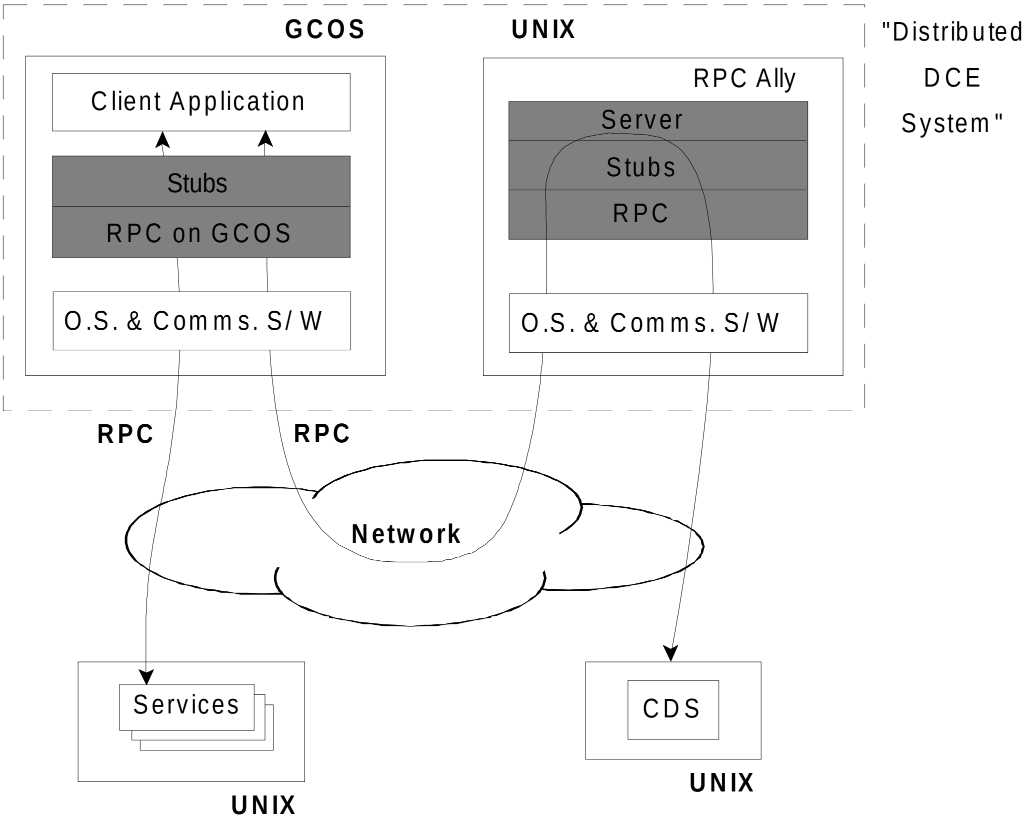

Figure 0-2. GCOS DCE Architecture*Error! Reference source not found.*

All other functions are ported as a UNIX daemon onto another system. The UNIX daemon is the DCE RPC Ally. The GCOS system together with the Ally operate as a distributed DCE system. In this way, the RPC service offered on the GCOS system is a full OSF DCE service although certain functions are implemented on another machine.

5.3 THE DCE RPC ALLY

5.3.1 Functions

Figure 0-3 shows the function of the Ally in the Distributed Computing Environment. It provides access to the DCE basic services for the GCOS client. In this initial release, GCOS DCE only supports the CDS (Cell Directory Service).

The fact that this service is accessed via the Ally is not visible to the client application or to the application developer.

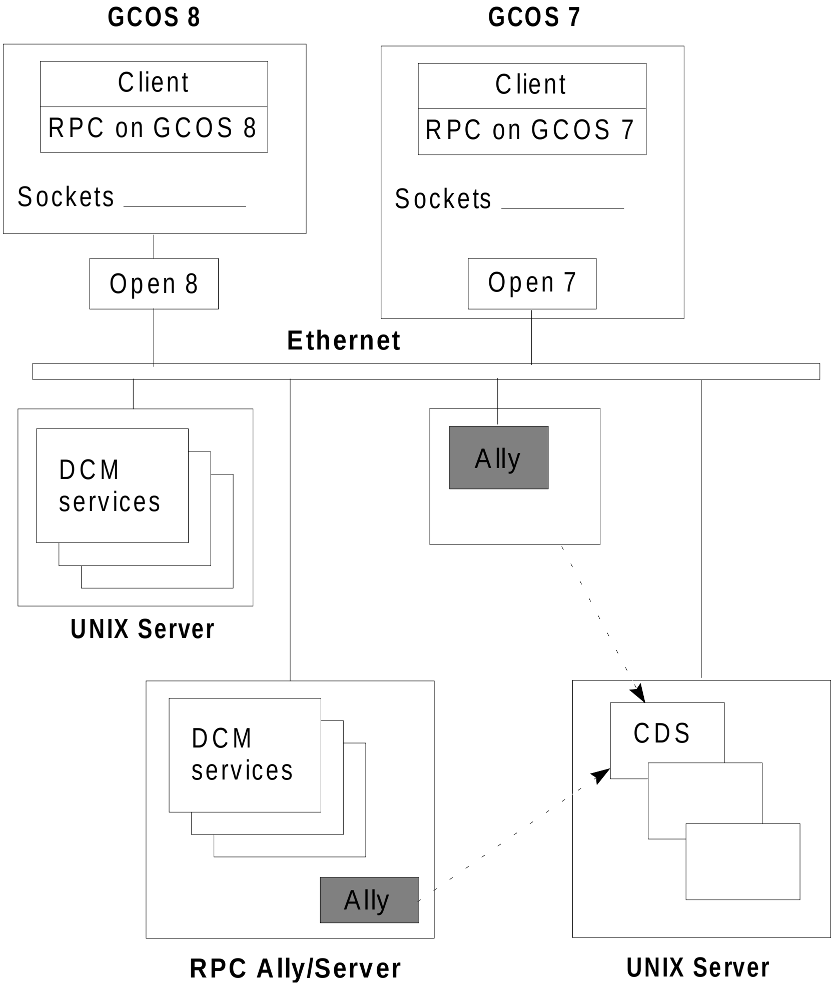

Figure 0-3. The Ally Functions*Error! Reference source not found.*

5.3.2 Location

The Ally introduces a large degree of flexibility into the GCOS DCE architecture. It may be implemented in any Bull DPX20 system in the network.

A GCOS client may interact directly with a UNIX server once the connection has been established. This means that, if the client does not use CDS, RPC messages will be sent directly from the GCOS client to the DCE application server. The server may be any vendor's system which implements a standard OSF DCE server function.

5.3.3 Bindings

When a client sends an RPC to a server, it must first locate the server. This process is called binding.

The client may have a direct way of locating the server. The server's address may be hard-coded into the program or it may be stored in a file.

The client can also find a server by asking the DCE Directory Service for the location of that server.

RPC on GCOS uses separate bindings for accessing Ally related services and for accessing other distributed services.

5.3.3.1 Bindings to the Ally

Each client on the GCOS system may access the DCE basic services through the Ally via a binding to the Ally. Each client uses a separate binding. But it can access several services via this binding. In order for bindings to the Ally system to be feasible, two conditions must be met:

- The client must know the location of the Ally because the client cannot access the Directory Service before accessing the Ally. This information must be obtained from configuration data at start-up of the client system.

- The link between the client system and the Ally system must be secure because any future security functions will not be active until the client has accessed the Ally. Thus the Ally must be installed on a trusted system connected via an exclusive communications path. An exclusive path is one on which only the GCOS client and the Ally can send, receive or modify data. If the security requirements are sufficiently high, this path may be a dedicated communications line between GCOS and the Ally.

5.3.3.2 Bindings to other services

Bindings from the client may be to the Ally or directly to the required service elsewhere in the network. If the binding is to the Ally, it is subject to the conditions listed above. If it is to another server, it is not subject to any particular security conditions.

On the initial call from a client, RPC may access the Ally in order to use the Directory Service. Once the location of the remote server is known to the client, the client can communicate via a binding directly to the server.

5.4 RPC ON GCOS 7 - ARCHITECTURE

In the present product, both client on UDP or TCP/IP and server on UDP/IP are implemented in GCOS 7. This function is DCE RPC on GCOS 7. Its architecture is shown in figure 0-4.

RPC on GCOS 7 uses the communications software of OPEN 7 to communicate with the network. The communications protocol used for GCOS-UNIX communications is RPC/DG (Datagram) over UDP/IP.

RPC runtime on GCOS is the standard RPC runtime package, adapted to run in a GCOS 7 environment. Its interface towards the network uses sockets. A pseudo socket adapter provides the interface with the HSL. At the network side of HSL a socket server provides the interface to the standard socket library of OPEN 7.

The RPC runtime on GCOS is also provided with pre-configured client stubs which provide access to the basic DCE services via the Ally.

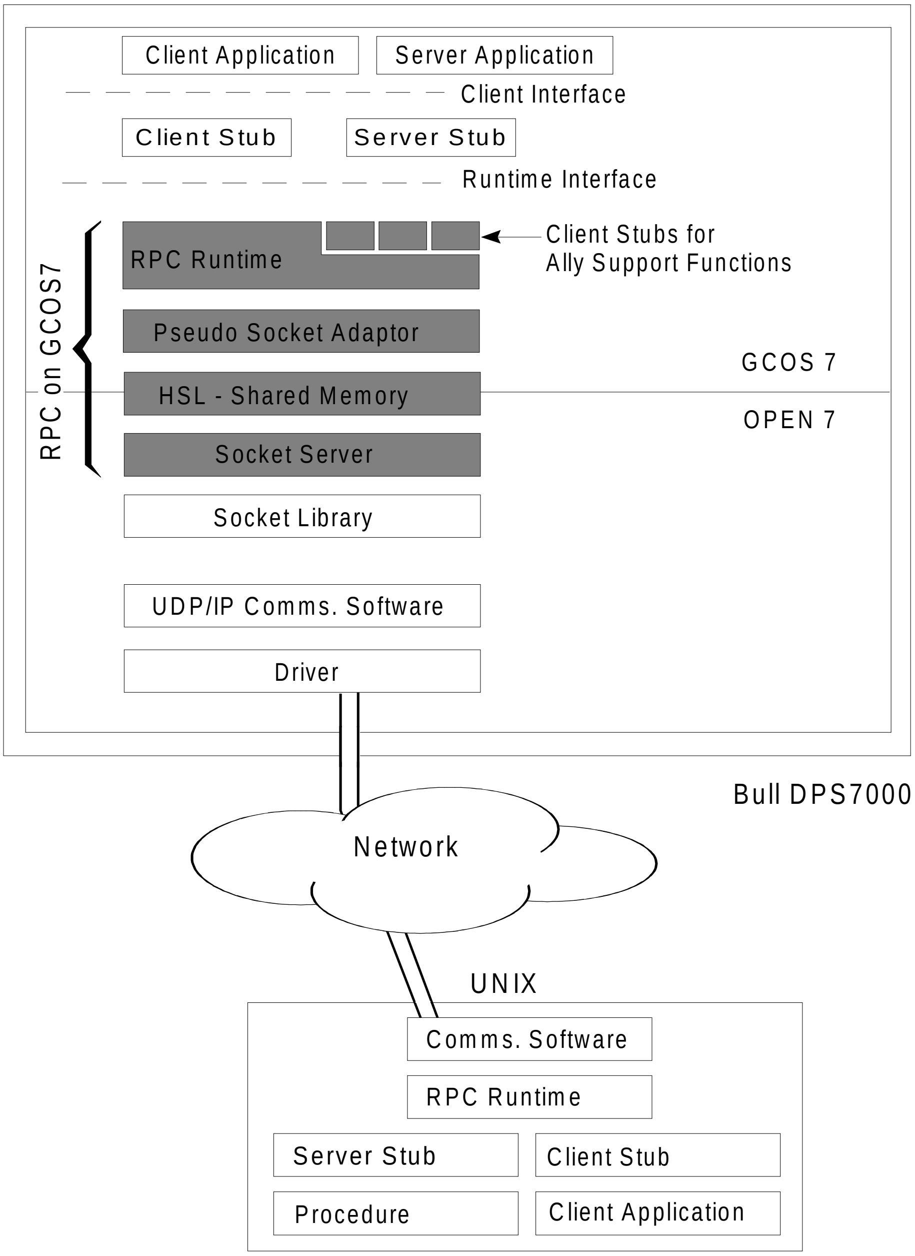

Figure 0-4. RPC on GCOS 7*Error! Reference source not found.*

5.5 RPC ON GCOS 8 - ARCHITECTURE

In the present product, a client only RPC function is implemented in GCOS 8. This function is DCE RPC on GCOS 8. Its architecture is shown in figure 0-5.

RPC on GCOS 8 uses the communications software of Open 8 to communicate with the TCP/IP network. The communications protocol used is RPC/DG (Datagram) over UDP/IP.

RPC runtime on GCOS is the standard RPC runtime package, adapted to run in a GCOS 8 environment. Its interface towards the Open 8 system uses the DI channel of the DPS 8. In the Open 8 system, the GCOS 8-Link provides communications services for GCOS 8 and provides the interface to the Open 8 sockets software. In addition, RPC on GCOS 8 runs over a GCOS 8-Link relay connection.

The RPC runtime on GCOS is also provided with pre-configured client stubs which provide access to the basic DCE services via the Ally.

5.6 FUNCTIONS SUPPORTED AND RESTRICTIONS

The following is a summary of the RPC services supported by GCOS DCE. For full details refer to the product specific documentation delivered with the software:

- CDS (accessed via the Ally),
- Client functions (those functions marked "Used by client applications" in the *DCE Application Development Reference*).

GCOS does not support:

- DTS The DTS maintains a standard time for all DCE systems.
- The Kerberos security service. However, the Ally system does support this service because it accesses the CDS which is a secure service. This is not visible from the GCOS system.
- Threads for client and server applications,
- DFS, GDS, XOM, XDS and "Diskless" services.

The communications protocol used for GCOS-UNIX communications is RPC/DG (Datagram) over UDP/IP and not RPC/CN (connection oriented) over TCP/IP.

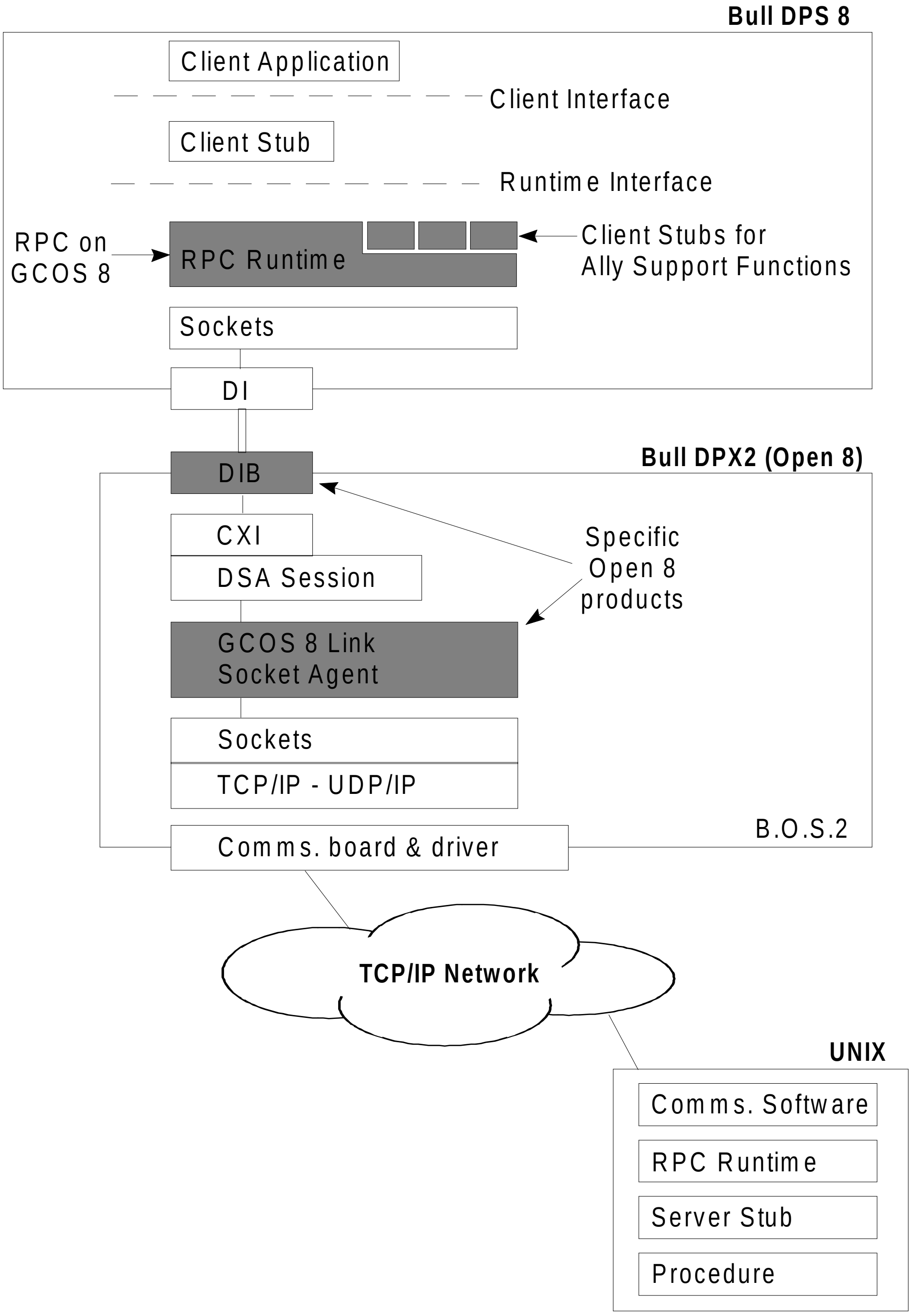

Figure 0-5. RPC on GCOS 8*Error! Reference source not found.*

Glossary

Adaptation code
A body of code that exists between the client and the client stub in Bull's use of the DCE RPC. This code handles the following tasks: (1) locating and binding to the server (except in cases where the client stub does automatic implicit binding), (2) handling of communication failure errors, (3) data marshalling for parameters too complicated for the stub to marshal, (4) passing global variables, and possibly some (5) security, and (6) memory management. Another body of adaptation code exists between the server stub and the server for performing complementary tasks.

AIX
The IBM implementation of the UNIX operating system. It is the operating system for the Bull DPX/20.

Ally
The concept of using a UNIX-based system to provide specialised services to GCOS. Eventually, all ally functions will be satisfied by the DPX/20. Initially the Ally is used for CDS (the directory service).

ANSI
American National Standards Institute. A standard-setting body in the United States of America.

API
Application Programmatic Interface. A library of routines provided for accessing a local service. When using RPC, an API can also access a remote service.

ARM
Automatic Resource Management. A GCOS 7 administration tool.

ASO
Automatic System Operation. A set of products on GCOS 7 and GCOS 8 for job management, and system monitoring and management.

at-most-once semantics
RPC: A characteristic of a procedure that restricts it to executing once, partially, or not at all - never more than once. See also **idempotent semantics**, **broadcast semantics**, **maybe semantics**.

attribute
(1) Threads: The individual components of the attributes object. Attributes specify detailed properties about the objects to be created. (2) RPC: (a) IDL or ACF syntax element, occurring within square brackets, [], and conveying information about an interface, type, field, parameter or operation. (b) An attribute of an entry in a name service database that stores binding, group, object or profile information for an RPC application and identifies the entry as an RPC server entry; an NSI attribute.

b
In lower case, refers to a bit. See “B”.

B
In upper case, refers to a byte. See “b”.

big endian
An attribute of data representation that reflects how multi-octet data are stored in memory. In big endian representation, the lowest-addressed octet of a multi-octet data item is the most significant. See also **endian**, **little endian**.

Binding
Information that identifies an RPC server and its interface to one of its current clients.

Binding handle
A reference to binding information that identifies one binding.

blocking call
A call in which a caller is suspended until a called procedure completes.

BOS2
Bull Open Software 2. The operating system for the DPX/2.

BOSX
Bull Open Software for UNIX. The original operating system for the Bull DPX/20. It has been replaced by AIX.

broadcast semantics
RPC: A form of idempotent semantics that indicates that an operation is always broadcast to all host systems on the local network, rather than delivered to a specific system. An operation with broadcast semantics is implicitly idempotent. Broadcast semantics are supported only by connectionless protocols. See also **at-most-once semantics**, **idempotent semantics**, **maybe semantics**.

BSD
Berkeley Software Distribution. UNIX software produced at the University of California at Berkeley.

C
The "C" programming language.

CASE
Computer Aided Software Engineering

CDS
Cell Directory Service. The name service for a OSF DCE cell.

CDS cache
The information that a CDS clerk stores locally to optimise name lookups. The cache contains attribute values resulting from previous lookups, as well as information about other clearinghouses and namespaces. The cache is written to disk periodically so that it can survive a system reboot.

Cell
A set of networks and systems managed as a group. Also known as an administrative domain.

clearinghouse
A collection of directory replicas on one CDS server. A clearinghouse takes the form of a database file. It can exist only on a CDS server node; it cannot exist on a node running only CDS clerk software. Usually only one clearinghouse exists on a server node, but there may be special cases where more than one exists.

clerk
(1) CDS: The software that provides an interface between client applications and CDS servers. The clerk receives a request from an application, sends the request to a CDS server, and returns any resulting information to the application. The clerk saves (caches) the results of lookups so that it does not have to repeatedly go to a CDS server for the same information. (2) DTS: A software component that synchronises the clock for its client system by requesting time values from servers, computing a new time from the values, and supplying the computed time to client applications.

Client
A program or machine that requests services.

client context
RPC: State in an RPC server's address space generated by a set of remote procedures (manager) and maintained across a series of calls for a particular client. See also **manager**, **context handle**.

client/server
A mode of processing in which client systems request and receive services from server systems.

context Handle
A client data structure (supplied by the user) which is used to identify information (context) in an RPC server regarding a particular RPC binding. See also **client context**.

Coordinated Universal Time (UTC)
An international time standard that DTS uses. The zero hour of Coordinated Universal Time is based on the zero hour of Greenwich Mean Time.

CPI-C
Common Programming Interface - Communications. A programming interface which allows X-Open compliant systems to intercommunicate.

CTP
Co-operating Transaction Processing. A peer to peer mode of communicating between two transaction applications.

datagram
(1) A type of socket that uses the UDP protocol. (2) An individual packet handled by IP.

DCE
Distributed Computing Environment. An architecture and set of technologies from OSF. It includes RPC, CDS, DTS, security etc. It is used by Bull as a building block for DCM.

DCE RPC on GCOS
See RPC on GCOS.

DCM
Distributed Computing Model. A computing model for transparent distribution of resources and services announced by Bull in March 1991.

DDA
Distributed Data Access. Bull's model for accessing databases on heterogeneous platforms.

DFS
Distributed File Service. An OSF DCE facility which allows users to access and store files stored anywhere in the network.

DI
Direct Interface channel. A high speed communications port in GCOS 8.

DIA
Direct Interface Attachment board. A communications board in the Bull Datanet which connects to the DI channel of GCOS 8.

DIB
Direct Interface Board. A communications board in the Bull DPX2 which connects to the DI channel of GCOS 8.

DJP
Distributed Job Processing. A GCOS 7 facility that allows a user to request the execution of a job on a remote system.

DMU
Data Management Utilities. A GCOS 7 service used to simplify file and data manipulation.

DNS
(1)Domain Name System. A name resolution mechanism for TCP/IP networks. (2)Distributed Network Supervisor, the operating system for Bull's Datanet (DN8 and DN7100) network processors.

DPF
Distributed Printing Facility. An OSF DCE facility which allows users to use the printing resources in the network.

DPS
Data Processing System. Bull's range of mainframe computers.

DPX/2
Bull's Motorola-based family of UNIX systems. A variety of models are offered.

DPX/20
Bull’s RISC-based family of UNIX systems. This family is based on RS/6000 RISC technology shared with IBM.

DSA
Distributed Systems Architecture. Bull's architecture for the design of communication hardware and software.

DTS
Distributed Time Service. A distribution service in the DCM for synchronising clocks of the systems on the network. Guarantees a clock will never go backwards.

EDI
Electronic Data Interchange

endian
An attribute of data representation that reflects how certain multi-octet data is stored in memory. See also **big endian**, **little endian**.

endpoint
RPC: An address of a specific server instance on a host.

endpoint map service
RPC: A service provided by the RPC daemon that maintains a systems endpoint map for local RPC servers. When an RPC client makes a remote procedure call using a partially bound binding handle, the endpoint map service looks up the endpoint of a compatible local server. See also **RPC daemon**.

Ethernet
A link layer (level 2) LAN protocol invented by DEC, Intel and Xerox that operates at 10 Mbps over coaxial or shielded twisted-pair cable.

export
To provide access to an interface (by responding to remote procedure calls). Servers export interfaces to clients. Also, to advertise, by registering binding information in the directory service, an interface as offered by a given server.

FEPS
Front End Processor Support. The GCOS 7 communications channel which provides the interface to a stand-alone network processor system (Datanet).

FTAM
File Transfer Access Method. A file transfer mechanism used in the OSI environment.

FTF6
File transfer Facility. It allows a DPS 6 user to request the transfer of a file to or from a DPS 7.

FTP
File Transfer Protocol. A file transfer mechanism used in the TCP/IP environment.

GAC
General Access Control. A GCOS 7 tool used to prevent conflict of access to the same file and inconsistency of data.

GCL
GCOS 7 Command Language. The command language used in the GCOS 7 IOF environment.

GCOS
General Comprehensive Operating System. When used without a numerical suffix, this term generally refers to GCOS 6, 7 or 8.

GCOS 7
The operating system for Bull's family of mainframe computers.

GCOS 8
The operating system for Bull's family of large proprietary mainframe computers.

GCOS 8-Link
Software on Open 8 that allows socket connections to GCOS 8 applications. It consists of the socket agent, the DIB driver and DSA session.

GDA
Global Directory Agent. Software which acts as a go between for the **CDS** and the **GDS**

GDS
Global Directory Service. The global name service for the whole network. It connects the local **CDS**s into one world-wide hierarchy.

handle
RPC: An opaque reference to information. See also **context handle**, **name service handle**.

Host
A computer system.

HSL
High Speed Link shared memory between GCOS 7 and OPEN 7.

IBM
International Business Machines corporation

idempotent semantics
RPC: A characteristic of a procedure in which executing it more than once with identical input always produces the same result, without any undesirable side effects; for example, a procedure that reads a particular block of an immutable file is idempotent. DCE RPC supports maybe and broadcast semantics as special forms of idempotent operations. See also **at-most-once semantics**, **broadcast semantics**, **maybe semantics**.

IDL
Interface Definition Language. A “C-like” language for describing interfaces between routines. Used as a tool for building Remote Procedure Calls. See “RPC” and “NIDL”.

IDS/II
Integrated Data Store II. A navigational database supported by GCOS 7 and GCOS 8.

IEEE
Institute of Electrical and Electronics Engineers. Organisation responsible for the POSIX set of standards.

import
(1) RPC: To obtain binding information from a name service database about a server that offers a given RPC interface by calling the RPC NSI import operation. (2) RPC: To incorporate constant, type and import declarations from one RPC interface definition into another RPC interface definition by means of the IDL import statement.

INET
The set of software on the DPX/2 that provides the TCP/IP protocol stacks and the applications that use them (e.g., FTP, SMTP, telnet, rlogin).

includes
UNIX source files to be included with compiled sources.

interface definition
RPC: A description of an RPC interface written in the DCE Interface Definition Language (IDL). See also **RPC interface**.

Internet
(1) When spelled with a lower-case "i", a collection of interconnected packet switching networks. (2) When spelled with an upper-case "I", (2a) the ARPANET, MILNET and NSFnet networks using TCP/IP, or (2b) the more formal name for the TCP/IP protocol suite, or (2c) A socket domain for intersystem communication over TCP/IP.

IOF
Interactive Operator Facility. The GCOS 7 time-sharing system which allows multiple users to access GCOS 7 simultaneously..

IP
Internet Protocol. The Internet standard for network layer (level 3) connectionless, best-effort packet delivery service.

IPS
Internet Protocol Suite. See "TCP/IP" and "Internet".

IQS
Integrated Query System - now called Relational Information System. A database access language.

ISL
Inter System Link. The communications channel which connects a DPS 7000 to its integrated network processor.

ISM
Integrated System Management. A tool for managing the resources in the Distributed Computing Model.

ISO
International Organisation for Standardisation. The body that developed the OSI reference model. The acronym "ISO" is often used where "OSI" should be used. In correct use, "ISO" refers only to the organisation and to standards such as ISO 8602. In other cases, use the term "OSI" (such as "OSI transport" or "OSI protocol" or "OSI network").

ISO/DSA
A networking protocol family in which layers 1 to 4 conform to OSI and layers 5 to 7 conform to DSA.

IUF
Installation and Updating Facility. A tool for installing GCOS 7 software.

JCL
Job Control Language. The GCOS 7 language for writing batch jobs.

JOBM
Job Management.

LAN
Local Area Network. A network technology that operates over short distances (up to a few thousand meters).

little endian
An attribute of data representation that reflects how multi-octet data are stored in memory. In little endian representation, the lowest addressed octet of a multi-octet data item is the least significant. See also **big endian**.

LM
Loading Module

LNM
LAN Network Module

LU6.2
Logical Unit 6.2. Session layer protocol in IBM's SNA, used as the basis for peer to peer exchanges.

manager
RPC: A set of remote procedures that implement the operations of an RPC interface and that can be dedicated to a given type of object. See also **object**, **RPC interface**.

Marshal
The process by which a calling RPC stub prepares local data to send it over the network to a called stub. Marshalling involves parsing, copying, and converting application data into protocol octet strings using a data representation known to the called stub, and then packaging the converted data for transmission.

maybe semantics
RPC: A form of idempotent semantics that indicates that the caller neither requires nor receives any response or fault indication for an operation, even though there is no guarantee that the operation is completed. An operation with maybe semantics is implicitly idempotent and lacks output parameters. See also **at-most-once semantics**, **broadcast semantics**, **idempotent semantics**.

Mbps
Megabits per second. Transmission speed expressed in millions of bits transferred per second.

Model, The
Short name of the Bull Distributed Computing Model. See DCM.

Mutex
A synchronisation object that provides mutual exclusion among threads. A mutex is often used to ensure that shared variables are always seen by other threads in a consistent state.

Name Service
Refer to CDS.

name service handle
RPC: An opaque reference to the context used by the series of next operations called during a specific NSI search or inquiry.

Name Service Interface (NSI)
RPC: A part of the application programming interface of the RPC runtime. NSI routines access a name service, such as CDS, for RPC applications.

NCA
Network Computing Architecture.

NCL1
New Common Line 1. This term refers to the current DPX/2 product line. See “DPS/2”.

NCL2
New Common Line 2. This term refers to the RISC-based DPX/20 product line. See “DPX/20”.

NCS
Network Computing System. The RPC component of OSF DCE based on technology from HP/Apollo.

NDR
Network Data Representation. The OSF/DCE standard for data representation within the RPC. Uses the “receiver-makes-it-right” strategy. Compare with “XDR”.

Network
The level 3 layer of the OSI reference model.

NIDL
Network Interface Description Language. A predecessor of IDL.

Non-blocking call
A procedure call in which control returns to the caller before the called procedure finishes executing.

NSI
Name Service Interface

object
(1) A data structure that implements some feature and has an associated set of operations. (2) RPC: For RPC applications, an object can be anything that an RPC server defines and identifies to its clients (using an object UUID). Often an RPC object is a physical computing resource such as a database, directory, device or processor. Alternatively, an RPC object can be an abstraction that is meaningful to an application, such as the service or the location of a server. See also **object UUID**.

object UUID
RPC: The universal unique identifier that identifies a particular RPC object. A server specifies a distinct object UUID for each of its RPC objects; to access a particular RPC object, a client uses the object UUID to find the server that offers the object.

octet
An 8-bit quantity of data.

opaque
A datum or data type whose contents are not visible to the application routines that use it.

Open Systems
Systems that conform to industry standards, rather than proprietary designs.

Open 8
A UNIX-based system acting as a partner to GCOS 8. It currently comes in four models: Open 8/CC, Open 8/AP2, Open 8/AP4 and Open 8/AP6. The “CC” model provides a TCP/IP connection only, while the “AP” models also provide UNIX-based distributed processing. Open 8 is a packaged set of hardware and software options on top of the DPX/2 platform.

OPEN 7
A UNIX subsystem of GCOS 7 which notably provides GCOS 7 with a gateway to the TCP/IP-UDP/IP environment.

OSF
Open Software Foundation. A consortium of vendors defining a set of technologies, among which is DCE, a fundamental component of DCM. Bull is a founding member of OSF. A primary goal of OSF is to make open software “enabling technologies” widely available.

OSF DCE
See “OSF” and “DCE”.

OSI
Open Systems Interconnection. A seven-layer model developed by ISO for the interconnection of co-operative computer systems. The seven layers are physical, link, network, transport, session, presentation, and application.

pipe
(1) RPC: A mechanism for passing large amounts of data in a remote procedure call.
(2) RPC: The data structure that represents this mechanism.

Port
A particular TCP/IP endpoint within a computer system.

Portmapper
A lookup service that records and supplies the port numbers of other servers.

POSIX
Portable Operating System Interface for UNIX. Name given to the set of IEEE standards applicable to UNIX. There are now about 20 standards in this set, only the first of which (operating systems programmatic interfaces for C language) is approved so far.

profile
RPC: An entry in a name service database that contains a collection of elements from which NSI search operations construct search paths for the database. Each search path is composed of one or more elements that refer to name service entries corresponding to a given RPC interface and, optionally, object.

Protocol
A formal description of message formats and the rules two or more machines follow to exchange those messages.

PSI
Peripheral Subsystem Interface. The communications channel between a DPS 7 or 7000 and a standalone network processor.

PTF
Program Temporary Fix.

RBF6
Remote Batch Facility. A facility which allows a user on a DPS 6 to request the execution of a job file.

RPC
Remote Procedure Call. A mechanism that allows a program on one machine to call a subroutine on a remote machine as if it were local.

RPC Client
The software subsystem that implements the OSF DCE request-response protocol on the calling machine. The client is the party that initiates communication with a server in order to make one or more remote procedure calls.

RPC Daemon (RPCD)
RPC: The process that provides the endpoint map service for a system. The RPC daemon is started by the **rpcd** command. See also **endpoint map service**.

RPC on GCOS
Implementation of the DCE RPC mechanism on GCOS platforms (GCOS 7 or 8).

RPC/CN
RPC Connection oriented.

RPC/DG
RPC Datagram.

RPC interface
RPC: A logical grouping of operation, data type, and constant declarations that serves as a network contract for calling a set of remote procedures. See also **interface definition**.

RPC protocol
RPC: An RPC specific communications protocol that supports the semantics of the DCE RPC API and runs over either connectionless or connection oriented communications protocols.

RPC runtime
RPC: A set of operations that manages communications, provides access to the name service database, and performs other tasks, such as managing servers and accessing security information, for RPC applications.

RPC Server
The software subsystem that implements the OSF DCE request-response protocol on the called machine.

RPC Simu Regulation
Valid only in a transactional context. TDS product regulating the number of RPC transactions that can be processed concurrently.

rundown procedure
RPC: A procedure, typically used with a context handle, that is called following a communications failure to recover resources reserved by a server for servicing requests by a particular client. See also **context handle**.

SBR
System Behaviour Reporter. A tool for tuning a GCOS 7 system.

SER
Software Error Recovery.

Server
An application or machine that provides a service.

SM
Sharable Module. The executable form of a re-entrant program on GCOS 7.

SNA
Systems Network Architecture. IBM's architecture for the design of communication hardware and software.

Socket
An API to TCP and UDP.

Socket Agent
An application on Open 8 that acts as a proxy for GCOS 8 to the TCP/IP network.

SQL
Structured Query Language. A database access language.

Stream
(1) A type of socket. (2) A stream of bytes in a communication or file.

STAR
Software Technical Action Request.

Stub
A routine used by the RPC mechanism. There are two types: (1) client stubs, and (2) server stubs. A client stub is a body of code with the same call interface as a remote procedure which hides RPC details from a calling procedure. Only a client stub can perform binding operations and make initial RPC calls. A server stub is a body of code that isolates RPC mechanisms from a remote program. A server stub cannot initiate remote procedure calls; after being called by a client, however, the server stub can act as a calling and called stub.

TCP
Transmission Control Protocol. A transport layer protocol in the TCP/IP family providing reliable delivery.

TCP/IP
Transmission Control Protocol over Internet Network protocol. Also a family of protocols (e.g., TCP/IP, UDP/IP etc.) developed in the late 1960s by DARPA, an agency of the U.S. Department of Defence, and the applications (e.g., FTP, SMTP, telnet, etc.) that run over these protocols.

TDS
Transaction Driven Subsystem. The GCOS 7 environment for creating and running transactional applications.

TDSGEN
A utility allowing generation of a customer-specific TDS sub-system.

TDS Master
The operator of the transactional system. There is a single master during a TDS session.

Thread
A single sequential flow of control with one point of execution at any instant.

TILS
Transactional and Interactive Load Simulator. A GCOS 7 tool for simulating system load.

Time Service
Refer to DTS

TNS
Transport and Network Subsystem. The interface between a DPS 7000 and its integrated network processor.

TP8
Transactional Processor. The GCOS 8 environment for creating and running transactional applications.

Transport
The level 4 layer of the OSI reference model.

UDP/IP
User Datagram Protocol over Internet Network protocol. A transport layer protocol in the TCP/IP family. It provides a simple transport without flow control, guaranteed delivery or sequencing.

UFAS
Unified File Access System. A file access method used by GCOS 7.

UFT
Unified File Transfer. A file transfer mechanism used in a DSA network.

unexport
RPC: To remove binding information from a server entry in a name service database. See also **export**.

UNIX
A popular operating system originally developed by AT&T Bell Laboratories.

Unmarshal
The process by which a called stub manages network data received from a calling stub. Unmarshalling involves disassembling protocol octet strings from a packet, parsing, copying, and converting them into application data.

User
(1) The person using a computer. (2) Sometimes used interchangeably with “client”.

U
TC See Coordinated Universal Time.

VCAM
Virtual Communications Access Method. The communications access method between GCOS 7 and a DSA network.

VMM
Virtual Memory Management. A GCOS 7 administration tool.

WAN
Wide Area Network.

Well-known Port
Any of a set of port numbers preassigned for specific uses by TCP and UDP.

WIL
Windowed Information Link. Integrates PC office applications with GCOS 7 host services.

Workstation
(1) A desktop computer system, often UNIX-based but sometimes MS-DOS or OS/2 based. (2) A collection of processes on GCOS 8.

X.25
CCITT & ISO standard protocol for the network layer (level 3) over a packet switching network. Also includes the HDLC specification for the link layer (level 2).

X.500
CCITT and ISO standard for directory services.

X/Open
An organisation involved in specifying standards for portability in the UNIX environment.

XCP2
Extended Co-operative protocol level 2. The X/Open version of IBM's LU6.2 protocol for peer to peer exchanges.

XDR
External Data Representation. The Sun Microsystems standard for machine-independent data representation. Each machine converts to and from this representation. Compare with “NDR”.

XDS
X/Open Directory Service interface. The X/Open defined API to X.500

XOM
X/Open Object management Interface. An API used in X.500.

Index

For explanations of terms used in this manual, refer also to the Glossary.

A

access method 1-6
  RPC 1-6
  specific 1-6
Affinity
  with GCOS 7 3-6
  with GCOS 8 4-8
Affinity Visual 1-4
ally 1-5, 5-3
  functions 5-5
  location 5-7
API
  exporting 2-6
ARM 3-12

B

batch
  GCOS 7 3-2
  GCOS 8 4-2
binding 2-8, 5-7
  from the client 5-8
  to the ally 5-7
BOS/TP 1-1

C

CASE 1-4
CDS 2-7, 2-9, 5-10
  architecture 2-7
  updating 2-8
client stub 2-5
client/server 2-3
communication
  access method 1-6
communications
  DSA 4-6
  TCP/IP 4-8
communications protocol 5-8, 5-10
Computer Aided Software Engineering see CASE
CPI-C 1-6, 3-9

D

Data Management-IV see DM-IV
DCE 2-1
  general 2-1
  hardware implementation 2-11
  implementation 2-9
  restrictions 2-11
  software components 2-9
  systems supported 2-11
  what is? 2-1
  why? 1-1, 2-4
DCE Ally see ally
DCE RCP on GCOS see RCP
DCE RPC ally see Ally
DCE service 5-3
DCF 1-3, 1-4
DDA 1-1
DFS 2-9
directory service
  architecture 2-7
Directory services 1-4
Distributed computing environment see DCE
Distributed Computing Facility see DCF
Distributed Computing Model see DCM
Distributed Printing Facility see DPF
DJP 3-9
DM-IV 4-4
documentation
  GCOS DCE 1-9
DOF7 3-9
DPS 7 3-1

DPS 7000 3-1
DPS 8
  general 4-1
DSA
  with GCOS 7 3-4
  with GCOS 8 4-6
DSA communications 4-6
DTS 2-9, 5-10

E

EDI 1-4
Electronic mail 1-4

F

FEPS 3-6
File Management Supervisor see FMS
FMS 4-2
FormsSQL 4-5
FTAM 1-6
FTAM7 3-9
FTP 1-6

G

GAC 3-9
GCOS 7
  administration 3-12
  batch 3-2
  database management 3-11
  DSA 3-4
  emulator supported 3-6
  file management 3-9
  limitations 3-13
  main features 3-1
  outline 3-1
  remote operation 3-9
  telecommunications 3-2
  terminal supported 3-6
  terminals supported 3-6
  working environment 3-2
GCOS DCE 1-1, 5-1
  advantages 1-4
  alternatives 1-6
  communications 5-11
  components 5-3
  documentation 1-9
  functions supported 5-10
  global architecture 1-5
  restrictions 5-10
GCOS-UNIX communications 5-1, 5-8, 5-10
GCOS 8
  data management 4-4
  DSA communications 4-6
  features 4-1
  file management 4-2
  implementation 4-1
  networking 4-5
  primary network 4-6
  secondary network 4-8
  SQL 4-5
  TCP/IP communications 4-8
  terminal connections 4-8
  time sharing 4-2
  transaction processing 4-4
GCOS 8-Link 4-10, 5-10

H

HSL 3-6, 5-1

I

IDL 2-6
IDS/II 3-11
Integrated System Development see ISD
Integrated System Management see ISM
INTEREL 4-5
INTEREL-SQL 4-5
interface definition language see IDL
IOF 3-2
  pass through 3-9
IQS 3-11
ISD 1-4
ISM 1-4
IUF 3-12

N

network design 1-8
networking
  GCOS 7 3-2
  GCOS 8 4-5

O

Open 8
- architecture 4-10
- Telnet see Telnet
- terminal connections 4-13

Open Software Foundation see OSF
OPEN 7
- communications 3-6, 5-1
- general 3-6

Open 8
- communications 5-1
- GCOS 8-Link 4-10
- general 4-10
- hardware 4-10
- software 4-10

Open 8/CC 4-10
ORACLE 3-11
OSF 2-1

P

primary network
- GCOS 7 3-4
- GCOS 8 4-6

pseudo socket 5-8

R

RBF6 3-9
RCP 1-3, 1-5
RCP on GCOS see RCP
remote procedure call see RPC
RPC 2-9, 5-3
- functions supported 5-10
- in DCE 2-1
- network design 1-8
- operation 2-5
- restrictions 5-10

RPC ally see Ally
RPC on GCOS 7
- architecture 5-8

RPC on GCOS 8
- architecture 5-10

S

SBR 3-12
secondary network
- GCOS 7 3-6
- GCOS 8 4-8

security 2-9
security function 5-10
server stub 2-5
SQL 1-6, 3-11, 4-5
stub
- general 2-5
- generating 2-6

T

TCP/IP 5-1
- with GCOS 7 3-6
- with GCOS 8 4-8

TDS 3-2
Telnet 4-13
terminal connections
- GCOS 7 3-6
- GCOS 8 4-8
- Open 8 4-13
- OPEN 7 3-8

threads 2-9, 5-10
TILS 3-12
Time Sharing Executive see TSE
Time Sharing System see TSS
TNS 3-6
TP8 4-4
Transaction Processor 8 see TP8
TSE 4-4
TSS 4-2

U

UDP/IP 5-1
- with GCOS 7 3-7

UFAS 3-9
UFT 1-6, 3-9

V

VCAM 3-4
VMM 3-12

X

XCP1 3-9
XCP2 3-9

Technical publication remarks form

Title :	DPS7000/XTA NOVASCALE 7000 GCOS 7 Overview Distributed Computing

Reference N° :	40 A4 61CF 01	**Date:**	August 1995

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.
If you require a written reply, please include your complete mailing address below.

NAME : ____________________ Date : __________

COMPANY : ____________________

ADDRESS : ____________________

____

Please give this technical publication remarks form to your BULL representative or mail to:

Bull - Documentation D$^{ept.}$
1 Rue de Provence
BP 208
38432 ECHIROLLES CEDEX
FRANCE
info@frec.bull.fr

Technical publications ordering form

To order additional publications, please fill in a copy of this form and send it via mail to:

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

Phone:	+33 (0) 2 41 73 72 66
FAX:	+33 (0) 2 41 73 70 66
E-Mail:	srv.Duplicopy@bull.net

CEDOC Reference #	Designation	Qty
__ __ ____ _ [__]		
__ __ ____ _ [__]		
__ __ ____ _ [__]		
__ __ ____ _ [__]		
__ __ ____ _ [__]		
__ __ ____ _ [__]		
__ __ ____ _ [__]		
__ __ ____ _ [__]		
__ __ ____ _ [__]		
__ __ ____ _ [__]		
__ __ ____ _ [__]		
__ __ ____ _ [__]		
[__] : The latest revision will be provided if no revision number is given.		

NAME: ______________________________ Date: __________

COMPANY: ______________________________

ADDRESS: ______________________________

____

PHONE: ____________________ FAX: ____________________

E-MAIL: ______________________________

For Bull Subsidiaries:

Identification: ______________________________

For Bull Affiliated Customers:

Customer Code: ______________________________

For Bull Internal Customers:

Budgetary Section: ______________________________

For Others: Please ask your Bull representative.

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
40 A4 61CF 01