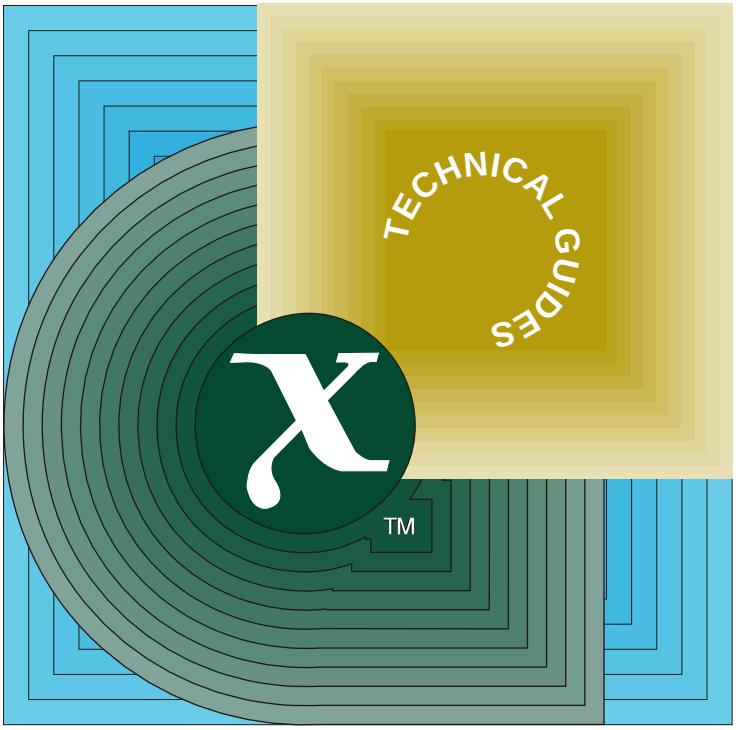
Guide

Distributed Computing Services (XDCS) Framework



THE Open GROUP

[This page intentionally left blank]

X/Open Guide

Distributed Computing Services (XDCS) Framework

X/Open Company Ltd.

X/Open Guide

Distributed Computing Services (XDCS) Framework

ISBN: 1 872630 64 2

X/Open Document Number: G212

Published by X/Open Company Ltd., U.K.

Any comments relating to the material contained in this document may be submitted to X/Open at:

X/Open Company Limited
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom

or by Electronic Mail to:

XoSpecs@xopen.co.uk

Contents

List of Figures

List of Tables

Preface

X/Open

X/Open is an independent, worldwide, open systems organisation supported by most of the world's largest information systems suppliers, user organisations and software companies. Its mission is to bring to users greater value from computing, through the practical implementation of open systems.

X/Open's strategy for achieving this goal is to combine existing and emerging standards into a comprehensive, integrated, high-value and usable system environment, called the Common Applications Environment (CAE). This environment covers the standards, above the hardware level, that are needed to support open systems. It provides for portability and interoperability of applications, and allows users to move between systems with a minimum of retraining.

The components of the Common Applications Environment are defined in X/Open CAE Specifications. These contain, among other things, an evolving portfolio of practical application programming interfaces (APIs), which significantly enhance portability of application programs at the source code level, and definitions of, and references to, protocols and protocol profiles, which significantly enhance the interoperability of applications.

The X/Open CAE Specifications are supported by an extensive set of conformance tests and a distinct X/Open trademark - the XPG brand - that is licensed by X/Open and may be carried only on products that comply with the X/Open CAE Specifications.

The XPG brand, when associated with a vendor's product, communicates clearly and unambiguously to a procurer that the software bearing the brand correctly implements the corresponding X/Open CAE Specifications. Users specifying XPG-conformance in their procurements are therefore certain that the branded products they buy conform to the CAE Specifications.

X/Open is primarily concerned with the selection and adoption of standards. The policy is to use formal approved *de jure* standards, where they exist, and to adopt widely supported *de facto* standards in other cases.

Where formal standards do not exist, it is X/Open policy to work closely with standards development organisations to assist in the creation of formal standards covering the needed functions, and to make its own work freely available to such organisations. Additionally, X/Open has a commitment to align its definitions with formal approved standards.

X/Open Specifications

There are two types of X/Open specification:

- *CAE Specifications*

  CAE (Common Applications Environment) Specifications are the long-life specifications that form the basis for conformant and branded X/Open systems. They are intended to be used widely within the industry for product development and procurement purposes.

Developers who base their products on a current CAE Specification can be sure that either the current specification or an upwards-compatible version of it will be referenced by a future XPG brand (if not referenced already), and that a variety of compatible, XPG-branded systems capable of hosting their products will be available, either immediately or in the near future.

CAE Specifications are not published to coincide with the launch of a particular XPG brand, but are published as soon as they are developed. By providing access to its specifications in this way, X/Open makes it possible for products that conform to the CAE (and hence are eligible for a future XPG brand) to be developed as soon as practicable, enhancing the value of the XPG brand as a procurement aid to users.

- *Preliminary Specifications*

  These are specifications, usually addressing an emerging area of technology, and consequently not yet supported by a base of conformant product implementations, that are released in a controlled manner for the purpose of validation through practical implementation or prototyping. A Preliminary Specification is not a ''draft'' specification. Indeed, it is as stable as X/Open can make it, and on publication has gone through the same rigorous X/Open development and review procedures as a CAE Specification.

  Preliminary Specifications are analogous with the ''trial-use'' standards issued by formal standards organisations, and product development teams are intended to develop products on the basis of them. However, because of the nature of the technology that a Preliminary Specification is addressing, it is untried in practice and may therefore change before being published as a CAE Specification. In such a case the CAE Specification will be made as upwards-compatible as possible with the corresponding Preliminary Specification, but complete upwards-compatibility in all cases is not guaranteed.

In addition, X/Open periodically publishes:

- *Snapshots*

  Snapshots are ''draft'' documents, which provide a mechanism for X/Open to disseminate information on its current direction and thinking to an interested audience, in advance of formal publication, with a view to soliciting feedback and comment.

  A Snapshot represents the interim results of an X/Open technical activity. Although at the time of publication X/Open intends to progress the activity towards publication of an X/Open Preliminary or CAE Specification, X/Open is a consensus organisation, and makes no commitment regarding publication.

  Similarly, a Snapshot does not represent any commitment by any X/Open member to make any specific products available.

X/Open Guides

X/Open Guides provide information that X/Open believes is useful in the evaluation, procurement, development or management of open systems, particularly those that are X/Open-compliant.

X/Open Guides are not normative, and should not be referenced for purposes of specifying or claiming X/Open-conformance.

This Document

The X/Open Distributed Computing Services (XDCS) Framework is a comprehensive blueprint for a complete system software environment that will allow open systems to better address the critical needs in enterprise-wide distributed computing.

The framework identifies:

- the required services
- the relationship between the services
- the programming interfaces to the services
- the protocols and data formats that provide interoperability between systems that support the services.

The framework represents a broad industry consensus on the elements in a complete distributed computing environment. The services in the framework were drawn from a wide variety of sources including:

- the X/Open Portability Guide (see reference **XPG**)
- the Object Management Group's (OMG) Common Object Request Broker Architecture (see reference **CORBA**)
- the Open Software Foundation's (OSF) Distributed Computing Environment (see reference **DCE**)
- UNIX International's UI-ATLAS Distributed Computing Framework (see reference **ATLAS**).

The goal of the framework is to provide portability and interoperability of distributed applications. Portability of applications is supported by specifying the programming interfaces to the services. Interoperability is supported by specifying the protocols and data formats that are used in communicating across the network to provide the service.

This document focuses on the components and structure of the framework. For each layer it identifies the supported technology, APIs, protocols and data formats. It also shows how each component fits into the overall structure of the framework.

Trade Marks

UNIX® is a registered trade mark of UNIX System Laboratories Inc. in the U.S.A. and other countries.

X/Open and the 'X' device are trademarks of X/Open Company Limited in the U.K. and other countries.

Palatino is a trademark of Linotype AG and/or its subsidiaries.

Referenced Documents

The following documents are referenced in this Guide:

ACL IEEE Computer Society, Mandatory Access Control, P1003.6, Draft for Ballot, November 1991.

ACSE ISO 8649, Information Processing Systems - Open Systems Interconnection - Services Definition for the Association Control Service Element, International Standard 8649, 1988 and ISO 8650, Information Processing Systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element, International Standard 8650, 1988.

ATLAS UNIX International, UI-ATLAS - A Technical Overview, UI, September, 1991.

CLNP ISO 8473, Information Processing Systems - Open Systems Interconnection - Protocol for providing the Connectionless-mode Network Service, International Standard 8473, 1988.

CMIP ISO 9596, Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol Specification, International Standard 9596, 1991.

CORBA Object Management Group, Common Object Request Broker Architecture (CORBA), OMG, January 1992.

CPIC X/Open Developer's Specification, CPI-C, 1990.

DCE Open Software Foundation, Introduction to OSF DCE, OSF, December 31, 1991.

DII Object Management Group, Common Object Request Broker Architecture (CORBA), OMG, January 1992.

DNS Request for Comments 1035, Domain names - implementation and specification, Mockapetris, P.V., November 1987.

FTAM ISO 8571, Information Processing Systems - Open Systems Interconnection - File Transfer, Access, and Management (FTAM), International Standard 8571, 1988.

FTP Request for Comments 959, File Transfer Protocol, Postel, Jon B., October 1985.

IDL Object Management Group, Common Object Request Broker Architecture (CORBA), OMG, January 1992.

IP Request for Comments 791, Internet Protocol, Postel, Jon B., September 1981.

ISAM X/Open Portability Guide, Data Management, Prentice Hall, 1988.

ISORPC ISO CD 11578, Information Processing Systems - Open Systems Interconnection - RPC, 1991.

KERB "The Kerberos Network Authentication Service", Kohl, John and Neuman, B Clifford. Internet Draft RFC, revision 5, 17 April 1992.

LU6.2 International Business Machines Corp., Systems Network Architecture, Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2, SC30-3269-2, December, 1984.

POSIX ISO/IEC 9945-1: 1990, IEEE Standard 1003.1, Information Technology - Portable Operating System Interface (POSIX) - Part 1 : System Application Program Interface (API) C Language, Institute of Electrical and Electronics Engineers, 1990.

ISO/IEC DIS 9945-2: 1992, IEEE Standard 1003.2-Draft, Information Technology - Portable Operation System Interface (POSIX) - Part 2 : Shell and Utilities, Institute of Electrical and Electronics Engineers, 1992.

PRES ISO 8823, Information Processing Systems - Open Systems Interconnection - Connection-Oriented Presentation Protocol Specification, International Standard 8823, 1988.

SESS ISO 8327, Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Protocol Specification, International Standard 8327, 1988.

SMTP Request for Comments 821, Simple Mail Transfer Protocol, Postel, Jon B., August 1982.

SNMP Request for Comments 1098, A Simple Network Management Protocol, Case, Jeffrey D.; Fedor, Mark S.; Schoffstall, Martin, L.; and Davin, James R. April 1989.

SQL X/Open, SQL, Preliminary Specification, X/Open, 1991.

SQLACC X/Open, SQL Remote Database Access, Snapshot, X/Open, 1991.

TFA ISO/IEC JTC1/SC22/WG15, POSIX P1113.8/D6, 22 January 1992: Draft, Transparent File Access, Supplement to Portable Operating System Interface for Computer Environments.

THREADS IEEE Computer Society, Threads Extension for Portable Operating Systems P1003.4a/D5, January, 1991.

UDP Request for Comments 854, TELNET Protocol Specification, Postel, Jon B., May 1983.

X11 X/Open CAE Specification, X/Open Window Management: X Window System Protocol, X/Open, 1991.

X.400 ISO 10021, Information Processing Systems - Open Systems Interconnection - MOTIS, International Standard 10021, 1990.

X.500 ISO 9594, Information Processing Systems - Open Systems Interconnection - The Directory, International Standard 9594, 1991.

XA X/Open document number C193, Distributed Transaction Processing: The XA Specification (XA), Snapshot, 1991.

XAP X/Open document number P203, ACSE/Presentation Services API (XAP) Preliminary Specificaton, 1992.

XDS X/Open document number C190, API to Directory Services (XDS), CAE Specification, 1991.

XFTAM X/Open document number P206, FTAM High-Level API (XFTAM) Preliminary Specificaton, 1992.

XMHS X/Open document number C191, API to Electronic Mail (X.400), CAE Specification, 1991.

XNFS X/Open document number C130, Protocols for X/Open Interworking: XNFS, CAE Specification, 1990. X/Open, 1991.

XPG4 X/Open, Portability Guide, Fourth Edition:

— System Interfaces and Headers, Issue 4, X/Open reference C202, July 1992

— Commands and Utilities, Issue 4, X/Open reference C203, July 1992

— System Interface Definitions, Issue 4, X/Open reference C204, July 1992.

XSM X/Open document number S140, Systems Management: Reference Model, Snapshot, 1991.

XSMB X/Open document number D110, Protocols for X/Open PC Interworking: SMB, Developers Specification, 1991.

XTI X/Open document number C196, Transport Interface (XTI), CAE Specification, 1992.

Chapter 1

Introduction

1.1 Scope

The X/Open Distributed Computing Services (XDCS) Framework is a comprehensive blueprint for a complete system software environment that will allow open systems to better address the critical needs in enterprise-wide distributed computing. The framework was developed within X/Open to add structure and consistency to the work of individual X/Open working groups. The framework is representative of the X/Open Common Applications Environment and supports, but goes beyond, a Distributed Computing model. It is recognised that such a framework cannot be a static entity; as new interfaces and protocols are defined they will be incorporated into the framework. The framework identifies:

- The required services
- The relationship between the services
- The programming interfaces to the services
- The protocols and data formats that provide interoperability between systems that support the services.

The framework represents a broad industry consensus on the elements in a complete distributed computing environment. The services in the framework were drawn from a wide variety of sources including:

- the X/Open Portability Guide (see reference **XPG4**)
- the Object Management Group's (OMG) Common Object Request Broker Architecture (see reference **CORBA**)
- the Open Software Foundation's (OSF) Distributed Computing Environment (see reference **DCE**)
- UNIX International's UI-ATLAS Distributed Computing Framework (see reference **ATLAS**).

1.2 Goals

The goal of the framework is to provide portability and interoperability of distributed applications. Portability of applications is supported by specifying the programming interfaces to the services. Interoperability is supported by specifying the protocols and data formats that are used in communicating across the network to provide the service. In some instances the framework supports multiple protocols (for example, OSI and TCP/IP as Transport Providers) for a given service. This is necessary to address diverse needs in a complex marketplace. To provide interoperability in a multiprotocol environment, it is necessary to select a set of protocol profiles[1] to support, in addition to the elements defined by the XDCS framework. If two systems are to interoperate, they must support the same protocol profile.

---

1. A *protocol profile* is a consistent set of protocols that support a set of related services. Such profiles are defined by ISO and the regional profile workshops (OIW, EWOS and AOW). X/Open profiles on the other hand cover both the functionality needed to meet a focused set of needs (a functional profile) and the protocols needed to support this functionality in a networked environment (a protocol profile).

Overview of the XDCS Framework

2.1 Overall Architecture

The XDCS Framework is a comprehensive framework for heterogeneous distributed computing. The overall architecture of the framework is portrayed in Figure 2-1.

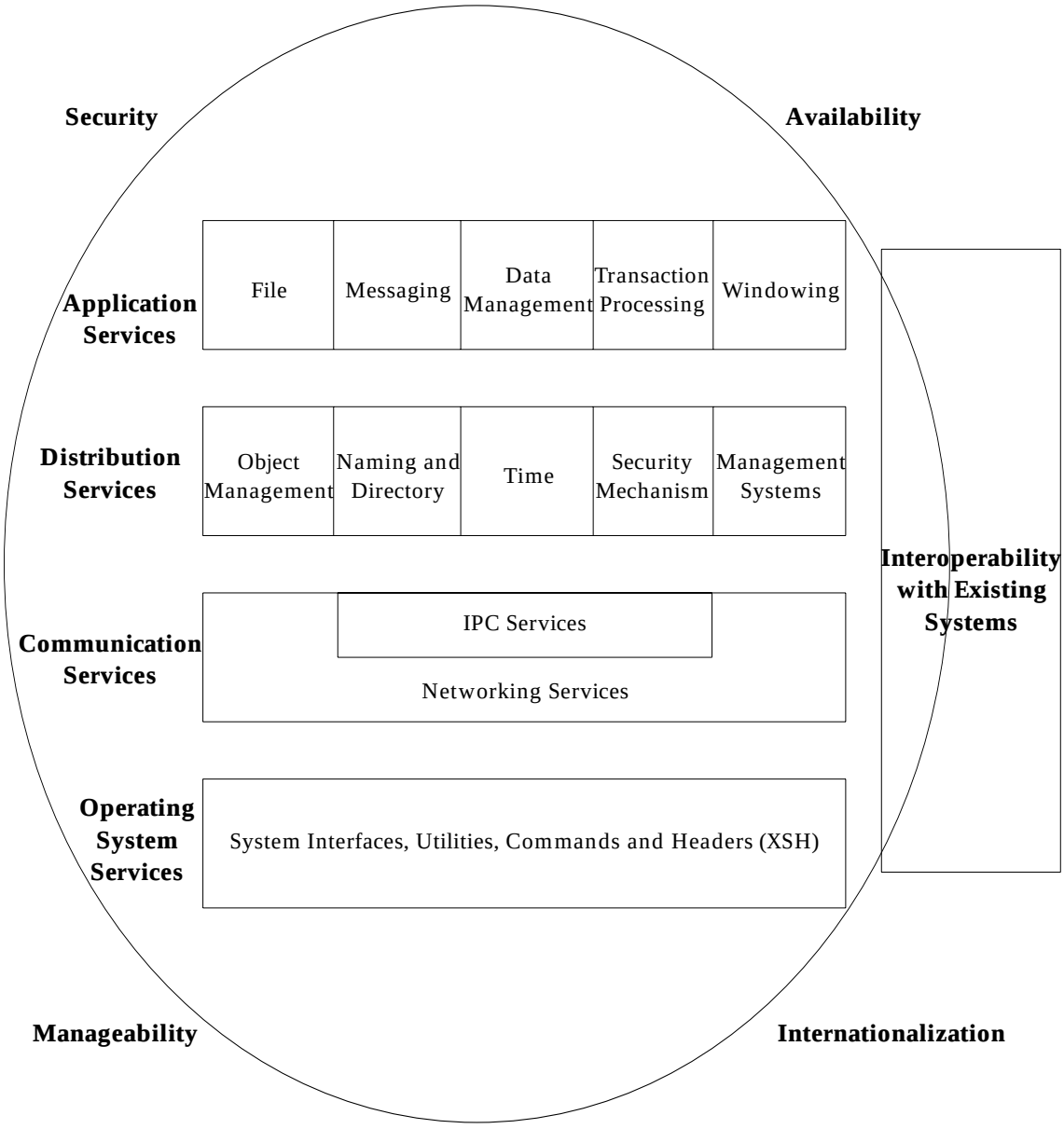


Figure 2-1 XDCS Framework

The XDCS Framework is organized as a set of four layers. Each layer provides a level of functionality in an enterprise computing network:

- The **Operating System Services** provide an environment for running distributed software on each node of the network.
- The **Communication Services** provide the services that allow applications to communicate reliably across the network independently of the underlying network topology, networking protocols or data representations.
- The **Distribution Services** provide a set of services that support consistency in distributed applications. These services address the fundamental issues in computing including the processing model, the naming model, the security model and the management model.
- The **Application Services** provide the enabling distributed system software to support the development of distributed applications.

The layering in the model expresses three relationships:

- **Support** - The services lower in the model support services higher in the model. For example, the Application Services and the Distribution Services can run over the networks supported in the Communication Services layer.
- **Conceptual Layering** - A higher level service is, in general, conceptually independent of the lower level service. Although the framework is structured in layers, there is no strict hierarchical dependency as in the OSI Reference Model. An application is free to directly invoke functionality at any layer in the model. The services in the framework are often interdependent. For example, the RPC Services (in IPC Services) use the Security Service to provide secure remote procedure calls and the Security Service is implemented using the RPC Services.
- **Consistency** - The lower level services can provide consistency among the higher level services. For example, the distribution services provide consistency in processing model, naming, security and management to the application services. Although the framework provides the means for consistency amongst services, it does not enforce it. The potential for introducing such consistency can be utilised when new specifications are developed by X/Open.

The boxes in the framework describe individual services. In addition, there are a set of qualities (security, availability, manageability and internationalization) that apply to all services. This is indicated in the framework diagram by the *quality circle*. These are requirements that should be considered by all of the individual X/Open work programmes when specifications for specific services are being developed.

The core framework describes services that provide application portability and interoperability among X/Open-conformant systems. There is also a need to provide interoperability with existing systems in the enterprise. The box marked *Interoperability with Existing Systems* provides these services. The goal is to provide interoperability with these systems as they are today. This means that the open systems must adapt to the networking and services provided on these systems. Services are required at the Communication, Distribution and Application Services levels in the XDCS framework. The XDCS qualities may be implemented on existing systems in ways which are not compatible with their implementation on open systems. This is why the circle only partially penetrates the Interoperability box.

2.2 Structure of the Document

The structure of this document follows the layers of the XDCS Framework. There is a chapter for each layer. The chapter begins with a table that summarizes the components in the layer. For each component it lists the supported programming interfaces, protocols and data formats. The chapter continues with a description of the individual components in the layer. The emphasis is on the structure of the component and its relation to other components in the framework. Figures are included to show the relationship between components.

The XDCS Framework will evolve as distributed computing technology matures.

2.3 Target Audience

This document is targeted primarily at technical readers who would like to understand the big picture of how individual X/Open Specifications fit together into a coherent scheme for enterprise computing and how this scheme is likely to evolve. The document concentrates on how the individual pieces of technology fit together. Although the document briefly describes each technology, it is assumed that the reader interested in a deeper understanding of the technology will consult the references.

This document is also intended to be used by X/Open as a framework implicitly identifying which new specifications it should develop and giving the individual technical programmes of X/Open the architectural context to work in.

Chapter 3

Base OS Services

The goal of the Operating System Services is to provide a consistent environment for running applications on each of the diverse nodes in the network. The interfaces in the OS Services enable the higher levels of XDCS to be independent of the underlying hardware and operating system implementation. Table 3-1 shows the components of the OS Services.

The interfaces to the Operating System Services are documented in the X/Open System Interfaces, Utilities, Commands and Headers (XSH) specifications in the X/Open Portability Guide (see reference **XPG4**). The XSH specifications describe the basic operating system services including process management, input/output, security and memory management. The XSH specifications are fully conformant to the standards ISO/IEC 9945-1: 1990 and ISO/IEC 9945-2: 1992 for Portable Operating System Interface (see reference **POSIX**).

Base Operating System		
Component	**Programming Interface**	**Protocol & Formats**
System Services	X/Open XSH, POSIX.1 (ISO/IEC 9945-1), POSIX.2 (ISO/IEC DIS 9945-2)	N/A
Threads	IEEE P1003.4a <draft>	N/A

Table 3-1 Base Operating System Services

To enable the execution of high performance distributed applications, the Operating Systems Services also include a threads facility. The threads component supports the creation, management and synchronization of multiple threads of control within a single process. This is useful in the implementation of servers that will be responding to multiple simultaneous requests and in the implementation of clients that will have multiple pending requests. The threads facility is derived from the draft IEEE P1003.4a threads specification (see reference **THREADS**). The threads facility will evolve to achieve conformance with the eventual P1003.4a specification.

A convenient class of distributed services is formed by extending the basic operating system services to work transparently over the network. The Distributed File Sharing Services in the XDCS Framework use this approach. In extending the file system services to work over the network, additional error codes and, in some cases, compromises in the file system semantics are necessary. The draft IEEE P1003.8 Transparent File Access specification deals with these issues (see reference **TFA**).

Chapter 4

Communication Services

The Communication Services provide the networking and inter-process communication (IPC) model to support distributed services between XDCS-conformant nodes. The communication services are partitioned into two categories:

- Networking Services

  The networking services provide the internetworking, transport and common higher level protocols and interfaces that enable applications to communicate independently of the underlying network topology and protocols.

- IPC Services

  — RPC Services

  The RPC services provide the protocols, data representation services and runtime environments to support client/server rendezvous and communication using a remote procedure call paradigm.

  — Peer-to-Peer Conversational Services

  The Common Programming Interface for Communications and OSI Transaction Processing protocol provide conversational or dialog paradigm programming services. This support enables distributed process conversation/dialog to be established, allowing short or long running transaction-oriented exchanges between two peer processes.

4.1 Networking Services

The networking services in the XDCS Framework are summarised in Table 4-1. The XDCS services are designed to be independent of the underlying network. This is achieved by implementing the network provider to support the XTI interface (see reference **XTI**) and implementing the services directly or indirectly to this interface. This allows the services to be used over any of the networks. The OSI and TCP/IP services in the XDCS Framework support the XTI interface.

Communication Services - Networking Services		
Component	**Programming Interface**	**Protocol & Formats**
OSI	X/Open XTI, XAP	ISO 8802 (LANS), 8208 (X.25), 8473 (CLNP), 8072/3(COTP), 8602 (CLTP), 8326/7 (Session), 8822/3 (Presentation), 8649/50 (ACSE)
TCP/IP	X/Open XTI	IETF RFCs 791 (IP), 793 (TCP), 768 (UDP), 1006 (OSI over TCP/IP), ISO 8802 (LANS), 8208 (X.25)

Table 4-1 Networking Services

4.1.1 OSI Services

The protocol stack supported by the OSI Services is shown in Figure 4-1. The core OSI stack consists of the 8802 LANs ("Ethernet"-style CSMA/CD, Token Ring and Token Bus), X.25, CLNP (Connectionless Network Layer Protocol), COTP (Connection-Oriented Transport Protocol), CLTP (Connectionless Transport Protocol), Session, Presentation and ACSE (Association Control Service Element) protocols - see references **ACSE**, **CLNP**, **PRES**, **SESS**.

The OSI transport protocols can be directly accessed through the XTI interface so that any properly written XTI-based application can run over the OSI network. Three protocol profiles are supported by the OSI Transport Services (Class 4, 2, 0/X.25; Class 4/CLNP/X.25+8802 LANs; and CLTP/CLNP/X.25+8802 LANs). The upper layers of OSI can be accessed directly through the X/Open ACSE/Presentation (XAP) interface (see reference OSI Applications in the framework are X.500 (Directory Services, see references **X.500** and **XDS**), CMIP (Common Management Information Protocol, see reference **CMIP**), X.400 (MHS Electronic Messaging, see references **X.400** and **XMHS**), and FTAM (File Transfer, Access and Management - see references **FTAM** and **XFTAM**). X.500 and CMIP are supported in the Distribution Services. FTAM and X.400 are supported in the Application Services.

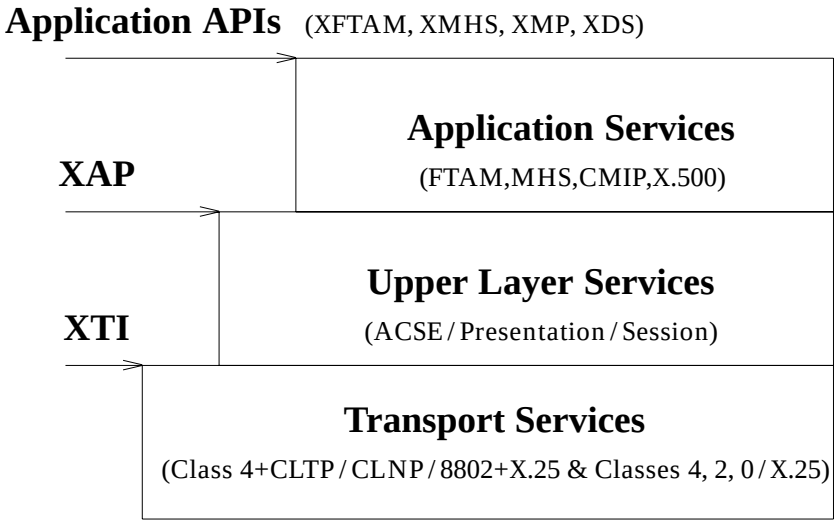


Figure 4-1 OSI Protocol Stack

4.1.2 TCP/IP Services

The protocol stack supported by the TCP/IP services[2] are shown in Figure 4-2. The core TCP/IP stack consists of 8802 LANs, X.25, IP (Internet Protocol), UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) protocols (see references **IP**, **TCP**, **UDP**). The TCP/IP transport providers (TCP and UDP) support the XTI interface so that any properly written XTI-based application can run over the TCP/IP network. Two protocol profiles are supported by the

2. The term the Internet Protocol Suite (IPS) is often used to more precisely refer to the whole suite of protocols. In this document the more popular term "TCP/IP" is used to refer to the IPS Transport Providers (UDP/IP and TCP/IP).

TCP/IP services (UDP/IP,X.25+8802 LANS; and TCP/IP/X.25+8802 LANs). The higher level TCP/IP services are supported by standard interfaces.

The framework supports interfaces to the Internet Domain Name Service (DNS directory services - see reference **DNS**), SNMP (Simple Network Management Protocol - see reference **SNMP**), FTP (File Transfer Protocol - see reference **FTP**), and SMTP (Simple Mail Transfer Protocol - see reference **SMTP**). The DNS and SNMP are used in the Distribution Services. FTP and SMTP are used in the Interoperability Services.

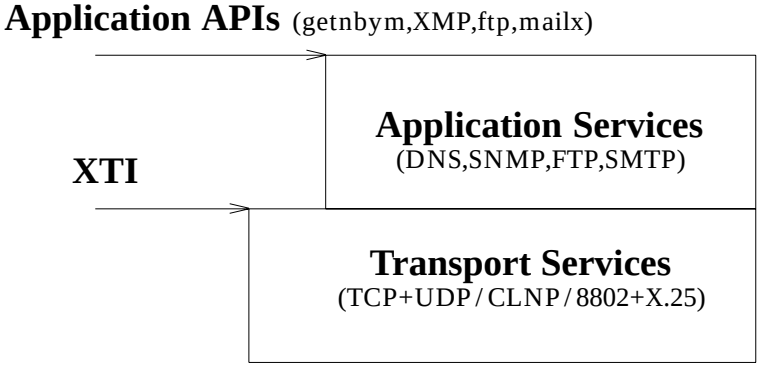


Figure 4-2 TCP/IP Protocol Stack

4.2 RPC Services

The Remote Procedure Call (RPC) Services support the familiar procedure call paradigm in a networked client/server environment. The client makes what looks like a procedure call. The procedure call is translated into network communications by the underlying RPC mechanism. The RPC Services are summarised in Table 4-2.

The RPC Services provide the specification techniques, protocols and runtime environment for supporting the remote procedure call model. The term *RPC* is often broadly used to cover a number of services including the communication protocols, run-time environment, specification techniques, discovery mechanisms and program development tools. In the XDCS Framework it is used more narrowly to cover the protocols, specification techniques and runtime environment. It is included in the Communication Services to provide a common set of services to all higher level services that conform to the client/server model. Two RPC protocols are supported - the RPC protocol from the OSF DCE and the ONC RPC protocol that is primarily used to support NFS (see reference **XNFS**). Associated with each protocol are:

- an interface specification notation - the Interface Definition Language (IDL) or Interface Definition Notation (IDN) - for expressing the end-to-end service contract between the client and the server (End-to-End Contract)
- the binding of this specification notation into one or more specific programming languages (Language Binding)
- an interface for accessing the RPC protocol at runtime (Runtime API)
- a specification for the RPC protocol (Protocol)

- a set of encoding techniques for representing data in a machine-independent fashion (Formats).

The programming interfaces to ONC RPC are supported for interoperability with existing systems described in Chapter 7.

Communication Services - RPCs				
	Programming Interface			
Component	**End-to-End Contract**	**Language Binding**	**Runtime API**	**Protocol & Formats**
DCE RPC	OSF DCE IDL	OSF DCE IDL 'C'	OSF DCE RPC Runtime	OSF DCE RPC Protocol OSF DCE NDR
ONC RPC				X/Open XNFS RPC Protocol X/Open XNFS XDR

Table 4-2 RPC Services

Typically, an application programmer will express the remote procedure calls using the appropriate language binding. The interface specification will then be compiled using an RPC compiler. The RPC compiler expands these remote procedure calls into a set of client and server stubs for performing the operations over the network (see Figure 4-3). The stubs isolate the application from the details of the networking environment. At execution time the stubs will interact with the RPC runtime environment to communicate the service request across the network. The stubs automate the handling of different data formats and the application of the appropriate security controls.

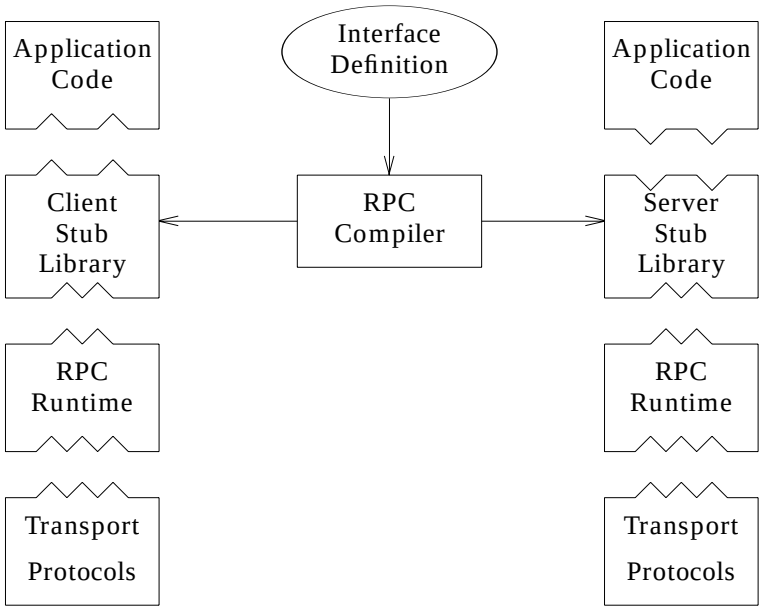


Figure 4-3 RPC Technology Isolates Application from Networking Details

4.3 Peer-to-Peer Conversational Services

Today and more so in the future, programmers are trained to develop distributed applications based on a conversational-like paradigm programming service. This conversational-like paradigm enables a program to tightly control the peer-to-peer interactions which occur between two processes. The X/Open Common Programming Interface for Communications (CPI-C) provides access to peer-to-peer conversational services (see reference **CPI-C**). These services are provided by the OSI Transaction Processing (TP) standard (ISO 10026). As well, CPI-C enables network independence by being able to map across OSI TP ASE as well as other existing peer-to-peer conversational protocols. Machine dependent character representation, authentication security services and primitive directory functions are part of the CPI-C specification.

Communication Services - Peer-to-Peer Conversational Services			
	Programming Interface		
Component	**Language Binding**	**Runtime API**	**Protocol & Formats**
Peer-to-Peer	'C'	X/Open CPI-C	OSI TP

Table 4-3 Peer-to-Peer Conversational Services

Chapter 5

Distribution Services

The components in the Distribution Services layer are shown in Table 5-1. The Distribution Services provide a coherent environment for developing distributed system software. The Distribution Services are composed of the following services:

- Object Management - provides an Object Management Group (OMG)-conformant distributed object management environment that shields the application from the complexities of knowing where an object resides in a network, how it performs its operations or how it stores its information.
- Naming and Directory Services - provide the services for identifying objects in the network and keeping information about these objects. The naming service integrates X.500, DNS, the OSF Cell Directory Service into a cooperative naming scheme. The directory service maintains information about the objects identified with the naming service.
- Time - coordinates a consistent view of time in the network
- Security - protects resources in the network
- Systems Management - provides an object-oriented distributed management framework to handle the complexities of heterogeneous, distributed systems management.

Distribution Services		
Component	**Programming Interface**	**Protocol & Formats**
Object Management	OMG CORBA (IDL, DII)	
Naming		
X.500	X/Open XDS	ISO 9594 (X.500)
DNS	getnbym	IETF RFC 1035
CDS	OSF DCE RPC NSI	OSF DCE CDS Protocol
Directory		
X.500	X/Open XDS	ISO 9594 (X.500)
CDS	X/Open XDS	OSF DCE CDS Protocol
Time	POSIX.1 time, OSF DCE DTS API	OSF DCE DTS Protocol
Security		
Authorization	OSF DCE Security ACLs	
Authentication		IETF RFC <draft> (Kerberos)
Protection	OSF DCE RPC Options	OSF DCE Security Protocols
Systems Management	X/Open XMP API	ISO 9596 (CMIP) IETF RFC 1098 (snmp) X/Open Managed Objects

Note: The OSF DCE CDS protocol is a temporary solution which is expected to be superseded in future versions of DCE by a generic Junction Protocol.

Table 5-1 Distribution Services

5.1 Object Management

The Object Management Services are based on the Object Management Group's (OMG) Common Object Request Broker Architecture (see reference **CORBA**), which is presented in Figure 5-1.

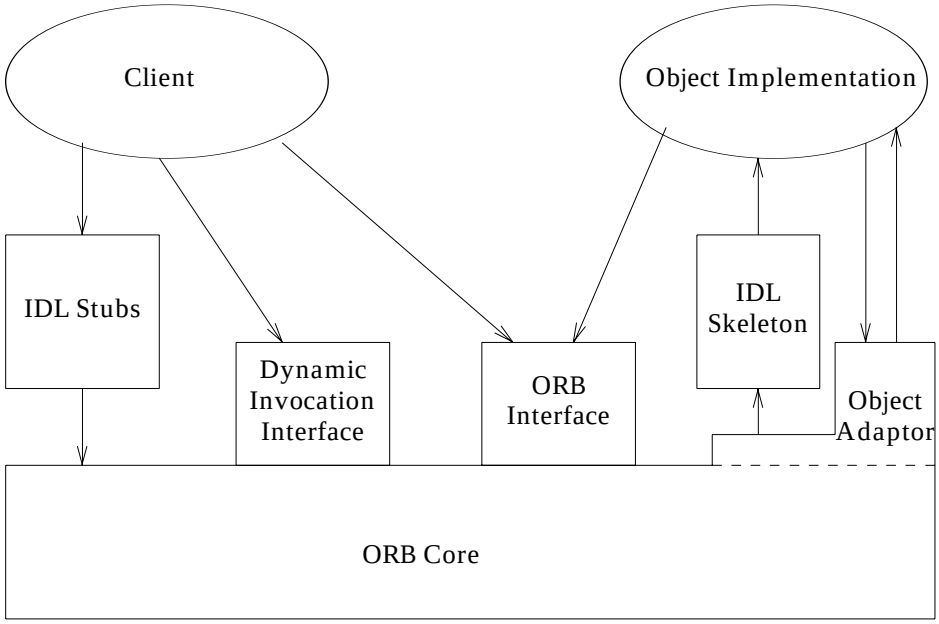


Figure 5-1 CORBA Architecture

The Object Management Services support the interfaces defined by the OMG. Objects are defined using the OMG standard object-oriented Interface Definition Language (IDL, different from the DCE IDL mentioned in Section 4.2 on page 11, see reference **IDL**). These object definitions are compiled using an IDL compiler to produce server IDL skeletons. The IDL has been designed so that it can be compiled into an RPC protocol or some other communication mechanism (see Figure 5-2).

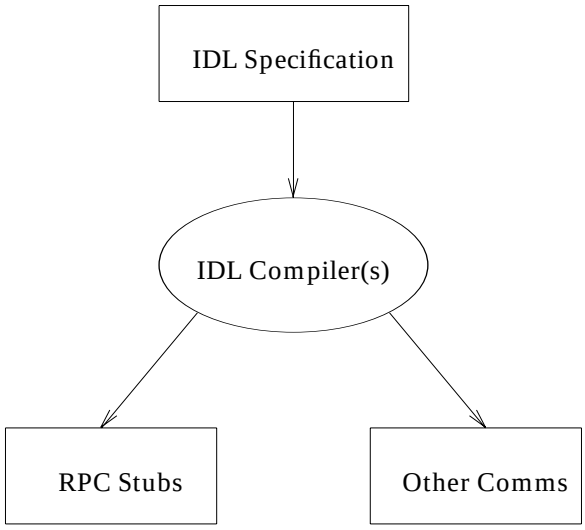


Figure 5-2 Independence of Communication Mechanisms

These IDL skeletons make calls to the interface routines in the actual object implementation. The IDL skeletons shield the application from the details of the underlying distributed object management system. The object adaptor is responsible for dispatching the operation to the object implementation. Since the type of the dispatching is dependent on the nature of the object, there are several possible object adaptors. The XDCS Framework supports the Basic Object Adaptor interfaces in the OMG specification. The Basic Object Adaptor is responsible for:

- generation and interpretation of object references
- authentication of the principal initiating the object request
- activation and deactivation of the object implementation
- activation and deactivation of individual objects
- invocation of methods through skeletons.

The Basic Object Adaptor supports a variety of dispatching techniques.

A client application may request that an object perform an operation through one of two interfaces. The object request may be expressed in IDL. In this case, the application is compiled with an IDL compiler that generates client IDL stubs. This approach shields the application from the details of the underlying object management system. Alternatively, the application can build the object request dynamically using the OMG's Dynamic Invocation Interface (see reference **DII**). This is useful when the application does not know the nature of the object request at compile time (for example, in a desktop manager).

5.2 Naming and Directory

The Naming and Directory Services provide the means for identifying objects in the network and for discovering information about the objects. These are two distinct, but related services:

- A naming service is the means by which objects are referenced in the network. A naming system can be a standalone service (for example, the Internet Domain Name Service), but it is often integrated with other services such as the file system, an RPC service or a directory service.
- A directory service provides information about objects in the network. The directory maintains an information base of attributes (for example, name, address, creation date, etc.) associated with the objects. Objects can be referenced with a naming service or by a set of defining attributes. The directory service will return requested attributes pertaining to the referenced object(s).

For referencing large-grained objects in an enterprise environment, XDCS has adopted the DCE naming and directory model. The DCE model balances the need for rapid access to information about local objects with the need to integrate with public global directory services. The DCE naming model is supported by a two-tiered directory structure (see Figure 5-3).

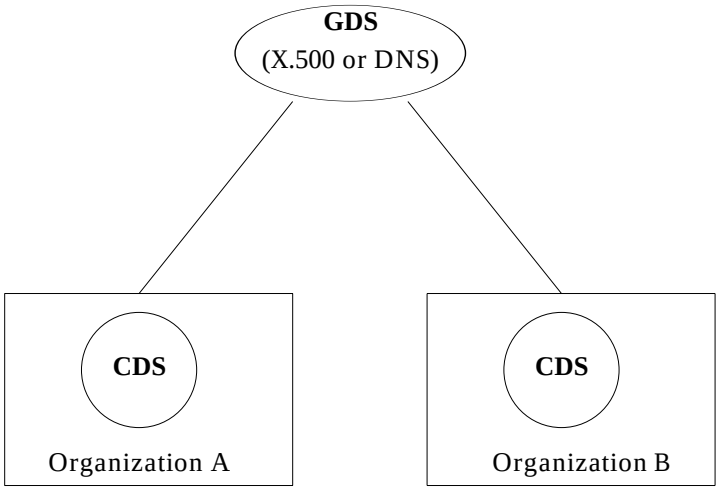


Figure 5-3 Naming Architecture

The Cell Directory Service (CDS) is used to keep information about objects in the local "cell". A cell is a set of machines that work together and are administered as a unit. A cell corresponds to the notion of a workgroup and may support a department or similar unit within an organization. The Global Directory Service (GDS) is used to keep information about cells so that objects in one cell can reference objects in another cell. The global directory can either be X.500 or the Internet Domain Name Service (DNS). The naming model supports a hierarchical namespace (for example, as shown in Figure 5-4).

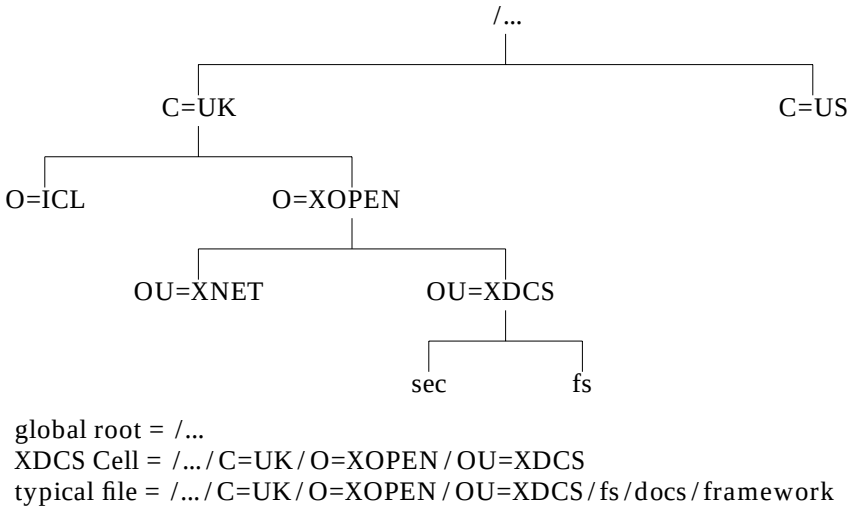


Figure 5-4 Sample Name Space

The name service understands compound names consisting of a global and a local part. For example, in the name **/.../C=UK/O=XOPEN/OU=XDCS/fs/docs/framework** the global part is

/.../C=UK/O=XOPEN/OU=XDCS

and the local part is

/fs/docs/framework.

The ‘‘**/...**’’ prefix indicates that this is a global name. Names in the local cell are indicated with a ‘‘**/.:**’’ prefix: for example,

/.:/fs/docs/framework.

Applications can make enquiries into the directory service through the X/Open XDS interface (see reference **XDS**), as shown in Figure 5-5.

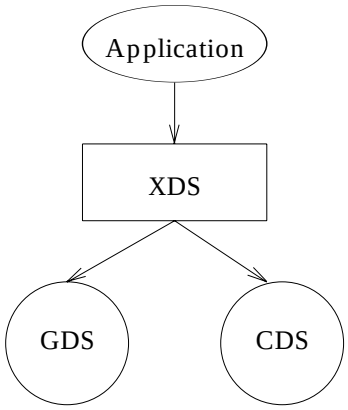


Figure 5-5 XDCS Directory API

5.3 Time

Audit trails, fault recovery and other applications require an accurate ordering of events. Event ordering is complicated in a distributed environment by the varying degree of inaccuracy of the clocks in the network. The Distributed Time Service (DTS) provides the means to synchronize the clocks in the distributed system within a tolerable degree of accuracy.

The DTS is based on two principals - interval time stamps and periodic synchronization. Since every clock has a degree of inaccuracy, time can never be known as a precise quantity. Instead, time can be known within an interval that is determined by the clock drift since the last synchronization with a more accurate time source. The DTS represents time with an interval time stamp that includes the putative time and an inaccuracy factor. Time synchronization is used to reduce the inaccuracy interval.

The DTS deals with three types of systems. The time provider has direct access to a more reliable provider of accurate time (either it receives the international standard Coordinated Universal Time (UTC) signal via radio or phone or it has access to an acceptable arbiter of time such as another cell or an NTP time provider). The local time servers frequently synchronize with the time provider to maintain tight tolerances on time. The majority of the systems are clients. The time clerks on the clients periodically synchronize with the local time servers.

An application that needs to know the time can either use the POSIX.1 **time** to get a simple scalar with an unknown inaccuracy or use the DTS API to get the interval timestamp.

5.4 Security

The Security Services provide the means to protect resources in the network. Security is applicable to every layer of the architecture. The Security Services in the Distribution Services layer provide the basic mechanisms that are used to provide security for all the services in the framework. The Security Services are based on the DCE security model. There are three primary services:

- Authentication - verification of the identity of clients and servers
- Authorization - determination of the permissions of a client to perform an action on an object
- Protection - ensuring the privacy and integrity of communication between the client and the server.

The diagram in Figure 5-6 shows how these services support secure client/server communication. The security server supports the mutual authentication of the client and the server using the Kerberos (see reference **KERB**) secret key authentication scheme from MIT project Athena. The Secure RPC carries an encrypted ticket that contains a Privilege Attribute Certificate (PAC) giving the client's authorization credentials and a session key for protecting client/server communications. The degree of communication protection varies according to application requirements - for example, integrity (protection against modification, implemented via a cryptographic checksum), and privacy or confidentiality (protection against disclosure, implemented via encryption of whole messages). When the server receives the client's request, it uses the Access Control List (ACL) Facility to compare the client's PAC against the authorization information in the Access Control List associated with the object. The ACL Facility is based on the draft IEEE P1003.6 Access Control List standard (see reference **ACL**) with extensions to support distributed computing requirements.

The Secure RPC mechanism shields the application from having to know the details of the security services. The application selects the desired level of security and the RPC runtime performs the appropriate authentication and protection. The ACL Management Facility handles the details of ACL manipulation. The only APIs to the security services are an option in the RPC Interface to specify the level of security and the API to the ACL Facility.

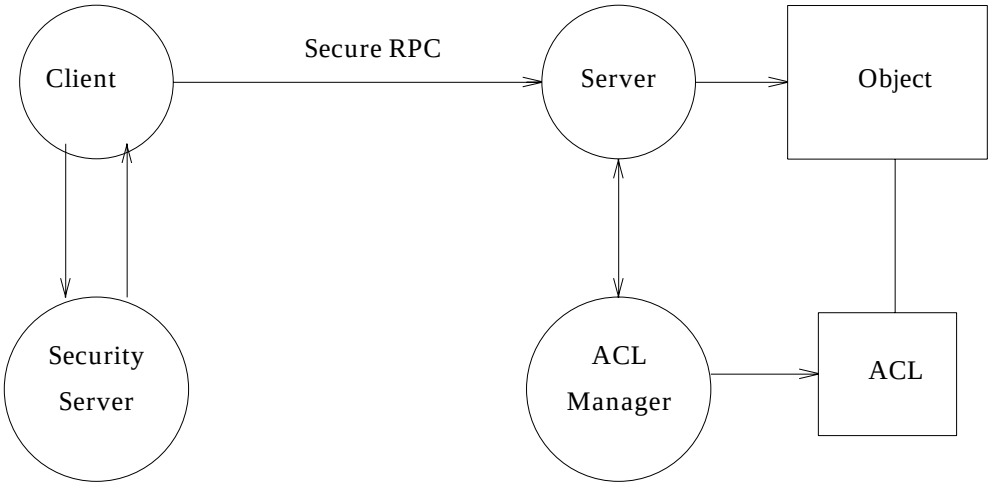


Figure 5-6 Secure RPC

5.5 Systems Management

Manageability is a quality that applies to all services in the XDCS Framework. The Systems Management Framework Services in the Distribution Services provide a common set of mechanisms for supporting distributed management applications. The purpose of these services is to enable Management Applications to manage the diverse Managed Resources in the network. The X/Open System Management (XSM) Reference Model (see reference **XSM**) models this interaction as the interaction between a set of Management Functions that perform useful tasks for administrators and a set of Management Agents that control resources in the network - see Figure 5-7.

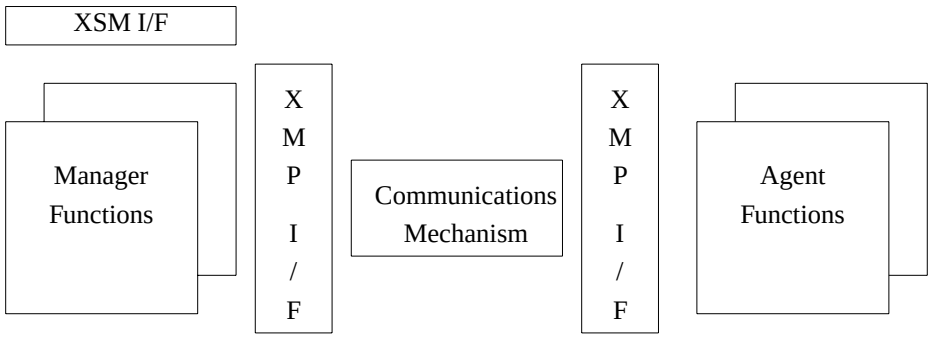


Figure 5-7 XSM Reference Model

The X/Open Management Protocol Interface (see reference **XMP**) currently defines the interface to the Communication Services. This interface allows a Management Function to communicate with a Management Agent independently of the underlying management protocol. The XMP interface works over either the OSI Common Management Information Protocol (CMIP) or the TCP/IP Simple Network Management Protocol (SNMP).

Chapter 6

Application Services

The components in the Application Services are shown in Table 6-1. The Application Services provide the distributed system software to support the development of distributed applications. The current version of XDCS contains key Application Services. This is not an exhaustive list. Other Application Services may be added.

Application Services		
Component	**Programming Interface**	**Protocol & Formats**
Distributed Files		
File Sharing	POSIX.1	X/Open XNFS, OSF DCE DFS
File Transfer	X/Open XFTAM API	ISO 8571(FTAM)
Messaging	X/Open XMHS (MT, MA)	CCITT X.400 (ISO 10021)
Transaction Processing	X/Open TX, XA	ISO 10026 (TP)
Data Management		
Record Management	X/Open ISAM	N/A
Relational	ISO SQL	N/A
Networked	X/Open SQL Access	ISO 9579 (RDA)
Windowing	X/Open Xlib, Xt Intrinsics	'X' Consortium X11R5

Table 6-1 Application Services

6.1 File Services

The File Services provide transparent access to files in the network. The application accesses files using the POSIX file system interface. Most open systems today provide transparent file access using the Network File System (NFS) protocol as documented in XNFS (see reference **XNFS**). NFS provides access to files in a wide variety of environments. In order to provide heterogeneous file sharing, NFS makes some small compromises on the P1003.1 file semantics. These semantics along with additional error codes for file sharing in a network environment are specified in the draft P1003.8 Transparent File Access specification (see reference **TFA**). The OSF DCE Distributed File System (DFS) supports the full POSIX P1003.1 File Semantics in a network environment. The DCE DFS also supports a single file system view, replicated files and client-caching in a wide-area network.

In addition, the framework provides an interface to the OSI File Transfer, Access and Management (FTAM) protocol (see references **FTAM** and **XFTAM**) for providing file transfer and interactive file access to files on an OSI-conformant system.

6.2 Messaging Services

The messaging services provide the ability to send and receive arbitrary electronic messages using the store and forward OSI Message Handling System (MHS or X.400, see reference **X.400**) standards using the XMHS interfaces (see reference **XMHS**).

XMHS defines two interfaces. The Message Access (MA) interface provides message delivery and retrieval services to general purpose applications. In the MHS architecture, it is at the interface between the User Agent and the Message Transfer Agent. The interface can be used for developing User Agents that provide human users with a convenient environment for sending and receiving electronic mail. The Message Transfer (MT) interface is at a lower level. It provides access to the basic X.400 service that allows one Message Transfer Agent to send message to another Message Transfer Agent in the Message Transfer System. One use of this interface is in the implementation of mail gateways that transfer mail between a proprietary mail system and an X.400 Message Handling System.

6.3 Data Management

The Data Management Services provide applications with access to structured data in the distributed environment. ISO standard SQL (see reference **SQL**) is the primary interface to relational databases. SQL Access (see reference **SQLACC**) extends this interface to access databases over a network. ISAM (Indexed Sequential Access Method, see reference **ISAM**) is utilized to access information from a high performance record manager.

6.4 Transaction Processing

The Distributed Transaction Processing Services enable an application to perform multiple distributed services under atomic transaction control. The DTP Services follow the X/Open Distributed Transaction Processing model (see Figure 6-1).

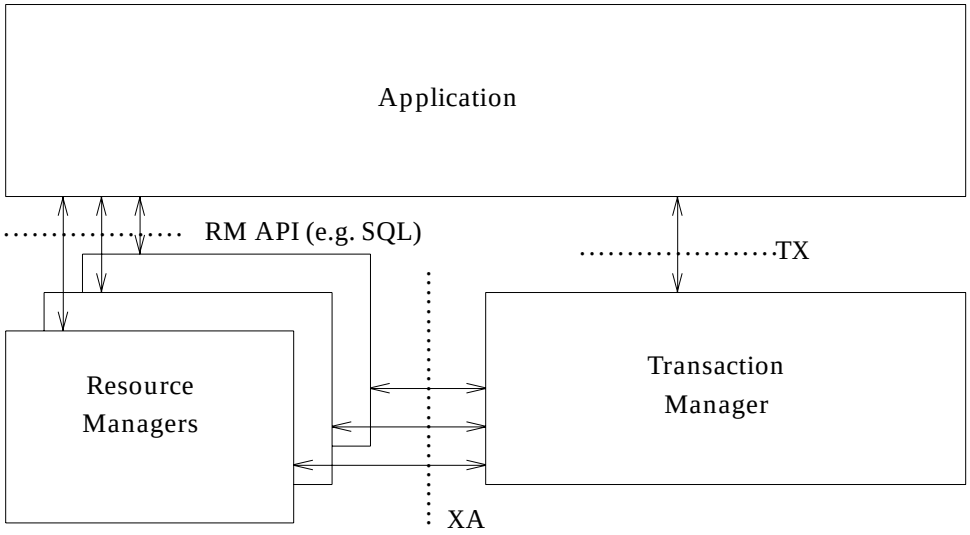


Figure 6-1 XTP Architecture

The application interacts with a Transaction Manager through the TX interface. The Transaction Manager is responsible for coordinating distributed transactions across the network. The application uses the TX interface to bracket transactions. The Resource Managers perform the actions that compose the transactions. A Resource Manager may be a relational database management system accessed through SQL, a transactional file system, an audit log, or other service.

The Transaction Manager coordinates the Resource Managers involved in the transaction, through the XA interface (see reference **XA**). The Transaction Manager first asks all the resource managers to precommit their actions. A precommit means that the Resource Manager has reserved all the resources so that it guarantees that it can perform the action if asked. Once all of the Resource Managers have precommitted, the Transaction Manager then commits the transaction by asking each of the Resource Managers to perform the precommitted action. If one of the Resource Managers cannot precommit, the transaction is aborted.

6.5 Windowing

The Windowing Services allow applications to interact with a human user in the network through a graphic display screen and a keyboard and pointing device. The Windowing Services are based on the X Window System (see reference **X11**) from MIT Project Athena.

The X Window System supports a client/server model. The display server controls the graphical display and keeps track of user input through the keyboard and pointing device. The client is a network-based application that interacts with the human user through the display server. The client and server communicate using the X protocol.

The X Window System allows the application to manipulate windows on the screen, associate arbitrary data with a window, access fonts and colors, perform general graphical output and obtain input from the keyboard and pointing devices.

The Windowing Services support two levels of interface. The Xlib interface allows the application to build X Protocol Messages. The Xt Intrinsics interface provides the basic constructs to support the creation and use of user interface objects. These interfaces provide the foundations for implementing a variety of higher level Graphical User Interfaces (GUIs). The X/Open interfaces to these services are based on the specifications coming from the "X' Consortium.

Chapter 7

Interoperability with Existing Systems

The components of the Interoperability Services are shown in Table 7-1. The Interoperability Services are targeted at four major environments.

- The Common Programming Interface for Communications (CPI-C) interface (see reference **CPI-C**) and the Logical Unit 6.2 (LU 6.2) protocol (see reference **LU6.2**) are used to communicate to applications in an IBM Systems Application Architecture (SAA) environment.
- The Server Message Block (SMB) protocols from Microsoft's LAN Manager (see reference **XSMB**) and PC NFS are used to provide file and print sharing for MS DOS-based personal computers.
- The Internet Protocol Suite (IPS) Services are used to provide interoperability with a wide variety of existing systems that support the TCP/IP family of protocols.
- The ONC RPC services are used to provide interoperability with a variety of existing systems that support the ONC RPC protocols from Sun Microsystems.

Interoperability with Existing Systems		
Component	**Programming Interface**	**Protocol & Formats**
Mainframe	X/Open CPI-C	IBM LU 6.2
PC Networks LAN Manager PC NFS	 X/Open IPC for SMB X/Open XSH	 X/Open SMB X/Open (PC) NFS
IPS File Transfer Messaging Terminal Emulation	 ftp mailx telnet command	 IETF RFC 959 (ftp) IETF RFC 821 (smtp) IETF RFC 854 (telnet)
ONC RPC	Sun ONC RPC IDL & Run Time	X/Open XNFS RPC & XDR

Table 7-1 Interoperability

Glossary

ACL
Access Control List. Data associated with an object that specifies the authorization policies that govern the operations that may be invoked on the object by authenticated subjects.

ACSE
Association Control Service Element. The OSI application layer service element responsible for association establishment and release.

Adaptor
See object adaptor.

ANSA
Advanced Networked Systems Architecture. ANSA supports the design, implementation, operation, and evolution of distributed information processing systems where the different components that make up the system come from different vendors.

API
Application Programming Interface. A set of source-level interfaces, usually procedure calls, used by application developers.

Application Services
Provide the enabling distributed system software within the XDCS Framework to support the development of distributed applications.

CCITT
International Telegraph and Telephone Consultative Committee. A standards organization comprised of national Post, Telephone, and Telegraph (PTT) administrations.

CD
Committee Draft. A designation for standards that are in the early stages of development.

CDS
The Cell Directory Service of OSF DCE. The Cell Directory Service stores names and attributes of resources located in aDCE cell. It is optimized for local access and may be replicated for high availability.

Cell
A single administrative and security domain in OSF DCE. A cell is able to operate independently; it provides all required servers for the clients in the cell. Intra-cell communications are optimized.

Client
A relation on objects. Object A is a client of object B if A uses the services of B.

CLNP
Connectionless-mode network protocol.

CLTP
Connectionless-mode transport protocol.

CMIP
Common Management Information Protocol. The OSI application layer protocol that supports CMIS network management services.

CMIS
Common Management Information Service. The OSI application layer service used for the management of networks.

Communication Services
Provide services within the XDCS Framework that allow applications to communicate reliably across the network independent of the underlying network topology, network protocols or data representations.

CORBA
Common Object Request Broker Architecture. CORBA defines a framework for different ORB implementations to provide common ORB services and interfaces to support portable clients and implementations of objects.

COTP
Connection-oriented transport protocol.

CPI-C
Common Programming Interface for Communications. An interface, part of IBM's Systems Application Architecture (SAA), to LU 6.2 services.

CSMA/CD
Carrier Sense Multiple Access / Collision Detection.

DCE
See OSF DCE.

DFS
Distributed File System of OSF DCE. A file service that joins the local file systems of server file server machines,making the files equally available to all DFS client machines.

DII
Dynamic Invocation Interface. DII supports the needs of applications that do not know the nature of the objectrequest at compile time. It is part of CORBA.

Directory Service
A specialized information storage service integrated with a naming system.

Distribution Services
Provide a consistent applications environment within the XDCS Framework across aset of heterogeneous machines.

DNS
Internet Domain Name System.

DTP
Distributed Transaction Processing.

DTS
Distributed Time Services. DTS is a DCE service that provides a way to periodically synchronize the clocks on different hosts in a distributed system. It can also keep that synchronized notion of time reasonably close to UTC.

FTAM
File Transfer, Access and Management. The OSI application layer service that supports transfer, access and management of files.

FTP
File Transfer Protocol. FTP is part of the Internet Protocol Suite.

GDS
Global Directory Service.

GSSAPI
The Generalized Security Services API.

GUI
Graphical User Interface. A form of communication between users and computers that uses graphics-oriented software such as windows, menus and icons to facilitate the interaction.

IDL
Interface Definition Language. A high-level declarative language that provides the syntax for interface definitions. Several IDLs are referenced in the XDCS framework.

IDN
Interface Definition Notation. A language-independent notation for specifying interface definitions for ISO RPC.

IEEE
Institute of Electrical and Electronic Engineers.

IETF
The Internet Engineering Task Force. A task force of the Internet Activities Board charged with solving the short-term needs of the Internet.

Interface Definition Language
The standard language for interface definition.

Internet
A large collection of connected networks, primarily in the United States, running the Internet Protocol Suite.

IP
Internet Protocol. The network protocol offering a connectionless-mode network service in the Internet Protocol Suite.

IPC
Inter-Process Communication.

IPS
Internet Protocol Suite. A family of network protocols defined by the U.S. Department of Defense (DoD).

IRDS
Information Resource Dictionary Systems.

ISAM
Indexed Sequential Access Method. A high performance, record-oriented file manager.

ISO
International Organization for Standardization. The international organization that develops, among others, OSI standards.

LAN
Local Area Network.

LU 6.2
An SNA communication protocol released by IBM in 1982; also known as Advanced Program-to-Program Communication (APPC)

MA
The X/Open Message Access Interface.

MHS
Message Handling Service. The OSI application layer service that supports message distribution.

MIB
Management Information Base.

MT
X/Open Message Transfer Interface.

NDR
Network Data Representation. The set of encoding rules used by much of the DCE for converting application data to and from different local data representations.

NFS
Network File System. An operating system-independent service that allows user to mount directories across a network, and then to treat those directories as if they are local.

NSI
Name Service Interface. These name service access a directory service, such as CDS, for DCE RPC applications.

NTP
Network Time Protocol.

Object
An object provides a behavior defined by an interface which consists of a set of operations. An object hides the representation of its state and the implementation of its operations. An object is an instance of an object type which defines the object's interface..gT Object Adapter The ORB component that generates and interprets object references, invokes methods, and performs activation and deactivation of an object implementation. There may be multiple adapters within an ORB to support different object implementations.

ODP
Open Distributed Processing.

OMG
Object Management Group. An industry consortium working to define standards for object management.

ONC
Sun Microsystem's Open Network Computing architecture.

Operating System Services
Provide an environment for running distributed software within the XDCS Framework on each node in the network.

ORB
Object Request Broker. An object request broker provides the means by which objects make and receive requests and responses.

OSF
Open Software Foundation, Inc. The industry consortium developing OSF DCE.

OSF DCE
OSF's Distributed Computing Environment.

OSI
Open Systems Interconnection. A set of international standards, developed by the International Organization for Standardization (ISO), including the seven-layer reference model for data communication systems, and service/protocol recommendations.

PAC
Privilege Attribute Certificate.

PC NFS
The NFS protocol for PCs as documented by X/Open. The protocol supports file and print services for personal computers.

POSIX
The Institute of Electrical and Electronics Engineers' (IEEE) and ISO standard for a portable operating systems environment. The POSIX interfaces are based on the UNIX Operating System.

P1003
The IEEE project number assigned to the POSIX group of standards.

RDA
Remote Data Access. The OSI application layer service that supports data access by an application program to a database.

Resource Manager
A resource manager provides access to shared resources such as databases or file access systems.

RFC
Request For Comment. The document series describing the Internet Protocol Suite and related experiments.

RFP
Request For Proposal.

RM
Resource Manager.

RPC
Remote procedure call. The mechanism for passing values and control across a network to a remote procedure.

SAA
Systems Application Architecture. SAA is IBM's collection of standards that define a software environment that is supported across IBM's principal hardware systems.

Server
A relation on objects. Object A is a server of object B if A provides services to B.

SIC
Semantic Integrity Constraint.

SICM
Semantic Integrity Constraint Manager.

SMB
Server Message Block. The SMB protocol provides file and print sharing facilities.

SMTP
Simple Message Transfer Protocol. SMTP is part of the Internet Protocol Suite.

SNA
Systems Network Architecture. SNA was released originally by IBM in 1974, and enhanced many times. SNA defines the predominant communication methods used by IBM computers. There are many products from IBM and other vendors that implement different portions of the SNA architecture.

SNMP
Simple Network Management Protocol. SNMP is part of the Internet Protocol Suite.

SQL
Structured Query Language.

Stub
A form of adaptor which provides a proxy for a server at the client side or a proxy for a client at the server side of a distributed application.

TAI
International Atomic Time.

TCP
Transmission Control Protocol. The Transport protocol offering a connection-oriented transport service in the Internet Protocol Suite.

TCP/IP
Transmission Control Protocol / Internet Protocol.

TFA
Transparent File Access. A standard being developed by POSIX.

TP
Transaction Processing.

TP Monitor
A distributed system of objects that comprise a transaction manager together with facilities for TP-oriented communications, transaction scheduling, administration, and naming.

Transaction
A sequence of operations on one or more objects that either execute to completion, or have no effect on the states of any of the objects.

Transaction Manager
A subsystem of objects within a TP Monitor which interact with resource managers (e.g., databases) to supply transaction semantics to sequences of distributed operations.

TX
The X/Open Transaction Management Interface. The API by which an application program calls a transaction manager to delimit global transactions, retrieve information regarding transaction status and provide other supporting functions.

UDP
User Datagram Protocol. The Transport protocol offering a connectionless-mode transport service in the Internet Protocol Suite.

UDP/IP
User Datagram Protocol / Internet Protocol.

UI-ATLAS
Unix International's Distributed Computing Framework.

UI
Unix International, Inc. The industry consortium developing UI-ATLAS specifications.

UTC
Coordinated Universal Time. Time standards of the world are based on International Atomic Time (TAI), which is maintained using multiple cesium-beam clocks. The Bureau International de l'Heure uses astronomical observations provided by the U.S. Naval Observatory and other observatories to determine corrections to TAI for small changes in the mean solar rotation period of the Earth, resulting in UTC.

XA
The X/Open DTP bidirectional, system-level interface between a transaction manager and a resource manager.

XAP
The X/Open ACSE/Presentation Interface. This API may be used for connection management in an OSI environment.

XDCS
X/Open Distributed Computing Services.

XDR
External Data Representation. XDR provides a common way for describing data types and for representing data types in a canonical format across a network. XDR may be used with NFS and TIRPC.

XDS
The X/Open Directory Service API. The API is used for accessing information managed by a directory service.

XFTAM
The X/Open FTAM APIs.

Xlib
The X/Open Xlib interface allows the application to build X protocol messages.

XMHS
The X/Open Message Handling System set of APIs. XMHS includes XMT and XMA.

XMP
X/Open Management Protocol Interface. The XMP API defines a protocol-independent API to the OSI CMIS and Internet SNMP management information services.

XNFS
The NFS protocol as documented by X/Open.

XPG
The X/Open Portability Guide.

XSM
The X/Open System Management reference model.

XSH
X/Open System Interface, Utilities, Commands and Headers specification.

Xt Intrinsics
The X/Open X-Windows Intrinsics interface provides the basic constructs to support the creation and use of user interface objects.

XTI
X/Open Transport Interface. A transport-independent API.

X11R5
Release 5 of the X-Windows protocol.

X.25
A point-to-point connection-oriented network facility.

X.400
The CCITT designation for the OSI Message Handling Systems standard.

X.500
The CCITT designation for the OSI public directory service.

Index