Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: **Cambridge (01223) 515010**
INTERNATIONAL: **+44 1223 515010**
FAX: **+44 1223 359779**
E-MAIL: **apm@ansa.co.uk**

Training

ANSAwise - Comparing CORBA and DCE

Chris Mayers

Abstract

Needing to select a environment within which to procure and build open distributed systems, organizations find it difficult to compare their features and benefits. Two important standards are CORBA and DCE.

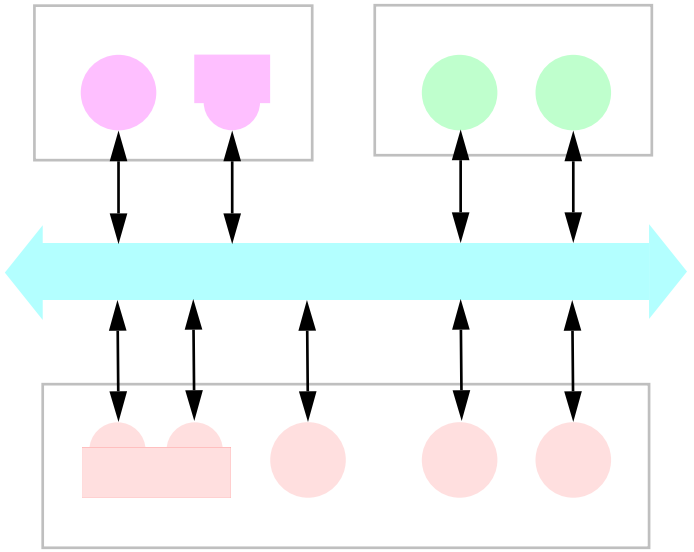
This module of the ANSAwise training programme compares and contrasts the OMG's Common Object Request Broker Architecture (CORBA), and the OSF's Distributed Computing Environment (DCE), both in the way they are standardized, and also in the overall content and structure of their environments.

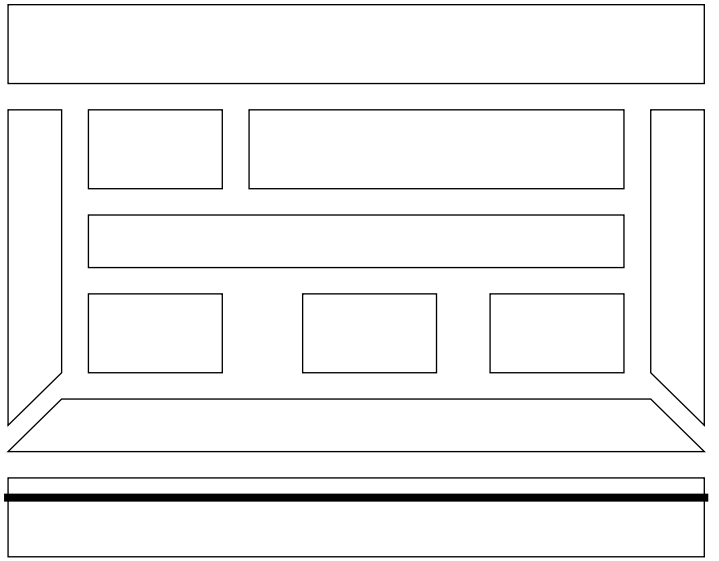
[This variant of APM.1322 does not give an overview of CORBA since this is done in an earlier module.]

APM.1745.01 **Approved** 2nd April 1996
Briefing Note

Distribution:
Supersedes:
Superseded by:

Comparing CORBA and DCE



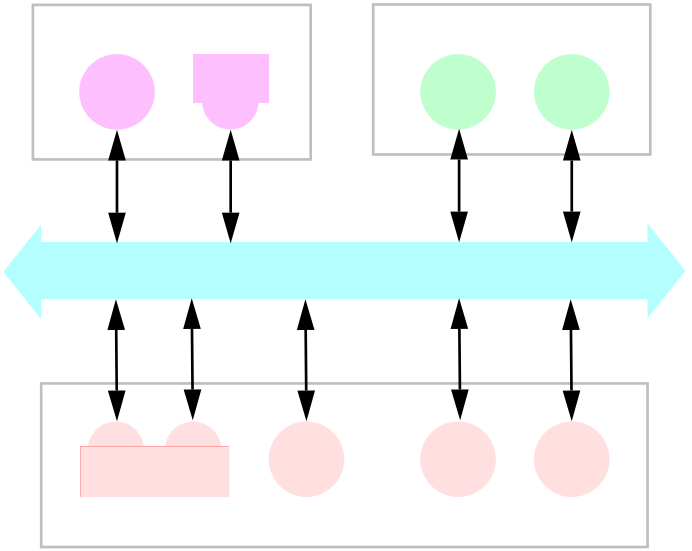


In this session

- **Describe the key features one open environment for distributed computing**
  - **DCE (Distributed Computing Environment)**
- **Compare and contrast this with another such environment**
  - **CORBA (Common Object Request Broker Architecture)**
- **Indicate their place in your distributed systems strategy**

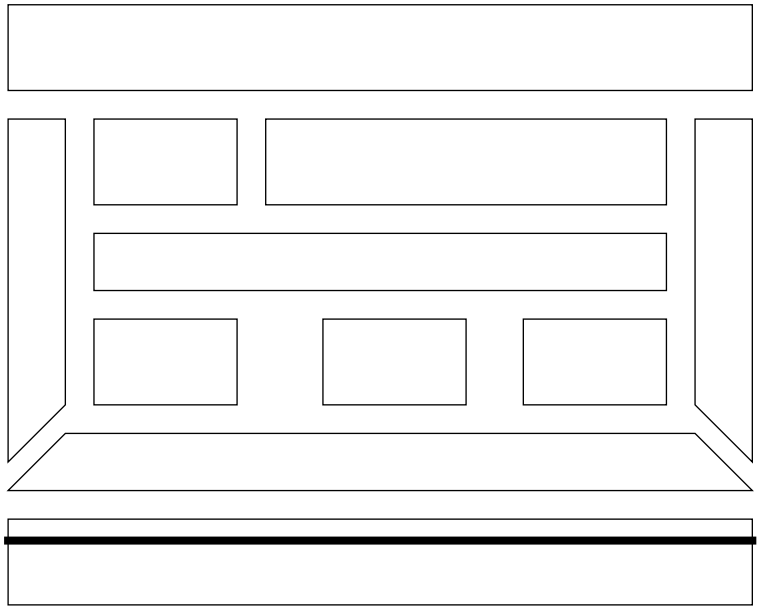
CORBA
(Common Object Request Broker Architecture)

- **Architecture standardized by the Object Management Group (OMG)**



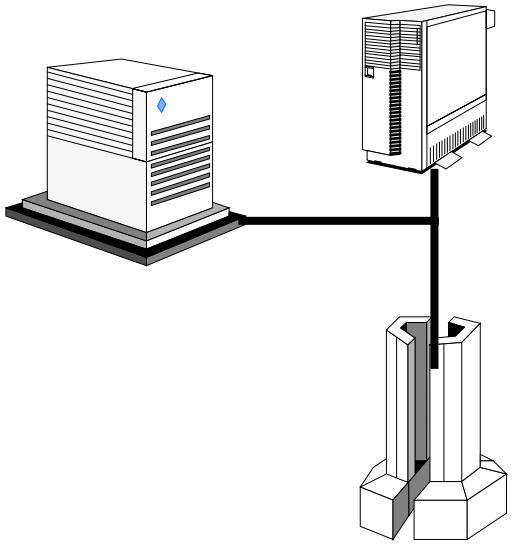
DCE (Distributed Computing Environment)

- **Architecture produced by the Open Software Foundation (OSF)**



The OSF's Focus

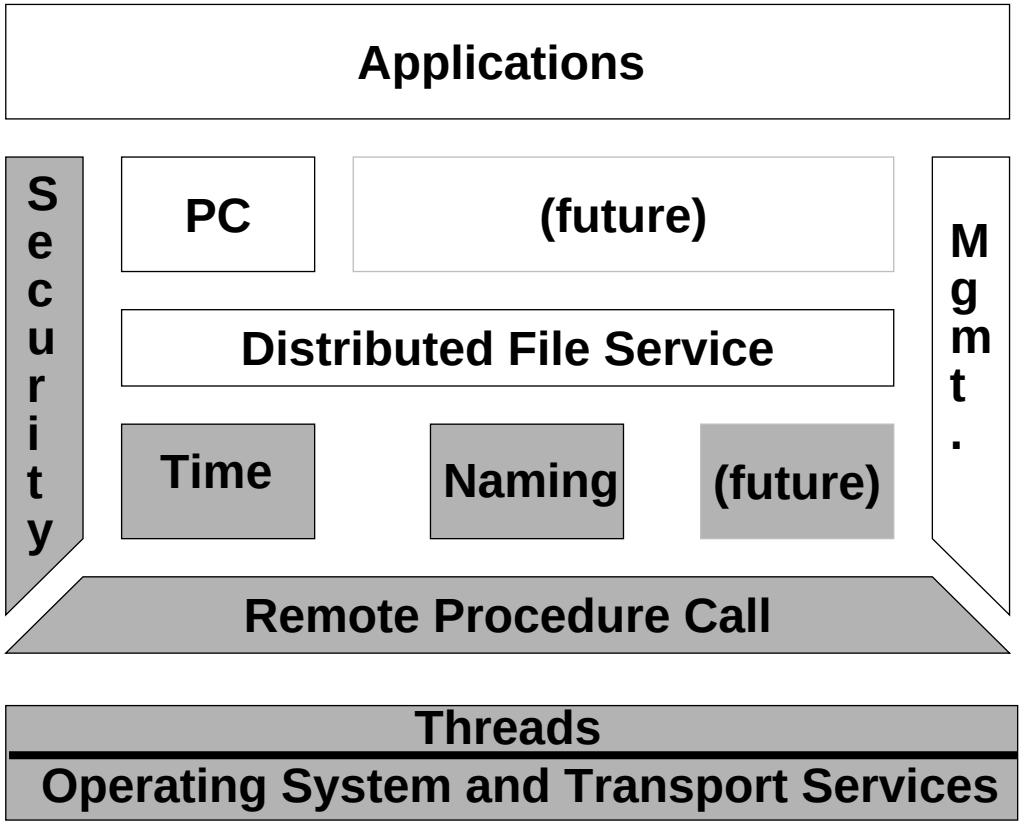
- **Interoperability**



The OSF's Method of Work

- **Obtain technology by soliciting offerings**
- **Integrate it**
- **License it to vendors**

The DCE Component Architecture



DCE Services

- Uses a layered model
- Fundamental Services are explicitly used by applications
  - for example, Distributed Time Services
- Data-sharing services are integrated into the operating system
  - for example, PC file and printer service
- The ‘secure core’ services are required components

DCE Origins - infrastructure

- **Threads**
  - **Digital's CMA (Concert Multithread Architecture), and POSIX pthreads**
- **Remote Procedure Call (RPC)**
  - **Hewlett Packard's Network Computing Architecture NCS 2.0**
- **Security**
  - **MIT's Kerberos, plus other services from Hewlett Packard**
- **Diskless support**
  - **Hewlett Packard**

DCE Origins - services

- **Time Service**
  - **Digital's DECdts**
- **Directory Service**
  - **Digital's DECdns, and Siemens-Nixdorf's DIR-X**
- **Distributed File System**
  - **Transarc's AFS (Andrew File System)**

DCE Cells

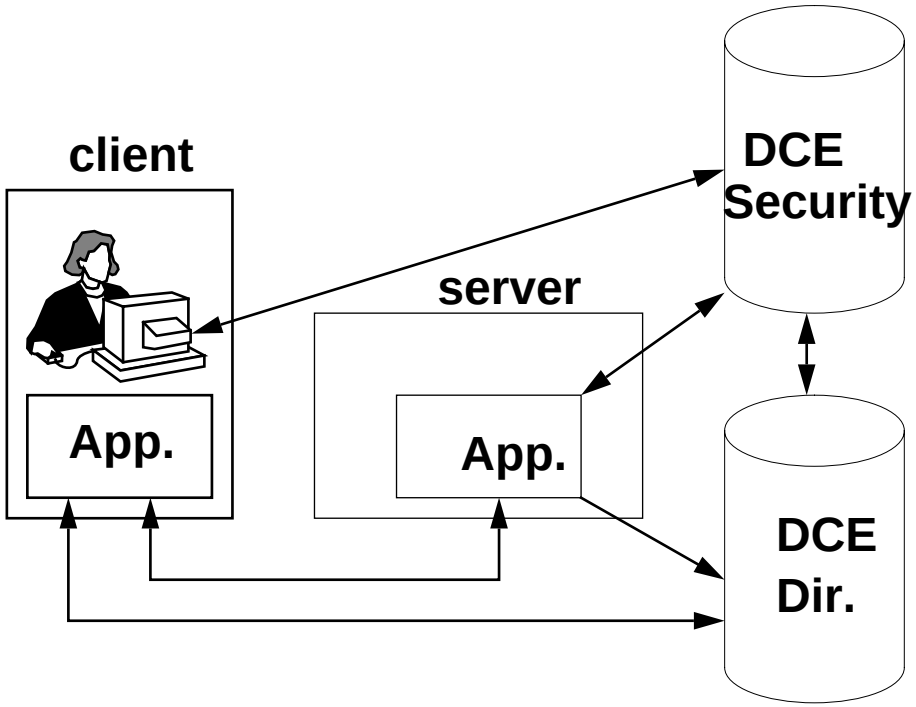
- DCE systems are organized into administrative domains called cells
- A machine can only be in one cell
- Resources are registered in cells
- Cells are intended to support up to thousands of machines - or more

DCE Cell Organization Considerations

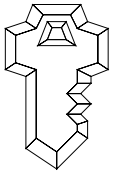
- **Communications**
  - Speed and cost of inter-cell and intra-cell links
- **Legal and regulatory**
  - storage of information and access to it
- **Population**
  - campus or territory
- **Business**
  - integration with other systems, and communication with customers
- **System administration**
  - management and operations centres, and location of support staff

The DCE Environment

- **Clients and servers interact with the ‘core services’**



DCE Security Services



- **Based on MIT Project Athena's Kerberos technology (Version 5)...**
- **... and POSIX 1003.6 (Draft 12) Access Control Lists**
- **The DCE security protocols are complex, but even the programmer need not see them**
  - **the security services are the interface, not the protocols**

DCE Security Components

- **Authentication Service**
  - allows a process to verify the identity of another process
- **Authorization (Privilege) Service**
  - allows a server to determine whether client access should be granted to a resource
- **Registry Service**
  - maintains the DCE security database
- **Access Control List Facility**
  - allows users to grant and revoke access to resources they own
- **Login Facility**
  - authenticates a user to the security service by means of a password

Using DCE Security

- End-users use the Login Facility
  - and probably the Access Control List Facility
- Administrators use the Registry service
  - for creating user accounts
  - for cross-cell authentication, between clients and servers in different cells
- Administrators control security servers
  - including controlling the replication of security data
- Administrators control local machine access

Distributed Security Is Mutual

- Servers must protect themselves against clients
- Clients must protect themselves against servers
- Client applications do not need to use the security services directly
  - typically, they just use Authenticated RPC
- Server applications use Authenticated RPC too
  - and also Access Control Lists to control client access to their objects

Authenticated RPC Options

- **Authentication service**
  - No authentication
  - Secret Key
- **Protection level**
  - Beginning of RPC session only
  - Message/packet integrity
  - Encryption
- **Authorization service**
  - Uncertified
  - Certified

Authentication Responsibility

- Authentication is a shared responsibility...

	Server Preference No Authentication	Server Preference Authentication
Client Preference No Authentication	Unauthenticated	Unauthenticated
Client Preference Authentication	(Fails)	Authenticated

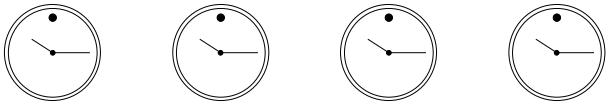
- ... but servers must beware!

- Servers must check client preference, if they wish authentication

Security issues with DCE 1.0

- Administration is tiresome
- Many vendors have non-integrated login
  - must log in to operating system and DCE separately
- Security policy constraints
  - trivial passwords allowed even under the strictest policy
  - no password expiry
  - unlimited login attempts
  - no auditing
  - only 32 ACL permissions
- DCE 1.1 has improved administration, and security enhancements

Distributed Time Service (DTS)



- Keeps machine clocks approximately synchronized
  - without DTS, drift could be as much as 8 seconds per day
- Can interwork with the Internet Network Time Protocol (NTP)
  - NTP and DTS servers can provide time to each other
- Programmers unlikely to interact directly with DTS
  - but must be prepared to handle approximate time if they do...
  - ... with overlapping time intervals, neither is ‘earlier’ than the other

How DTS Keeps Time

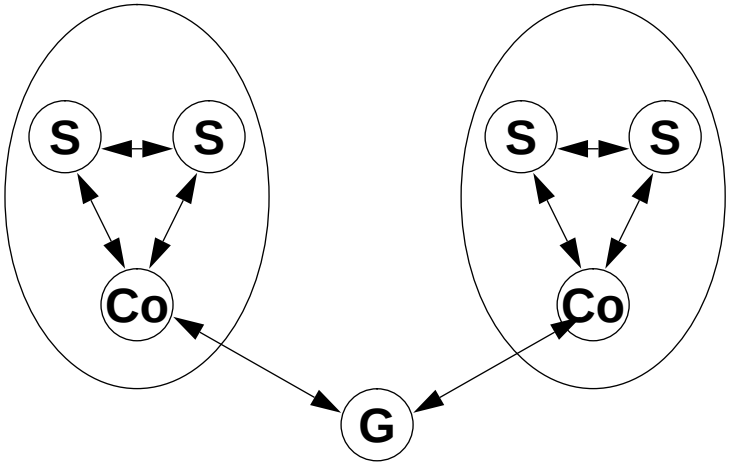
- DTS keeps track of time in software
  - it does not alter the system hardware clock
- DTS adjusts the system software clock
  - by adjusting the clock tick increment in software until the correct time is reached
- DTS adjusts one time unit of error in 100 time units

DTS Components

- **Time Clerk**
  - runs on client
  - periodically synchronizes with Time Servers
- **Time Server**
  - typically 3 time servers per LAN in each DCE cell
  - synchronizes with the other time servers

Time Synchronization between LANs

- Each LAN can have a Courier Time Server...



- ...synchronized with a Global Time Server

Time Provider interface

- Time Servers are responsible for synchronization
- The actual time source comes from an External Time Provider
  - a hardware source
  - the system administrator

Portability in DCE and CORBA - Platforms

- **Both DCE and CORBA are portable**
  - **implementations now exist for most platforms and operating systems...**
  - **...mainframe, minicomputers, micros,...**
- **For practical portability, you must take into account**
  - **operating system version**
  - **compiler version and vendor**
  - **processor type**

Portability in DCE and CORBA - Programming Languages

- **Both CORBA and DCE can be used with a range of programming languages**
  - **for both CORBA and DCE, and for each programming language, a mapping must be standardized**
- **Currently these are standardized**
  - **DCE: C (C++ to follow)**
  - **CORBA: C, C++, Smalltalk, Ada (COBOL and others to follow)**

CORBA and DCE - Programming Interfaces

- **CORBA is object-oriented; DCE is mainly procedural**
  - **CORBA requires a commitment to object-oriented principles**
  - **DCE is cumbersome to use from an object-oriented language**
  - **DCE requires code ‘scaffolding’**

Portability, Diversity, and Interoperability

- Remember that you are not constrained to use the same platform and programming language everywhere
  - a client application can be written in Smalltalk on a PC...
  - ... the server can be written in COBOL on a mainframe
- You can change platform and programming language later
- This relies on interoperability between distributed systems implementations from different vendors
  - both DCE and CORBA offer interoperability

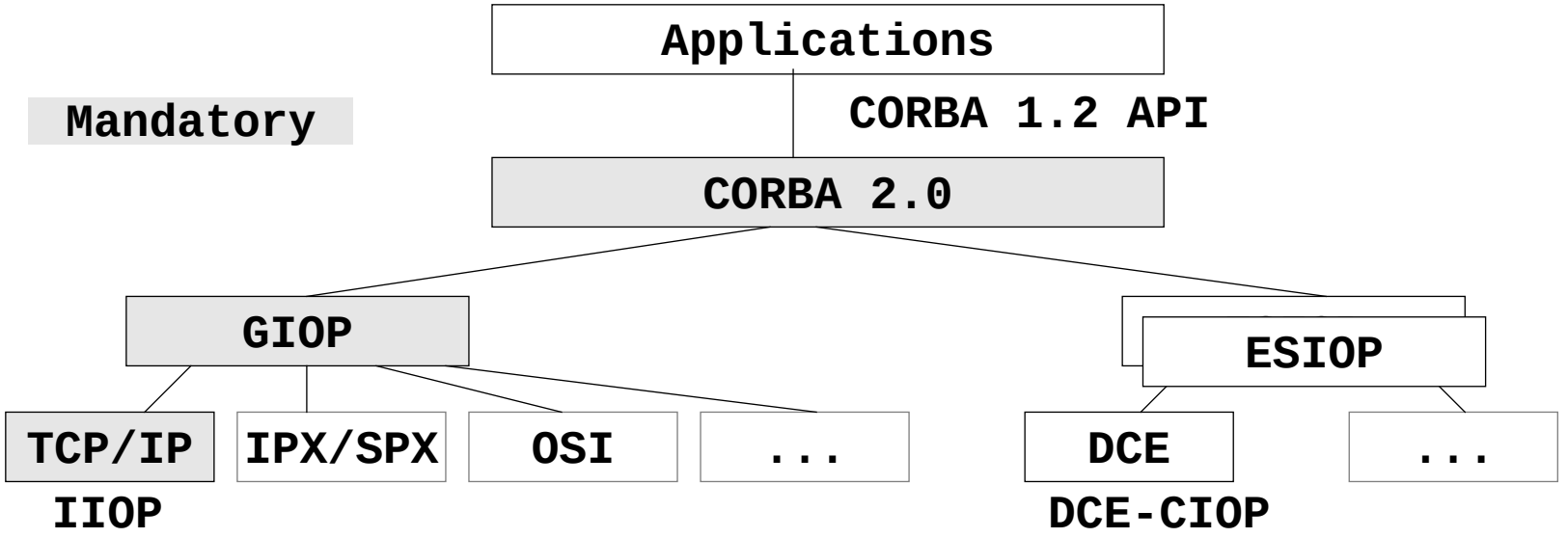
Interoperability in DCE and CORBA - a key issue

- How can distributed systems interoperate? Two approaches
  - a common protocol
  - protocol gateways

Interoperability in DCE

- OSF selected a common protocol (the DCE RPC)
  - This has now been made freely available to anyone, royalty-free
- Microsoft have also selected the DCE RPC for use in Distributed OLE
  - but not the DCE services

CORBA Interoperability Architecture



- The IIOP (Internet Inter-ORB Protocol) is mandatory
  - guarantees interoperability between any ORBs
- The DCE-CIOP (Common Inter-ORB Protocol) is optional
  - if provided, guarantees interoperability with likewise ORBs

Interoperability between DCE and CORBA?

- **The CORBA DCE-CIOP does not give service interoperability between DCE and CORBA**
  - **it only gives protocol interoperability for requests and responses**
- **Service interoperability could in principle also be achieved....**
  - **... like all high-level gateway solutions, transparency is difficult**
  - **... the two architectures are different**

CORBA and DCE - Usability

- **DCE implementations provide services that CORBA implementations do not yet**
  - **CORBA specifications appeared later than the DCE equivalents**
- **DCE programs are large and tend to be slow**
  - **but there is now expertise in tuning DCE applications**

CORBA and DCE - Markets

- DCE implementations are mainly from large vendors
- Some large user and vertical markets have settled on DCE
- CORBA implementations include smaller vendors
- There are application development environments for both CORBA and DCE
  - but take care when evaluating the level of tool support

Summary

- **CORBA and DCE are two of the open environments for distributed computing**
  - **take care to assess your needs, and compare like with like**
- **For more on this topic**
  - **on CORBA, see *CORBA* (Object Management Group Inc.)...**
  - **... see also *First Class* magazine, published by OMG**
  - **on DCE, see *Introduction to DCE* (Prentice-Hall)**
  - **... see also *Understanding DCE* by Ward Rosenberry et al (O'Reilly)**