



**Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom**

TELEPHONE:
INTERNATIONAL:
FAX:
E-MAIL:

**Cambridge (01223) 515010
+44 1223 515010
+44 1223 359779
apm@ansa.co.uk**

Training

ANSAwise - Introduction to 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.

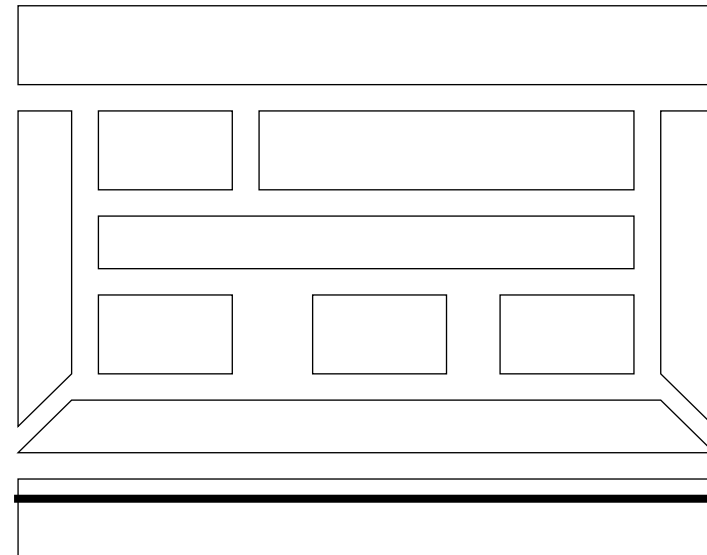
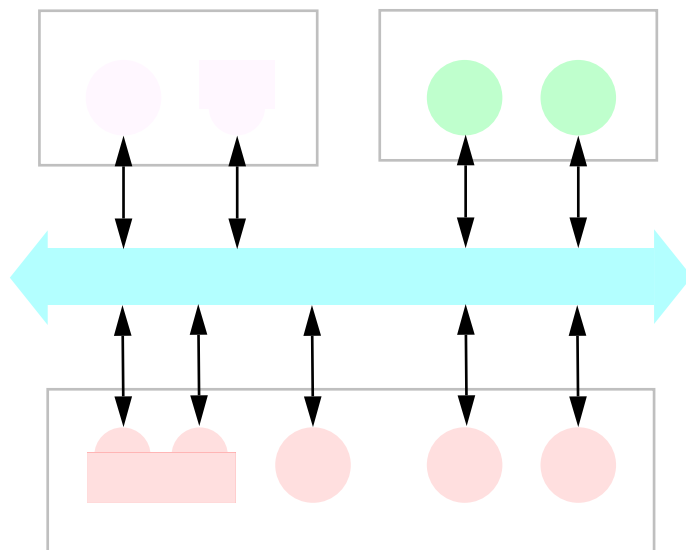
APM.1322.01

Approved
Briefing Note

21st October 1994

Distribution:
Supersedes:
Superseded by:

Introduction to CORBA and DCE



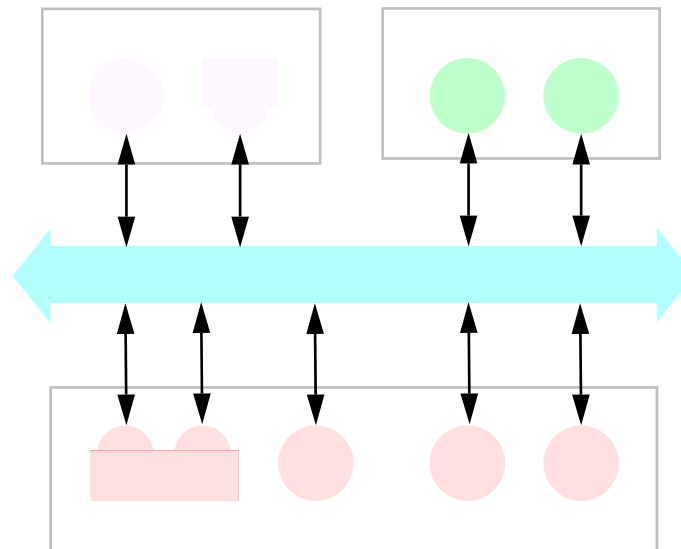


In this session

- *Describe the key features of two important open environments for distributed computing*
 - CORBA (Common Object Request Broker Architecture)
 - DCE (Distributed Computing Environment)
- *Compare and contrast them*
- *Indicate their place in your distributed systems strategy*

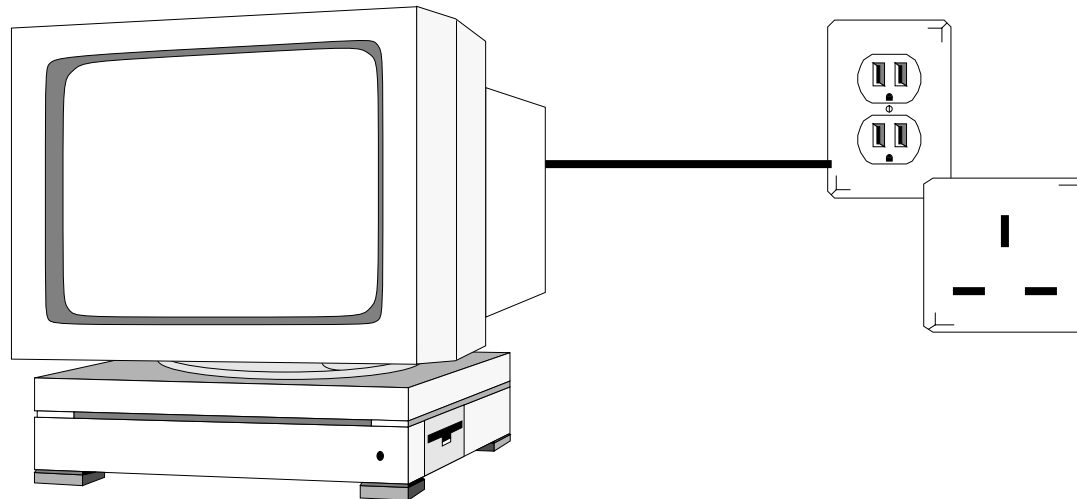
CORBA (Common Object Request Broker Architecture)

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



The OMG's Vision

- *The Global Information Appliance*



- *“It should be as easy to plug a computer into a world of computing services as it is to plug a computer into the world power grid”*



The OMG's Focus

- ***Application integration***
 - the same thing as distributed processing
- ***Constructing information-sharing distributed systems from diverse sources***
 - heterogeneous
 - networked
 - physically disparate
 - multi-vendor



The OMG's view of existing approaches

- *They are too low-level*
 - excellent building blocks, but not at the level the application developer is interested in
- *There is no standardized integrating framework for applications*



The OMG's view of objects

- *Objects provide services*
- *Objects simplify the problem through*
 - **encapsulation**
 - **polymorphism**
 - **inheritance**



The OMG's Approach

- *Create consensus based on commercially-available software...*
 - proposals for standards must describe technology that is *imminently available*
 - paper-only standards are inadequate
- *... Create a marketplace for off-the-shelf standards-compliant software*



The OMG's Method of Operation

- *Not-for-profit company*
- *Small staff; no internal development*
- *Object World subsidiary spreads the word through shows, conferences, market studies, seminars,...*
- *Now has more than 350 members*



How CORBA Standards are Made

- ***OMG selects interfaces***
 - through competitive selection from industrial proposals
- ***OMG publishes interfaces***
 - the specifications are freely available to anyone who wants them
- ***OMG controls interfaces***
 - they belong to OMG, not the original submitter; OMG controls their evolution
- ***OMG liaises with standards bodies***
 - so that specifications become standards

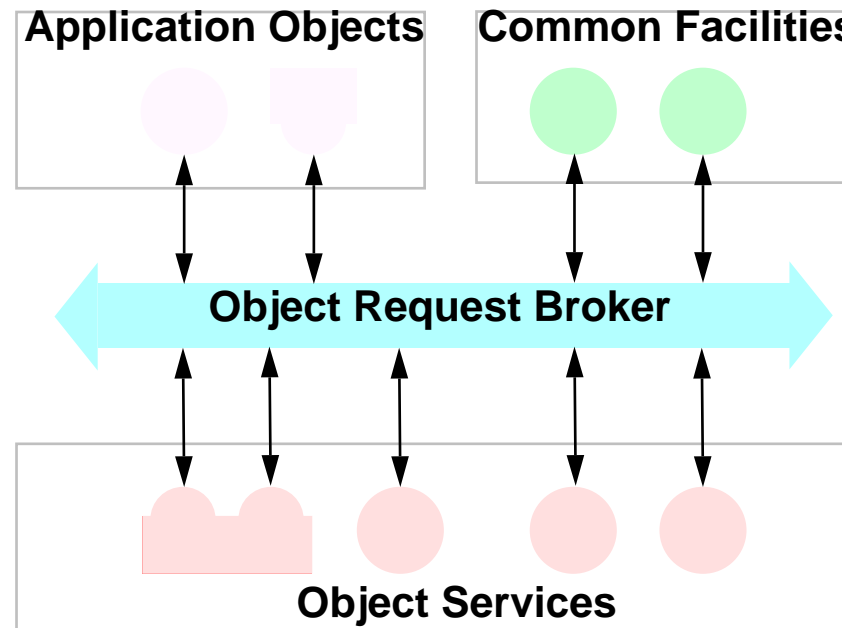


OMG is Neutral

- ***OMG does not deal with implementations***
 - **it does not create, sell, or resell implementations**

- ***OMG does not test implementations***
 - **this is done by X/Open, who have a strong reputation in conformance testing**

The Object Management Architecture



- **Consists of the Object Request Broker (ORB), plus objects**
 - **Objects are Object Services, Common Facilities, or Application Objects**



The Object Request Broker (ORB)

- *Objects request the services of other objects via the ORB*
 - the ORB is responsible for locating the object implementation, and all the communications mechanisms that support the request
 - client and object implementation can be written in different languages, and run on different types of machines





Object Services

- ***Common Object Services are basic 'system-level' services***
 - but they are themselves *objects*
 - they have interface specifications, just like all objects...
 - so alternative implementations are possible, with different quality-of-service, say
- ***Example Object Services***
 - Transactions
 - Security
 - Events



Common Facilities

- *Common Facilities are application-level objects that can be shared between applications*
- *Horizontal-market, for example*
 - e-mail
 - printing
 - compound documents
- *Vertical-market, for example*
 - geo-spatial data processing
 - system management
- *Common Facilities provide interoperability between ISV products*
 - they may be highly specialist



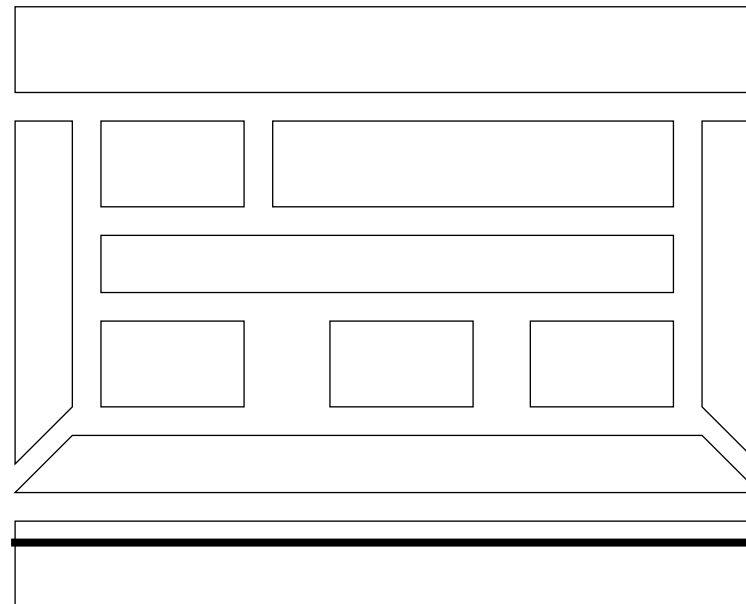
Application Objects

- *Application Objects are application-specific*
 - provided by ISV
 - provided by end-user



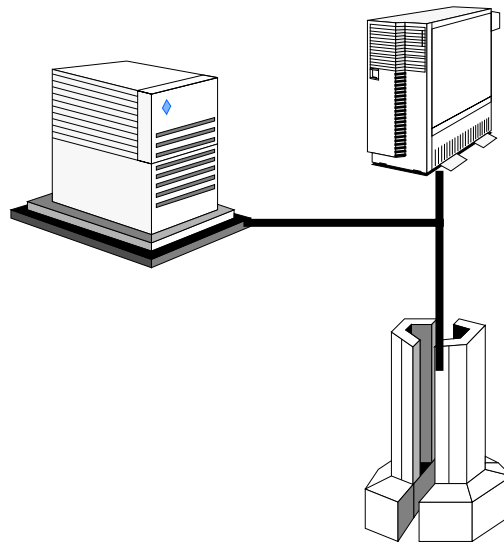
DCE (Distributed Computing Environment)

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



The OSF's Focus

- *Interoperability*



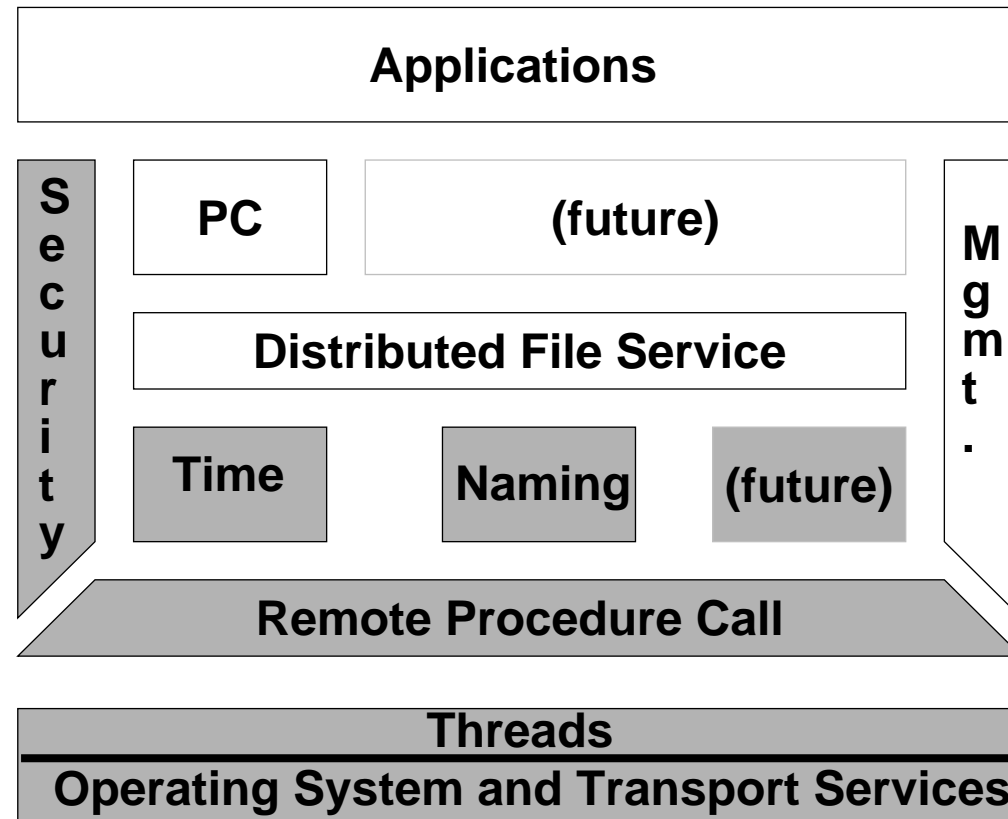


The OSF's Method of Work

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



The DCE Component Architecture



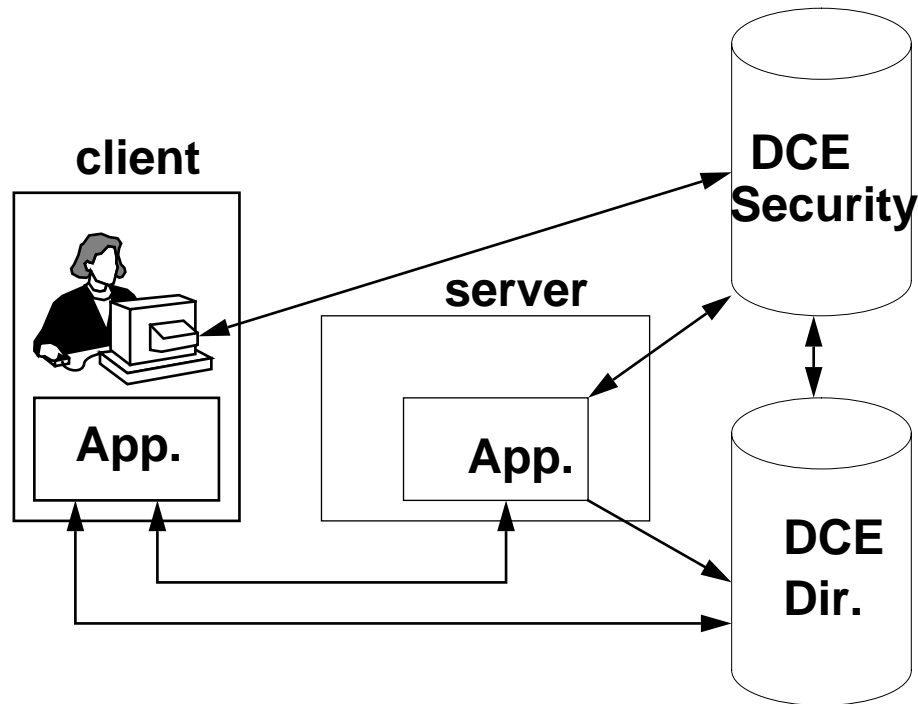


DCE Services

- *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*

The DCE Environment

- *Clients and servers interact with the 'core services'*





Portability in DCE and CORBA

- *Both DCE and CORBA are portable*
 - implementations now exist for most platforms and operating systems
- *Language mappings are being standardized,*
 - C, C++, Smalltalk, COBOL,...



Interoperability in DCE and CORBA - a key issue

- *How can distributed object systems interwork? Two approaches*
 - a common protocol
 - protocol gateways
- *DCE selected a common protocol (the DCE RPC)*
 - This has now been placed in the public domain
- *CORBA is about to resolve the interoperability issue*
 - but interoperation between CORBA vendors will still not be straightforward, at least in the short term



Interoperability between DCE and CORBA?

- *This can also be achieved....*
- *... like all high-level gateway solutions, transparency is difficult*
- *... the two architectures are different*



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'
- *Both CORBA and DCE support procedural languages (including C)*



CORBA and DCE - Usability

- *DCE provides services that CORBA does not yet provide*
 - *But CORBA is filling in the gaps fast*
- *DCE programs are large and tend to be slow*



CORBA and DCE - Markets

- *DCE implementations are mainly from large vendors*
- *Some large vertical markets have settled on DCE*
- *CORBA implementations include smaller vendors*



Summary

- ***CORBA and DCE are just two of the many environments for distributed computing***
 - take care to assess your needs, and compare like with like
- ***Both CORBA and DCE acknowledge the influence of ANSA***
- ***For more on this topic***
 - on CORBA, see *Object Management Architecture Guide* (Object Management Group Inc.)...
 - ... see also *First Class* magazine, published by OMG
 - on DCE, see *Introduction to DCE* (Prentice-Hall)