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**

APM Business Unit

Selecting Distribution Requirements and Policy [for GPT]

Chris Mayers

Abstract

GPT wish to employ CORBA technology in conjunction with the Shlaer/Mellor OOA/RD development method. CORBA technology has only limited support for distribution transparency; OOA/RD currently has few distribution concepts.

This presentation summarizes the ANSA/ODP distribution transparency mechanisms, and is a basis for the discussion of their detailed requirements.

APM.1658.01 **Approved** 15th November 1995
Briefing Note

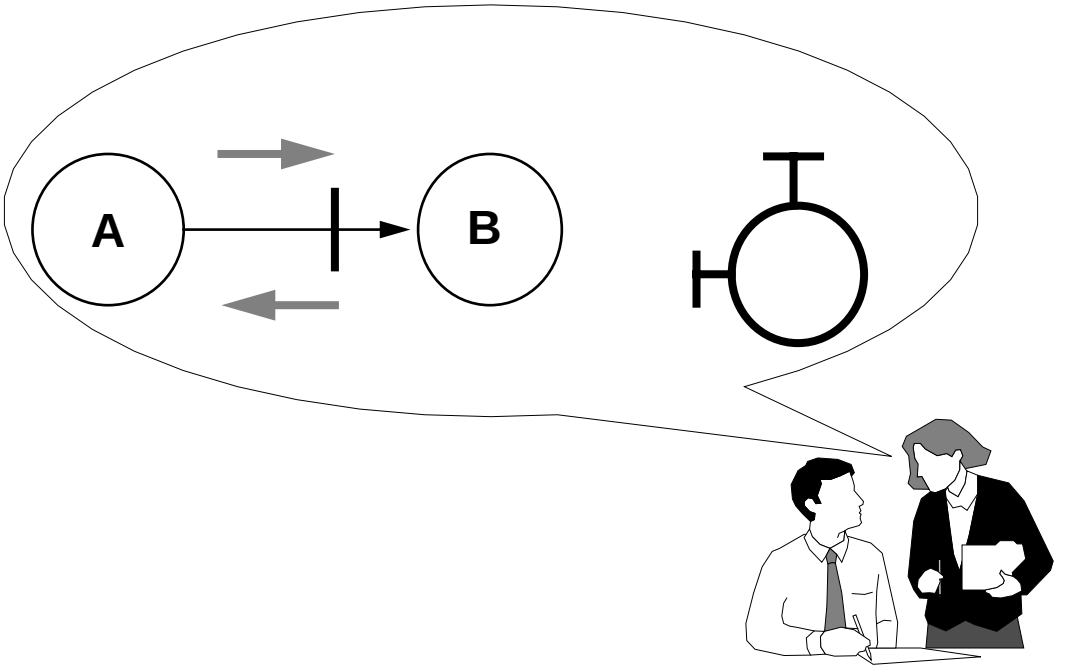
Distribution:
Supersedes:
Superseded by:

Distribution Requirements and Policy Workshop

Major topics

- ***Usage of Computational specifications (CORBA IDL)***
- ***ANSA/ODP distribution transparencies***
- ***ANSA/ODP domains***

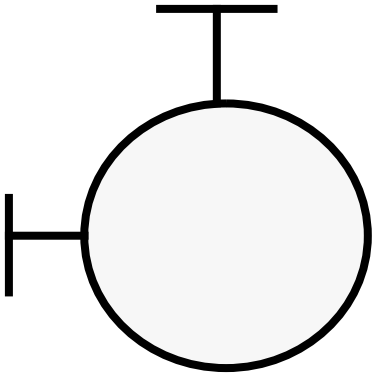
Computational specifications



- ***Describe objects, interfaces, operations,...***

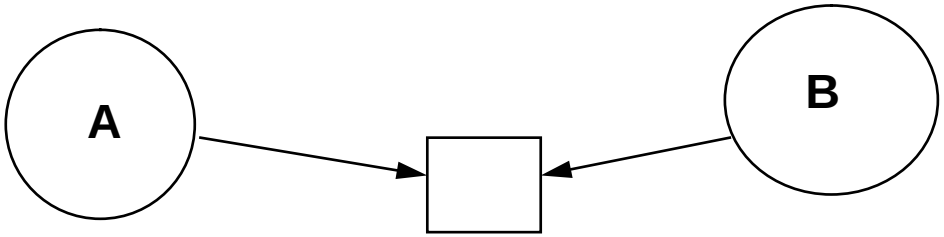
Encapsulation - the key concept

- ***Objects are encapsulated***
  - **every object provides a service via interfaces**
  - **the interface is public; the implementation is private and hidden**
  - **encapsulation forms a boundary; the only access to an object is via its interfaces**

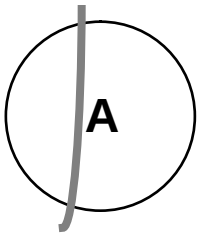


Encapsulation rules

- ***Objects cannot share state directly (only access it via interfaces)***
  - **this is not allowed**



- ***An object cannot be distributed in pieces; it must all be in one place***
  - **this is not allowed**



A simple service in CORBA IDL

- *This looks rather like C++*

```
interface Echo {
// Comment lines start with two slashes

   string Echo (in string Src);

   string Reverse (in string Src);

};
```

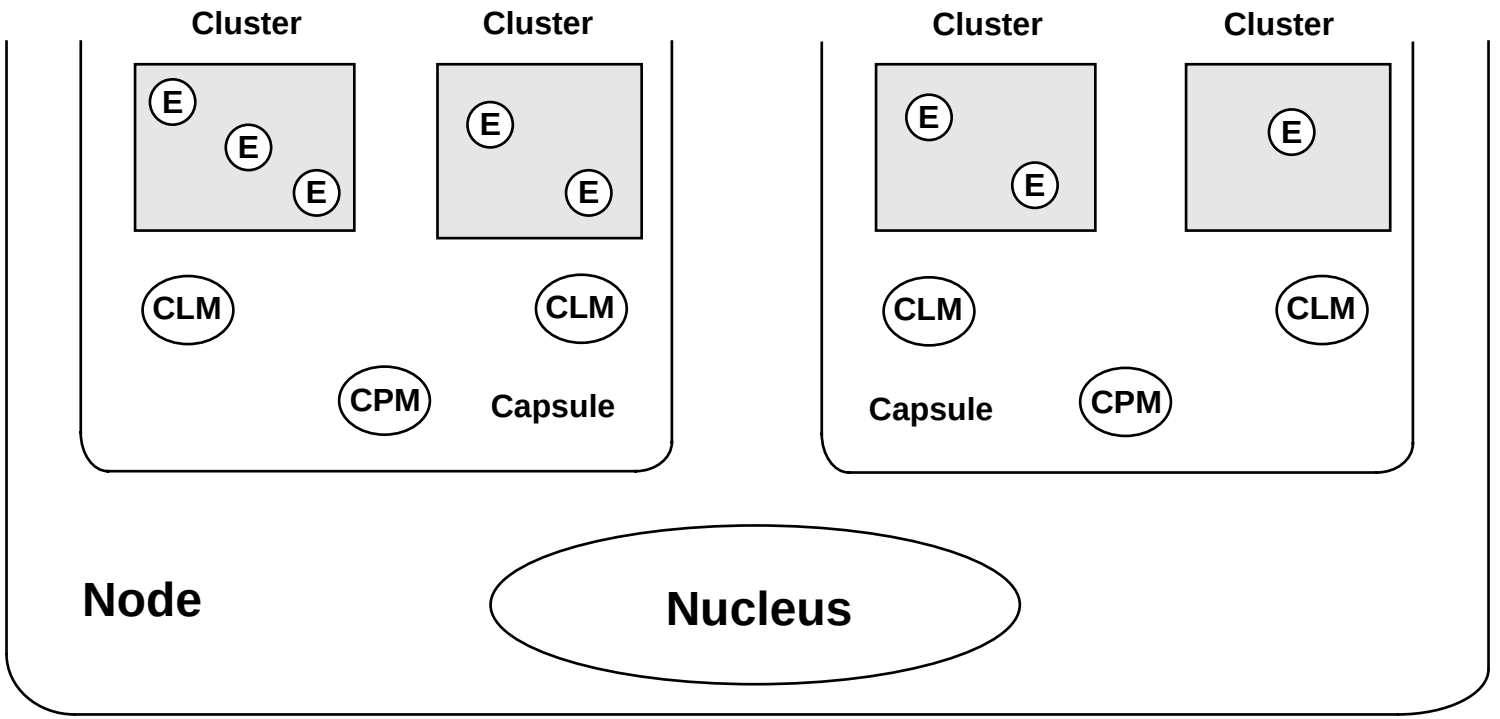
Computational Requirements

- ***Where are computational interfaces required?***

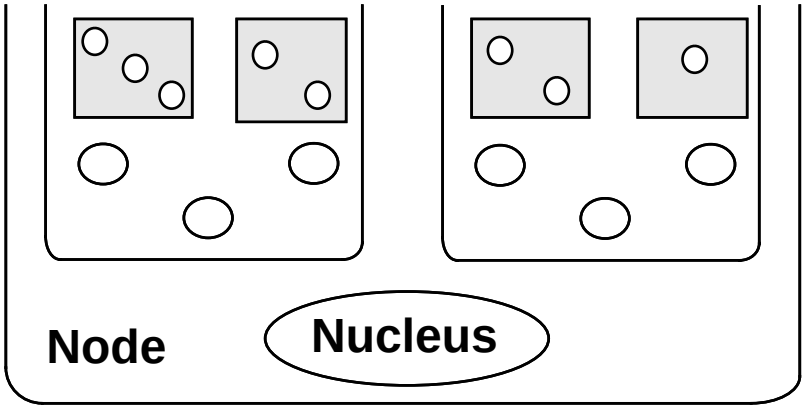
- ***How are these to be expressed in CORBA IDL?***

For discussion...

Engineering Encapsulation Infrastructure

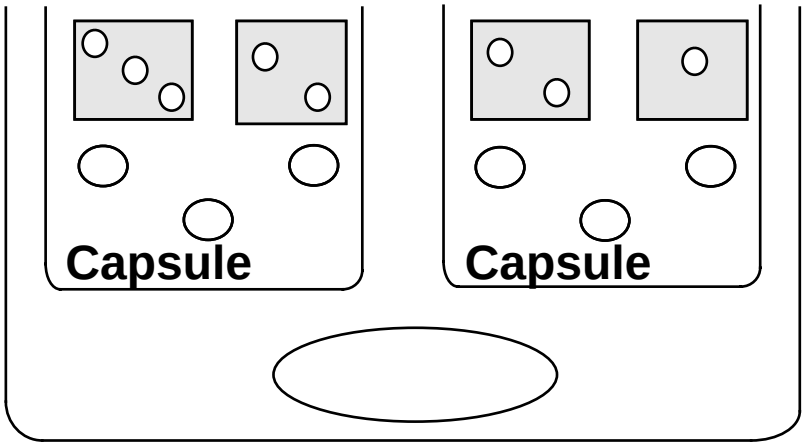


Node and Nucleus



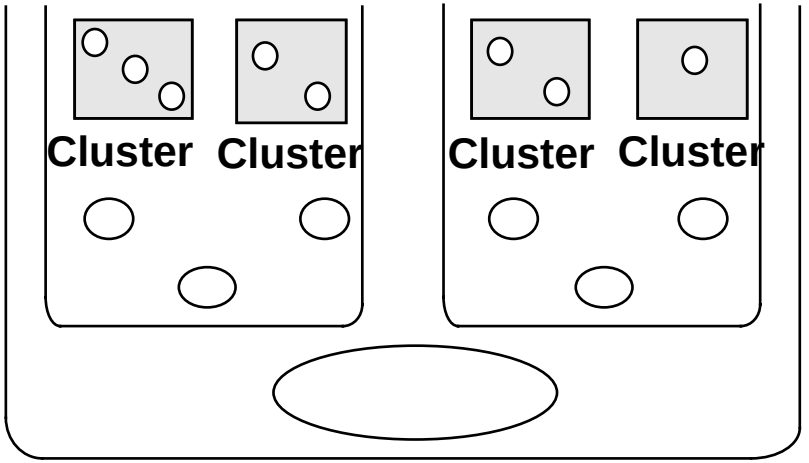
- *A node is the unit of network addressing*
- *The nucleus handles communications between nodes and capsules*
  - it may be part of the operating system (if there is one), or a layer on top of it

Capsules



- ***Capsules contain objects***
  - **objects must be encapsulated**
  - **every object is contained in a capsule**
  - **different capsules separate sets of objects**

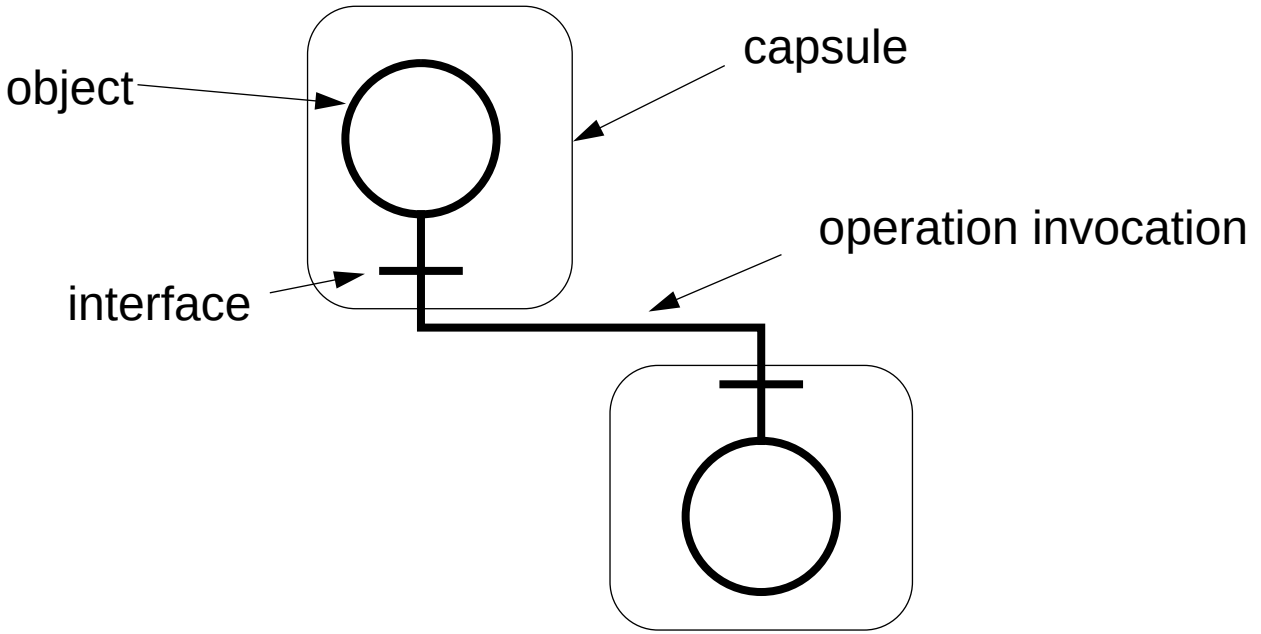
Clusters



- ***A cluster is a collection of objects that must be 'kept together'***
  - **if an object moves (*migrates*) between capsules, clustered objects must move together with that object**
  - **if an object fails, you might want to shut down other objects automatically**
  - **when starting up one object, you might want to start up other objects automatically**

Transparency examples

- ***This shows an object invoking an operation***



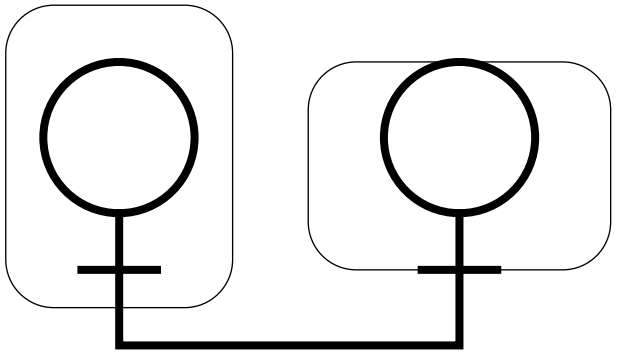
Transparency Types

- ***Access***
- ***Failure***
- ***Location***
- ***Migration***
- ***Persistence***
- ***Relocation***
- ***Replication***
- ***Transaction***

Selective Transparency Engineering - Access

- ***Access Transparency***

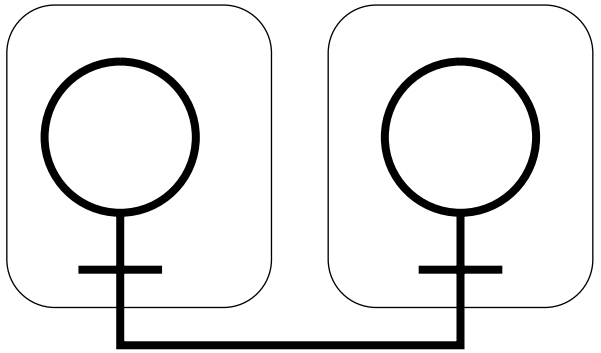
  - **application need not know the type of machine where the object is executing**



Selective Transparency Engineering - Location

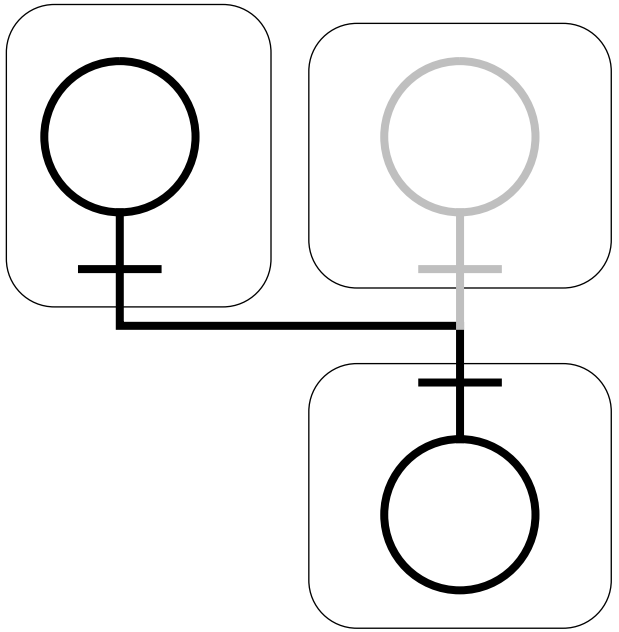
- ***Location Transparency***

  - **application need not know where object is to use it**



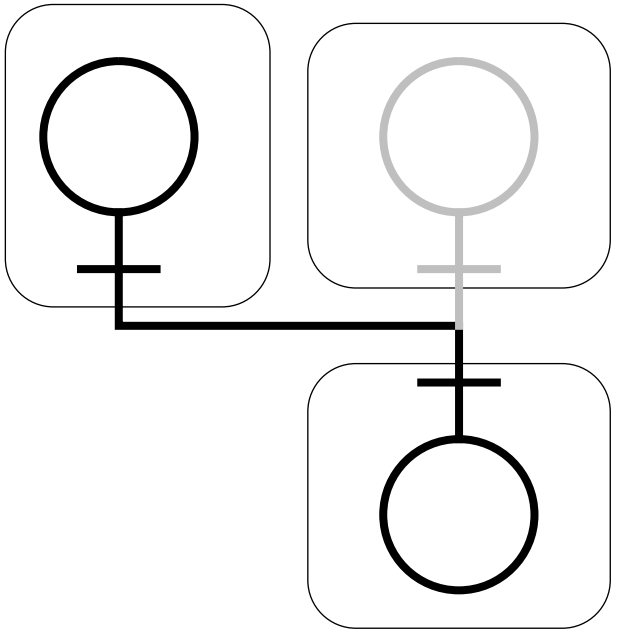
Selective Transparency Engineering - Relocation

- ***Relocation Transparency***
  - **application need not know where the object has moved to**



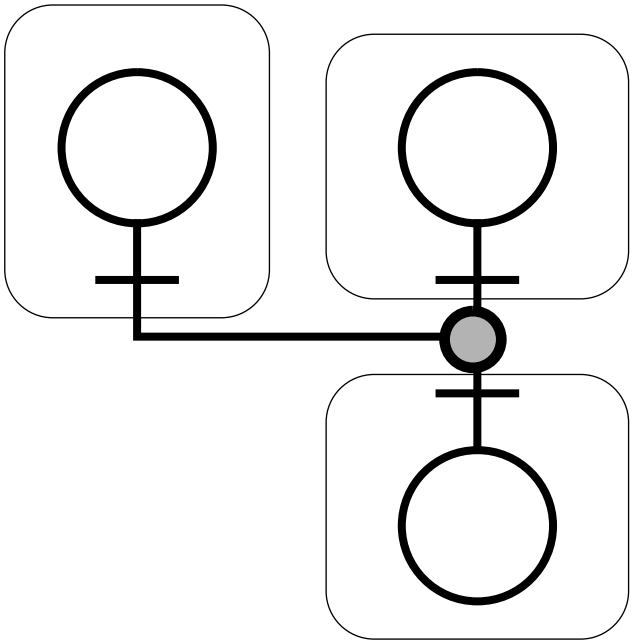
Selective Transparency Engineering - Migration

- ***Migration***
  - **server need not know that it has itself moved**



Selective Transparency Engineering - Replication

- ***Replication Transparency***
  - **application need not know how many copies**



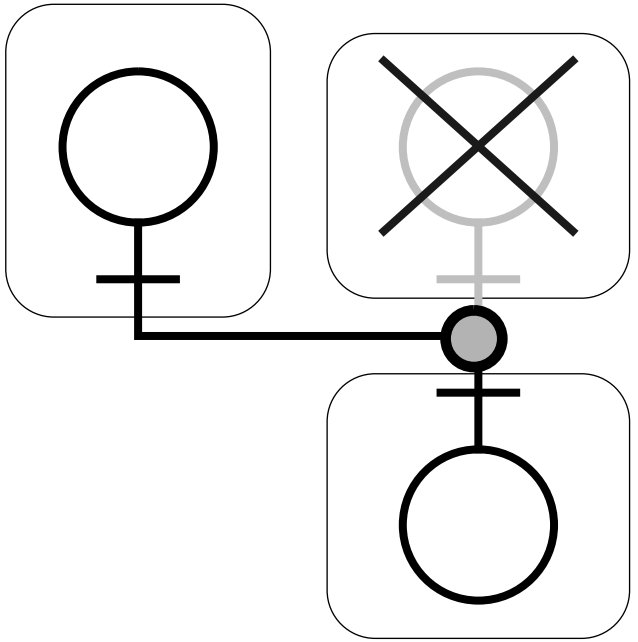
  - **application only sees a single interface**

Replication Transparency

- ***Server objects are members of a group***
  - **do not confuse with a cluster!**
- ***Replication transparency uses special mechanisms to make sure the group members are consistent***
  - **for instance, it may use multi-point channels and special protocols**
- ***Implementing replication transparency efficiently is difficult***
  - **it may need information from the application**
  - **it is under active research in the distributed systems community**

Selective Transparency Engineering - Failure

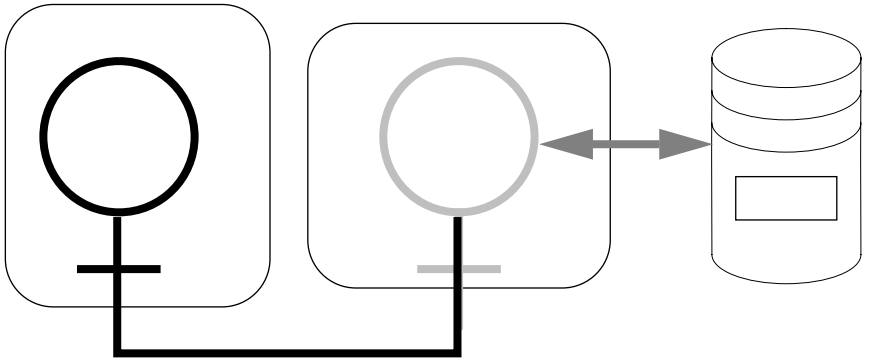
- ***Failure Transparency***
  - **application need not know when an object fails**



  - **may use replication transparency to achieve this**

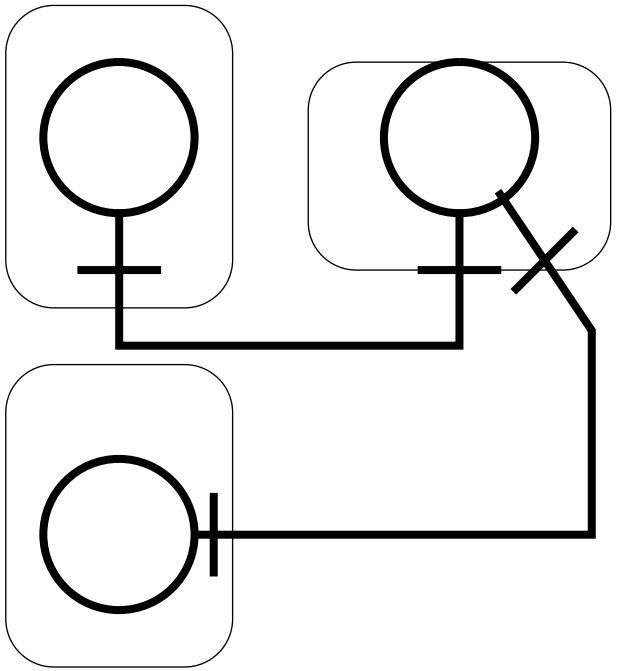
Selective Transparency Engineering - Persistence

- ***Persistence Transparency***
  - **application need not know of object activation/deactivation**



Selective Transparency Engineering - Transaction

- ***Transaction Transparency***
  - **applications need not be aware of inconsistent states**



Transparency Requirements

- ***Which transparencies are required?***

- ***Under what circumstances are they required?***

For discussion...

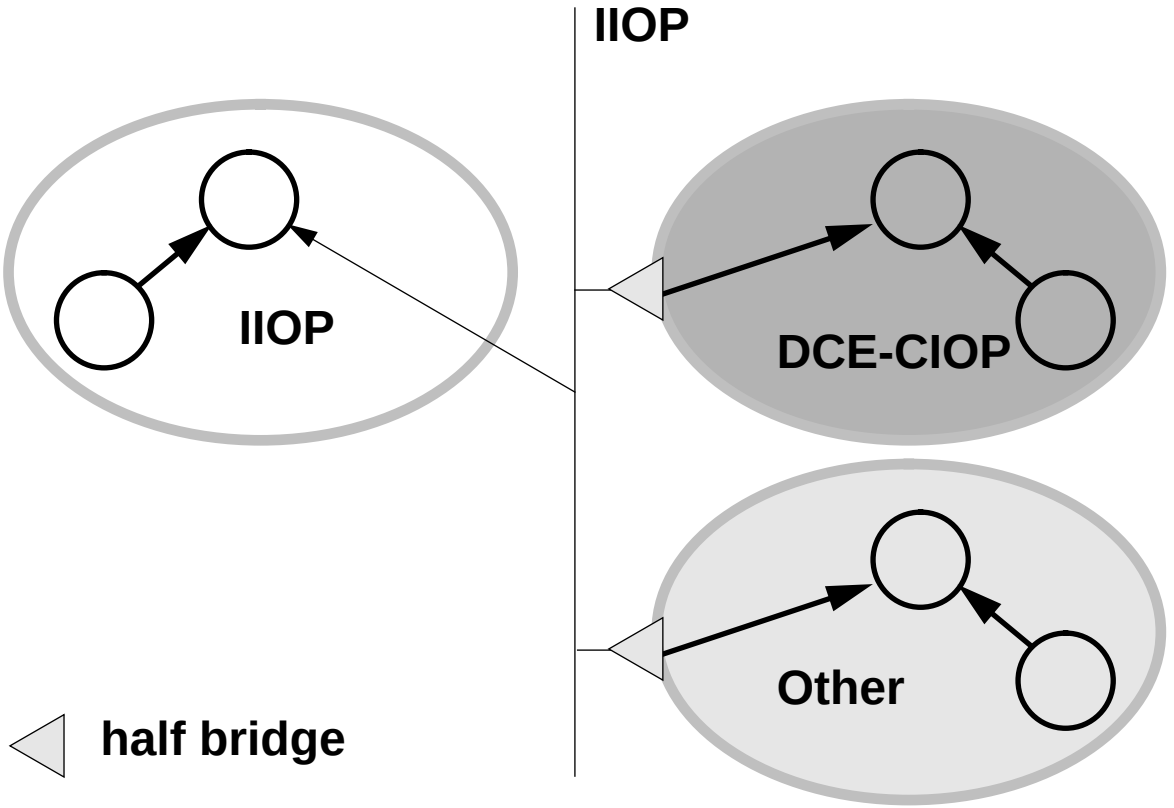
Domains

- *We can consider the distributed system as a set of domains*
  - a domain is a set of objects sharing a common set of characteristics
- *Domains may be*
  - technological
  - administrative
  - application-oriented
- *Interoperability is concerned with bridging these domain boundaries*
- *NB: these are not the same concepts as Shlaer/Mellor OOA/RD 'domains' and 'bridges'*

Types of domains

- ***Technology domains***
  - distributed system infrastructure (CORBA, DCE, OLE,...)
  - communication (RPC, RDA, queued messaging,...)
  - data representation (ASN.1 BER, CORBA CDR, ONC XDR,...)
  - quality of service (latency, bandwidth,...)
- ***Administrative domains***
  - security
  - resource management
  - remuneration
- ***Application-orientated***
  - subject matter
  - information model

CORBA Communication Domains



Domain Requirements

- ***Which types of domain are required?***

- ***How many domains of each type will exist?***

For discussion...