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 - Building Applications with ANSA

Chris Mayers

Abstract

Organizations may wish to evaluate ANSAware and compare it with other distributed systems environments.

ANSAware is unusual in that it was built as a prototype system to demonstrate ANSA principles. An appreciation of these principles makes it easier to evaluate other distributed systems environments as well.

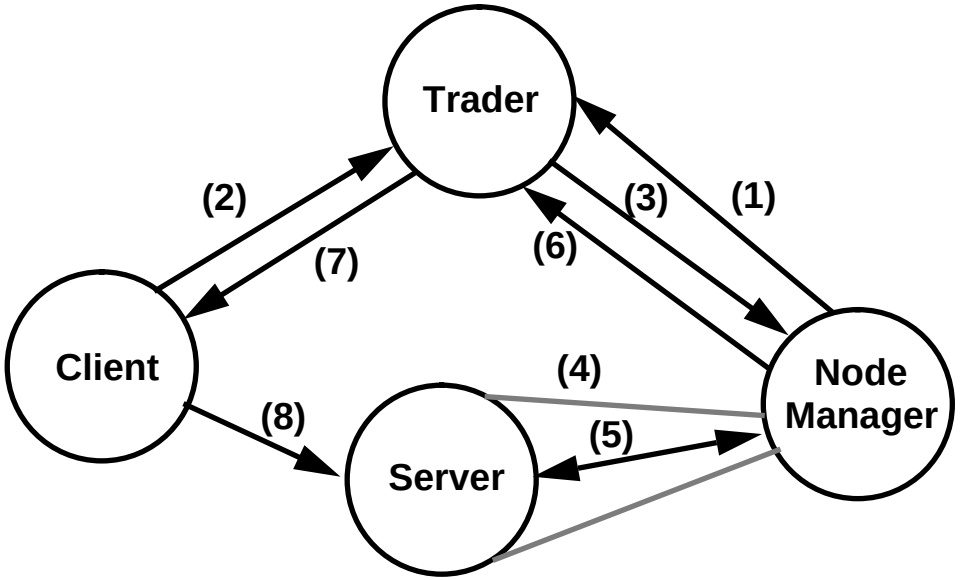
This module of the ANSAwise training programme gives a brief review of the ANSA concepts that are most prominent in ANSAware, discusses key engineering features of ANSAware, showing how these contribute to the services in ANSAware 4.1.

APM.1366.01 **Approved** 28th November 1994

Briefing Note

Distribution:
Supersedes:
Superseded by:

Building Applications with ANSA



In this session

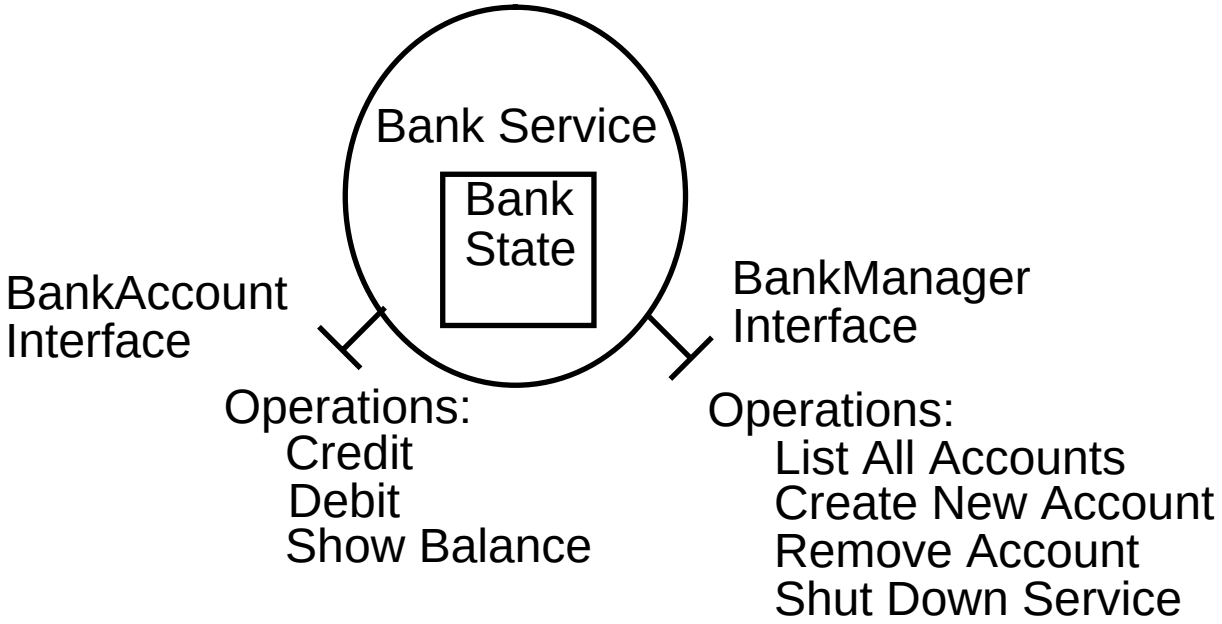
- ***Briefly review some concepts of the Computational Model***
  - **those involved in Trading**
- ***Describe the components that form ANSAware 4.1***
- ***Show how some of the ANSAware services fit together***

Computational Model

- ***Based on services provided by encapsulated objects***
  - **interacting via operations defined in interfaces**
- ***Computational model assumes that services are always available when used...***
  - **...and that resources are always available too**

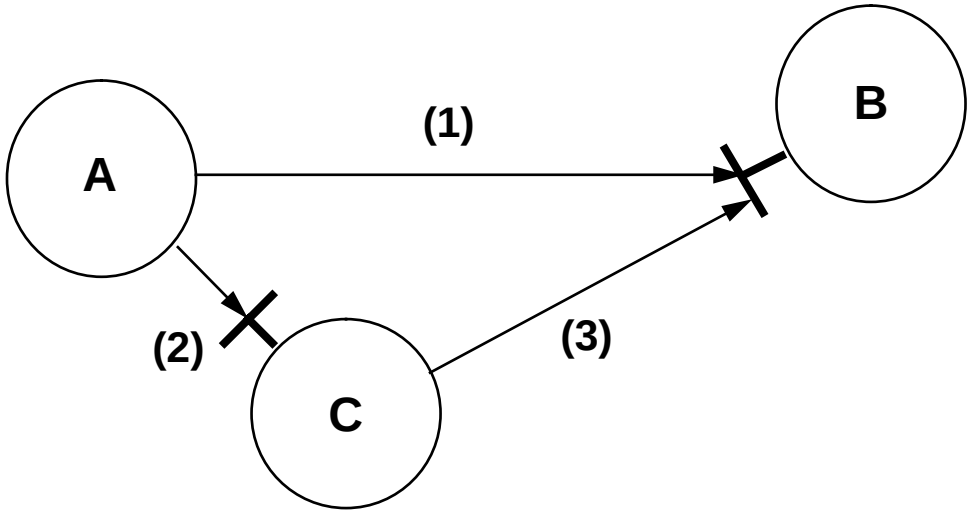
Simple Bank - Operations in multiple interfaces

- ***A service can have more than one interface***
- ***Each interface has its own operations***



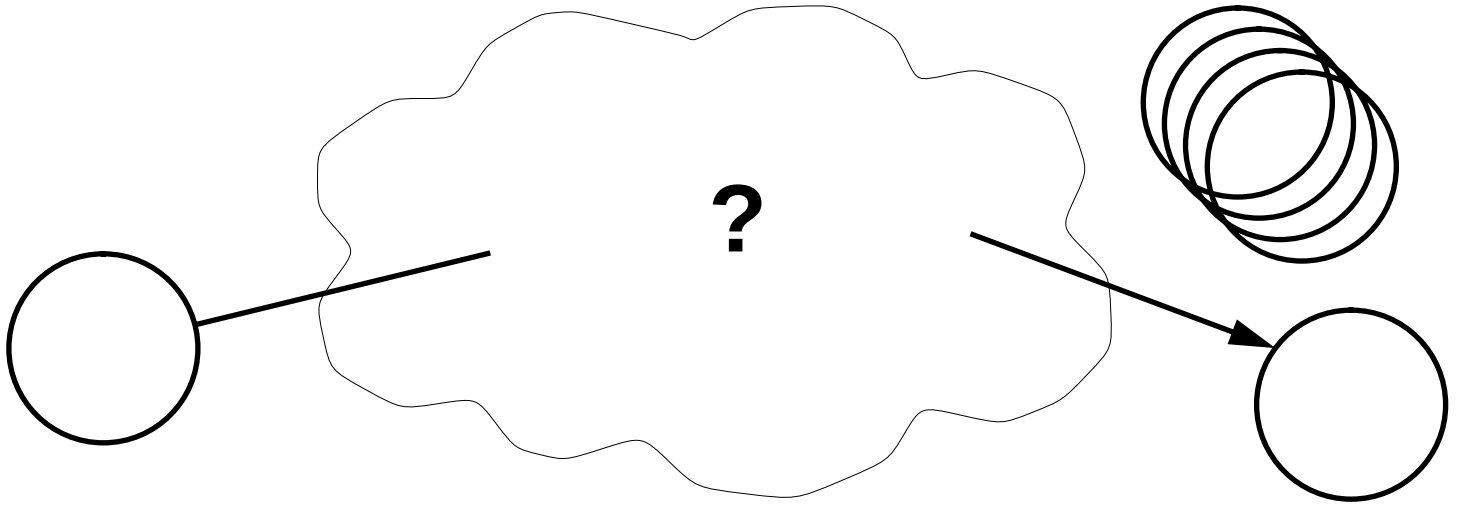
Using Interfaces

- *Object A is using one of object B's interfaces*
- *Suppose it needs to tell C to use the same interface*



- *It must be possible to pass a binding to B's interface between A and C*

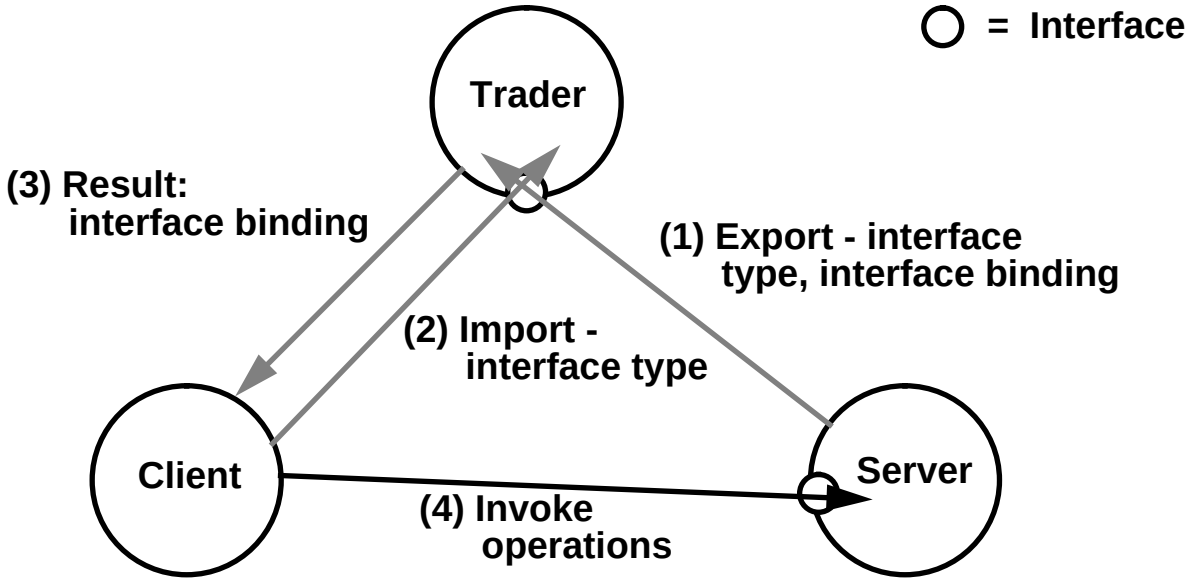
The need for Trading



- ***How can clients find servers that provide the services that they need?***
  - **in the future, there will be millions of interconnected servers around the world**
  - **clients will come and go dynamically**
  - **servers will come and go dynamically**

Interface Transfer in Trading

- ***Interface transfer is used in basic Trading interactions***



What is ANSAware?

- ***A toolkit for building distributed applications***

  - **based on the ANSA architecture**

  - **supporting diverse operating systems and hardware platforms**

Distinctive ANSAware features

- *Based on long-term research; a mature toolkit*
- *Implements ANSA transparencies*
- *Supports concurrency and light-weight implementations*
- *Supports group execution*
- *Third-party extensions available*
- *Ported to many environments*
- *Not a shrink-wrapped product - you can experiment with it*

ANSAware 4.1 - Infrastructure and Tools

- ***Basic infrastructure support***
  - **the ANSAware run-time library**
  - **the ANSAware RPC protocol (REX/GEX)**
- ***Tools***
  - **STUBC: Compiler for Interface Definition Language (IDL)**
  - **PREPC: Preprocessor for service calls embedded in C**

How do applications interface with ANSAware?

- ***Toolkits can provide two general approaches***
  - **Application Programming Interface (API): the procedural approach**
  - **Language Extension**
- ***ANSAware provides Language Extension***

The procedural approach

- ***Other toolkits provide Application Programming Interfaces (APIs) between applications and systems***

Applications and Applications Programmers

Application Programming Interface

Systems and Systems Programmers

  - **the *procedural* approach**

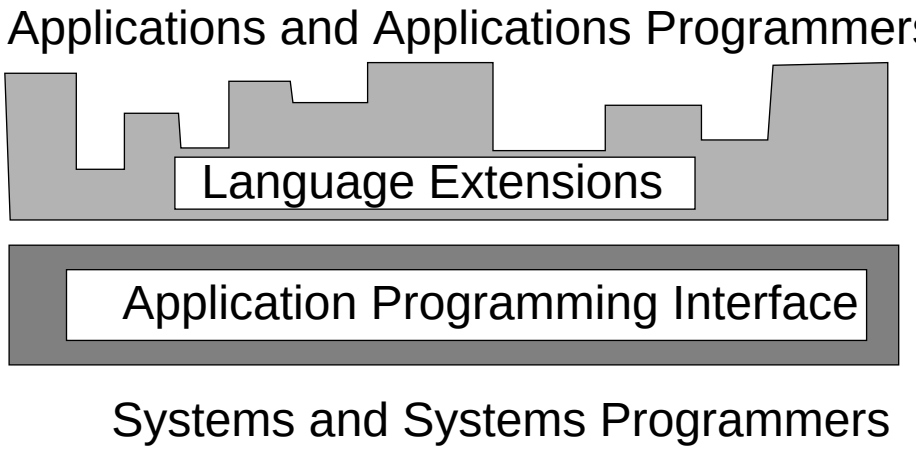
- ***This approach causes problems...***

Problems with the API/procedural approach

- ***Large APIs are difficult to comprehend***
  - **you often need to understand *all* the API to use *any* of it**
- ***Large APIs grow larger***
  - **the API widens as application requirements grow...**
  - **... it gets harder and harder to comprehend**
- ***APIs do not adequately hide system detail***
  - **the detail shows through to the applications**
- ***Mistakes are easy to make, and are not detected early enough***
  - **there is a lot of code to write**
  - **most checks are done at run time**

The ANSA approach - Language Extension

- ***ANSA takes a more powerful programming language view:***



- ***The interface seen by the application programmer is expressed as a series of Language Extensions***
- ***Simpler to understand and use than a procedural API***

Language Extension - Advantages

- ***Advantages***
  - a simple, system-independent programming model
  - Independence between application and system designers
  - easy migration of software between platforms
  - compile-time checking
- ***Benefits***
  - increased confidence in software
  - more robust, error-free and dependable software
  - system evolution
  - applications unaffected by system re-engineering
  - system unaffected by application re-design

ANSAware Node and Nucleus

- ***Node: a single machine or a tightly-coupled set of machines***
  - a node allows for the creation/destruction of processes with a unique process id
- ***Nucleus: an engineering object which manages the resources of a node***
  - the Nucleus is implemented as a set of libraries that get linked together with the user's application to form a *capsule*
  - the Nucleus is also called the *infrastructure* or *capsule library*

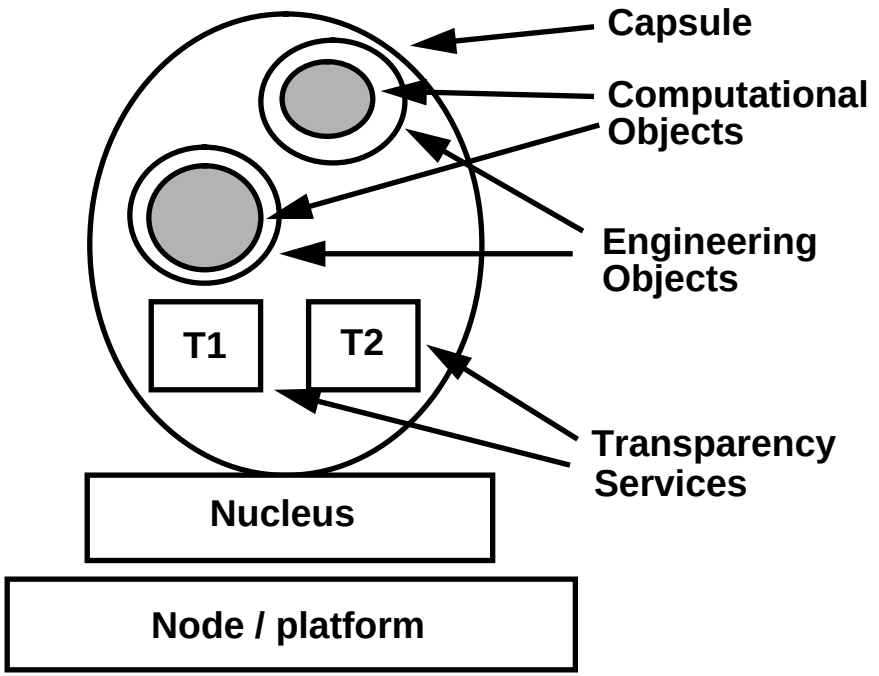
ANSAware Capsule

- *A capsule is*
  - the unit of autonomous execution within ANSAware
  - an address-space supporting a single instance of the run-time system...
  - ... providing a memory protection boundary
- *A capsule provides:*
  - efficient, transport-independent and portable RPC protocol
  - light-weight threads
  - synchronisation operations
  - timer handling
  - support for interworking with other systems - e.g. X11 graphical user interface

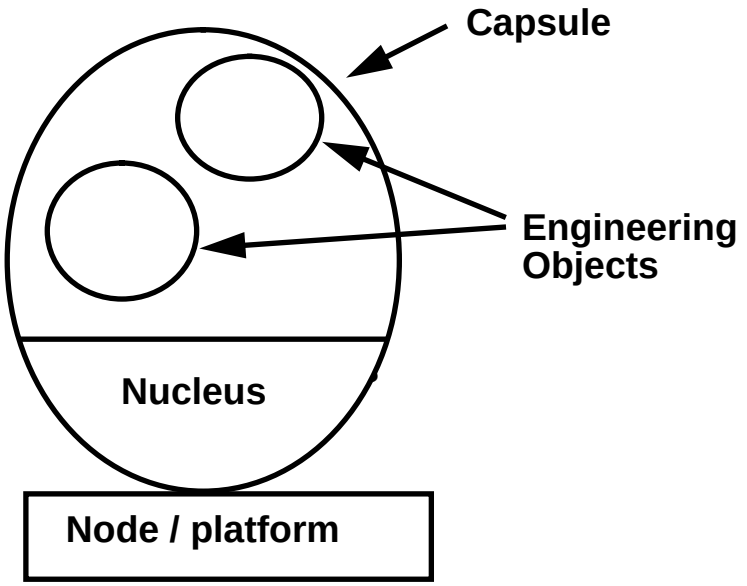
Node support for capsules

- ***The node support for capsules depends on the underlying operating system***
  - **Unix supports multiple capsules per node**
  - **DOS only supports one**
- ***Capsules can be created and destroyed dynamically***

ANSA Node, Nucleus, and Capsule structure

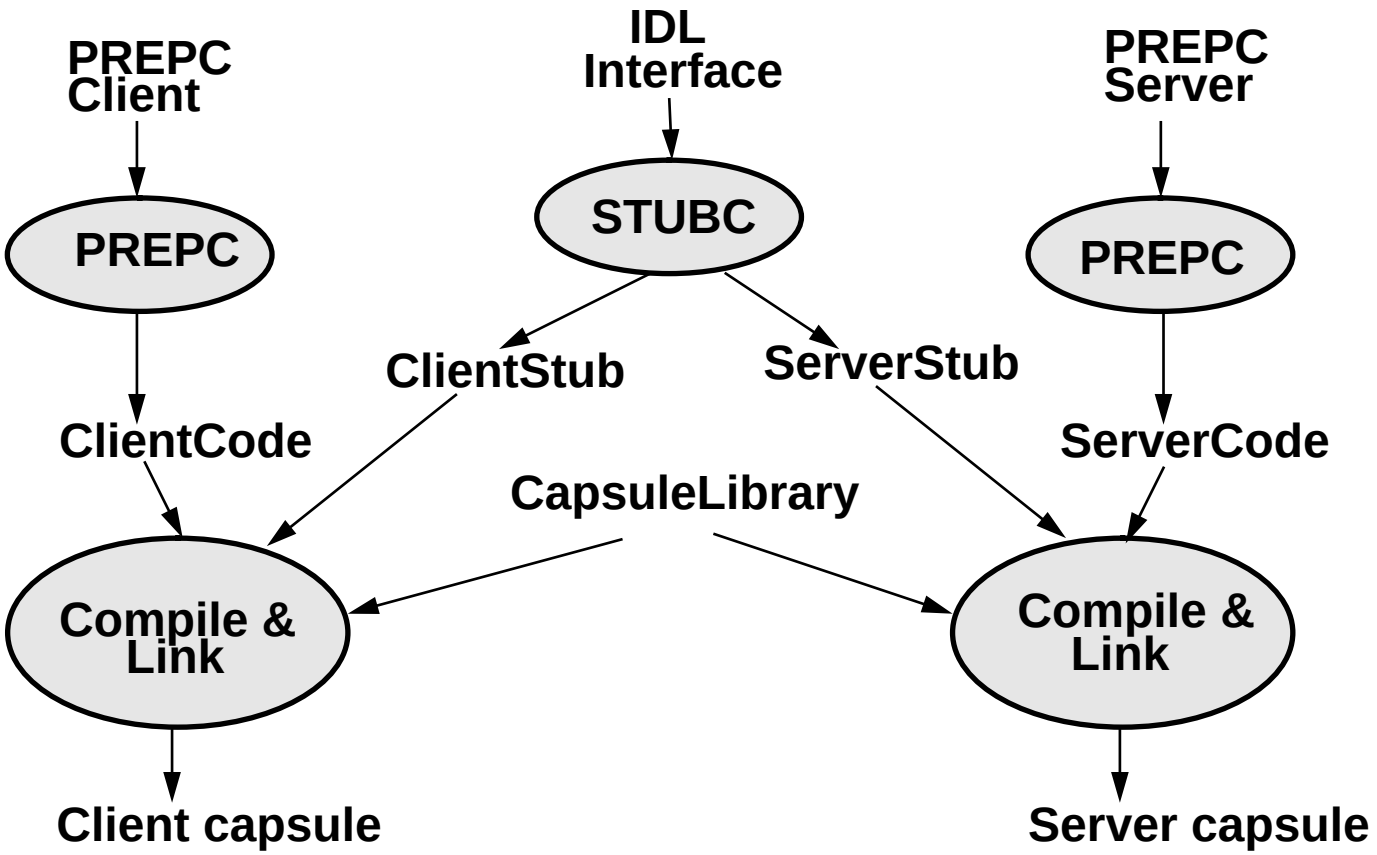


ANSAware Node, Nucleus, and Capsule - Implementation



- the Nucleus is linked into the capsule (as the capsule library)
- computational objects disappear

Building an ANSAware application



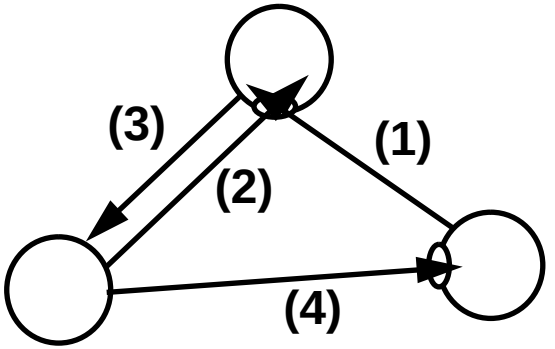
ANSAware Services

- ***Trader***: **provides run-time matching of service requests to available services**

- ***Factory***: **dynamic creation and destruction of services on a single node**

- ***Node Manager***: **per-node service management**

ANSAware Trader



- *Implements most of the ANSA trading concepts*
- *Allows a client to find a server which offers a service:*
  - of an appropriate (matching) type
  - with other appropriate characteristics (for example, Quality of Service)
- *Supports federation of multiple traders*
- *Allows administrative management of service offers*

The ANSAware Trader interfaces

- *The Trader service has 4 separate interfaces:*
- *1 service interface (of type Trader)*
  - for registering, looking up, and deleting service offers
- *3 management interfaces (of types TrType, TrCtxt, TrFed)*
  - for managing interface types
  - for managing trader contexts
  - for managing trader federation

Information in the Trader

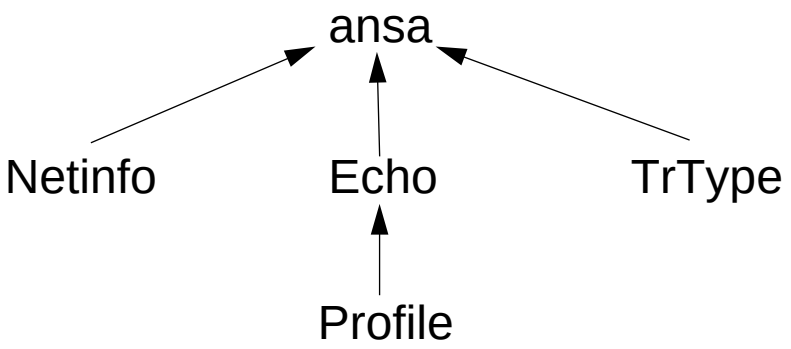
- ***The ANSAware Trader holds information about:***
  - **interface types**
  - **trader contexts**
  - **service offers and their properties**

Trader interface types

- ***Every interface type has a name***
  - **as given in the IDL**
- ***Each type has one or more immediate supertypes, to which it conforms***
  - **The root type is conventionally called *ansa* (it has no supertype)**
  - **The Relationship of new type to others is given when registering a type with the Trader**
  - **The Trader maintains a list of the type relationships**

The Trader type graph

- ***The relationships form a directed acyclic graph***
  - **more general than a tree, because types can have more than one supertype**



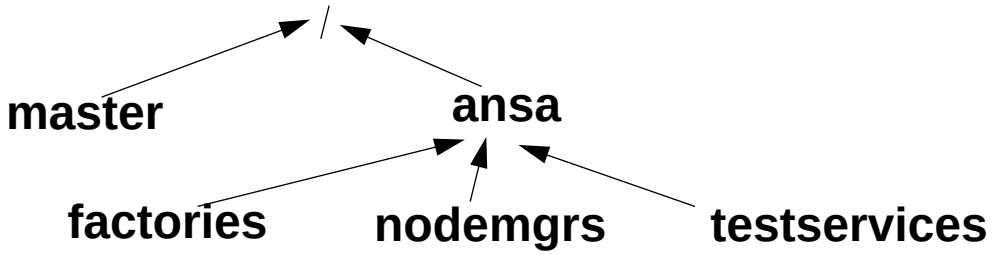
- ***Conformance of offer type to requested type ensures that client and server can exchange information***
  - **Type relationships are asserted by user, and checked at invocation time**
  - **Obviously, nothing can check whether the interface is correctly implemented; this depends on the implementations of client and server**

Trader contexts

- *Contexts allow administrators to group service offers together*
- *Every offer is registered in a context*
  - normally chosen by the application
- *Trader maintains a tree of contexts*
  - each context can register many service offers
  - contexts can contain other contexts within them (like Unix subdirectories)
- *The Trader uses context pathnames (like Unix pathnames)*
  - the root of the context tree is "/"

Administering Trader contexts

- **After a normal ANSAware installation, the context tree looks like this**

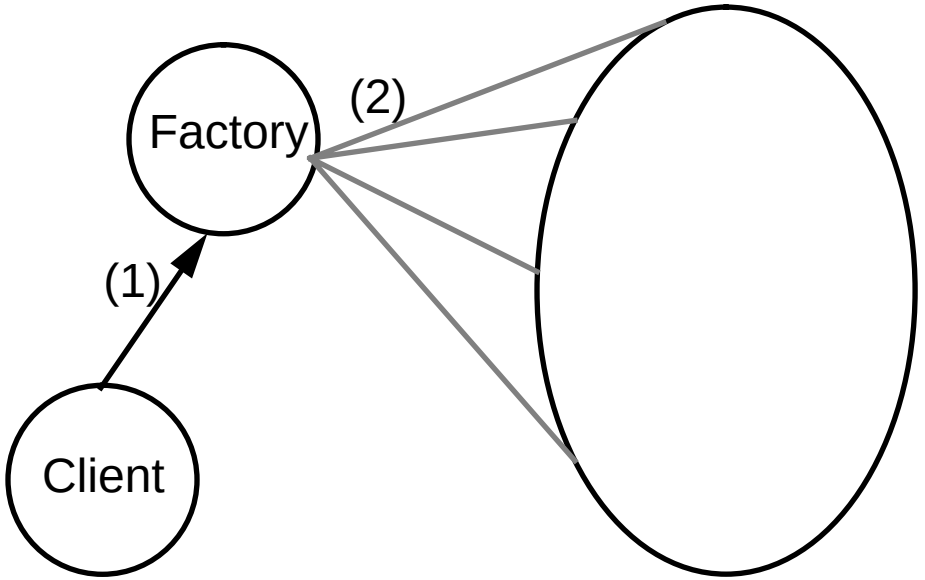


- ***Applications can create their own contexts and place offers in them if they wish***

Trader properties and constraints

- *Properties hold application-specific information about a service offer*
  - e.g. for a print service: LinesPerMinute, PaperSize, costPerPage, Turnaround
- *Property Name/value pairs are specified when offer is registered*
- *The Trader automatically generates some properties*
  - for example, it generates the (Type, TypeName) property for every offer registered
- *Clients requesting services may specify:*
  - Acceptable property constraints (e.g. PaperSize==’A4’)
  - Selection of offer with maximum (or minimum) value of a property
- *Constraint expressions can be as simple or as complex as necessary*

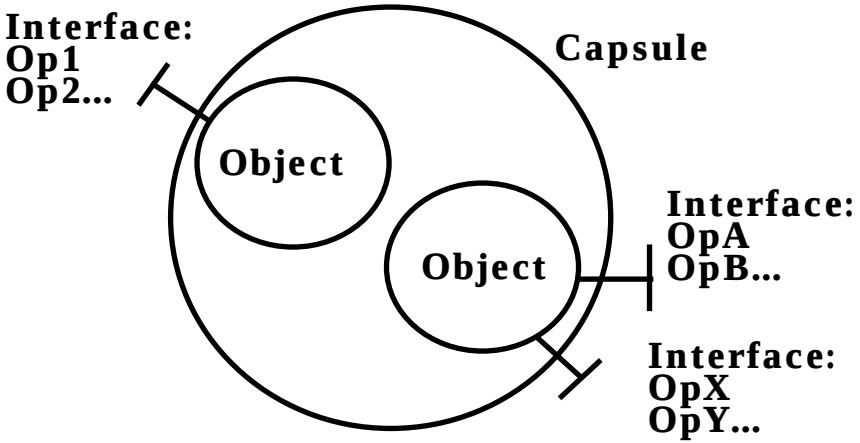
ANSAware Factory



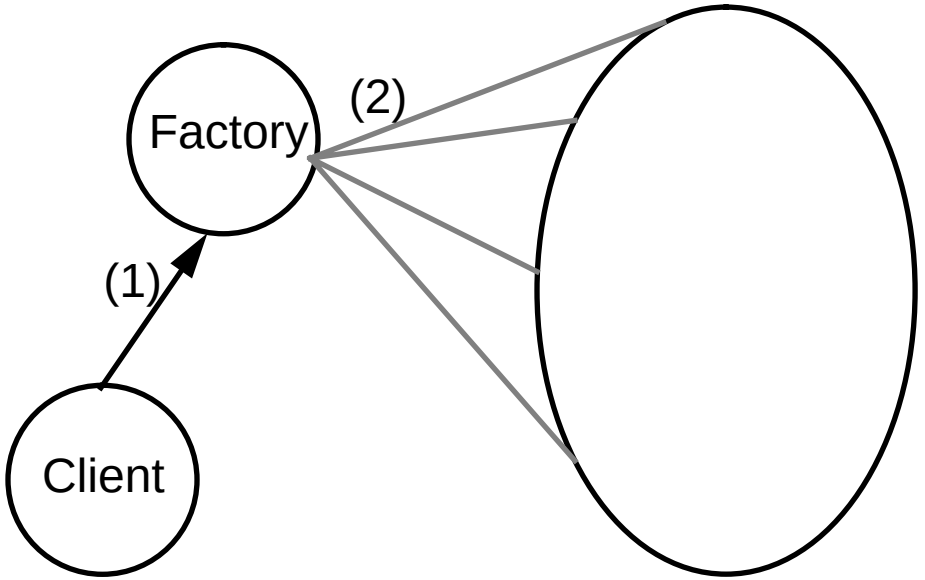
- ***Provides a service for:***
  - **Creation/destruction of capsules**
  - **Simple monitoring of the capsules it creates**

Factories and Capsules

- *Once created, capsules then provide a service for:*
  - Creation/destruction of objects within a capsule
  - Creation/destruction of interfaces within an object



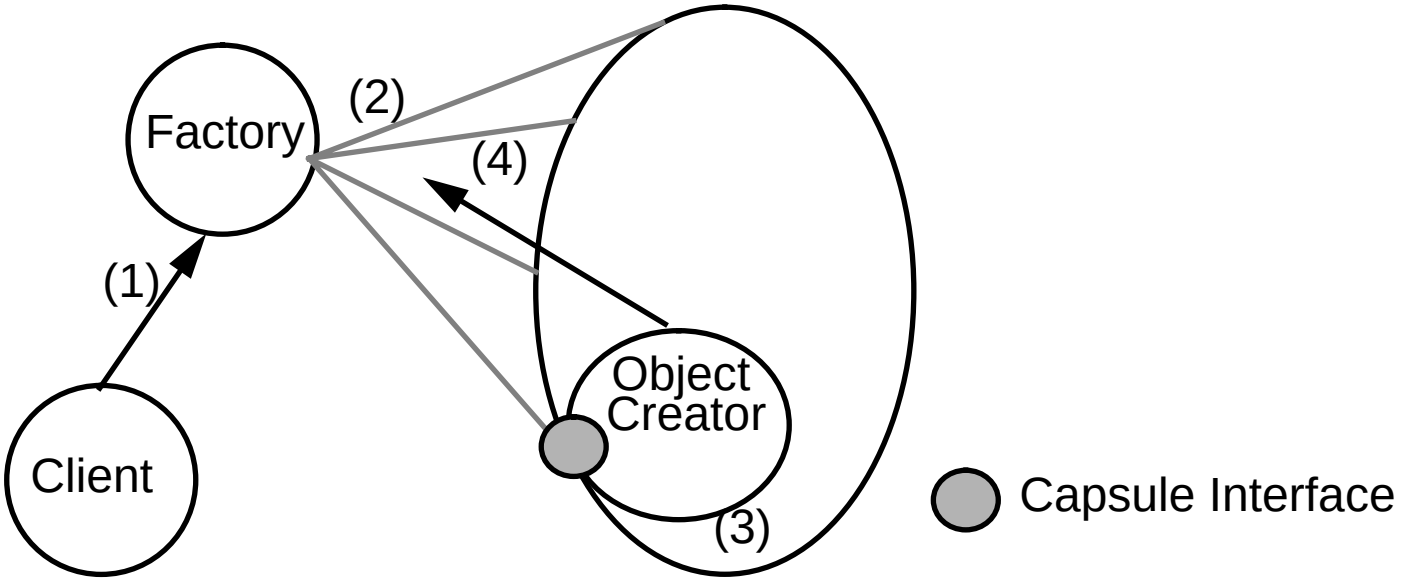
Capsule Creation



1. Client asks factory to Instantiate a capsule

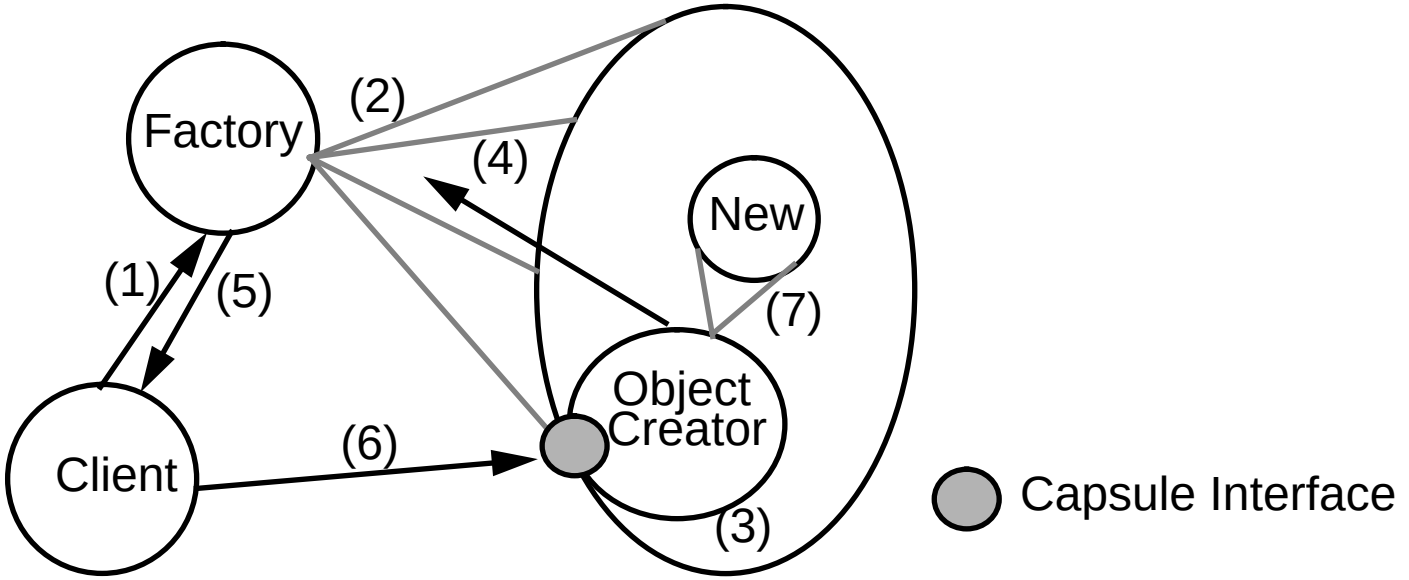
2. Factory instantiates a new capsule

Initial Object Creation



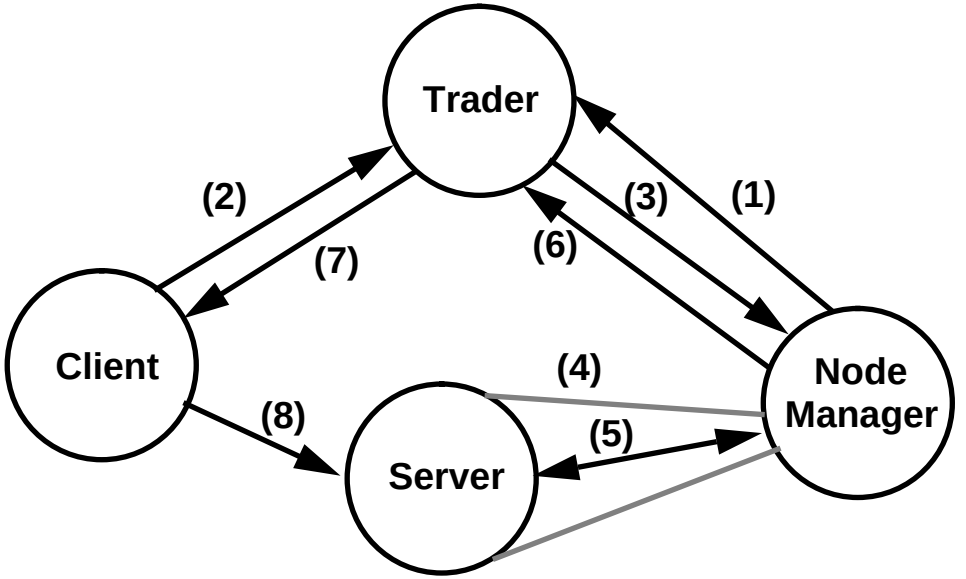
3. **New capsule creates a single object with at least a single Capsule interface. Any number of other interfaces may also be created**
4. **References to the Capsule and any other interfaces are returned to the factory**

Interface Creation



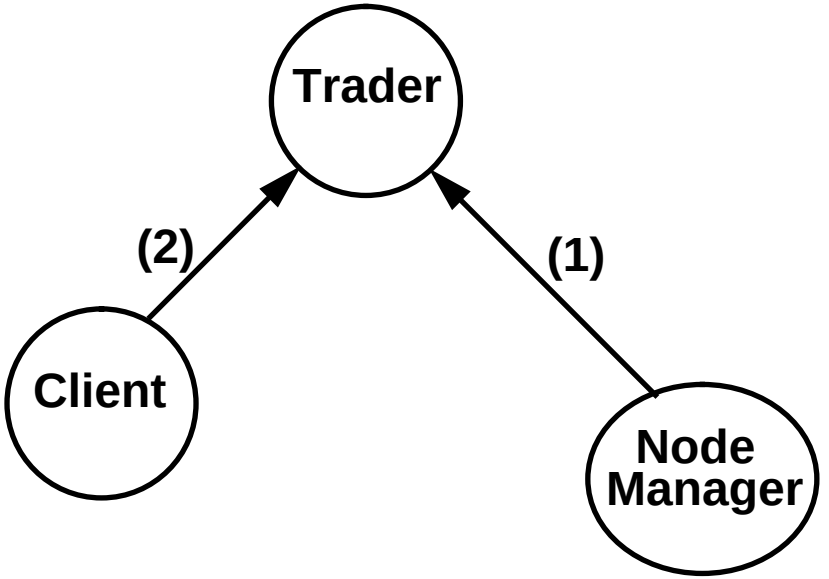
5. **These are in turn returned to the client**
6. **The client may then use the Capsule interface reference to create yet more objects**
7. **A new object is created**

ANSAware Node Manager



- ***Provides a service for the dynamic creation and activation of other services***

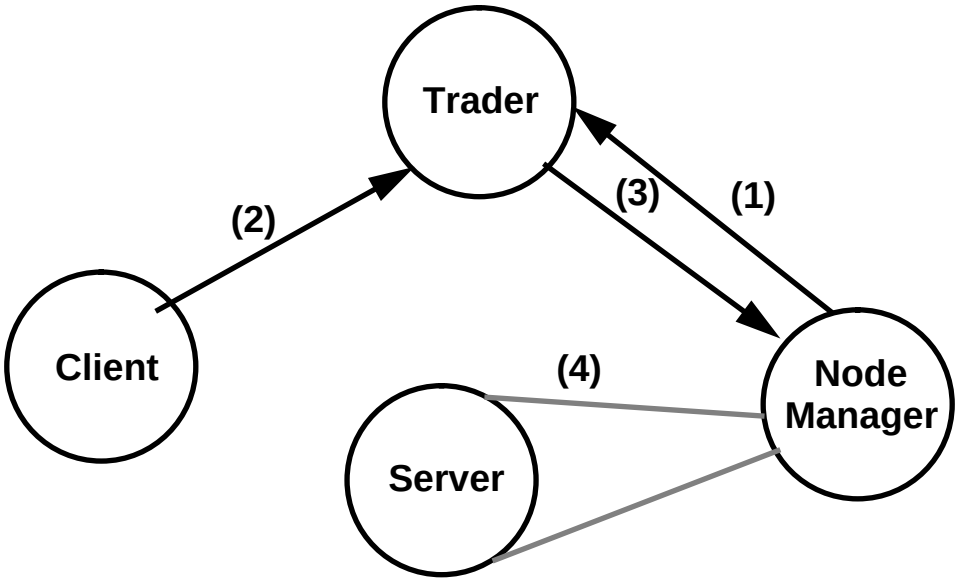
Node Manager - Registering the proxy offer



1. Node manager registers a proxy offer with the trader

2. Client performs an import

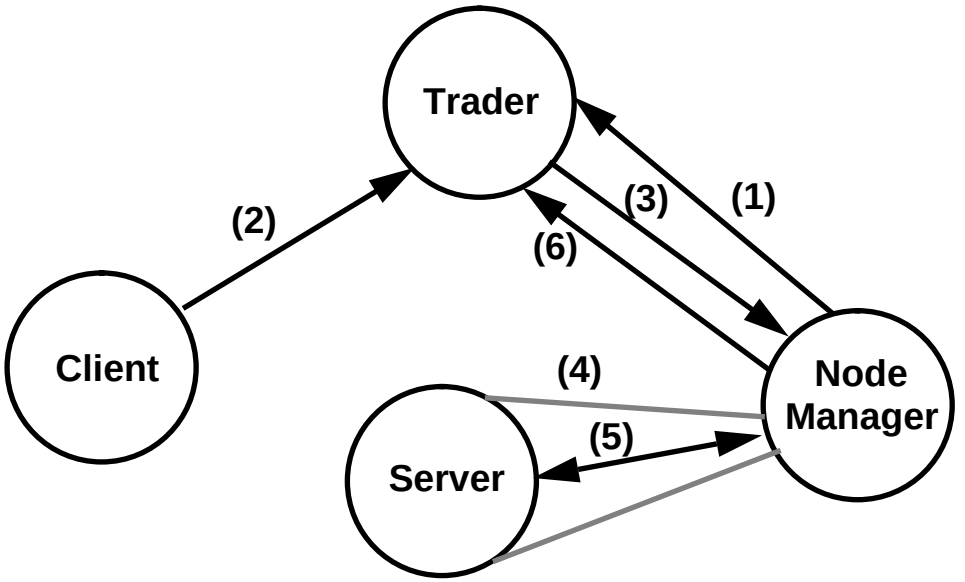
Node Manager - Creating the server capsule



3. Trader, recognising that the offer is federated, forwards the import to the node manager

4. Node manager creates the server capsule (using the factory)

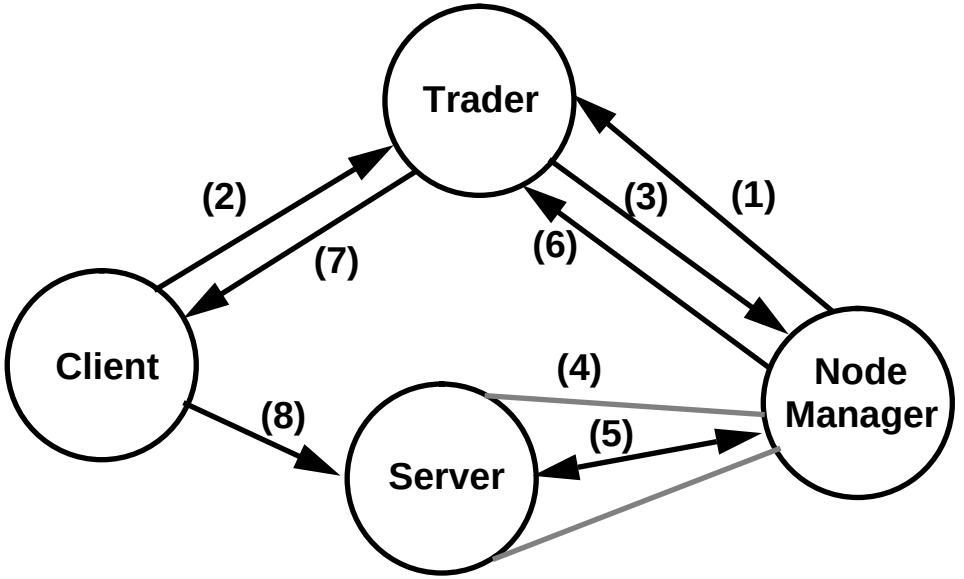
Node Manager - Instantiating the capsule



5. Node manager invokes the Instantiate operation on the newly created capsule's Capsule interface

6. Node manager returns the interface (result of Instantiate operation) to the trader

Node Manager - Return of interface

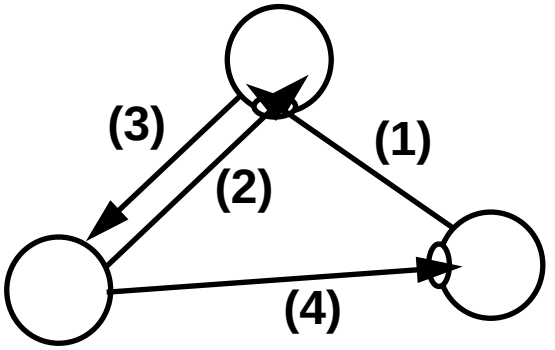


7. Trader returns this interface to the original client

8. Client can now invoke operations on the server

Trader, Factory, Node Manager

- ***Each of these services has interfaces that can be used independently***
- ***These services also cooperate to provide dynamic service creation***
  - **transparently to the client application...**
  - **... it only saw the ordinary trader interactions**



Summary

- *A light-weight, portable implementation of the ANSA Architecture*
- *Applications written in C with embedded ANSAware calls*
  - These calls are a language extension
- *ANSA Trader, Factory, and Node Manager services are provided*
  - for locating, creating, and managing services
- *For more detail, see Application Programming in ANSAware (ANSAware Volume B)*

ANSAware Supported platforms

- *Unix*
  - HP/UX (HP/UX version 7.0/8.0/9.0 on HP 9000 series)
  - SunOS (Sun3 and Sun4)
  - OSF/1 (DEC Alpha)
- *DOS/Windows*
  - bare DOS
  - Windows 3.1 386 Enhanced Mode
- *VMS*
- *... and others to come*

Other ANSAware platforms

- *ANSAware has been ported to many other platforms*
- *Some are in the ANSAware distribution*
- *... others are available on request*
- *And other products based on ANSAware are available elsewhere*