



**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**

ANSA Phase III

Trading in the WWW

Andrew Herbert

Abstract

This presentation is for the OMG / World Wide Web Consortium Workshop on Distributed Objects, Boston, June 24/25th 1996.

It outlines the concept of trading and explains its relevance to the WWW.

APM.1783.03

Approved
Standards Contribution

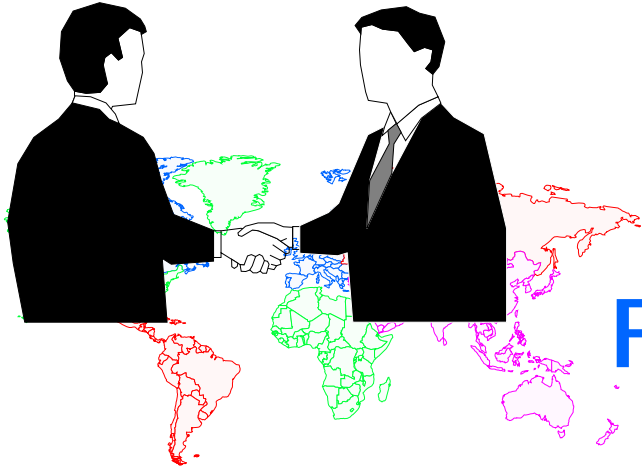
21st June 1996

Distribution:

Supersedes:

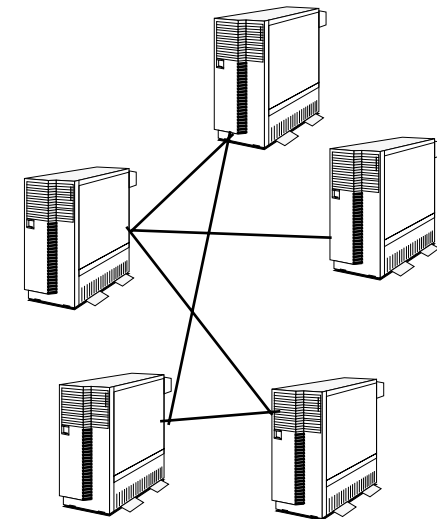
Superseded by:

Copyright © 1996 Architecture Projects Management Limited
The copyright is held on behalf of the sponsors for the time being of the ANSA Workprogramme.



TRADING FOR SERVICES ON THE WWW

Andrew Herbert
ajh@ansa.co.uk
<http://www.ansa.co.uk>



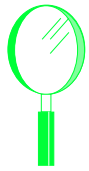


Name Services (and hence simple URLs) Aren't Enough

- using a name to bind to a service implies
 - I know in advance there are named objects which deliver the service
 - these objects will do exactly what I expect of them
 - in other words the **contract** between the importer and the exporter is *implicit* and *unchecked*.
- trading is about *making contracts* between importers and exporters
 - “Yellow Pages” for finding a suitable object in terms of *properties*
 - guarantees found object is of suitable *type*
 - discovery might include *instantiation* or *resource management*.



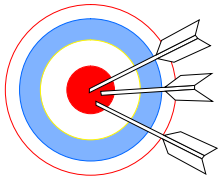
Different sorts of names: trading vs binding vs re-location



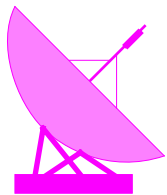
Trading finds a suitable object instance.
Trading fails if there is no such object.
Trading yields an *object pointer*



ORBs *bind* object pointers (long term names) to addresses (short term names); a binding fails if the target object fails or if the network fails



Rebinding gets you to the *same* object instance;
re-trading might get you to another object instance

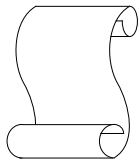


Re-location tracks an object to preserve a binding
e.g. across migration or recovery!

URLs can't easily do all of these at the same time!

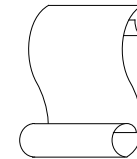
Key Trading Operations

Import



Type
Properties

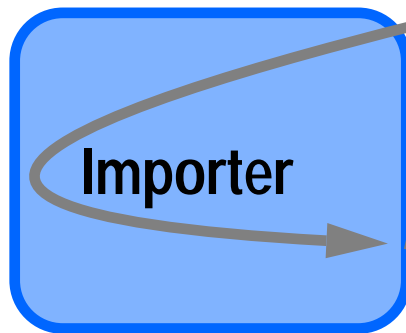
Export



Type
Properties
Object ptr



Withdraw

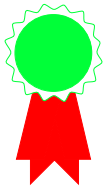


Contract between Trader,
Importer and Exporter



(BIND)

Type Matching



Type safety is a good thing
Compilers type check *programs* (local bindings)
Traders type check *systems* (remote bindings)



Minimum useful check is interface
signature conformance



Beware of adding extra checks that might prevent servers
and client from evolving separately



Trading and Properties



Signatures and/or interface UUIDs are not sufficient to choose a service



Who owns the service?



What will it cost to use?



Where is the service?



What does it do?

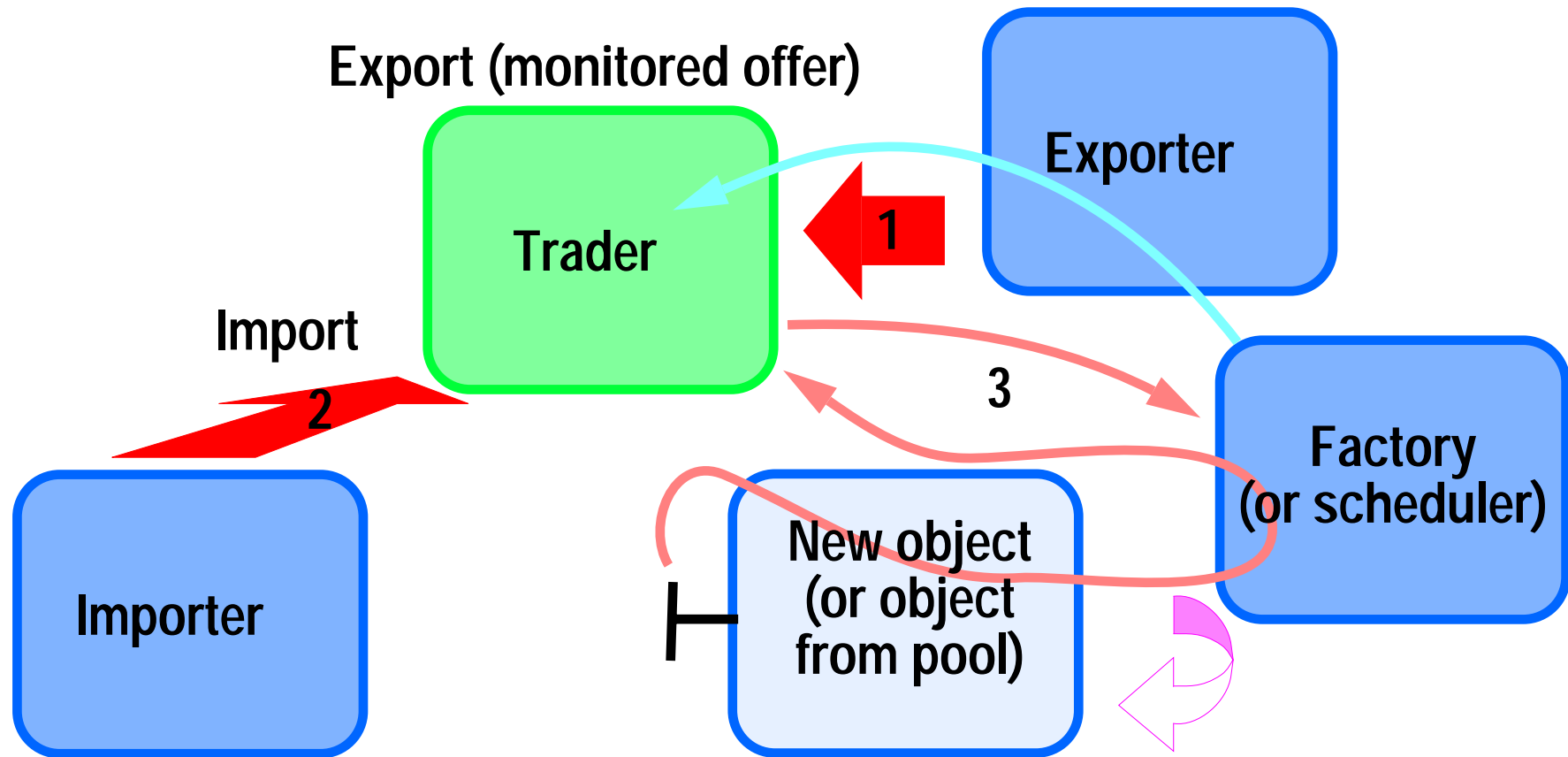


Can I bind to it?



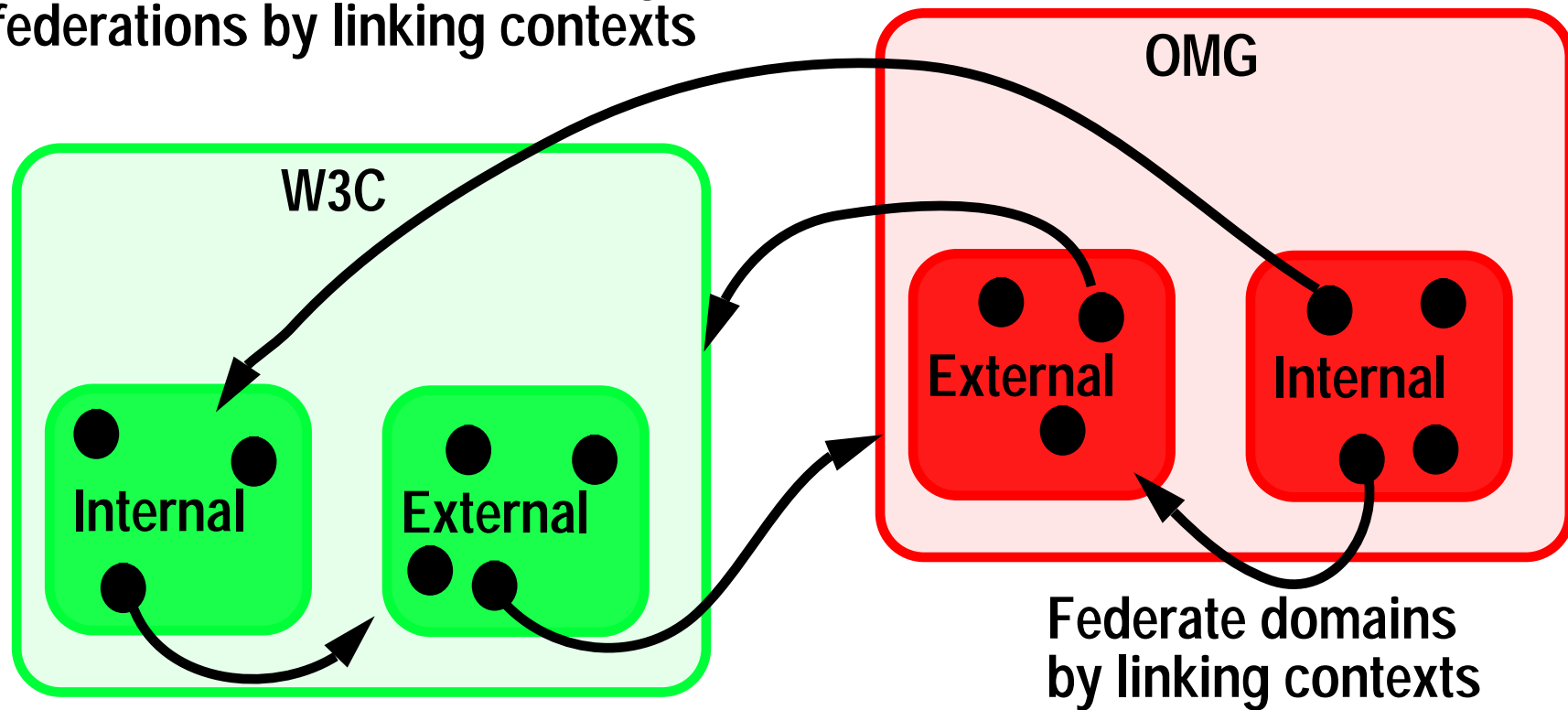
Qualify import with desired properties.

Trading and Life Cycle



Trader Federation

Administrators set up trading federations by linking contexts



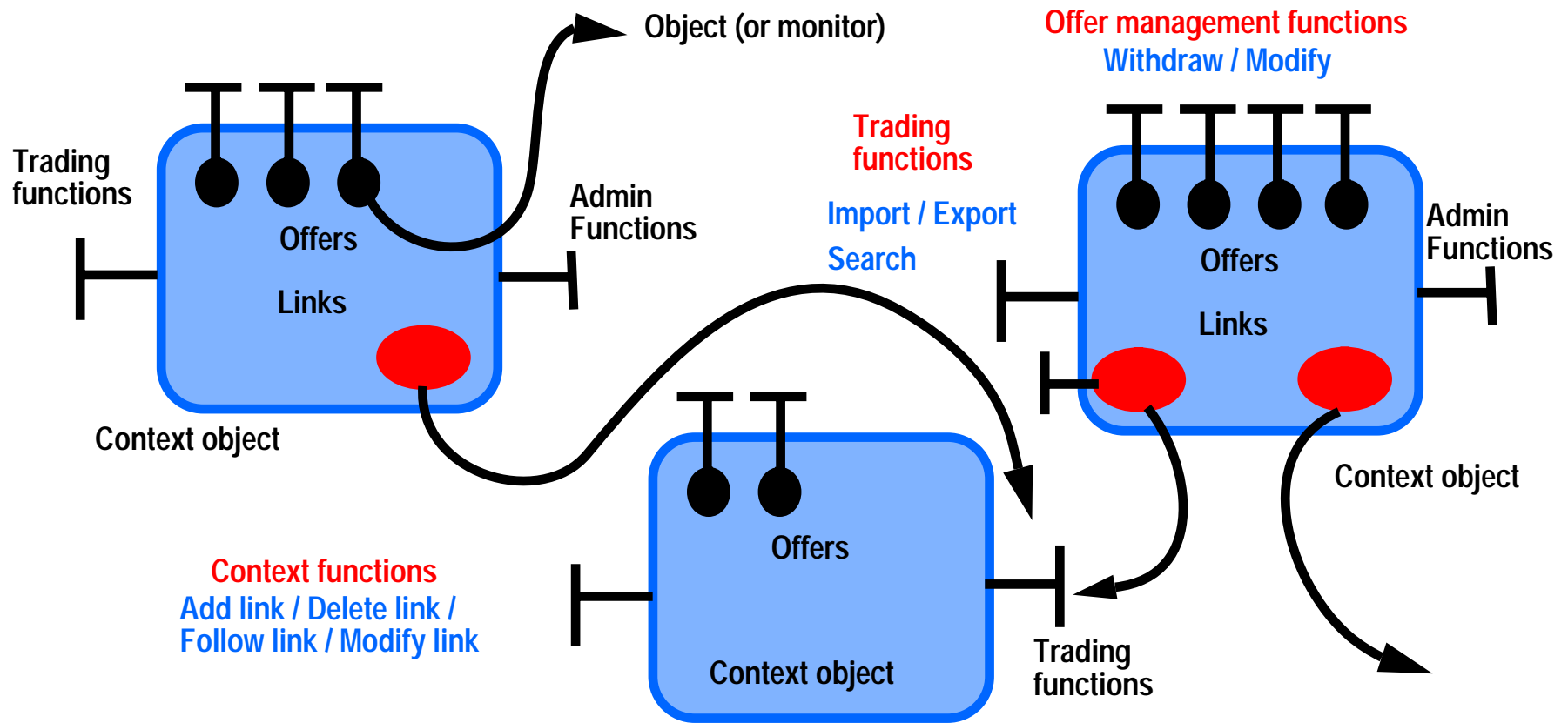
Federate domains by linking contexts



Benefits of Federation

- **Context relative naming**
 - Controlled visibility of your name space to others
 - Controlled aliasing of foreign name spaces for your users
 - An object's "initial" trader defines the object's universe
 - Conventions to aid usability (e.g. ubiquitous names)
 - "join in" model fits Internet culture better than global schemes
- **Choice of implementation strategies**
 - customized implementation (built-in resource policy, locality, ...)
 - leveraged name server (distributed, fault tolerant)
 - data base (volume, searching)

System Model





Traders and Downloaded Applets

- Trade to find the right server for the object (and the right version of the object)
 - more powerful than class -> file name mappings
 - Web server becomes a Web object repository (with integrated development and lifecycle tools)
 - Server life cycle made transparent to clients
- Context relative vs “global” URLs
 - controlled visibility
 - substitution / filtering of dangerous services
 - agreed conventions for naming “standard” functions



Traders and Object Type Systems

- **Assertion: signatures + context + properties gives practical disambiguity without need to manage a global name space**
 - property matching is evolutionary and flexible
 - name spaces are too rigid and authoritarian
- **Assertion: global name spaces are administratively fragile**
 - q.v. DNS security and performance problems
 - transparency getting in the way - context relative naming can side step this
- **Important properties: creator, location, version, quality of service characteristics**
 - what is the information model for Web objects?
 - where should the management information be stored (see later)?
 - are properties "guaranteed" - what is the trust model for federated traders?



Aligning WWW and Trader Concepts

- **Briscoe, BT: “a Web page is a trader”**
 - set of object pointers (alternative locations) plus description
 - references have open structure enabling arbitrary processing
 - context relative naming
 - description is plain text, but authoring tools could structure the meta data
- **Browser facilities for client side processing of URLs, driven by policy**
- **Analyse web page idioms and define higher level concepts (e.g. alternative URLs) to simplify user model**
- **Conventions for structuring URLs to permit wrapping of alternatives**
 - c.f. CORBA IORs
- **Investigate means to distill system management information from content**



Trading and the WWW Issues

- Should URLs be structured like CORBA IORs?
- Should URLs be interpreted relative to a trader?
 - impact on URL vs URN debate?
- What system information do we want for Web objects?
 - What kind of contract (type) system do we want?
 - Where will we keep contract (type) information?
 - HTML content? HTML semantic tags? URNs? URLs?
- What life cycle support do we want for Web objects?
- Should the WWW be used for the user interface to traders?
 - Why stick at traders, why not to all managed objects?