



---

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

TELEPHONE:  
INTERNATIONAL:  
FAX:  
E-MAIL:

**Cambridge (0223) 323010  
+44 223 323010  
+44 223 359779  
apm@ansa.co.uk**

---

## **ANSA Phase III**

# **Slides for Property Repository talk at TC, Jun 94**

## **Gray Girling**

### **Abstract**

The promotion and advertising of existing and forthcoming electronic services and their efficient location by potential clients are important prerequisites for the wide scale deployment of an electronic services market place.

Trader servers provide advertising and location services but current implementations are not easily updated to make use of newly developed technology, nor to extend the range of information about services that they hold.

This document describes the goals of the work on the property repository current in the federation task group at the end May 1994 as then reflected in document APM.1177.

This set of slides is used for the TC meeting of 6 and 7 March 1994. Their presentation should take about 30 minutes.

---

APM.1232.00.01

**Draft**

1 June 1994

Request for Comments (confidential to ANSA consortium for 2 years)

---

**Distribution:**

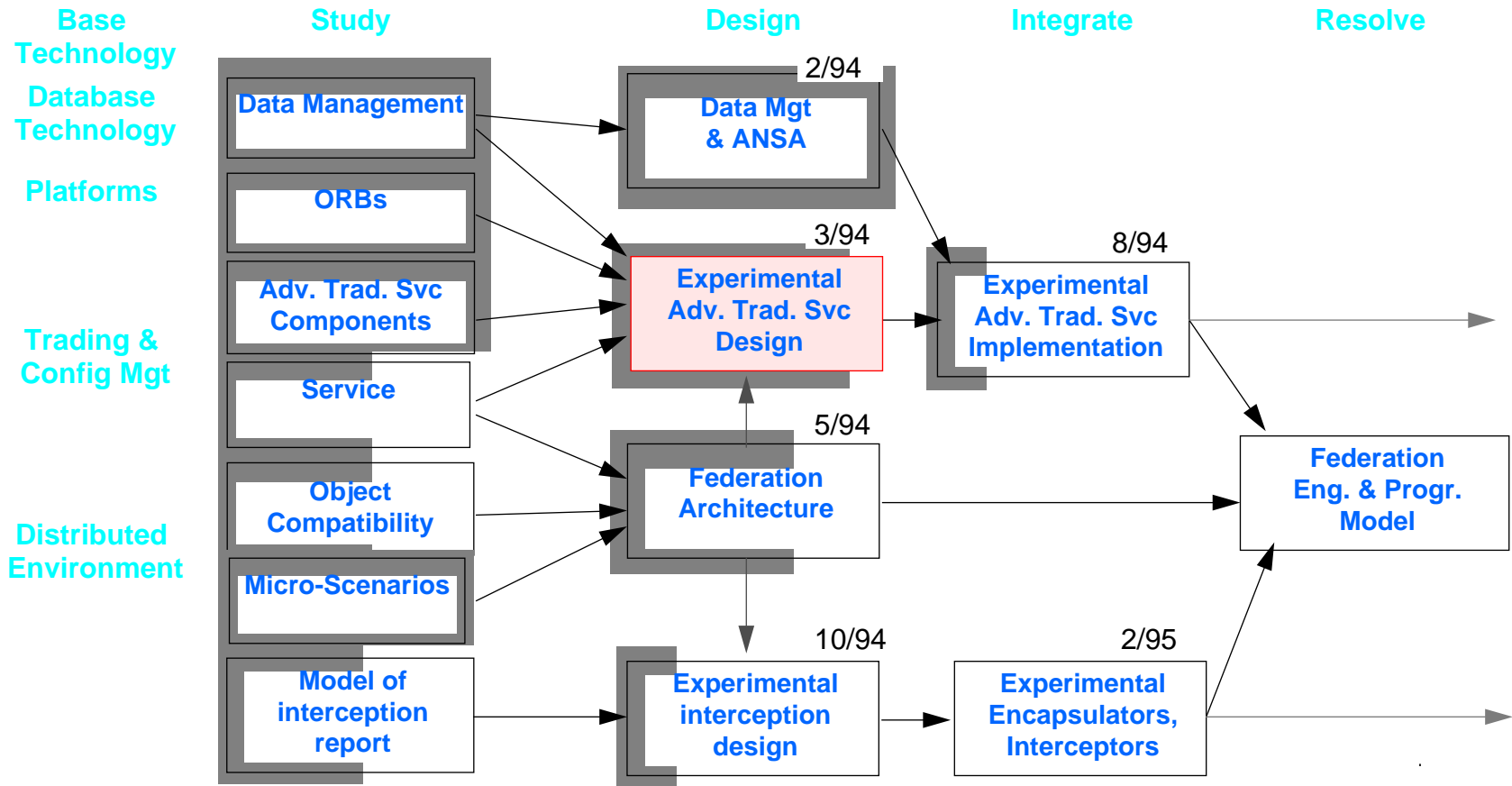
**Supersedes:**

**Superseded by:**





# Federation Work





---

## Services Providing Information About Servers

- ☆ **Traditionally:** address servers, “name servers”, X.500 directory
- ☆ **Since ODP:**
  - ❖ traders (interface type key, properties),
  - ❖ type managers (interface types)
- ☆ **Now:**
  - ❖ CORBA interface repositories (IDL + stubs, skeletons, debugging information, formatters, browsers)
  - ❖ CORBA implementation repositories (all kinds of instantiation information)
- ☆ **ANSA Phase III:**
  - ❖ **federation:** information to describe authority, management, remuneration, infrastructure ... domains (are they different? is there some compatibility?)
  - ❖ information for creating interception infrastructure (e.g. for performance, dependability)
- ☆ **We need a prototype to support ad hoc requirements**

**Are these all shadows of a larger requirement?**



## An “Advanced” Trader?

- ☆ Federation requires support for co-operation at a number of levels not just binding
- ☆ A new service should potentially be capable of holding:
  - ❖ interface types, interface references – but also:
  - ❖ “implementations” – object templates
  - ❖ local interceptor factory interfaces
  - ❖ management, remuneration policies
  - ❖ access control information, QoS specifications, client requirements
  - ❖ manual pages (e.g. for World Wide Web) ...
- ☆ Not an “advanced trader” a “property repository”
- ☆ Overview:
  - ❖ Design and Implementation Goals
  - ❖ Limitations of the Proposed Design



## Goals – Use as Trader

- ☆ **Important initial use of prototype**
- ☆ **Support a wider variety of property type**
  - ❖ some traders support only number, string and set of string (others unspecified)
  - ❖ want to support structured property values (e.g. public key <key, signature> certificate, QoS characteristic)
  - ❖ ideally want to support BLOBs (e.g. HTML manual, implementation template)
- ☆ **Represent contexts as objects (as ODP)**
  - ❖ contexts are labelled areas into which offers can be categorized
  - ❖ ANSAware and (until recently) ODP distinguished these from traders



## Goals – Use as Trader (cont.)

☆ Incorporate functionality provided by traditional trader interface:

**Table 1: Import/Export**

ANSA Trader	ANSAware Trader	ODP Trader
Register	Register	Export
LookUpAll	Lookup	Search + List Offer
RegisterMonitor	ProxyExport	Export
LookUpOne	Lookup	Select
(Offer.withdraw)	delete, (ProxyDelete)	Withdraw
		Replace = Withdraw+Export

**Table 2: Trading context management**

ANSA Trader	ANSAware Trader	ODP Trader
Link	BindContext, AddName	Add Link
Unlink	UnbindContext, DelName	Remove Link
Contexts	ListNames	List Link Details
Select		(Resolve)
SelectWithPolicy		(Resolve)
		Modify Link = Remove Link+Add Link



## Goals – Extensibility

- ☆ **Need a flexible basis on which to mount sets of information experimentally**
- ☆ **Support an extensible range of named properties**
  - ❖ *property profile*: the set of named properties supported by a property repository
  - ❖ add definition of new property types and named properties dynamically
- ☆ **Partition named properties**
  - ❖ properties with the same purpose grouped into property “sub-profiles” (e.g. one for performance properties, one for dependability properties ...)
  - ❖ properties do not all have to be declared at offer creation time (e.g. development phase properties such as implementation specifications)
  - ❖ properties can be declared by independent property providers (e.g. remuneration properties don't have to be specified by access providing infrastructure)



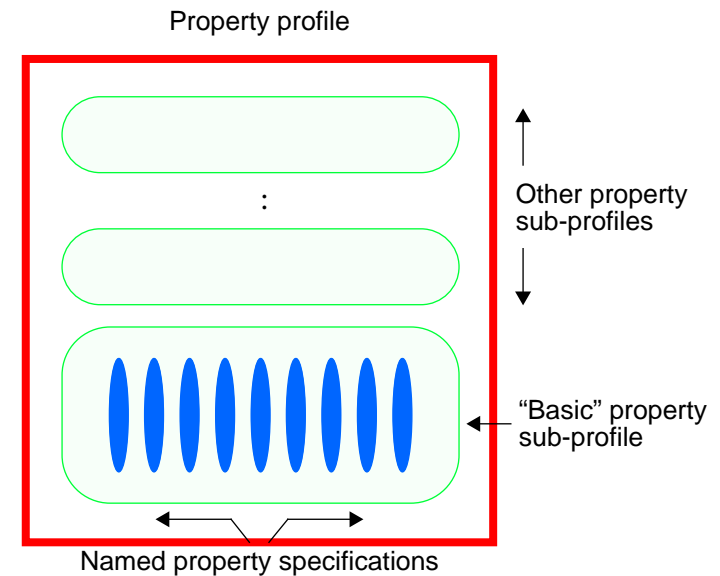
## Goals - Extensibility (Cont.)

☆ All repository property profiles consist of

- ◆ an integral number of property sub-profiles
- ◆ a “basic” property sub-profile

☆ The “basic” property sub-profile has properties to support “trader” functionality:

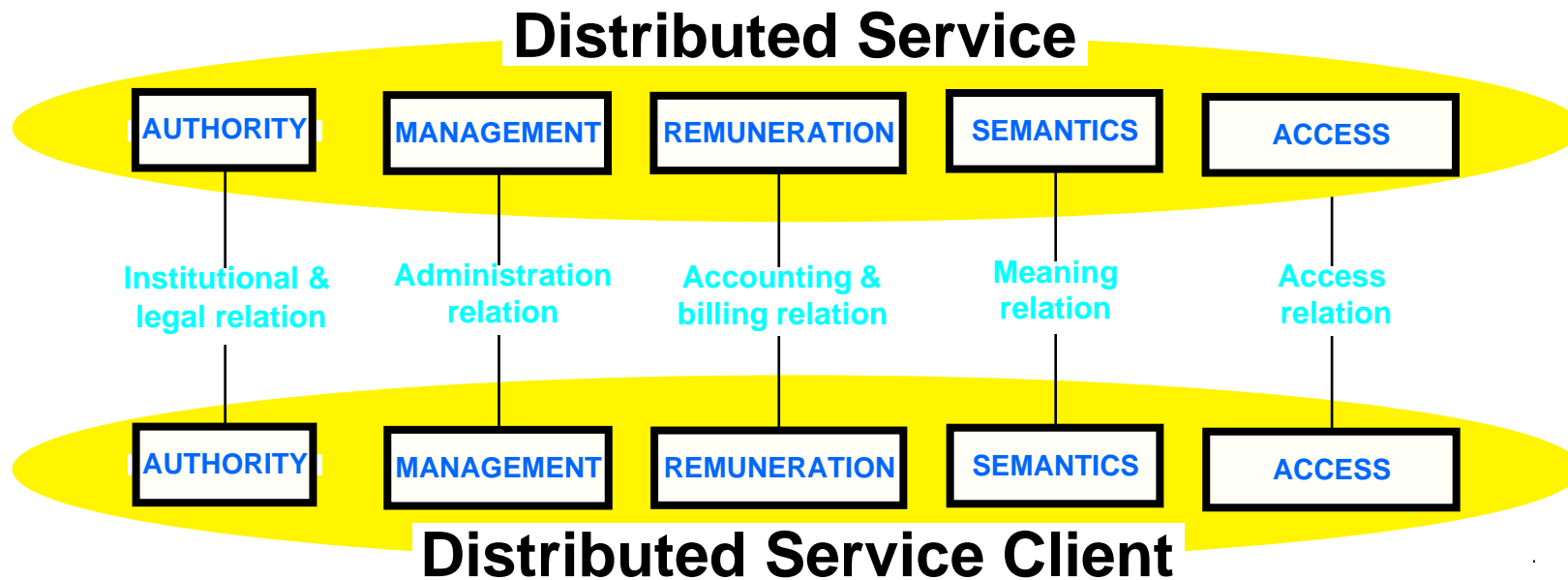
- ◆ interface types
- ◆ interface offers



## Goals – Extensibility (Cont.)

### ☆ Property Sub-Profiles and Federation

- ◆ can be used to support the information required to fulfil achieve federation goals
- ◆ different sub-profiles could be developed to support different parts of federation: agreement registration, search, selection, boundary resolution





## Goals – Proxy and Generic Offers

- ☆ **Feature of federation and performance work –**
  - ❖ binding may require use of special purpose engineering components
  - ❖ components may vary with requirements on service required
  - ❖ sometimes efficient to create on demand
  - ❖ sometimes variety of components too great to store one offer for each (e.g. application dependent ORB-ORB interceptor)
- ☆ **Provide information about dynamically selectable services**
  - ❖ either located in a pre-existing selection or created on demand
  - ❖ offers can refer to proxies and contain instantiation parameters
- ☆ **Associate a range of services with an offer**
  - ❖ allow an offer to refer to generic services (e.g. Stack.of[X])
  - ❖ would also allow an offer to refer to a set of trader exports (e.g. export instantiation)
- ☆ **Separate interface reference seeking and information seeking**
  - ❖ don't want servers created just to obtain information about them



---

## Goals – Implementation

☆ **Use relational database management technology**

- ◆ large amount of data (e.g. from BLOBs for specifications) – integrity & durability
- ◆ need to export data management from repository – don't want to re-invent the wheel – RDBMS technology is available
- ◆ concurrency control yields potential performance gains

☆ **Some implementation issues covered in Gomer's talk**



## Limitations

- ☆ **No support for property repository federation (sub-profile use, search policies)**
- ☆ **No client-server symmetry**
  - ◆ no client offer (guarantees) storage
  - ◆ no client based requirements
- ☆ **Access to structured information is not object orientated (as CORBA interface repository is)**
- ☆ **Implementation priorities initially exclude (included later)**
  - ◆ signature based subtyping
  - ◆ accessible property repository creation
  - ◆ automatic distributed searching
- ☆ **Implementation priorities will initially include (excluded later)**
  - ◆ statically typed repository interactions