



Naming

What's in a name?
That which we call a rose
By any other name
would smell as sweet
Romeo and Juliet II:2
William Shakespeare (1564-1616)

Slide 4
A linguistic entity
Of interest to the caller
Distinguished from others
Agreement of real world model

Slide 10, 19 & 24
Aliases
Synonyms
Change and update

Slide 8
Naming conventions
Handle & remainder

Slides 7 & 8
Name may imply properties
Contexts, sets, domains



In this Session

- ***Naming concepts, in general, will be discussed***
 - what are they, what are the problems
- ***Naming issues of distributed systems will be emphasised***
- ***Naming systems will be compared and contrasted***
- ***Enough detail will be given to:***
 - recognise deployment of naming concepts, even where implicit in applications or infrastructures
 - design a simple naming system



The Structure of the Presentation

- *General naming principles*
 - Concepts
 - Rules
 - Application
- *Distribution and Federation issues*
 - Name services
 - Name resolution
 - Interface references
- *Examples*
- *Summary - taxonomy of issues*
- *Conclusions*



Definition - what is a name?

- *A name is a linguistic entity that denotes a particular entity from a group of entities. [Hauzer 86, Comer 87]*
- *Entities (to which names refer) may be anything of interest*
- *An entity named must be distinguishable from all other entities*
 - but the entity could have more than one name
 - there is an implied agreement about the real world model
- *A name may substitute the (logical) action of pointing*



Use of Names

- *An attributive name may stand directly for the entity*
 - for two parties to refer to a third - X and Y talking about Z
- *An invocation name may stand for a pointer to the entity*
 - to select one target from many - X calling Z
 - to name a pointer to a target - X referring to (Y which names Z)
- *Invocation names are of great importance in distributed systems*
- *A name itself may be named*
- *A name may form part of a predicate*
 - eg, cmm in “a paper given by cmm”



Approach to Naming

- *Naming may be implicit - hidden within applications*
- *Naming may be explicit - through name management services*
- *Explicit naming may be:*
 - **Global - an assumption of universally unique names, or**
 - **Context relative - unambiguous only within a local context**
- *Naming schemes are not exclusive. For example, use a:*
 - **centralised scheme in a local context. eg within a business**
 - **context relative naming in federations. eg between businesses**



General Naming Concepts

- ***Naming domain - answers which entities can be named***
 - thus defines the “Universe of Discourse” - a set of distinguishable entities
- ***Name sets - answers which names are allowed or prescribed***
- ***Naming context - relates names to entities . Within a context:***
 - all names must be drawn from one name set
 - all entities must be members of a single naming domain
- ***Naming network - expresses constraints:***
 - can entities named in one context be referred to from another?
- ***Binding :***
 - noun - that x stands for an entity***
 - verb - the act of setting x to stand for an entity***



Naming Conventions

- **Name: A predicate that defines a name set - defining a style**
 - eg: /usr/tmp/foo; apm@ansa.co.uk; 192.005.254.006
- **Properties: Names may imply some properties**
 - eg: file.bas; file.sit; file.doc; file.eps.sit
- **Names may be structured in any chosen format - subject to:**
 - a handle (context if present or name if not)
 - plus remainder (name if context present or null if not)

cmmfile

handle: cmmfile

remainder: null

cmm@ansa.co.uk

handle: @ansa.co.uk

remainder: cmm

/usr/tmp/cmmfile

handle: /user/tmp/

remainder: cmmfile



Naming Rules

- ***Anything at all can be named, including:***
 - names themselves
 - pointers to entities and names
 - entities, names or pointers outside the local system
- ***Any symbols can be used***
- ***The naming network can be structured in any way***
- ***All names shall be structured with handle, plus remainder***
- ***No single outermost context can be assumed***
 - ie, no optimisation from assumptions about global context
- ***Also, no innermost context may be assumed (eg security)***



Application of Names

- *An entity may have more than one attributive name*
 - within a context - aliases
 - from more than one context - synonyms
- *An entity may have only one invocation name - eg, an address*
- *A name may be unresolved - not bound to anything*
- *Names may be dynamic - eg, such as with version control*
 - same name bound to a different entity
 - » but only one at a time
 - same entity with a modified or updated name



Naming Domains

- *Entities may be spread over many domains.*
 - eg, to delineate authority, security, management, etc
- *Administration may be hierarchical*
 - with transitivity from domain to subdomain
- *Administration may be in federations*
 - with domains disjoint
- *Context relative naming is necessary in federations*



The Case Against Global Naming

- *Need autonomy for different administrations*
 - how could global agreement be achieved?
- *Need to avoid committing to centralised management*
 - who would be responsible for the central service?
- *Need to avoid scaling problems as name universe grows*
- *Need to operate within different legal and political systems*
 - in a conflict, which law would preside?
 - what sanctions could apply against dissidents?
- *Need flexibility to allow evolution - partition, mergers, etc.*

If global naming worked, there would only be one scheme in the world



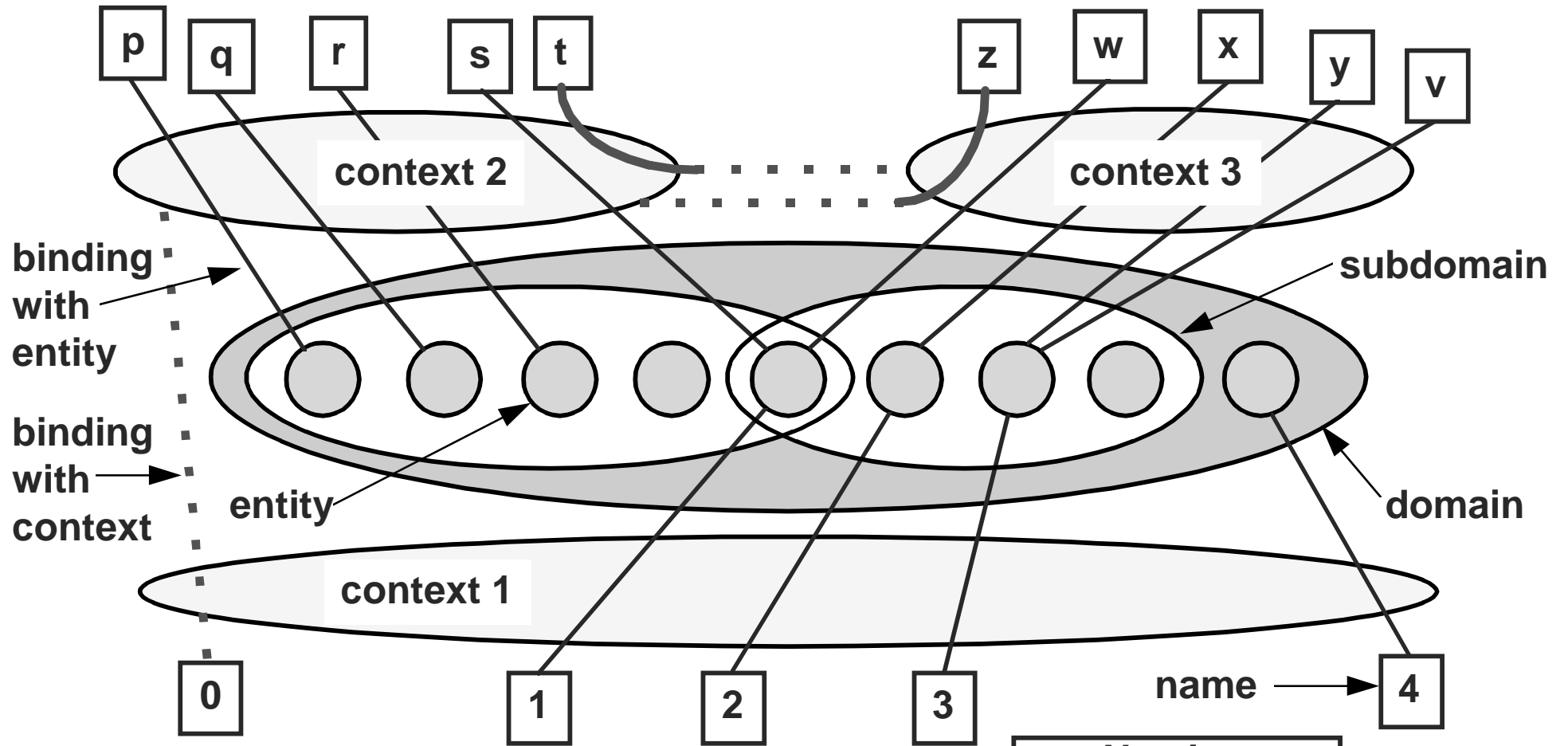
The Case for Context Relative Naming

- *Avoids the global naming problems - as discussed*
- *Applies a name entity binding only within a local context*
- *Allows the existence of multiple naming “roots”*
 - *thereby tolerates “multiple global schemes” (sic)*
 - *thereby allows aliases*
- *Forces the context to be declared - thus avoiding errors*

And the price paid

- *Possible name translation at context boundaries*
- *Possible holding or passing of alien names*
 - *name cannot be resolved in the local context*

Naming domain with subdomain and contexts





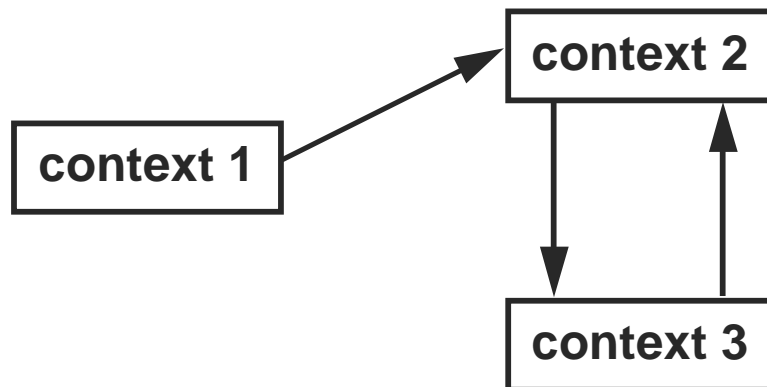
Context Relative Names

Name from context 1	Name from context 2	Name from context 3
0.p	p	z.p
0.q	q	z.q
0.r	r	z.r
1 or 0.s	s or t.w	w or z.s
2	t.x	x
3	t.y or t.v	y or v
4		

From the previous diagram



Naming Network



context 2 can be reached from context 1
contexts 2 and 3 can be reached from each other

Consistent with the previous table



Federation

- *Autonomous action - not subject to a central control*
- *Choice with whom to associate, including:*
 - when to enter into an association
 - when to leave an association
- *Choice with what services and data to offer, accept, and share*
- *Choice of perspective on the rest of the world.*
Each component has:
 - its own local naming system
 - its own view of another's naming system



Names in a Federation

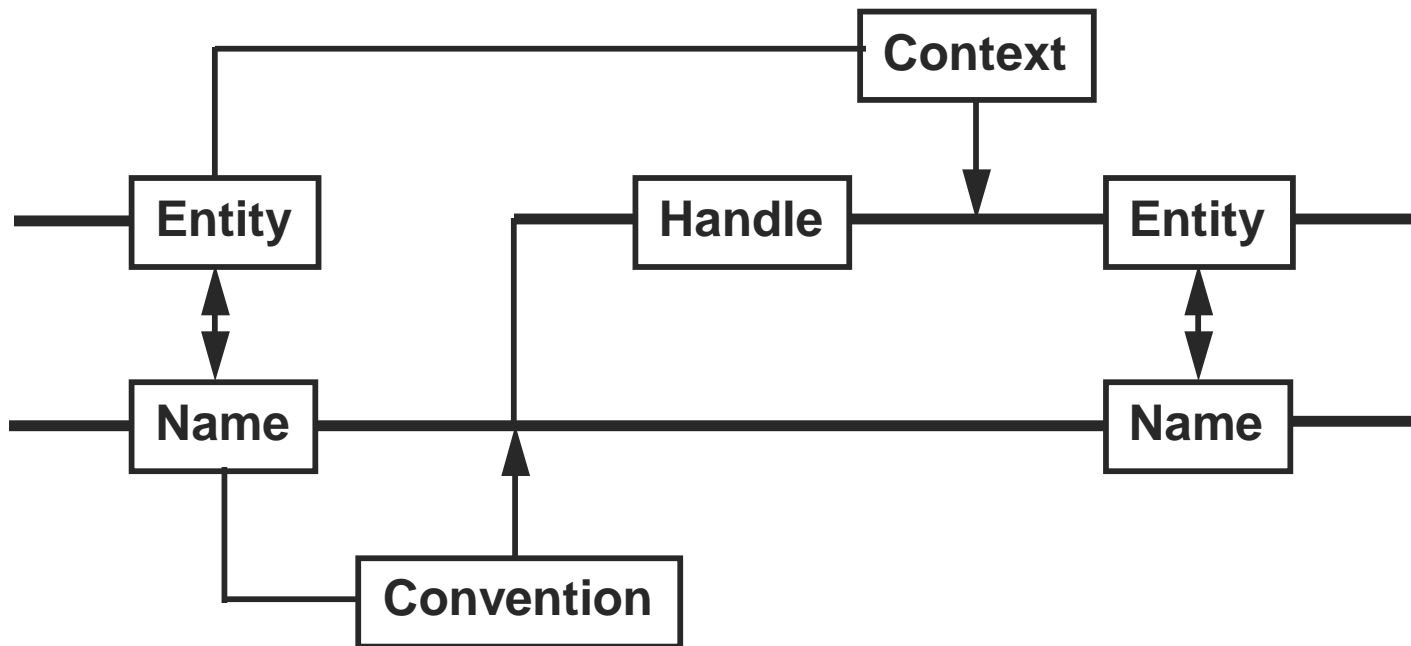
Federation of several naming systems create a new naming system in which:

- *each naming system may export a specially created naming context to allow others to access selected entities*
- *each naming system may import a naming context, for referring to entities in other naming systems*
- *each naming system maintains its own naming conventions*
- *naming between entities is always indirect*
 - through an import export context pair



Name Resolution

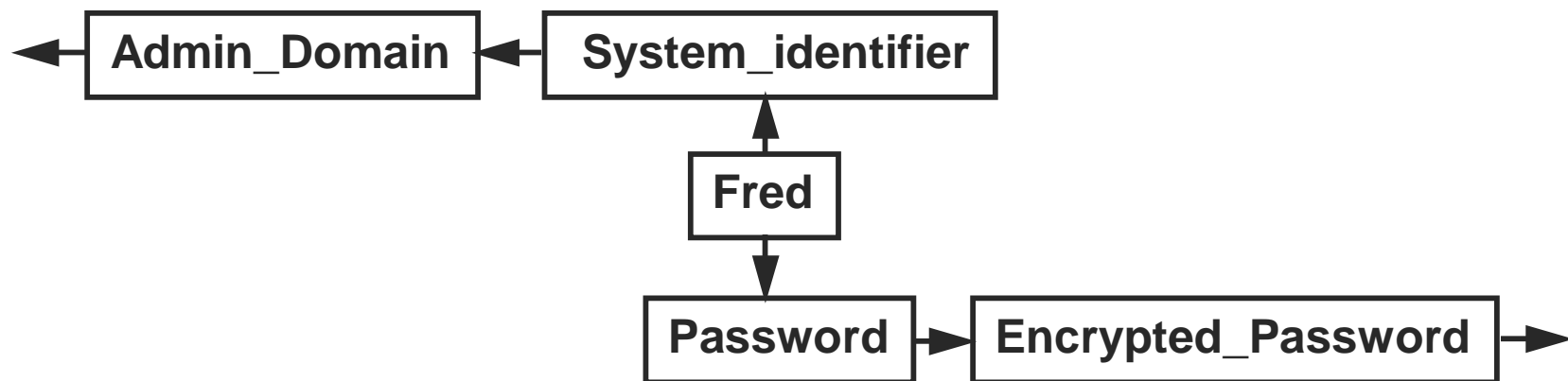
The process of taking a name and context to produce a new name (remainder) and new context (from the handle)





Naming and Security

- *There are no absolute identities*
 - an object identity is a name for (named) object attributes
 - an object secret may be a structured name
- *Name resolution may go to wider or narrower contexts*
- *Narrower contexts are more likely to be implicit*





Interface References - IRs

- *Introduce indirection between names and the entities they denote*
- *When an application sends a name:*
 - bind the name to a reference, and the reference to the entity
 - make the indirection transparent to both sender and receiver
 - allow different sender and recipient attributed names
 - allow the recipient to imagine a direct name-entity binding
- *Optimise when authorities align name services*
- *IRs offer an opportunity (not a right) to use a service*
- *Thus, IR features and the naming model are closely related*
 - ANSA IRs comply with the ANSA naming model



Relationship between IRs and names

- *The naming scheme employed in the IR is context relative*
- *The IR behaves as a data type, like other data types*
- *The IR can contain information that:*
 - supports binding for several different environments
 - lists creator supported protocols
 - provides the context relative server port address
 - allows an end-to-end check (a nonce)
 - can be used to locate servers that have migrated
 - can reflect the properties of a server group
- *An IR is NOT a capability - it confers an opportunity; not a right*



Name Service

- ***Maintains a relation between attributive and invocation names:***
 - **Attributive:** usually descriptive - meaningful to humans
 - **Invocation:** usually non-descriptive - meaningful to computers
- ***A name server provides an interface to a name service***
 - note that the name service may be distributed or federated
- ***Connected entities may exchange sets of symbols. They interwork when the symbols convey both syntactic and semantic meaning.***
Ways to exchange service are:
 - pass an entity name (attributive), and insist on name generation and resolution in the recipients' context
 - pass a reference (invocation name), and claim the same effect if used in the senders' context



Name Service Example

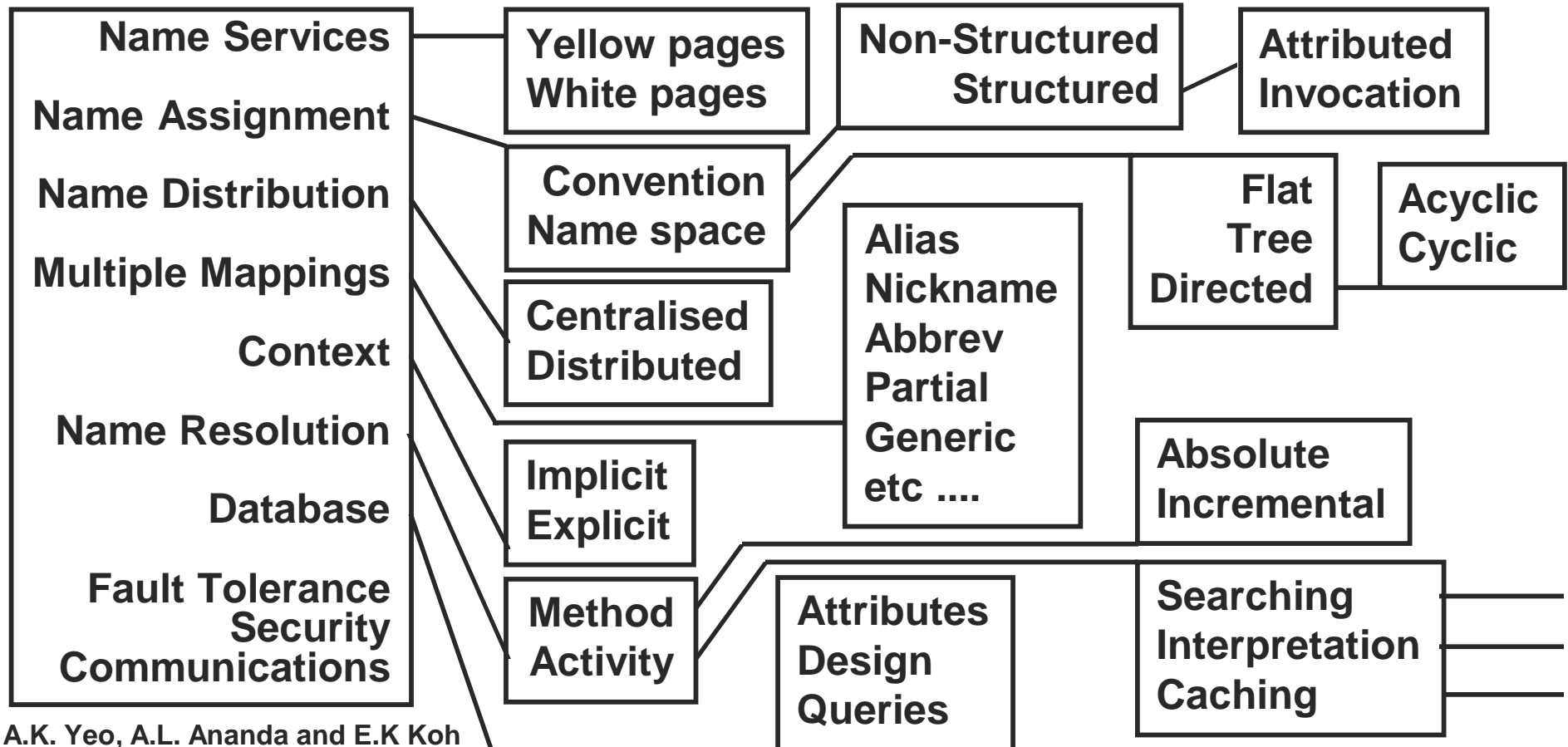
- **add_name(object_id, name, user_id)**
- **remove_name(object_id, name, user_id)**
- **change_name(object_id, user_id, old_name, new_name)**
- **resolve_name(name, user_id): set_of_object_id**
- **translate_name(name, user_id, user_id):set_of_names**
- **list_users_of_name(name, object_id):set_of_objects**
- **list_names_by_user(object_id):set_of_names**



Examples of Directory Systems

- *Internet DNS*
- *OSI - X.500*
- *Naming components of the Unix file system*
- *Naming within applications: Excel, Databases,*
- *OSF/DCE's CDS*
- *NetWare's NDS*
- *SunSoft et al - XFN*

Taxonomy of Issues in Name System Design



A.K. Yeo, A.L. Ananda and E.K Koh
OS Review. Vol 27, No. 3, July 1993



Conclusions

- *There is a clear distinction between identity and name*
- *(Truly) global naming schemes are flawed - they do not scale*
 - but names may be centrally administered in a local context
- *A naming system is always present*
 - even if implicit in applications or infrastructures
- *Naming systems carry considerable navigation connotations*
- *A naming system design is fundamental to system quality*
 - poor naming system design implies
poor integration into Open Distributed Systems



References

- *Further information can be found in:*
 - ANSA Architecture Report - AR.003.01 Feb '93
 - Saltzer J.H.
“Naming and Binding of Objects”
Operating Systems: An Advanced Course
Bayer et al, (Eds). Springer-Verlag, Berlin 99-208 (1977)
 - Comer D.E. & Peterson, L.L
“Understanding Naming in Distributed Systems”
Distributed Computing 3(2), 51-60 (May '89)
 - Other references listed in AR.003.01