Distributed Systems

ID2201

name services

Johan Montelius

Name service

- To communicate with an object we need to find the object, the address to the object, its access methods etc.
- A name service will give us the attributes associated with a name:
  - an address
  - the services
  - ...
- This is called *resolving a name*.

Examples

- File systems: file name
  - *file descriptors*
  - *size, type, owner ...*
- DNS: domain name
  - IP address
  - email server, owner,...
- RPC, CORBA, Java RMI: object name
  - remote reference

Directory service

- Directory services are related creatures where we use a database query to find the attributes of an object
- Examples:
  - Global Name Service
  - X.500
  - LDAP

Names

- Accessing objects using names instead of addresses allows for relocation of objects.
  - domain names and web services
  - phone numbers and mobile phone connections
- Who is allocating names?
  - ICANN – domain names
  - ITU – phone numbers

Name / Address

- What is a *name* and what is a *address*?
- An address in one system could be a name in an underlying system.
  - domain names - IP address
  - IP address - MAC address
- Is an address a *sequence of names?*

Flat name spaces

- A flat name space
  - the name does not hold a structure
  - objects with related names does not share any properties
- Easy
  - to compare two names
- Trouble
  - need the full name
  - creating groups
  - delegating rights to create names

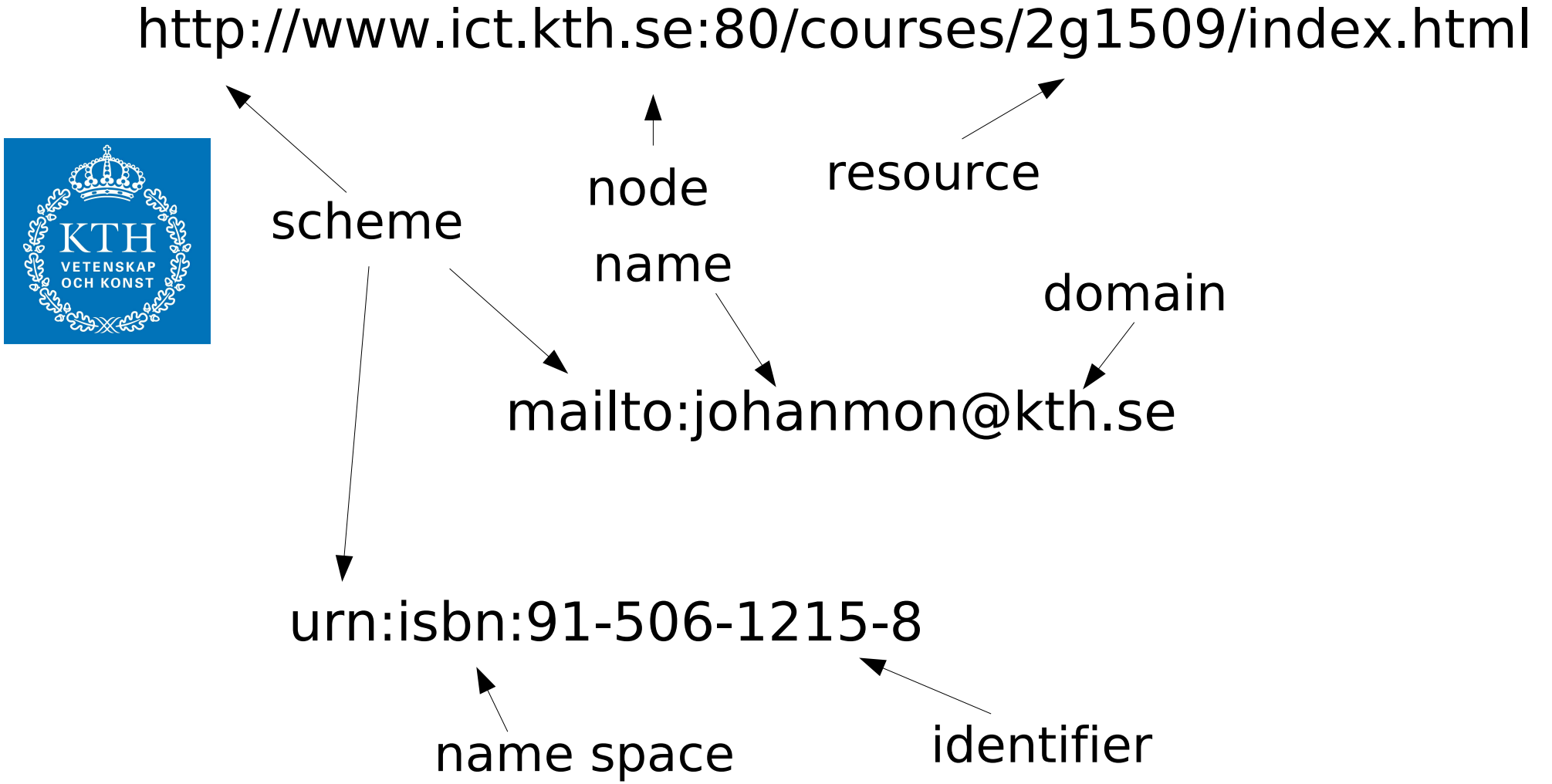
Hierarchical name spaces

- hierarchical name space
  - each name is resolved in its context
  - potentially infinite name space
- global or local names spaces
  - domain names
  - phone numbers are global but resolution is done in a local context

Global or local

- is a name local to ..
  - an application
  - a server
  - a local network
  - the Internet
- Important when we deal with distributed systems, a name can be bound to different objects depending on the context.

URI - uniform resource identifiers



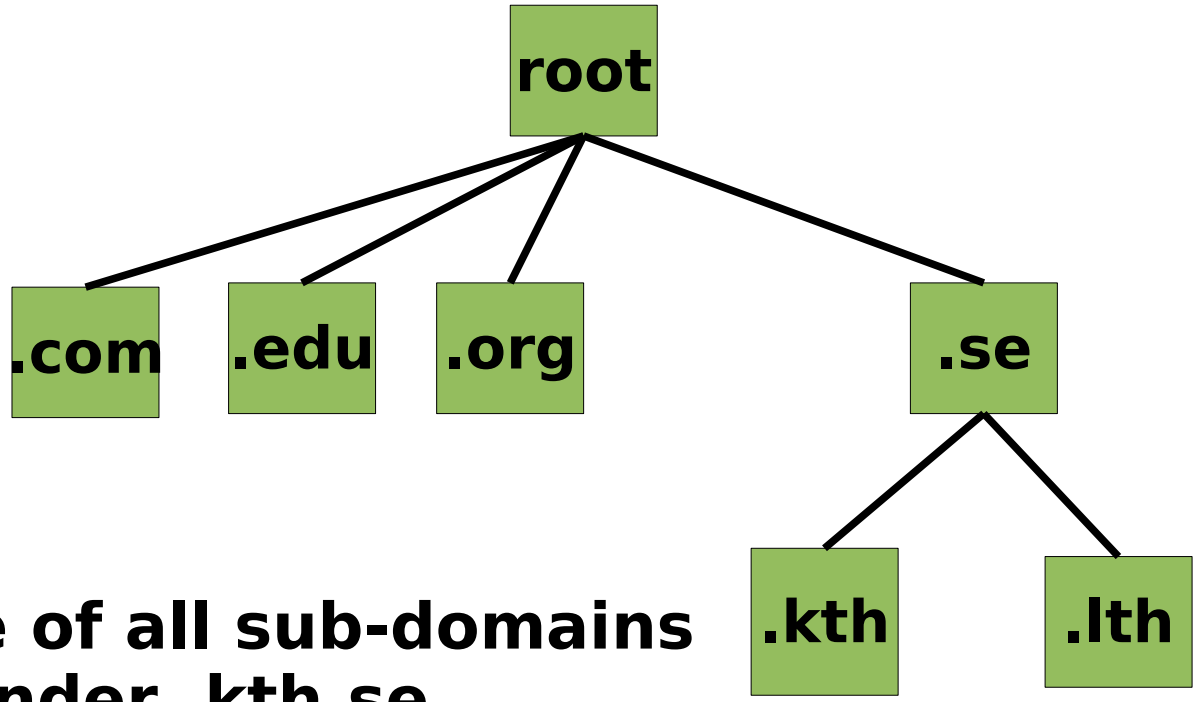
DNS

- Originally the name space was flat and stored in the *hosts* file on each client.
- It worked for a while,but:
  - problem finding new names
  - central authority
- John Postel developed DNS in -82, finally defined in Mockapetris RFC 1035 -87
- Grown from a few thousand entries to over 100 million entries.
  - That's scaling!

DNS names and attributes

- A name consist of:
  - a top-level domain
  - possibly a sequence of sub-domains
  - possibly a host name
- A name is mapped to a set of attributes:
  - IP address
  - mail server
  - host, operating system .. ??

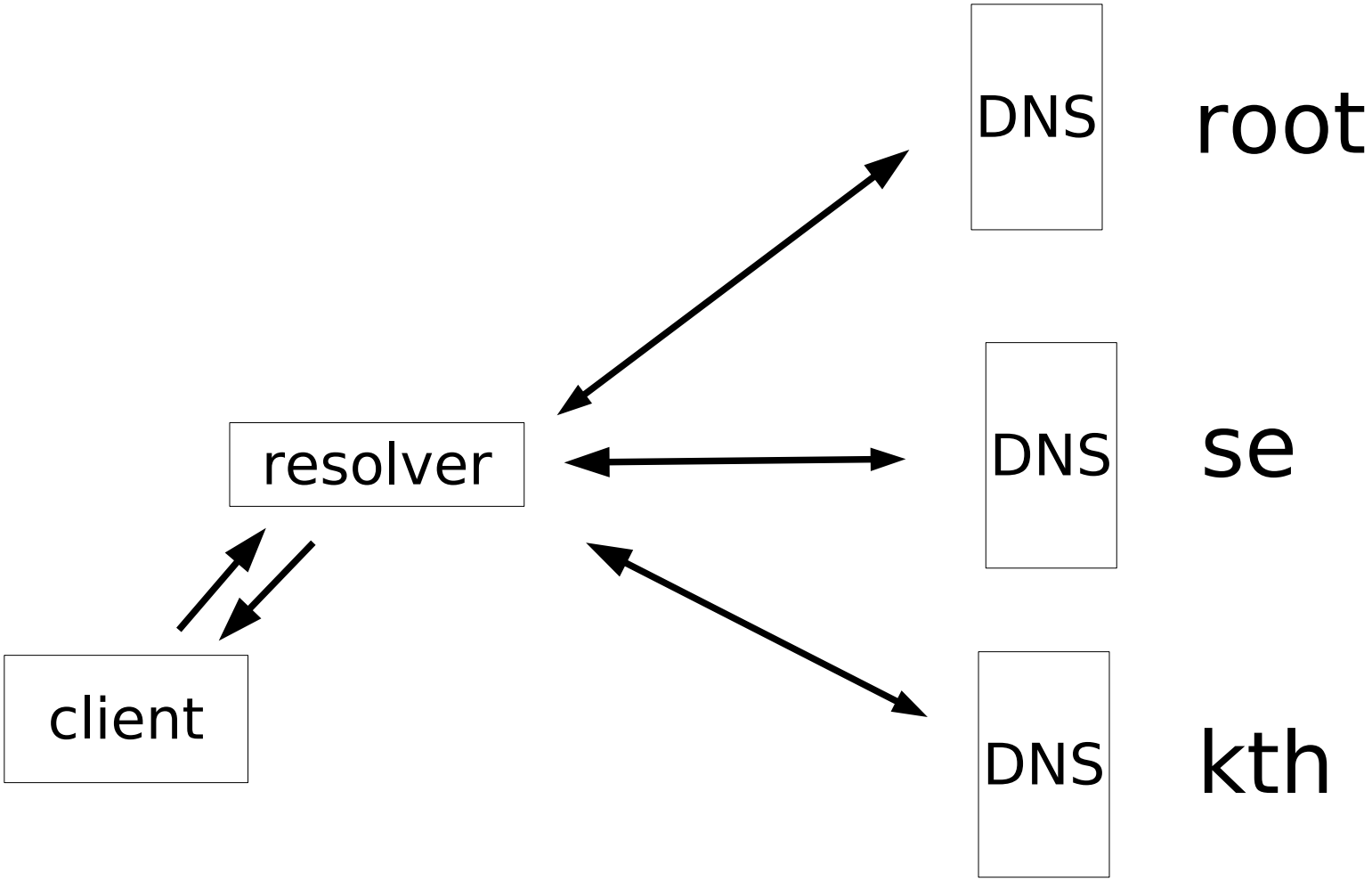
Server architecture



Responsible of all sub-domains and hosts under .kth.se

Also secondary server for .lth.se

DNS resolution



Resolution

- Client sends a request to a resolver.
- Resolver will query the root and iteratively move its way down.
- All replies are cached by resolver.
  - time out specified in reply
- Cache is soon filled with most important DNS servers and host entries.

Inconsistency

- What happens if an attribute is updated?
  - How are cached copies invalidated?
  - Does it matter?

Replication

- There are thirteen roots (each operating several distributed servers) with replicated content.
- http://www.root-servers.org/
- http://k.root-servers.org/

More

- Is DNS a one-to-one mapping?
- How can DNS servers be used to load balance access to a service?
- Can DNS give us mobility?