

Principles for naming & configuration (1)

- Assume a back-end database (relational)
- It should be possible to use file(s) instead of database (e.g. for testing)
- Do not assume a central database (PS, SL, ST are administrated separately)
- Define API, access method hidden (Corba server, JDBC, Jan's directory service API ?)
- Provide a generic query-based service for
 - server configuration
 - device queries

Principles for naming & configuration (2)

- Separate generic name resolution (e.g. device name to server name) from MW specific resolution (e.g. broker address). Corba and MoM use their own address resolution and repositories.
- Functionality
 - Server name resolution
 - Server configuration (any parameters needed by device adapters)
 - Queries for device selection (find all devices which are)
 - Basic introspection (device class, property type) - needed?
- Read-only (but solution for “dynamic devices” needed)

Configuration of naming service

- Need to resolve PS, SL, ST ... devices
- Specify file to use (e.g. testing)
- Default name server

Functionality

- Resolve server name based on device name
 - has to deal with PS, SL, ST, ... specifics
- Server configuration
 - In form of SQL query
 - Device adapter specific
- Device selection
 - In form of SQL query
 - Flexible, specific extensions can be defined later (all devices of a transfer line, class, F-E)
- Retrospection
 - In principle clients and servers know their classes, only needed for generic applications
 - SQL query sufficient?

Strategy

- Make requirements
- Start with CORBA to ORACLE server as evaluated by Nikolai

Open questions

- Future of Jan's directory service
- How much of retrospection do we need?
- Is CORBA based solution not too heavy in terms of resources?