Digital Library Federation

Forum on Digital Library Practices
Washington, DC, July 17-18, 1999

Briefing Session 4 — Naming Systems

Stephen L. Abrams
Harvard University Library
stephen_abrams@harvard.edu

Outline

- Technical
  - Requirements
  - Architecture
  - Implementation
  - Namespace
  - Data Model
  - Workflow
- Policy
  - “Who can name things?”
  - “What things are named?”

Requirements

- Requests and resolution via HTTP
- No assumptions regarding HTTP clients
  - No plug-ins
- Many-to-many resolution
  - Abstract resources can have multiple aliases, and multiple instantiations
- Decentralized name assignment/registration
- Compliance with accepted standards
- Initial deployment, Fall 1999

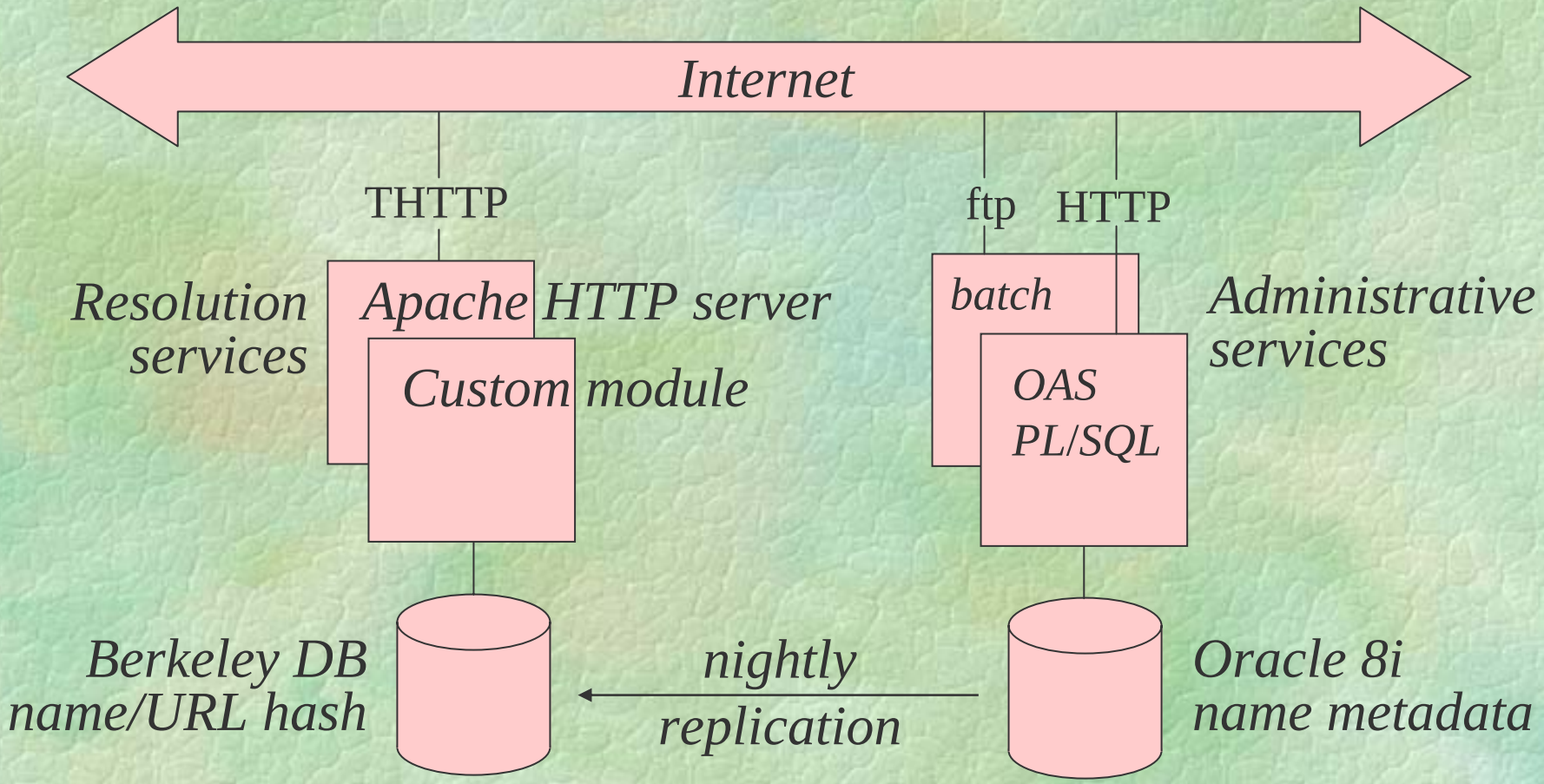
PURLs vs. Handles vs. URNs

- PURLs
  - Location dependence
- Handles
  - Plug-in required; not available for Unix
  - Proxy server: heavy-weight PURL
- URNs
  - HTTP clients not URN-compliant
  - RDS not widely deployed

Name Resolution Service (NRS)

- Initial PURL-like system
  - Consistent with RFCs 2169 and 2483
- URN-compatible namespace (RFC 2141)
- Separation of administrative and resolution services and servers
  - Robust data store for metadata
  - Light-weight HTTP server for resolution
- Future migration to more permanent solution

NRS Architecture



Why Not Use...

- PURL server
  - No one-to-many resolution mapping
- DBM, NDBM, GDBM,...
  - Disk-resident, rather than memory-resident
  - Size restrictions
- LDAP
  - Overhead of external server

Resolution Services

- THTTP protocol (RFC 2169)

  urn:<*nid*>:<*nss*> →
  GET /uri-res/<*service*>?urn:<*nid*>:<*nss*> **HTTP/1.0**

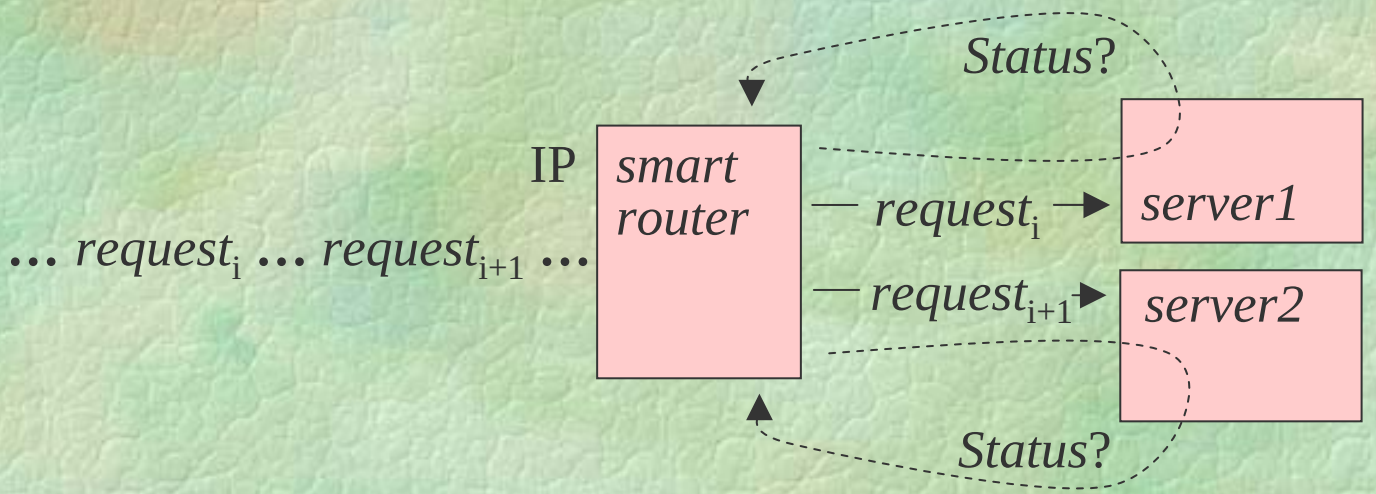
  - Services include:
    - I2L (URI to URL) — server redirect to URL
    - I2Ls (URI to URLs) — return text/uri-list
- PURL-like protocol

  http://<*server*>/<*nid*>:<*nss*> →
  GET /<*nid*>:<*nss*> **HTTP/1.0**

  - I2L service assumed

System Availability

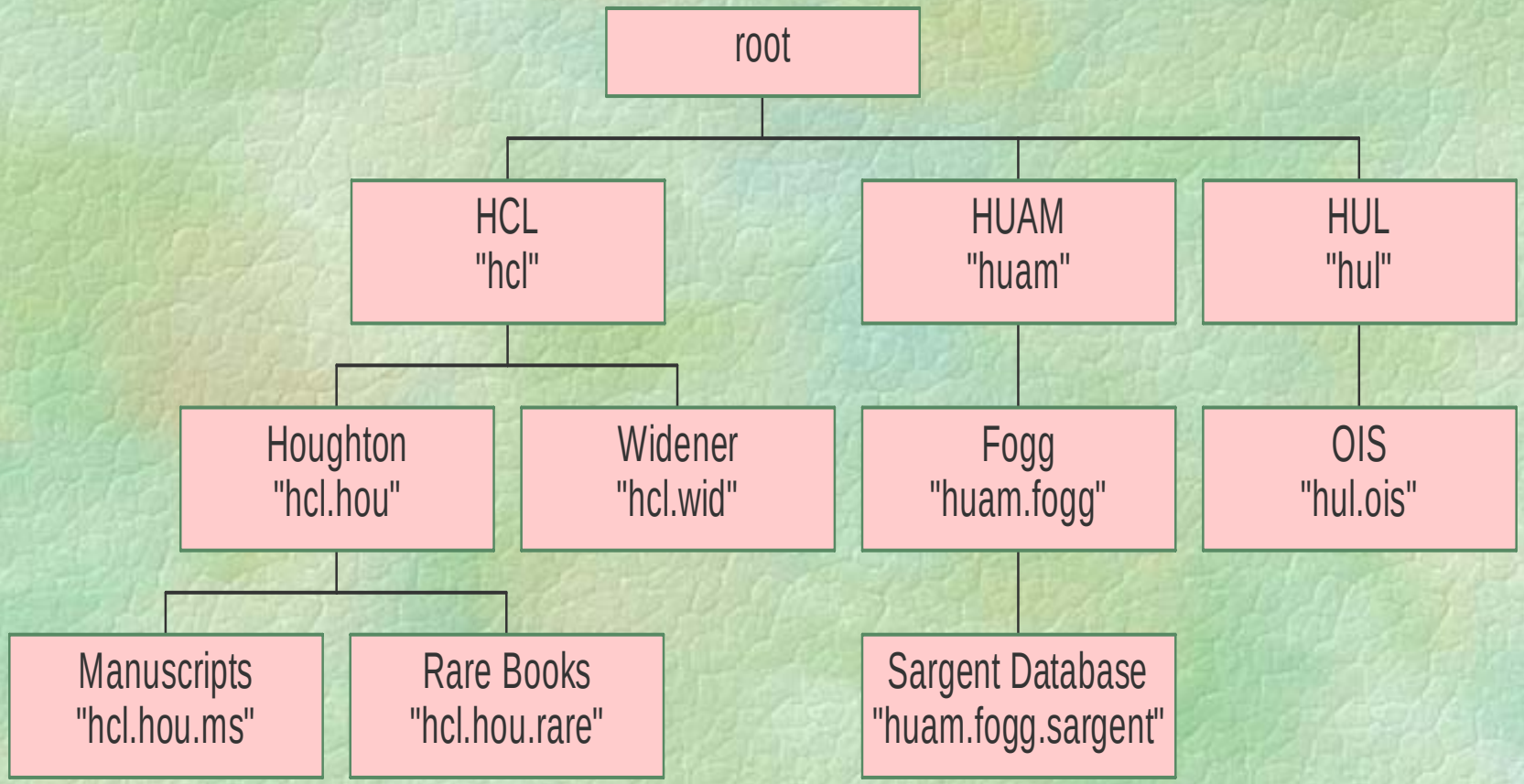
- Two identical resolution servers, each running on isolated machines, sharing a single IP address through “smart router”
- Load balancing and fail-over



Naming Authority

- Administrative unit authorized to assign names in a specific subset of the namespace
- All names are “owned by” a naming authority
- Decentralized delegation of naming authority
  - Analogous to DNS

Hierarchical Authority Path



Users

- Distinguish between “authority path”
  - Syntactic construction that decomposes the namespace into sub-spaces

  and “user”
  - Person granted privileges to request services
- Authentication
  - “Are you who say you are?”
- Authorization
  - “Can you do what you want to do?”

User Privileges

- Privileges to execute naming services are aggregated into named *roles*
- User *profiles* associate users with roles with respect to specific authority paths
  - "Cascade" privilege applies privileges to all delegated authorities
- Every naming authority has one *primary* user ("authority" role), who may designate additional *proxy* users ("proxy" role)

Namespace

- URN-compatible syntax: <*nid*>:<*nss*>

  nrs:<*nss*>

- Incorporate authority path within name:

  nrs:<*authority-path*>:<*resource-name*>

- The <*resource-name*> is unique within context of <*authority-path*>, which is unique within context of <*nid*>

Examples

nrs:div.findaids:1stUnitarianSoc.register

nrs:hcl.fal.archives:David_Smith.correspondence

nrs:hcl.hou.theatre:Oh,Kay!

nrs:huam.fogg.psd:di_Paolo.St_Catherine_of_Siena

nrs:huam.fogg.acc:1921.13

nrs:huam.fogg.straus:di_Paolo.St_Catherine_of_Siena

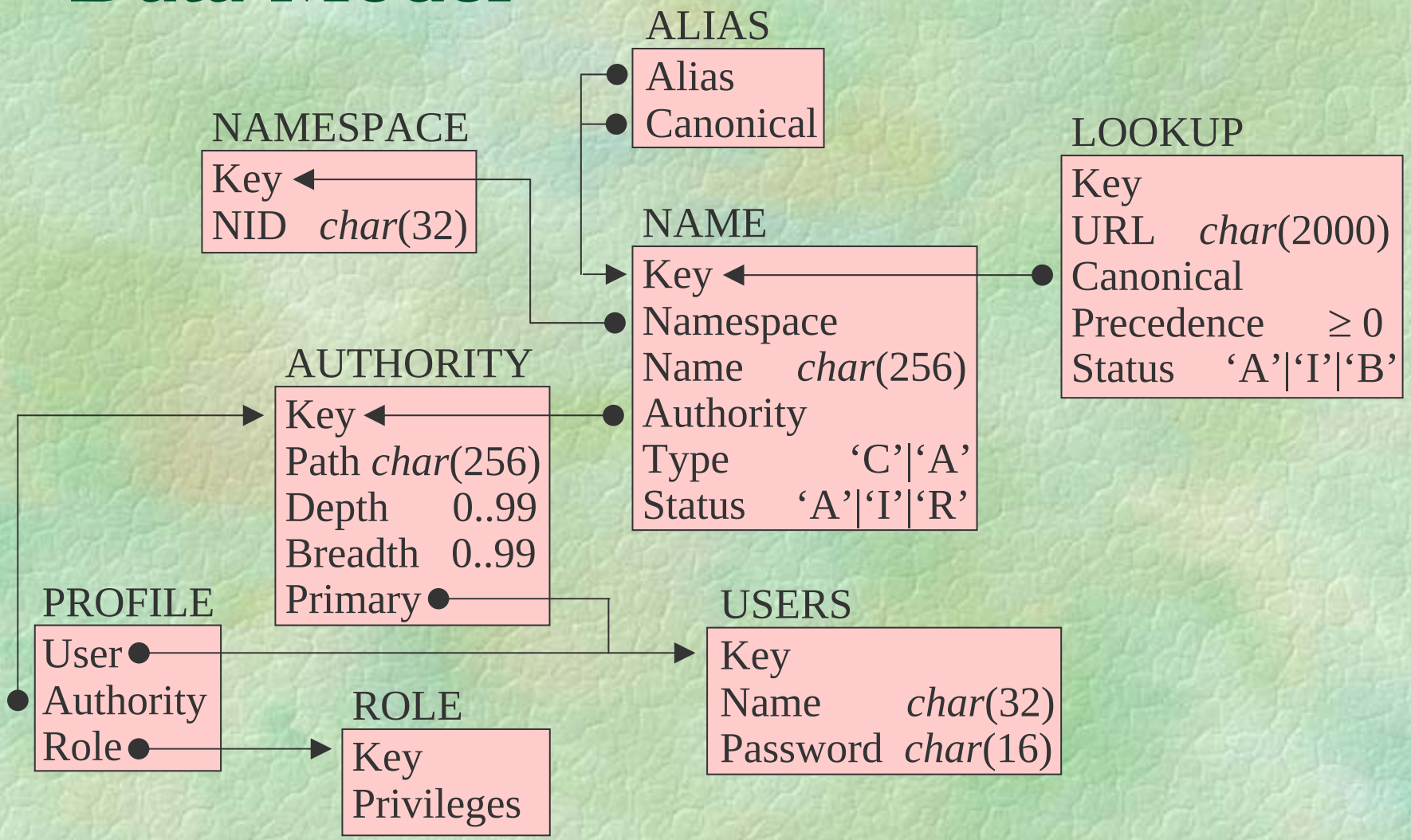
nrs:hul.archives:rad.pres-report-1899.p15

Migration Path

- Name is independent of resolution mechanism
- Algorithmic transformation between resolution syntax

http://<*nrs*>/nrs:<*authority-path*>:<*resource-name*>
→ urn:nrs:<*authority-path*>:<*resource-name*>
→ hdl:nrs/<*authority-path*>:<*resource-name*>
→ urn:hdl:nrs/<*authority-path*>:<*resource-name*>

Data Model


ALIAS
Alias
Canonical
NAMESPACE
Key
NID char(32)
LOOKUP
Key
URL char(2000)
Canonical
Precedence ≥ 0
Status 'A'|'I'|'B'
NAME
Key
Namespace
Name char(256)
Authority
Type 'C'|'A'
Status 'A'|'I'|'R'
AUTHORITY
Key
Path char(256)
Depth 0..99
Breadth 0..99
Primary
PROFILE
User
Authority
Role
USERS
Key
Name char(32)
Password char(16)
ROLE
Key
Privileges

Administrative Workflow

- “Private” services
  - Registration of namespaces
  - Registration of roles
- “Public” services
  - Registration of naming authorities
  - Registration of users and assignment of profiles
  - Registration of names
  - Reporting

Naming Workflow

- User has no name or URL
  - Request name, based on template
  - Add URL when known
- Name, but no URL
  - Lookup name to check uniqueness
  - Reserve name; add URL when known
- URL, but no name
  - Request name with URL
- Name and URL
  - Lookup name; add name with URL

Policy Questions

- Criteria for top-level naming authorities
- Unlimited authority delegation?
- Business model
  - Balance ubiquitous acceptance with cost recovery
- What things should be named?
  - Are we naming “files” or “objects”?
  - How to handle versions and formats?

Named “File” vs. “Object”

- The named thing can be an abstract resource
  - Abstract resources cannot be delivered; only specific, tangible representations
- At some point between request and delivery a specific instantiation of the content must be fully identified
  - Client configuration or heuristic
  - Selection by end-user

Multiple Versions, Formats

- Assuming a dumb client, each specific instantiation is given one name, resolvable to single URL, requested via I2L service
- Assuming a smart client, abstract resource is given one name, resolvable to multiple URLs, requested via I2Ls service
  - Client must incorporate mechanism to select appropriate URL

Status

- Operational
  - Resolution server (I2L and I2Ls)
  - Oracle tables defined
- In progress
  - Data replication
  - Interactive/batch administrative interfaces
- Future
  - Wait for “standards” to emerge
  - URN RDS: how to “publish” NRS service?
  - Java for adding intelligence to clients?
  - CORBA/IIOP for interoperability?