URN Resolution Requirements and Plans

Lewis Girod

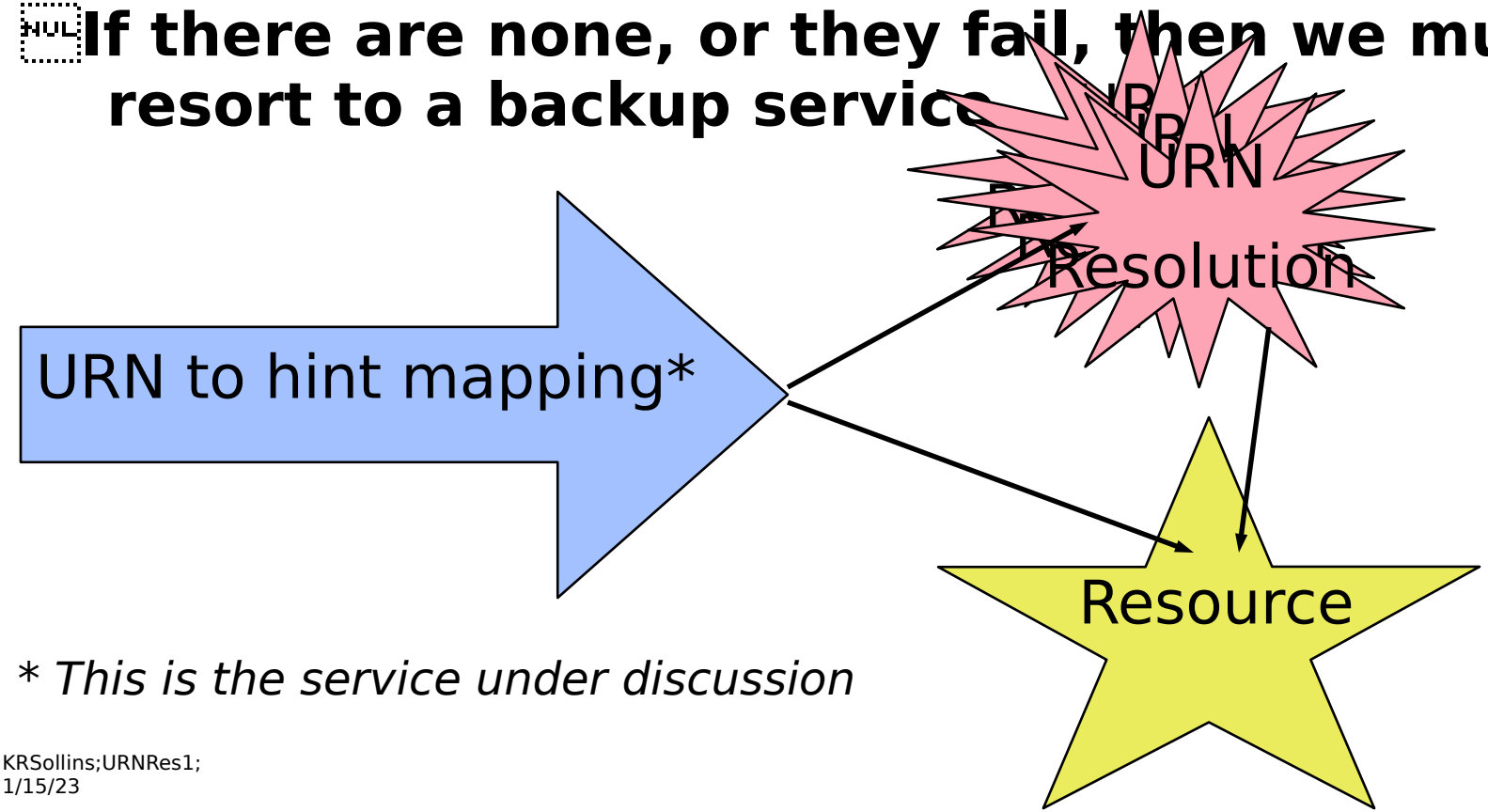
Karen Sollins

MIT Laboratory for Computer Science

June 27, 1996

URN Resolution

- **Hints help find resources, mutable and non-authoritative**
- **Generally, hints will be cached in lots of places**
- **If there are none, or they fail, then we must resort to a backup service**



** This is the service under discussion*

From NAPTR framework paper

- **Persistence/longevity**
  - flexibility in choice of resolution services
  - modularity of layers of information management
- **Name scheme**
  - resolution should be independent of NS
  - verifiable checklist of requirements (URN Requirements list?)
- **Authority**
  - modularize
  - distribute to keep information maintenance close to authority for information

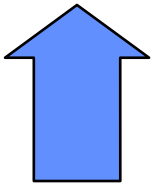
Caveats

- **URN namespaces must conform to URN Requirements doc.**
- **Syntax: URN:<NID>:<NSS>**
- **We must be able to convince the community that the proposal is reasonable - we are already getting negative press.**

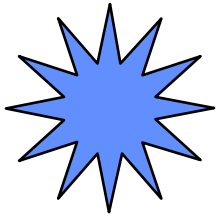
Our requirements

- **Usability**
  - publishers
  - clients
  - information mgrs.
- **Security & Privacy**
  - access control on updating hint information
  - server authenticity
  - server availability: resistence to denial of service
  - some degree of privacy
- **Evolution**
  - new NIDs (URN schemes)
  - new resolution services
  - authentication and other security mechanisms
  - new or multiple top level models for URN to hint mapping

Quick Look at Fixing NAPTR



- **Evolution:**
  - Require additional client gateway protocol



- **Security:**
  - Build authentication into the new records

- **Usability:**
  - Rewrite rules should be generated by special management software that implements a set policy

Evolution

- **The NAPTR proposal can be implemented quickly because it uses the DNS.**
  - This may mean that it is the only resolution method when some clients are implemented
- **We should explicitly require clients to support an additional gateway protocol**
  - clients running a DNS-based protocol cannot easily escape to a new one
  - therefore, require a simple protocol that sends the whole URN to a gateway address and waits for a list of hints to come back (i.e. SRV, A, or NAPTR records, or whatever clients understand)

Security

- **DNSSEC will make the DNS more secure**
  - It gives each zone a private key; each record in that zone has an associated SIG record
  - Public keys for zones can be acquired from the parent and child zones
  - mods to the DNS DB must be authorized
- **NAPTR, SRV should still have extra auth info**
  - DNSSEC cannot authenticate resolution information except at granularity of zones
  - there are other security issues that are part of the policies set by individual NAs
    - ex. interference between publishers' rules vs. restrictions on types of rules allowed

Usability and Evolution

- **Large systems of rewrite rules are hairy**
  - Difficult to understand, verify, maintain
  - Hard to translate for use by other systems
- **Many problems with rewrite rules can be solved with management software**
  - need software that takes namespace map as input, produces system of rewrite rules
  - forces security policies to be clearly defined and implemented in the software
  - easy to add a simple & secure publisher's interface to rewrite rule systems
  - ``source files'' can be moved over to future systems, updates can be mirrored

Quick Look at Fixing NAPTR

- **Require additional client gateway protocol**
  - clients running the current DNS-based protocol cannot easily escape to a new one
- **Build authentication into the new records**
  - DNSSEC can auth. at the granularity of a zone but cannot separately auth. individual records
- **Confusing rewrite rules -> mgmt software**
  - need software that takes namespace map as input, produces system of rewrite rules
  - software can implement security policies, prevent publishers' rules from interfering
  - can easily include simple & secure publisher interface to rewrite rule systems