

# Coping with Un-Cool URIs in the Web of Linked Data

John Arwe (author)  
Steve Speicher (kind presenter)  
IBM Corporation

# Problem Statement

- \* Linked Data's assumptions are based on HTTP URLs being immutable.
- \* Production URLs sometimes do change. For valid reasons.
- \* In the Web of Linked Data, broken links violate the principle of providing useful information when a URI is "looked up".
- \* We are dealing with a distributed ecosystem of components: humans, organizations, code, legal systems
  - \* Those components have different abilities to respond to URL changes, and different incentives to invest in responding.

# Classes of URL-Handling Behavior: Humans/Organizations

- \* Cool-URL-capable – “there is no spoon”
- \* Durable-URL-capable – URL change only a last resort
  - \* Various levels of virtualization allow a durable URL to work through most location changes, e.g. DNS aliases, (D)VIPA
- \* Redirect-capable – able + willing to deploy 301 server
  - \* Some organizations cannot/will not do this today
- \* URL prefix-mapping-capable
  - \* Able and willing to provide URL prefix mappings to client and/or server components

# Classes of URL-Handling Behavior: HTTP Clients

- \* Redirect-aware – processes 301 Moved Permanently
- \* Redirect-caching/persisting – updates stored URL too
- \* Consultative
  - \* Use of implementation-specific “Link Oracle” to find likely replacement for a URL whose requests result in 404 Not Found
  - \* Current set of mappings is an external input to the Oracle
  - \* Assumption: string replacement using URL prefix only
    - \* Still a weakening of WebArch’s URL Opacity principle
- \* Link-mappable – actively translate URLs
  - \* Essentially moves the Link Oracle into the client

# Classes of URL-Handling Behavior: HTTP Servers

- \* Cool – configurable stable URL prefix
- \* Relocatable – stable URL prefix can be changed
  - \* URL persistence decisions bear on relocation cost
  - \* Representation and parameter handling can be tricky
- \* Cloneable – snapshot, quickly “uniquify” data, deploy
  - \* Cloud-deployment friendly
- \* Trackable – advertises former stable URL prefix(es)

# Big Issue with URL Prefix Mapping

- \* How to handle “backfill”, i.e. deploying a new server at an old server’s URL prefix
  - \* Outlawed forever: redirects (and/or URL mappings) can be applied on an on-going basis
    - \* Client-side persisted links can be fixed gradually, based on clients’ need/motivation to avoid the extra latency
    - \* “forever” tends not to work well in practice though
  - \* Allowed at all: we introduce semantic ambiguity
    - \* All existing links must be updated before the replacement is deployed
    - \* A client with a stale URL, especially as part of a query, will be asking a different question than the server answers

# Classes of URL-Handling Behavior: Standards Organizations

- \* Silent
- \* Awareness-building
  - \* E.g. summarizing issues, ala RFC 2616 Security chapter
- \* Actively engaging
  - \* Assessing need for standards
    - \* URL prefix mappings?
  - \* Asserting normative behavior
    - \* E.g. “origin servers SHOULD be movable”