(Persistent) Identifiers for Concepts / Terms / Relationships

Andy Powell, UKOLN, University of Bath

a.powell@ukoln.ac.uk

NKOS Special Session – DC-2005, Madrid

UKOLN is supported by: JISC MLA MUSEUMS LIBRARIES ARCHIVES

Persistent identifiers for metadata terms in a Web environment

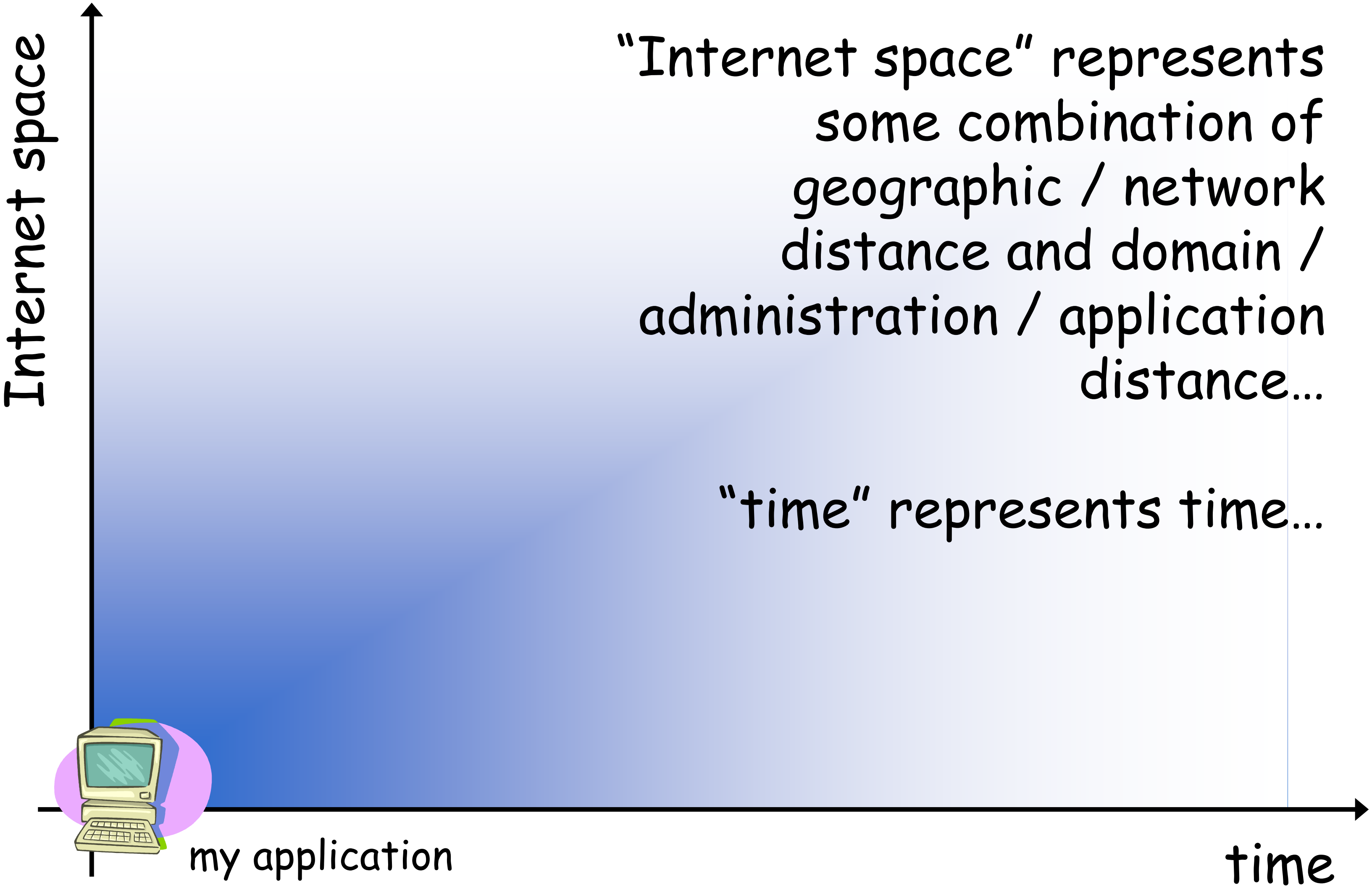
Contents

- functional requirements
- generic stuff about all identifiers in the context of the W3C “Web architecture”
- specific stuff about identifiers for metadata terms

Functional requirements

- declare and use metadata terms
- identify terms uniquely and globally
  - enable other people to use our terms
- attach definitions to our terms
  - indicate relationships between our terms and other terms
  - allow other people and applications to get to (and understand) our term definitions and relationships
- do all this persistently – so that stuff works into the future

Space/time continuum



Internet space

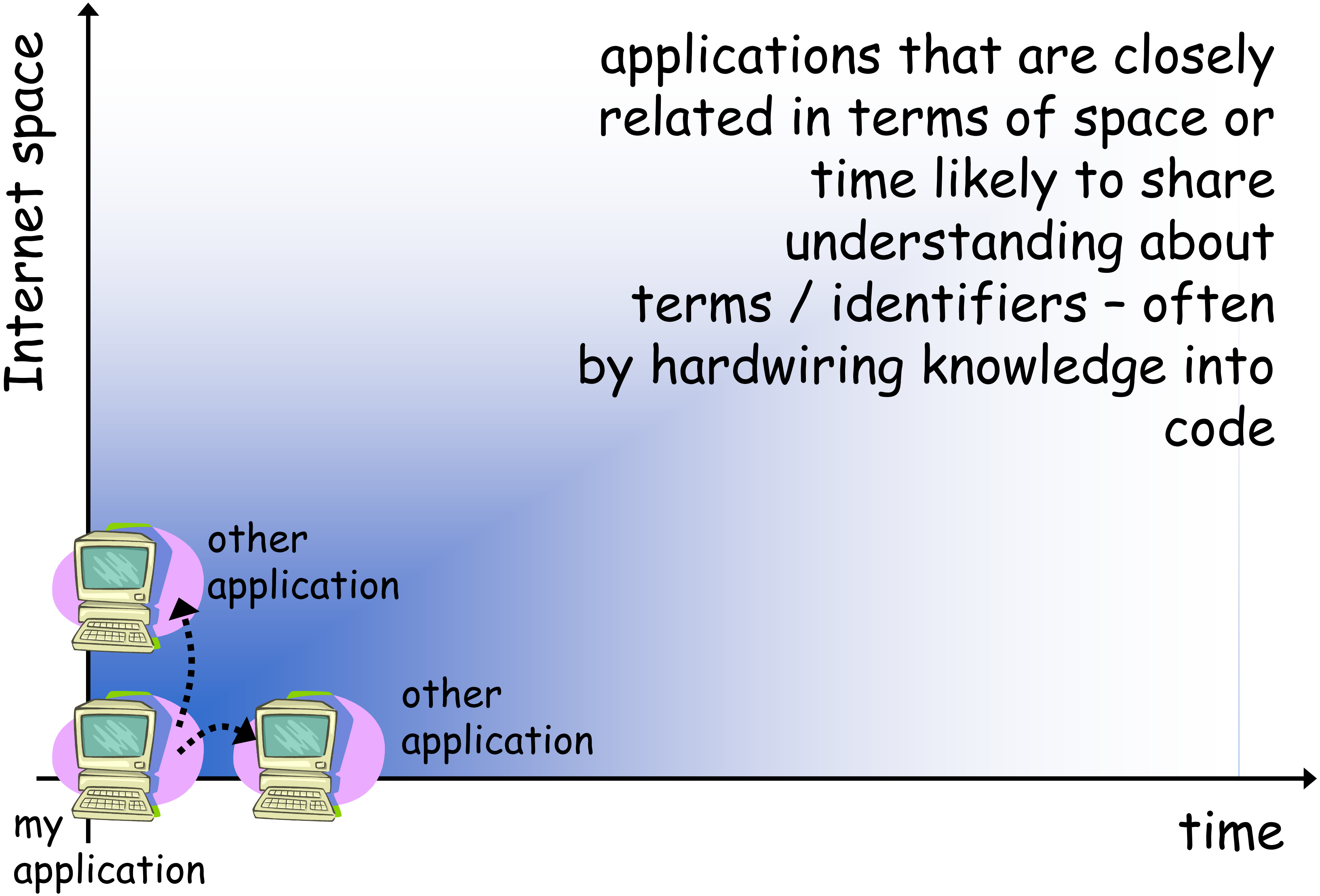
“Internet space” represents some combination of geographic / network distance and domain / administration / application distance…

“time” represents time…

my application

time

Space/time continuum



Internet space

applications that are closely related in terms of space or time likely to share understanding about terms / identifiers – often by hardwiring knowledge into code

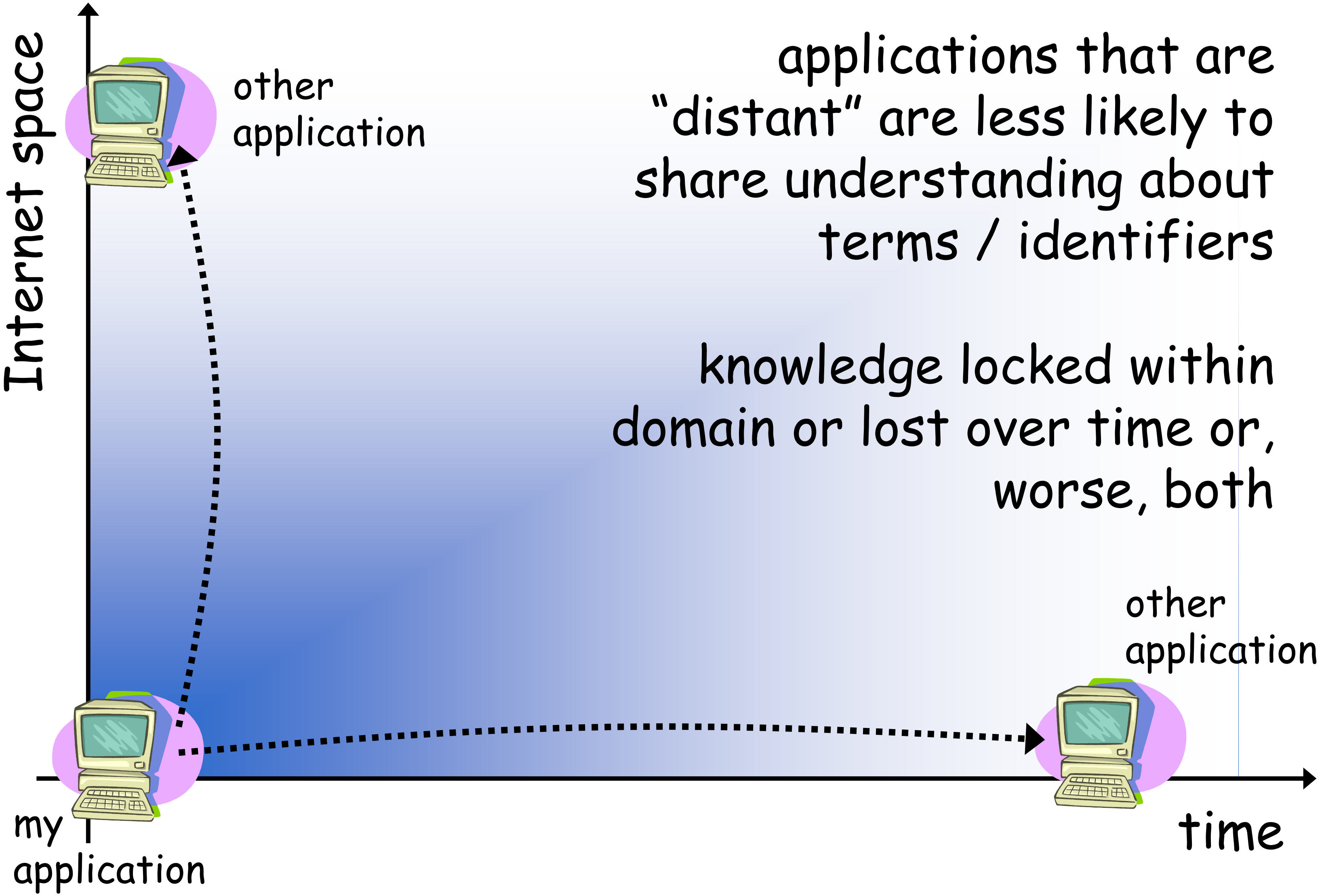
other application

other application

my application

time

Space/time continuum



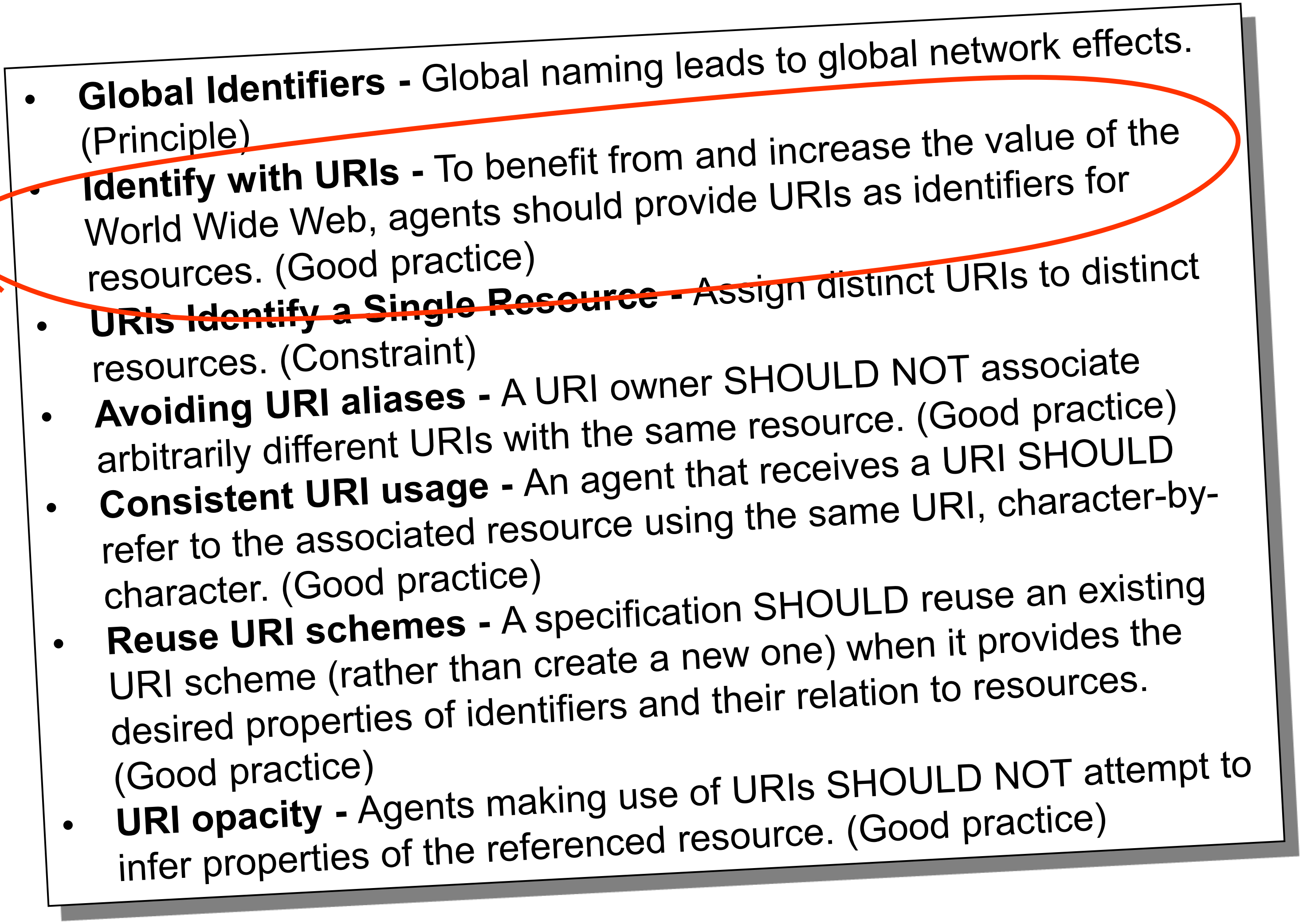
applications that are "distant" are less likely to share understanding about terms / identifiers

knowledge locked within domain or lost over time or, worse, both

Pushing the boundaries

- how do we push the boundaries of term / identifier understanding further out across the space/time continuum?
  - standards, standards, standards
  - go with the crowd
  - use identifiers that already work and are widely deployed

W3C Web Architecture



- **Global Identifiers -** Global naming leads to global network effects. (Principle)
- **Identify with URIs -** To benefit from and increase the value of the World Wide Web, agents should provide URIs as identifiers for resources. (Good practice)
- **URIs Identify a Single Resource -** Assign distinct URIs to distinct resources. (Constraint)
- **Avoiding URI aliases -** A URI owner SHOULD NOT associate arbitrarily different URIs with the same resource. (Good practice)
- **Consistent URI usage -** An agent that receives a URI SHOULD refer to the associated resource using the same URI, character-by-character. (Good practice)
- **Reuse URI schemes -** A specification SHOULD reuse an existing URI scheme (rather than create a new one) when it provides the desired properties of identifiers and their relation to resources. (Good practice)
- **URI opacity -** Agents making use of URIs SHOULD NOT attempt to infer properties of the referenced resource. (Good practice)

http://www.w3.org/TR/webarch/

URIs and XML

- in order for terms / identifiers to work across the space/time continuum we need
  - global and unambiguous identifiers
  - global and unambiguous ways of exchanging identifiers between software applications
- the Uniform Resource Identifier (URI) is the only option for the former
- XML is the "best" option for the latter
  - and in particular the XML Schema AnyURI datatype

"global" means "very widely deployed technology" – e.g. even in my mum's house!

1st conclusion

- identify all metadata terms with URIs

URI scheme registration

- registration of URI schemes is important
- registration helps to ensure uniqueness
- without registration the same scheme can be used in ignorance by someone, somewhere else in the space/time continuum
- registration doesn't guarantee that every URI with a scheme will be unique – but it helps!
- without registration there are no guarantees of uniqueness or persistence

2nd conclusion

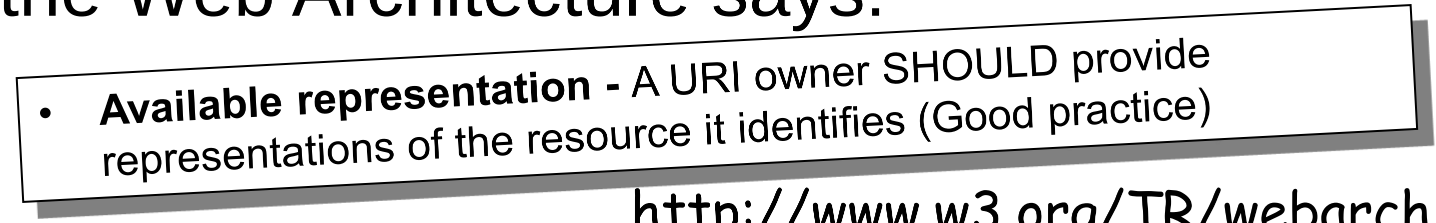
- identify all metadata terms with URIs taken from registered URI schemes

Semantic Web

- the Semantic Web relies on URIs to identify resources
- resources == stuff (digital/physical/conceptual things)
- the semantic Web is built on a global, shared body of metadata (RDF)
- terms in the metadata language are identified using URIs
- those URIs must be “resolvable”… in order that “reasoning” can be performed
  - sharing knowledge about terms

Note: dereferencing URIs

- the Web Architecture talks about "dereferencing" URIs rather than "resolving" them
  - in many cases "dereferencing" a URI results in obtaining a "representation" of the resource
  - several representations may be available
- the Web Architecture says:



- **Available representation -** A URI owner SHOULD provide representations of the resource it identifies (Good practice)

http://www.w3.org/TR/webarch/

- only 'http' URIs offer simple, widely deployed dereferencing mechanism

Quick quiz…

- what kind of identifier is this?
  - info:lccn/n78890351

is an 'info' URI

it identifies a Library of Congress metadata record (an authority file) but I don't know which

Quick quiz…

- what kind of identifier is this?
  - info:lccn/n78890351
  - 10.1000/182

is a DOI

it is also a Handle

it identifies the "DOI Handbook"

Quick quiz…

- what kind of identifier is this?
  - info:lccn/n78890351
  - 10.1000/182
  - http://purl.org/dc/terms/audience

is an 'http' URI
a.k.a. a URL
it is also a PURL

it identifies a DCMI metadata term – i.e. a conceptual resource

Quick quiz…

- what kind of identifier is this?
  - info:lccn/n78890351
  - 10.1000/182
  - http://purl.org/dc/terms/audience
- only one of these can be understood and dereferenced by every single bit of currently deployed Internet software…

Hint: it's the last one!

Question: why would we want to use anything else?

But…

But, ‘http’ URIs are just locators aren’t they?

- ‘http’ URIs are identifiers, just like any other

But, ‘http’ URIs can only be used for Web resources, accessed over HTTP, can’t they?

- ‘http’ URIs can identify any resource – digital, physical or conceptual

But, ‘http’ URIs break every 30 days or something, don’t they?

- ‘http’ URIs don’t have to break, they just need to be assigned/managed carefully

3rd conclusion

- identify all metadata terms with ‘http’ URIs (because that provides a widely deployed mechanism for obtaining information about the term)

Case study - DOI

http://dx.doi.org/10.1000/182

- the DOI “10.1000/182” can be encoded as a URI in several ways:
  - http://dx.doi.org/10.1000/182
  - doi:10.1000/182
  - urn:doi:10.1000/182
- however…
  - DOI-aware applica e of these encodings the DOI itself is just a string)
  - nothing in the URI specification indicates that these URIs are equivalent
  - note that the 2nd and 3rd forms are not registered

Question: which of these forms is most persistent and why?

Case study – ‘info’ URI

http://info-uri.info/registry/

- consider the following ‘info’ URI:
  - info:lccn/n78890351
- ‘info’ URIs are explicitly defined to be non-dereferencable
- therefore, there is no documented way of finding out what this URI identifies
- there is no documented way of getting a representation of the resource it identifies
- and there is no documented way of finding out any more about it

Question: how is this useful?

But, what happens when…

- **…the Internet and/or HTTP disappears?**
- who cares!
- we’ll deal with it
- we’ll be with the crowd
- there’ll be a global transition
- everyone will need to deal with it
- every software component on the whole Internet will need fixing
- the people left behind will be the people who invented their own solutions

Technical practicalities

- terms are ‘conceptual’ resources
- therefore, the “Web architecture” suggests that they should be dereferenced via an HTTP 303 redirect
  - HTTP 303 redirect should result in a description of the term being returned
  - use HTTP ‘content negotiation’ to select between a human-readable description (text/html) and a machine-readable description (application/xml+rdf)
- SKOS Core looks like good candidate for the RDF description

How do I choose a URI?

- Guidelines for assigning identifiers to metadata terms
  http://www.ukoln.ac.uk/metadata/dcmi/term-identifier-guidelines/
- …makes some recommendations for assigning ‘http’ URIs
  - using project and/or service URIs
  - using the xmlns.com domain
  - using PURLs
- of these, PURLs seem to be the most appropriate and persistent

Conclusion

- assign ‘http’ URIs to your terms
- use PURLs as your ‘http’ URI
- dereference them via an HTTP 303 redirect to both human-readable and machine-readable information about the term
- use RDF/RDFS/OWL/SKOS Core to encode the machine-readable information ??

Questions