Designing Interoperable Metadata on Linked Data Principles
“Application Profile” – something that takes an implementer…

Based on: http://dublincore.org/documents/singapore-framework/
…from Functional Requirements and a Domain Model

What a metadata application must do

- “Support navigation between ‘versions’.”
- “Enable searching on the ‘owner’ of a collection.”

What “things” are described in a metadata application

- Resources (generically)
- Authors and Books
- Scholarly Works, Expressions, Manifestations, Items, Agents
  - as in Functional Requirements for Bibliographic Records (FRBR)
…to a Description Set Profile and Data Format

How things of the Domain Model are described
- Use of Properties and Classes.
- Their cardinality
- Whether mandatory or optional
- Lists of allowable values…

Structure of the actual data format used by an application
- Components of the data structure.
- How the components map to Linked Data.
…on the basis of Community Standards, which in turn are grounded in…

- **Community Domain Models**
  - Using well-known entities, such as FRBR classes

- **Metadata Vocabularies**
  - Using well-known, such as Dublin Core and FOAF
...in Foundation Standards for Linked Data (RDF).
DCMI guidelines, under review, offer one approach\(^1\) to designing Linked-Data-compatible metadata records.

\(^1\) There is more than one way to do it. To be discussed at DC-2010 in Pittsburgh, October 2010, http://dc-2010.org.
“Closed” ("normal") IT: Integrate across silos by mapping ad-hoc data structures

Schema A \hspace{2cm} \text{same as} \hspace{2cm} \text{mapped to} \hspace{2cm} Schema B \hspace{2cm} \text{mapped to} \hspace{2cm} Schema C
New Linked Data approach: Diverse applications create good triples

Application A

Application B

Application C
Good triples can be merged coherently
Applications come and go…

Profile A

Profile B

Profile C
The data remains
Good triples, based on known vocabularies such as Dublin Core, make data “self-descriptive”