Interoperability of Heterogeneous Information Systems

ITU-T Recommendation X.1255 Discovery of Identity Management Information

Robert E Kahn

Corporation for National Research Initiatives (CNRI)

NITRD Presentation
December 3, 2013

Framework for Discovery

- X.discovery is an approved ITU Recommendation about a framework for discovery of “Identity Management Information”
- However, problem is the same as discovery of any kind of information
- So, the framework is applicable to any discovery application and also to any access requirement
- It is thus a generic framework to discover and access any kind of information in digital form
- In systems that adhere to the overall framework and participate in the discovery/access process

What Problem is being solved?

- Multiple Information Systems
  - Of different kinds
- Information desired is not always available from a given system
- But can be made available from another system – if it can be discovered and accessed
- How can this best be accomplished?

Comparison with the Original Internet Challenge

- Multiple networks and computers
  - Of different kinds
- Connectivity may require traversing multiple networks to reach the destination
- How to discover and access the desired destination network and computer?
- Basic solution – protocols and procedures
  - Such as TCP/IP and gateways (now routers)

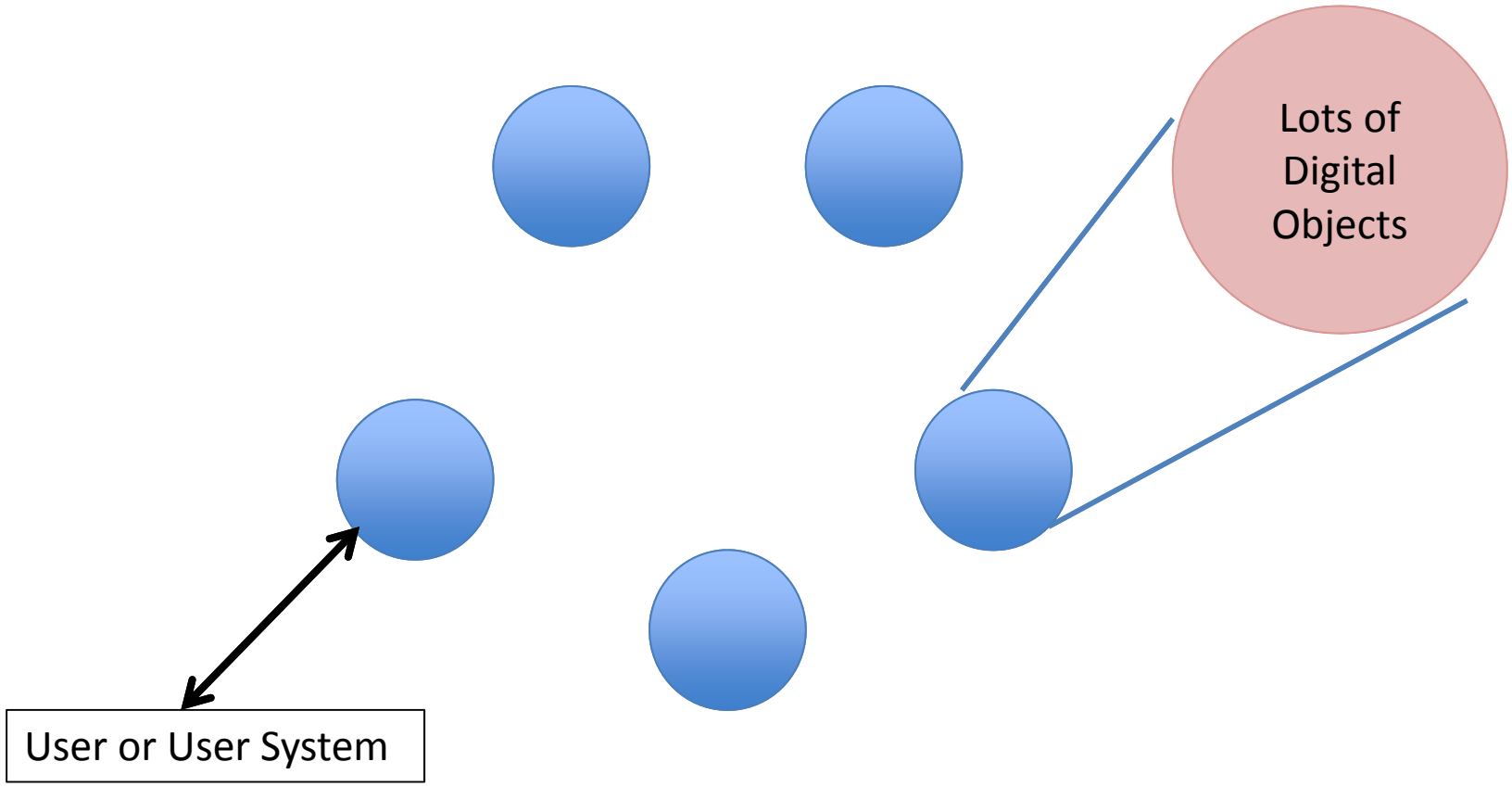
Bindings to Technology vs. Information

- Arpanet – 16 bit addresses → wires
- Internet – 32 bit IP addresses → machines
- Web - URLs → <IP Address/filename>
- Digital Object Architecture → state information about the desired information such as access means, multiple locations, authentication, public keys, terms and conditions for use, etc.

Some Terminology

- Digital Entities (DEs) & Digital Objects (DOs)
  - An Entity is something that has a separate existence and are capable of being uniquely identified.
  - Digital Entity is an entity represented as or converted to a machine independent data structure consisting of one or more elements in digital form that can be parsed
  - A DE is a more abstract notion of a DO – both are structured data with a unique persistent identifies.
- Unique Persistent Identifiers
  - resolvable in the Internet to “state information” or to produce relevant metadata
  - Used to discover and access DOs
- Generic types
  - Each DO consists of multiple elements of <type,value>
- Other components store digital objects, metadata
  - Namely DO Repositories and DO Registries

Multiple Information Systems

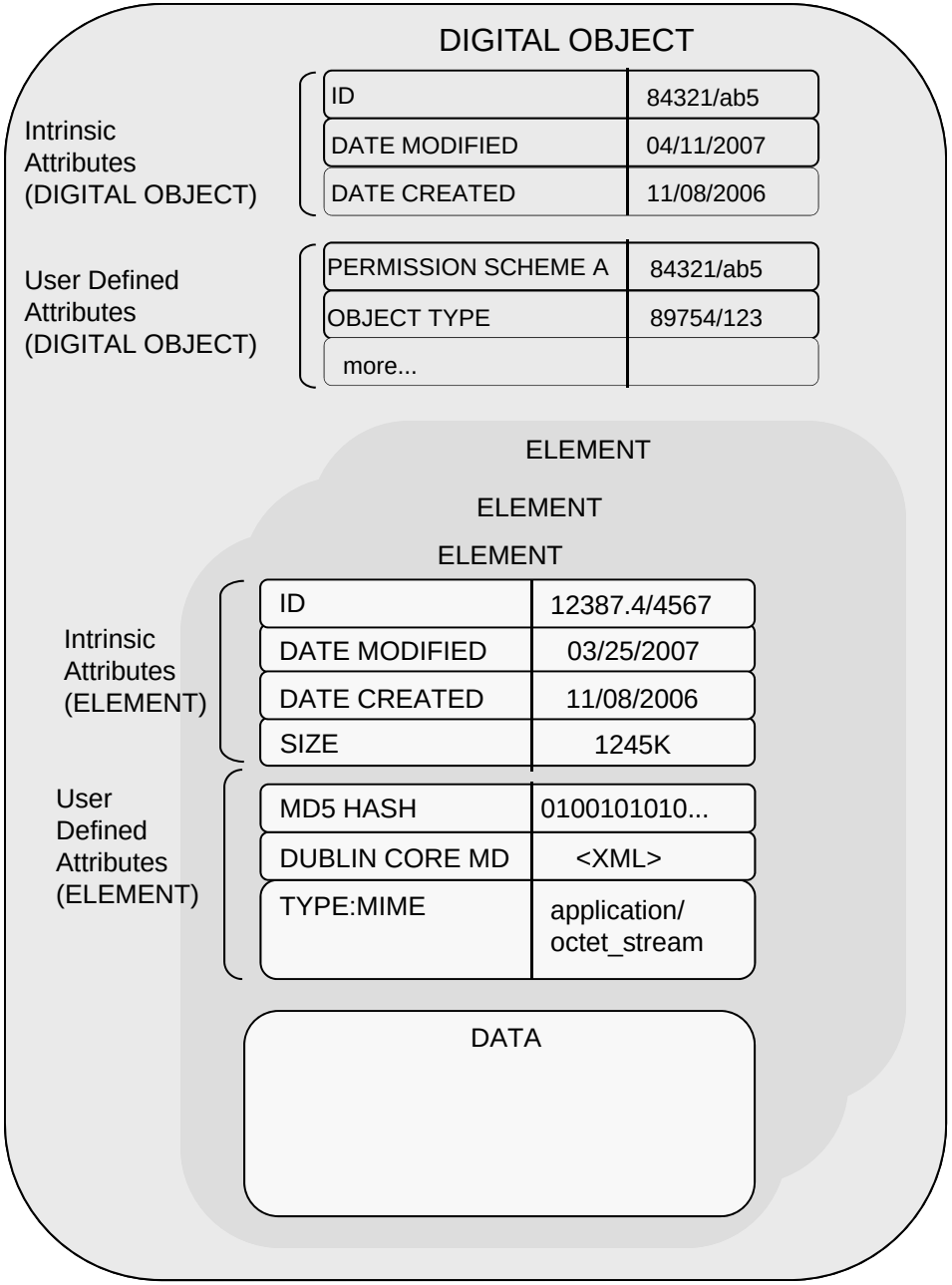


Framework for Discovery

- X.discovery is an approved ITU Recommendation about a framework for discovery of “Identity Management Information”
- However, problem is the same as discovery of any kind of information
- So, the framework is applicable to any discovery application and also to any access requirement
- It is thus a generic framework to discover and access any kind of information in digital form
- In systems that adhere to the overall framework and participate in the discovery/access process

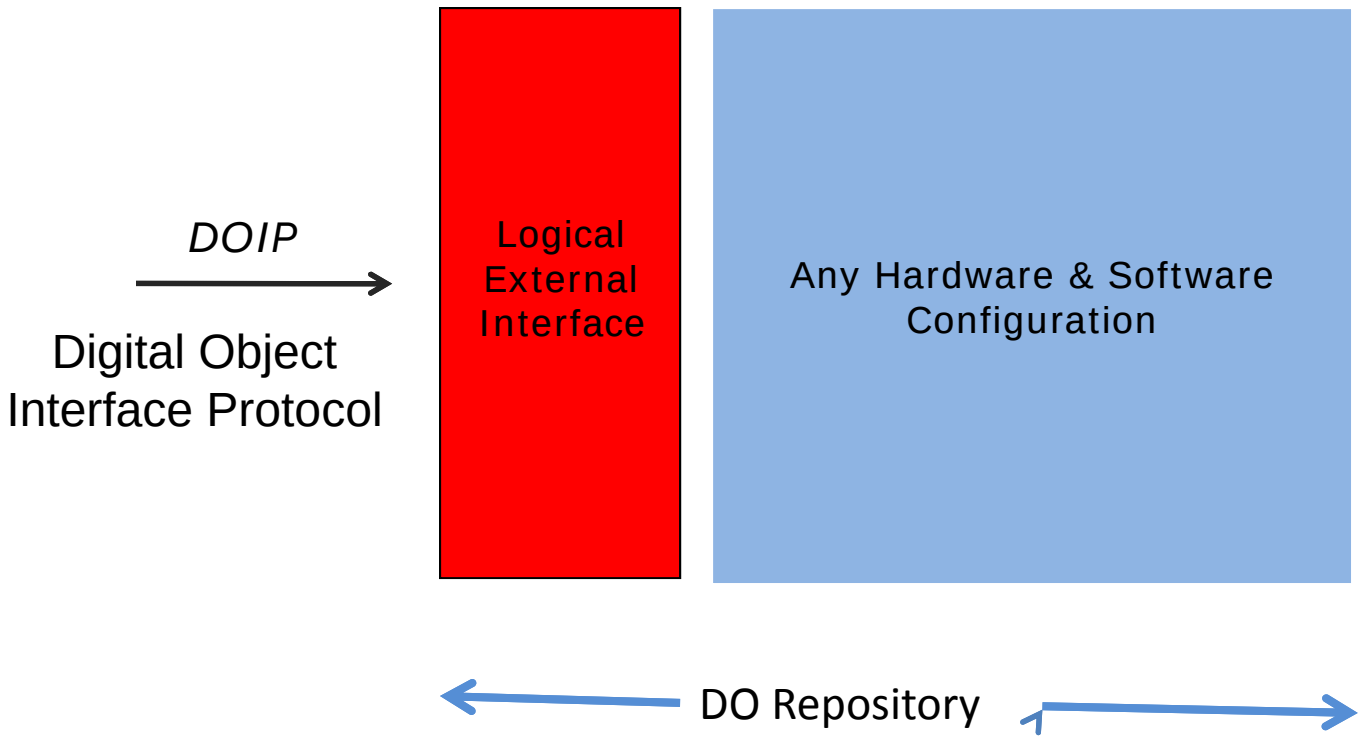
Digital Object Architecture

- **Digital Object Data Model & Protocol**
  - Logical interface to heterogeneous information management and storage systems
  - Built-in strong authentication and encryption
- **Digital Object Repository**
  - Implements the digital object data model and protocol
  - Portal into multiple info and storage systems
  - Security is at the object level & objects can be securely shared
  - Current version successfully used by industry and government
- **Handle System**
  - Highly scalable identifier resolution system for digital objects
  - Provides referential integrity as objects move and environments change
  - Proven and in wide use
- **Digital Object Registry**
  - Manages metadata records about resources
  - Assigns handles to metadata records and resources
  - Normalizes organizational boundaries through commonly agreed API's and metadata models

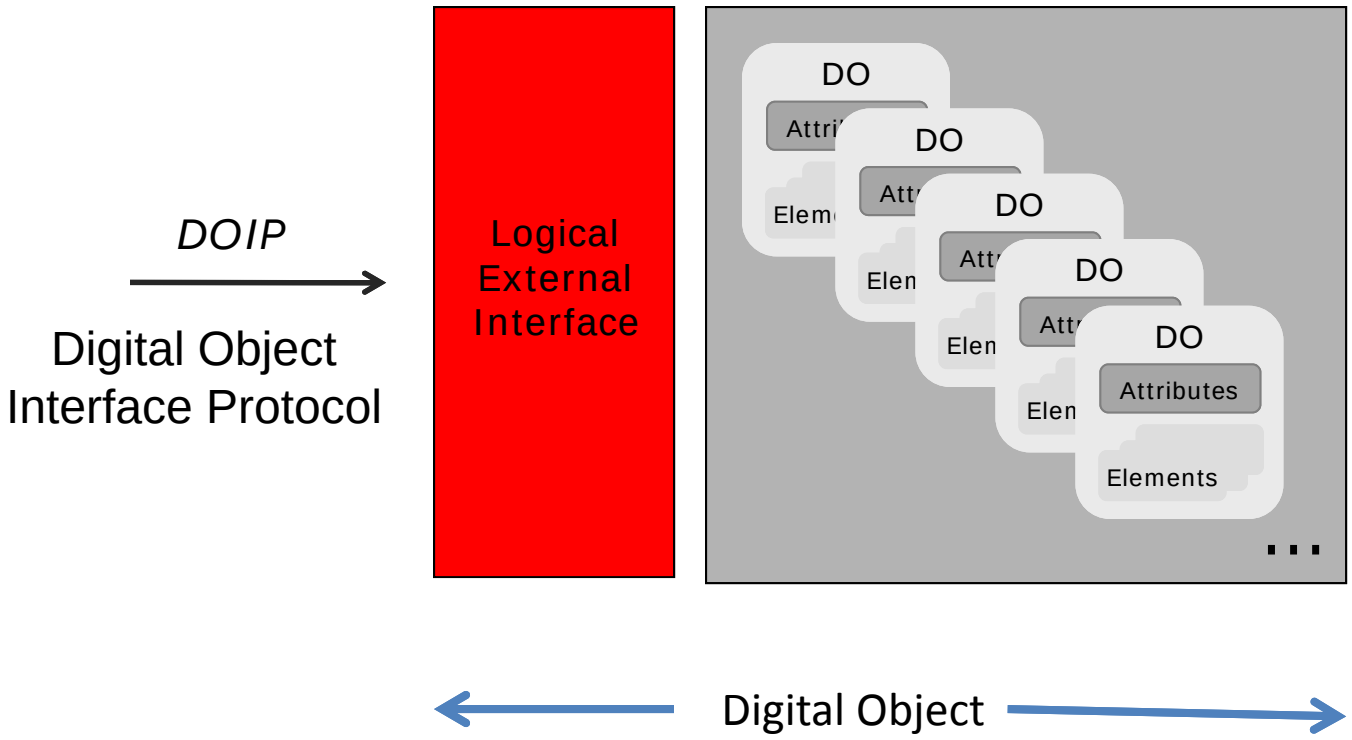


- Each Object contains structured data and extensible metadata
- Metadata includes types, dates, permissions, and other relevant attributes

DO Repository Notion



DO Interface Protocol



Operations on Digital Objects

- An operation on a digital object consists of the following elements:
  - User ID: The identifier of the entity requesting invocation of the operation
  - ObjectID: The identifier of the digital object to be operated upon
  - OperationID: The identifier that specifies the operation to be performed
  - Input: A stream of bytes that contains the input for the operation, including any parameters, or content
  - Output: A stream of bytes that contains the output of the operation, including any content or messages
- All identifiers are handles or more generically digital object identifiers
- Examples of operations:
  - Create object
  - Update data element
  - Get data element
  - Delete data element
  - Delete object
  - And so on

Digital Object Interface Protocol

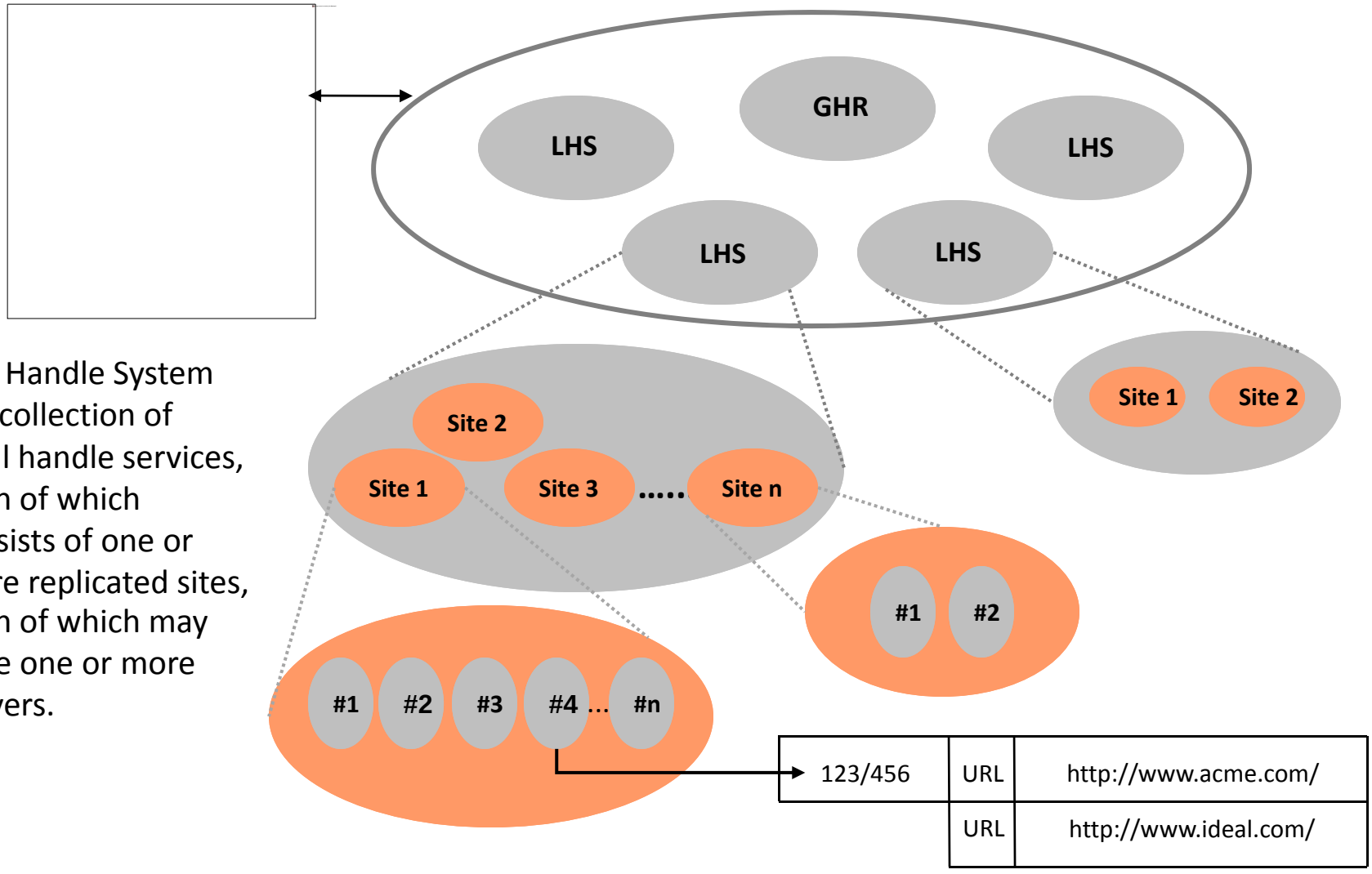
- Establish Connection with the desired resource
  - Currently using TCP/IP but other protocols are possible
- (Optionally) Validate the target resource
- If valid, present the request string
  - **<input><operation ID><object ID> <parameters><output>**
- (Optionally) Validate the User
- Fulfill the request or terminate the request
- If last active user on the connection, disconnect or
- Repeat the above without reconnecting

Handle String

- <prefix> / <suffix>
- Examples
  - 11.1002/1000/11951-en
  - 4263537/5030
- Character Set: Unicode 2.0
- Encoding: UTF-8
- Prefixes
  - Currently allocating only numeric TLPs
  - Alphanumeric allowed everywhere else

Handle Resolution

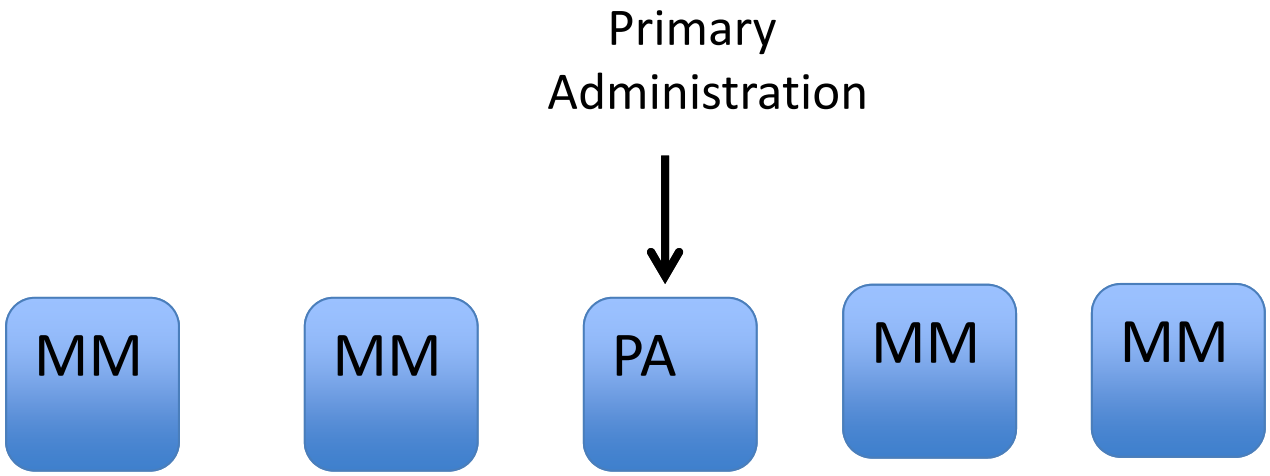
The Handle System is a collection of local handle services, each of which consists of one or more replicated sites, each of which may have one or more servers.



Handle System

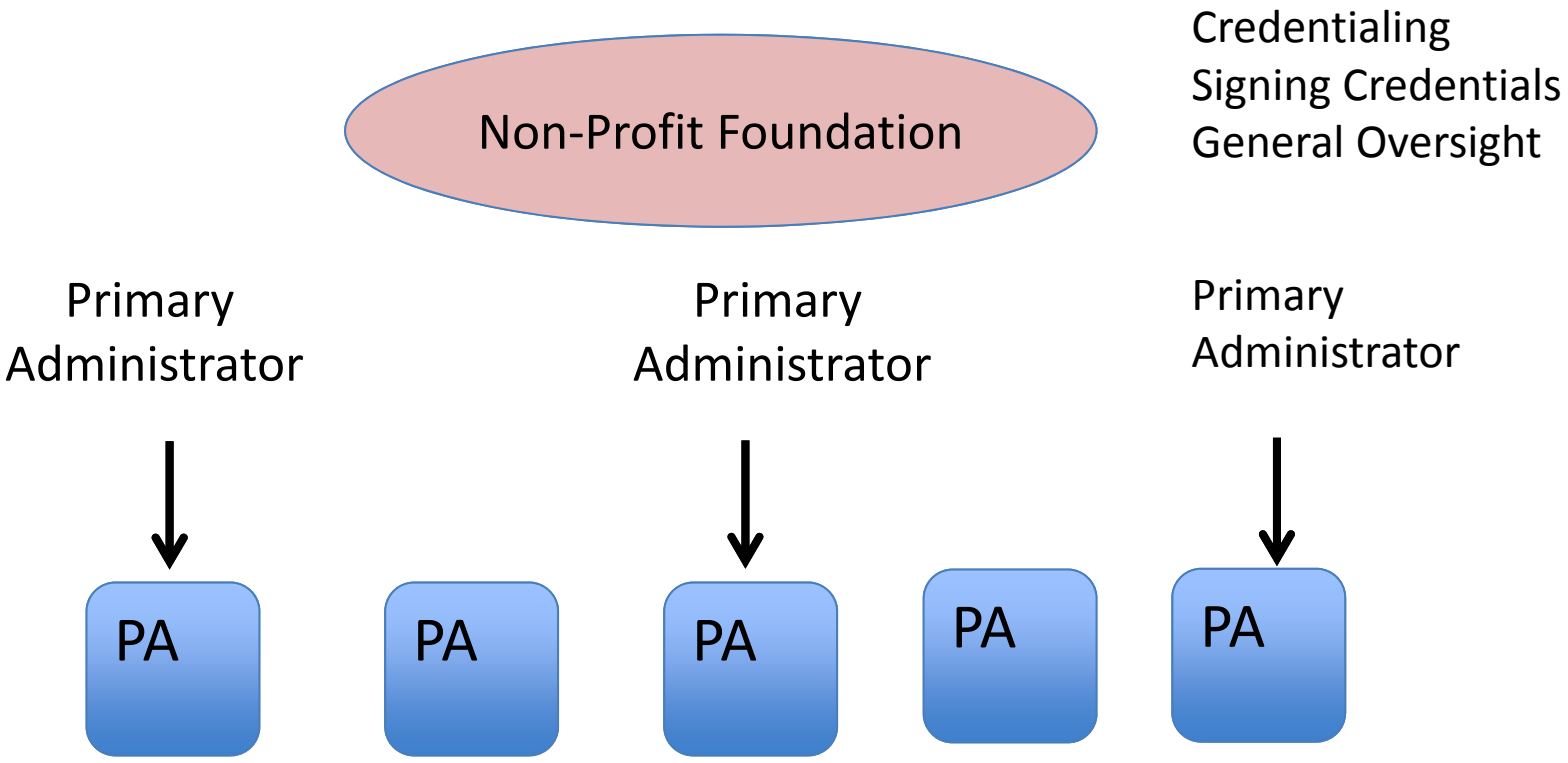
- Provide basic identifier resolution services
  - Resolves digital object id to current state data
  - Id can persist over changes in location, ownership, and other attributes
  - Only the state data changes
- System currently consists of a Global Handle Registry (GHR) and many distributed local handle services
  - Each service responsible for defined subset of id space
  - Each service, including the GHR, can itself be distributed and consist of many servers – thus is scaleable
- Resolution returns type/value pairs
  - Typing is itself scaleable; handles are used as type identifiers
  - No limit on number and length of type/value pairs
  - Each value includes permissions and time to live (TTL)
- Distributed handle administration in the Internet
- Handle System Protocol runs over UDP, TCP, or HTTP
- System is compatible with IPv4 and IPv6
- More information at handle.net site; RFCs 3650 - 3652

Present Administration of the GHR



MM = Mirrors

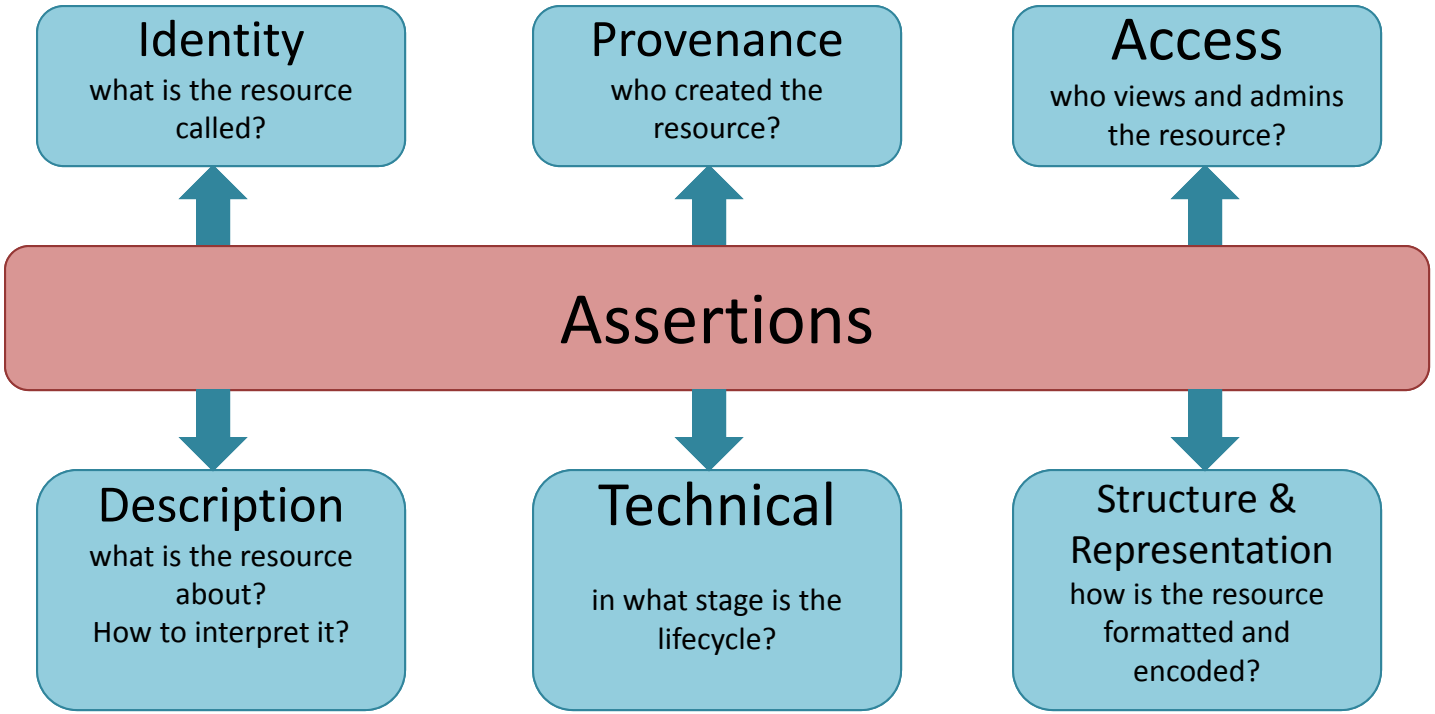
Future Administration of the GHR



We call these Multi-Primary Administrators or simply MPAs

What is Metadata

- People commonly define metadata as “data about data”
- A more complete definition:
  - Metadata is a set of (structured) assertions about an entity/resource
  - Multiple parties may make those assertions
  - Veracity of those assertions is usually outside the scope of metadata
- Those assertions could be about



Registry Design: Metadata Design

- **Identify the model(s)**
  - What is the registry for? Books, Movies, Documents, etc.
  - Oftentimes, a registry manages multiple models: Books and Publishers, Movies and Actors, etc.
  - Include models for *user* and *group* if ownership and sharing functionality is needed
- **Identify the properties for each model**
  - What do you want to capture about Books? title, description, author, genre, publisher, etc.
- **Identify the attributes for each of the properties**
  - What is title's data-type? text? How about genre's? controlled vocabulary?
  - Do you expect multiple titles for a Book? Probably not. Do you expect at least one title to exist? Probably yes.
  - Handles: Which properties should have separate identifiers?

Registry Design: Metadata Design (cont'd)

- **Identify the structure of the properties**
  - Perhaps author property is more than just a name. Author could be a parent node to name, address and organization properties
- **Identify the locale characteristics**
  - What natural languages are expected in metadata?
  - What formats are expected for a date property? MM-DD-YYYY or DD-MM-YYYY
- **Identify the representation**
  - What are the input and output formats? XML, JSON?
- **Design a schema based on answers to above questions**