Background and Overview of the Digital Object Architecture

By Robert E. Kahn

Corporation for National Research Initiatives

Reston, Virginia USA

IMSAC Meeting
Chester, Virginia
December 5, 2016

Fundamental properties of the Digital Object Architecture

- It is based on the same architectural ideas that are embedded in the Internet's architecture and which have sustained its evolution, the two most important of which are:
  - **Open Architecture** (defined protocols & interfaces)
  - **Independence** from the underlying technology

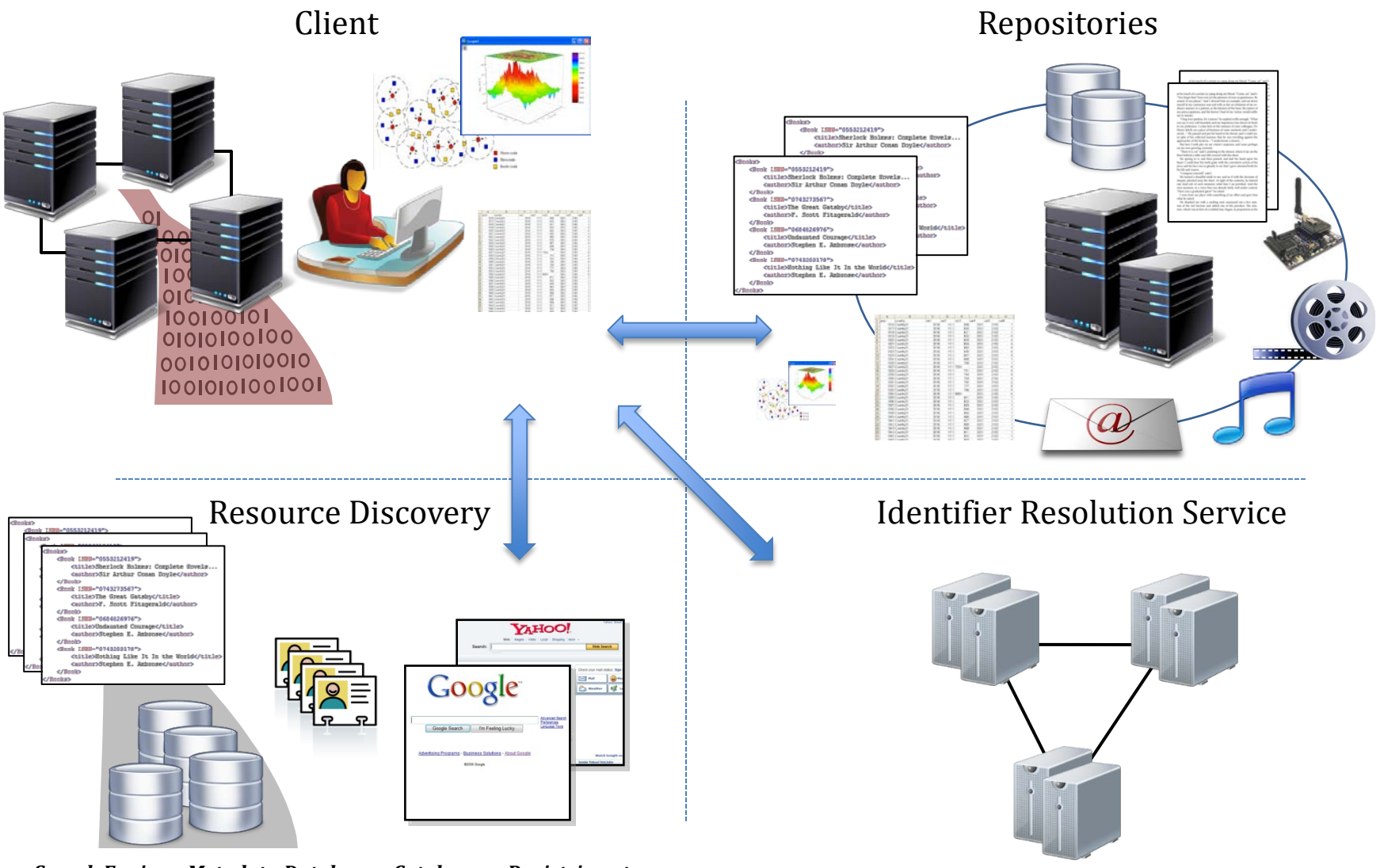
Bindings to Technology vs. Information

- Arpanet – 16 bit addresses → wires
- Internet – 32 bit IP addresses → machines
- Web - URLs → <IP Address/filename>
- Digital Object Architecture
  - Describes a means of managing information over both short and long time frames
  - Digital Objects are the basic structures
  - includes a resolution component that resolves identifiers to “state information” about the desired information
  - such as access means, multiple locations, authentication, public keys, and terms and conditions for use.

General Characteristics of the Digital Object Architecture

- Basic starting point is the concept of a "Digital Object" (DO) defined as a set of bits, or set of sequences of bits, having an associated unique persistent identifier; and one or more DOs may be integrated as a single operational entity.
- Describes the management of DOs with an "open architecture approach," and supports direct interaction with DOs using identifiers.
- Enables short-term or longer-term storage/retrieval, and term & conditions as appropriate.

Digital Object Architecture: Information Management on Networks



Search Engines, Metadata Databases, Catalogues, Registries, etc.

Framework for Discovery
ITU-T Recommendation X.1255

- Based largely on the Digital Object Architecture, ITU-T Recommendation X.1255: “Framework for discovery of identity management information” was approved in September 2013.
- Focused specifically on discovery and access to information in digital form, X.1255 is applicable to operational requirements for information management more generally.
- For purposes of X.1255, a digital object is defined as a digital entity; and the Recommendation describes a data model and interface protocol.

How the DOA Applies to Networking?

- Enables the requirements for processing digital objects to be easily understood in real-time
- Provides a methodology for understanding the structure of digital objects based on "**types**"
- A challenge for the future is to develop ways of describing information structures without the need to preserve the actual information structures: a digital object may represent such information descriptions.

Manifesting the Information (by the user)

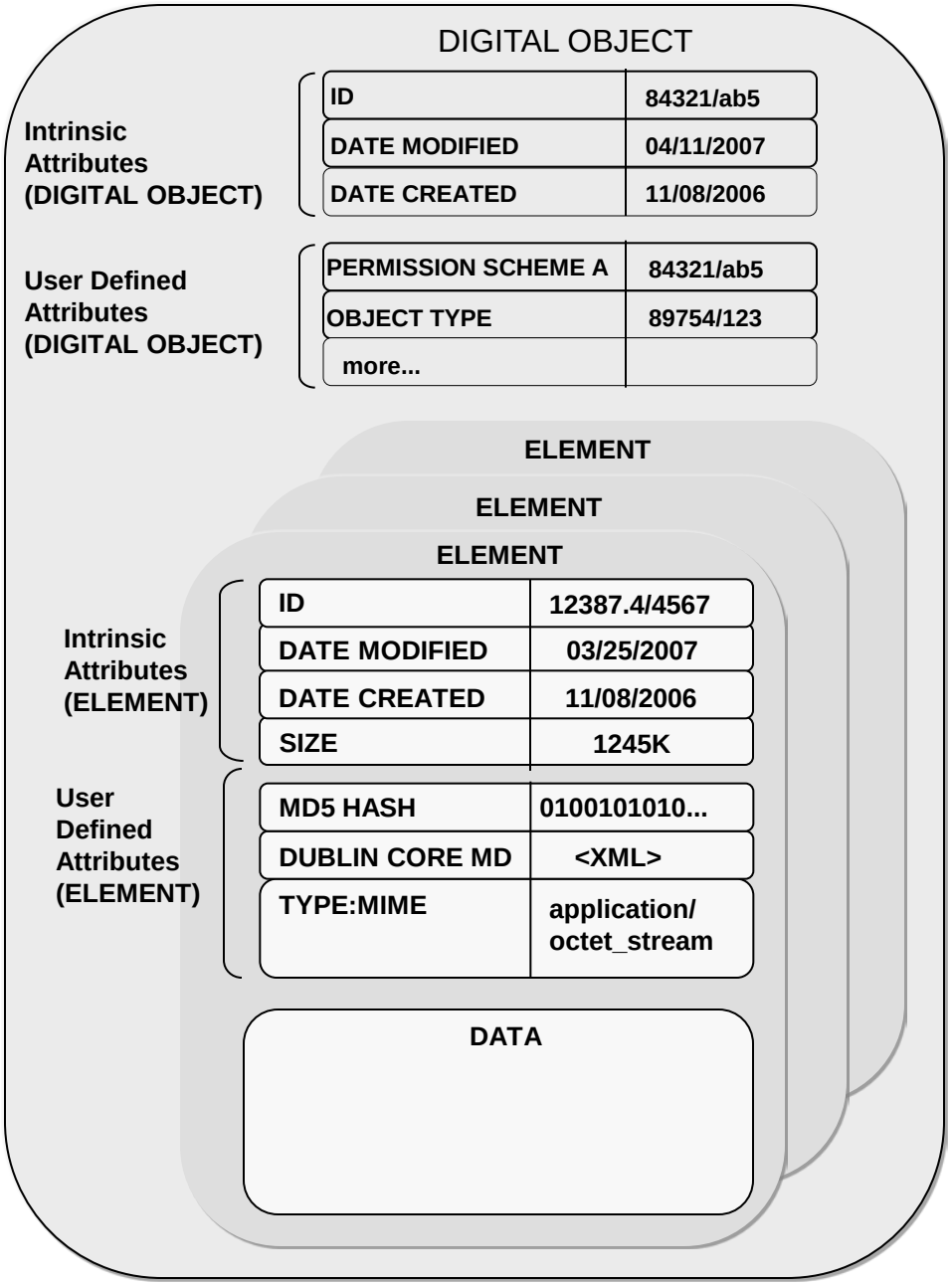
- **Proprietary data formats** – need to preserve programs that can interpret the formats, and the environments for running those programs
- **Open data formats** – need to preserve software interpreters for those formats and the environments for running them.
- **Information Structure Descriptions** – need to maintain resolvable type information to recreate the original data structure so that it can be processed with current technology.

Components of the DOA

- **Identifier/resolution mechanism**, known as the Handle System
- **DO Repositories** store DOs and enable access by means of identifiers
- **DO Registries** store metadata records about DOs and enable them to be found by searching.
- Registries use Repositories; Repositories require registries.

CORDRA

- An integration of the DO Repository and DO Registry components
- Available for download from cordra.org in one of two modes:
  - Experimental mode to evaluate the technology where CNRI provides both the prefix to use for identification of DOs and also provides the handle resolution service
  - Regular mode where the party downloading the software must make separate arrangements to obtain a prefix and the handle resolution service



- Each Object contains structured data and extensible metadata
- Key metadata is the Object's associated unique persistent identifier
- Metadata also includes types, dates, permissions, and other relevant attributes

DATA MODEL

The Handle System ®

- **A basic identifier/resolution system for the Internet.**
  - **Resolves a digital object's identifier to that object's current state information**
  - **Identifier persists when location and other attributes of the object changes.**
- **Logically a single system, but physically and organizationally distributed; it is highly scalable**
- **Associates one or more typed values, e.g., IP address, public key, URL, metadata, to each identifier.**
- **Secure resolution and administration using an integrated PKI capability as an option; optimized for speed and reliability.**
- **Open, well-defined protocol and data model, IPR free.**
- **Provides infrastructure for a wide range of application domains, e.g., digital libraries, network management, service discovery, and IoT.**

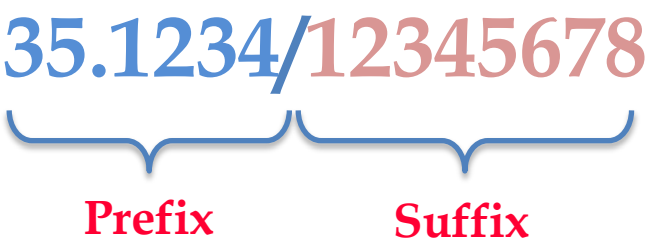
Handle System Security Features

- Authentication
  - Using an optional PKI capability.
  - Handle server and client authentication.
- Authorization
  - Handles and associated handle records are administered by authenticated and authorized digital entities such as an organization providing local handle services.
  - A handle service can restrict access to any of its values in a handle record.
- Confidentiality
  - All handle requests and responses can be encrypted.
- Non-Repudiation and Integrity
  - Handle record responses may be signed by the hosting server
  - Handle records may be signed by any authorized administrator.
- Audit logs
  - All handle servers may log accesses.

Structure of the Handle System

- System currently consists of a Global Handle Registry (GHR) and many distributed local handle services
  - Each service responsible for defined portion of the identifier space
  - The GHR is distributed and scalable; each local handle service can itself be distributed – and may be separately branded
- Resolution returns a handle record containing type/value pairs
  - Typing is itself scalable; handles are used as type identifiers
  - No limit on number and length of type/value pairs
- Each handle record may provide specification for processing the digital object
- Supports distributed handle administration in the Internet
- Handle System Protocol runs over UDP, TCP, or HTTP
- System is compatible with IPv4 and IPv6

What is a Handle?



- Handles (or more generically “digital object identifiers”) are globally unique and resolvable
  - Prefixes are allotted to local handle service providers and most prefix handle records are currently stored in the Global Handle Registry (GHR).
  - A handle prefix is typically resolvable by the GHR to an IP address for a handle resolution service such as an organization providing local handle services.
  - The full handle is resolvable by the handle resolution service into set of typed values.
- Character Set: Unicode 2.0
- Encoding: UTF-8
- Prefix: Currently allotting only numeric values.
- Suffix: No restrictions.

Handles Resolve to Typed Data

Handle	Data Type	Handle Data
35.1525/b.2009.59.5.9	HS_ADMIN	handle=0.na/35.1525; index=200; [delete hdl,add val,read val,modify val,del admin,add admin,list]
	URL	http://www.caliber.net/abs/35.1525/2009.59.5.9
	35.TYPE/DEVICE	35.1/1.2.3
	10320/loc	<location id="1" cr_type="MR-LIST" href="http://www.acme.org/iPage?doi=35.1525%2Fbio.20.5.9" weight="1" /> <location id="2" cr_src="unca" label="SECONDARY_BIOONE" cr_type="MR-LIST" href="http://www.bioone.org/doi/full/35.1525/ bio.2009.59.5.9" weight="0" /> </locations>
	Processing	Instructions go here!
	HS_PUBKEY	0000000B4453415F5055425F4B4559000000000015009760508F15230B….
	HS_SIGNATURE	eyJhbGciOiJSUzI1NiJ9.eyJkaWdlc3RzIjp7ImFsZyI6IlNIQS0yNTYiLCJkaWdlc….

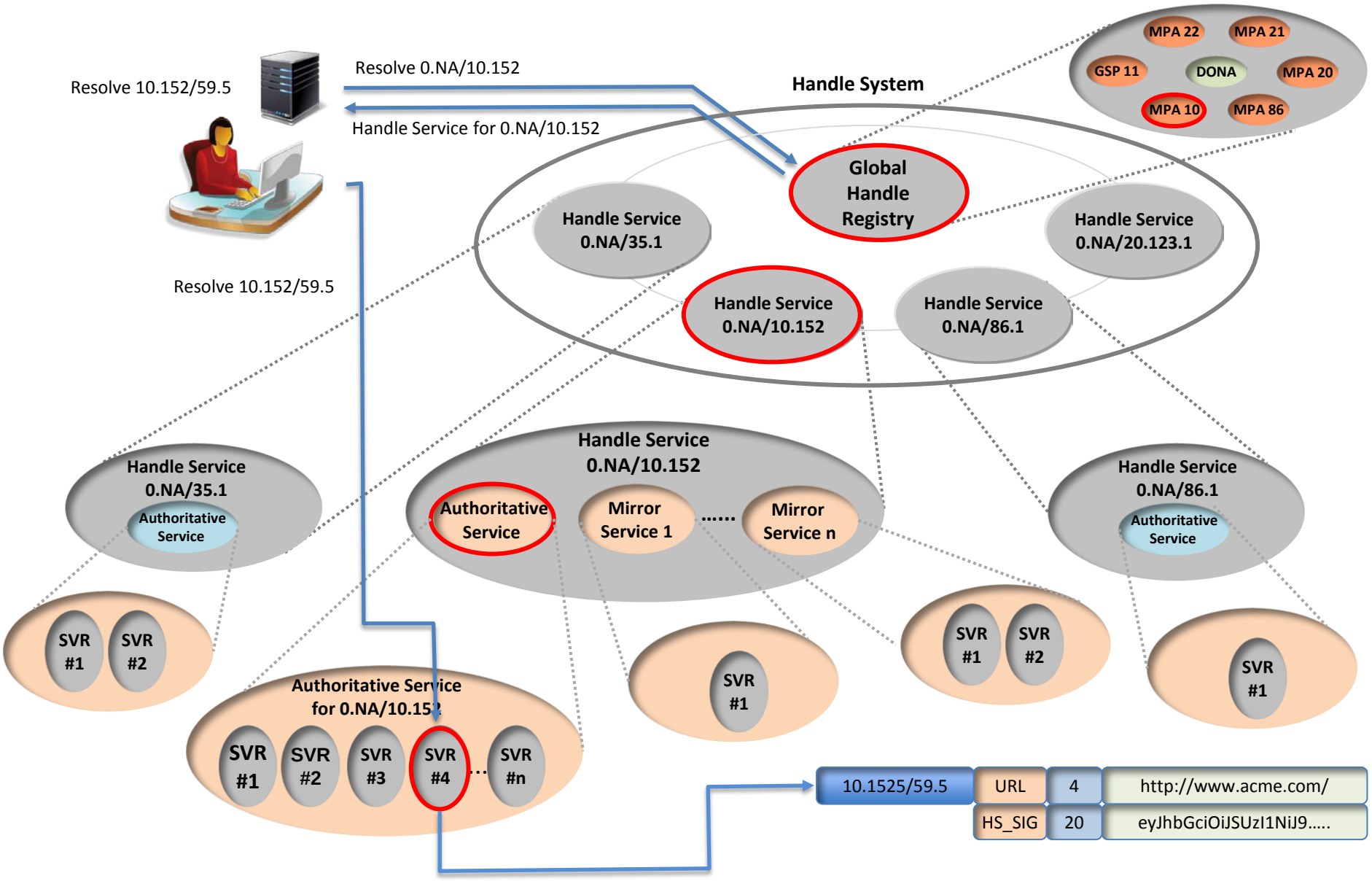
Data Types are also resolvable handles and can be specific to:

- The Handle System (*)
  - **HS_ADMIN**
  - **HS_PUBKEY**
  - **HS_SIGNATURE**
  - **URL etc…**
- An application or service
  - **10320/loc**
- A group/community
- A device type

Types should be identified with a handle and resolve to a type description.

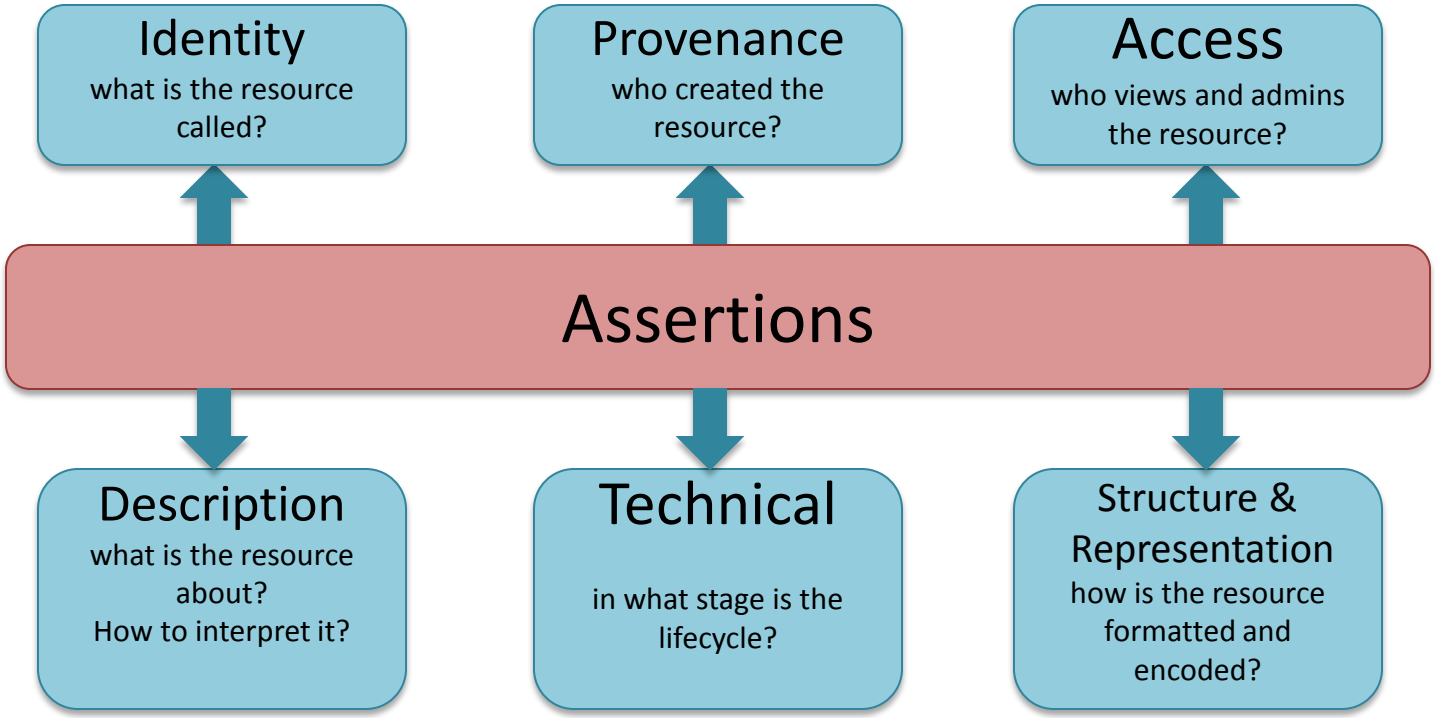
(*) Handle System types are registered as handles starting with the “0.TYPE/” prefix. (URL -> 0.TYPE/URL)

Handle Resolution - Overview



What is Metadata

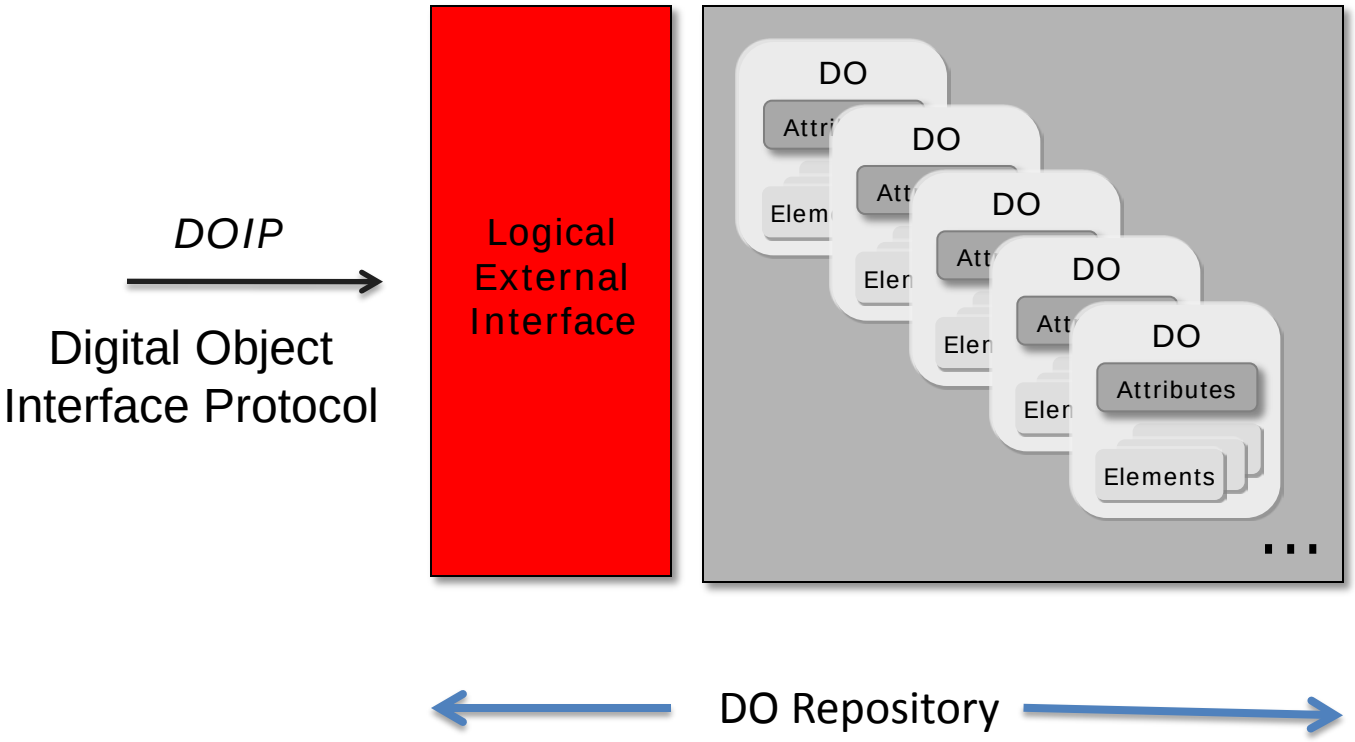
- People commonly define metadata as “data about data”
- A more complete definition:
  - Metadata is a set of (structured) assertions about an entity/resource
  - Multiple parties may make those assertions
  - Veracity of those assertions is usually outside the scope of metadata
- Those assertions could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



Digital Object Interface Protocol

- Allows access to the entire object, or parts of it (each DO consists of multiple elements)
- Some elements may be DOs by themselves or contain identifiers for other DOs
- Enables global interoperability of repositories with security
- Assumes that state information about resources, users, and organizations are represented as digital objects and may be referenced by their unique persistent identifiers

DO Interface Protocol



Certain DOs contain desired data or information
Other DOs contain metadata

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

DONA Foundation

- Non-profit organization established January 2014 in Geneva, Switzerland
- Currently 9 members of the Board of Directors
- Provides overall administration of the Global Handle Registry (GHR), will support relevant standards and outreach activities, including pilot projects, to increase awareness of the DOA
- Currently, seven organizations have signed MPA Service Agreements; and several others are under consideration.
- Approximately 12 MPAs are anticipated by 2018.

Concluding Remarks

- For real-time communications, information as to how best to process a digital object is useful
- Context may be needed to support interoperability across heterogeneous information systems
- System interoperability is critical, and also requires security for more wide-spread application
- Some level of abstraction is necessary for long-lived systems, since the underlying technologies will surely change over time
- The Digital Object Architecture is an effective choice to satisfy these objectives