Data Type Registries (DTR) WG

RDA P3 (Short)

March 2014

Larry Lannom
Corporation for National Research Initiatives
http://www.cnri.reston.va.us/
http://www.handle.net/

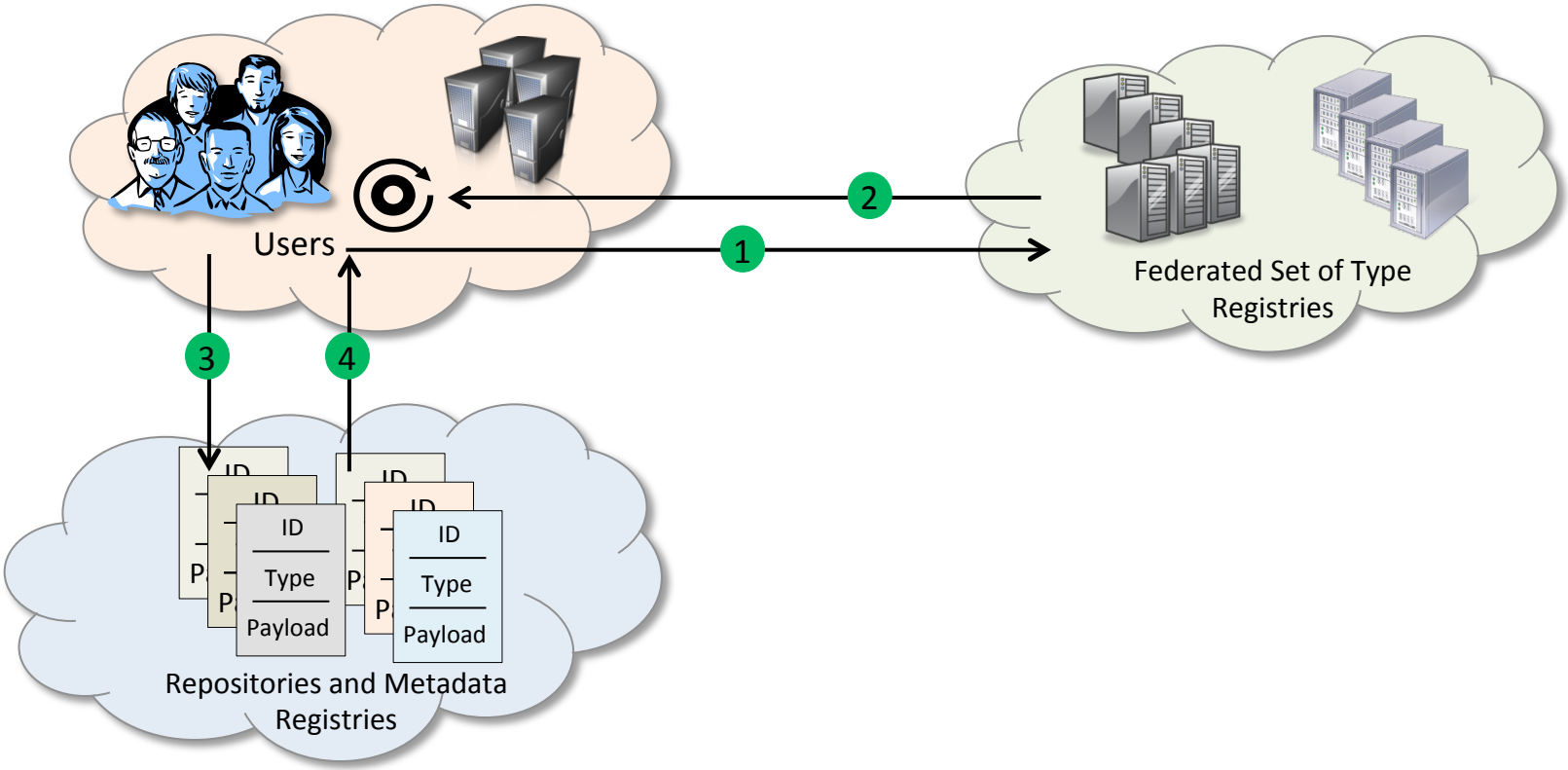
What are Data Types?

- Characterize data structures at multiple levels of granularity
  - Serve as macro or shortcut for understanding and processing data
- File formats & mime types are examples of solved problems at the container level but don't solve finer grained interpretation
  - It's a number in cell A3 but what does it mean
- Other structures with more limited use, e.g., many sci. data sets, may need multiple levels of typing
- Data types enable humans and machines to discover, process, and reason about data

RDA Data Type Registries WG

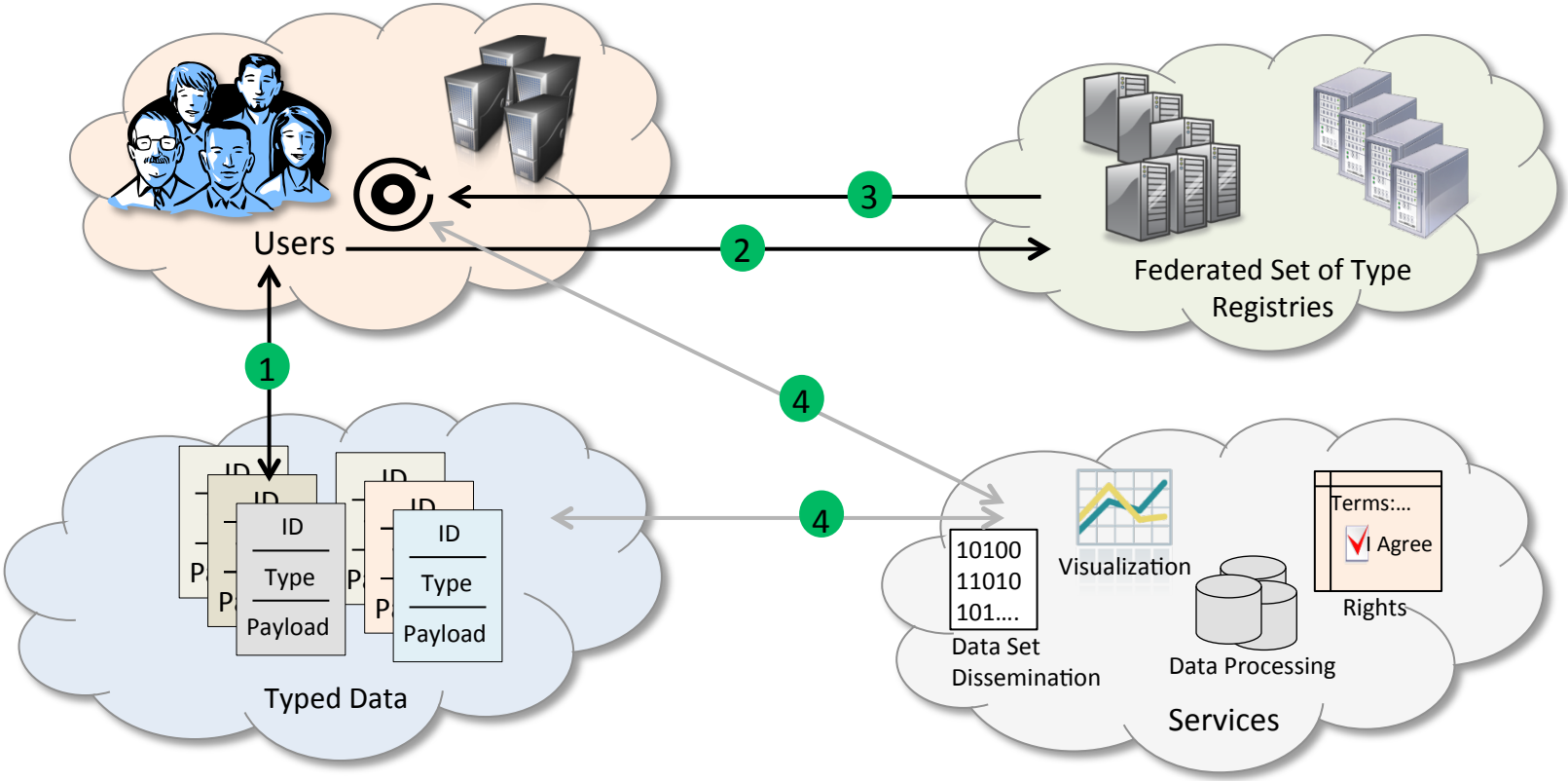
- Goal: Interoperable set of Type Registries
- Each type registered with unique identifier
- Common data model and expression
- Associate with services, tools, format registries, etc.
- Common API for machine consumption
- Schedule
  - 3/2013 – 9/2013
    - Gathering use cases
    - Investigating other work in the area
    - First drafts of data model and functional specs for a type registry
  - 10/2013 – 12/2013
    - Refine data model and functional specs
    - Deploy initial prototype
  - 1/2014 – 5/2014
    - Finalize data model and functional specs
    - Deploy functional type registry for PID types
    - Release turnkey registry conforming to functional specs

Discovery Use Case



1. Clients (process or people) look for types that match their criteria for data. For example, clients may look for types that match certain criteria, e.g., combine location, temperature, and date-time stamp.
2. Type Registry returns matching types.
3. Clients look up in repositories and metadata registries for data sets matching those types.
4. Appropriate typed data is returned.

Process Use Case



1. Client (process or people) encounters unknown type.
2. Resolved to Type Registry.
3. Response includes type definitions, relationships, properties, and possibly service pointers. Response can be used locally for processing, or, optionally 4 typed data or reference to typed data can be sent to service provider.

What do Data Type Records contain?

- Data type records contain
  - textual description for human understanding
  - provenance information (who created when and what)
- Records could contain
  - structured metadata about types for machines to process
  - encoding information (think file formats)
  - service information (think APIs to systems or applications that can process typed data)
  - semantic information (think description or predicate logic, useful for reasoning)
- Records do not enforce or define new ways to describe or represent data structures, but rely on existing frameworks and technologies
  - File formats (mime types), etc., may be used for describing encoding information
  - WSDL, REST APIs, etc., may be used for describing service information
  - OWL, KIF, etc., may be used for representing semantics and knowledge

Proposed Data Type Data Model

Element	Cardinality (min, max)	Notes
ID	(1,1)	A unique, persistent identifier. Assigned by a type registry
Human Description	(1,*)	Description in English mandatory. Descriptions in other languages as needed
Provenance	(1,1)	Who created it, when, etc.
Properties	(0,*)	Properties that describe data. Aka predicates. For example, a weather dataset contains time, location, and temperature properties
Encoding Information	(0,*)	File-formats (mime-types), etc.
Semantic Information	(0,*)	OWL, KIF, etc.
Service Information	(0,*)	WSDL, WADL, APIs, etc.

Proposed Use of Data Types

- Multiple type registries will be deployed; perhaps one per community
- Type registries federate across each other; local policies may restrict (the scope of) such federation
- Users register data structures within a type registry and acquire a unique, persistent identifier (data type)
- Data type identifiers are then associated with corresponding data
- Registered type records are additionally disseminated by type registries as Linked Data compatible outputs
- General Guidelines
  - Users decide what data structures to register or not. If a data structure is expected to play a global role, then users are encouraged to register that data structure
  - Users are encouraged to first search if the data structure is registered prior to registering to avoid duplicates
  - Users decide the encoding, service, and semantic technology or framework that best suits them

Example DTR Use Cases

- Broad Functional Classification
  - Repos hold widely varying levels of data & metadata
  - High-level functional classification of the identified object needed to make sense of what is available, e.g., data object, metadata, repo description, contact info, etc.
- Simple License Information via PID Resolution
  - Data set access conditions cannot be predicted based on ID
  - For DataCite DOIs, a handle/type/value triple could be used to provide access information, probably through a level of indirection, resulting in a pop-up or intervening page or open linked data
- Object Types as a Short-cut for Dependent Services to Match Processing Requirements to Data Objects
  - Using data acquisition as an example
    - Determine object type you are trying to build
    - Consult registry to index into an ontology to dynamically define required and optional properties
    - Does the input data have what is needed?
- Registration of PID Types (in ID/Type/Value triples) for Data Processing and Interpretation
  - Distinguish pointers to objects from pointers to metadata from pointers to services
  - Enable complex client interactions as opposed to simple one-to-one re-direction

Types and the Handle System

- Typing makes sense of data, which is just bits
- Handles resolve to type/value pairs – all other functions reside in the applications
- Handles identify digital entities which are implicitly or explicitly typed
- So – to develop Handle-based applications
  - Must understand the types of returned values
  - Will at some point need to understand the downstream data identified by handles
- Data Type Registries Working Group established within RDA (Resarch Data Alliance)

Initial Prototype: http://typeregistry.org/registrar/