Identifiers, Types, Archives and the Research Data Alliance (RDA)
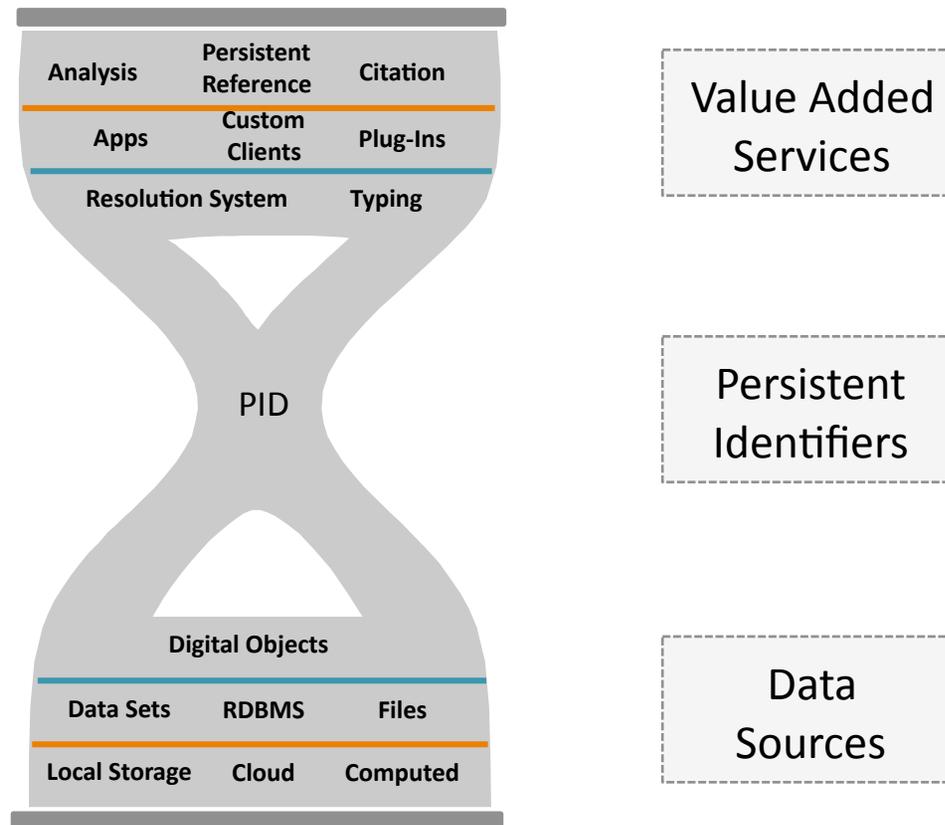
Designing Storage Architectures 2013
23 - 24 September 2013

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

Hourglass Model: Information Management on Networks

Research Data Alliance (RDA)

- RDA's purpose is to enable data exchange. It will not manage or provide direct access to data but will provide structured discussion fora and organize working groups to develop methods and best practices for data management and interoperability

- Supported by science funding agencies in EU, US, AU

- Plenary 1 in Goteborg (240), P2 in DC (~400), P3 in Dublin

- Rapidly growing collection of Working Groups and Interest Groups. See rd-alliance.org

- Many of those Working Groups will address issues around archiving and long term data storage

- Data Type Registries is an accredited Working Group

Goal: Interoperable Set of Data Type Registries

- Data Types
  - Characterize data structures at multiple levels of granularity
  - Formats are just part of the story
  - Optimize interactions between data producers & consumers by having types defined and associated with the data they describe
  - Types should be standardized, discoverable, and unique

- 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

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

One Use of Type Registries
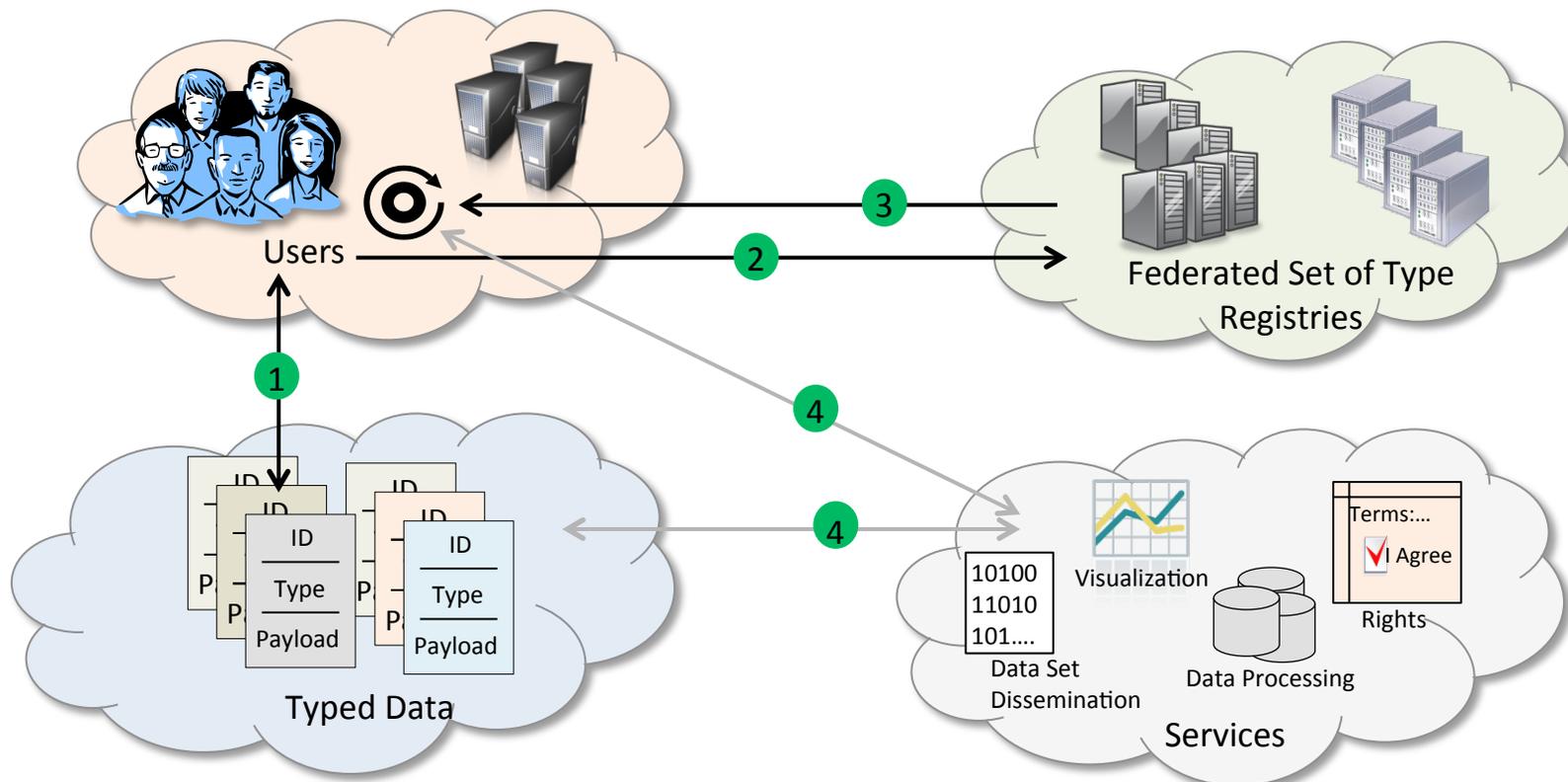
1. Client (process or people) encounter unknown type

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