RDA-PRAGMA Sprint -- Biodiversity Model Replay

PRAGMA 30 @ Philippines

Quan (Gabriel) Zhou, Nadya Williams, Aimee Stewart
Jason Haga, Beth Plale

2/4/16

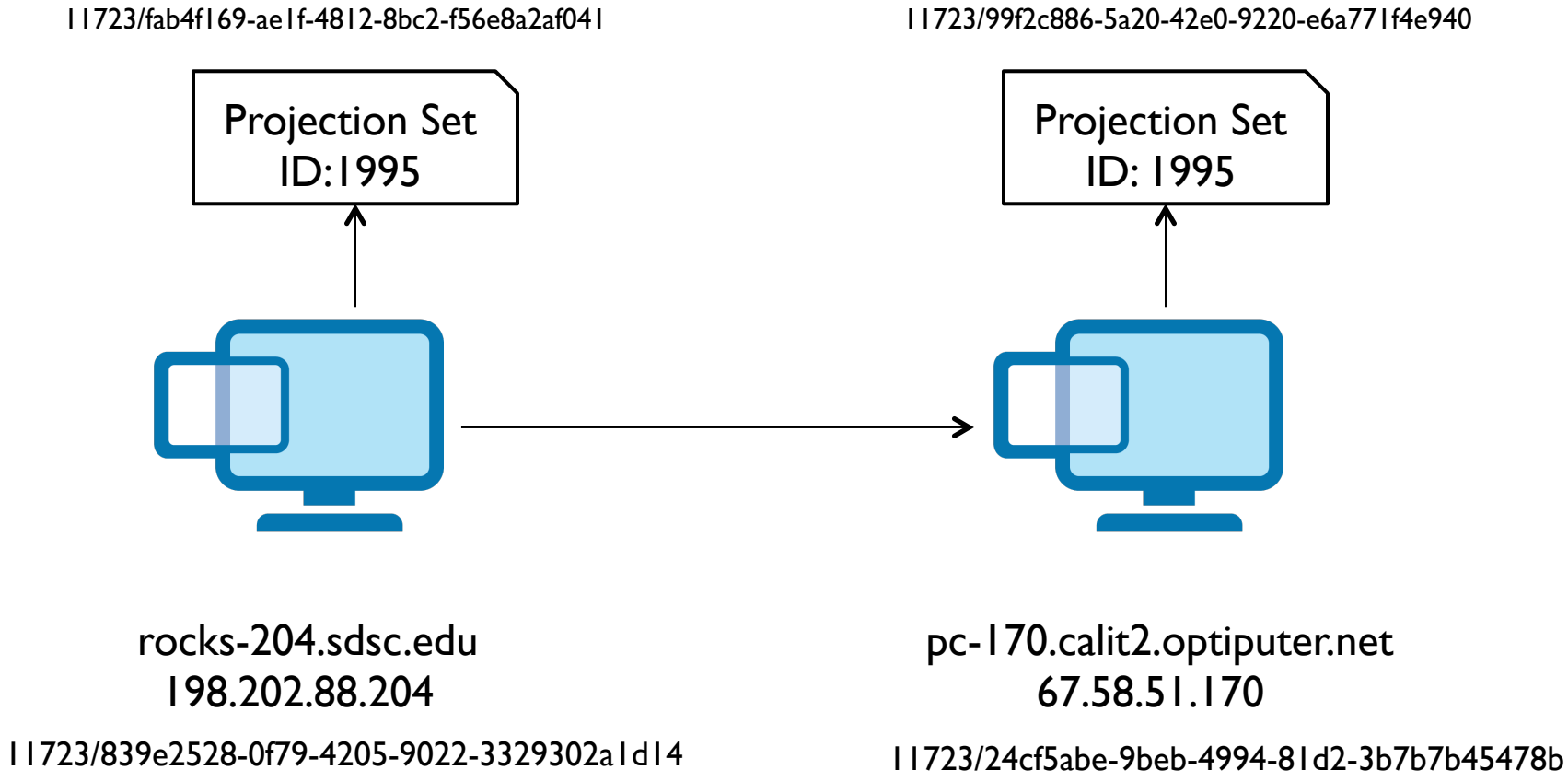
Objectives

- Assess recently released tools and best practices from RDA for contribution to PRAGMA services. Carry out assessment through 2 phase demo.
- Demo: verify lineage of projection data objects, and enable data difference comparison
- Enhancements to PRAGMA testbed: Provide common persistent identifiers with minimum metadata, Information Type Definition and landing pages to VMs and datasets of Lifemapper
- Feed results back to RDA

Demo Phases

- Phase I: Use static GBIF subset for Southeast Asia as input to Lifemapper,
  - dataset bundled into VM.
  - User has ID of two projection result datasets (both result sets have same internal ID (e.g., 317), and uses RDA services to determine whether they came from the same VM or from the primary VM and its clone
- Phase II: Workflow dynamically accesses iDigBio. After seeing that change is to iDigBio input dataset, use new PRAGMA data infrastructure to identify, download, and faithfully replay run with new iDigBio input dataset to visually compare before and after.

Data Diff Comparison Service



New architectural components

- Handle service : handle.net V8 instance
  - What: Assign and resolve handle PIDs for individual data objects
  - Where: Deployed at CNRI server https://38.100.130.12 with prefix 11723
- Landing pages : per objects, to augment metadata repository as UI/web services
  - What: improve accessibility of LM VMs and Objects including metadata and downloading URLs
  - Where: Deployed at pragma8.cs.indiana.edu IU data node.
- Metadata repository: for minimal metadata about data objects
  - What: Used to hold Lifemapper VM instance and data objects and related metadata descriptions
  - Where: 1 primary repo at pragma8.cs.indiana.edu and 2 secondary repos at 2 VM containers; Using MongoDB as metadata repository.
- RDA PIT/DTR Service : PIT server V0.1 and CNRI Cordra Server V1.0.4
  - What: Used to host information type definition of Lifemapper VM instance and data objects for interoperability across service providers
  - Where: Deployed at pragma8.cs.indiana.edu IU data node.

How to find data? - PIDs

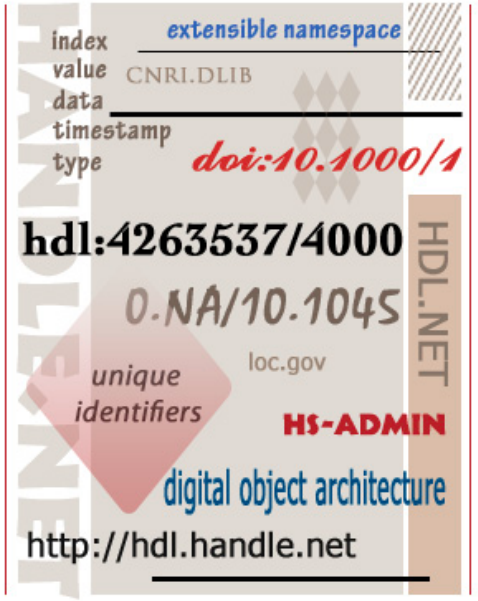
- Persistent Identifier also known as PID that identify a unique individual data object and registered globally (Biometric identifiers like fingerprint for individual)

- Handle.net provides handle service that easily assigns PID to data object and can be resolved global to "find" the target data object

Handle Service @ CNRI

- CNRI hosted a handle server V8 instance for our evaluation;

- Handle instance configurations:
  - https://38.100.130.12:8000/
  - Handle prefix: 11723



RDA-PRAGMA Data Service

- Challenges
  - Lifemapper data objects are coupled with individual Lifemapper VM instance;
  - Generated data products must be exposed and persisted outside the Lifemapper VM.

- Unique Design Decisions
  - Treat Lifemapper VM instances as data objects with proposed VM metadata information type
  - Develop landing page with data store and metadata store to provide long-term data accessibility
  - Benefiting from RDA PIT and DTR service to improve interoperability

Lifemapper VM ID scheme

- Lifemapper VM can be uniquely identified by the following 5 attributes
  - Host IP
  - Rocks version number
  - SpeciesDataset ID
  - EnvironmentalDataset ID
  - Github roll tag

- Proposed ID scheme: single common ID (UUID, DOI, handle) with attributes as part of the minimal metadata stored to RDA Persistent Data Type Service

Metadata management question

- How can my service know what minimum metadata is needed to uniquely define data objects?
- Specifically, is checksum in there? Is the terminology right? (CHSM, checksum, etc.)

Metadata Store
@ IU

MongoDB

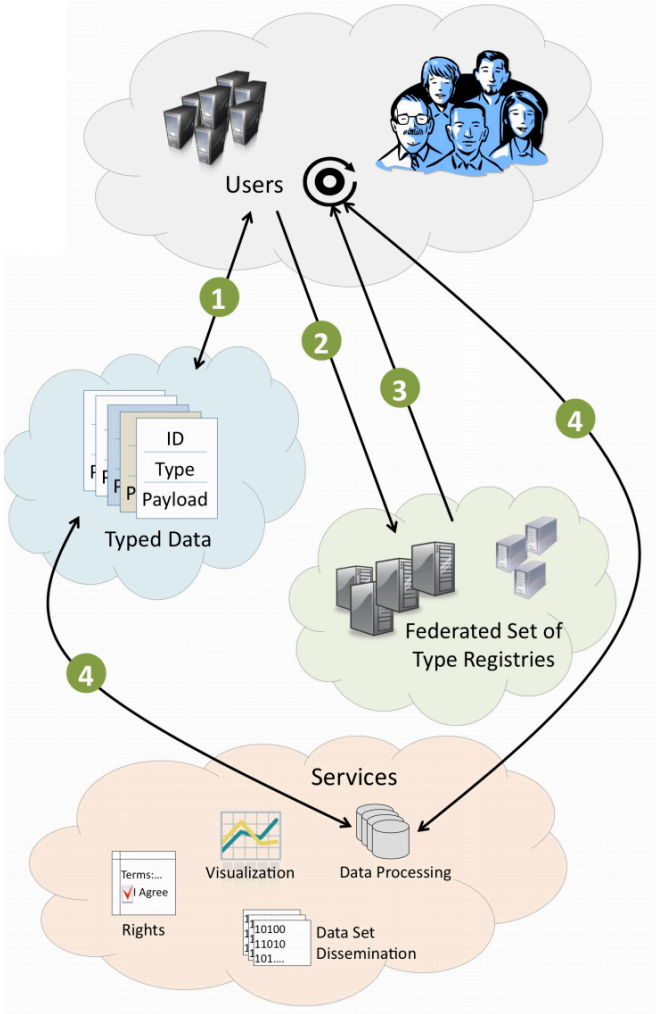
RDA Data Type Registry

- Data (Information) Types are:
  - Characterizations of data at any level of granularity;
  - Identified, defined, and registered

- Registered Data Types are used for:
  - Interpreting data (by humans)
  - Processing data (by machines)

DTR Workflow

1. User encounter data of an unknown type (Identifier metadata or data repository)
2. Users query Type Registries;
3. Type registries response includes type definition that can be used to request an external service;
4. The typed data can be sent to a type-appropriate service or application to be rendered or processed



PID with minimum metadata

- PID records can contain a small subset of digital object metadata; We add "Data Type Identifier" property to define minimum metadata needed to uniquely identify data objects
- Create handle for an occurrence set:
  - 11723/c0fb7b92-4b7d-4fb5-b60a-6fbf8c8280ed
  - {"Landing page address":<URL>, "Data Type Identifier": <Identifier>}

Handle.Net®

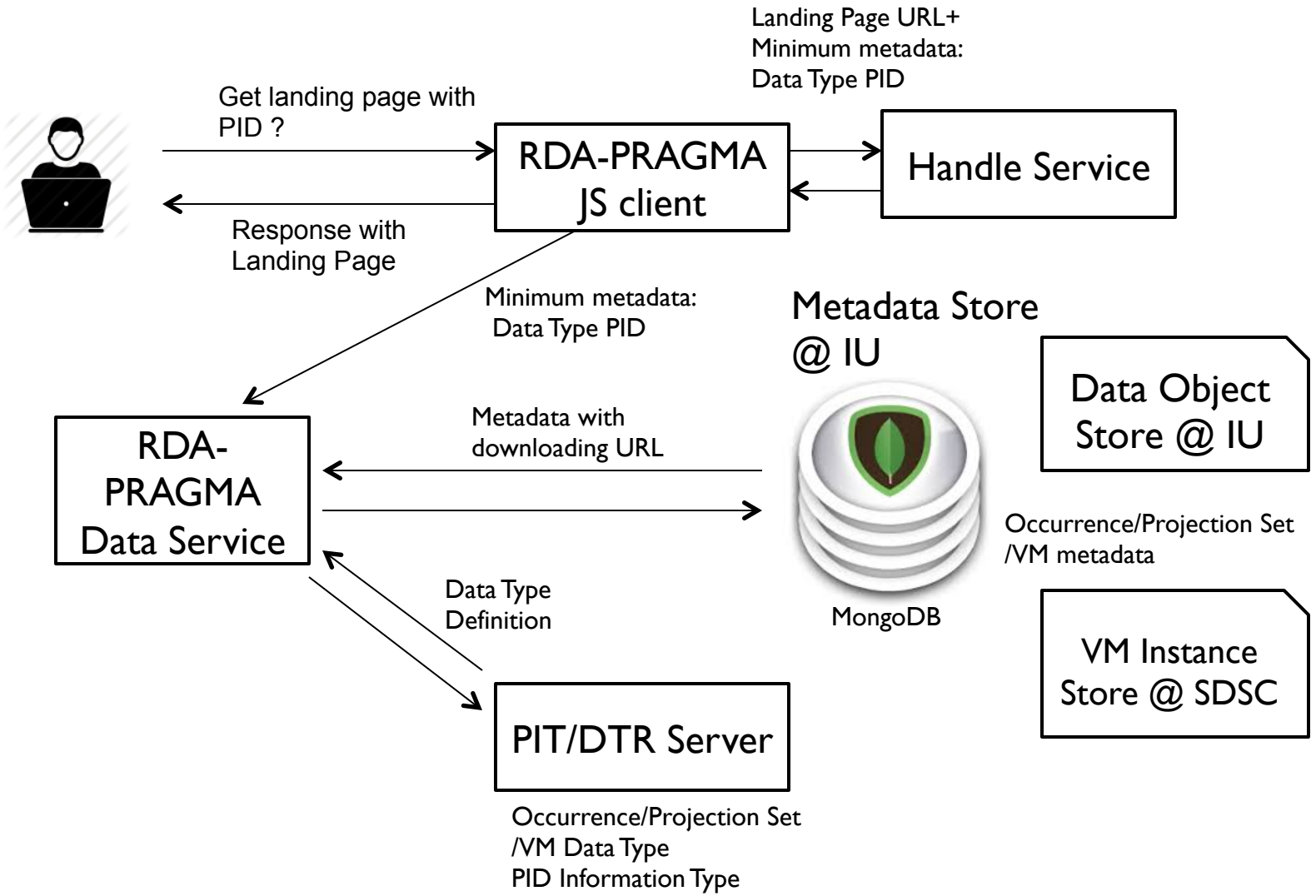
Handle Values for: 11723/c0fb7b92-4b7d-4fb5-b60a-6fbf8c8280ed

Index	Type	Timestamp	Data
1	**11314.2/66af2639d388977e81b85f6413df1e2c**	2016-01-27 19:22:35Z	http://pragma8.cs.indiana.edu:9002/occurrence.html?pid=5988711
2	**11314.2/ac2c3c419edecc485e8c7108d563ed5d**	2016-01-27 19:22:35Z	http://pragma8.cs.indiana.edu:8079/objects/20.5000.239/9e873b2a5690da5b0455

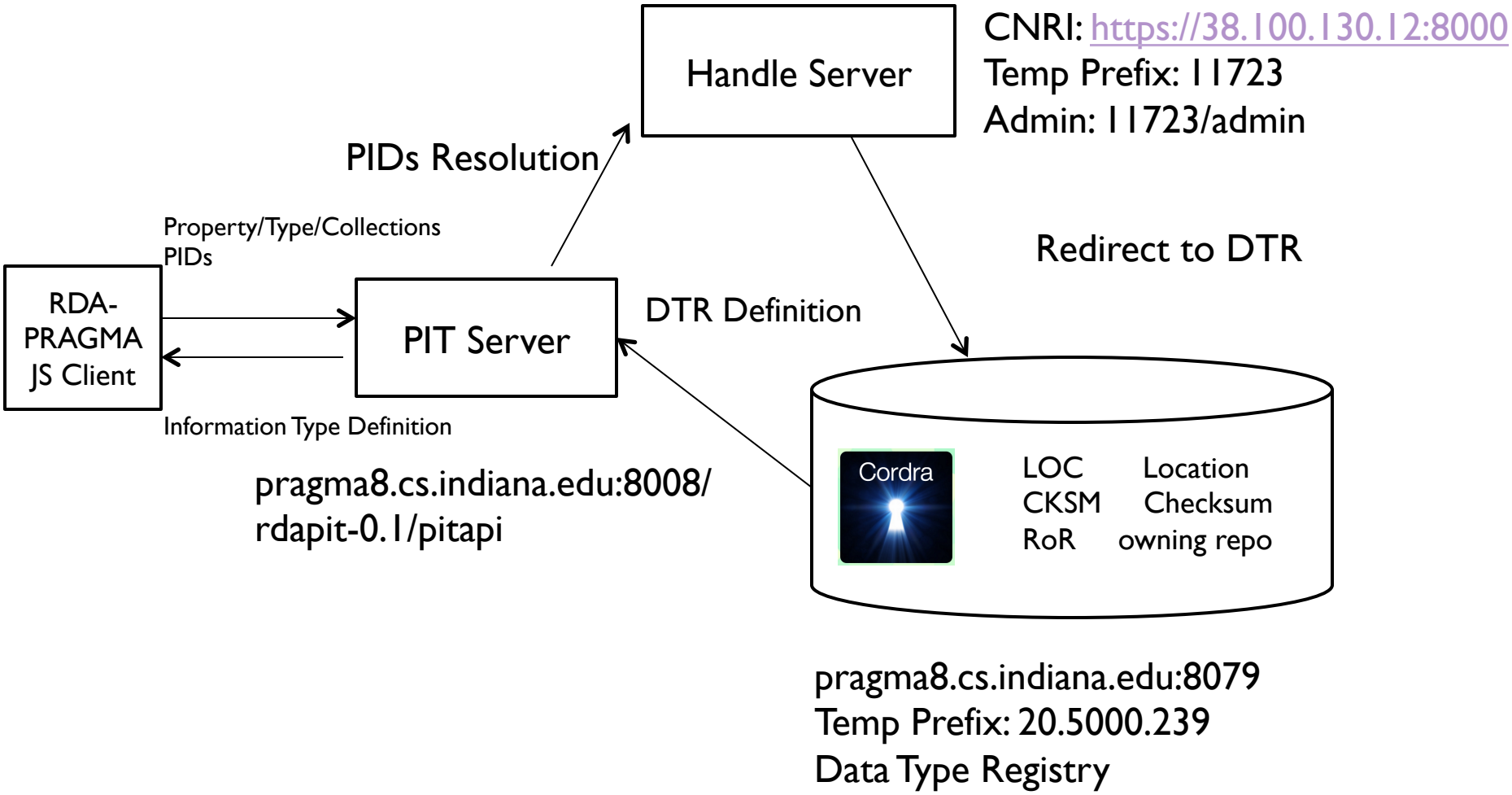
PID Information Type

- Different service providers such as DataCite, CrossRef, Handle can provide identifier metadata which can use different terminologies;
- PIT WG develops an API enables consensus on some essential types;
- In this model, every PID record consists of a number of properties. Every property bears a PID and its essential elements are a name, a range and a value. Only the PID and the value are stored in PID records, while the name and range are available from the registered property definition in the data type registry.

RDA-PRAGMA Data Service



RDA PIT/DTR Architecture



Information Type @ RDA PIT/DTR WG

- RDA PIT working group allows service providers agree on a common API, register their information types in a common data type registry and agree on some core types;

- We deployed the PIT/DTR services on PRAGMA IU nodes and registered information types of Lifemapper generated (projection) sets and Lifemapper VM instances with useful metadata units.

Future Work

- Pull input datasets out of the Lifemapper VM and make the Lifemapper VM automatically responsive to changes to the input data sources such as GBIF and iDigBio;

- Provide functionality for users to automatically bootstrap Lifemapper VM instance in order to easily compare data objects against the mutation of VM versions and physical environment settings;

More Information

- For more information, please visit the following URLs:
  - Code Base:
    https://github.com/Gabriel-Zhou/RDA-PRAGMA-Data-Service
  - RDA PID Information Types Working Group
    https://rd-alliance.org/groups/pid-information-types-wg.html
  - RDA Data Type Registries WG:
    https://rd-alliance.org/groups/data-type-registries-wg.html
  - CNRI Handle.Net Registry
    https://www.handle.net/

IU Data Node Resources

- RDA PIT/DTR Service
- Persistent Metadata Repo
- Handle V8 service client
- PRAGMA-ENT Mesh
- Open HathiTrust Corpus

Acknowledgement

- This project is funded by PRAGMA. (NSF OCI 1234983)

- With special thanks to CNRI for hosting handle V8 server for evaluation RDA PIT/DTR tool. We thank Tobias Weigel from RDA for all the instructions and discussions about RDA output.

The End

Mr. Spock:
Hope our data live long and prosper!