RPID: An Overview

Rob Quick (Beth Plale)
PI
Larry Lannom and Alison Babeu (Bridget Almes)
Co-PIs
Chief Technical Contacts
Yu Luo and Scott Teige

The Internet is a worldwide network of connected computers. Computers have an IP address that uniquely identifies a device on network.

Imagine worldwide network of data objects. Data objects persist (until they don't). Objects are findable, accessible, interoperable, and usable (especially reusable)

The Digital Object Architecture serves as base infrastructure only. DOA is silent on issues of modeling data objects themselves: their *content*, their *relationship to their own metadata*, and *relationship between data objects*

For object modeling we turn to FAIR principles and PID Kernel Information

https://www.internetsociety.org/resources/doc/2016/overview-of-the-digital-object-architecture-doa/

https://link.springer.com/article/10.1007/s00799-005-0128-x

Persistent IDs are the backbone of data sharing

[primary and secondary use]

PID makeup

- Handles have a prefix assigned to a Local Handle Server
- Suffix is under control of Local Handle Server
- e.g., RPID testbed assigns only test temporary handles:
  - *11723.1.test, 11723.2.test, ... 11723.8.test :* assigned for internal use
  - 11723.9.test.<*proj name*> : assigned to projects

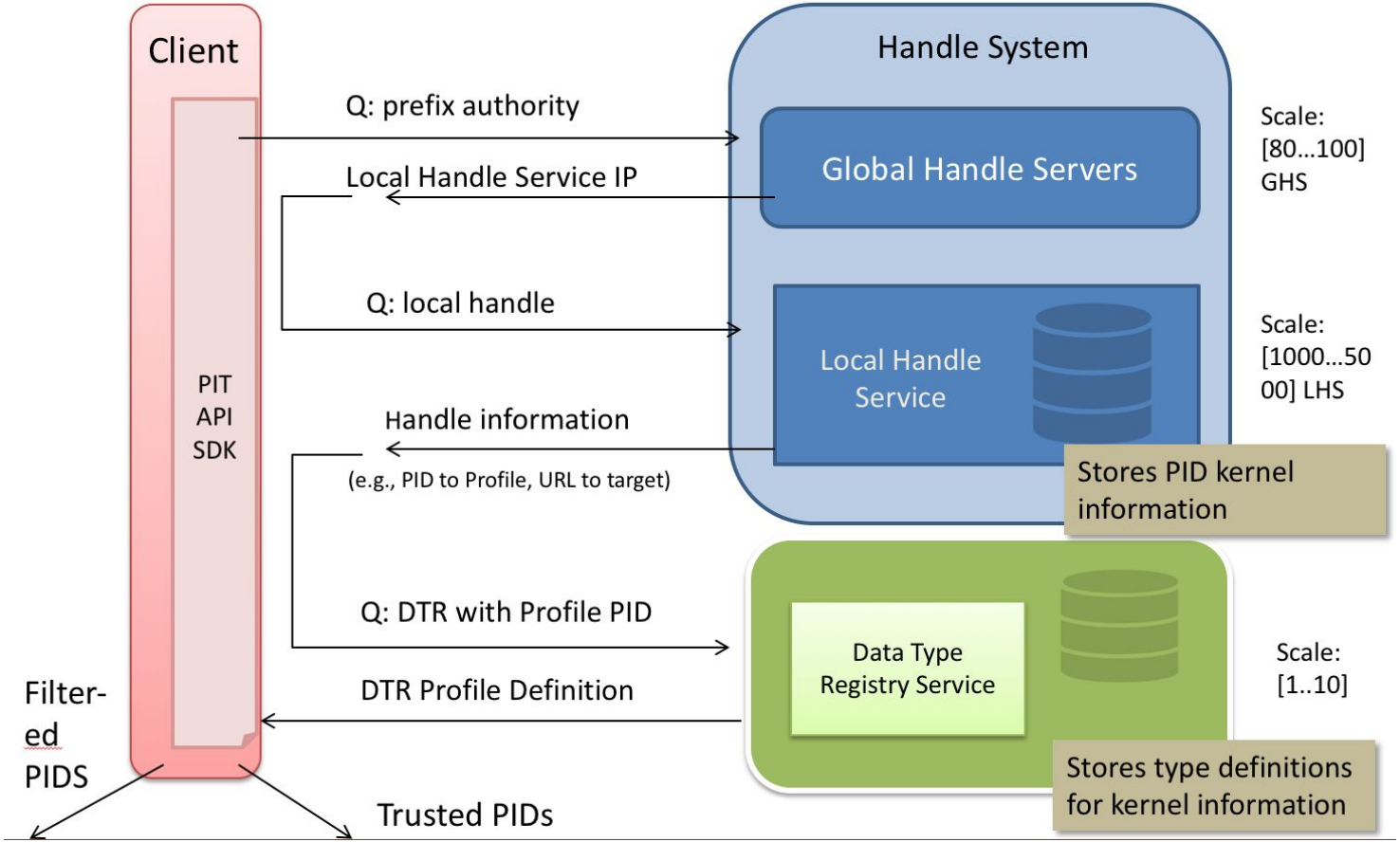
Strawman: Minimal metadata as part of the PID KI

	Type of Content	Content format	Mandatory?	Explanation
1	PID	Handle	YES	Global identifier for the object; external to the PID Kernel Information
2	RDAKIProfileType	Handle	YES	Handle to the Kernel Information type profile; serves as pointer to profile in DTR. Address of DTR federation expected to be global (common) knowledge.
3	digitalObjectType	Handle	YES	Handle points to type defn in DTR. The type of the object (this should always be the same for this type of data, but would distinguish it from other data types). Distinguishing metadata from data objects is a client decision within a particular usage context, which may to some extent rely on the digitalObjectType value provided.
4	digitalObjectLocation	URL	YES	Pointer to the content object location (pointer to DO)
5	etag	Hex String	YES	Checksum of object contents
6	lastModified	ISO Date	YES	Last time of digital object modification
7	creationDate	ISO Date	YES	Date of digital object
8	version	String	YES	If tracked, a numerical version for the object

Strawman: Provenance fields as part of PID KI

	Type of Content	Content Format	Mandatory?	Explanation
1	wasDerivedFrom	IDENTIFIER	False	Transformation of an entity into another, an update of an entity resulting in a new one, or the construction of a new entity based on a pre-existing entity.
2	specializationOf	IDENTIFIER	False	Entity is of another shares all aspects of the latter, and additionally presents more specific aspects of the same thing as the latter.
3	revisionOf	IDENTIFIER	False	A derivation for which the resulting entity is a revised version of some original.
4	primarySourceOf	IDENTIFIER	False	Used for a topic refers to something produced by some agent with direct experience and knowledge about the topic, at the time of the topic's study, without benefit from hindsight.
5	quotationOf	IDENTIFIER	False	Used for the repeat of (some or all of) an entity, such as text or image, by someone who may or may not be its original author.
6	alternateOf	IDENTIFIER	False	Entities present aspects of the same thing. These aspects may be the same or different, and the alternate entities may or may not overlap in time.
7	hadMember	IDENTIFIER	False	A membership relation is defined for stating the members of a Collection.
8	externalW3CPROVDoc	URL	False	A URL referring to a W3C PROV document from an external repository.

Handle resolution in a Digital Object Architecture



Client working with PID Kernel Information looks at each PID in list, accepts those that have:

- Kernel Information profile stored in Data Type Registry (DTR),
- That profile is associated with RDA (in some unspecified manner)
- PID Kernel Information holds tiny amount of data provenance from which basic sense of trust is derived

RPID Testbed

- Suite of software services for use by community
  - Data type registry (RDA)
  - PIT API (RDA)
  - Handle service
  - RDA Collection API
- Exploratory services
  - PID Kernel Information
  - Mapping CTS URNs to handles
  - Packaging for use by others
- Help and advice
- User advisory group

RPID Testbed

36-Month Testbed

- In conclusion, this work proposes
  - data resolution: Digital Object Architecture [Kahn]
  - high level data filtering: PID Kernel Information
  - FAIR principles as data object layer
- Thus contributes to Open Science with foundational infrastructure enabling new ecosystem of data services
- Follow our work at:
  - https://github.com/rpidproject/rpid
  - RDA PID Kernel Information Working Group
  - Reach us at rpid-l@iu.edu

Additional Slides Covering Use Cases and the User Advisory Group Details

SEADTrain Use Case

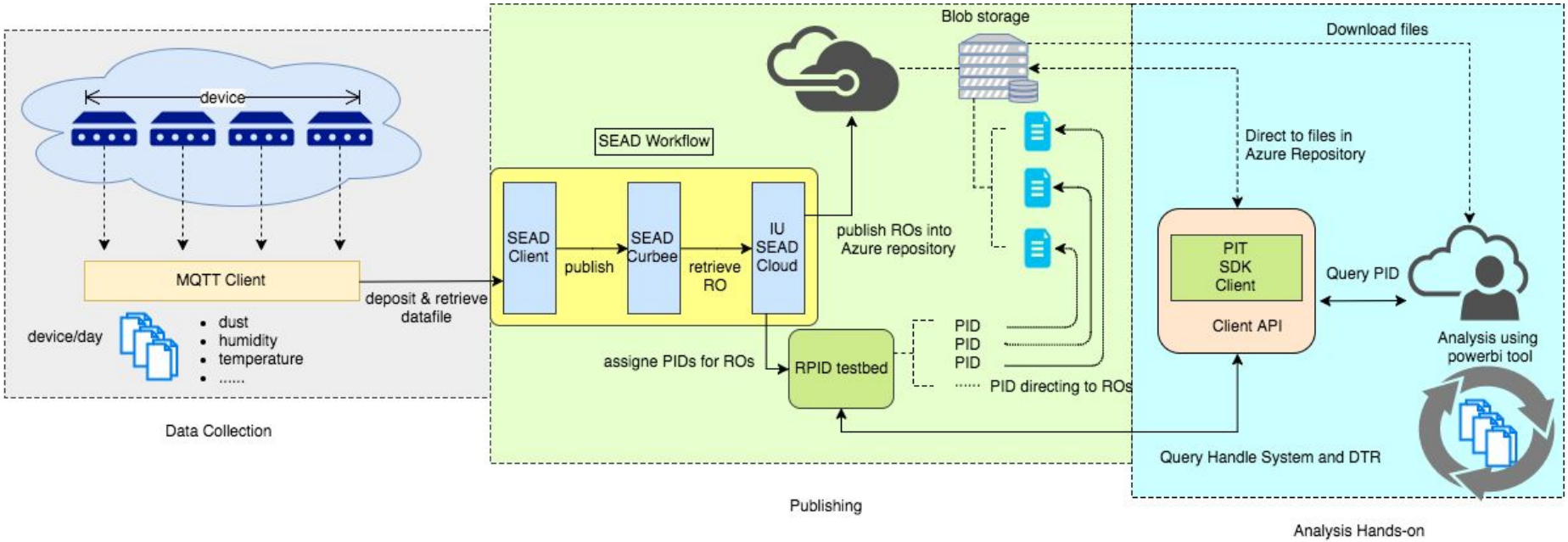
Yu Luo

Description

- Microsoft Azure SEADTrain is to create and evaluate an environment for data analysis that allows students to interact with data. It will extend SEAD's publishing tool suite to support Azure as a destination and will use a persistent identification framework (PIDs) to reference datasets at varying granularity.

- SEADTrain project publishes data from AirBox devices for the purposes of creating data science learning modules. We assign a PID to each daily feed from one device, generating daily files per device, per day from the raw readings. For further analysis, this data needs to be queried and subset for specific time ranges.

Architecture (workflow)



Progress

- Strawman Profile
  - Sixteen fields are developed for PID Kernel Information, describing PID with human-readable information
- Published data
  - IU SEAD Discovery page http://d2i-dev.d2i.indiana.edu:8081/iusc-azure-search/search.html
  - Sample PID:
    - http://hdl.handle.net/11723/test.seadtrain.5e66ce48-236a-4af7-b3e1-8a6700d36abf
  - PID Kernel Information

{"digitalObjectType":"http://hdl.handle.net/20.5000.347/rdastrawman","digitalObjectLocation":"https://iusc.blob.core.windows.net/0a203465-b853-4556-87c0-ac172fb55674/2017_D239_9E65F90C537D.txt","PID":"http://hdl.handle.net/11723/test.seadtrain.5e66ce48-236a-4af7-b3e1-8a6700d36abf","etag":"2f0733b956baf24c3108fee8e9d767de","RDAKIProfileType":"http://hdl.handle.net/20.5000.347/rdastrawman","lastModified":"2017-08-28T00:00:00Z","creationDate":"2017-08-27T00:00:00Z"}

Jetstream Use Case

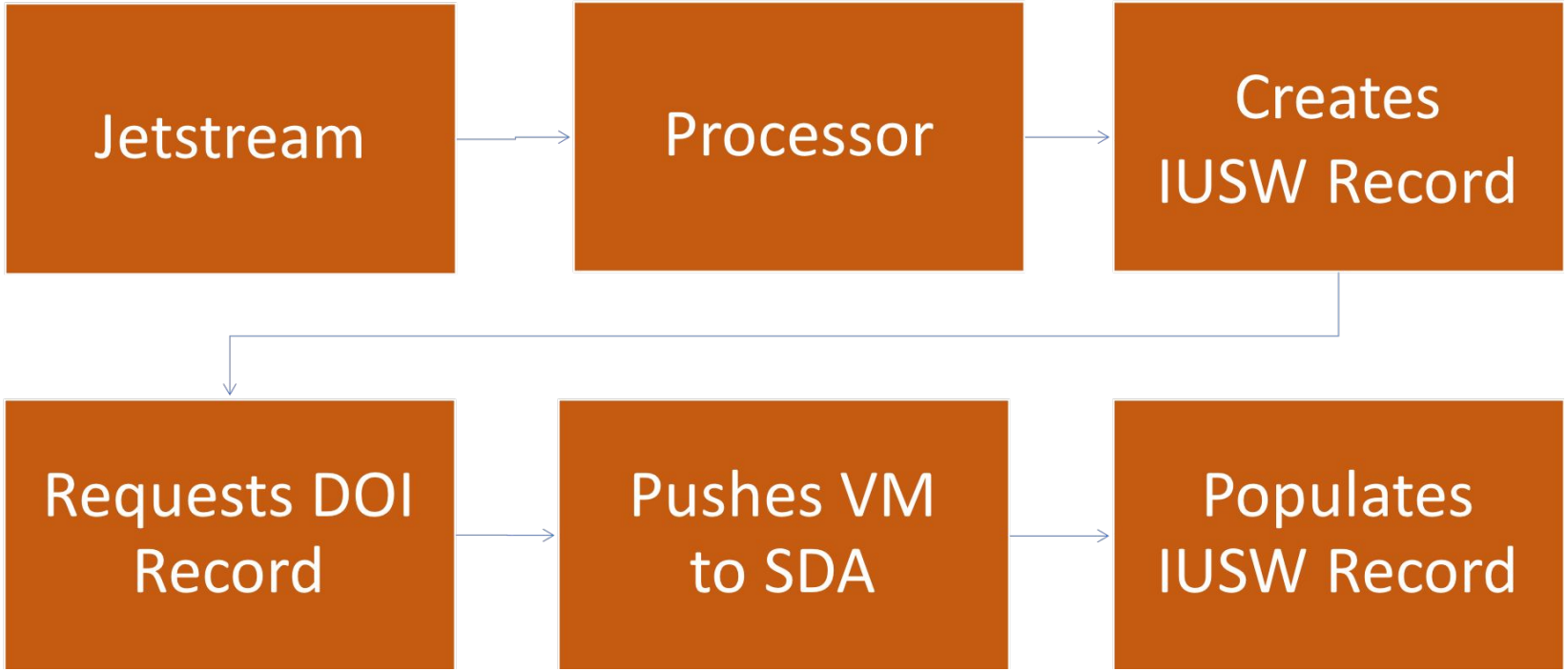
Richard Higgins, Yu Luo, & Robert McDonald

What is Jetstream?

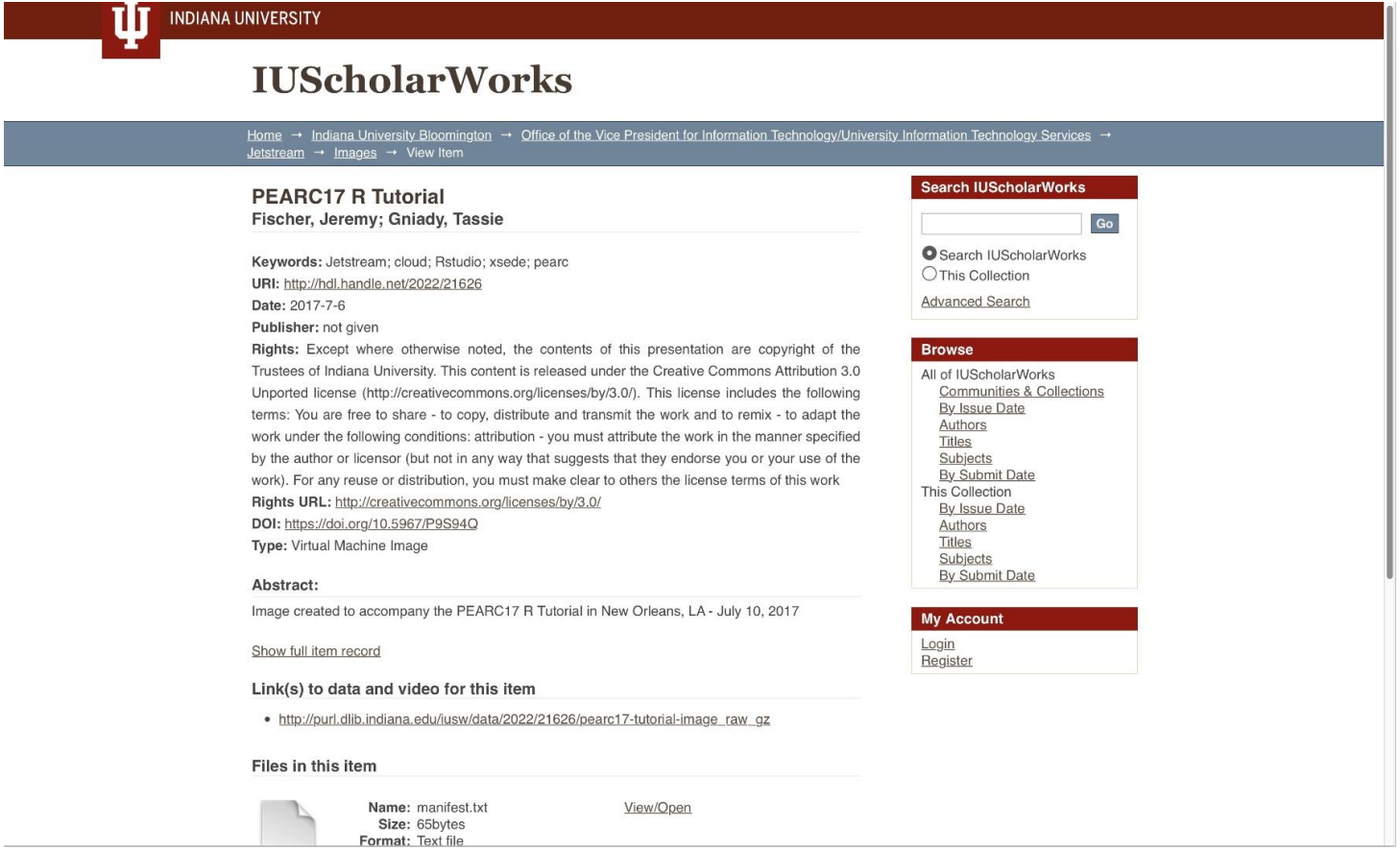
- IU operates the Jetstream project amongst others for national community or regional organizations. Jetstream is the National Science Foundations first production cloud for science and will consist of 640 nodes geographically dispersed into 320-node system.

https://jetstream-cloud.org

Current Jetstream VM Archival Process



Jetstream VM in IUScholarWorks Registry Published



INDIANA UNIVERSITY

IUScholarWorks

Home → Indiana University Bloomington → Office of the Vice President for Information Technology/University Information Technology Services → Jetstream → Images → View Item

PEARC17 R Tutorial

Fischer, Jeremy; Gniady, Tassie

Keywords: Jetstream; cloud; Rstudio; xsede; pearc
URI: http://hdl.handle.net/2022/21626
Date: 2017-7-6
Publisher: not given
Rights: Except where otherwise noted, the contents of this presentation are copyright of the Trustees of Indiana University. This content is released under the Creative Commons Attribution 3.0 Unported license (http://creativecommons.org/licenses/by/3.0/). This license includes the following terms: You are free to share - to copy, distribute and transmit the work and to remix - to adapt the work under the following conditions: attribution - you must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). For any reuse or distribution, you must make clear to others the license terms of this work
Rights URL: http://creativecommons.org/licenses/by/3.0/
DOI: https://doi.org/10.5967/P9S94Q
Type: Virtual Machine Image

Abstract:

Image created to accompany the PEARC17 R Tutorial in New Orleans, LA - July 10, 2017

Show full item record

Link(s) to data and video for this item

- http://purl.dlib.indiana.edu/iusw/data/2022/21626/pearc17-tutorial-image_raw_gz

Files in this item

Name: manifest.txt
Size: 65bytes
Format: Text file
View/Open

Search IUScholarWorks

Go

Search IUScholarWorks
This Collection

Advanced Search

Browse

All of IUScholarWorks
Communities & Collections
By Issue Date
Authors
Titles
Subjects
By Submit Date
This Collection
By Issue Date
Authors
Titles
Subjects
By Submit Date

My Account

Login
Register

Preliminary use of straw man profile (1)

	Property identifier	Content format	Example Value
1	PID	Handle	http://hdl.handle.net/2022/21626
2	RDAKIProfileType	Handle	http://hdl.handle.net/2022/21626
3	digitalObjectType	DCMI	Virtual Machine Image
4	digitalObjectLocation	PURL	http://purl.dlib.indiana.edu/iusw/data/2022/21626/pearc17-tutorial
5	etag	Hash	2f29943275fa5f41c7c5aefbaf8d4382

Preliminary use of straw man profile (2)

	Property identifier	Content format	Example Value
6	lastModified	ISO Date	2017-08-22T08:20:11Z
7	creationDate	ISO Date	2017-06-09T13:46:30Z
8	version	DOI	10.5967/P9S94Q
9	wasDerivedFrom	UUID	7f04320f-8a53-407e-b918-6690b584c6c2
10	specializationOf	Description	R/RStudio Tutorial
11	revisionOf	Hash	d6c3e12ef9b1dcd4b7d593bde280d13a
12	primarySourceOf	URL	https://use.jetstream-cloud.org/application/images/107

RPID Testbed Applied to Jetstream Active VM Management

- Handle System would be a flexible approach of representing Jetstream images and projects.
  - Assigning PID for each individual Project and Jetstream Image
  - Using Strawman Profile to present the kernel information of Jetstream objects
  - Showing human-readable information for users to understand the image and project
  - Potentially providing technical metadata for users to rebuild the project and image in other platforms.

Progress

- Ingesting the metadata information from IUScholarWorks and Jetstream
- Developing a extension profile to support the Strawman Profile for presenting the metadata of projects
- Combining the extension profile and Strawman Profile to be a integrated metadata of PID

PID Wishlist for Jetstream VMs

- Early PID assignment with maximal maschine metadata about the VM
- Management of the PID for VM at the active level (Jetstream OpenStack admin interface)
- Management of the set of PIDs for VM at the archival level with published state Datacite DOI assignment
- Search and discovery of RPID testbed for data gathering for published state DOI assignment

CTS URN Use Case

Alison Babeu & Yu Luo

CTS URN

- The Canonical Text Services (CTS) Protocol defines a URN-based identifier structure for identifying texts and canonically cited passages of texts.
- It also defines a companion CTS Application Programming Interface (API) protocol for a service to retrieve fragments of texts by canonical reference, as expressed by their CTS URNs.
- The CTS URN syntax allows expressions of texts and parts of them to be identified as first class and stably identified data objects.
- CTS-API allows these URN based identifiers to be resolved into data they represent.

CTS URN-A quick example

- urn:cts:greekLit:tlg0012.tlg001.perseus-grc1.
- tlg0012 is the textgroup identifier for Homer, defined as author 0012 in classics canon the *Thesaurus Linguae Graecae* (TLG)
- tlg001 is the work identifier for the *Iliad* also assigned by the TLG
- perseus-grc1 stands for a particular edition/version (1920-Oxford-Allen) that has been published and is available as part of the Perseus Digital Library
- Identifier has no permanence or semantic meaning outside classics domain, desire PIDs (e.g. Handles) for texts to make them permanently citable and referenceable outside digital classics domain

Description

- Handle System with the flexible resolution capabilities is the approach to automatically linking CTS URNs with an instance or instances of the CTS APIs

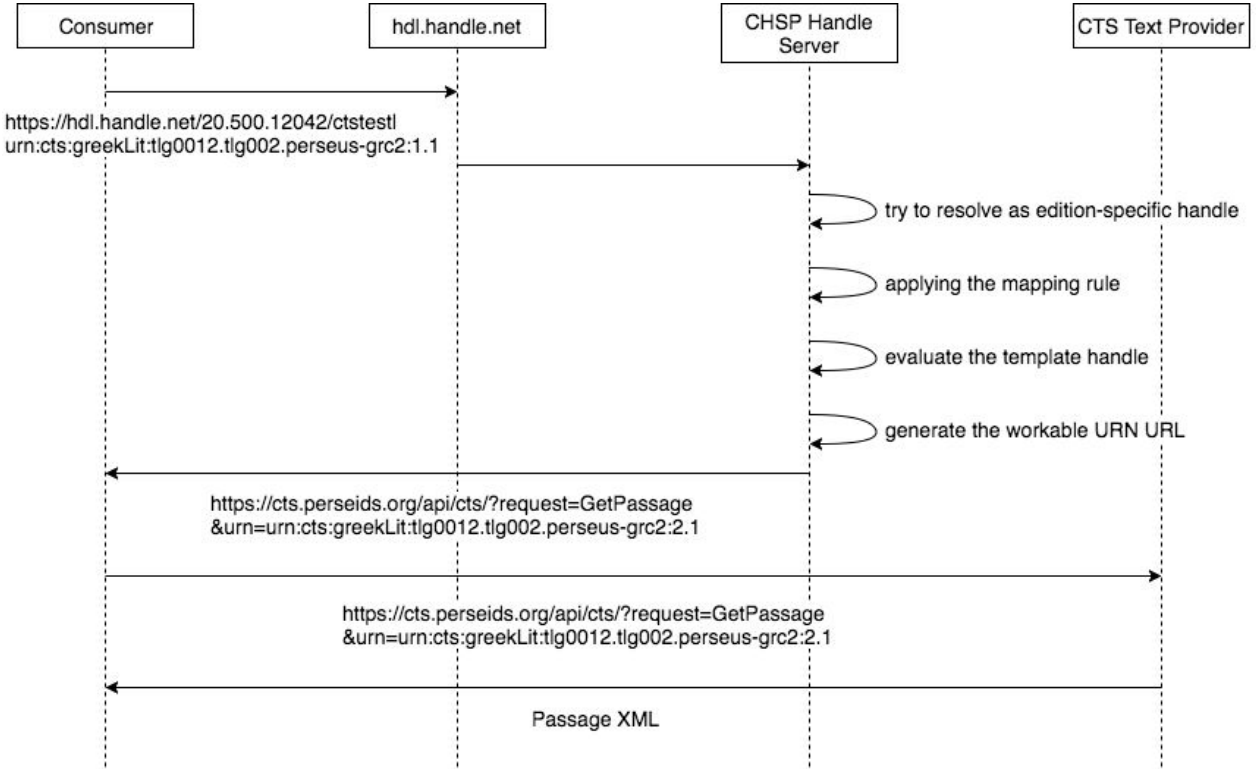
- Four Roles in CTS:
  - **Centralized Handle System Provider (CHSP)** - one or more organizations assuming responsibility for registering and administering Handle prefixes for CTS Namespaces
  - **Participating CTS Text Publisher (PCTP)** - a publisher of CTS URN identified texts who wants their text URNs to be globally resolved by the Handle System
  - **hdl.handle.net provider (HDL)** - the provider of the global hdl.handle.net proxy service
  - **Non-Participating CTS Text Publisher (NPCTP)** - a publisher of CTS URN identified texts who does not want to participate in the centralized solution but wants to publish a Handle for their text

Progress

- Template Handle and Mapping Rule
  - A single Template Handle can be created as a base that will allow any number of extensions to that base to be resolved as full handles, according to a pattern, without each such handle being individually registered.
  - Mapping rule provides a function to map the unregistered Handle into a workable URN URL.
- Sample Handle Request
  - https://hdl.handle.net/20.500.12042/ctstest|urn:cts:greekLit:tlg0012.tlg002.perseus-grc2:1.1
- Mapping result
  - https://cts.perseids.org/api/cts/?request=GetPassage&urn=urn:cts:greekLit:tlg0012.tlg002.perseus-grc2:2.1

Workflow

Sample use case



UAG Goals and Expectations

Robert McDonald

RPID Testbed UAG Goals and Expectations

- Members are expected to attend two meetings over the next year to give feedback to the RPID Team. These will be held in a zoom virtual environment at a convenient time for all US timezones. (Today is the first of these meetings).
- Members commit to emailing any feedback or issues while using the RPID tools to rpid-l@list.iu.edu
- Members commit to sharing their PID data interest or needs with the RPID Team.
- The RPID team is very interested in trying out new use cases so please let us know how we can help you get started...