



Fostering resource integration: EarthCube Resource Registry

Stephen M. Richard, LDEO, Columbia University and US Geoscience Information Network (USGIN); Ilya Zaslavsky, San Diego Supercomputer Center; and Ruth Duerr, Ronin Institute for Independent Scholarship



Richard, Stephen; Zaslavsky, Ilya; Duerr, Ruth (2019): Fostering Resource Integration: EarthCube Resource Registry. ESIP. Poster. <https://doi.org/10.6084/m9.figshare.8945093.v1>

What is the EarthCube Resource Registry?

A database of cyberinfrastructure resources with documentation focused on enabling EarthCube users and developers find, understand, get, and use those resources to increase research productivity

Resource focus: applications, reusable code components, ontologies, vocabularies, specifications (extending the GeoCODES dataset registry)

Why?

- Improve discovery of usable resources
- Enable seamless connection from discovery environment to working with the data or software.

Some usage scenarios:

- Identification of resources
- Systematic documentation of resource characteristics
- Identify gaps or duplicate resources
- Find tools, APIs, or data that can work with a given resource
- Support maturity assessment

The Registry should:

- Help researchers connect multiple data types and resources to address a specific research problem;
- Enable developers to learn about components they can reuse to increase development efficiency
- Enable discovery of components with a particular functionality, that can be used in an existing research workflow
- Provide a platform for resource producers to inform the community about products.
- Answer the question: “What has EC produced that is of use to my science?”

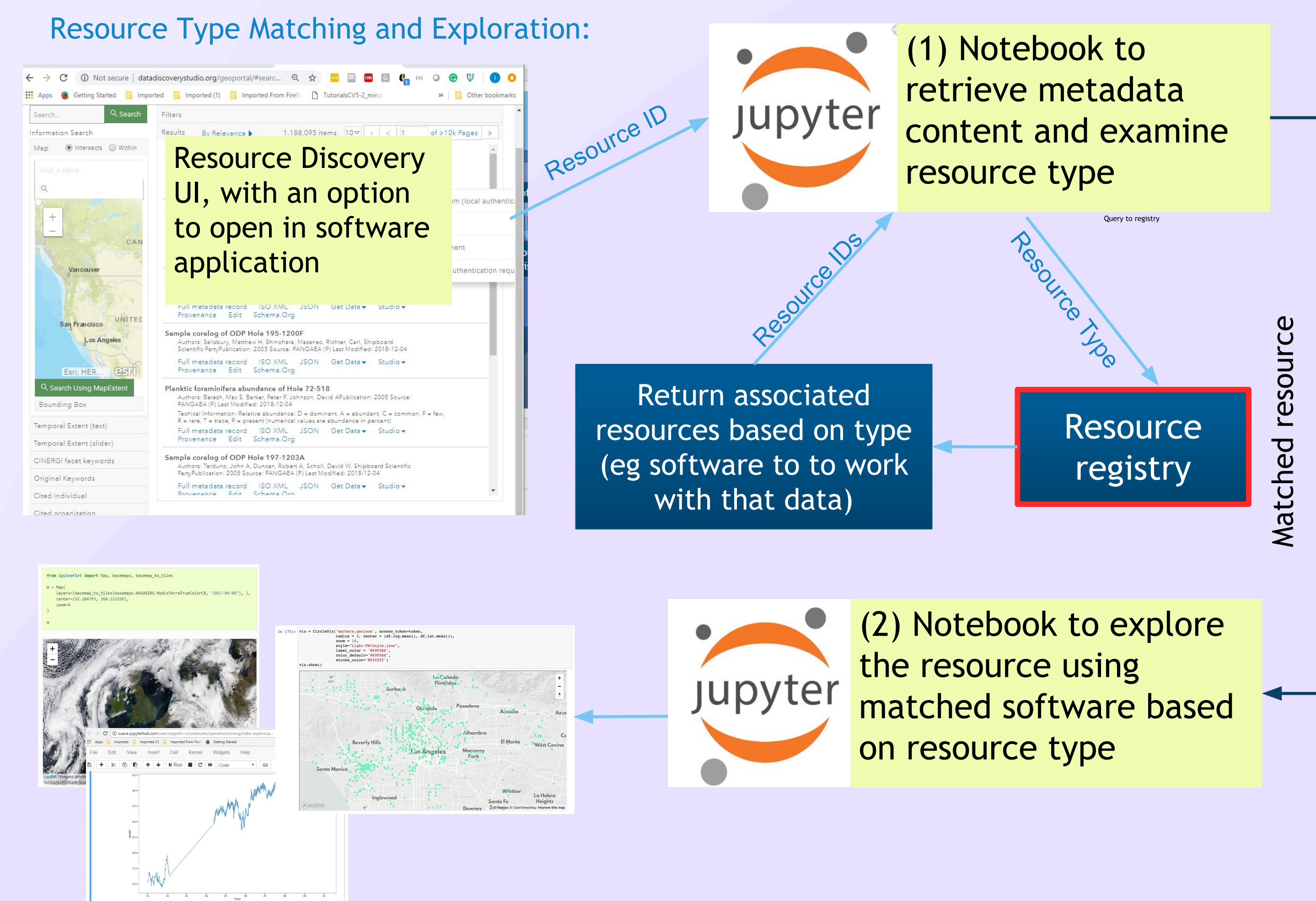
Links

- ✓ [SuAVE viewer](#) for EarthCube resources
- ✓ [Resource Registration Form](#) for submitting resources:
- ✓ [Viewer for adding information](#)
- ✓ [Edit view for modifying the form](#)
- ✓ [Resource Registry SPARQL endpoint](#) (username/password sent by email)
- ✓ [Documentation on mapping resource descriptions](#) to the registry ontology and JSON-LD serialization.
- ✓ [Controlled vocabularies Google Sheet](#) (one tab for each vocabulary)
- ✓ [EarthCube Resource Registry Presentation](#)
- ✓ <https://drive.google.com/file/d/1dwiUliAurFstWRSy74UBoUSaJYlVvc>



This work was supported by funding from the 2019 ESSO EarthCube Office and the 2020 ECO EarthCube office

Link data and applications



Resource Types	Bundled Object Catalog Dataset Document Information Model	Interchange Format Interface Platform Repository	Service Software Specification UseCase
----------------	---	--	--

Resource Registration Web Form

An [online form](#) generated using Google Forms to submit resource descriptions.

- a series of views
 - First views are for fields that apply to all resources.
 - User then identifies the kind of resource they are registering,
 - Subsequent views are specific to that resource.

Required fields are indicated with a red asterisk (*)

A submitted resource is assigned an ARK URI identifier,

Documentation is stored in the EarthCube registry, made available for browsing and discovery via the SuAVE viewer, and accessible via API.

First page of registration form

docs.google.com/forms/d/e/1FAIpQLSfyINCPv6Yck0lv2Ju7K3tL05jhiyUldLcNm67QwXn6kqQ/viewform

* Required

Section 1. General Resource Information
Resource name, description, identifier, Web page and citation

Resource Name *

Short name by which this resource will be recognized. Resource names should uniquely identify the resource for human users.

Your answer

ⓘ This is a required question

Alternate Resource Name(s)

Alternate names by which the resource might be known or discovered. If names are provided in a non-English language, please suffix a language identifier using the '@' delimiter and an ISO-639-1 two letter language code, e.g. Forme universelle binaire de l'OMM@fr. Separate entries with a pipe (|) character.

Your answer

URL to User-Readable Page

URL to access information about the resource on the Web. To provide a label for the URL (e.g. 'home...)

Example controlled pick list

License *

Note that a resource may be available under more than one license.

☐ Creative Commons Attribution (CC BY)

☐ Creative Commons Attribution ShareAlike (CC BY-SA)

☐ Creative Commons Attribution-NoDerivatives 1.0

☐ Creative Commons Attribution-NonCommercial (CC BY-NC)

☐ Creative Commons Attribution-NonCommercial-ShareAlike (CC BY-NC-SA)

☐ Creative Commons Attribution-NonCommercial-NoDerivatives (CC BY-NC-ND)

☐ Creative Commons Public Domain

☐ Creative Commons CC0 "No Rights Reserved"

☐ Creative Commons PDM "No Known Copyright"

☐ Common Development and Distribution License 1.0

☐ Eclipse Public License version 2.0

☐ MIT

☐ Apache

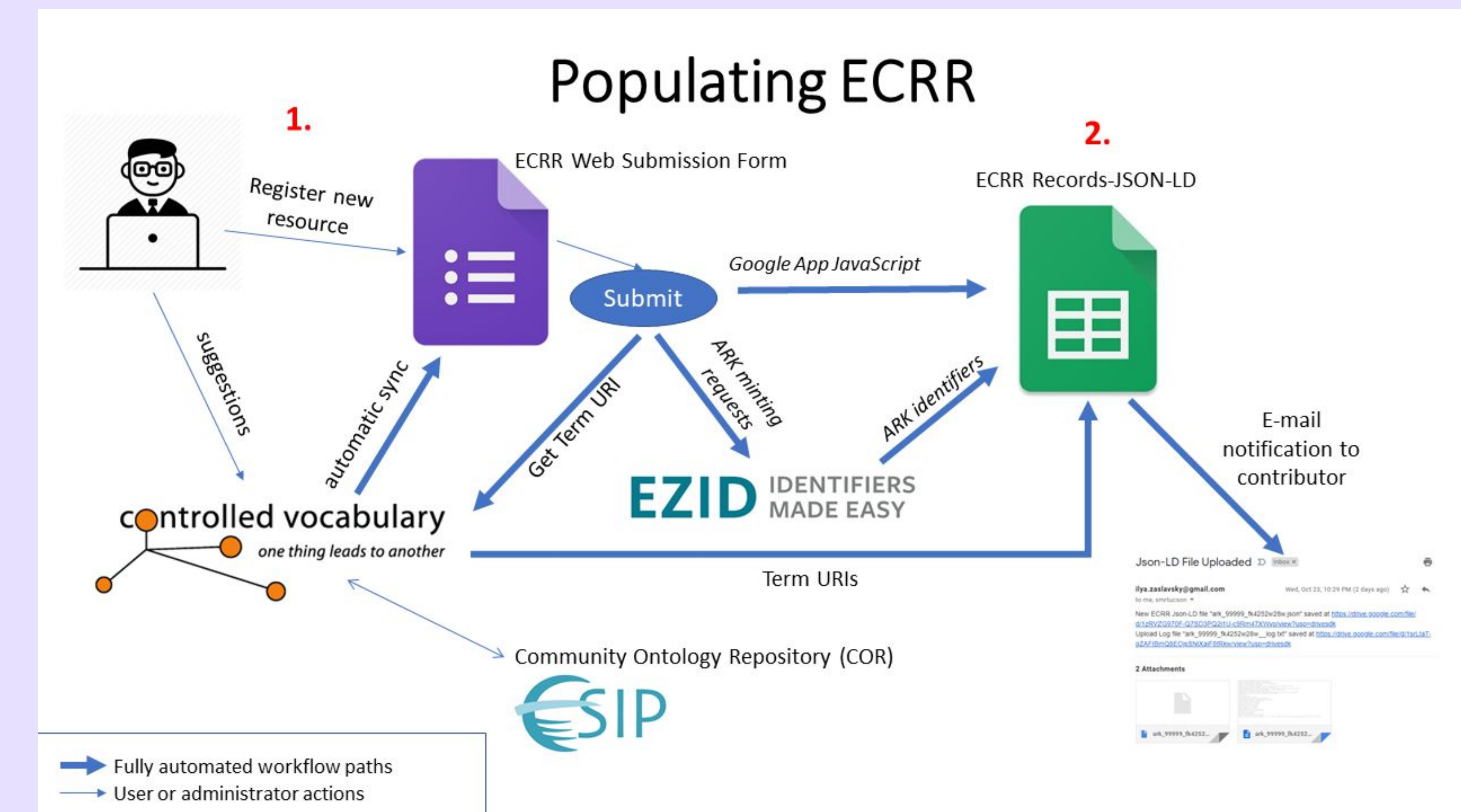
☐ BSD

☐ Mozilla Public License 2.0 (MPL-2.0)

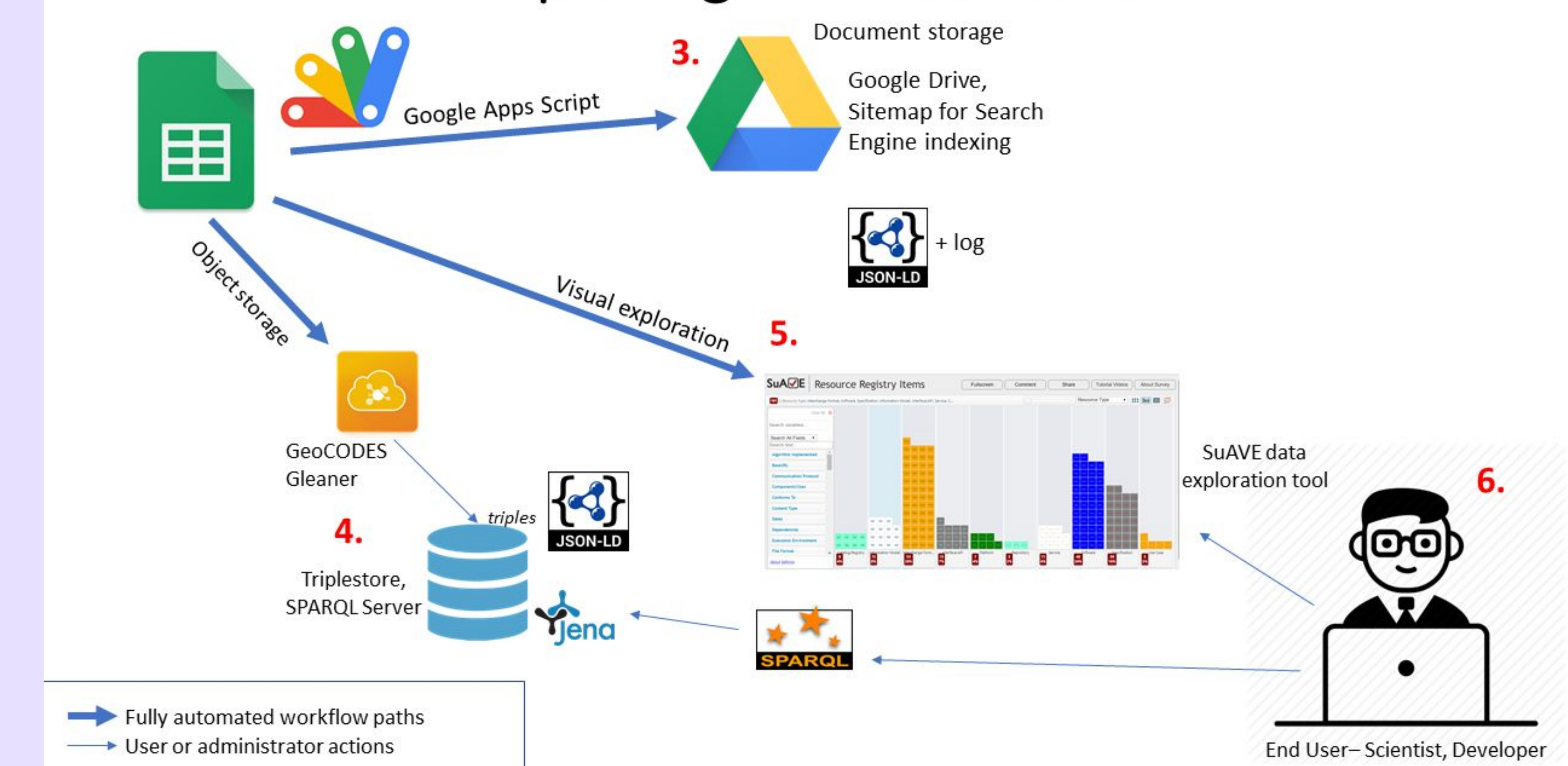
☐ GNU General Public License (GPL)

☐ GNU Lesser General Public License (LGPL)

Workflow



Exploring ECRR Content



Browse and Search registry content: SuAVE Viewer

SuAVE | ECRR 10 20 Responses

Search variables...

Search All Fields

SPARQL

Name

Search values...

Sort: A-Z

SPARQL XML Results For: 1

SPARQL JSON Results For: 1

SPARQL CSV/TSV Results: 1

SPARQL 1.1 Protocol sp: 1

SPARQL protocol v1.1 A: 1

EarthCube Resource Re: 1

Scientific Variables On: 1

14%

43%

14%

14%

14%

SPARQL XML Results Format

Timestamp

Fri Jun 26 2020 08:24:08 GMT-0700 (Pacific Daylight Time)

Name

SPARQL XML Results Format

http://www.w3.org/TR/rdf-sparql-XMLres/

Description

An XML format for the variable binding and boolean results formats provided by the SPARQL query language for RDF, developed by the W3C RDF Data Access Working Group (DAWG), part of the Semantic Web Activity

Target Audience

Informaticists

Technologists

Usage

Unknown

Science Domain

Discipline-agnostic

Type of Resource

Interchange file format

Example search for resources related to SPARQL



Fostering resource integration: EarthCube Resource Registry

Links to detail slides



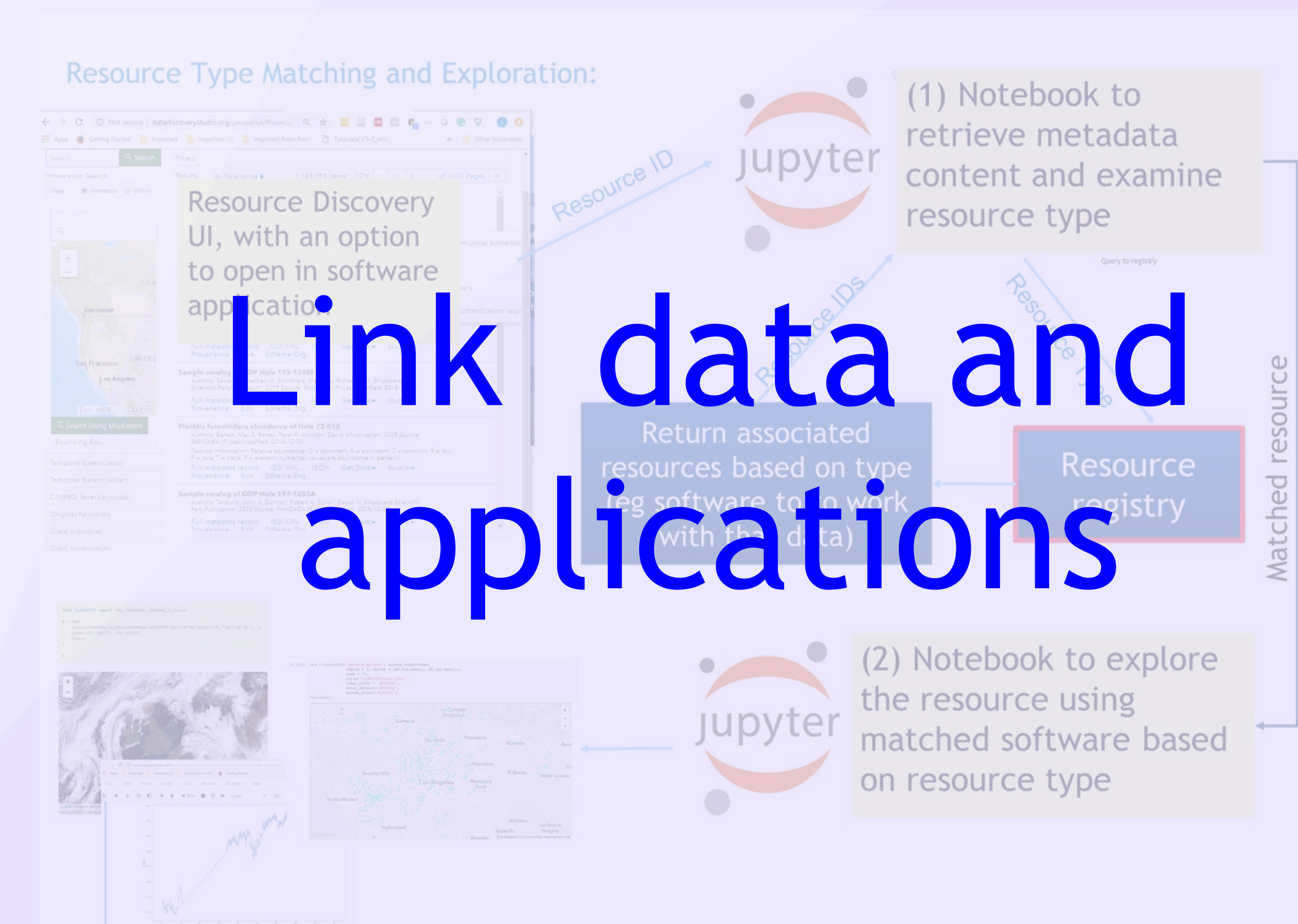
Resource Registry Implementation

- Modular: system components (e.g. controlled vocabularies, form validations) can be independently updated, with minimal (if at all) configuration
- Easy to deploy and use: no additional software or installation required, built from standard G-Suite components, management can be shared with other Google accounts (input forms, data, code)
- All records receive persistent identifiers managed via EarthCube EZID account (setup in coordination with ECO)
- Ontologies managed in the ESI Community Ontology Repository
- Easy to extend with Google APIs

Resource Registry Vocabularies:

15 controlled vocabularies were developed (see list below) to support the EarthCube Resource Registry Ontology; these are used to populate ECRR submission form dropdown lists. With the exception of the Academic Disciplines Ontology, all are [publicly available](#) under CC0.

- Academic Disciplines Ontology
- Audience Types Controlled Vocabulary
- Communication Protocols Controlled Vocabulary
- Creative Commons Licenses Ontology
- Data Licenses Ontology
- Data Model Language Controlled Vocabulary
- Expected Lifetime Controlled Vocabulary
- Funding Agency Controlled Vocabulary
- Fundref Funding Agencies
- Maturity Controlled Vocabulary
- Runtime Environments Ontology
- Semantic Resource Types Vocabulary
- Software Functions Ontology
- Software Licenses Ontology
- Technical Specifications Controlled Vocabulary
- Usage Volume Controlled Vocabulary



Link data and applications

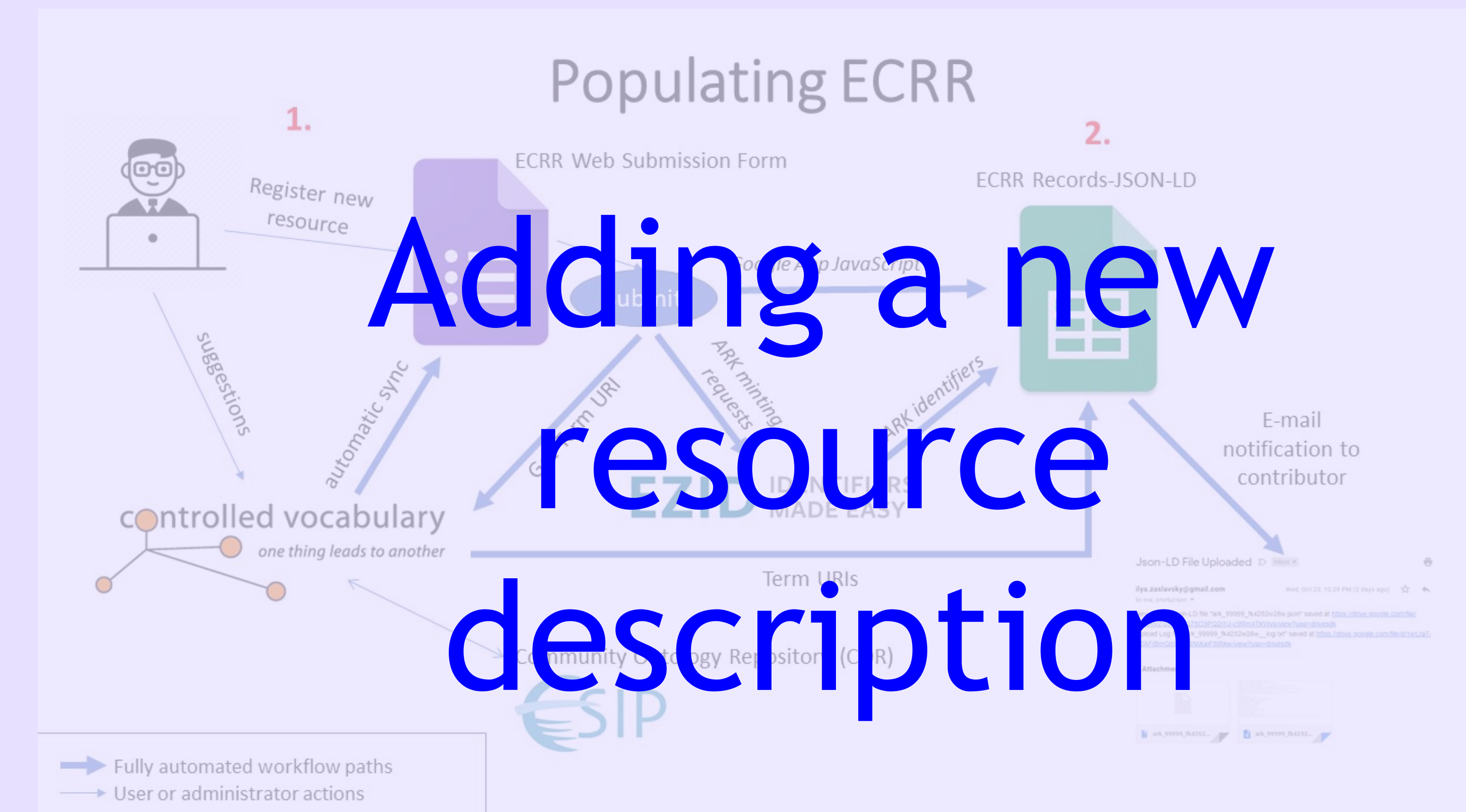
Resource Types

Example JSON-LD Resource Description

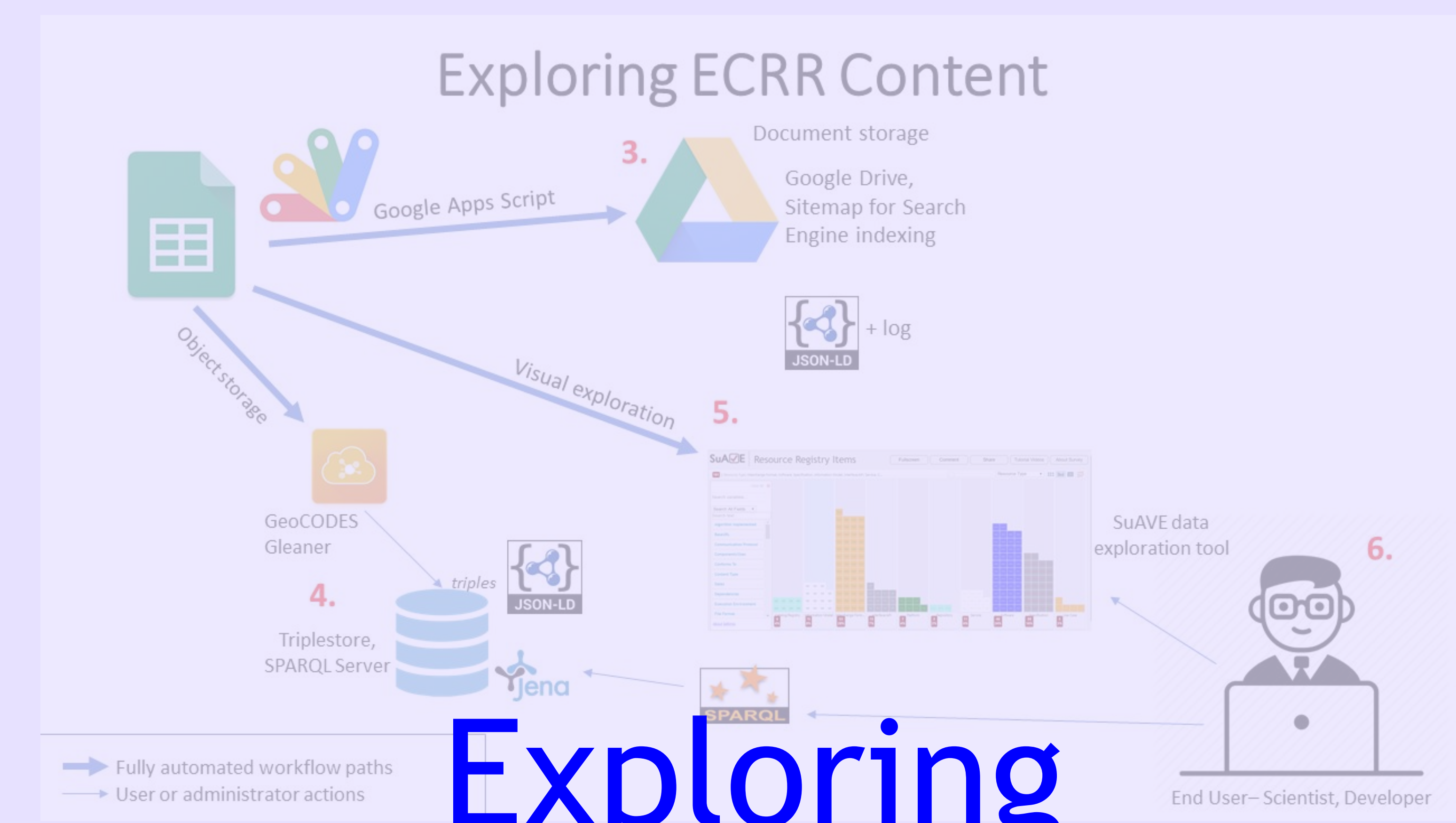
Other Links

- ✓ [SuAVE viewer](#) for EarthCube resources
- ✓ Resource Registration Form for submitting resources:
[Viewer for adding information](#)
[Edit view for modifying the form](#)
- ✓ [Resource Registry SPARQL endpoint](#) (username/password sent by email)
- ✓ [Documentation on mapping resource descriptions](#) to the registry ontology and JSON-LD serialization.
- ✓ [Controlled vocabularies Google Sheet](#) (one tab for each vocabulary)
- ✓ [EarthCube Resource Registry Presentation](#)
[- https://drive.google.com/file/d/1dwUliAurfFstWRSiY74UBoUSaJYIVvc](https://drive.google.com/file/d/1dwUliAurfFstWRSiY74UBoUSaJYIVvc)

Workflow



Adding a new resource description



Exploring Content

Browse and search registry content: SuAVE viewer



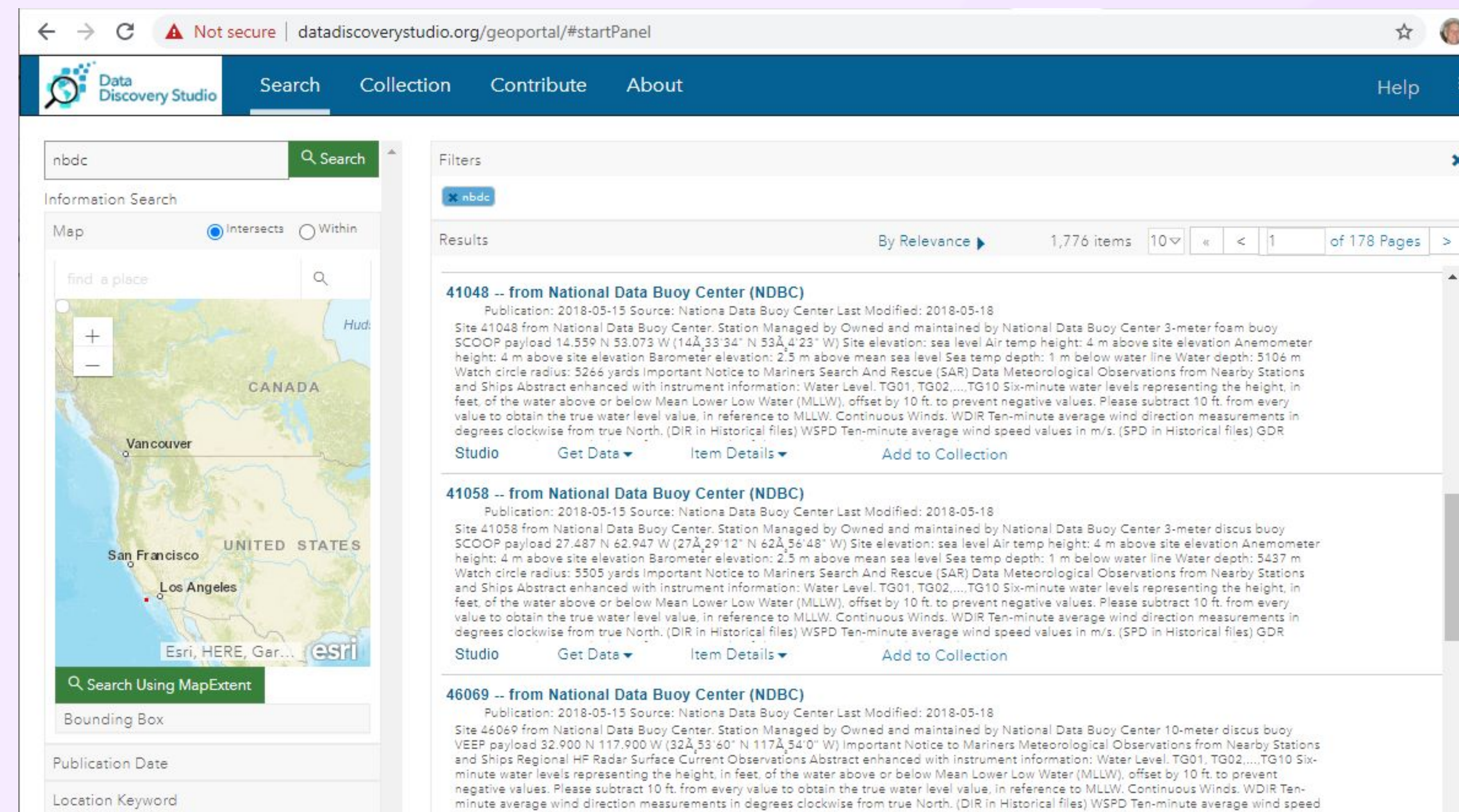
Example search for resources related to SPARQL



Resource Type Matching and Exploration:

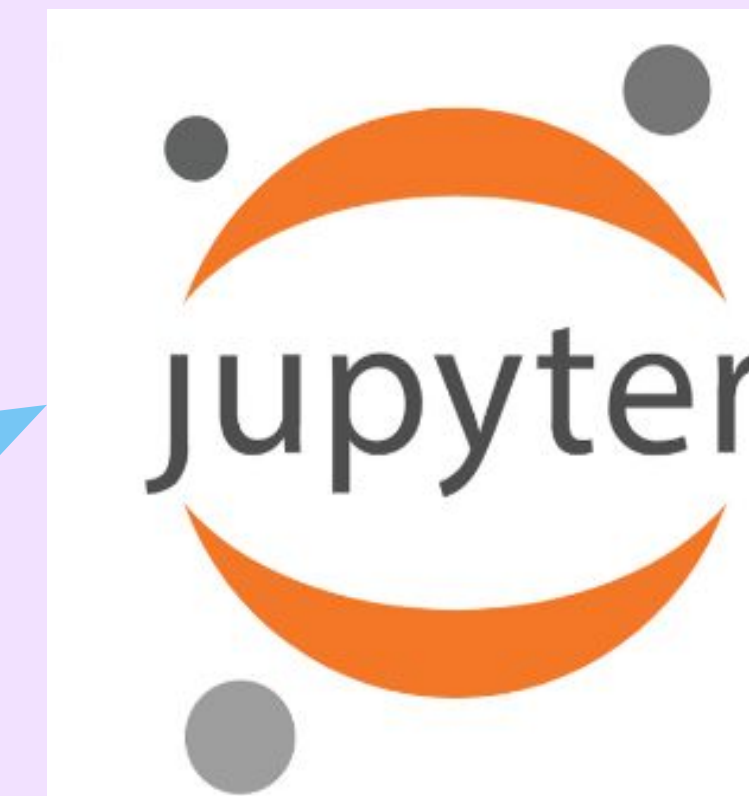
Resource Discovery UI Example (EarthCube DataDiscoveryStudio) with an option to open in software application. If you want to try this yourself, instructions are included in the blue boxes as you move through this slide

To run this example,
1. go to <http://datadiscoverystudio.org/> and type 'nbdc' in the search box. This will bring up data from ocean buoys monitoring a variety of environmental conditions.
2. Click on the 'Studio' link below a search result. A dialog will open to select a Jupyter notebook host. Select 'MyBinder-Production' (other options require login or local Jupyter notebook setup...).



Selected Resource ID

Compatible Resource IDs



(1) Notebook to retrieve metadata content and examine resource type

Query to registry: the resource registry is intended to provide linkage between resource types (e.g. NBDc text data format) and software resources (e.g. our POC Jupyter Notebook) that operate on that data type.

Resource Type

To run this example (continued)
3. MyBinder will start-- takes a while; once the dispatcher notebook is running, run the cells one at a time; you can skip 2a. The code gets the metadata record for the selected dataset, and inspects the distributions for the data. This proof of concept notebook is hardwired for NBDc or NWIS data.
4. When you get to cell 3 in the notebook, there should be a pick list to select a target notebook; select 'NBDc Explore'; run cell 4, and click on the URL that appears under the cel.

Return associated resources based on type (eg software to to work with that data)

Query

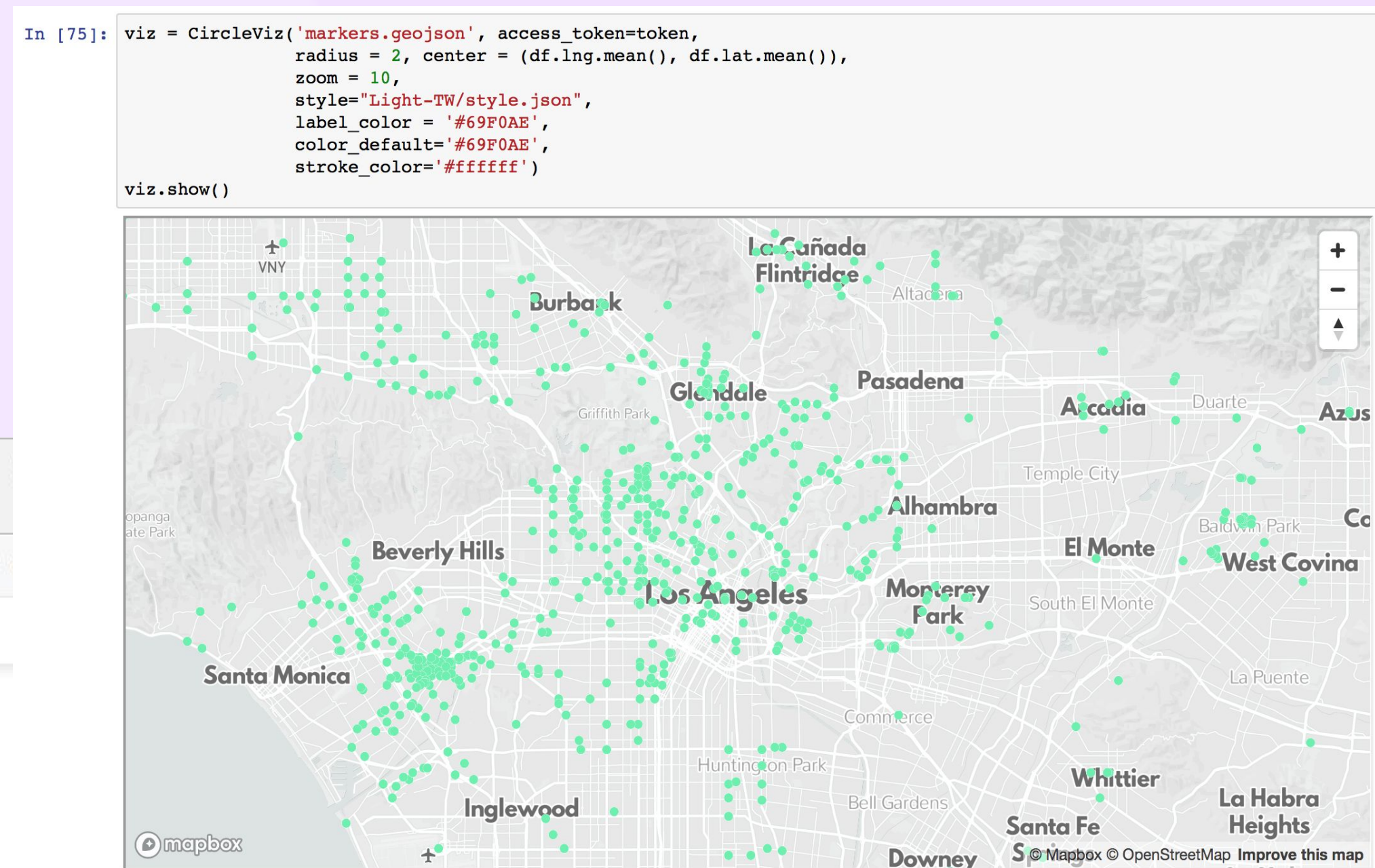
Resource registry

Matched resource

```
from ipyleaflet import Map, basemaps, basemap_to_tiles

m = Map(
    layers=(basemap_to_tiles(basemaps.NASAGIBS.ModisTerraTrueColorCR, "2017-04-08"), ),
    center=(52.204793, 360.121558),
    zoom=4
)

m
```



(2) Notebook to explore the resource using matched software based on resource type

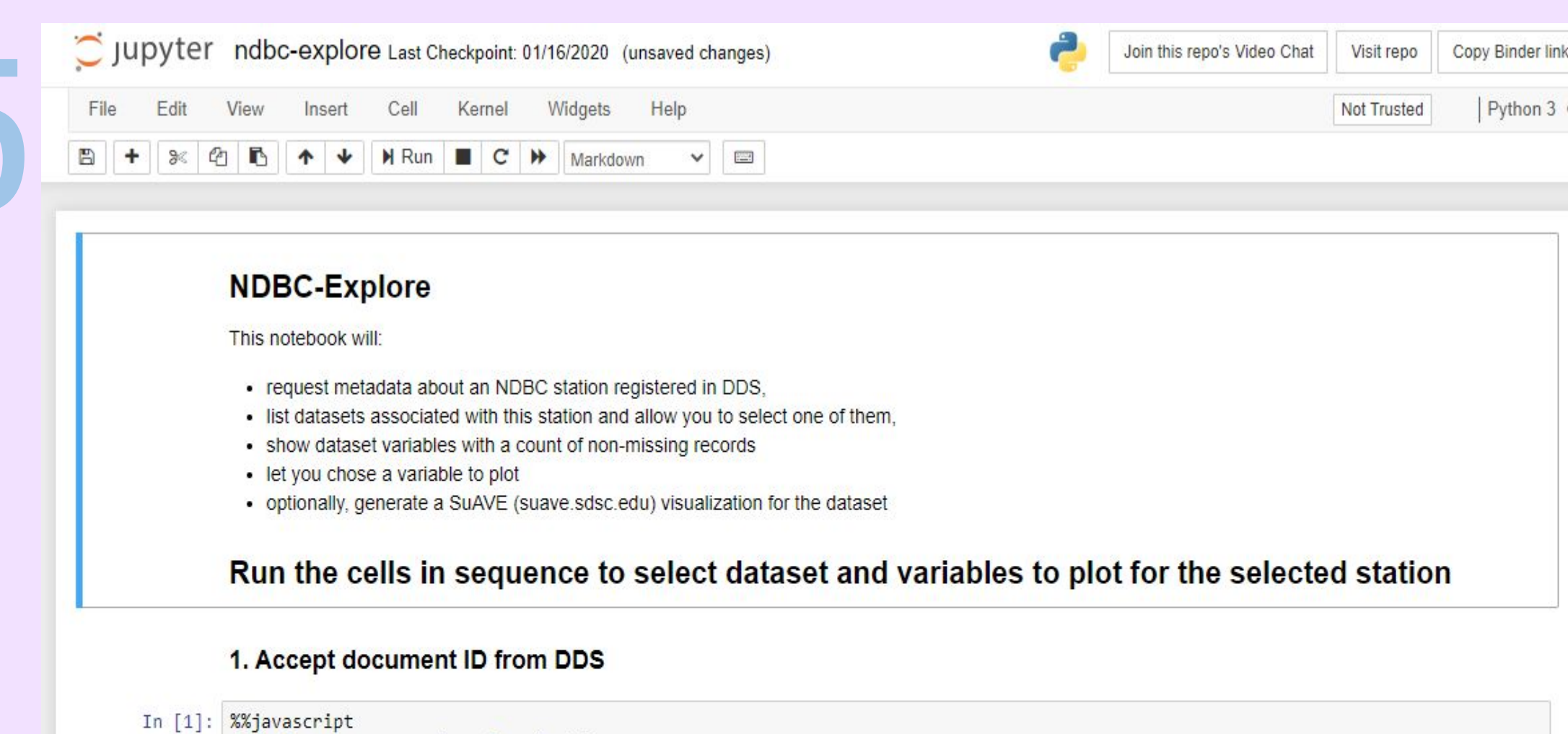
To run this example (continued)

5. This will open a notebook to explore the data

6. Run Cells 1 and 2. After running Cell 2, a pick list should appear below cell 2 to select which dataset in a series you wish to explore. Pick a dataset, then run cell 3 and 4

7. After running cell 3, a summary of the number of observations for the variables in that data will appear. After running cell 4, a pick list to select a variable to plot will appear. Select a variable and a plot of that time series will appear. Re-run cell 4 to plot a different variable

5



6

```
3. Get the selected dataset and prepare it for mapping; get measured (not missing) values

In [5]: chosen_url = data_dropdown_options[widget.result]
data = helpers.create_data_from_urls([chosen_url])

# Create Key Values for graph.
```

7

```
4. Select a variable for charting. Re-run this cell to chart other variables.

In [7]: # Example plot of the image based on values of WTMP (Can be changed to other var
# rerun this cell if need to plot other variables

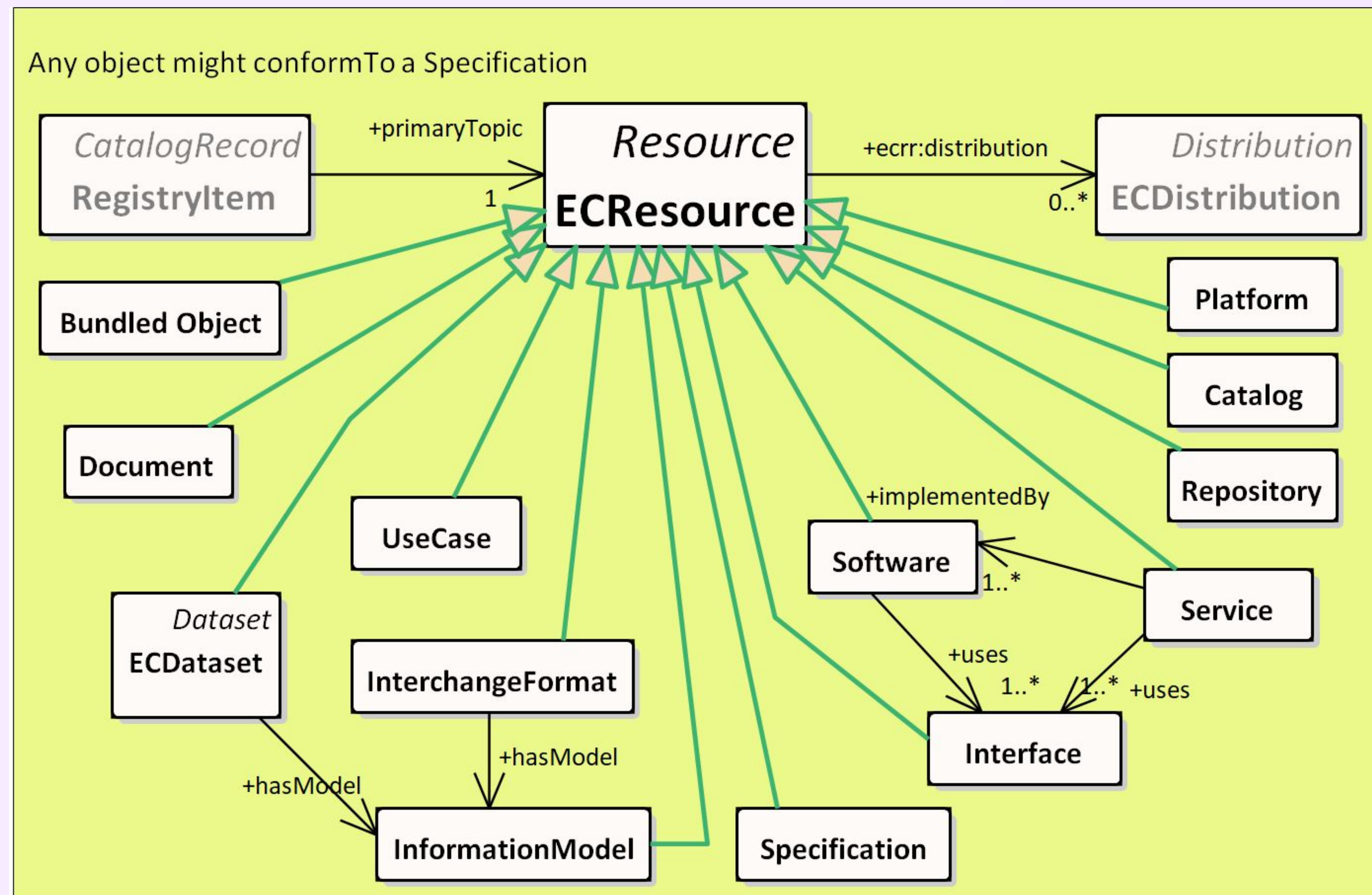
#Create widget with dropdown options from list created above
w = widgets.Dropdown(
    options=dropdown_options,
    not_all_code_show...

pylab.rcParams['figure.figsize'] = (25, 15)
helpers.plotTimeSeriesData(data, change['new'], graphKeyVals)

w.observe(on_change)
display(w)

Number: Choose Variable
```


Resource types in the Registry



Catalog

A curated collection of descriptions of resources, accessible through one or more interfaces. Technically a catalog should be a dcat bundled dataset (the metadata collection) and one or more sdo:Services (i.e., the access interfaces). Implicitly there is also a stewardship/policy component determining management of the catalog.

Interchange Format

A serialization scheme (file format) that implements some information model (schema and vocabulary) to communicate information between agents.

Most general types are generic formats specified by typical MIME types, but this resource type includes much more granular identification of schemes/Data Types/Data Specification that are specific to particular applications. These generally implement some underlying information model (schema and vocabulary) that enables semantic as well as syntactic interoperability. An Interchange Format SHOULD have an associated specification that documents its provisions and usage. Registered interchange formats are intended to provide identifiers for data serialization formats that work with specific applications. Links to registered formats would be via URI values for the inputFormat and outputFormat properties on an Interface/API, Software, or Service resource.

In situations where no specification resource is available to document the format, it is sufficient to simply identify the format as an application specific format. This is the case for various commercial applications that use proprietary formats, as well as for scientific model input and output formats. The interchange format can be linked to an information model that defines the entities and properties represented using the format.

Bundled Object

A composite resource that is a collection of other resources assembled into a package for some purpose, and accessible as a single Unit. Examples: Docker containers; Bagit, Data on the Web, or ORE packages with heterogeneous content.

Semantic Resource (Information Model)

A specification of concepts, and optionally relationships, representing a conceptualization of some domain of discourse. Includes a spectrum of resources including ontology, controlled vocabulary, thesaurus, information model, taxonomy, glossary, vocabulary..... For linked data purposes, it is useful to specify the entities and properties of the entities represented by the model using URI references to widely used ontologies. For discovery purposes in the registry, the entities and properties should be described in English language terms.

Interface (API)

Specification of a set of operations, messages used to invoke the operations, and inputs necessary to execute the operations, along with an expected output content and format. An Interface/API should have an associated specification document.

Platform

A composite software entity that enables execution of a variety of tools e.g. MatLab, ArcGIS. The key defining feature is extensibility - with plugin architecture and API-based interaction between components to enable reconfiguration/extension for new problems.

Repository

A storage system in which objects may be stored for subsequent access or retrieval (generalize from Kahn and Wilensky, 1995, <http://www.cnri.reston.va.us/k-w.html>). A place where things may be put for safekeeping.

Service Instance

A software entity on one side of an interface that executes some kind of processing in response to a request made by an agent on the other side of the interface. The service presents a set of offerings, each of which entails a resource and one or more operations on that resource. Each offering also specifies the interchange format for input and output, and the syntax for invoking the operations, including any parameters in the operation signature. A service instance description should include a link to specification(s) for the API that the service implements.

Software

A packaged set of instructions that can be executed by a machine to perform one or more functions. Distribution might be as source code, an executable file, an installer package, or as a web application that can be invoked via URL. Software descriptions often reference specifications for algorithms and interfaces that are implemented, as well as interchange formats used for input and output.

Specification

A document that describes the technical characteristics of an artifact or practice, possibly including a description of what it should do, or an explicit set of requirements that it must satisfy. All resources may have one or more conformsTo properties that should link to a specification. Specifications may be hierarchical, e.g. a profile specification might inherit provisions from a parent specification, and add additional constraints to further restrict usage of the artifact or practice.

UseCase

A specification of a scenario for a work item with some specific context and goal.

Example resource description

JSON-LD Context. Elements with no prefix are schema.org. ecrrro are elements defined by the [EarthCube Resource Registry Ontology](#). The ontology and and controlled vocabularies to populate elements defined in the ontology are registered in the ESIP Community Ontology Registry (COR).

The identifier is an Archival Resource Key (ARK) identifier, assigned through the California Digital Library. EarthCube has a registered Name Assigning Authority Number (NAAN, 23942) in the ARK system.

ecrrro; namespace elements are identified by opaque, numeric URIs, defined in the ESIP CORs registry.

schema:DefinedTerm references concepts registered in the ESIP COR. The term label is included as an schema:name in the JSON-LD

```
{
  "@context": {
    "@vocab": "http://schema.org/",
    "dcat": "http://www.w3.org/ns/dcat#",
    "ecrrro": "http://cor.esipfed.org/ont/earthcube/ECRRO_",
    "dc": "http://purl.org/dc/terms/"
  },
  "@id": "https://n2t.net/ark:/23942/g2500001",
  "@type": [
    "CreativeWork",
    "Product"
  ],
  "name": "Data Discovery Studio",
  "subjectOf": {
    "@type": "CreativeWork",
    "name": "user readable web page",
    "URL": "https://www.earthcube.org/group/earthcube-data-discovery-hub"
  },
  "description": "Aggregating catalog, harvests metadata for geoscience resources from 36 sources, processes through records utilize other schemes.",
  "dc:BibliographicCitation": "",
  "license": {
    "@type": "CreativeWork",
    "name": "Public Domain Dedication",
    "identifier": "https://creativecommons.org/publicdomain/zero/1.0/"
  },
  "ecrrro:0000138": {
    "@type": "DefinedTerm",
    "name": "In production",
    "identifier": "http://cor.esipfed.org/ont/earthcube/MTU_0000002"
  },
  "http://cor.esipfed.org/ont/earthcube/ELT_0000003": {
    "ecrrro:0000219": {
      "@type": "DefinedTerm",
      "name": "1 - 5 years",
      "identifier": "http://cor.esipfed.org/ont/earthcube/ELT_0000003"
    },
    "funder": {
      "@type": "Organization",
      "name": "US National Science Foundation (NSF)",
      "identifier": "https://ror.org/021nxhr62"
    }
  },
  "ecrrro:0000219 label is 'expectedLifetime'"
}
```

ecrrro:0000219 label is 'stewarded by'

ecrrro:0000017 label is 'has usage'

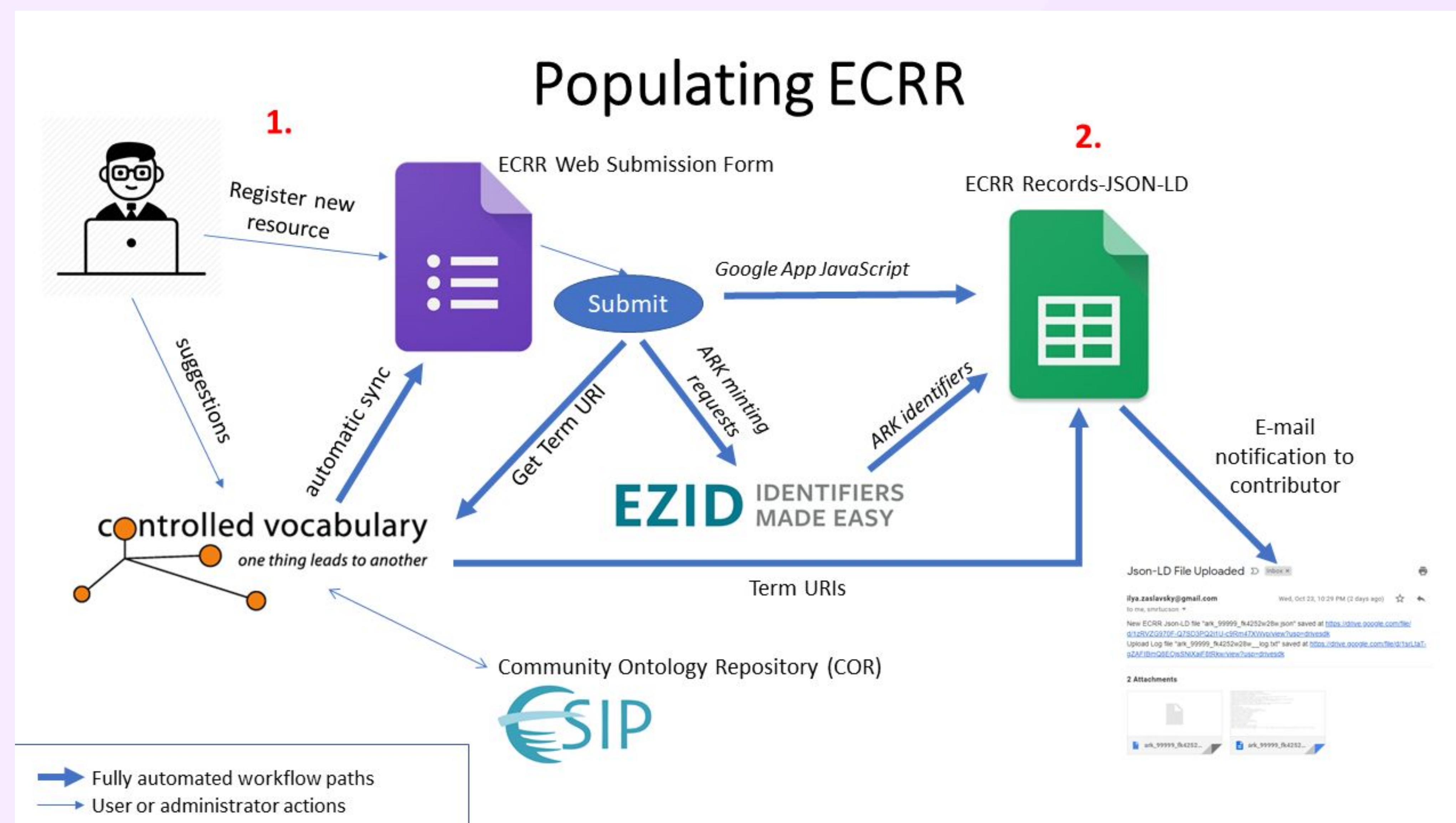
http://purl.obolibrary.org/obo/RO_0002502 is term from obo ontology; label is 'depends on'

ecrrro:0001301 label is 'registrationMetadata'

ecrrro:0000213 label is 'EC Catalog Record'; This is the identifier for the metadata record, distinct from the identifier for the described resource

```
"ecrrro:0000218": {
  "@type": "Organization",
  "name": "US National Science Foundation (NSF) projects"
},
"audience": {
  "@type": "Audience",
  "audienceType": "Scientists",
  "identifier": "http://cor.esipfed.org/ont/earthcube/AUT_0000007"
},
"ecrrro:0000017": {
  "@type": "DefinedTerm",
  "name": "Low usage",
  "identifier": "http://cor.esipfed.org/ont/earthcube/UBA_0000003"
},
"about": [
  {
    "@type": "Thing",
    "name": "Earth Science",
    "identifier": "http://cor.esipfed.org/ont/earthcube/ADO/ADO__0000021"
  },
  {
    "@type": "Thing",
    "name": "Dataset",
    "identifier": "undefined"
  }
],
"http://purl.obolibrary.org/obo/RO_0002502": {
  "@type": "CreativeWork",
  "name": "ESRI Geoportal v2.6",
  "url": "ISO19115"
},
"isRelatedTo": {
  "@type": "Product",
  "name": "SciGraph"
},
"additionalType": "http://cor.esipfed.org/ont/earthcube/ECRRO_0000212",
"ecrrro:0001301": {
  "@type": "ecrrro:0000213",
  "@id": "ark:/23942/g2v33z",
  "creator": {
    "@type": "Person",
    "name": "Stephen M. Richard",
    "identifier": "https://orcid.org/0000-0001-6041-5302",
    "email": "not provided"
  },
  "datePublished": "October 29, 2019",
  "dateModified": "October 29, 2019"
}
}
```


Adding a new resource description



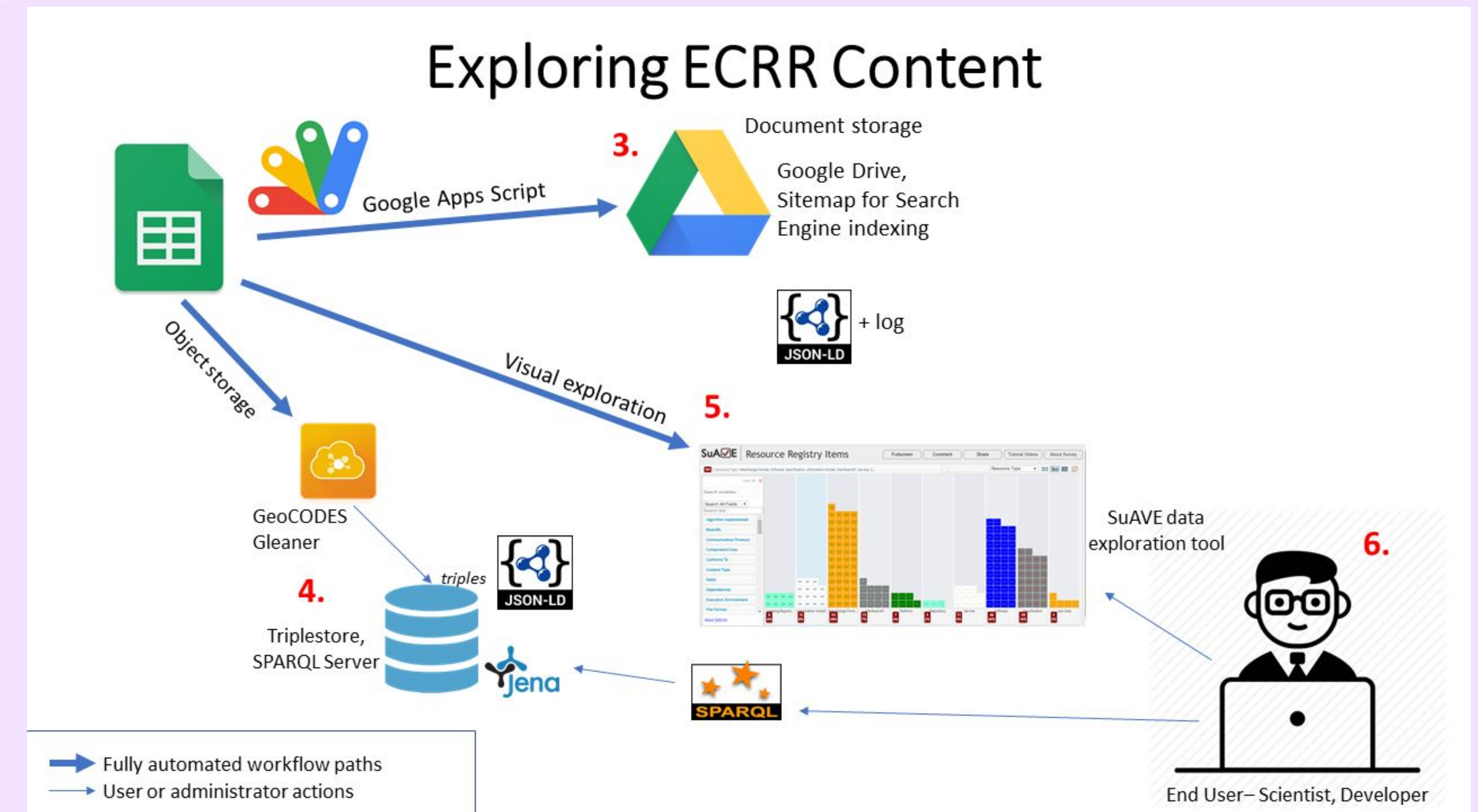
- 1) User submits a form at <https://docs.google.com/forms/d/e/1FAIpQLSfyINCpV6Yck0lv2Ju7K3fLLO5jhiyUlIdLcvNm67QwXn6kqQ/viewform>
 - a) The form includes questions organized in 15 sections: 1) General Resource Information; 2) Resource Availability and Stewardship; 3) Resource Audience and Usage; 4) Related Resources and Dependencies; 5) Resource Type; 6-14) Resource type-specific questions (The form branches to pages for specific resource types); 15) Information about the form submitter.
 - b) The form responses include free text, dates, or terms from a controlled vocabulary. Fields populated from a controlled vocabulary present a pick list of valid terms from the controlled vocabulary; vocabularies are managed separately (see below) and synchronized with the form as necessary.
 - c) While working through the form, users can browse controlled vocabularies, view examples of previously entered resource descriptions, or visualize the resource registry in a user interface. From the form landing page, they can submit issues to the ECRR github to report problems, bugs, and suggestions.
 - d) Controlled vocabularies are assembled in a single Google spreadsheet (<https://docs.google.com/spreadsheets/d/1ykxgdeDjBzcTc1y64CuyaS2PhDuUtLuO5AFUehwhxDS/edit#gid=0>) that is accessed by the submission form. Users can suggest controlled vocabulary changes from any sheet in this spreadsheet, by clicking the "Suggest CV Changes" button and providing a suggested term, definition, and source.
 - e) Authoritative versions of the controlled vocabularies are registered in the ESIP Community Ontology Repository (CORS) for version control, community access, and persistence. The controlled vocabularies in the Google spreadsheet are synchronized with the vocabulary managed in the ESIP Community Ontology Repository as necessary.
 - f) Once a user enters a resource description, she has an option to revise and resubmit the entry.

2. Once the resource description form is submitted by the user, several operations are automatically triggered and executed. These are implemented by Google Apps Script (a dialect of Javascript) attached to the target Google spreadsheet where the form writes responses. The operations are executed in the following sequence:
 - a) A request to mint an ARK identifier is sent to the EZID identifier management system (<https://ezid.cdlib.org/>) at the California Digital Library. ARK (Archival Resource Key) persistent identifiers are described at (https://n2t.net/e/ark_ids.html) and have been used by 500+ organizations since 2001. Working with the new EarthCube Office at UCSD, we set up an EarthCube EZID account that allows EarthCube to generate DOI and ARK identifiers. EZID identifier management can be accessed from the workflow in two ways:
 - i) Minting new ARKs directly from a Google Apps Script (this is the default operation)
 - ii) Managing ARKs via an EZID proxy we developed (for operations that require ARK IDs as part of the request: this was used to generate ARKs for pre-built registry records). The proxy is running at https://maxim.ucsd.edu/ARK_ID.php. It accepts requests in the form https://maxim.ucsd.edu/ARK_ID.php?operation=<operation>, where <operation> can be one of 'get', 'create', 'mint', 'update', 'crupdate', or 'delete', along with the name:value pairs in the POST payload. The name:value pairs are consistent with EZID documentation at <https://ezid.cdlib.org/doc/apidoc.html#metadata-profiles>. Depending on the operation, they may include:
 - (1) Ark_Id (the fully qualified ARK Identifier to be created or retrieved);
 - (2) userid;
 - (3) password;
 - (4) shoulder (as described in ARK documentation);
 - (5) metadata (a multiline string containing the metadata.)
 - b) Using the information entered through the form, and the generated ARK identifier, the system generates a JSON-LD file and a JSON-LD creation log file. In the process of generating the JSON-LD, the system retrieves URIs for controlled vocabulary terms from the spreadsheet of controlled vocabularies to match user-selected terms. The URI's are included in the JSON-LD document to provide linked data connectivity with term definitions in the ESIP COR..
 - c) The JSON-LD document and a log file are generated and automatically emailed to the system administrator. Optionally, we can email these files to the user submitting the form (note: email quotas for Google services are 100 email recipients per day for a standard Gmail account and 1500 email recipients per day for a G-Suite Education account. Moving to G-Suite will let us be more liberal with email notifications. See <https://developers.google.com/apps-script/guides/services/quotas> for details.) Additional email notifications to the administrator include the content of individual submissions, and of the count of recently submitted records.

Exploring Content

The JSON-LD document and the associated log file are used as follows:

- Both files are automatically saved to an “ECRR Resources” directory on Google drive.
- The JSON-LD file is added to an object store, for subsequent conversion to triples using GeoCODES’ Gleaner routine (the Gleaner component has to be done manually) and loading into an Apache Jena Fuseki triple store and SPARQL server. This process currently involves several manual steps, and will need to be refined and automated for a production system, but the current process is described here. The JSON-LD files generated from user submissions are placed in a Google Drive that is synchronized with a cloud object store in the UCSD cloud environment. The synchronization is done by an hourly cron job. The p418 Gleaner has been modified to access the UCSD cloud object store; it generates a set of rdf triples from the JSON-LD documents. After validation of the triples, they are saved back to the UCSD cloud object store in a new file in N3 format. The triples are then manually loaded into the triple store using the ‘upload file’ function in the Jena Fuseki user interface.
- The ECRR workflow automatically updates a Google sheet that feeds into a user interface for browsing and searching registry records. The user interface is built using a new version of SuAVE (<http://suave.sdsc.edu>), which supports various modes of visual exploratory analysis and integrates with Jupyter notebooks. The new version extended SuAVE to additional sources of data including Google sheets. The SuAVE view of the registry is accessed at <https://tinyurl.com/y485a9rm>



User interaction

- Several modes of user interaction are enabled. The [SuAVE web application](#) enables full-text and faceted search of ECRR content; lets users visualize distribution patterns based on any of the resource properties, e.g. by resource type, maturity, usage level, or science domain; and zoom in to view complete metadata for a specific resource. In addition, it allows users to annotate resource descriptions, and share ECRR data views with others.

Registry records can be queried via SPARQL through the Apache Jena Fuseki interface, and can be accessed via URL from the Google Drive. The ark identifiers will be bound to a URL that will access the JSON-LD representation of the resource description. The URL pattern to access the JSON-LD for a resource is:

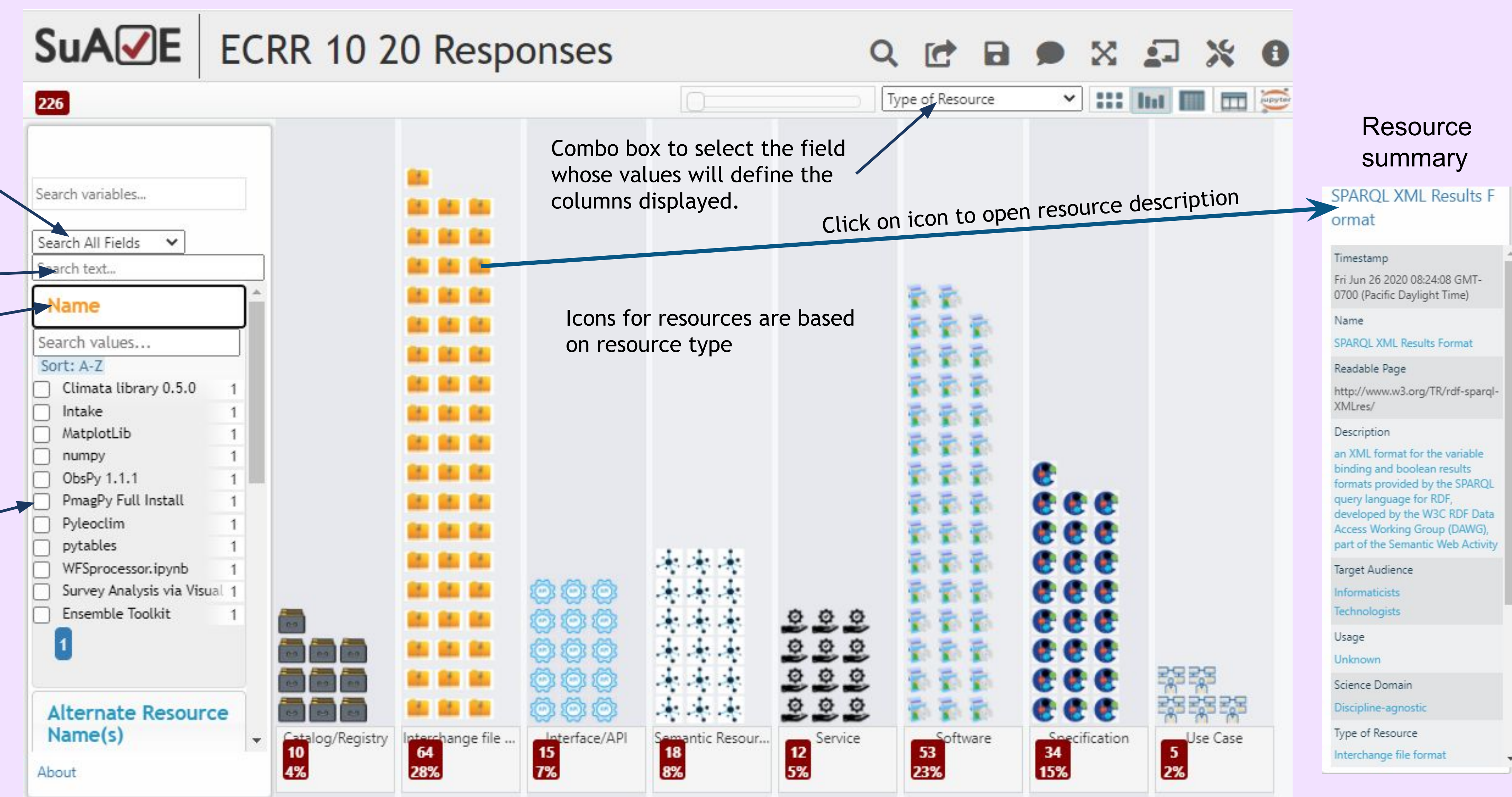
https://object.cloud.sdsc.edu/v1/AUTH_85f46aa78936477d8e71b186269414e8/gleaner-summoned/ecrr/{clean ark id}

Combo box to select fields to search, choices are {All Fields, Description, Citation, Primary Publication}

Search text goes here

Select field to see facets.

Individual values from the selected field and the frequency count for that value. Clicking the check box will display resources that have that value.



red boxes show hit count in that column, and fraction of total hits.