How to Build a Data Mesh Using Gen3

Center for Translational Data Science (CTDS), University of Chicago

Department of Public Health Sciences, Biostatistics Laboratory, University of Chicago

The Agenda

- A Quick Introduction to Gen3 Data Meshes - Bob Grossman, CTDS, UChicago
- HEAL Data Mesh - Phil Schumm, Department of Public Health Sciences, UChicago
- Biomedical Research Hub (BRH) Data Mesh - Aarti Venkat, CTDS, UChicago
- How to Set Up a Gen3 Data Mesh - Sai Narumanchi, CTDS, UChicago
- Open Discussion
- Discussion on topics for next event

A Quick Introduction to Gen3 Data Meshes

Robert Grossman

THE UNIVERSITY OF CHICAGO
Center for Translational Data Science

GEN3

Open Commons Consortium

1. Why data meshes (aka data ecosystems)?

GEN3 DATA COMMONS

GEN3

1,696,343 Total Subjects | 91,676,537 Total Files | 17.22 PB Total File Size

Data Meshes

NIH HEAL INITIATIVE

The HEAL Data Platform enables search and discovery across multiple data repositories supporting the hundreds of projects that are part of the Helping to End Addiction Long-term (HEAL) Initiative.

Data Repositories: 9

Biomedical Research Hub
Powered by Gen3

The Biomedical Research Hub enables search, discovery and the analysis of data from over 10 data commons from NIH Institutes, Centers and projects.

Data Repositories: 11

Data Commons

NIH NATIONAL CANCER INSTITUTE Cancer Research Data Commons

83,709 Subjects
622 Attributes
52,141,509 Files
4.90 PB Total Size

BDGC

107,418 Subjects
786 Attributes
16,824 Files
10.94 TB Total Size

Veterans Affairs Data Commons

658,278 Subjects
1,606 Attributes
468 Files
1.40 TB Total Size

NIH National Institute of Allergy and Infectious Diseases AccessClinicalData@NIAID

2,096 Subjects
151 Attributes
10 Files
3.88 MB Total Size

CANINE Data Commons

1,499 Subjects
1,048 Attributes
3,820 Files
1.88 TB Total Size

GenoMEL the Melanoma Genetics Consortium

1,390 Subjects
387 Attributes
6,555 Files
31.60 TB Total Size

NIH HEAL INITIATIVE Justice Community Opioid Innovation Network (JCOIN)

237 Subjects
517 Attributes
391 Files
1.30 GB Total Size

Environmental Data Commons

265 Attributes
33,441,289 Files
99.20 TB Total Size

BloodPAC BLOOD PROFILING ATLAS IN CANCER

4,839 Subjects
888 Attributes
35,549 Files
34.57 TB Total Size

CHICAGOLAND COVID-19 COMMONS

53,728 Subjects
1,464 Attributes
285,849 Files
117.64 TB Total Size

The AnVIL

41,933 Subjects
551 Attributes
200,397 Files
803.96 TB Total Size

NIH National Heart, Lung, and Blood Institute BioData CATALYST

438,874 Subjects
770 Attributes
712,329 Files
3.88 PB Total Size

End to End Design Principle

What are the fewest number of services that can support the interoperability of data commons?

These are the data mesh (aka framework services).

These include Fence, Indexd, Gen3 Metadata Service, etc.

End-To-End Arguments in System Design

J. H. SALTZER, D. P. REED, and D. D. CLARK
Massachusetts Institute of Technology Laboratory for Computer Science

This paper presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. Examples discussed in the paper include bit-error recovery, security using encryption, duplicate message suppression, recovery from system crashes, and delivery acknowledgment. Low-level mechanisms to support these functions are justified only as performance enhancements.

CR Categories and Subject Descriptors: C.0 [**General**] Computer System Organization—*system architectures*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*protocol architecture*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; D.4.7 [**Operating Systems**]: Organization and Design—*distributed systems*

General Terms: Design

Additional Key Words and Phrases: Data communication, protocol design, design principles

1. INTRODUCTION

Choosing the proper boundaries between functions is perhaps the primary activity of the computer system designer. Design principles that provide guidance in this choice of function placement are among the most important tools of a system designer. This paper discusses one class of function placement argument that

Source: ACM Transactions on Computer Systems (TOCS), Volume 2 Issue 4, Nov. 1984, Pages 277-288

2. Data Commons vs Data Meshes

Data Commons vs Data Ecosystems

GEN3 DATA COMMONS

Data Commons

Question: Given data from multiple projects within a particular discipline or domain, how do you curate and harmonize the data and make it available to the research community?

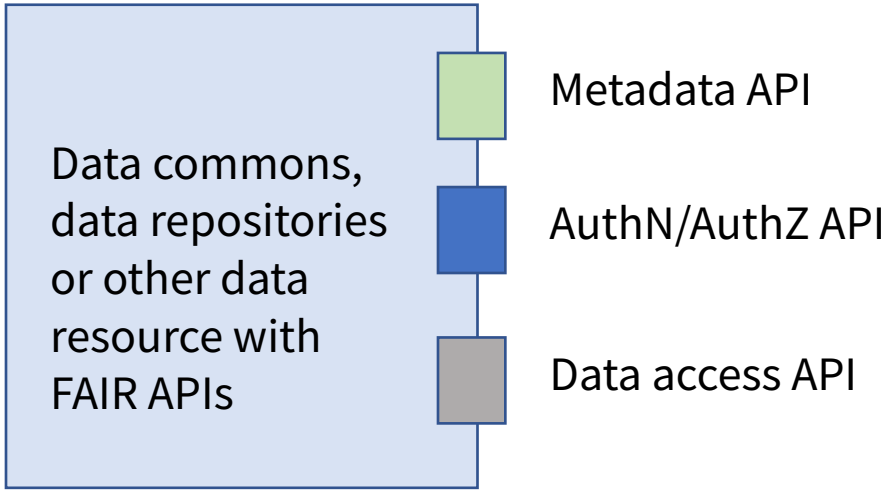
Assumptions: there is a common data model and data is curated and harmonized

Data Ecosystems

Question: Given multiple data repositories and data commons, how do you search for relevant data and bring it into a workspace to explore and analyze it?

Assumptions: there are multiple data models, but standard APIs for authN/authZ and data access; data is (generally) accessed at the dataset level or the data object level

Connect to a Gen3 Data Mesh

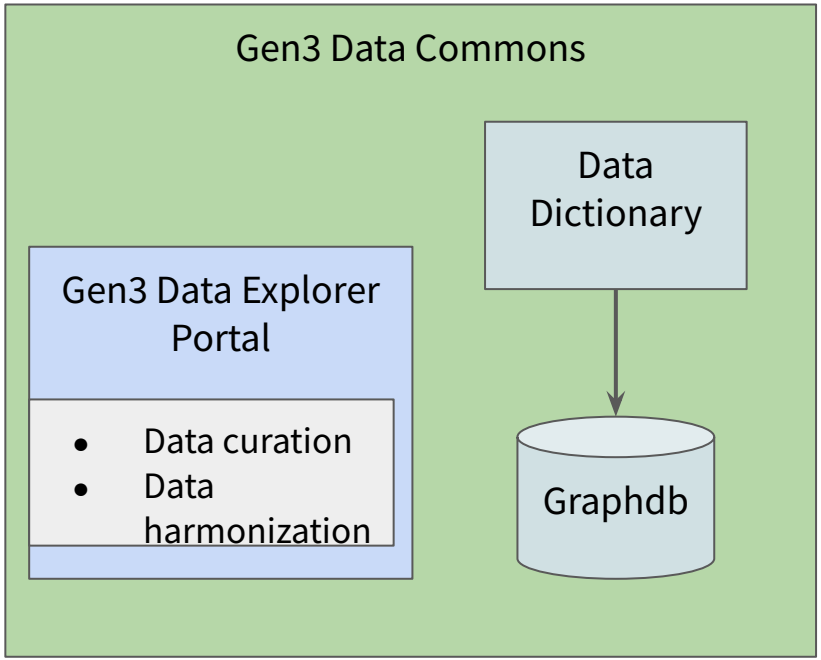


To be part of the data ecosystem a data commons or data resource must expose three APIs:

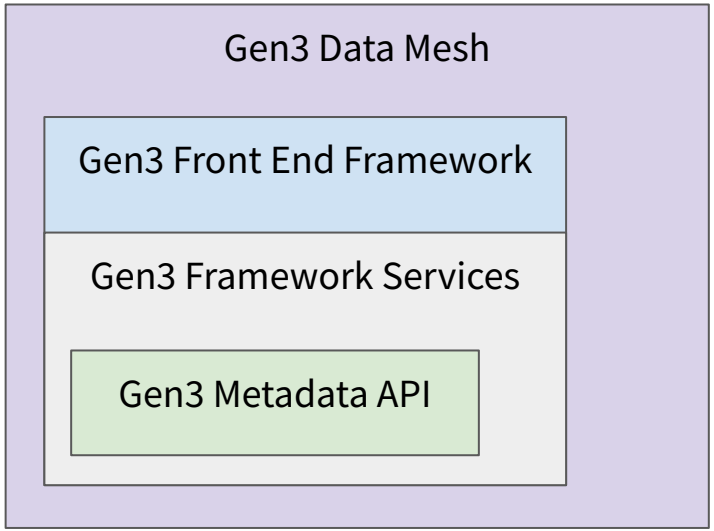
1. AuthN/AuthZ API
2. (FAIR) Metadata API
3. (FAIR) Data API

Gen3 data meshes discovery portals enable interactive data discovery and data exploration over all data commons, data repositories, and other cloud-based resources that expose FAIR APIs.

Data Commons Portal vs Data Ecosystem Browser



1. Set up a data model
2. Harmonize data at the subject level
3. Ingest and curate data
4. Build a front end



1. Leverage dataset metadata
2. Use framework services to assign digital IDs (GUIDs) and metadata
3. Use front end framework to build ecosystem browser.
4. Select datasets export to workspace

Gen3 Data Commons vs Data Meshes

- **Data commons** software platforms that co-locate: 1) curated data, 2) cloud-based computing infrastructure, and 3) commonly used software applications, tools and services to create a governed resource for managing, analyzing and sharing data with a research community.

- **Data meshes** (aka data ecosystems) integrate multiple data commons, computational platforms, and other cloud-based resources operated by different organizations, along with a hybrid governance framework, and enable the management, discovery, analysis and sharing of data.

- **Data Mesh Services** (aka Data Commons Framework Services) are a set of services to to develop and operate data commons and data meshes.

- **Gen3** is an open-source data platform for building data commons and data meshes over a set of data mesh services.

Source: - Robert L. Grossman, Data Lakes, Clouds and Commons: A Review of Platforms for Analyzing and Sharing Genomic Data, Trends in Genetics 35, 2019, pages 223-234. arxiv.org/abs/1809.01699 PMID: 30691868 PMCID: PMC6474403
- Craig Barnes, Binam Bajracharya, …, and Robert L. Grossman, The Biomedical Research Hub: A Federated Platform for Patient Research Data, Journal of the American Medical Informatics Association, 2021, doi:10.1093/jamia/ocab247.

Gen3 Data Platform (2023 Version)

- Gen3 Data Commons

- Gen3 Data Commons and Data Ecosystems
- Gen3 Framework Services
  - (**Fence, Indexd, Metadata Service**, etc.)
- Gen3 Workspaces, Data Ingestion, Integration & Rel. Man. (DIIRM), Gen3 Frontend Framework, etc.

1.6 million subjects
91 million data objects
17 PB of FAIR data

Gen3 Data Commons (G3DC)

G3 Workspaces | G3 Apps | Third Party Apps

Gen3 Frontend Framework (G3FF)

Gen3 Services for ETL, search, query, etc.

G3DC | Data Commons | Data Repositories

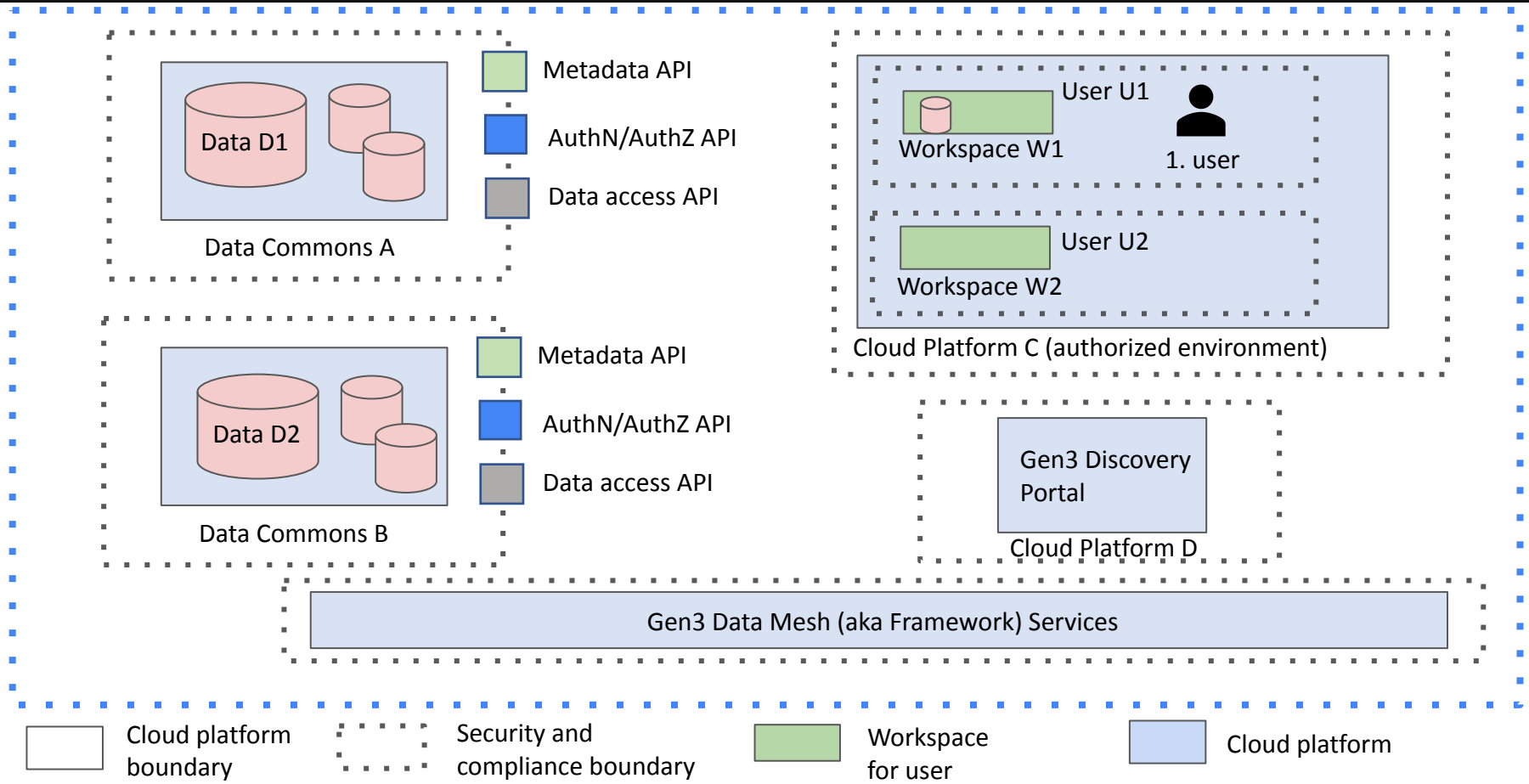
Gen3 Mesh Services

2017-2020

2021 - 2024

3. Security, Compliance & Governance, for Gen3 Data Meshes

Gen3 Security & Compliance Follows NIST SP 800-53 Moderate



Hybrid Governance Model

GEN3 DATA COMMONS

Data Commons Governance

dbGaP GENOTYPES and PHENOTYPES

MIDRC MEDICAL IMAGING AND DATA RESOURCE CENTER

NIH NIDDK BDGC

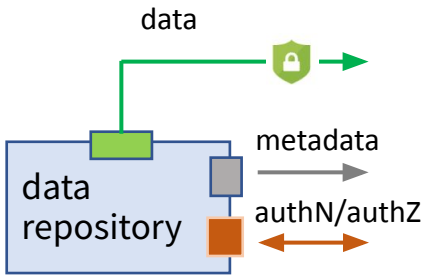
Gabriella Miller Kids First PEDIATRIC RESEARCH PROGRAM Data Resource Center

NIH National Heart, Lung, and Blood Institute BioData CATALYST

NIH HEAL INITIATIVE | Justice Community Opioid Innovation Network (JCOIN)

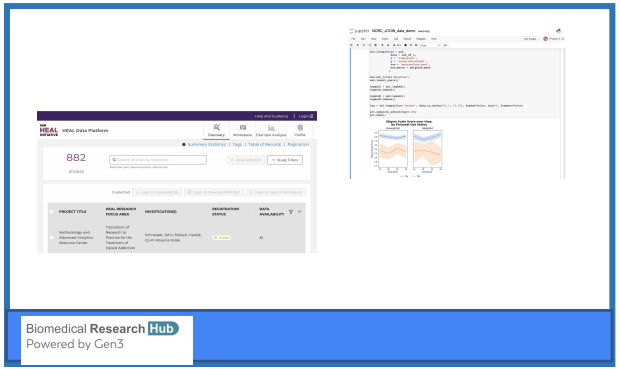
- DUA agreements between data submitters & repositories
- Required metadata, CDE, etc.
- Any data curation, etc.

Shared Governance (between commons & mesh platform)



- Data Mesh Services
- FAIR APIs
- Interoperating AuthN/AuthZ
- System “Interoperability Agreement”

Data Mesh Governance (mesh platform)

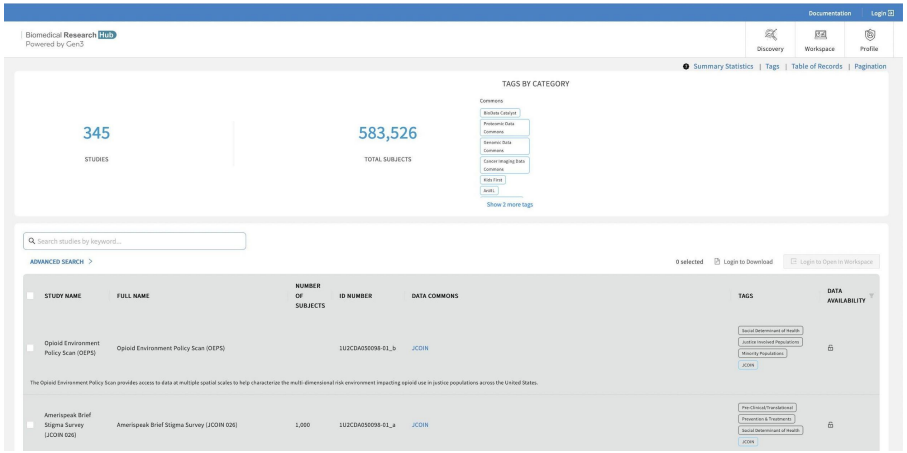


- Which data repositories to connect to
- Governance rules for authorizing workspaces

Source: Craig Barnes, Binam Bajracharya, …, and Robert L. Grossman, The Biomedical Research Hub: A Federated Platform for Patient Research Data, Journal of the American Medical Informatics Association, 2021, doi:10.1093/jamia/ocab247.

4. Examples of Gen3 Data Meshes

The Biomedical Research Hub (BRH)



BRH Discovery Portal. Each data commons or data resource in the BRH data mesh exposes metadata about its datasets through FAIR APIs. The Gen3 Discovery Portal uses the metadata to power search. Data can then be explored and analyzed in workspaces. BRH is a joint project between the Center for Translational Data Science at the University of Chicago, OCC and AWS.

- The Biomedical Research Hub (BRH) is a data platform operated by the Center for Translational Data Science (CTDS) at the University of Chicago at a FISMA Moderate security & compliance level with an ATO from NIH.
- The BRH is part of the NIH STRIDES program.
- Projects can set up their own data commons within the BRH and use the BRH for their data mesh services and their security and compliance services.
- Researchers can use the BRH Discovery Portal to find datasets of interest and use secure and compliant workspaces to access and analyze their data.
- BRH uses a hybrid governance model between the projects that operate the data commons and CTDS that operates the mesh services.

THE UNIVERSITY OF CHICAGO | Center for Translational Data Science

Open Commons Consortium

The HEAL Data Platform is a Gen3 Mesh for the NIH HEAL Initiative

GEN3 DATA COMMONS

FAIR API for metadata

FAIR API for data (e.g. GA4GH DRS)

NIH RAS, GA4GH Visas and Passports

FAIR = Findable, Accessible, Interoperable & Reusable

HARVARD Dataverse

figshare

NDA

dbGaP GENOTYPES and PHENOTYPES

Vivli

MIDRC

ICPSR 60 years

NIH HEAL INITIATIVE | Justice Community Opioid Innovation Network (JCOIN)

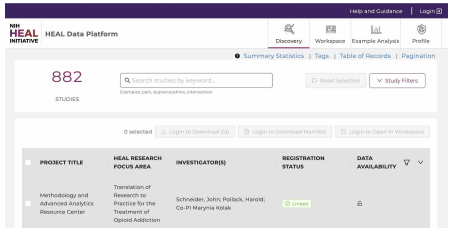
data submission

data

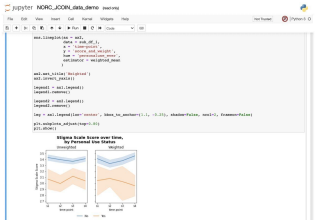
metadata

data repository

Authentication & Authorization



HEAL Data Portal for search and discovery



Notebooks in secure workspaces supporting interactive data analysis

NIH HEAL INITIATIVE

1

2

GEN3

HEAL uses Gen3 for the Platform and the FAIR data services

HEAL Data generators and providers

Multiple data repositories

HEAL Platform

Researcher

GEN3 DATA COMMONS

HEAL Data Platform

Phil Schumm

NIH HEAL INITIATIVE HEAL Data Platform

HEAL Data Mesh

Phil Schumm, Department of Public Health Sciences, University of Chicago (with Bob Grossman and the HEAL Platform Team at CTDS)

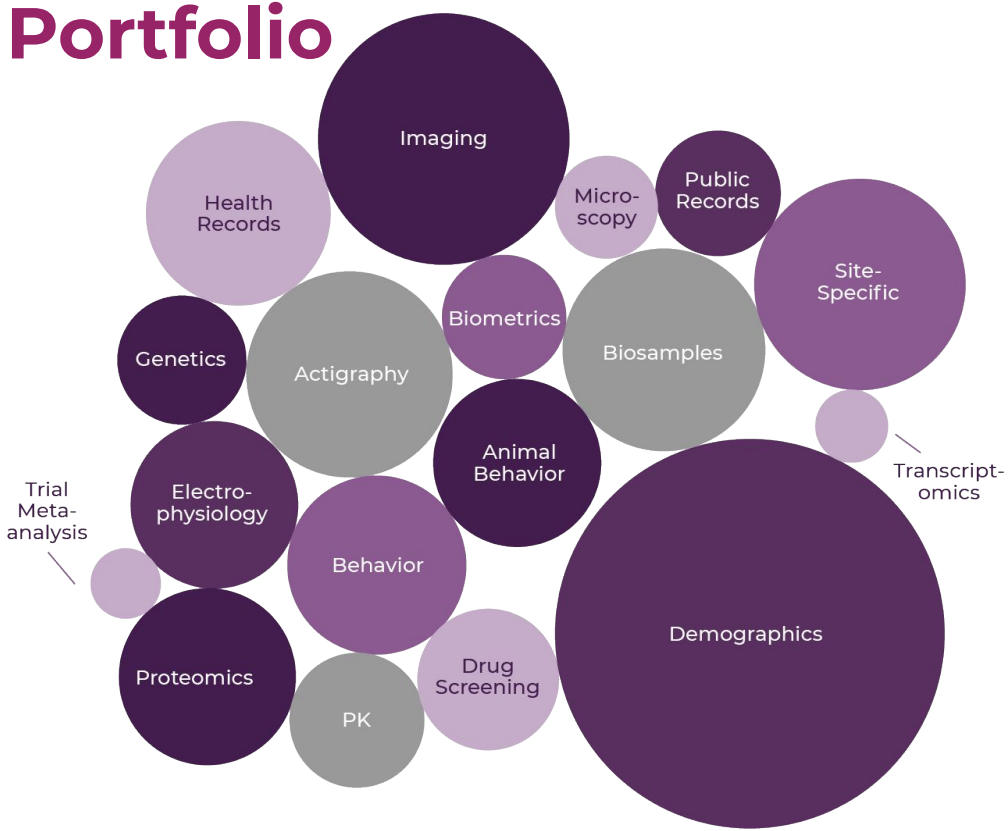
NIH's Helping to End Addiction Long-term (HEAL) Initiative

- Trans-agency effort to address the U.S. national opioid crisis
- 800+ NIH-funded studies within two broad areas:
  - Improving Prevention and Treatment for Opioid Misuse and Addiction
  - Enhancing Pain Management
- Strong data sharing mandate

HEAL Studies Comprise a Remarkably Diverse Research Portfolio

~800 Studies

20+ data types

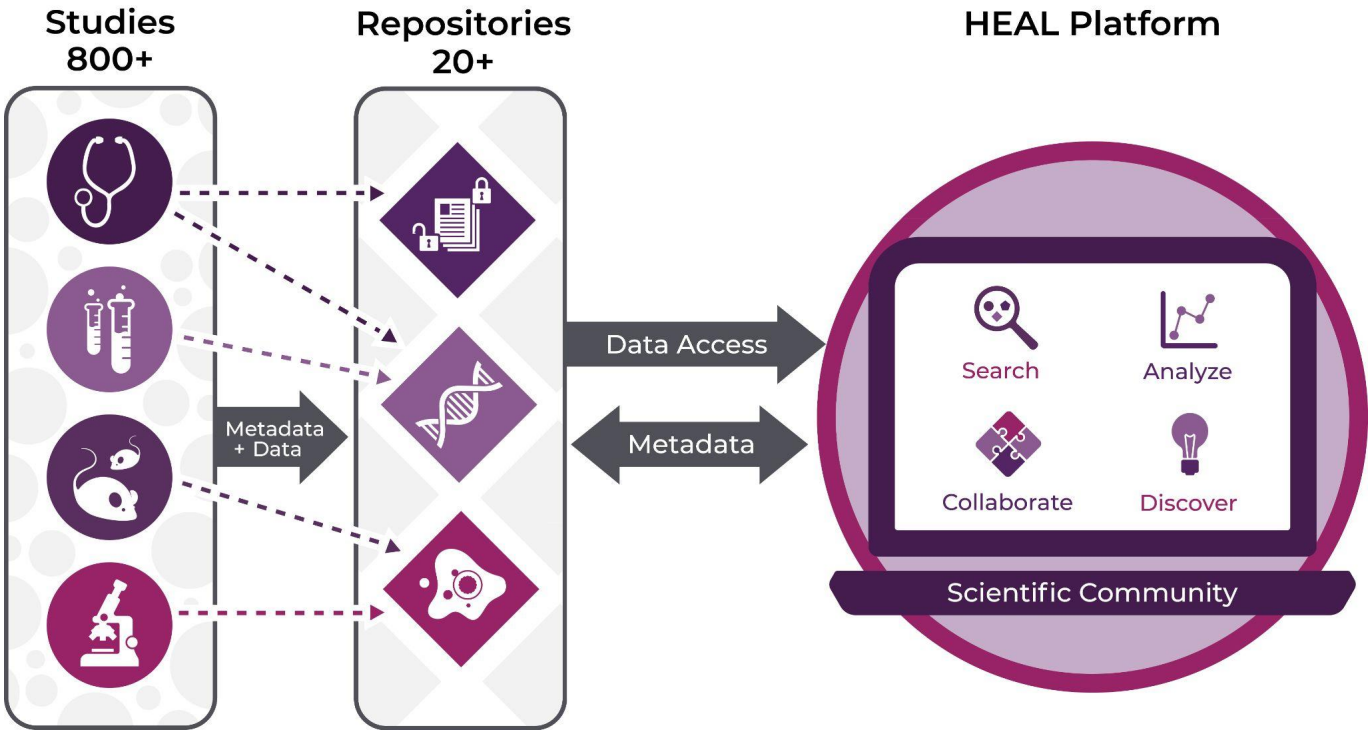


**This figure represents a small sampling of the 800 HEAL-funded studies*

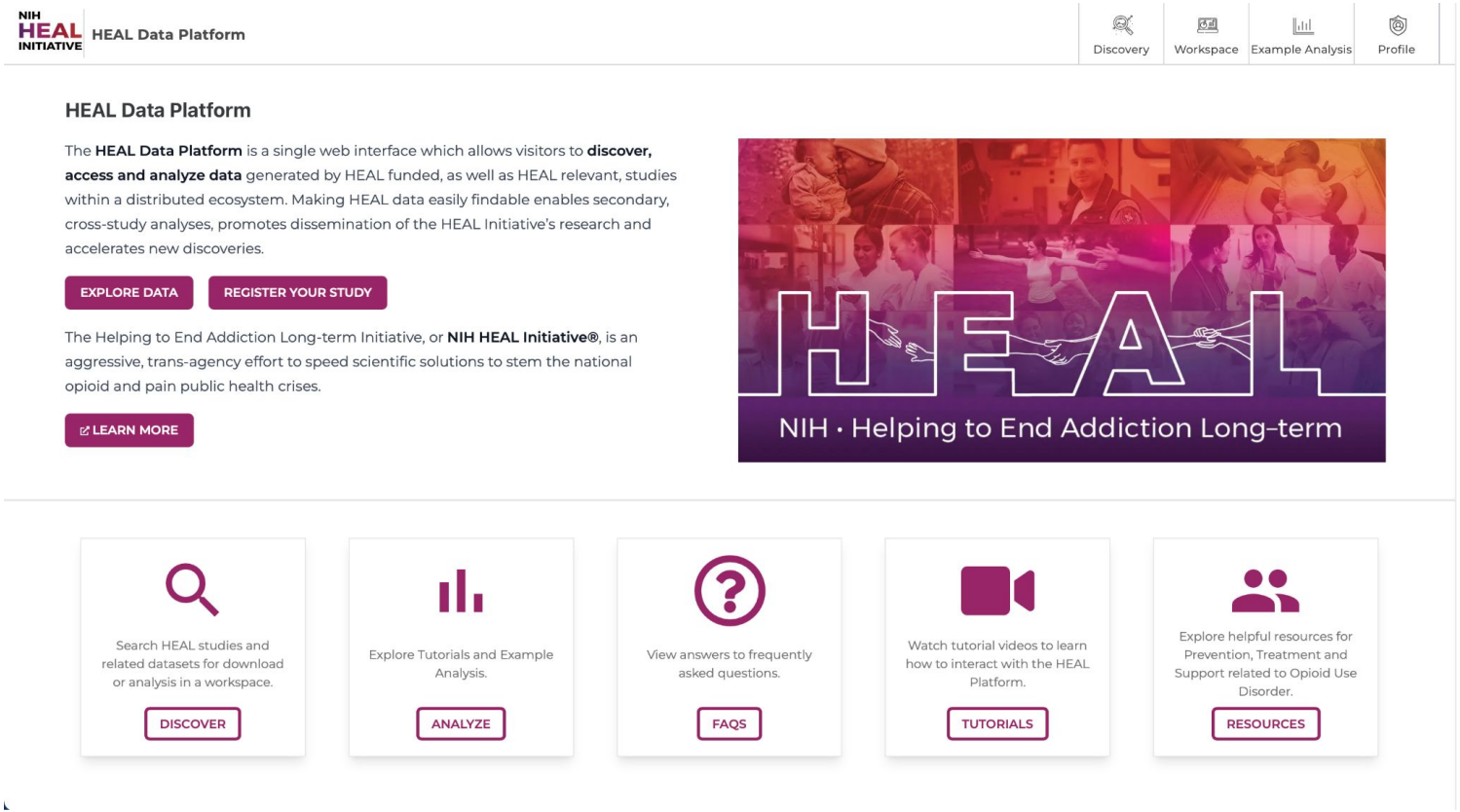
Rationale for a Data Mesh

- Extremely broad range of disciplines, study designs, and data types favor specialized repositories
- NIH's desire to reuse existing repository resources (including procedures for requesting and granting data access)
- Need for rapid sharing and collaboration, including across disciplines, to speed scientific discovery

HEAL Data Ecosystem



HEAL Data Platform

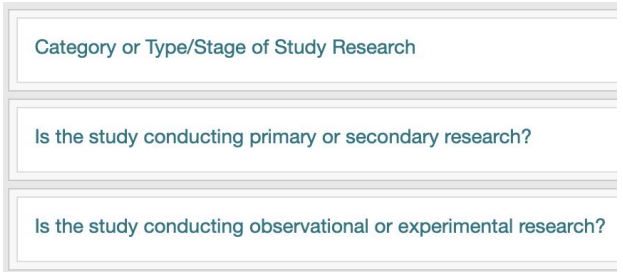


HEAL Data Platform

- Metadata for 800+ HEAL-funded studies ingested from NIH RePORTER
- 160+ HEAL investigators have registered their studies
- Aggregates additional study-level metadata from ClinicalTrials.gov and individual data repositories
- Investigators enter and update study-level and variable-level metadata for their own studies
- Investigators can request access to cloud-based workspaces
- Working on establishing interoperability with HEAL-recommended repositories

Pre-collection of Metadata

- **Study-level** metadata include information about your objectives and execution
  - Provided via a structured form in CEDAR

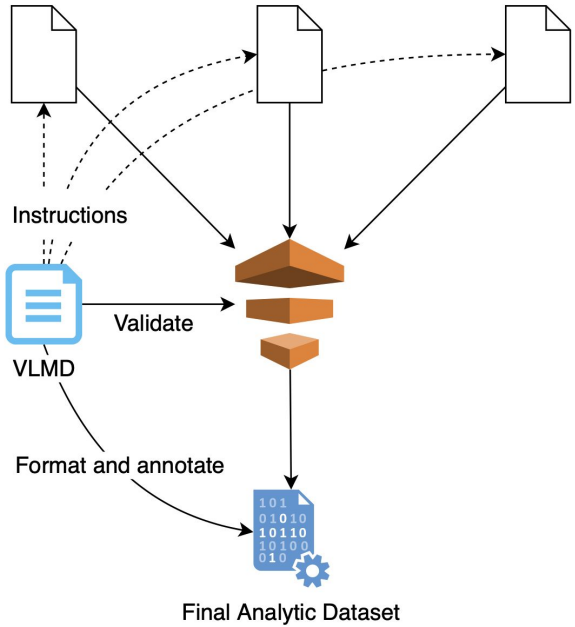


- **Variable-level** metadata (sometimes referred to as data dictionaries) include information about each of the variables you are collecting
  - Extracted automatically from datasets using tools we provide
  - May be further annotated to include descriptions and **linkages to Common Data Elements (CDEs), vocabularies and ontologies**

	A	B	C
1	**name**	**description**	**type**
2	participant_id	Unique identifier for participant	string
3	race	Self-reported race	integer
4	age	What is your age? (age at enrollment)	integer
5	hispanic	Are you of Hispanic, Latino, or Spanish origin?	boolean
6	sex_at_birth	Sex of the participant at birth	string
7			
8			

Metadata Can be Used to Plan and Facilitate Meta-analyses and Collaborative Research

- Identify candidate studies for inclusion based on characteristics of study and the data collected
- Variable-level metadata provide a formal, standardized way to communicate data requirements among collaborators and help data harmonization
- Variable-level metadata can be used automatically to validate data submissions and format data for analysis



Each of these can be done **prior to submission of data to a repository.**

Heterogeneity Across Repositories

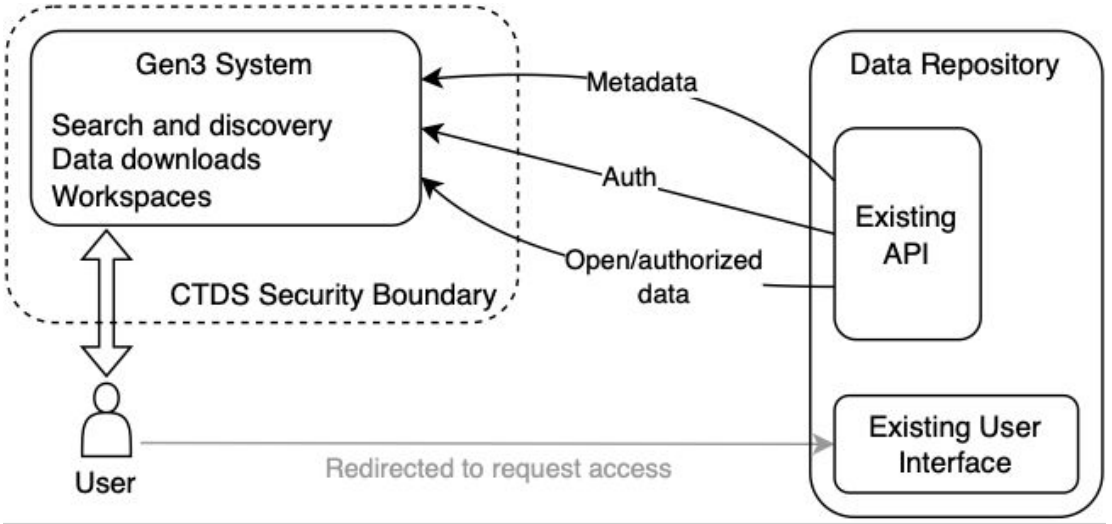
- Different disciplines and cultures
- Different degrees of openness and mechanisms for requesting and approving access
- Different API capabilities
- Exposure to interoperability

Gen3's microservice architecture, openness and adherence to standards provides the flexibility needed.

Example 1. Data repository or commons with fully functional API

Ideal for:

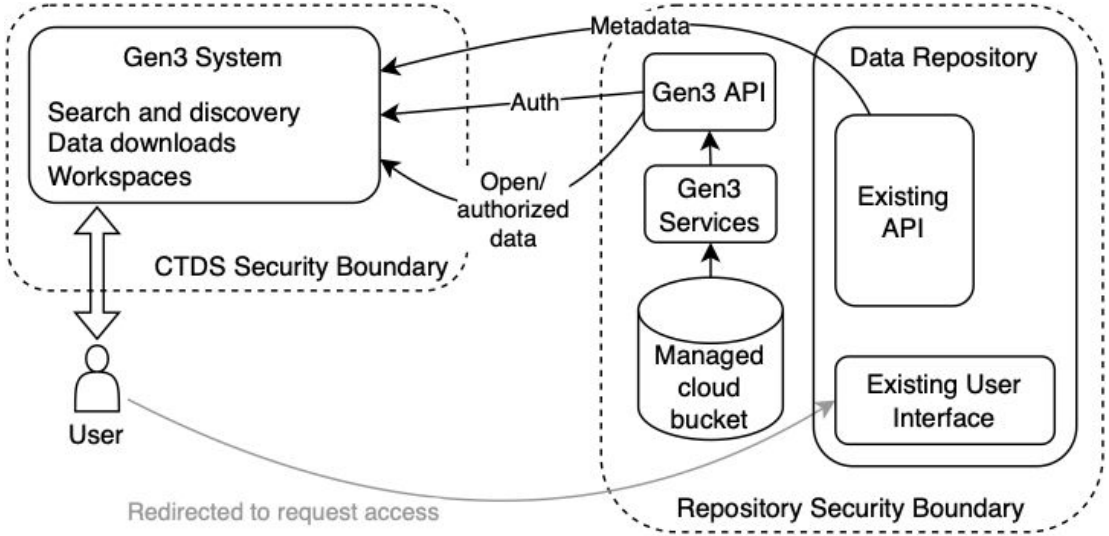
- Repositories for which all metadata and data are openly accessible and which have at least minimal APIs permitting access to metadata and data
- Repositories which contain restricted access data but which have an API permitting secure authentication and authorization



Example 2. Repositories able to manage Gen3 FAIR services-enabled cloud bucket

Ideal for:

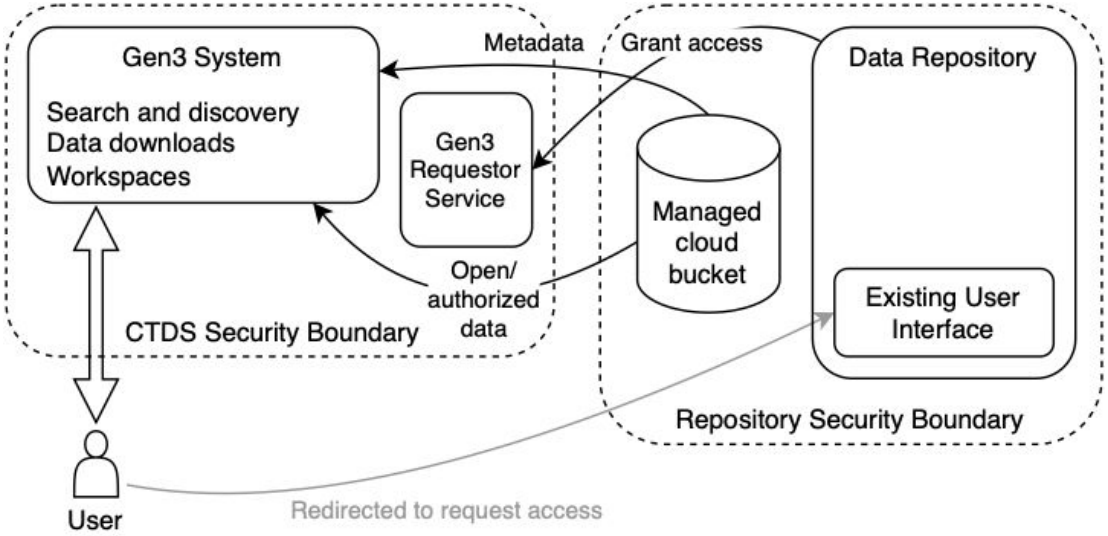
- Repositories that currently have only a partially-functional API (e.g., API for metadata only)
- Repositories that do not currently have an API but are planning to develop one
- Repositories that have no current plans to develop an API but are interested in trying out one or more Gen3 components



Example 3. Repository does not run additional services

Ideal for:

- Repositories that do not have a current API nor plans to develop one, and do not have the resources to manage additional services



Biomedical Research Hub

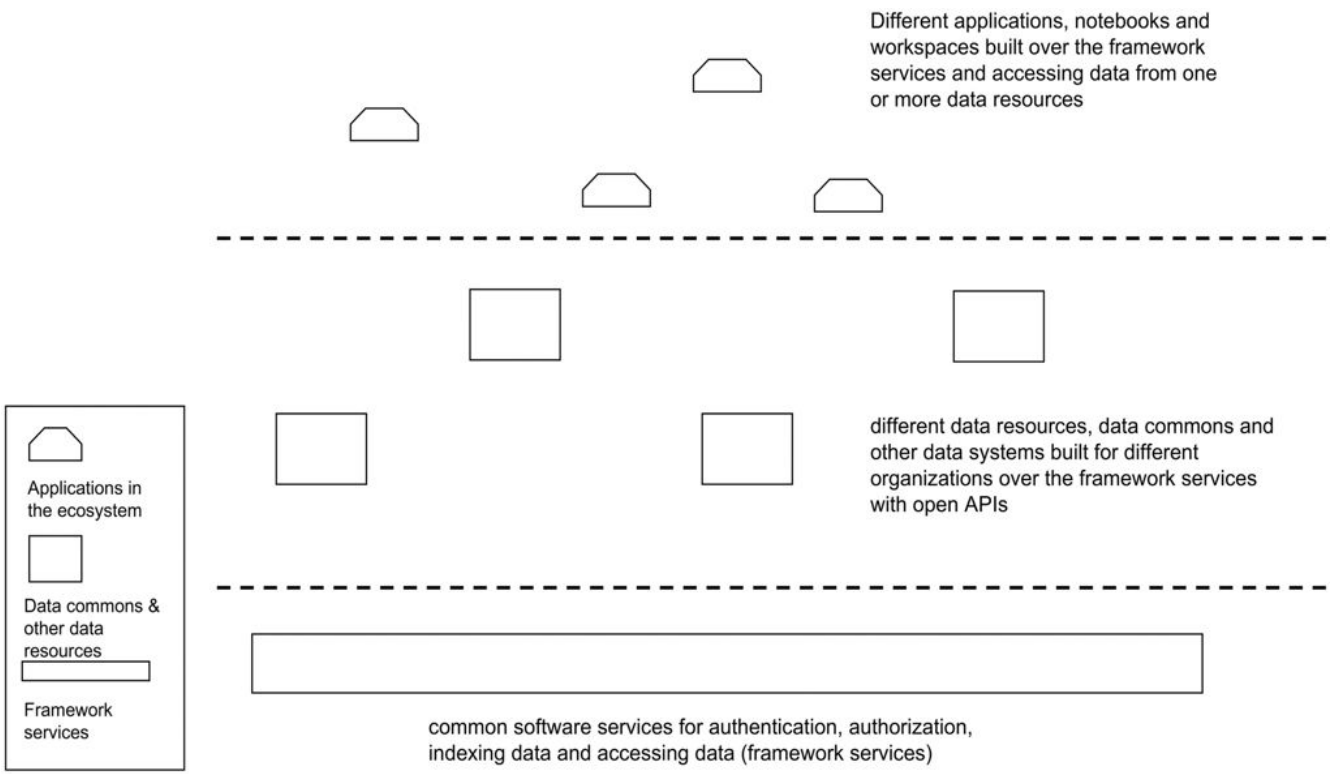
Aarti Venkat

Biomedical Research Hub
Powered by Gen3

How to search and discover effectively across multiple data commons?

GEN3 DATA COMMONS

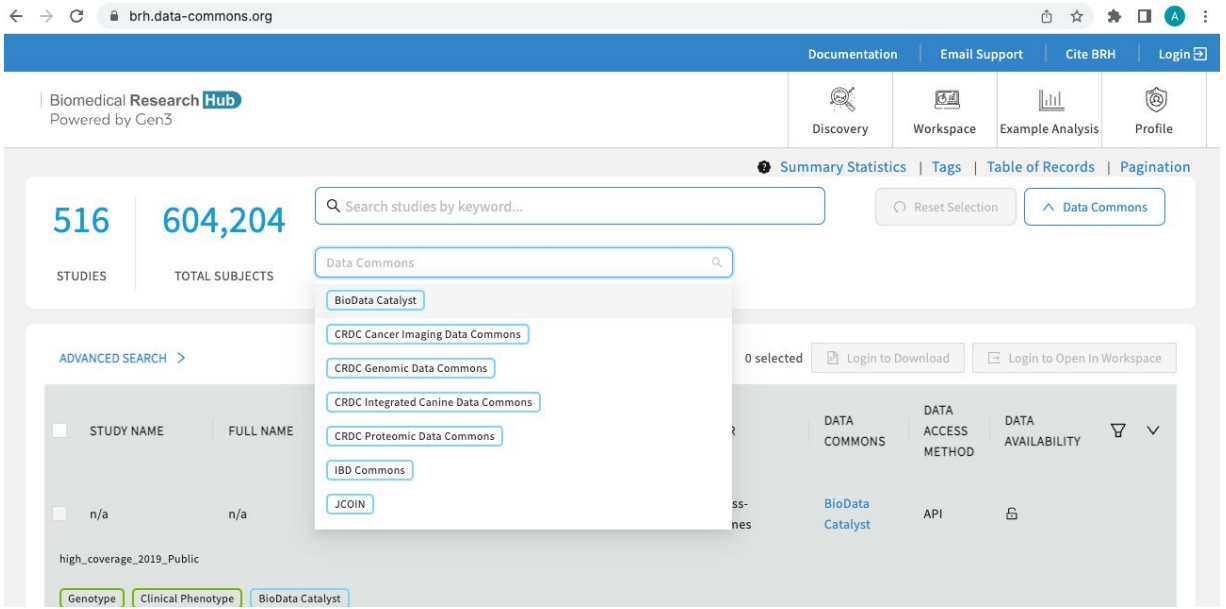
Architectural overview of BRH



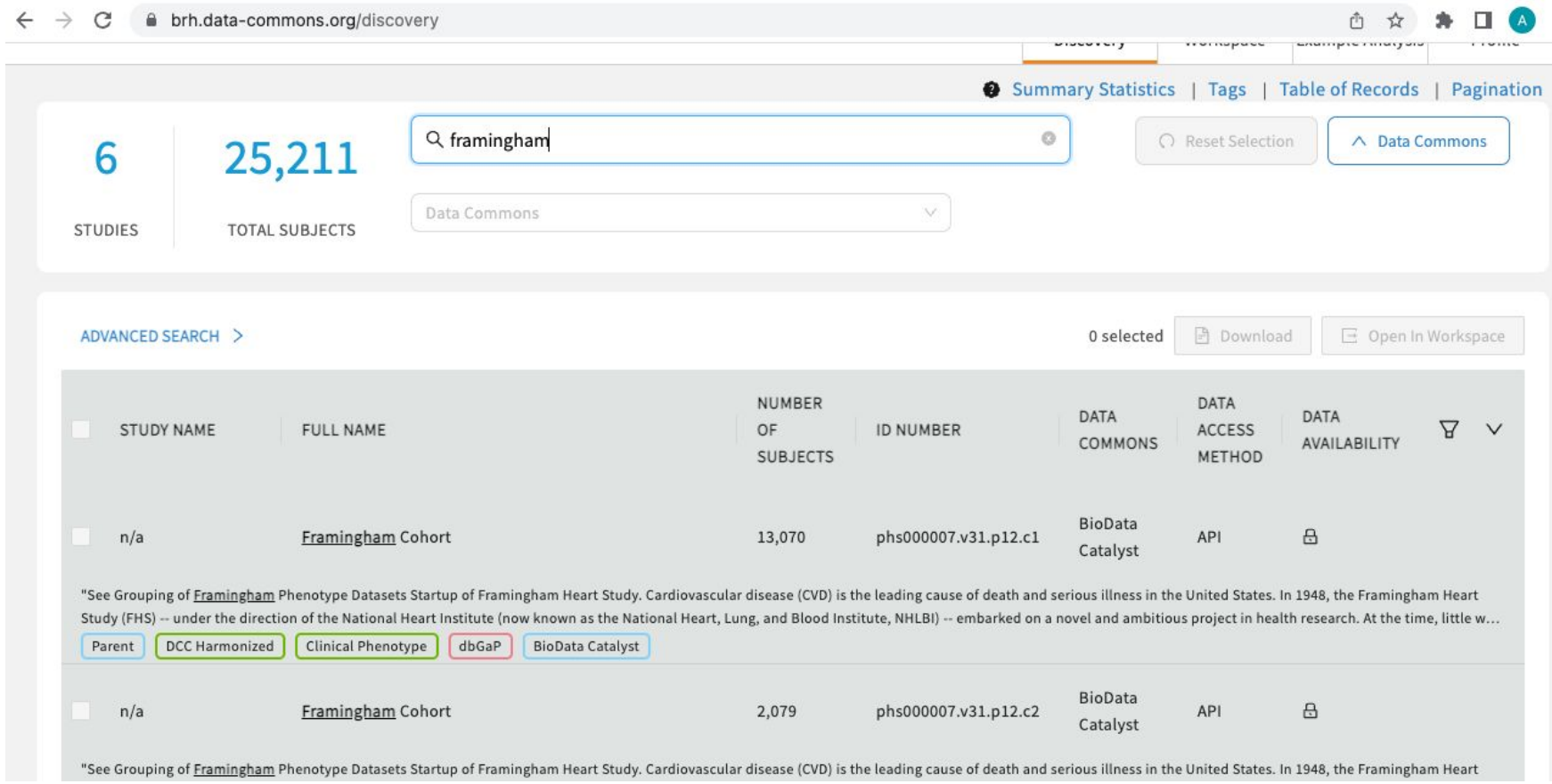
Barnes et al, 2021

Broad range of study types and resulting data

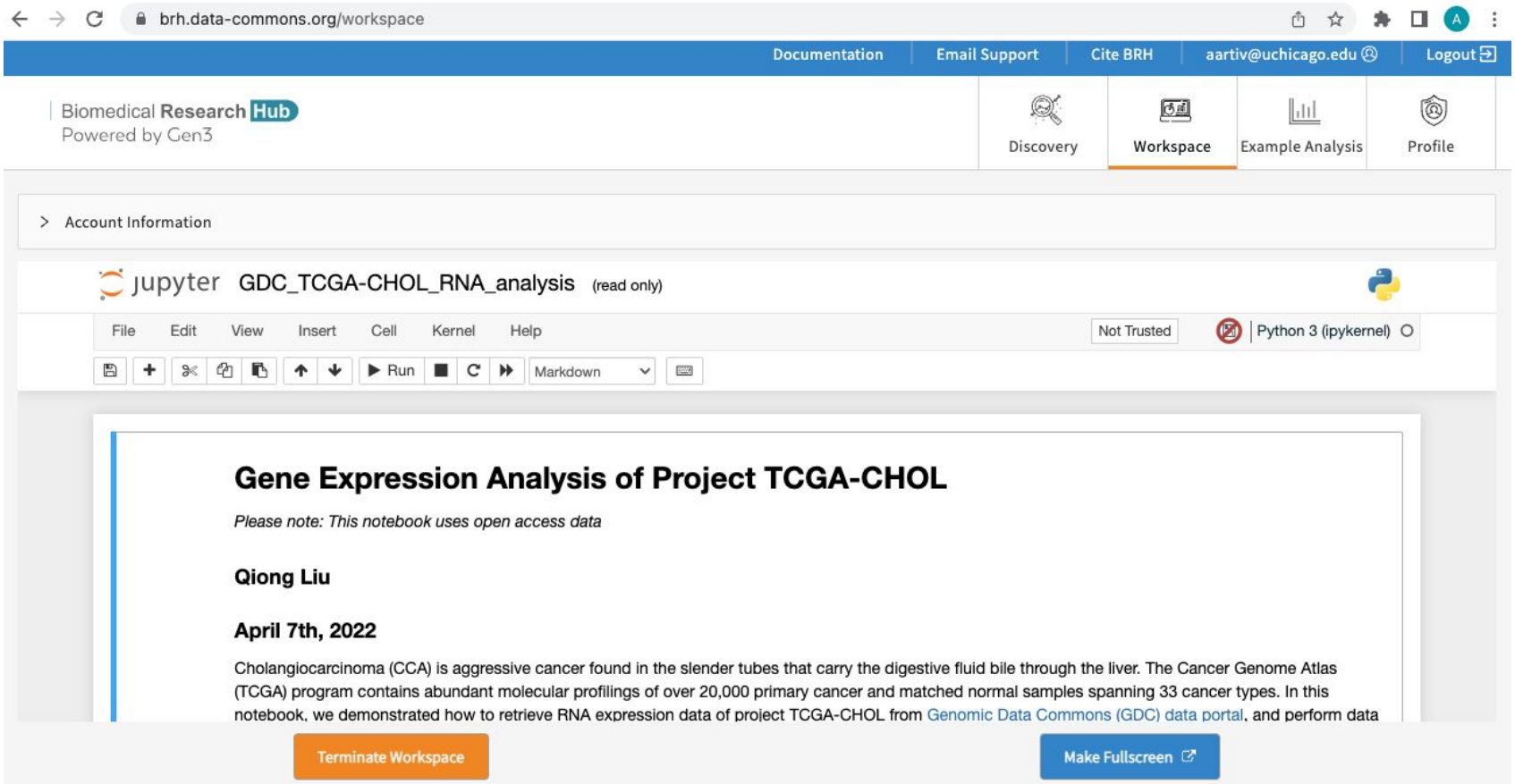
- Imaging
- Transcriptomics
- Genetics and genomics
- Single cell
- Proteomics
- Clinical trials/phenotypes
- Demographics
- Public records



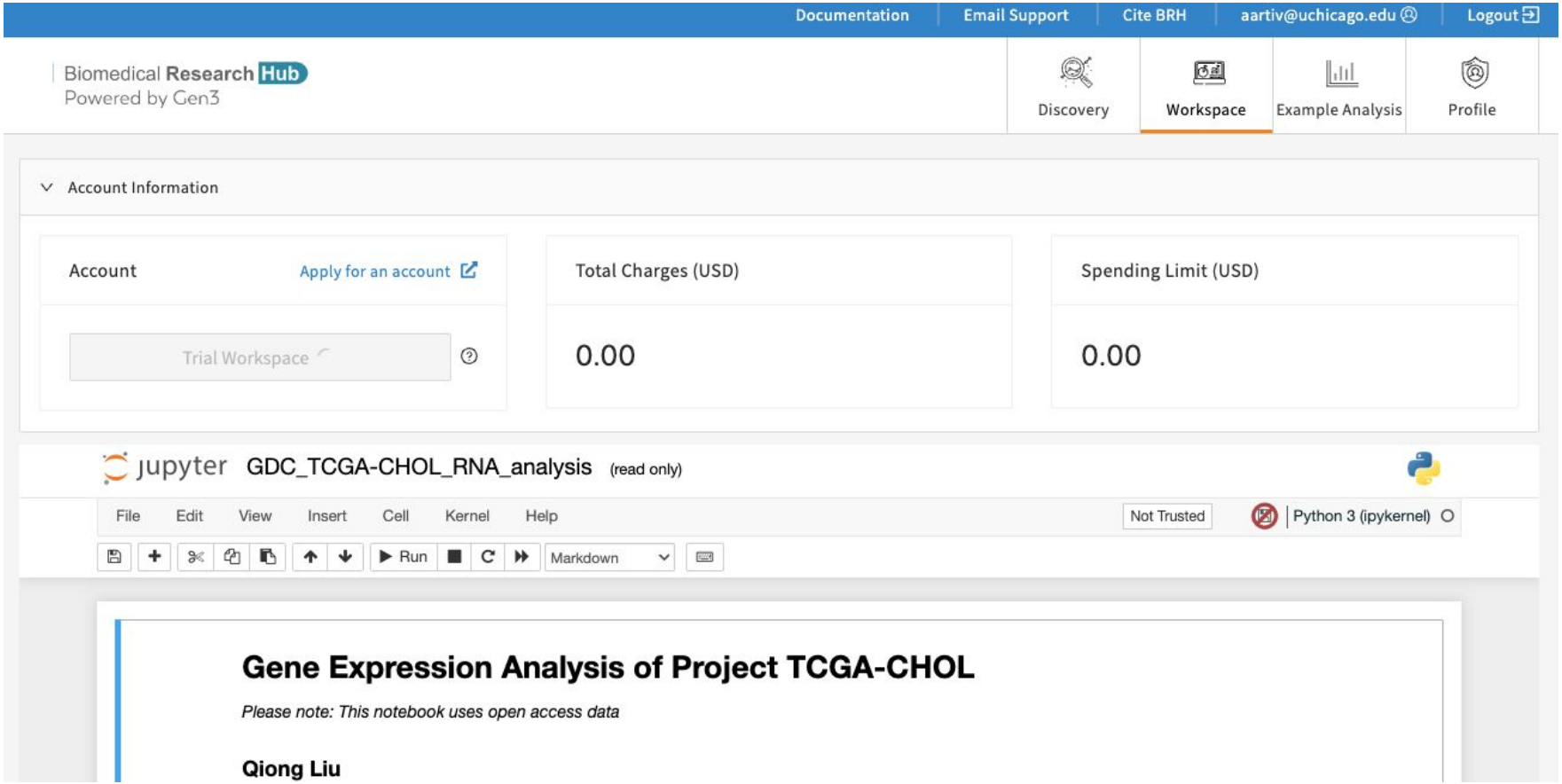
Special features of BRH: Search, Data access, Data availability



Special features of BRH: Compute workspaces

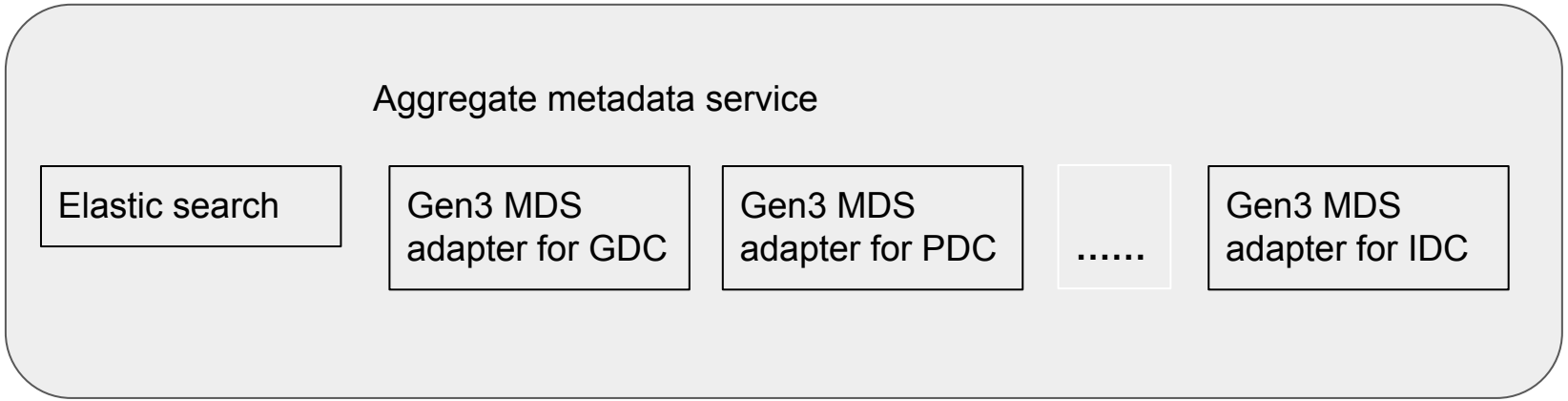


Special features of BRH: Track spending limits



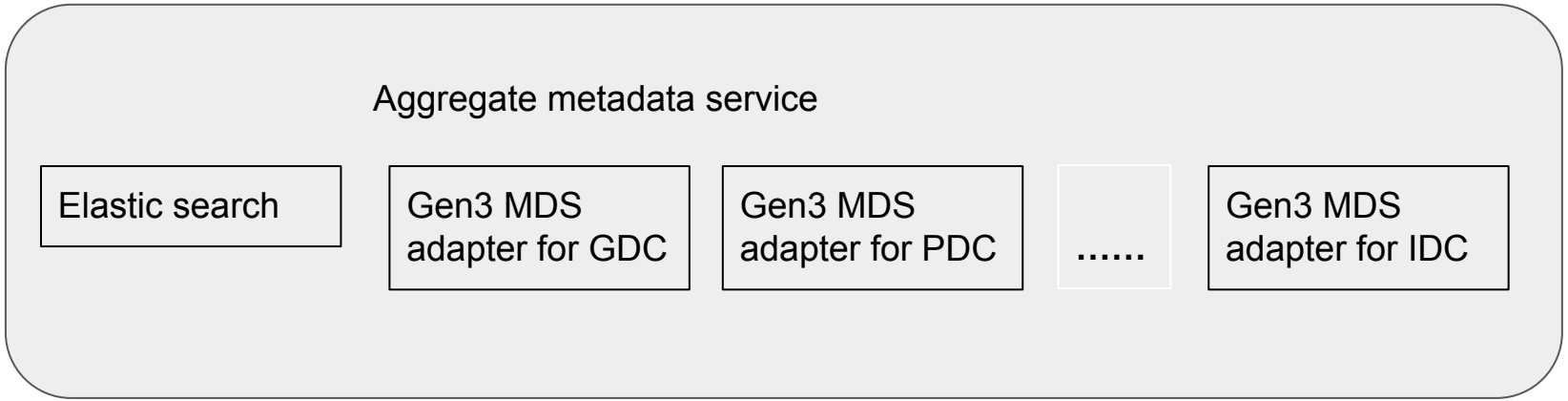
Special features of BRH: AggMDS

Aggregate metadata service (aggMDS) caches the metadata from two or more metadata sources to provide a unified view of the commons on the discovery page



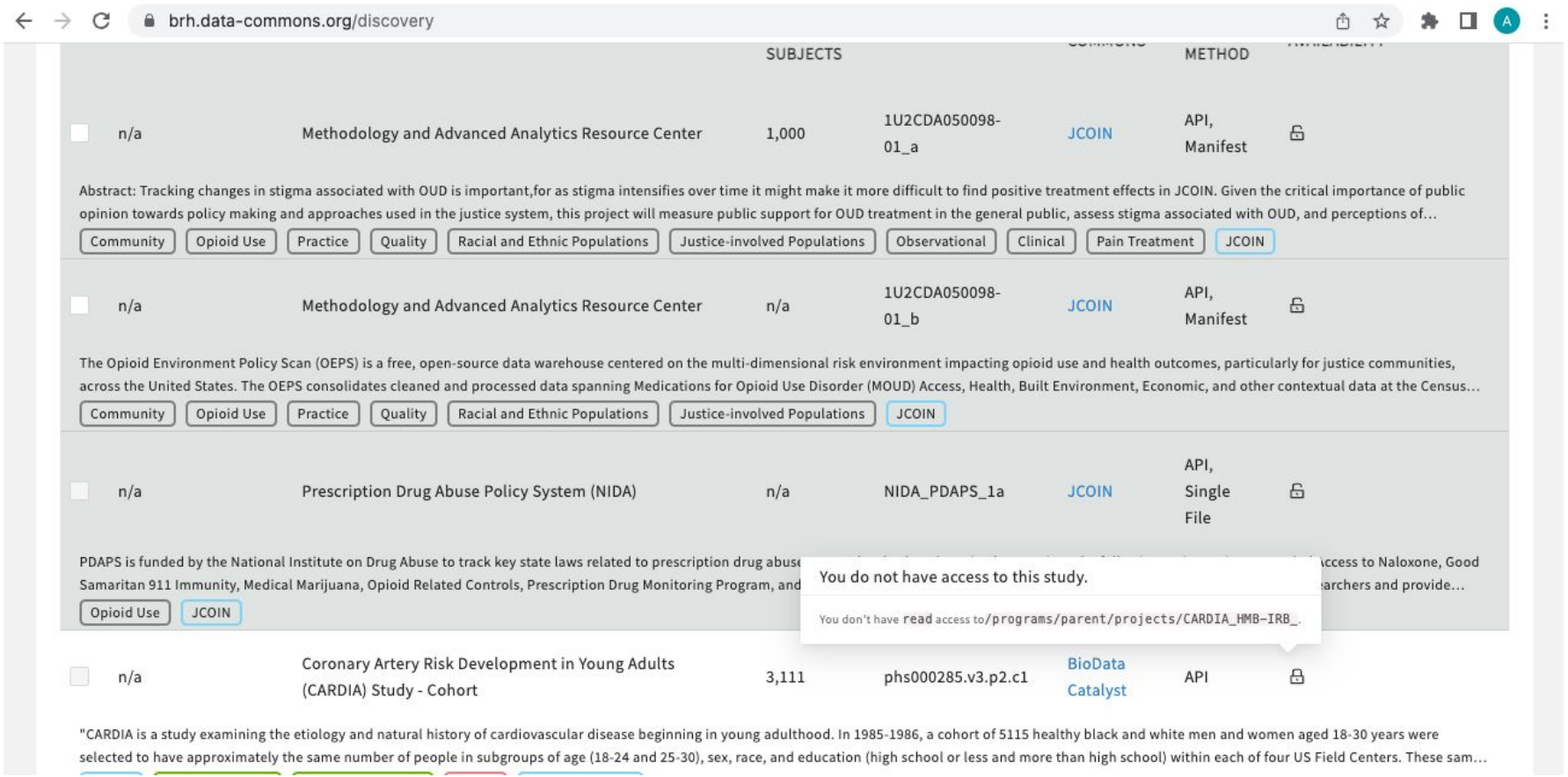
Special features of BRH: AggMDS

Aggregate metadata service (aggMDS) caches the metadata from two or more metadata sources to provide a unified view of the commons on the discovery page

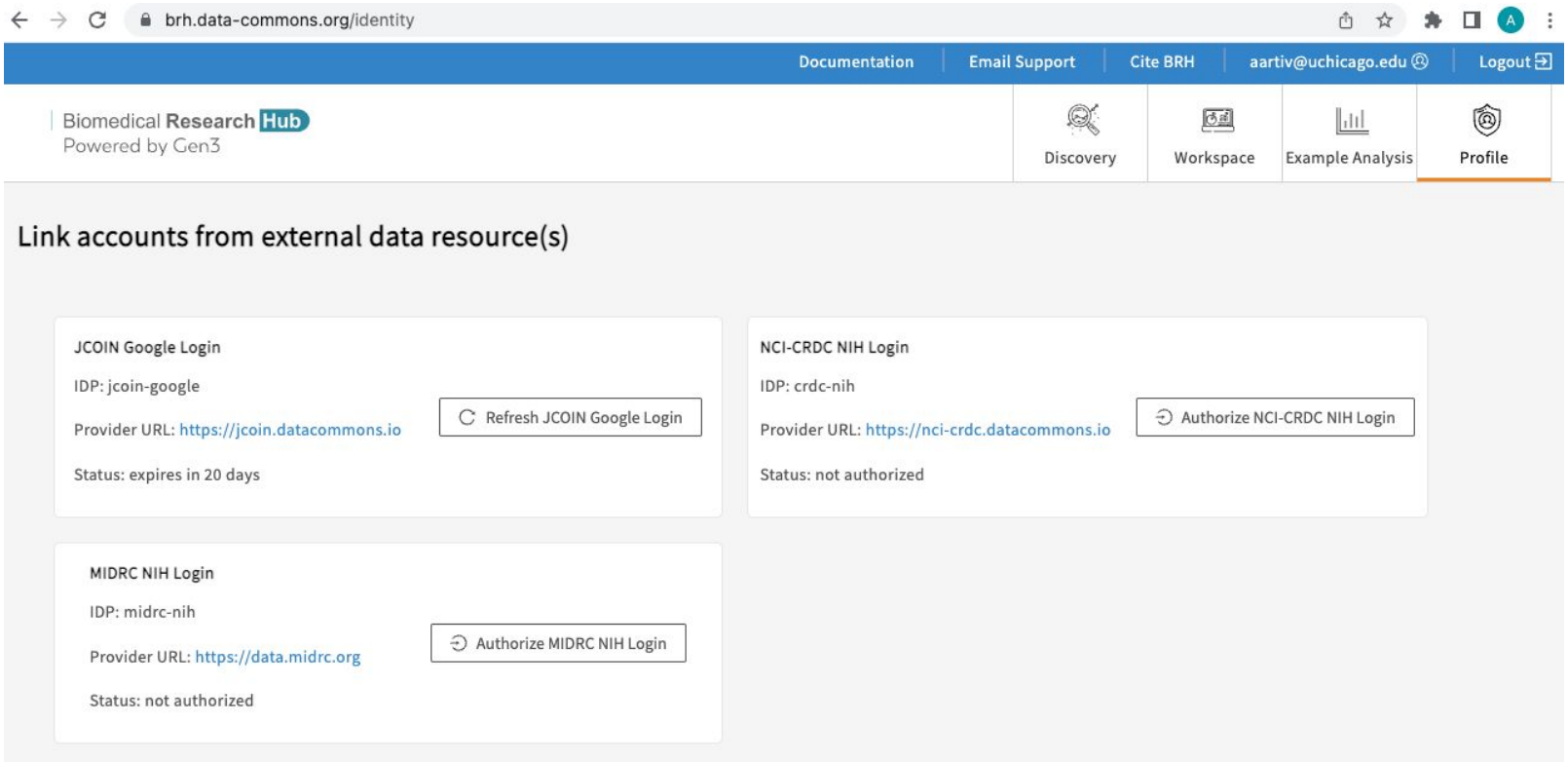


How to update aggMDS with an additional data commons and power portal?

Special features of BRH: Workspace token service



Special features of BRH: Workspace token service



How to Set up a Mesh

Sai Narumanchi

Agenda

- **Data Commons and Meshes**
- Key features for Data Meshes
- Aggregate Metadata Sync
- Token Service (Workspace Token service)

Data Mesh (aka “Data Ecosystem”)

- Contain **multiple data commons and/or data repositories** and cloud computing infrastructure
- Use the same framework services as a data commons for the fundamental behavior.
  - Fence – used for AuthN and AuthZ services utilizing OpenID connect flow.
  - Arborist – an Attribute Based Access control policy engine.
  - Windmill (aka Data Portal) – a Front end interactive website.
- Also has a few additional services that allow connecting and interacting with multiple data commons.

E.g., BRH, HEAL

Agenda

- Data Commons and Meshes
- **Key features for Data Meshes**
- Aggregate Metadata Sync
- Token Service (Workspace Token service)

Key features for Data Meshes

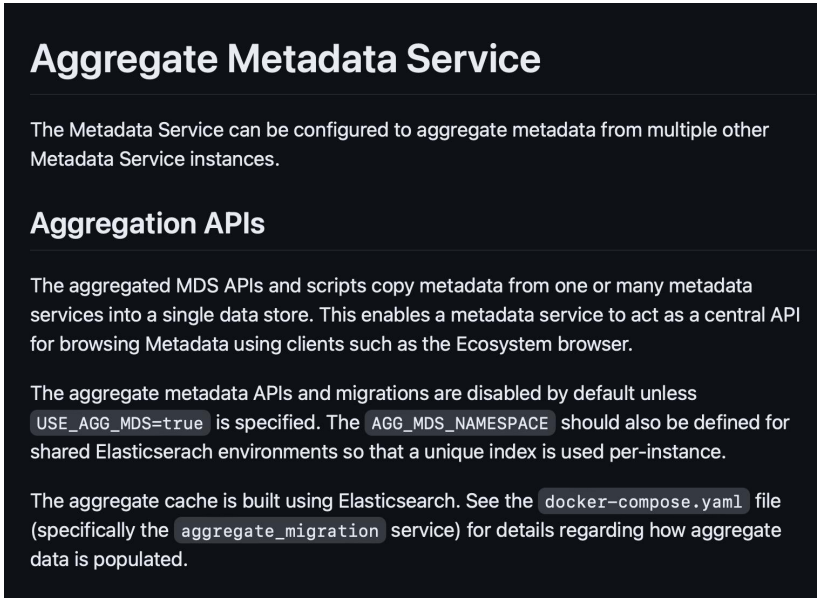
- To allow a Data mesh to connect and interact with data from multiple data commons we need to make sure the following are possible
  a. Fetch metadata from connected commons.
  b. Access appropriate studies
  c. View and provide authorization connected commons

Agenda

- Data Commons and Meshes
- Key Services and Jobs for Data Meshes
- **Aggregate Metadata Sync**
- Token Service (Workspace Token service)

Aggregate Metadata Sync

- **Aggregate Metadata Service**
- AggMDS sync job – copies metadata from multiple data commons into a single data store.
- New `/aggregate` endpoint of the Metadata Service.
- JSON based configuration defines information about
  - Data source and Data Adapter information
  - Normalizing data fields
  - Adding optional individual overrides
- Gen3 data adapter and adding new Custom Adapters.



Aggregate Metadata Service

The Metadata Service can be configured to aggregate metadata from multiple other Metadata Service instances.

Aggregation APIs

The aggregated MDS APIs and scripts copy metadata from one or many metadata services into a single data store. This enables a metadata service to act as a central API for browsing Metadata using clients such as the Ecosystem browser.

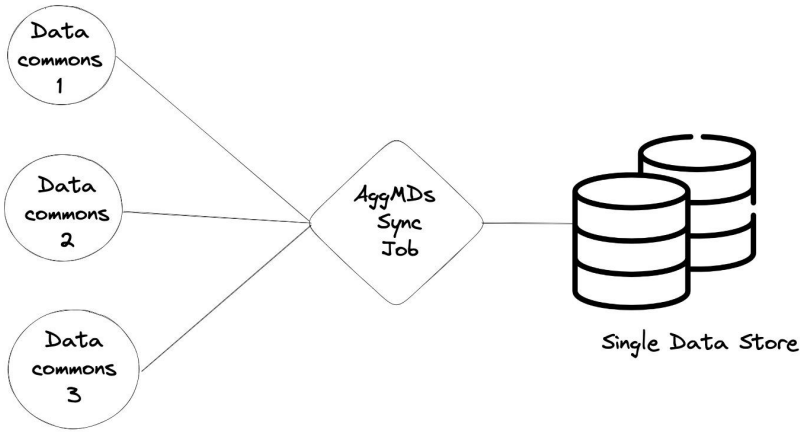
The aggregate metadata APIs and migrations are disabled by default unless `USE_AGG_MDS=true` is specified. The `AGG_MDS_NAMESPACE` should also be defined for shared Elasticserach environments so that a unique index is used per-instance.

The aggregate cache is built using Elasticsearch. See the `docker-compose.yaml` file (specifically the `aggregate_migration` service) for details regarding how aggregate data is populated.

Aggregate Metadata Service

Being able to fetch metadata from all the connected data commons.

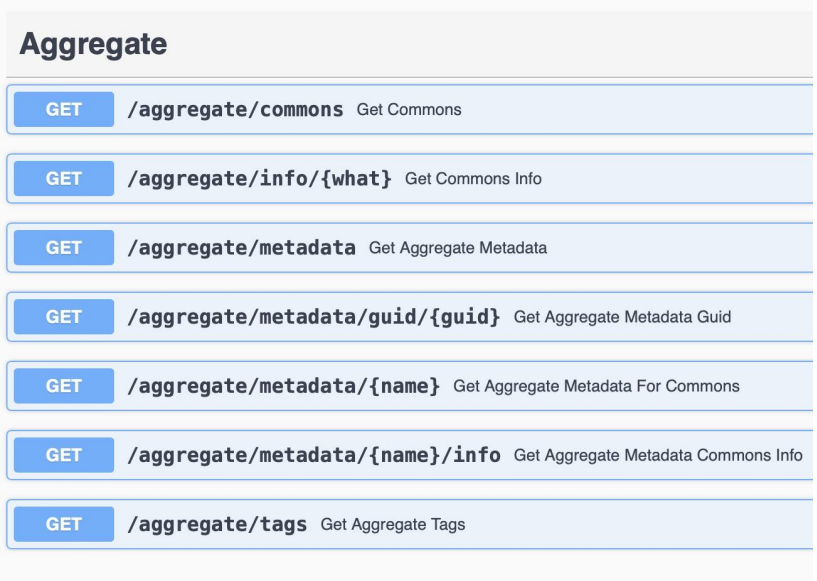
- Aggregate Metadata Service
- **AggMDS sync job – copies metadata from multiple data commons into a single data store.**
- New `/aggregate` endpoint of the Metadata Service.
- JSON based configuration defines information about
  - Data source and Data Adapter information
  - Normalizing data fields
  - Adding optional individual overrides
- Gen3 data adapter and adding new Custom Adapters.



Aggregate Metadata Sync Job

Being able to fetch metadata from all the connected data commons.

- Aggregate Metadata Service
- AggMDS sync job – copies metadata from multiple data commons into a single data store.
- **New `/aggregate` endpoint of the Metadata Service.**
- JSON based configuration defines information about
  - Data source and Data Adapter information
  - Normalizing data fields
  - Adding optional individual overrides
- Gen3 data adapter and adding new Custom Adapters.



`/aggregate` API endpoint

Being able to fetch metadata from all the connected data commons.

- Aggregate Metadata Service
- AggMDS sync job – copies metadata from multiple data commons into a single data store.
- New `/aggregate` endpoint of the Metadata Service.
- **JSON based configuration defines information about**
  - Data source and Data Adapter information
  - Normalizing data fields
  - Adding optional individual overrides
- Gen3 data adapter and adding new Custom Adapters.


```
"ICPSR": {
  "mds_url": "https://www.icpsr.umich.edu/icpsrweb/neutral/oai/studies",
  "commons_url": "https://www.icpsr.umich.edu",
  "adapter": "icpsr",
  "filters": {
    "study_ids": [30122, 37887, 37833, 37842, 37841, 35197 ]
  },
  "field_mappings" : {
      "tags": [],
      "sites": "",
      "year" : "2020",
      "shortName":"study_name",
      "location": "path:coverage[0]",
      "summary": {
            "path":"description",
            "filters": ["strip_html"],
            "default_value" : "N/A"
      },
      ...
    },
    "per_item_values" : {
      "10.3886/ICPSR30122.v5": {
        "__manifest": [
          {
            "md5sum": "7cf87ce47b91e3a663322222bc22d098",
            "file_name": "example1.zip",
            "file_size": 23334,
            "object_id": "dg.XXXX/208f4c52-771e-409a-c920-4bcba3c03c51",
            "commons_url": "externaldata.commons1.org"
          }
        ],
        "data_availability": "available",
        "authz": "/programs/open",
      },
      ...
    }
  }
```


Adapter Configuration documentation

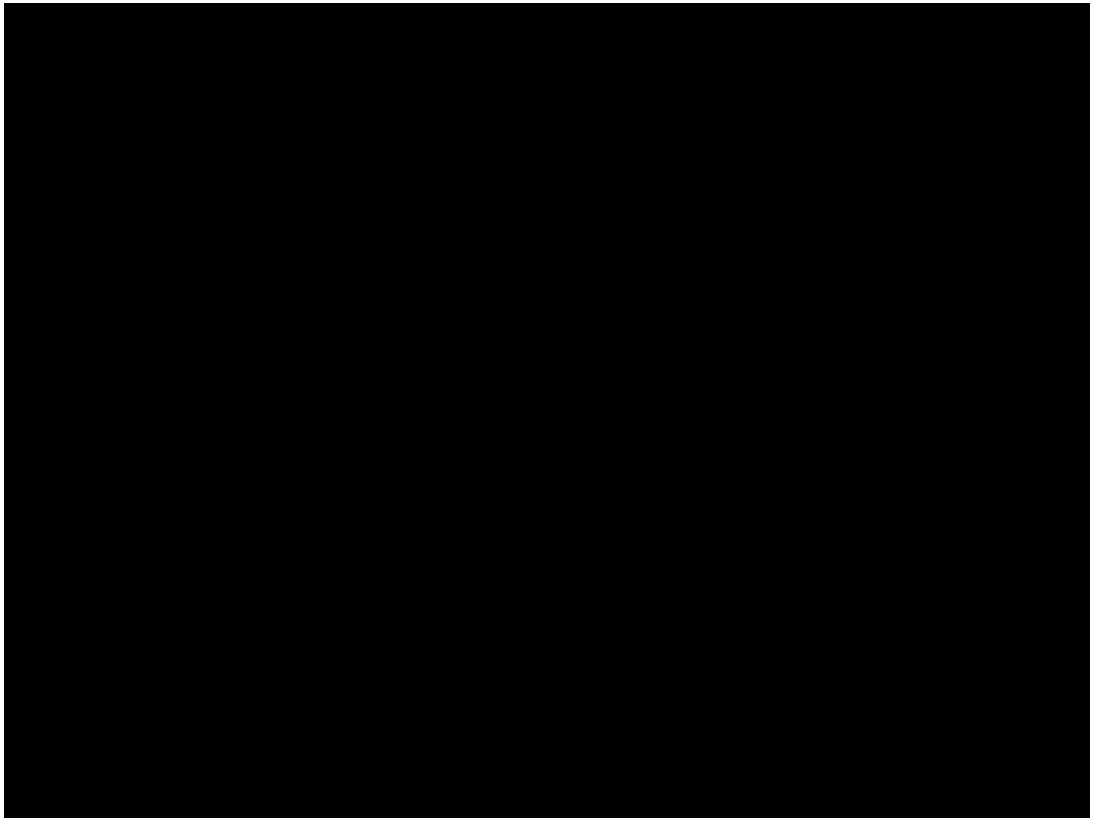
Being able to fetch metadata from all the connected data commons.

- Aggregate Metadata Service
- AggMDS sync job – copies metadata from multiple data commons into a single data store.
- New `/aggregate` endpoint of the Metadata Service.
- JSON based configuration defines information about
  - Data source and Data Adapter information
  - Normalizing data fields
  - Adding optional individual overrides
- **Gen3 data adapter and adding new Custom Adapters.**

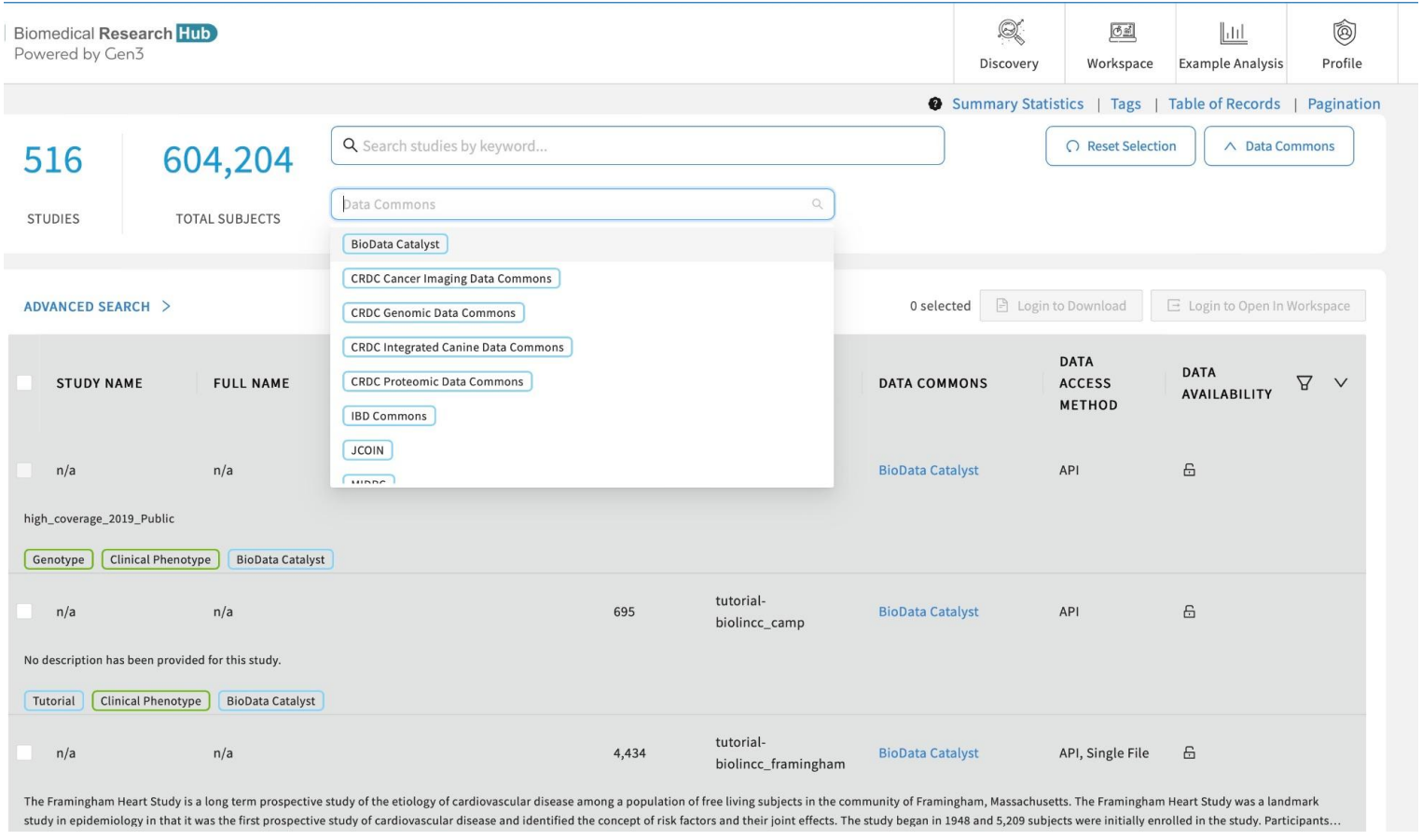
```python
class RemoteMetadataAdapter(ABC):

    @abstractmethod
    def getRemoteDataAsJson(self, **kwargs) -> Tuple[Dict, str]:
        """ needs to be implemented in derived class """

    @abstractmethod
    def normalizeToGen3MDSFields(self, data, **kwargs) -> Dict:
        """ needs to be implemented in derived class """

    @staticmethod
    def mapFields(item: dict, mappings: dict, global_filters: list = []) ·
      """
      maps fields from the remote field name to the normalized, or
      standardized version. Unless you need special processing this funct
      parameters:
        * item: metadata entry to be mapped
        * mappings: a dictionary of the remote field to normalize, this
                    passed in from the configuration file_name
        * global_filters to apply
      """

    @staticmethod
    def setPerItemValues(item: dict, perItemValues: dict) -> None:
      """
      Overrides the item field values with those in perItemsValues.
      parameters:
        * item: metadata entry to override
        * perItemValues: a dictionary of field names to values
      """
```

Custom Adapter documentation

Demo - AggMDS



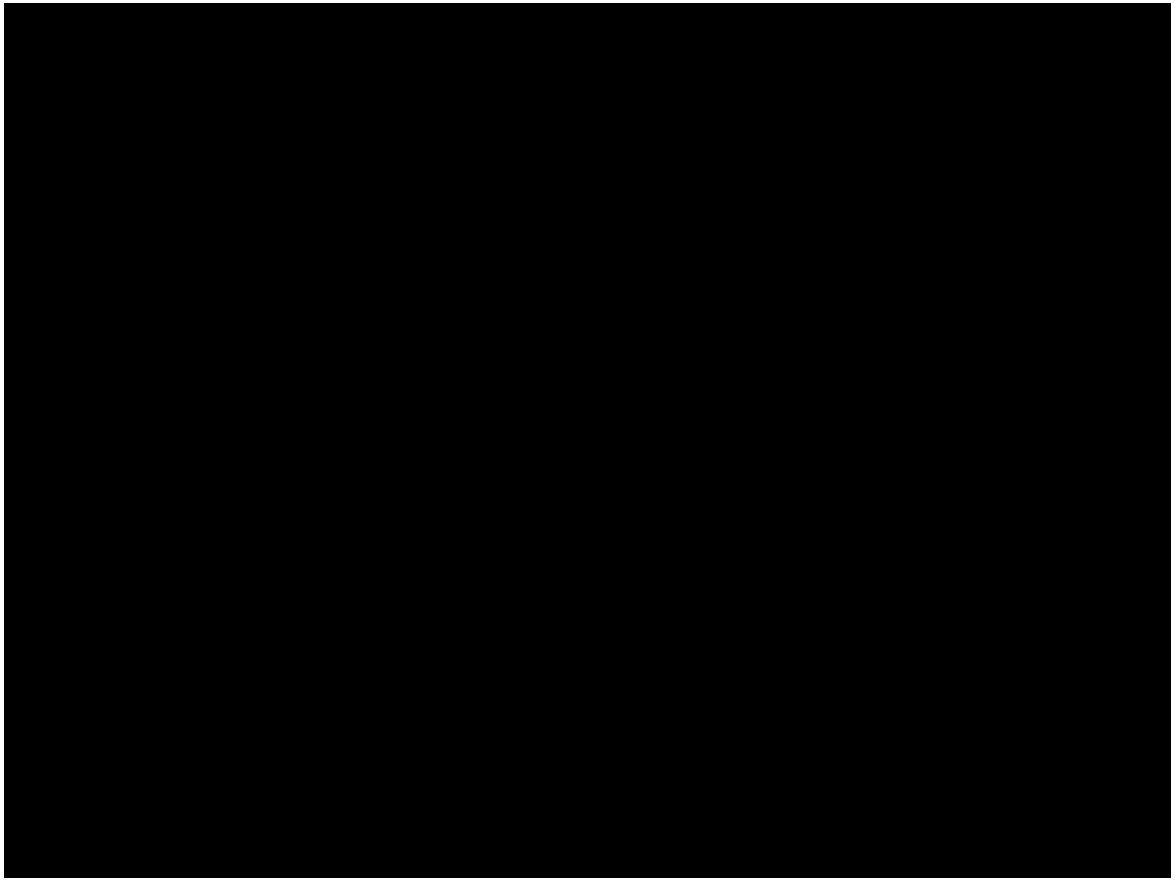
View & Search Studies on Discovery Page



Agenda

- Data Commons and Meshes
- Key Services and Jobs for Data Meshes
- Aggregate Metadata Sync
- **Token Service (Workspace Token service)**

Demo - Using WTS in a Data Mesh



Token Service (Workspace Token service)

- Acts as an additional OIDC client to interact with Fence on behalf of the user.
- Configuring additional data commons and updating **external_oidc** block.
    - Creating fence-clients on the external commons to generate **client-id** and **client-secret**
- New **/aggregate** endpoint of WTS allows users to fetch authz mapping of all the connected data commons.

Token Service (Workspace Token service)

Configuring additional data commons

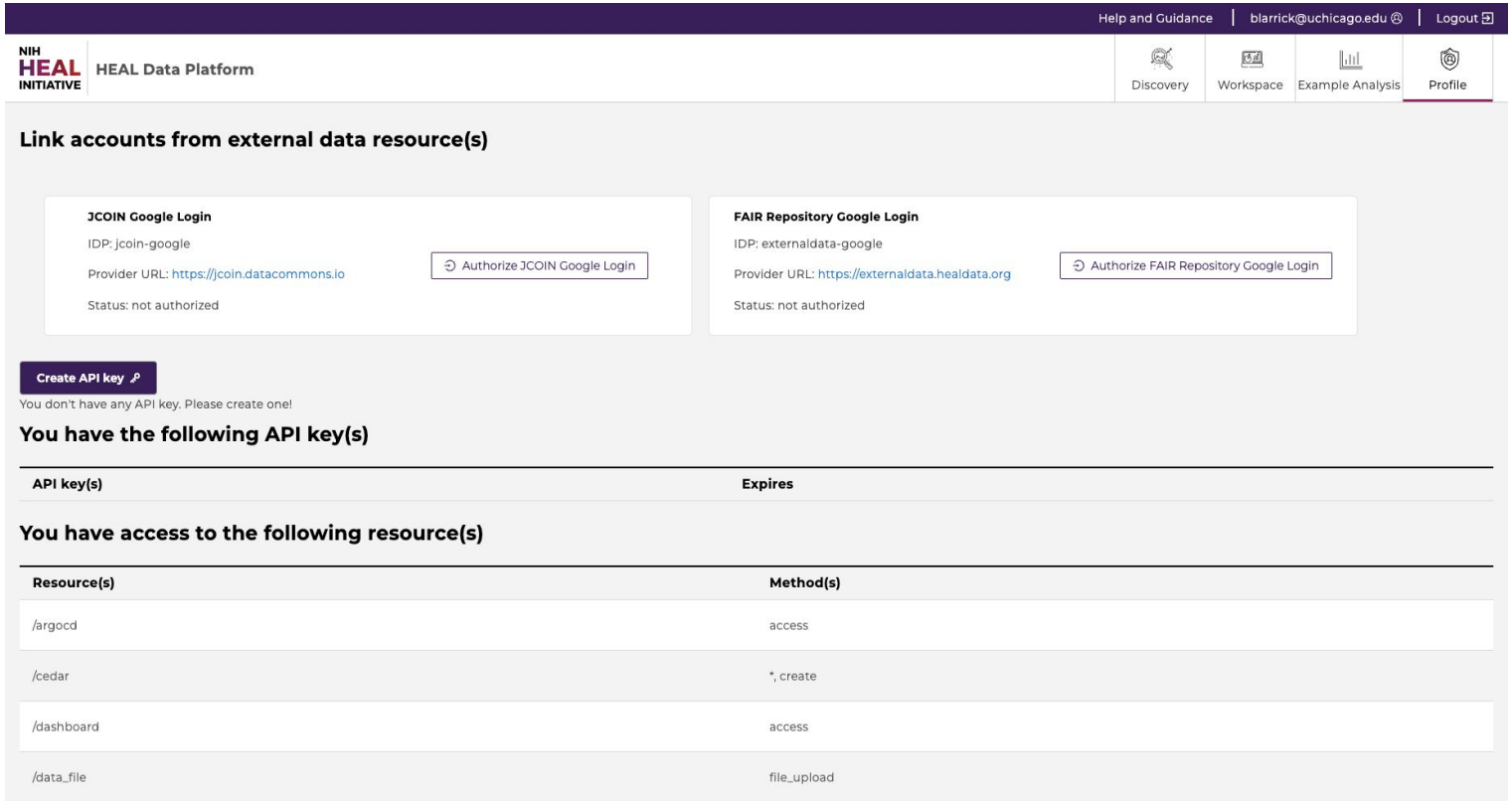
- **external_oidc** field in appcreds.json
- Create this WTS as a fence-client in the other commons' fence.
    - Store the **client-id** and **client-secret** in the **external_oidc** block.
- **GET** request to **/wts/external_oidc**

Token Service (Workspace Token service)

/aggregate endpoint of WTS

- For users to be able to access the studies based on the access control policies they have in the respective connected commons, we need to have a mechanism to fetch the access control policies of the user in all the connected commons.
- This can be achieved by fetching the response from **/authz/mapping** from each of the connected commons.
- To get this , one needs to add **/authz/mapping** in the **aggregate_endpoint_allowlist** in appcreds.json
- Then send a GET request to the **/wts/aggregate/authz/mapping** in the data mesh

Connect Profile to External Data Resources



Acknowledgements

- Gen3 Forum Steering Committee
  - Robert Grossman, Center for Translational Data Science, University of Chicago
  - Steven Manos, Australian BioCommons
  - Claire Rye, New Zealand eScience Infrastructure
  - Plamen Martinov, Open Commons Consortium
  - Michael Fitzsimons, Center for Translational Data Science, University of Chicago
- Speakers
  - Robert Grossman, Center for Translational Data Science, University of Chicago
  - Sai Narumanchi, Center for Translational Data Science, University of Chicago
  - Aarti Venkat, Center for Translational Data Science, University of Chicago
  - Phil Schumm, Biostatistics Laboratory, Department of Public Health, University of Chicago

Open Discussion

Topic Ideas for Gen3 Community Events