

# Active Research Data Management with the Django Globus Portal Framework

Nickolaus Saint  
nsaint@uchicago.edu  
University of Chicago  
USA

Jim Pruyne  
pruyne@globus.org  
University of Chicago  
USA

Michael E. Papka  
papka@anl.gov  
Argonne National Laboratory and  
University of Illinois Chicago  
USA

Ryan Chard  
rchard@anl.gov  
Argonne National Laboratory  
USA

Ben Blaiszik  
blaiszik@uchicago.edu  
University of Chicago  
USA

Kyle Chard  
chard@uchicago.edu  
University of Chicago  
USA

Rafael Vescovi  
ravescovi@anl.gov  
Argonne National Laboratory  
USA

Rachana Ananthakrishnan  
rachana@globus.org  
University of Chicago  
USA

Ian Foster  
foster@uchicago.edu  
Argonne National Laboratory and  
University of Chicago  
USA

## ABSTRACT

Publishing and sharing data is critical to fostering collaboration and advancing scientific research. Data portals are commonly used to organize, publish, and securely disseminate data—a critical step toward making data findable, accessible, interoperable, and reusable (FAIR). However, the diversity of scientific data types, sizes, and their location present significant challenges, e.g., it is difficult for portals to accommodate heterogeneous research products when using strict metadata schemas and rigid interfaces. Thus, there is a need for a user-customizable data portal solution that enables rapid creation of new portals that may be tailored to a researcher's needs while accommodating distributed data sources and engaging advanced computing resources. In this paper, we present the Django Globus Portal Framework (DGPF), a tool designed to help users rapidly create secure, customizable, and extensible data portals. DGPF is a powerful and flexible framework that builds upon the Globus platform for authentication, data sharing, creation of automation flows, and search capabilities, allowing for seamless integration with existing research workflows. We present the design and implementation of the DGPF and describe our experiences operating the Argonne Community Data Co-op (ACDC)—a collection of DGPF portals with over 1 M records and over 100 TB of published data that has been accessed by more than 300 users.

## KEYWORDS

Modern Research Data Portal, FAIR Data, Globus

### ACM Reference Format:

Nickolaus Saint, Ryan Chard, Rafael Vescovi, Jim Pruyne, Ben Blaiszik, Rachana Ananthakrishnan, Michael E. Papka, Kyle Chard, and Ian Foster.

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the United States government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

PEARC '23, July 23–27, 2023, Portland, OR, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-9985-2/23/07...\$15.00  
<https://doi.org/10.1145/3569951.3593597>

2023. Active Research Data Management with the Django Globus Portal Framework. In *Practice and Experience in Advanced Research Computing (PEARC '23)*, July 23–27, 2023, Portland, OR, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3569951.3593597>

## 1 INTRODUCTION

Sharing scientific data is an essential part of the scientific process and a key factor in driving progress. When scientists share their data, they allow other scientists to replicate their experiments, validate their results, and build upon their findings. This helps to reduce the likelihood of errors, increase transparency and credibility in scientific research, and foster collaboration. Further, sharing scientific data enables interdisciplinary research, where experts from different fields can work together to address complex problems.

Data portals play a crucial role in facilitating the sharing of scientific data and making it findable, accessible, interoperable, and reusable (FAIR). Data portals act as a nexus for organizing and disseminating data, incorporating advanced search and visualization tools that help users quickly and securely discover and access data, providing efficient methods to access and download data, and in some cases providing active components to manage downstream scientific tasks (e.g., reconstruction). Further, by standardizing the format of metadata, portals can ensure data quality and compatibility, and provide a permanent and secure repository to preserve valuable data.

In prior work, we presented the Modern Research Data Portal [8] (MRDP)—a design pattern for providing secure, scalable, and high performance access to research data. The central ideas of the MRDP pattern are 1) that control logic is decoupled from data storage; and 2) crucial functionality (e.g., data transfer) is outsourced to reliable and performant cloud-hosted services. Collectively, these two approaches increase performance and reduce development and operations costs. The reference implementation of the MRDP offered a simple web-based user interface for researchers to share, discover, and access scientific data. It also used Globus [11] services for data transfer, user management, and access control. The MRDP is designed to support collaborative research communities, enabling

researchers to share data with other members of their community, regardless of their physical location, type or size of data, or the systems they use.

In this paper we revisit the MRDP pattern we first proposed in 2018 and review our experiences applying it to various data sharing scenarios. We describe our need to implement a more active and customizable reference implementation, integrating cloud-hosted search capabilities, on-demand computation, and sophisticated research flows. We implement this new instance of the MRDP pattern in the Django framework, the Django Globus Portal Framework (DGPF). DGPF is designed to facilitate the rapid deployment of project-specific data portals that allow sharing of scientific data in a secure, scalable, and user-friendly manner. The DGPF is publicly available on GitHub [30].

In the remainder of this paper we review the needs of the scientific community in §2, before presenting the design and implementation of the DGPF in §3 and its integration with Globus services. We describe how the DGPF is used in §4 and then, in §5, showcase the capabilities of the DGPF by reviewing several real-world implementations of DGPF portals, with particular focus on the Argonne Community Data Co-Op (ACDC)—a collection of DGPF portals currently operating for a diverse range of Argonne projects. We present related work and conclusions in §6 and §7.

## 2 RESEARCH DATA PORTAL REQUIREMENTS

There have been many implementations of the MRDP pattern since it was first presented in 2018 [13, 22, 32]. Reviewing these portals has highlighted the key roles and requirements of research data portals and our experiences have identified several new requirements, which we describe below.

**Discoverable:** The primary role of a data portal is to make data *searchable* and *discoverable*. This is typically achieved by indexing metadata into a database or catalog that facilitates search and allows for annotation. However, scientific data comes in many forms and sizes, with useful metadata that may be embedded within the file contents, filename, or even data types. It is essential to be able to capture this wide array of data and metadata in order to facilitate efficient search across heterogeneous datasets.

**Customizable:** We have found that different portals have very different requirements. As such it is important that portals can be *customized* and *extended* to enable users to configure what and how data and metadata are displayed to the user. For example, a developer may choose to present data in ways that best provide insight into quality and accuracy of the results. Images depicting some aspect of the dataset are a common approach to conveying information about the data. Further, many research portals require customizable features, such as search facets, to rapidly filter through results on well known dimensions, while others may need visualization tools and embed interactive capabilities to explore datasets.

**Active data management:** The rapid growth of data-generation capabilities presents both a challenge and opportunity for data portals. Large scientific experiments can generate data at extreme rates and volumes, making it impossible for a user to manually process every dataset, and even impractical to act on many data using local resources. Indeed, data portals provide a mechanism to catalog large datasets, but can also serve as a central interface to access

and act on distributed data. With appropriate integrations, a portal can transcend discovery and be used to orchestrate, manage, and perform analysis, for example.

**Fine-grained access control:** Research data portals require multi-faceted *security* mechanisms that support precise and reliable access control. Lapses in security can be catastrophic if private data are inappropriately shared. Thus it is critical that data are only discoverable and accessible to authorized users. This is further complicated by the need to manage access control at a per-record level, enabling users to share specific data, records, or entire catalogs in a transparent and understandable way.

From these requirements we propose extending the MRDP pattern in a new data portal framework. Our framework includes the necessary packages to quickly and securely create and deploy a MRDP.

## 3 DJANGO GLOBUS PORTAL FRAMEWORK

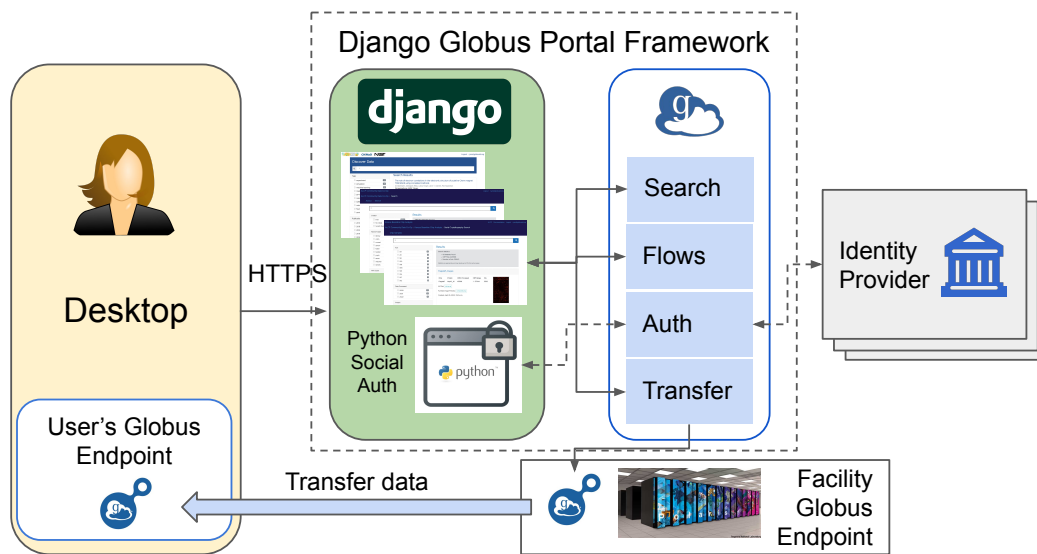
We designed the DGPF architecture, depicted in Figure 1, to satisfy the requirements described in §2. DGPF combines a Django Web framework front-end with a Globus back-end, providing a highly customizable user experience with high performance data management capabilities to securely visualize and act upon large and distributed data (accessible via Globus). The DGPF, as a Django-based framework can be easily deployed and extended, and builds upon well understood and trusted authentication fabric. Here we describe the DGPF and the building blocks upon which it is implemented.

### 3.1 Django Framework

The Django Framework provides a number of strengths which make it attractive for building data portals. Most notably, the built-in Django ORM [18], which provides an abstraction for database models, and the Django Template Language, which enables straightforward customization of HTML web template pages in a Django application. Further, Django has an extensive ecosystem of packages that can be integrated to add new functionality.

Our implementation depends on the Python Social Auth [3] package to implement OAuth2 [21] and Open ID Connect [26] specifications for many authentication back-ends, including Globus [31]. Python Social Auth defines database models for storing user tokens and ties into the standard Django User Model. This level of abstraction is ideal for many applications, which may not require knowledge of the underlying authentication mechanisms, but contain sensitive views or functionality that can only be accessed by authorized users. This integration allows both for the incorporation of Globus Services through the availability of Globus Access tokens, but also provides the high level authorization mechanisms within Django, which can be assumed and used by third party package developers.

DGPF bundles a handful of Django templates to make the process of starting a new portal fast and easy. The Django Template Language enables DGPF templates to be overridden and customized to suit the needs of the specific data portal and use cases.



**Figure 1: An overview of the DGPf architecture.** Users interact with DGPf via HTTPS requests to a Django web service. Python Social Auth communicates and Globus Auth to allow users to authenticate using existing identity providers. Portal records are populated from Globus Search and results can optionally be acted upon using other Globus services, such as third-party transfer via Globus Transfer.

### 3.2 Globus

Globus is a cloud-hosted research data management platform that, among its many capabilities, supports high-performance and secure data movement, rich data search, and flexible identity and access management. Globus implements a comprehensive security fabric that supports authentication via hundreds of existing identity providers and facilitates secure access to both data storage and compute resources as well as external resources. This coupling allows DGPf portals to seamlessly integrate with research computing infrastructures, enable secure access to data stores, and reuse well-known identities and authentication mechanisms.

**3.2.1 Search.** Globus Search [1] is a cloud-hosted service that allows users to create, populate, and curate indices of searchable metadata. It implements a rich, full-text search model, offering various common search patterns (e.g., wildcard, range queries, search facets). Globus Search implements fine-grained security model in which each record (i.e., set of metadata associated with an entity) has associated visibility permissions that can be restricted to arbitrary users or groups [9]. These capabilities are ingrained in the Globus Search APIs, providing visibility-filtered query results that restrict data discoverability to only authorized users.

Globus Search is built on Elasticsearch [20] and therefore supports a broad query language that facilitates a wide range of performant, free text and structured queries. Each Search index can have up to 1000 statically typed keys, or distinct fields within a metadata record, which can be used when performing searches. This static typing allows queries to encompass date ranges, free text, and other query properties. There is no explicit schema that an index must follow, instead, the index's schema is dynamically created as new data are added. DGPf uses Globus Search as the means to store and

organize metadata. Each data portal has an associated Search index, where any data ingested into the index are then rendered through the portal.

**3.2.2 Globus Auth.** Globus Auth is an identity and access management system which brokers authentication and authorization between end users, identity providers, and Globus Services. It replaces the need to track usernames and passwords for each service and allows for interoperability with existing identity management systems.

When a user logs into a DGPf portal they are taken through an authentication flow (OAuth2) through a selected identity provider (e.g., an institution, ORCID, Google). Once redirected back to the portal, access tokens can then be requested on behalf of the user. These access tokens are used to authorize actions on other external services (e.g., Globus Search, Globus Transfer). Access tokens are issued by Globus Auth and can be stored for the duration of the user's session, until they expire, or until the user revokes them via logout.

A Globus Access token, when introspected by a service such as Globus Search, is used to derive information about the user and what their consents are for requested resources. When a user performs a query for confidential information within a search index, Globus Search uses the access token and Globus Auth to determine whether the user is permitted to access the results.

The OAuth architecture provides flexibility in its design. Access control decisions are outsourced to Globus services and can be managed externally through Globus Auth and Groups. Portal developers need only to obtain service tokens for their preferred service and Globus Auth will determine authorization for a user's requested resources.

**3.2.3 Globus Transfer.** Globus Transfer [2] allows users to access and transfer large amounts of data quickly and securely. It provides an intuitive web interface and REST API for users easily to initiate transfers, monitor their progress, and view transfer histories. Under the covers, it maximizes transfer speed by leveraging high-performance networking infrastructure, such as dedicated wide-area networks and high-speed data transfer nodes.

Integrating Globus Transfer into DGPF allows users to act on remote datasets (e.g., to download them, move them directly to an analysis machine, or archive them). Globus implements a unique data authorization model via which access permissions can be associated with remotely stored data. Thus, DGPF can manage access to even remote data to meet user needs. A common practice is to grant public access to metadata about records within Globus Search (such that they can be discovered), but constrain access to the data on a Globus Collection to a set of authorized users. This means that while metadata records are discoverable, access to the underlying data requires authentication, which can be requested within the portal.

**3.2.4 Automation Services.** Globus Automation Services [12] provide the capabilities to process data from scientific instruments and describe tools that enable convenient specification of high-level *flows* that combine diverse actions with a flexible mapping onto diverse physical resources to meet reliability, scalability, timeliness, and security goals as an experiment runs. Flows are composed by linking distinct actions, or steps, into a pipeline of data management and analysis tasks. A common application of these flows is to perform analysis steps and then publish metadata into a search index. This allows for the end-to-end automation, from data capture and analysis, through to the extraction of metadata and publication of the data to a portal.

### 3.3 Plugins

Here we outline exemplar integrations we provide as Django plugins that can be incorporated into a DGPF portal to meet the diverse needs of scientific use cases.

**Token-based HTTPS rendering:** A common feature for enriching content within a portal is to embed images and files. Rather than requiring these data be present on the portal itself, we leverage Globus's ability to retrieve content over HTTPS. This allows media such as images or raw text to be displayed in search results where it is contextually most helpful. DGPF provides this functionality by employing Globus Auth tokens when requesting remote data, allowing data to be retrieved while respecting authentication by restricting data to users granted access through Globus Transfer ACLs.

**Globus Flows:** It is often beneficial to be able to perform a curated analysis task on a dataset discovered through a portal. Integrating Globus Flows into a portal allows the user to act on data in a predefined way. For example, a common requirement is to reprocess a dataset using a new set of input parameters. Reprocessing refers to querying data already cataloged within a portal, selecting input parameters, and re-running it through an analysis flow to produce a new dataset.

**Globus Transfer:** DGPF portals can catalog millions of entries, many of which may reflect files available through Globus Transfer. We provide a plugin to include Globus Transfer accessibility within a portal, making it easy to reference and act on individual files or entire collections of data through the portal. The Transfer integration allows users to select entries within a portal and initiate a transfer to, if permitted, move those data to another Globus endpoint.

## 4 USING THE DGPF

The manner in which a user engages with DGPF depends on their desired action, from using the portal to discover data, to publishing and reprocessing datasets, or establishing and deploying a new portal. Here we describe how users engage with DGPF portals in these different contexts.

### 4.1 DGPF Views

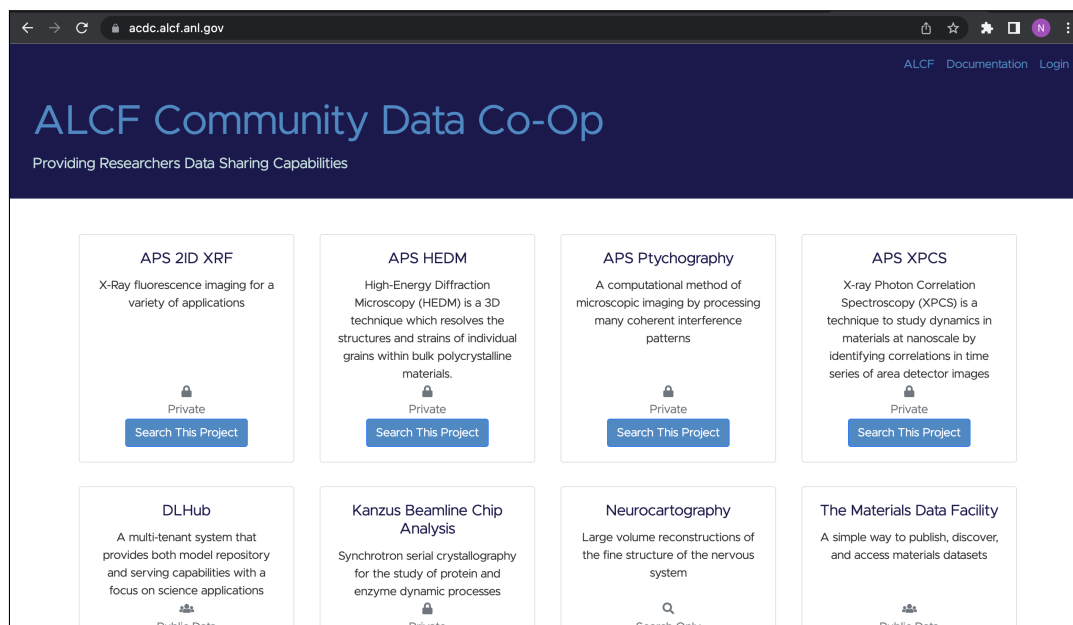
The DGPF package supports three principal search views: Index selection, general search, and a detailed record view. Custom plugins, such as processing through Globus Flows, often have an additional view to act on a collection of data. As illustrated in Figure 2, the *index selection* view supports selection from among multiple Globus Search indices. On selecting an index, the general search view is presented to the user, as shown in Figure 3. This provides access to the catalog's query and faceting features. The results of a user query are presented with a customizable list of search results. Selecting a specific result brings up the record's detail view, as shown in Figure 4.

The interface for the search view translates the user's query into the request to Globus Search. User queries are captured in each URL, and can be shared with other users. Each DGPF portal will have different faceted information reflecting the given data and custom templates to display search results and record details. The details view can be customized extensively, requiring the developer to carefully consider how best to structure a record's metadata. Additionally, developers may also manipulate the styling of the core templates to brand all pages.

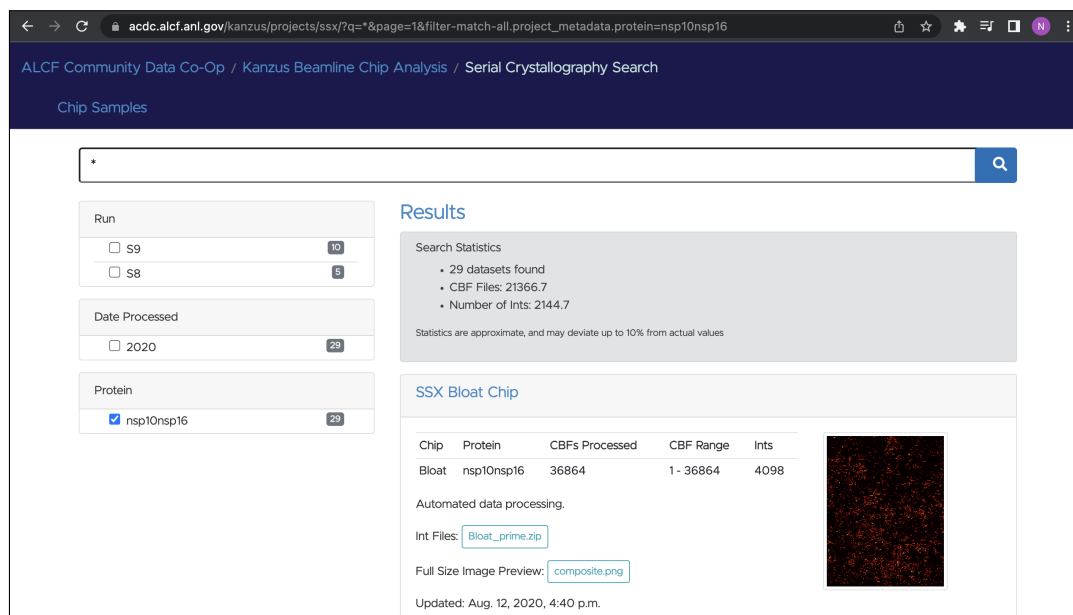
### 4.2 Publishing, Browsing, and Acting on Records

Records can be published into Globus Search through two approaches: batch, or online. In the batch model, existing data can be cataloged and inserted into a Search record as a one-off process. This involves parsing a filesystem or Globus Collection (see Xtract [28]) and inserting a record for each dataset. The online process adds records to an index as they are generated. This is typically achieved in DGPF portals using Globus Flows for active publication of data as an experiment takes place. Online publication comes with a number of advantages, including a rapid turn-around time for experiments and the option to drive the decision making process of the experiment itself. Publication usually consists of transferring data into a Globus Collection and ingesting metadata about the files into Globus Search. From there, records can then be viewable by users.

Once a portal is populated a user may authenticate and interact with the records. Once authorized, the user is able to issue search



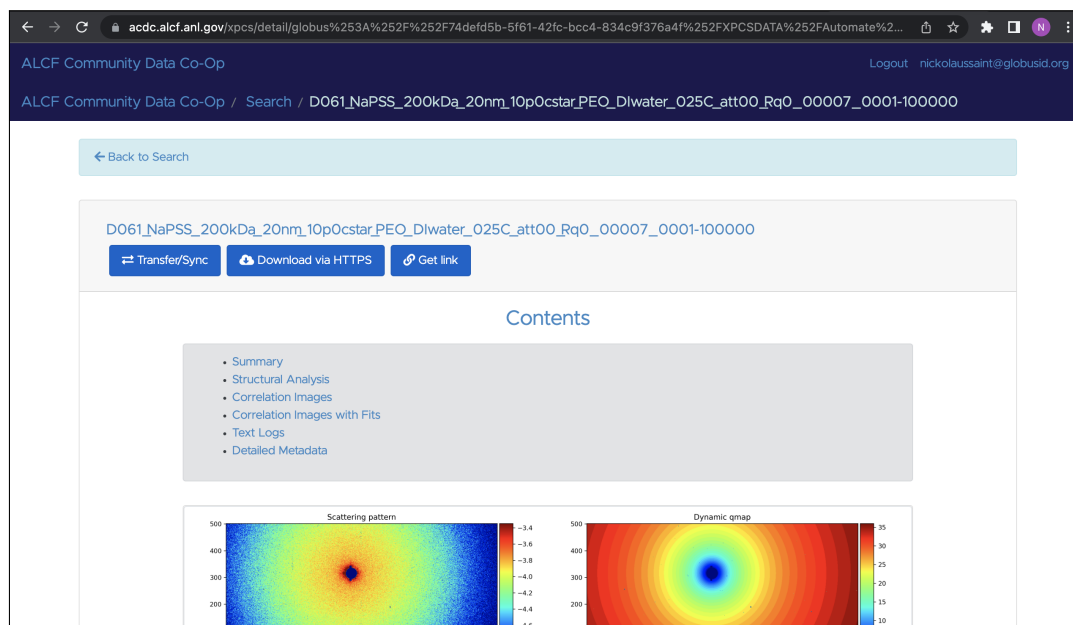
**Figure 2: The index selection view of the Argonne Community Data Co-op. Users can select an available portal to search within. Visible portals are restricted via access control lists, so users can only select portals which they are permitted to access.**



**Figure 3: The default search view of the Serial Crystallography portal. The search bar is shown at the top and relevant facets can be selected on the left. Results from the search are listed on the right. This portal includes a figure depicting the sample's collection alongside each record.**

queries into the search bar, filtering results via facets, and inspecting specific records. While the user may have access to the portal, they do not necessarily have access to all the portal's contents. Each entry has its own access control, permitting discoverability and access to specific users, groups, or public.

When enabled with Globus Flows integration, a portal can be used to initiate a flow to act on selected data. This capability is typically used to reprocess data that already resides within the portal. Using a custom view users can specify new analysis configurations or input variables and tag the runs with a custom name so as to



**Figure 4: The detailed record view in the XPCS portal. Applicable actions, such as downloading or transferring the associated data, are available as buttons at the top of the page. A table of contents allows the user to navigate to relevant metadata and figures are prominently included to provide context to the data.**

distinguish the results from existing records. Once a flow is initiated it will generally transfer the data from the location it is stored to an intermediary compute resource to perform analysis. Metadata are then extracted from the newly created results and published back into the portal with a new tag to identify them.

### 4.3 Portal Development

The DGPF package provides a template to quickly bootstrap new portals. The template includes authentication, a set of simple search views, and templates to render a simple search interface. Before deploying a new DGPF portal a developer must first create a new Globus Auth client to manage authentication and authorization, and a Globus Search index to store data for the portal. This gives users a working portal out-of-the-box that can be modified to include existing components to fit their use case.

Deploying DGPF portals is similar to other Python WSGI applications. DGPF can be deployed with any number of popular web service tools, such as Apache or NGINX. The outsourcing of much functionality to external Globus services means the portal requires very few resources.

When deploying a new portal the administrator must select an appropriate database and credential storage mechanism. DGPF does not restrict database selection, any Django-supported database may be used. The database is primarily used for storing temporary user information, such as Globus access tokens and steps must be taken to ensure it is kept secure. The second consideration relates to the storage of the Globus App client credentials and Django security key. The Globus App credentials are used during the Globus Auth flow when performing user login, and are typically stored outside the database in a configuration file or as environment variables.

The Django Secret Key is used for cryptographic signing of user sessions and ensures users are logged in between requests.

The process to create a new portal is described extensively in the DGPF documentation [16]. The process is also encoded in the DGPF Cookiecutter [14], providing a recipe that installs and walks through the development of a DGPF portal.

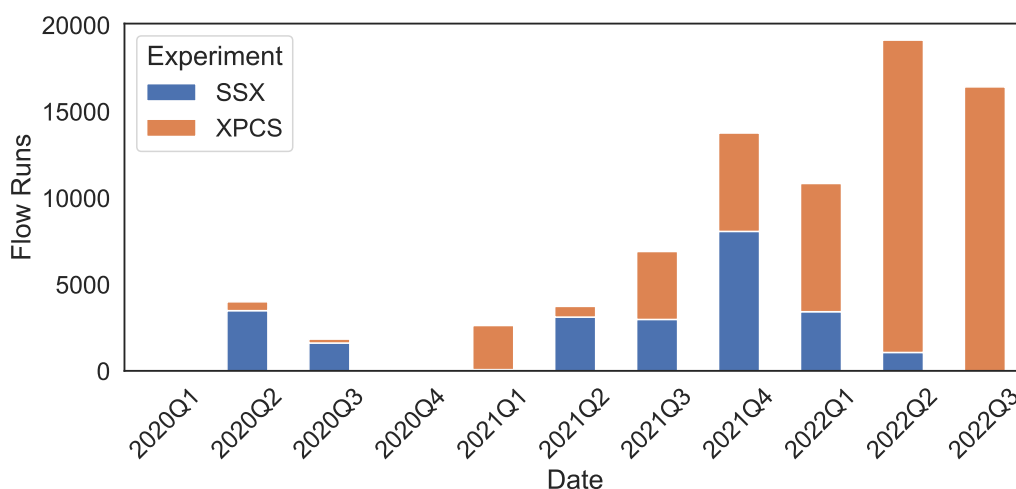
## 5 ALCF COMMUNITY DATA CO-OP

The Argonne Community Data Co-Op (ACDC) is a collection of DGPf portals designed to support collaborative research communities by providing a secure and scalable infrastructure for publishing and sharing scientific data. ACDC currently supports 11 distinct portals and serves 340 users; these portals currently catalog 1.05 million records that comprise over 120 TB of indexed data. Publications to these portals are growing steadily, as depicted in Figure 5, which shows flow executions that have published data to two of the portals (XPCS and SSX) since 2020. Each flow added or updated information in their respective portal, with XPCS flows typically adding entirely new records and SSX flows updating the record for the sample currently being processed. Overall, more than 80 000 flows have been run by these two use cases.

Here we discuss our experiences developing, deploying, and operating a collection of DGPF portals for the ACDC.

## 5.1 Deployment and Usage

ACDC is deployed on an Amazon EC2 instance. It uses NGINX to handle requests and uWSGI as the interface for processing the requests via Python. uWSGI is run as a separate linux service. This



**Figure 5: Number of Globus Flows run by the SSX and XPCS portals over time for the purpose of data publication.**

setup allows NGINX to serve static files while handing off dynamic requests to uWSGI.

The code for various portals served within ACDC are managed as separate Github repositories. Each repository includes a Python package with a Django application. These applications can be run standalone for local testing. Upon request, a portal repository can be included into the collection of ACDC portals.

## 5.2 ACDC Portals

**The Materials Data Facility (MDF)** [4, 5] provides a simple way to publish, discover, and access materials datasets. A key aim of MDF is to build data services and portals for materials researchers that promote open data sharing, simple publication, coupled with powerful data discovery interfaces to encourage data discovery and reuse. The MDF portal is available within the ACDC, and contains over 650 materials datasets, consisting of over 80 TB of data. Users can self-publish datasets into the MDF and manage fine-grained access control over their records to facilitate sharing.

The MDF highlights two clear needs relating to security and customization. Data published to the MDF require strict enforcement of access control, along with the flexibility to share specific records, collections of data, or entire catalogs. The MDF also implements custom branding, requiring the portal to accommodate site-wide templates.

**X-ray photon correlation spectroscopy (XPCS)** [23] is an experimental technique used at Argonne’s Advanced Photon Source (APS) to study dynamics in materials at nanoscale by identifying correlations in time series of area detector images. The APS operates as a user facility where users submit a proposal and, if successful, are allocated resources and expertise to perform their experiment. The APS’s Data Management system (DM) [33] can be used during data collection to manage data storage and create a Globus Group consisting of the users performing the experiment and initiate analysis flows. This flow accepts the Globus Group and data files as

input and performs analysis at Argonne’s Leadership Computing Facility before publishing results, with group access control, to the XPCS portal.

Once the data are in the portal, a secondary mode of processing can be applied to reprocess data using the Globus Flows plugin. Here, the general search functionality is used to collect and filter datasets in bulk, and new input parameters are selected. A new flow is started for each dataset and results are published to the portal with the new metadata label.

Our experiences with XPCS highlighted the need for customization. The XPCS portal has undergone a number of iterations to optimize facets, prioritize figures, and enable users to select (and remember) which images should be visible in the detailed record view.

**Synchrotron serial crystallography (SSX)** [27] enables the study of protein and enzyme dynamic processes with time resolution with light activation, very low X-ray dose, and at room temperature. SSX relies on rapidly imaging small crystal samples 1–2 orders of magnitude faster than traditional crystallography techniques, gathering data at rates that necessitate the use of HPC resources for data processing and analysis.

Serial Crystallography data are collected in the form of many small images, consisting of tens to hundreds of thousands of images per sample and generated at 10–100 Hz. Due to the quantity of data, each analysis flow is designed to act on a batch of 256 images. The SSX flows process these data and compute statistics of for the current sample. These statistics, results, and an overview figure are then published to the portal. The portal is used by beamline scientist to actively monitor samples and determine when sufficient data have been collected and a new sample can be analyzed. Similarly, if the current sample is performing poorly it can be replaced.

The SSX portal exemplifies the need for interactivity. Using the portal as a means of monitoring an ongoing experiment requires that it be responsive as new data are made available. This use case

is common for ACDC portals and makes the portal central to the active research process.

## 6 RELATED WORK

Numerous platforms have been developed with the purpose of encouraging data sharing and publication [10, 25]. Dataverse [15] provides a way to publish scientific data, increase visibility, and associate persistent identifiers. Zonodo [17] is another product that allows researchers to publish research artifacts, including papers, datasets, and software. Both Dataverse and Zonodo implement the traditional data portal model in which data and catalog are co-located, and thus suffer from an inability to exploit high performance external data storage services or to outsource functionality to robust cloud-hosted services.

The Common Fund Data Ecosystem [7] consists of datasets from several different NIH Common Fund projects. It is similar to ACDC in that it serves as a location to publish datasets from different organizations. However, CFDE requires strict structure to ensure datasets in the catalogue match the pre-defined schemas. FaceBase [6] is a data repository designed to collect and compile the biological phenomena to construct the human face and derive the causes of common disorders.

Scientific gateways are a common approach to combining data and compute and simplifying the user experience for acting on large datasets. Apache Airavata [24] is a middleware framework for developing scientific gateways that can compose, manage, and execute applications and workflows. Galaxy [19] provides tools for users to construct workflows using a graphical interface and then apply them to available datasets. Tapis [29] is another example of a scientific platform that aims to simplify using research computing resources and managing scientific data.

DGPF differs from existing data services by providing a lightweight tool that integrates the broad range of Globus services and streamlines data publication and processing. This configuration fits both smaller projects that need to be rapidly deployed, as well as larger projects that provide custom integrations with Globus.

## 7 SUMMARY

Making data findable, accessible, interoperable, and reusable (FAIR) is crucial to the scientific method. Data portals play a key role in making data discoverable and securely sharing them between collaborators and the wider community. Here we presented the Django Globus Portal Framework—an implementation of the MRDP designed to accelerate the development of scientific data portals. The DGPF leverages Globus services to facilitate secure data access across distributed resources and customizable search indexes that make vast quantities of metadata searchable. We discussed features of DGPF portals and described the Argonne Community Data Co-op and three of its real-world portals. The ACDC consists of 11 portals with over 120 TB of published data. More than 340 users have engaged with ACDC portals. In future work we plan to continue developing plugins to facilitate diverse deployment use cases. We also aim to explore serving results from multiple indexes within an individual portal, enabling users to publish related data across indexes without hindering discoverability.

## ACKNOWLEDGMENTS

This work was supported in part by NSF grants OAC-1835890 and OAC-2004894; award 70NANB14H012 from the U.S. Department of Commerce, National Institute of Standards and Technology as part of the Center for Hierarchical Material Design (CHiMaD); and by the U.S. Department of Energy under Contract DE-AC02-06CH11357, including by the Office of Advanced Scientific Computing Research's Braid project. The research used Argonne Leadership Computing Facility resources.

## REFERENCES

- [1] Rachana Ananthakrishnan, Ben Blaiszik, Kyle Chard, Ryan Chard, Brendan McCollam, Jim Pruyne, Stephen Rosen, Steven Tuecke, and Ian Foster. 2018. Globus Platform Services for Data Publication. In *Practice and Experience on Advanced Research Computing* (Pittsburgh, PA, USA) (PEARC '18). ACM, New York, NY, USA, Article 14, 7 pages.
- [2] Rachana Ananthakrishnan, Kyle Chard, Ian Foster, and Steven Tuecke. 2015. Globus platform-as-a-service for collaborative science applications. *Concurrency and Computation: Practice and Experience* 27, 2 (2015), 290–305.
- [3] Python Social Auth. 2023. *Python Social Auth*. Retrieved March 2, 2023 from <https://python-social-auth.readthedocs.io/en/latest/>
- [4] B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, and I. Foster. 2016. The Materials Data Facility: Data Services to Advance Materials Science Research. *JOM* 68, 8 (July 2016), 2045–2052. <https://doi.org/10.1007/s11837-016-2001-3>
- [5] Ben Blaiszik, Logan Ward, Marcus Schwarting, Jonathon Gaff, Ryan Chard, Daniel Pike, Kyle Chard, and Ian Foster. 2019. A data ecosystem to support machine learning in materials science. *MRS Communications* 9, 4 (2019), 1125–1133.
- [6] James F Brinkley, Shannon Fisher, Matthew P Harris, Greg Holmes, Joan E Hooper, Ethlyn Wang Jabs, Kenneth L Jones, Carl Kesselman, Ophir D Klein, Richard L Maas, et al. 2016. The FaceBase Consortium: a comprehensive resource for craniofacial researchers. *Development* 143, 14 (2016), 2677–2688.
- [7] Amanda L Charbonneau, Arthur Brady, Karl Czajkowski, Jain Aluvathingal, Saranya Canchi, Robert Carter, Kyle Chard, Daniel JB Clarke, Jonathan Crabtree, Heather H Creasy, et al. 2022. Making Common Fund data more findable: catalyzing a data ecosystem. *GigaScience* 11 (2022).
- [8] Kyle Chard, Eli Dart, Ian Foster, David Shifflett, Steven Tuecke, and Jason Williams. 2018. The Modern Research Data Portal: A design pattern for networked, data-intensive science. *PeerJ Computer Science* 4 (2018), e144.
- [9] Kyle Chard, Mattias Lidman, Brendan McCollam, Josh Bryan, Rachana Ananthakrishnan, Steven Tuecke, and Ian Foster. 2016. Globus Nexus: A Platform-as-a-Service provider of research identity, profile, and group management. *Future Generation Computer Systems* 56 (2016), 571–583. <https://doi.org/10.1016/j.future.2015.09.006>
- [10] Kyle Chard, Jim Pruyne, Ben Blaiszik, Rachana Ananthakrishnan, Steven Tuecke, and Ian Foster. 2015. Globus data publication as a service: Lowering barriers to reproducible science. In *2015 IEEE 11th International Conference on e-Science*. IEEE, 401–410.
- [11] K. Chard, S. Tuecke, and I. Foster. 2014. Efficient and Secure Transfer, Synchronization, and Sharing of Big Data. *IEEE Cloud Computing* 1, 3 (2014), 46–55.
- [12] Ryan Chard, Jim Pruyne, Kurt McKee, Josh Bryan, Brigitte Raumann, Rachana Ananthakrishnan, Kyle Chard, and Ian T Foster. 2023. Globus automation services: Research process automation across the space–time continuum. *Future Generation Computer Systems* (2023).
- [13] LSST Dark Energy Science Collaboration. 2023. *LSSTDESC Data Portal*. Retrieved March 2, 2023 from <https://data.lsstdesc.org/>
- [14] Django Globus App Cookiecutter. 2023. *Django Globus App Cookiecutter*. Retrieved March 2, 2023 from <https://github.com/globus/cookiecutter-django-globus-app>
- [15] Mercè Crosas. 2011. The dataverse network: an open-source application for sharing, discovering and preserving data. *D-lib Magazine* 17, 1/2 (2011).
- [16] Django Globus Portal Framework Documentation. 2023. *Django Globus Portal Framework Documentation*. Retrieved March 2, 2023 from <https://django-globus-portal-framework.readthedocs.io/>
- [17] European Organization For Nuclear Research and OpenAIRE. 2013. Zenodo. <https://doi.org/10.25495/7GXX-RD71>
- [18] Django Software Foundation. 2023. *Object Relational Mappers*. Retrieved June 9, 2023 from <https://docs.djangoproject.com/en/4.2/topics/db/models/>
- [19] Jeremy Goecks, Anton Nekrutenko, James Taylor, and Galaxy Team team@galaxyproject.org. 2010. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology* 11 (2010), 1–13.
- [20] Clinton Gormley and Zachary Tong. 2015. *ElasticSearch: The definitive guide: a distributed real-time search and analytics engine*. O'Reilly Media, Inc.

- [21] Dick Hardt. 2012. *The OAuth 2.0 authorization framework*. Technical Report.
- [22] Katrin Heitmann, Thomas D Uram, Hal Finkel, Nicholas Frontiere, Salman Habib, Adrian Pope, Esteban Rangel, Joseph Hollowed, Danila Korytov, Patricia Larsen, et al. 2019. Hacc cosmological simulations: First data release. *The Astrophysical Journal Supplement Series* 244, 1 (2019), 17.
- [23] Faisal Khan, Suresh Narayanan, Roger Sersted, Nicholas Schwarz, and Alec Sandy. 2018. Distributed X-ray photon correlation spectroscopy data reduction using Hadoop MapReduce. *Journal of Synchrotron Radiation* 25, 4 (2018), 1135–1143.
- [24] Suresh Marru, Lahiru Gunathilake, Chathura Herath, Patanachai Tangchaisin, Marlon Pierce, Chris Mattmann, Raminder Singh, Thilina Gunarathne, Eran Chinthaka, Ross Gardler, et al. 2011. Apache airavata: a framework for distributed applications and computational workflows. In *Proceedings of the 2011 ACM workshop on Gateway computing environments*. 21–28.
- [25] Michael McLennan and Rick Kennell. 2010. HUBzero: a platform for dissemination and collaboration in computational science and engineering. *Computing in Science & Engineering* 12, 2 (2010), 48–53.
- [26] Natsuhiko Sakimura, John Bradley, Mike Jones, Breno De Medeiros, and Chuck Mortimore. 2014. Openid connect core 1.0. *The OpenID Foundation* (2014), S3.
- [27] Darren A Sherrell, Alex Lavens, Mateusz Wilamowski, Youngchang Kim, Ryan Chard, Krzysztof Lazarski, Gerold Rosenbaum, Rafael Vescovi, Jessica L Johnson, Chase Akins, et al. 2022. Fixed-target serial crystallography at the Structural Biology Center. *Journal of Synchrotron Radiation* 29, 5 (2022).
- [28] Tyler J Skluzacek, Ryan Wong, Zhuozhao Li, Ryan Chard, Kyle Chard, and Ian Foster. 2021. A serverless framework for distributed bulk metadata extraction. In *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing*. 7–18.
- [29] Joe Stubbs, Richard Cardone, Mike Packard, Anagha Jamthe, Smruti Padhy, Steve Terry, Julia Looney, Joseph Meiring, Steve Black, Maytal Dahan, et al. 2021. Tapis: an API platform for reproducible, distributed computational research. In *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 1*. Springer, 878–900.
- [30] The Globus Team. 2023. *Django Globus Portal Framework Github*. Retrieved March 2, 2023 from <https://github.com/globus/django-globus-portal-framework>
- [31] Steven Tuecke, Rachana Ananthakrishnan, Kyle Chard, Mattias Lidman, Brendan McCollam, Stephen Rosen, and Ian Foster. 2016. Globus Auth: A research identity and access management platform. In *IEEE 12th International Conference on e-Science (e-Science)*. IEEE, 203–212.
- [32] Rafael Vescovi, Ryan Chard, Nickolaus Saint, Ben Blaiszik, Jim Pruyne, Tekin Bicer, Alex Lavens, Zhengchun Liu, Michael E. Papka, Suresh Narayanan, Nicholas Schwarz, Kyle Chard, and Ian Foster. 2022. Linking Instruments and HPC: Patterns, Technologies, Experiences. Arxiv.
- [33] Siniša Veseli, Nicholas Schwarz, and Collin Schmitz. 2018. APS data management system. *Journal of Synchrotron Radiation* 25, 5 (2018), 1574–1580.