Building the Modern Research Data Portal using the Globus Platform
Platform Questions

• How do you leverage Globus services in your own applications?
• How do you extend Globus with your own services?
• How do we empower the research community to create an integrated ecosystem of services and applications?
Example: NCAR RDA

NCEP Climate Forecast System Version 2 (CFSv2) Monthly Products
ds094.2

For assistance, contact Bob Dattore (303-497-1825).

Mouse over the table headings for detailed descriptions

<table>
<thead>
<tr>
<th>Data Description</th>
<th>Data File Downloads</th>
<th>Customizable Data Requests</th>
<th>Other Access Methods</th>
<th>NCAR-Only Access</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Central File System (GLADE) Holdings</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GLADE File Listing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GLADE File Listing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GLADE File Listing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Union of Available Products</th>
<th>Web Server Holdings</th>
<th>Request Globus Invitation</th>
<th>Get a Subset</th>
<th>TDS Access</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Diurnal monthly means</td>
<td>Web File Listing</td>
<td>Request Globus Invitation</td>
<td>Get a Subset</td>
<td>TDS Access</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regular monthly means</td>
<td>Web File Listing</td>
<td>Request Globus Invitation</td>
<td>Get a Subset</td>
<td>TDS Access</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selected Parameter/Level Time Series</td>
<td>Web File Listing</td>
<td>Request Globus Invitation</td>
<td>Get a Subset</td>
<td>TDS Access</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Demo

Sample

Research Data Portal
Prototypical research data portal
Globus PaaS

- Data Publication & Discovery
- File Sharing
- File Transfer & Replication
- Auth & Groups
- Globus Toolkit
Introduction to REST APIs

- Remote operations on resources via HTTPS
  - POST ~= Create (or other operations)
  - GET ~= Read
  - PUT ~= Update
  - DELETE ~= Delete

- Globus APIs use JSON for documents and resource representations

- Resource named by URL
  - Query params allow refinement (e.g., subset of fields)

- Requests authorized via OAuth2 access token
  - Authorization: Bearer asdflkqhafdsafeawk
Globus Transfer API

- Nearly all Globus Web App functionality implemented via public Transfer API

[docs.globus.org/api/transfer](docs.globus.org/api/transfer)

- Fairly stable
- Deprecation policy
Globus Python SDK

- Python client library for the Globus Auth and Transfer REST APIs

[Link to documentation](globus.github.io/globus-sdk-python)
TransferClient class

- `globus_sdk.TransferClient` class

```python
from globus_sdk import TransferClient
tc = TransferClient()
```

- Handles connection management, security, framing, marshaling
TransferClient low-level calls

- Thin wrapper around REST API
  - post(), get(), update(), delete()

```python
get(path, params=None, headers=None, auth=None, response_class=None)
```

- path – path for the request, with or without leading slash
- params – dict to be encoded as a query string
- headers – dict of HTTP headers to add to the request
- response_class – class for response object, overrides the client’s default_response_class
- Returns: GlobusHTTPResponse object
TransferClient higher-level calls

- One method for each API resource and HTTP verb
- Largely direct mapping to REST API

```python
endpoint_search(filter_fulltext=None,
                filter_scope=None,
                num_results=25,
                **params)
```
• Jupyter (iPython) notebook demonstrating use of Python SDK

[github.com/globus/globus-jupyter-notebooks](https://github.com/globus/globus-jupyter-notebooks)

• Overview

• Open source, enjoy
Walk-through

Jupyter Notebook
Endpoint Search

• **Plain text search for endpoint**
  – Searches owner, display name, keywords, description, organization, department
  – Full word and prefix match

• **Limit search to pre-defined scopes**
  – all, my-endpoints, recently-used, in-use, shared-by-me, shared-with-me

• **Returns:** List of endpoint documents
Endpoint Management

- Get endpoint (by id)
- Update endpoint
- Create & delete (shared) endpoints
- Manage endpoint servers
Endpoint Activation

• Activating endpoint means binding a credential to an endpoint for login

• Globus Connect Server endpoint that have MyProxy or MyProxy OAuth identity provider require login via web

• Auto-activate
  – Globus Connect Personal and shared endpoints use Globus-provided credential
  – An endpoint that shares an identity provider with another activated endpoint will use credential

• Must auto-activate before any API calls to endpoints
File operations

• List directory contents (ls)
• Make directory (mkdir)
• Rename

• Note:
  – Path encoding & UTF gotchas
  – Don’t forget to auto-activate first
Task submission

• **Asynchronous operations**
  – Transfer
    o Sync level option
  – Delete

• **Get submission_id, followed by submit**
  – Once and only once submission
Task management

- Get task by id
- Get task_list
- Update task by id (label, deadline)
- Cancel task by id
- Get event list for task
- Get task pause info
Bookmarks

• Get list of bookmarks
• Create bookmark
• Get bookmark by id
• Update bookmark
• Delete bookmark by id

• Cannot perform other operations directly on bookmarks
  – Requires client-side resolution
Shared endpoint access rules (ACLs)

- Access manager role required to manage permission/ACLs

- Operations:
  - Get list of access rules
  - Get access rule by id
  - Create access rule
  - Update access rule
  - Delete access rule
Management API

- Allow endpoint administrators to monitor and manage all tasks with endpoint
  - Task API is essentially the same as for users
  - Information limited to what they could see locally

- Cancel tasks

- Pause rules
Exercise: Jupyter notebook

Install Jupyter notebook either locally or on EC2 instance

github.com/globus/globus-jupyter-notebooks.git

Modify Jupyter notebook to:

1. Find the endpoint id for XSEDE Comet
2. Set all the metadata fields on your shared endpoint
3. Set permissions to allow your neighbor to access your shared endpoint
4. Transfer all files *.txt from the tourexercise directory on the Globus Vault endpoint to any other endpoint.
Maximizing the value of the Science DMZ
Prototypical research data portal

Identity Provider

Globus Web Helper Pages

Globus Auth

Globus Transfer

Globus Cloud

Portal Web Server (Client)

REST

Other Services

Other Endpoints

User’s Endpoint (optional)

Portal Endpoint

Science DMZ

Firewall

Desktop

Browser

Applications

Login

HTTPS

GridFTP
Prototypical research data portal

- Identity Provider
- Globus Web Helper Pages
- Globus Auth
- Portal Web Server (Client)
- Portal Endpoint
- Globus Transfer
- Other Services
- Other Endpoints
- Science DMZ
- FireWall
- Desktop
- Gateway
- Portal Web Server (Client)
- Globus Auth
- Browser
- Applications
- User’s Endpoint (optional)

HTTPS

REST

GridFTP
Globus PaaS

- Data Publication & Discovery
- File Sharing
- File Transfer & Replication
- Auth & Groups
- Globus Toolkit
Challenge

• **How to provide:**
  – Login to apps
    o Web, mobile, desktop, command line
  – Protect all REST API communications
    o App → Globus service
    o App → non-Globus service
    o Service → service

• **While:**
  – Not introducing even more identities
  – Providing least privileges security model
  – Being agnostic to programming language and framework
  – Being web friendly
  – Making it easy for users and developers
Globus Auth

• Foundational identity and access management (IAM) platform service
• Simplify creation and integration of advanced apps and services
• Brokers authentication and authorization interactions between:
  – end-users
  – identity providers: InCommon, XSEDE, Google, portals
  – services: resource servers with REST APIs
  – apps: web, mobile, desktop, command line clients
  – services acting as clients to other services
Globus Auth

- Identity and access management PaaS

[docs.globus.org/api/auth](https://docs.globus.org/api/auth)

- Specification
- Developer Guide
- API Reference
Based on widely used web standards

- OAuth 2.0 Authorization Framework
  - aka OAuth2

- OpenID Connect Core 1.0
  - aka OIDC

- Use various OAuth2 and OIDC libraries
  - Google OAuth Client Libraries (Java, Python, etc.), Apache mod_auth_openidc, etc.
  - Globus Python SDK
Scopes

• APIs that client is requesting access to

• Scope syntax:
  – OpenID Connect: openid, email, profile
  – urn:globus:auth:scope:<service-name>:<scope-name>

• If client requests multiple scopes
  – Token response has tokens for first scope
  – other_tokens field in response has list of token responses for other scopes
  – Client must use correct token with each request
Consent

- Resource owner authorization that a client can request access to a service scope on the resource owner's behalf within a limited scope
  - If service has dependent scopes, they are part of the consent

- User can rescind a consent at any time
  - Invalidates all access, dependent, and refresh tokens originating from the client
Globus account

• A Globus account is a set of identities
  – A primary identity
    o Identity can be primary of only one account
  – One or more linked identities
    o Identity can (currently) be linked to only one account

• Account does not have own identifier
  – An account is uniquely identified using its primary identity
Identity id vs. username

• **Identity id:**
  – Guaranteed unique among all Globus Auth identities, and will never be reused
  – UUID
  – Always use this to refer to an identity

• **Identity username:**
  – Unique at any point in time
    o May change, may be re-used
  – Case-insensitive user@domain
  – Can map to/from id, for user experience

• **Auth API allows mapping back and forth**
Use case: Log in with Globus

• Similar to: “Log in with Google” “Log in with Facebook”
• Using existing identities
• Providing access to community services
Demo

Jetstream App use of Globus Auth
Sample Research Data Portal

- UChicago Identity Provider
- Globus Web Helper Pages
- Globus Auth
- Globus Transfer
- Demo Portal (Client)
- Firewall
- HTTPS
- REST
- Graph Services
- GridFTP
- UChicago Midway Endpoints
- Science DMZ
- My Laptop
- Desktop
- Applications
- Login
- Browser
Use case: Portal calling services on user’s behalf

• **Examples:**
  – Portal starting transfer for user

• **Authorization Code Grant**
  – With service scopes
  – Can also request OIDC scopes

• **Confidential client**

• **Globus SDK:**
  – To get tokens: ConfidentialAppAuthClient
  – To use tokens: AccessTokenAuthorizer
Authorization Code Grant

1. Access portal

2. Redirects user

3. User authenticates and consents

4. Authorization token

5. Authenticate using client id and secret, send authorization code

6. Access tokens

7. Authenticate with access tokens to invoke transfer service as user
App registration

- Client_id and client_secret for service
- App display name
- Declare required scopes
  - Need long-term, offline refresh tokens?
  - May require authorization from scope admin
- OAuth2 redirect URIs
- Links for terms of service & privacy policy
- Effective identity policy (optional)

developers.globus.org
Sample Research Data Portal

Demo: Install and Register

Code walk through
Prototypical research data portal

- Science DMZ
- Firewall
- Desktop
- Globus Cloud
- Globus
  - Web Helper Pages
  - Auth
  - Transfer
- Other Services
- Other Endpoints

- HTTPS
- REST
- GridFTP

- Identity Provider
- Portal Web Server (Client)
- Portal Endpoint
- Other Endpoints
- User’s Endpoint (optional)
- Applications
- Browser
- Firewall
- Login

- Globus
- Portal Web
- Helper Pages
- Auth
- Transfer
- Other Services
- Other Endpoints
- User’s Endpoint (optional)
- Applications
- Browser
- Firewall
- Login
Use case: Native apps

• **Examples**
  – Command line, desktop apps
  – Mobile apps
  – Jupyter notebooks
  – Any client that cannot keep a secret (downloaded)

• **Native app is registered with Globus Auth**
  – Not a confidential client

• **Native App Grant is used**
  – Variation on the Authorization Code Grant

• **Globus SDK:**
  – To get tokens: NativeAppAuthClient
  – To use tokens: AccessTokenAuthorizer
Native App grant

1. Run application
2. URL to authenticate
3. Authenticate and consent
4. Auth code
5. Register auth code
6. Exchange code
7. Access tokens
8. Authenticate with access tokens to invoke transfer service as user

Browser

Native App (Client)

Globus Auth (Authorization Server)

Globus Transfer (Resource Server)
Use case: Apps that need access tokens for long time

• **Examples:**
  – Portal checks for transfer status when user is not logged in
  – Run command line app from script

• **App requests refresh tokens**

• **Globus SDK:**
  – To get token: ConfidentialAppClient or NativeAppClient
  – To use tokens: RefreshTokenAuthorizer
Refresh tokens

• For “offline services”
  – E.g., Globus transfer service working on your behalf even when you are offline

• Refresh tokens issued to a particular client for use with a particular scope

• Client uses refresh token to get access token
  – Confidential client: client_id and client_secret required
  – Native app: client_secret not required

• Refresh token good for 6 months after last use

• Consent rescindment revokes resource token
1. Run application

2. URL to authenticate

5. Register auth code

3. Authenticate and consent

4. Auth code

6. Exchange code, request refresh tokens

7. Access tokens and refresh tokens

8. Store refresh tokens

9. Exchange refresh token for new access tokens

10. Access tokens

11. Authenticate with access tokens to invoke transfer service as user
Demo: Native App/Refresh Tokens

https://github.com/globus/native-app-examples

• README for install instructions
• ./example_copy_paste.py
  – Copy paste code to the app
• ./example_local_server.py
  – Local server to get the code
• ./example_copy_paste_refresh_token.py
  – Stores refresh token locally, uses it to get new access tokens
Use case: App invoking services as itself

- **Examples**
  - Sample portal invoking graph service and accessing endpoints as itself
  - Robots, agents, services

- **Every app is/has an identity in Globus Auth**
  (<client_id>@clients.auth.globus.org)

- **App registers with Globus to get client id/secret**
  - Native app cannot do this (no client_secret)

- **Client Credential Grant is used**

- **Can use the client_id just like any other identity_id**
  - Sharing access manager role, permissions, group membership, etc.

- **Globus SDK:**
  - To get tokens: ConfidentialAppAuthClient
  - To use tokens: AccessTokenAuthorizer
1. Authenticate with portal client id and secret

2. Access Tokens

3. Authenticate as portal with access tokens to invoke service
• User logging into portal results in portal having user’s identity and access token
  – Used to make requests on the user’s behalf

• Portal may also need its own identity
  – Access and refresh tokens for this identity
  – Used to make requests on its own behalf, e.g. set an ACL on a shared endpoint
Exercise: Using Client credential grant

- Start with native app examples
- Register a new app to get client id and secret
- **Globus SDK:**
  - ConfidentialClientApp
  - AccessTokenAuthorizer
- **Using the Globus webapp:**
  - Create a shared endpoint
  - Set Access Manager role for the new client id
- **List files on the shared endpoint as the client identity**
- **Change permissions on the shared endpoint as the client identity**
- **Hint:** Look at Jupyter notebook for SDK calls for the transfer operations
Automating Common Tasks with Globus
1. Repeated replication

- Recurring transfers with sync option
- Using Globus CLI or SDK
- Meant to be run via cron or other task manager
- Native app grant

Copy /ingest Daily @ 3:30am
2. Data distribution using sharing

- Uses Auth and Transfer API via SDK
- Native app grant
- Client credential grant
  - portal or service
  - Permission for the client id
3. Monitor and clean up

- Poll model to get status
- Delete files

1. Check if successful transfer
2. Delete data from staging area
Prototypical research data portal

- Identity Provider
- Globus Web Helper Pages
- Globus Auth
- Portal Web Server (Client)
- Globus Transfer
- Other Services
- Other Endpoints

- HTTPS
- REST
- GridFTP
HTTPS to Endpoints

• Each endpoint HTTPS server is a Globus Auth service (resource server)

• Web page can link to file on server
  – Browser GET will cause HTTPS server to authorize request via Globus Auth (note SSO)

• Portal (client) can request scope for endpoint resource server
  – Use access token in requests
Globus Helper Pages

- Globus provided web pages designed for use by your web apps
  - Browse Endpoint
  - Activate Endpoint
  - Select Group
  - Manage Identities
  - Manage Consents
  - Logout

[docs.globus.org/api/helper-pages]
Client Logout

- **Call token revocation on access tokens**
  - [https://auth.globus.org/v2/oauth2/token/revoke](https://auth.globus.org/v2/oauth2/token/revoke)
  - Doc: [docs.globus.org/api/auth/reference](https://docs.globus.org/api/auth/reference)
  - Note: Does not revoke dependent tokens

- **Delete access tokens**

- **Redirect to logout helper page**
  - [https://auth.globus.org/v2/web/logout](https://auth.globus.org/v2/web/logout)
  - Doc: [docs.globus.org/api/helper-pages](https://docs.globus.org/api/helper-pages)
Prototypical research data portal

- Identity Provider
- Globus Web Helper Pages
- Globus Auth
- Portal Web Server (Client)
- GridFTP
- REST
- Portal Endpoint
- Other Endpoints

HTTPS

Desktop
- Browser
- Applications
- User’s Endpoint (optional)

Firewall

Globus Cloud
- Globus Transfer
- Other Services

Science DMZ

Other Services
Why create your own services?

• Front-end / back-end within your portal
  – Remote backend for portal
  – Backend for pure Javascript browser apps

• Extend your app/portal with a public REST API, so that other developers can integrate with and extend it
Why Globus Auth for your service?

• Outsource all identity management and authentication
  – Federated identity with InCommon, Google, etc.

• Outsource your REST API security
  – Consent, token issuance, validation, revocation
  – You provide service-specific authorization

• Apps use your service like all others
  – It's standard OAuth2 and OIDC

• Your service can seamlessly leverage other services

• Other services can leverage your service

• Implement your service using any language and framework

Add your service to the science cyberinfrastructure platform
1. Login and consent for portal and use of graph & transfer service.

2. Client credential grant to get access tokens

3. Authenticate with access tokens to invoke graph service: HTTPS with access token as header

4. Authenticate with graph service client id and secret to introspect token

5. Return validity, client, scope, effective identity, identity set (for the portal)

6. Verifies token, authorization checks

7. Graph service response
Summary of how resource works

• Registration of resource servers
  – Scopes

• Dependent services

• Validation
Additional Features for Service Developers
Service registration

- Client_id and client_secret for service
- Service display name
- Validated DNS name for service
- One or more scopes
- Authorize clients to use each scope
  - All clients (public API), or specific clients
- Declare dependent scopes
  - Need long-term, offline refresh tokens?
  - May require authorization from scope admin
- Links for terms of service & privacy policy
- Effective identity policy (optional)
- Email: support@globus.org
Effective identity

• App or service can choose to operate only with identities from a particular identity provider
  – Globus Auth login will require an identity from that provider to be linked to user’s account
  – OIDC id_token uses this “effective identity”

• If app or service does not set an effective identity policy, then the primary identity of the account is used as the effective identity for that app
Branding

• Can skin Globus Auth pages

Header

Text

Default IdP
Token caching

• **Service should cache tokens and related information**
  – Improves performance of service
  – Reduces load on Globus Auth

• **Access token -> introspect response**
  – Cache timeout: 1-30 seconds recommended
  – To improve performance and load related to bursty use of REST API
  – Validity: Timeout duration determines responsiveness to token revocation and rescinding consent
  – client, scope, effective_identity: these will never change for an access token

• **Refresh tokens**
  – For however long they are needed for specific operations.
Join the Globus developer community

• Join developer-discuss@globus.org mailing lists: globus.org/mailing-lists
• Python SDK is open source
  – github.com/globus/globus-sdk-python
  – Submit issues, pull requests
  – Discussions on developer-discuss@globus.org
• All tutorial materials are open source on github
• Documentation: docs.globus.org