Building Research Data Management Solutions with the Globus Platform

Vas Vasiliadis  
vas@uchicago.edu

Rachana Ananthakrishnan  
rachana@globus.org

Stanford University – February 9, 2018
Useful developer links
> github.com/globus
> docs.globus.org
Topics

- Globus CLI and Platform Overview
- Globus Transfer and Globus Auth APIs
- Automating common data management tasks
- Integrating Globus into Science Gateways and Portals
- Open Discussion
How can I integrate Globus into my research workflows?
Use(r)-appropriate interfaces

Use:
- appropriate interfaces

Examples:
- GET /endpoint/go%23ep1
- PUT /endpoint/vas#my_endpt

Response:
- 200 OK
- X-Transfer-API-Version: 0.10
- Content-Type: application/json

Interfaces:
- Web App
- CLI
- Rest API
Globus Command Line Interface

Open source, uses Python SDK

docs.globus.org/cli

github.com/globus/globus-cli
Demonstration
Globus CLI
Install the Globus Command Line Interface (CLI)
(or access via user “globus” on EC2 instance)

docs.globus.org/cli/installation
Exercise: Transfer via the CLI

- Join the Tutorial Users group
- Use the CLI to copy files in “/sometext/” on endpoint “Stanford Workshop” to “Globus Tutorial Endpoint 1”
- Later we will look at a more robust example: github.com/globus/automation-examples/blob/master/cli-sync.sh
Solution: Transfer via the CLI

• Get the source and destination endpoint IDs

$ globus endpoint search "Stanford Workshop"
$ globus endpoint search "Globus Tutorial Endpoint 1"

• Submit the transfer request

$ globus transfer --recursive 7026c6d4-0c84-11e8-a763-0a448319c2f8:/sometext/ ddb59aef-6d04-11e5-ba46-22000b92c6ec:/~/
How do I go beyond simple scripts?
Globus serves as...

A platform for building science gateways, portals and other web applications in support of research and education
Globus Platform-as-a-Service

Integrate file transfer and sharing capabilities into scientific web apps, portals, gateways, etc.

Use existing institutional ID systems in external web applications.
Globus Platform Transfer API
Globus Transfer API

• Globus Web App consumes public Transfer API
• Globus APIs use JSON for documents and resource representations
• Resource named by URL (standard REST approach)
  – Query params allow refinement (e.g., subset of fields)
• Requests authorized via OAuth2 access token
  – Authorization: Bearer asdflkqhafsdafewk

[Link to docs: docs.globus.org/api/transfer]
Globus Python SDK

- Python client library for the Globus Auth and Transfer REST APIs
- `globus_sdk.TransferClient` class handles connection management, security, framing, marshaling

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

[globus.github.io/globus-sdk-python](https://globus.github.io/globus-sdk-python)
TransferClient low-level calls

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

get(path, params=None, headers=None, auth=None, response_class=None)
 o path – path for the request, with or without leading slash
 o params – dict to be encoded as a query string
 o headers – dict of HTTP headers to add to the request
 o response_class – class response object, overrides the client’s default_response_class
 o Returns: GlobusHTTPResponse object
Walkthrough Jupyter Notebook

github.com/globus/globus-jupyter-notebooks

(install locally or run on EC2 instance)
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: Transfer API - Data distribution

Modify Jupyter notebook to…

– Find the endpoint ID for the “Stanford Workshop” endpoint
– Bonus points: how many endpoints are associated with “Stanford“?
– Make a directory for your files: /<your_globus_id>
– Transfer some files to your directory
  o Get the ID of the “ESnet Read-Only Test DTN at Sunnyvale“ endpoint
  o Transfer the “/data1/5GB-in-small-files“ directory to your directory
– For overachievers: check if files were successfully transfered and delete them
Solution: Transfer API - Data distribution

• Find ID for “Stanford Workshop” endpoint [29]*
  
r = tc.get("/endpoint_search",
             params=dict(filter_fulltext="Stanford Workshop", limit=1))

• Make a directory for your files: /<your_globus_id> [36]
  
  endpoint_id = stanford_workshop_ep_id
  endpoint_path = "/vas"

• Transfer ESnet files to your directory [45]
  
  source_endpoint_id = "db57ddde-6d04-11e5-ba46-22000b92c6ec"
  source_path = "/data1/5GB-in-small-files/"
  dest_endpoint_id = stanford_workshop_ep_id
  dest_path = "/vasv/5GB-in-small-files/"

* The number in brackets refers to the Jupyter notebook code block to be modified/extended
Solution: Transfer API - Data distribution

- Check if successfully transferred and delete files [50]*

  ```python
  If status == "SUCCEEDED":
      source_endpoint_id = "db57ddde-6d04-11e5-ba46-22000b92c6ec"
      endpoint_id = stanford_wrokshop_ep_id
      path = "/vasv/5GB-in-small-files/"
      r = globus_sdk.DeleteData(tc, endpoint_id, recursive=True)
      ddata.add_item(path)
      tc.endpoint_autoactivate(endpoint_id)
      submit_result = tc.submit_delete(ddata)
      print("Task ID:", submit_result["task_id"])
  ```

* The number in brackets refers to the Jupyter notebook code block to be modified/extended
How can I do this in my [science gateway, data portal, web app, ...]?
Prototypical research data portal
Demonstration
Modern Research
Data Portal
Maximizing the value of the Science DMZ
Can we “disassemble” this design and reassemble it for improved performance?
Modern data apps leverage the Science DMZ

fasterdata.es.net/
Data Distribution: ARM Climate Research Facility
Analysis Workflow Integration: Wellcome Sanger

Sanger Imputation Service
This is a free genotype **imputation** and **phasing** service provided by the Wellcome Trust Sanger Institute. You can upload GWAS data in VCF or 23andMe format and receive imputed and phased genomes back. Click here to learn more and follow us on Twitter.

**Before you start**
Be sure to read through the instructions. You will need to set up a free account with Globus and have Globus Connect running at your institute or on your computer to transfer files to and from the service.

**Ready to start?**
If you are ready to upload your data, please fill in the details below to register an imputation and/or phasing job. If you need more information, see the about page.

- **Full name**
- **Organisation**
- **Email address**

**What is this?**
Globus user identity

**News**
- @sangerimpute

- **11/05/2016**
  - Thanks to EAGLE, we can now return phased data. The HRC panel has been updated to r1.1 to fix a known issue. See ChangeLog for more details.

- **15/02/2016**
  - Globus API changed, please see updated instructions.

- **17/12/2015**
  - New status page and reworked internals. See ChangeLog.

- **09/11/2015**
  - Pipeline updated to add some features requested by users. See ChangeLog.
Globus Auth: Foundational IAM service

- Enables login for diverse app ecosystem
- Simplifies creation/integration of apps, services
- Outsources mundane feature development
- Brokers authentication and authorization interactions
- Protects REST API communications
- No new identity required
- Employs least privileges security model
- Programming language and framework agnostic
Globus Auth

docs.globus.org/api/auth

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

- OAuth 2.0 Authorization Framework (a.k.a. OAuth2)
- OpenID Connect Core 1.0 (a.k.a. OIDC)
- Access via OAuth2 and OIDC libraries of your choice
  - Google OAuth Client Libraries (Java, Python, etc.), Apache mod_auth_openidc, etc.
  - Globus Python SDK

[docs.globus.org/api/auth]
Fundamental Concepts

• **Scopes**: APIs that client is requesting access to
  – Scope syntax: OpenID Connect: openid, email, profile
  – urn:globus:auth:scope:<service-name>:<scope-name>

• **Consents**: authorization client to access a service, within limited scope, on the resource owner's behalf
Globus account

• **Globus Account = Primary identity + Linked Identities**
  – An identity can be primary on only one account
  – Identities can 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***
  - Unique among all Globus Auth identities; will never be reused
  - UUID
  - Always use this to refer to an identity

- **Identity *username***
  - Unique at any point in time; 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**
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](developers.globus.org)
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
Use case: Native apps

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

• **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 long-lived access tokens

• **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
Automation Example: Repeated replication

- Using Globus CLI or SDK
- Meant to be run via cron or other task manager
- Native app grant

Recurring transfers with sync option

Daily @ 3:30am
Exercise: Automation using the Globus CLI

• Use the cli-sync script to sync files between the ESnet test endpoint and your personal endpoint
  – Find the respective endpoint IDs and update the script (or parameterize it, if you feel adventurous!)
  – Decide on a source and target directory and reflect this in the script – please use one of the small(er) datasets

https://github.com/globus/automation-examples/blob/master/cli-sync.sh
Refresh tokens

- For “offline services”
  - e.g., transfer working on your behalf even when you are offline

- Refresh tokens issued to client, in 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

3. Authenticate and consent

4. Auth code

5. Register 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

Browser

Native App (Client)

Globus Auth (Authorization Server)

Globus Transfer (Resource Server)
Demo: Native App/Refresh Tokens

github.com/globus/native-app-examples

• See README for installation
• ./example_copy_paste.py
  – Copy paste code to the app
• ./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)

• **Uses Client Credential Grant**

• **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
User identity vs. portal identity

• **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
Automation Example: Data distribution

- Uses Auth and Transfer API via SDK
- Native app grant
- Client credential grant
  - Portal or service
  - Permission for the client id
Client credential grant

1. Authenticate with portal client id and secret

2. Access Tokens

3. Authenticate as portal with access tokens to invoke service

Modern Research Data Portal

Portal (Client)

Globus Auth (Authorization Server)

Globus Transfer (Resource Server)
Exercise: Automation using the REST API

- Use the share_data Python script to share files with your neighbor
  - Modify the script to:
    - Create a shared endpoint
    - Add access permissions to the shared endpoint for your neighbor
  - Move data to the shared endpoint
  - Check that your neighbor received the sharing notification
Globus Helper Pages

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

[Link to documentation: docs.globus.org/api/helper-pages]
Globus PaaS developer resources

Python SDK

Sample Application

Jupyter Notebook

requirements

You need to be in the tutorial users group for sharing: https://www.globus.org/app/groups/50b6a29c-63ac-11e4-8062-22000ab68755

Configuration

First you will need to configure the client with an OAuth2 access token. For the purpose of this tutorial, you can use the website. Click the "Jupyter Notebook" option and copy the resulting text below, or click on "Globus CLI" and:

docs.globus.org/api
github.com/globus
Support resources

• Globus documentation: docs.globus.org
• Helpdesk and issue escalation: support@globus.org
• Customer engagement team
• Globus professional services team
  – Assist with portal/gateway/app architecture and design
  – Develop custom applications that leverage the Globus platform
  – Advise on customized deployment and integration scenarios
Join the Globus community

- Access the service: globus.org/login
- Create a personal endpoint: globus.org/app/endpoints/create-gcp
- Documentation: docs.globus.org
- Engage: globus.org/mailing-lists
- Subscribe: globus.org/subscriptions
- Need help? support@globus.org
- Follow us: @globusonline