Building the Modern Research Data Portal with Globus PaaS + Science DMZ

Vas Vasiliadis
vas@uchicago.edu

Stephen Rosen
sirosen@globus.org

Harvard University – September 13, 2017
Motivating 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
Example: ARM Climate Research Facility
Globus serves as...

A platform for building science gateways, portals and other web applications in support of research and education
Demonstration
Web App Integration
Prototypical research data portal

- **Identity Provider**
- **Globus Cloud**
- **Globus Web Helper Pages**
- **Globus Auth**
- **Globus Transfer**
- **Portal Web Server (Client)**
- **REST**
- **Other Services**
- **Other Endpoints**
- **GridFTP**
- **HTTPS**
- **Firefox**
- **Prototypical Flow**
- **Portal Web Server**
- **Portal Endpoint**
- **User’s Endpoint (optional)**
- **Globus Web Helper Pages**
- **Globus Auth**
- **Globus Transfer**
- **Portal Web Server (Client)**
- **REST**
- **Other Services**
- **Other Endpoints**
- **GridFTP**
- **HTTPS**
- **Firefox**
- **Prototypical Flow**
- **Portal Web Server**
- **Portal Endpoint**
- **User’s Endpoint (optional)**
Prototypical research data portal

- **Identity Provider**
- **Globus Web Helper Pages**
- **Globus Auth**
- **Globus Transfer**
- **Portal Web Server (Client)**
- **Portal Endpoint**
- **Globus Cloud**
- **Science DMZ**
- **Firewall**
- **Other Services**
- **Other Endpoints**
- **GridFTP**
- **REST**
- **HTTPS**

- **Browser**
- **Applications**
- **Desktop**
- **User’s Endpoint (optional)**
- **Login**
- **Portal Web**
- **Globus Auth**
- **Globus Transfer**
- **Other Services**
- **Other Endpoints**
- **GridFTP**
- **REST**
- **HTTPS**
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

- Stable API with defined deprecation policy
Globus Python SDK

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

[globus.github.io/globus-sdk-python](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
def 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: GlobusHTTResponse 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)
```
Python SDK Jupyter notebook

- 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
Walkthrough
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**
Exercises: Jupyter notebook

• Install the Jupyter notebook (locally or on EC2)
  
github.com/globus/globus-jupyter-notebooks.git

• Modify the Jupyter notebook to:
  – Find the endpoint id for XSEDE Comet
  – Set some metadata fields on your shared endpoint
  – Transfer all .txt files from the GlobusWorld Tour endpoint to any other endpoint
Next-Generation Portal Leverages Science DMZ

https://fasterdata.es.net/
Prototypical research data portal

- Identity Provider
- Globus Web Helper Pages
- Globus Auth
- Portal Web Server (Client)
- Portal Endpoint
- Science DMZ
- Firewall
- Other Endpoints
- Other Services
- GridFTP
- REST
- HTTPS
Challenge

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

• While:
  – Not introducing even more identities
  – Ensuring 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) service
• Simplify creation/integration of advanced apps & services
• Brokers authentication and authorization interactions between:
  – end-users
  – identity providers: enterprise IdP, external IdPs, e.g. Google
  – services: resource servers with REST APIs
  – apps: web, mobile, desktop, command line clients
  – services acting as clients to other services
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
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 = A set of identities**
  - A primary identity
    - Identity can be primary of only one account
  - One or more linked identities
    - 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
    - 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”
• Using existing identities
• Providing access to community services
Demonstration
Jetstream login using Globus Auth
Sample Research Data Portal

- UChicago Identity Provider
- Globus Web Helper Pages
- Globus Auth
- Globus Transfer
- Demo Portal (Client)
- Graph Services
- Portal Endpoint
- UChicago Midway Endpoints

Connections:
- HTTPS
- REST
- GridFTP

Devices:
- Browser
- Applications
- Desktop
- My Laptop
- Firewall
- Science DMZ
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
Use case: Native apps

• **Examples**
  – Command line, desktop apps
  – Mobile apps
  – Automation scripts
  – 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
Use case: Apps that need access token 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

Native App (Client)

Browser

3. Authenticate and consent

4. Auth code

Globus Auth (Authorization Server)

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

Globus Transfer (Resource Server)

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

Source: [github.com/globus/native-app-examples](https://github.com/globus/native-app-examples)
Doc examples: [globus-sdk-python.readthedocs.io](https://globus-sdk-python.readthedocs.io)
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
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

Globus Auth (Authorization Server)

Globus Transfer (Resource Server)
Automating common tasks with Globus
Data Distribution: APS - DMagic

dmagic.readthedocs.io

Courtesy of Francesco De Carlo, Argonne National Laboratory (2016)
1. Scheduled replication

- Using Globus CLI or SDK
- Designed to be run via cron or other task manager
- Native app grant
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
Demo: Automation examples

- README for install instructions
- Replication: globus_folder_sync.py, cli-sync.sh
- Data distribution: share_data.py, share-data.sh
- Monitor and clean up: cleanup_cache.py

[link to GitHub: github.com/globus/automation-examples]
Prototypical research data portal

Science DMZ

Firewall

Desktop

Globus Cloud

 HTTPS

Globus Web Helper Pages

Globus Auth

Portal Web Server (Client)

Globus Transfer

Other Services

User’s Endpoint (optional)

Portal Endpoint

Browser

Applications

Login

Globus Web Helper Pages

Globus Auth

Portal Web Server (Client)

Globus Transfer

Other Services

Other Endpoints

GridFTP

REST

Other

Endpoints

Portal Web

Helper Pages

Globus

Auth

Globus

Transfer

Science DMZ

User’s

Endpoint

(optional)

Portal Endpoint

Desktop

Login

Browser

Applications

Prototypical research data portal

HTTPS

GridFTP

REST

Other Services

Other Endpoints

Science DMZ
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
Prototypical research data portal
Globus Helper Pages

- Globus 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
  – Doc: 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
  – Doc: docs.globus.org/api/helper-pages
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**
  – Its 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.
Support resources

- Customer engagement team
- Globus documentation: [docs.globus.org](http://docs.globus.org)
- Helpdesk and issue escalation: support@globus.org
- 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