The globusworld TOUR

Building the Services Ecosystem

Steve Tuecke
tuecke@uchicago.edu

NCAR – September 5, 2018

globus

Globus Auth enables
an integrated ecosystem
of services and applications
for the research community

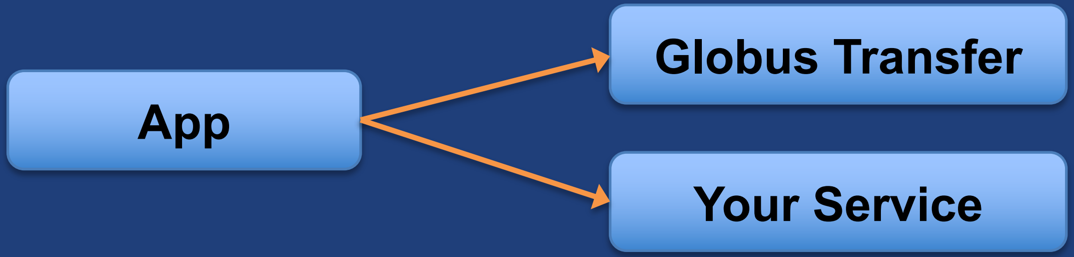
Based on widely used web standards

- **OAuth 2.0 Authorization Framework (a.k.a. OAuth2)**
- **OpenID Connect Core 1.0 (a.k.a. OIDC)**

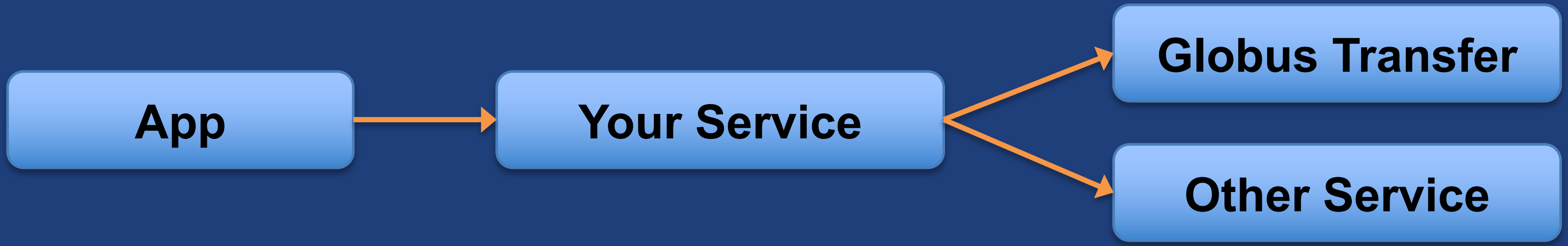
docs.globus.org/api/auth

What is a services ecosystem?

- **Build services with secure REST APIs**



- **Services can leverage other services securely**



Why create your own services?

- **Make your specialized capabilities available to your research community as a service**
- **Extend your web portal with a public REST API, so that other developers can integrate with and extend it**
- **Front-end / back-end within your portal / app**
  - Remote backend for portal
  - Backend for pure Javascript browser apps

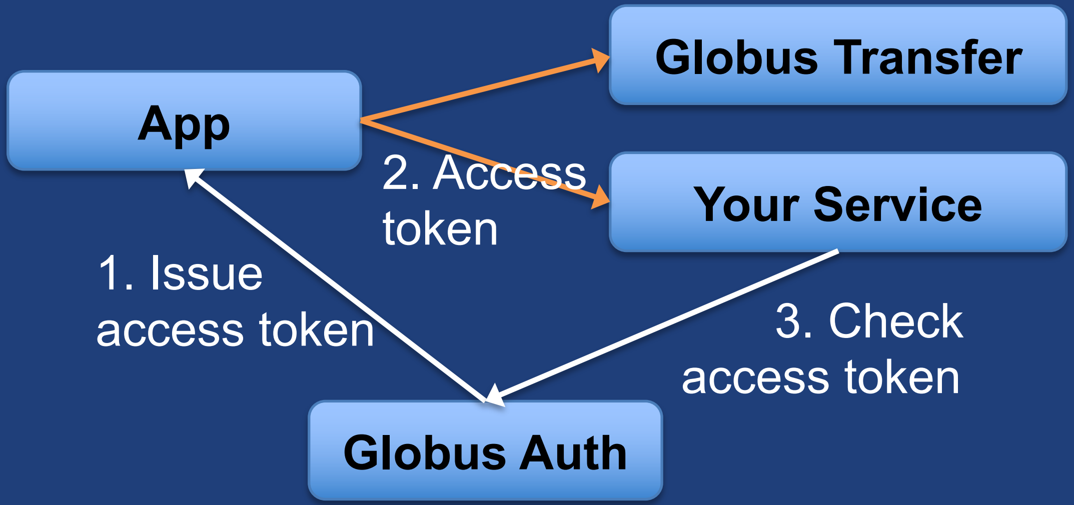
Why Globus Auth for your service?

- **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, with standard OAuth2 & 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 services ecosystem

Role of Globus Auth for services

- **Issue and check OAuth2 access tokens**



- **With a token, your service can get attributes about the user, which it can use to authorize the request**

Fundamental Concepts

- **Scopes: APIs that client is requesting access to**
  - Scope syntax: OpenID Connect: openid, email, profile
  - https://auth.globus.org/scopes/<service-name>:<scope-name>
  - A service can have multiple scopes

- **Consents: authorize client to access a service, within limited scope, on the resource owner’s (user’s) behalf**

Globus account

- **Globus Account = Primary identity + Linked Identities**
  - An identity can be primary on only one account
  - (Currently) Identities can be linked to only one account
- **Account does not have own identifier**
  - An account is uniquely identified using its primary identity
- **Effective identity = linked identity from a particular identity provider required by a client or service**

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

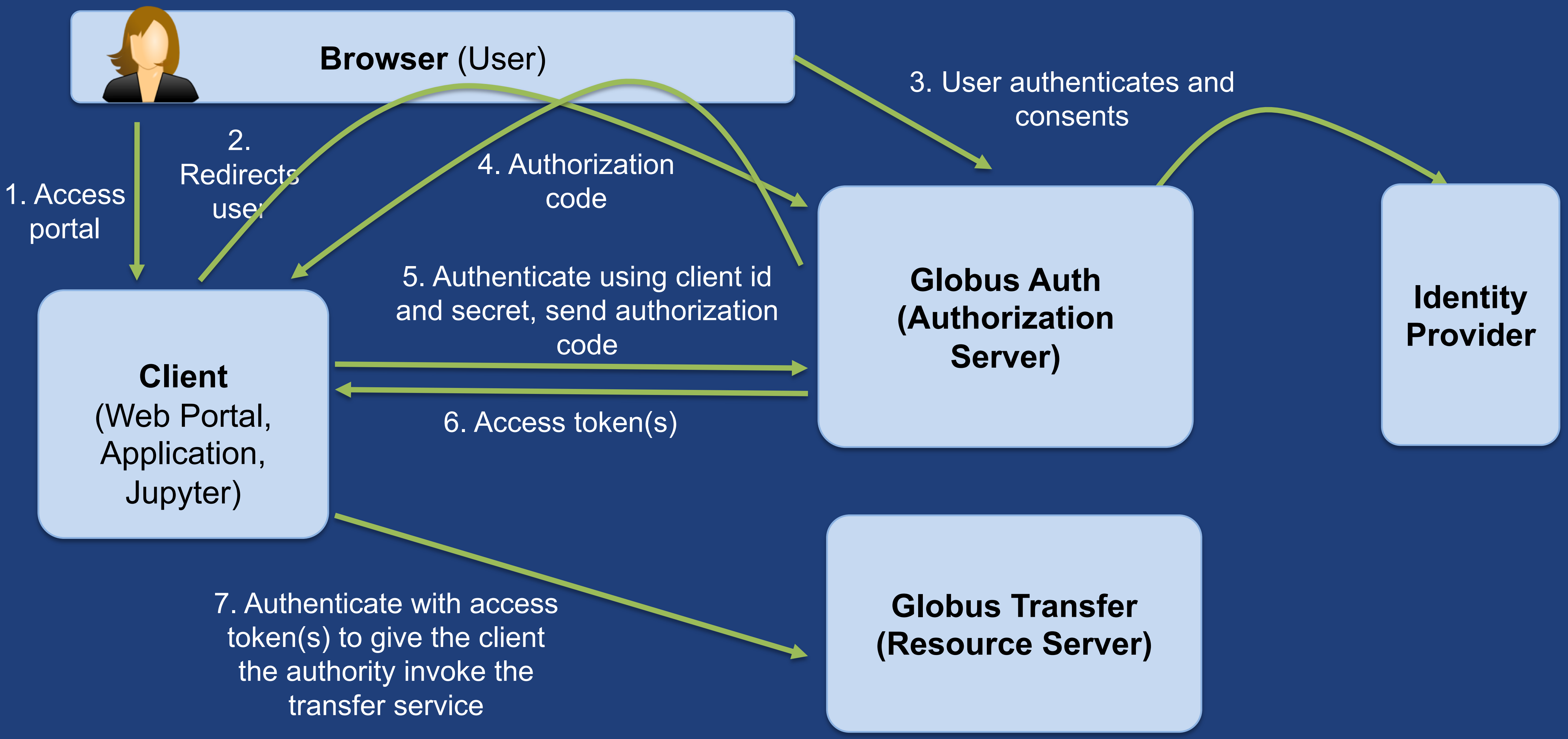
- **Admin registers an App (“confidential client”) with Globus Auth**
  - https://developers.globus.org
- **Gets client_id and client_secret for service**
- **Sets a\pp display name**
- **Declare required scopes (optional)**
  - Need long-term, offline refresh tokens?
  - May require authorization from scope admin
- **OAuth2 redirect URIs**
- **Links for terms of service & privacy policy**
- **Effective identity policy (optional)**

developers.globus.org

Native App Auto-registration (new)

- **New Native App bootstrap model**
- **Developer registers a native app template at developers.globus.org**
- **Each installed instance of the native app auto-registers itself on first use via Globus Auth Client API**
  - Registers with a native app template
  - Gets a client id and secret for itself
  - Securely stores client id and secret (e.g., into user read-only file)
- **Native app is now a “confidential client”, just like any other**

Authorization Code Grant



Browser (User)

1. Access portal

2. Redirects user

3. User authenticates and consents

4. Authorization code

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

6. Access token(s)

7. Authenticate with access token(s) to give the client the authority invoke the transfer service

Client
(Web Portal, Application, Jupyter)

Globus Auth (Authorization Server)

Identity Provider

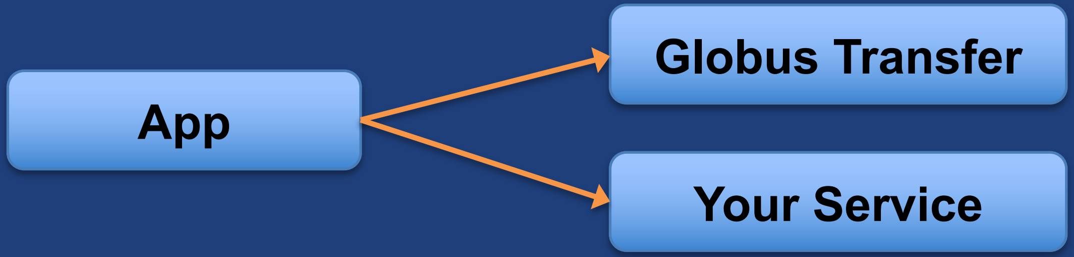
Globus Transfer (Resource Server)

OAuth2 grants for apps

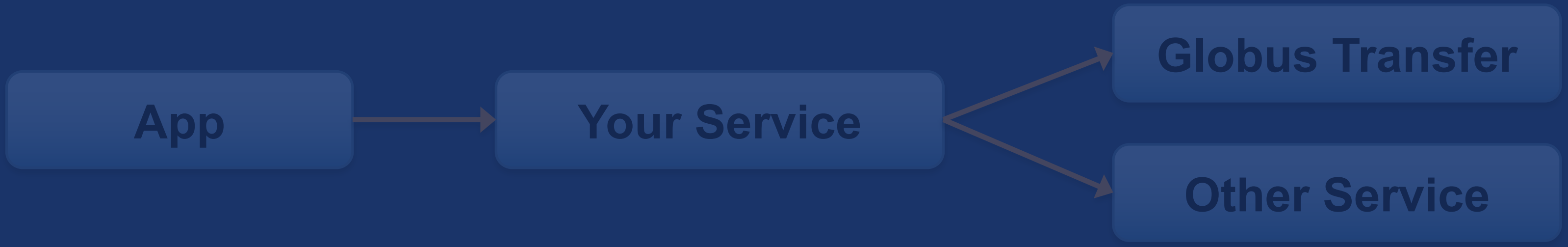
- **Authorization code grant**
  - Native app grant variant
- **Refresh token grants**
  - For apps that need long-lived, “offline” access to a service
- **Client credential grant**
  - For app invoking services as itself, instead of as the user

What is a services ecosystem?

- **Build services with secure REST APIs**



- **Services can leverage other services securely**



Service registration

- **Client_id and client_secret for service**
- **Service display name**
- **Validated DNS name for service**
- **One or more scopes**
  - Who is authorized to use each scope: all client (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)**

mailto:support@globus.org

Typical service interactions

- **Service receives HTTPS request with header**
  - Authorization: Bearer <request-access-token>
- **Introspects the request access token**
  - Auth API: POST /v2/oauth2/token/introspect
  - Authorized by client_id and client_secret
  - Returns: active, client_id, scope, sub (identity), identities_set
- **Verifies token info (e.g., active, aud, scope)**
- **Authorizes request based on token info (e.g., sub)**
- **Service processes request**
- **Responds to client HTTPS request**

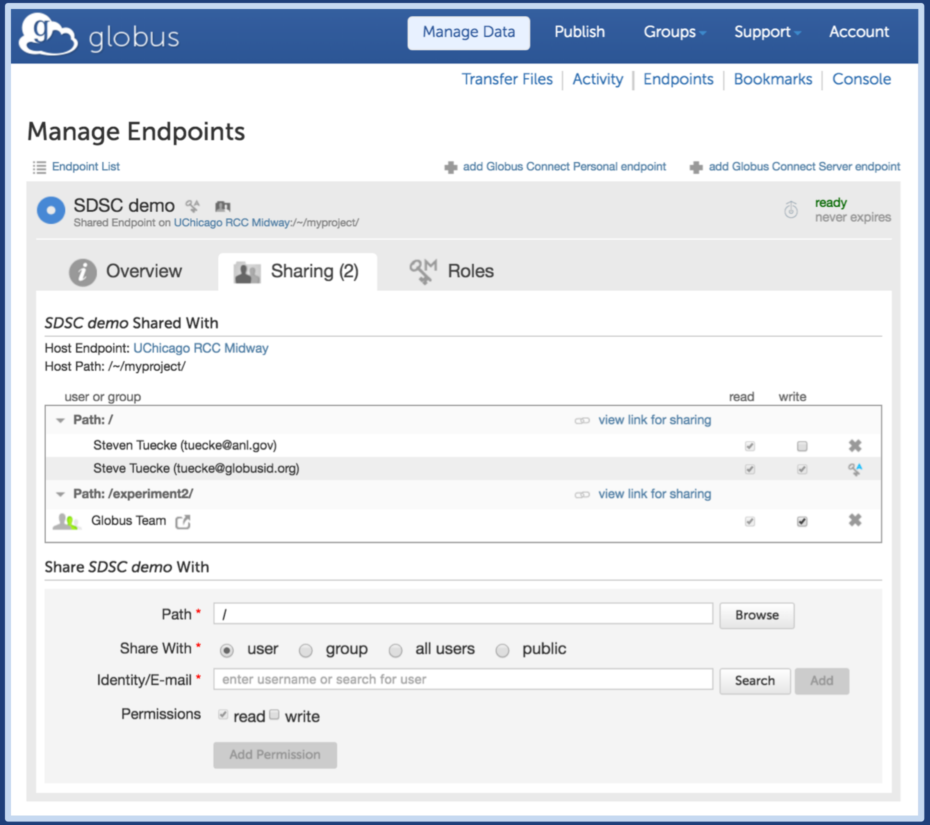
Sample Research Data Portal Service Walk-through

Authorization based on identity set

- **Use identities_set when authorizing a request based on the resource owner associated with an access token**
  - E.g., ACLs on Globus shared endpoints
- **Authorizing based on set of identities is same complexity as authorizing based on group membership set**

Groups

- **Globus group service is identity set aware**
  - “Tell me all groups for all identities of the logged in user”
- **Services can leverage this for authorization (soon)**



globus
Manage Data
Publish
Groups
Support
Account
Transfer Files | Activity | Endpoints | Bookmarks | Console
Endpoint List
add Globus Connect Personal endpoint
add Globus Connect Server endpoint
Overview
Sharing (2)
Roles

Manage Endpoints

SDSC demo
Shared Endpoint on UChicago RCC Midway:/~/myproject/

ready
never expires

SDSC demo Shared With

Host Endpoint: UChicago RCC Midway
Host Path: /~/myproject/

user or group		read	write	
Path: /	view link for sharing			
Steven Tuecke (tuecke@anl.gov)		☑	□	✖
Steve Tuecke (tuecke@globusid.org)		☑	☑	
Path: /experiment2/	view link for sharing			
Globus Team		☑	☑	✖

Share *SDSC demo* With

Path * / Browse

Share With * ● user ○ group ○ all users ○ public

Identity/E-mail * enter username or search for user Search Add

Permissions ☑ read □ write

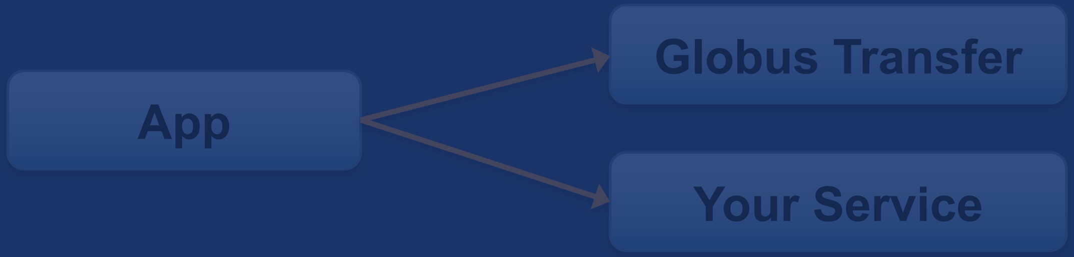
Add Permission

Django REST Framework

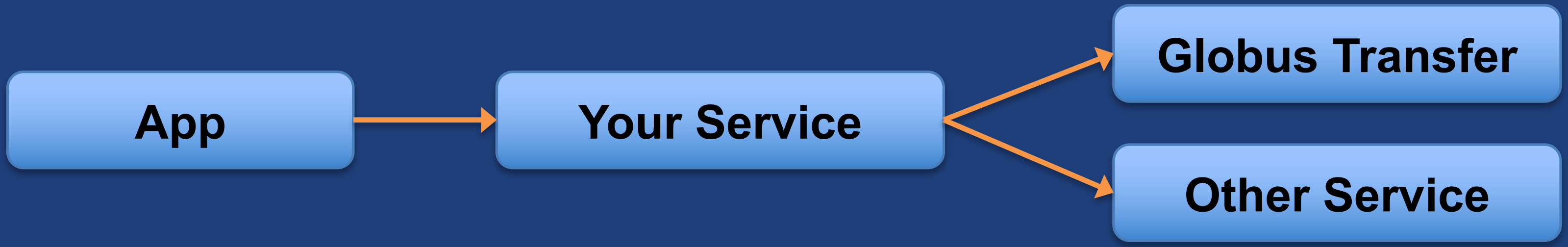
- Use Globus Auth to protect REST APIs in the Django REST Framework
- Simple enhancements to do token introspection, validation, and map to Django account
- Contact support@globus.org if you are interested

What is a services ecosystem?

- Build services with secure REST APIs



- Services can leverage other services securely



Examples uses of dependent services

- **NCAR RDA REST API**
  - – App → RDA → Globus Transfer
- **Globus Transfer**
  - – Globus Connect Server v5 expects Globus Auth access tokens
  - – App → Transfer → Collections

Dependent tokens

- **Your service can act as client to other services (scopes)**
  - Globus Transfer, Search, Identities, Auth
  - Other community services
- **Entire service call tree consented by user and service owners**
  - Rescinding consent revokes all dependent tokens
- **Dependent tokens are restricted to a particular client, calling a particular scope, on behalf of a particular resource owner (e.g., user)**
  - Restricted delegation!

Typical service interactions

- **Service receives HTTPS request with header**
- **Introspects the request access token**
- **Verifies token info (e.g., active, aud, scope)**
- **Authorizes request based on token info (e.g., sub)**
- **If service needs to act as client to other services:**
  - Calls Globus Auth Dependent Token Grant
    - Returns a token for each dependent service
  - Uses correct dependent token for downstream REST call
- **Service processes request, including calls to other services**
- **Responds to client HTTPS request**

Sessions

- **Higher authentication assurance**
  - For services with PHI, PII, sensitive but unclassified data
- **Introspection response includes session attributes**
  - Which identities have authenticated in this “session”
  - When each authenticated last
  - Whether multi-factor authentication was used
- **Return authentication assurance error if not sufficient**

Refresh tokens

- **For “offline service”: Service working on your behalf even when you are offline**
  - Example use: Globus Transfer service, for async transfers
- **Refresh tokens issued to a particular client for use with a particular scope**
- **Client uses refresh token to get access token**
  - Client_id and client_secret required
- **Refresh token good for 6 months after last use**
- **Consent rescindment revokes resource token**

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
- Access token -> dependent access tokens
  - Cache timeout: lifetime of access token
  - To avoid costly dependent token re-issuance
  - Rescinding consent will invalidate everything
- Refresh tokens
  - For however long they are needed for specific operations.
  - Keep distinct refresh tokens for each access token.

Coming soon to Auth

- **Incremental auth**
  - Add consents and tokens for new services dynamically
- **Optional scopes**
  - Allow user to optionally deny access to a scope, but allow the client to continue functioning with reduced capability
- **SSH with Globus Auth**

Summary

- Globus Auth makes it easy to:
  - add OAuth2 support to secure your service’s REST API
  - create services to leverage other services

Globus enables an integrated ecosystem of services and applications for the research community

Support resources

- **Globus documentation: docs.globus.org**
- **Community email list: developer-discuss@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