

Tutorial: Building the Services Ecosystem

GlobusWorld 2018

Steve Tuecke

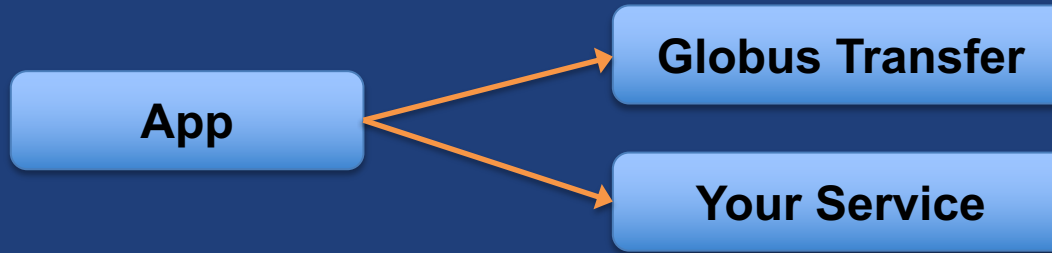
tuecke@globus.org



globus

What is a services ecosystem?

- Anybody can 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?

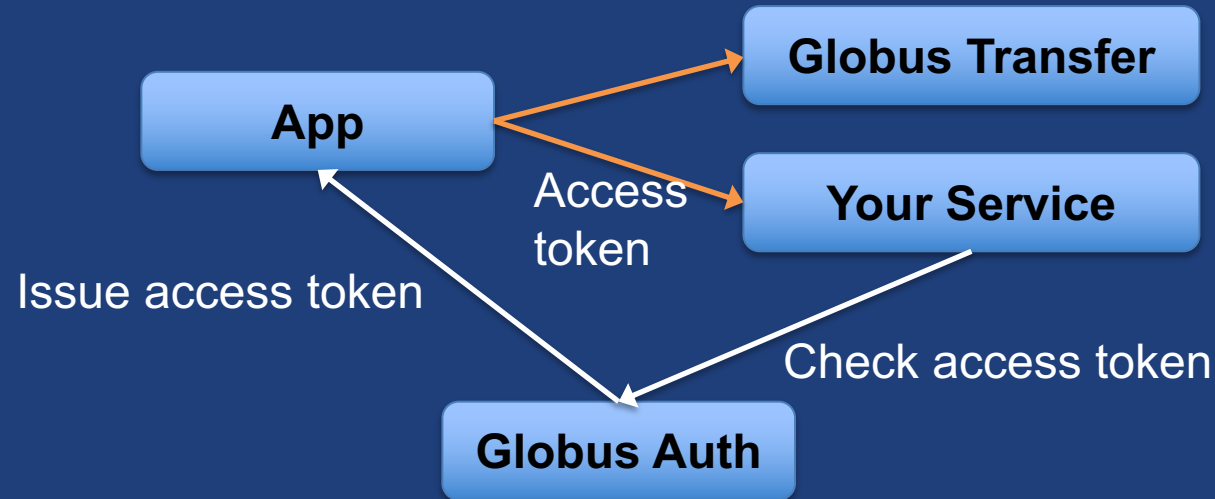
- **Outsource all identity management and authentication**
 - Federated identity with InCommon, Google, etc.
- **Outsource your REST API security**
 - Consent, token issuance, validation, revocation
 - You provide service-specific authorization
- **Apps use your service like all others, with standard 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

 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

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**

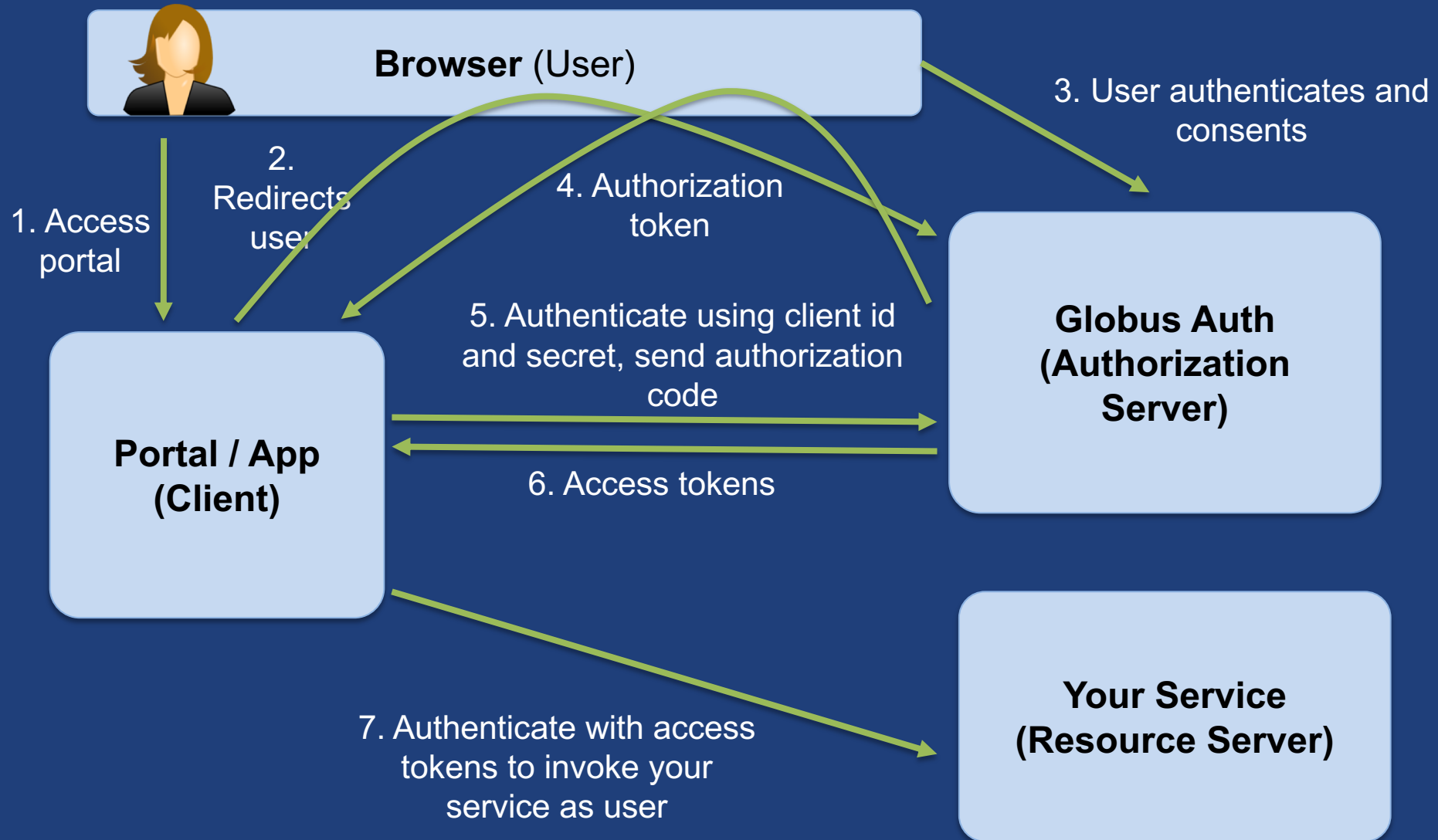
App registration

- **Client_id and client_secret for service**
- **App 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



Authorization Code Grant

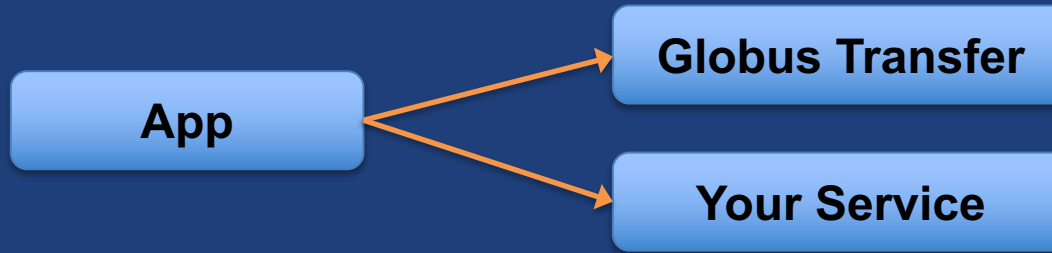


OAuth2 grants for apps

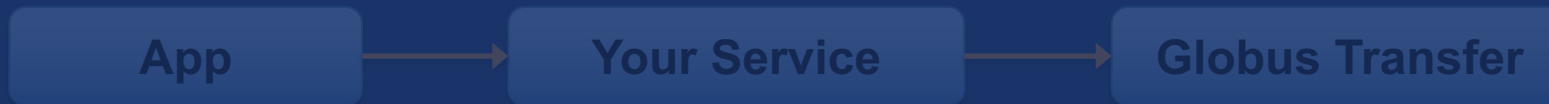
- **Authorization code grant**
 - Native app grant variant
- **Refresh token grants**
 - For apps that need long-lived, “offline” access to a service
- **Client credential grant**
 - For app invoking services as itself, instead of as the user

What is a services ecosystem?

- Anybody can 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

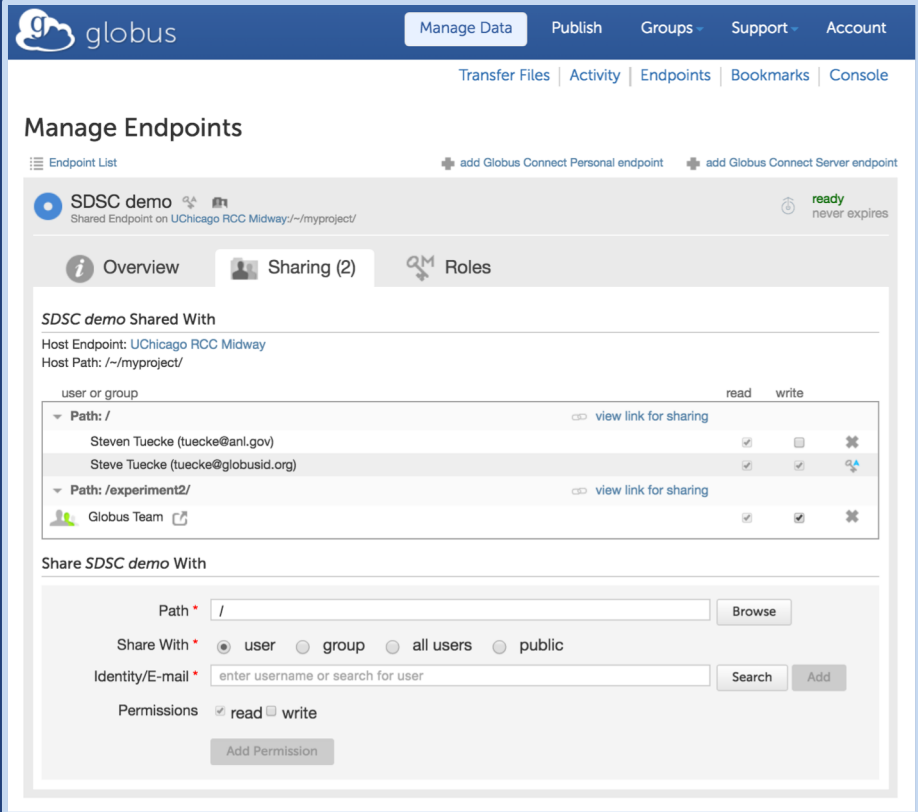


Authorization based on identity set

- **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**



The screenshot shows the 'Manage Endpoints' page in the Globus interface. The endpoint is 'SDSC demo', a shared endpoint on UChicago RCC Midway. The 'Sharing (2)' tab is active, showing a table of shared users and groups. Below the table, there is a 'Share SDSC demo With' section with input fields for path, share with (user, group, all users, public), identity/e-mail, and permissions (read, write).

user or group	read	write
Path: /		
Steven Tuecke (tuecke@anl.gov)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Steve Tuecke (tuecke@globusid.org)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Path: /experiment2/		
Globus Team	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

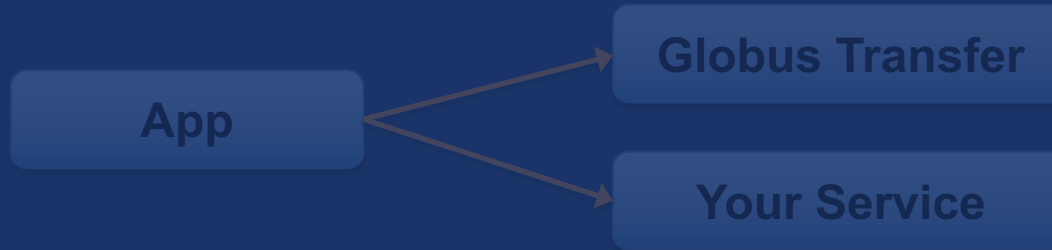
Django REST Framework

- **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?

- Anybody can build services with secure REST APIs



- **Services can leverage other services securely**





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



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

- **Higher authentication assurance**
 - For services with PHI, PII, sensitive but unclassified data
- **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



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