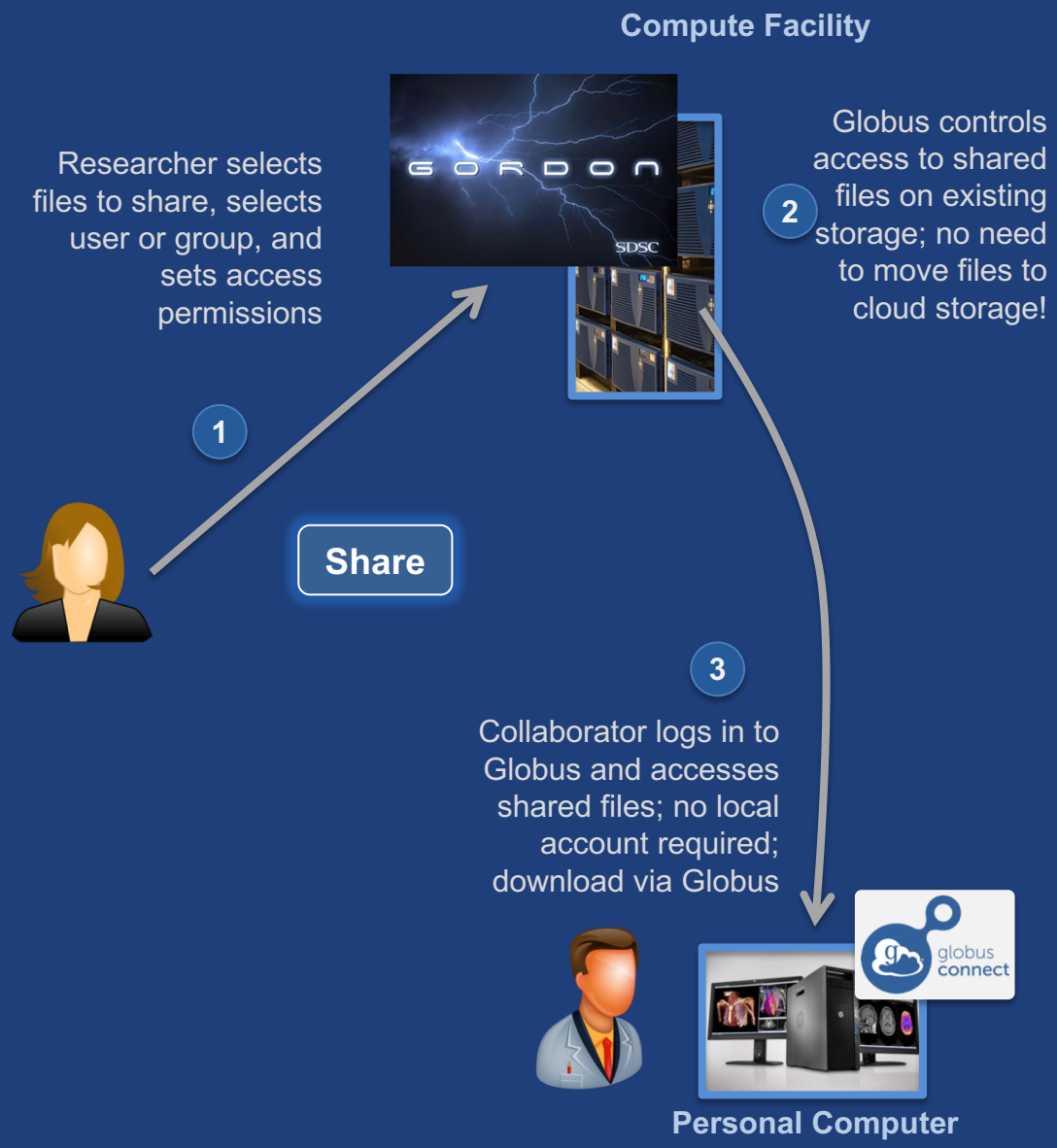
Best Practices for Data Sharing

GlobusWorld 2018

Rachana Ananthakrishnan
rachana@globus.org

Ad hoc data sharing

- **Individual users share data with collaborators**
- **Using email/identity they know for the user**
- **Make data publicly available**



Data from instrument facility

- **Provide near-real time access to data**
- **Access permissions automated based on policy information at site**
- **Self managed by the PI**
- **Federated login to access data**

Raw data store

Local Policy Store

--/cohort045
--/cohort096
--/cohort127

Personal Computer

Remote visualization/analysis

Data from provider/archive

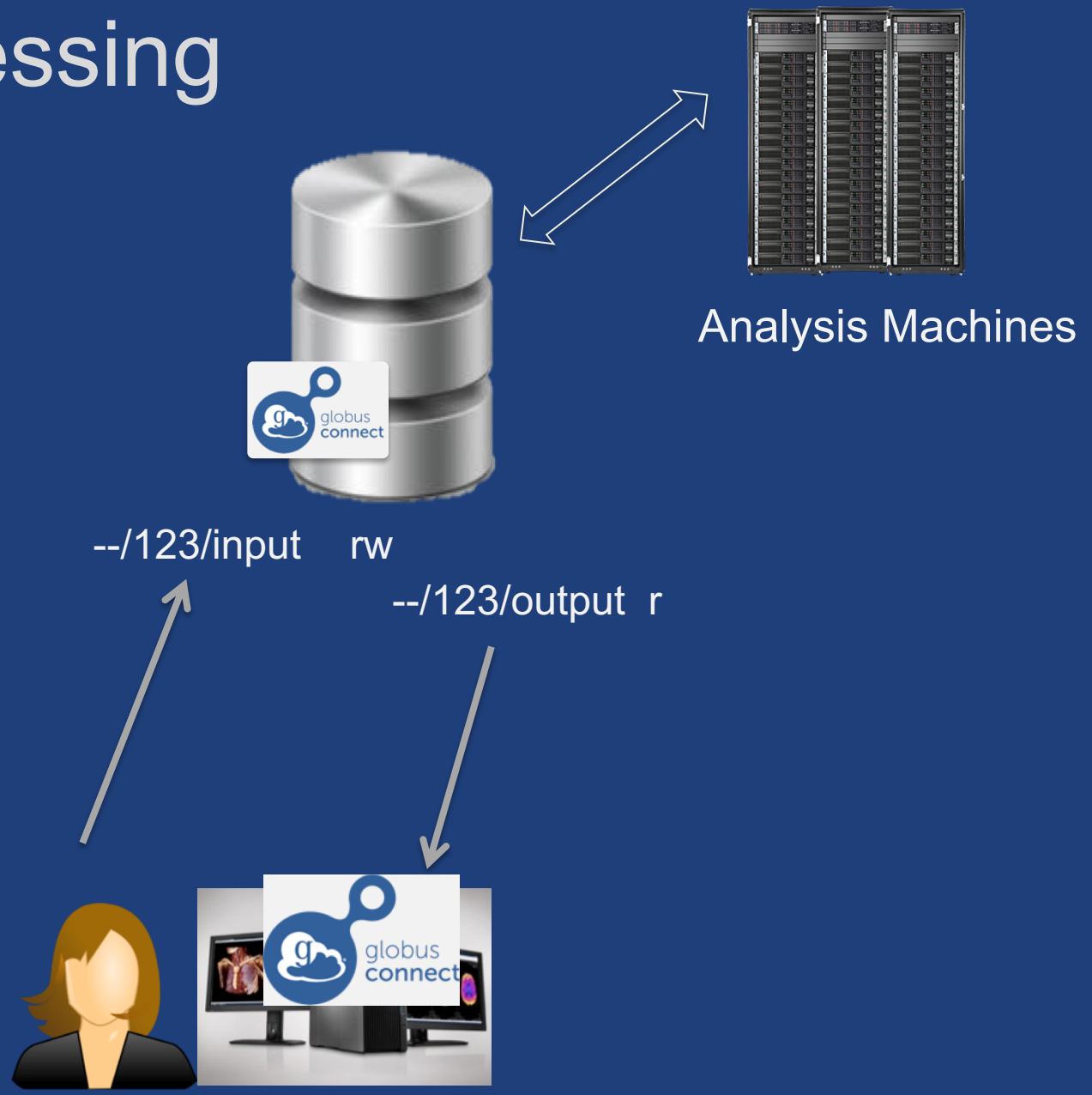
- **Portal/science gateway to distribute data**
- **Interface for search and gathering of data of interest**
- **Data transfer**
  - Asynchronous
  - Directly to the user's machine
  - With authentication & authorization

Transfer data to destination

Search and request data of interest

Core center for processing

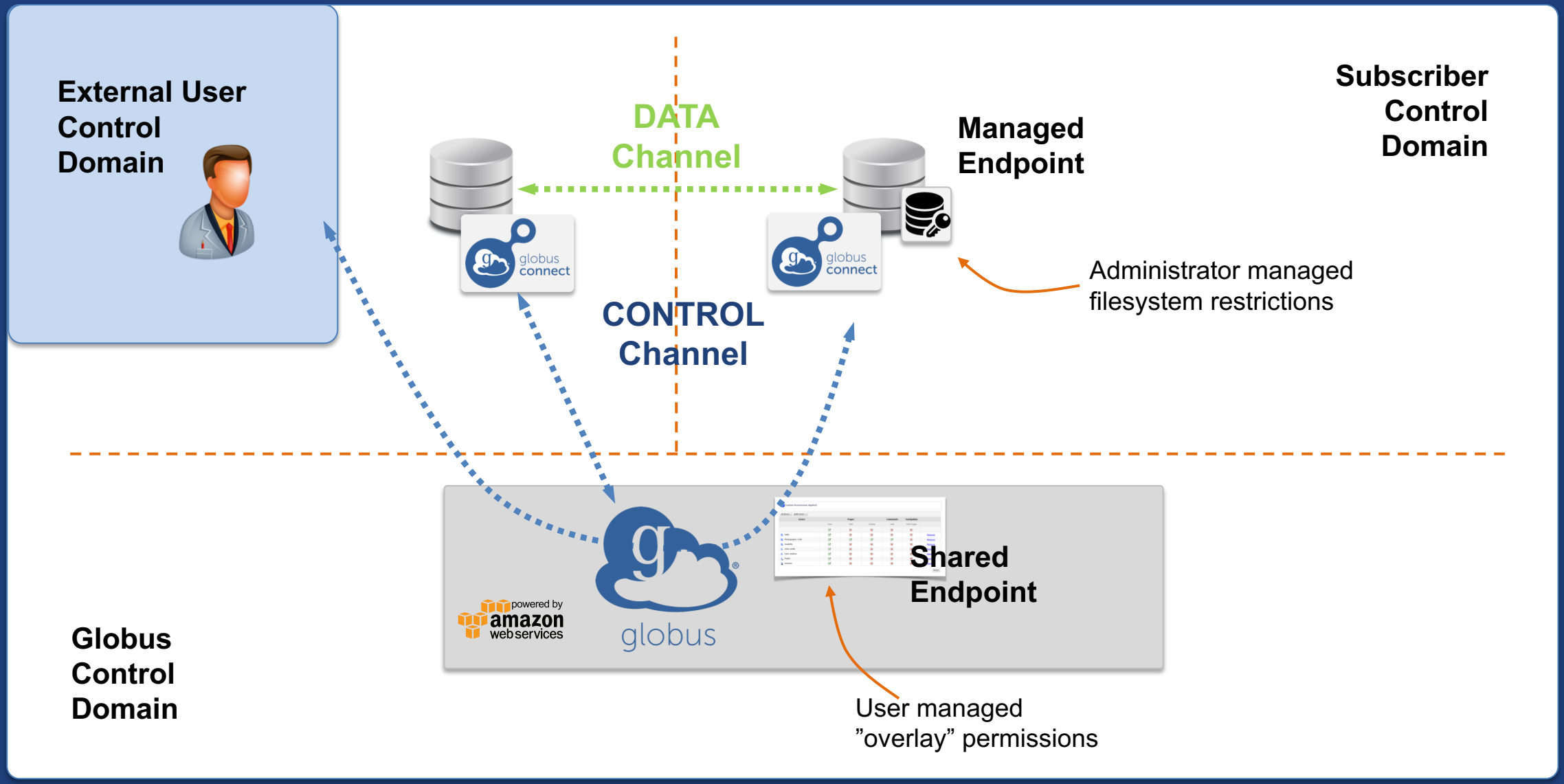
- **Allow user to securely upload data for processing**
- **Post analysis make the user's results available to them**
- **Automate setup and tear down of folders and permissions**



Components in the solution

- **Shared endpoint for the data**
- **An application that can manage permissions**
- **Data transfer, both in and out of the shared endpoint**

Conceptual architecture: Sharing



Data sharing features

- **Shared endpoint creation requires user authentication**
  - Cannot be completely automated
  - Must be a managed endpoint
- **Roles for management of endpoint and tasks**
  - Grant rights to other users, groups or applications
- **Access manager role grants others the rights to manage permissions**
  - Grant to users, groups, applications

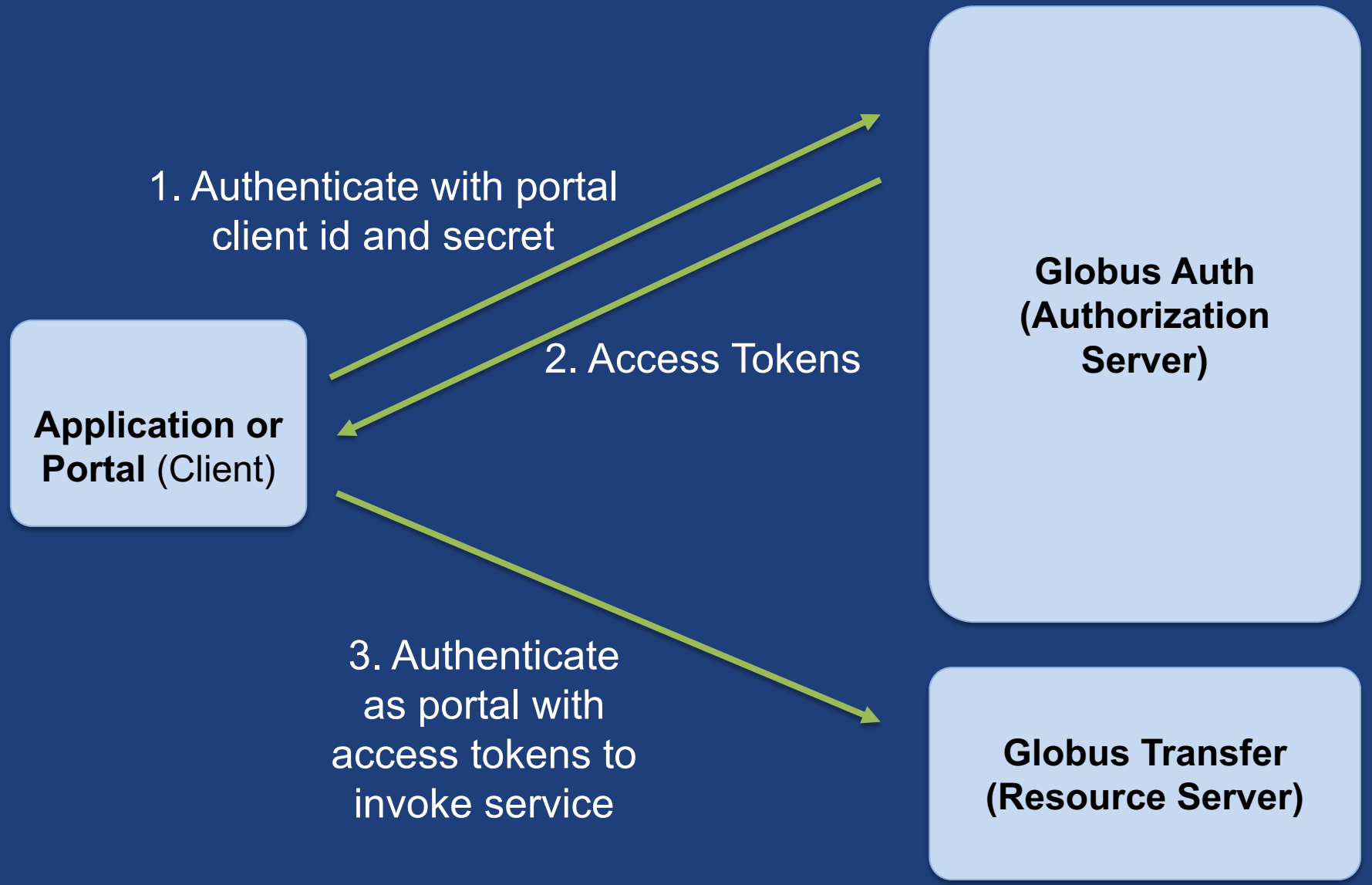
Data sharing features

- **Permissions are per folder on a shared endpoint**
  - Any folder with in the shared endpoint
- **Permissions management can be automated**
- **For a user**
  - Identity: user logs in with that
  - Email: user gets a code to the email, link it to their Globus Account
- **For a group**
  - Search for the group, group UUID is set in the policy
  - Access is then governed by membership to the group
- **For an application**
  - Application identity – appclientid@clients.auth.globus.org

Application concepts

- **Custom application for (automated) permission management**
  - Globus CLI can be used (more in next presentation)
- **Confidential apps**
  - Have client id and secret
  - Ensure the application is on a secure device
  - Setup policy on secret rotation
  - Identity is appclientid@clients.auth.globus.org

Client credential grant



Data transfer

- **Application moving data**
  - Application has access to data on source and can write to destination
  - Requires shared endpoints on both sides
  - Client credential grant
- **Application moving data as user**
  - Only user has access to data on source/destination
  - Authorization code grant
  - Similar to the portal case presented earlier

Walkthrough

Goal

An app that can create folder on shared endpoint, move data and set permissions for some other user

Try this on your own

Modify sample code at https://github.com/globus/automation-examples/blob/master/share_data.py

Shared endpoint

- **Create at top level folder**
- **Set endpoint administrator role**
  - Can change endpoint definition
- **Set endpoint manager role**
  - Can monitor and manage tasks
- **Set endpoint monitor roles**
  - Can monitor tasks

Application registration

- **To manage permissions to automate or run by admin**
- **Register the application with developers.globus.org**
- **Add other users as project administrators**
- **Scopes needed:**
  g*lobus:auth:scope:transfer.api.globus.org:all*
- **Get client id and secret**

Authenticate as application

- **Use client credential grant**
  - Client id and secret used for obtaining tokens
  - Identity username is appclientid@clients.auth.globus.org
- **Create a folder for each user or project**
- **Set permissions at folder**

Support resources

- **Globus documentation: docs.globus.org**
- **Sample code: github.com/globus**
- **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**