Automating Research Data Workflows

ALCF Dev Session February 19, 2020
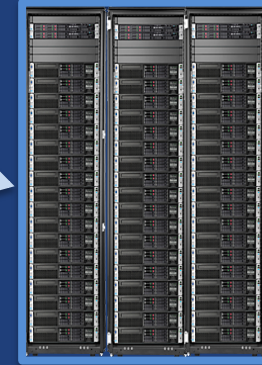
Rick Wagner
rick@globus.org

globus

Data replication

Recurring transfers
with sync option

Copy /ingest
Daily @ 3:30am

- **For backup: initiated by user or system back up**

- **Automated transfer of data from science instrument**

Staging data with compute jobs

- **Stage data in or out as part of the job**

- **Transfer task is submitted when the job is run**
  - Endpoint might not be activated at that time

- **Two approaches:**
  - User adds directives to job submission script
  - An application manages staging of data for user

Automation from portals/gateways

- **Portal/science gateway submits a transfer of compute results as the user**

- **Portal/science gateway monitors users transfer, and initiates processing or backup of data.**
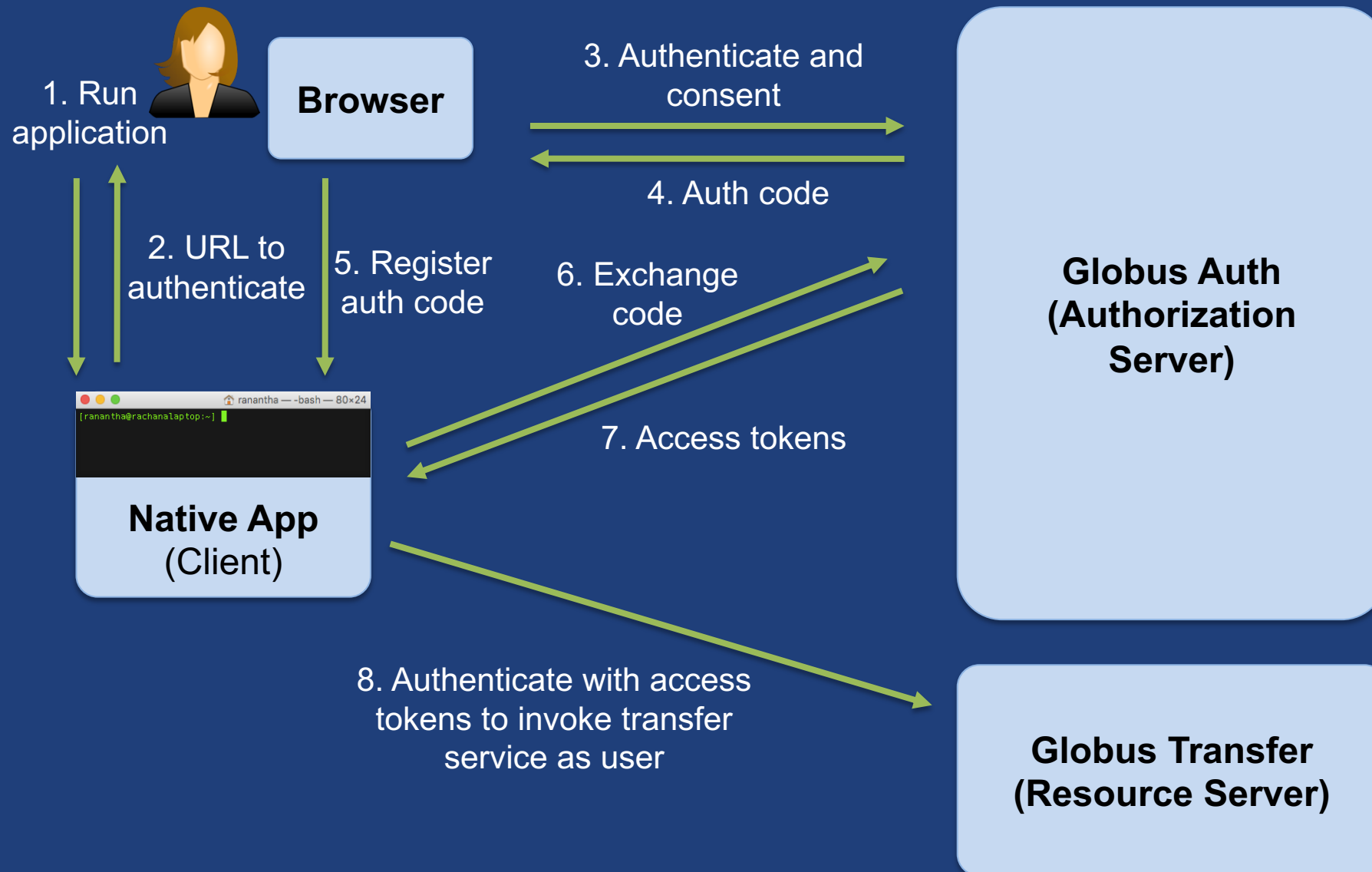
Relevant Platform Capabilities

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

Browser

3. Authenticate and consent

4. Auth code

2. URL to authenticate

5. Register auth code

6. Exchange code

7. Access tokens

Native App (Client)

Globus Auth (Authorization Server)

8. Authenticate with access tokens to invoke transfer service as user

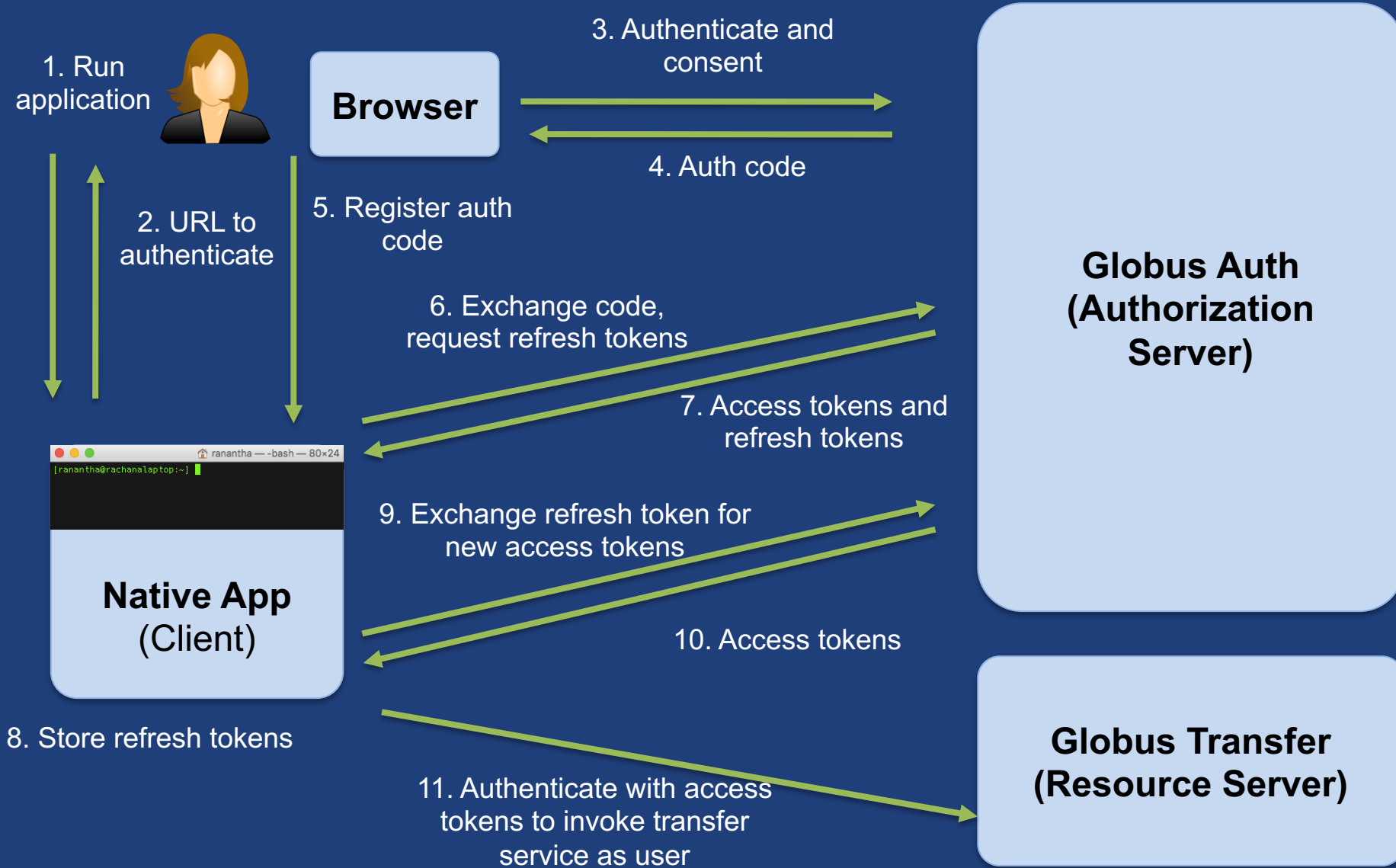**Globus Transfer (Resource Server)**

7

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**
- **Refresh token good for 6 months after last use**
- **Consent rescindment revokes resource token**
- **Globus SDK:**
  - To get token: ConfidentialAppClient or NativeAppClient
  - To use tokens: RefreshTokenAuthorizer

Refresh tokens

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)

Native App/Refresh Tokens Sample Code

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

Using Globus CLI for automation

Globus CLI

- **It's a native application distributed by Globus**

- **Easy install and updates**

- **Command "globus login" gets access tokens and refresh tokens**
  - Stores the token locally (~/.globus.cfg)

- **All interactions with the service use the tokens**
  - Tokens for Globus Auth and Transfer services

- **Command *globus logout* deletes those**

- **See https://docs.globus.org/cli/examples/**

UUIDs everywhere

- **UUIDs for endpoint, task, user identity, groups..**
  - Use search/list options
  - get-identities for identity username to UUID

- *$ globus endpoint search "Globus Tutorial Endpoint"*

- *$ globus task list*

- *$ globus get-identities rachana@globus.org*

Batch Transfers

- A transfer task can only be from one source to one destination, but can have any number of files

- Provide input from local files

- Move files listed in files.txt from Tutorial Endpoint 1 to Tutorial Endpoint 2

  – *$ globus transfer $ep1:/share/godata/ $ep2:~/ \ --batch --label "CLI Batch" < files.txt*

Submission commands

- **Safe resubmissions**
  - Applies to all tasks (transfer and delete)
  - Get a task UUID, use that in submission
  - $ *globus task generate-submission-id*
  - --submission-id option in transfer

- **Task wait**
  - useful for scripting conditional on transfer task status

Parsing CLI output

- **Default is text**
  - $ *globus endpoint search --filter-scope my-endpoints*

- **JSON: --format JSON**
  - $ *globus endpoint search --filter-scope my-endpoints --format JSON*

- **Extract specific : --jmespath <expression>**
  - $ *globus endpoint search --filter-scope my-endpoints -- jmespath 'DATA[0].[id, display_name]'*

Turn off notifications

- **Turn off emails sent for the task**

- **Useful when an application manages tasks for a user**

- **All notifications can be turned off:**
  - *--notify off*

- **Select type of notification to turn off:**
  - *--notify failed, inactive*

Permission management

- **Set and manage permissions on shared endpoint**

- **Need access manager role**

- *$ globus endpoint permission create --permissions r --identity greg@globus.org $share:/forGreg/*

- *$ globus endpoint permission list $share*

- *$ globus endpoint permission delete $share 9edd3956-4736-11e8-8ed2-0a6d4e044368*

Automation with CLI

- **A script that uses the CLI to transfer data repeatedly via task manager/cron**
  - Interactions are as user: both for data access and to Globus services
- **CLI commands used in the job submission script**
  - CLI is installed on head node
  - User runs "globus login", the tokens are stored in user's home directory
  - Tokens accessible when the job runs and submits stage in or stage out tasks
  - Use the –skip-activation-check to submit the task even if endpoint is not activated at submit time

Automation with portals

- **Portal needs to act as the user**

- **User grants "offline" access to the portal**
  - Refresh tokens are obtained for each user by the portal and stored
  - The portal's client id and secret, and refresh tokens are needed to obtain new access tokens
  - Typically state maintained in the portal about transfer initiated or to be managed (task id)

Walkthrough

- **Sample script that use sync option to transfer files between endpoints**

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

- **Sample app that uses Python SDK for the same**

    https://github.com/globus/automation-examples/blob/master/globus_folder_sync.py

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**

- Globus CLI: https://docs.globus.org/cli/
CLI PYPI page: https://pypi.org/project/globus-cli/
Repo with examples:
https://github.com/globus/automation-examples
CLI Batch Recipe: https://github.com/globus/automation-examples/blob/master/batch.md
CLI Sync Example: https://github.com/globus/automation-examples/blob/master/cli-sync.sh
Globus Demo JupyterHub:
https://jupyter.demo.globus.org/