The globusworld TOUR

Automating Research Data Workflows

Rachana Ananthakrishnan - **rachana@globus.org**

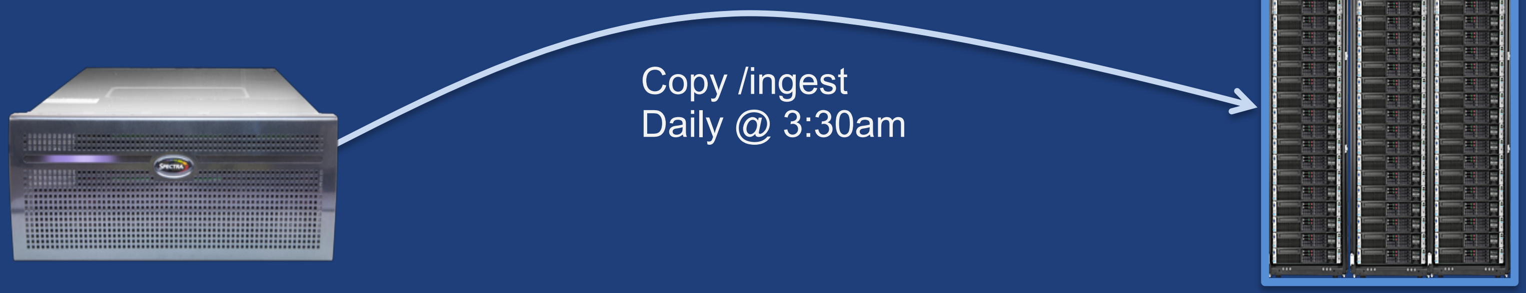
Greg Nawrocki - **greg@globus.org**

Johns Hopkins University
April 11, 2019

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
- Replication to a data share

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 may not be currently activated
- **Alternative approaches**
  1. User adds directives to job submission script
  2. Application manages data staging on user’s behalf

Application driven automation

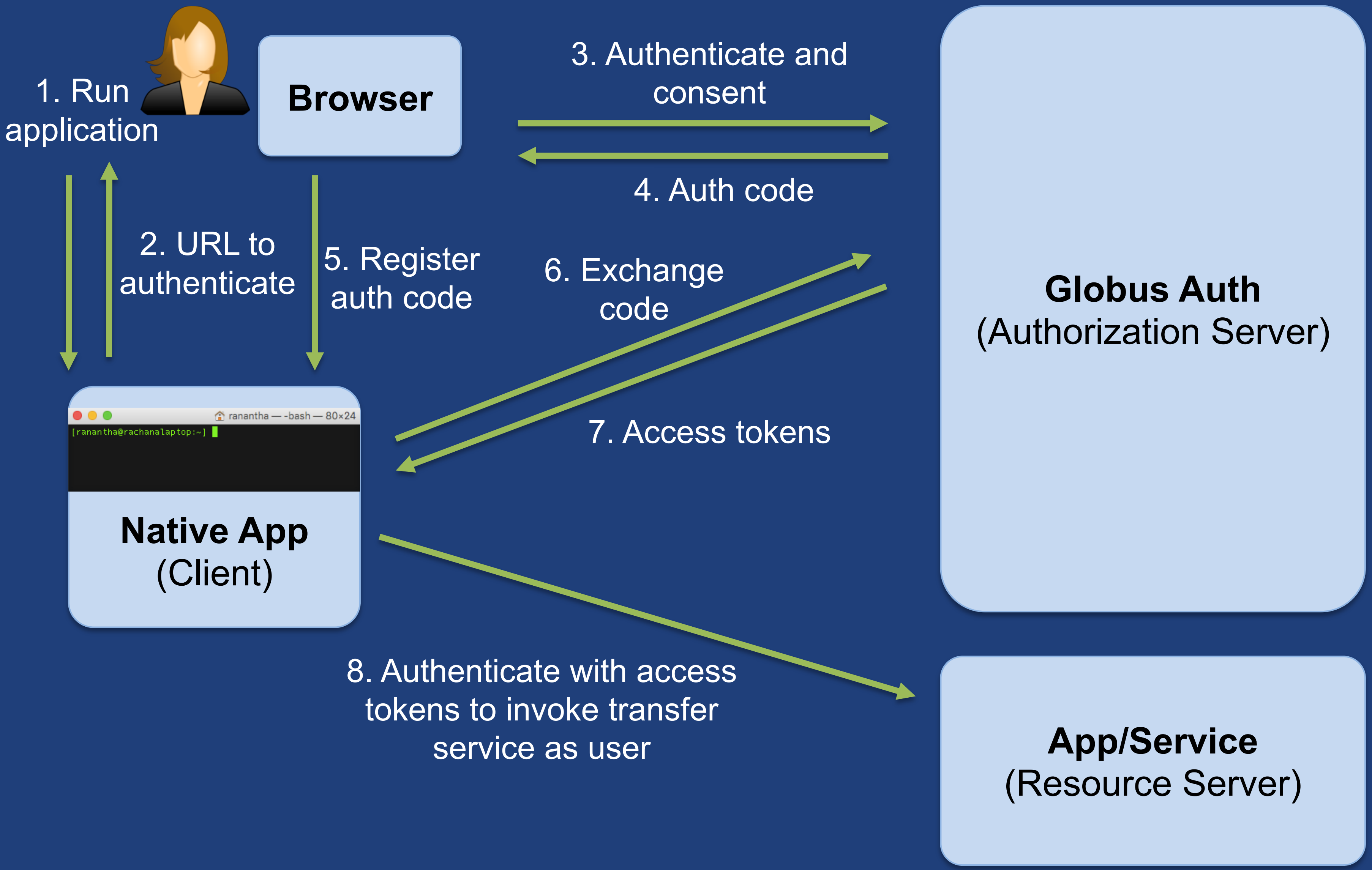
- **Application (e.g. portal, science gateway) submits a transfer of compute results as the user**
- **Application monitors transfer, and initiates additional processing and/or backup of data**

Relevant Platform Capabilities

Globus Auth: Native apps

- **Client that cannot keep a secret, e.g…**
  - Command line, desktop apps
  - Mobile apps
  - Jupyter notebooks
- **Native app is registered with Globus Auth**
  - Not a confidential client like we’ll learn about later
- **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

```
[ranantha@rachanalaptop:~]
```

Native App
(Client)

Globus Auth
(Authorization Server)

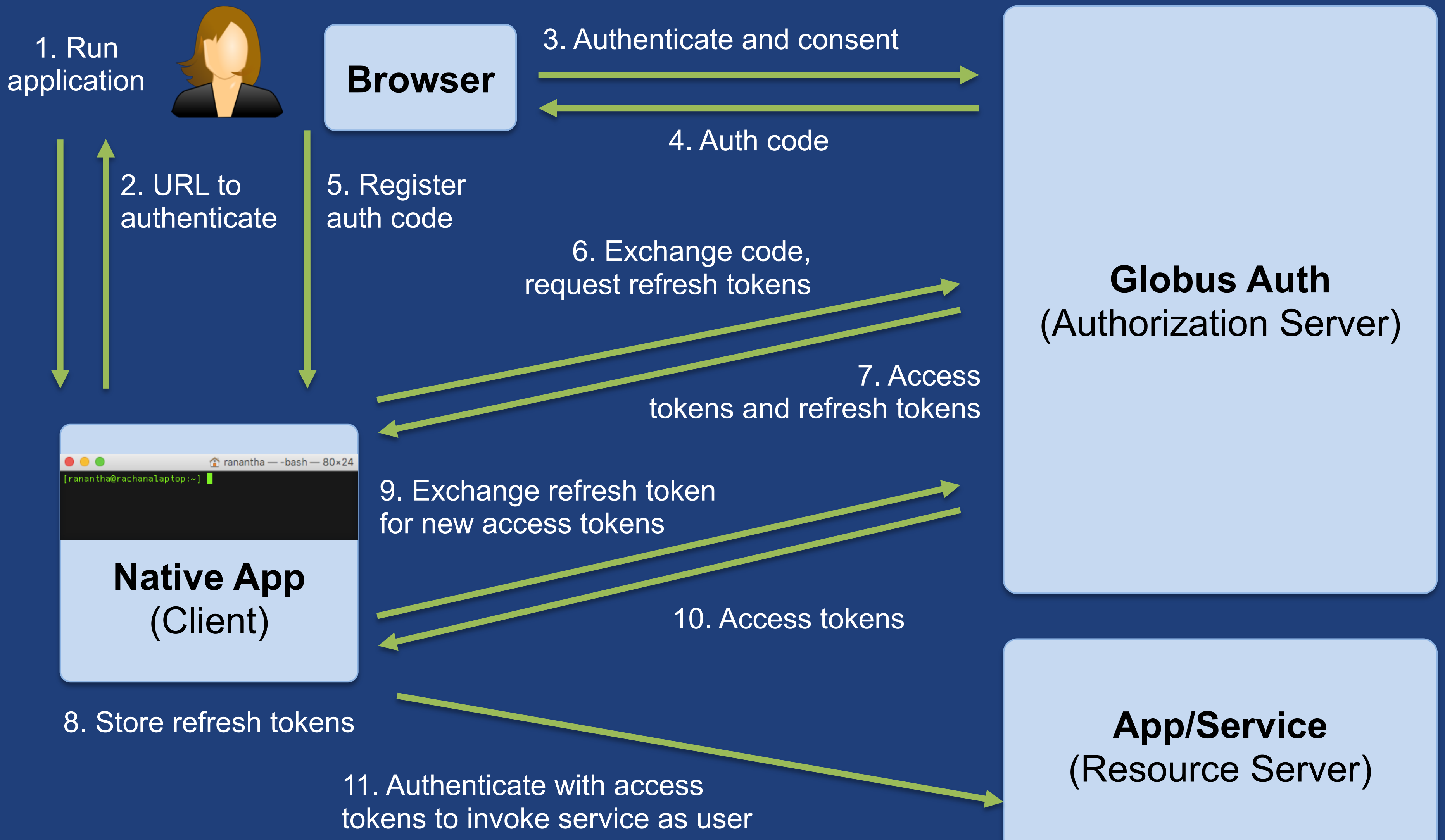
8. Authenticate with access tokens to invoke transfer service as user

App/Service
(Resource Server)

Refresh tokens

- **Common use cases**
  - Portal checking transfer status when user is not logged in
  - Running command line app from script
    - The CLI gets access and refresh tokens upon ”globus login”
- **Refresh tokens issued to client, in 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**

Refresh tokens



1. Run application

Browser

3. Authenticate and consent

4. Auth code

2. URL to authenticate

5. Register auth code

6. Exchange code, request refresh tokens

Globus Auth
(Authorization Server)

7. Access tokens and refresh tokens

```
ranantha — -bash — 80×24
[ranantha@rachanalaptop:~]
```

Native App
(Client)

9. Exchange refresh token for new access tokens

10. Access tokens

8. Store refresh tokens

11. Authenticate with access tokens to invoke service as user

App/Service
(Resource Server)

Native App/Refresh Tokens Sample Code

github.com/globus/native-app-examples

- **`./example_copy_paste.py`**
  - User copies and pastes code to the app
- **`./example_copy_paste_refresh_token.py`**
  - Stores refresh token locally, uses it to get new access tokens
- **See README for installation**

On your EC2 instance in ~/native-app-examples

Automation via the **Globus CLI**

Globus CLI

- **It’s a native application distributed by Globus**
  - https://docs.globus.org/cli/
  - https://github.com/globus/globus-cli
- **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
  - Just like we did in the Platform examples with the API
- **Command *globus logout* deletes those**
- **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'
$ globus task list
$ globus get-identities vas@globus.org bfc122a3-af43-43e1-8a41-d36f28a2bc0a
```

Batch Transfers

- Transfer tasks have one source/destination, but can have any number of files
- Provide input source-dest pairs via local file
- e.g. move files listed in files.txt from $ep1 to $ep2

```
$ ep1=ddb59aef-6d04-11e5-ba46-22000b92c6ec
$ ep2=ddb59af0-6d04-11e5-ba46-22000b92c6ec
$ globus transfer $ep1:/share/godata/ $ep2:/~/ --
batch --label 'CLI Batch' < files.txt
```

Useful 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 output is text; for JSON output use** `--format json`

```
$ globus endpoint search --filter-scope my-endpoints
$ globus endpoint search --filter-scope my-endpoints --
format json
```

- **Extract specific attributes using** `--jmespath <expression>`

```
$ globus endpoint search --filter-scope my-endpoints --
jmespath 'DATA[].[id, display_name]'
```

Managing notifications

- **Turn off emails sent for tasks**
- **Useful when an application manages tasks for a user**
- **Disable notifications with the `--notify` option**

  `--notify off` (all notifications)

  `--notify succeeded|failed|inactive` (select notifications)

Permission management

- **Set and manage permissions on shared endpoint**
- **Requires access manager role**

```
$ share=<shared_endpoint_UUID>
$ globus endpoint permission create --permissions r --
identity greg@nawrockinet.com $share:/nawrockipersonal/
$ globus endpoint permission list $share
$ globus endpoint permission delete $share <perm_UUID>
```

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 Examples

- Syncing a directory
  - Bash script that calls the Globus CLI and a Python module that can be run as a script or imported as a module.
- Staging data in a shared directory
  - Bash / Python
- Removing directories after files are transferred
  - Python script
- Simple code examples for various use cases using Globus
  - https://github.com/globus/automation-examples

Support resources

- **Globus documentation: docs.globus.org**
- **Sample code: github.com/globus**
- **Helpdesk and issue escalation: support@globus.org**
- **Mailing lists**
  - https://www.globus.org/mailing-lists
  - developer-discuss@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**