

justIN 00.40 security model

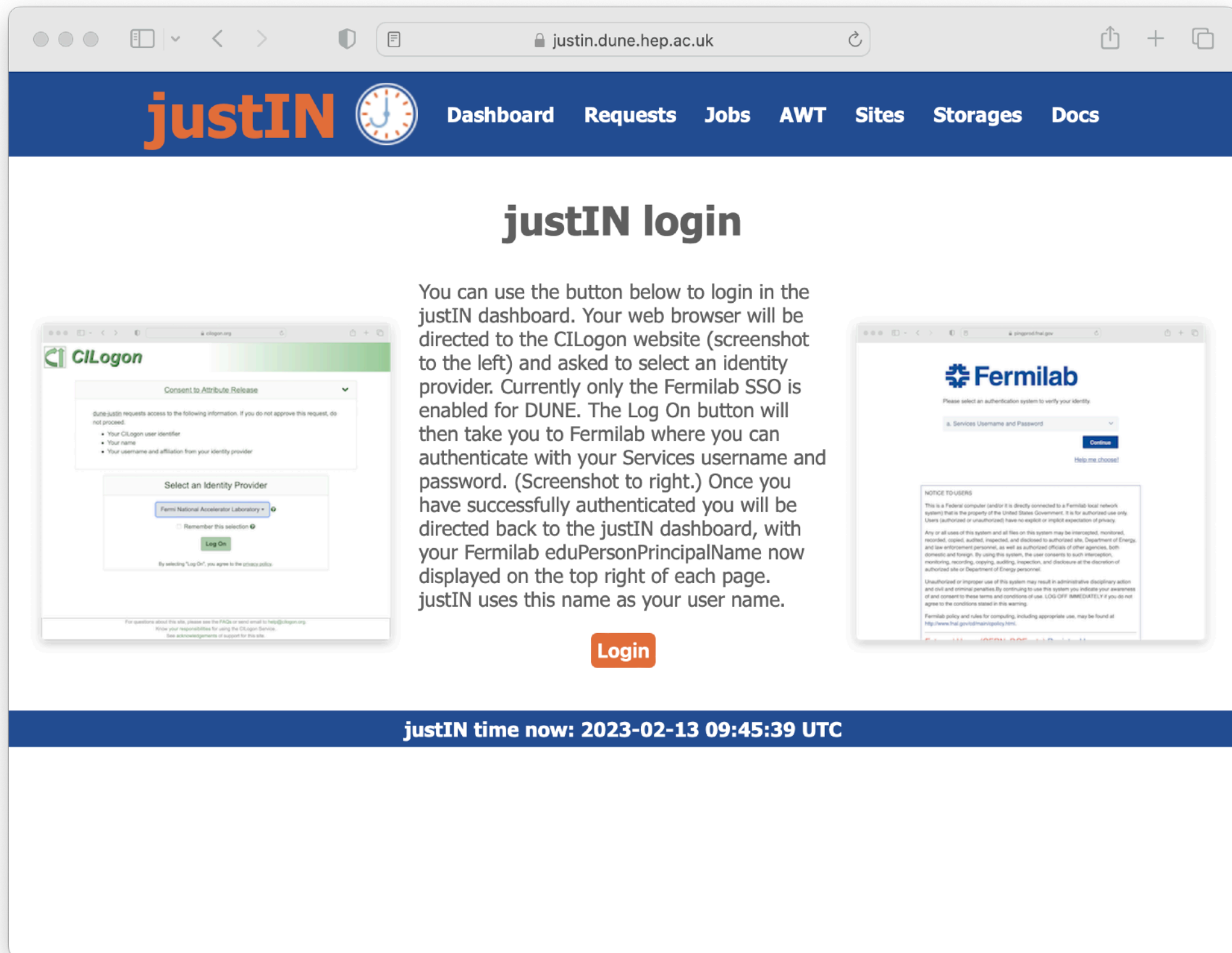
Andrew McNab

University of Manchester

justIN security model

- Quick tour of security model in 00.04
 - This is the version deployed in cvmfs and on the justIN server
- Covering:
 - Logins for the web dashboard
 - Logins for the justin command
 - Scopes access via groups
 - VOMS proxies in jobs
 - Singularity in jobs

Login on the web dashboard



The screenshot shows a web browser window with the URL `justin.dune.hep.ac.uk`. The page features a dark blue header with the **justIN** logo and a navigation menu with links: **Dashboard**, **Requests**, **Jobs**, **AWT**, **Sites**, **Storages**, and **Docs**. The main content area is titled **justIN login** and contains the following text:

You can use the button below to login in the justIN dashboard. Your web browser will be directed to the CILogon website (screenshot to the left) and asked to select an identity provider. Currently only the Fermilab SSO is enabled for DUNE. The Log On button will then take you to Fermilab where you can authenticate with your Services username and password. (Screenshot to right.) Once you have successfully authenticated you will be directed back to the justIN dashboard, with your Fermilab `eduPersonPrincipalName` now displayed on the top right of each page. justIN uses this name as your user name.

Below the text is an orange **Login** button. To the left of the text is a screenshot of the CILogon website showing a "Consent to Attribute Release" dialog and a "Select an Identity Provider" section with "Fermilab National Accelerator Laboratory" selected. To the right is a screenshot of the Fermilab login page with a "Services Username and Password" field and a "Continue" button. At the bottom of the page, a dark blue bar displays the text: **justIN time now: 2023-02-13 09:45:39 UTC**.

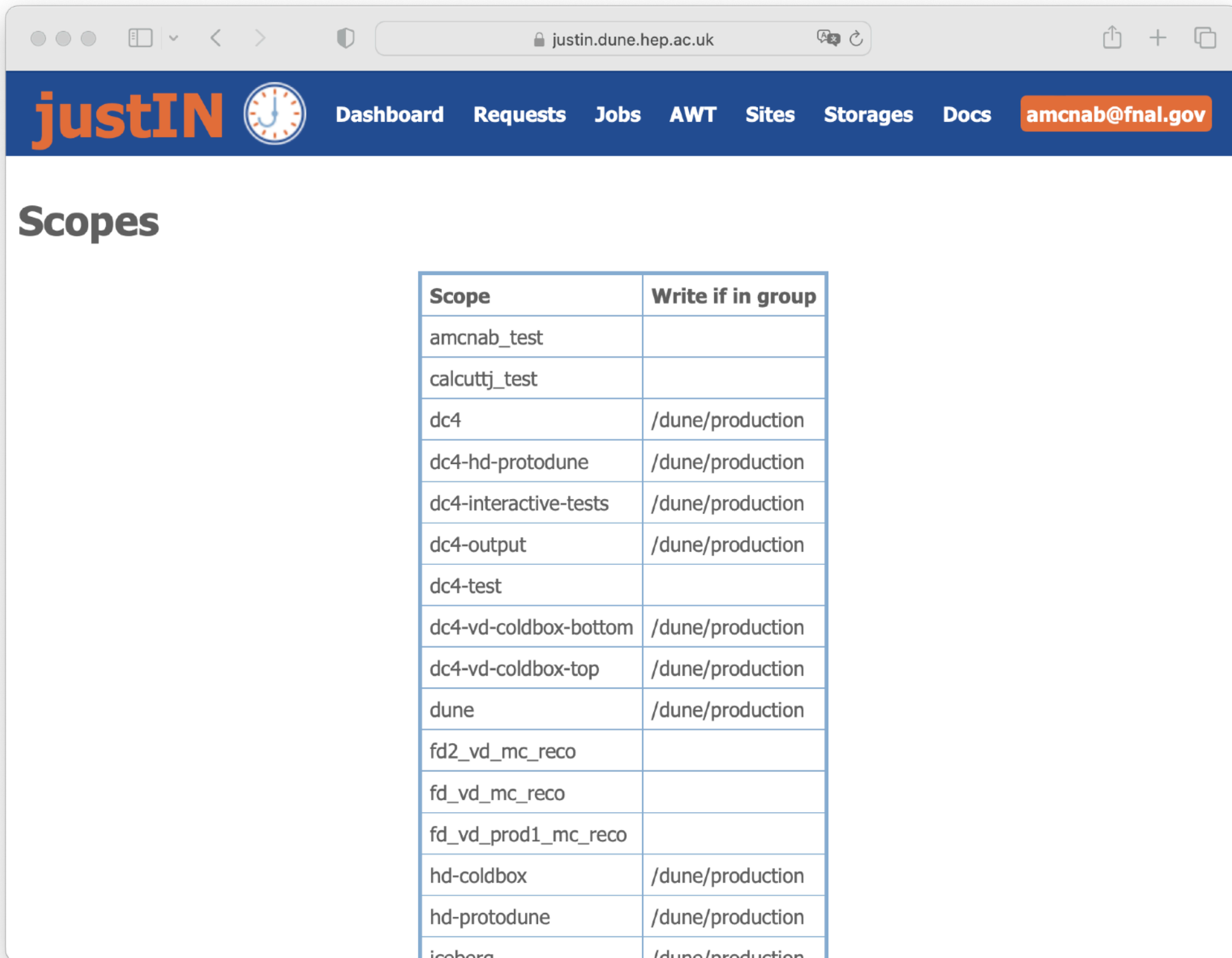
justIN web login

- Login page explains what the user will see
 - With CILogon and Fermilab SSO screenshots
 - Click on the real Login button under the explanation
- CILogon page sends user to Fermilab SSO to login
- Then directed back to justIN, now logged in
- In the background, justIN uses OIDC call to CILogon
- Capture user details and associates them with user's web session in the database
 - eduPersonPrincipalName from SSO (amcnab@fnal.gov)
 - wlcg.groups = /dune, /dune/production, ...
- User browses justIN dashboard, with web cookie, maintaining that session, and allowing the user to act as that user with those groups

justIN command login

- User runs justin command from cvmfs (or downloaded)
- justin contacts justIN services and gets a session ID and secret
 - Puts them in /tmp/justin.session.USERID
- User invited to visit an authorisation page on justIN dashboard
 - Login as for the dashboard itself
 - This time the command's session in the database gets groups and user name
- Next time the justin command is run, the session and secret are presented to the justIN services
- What the user can do is based on the groups stored for the session in the database
- Session is valid for 7 days - same at htgettoken vault tokens

Groups added to Rucio scopes



The screenshot shows a web browser window with the URL `justin.dune.hep.ac.uk`. The page features a navigation bar with the **justIN** logo, a clock icon, and links to **Dashboard**, **Requests**, **Jobs**, **AWT**, **Sites**, **Storages**, and **Docs**. A user profile button shows `amcnab@fnal.gov`. The main content area is titled **Scopes** and contains a table with two columns: **Scope** and **Write if in group**.

| Scope | Write if in group |
|-----------------------|-------------------|
| amcnab_test | |
| calcuttj_test | |
| dc4 | /dune/production |
| dc4-hd-protodune | /dune/production |
| dc4-interactive-tests | /dune/production |
| dc4-output | /dune/production |
| dc4-test | |
| dc4-vd-coldbox-bottom | /dune/production |
| dc4-vd-coldbox-top | /dune/production |
| dune | /dune/production |
| fd2_vd_mc_reco | |
| fd_vd_mc_reco | |
| fd_vd_prod1_mc_reco | |
| hd-coldbox | /dune/production |
| hd-protodune | /dune/production |
| iceberg | /dune/production |

Scopes and groups

- If the user's command session is associated with a group then they can do things to the scopes "owned" by that group
 - For example, for scopes "owned" by /dune/production then users in /dune/production can create files with that scope
- justIN requests can be associated with a scope when they are created
 - Need to do this if you want to create files on Rucio managed storage, which requires giving the scope
- The user who created a request can pause/restart/end it
 - Users with access to the same scope can also do that to the request
 - This is for teams running working group productions etc

VOMS proxies in jobs

- The generic jobs need X.509 proxies with VOMS attributes
 - So user jobscripts can read data from storages
 - So generic job can upload their output files to storages
- Generic job makes 2 keys and Certificate Signing Requests
 - CSRs sent to justIN as part of queries about what work to do
- justIN returns 2 certificates and chains matching the 2 keys
 - One with /Role=Production and one with no roles
 - The generic job assembles these to make 2 VOMS proxy files
- The generic job gives the no roles “readonly” proxy to the user jobscript when it is run: user cannot do uploads / delete
- The generic job will use the production role proxy for uploads
 - Still using the “jobsub” proxy for now - which is the same

Singularity in jobs

- All justIN jobs run inside OSG 3.6 singularity containers on the worker nodes
 - Requested by jobsub_submit and done that way since the autumn
- The generic job now runs the user's jobscript inside another, inner, singularity container, again using the same image tree
- Replacement home directory, /tmp and /var/tmp are created by the generic job
 - Only those (and cvmfs) are mounted by the inner container
 - /home is used to pass in proxies, scripts etc
- So the user's job script cannot go "hunting around" for the generic job's production proxy