CAS

Robert M. Dondero, Ph.D.
Princeton University

CAS

- ***Central Authentication Service (CAS)***

> “The **Central Authentication Service (CAS)** is a single sign-on protocol for the web. **Its purpose is to permit a user to access multiple applications while providing their credentials (such as userid and password) only once. It also allows web applications to authenticate users without gaining access to a user’s security credentials, such as a password.**”
>
> – https://en.wikipedia.org/wiki/Central_Authentication_Service

CAS

- CAS authentication
    - **Part 1**: User logs into CAS server
        - Unnecessary if user is already logged into CAS server
        - User must provide credentials
    - **Part 2**: User logs into PennyAdmin
        - User need not provide credentials

CAS

- See **PennyAdminCas** app

Agenda

- CAS
  - **The code**
  - Running locally
  - Running on Render
  - How it works
  - Assessment

The Code

- See **PennyAdminCas** app (cont.)
  - runserver.py
  - penny.sql, penny.sqlite
  - database.py
  - header.html, footer.html
  - index.html, show.html,
  - add.html, delete.html, reportresults.html
  - top.py, penny.py, auth.py

Agenda

- CAS
  - The code
  - **Running locally**
  - Running on Render
  - How it works
  - Assessment

Running Locally

- See **PennyAdminCas** app (cont.)
  - How to run it on your local computer…

Running Locally

1. Create and activate an appropriate Python virtual environment. To do that, issue these commands:

```
mkdir ~/.virtualenvs
python -m venv ~/.virtualenvs/somedir
source ~/.virtualenvs/somedir/bin/activate
```

2. Install modules into your Python virtual environment. To do that, in your app directory issue this command:

```
python -m pip install -r requirements.txt
```

3. Run the server. To do that, in your app directory issue this command:

```
python runserver.py someport
```

Running Locally

4. Browse to **http://localhost:someport**.

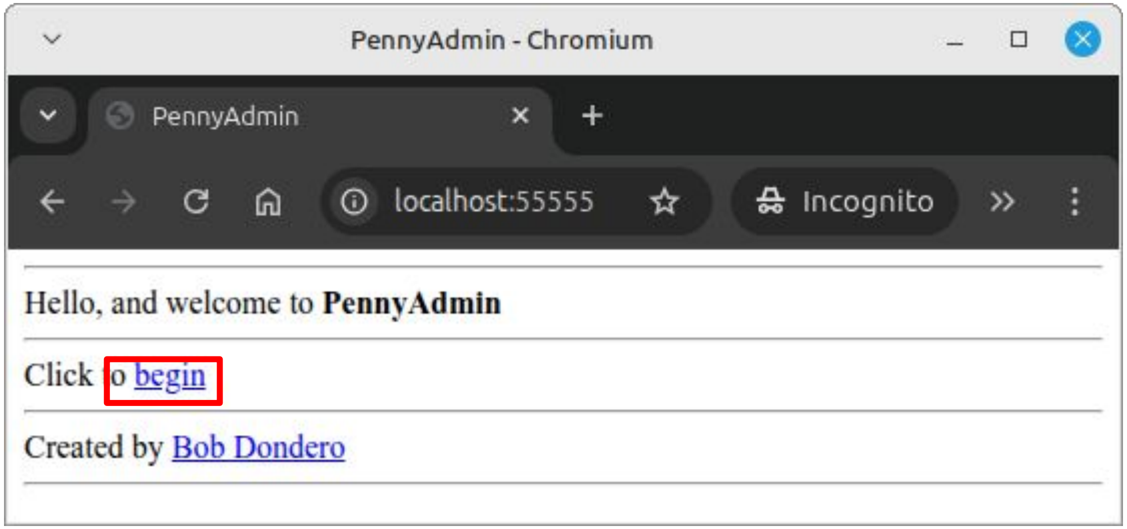
    4.1. You must use **localhost** as the host. Using the real IP address of your computer won't work. Using **127.0.0.1** won't work.

5. Authenticate using CAS.

6. When prompted for an author name, enter Ker.

Running Locally

- See **PennyAdminCas** app (cont.)



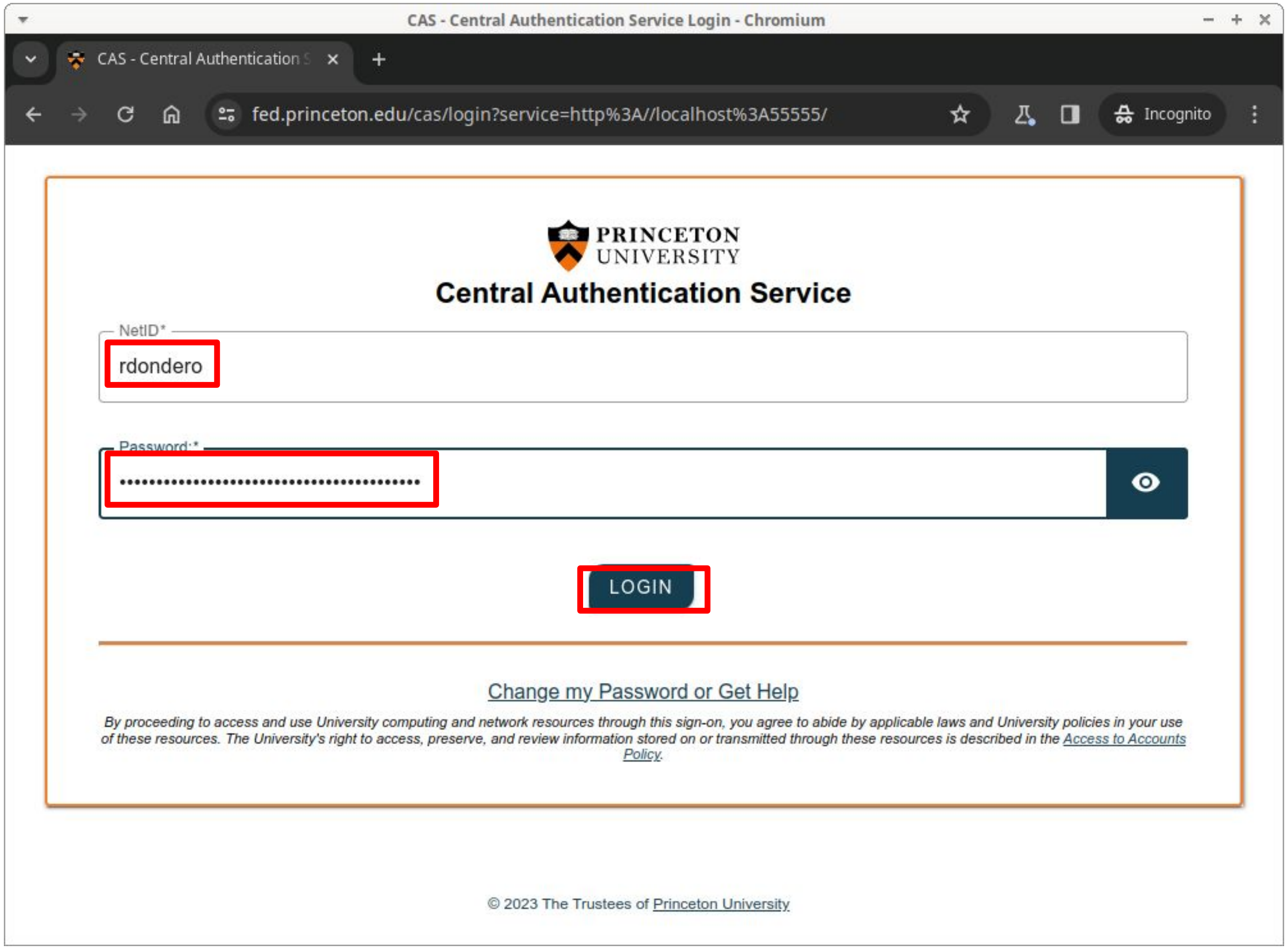
Hello, and welcome to **PennyAdmin**

Click to begin

Created by Bob Dondero

Running Locally

- See **PennyAdminCas** app (cont.)



CAS - Central Authentication Service Login - Chromium
CAS - Central Authentication S
fed.princeton.edu/cas/login?service=http%3A//localhost%3A55555/
Incognito
PRINCETON UNIVERSITY

Central Authentication Service

NetID* rdondero

Password:* ••

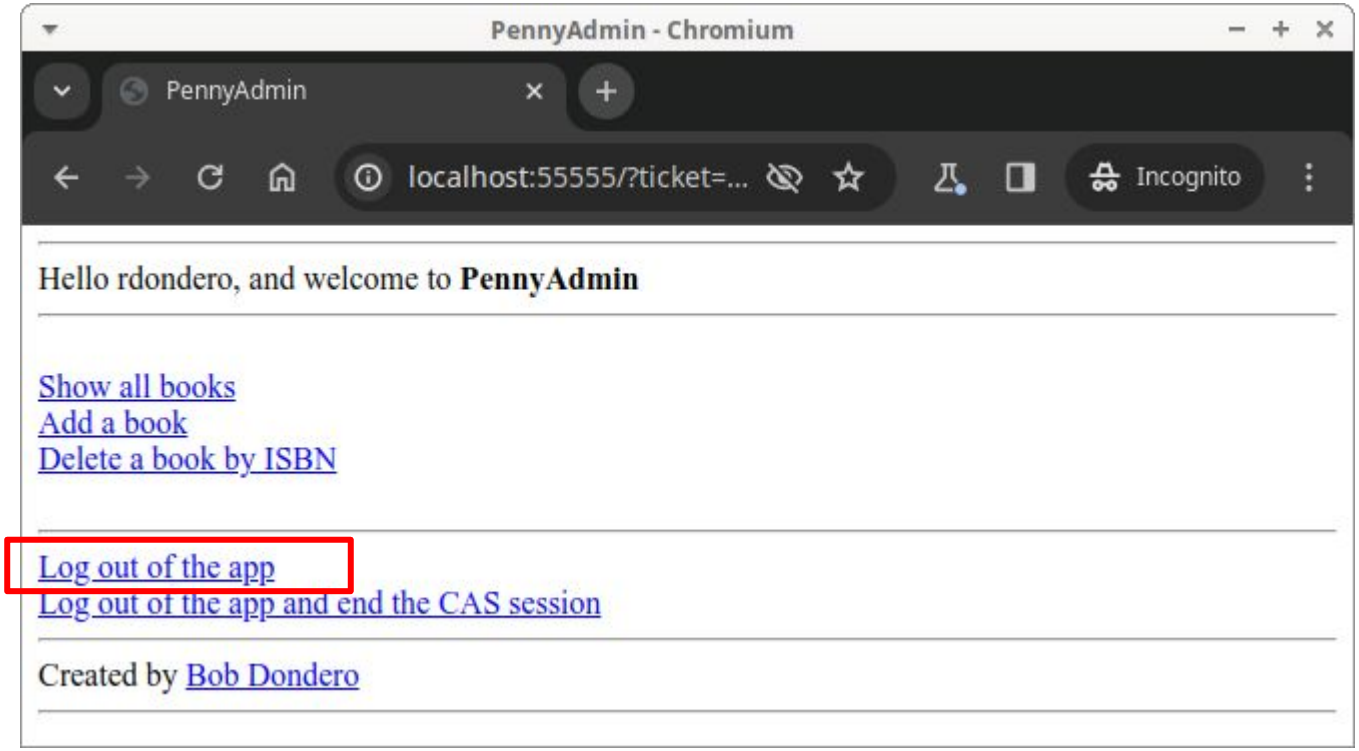
Change my Password or Get Help

By proceeding to access and use University computing and network resources through this sign-on, you agree to abide by applicable laws and University policies in your use of these resources. The University's right to access, preserve, and review information stored on or transmitted through these resources is described in the Access to Accounts Policy.

© 2023 The Trustees of Princeton University

Running Locally

- See **PennyAdminCas** app (cont.)



Hello rdondero, and welcome to **PennyAdmin**

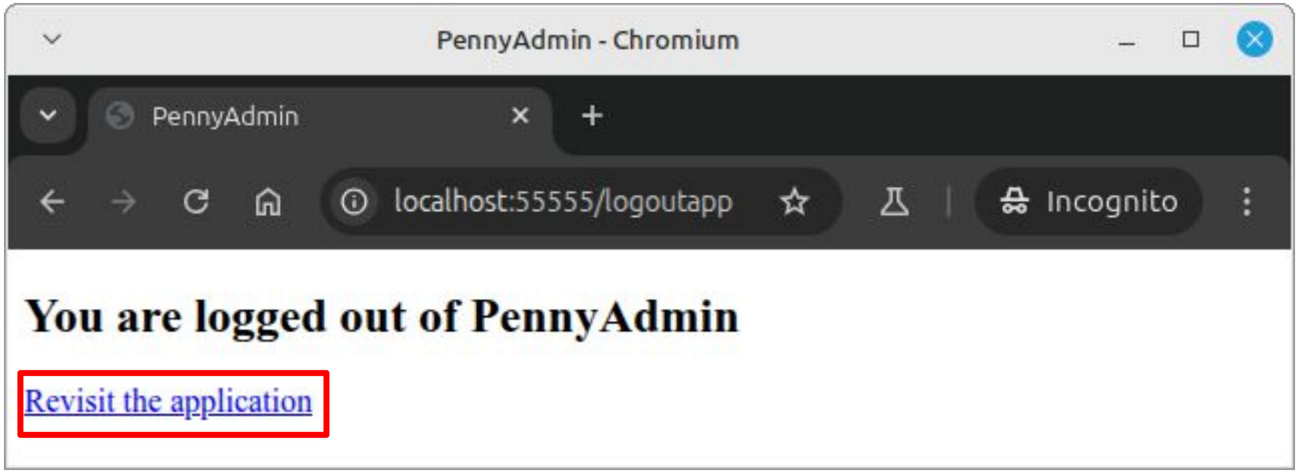
Show all books
Add a book
Delete a book by ISBN

Log out of the app
Log out of the app and end the CAS session

Created by Bob Dondero

Running Locally

- See **PennyAdminCas** app (cont.)

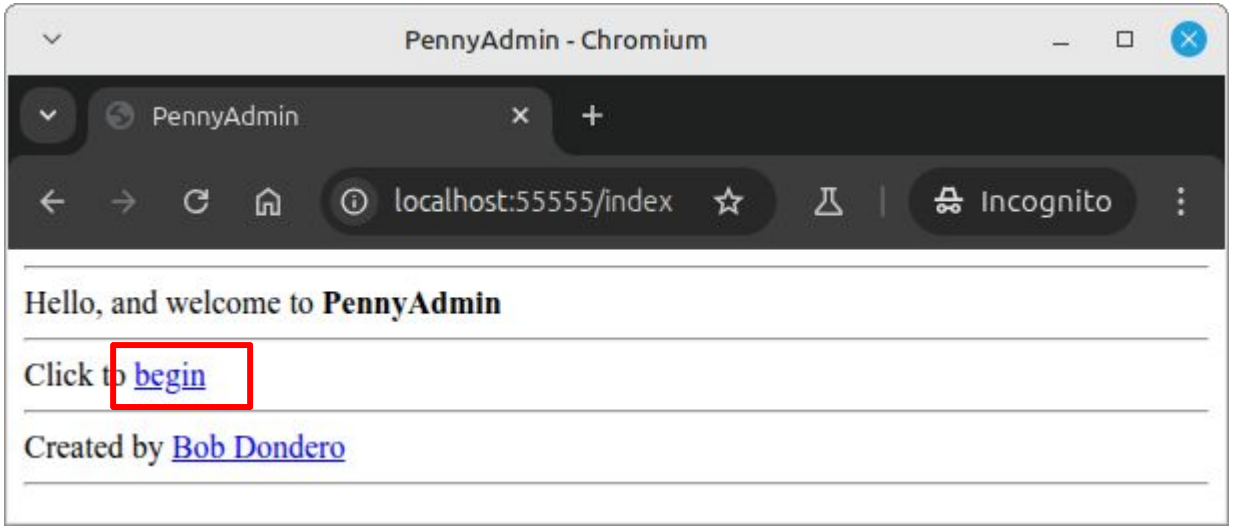


You are logged out of PennyAdmin

Revisit the application

Running Locally

- See **PennyAdminCas** app (cont.)



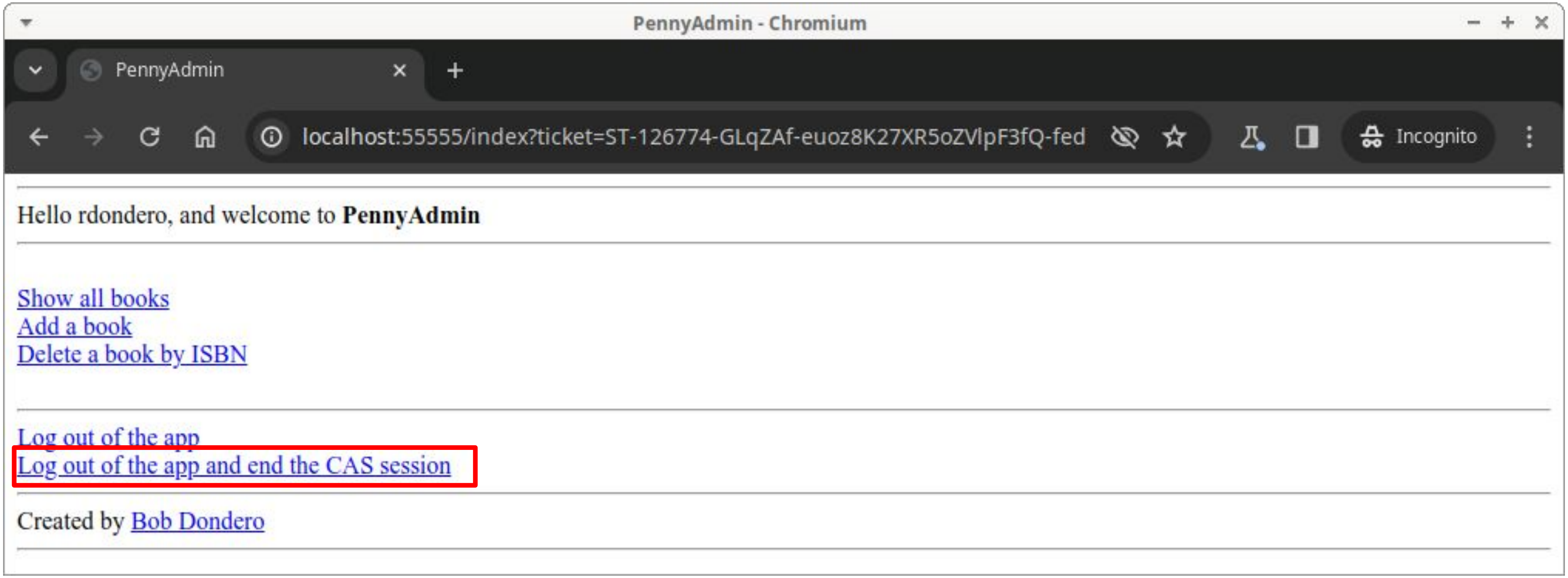
Hello, and welcome to **PennyAdmin**

Click to begin

Created by Bob Dondero

Running Locally

- See **PennyAdminCas** app (cont.)



PennyAdmin - Chromium
PennyAdmin
localhost:55555/index?ticket=ST-126774-GLqZAf-euoz8K27XR5oZVlpF3fQ-fed
Incognito

Hello rdondero, and welcome to **PennyAdmin**

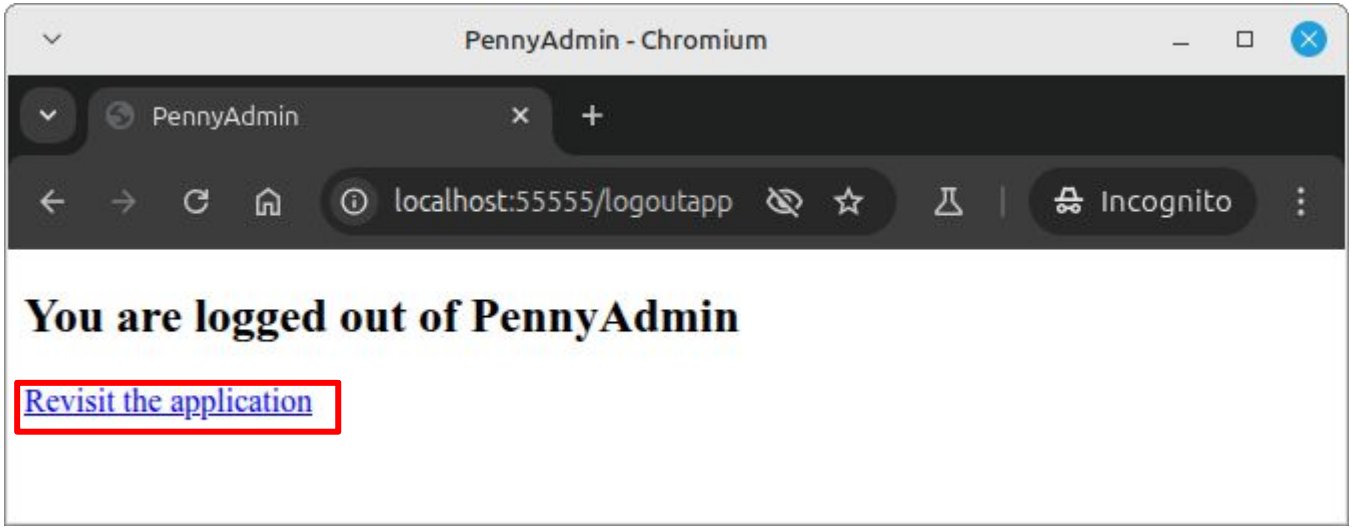
Show all books
Add a book
Delete a book by ISBN

Log out of the app
Log out of the app and end the CAS session

Created by Bob Dondero

Running Locally

- See **PennyAdminCas** app (cont.)

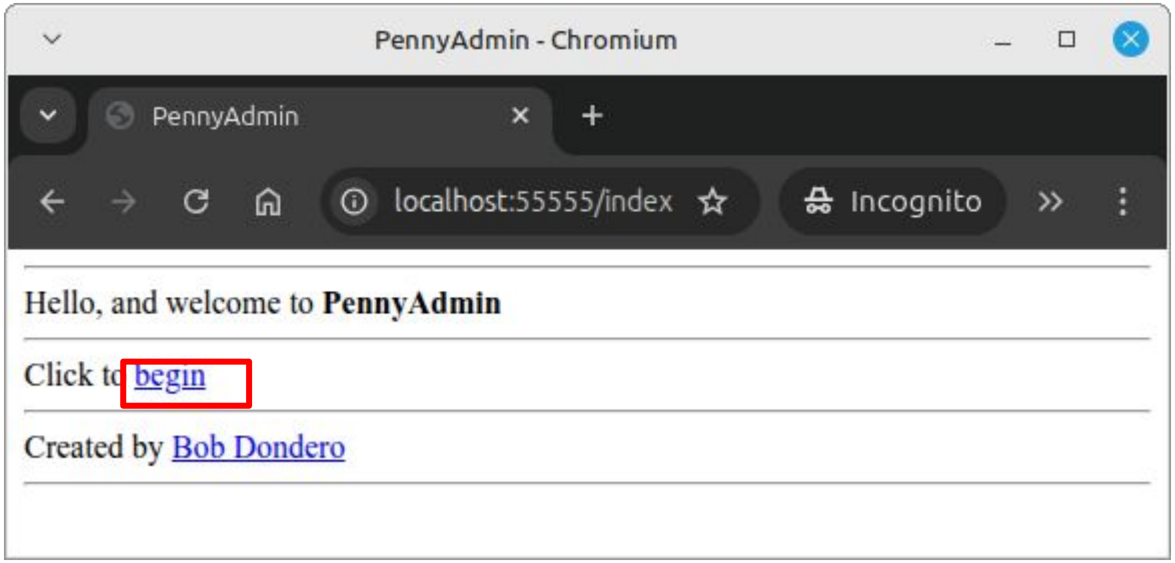


You are logged out of PennyAdmin

Revisit the application

Running Locally

- See **PennyAdminCas** app (cont.)



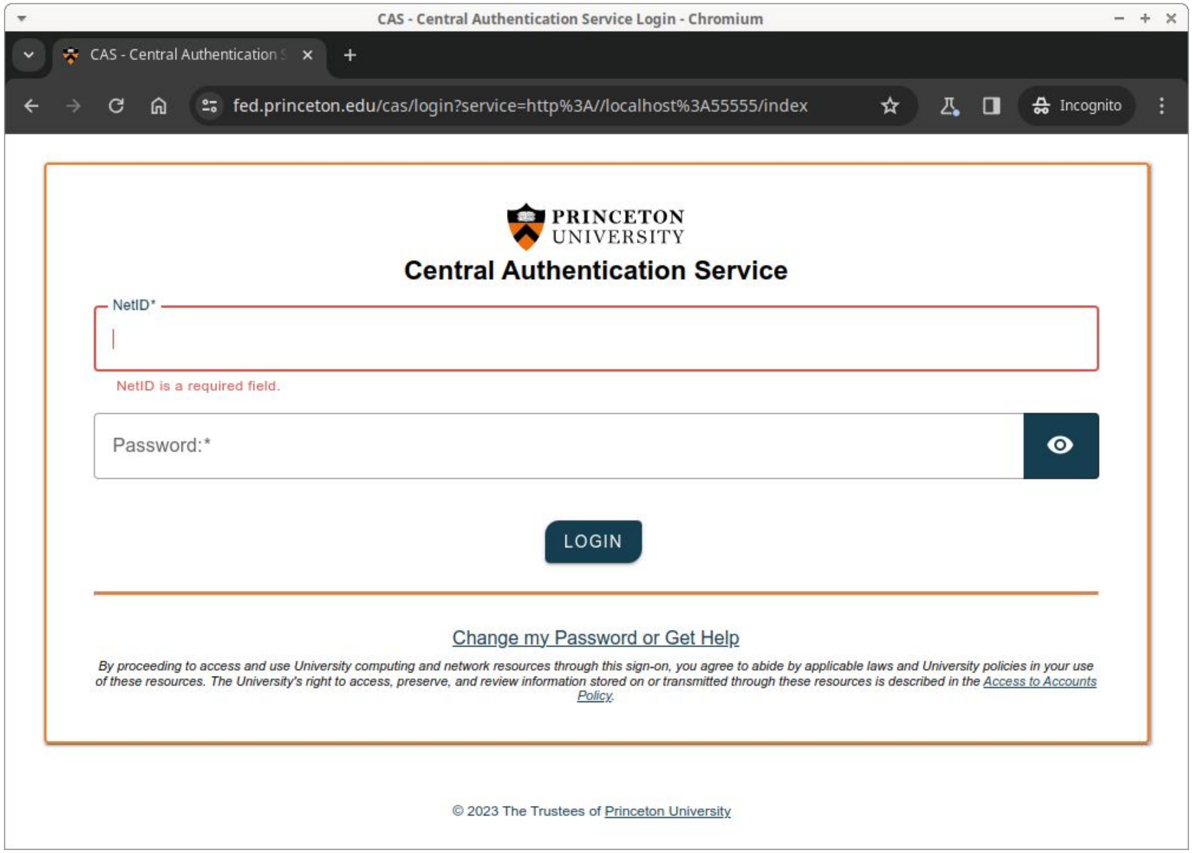
Hello, and welcome to **PennyAdmin**

Click to begin

Created by Bob Dondero

Running Locally

- See **PennyAdminCas** app (cont.)



CAS - Central Authentication Service Login - Chromium

CAS - Central Authentication S

fed.princeton.edu/cas/login?service=http%3A//localhost%3A55555/index

Central Authentication Service

NetID*

NetID is a required field.

Password:*

Change my Password or Get Help

By proceeding to access and use University computing and network resources through this sign-on, you agree to abide by applicable laws and University policies in your use of these resources. The University's right to access, preserve, and review information stored on or transmitted through these resources is described in the Access to Accounts Policy.

© 2023 The Trustees of Princeton University

Agenda

- CAS
  - The code
  - Running locally
  - **Running on Render**
  - How it works
  - Assessment

Running on Render

- See **PennyAdminCas** app (cont.)
  - How to deploy it to Render and run it there…

Running on Render

1. Deploy the app to Render as a **Web Service**.

    1.1. Provide this as the build command:

    pip install -r requirements.txt

    1.2. Provide this as the start command:

    gunicorn penny:app

2. Register your Render app with Princeton's OIT.

    2.1. Ask OIT to place your Render app, as identified by its URL, on the Princeton CAS white list. The COS 333 *Princeton Data Sources* web page describes how to do that.

3. Browse to **https://yourappname.onrender.com**.

4. Authenticate using CAS.

5. When prompted for an author name, enter Ker.

Agenda

- CAS
  - The code
  - Running locally
  - Running on Render
  - **How it works**
  - Assessment

How It Works

- See **PennyAdminCas** app (cont.)
  - How it works…

How It Works

- Procedure
  - **Part 1**: User logs into CAS server
    - User must provide credentials
  - **Part 2**: User logs into PennyAdmin
    - User need not provide credentials

How It Works

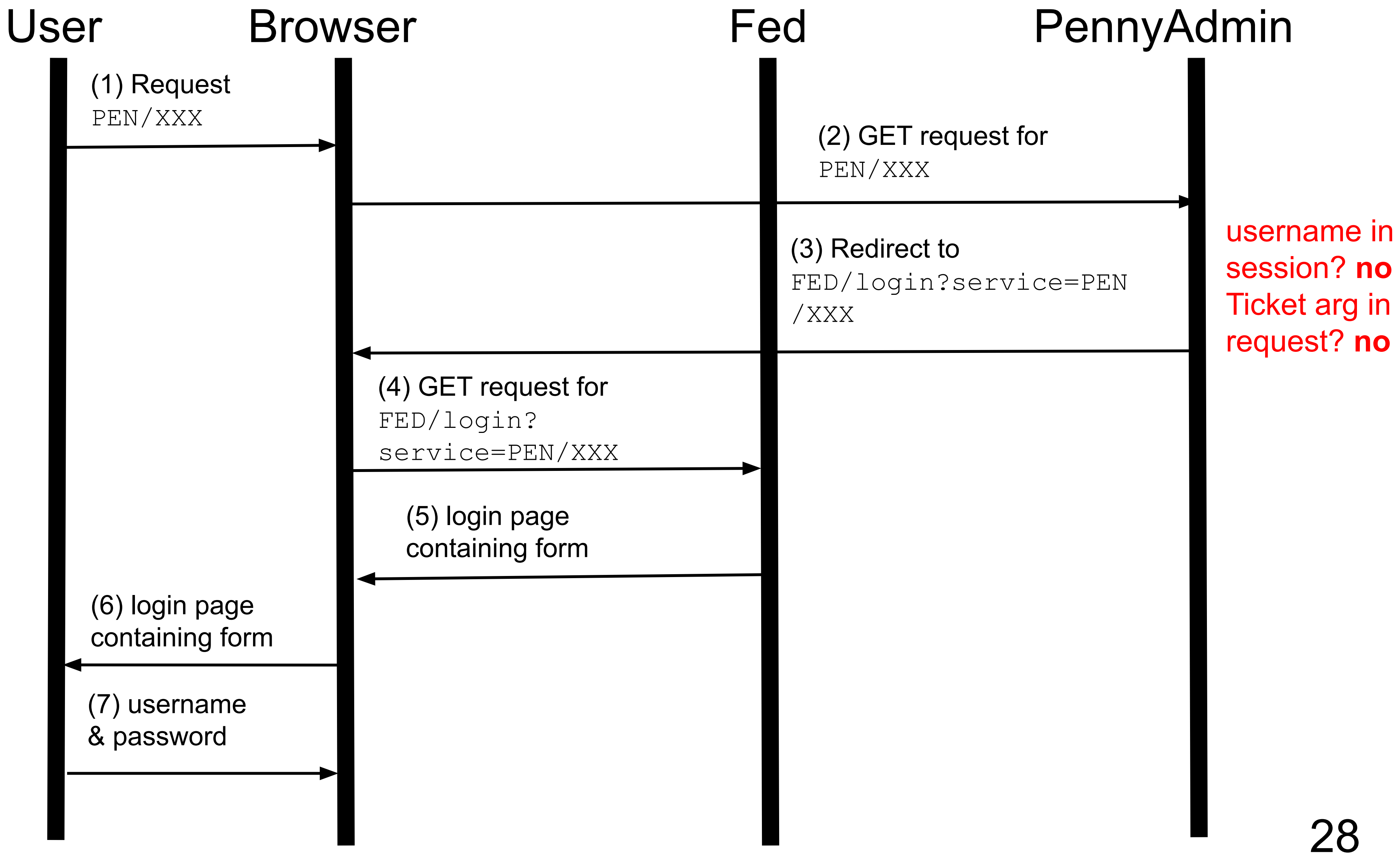
- See **PennyAdmin16Cas** app (cont.)
  - The flow…

Abbreviations:
  FED = https://fed.princeton.edu/cas
  PEN = https://localhost:55555

How It Works

First use of PennyAdmin in browser session, browser session not CAS authenticated…

How It Works



User

Browser

Fed

PennyAdmin

(1) Request `PEN/XXX`

(2) GET request for `PEN/XXX`

username in session? **no**
Ticket arg in request? **no**

(3) Redirect to `FED/login?service=PEN/XXX`

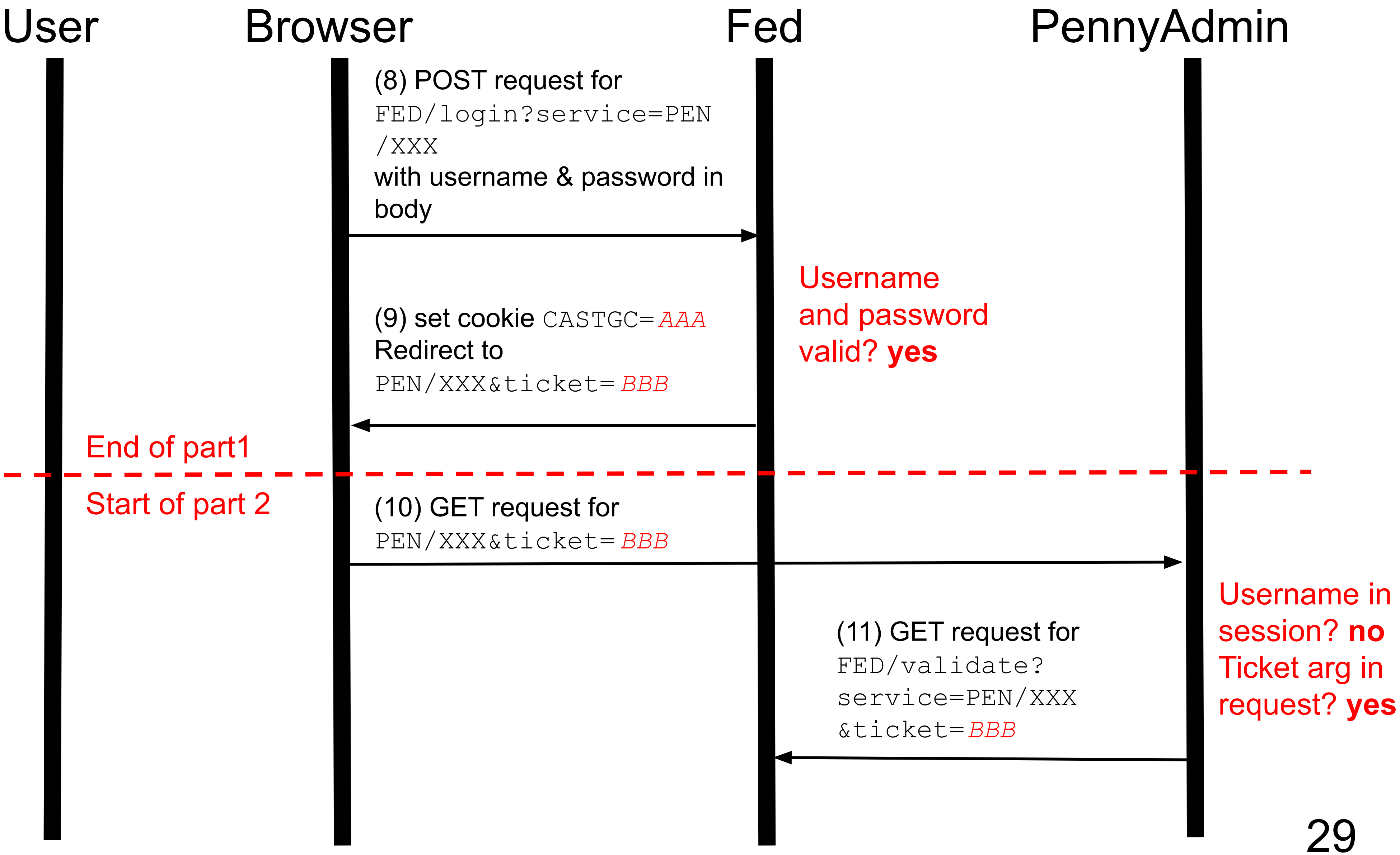
(4) GET request for `FED/login?service=PEN/XXX`

(5) login page containing form

(6) login page containing form

(7) username & password

How It Works



User

Browser

Fed

PennyAdmin

(8) POST request for FED/login?service=PEN/XXX with username & password in body

Username and password valid? **yes**

(9) set cookie CASTGC=*AAA*
Redirect to PEN/XXX&ticket=*BBB*

End of part1

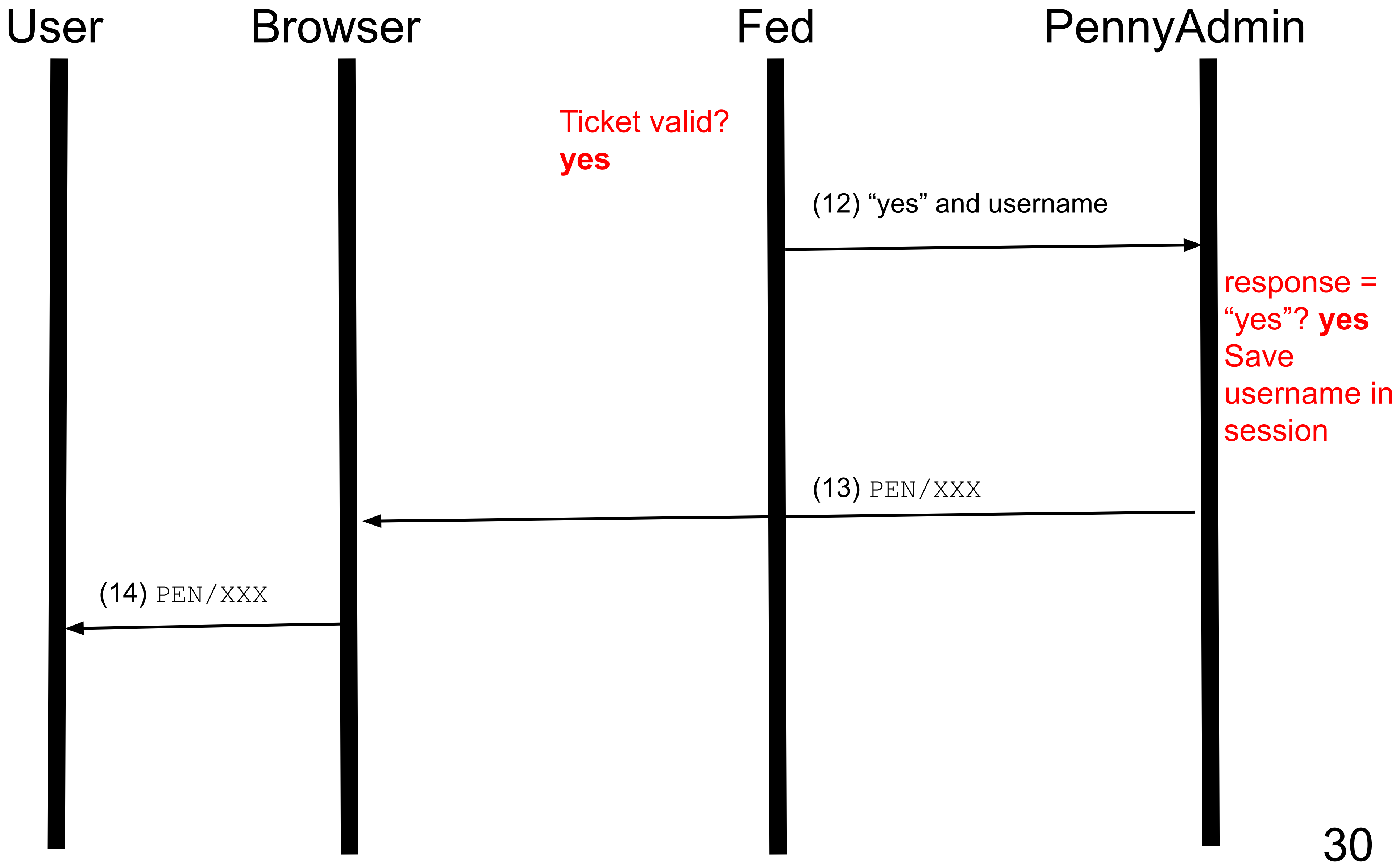
Start of part 2

(10) GET request for PEN/XXX&ticket=*BBB*

Username in session? **no**
Ticket arg in request? **yes**

(11) GET request for FED/validate?service=PEN/XXX&ticket=*BBB*

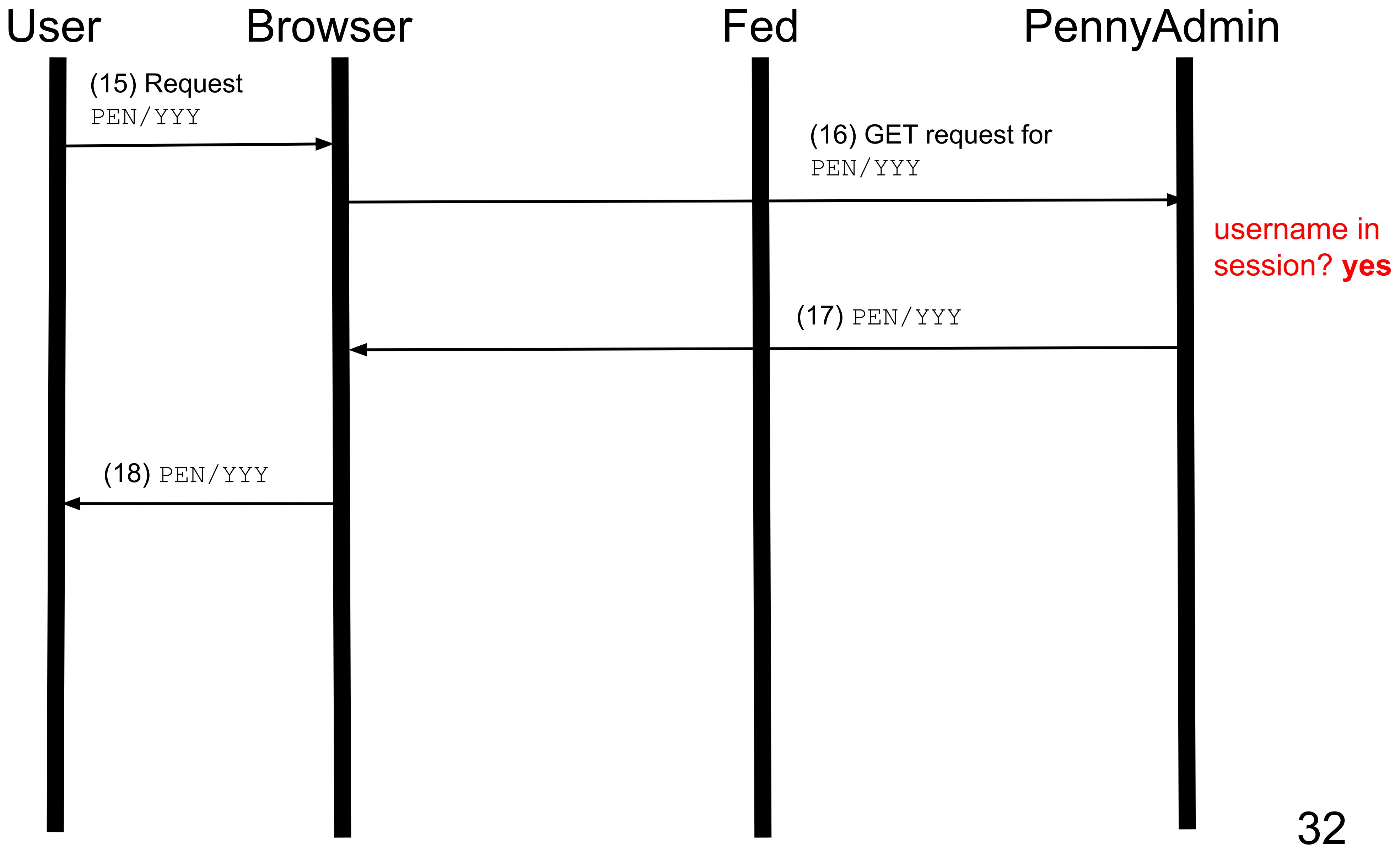
How It Works


User
Browser
Fed
PennyAdmin
Ticket valid?
yes
(12) “yes” and username
response = “yes”? **yes**
Save username in session
(13) PEN/XXX
(14) PEN/XXX

How It Works

Second use of PennyAdmin in same browser session…

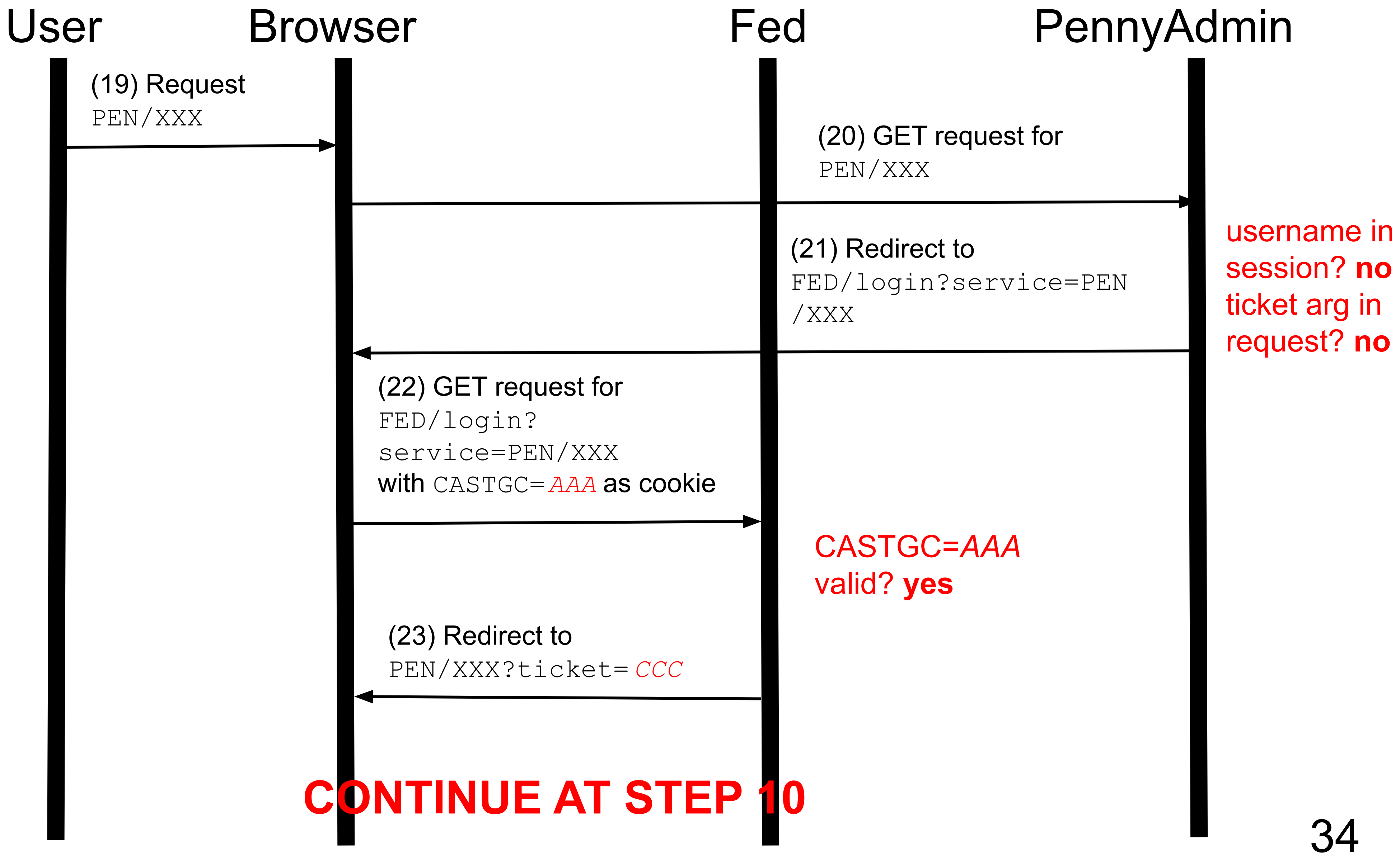
How It Works


User
Browser
Fed
PennyAdmin
(15) Request PEN/YYY
(16) GET request for PEN/YYY
username in session? **yes**
(17) PEN/YYY
(18) PEN/YYY

How It Works

First use of PennyAdmin in browser session, browser session already CAS authenticated…

How It Works



User

Browser

Fed

PennyAdmin

(19) Request `PEN/XXX`

(20) GET request for `PEN/XXX`

(21) Redirect to `FED/login?service=PEN/XXX`

username in session? **no**
ticket arg in request? **no**

(22) GET request for `FED/login?service=PEN/XXX` with `CASTGC=`*`AAA`* as cookie

CASTGC=*AAA* valid? **yes**

(23) Redirect to `PEN/XXX?ticket=`*`CCC`*

CONTINUE AT STEP 10

Agenda

- CAS
  - The code
  - Running locally
  - Running on Render
  - How it works
  - **Assessment**

Assessment

- **Pros**
  - Application need not manage usernames or passwords
  - Application ***cannot*** access passwords!
  - Application is constrained to one user community
  - Princeton OIT will support for the foreseeable future

Assessment

- **Cons**
  - Complex
  - Adds overhead, but only during user’s first visit to the app per browser session
  - Application is constrained to one user community!
  - Princeton OIT is phasing out
  - Deployment requires (minimal) Princeton OIT intervention

Assessment

- For more information…
- https://apereo.github.io/cas/6.5.x/protocol/CAS-Protocol.html