VERA C. RUBIN
OBSERVATORY

Enter the Rubinverse

Unlocking Discovery & Building Community with Citizen Science

Clare Higgs, Sreevani Jarugula, Becky Nevin, Aaron Meisner, Chris Lintott, Brian Nord, Eric Rosas, Aprajita Verma

NSF NOIRLab AURA SLAC U.S. DEPARTMENT OF ENERGY

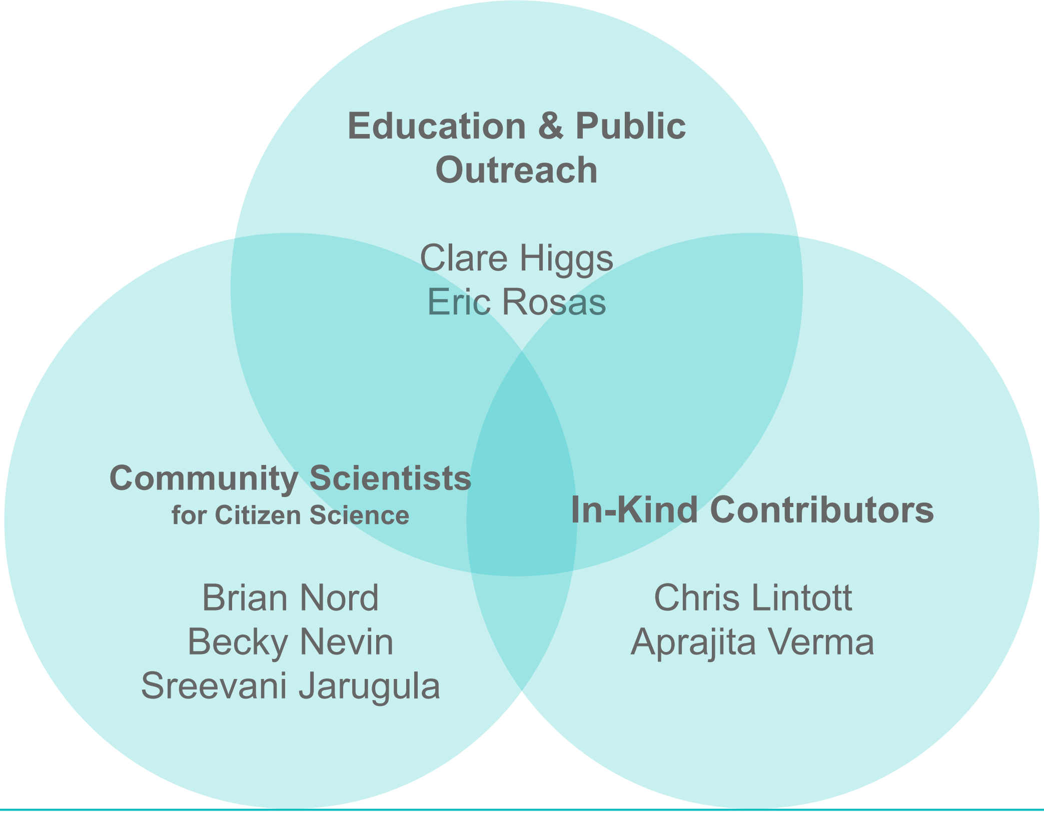
VERA C. RUBIN OBSERVATORY

Introduction

Clare Higgs

NSF NOIRLab AURA SLAC U.S. DEPARTMENT OF ENERGY

Who are we



Education & Public Outreach

Clare Higgs
Eric Rosas

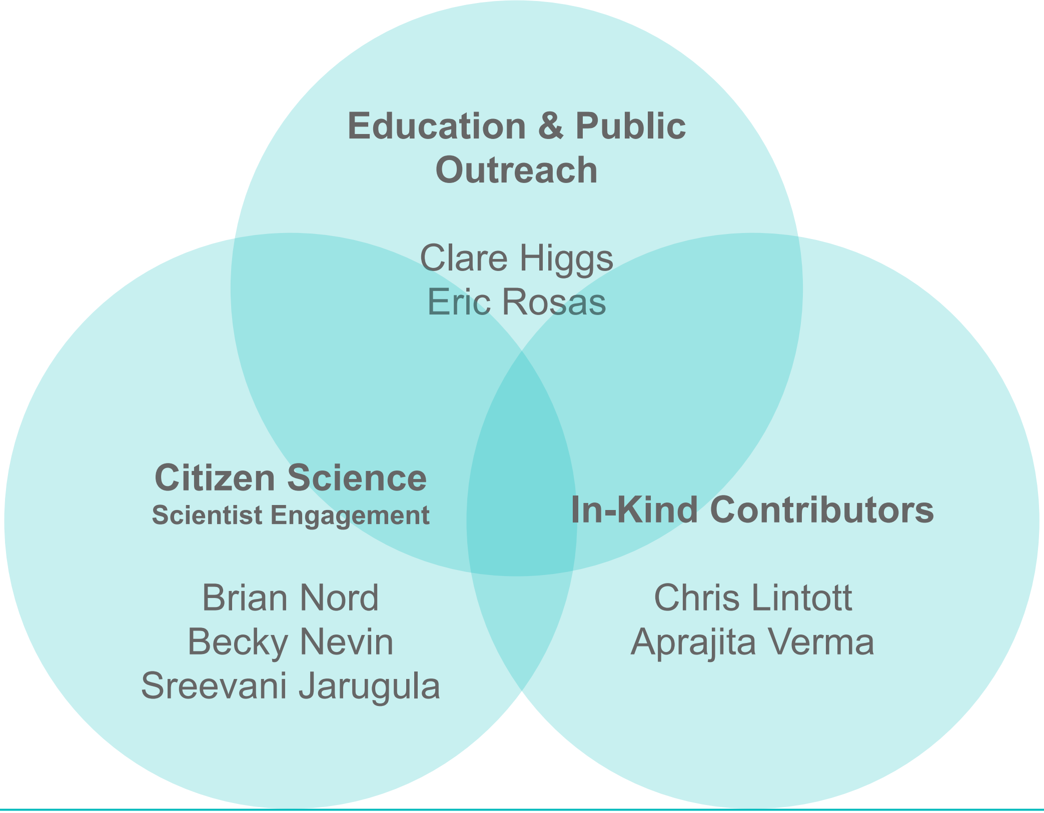
Community Scientists
for Citizen Science

Brian Nord
Becky Nevin
Sreevani Jarugula

In-Kind Contributors

Chris Lintott
Aprajita Verma

Who are we



Education & Public Outreach

Clare Higgs
Eric Rosas

Citizen Science
Scientist Engagement

Brian Nord
Becky Nevin
Sreevani Jarugula

In-Kind Contributors

Chris Lintott
Aprajita Verma

… with support, advice and input from many others in the Rubin community!

What is Citizen Science and why is it important?

“a form of open collaboration where members of the public participate in the scientific process, including identifying research questions, collecting and analyzing the data, interpreting the results, and problem solving.” (Balcom 2015)

- CitSci takes many forms and is used in many fields
- A highly successful and unique form of analysis
    - Galaxy Zoo has lead to 450+ publications

What is Citizen Science and why is it important?

“a form of open collaboration where members of the public participate in the scientific process, including identifying research questions, collecting and analyzing the data, interpreting the results, and problem solving.” (Balcom 2015)

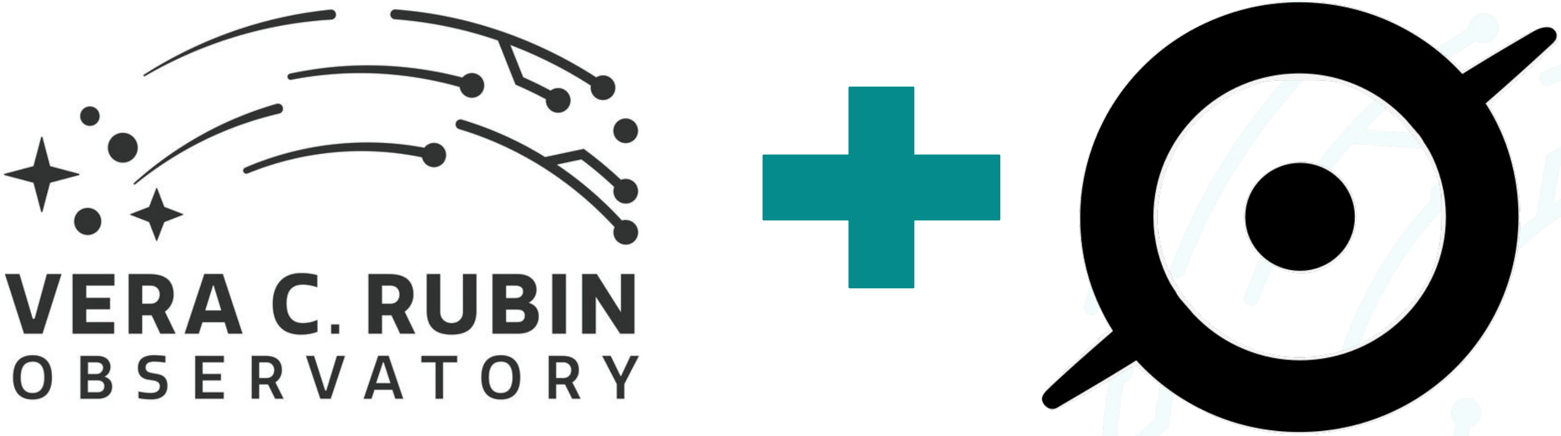
- CitSci takes many forms and is used in many fields
- A highly successful and unique form of analysis
  - Galaxy Zoo has lead to 450+ publications

CitSci is far from perfect…

- Calling it “Citizen Science” is problematic
- Does not necessarily reach a broader group of participants
- Takes effort to be a meaningful to the broader community

Citizen Science with Rubin

- LSST will be a prime source for citizen science projects in many fields
- **Our primary goal:**
  - Facilitate easy and straightforward project development for principal investigators and highlight the amazing science from the Rubin Observatory



Citizen Science with Rubin

We are building tools and a community for the Rubin Observatory

- We are ready to help YOU make projects happen!
    - The CitSci team doesn’t run individual projects.
    - We manage the system of projects and the Rubin infrastructure to enable projects.
- We want this to be the best possible tool for you
    - We welcome your input!
- Anyone can science.
    - For the public as well as our science community

Session Outline

- **Introduction** - *Clare Higgs*
- **Scope of CitSci** - *Aaron Meisner and Chris Lintott*
- **The RubinVerse Overview** - *Clare Higgs*
- **CitSci Notebook Walkthrough and testing** - *Becky Nevin and Sreevani Jarugula*
- **Town Hall Discussion** - *everyone!*
- **Wrap up**

Session Logistics

- **In the room**
  - Clare Higgs
  - Chris Lintott
  - Aaron Meisner
  - Eric Rosas
  - Aprajita Verma
- **Online**
  - Brian Nord
  - Sreevani Jarugula
  - Becky Nevin

- **Feedback/Questions**
  - Use our PCW Slack Channel: #day2-tue-1600-rubinverse-citizen-sci
  - Email: cscience@lsst.org
  - Website: rubinobservatory.org/for-scientists/citizen-science
  - Come by our EPO Open House on Thursday!
  - Ask/comment anonymously: https://ls.st/cscifeedback

We'd love to have your input and thoughts!

VERA C. RUBIN
OBSERVATORY

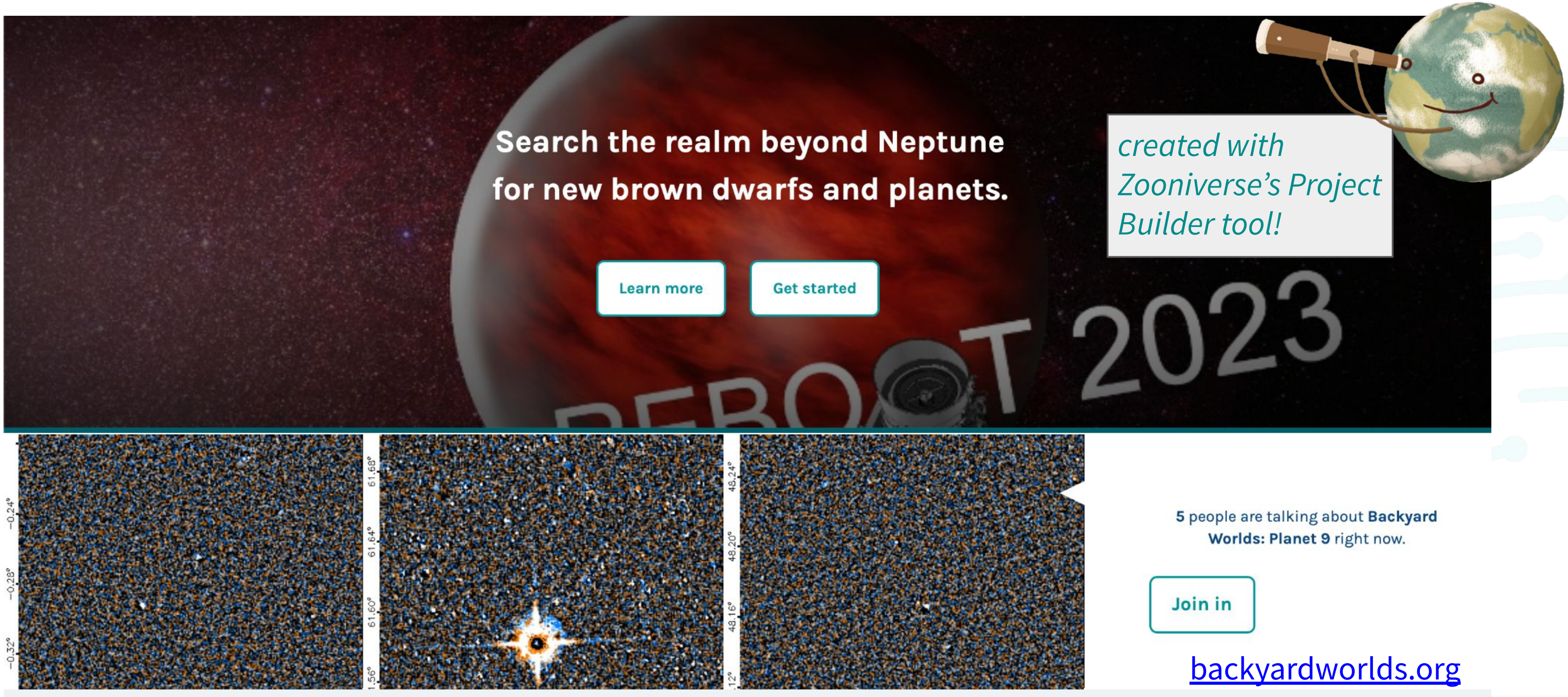
The World of CitSci

Look at all the things CitSci can do!

Aaron Meisner and Chris Lintott

NSF NOIRLab AURA SLAC U.S. DEPARTMENT OF ENERGY

the Backyard Worlds: Planet 9 project



Search the realm beyond Neptune for new brown dwarfs and planets.

5 people are talking about **Backyard Worlds: Planet 9** right now.

created with Zooniverse's Project Builder tool!

backyardworlds.org

Backyard Worlds: CitSci + ‘big data’

- volunteers visually inspect “unWISE” time-series image blinks, looking for brown dwarfs
- 50 trillion input pixels of raw image data
- more than 78,000 registered users, ~200,000 unique volunteer contributors
- participants from all 50 US states plus DC, Puerto Rico and 167 countries

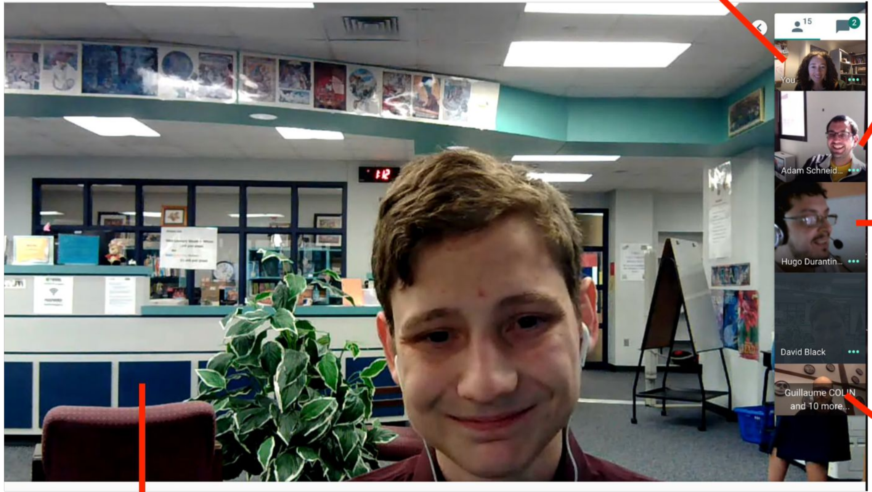
“I do citizen science with BACKYARD WORLDS: PLANET 9 because it’s an amazing opportunity to contribute to an important scientific project while working with great people from all over the world.”
– Michaela Allen

- ~4,000 motion-confirmed LTY dwarf candidates discovered so far

who are the Backyard Worlds citizen scientists?

who are the Backyard Worlds citizen scientists?

community building through citizen science



Jackie Faherty (AMNH)

Adam Schneider
(Arizona State)

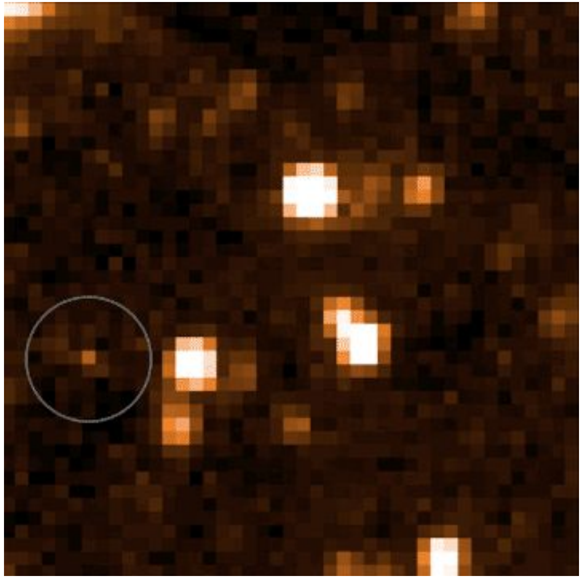
Córdoba,
Argentina

Pau, France

high school lunch period!

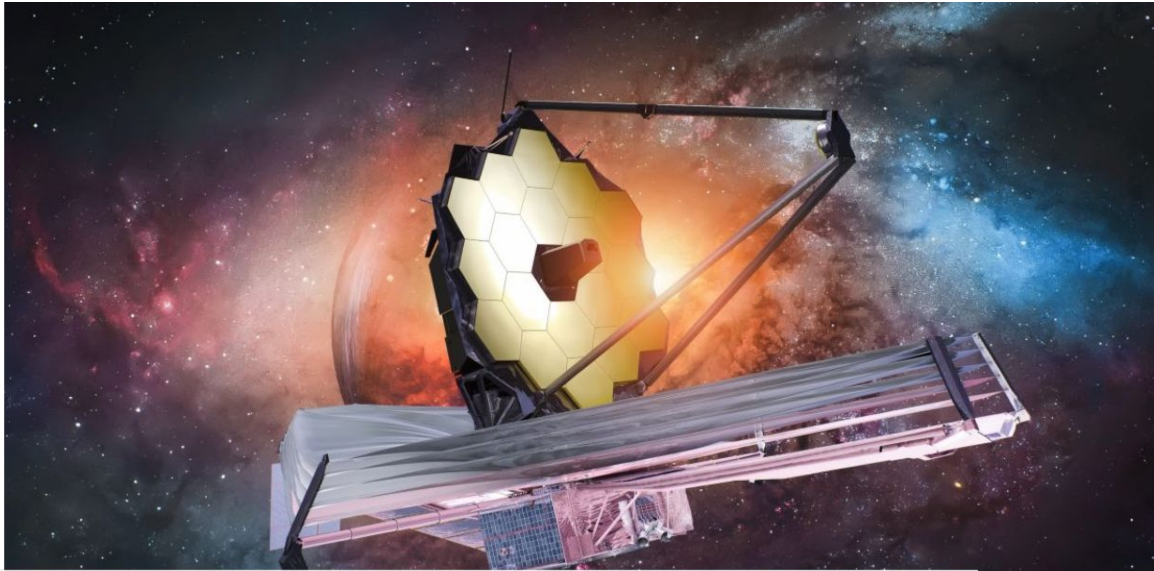
volunteers always invited to join weekly science team hangouts

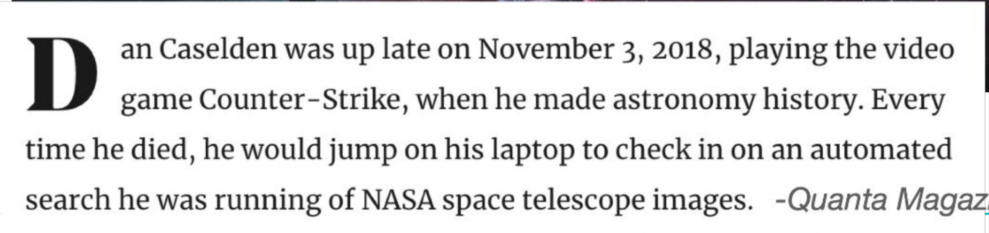
sometimes the best discoveries happen by accident



our fastest-moving discovery so far!

“The Accident”





Dan Caselden was up late on November 3, 2018, playing the video game Counter-Strike, when he made astronomy history. Every time he died, he would jump on his laptop to check in on an automated search he was running of NASA space telescope images.

-Quanta Magazine

VERA C. RUBIN OBSERVATORY

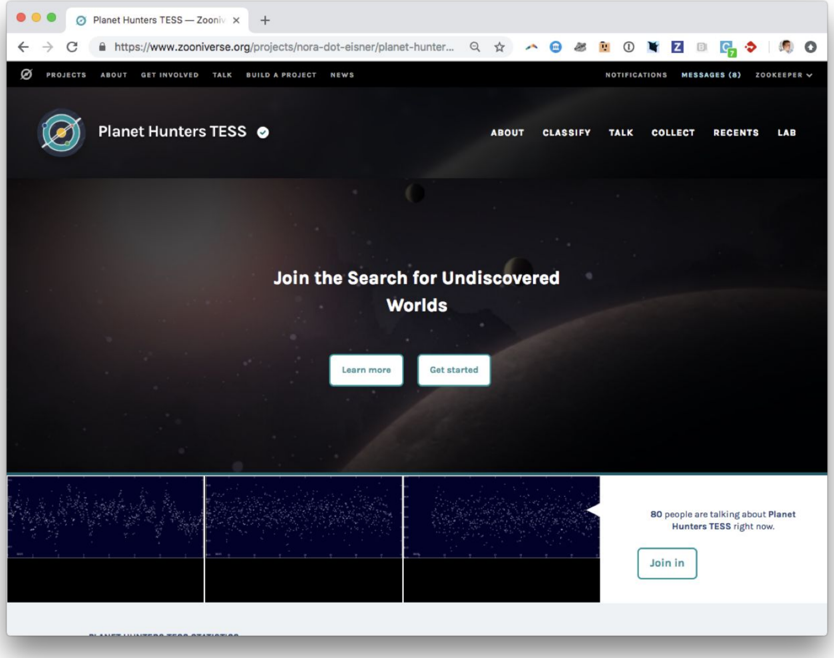
AI+CitSci for the solar neighborhood

- Backyard Worlds: Cool Neighbors — just launched in late June — is a new spinoff project that uses pixel-level deep learning to preselect brown dwarf candidates shown to volunteers

- More than 500,000 classifications in our first month!

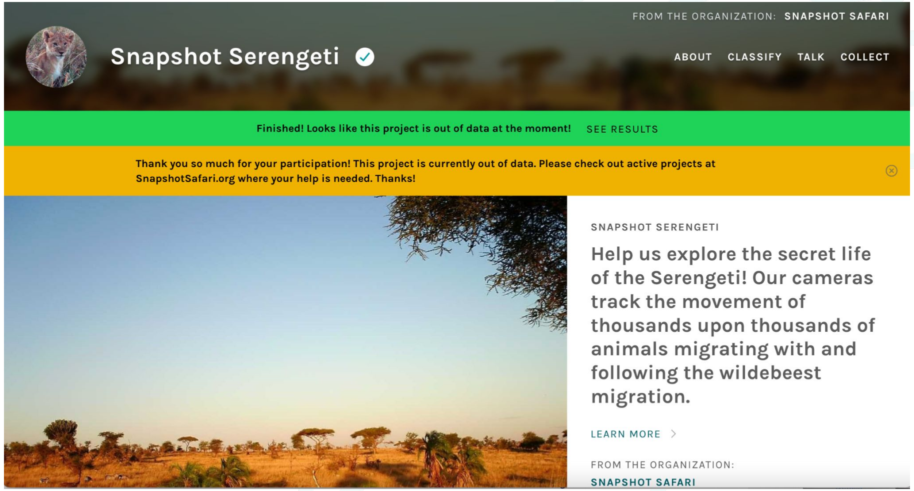
coolneighbors.org

Volunteers are motivated by science



Join the Search for Undiscovered Worlds

80 people are talking about Planet Hunters TESS right now.



Finished! Looks like this project is out of data at the moment!

Thank you so much for your participation! This project is currently out of data. Please check out active projects at SnapshotSafari.org where your help is needed. Thanks!

Help us explore the secret life of the Serengeti! Our cameras track the movement of thousands upon thousands of animals migrating with and following the wildebeest migration.

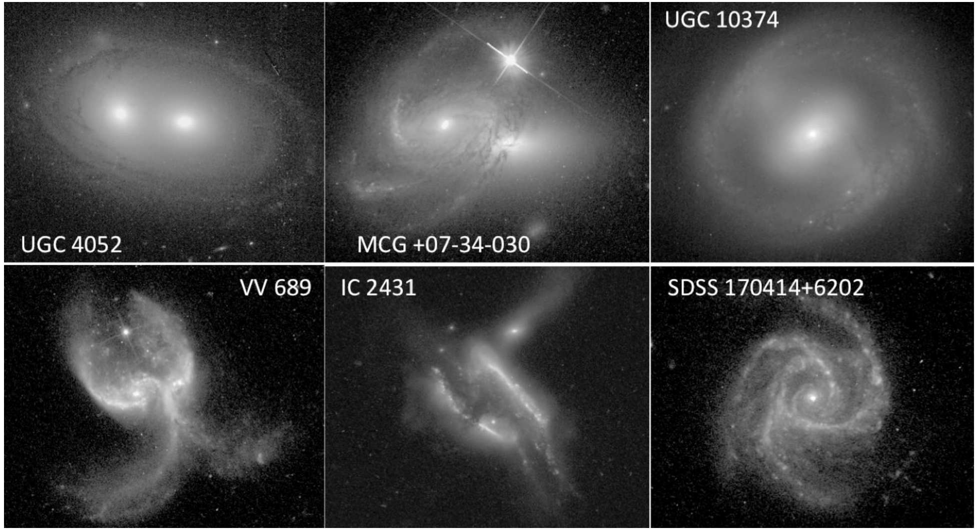
See Raddick et al 0909.2925 & 1303.6886

Transformative casual encounters with science

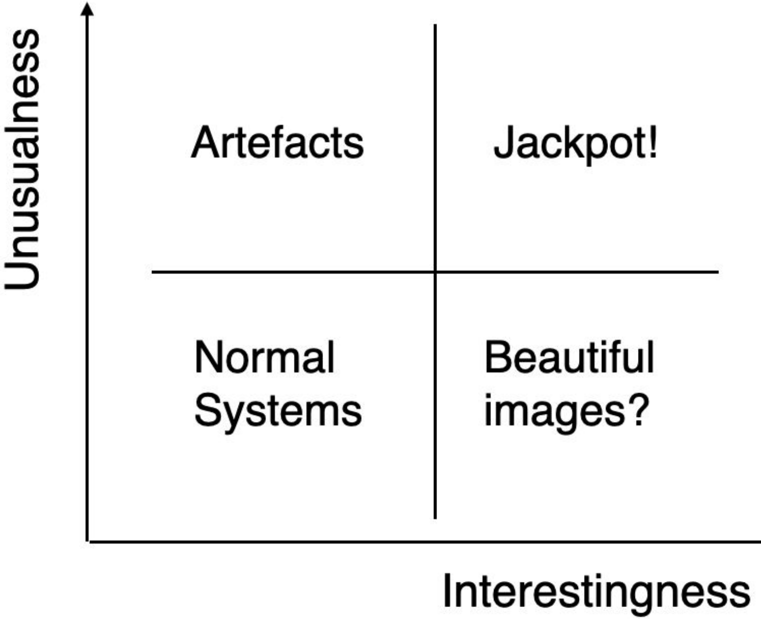
Countering 'Threshold Fear'

(Elaine Heumann Gurian)

Transformative science: Enabling discovery



Keel et al 2202.01098



Unusualness

Artefacts

Jackpot!

Normal Systems

Beautiful images?

Interestingness

See Walmsley et al. arXiv 2110.12735 for Machine Learning input.

VERA C. RUBIN
OBSERVATORY

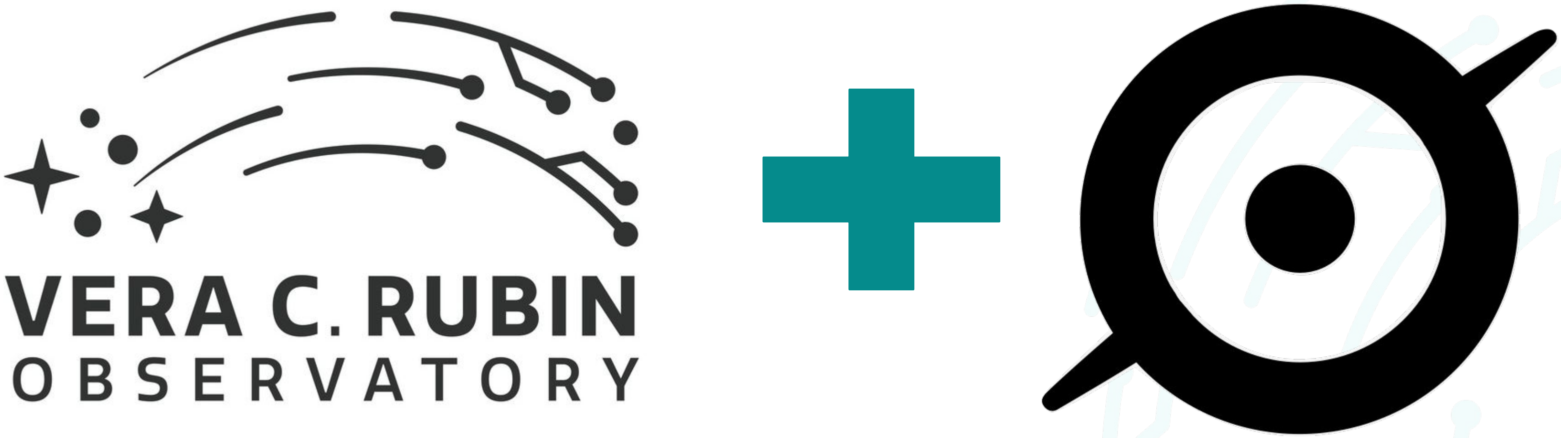
Rubinverse Overview

Clare Higgs

NSF NOIRLab AURA SLAC U.S. DEPARTMENT OF ENERGY

Citizen Science with Rubin

- LSST will be a prime source for citizen science projects in many fields
- **Our primary goal:**
  - Facilitate easy and straightforward project development for principal investigators and highlight the amazing science from the Rubin Observatory



How the Process Works

- **Construction Phase:** “The Sandbox” notebook will exist on the Rubin Science platform
  - Make a test project with <100 objects
  - Design and create your project on Zooniverse with the help of the RubinVerse Citizen Science software products

Rubin Observatory/NSF/AURA

B. Stalder/RubinObs/NSF/AURA

How the Process Works

- **Construction Phase:** “The Sandbox” notebook will exist on the Rubin Science platform
  - Make a test project with <100 objects
  - Design and create your project on Zooniverse with the help of the RubinVerse Citizen Science software products

- **Project Phase:** Check in with the CitSci Team
  - Acquire access to send >100 objects
  - Run your project!

Rubin Observatory/NSF/AURA

B. Stalder/RubinObs/NSF/AURA

How the Process Works

- **Construction Phase:** “The Sandbox” notebook will exist on the Rubin Science platform
  - Make a test project with <100 objects
  - Design and create your project on Zooniverse with the help of the RubinVerse Citizen Science software products
- **Project Phase:** Check in with the CitSci Team
  - Acquire access to send >100 objects
  - Run your project!
- **Science Time:** Do science with classified data!

Rubin Observatory/NSF/AURA

B. Stalder/RubinObs/NSF/AURA

CitSci Software

- Notebooks crafted for data curation that allow you to programmatically send RSP data to your Zooniverse project
- Notebook examples:
  - A “basic” notebook
    - query and send data to the Zooniverse and retrieve classified data
  - More complex example using a “flipbook”
    - E.g., for time series data, for multiple views
    - Demonstrates how to use some of Zooniverse’s capabilities
  - An “alert broker” based example
    - Ingests and returns data to an alert broker, rather than the data release

We will be demonstrating the basic version today and its available for you to try!

Citizen Science with Rubin

The CitSci team provides the following:

- Software pipeline to connect Rubin Science Platform to Zooniverse
- Consultation on project management
- Custom Zooniverse project template
- Rubin branding
- Social media engagement and promotion

Elements for a successful project:

- Engaged PIs keen to share Rubin with the public
- Coordination and collaboration
- Equity and accessibility initiatives

/VRubinObs

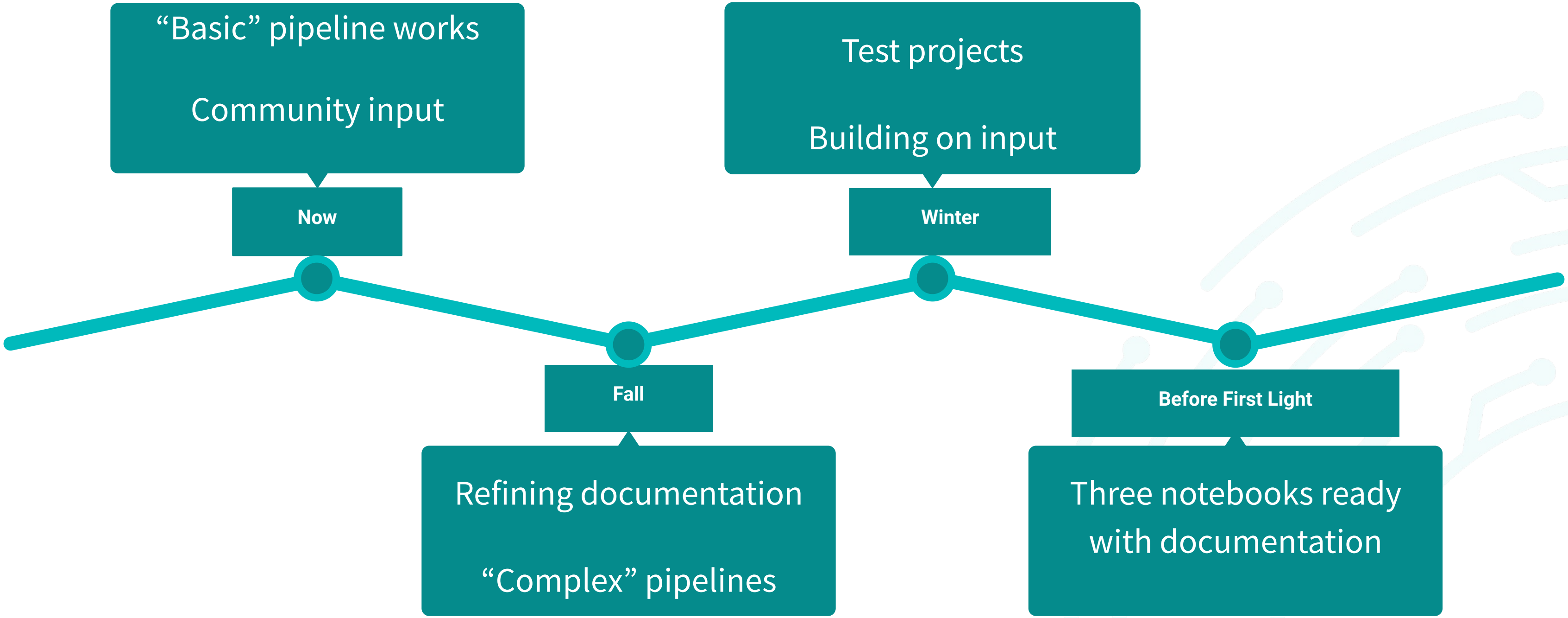
/rubin_observatory

/company/rubinobservatory

/VRubinObs

/RubinObservatory

Timeline for the CitSci Pipeline



Now

“Basic” pipeline works

Community input

Fall

Refining documentation

“Complex” pipelines

Winter

Test projects

Building on input

Before First Light

Three notebooks ready with documentation

Data Rights Considerations

Rubin Data Rights:

- Data shared on Zooniverse is world public
- Sharing only derived data products is encouraged
- Citsci team is providing documentation and guidance to navigate

Citsci projects are an important, unique and valuable part of Rubin science.

Outstanding Questions under consideration

- What would make you most likely to take on a project?
  - How can we make this the best tool for you?
- How do you feel about projects that overlap in subject area?
  - Duplicated projects devalue the public’s time
- What would be your weekly time commitment for a project like this?
  - How much effort would you be able to commit to your community of citizen scientists?
- When is a project finished?
  - Data sharing and archiving to best respect the effort of the public

We look forward to your thoughts and discussion later this session!

VERA C. RUBIN
OBSERVATORY

Notebook Walkthrough

This is an interactive walkthrough!
Please ask general questions via a hand raise, put technical issues in the slack.
#day2-tue-1600-rubinverse-citizen-sci

Sreevani Jarugula and Becky Nevin

NSF NOIRLab AURA SLAC U.S. DEPARTMENT OF ENERGY

The Rubinverse unites three key components

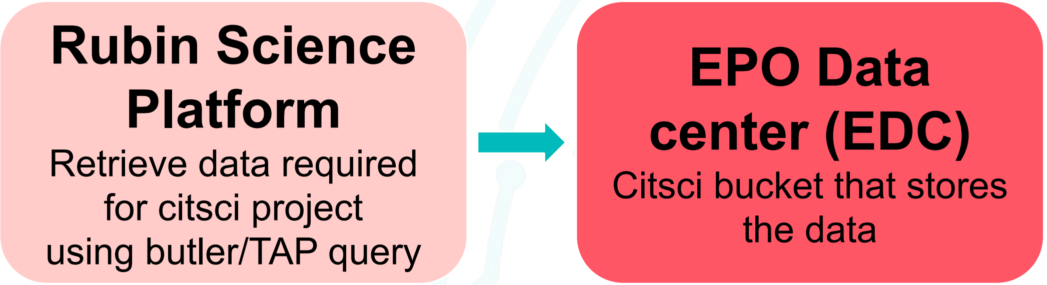
- Rubin Science Platform
  - ○ Raw survey data available for Principal Investigators (PIs)
  - ○ Tutorial Notebooks

Rubin Science Platform

Retrieve data required for citsci project using butler/TAP query

The Rubinverse unites three key components

- Rubin Science Platform
  - Raw survey data available for Principal Investigators (PIs)
  - Tutorial Notebooks
- EPO data centre (EDC)
  - Host the curated data for citizen science project on EDC



Rubin Science Platform

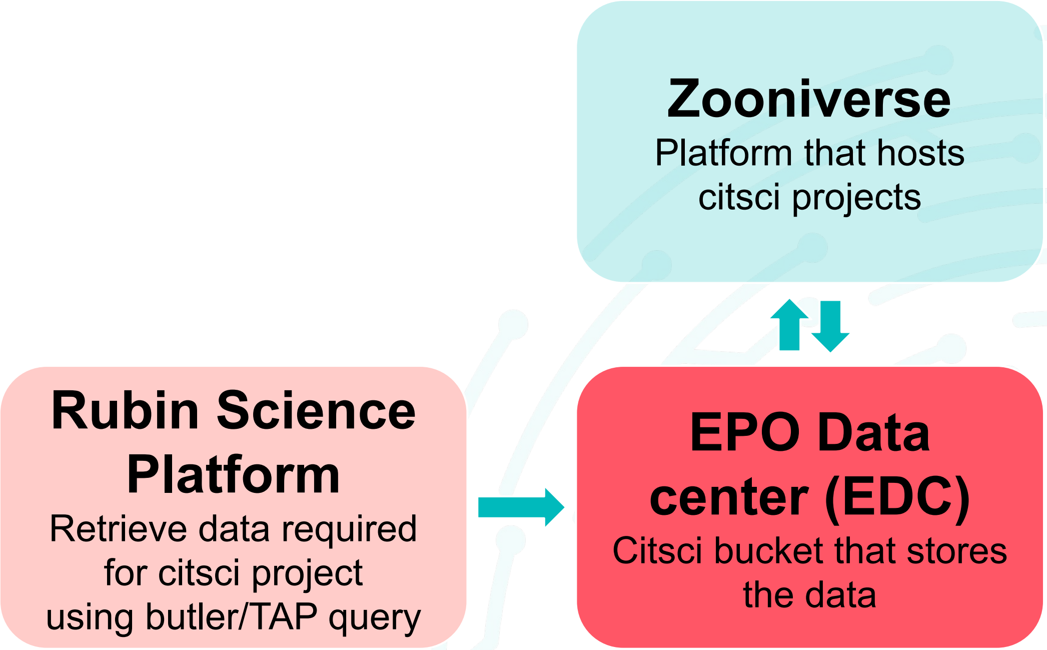
Retrieve data required for citsci project using butler/TAP query

EPO Data center (EDC)

Citsci bucket that stores the data

The Rubinverse unites three key components

- Rubin Science Platform
  - Raw survey data available for Principal Investigators (PIs)
  - Tutorial Notebooks
- EPO data centre (EDC)
  - Host the curated data for citizen science project on EDC
- Zooniverse
  - Host the citsci project
  - Link between the project and data in EDC



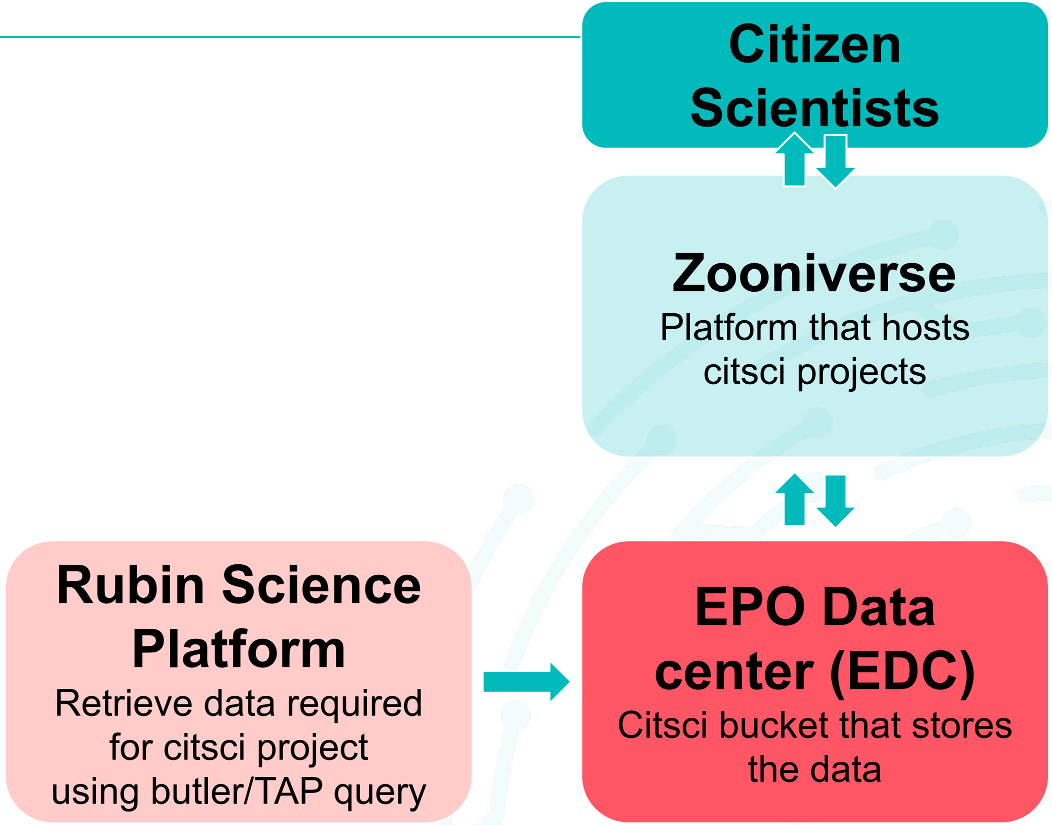
Zooniverse
Platform that hosts citsci projects

Rubin Science Platform
Retrieve data required for citsci project using butler/TAP query

EPO Data center (EDC)
Citsci bucket that stores the data

The Rubinverse unites three key components

- Rubin Science Platform
  - Raw survey data available for Principal Investigators (PIs)
  - Tutorial Notebooks
- EPO data centre (EDC)
  - Host the curated data for citizen science project on EDC
- Zooniverse
  - Host the citsci project
  - Link between the project and data in EDC



Citizen Scientists

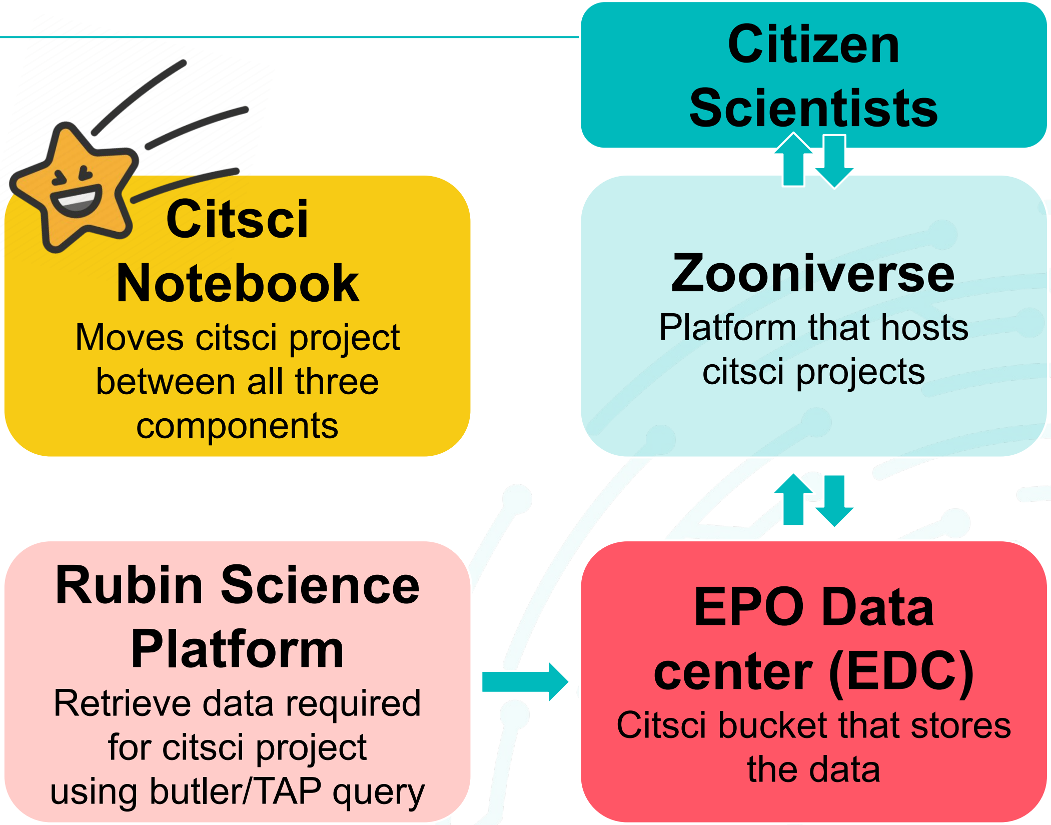
Zooniverse
Platform that hosts citsci projects

Rubin Science Platform
Retrieve data required for citsci project using butler/TAP query

EPO Data center (EDC)
Citsci bucket that stores the data

The Rubinverse unites three key components

- Rubin Science Platform
  - Raw survey data available for Principal Investigators (PIs)
  - Tutorial Notebooks
- EPO data centre (EDC)
  - Host the curated data for citizen science project on EDC
- Zooniverse
  - Host the citsci project
  - Link between the project and data in EDC



Citizen Scientists

Citsci Notebook
Moves citsci project between all three components

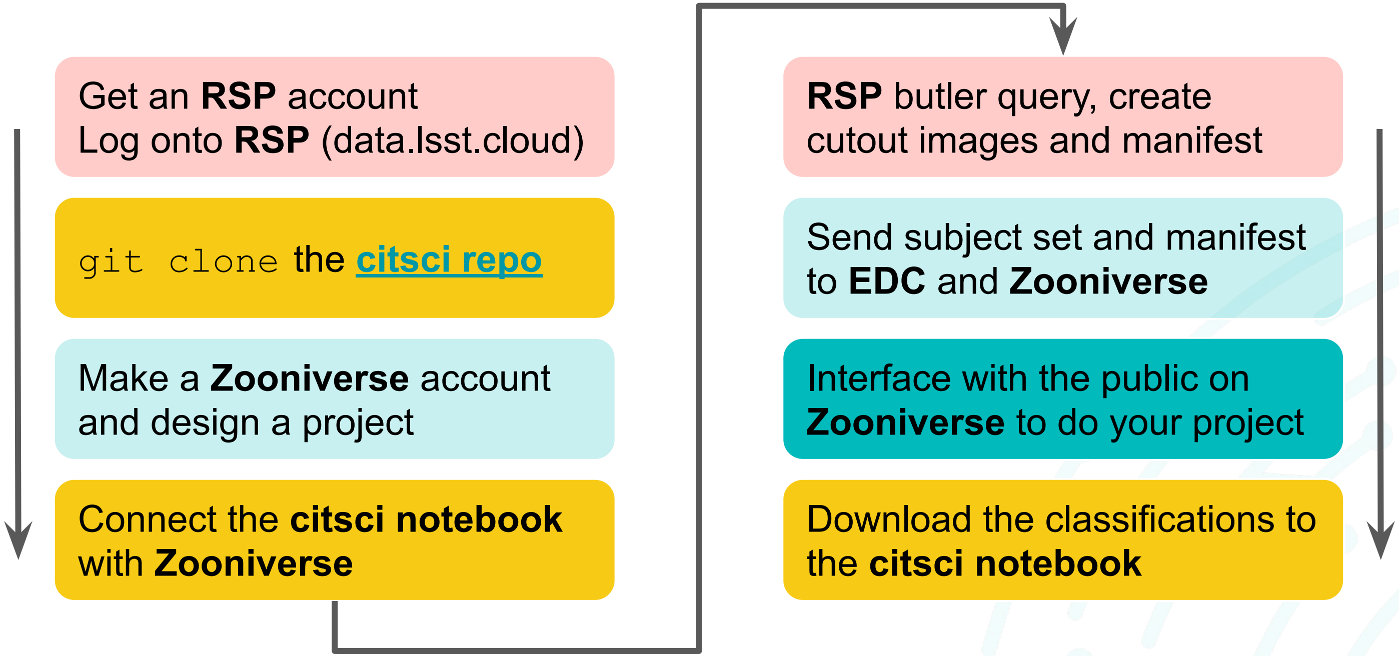
Zooniverse
Platform that hosts citsci projects

Rubin Science Platform
Retrieve data required for citsci project using butler/TAP query

EPO Data center (EDC)
Citsci bucket that stores the data

RubinVerse workflow

The details of the workflow will be described as we run the notebook



Get an **RSP** account
Log onto **RSP** (data.lsst.cloud)

`git clone` the citsci repo

Make a **Zooniverse** account and design a project

Connect the **citsci notebook** with **Zooniverse**

RSP butler query, create cutout images and manifest

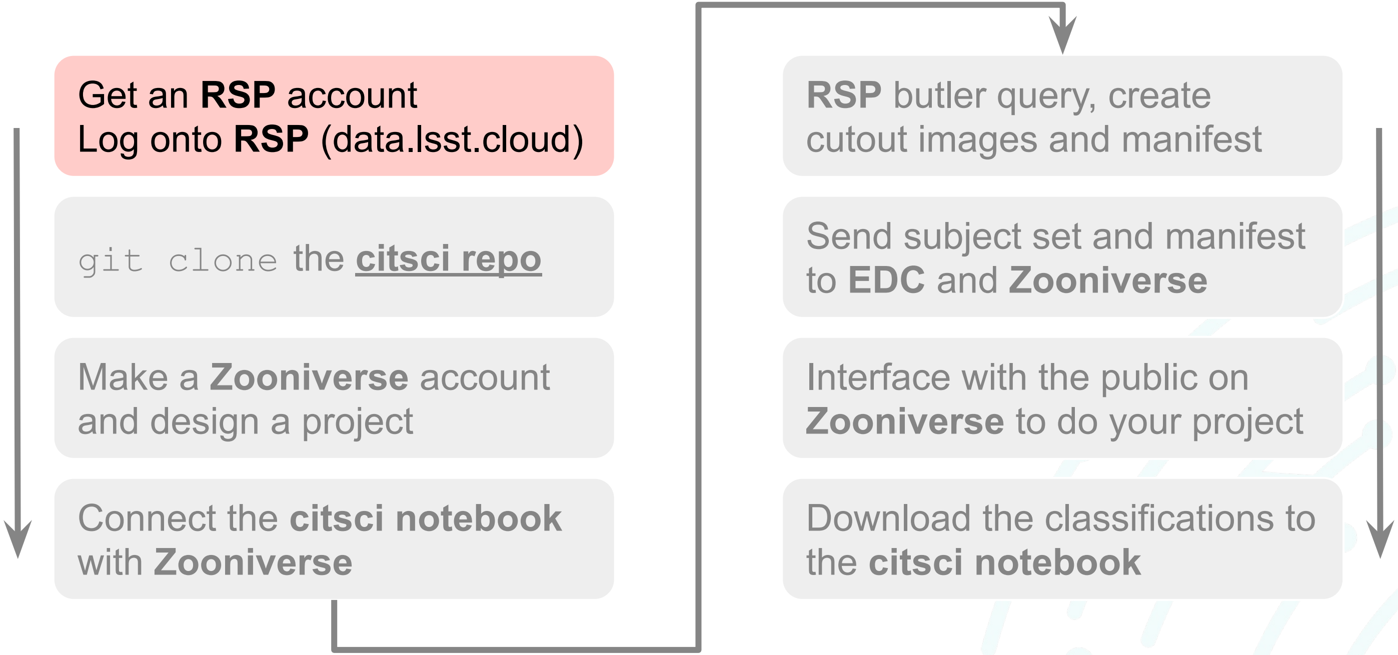
Send subject set and manifest to **EDC** and **Zooniverse**

Interface with the public on **Zooniverse** to do your project

Download the classifications to the **citsci notebook**

And repeat!

RubinVerse workflow



Get an **RSP** account
Log onto **RSP** (data.lsst.cloud)

`git clone` the **citsci repo**

Make a **Zooniverse** account and design a project

Connect the **citsci notebook** with **Zooniverse**

RSP butler query, create cutout images and manifest

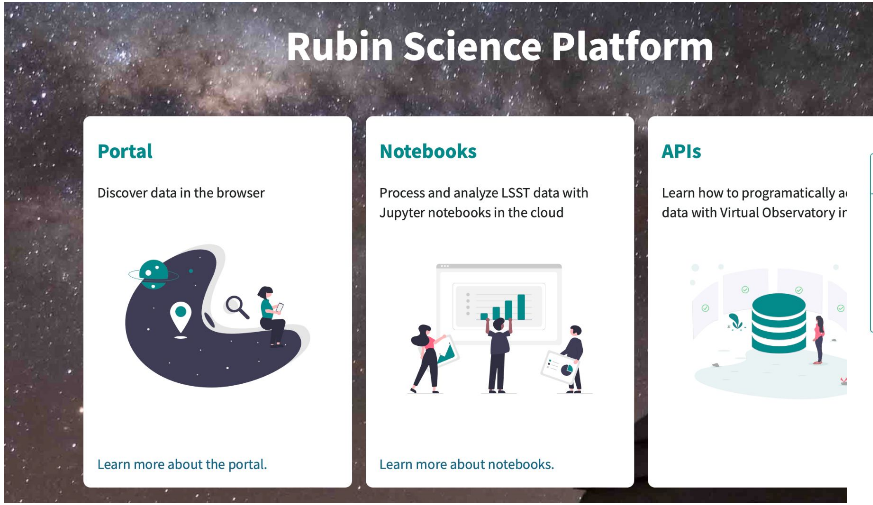
Send subject set and manifest to **EDC** and **Zooniverse**

Interface with the public on **Zooniverse** to do your project

Download the classifications to the **citsci notebook**

Connect to the Rubin Science Platform (RSP)

The tutorial notebook is hosted on RSP (go to data.lsst.cloud)



Rubin Science Platform

Portal

Discover data in the browser

Learn more about the portal.

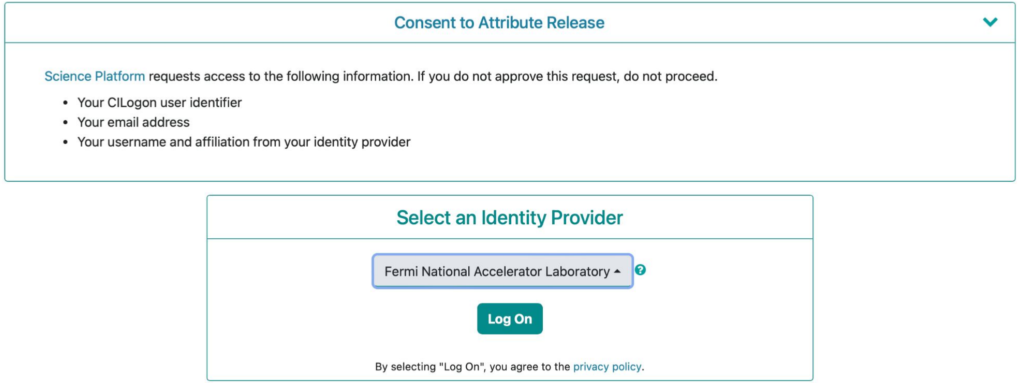
Notebooks

Process and analyze LSST data with Jupyter notebooks in the cloud

Learn more about notebooks.

APIs

Learn how to programatically a
data with Virtual Observatory ir



Consent to Attribute Release

Science Platform requests access to the following information. If you do not approve this request, do not proceed.

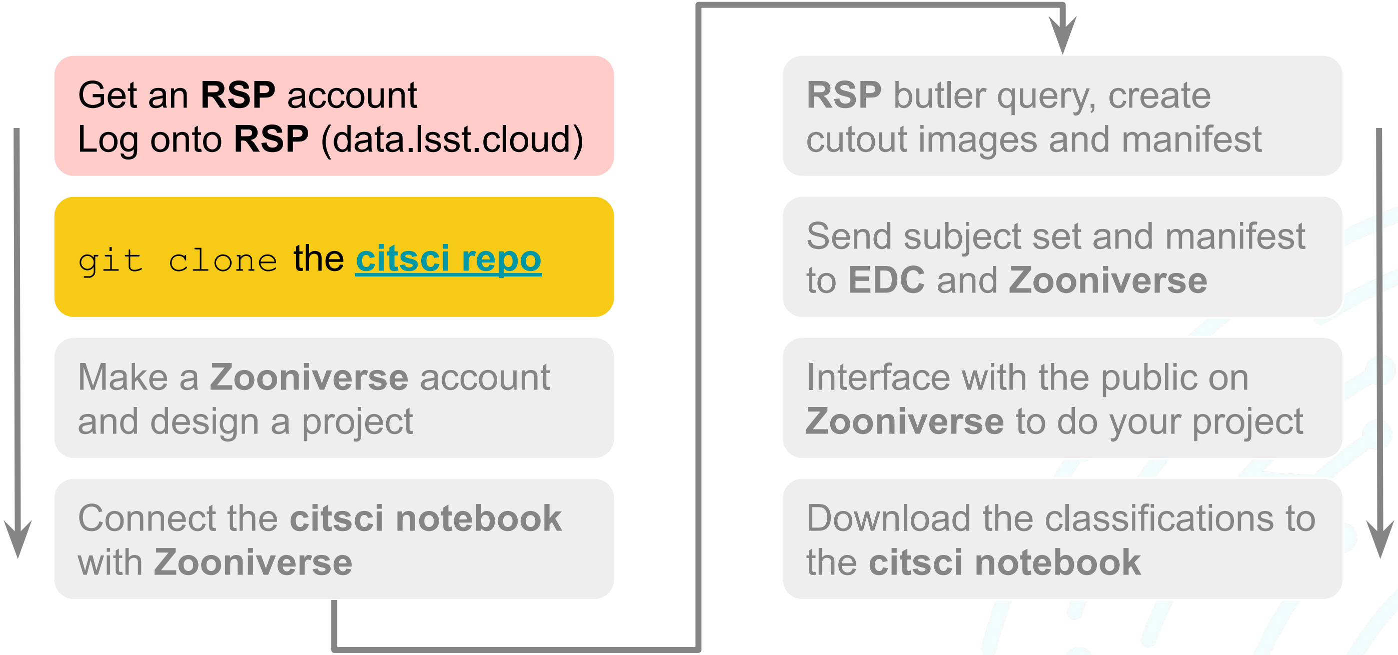
- Your CILogon user identifier
- Your email address
- Your username and affiliation from your identity provider

Select an Identity Provider

Fermi National Accelerator Laboratory

By selecting "Log On", you agree to the privacy policy.

RubinVerse workflow



Get an **RSP** account
Log onto **RSP** (data.lsst.cloud)

`git clone` the citsci repo

Make a **Zooniverse** account and design a project

Connect the **citsci notebook** with **Zooniverse**

RSP butler query, create cutout images and manifest

Send subject set and manifest to **EDC** and **Zooniverse**

Interface with the public on **Zooniverse** to do your project

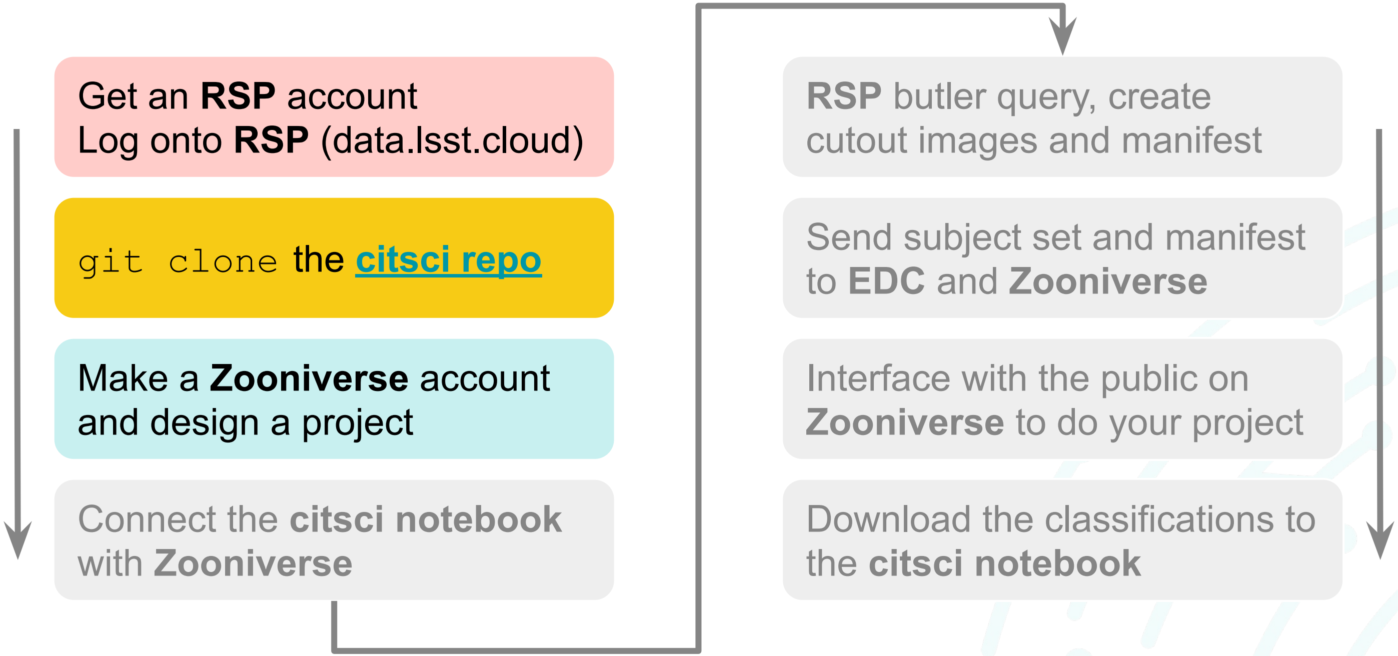
Download the classifications to the **citsci notebook**

Git pull slide

Go to /home/username

```
git clone https://github.com/lsst-epo/citizen-science-notebooks.git
```

RubinVerse workflow



Get an **RSP** account
Log onto **RSP** (data.lsst.cloud)

`git clone` the citsci repo

Make a **Zooniverse** account and design a project

Connect the **citsci notebook** with **Zooniverse**

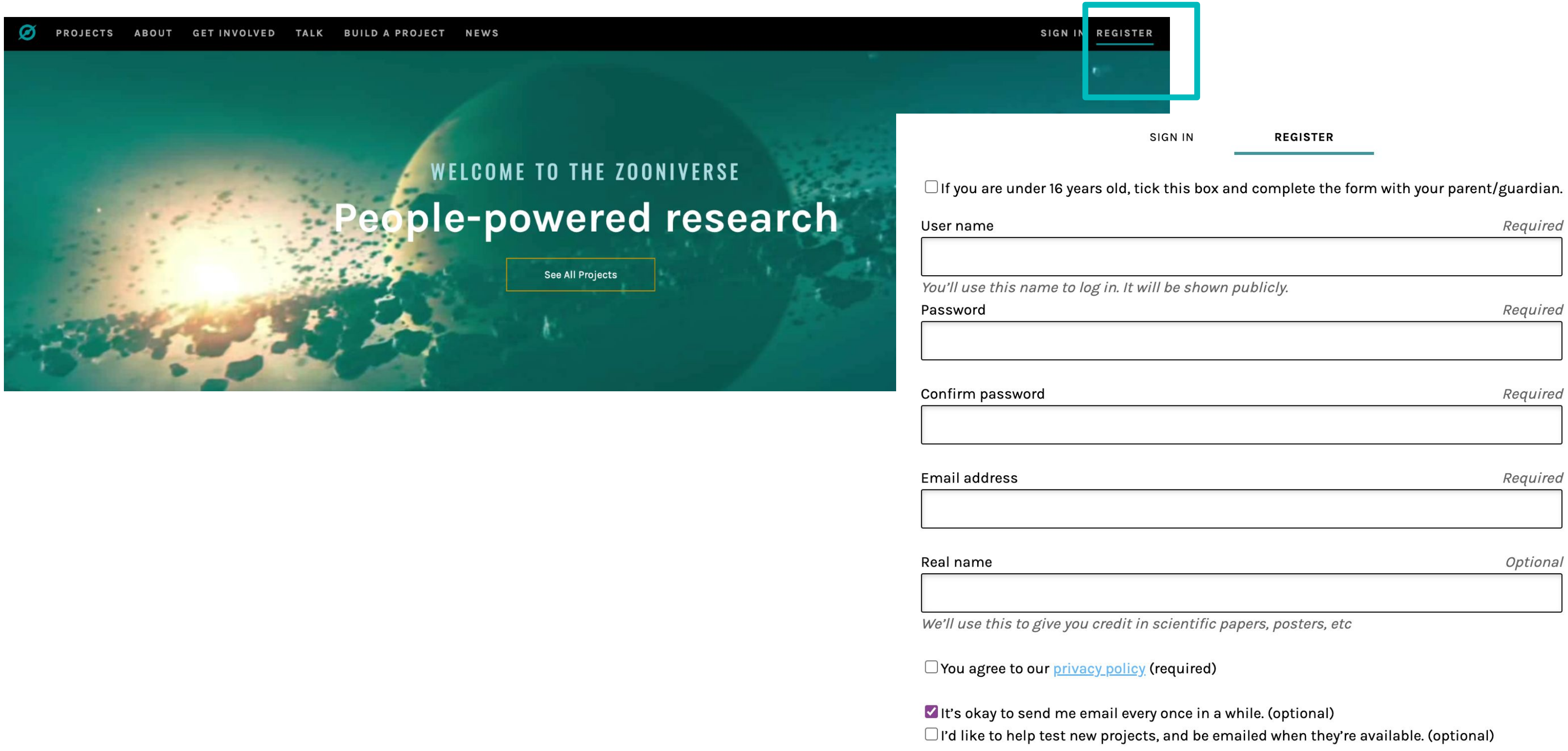
RSP butler query, create cutout images and manifest

Send subject set and manifest to **EDC** and **Zooniverse**

Interface with the public on **Zooniverse** to do your project

Download the classifications to the **citsci notebook**

Create the Zooniverse account



SIGN IN REGISTER

□ If you are under 16 years old, tick this box and complete the form with your parent/guardian.

User name *Required*

You'll use this name to log in. It will be shown publicly.

Password *Required*

Confirm password *Required*

Email address *Required*

Real name *Optional*

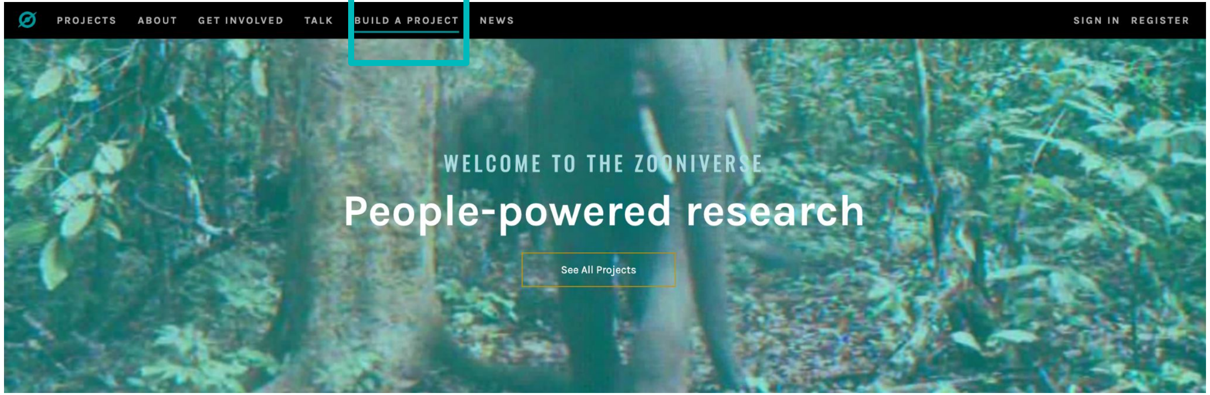
We'll use this to give you credit in scientific papers, posters, etc

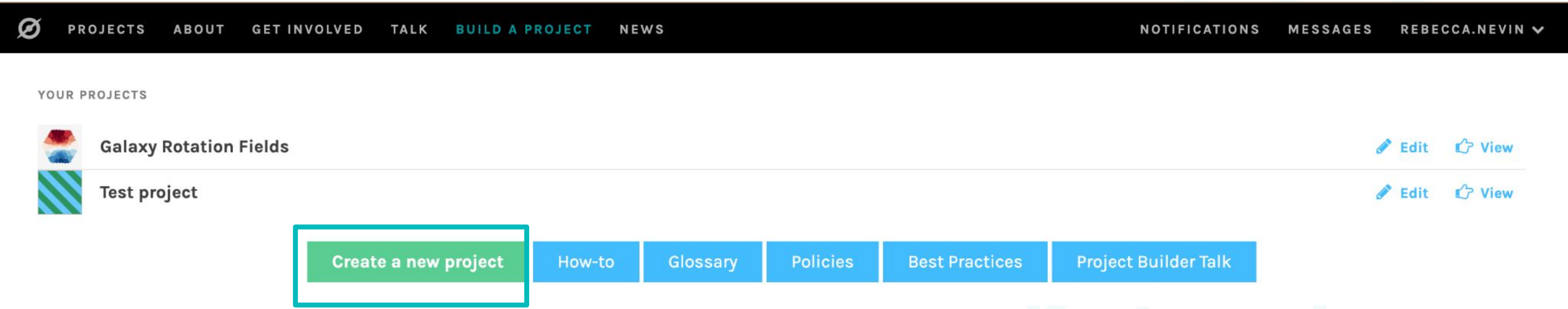
□ You agree to our privacy policy (required)

☑ It's okay to send me email every once in a while. (optional)

□ I'd like to help test new projects, and be emailed when they're available. (optional)

Create a Zooniverse project

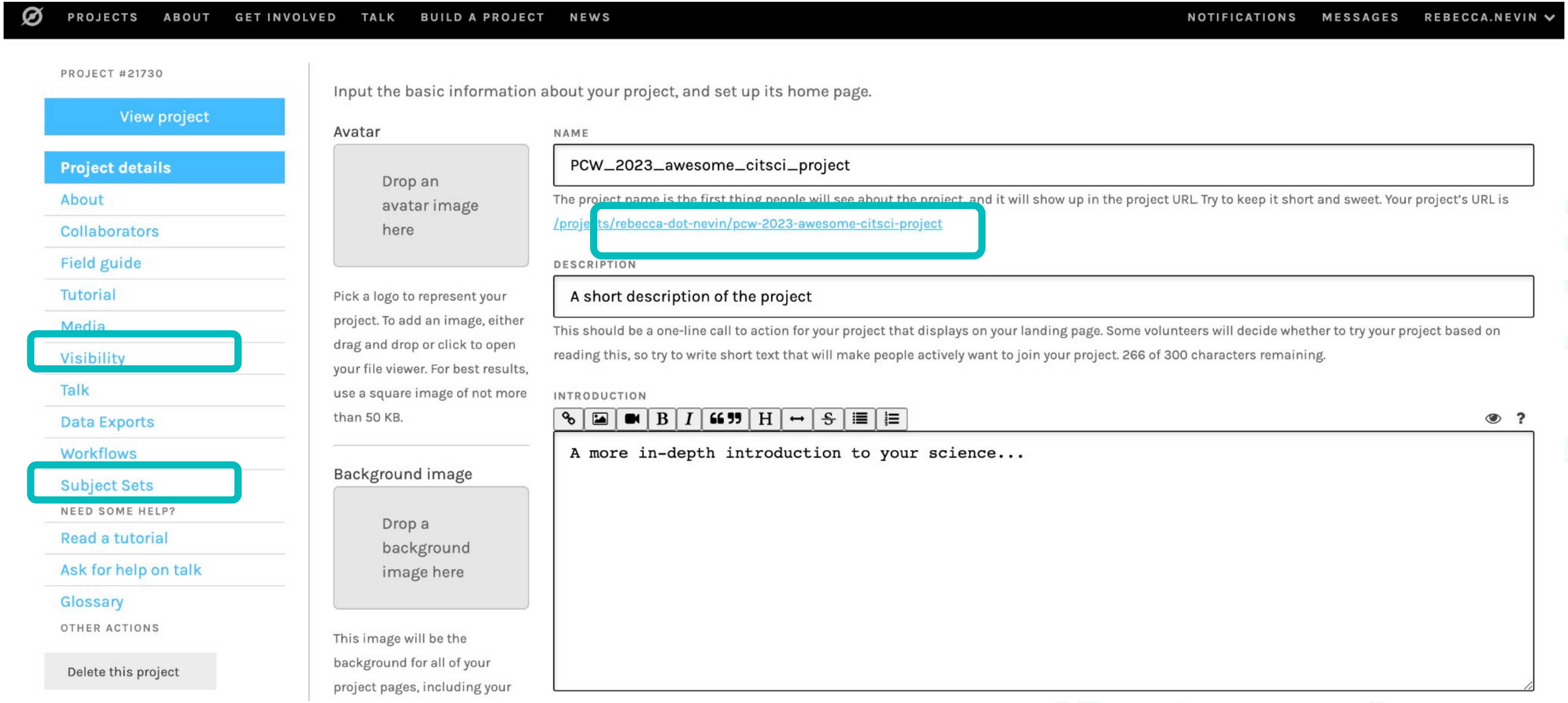




YOUR PROJECTS

Galaxy Rotation Fields	Edit	View
Test project	Edit	View

Create a Zooniverse project



PROJECT #21730

View project

Project details
About
Collaborators
Field guide
Tutorial
Media
Visibility
Talk
Data Exports
Workflows
Subject Sets

NEED SOME HELP?

Read a tutorial
Ask for help on talk
Glossary

OTHER ACTIONS

Delete this project

Input the basic information about your project, and set up its home page.

Avatar

Drop an avatar image here

Pick a logo to represent your project. To add an image, either drag and drop or click to open your file viewer. For best results, use a square image of not more than 50 KB.

Background image

Drop a background image here

This image will be the background for all of your project pages, including your

NAME

PCW_2023_awesome_citsci_project

The project name is the first thing people will see about the project, and it will show up in the project URL. Try to keep it short and sweet. Your project's URL is /projects/rebecca-dot-nevin/pcw-2023-awesome-citsci-project

DESCRIPTION

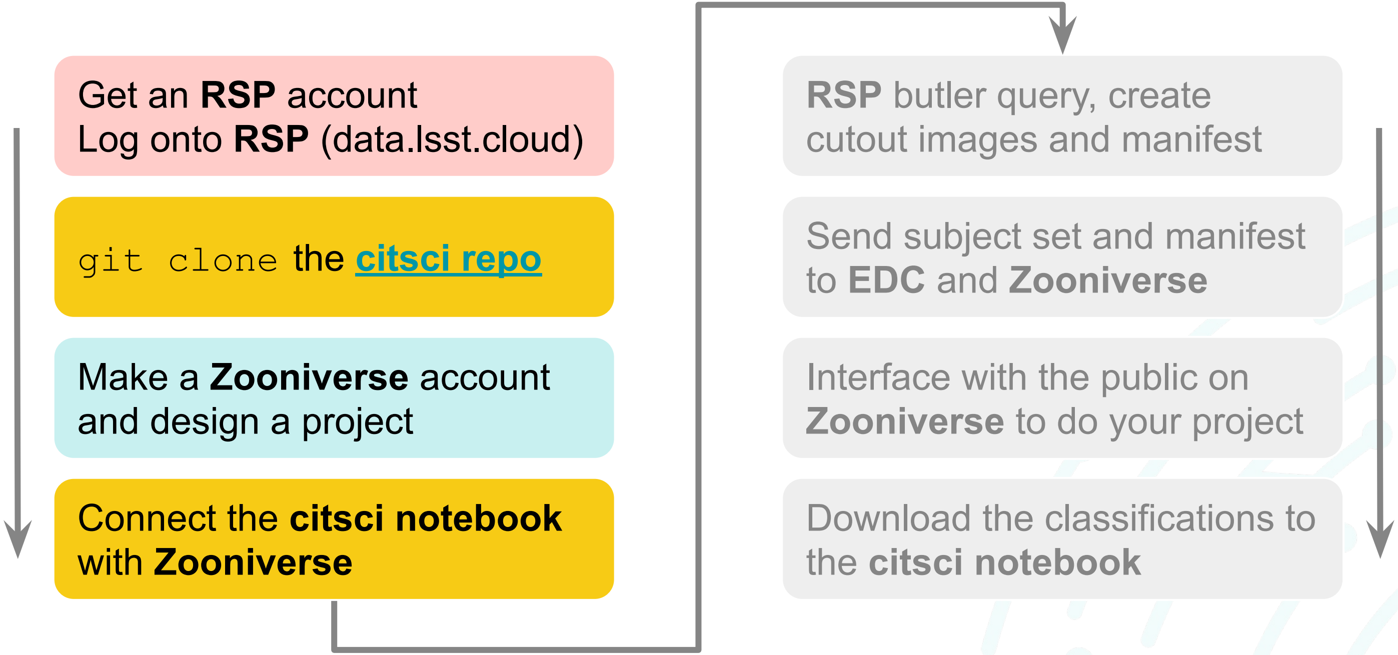
A short description of the project

This should be a one-line call to action for your project that displays on your landing page. Some volunteers will decide whether to try your project based on reading this, so try to write short text that will make people actively want to join your project. 266 of 300 characters remaining.

INTRODUCTION

A more in-depth introduction to your science...

RubinVerse workflow



Get an **RSP** account
Log onto **RSP** (data.lsst.cloud)

`git clone` the citsci repo

Make a **Zooniverse** account and design a project

Connect the **citsci notebook** with **Zooniverse**

RSP butler query, create cutout images and manifest

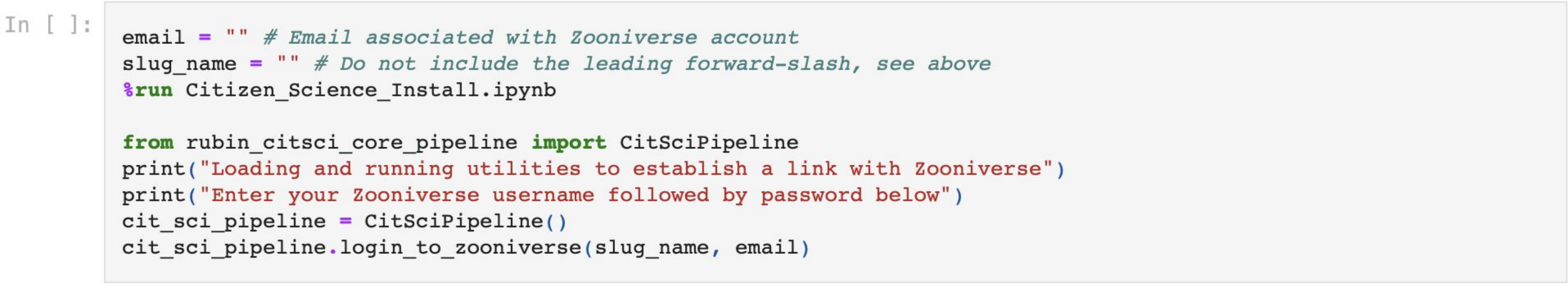
Send subject set and manifest to **EDC** and **Zooniverse**

Interface with the public on **Zooniverse** to do your project

Download the classifications to the **citsci notebook**

Connect to Zooniverse

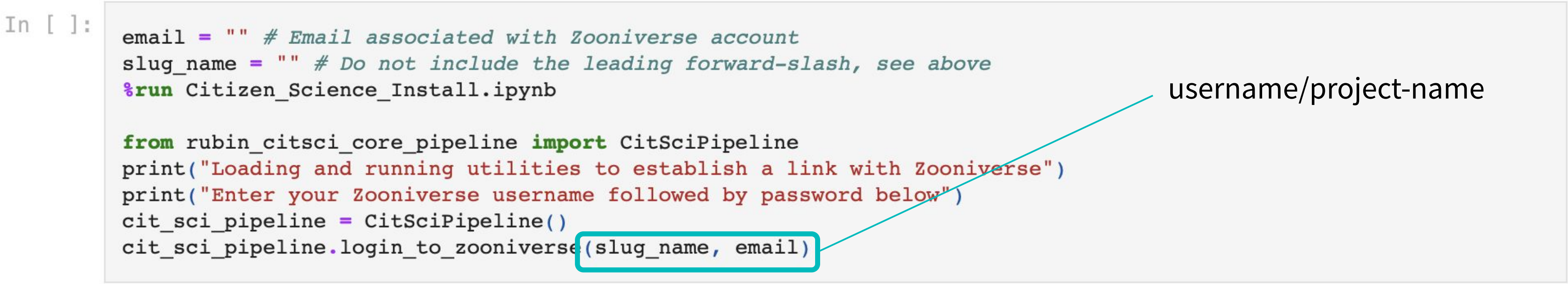
- Install packages
  - Panoptes client: software development kit (SDK) for the platform behind Zooniverse
  - Google cloud storage: store data in EDC


In [ ]:

```
email = "" # Email associated with Zooniverse account
slug_name = "" # Do not include the leading forward-slash, see above
%run Citizen_Science_Install.ipynb

from rubin_citsci_core_pipeline import CitSciPipeline
print("Loading and running utilities to establish a link with Zooniverse")
print("Enter your Zooniverse username followed by password below")
cit_sci_pipeline = CitSciPipeline()
cit_sci_pipeline.login_to_zooniverse(slug_name, email)
```

Connect to Zooniverse

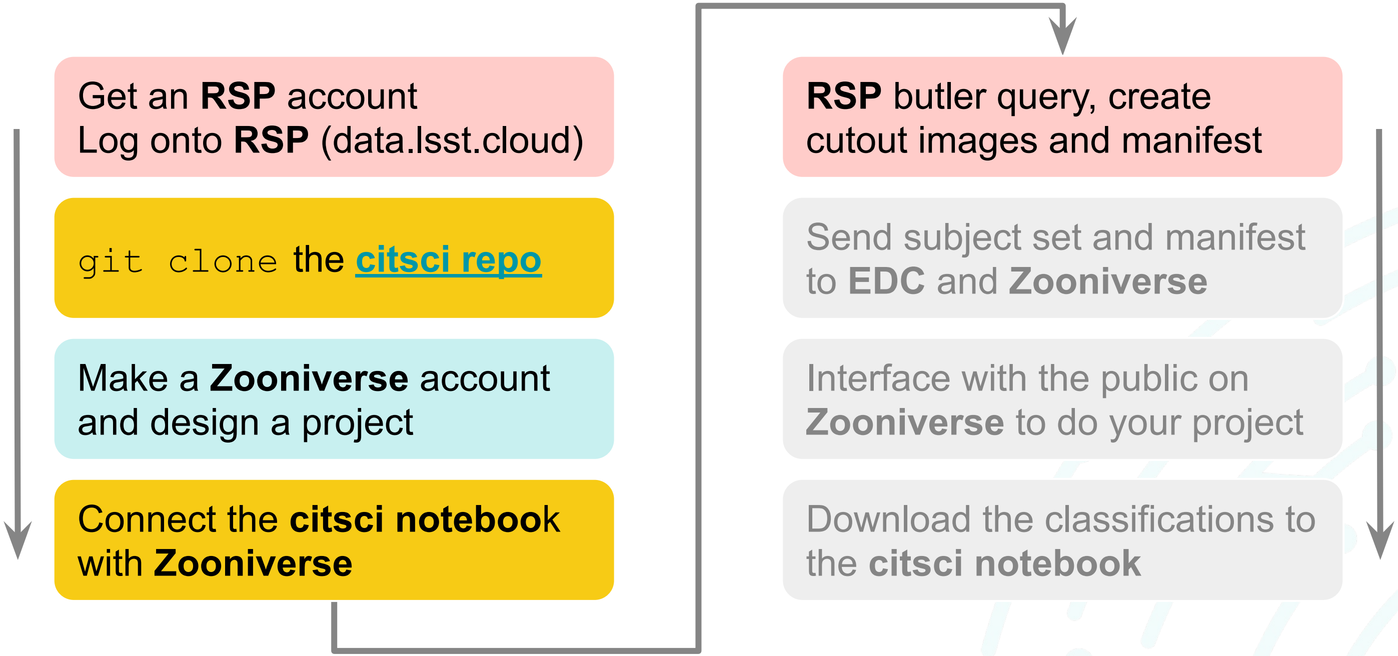
- Install packages
  - Panoptes client: software development kit (SDK) for the platform behind Zooniverse
  - Google cloud storage: store data in EDC
- Logging into Zooniverse
  - Zooniverse API authenticates the RSP notebook user to log into the platform



```
In [ ]:
email = "" # Email associated with Zooniverse account
slug_name = "" # Do not include the leading forward-slash, see above
%run Citizen_Science_Install.ipynb

from rubin_citsci_core_pipeline import CitSciPipeline
print("Loading and running utilities to establish a link with Zooniverse")
print("Enter your Zooniverse username followed by password below")
cit_sci_pipeline = CitSciPipeline()
cit_sci_pipeline.login_to_zooniverse(slug_name, email)
```

username/project-name

RubinVerse workflow



Get an **RSP** account
Log onto **RSP** (data.lsst.cloud)

`git clone` the [citsci repo]

Make a **Zooniverse** account and design a project

Connect the **citsci notebook** with **Zooniverse**

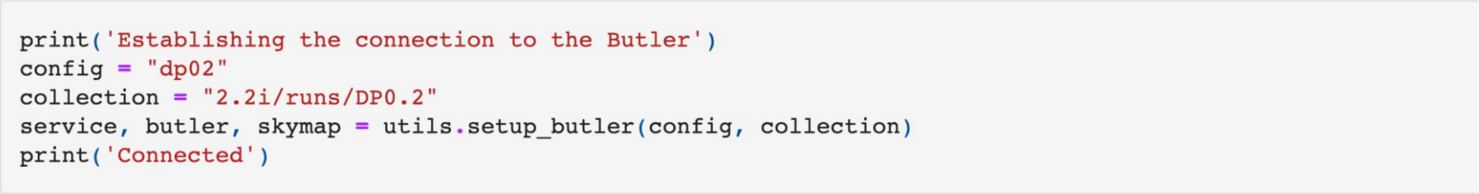
RSP butler query, create cutout images and manifest

Send subject set and manifest to **EDC** and **Zooniverse**

Interface with the public on **Zooniverse** to do your project

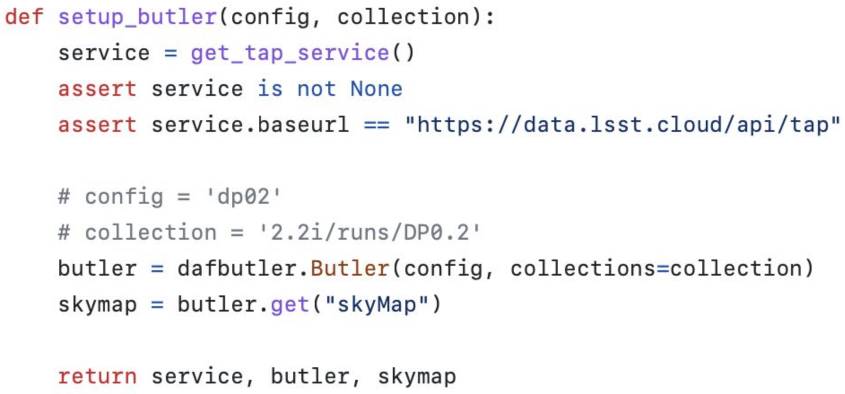
Download the classifications to the **citsci notebook**

Set up TAP and Butler for the data query

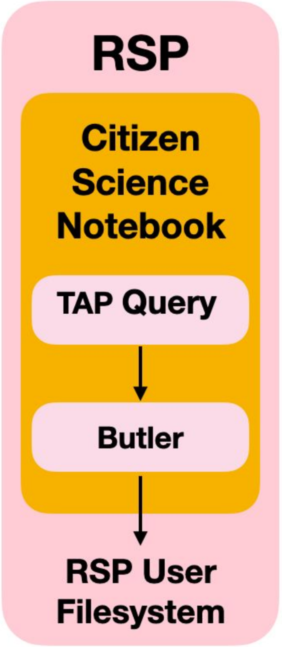


```
print('Establishing the connection to the Butler')
config = "dp02"
collection = "2.2i/runs/DP0.2"
service, butler, skymap = utils.setup_butler(config, collection)
print('Connected')
```

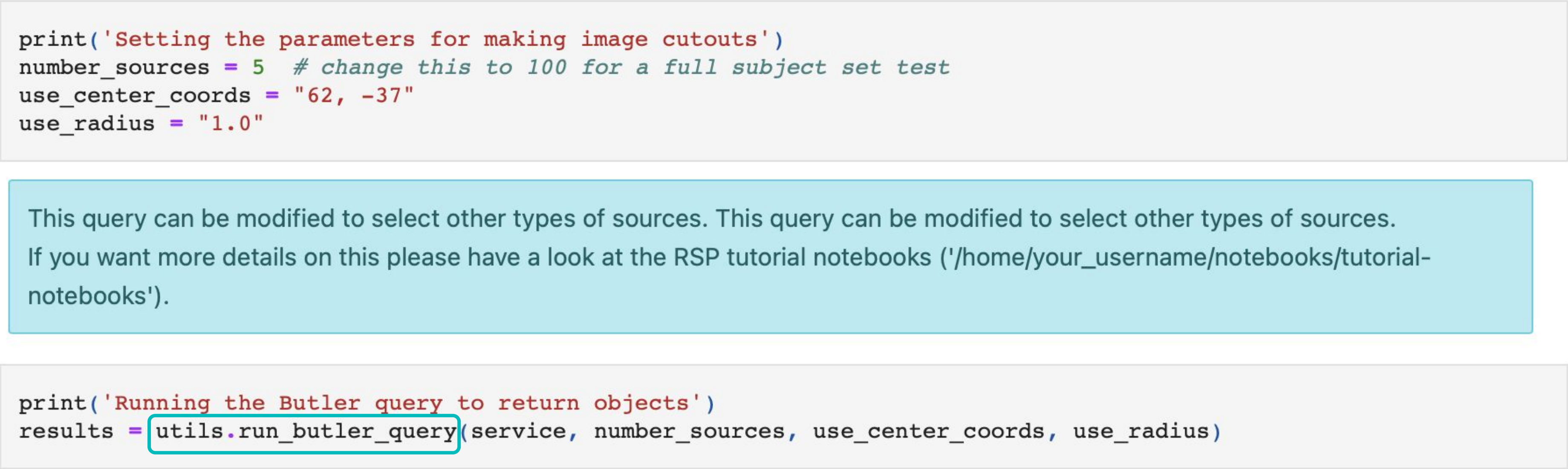
- TAP service query:
  - Retrieve tabular data
- Butler:
  - The query result is fed into Butler to retrieve FITS image



```
def setup_butler(config, collection):
    service = get_tap_service()
    assert service is not None
    assert service.baseurl == "https://data.lsst.cloud/api/tap"

    # config = 'dp02'
    # collection = '2.2i/runs/DP0.2'
    butler = dafbutler.Butler(config, collections=collection)
    skymap = butler.get("skyMap")

    return service, butler, skymap
```



Query the data for the citsci project

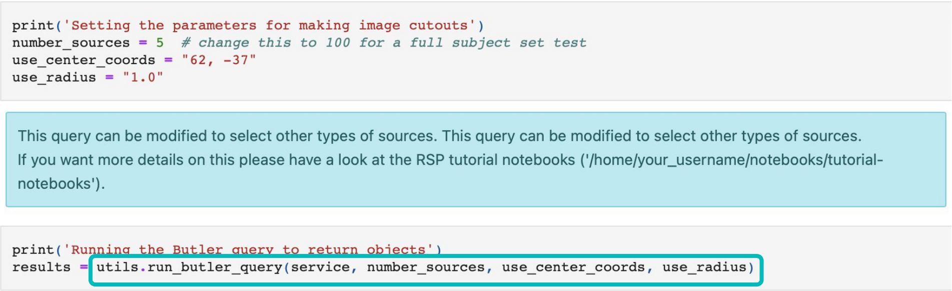


```
print('Setting the parameters for making image cutouts')
number_sources = 5  # change this to 100 for a full subject set test
use_center_coords = "62, -37"
use_radius = "1.0"
```

This query can be modified to select other types of sources. This query can be modified to select other types of sources.
If you want more details on this please have a look at the RSP tutorial notebooks ('/home/your_username/notebooks/tutorial-notebooks').

```
print('Running the Butler query to return objects')
results = utils.run_butler_query(service, number_sources, use_center_coords, use_radius)
```

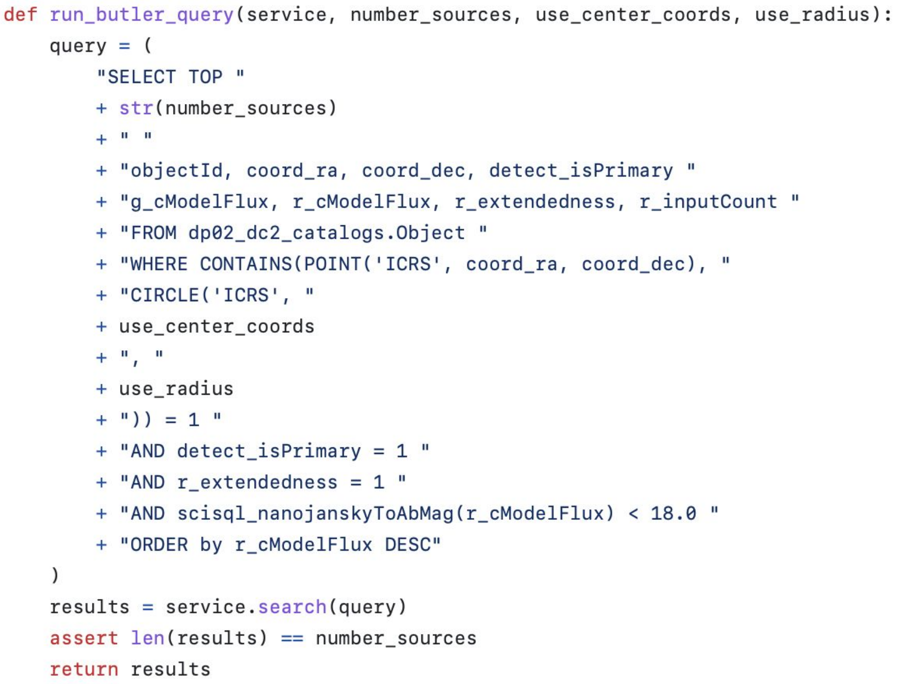
Query the data for the citsci project



```
print('Setting the parameters for making image cutouts')
number_sources = 5  # change this to 100 for a full subject set test
use_center_coords = "62, -37"
use_radius = "1.0"
```

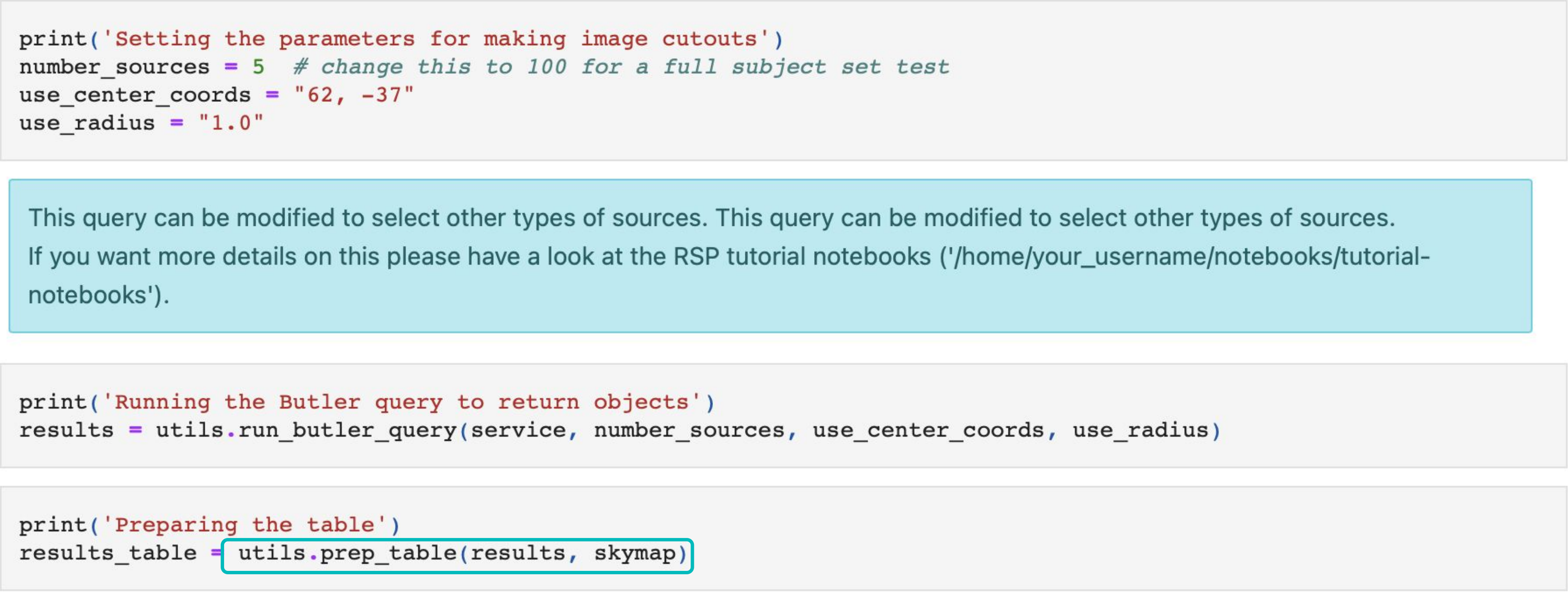
This query can be modified to select other types of sources. This query can be modified to select other types of sources.
If you want more details on this please have a look at the RSP tutorial notebooks ('/home/your_username/notebooks/tutorial-notebooks').

```
print('Running the Butler query to return objects')
results = utils.run_butler_query(service, number_sources, use_center_coords, use_radius)
```



```
def run_butler_query(service, number_sources, use_center_coords, use_radius):
    query = (
        "SELECT TOP "
        + str(number_sources)
        + " "
        + "objectId, coord_ra, coord_dec, detect_isPrimary "
        + "g_cModelFlux, r_cModelFlux, r_extendedness, r_inputCount "
        + "FROM dp02_dc2_catalogs.Object "
        + "WHERE CONTAINS(POINT('ICRS', coord_ra, coord_dec), "
        + "CIRCLE('ICRS', "
        + use_center_coords
        + ", "
        + use_radius
        + ")) = 1 "
        + "AND detect_isPrimary = 1 "
        + "AND r_extendedness = 1 "
        + "AND scisql_nanojanskyToAbMag(r_cModelFlux) < 18.0 "
        + "ORDER by r_cModelFlux DESC"
    )
    results = service.search(query)
    assert len(results) == number_sources
    return results
```

Query the data for the citsci project



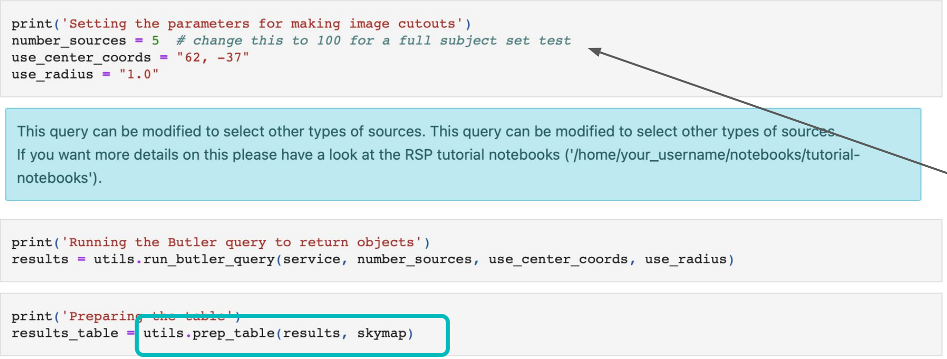
```
print('Setting the parameters for making image cutouts')
number_sources = 5  # change this to 100 for a full subject set test
use_center_coords = "62, -37"
use_radius = "1.0"
```

This query can be modified to select other types of sources. This query can be modified to select other types of sources.
If you want more details on this please have a look at the RSP tutorial notebooks ('/home/your_username/notebooks/tutorial-notebooks').

```
print('Running the Butler query to return objects')
results = utils.run_butler_query(service, number_sources, use_center_coords, use_radius)
```

```
print('Preparing the table')
results_table = utils.prep_table(results, skymap)
```

Query the data for the citsci project



```
print('Setting the parameters for making image cutouts')
number_sources = 5  # change this to 100 for a full subject set test
use_center_coords = "62, -37"
use_radius = "1.0"
```

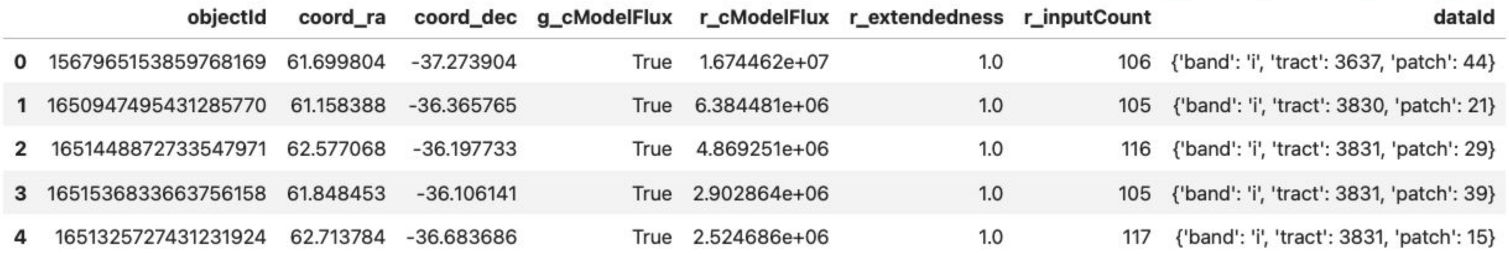
This query can be modified to select other types of sources. This query can be modified to select other types of sources. If you want more details on this please have a look at the RSP tutorial notebooks ('/home/your_username/notebooks/tutorial-notebooks').

```
print('Running the Butler query to return objects')
results = utils.run_butler_query(service, number_sources, use_center_coords, use_radius)
```

```
print('Preparing the table')
results_table = utils.prep_table(results, skymap)
```

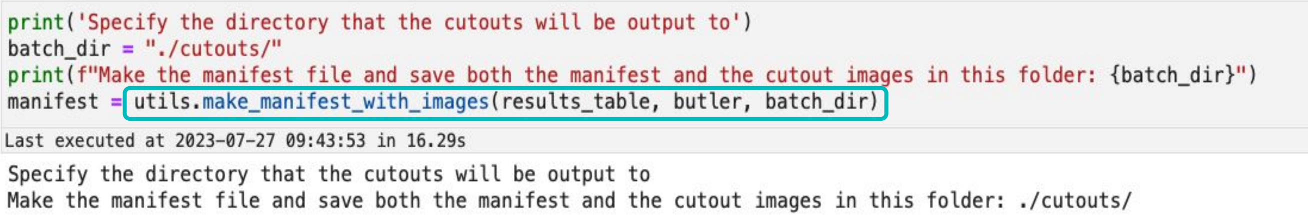
Subject set size:

- Before Approval:
  - Maximum 100 cutouts
- After Approval:
  - 10k cutouts per batch



	objectId	coord_ra	coord_dec	g_cModelFlux	r_cModelFlux	r_extendedness	r_inputCount	dataId
0	1567965153859768169	61.699804	-37.273904	True	1.674462e+07	1.0	106	{'band': 'i', 'tract': 3637, 'patch': 44}
1	1650947495431285770	61.158388	-36.365765	True	6.384481e+06	1.0	105	{'band': 'i', 'tract': 3830, 'patch': 21}
2	1651448872733547971	62.577068	-36.197733	True	4.869251e+06	1.0	116	{'band': 'i', 'tract': 3831, 'patch': 29}
3	1651536833663756158	61.848453	-36.106141	True	2.902864e+06	1.0	105	{'band': 'i', 'tract': 3831, 'patch': 39}
4	1651325727431231924	62.713784	-36.683686	True	2.524686e+06	1.0	117	{'band': 'i', 'tract': 3831, 'patch': 15}

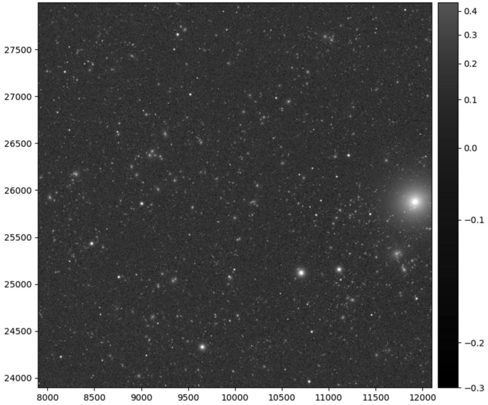
Create image cutouts and manifest list



```
print('Specify the directory that the cutouts will be output to')
batch_dir = "./cutouts/"
print(f"Make the manifest file and save both the manifest and the cutout images in this folder: {batch_dir}")
manifest = utils.make_manifest_with_images(results_table, butler, batch_dir)
```

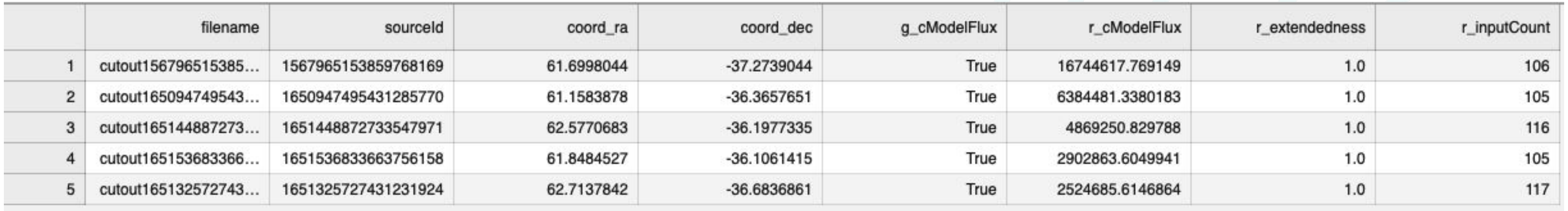
```
Last executed at 2023-07-27 09:43:53 in 16.29s
```

```
Specify the directory that the cutouts will be output to
Make the manifest file and save both the manifest and the cutout images in this folder: ./cutouts/
```



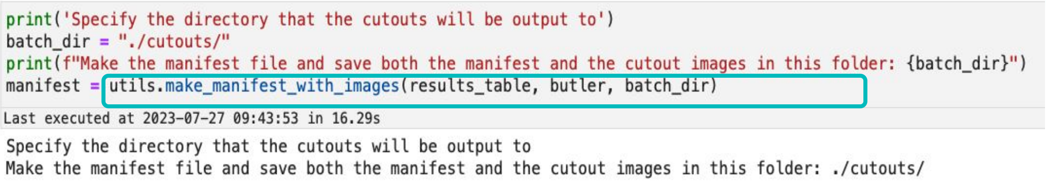
- Creates image cutouts and manifest list
- Saves them to a directory

manifest.csv



	filename	sourceId	coord_ra	coord_dec	g_cModelFlux	r_cModelFlux	r_extendedness	r_inputCount
1	cutout156796515385...	1567965153859768169	61.6998044	-37.2739044	True	16744617.769149	1.0	106
2	cutout165094749543...	1650947495431285770	61.1583878	-36.3657651	True	6384481.3380183	1.0	105
3	cutout165144887273...	1651448872733547971	62.5770683	-36.1977335	True	4869250.829788	1.0	116
4	cutout165153683366...	1651536833663756158	61.8484527	-36.1061415	True	2902863.6049941	1.0	105
5	cutout165132572743...	1651325727431231924	62.7137842	-36.6836861	True	2524685.6146864	1.0	117

Create image cutouts and manifest list

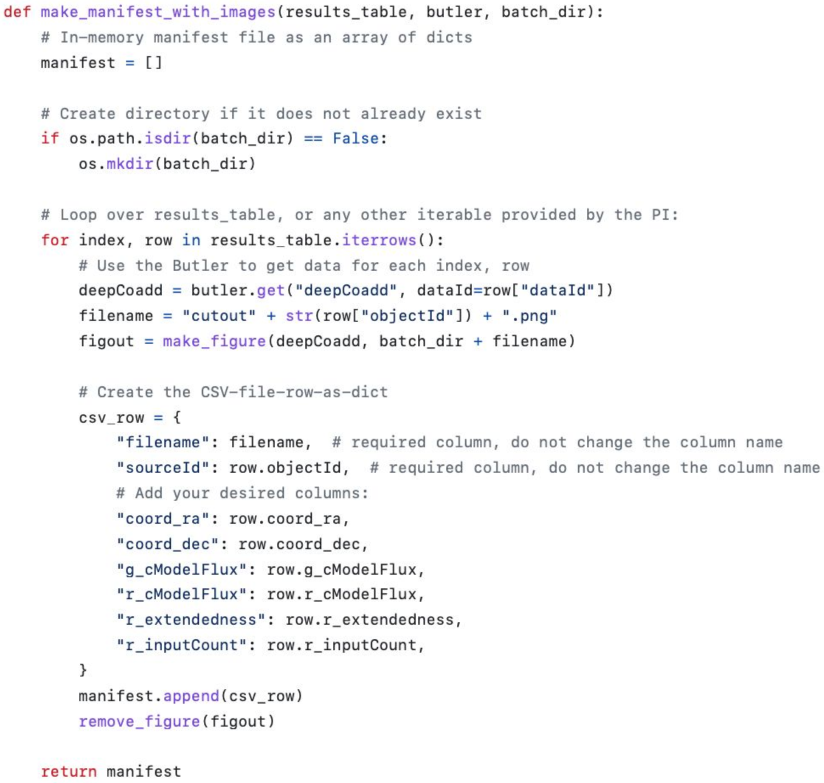


```
print('Specify the directory that the cutouts will be output to')
batch_dir = "./cutouts/"
print(f"Make the manifest file and save both the manifest and the cutout images in this folder: {batch_dir}")
manifest = utils.make_manifest_with_images(results_table, butler, batch_dir)
```

Last executed at 2023-07-27 09:43:53 in 16.29s

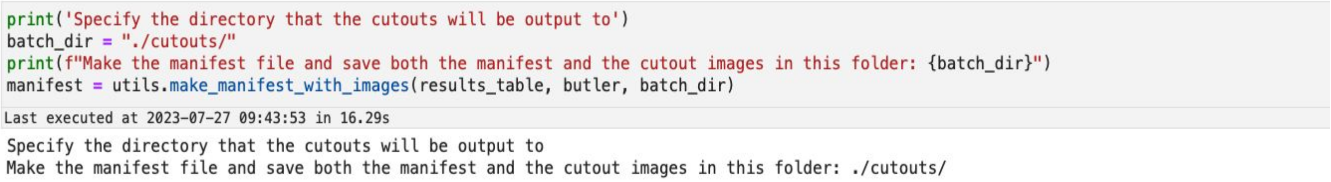
```
Specify the directory that the cutouts will be output to
Make the manifest file and save both the manifest and the cutout images in this folder: ./cutouts/
```

- Creates image cutouts and manifest list
- Saves them to a directory



```
def make_manifest_with_images(results_table, butler, batch_dir):
    # In-memory manifest file as an array of dicts
    manifest = []

    # Create directory if it does not already exist
    if os.path.isdir(batch_dir) == False:
        os.mkdir(batch_dir)

    # Loop over results_table, or any other iterable provided by the PI:
    for index, row in results_table.iterrows():
        # Use the Butler to get data for each index, row
        deepCoadd = butler.get("deepCoadd", dataId=row["dataId"])
        filename = "cutout" + str(row["objectId"]) + ".png"
        figout = make_figure(deepCoadd, batch_dir + filename)

        # Create the CSV-file-row-as-dict
        csv_row = {
            "filename": filename,  # required column, do not change the column name
            "sourceId": row.objectId,  # required column, do not change the column name
            # Add your desired columns:
            "coord_ra": row.coord_ra,
            "coord_dec": row.coord_dec,
            "g_cModelFlux": row.g_cModelFlux,
            "r_cModelFlux": row.r_cModelFlux,
            "r_extendedness": row.r_extendedness,
            "r_inputCount": row.r_inputCount,
        }
        manifest.append(csv_row)
        remove_figure(figout)

    return manifest
```

Create image cutouts and manifest list

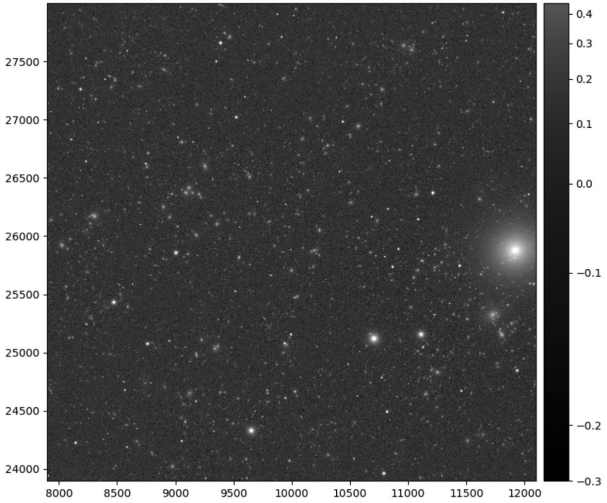


```
print('Specify the directory that the cutouts will be output to')
batch_dir = "./cutouts/"
print(f"Make the manifest file and save both the manifest and the cutout images in this folder: {batch_dir}")
manifest = utils.make_manifest_with_images(results_table, butler, batch_dir)
```

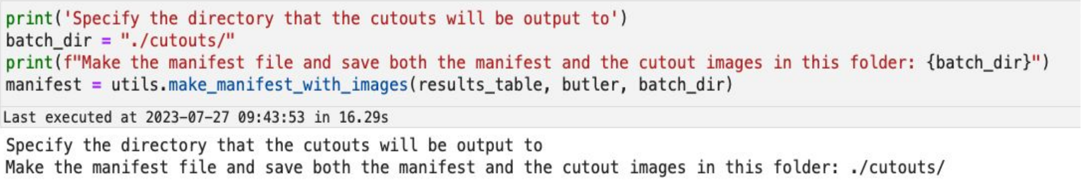
Last executed at 2023-07-27 09:43:53 in 16.29s

```
Specify the directory that the cutouts will be output to
Make the manifest file and save both the manifest and the cutout images in this folder: ./cutouts/
```

- Creates image cutouts and manifest list
- Saves them to a directory



Create image cutouts and manifest list

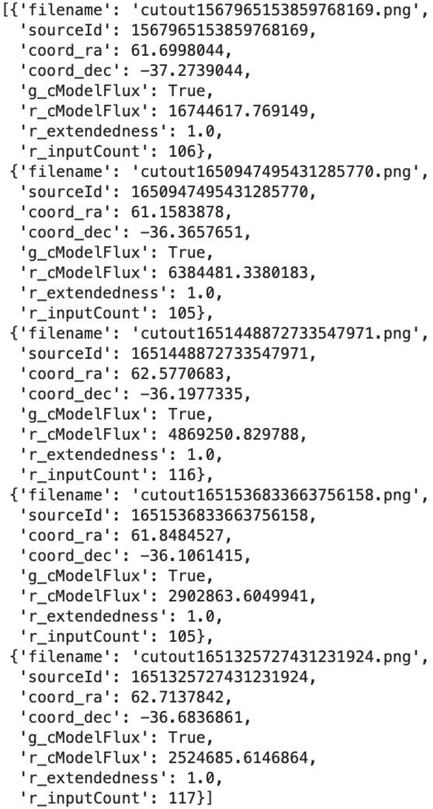


```
print('Specify the directory that the cutouts will be output to')
batch_dir = "./cutouts/"
print(f"Make the manifest file and save both the manifest and the cutout images in this folder: {batch_dir}")
manifest = utils.make_manifest_with_images(results_table, butler, batch_dir)
```

```
Last executed at 2023-07-27 09:43:53 in 16.29s
```

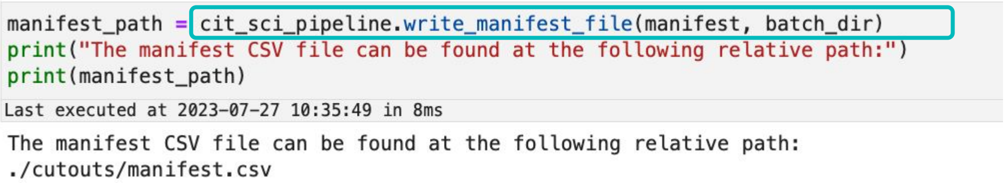
```
Specify the directory that the cutouts will be output to
Make the manifest file and save both the manifest and the cutout images in this folder: ./cutouts/
```

- Creates image cutouts and manifest list
- Saves them to a directory

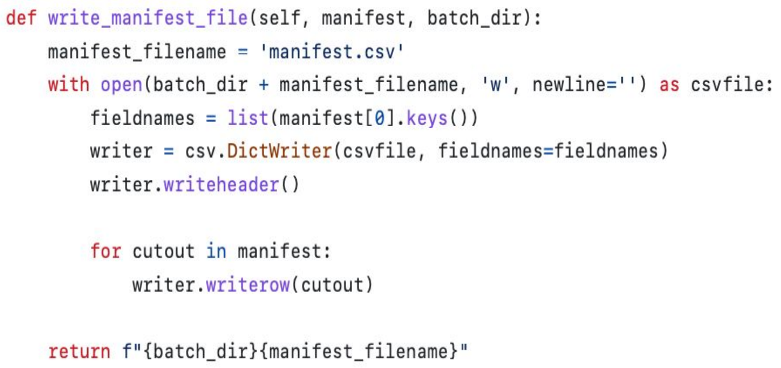


```
[{'filename': 'cutout1567965153859768169.png',
  'sourceId': 1567965153859768169,
  'coord_ra': 61.6998044,
  'coord_dec': -37.2739044,
  'g_cModelFlux': True,
  'r_cModelFlux': 16744617.769149,
  'r_extendedness': 1.0,
  'r_inputCount': 106},
 {'filename': 'cutout1650947495431285770.png',
  'sourceId': 1650947495431285770,
  'coord_ra': 61.1583878,
  'coord_dec': -36.3657651,
  'g_cModelFlux': True,
  'r_cModelFlux': 6384481.3380183,
  'r_extendedness': 1.0,
  'r_inputCount': 105},
 {'filename': 'cutout1651448872733547971.png',
  'sourceId': 1651448872733547971,
  'coord_ra': 62.5770683,
  'coord_dec': -36.1977335,
  'g_cModelFlux': True,
  'r_cModelFlux': 4869250.829788,
  'r_extendedness': 1.0,
  'r_inputCount': 116},
 {'filename': 'cutout1651536833663756158.png',
  'sourceId': 1651536833663756158,
  'coord_ra': 61.8484527,
  'coord_dec': -36.1061415,
  'g_cModelFlux': True,
  'r_cModelFlux': 2902863.6049941,
  'r_extendedness': 1.0,
  'r_inputCount': 105},
 {'filename': 'cutout1651325727431231924.png',
  'sourceId': 1651325727431231924,
  'coord_ra': 62.7137842,
  'coord_dec': -36.6836861,
  'g_cModelFlux': True,
  'r_cModelFlux': 2524685.6146864,
  'r_extendedness': 1.0,
  'r_inputCount': 117}]
```

Write manifest file to RSP filesystem

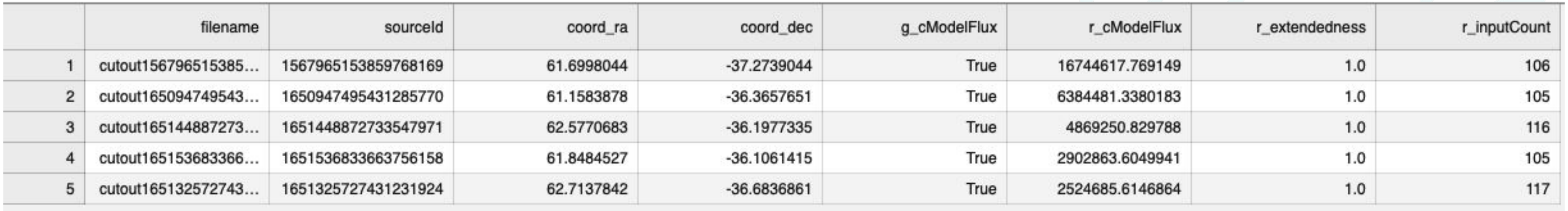


```
manifest_path = cit_sci_pipeline.write_manifest_file(manifest, batch_dir)
print("The manifest CSV file can be found at the following relative path:")
print(manifest_path)
Last executed at 2023-07-27 10:35:49 in 8ms
The manifest CSV file can be found at the following relative path:
./cutouts/manifest.csv
```



```
def write_manifest_file(self, manifest, batch_dir):
    manifest_filename = 'manifest.csv'
    with open(batch_dir + manifest_filename, 'w', newline='') as csvfile:
        fieldnames = list(manifest[0].keys())
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()

        for cutout in manifest:
            writer.writerow(cutout)

    return f"{batch_dir}{manifest_filename}"
```

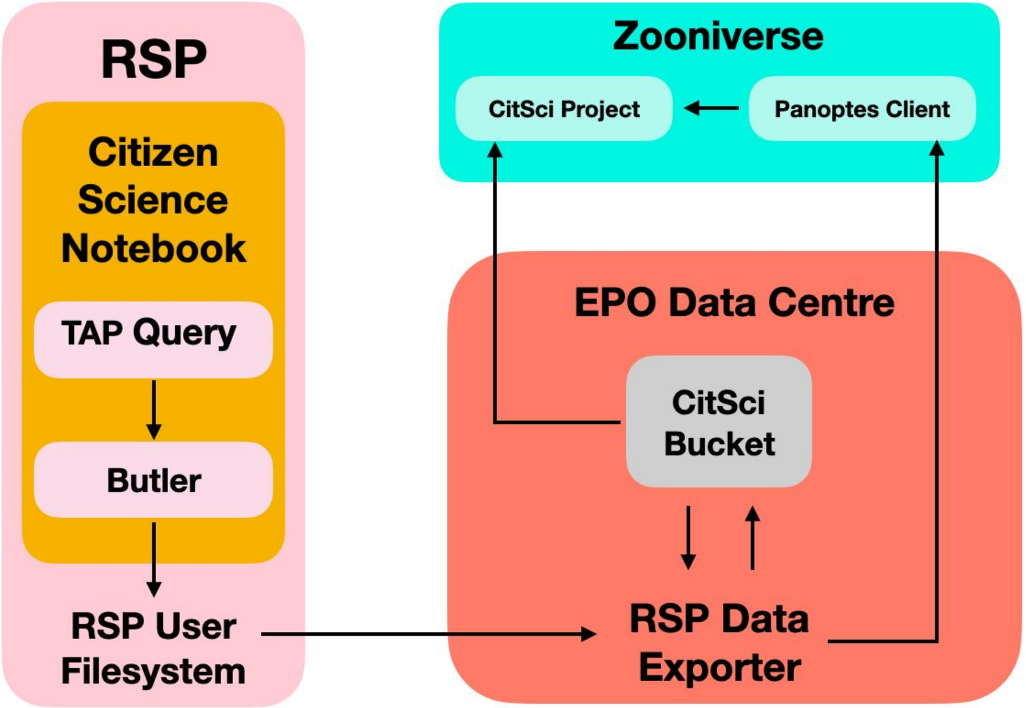
manifest.csv



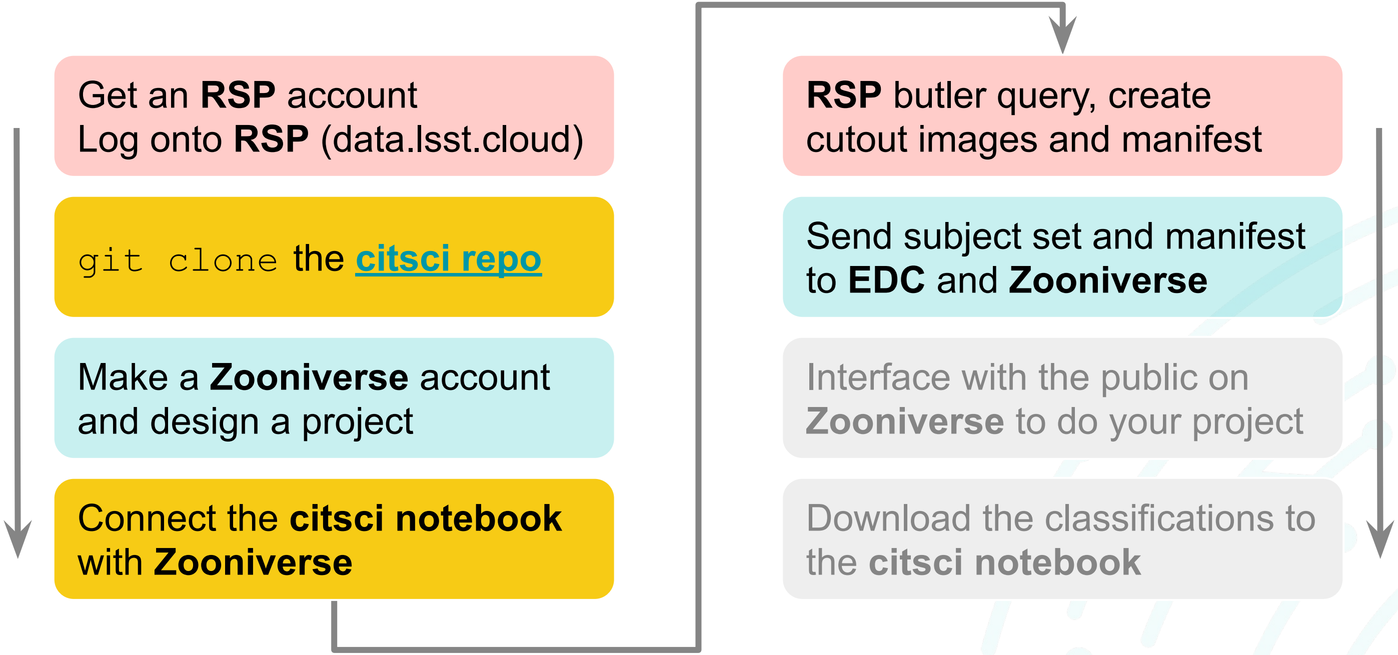
	filename	sourceId	coord_ra	coord_dec	g_cModelFlux	r_cModelFlux	r_extendedness	r_inputCount
1	cutout156796515385...	1567965153859768169	61.6998044	-37.2739044	True	16744617.769149	1.0	106
2	cutout165094749543...	1650947495431285770	61.1583878	-36.3657651	True	6384481.3380183	1.0	105
3	cutout165144887273...	1651448872733547971	62.5770683	-36.1977335	True	4869250.829788	1.0	116
4	cutout165153683366...	1651536833663756158	61.8484527	-36.1061415	True	2902863.6049941	1.0	105
5	cutout165132572743...	1651325727431231924	62.7137842	-36.6836861	True	2524685.6146864	1.0	117

Rubinverse Workflow

- RSP filesystem:
  - The image cutouts and manifest csv file are saved in the RSP filesystem
  - Zipped and uploaded to EDC
- RSP data exporter:
  - Looks up user and project ID in Zooniverse
  - Uploads the cutouts to Citsci Bucket and manifest URL created
- Zooniverse:
  - Manifest URL sent from RSP to Zooniverse is parsed



RubinVerse workflow



Get an **RSP** account
Log onto **RSP** (data.lsst.cloud)

`git clone` the [citsci repo]

Make a **Zooniverse** account and design a project

Connect the **citsci notebook** with **Zooniverse**

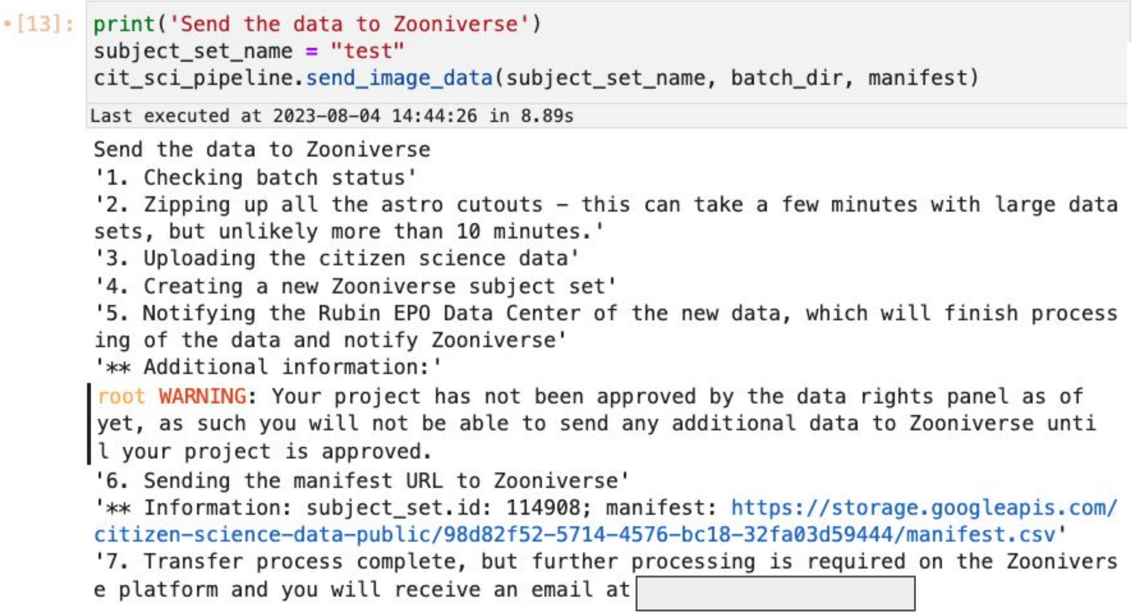
RSP butler query, create cutout images and manifest

Send subject set and manifest to **EDC** and **Zooniverse**

Interface with the public on **Zooniverse** to do your project

Download the classifications to the **citsci notebook**

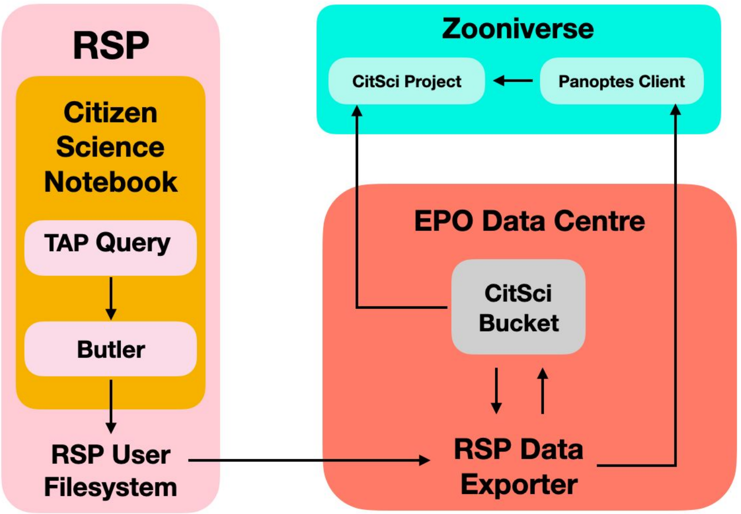
Send data to EDC and Zooniverse



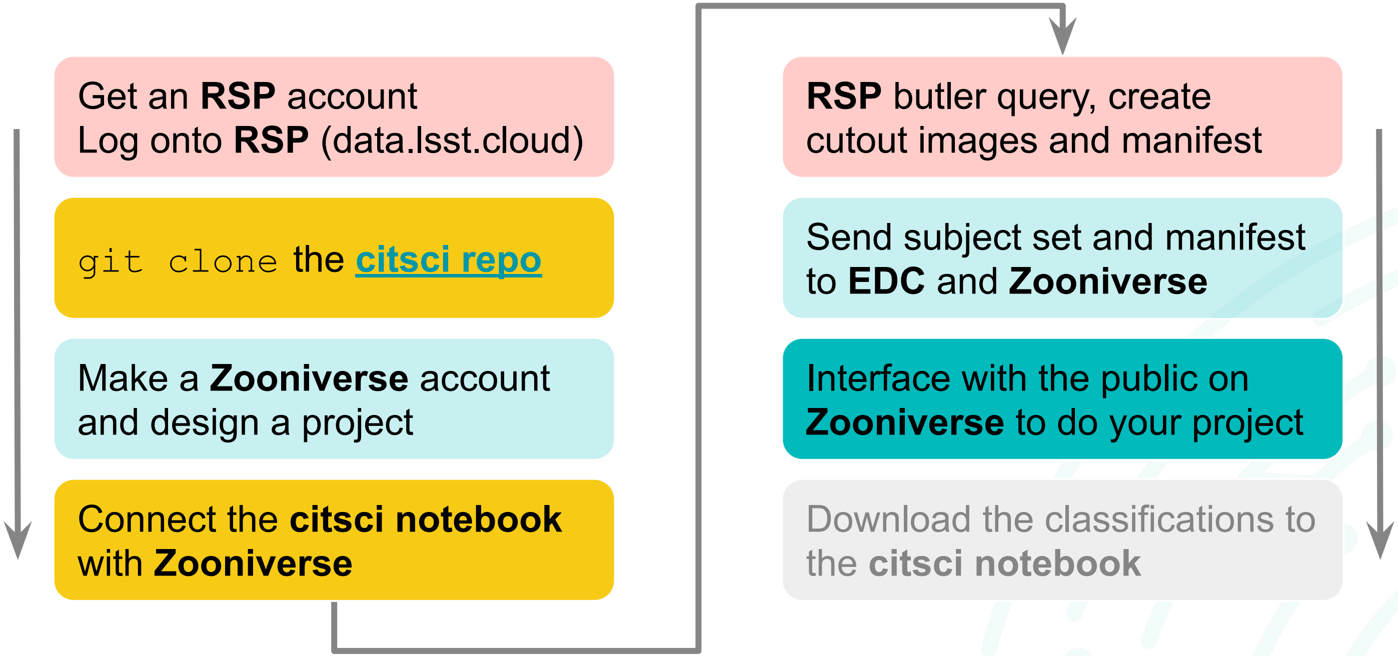
```
[13]: print('Send the data to Zooniverse')
      subject_set_name = "test"
      cit_sci_pipeline.send_image_data(subject_set_name, batch_dir, manifest)
```

Last executed at 2023-08-04 14:44:26 in 8.89s

```
Send the data to Zooniverse
'1. Checking batch status'
'2. Zipping up all the astro cutouts - this can take a few minutes with large data
sets, but unlikely more than 10 minutes.'
'3. Uploading the citizen science data'
'4. Creating a new Zooniverse subject set'
'5. Notifying the Rubin EPO Data Center of the new data, which will finish process
ing of the data and notify Zooniverse'
'** Additional information:'
root WARNING: Your project has not been approved by the data rights panel as of
yet, as such you will not be able to send any additional data to Zooniverse unti
l your project is approved.
'6. Sending the manifest URL to Zooniverse'
'** Information: subject_set.id: 114908; manifest: https://storage.googleapis.com/
citizen-science-data-public/98d82f52-5714-4576-bc18-32fa03d59444/manifest.csv'
'7. Transfer process complete, but further processing is required on the Zoonivers
e platform and you will receive an email at [redacted]
```



RubinVerse workflow



Get an **RSP** account
Log onto **RSP** (data.lsst.cloud)

`git clone` the citsci repo

Make a **Zooniverse** account and design a project

Connect the **citsci notebook** with **Zooniverse**

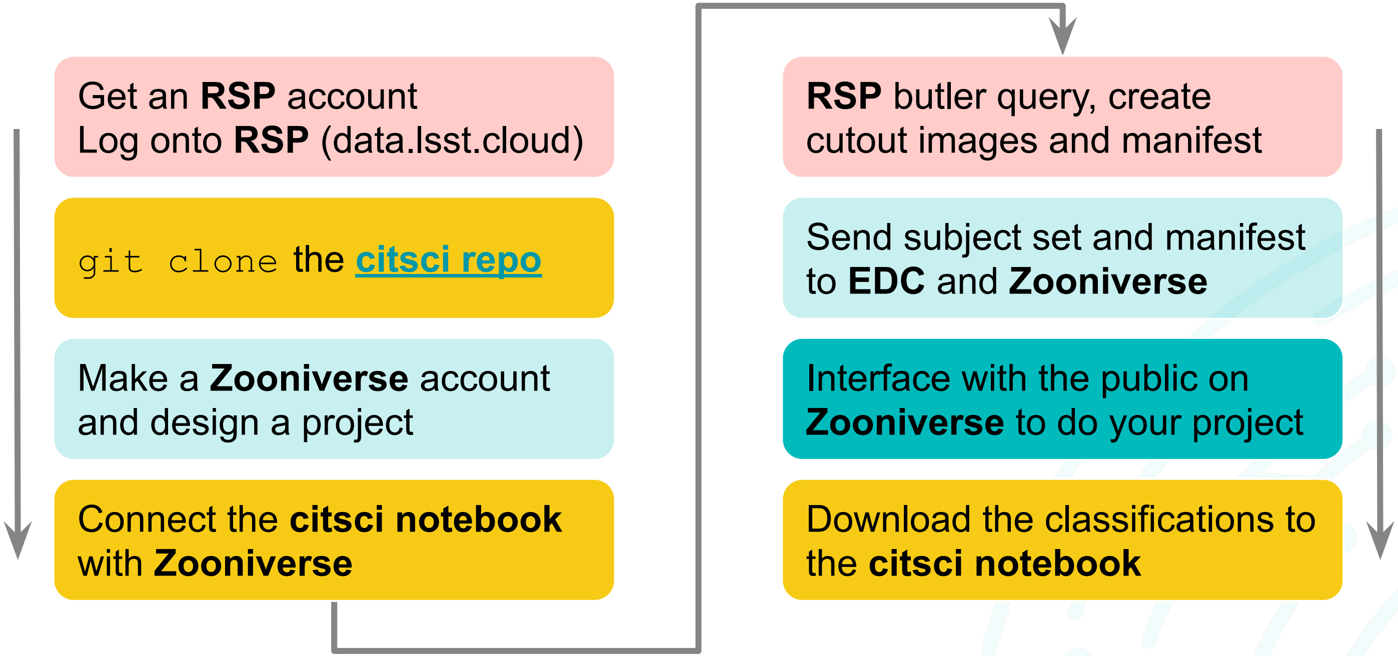
RSP butler query, create cutout images and manifest

Send subject set and manifest to **EDC** and **Zooniverse**

Interface with the public on **Zooniverse** to do your project

Download the classifications to the **citsci notebook**

RubinVerse workflow



Get an **RSP** account
Log onto **RSP** (data.lsst.cloud)

`git clone` the citsci repo

Make a **Zooniverse** account and design a project

Connect the **citsci notebook** with **Zooniverse**

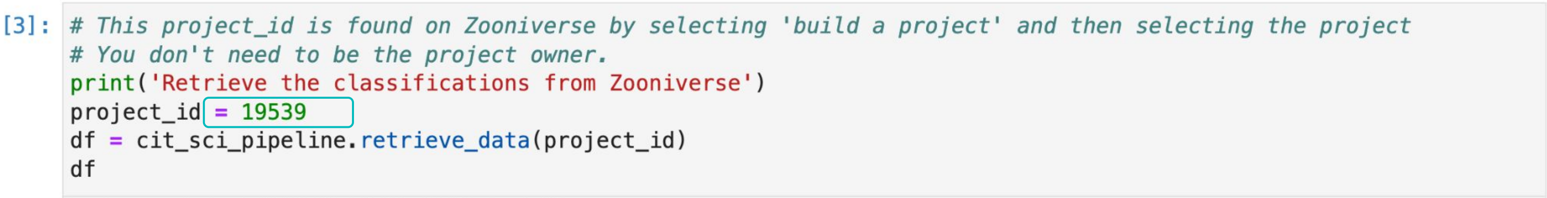
RSP butler query, create cutout images and manifest

Send subject set and manifest to **EDC** and **Zooniverse**

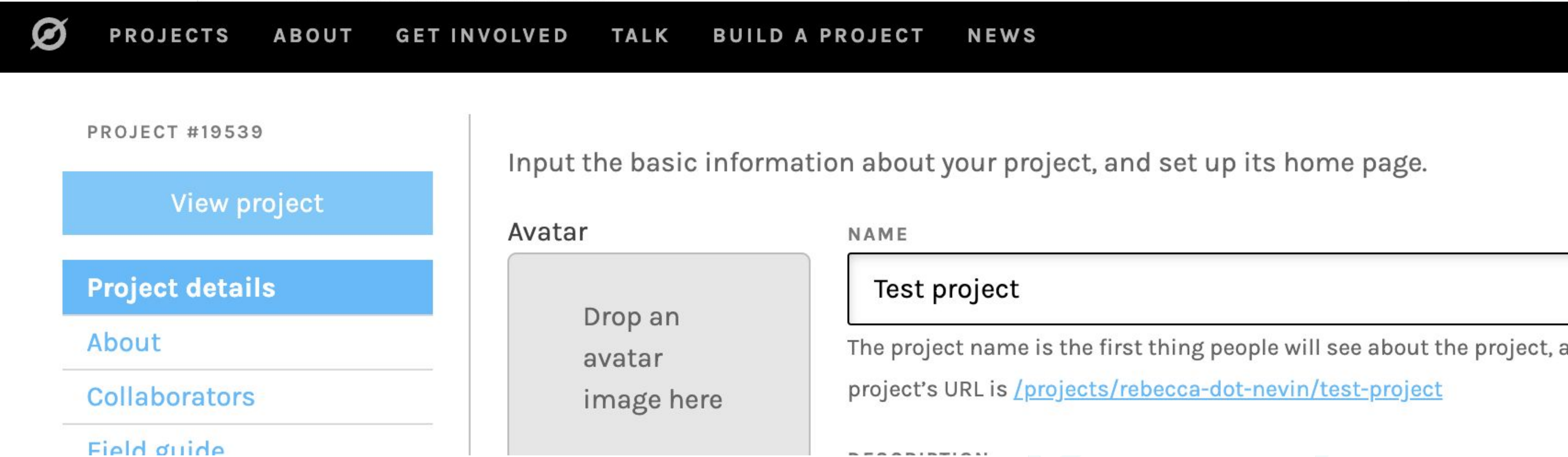
Interface with the public on **Zooniverse** to do your project

Download the classifications to the **citsci notebook**

Retrieve the classifications



```
[3]: # This project_id is found on Zooniverse by selecting 'build a project' and then selecting the project
     # You don't need to be the project owner.
     print('Retrieve the classifications from Zooniverse')
     project_id = 19539
     df = cit_sci_pipeline.retrieve_data(project_id)
     df
```



Input the basic information about your project, and set up its home page.

Avatar

Drop an avatar image here

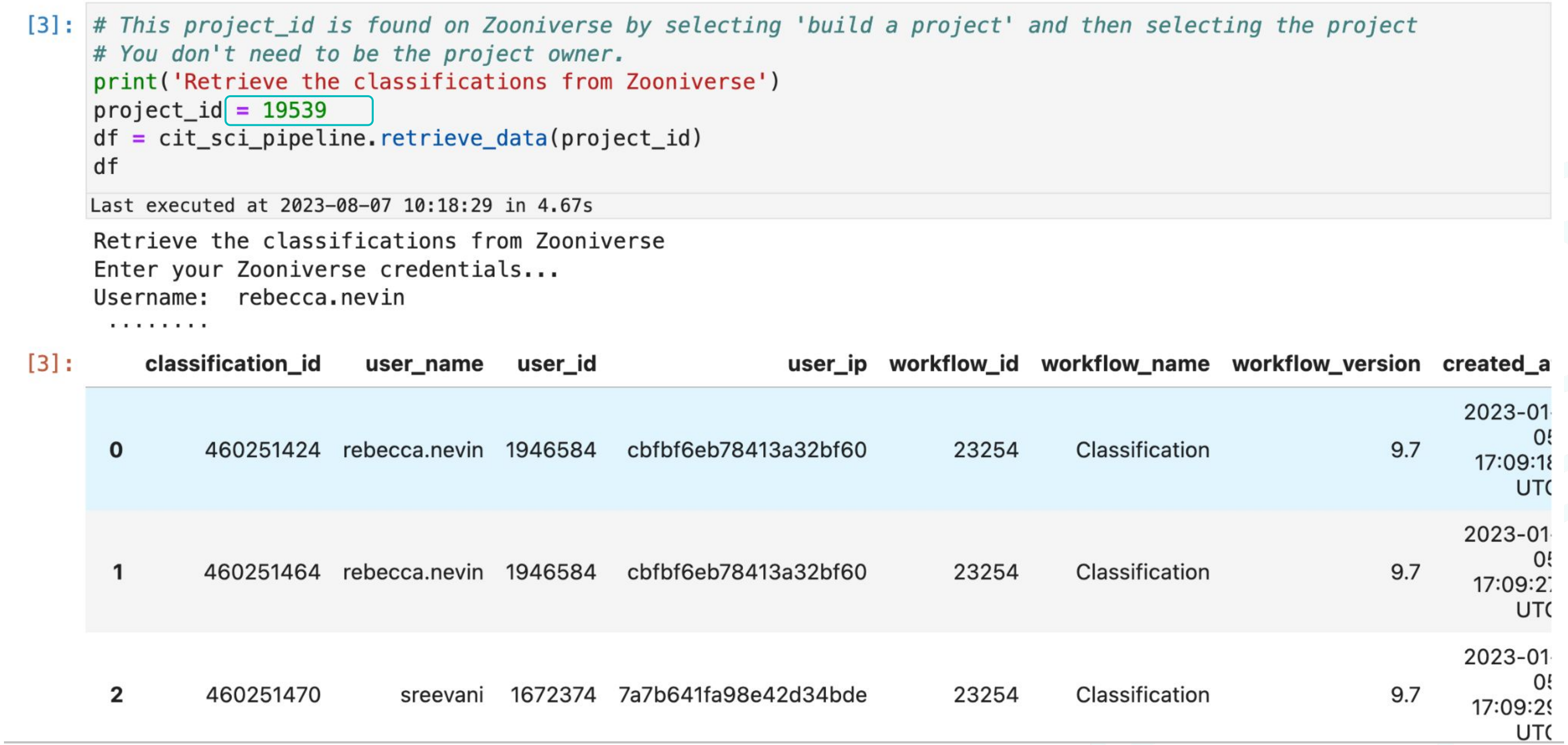
NAME

Test project

The project name is the first thing people will see about the project, a

project's URL is /projects/rebecca-dot-nevin/test-project

Retrieve the classifications



```
[3]: # This project_id is found on Zooniverse by selecting 'build a project' and then selecting the project
     # You don't need to be the project owner.
     print('Retrieve the classifications from Zooniverse')
     project_id = 19539
     df = cit_sci_pipeline.retrieve_data(project_id)
     df
```

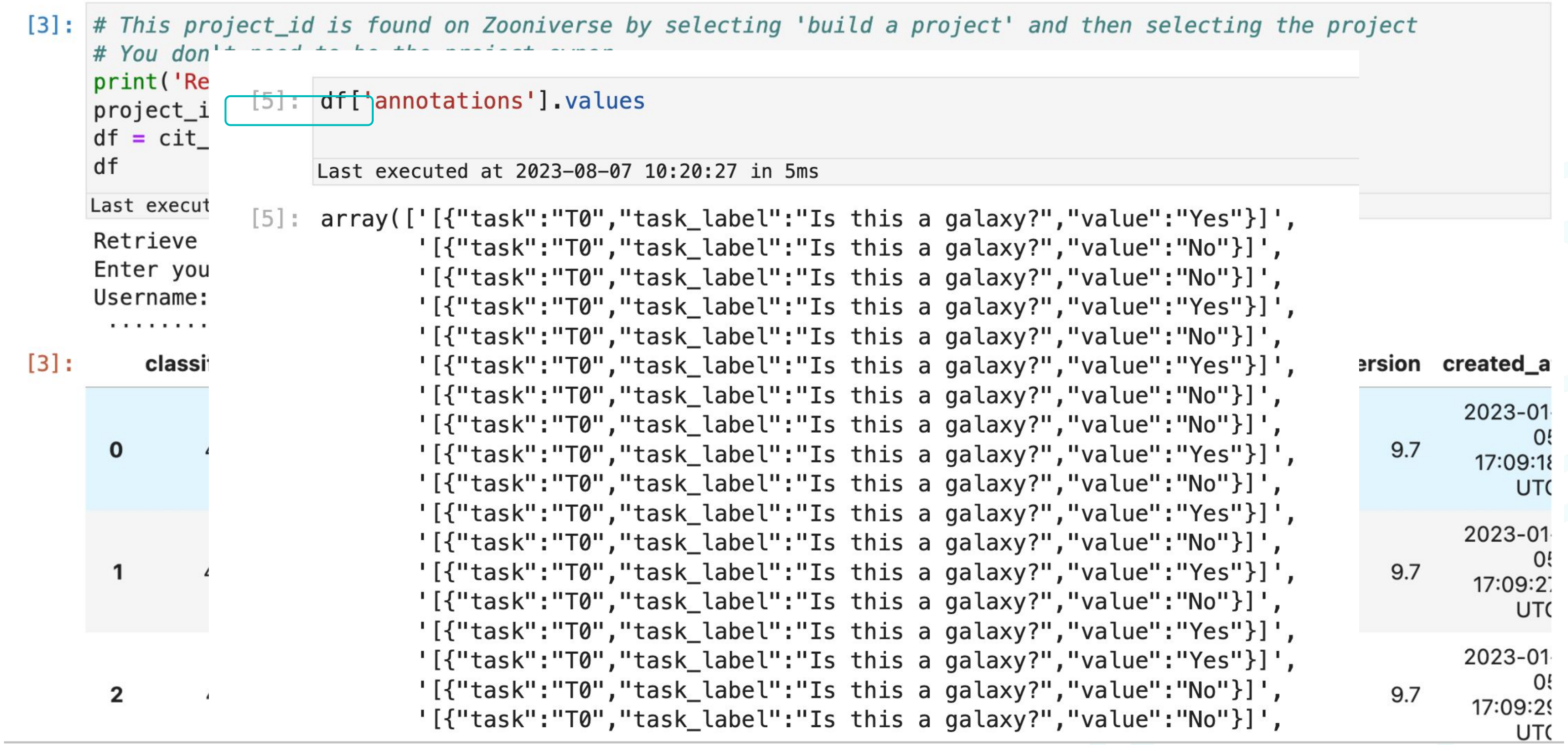
```
Last executed at 2023-08-07 10:18:29 in 4.67s
```

```
Retrieve the classifications from Zooniverse
Enter your Zooniverse credentials...
Username:  rebecca.nevin
 ........
```

[3]:

	classification_id	user_name	user_id	user_ip	workflow_id	workflow_name	workflow_version	created_a
0	460251424	rebecca.nevin	1946584	cbfbf6eb78413a32bf60	23254	Classification	9.7	2023-01-05 17:09:18 UTC
1	460251464	rebecca.nevin	1946584	cbfbf6eb78413a32bf60	23254	Classification	9.7	2023-01-05 17:09:27 UTC
2	460251470	sreevani	1672374	7a7b641fa98e42d34bde	23254	Classification	9.7	2023-01-05 17:09:29 UTC

Retrieve the classifications



```
[3]: # This project_id is found on Zooniverse by selecting 'build a project' and then selecting the project
     # You don'
     print('Re
     project_i
     df = cit_
     df

     Last execut

     Retrieve
     Enter you
     Username:
      ........

[3]:     classi                                              ersion   created_a
     0                                                        9.7    2023-01-
                                                                          05
                                                                     17:09:18
                                                                         UTC
     1                                                        9.7    2023-01-
                                                                          05
                                                                     17:09:2
                                                                         UTC
     2                                                        9.7    2023-01-
                                                                          05
                                                                     17:09:2
                                                                         UTC
```

```
[5]: df['annotations'].values

     Last executed at 2023-08-07 10:20:27 in 5ms

[5]: array(['[{"task":"T0","task_label":"Is this a galaxy?","value":"Yes"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"No"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"No"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"Yes"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"No"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"Yes"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"No"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"No"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"Yes"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"No"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"Yes"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"No"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"Yes"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"No"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"Yes"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"Yes"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"No"}]',
            '[{"task":"T0","task_label":"Is this a galaxy?","value":"No"}]',
```

VERA C. RUBIN
OBSERVATORY

Town Hall Discussion

NSF NOIRLab AURA SLAC U.S. DEPARTMENT OF ENERGY

Outstanding Questions under consideration

- What would make you most likely to take on a project?
  - How can we make this the best tool for you?
- How do you feel about projects that overlap in subject area?
  - Duplicated projects devalue the public’s time
- What would be your weekly time commitment for a project like this?
  - How much effort would you be able to commit to your community of citizen scientists?
- When is a project finished?
  - Data sharing and archiving to best respect the effort of the public

- Slack Channel: #day2-tue-1600-rubinverse-citizen-sci
- Email: cscience@lsst.org
- Come by our EPO Open House on Thursday!
- Ask/comment anonymously: https://ls.st/cscifeedback

VERA C. RUBIN
OBSERVATORY

Wrap Up

Looking towards an exciting future!

NSF NOIRLab AURA SLAC U.S. DEPARTMENT OF ENERGY

What to do now?

- Planning and preparation
  - Science collaborations
  - Information on our website
- “Expression of interest” form
  - Participate in early testing
- Reach out - we’d love to hear from you!
  - Developing this pipeline further and we appreciate your input

rubinobs.org/for-scientists/citizen-science

ls.st/citsciform

cscience@lsst.org

- Email: **cscience@lsst.org**
- Website: **rubinobs.org/for-scientists/citizen-science**
- Expression of interest form: **https://ls.st/citsciform**
- Come by our EPO Open House on Thursday!

We are striving to make the best tool for and support the meaningful and exciting projects for the public to engage in - we look forward to hearing from you!

Rubin Observatory/NSF/AURA