

# JUPYTER HUB & CANVAS INTEGRATION

DSCI 100

# Goals for DSCI 100

## **Students - Focus on the code!**

- Code in the cloud—no device restrictions or complex installation prerequisites.
- Canvas as **single location** for students to navigate to.
- No need to interface directly with git or GitHub.
- Transferable knowledge—no “special” sandboxed environments that do not translate to real-world use.

# Goals for DSCI 100

**Instructors - Reduce administrative burden!**

- Ease of configuration
- Automated course setup
- True automated grading



- A learning management system to manage assignments.
- Serves as single location for students to visit.



nbgrader

- A server framework to host Jupyter notebooks and authenticate users
- An automated grading system.

# Overview

CANVAS

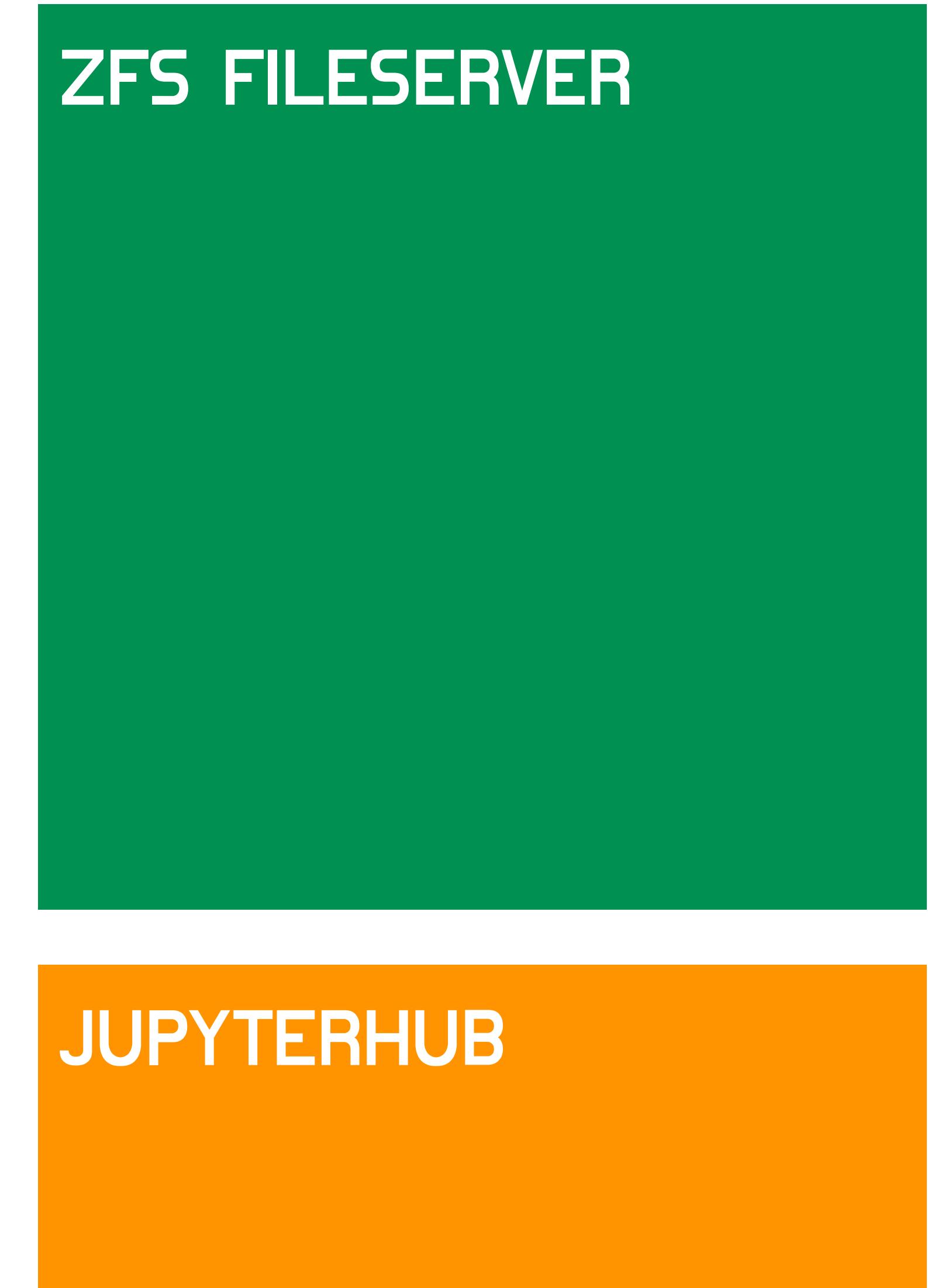
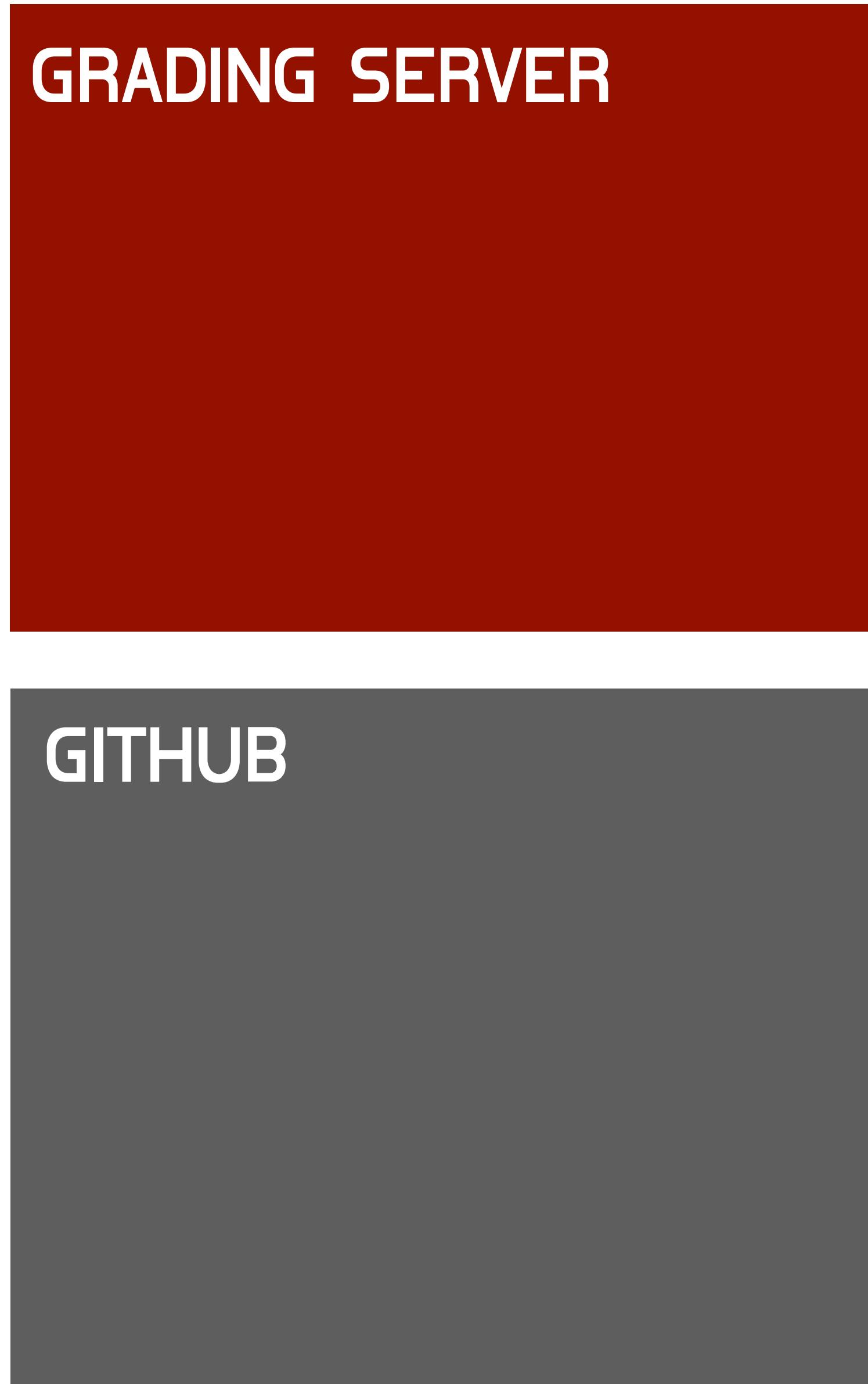
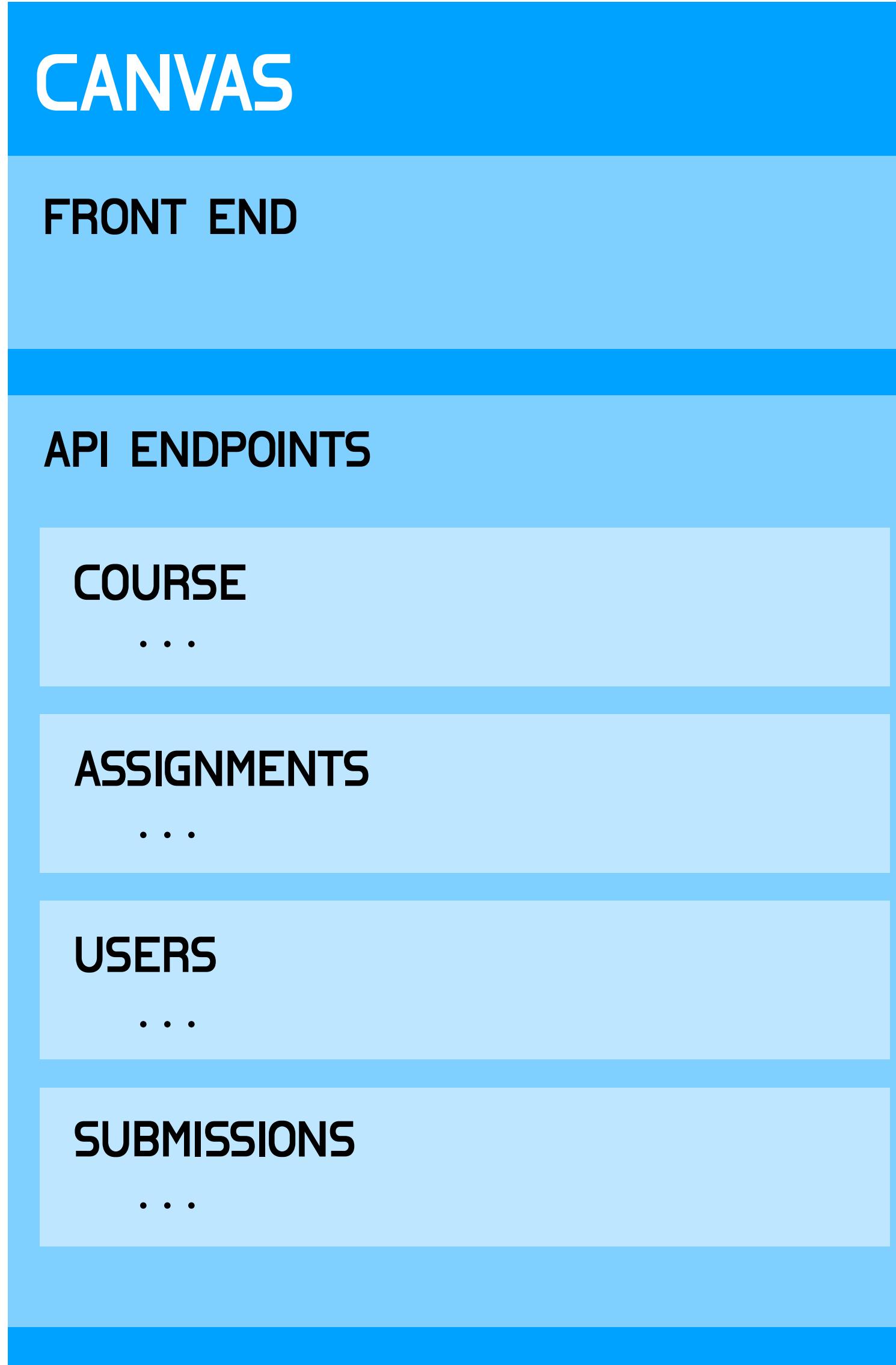
GRADING SERVER

ZFS FILESERVER

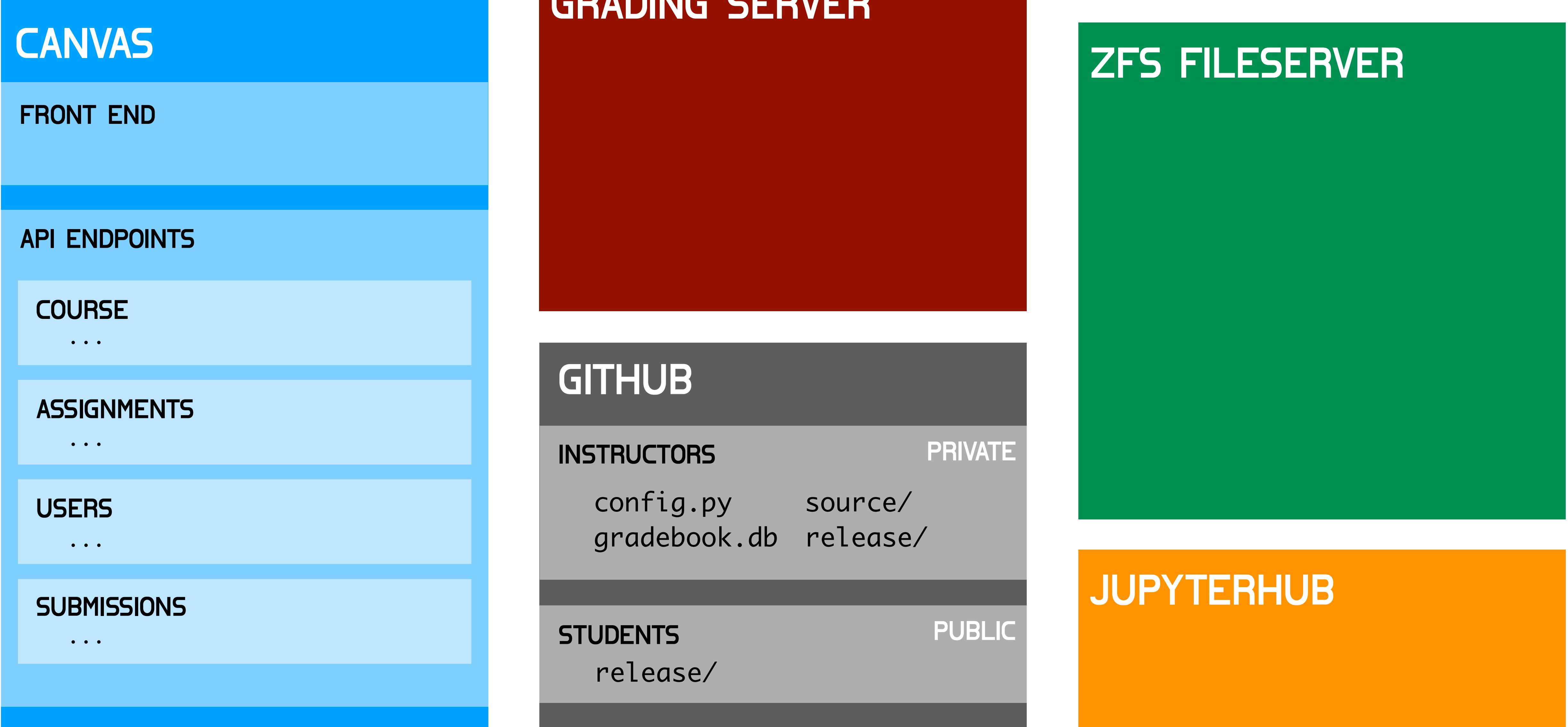
GITHUB

JUPYTERHUB

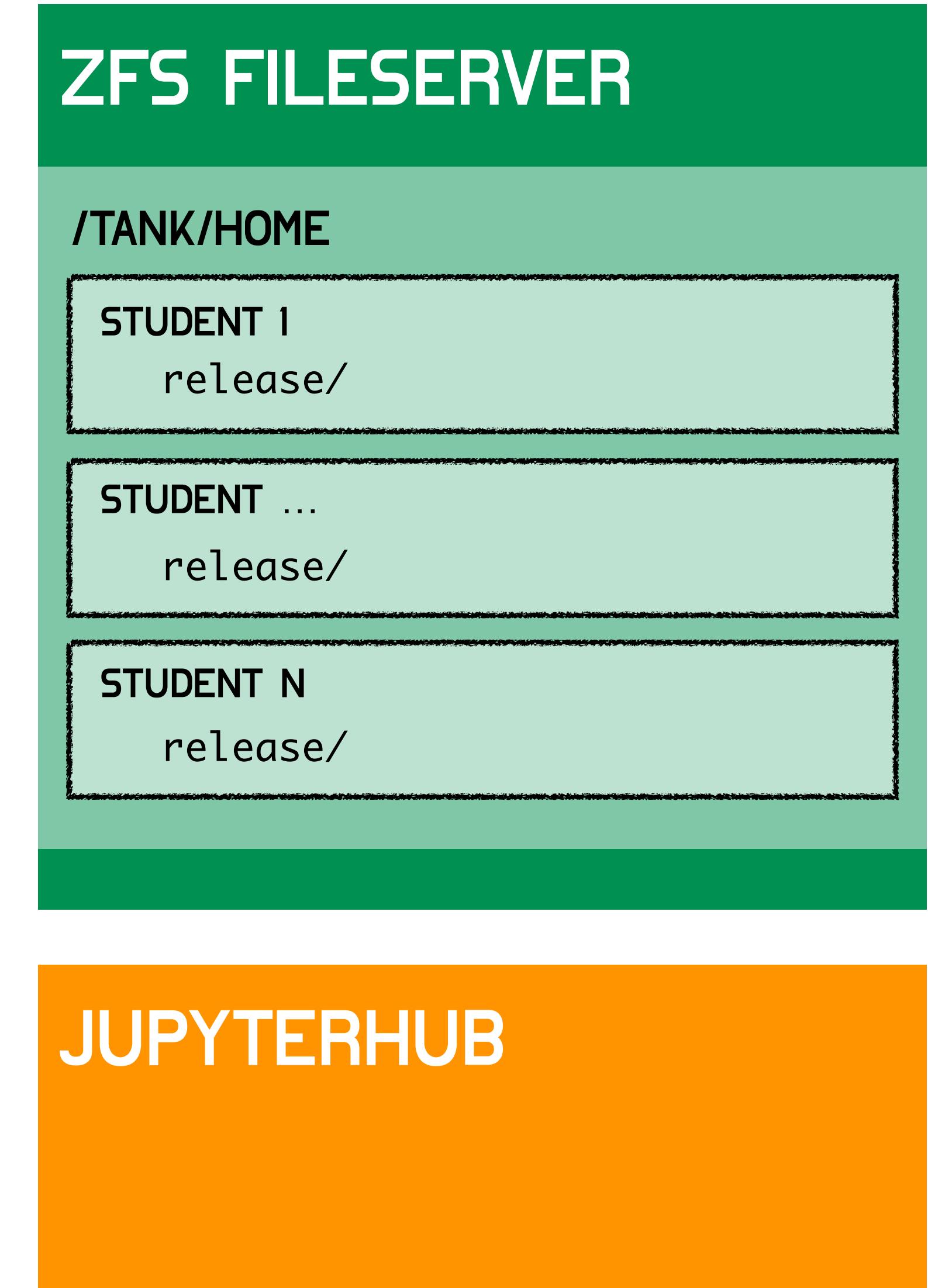
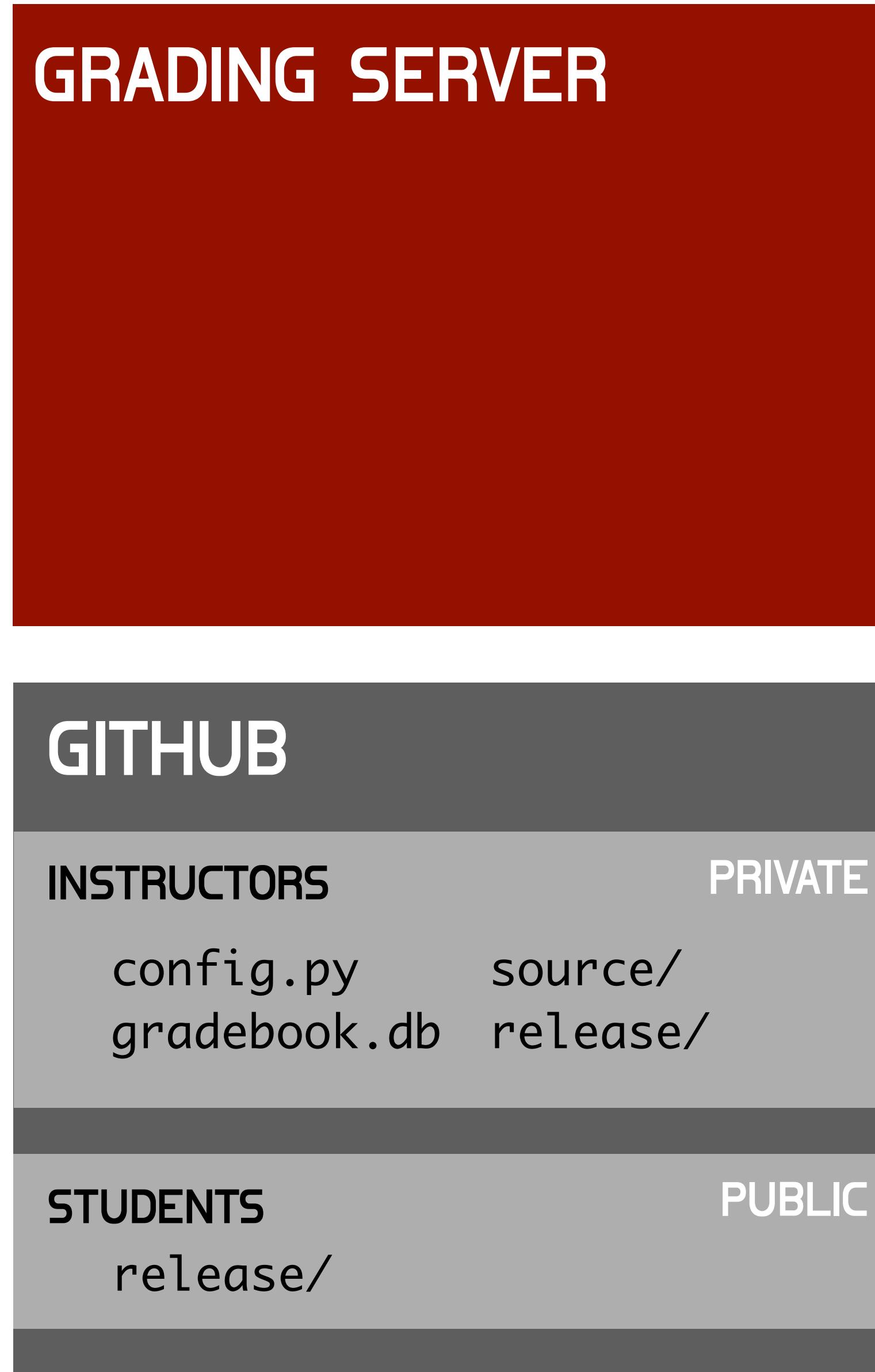
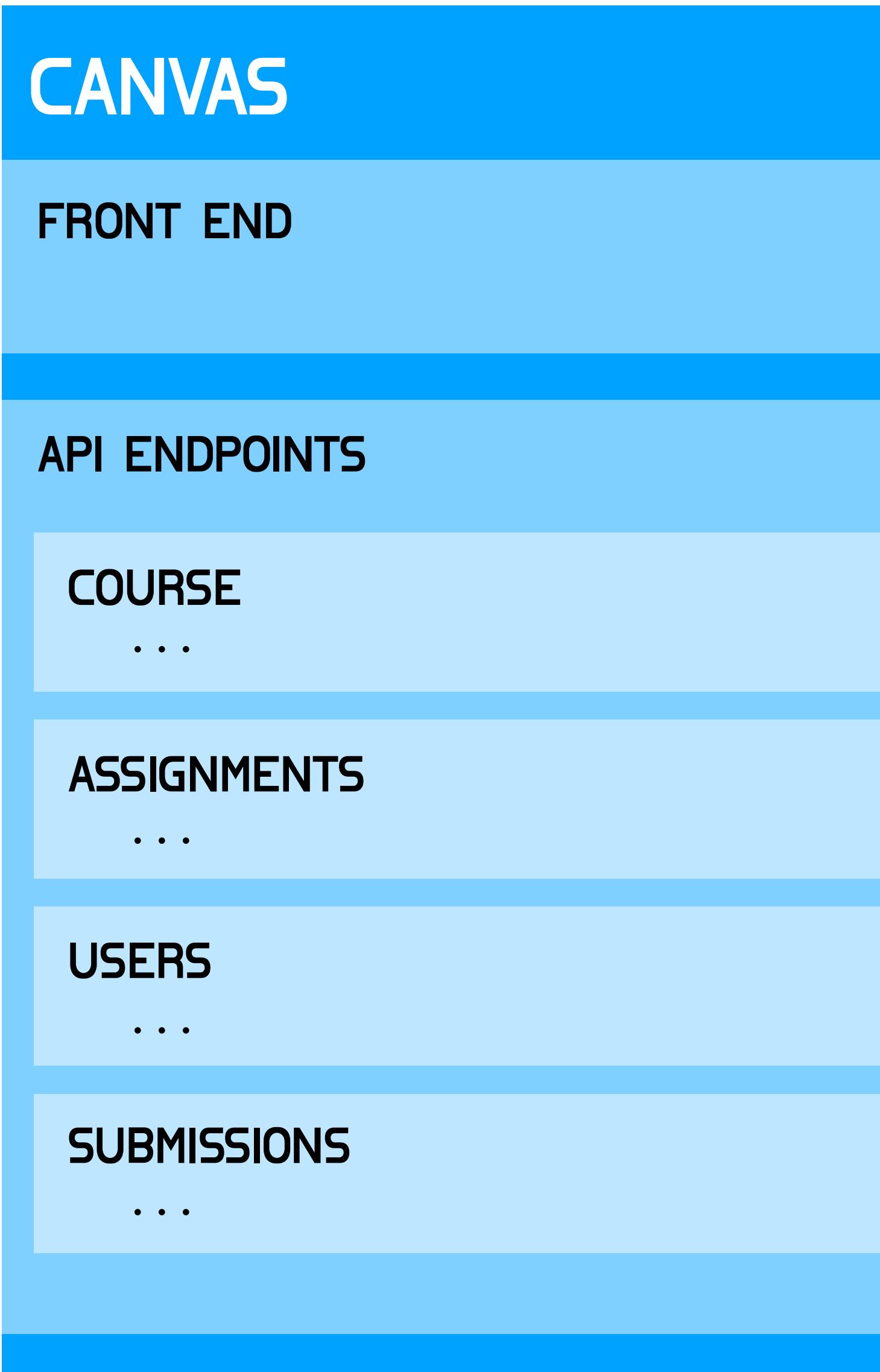
# Overview



# Overview

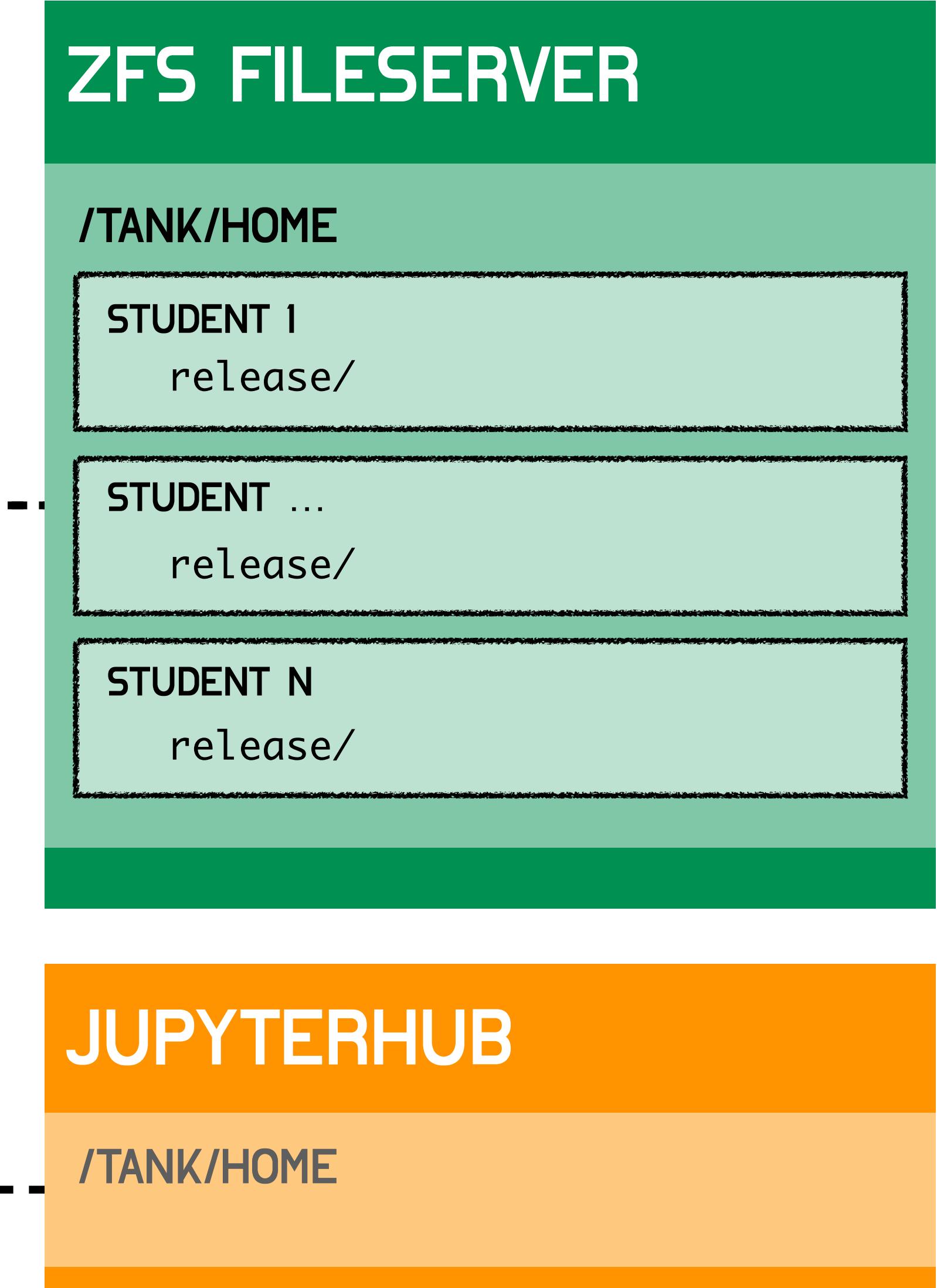
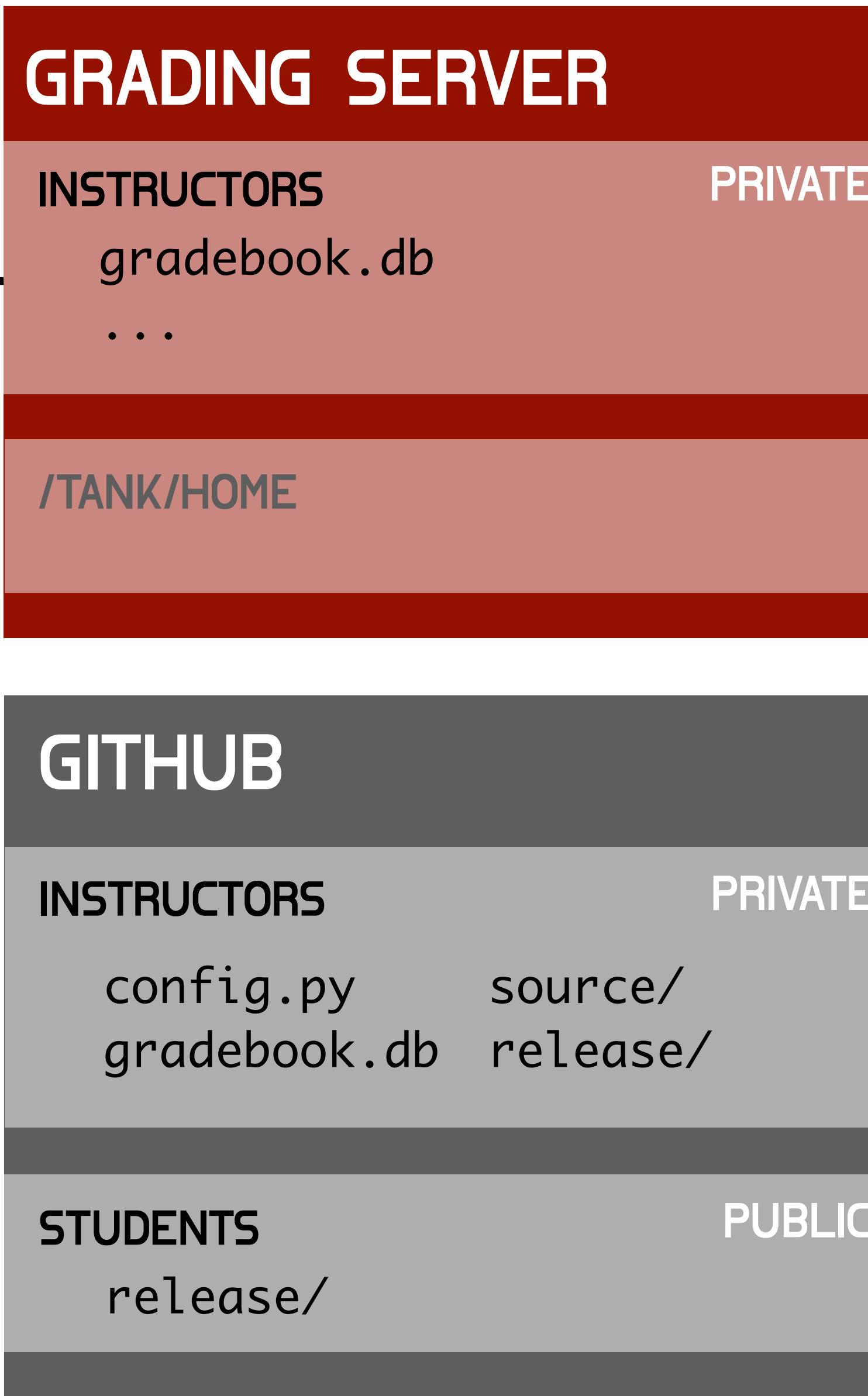
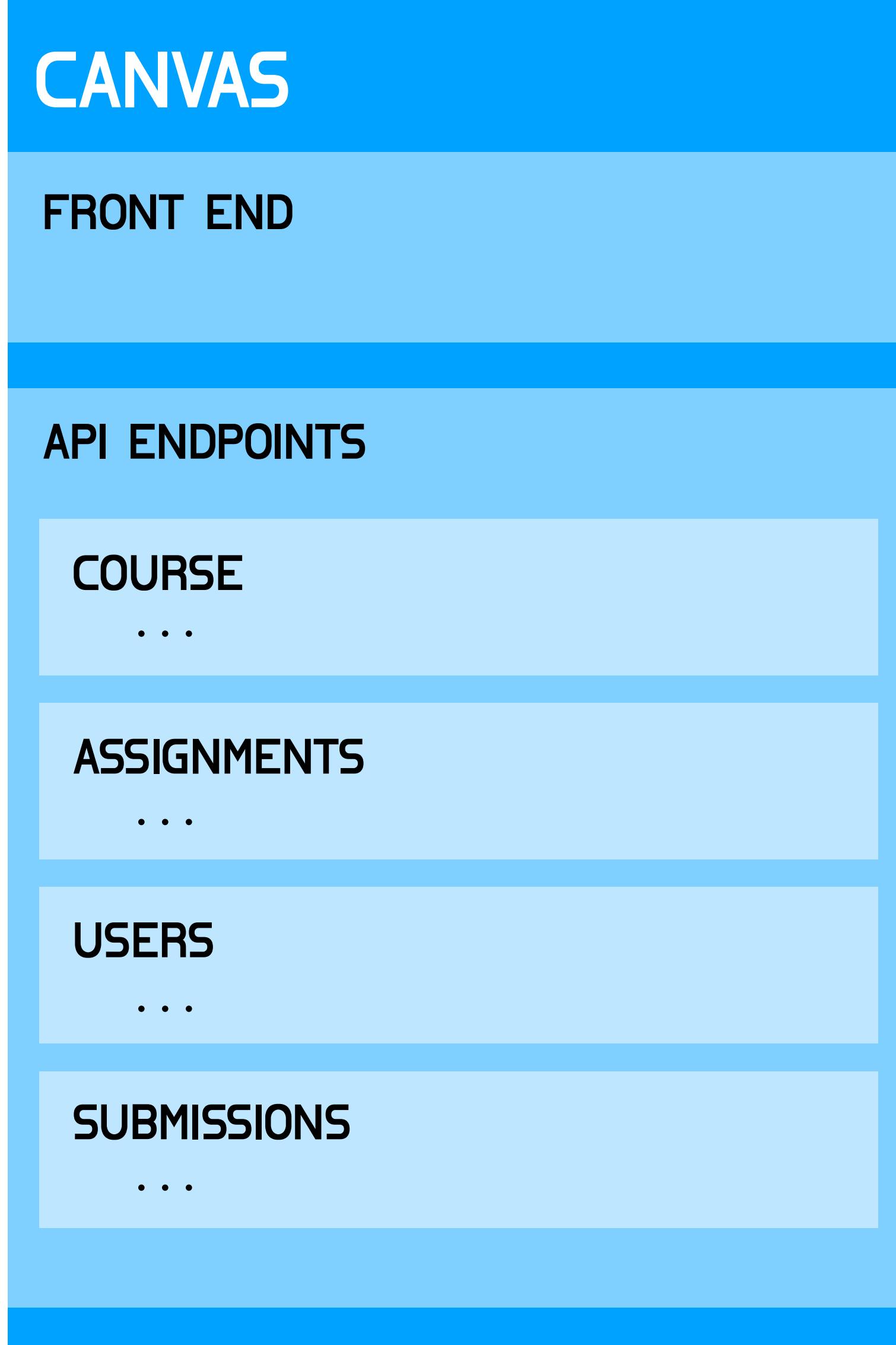


# Overview



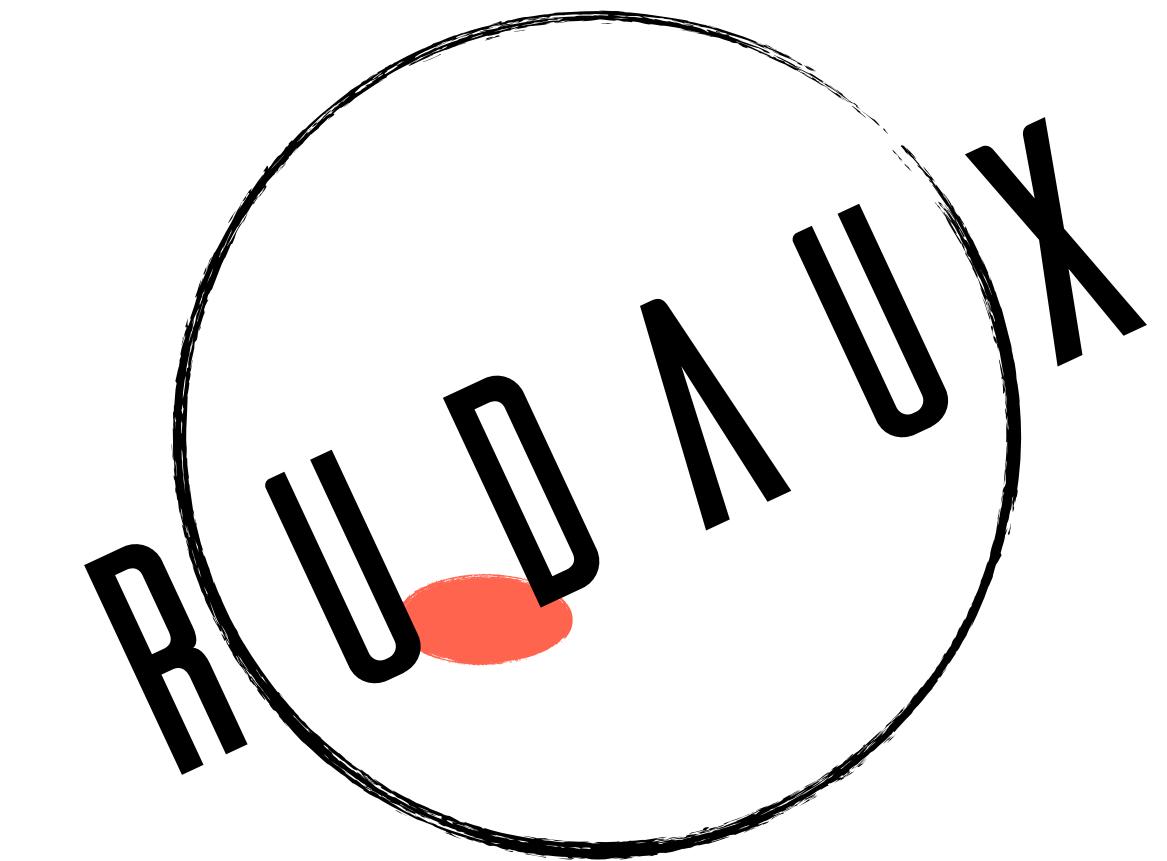
JUPYTERHUB

# Overview



# What Ties Everything Together?

- `rudaux` is a Python module which provides the framework for managing a course with Canvas and JupyterHub
- `ltiauthenticator` is a JupyterHub authenticator that receives Canvas launch requests and grabs Canvas ID
- `nbgitpuller` is a Jupyter plugin that allows unidirectional git sync and redirection
- The *Canvas API* enables bidirectional data transfer between JupyterHub and Canvas



UBC Canvas homework\_1 Sam

Secure | https://ubc.test.instructure.com/courses/5394/assignments/192600

DSCI100\_sb > Assignments > homework\_1

# homework\_1

This tool needs to be loaded in a new browser window

Load homework\_1 in a new window

Clicking this...

- Authenticates student by Canvas ID
- Pulls `homework_1.ipynb` to JupyterHub
- Opens sandboxed Jupyter notebook (`homework_1.ipynb`) in new window

Home  
Announcements  
**Assignments**  
Discussions  
Grades  
People  
Pages  
Files  
Syllabus  
Outcomes  
Quizzes  
Modules  
Chat  
CoursEval  
Piazza  
My Media  
Media Gallery

(ltiauthenticator)

(nbgitpuller)

(nbgitpuller &  
dockerspawner)

06-classification

Not Secure | <https://hub-prod-dsci.stat.ubc.ca/jupyter/user/student/notebooks/worksheets/worksheets/06-classification.ipynb>

jupyter 06-classification Last Checkpoint: a day ago (autosaved)

Logout Control Panel

File Edit View Insert Kernel Help

Not Trusted R O

Markdown

## Chapter 6 Lecture Worksheet - Classification

### Learning Objectives

- Recognize situations where a simple classifier would be appropriate for making predictions.
- Explain the k-nearest neighbour classification algorithm.
- Interpret the output of a classifier.
- Compute, by hand, the distance between points when there are two attributes.
- Describe what a training data set is and how it is used in classification.
- In a dataset with two attributes, perform k-nearest neighbour classification in R using `caret::train(method = "knn", ...)` to predict the class of a single new observation.

```
In [21]: library(repr)      # options() to change the plot sizes  
library(tidyverse)
```

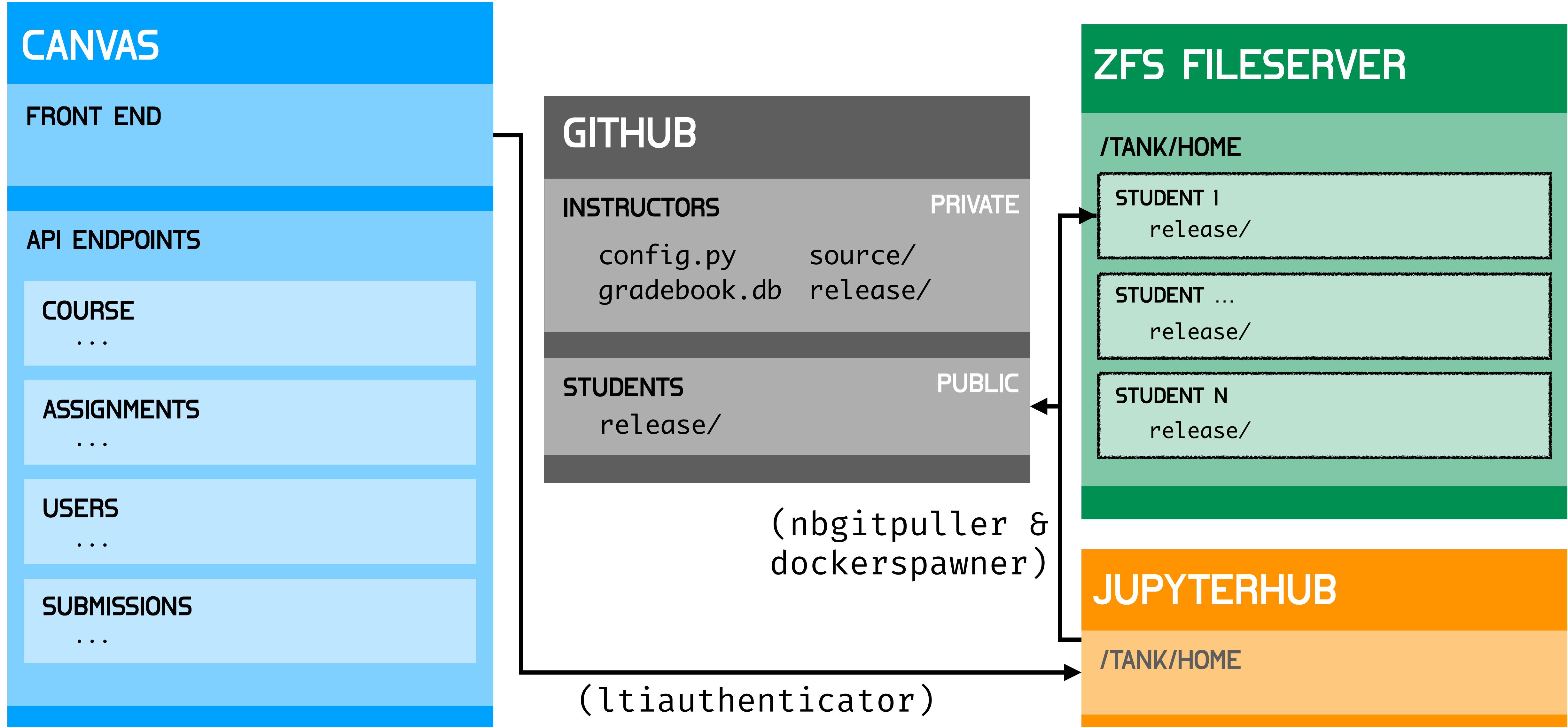
### Part 1 - Breast cancer dataset

We will work with the breast cancer data from this week's pre-reading. Load the appropriate packages and the "clean-wdbc-data.csv" dataset into the notebook.

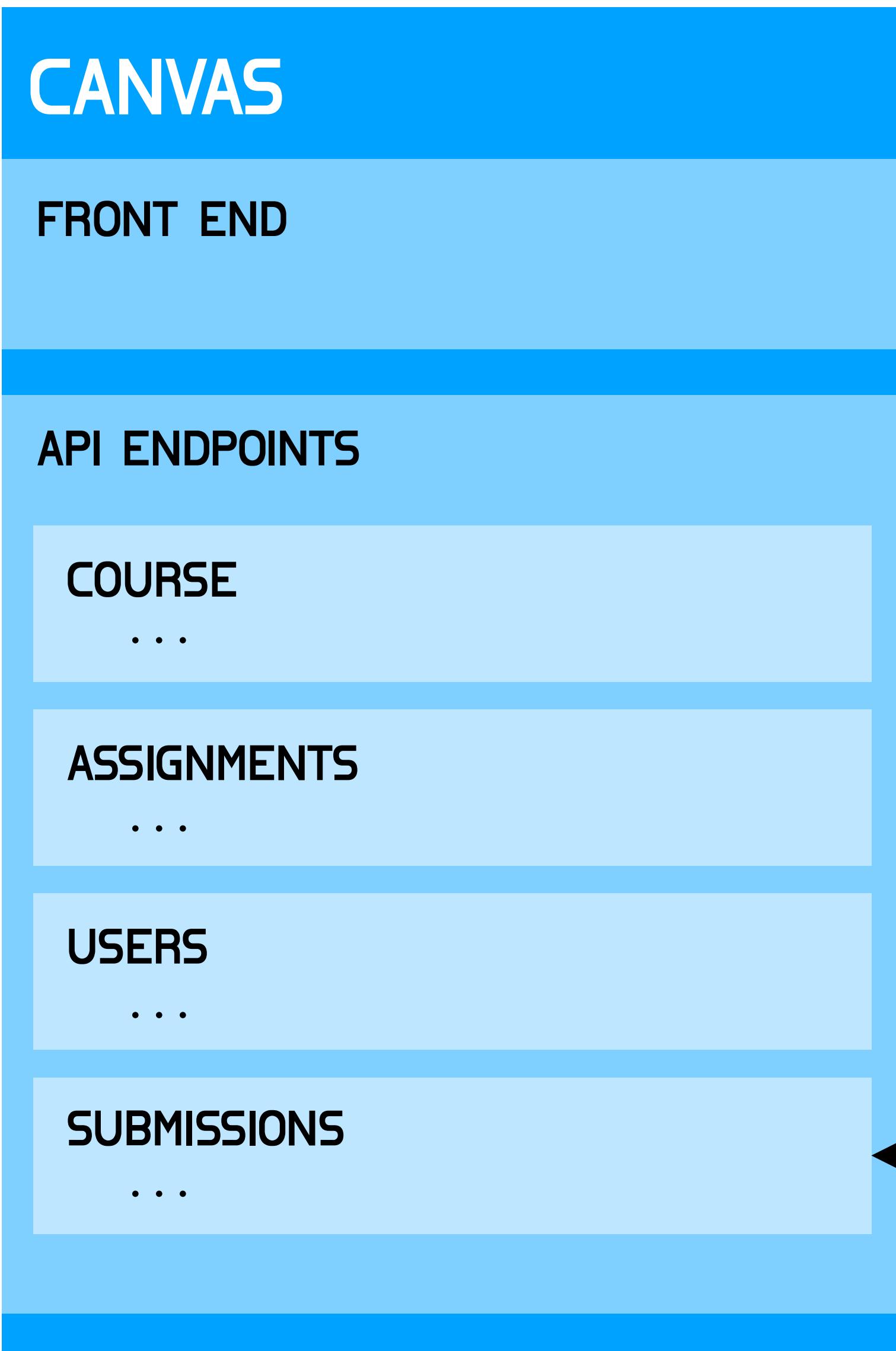
```
In [22]: bcDat <- read_csv("data/clean-wdbc-data.csv")
```

```
Parsed with column specification:  
cols(  
  ID = col_integer(),
```

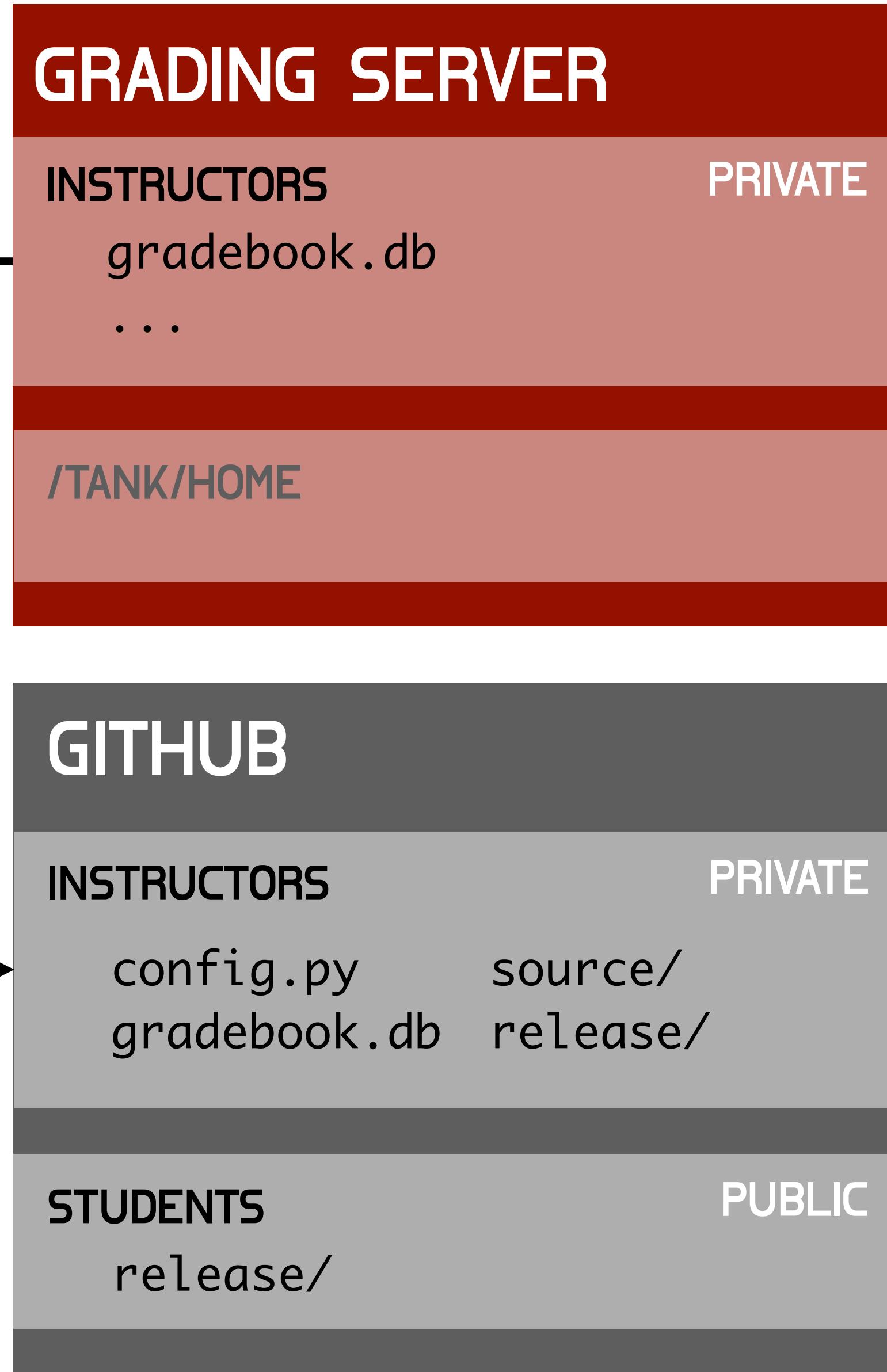
# User authentication



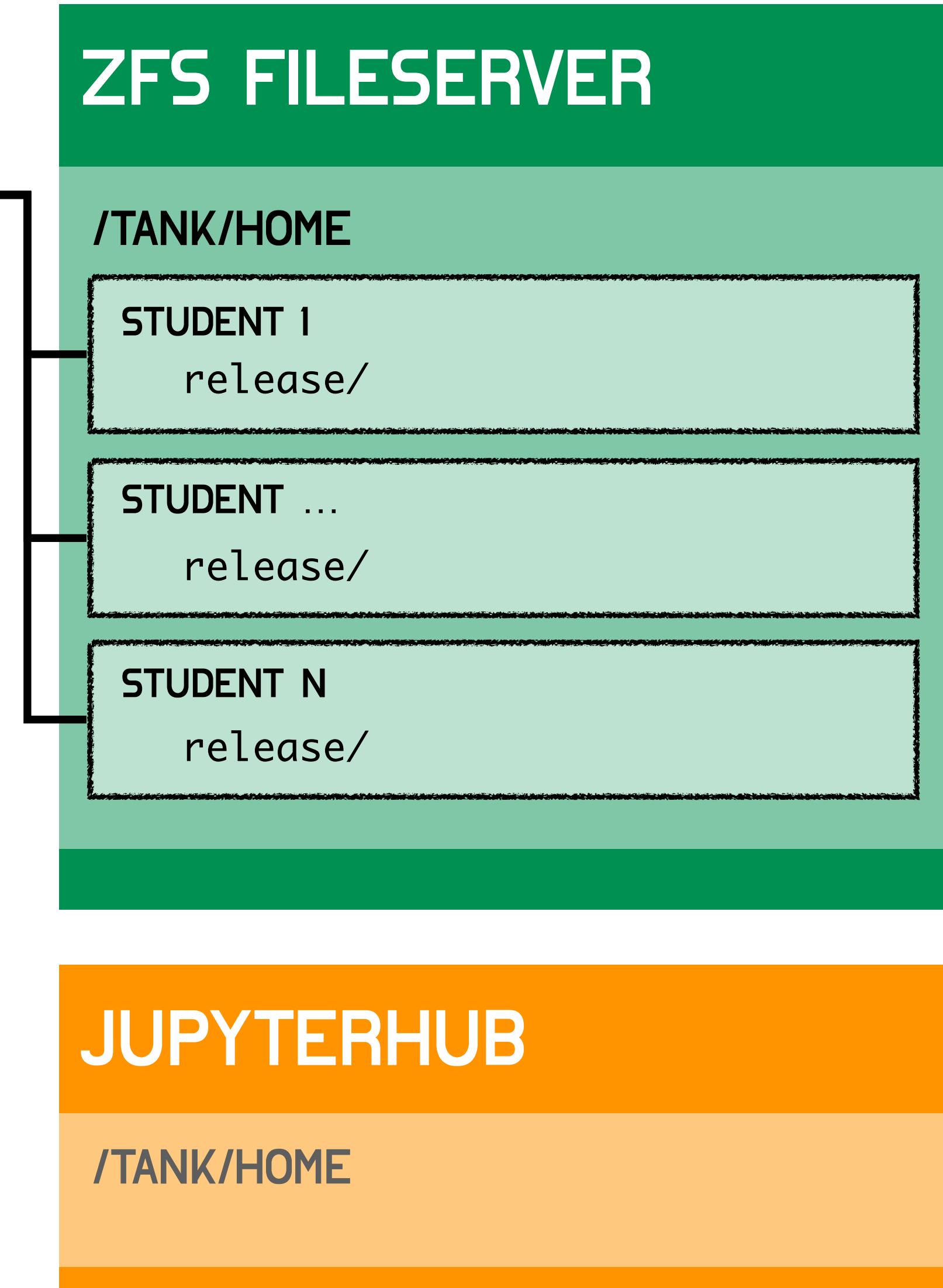
# autograding



cron job initiates nbgrader  
grading at close time



ZFS filesystem  
snapshots at close time



# Features

- Students' progress is saved
- Students' notebooks are sandboxed with Docker
- Students can only access their own work
- JupyterHub access is **restricted to LTI launch requests**
- Student work is automatically gathered for submission at the assignment close date

# Student Goals

- Students only ever need to visit **Canvas**.
- JupyterHub provides a sandboxed environment with all dependencies preinstalled.
- Learning data science with Jupyter provides skills which can be **directly** applied to future endeavors.

# Instructor Goals

- Instructors can configure entire course from **one file**.
- Canvas assignments are automatically created for the instructor that point to corresponding assignment URLs.
- Grading can be **100% automated**, or incorporate feedback from TAs.



THANKS TO

Tiffany Timbers  
Ian Allison  
Mike Gelbart  
The Ha  
Calvin Leung  
Yuvi Panda

# rudaux\_config.py

```
c.Canvas.course_id = 5394
c.Canvas.canvas_url = 'https://ubc.test.instructure.com'

c.GitHub.ins_repo_url = 'git@github.ubc.ca:hinshaws/DSCI_100_instructors.git'
c.GitHub.stu_repo_url = 'git@github.com:samhinshaw/dsci-100.git'

c.JupyterHub.hub_url = 'https://hub-prod-dsci.stat.ubc.ca'

c.Course.timezone = 'US/Pacific'
```

Note: this can be incorporated into nbgrader\_config.py if you wish to have only one config file

# rudaux\_config.py

```
c.Course.assignments = [  
    {  
        "name": "week_1",  
        "duedate": "2018-08-14",  
        "duetime": "23:59:59",  
        "points": 2,  
        "manual": True  
    },  
    {  
        "name": "homework_1",  
        "duedate": "2018-08-15",  
        "duetime": "11:00:00",  
        "points": 5  
    }  
]
```

Note: this can be incorporated into nbgrader\_config.py if you wish to have only one config file