# Intro to Coding with Python– Intro to Python

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

# Plan for Today

- Variables
- Merge conflicts
- Finish ico1

# Variables

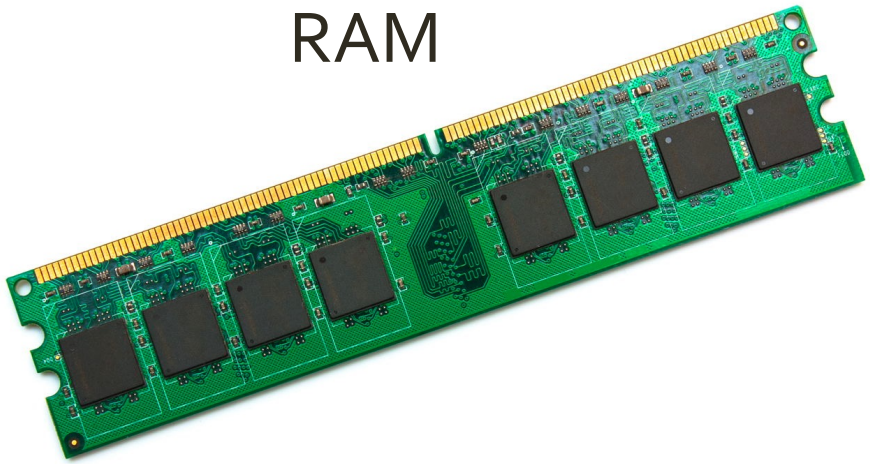
Storage

Want to store this important information for later: 3

# Storage

Want to store this important information for later: 3

RAM

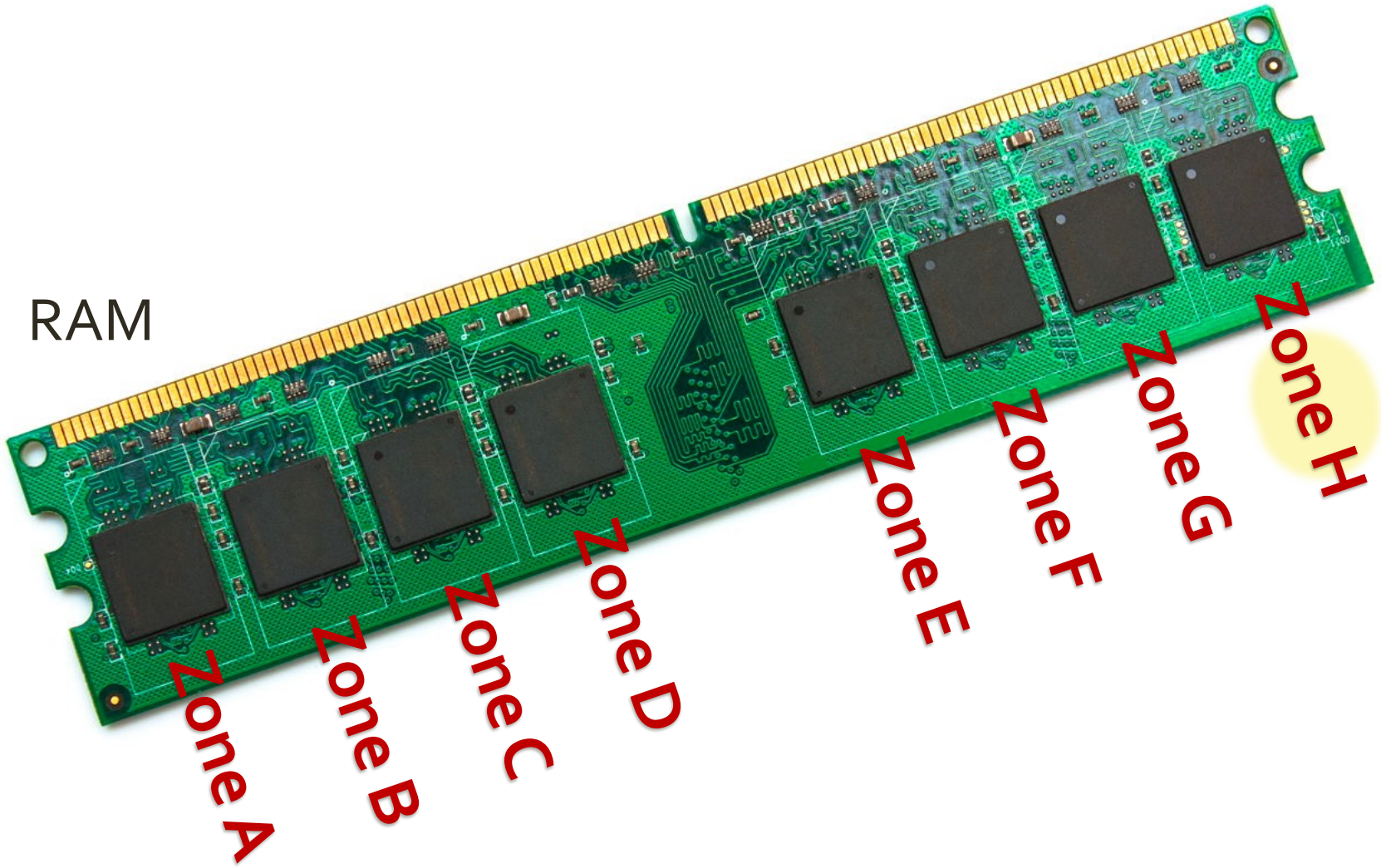


Hard Disk

# Storage

Want to store this important information for later: 3

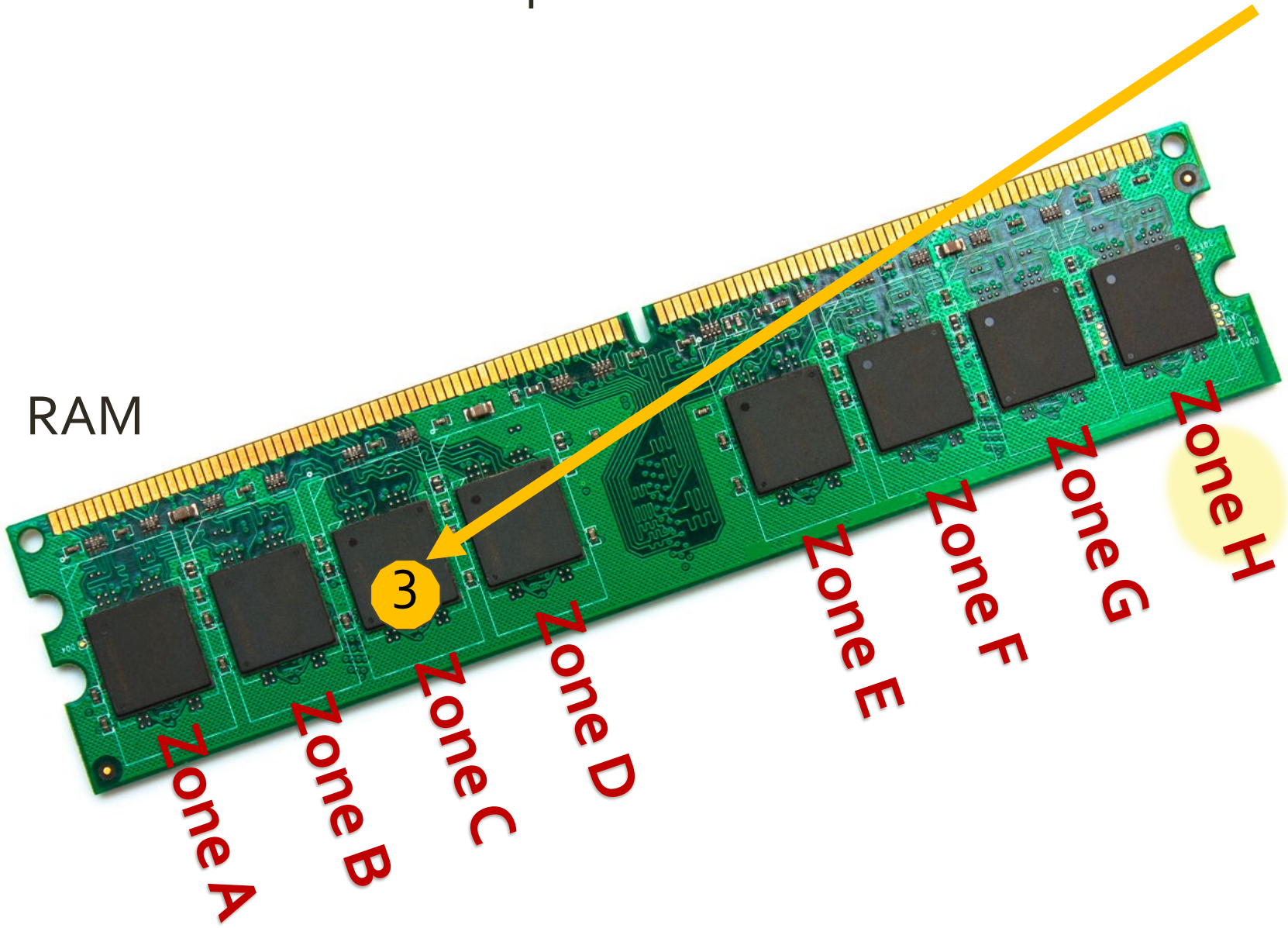
RAM



# Storage

Want to store this important  information for later:

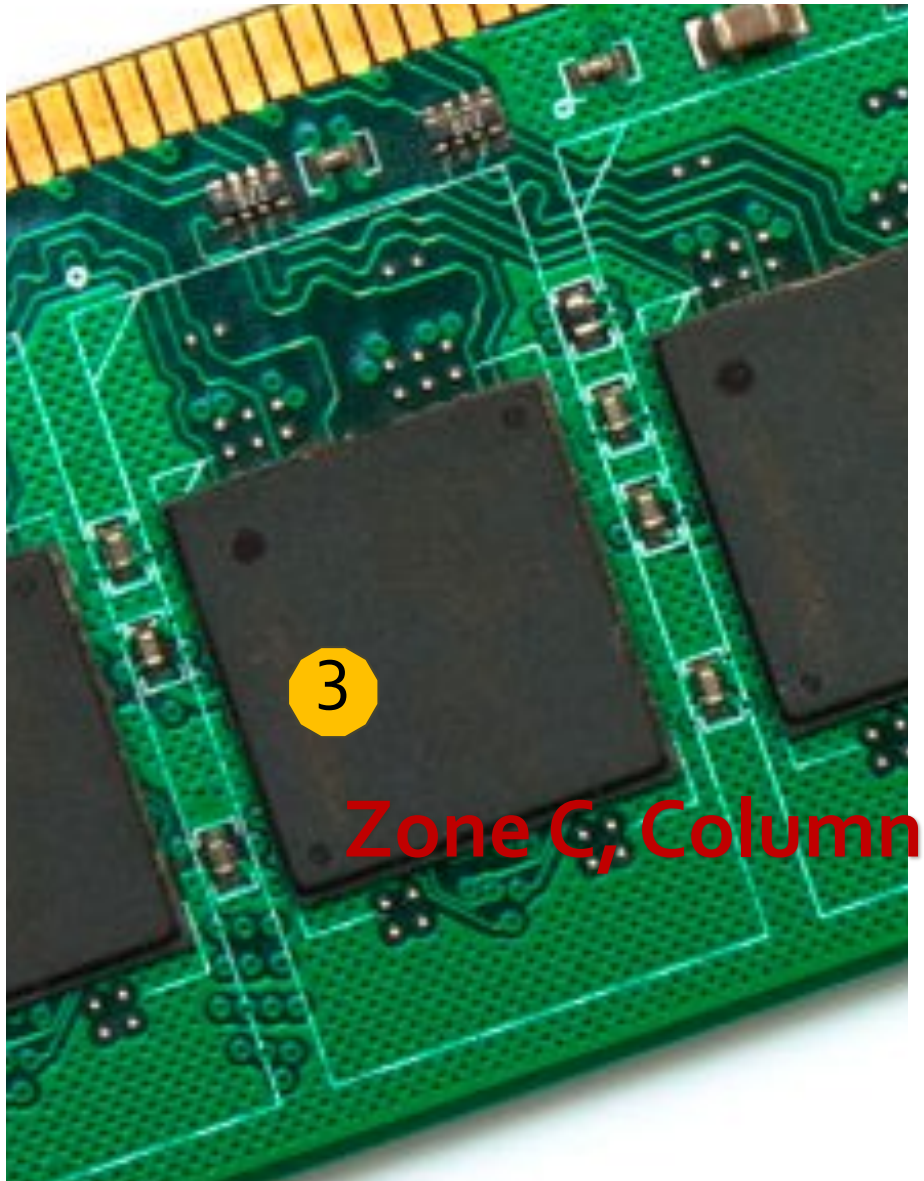
RAM



# Storage

Want to store this important  information for later:

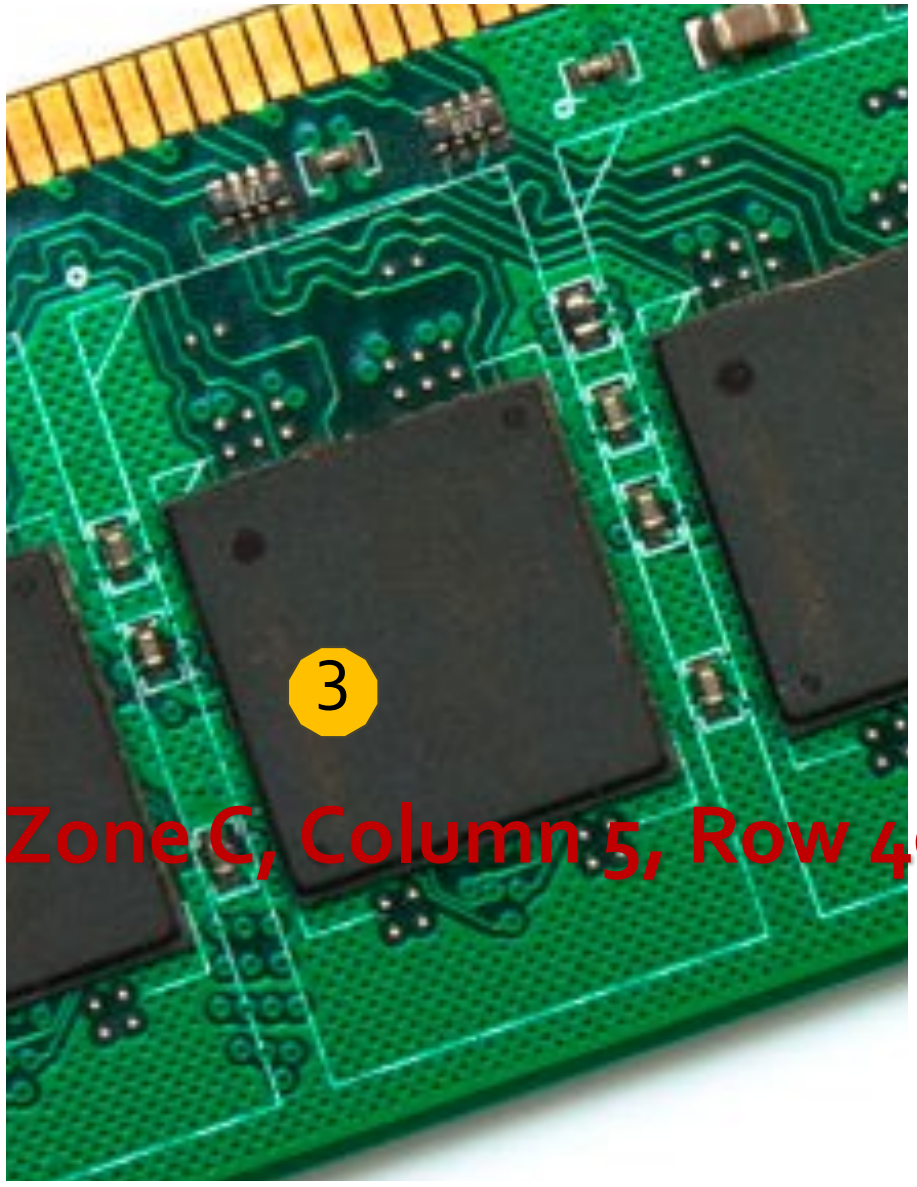
RAM




Zone C, Column 5, Row 40

# Storage

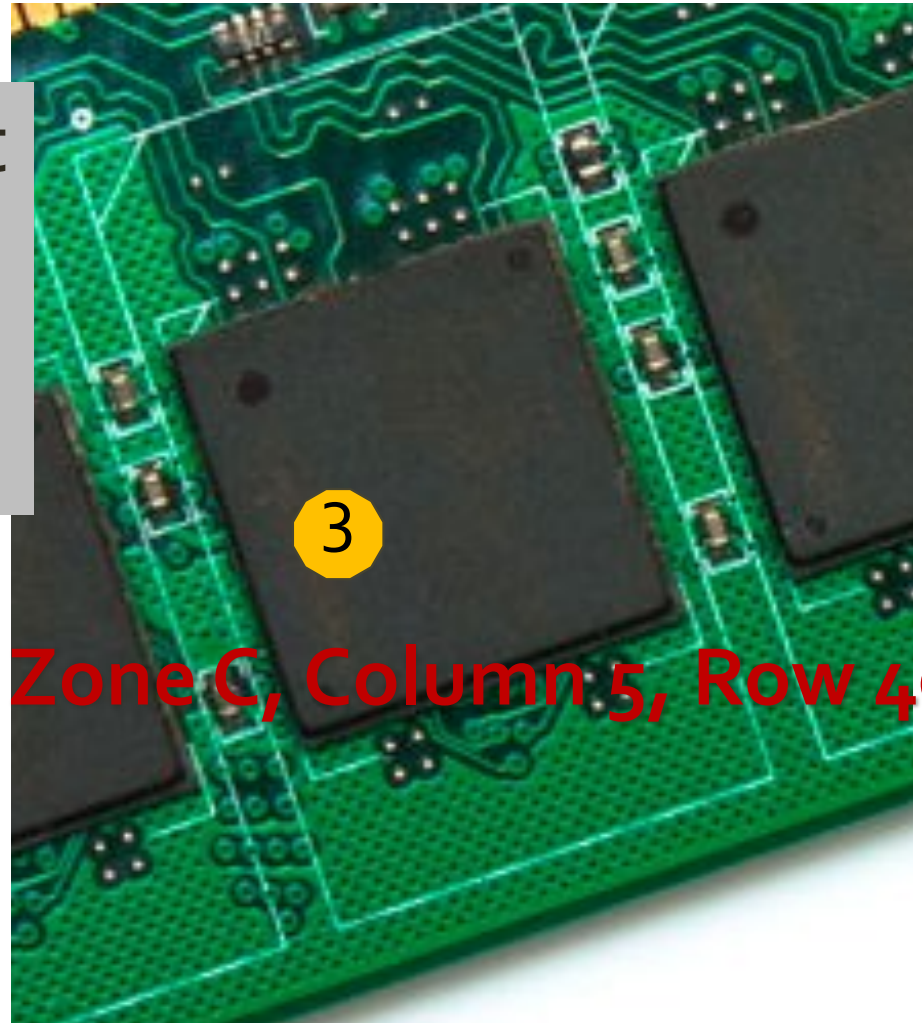

- Want the CPU to double the important information



# Storage


- Want the CPU to double the important information
- We need to tell it where to look

```
double “important information” at “Zone C, Column 5, Row 40”
```

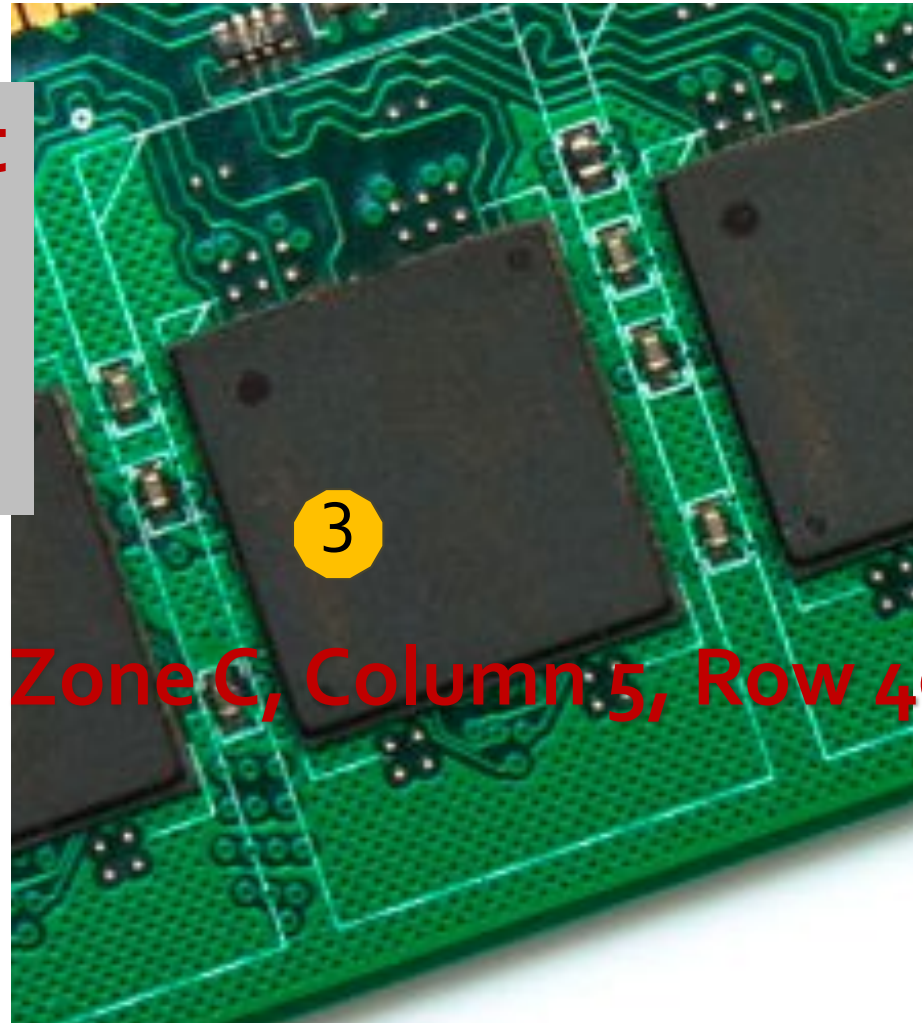


# Storage


- Want the CPU to double the important information
- We need to tell it where to look

double “**important information**” at “**Zone C, Column 5, Row 40**”

- Wordy
- Refer to the same thing

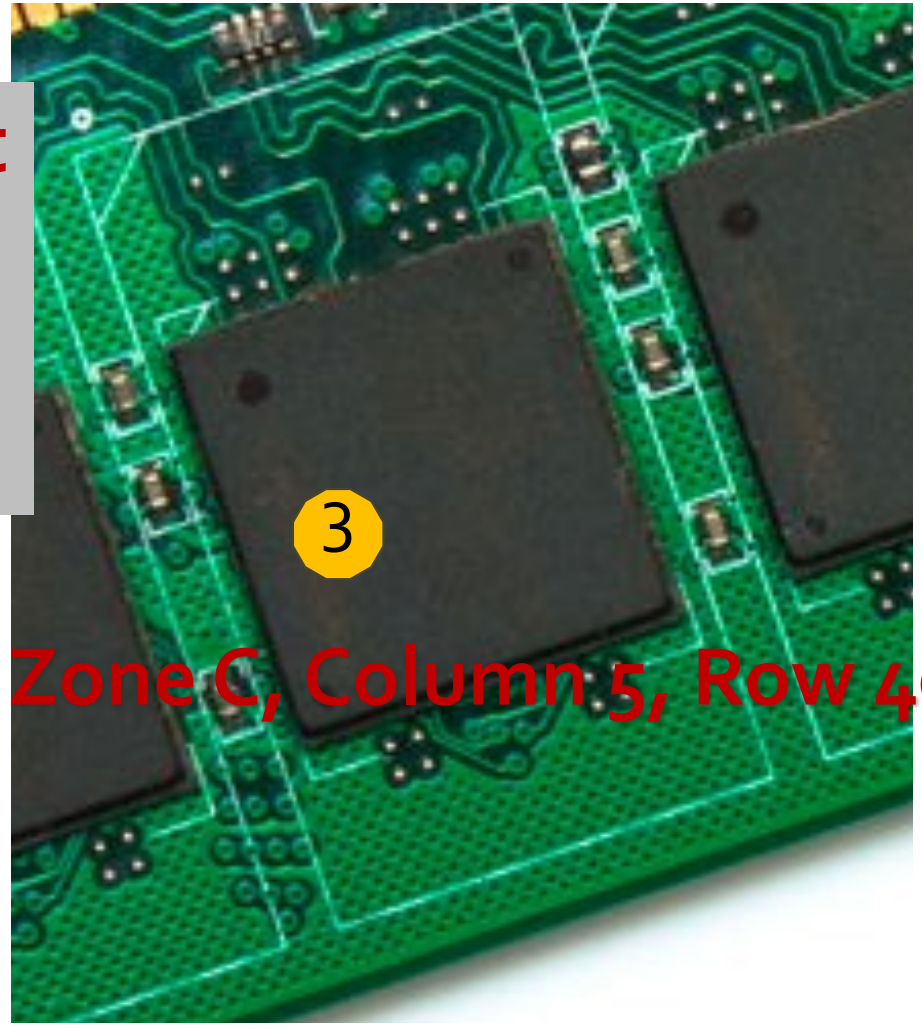


# Storage


- Want the CPU to double the important information
- We need to tell it where to look

double “**important information**” at “**Zone C, Column 5, Row 40**”

- Wordy
- Refer to the same thing
- Let’s use a shorthand



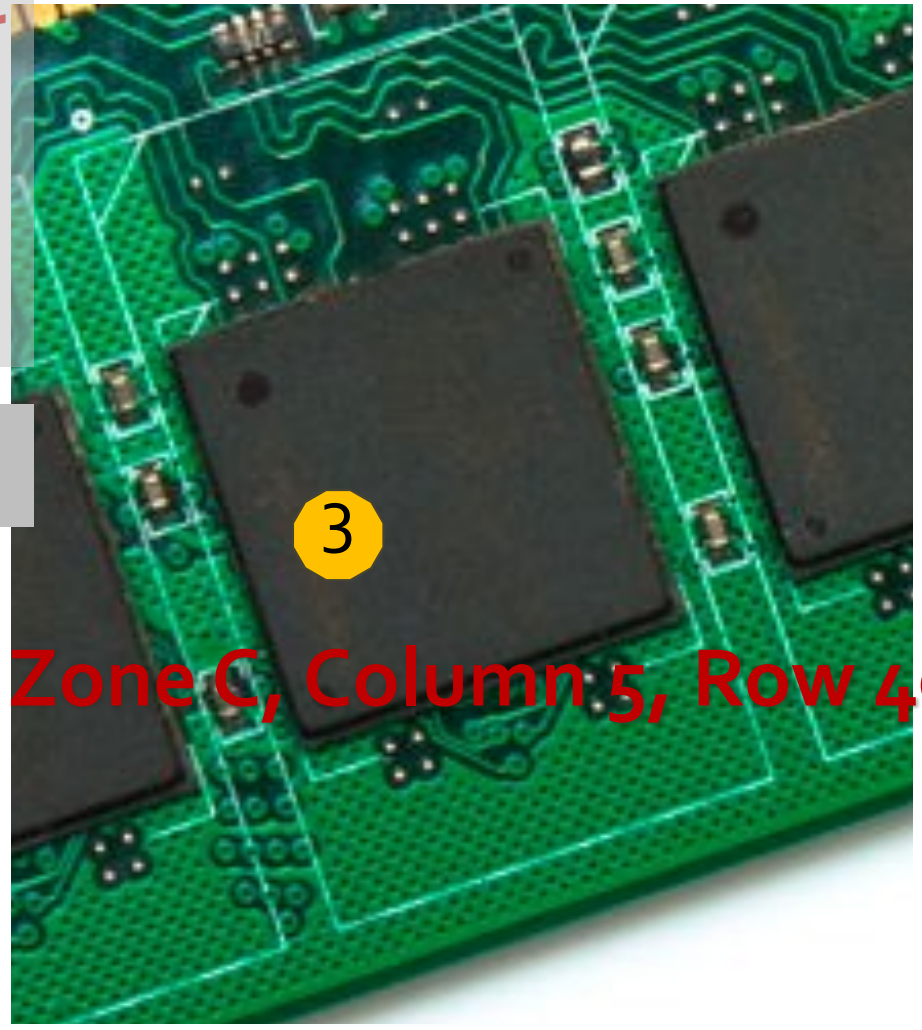
# Storage


- Want the CPU to double the important information
- We need to tell it where to look

```
double "important information" at "Zone C, Column 5, Row 40"
```

```
double x
```

- Wordy
- Refer to the same thing
- Let's use a shorthand, **x**



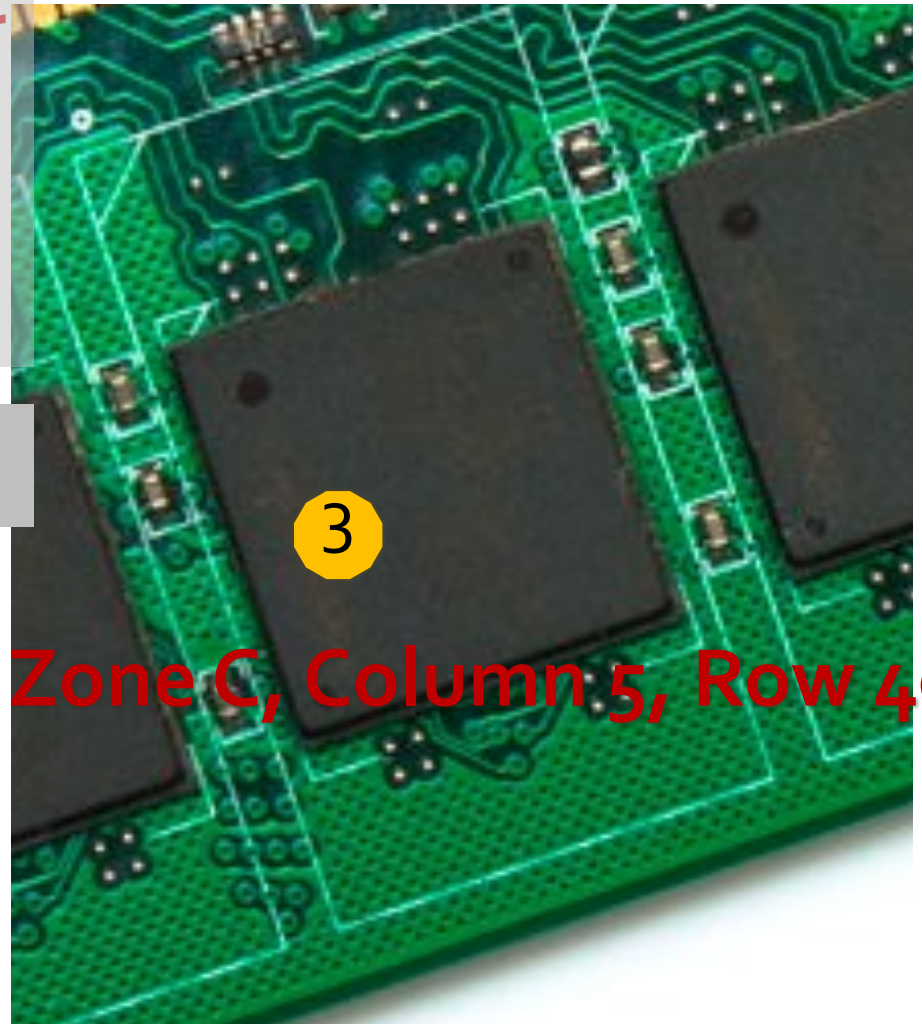
# Storage


- Want the CPU to double the important information
- We need to tell it where to look

```
double "important information" at "Zone C, Column 5, Row 40"
```

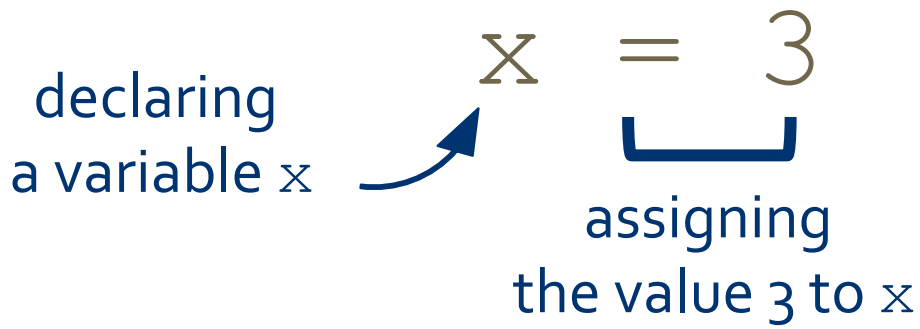
```
double x
```

- Wordy
- Refer to the same thing
- Let's use a **variable**



# Core concept: `variables`

- In CS, a **`variable`** is a place to store a piece of data
- In Python, variables are:
  - **declared** by giving them a name
  - **assigned** using the equals sign
- Example:



# Core concept: numeric values

- Two kinds of **numbers** in CS:
  - integers ("whole numbers")
  - floats ("decimals" or "floating point numbers")
- In Python, the kind of number is implied by whether or not the number contains a **decimal point**
- Example:

```
x = 3
x = 3.0
```

# Core concept: strings

- In CS, a sequence of characters that isn't a number is called a **`string`**
- In Python, a string is declared using **quotation marks**
- Strings can contain letters, numbers, spaces, and special characters
- Example:

```
x = "Ab"
x = "Bass Hall"
```

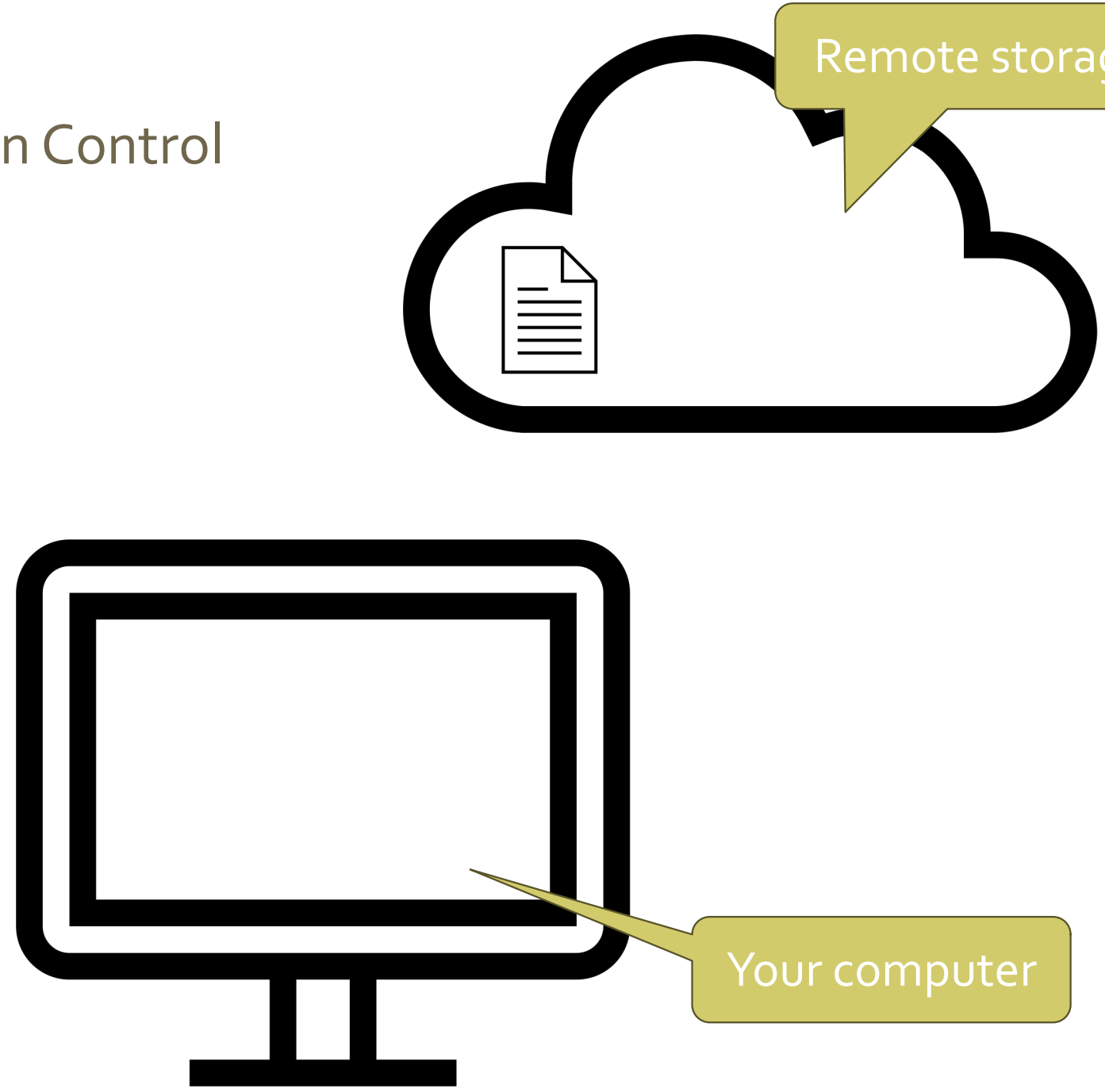
# Core concept: `print()`

- A function is a procedure / routine that takes in some input and does something with it (just like in math)
- In Python, the **`print()`** function takes in a value and outputs the value to the console
- This seems silly now, but will come in handy in lab when you write/run your first program inside a **file**

# Merge Conflicts

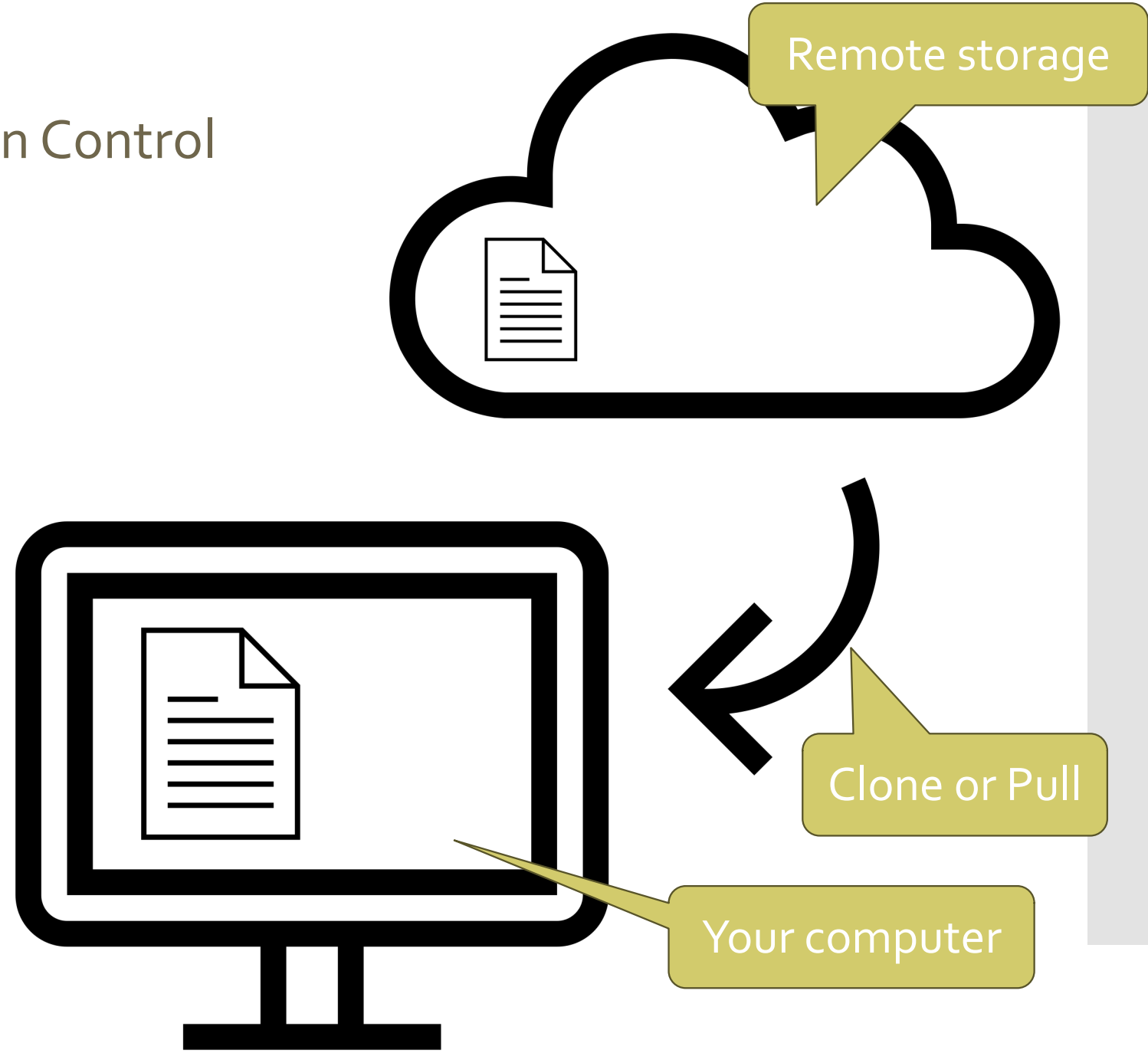
# Benefits to GitHub

- Version Control



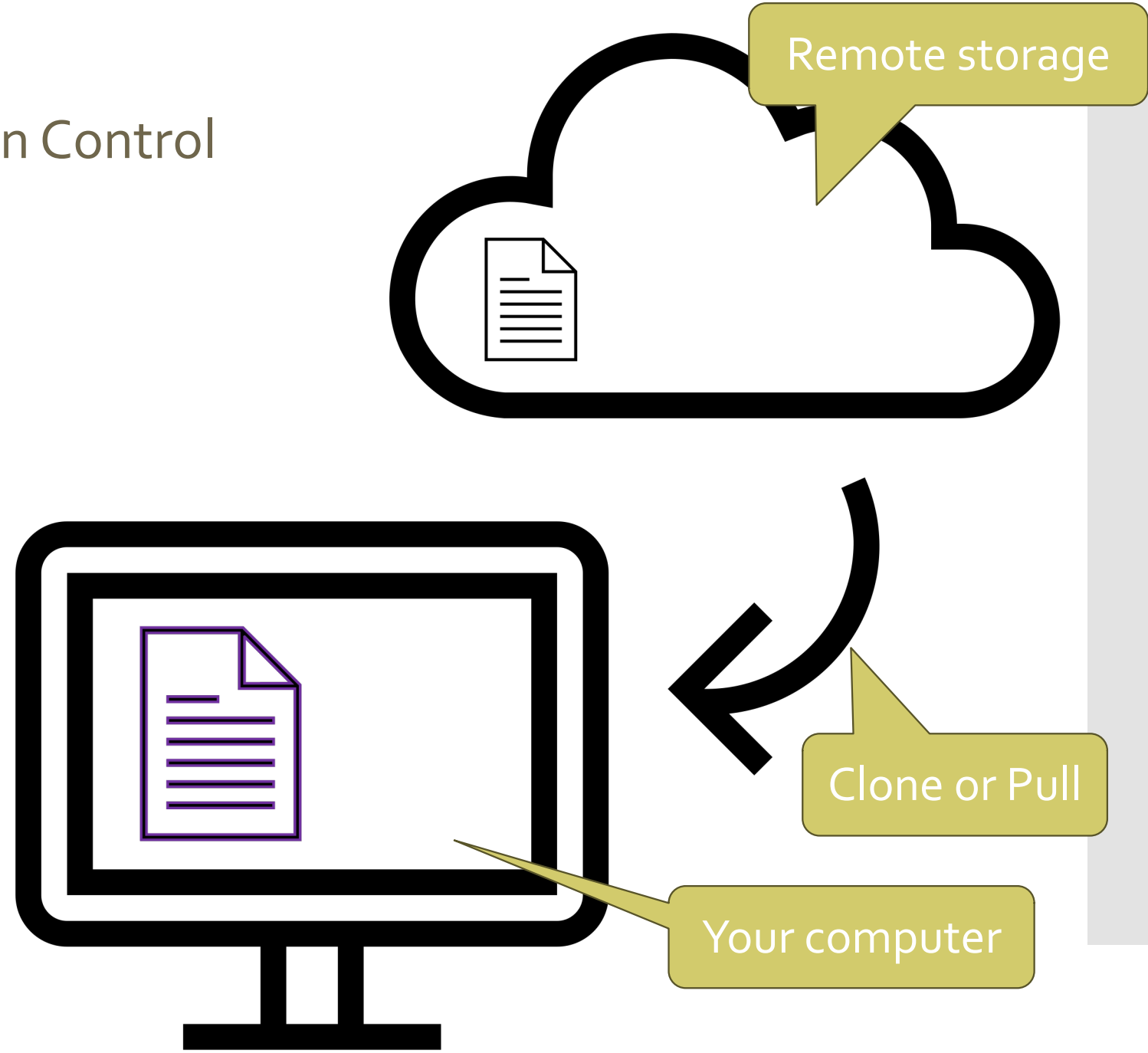
# Benefits to GitHub

- Version Control



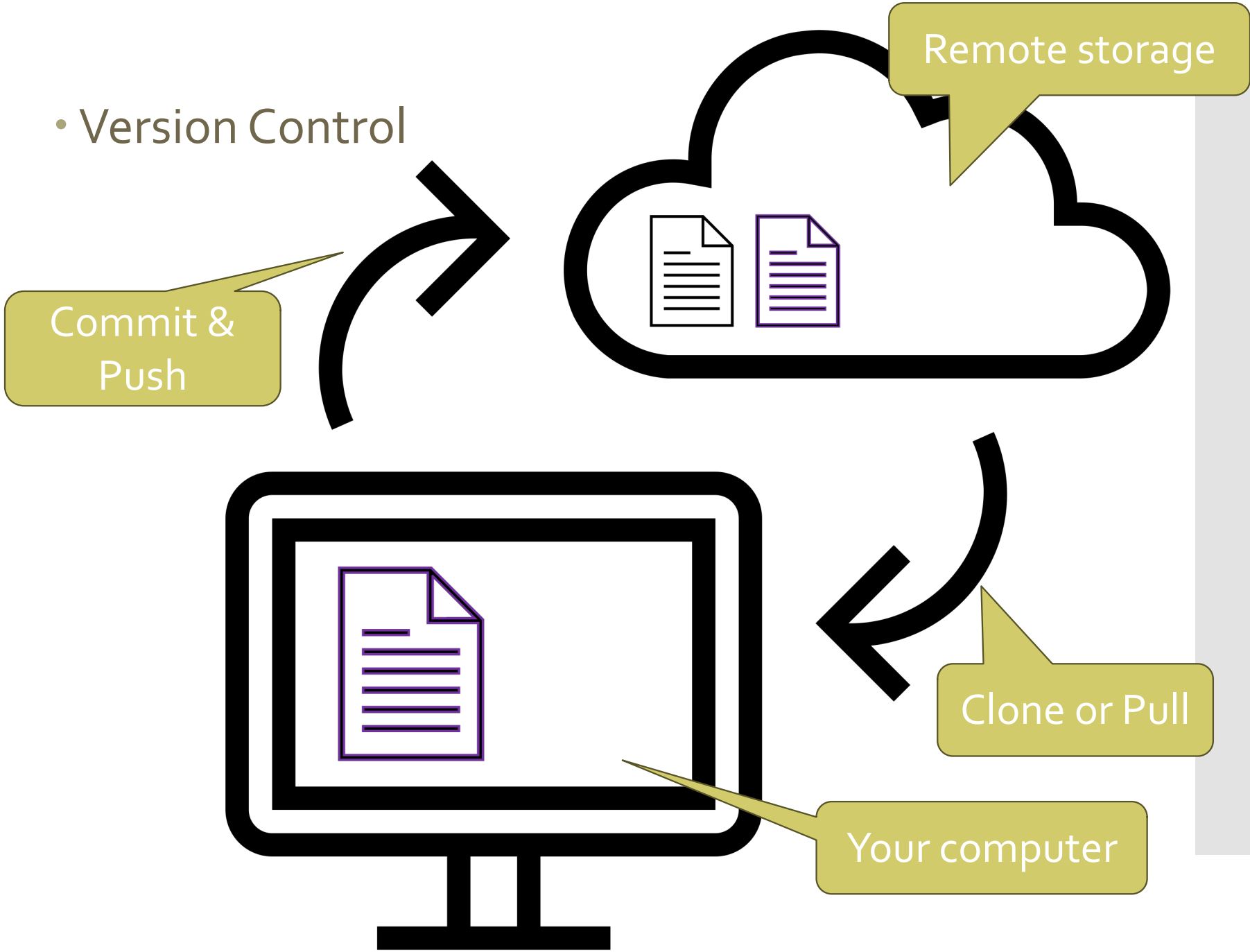
# Benefits to GitHub

- Version Control



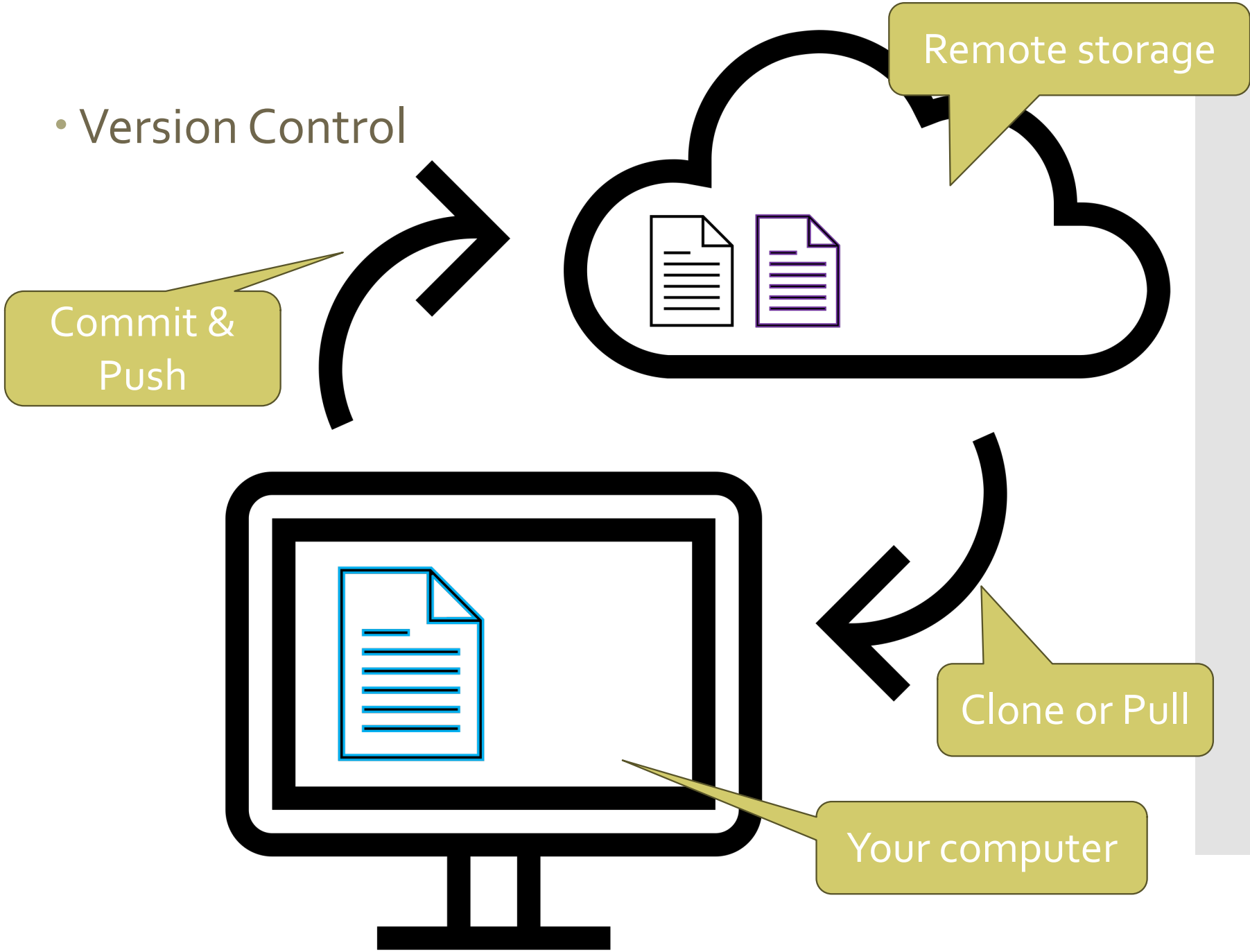
# Benefits to GitHub

- Version Control



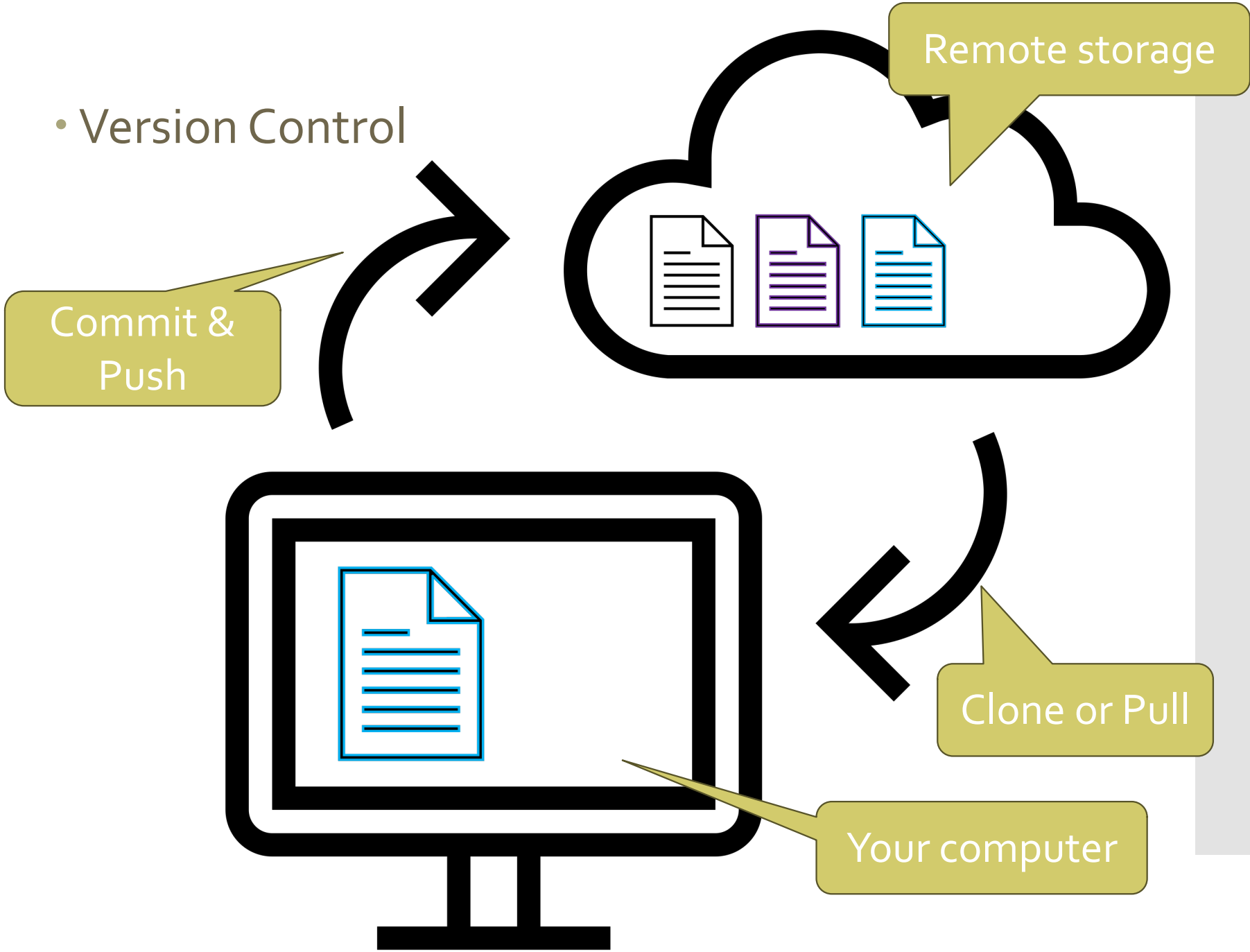
# Benefits to GitHub

- Version Control



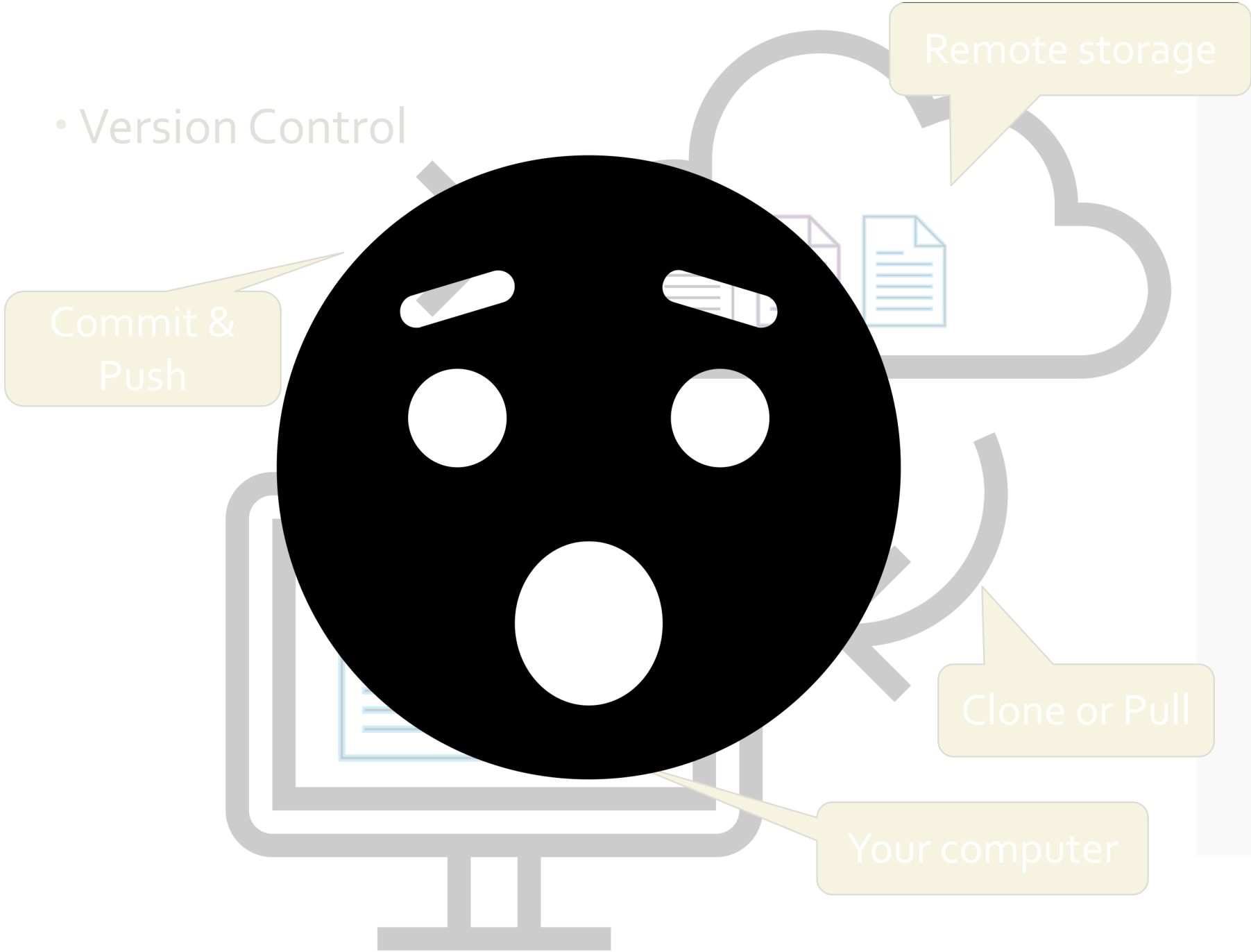
# Benefits to GitHub

- Version Control



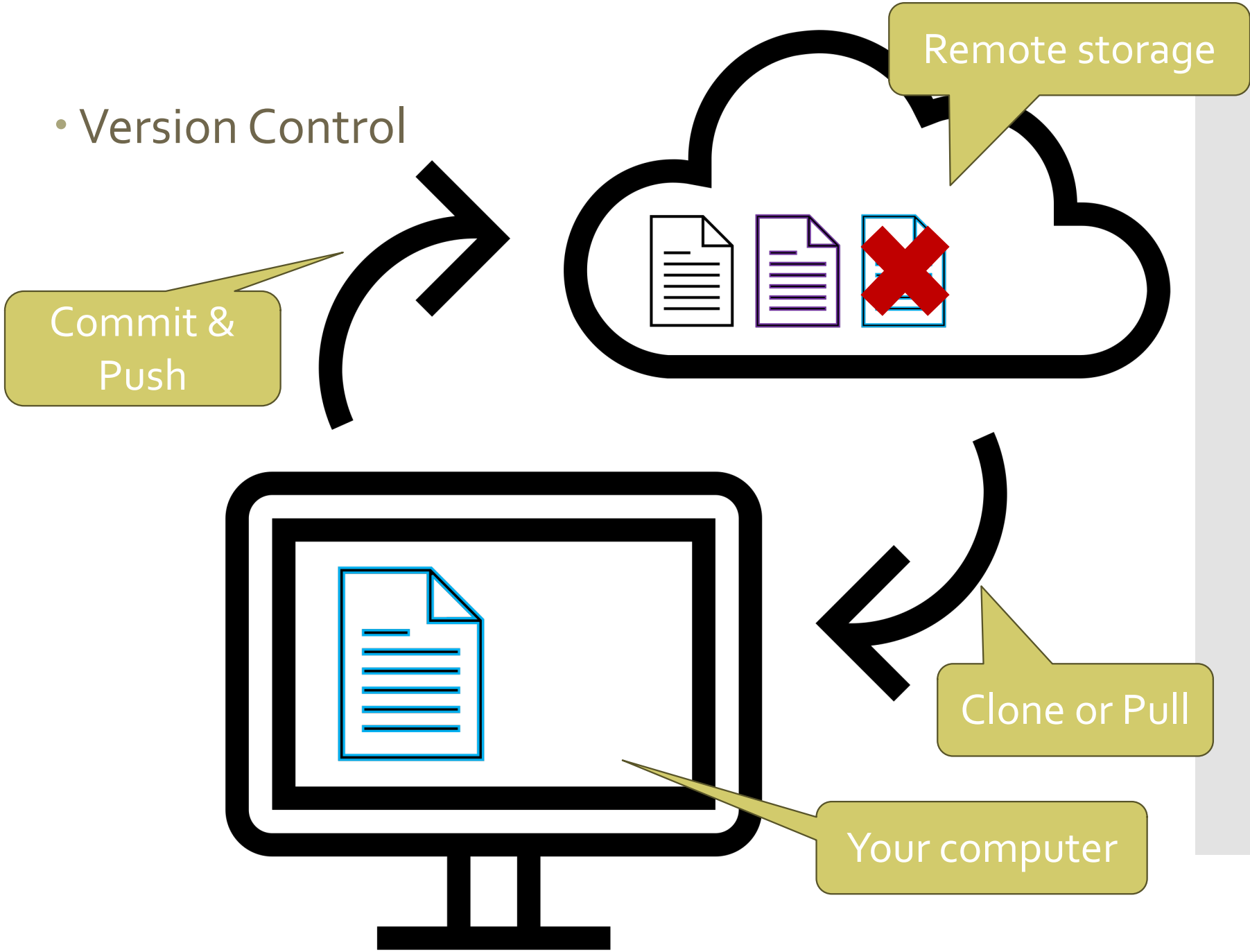
# Benefits to GitHub

- Version Control



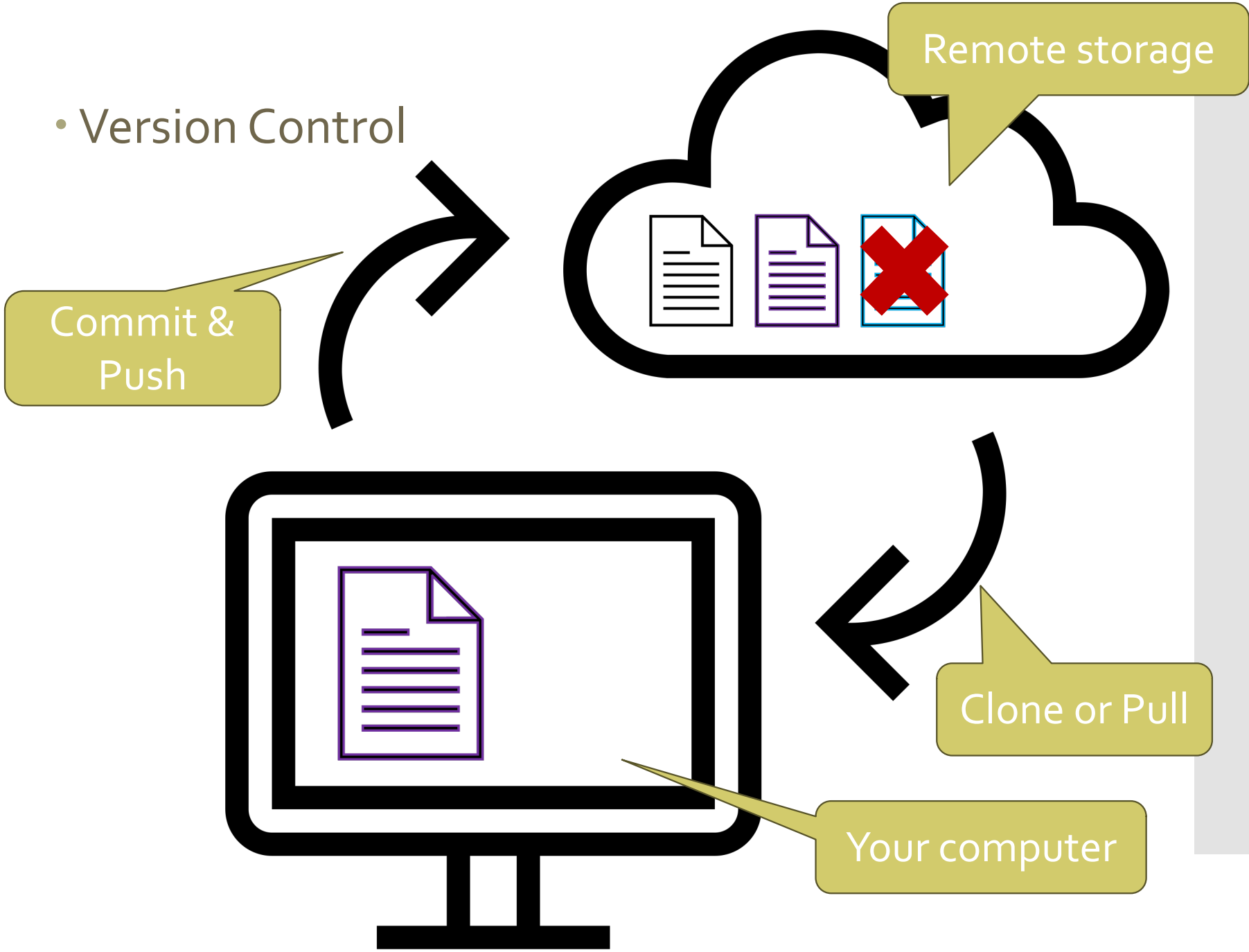
# Benefits to GitHub

- Version Control



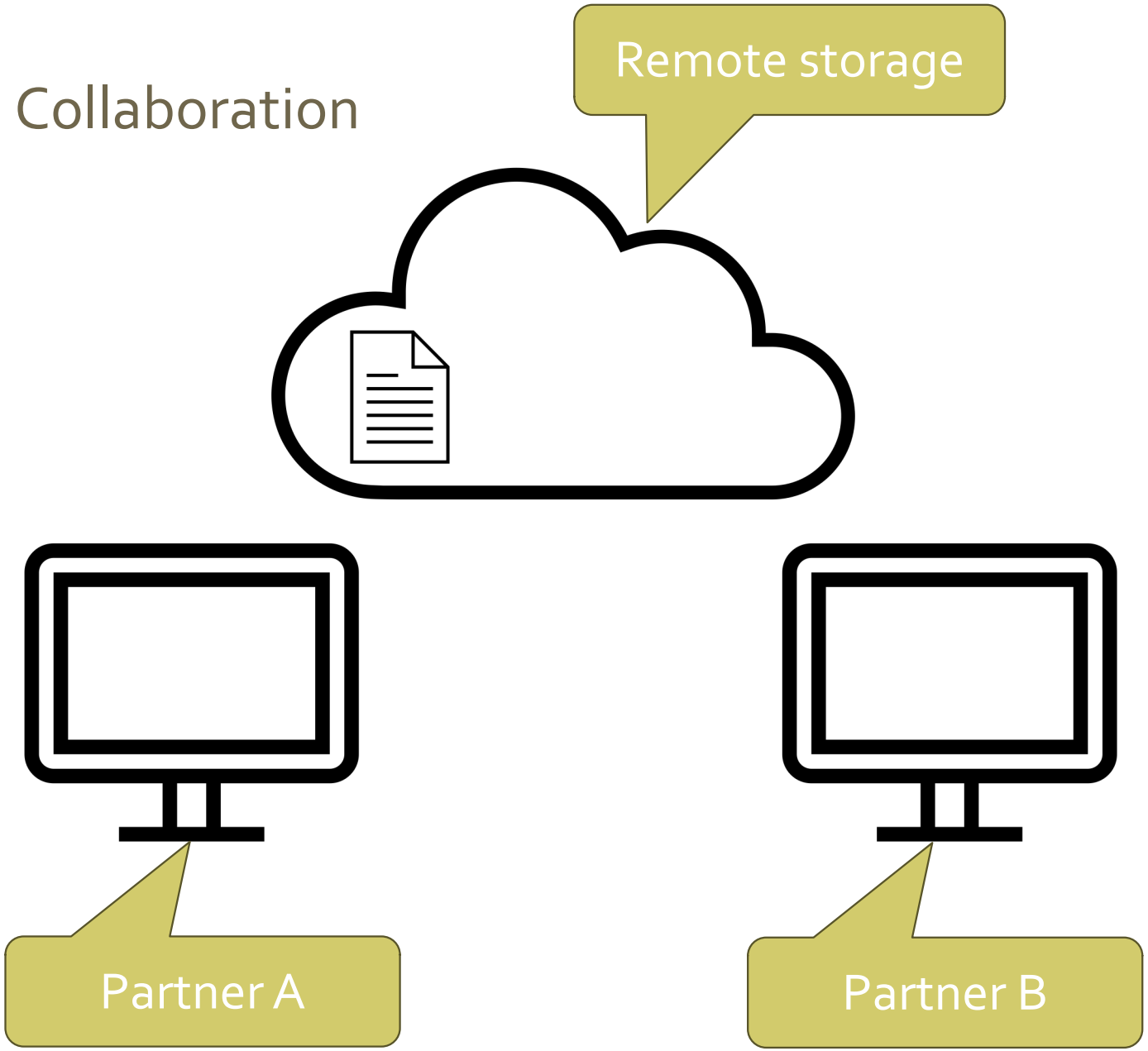
# Benefits to GitHub

- Version Control



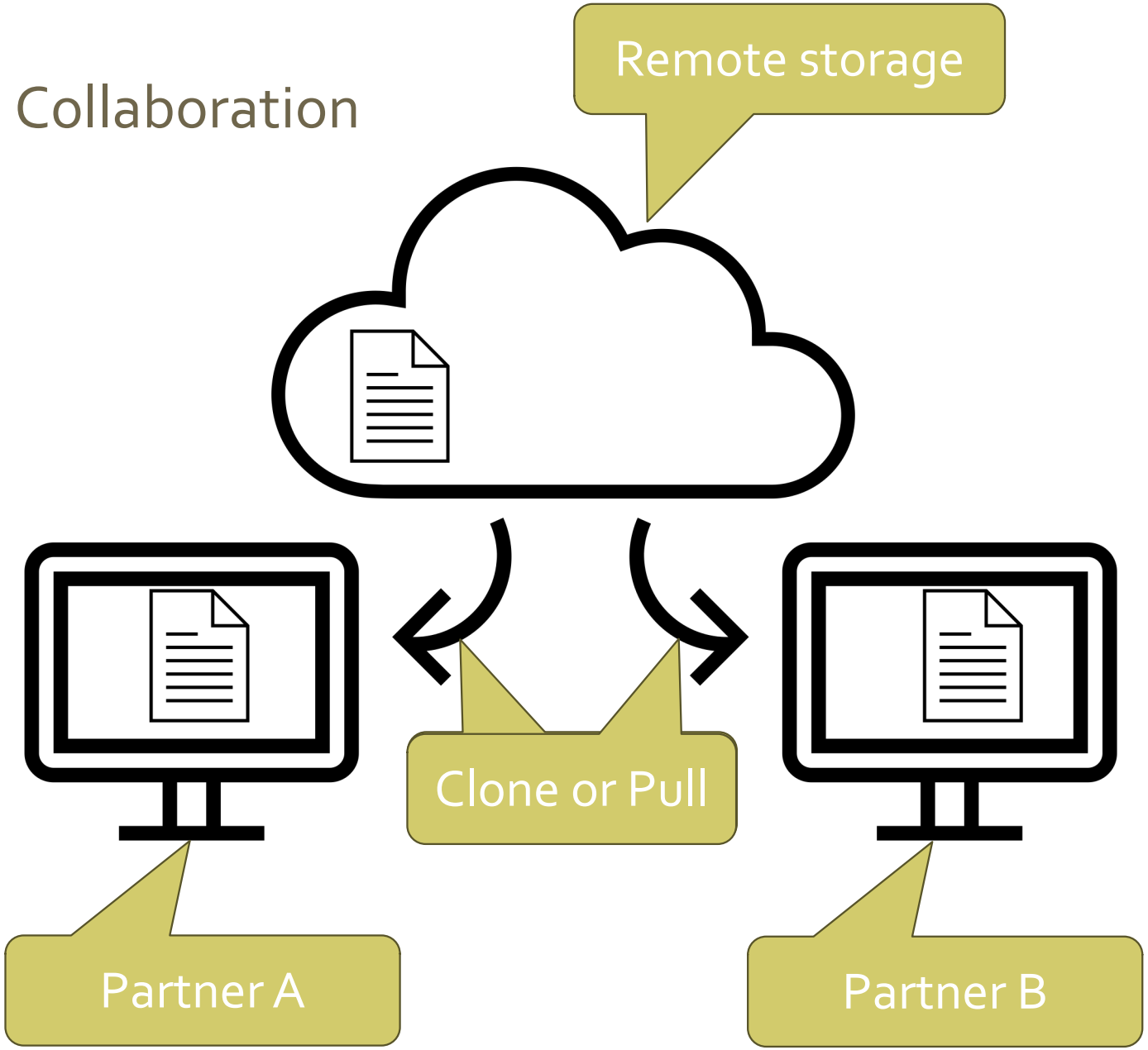
# Benefits to GitHub

- Collaboration



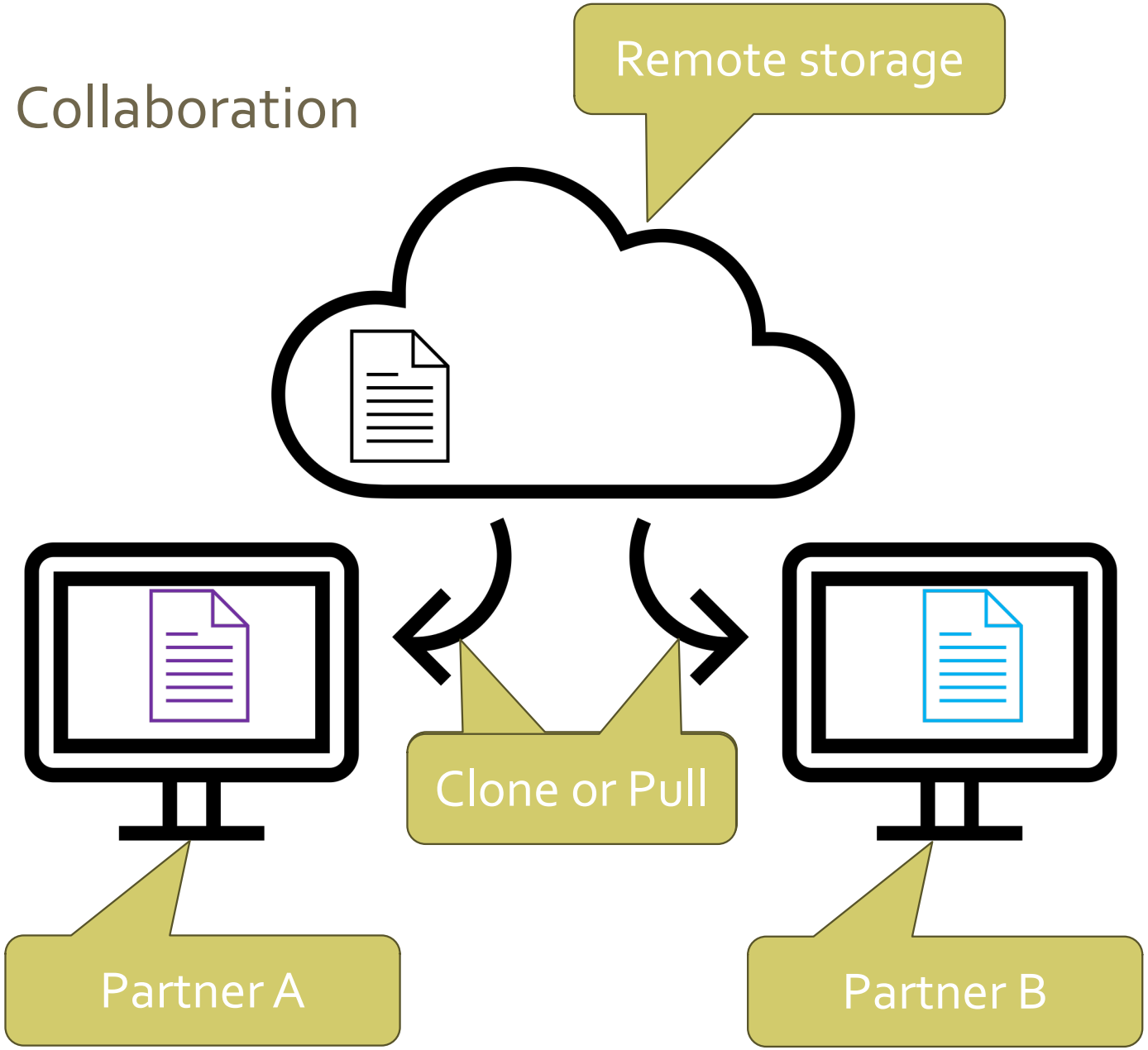
# Benefits to GitHub

- Collaboration



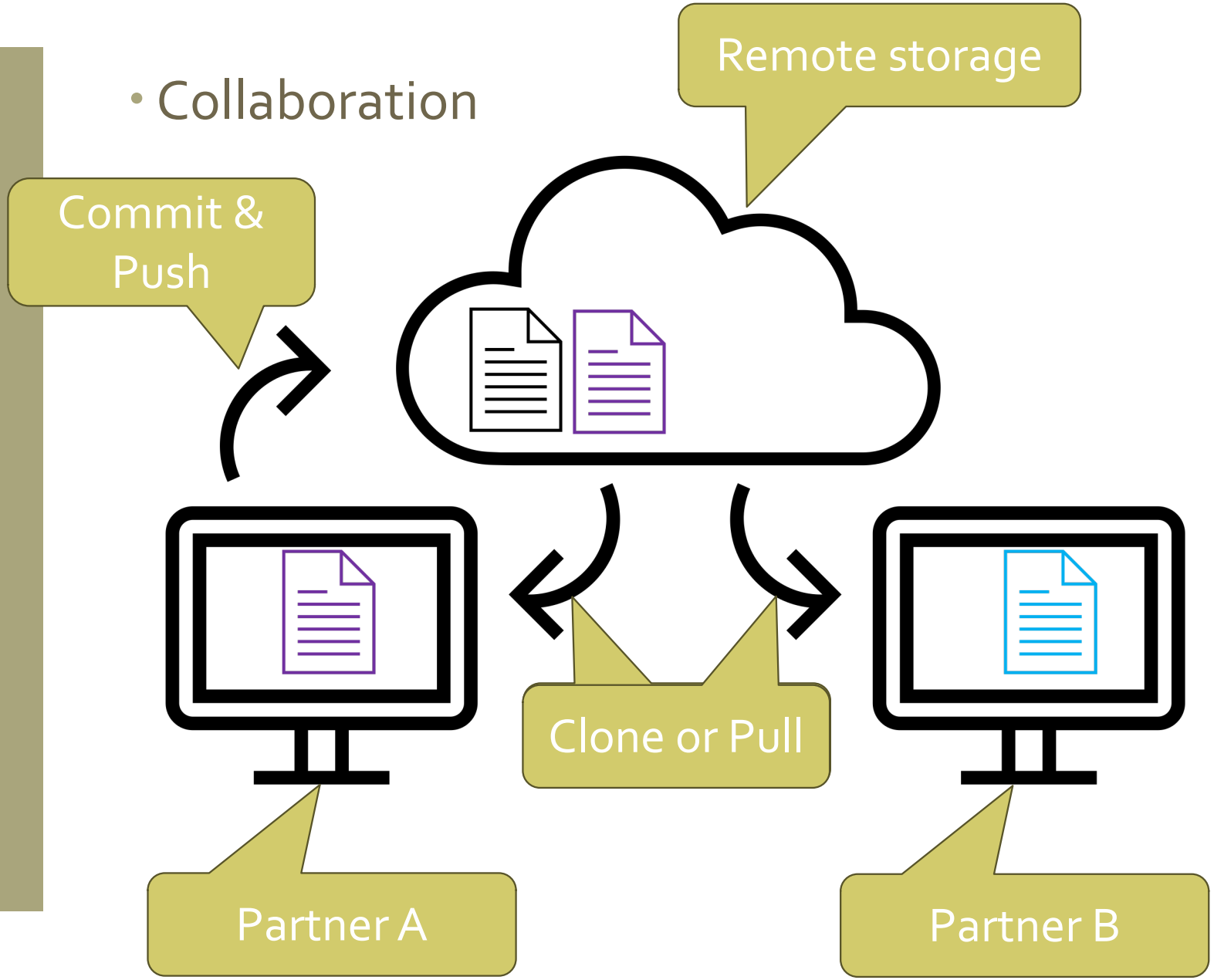
# Benefits to GitHub

- Collaboration



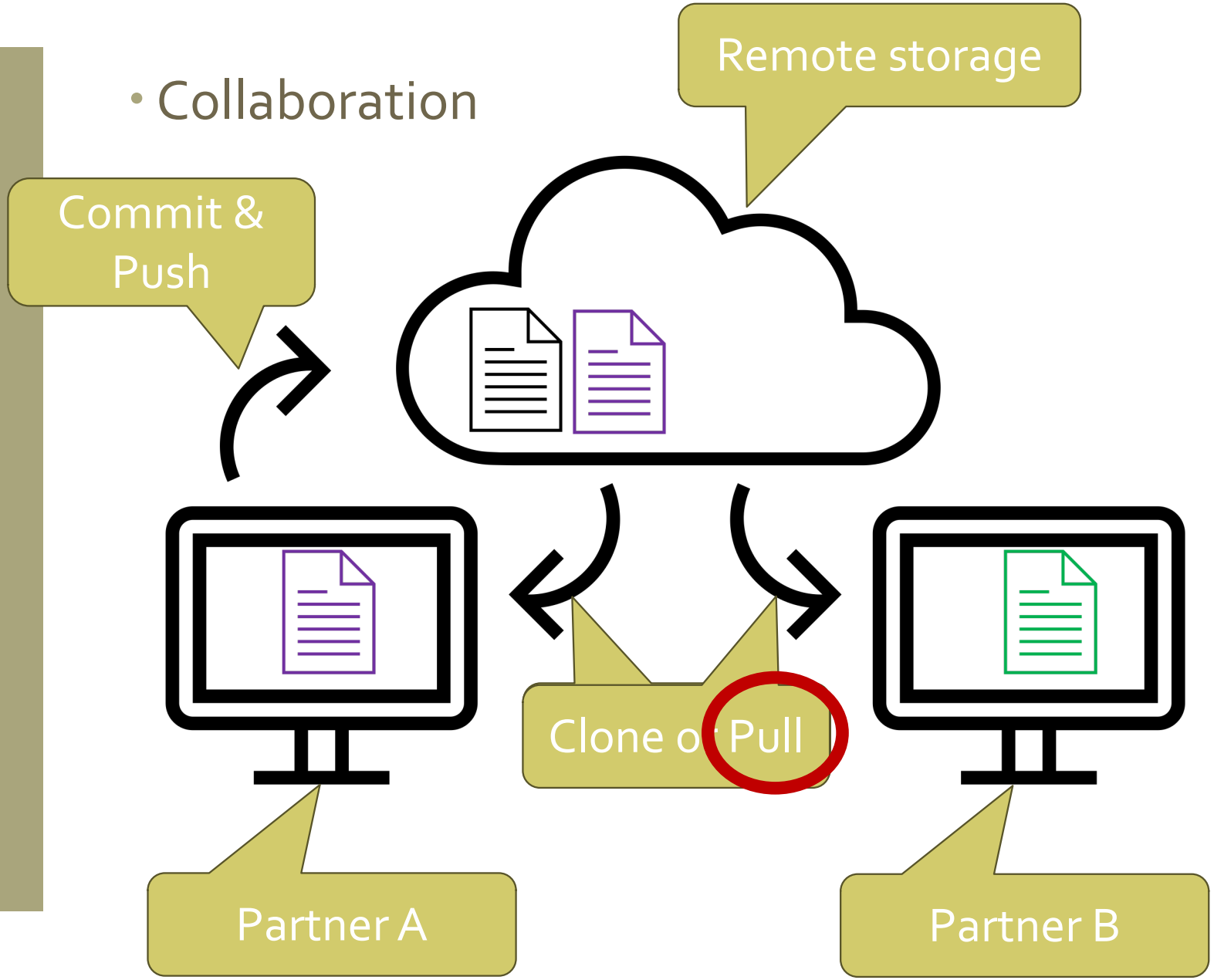
# Benefits to GitHub

- Collaboration



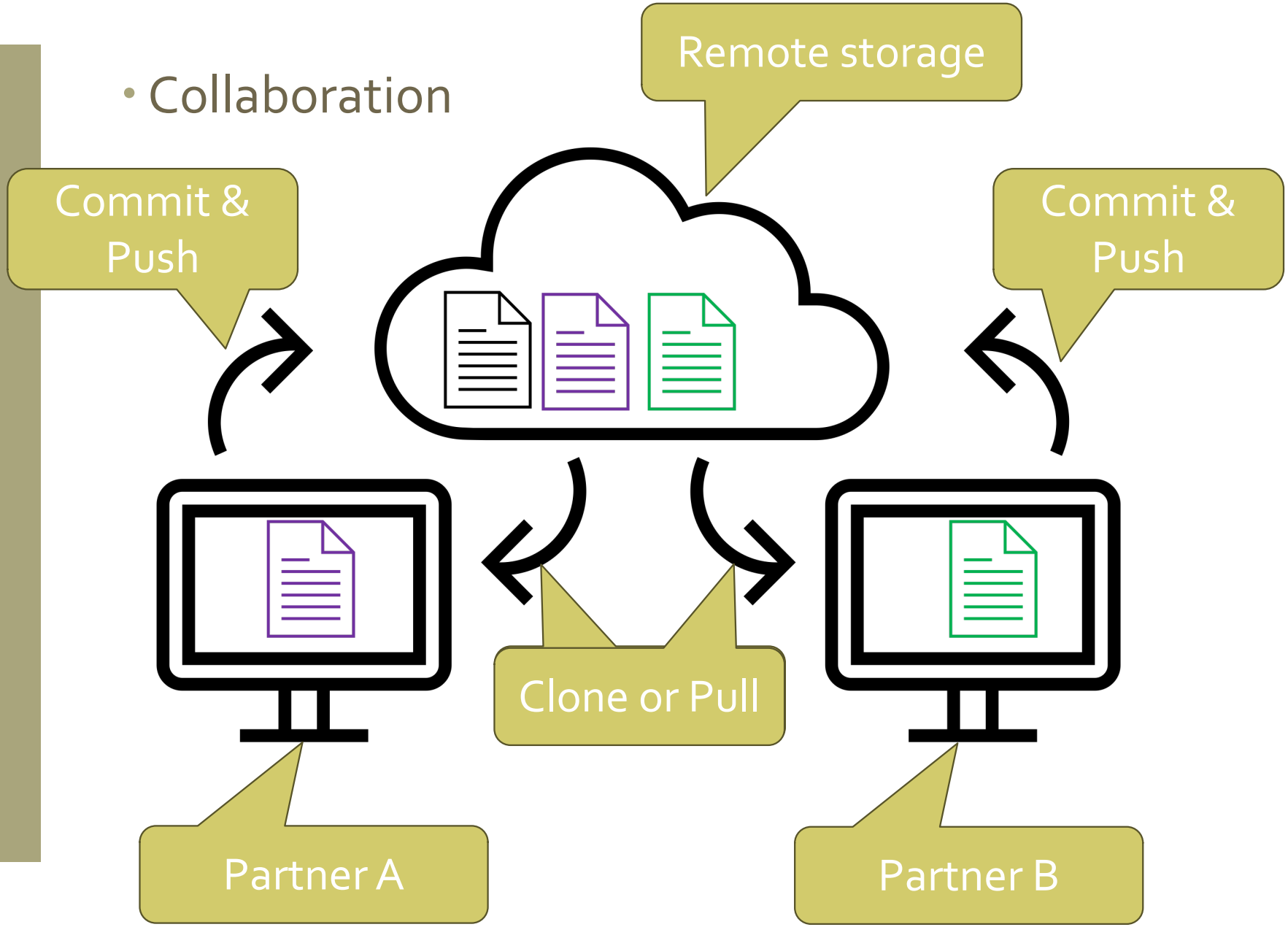
# Benefits to GitHub

- Collaboration



# Benefits to GitHub

- Collaboration



# Benefits to GitHub

What if the two versions can't be combined?

GitHub will tell you by saying there's a merge conflict, and will ask you to decide what to do.

# Benefits to GitHub

What if the two versions can't be combined?

GitHub will tell you by saying there's a merge conflict, and will ask you to decide what to do.

https://youtu.be/MIVWosijSjY?si=nplKpgdi2JgjOo_U

# In-Class Activity

- Wrap up In-class Activity 1 (ico1)
- Be sure to submit on Grade!cope and through GitHub!