



Source Control Systems

- A.k.a revision control, source control
- Source control is the management and tracking of changes to source code, documents, data, etc.
- **Allows collaborative development**
- Keeps track of **who** made a change, **when** the change was made, and **what** the change was
- Permits reverting any change and rolling back to a previous state

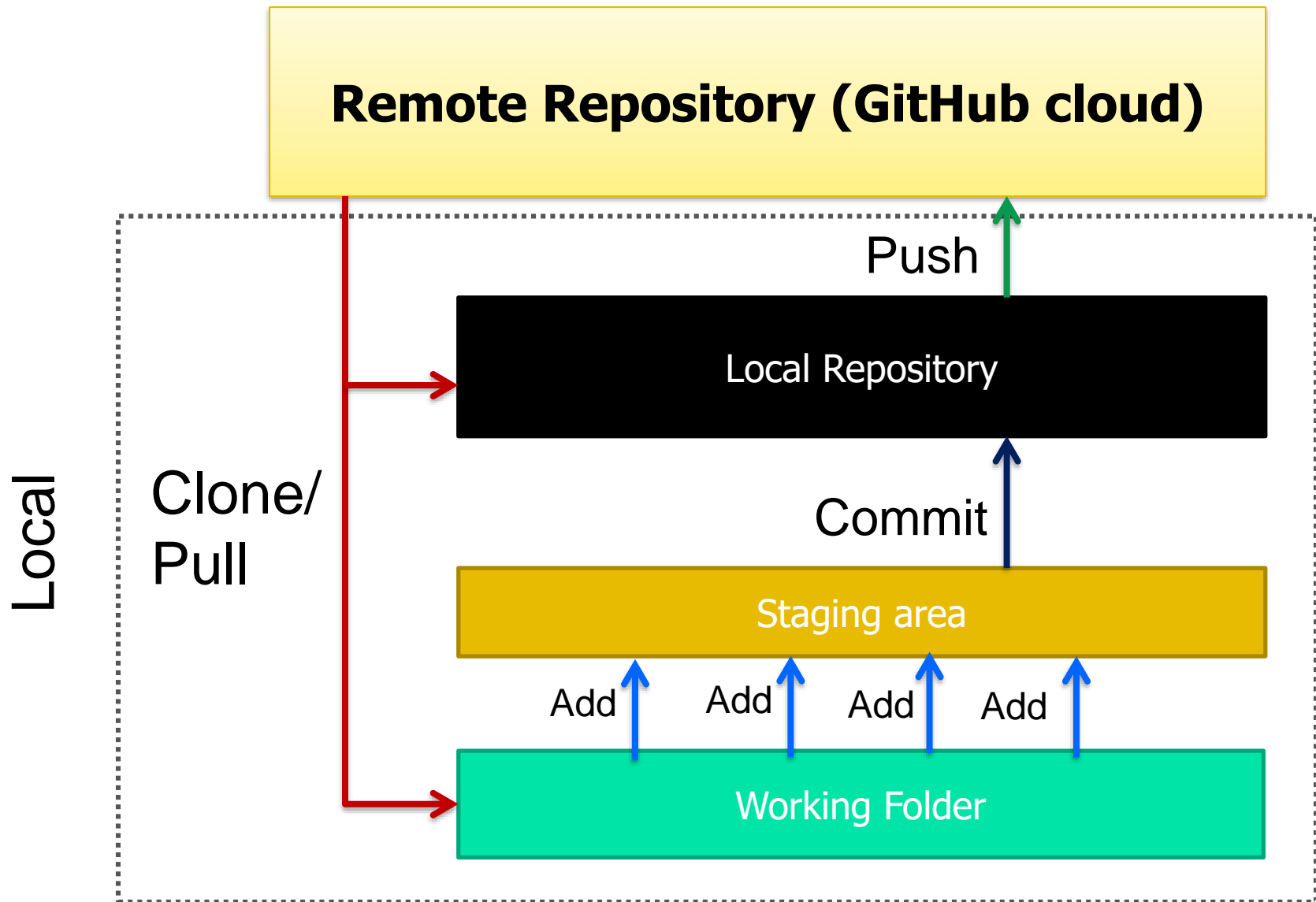
Github

- Github is a distributed source control management system
 - It also provides several collaboration features such as **wikis**, **task management**, and **bug tracking**
- Main characteristics:
 - Entire code and history is kept on the client (user) machine
 - Users can work (make changes to code) even without internet connection
 - Internet connection required only for pushing and pulling from remote repository

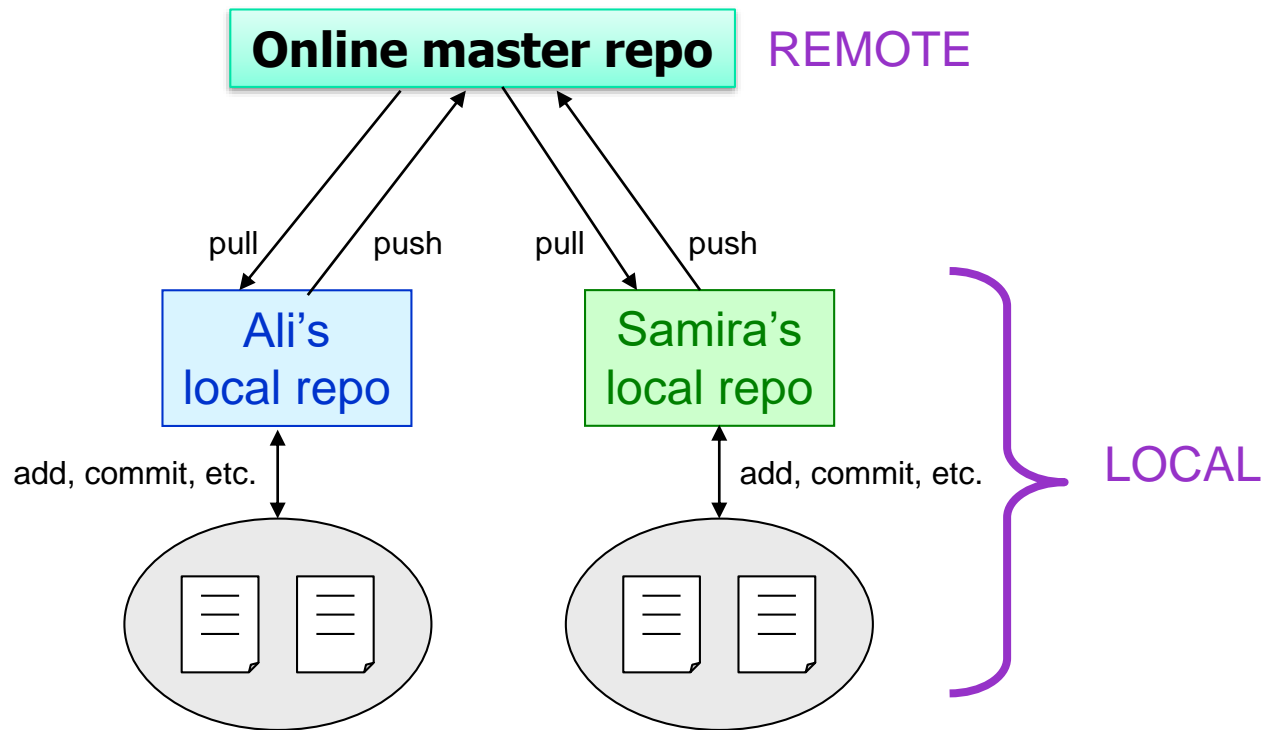
GitHub Basics

- A **repository** (or 'repo') is a collection of all the files and their commit history
- Copying a repository from a remote server is called **cloning**
 - Cloning allows teams to develop collaboratively
- **Pulling**: downloading commits that do not exist on the local machine from a remote repository
- **Pushing**: adding local changes (commits) to a remote repository

Architecture & Terminology

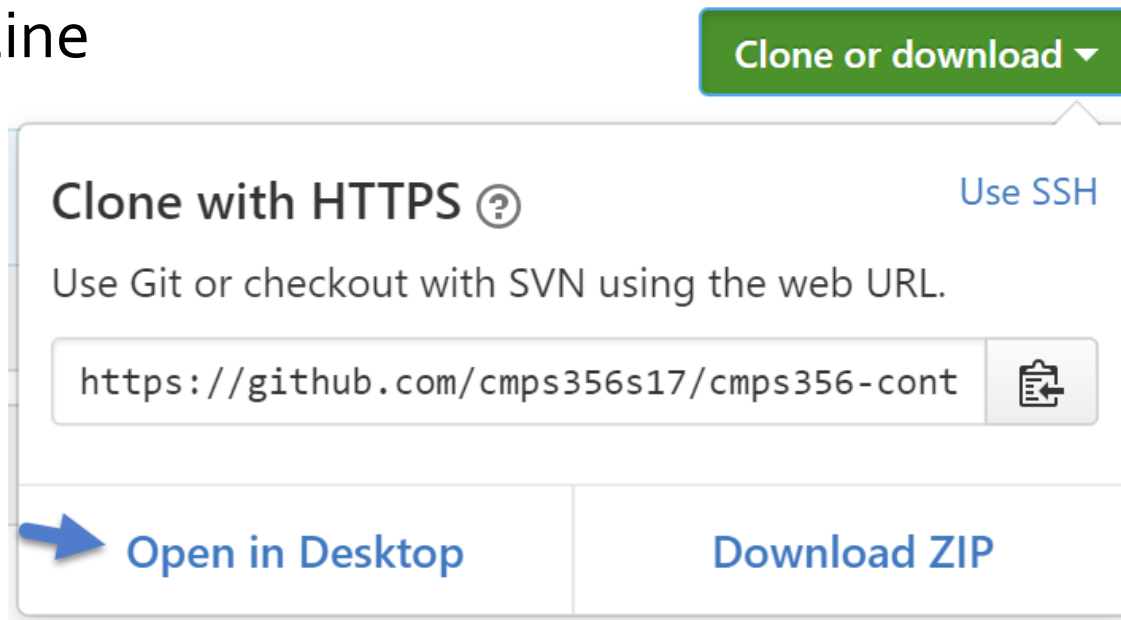


Local and Remote Repositories



GitHub: Create Local Repository

- Each team member creates local repository that is a **clone** of the master repository
 - Log into your personal GitHub account
 - Navigate to the team repository
 - Clone the Repository using GitHub GUI or the Command Line



GitHub: Create Local Repository, *cont'd*

- **cd** to the directory where you want the local repository to reside on your local machine.
- Enter the git command

```
git clone URL
```

- Where *URL* is the repository URL
 - Example:

```
git clone https://github.com/cms356s18/cms356-content.git
```


Git: Make Local Changes

- Get the status of files in your local repository:

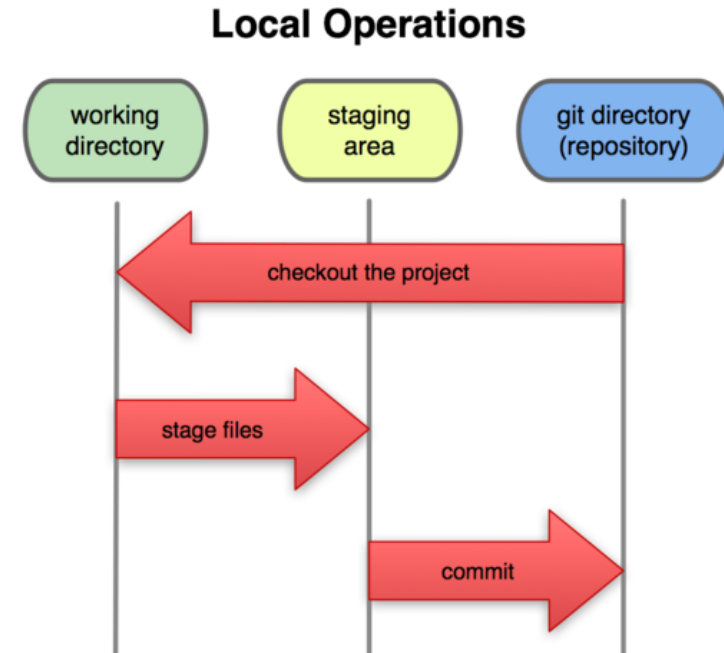
```
git status
```

- After you've updates/created new files on your working directory, first **add** them to the **local staging area**:

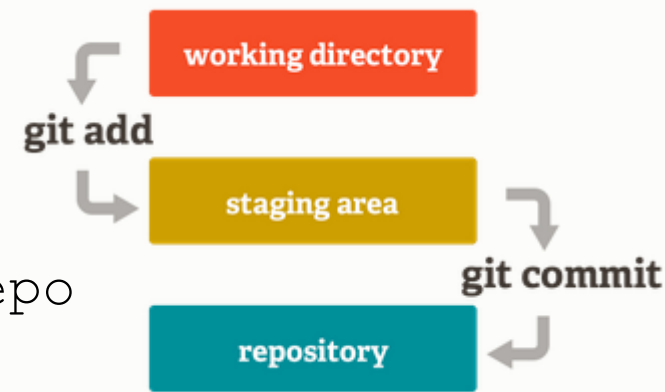
```
git add -A
```

- Commit** your staged files to the **local repository**:

```
git commit -m "commit message"
```



Git Basic Commands Summary



git init //initializes a new git repo

git add -A //adds file to the local staging area

git diff //prints difference made in files

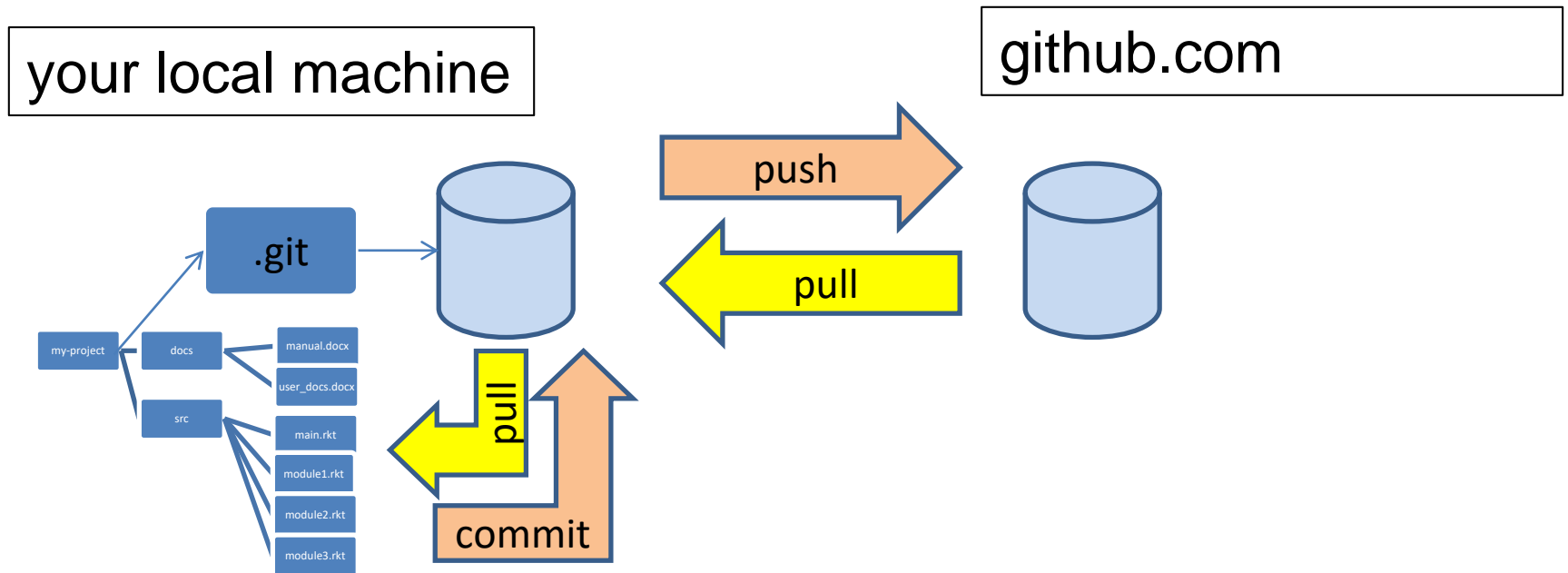
git commit -m "Message here" //save changes to local repository

git status //prints status of current repository

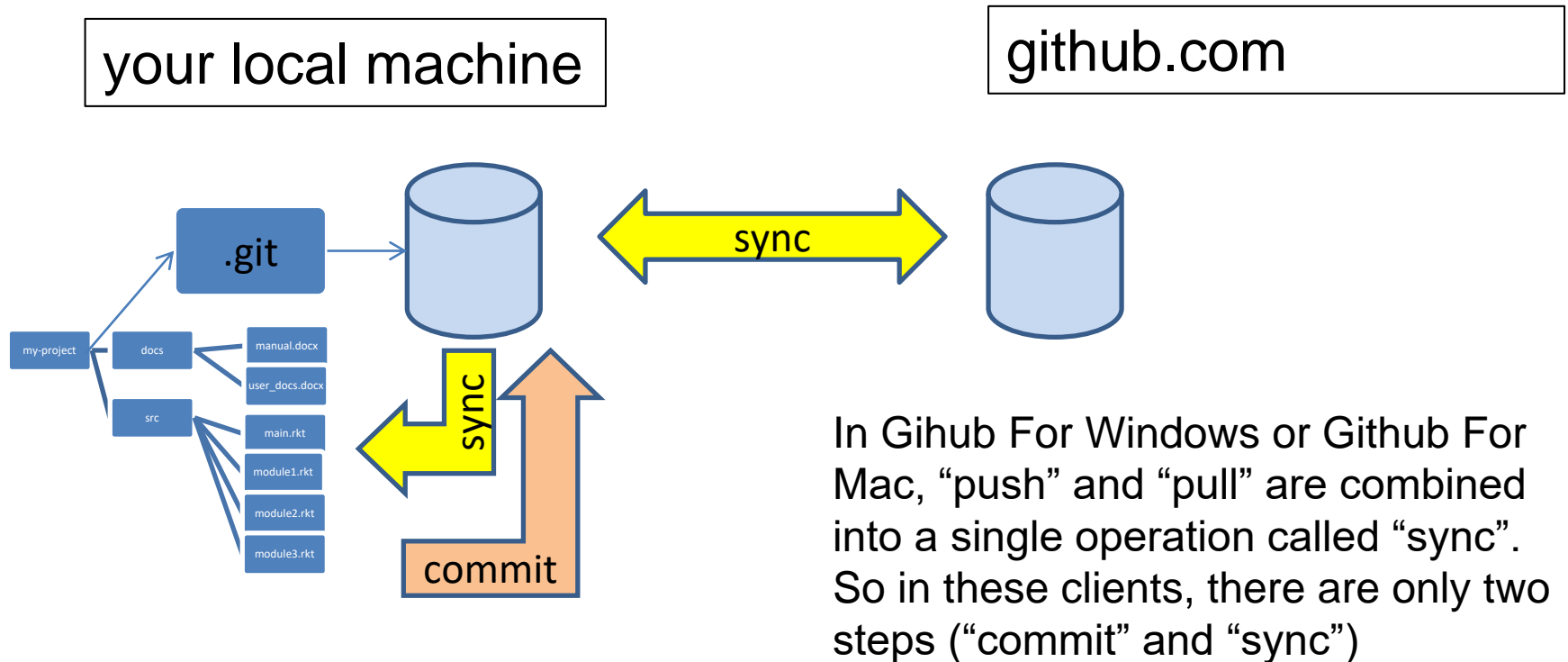
git log //history

git push origin master //push your local changes to your online repository

The Whole Picture



The Whole Picture using GitHub Desktop



In this course, we will mainly use GitHub Desktop

Resources

- GitHub Desktop

<https://desktop.github.com/>

- GitHub foundation short videos

<https://www.youtube.com/playlist?list=PLologMOBetEHhfG9vJzVCTiDYcbhAiEqL>

- GitHub Help

<https://help.github.com/>

- Git Book

<https://git-scm.com/book/>