GitHub

Source Control Systems

- Aka 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

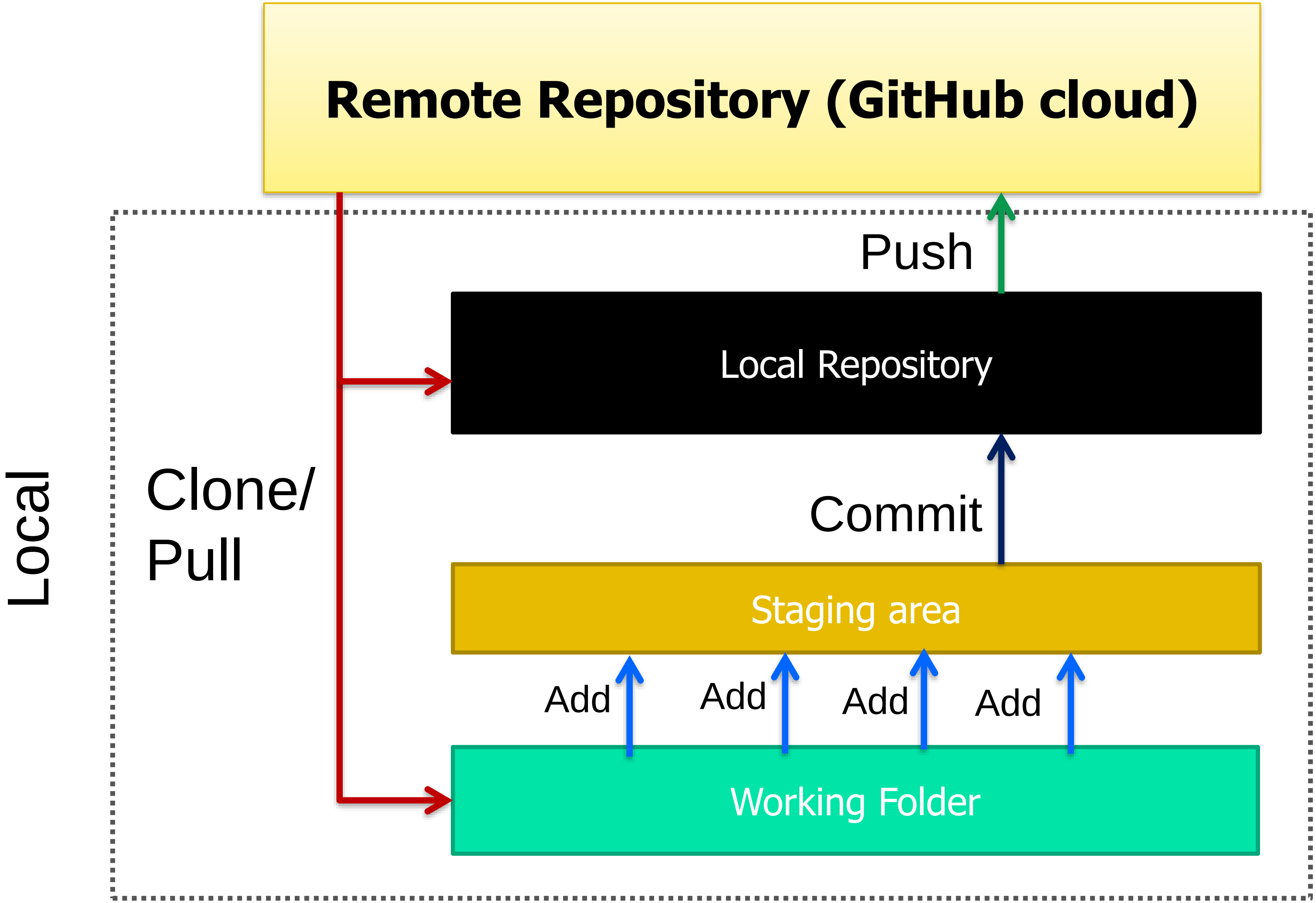
Github

- 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

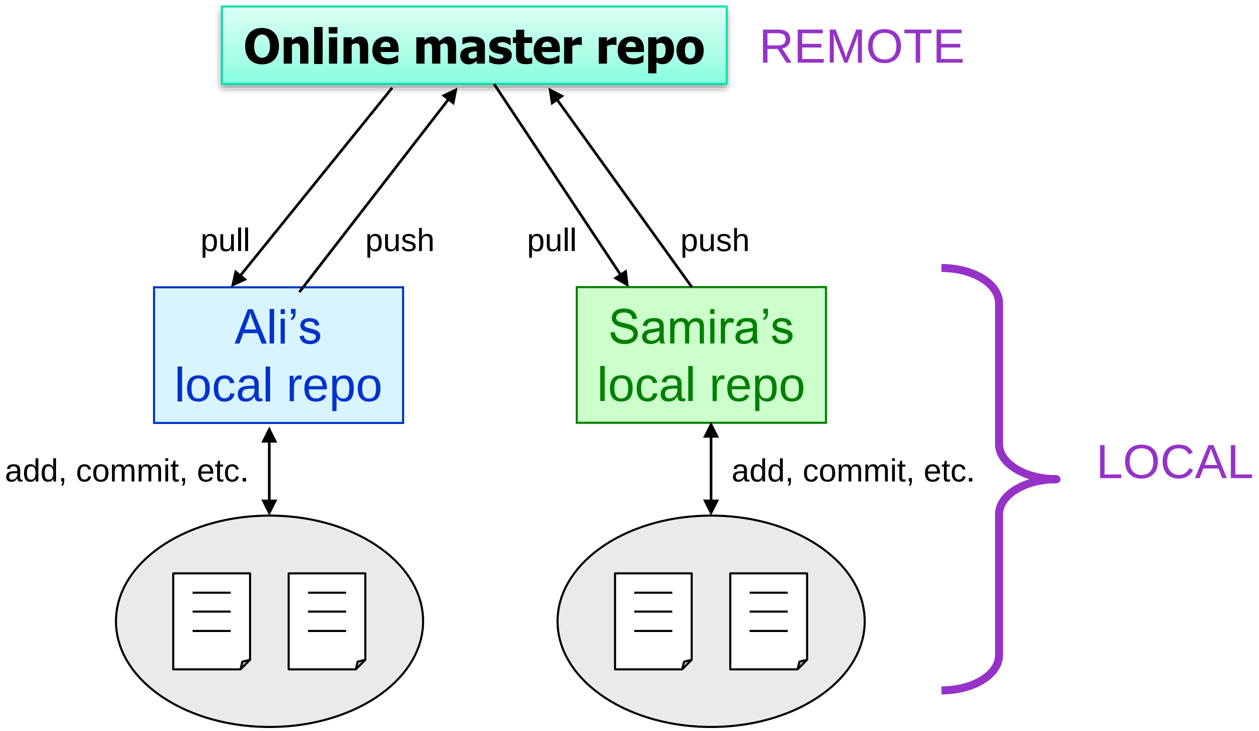
GitHub Basics

- A repository (or 'repo') is a collection of all the files and their commit history
  - contains all commits
  - can be local or remote
- 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

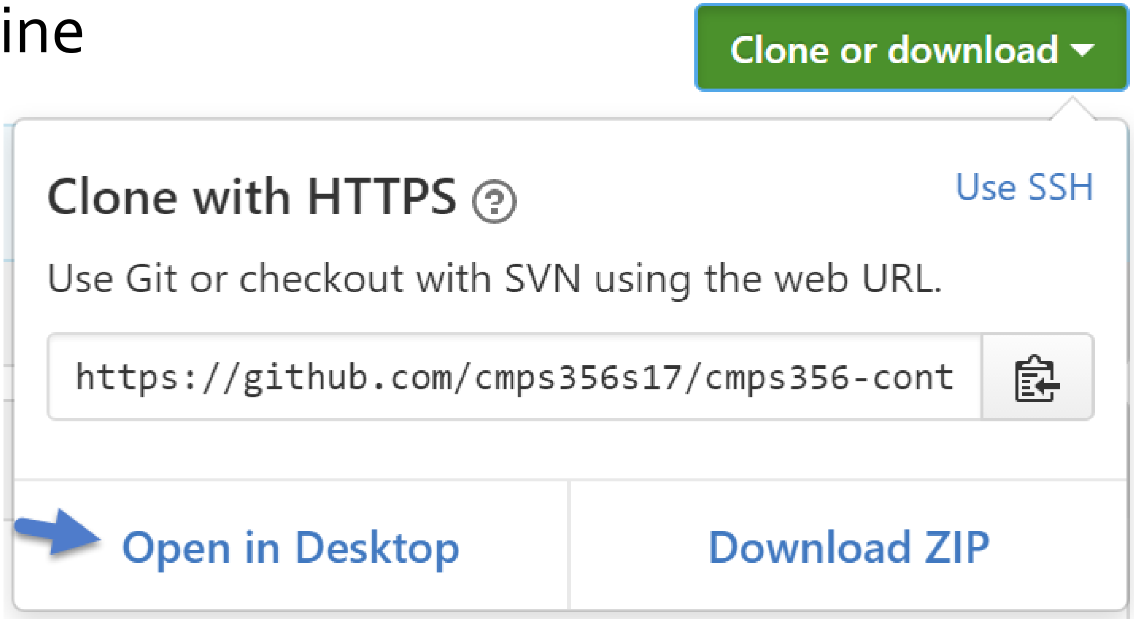


Local and Remote Repositories



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



Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/cmps356s17/cmps356-cont
```

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
```

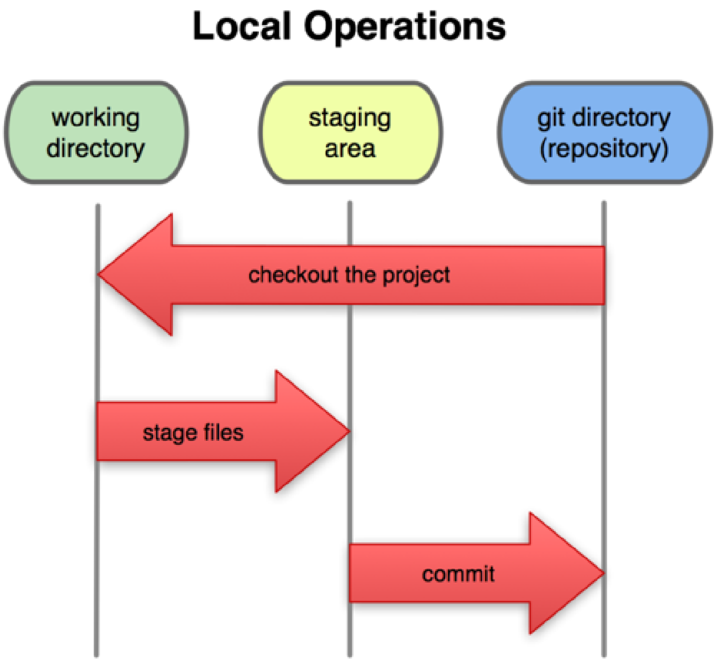
- Where *URL* is the repository URL
  - Example:

```
git clone https://github.com/cmps356s17/cmps356-content.git
```

Git: Make Local Changes

- Get the status of files in your local repository:

```
git status
```



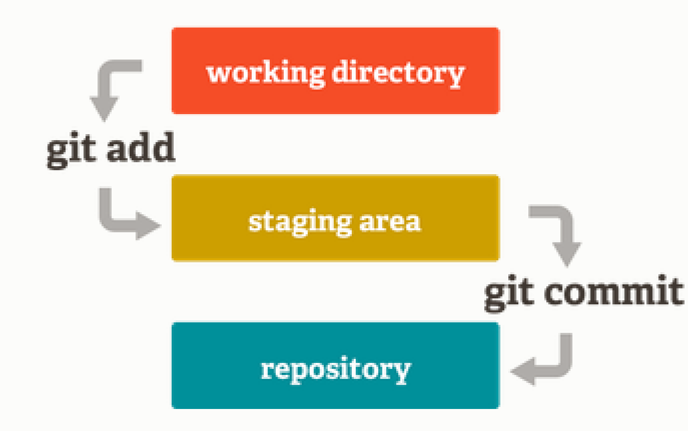
- After you've created new files on your working directory, first add them to the local staging area:

```
git add myfile1 myfile2
```

- Commit your staged/modified files to the local repository:

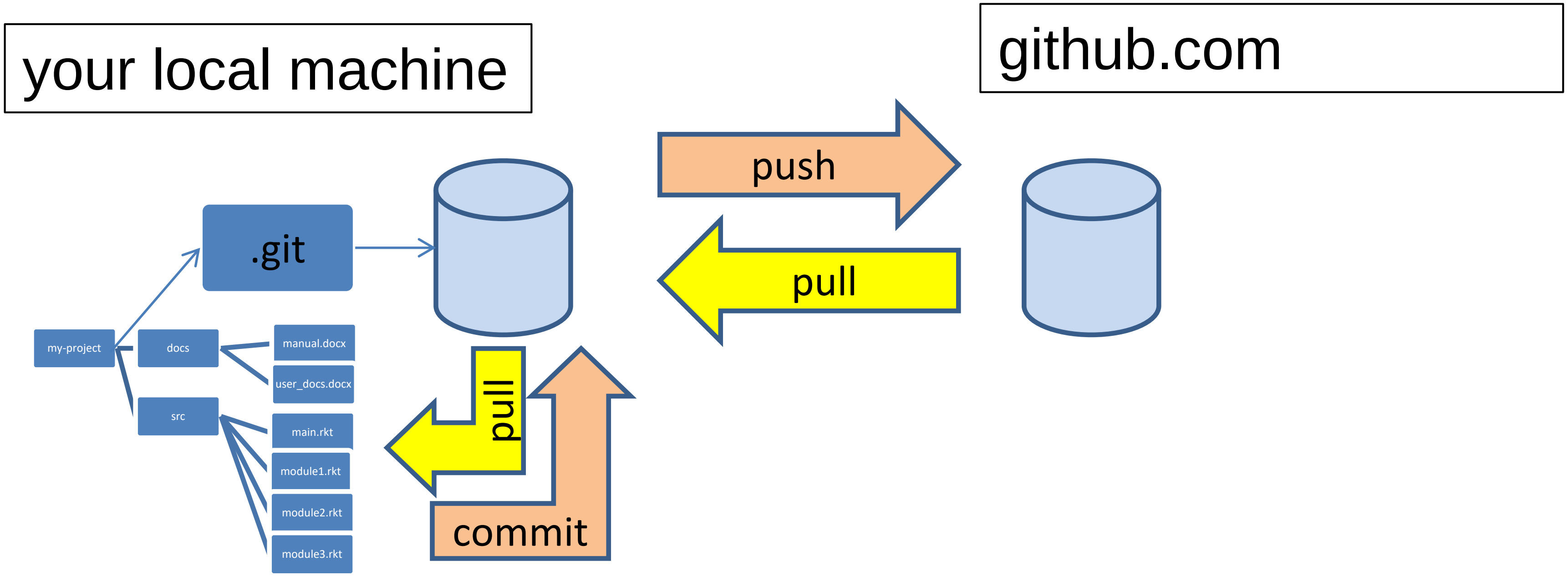
```
git commit –m "commit message"
```

Git Basic Commands Summary



```
git init  //initializes a new git repo

git add filename  //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 [options] origin branch_name //updates a remote
repository with the changes made locally
```

The Whole Picture



your local machine

github.com

push

pull

.git

my-project

docs

manual.docx

user_docs.docx

src

main.rkt

module1.rkt

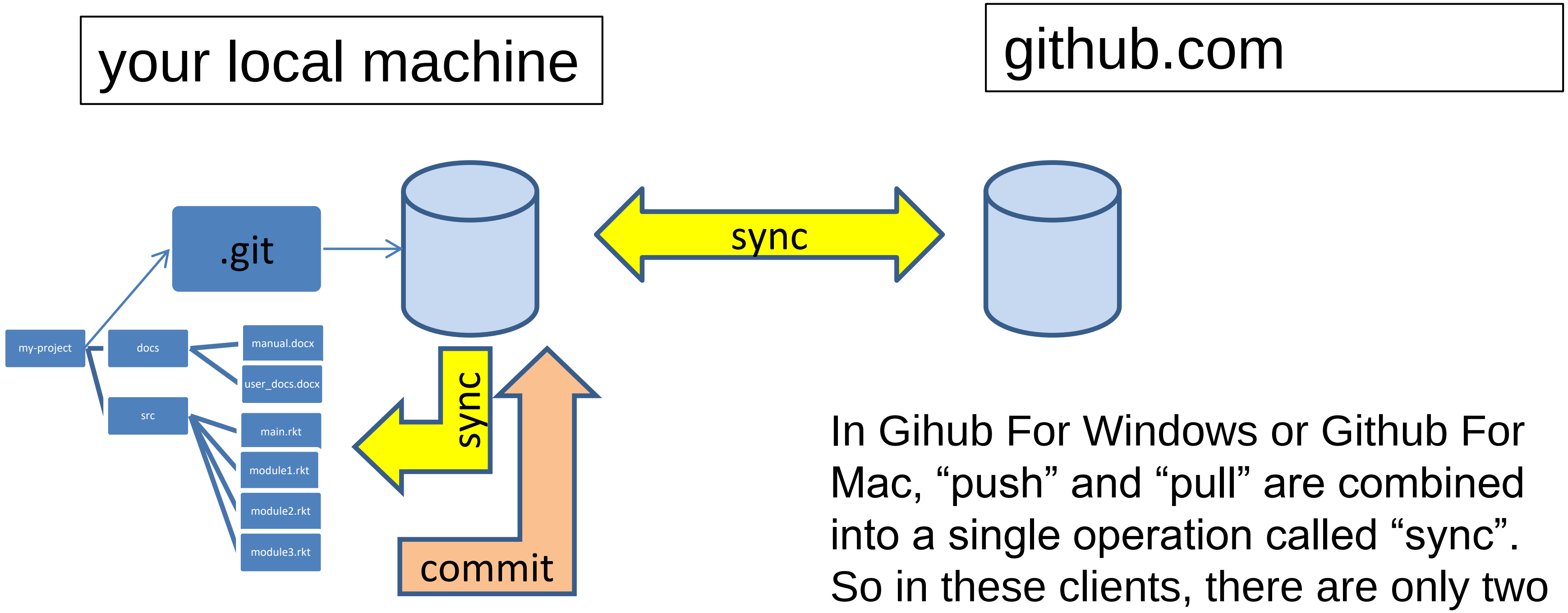
module2.rkt

module3.rkt

pull

commit

The Whole Picture using GitHub Desktop



In Gihub For Windows or Github For Mac, “push” and “pull” are combined into a single operation called “sync”. So in these clients, there are only two steps (“commit” and “sync”)

In this course, we will mainly use GitHub Desktop

Resources

- Github foundation short videos @

https://www.youtube.com/playlist?list=PLolo9MOBetEHhfG9vJzVCTiDYcbhAiEqL