git + Github

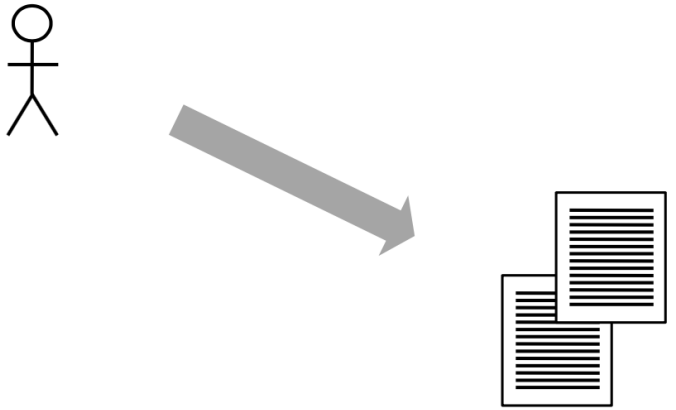
a better workflow

Megha Aggarwal

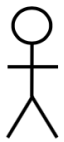
http://github.com/codeblooded

Why git?

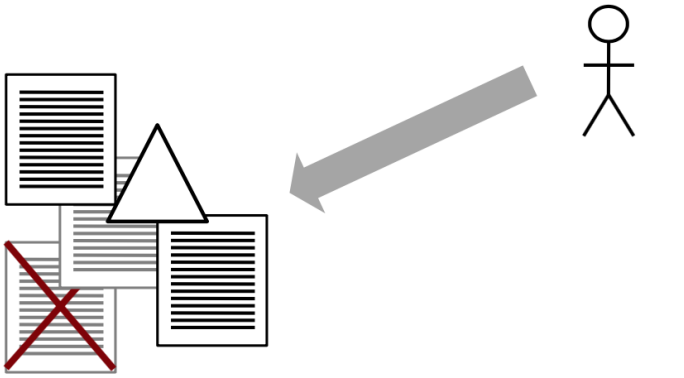
Bob

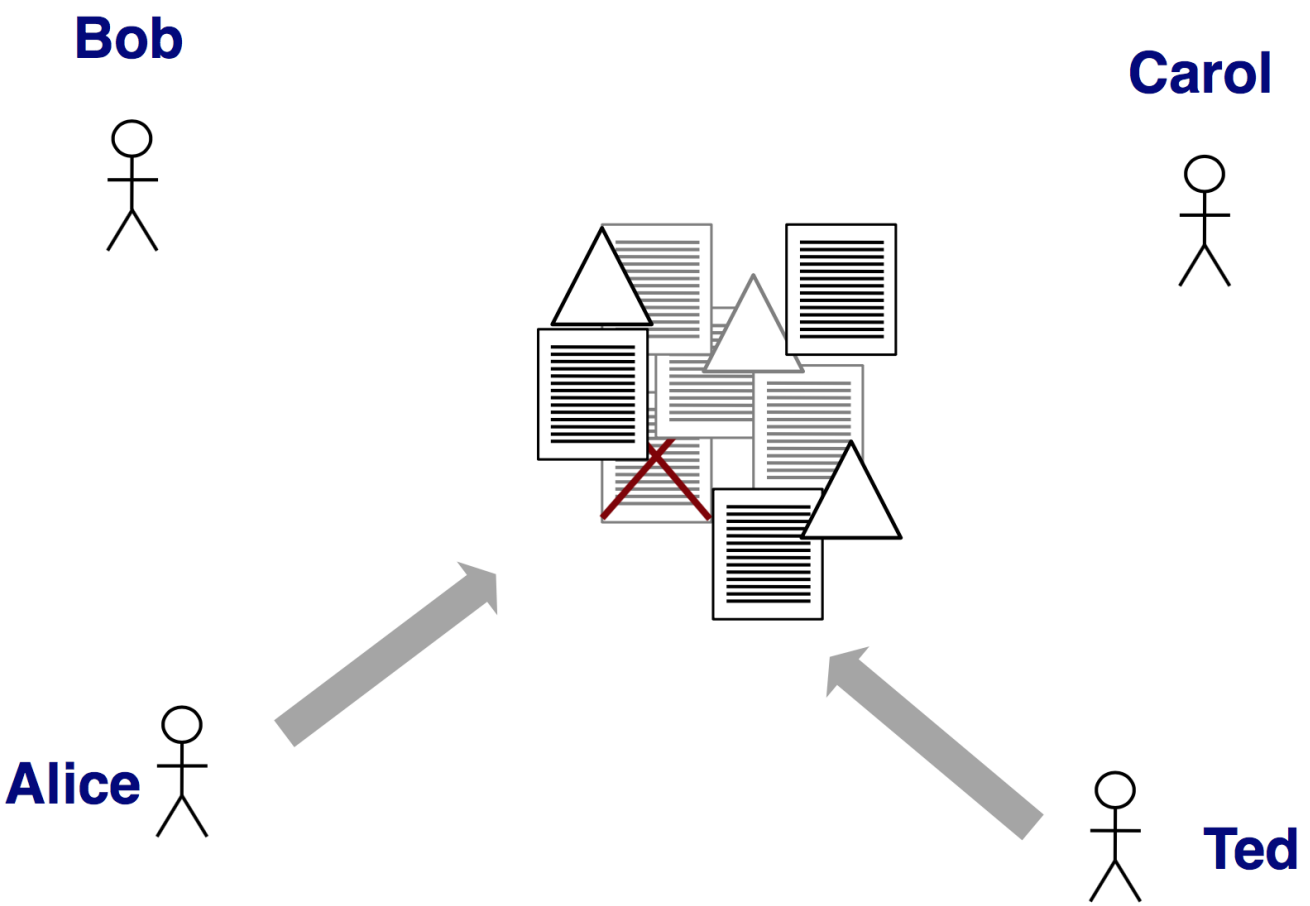


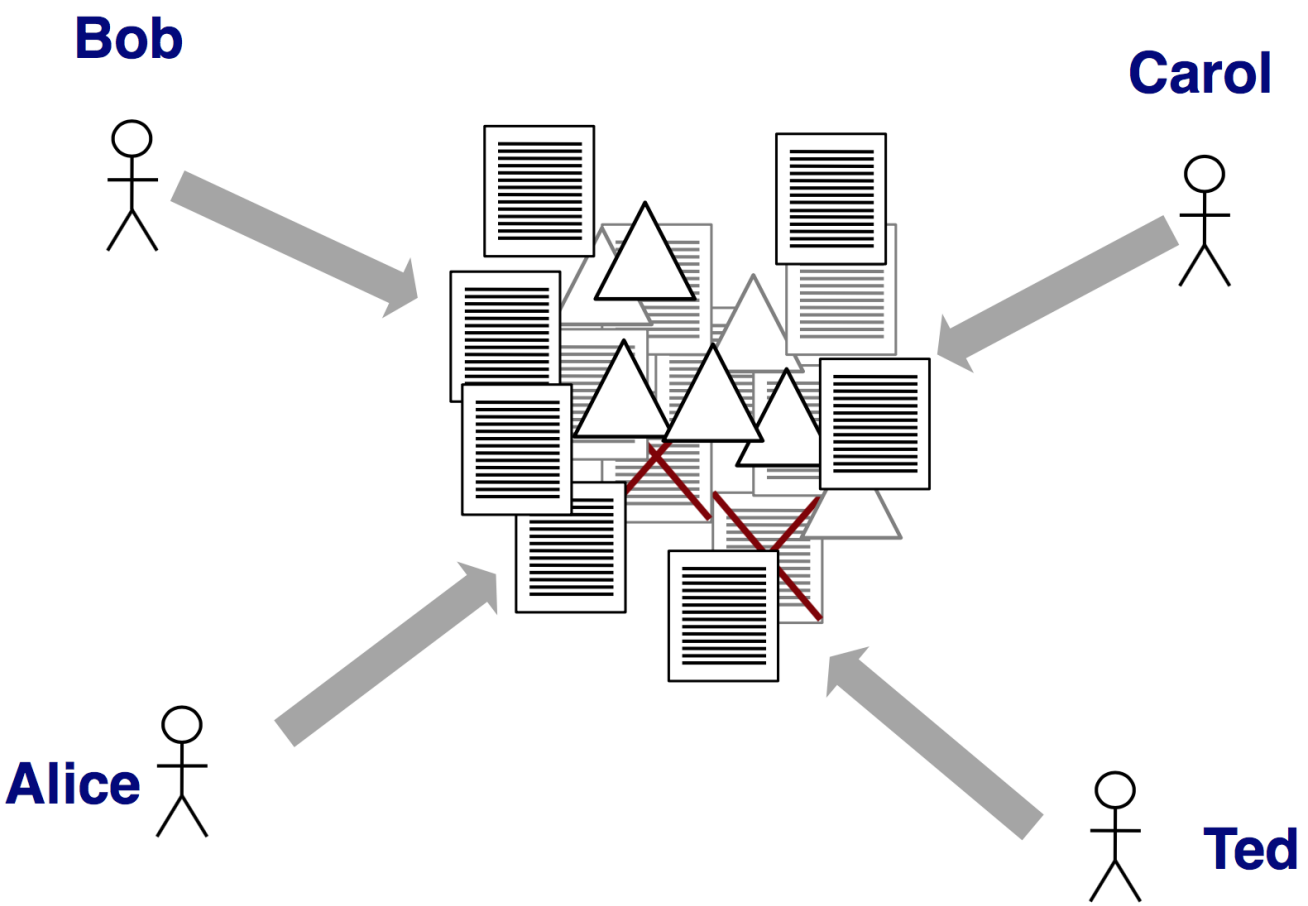
Bob



Carol



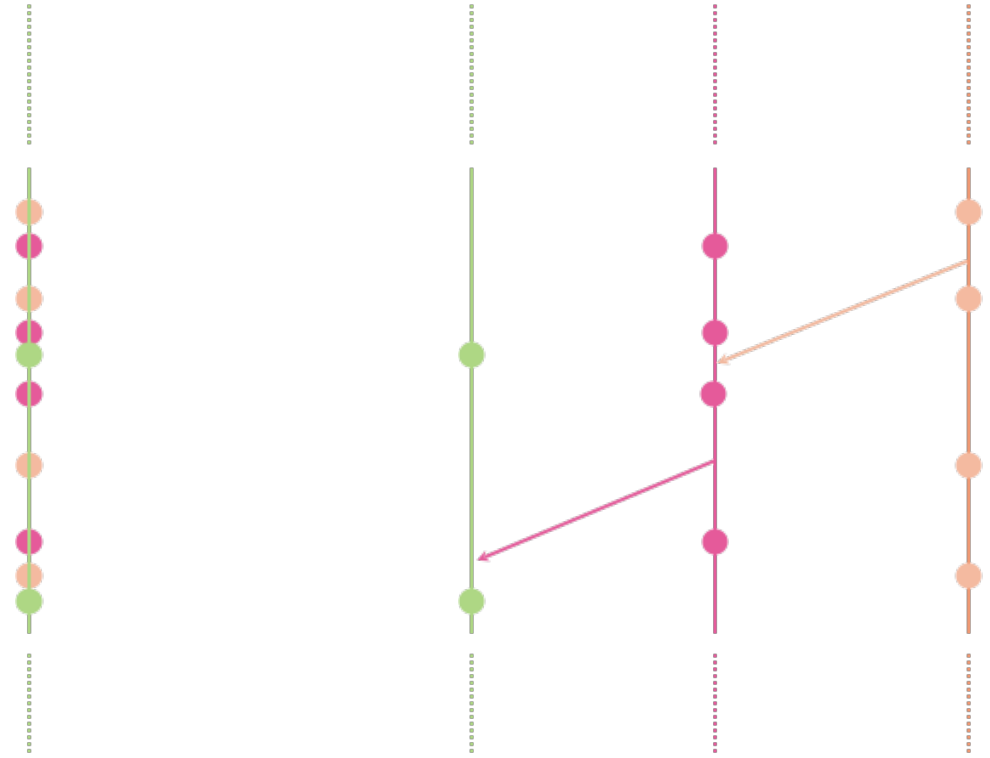

Bob
Carol
Alice
Ted

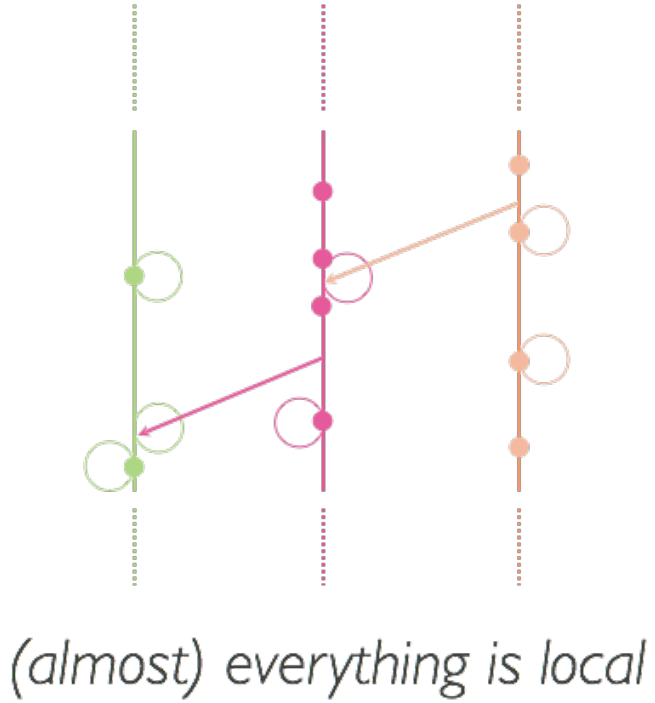


A recipe for disaster!

What is git?

Git is an open source, distributed version control system designed for speed and efficiency





Version control system

distributed

Speed and efficiency

Installing Git

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

Github

https://github.com

Setup

```
$ git config --global user.name "<YOUR_NAME>"

$ git config --global user.email "<YOUR_EMAIL>"
```

Hands-on

My first repo

Over github

Over local machine

- *Make a project by simply creating a folder or choose an existing one*
- *Initialize git repository*

  $ git init
- *Lets check status of our repo*

  $ git status

- *Lets add some files*

  $ git add <file_name>

  Note: *to add all files use:* $ git add .

- *Commit all the changes*

  $ git commit -m <commit_message>

- *Add origin of your local repo or we can say setup remote of local repo*

  $ git remote add origin <remote_url>

- *Push changes (commit) to github*

  $ git push -u origin master

Working on already present repo

Cloning a repo

- *Go to github and get url of repo you want*
- *Clone the repo in your machine*

```
$ git clone <url_of_repo>
```

First time pushing command

```
$ git push -u origin master
```

Making changes to repo

If you have access to make changes

- *Make changes in your project*
- *Stash those changes*

  $ git stash
- *Pull changes from remote repo or say, synchronize your local repo with remote repo*

  $ git pull origin master
- *Get your changes back from remote repo*

  $ git stash pop

- *Add your changes*

  $ git add .
- *Make a commit of your changes*

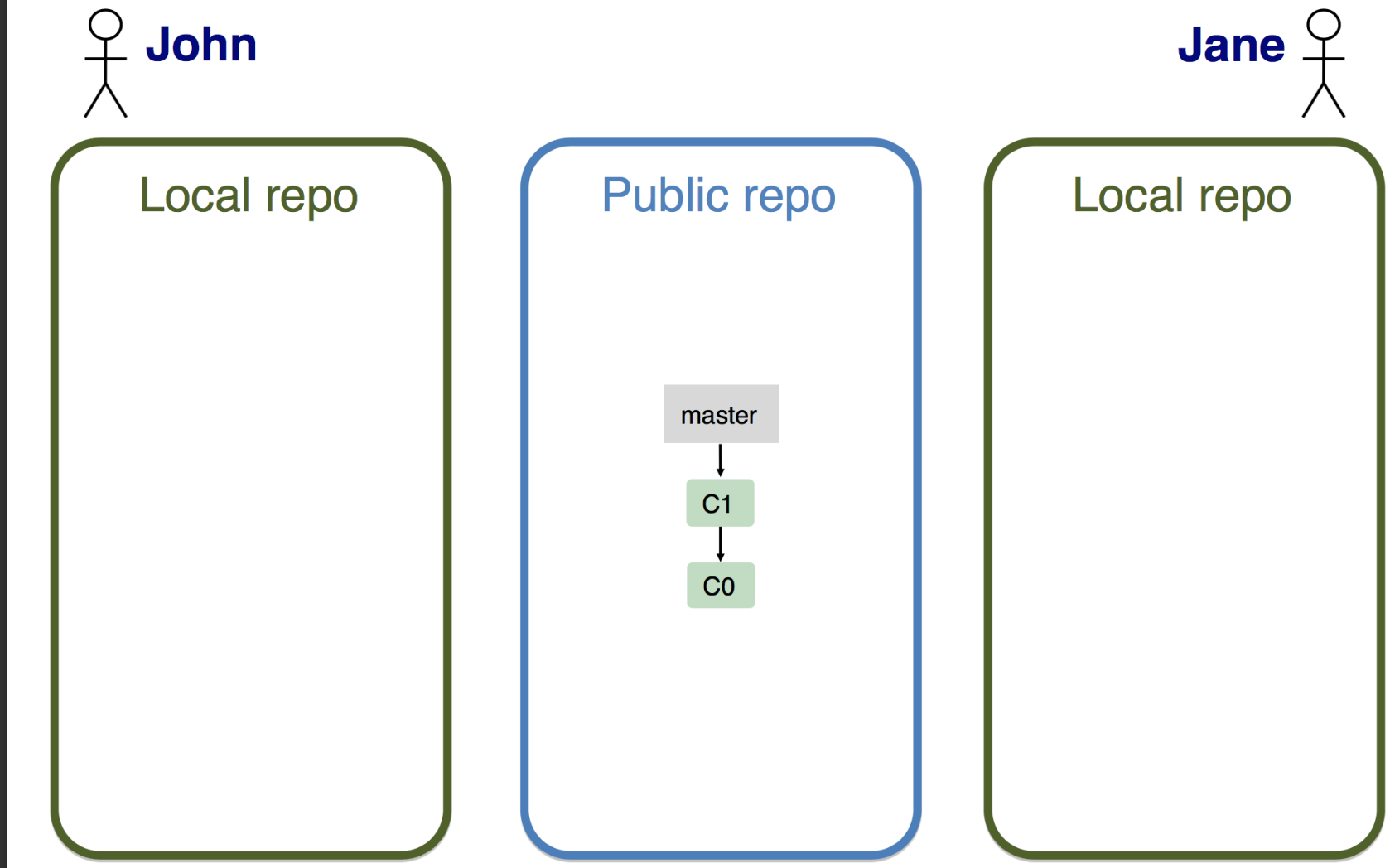
  $ git commit -m <commit_message>
- *Push your changes to the remote repo*

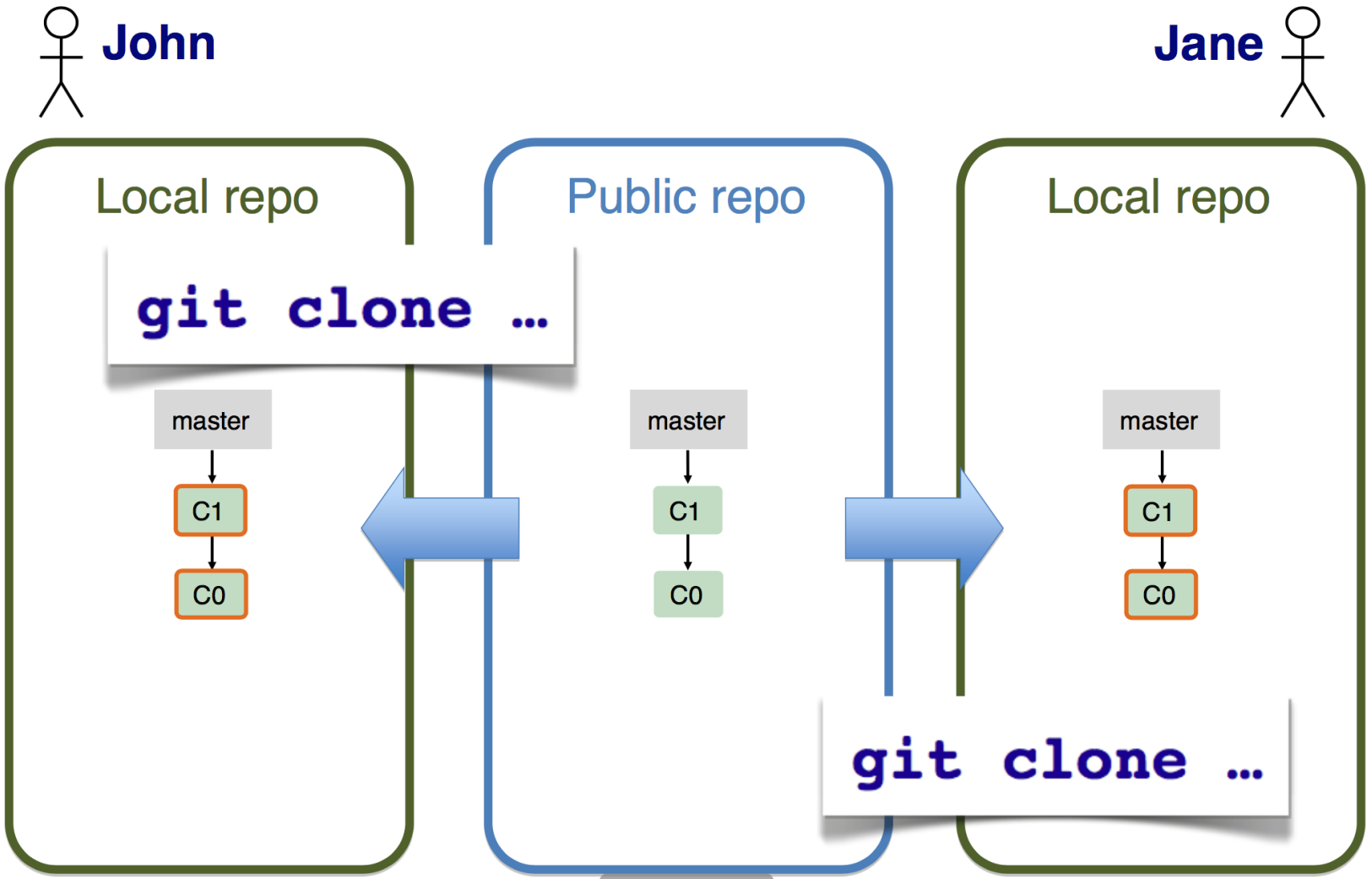
  $ git push origin master

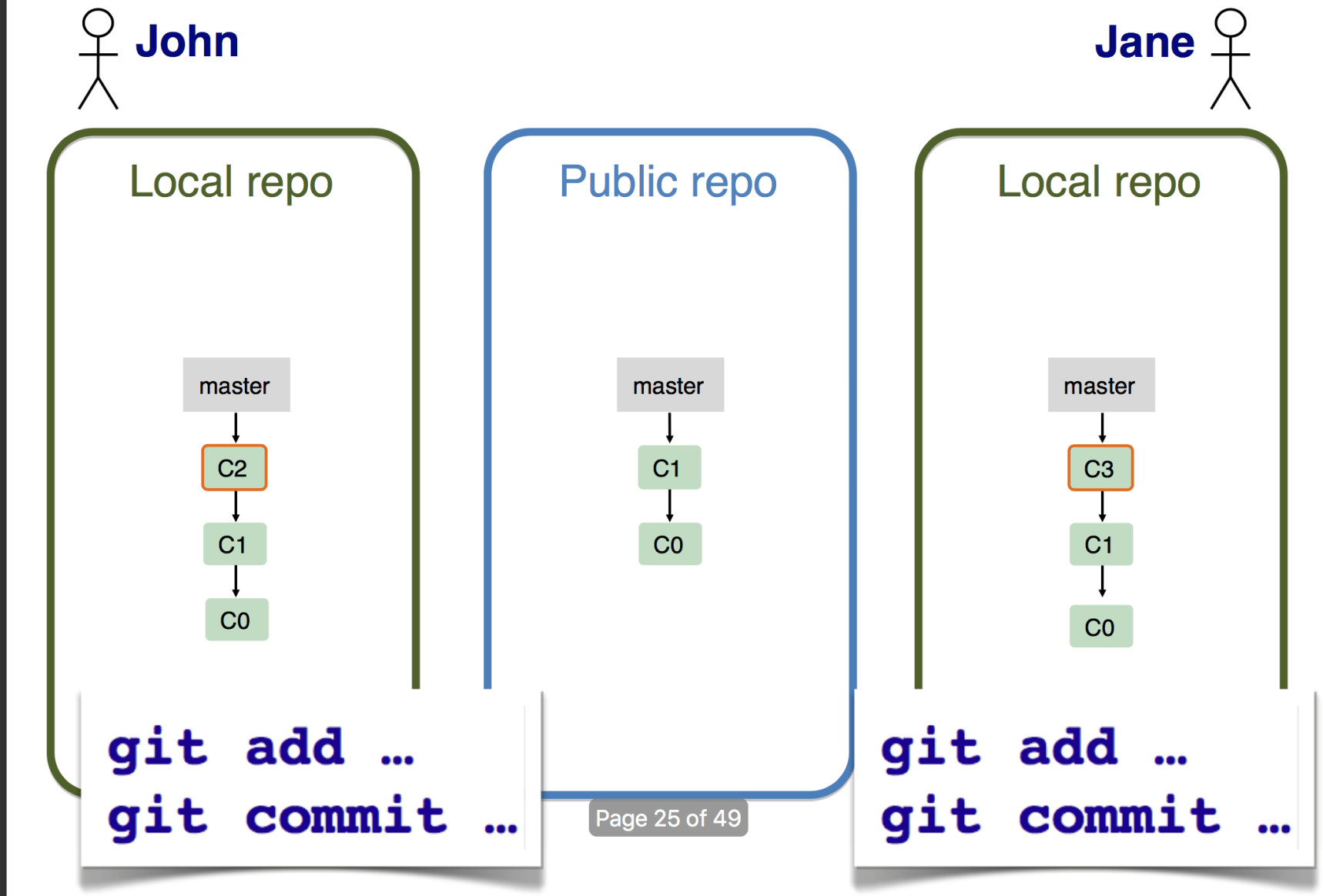
If you don't have access

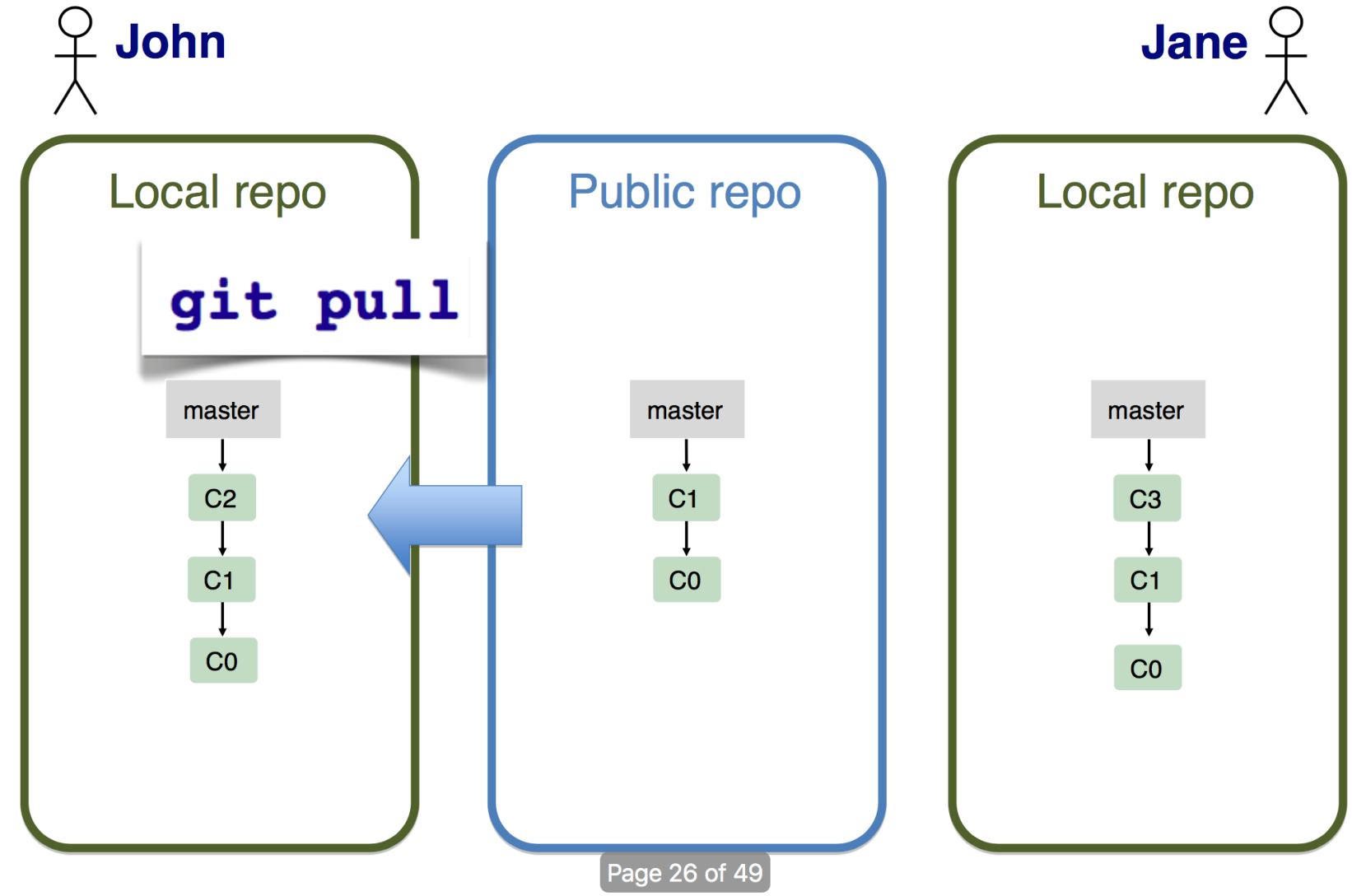
- *Make a pull request. Read about it online.*

Collaborating

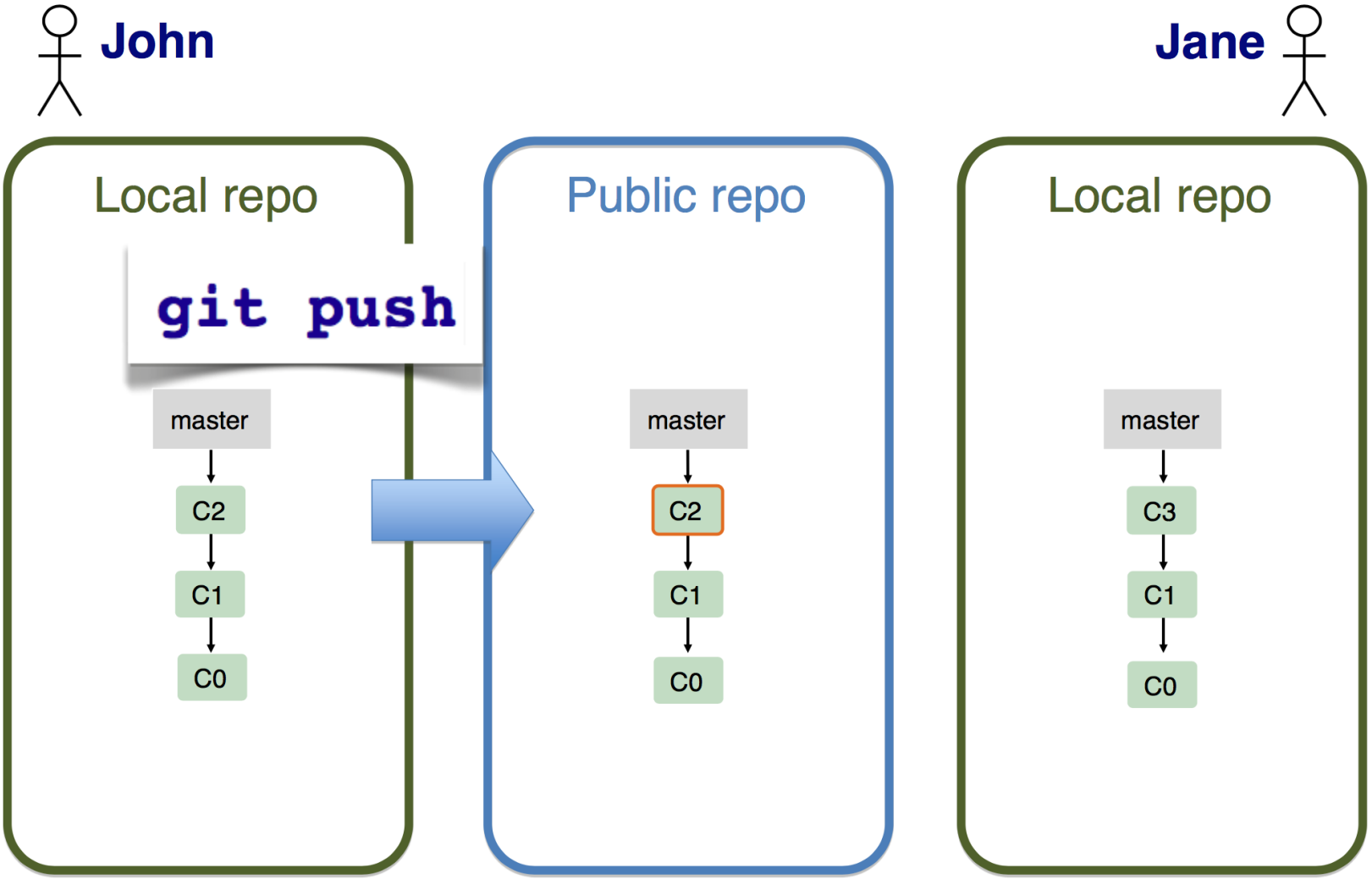

John
Jane
Local repo
Public repo
Local repo
master
C1
C0

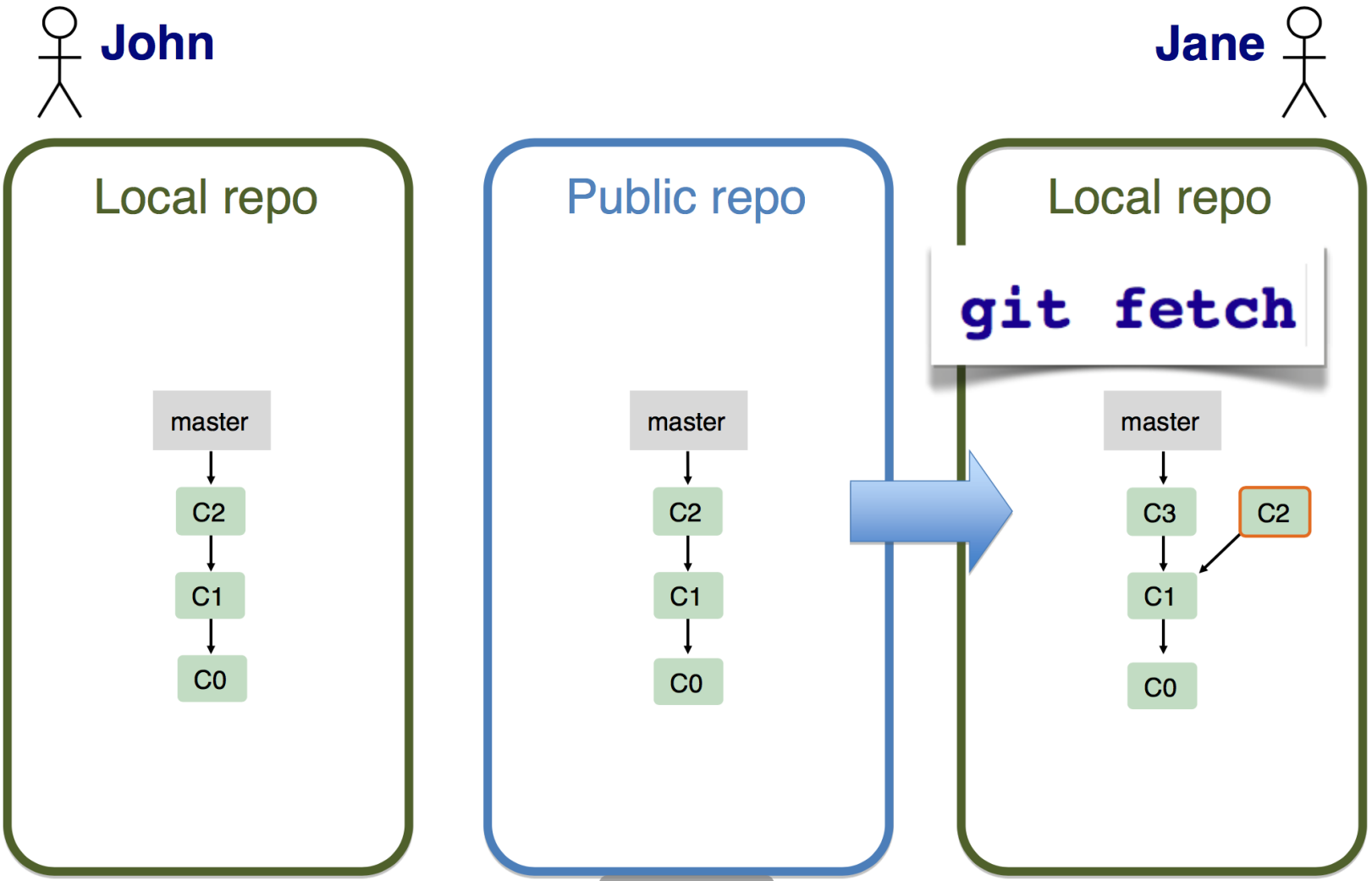

John
Jane
Local repo
Public repo
Local repo
git clone …
master
C1
C0
master
C1
C0
master
C1
C0
git clone …

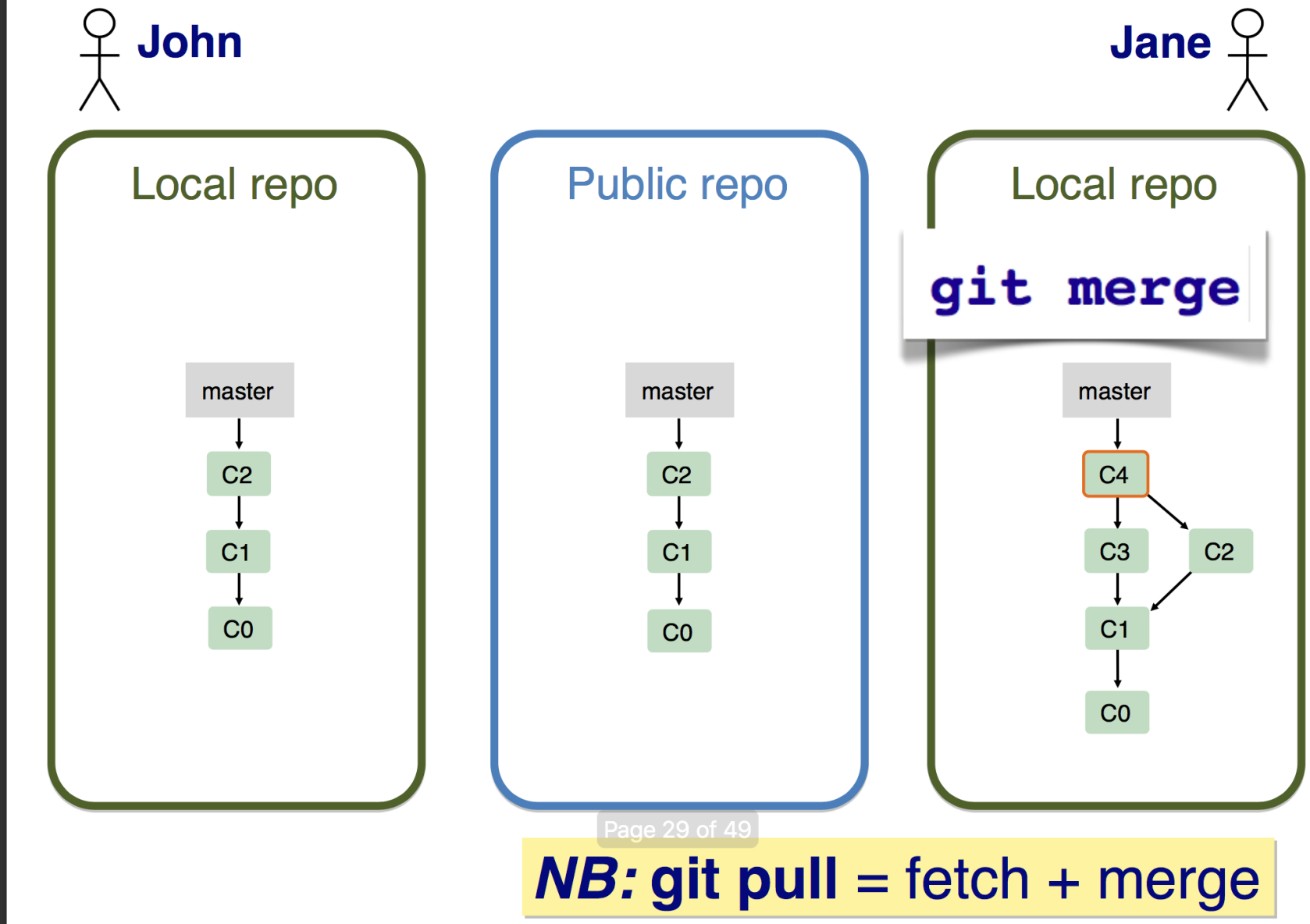

John
Jane
Local repo
Public repo
Local repo
master
C2
C1
C0
master
C1
C0
master
C3
C1
C0
git add …
git commit …
git add …
git commit …



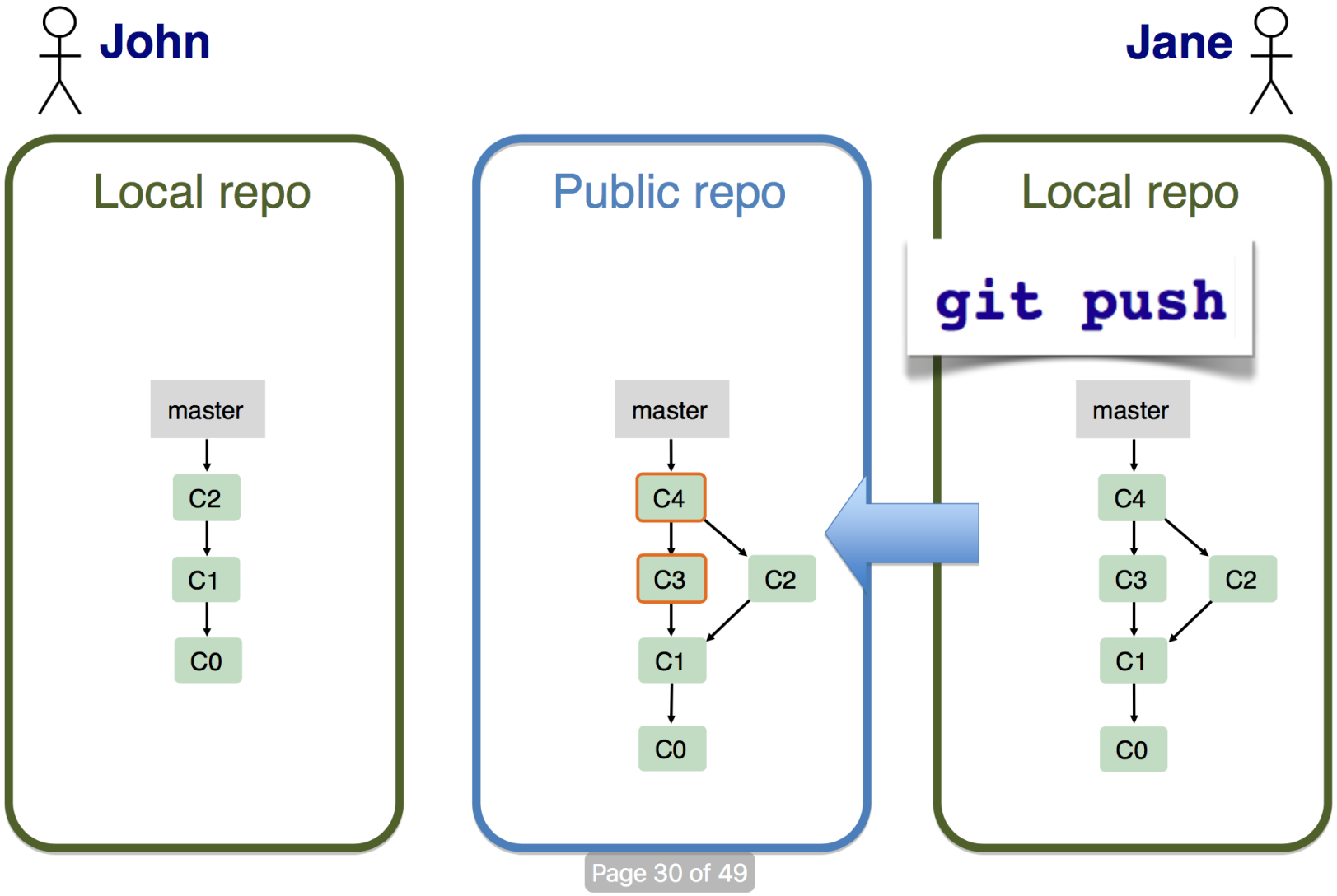
(nothing new to pull)

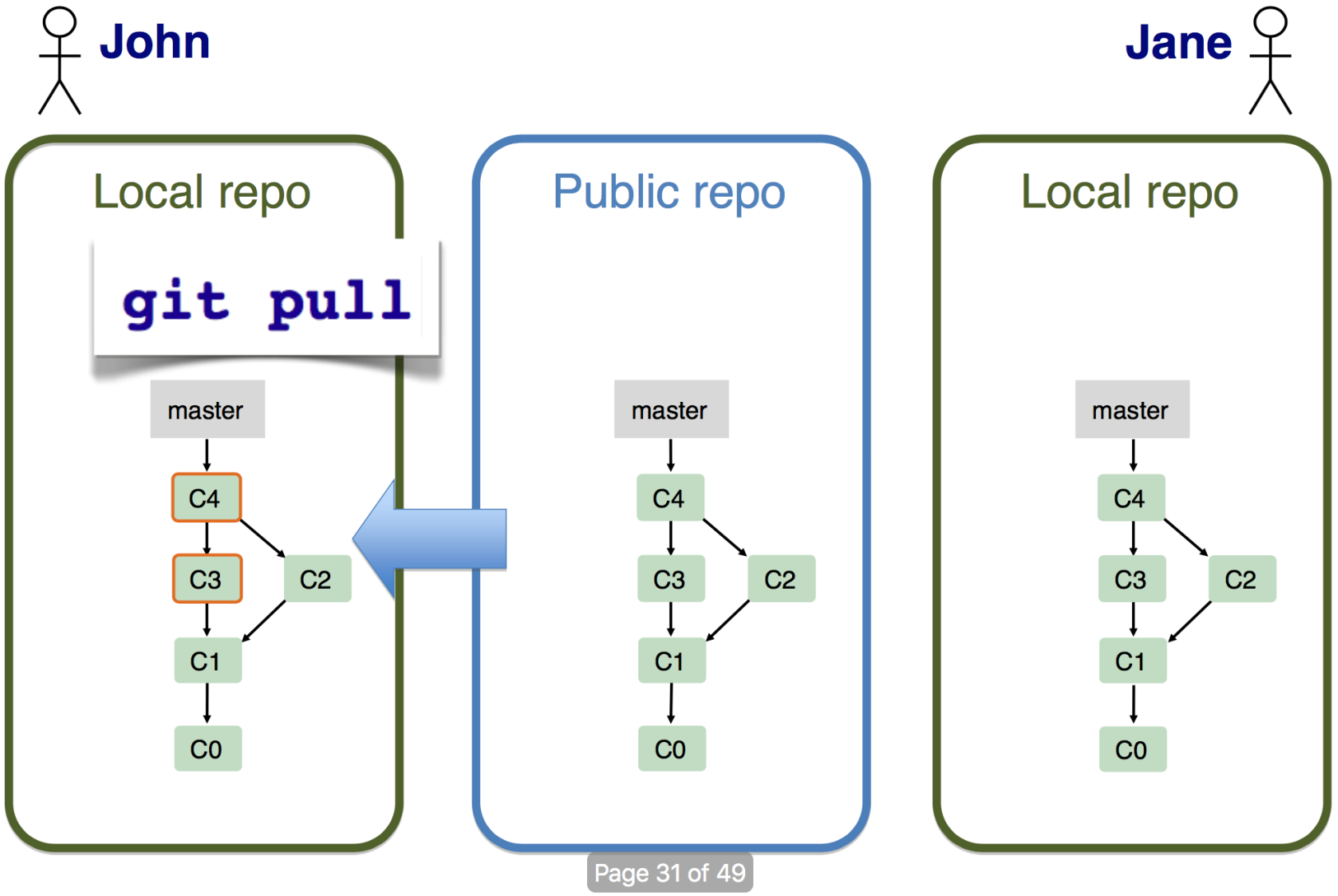

John
Jane
Local repo
git push
master
C2
C1
C0
Public repo
master
C2
C1
C0
Local repo
master
C3
C1
C0


John
Jane
Local repo
Public repo
Local repo
git fetch
master
C2
C1
C0
master
C2
C1
C0
master
C3
C2
C1
C0



NB: **git pull** = fetch + merge


John
Jane
Local repo
Public repo
Local repo
git push
master
C2
C1
C0
master
C4
C3
C2
C1
C0
master
C4
C3
C2
C1
C0


John
Jane
Local repo
git pull
master
C4
C3
C2
C1
C0
Public repo
master
C4
C3
C2
C1
C0
Local repo
master
C4
C3
C2
C1
C0

Branching

- *git branch <branch-name> #create a branch*
- *git checkout <branch-name> #shift to that branch*
- *git checkout -b <branch-name> #create a branch and shift to it*
- *git push origin <branch-name> #to push branch changes on git*
- *git fetch origin #receive a reference to branch on the server but not getting all the files*
- *git checkout -b <branch-name> origin/<branch-name> #to get the whole entire branch*
- *git branch #see all your branches*
- *git branch --merged #see all merged branches*
- *git branch --no-merged #see unmerged branches*

- git branch -v #see all branches and last commit
- git merge <branch-name> #to merge the branch to master branch
- git branch -d <branch-name> #to delete a merged branch
- git branch -D <branch-name> #to delete a unmerged branch
- git push origin :<branch-name> #to delete the branch from github
- git branch -m <new-branch-name> #to change name of the branch
- git mergetool #default merge tool of git to resolve merge conflicts
- git rebase <branch-name-to-rebase-with> #rebase current branch with given branch

What next?

Practice makes a man perfect.

Manage your projects through github.

Github- an influential resume.

Github- your way to open-source.

Maintain your github profile and become a DEVELOPER.

Thank you

Contact me:

meghaaggarwal493@gmail.com