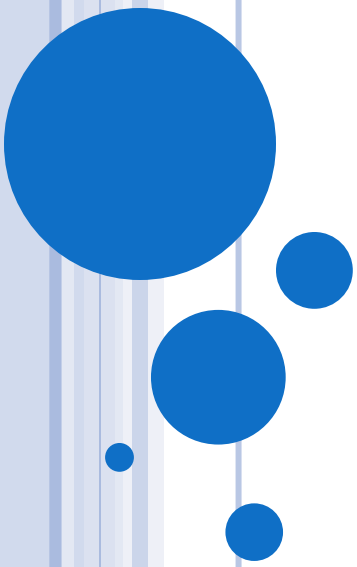
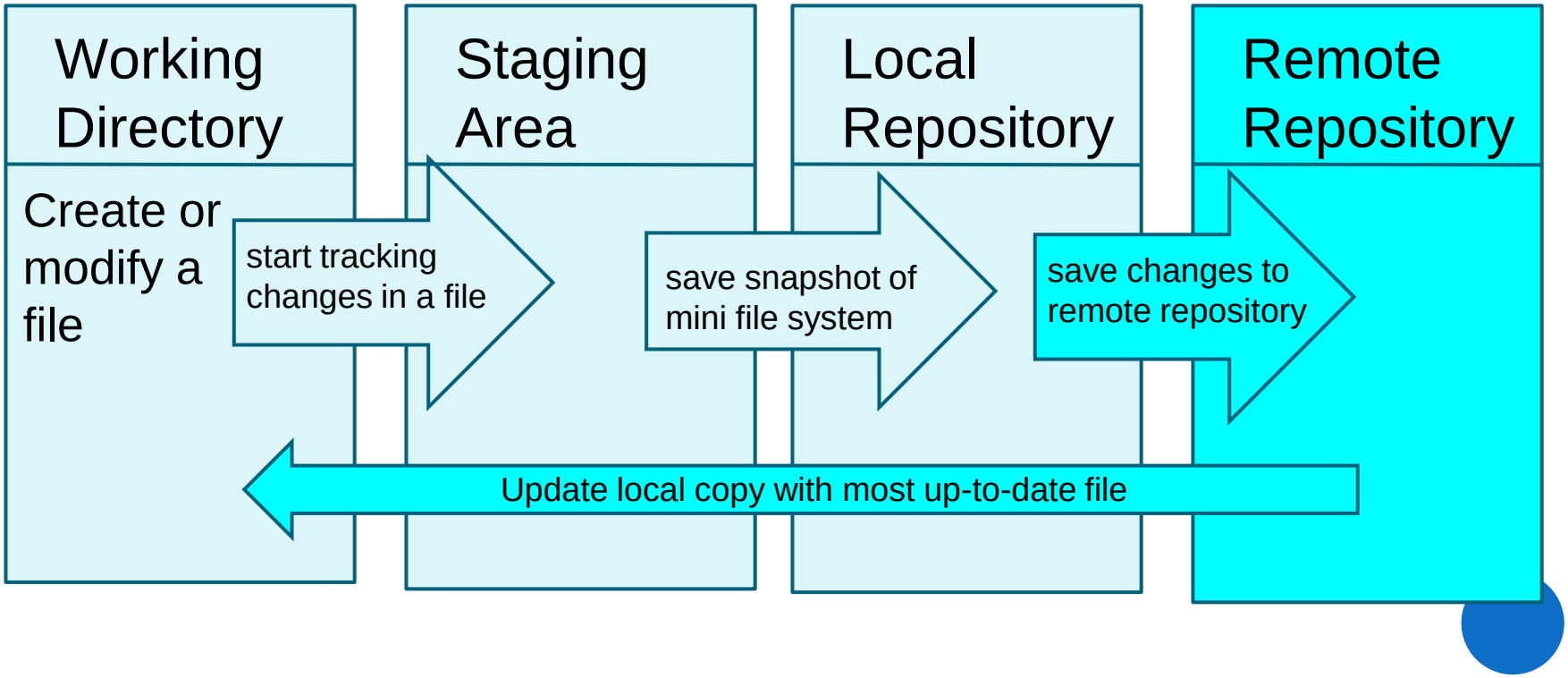


Remote Repositories

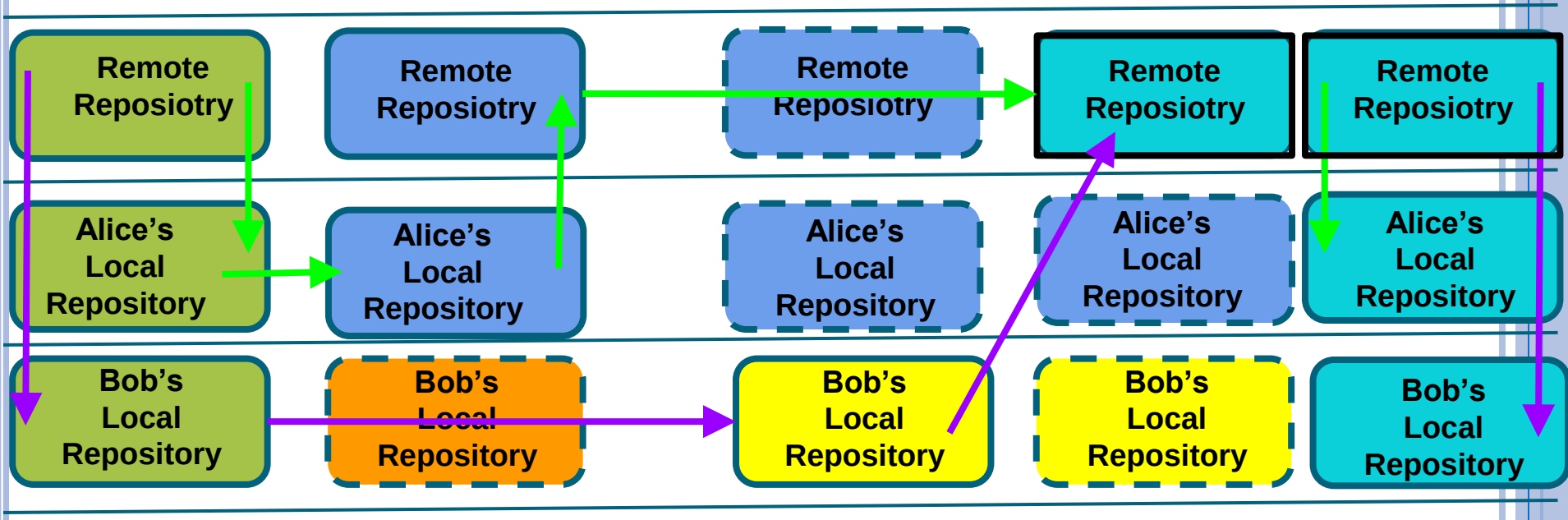
Find a partner

How? The Structure of Git

Remote repository - track changes of multiple users



Multi-Users Workflow



Time

Alice and Bob update their local repositories

Alice modifies her file and updates her local and the remote repository

Bob modifies his version of the file and updates his local repository.

Bob resolves any conflicts between his current version and the new version that Alice added to the remote repository

Alice and Bob update their local repositories

GITHUB

- create a github account
- Remote repository server
- Lots of good projects
- Easy to explore code

Cloning a repository

1. In github, click on the repository you want to clone.
2. Copy the url (make sure http is selected)
3. in git bash type:
   a. git clone <repo url>
4. This should have created a local copy of your repository

Exploring the cloned repository

- git remote –v
  - What does git call my remote repositories?
  - Default for cloned repositories: origin
- git branch
  - What branch am I on?
    - Default: master
- git push <remote> <branch>
  - (e.g. git push origin master)
  - You will have to enter your github username and password.

Exercise 1: Remote Repository

- Sign in to GitHub
- Click on the repositories tab
- Click the new botton (its green)
- Fill in repository name, description
- check "initialize this repository with a readme file“
- Clone the repository (using the https url) to your local machine

Your repository will always be public if you are using the free version of github

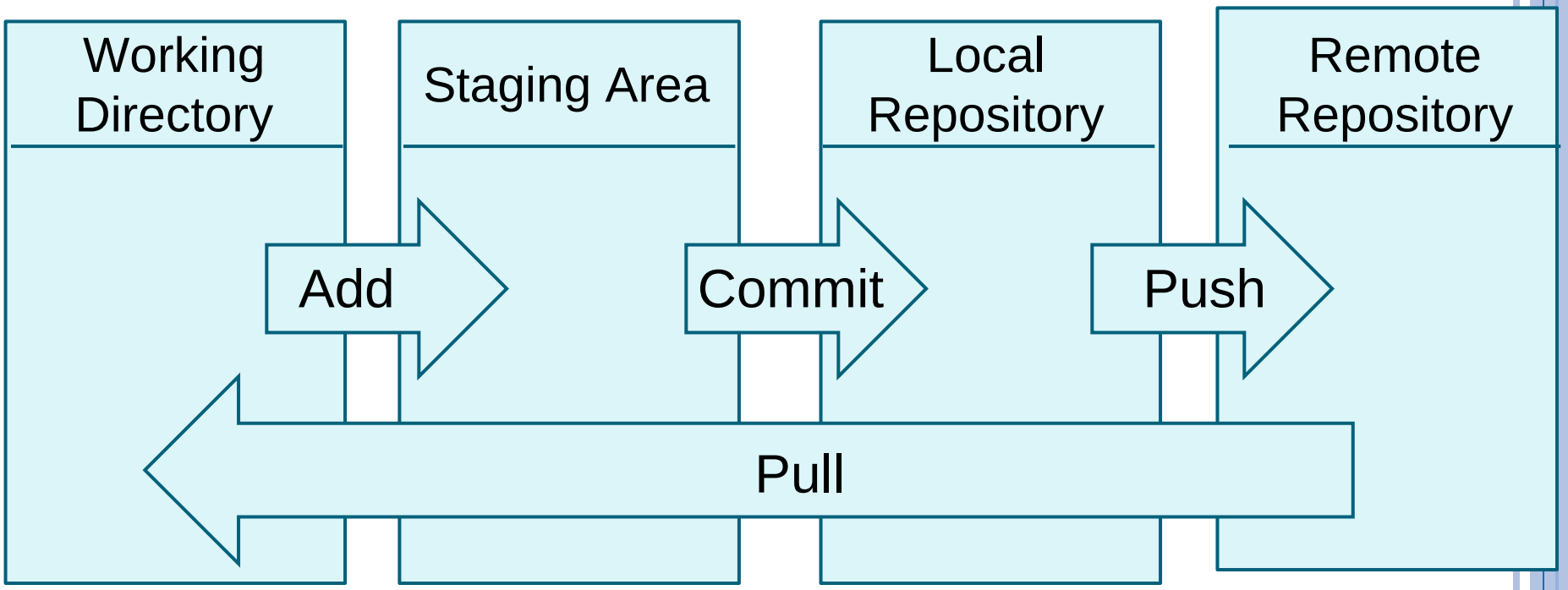
Exercise 2

- Make some changes to your local copy
- Push the changes to GitHub
- Verify the changes on the GitHub website

Fetching remote changes

- git fetch
  - Download changes from remote repository
- git merge
  - Merge changes into current branch
- git pull
  - git fetch + merge combined

Workflow:



Before starting work, you should always pull to make sure you are modifying the most up to date files

Final Excersize

- With a friend (or a small group) take turns:
- Designate one person to be the 'upstream' person, the others are the 'forkers‘
- Have the upstream make a repository on github
- Have the forkers fork the upstream's repository
- Everyone clones their own version of the repository
- Have forkers add upstream's repository as a remote
- Have upstream add a few lines to the README, commit and push the changes
- Have Forkers pull the changes, make changes of their own and commit them
- Have upstream make another change, commit and push
- Forkers pull this new change creating a conflict and resolve the conflict
- Trade off roles

Not covered (incomplete list)

remote branching

tags

Learn More:

http://git-scm.com/book