Using Git and GitHub

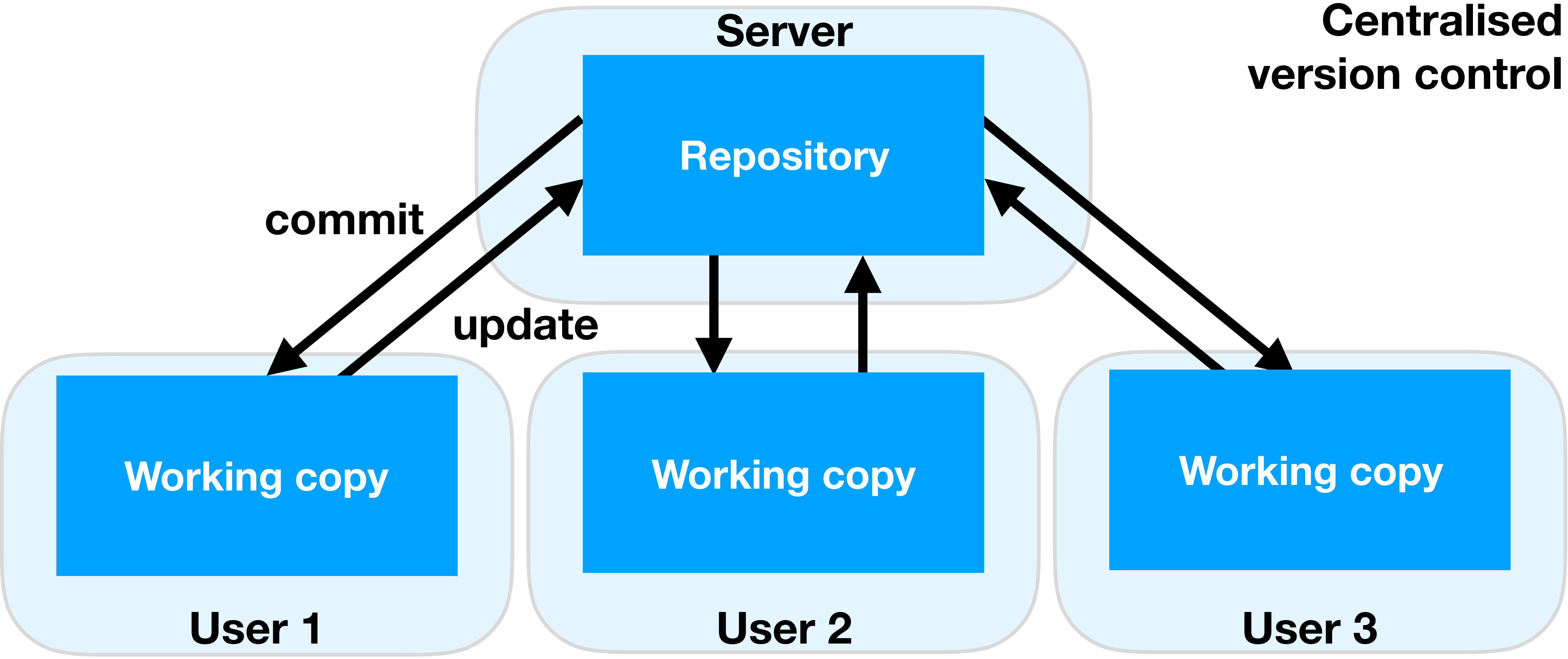
Cees Carels

Overview

- What is Git
- Git Commands
- Workflow on GitHub

Version Control

Examples
SVN, CVS

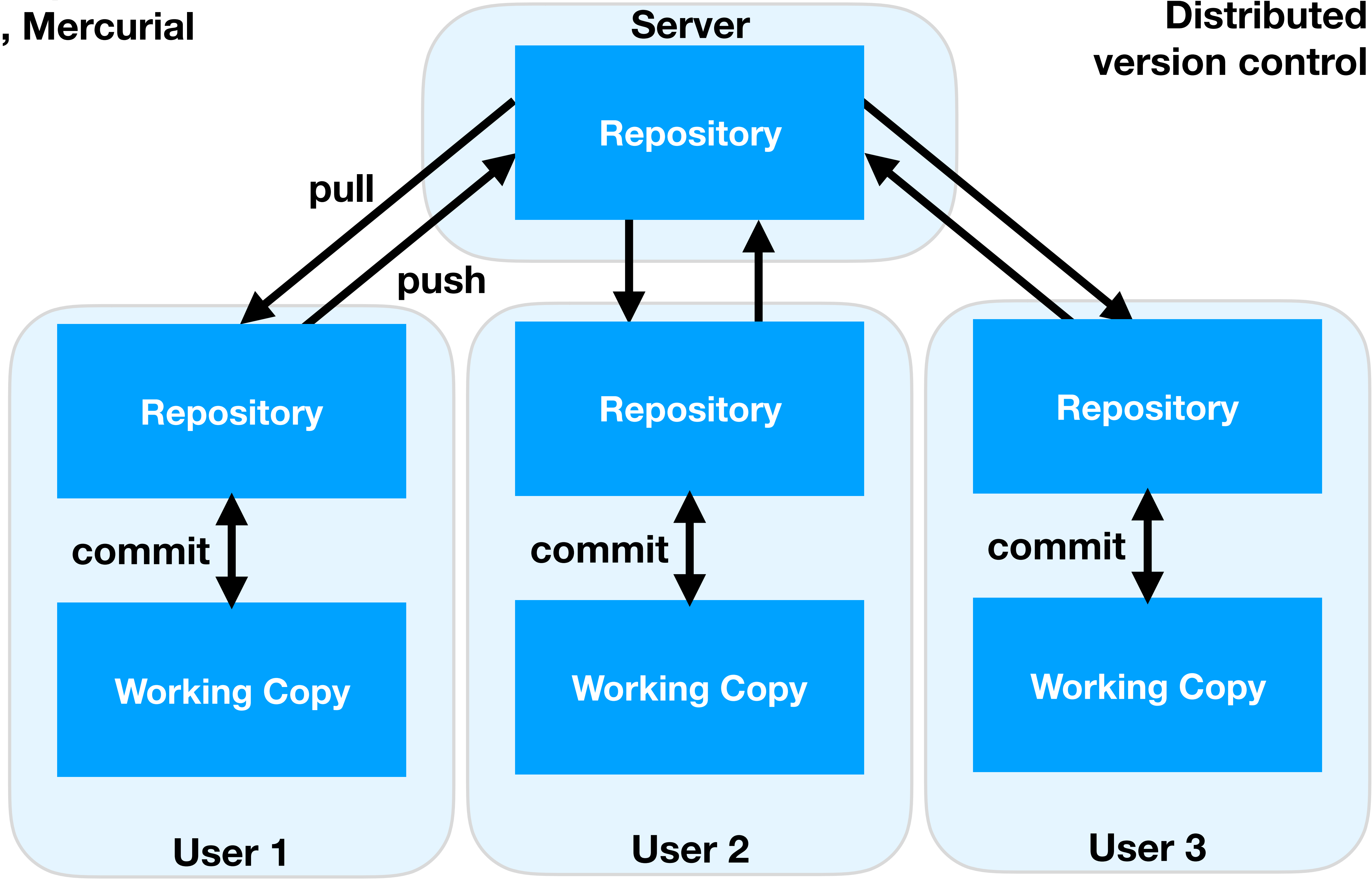


Centralised version control

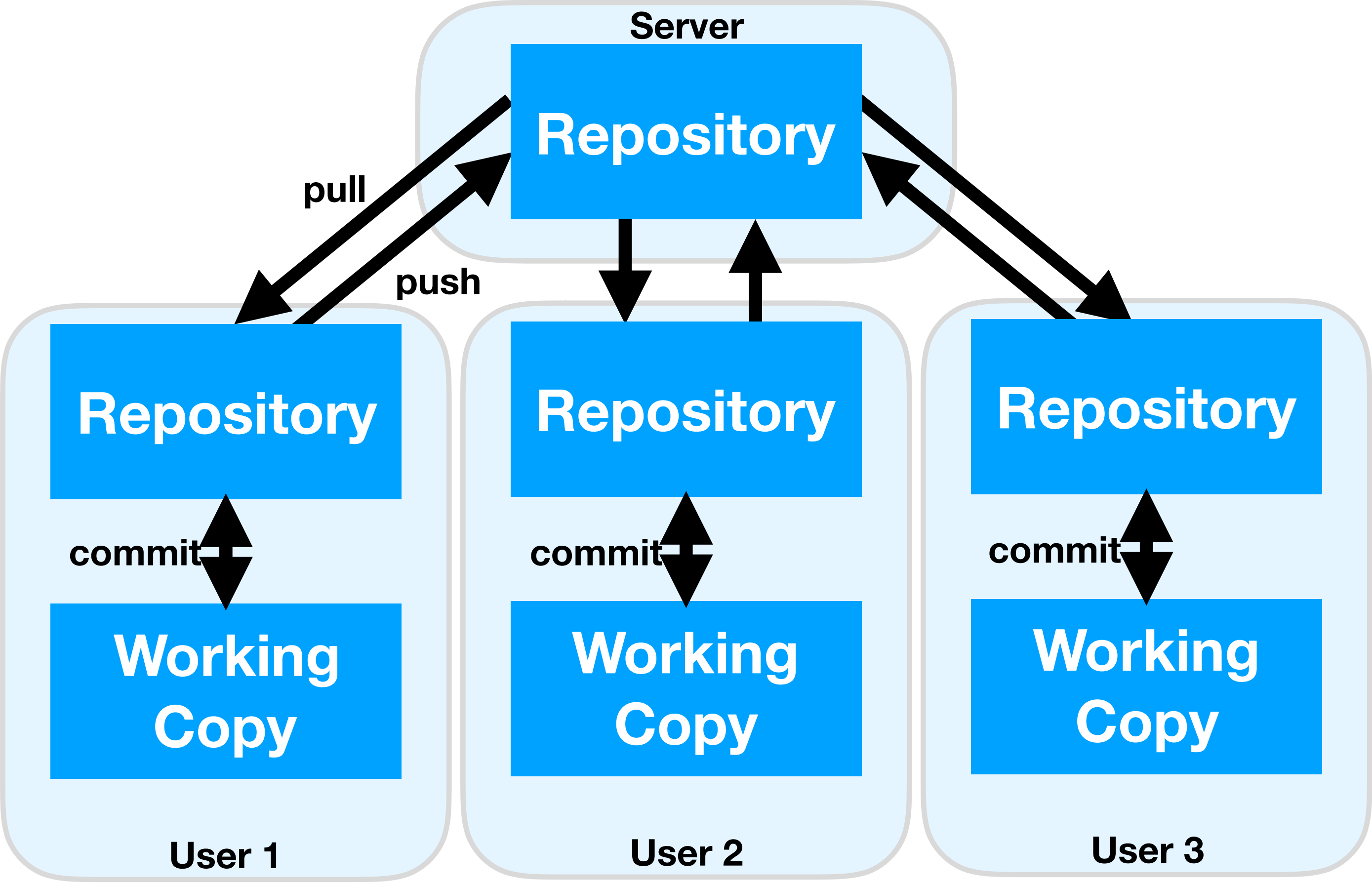
Version Control

<u>Examples</u>
Git, Mercurial

Distributed version control



Git Version Control



Useful command

- `git status`

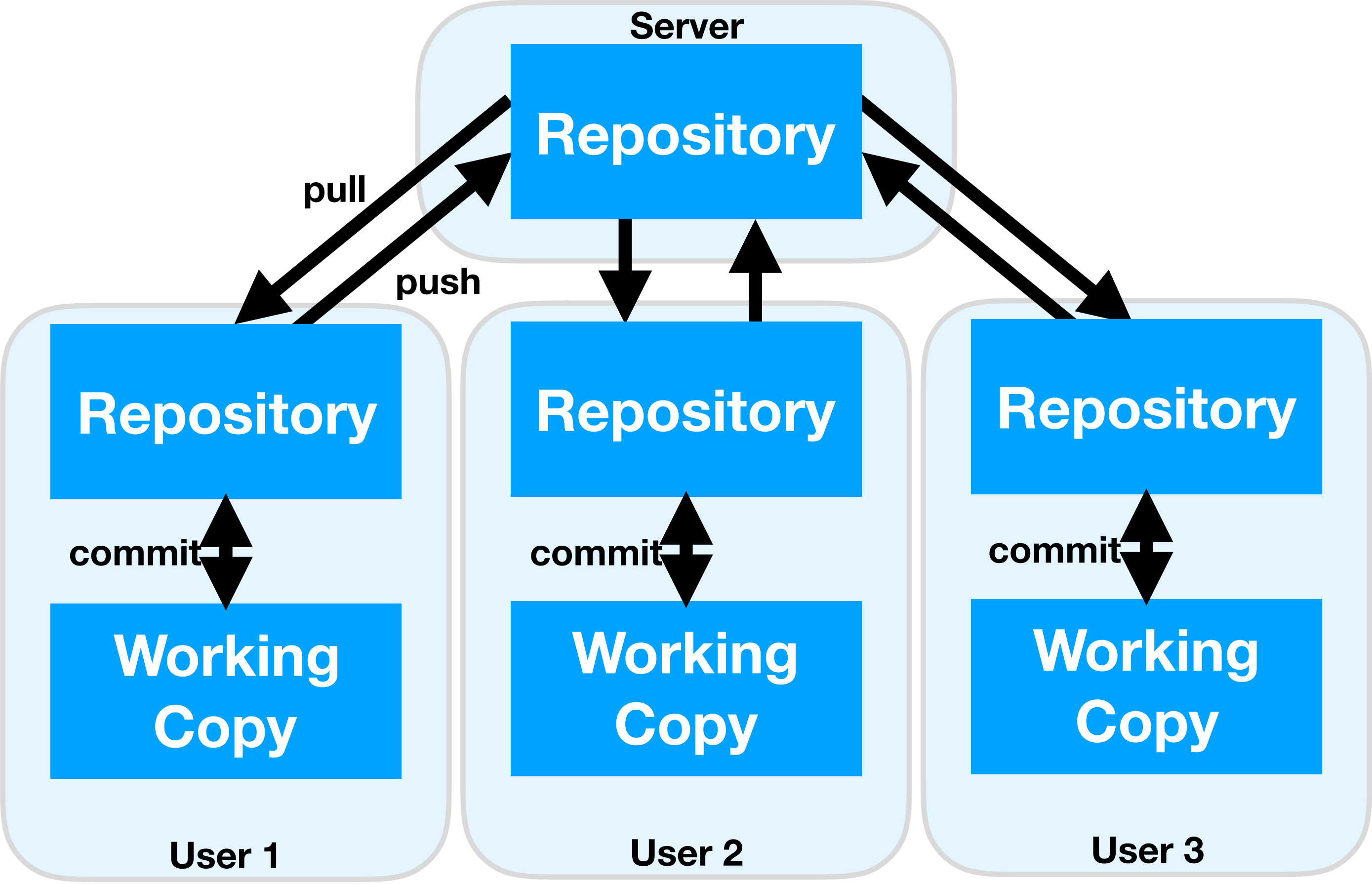
Before you start, always do `git status`

The status command shows the changes with respect to the last commit.

"Displays paths that have differences between the index file and the current HEAD commit, paths that have differences between the working tree and the index file, and paths in the working tree that are not tracked by Git (and are not ignored by gitignore[5]). The first are what you *would* commit by running `git commit`; the second and third are what you *could* commit by running *git add* before running `git commit`."

https://git-scm.com/docs/git-status

Git Version Control



- “git-pull - Fetch from and integrate with another repository or a local branch”
    - “Incorporates changes from a remote repository into the current branch. In its default mode, git pull is shorthand for git fetch followed by git merge FETCH_HEAD.”
- “git-push - Update remote refs along with associated objects”
    - “Updates remote refs using local refs, while sending objects necessary to complete the given refs.”

https://www.git-scm.com/docs/git-pull
https://git-scm.com/docs/git-push

Useful commands

- `git pull`
- `git push`

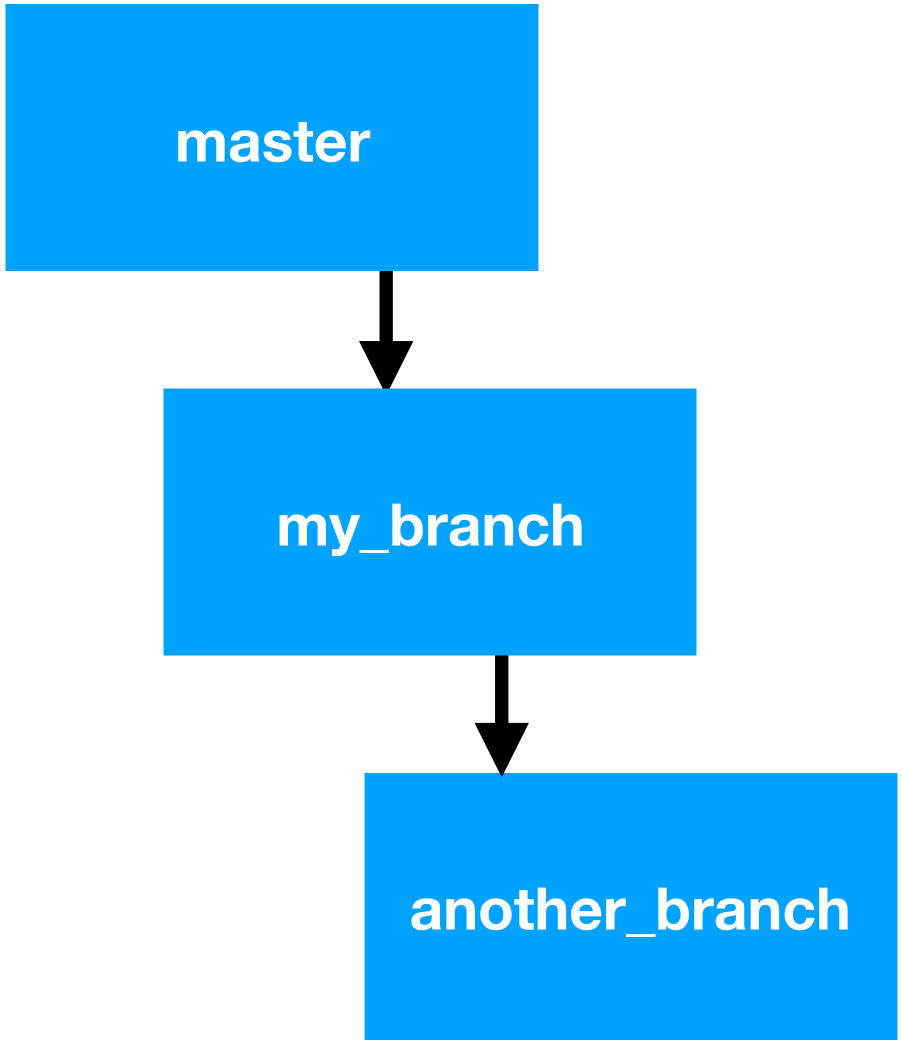
If the branch does not yet exist on the remote repository:

```
git push --set-upstream origin <branch_name>
```

Branching

Useful commands

- `git branch -a`
- `git branch`
- `git branch -b branch_name`
- `git branch -d branch_name`



Create a branch from another branch

```
$ git checkout master
```

(switch to master branch)

```
$ git checkout -b my_branch master
```

(create a branch called my_branch from master and switch to it)

```
$ git checkout -b another_branch my_branch
```

(create a branch called another_branch from my_branch and switch to it) https://git-scm.com/docs/git-commit

Committing

Useful commands

- `git rm file`
- `git add file`
- `git mv file1 file2`

- After making changes
  - `git commit -m “commit message”`
  - Or do:
  - `git commit -am “commit message”`
  - Make the commit message meaningful for the sake of others and posterity.

“by using the -a switch with the *commit* command to automatically "add" changes from all known files (i.e. all files that are already listed in the index) and to automatically "rm" files in the index that have been removed from the working tree, and then perform the actual commit;”

Configuring Git

These fields do not have to match your GitHub account, but it is useful if they are consistent.

- `git config --global user.name “yourusername”`
- `git config --global user.email “yourmail@yourserver”`
- `git config --global core.editor emacs`

Editor of your choice: e.g. vi, emacs,…
The default option is vi

Tips

- **Do:**
  - Use `git pull origin` to get the latest changes from the remote
  - Use `git status` to check what changes have been made
  - Name your branches sensibly
  - Commit and push regularly
  - Compare your changes carefully when making a pull request
- **Do Not:**
  - Attempt to circumvent Git by copy/pasting files external to the repository into the repository.
  - Work on someone else's branch
  - Work on protected branches

Tips

- Undoing changes
    - In general, if you committed (and pushed) a small amount of work on the wrong branch, it might be easier to start again and commit on a different branch
    - See here for a guide on how to undo changes: https://sethrobertson.github.io/GitFixUm/fixup.html
    - Try to avoid making mistakes in the first place by using **`git status`** before you start
- Be especially careful if you are an owner or administrator and have access to the protected master branch. You don't want to accidentally revert/change the master branch to a bad state!

Git Workflow

- A sample workflow is documented here
    - See Repository Conventions
    - https://github.com/ccarels/Introduction-to-Git-Presentation/blob/master/CONTRIBUTING.md