VERSION CONTROL WITH GIT AND GITHUB

WHAT'S THE PLAN FOR TODAY?

1. Version Control - Concepts, Terminology
2. Git – Commands
3. Collaborating with Git using GitHub
4. Version Control practices and workflow in the industry.

HOW DO YOU WORK WITH OTHER DEVELOPERS?

Work in a team, probably on particular components ?

Integrate your code together.

Make copies of your files in case something you lose them. (Not really?)

HOW DO YOU WORK WITH OTHER DEVELOPERS?

Work in a team, probably on particular components ?

Integrate your code together.

Make copies of your files in case something you lose them. (Not really?)

SO WHAT'S VERSION CONTROL?

- Version control is the management of changes to documents, primarily computer programs.
- Also known as revision control or source control.
- Examples: git, mercurial, subversion

WHY VERSION CONTROL?

Makes working in a team easy!

- *Code without interference.*
- *Go back to a previous version (iOS 10 anyone?)*
- *Integrate code of multiple developer's easily.*
- *Know who did what, when.*

Keep your code **secure**.

DO YOU REALLY NEED VERSION CONTROL?

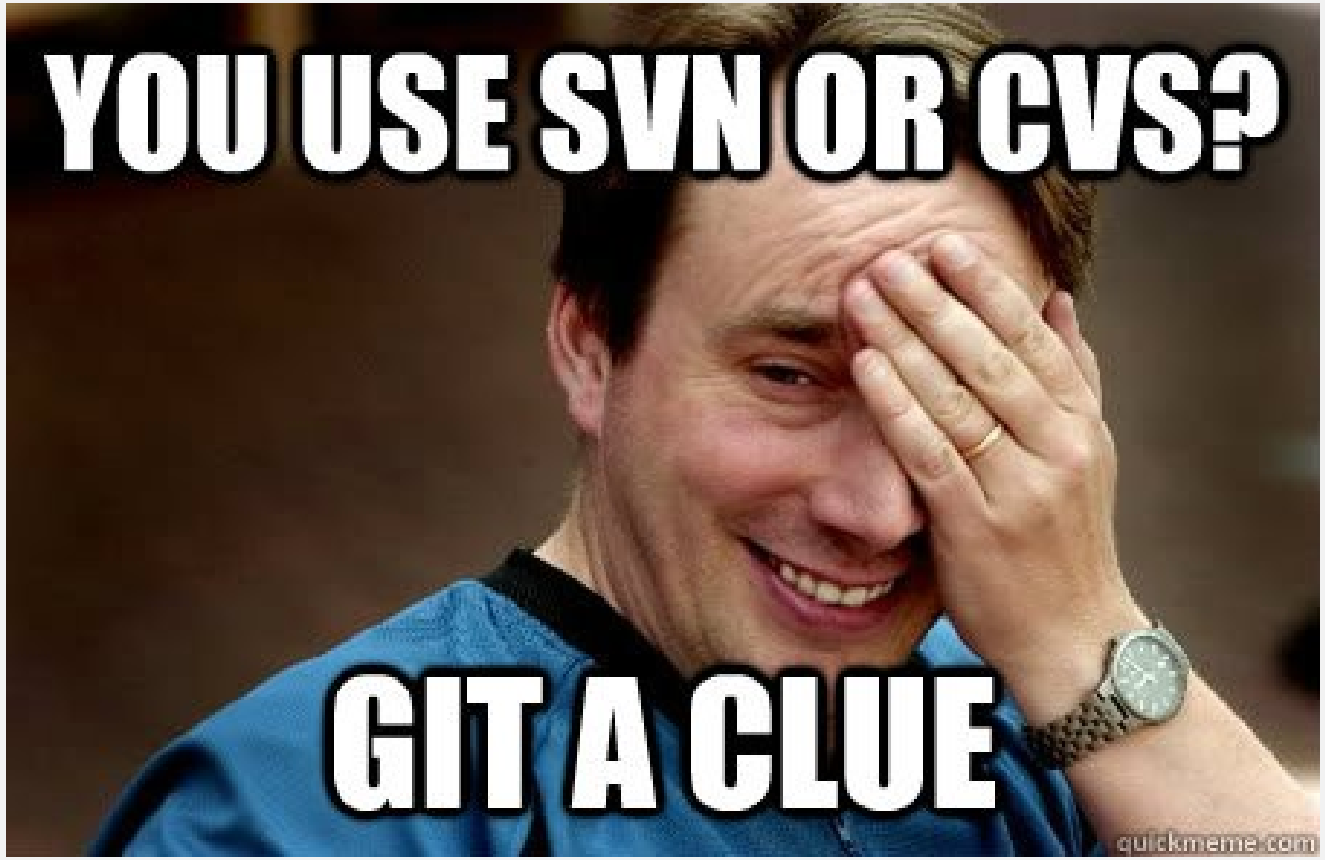
DO YOU REALLY NEED VERSION CONTROL?

GIT

- A distributed version control system
- Command-Line Tool (accessible with Terminal on the Mac or Git Bash on Windows)

WHY GIT AND NOT OTHER VCS?

- Git's the most popular version control system in the industry, by far.
- A proper and detailed understanding of Git will allow you to make a transition to any other distributed VCS easily.
- Most popular VCS are similar to Git



INSTALLING GIT

Things you'll need:

1. You need Git installed on your system, and you can access it in a UNIX Terminal, either the Terminal on the Mac or Git Bash on Windows.
2. Download Git from the following link:

   https://git-scm.com/downloads

VERSION CONTROL TERMINOLOGY


Version Control System (VCS) or (SCM)


Repository


Commit


SHA


Working Directory


Checkout


Staging Area/Index


Branch

VERSION CONTROL TERMINOLOGY

1. Version Control System :

A VCS allows you to: revert files back to a previous state, revert the entire project back to a previous state, review changes made over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more.

2. Repository:

A directory that contains your project work which are used to communicate with Git. Repositories can exist either locally on your computer or as a remote copy on another computer.

VERSION CONTROL TERMINOLOGY

3. Commit

Git thinks of its data like a set of snapshots of a mini file system.

Think of it as a save point during a video game.

4. SHA

A SHA is basically an ID number for each commit.

Ex. E2adf8ae3e2e4ed40add75cc44cf9d0a869afeb6

5. Working Directory

The files that you see in your computer's file system. When you open your project files up on a code editor, you're working with files in the Working Directory.

VERSION CONTROL TERMINOLOGY

6. Checkout

When content in the repository has been copied to the Working Directory. It is possible to checkout many things from a repository; a file, a commit, a branch, etc.

7. Staging Area

You can think of the staging area as a prep table where Git will take the next commit. Files on the Staging Index are poised to be added to the repository.

8. Branch

A branch is when a new line of development is created that diverges from the main line of development. This alternative line of development can continue without altering the main line.

LET’S DIVE INTO THE GOOD STUFF NOW

GIT COMMANDS:

(THINGS WE’LL COVER)

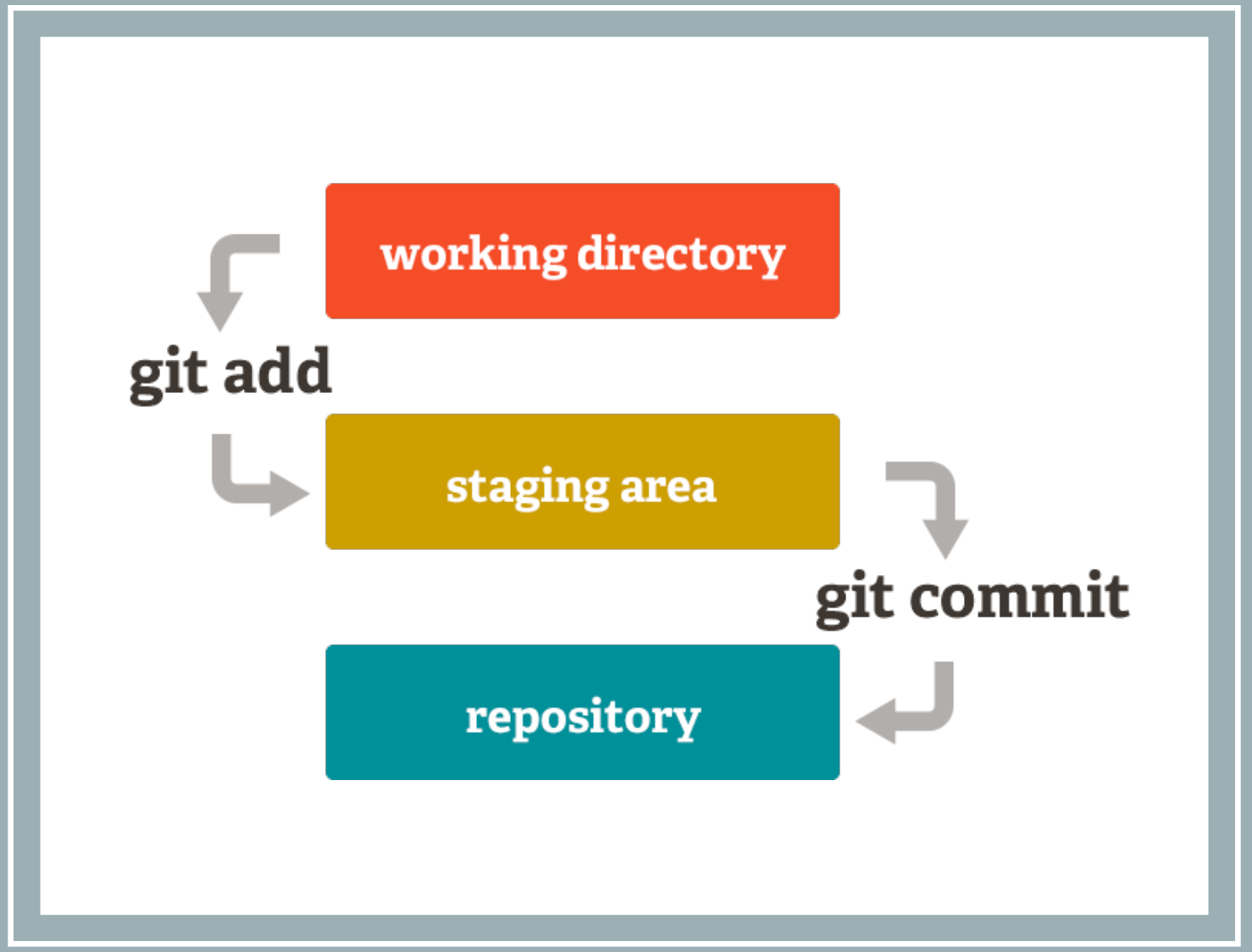
- BASIC GIT COMMANDS
- REMOTE REPOSITORY – PUSH & PULL
- BRANCHING, TAGGING AND MERGING
- DISASTER HAS STRUCK!

BASIC GIT COMMANDS

- git init
- git status
- git add <filename> / git add .
- git commit / git commit –m “commit message”
- git log –oneline / git log —stat
- git clone
- Other commands like git show, git ignore, git diff etc.

DEMO

BASIC GIT MODEL LOCALLY



BASIC GIT COMMANDS

- git init – *Initialize a Git repository/working directory*
- git status – *Status of your working directory*
- git add <filename> or git add . *(for all files in your working directory)*
- git commit – *Stash changes in your working directory*
- git log –oneline – *View your commit history*
- git clone – *Create an identical copy*

GITHUB

- It’s a hosting medium/website for your Git repositories
- Offers powerful collaborative abilities
- A good indicator of what you code/how much you code/quality of your code

WORKING WITH A REMOTE REPOSITORY

- Remote?

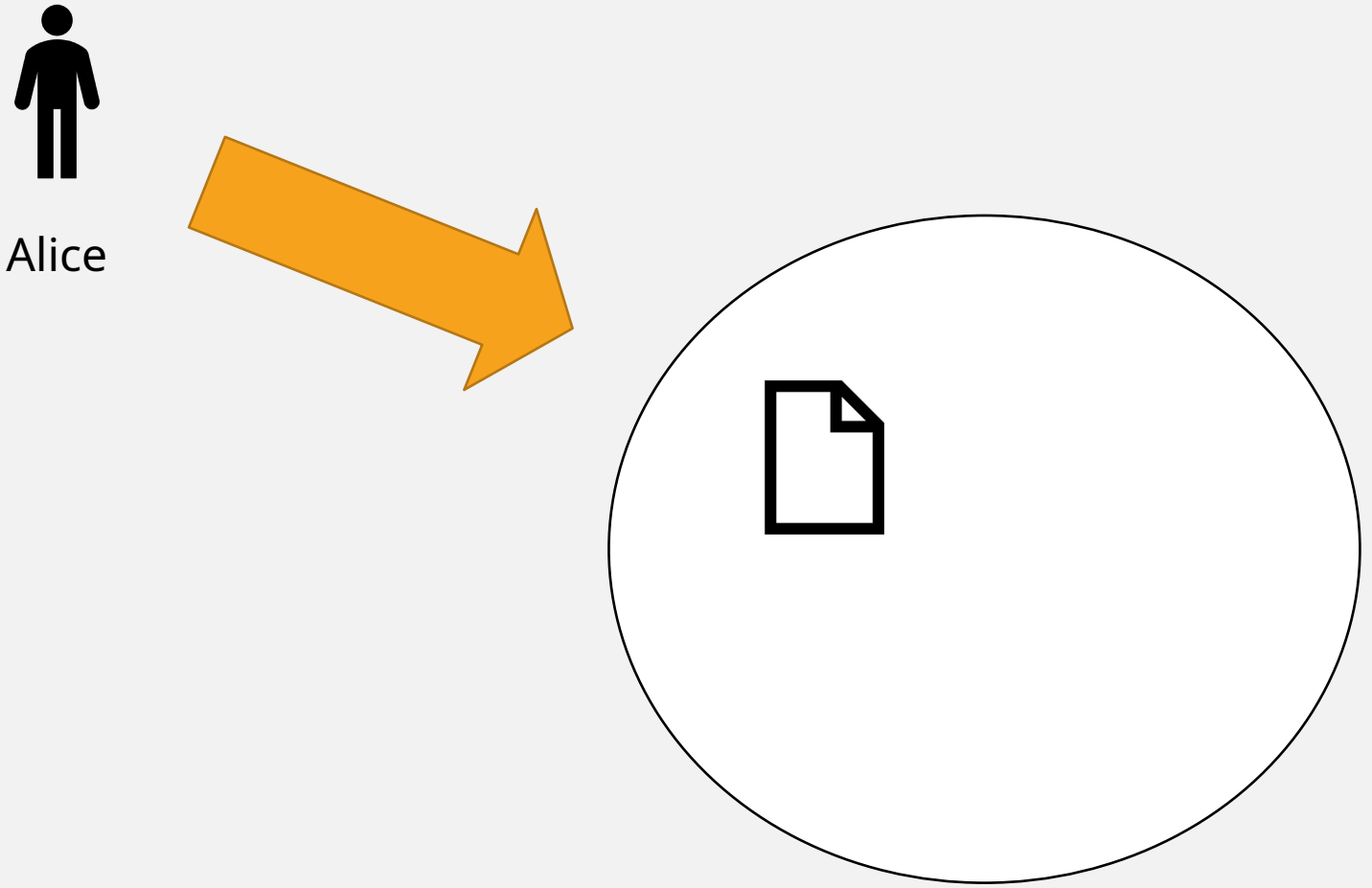
It’s the place where your code is stored.

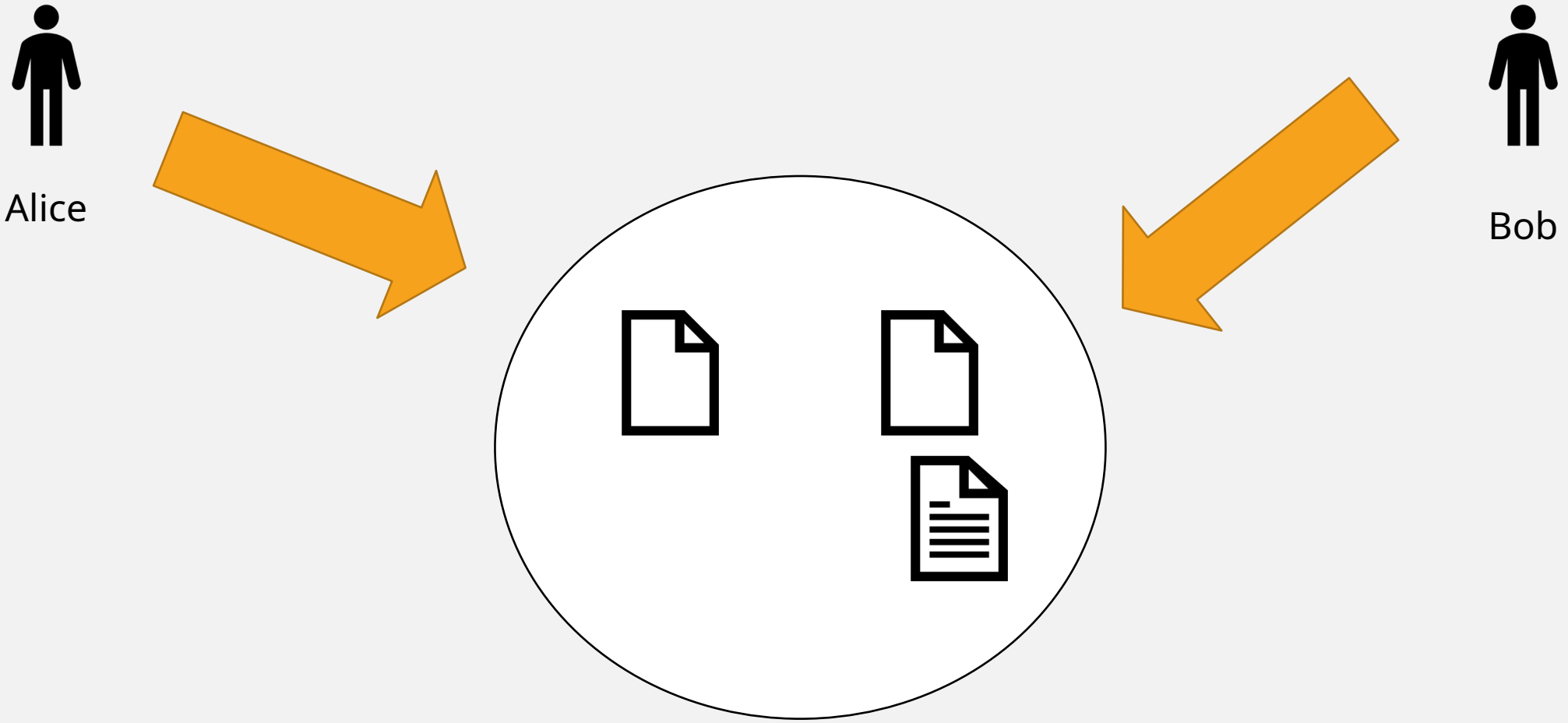
By default, remote name is origin and default branch is master.

- Certain things that come to play, namely collaboration.

How are we going to handle that with Git.

So here comes, push, pull, branching, merging, **forking**.


Alice



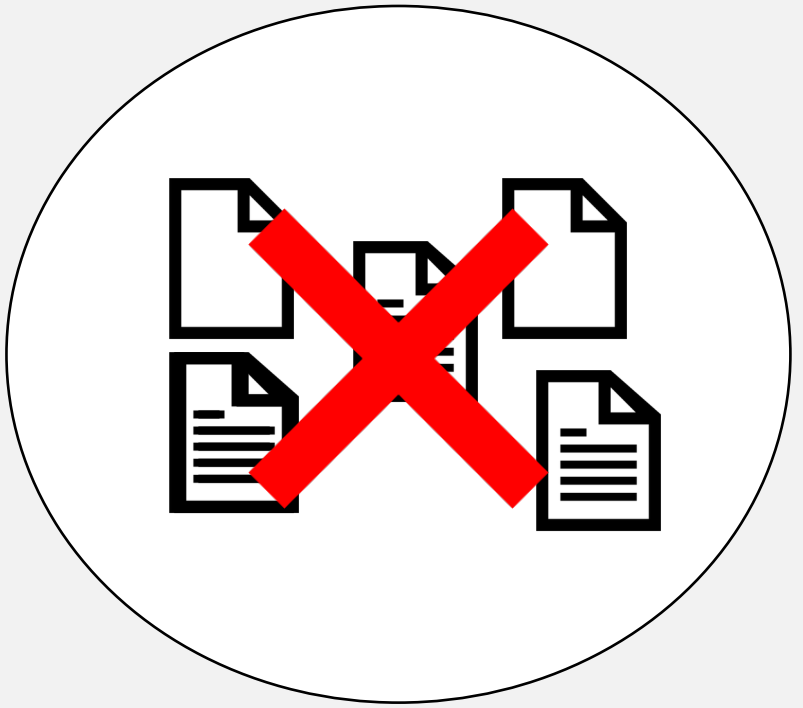
DeepCS

DeepCS

Alice

Joe

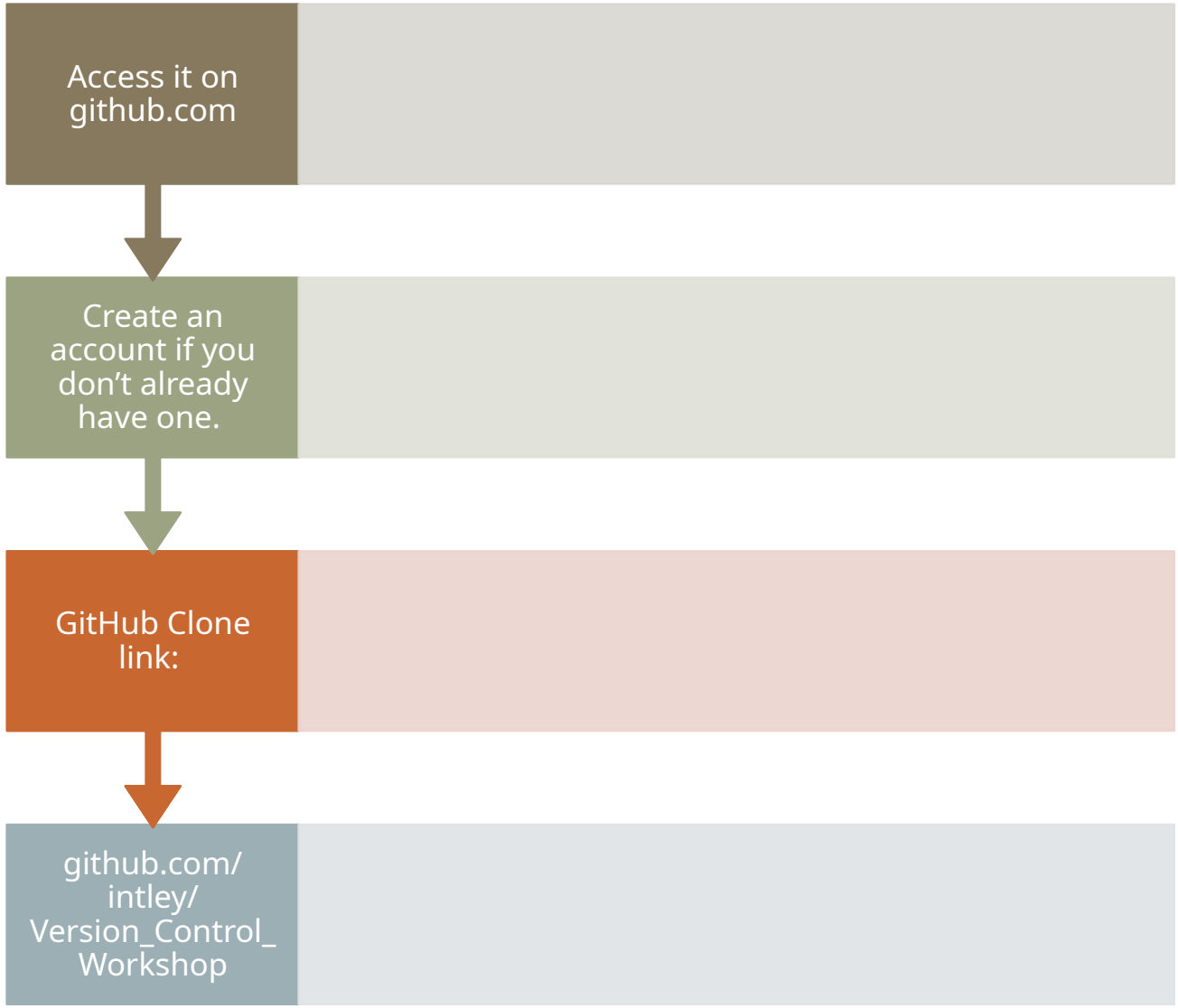
Bob



Who replaced the files? When ?

DeepCS

HOW TO ACCESS GITHUB



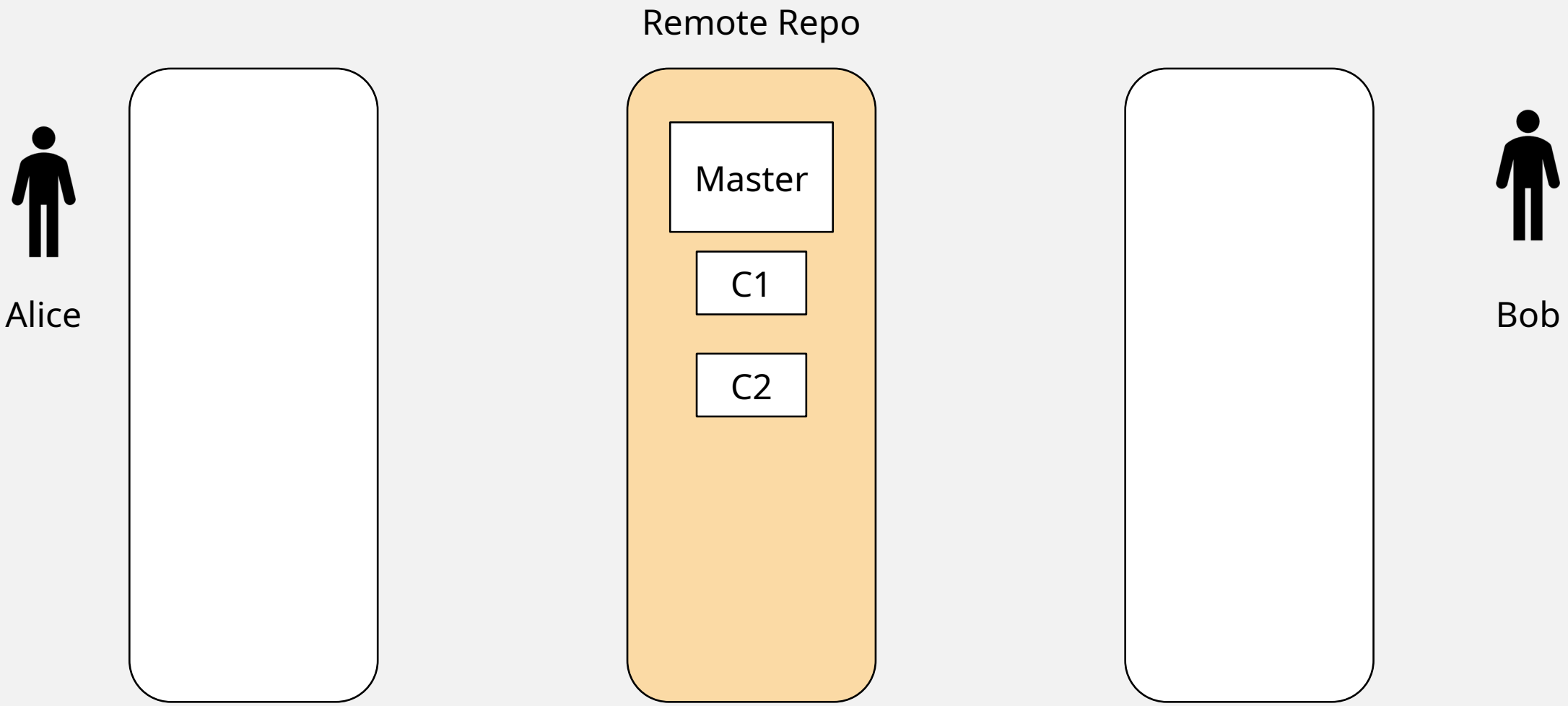
GITHUB DEMO

MORE GIT COMMANDS

- git push – push your changes into the remote repository
- git pull – pull your latest changes from the remote repository
- git branch – view branches in your repository
- git branch <branchname> - create a branch
- git checkout <branchname> - move to that branch
- git merge <branchname> - merge into that branch
- git revert <commit sha>

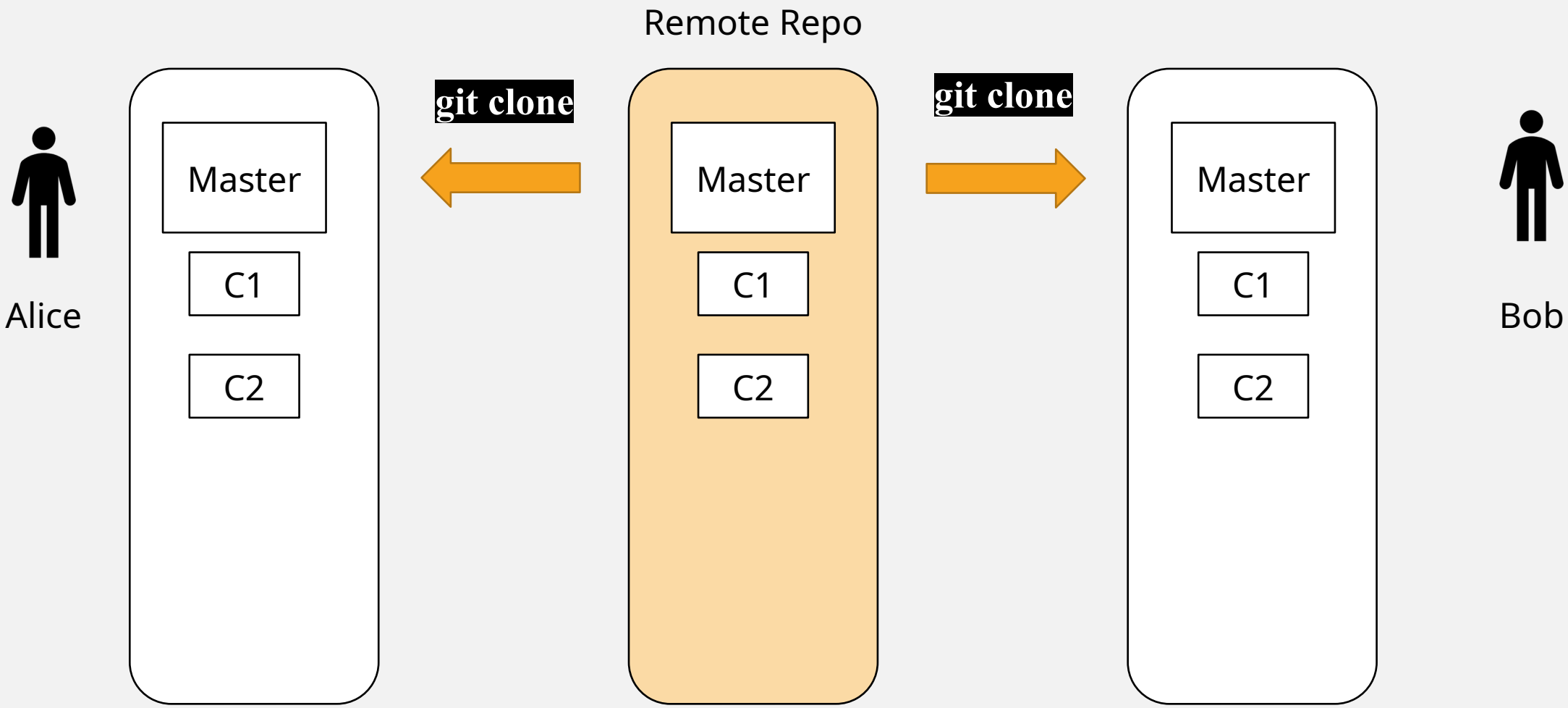
DEMO

COLLABORATION WITH GITHUB



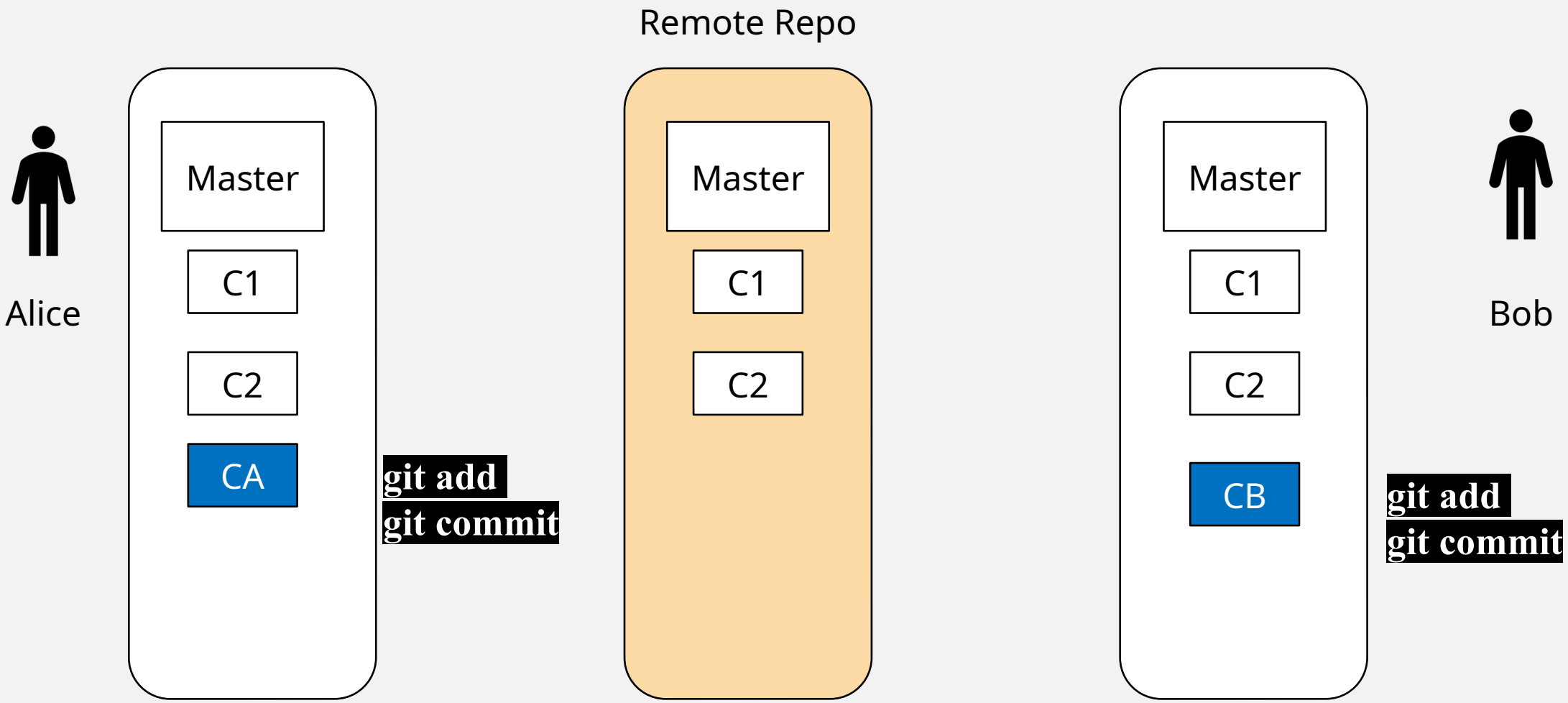
DeepCS

COLLABORATE



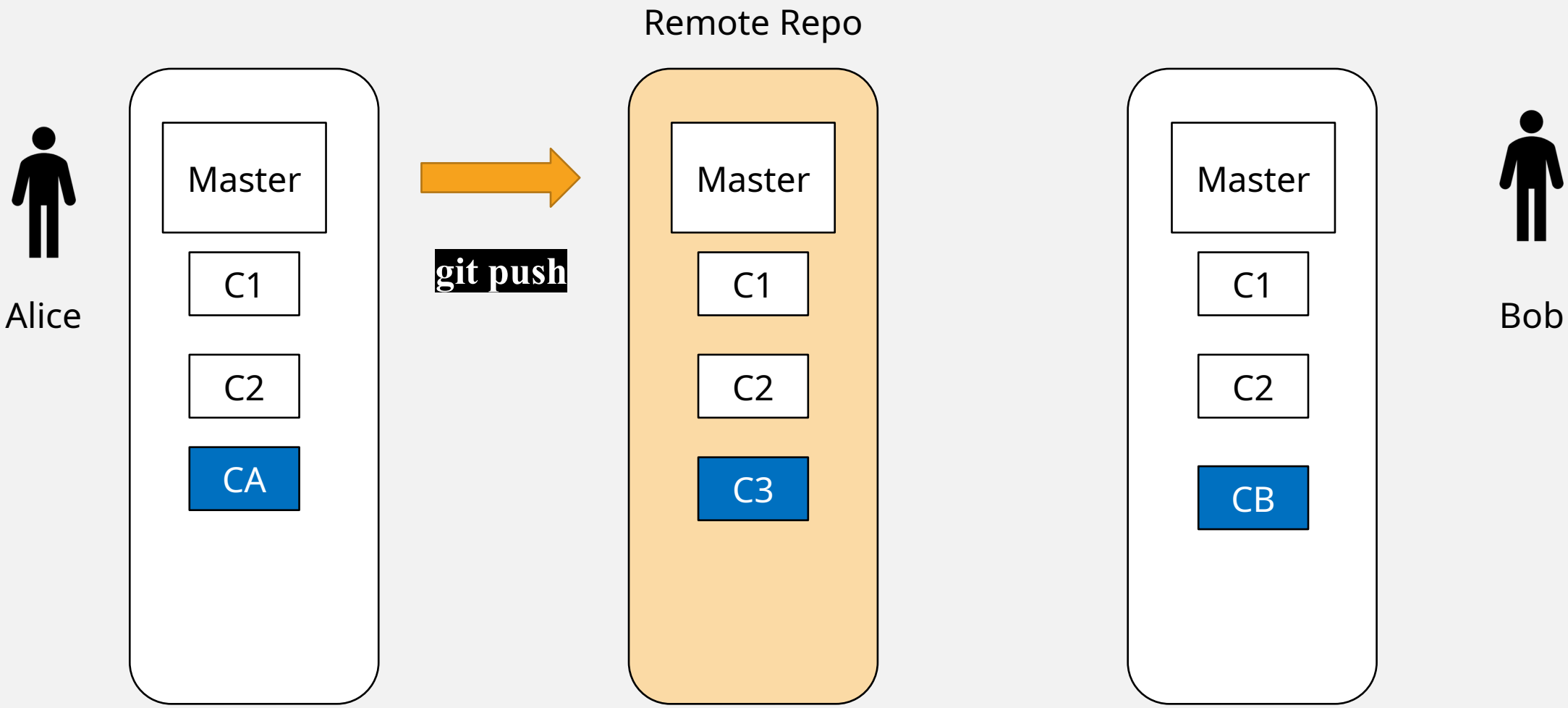
DeepCS

COLLABORATE



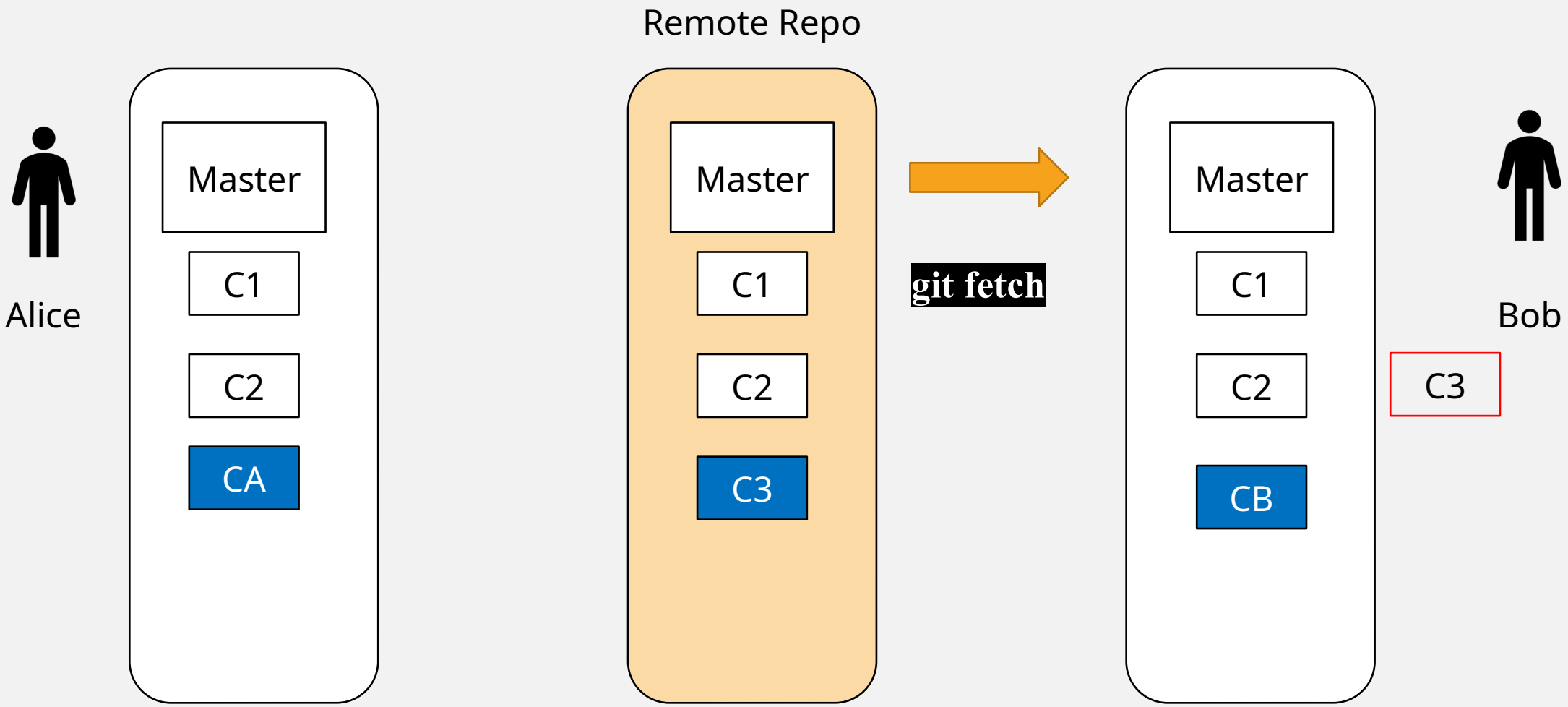
DeepCS

COLLABORATE



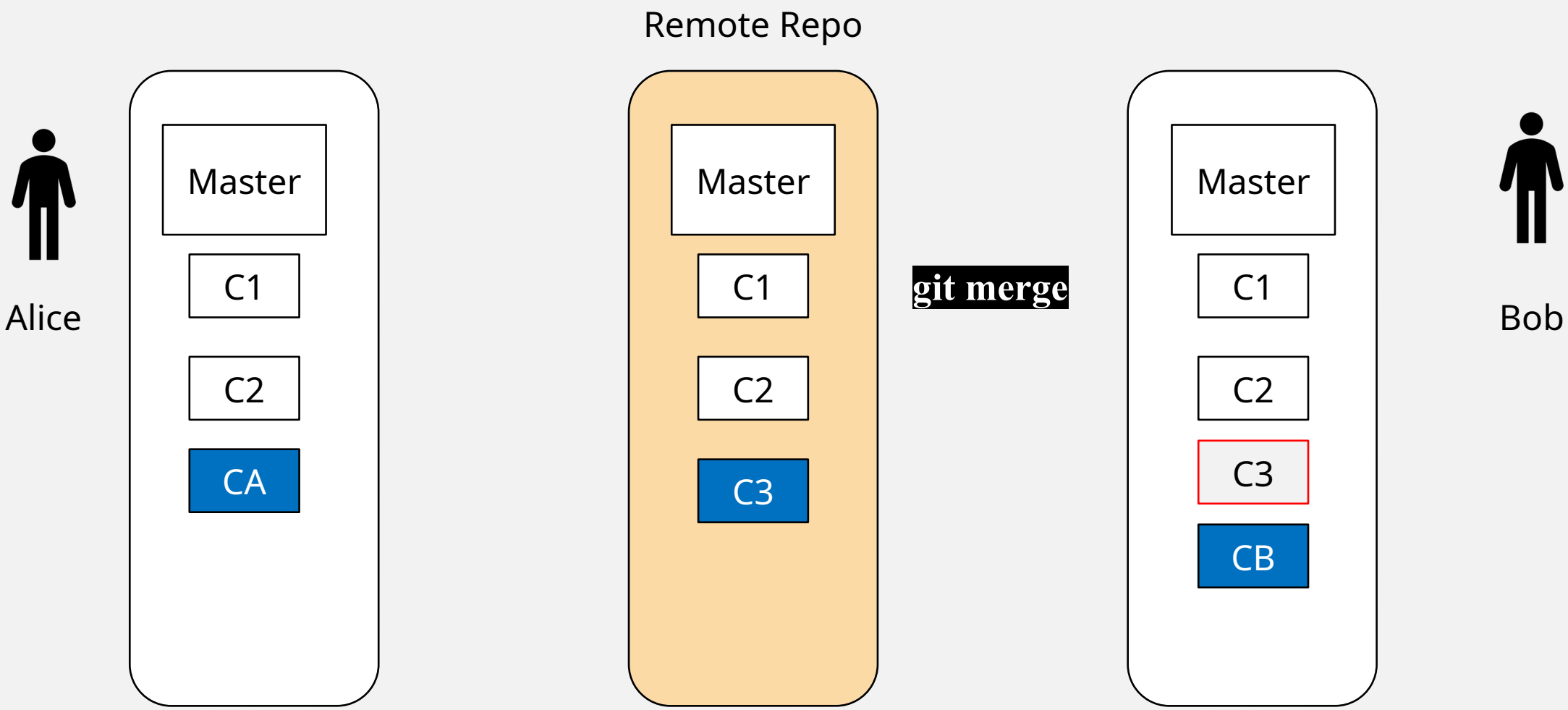
DeepCS

COLLABORATE



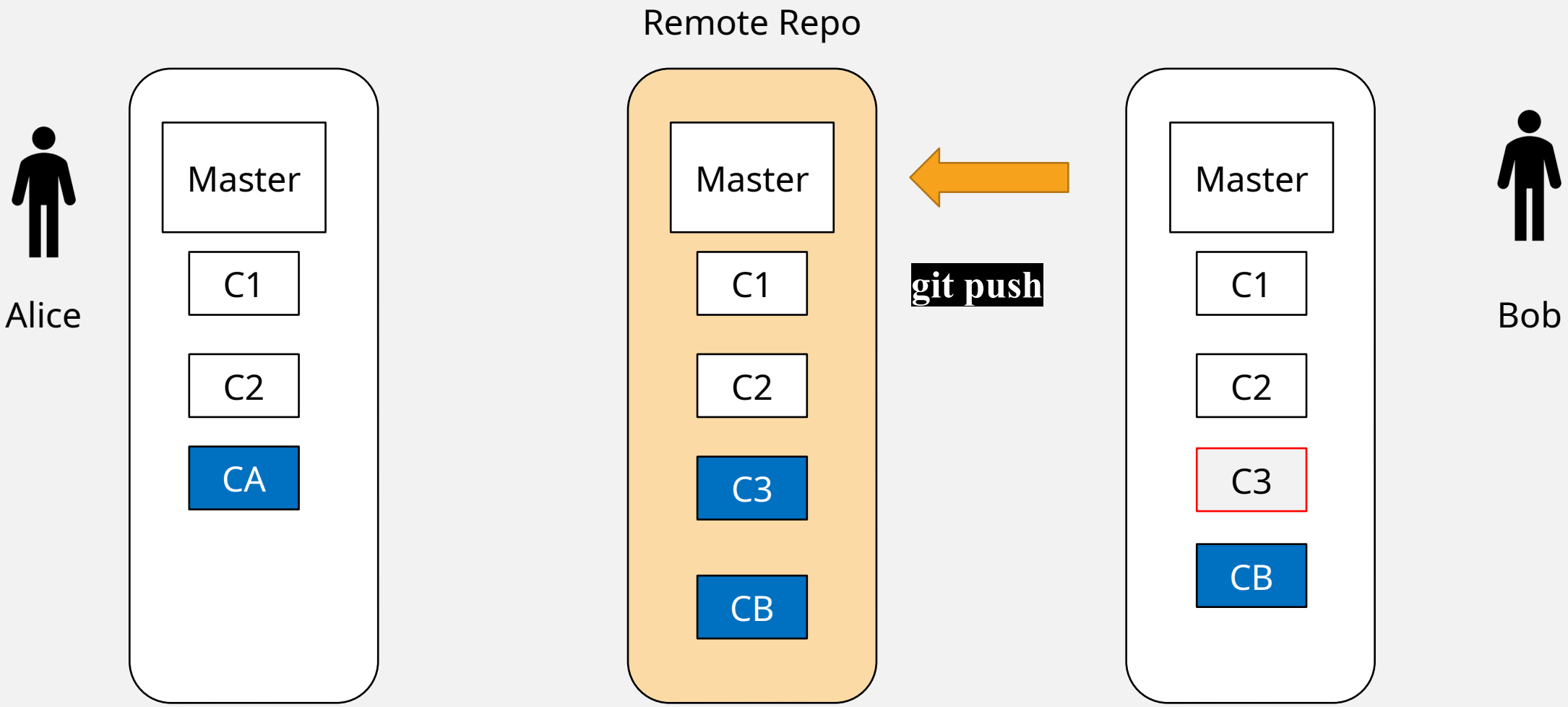
DeepCS

COLLABORATE



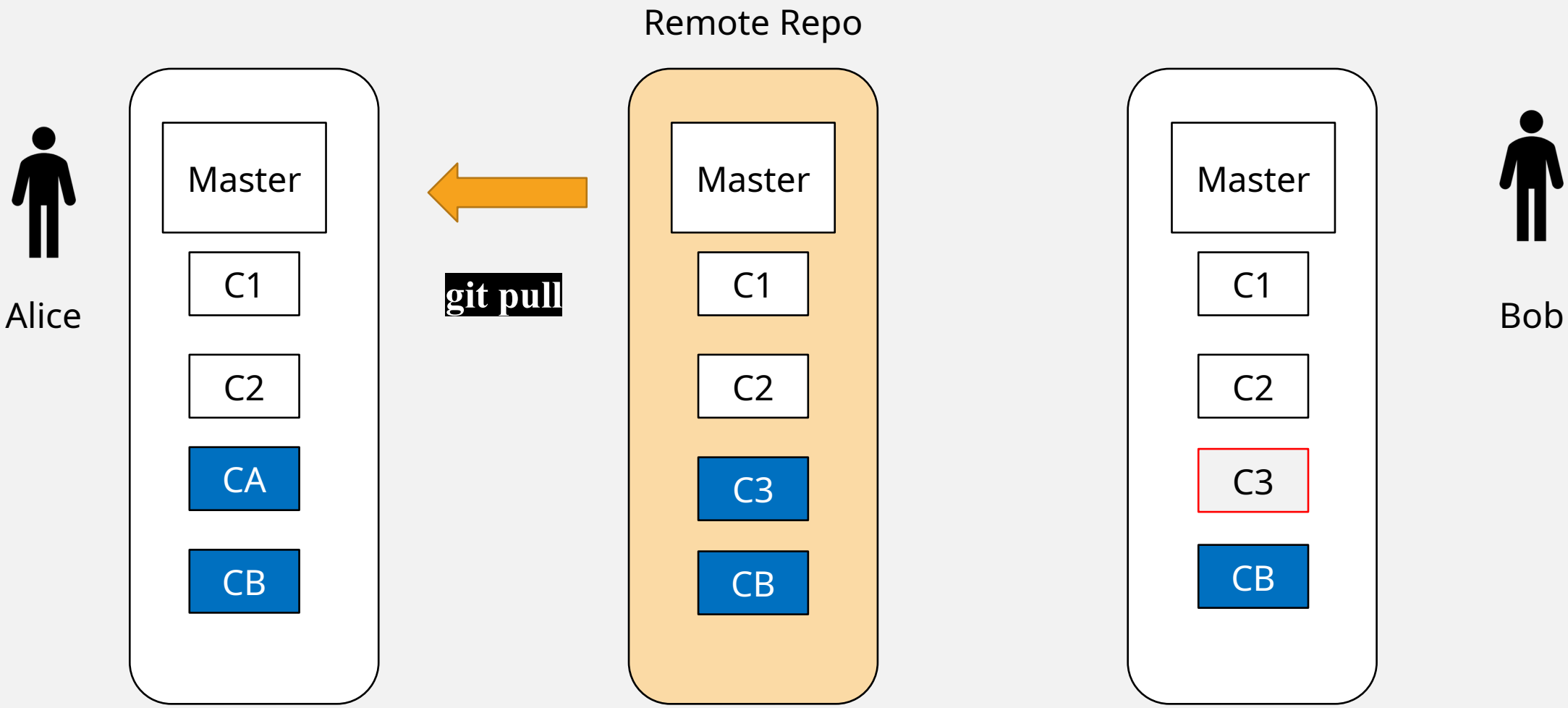
DeepCS

COLLABORATE



DeepCS

COLLABORATE



DeepCS

MORE ON COLLABORATIVE GIT COMMANDS

- Deleting a branch

git branch –d <branchname>

- Merge conflicts! How do you resolve?
- Pull requests ?
- git tag –a v1.0 / git tag –a v1.0 <commit sha>

DEALING WITH MERGE CONFLICTS

- Two typical cases of merge conflicts

1. Normal merge where you’re a collaborator
2. Pull request (handled by the repository owner)

MORE TO LEARN IN GIT/FORWARD STEPS

- Firstly, I highly recommended building a project from scratch with Git integrated and if possible, with other programmers.
- As far as Git goes, this PowerPoint covers the basics to help you get started, but as you work with version control, you will encounter new commands required.
- Rebasing – concept you could cover.

Credit to DeepCS and the CS department.