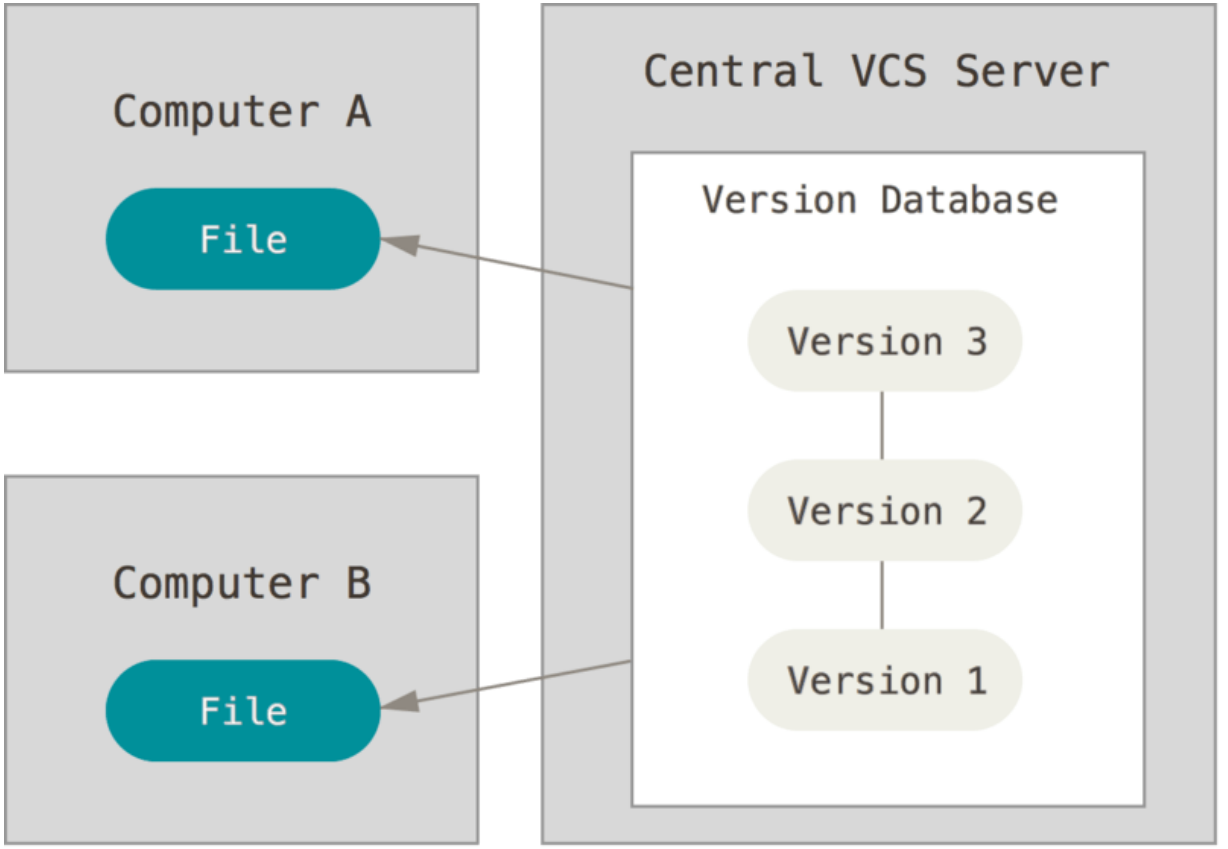

# What is Version Control?

- Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.
- Ability to have as many developers working on the same code base.
- Revert back your files incase there is some problem

- **Centralized Version Control Systems**
- **Distributed Version Control Systems**

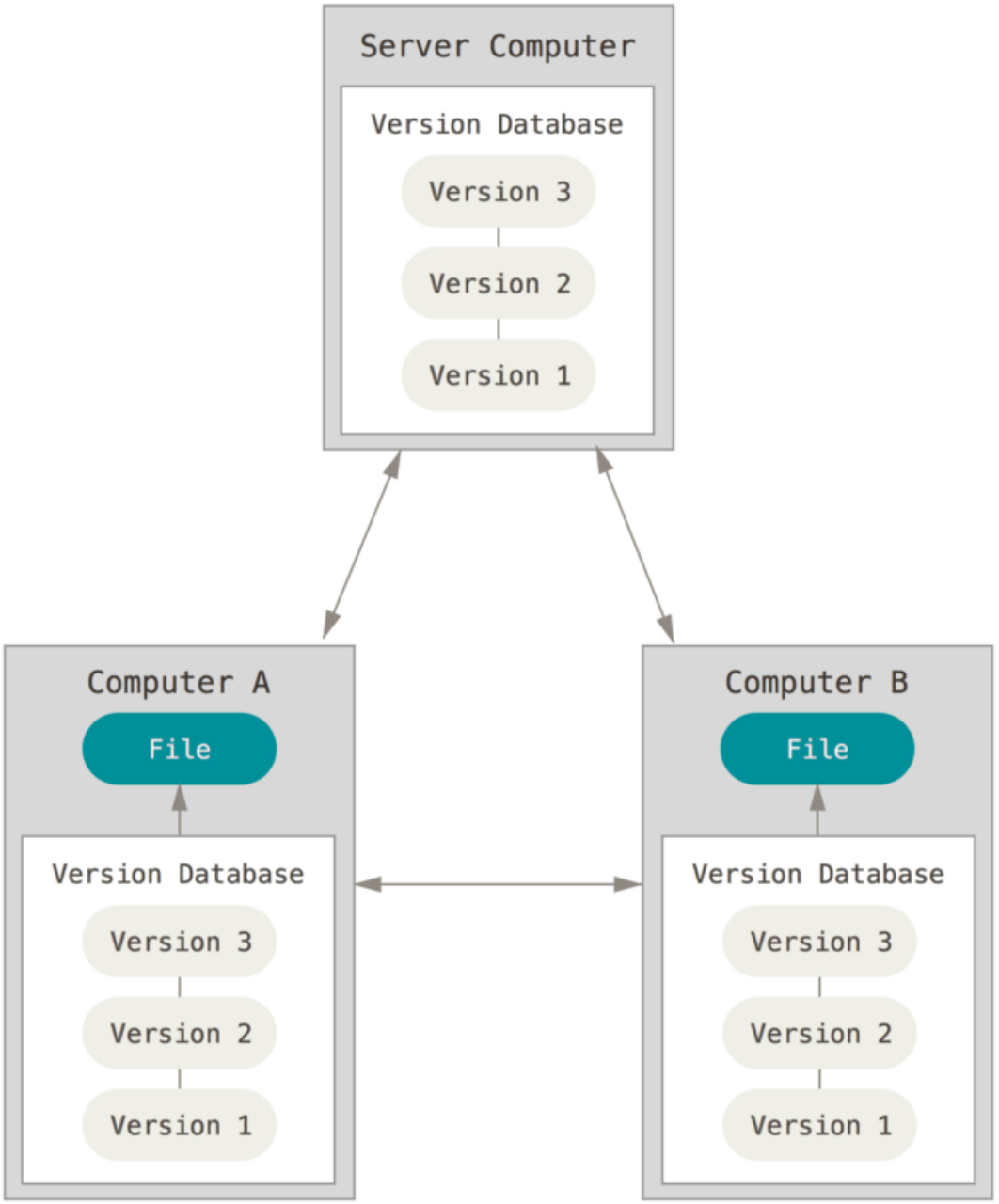
## Centralized Version Control Systems

It is system that records changes to a file or set of files over time so that you can recall specific versions later



## Distributed Version Control Systems

It is system that records changes to a file or set of files over time So that you can recall specific versions later





# What is Git?

- Open Source Project originally developed in 2005 by Linus Torvalds
- Imagine git as something that sits on top of your file system and manipulates files.
- This **something** is a tree structure where each commit creates a new node in that tree.

# What is Git Repository?

- The purpose of **git** is to manage a set of files, or a project as they change over time. Git stores this information in a data structure called a repository.
- A git repository contains, mainly:

    Set of commits

# What is a Commit?

- A commit object mainly consists of three things:

  A set of **changes** the **commit** introduces

  **Commit message** describing the changes

  A **hash**, a 40-character string that uniquely identifies the commit object

HEAD : It is a pointer to last commit on branch

# The Three States

- **modified**

  Modified means that you have changed the file but have not committed it to your database yet.

- **staged**

  Staged means that you have marked a modified file in its current version to go into your next commit snapshot.

- **commited**

  Committed means that the data is safely stored in your local database.

# 3 Steps of git

- **Introduce a change**

  introduce a change to a file that is being tracked by git

- **Add the actual staging aread**

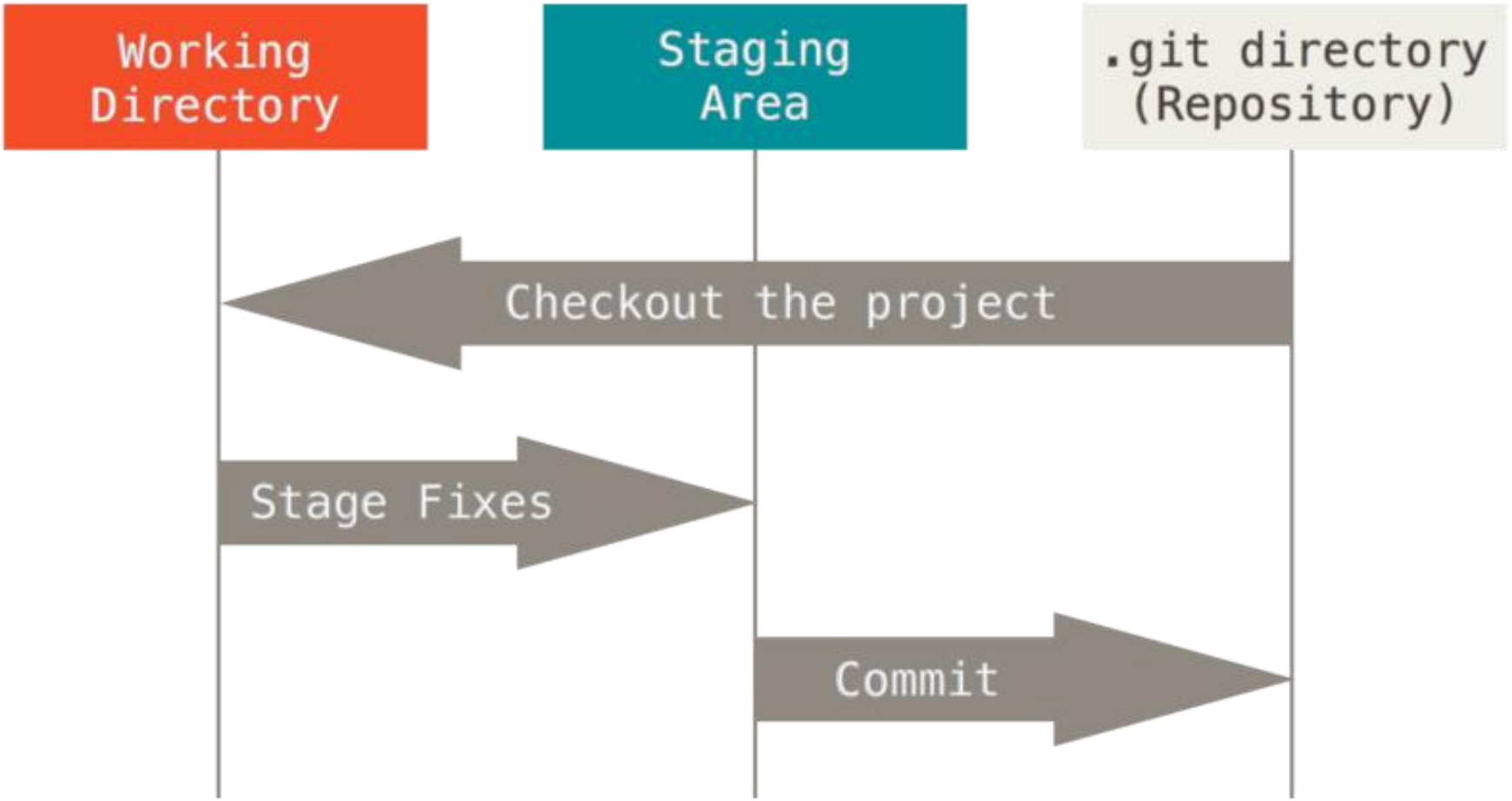
  Add the change you actually want using “**git add**”

- **Commit**

  Commit the change that has been added using

  “**git commit**”

# 3 Steps of git





- The ***working tree*** is a single checkout of one version of the project. These files are pulled out of the compressed database in the Git directory and placed on disk for you to use or modify.
- The ***staging area*** is a file, generally contained in your Git directory, that stores information about what will go into your next commit. Its technical name in Git parlance is the "index", but the phrase "staging area" works just as well.
- The ***Git directory*** is where Git stores the metadata and object database for your project. This is the most important part of Git, and it is what is copied when you clone a repository from another computer.



# Git Setup

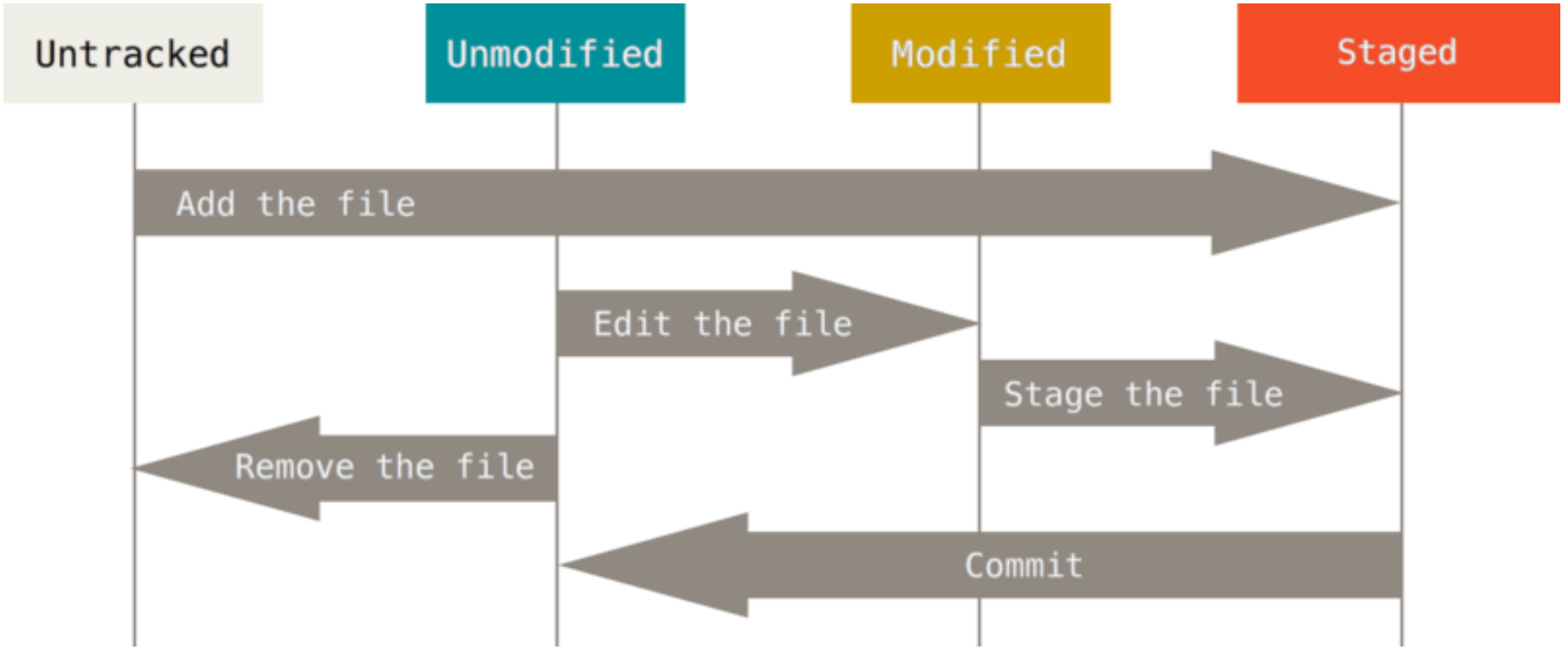
- You can take a **local directory** that is currently not under version control, and turn it into a Git repository.If you have a project directory that is currently not under version control and you want to start controlling it with Git, you first need to go to that project's directory.

**$ git init**

- You can **clone** an existing Git repository from elsewhere.

**$ git clone "Git-Repo-URL"**

# Changes to a Repository

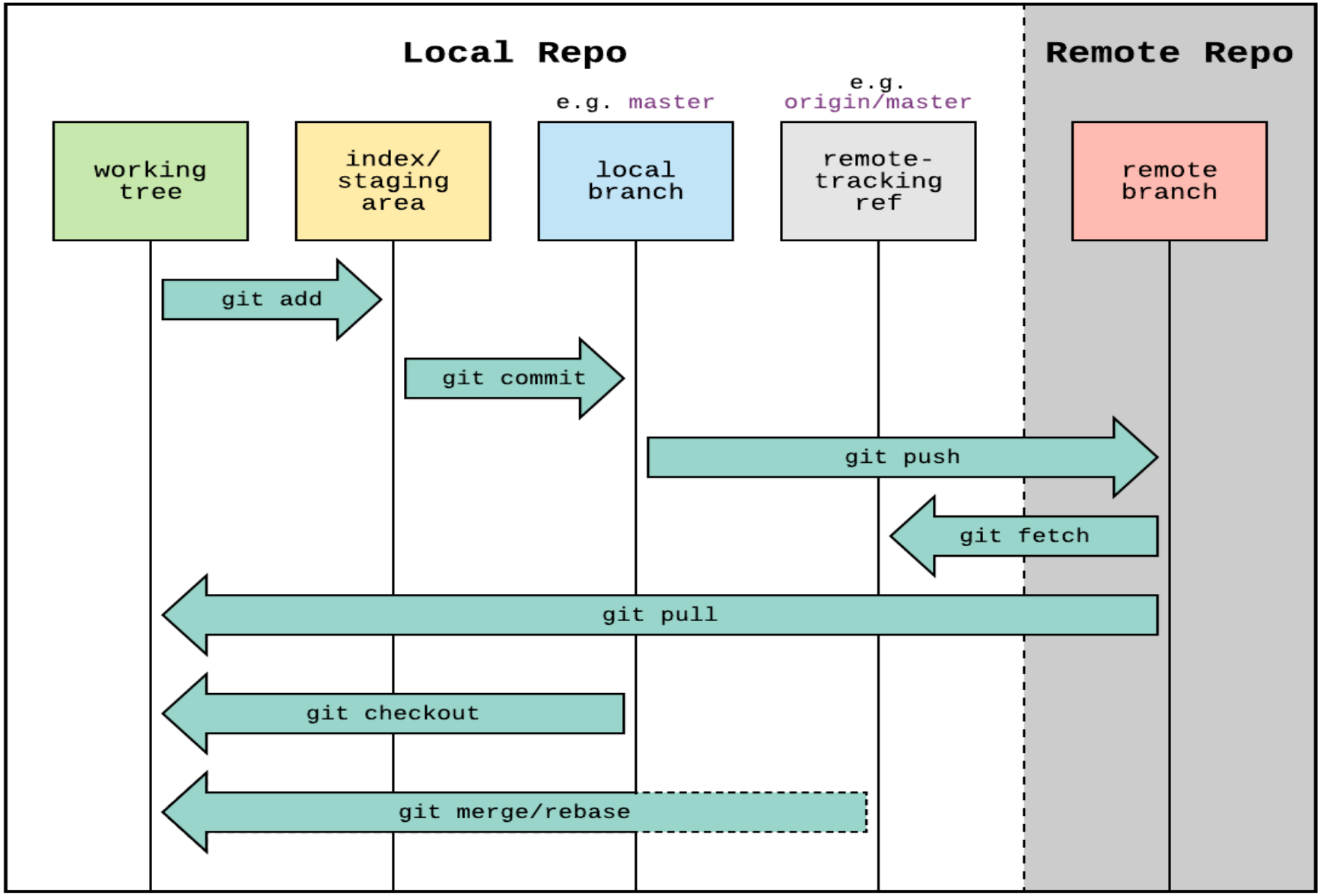




# Basic git commands

- git init
- git clone
- git log
- git diff
- git add
- git status
- git commit

# Git commands Workflow



# .git directory structure

- ├─.git
- ├─ HEAD/ (A pointer to your current branch)
- ├─ config/ ( contains all configuration preferences)
- ├─ description/(description of your project )
- ├─ Index/ (is used as staging area between working directory and repo)
- ├─logs/ (keeps records to changes that are made in ref)
- ├─objects/ (all data are stored here: commits, trees and tags )
- ├─ hooks/ (shell scrips that are invoked after executing a command)
- ├─refs/ (holds your local branch remote branch and tags)