GIT & GITLAB

INTRODUCTION & DEMONSTRATION

ALYSA TRUONG, IT INTERN

AGENDA

- Version Control System
- Terminology
- File Lifecycle
- Tools
- Demonstration

WHAT IS A VERSION CONTROL SYSTEM?

- **Version Control System (VCS)** – a software tool that records the changes of a file or set of files over time and share those changes with others.
  - Promote collaboration
  - Safe & proper storage of projects
  - Revert/Restore previous state/snapshot of files
  - VCS types available to fit needs and workflow
    - Centralized vs. Distributed

WHAT IS GIT?

- **Git** – a open-source version control system that tracks changes in computer files and coordinates work on those files among multiple people
  - Distributed version control system (DVCS) type
  - Track changes in any file types
  - Characterized by speed, data integrity, and support for distributed, non-linear workflows
  - Fully mirrors the project when copied
    - A full backup of all data

TERMINOLOGY

- **Repository (repo)** – a place (locally or remotely) that usually contains all files of a specific project and stores the revision history of all files within
- **Init** – initializing Git in a specific local project folder
- **Clone** - the process of copying a repository from the server to local computer
- **Unstaged** – untracked file or set of files
- **Staged** – tracked and prepared file of set of files for a snapshot
- **Adding** – the process of adding a file or set of files from *unstaged* to *staging/staged* in order for Git to track its changes
- **Snapshot** – the process of capturing staged file(s)' current state at a particular point of time

TERMINOLOGY CONT.

- **Commit** – the creation and local storage of full snapshot of changed, tracked file(s) with a brief description of completed work
- **Fetching** – the process of retrieving and downloading the latest changes from server's repo without combining it into local computer's repo
- **Merging** – the process of comparing changes between the server's repo and computer's repo, and then combining into computer's repo
- **Pulling** – the combined process of fetching and merging changes from the server's repo into the computer's repo as a single command
- **Pushing** – the process of uploading and combining new changes into the remote repo

TERMINOLOGY CONT.

- **Cloning** – the process of downloading the project from server to local computer
- **Forking** – the process of copying an existing repo for an user as their personal repo
- **Branching** – the process of creating active development line(s) contained within a repo but doesn't affect the master/main branch
- **Merge Request/Pull Request** – asking the owner of the remote repo if local repo changes are permitted to merge into remote repo
- **Merge Conflict** – an issue where content of file(s) from the remote repo does not match with content of file(s) from the local repo when pushing

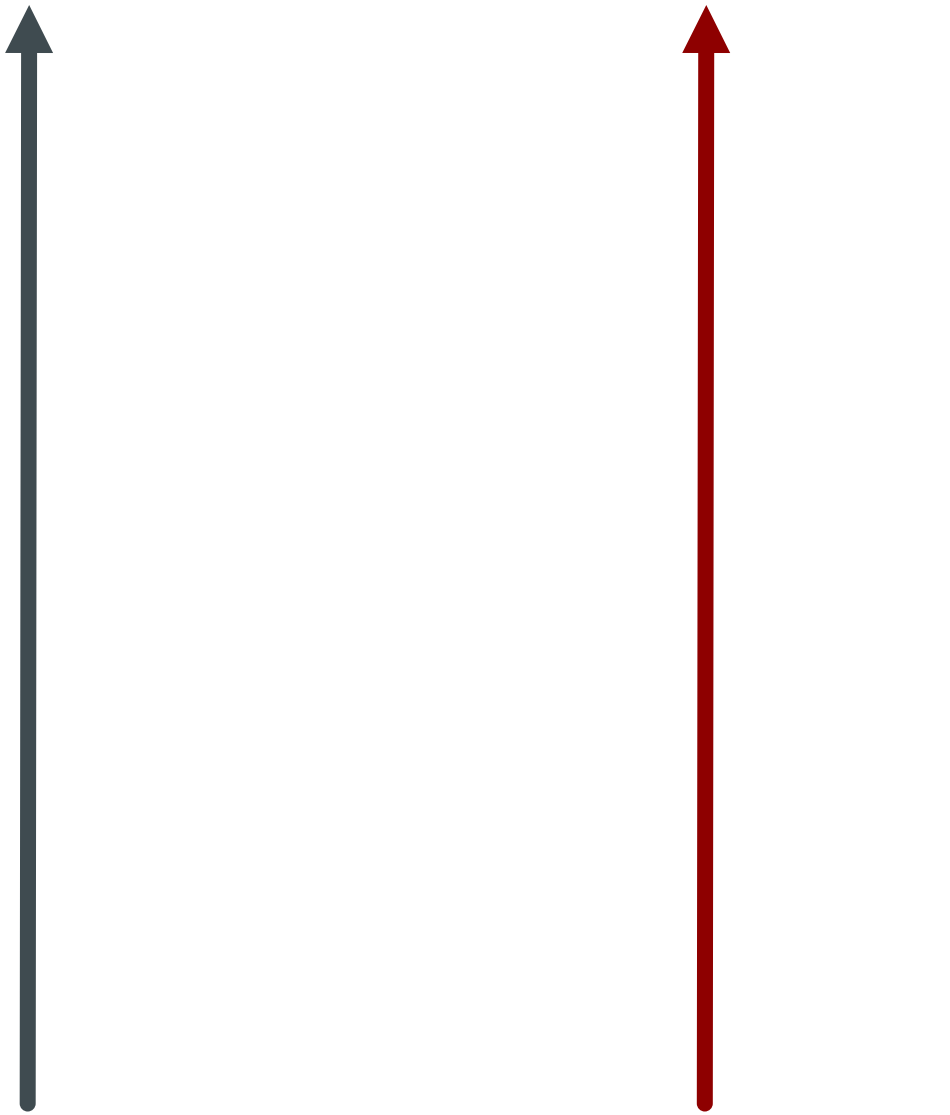
FORK VS BRANCH

- ***FORK***
  - Two Repositories, Independent Permissions

MASTER BRANCH

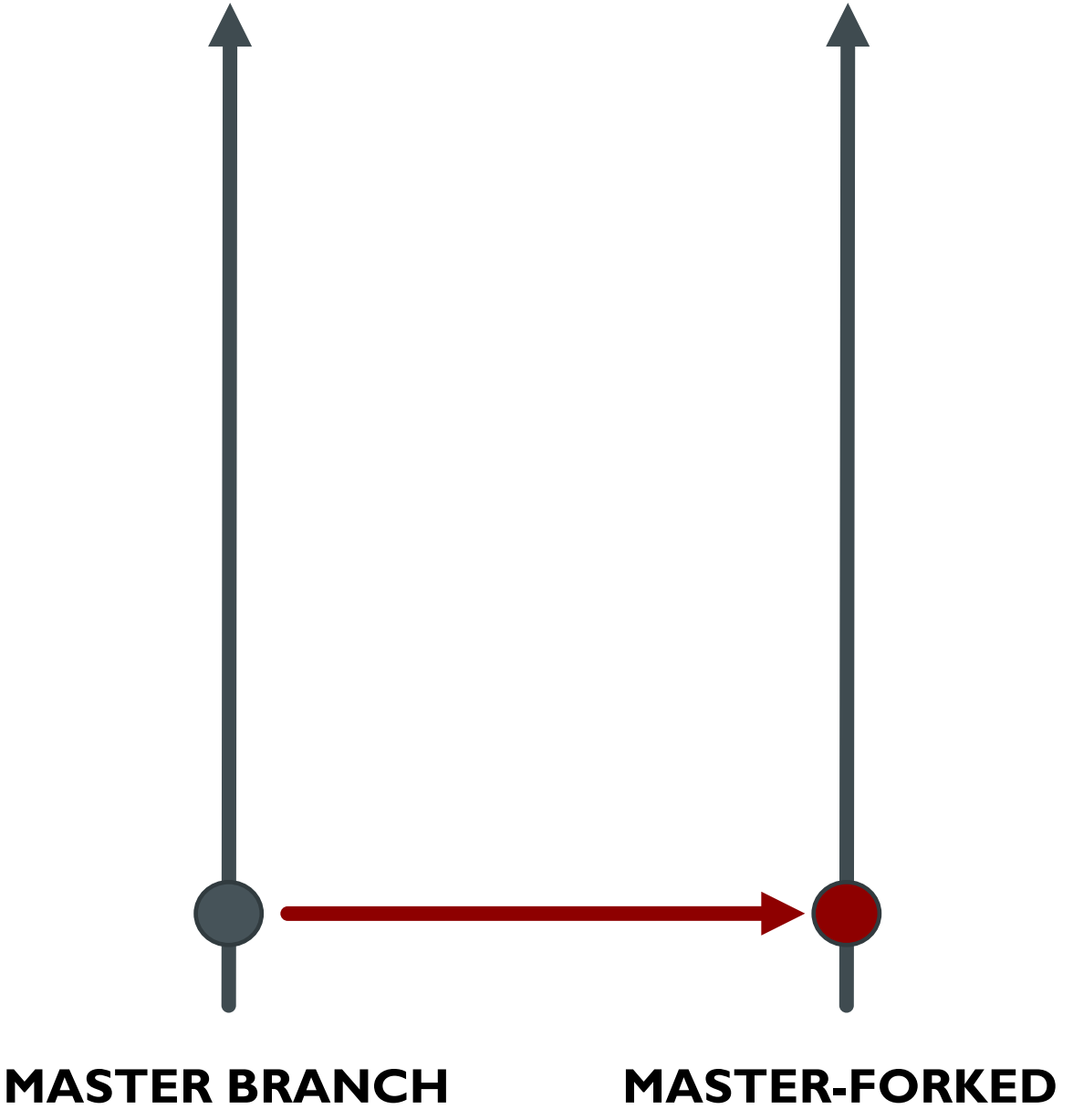
FORK VS BRANCH

- ***FORK***
  - Two Repositories, Independent Permissions



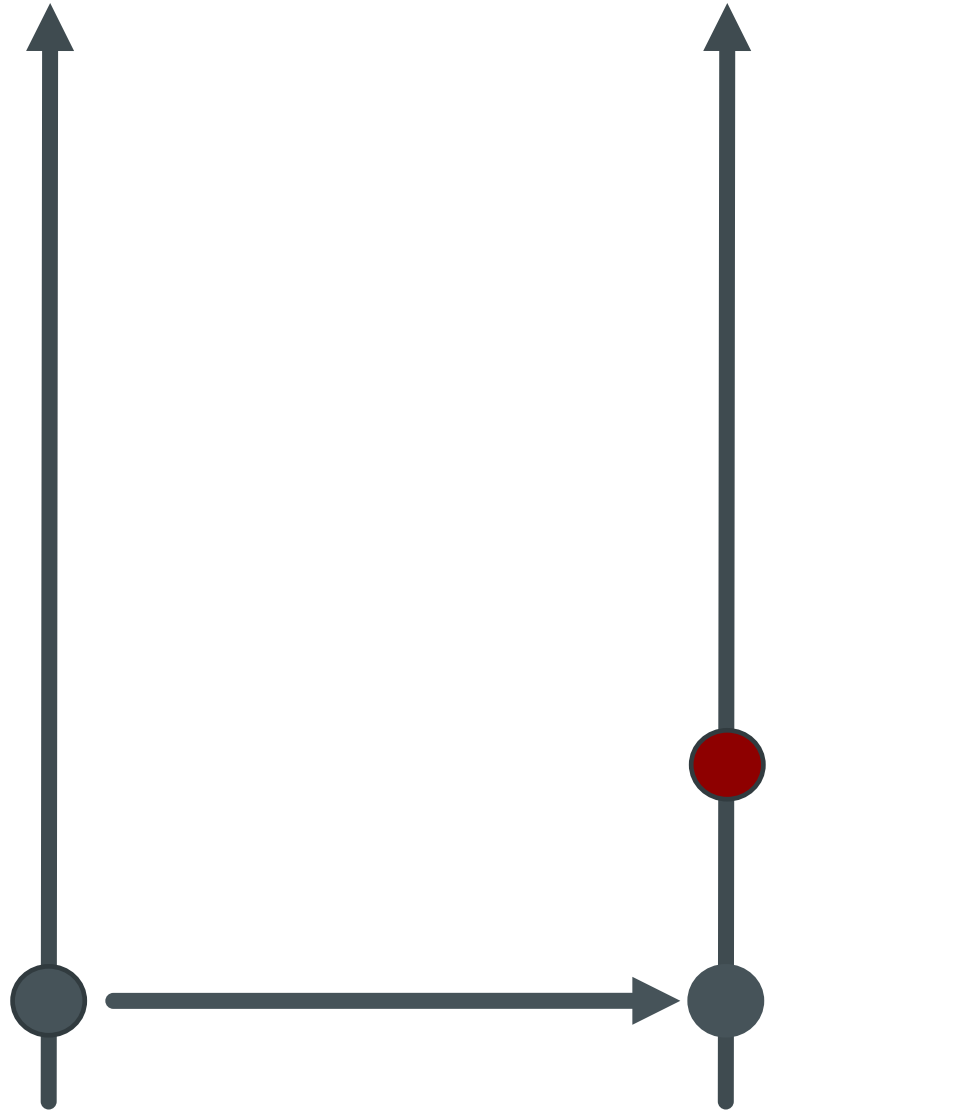
FORK VS BRANCH

- ***FORK***
  - Two Repositories, Independent Permissions



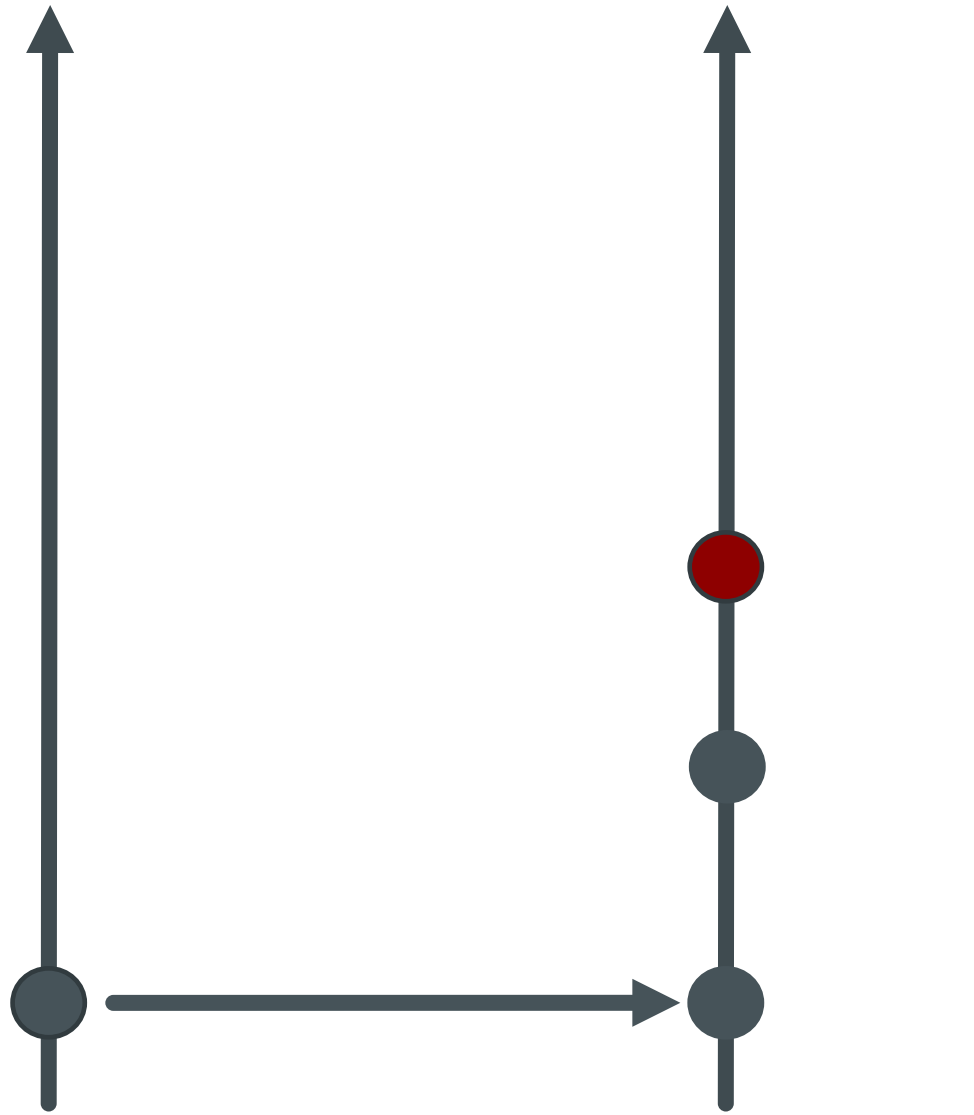
FORK VS BRANCH

- ***FORK***
  - Two Repositories, Independent Permissions



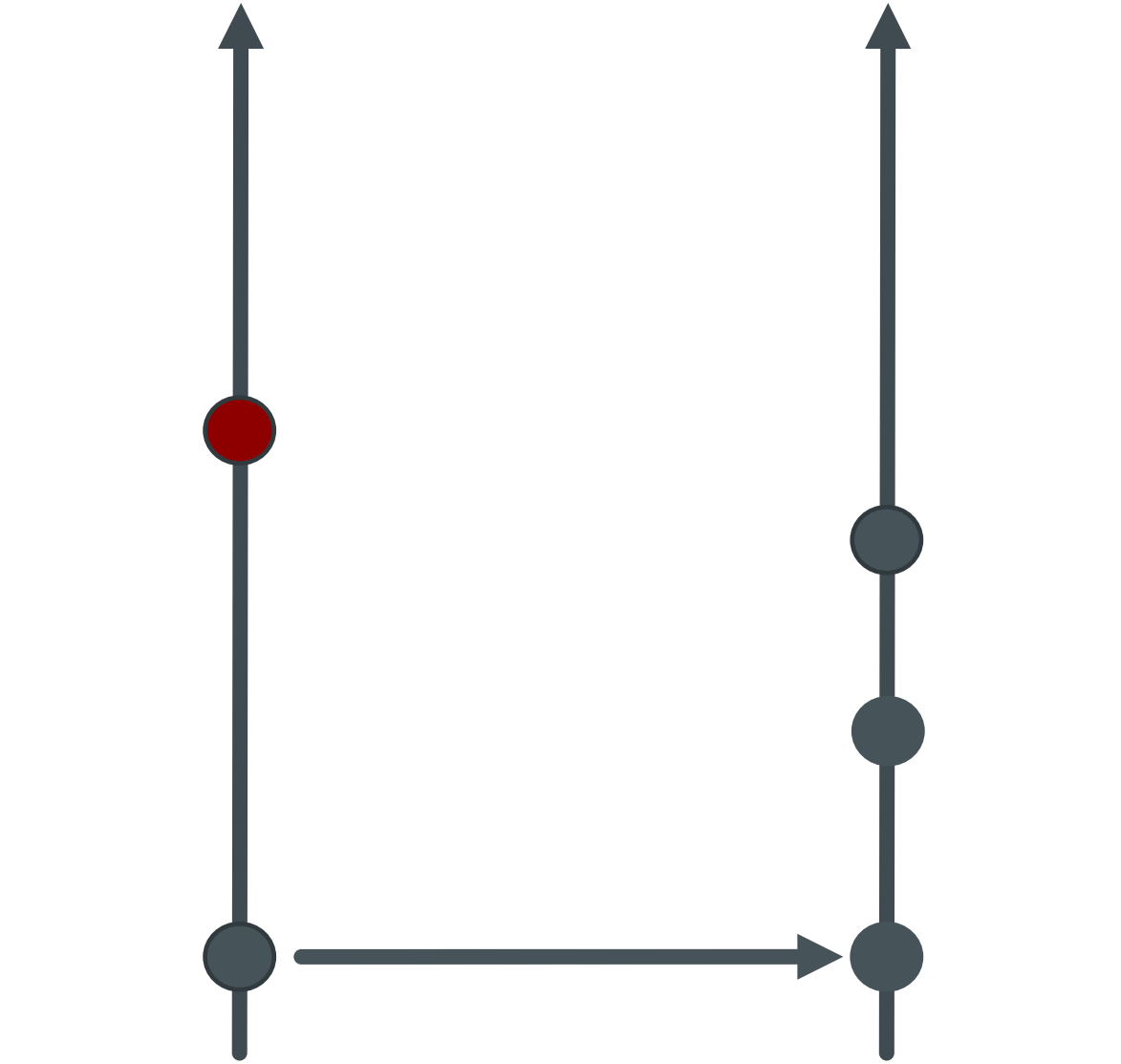
FORK VS BRANCH

- ***FORK***
  - Two Repositories, Independent Permissions



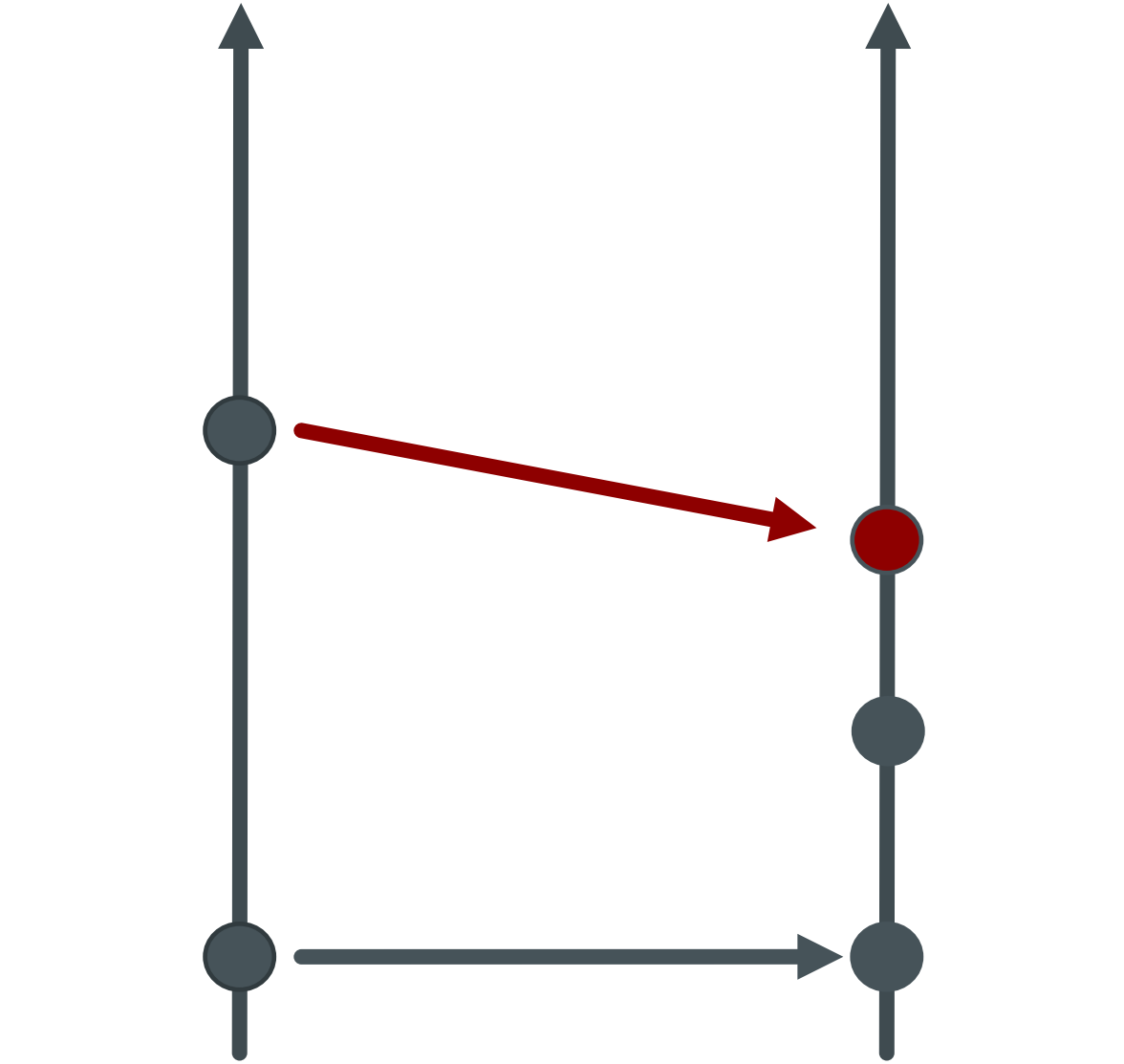
FORK VS BRANCH

- ***FORK***
  - Two Repositories, Independent Permissions



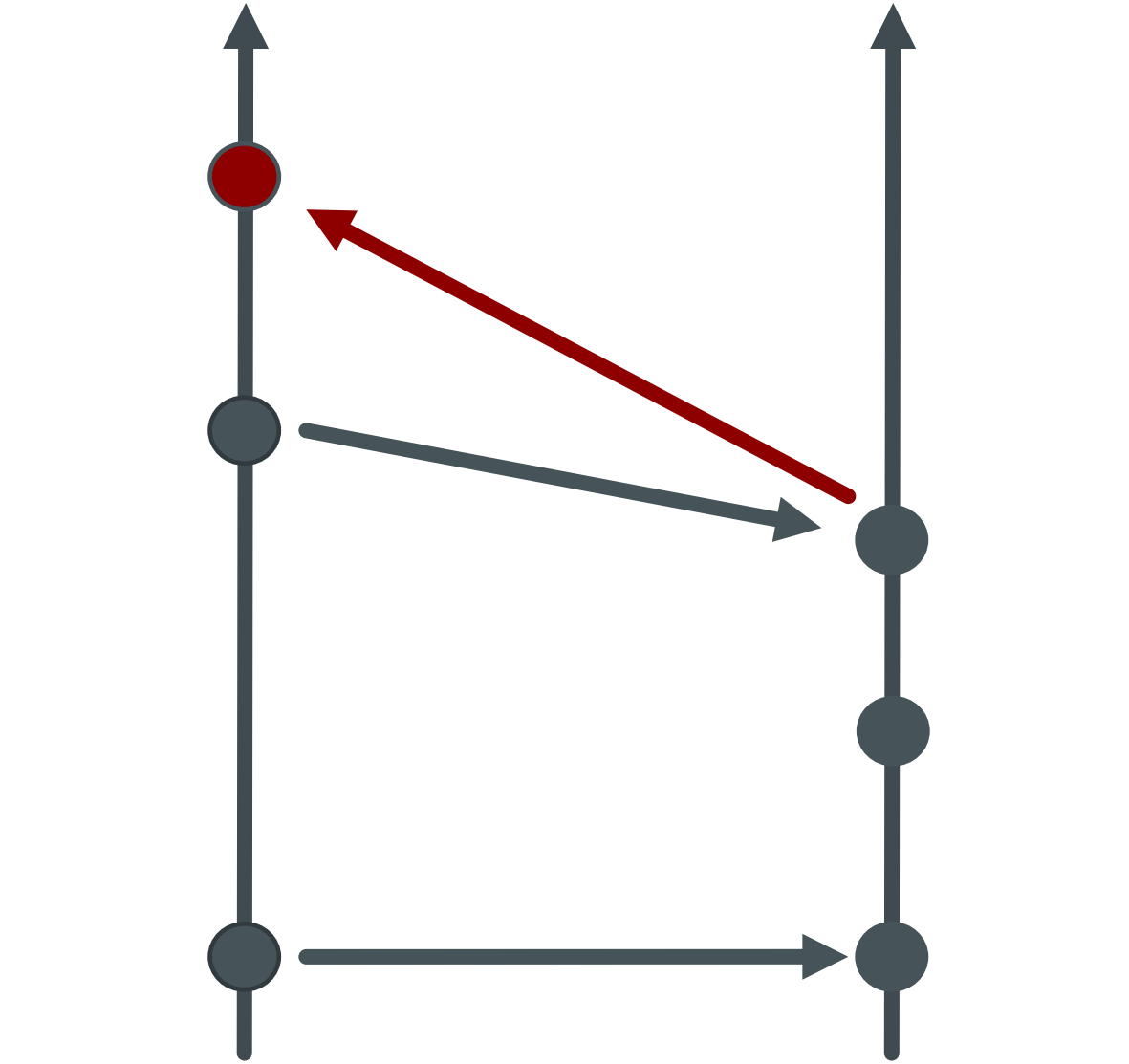
FORK VS BRANCH

- ***FORK***
  - Two Repositories, Independent Permissions



FORK VS BRANCH

- ***FORK***
  - Two Repositories, Independent Permissions



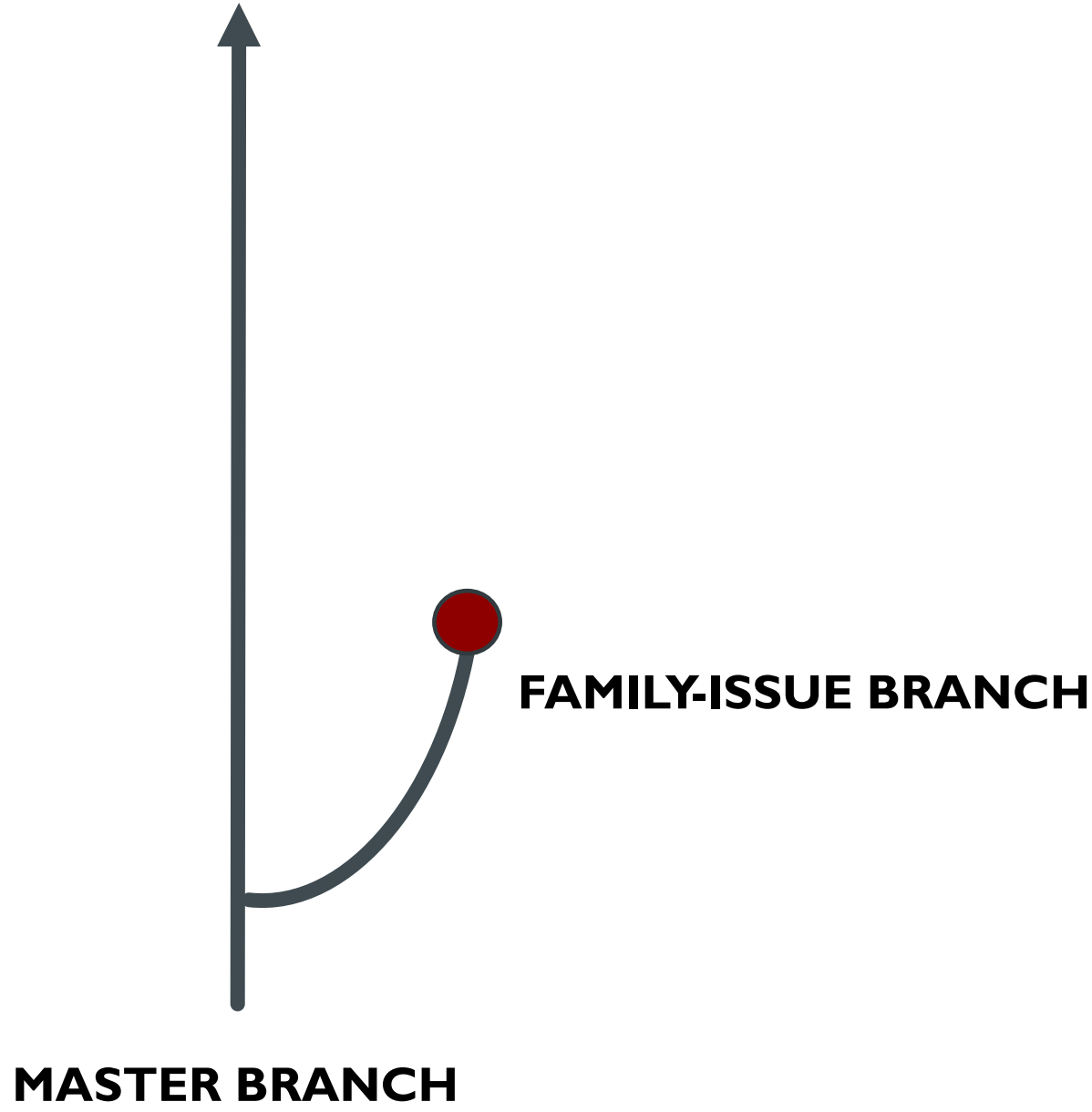
FORK VS BRANCH

- ***BRANCH***
  - One Repository,
    No Permissions

MASTER BRANCH

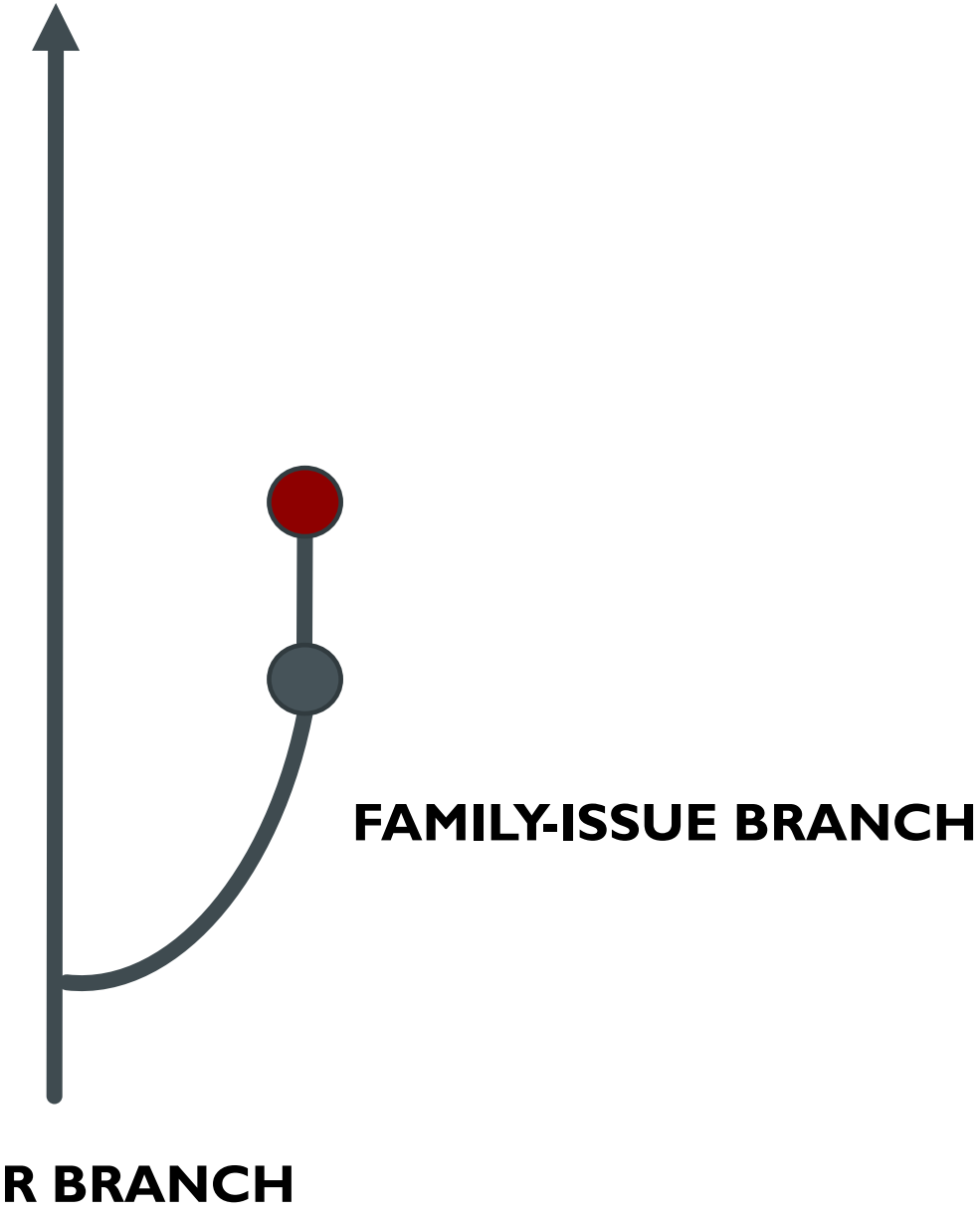
FORK VS BRANCH

- ***BRANCH***
  - One Repository,
    No Permissions



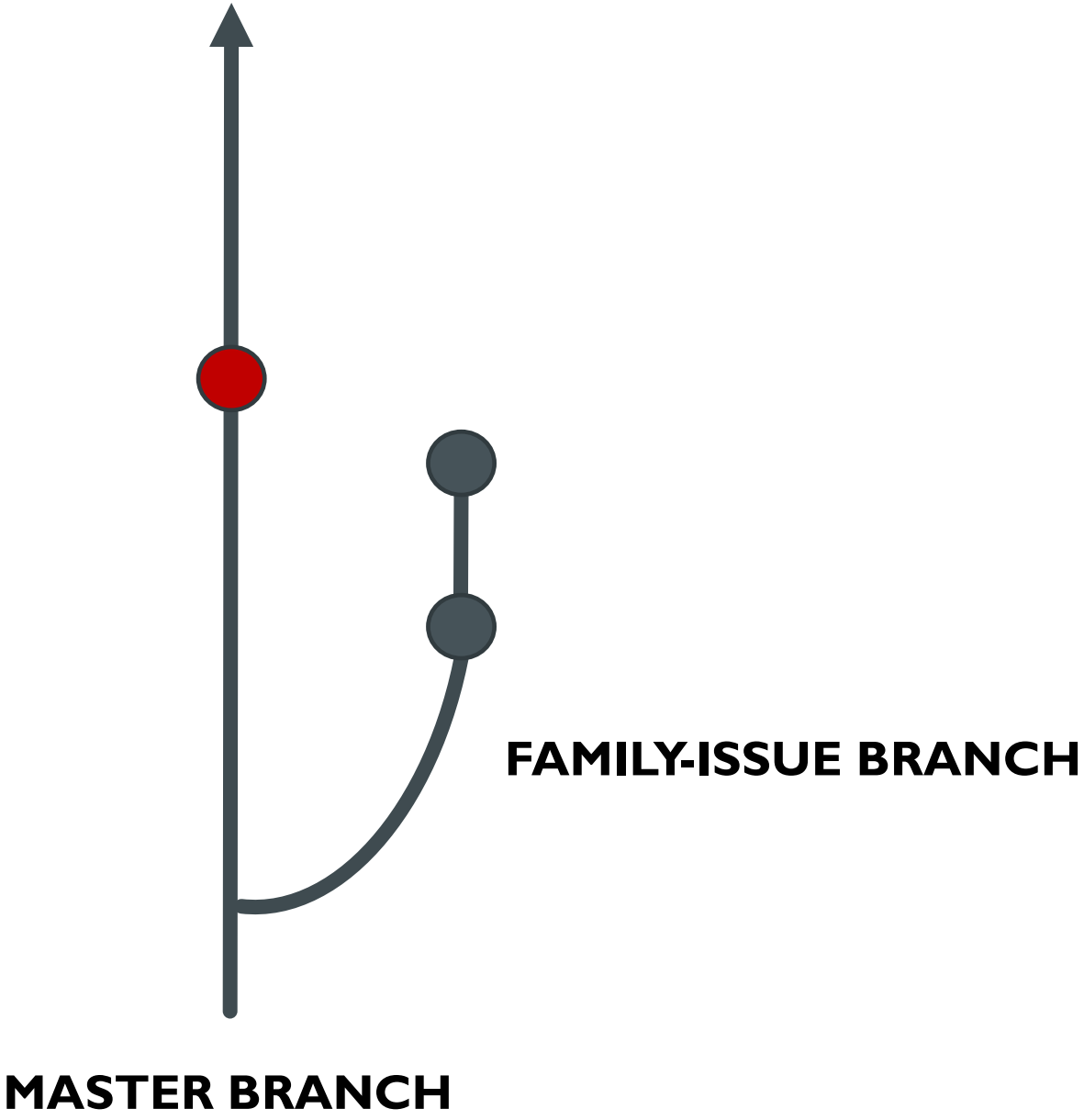
FORK VS BRANCH

- ***BRANCH***
  - One Repository,
    No Permissions



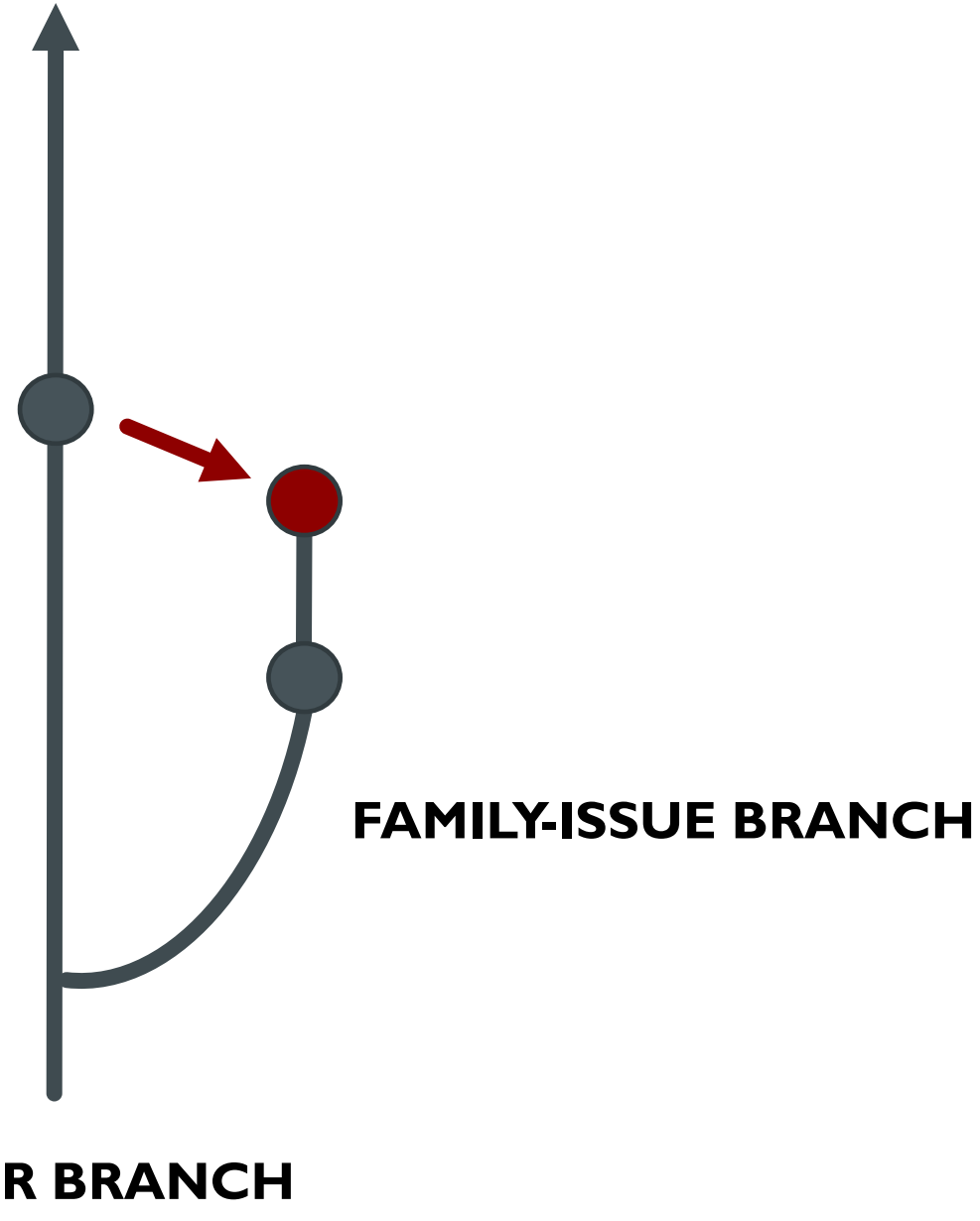
FORK VS BRANCH

- ***BRANCH***
  - One Repository,
    No Permissions



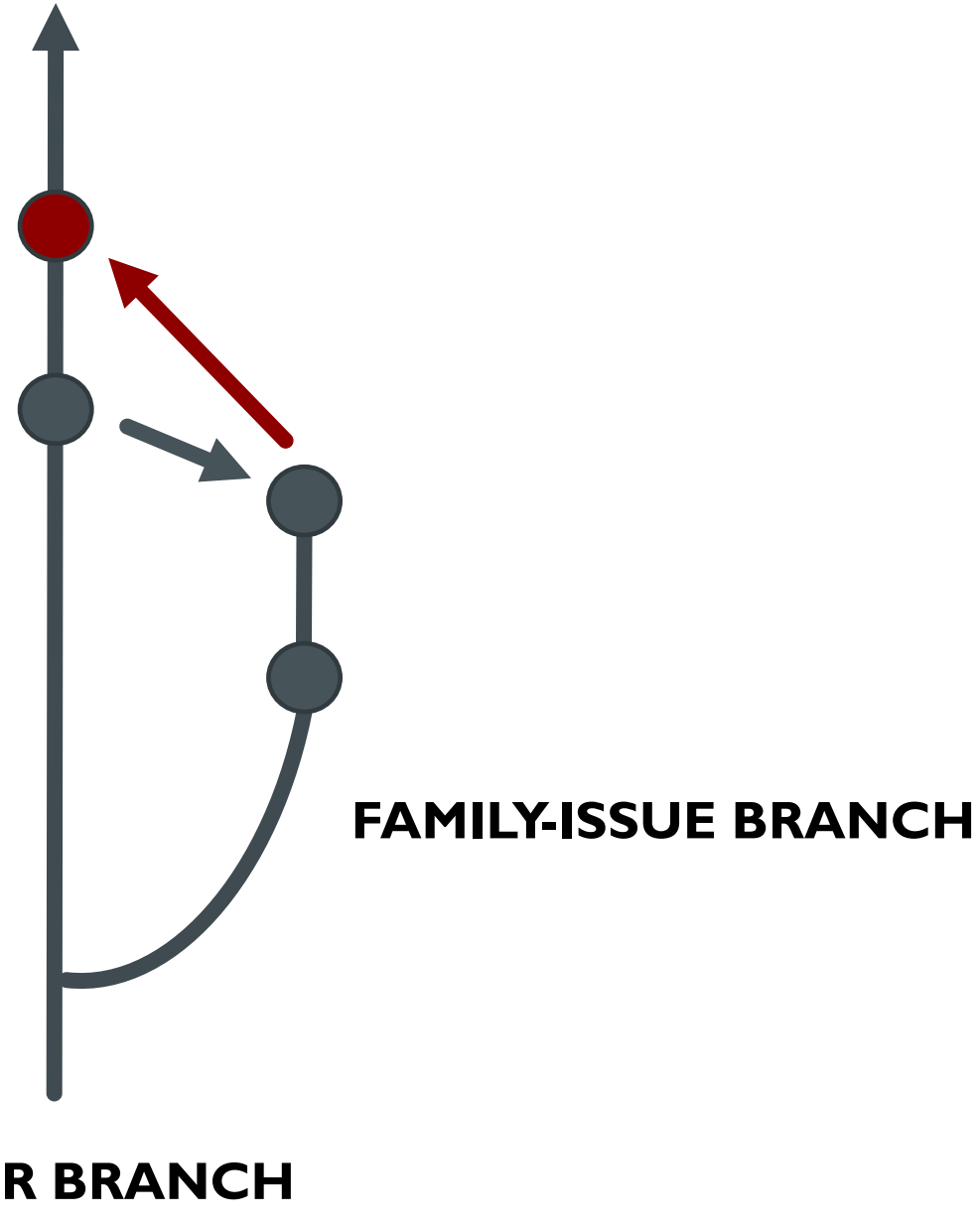
FORK VS BRANCH

- ***BRANCH***
  - One Repository,
    No Permissions



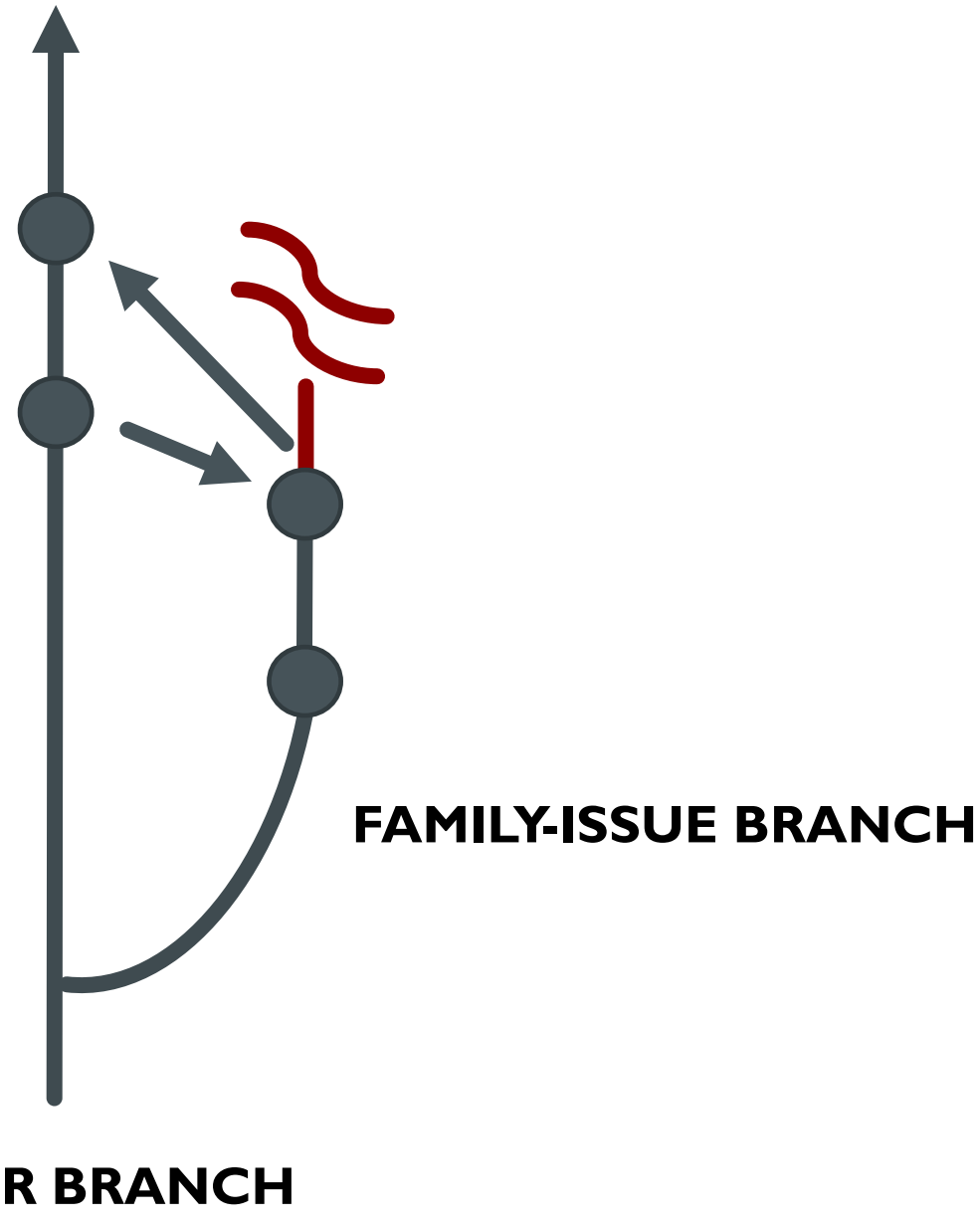
FORK VS BRANCH

- ***BRANCH***
  - One Repository,
    No Permissions



FORK VS BRANCH

- ***BRANCH***
  - One Repository,
    No Permissions



GIT VS GITHUB/GITLAB

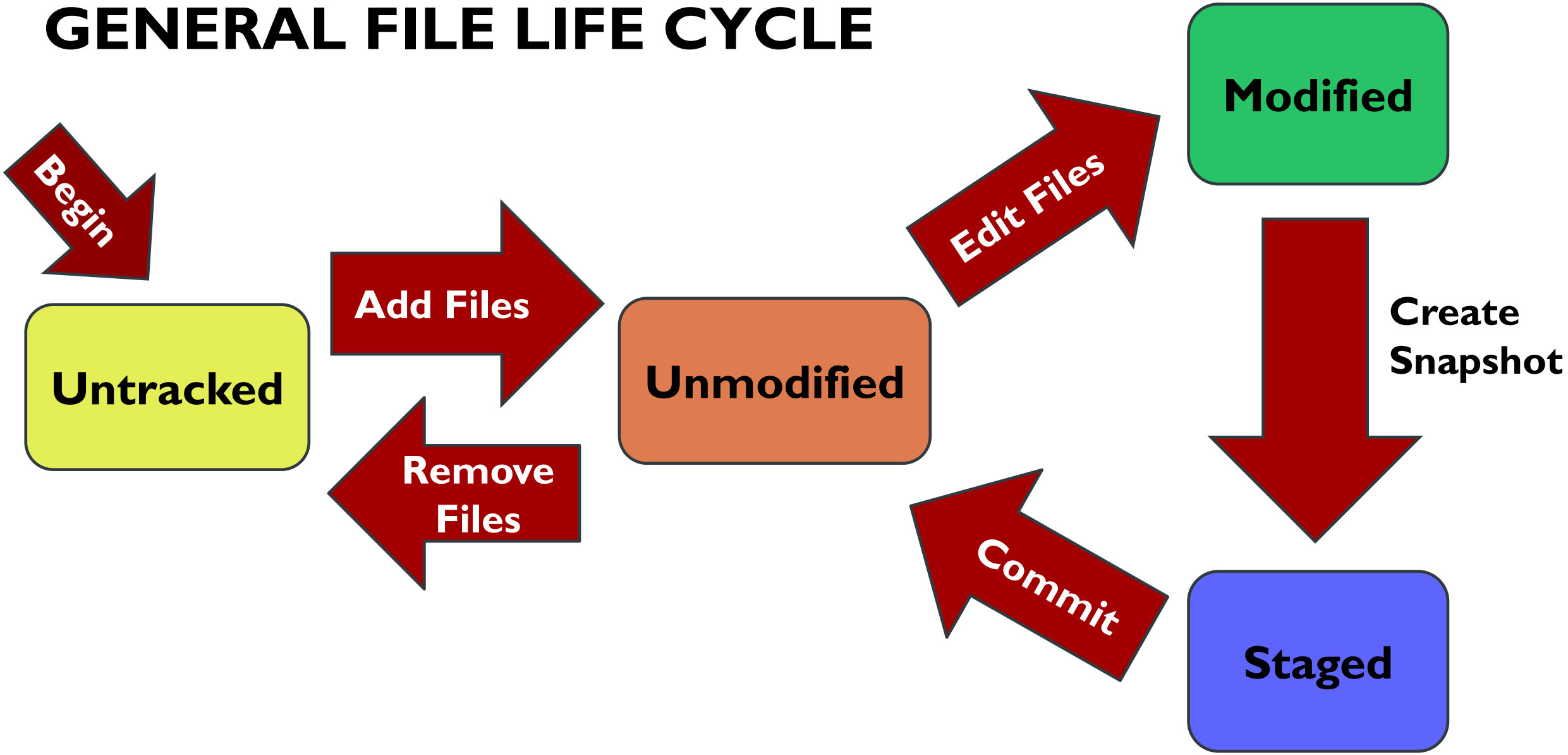
GIT

- A software tool available for usage of keeping track on file changes
- Ability to share content easily with others while managing files history
- Use commands to conduct certain actions

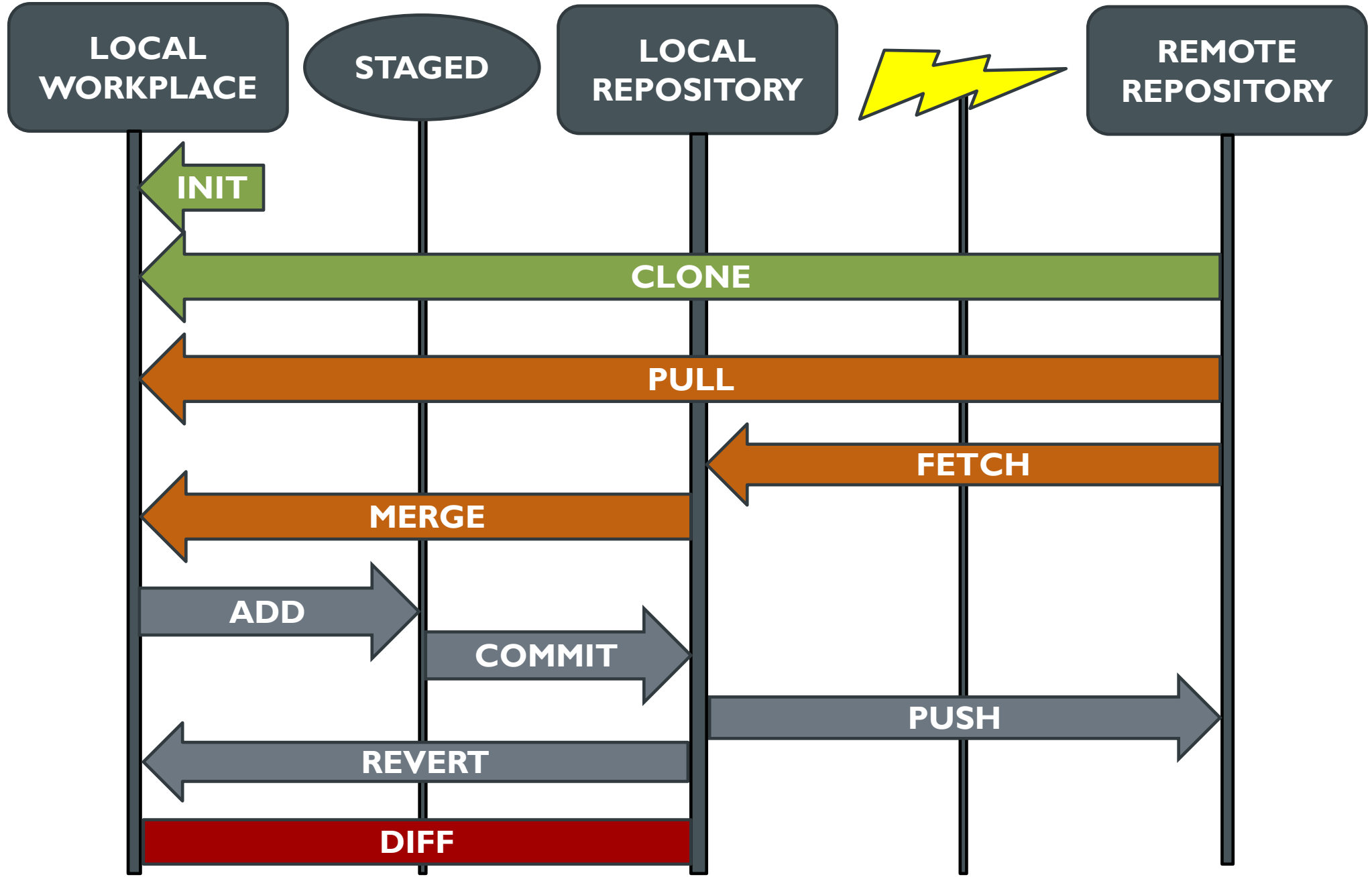
GITHUB/GITLAB

- Companies that offers and provides hosting services for Git repositories on their server
- Can either be open-source, paid, or free to use
- Difference between other hosting sites depends on performances, features, etc.

GENERAL FILE LIFE CYCLE



FILE LIFE CYCLE



INITIALIZE

UPDATE

CHANGE

DIFF

GIT, GIT, AND MORE GIT – TOOLS

- **Visual Studio Community 2015/2017**
- Offers Git features
  - Included in installation package
  - Does not need command lines to configure repository settings
- **RStudio**
  - Option to include/exclude during installation
    - Need to activate in settings
  - Uses command lines to configure new projects & settings

DEMONSTRATION

GIT/GITLAB, VISUAL STUDIO, & RSTUDIO

RESOURCES

- **Visual Studio Git FAQ**
  - *https://www.visualstudio.com/en-us/docs/git/gitquickstart*
  - *https://git-scm.com/book/it/v2/Git-in-Other-Environments-Git-in-Visual-Studio*
  - *https://marketplace.visualstudio.com/items?itemName=TFSPowerToolsTeam.VisualStudioToolsforGit*
- **Git**
  - *http://www.git-scm.com*
- **GitLab**
  - *http://www.gitlab.com*
  - *http://gitlab.reinvestment.com*
- **Cheat Sheet & Manual**
  - *IT will provide*
    - *Screenshots included!*

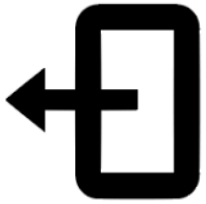
COMMAND LINES

- “git init” – command for initializing Git in the local project folder
- “git clone” – command for copying an project from a remote repository to a local repository
- “git add” – command for staging file or set of files from unstaged to staged before commiting
- “git commit” – command for creating a snapshot of the changed file(s) along with a brief description of the change and saved locally
- “git fetch” – command for retrieving and downloading new changes from the remote repository
- “git merge” – command for combining newly downloaded changes from the remote repository into local repository
- “git pull” – command for retrieving, downloading, and combining new changes from the remote repository to local repository
- “git push” – command for uploading and combining user’s changes from local repository to remote repository

QUESTIONS?

1. git commit

2. git push



3. leave building

THANK YOU!

Version Control Flowchart