Version Control with Git

Agenda

- What why and how of version control
- Centralized vs. Distributed version control
- Introduction to Git
- Basic Git concepts
- Setting up Git
- Git commands
- Git good practices

What is Version Control ?

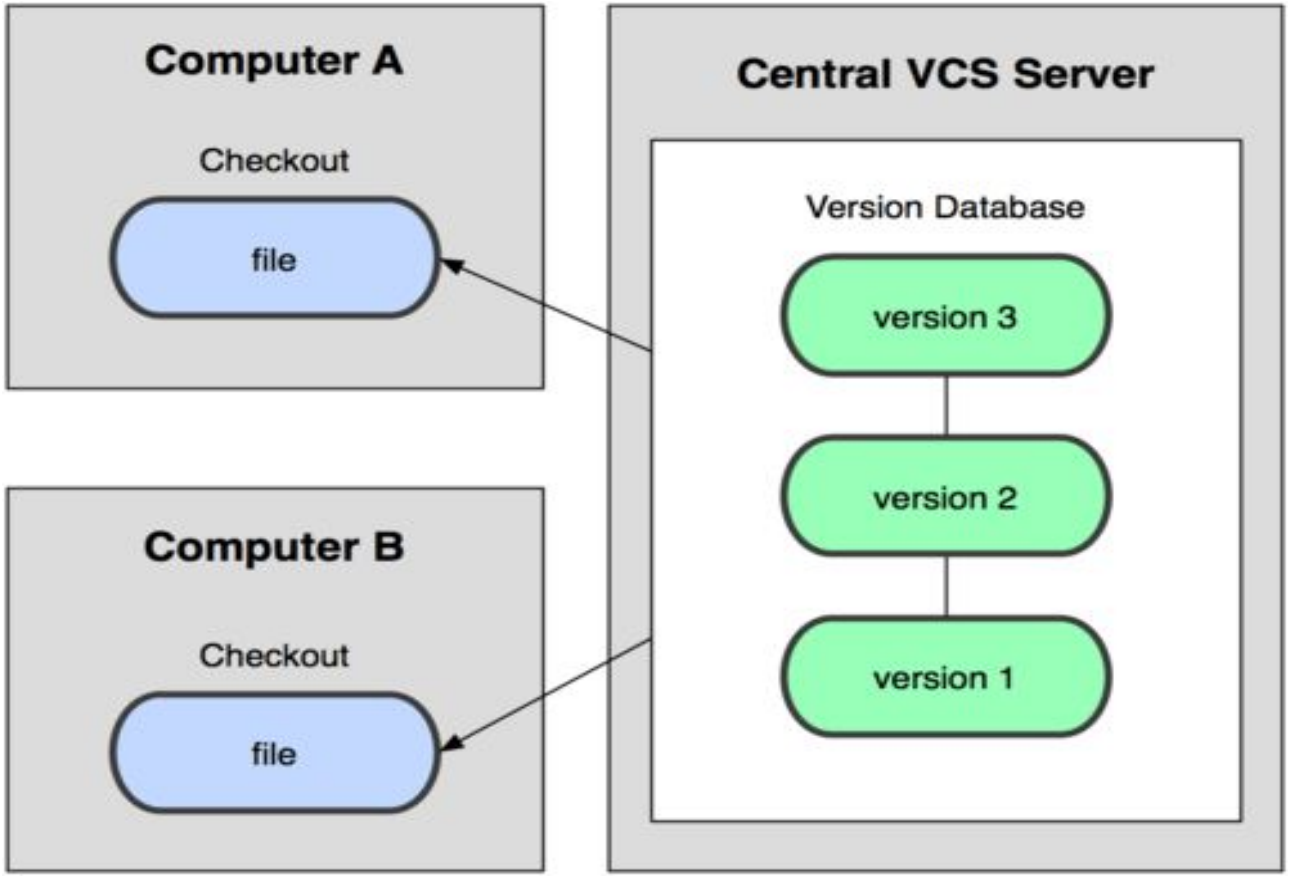
- A system that records changes to file(s)
- Provides control over changes
- Every change is tracked
    - Why the change happened?
    - References to problem fixed
    - Who introduced an issue and when
    - Revert back to a previous state

Why do we need it?

- Version tracking
- Maintaining multiple versions of software
- Coordinating within/between teams
- Not to loose anything
- Experimentation without interference

Centralized Version Control

- A single server that contained all versioned files(Eg- subversion)



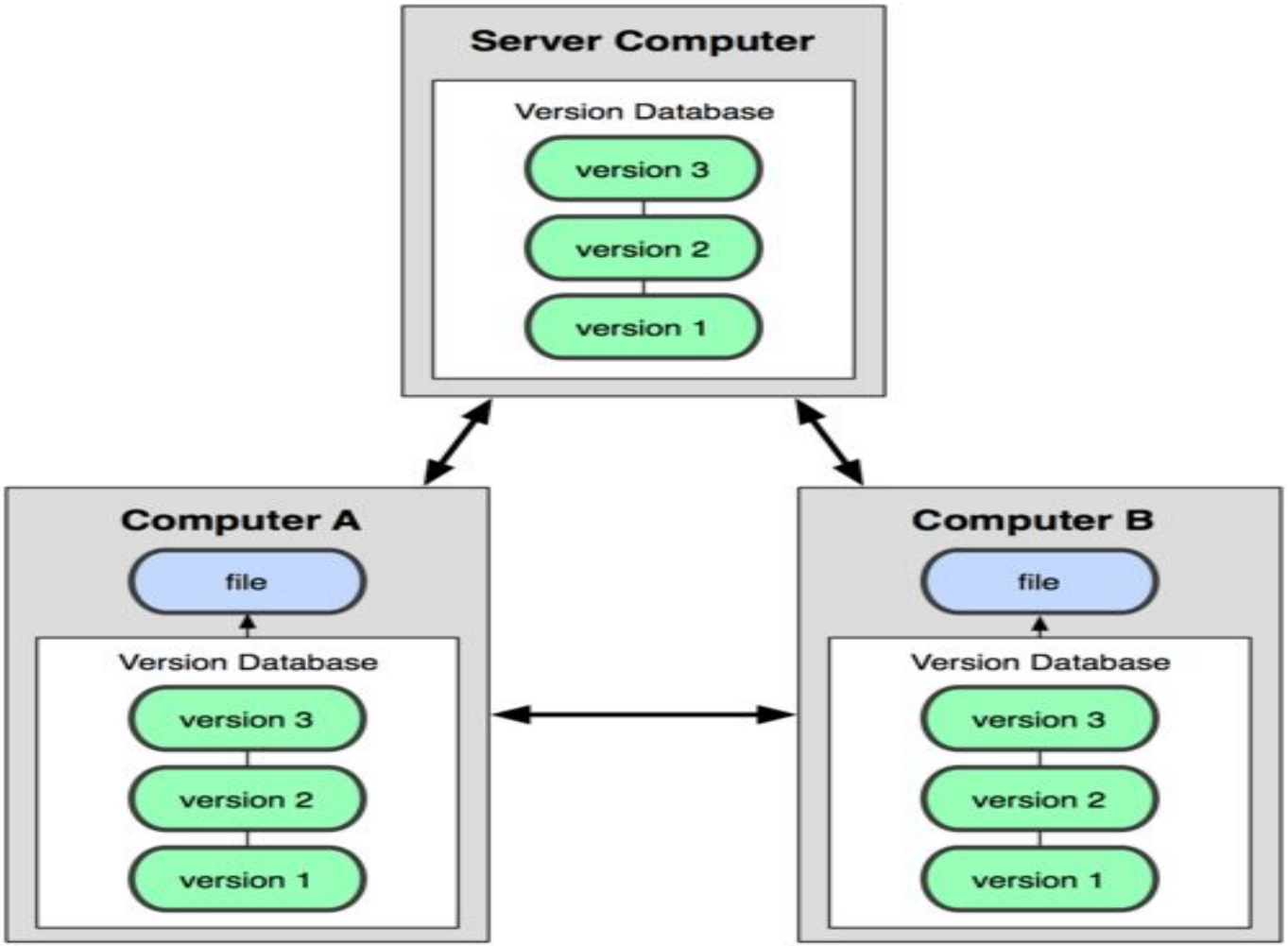
Disadvantages of Centralized System

- Single point of failure
- If hard disk gets corrupted without proper backups everything is lost.

Distributed Version Control

- Every developer has a copy
- Have full history on hard drive
- Faster except for remote repository operations

Distributed Version Control



Centralized vs Distributed

http://www.youtube.com/watch?v=_yQlKEq-Ueg

Introduction to Git

- Developed and designed by Linus Torvalds
- Very high performance
- Strong safeguard against accidental changes

Characteristics of Git

- Strong support for non linear development
- Fully Distributed
- Efficient handling of large projects
- Simple Design

Basic Git Concepts

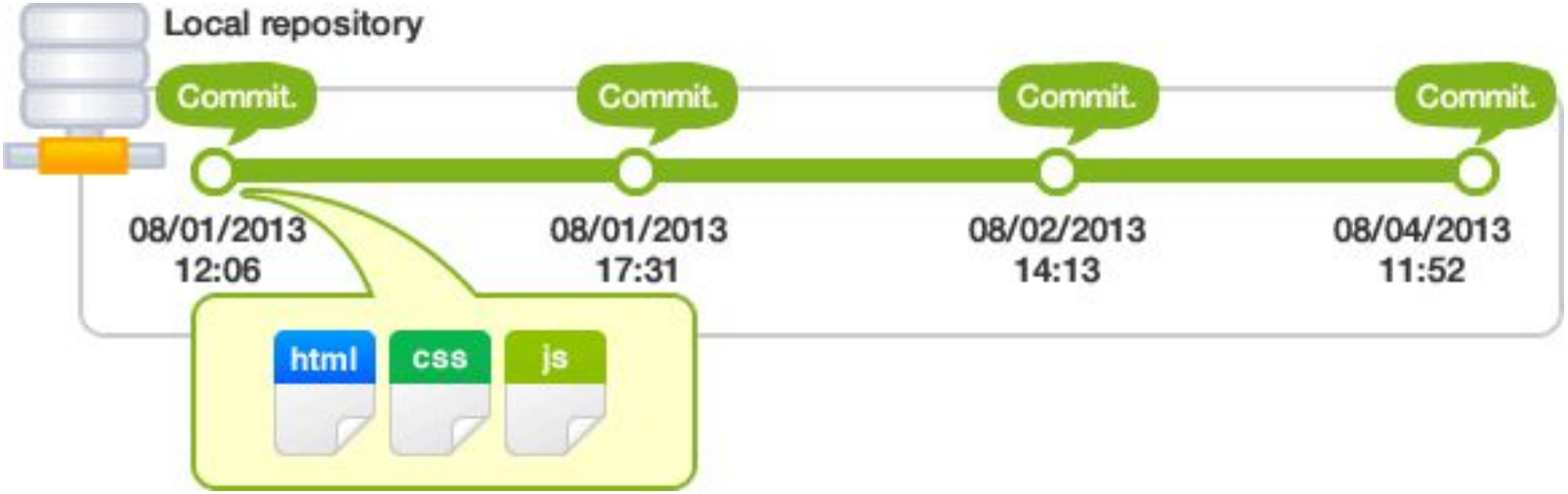
Git has following three states for files:

- Committed : it means that the data is safely stored in local database.
- Modified : it means that the file is changed but not committed to database yet.
- Staged : it means that the file is marked to go into next commit snapshot.

Basic Git Concepts

- Commit
  - A snapshot
  - Represents checked in version of files/directories
  - Uniquely identified by a 40 char string
  - Has a reference to the parent commit
  - Tree based structure when branching

Basic Git Concepts



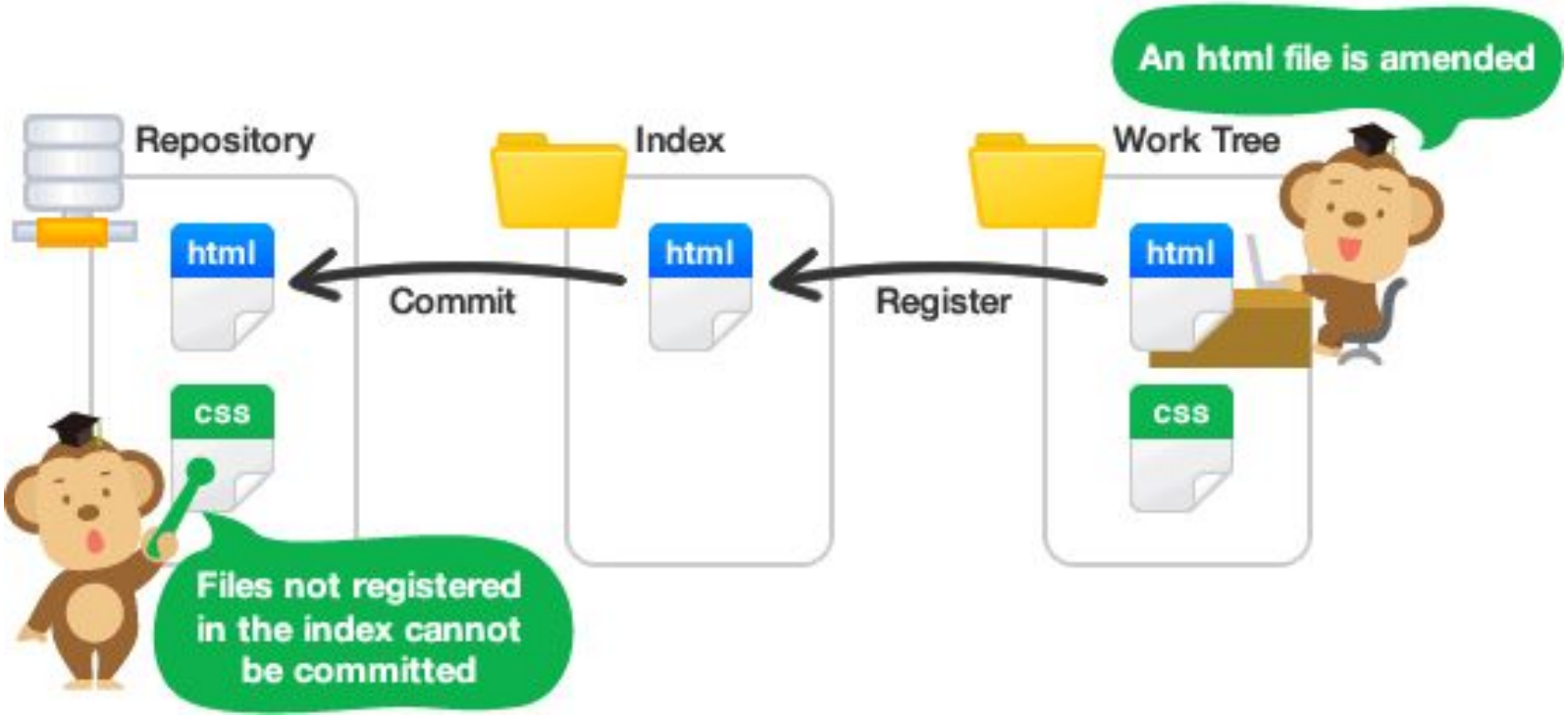
Basic Git Concepts

Working tree and index

- working tree : consists of files that you are currently working on.
- Index : is a staging area where new commits are prepared. It acts as an interface between repository and working tree.

Changes made on the working tree will not be committed directly to the repository. They need to be staged on index first.

Basic Git Concepts

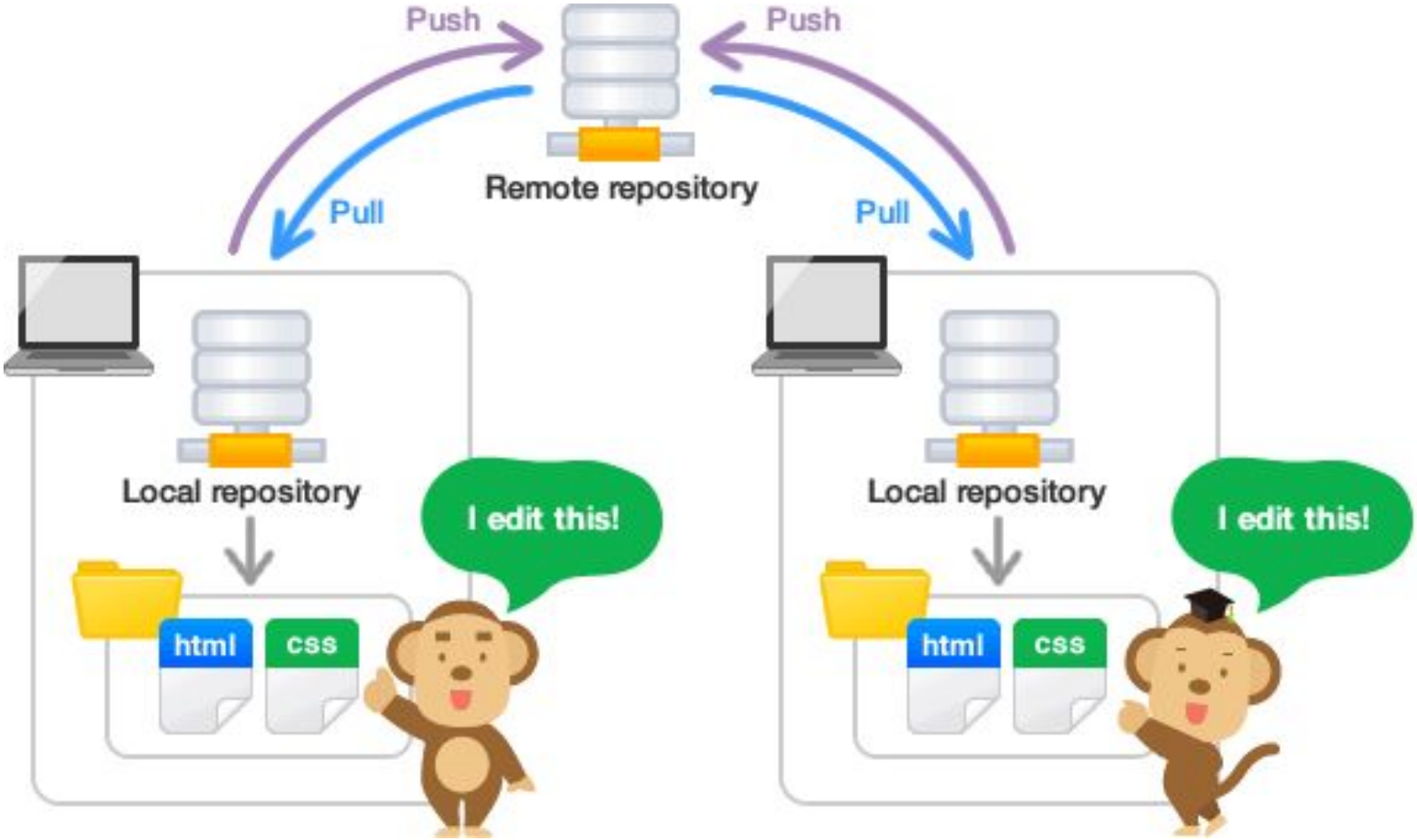


Basic Git Concepts

Remote Repository and Local Repository

- Remote Repository : Repository that resides on a server and is shared among all team members.
- Local Repository : Repository that resides on a local machine of an individual.

Basic Git Concepts



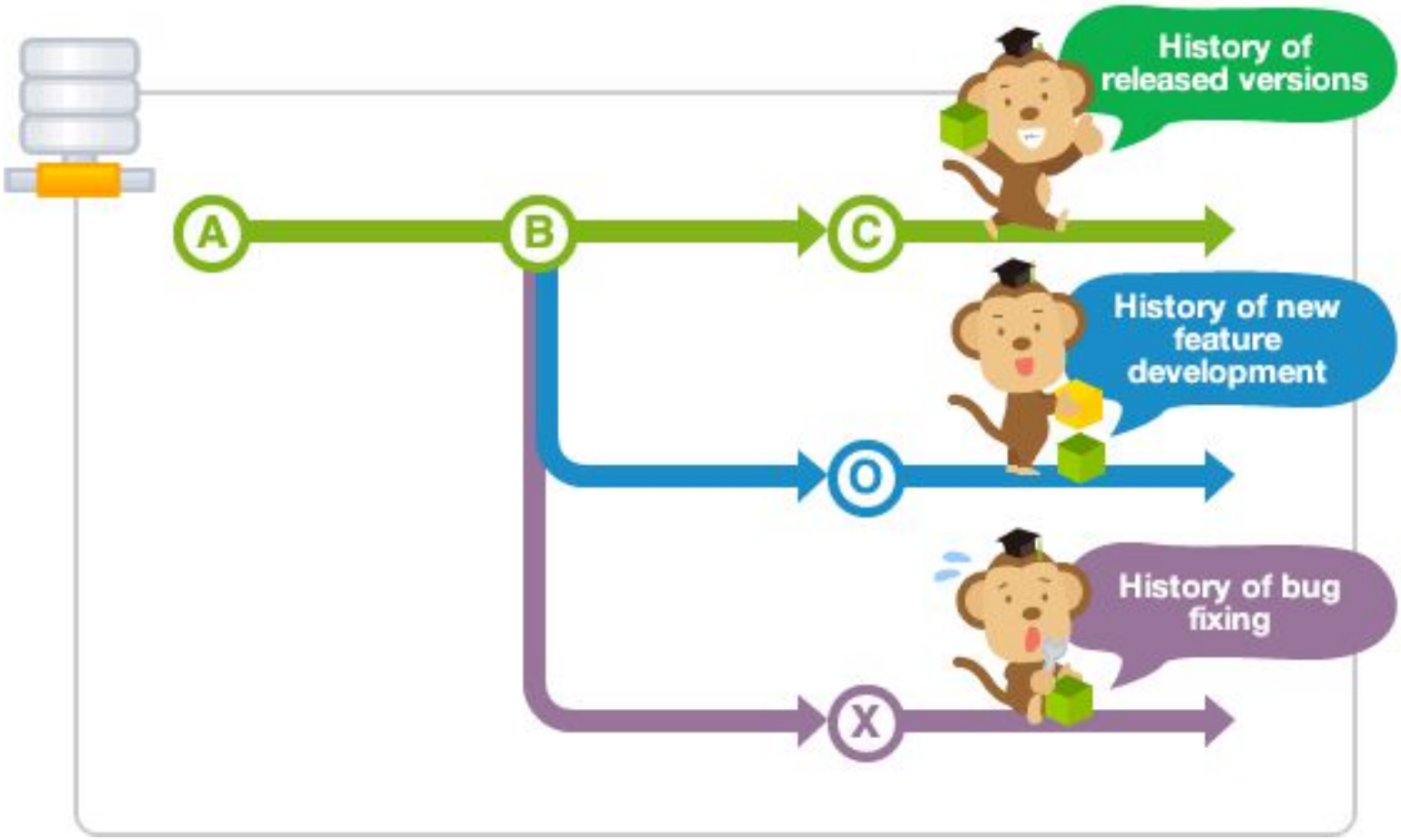
Basic Git Concepts

- Remote Repository -- Remote
    - Possible to have more than one repository
    - An alias for each repository
    - Update from it with fetch and pull commands
    - Send updates with push

Basic Git Concepts

- Branch
  - An active line of development
  - Most recent commit on branch is referred as “tip”
  - A single git repo can track an arbitrary number of branches.
  - But working tree is associated with one of them and HEAD points to that branch.

Basic Git Concepts



Basic Git Concepts

- Merge Conflict
    - Same line of a file can be edited by multiple people
    - Or even in different branches
    - Git is intelligent in case the changes are at different line numbers in a file

Setting up Git

- Sudo apt-get intall git
- git needs your SSH public key to allow pushing and pulling
- git config –global user.name <name>
- git config –global user.email <email>

Git Commands

- init : initialize a directory as git repo
- add: add file(s) to git
- rm : remove file(s) from git
- commit : commit staged files
- checkout: checkout files/branches
- branch: view branches
- push : update remote repo with changes of local repo
- pull : update local repo with changes of remote repo

Git Commands

- stash : manage stash
- remote: manage set of tracked repos
- merge : merge changes
- Resolving merge conflicts
    - Edit
    - Add edited files
    - Commit files

Ignoring files in git

- .gitignore : Its not a command but a file which consists of file names which should not be checked in the VCS. Please remember that this file itself will be checked in the VCS.

Git good practices

- Atomic commits
- Meaningful commit messages
- Use of gitignore

Git References

- Git Manual page
- Git cheat sheet
- https://www.codeschool.com/courses/git-real

Questions....