Git

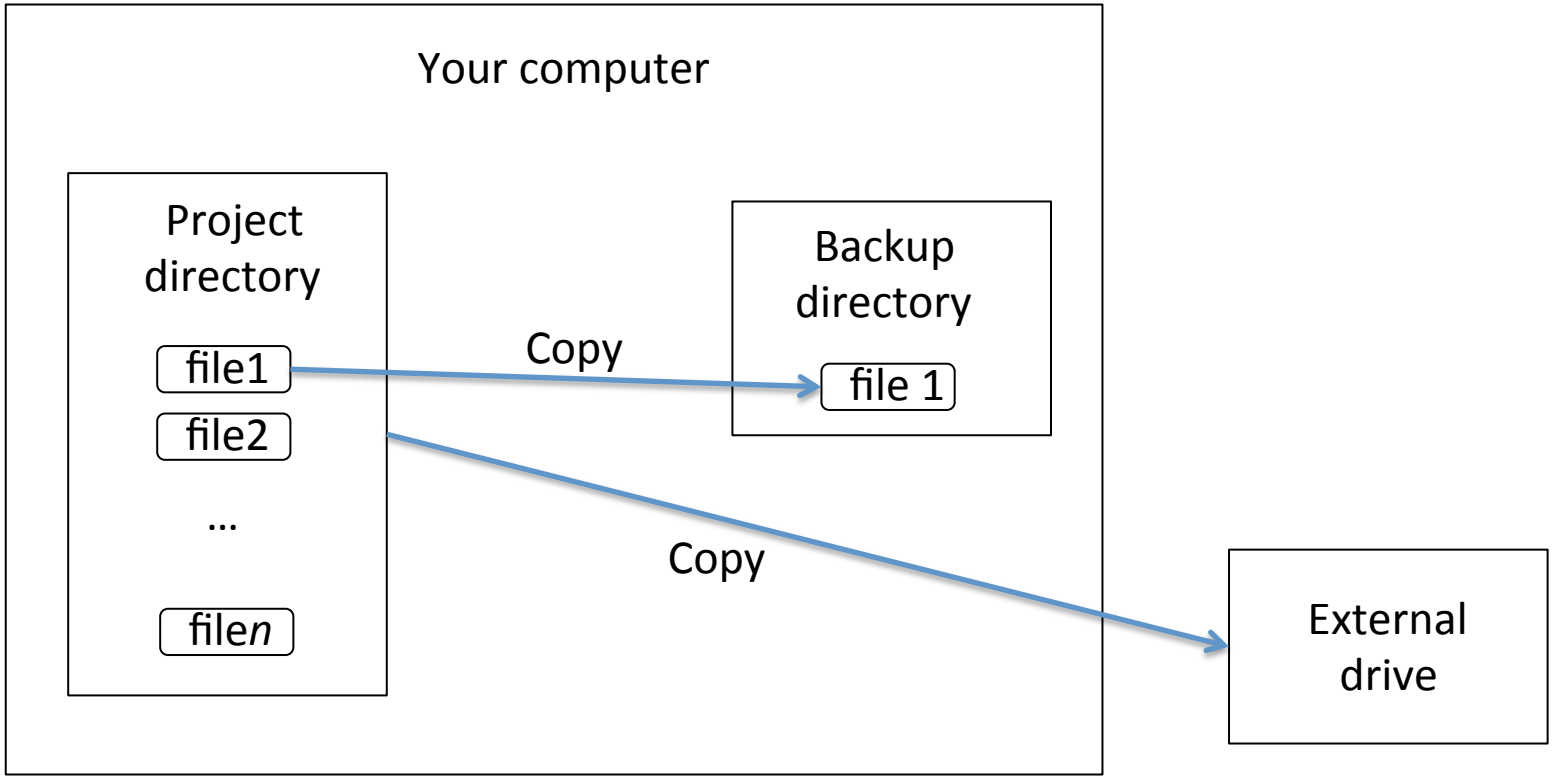
Carlos Cruz
NASA GSFC

Python bootcamp 2016

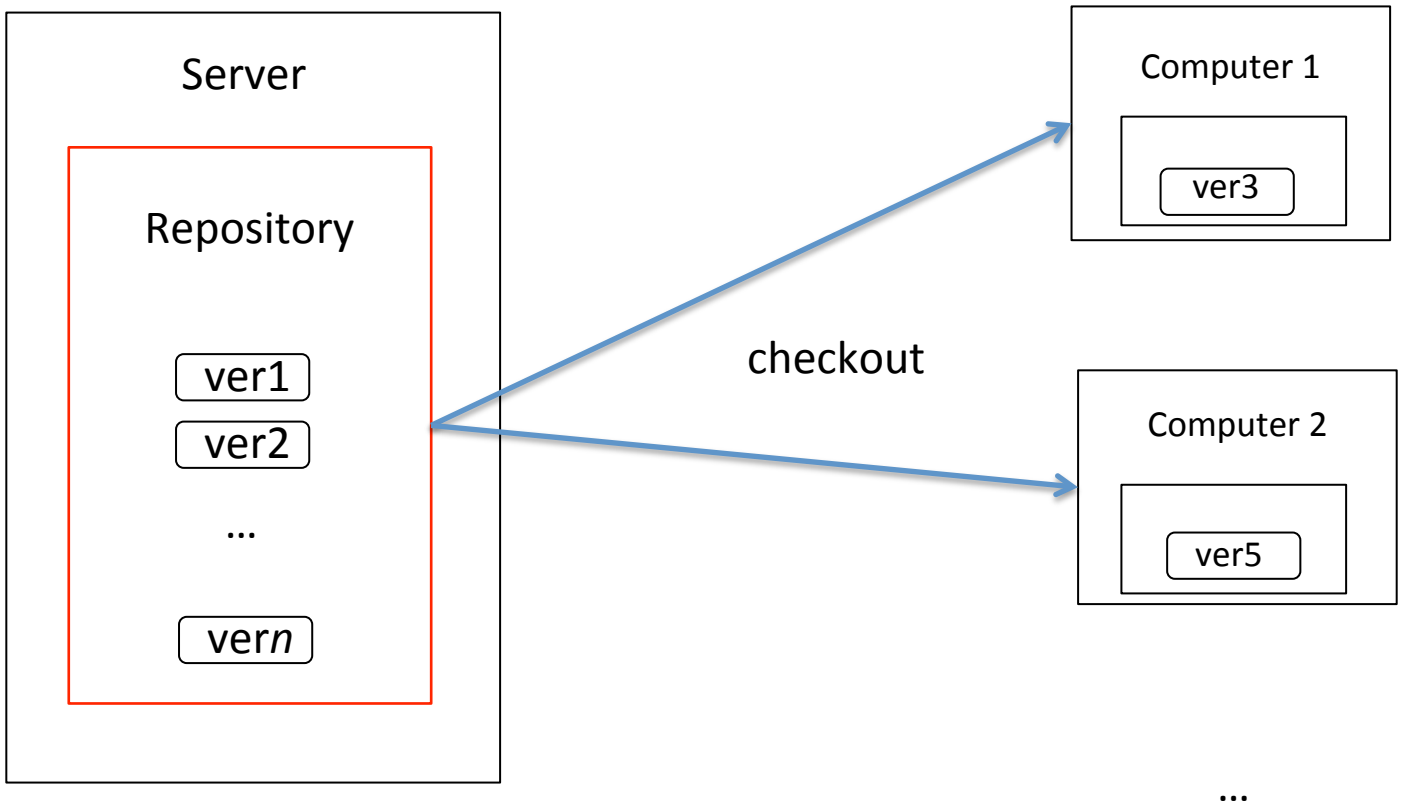
Version control system

A version control system is a program that can record multiple versions of a source file, storing information such as the creation time of each version, who made it, and a description of what was changed.

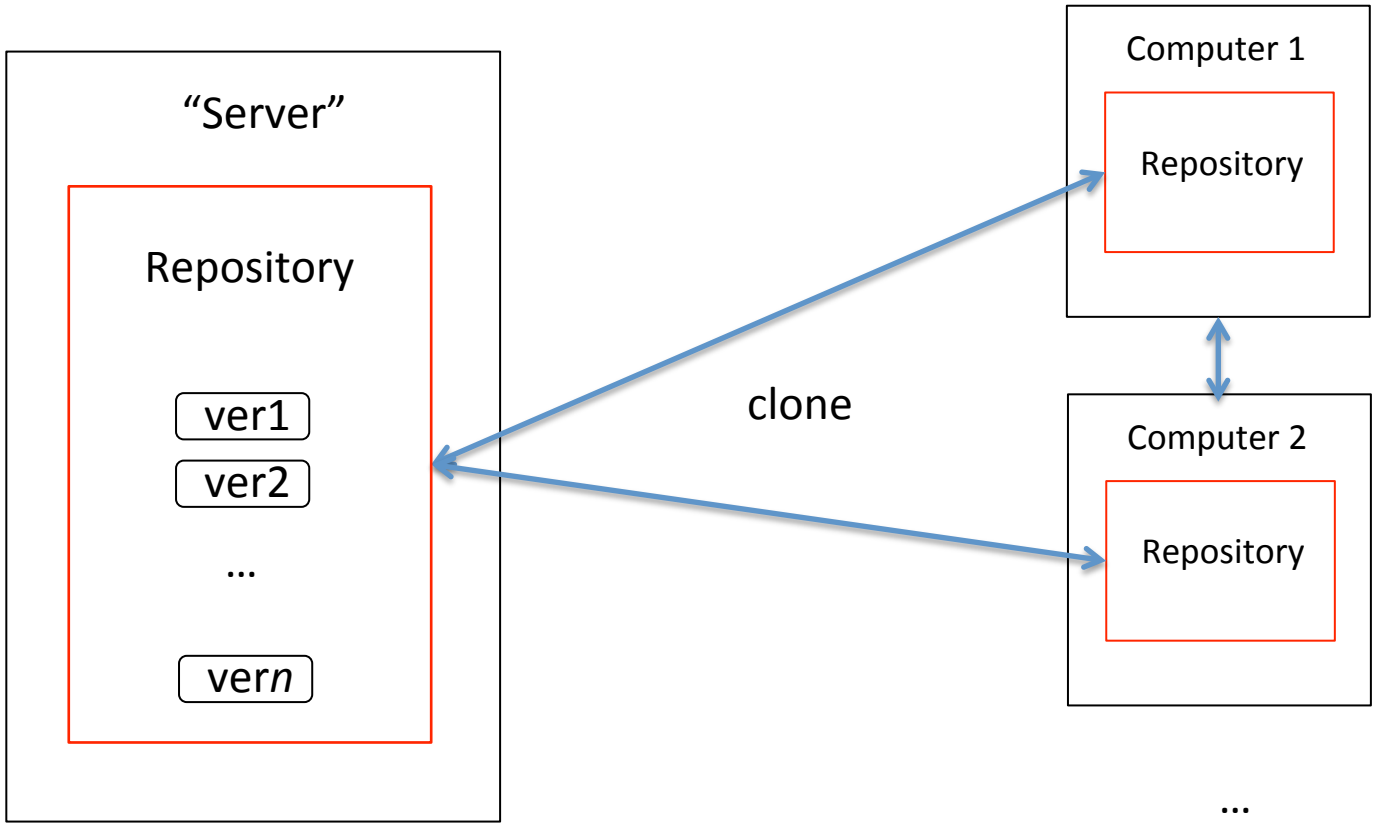
Why version control?



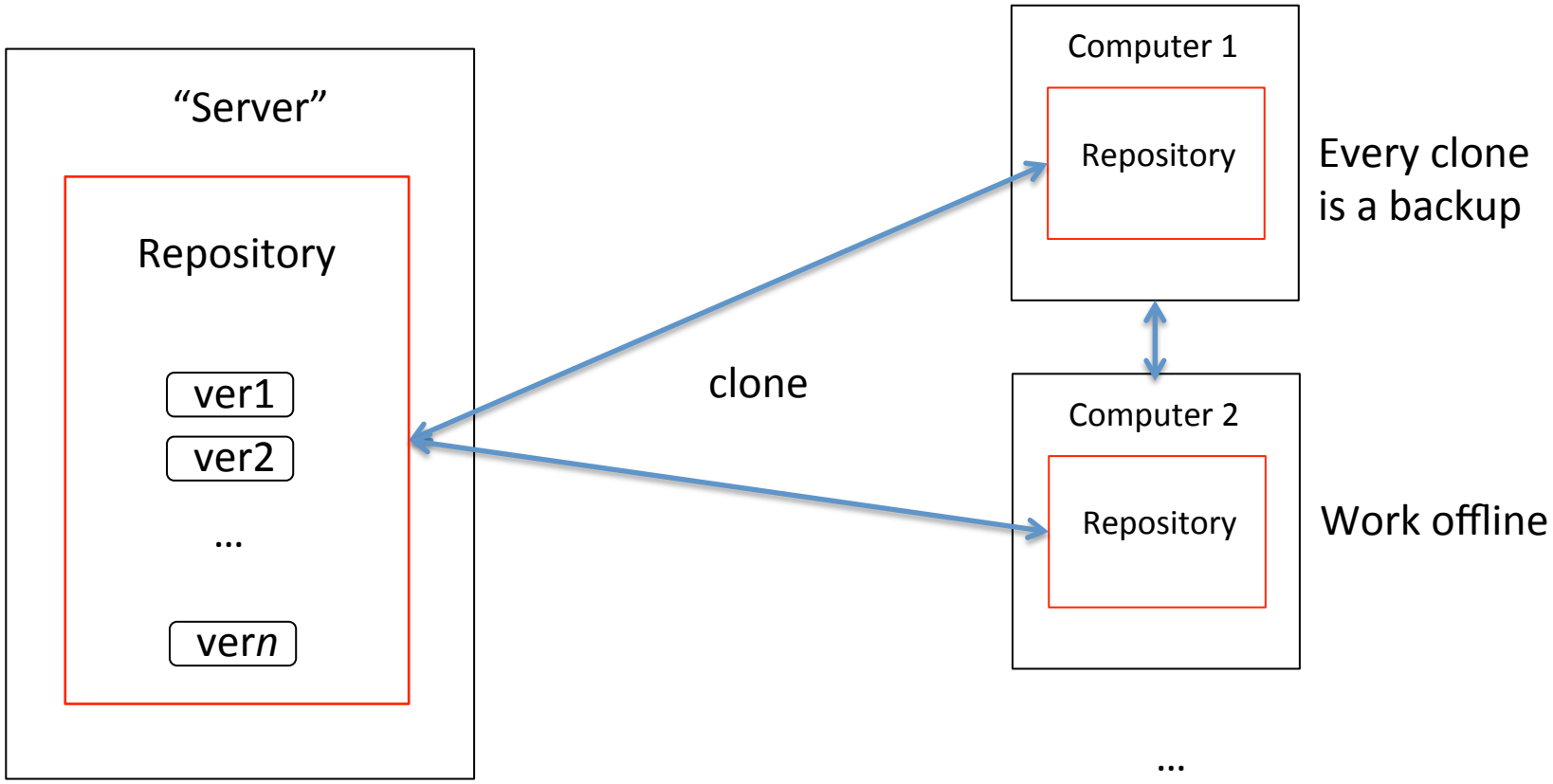
Centralized VCS



Distributed VCS



Distributed VCS



There is by convention – an upstream repo – to stay in sync

Getting started

- Install git ✓
- Configure git

Configure git

Git customization

- System /etc/gitconfig
- User $HOME/.gitconfig
- Project my_project/.git/config

Git commands to edit configuration:

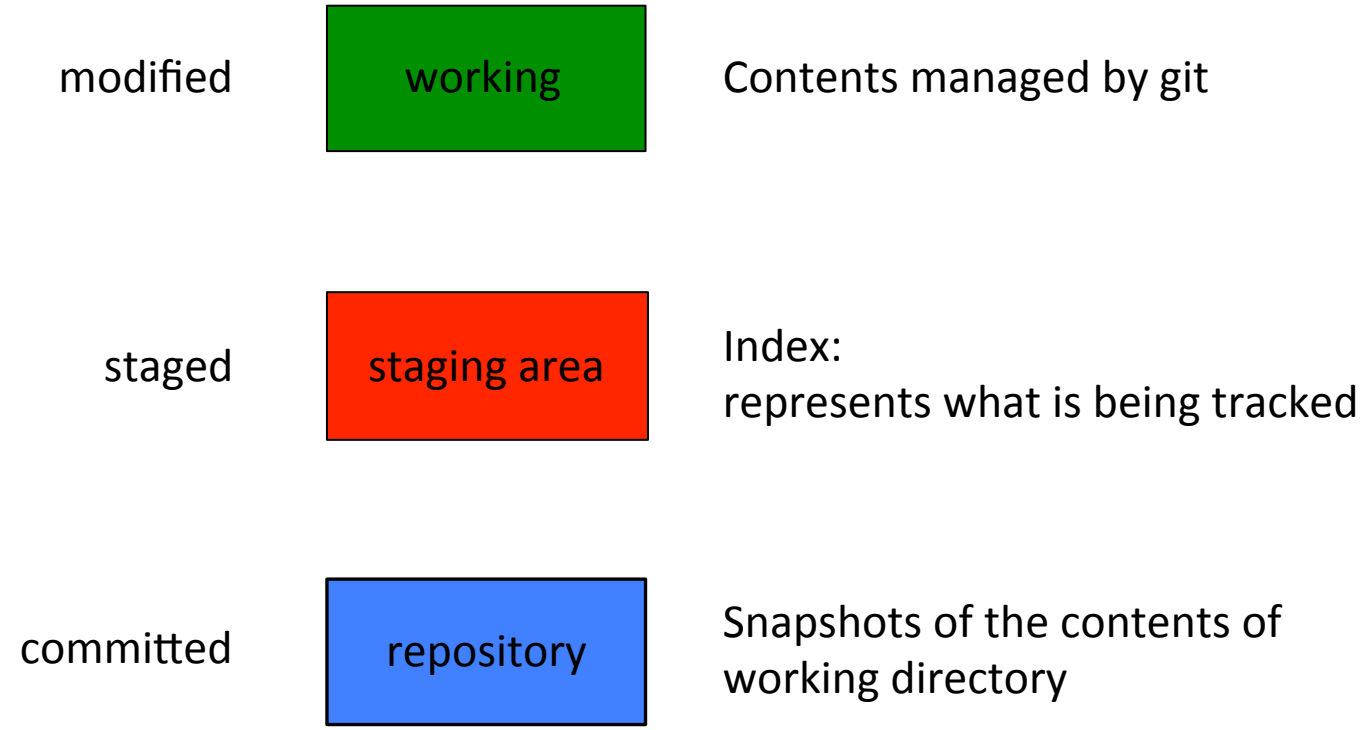
git config --system (system)
git config --global (user)
git config (project)

$ git config --global user.name "[name]"
Sets the name you want atached to your commit transactions
$ git config --global user.email "[email address]"
Sets the email you want atached to your commit transactions
$ git config --global color.ui auto
Enables helpful colorization of command line output

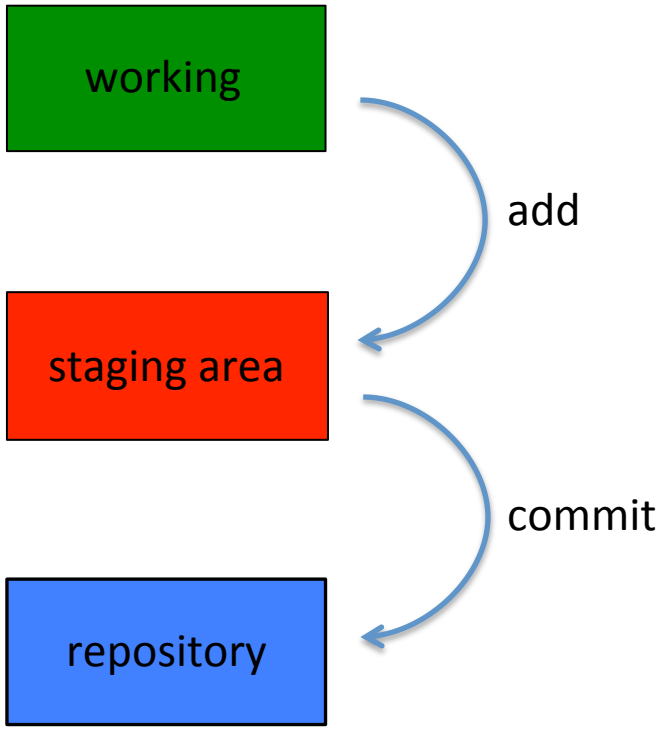
Getting started

- Install git ✓
- Configure git ✓
- Git concepts
- Demo
  - Initialize a repository
    - Working with repository
    - add, commit, etc…
  - Working with a remote repository
    - Pushing changes

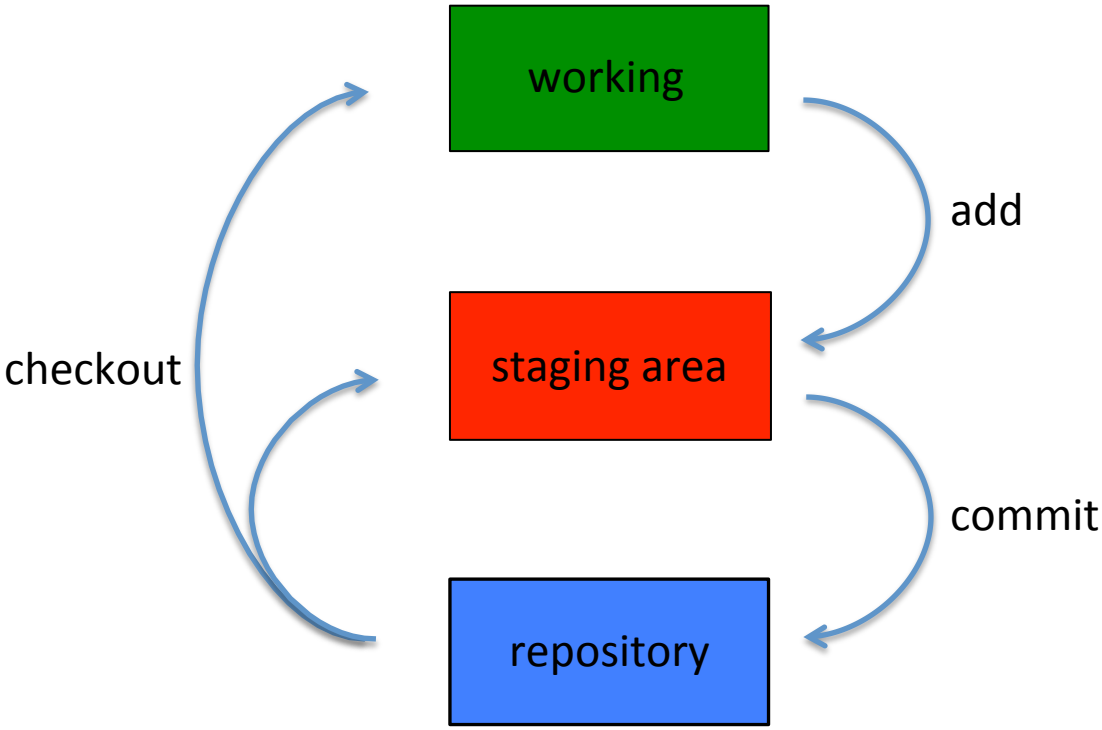
Git concepts



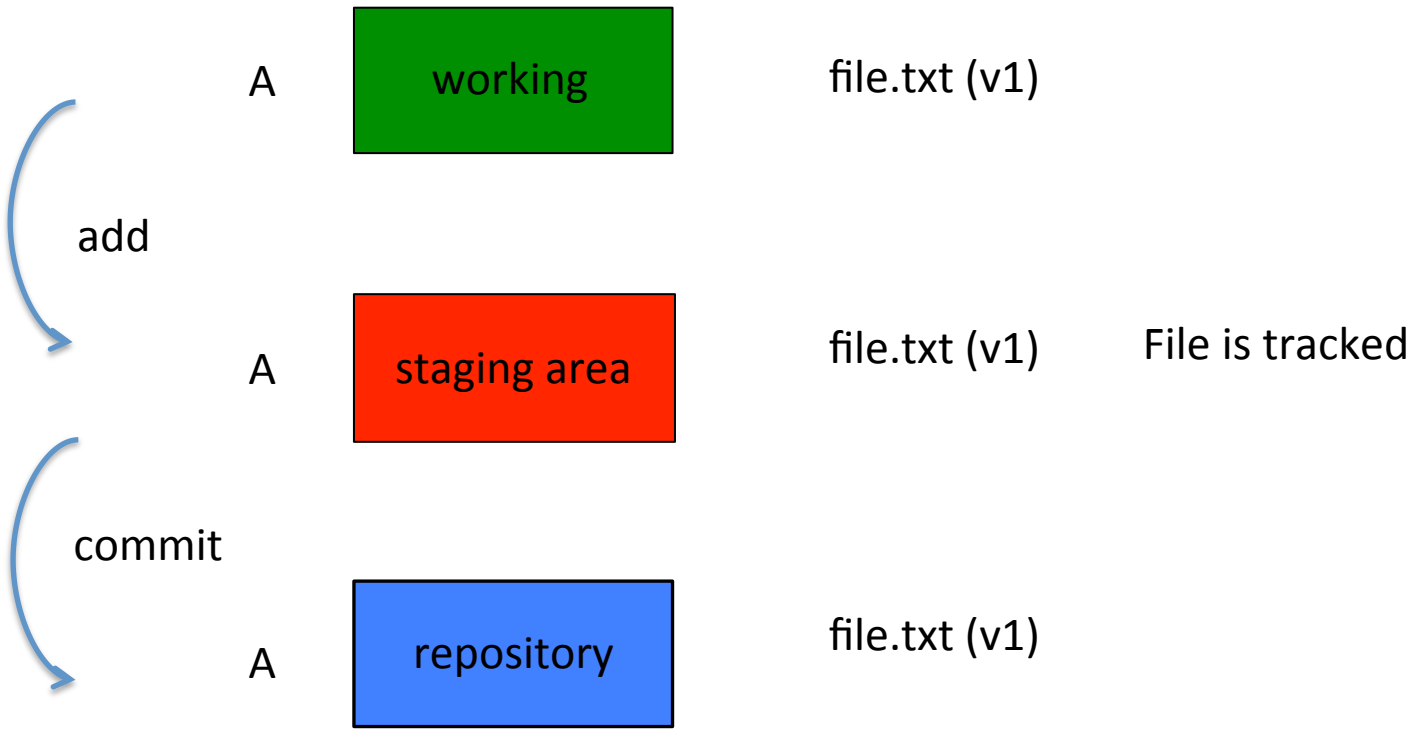
Git concepts



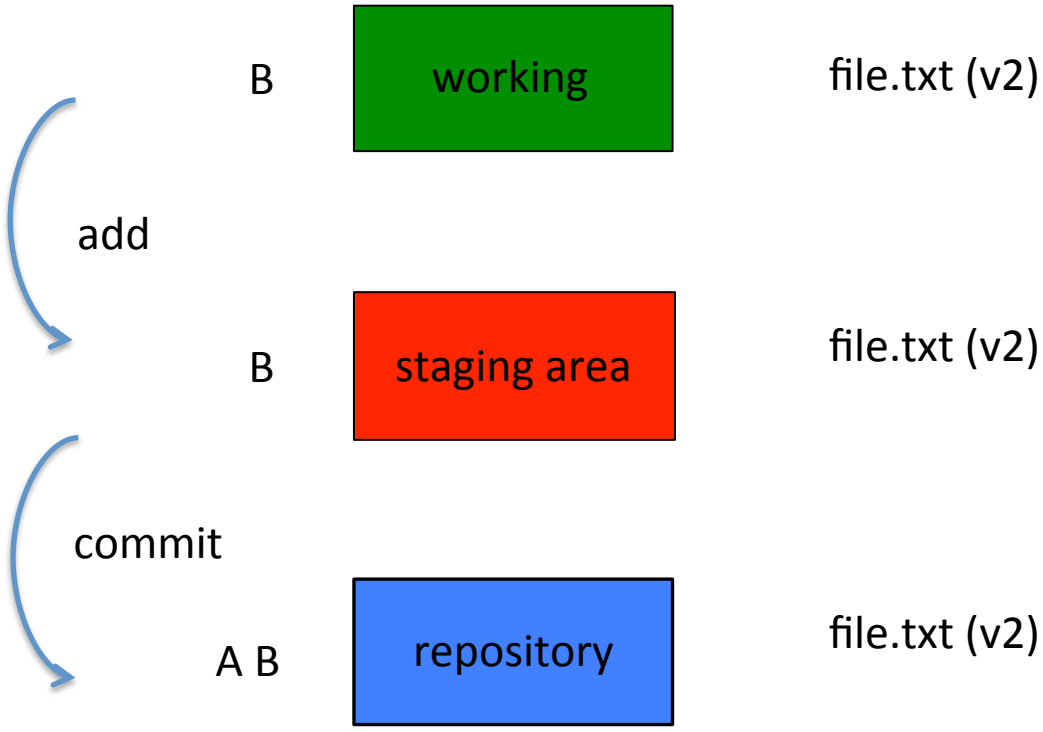
Git concepts



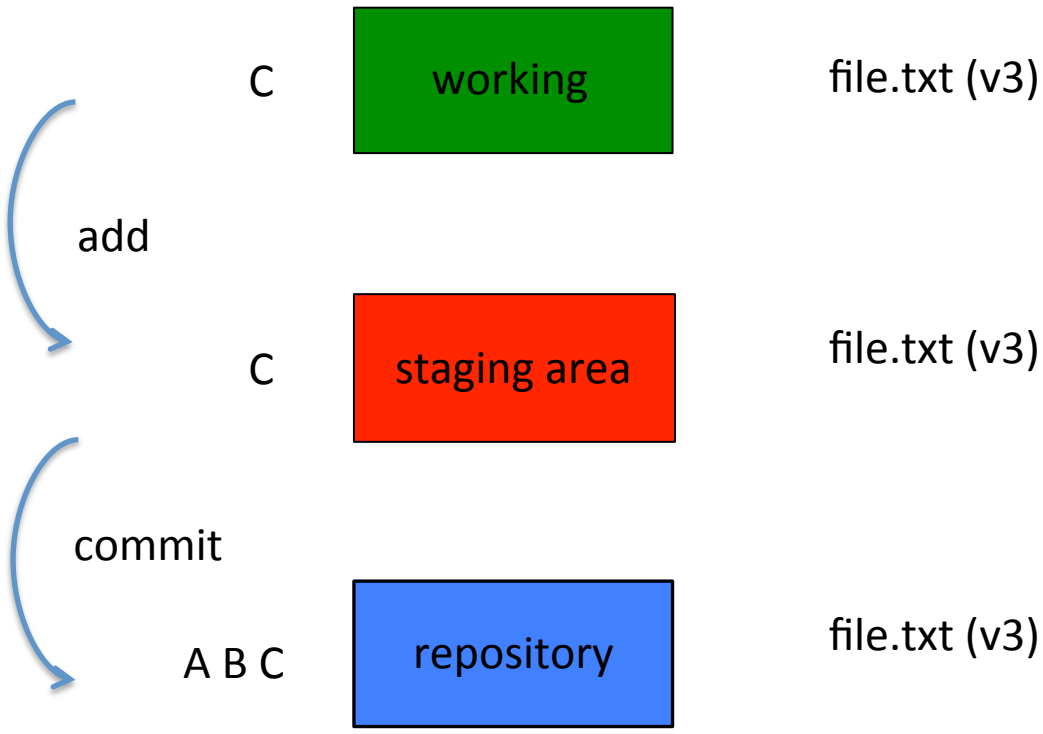
Git concepts



Git concepts



Git concepts



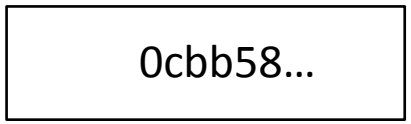
A, B and C are change sets

Git concepts

For each change set (i.e. each commit A,B,C):

- Git generates a checksum
- Git uses SHA-1 algorithm to create checksums
  - 40-character hexadecimal string
  - e.g. 1fbb5af06b9e4facff4170fc687ecdd143daad50

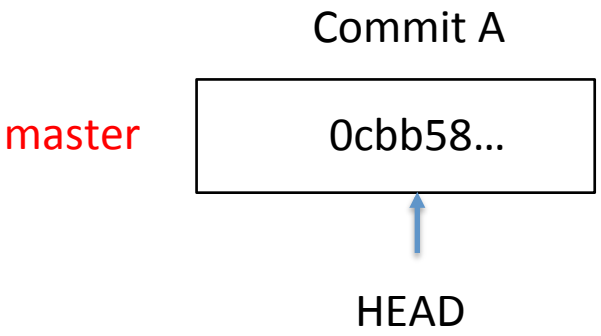
Git concepts



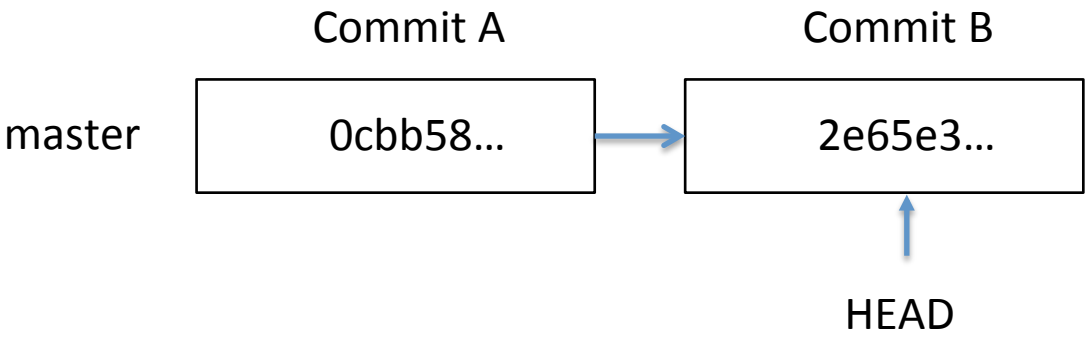
Commit:

Represented by 40-char hex string
Author
Timestamp
Message
Parent

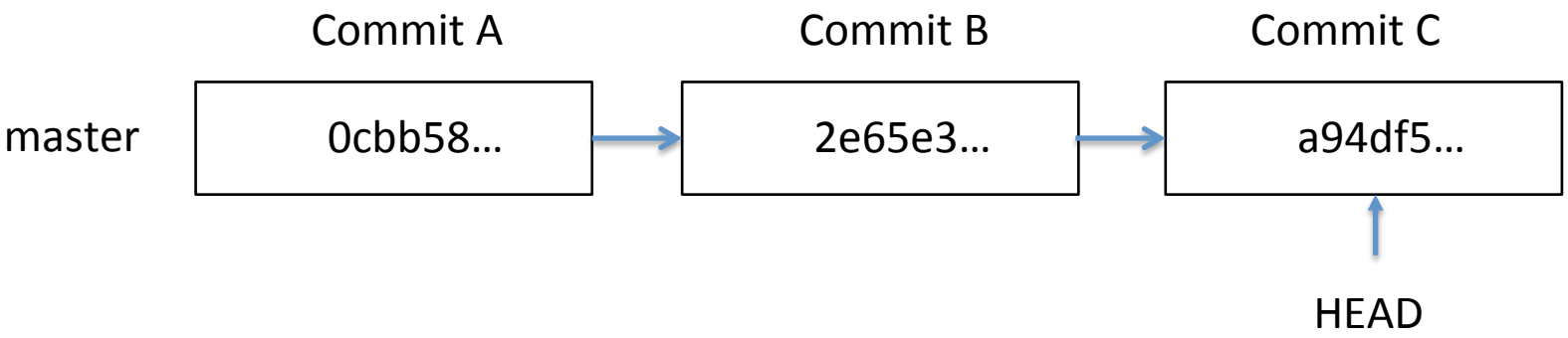
Git concepts



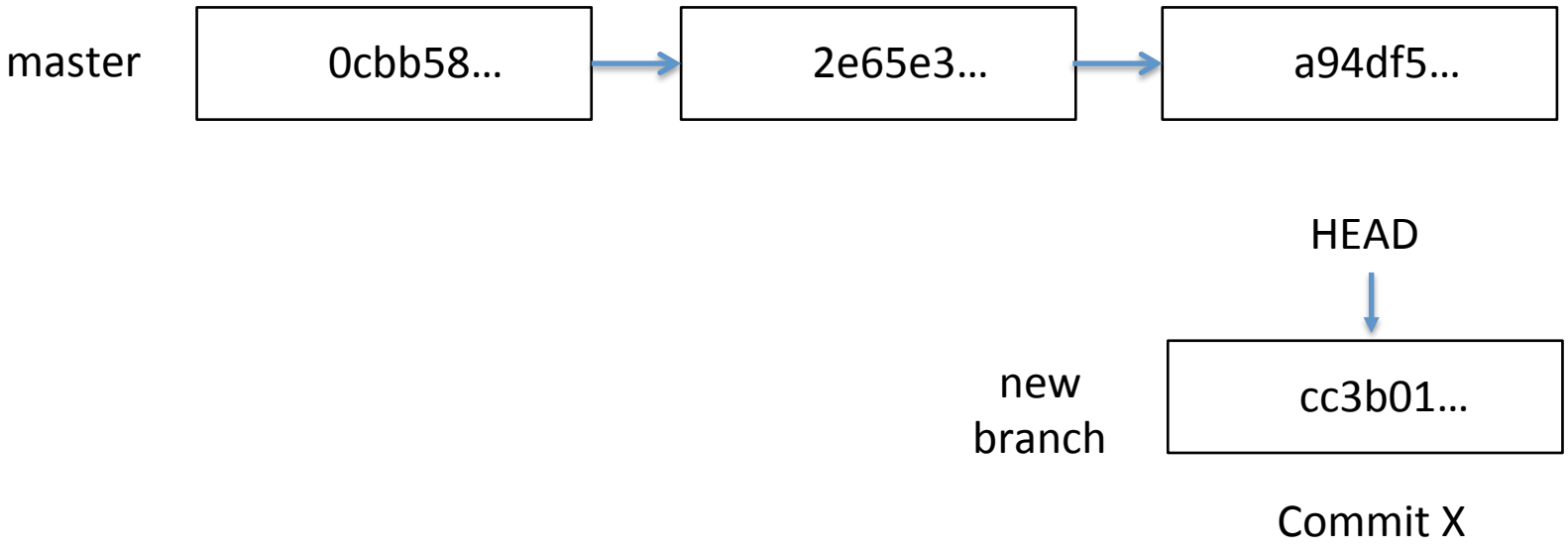
Git concepts



Git concepts

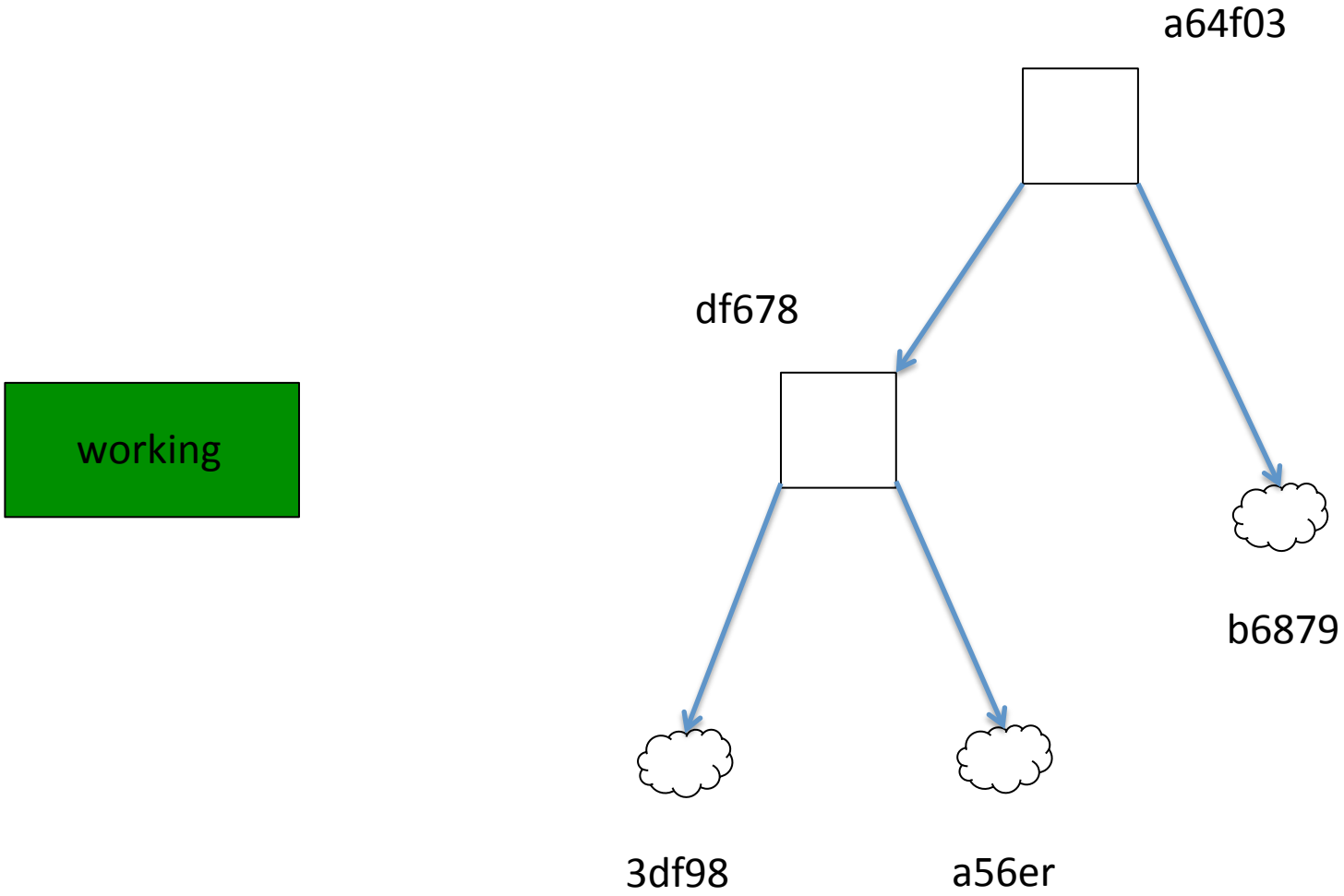


Git concepts



HEAD points to tip of current branch in repository

Git concepts



Demo

Common commands

Create repositories

$ git init [project-name]

Creates a new local repository with the specified name

$ git clone [url]

Downloads a project and its entire version history

Make changes

$ git status

Lists all new or modified files to be commited

$ git add [file]

Snapshots the file in preparation for versioning

$ git reset [file]

Unstages the file, but preserve its contents

$ git diff

Shows file differences not yet staged

$ git diff --staged

Shows file differences between staging and the last file version

$ git commit -m "[descriptive message]"

Records file snapshots permanently in version history

Group changes

$ git branch

Lists all local branches in the current repository

$ git branch [branch-name]

Creates a new branch

$ git checkout [branch-name]

Switches to the specified branch and updates the working directory

$ git merge [branch]

Combines the specified branch's history into the current branch

$ git branch -d [branch-name]

Deletes the specified branch

Review history

$ git log

Lists version history for the current branch

$ git log --follow [file]

Lists version history for a file, including renames

$ git diff [first-branch]...[second-branch]

Shows content differences between two branches

$ git show [commit]

Outputs metadata and content changes of the specified commit

References

- https://git-scm.com/doc
- http://gitref.org/