

Git crash course

Agenda

- Overview
- Basic operations

Overview

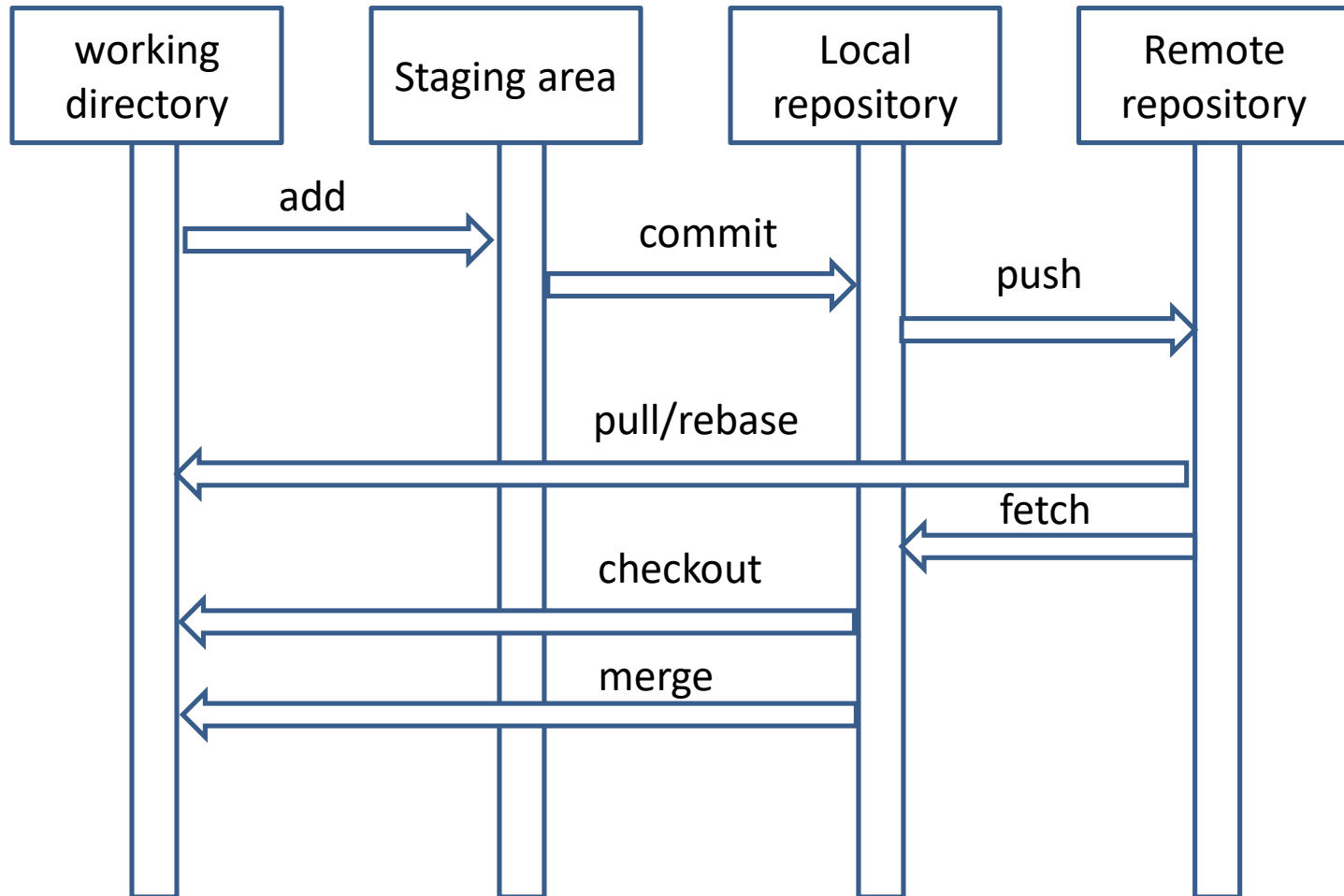
Git

- Git is a distributed version control system for tracking of changes in source code
- Designed to coordinate work among programmers
- Originally created in 2005 for development of the Linux Kernel

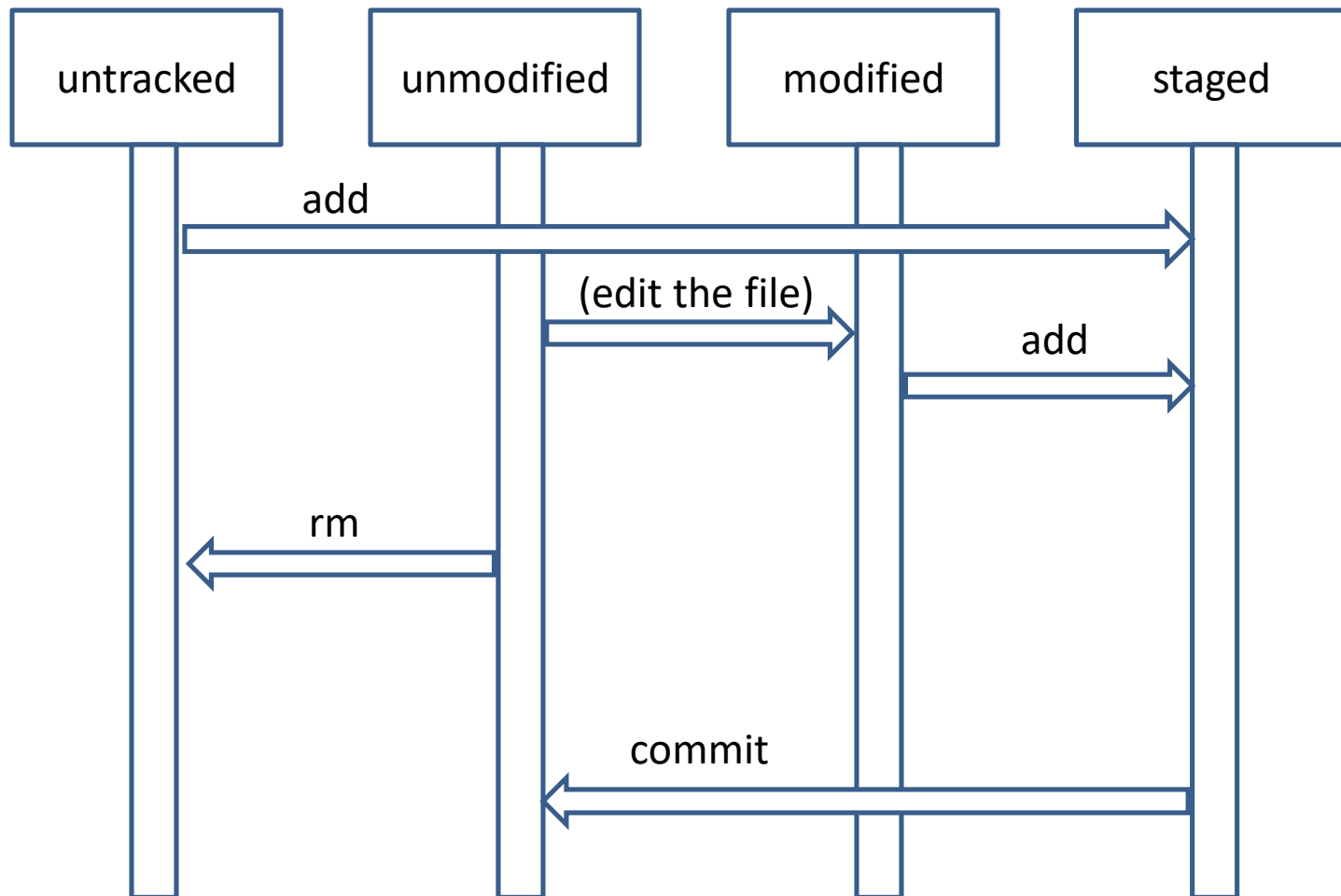
Git

- It is distributed as every copy of a repository is a repository on its own
- All history of changes is distributed among repository copies
- Repository contents are represented as binary objects that are packed by Git and provide more efficient storage

Git areas

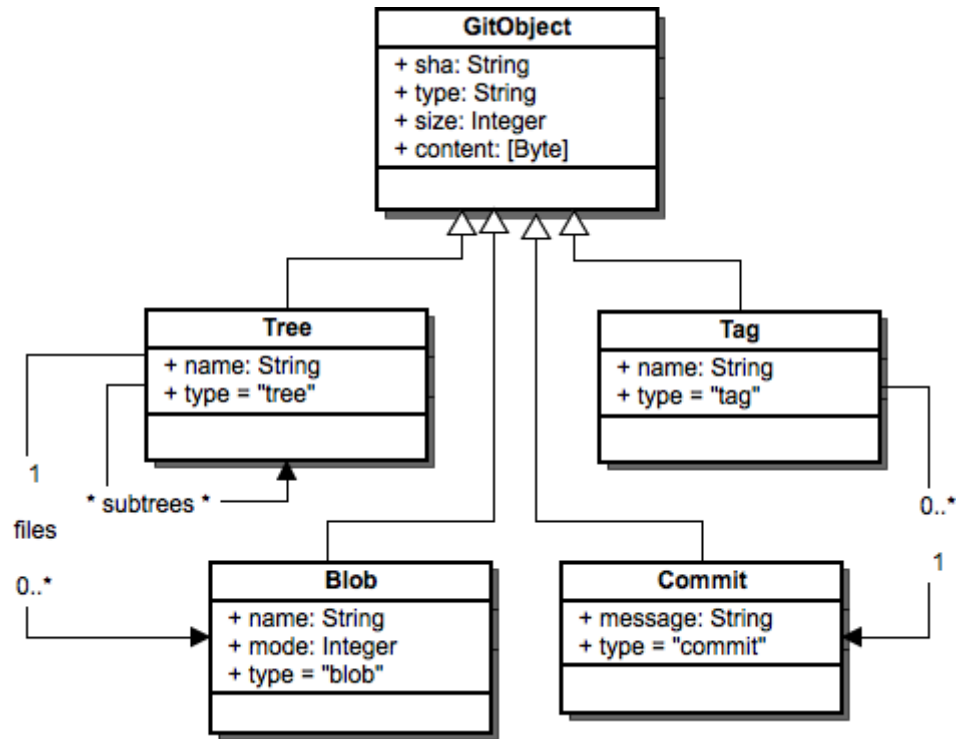


Git files



Files and directories can be ignored by git using proper patterns added to a .gitignore file in the Git repository

Git object database



courtesy: <https://www.aosabook.org/en/git.html>

Git vs SVN

- SVN has been also one of the most popular version control systems before the creation of Git
- SVN has a client-service architecture whereby the SVN repository and clients are different (in regard to Git)
- Branching in SVN is implemented by having each branch in a subdirectory while in Git there is a single directory managed by switching between branches

Git repositories

- There are a number of Git repository hosting platforms providing the possibility to create public or private repositories (with certain subscription options):
 - GitHub
 - GitLab
 - Atlassian BitBucket

Basic operations

Basic operations

git init	Create an empty repository
git clone <remote_repo>	Clone remote repository locally
git config	Provides a possibility to configure Git
git branch -a	View existing branches
git checkout -b <branch>	Creates a new branch from the current one
git checkout origin/<branch-name> -b <branch-name>	Creates a new branch from a target branch
git checkout <branch>	Switch to another branch
git branch -d git branch -D	Delete a branch (can be forced)
git add . git add <file1> <fileN>	Adds all or specified files to the staging area
git commit	Commits file from the staging area to the local repository
git status	Shows status of files in local repository

Basic operations

git push <remote-name> <branch-name> i.e. git push origin <branch-name>	Creates a remote branch from a local one
git push	Push all commits from local branches to the ones tracked in the origin remote
git pull	Pulls changes from the remote origin
git fetch origin <branch>	Pulls a remote branch to the local repository
git rebase <branch>	Rebase from another branch
git merge <branch>	Merge from another branch
git reset --hard HEAD	Throws away any staged and unstaged changes
git reset --soft HEAD^	Removes last commit from staging area
git log	View the commit history
git tag <tag_name>	Creates a lightweight tag

Basic operations

git revert HEAD	Reverts the last commit in the HEAD
git diff HEAD HEAD^	Compare the last two commits
git stash save	Saves the modifications of the current working directory to a separate area called stash
git stash pop	Pops changes from the stash
git stash clear	Clears the stash
git stash show	Shows files in the stash
git branch	Sets of commands used to manage branches (like adding a remote tracking branch)
git checkout <file>	Restore a file from HEAD
git apply <patch_file>	Applies the changes from a diff file
git remote	A set of commands used to manage git remotes
git cherry-pick	Cherry-pick a commit from a remote branch

Basic operations

gitk gitk <file>	Displays commit history (only history for certain files can also be displayed)
git gui	Displays a GUI tool for managing Git files
git mergetool	Starts the configured merge tool used to merge files
git difftool	Starts the configured diff tool used to merge files

Questions ?