



Version Control with Git

"FINAL".doc



FINAL.doc!



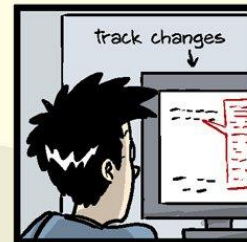
FINAL_rev.2.doc



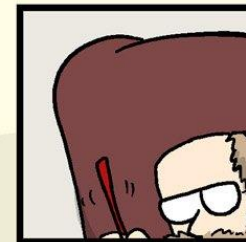
FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.##\$%WHYDID
ICOMETOGRADSCHOOL?????.doc

JORGE CHAM © 2012

Version control
helps avoid this →



What is Version Control?

Practice of **controlling, organizing, and tracking** different versions of computer files

- ◆ Allows you to revert to specific versions later
- ◆ Essential for software development, but useful for any evolving digital work (manuscripts, data, etc.)
- ◆ “Ctrl+Z” on steroids, with a complete history





Version Control with Git

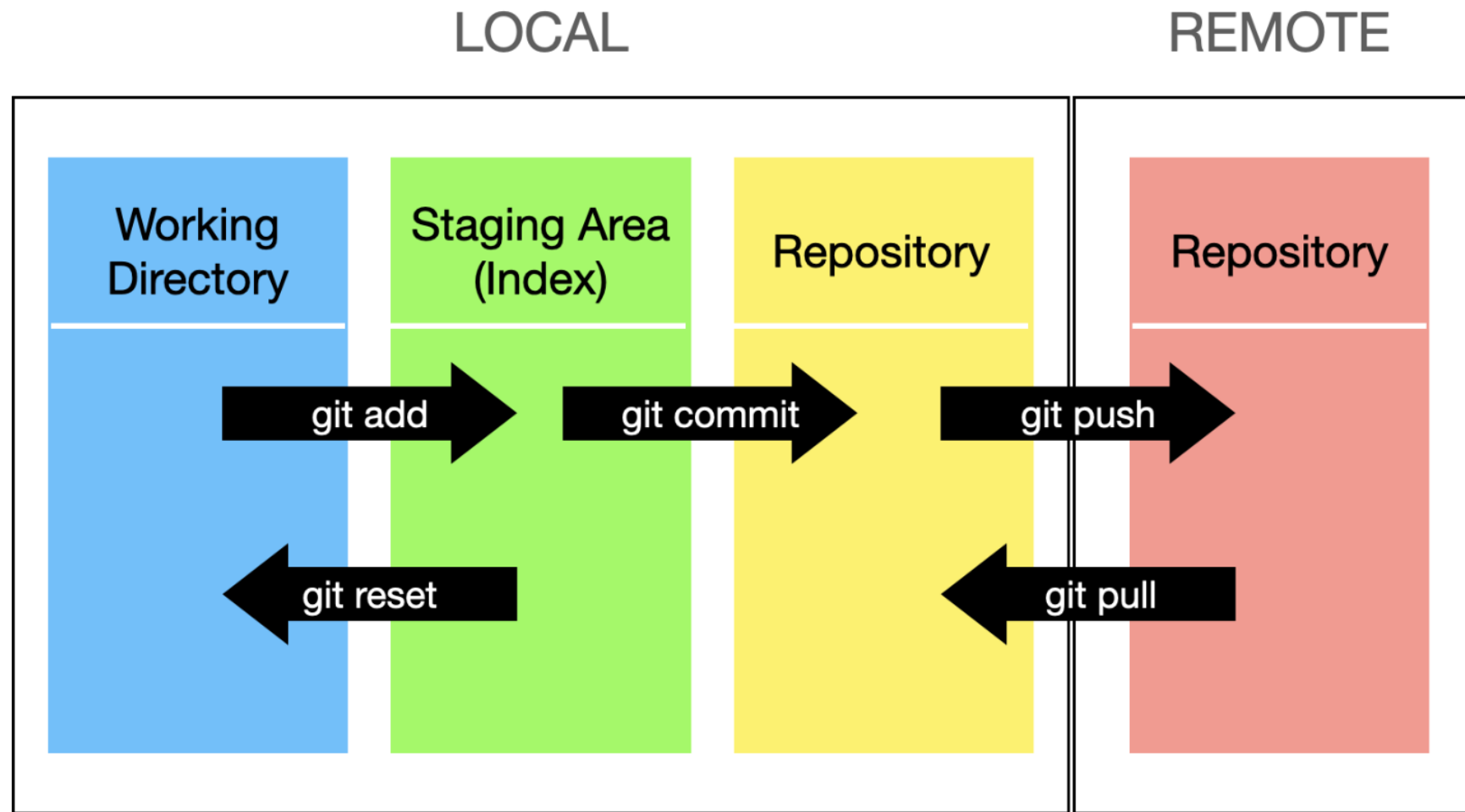
- ◆ Developed in 2005 by Linus Torvalds, the creator of Linux
- ◆ **Every developer has a full copy of the repository**
 - ◆ *With a full history of all changes made to files in the repository!*
- ◆ Widely used in the industry and academia
- ◆ Open source and free!



How does Git do this?

- ◆ **Working Directory:** Local files you see and edit
- ◆ **Staging Area:** A temporary area to prepare changes for the next commit
- ◆ **Repository (.git directory):** The hidden folder where Git stores all the history and metadata for your project
- ◆ **R Repository:** A remote repository where you can push your committed data for sharing, collaborating, showcasing skills

How do these structures interact?





Github!

- ◆ Web-based platform built around Git
- ◆ A remote location for your tracked files
- ◆ Allows teams to collaborate on code, track changes, and revert to previous versions
- ◆ **Beyond Version Control:** Also used for documentation, project management, and hosting websites (GitHub Pages)



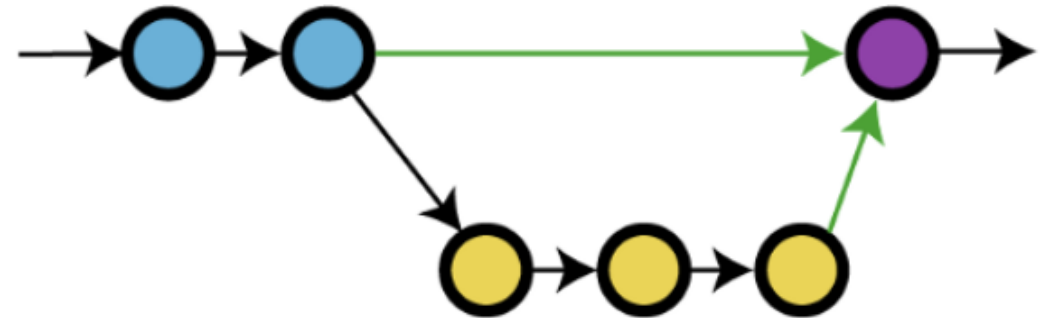
Branching & Merging with Git

Main branch



Isolates development of new features
from your main branch

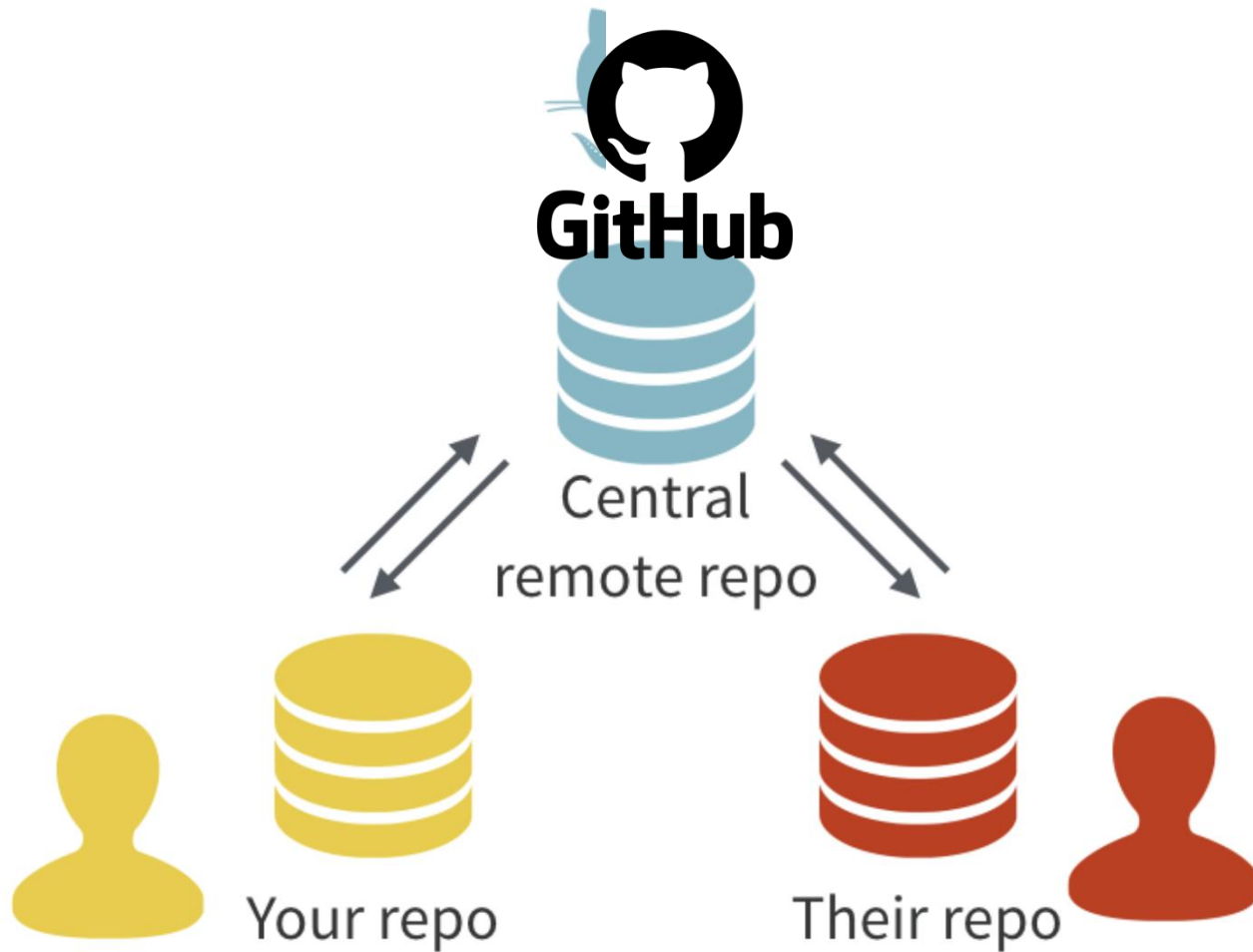
Main branch



Merge

Feature branch

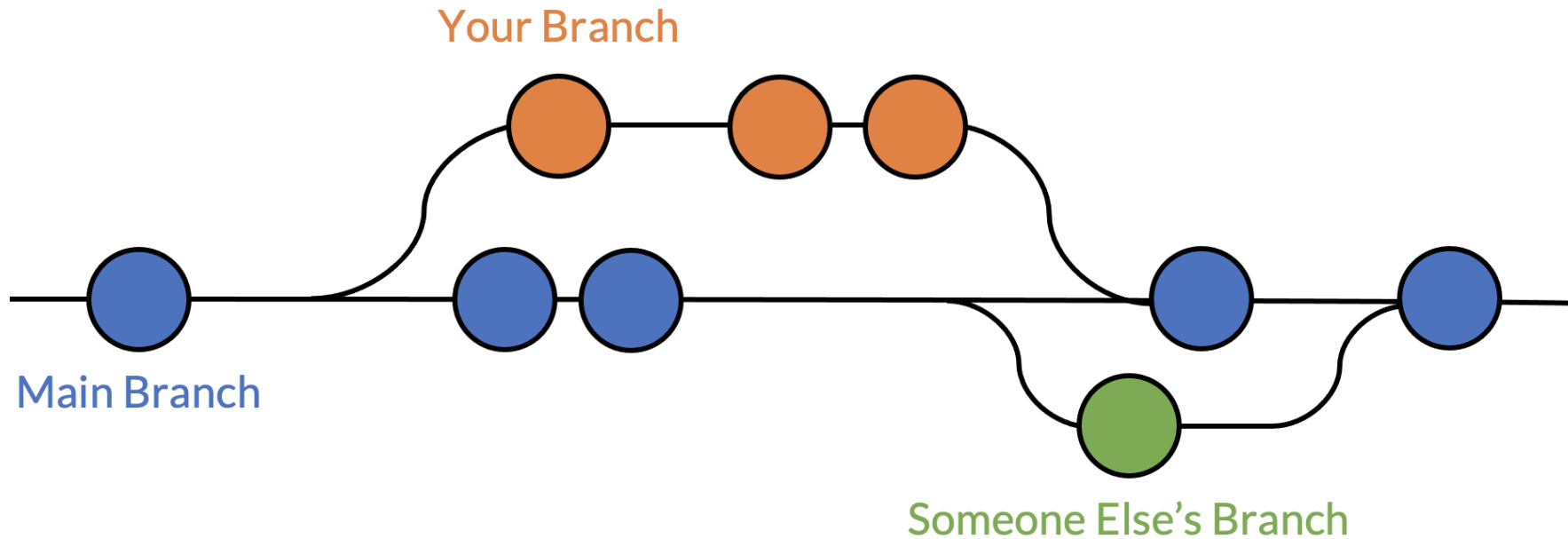
Collaborating on projects



All contributors have a **local** copy of the Git repository. These stay in sync through a central **remote** repository hosted on GitHub.



Collaborating through Branching & Merging



Branches let multiple collaborators make updates at the same time to later **merge** into the main branch.

Collaborating through Branching & Merging

