Software Engineering Essentials

Distributed Version Control

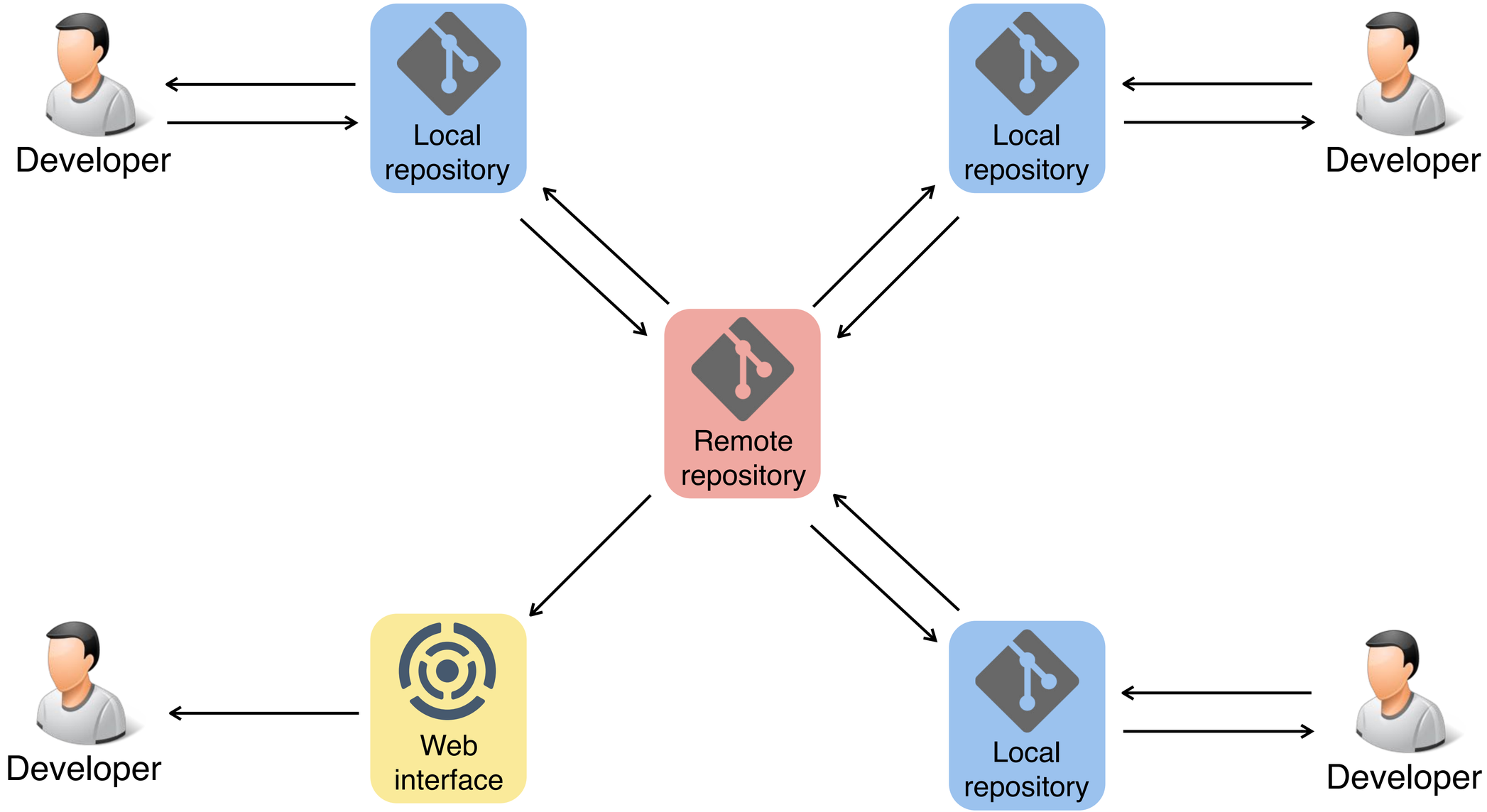
Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch
Chair for Applied Software Engineering — Faculty of Informatics

Learning goals

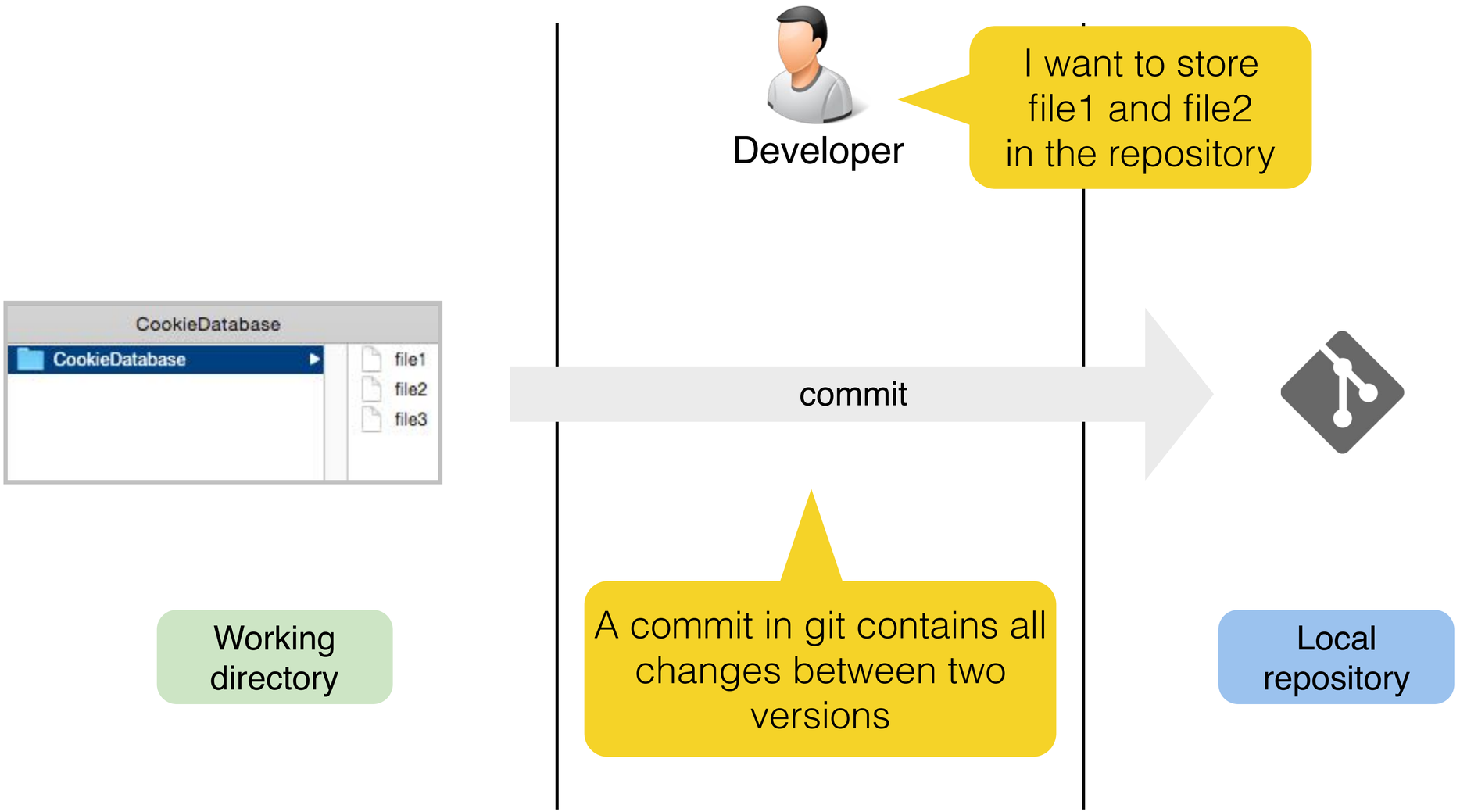
1) Apply distributed version control in SourceTree

2) Understand different git commands and best practices

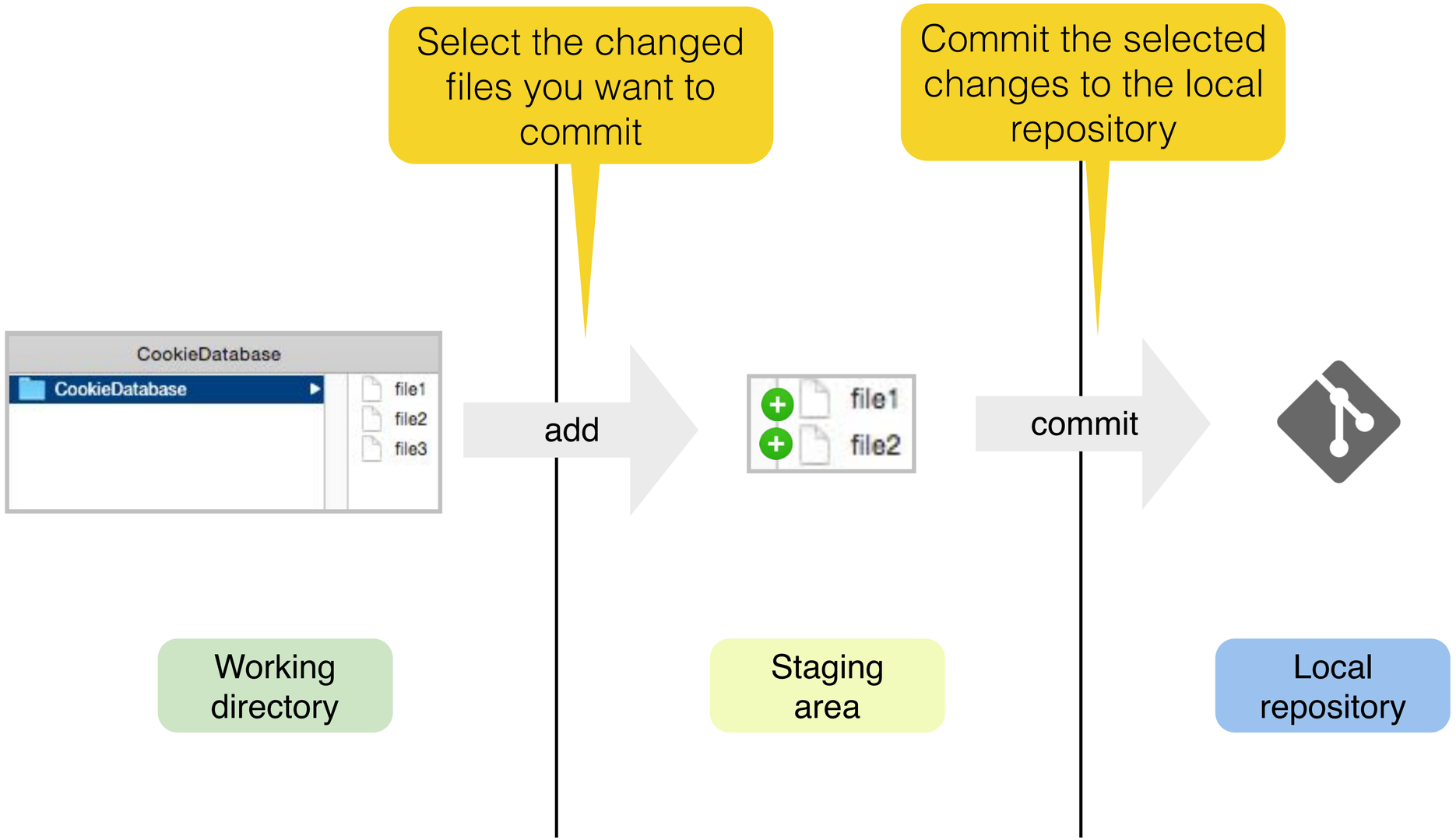
Distributed version control overview



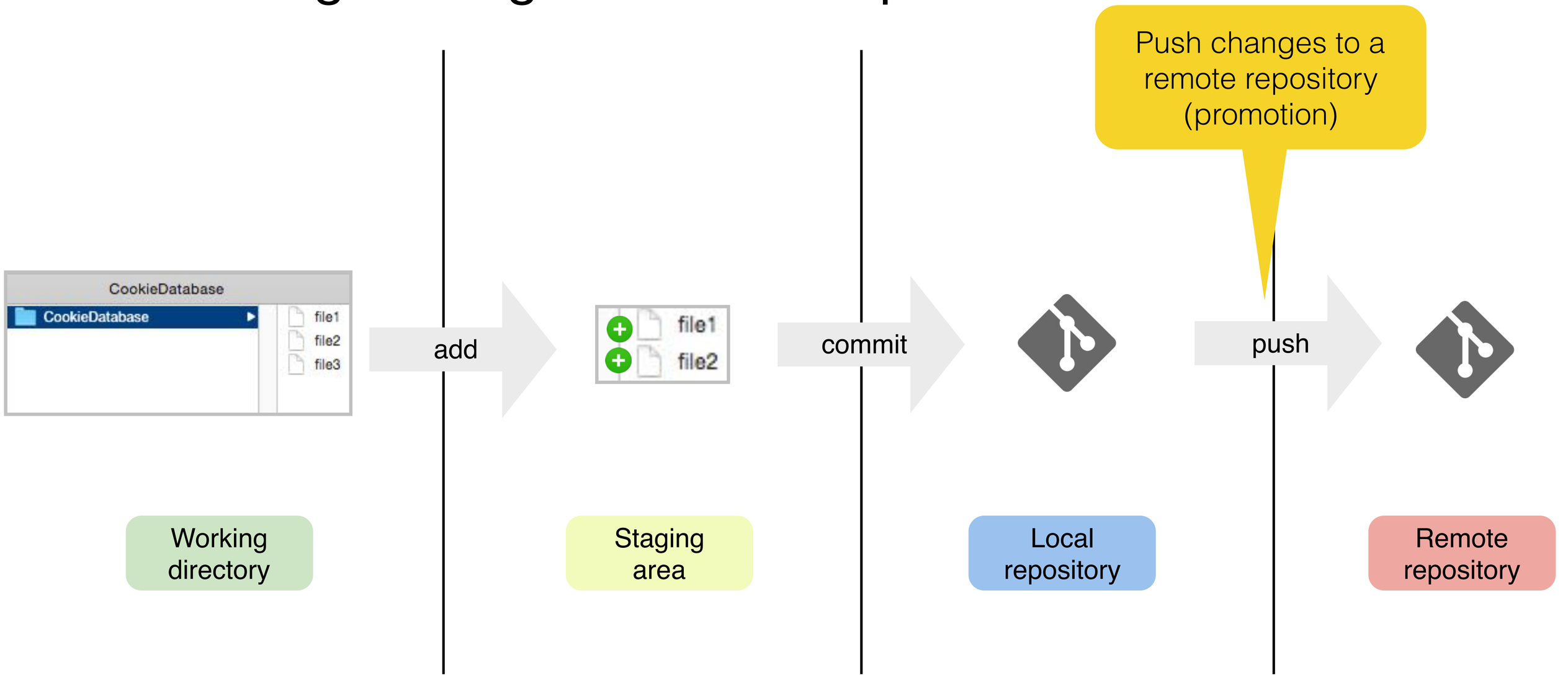
Share changes: commit



Share changes: stage + commit

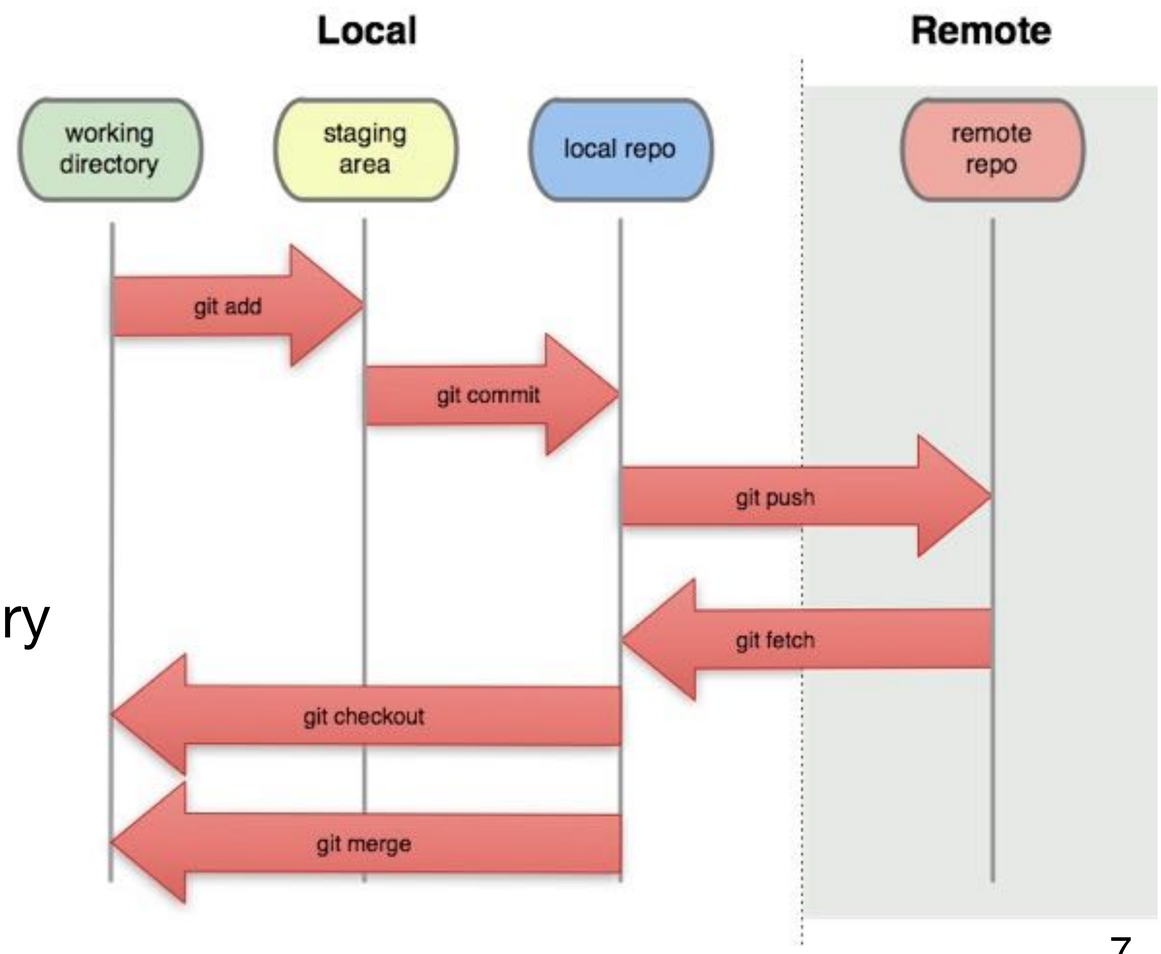


Share changes: stage + commit + push



Overview of git commands

- git add: add changes to the staging area
- git commit: commit selected changes of the staging area to your local repository
- git push: upload local commits to a remote repository
- git pull (fetch & merge): download and merge remote commits into your working directory
- git clone (fetch & initial checkout): clone a complete repository into a new working directory



Best practices

- Commit related changes
- Commit and push often, at least daily, but do not commit half done work
- Test changes before committing them
- Write meaningful and understandable commit messages

Best practices

- Commit related changes
- Commit and push often, at least daily, but do not commit half done work
- Test changes before committing them
- Write meaningful and understandable commit messages
- Do not use version control as a backup system for binary files
- Keep the local working directory of the repository up to date (regularly pull and push)
- Do not change published history