CIS241

System-Level Programming and Utilities

git (intro)

Erik Fredericks, frederer@gvsu.edu

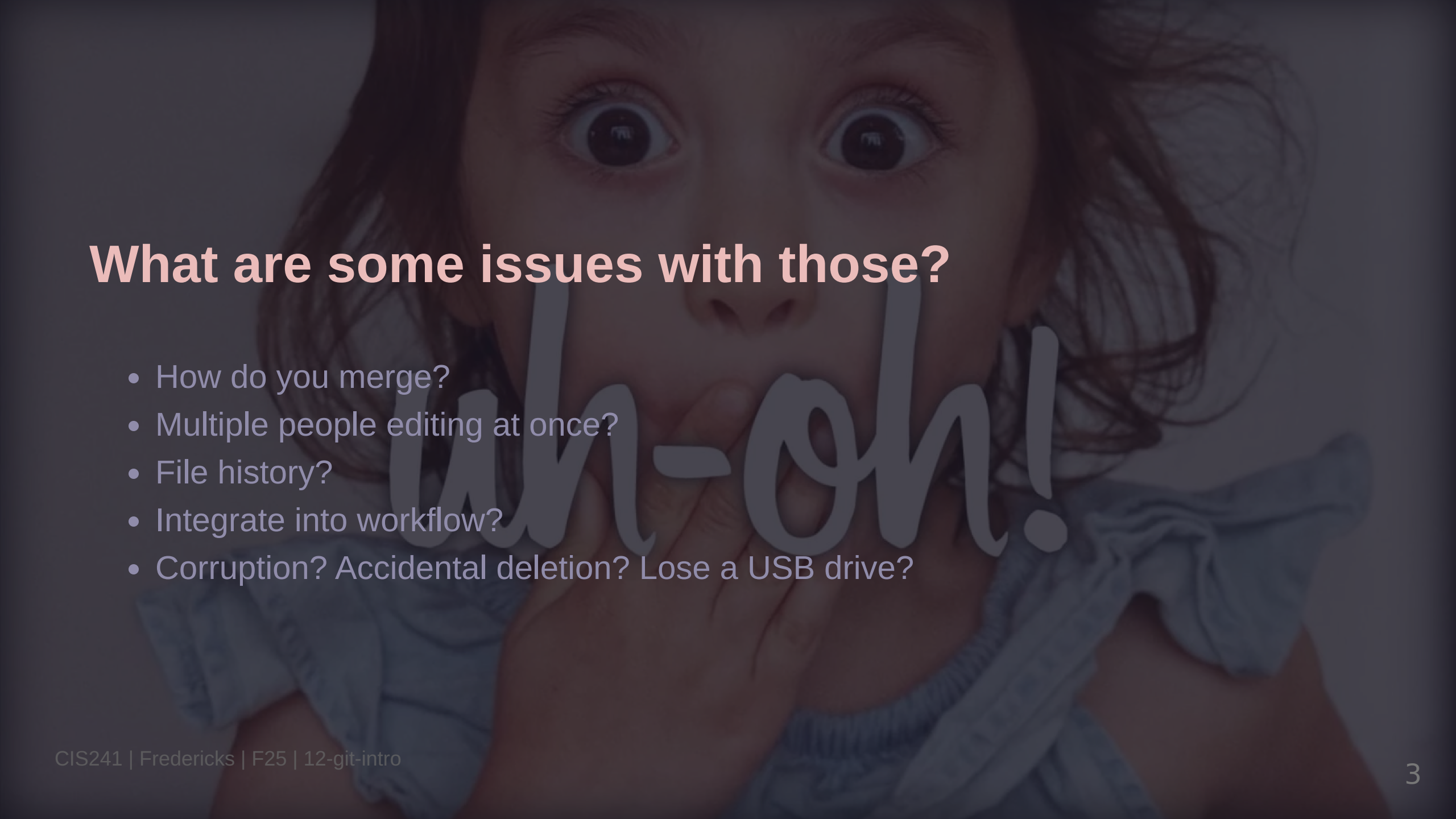
Fall 2025

> **Based on material provided by Erin Carrier, Austin Ferguson, and Katherine Bowers**

How do you share files?

Or backup/transfer files?

Or manage a group project?



What are some issues with those?

- How do you merge?
- Multiple people editing at once?
- File history?
- Integrate into workflow?
- Corruption? Accidental deletion? Lose a USB drive?

Source control!

Common options:

- **git**
- subversion

Version control systems

Software that tracks all changes to a repository of files

Benefits:

- Full history
  - Rollbacks, traceability
- Working **concurrently**
- Merging
- Files never fully deleted

Distributed version control systems

- Each user has a local copy of *all* files
  - Including full history!
- Allows for working offline
- In theory, should make merging easier

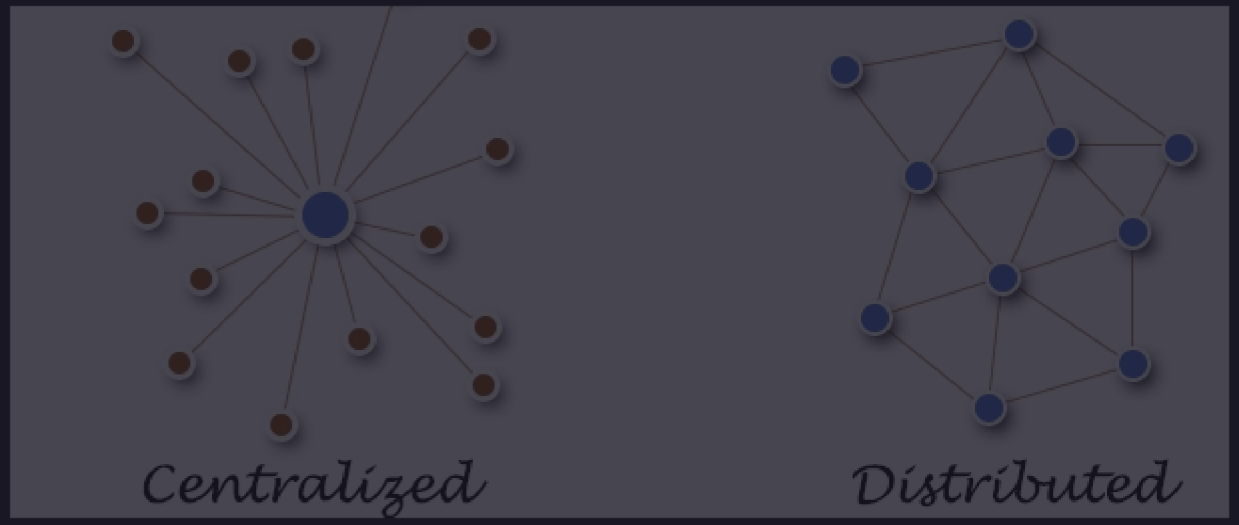
Common version control systems

Distributed

- git
- mercurial

Centralized

- subversion
- perforce



Wot about GitHub?

git -- *local*

- software
- handles version control
- operates locally

GitHub -- *remote*

- Website/server
- Stores git repositories
- Additional features
  - Public sites
  - GitHub Actions

Touch of history on git

git:

- **The most widely used source code management tool: 42.9% of professional software developers use it as their primary source control system.**
  - 90% respondents say they use Git (2021 StackOverflow survey)
- **Developed by Linus Torvalds in 2005 for use developing the Linux kernel**
- **GitHub was developed in 2008, and provides free (and paid) Git repository hosting**
  - Currently owned by none other than Microsoft!

Personal notes

git is *complicated*

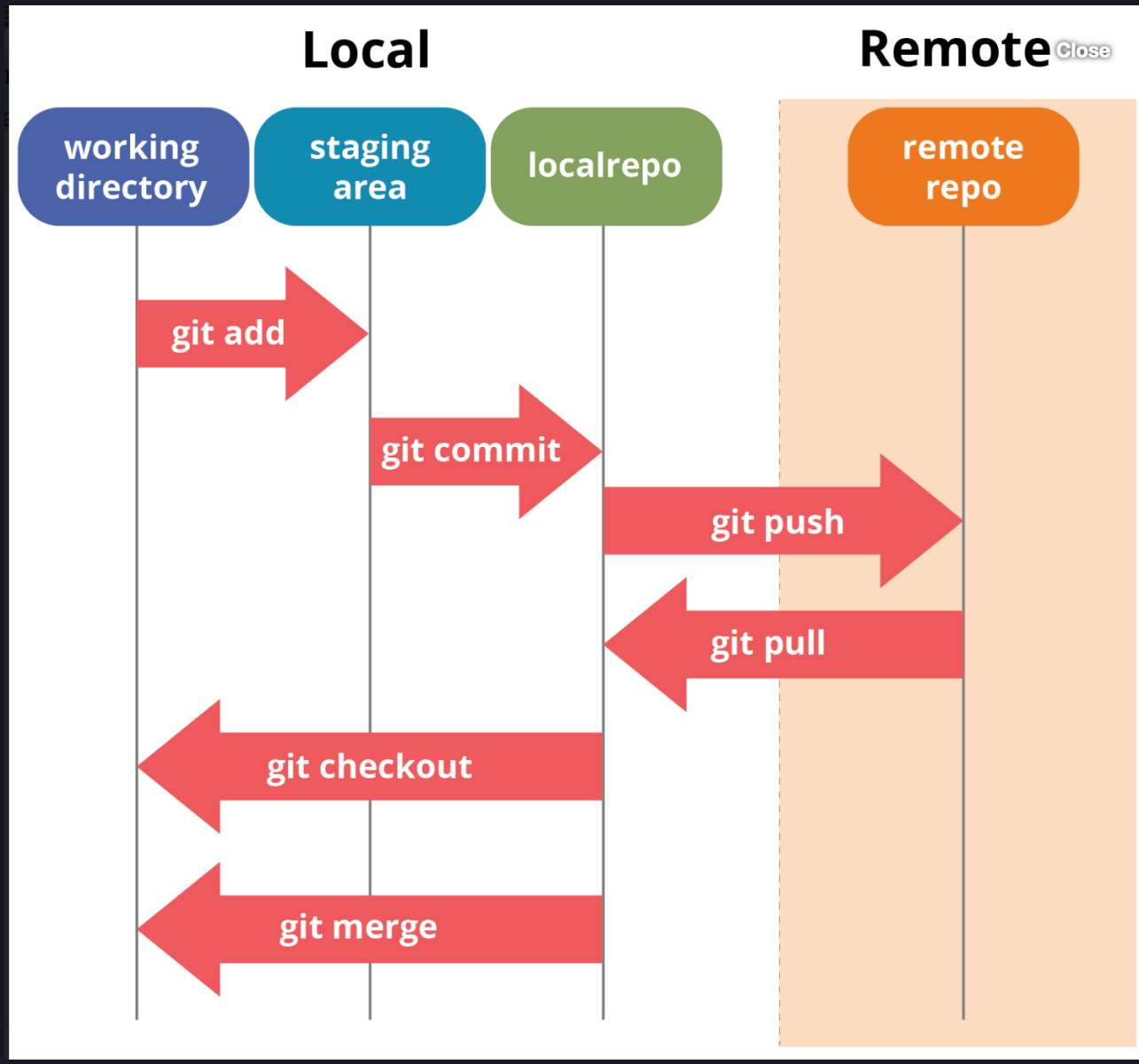
You will mess things up / break things

- I still do
- Most mistakes are recoverable
- Learn **how** to recover things!

Don't worry too much about the details

- Get comfortable with the big picture
- Look things up when you need (or are curious)

The process


Local
Remote
Close
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull
git checkout
git merge

git: repositories

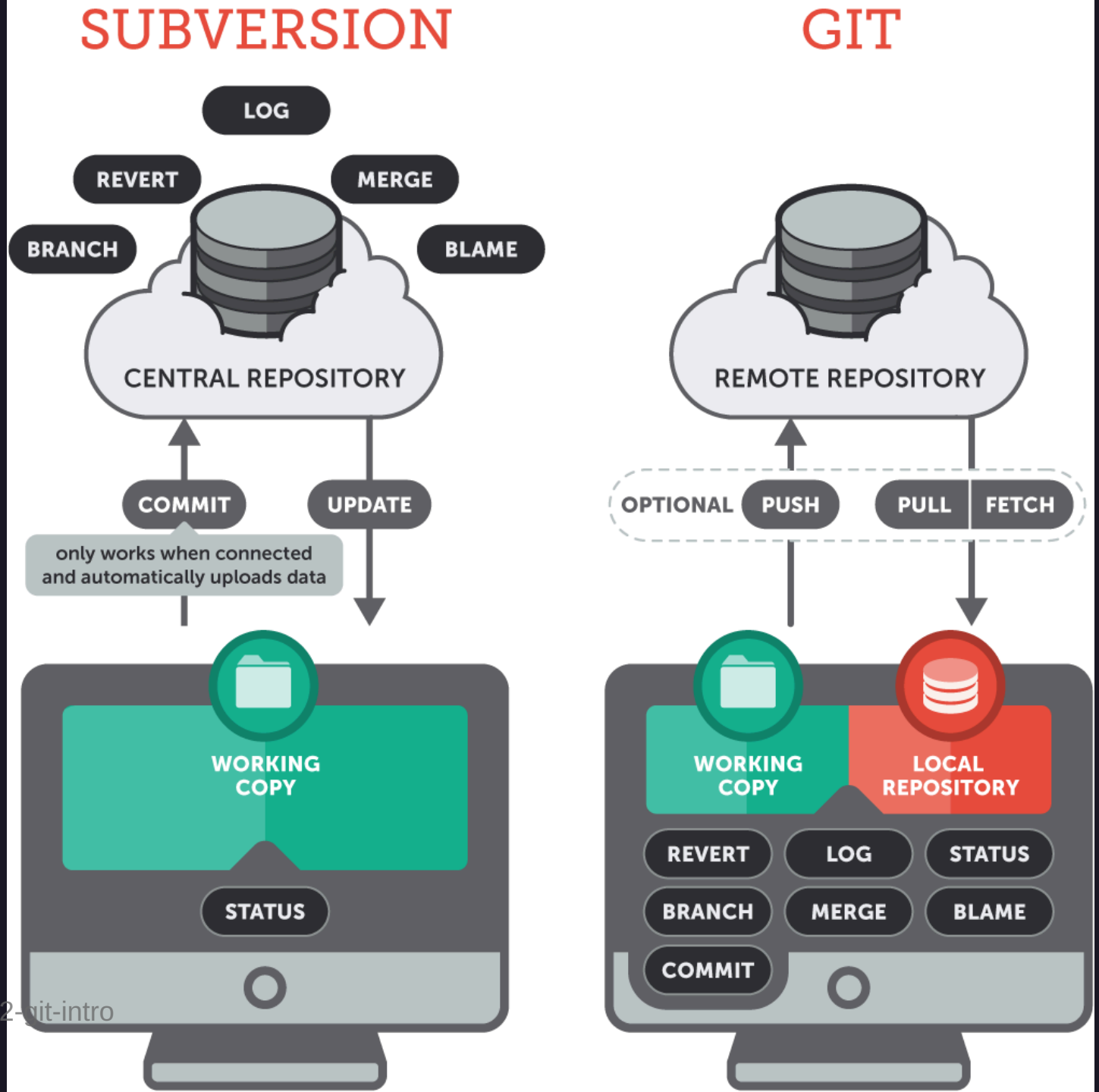
Git repositories contain:

- A set of commit objects, and
- A set of references to commit objects (heads)

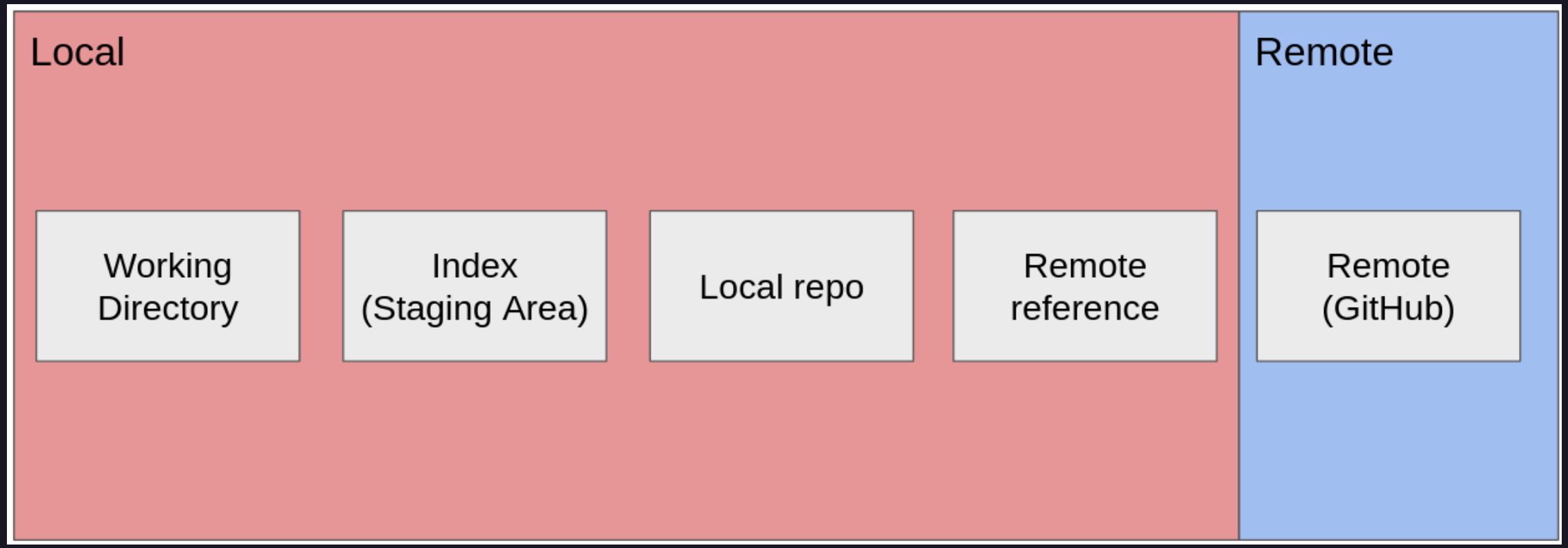
Git repositories are stored:

- In a .git sub-directory in the same directory as the project
- There is no central repository server

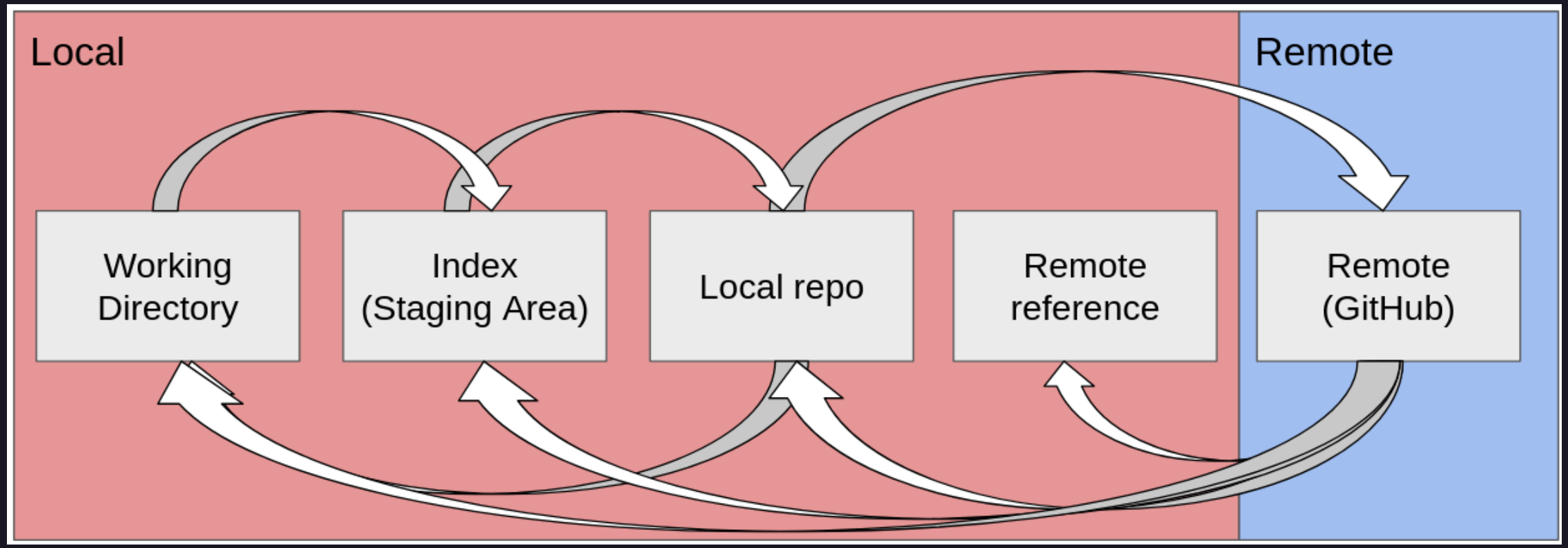
Git is distributed!


SUBVERSION
LOG
REVERT
MERGE
BRANCH
BLAME
CENTRAL REPOSITORY
COMMIT
UPDATE
only works when connected
and automatically uploads data
WORKING
COPY
STATUS
GIT
REMOTE REPOSITORY
OPTIONAL
PUSH
PULL
FETCH
WORKING
COPY
LOCAL
REPOSITORY
REVERT
LOG
STATUS
BRANCH
MERGE
BLAME
COMMIT

Where we're going



Where we're going



Today's work

1. Install git

- Probably already installed, but you never know
  - https://github.com/git-guides/install-git
  - We are **not** using GitHub Desktop! Terminal only!
  - To test, try `git version`

2. **Create a GitHub account - you'll be sending me your username**

3. **Configure git on your machine**

- `git config --global user.name "Your name"`
- `git config --global user.email "Your email"`