

Git Fundamentals

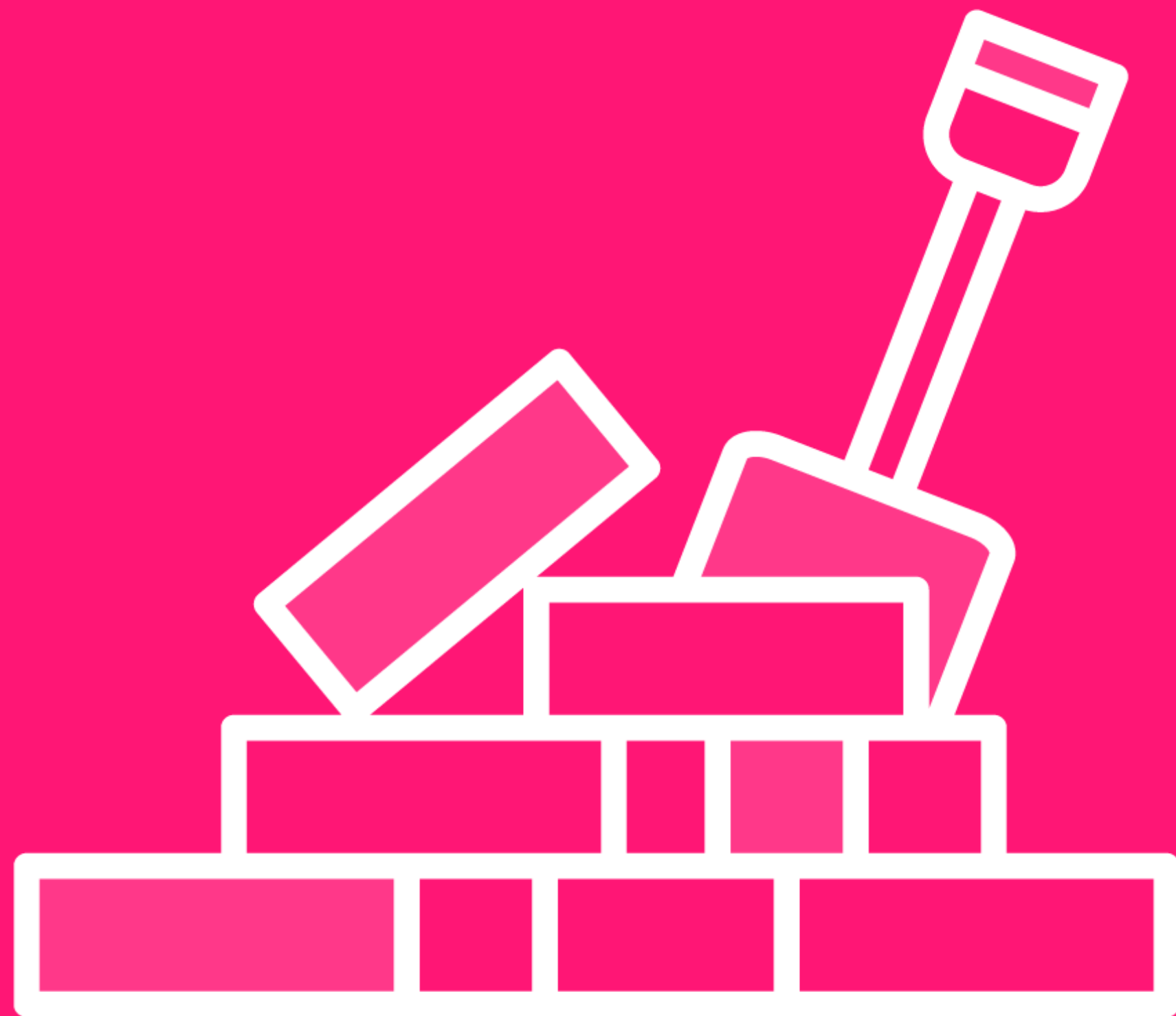
Thinking in Git



Aaron Stewart

Senior Program Manager, GitHub

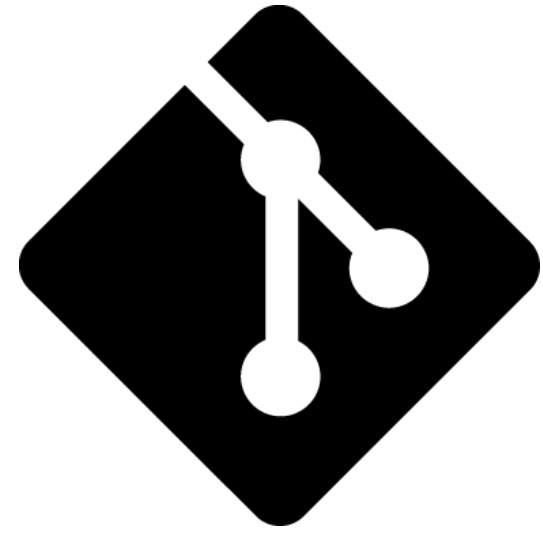
@a-a-ron | <https://www.linkedin.com/in/aaron-stewart1>



Goal of This Course

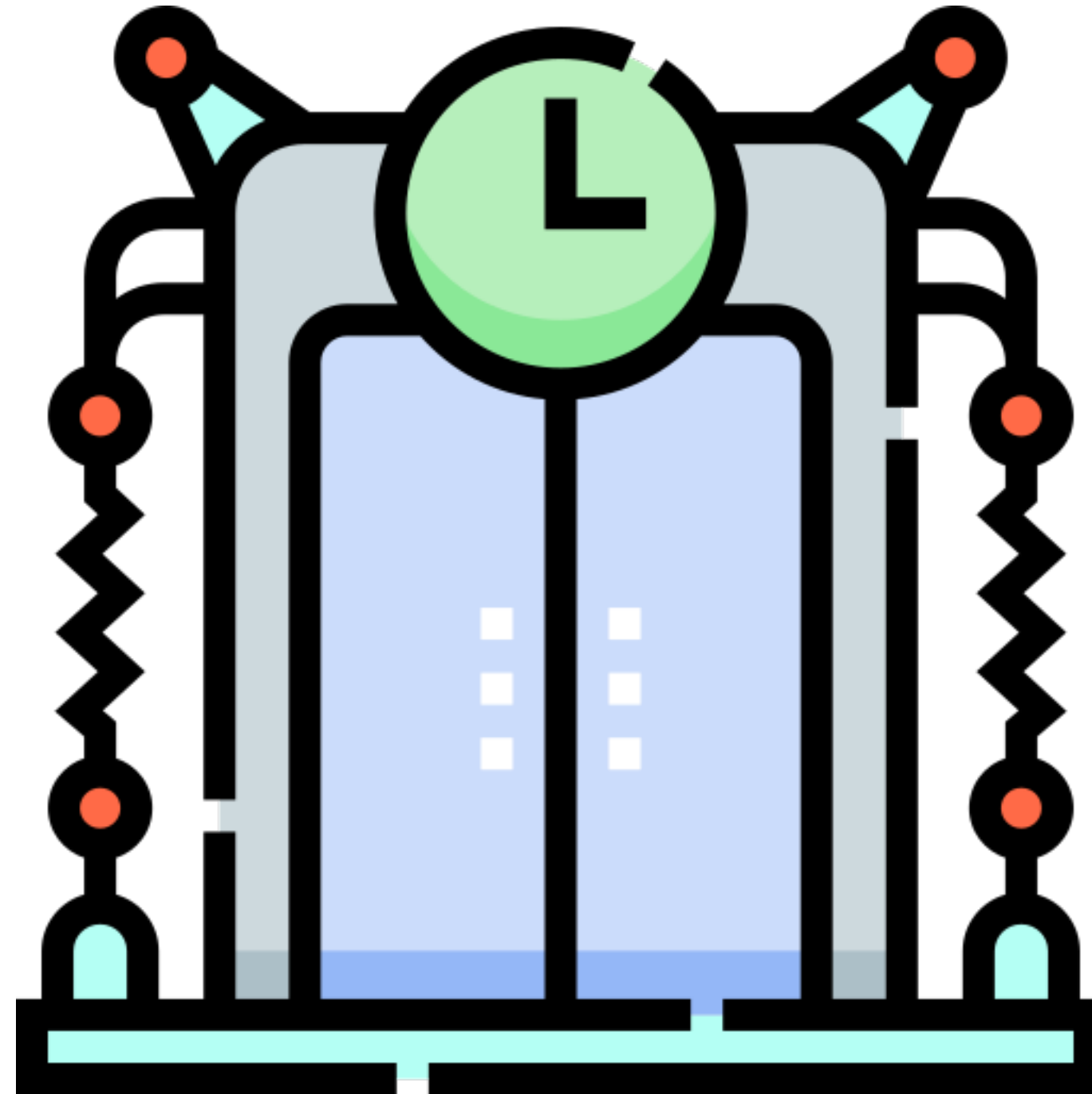
To provide a solid foundation to understanding the fundamentals of Git.

Git



Git is a free and open source distributed **version control system designed to handle everything from small to very large projects with speed and efficiency.**

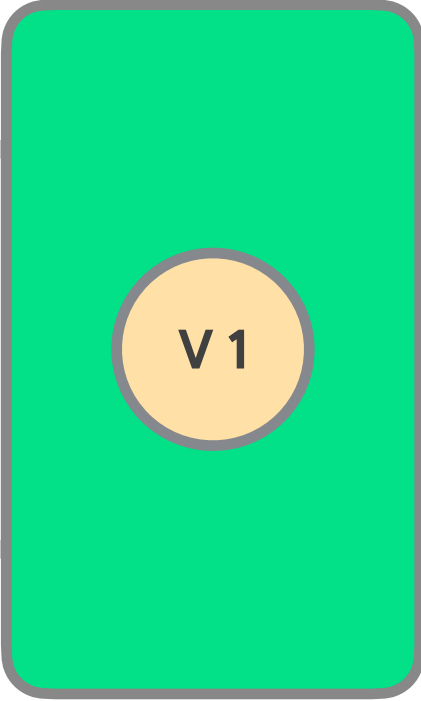
Git



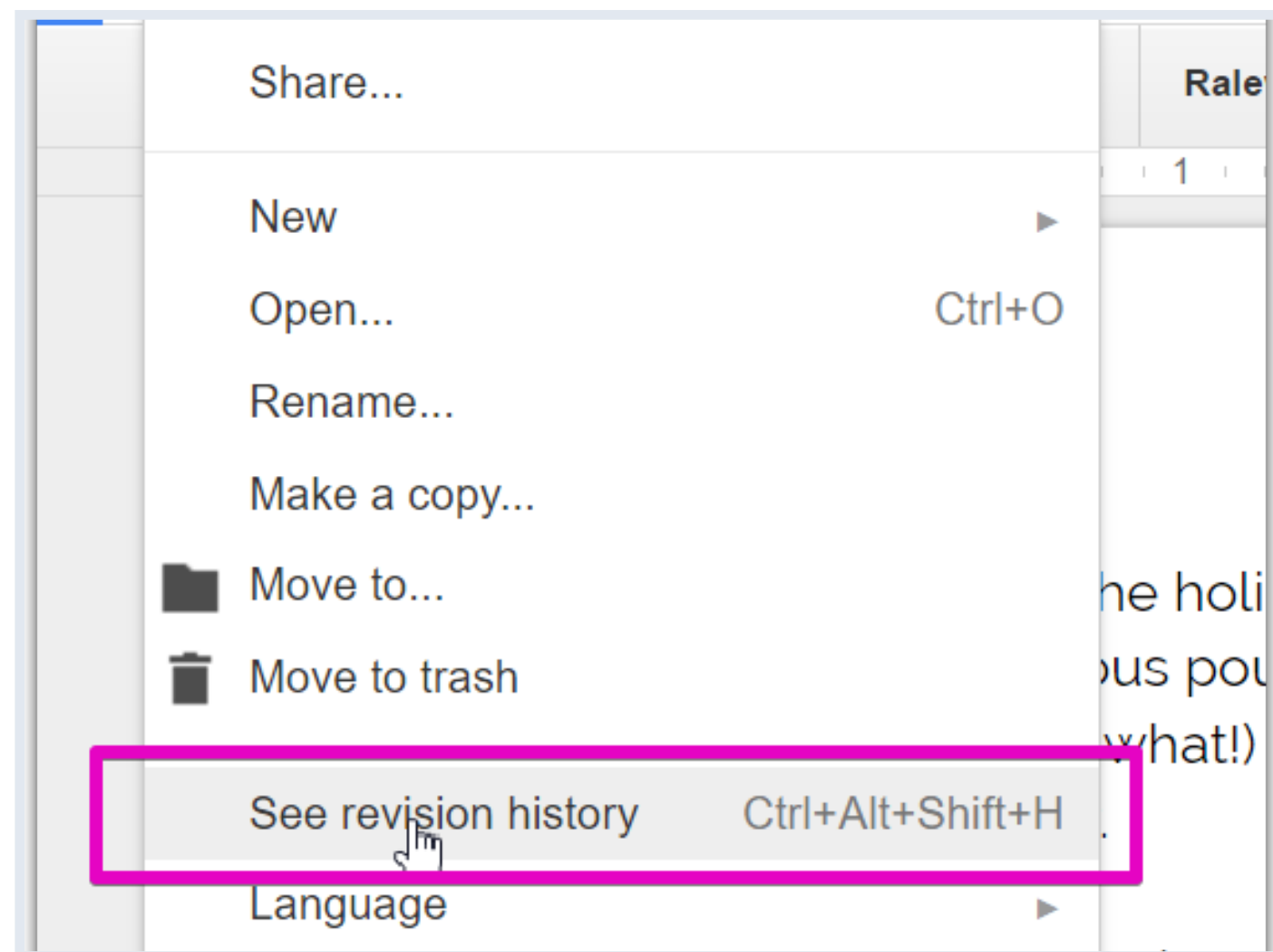
Git



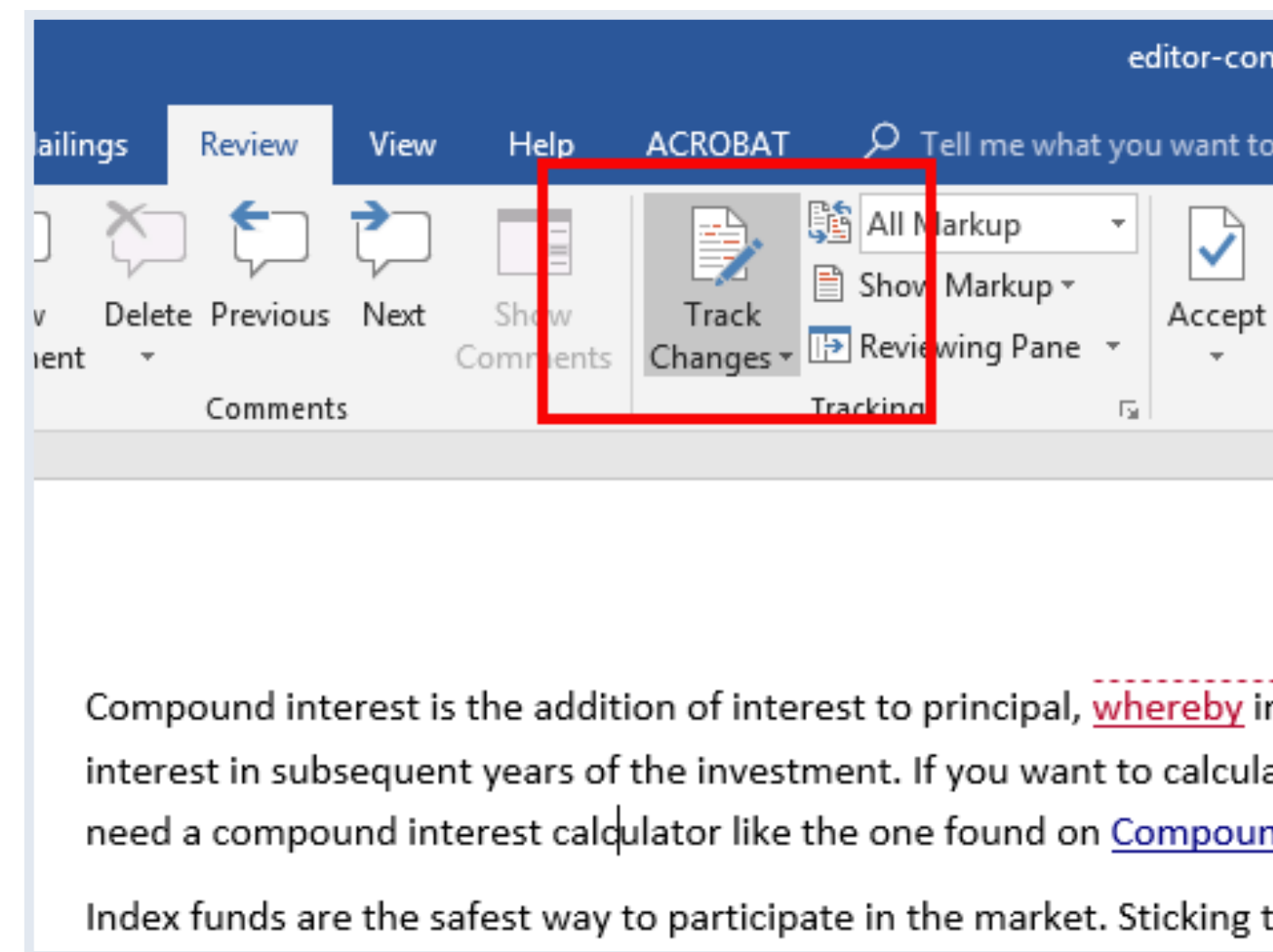
Git



Types of Version Control



Version Control

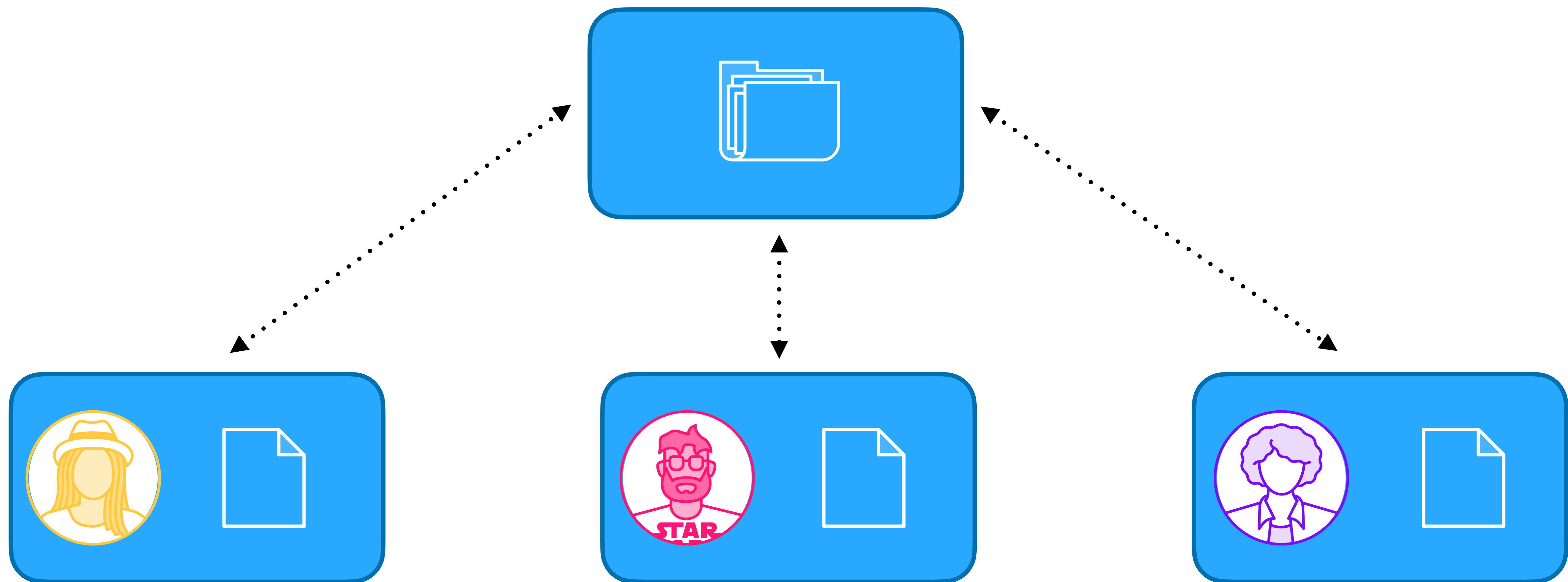


Version Control

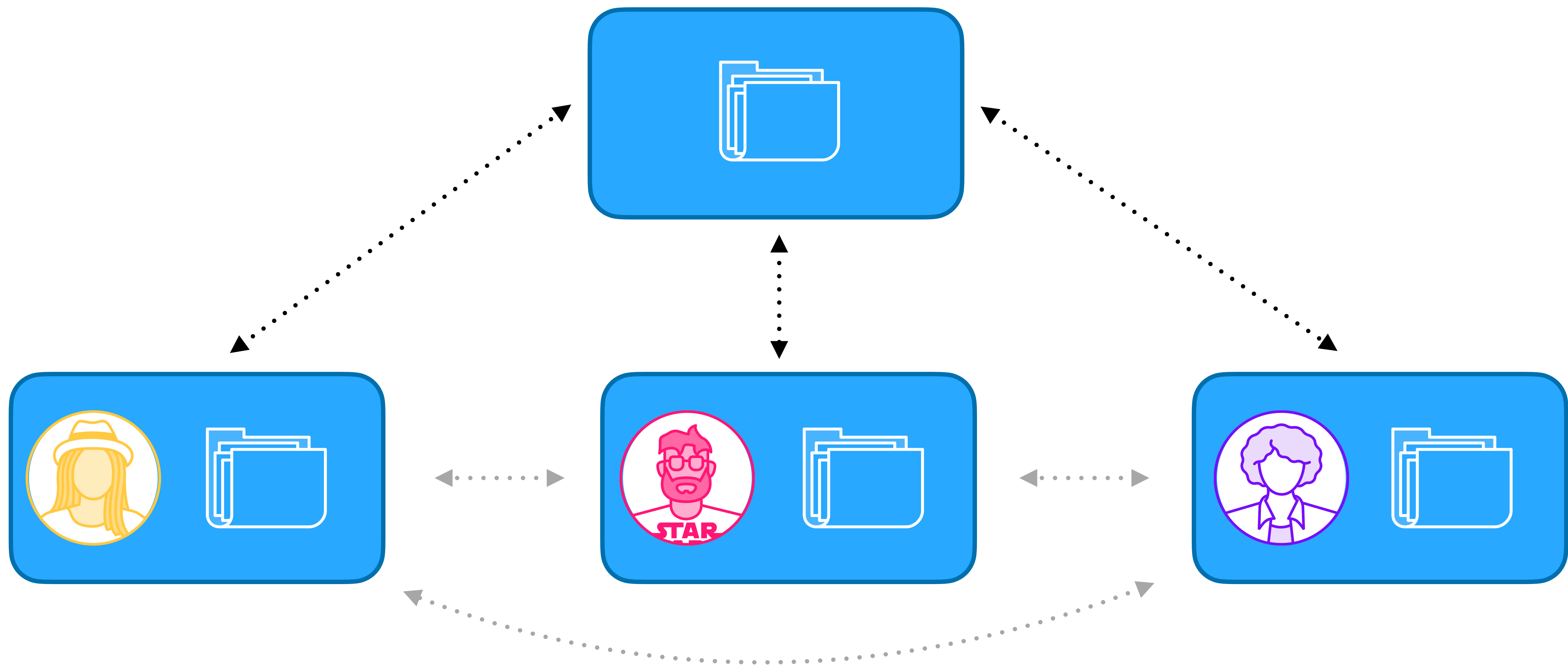


Version Control

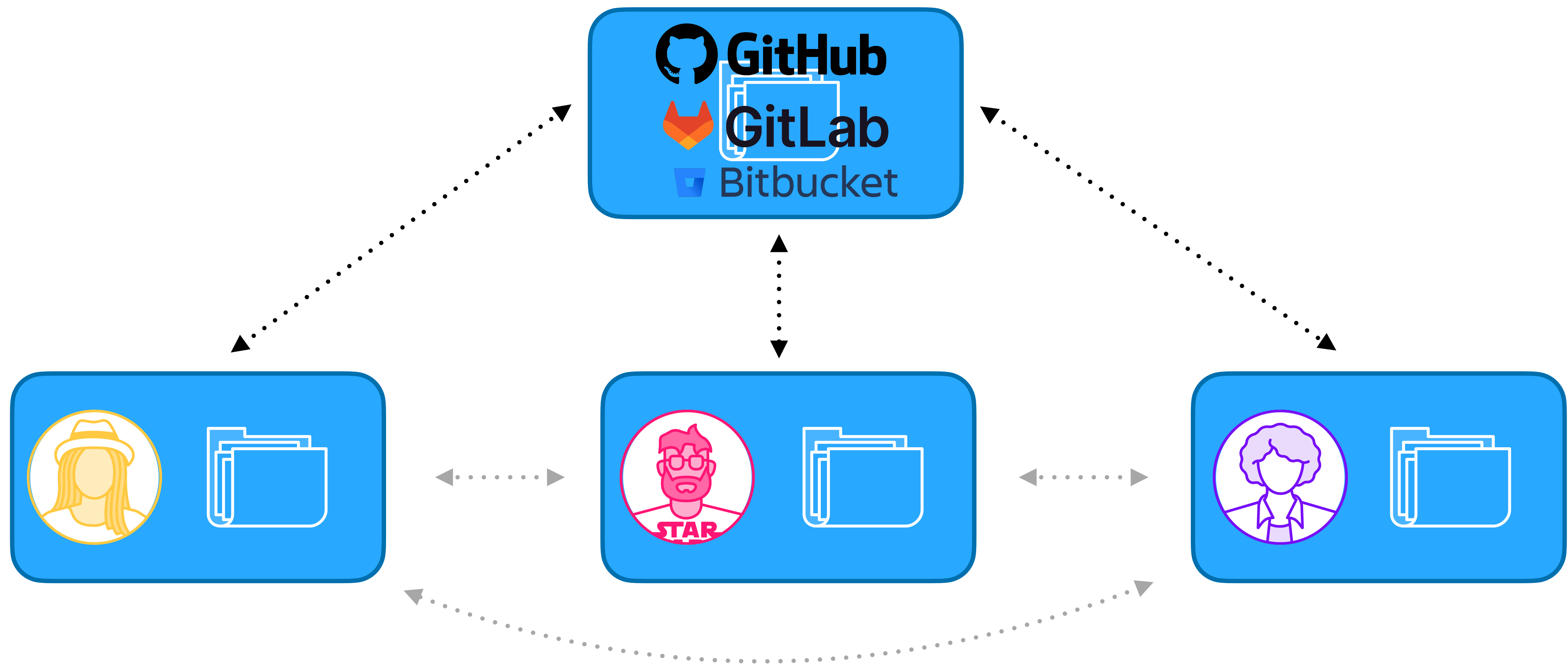
Centralized Version Control



Distributed Version Control

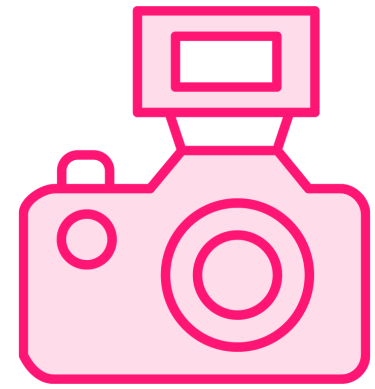


Distributed Version Control

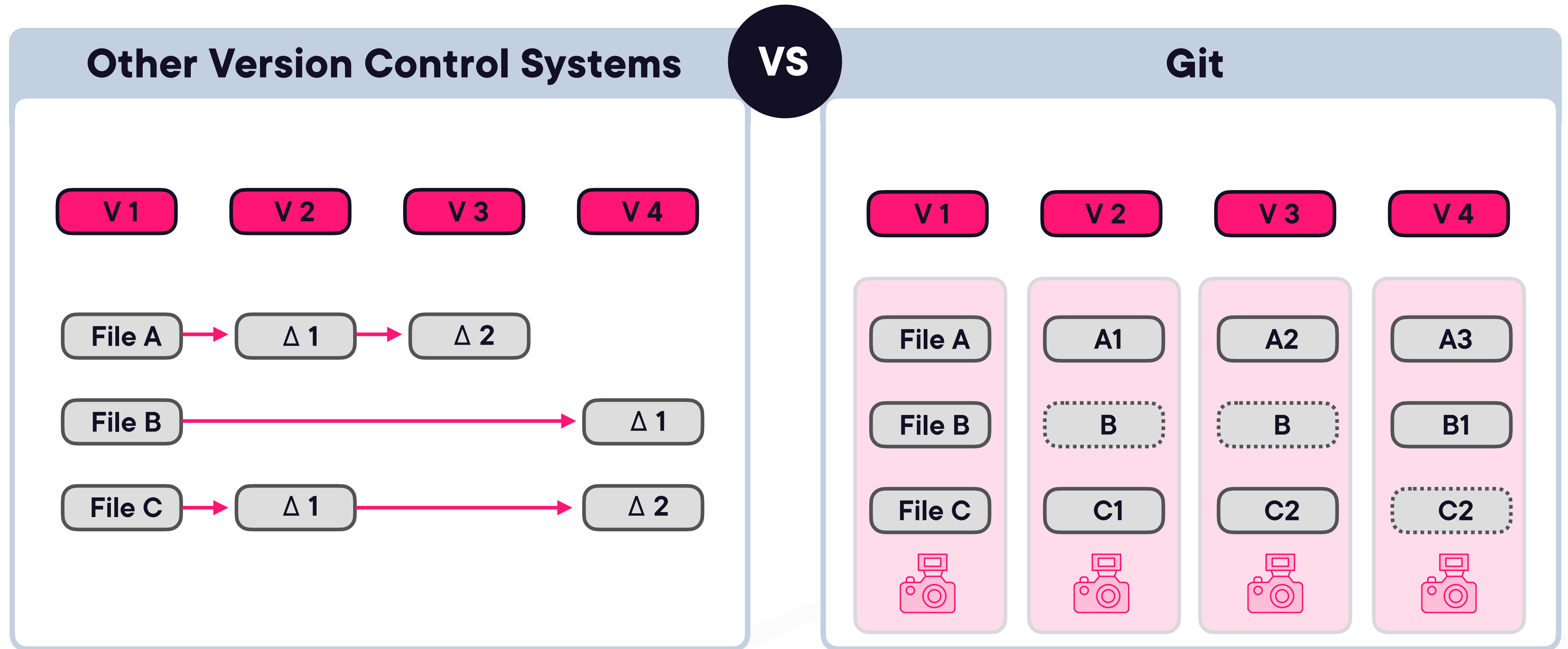




Characteristics of Git



Git Stores Versions as Snapshots Not Diffs



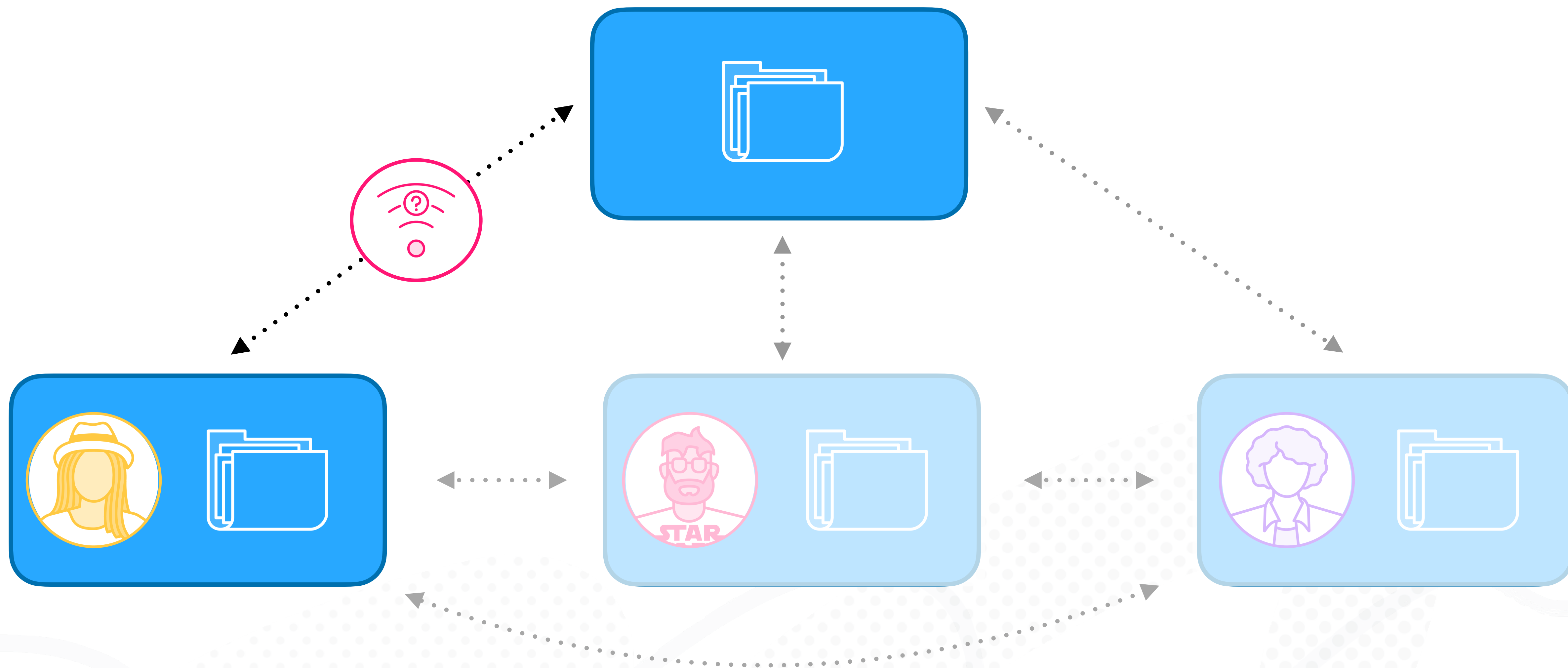


Git Snapshots

Every time you save or “commit” to your project, you are creating a snapshot of all the files, history, and metadata for your project at that specific point in time.

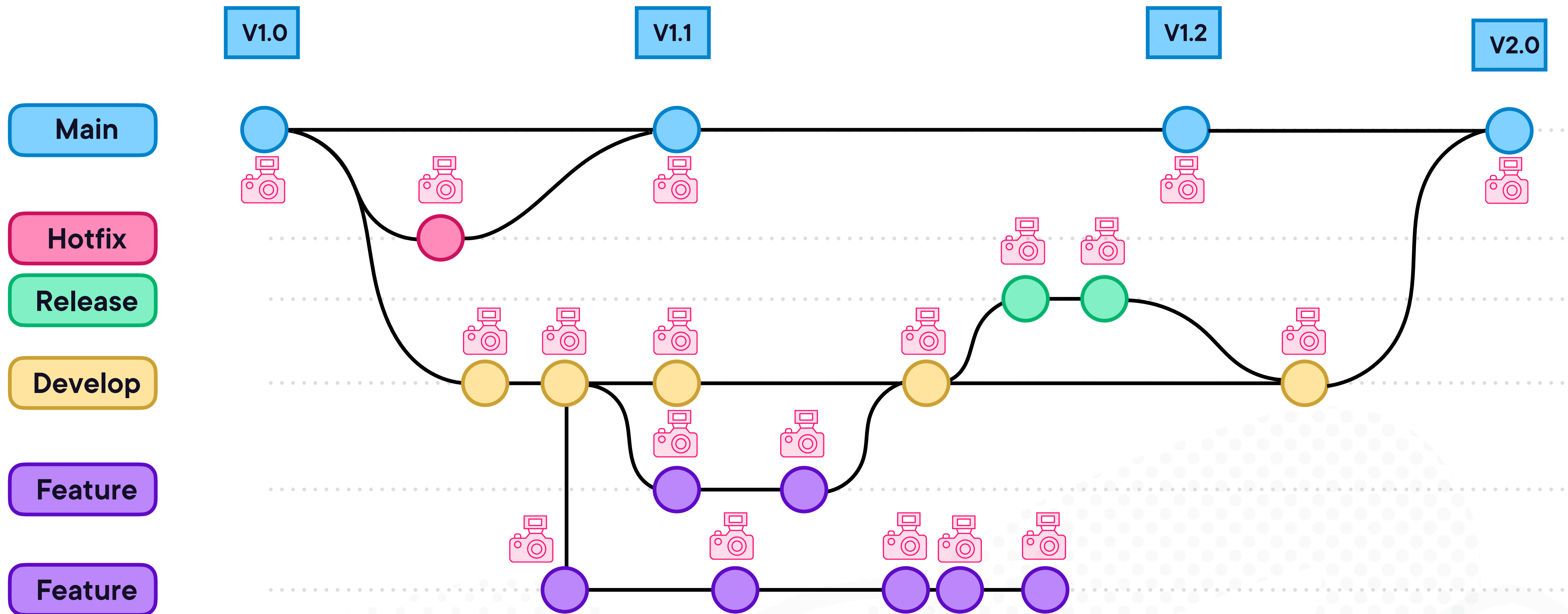


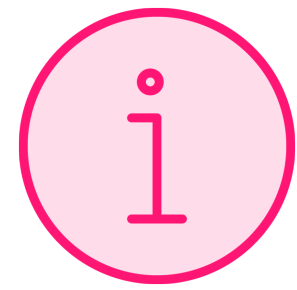
Git is Enhanced for Local Development





Git is Designed for Non-linear Development

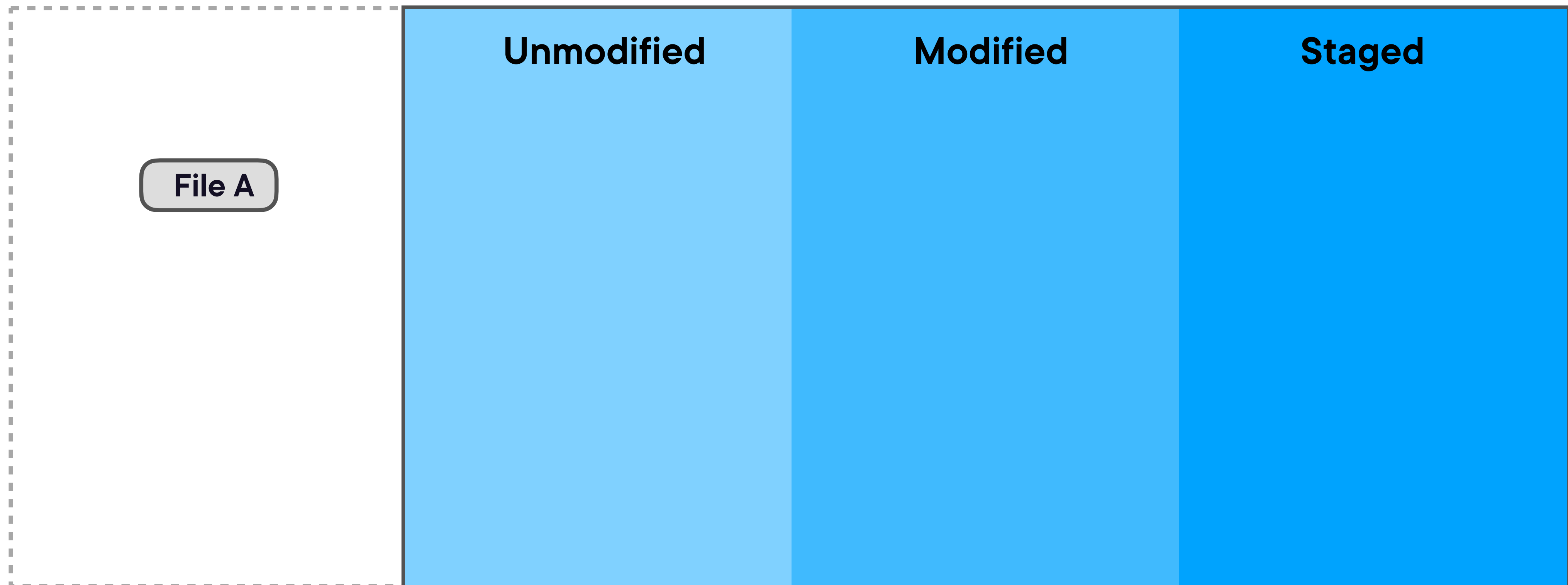


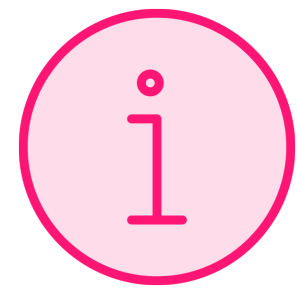


Different Statuses of Your Files

Untracked

Tracked

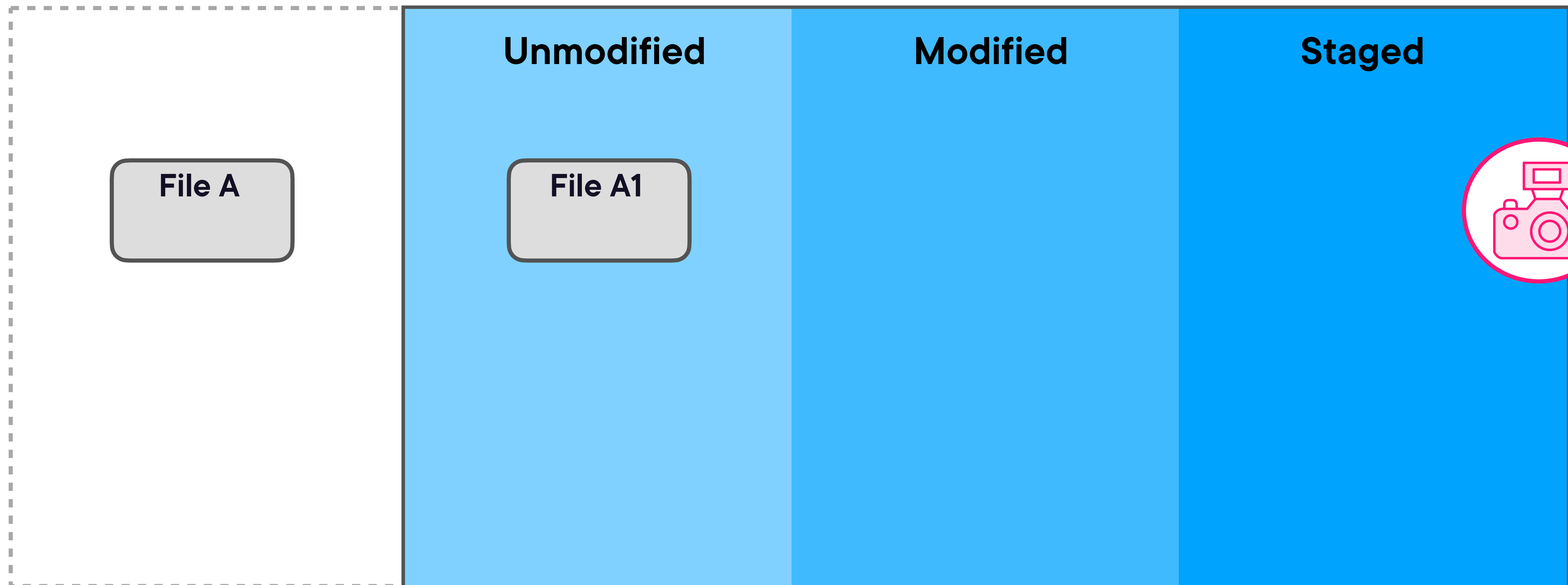


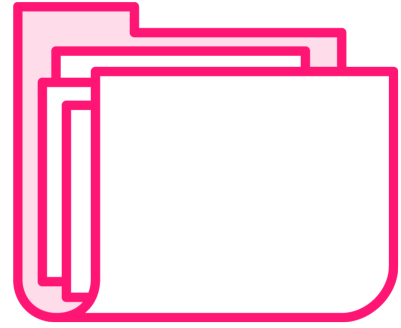


Different Statuses of Your Files

Untracked

Tracked

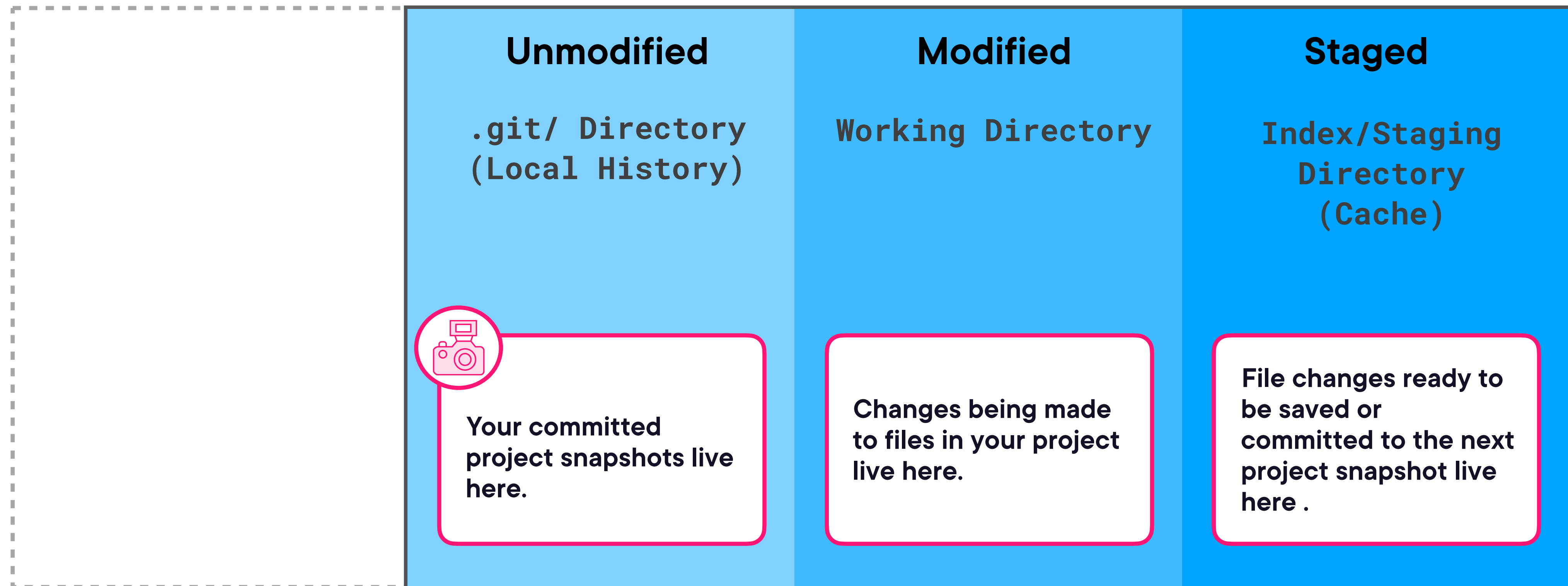


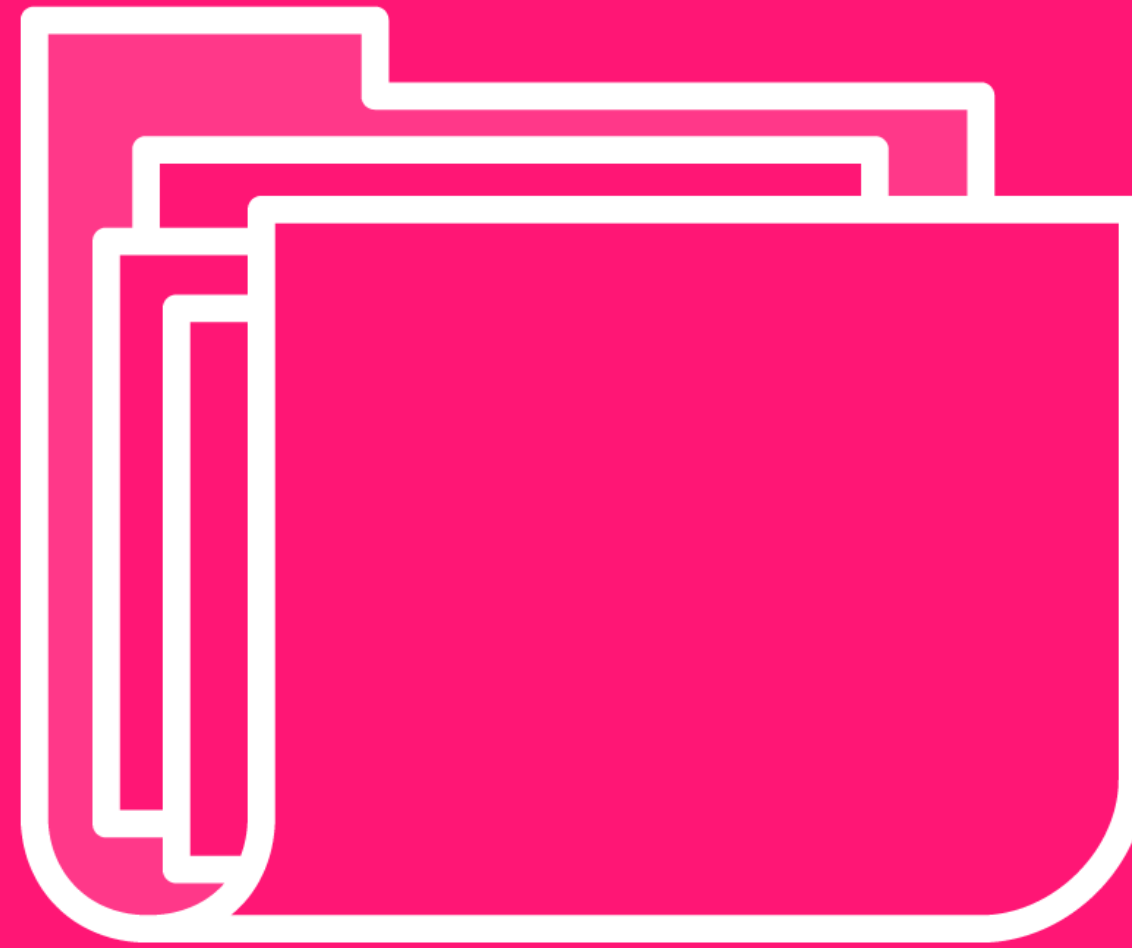


Different Directories of Git

Untracked

Tracked





Git Repository

**A repository contains all of your project's files
and each file's revision history**

Git Basics

<code>git init</code>	Create a new .git repository and begins tracking
<code>git add</code>	Move modified files into the staging area
<code>git status</code>	Shows you the status of your files
<code>git commit</code>	Create a snapshot and commit to .git
<code>git config</code>	Set and read specific Git configurations
<code>git log</code>	Shows the committed snapshot history
<code>git diff</code>	Shows changes between your working directory and staging area

Git Branches

<code>git branch</code>	List, create, or delete branches
<code>git checkout</code>	Switch between branches
<code>git merge</code>	Bring changes from one branch into another

Remote Repositories

<code>git clone</code>	Copies an entire repository into a new local .git directory
<code>git remote</code>	Create and show linked repositories
<code>git push</code>	Send updates to associate repositories
<code>git pull</code>	Retrieves and integrates changes from other repositories
<code>git fetch</code>	Retrieves but doesn't integrate changes from other repositories

Undoing Changes

<code>git revert</code>	Create a new commit that undoes a previous commit
<code>git reset</code>	Remove files from the staging area

git <command> --help

git add --help

GIT-ADD(1)
Manual

Git
GIT-ADD(1)

NAME

git-add - Add file contents to the index

SYNOPSIS

```
git add [--verbose | -v] [--dry-run | -n] [--force | -f] [--interactive | -i] [--patch | -p]
        [--edit | -e] [--[no-]all | --[no-]ignore-removal | [--update | -u]]
        [--intent-to-add | -N] [--refresh] [--ignore-errors] [--ignore-missing] [--
renormalize]
        [--chmod=(+|-)x] [--] [<pathspec>...]
```

DESCRIPTION

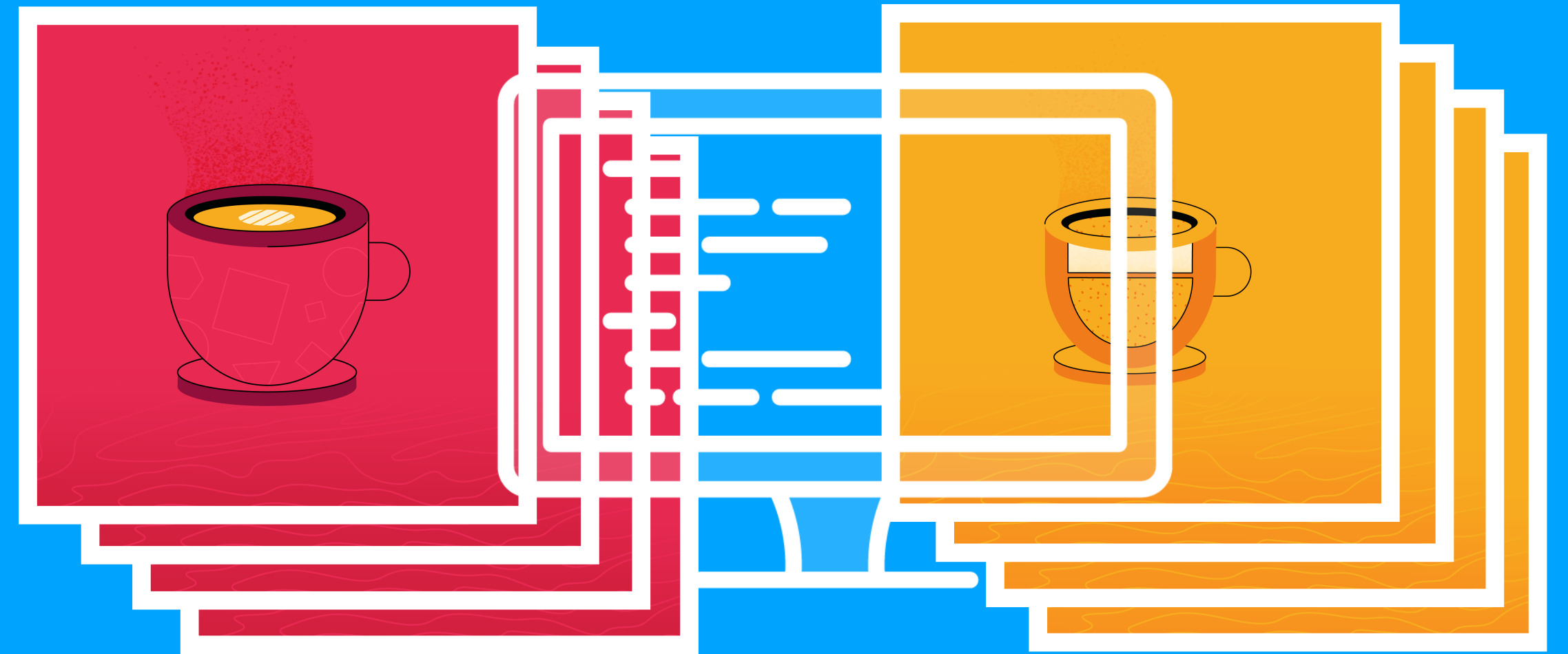
This command updates the index using the current content found in the working tree, to prepare the content staged for the next commit. It typically adds the current content of existing paths as a whole, but with some options it can also be used to add content with only part of the changes made to the working tree files applied, or remove paths that do not exist in the working tree anymore.

The "index" holds a snapshot of the content of the working tree, and it is this snapshot that is taken as the contents of the next commit. Thus after making any changes to the working tree, and before running the commit command, you must use the add command to add any new or modified files to the index.

This command can be performed multiple times before a commit. It only adds the content of the specified file(s) at the time the add command is run; if you want subsequent changes included in the next commit, then you must run git add again to add the new content to the index.

The git status command can be used to obtain a summary of which files have changes that are staged for the next commit.

:



Distributed Version Control

