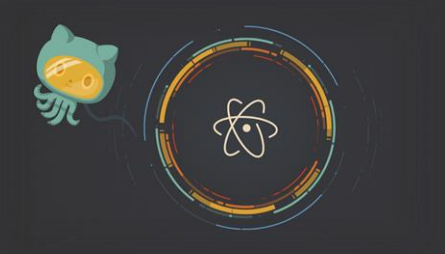


Introduction to Git

Version control for scientists using Atom IDE

What to expect

- Overview of Git
- GitHub and Atom
- Practical example using coffee data

Overview

- How do you manage different file versions?
- How do you work with collaborators on the same files?

Version control system

✓manage different versions of files

✓collaborate with yourself

✓collaborate with other people

> "Always remember your first collaborator is your future self, and your past self doesn't answer emails"

Git vs. GitHub vs. atom

Git

- Version control system
- Tracks changes
- Run locally

GitHub

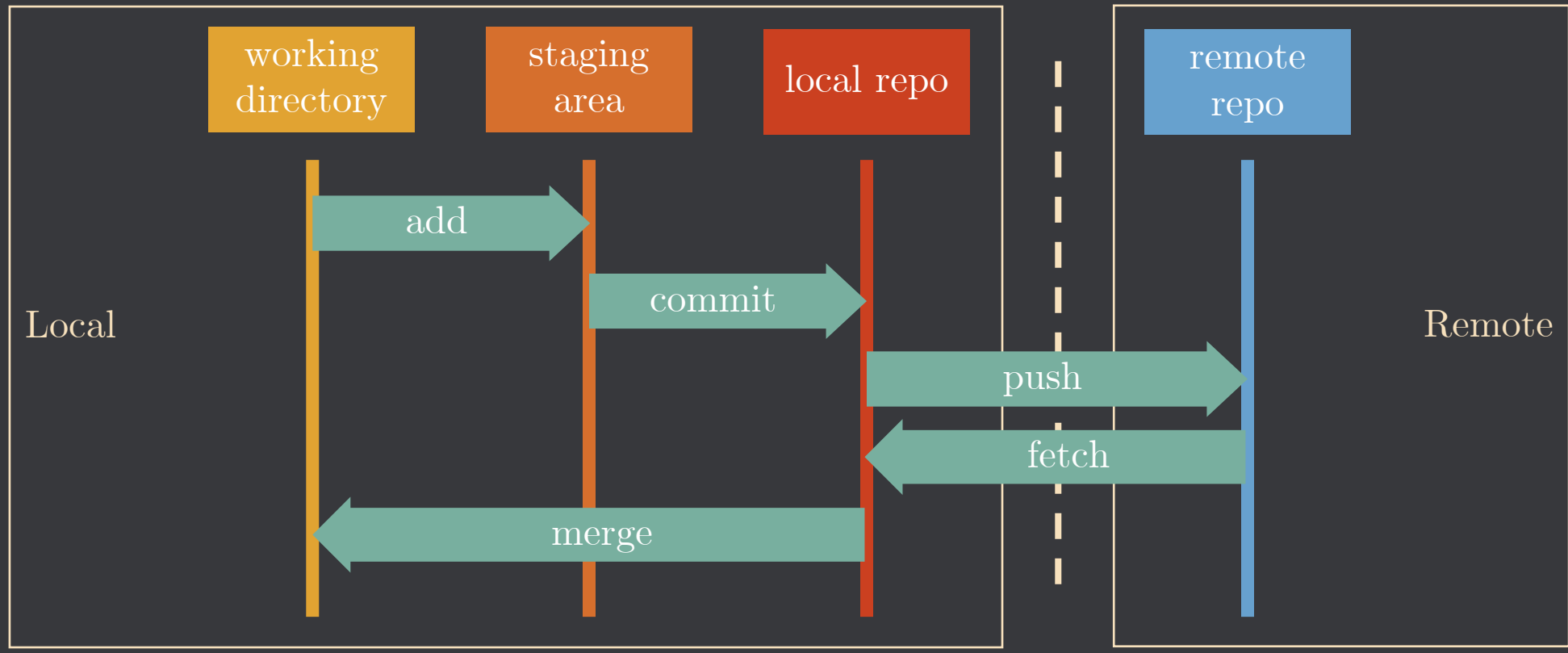
- Cloud based hosting service
- Manages Git repositories

ATOM

- Editor to connect to GitHub and manages repositories

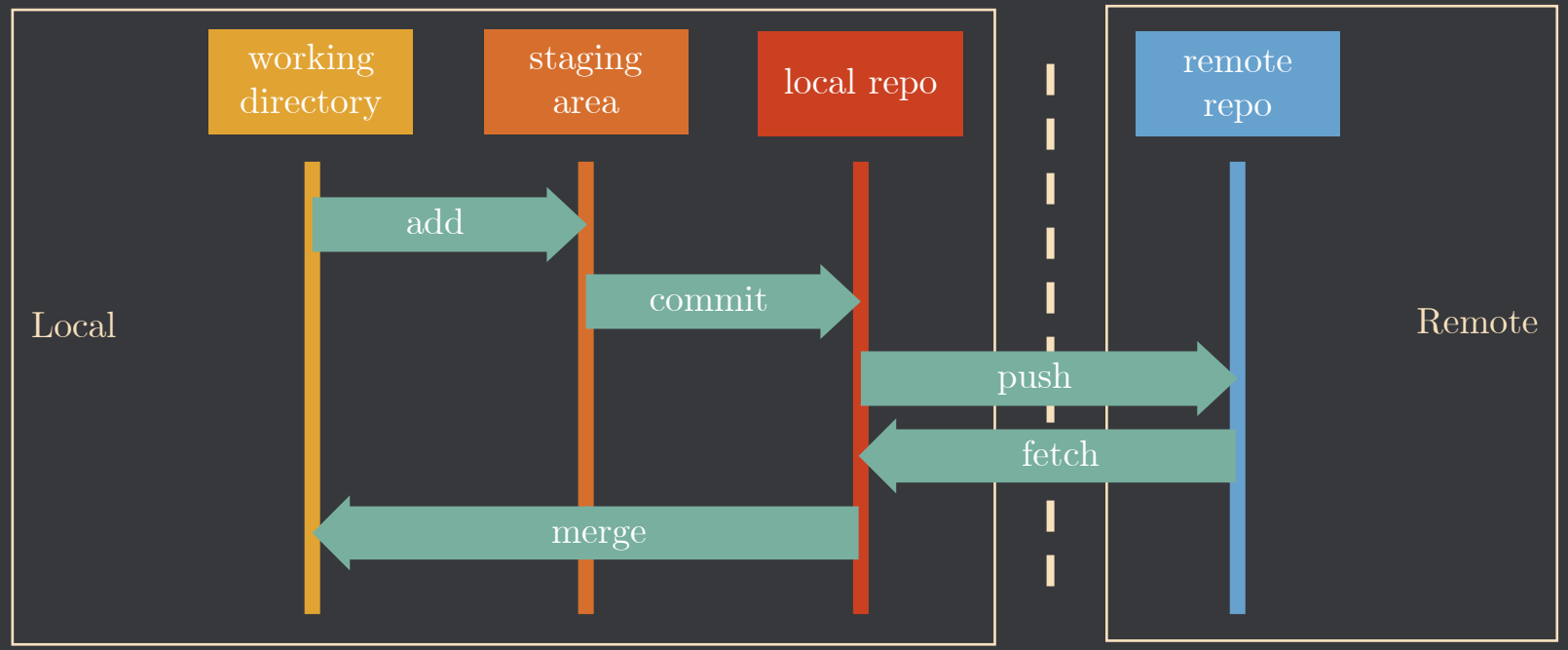
Why Git?

- Version control – manage different versions of your files
- Collaborate – working on files as a team ($n \geq 1$)



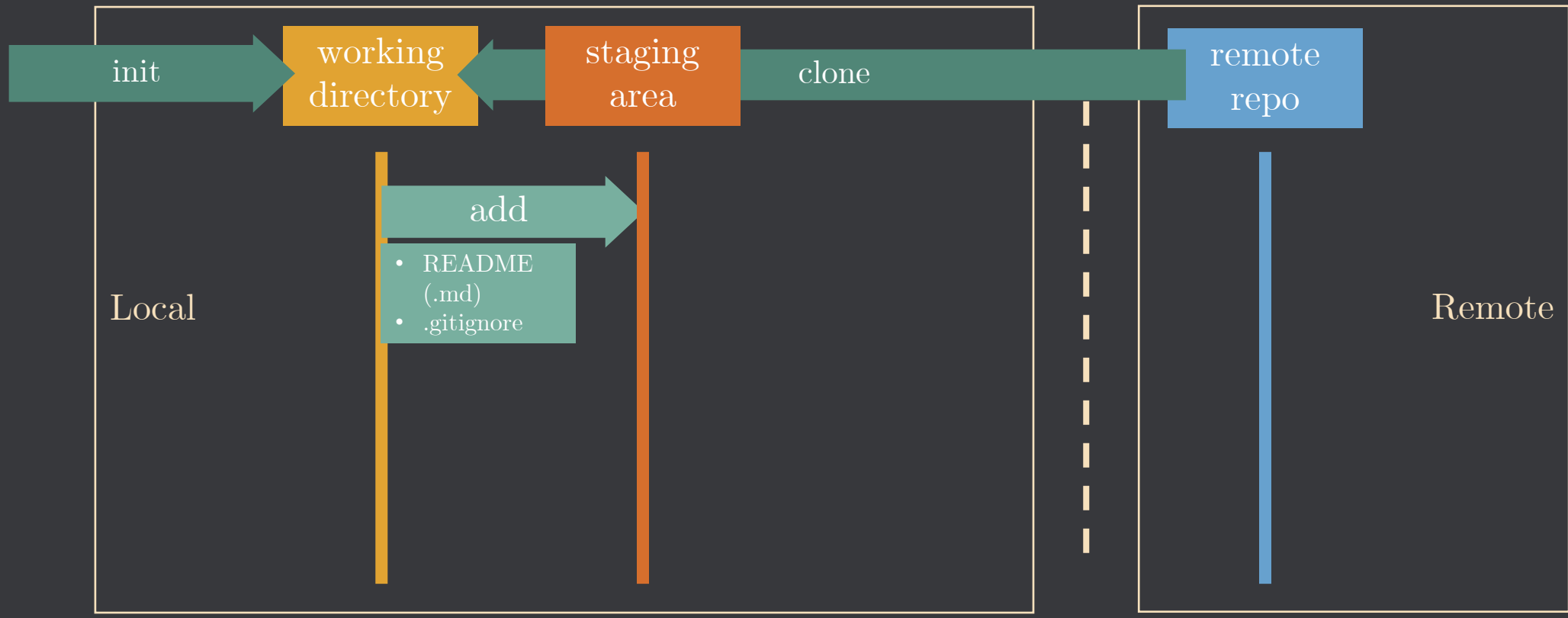
Tour of Git*Hub*

1. Init/ Clone
2. Commit
3. Push/ Pull
4. Branches/ merge



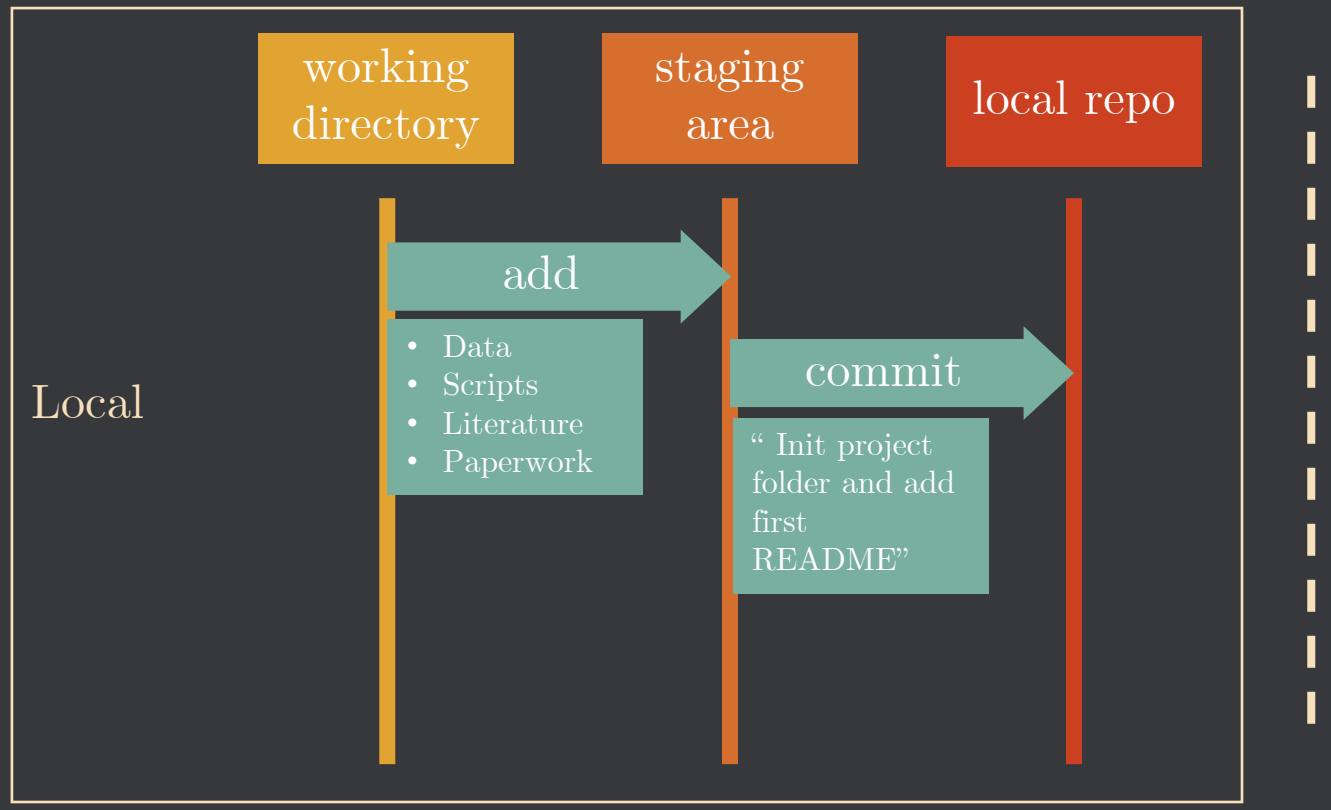
1. init/ clone

- Create your project by cloning an existing one or creating a new one



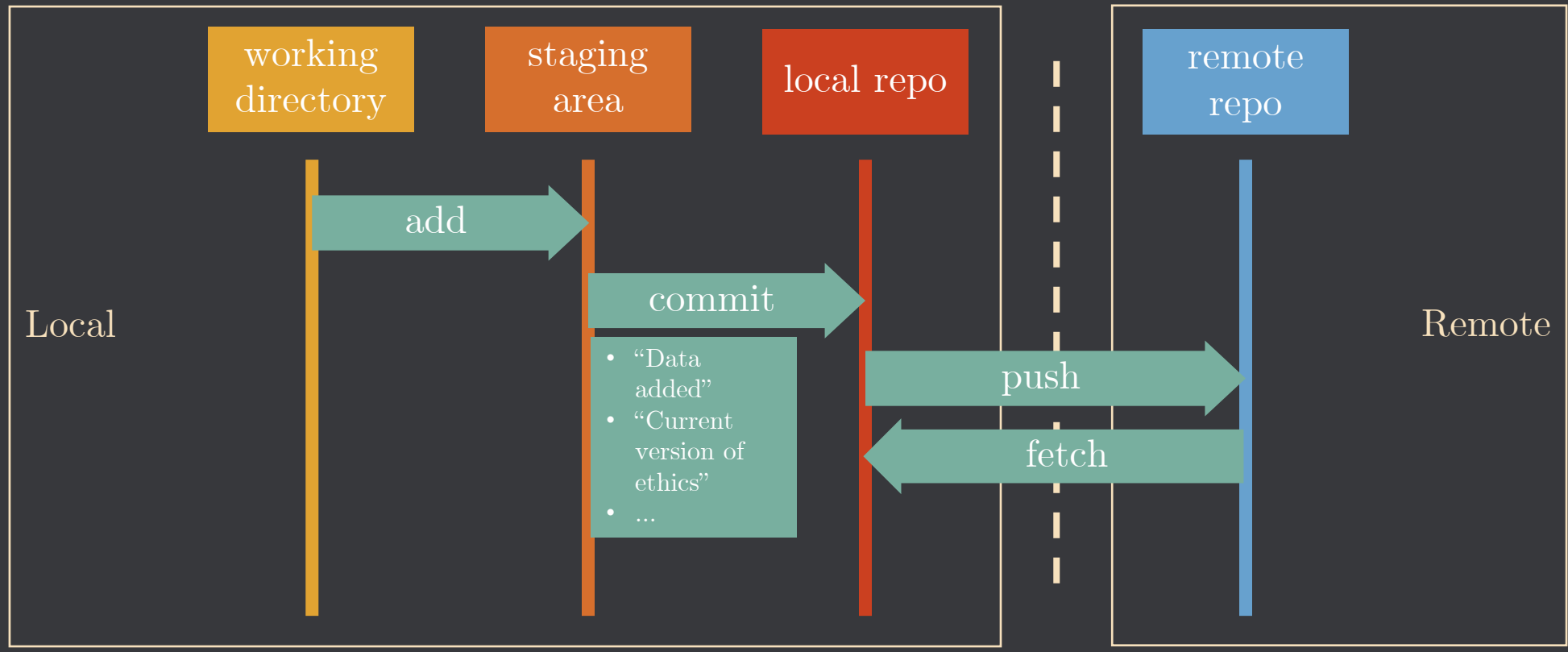
2. commit

- Be committed, save your interim results and provide a description



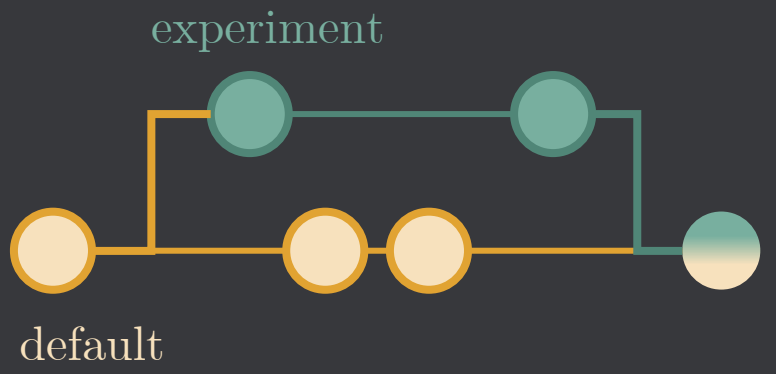
3. push/pull

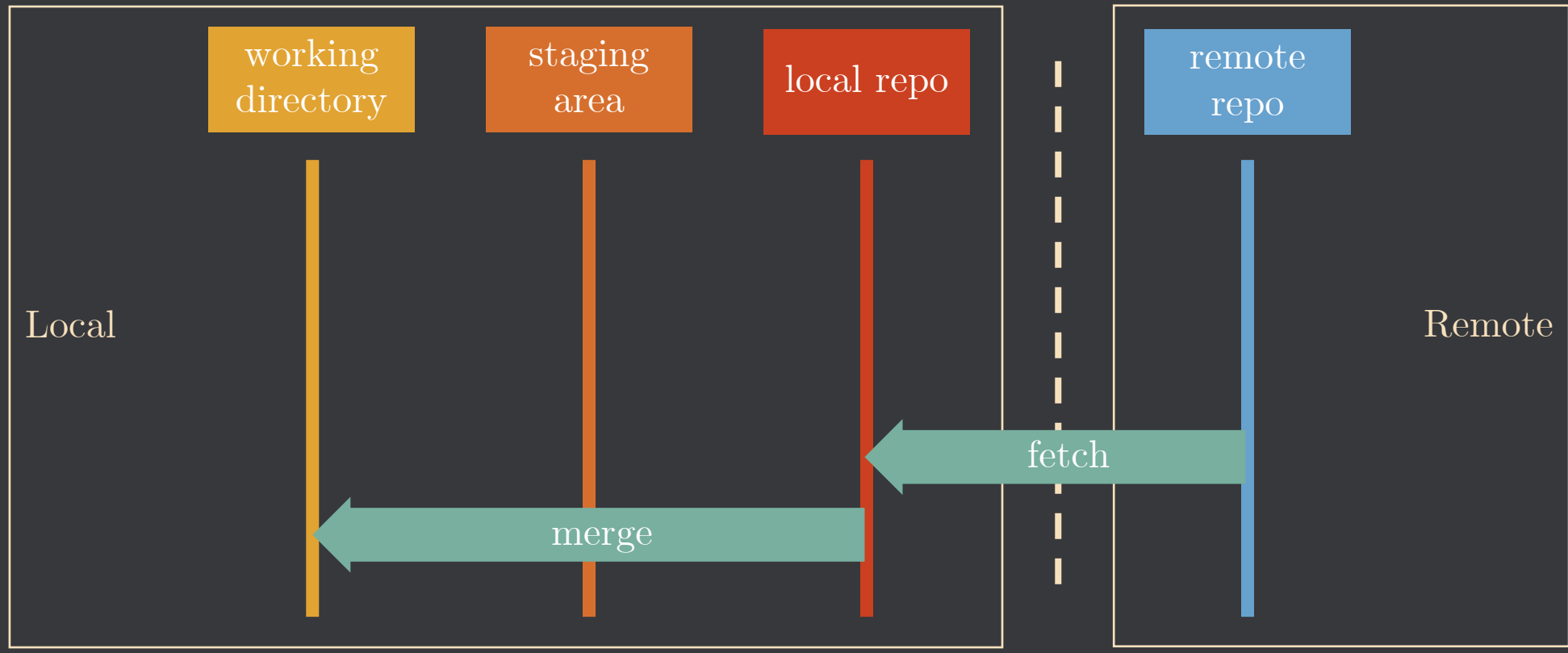
- Push changes/ commits to remote repo and pull to see if anything changed on remote repo



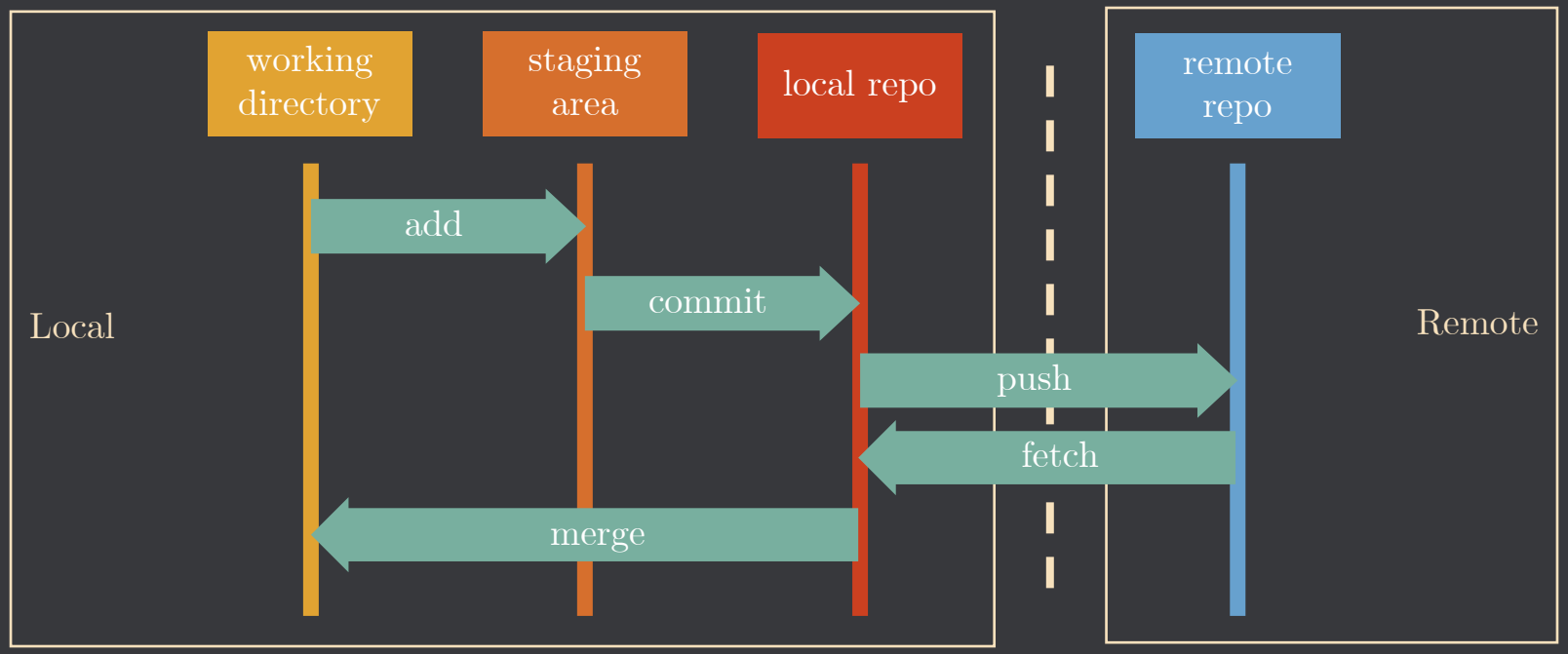
4. branches

- Multiple versions of the same project

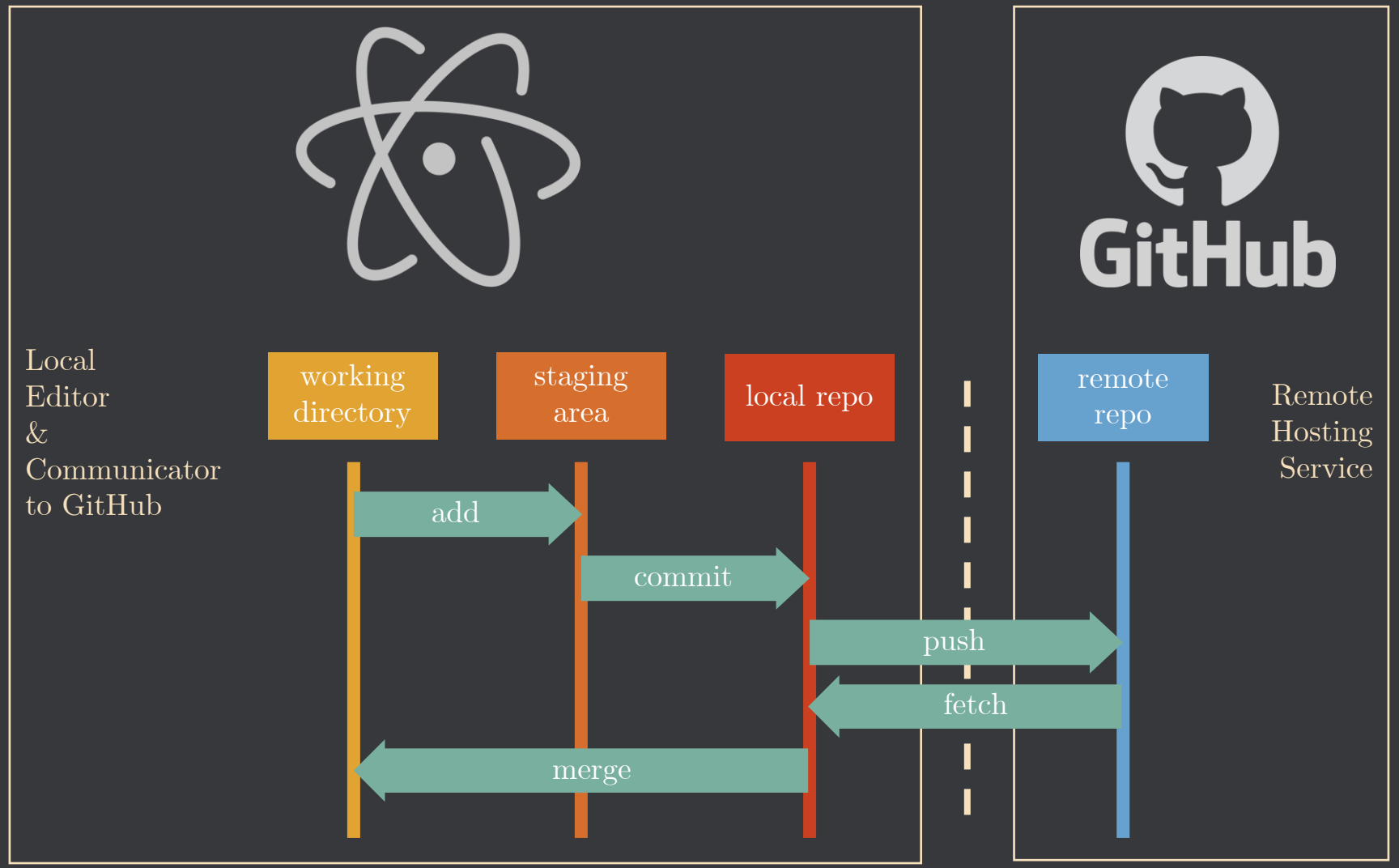




GitHub and Atom



GitHub and Atom



Practical example using coffee data

1. Init project
   - Add README
   - Add .gitignore
2. Add script and commit changes
3. Push to remote
4. Pull remote changes