Git and GitHub Workshop

Version Control for Economists

Kazuharu Yanagimoto

September 24, 2021

Why Do People Use Git?

📝 Recording

take logs of all coding activity of you and your collaborators

🕒 Restorable

go back to a previous version of codes

🔍 Comparable

focus on the change in the codes, and detect bugs

🔀 Branch

separate things complete and things under development

Is Git Easy?

No. I am sorry.

- Git has many commands with many options
- Need some knowledge to recover from a trouble
- Git allows various styles to use, which are different across people and organization

I propose

- First follow my workflow, which requires the minimum knowledge
- Once you’re comfortable with it, learn the detail

Is GitHub Git?

Git Is a Version Control Tool

- App
- Command Line
- Works Locally

GitHub Is a Web-Service

- Publish the code
- Collaboration

CUI vs GUI?

GUI Applications for Git & GitHub

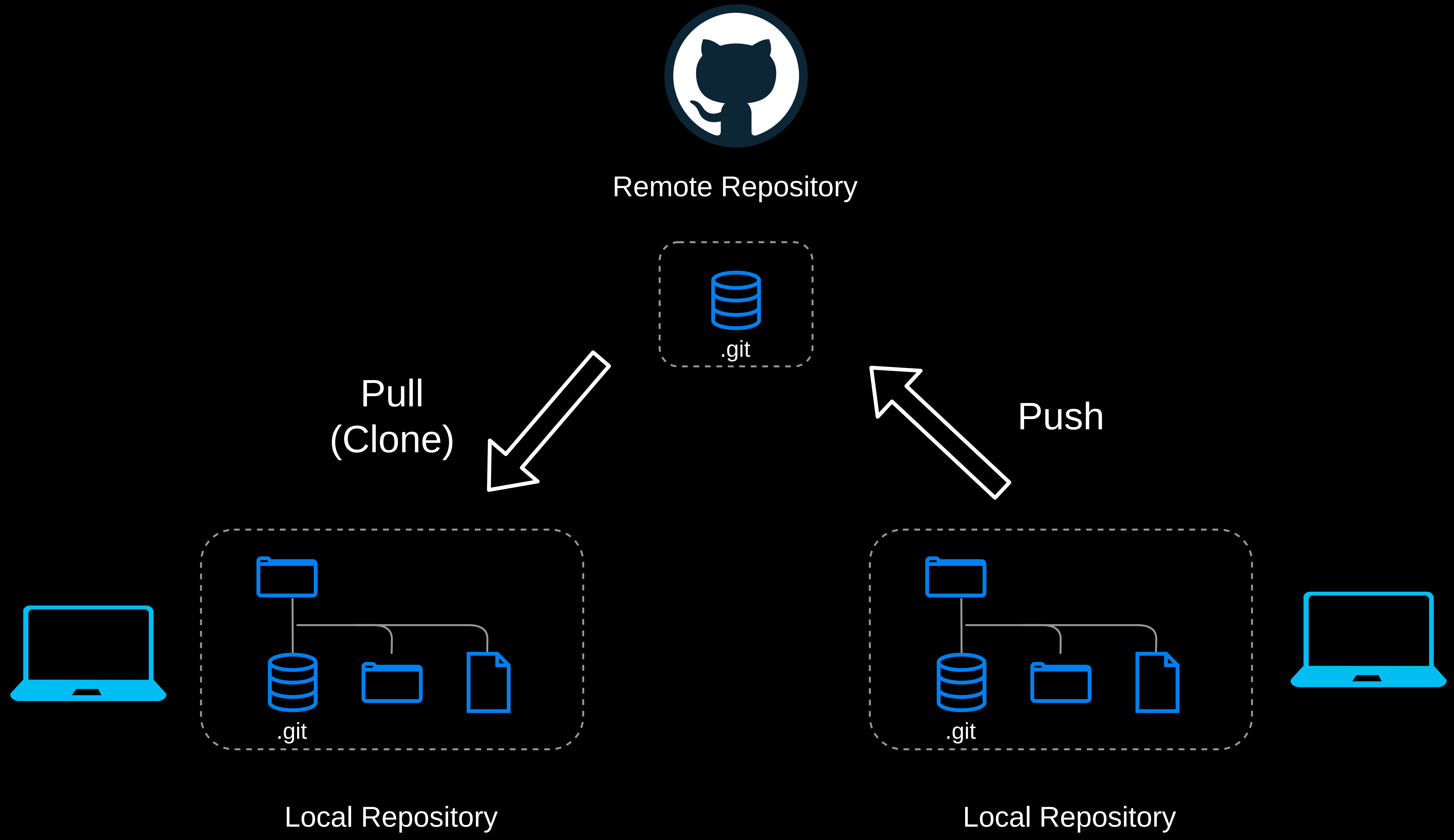
- GitHub Desktop
- VSCode
- Fork

I propose

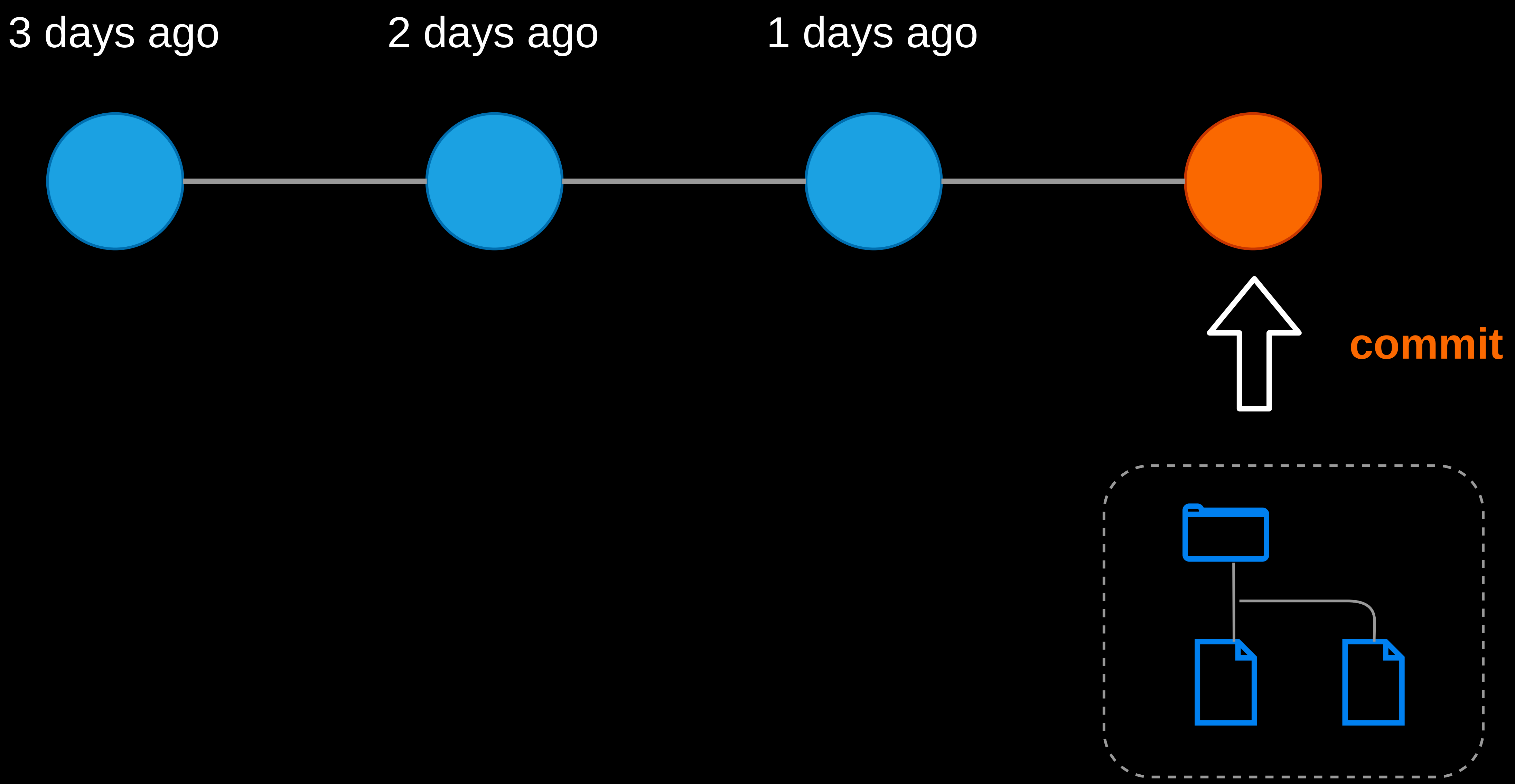
- Hybrid way in VSCode
- CUI knowledge is necessary for GUI

Basics of Git

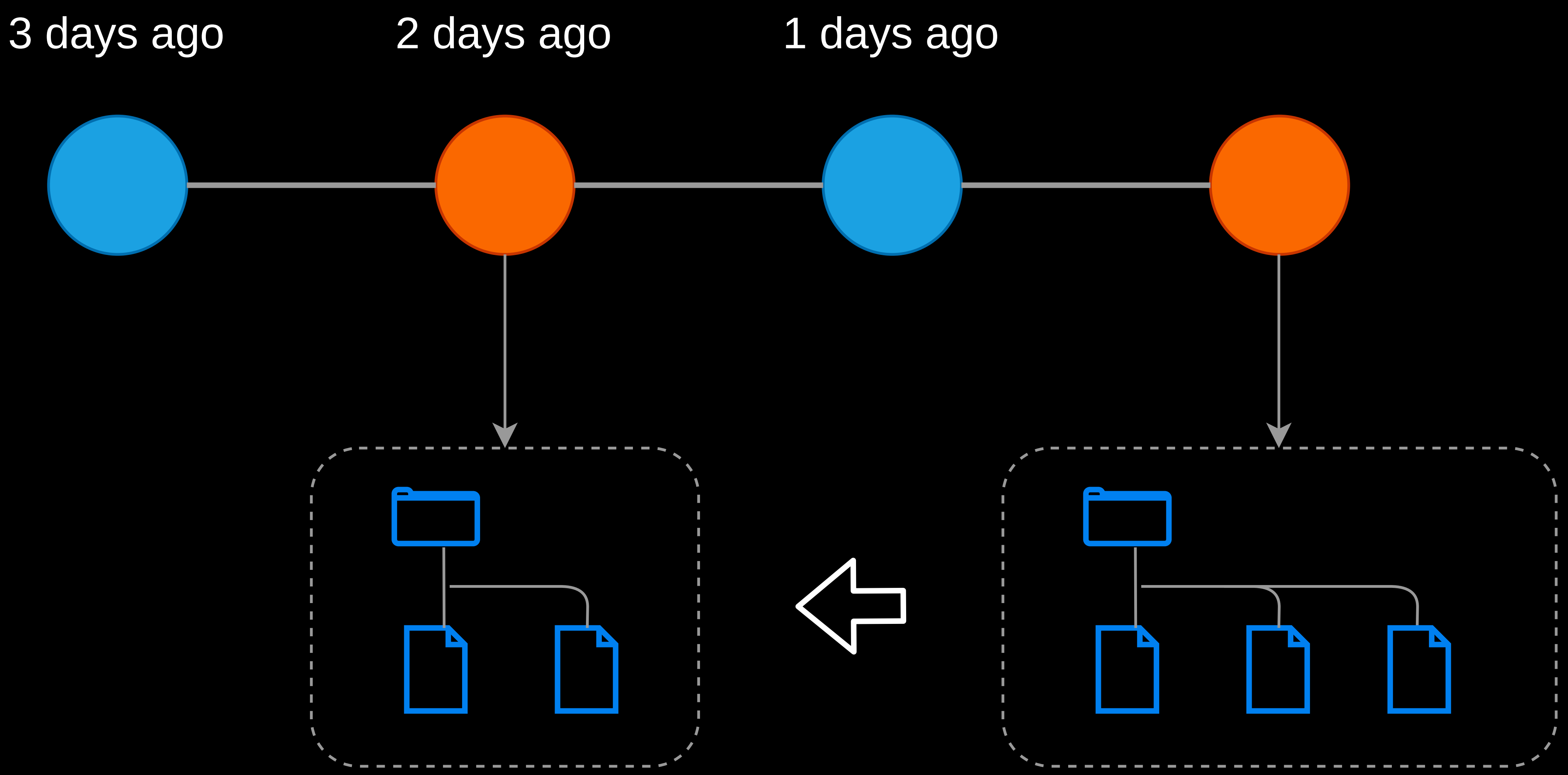
Local & Remote Repository



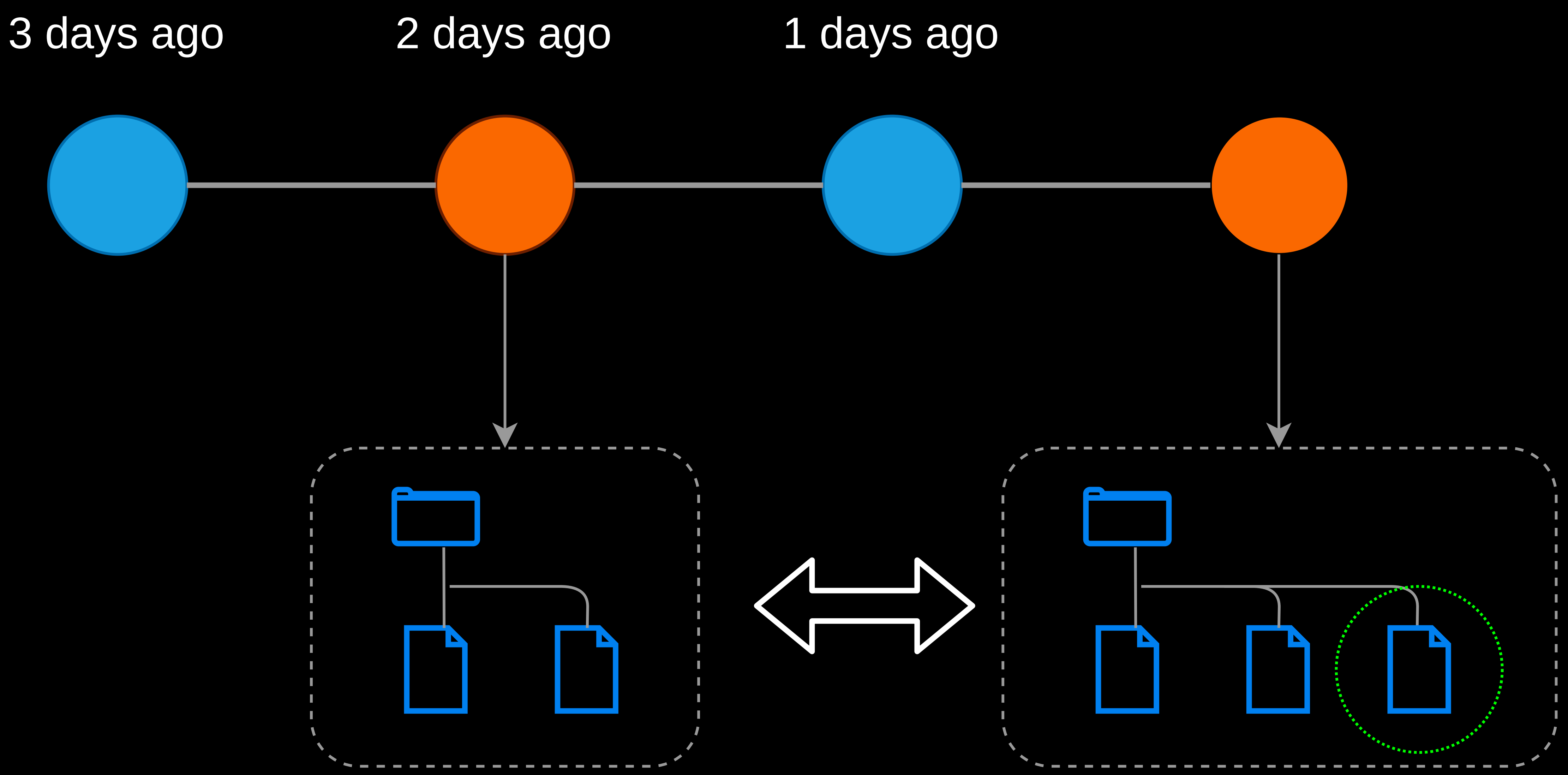
Commit Is a Save Point!



You Can Go Back to Any Commit

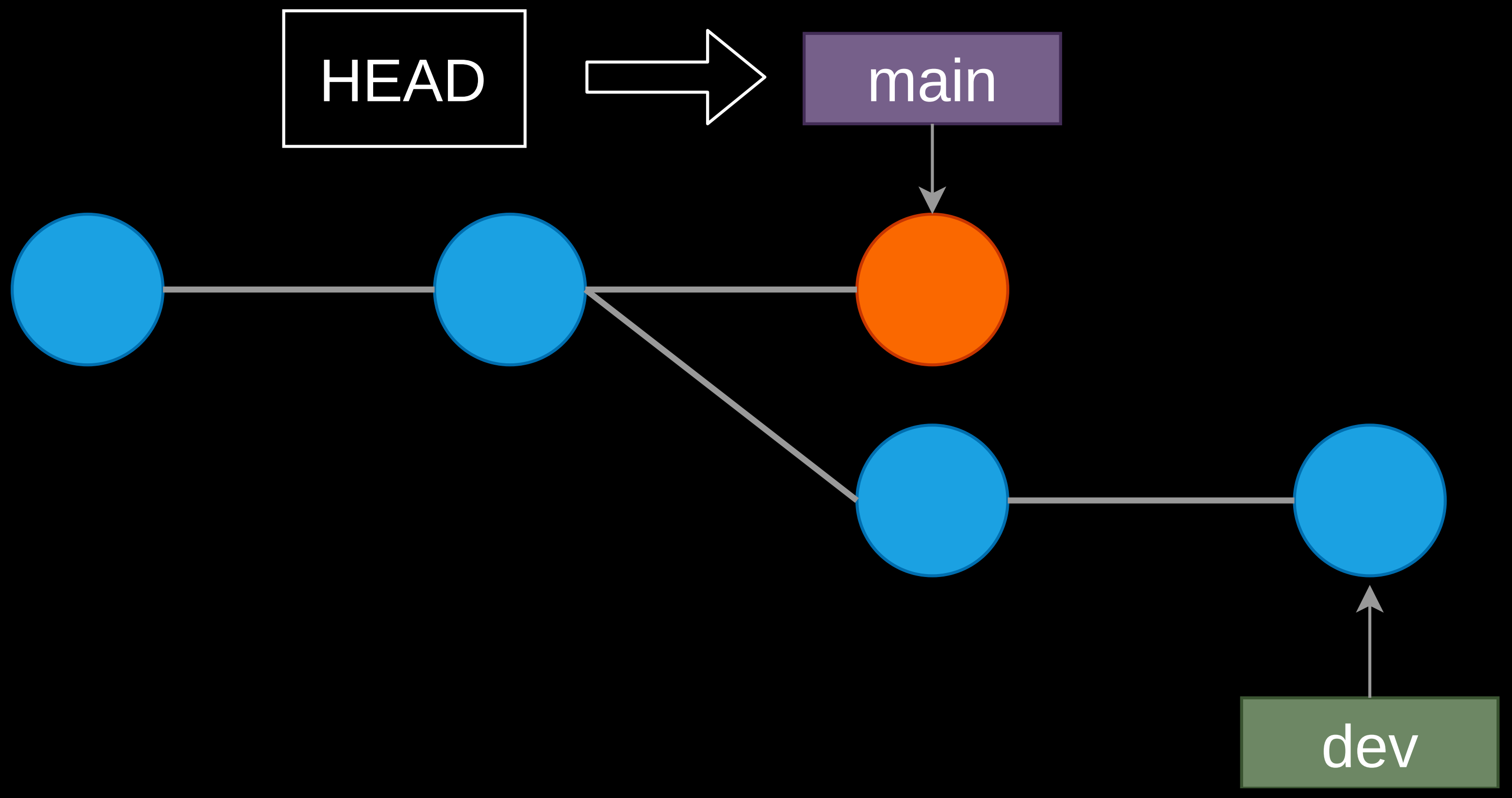


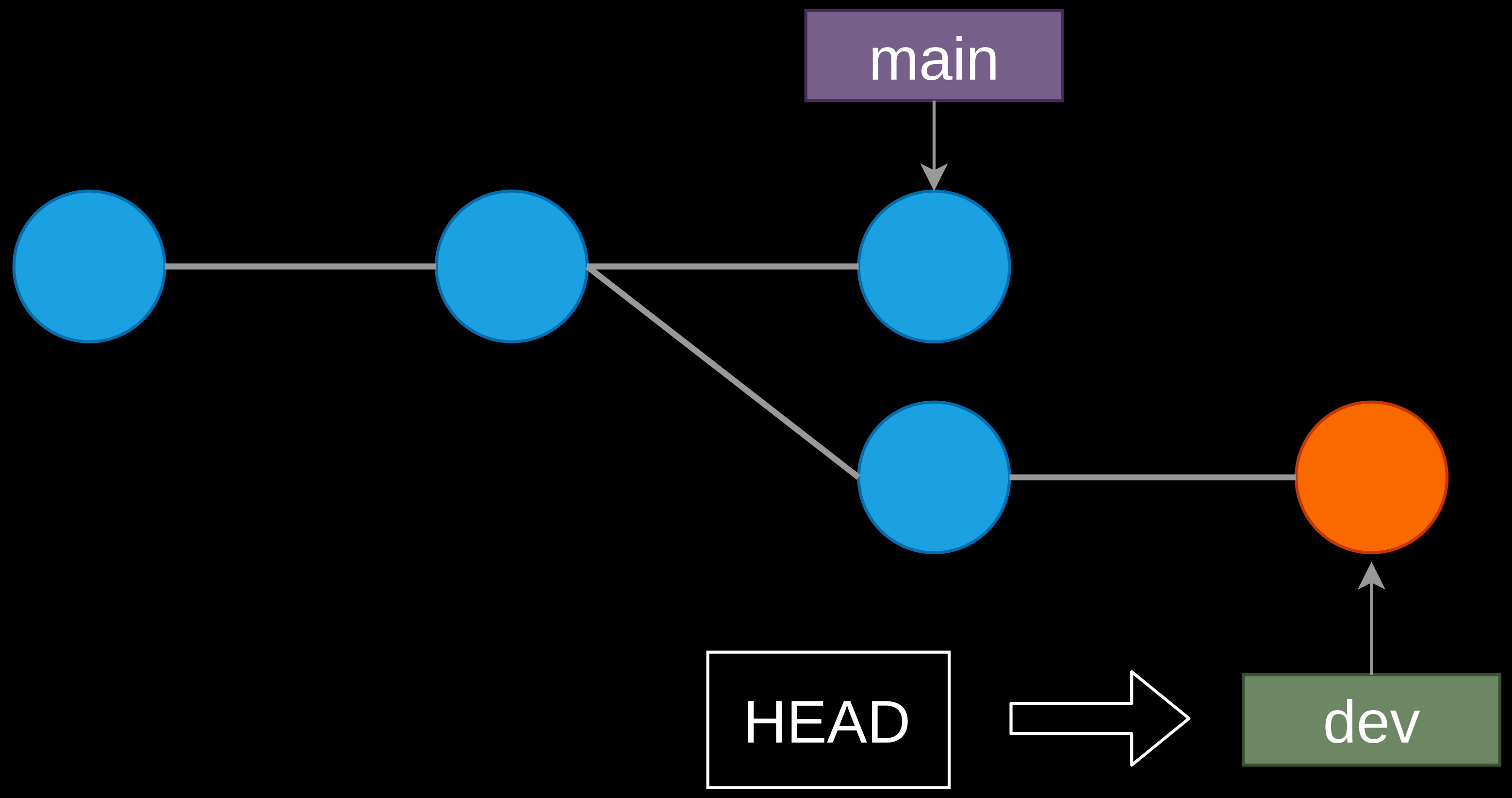
You Can Compare Any Two Commits



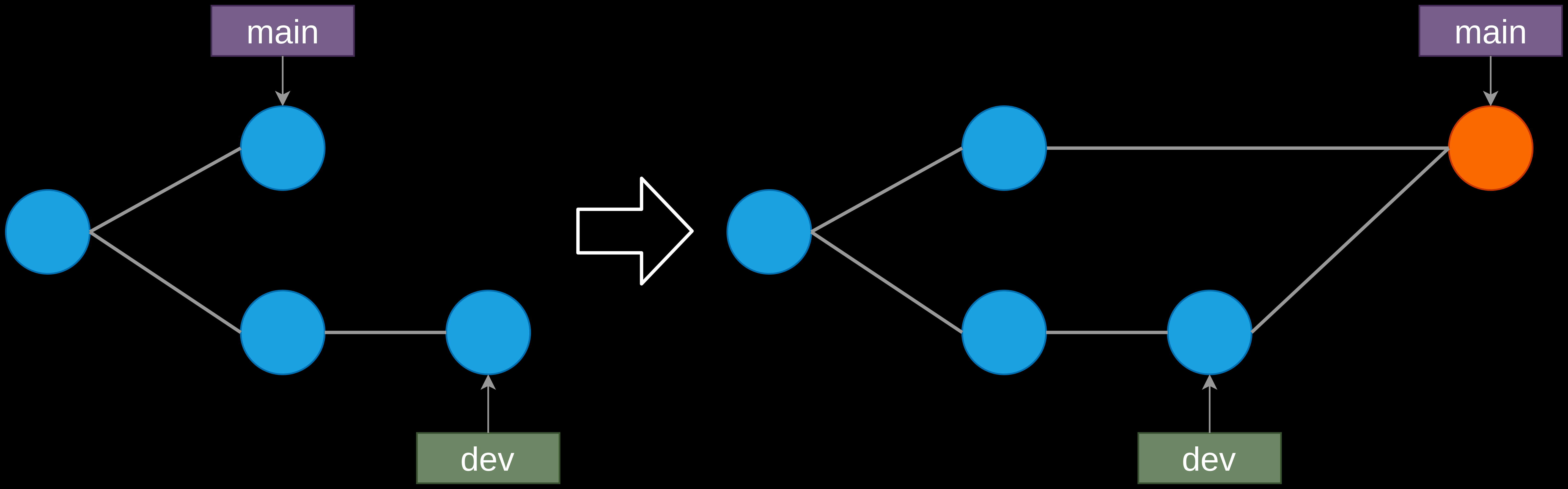
HEAD and Branch

- Branch: a label of commit
- HEAD: the branch you are seeing





Merge

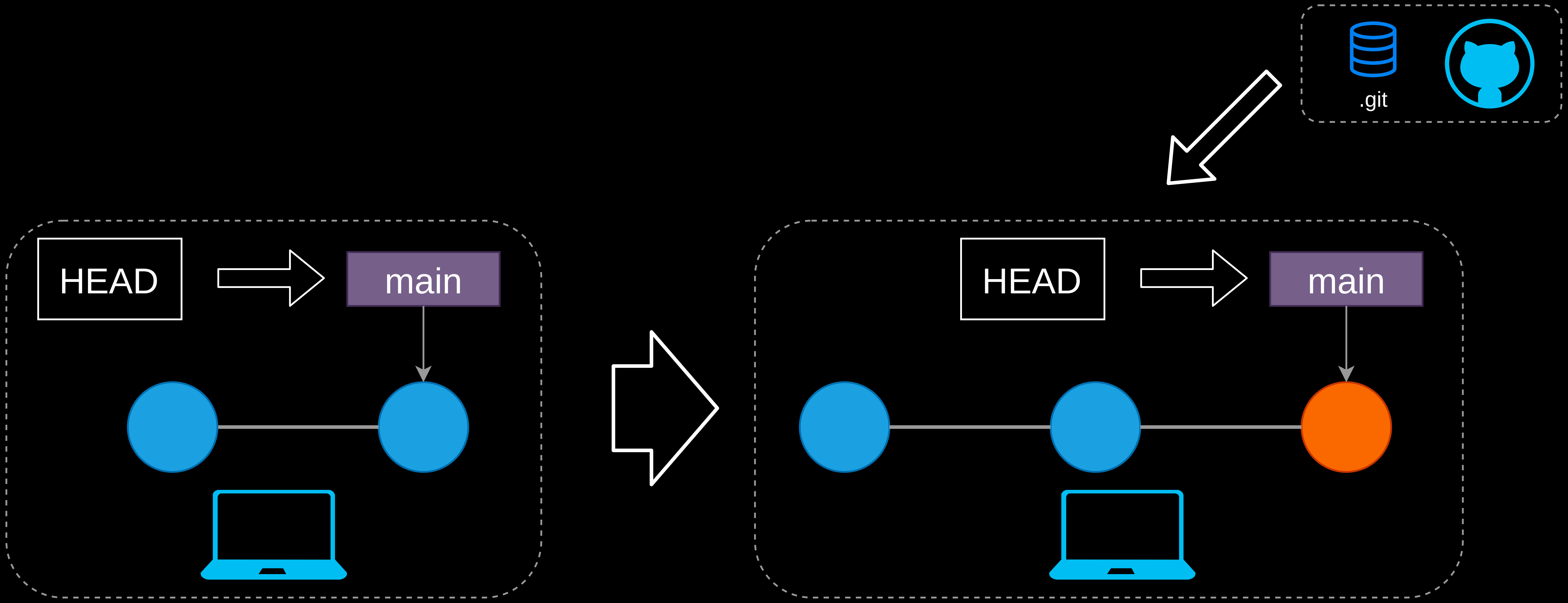


Let’s Create a Repository

Git & GitHub Workflow

1. Sync Local Repository

```
git checkout main
git pull origin main
```

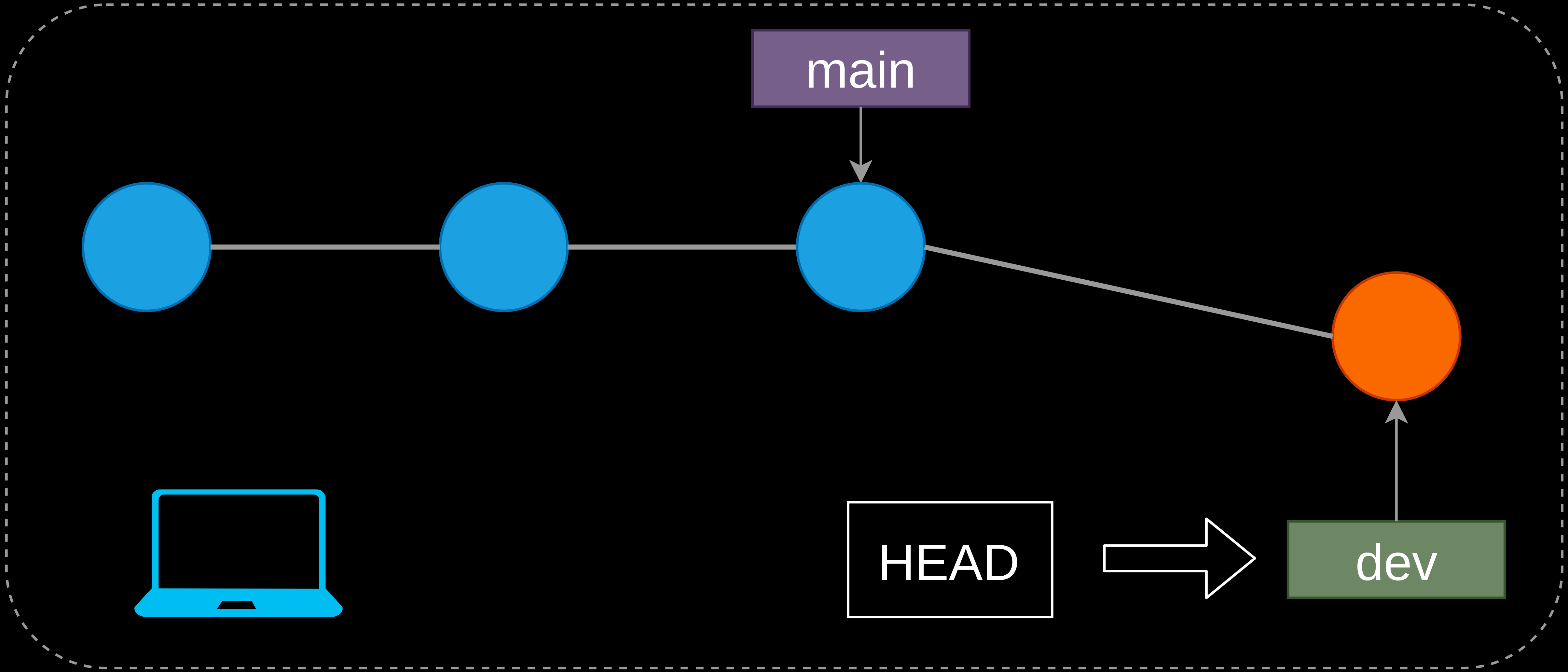


2. Write Your Codes

After checkout to new developing branch

```
git checkout -b "dev"
```

Write your codes

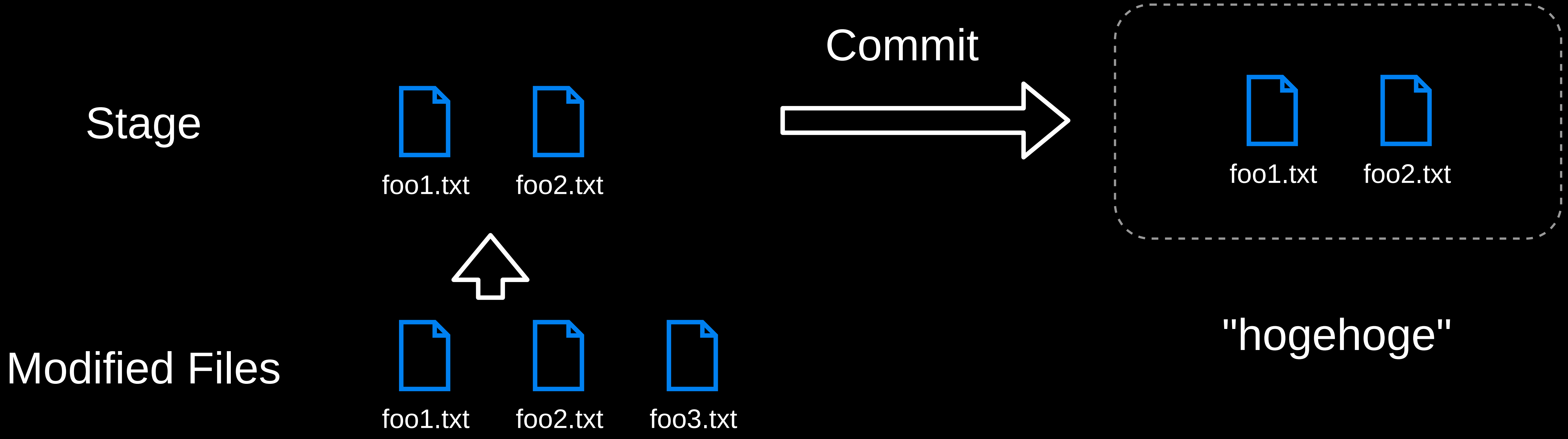


3. Commit

At a good saving point

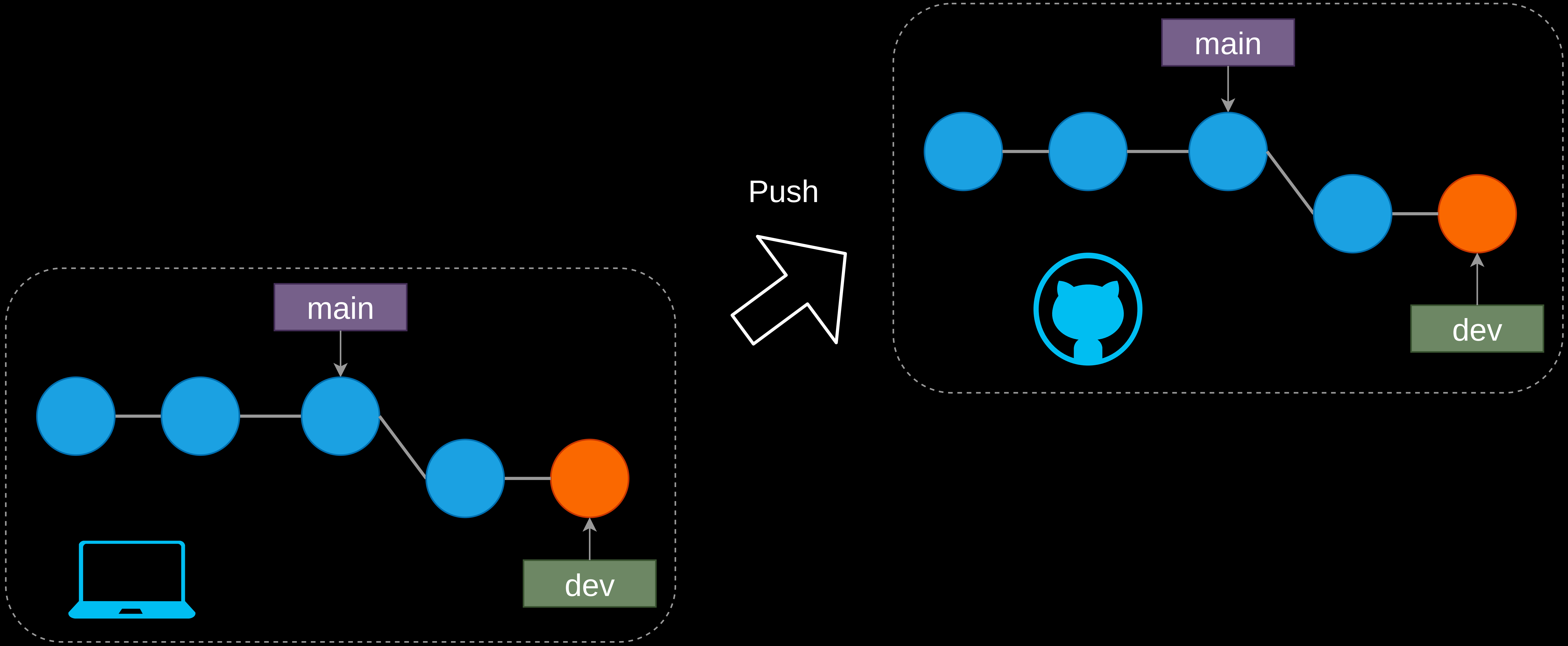
```
git add foo1.txt foo2.txt
git commit -m "hogehoge"
```

If you want to stage all modified files, `git add .`

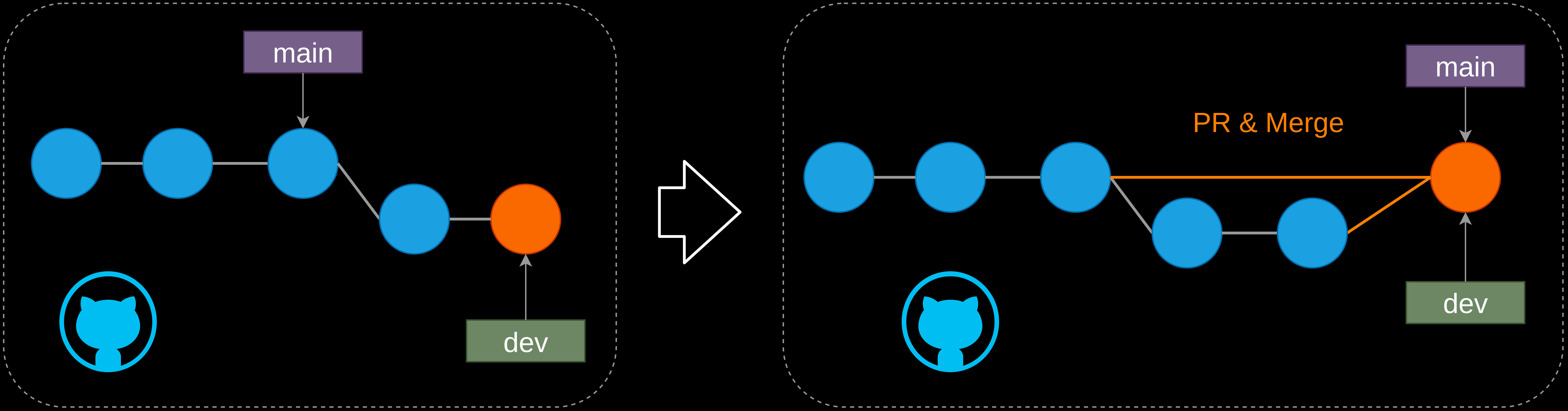


4. Push to the Remote Repository

```
git push origin dev
```



5. Pull Request and Merge



Questions about Branches

Why Do We Use Branch?

- Keep "main" branch clean
- Easy to detect a bug (because "main" works perfectly)

When Should I Create a Branch?

A simple suggestion is "feature branch workflow"

- Create a branch if you want to add a new feature
- For the economic research, model, slides, paper, and a BUG-FIX
- Delete branch when it’s done
  - In remote repository, after the merge, the button appears
  - In local repository, run ``` `git branch -d BRANCH_NAME` ```

Let’s Handson

Git with Data

Data Version Control (DVC)

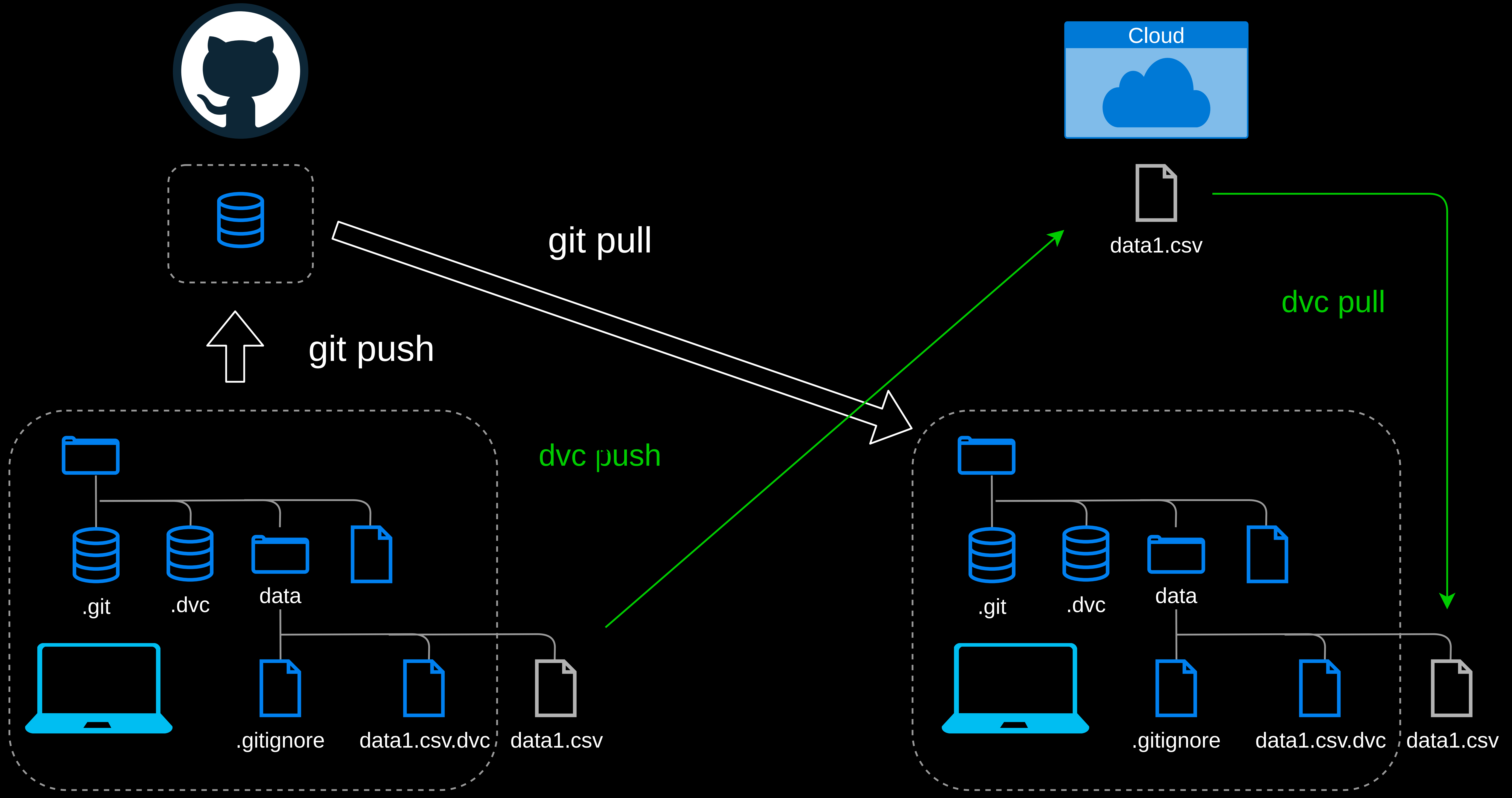
How Can We Work with Data in Git?

- We want to store the data in the Git project (because referred in the code)
- Not interested in change in each line of the data (imagine data cleaning)
- There is a limit for the file size in GitHub (100 MB)

How Does DVC Work?

- Create a text file for the meta-information of the data
- Git manages only the text file
- Git can follow when the data is added, modified, deleted through the text file
- The original data is stored in the remote storage (Google Drive, Amazon S3, …)

How Does DVC Work?



DVC Commands

1. Before your commit, create DVC files

```
dvc add foo1.csv
```

You can specify a folder with *-R* option `` `dvc add -R data` ``

2. After Git push, push data to the remote storage

```
dvc push
```

3. After Git pull, pull data from the remote storage

```
dvc pull
```

Let’s Handson

Troubleshoot

Git Reference log

You can see all the git command activity

```
git reflog
```

Git Reset

You can reset any git activity

```
git reset --soft COMMIT_ID_OR_REFLOG_ID
```

Conflict

Conflict might occur

- When merge
- If the main and development branches have different lines of codes in the same file

To solve this

- You can open a text editor
- Choose which lines you keep

Learn More

Introduction to Git   Introduction to GitHub