



Git : Part 2

Branches & Merges

Amine FERDJAOU
amine.ferdjaoui@sogeti.com

Année universitaire 2024/2025

BRANCHES

.....

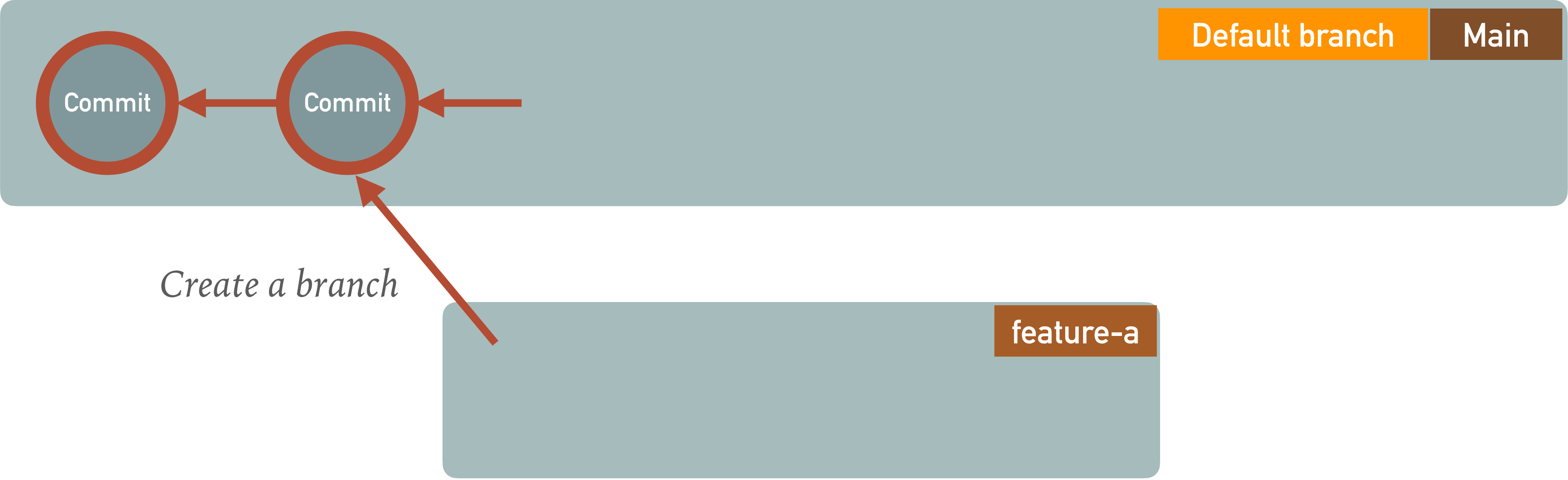


Branches

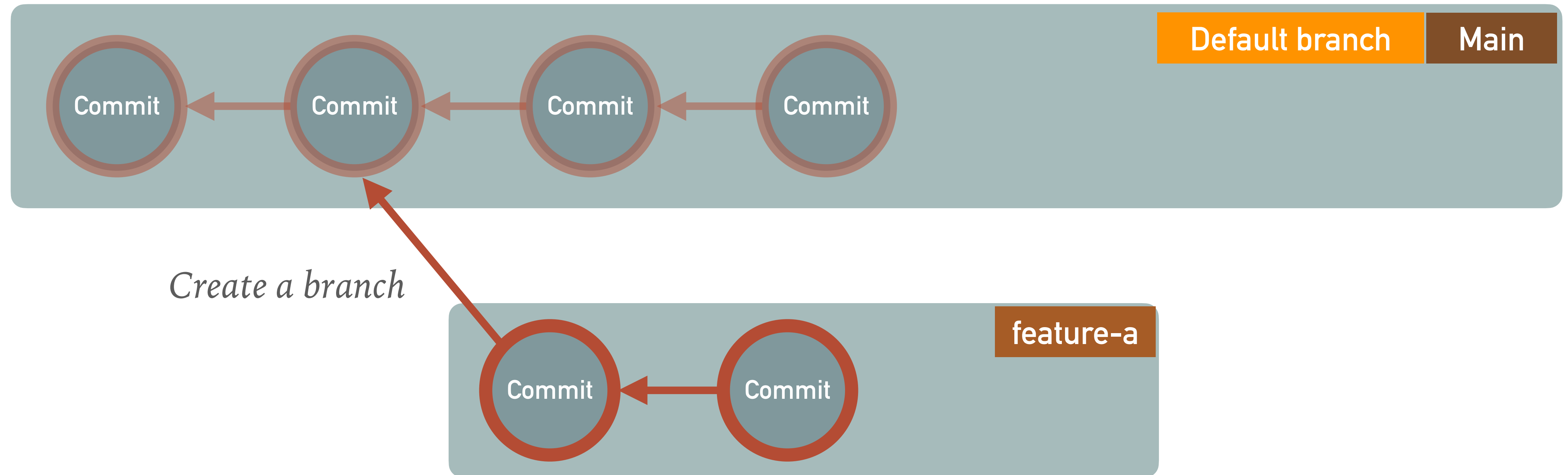


*in old git versions the default branch is called "*master*"

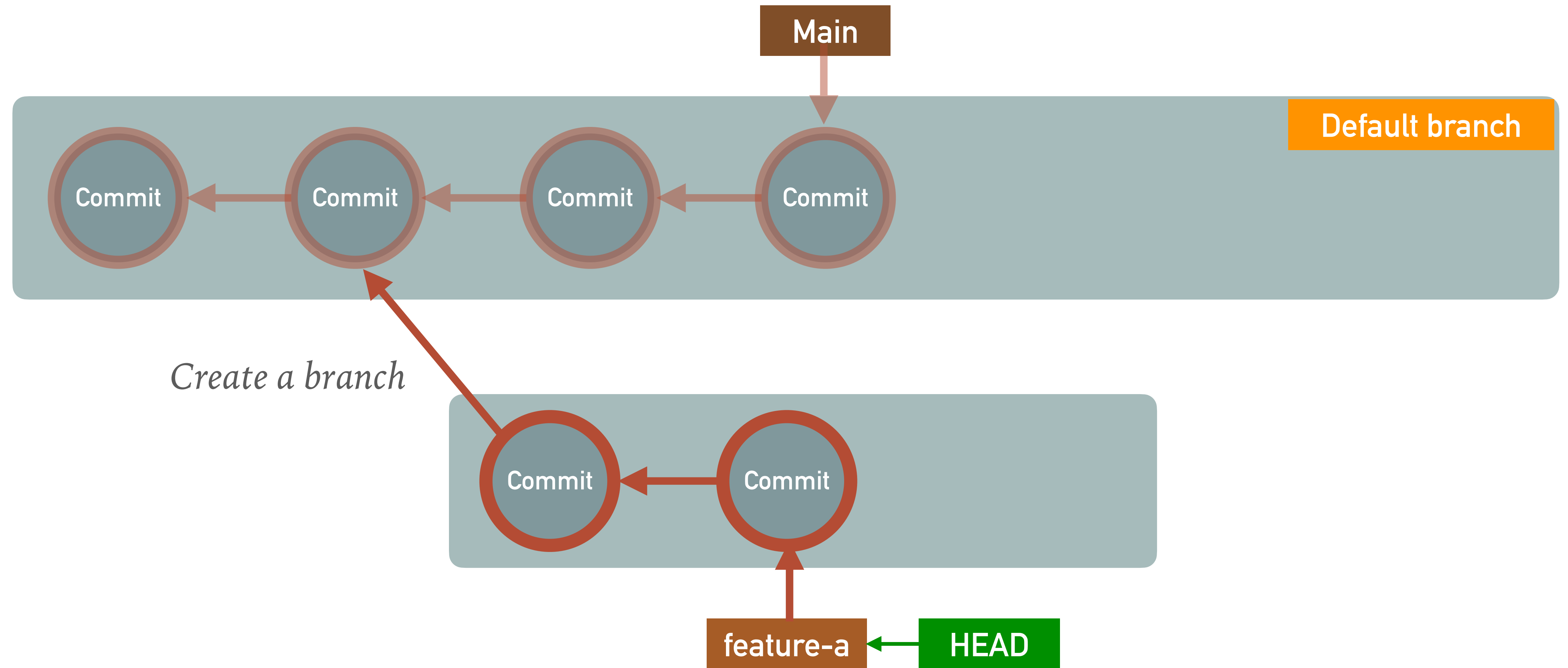
Branches



Branches



Branches



Git restores the working directory according to the position of the HEAD pointer

Git branches

Get the current branch

```
$ git branch  
$ git branch -a #for all
```

Ps. Always commit before changing branch (otherwise use git stash et git stash apply)

Create new branch and switch

```
$ git branch <branch-name>  
$ git switch <branch-name> #or git checkout  
$ git switch - #go back to the last branch  
$ git switch -C <branch-name> #create and jump into new branch / or git checkout -b
```

Delete a branch

```
$ git branch -d <branch-name> # to delete
```

Log & diff

Log commits of all branches

```
$ git log --oneline --all
```

```
$ git log --oneline --all --graph #you can save it as alias
```

```
$ git log <branch-A>..<branch-B> #commit difference between 2 branches
```

See differences between two branches

```
$ git diff <branch-A>..<branch-B> #incide file difference
```

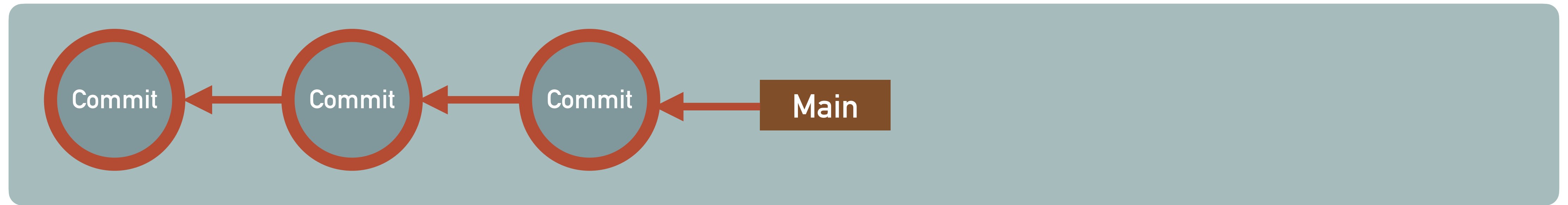
```
$ git diff --name-status <branch-A>..<branch-B> #file difference
```


MERGE

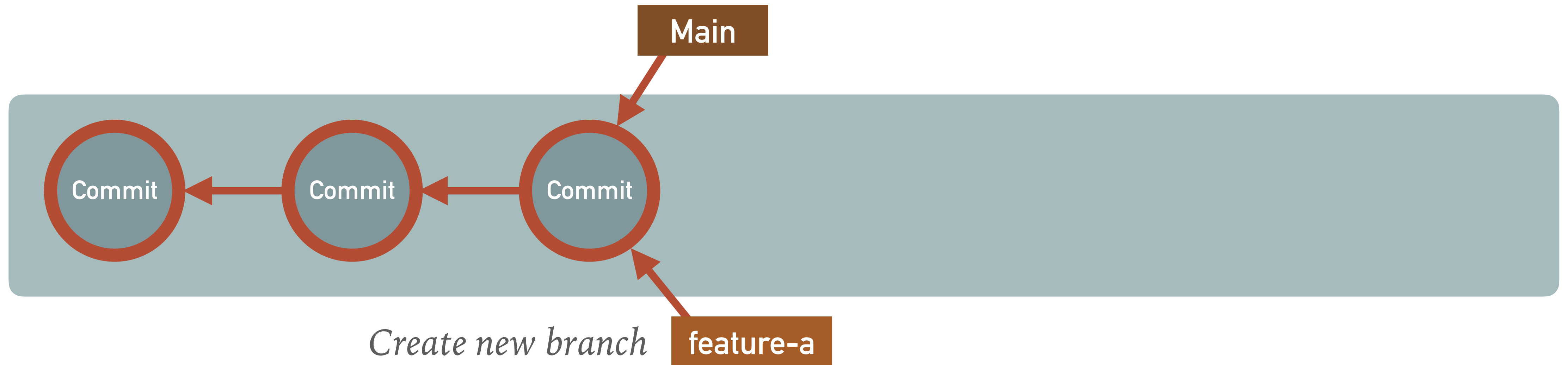
.....



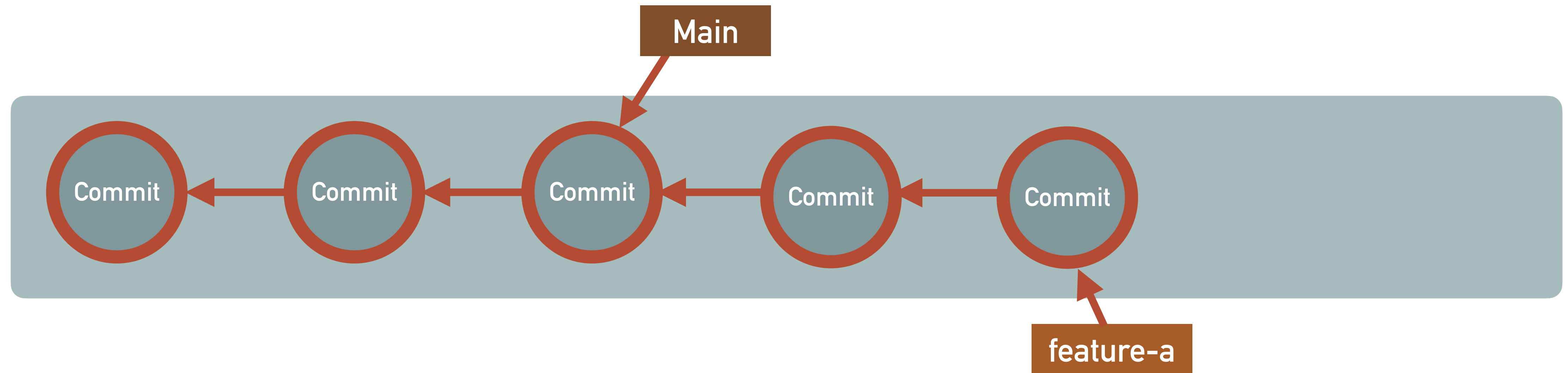
Merges



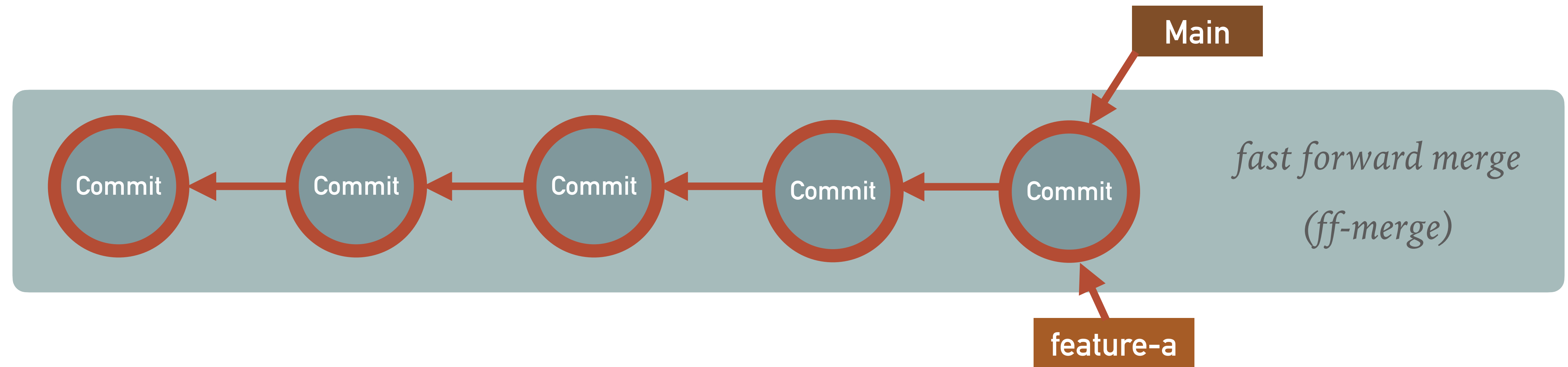
Merges



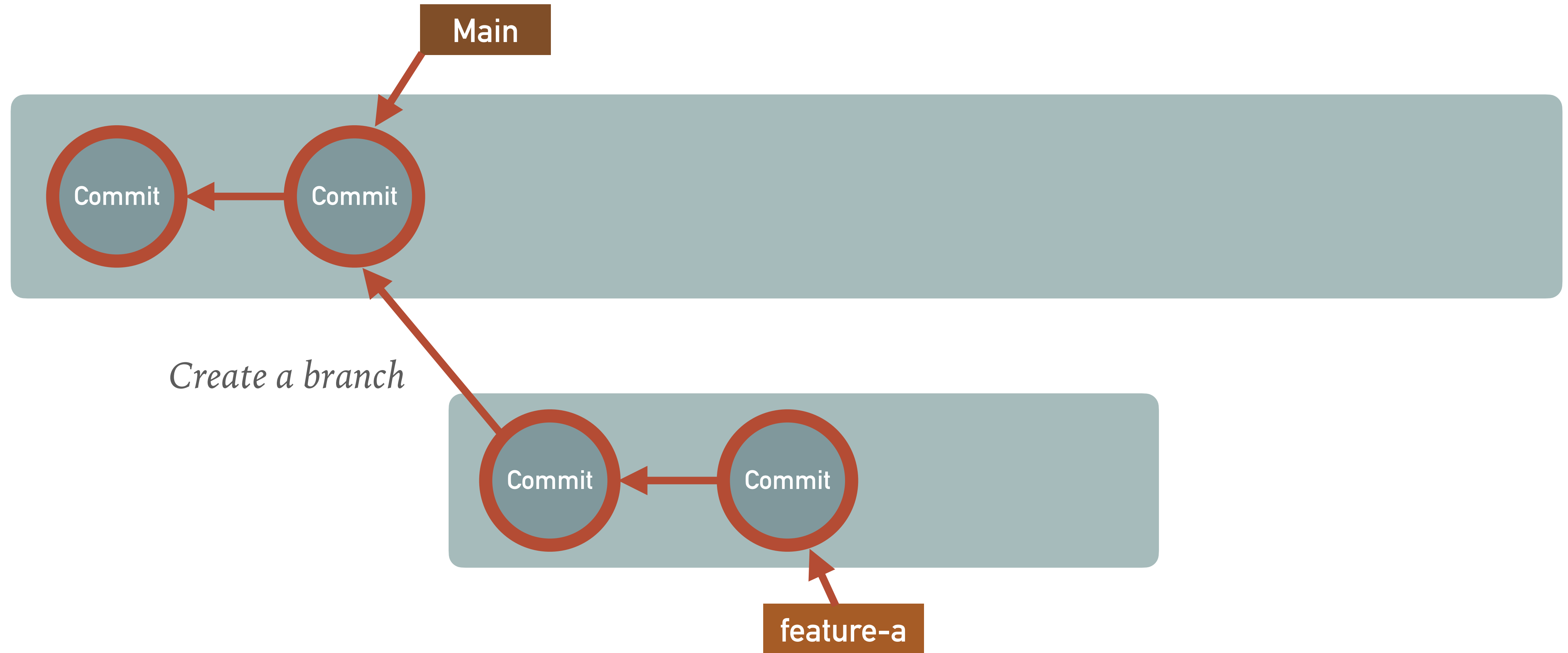
I. Fast forward merges : linear changes



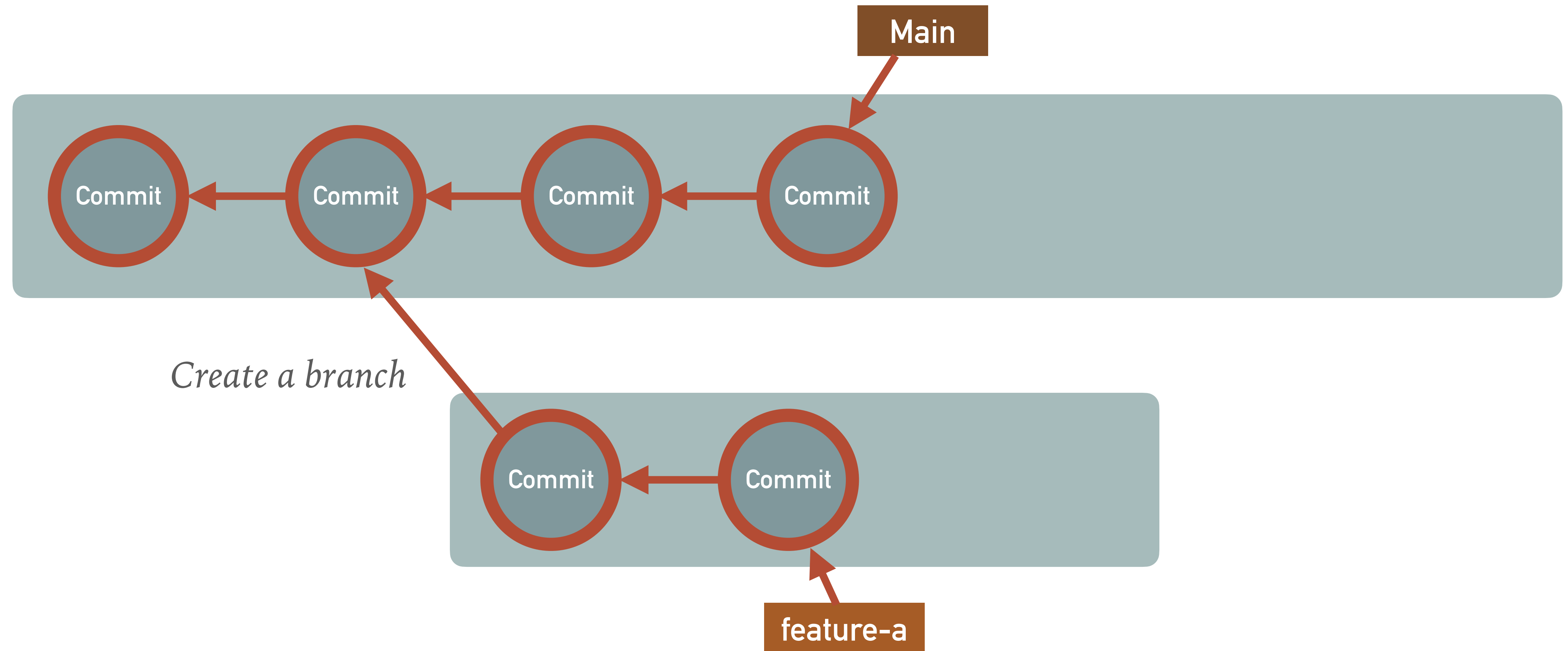
I. Fast forward merges : linear changes



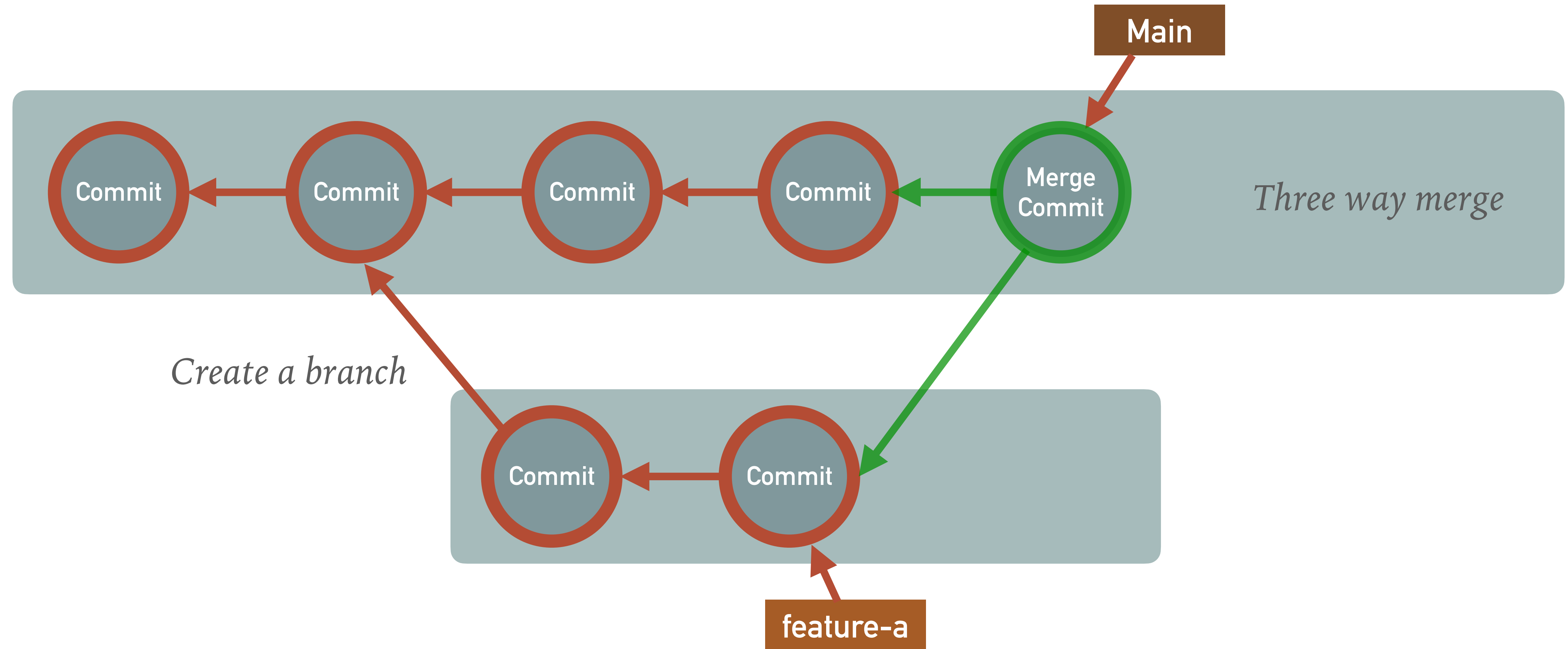
II. a. Three way merge



II. a. Three way merge



II. a. Three way merge



Git merge

Go to the main branch and merge with another branch

```
$ git merge <name-of-the-branch>
```

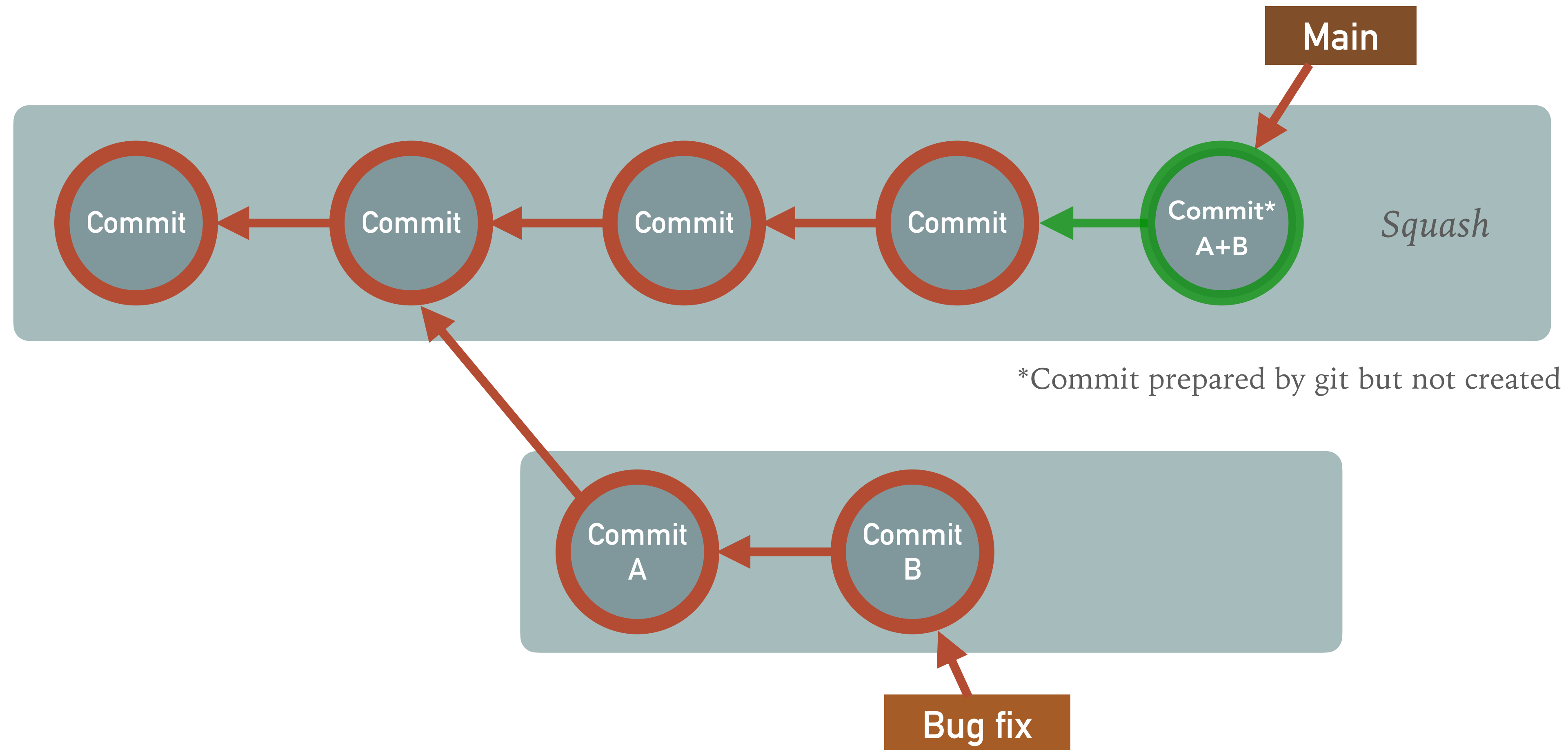
```
$ git log --oneline --graph --all
```

```
$ git switch main
```

```
$ git merge classification
```

```
$ git branch -d classification #once your merge your branch with the main you can delete it
```

II. b. Git merge --squash



Git merge --squash

In case you want to add branche's changes without merge you can do it using squash

```
$ git merge --squash <name-of-the-branch>
```

- Create a branch named : bug-fix
- Add two new files and commit changes
- Squash the merge

```
$ git switch main
```

```
$ git merge --squash bug-fix
```

```
$ git commit
```

```
$ git branch -D bug-fix #You have to force deletion because the branch is considered unmerged
```

Ps. Conflicts are resolved in the same way as merge

CONFLICTS

.....



Conflicts

- ◆ Conflicts may occur for different reasons :
 - ◆ Different versions of the same line inside a file
 - ◆ File content different from one branch to another
 - ◆ File no longer existing

Fix conflict

```
$ vi pre-processing  
$ git merge  
$ git add pre-processing #get the change back to the staging  
$ git commit
```

Nb. In case you don't want to resolve the conflict you can go back to the previous state

```
$ git merge --abort
```