

HOW TO BE A GIT MASTER

QUESTIONS

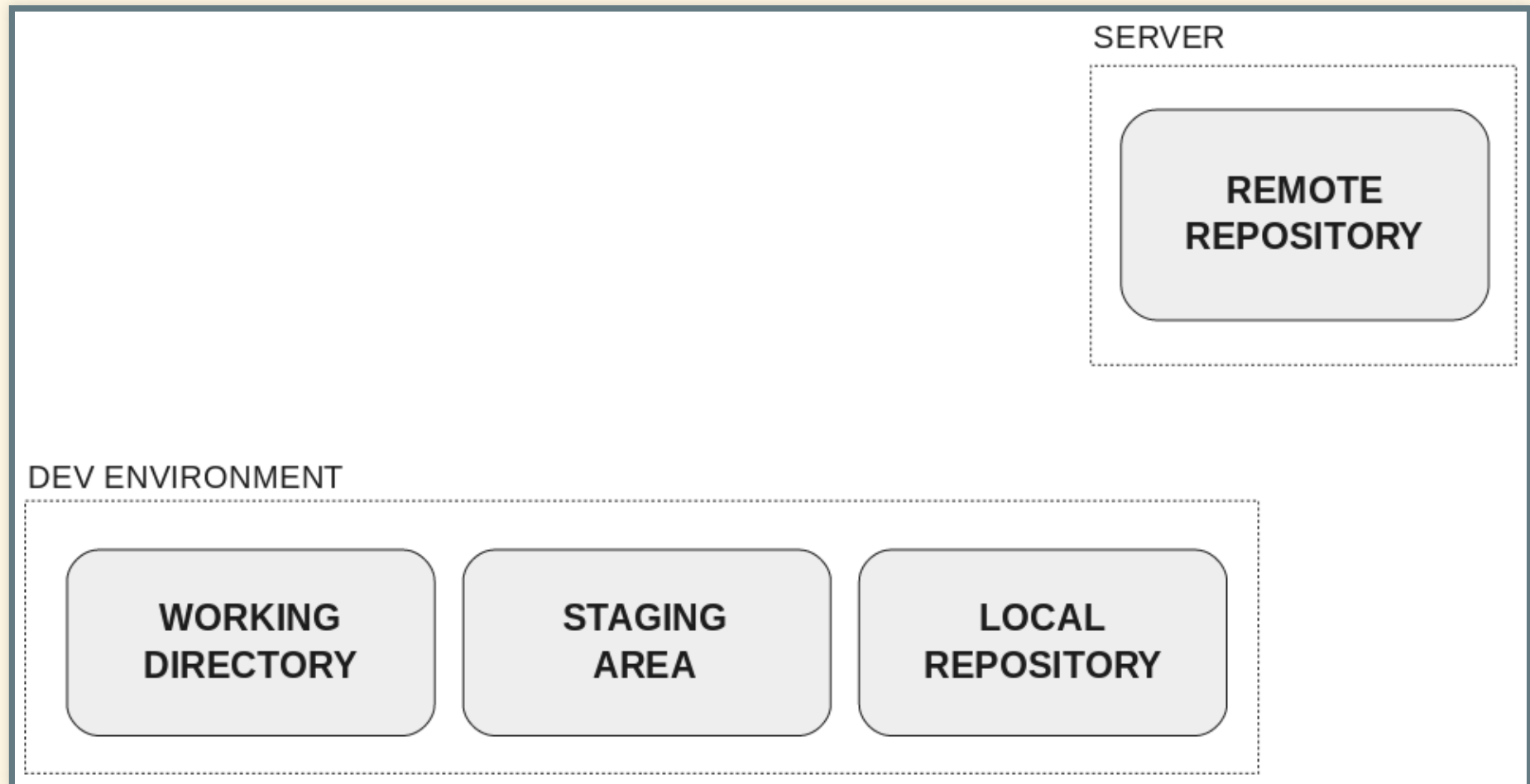
- What other VCS have you used?
- Why use Git?
- Why does Git have a staging area?

ANATOMY OF A GIT

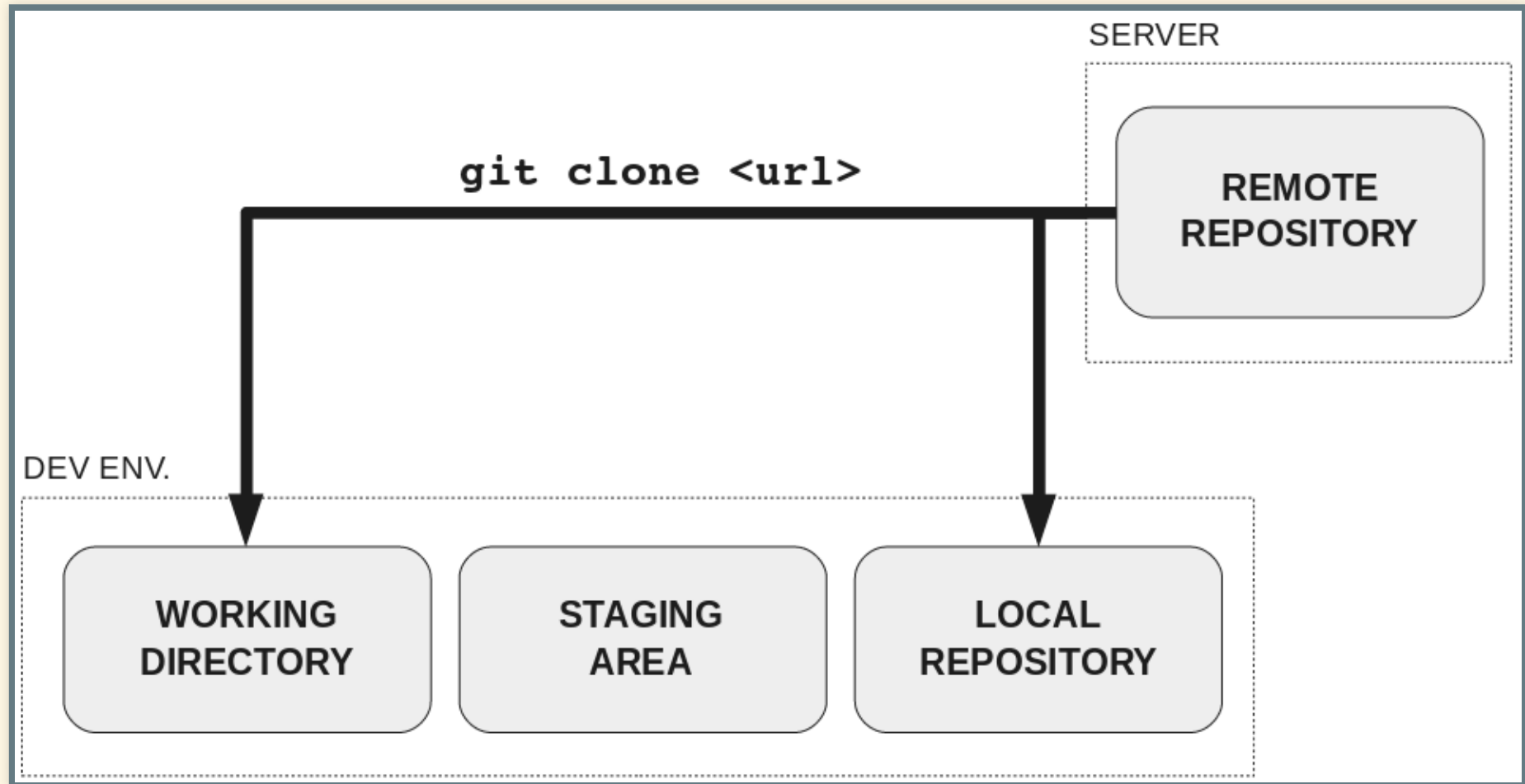
Images from Rachel M. Carmena:

<https://rachelcarmena.github.io/2018/12/12/how-to-teach-git.html>

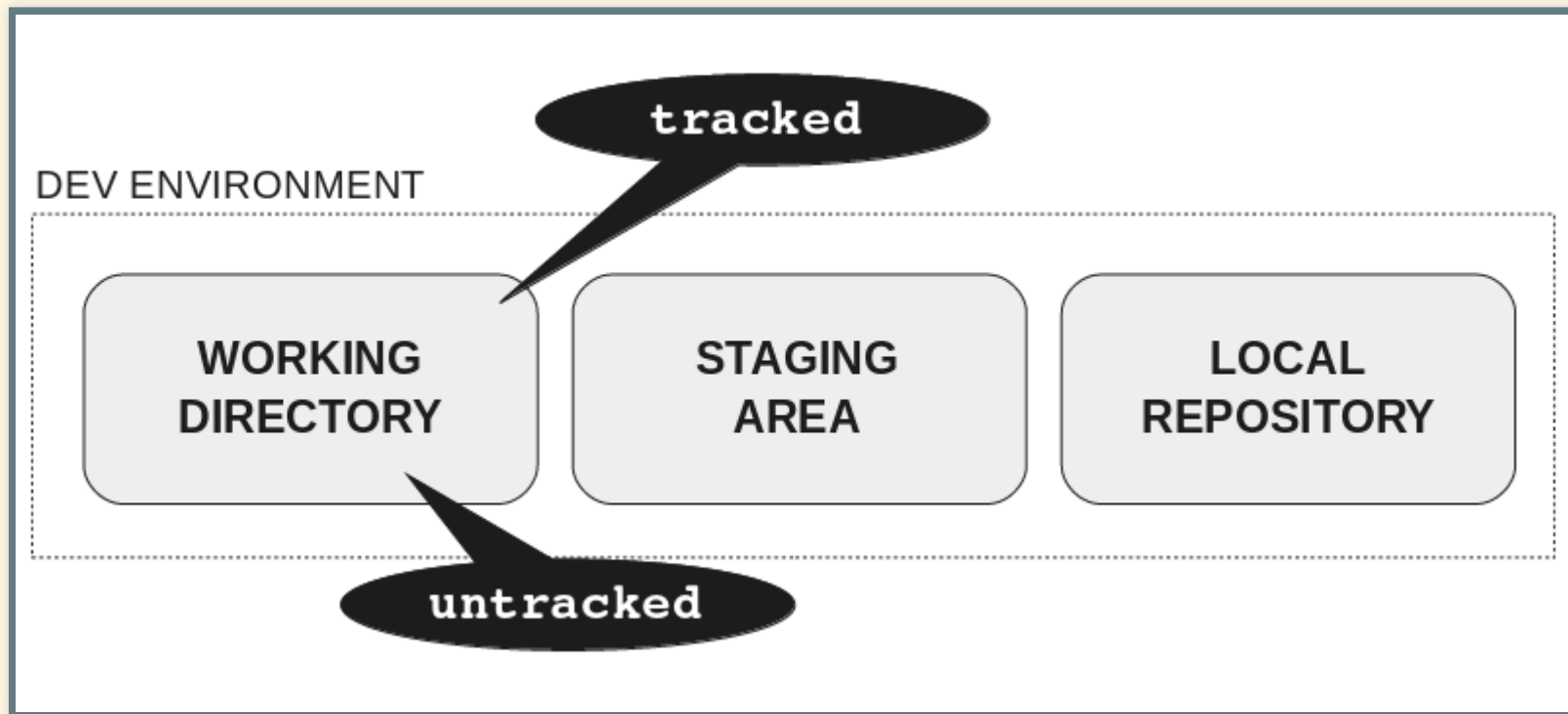
DISTRIBUTED VCS



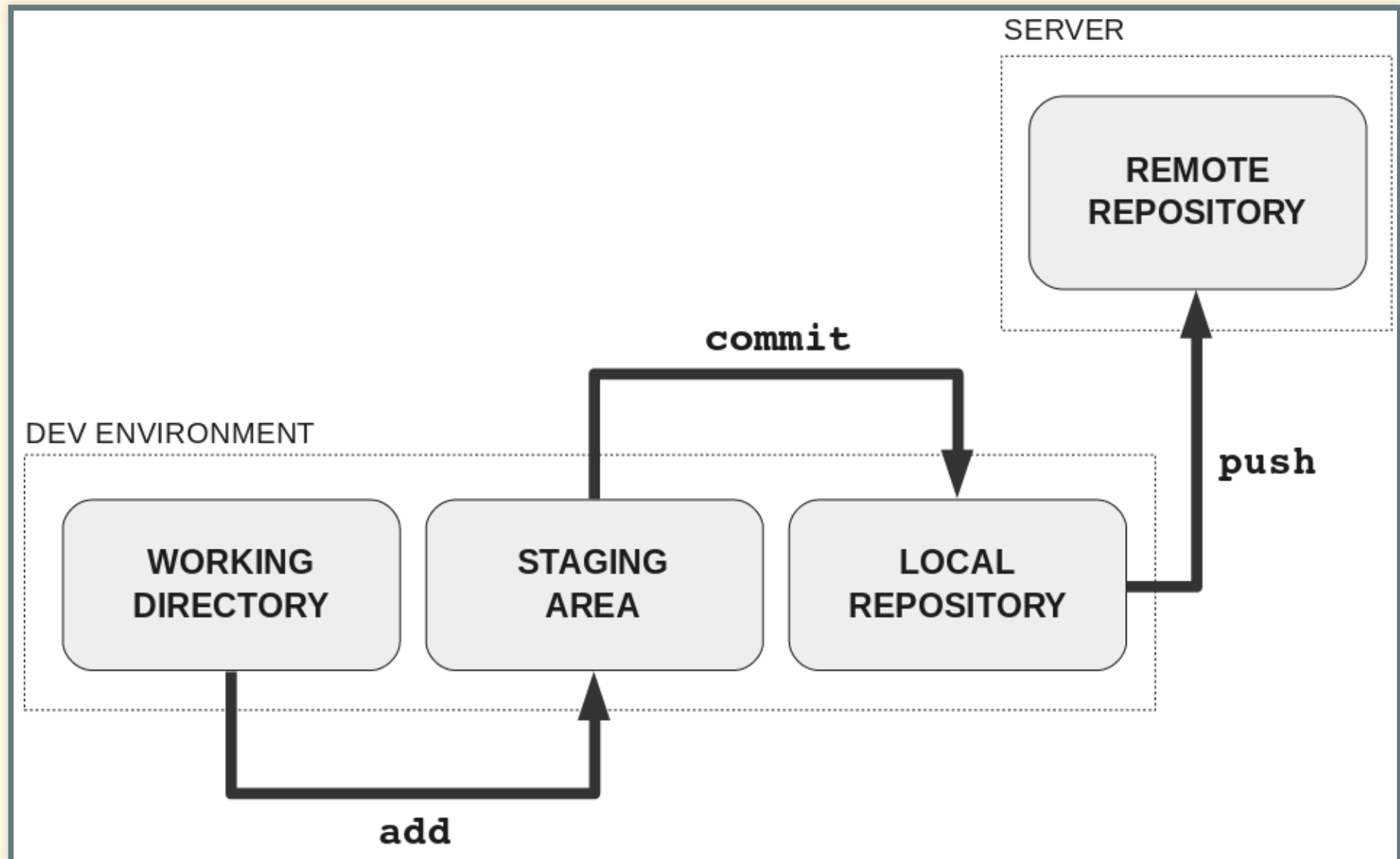
CLONING A REPO



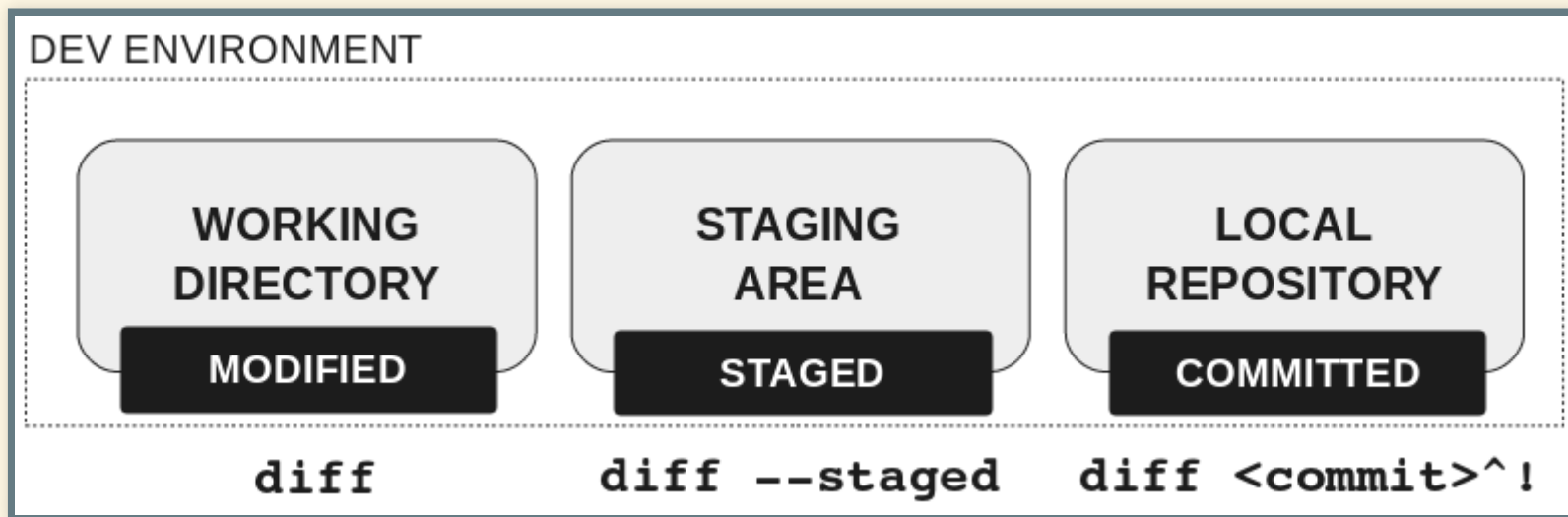
MAKING CHANGES IN THE WORKING DIRECTORY



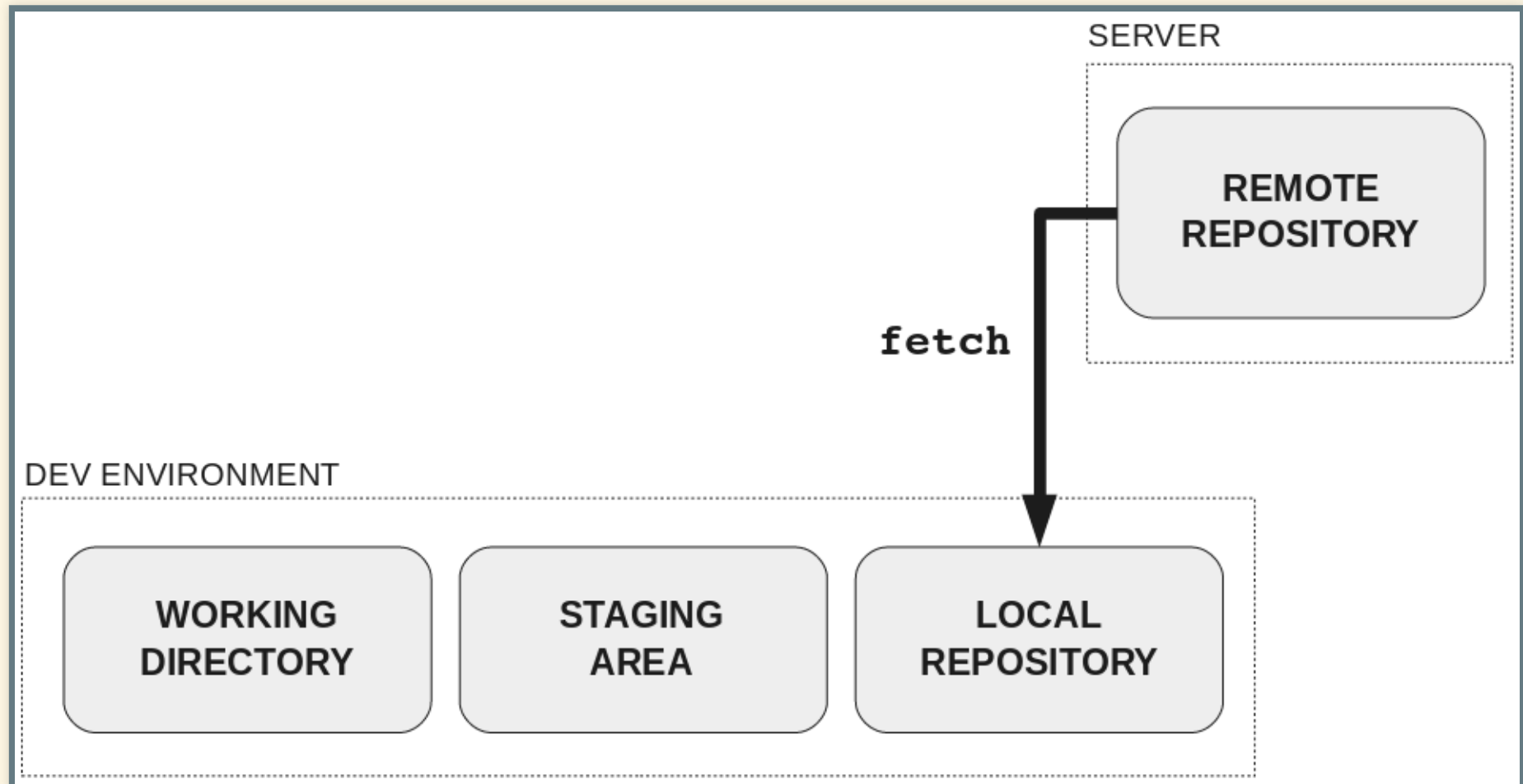
UPDATING THE REMOTE REPOSITORY 1/2



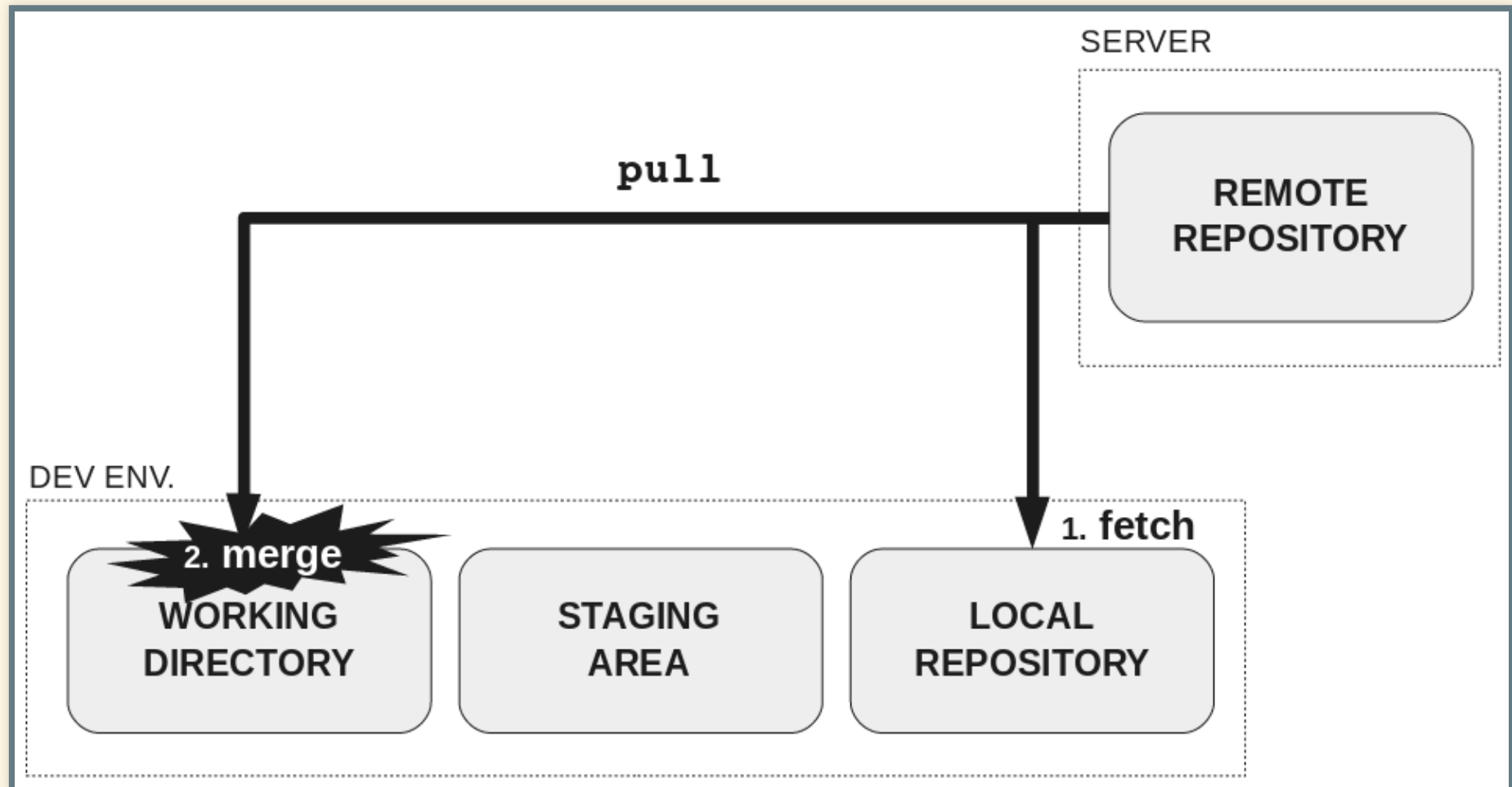
UPDATING THE REMOTE REPOSITORY 2/2



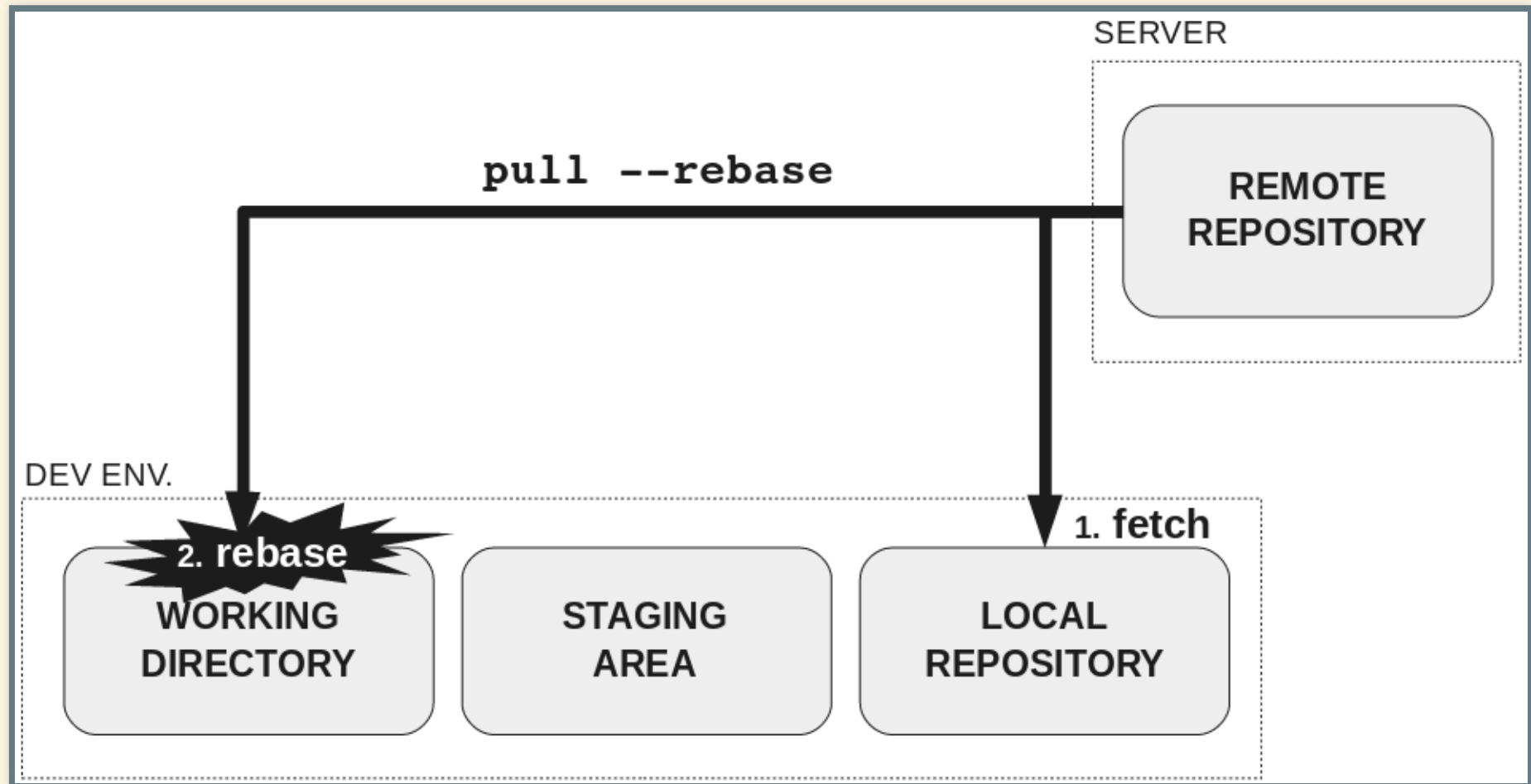
UPDATING THE DEVELOPMENT ENVIRONMENT 1/3



UPDATING THE DEVELOPMENT ENVIRONMENT 2/3



UPDATING THE DEVELOPMENT ENVIRONMENT 3/3



EXERCISE!

(IN PAIRS)

- <http://bit.do/learngit1>

SERVER

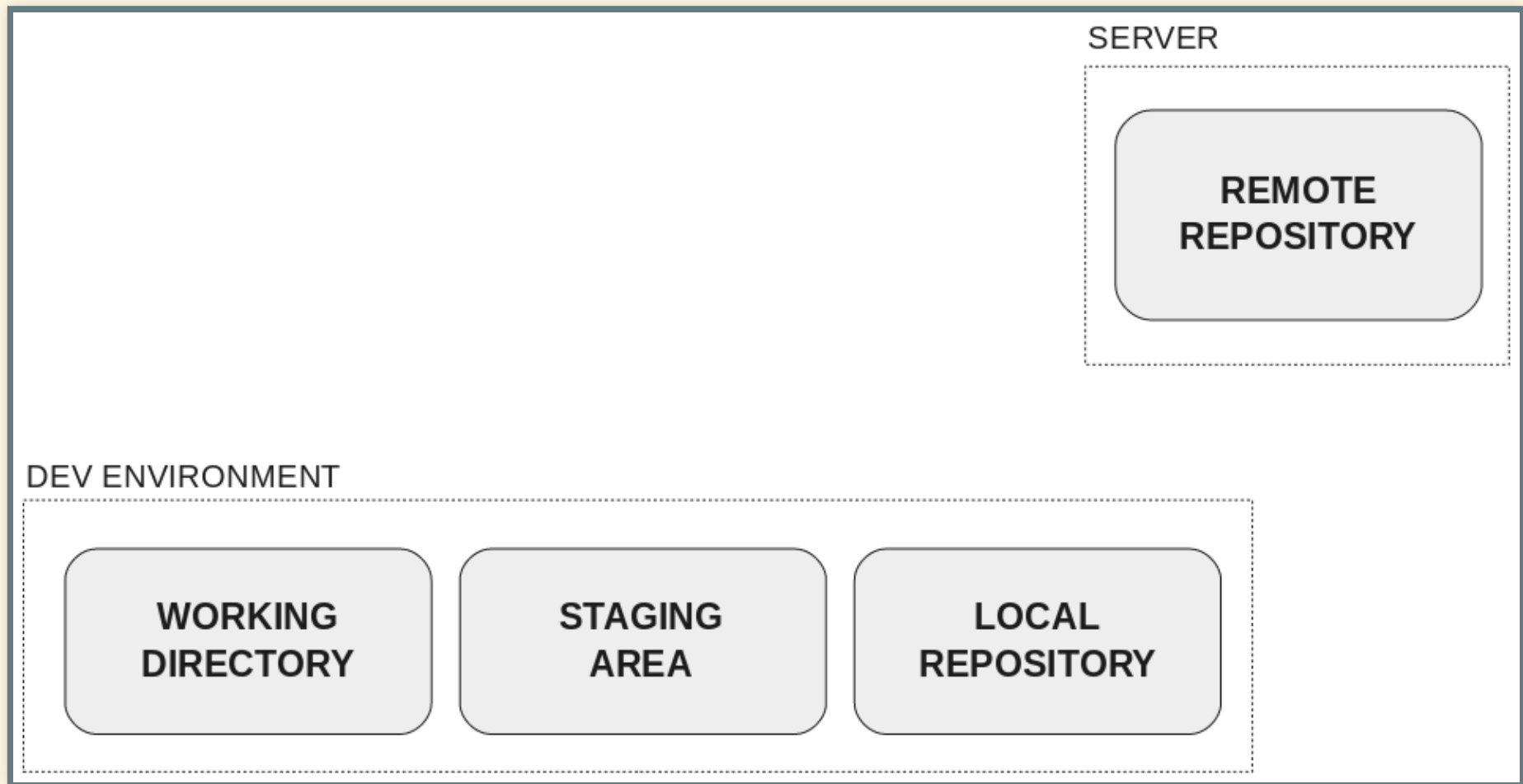
**REMOTE
REPOSITORY**

DEV ENVIRONMENT

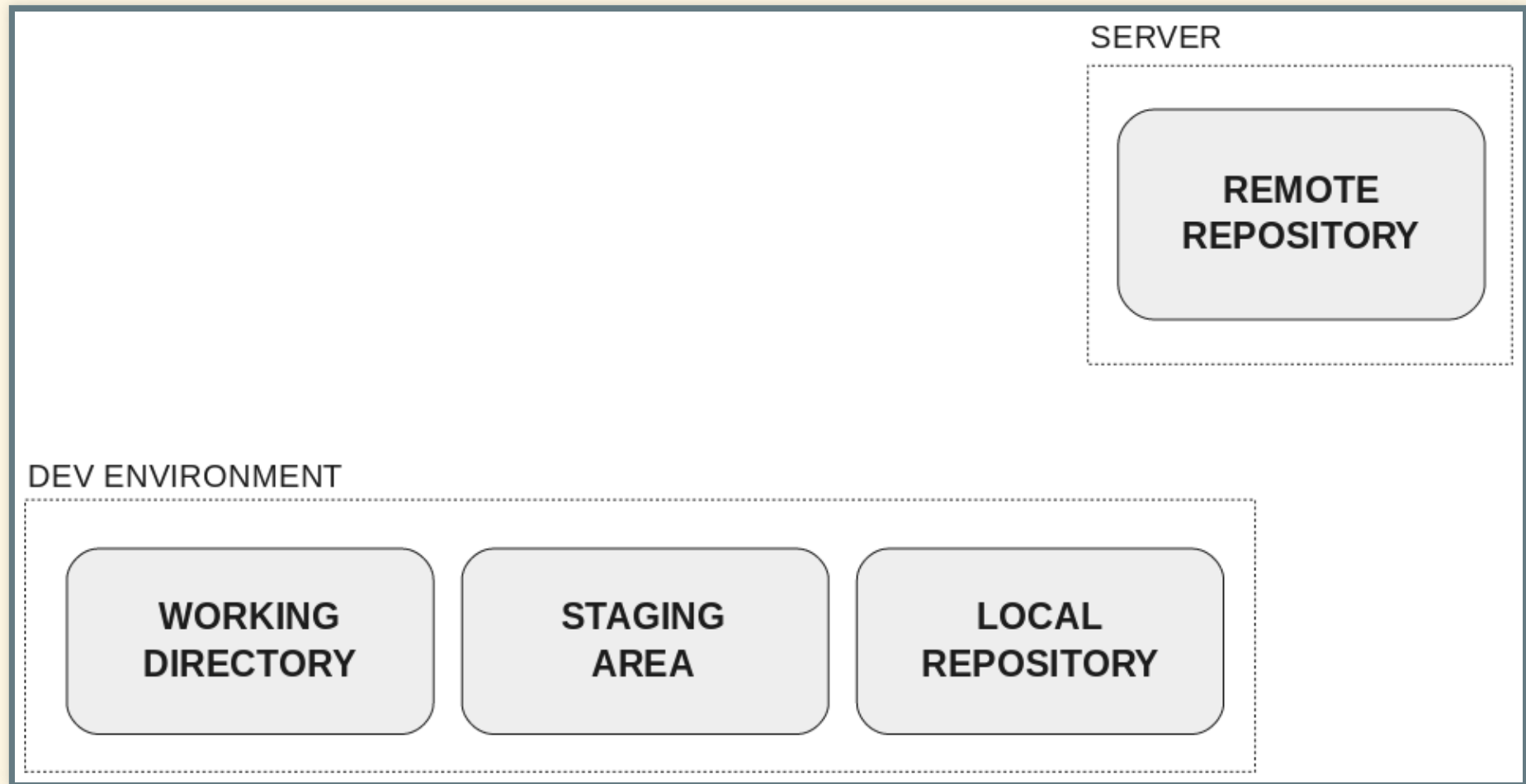
**WORKING
DIRECTORY**

**STAGING
AREA**

**LOCAL
REPOSITORY**



ACTIVITY: DRAW THE ANATOMY



**WHEN THINGS GO
WRONG**

QUESTIONS (IN PAIRS)

- What do you get stuck with in Git?

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



DESTRUCTIVE COMMANDS

- checkout
- reset
- gc

```
git reflog
```

```
->
```

```
git reset HEAD@{index}
```



EXERCISE!

(IN PAIRS)

- <http://bit.do/learngit2>

DISCUSSION (IN FOURS)

- What is one fact you've learned about getting unstuck?

REBASE VS MERGE

DISCUSSION (IN PAIRS)

- Do you usually use merge or rebase?
- Why?

MERGING THROUGH THE MEDIUM OF LEGO

MERGE

- No new commits
- Local changes (merge commit)
- Local changes with conflict

REBASE

- Change local history
- Avoid uninteresting merge commits
- Change shared history (force push)

FEATURE BRANCH

- rebase vs merge from master

REBASE WITH CONFLICTS

- Reuse recorded resolution

```
git config --global rerere.enabled 1
```

EXERCISE!

(IN PAIRS)

- <http://bit.do/learngit3>

ACTIVITY

(IN 3S)

- Use the LEGO to show how a rebase or merge happens in Git

FOUR MODERN WORKFLOW MODELS

ACTIVITY

(IN 3S)

- Draw and show how changes happen in your teams
Git repo

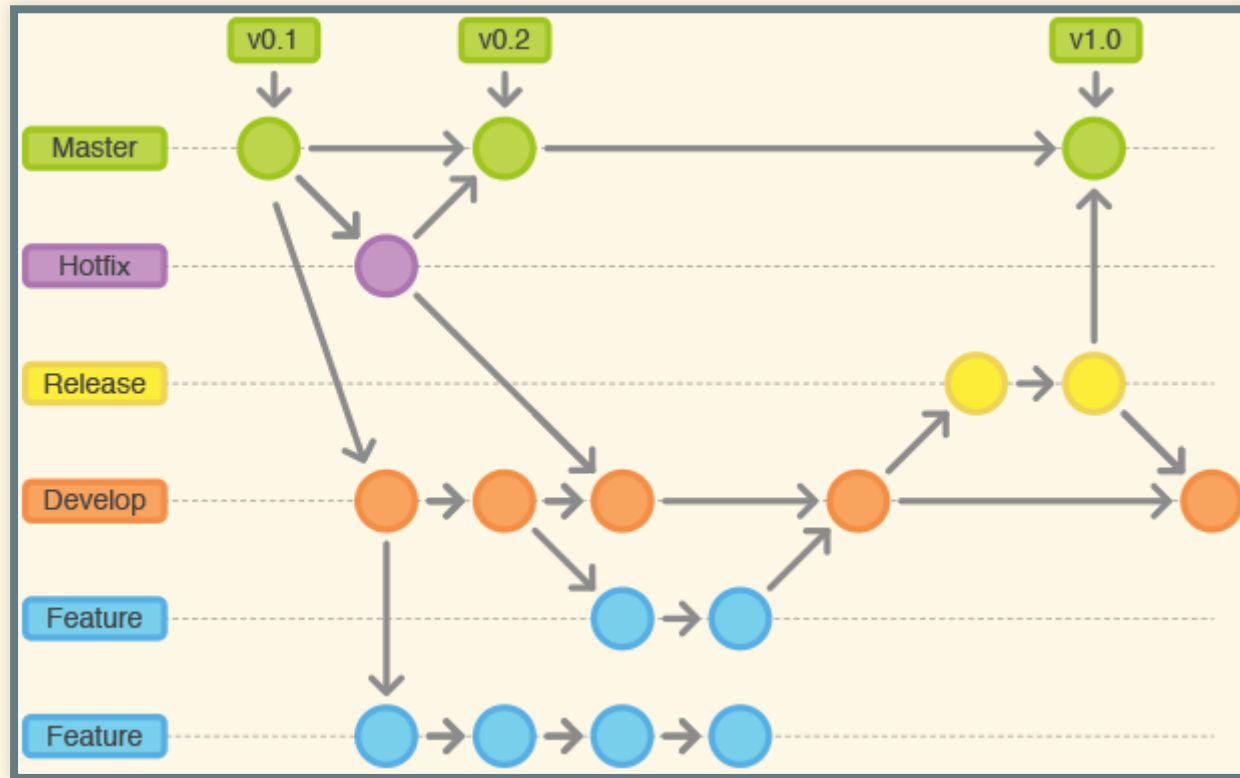
COMMIT MESSAGES

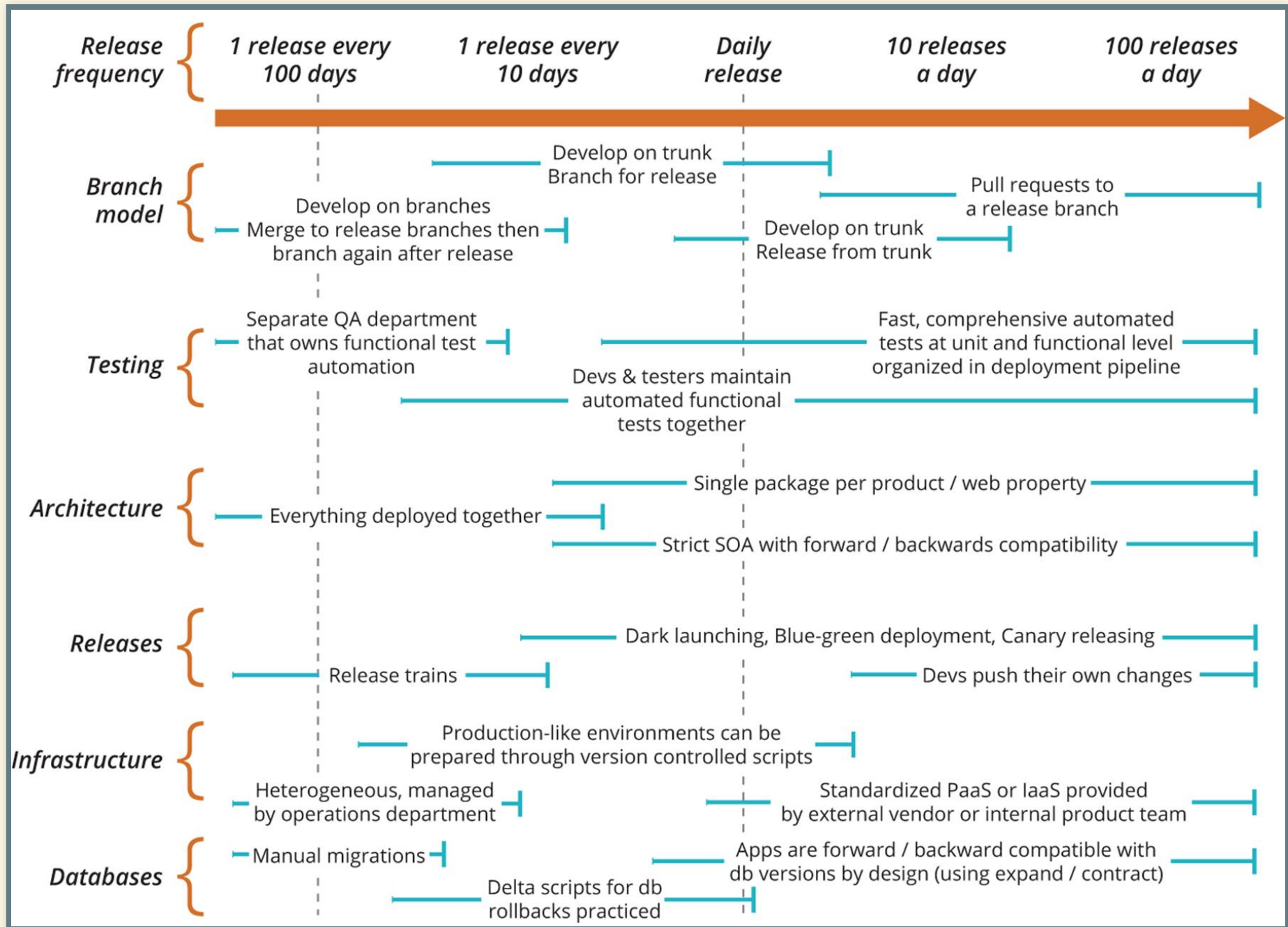
If applied this commit will...

```
git commit -m 'Add logging to the  
fizzbuzz'
```

FOUR MODERN WORKFLOW MODELS

- Trunk based
- Feature branch
- GitFlow
- Forking





ACTIVITY

(SOLO)

- Write three facts you've learned on back of index card
- Share one with the whole group