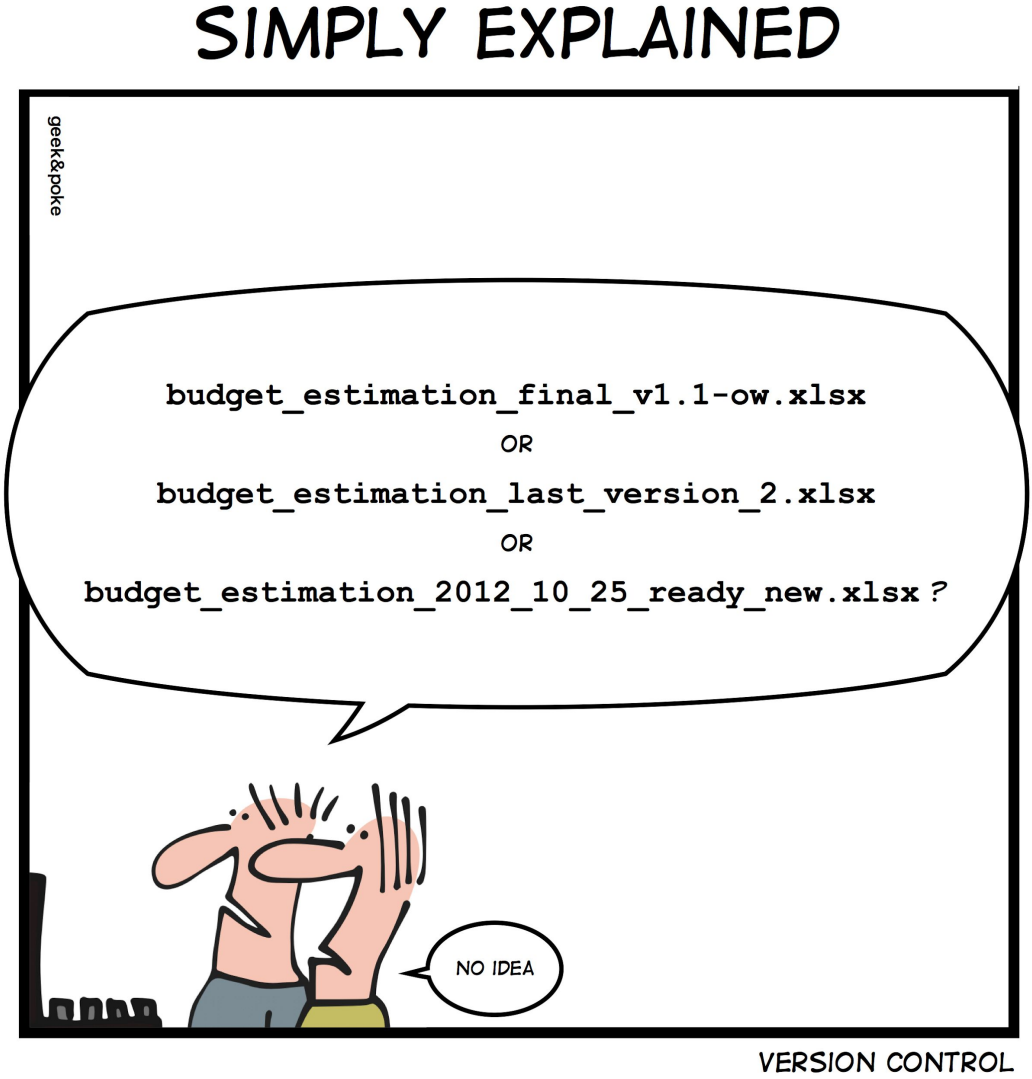
VER 1.0
VERSION CONTROL
VER 2.0
VER 1.1
VER 1.2

What is version control?

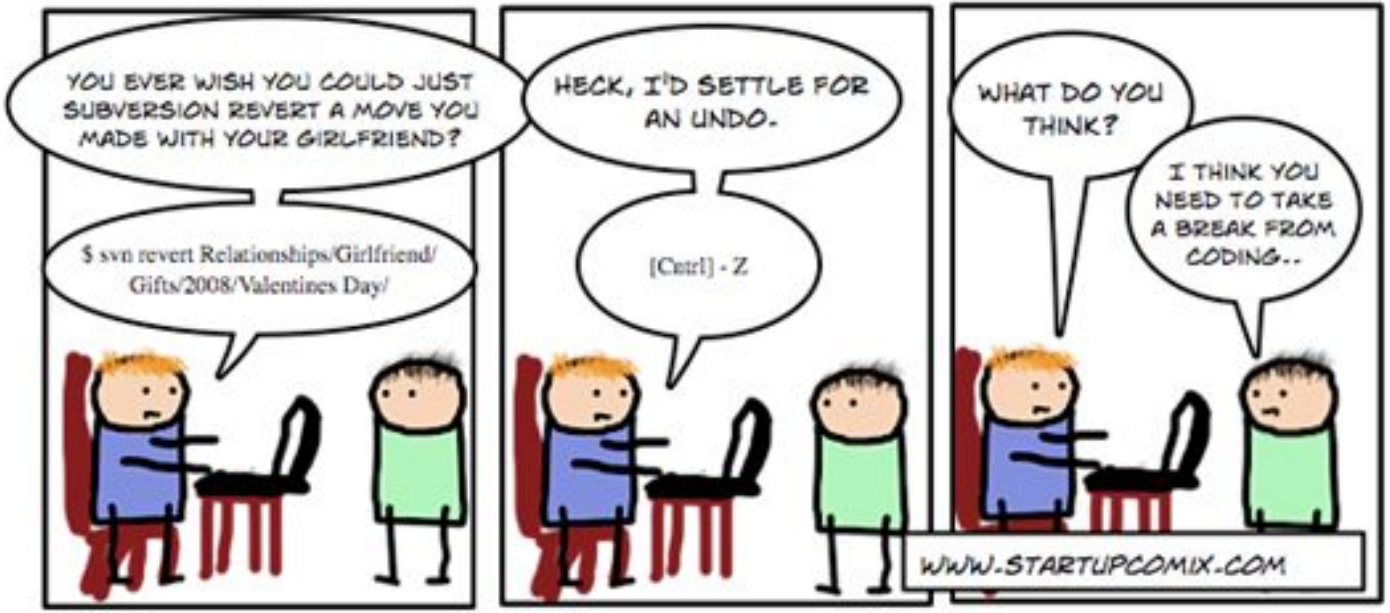
Track your files over **TIME**.

Why do you need this?



What about External Drives?

What about going back to a certain Version?



The elegant Way

[master] 6c6faa5 My first commit - John Doe

[develop] 3e89ec8 Develop a feature - part 1 - John Doe

[develop] e188fa9 Develop a feature - part 2 - John Doe

[master] 665003d Fast bugfix - John Fixer

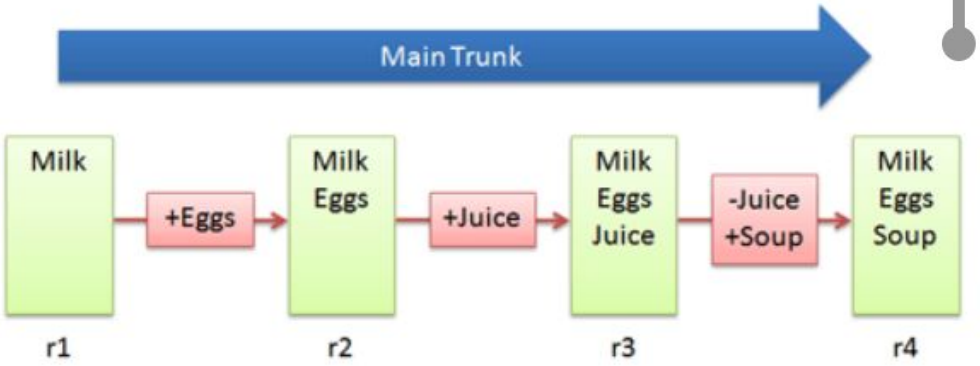
[myfeature] eaf618c New cool feature - John Feature

[master] 8f1e0e7 Merge branch `develop` into `master` - John Doe

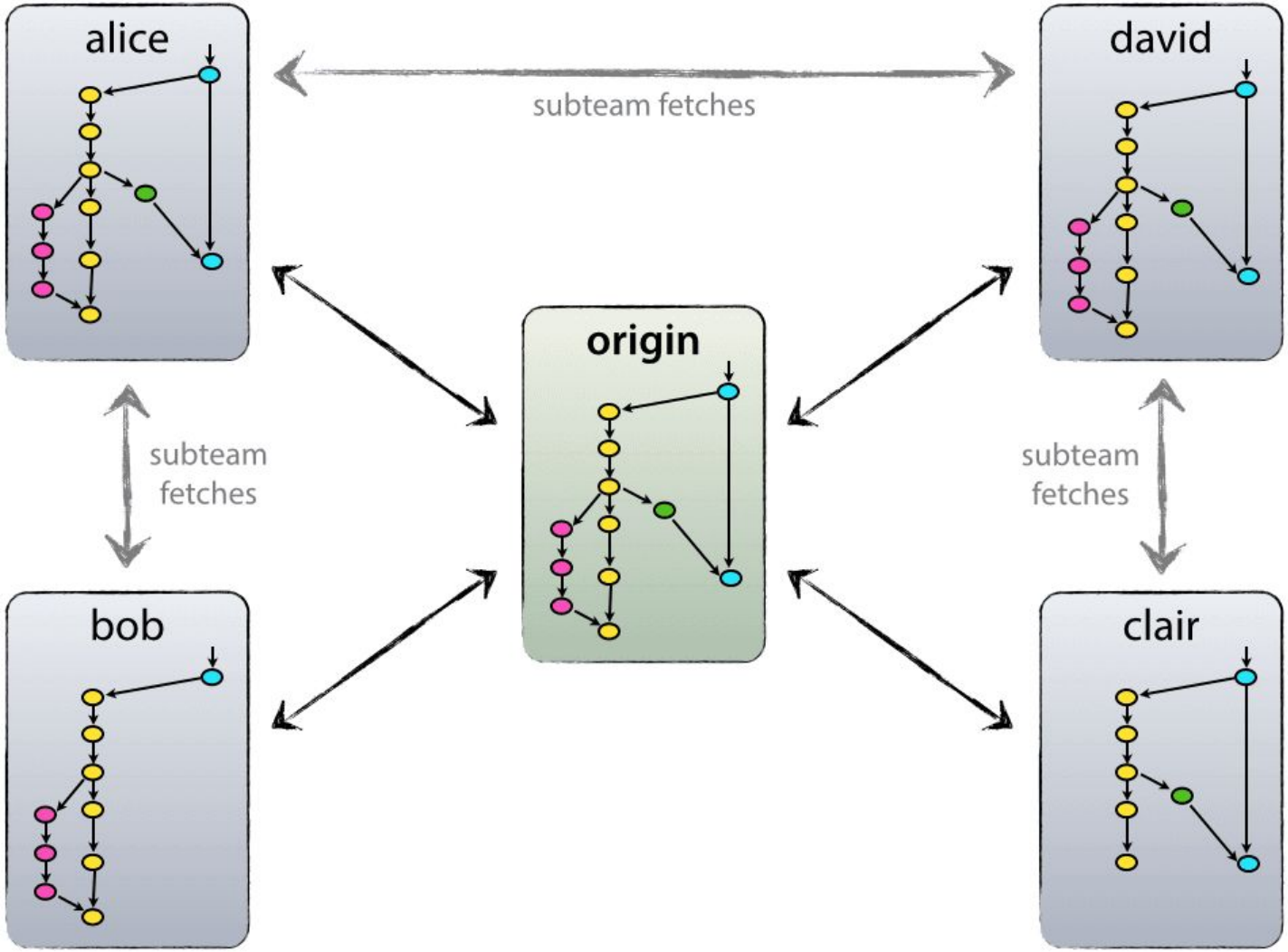
[master] 6a3dacc Merge branch `myfeature` into `master` - John Doe

0.1

[master] abcdef0 Release of version 0.1 - John Releaser

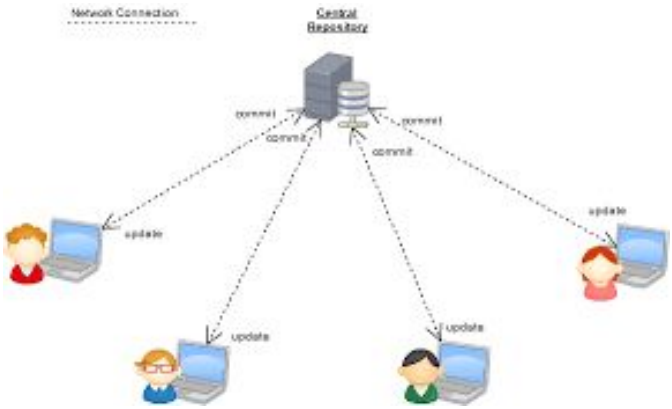


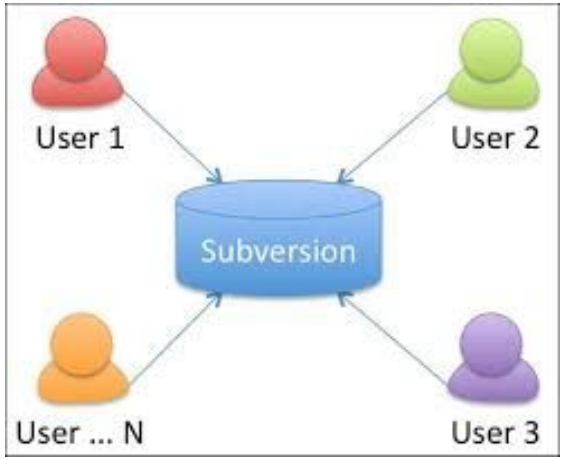
git

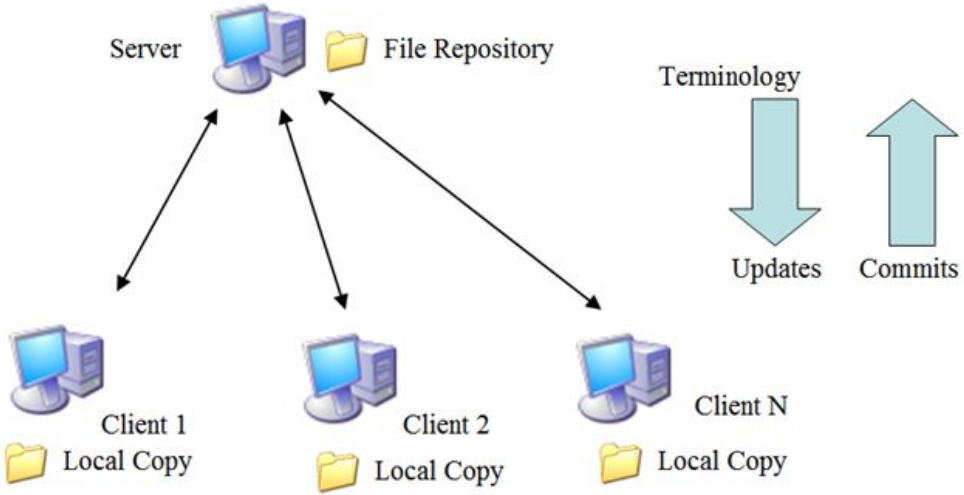

alice
david
subteam fetches
origin
subteam
fetches
subteam
fetches
bob
clair

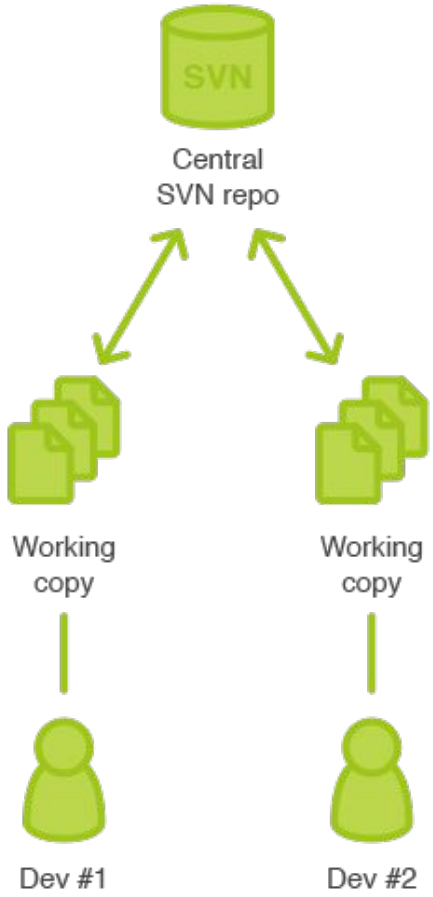
SUBVERSION

SVN

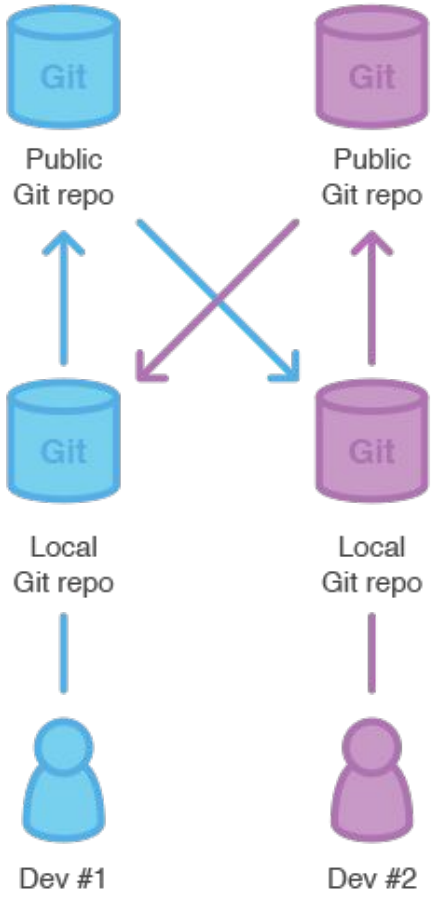

Network Connection
Central Repository
commit
commit
commit
commit
update
update
update
update


User 1
User 2
Subversion
User ... N
User 3

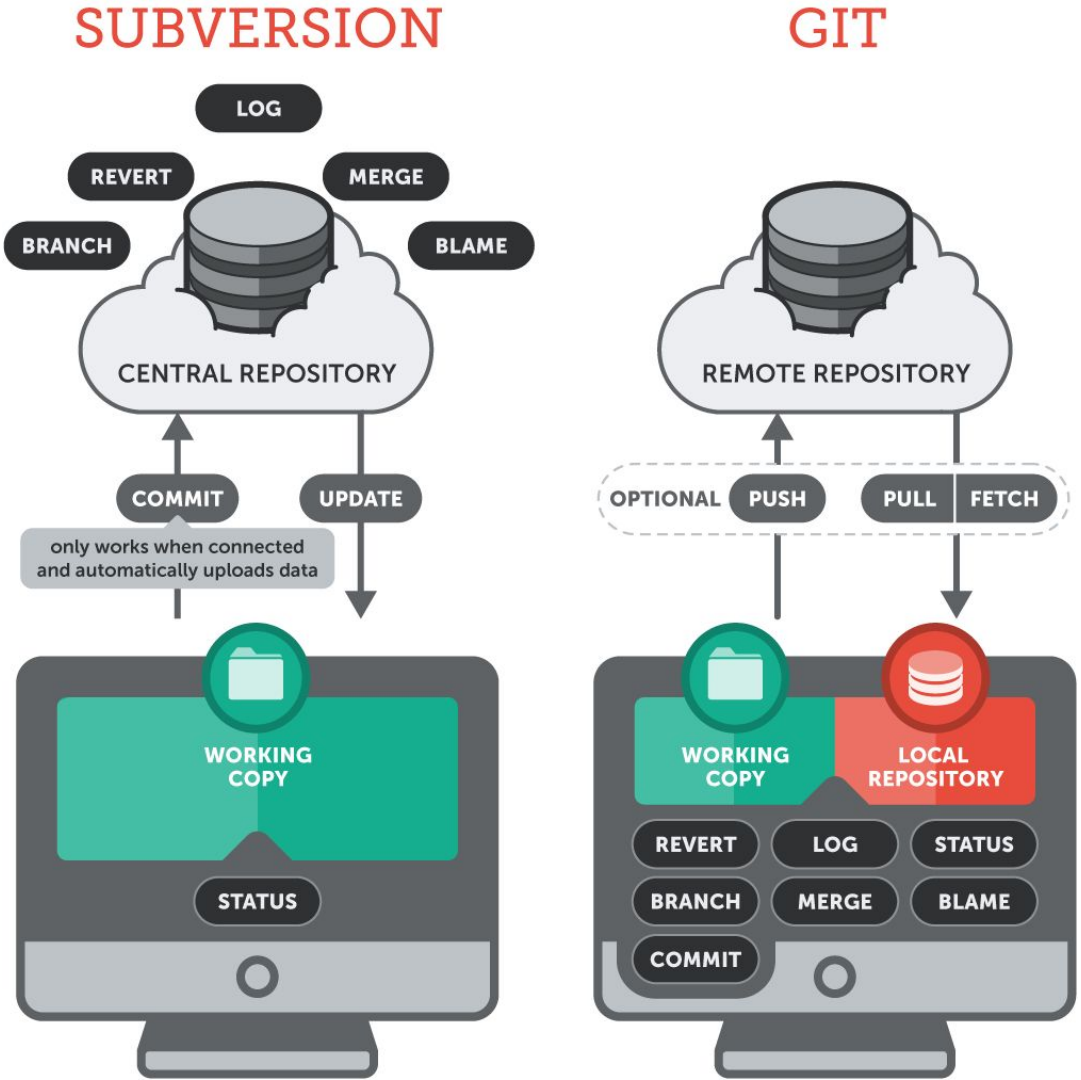

Server
File Repository
Terminology
Updates
Commits
Client 1
Local Copy
Client 2
Local Copy
Client N
Local Copy



Centralized
SVN development

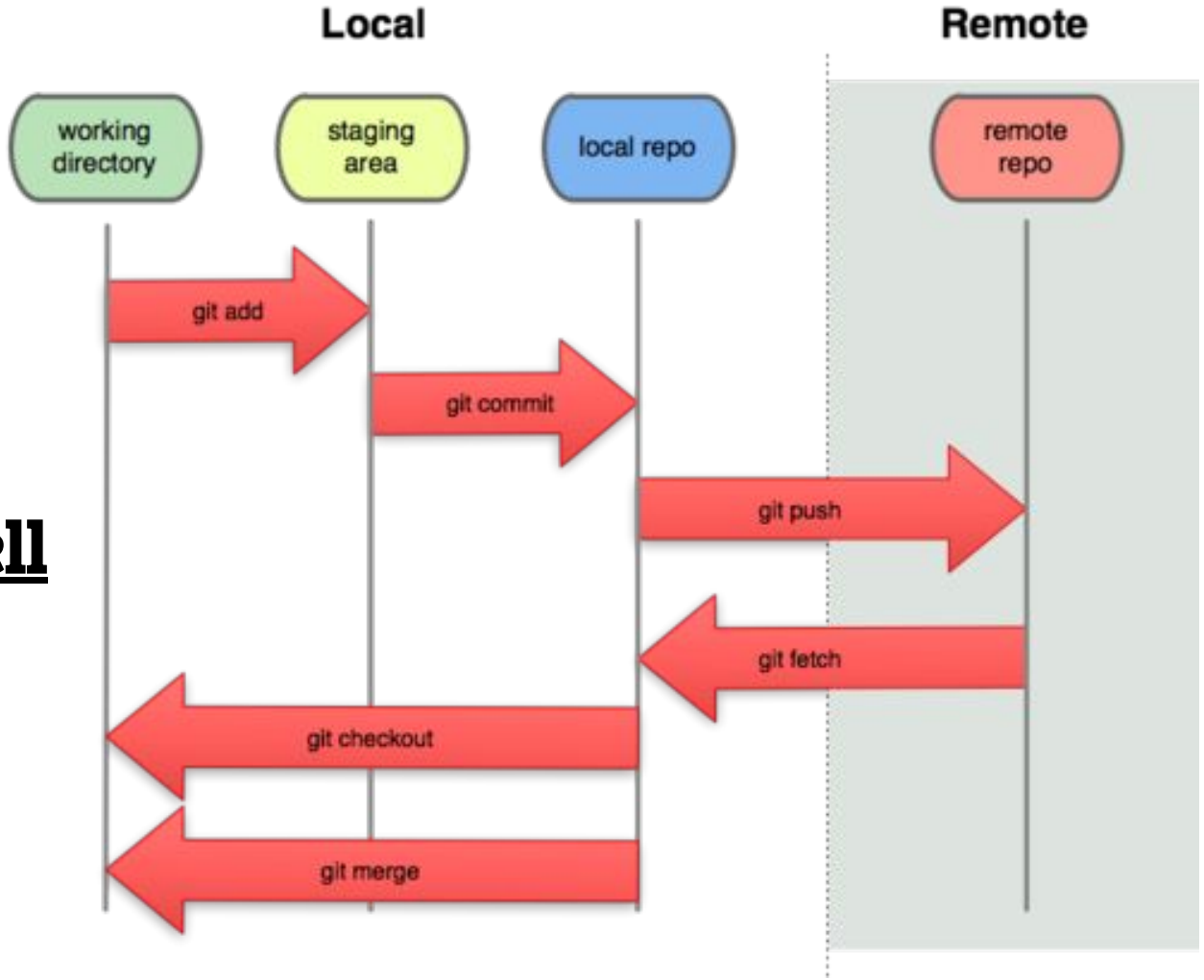


Distributed
Git development

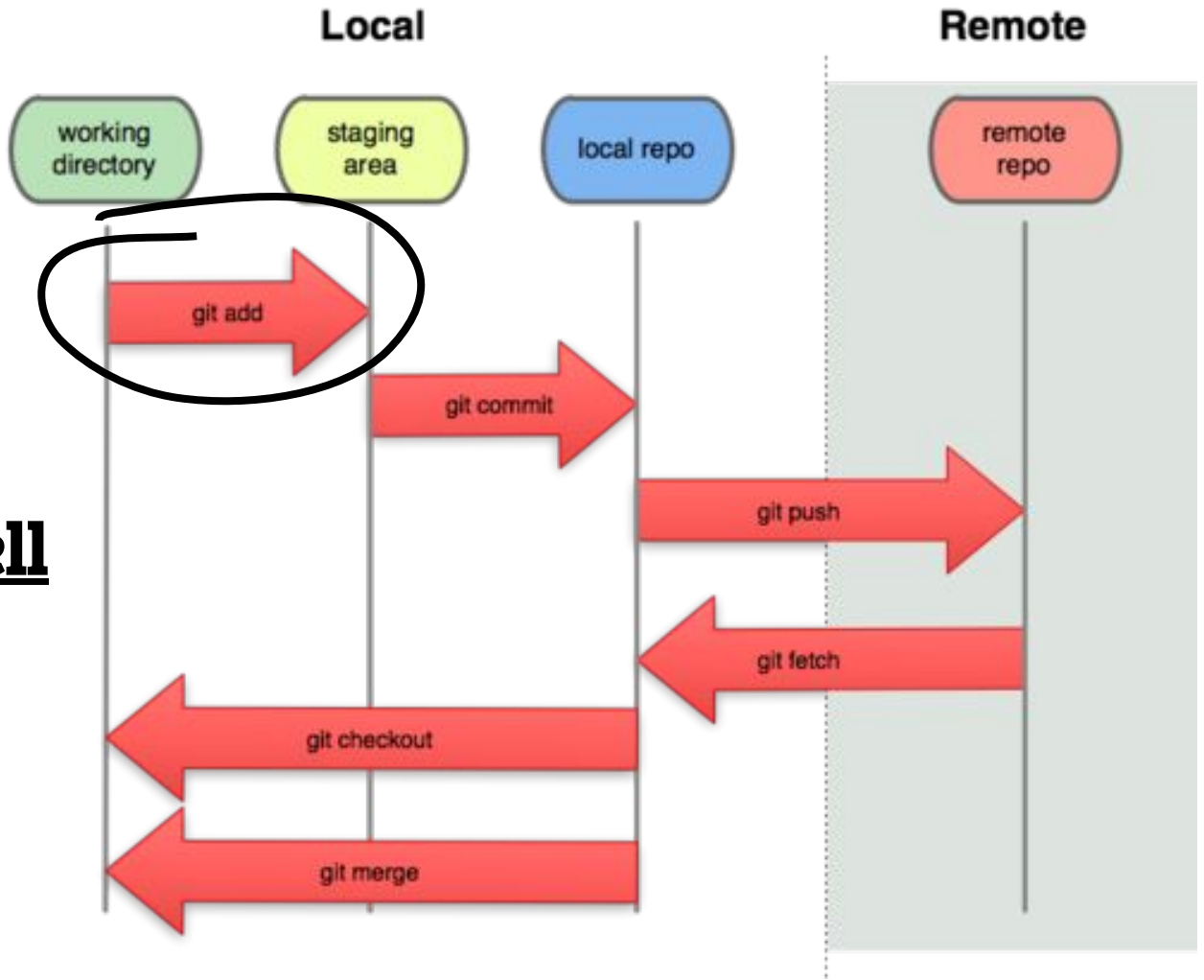

SUBVERSION
LOG
REVERT
MERGE
BRANCH
BLAME
CENTRAL REPOSITORY
COMMIT
UPDATE
only works when connected
and automatically uploads data
WORKING
COPY
STATUS
GIT
REMOTE REPOSITORY
OPTIONAL
PUSH
PULL
FETCH
WORKING
COPY
LOCAL
REPOSITORY
REVERT
LOG
STATUS
BRANCH
MERGE
BLAME
COMMIT

git

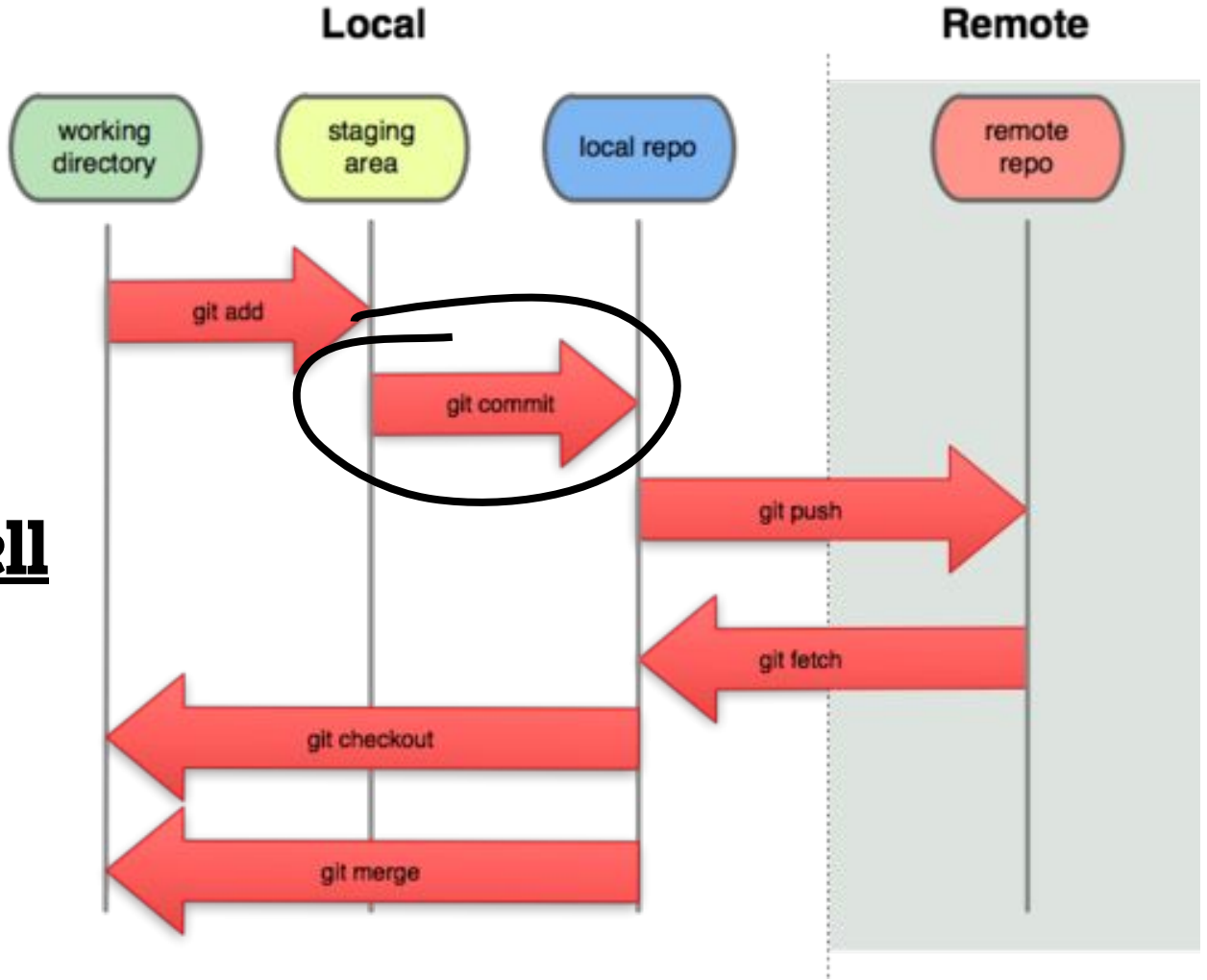
GIT
in a nutshell



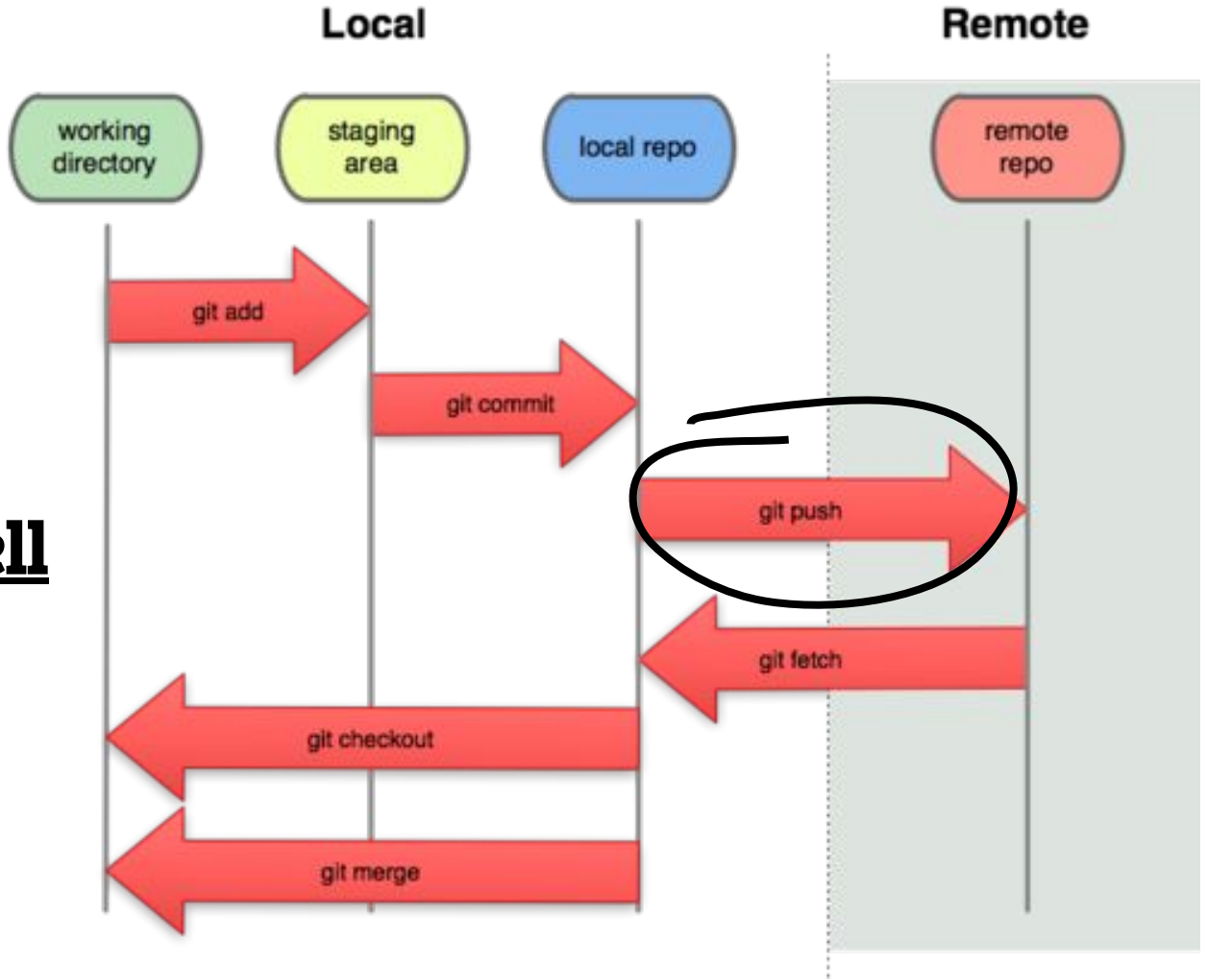
GIT
In a nutshell



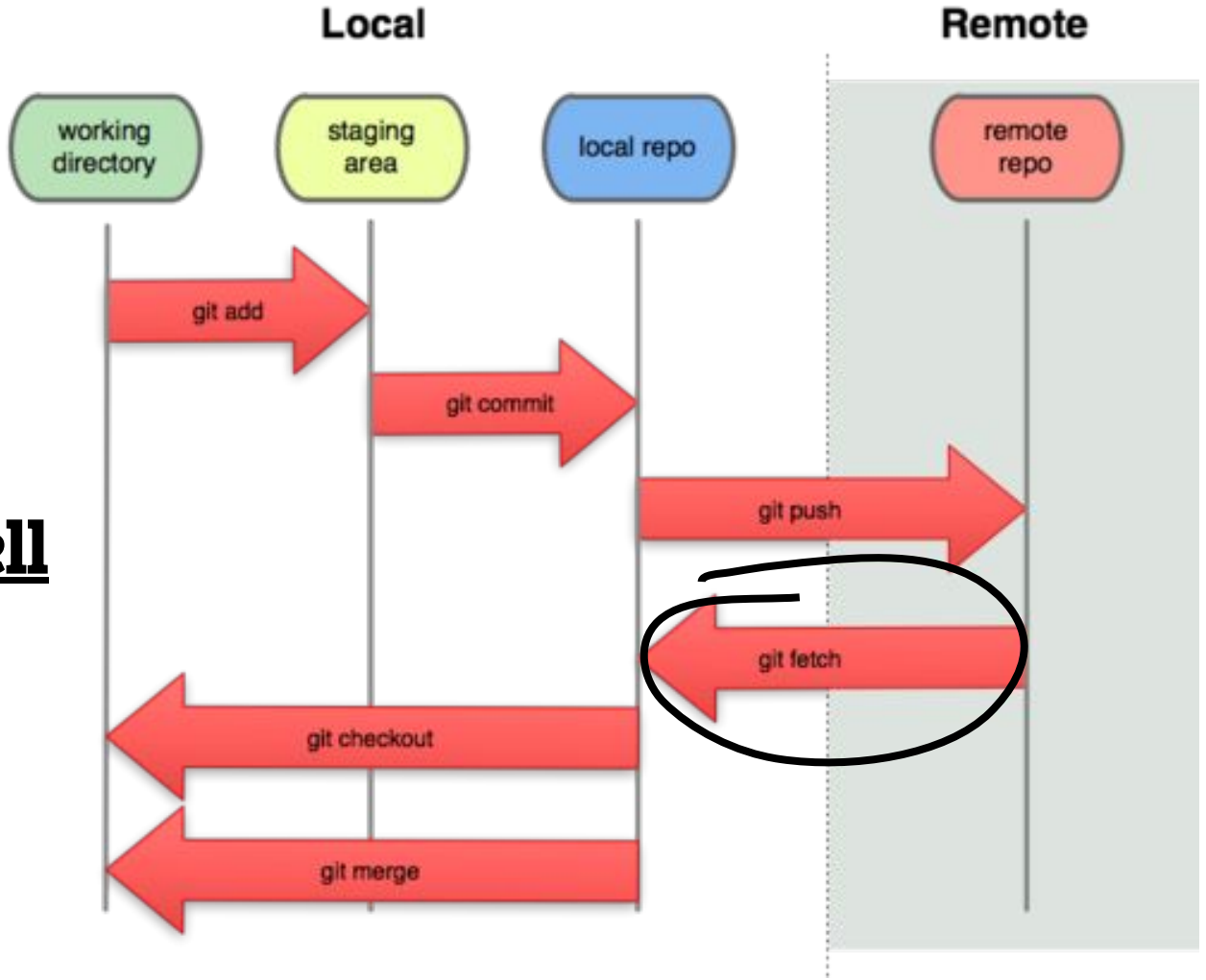
GIT
In a nutshell



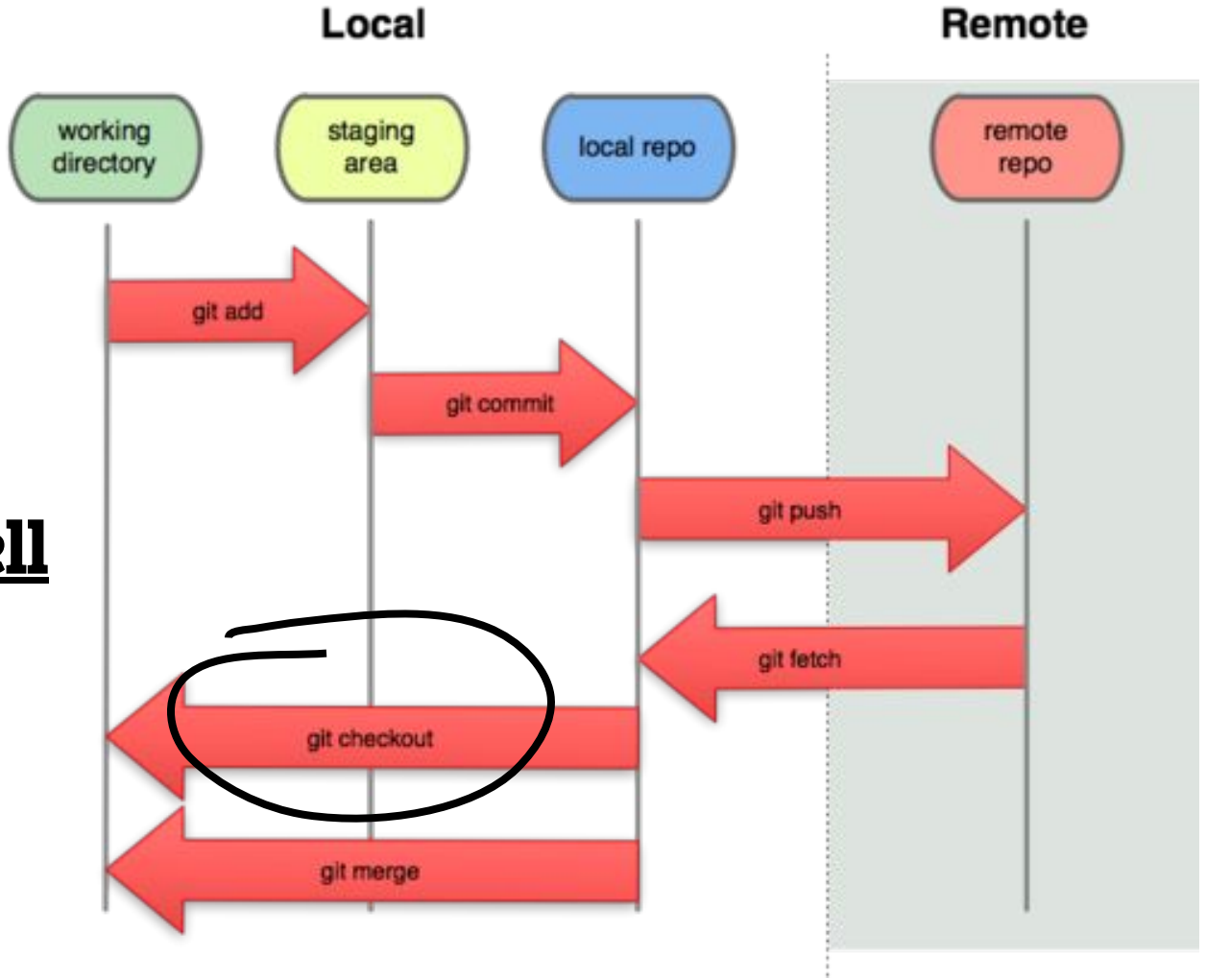
GIT
In a nutshell



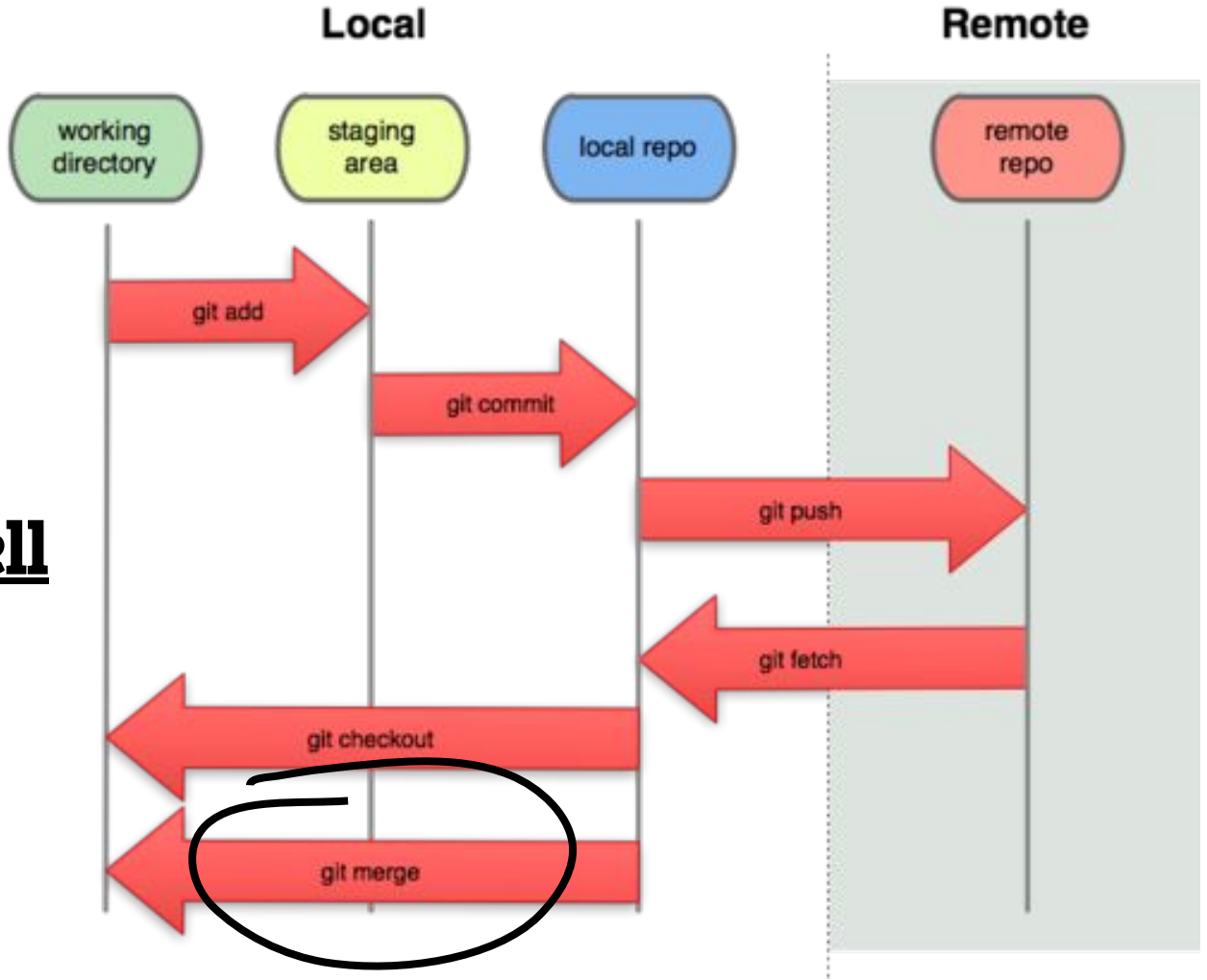
GIT
In a nutshell



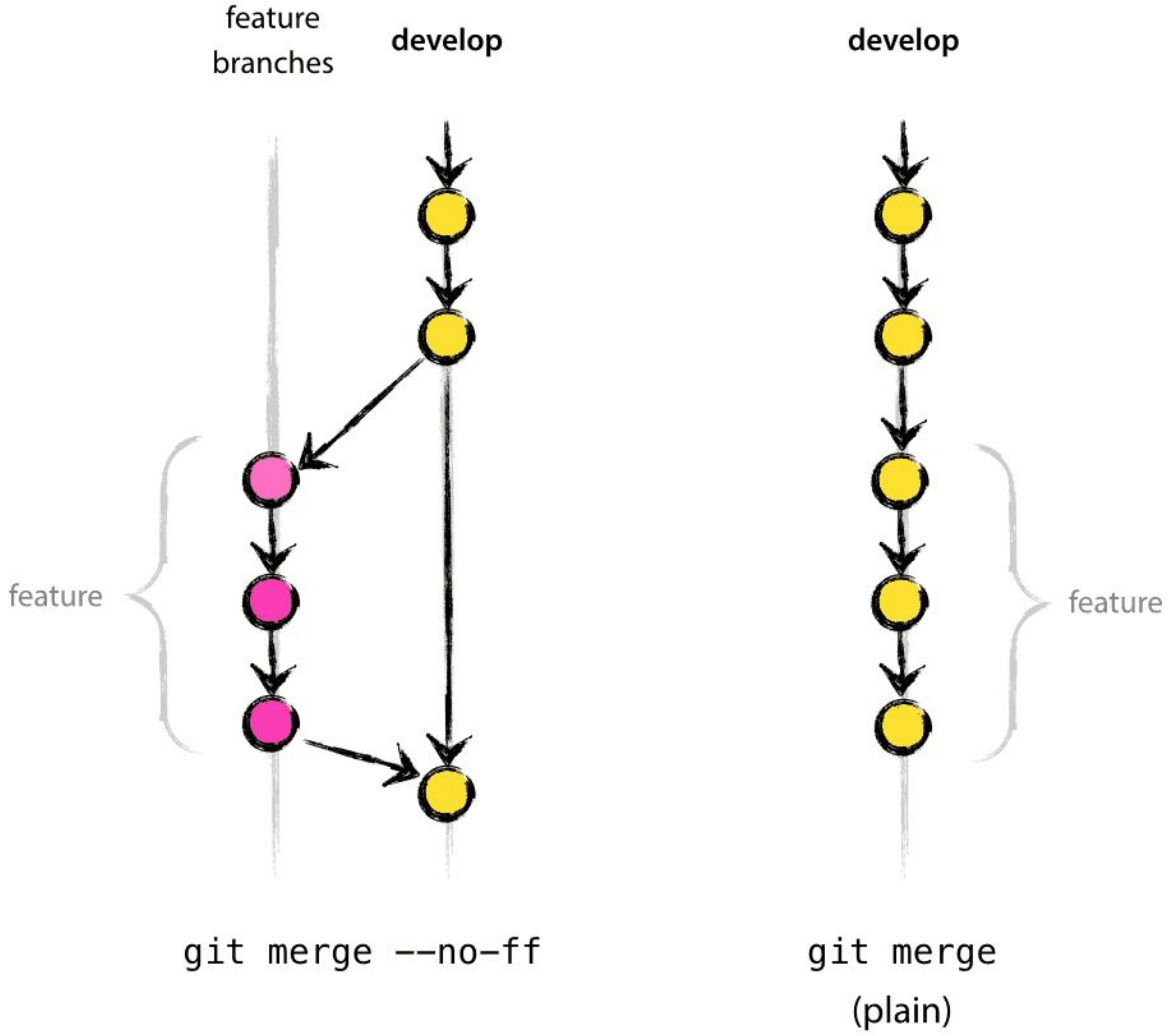
GIT
In a nutshell

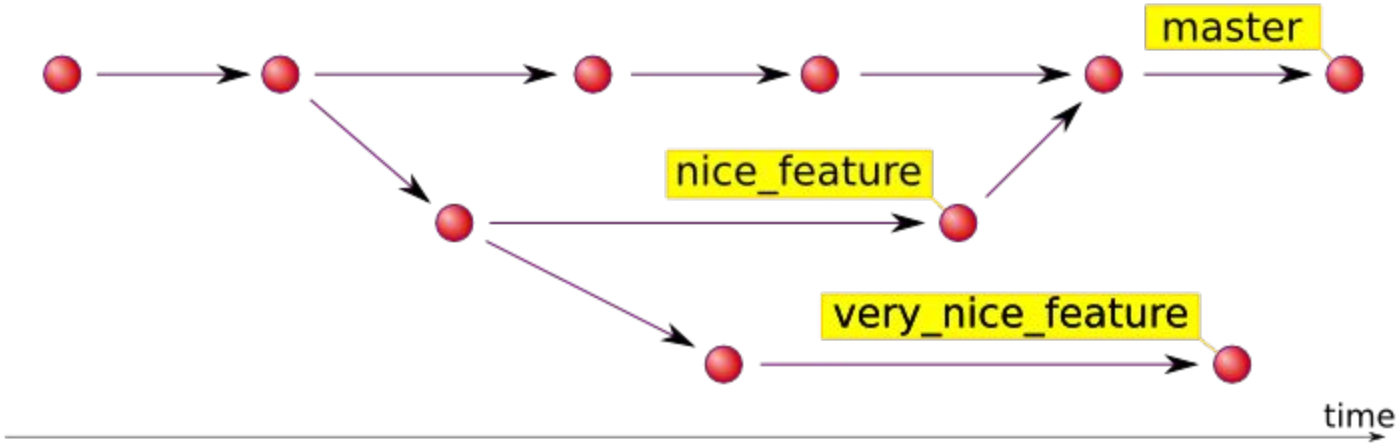


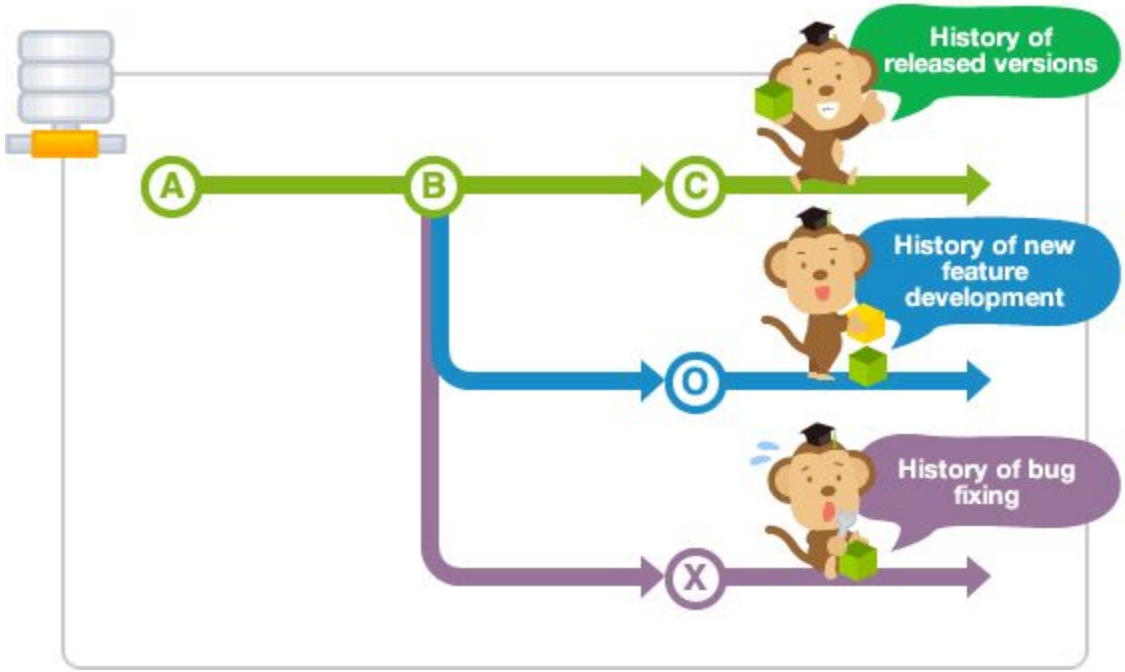
GIT
In a nutshell



Branches


feature
branches
develop
feature
git merge --no-ff
develop
feature
git merge
(plain)


master
nice_feature
very_nice_feature
time


History of released versions
A
B
C
History of new feature development
O
History of bug fixing
X

GitHub

Atlassian
Bitbucket

Atlassian
SourceTree

LIVE DEMO