What is git?

How does it work?

How to use it?

Hands-on use case.

REU lunch talk
Monday, June 25th 2018

Christophe Hauser
hauser@isi.edu

What is git?

- Created by Linus Torvalds in 2005.
- Version control for the Linux kernel.
- “Take Concurrent Versions System (CVS) as an example of what not to do; if in doubt, make the exact opposite decision.”

Design

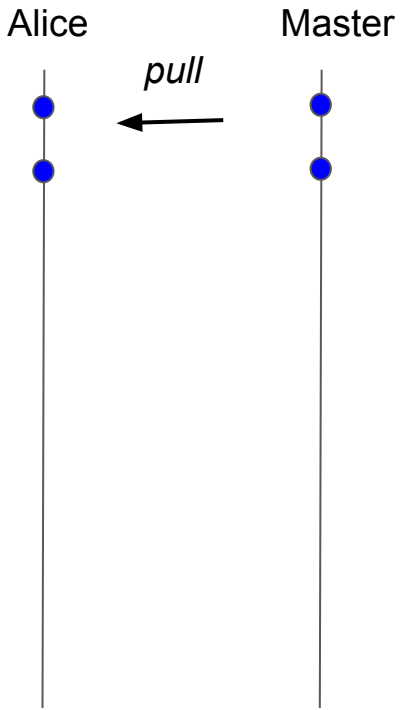
- Very fast and scalable (an order of magnitude faster than some other systems).
- Distributed (each repository contains the entire history).
- Include very strong safeguards against corruption, either accidental or malicious.

Design

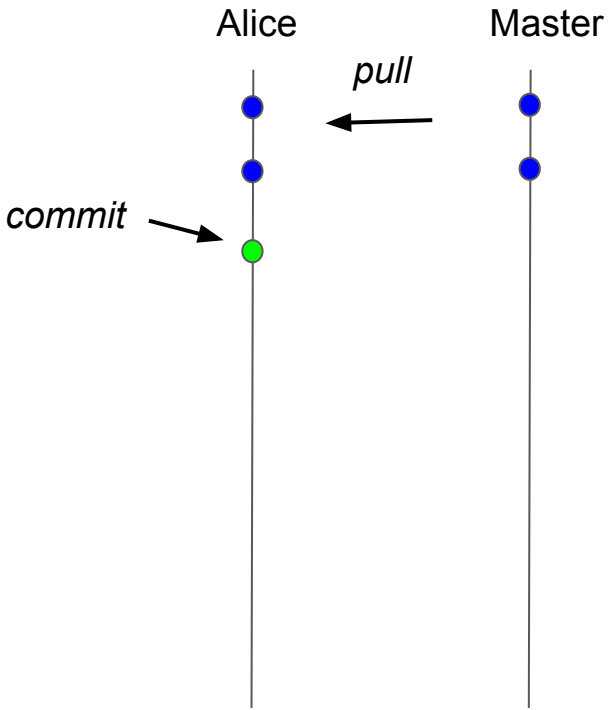
- Everything is SHA-1 protected.
- If anything changes, git will know about it.

How does it work?

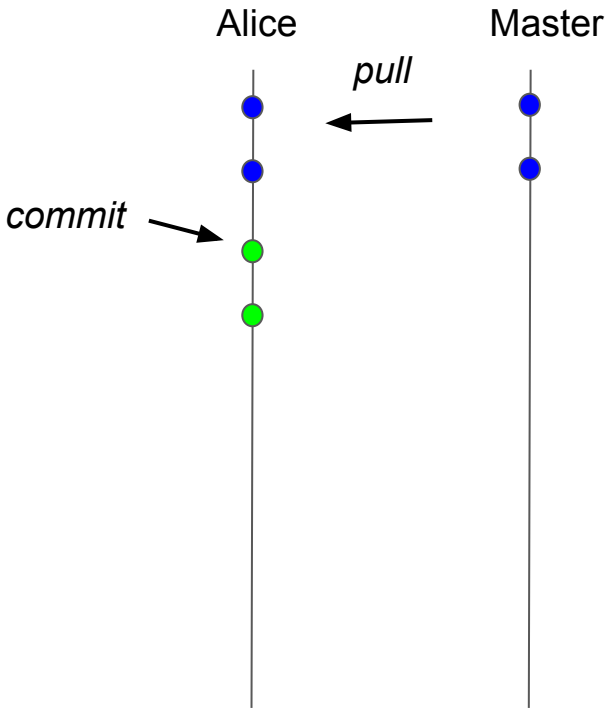
Workflow



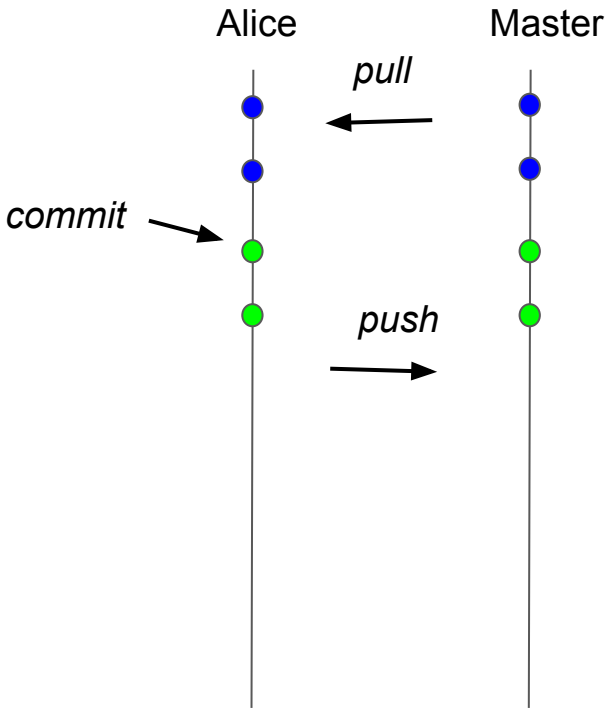
Workflow



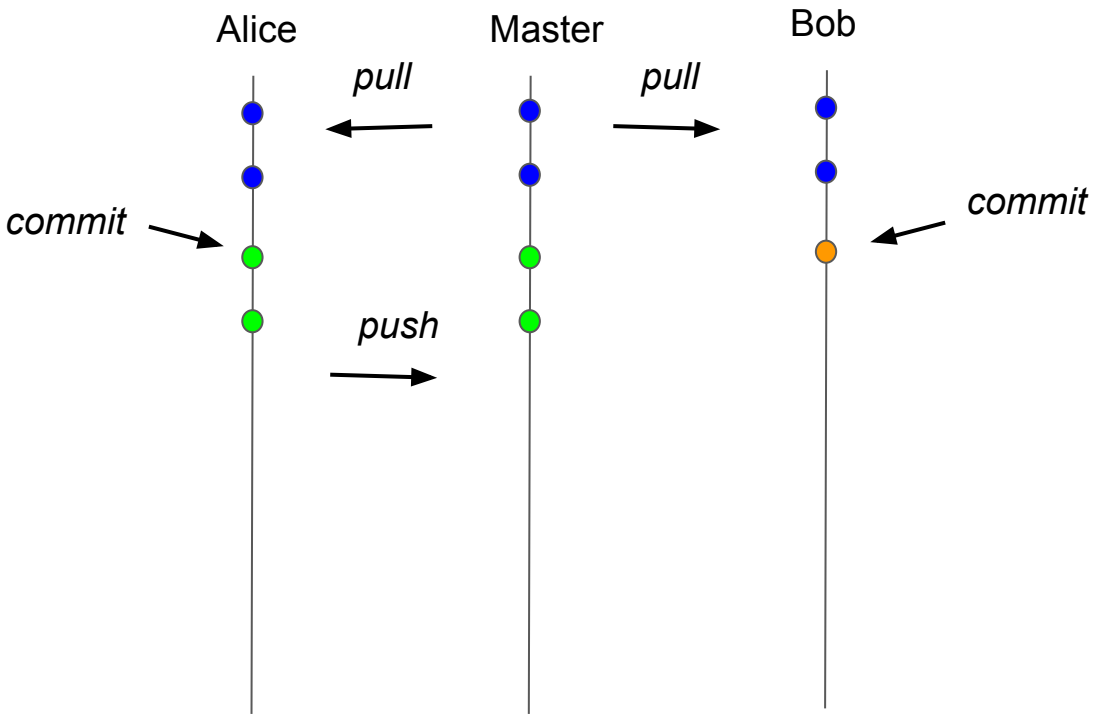
Workflow



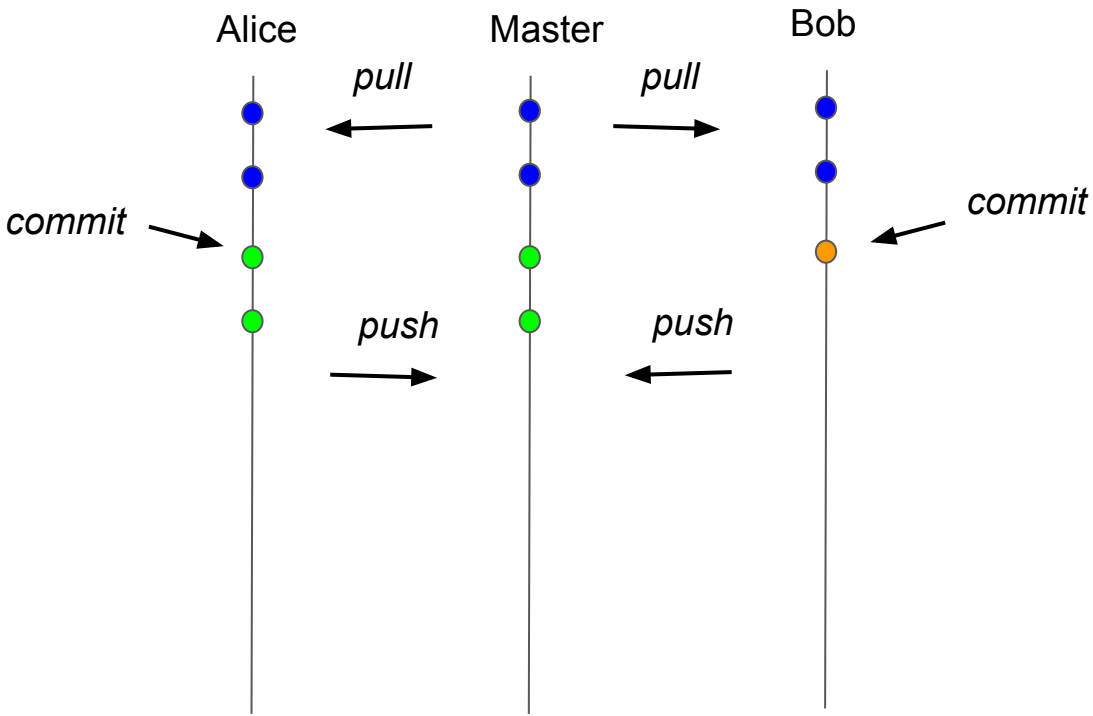
Workflow



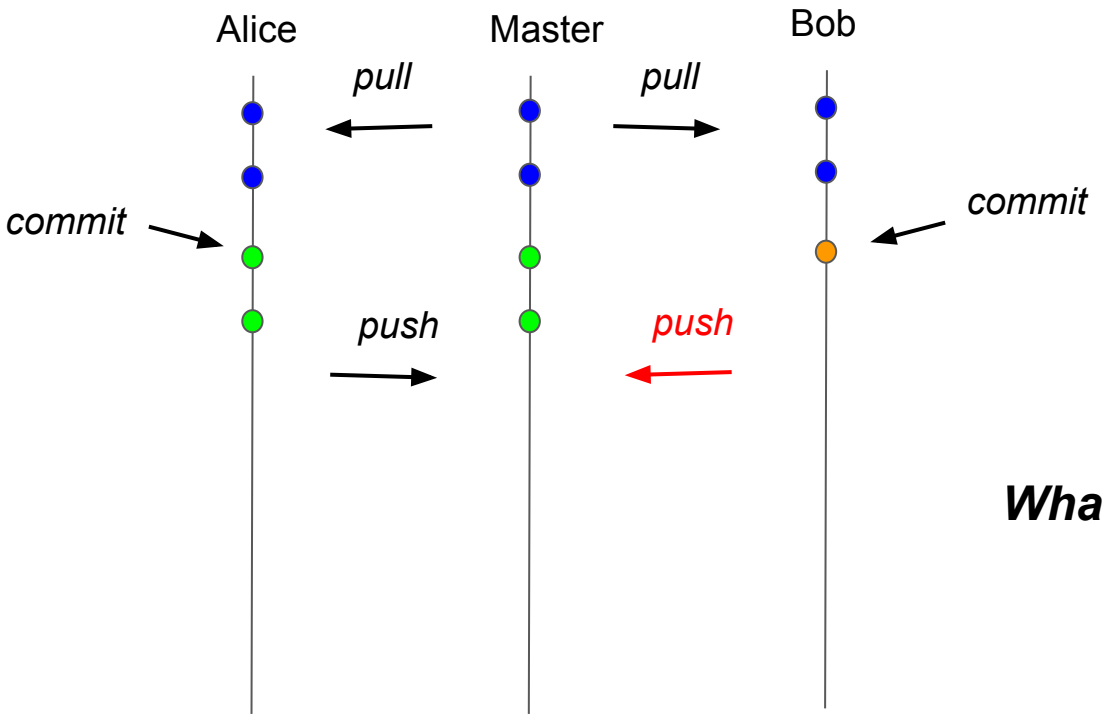
Workflow



Workflow

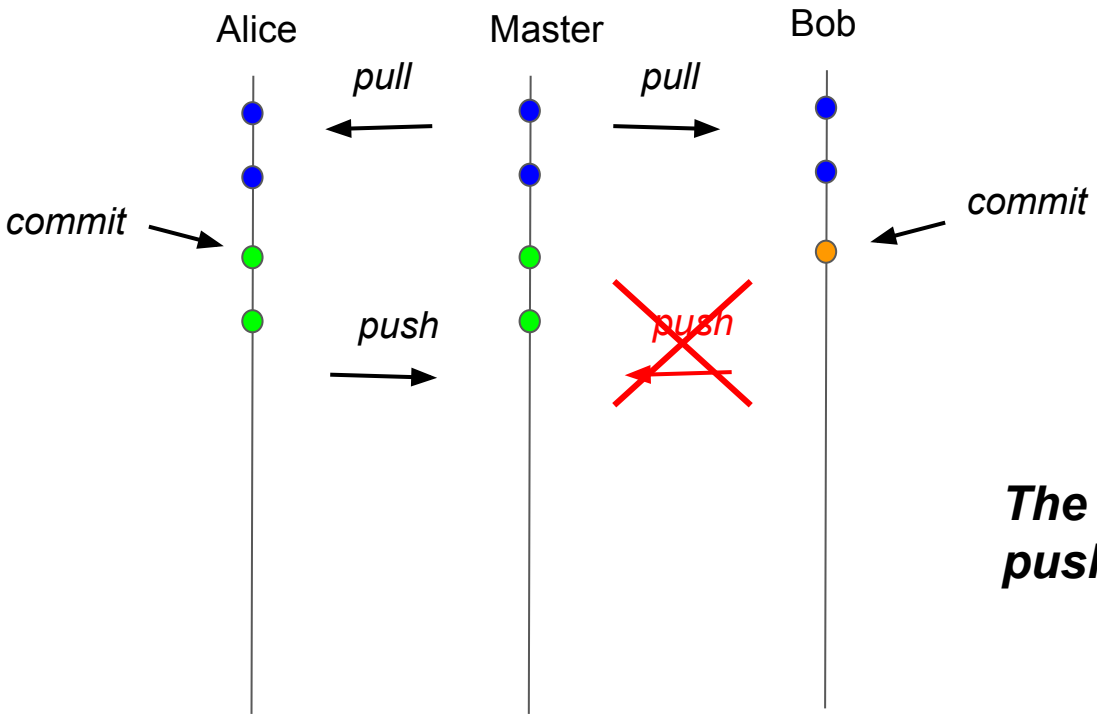


Workflow



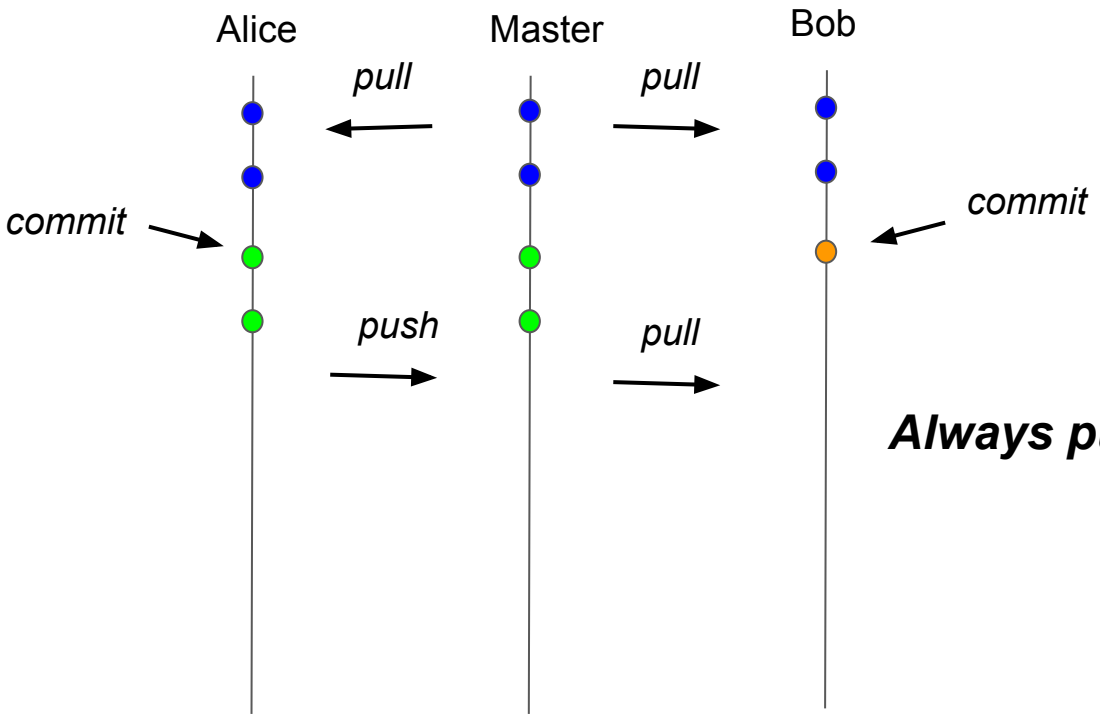
What happens?

Workflow



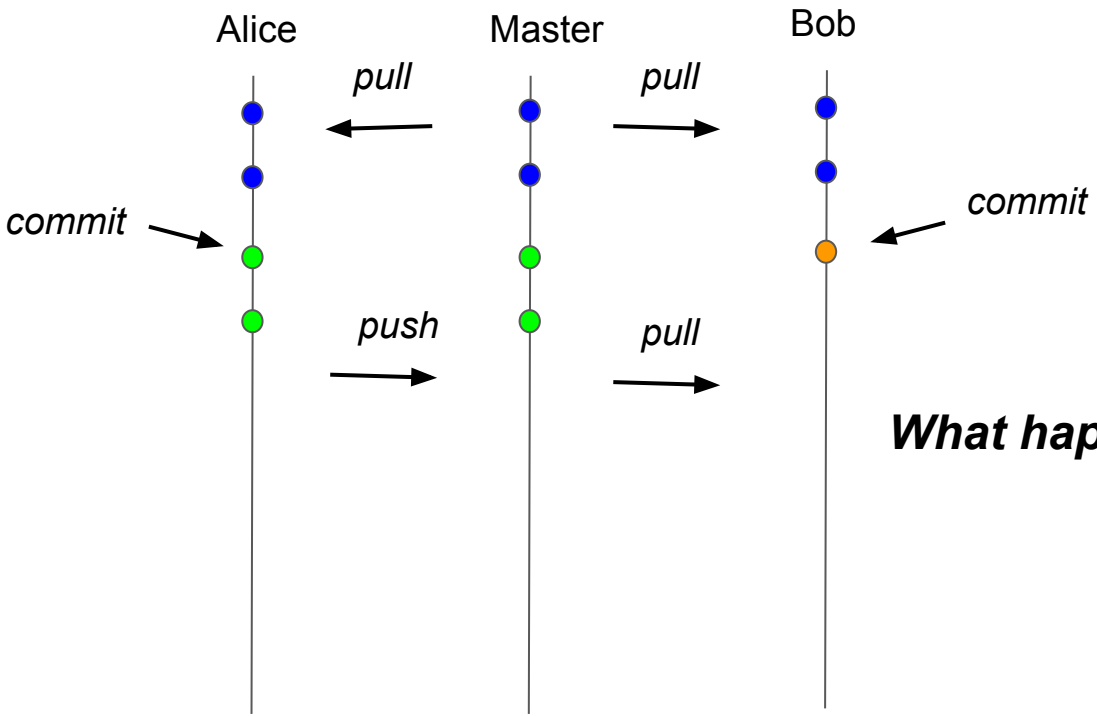
The server refuses the push request.

Workflow

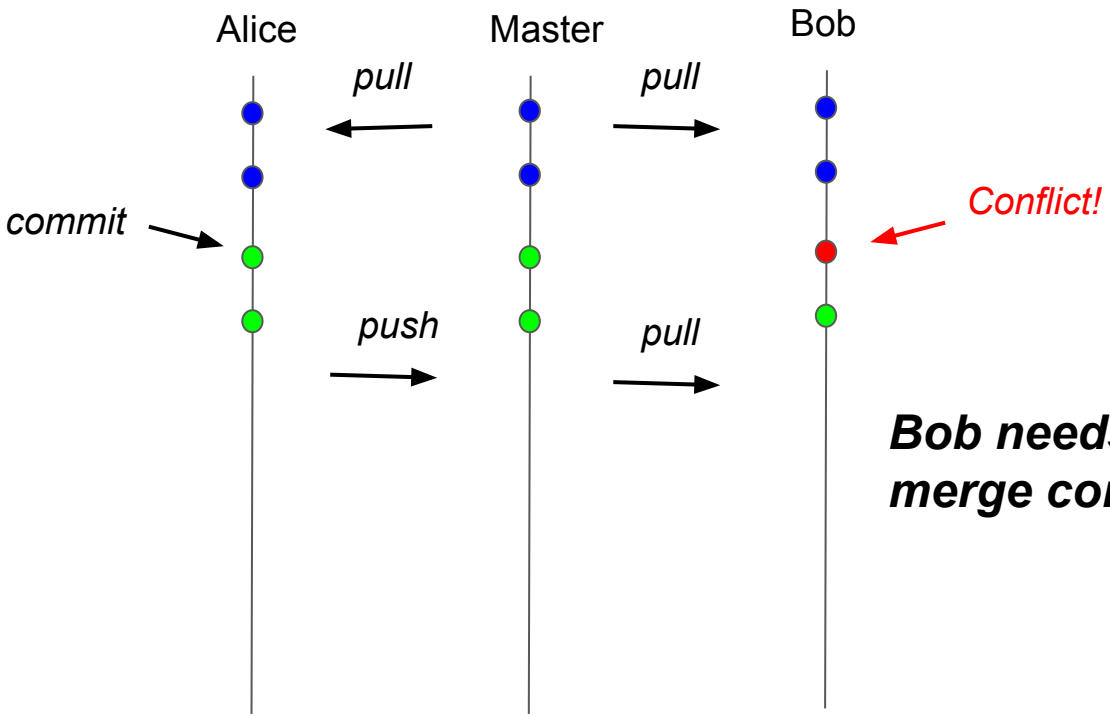


Always pull before you push!

Workflow

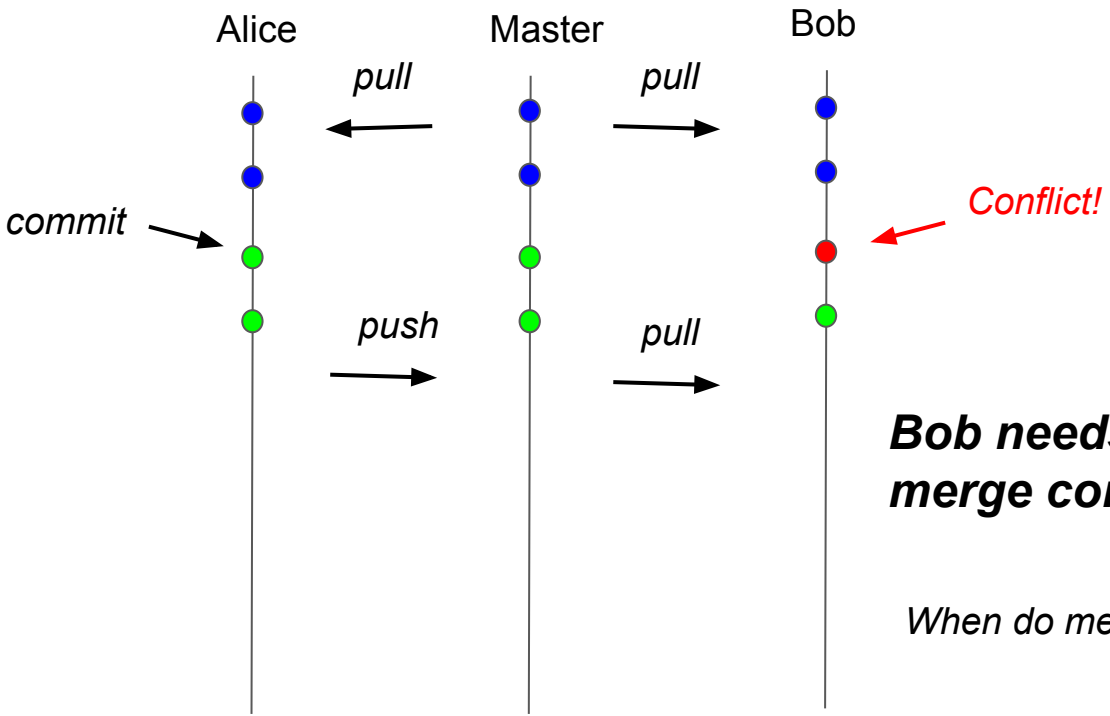

Alice
Master
Bob
pull
pull
commit
commit
push
pull
What happens now?

Workflow



Bob needs to resolve the merge conflict.

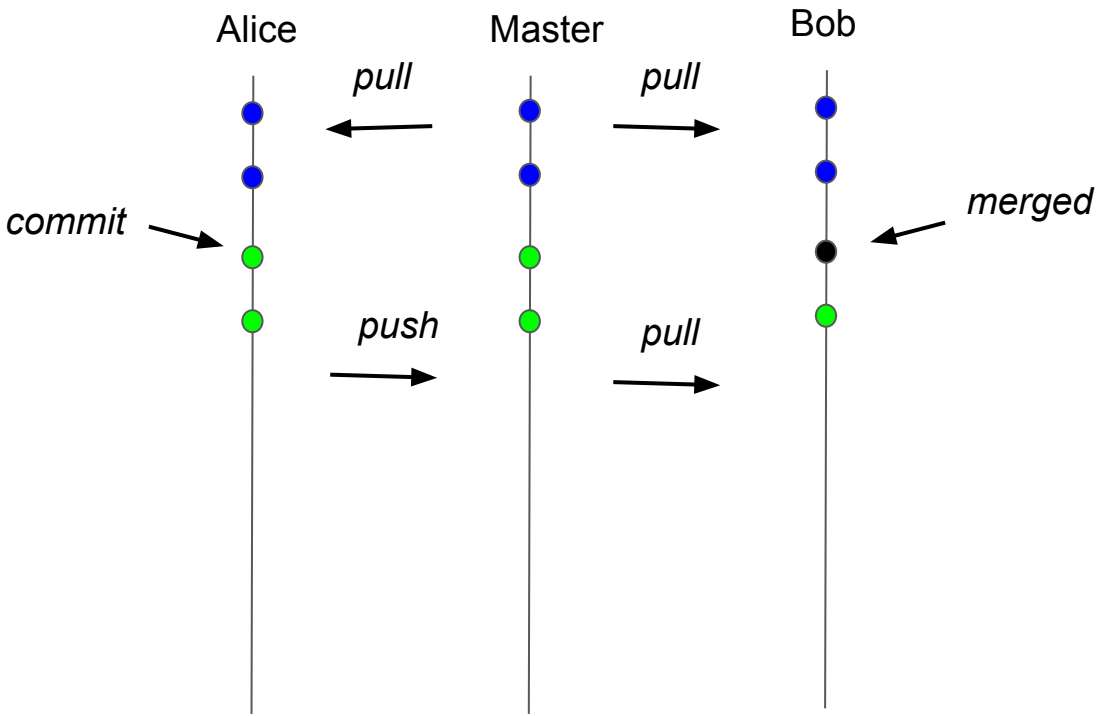
Workflow



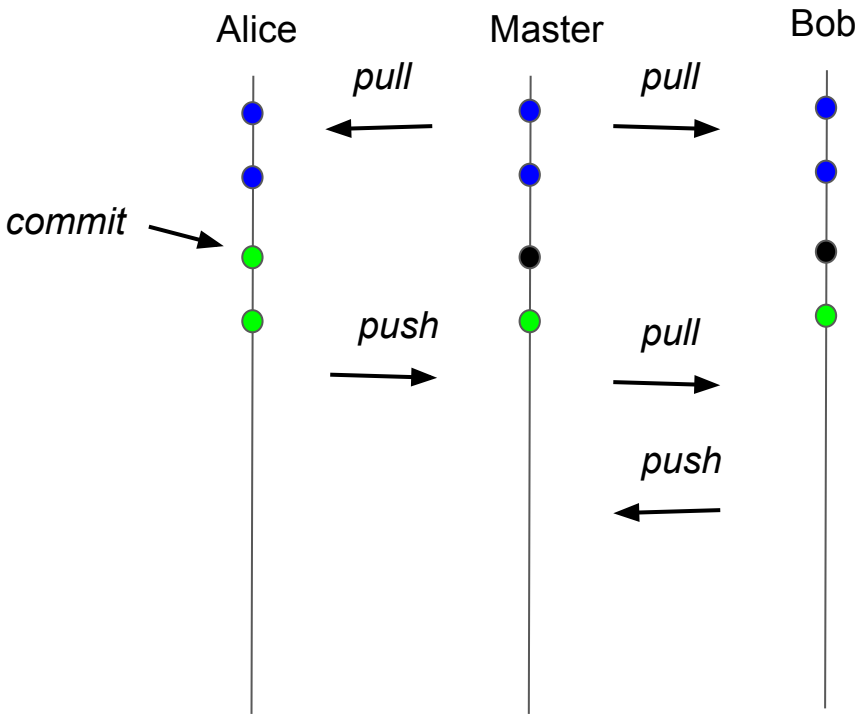
Bob needs to resolve the merge conflict.

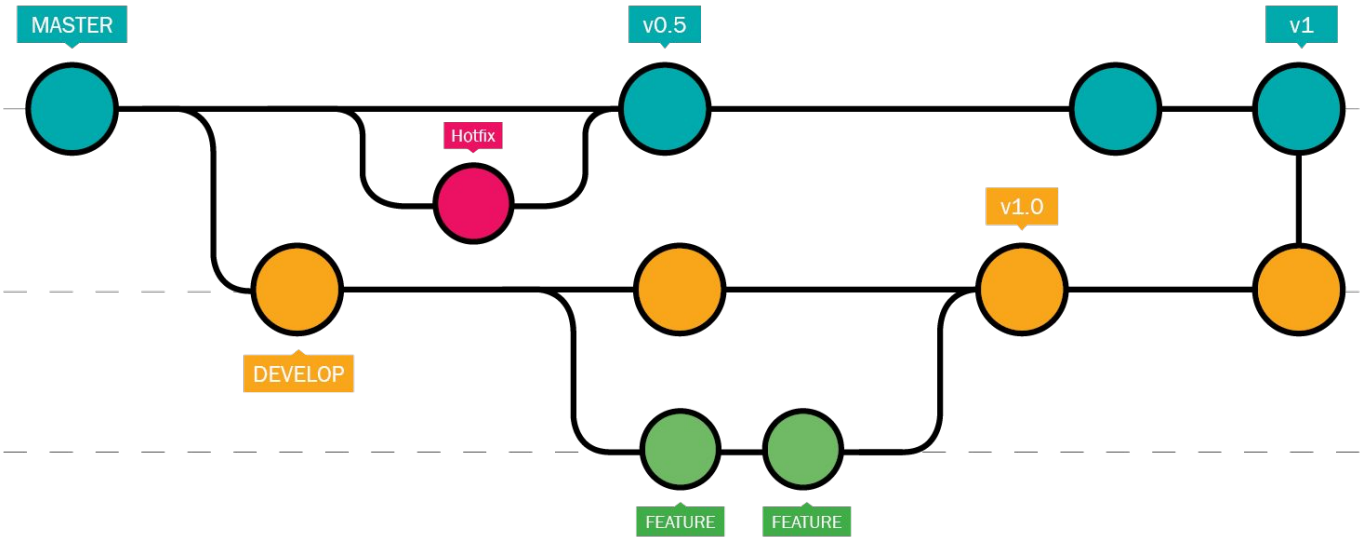
When do merge conflicts occur?

Workflow



Workflow




MASTER
v0.5
v1
Hotfix
v1.0
DEVELOP
FEATURE
FEATURE

How to use it?

Set up your identity

Edit ~/.gitconfig

```
[user]
›    email = hauser@isi.edu
›    name = Christophe Hauser
```

New repo

```
git init

git add <files>
    e.g., git add *

git commit <files>
    e.g., git commit -a
```

Understanding what changed

What did I modify since last commit?

```
git diff
```

Am I tracking all files? Are there deleted/modified files?

```
git status
```

What is the history of changes?

```
git log
```

What changed since two commits ago?

```
git diff HEAD~2
```

Branches

New branch

```
git checkout -b <branch name>
```

Switch to (“checkout”) another branch

```
git checkout <branch name>
```

What branches are there?/What branch am I on?

```
git branch
```

Merge another branch into local branch

```
git merge <branch name>
```

Merge conflicts

Understanding what changed

- Use colors!

git diff --colors

Edit ~/.gitconfig

```
[color]
›       branch = auto
›       diff = auto
›       interactive = auto
›       status = auto
```

Understanding what changed

git log -p
git blame
git diff --color-words | fold
git diff --word-diff

Hands-on use case.

Let's try it out.

```
git clone
https://git.overleaf.com/17341195ytzbqfkdyvjh

git log
git blame main.tex
```

Let's try it out.

1. Fix just one spelling mistake and `commit/pull/push`

2. A. Create your own local branch off the original commit.
   `git checkout -b "my_branch"`
   B. Fix all spelling mistake and commit.
   C. checkout the master branch, pull and merge your branch into it.

3. Reset the master branch to the original commit.
4. Fix one grammar mistake or rephrase one bit of a sentence and
   `commit/pull/push`