

# Git

Guide pratique de démarrage (presque) en douceur

soon



# Planning

- 1 Rapide introduction
- 2 Quel intérêt ?
- 3 Workflows types
- 4 Processus SOON
- 5 Questions ?

pronconciation : [gít]

“guite”

système de gestion du *Kernel Linux*

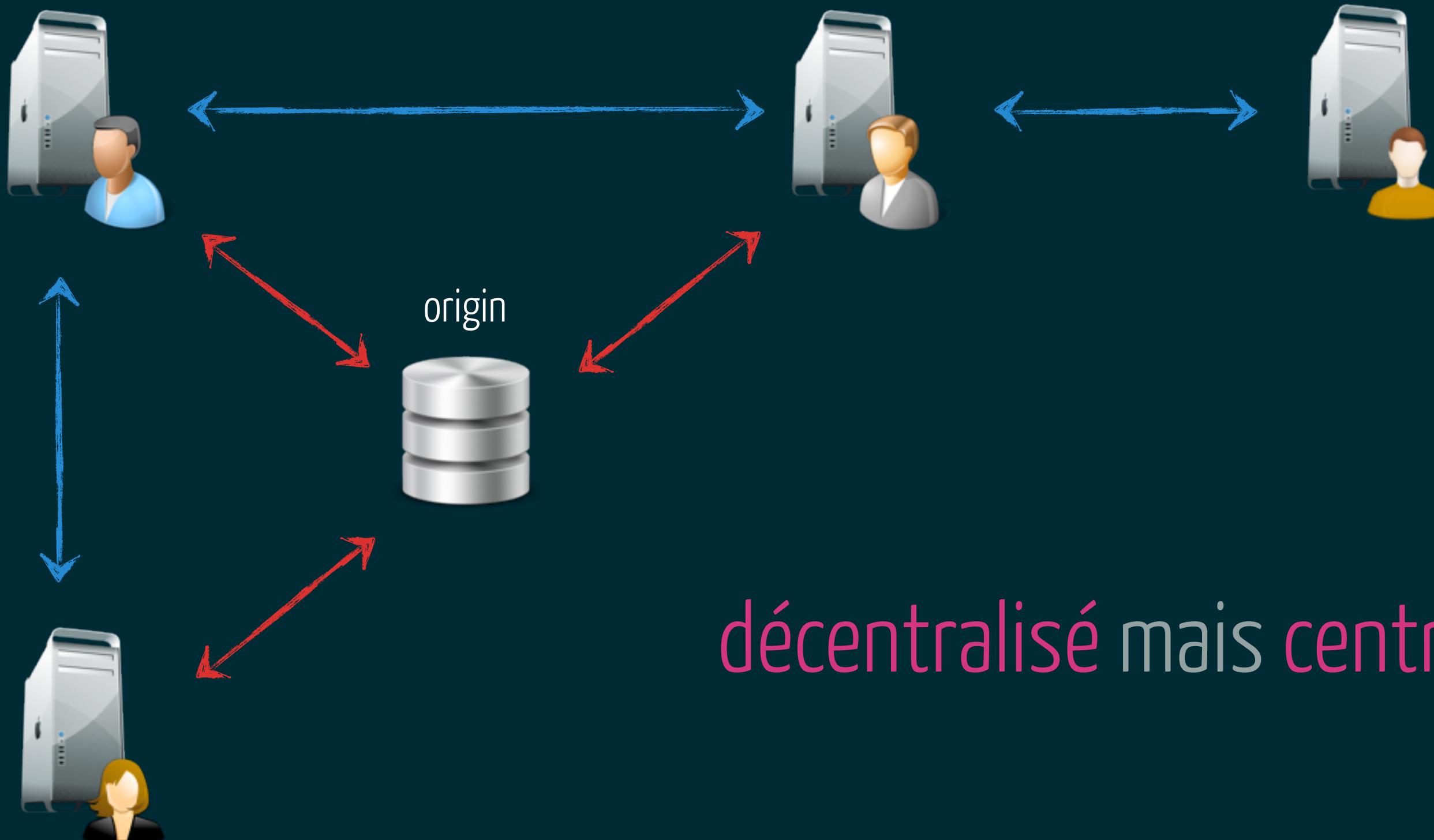
créé par Linus Torvalds



Des utilisateurs de SVN ?

master != trunk

branch, branch, branch.  
merge, merge, merge. tag.



décentralisé mais centralisé

git is forgiving

git is fast

git is easy

git works on your OS

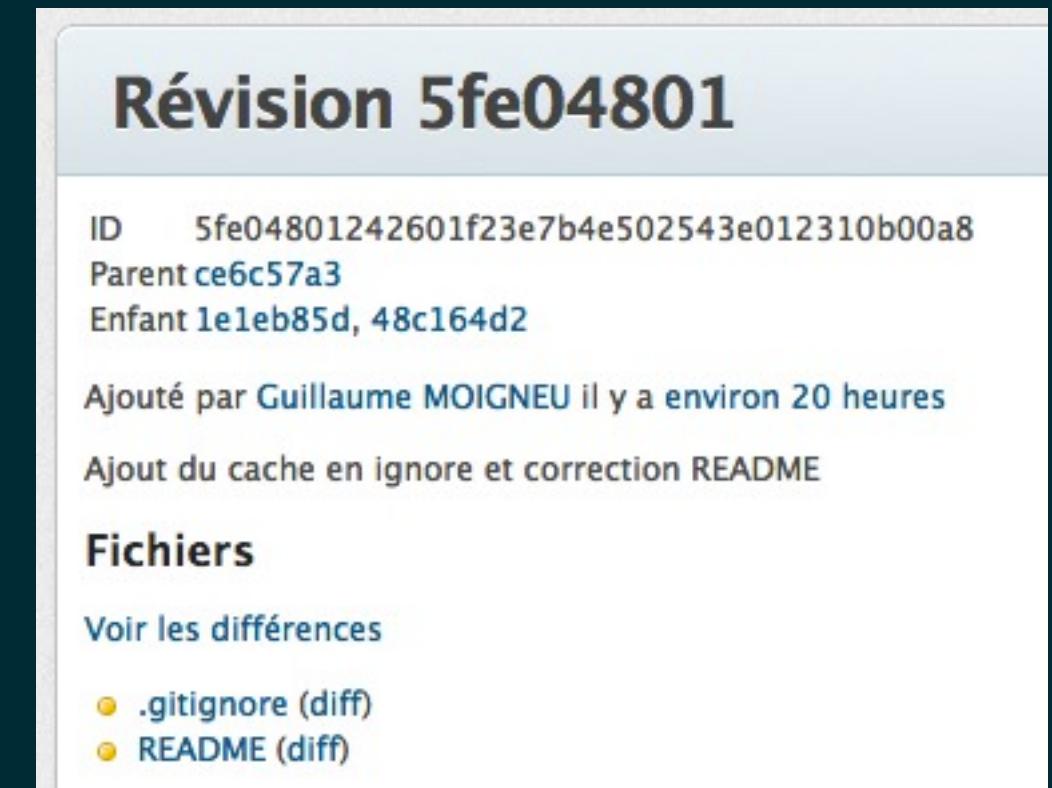
Pourquoi Git ?

git ❤ branches

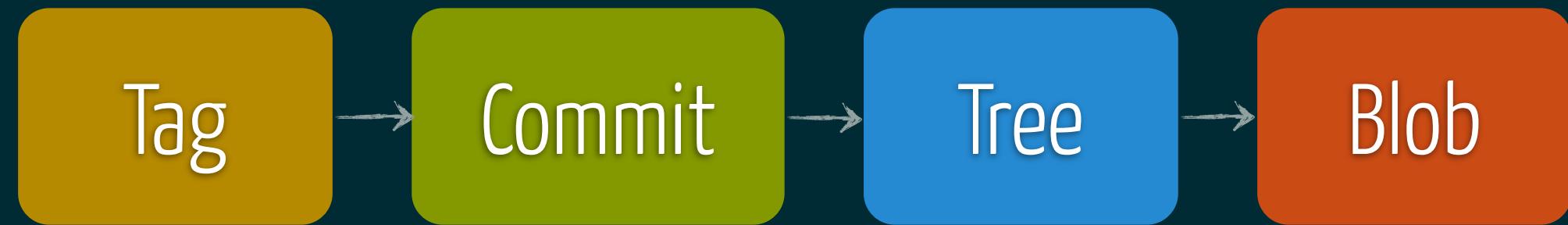
<http://whygitisbetterthanx.com>

# commit ?

- lot de modifications
- identifié par un hashtag unique
- référence d'un arbre de blob
- relations entre les commits



tag ?  
alias de commit

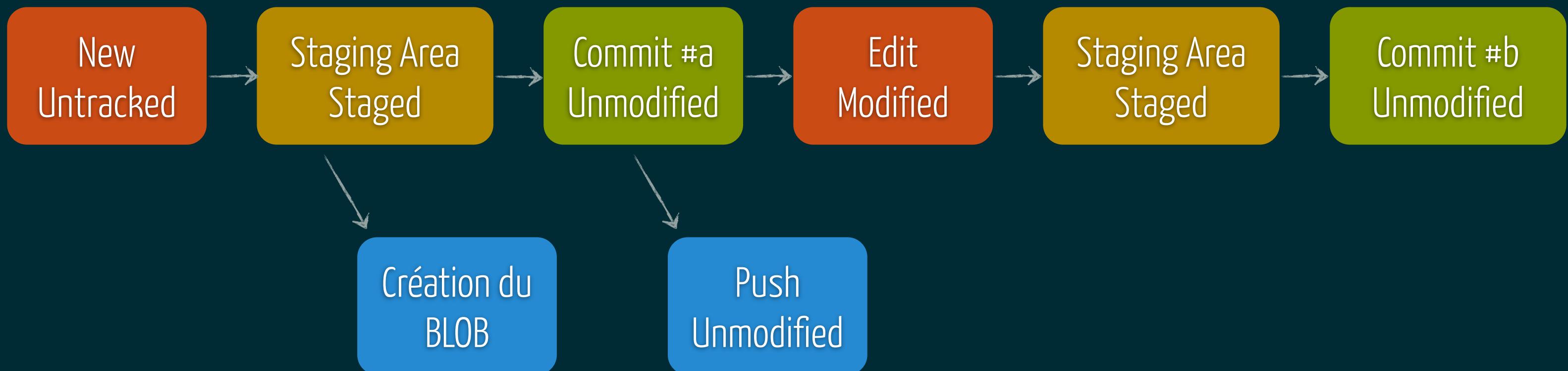


# Workflow de base Git

```
$ git clone git@git.agence-soon.fr:/repo-test.git
$ vim README
$ git add .
$ git commit -m "Ajout du README"
$ git push
```

```
$ vim README
$ git commit -m "Modification du README"
$ git push
```

# Cycle de vie d'un fichier



# Diff

git status

git diff (Compares working directory with staging area)

git diff --staged

git diff HEAD

git diff v1.0 v1.1

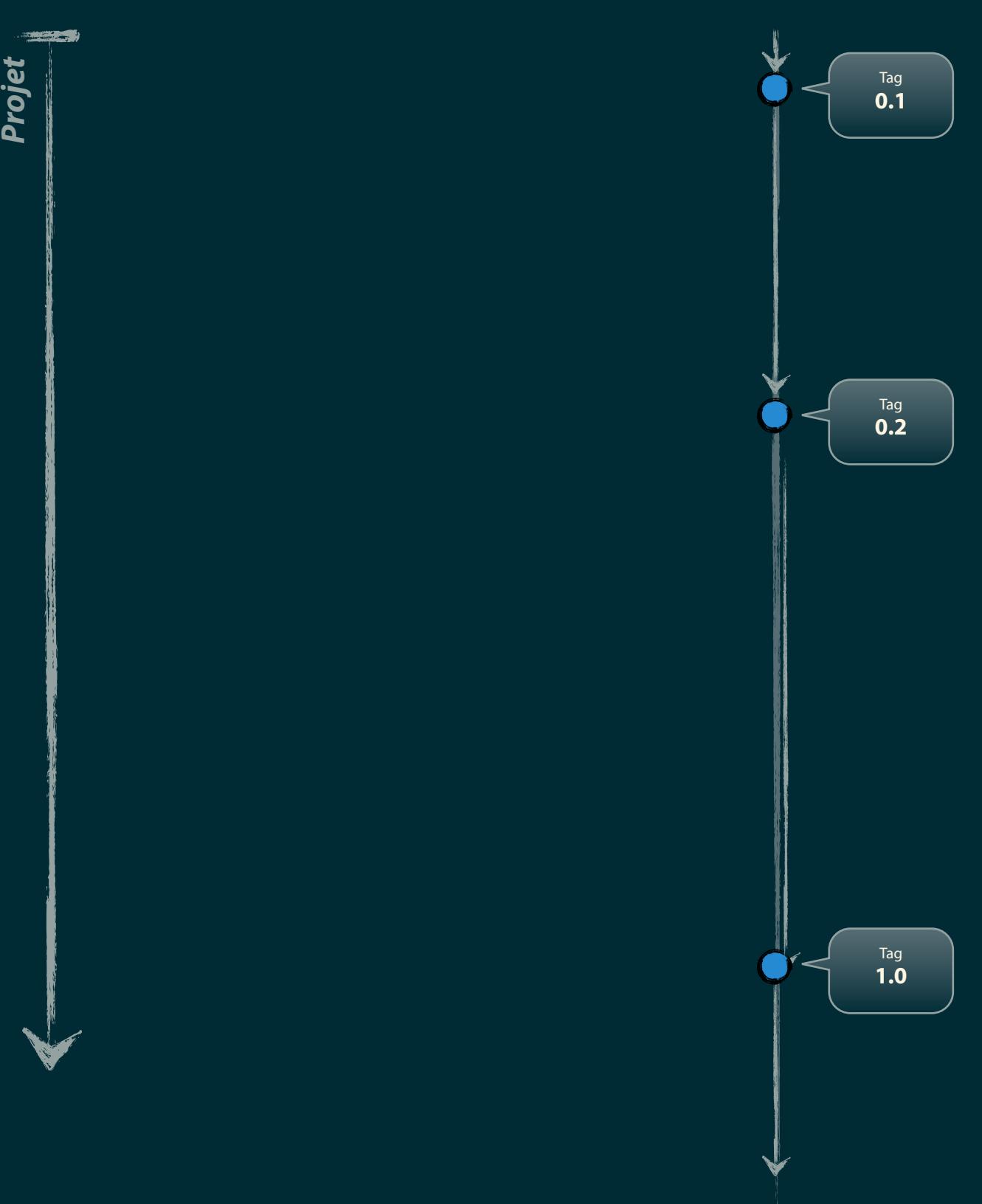
git diff master feature-1

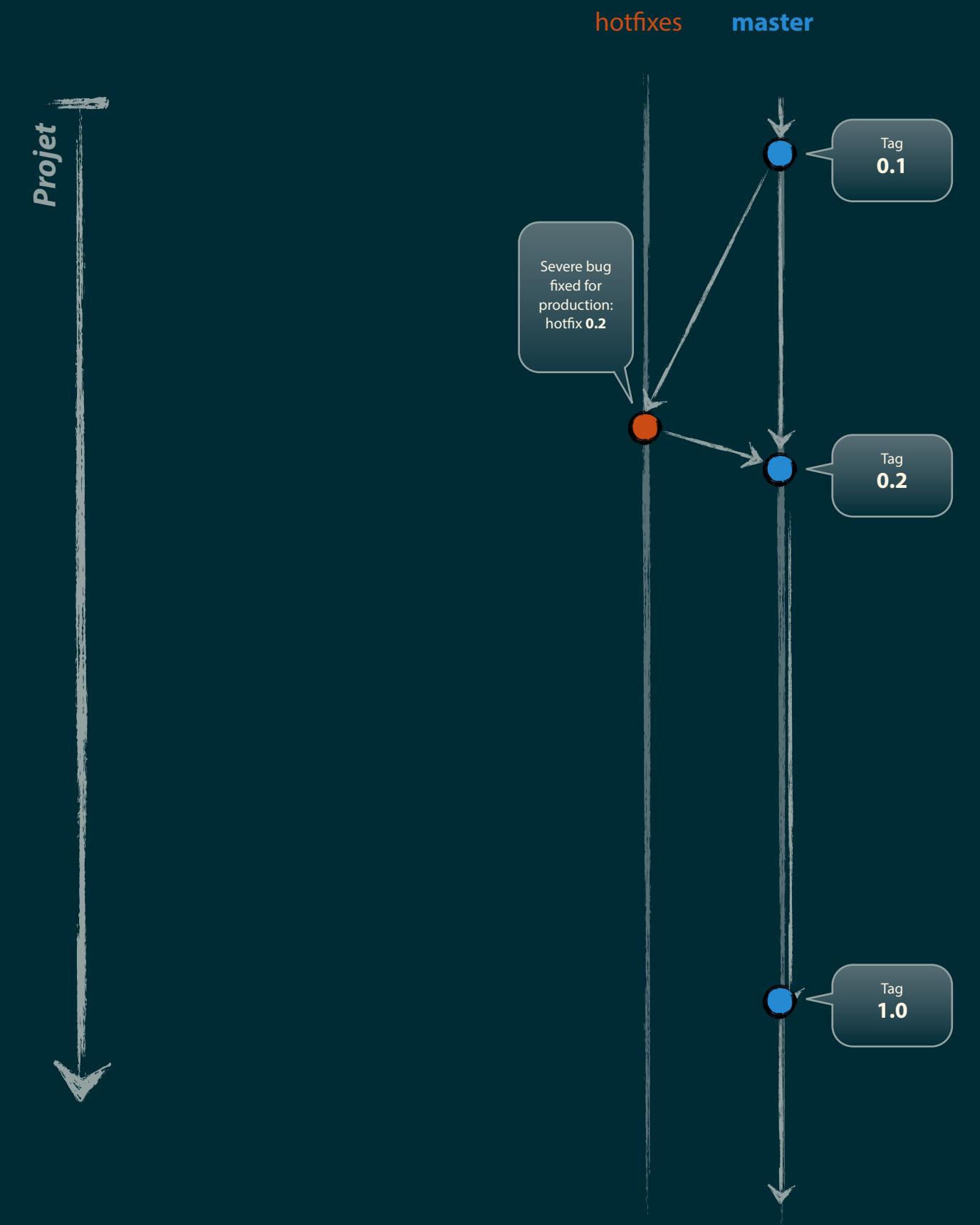
# .gitignore

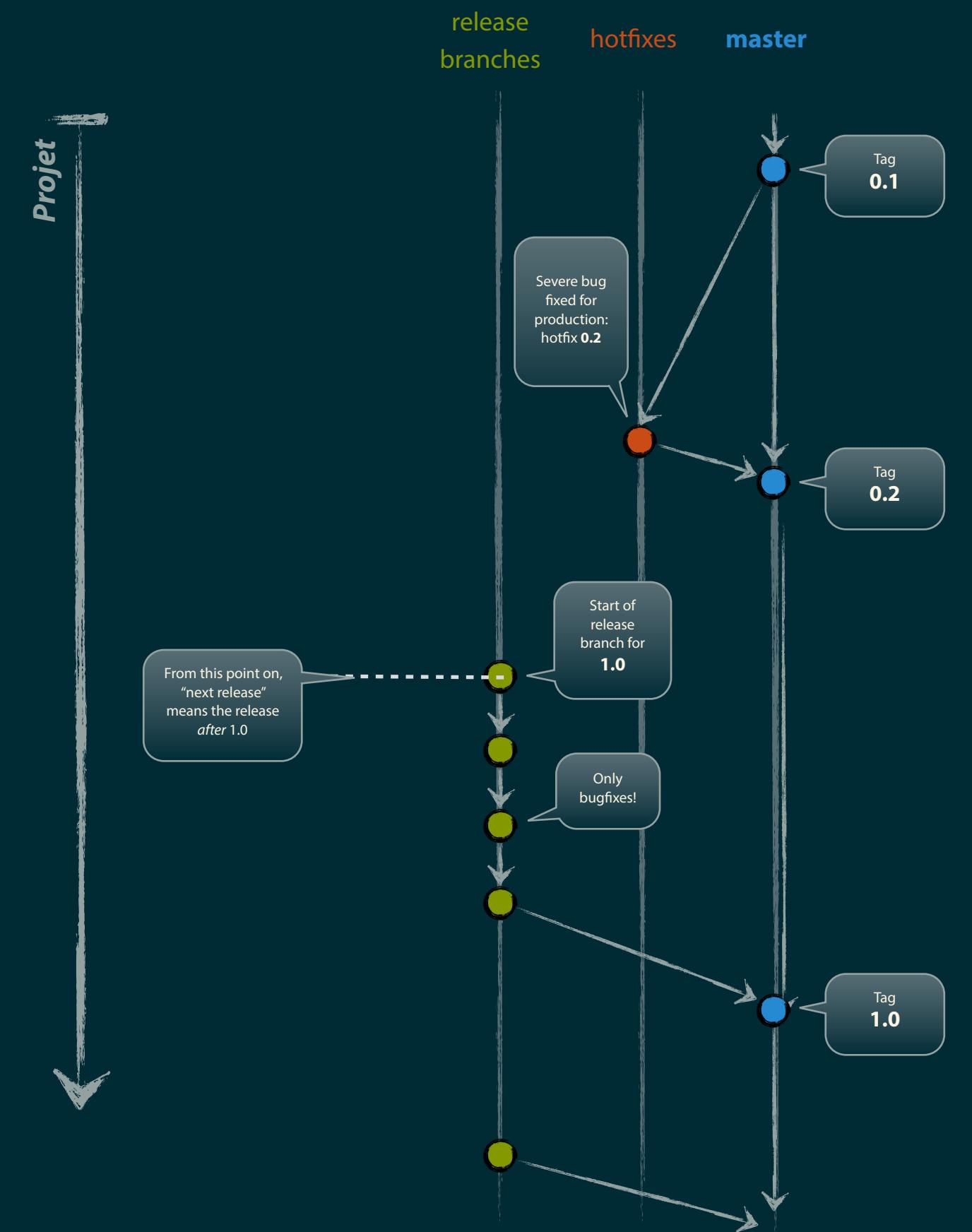
```
niels@idunn ~/Documents/Projets/en-bon-pere-de-famille [master *]
± % cat .gitignore
local-config.php
.DS_Store
sitemap.xml.gz
sitemap.xml
content/themes/bpf/sass-cache/*
```

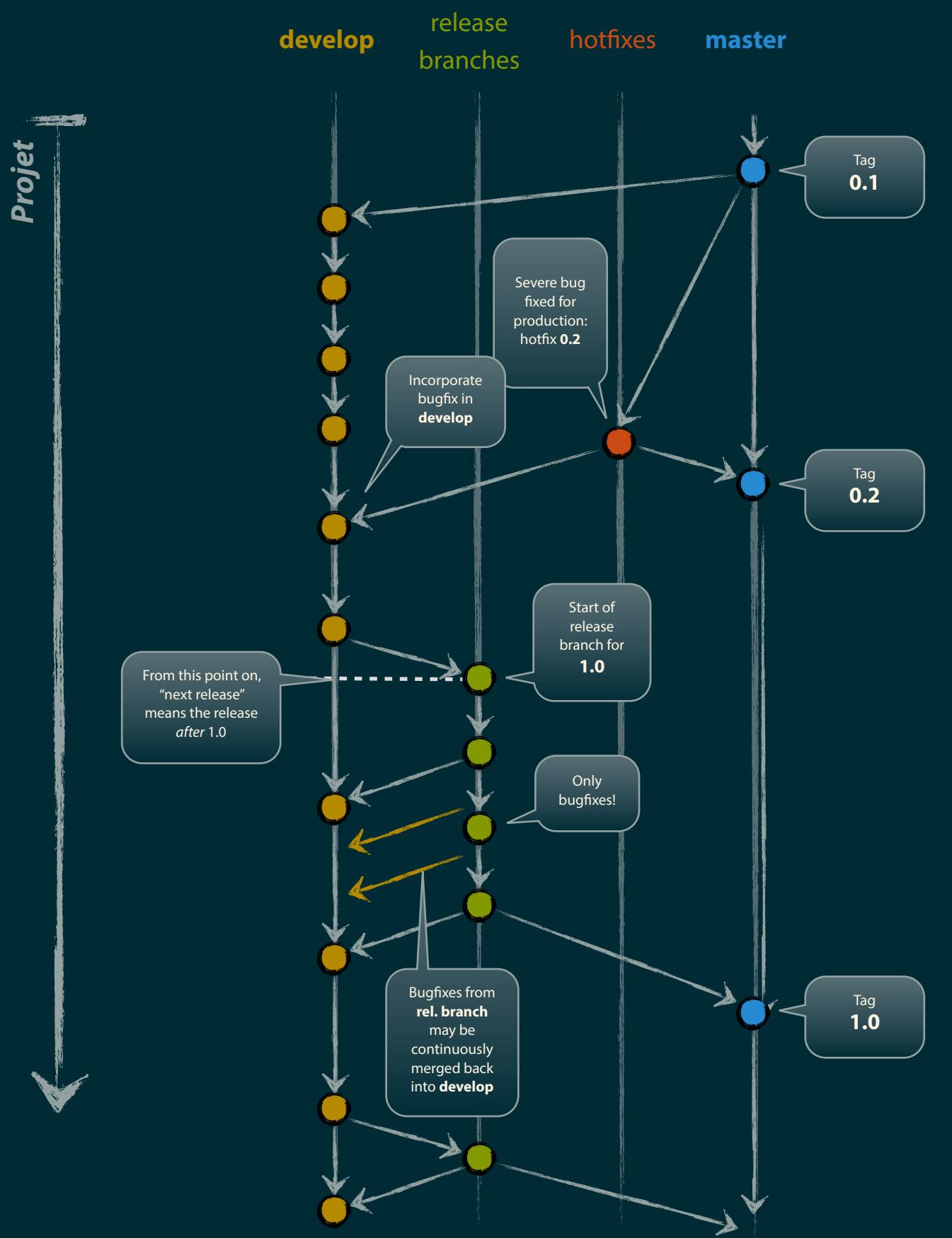
# Modèle de branches

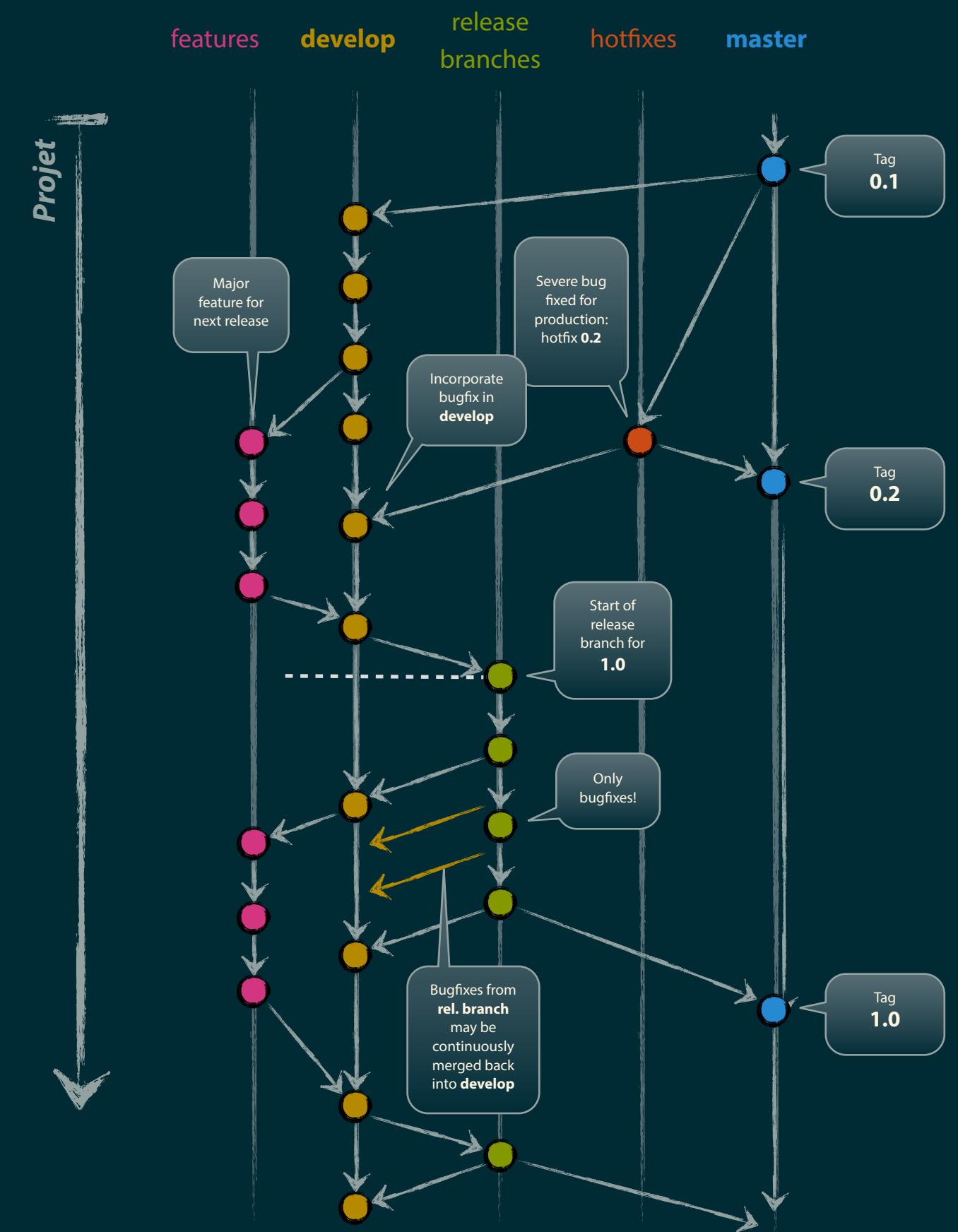
master

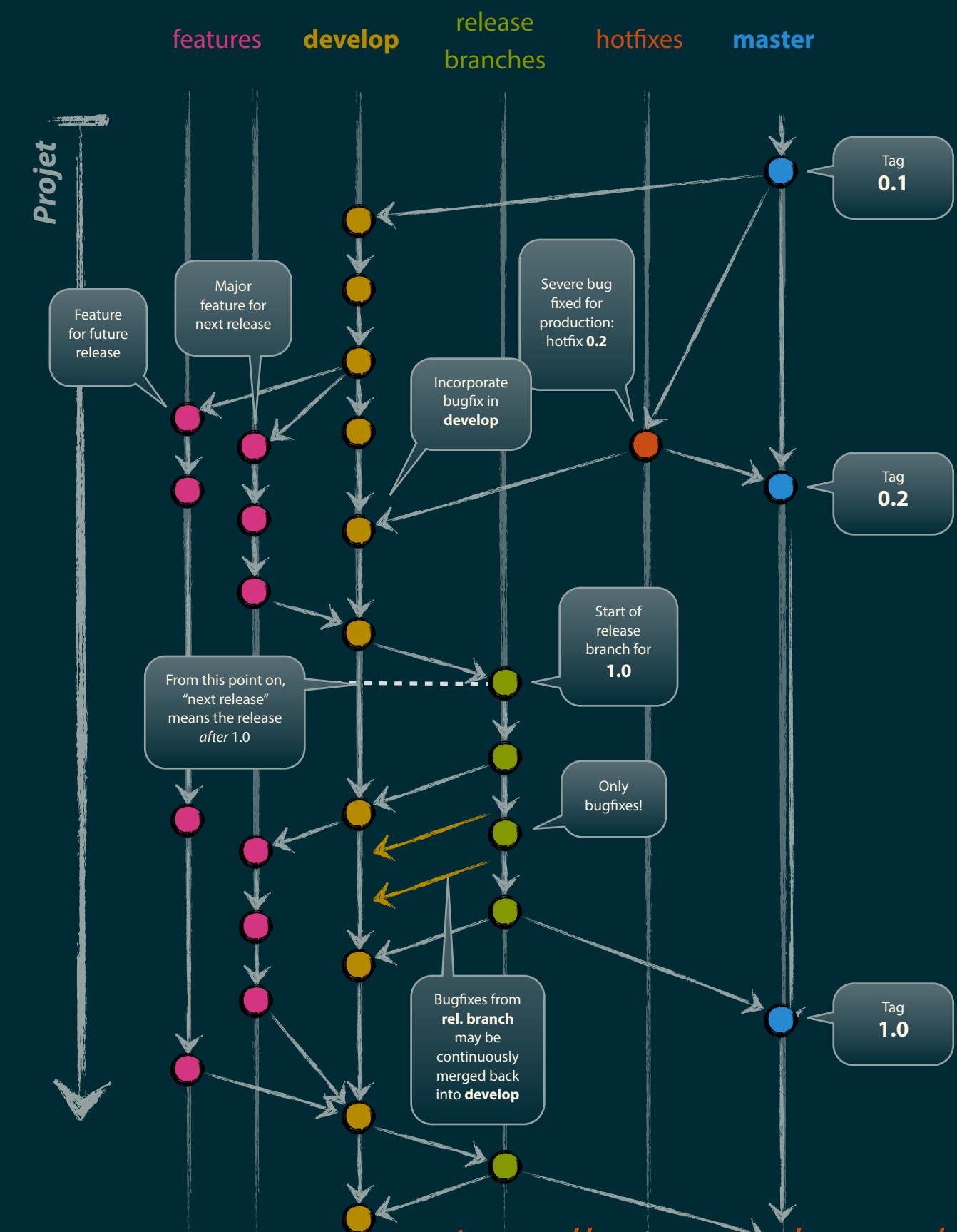




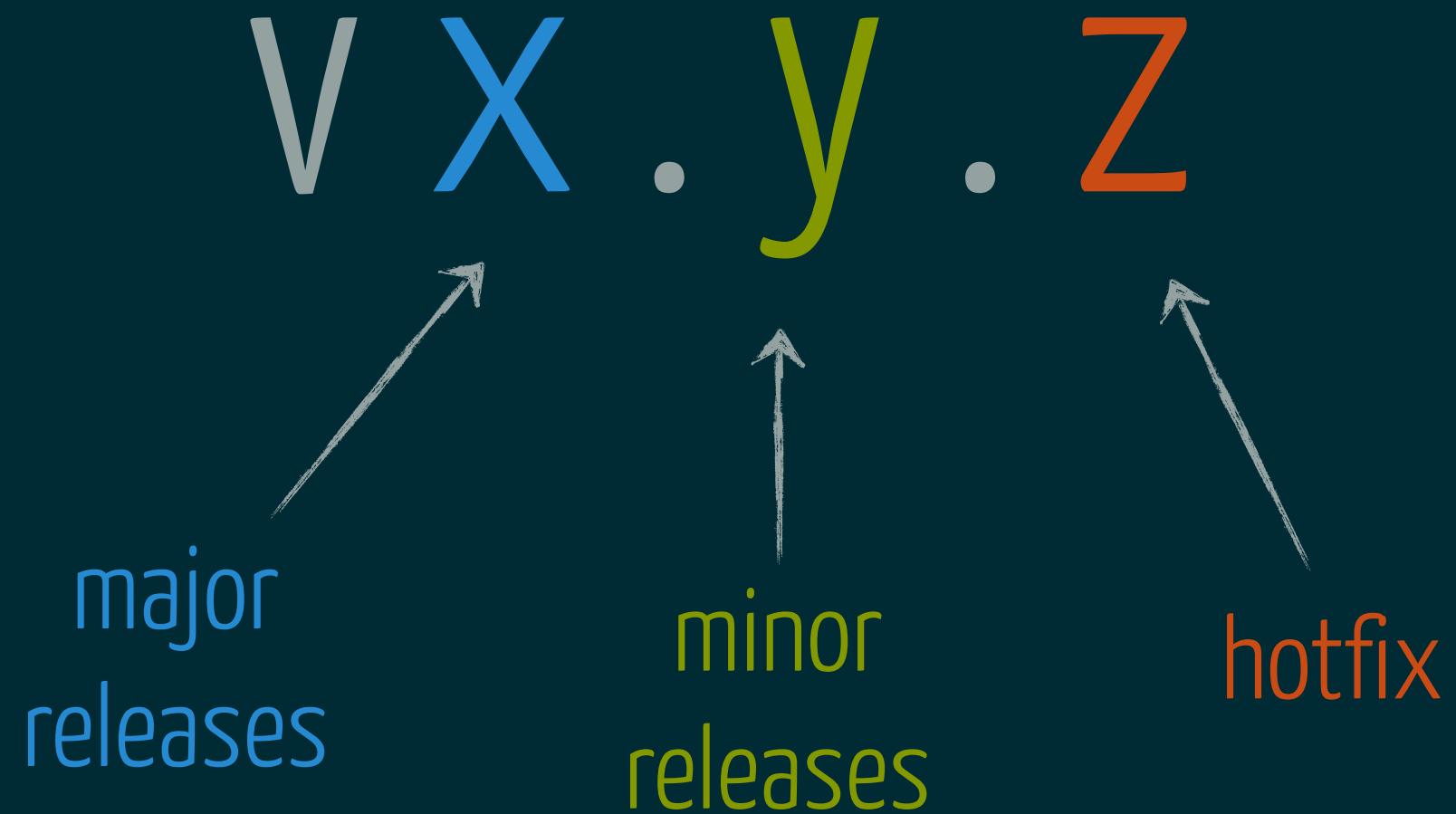








# Numérotter les versions

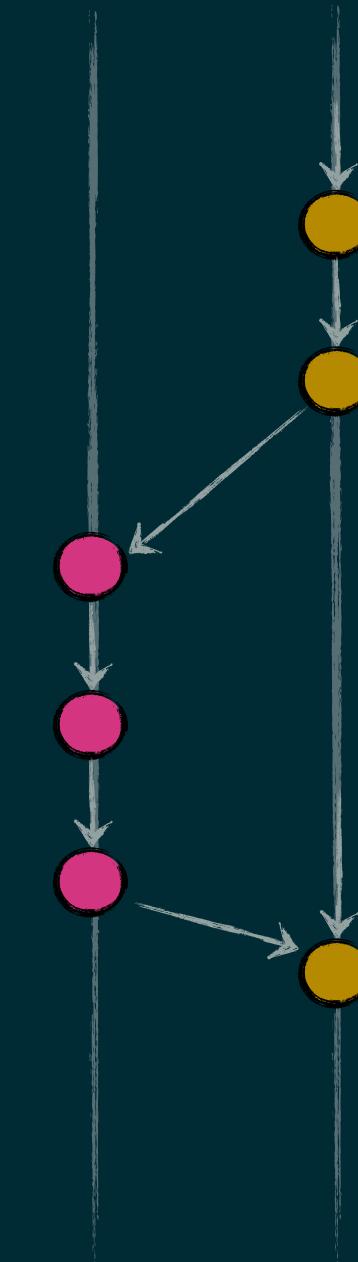


# Votre Workflow de base Git

```
$ git clone git@git.agence-soon.fr:/repo-test.git
(master) $ git checkout develop
(develop) $ git checkout -b task-666
(task-666) $ vim README
(task-666) $ git add .
(task-666) $ git commit -m "Ajout du README"
(task-666) $ git checkout develop
(develop) $ git merge --no-ff task-666
```

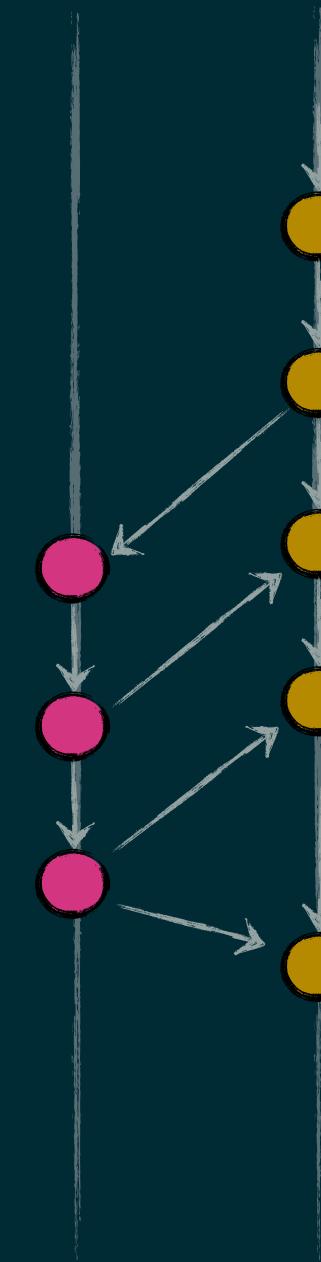
*merge, oui,  
mais en douceur :*

features      develop



git merge

features      develop



git rebase

I HAZ REBAZED



# Setup

- <http://help.github.com/set-up-git-redirect>
- <http://git-scm.com/download>

# Configurer votre shell Debian

```
aptitude install git-core git-completion zsh
```

```
chsh => /bin/zsh
```

```
wget --no-check-certificate https://github.com/robyrussell/oh-my-zsh/raw/master/tools/install.sh -O - | sh
```

```
vim .zshrc => ZSH_THEME="blinks"
```

The screenshot shows a Mac OS X terminal window with a dark theme. The title bar reads "2. niels@Idunn: ~/Documents/Projets/en-bon-pere-de-famille (zsh)". The window contains the following session:

```
niels@Idunn ~
% p
niels@Idunn ~/Documents/Projets
% cd passvip-backend
niels@Idunn ~/Documents/Projets/passvip-backend [develop]
± % cd ../en-bon-pere-de-famille
niels@Idunn ~/Documents/Projets/en-bon-pere-de-famille [master *]
± %
```

Terminal status indicators include a red dot at the top left, a yellow dot at the top center, and a green dot at the top right. The bottom right corner of the window has a small icon.

# Outils

- <http://code.google.com/p/gitextensions/>
- Desktop clients : Tower (Mac), SmartGit (Windows & Linux)
- (Bonus) git clone <https://github.com/Marak/The-Git-Rap.git>

# Notre workflow

- 1 Centralisation des repos sur vmtest
- 2 Développement en local sur vos machines
- 3 Tests sur VMs à dispositions
- 4 Recette client centralisée
- 5 Automatisation des déploiements

# Démo & exemples

### **== Clone ==**

```
git clone git@git.agence-soon.fr:/projet-test.git  
cd projet-test/  
ls  
git status
```

### **== Deploy ==**

```
cd /home/httpd/  
git clone git@git.agence-soon.fr:/projet-test.git projet-test.vmtest.agence-soon.fr
```

### **== Commit ==**

```
git checkout develop  
git checkout -b feature-1  
mate .gitignore  
git add .  
git commit -m "#1 feature-1 : Thème SASS"
```

### **== Hotfix ==**

```
git checkout -b feature2  
vim index.html  
git checkout master  
git pull  
git checkout -b hotfix1  
vim index.html  
git add .  
git commit -m "#2 hotfix1"  
git merge hotfix1  
git branch -d hotfix1
```

### **== Deploy Hotfix ==**

```
git pull
```

### **== Test ==**

```
git clone git@git.agence-soon.fr:/projet-test.git  
cd projet-test/  
ls  
git log  
git tag stable-1 id_commit
```

### **== Revert ==**

```
git checkout id_commit
```

Q & A ?

Fork me on GitHub.



git@github.com:gmoigneu/git-keynote-presentation.git

Thanks !