Informatique

Introduction à git et github

Matthieu Zimmer, Jean-Philippe Mangeot

Cycle Préparatoire Polytechnique

4 avril 2017

Plan

Historique / Inventaire

Git

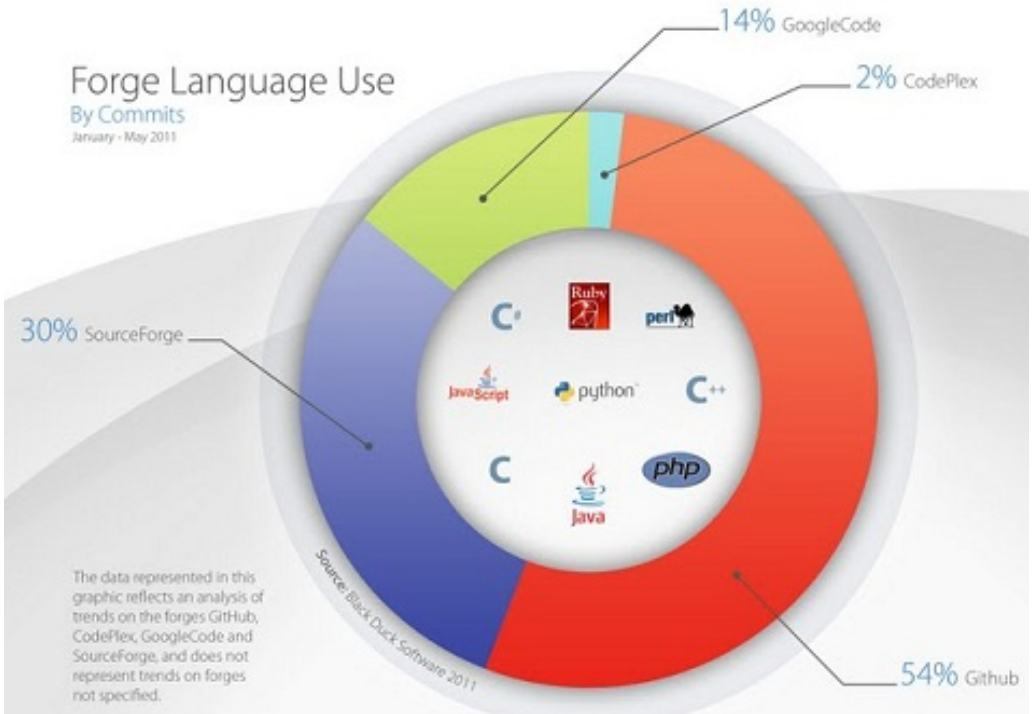
- Créer en 2005 par Linus Torvald
- Gérer les sources du noyau Linux
- Open source

```
git-add
git-add--interactive
git-am
git-annotate
git-apply
git-applymbox
git-applypatch
git-archimport
git-archive
git-bisect
git-blame
git-branch
git-bundle
git-cat-file
git-check-attr
git-check-ref-format
git-checkout
git-checkout-index
git-cherry
git-cherry-pick
git-citool
git-clean
git-clone
git-commit
git-commit-tree
git-config
git-convert-objects
git-count-objects
git-cvsexportcommit
git-cvsimport
git-cvsserver
git-daemon
git-describe
git-diff
git-diff-files
git-diff-index
git-diff-stages
git-diff-tree
git-fast-export
git-fast-import
git-fetch
git-fetch--tool
git-fetch-pack
git-filter-branch
git-fmt-merge-msg
git-for-each-ref
git-format-patch
git-fsck
git-fsck-objects
git-gc
git-get-tar-commit-id
git-grep
git-gui
git-hash-object
git-http-fetch
git-http-push
git-imap-send
git-index-pack
git-init
git-init-db
git-instaweb
git-local-fetch
git-log
git-lost-found
git-ls-files
git-ls-remote
git-ls-tree
git-mailinfo
git-mailsplit
git-merge
git-merge-base
git-merge-file
git-merge-index
git-merge-octopus
git-merge-one-file
git-merge-ours
git-merge-recur
git-merge-recursive
git-merge-recursive-old
git-merge-resolve
git-merge-stupid
git-merge-subtree
git-merge-tree
git-mergetool
git-mktag
git-mktree
git-mv
git-name-rev
git-pack-objects
git-pack-redundant
git-pack-refs
git-parse-remote
git-patch-id
git-peek-remote
git-prune
git-prune-packed
git-pull
git-push
git-quiltimport
git-read-tree
git-rebase
git-rebase--interactive
git-receive-pack
git-reflog
git-relink
git-remote
git-repack
git-repo-config
git-request-pull
git-rerere
git-reset
git-resolve
git-rev-list
git-rev-parse
git-revert
git-rm
git-runstatus
git-send-email
git-send-pack
git-sh-setup
git-shell
git-shortlog
git-show
git-show-branch
git-show-index
git-show-ref
git-ssh-fetch
git-ssh-pull
git-ssh-push
git-ssh-upload
git-stash
git-status
git-stripspace
git-submodule
git-svn
git-svnimport
git-symbolic-ref
git-tag
git-tar-tree
git-unpack-file
git-unpack-objects
git-update-index
git-update-ref
git-update-server-inf
git-upload-archive
git-upload-pack
git-var
git-verify-pack
git-verify-tag
git-web--browse
git-whatchanged
git-write-tree
```

Historique / Inventaire

Github *Social Coding*

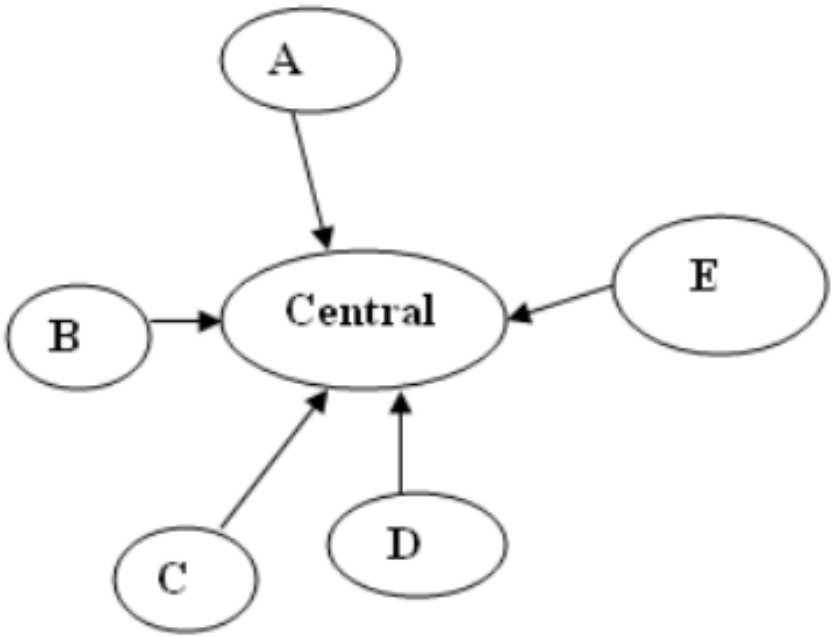
- Créer en 2008 par Wanstrath, Hyett et Preston-Werner
- Closed source
- Noyau linux, Ruby on Rails, jQuery, Diaspora, Spring, php, ..

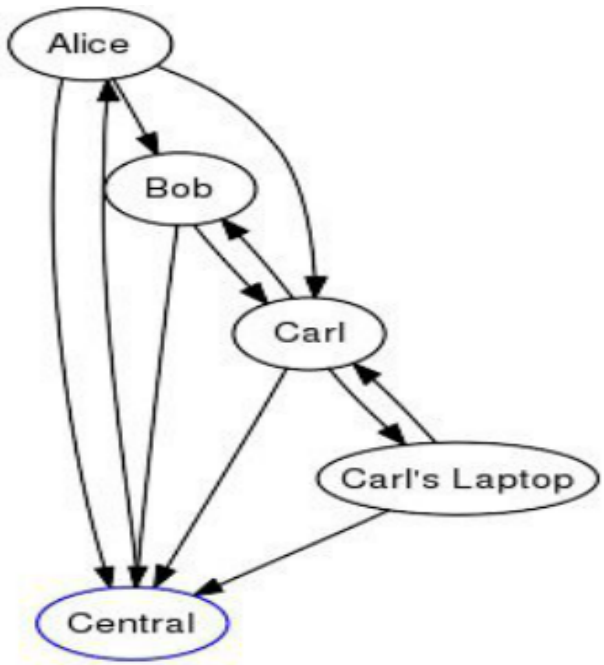


Pourquoi git ?

- Développement décentralisé
  - Chaque développeur possède l'historique complet du projet
  - Création/Fusion des branches facilités
  - Branches locales
  - Pas besoin de connexion pour commiter
- Efficace sur les toutes les tailles de projets
- Rapide (accès locals)
- ...

Développement SVN vs GIT





Régler github

- Créer compte sur `https://github.com/`
- En console
  - cd && mkdir .ssh
  - cd .ssh
  - ssh-keygen -t rsa -C 'your_email@youremail.com'
  - ssh-agent /bin/bash
  - ssh-add nom_cle_privée
- Ajouter la clé publique (.pub) sur le site :
  Account Settings > SSH Keys > Add SSH key
- En console
  - rm clé_publique
  - ssh -T git@github.com

Régler git

- En console
  - git config −−global user.name 'Firstname Lastname'
  - git config −−global user.email 'your_email@youremail.com'
  - echo "export EDITOR='gedit' " >> *.bashrc*

Créer un repository test

- Github : New repository (Initialize this repository)

Cloner un repository

Commande

git clone <repository>
repository : git@github.com :votre_compte/votre_projet_test.git

Action

Récupère/Clone tout l'historique (commit, branch, tags, ...) d'un repository distant.

Modifications

Test.java :

Ajouter “Hello word” dans la méthode main.

Ajouter/Supprimer des fichiers/modifications au repository

Commande

git add <files>
git rm <files>
files : Test.java

Action

Ajoute/Supprime les modifications/fichier

Astuce

Pour ajouter tout les fichiers/modifications apportées : git add .

Valider les modifications

Commande

git commit

Action

Valide les modifications et créer une nouvelle version locale.

Attention

Le commit s'effectue uniquement sur la branche actuelle.

Envoyer ces modifications

Commande

git push

Action

Envoyer ces modifications au serveur distant

Récupérer les modifications

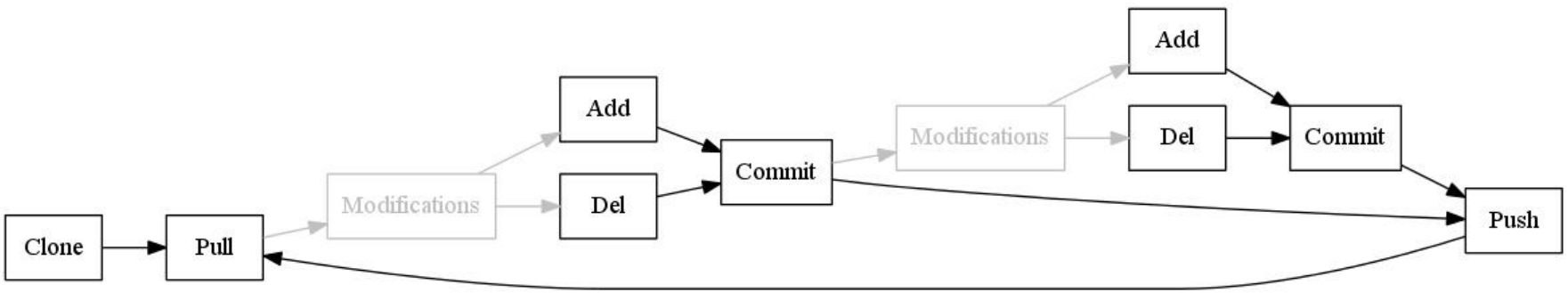
Commande

git pull

Action

Récupérer les modifications des autres utilisateurs à partir du serveur distant

Workflow


Clone
Pull
Modifications
Add
Del
Commit
Modifications
Add
Del
Commit
Push

Unstage

Problème

Vous avez “git add” des modifications/des fichiers que vous ne vouliez pas ajouter. Mais vous ne voulez pas les supprimer non plus.

Solution

git reset HEAD <fichier>

Revenir en arrière

Problème

Vous avez “git add” des modifications/des fichiers que vous ne vouliez pas ajouter. Ou, vous voulez revenir en arrière, parce que vos modifications ne sont plus d’actualitées.

Solution

git checkout −− <fichier>

Perdu

Problème

Vous ne savez plus où vous en êtes : quel fichier à été ajouté, lequel ne l'a pas été, lequel a subit des modifications

Solution

git status

Quelles modifications ont été apportées sur le fichier

Problème

Vous ne savez plus ce que vous avez modifié dans un fichier.

Solution

git diff <fichier>

Historique

Afficher l'historique des commits

git log

Un oubli

Problème

Vous avez oublié d'ajouter un fichier lors du dernier commit.

Solution

git commit –amend

C'est quoi ?

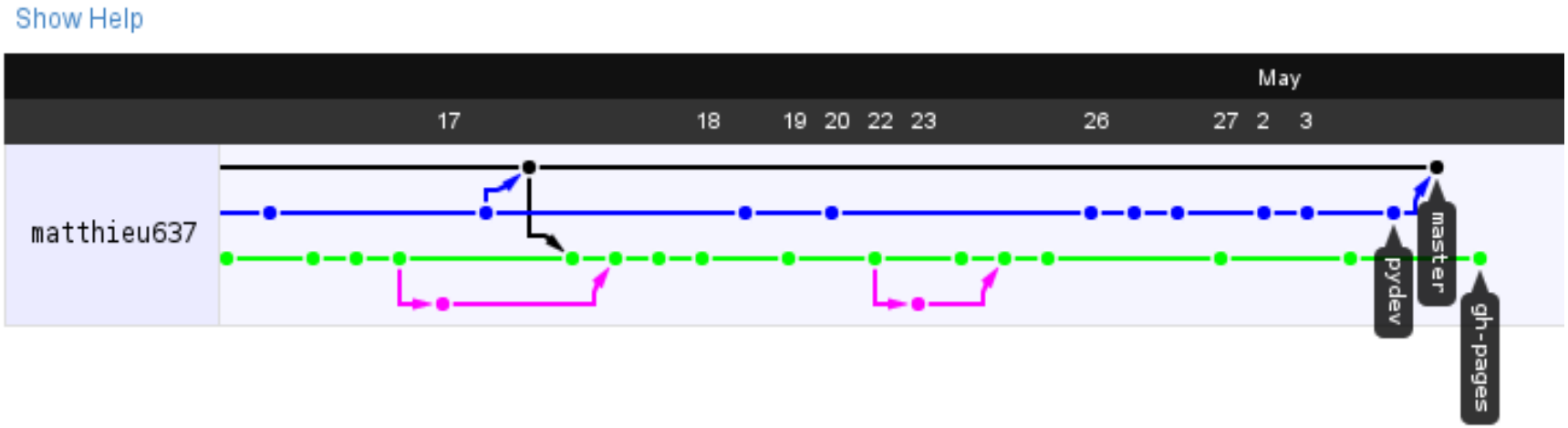
Permet d'avoir plusieurs fils d'historique :

1 par ticket/feature

essayer une idée

faire un hotfix

...



Visualiser les branches

Afficher les branches, et celle où on est

git branch

Changer de branche

git checkout ‹nom branche›

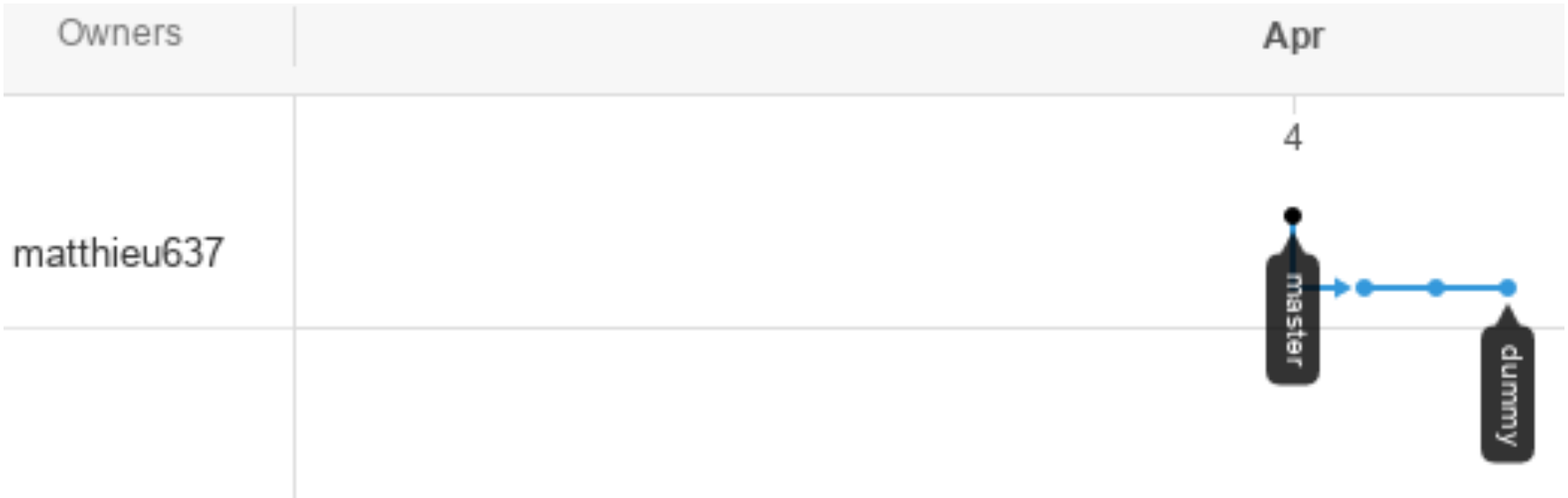
Créer une branche

Créer une branche **locale** à partir de master

git checkout -b ‹nom branche›

Publier ma branche locale

git push origin ‹nom branche›



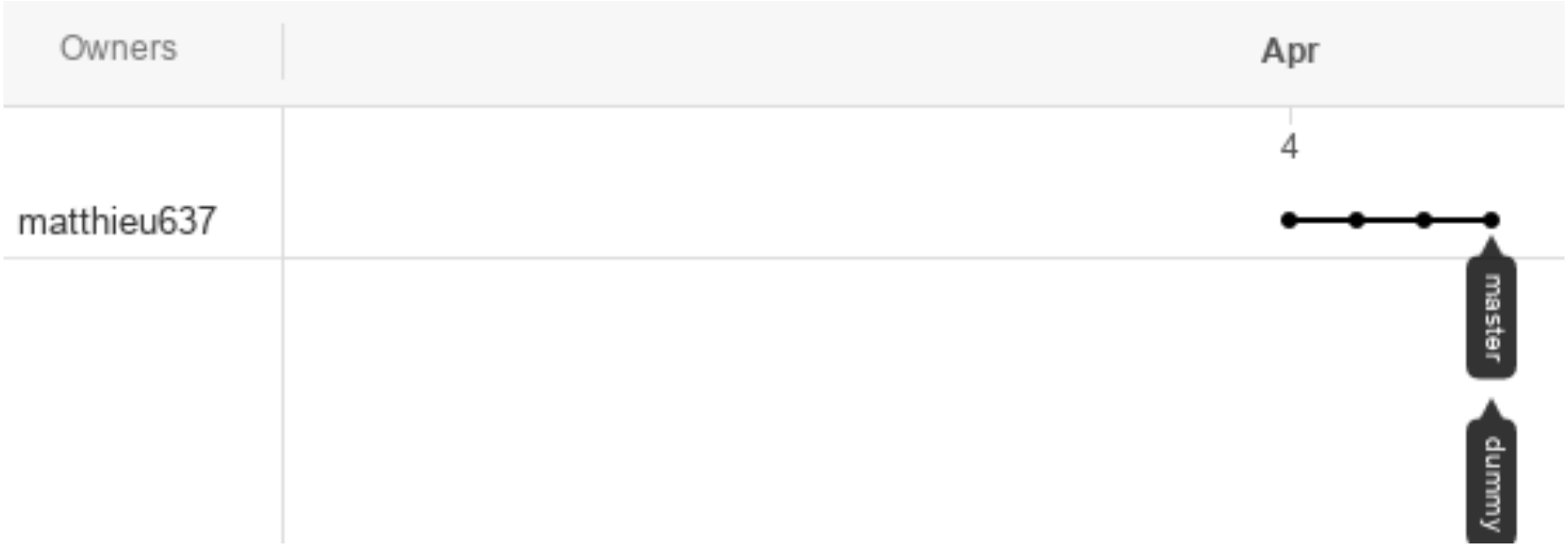
Merge une branche

Se placer dans la branche de destination

git checkout ‹nom branche destination›

Fusionner

git merge ‹nom branche provenance›



Merge une branche

Publication

Ne pas oublier de push si tout va bien

Supprimer la branche distante

Ne pas oublier de push si tout va bien

Synchronisation

Si la branche master a été mis à jour depuis ? Il faut déjà d'abord la merge.

Supprimer branche locale

git branch -D ‹nom branche›

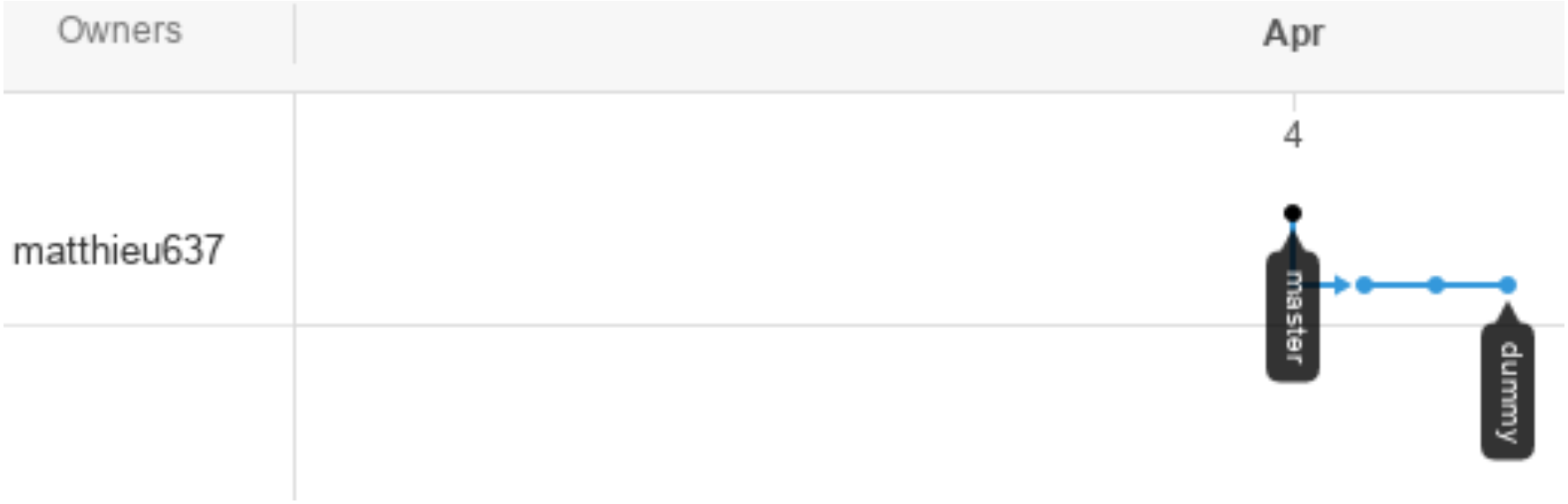
Conflit

Outils pour merge

git mergetool

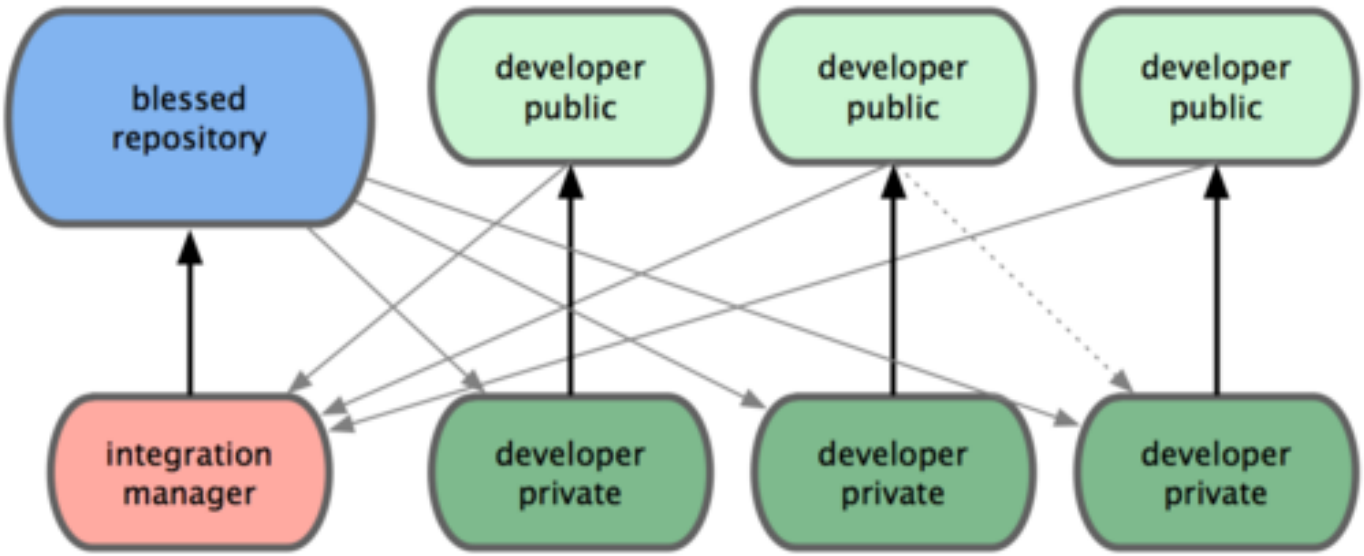
Le merge produit des bugs

git reset –hard ‹commit id de master avant merge›
git push -f



Organisation 1

- Le mainteneur du projet push vers son dépôt public.
- Un contributeur clone ce dépôt et introduit des modifications.
- Le contributeur push son travail sur son dépôt public.
- Le contributeur demande au mainteneur de pull depuis son dépôt.
- Le mainteneur fusionne le dépôt distant.
- Le mainteneur push les modifications sur le dépôt principal.



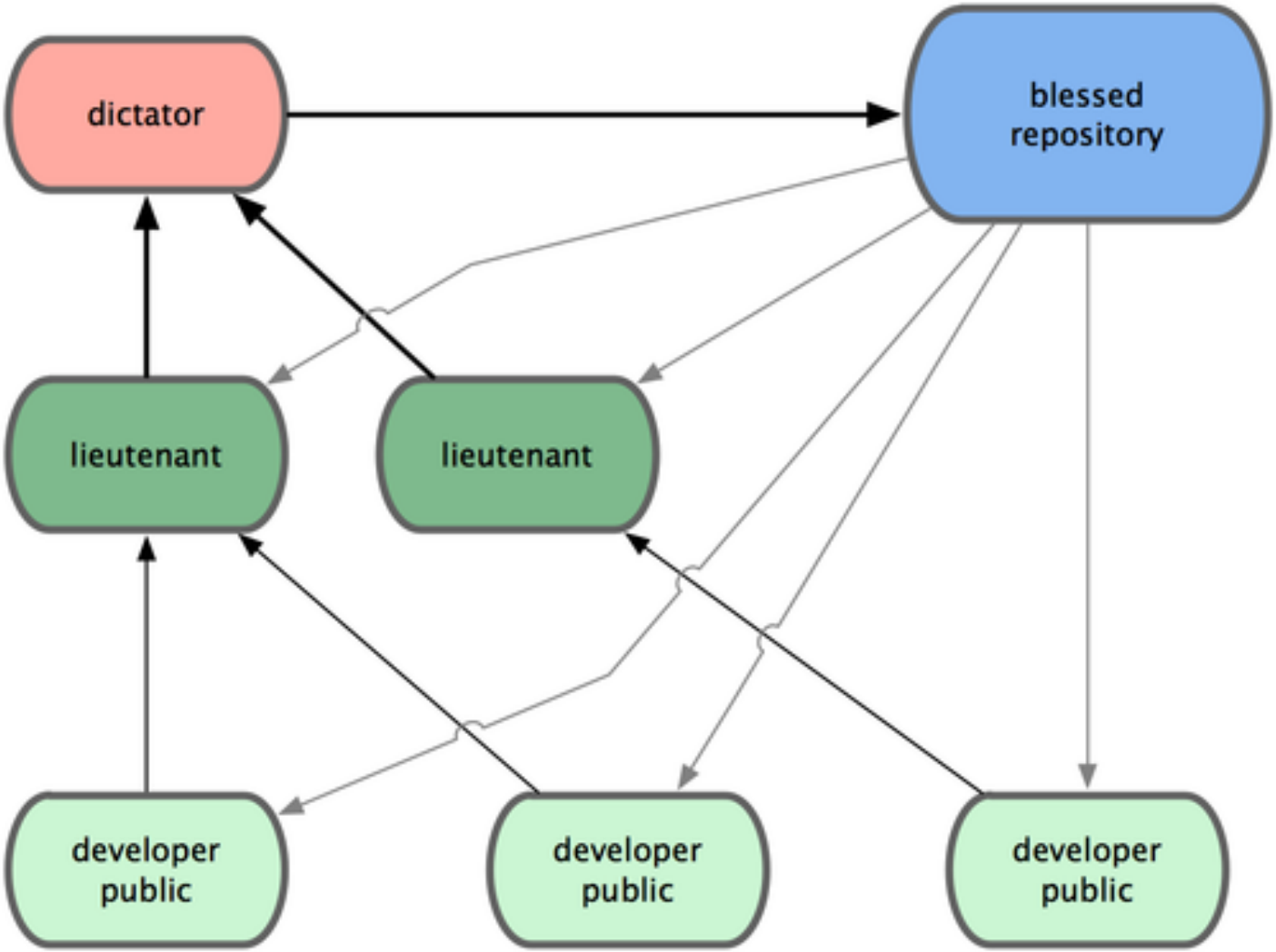
Organisation 1 avec Github

Avec Github

Fork : cloner un projet github, dans son github personnel

Pull request : demander au mainteneur de pull depuis son dépôt.

Organisation 2

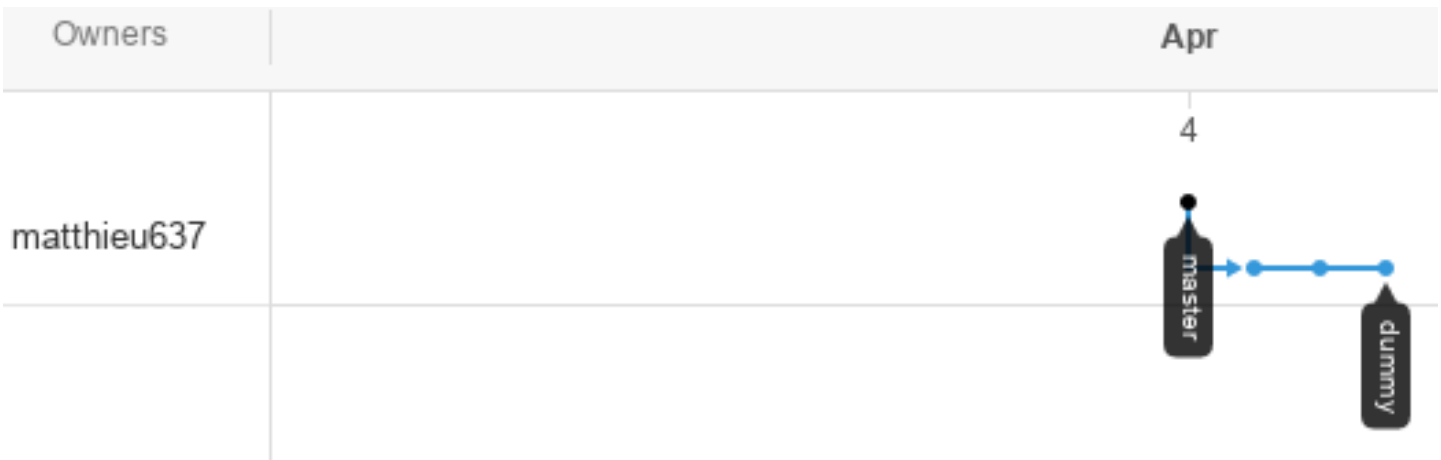


Pull request - Github

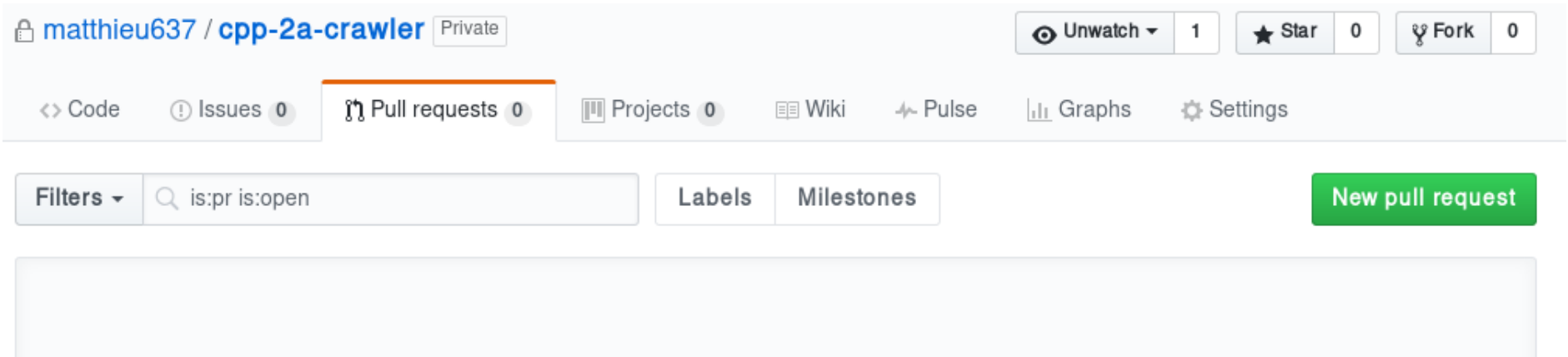
Double relecture

Pas de merge dans master soi-même

On ne merge que les pull request des autres



Pull request - Github



Votre branche doit avoir été publiée avant.

Pull request - Github

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base: master ... compare: dummy ✓ Able to merge. These branches can be automatically merged.

Create pull request Discuss and review the changes in this comparison with others.

3 commits 3 files changed 0 commit comments 1 contributor

Commits on Apr 04, 2017

matthieu637	t1	ea371b2
matthieu637	t2	86d2c36
matthieu637	t2	ec08b24

Liens

Git pro : `http://git-scm.com/book/fr`