

Développement coopératif versus développement collaboratif :

**Quels processus de partage de
code et de contextes d'exécution ?**

Quels workflows choisir ?

Claire Mouton - CREATIS, Lyon, France

claire.mouton@creatis.insa-lyon.fr

Du développement collaboratif au développement coopératif

Développement collaboratif

Une équipe

Un seul objectif

Développement coopératif

Des équipes/personnes

communauté plus ou moins informelle

Plusieurs objectifs

nécessité de coopérer pour réutiliser des codes existants

systèmes de plus en plus complexes

Développement coopératif

Plusieurs solutions

Modularité

Architecture micro-services SOA / WOA

Standard

Urbanisation

...

Ici, nous nous intéressons plus spécifiquement au développement d'une brique
logicielle

Développement par composants

Le contexte d'exécution et les conteneurs

Contexte d'exécution

Collaboration implique une variabilité potentielle des contextes d'exécution

Conteneur

Une solution pour s'affranchir de la variabilités des contextes d'utilisation intra ou inter collaborateurs

1 service = 1 conteneur = 1 projet

A versionner avec le code

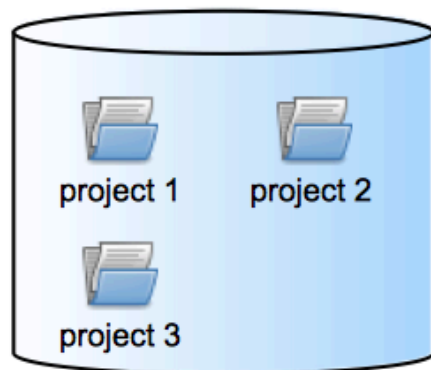
Rappel des processus

Développement collaboratif: dépôt de code centralisé (SVN)

Développement collaboratif: dépôt de code centralisé (SVN)

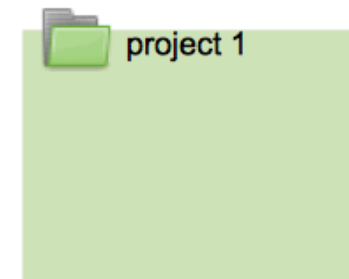
Organisation

Repository
[//server/repository/]



- all versions
- ancestry information

Working copy
[~/my_project/]



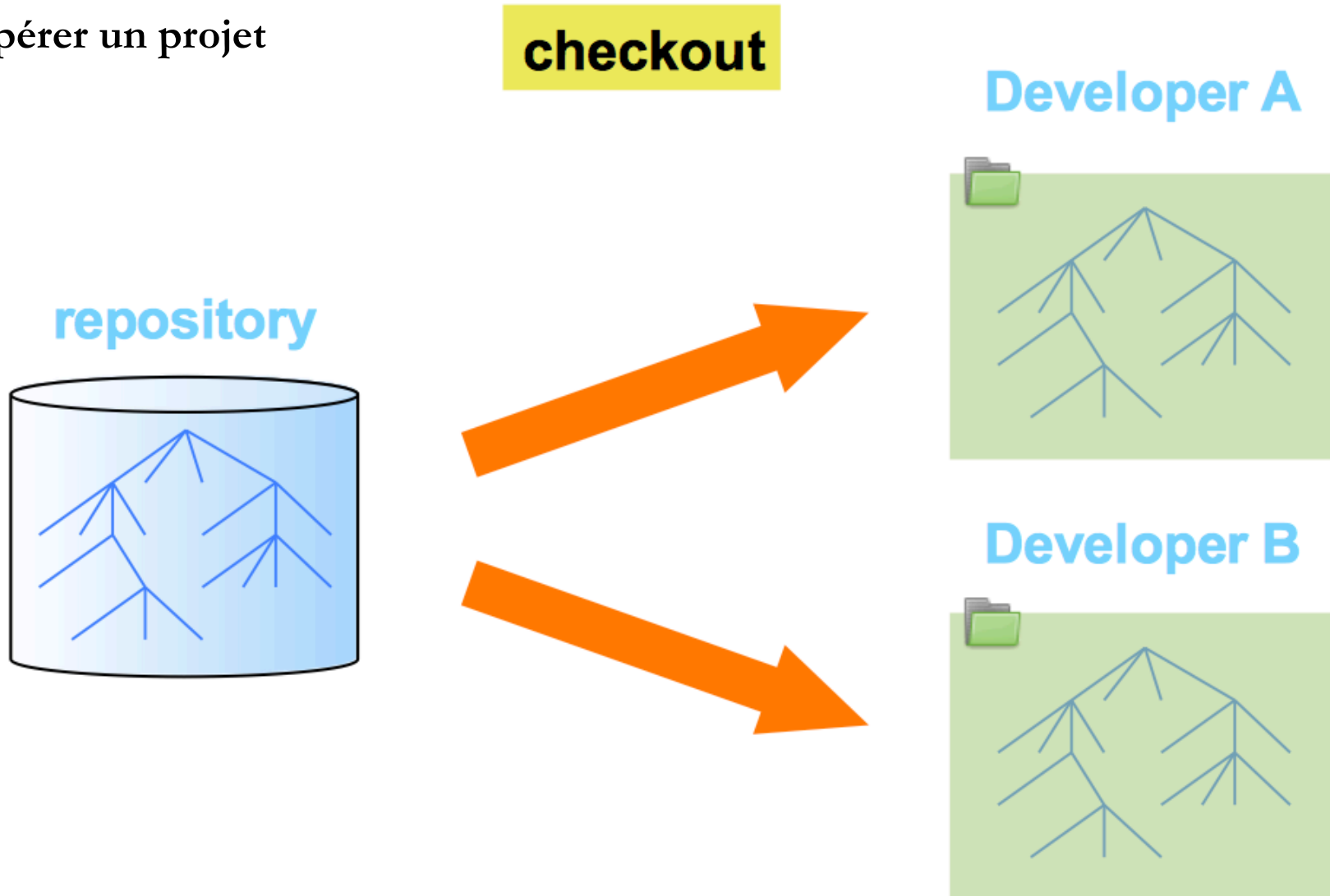
- private workspace
- state information



Développement collaboratif: dépôt de code centralisé (SVN)

Rappel des processus :

Récupérer un projet

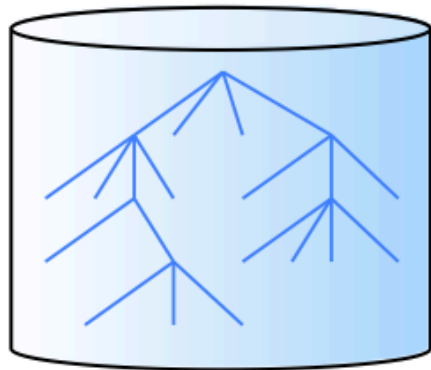


Développement collaboratif: dépôt de code centralisé (SVN)

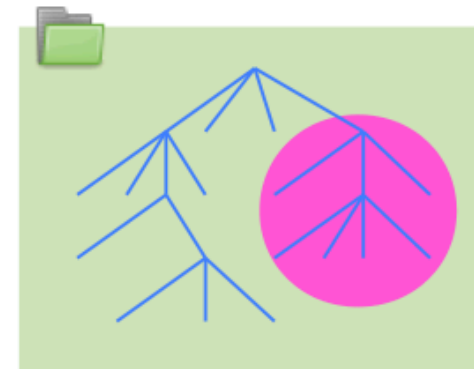
Rappel des processus

development

repository



Developer A

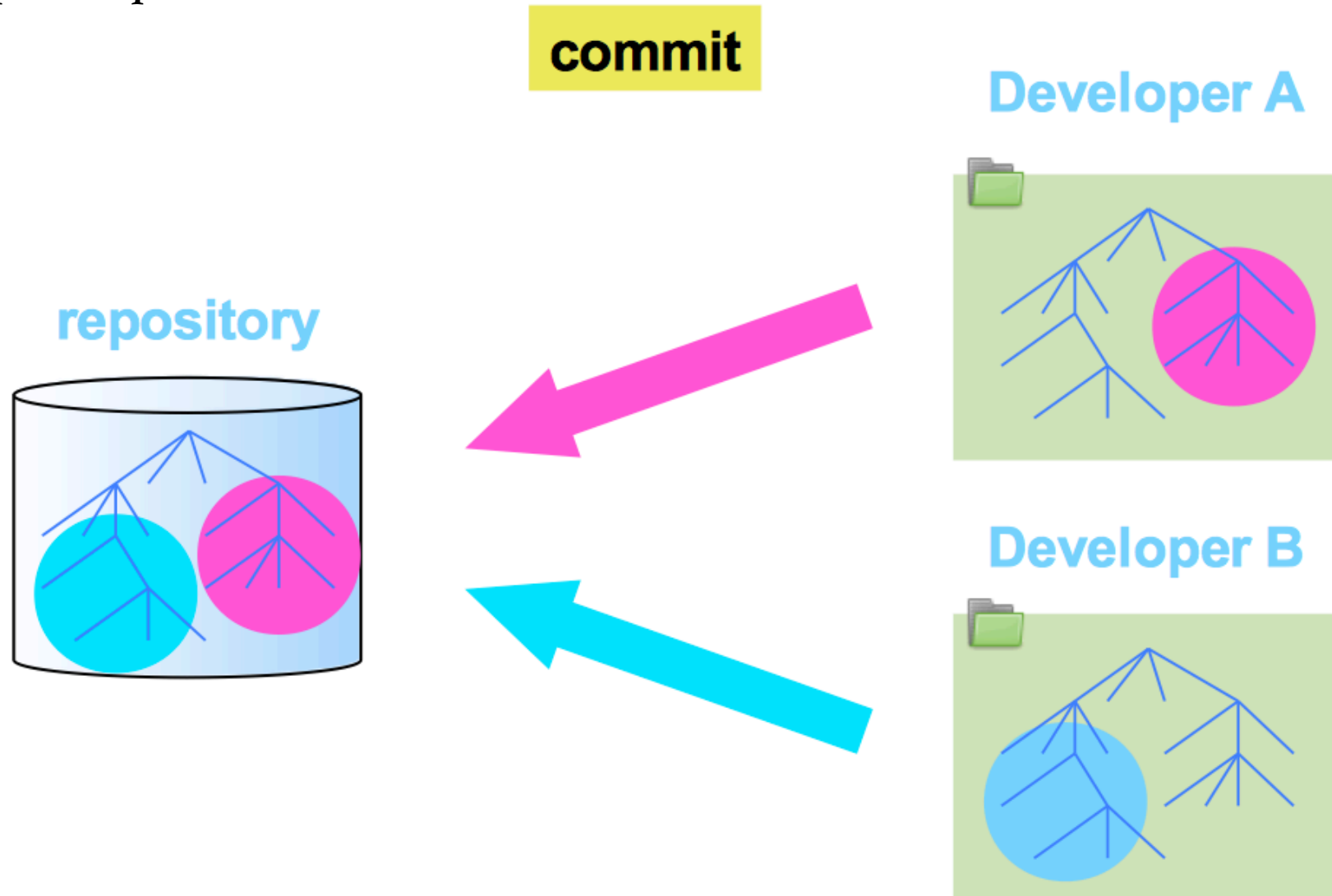


Developer B



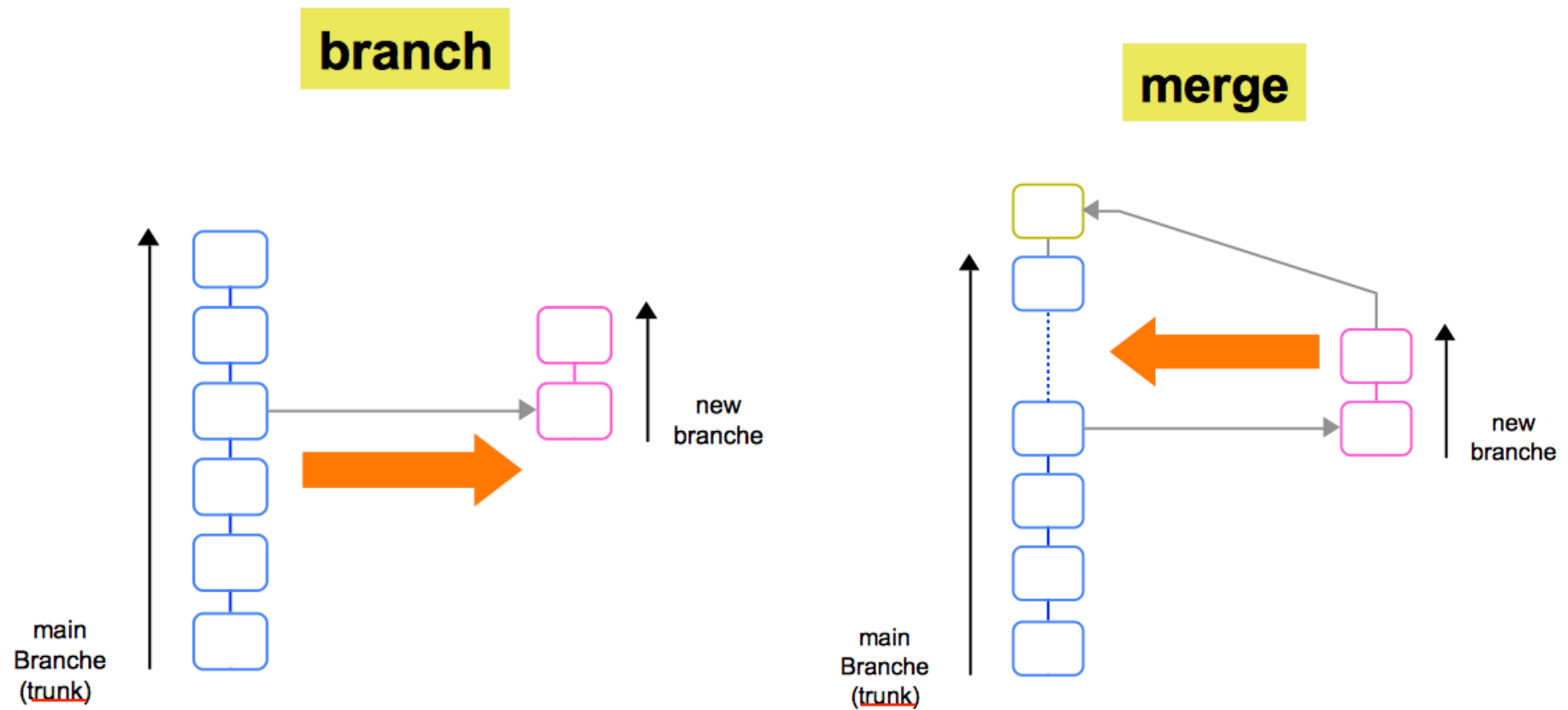
Développement collaboratif: dépôt de code centralisé (SVN)

Rappel des processus



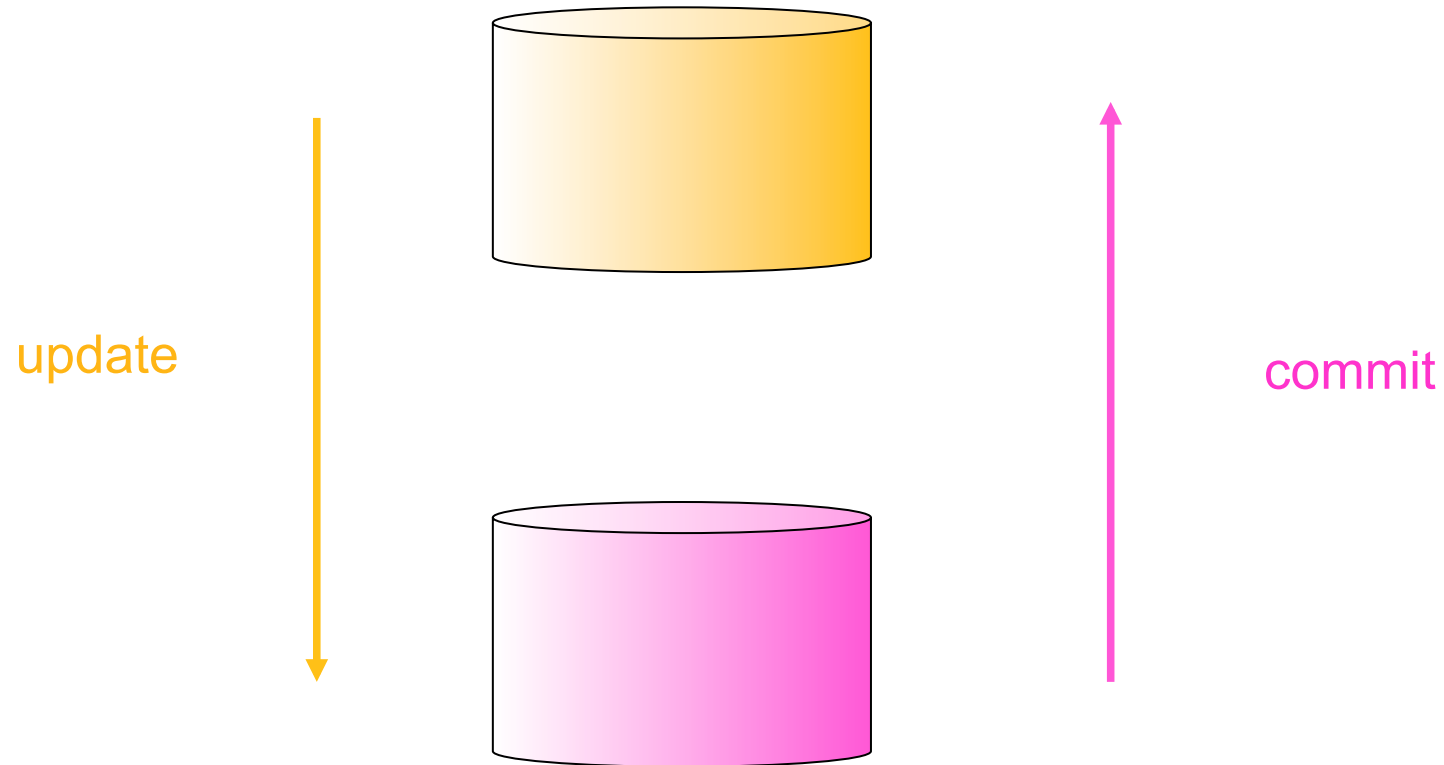
Développement collaboratif: dépôt de code centralisé (SVN)

Rappel des processus



Développement collaboratif: dépôt de code centralisé (SVN)

Echanges avec le dépôt distant



Développement coopératif: dépôt de code décentralisé (Git)

Développement coopératif: dépôt de code décentralisé (Git)

Nouveautés par rapport à SVN :

Aucun dépôt de référence par défaut

Copie de travail = bac à sable

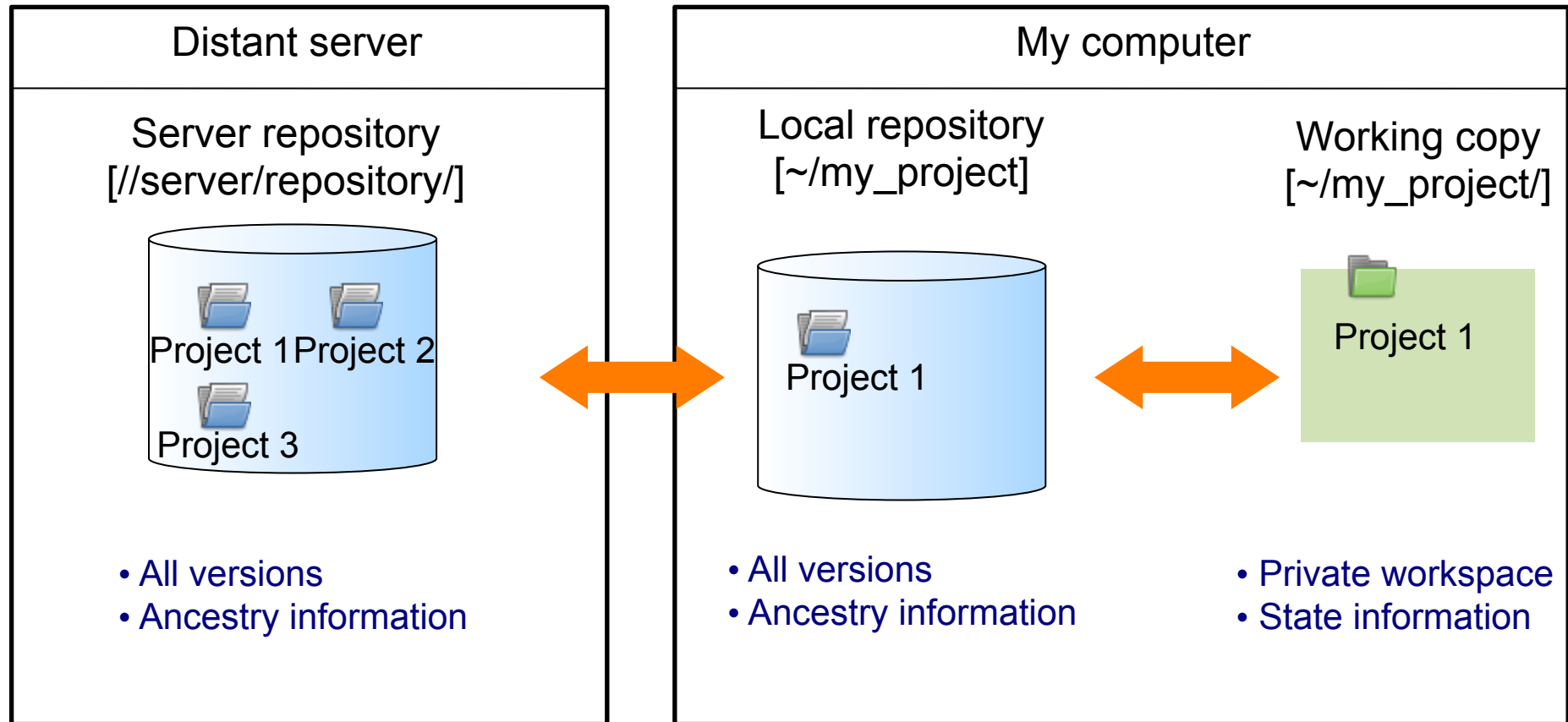
Facilité de création /manipulation des branches

Flexibilité

Commandes disponibles hors ligne

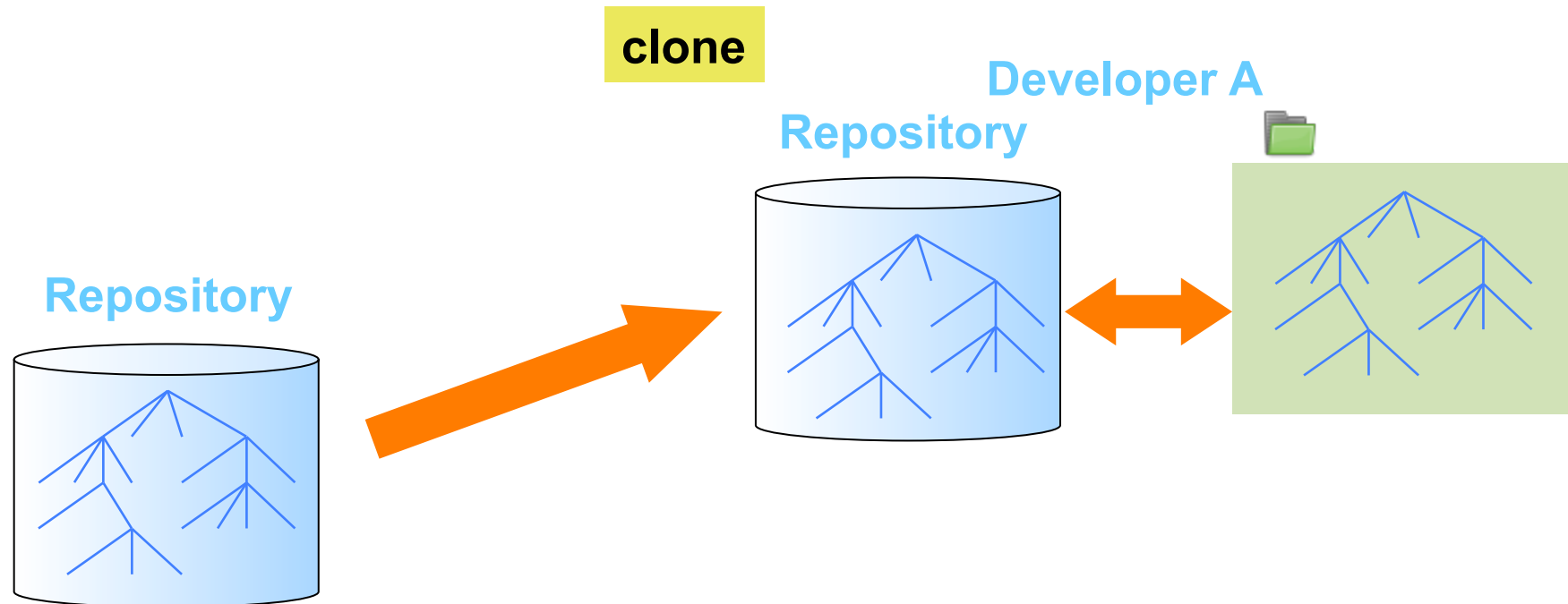
Développement coopératif: dépôt de code décentralisé (Git)

Organisation



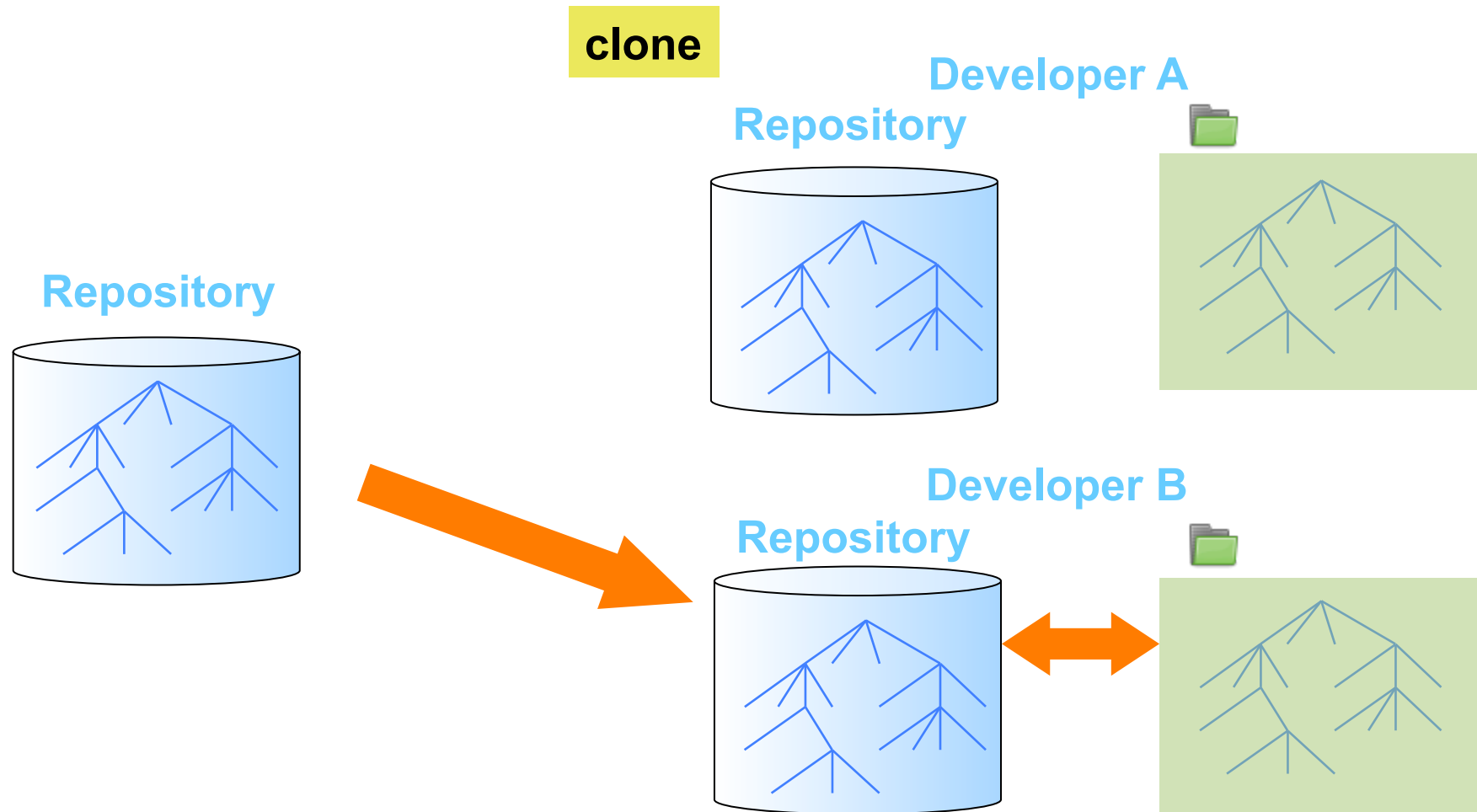
Développement coopératif: dépôt de code décentralisé (Git)

Rappel des processus



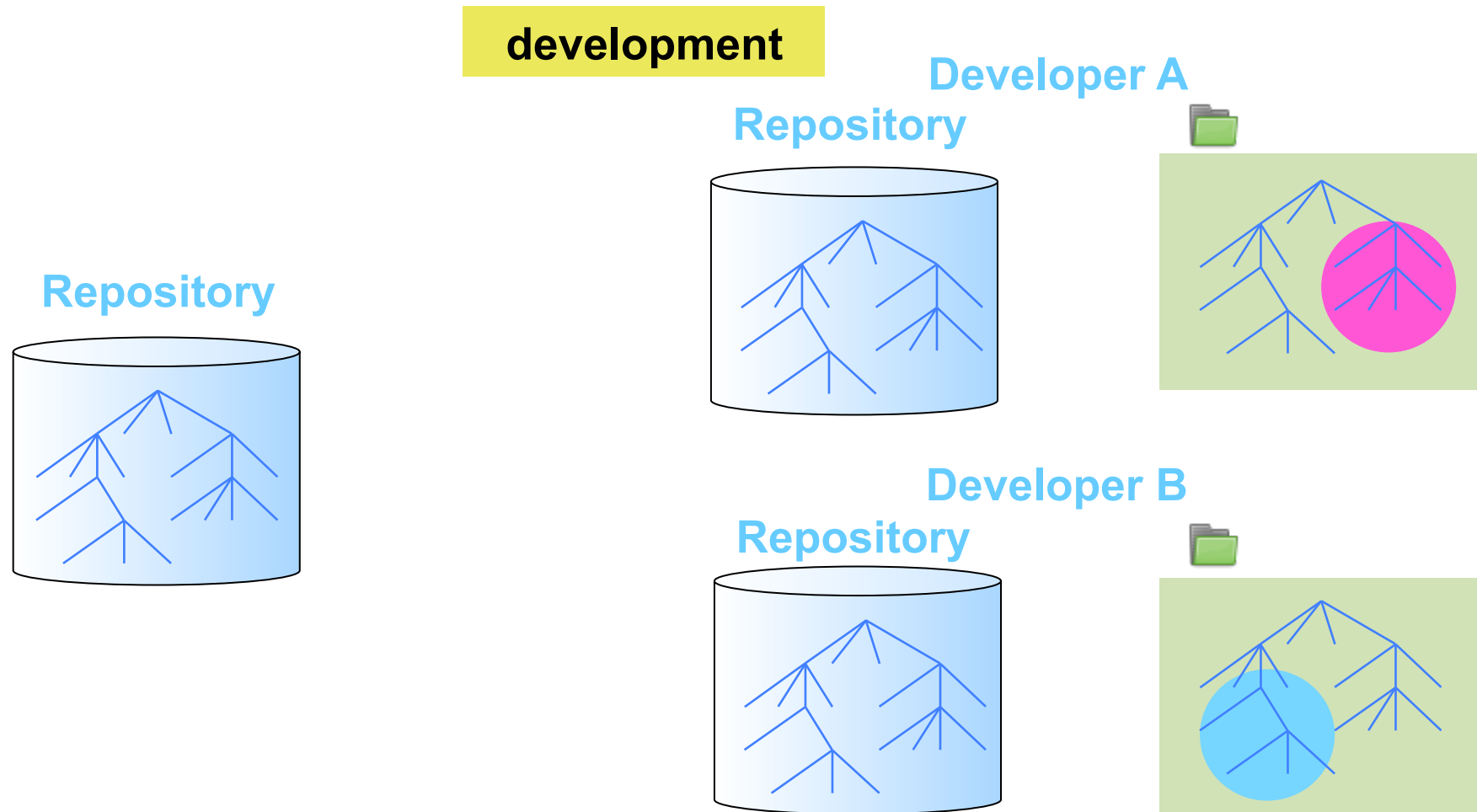
Développement coopératif: dépôt de code décentralisé (Git)

Rappel des processus



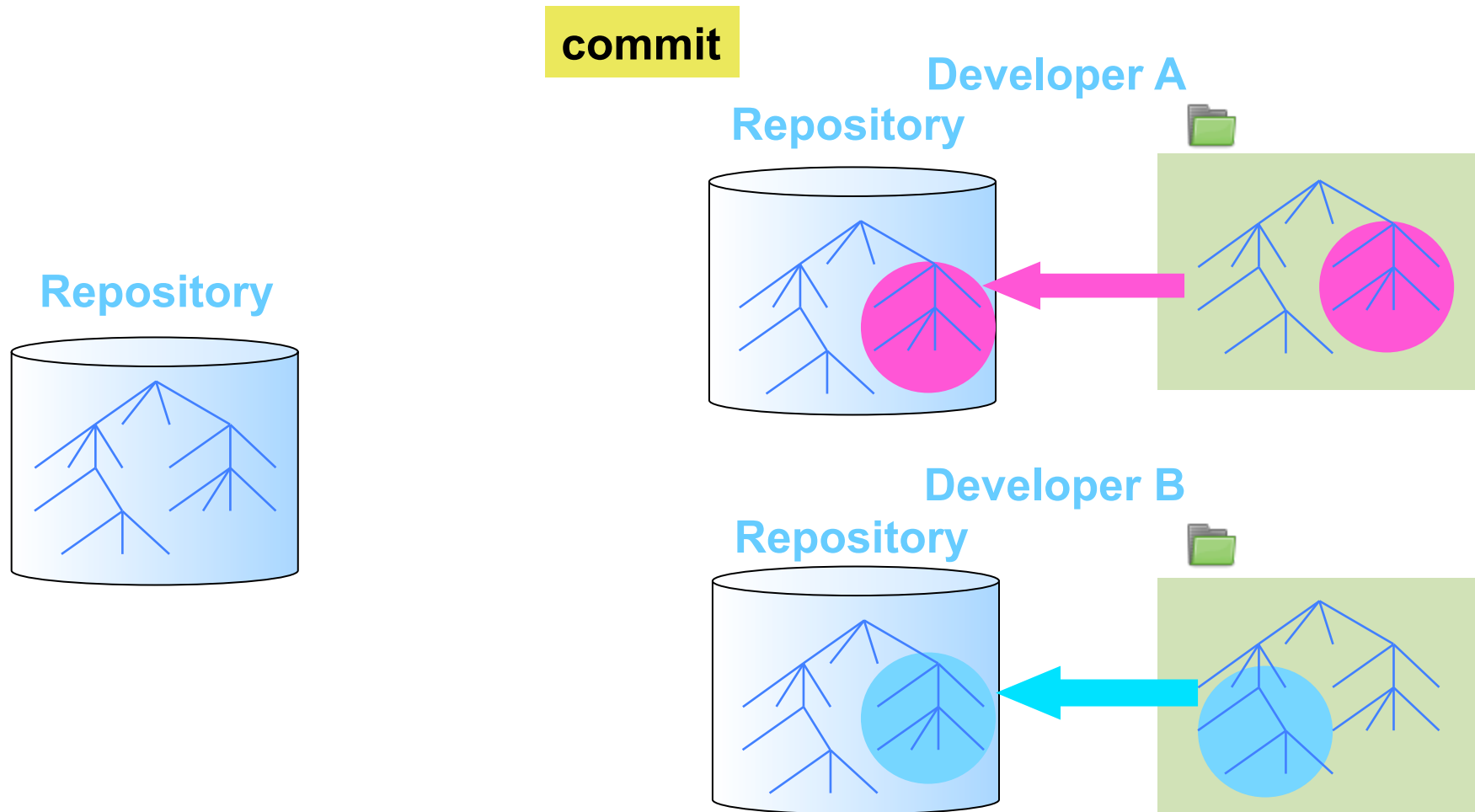
Développement coopératif: dépôt de code décentralisé (Git)

Rappel des processus



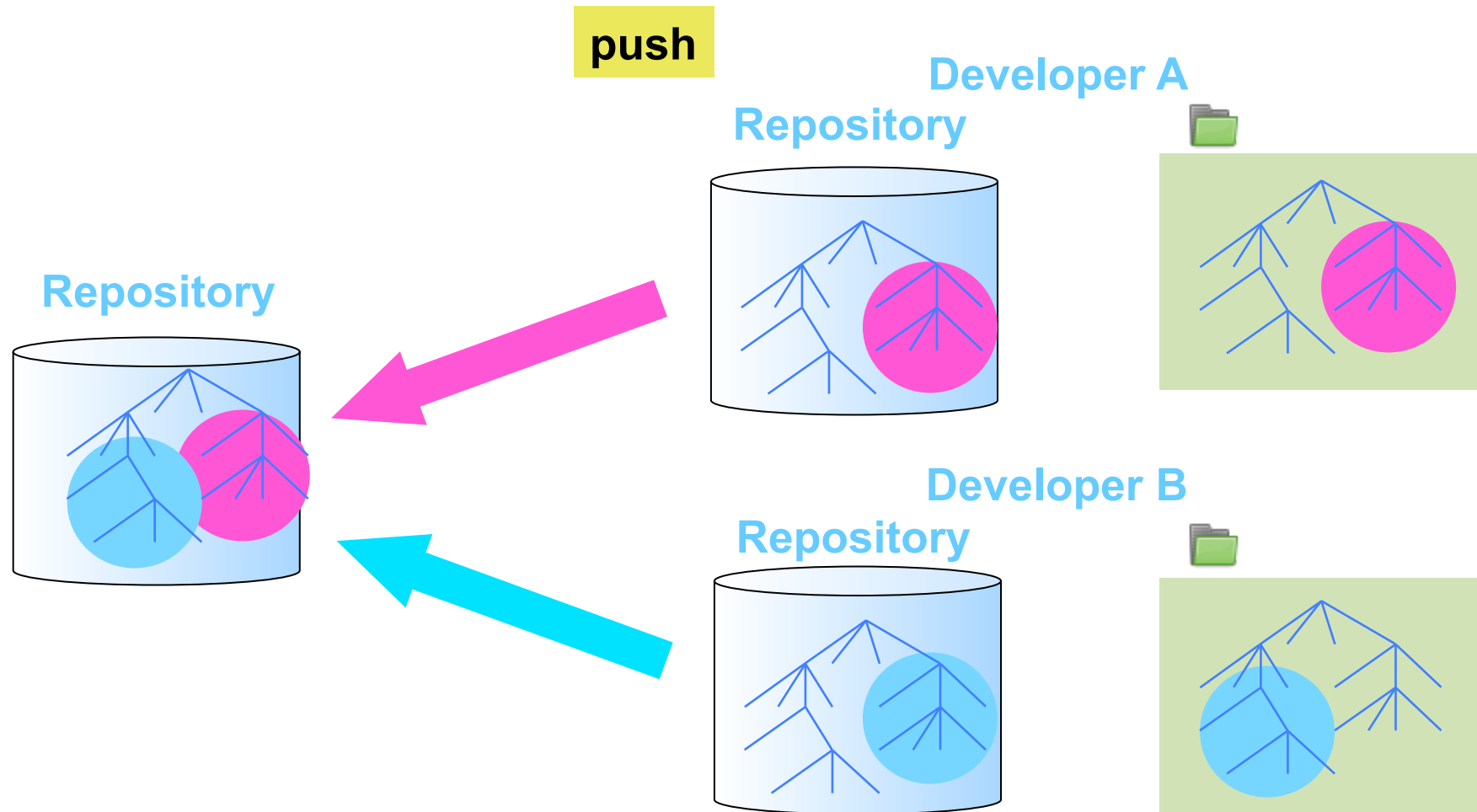
Développement coopératif: dépôt de code décentralisé (Git)

Rappel des processus



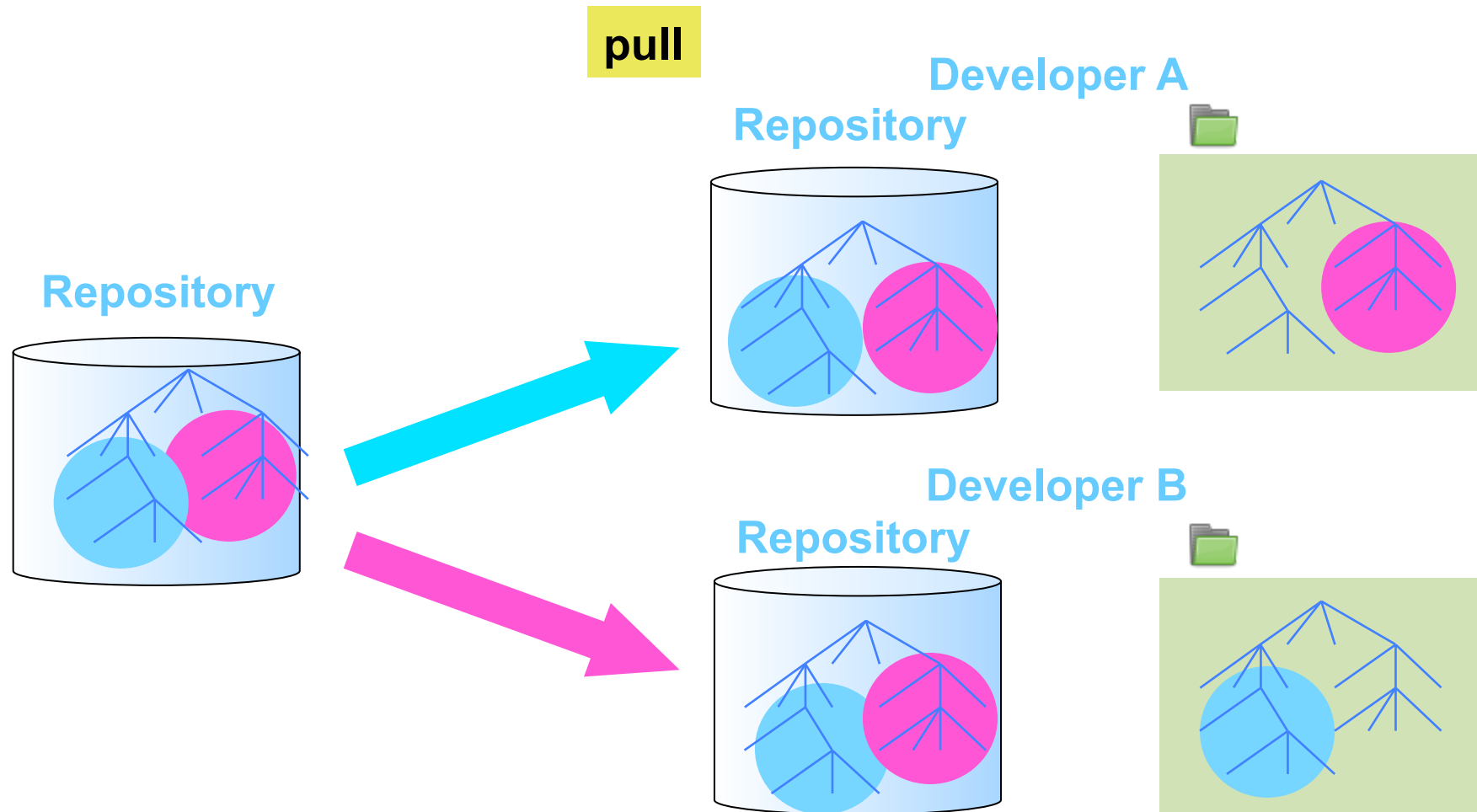
Développement coopératif: dépôt de code décentralisé (Git)

Rappel des processus



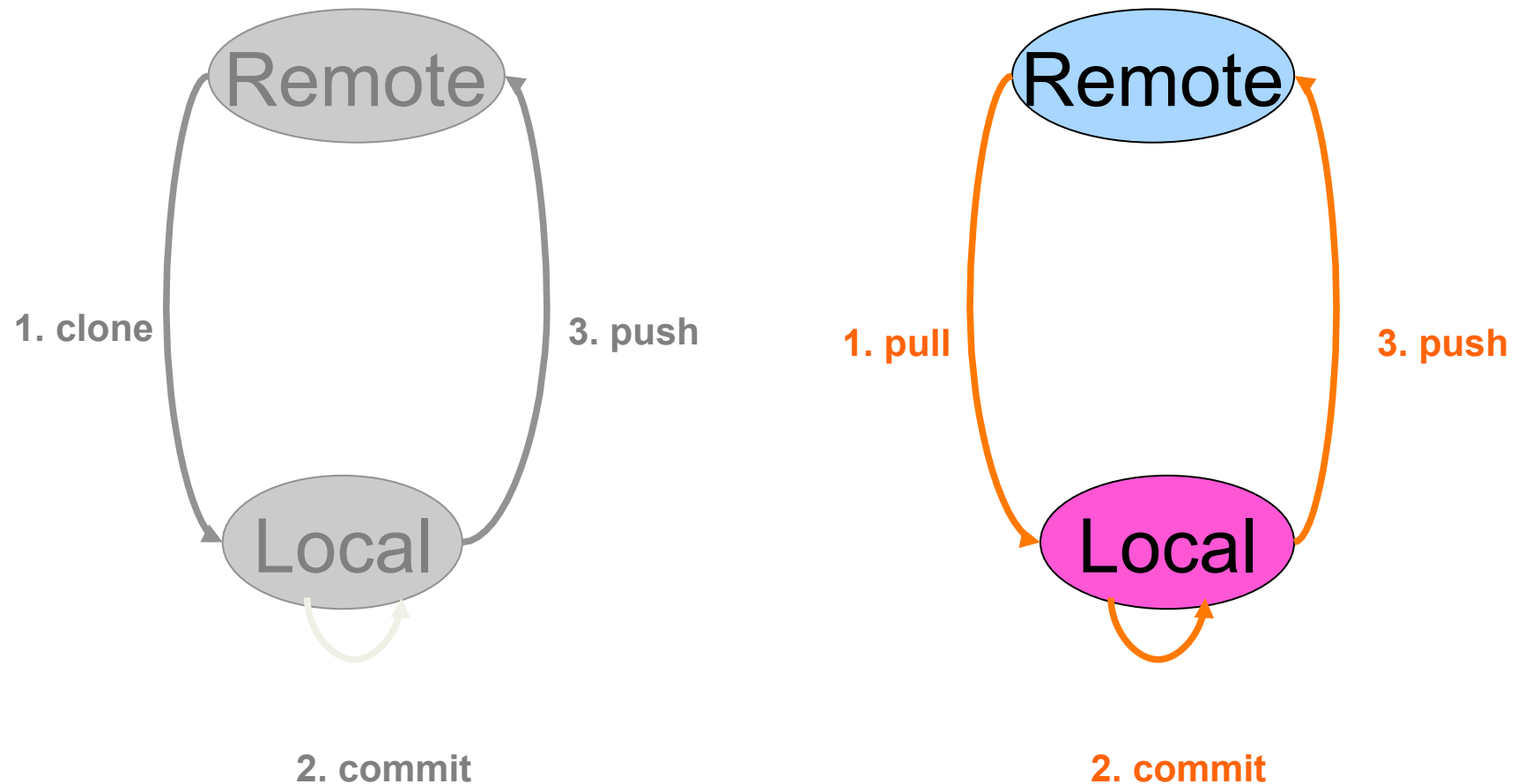
Développement coopératif: dépôt de code décentralisé (Git)

Rappel des processus



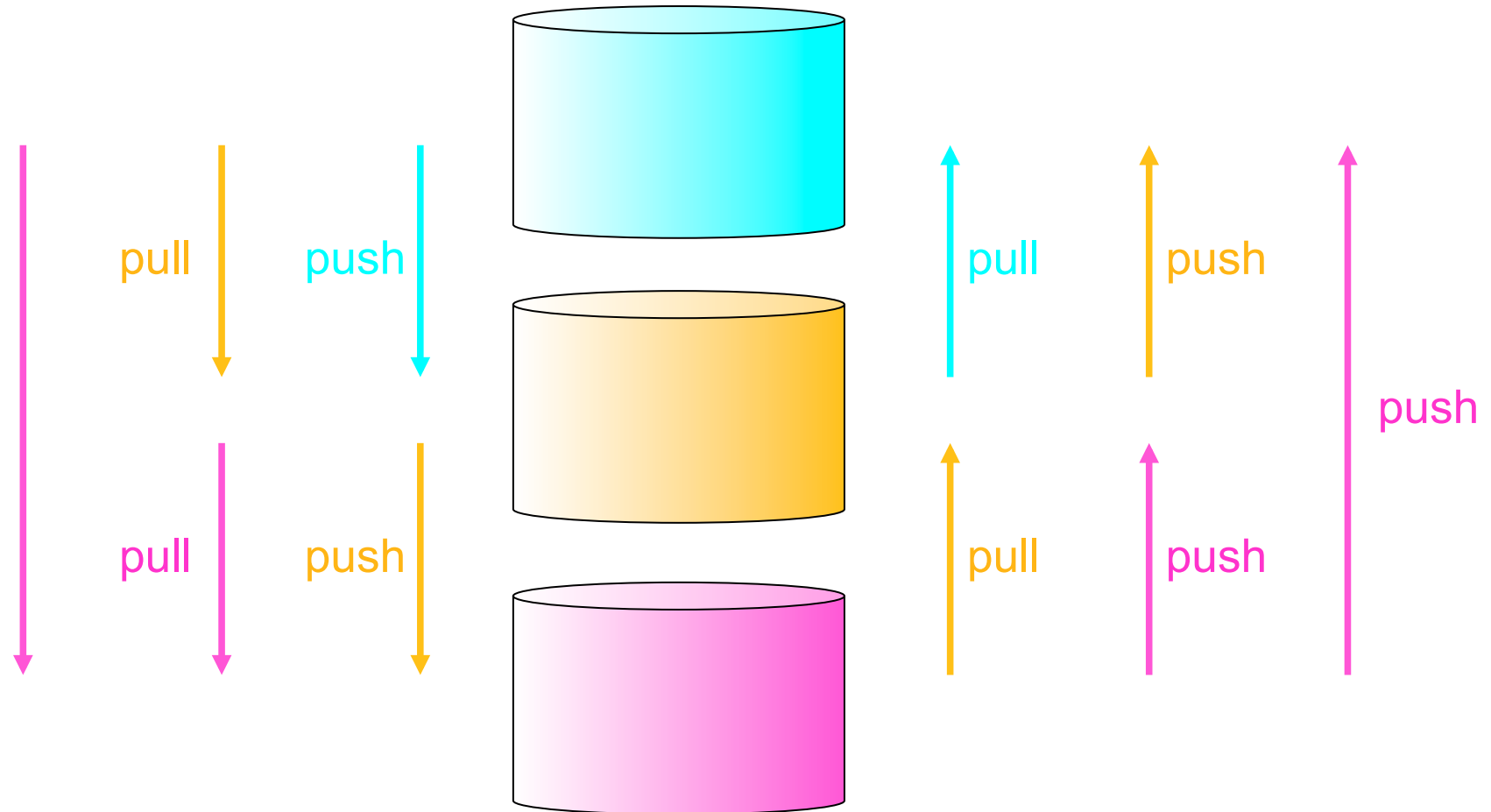
Développement coopératif: dépôt de code décentralisé (Git)

Echanges entre dépôts : classiquement



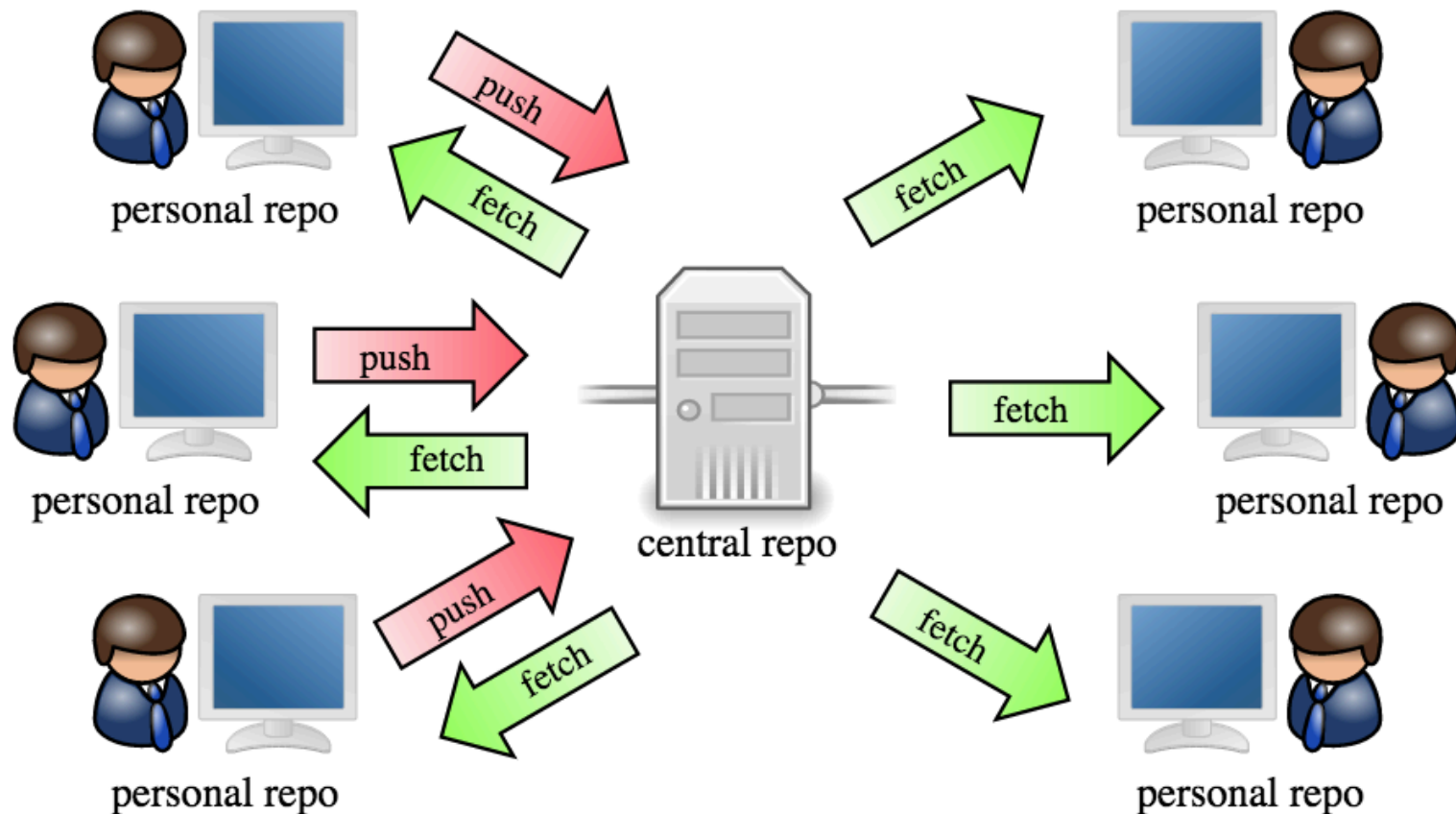
Développement coopératif: dépôt de code décentralisé (Git)

Echanges entre dépôts : tout est possible !



Développement coopératif: dépôt de code décentralisé (Git)

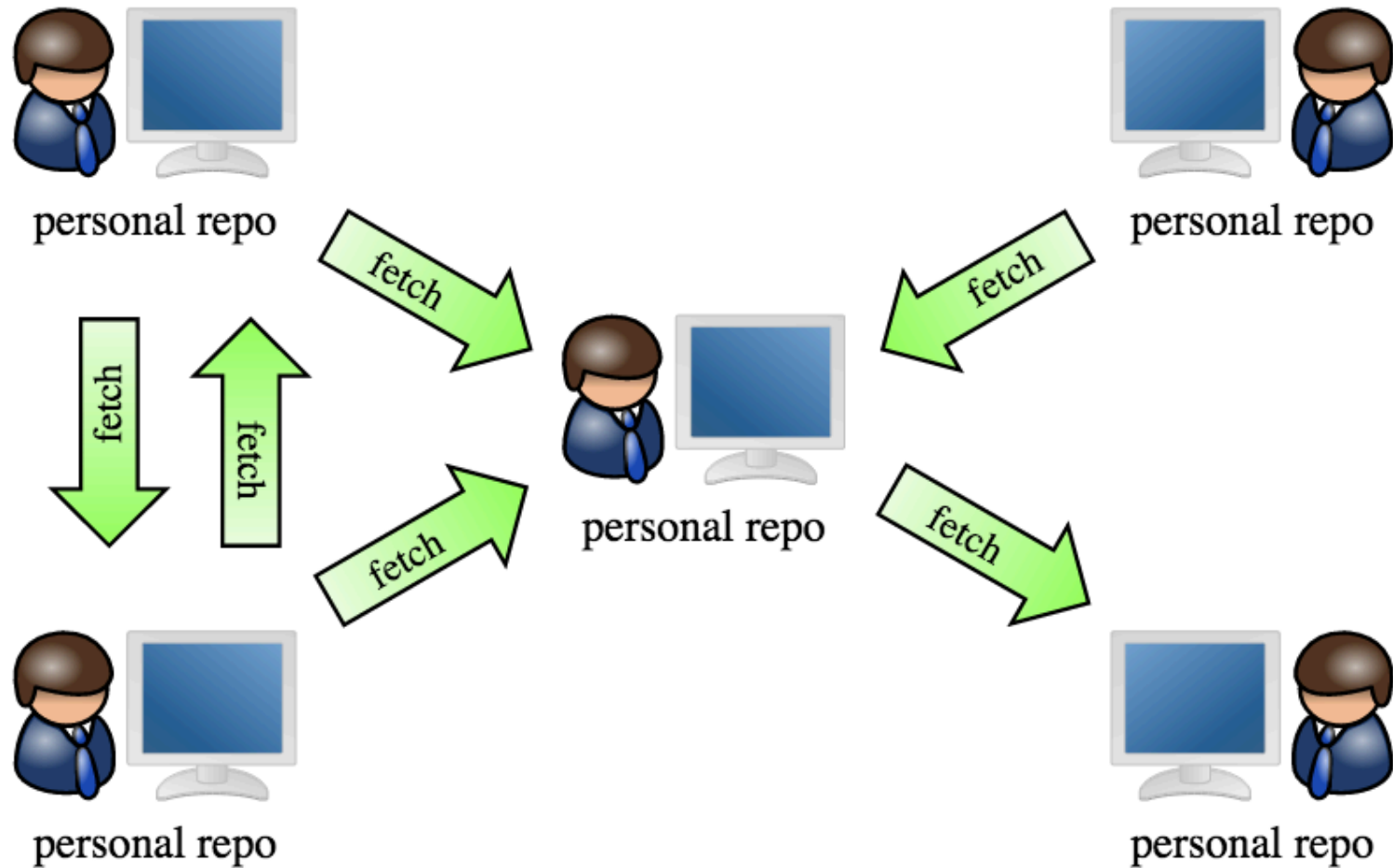
Organisation en étoile



<http://liris.cnrs.fr/~pchampin/enseignement/intro-git/>

Développement coopératif: dépôt de code décentralisé (Git)

Organisation pair-à-pair



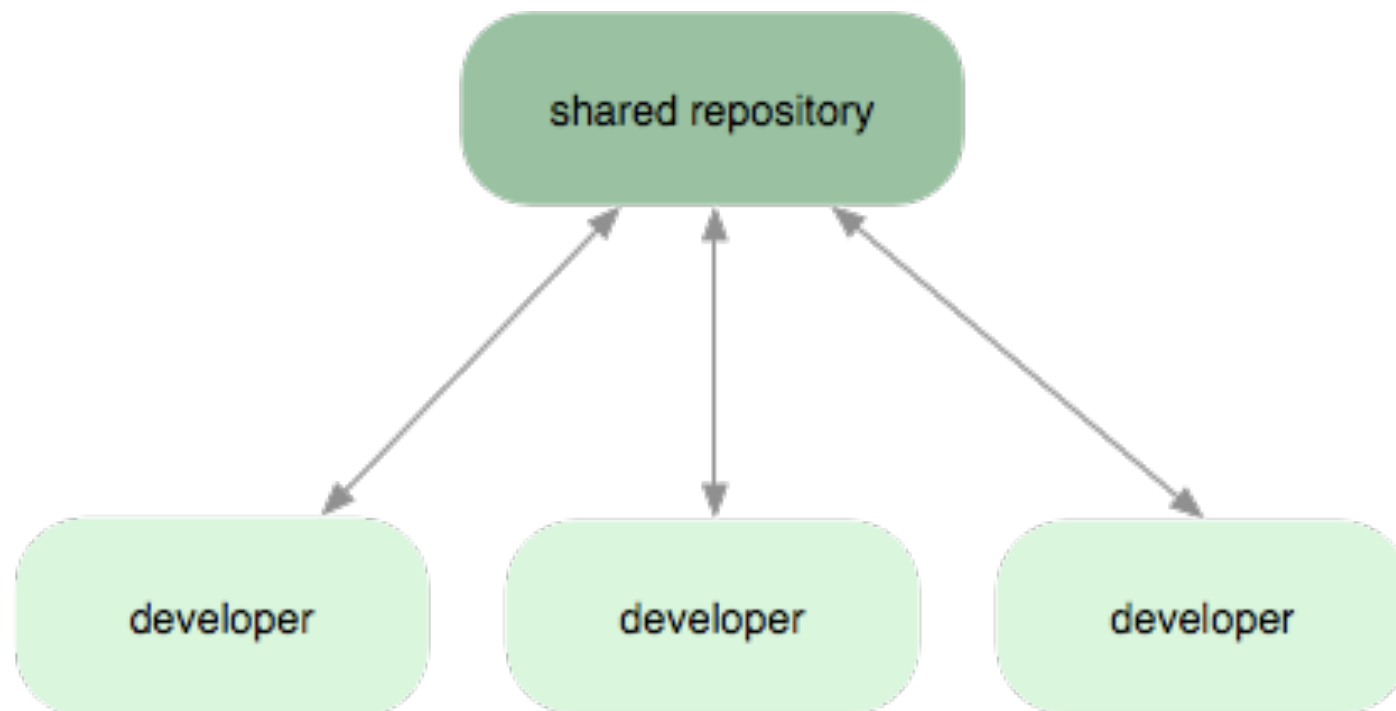
Workflows

Workflows à l'échelle des dépôts

Workflows à l'échelle des dépôts

Centralized workflow

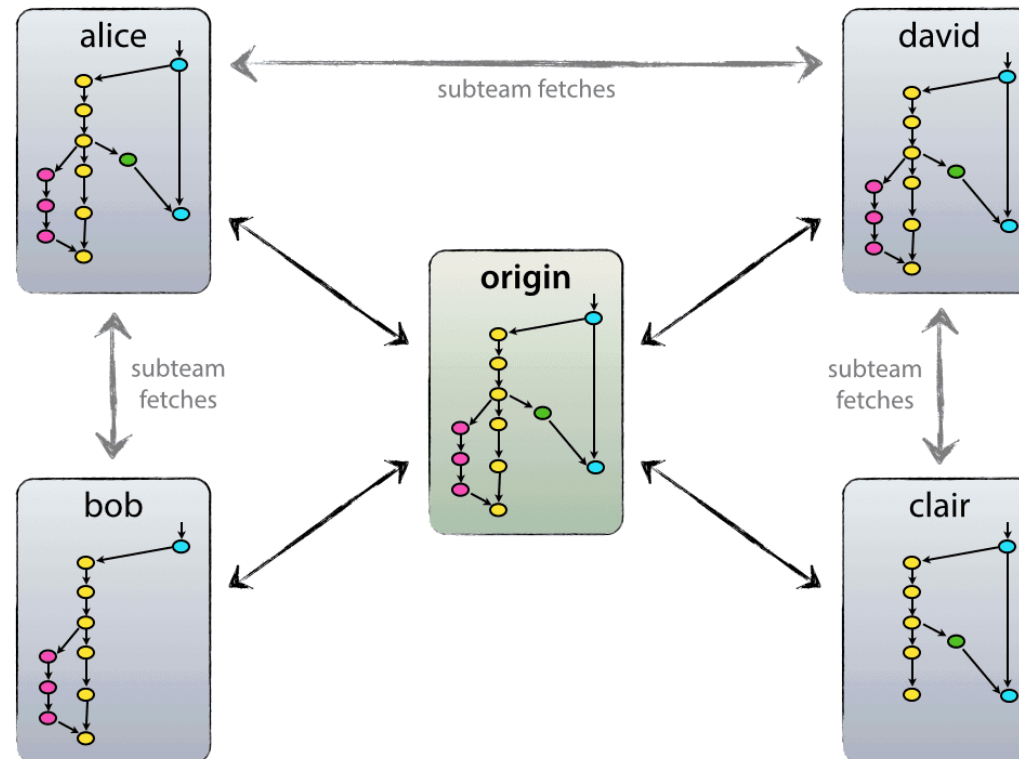
Similar to centralized version control, everybody can edit central repository.



Workflows à l'échelle des dépôts

Centralized workflow

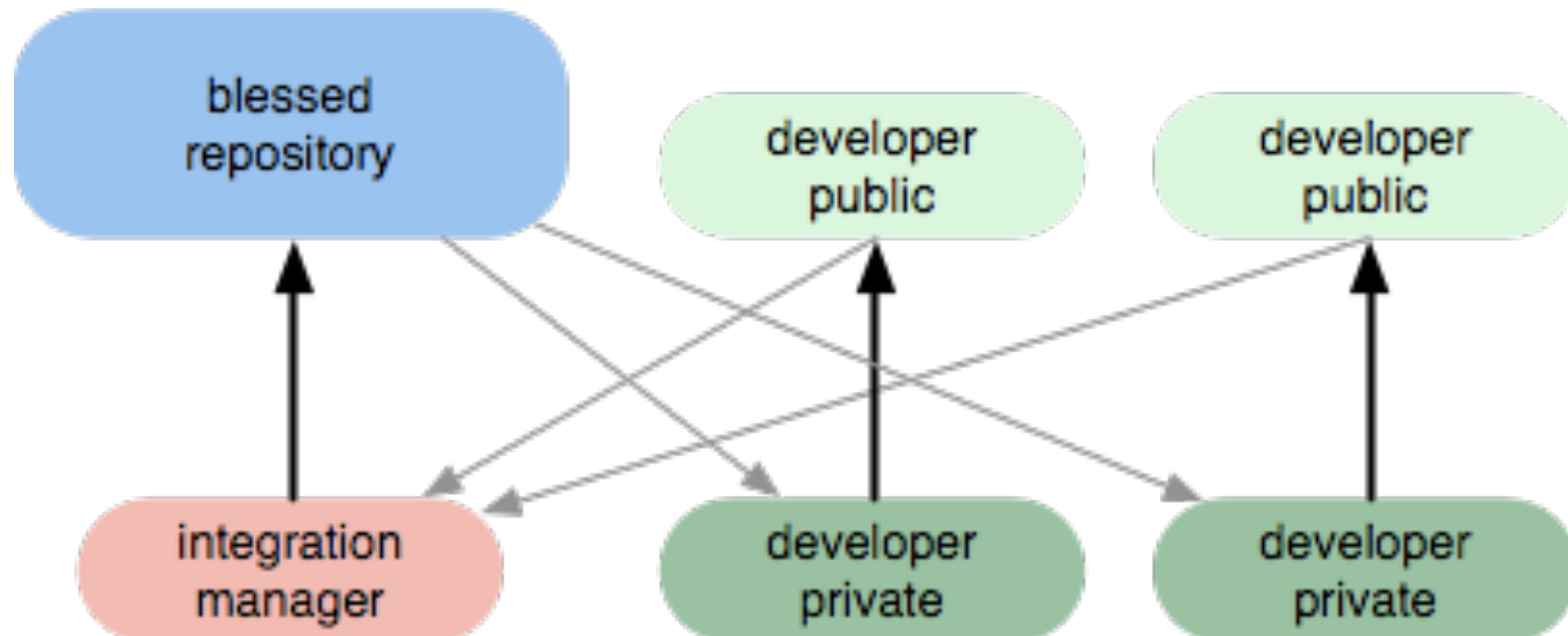
Similar to centralized version control,
everybody can edit central repository.



Workflows à l'échelle des dépôts

Integration manager workflow

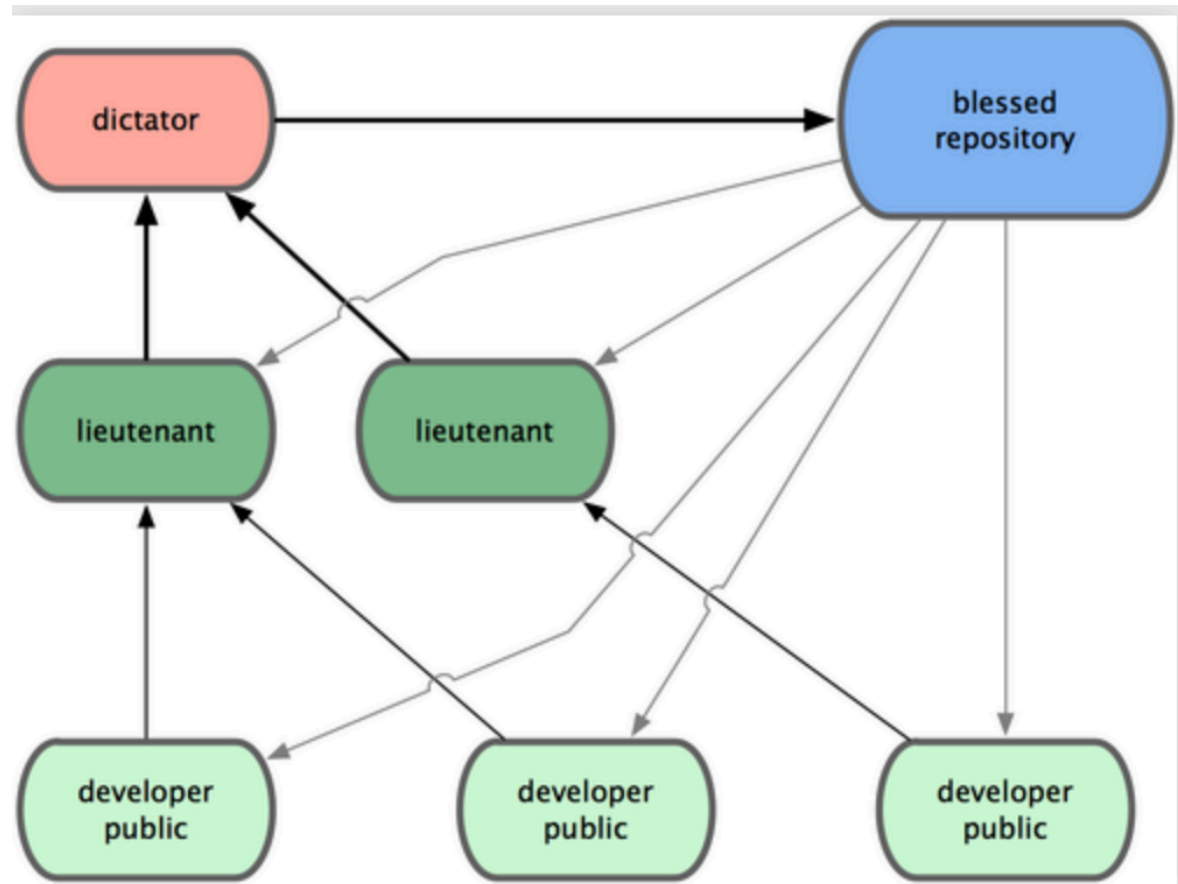
Developer create a request to merge its changes.
Release masters chose which requests to merge.



Workflows à l'échelle des dépôts

Lieutenant workflow

In **huge** projects, lieutenants are **trusted** individuals in charge of reviewing requests.



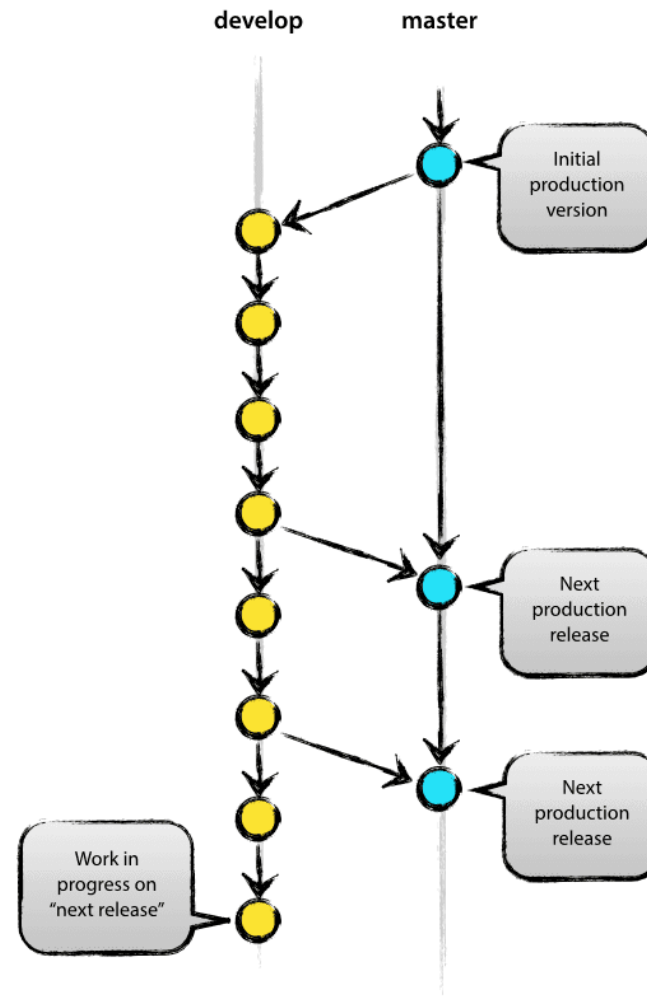
<http://themouette.github.io/slides-git/?theme=clermontech#/8/3>

Workflows à l'échelle des branches

Workflows à l'échelle des branches

Branches principales permanentes

Master
Develop



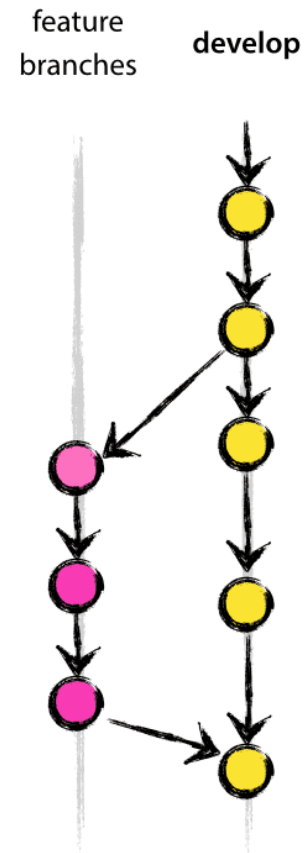
Workflows à l'échelle des branches

Branches support temporaires

Workflows à l'échelle des branches

Branches support temporaires

Feature/topic branches :
workflow branche par fonctionnalité / ticket



Workflows à l'échelle des branches

Branches support temporaires

Release branches

- Préparation de prochaine release

- Correction de bugs mineurs

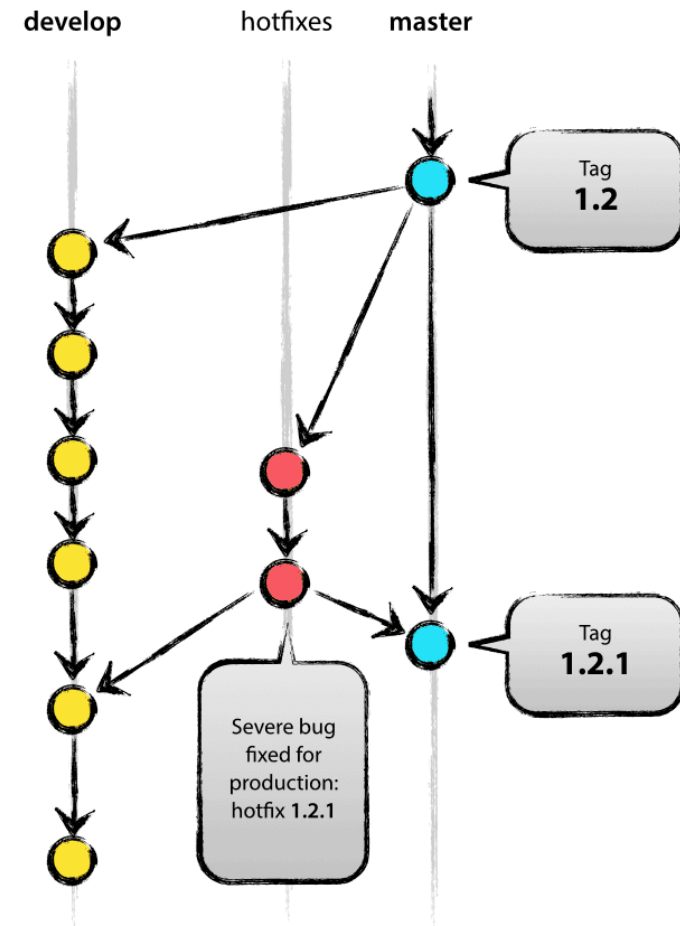
- Préparation des meta-data pour la release

Workflows à l'échelle des branches

Branches support temporaires

Hotfix branches

Résolution d'un bug critique dans une version de production
Le travail sur develop continue



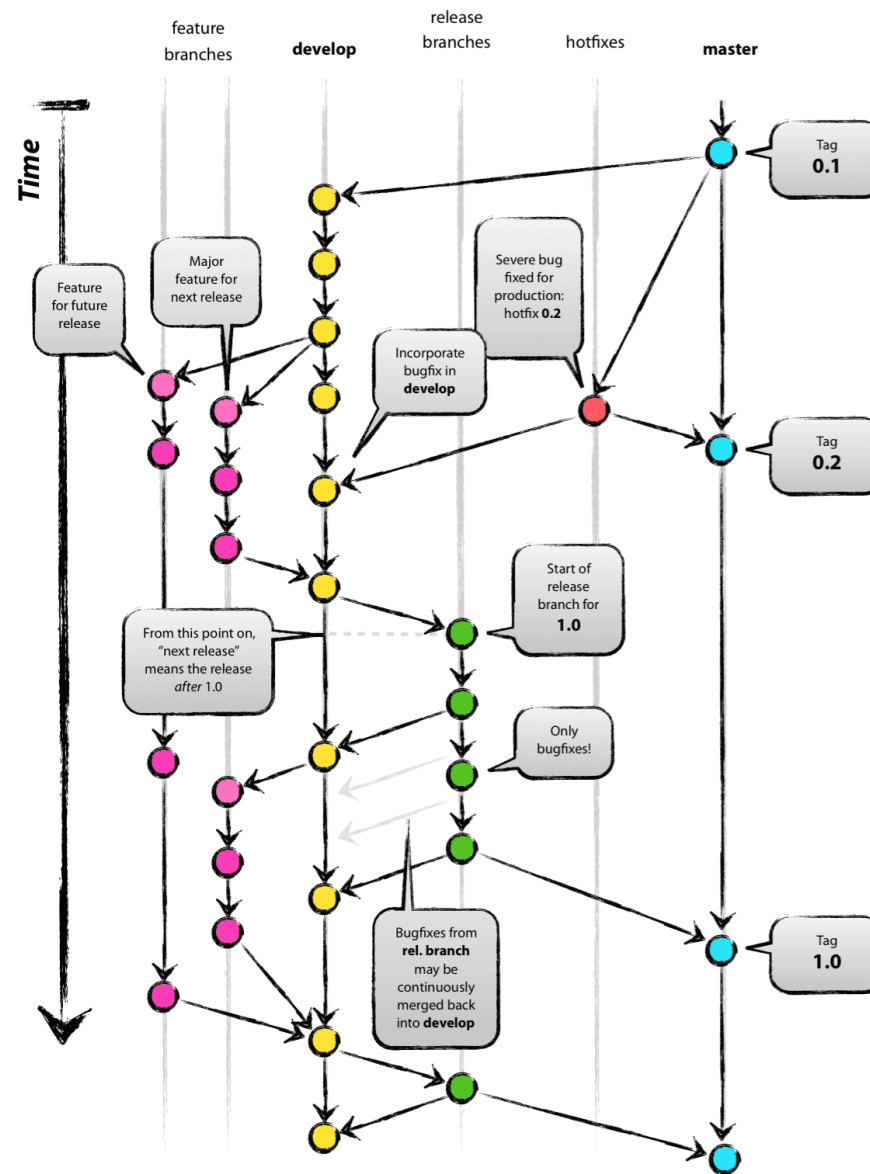
Workflows à l'échelle des branches

Synthèse : workflow gitflow

(Vincent Driessen)

Branches master / develop, release-v*

Branches de livraison versus maintenance

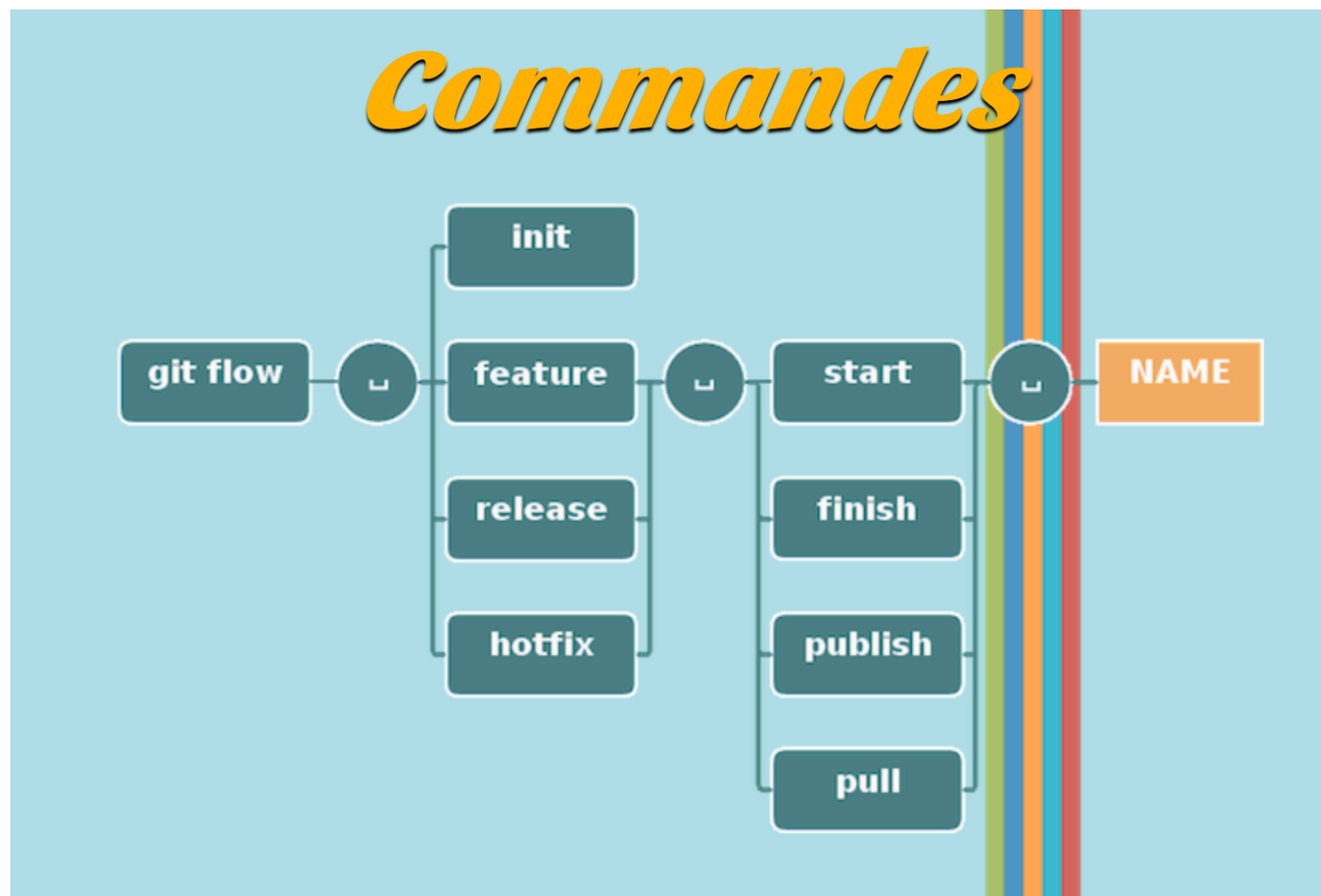


Source :

<http://nvie.com/posts/a-successful-git-branching-model/>

Workflows à l'échelle des branches

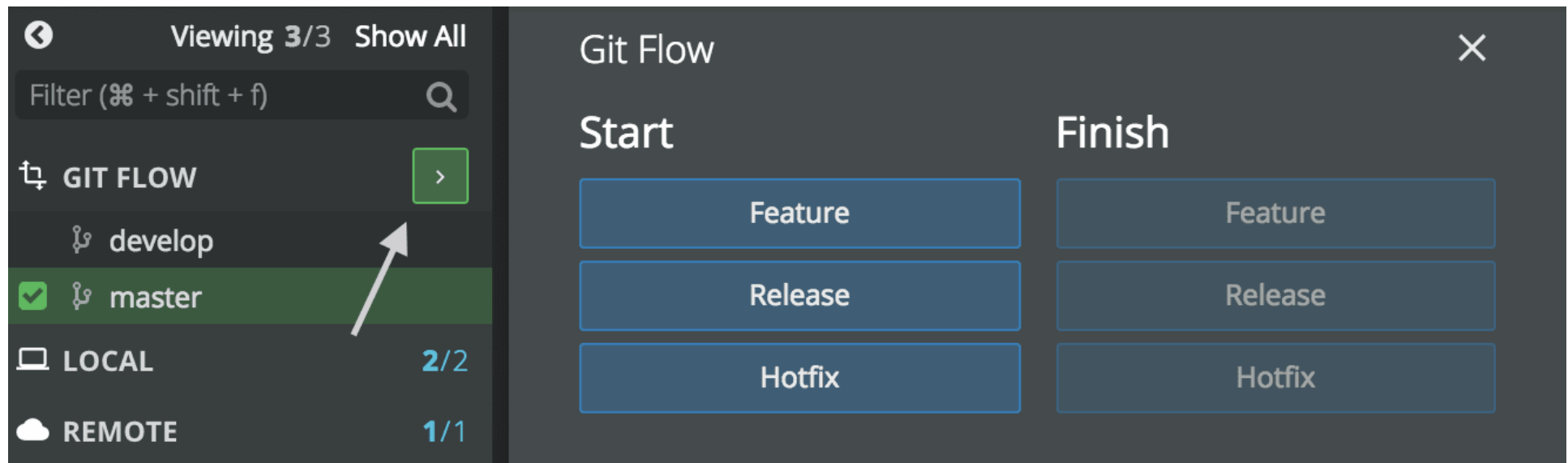
git-flow est un ensemble d'extensions git permettant des opérations de haut niveau sur un dépôt pour appliquer le modèle de branches de Vincent Driessen.



https://danielkummer.github.io/git-flow-cheatsheet/index.fr_FR.html

Workflows à l'échelle des branches

git-flow dans GitKraken



<https://support.gitkraken.com/repositories/git-flow>

Workflow de duplication

Workflow de duplication

Fork

Clone en local

Développement d'une nouvelle fonctionnalité

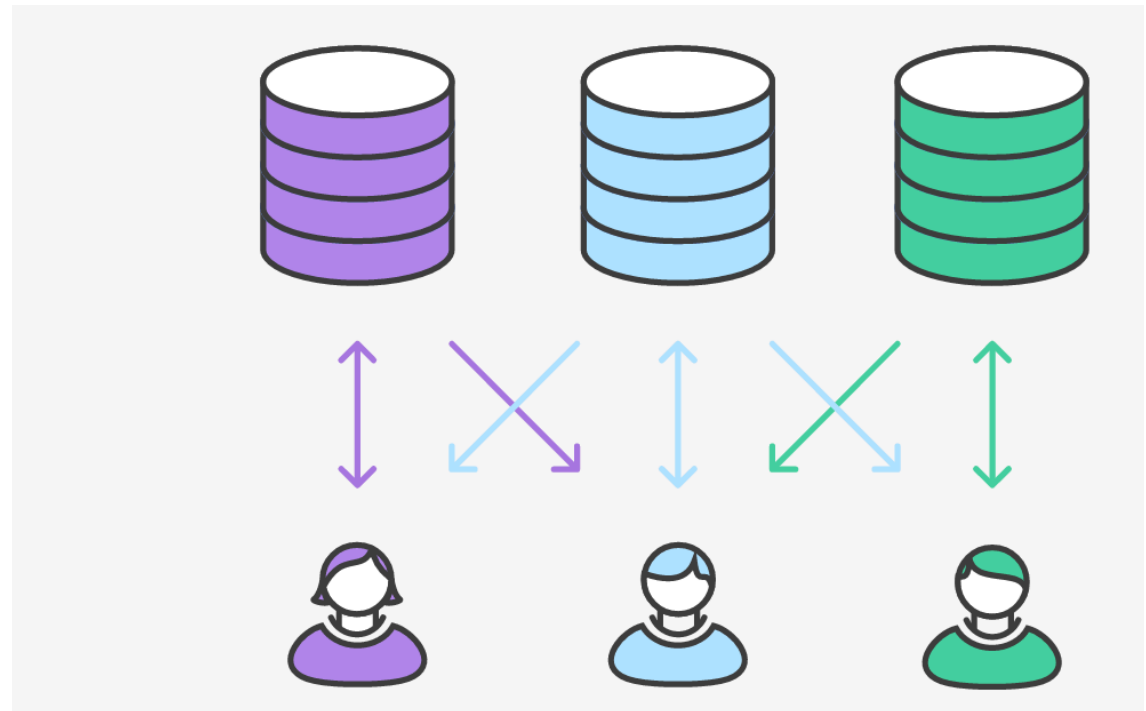
Push

Création de la **pull request** / merge request

Review

Commit de suivi

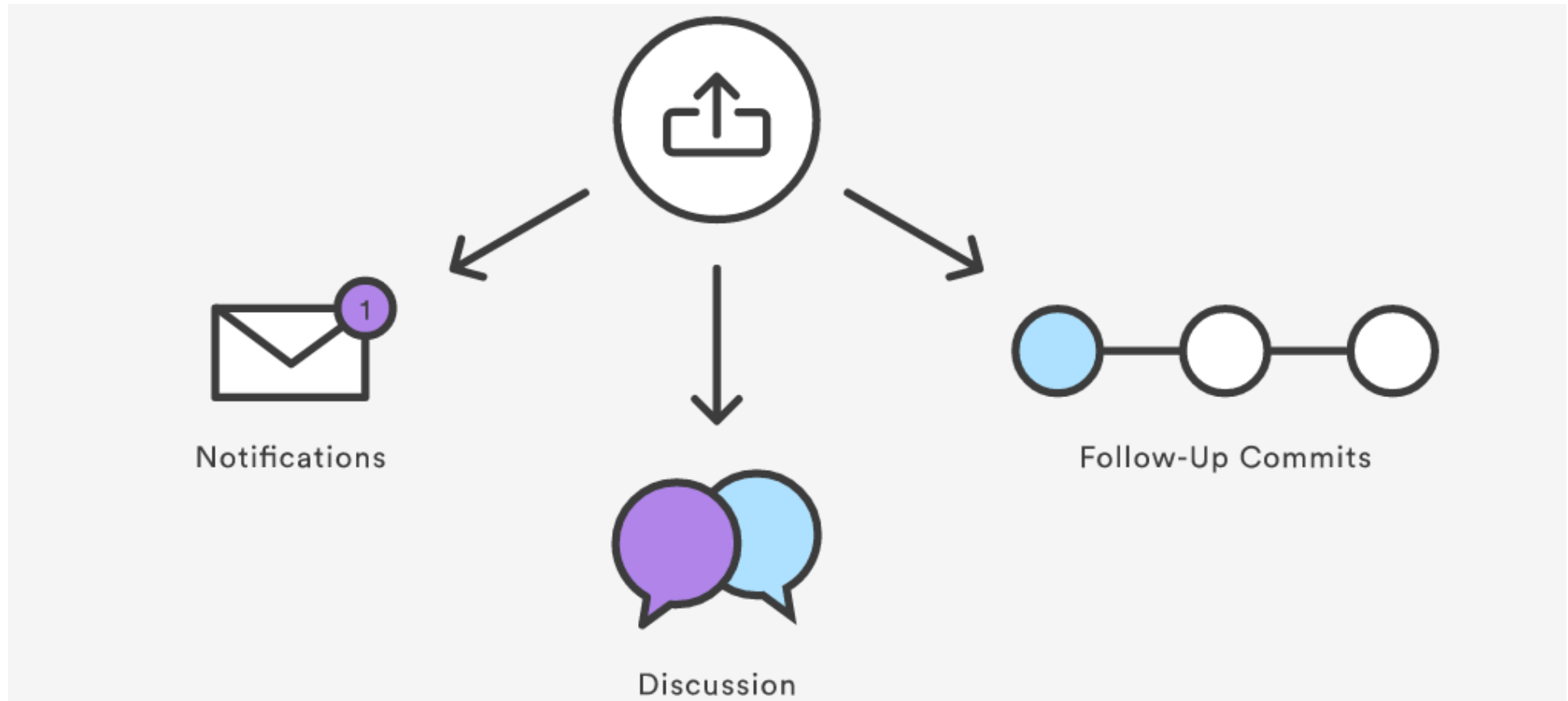
Acceptation de la pull request



Outil puissant pour équipes peu soudées

Workflow de duplication

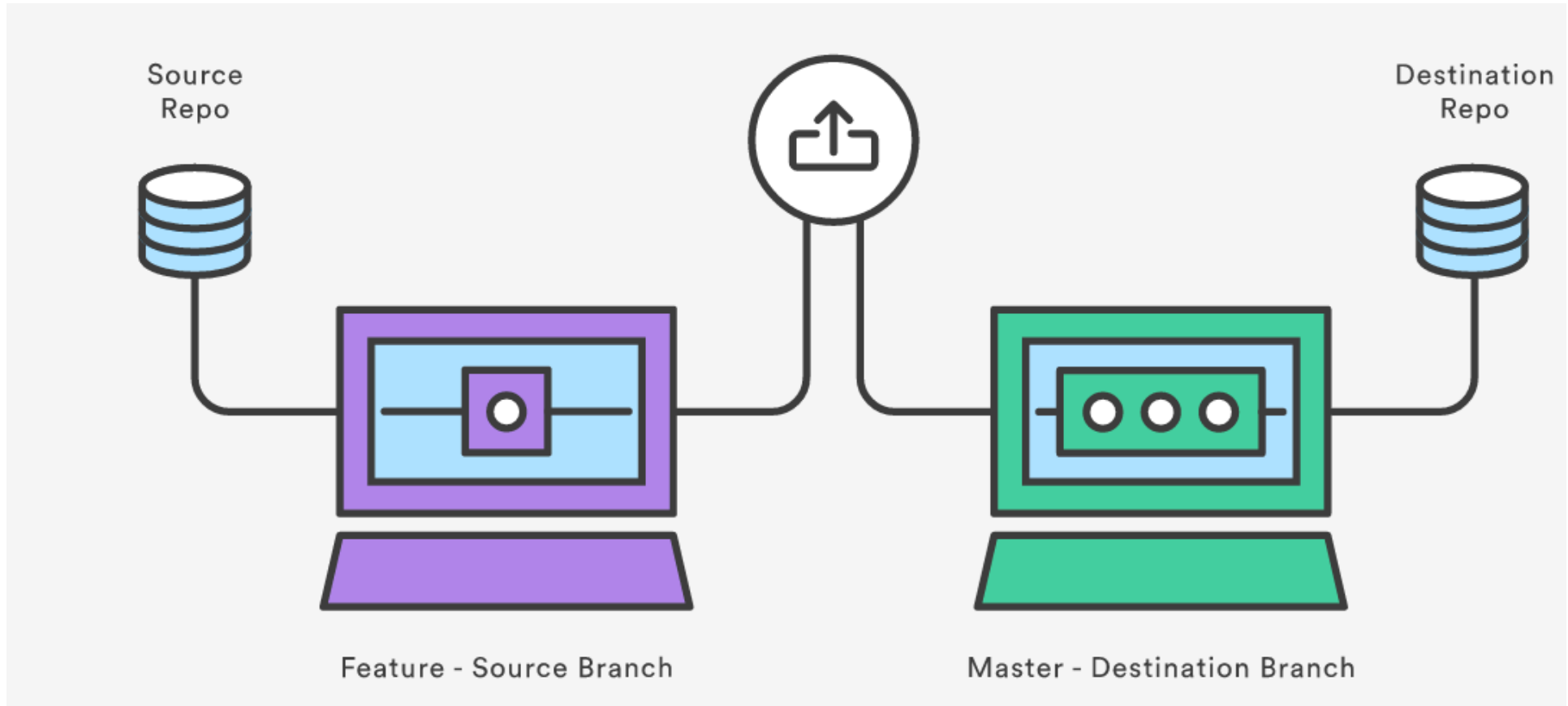
Pull request



<https://fr.atlassian.com/git/tutorials/making-a-pull-request>

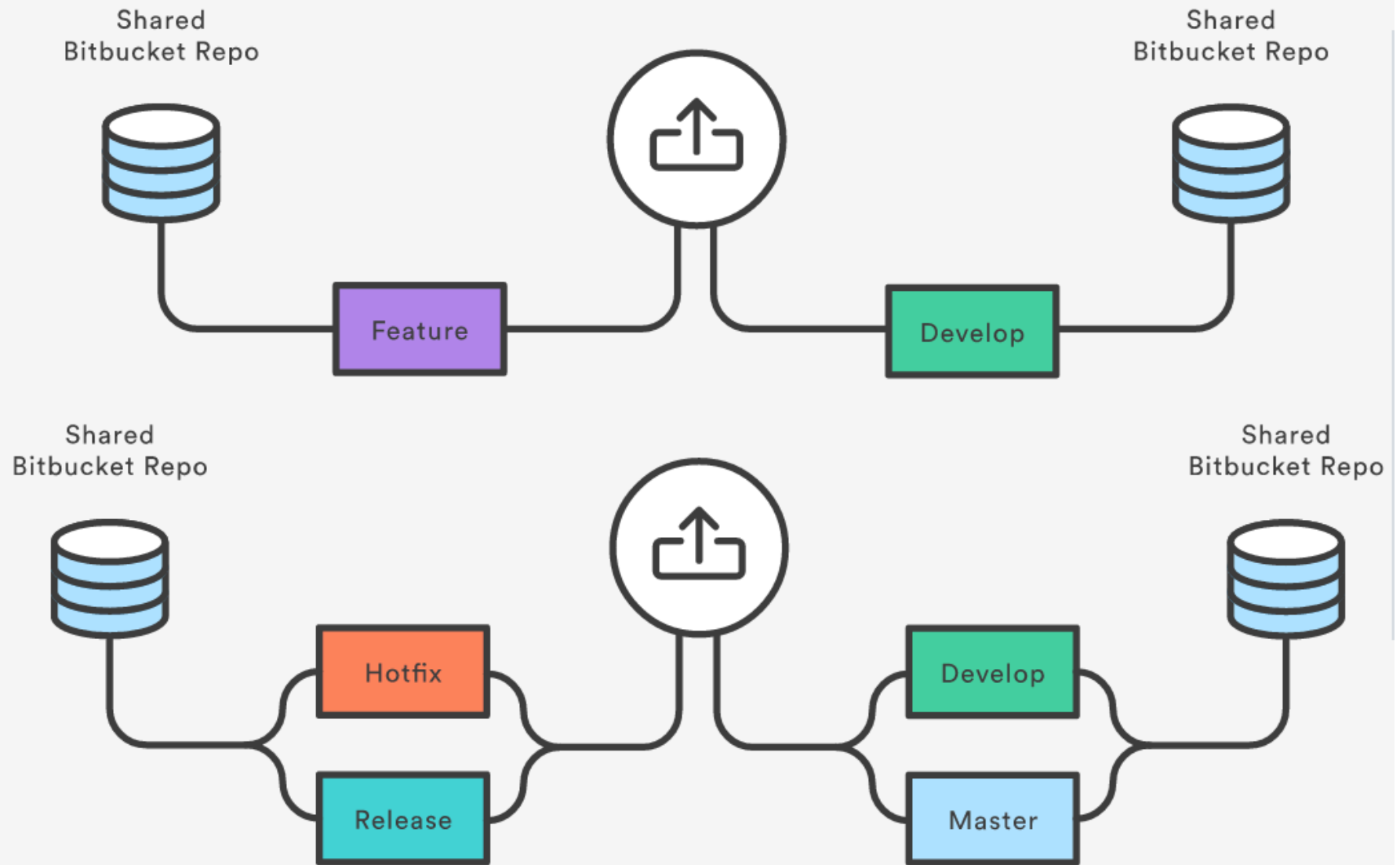
Workflow de duplication

Pull request : structure

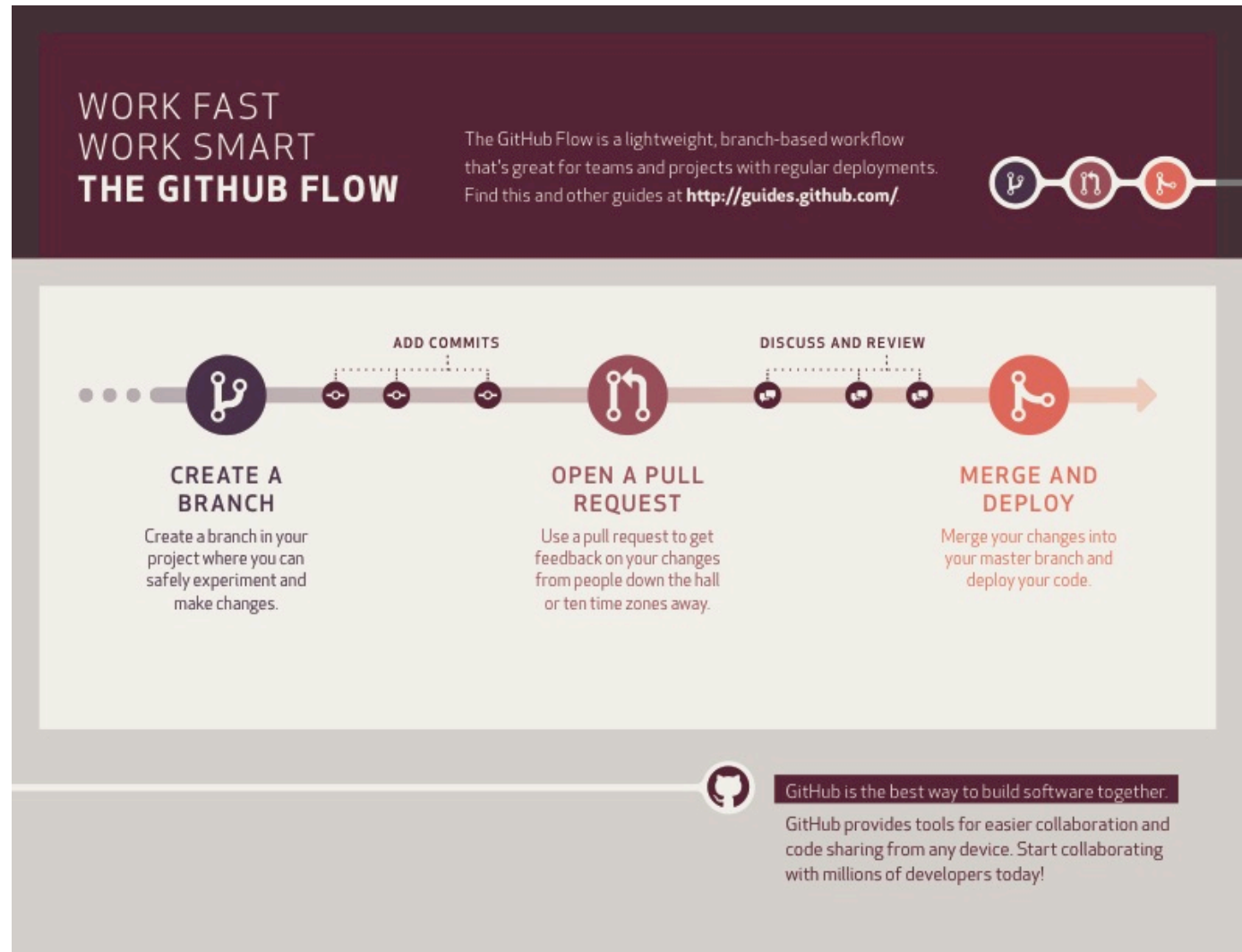


Workflow Gitflow avec pull requests

Workflow Gitflow avec pull requests



Workflow Github flow



<https://guides.github.com/introduction/flow/>

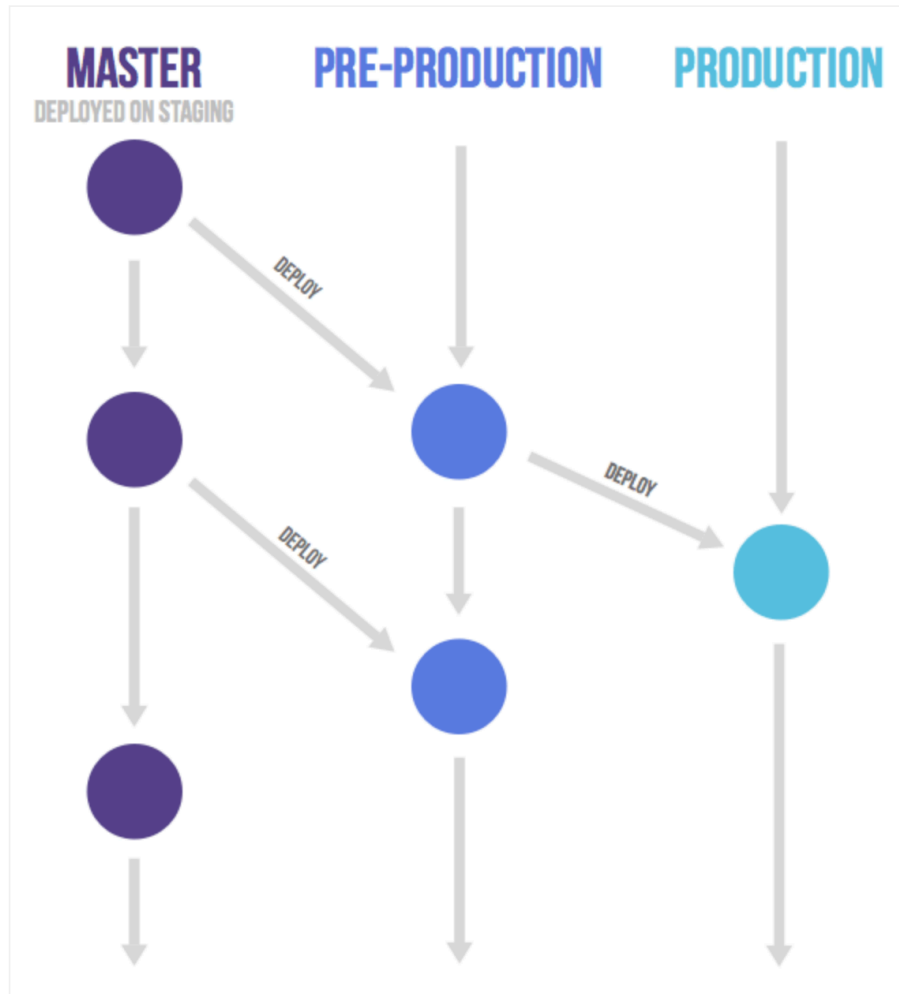
Workflow Github flow

GitHub flow as a simpler alternative

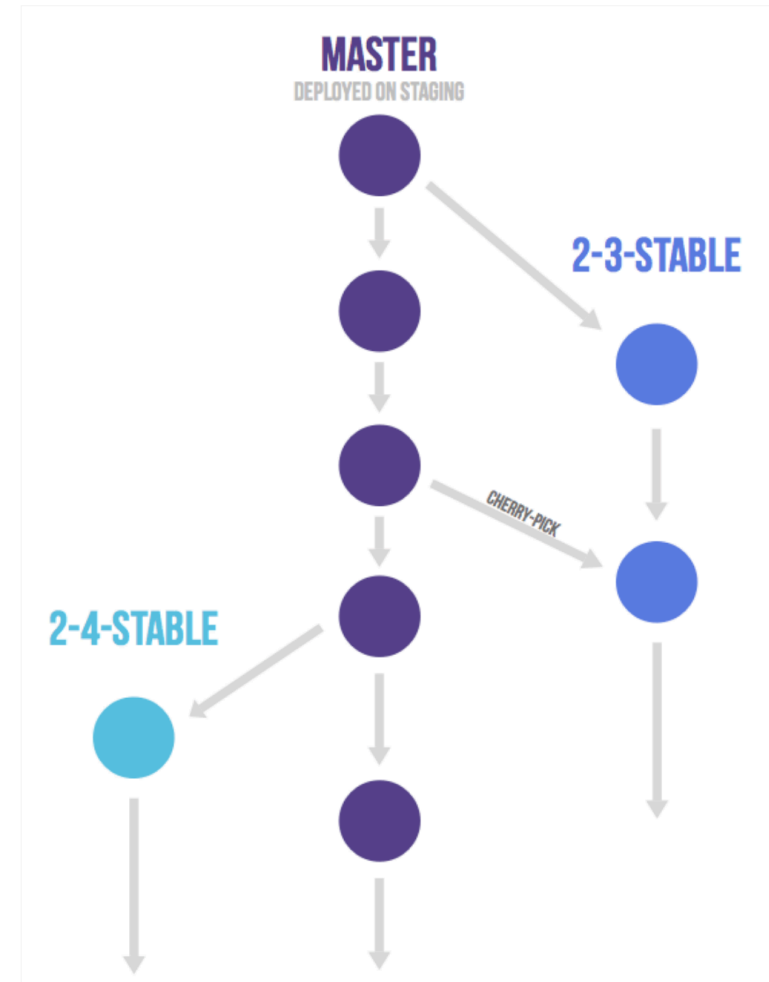


Workflow Gitlab flow

Environment branches with GitLab flow



Release branches with GitLab flow



Références

<http://nvie.com/posts/a-successful-git-branching-model/>

<https://fr.atlassian.com/git/tutorials/comparing-workflows>

https://docs.gitlab.com/ce/workflow/gitlab_flow.html

https://forge.cnrs.fr/git/guide_d_utilisation_GIT-branchements.pdf

11 rules of Gitlab flow :

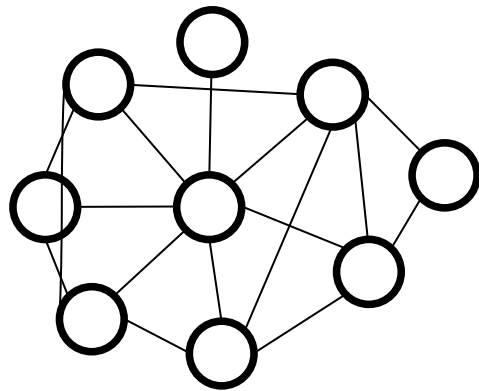
<https://about.gitlab.com/pdfs/the-eleven-rules-of-gitlab-flow.html>

git-flow cheatsheet :

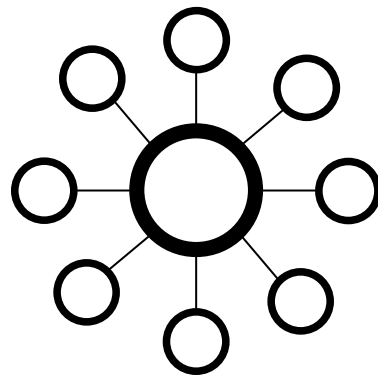
https://danielkummer.github.io/git-flow-cheatsheet/index.fr_FR.html

Conclusion

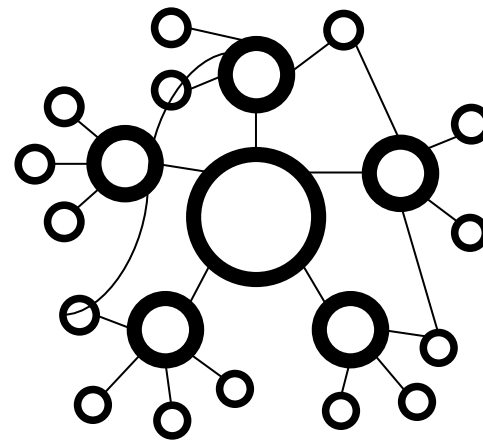
- **A decentralized SCM remains a tool**
 - No default usage policy
 - Policy to be defined
 - From centralized to decentralized
 - Pull-only vs shared-push



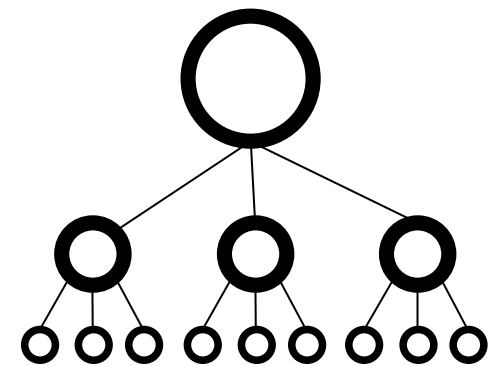
Anarchic



Centralized



Linux kernel model



Branch by functionality
or
Release train

Conclusion

Choix d'un workflow dépend :

niveau d'autonomie que l'on souhaite donner aux développeurs?

taille de la structure?

comment situés géographiquement?

processus automatisés? (ex: vérification de syntaxe)

temps disponible par les reviewers?

Les workflows présentés ici ne sont que des exemples de ce qui est possible.

Ce ne sont pas des règles immuables sur l'utilisation de Git au travail.

N'hésitez pas à adopter certains aspects d'un workflow et à en ignorer d'autres

L'objectif est que Git travaille pour vous, pas l'inverse !

Un moment pour échanger !

Et toi, tu fais comment pour collaborer ?!?

Quel workflow ?

Vos pratiques ?

Vos retours d'expérience ?

Des erreurs à ne pas faire / des conseils ?

Des compléments à ma présentation ?