

Introduction à Git

Ayoub EL HALI, ayoub.elhali@soprasteria.com

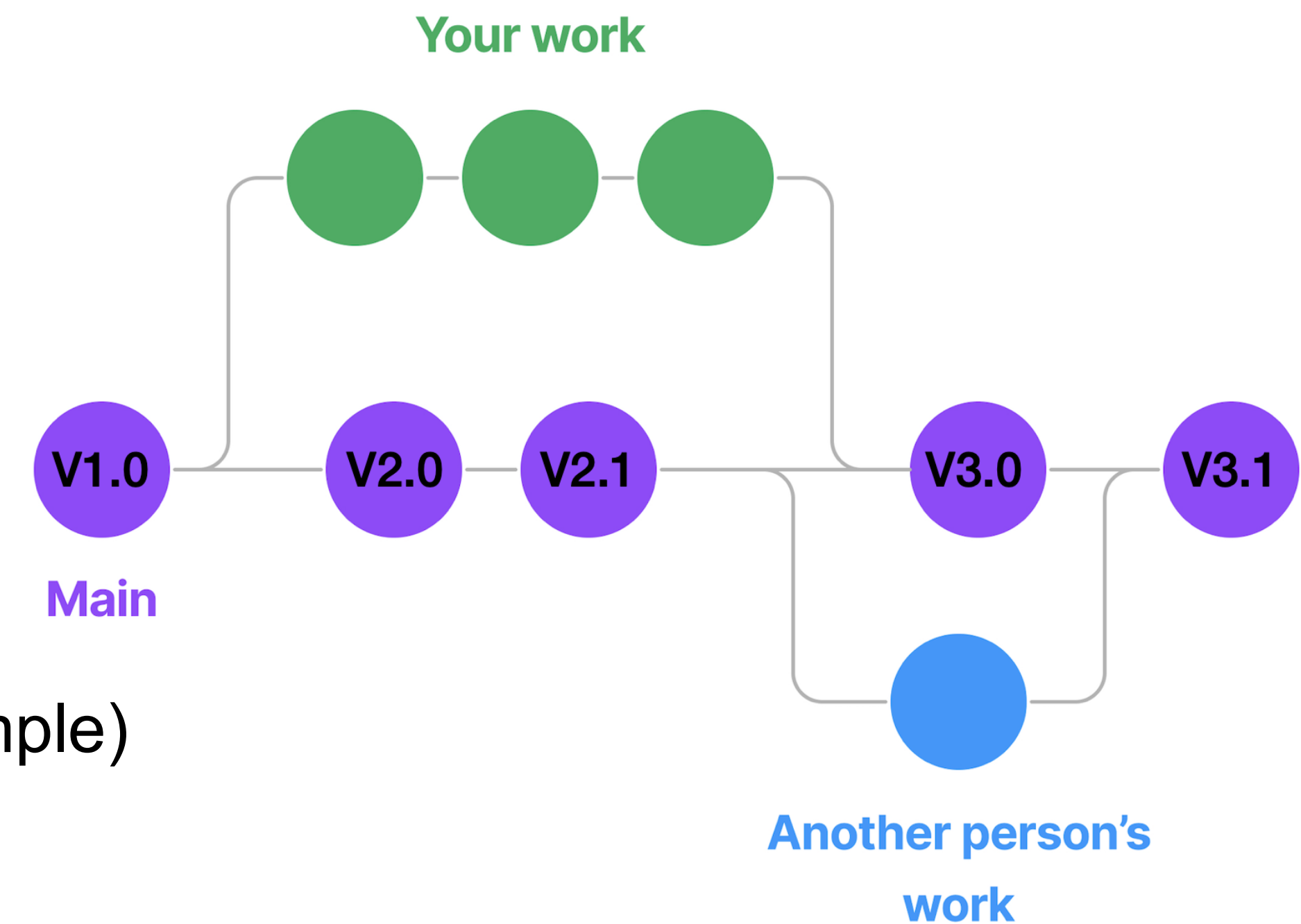
Format

- Cours 1 : 18 décembre
 - Gestionnaires de versions
 - Git (CLI, GUI)
 - GitLab
- Cours 2 : 8 janvier
 - TP
- Examen : 22 janvier

Gestionnaires de versions

Gestionnaires de versions (version control)

- Système basé sur le suivi de versions de fichiers (texte, programme, images, documents...)
- Avantages :
 - Travailler à plusieurs en parallèle sur un même projet tout en ayant un dépôt de fichiers
 - Créer et avoir accès à un historique des modifications
 - Faciliter la résolution des anomalies
 - Avoir accès aux anciennes versions de fichiers
 - Voir seulement ce qui a été modifié entre 2 versions
 - Revenir facilement et rapidement à un état antérieur
 - Indépendant des pannes d'un PC (perte de données par exemple)
 - Travailler hors ligne reste toujours possible



Centralisé ou décentralisé

- 2 types : centralisé ou décentralisé
 - Mode Centralisé : un serveur central est le dépôt de référence. Les membres du projet ne peuvent pas se communiquer du code en l'absence du dépôt central.
 - Mode Décentralisé : chaque membre du projet possède une copie complète des fichiers et de leur historique, et peut travailler et communiquer aux autres sans dépendre d'un point central.

Quelques logiciels de gestion de version

Centralisé :

- Subversion (SVN)
- Concurrent Versions System (CVS)

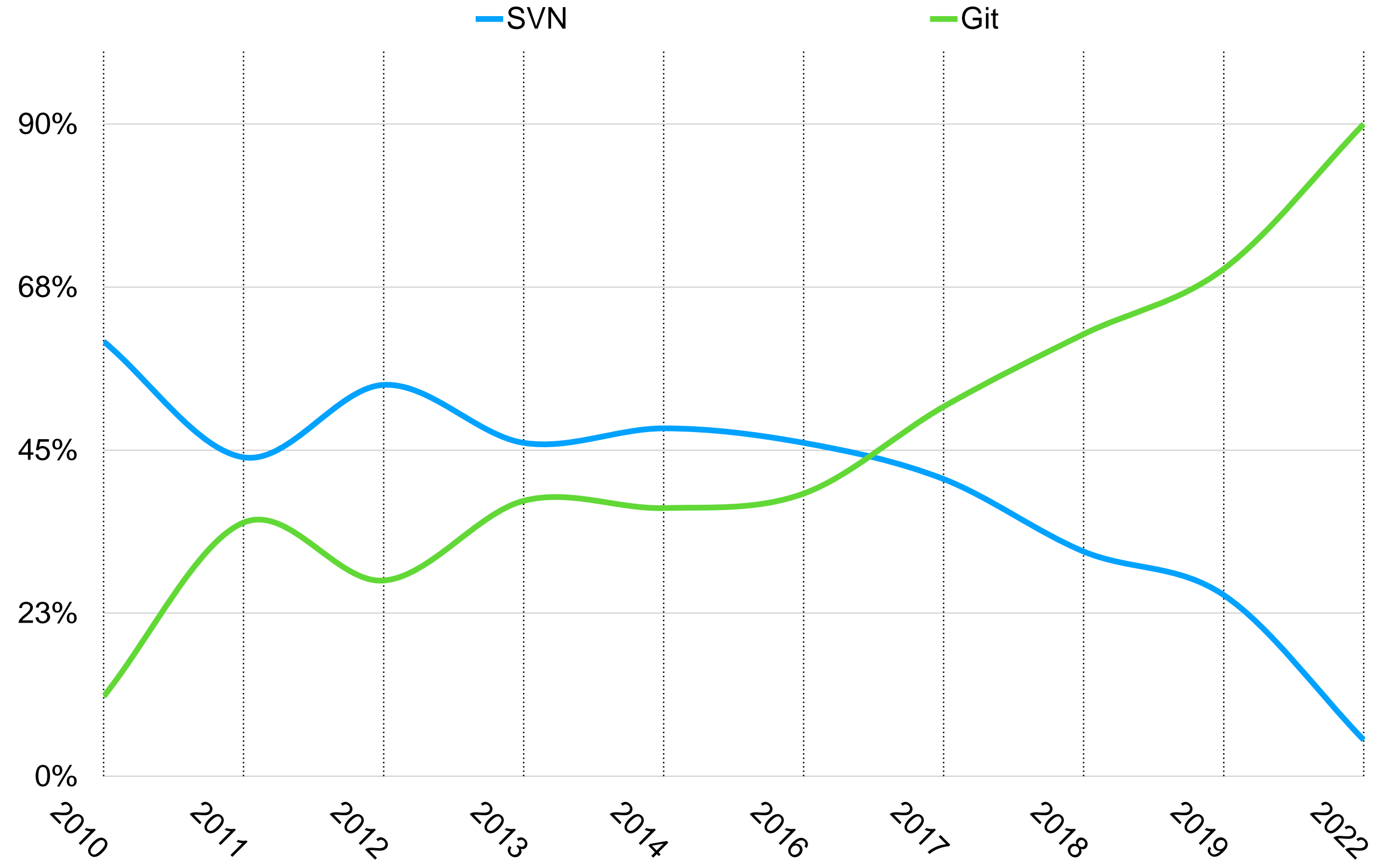


Décentralisé :

- Git
- Mercurial



Git vs SVN



Pourcentage d'utilisation de Git et SVN au sein de projets informatiques

Git vs SVN

	Git	SVN
Versionning	Décentralisé	Centralisé
Dépôt	Des copies de dépôt, présentes localement, dans lesquelles il est possible de travailler	Un dépôt central dans lequel les copies de travail sont créées
Droit d'accès	Pour le répertoire complet	Basé sur le chemin
Suivi des modifications	Enregistre des données	Enregistre des données
Journal de modifications des données	Le dépôt et les copies de travail contiennent l'historique complet.	Complet seulement dans le dépôt. Les copies de travail ne contiennent que la version la plus récente.
Connectivité au réseau	Nécessaire seulement pour réaliser une synchronisation	Pour tous les accès

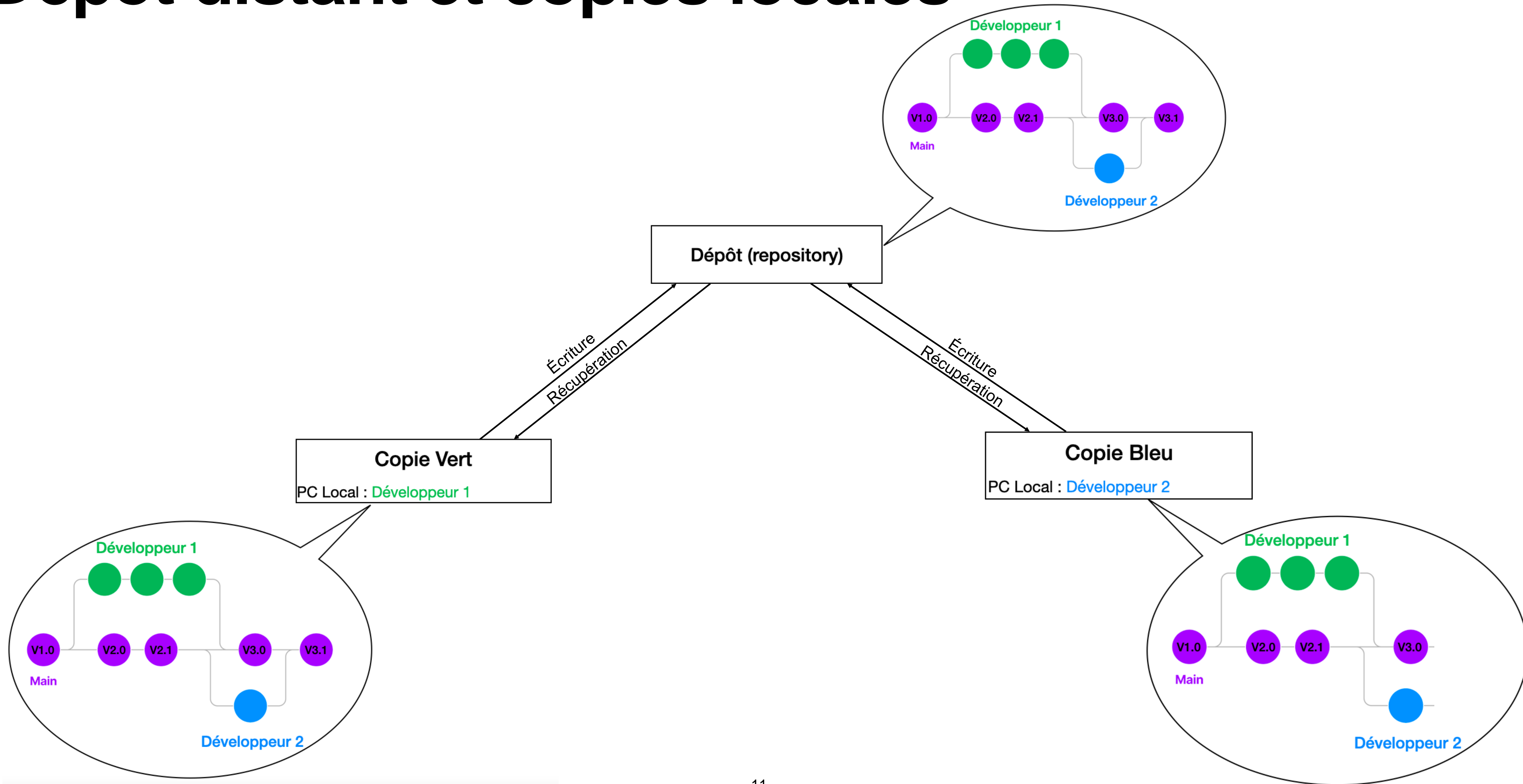
www.ionos.fr

GIT

Git

- Créé par Linus Torvalds en 2005
- Système décentralisé
 - Connexion réseau utile uniquement pour propager ses modifications ou récupérer celles de ses collègues
 - On peut configurer plusieurs dépôts distants pour un même dépôt local
 - Chaque développeur a une copie du dépôt en local permet de bénéficier de git même sans réseau
- Stockage uniquement des modifications des fichiers
 - Trafic réseau limité
 - Fusion de modifications facilitée
- Grande souplesse
 - Aucune obligation, mais des « choix par défaut »
- Documentation détaillée des commandes Git disponible sur <https://git-scm.com/docs>

Dépôt distant et copies locales



Dépôt local

- Les données sont toujours dans un dépôt local
- Répertoire dans lequel se trouvent :
 - Les fichiers versionnés
 - Un dossier de configuration Git unique (.git)
- Ce répertoire contient :
 - La configuration du dépôt distant
 - Le suivi des références (i.e. les branches et les identifiants des commits)
 - L'historique des modifications

Créer/récupérer un dépôt Git

- Créer un dépôt Git :

```
$ cd ...  
$ git init
```

- Récupérer un dépôt Git déjà existant (clone), 2 possibilités :

- Clone avec l'url ssh nécessitant une clé publique

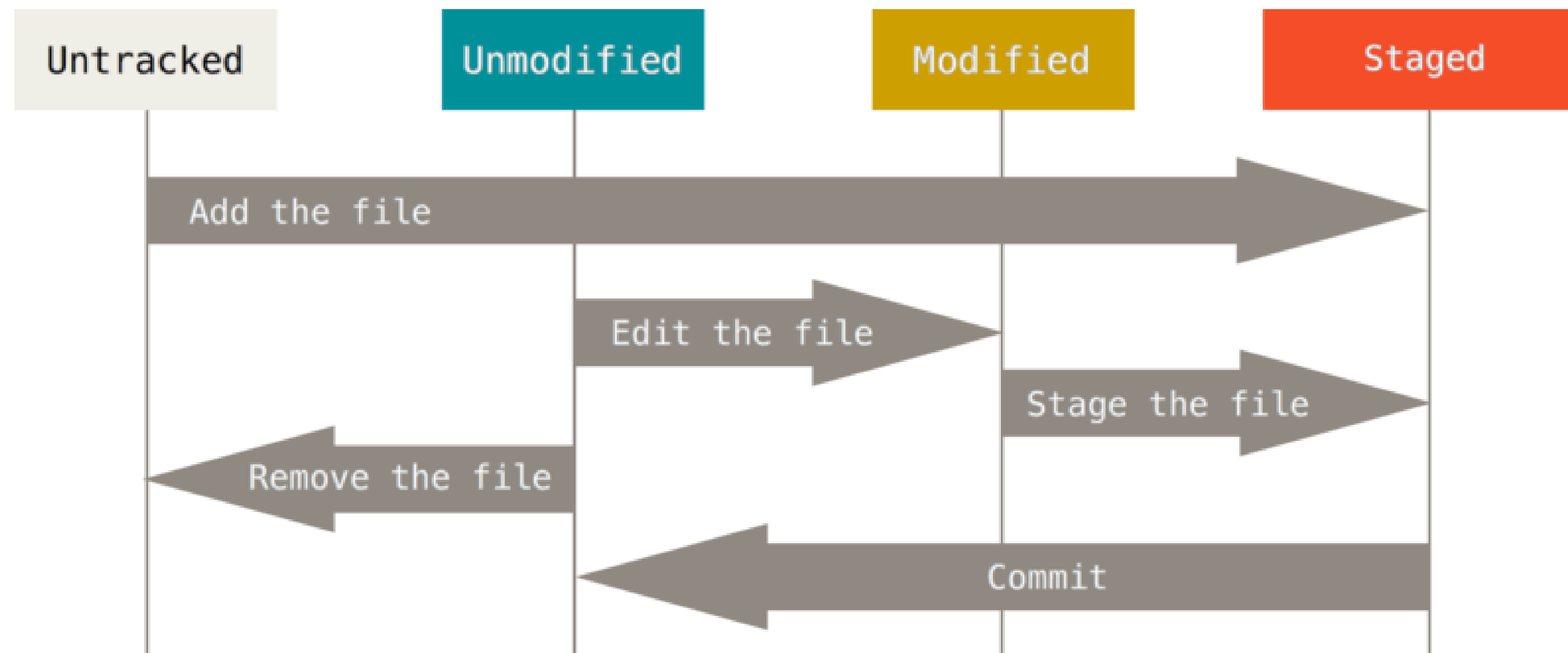
```
$ git clone ssh://urlSSH
```

- Clone en https nécessitant une authentification et un token

```
$ git clone https://usernameGit:token@urlHTTPS
```

Suivi des fichiers

- 4 états (states) pour un fichier:
 - Untracked (non suivi) : fichier qui n'est pas/plus géré par Git
 - Unmodified (non modifié) : fichier sauvegardé sur Git et qui n'a pas été modifié
 - Modified (modifié) : fichier sauvegardé sur Git mais qui a été amené à être modifié
 - Staged (indexé) : fichier modifié, prêt à être mis à jour/ajouté sur Git après soumission (commit)



Suivi des fichiers

- Vérifier l'état des fichiers (non modifié, modifié ou indexé) :

```
$ git status
```

- Indexer des fichiers :

```
$ git add <file>
```

```
$ git add <directory>
```

```
$ git add -A
```

- Voir les différences entre les fichiers non indexées et indexées :

```
$ git diff
```

- Voir les différences entre les fichiers indexées et le dernier commit :

```
$ git diff --cached
```

Gitignore

- Pour ignorer certains fichiers dans les différences Git, on crée un fichier `.gitignore` à la racine du dépôt en spécifiant les fichiers et dossiers à ignorer :
 - Commentaire avec `#`
 - Joker avec `*`
 - ...

```
# Ignore
# Debug directory
[Dd]ebug/

# Binary directory
Bin/

# Log files
*.log

# temp directory
tmp/
```


Commit et Push

- Commit : soumettre les modifications indexées dans dépôt local

```
$ git commit -m « First commit »
```

- Voir l'historique des commit

Historique simplifiée

```
$ git log
```

Historique détaillée et plus visuelle

```
$ git log --all --decorate --oneline --graph
```

- Push : soumettre les commit dans le repository distant (git push NomDuDépôt NomBranche)

```
$ git push origin master
```

Pull et fetch

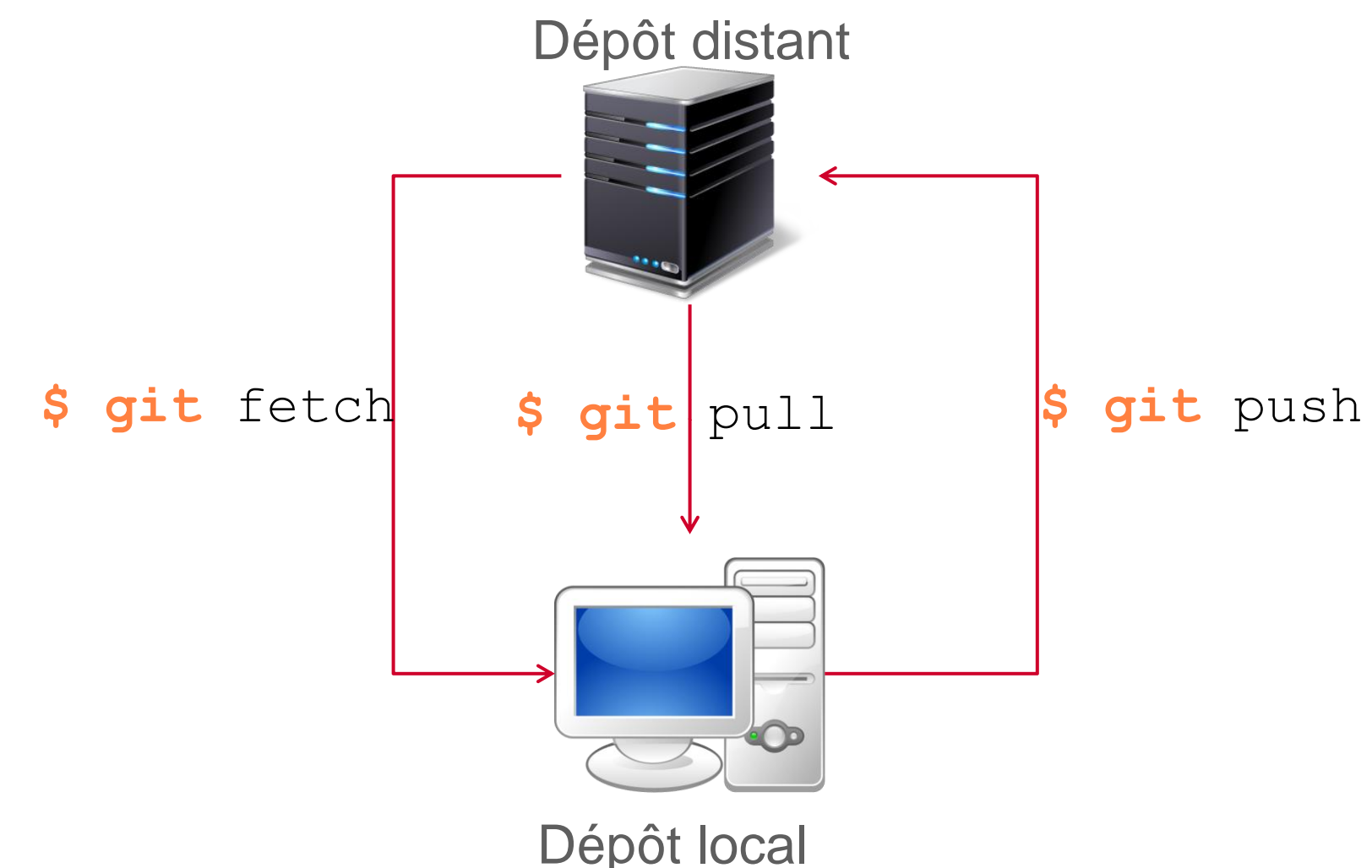
- Pull : récupérer/tirer depuis le dépôt distant pour mettre à jour le dépôt local

```
$ git pull
```

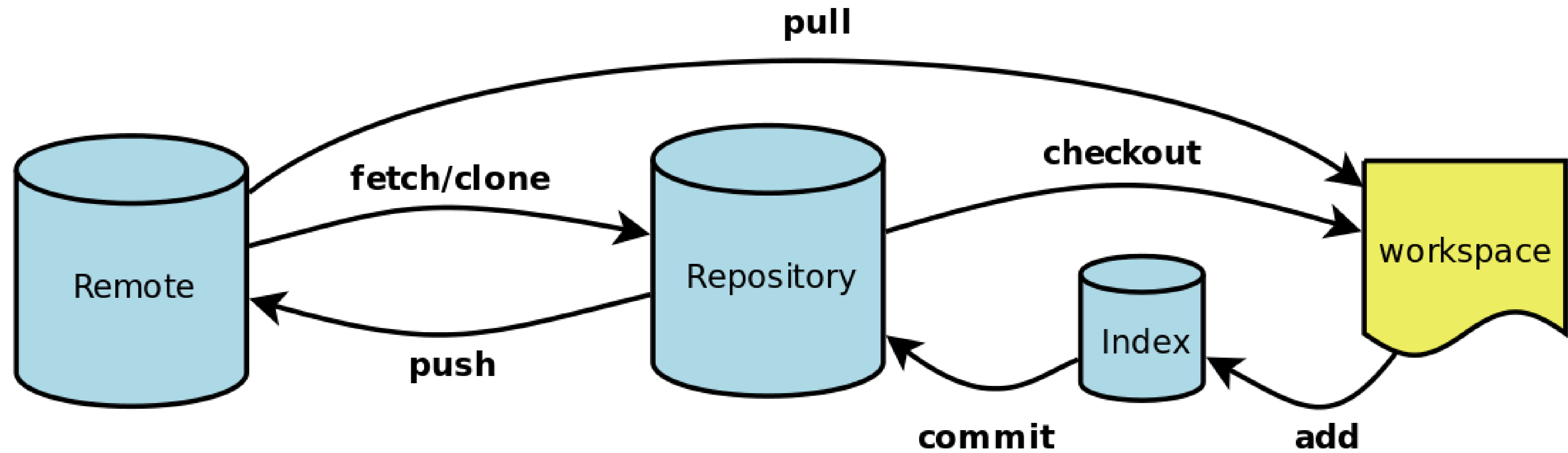
- Fetch : Récupérer les modifications présentes sur le serveur distant que vous n'avez pas encore sur votre copie en local

```
$ git fetch <remote>
```

- Différence entre pull et fetch : Le fetch télécharge le contenu distant alors que le pull met à jour le contenu local avec le contenu distant connu



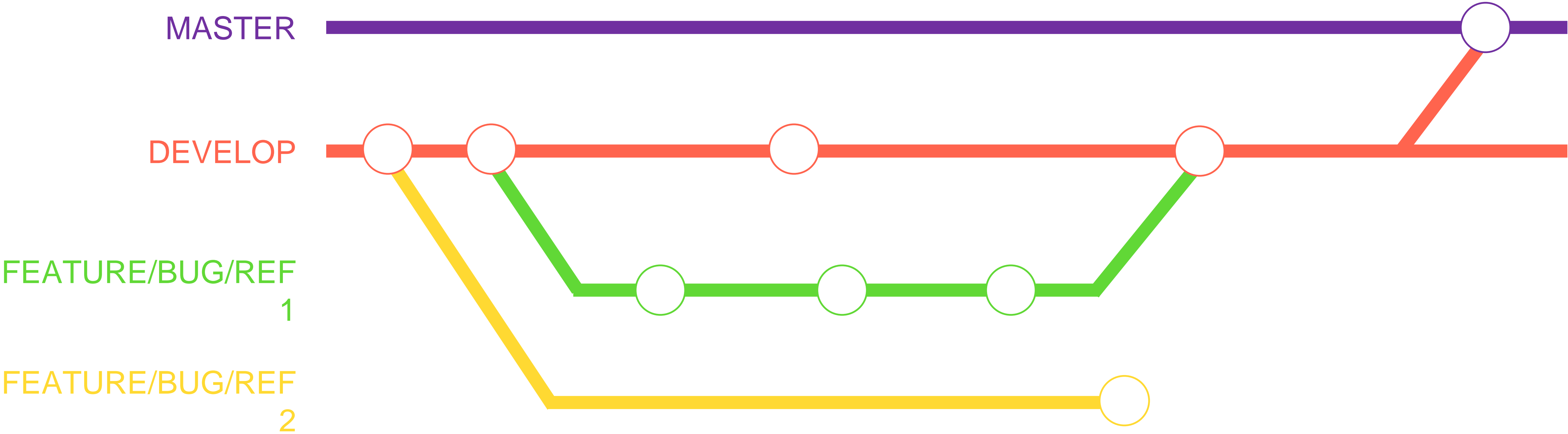
Récapitulatif



Branches

- Créer une version à partir d'un commit sans être impacté par des modifications ultérieures.
- Avantages :
 - Souplesse complète
 - Bascule rapide entre branches
 - Permet d'isoler facilement le travail de chacun
- Modèle le plus répandu :
 - Master ou main : branche de livraison
 - Develop : branche de développement
 - Feature : branche temporaire pour les nouvelles fonctionnalités

Branches



Branches

- Lister les branches locales

```
$ git branch
```

- Lister les branches distantes

```
$ git branch -r
```

- Lister toutes les branches (locales et distantes)

```
$ git branch -a
```

Branches

- Création d'une branche

```
$ git branch nomBranche
```

- Se déplacer sur une branche

```
$ git checkout nomBranche
```

- Créer et se déplacer sur une branche en une commande

```
$ git checkout -b feature/Bug/Ref1
```

- Supprimer une branche

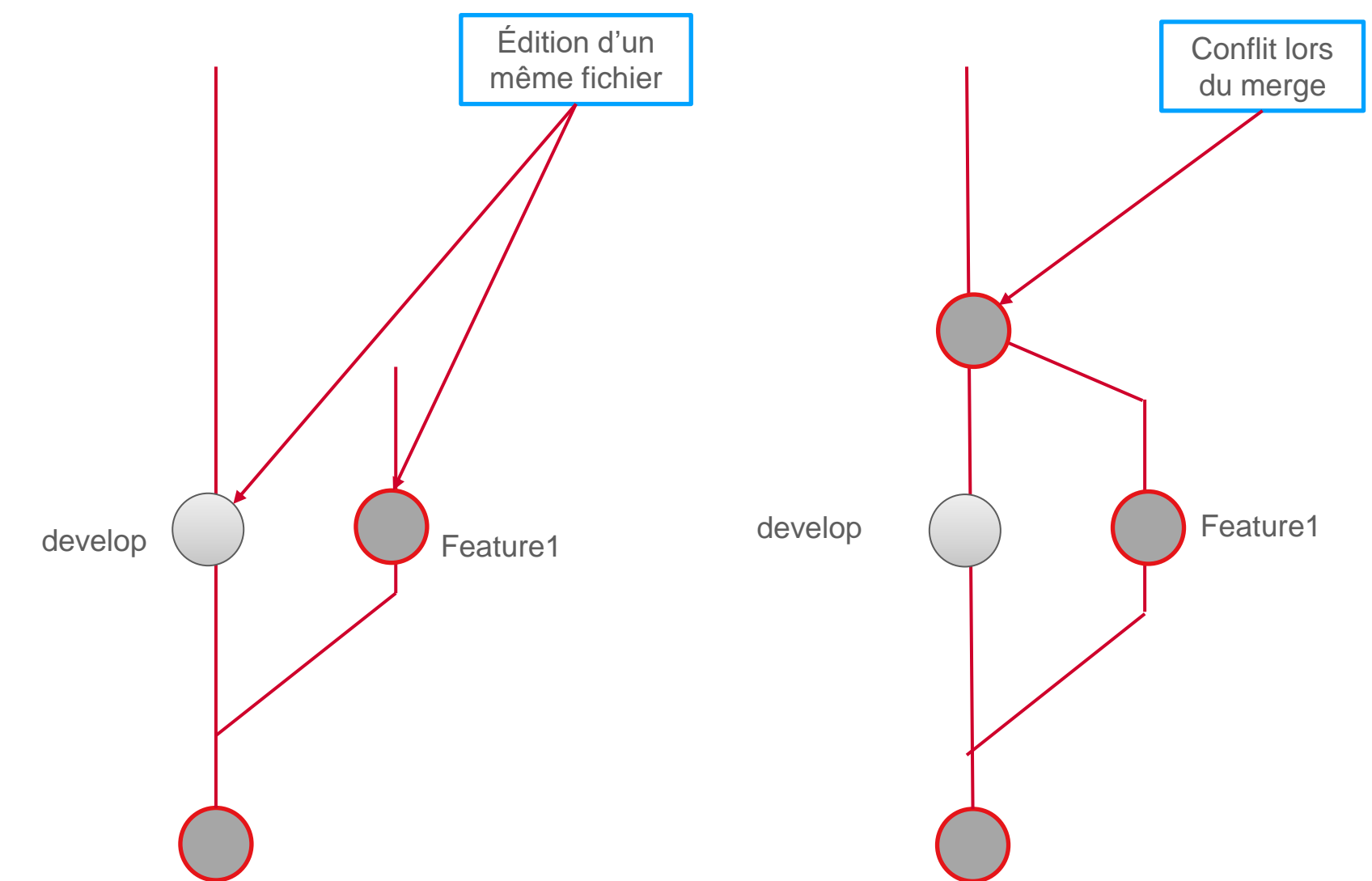
```
$ git branch -d nomBranche
```

- Fusionner la branche feature/Bug/Ref1 dans la branche develop

```
$ git checkout nomBranche  
$ git merge nomBranche
```

Git conflicts

- Un conflit survient lorsque Git ne peut pas résoudre automatiquement les différences de code entre deux versions (dans un merge, pull).
- Cas pratique :
 - Dans les 2 cas, une même ligne a été modifiée
 - Dans un cas, un fichier a été supprimé (par exemple une image) alors que dans l'autre cas le fichier a été modifié
- Dans ces cas, le processus (merge, pull...) est mis en attente tant que les conflits ne sont pas gérés :
 - Il faut voir quels fichiers n'ont pas été fusionnés (git status)
 - Gérer les conflits en choisissant les bons changements (git add)
 - Soumettre la fusion (git commit)



Modification à l'aide d'un précédent commit

- Annule un commit existant (revert)

```
$ git revert IdCommit
```

- Copie le contenu d'un commit dans un nouveau commit (cherry-pick) : utile pour dupliquer le code d'une branche à une autre

```
$ git cherry-pick IdCommit
```

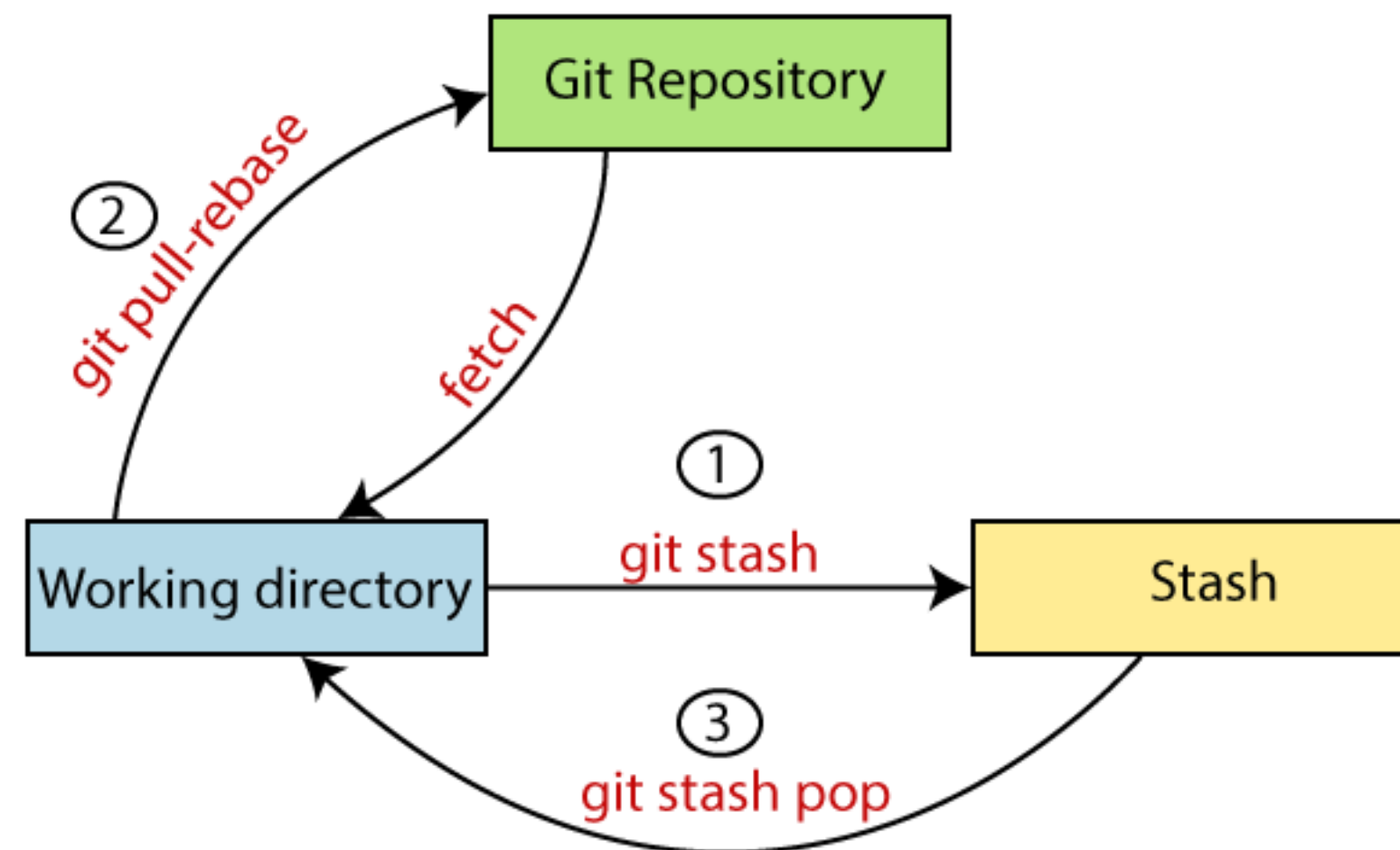
Stockage temporaire des modifications (stash)

- Stocker les modifications actuelles

```
$ git stash      ou      $ git stash push
```

- Appliquer le stash

```
$ git stash pop
```



Etiqueter un commit (tag)

- Un tag est une référence à un commit (utile pour nommer les versions)
- Lister les tags

```
$ git tag
```

- Récupérer le code correspondant à un tag

```
$ git checkout NomTag
```

- Créer un tag

```
$ git tag -a NomTag -m MessageTag
```

- Push des tags vers le dépôt distant

```
$ git push --tags
```

Exemple de versioning pour un tag

- En utilisant un ou plusieurs chiffres pouvant être séparés par des points : 1.4, 0.9.95.
- En suivant une règle mathématique.
 - La version de TEX tend vers π ; la version actuelle est 3.14159265.
- Grâce à l'année de sortie du logiciel
 - Windows 98.
- Grâce à la date de sortie
 - Ubuntu 8.04 (pour la version sortie en avril 2008).

Interface Git

- Il existe plusieurs interfaces permettant d'utiliser Git sans commandes :
 - GitHub Desktop
 - Tortoise Git
 - SourceTree
 - Fork
 - Fonctionnalités Git des IDE

Interface Git

- Principaux avantages :
 - Facile d'utilisation : pas besoin de connaître par cœur les commandes pour effectuer une action et pas de risque de se tromper sur les commandes
 - Meilleure lisibilité (modifications, historique ...)
- Principaux inconvénients :
 - Aucune certitude sur l'action Git effectuée après un clic
 - Les GUI utilisent souvent des commandes compliquées donc dangereux et incompréhensibles

Fork

swift

Local Changes (41)

All Commits

Filter

Starred

main 853↓

Branches

main 853↓

ad-side-effects

mracek

release

shahmishal

tensorflow-0.8 1↓

tensorflow-0.7

swift-5.0 2↓

Remotes

origin

main

dl

maxd

mracek

QuietMisdreavus

release

5.3

5.3-20201012

revert-26831-fix

Quick Launch Fetch Pull Push Stash

swift*
main 853↓

New Branch Open in Feedback

Fork* react* swift*

origin/main • Merge remote-tracking branch 'origin/main' into next

main Merge pull request #34916 from slavapestov/with-unsafe-conti...

Concurrency: Implement high-level withUnsafe[Throwing]Continuation entry...

AST: Add withUnsafe[Throwing]Continuation built-ins

Concurrency: Change Unsafe[Throwing]Continuation<T> to store a RawUnsaf...

SIL: Refactor get_async_continuation[_addr] to return a RawUnsafeContinuati...

AST: Add support for building 'throws' and 'async' built-in functions

AST: Introduce Builtin.RawUnsafeContinuation type

Merge remote-tracking branch 'origin/main' into next

Merge pull request #34937 from aschwaighofer/clarify_comment

Clarify comment

swift_jenkins 68b6a7a Dec 3, 2020 at 17:30

Slava Pestov c133e13 Dec 3, 2020 at 17:18

Slava Pestov 9b3cd28 Dec 1, 2020 at 23:20

Slava Pestov 5c5f012 Nov 20, 2020 at 05:11

Slava Pestov 361eef2 Nov 20, 2020 at 22:31

Slava Pestov a2dbdec Nov 20, 2020 at 07:49

Slava Pestov c2bfff0 Nov 20, 2020 at 05:10

Slava Pestov 1a89813 Nov 20, 2020 at 04:09

swift_jenkins 1d8214e Dec 3, 2020 at 16:35

swift-ci 14ea0e5 Dec 3, 2020 at 16:26

Arnold Schwaighofer 71a993c Dec 3, 2020 at 14:01

Commit Changes File Tree

AUTHOR

Slava Pestov

spestov@apple.com

December 3, 2020 at 17:18:49 GMT+1

COMMITTER

GitHub

noreply@github.com

December 3, 2020 at 17:18:49 GMT+1

REFS main

SHA c133e13efda6fe9da1baa7451cb5d1e6ee642228

PARENTS 14ea0e5 9b3cd28

Merge pull request #34916 from slavapestov/with-unsafe-continuation

Concurrency: Implement withUnsafe[Throwing]Continuation

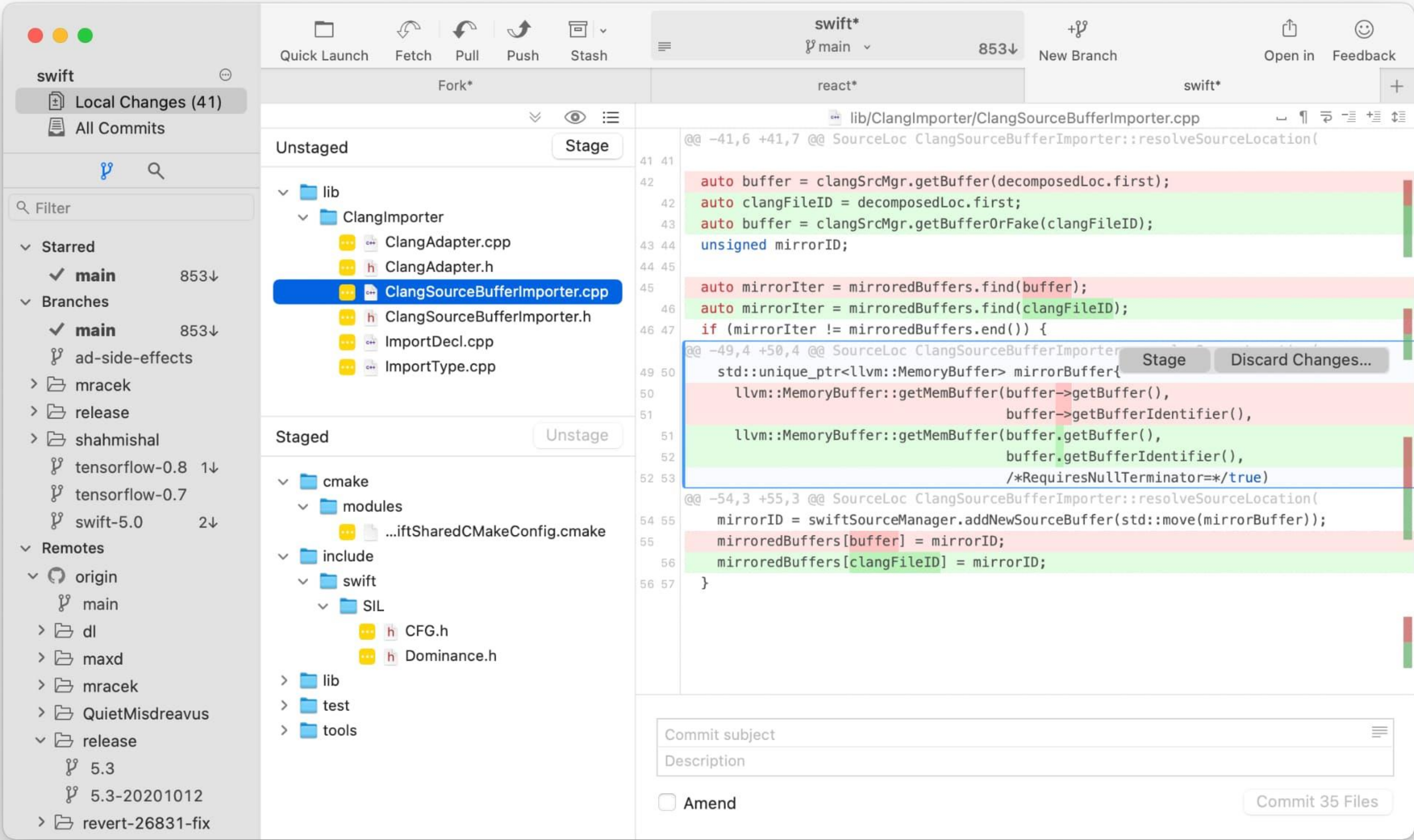
Expand All

docs/ABI/Mangling.rst

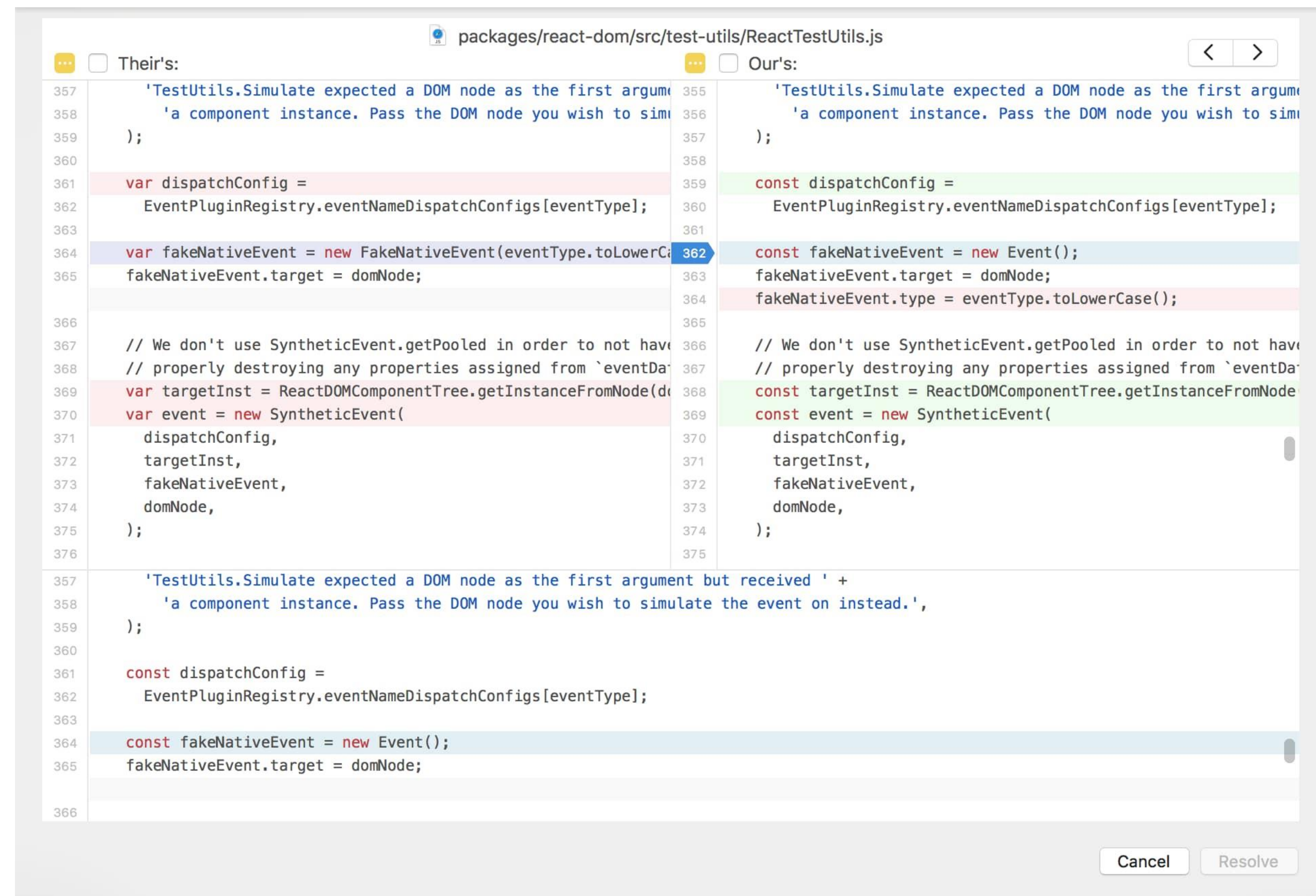
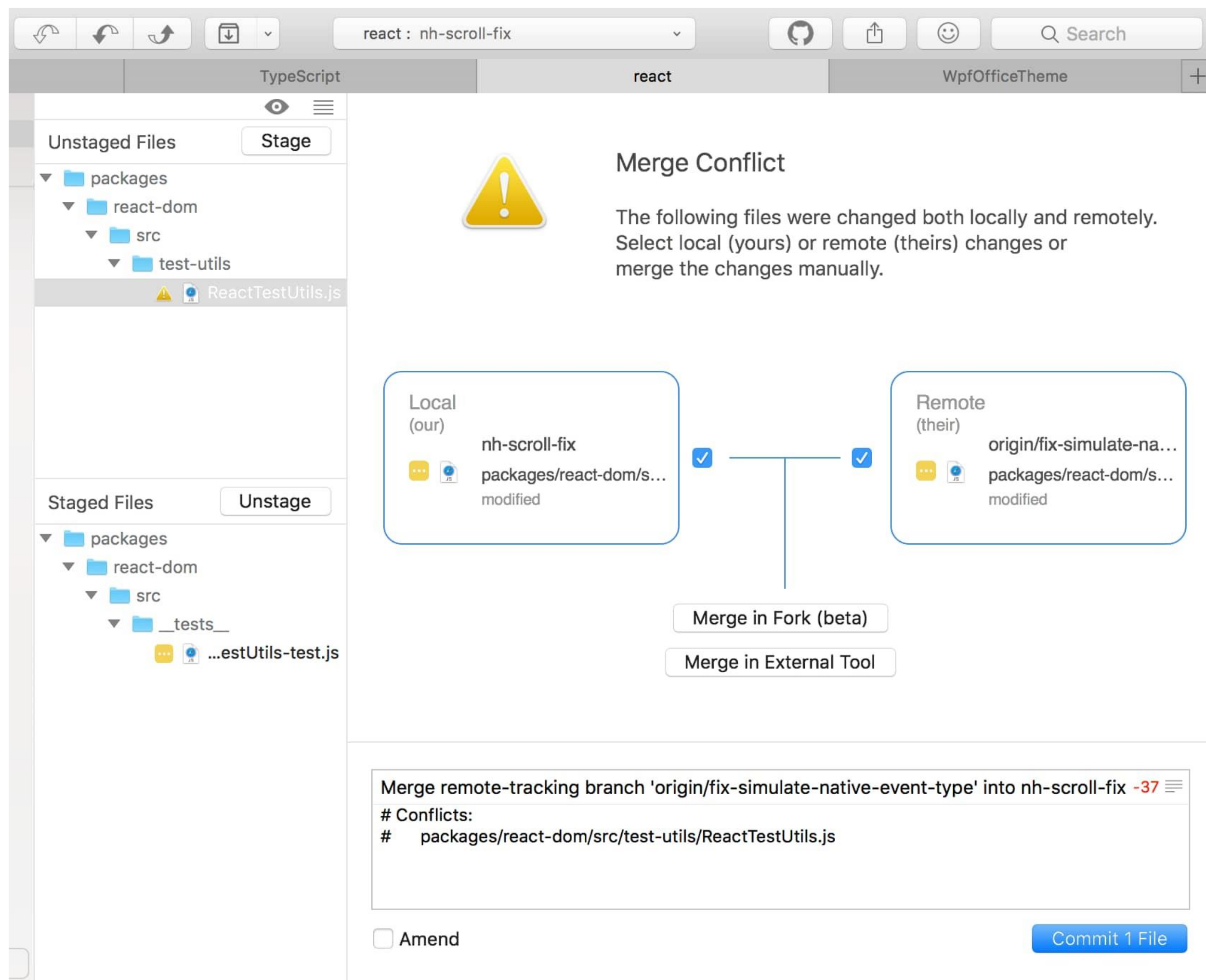
include/swift/AST/ASTContext.h

include/swift/AST/Builtins.def

Fork



Fork



Fork

Blame scripts/set-rn-version.js 714b50 | RN Monorepo | Migrate to package (#36434) (17. 3. 2023)

2017 2018 2019 2020 2021 2022 2023

ea6e34 Extract release agnosti... 1. 12. 2021
10e47b Do not depend on an E... 22. 9. 2022
714b50 | RN Monorepo | Migrat... 17. 3. 2023

```
135 125 // Change ReactAndroid/gradle.properties
136 126 saveFiles(['ReactAndroid/gradle.properties'], tmpVersioningFolder);
137
127 saveFiles(
128   ['packages/react-native/ReactAndroid/gradle.properties'],
129   tmpVersioningFolder,
130 );
138 131 if (
139 132   sed(
140 133     '-i',
141 134     /^VERSION_NAME=.*/,
142 135     `VERSION_NAME=${version}`,
143     'ReactAndroid/gradle.properties',
136     'packages/react-native/ReactAndroid/gradle.properties',
144 137   ).code
145 138 ) {
146 139   echo("Couldn't update version for Gradle");
147 140   exit(1);
148 141 }
149 142
150 143 // Change react-native version in the template's package.json
151 144 exec(`node scripts/set-rn-template-version.js ${version}`);
152 145
153 146 // Make sure to update ruby version
154 147 if (exec('scripts/update-ruby.sh').code) {
```

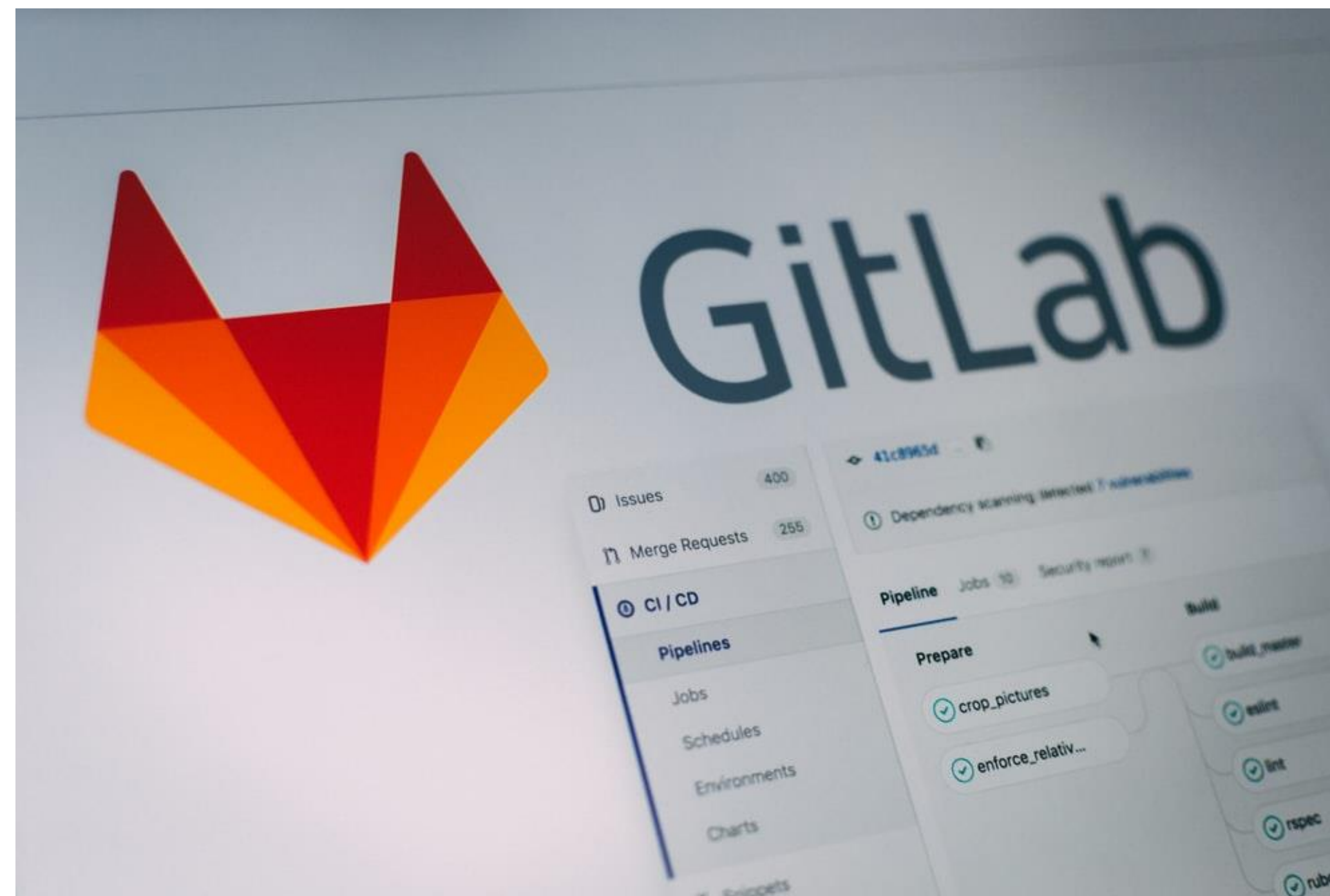
Limites de Git

- Non adapté aux fichiers binaires, les fichiers audio (WAV, MP3...) ou vidéo (MP4, MOV...)
- Beaucoup de commandes → contournement avec une GUI
- Aucune gestion de tâches à faire, revues de code... → Gitlab

GitLab

GitLab

- Plateforme web open source permettant la gestion de repository Git avec un grand nombre d'options de suivi (collaboratives par exemple).



GitLab

- 3 offres :
 - gitlab.com → gratuit
 - Community Edition → gratuit
 - Enterprise Edition (Community Edition + fonctionnalités additionnelles) → payant

GitLab - Page d'accueil d'un projet

GitLab

Next

Menu

GitLab

Project information

Repository

Issues39,264

Merge requests1,208

Requirements

CI/CD

Deployments

Monitor

Packages & Registries

Analytics

Snippets

Collapse sidebar

GitLab.org > GitLab

GitLab

Project ID: 278964

Star3030

249,102 Commits

8,152 Branches

1,976 Tags

8 GB Files

66.4 TB Storage

115 Releases

GitLab is an open source end-to-end software development platform with built-in version control, issue tracking, code review, CI/CD, and more. Self-host GitLab on your own servers, in a container, or on a cloud provider.

pipeline

running

Ruby Coverageunknown

JS Coverageunknown

master

gitlab

History

Find file

Download

Clone

Merge branch 'jv-gitaly-pack-objects-feature-flag' into 'master'

Sean McGivern authored 20 minutes ago

741c5eb2

README

Other

CHANGELOG

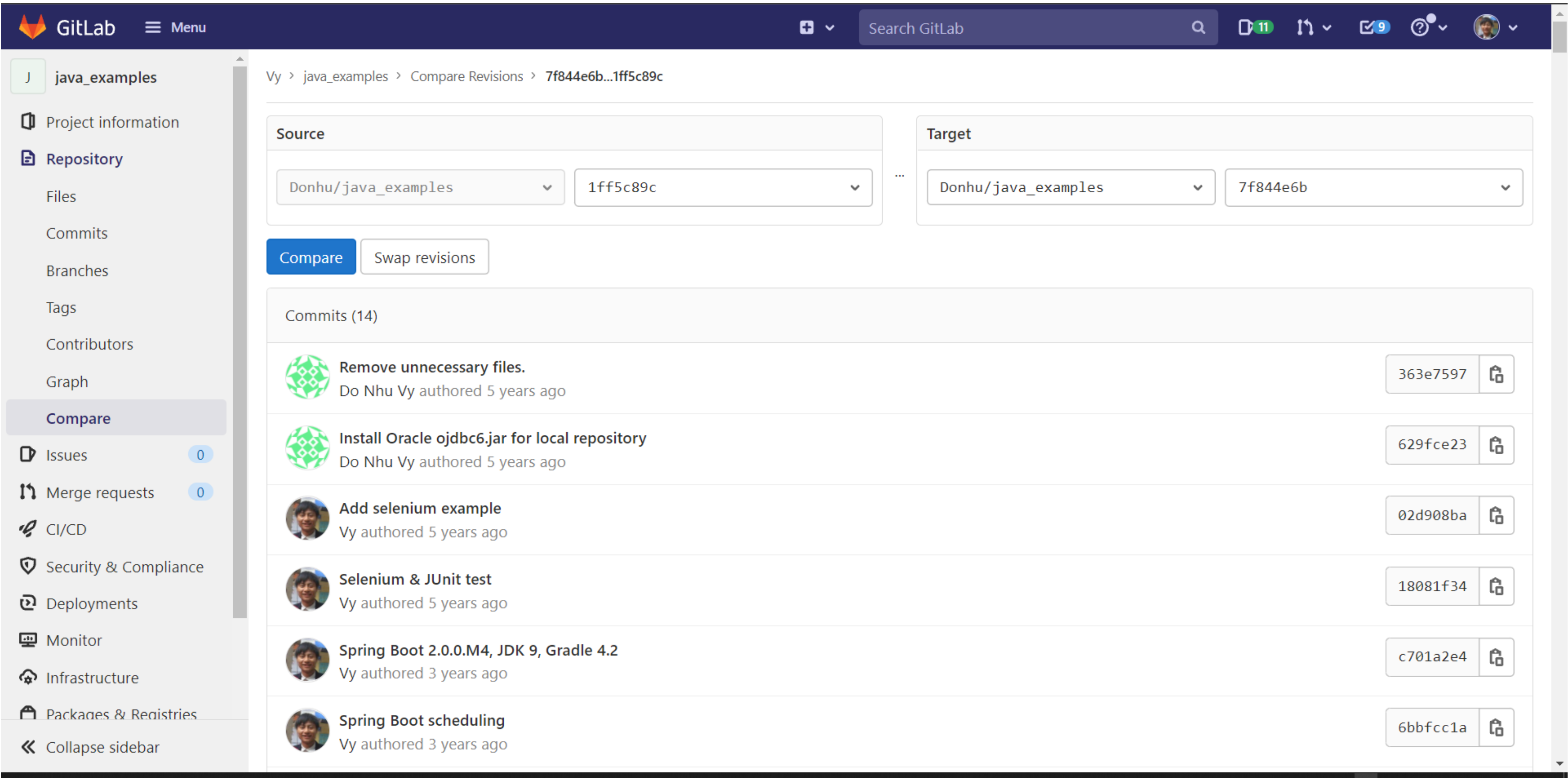
CONTRIBUTING

CI/CD configuration

Name	Last commit	Last update
.github	Rename GitLab CE to FOSS in GitHub issue ...	1 year ago
.gitlab	Merge branch 'always-run-full-migration-sp...	2 hours ago
.theia	Revert "Move host authorization to to gitp...	10 months ago
app	Merge branch 'ph/324166/newReadyToMe...	1 hour ago
bin	Skip hooks when pushing security branche...	3 weeks ago
builds	Add missing builds/ folder to fix backup tests	5 years ago
changelogs	Remove wrapper methods from Gitlab::Dat...	4 weeks ago
config	Merge branch 'jv-gitaly-pack-objects-featur...	20 minutes ago
danger	Clarify the milestone requirements for sec...	4 days ago
data/whats_new	Create 14.2 In-app top 7 product highlights	1 day ago
db	Merge branch 'mc-remove-column-from-pr...	3 hours ago
doc	Merge branch 'docs-fix-dag-detail-2' into '...	32 minutes ago

39

GitLab – Vue de l'ensemble des commits



GitLab – Comparaison entre branches

Flightjs > Flight > Compare Revisions > alunny/publish_lib...master

Source

flightjs/Flight

▼

master

▼

...

Target

flightjs/Flight

▼

alunny/publish_lib


▼

Compare

Swap revisions

Create merge request


Commits (3)

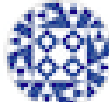


Merge pull request #378 from flightjs/alunny/publish_lib
angus croll authored 3 years ago

!

8749d499







v1.5.2
Andrew Lunny authored 3 years ago

!

f15b3227






Update README.md
Administrator authored 43 minutes ago

||

a7b48b50





Showing 2 changed files ▼ with 2 additions and 2 deletions

Hide whitespace changes

Inline

Side-by-side

▼


 README.md 

View file @a7b48b50

1		- # Please note
	1	+ # Please note (main)
2	2	
3	3	Flight is **not under active development** . New pull requests will not be accepted unless they fix core bugs or security issues.
4	4	
...	...	

41


GitLab - Clone


 **GitLab**


Projects ▾ Groups ▾ More ▾


+


 ▾














W

w3schools-test.gitlab.io

Project overview

Details

Activity

Releases

Repository

Issues0

Merge requests0

Requirements

CI/CD

w3schools test > w3schools-test.gitlab.io

W

w3schools-test.gitlab.io

Project ID: 26232378 [Request Access](#)

🔗 13 Commits

🌿 1 Branch

🏷 0 Tags

📄 543 KB Files

💾 638 KB Storage

w3schools.com repository used in demonstrating Git and GitLab Pages

master ▾


w3schools-test.gitlab.io /

+ ▾


History

Find file

Edit fork in Web IDE ▾



Clone ▾



Added configuration for GitHub Pages

w3schools test authored 2 hours ago


📄 README

📄 CI/CD configuration


🔒 No license. All rights reserved

Name	Last commit
------	-------------


Clone with SSH



Clone with HTTPS



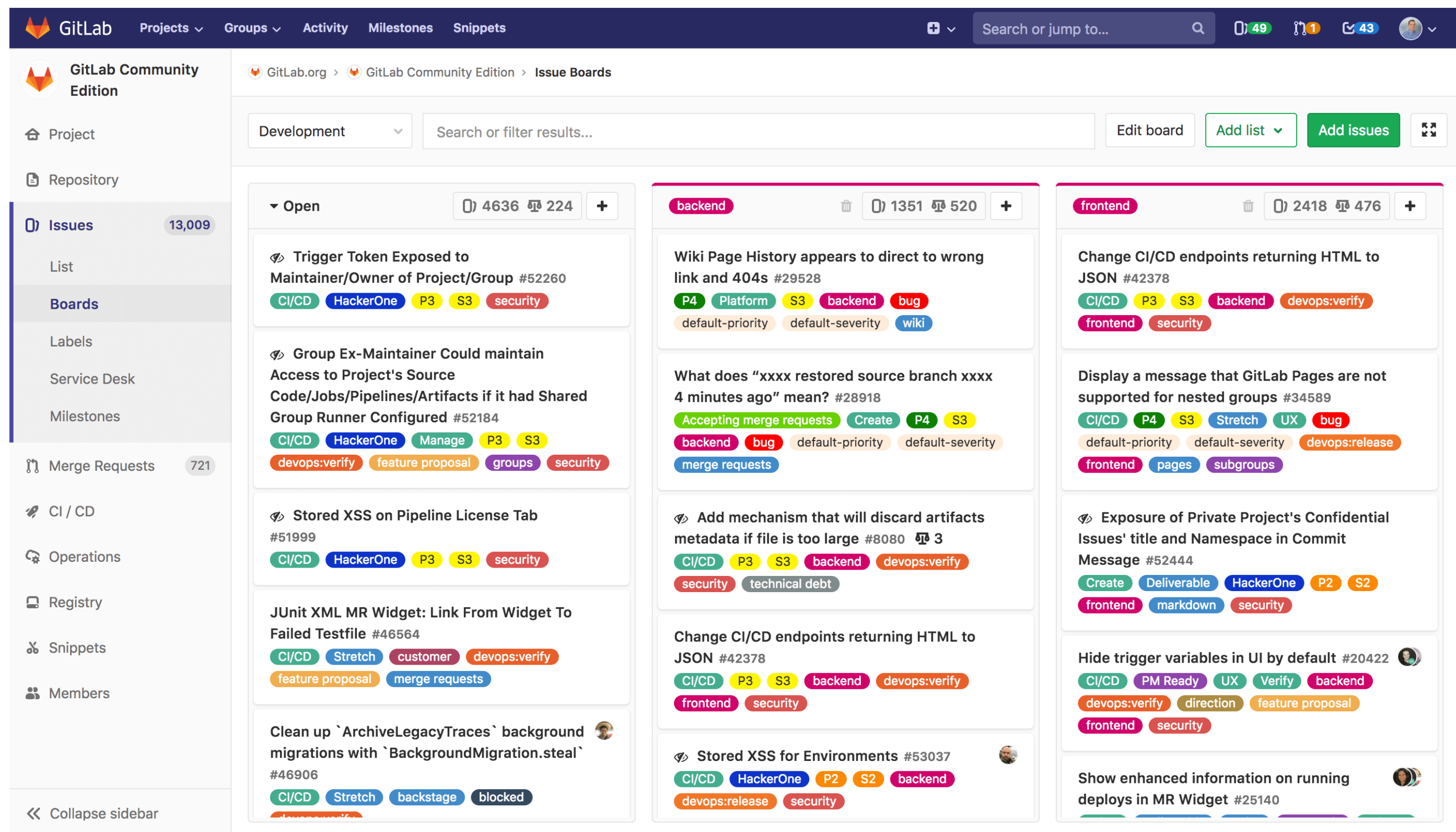
Open in your IDE

 Copy URL

42

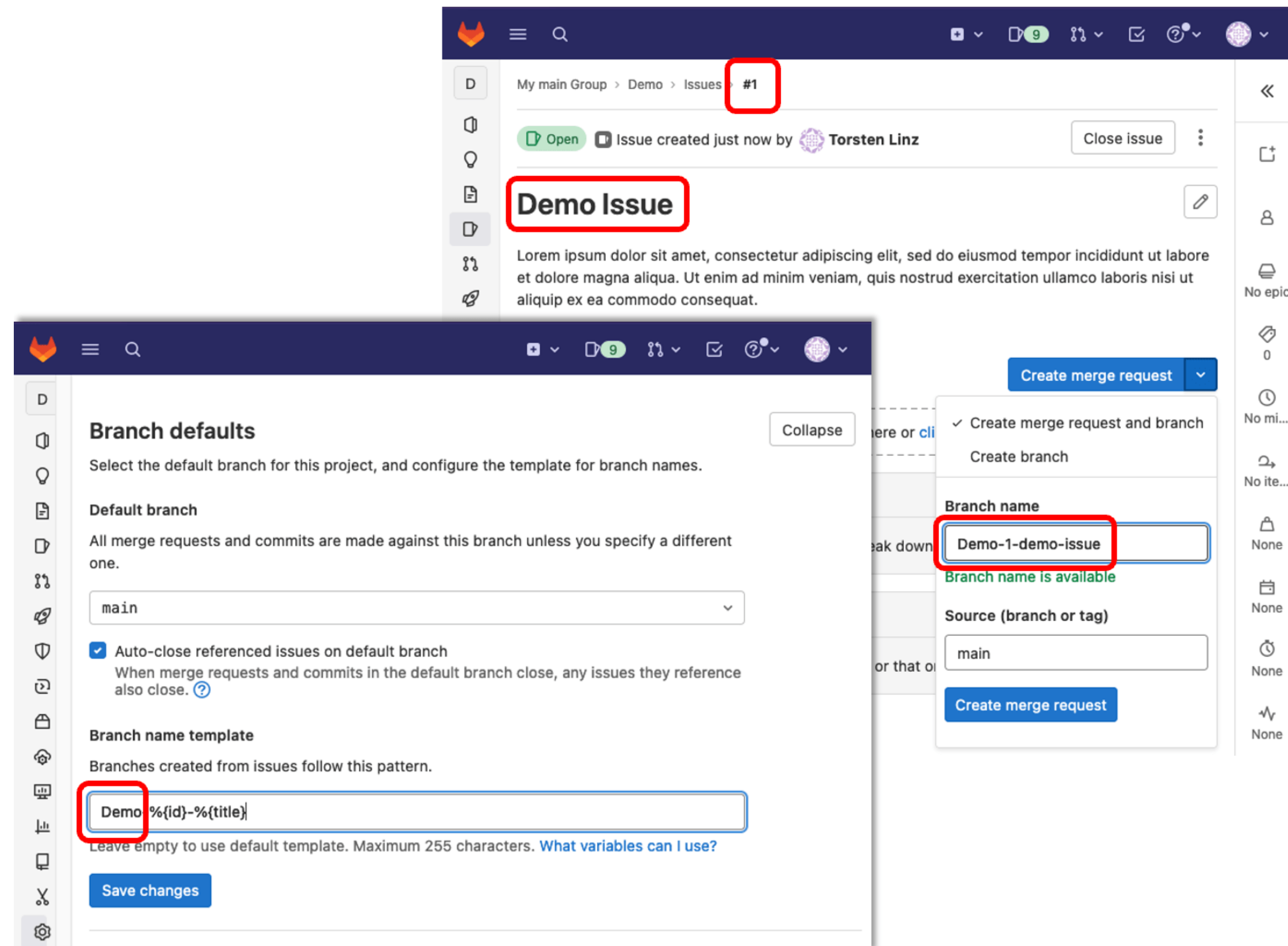
GitLab - Issues

- Permet de recenser les différentes tâches à faire
- Possibilité de l'associer à des labels, milestones...



GitLab - Issues

- Depuis une issue, il est possible de créer une branche et une merge request



GitLab – Merge requests

- Il s'agit d'une requête pour fusionner les changements d'une branche vers une autre. Cela permet de revoir les modifications (code review), modifier les erreurs pour valider la fusion
- Remarques :
 - Il peut être relié ou non à une issue
 - 1 Merge Request \neq 1 commit
 - La branche de destination n'est pas forcément la branche master
 - La fusion peut être impossible sans l'approbation de 1 ou plusieurs personnes

GitLab – Merge requests

Discussion 22 Commits 7 Pipelines 9 **Changes 10**

Show all activity ▾

✔ 5/5 discussions resolved

🔍 Filter files



📁 ...mponents/tree_list.vue	+95 -21
📁 ...onents/file_row.vue	+33 -1
📁 ...stylesheets/pages/diff.scss	+12 -2
📁 ...s/unreleased/mr-file-list.yml	+5 -0
📁 locale/gitlab.pot	+6 -0
📁 package.json	+1 -1
📁 ...onents/tree_list_spec.js	+70 -2
📁 ...pts/diffs/store/utils_spec.js	+22 -0
📁 ...nts/file_row_spec.js	+36 -0
📁 yarn.lock	+6 -1

10 changed files

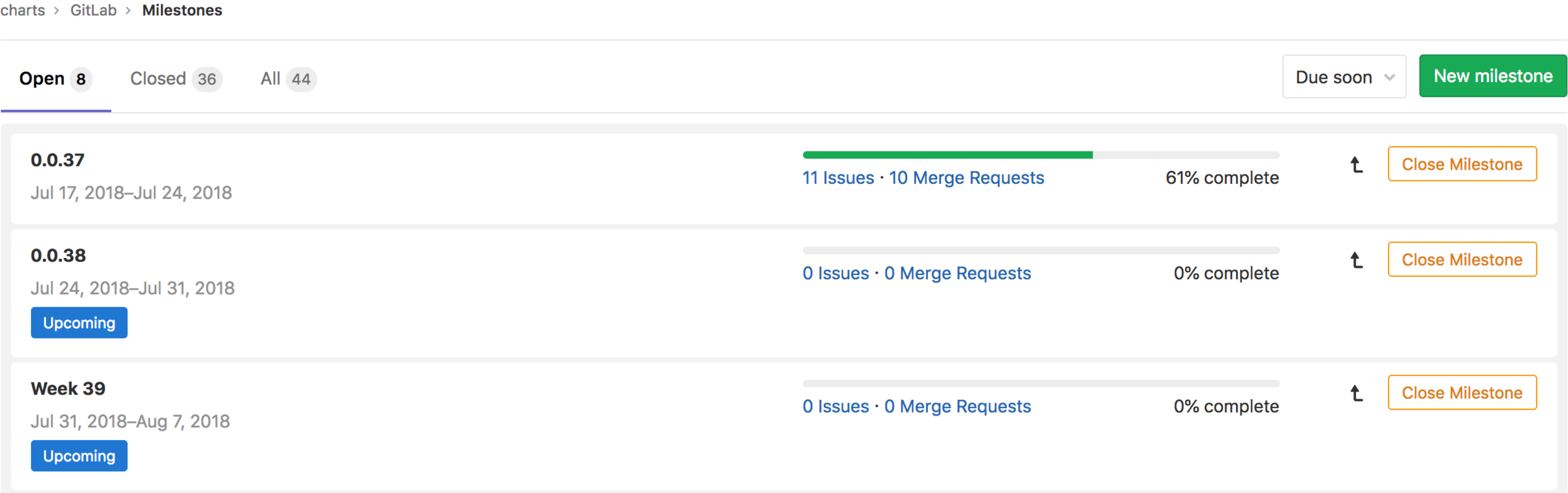
286 additions 28 deletions

▼ ▼ app/assets/javascripts/diffs/components/tree_list.vue 🔗 Edit View file @ 3b3aa28c

1	1	<script>
2	2	import { mapActions, mapGetters, mapState } from 'vuex';
	3	+ import { TooltipDirective as Tooltip } from '@gitlab-org/gitlab-ui';
	4	+ import { convertPermissionToBoolean } from '~/lib/utils/common_utils';
3	5	import Icon from '~/vue_shared/components/icon.vue';
4	6	import FileRow from '~/vue_shared/components/file_row.vue';
5	7	import FileRowStats from './file_row_stats.vue';
6	8	
	9	+ const treeListStorageKey = 'mr_diff_tree_list';
	10	+
7	11	export default {
	12	+ directives: {
	13	+ Tooltip,
	14	+ },
8	15	components: {
9	16	Icon,
10	17	FileRow,
11	18	},
12	19	data() {
	20	+ const treeListStored = localStorage.getItem(treeListStorageKey);

GitLab - Milestones

- Permet de recenser les différentes versions préparées/à préparer



GitLab – Configuration

- Gestion des membres (accès, droits)

Members 5

Export as CSV

Filter members

Account

Account	Source	Access granted	Max role	Expiration
<div><div></div><div>Administrator</div><div>@root</div><div>It's you</div></div>	Direct member	1 day ago by Administrator	Owner	Expiration date
<div><div></div><div>User 0</div><div>@user0</div></div>	One	1 day ago by Administrator	Reporter	Expiration date
<div><div></div><div>User 1</div><div>@user1</div></div>	One / Two	1 day ago by Administrator	Developer	Expiration date
<div><div></div><div>User 2</div><div>@user2</div></div>	<div>Inherited</div> <div>One / Two / Three</div>	1 day ago by Administrator	Developer	Expiration date
<div><div></div><div>User 3</div><div>@user3</div></div>	Direct member	1 day ago by Administrator	Maintainer	Expiration date

GitLab - Configuration

- Gestion du format de messages du commit (à l'aide des regex)

The screenshot displays the GitLab web interface for configuring 'Push Rules'. The top navigation bar includes the GitLab logo, a 'Next' button, and links for Projects, Groups, Activity, Milestones, and Snippets. A search bar is also present. The left sidebar shows the 'Settings' menu with options like Wiki, Snippets, General, Members, Integrations, Repository (highlighted), CI / CD, Operations, Pages, and Audit Events. The main content area is titled 'Push Rules' and includes a 'Collapse' button. It explains that push rules outline what is accepted for the project and provides an 'Add new push rule' section with several unchecked checkboxes: 'Committer restriction', 'Reject unsigned commits', 'Do not allow users to remove git tags with git push', 'Check whether author is a GitLab user', and 'Prevent committing secrets to Git'. The 'Commit message' section features a text input field containing the regular expression: `^((feature|fix)\\/([a-z,A-Z]+))(-)(\\d*)(:)([a-z,0-9])$`. A note below the input states: 'All commit messages must match this [regular expression](#) to be pushed. If this field is empty it allows any commit message. For example you can require that an issue number is always mentioned in the commit message.'

GitLab - Configuration

- Contrôle des branches pour déterminer qui peut merge/push/modifier une branche


Protected branches

Collapse

Keep stable branches secure and force developers to use merge requests. [What are protected branches?](#)




⚠

Giving merge rights to a protected branch also gives elevated permissions for certain CI/CD features. [What are the security implications?](#)

Protected branches  2

Add protected branch

By default, protected branches restrict who can modify the branch. [Learn more.](#)

Branch	Allowed to merge	Allowed to push and merge	Allowed to force push 	
main <div>default</div>	<div>Maintainers</div>	<div>2 roles</div>	<div></div>	<div>Unprotect</div>
master	<div>Maintainers</div>	<div>2 roles</div>	<div></div>	<div>Unprotect</div>

50

GitLab - Configuration

- On peut aussi déterminer quelle est la branche par défaut, fermer automatiquement une issue après le merge de la branche, prévoir un modèle de nom de branches depuis l'issue

Branch defaults

Collapse

Select the default branch for this project, and configure the template for branch names.

Default branch

All merge requests and commits are made against this branch unless you specify a different one.

main

☒ Auto-close referenced issues on default branch

When merge requests and commits in the default branch close, any issues they reference also close. [?](#)

Branch name template

Branches created from issues follow this pattern.

Feature/Bug/

Leave empty to use default template. Maximum 255 characters. [What variables can I use?](#)

Save changes