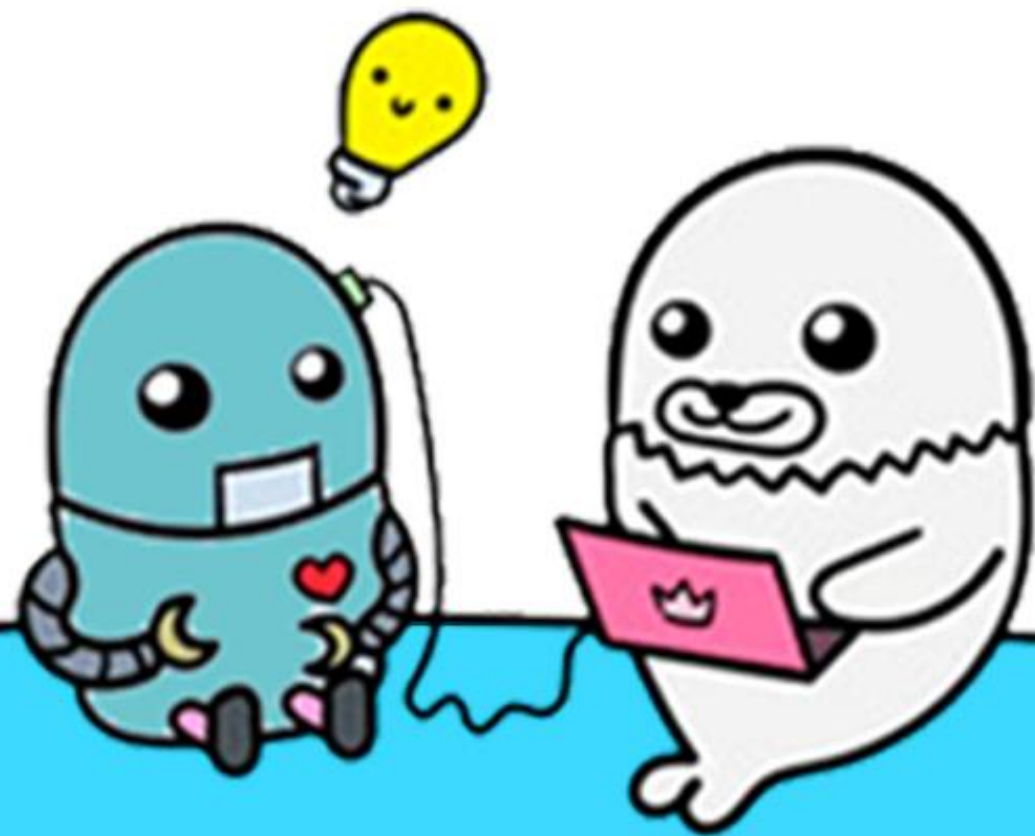


MLOps и production подход к ML исследованиям



28 марта - 28 мая



MLOps и production подход к ML исследованиям

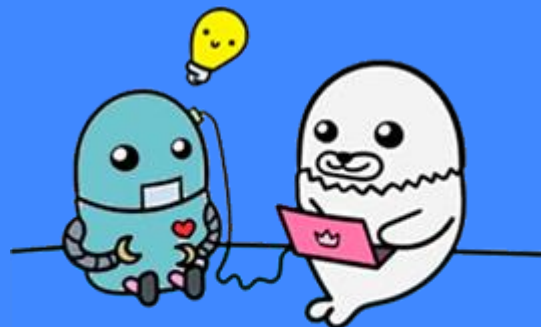
Хранение и версионирование кода с git

Павел Кикин

Газпромнефть ЦР

Руководитель направления NLP

t.me/pavel_kikin





План занятия

- Общая информация о Git
- Основные команды, возможности и принципы работа
- Работа с удаленными репозиториями (Gitlab)
- Настройка репозитория
- Разрешение конфликтов
- Github-flow
- Версионирование с Semver

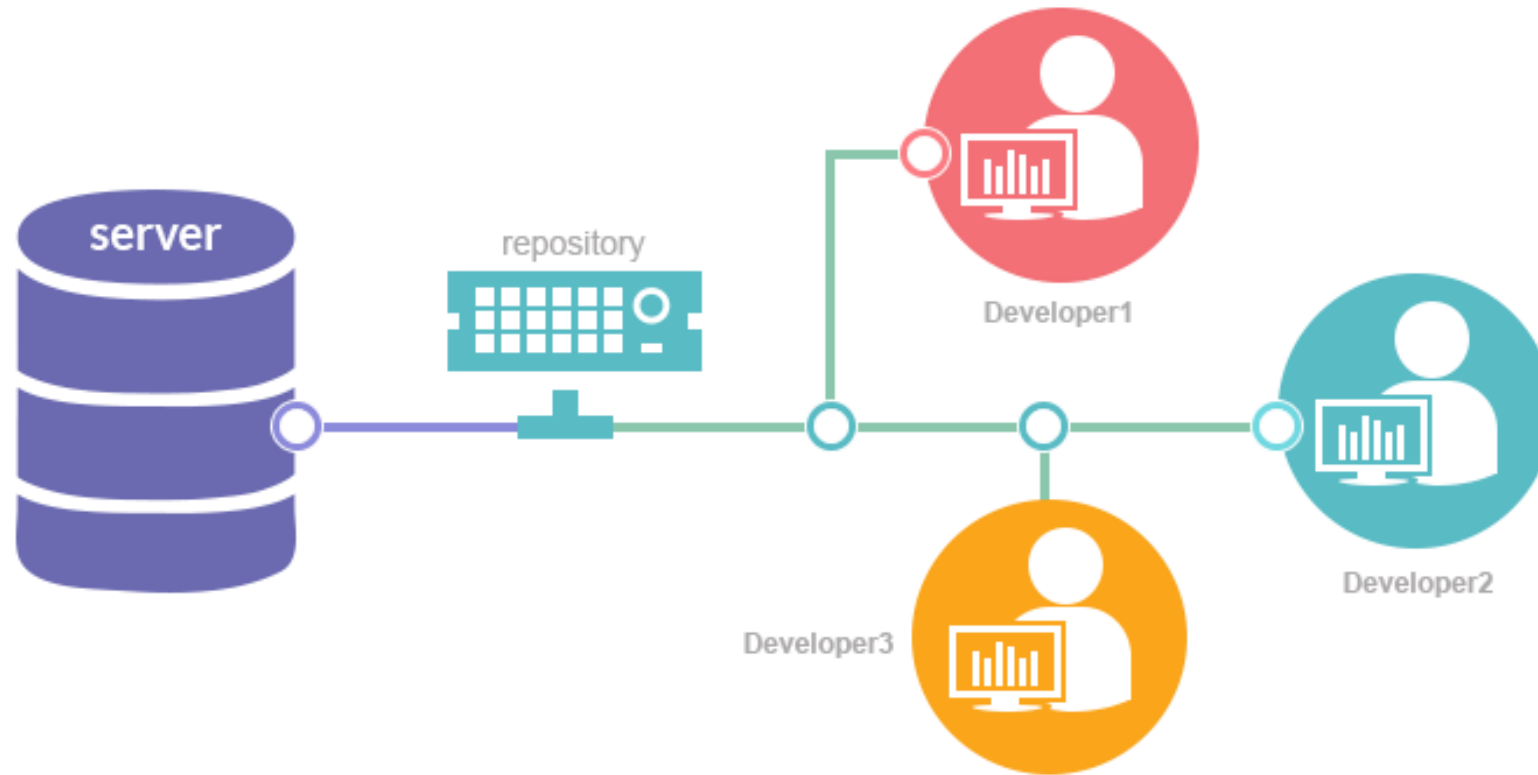


Назначение Git





Назначение Git





Git это Github?



ATLASSIAN

Bitbucket



GitLab



r/github

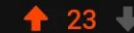
Posted by u/Kobayashimika • 9h

Help me understand Github

I don't really understand github and the difference between git and github.

What is it exactly? Please explain in basic words as english is not my mother tongue..And can i post my codes like simple css? can i make it private and only accessible to those who have the link?

thankyou in advance



23



33

Share



BEST COMMENTS ▾



surprisedduck26 • 8h



2 Awards

The difference between Git and Github is like the difference between porn and pornhub



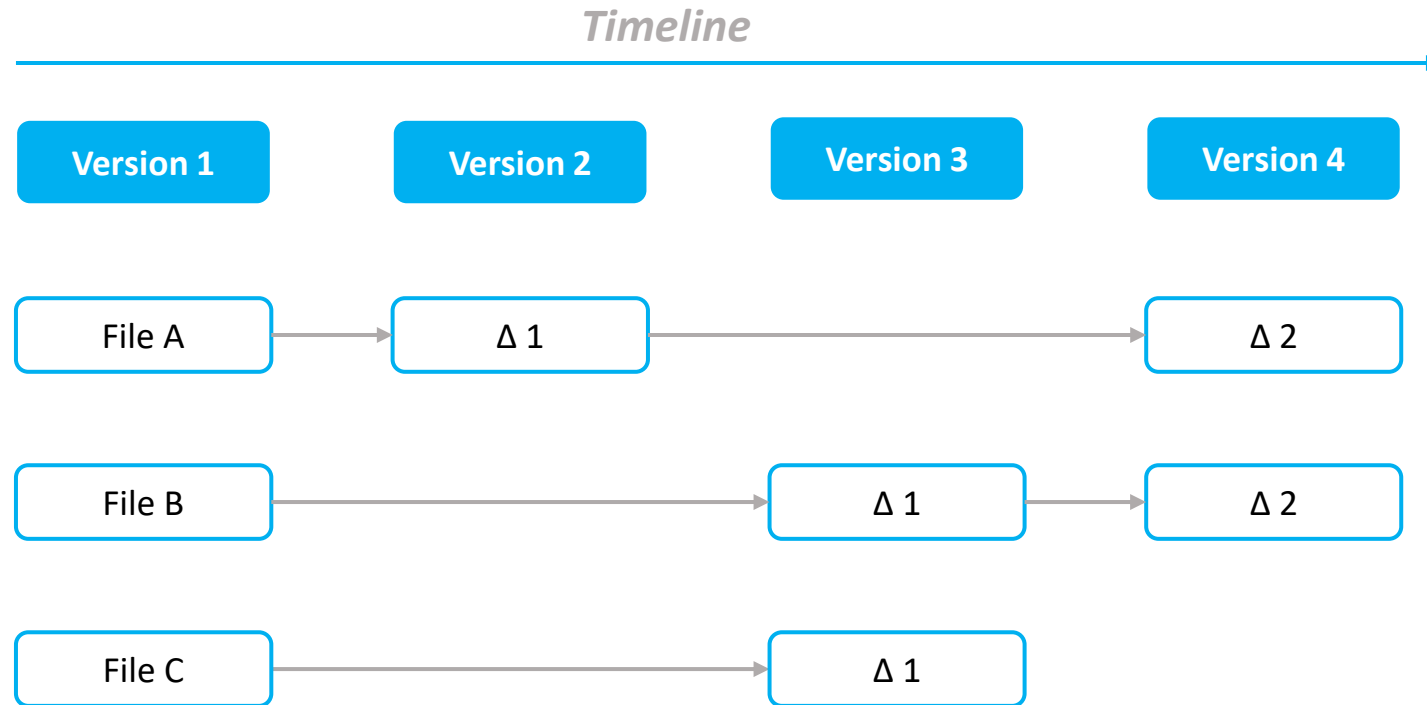
Reply



95

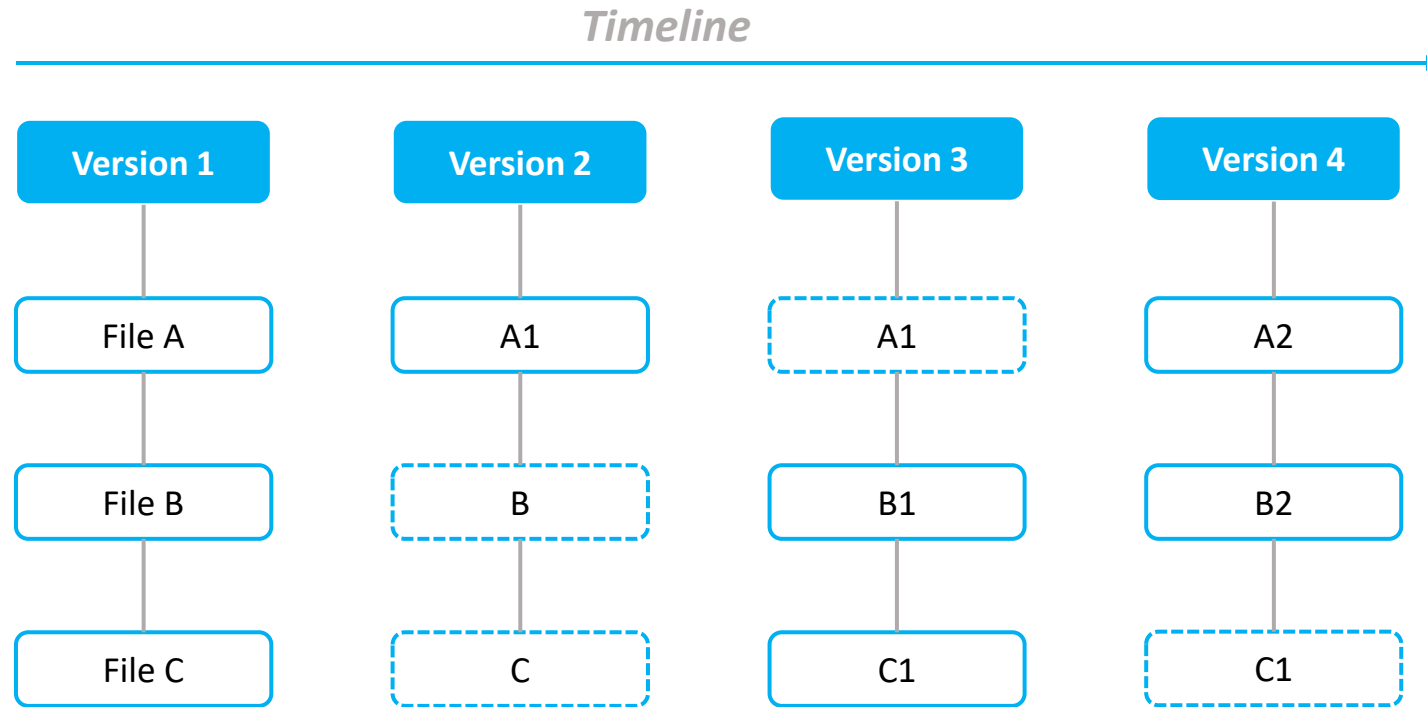


Принципы работы VCS





Принципы работы Git





Принципы работы Git





Команды Git

Первоначальная настройка конфигурации

- config
 - уровень конфигурации
 - --local
 - --global
 - --system
 - user.name
 - user.email



Команды Git

Создание репозитория

- `init`
 - `--bare <directory> (opt)`
 - `--template <directory> (opt)`
 - `<directory> (opt)`



Команды Git

История состояний

- log
 - -p

```
$ git log
commit be07ffe4d71f8e58a82bb286fd7fec245e3d1705 (HEAD -> master)
Author: Pavel Kikin <it-technologies@yandex.ru>
Date:   Sun Apr 3 00:11:58 2022 +0300

    initial commit
```



Команды Git

Просмотр последних изменений

- Status
 - Untracked
 - Tracked
 - Modified
 - Staged
 - Committed

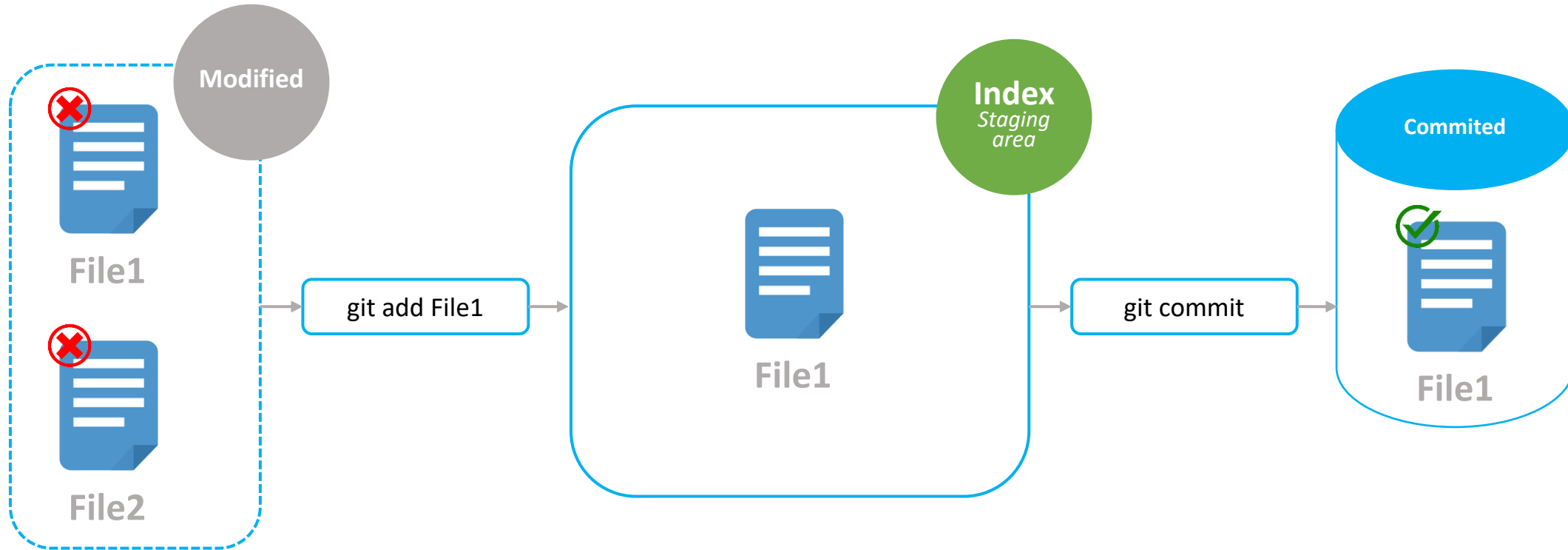
```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   real_estate.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   real_estate.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .idea/
        data/
```



Статусы файлов





Команды Git

Добавление изменений в index

- add
 - <file_name>
 - <directory>
 - .
 - *



Команды Git

Снимок текущего состояния индекса

- commit
 - -m
 - -a
 - -am





Команды Git

Информация о commit

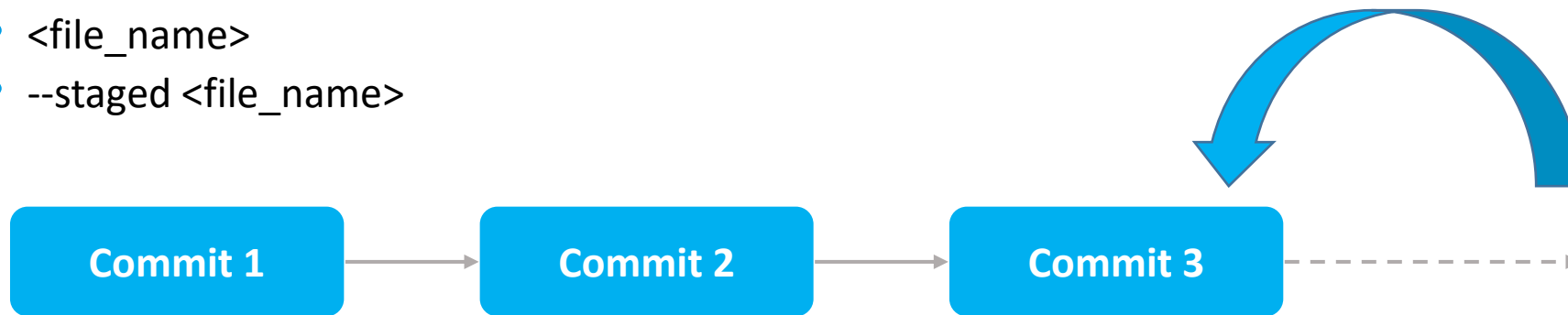
- show
 - <commit hash>
 - <tag>
 - <branch> <commit number>



Команды Git

Откат изменения на состояние последнего commit

- Restore
 - `<file_name>`
 - `--staged <file_name>`





Команды Git

Просмотр изменений с последнего commit

- diff
 - --staged

```
$ git diff
diff --git a/real_estate.py b/real_estate.py
index 5541f8c..3cd0903 100644
--- a/real_estate.py
+++ b/real_estate.py
@@ -15,8 +15,6 @@ MAX_KITCHEN = 70
     MIN_PRICE = 1_000_000 # Outlier range for price
     MAX_PRICE = 100_000_000

-data = pd.read_csv(FILE_PATH)
-

def clean_data(df: pd.DataFrame) -> pd.DataFrame:
    """Function removes excess columns and enforces
```



Команды Git

Удаление файлов

- `rm`
 - `<file_name>`
 - `--cached <file_name>`



.gitignore

Системный файл git, содержащий патерны игнорирования

- Кэши зависимостей
- Скомпилированный код
- Каталоги для выходных данных сборки
- Файлы, сгенерированные во время выполнения
- Скрытые системные файлы
- Личные файлы конфигурации IDE

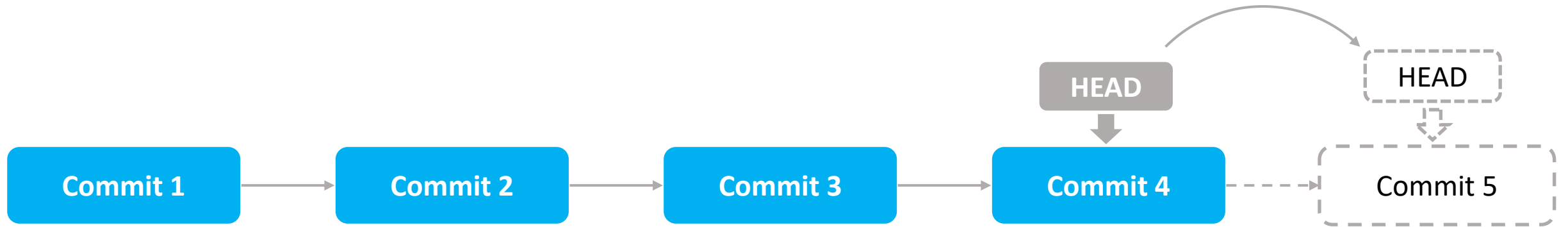
```
# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml
.idea/**/dynamic.xml
.idea/**/uiDesigner.xml
.idea/**/dbnavigator.xml
```

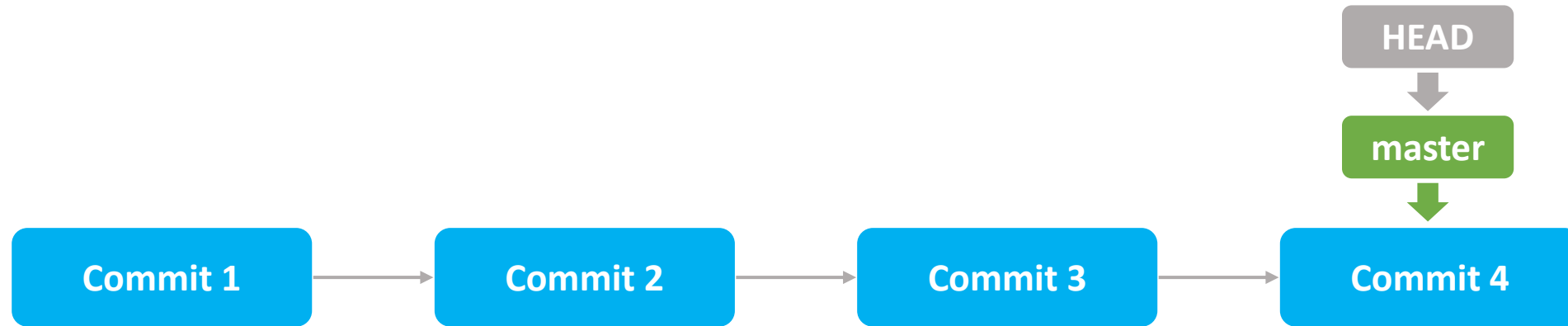


Branches | Метка HEAD





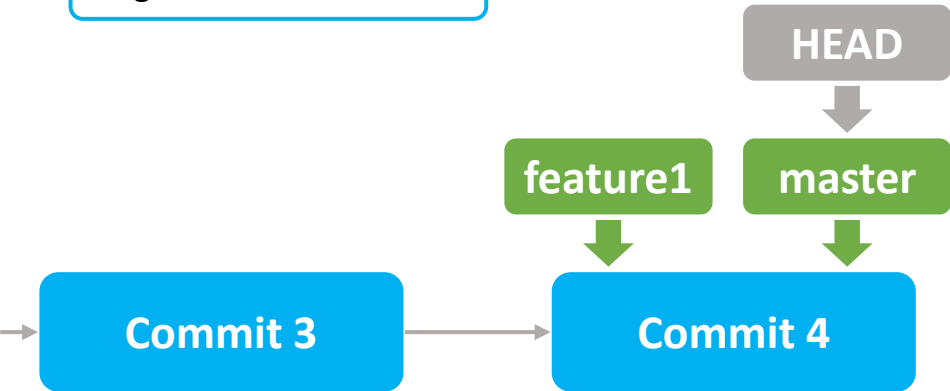
Branches





Branches | Создание ветки

git branch feature1

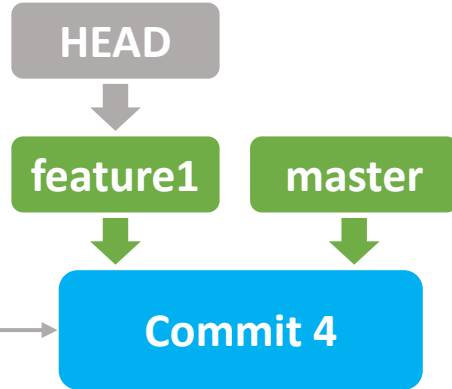


- **Branch**
 - -d



Branches | Переключение между коммитами (ветками)

git checkout feature1

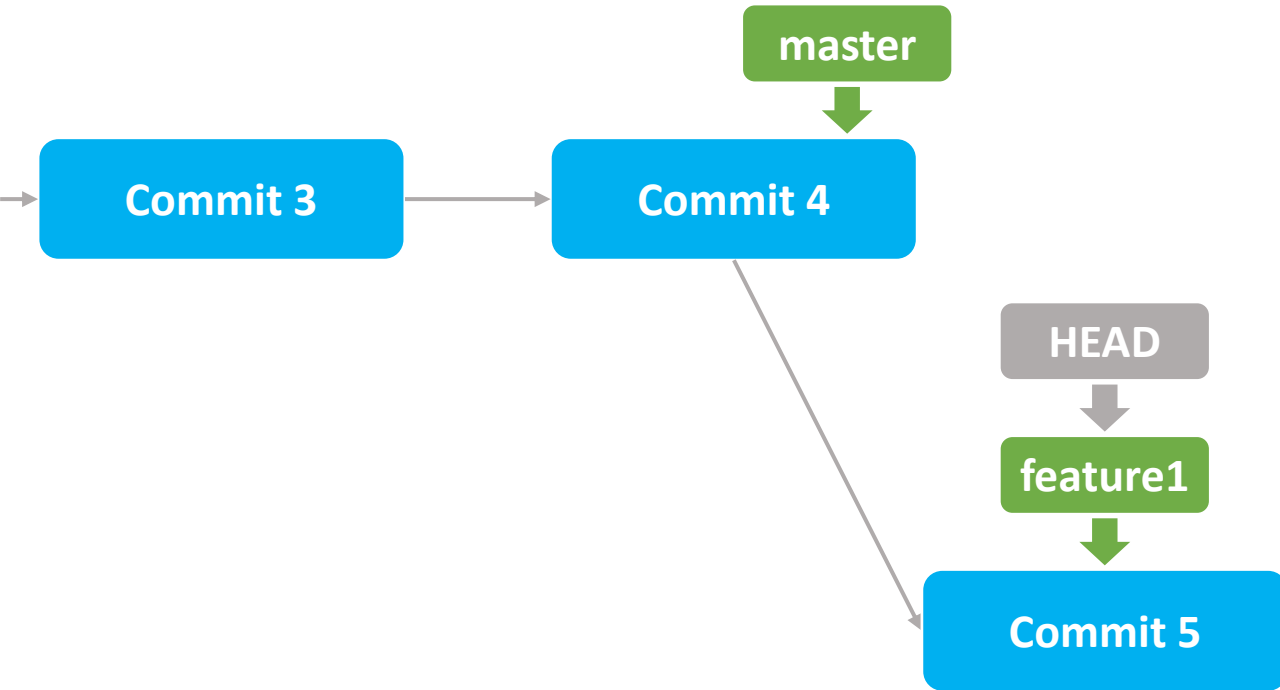


- **Checkout**
 - -b branch commit_hash
 - filename



Branches | Пример: делаем commit в feature-ветку

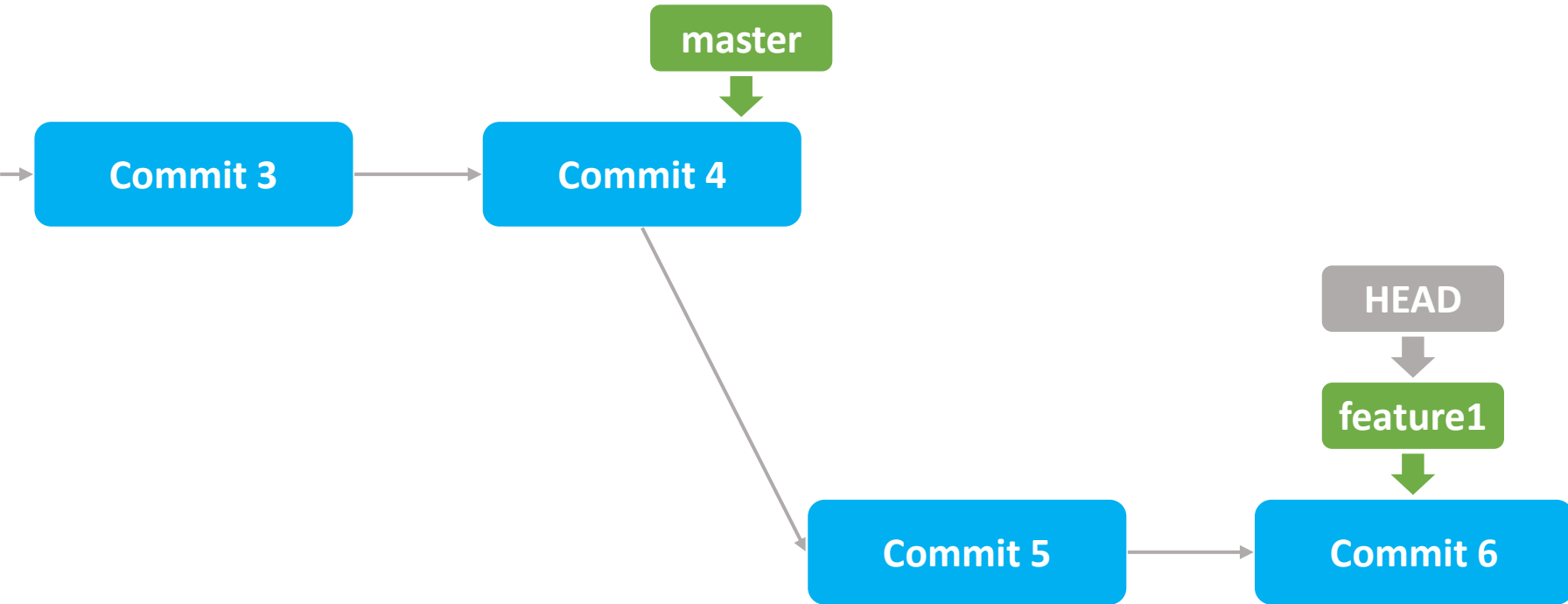
git commit ...





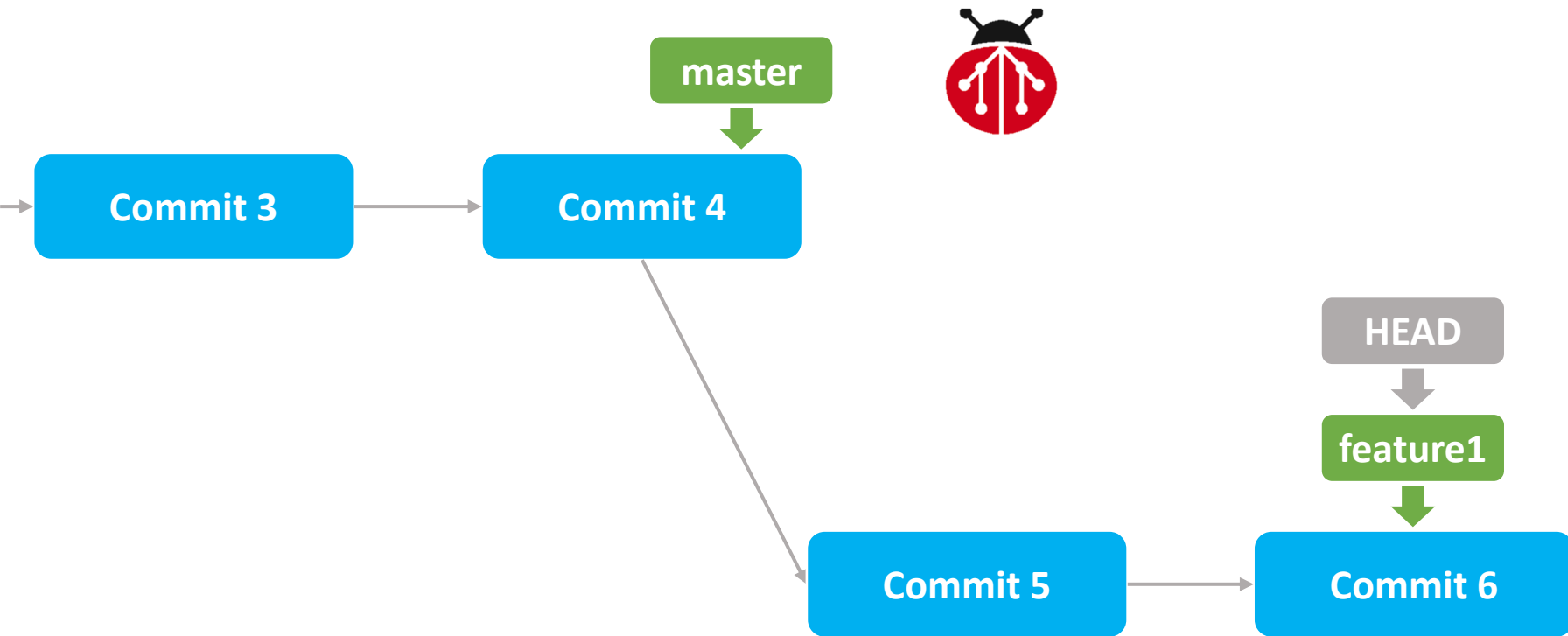
Branches | Пример: делаем commit в feature-ветку

git commit ...





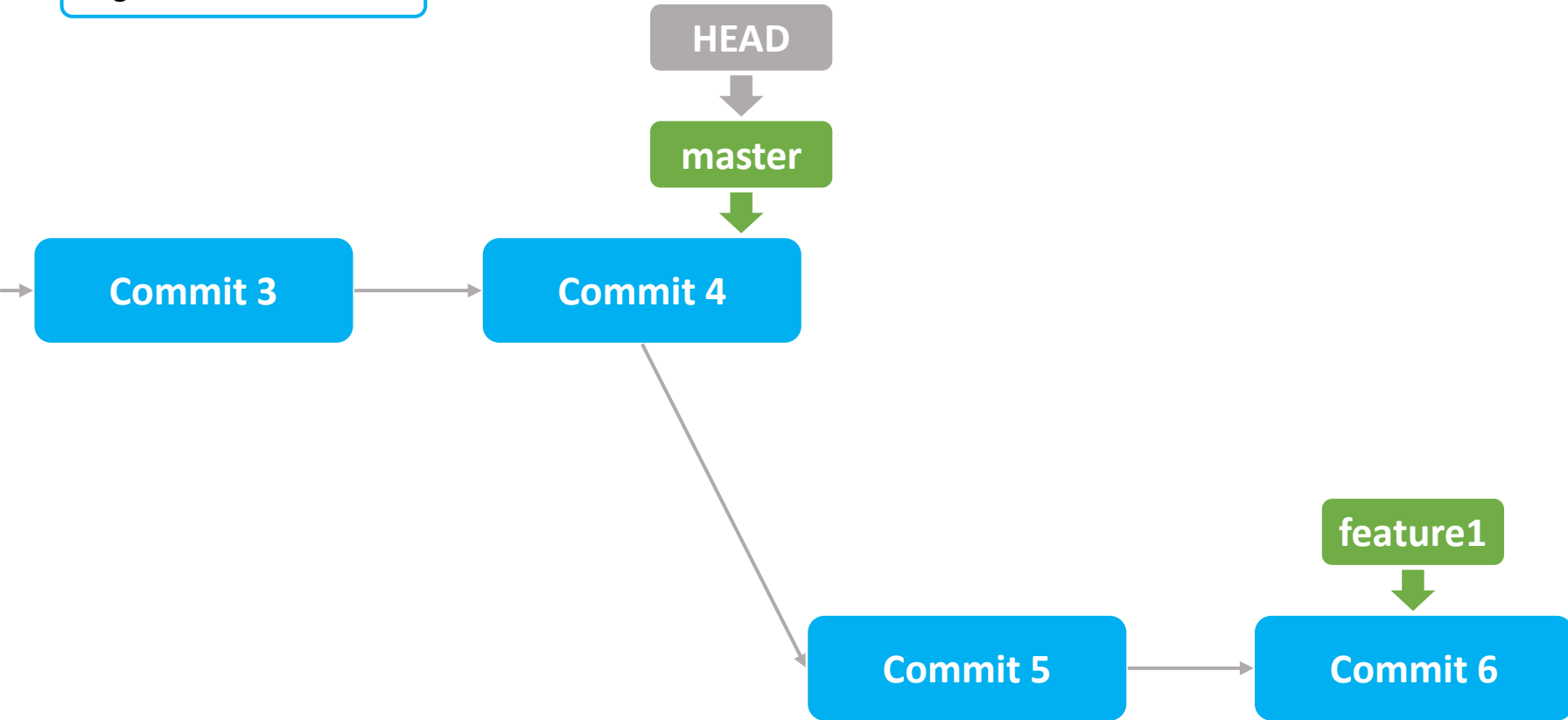
Branches | Пример: находим баг





Branches | Пример: переключаемся в мастер

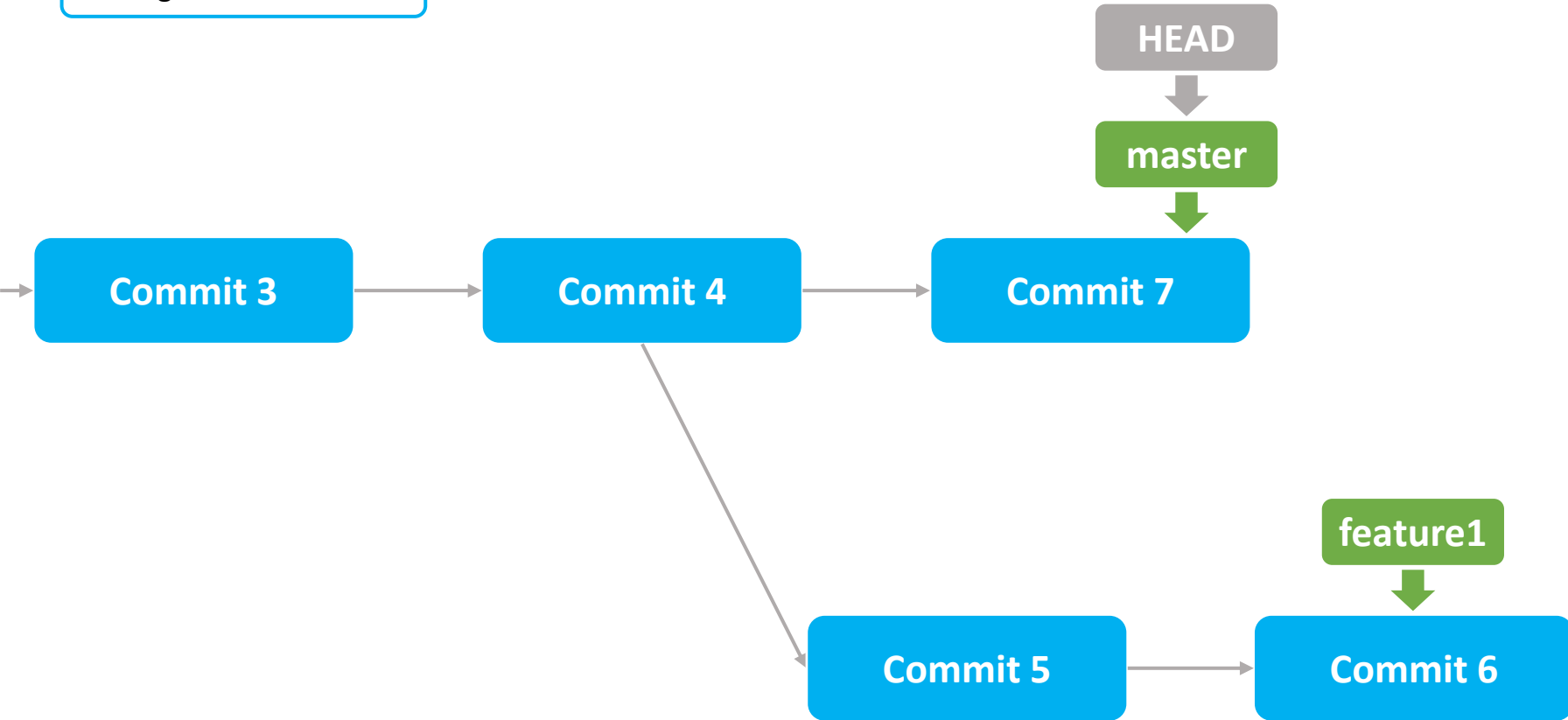
git checkout master





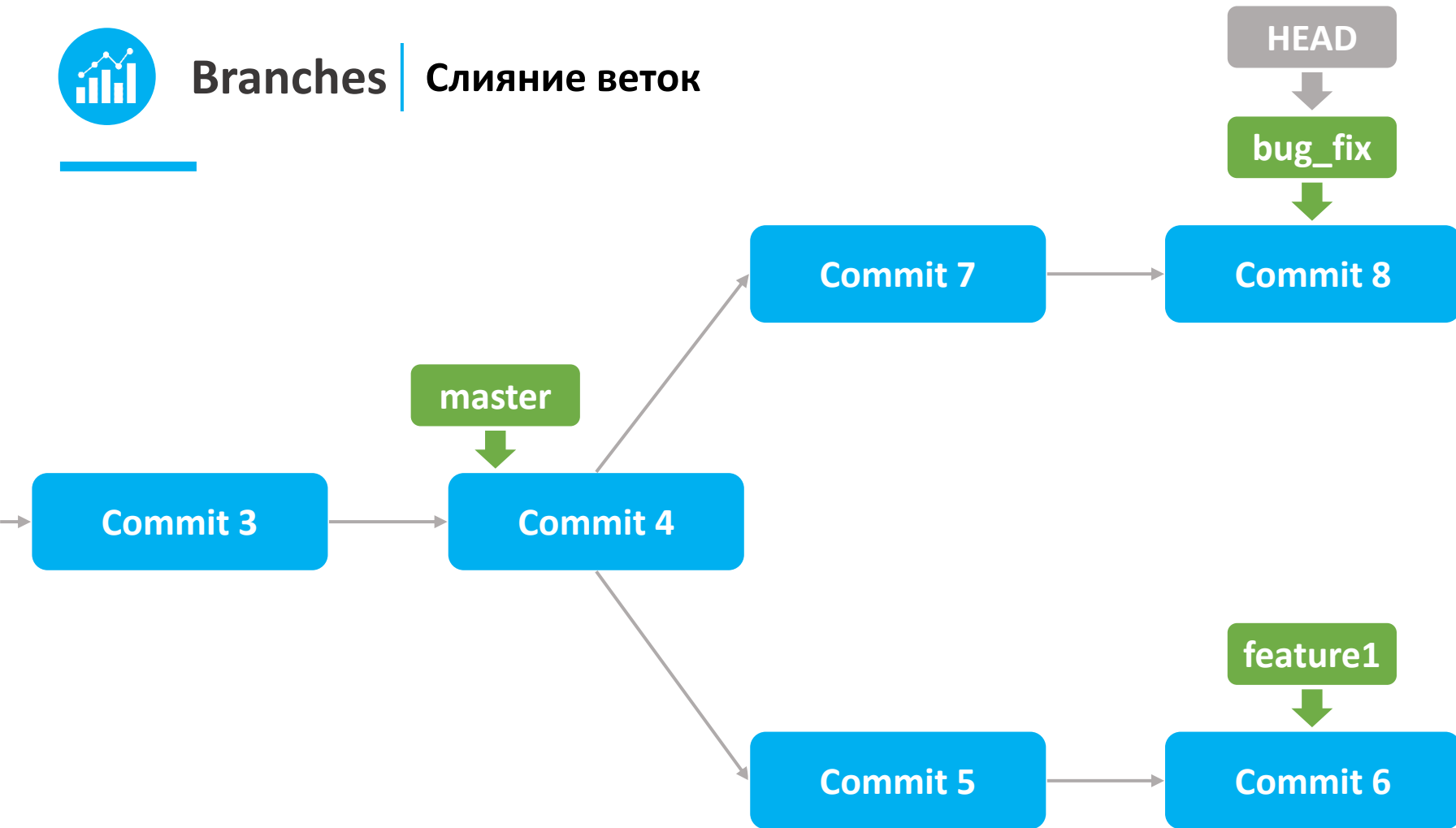
Branches | Пример: делаем commit исправлений

git commit ...





Branches | Слияние веток





Branches | Слияние веток

git checkout master

HEAD

master

Commit 3

Commit 4

Commit 7

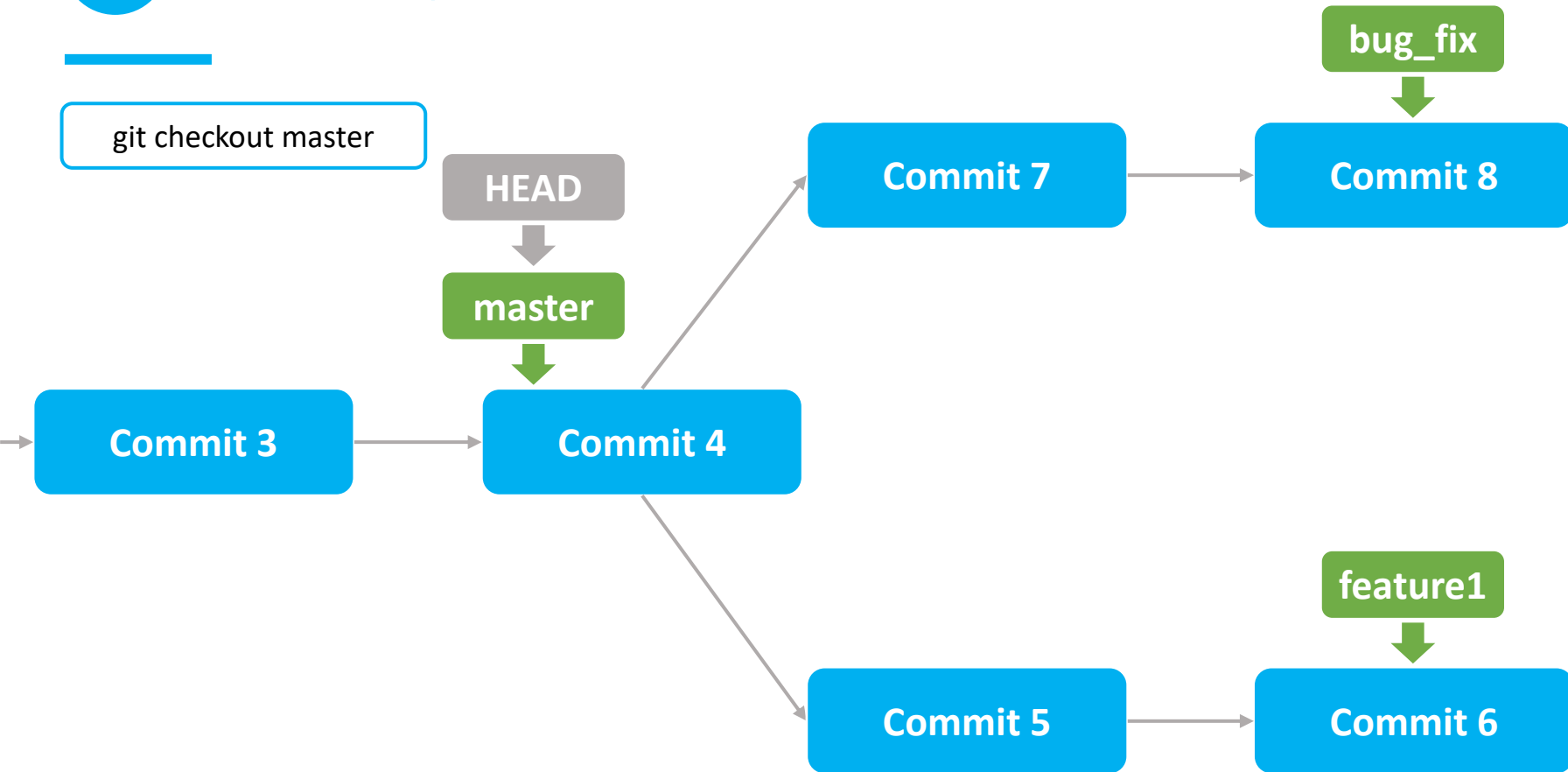
bug_fix

Commit 8

feature1

Commit 5

Commit 6





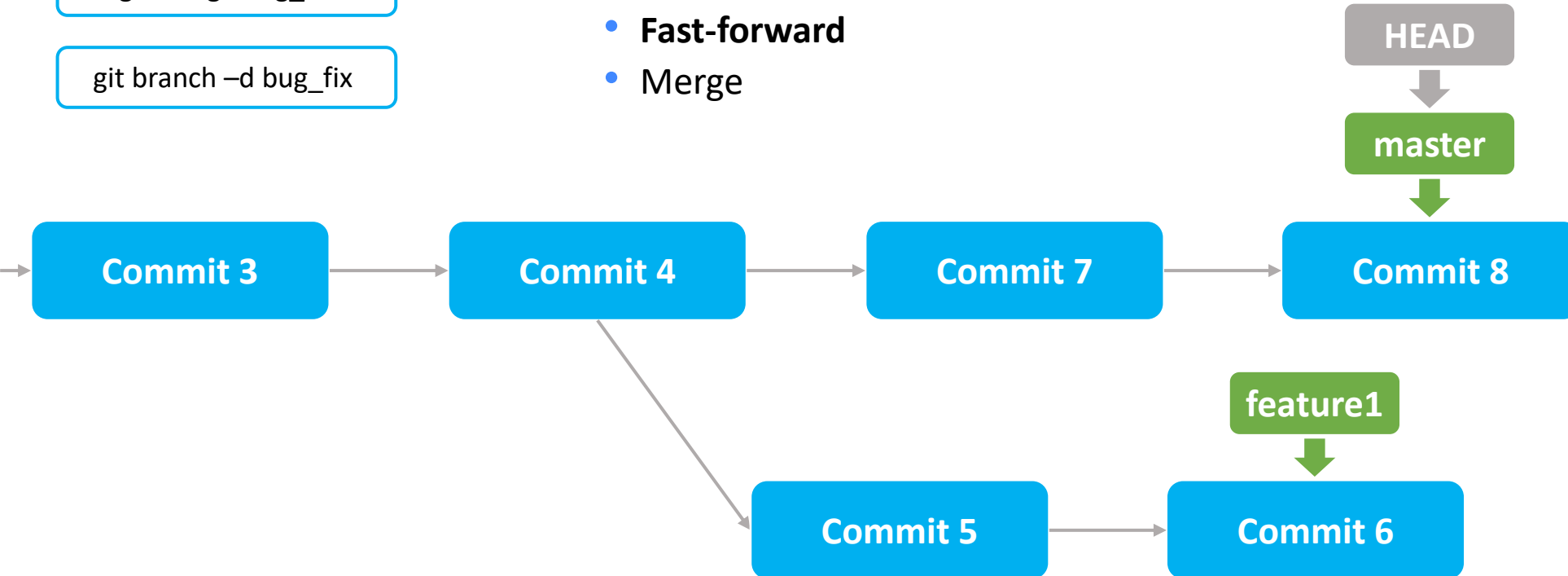
Branches

Слияние веток: Fast-forward и удаление старой ветки

```
git merge bug_fix
```

```
git branch -d bug_fix
```

- Merge
 - Fast-forward
 - Merge

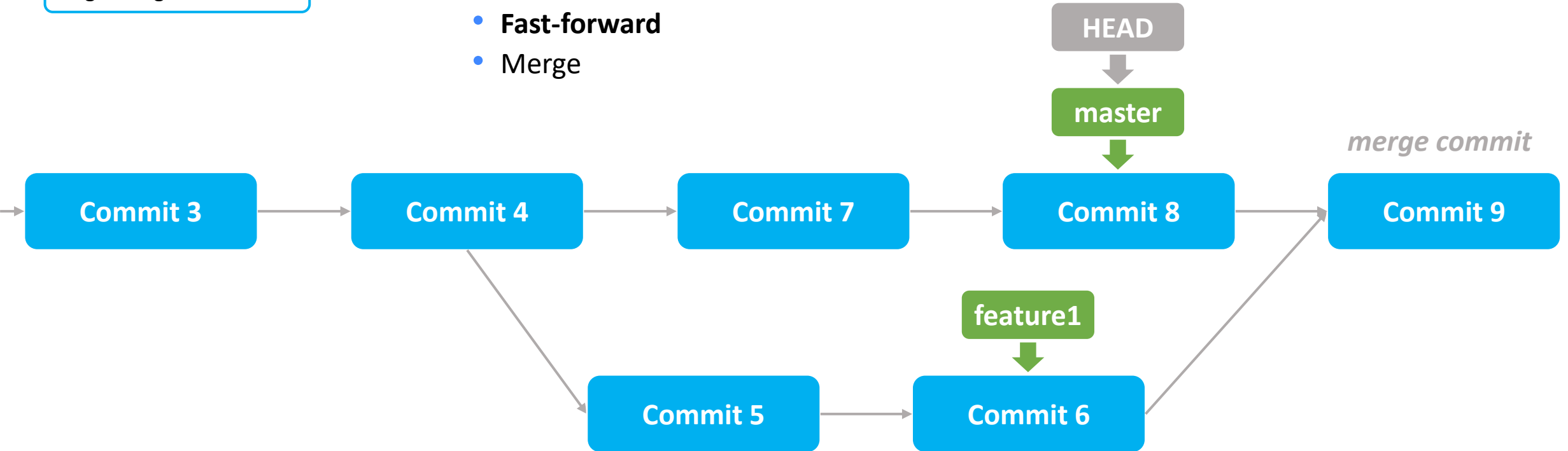




Branches | Слияние веток: Fast-forward merge

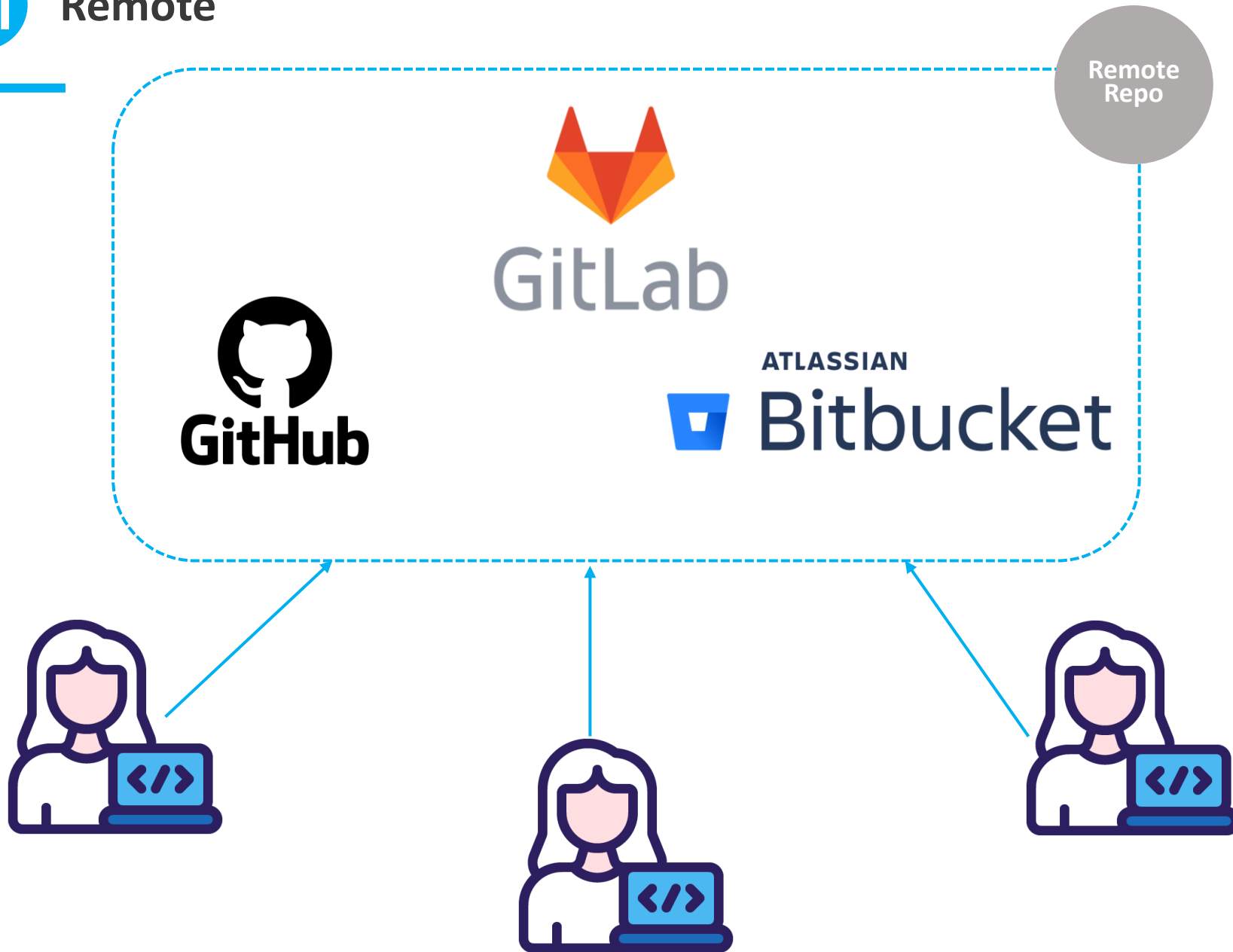
git merge feature1

- Merge
 - Fast-forward
 - Merge





Remote



Преимущества

- Надежность хранения
- История коммитов
- Детали коммитов
- Настройки доступа
- Merge из UI
- Code review
- Автоматизация CI/CD



Уже имеем локальный репозиторий

- `git init`
- `git remote add origin git@gitlab.com:m1f/mlops-course.git`
- `git add .`
- `git commit -m "Initial commit"`
- `git push -u origin main`

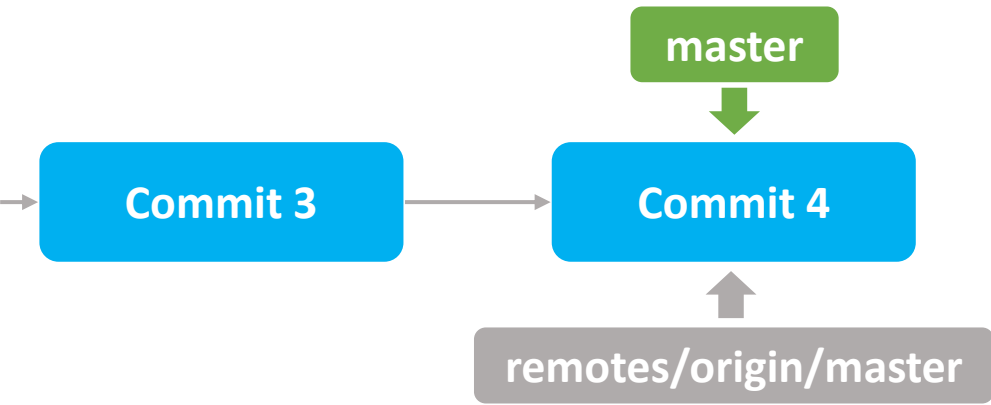
Клонируем существующий

- `git clone git@gitlab.com:m1f/mlops-course.git`



Remote | Указатель удаленного репозитория

```
git remote add origin...
```

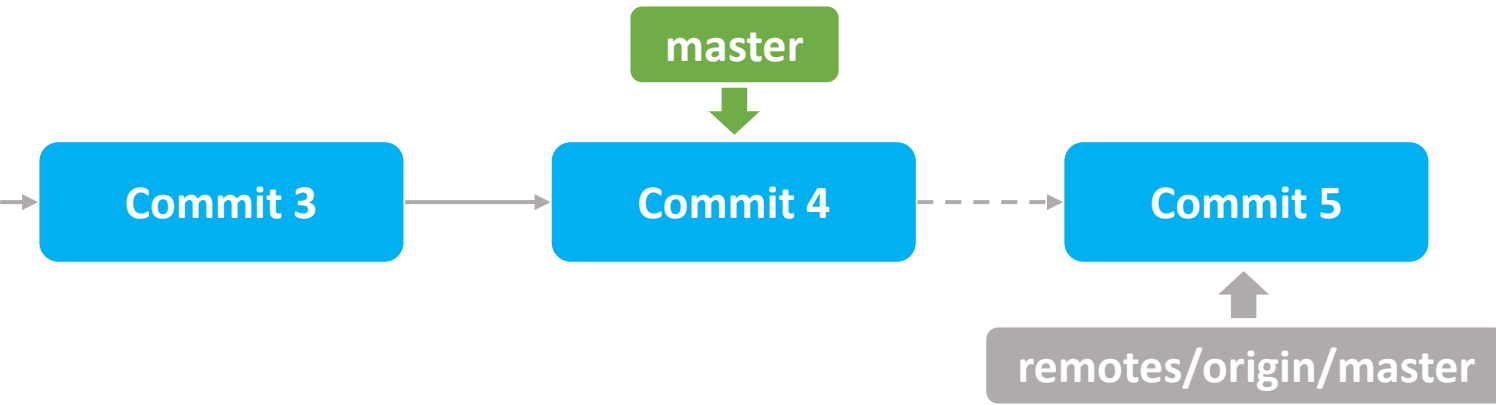




Remote

Обновление указателя удаленного репозитория и скачивание коммитов

`git fetch origin`





Remote | Производим слияние

```
git merge origin/master
```





Remote | Git pull

`git pull origin master`

=

`git fetch origin`

+

`git merge origin/master`



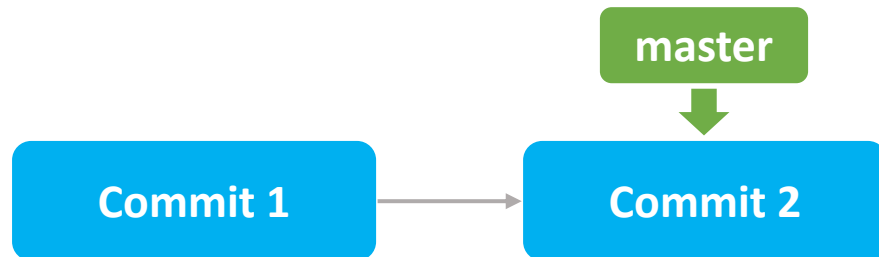
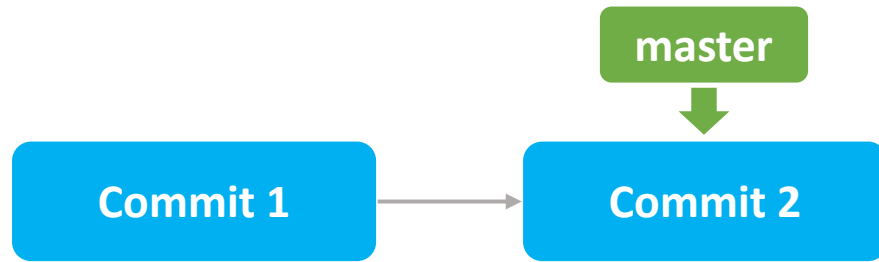
Remote | Получение новых правок

- `git push`
 - `--set-upstream (-u) origin main`
 - `--force`





Remote | Кейс: 2 разработчика





Remote | Кейс: 2 разработчика



Commit 1

Commit 2

master

Commit 3

git commit

Commit 1

origin/master

Commit 2





Remote | Кейс: 2 разработчика



Commit 1

Commit 2

master

Commit 3

git push

Fast forward

origin/master

Commit 1

Commit 2

Commit 3





Remote | Кейс: 2 разработчика



Commit 1

Commit 2

origin/master

Commit 3

git push



Commit 1

Commit 2

master

Commit 4





Remote | Кейс: 2 разработчика



Commit 1

Commit 2

origin/master

Commit 3

git pull



Commit 1

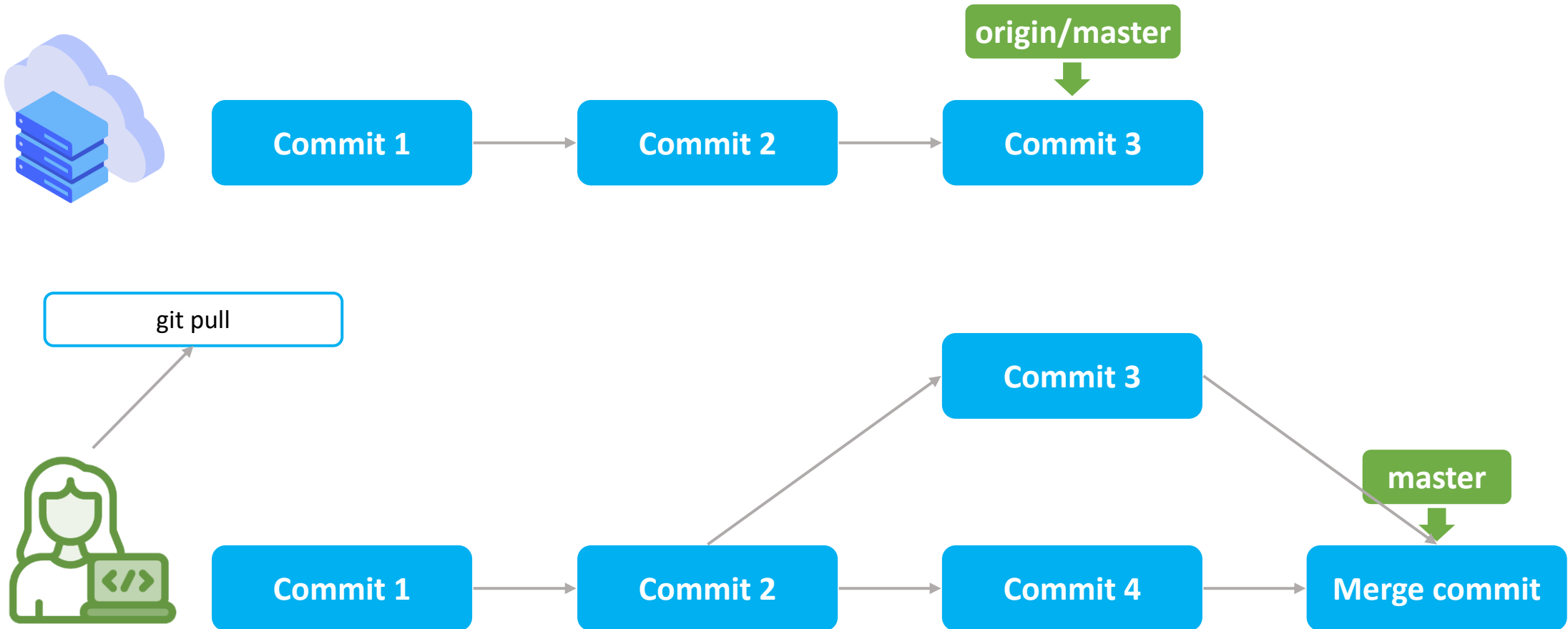
Commit 2

Commit 3

Commit 4

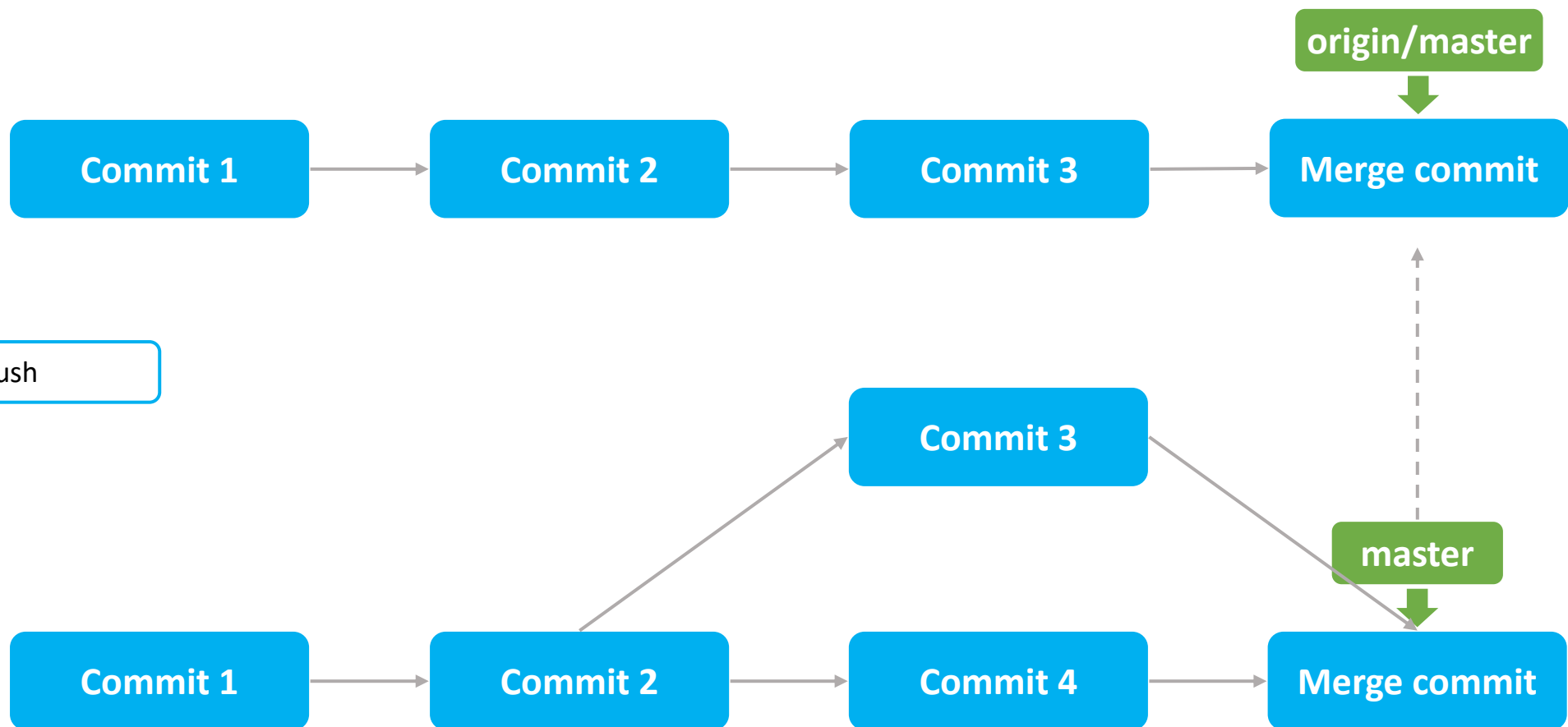
master

Merge commit





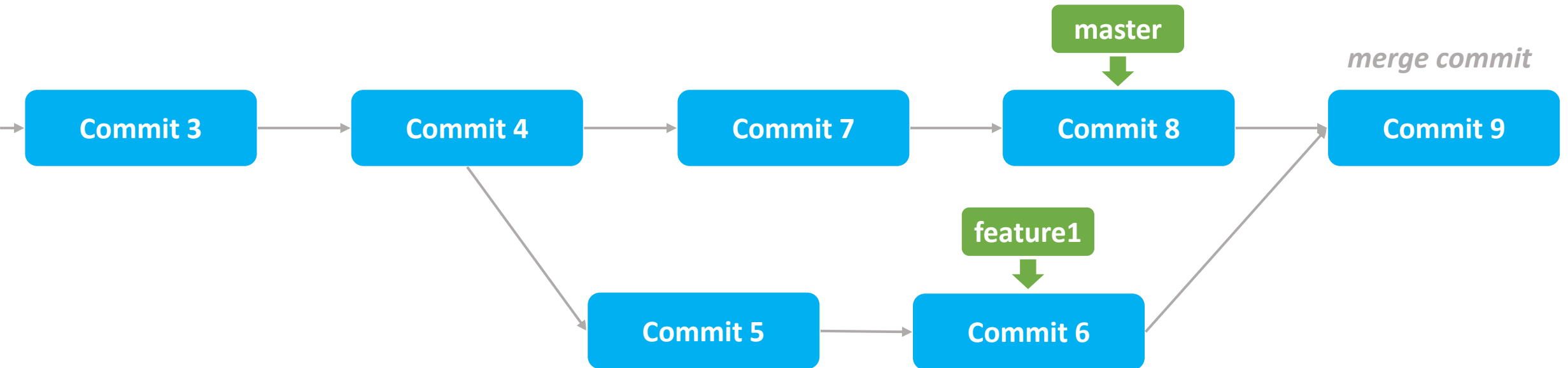
Remote | Кейс: 2 разработчика





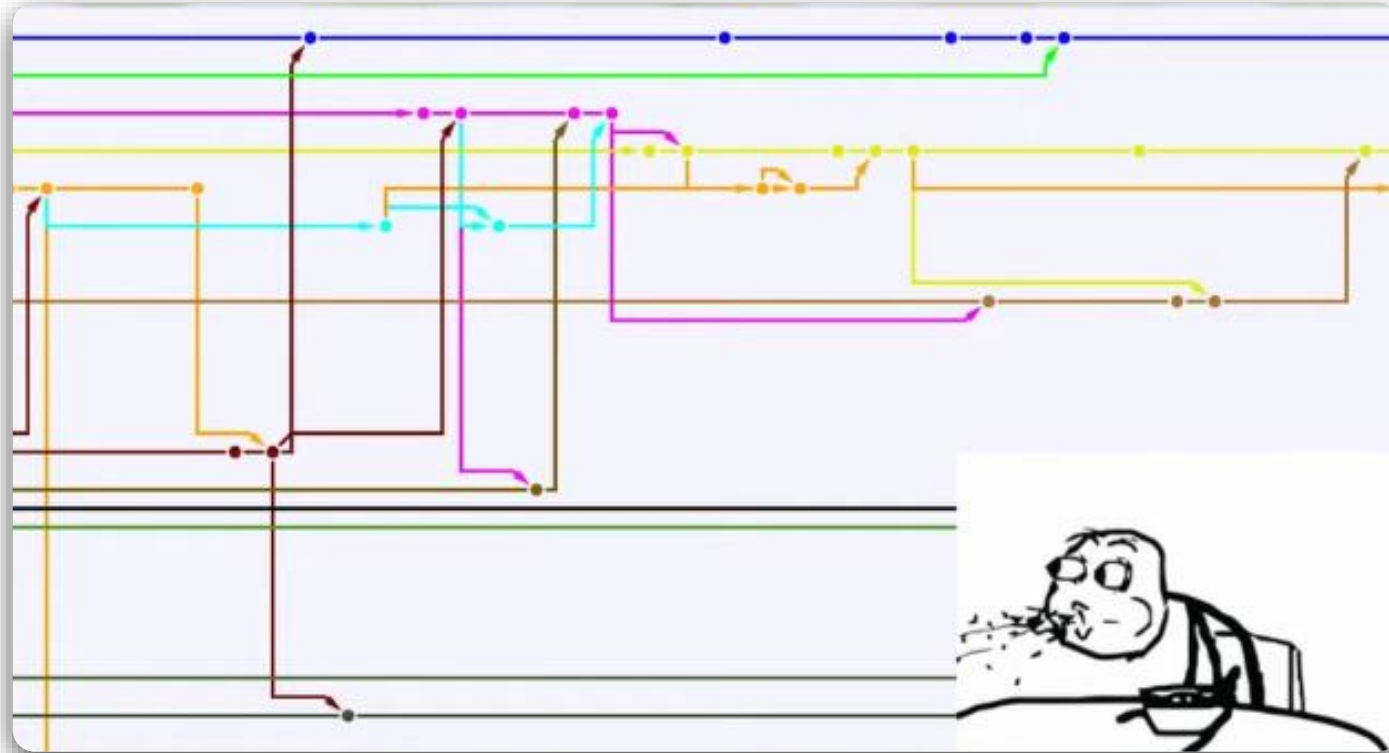
Remote

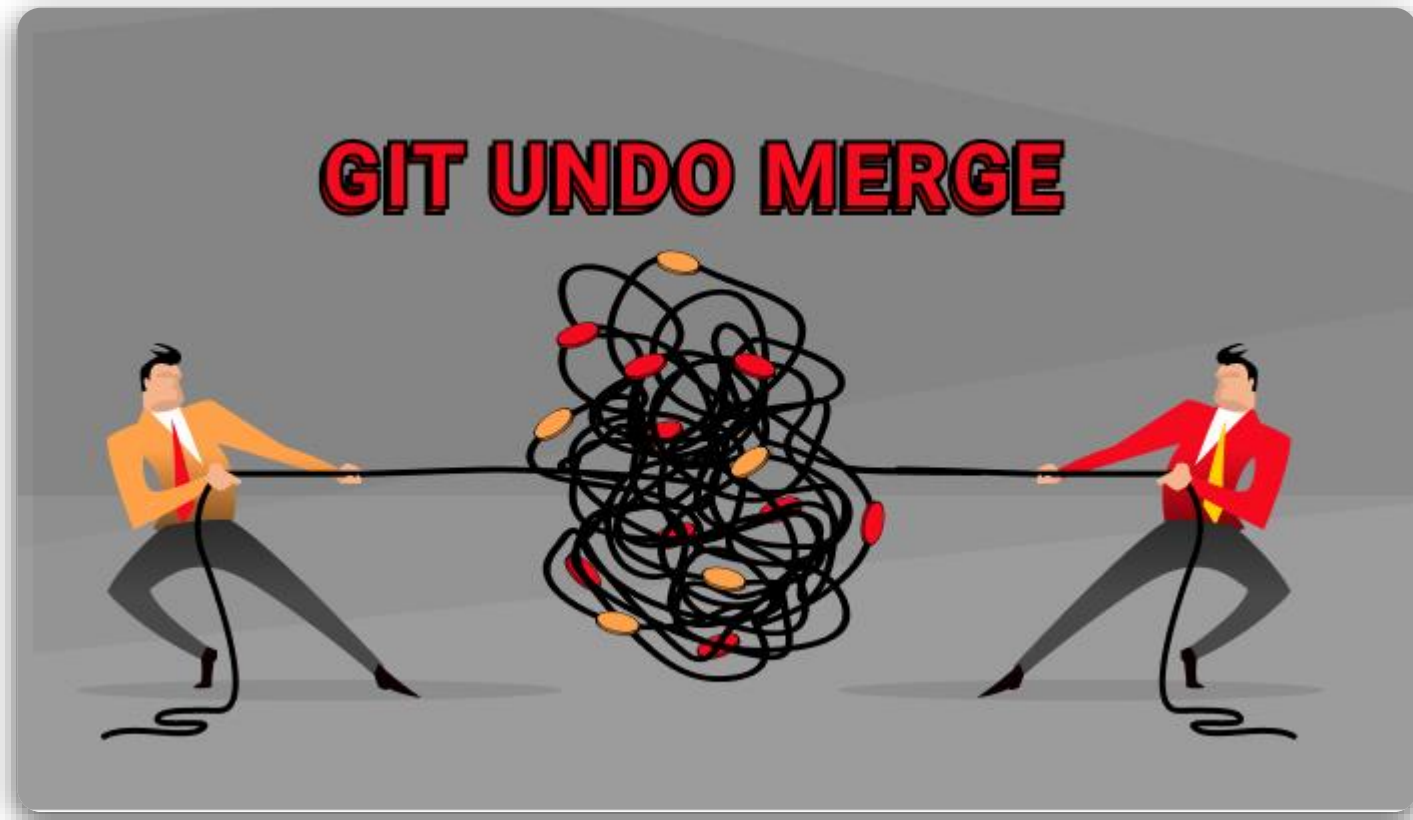
Проблема merge





Remote | Проблема merge





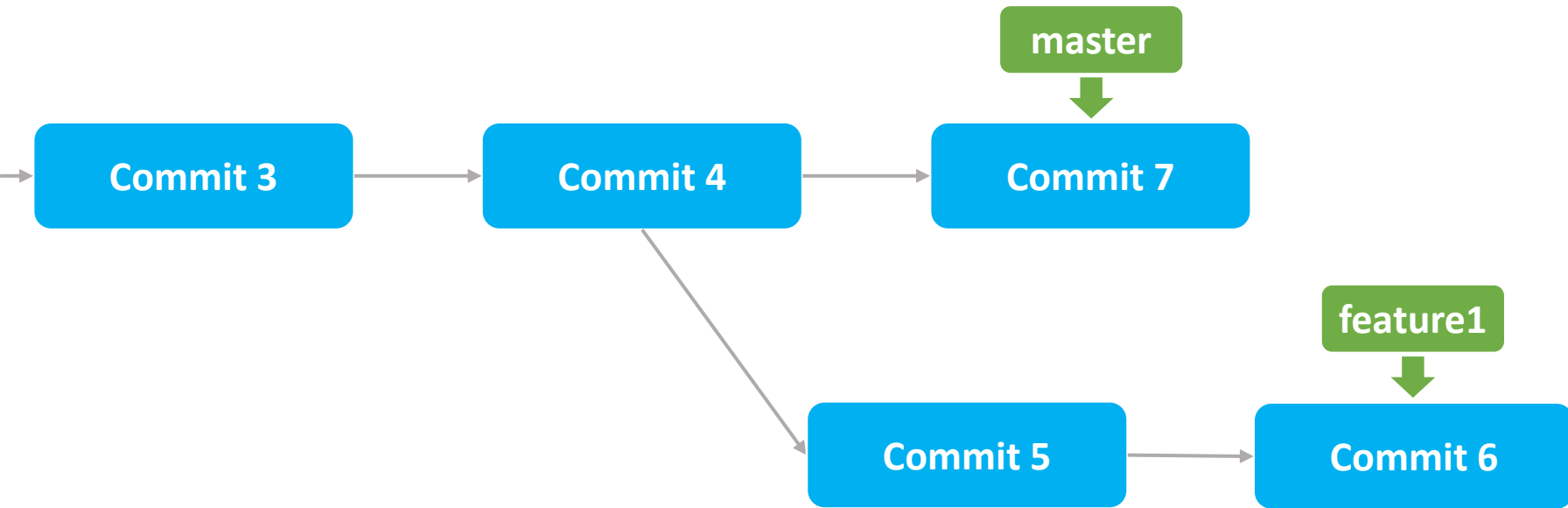


Remote | Rebase



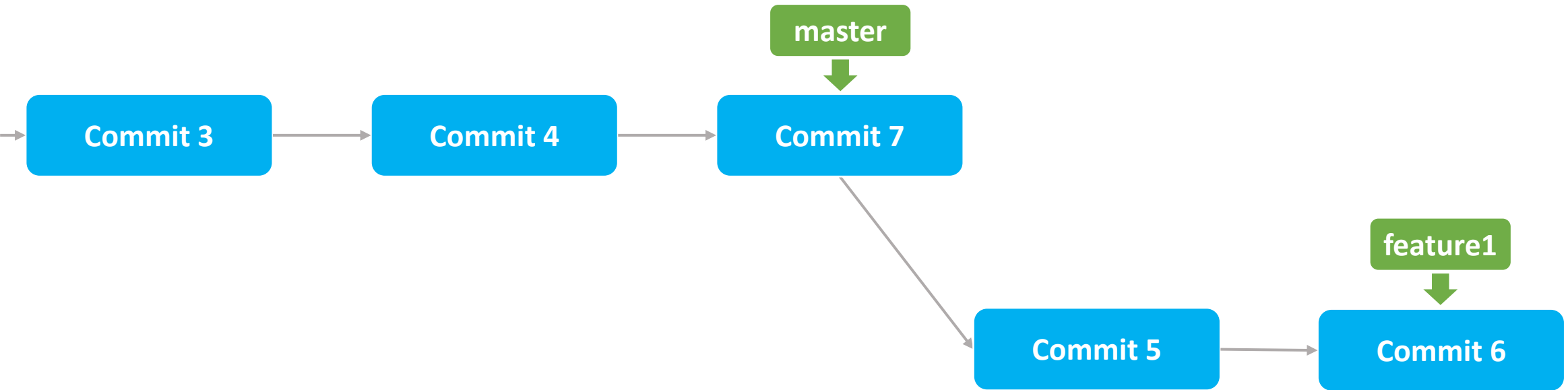


Remote | Rebase



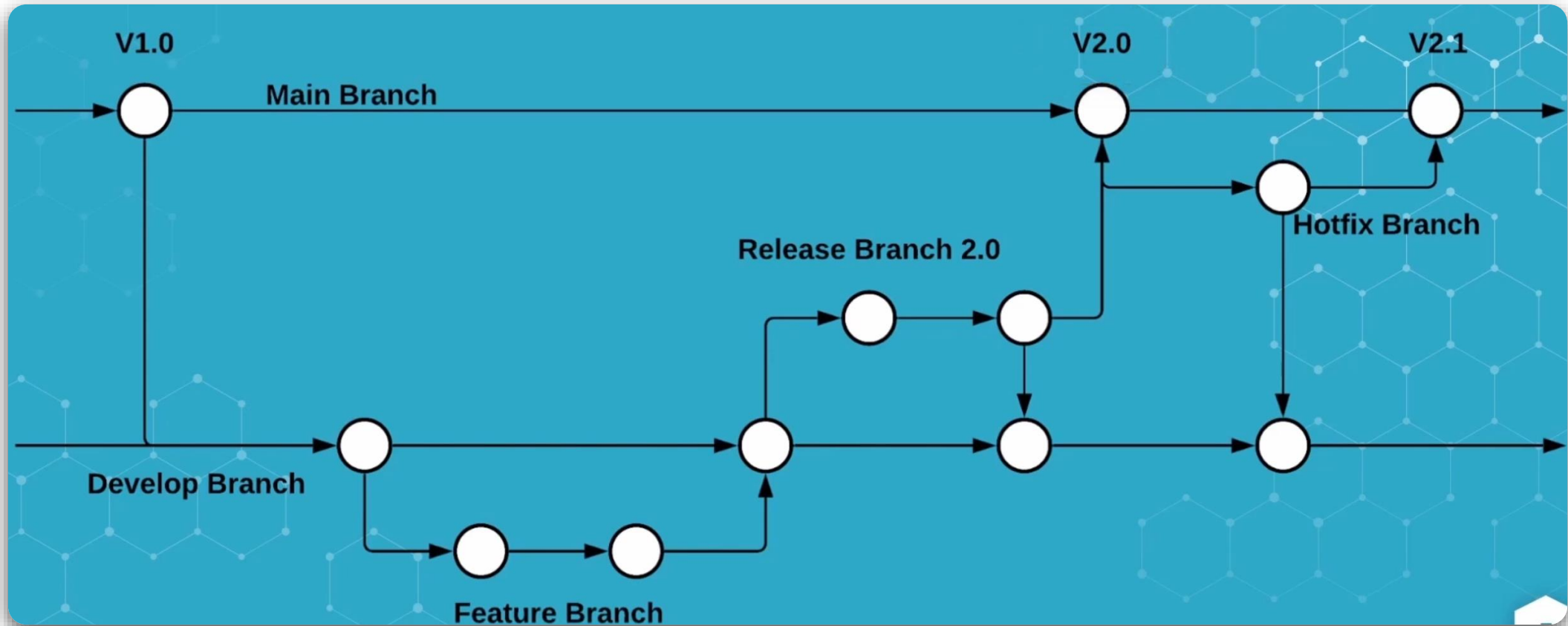


Remote | Rebase





Стратегии ветвления | Git Flow





Стратегии ветвления | Git Flow

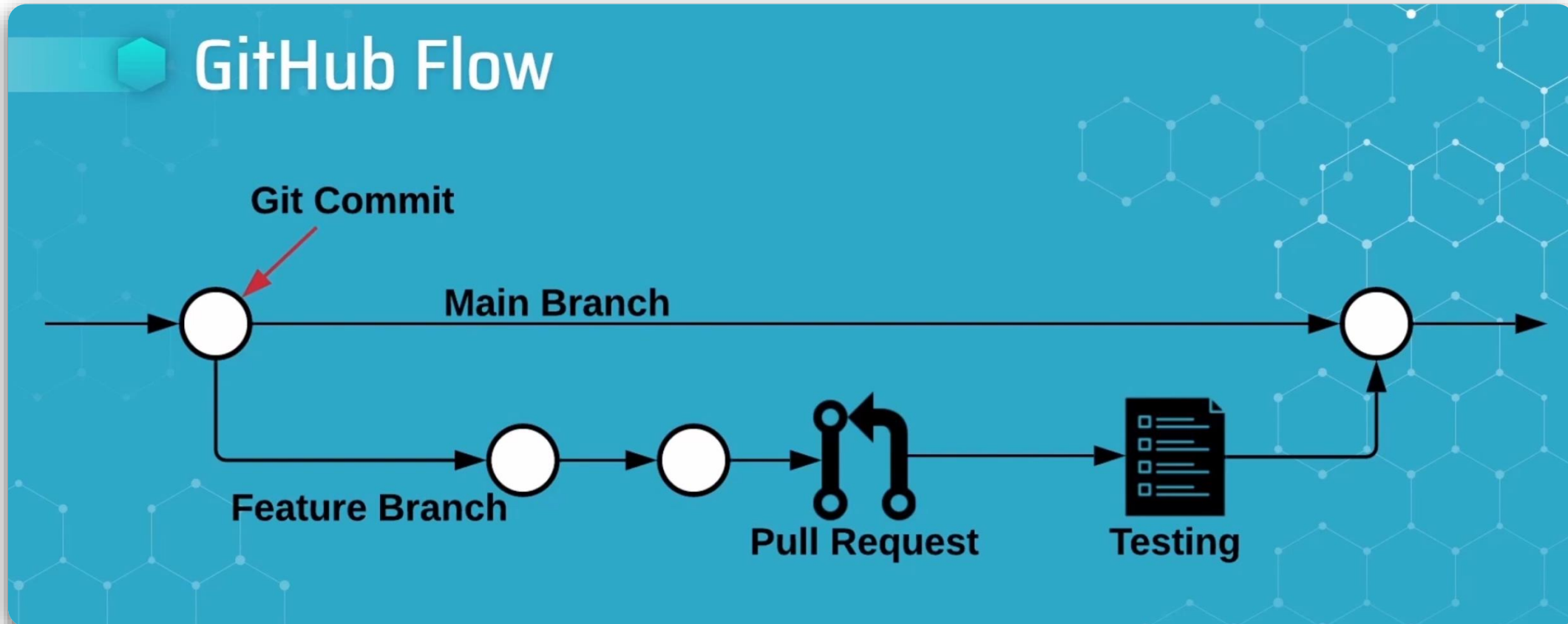
Плюсы

- Удобен для распределенных команд
- Подходит для редких релизов
- Несколько версий продукта
- Надежен



Минусы

- Неудобен для частого CD
- Сложная история коммитов
- Сложный merge крупных фичей



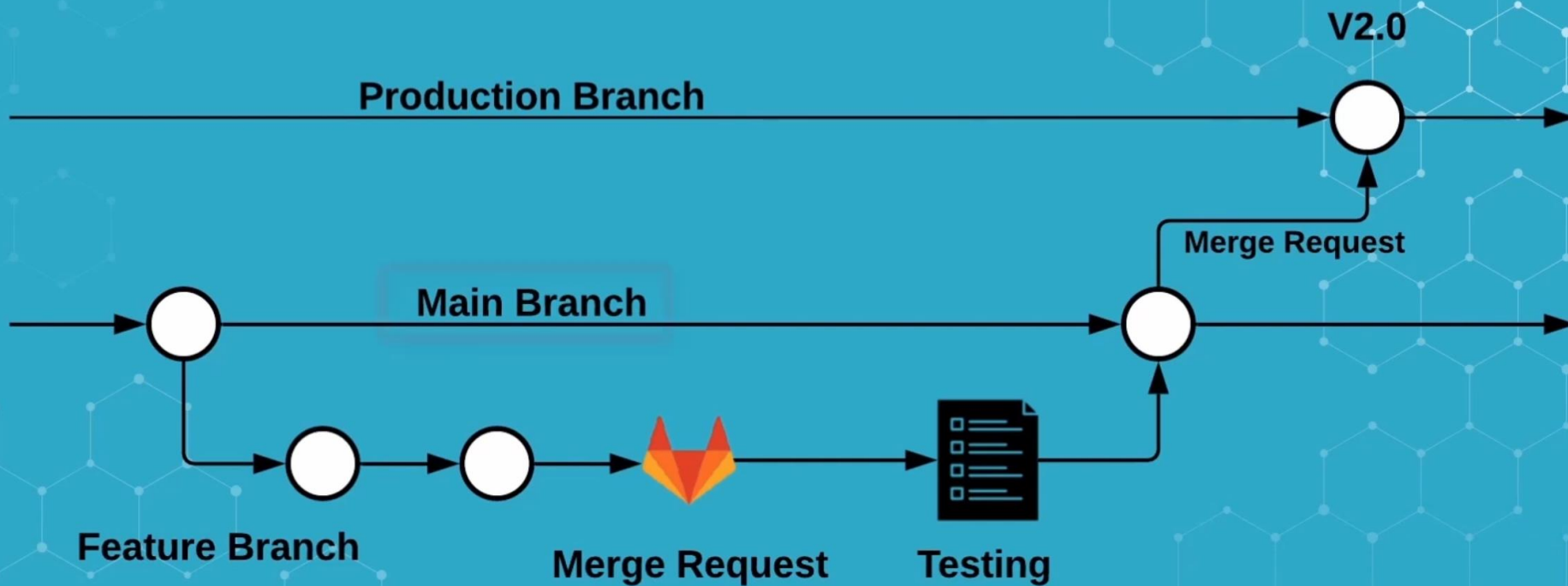


Deploy

- Закрываем ветку master
- Merge master в deploy ветку
- Build/test
- Deploy в пре-прод
- Deploy в прод
- Мониторинг
- PR в master
- Открываем мастер



GitLab Flow - Environment Branches





SemVer

1 . 3 . 1

BREAKING . FEATURE . FIX

incompatible
API changes

breaking
change

add backwards-
compatible
functionality

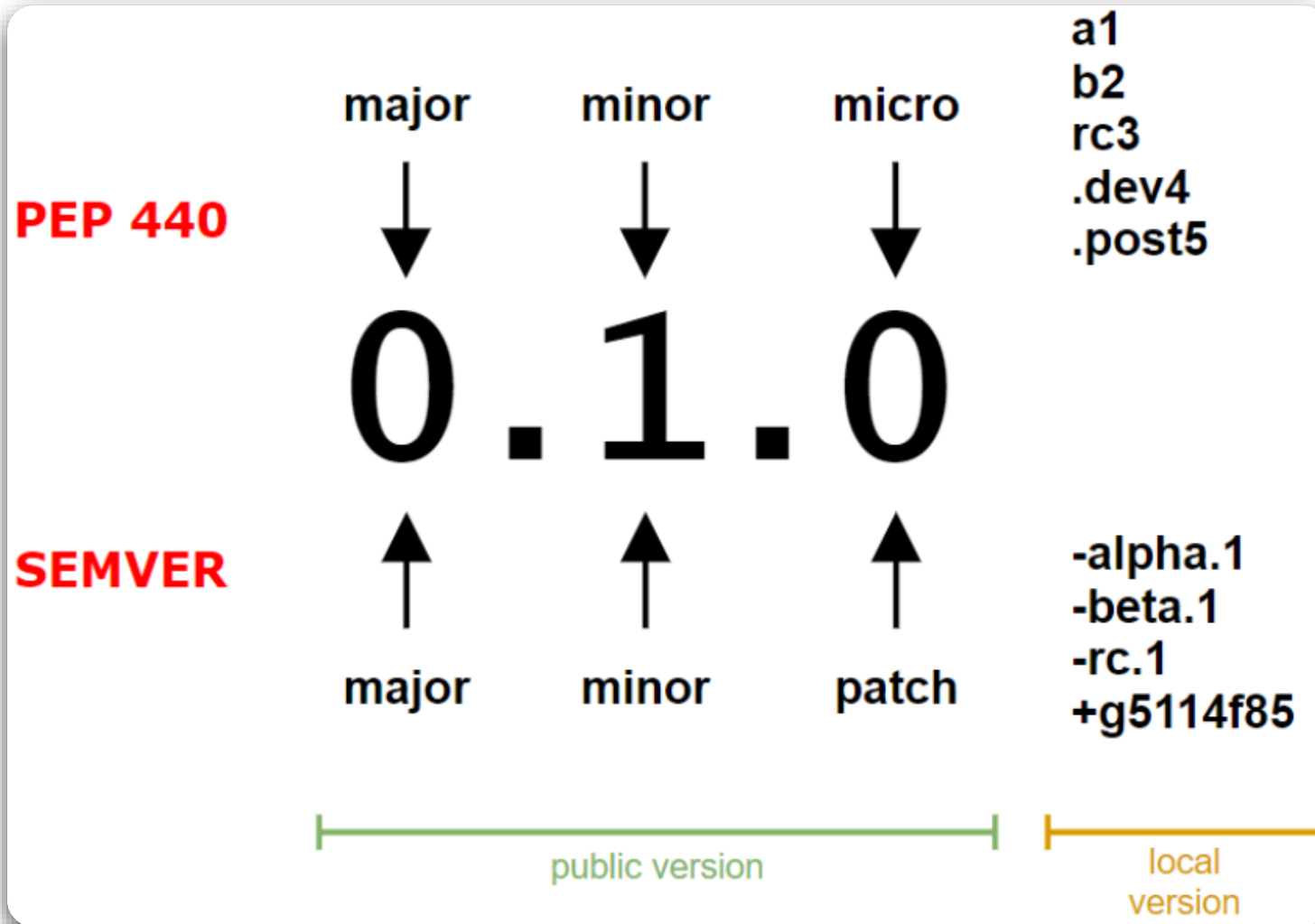
new
feature

make backwards-
compatible bug fix

bug
fix



SemVer | PEP440





SemVer

@semantic-release/gitlab

[semantic-release](#) plugin to publish a [GitLab](#) release.

🔄 Test passing npm@latest v8.0.2 npm@next v5.0.0

Step	Description
<code>verifyConditions</code>	Verify the presence and the validity of the authentication (set via environment variables).
<code>publish</code>	Publish a GitLab release.
<code>success</code>	Add a comment to each GitLab Issue or Merge Request resolved by the release.

Install

```
$ npm install @semantic-release/gitlab -D
```



Домашка

- Создать локальный репозиторий со своим проектом
- Настроить gitignore
- Склонировать код в удаленный репозиторий (GitLab)
- Отработать github-flow:
 - Добавить новый код и закоммитить его в удаленный репозиторий
 - Создать новую ветку и выполнить в нее коммит
 - Создать еще одну ветку и выполнить коммит в нее
 - Выполнить слияние первой ветки и master с помощью fast-forward
 - Выполнить слияние второй ветки и master с помощью merge
 - Создать конфликт и разрешить
 - Настроить добавление кода в ветвь master только через pull request
 - Добавить второго пользователя и добавить новый код через механизм pull request



MLOps и production подход к ML исследованиям

Хранение и версионирование кода с git

Павел Кикин

Газпромнефть ЦР

Руководитель направления NLP

t.me/pavel_kikin

