



**git**

# Andrea Leopardi

<mailto:an.leopardi@gmail.com>

[@whatyouhide](#)

<https://github.com/whatyouhide>

# Cos'è git?

Git è un *distributed version control system*

**open source**

**veloce**

**potente**

# Centralized Version Control Systems (svn)

Vantaggi (rispetto a sistemi più primitivi)

- Ogni developer si riferisce alla stessa versione del progetto
- Ogni developer nel progetto ha un'idea del lavoro che stanno facendo gli altri
- Gli amministratori del progetto hanno controllo sulla repository

# Centralized Version Control Systems (svn)

Svantaggi (rispetto a VCS distribuiti)

- Il server è un single point of failure
  - **va down**: nessun download/upload di codice
  - **necessita backup**: rischio di perdere tutta la history del progetto
- Ogni developer dovrà sempre scaricare **tutta** la repository alla versione corrente + modificare + uploadare di nuovo

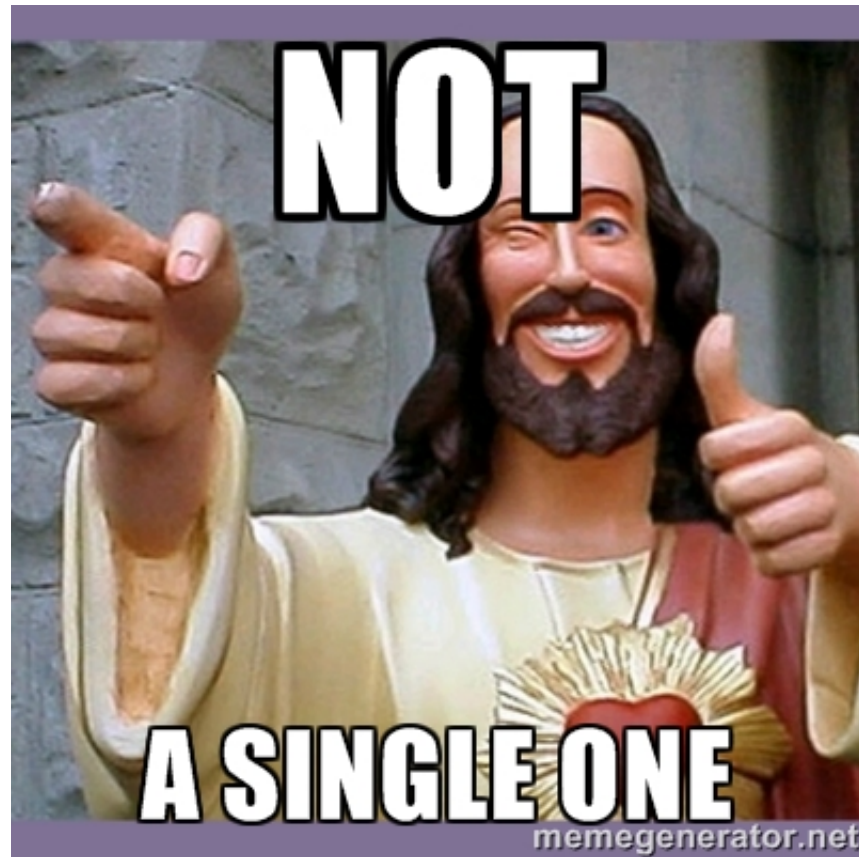
# Distributed VCS (git, mercurial)

## Vantaggi

- Ogni client è un *mirror* della repository (clonare significa clonare **tutta** la history)
- Si può lavorare (committare) in locale e uploadare/downloadare quando si ha una connessione
- Ogni client è un "backup" della repository → probabilità ~0 di perdere dati
- Quando si uploada/downloada codice, si muovono solo i cambiamenti

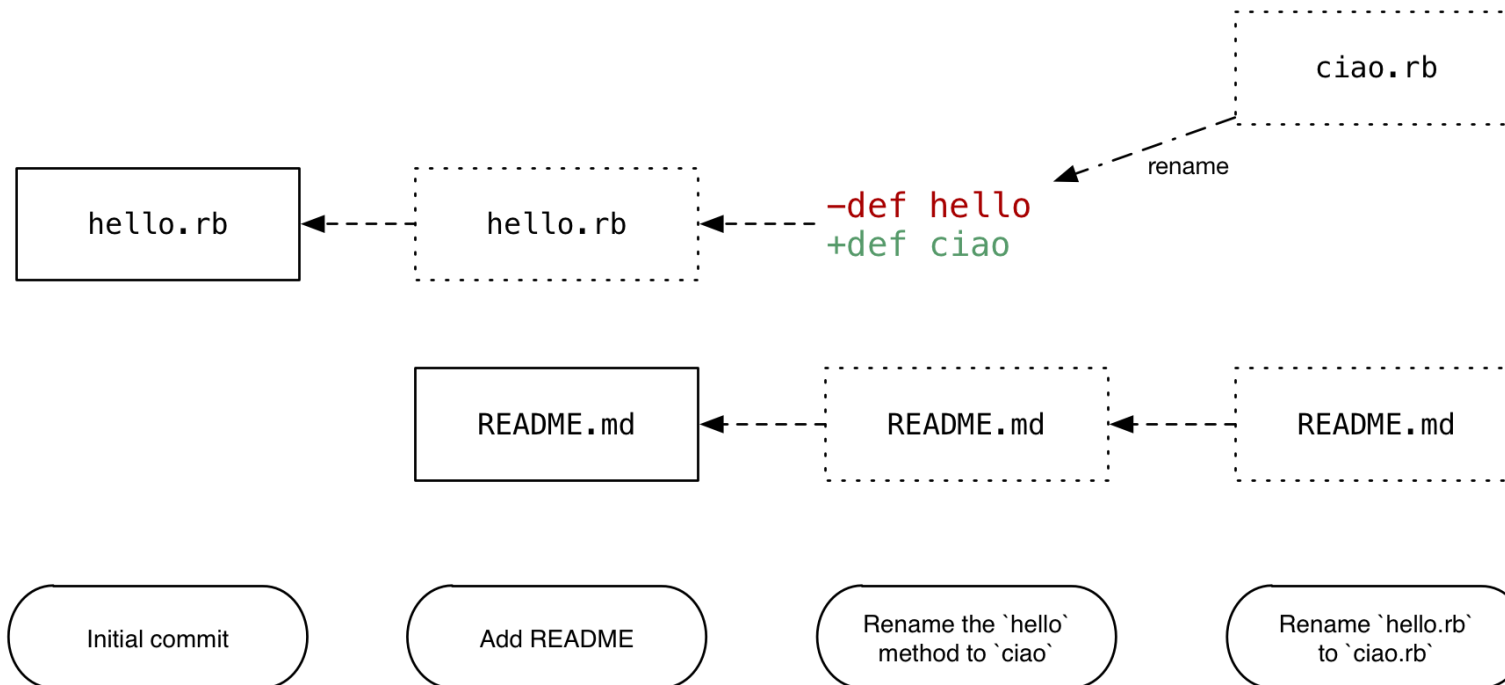
# Distributed VCS (git, mercurial)

Svantaggi



# Architettura di git





# Quasi tutte le operazioni sono locali

Non si ha quasi mai bisogno della rete per svolgere operazioni.

## Git ha *integrity*

- In git viene fatto un checksum (SHA1) di qualsiasi cosa (in modo da poter identificare univocamente quel qualcosa)
- È impossibile cambiare il contenuto dei file senza che git se ne accorga

b858a87c07b04c4568f51b0dce655f78d73c02b3

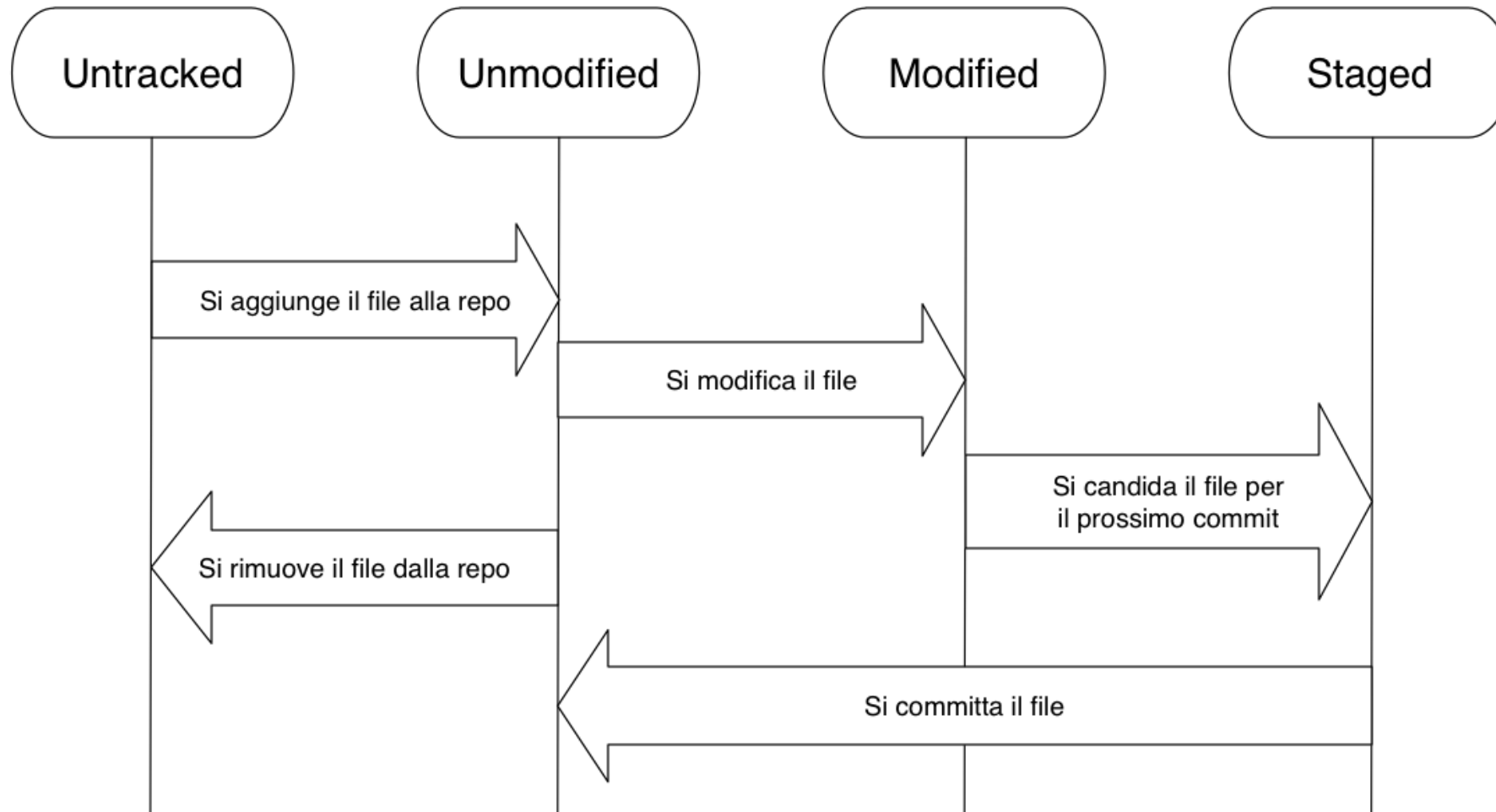
# Git (quasi sempre) aggiunge solo dati

Ci sono poche operazioni (spesso sconsigliate/pericolose) che costringono git a *cancellare* dati.

In genere qualsiasi operazione (anche di cancellazione) **aggiunge** dati alla repository.

# Lifecycle di un file

# Lifecycle di un file



# Commit

**12dc4e9...**

*parent commit*

*Tony Stark* <ironman@avengers.com>

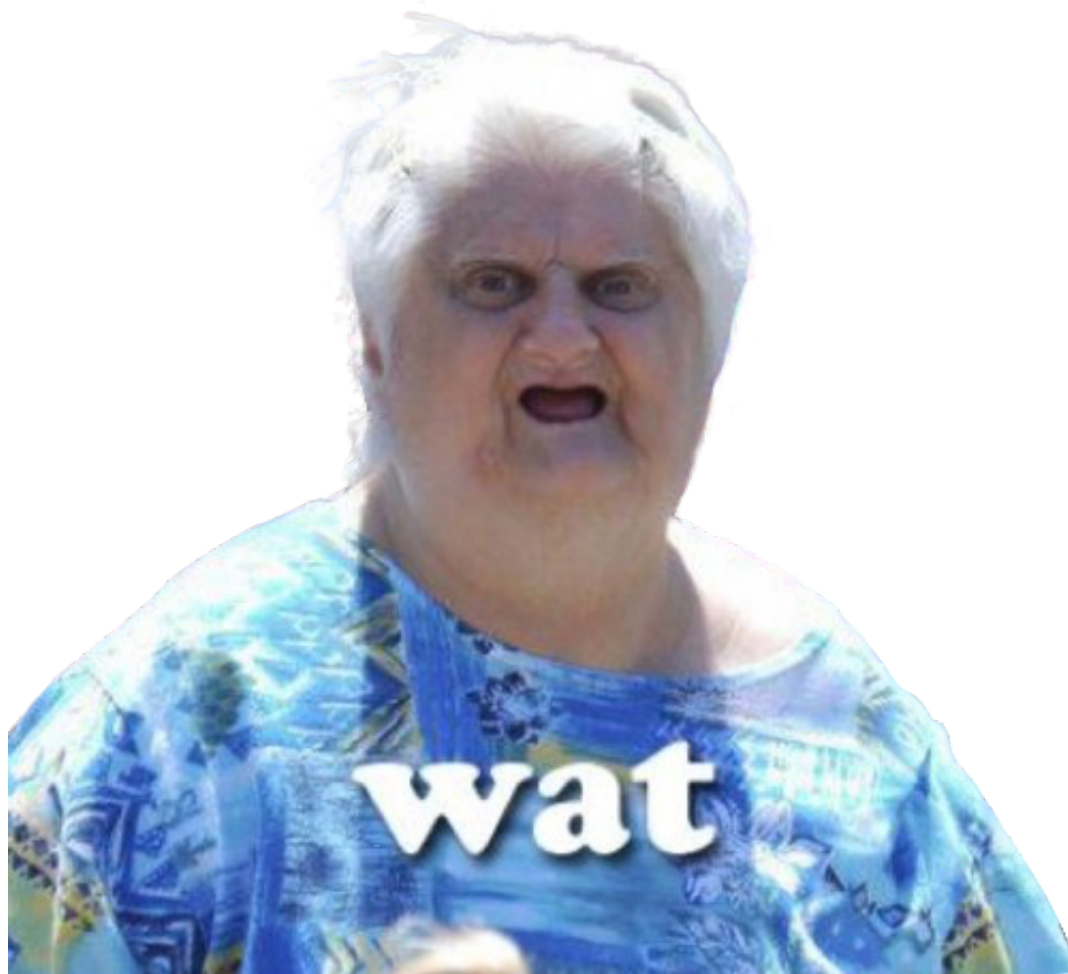
\* *tree con modifiche*

Add README.md...

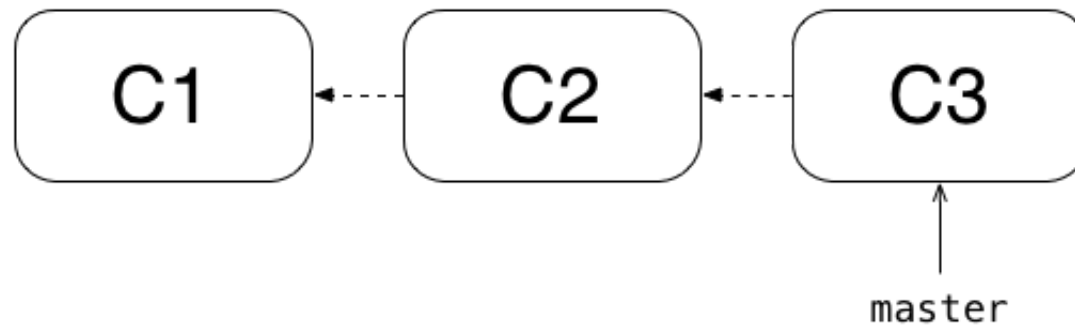


# Branching

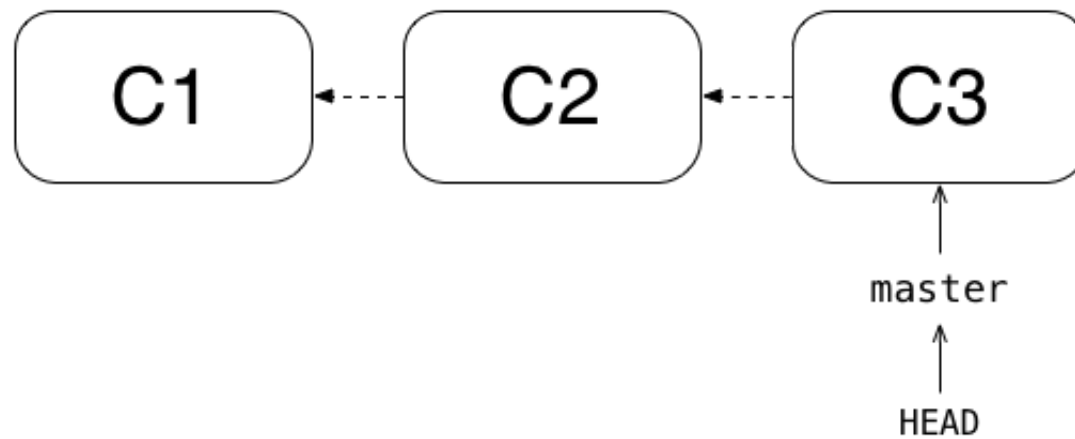
Un branch è un *puntatore a un commit* che permette di sviluppare più versioni della stessa repository parallelamente.



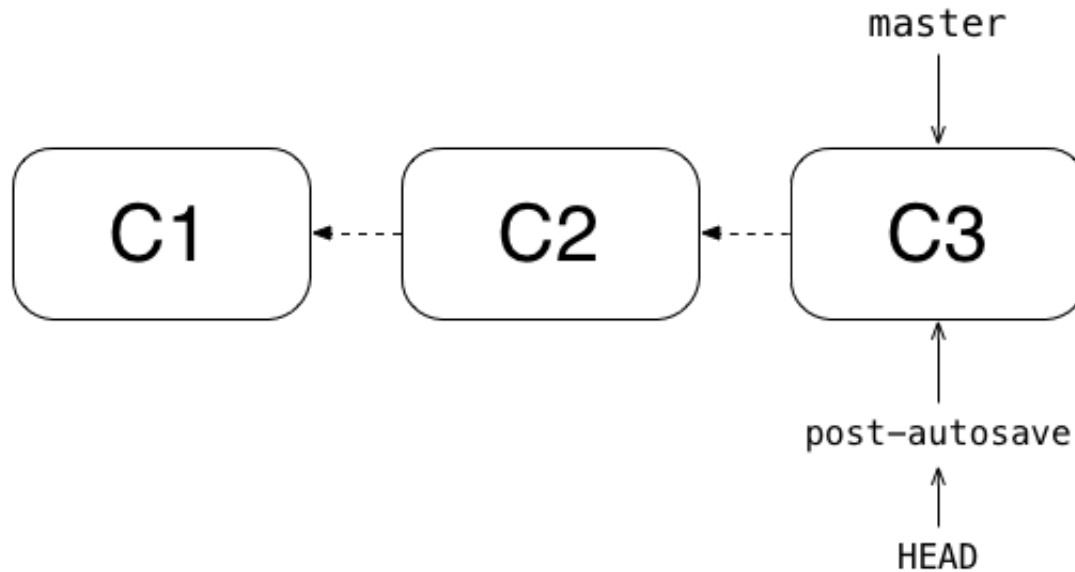
Di solito il branch principale si chiama **master**



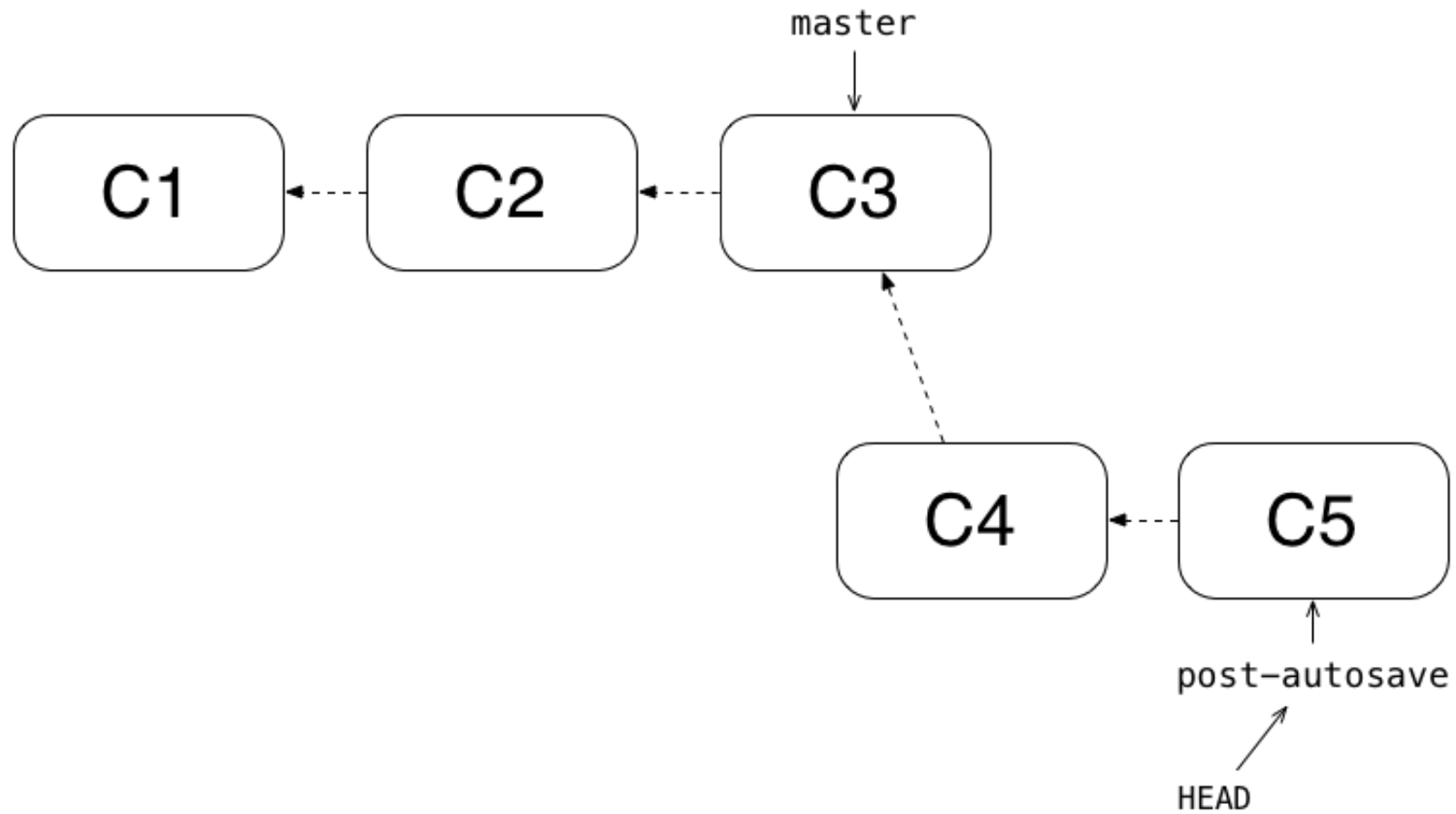
In ogni momento mi trovo in un branch: quello  
puntato da HEAD



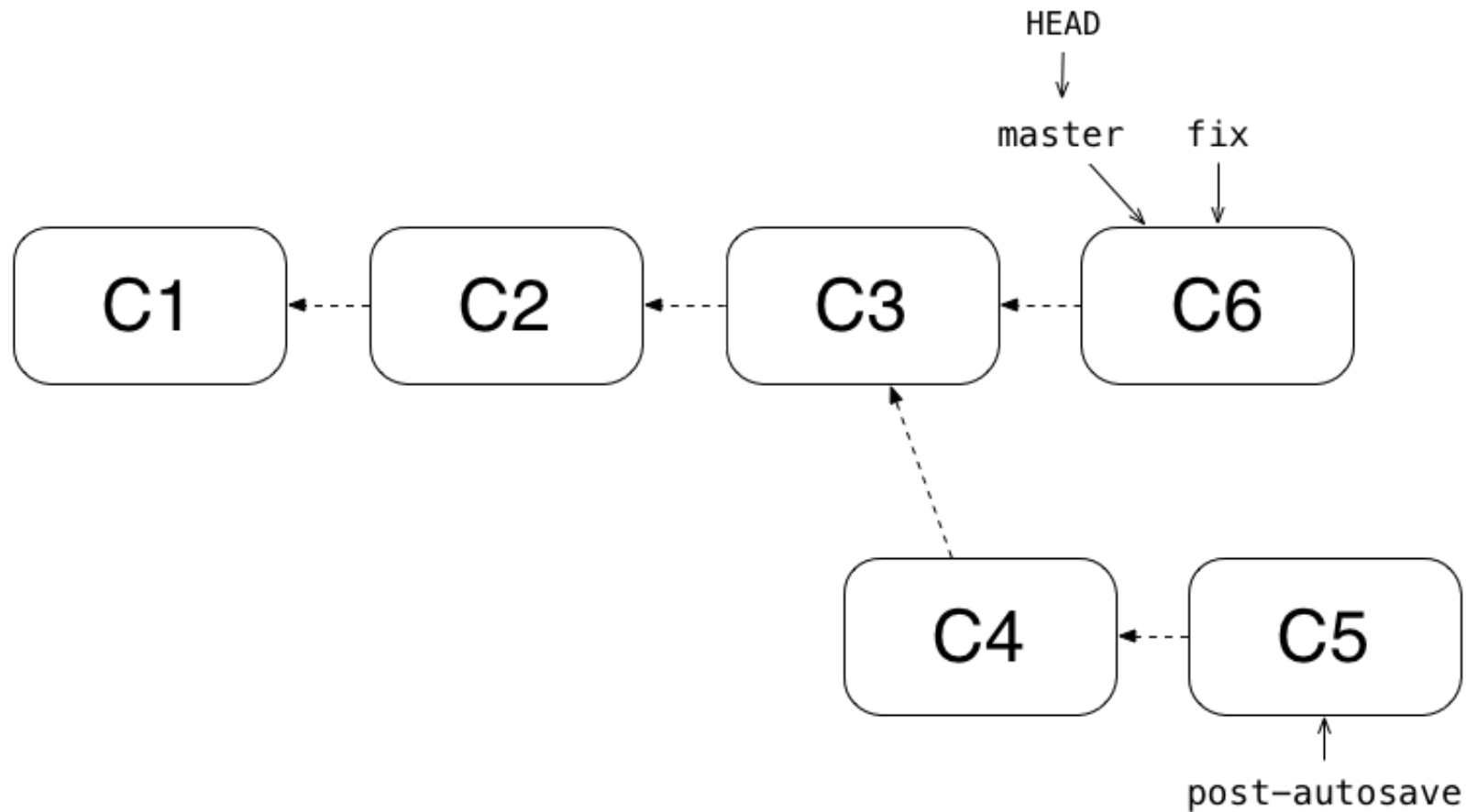
Ora branchamo per creare, ad esempio, una nuova  
feature: `post-autosave`



Avanziamo in `post-autosave` facendo due nuovi commit



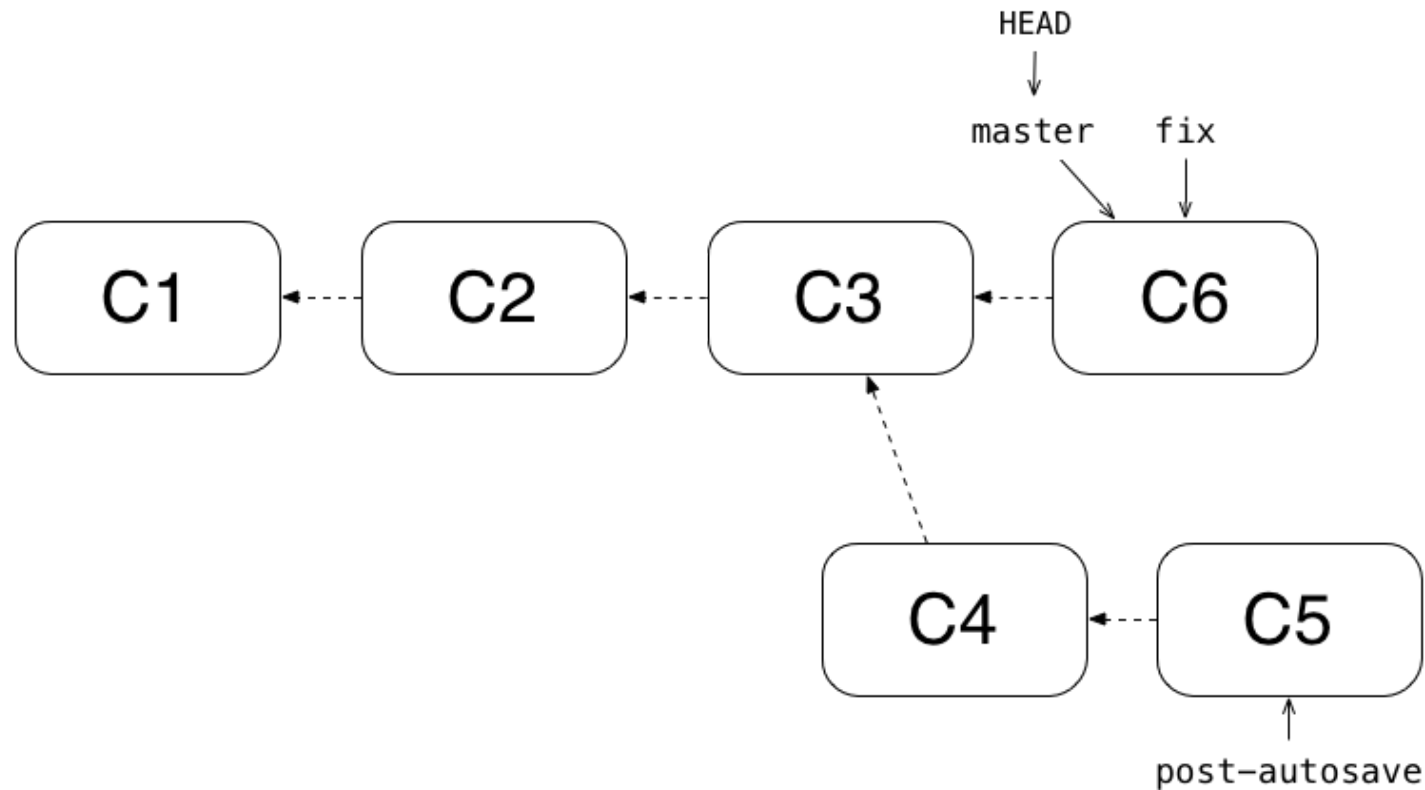
*Bug report* per un bug urgente: torniamo su **master**,  
branchiamo e committiamo una volta nel branch **fix**



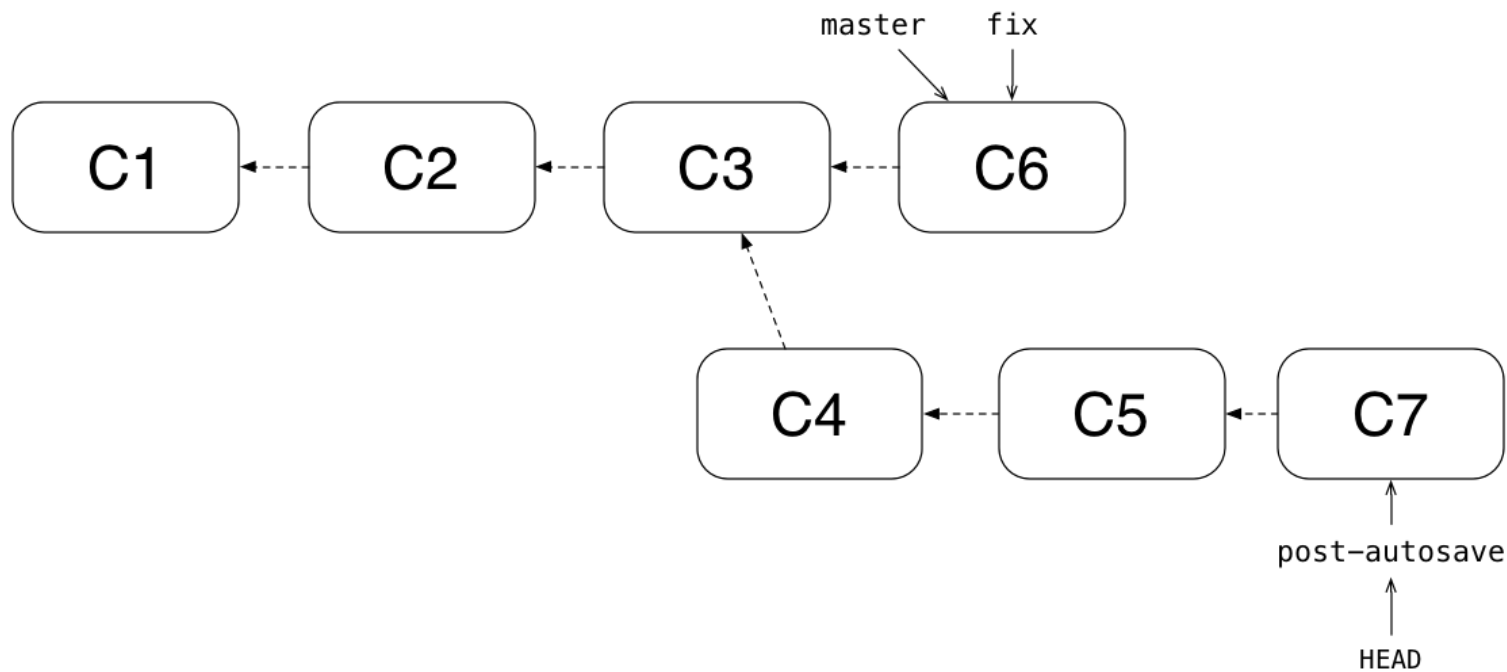


# Merge di `fix` in `master`: fast-forward

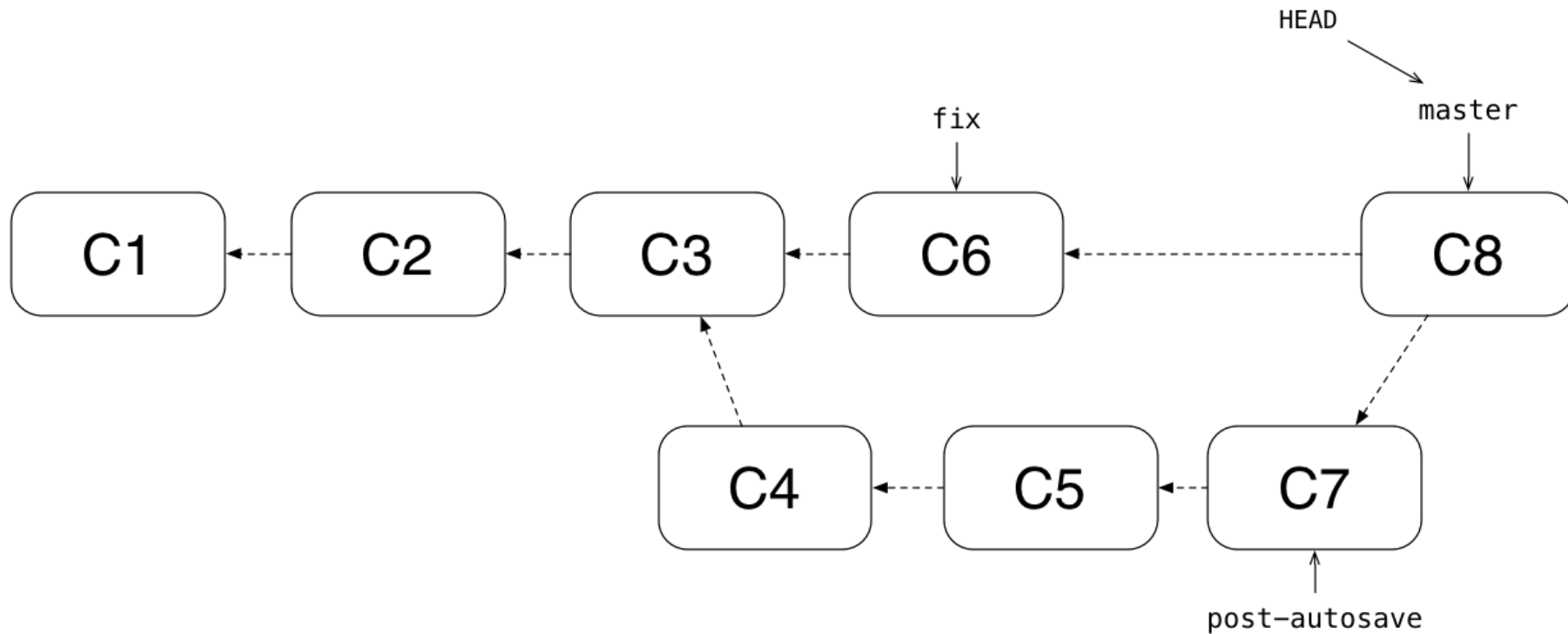
Abbiamo finito in `fix`, vogliamo rendere disponibile il `fix` anche in `master`:



Ora torniamo su `post-autosave` e facciamo un altro commit per finire la feature



Torniamo su **master** e mergiamo con **post-autosave**: git crea un nuovo **merge commit**.



# Remotes

Di solito il remote principale si chiama `origin`.

I branch remoti vengono prefissati da `origin/`:  
`origin/master`, `origin/fix`.

Si possono avere più di un remote (es. `production`,  
`master`, `experimental`).

# push/pull

- **push**: upload dei cambiamenti in locale su un remote
- **pull**: download dei cambiamenti su un remote in locale



È il più famoso hosting service che usa git come VCS.

<https://github.com/>



Simile a GitHub ma permette un numero limitato di  
repo *private*.



Hands down

# Installazione:

## Linux

```
apt-get install git-core # *buntu, debian  
yum install git-core     # fedora
```

## OS X

<http://git-scm.com/download/mac>

...Windows?

<http://msysgit.github.io/>

# Advanced concepts

`git rebase`

`git reset [--hard|--soft]`

`git bisect`

Tags

Hooks

# Resources

*Pro Git*, Scott Chacon (disponibile anche online su <http://git-scm.com/book>)

<https://try.github.io> (super beginner level)

<http://gitready.com/>

<http://gitready.com/>

<http://www.giyf.com/>

?

Grazie!

