



# GIT

jedyny system kontroli wersji jakiego potrzebujesz

Radosław Bułat (radarek), <http://radarek.jogger.pl>

email: ["6d6f632e6c69616d674074616c75622e6b65646172"].pack("H\*").reverse

# Co to jest Git?

- Git to rozproszony (zdecentralizowany) system kontroli wersji (DSCV) zaprojektowany pod duże projekty (linux kernel) z myślą o wydajności i efektywności.



# Dlaczego Git?

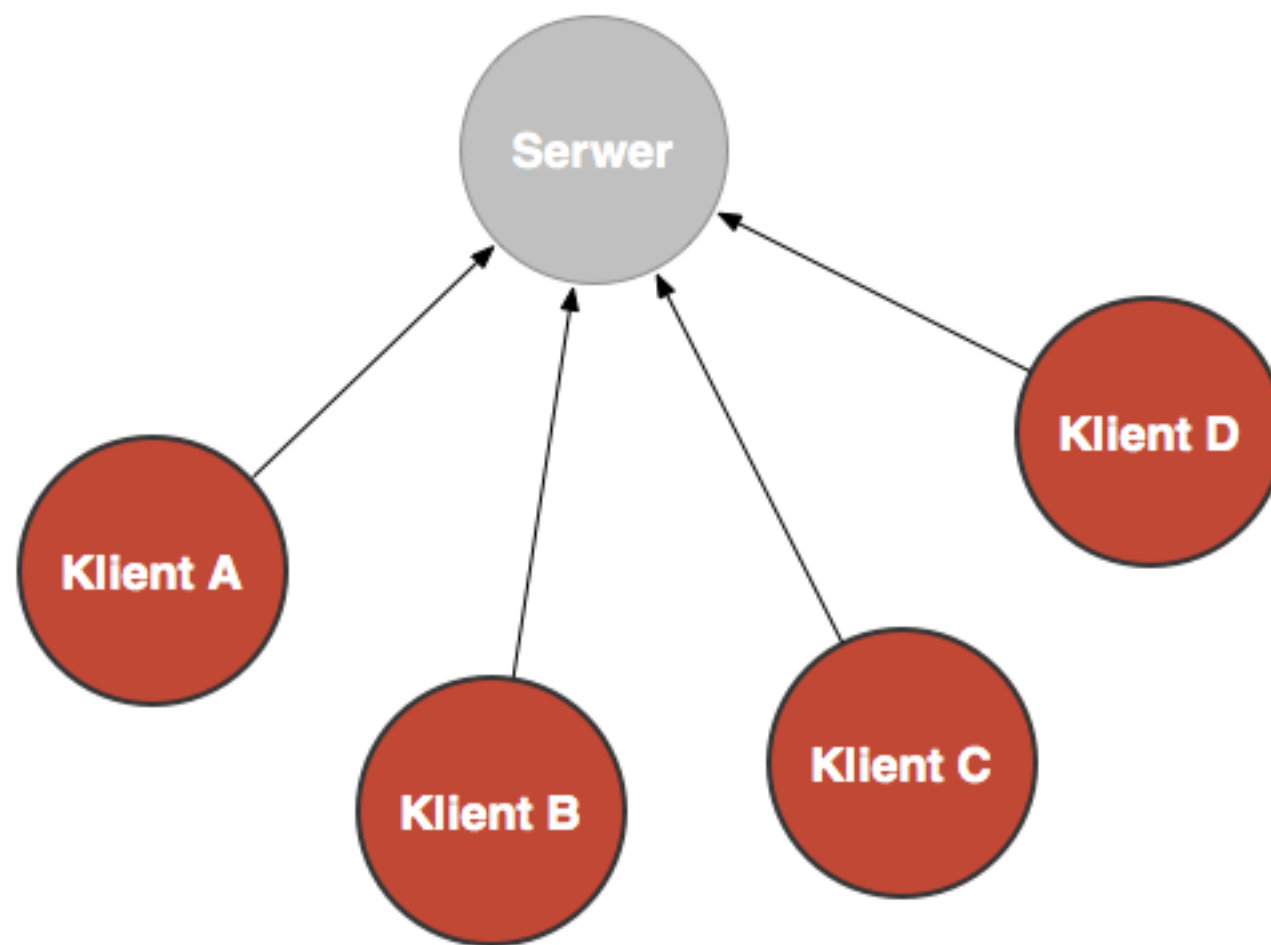
- zdecentralizowany
- nieinwazyjny
- niesamowicie szybki i wydajny
- łatwe, szybkie i lekkie branche
- index (staging area)
- pełna kontrola nad kodem
- bezpieczeństwo
- perełki



# scentralizowany vs zdecentralizowany



# scentryalizowany (SVN)

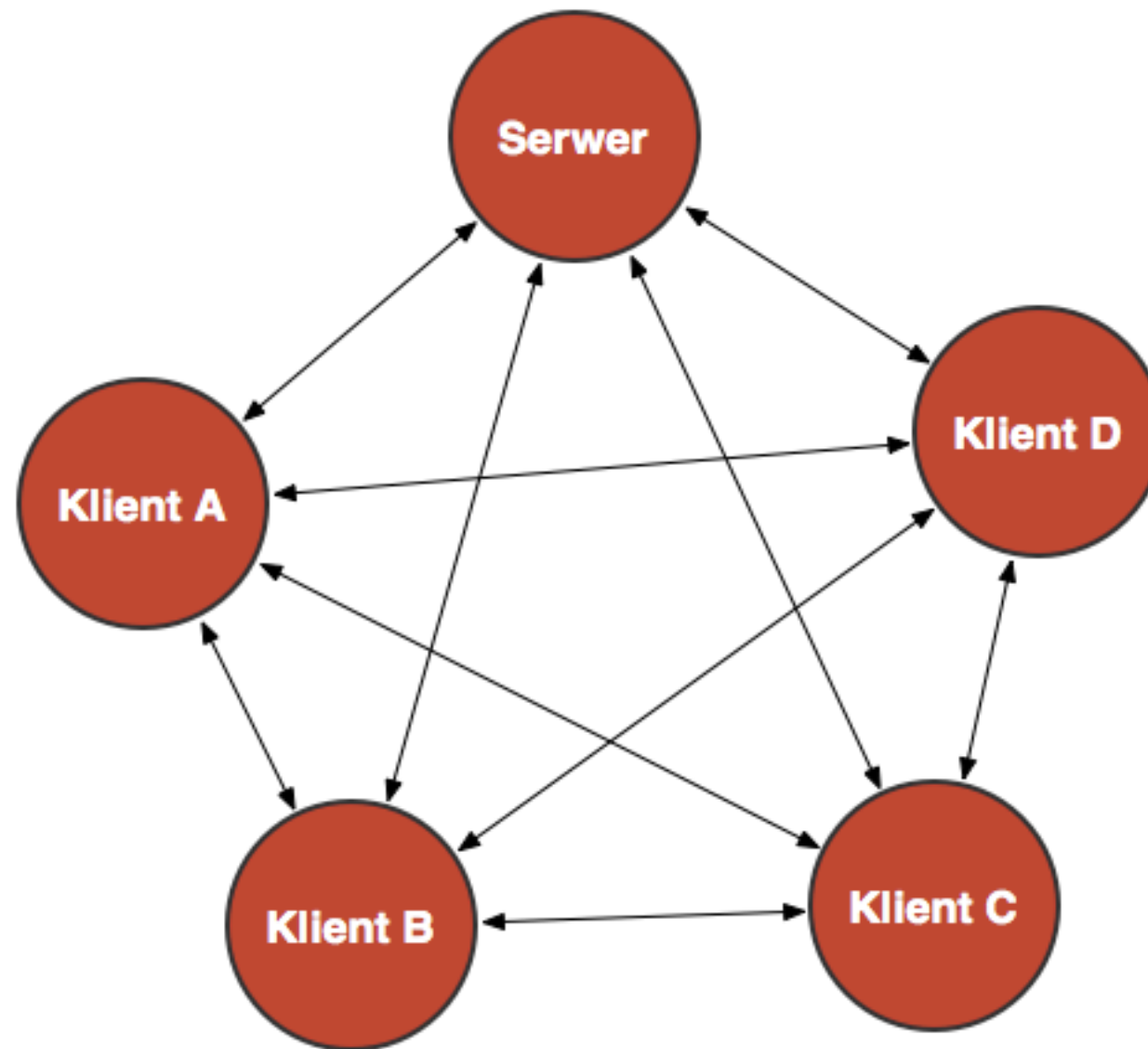


# scentralizowany (SVN)

- tylko serwer posiada repozytorium
- klonując repozytorium stajemy się właścicielami tylko jednej konkretnej rewizji (najczęściej HEAD)
- dostęp do poprzednich wersji tylko gdy jesteśmy online (i serwer przypadkiem nie padł...)
- sporo poleceń wymaga aktywnego połączenia z serwerem (kto korzysta często z svn log, svn checkout?)
- jesteśmy zmuszeni do pracy w jednym stylu (commit, update z głównego repozytorium)



# zdecentralizowany (Git)



# zdecentralizowany (Git)

- klonując repozytorium stajemy się właścicielami całego kodu (cała historia) - darmowy backup
- tylko 3 z powszechnie stosowanych poleceń potrzebują połączenia internetowego: git clone, git fetch\* i git push
- umożliwia stosowanie praktycznie dowolnego stylu





# nieinwazyjny



# nieinwazyjny

- `cd simple_repo`
- `git init`
- Initialized empty Git repository in `/Users/radarek/Desktop/prezentacja/examples/simple_repo/.git/`



# niesamowicie szybki i wydajny



# niesamowicie szybki i wydajny

- praktycznie wszystkie polecenia zwracają rezultat natychmiast
- brak lagów sieciowych (praca offline), np. git log, git checkout
- repozytorium zajmuje mało miejsca na dysku

```
$ git log --oneline | wc -l  
4917  
$ du -sh .git/ .  
31M .git/  
79M .
```





# łatwe, szybkie i lekkie branchy





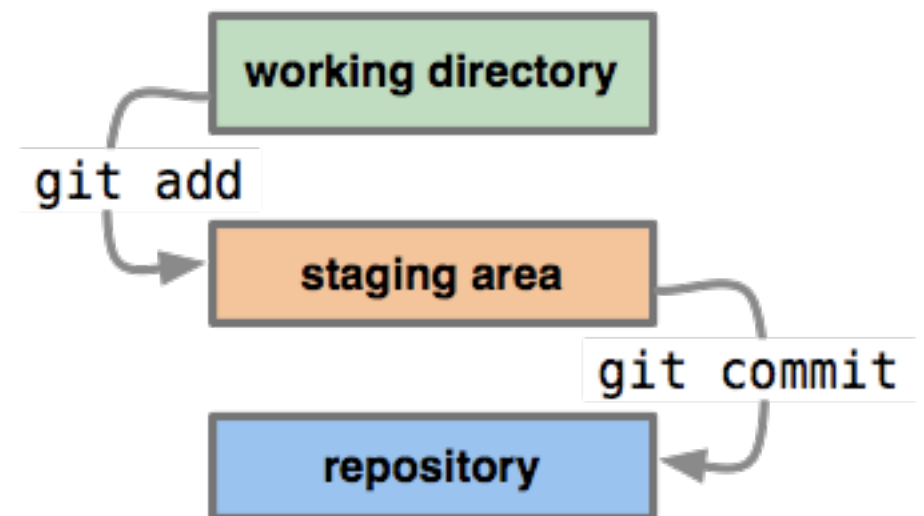
# łatwe, szybkie i lekkie branche

- utworzenie nowego brancha jest równoważne z utworzeniem pliku i zapisaniem do niego 40 bajtów
- domyślnie branche są lokalne
- eksperymenty z kodem, wielowątkowość, nowe ficzery

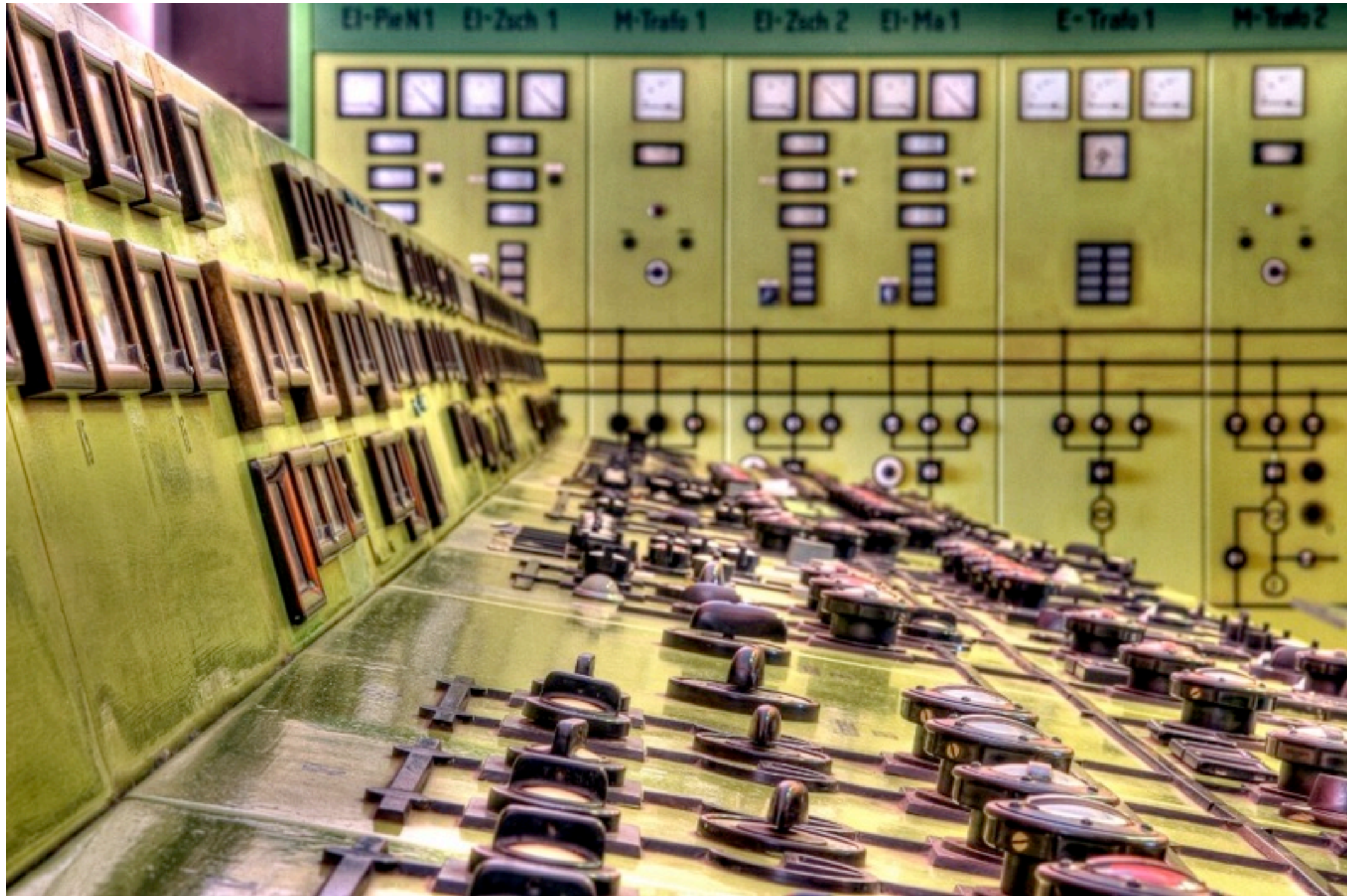


# index (staging area)

- pośredniczy pomiędzy tym co zmieniliśmy w katalogu roboczym a tym co faktycznie commitujemy
- daje pełną kontrolę nad tym co commitujemy
- `git add`, `git add -i`



# pełna kontrola nad kodem





# pełna kontrola nad kodem

- Jakie zmiany poczyniłem w katalogu roboczym?
- Który commit utworzył dany plik?
- Jakie zmiany poczynili inni?
- Chcę wycofać commit X.
- Jestem w trakcie pracy nad ficzerem X, ale dostałem polecenia naprawy błędu Y. Co mam zrobić?
- Jak podejrzeć jak wyglądał plik w commicie X?
- Jak zrobić commita z częściowymi zmianami z pliku?
- Popełniłem błąd w opisie commita - czy mogę go poprawić jeszcze?
- Jak znaleźć commity, których zmiany zawierają słowo X?
- Jak znaleźć commity, których komunikat zawiera słowo X?
- Jak wyświetlić zmiany (diff) z commita X?



# bezpieczeństwo



# bezpieczeństwo

```
$ git log --format=oneline -3
```

```
bf5d15456757e63598575db42917d702af9da729 Print proper "Usage:" messages for "rails plugin" command  
ff44cc284441be894cc6f2bbc1798a21e881414e whitespace  
de51cbccf8c9d4e59a128ca8dca8c42d8d7c4dc9 Fixed grouped_by_title spelling [#5063 state:committed]
```

- SHA1 commita jednoznacznie identyfikuje jego zawartość a także całą jego historię!
- praktycznie wszystkie operacje dają się odwrócić (usunięcie commita, zgubienie referencji itp.)





# peretki



# perelki

- git bisect
- git add -i
- git log -S, git log --grep
- git cherry-pick sha1
- git format-patch
- git stash
- git commit --amend
- git rebase -i
- git merge --squash
- git fetch && git log ..origin/master
- git show sha1
- git show sha1:path/to/file
- git reflog



# flickr credits

<http://www.flickr.com/photos/norbertloev/4411625814/sizes/o/>

<http://www.flickr.com/photos/-sel-/83785279/sizes/l/>

<http://www.flickr.com/photos/shapeshift/136184752/>

[http://www.flickr.com/photos/paco\\_calvino/4643224942/](http://www.flickr.com/photos/paco_calvino/4643224942/)

<http://www.flickr.com/photos/webwizzard/3931165508/sizes/l/>

<http://www.flickr.com/photos/muehlinghaus/241755891/sizes/o/>

<http://www.flickr.com/photos/tinkernoonoo/195686360/sizes/l/>

