

CodeWeek. 

**Solvw'**  | **Let's  
Solve  
It**

*WPROWADZENIE DO GIT'a*

Łukasz Rybka  
Gdańsk 2014-10-20

# O MNIE



- Absolwent wydziału Fizyki Technicznej i Matematyki Stosowanej Politechniki Gdańskiej
- Development Team Lead w firmie Solwit S. A.
- Wykładowca na Uniwersytecie Gdańskim / Politechnice Gdańskiej / Polsko-Japońskiej Akademii Technik Komputerowych
- Trener infoShare Academy

# PODSTAWOWE USTALENIA

- Nie jestem alfą ani omegą

# PODSTAWOWE USTALENIA

- Nie jestem alfą ani omegą
- Podstawowa znajomość GIT'a – “sufficient to be efficient”

# PODSTAWOWE USTALENIA

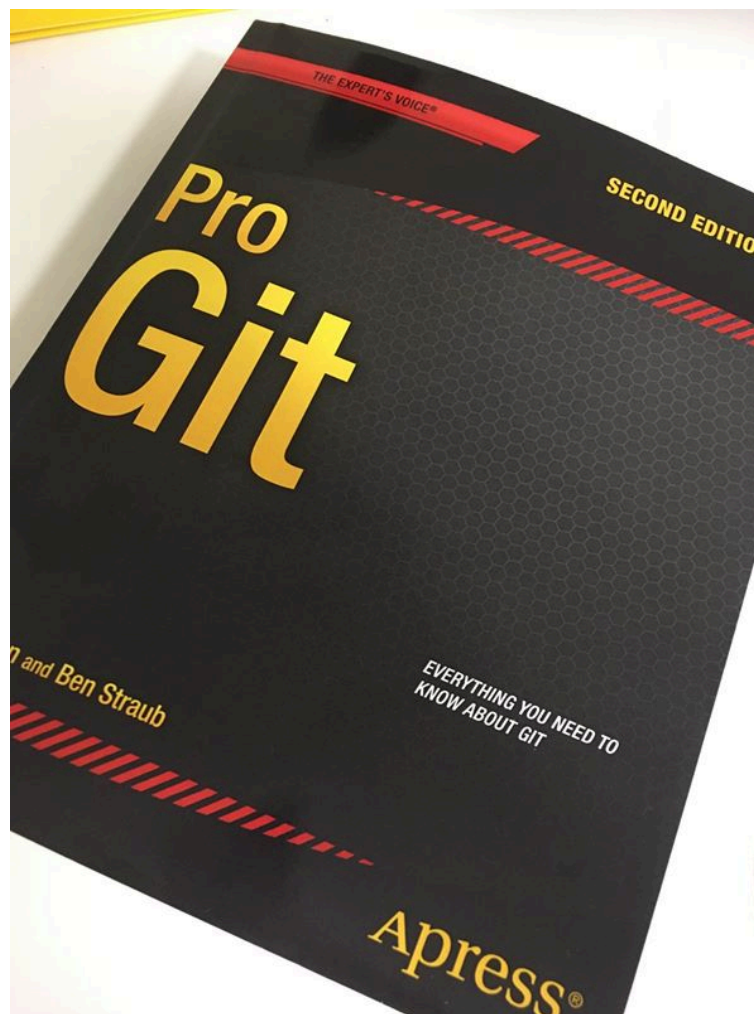
- Nie jestem alfą ani omegą
- Podstawowa znajomość GIT'a – “sufficient to be efficient”
- Pytania mile widziane, szczególnie w trakcie

# PODSTAWOWE USTALENIA

- Nie jestem alfą ani omegą
- Podstawowa znajomość GIT'a – “sufficient to be efficient”
- Pytania mile widziane, szczególnie w trakcie
- Slajdy to tylko notatki

<http://www.dragonia.org.pl/#/presentations>

# AKTYWNOŚĆ



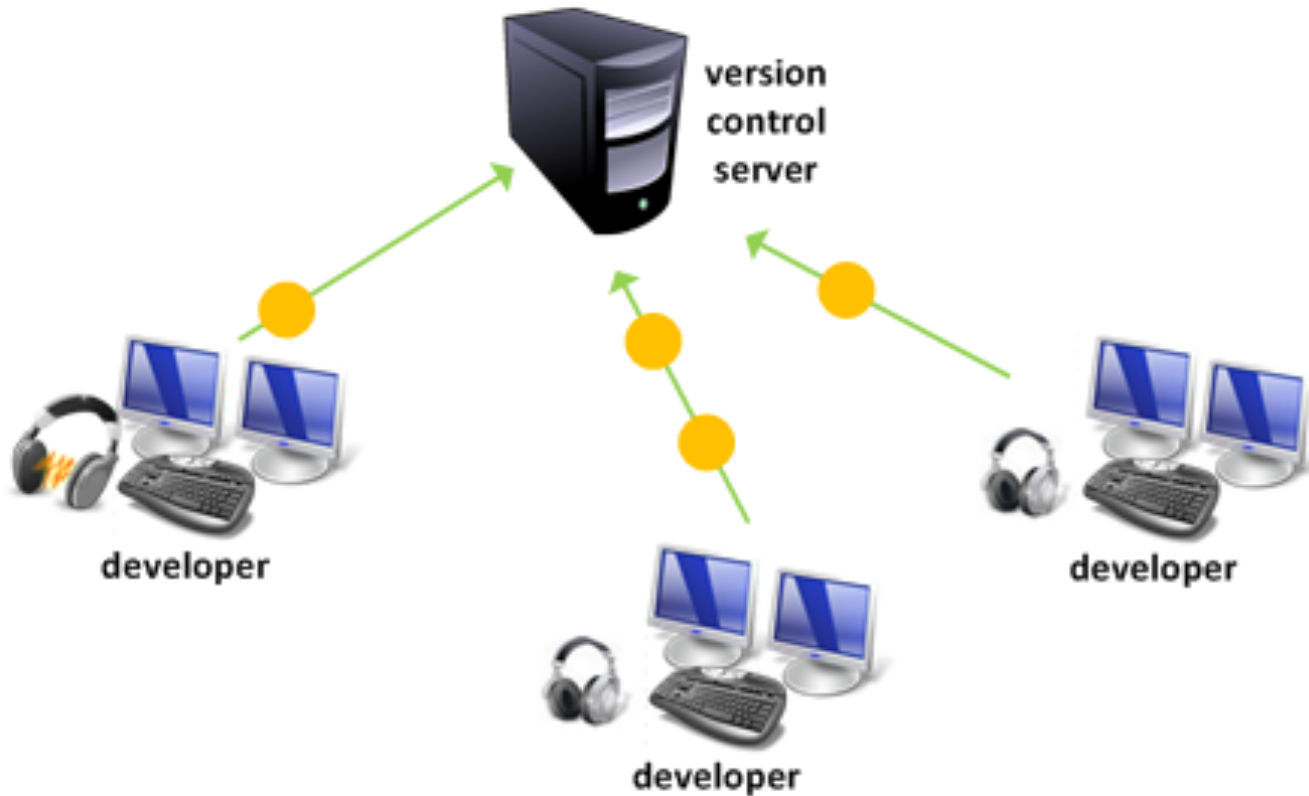
# RODZAJE SYSTEMÓW KONTROLI WERSJI



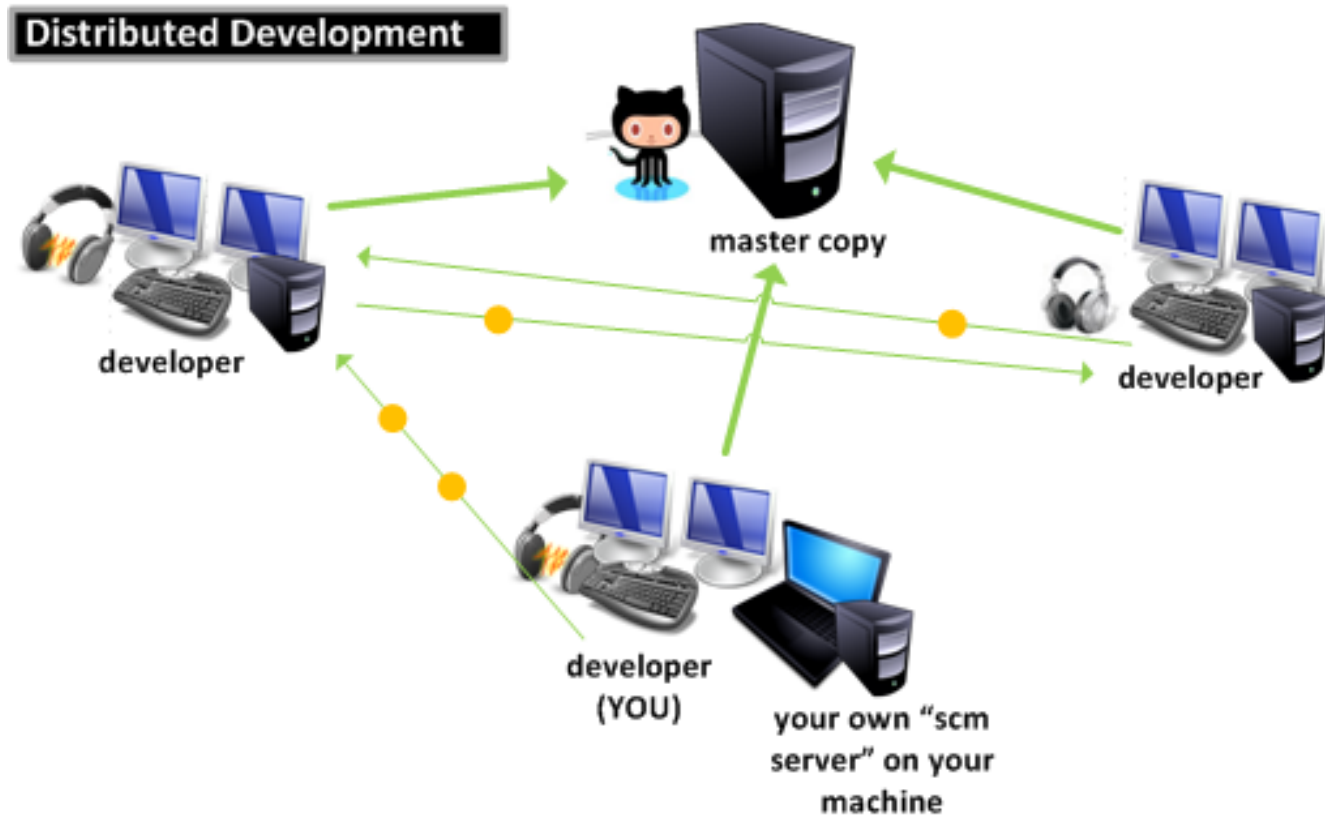


# RODZAJE SYSTEMÓW KONTROLI WERSJI

## Centralized Development



# RODZAJE SYSTEMÓW KONTROLI WERSJI

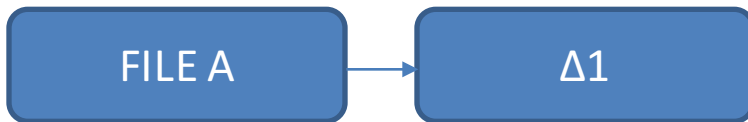
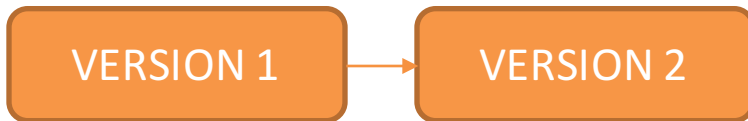


# CZYM JEST COMMIT?

VERSION 1

FILE A

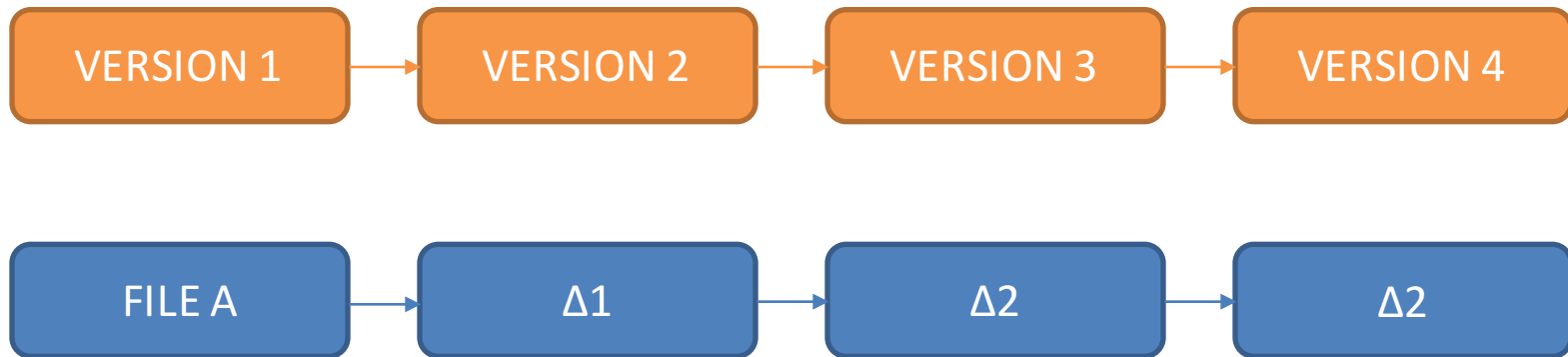
# CZYM JEST COMMIT?



# CZYM JEST COMMIT?



# CZYM JEST COMMIT?



# CZYM JEST COMMIT? (MODEL SVN-LIKE)

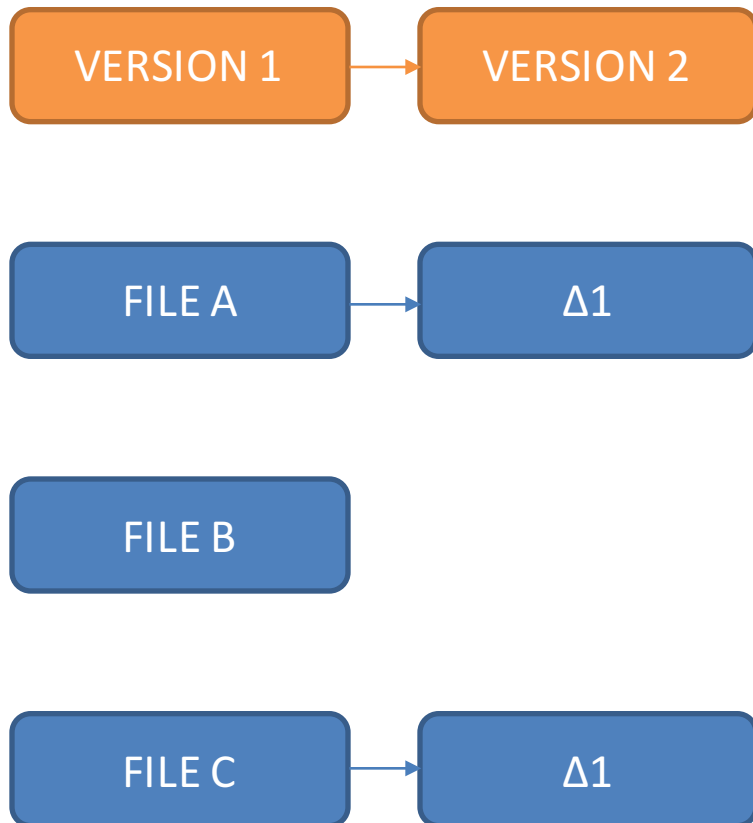
VERSION 1

FILE A

FILE B

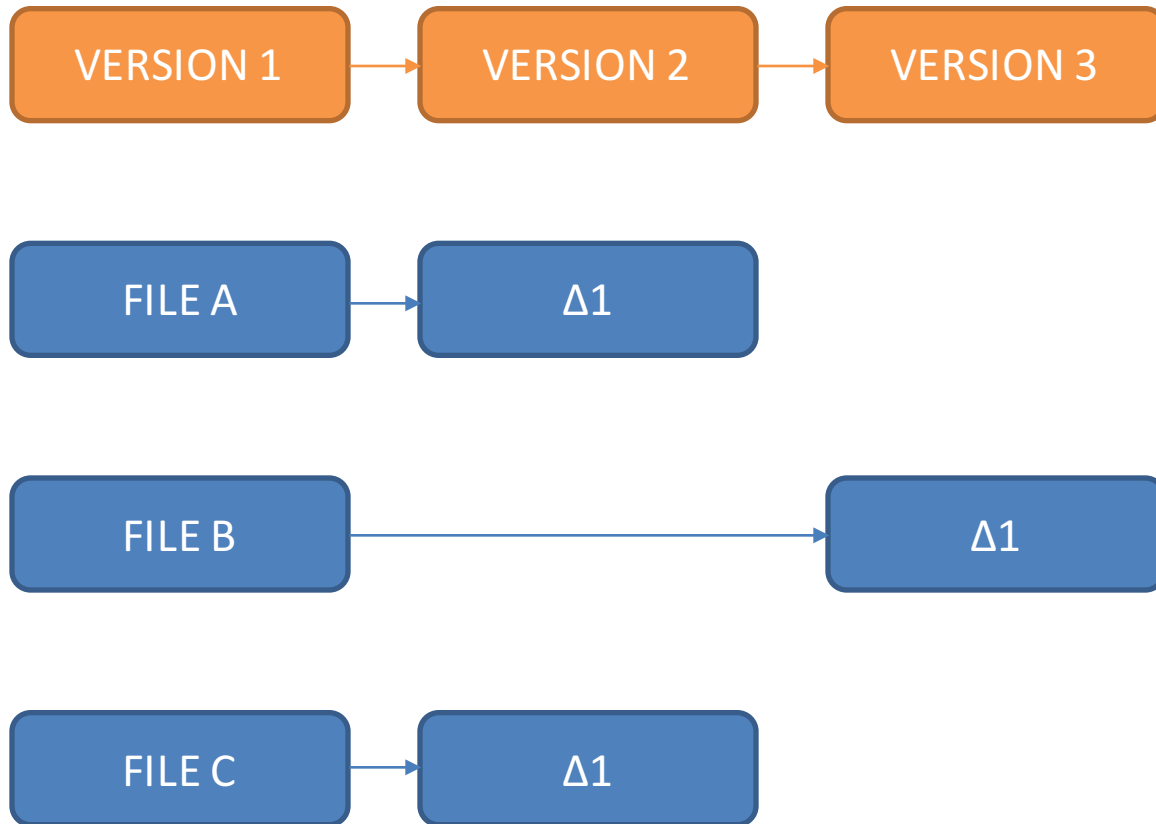
FILE C

# CZYM JEST COMMIT? (MODEL SVN-LIKE)

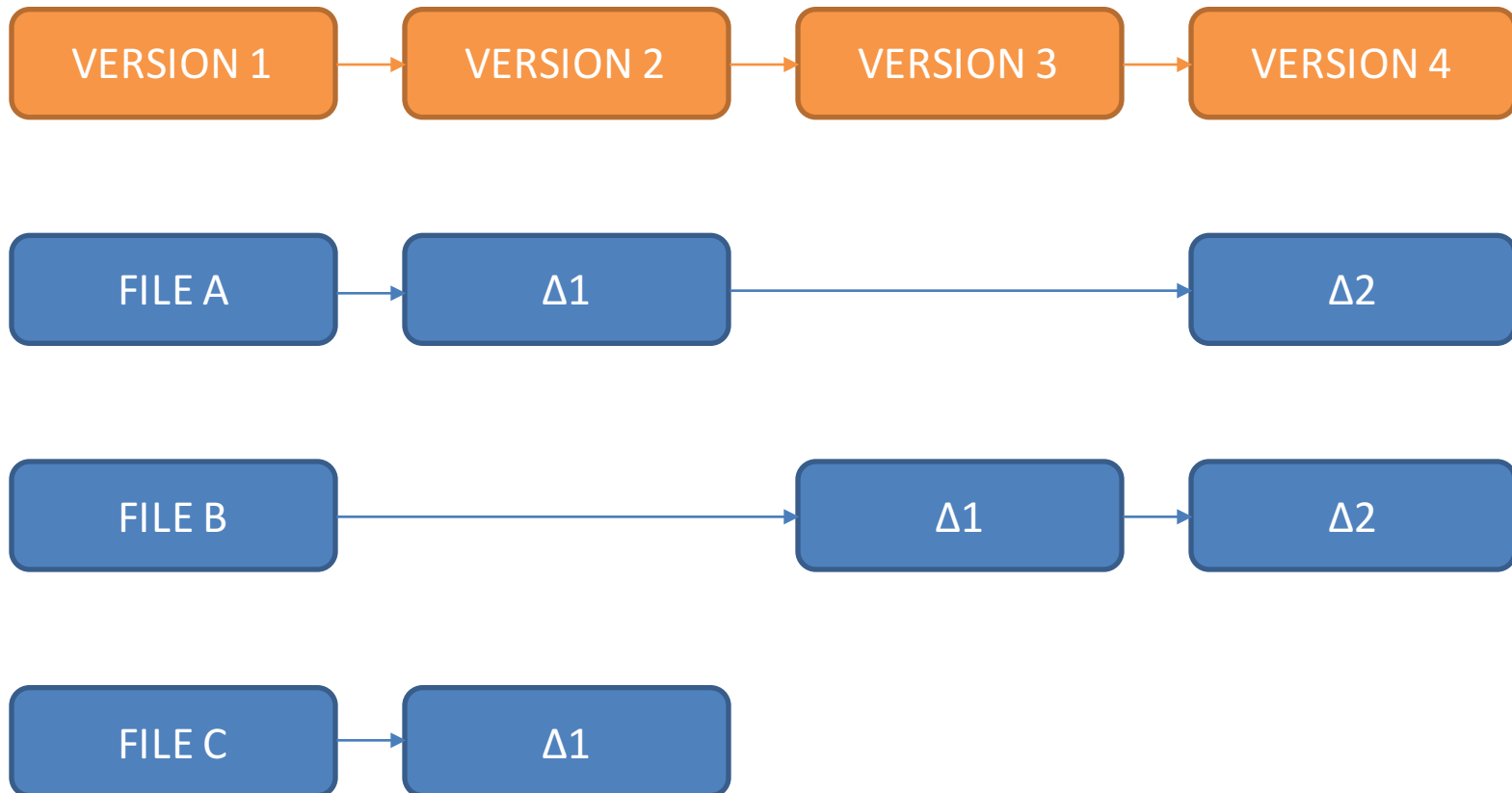




# CZYM JEST COMMIT? (MODEL SVN-LIKE)



# CZYM JEST COMMIT? (MODEL SVN-LIKE)



# CZYM JEST COMMIT? (MODEL GIT-LIKE)

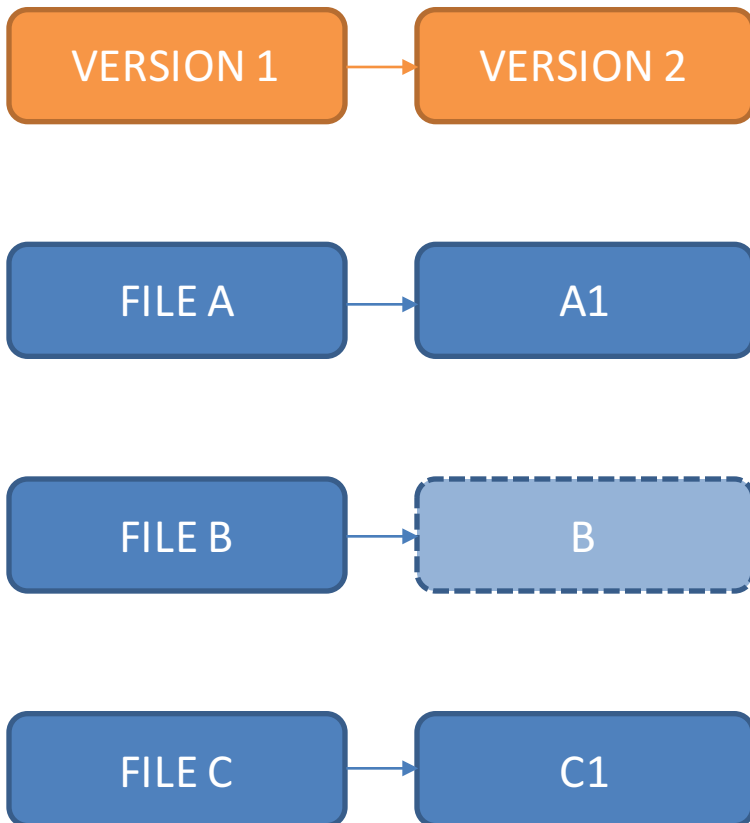
VERSION 1

FILE A

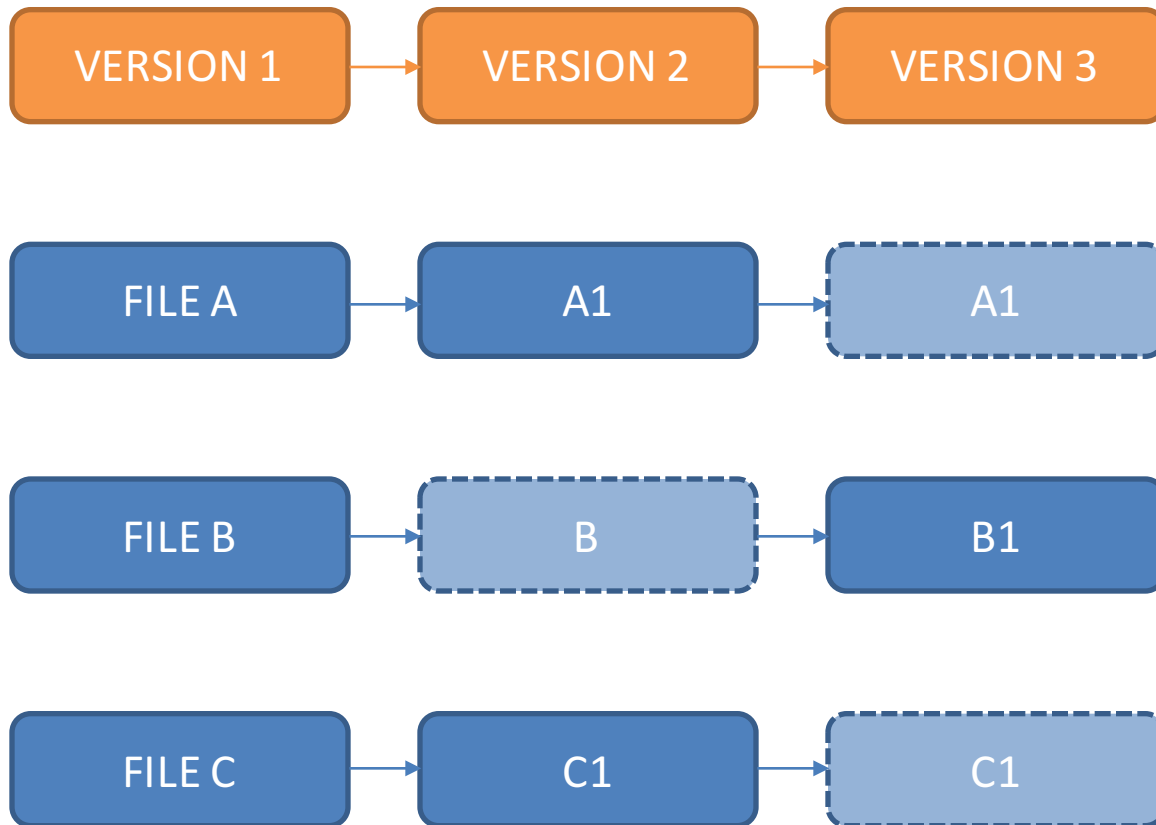
FILE B

FILE C

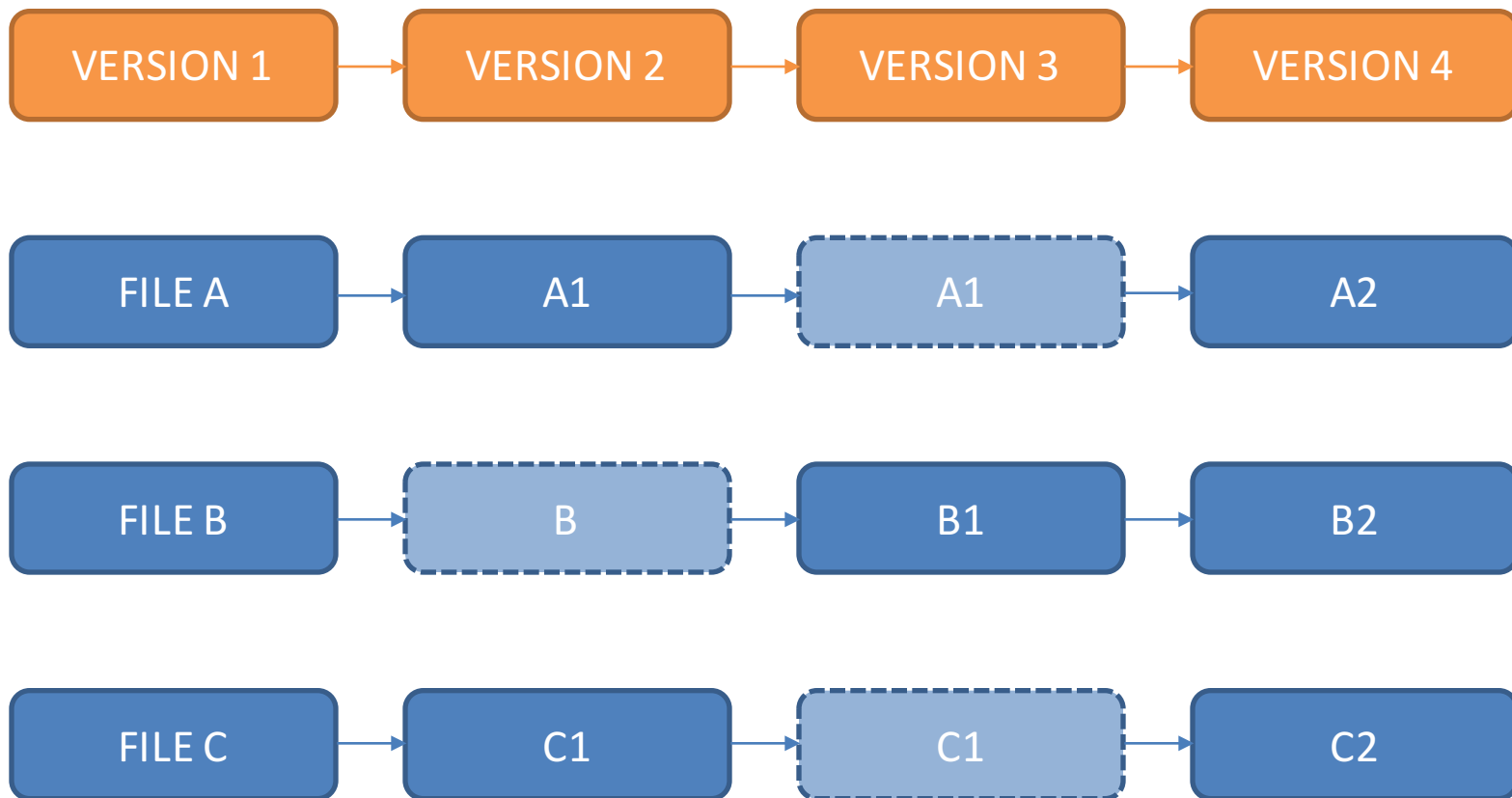
# CZYM JEST COMMIT? (MODEL GIT-LIKE)



# CZYM JEST COMMIT? (MODEL GIT-LIKE)



# CZYM JEST COMMIT? (MODEL GIT-LIKE)



# LOKALNE REPOZYTORIUM

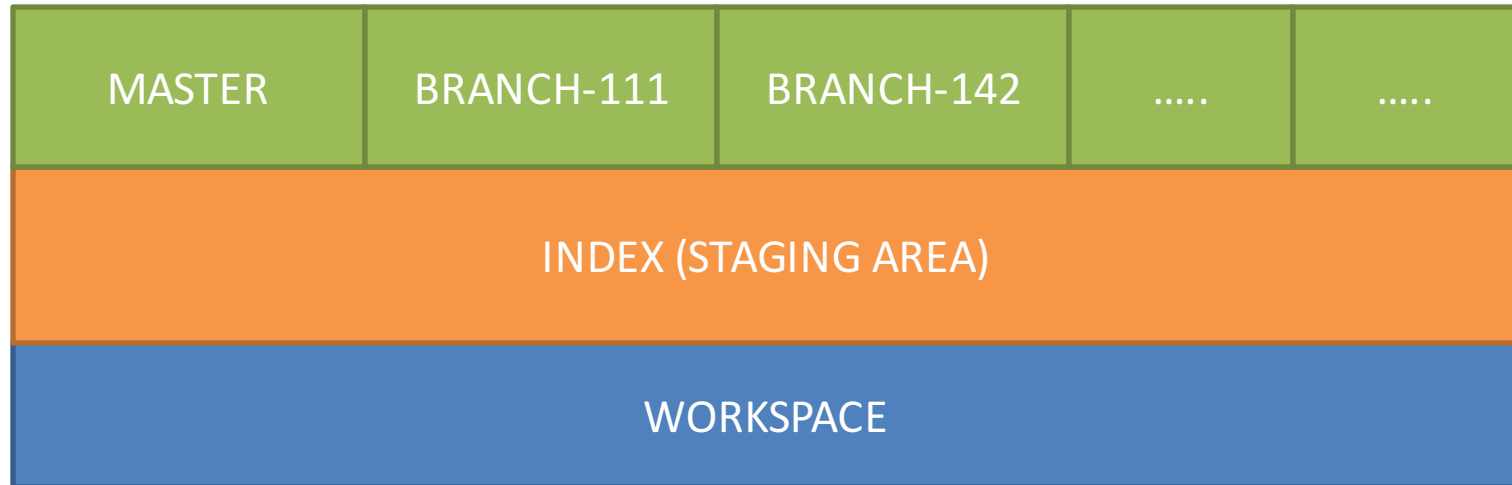
WORKSPACE

# LOKALNE REPOZYTORIUM

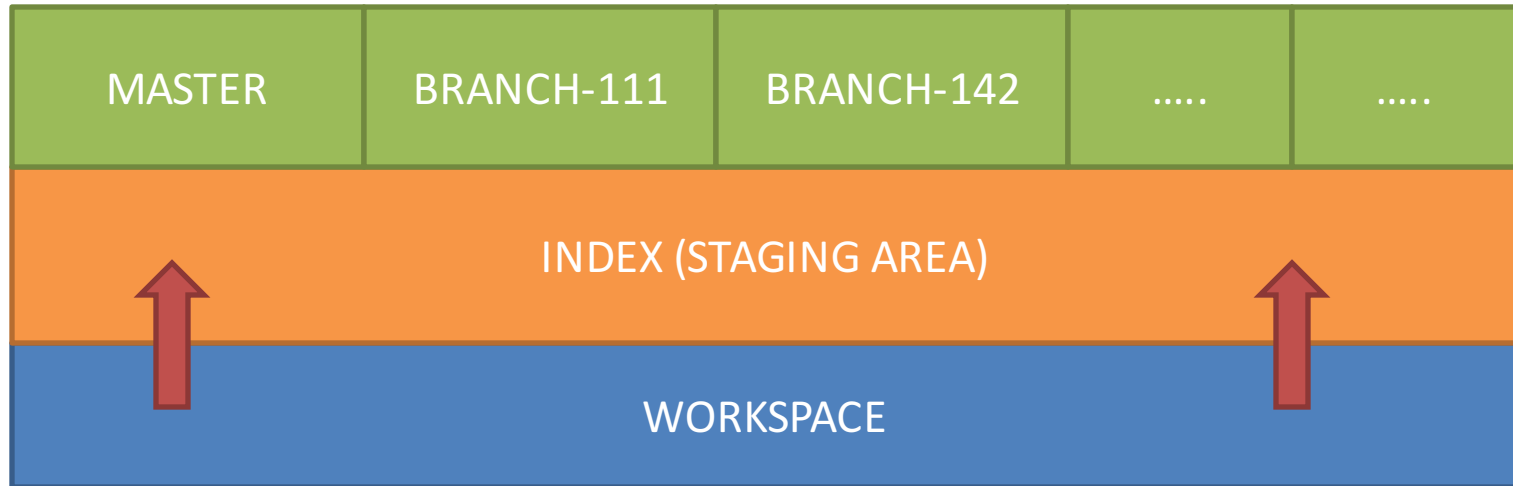




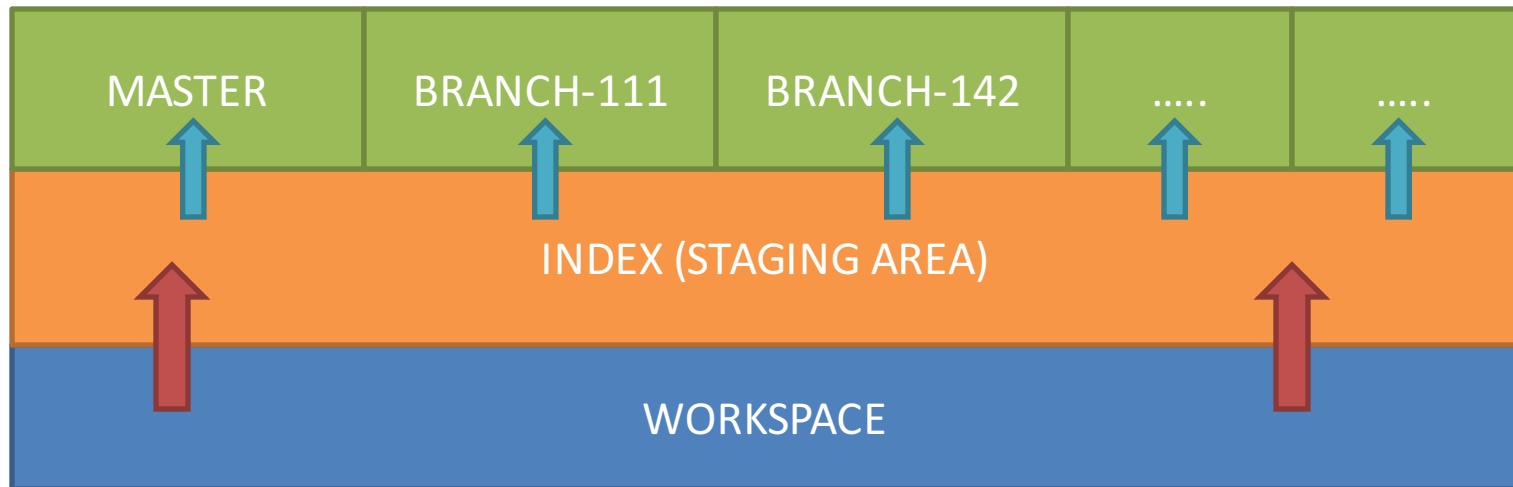
# LOKALNE REPOZYTORIUM



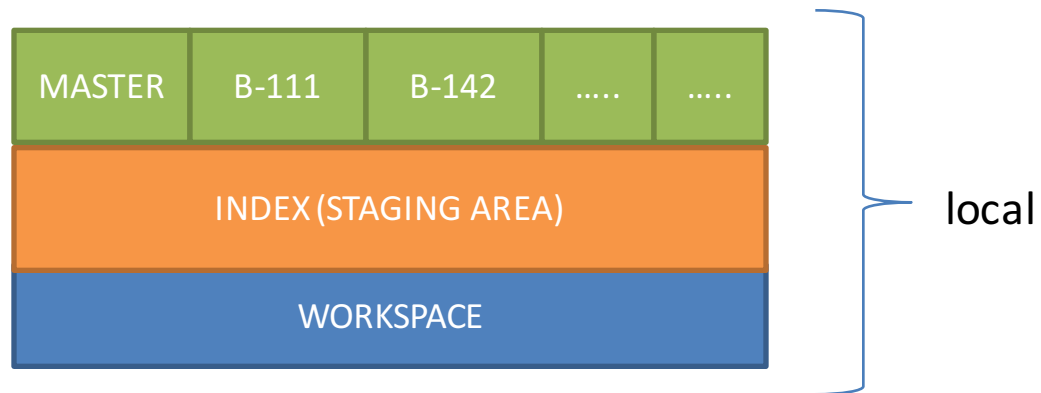
# LOKALNE REPOZYTORIUM



# LOKALNE REPOZYTORIUM

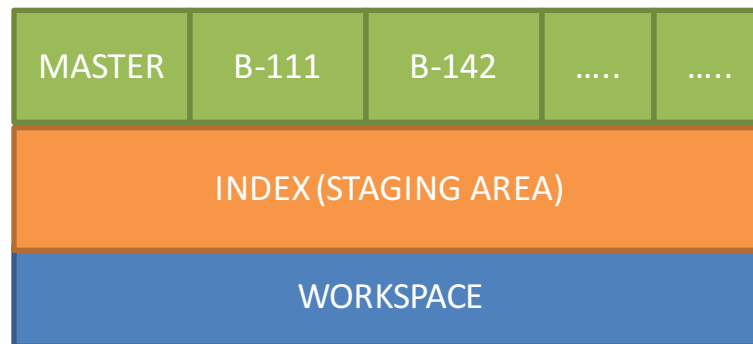
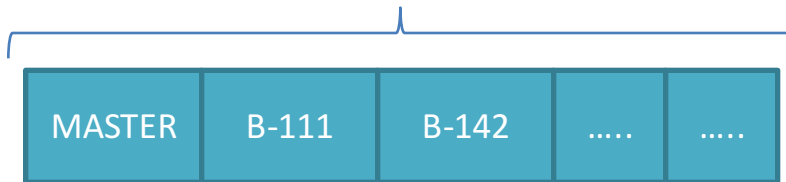


# LOCAL REPOSITORY



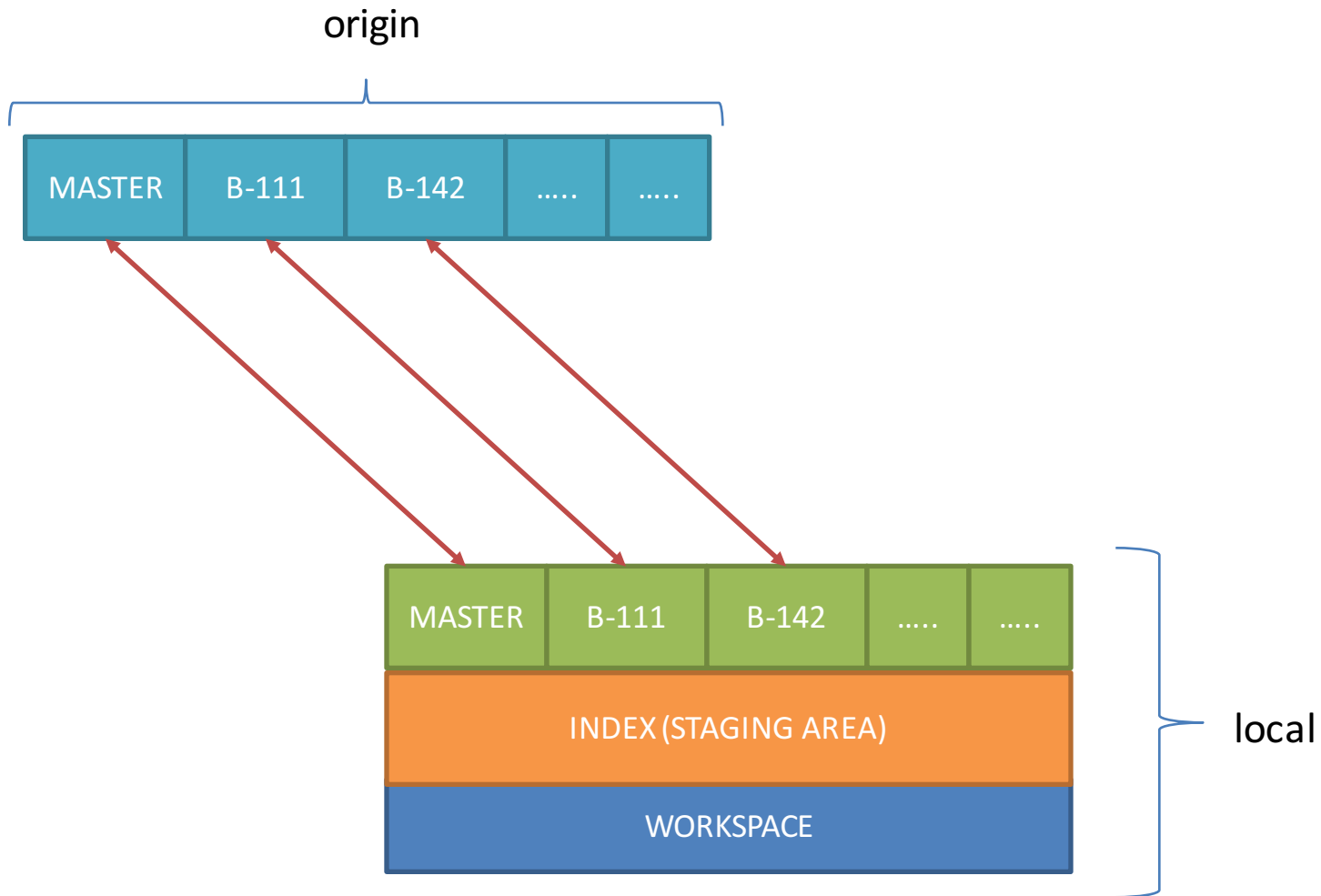
# ORIGIN

origin

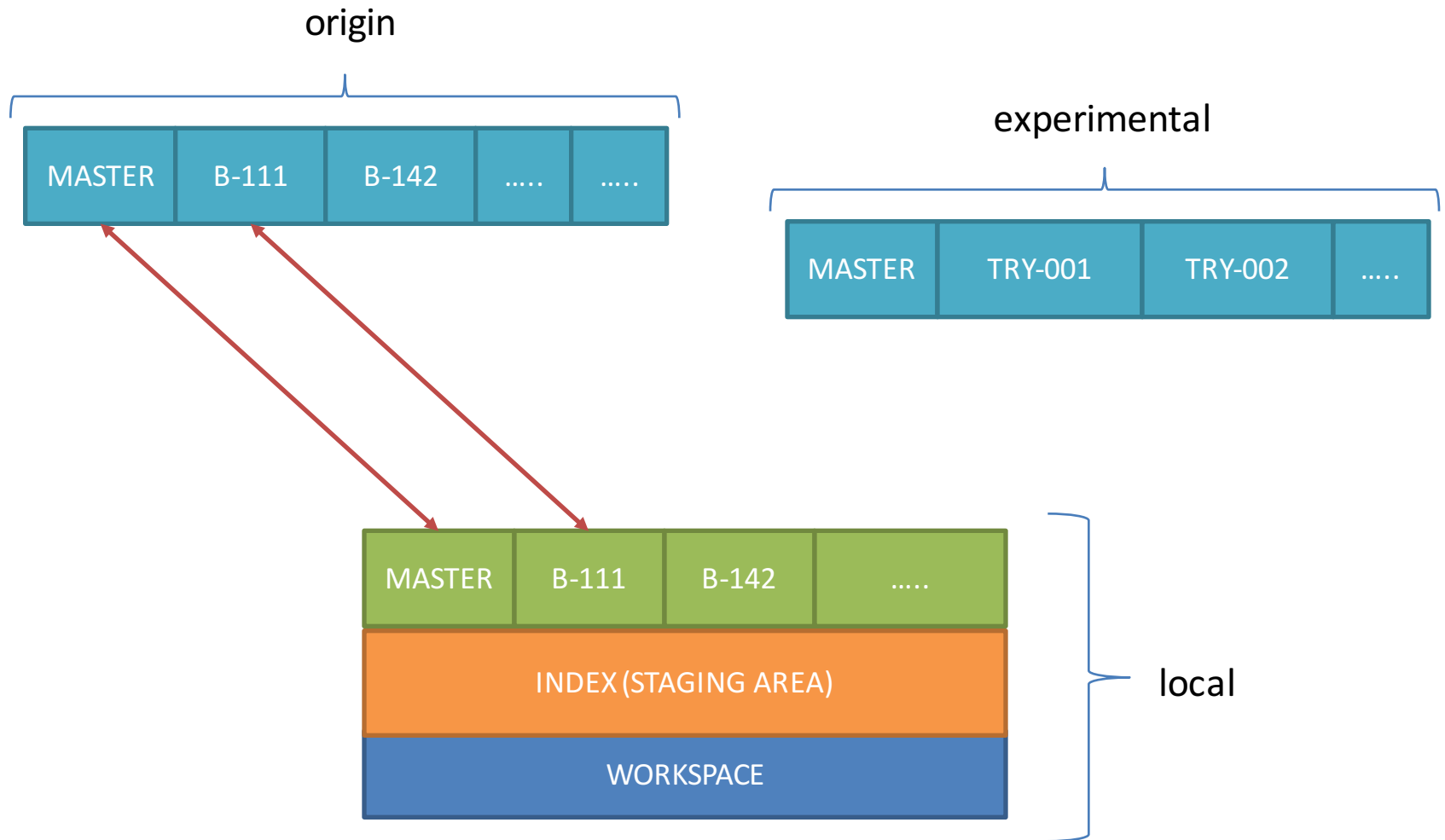


local

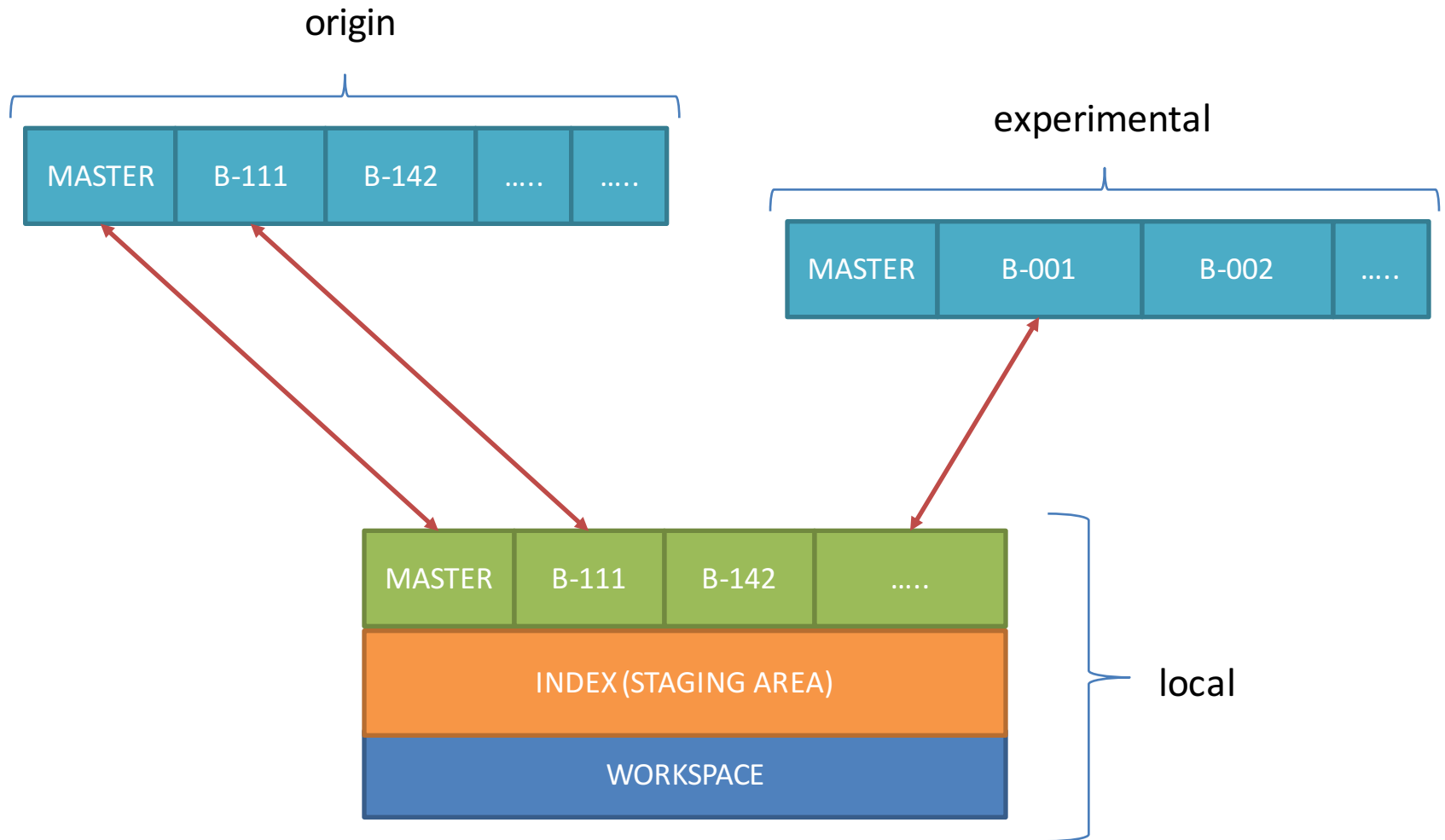
# ORIGIN



# MULTIPLE REMOTES



# MULTIPLE REMOTES





# PODSTAWOWE KOMENDY

- git init
- git status
- git branch
- git log

# PODSTAWOWE KOMENDY

- git init
- git status
- git branch
- git log
  
- git merge
- git pull (fetch)
- git push

# PODSTAWOWE KOMENDY

- git init
- git status
- git branch
- git log
  
- git merge
- git pull (fetch)
- git push
  
- ...
- ???

# PODSTAWOWE KOMENDY

- **git init** - inicjalizacja lokalnego repozytorium
- **git status** - informacje o stanie lokalnego repozytorium
- **git branch** - pokazanie informacji o lokalnych branch'ach
- **git log** - pokazywanie historii lokalnego repozytorium
  
- git merge
- git pull (fetch)
- git push
  
- ...
- ???

# KONFIGURACJA DANYCH COMMITERA

- `git config user.name`
- `git config user.name "Billy Everyteen"`
  
- `git config user.email`
- `git config user.email "your_email@example.com"`

# KONFIGURACJA DANYCH COMMITERA

- `git config user.name`
- `git config user.name "Billy Everyteen"`
  
- `git config user.email`
- `git config user.email "your_email@example.com"`
  
- `git config --global user.name "Billy Everyteen"`
- `git config --global user.email "your_email@example.com"`

# KONFIGURACJA LINE ENDINGS

- W zależności od systemu operacyjnego nowa linia skutkuje innym znakiem(kami) końca linii:
  - LF - Unix
  - CRLF - Windows
  - CR - Mac OS

# KONFIGURACJA LINE ENDINGS

- W zależności od systemu operacyjnego nowa linia skutkuje innym znakiem(kami) końca linii:
  - LF - Unix
  - CRLF - Windows
  - CR - Mac OS
- Objawia się różnymi komunikatami:
  - “LF will be replaced by CRLF”
  - “CRLF will be replaced by LF”



# KONFIGURACJA LINE ENDINGS

- W zależności od systemu operacyjnego nowa linia skutkuje innym znakiem(kami) końca linii:
  - LF - Unix
  - CRLF - Windows
  - CR - Mac OS
- Objawia się różnymi komunikatami:
  - “LF will be replaced by CRLF”
  - “CRLF will be replaced by LF”
- Trzy tryby obsługi znaków końca linii:
  - `core.autocrlf=true` (zamiana `crlf->lf` | `lf->crlf`)
  - `core.autocrlf=false` (zamiana `crlf->lf`)
  - `core.autocrlf=input` (brak konwersji)

# WYSYŁANIE NAJNOWSZYCH ZMIAN

**git commit**  
**git push origin master**

---

**git push -u origin master**

---

**git branch --set-upstream-to=<branchname>**  
**git branch -u <branchname>**

# POBIERANIE NAJNOWSZYCH ZMIAN

**git pull**

---

**git fetch origin master**  
**git merge origin/master**

# MERGE VS REBASE

**git merge**

---

**git merge --squash**

---

**git rebase (git pull --rebase)**

# PRACA ZE ZDALNYMI REPOZYTORIAMI

```
git remote add <name> <repo_url>
```

---

```
git remote show <name>
```

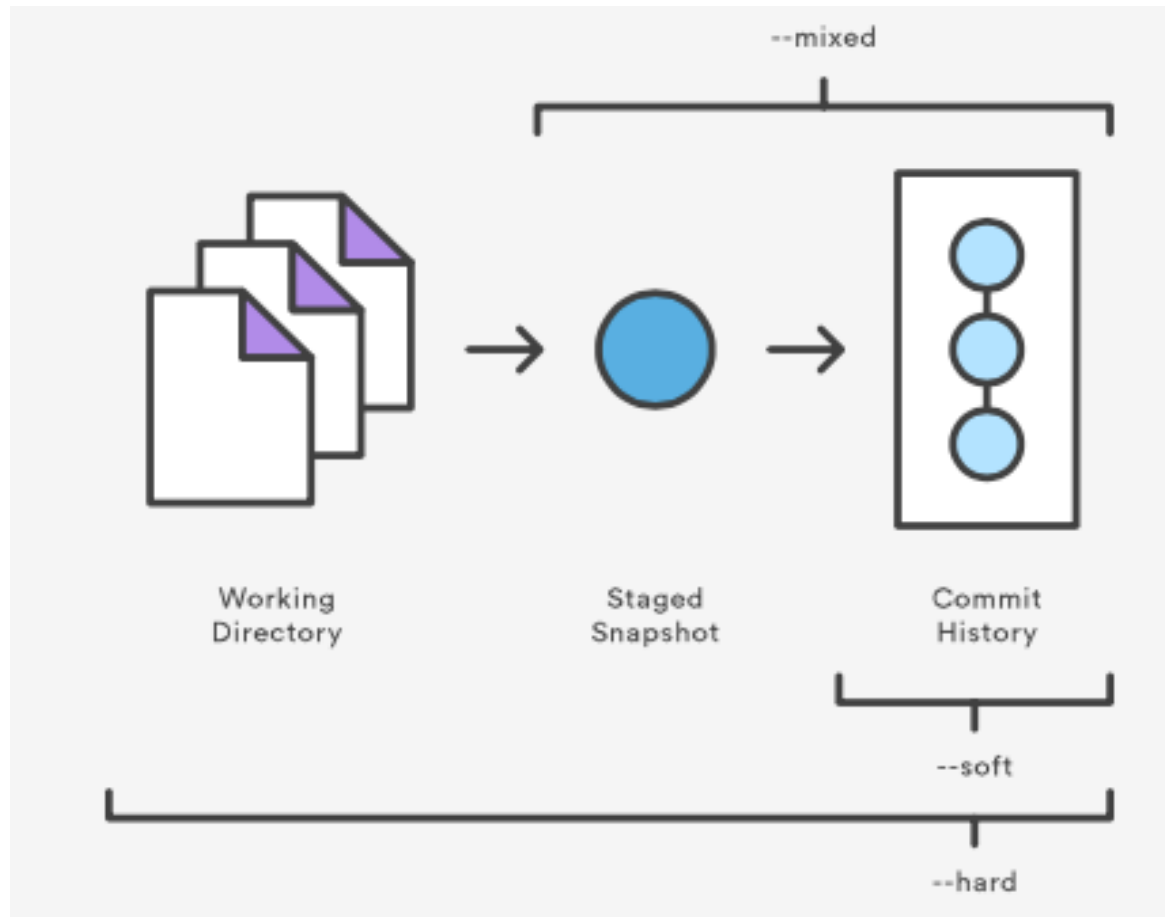
---

```
git remote rename <name> <new_name>
```

---

```
git remote rm <name>
```

# GIT RESET



# GIT RESET

```
git reset --soft HEAD~<n>  
git reset --soft <sha>
```

---

```
git reset --hard HEAD~<n>  
git reset --hard <sha>
```

---

```
git reset --mixed HEAD~<n>  
git reset --mixed <sha>
```

**git rm --cached <file>**

**(usuwa plik z repozytorium, ale pozostawia go na dysku)**



**git rm --cached <file>**

**(usuwa plik z repozytorium, ale pozostawia go na dysku)**

**git mv <file> <new\_file\_name>**

**(zmiana nazwy pliku)**

**git log -<n>**  
**(pokazuje jedynie ostatnich n-commitów)**

**git log -<n>**  
**(pokazuje jedynie ostatnich n-commitów)**

**git log -p**  
**(dodaje diff do każdego commitu)**

**git log -<n>**  
(pokazuje jedynie ostatnich n-commitów)

**git log -p**  
(dodaje diff do każdego commitu)

**git log --stat**  
(zwiększa ilość informacji w logu)

# ULEPSZONE LOGOWANIE

**git log -<n>**  
(pokazuje jedynie ostatnich n-commitów)

**git log -p**  
(dodaje diff do każdego commitu)

**git log --stat**  
(zwiększa ilość informacji w logu)

**git log --graph**

# ULEPSZONE FORMATOWANIE LOGOWANIA

**git log --oneline**  
(skrótowa informacja o każdym commicie, max 1 linia)

# ULEPSZONE FORMATOWANIE LOGOWANIA

**git log --oneline**

**(skrótowa informacja o każdym commicie, max 1 linia)**

**git log --pretty=oneline | short | full | fuller | format=""**

**(zmienia sposób formatowania każdego commitu)**

PYTANIA?





# Let's Solve IT