Corso ing.Sw:

Struttura repository e .gitignore

Riassunto episodi precedenti

Gian Enrico Conti

March 2024

Motivazione

- Riprodurre artefatti software in maniera standard…

  … con il minimo essenziale (pom.xml + sorgenti + asset + indicazioni)

- Efficienza: upload/downoad solo dei file “utili”
- Estetica: repo pulito e ordinato

Quale file cartelle ignorare?

Rule of thumb:

git-ignorare tutto ciò che viene generato

quindi che **non è codice o asset** dell'applicativo

Esempi:

- Binari o cartelle prodotti dalla compilazione
- Runtime file, come `log`, `lock`, `tmp`
- Configurazioni dell'IDE
- File nascosti come `.DS_Store`
- (Cache dipendenze)

Esempi:

Nel nostro caso:

- Cartelle **target**, **.idea**, (out)
- File **.class**
- File **.iml**

intelliJ help:

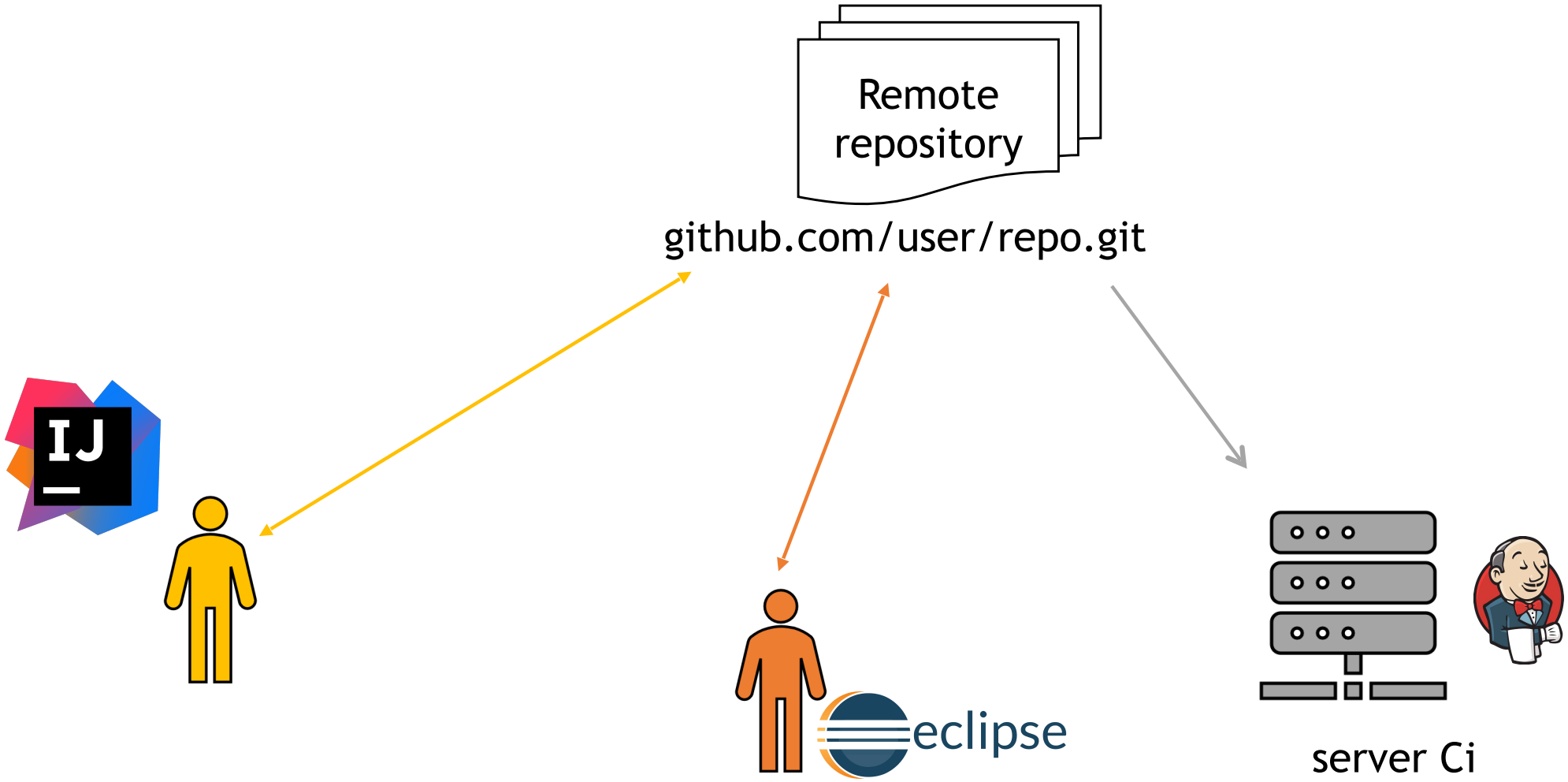
Directory based project format (.idea directory)

This format is **used by all the recent IDE versions** by default. Here is what you need to share:

- All the files under .idea directory in the project root except the workspace.xml, usage.statistics.xml, and tasks.xml files which store user specific settings
- All the .iml module files that can be located in different module directories (applies to IntelliJ IDEA)

- Noi NON usiamo questo formato… ma in altre realtà industriali e' necessario.

Perché .iml e .idea no?



.gitignore

- Aiuta a specificare i file **non tracciati** da ignorare
- Specifica cosa escludere
- Specifica cosa tra gli esclusi va incluso (prefisso "!")

- Doc:
- https://git-scm.com/docs/gitignore
- https://www.gitignore.io

Ignorare file già tracciati

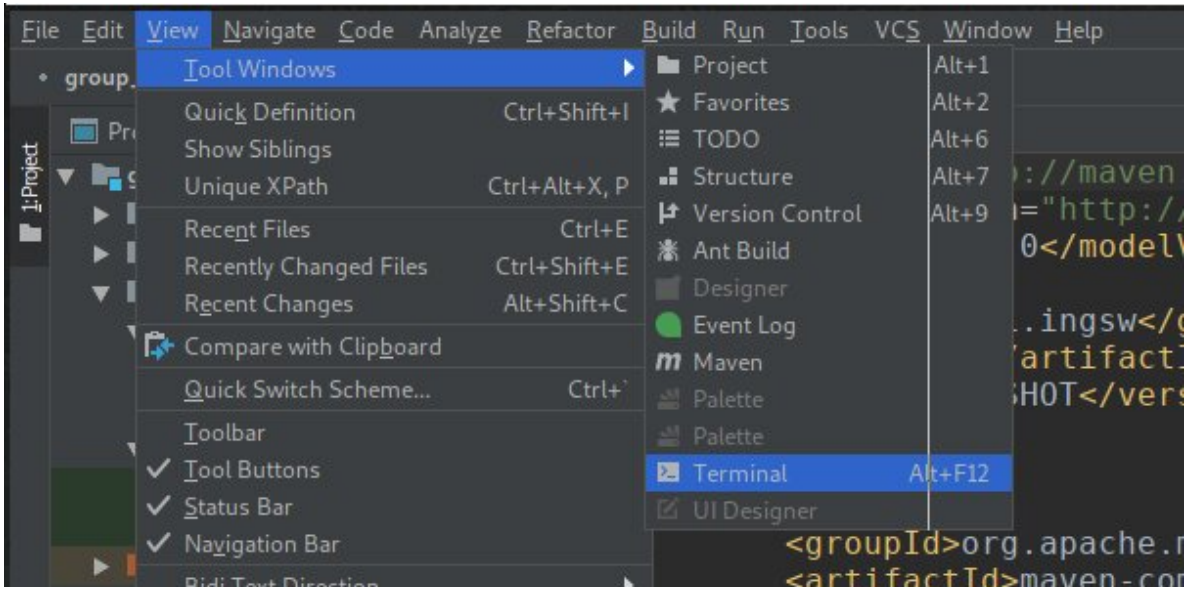
- Forse perché sono stati tracciati prima di essere ignorati…

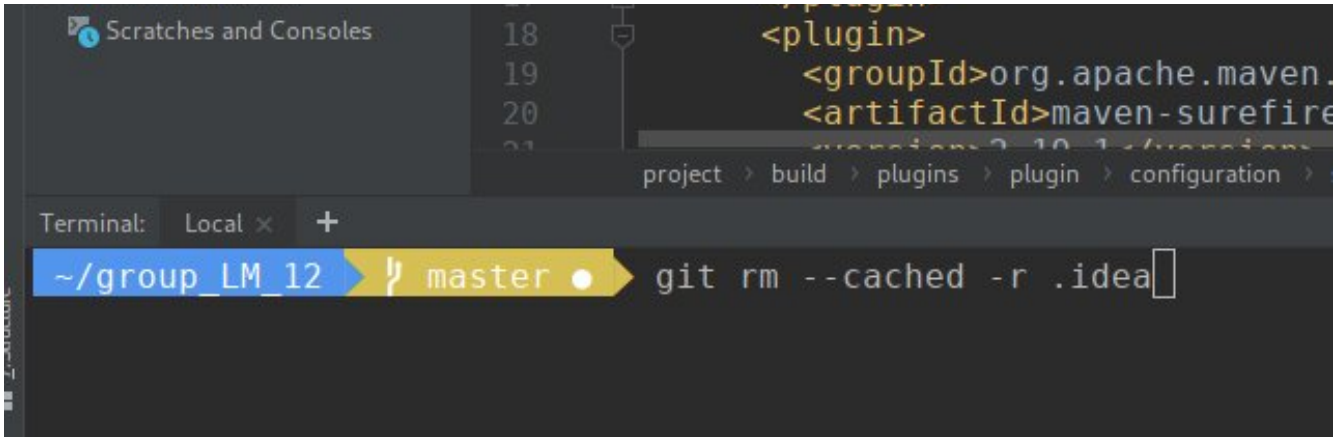
- Per esempio: `.idea/` e `project.iml` sono stati gia' committati, ma andrebbero esclusi:
  - Aggiungete dei pattern al .gitignore per ignorarli (se non c'e' gia')
  - Da linea di comando:

```
$ git rm --cached project.iml
$ git rm --cached -r .idea
$ git commit -m "Start ignoring .idea and
project.iml"
$ git push
```

Ignorare file già tracciati

- Da Intellij:





Nel nostro repository remoto:

-

```
Deliverables
src:
    main:
        java/it/polimi/ingsw
        resources
    test:
        java/it/polimi/ingsw
        [resources]
 pom.xml
 README.md
```

ho molti file “indesiderati” (p.es. .DS_Strore)

```
find . -name .DS_Store -print0 | xargs -0 git rm -f --
ignore-unmatch
```

```
echo .DS_Store >> .gitignore
```