



Finanziato
dall'Unione europea
NextGenerationEU



*Ministero dell'Istruzione
e del Merito*



Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA



FINTECH DEVELOPER

Fondamenti di version control

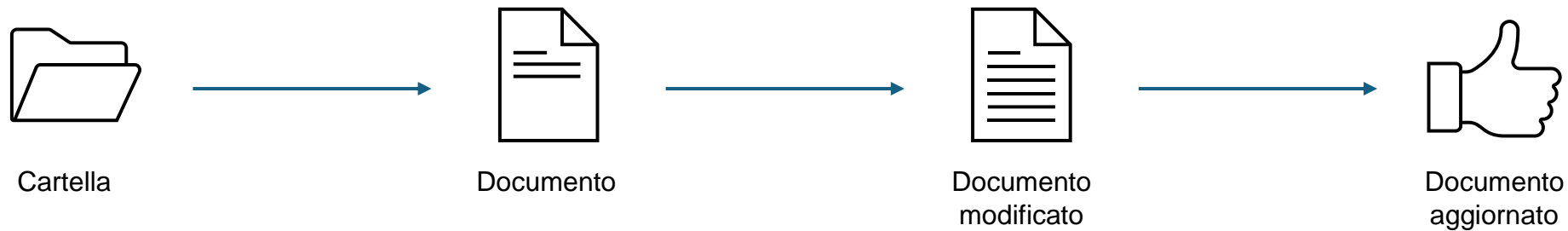
Docente: Loredana Frontino

Titolo argomento: Introduzione al version control

Cos'è il version control?

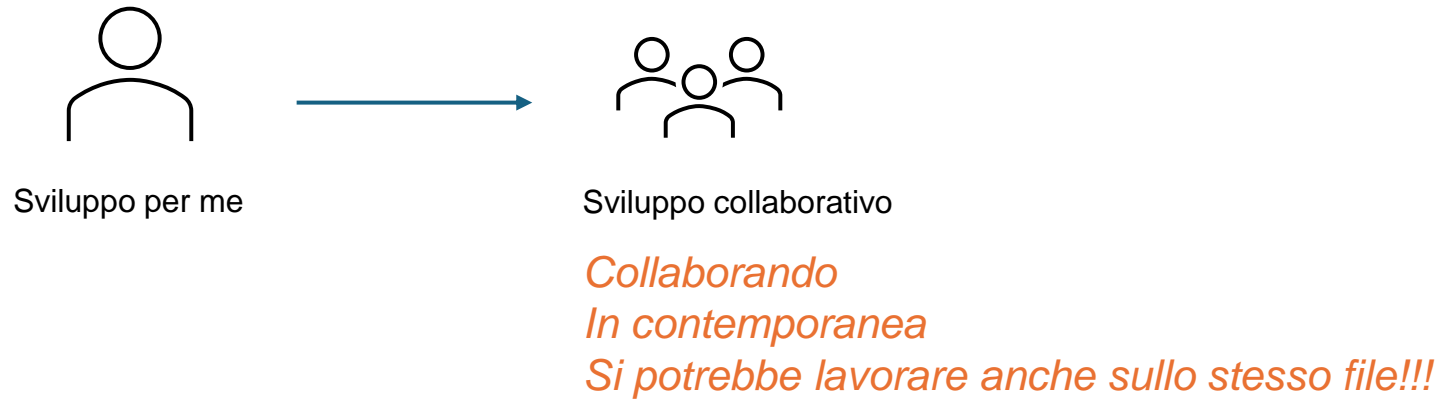
Fondamenti di Version Control – Introduzione al version control

AGGIORNAMENTO DI UN FILE IN UNA CARTELLA SUL PC



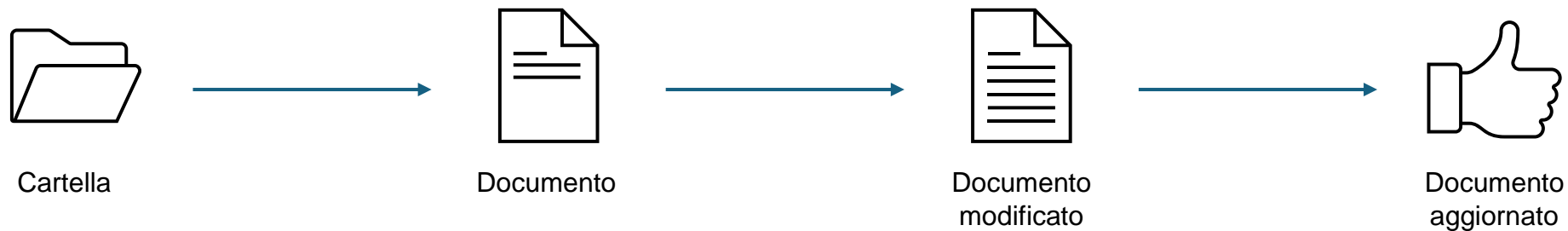
Fondamenti di Version Control – Introduzione al version control

Sviluppo collaborativo



Fondamenti di Version Control – Introduzione al version control

AGGIORNAMENTO DI UN FILE IN UNA CARTELLA SUL PC



- Dov'è lo storico cambiamenti?
- Come riportare un file allo stato precedente?
- Se voglio condividere il mio progetto con altri come gestisco le modifiche?
- Chi ha modificato quel punto?

Fondamenti di Version Control – Introduzione al version control

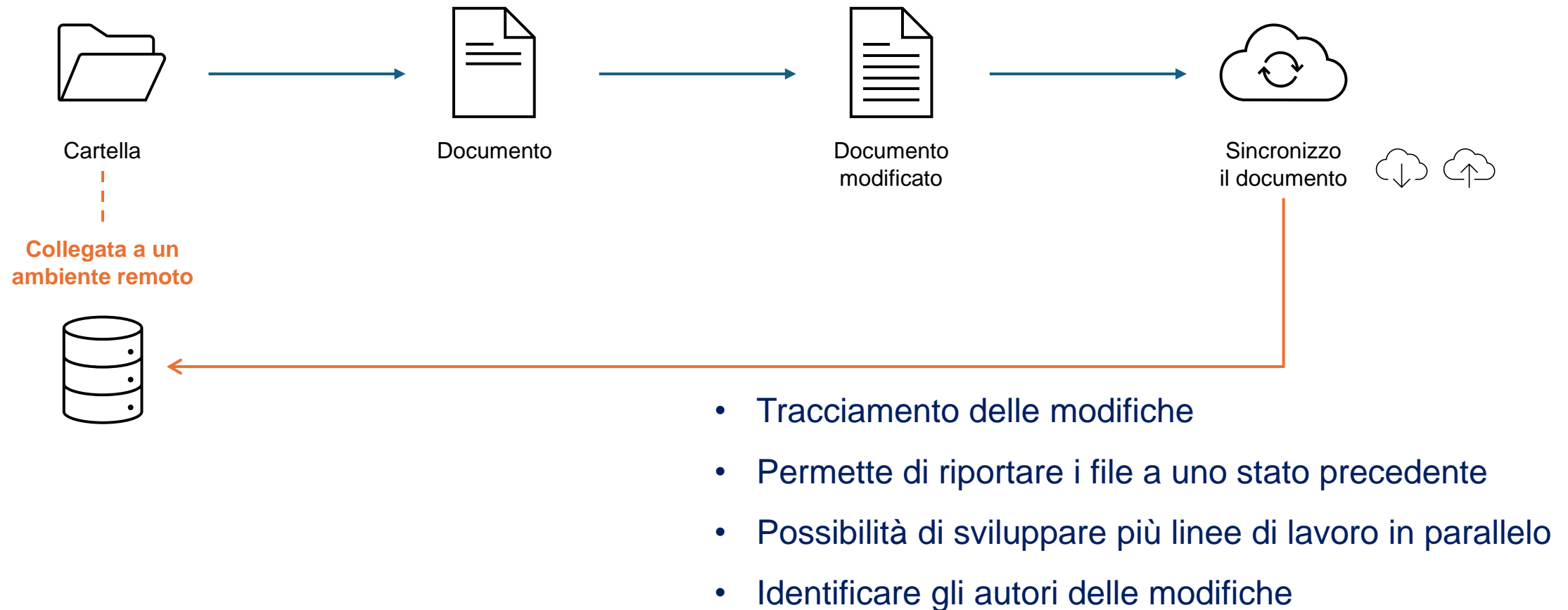
Definizione

Il **Version Control** o versionamento consente di **tracciare i cambiamenti** di un file o a un insieme di file. Permette, tra le altre cose, di **riportare i file** o l'intero progetto **a uno stadio precedente**, **visualizzare le modifiche nel corso del tempo**, sviluppare **più linee di lavoro in parallelo** e **identificare gli autori** delle modifiche.

docs.italia.it

Fondamenti di Version Control – Introduzione al version control

AGGIORNAMENTO DI UN FILE CON IL VERSION CONTROL



Fondamenti di Version Control – Introduzione al version control

Sviluppo collaborativo

Nel momento in cui il codice che scrivo non è più solo “mio”, bisogna:

- Stabilire delle regole per la condivisione di questo codice
- Utilizzare degli strumenti che permettano di applicare queste regole
- Questi strumenti sono detti **Version Control Systems**

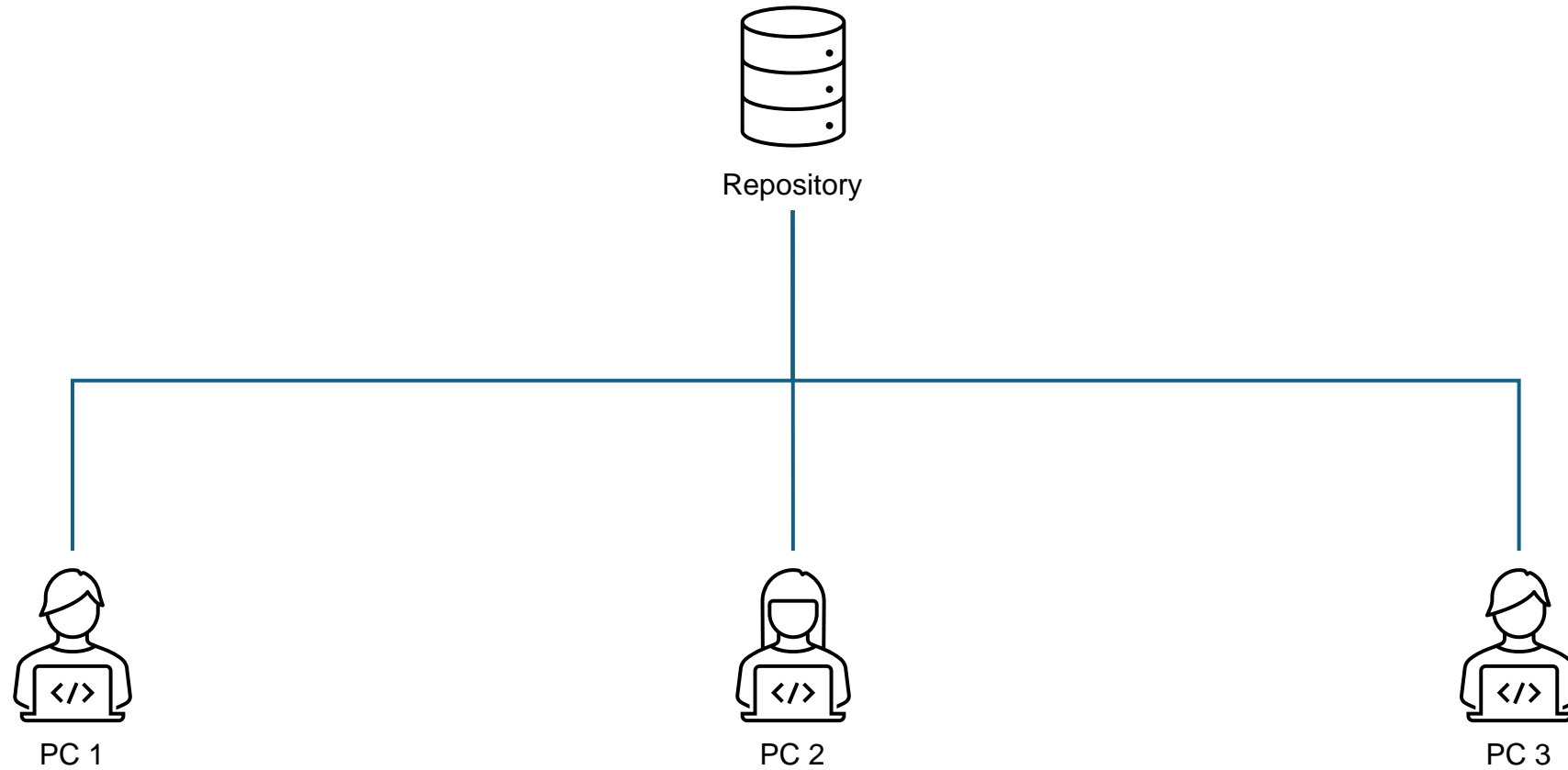
Fondamenti di Version Control – Introduzione al version control

Version Control System (VCS)

- Fornisce supporto alla memorizzazione dei codici sorgenti
- Fornisce uno storico di ciò che è stato fatto
- Può fornire un modo per lavorare in parallelo su diversi aspetti dell'applicazione
- Può fornire un modo per lavorare in parallelo senza intralciarsi a vicenda

Fondamenti di Version Control – Introduzione al version control

Version control systems

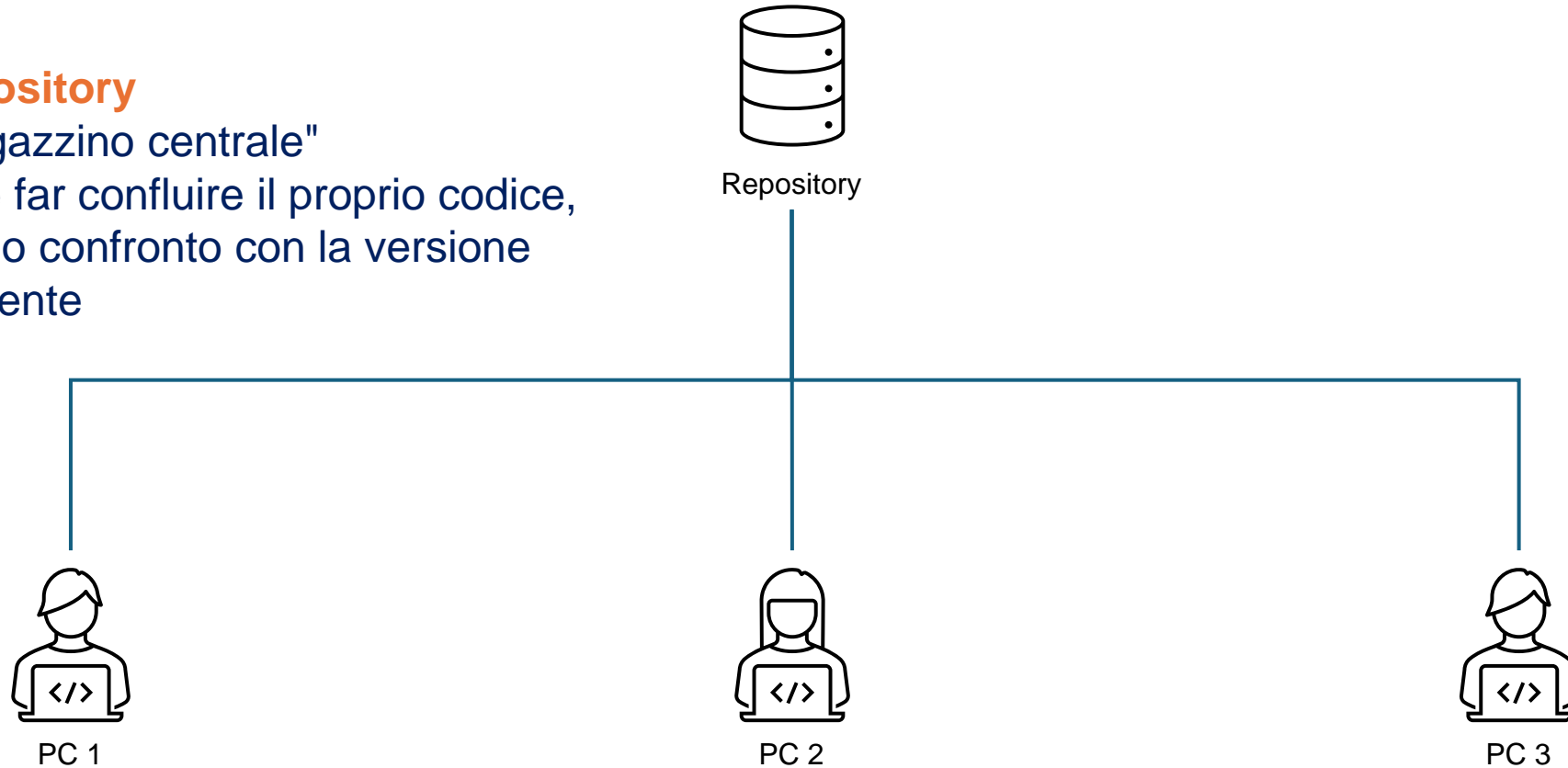


Fondamenti di Version Control – Introduzione al version control

Version control systems

Repository

"magazzino centrale"
dove far confluire il proprio codice,
previo confronto con la versione
esistente



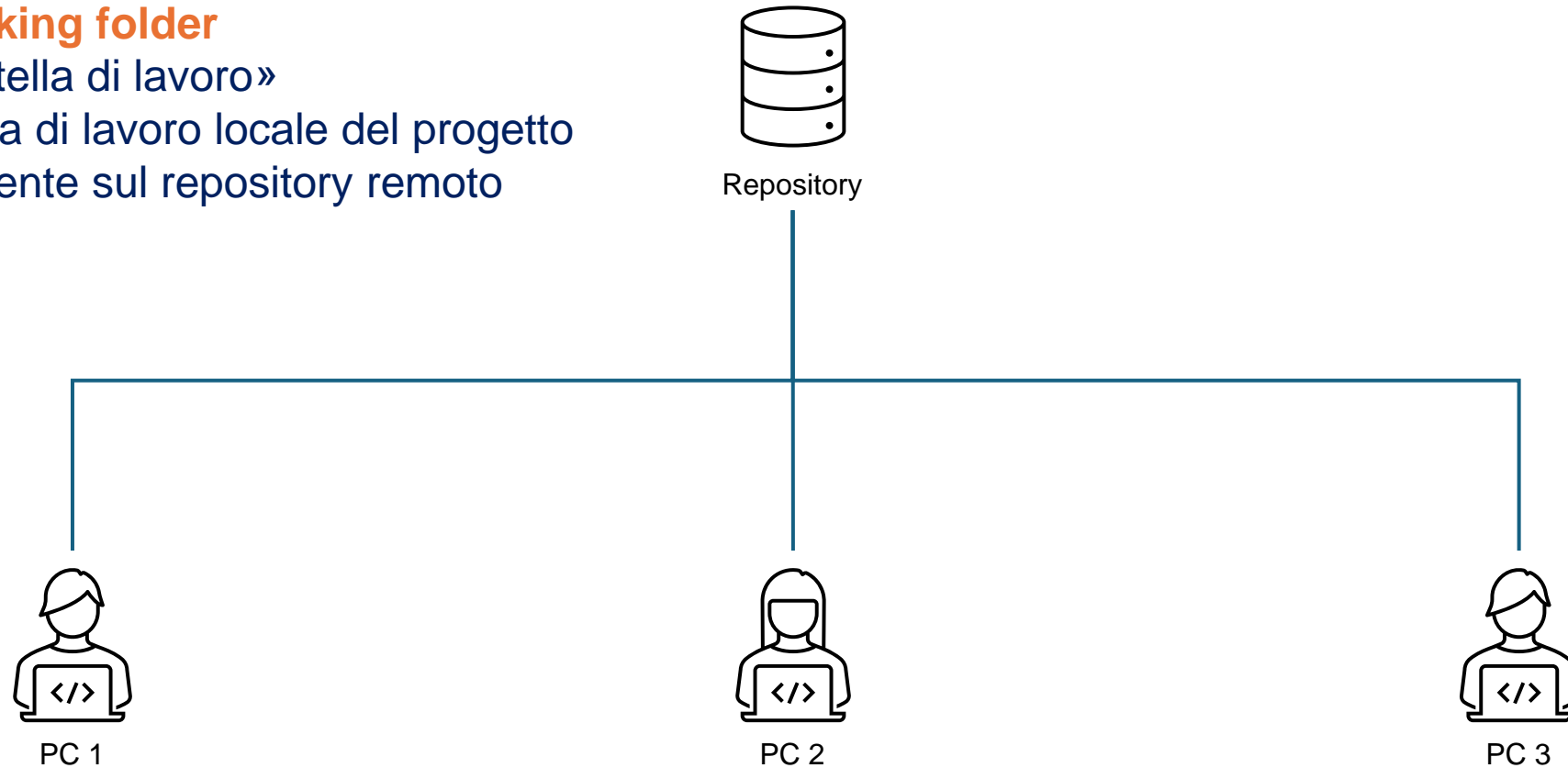
Fondamenti di Version Control – Introduzione al version control

Version control systems

Working folder

«cartella di lavoro»

Copia di lavoro locale del progetto
presente sul repository remoto



Fondamenti di Version Control – Introduzione al version control

Come funziona

1. Ho appena finito una modifica al progetto e ho il mio codice pronto
2. Non lo lancio a caso nel mucchio di codice già esistente (il repository)



Fondamenti di Version Control – Introduzione al version control

Come funziona

1. Ho appena finito una modifica al progetto e ho il mio codice pronto
- ~~2. Non lo lancio a caso nel mucchio di codice già esistente (il repository)~~
3. Confronto il mio codice con la versione presente sul repository, verifico se nel frattempo qualcun altro ha "inviato" modifiche sui miei stessi sorgenti, se queste sono compatibili con le mie ecc...



Fondamenti di Version Control – Introduzione al version control

Come funziona

1. Ho appena finito una modifica al progetto e ho il mio codice pronto
- ~~2. Non lo lancio a caso nel mucchio di codice già esistente (il repository)~~
3. Confronto il mio codice con la versione presente sul repository, verifico se nel frattempo qualcun altro ha "inviato" modifiche sui miei stessi sorgenti, se queste sono compatibili con le mie ecc...

sincronizzazione con il repository sorgente



Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio

1. 3 di voi lavorano su un progetto
2. Avete apportato modifiche tutti alla homepage del progetto
3. PANICO!!!



Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio

1. 3 di voi lavorano su un progetto
2. Avete apportato modifiche tutti alla homepage del progetto
3. ~~PANICO!!!~~
Bisogna fare in modo che si giunga a **una versione unica** che **comprenda tutte le modifiche** effettuate e **non escluda nulla** né di ciò che già esisteva, né delle novità



Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio

1. 3 di voi lavorano su un progetto
2. Avete apportato modifiche tutti alla homepage del progetto
3. **PANICO!!!**
Bisogna fare in modo che si giunga a **una versione unica** che **comprenda tutte le modifiche** effettuate e **non escluda nulla** né di ciò che già esisteva, né delle novità

4. Non buttiamo il codice sul repository sperando nella fortuna

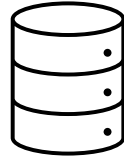
Ma sfruttiamo i vantaggi del **versioning**

Versionamento del codice

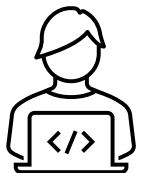


Fondamenti di Version Control – Introduzione al version control

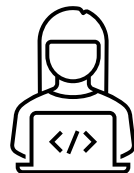
Come funziona - Esempio



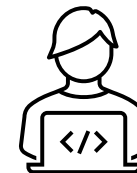
Versione remota



PC 1



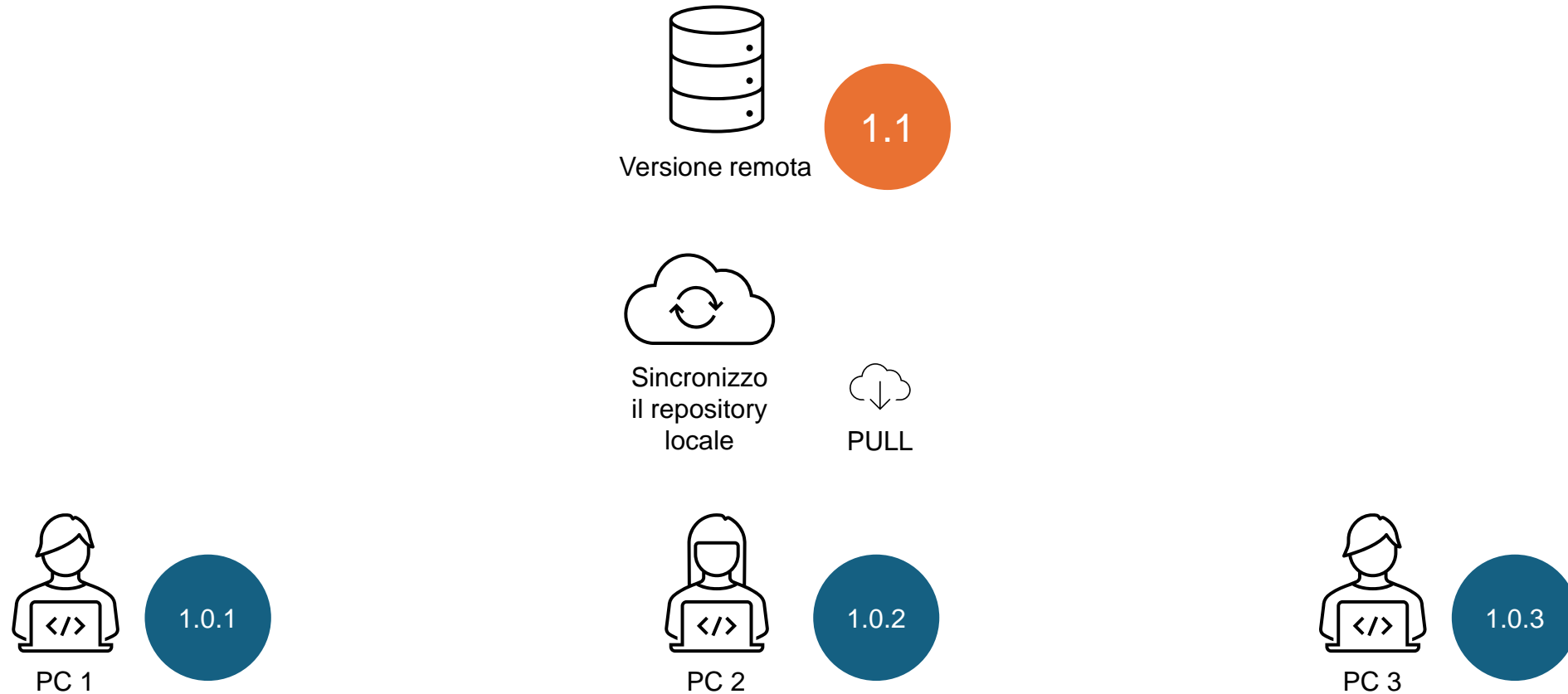
PC 2



PC 3

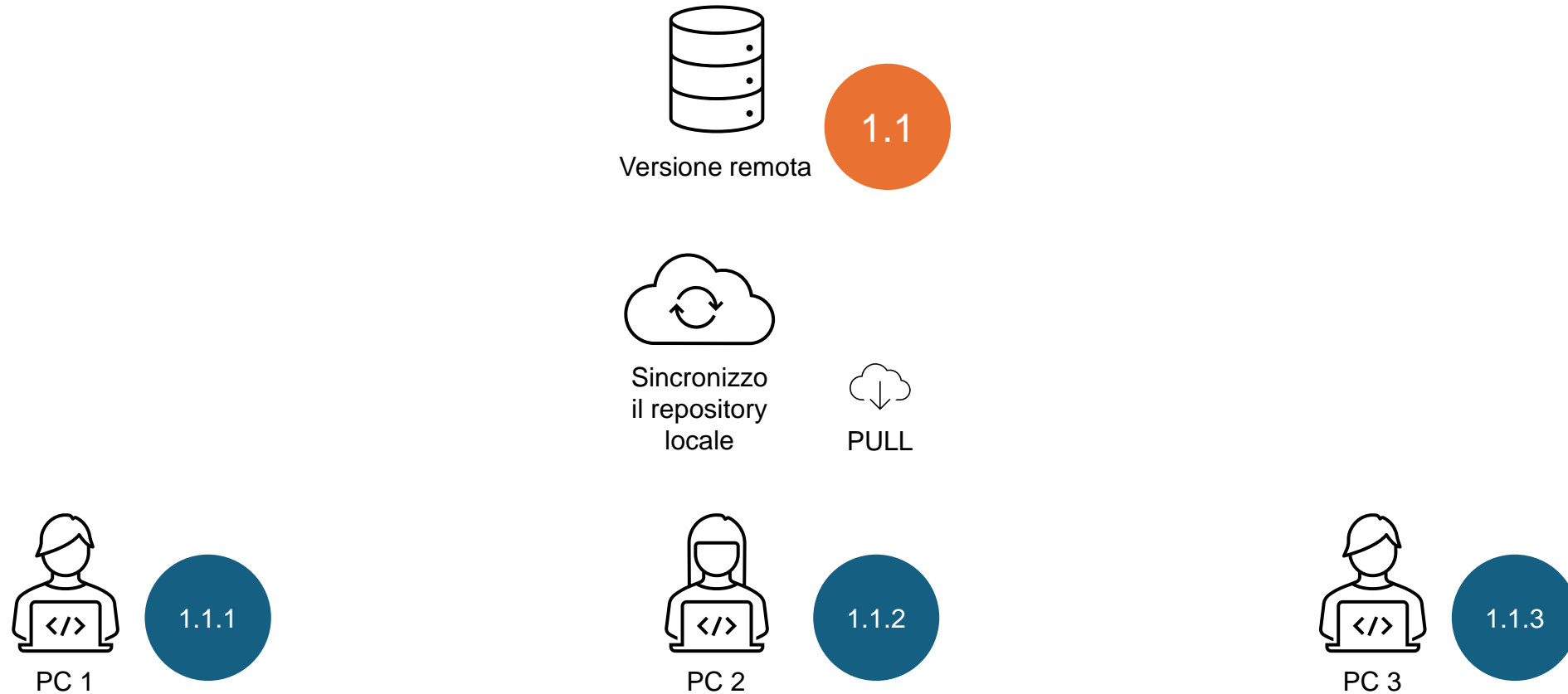
Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio



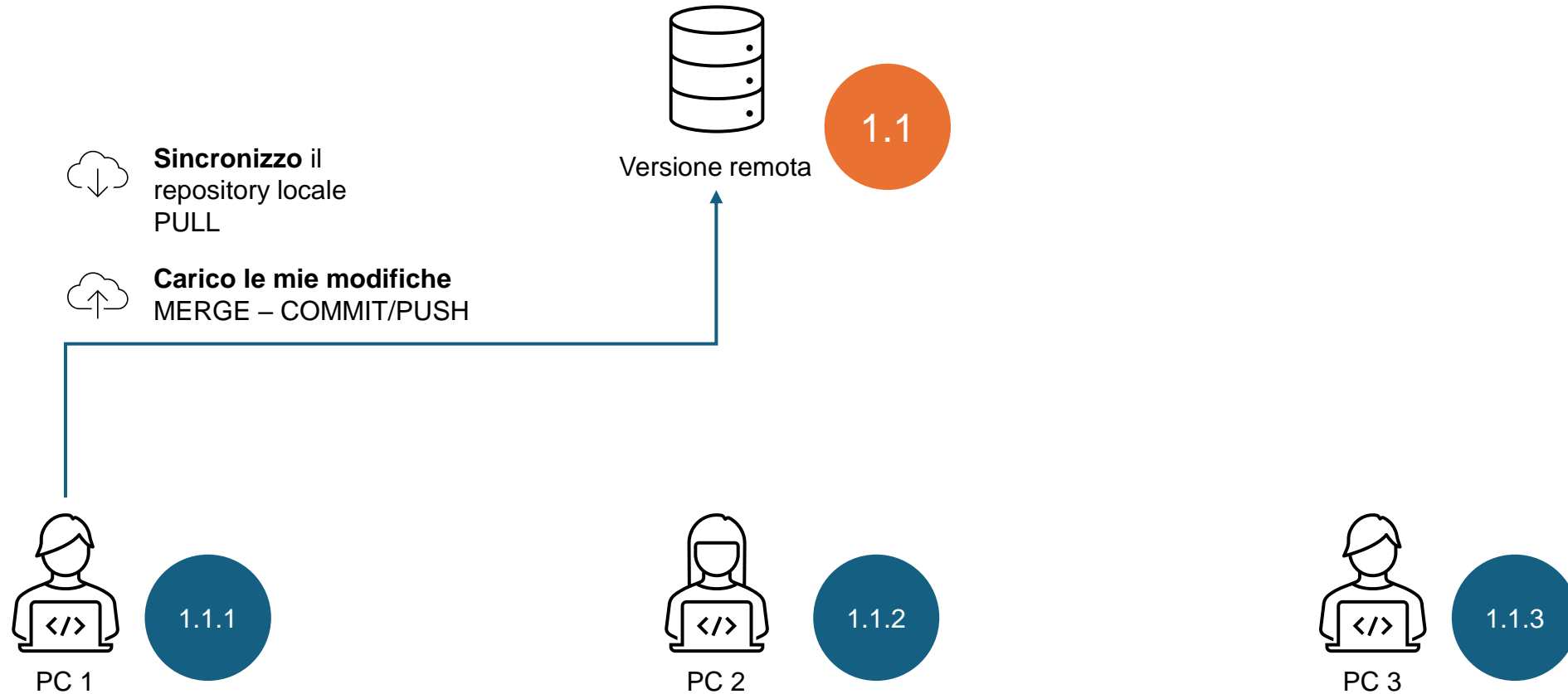
Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio



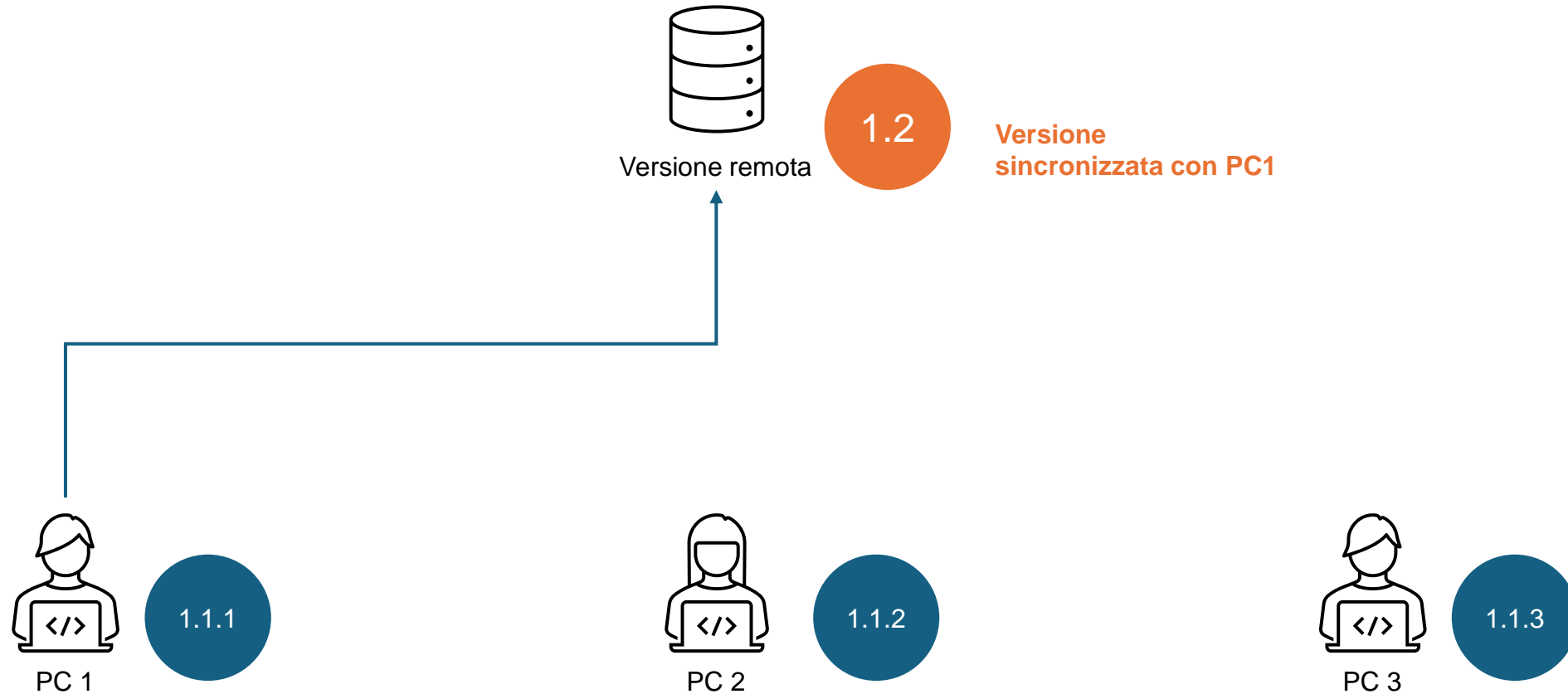
Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio



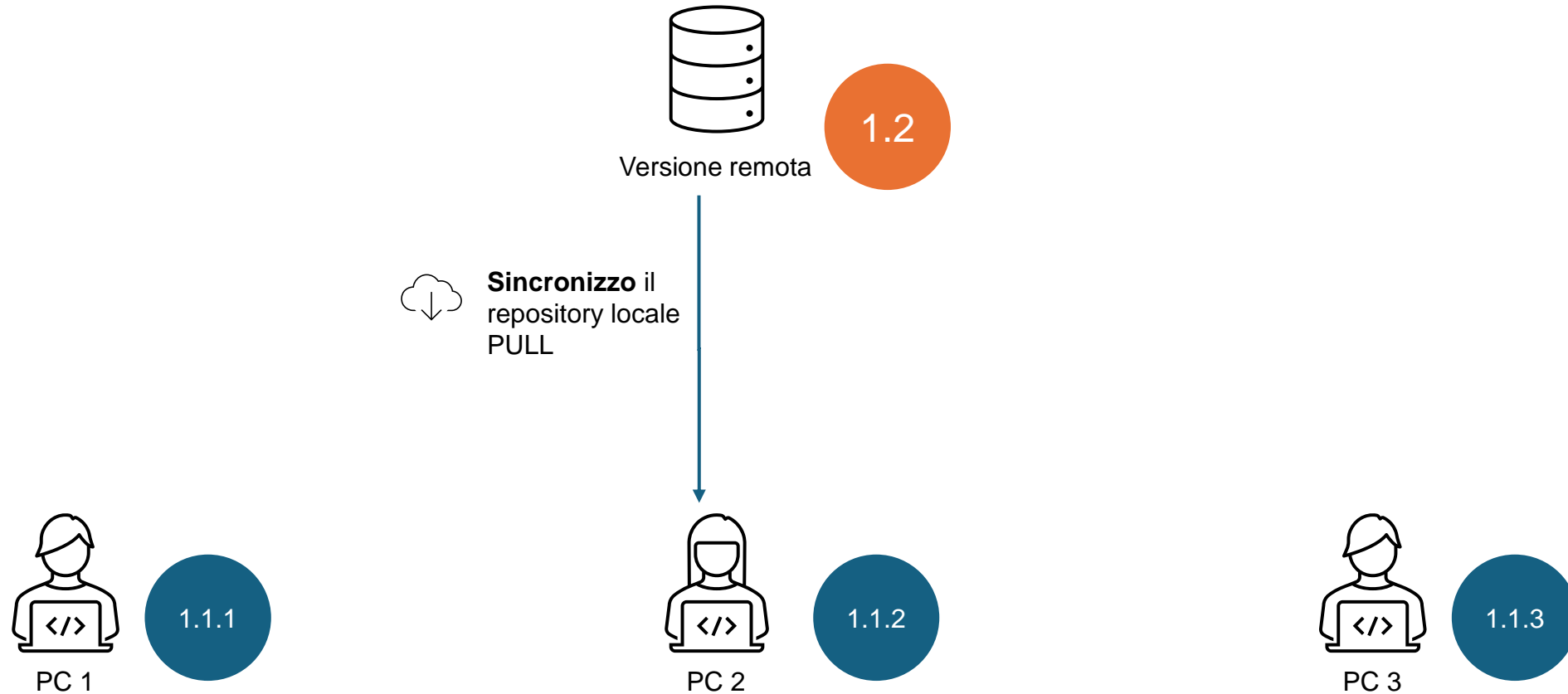
Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio



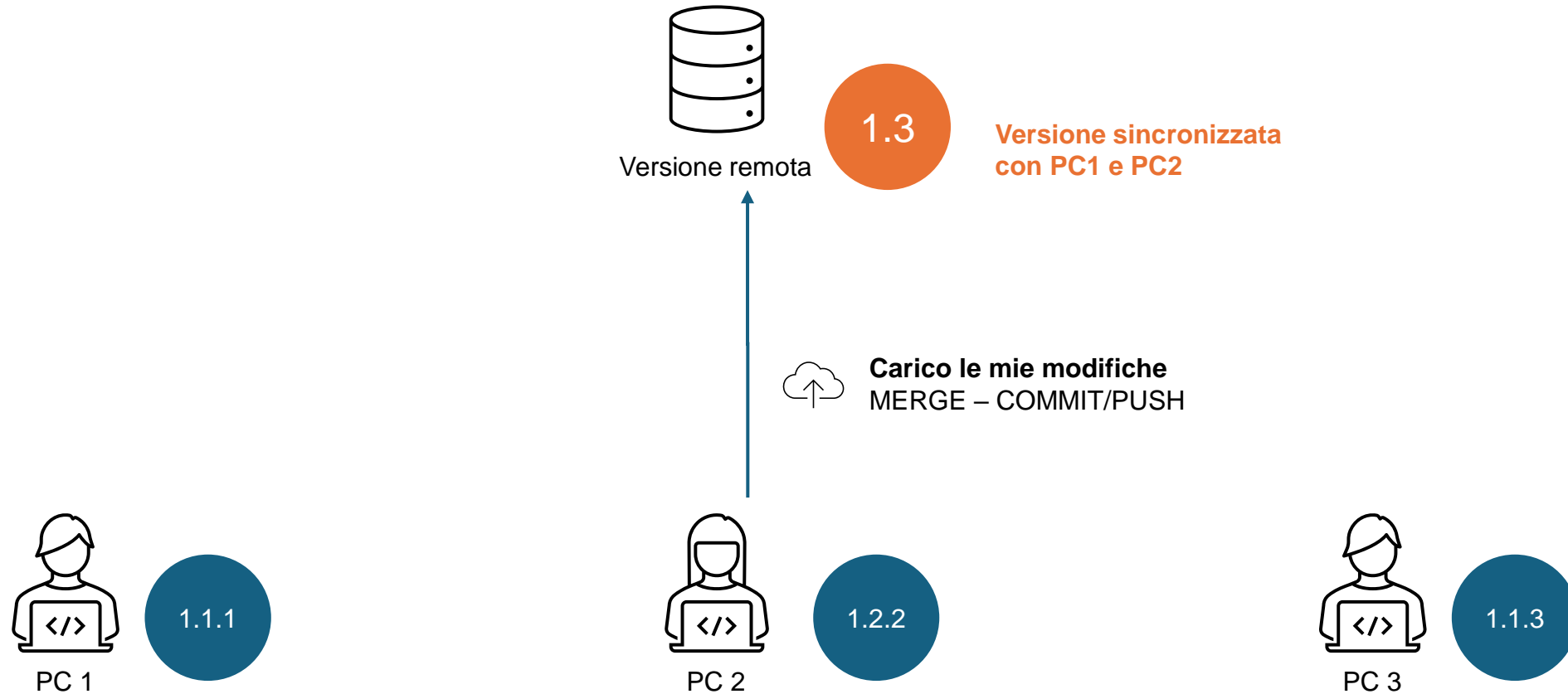
Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio



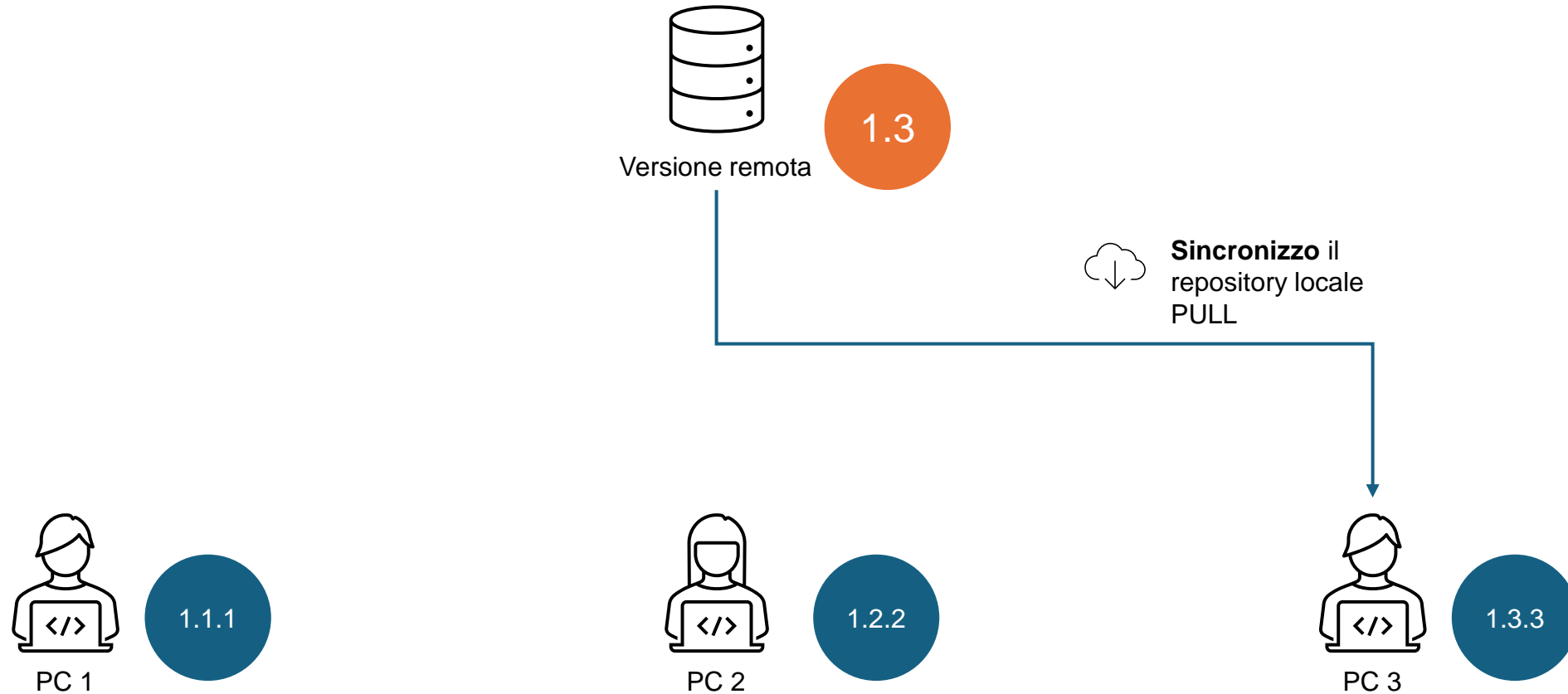
Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio



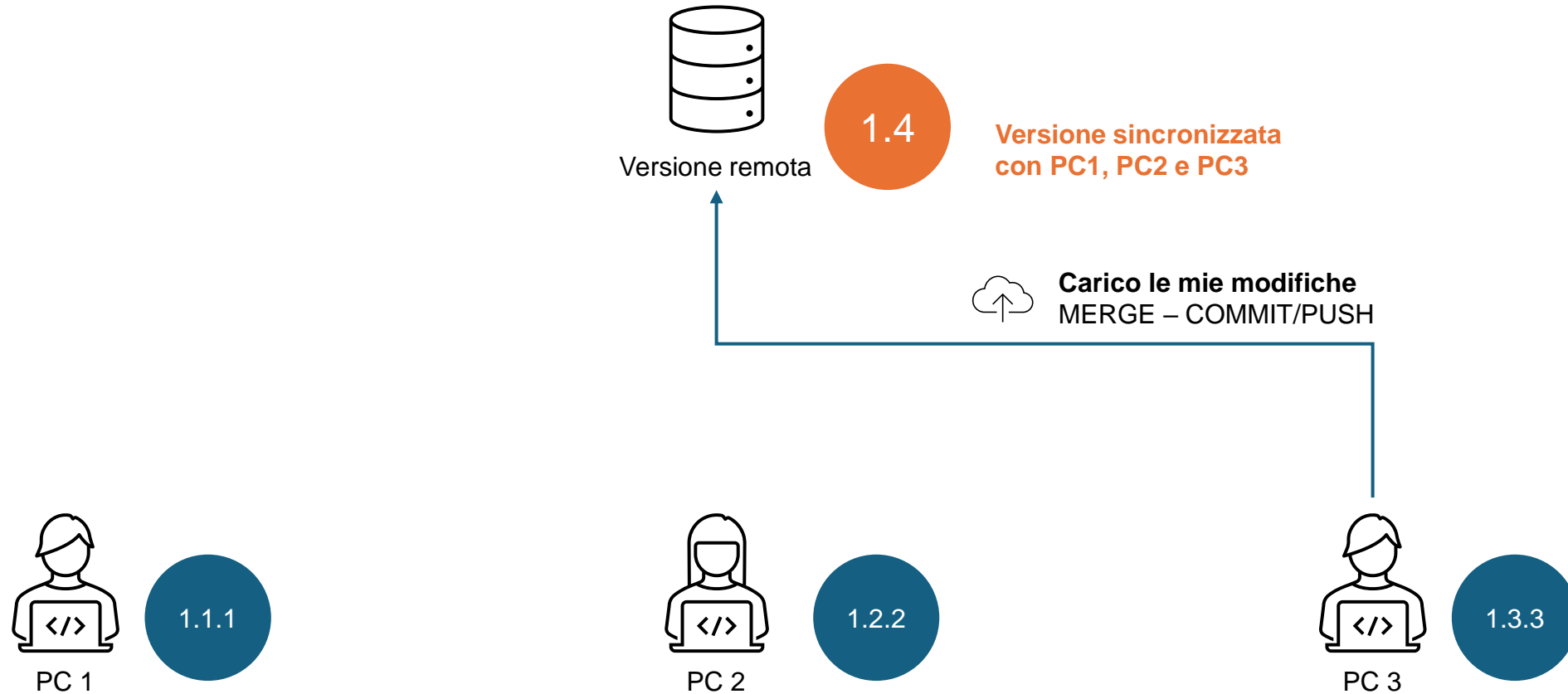
Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio



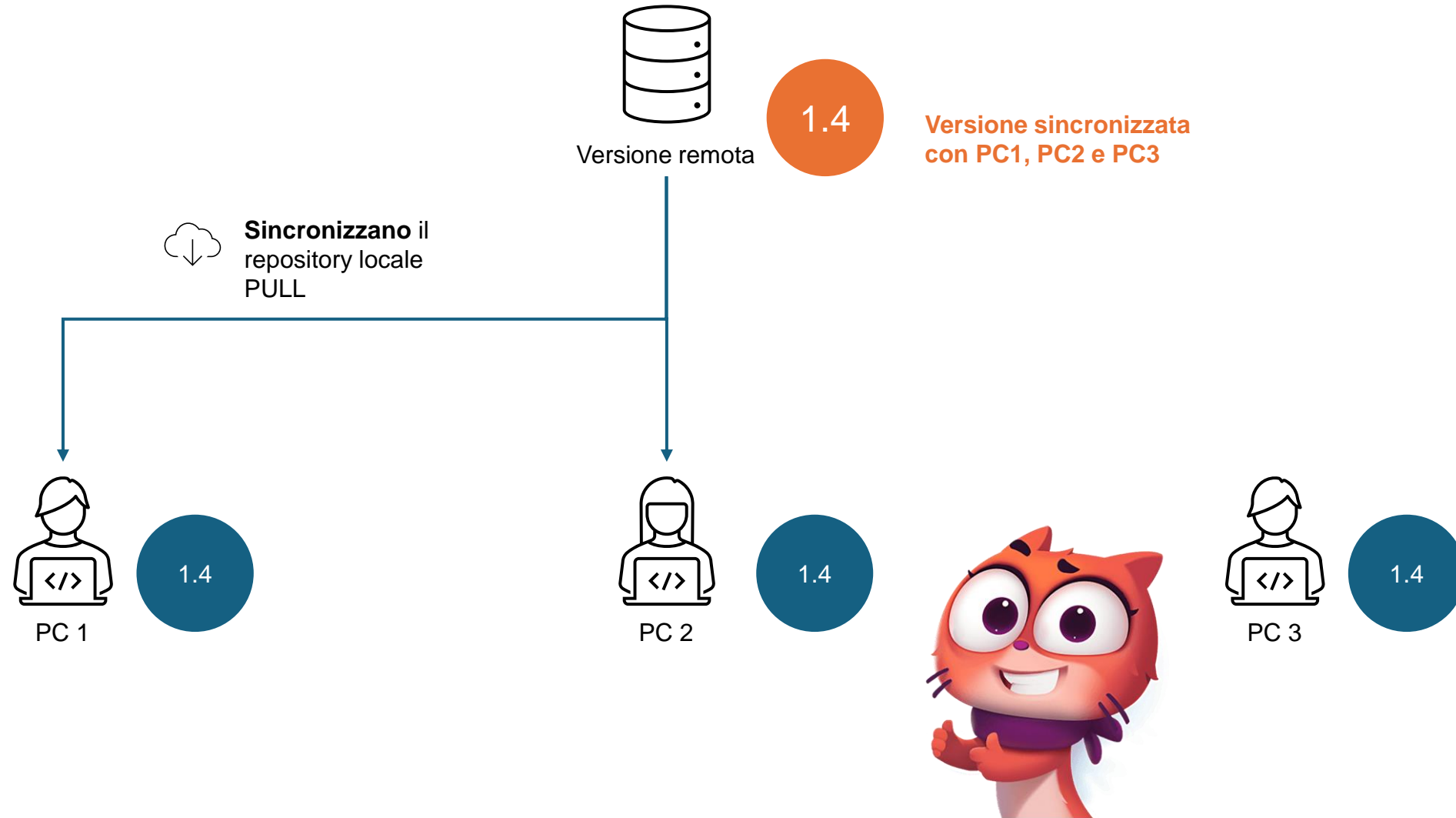
Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio



Fondamenti di Version Control – Introduzione al version control

Come funziona - Esempio



- **Repository:** "contenitore" del codice del progetto, è la "verità assoluta" con cui bisogna confrontarsi ogni volta che si vuole committare (vedi sotto).
- **Commit:** inviare le proprie modifiche al Repository
- **Push:** "spingere" le proprie modifiche dal Repository Locale al Repository Remoto; è necessario aver committato le modifiche prima di effettuare un push
- **Pull:** "tirare giù" le modifiche presenti nel Repository Remoto per averle a disposizione sul Repository Locale. È l'operazione opposta della Push
- **Sincronizzazione:** Verifica della situazione del repository per poter effettuare un confronto con le proprie modifiche (il proprio ambiente di sviluppo); è necessaria una connessione a internet
- **Merge:** "Mixare" le proprie modifiche su un sorgente con quelle presenti sul repository, nel caso in cui qualcuno avesse effettuato un commit sullo stesso file che vogliamo committare. Il risultato deve contenere tutte le modifiche effettuate per non perdere nulla. Può essere automatico, se il sistema è in grado di farlo, o manuale nel momento in cui ci sono modifiche sulla stessa porzione di codice

Link utili

<https://rogerdudler.github.io/git-guide/index.it.html>