GIT CƠ BẢN

Bộ phận Đào tạo

2019/09

【Bí Mật】 # Nội dung

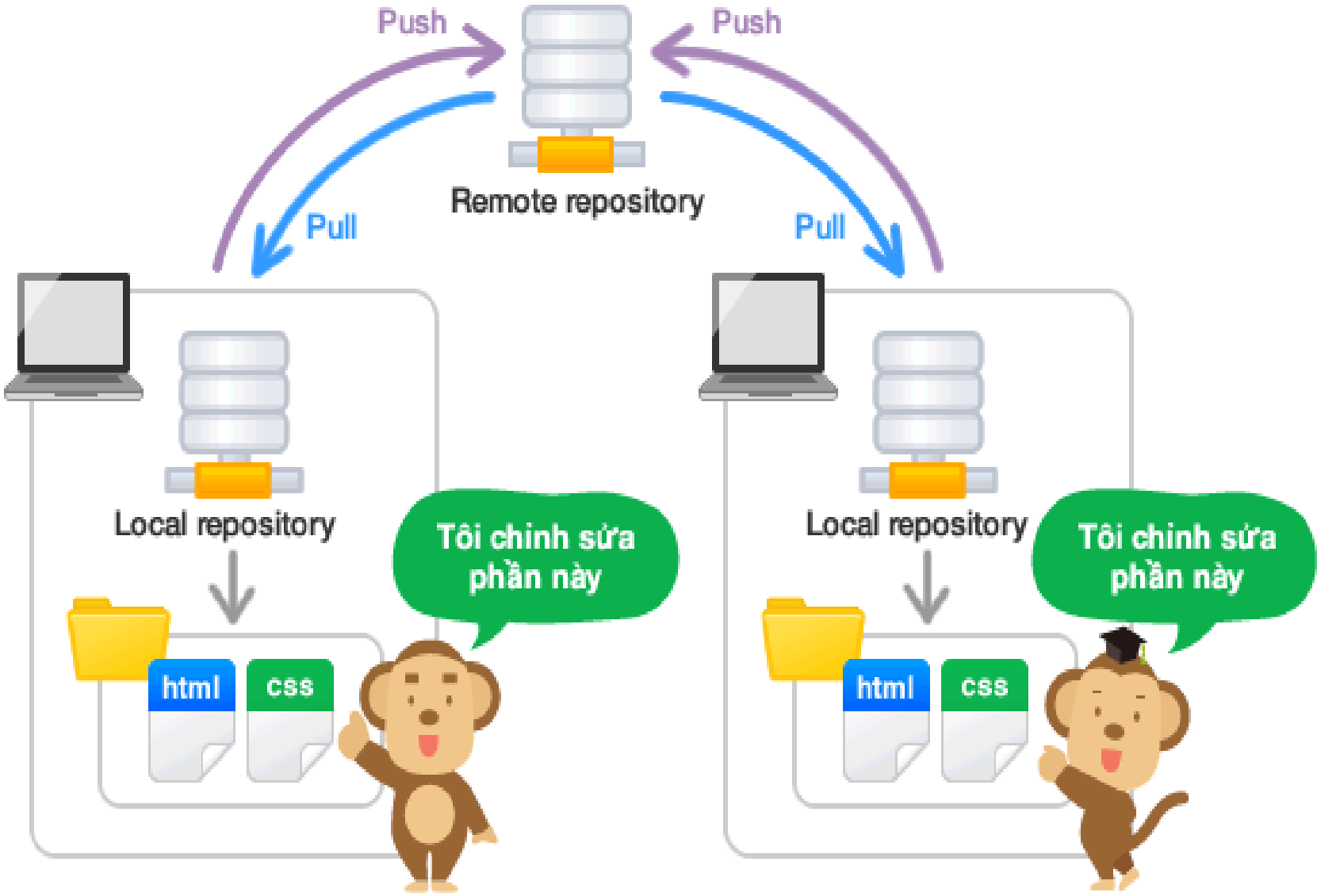
- Repository
- Commit
- Working Tree - Index
- Clone - Pull - Push
- Merge - Conflict
- Branch - Stash

【Bí Mật】 Repository (1)

- Repository là nơi ghi lại trạng thái của thư mục và file

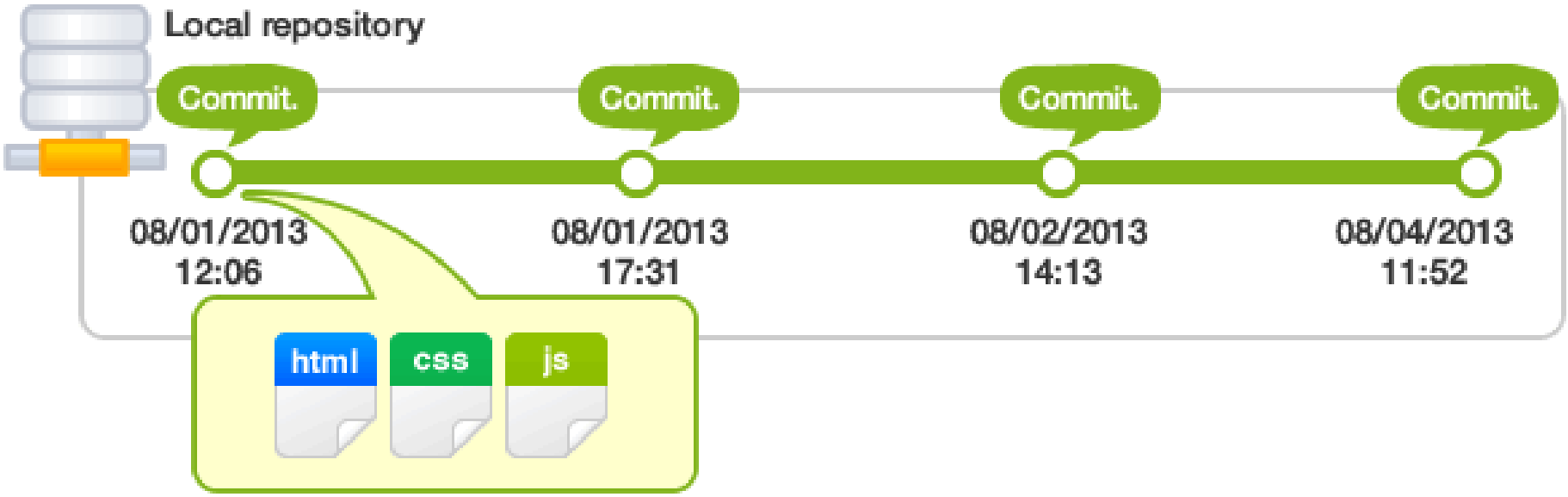
【Bí Mật】 Repository (2)

- Remote
- Local



【Bí Mật】 Commit (1)

- Những thay đổi mang ý nghĩa khác nhau chẳng hạn như thêm chức năng hay sửa lỗi thì hãy cố gắng chia ra rồi commit. Để sau này khi xem lịch sử và tìm kiếm một nội dung thay đổi định sẵn sẽ dễ dàng hơn.



【Bí Mật】 Commit (2) – Mẫu

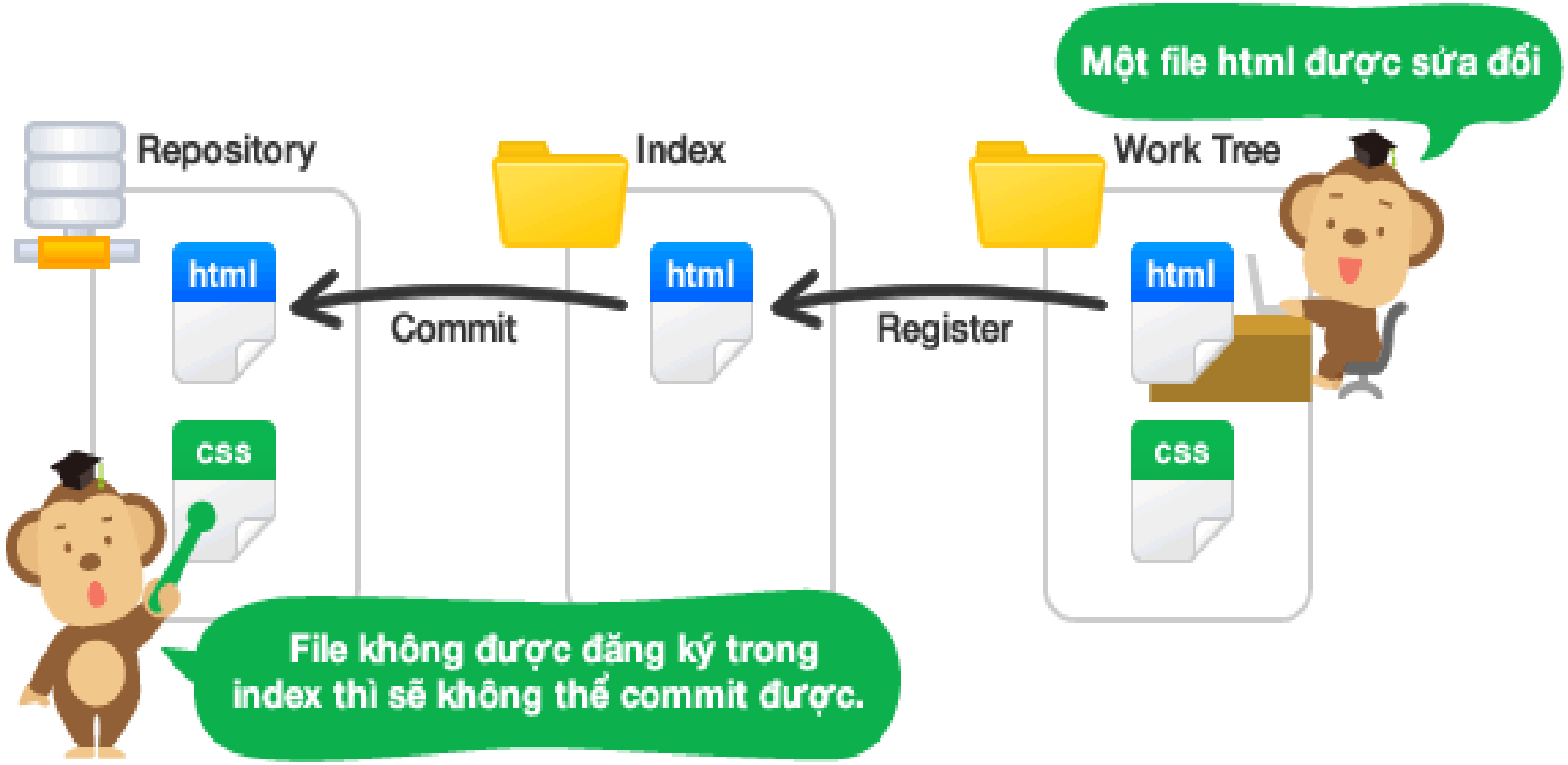
- Dòng thứ 1: refs #MãSốTicket MôTảKháiQuát
- Dòng thứ 2: Dòng trống
- Dòng thứ 3 trở đi: Lý do đã thay đổi
- Ví dụ

  refs #123456 [A03298] [PG301] [A/P List]

  * [fix]: Fix bug caption name
  * [del]: Xoá config database

【Bí Mật】 Working Tree và Index

- Những thư mục được đặt trong sự quản lý của Git mà mọi người đang thực hiện công việc trong thực tế được gọi là WORKING TREE
- Giữa repository và working tree tồn tại một nơi gọi là INDEX. INDEX là nơi để chuẩn bị cho việc commit lên repository

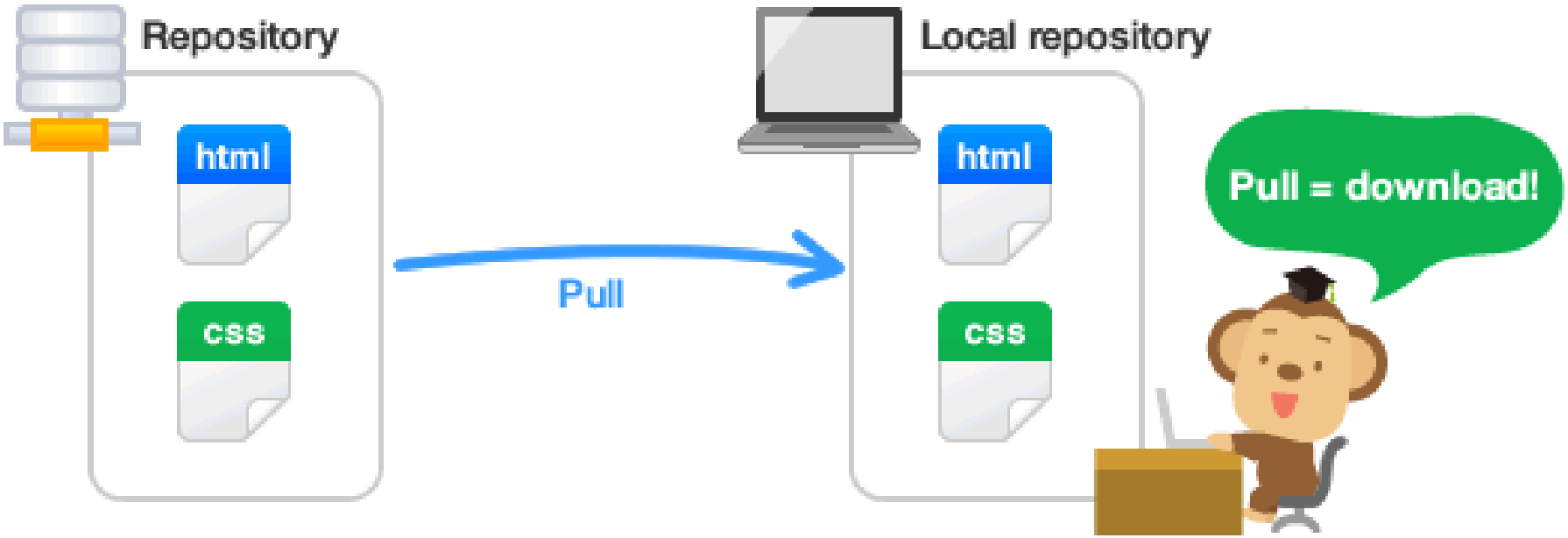


【Bí Mật】 Chia sẻ Repository (1) - Clone

- Để sao chép remote repository, sẽ thực hiện thao tác gọi là "clone".
- Khi thực hiện Clone, sẽ tải về toàn bộ nội dung của remote repository, và có thể tạo thành local repository ở máy khác.

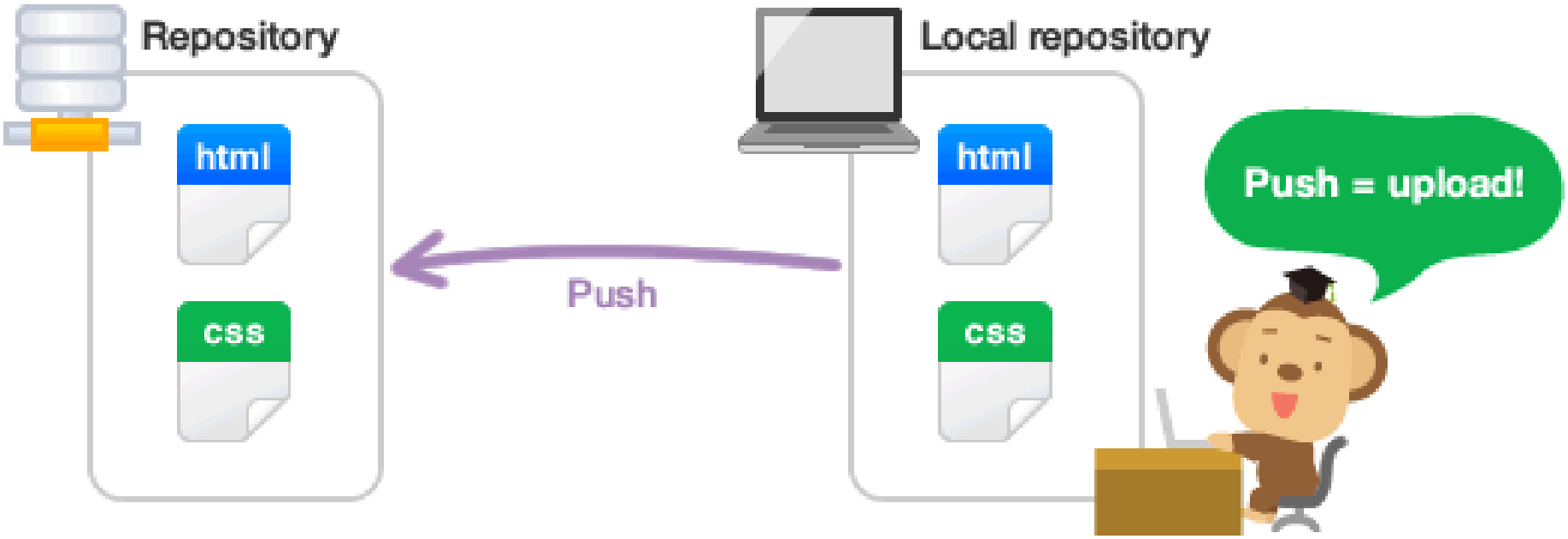
【Bí Mật】 Chia sẻ Repository (2) - Pull

- Để cập nhật local repository từ remote repository thì thực hiện thao tác gọi là Pull.
- Khi thực hiện Pull, sẽ tải lịch sử thay đổi mới nhất từ remote repository về, rồi đưa nội dung đó vào local repository của bản thân.



【Bí Mật】 Chia sẻ Repository (3) - Push

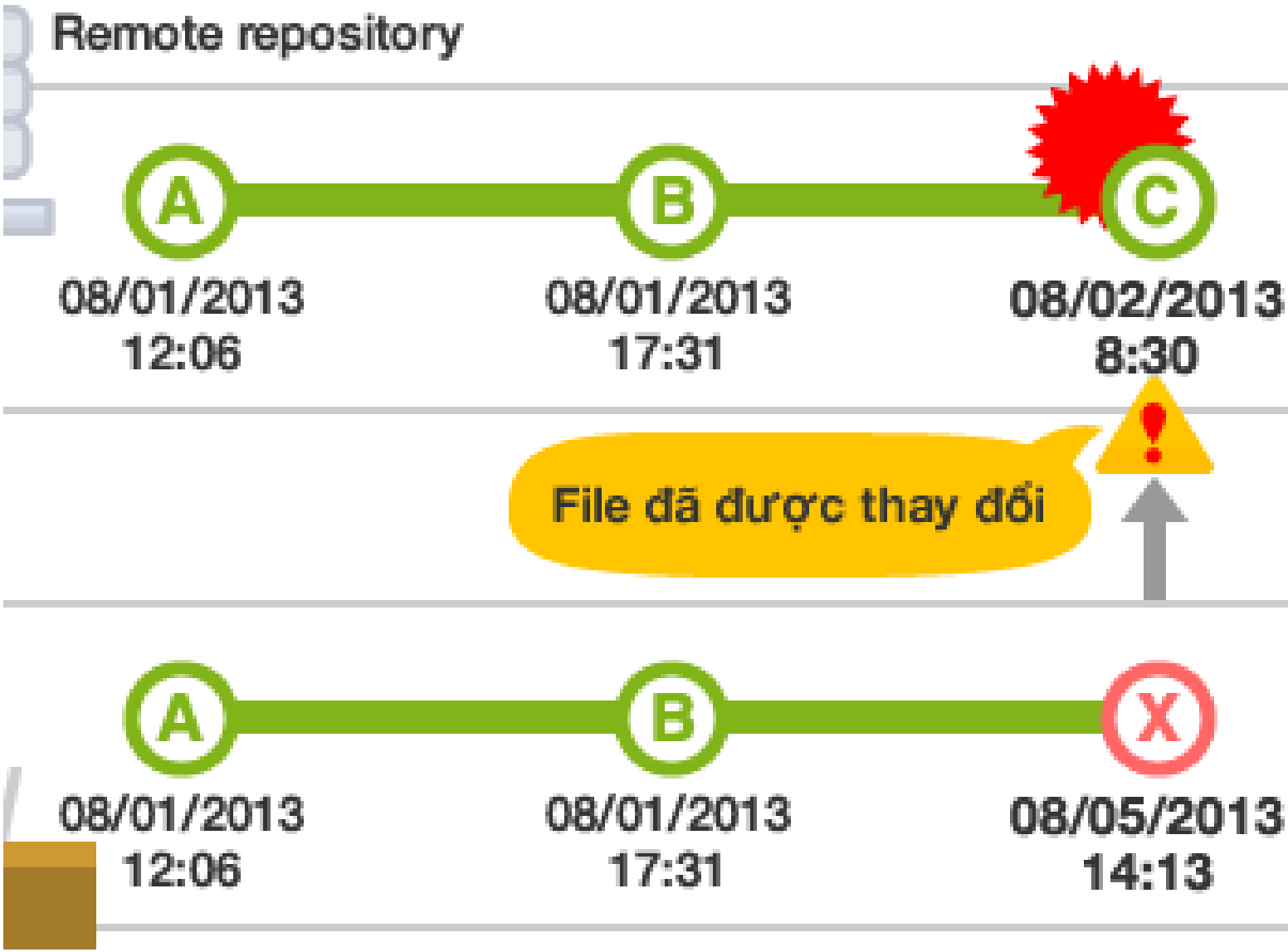
- Để chia sẻ lịch sử thay đổi của local repository mà bản thân đang có bằng remote repository, cần phải upload lịch sử thay đổi trong local repository. Thao tác gọi là PUSH trên Git.
- Khi thực hiện Push, lịch sử thay đổi của bản thân sẽ được upload lên remote repository và lịch sử thay đổi của remote repository sẽ có trạng thái giống với local repository.



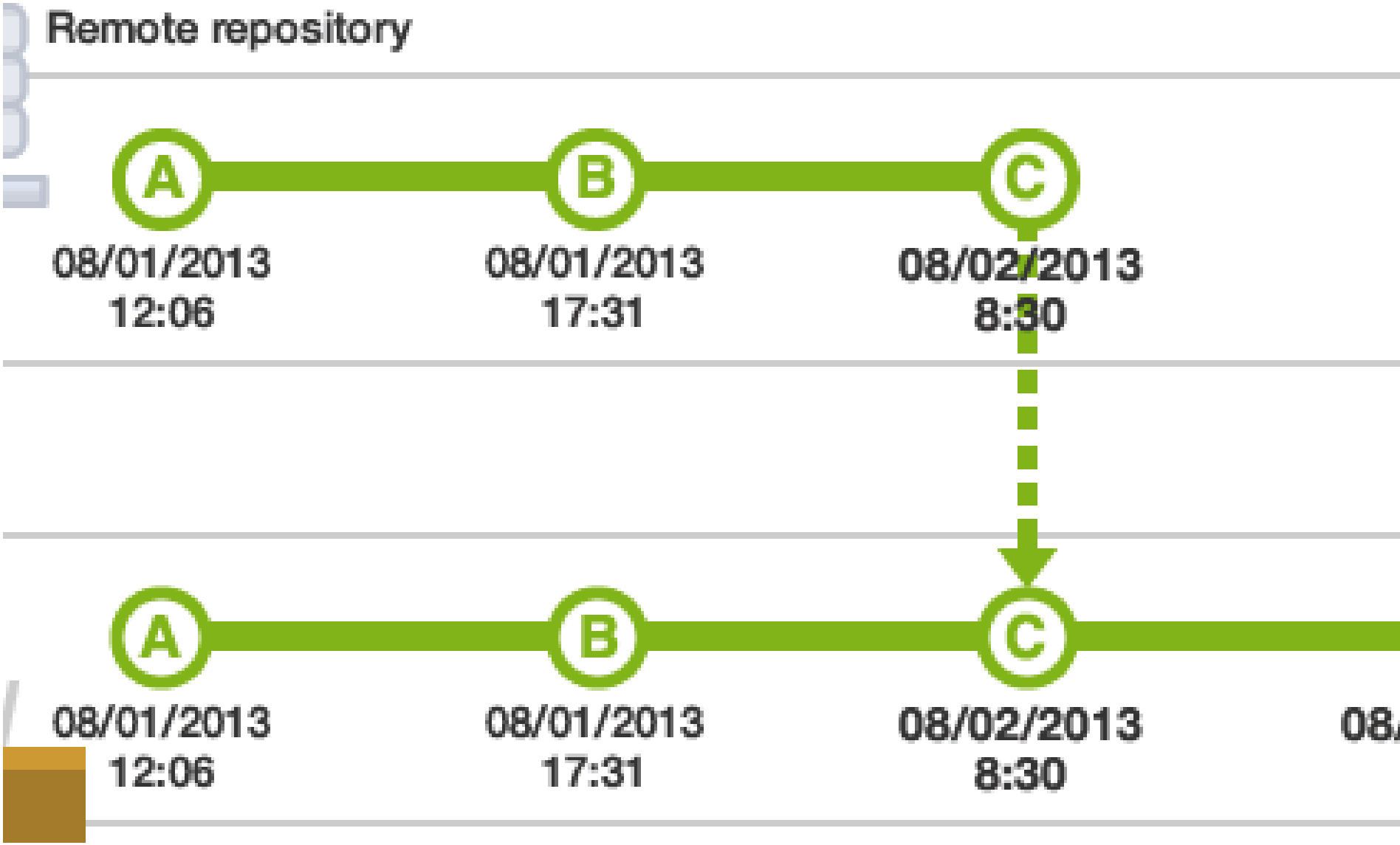
【Bí Mật】 Merge (1) – Vì sao phải Merge

- Trường hợp trong khoảng thời gian từ sau khi pull lần cuối cho đến khi push lần tiếp theo, mà có người khác lỡ push rồi cập nhật remote repository, thì push của chính mình sẽ bị từ chối.
- Lý do là vì khi không merge mà cứ ghi đè lên lịch sử, thì những thay đổi mà người khác đã push (commit C trong sơ đồ) sẽ bị mất

【Bí Mật】 # Merge (2) – Vấn đề



【Bí Mật】 # Merge (3) – Kết quả



【Bí Mật】 Conflict (1) – Khi nào xảy ra?

- Khi thực hiện merge, Git sẽ tích hợp tự động những chỗ thay đổi. Tuy nhiên, cũng có trường hợp không thể tích hợp tự động được.
- Đó là trường hợp đã thay đổi ở nơi giống nhau trong file trên remote repository và local repository. Trường hợp này, vì nó không thể tự phán đoán được lấy thay đổi từ bên nào nên sẽ phát sinh lỗi.

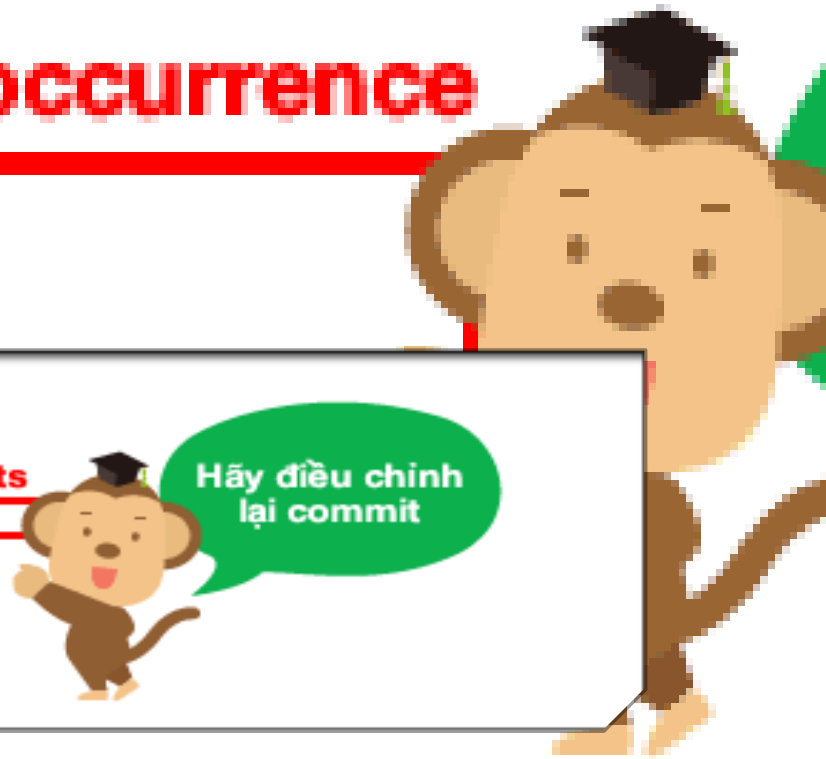
【Bí Mật】 Conflict (2) – Giải quyết xung đột

itle>

Example of a conflict occurrence

```
1  <html>
2    <head>
3      <title>hello</title>
4    </head>
5    <body>
6      <strong>Hello</strong>
7    </body>
8  </html>
```

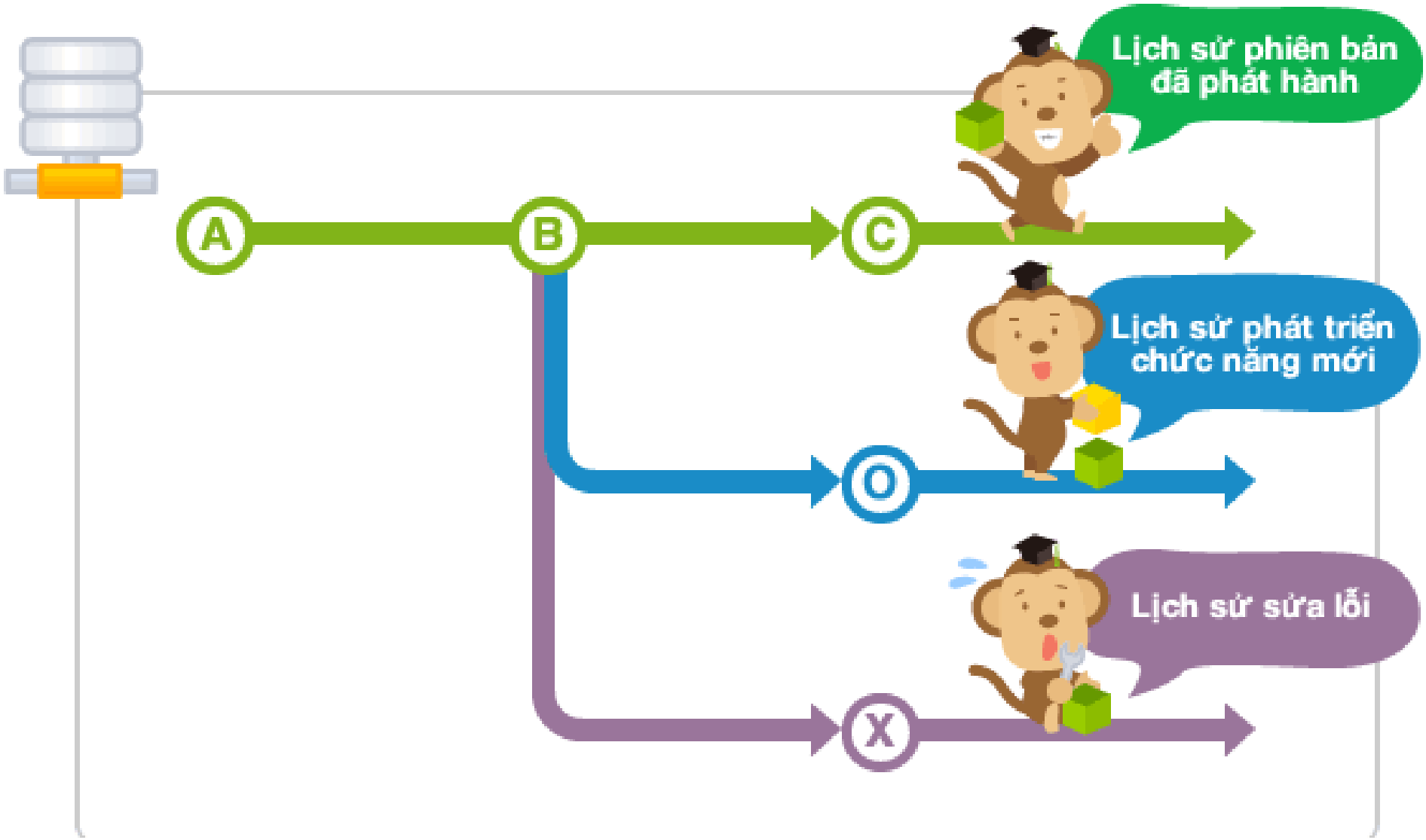
Revising the part of conflicts



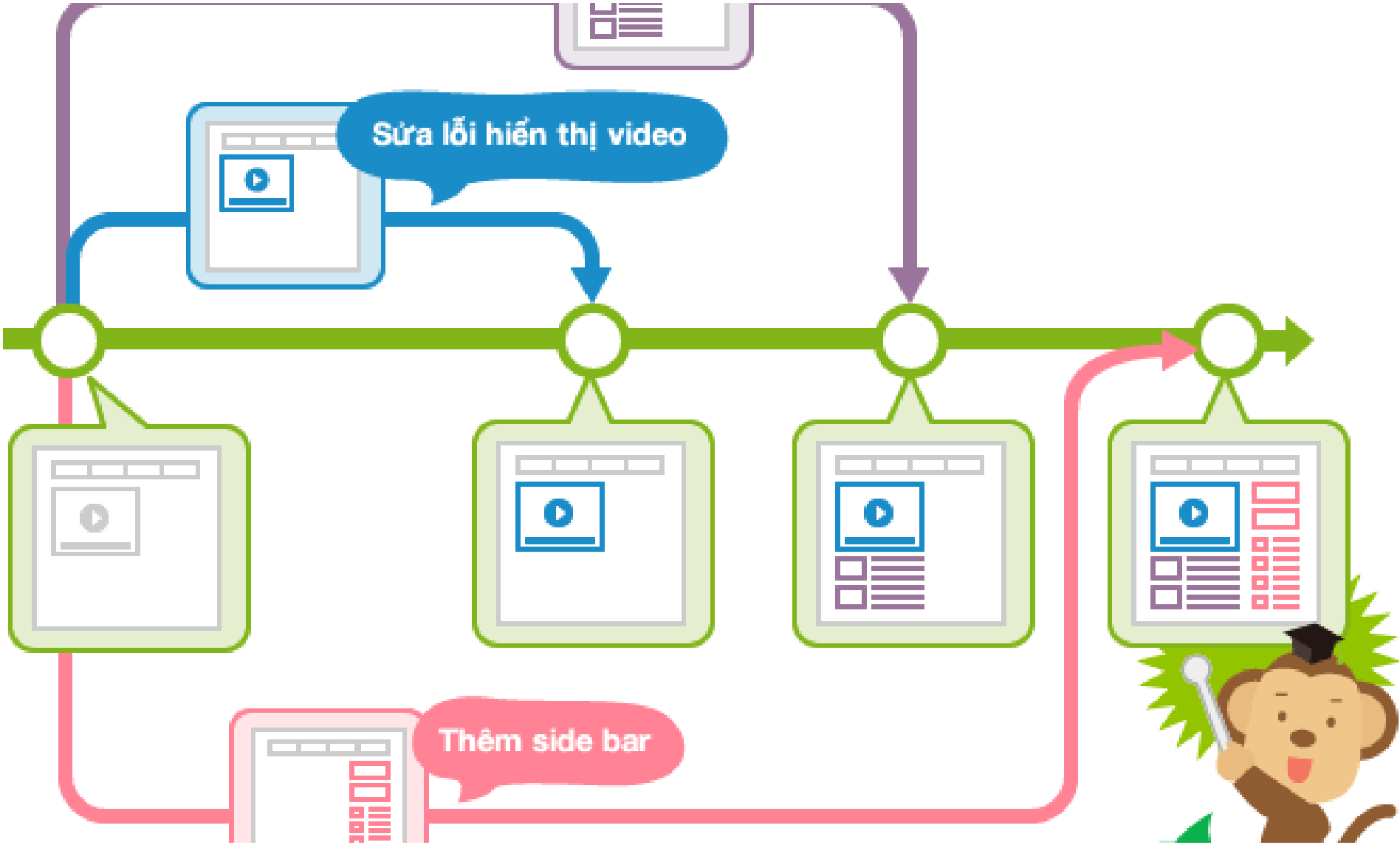
【Bí Mật】 # Conflict (3) – Giải quyết xung đột

- Khi thực hiện merge, Git sẽ tích hợp tự động những chỗ thay đổi. Tuy nhiên, cũng có trường hợp không thể tích hợp tự động được.
- Đó là trường hợp đã thay đổi ở nơi giống nhau trong file trên remote repository và local repository. Trường hợp này, vì nó không thể tự phán đoán được lấy thay đổi từ bên nào nên sẽ phát sinh lỗi.

【Bí Mật】 Branch (1) – Khái quát



【Bí Mật】 Branch (2) – Khái quát

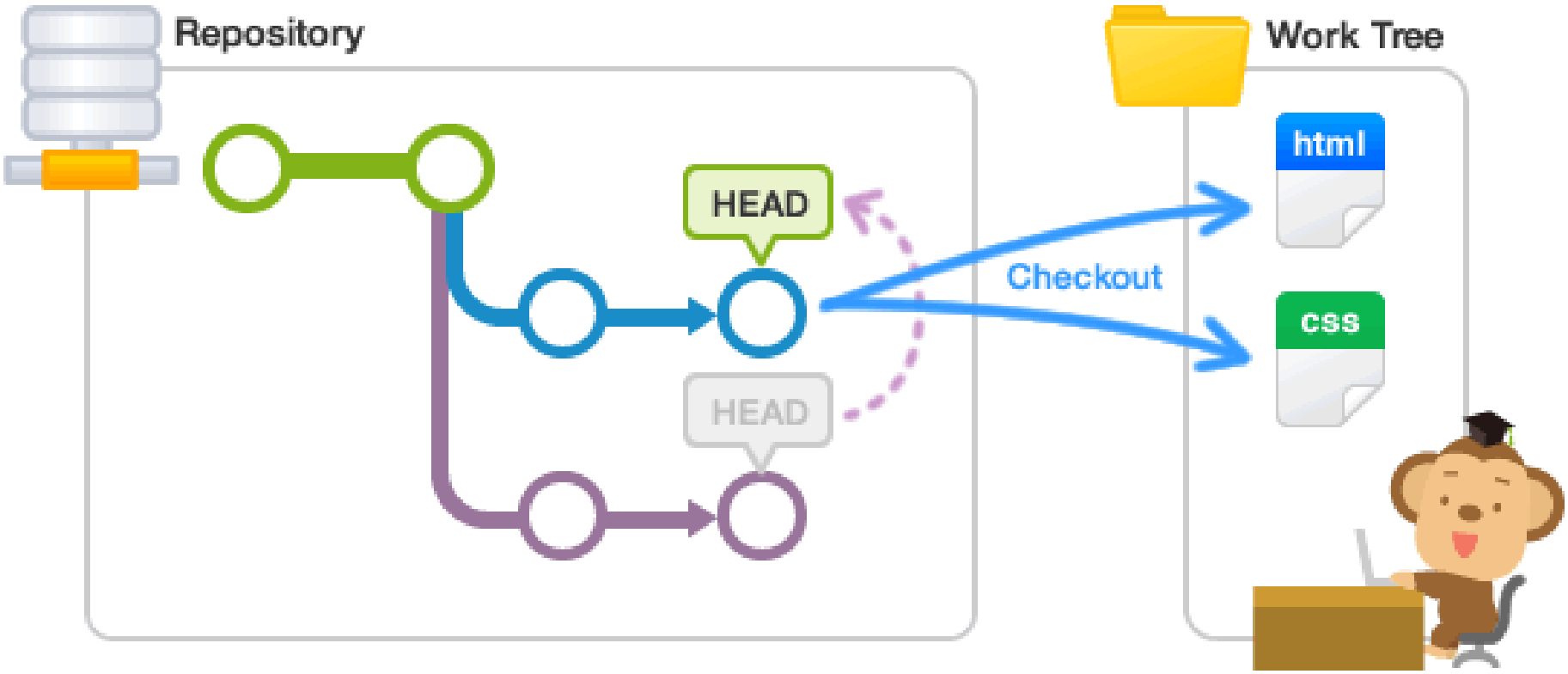


【Bí Mật】 # Branch (3) – git-flow

- git-flow là một tập các thao tác mở rộng của git nhằm cung cấp các thao tác repository ở mức cao dựa trên mô hình phân nhánh
  - master
  - develop
  - feature
  - release
  - hotfix

【Bí Mật】 Branch (3) – Checkout

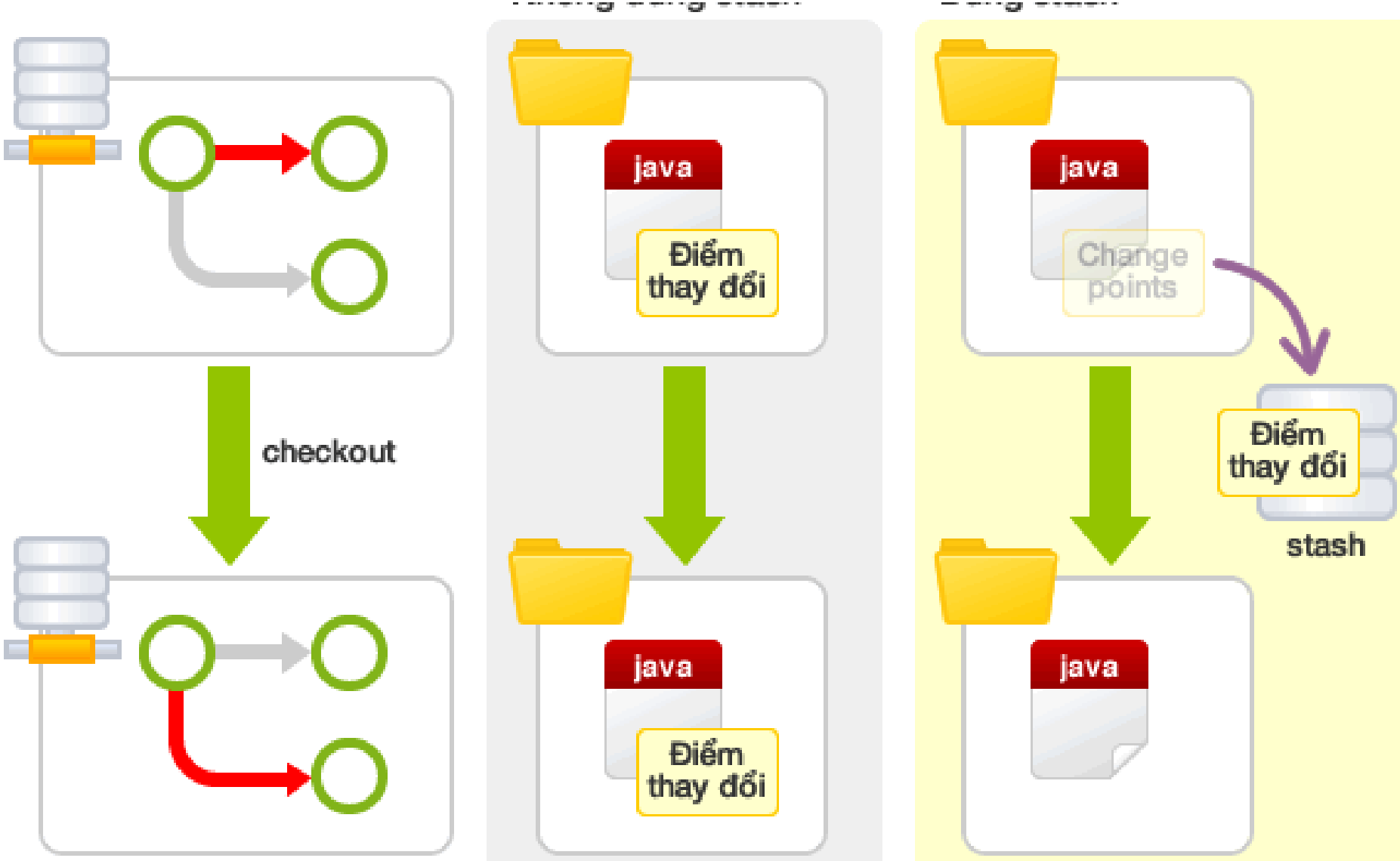
- Để chuyển đổi branch làm việc thì sẽ thực hiện thao tác gọi là CHECKOUT.
- Khi thực hiện checkout, trước tiên nội dung của lần commit cuối cùng trong branch chuyển đến sẽ được mở ra trong work tree. Và commit đã tiến hành sau khi check out thì sẽ được thêm vào branch sau khi di chuyển đến.



【Bí Mật】 Branch (3) – Head và Stash

- HEAD là tên hiển thị phần đầu của branch đang sử dụng hiện tại. Mặc định là đang hiển thị phần đầu của master. Bằng việc di chuyển HEAD thì branch đang sử dụng sẽ được thay đổi.
- Stash là khu vực ghi lại tạm thời nội dung thay đổi của file. Bằng việc sử dụng stash, trong work tree và index, những thay đổi chưa được commit có thể lưu lại tạm thời. Những thay đổi lưu tạm này về sau có thể lấy ra và hiển thị trên branch ban đầu hay là phản ánh lên branch khác.

【Bí Mật】 Branch (4) – Stash



【Bí Mật】 Nguồn tham khảo

- GIT Cơ bản
  - https://backlog.com/git-tutorial/
- GIT Develop
  - https://backlog.com/git-tutorial/vn/stepup/stepup1_1.html
- Khác
  - https://backlog.com/git-tutorial/vn/reference/
- Git-flow
  - https://danielkummer.github.io/git-flow-cheatsheet/index.vi_VN.html