

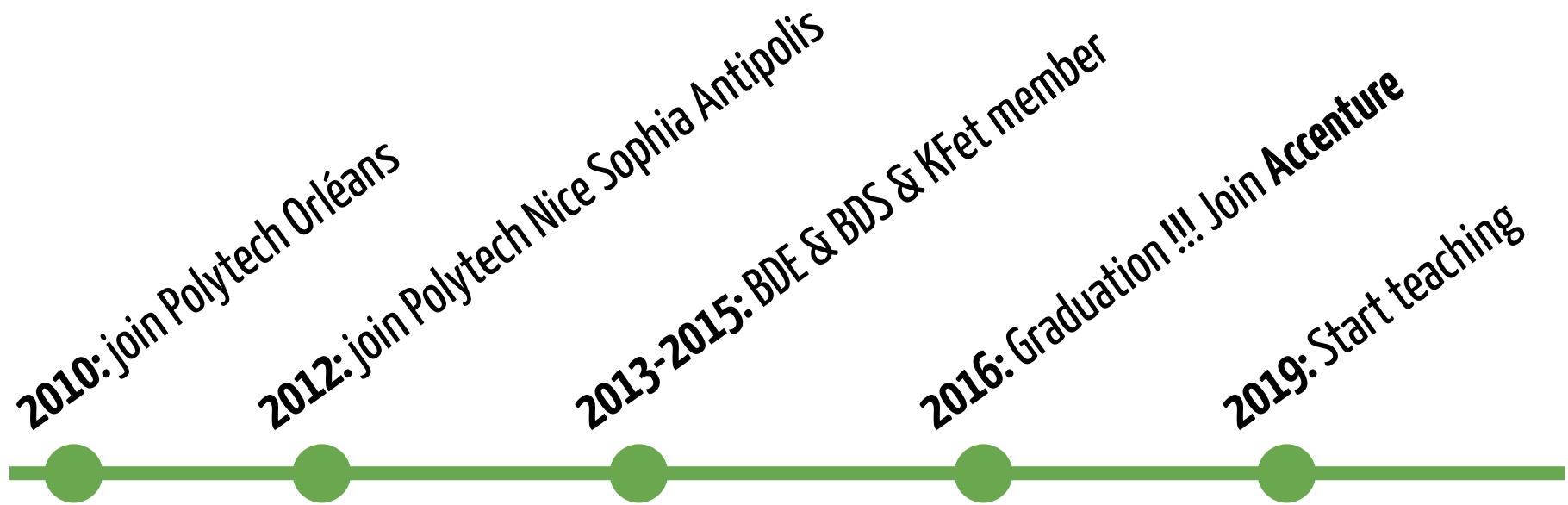


Collaborative Development & Source Code Versioning

Mathias COUSTÉ
22.09.2021



Who am I?



Who am I?

1 Test yourselves

Why code versionning **2**

3 Introduction to Git

Case study **4**

1



Go Kahoot!

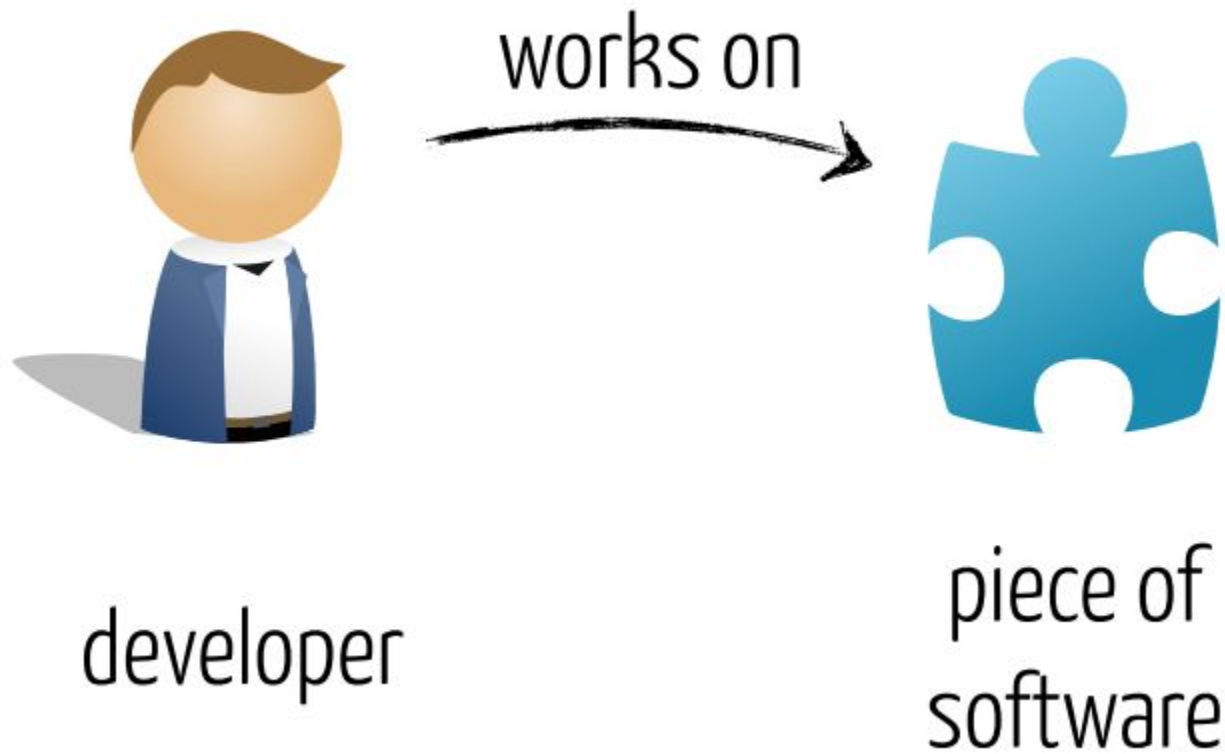
kahoot.it

2



Code versionning?
Why? How?

Why code versioning?

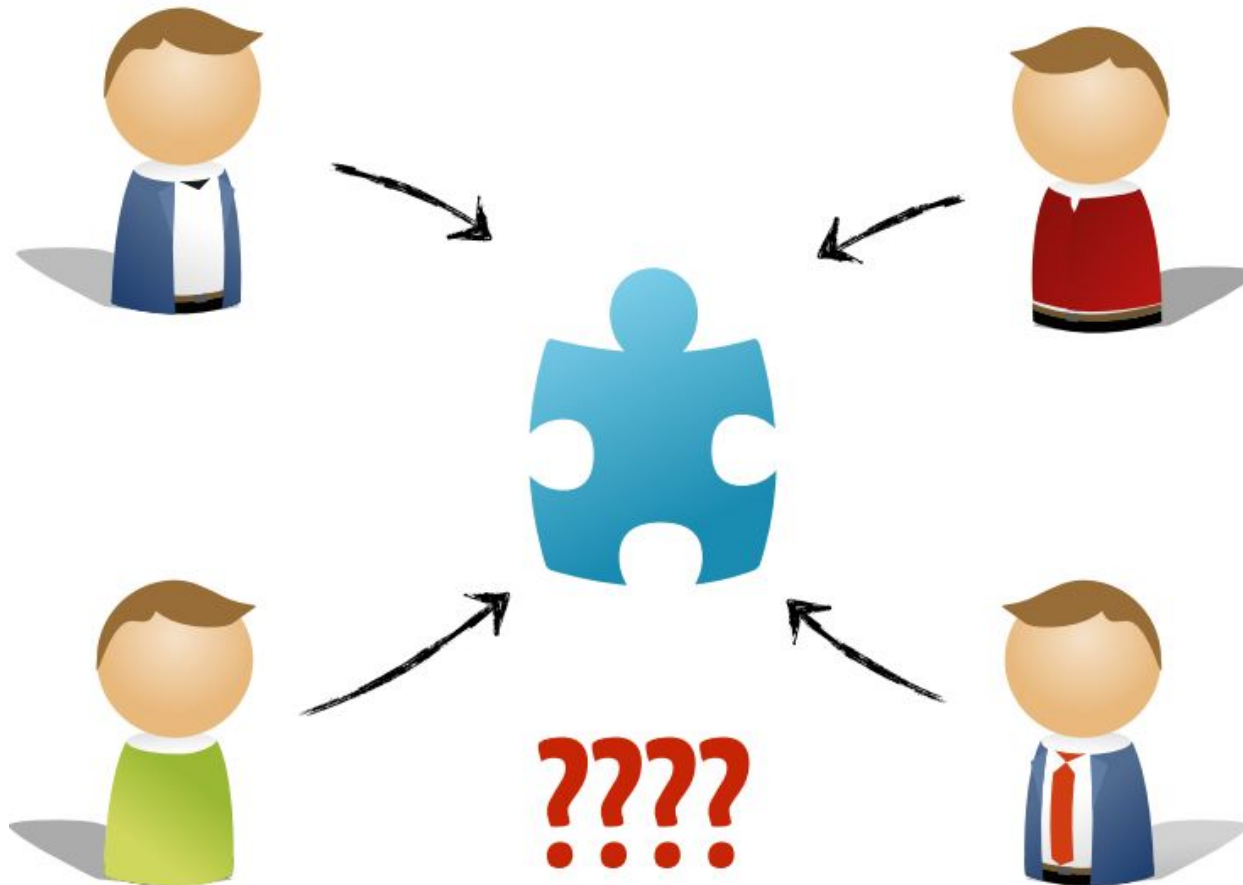


Why code versionning?

From **1** *developer ...*

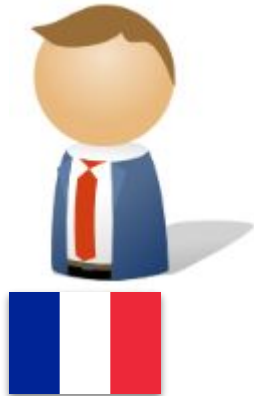
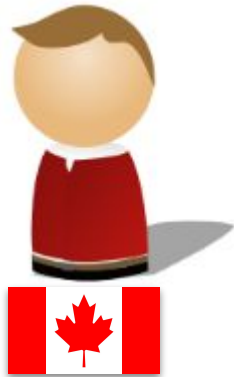
...to **n** *developers*

Why code versionning?



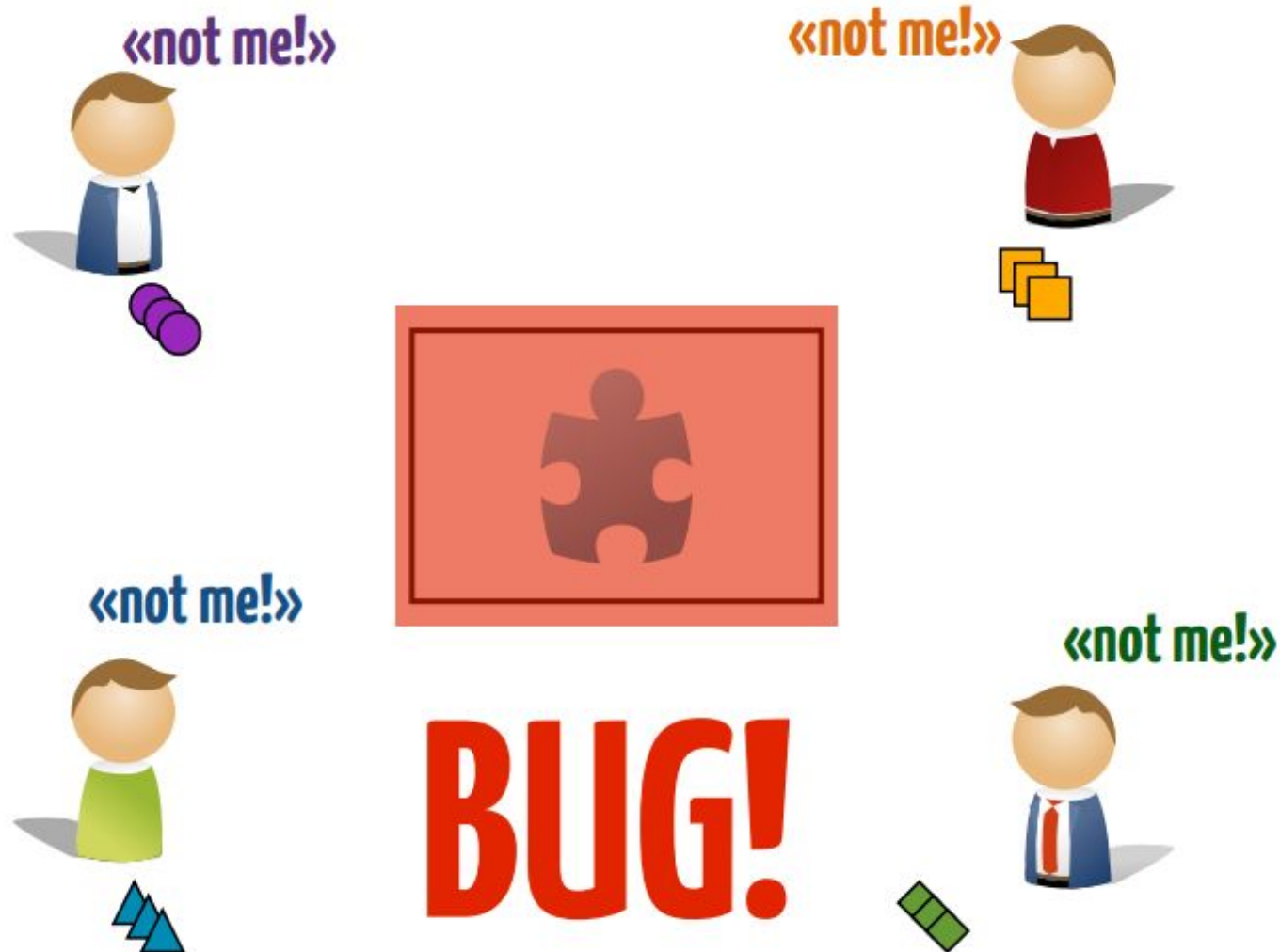
How do you share code in a big teams?

Why code versionning?



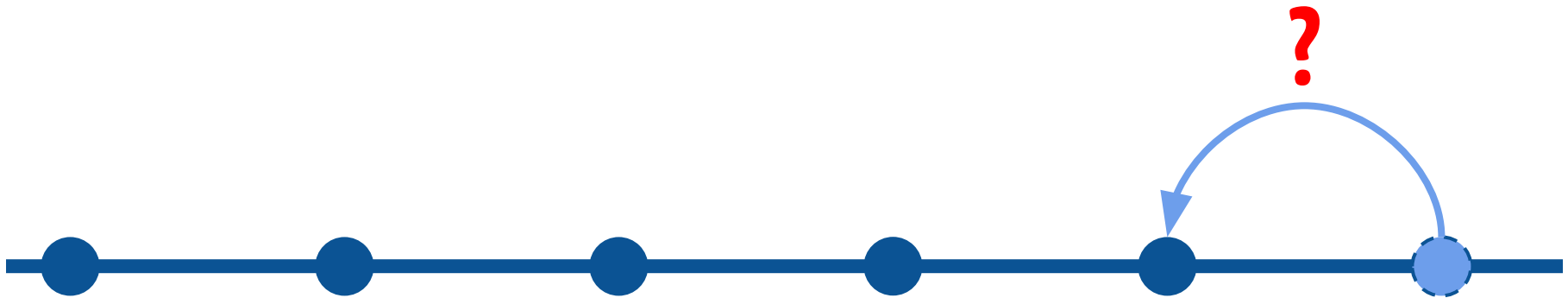
How do you share code in a distributed teams?

Why code versionning?



How do you keep track of your changes?

Why code versionning?



How do you revert a change?

Why code versionning?

We do code versioning to:

- have our code always available **7/24**
- have our code available from **everywhere**
- **keep track** of all the changes
- be able to **rollback** any changes

How? So many providers



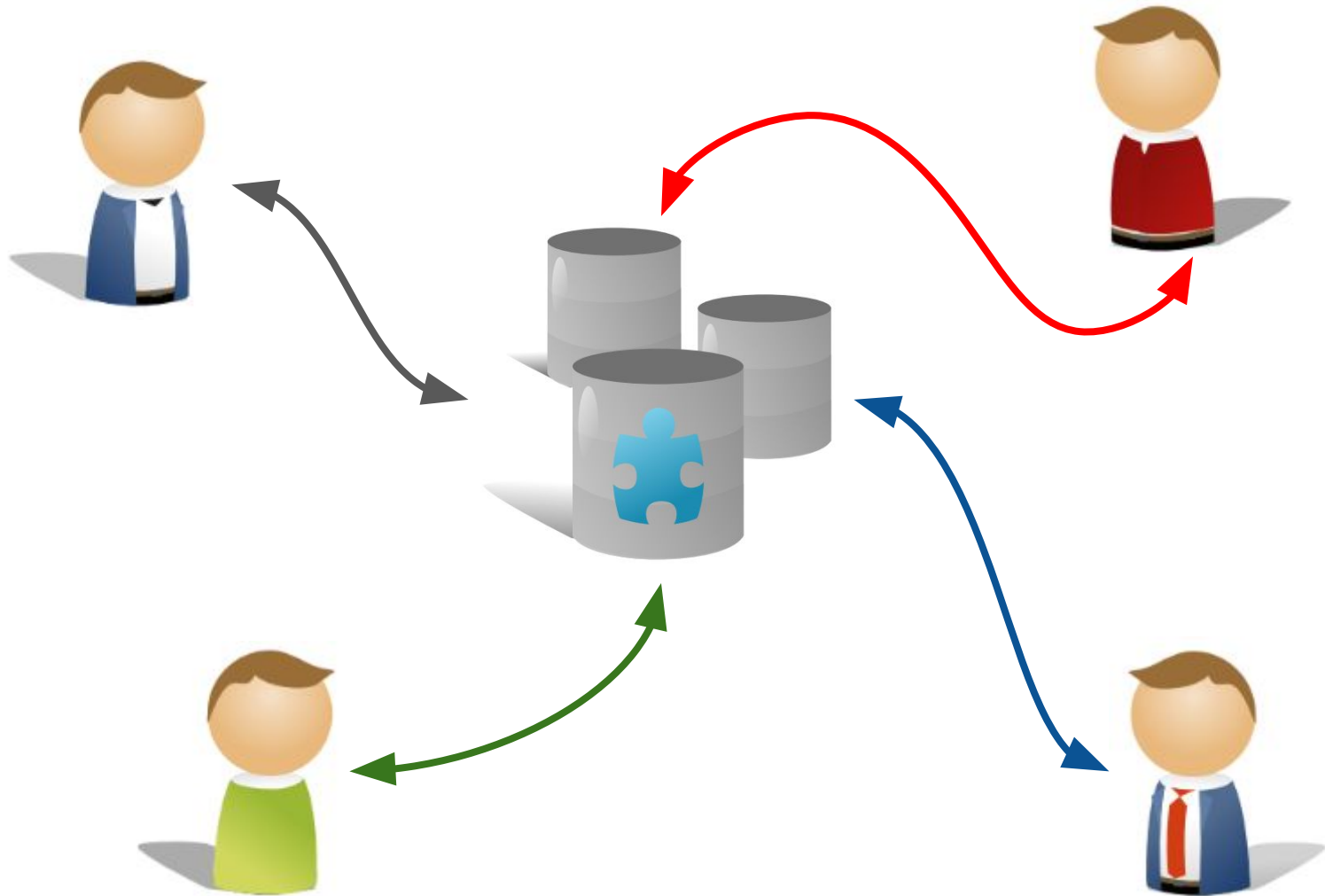
git

Rational ClearCase

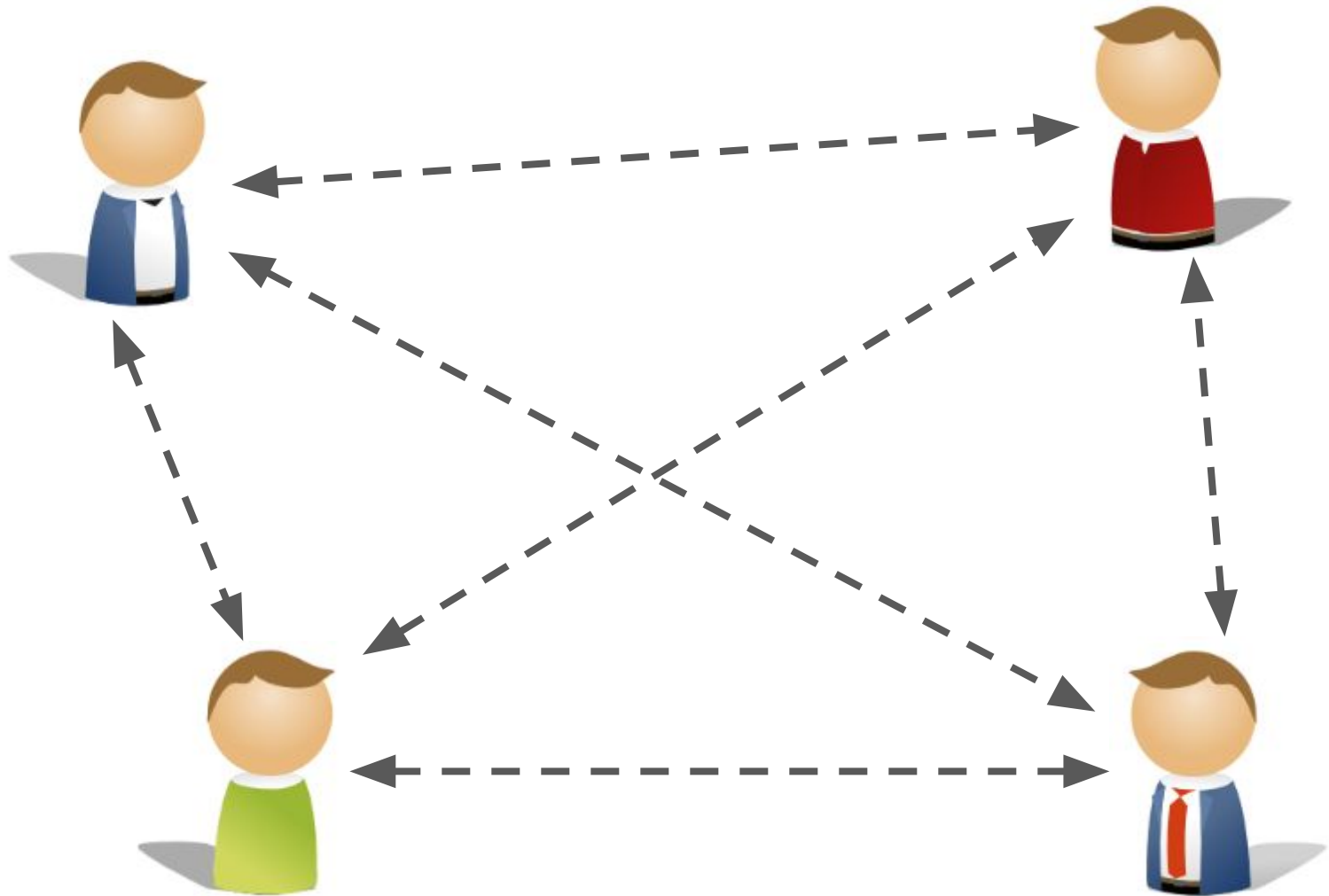


mercurial

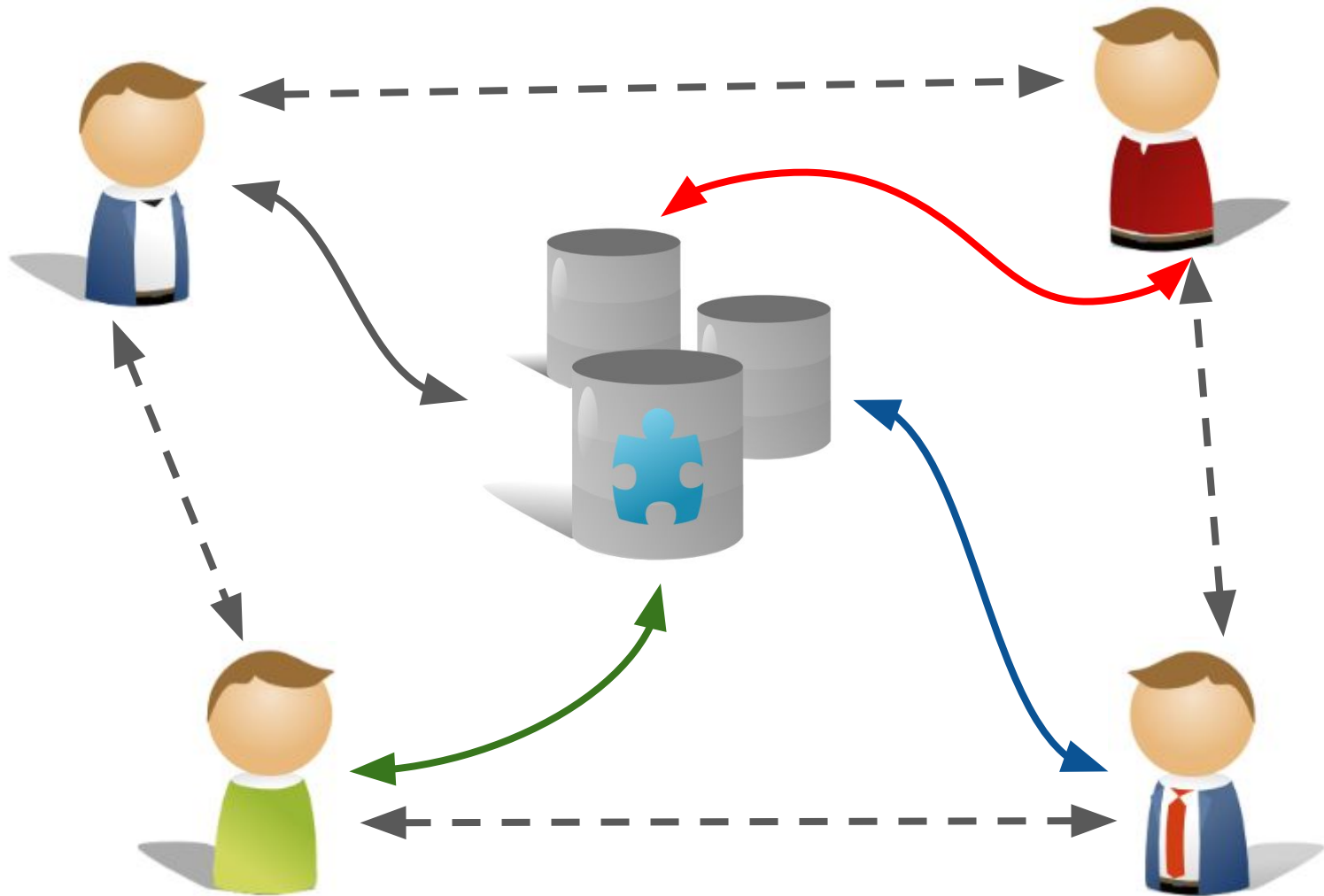
How? Centralized VS distributed



How? Centralized VS distributed



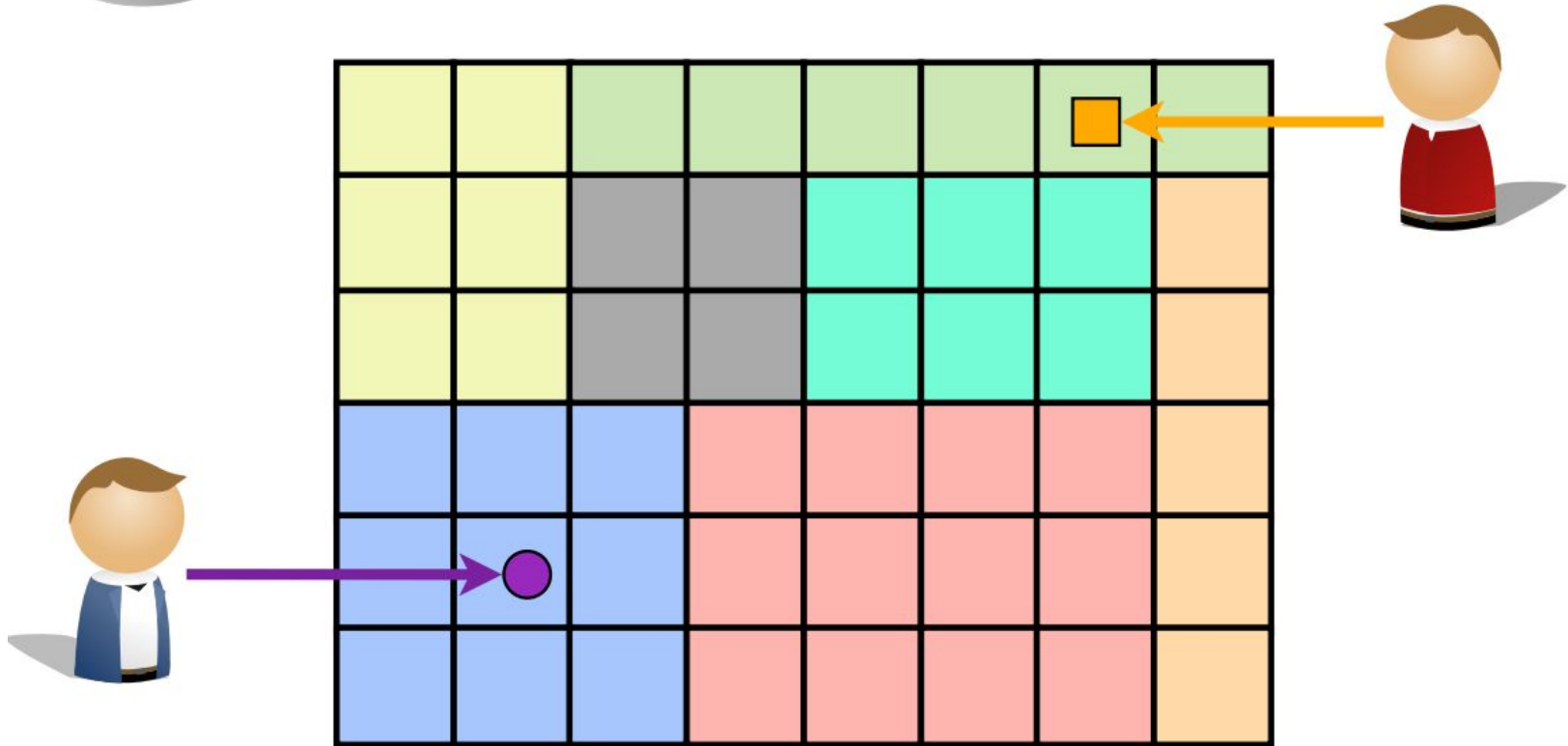
How? Centralized VS distributed



How? Locking VS Merging



#1: different files

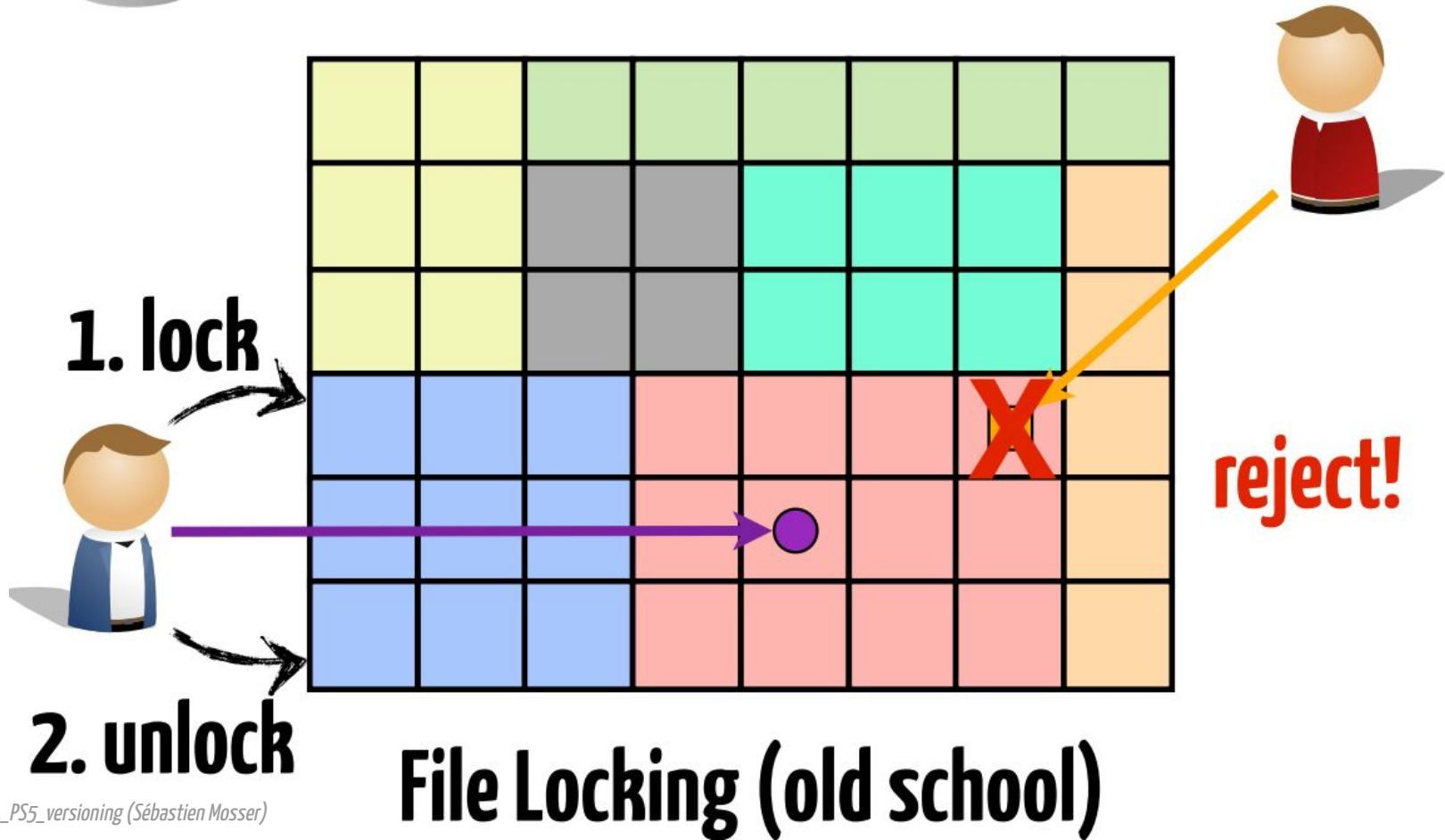


Atomic operations. No problem at all!

How? **Locking** VS Merging



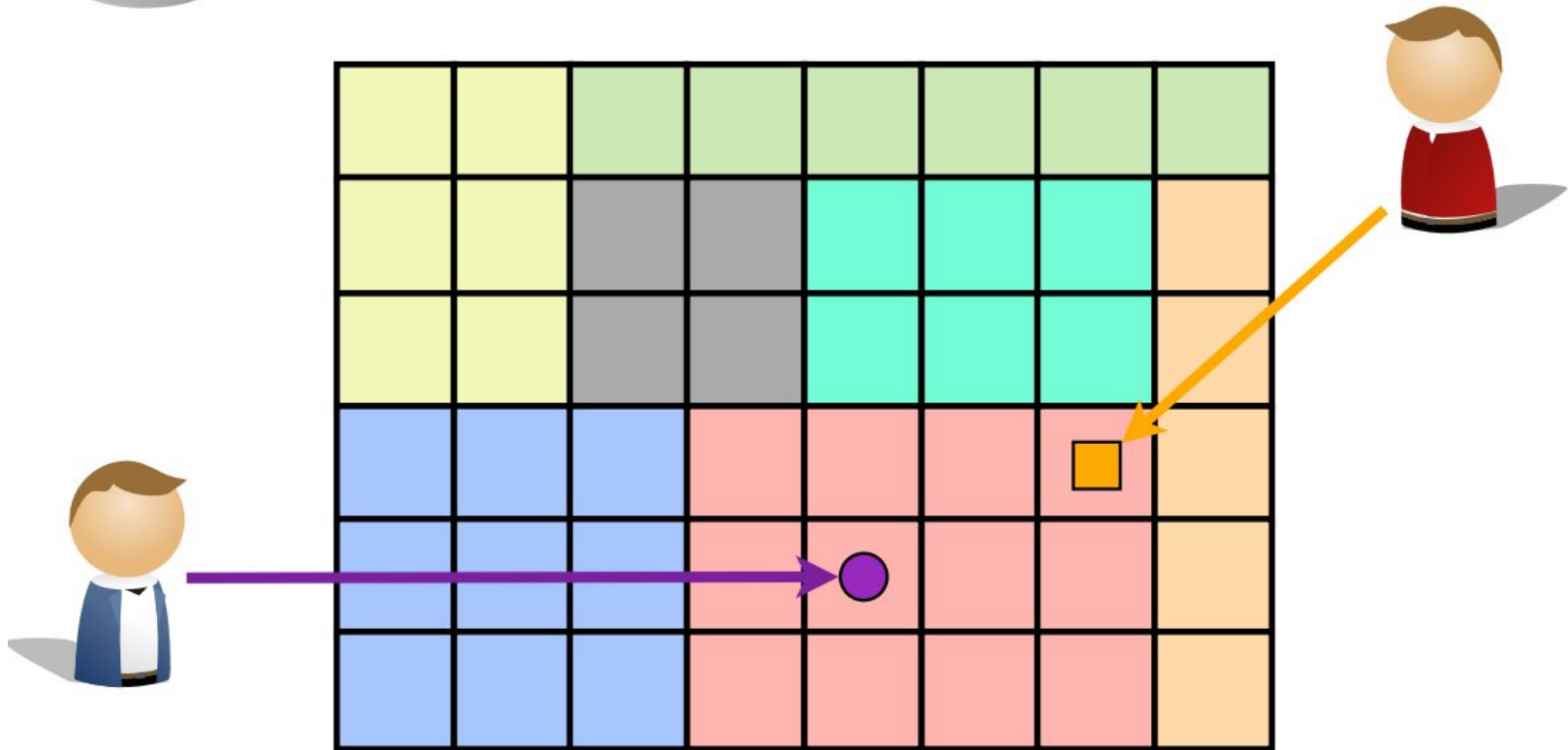
#2: different part of the same file



How? Locking VS Merging



#2: different part of the same file

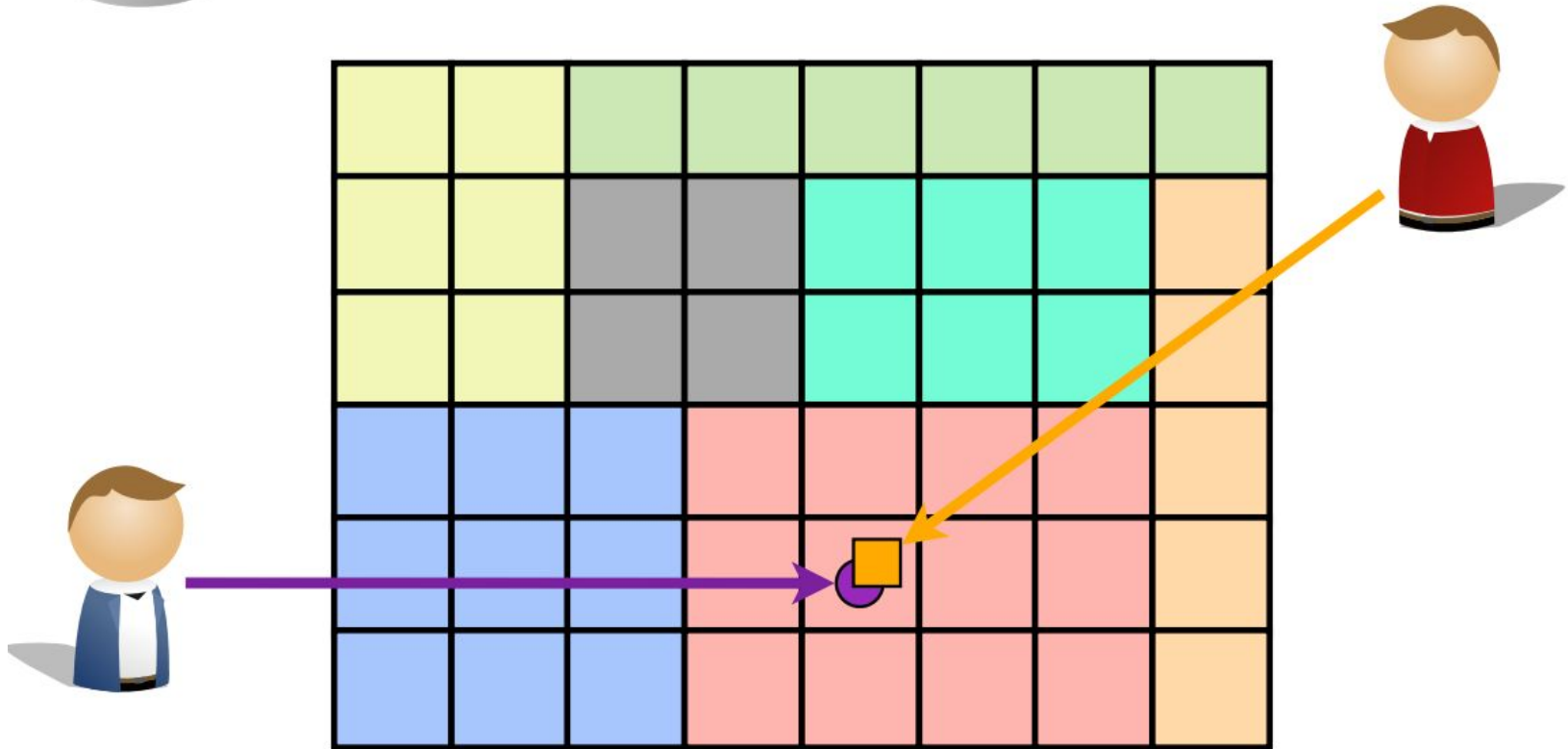


Automatic merge

How? Locking VS Merging



#3: same part of the same file



Conflict!

How? So many providers



git

Rational ClearCase



mercurial

How? Git I choose you !



Distributed & Using merge

Q&A

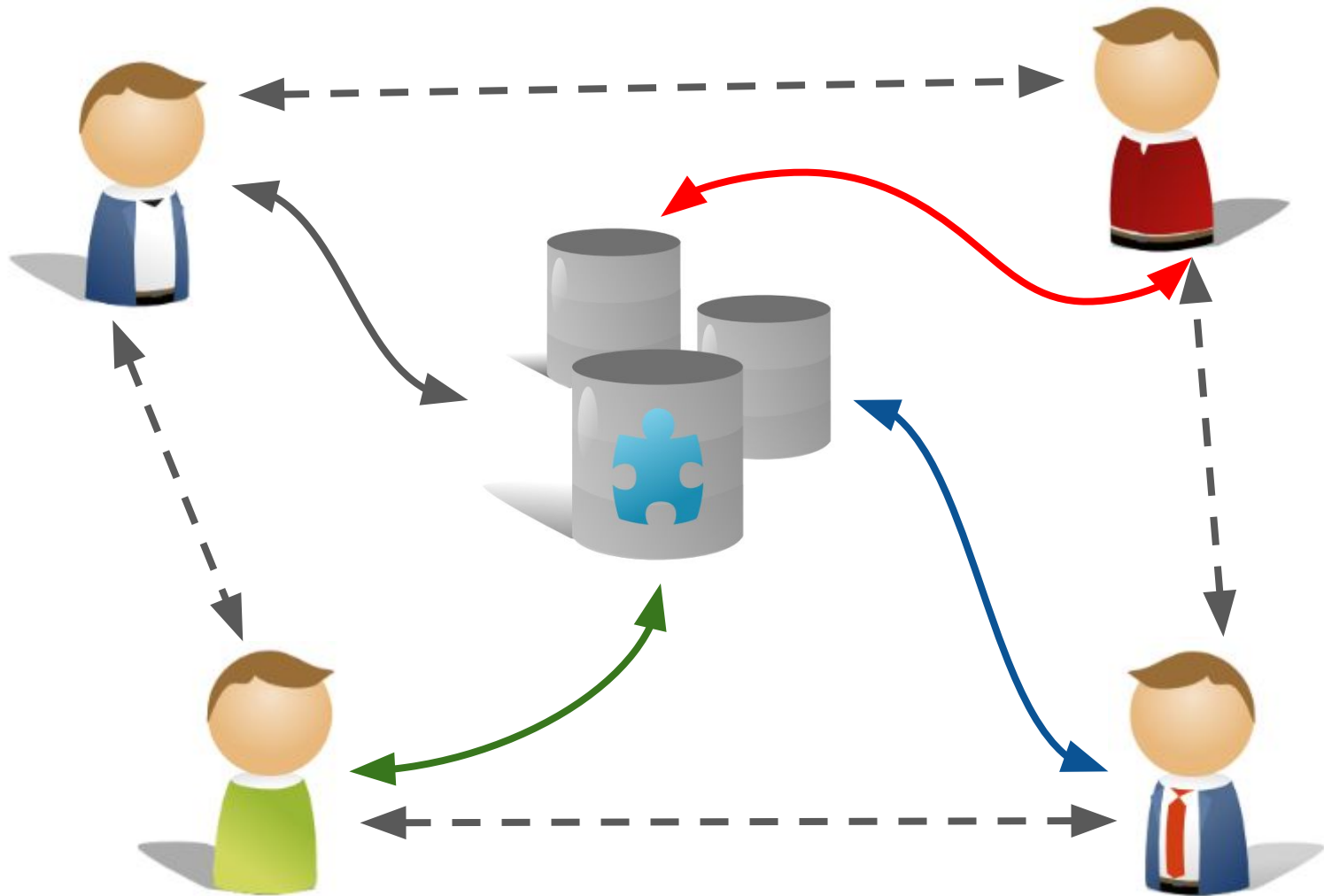
3



git

Introduction to GLT

Git: Repositories



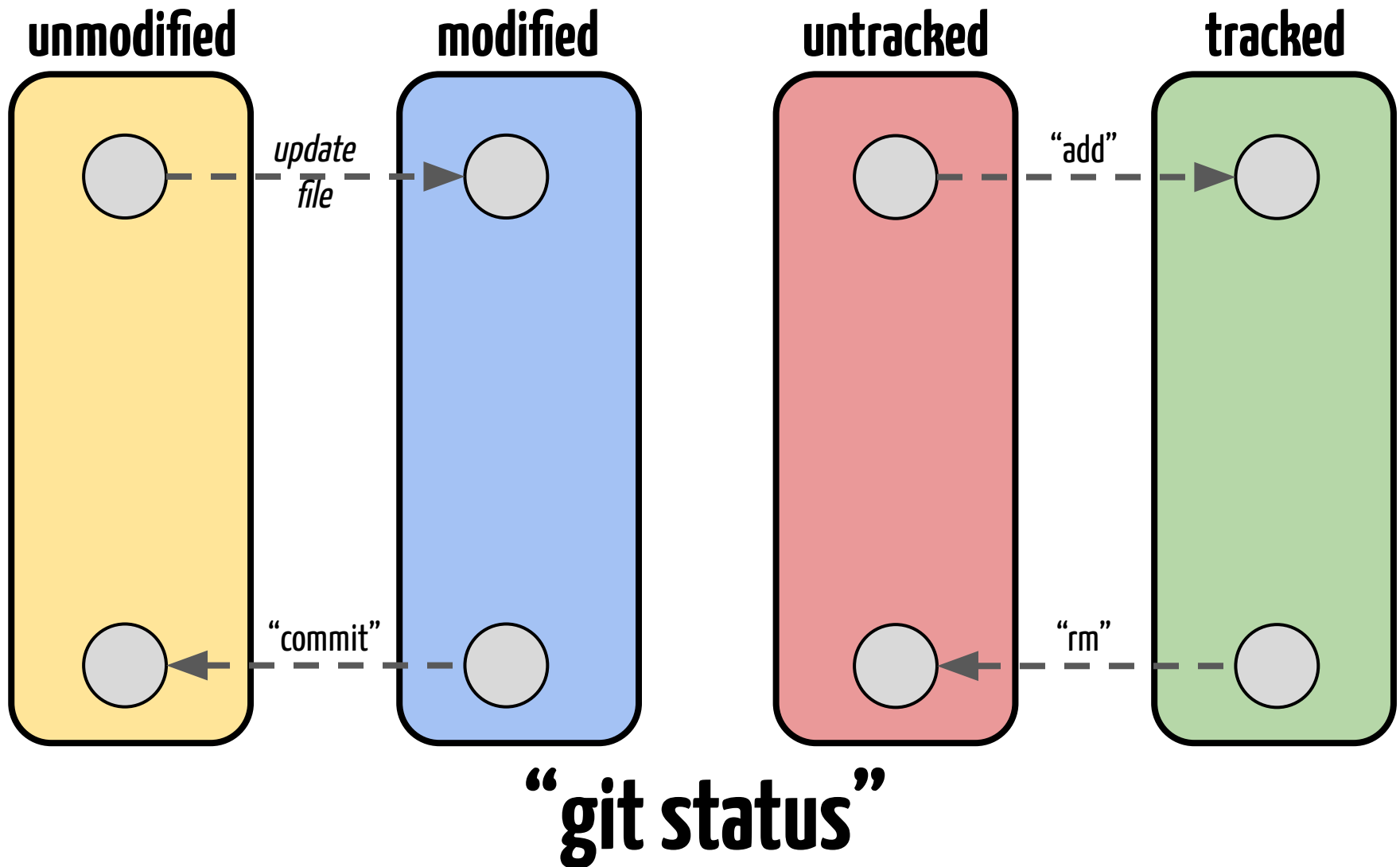
Git: Repositories

“git init”

Git: Repositories

“git clone”

Git: Saving changes



Git: Saving changes

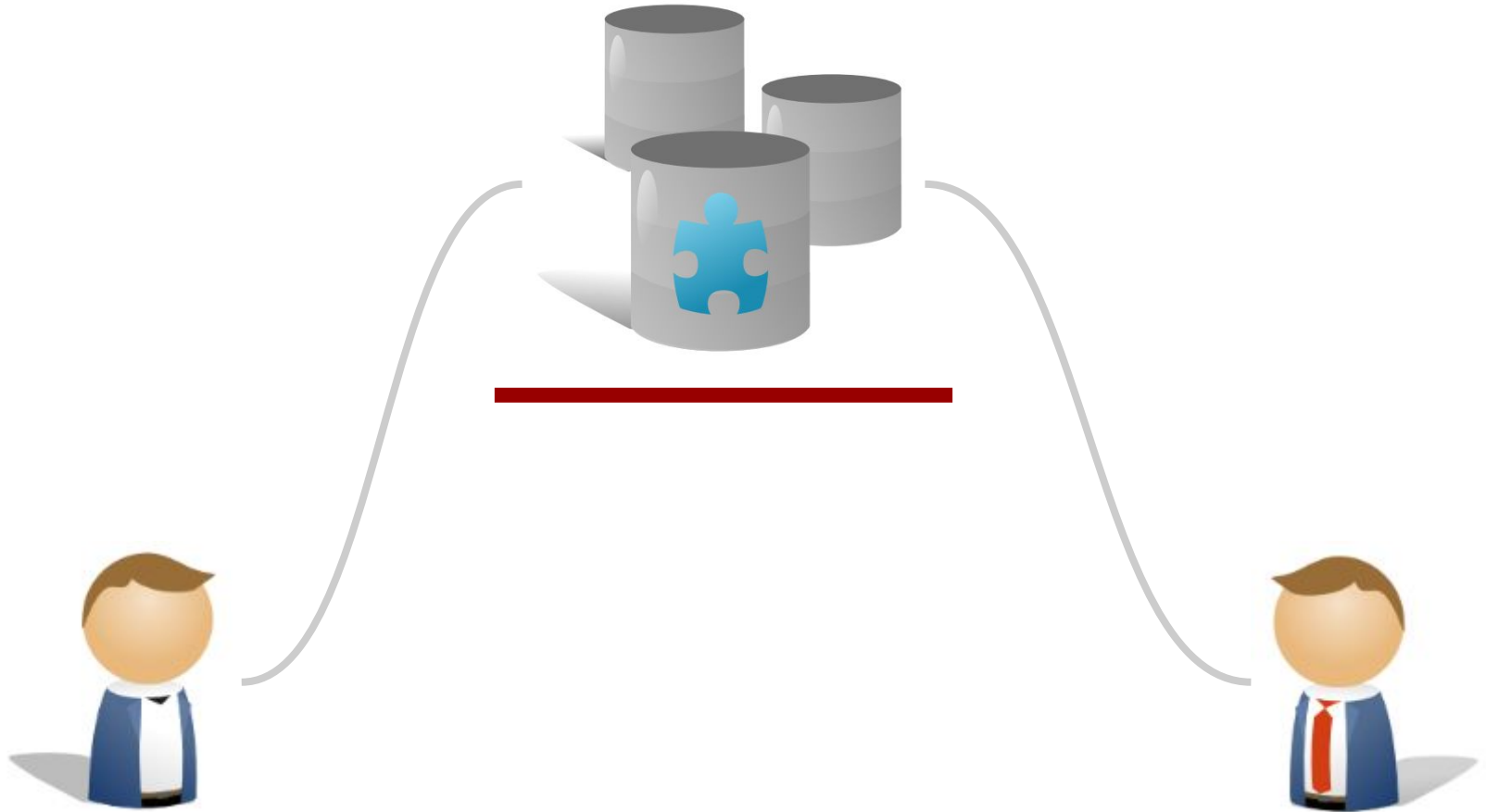
“git commit”

Git: Saving changes

What is saved in a commit?

- A commit message (*`git commit -m "my message"`*)
- Reference to the previous commit
- Some file deletions
- Some file creations
- Some file content changes

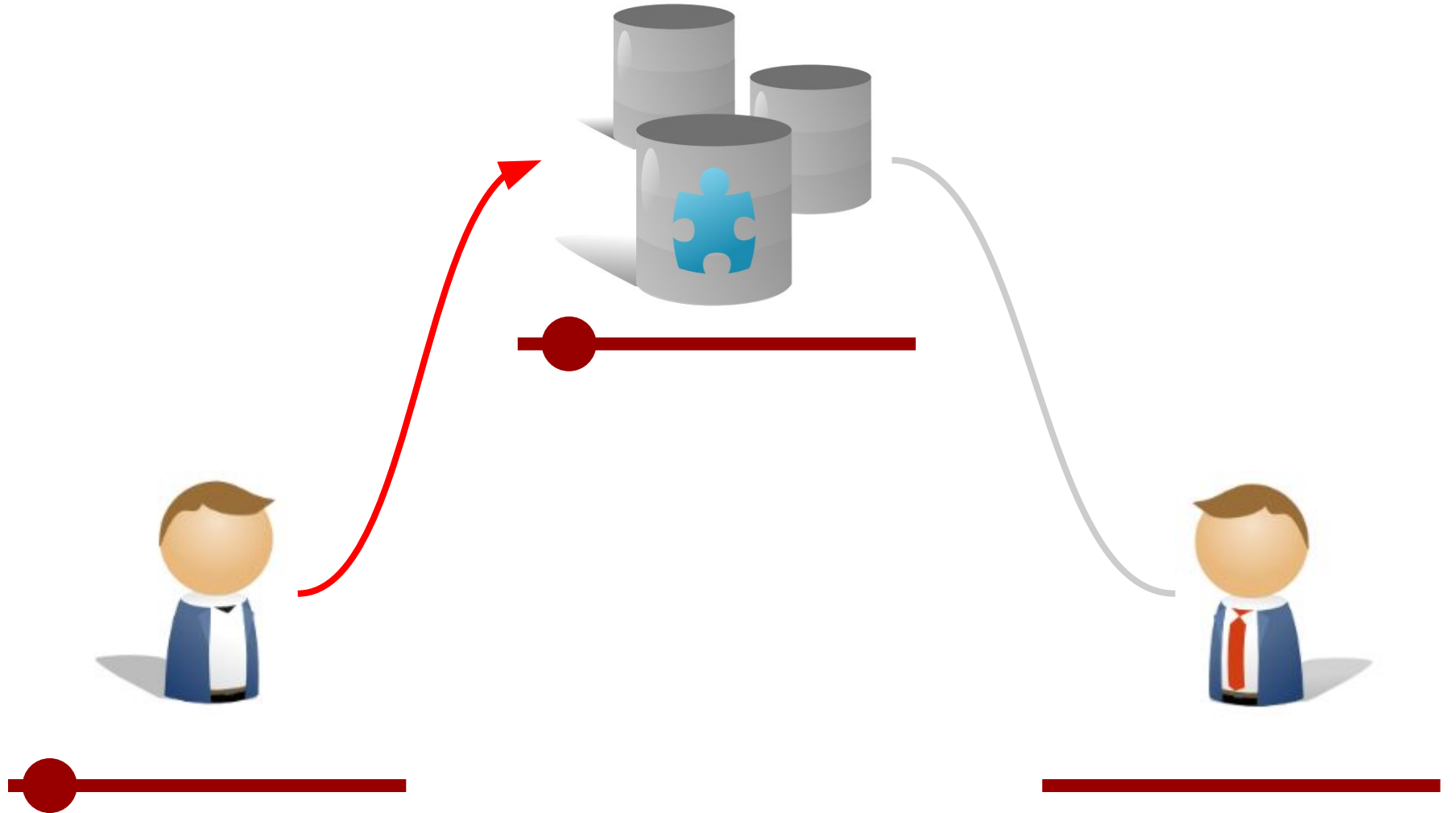
Git: push & pull



Git: push & pull

“git push”

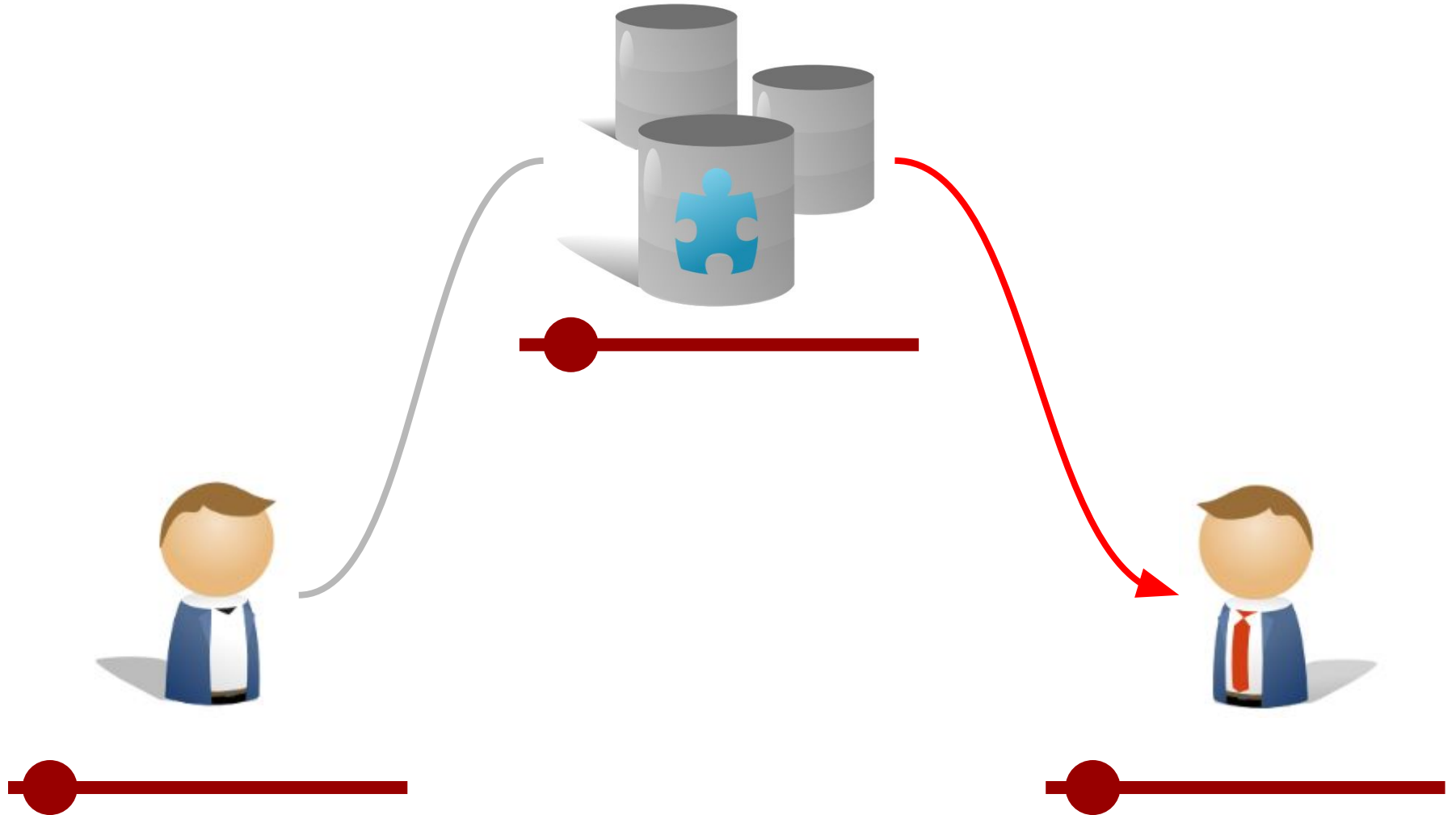
Git: push & pull



Git: push & pull

“git pull”

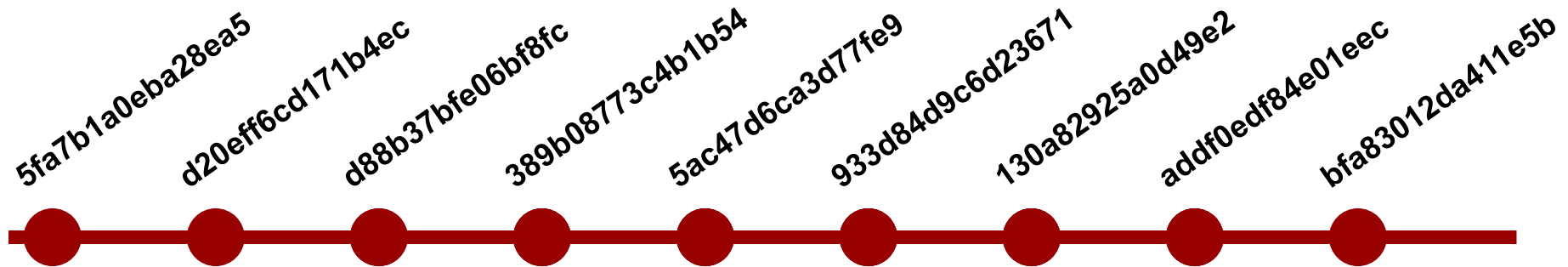
Git: push & pull



Git: tags

An alias to a commit

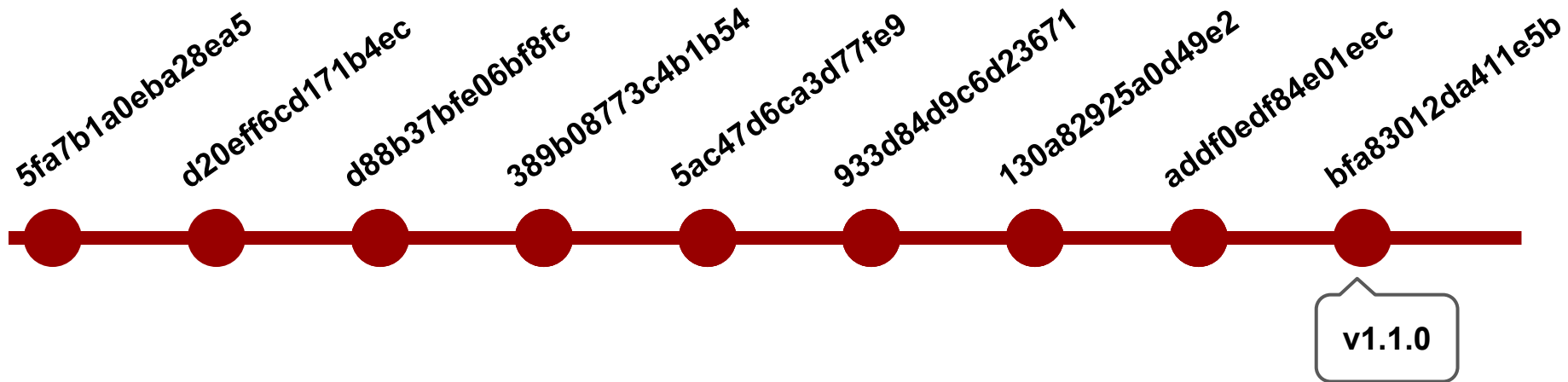
Git: tags



Git: tags

“git tag -a v1.1.0”

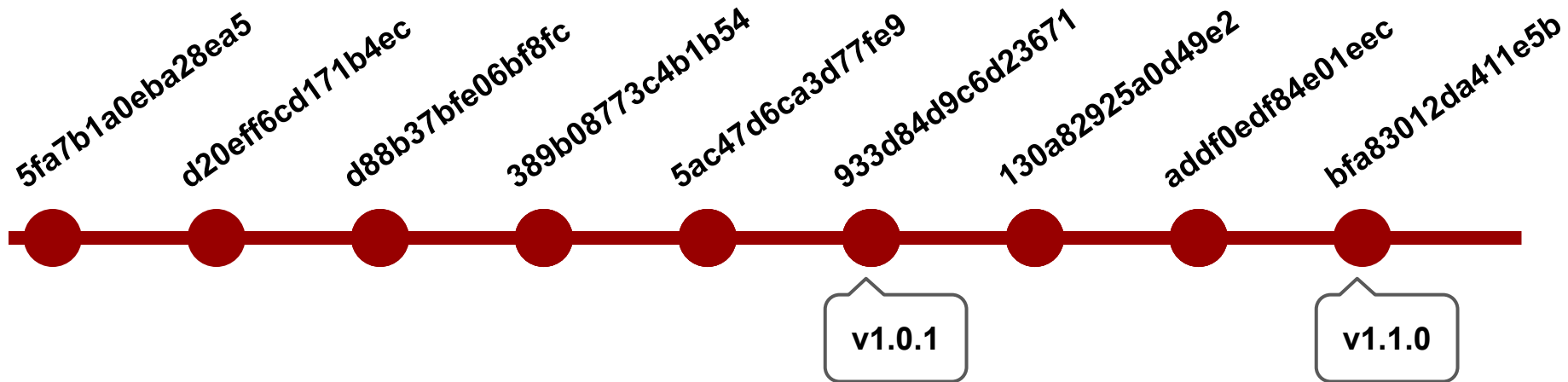
Git: tags



Git: tags

“git tag -a v1.0.1 933d84d9c6d23671”

Git: tags



Git: Do not forget!

“git clone <path to repository>”

“git add <path to files to track>”

“git commit -m “<a commit message>””

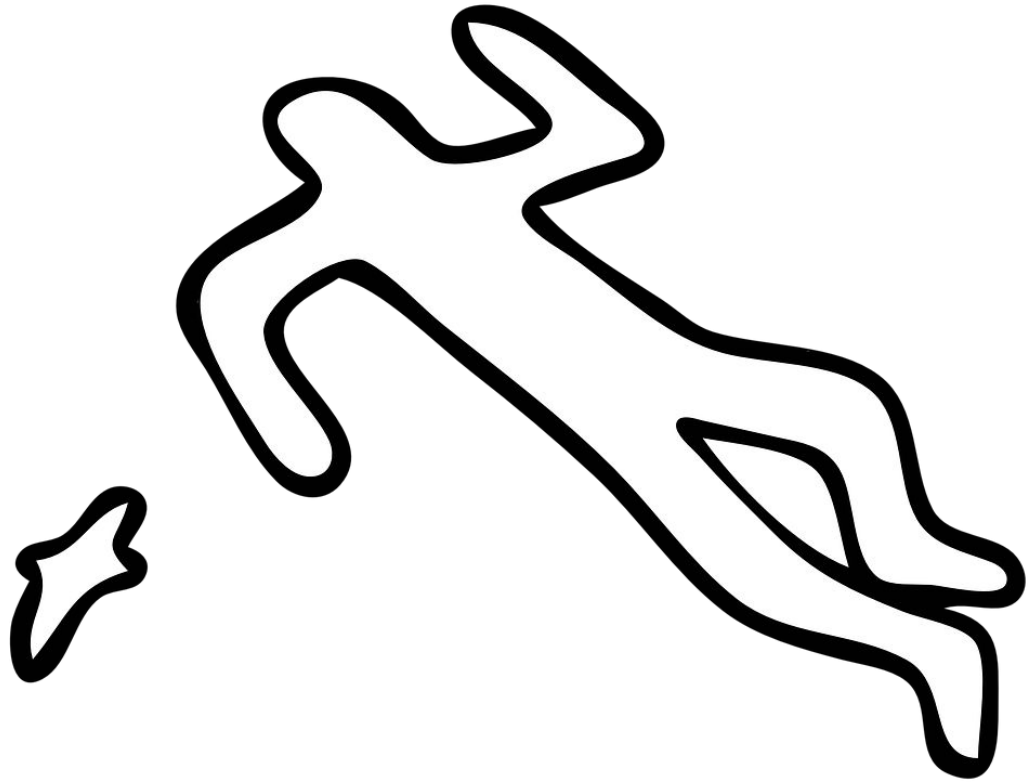
“git pull”

“git push”

“git tag -a <a tag alias> -m “<a tag message>””

Q&A

4



Case study

Case study: the Yahtzee project



2 developers



Alice & Bob

3 tasks

Create a Dice

Read dice from input


Write dice to output






Using Github



Case study: create the repository



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



New repository

Import repository

New gist

New organization


New project


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner


Repository name *

 mathiascouste ▾


/ Yahtzee 

Great repository names are short and memorable. Need inspiration? How about **cautious-octo-goggles**?

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**


You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **Java** ▾

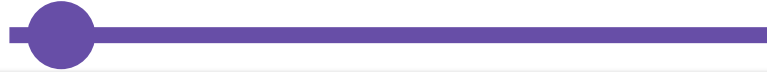
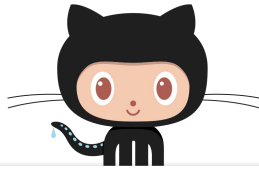
Add a license: **None** ▾ 

Create repository

Case study: create the repository



Case study: clone the repository



[Create new file](#) [Upload files](#) [Find File](#) [Clone or download ▾](#)

initial commit

initial commit

Clone with HTTPS ? [Use SSH](#)

Use Git or checkout with SVN using the web URL.

`https://github.com/mathiascouste/Yahtzee`

[Open in Desktop](#) [Download ZIP](#)



Case study: clone the repository



```
$alice > git clone https://github.com/alice/Yahtzee.git
```

```
$bob > git clone https://github.com/alice/Yahtzee.git
```



Case study: clone the repository



Case study: Create a Dice (code)



Dice.java:

```
public class Dice {  
    int value;  
  
    Dice(int value) {  
        this.value = value;  
    }  
}
```



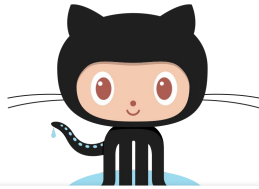
Case study: Create a Dice (code)



Dice.java



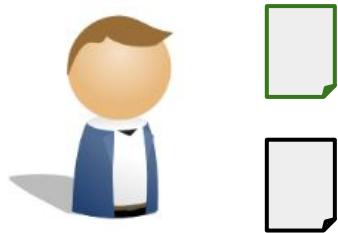
Case study: Create a Dice (add)



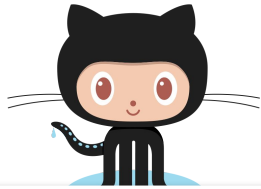
```
$alice > git add Dice.java
```



Case study: Create a Dice (add)



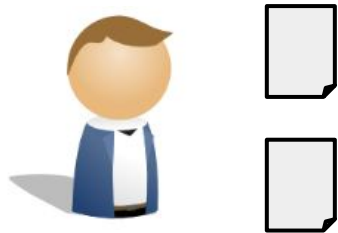
Case study: Create a Dice (commit)



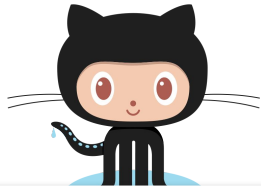
```
$alice > git commit -m "Dice creation"
```



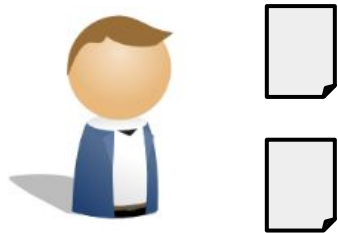
Case study: Create a Dice (commit)



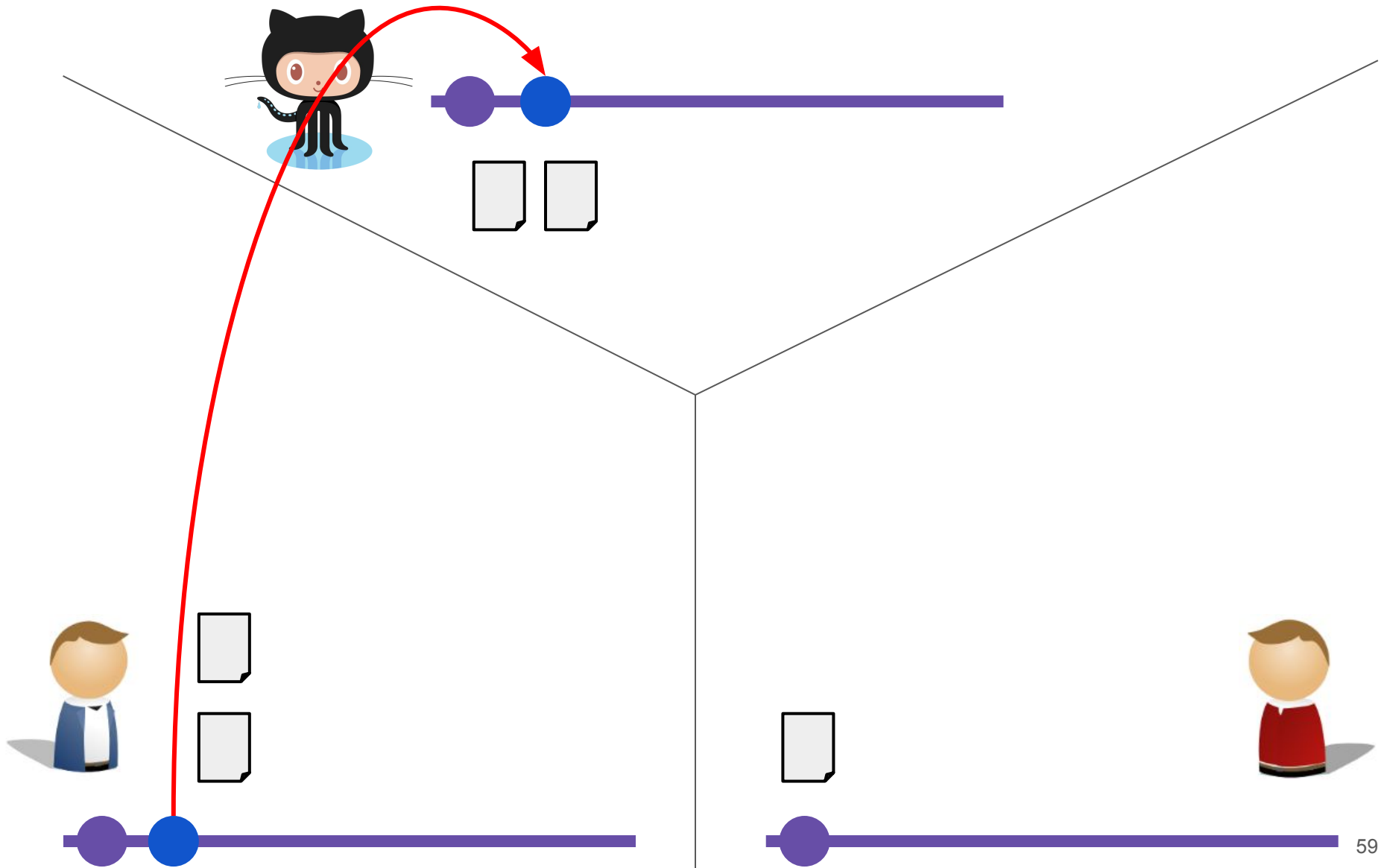
Case study: Create a Dice (push)



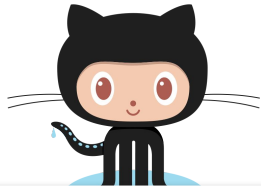
```
$alice > git push
```



Case study: Create a Dice (push)



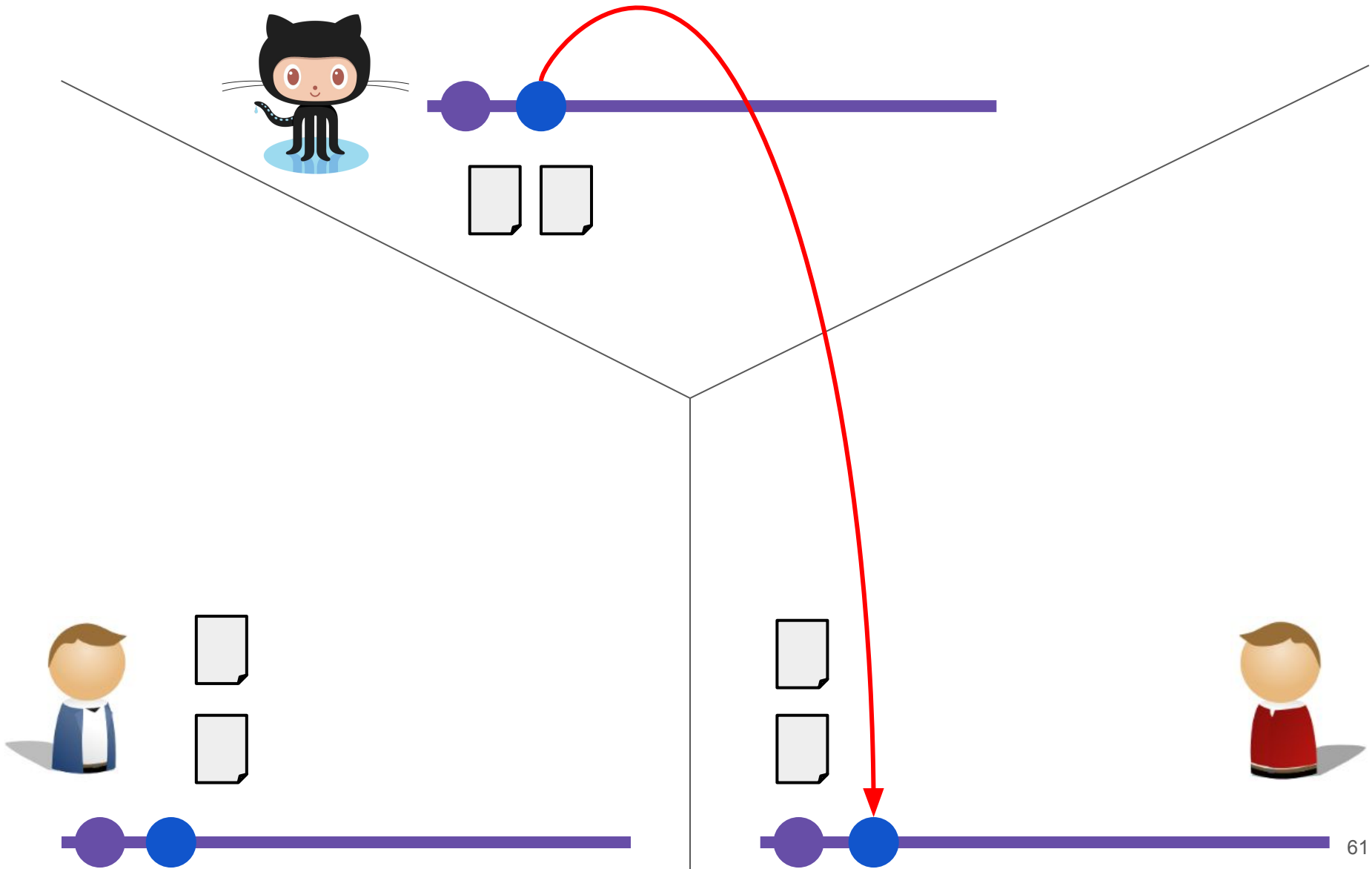
Case study: Read Dice (pull)



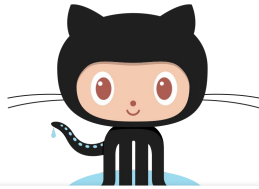
```
$bob > git pull
```



Case study: Read Dice (pull)



Case study: Read Dice (code)

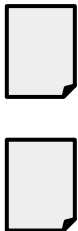
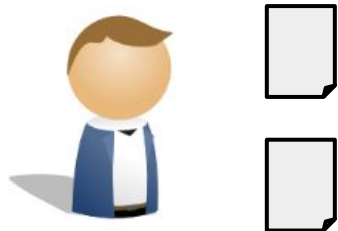
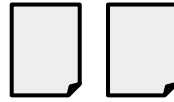


Dice.java:

```
public class Dice {  
    int value;  
  
    Dice(int value) {  
        this.value = value;  
    }  
  
    static Dice from(int val) {  
        return new Dice(val);  
    }  
}
```



Case study: Read Dice (code)



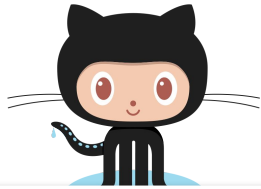
DiceReader.java



Dice.java



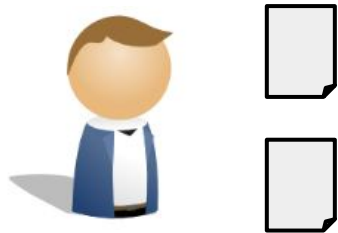
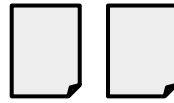
Case study: Read Dice (add & commit)



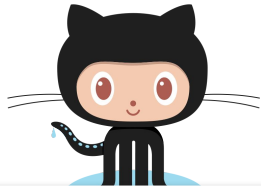
```
$bob > git add .  
$bob > git commit -m "Adding dice reader"
```



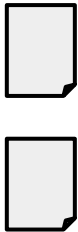
Case study: Read Dice (add & commit)



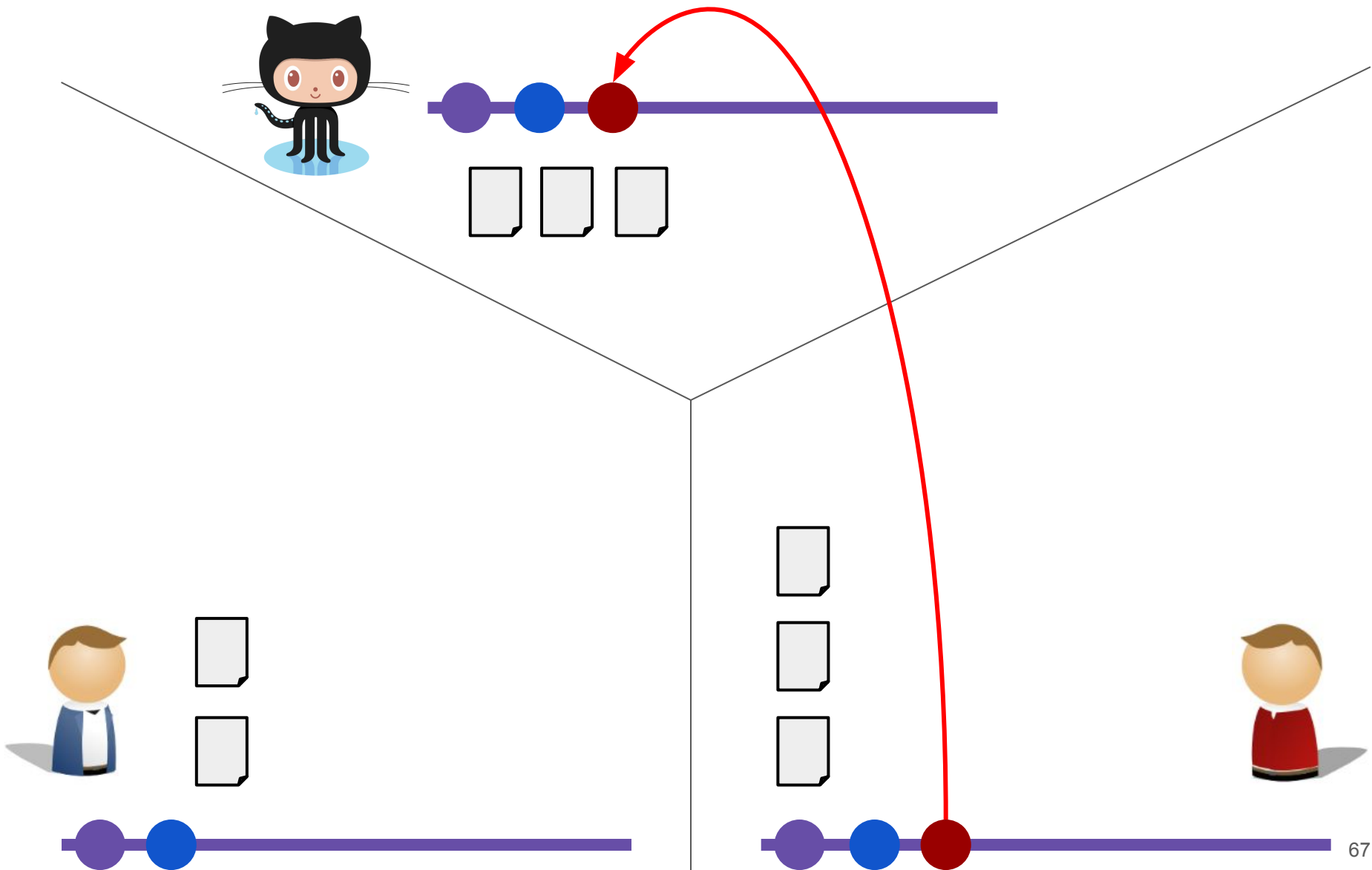
Case study: Read Dice (push)



```
$bob > git push
```



Case study: Read Dice (push)

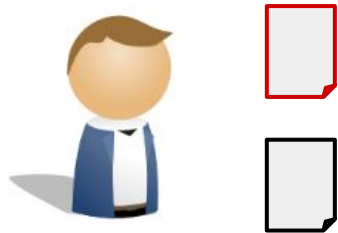


Case study: Write Dice (code)

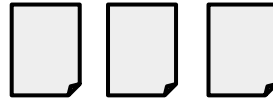


Dice.java:

```
public class Dice {  
    int value;  
  
    Dice(int value) {  
        this.value = value;  
    }  
    String toString() {  
        return "D" + this.value;  
    }  
}
```



Case study: Write Dice (code)



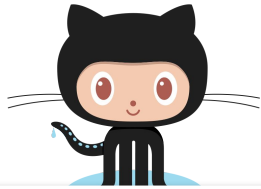
DiceWriter.java



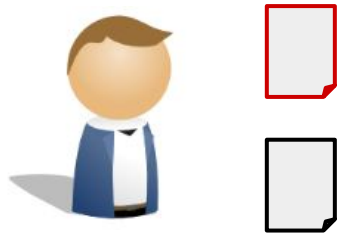
Dice.java



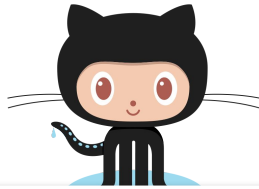
Case study: Write Dice (pull)



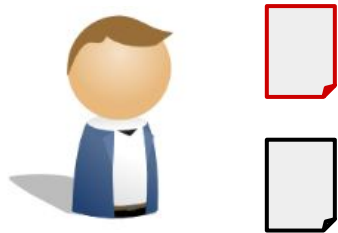
```
$alice > git pull
```



Case study: Write Dice (pull)



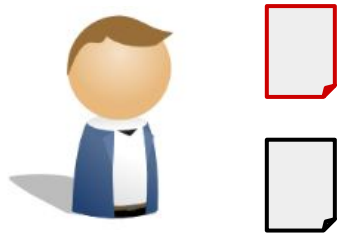
```
$alice > git pull
error: The following untracked working tree files would be overwritten by merge
       Dice.java
Please move or remove them before you merge.
Aborting
```



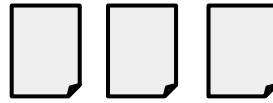
Case study: Write Dice (add & commit)



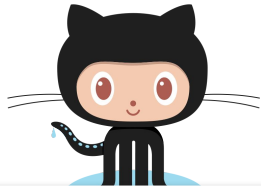
```
$alice > git add .  
$alice > git commit -m "Adding Dice writer"
```



Case study: Write Dice (add & commit)



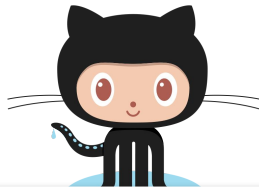
Case study: Write Dice (pull)



```
$alice > git pull
```



Case study: Write Dice (conflict resolution)



```
$alice > git pull
```

CONFLICT (content): Merge conflict in Dice.java
Automatic merge failed; fix conflicts and then commit the result.



Case study: Write Dice (conflict resolution)



Case study: Write Dice (conflict resolution)



bd82b81f6801276:

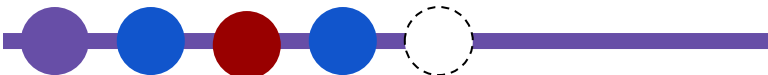
```
public class Dice {  
    int value;  
  
    Dice(int value) {  
        this.value = value;  
    }  
  
    static Dice from(int val) {  
        return new Dice(val);  
    }  
}
```

local:

```
public class Dice {  
    int value;  
  
    Dice(int value) {  
        this.value = value;  
    }  
    <<<<<<< bd82b81f6801276  
    static Dice from(int val) {  
        return new Dice(val);  
    }  
    =====  
    String toString() {  
        return "D" + this.value;  
    }  
    >>>>>>> fc225232e58ed9e  
}
```

fc225232e58ed9e:

```
public class Dice {  
    int value;  
  
    Dice(int value) {  
        this.value = value;  
    }  
  
    String toString() {  
        return "D" + this.value;  
    }  
}
```



Case study: Write Dice (conflict resolution)



bd82b81f6801276:

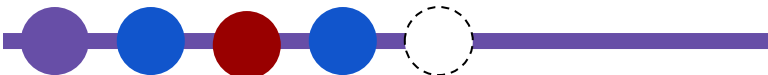
```
public class Dice {  
    int value;  
  
    Dice(int value) {  
        this.value = value;  
    }  
  
    static Dice from(int val) {  
        return new Dice(val);  
    }  
}
```

local:

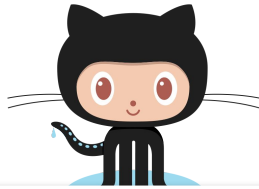
```
public class Dice {  
    int value;  
  
    Dice(int value) {  
        this.value = value;  
    }  
  
    static Dice from(int val) {  
        return new Dice(val);  
    }  
  
    String toString() {  
        return "D" + this.value;  
    }  
}
```

fc225232e58ed9e:

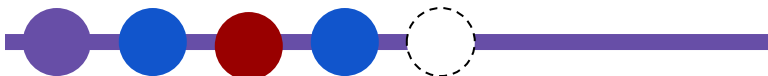
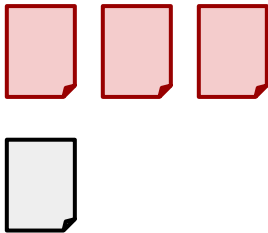
```
public class Dice {  
    int value;  
  
    Dice(int value) {  
        this.value = value;  
    }  
  
    String toString() {  
        return "D" + this.value;  
    }  
}
```



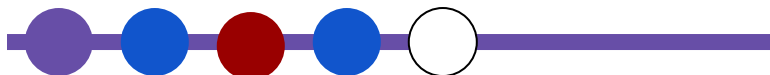
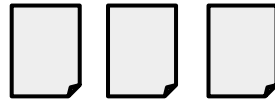
Case study: Write Dice (conflict resolution)



```
$alice > git add .  
$alice > git commit
```



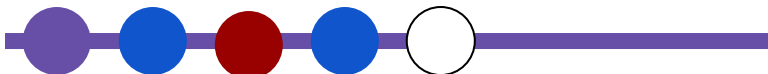
Case study: Write Dice (conflict resolution)



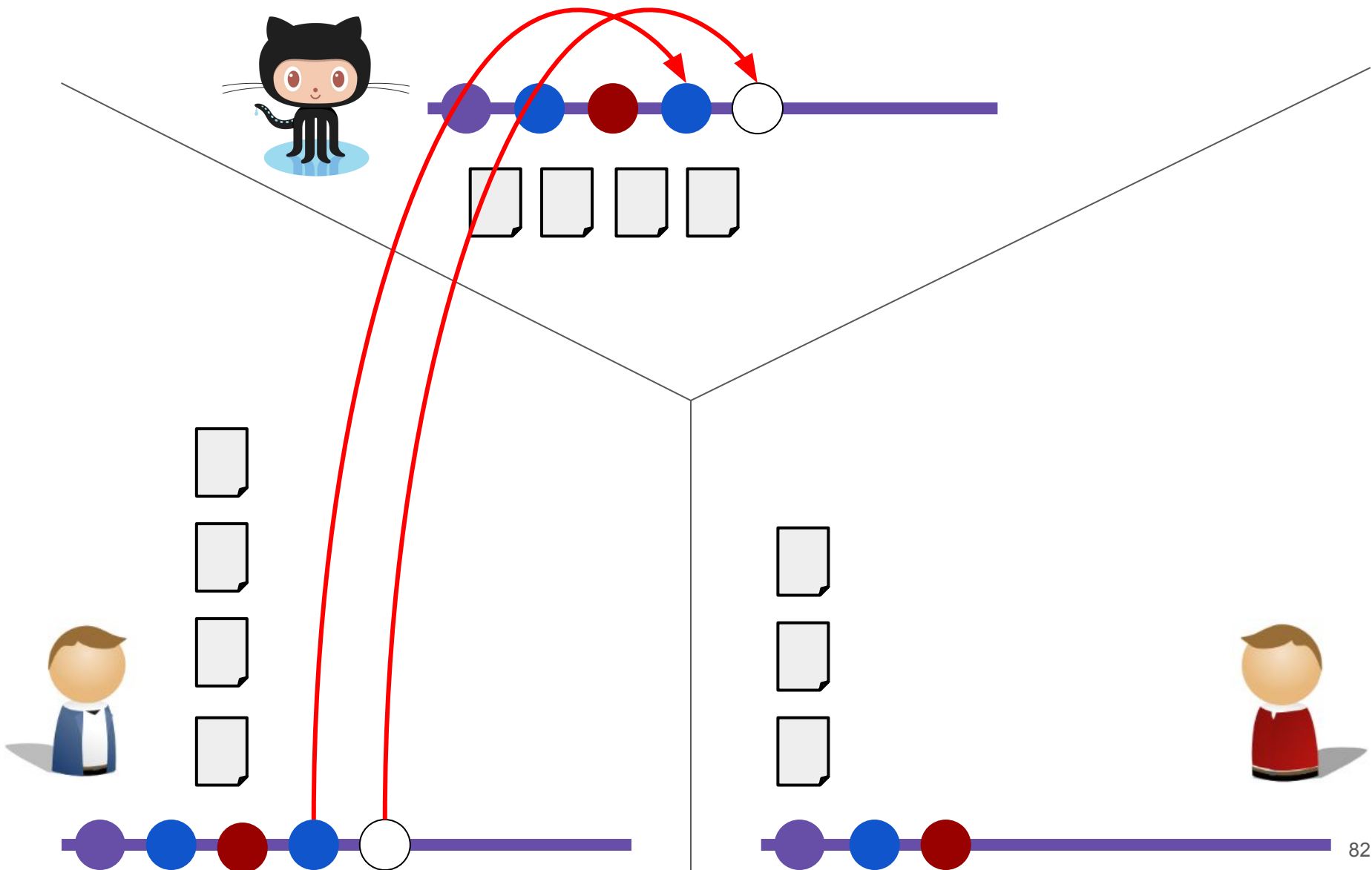
Case study: Write Dice (push)



```
$alice > git push
```



Case study: Write Dice (push)



Q&A

It's dangerous to go alone! Take this.

learngitbranchingstrategy.js.org





Next 3 hours!