Collaborative Development & Source Code Versioning

Sébastien Mosser
(modifié par M. Blay-Fornarino et des cours de M. Pallez)

credits: C.Line design & sxc.hu

Objectifs de ce cours

✓ Que vous compreniez l'intérêt d'un gestionnaire de version.

✓ Que vous aillez une vue un peu plus concrète de la gestion de projet outillé

➡ En TD :

- Au retour des vacances, vous apprenez à utiliser le gestionnaire de versions connecté à la forge.
- Dans la suite des TDs, vous utilisez la gestion de version

➡ Ensuite vous avez votre propre environnement pour gérer vos projets et vous vous auto-organizez.

MBF

Problématique

- Cas : travail multi-postes
  - Vous développez un projet logiciel
    - Dans la journée, sur des stations de travail de l'IUT
    - Le soir, sur votre ordinateur personnel
  - Comment transférez les modifications de l'un à l'autre ?

Rôles d’un gestionnaire de versions

- **Fonction**
  - **Faciliter le travail en multipostes en permettant l'accès à distance à un dépôt partagé par tous les postes**

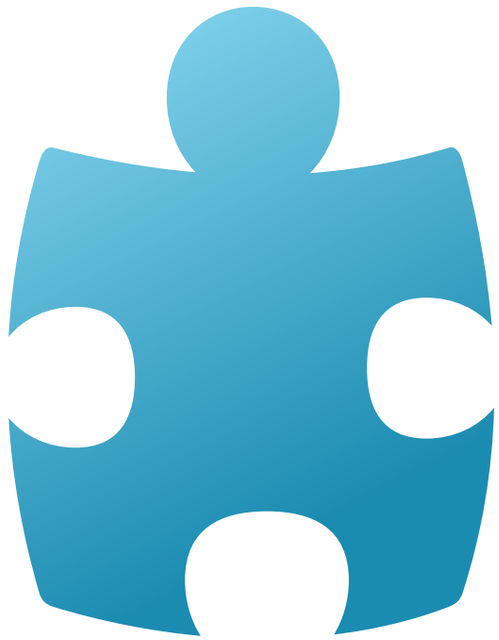
https://momastery.com/blog/2012/05/09/van-tastic-mothers-day-love-flash-mob/

« Collaborative Development ?»

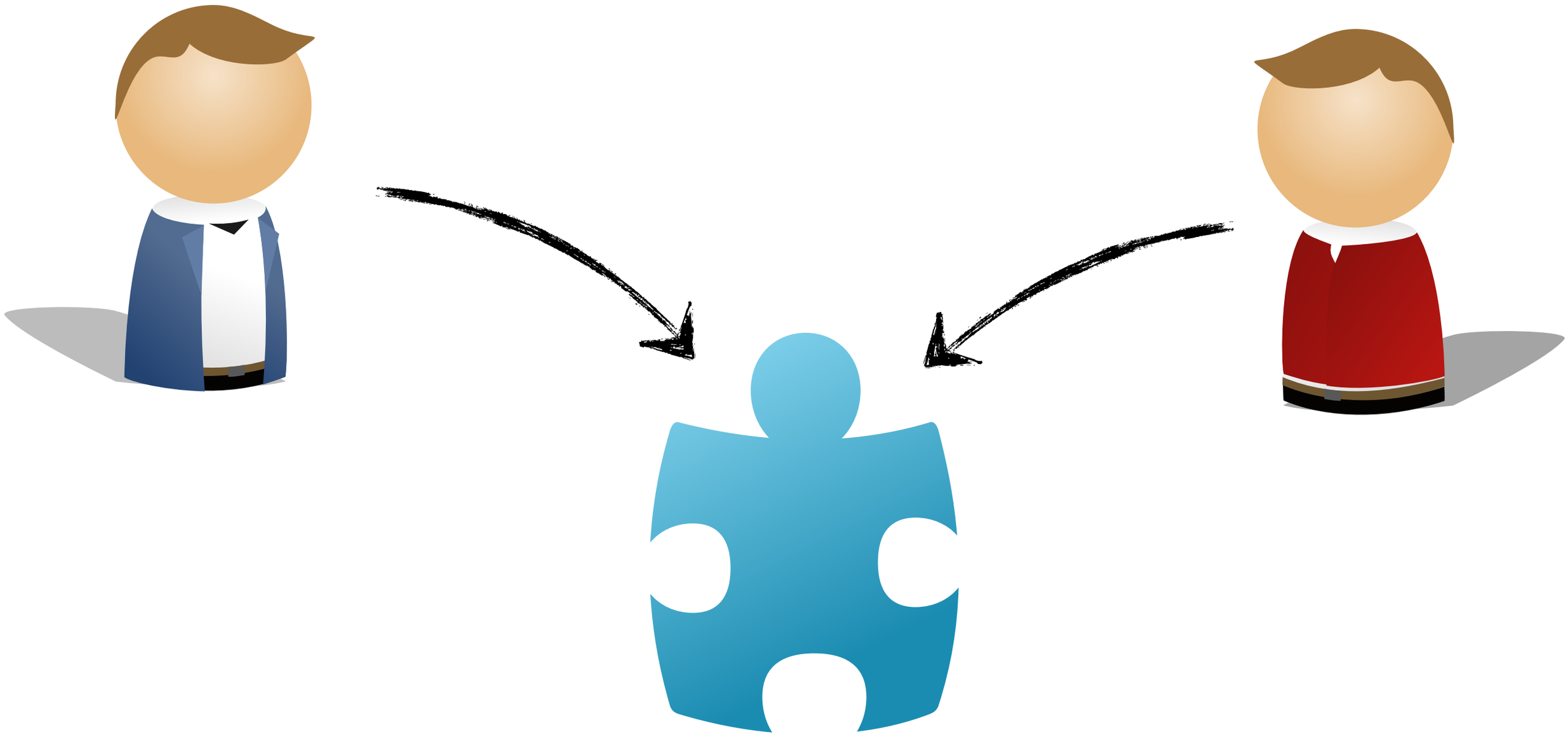
MBF

developer

works on



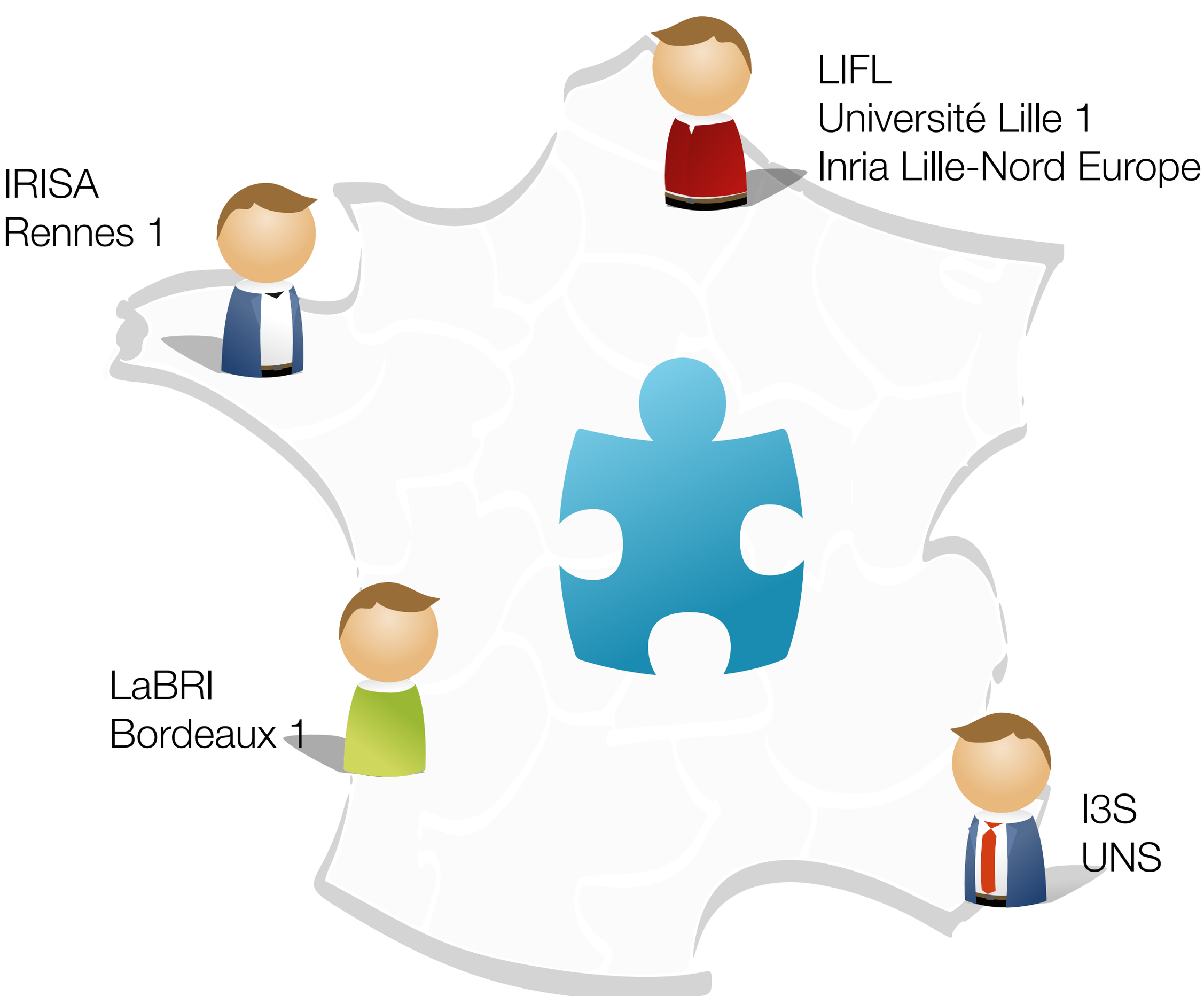
piece of software

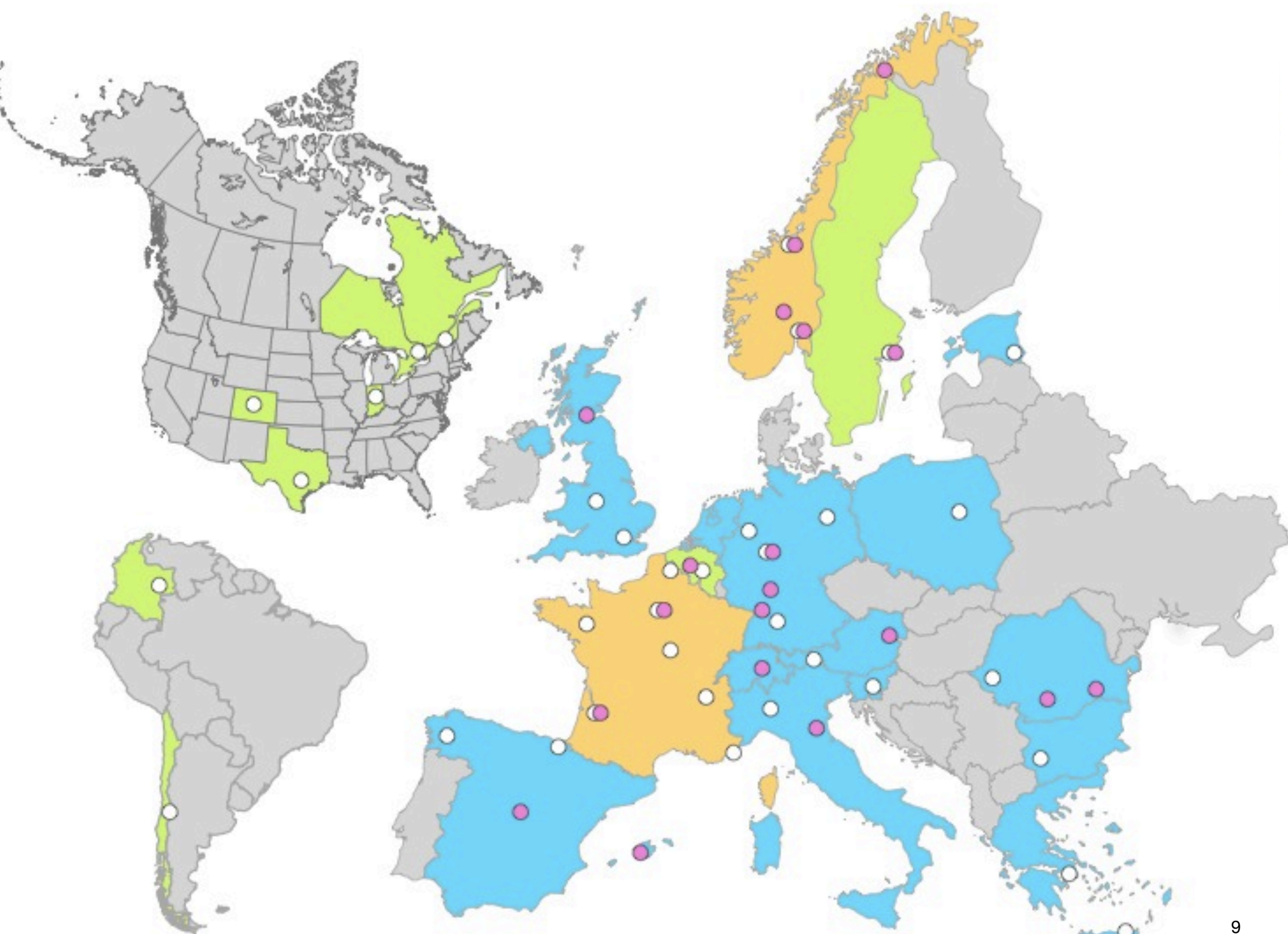


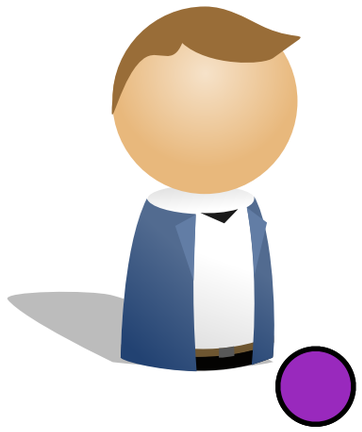
Email?

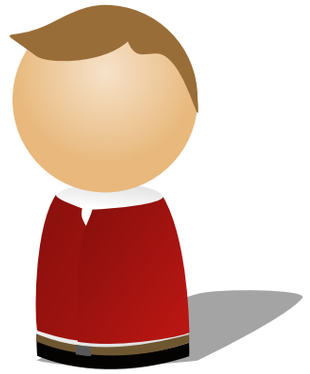
USB Key?

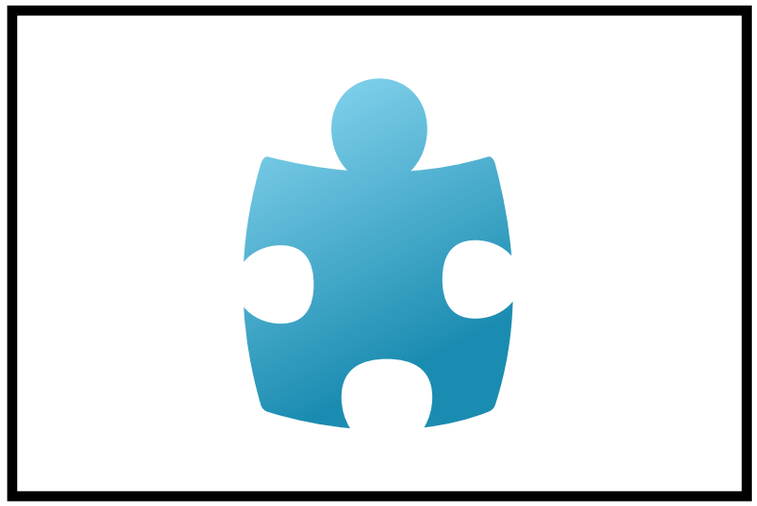
Shared directory?


LIFL
Université Lille 1
Inria Lille-Nord Europe
IRISA
Rennes 1
LaBRI
Bordeaux 1
I3S
UNS





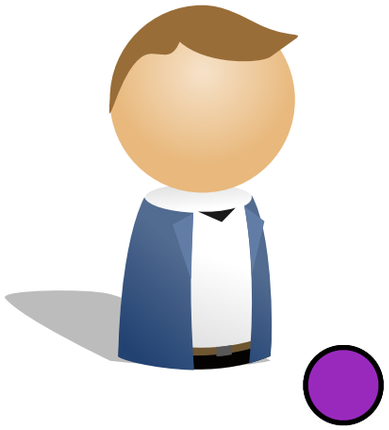


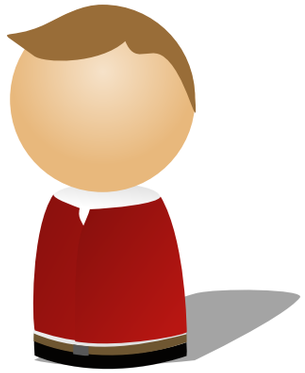


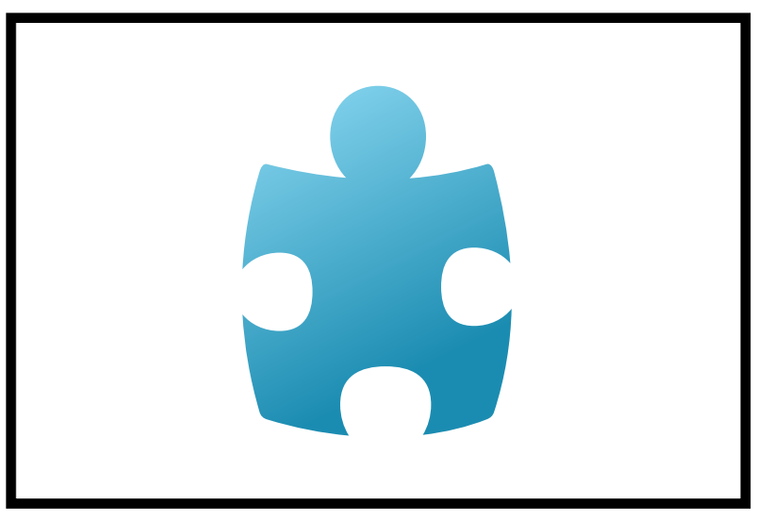
???

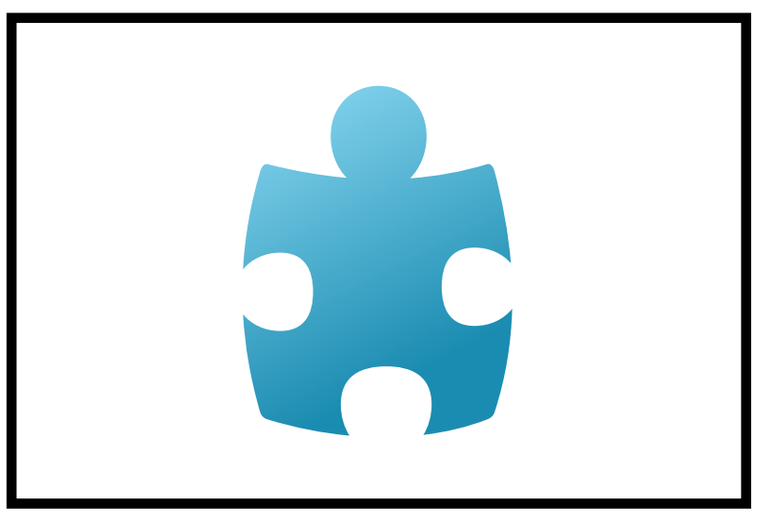
???

To share changes!









«Why do we version source code?»

Motivations
(among others)

Sébastien Mosser

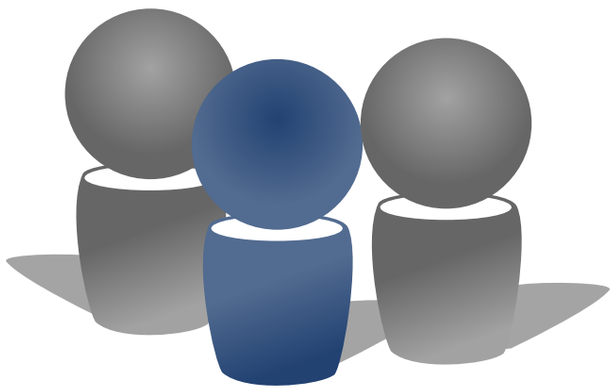
http://theplanetd.com/mongol-rally-back-in-time/

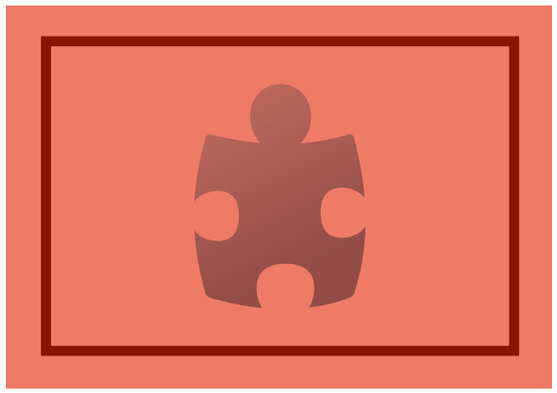
«Why do we version source code?»

To keep version history
To be able to rollback

MBF

Here is the new release!





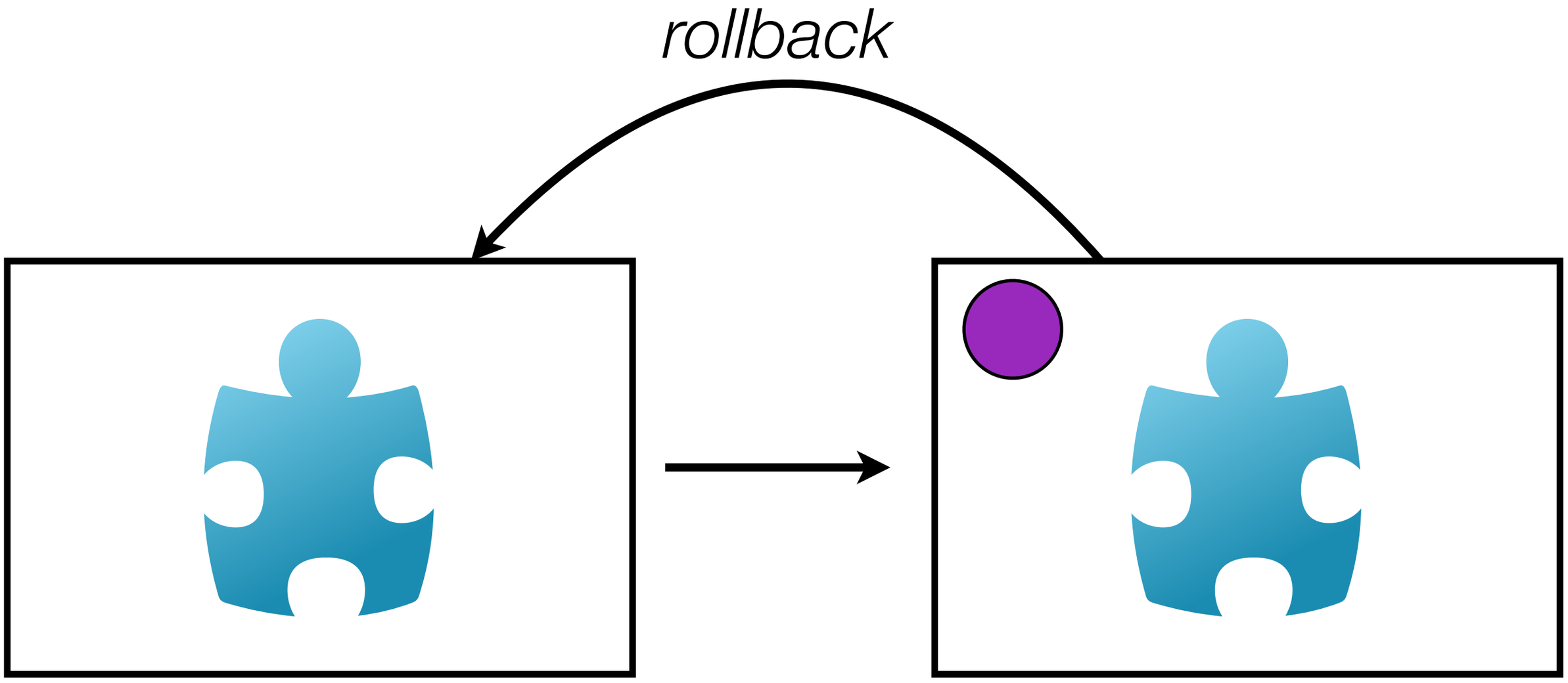
BUG!

???

It was working 2 days ago...

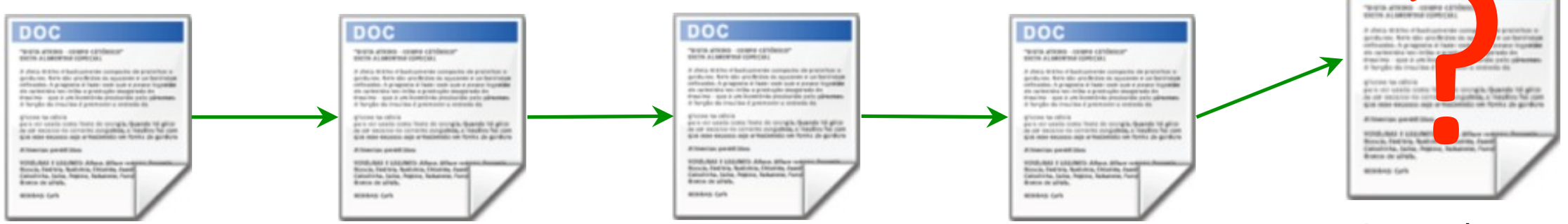
Can I see it?

To rollback changes!


rollback

Problématique

- Cas : édition d'un document



Rédaction section 1&2 | Rédaction section 3 | Échange sections 2 & 3 | Rédaction section 4 | 1 § initialement ds section 2 a disparu ?

- Erreur de manipulation ?
- Comment récupérer ?
  - Voyager dans le temps pour visiter une version antérieure du document
- **Version / Révision**
  - Une révision d'un document ou d'un projet est un *instantané* pris à un instant donné et sauvegardé dans l'historique du projet

Rôles d'un gestionnaire de version

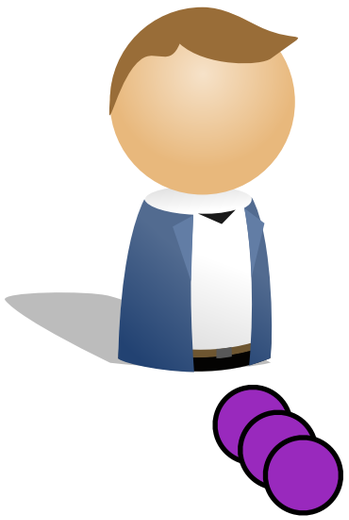
- Fonction 1
  - Gérer un historique qui permette
    - D'enregistrer de nouvelles révisions à tout moment
    - De récupérer n'importe quelle révision enregistrée

http://blogs.wsj.com/photojournal/2012/03/13/photos-of-the-day-march-13/

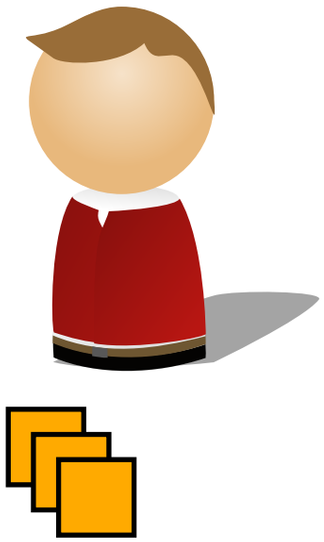
«Why do we version source code?»

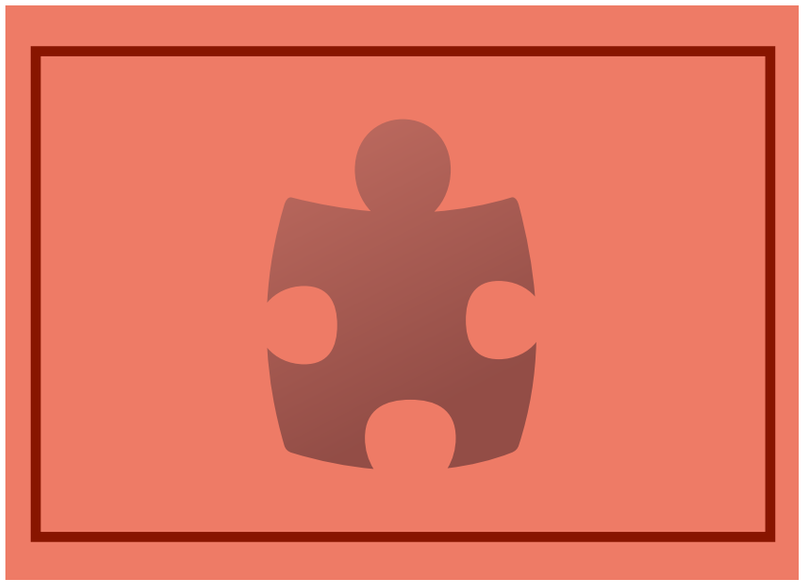
To know who worked on...

«not me!»



«not me!»



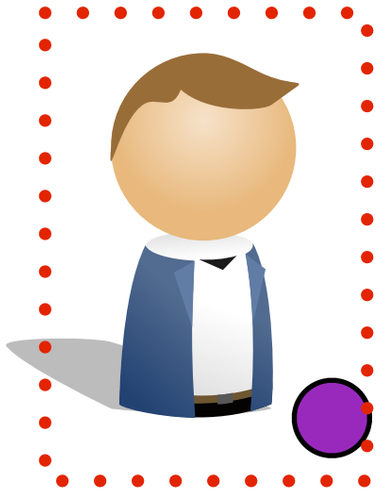


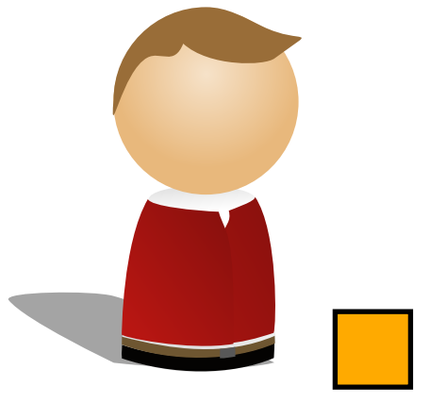
«not me!»

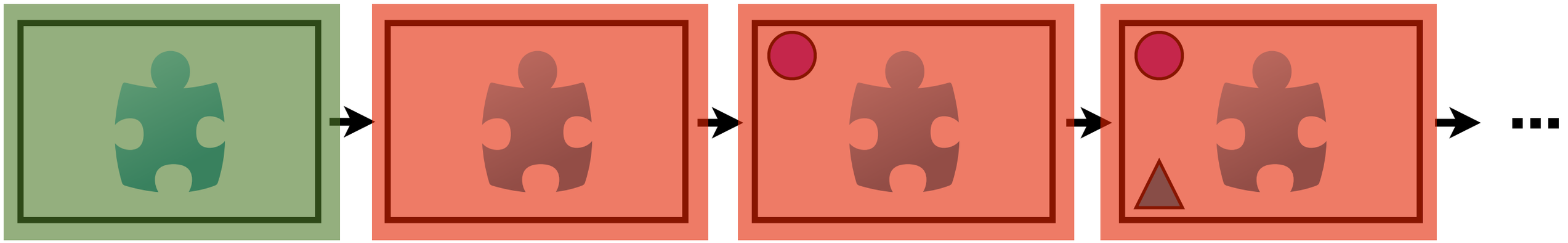
BUG!

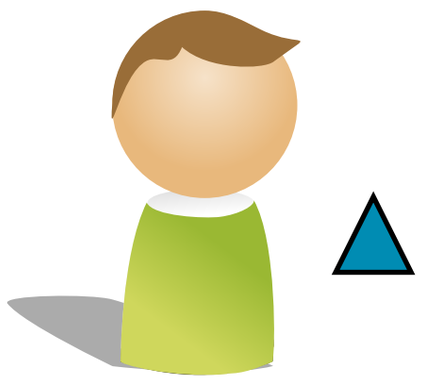
«not me!»

To trace changes!









BUG!

Sébastien Mosser

Problématique

- Cas : développement d'un logiciel
  - Après plusieurs mois de développement, l'impression ne fonctionne plus !!!!
  - Quand ce bogue a été introduit ? Quelle(s) modifications est(sont) responsable(s) ?
  - Réessayer les versions précédentes jusqu'à trouver la dernière sans bogue → super c'est la version k !
  - Problème
    - Qu'est ce qui a changé entre la version k et la version k+1 ???
    - Il aurait fallu documenter chaque modification !!!

Rôles d'un gestionnaire de versions

- Fonction 1
  - Gérer un historique qui permette
    - D'enregistrer de nouvelles révisions à tout moment
    - De récupérer n'importe quelle révision enregistrée
- **Fonction 2**
  - **Documenter chaque révision en lui associant un message**

Problématique

- Cas : développement d'un logiciel (suite)
  - Le projet contient une fonction incompréhensible !
    - Qui a écrit cette fonction? qui l'a modifié?
  - Vous avez localisé une ligne de code qui pourrait contenir un bogue mais pas sûr à 100 %
    - Ca serait bien de demander des explications à son auteur

Rôles d'un gestionnaire de version

- Fonction 1
  - Gérer un historique qui permette
    - D'enregistrer de nouvelles révisions à tout moment
    - De récupérer n'importe quelle révision enregistrée
- Fonction 2
  - Documenter chaque révision en lui associant un message
- **Fonction 3**
  - **Noter l'auteur de chaque révision**
  - **Ceci permet d'associer un responsable à chaque révision**

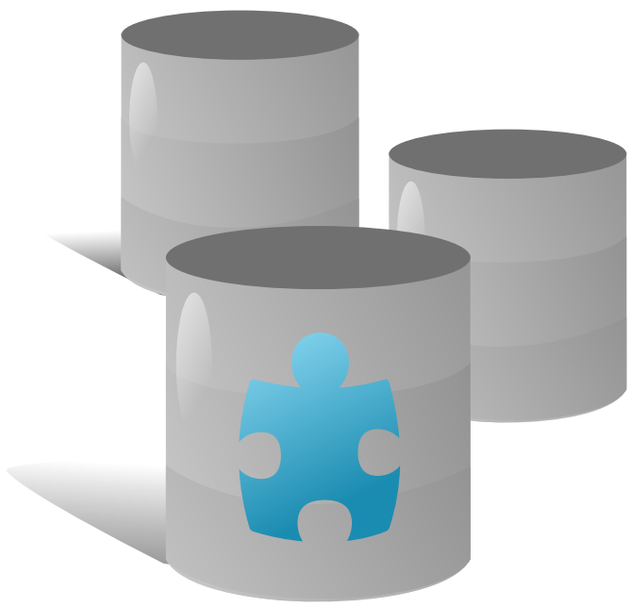
http://paysageislande.blogspot.fr/2013/06/periple-dans-le-sud-de-lislande-jour-5.html

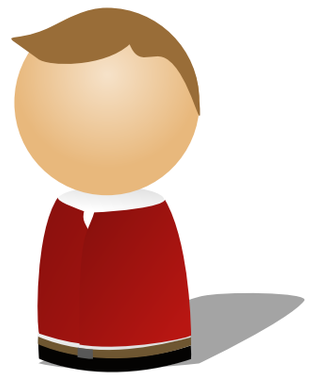
«Why do we version source code?»

To share changes
To support merging

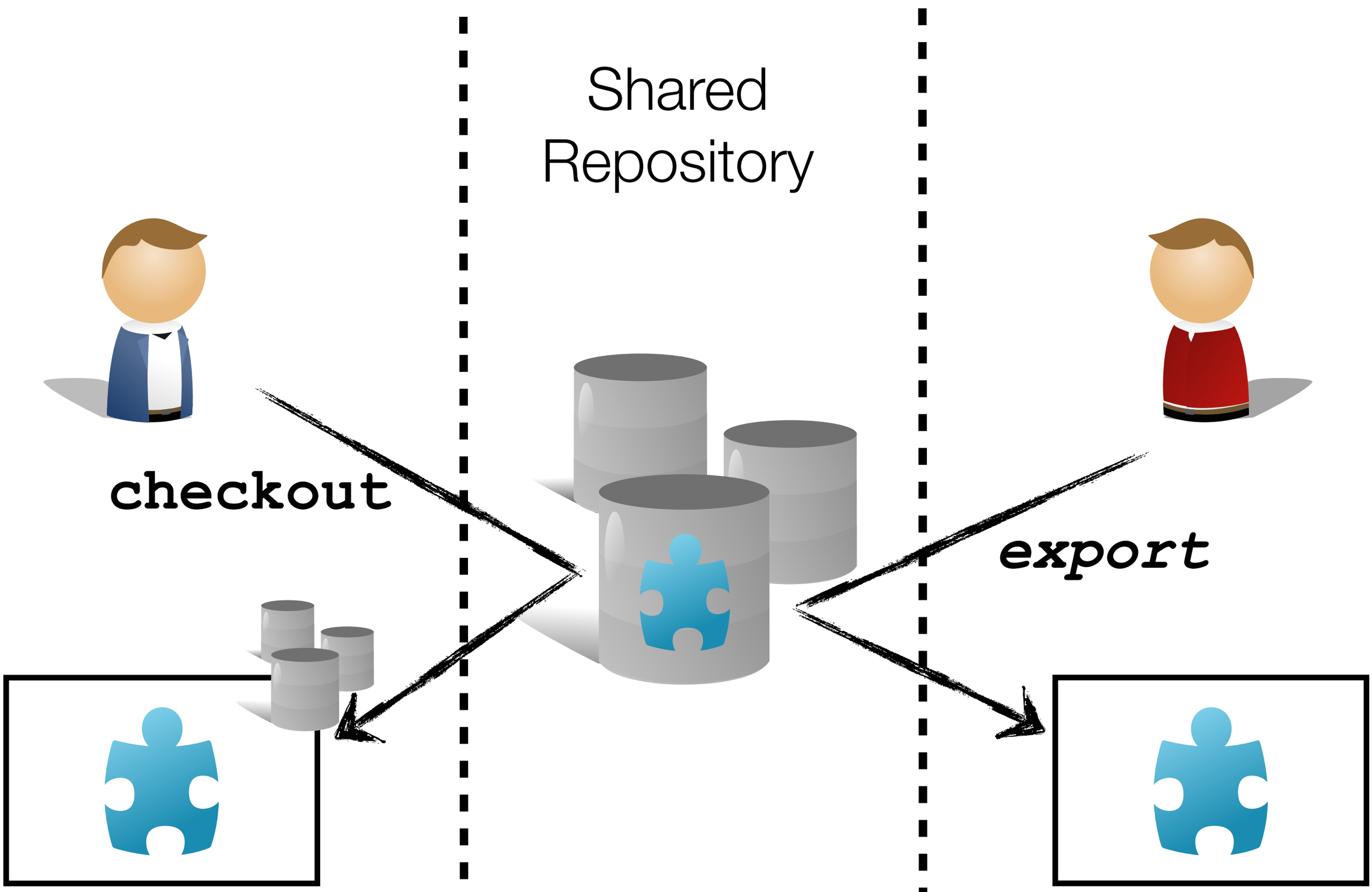
MBF

Shared
Repository



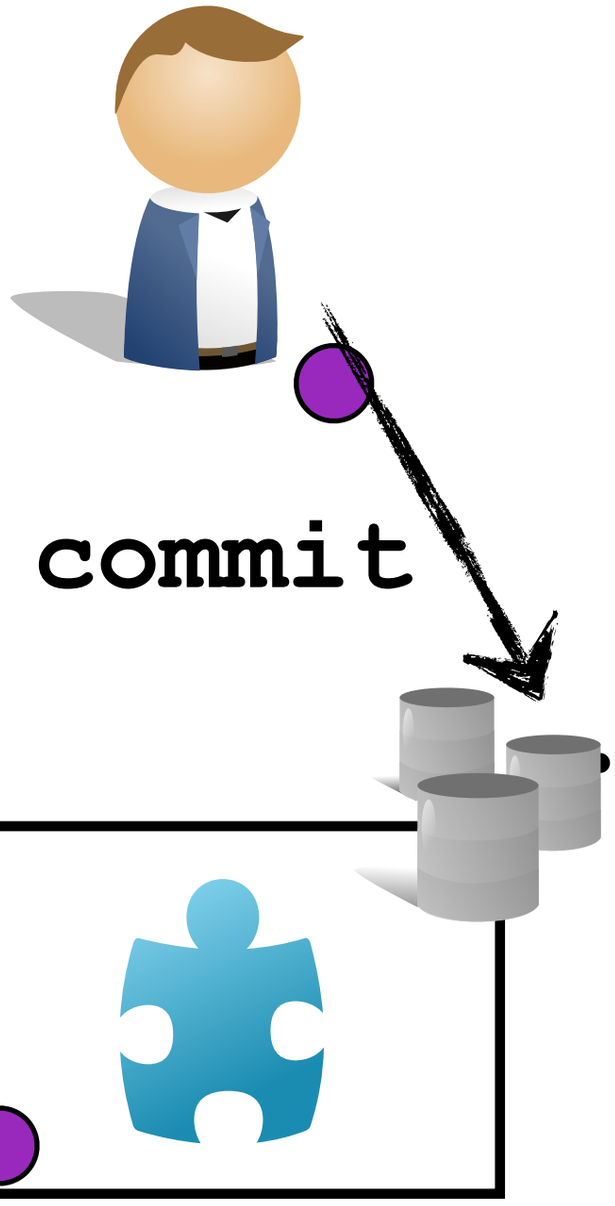


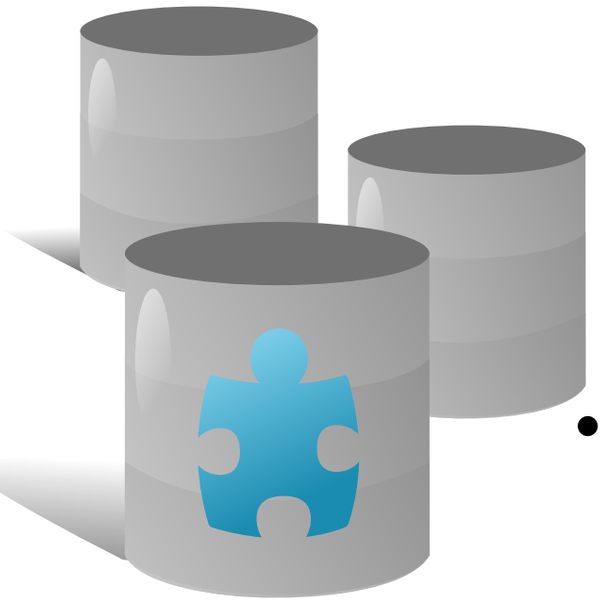
create

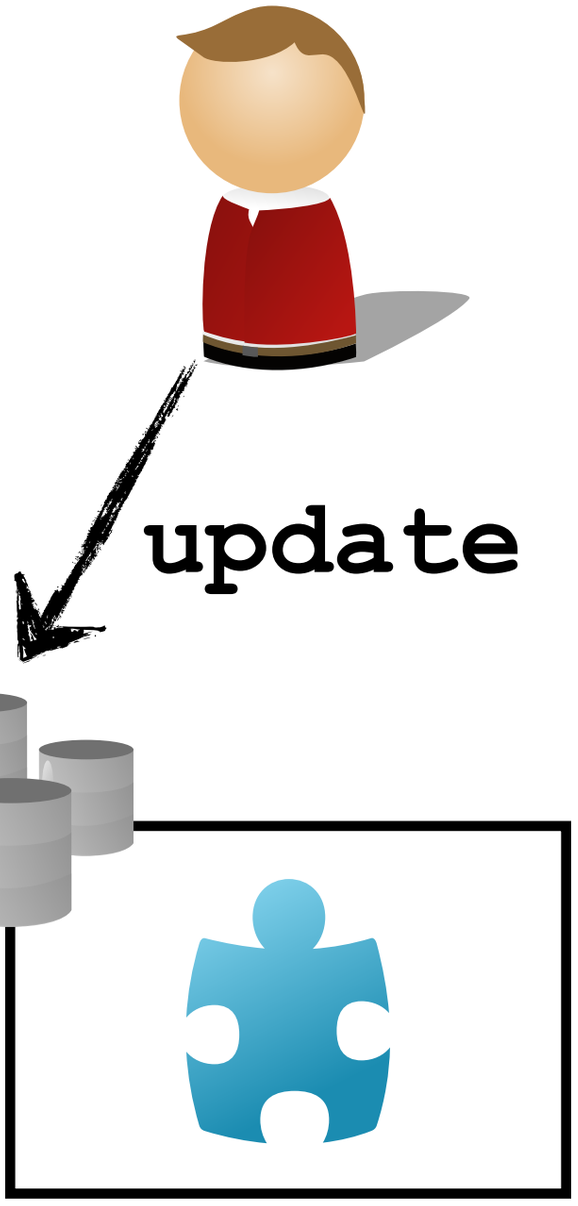


Récupérer un projet depuis le serveur (checkout)

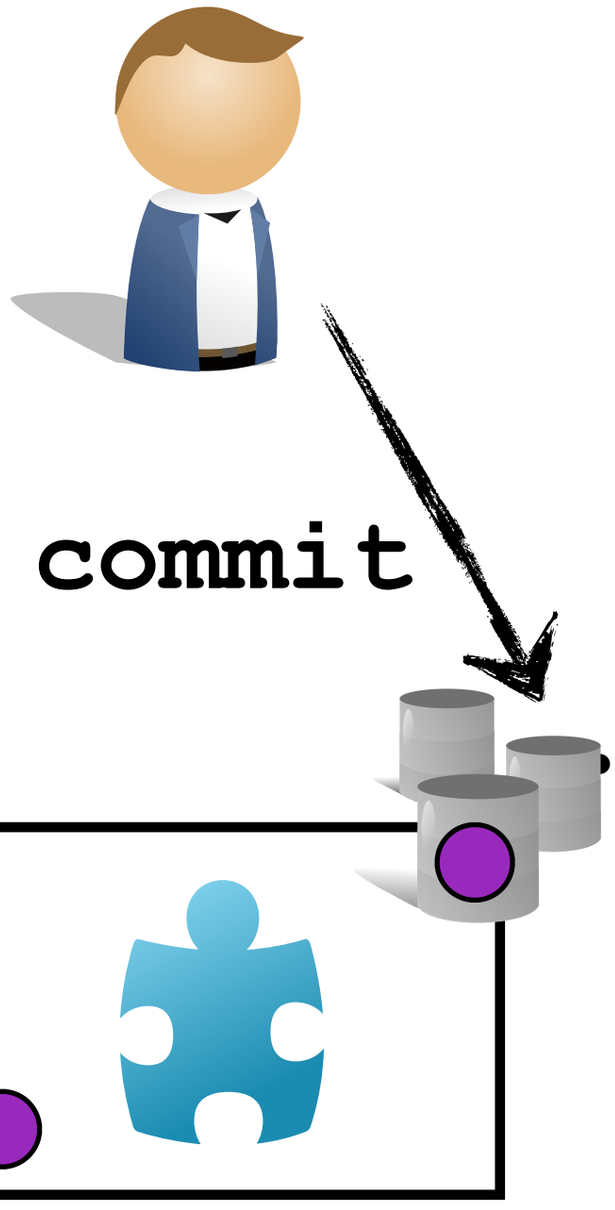
Shared Repository

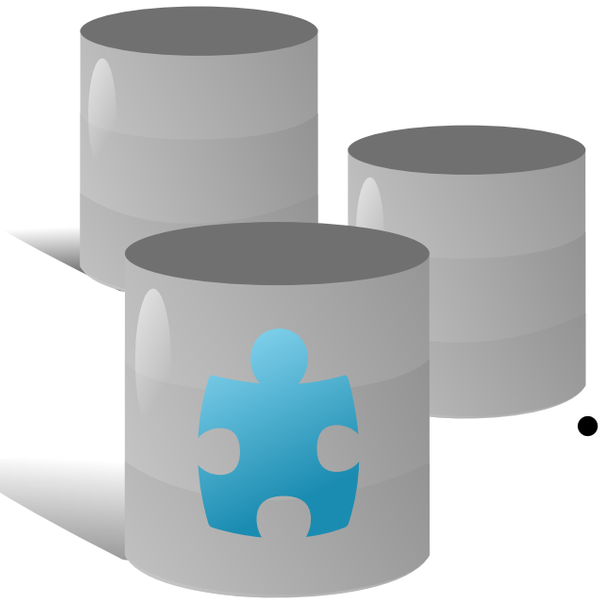


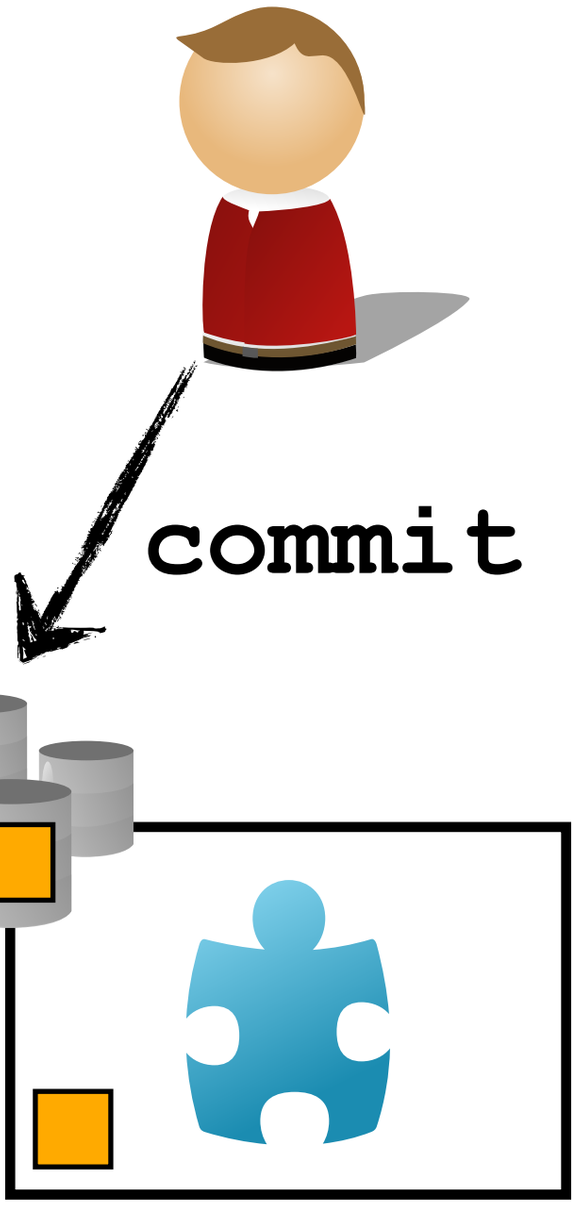




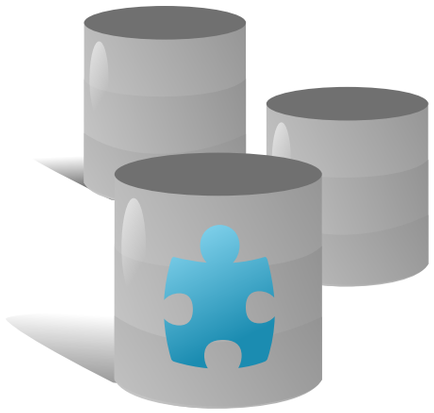
Shared Repository



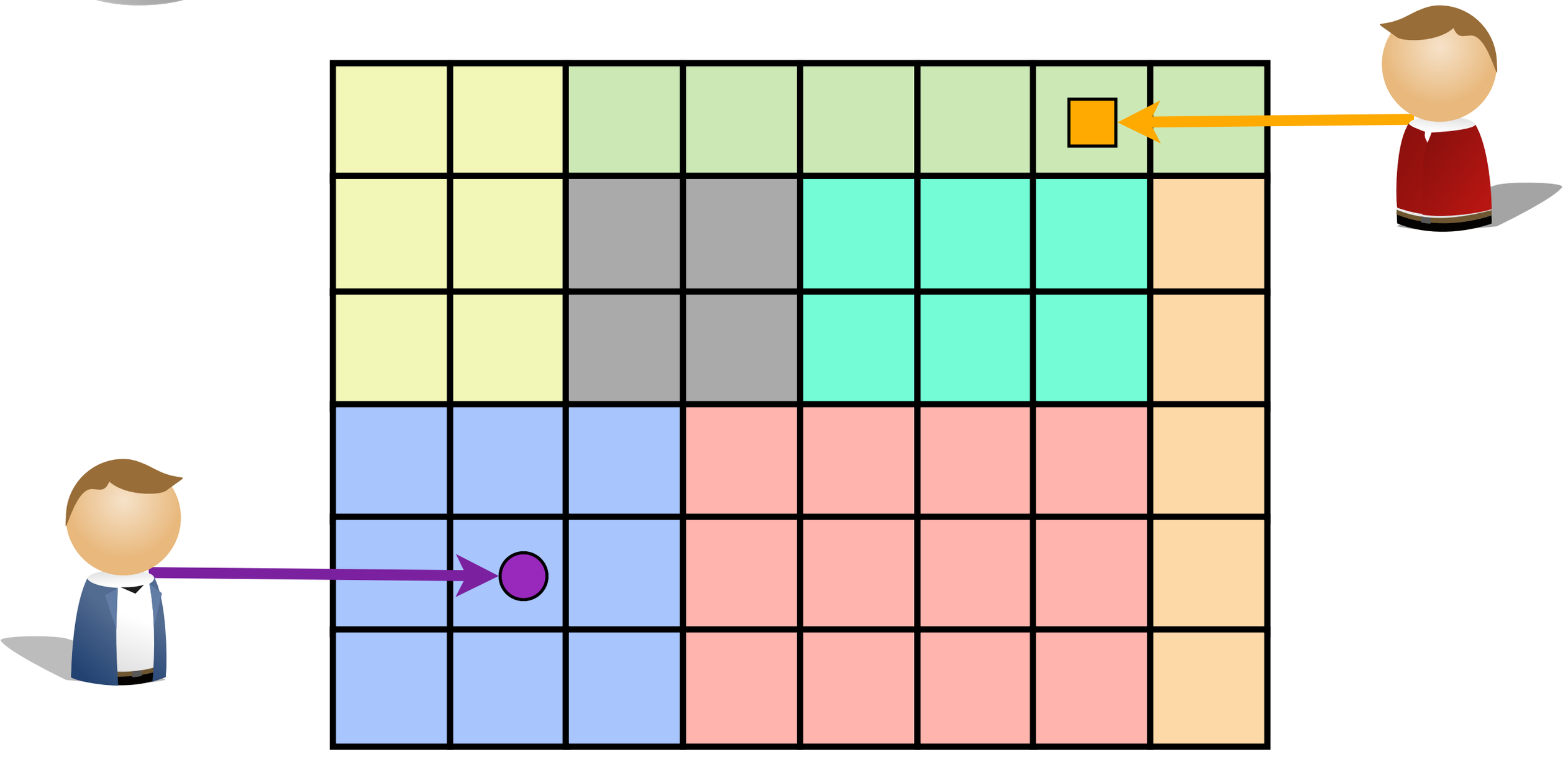




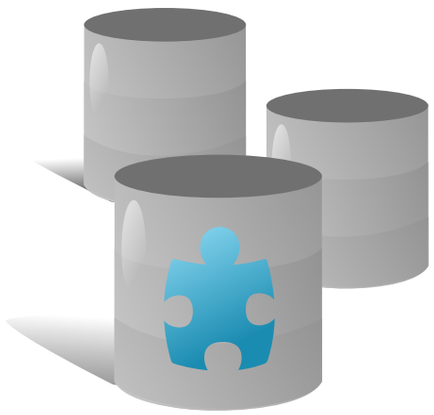
????



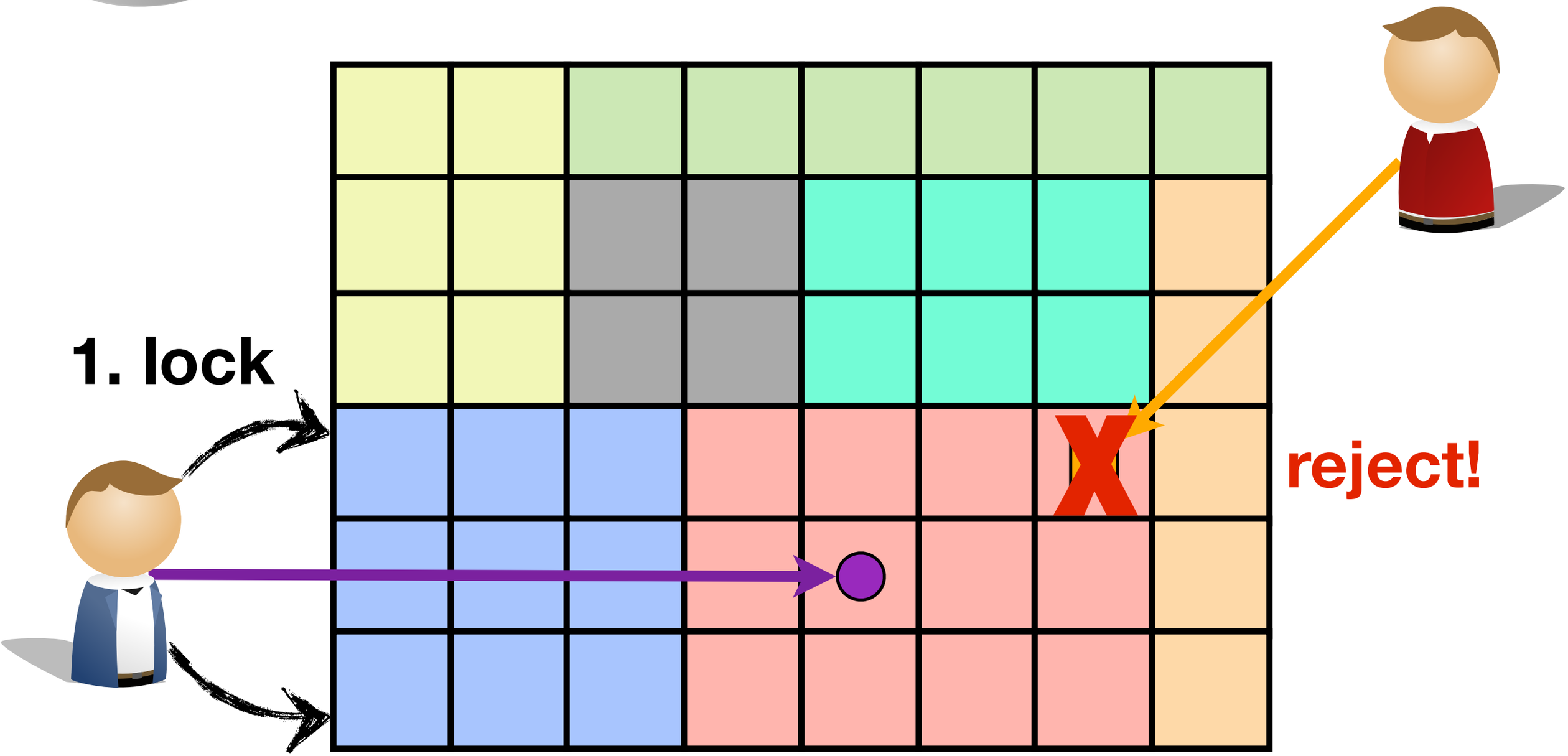
Case #1: different files



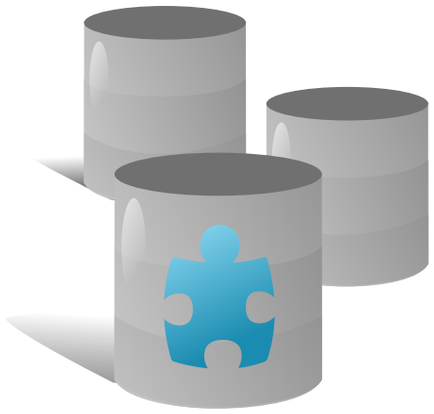
Atomic operations. No problem at all!



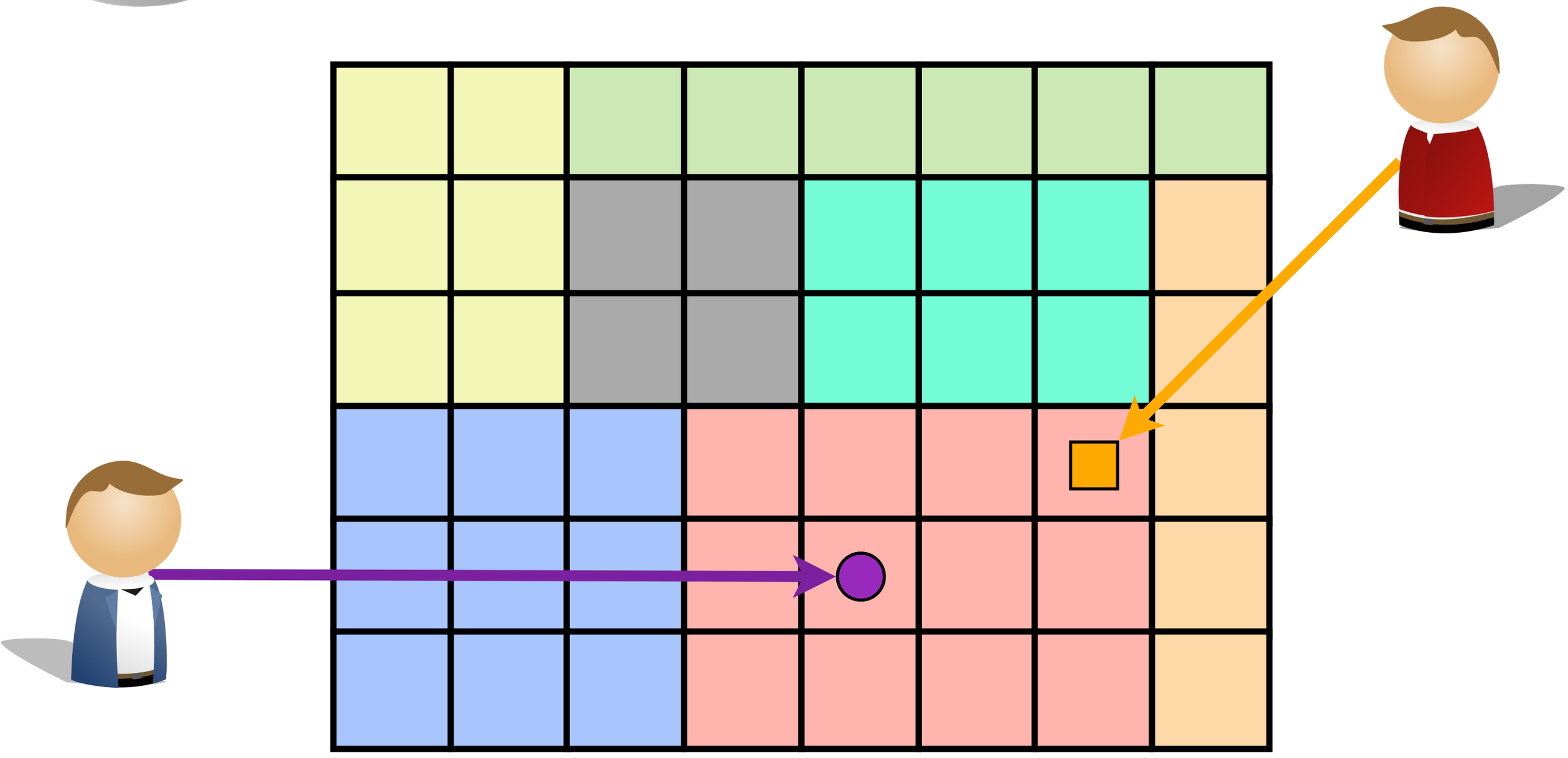
Case #2: different part of the same file



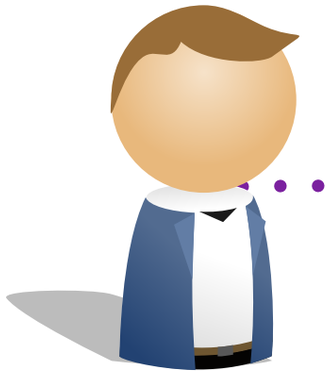
File Locking (old school)



Case #2: different part of the same file

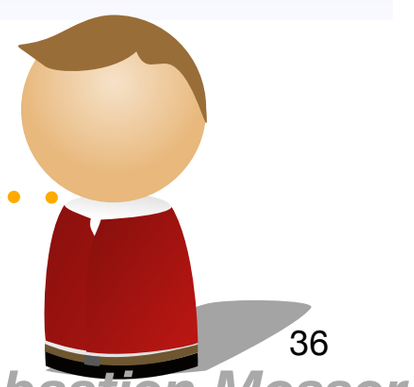


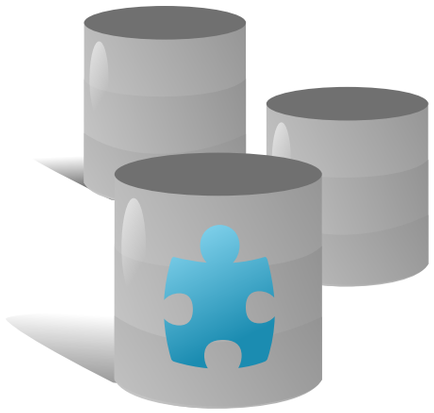
Automatic merge



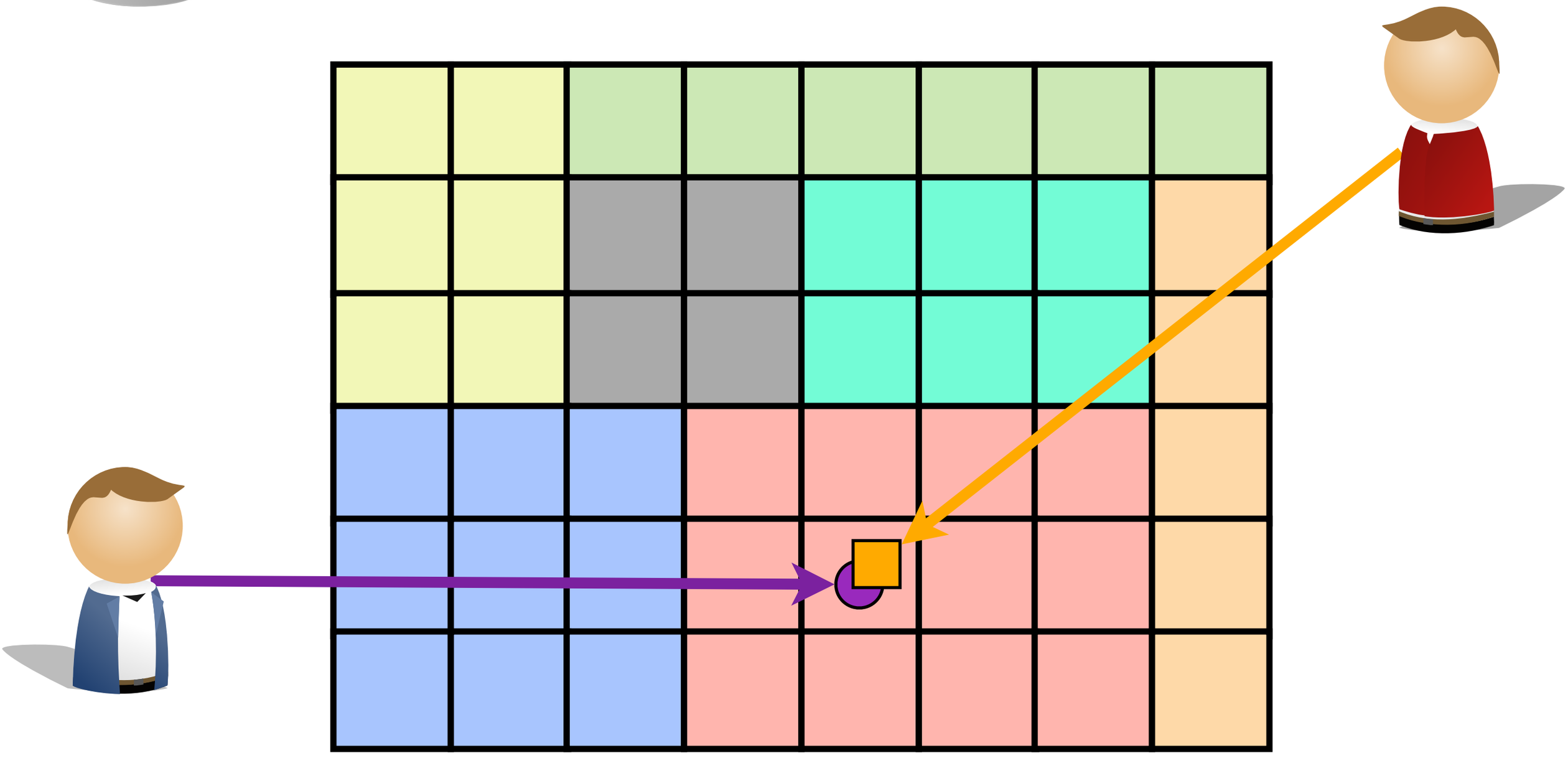
ScalaMOF

```
...  ...  @@ -45,8 +48,8 @@ class Operation extends TypedElement with MultiplicityElement {
45   48       */
46   49      def `class`: Class = _class
47   50      def class_=(c: Class) {
48        -     require(c != null)
49        -     require(c.ownedOperations contains this)
          51  +     require(c != null, "`class` attribute cannot be null")
          52  +     require(c.ownedOperations contains this, "`class` must contain this operation")
50   53        _class = c
51   54      }
52   55      private[this] var _class: Class = _
...  ...  @@ -54,7 +57,7 @@ class Operation  extends TypedElement with MultiplicityElement {
54   57      /**
55   58       * <em>"The parameters to the operation."</em>
56   59       */
57        -   def ownedParameters: Seq[Parameter] = _ownedParameters
          60  +   def ownedParameters: Seq[Parameter] = _ownedParameters.reverse
58   61      private[this] var _ownedParameters = List[Parameter]()
```



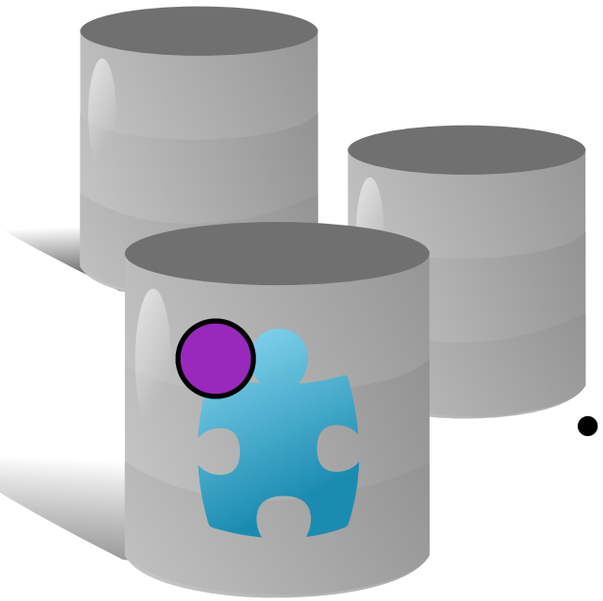


Case #3: **same** part of the **same** file

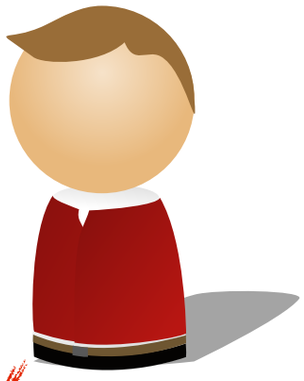


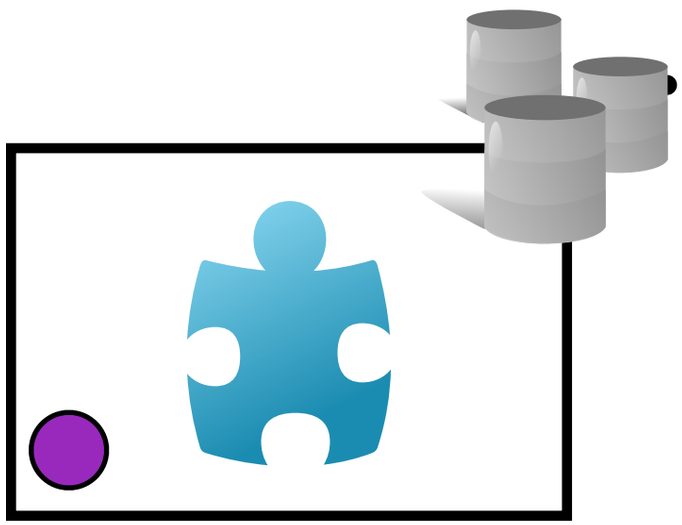
Conflict!

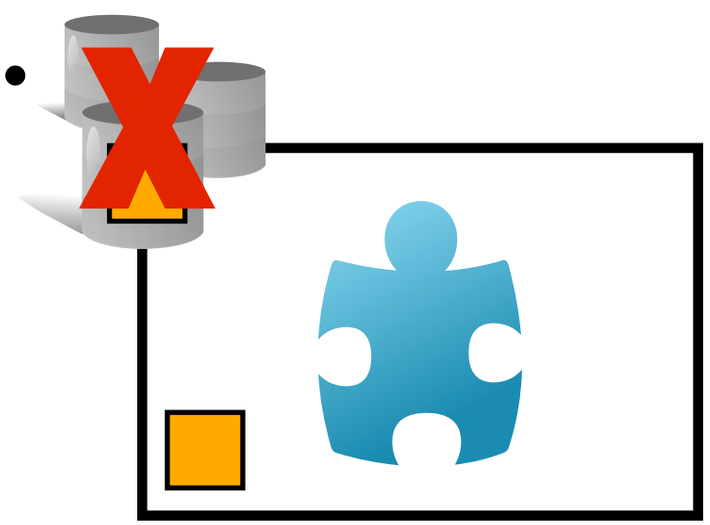
Shared
Repository



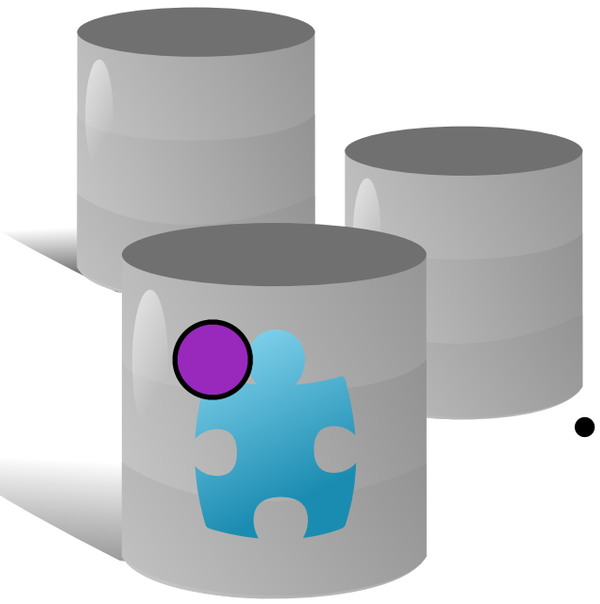
commit

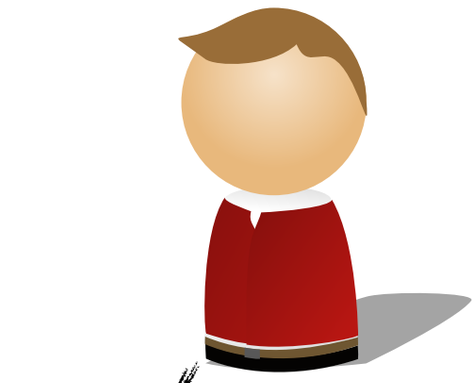




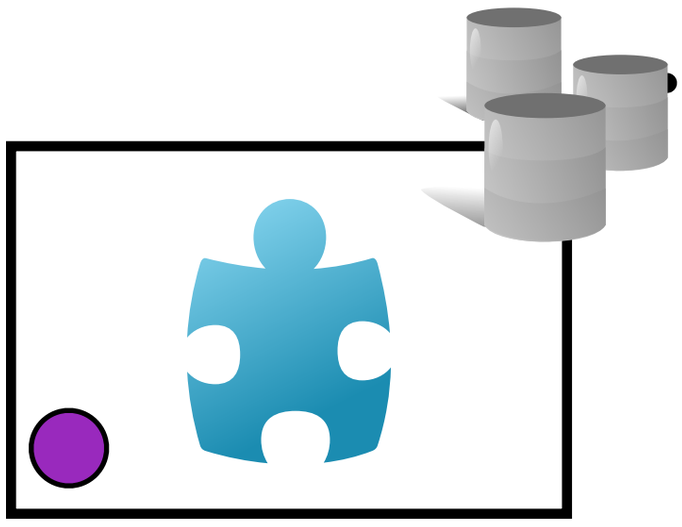


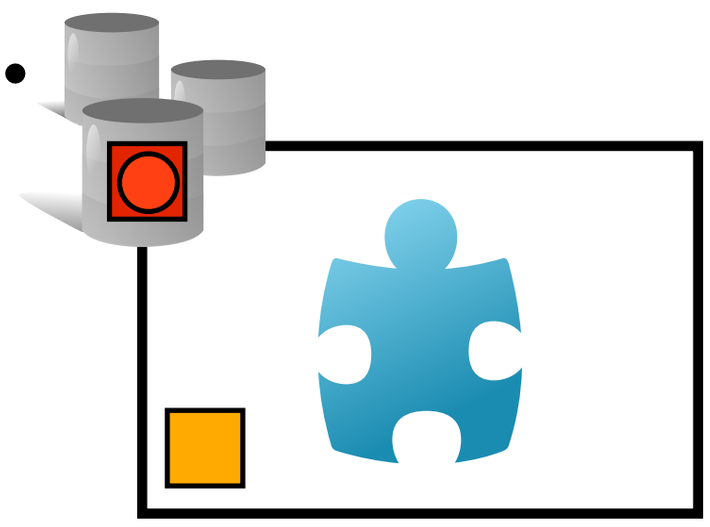
Shared Repository





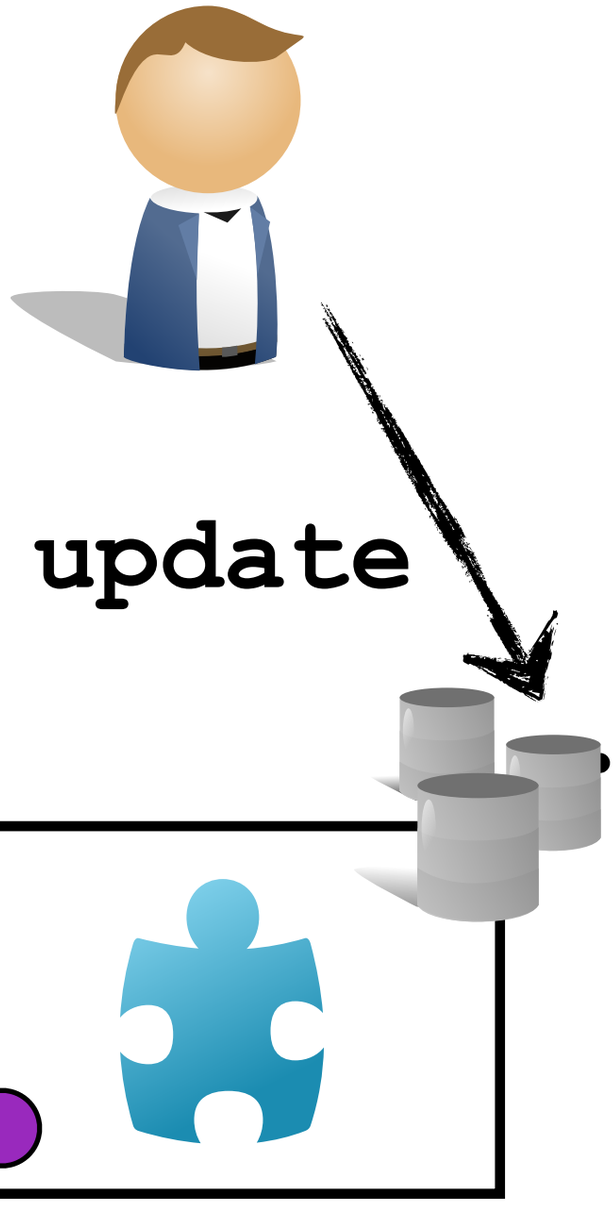
update

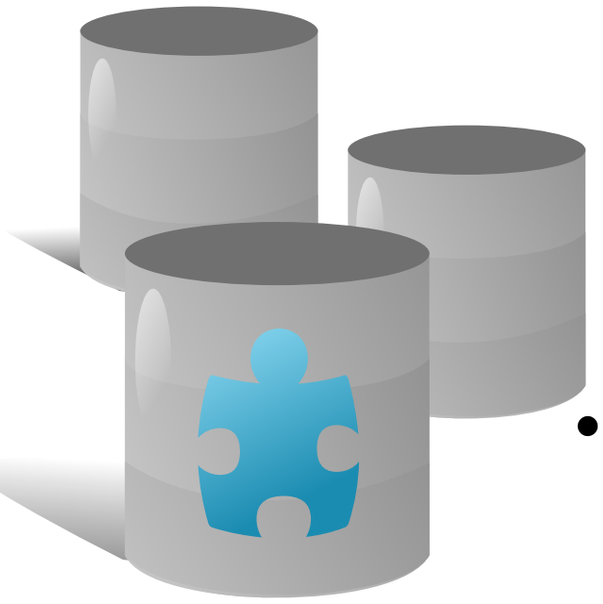


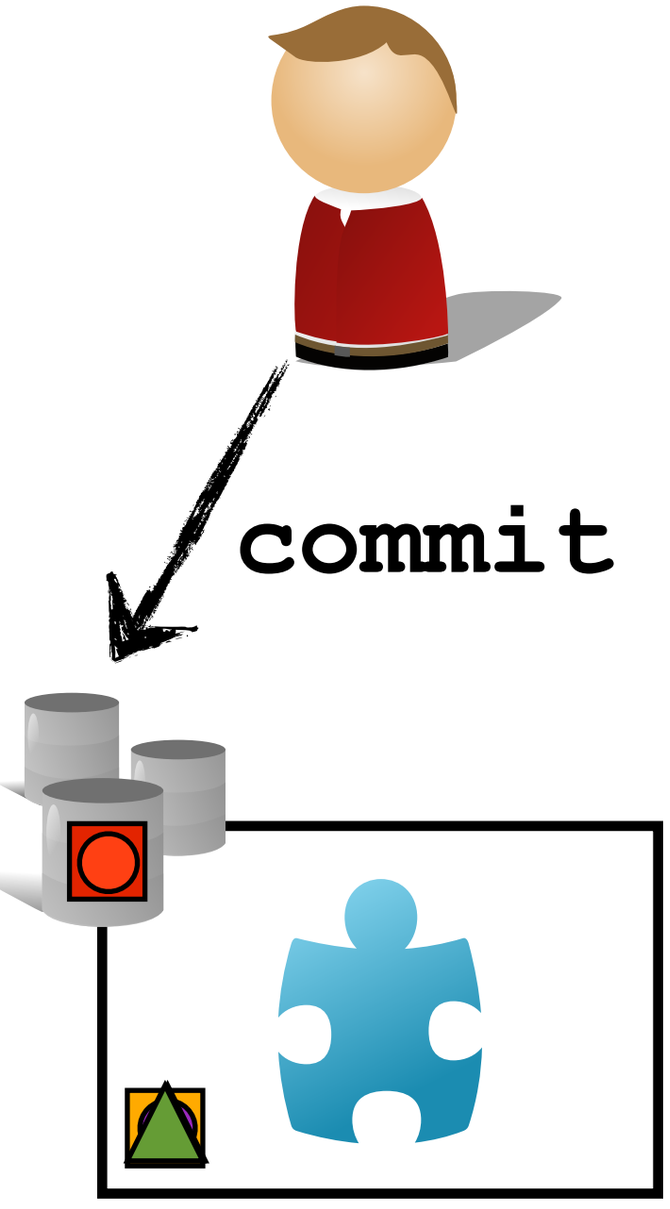


Conflict!

Shared Repository







Resolved!

Problématique

- Cas : travail en équipe

Dépôt initial

jeu.h

jeu.c

A fait une copie de travail

jeu.h

jeu.c

B fait une copie de travail

jeu.h

jeu.c

Problématique

- Cas : travail en équipe

Dépôt initial

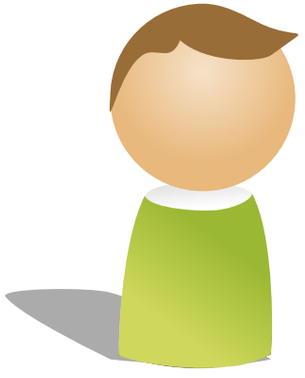
jeu.h

jeu.c

A modifie jeu.h

jeu.h

jeu.c



B modifie jeu.c

jeu.h

jeu.c

- A & B doivent échanger leurs modifs
- Le gestionnaire de version va servir d'intermédiaire

Problématique

- Cas : travail en équipe (autre cas)

Dépôt initial

A fait une copie de travail

B fait une copie de travail

jeu.h

jeu.c

jeu.h

jeu.c

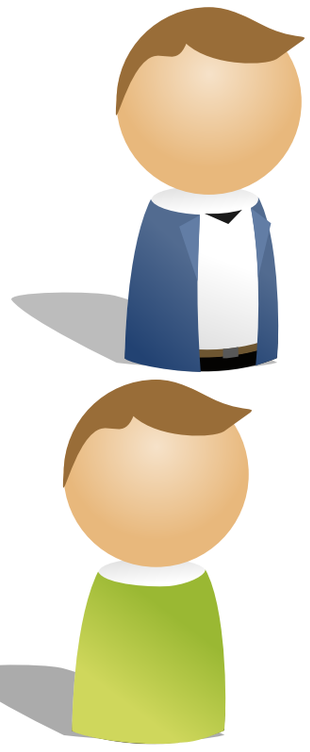
jeu.h

jeu.c

Problématique

- Cas 4 : travail en équipe (autre cas)

Dépôt initial



A modifie jeu.c

B modifie jeu.c

jeu.h

jeu.c

jeu.h

jeu.c

jeu.h

jeu.c

- A doit intégrer les modifs de B
- B doit intégrer les modifs de A
- Problème de la fusion (merge)

Rôles d'un gestionnaire de version

- Fonction 1
  - Gérer un historique qui permette
    - D'enregistrer de nouvelles révisions à tout moment
    - De récupérer n'importe quelle révision enregistrée
- Fonction 2
  - Documenter chaque révision en lui associant un message
- Fonction 3
  - Noter l'auteur de chaque révision
  - Ceci permet d'associer un responsable à chaque ligne de code dans chaque révision
- Fonction 4
  - Faciliter le travail en multipostes en permettant l'accès à distance à un dépôt partagé par tousles postes
- **Fonction 5**
  - **Faciliter la fusion des modifications en gérant les conflits**

Problématique

- Cas : même logiciel / différentes tâches
  - Plusieurs systèmes d'exploitation
    - Linux, Mac, Windows
  - Plusieurs lignes de logiciels
    - Produit commercialisé (stable)
    - Nouvelle idée (instable, en cours de développement)
  - Plusieurs cibles
    - Logiciel bridé (version de démo)
    - Logiciel « pour particulier »
    - Logiciel « pour entreprise »
  - Plusieurs bogues
    - Logiciel commercialisé
    - Repérage du bogue 277
    - Repérage du bogue 389

Rôles d'un gestionnaire de versions

- Fonction 1
  - Gérer un historique qui permette
    - D'enregistrer de nouvelles révisions à tout moment
    - De récupérer n'importe quelle révision enregistrée
- Fonction 2
  - Documenter chaque révision en lui associant un message
- Fonction 3
  - Noter l'auteur de chaque révision
  - Ceci permet d'associer un responsable à chaque ligne de code dans chaque révision
- Fonction 4
  - Faciliter le travail en multipostes en permettant l'accès à distance à un dépôt partagé par tousles postes
- Fonction 5
  - Faciliter la fusion des modifications en gérant les conflits
- **Fonction 6**
  - **Faciliter le développement parallèle de multiples branches et le transfert de modifications entre branches**

Les systèmes

Système	**Archive où ?**	**Archive quoi ?**	**Quand ?**
CVS	Centralisé	Fichiers	1989
SVN (Subversion ou tortoiseSVN)	Centralisé	Arborescence	2000
Git ou BZR	Décentralisé	Arborescence	2005

- Version Control System
  - Comparaison
  - http://better-scm.berlios.de/comparison/comparison.html
- Sources
  - http://www.univ-orleans.fr/lifo/Members/duchier/
  - http://kalysto.org/~nono/teaching/GL/GL-cvs.pdf

Visualiser l'évolution d'un repository

Sous le répertoire svn ou git :

gource -s 1 -c 4

lancer video sinon

Attention, ...

Anti-informative Commit Messages

master

Author	Commit	Message
[redacted]	2285889	Version 22/02/2013
[redacted]	4e06542	Version 21/02/2013
[redacted]	5798097	Version 2 16/02/2013
[redacted]	775f97a	Version 16/02/2013

Commit message = Intention of the version

master

Author	Commit	Message
[redacted]	6ad5d7a	Correction des warnings
[redacted]	a2bd26c	Correction des erreures de nom de methode
[redacted]	7a3e625	Suppression de fourmiliere.java dans le pack

One-shot activity

Sunday

Monday

Tuesday

Wednesday

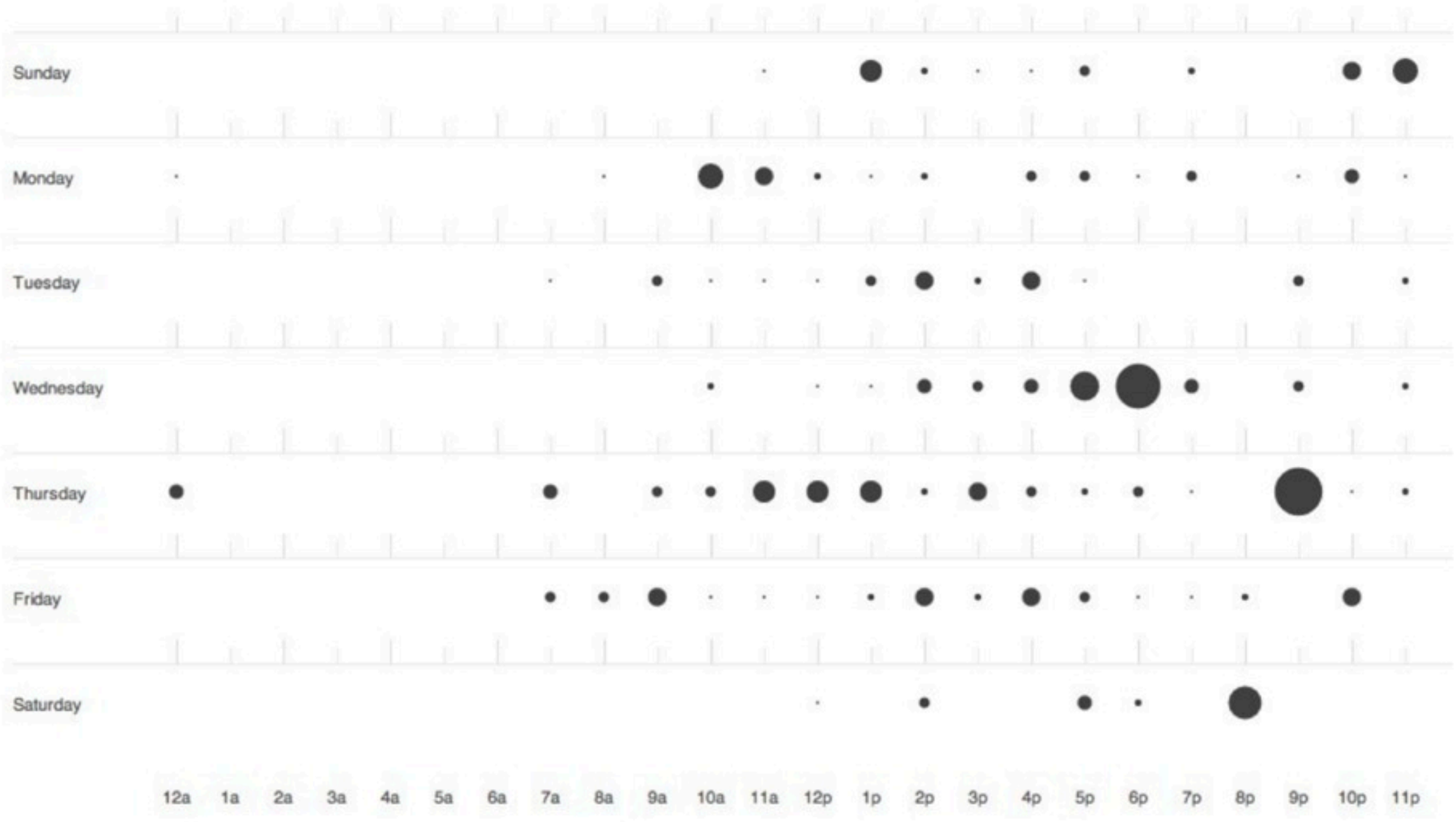
Thursday

Friday

Saturday

12a 1a 2a 3a 4a 5a 6a 7a 8a 9a 10a 11a 12p 1p 2p 3p 4p 5p 6p 7p 8p 9p 10p 11p

Expected: Continuous Development



The Archive Commit

master ▾ | public /

Myrmes[redacted].zip

test.txt

Sébastien Mosser

Versioned Binary Files

- src
  - fourmiliere
    - Evenement.java
    - Fourmiliere.java
    - Salle.java
    - SalleLarve.java
    - SalleOuvriere.java
    - SalleSoldat.java
    - SalleStock.java
- bin
  - fourmiliere
    - Fourmiliere.class
    - Salle.class
    - SalleLarve.class
    - SalleOuvriere.class
    - SalleSoldat.class
    - SalleStock.class

svn propset svn:ignore bin

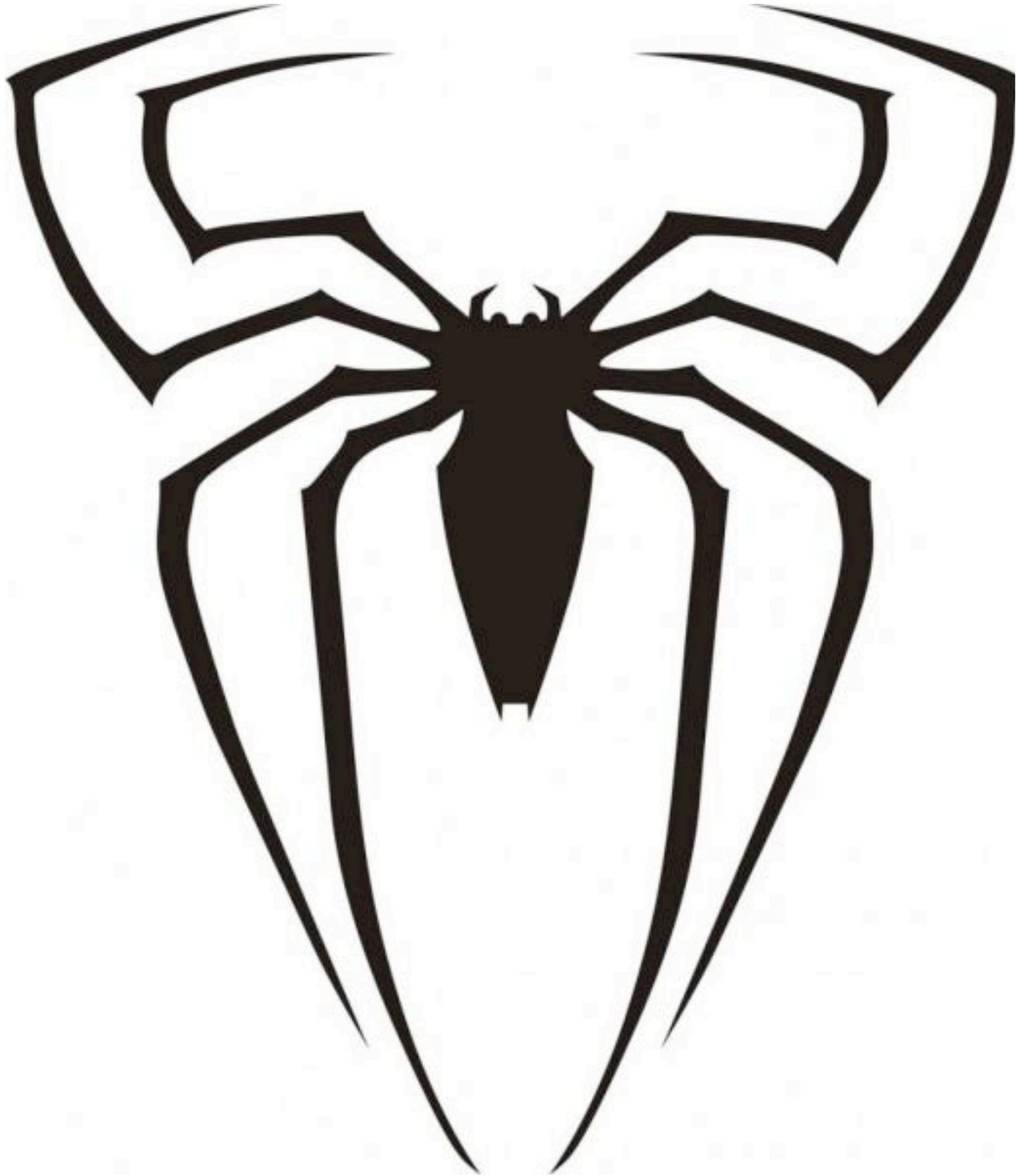
Commits must be related to tickets!

6f09558	TWTWGMM-38 Modification de la salle stock en fonction des ressources	2 days ago	TWTWGMM-38
336da47	TWTWGMM-40 Creation et implementation de la classe	2 days ago	TWTWGMM-40
daa568d	TWTWGMM-39 Creation et implementation de la classe	2 days ago	TWTWGMM-39
7084534	TWTWGMM-25 Modification de methode	2 days ago	TWTWGMM-25
eab5591	TWTWGMM-26 Modification de l'enum	2 days ago	TWTWGMM-26
b876abf	TWTWGMM-5 Modification des methodes	2 days ago	TWTWGMM-5
af70391	TWTWGMM-8 Definition d'un des	2 days ago	TWTWGMM-8
903a5f5	TWTWGMM-21 Definition d'un soldat	2 days ago	TWTWGMM-21
d7ca130	TWTWGMM-20 Definition d'une ouvriere	2 days ago	TWTWGMM-20
84be5fb	TWTWGMM-3 Definition d'une nourrice	2 days ago	TWTWGMM-3
399e9b0	TWTWGMM-22 Definition d'une larve	2 days ago	TWTWGMM-22
802fb3c	TWTWGMM-2 Definition d'une fourmi	2 days ago	TWTWGMM-2

Seriously?

Spiderman's Theorem

«With great power comes great responsibility»



Conclusions

Sébastien Mosser

To trace changes!

To rollback changes!

To share changes!

(among others)