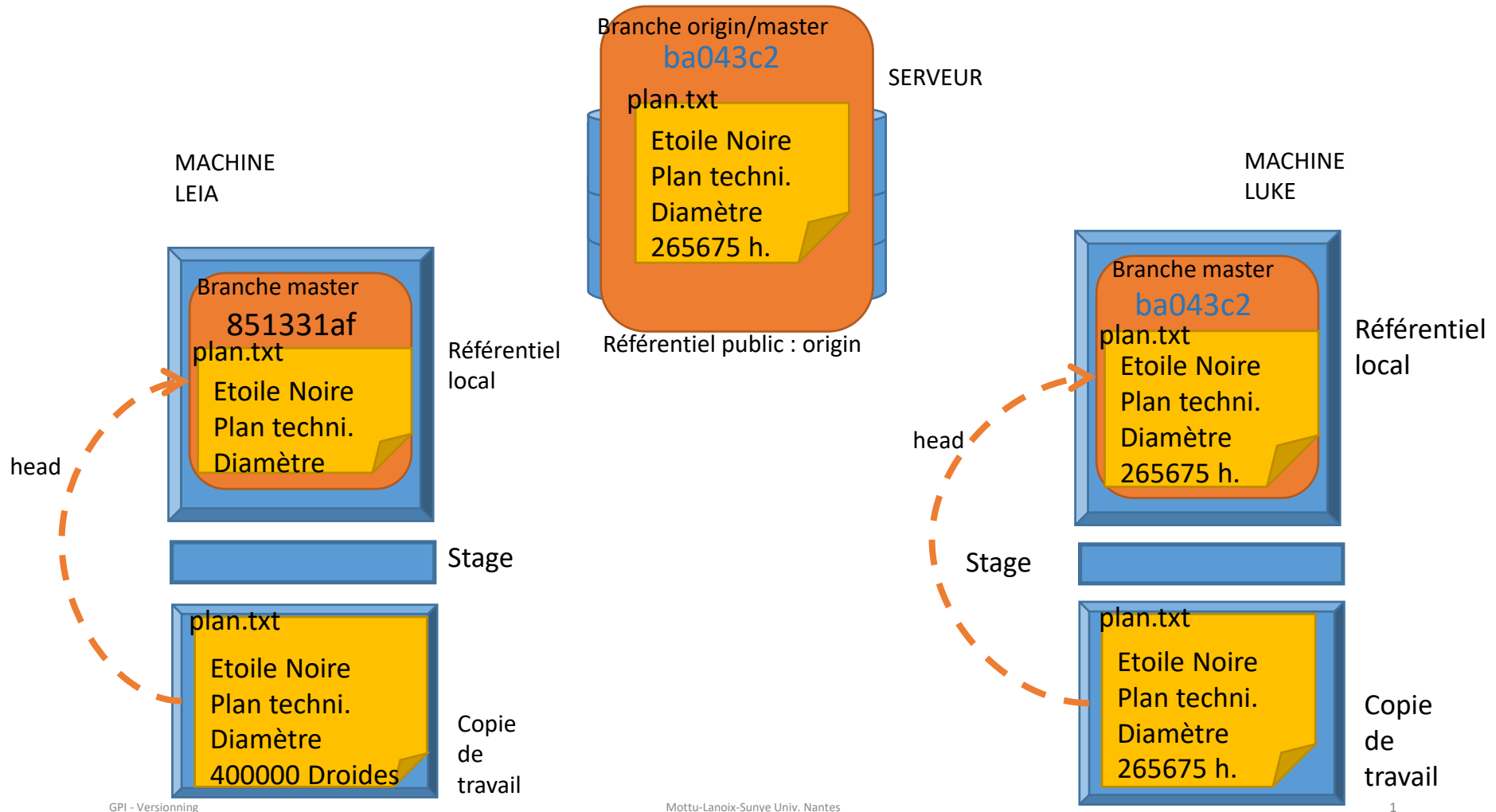


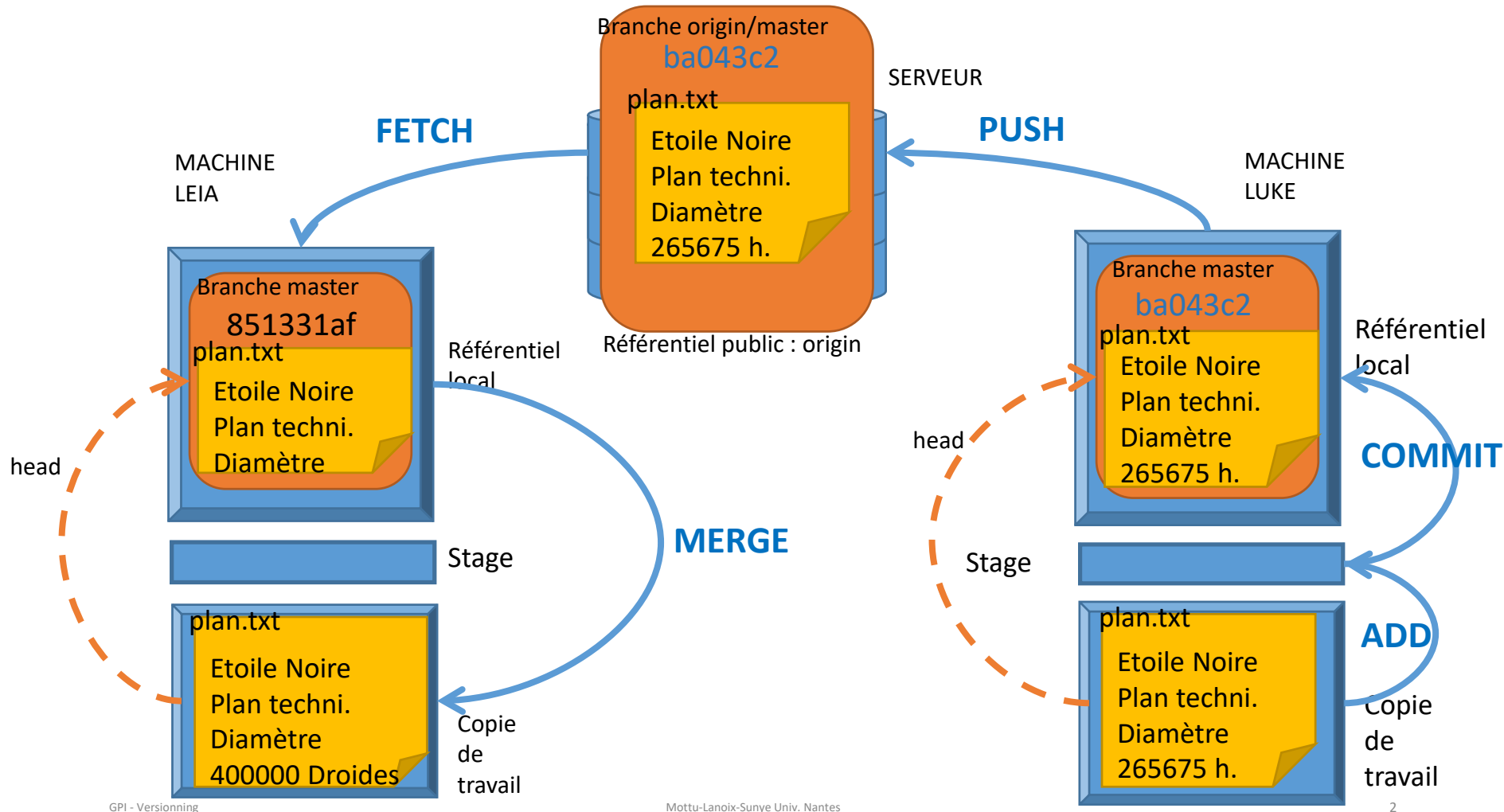
GIT rappel des notions (cf CM)

- Chacun versionne son travail sur sa machine puis le partage sur le serveur.
- La copie de travail (dans laquelle on écrit/travaille) pointe vers une version du référentiel local
- Quand on versionne son travail, on le committe vers le référentiel local (en passant par l'ajout en scène/stage)
- On pousse/push ensuite sa version sur le serveur pour le partage.
- Les autres acteurs tirent/pull (fetch-merge) ce qui est partagé sur le serveur vers leur référentiel local



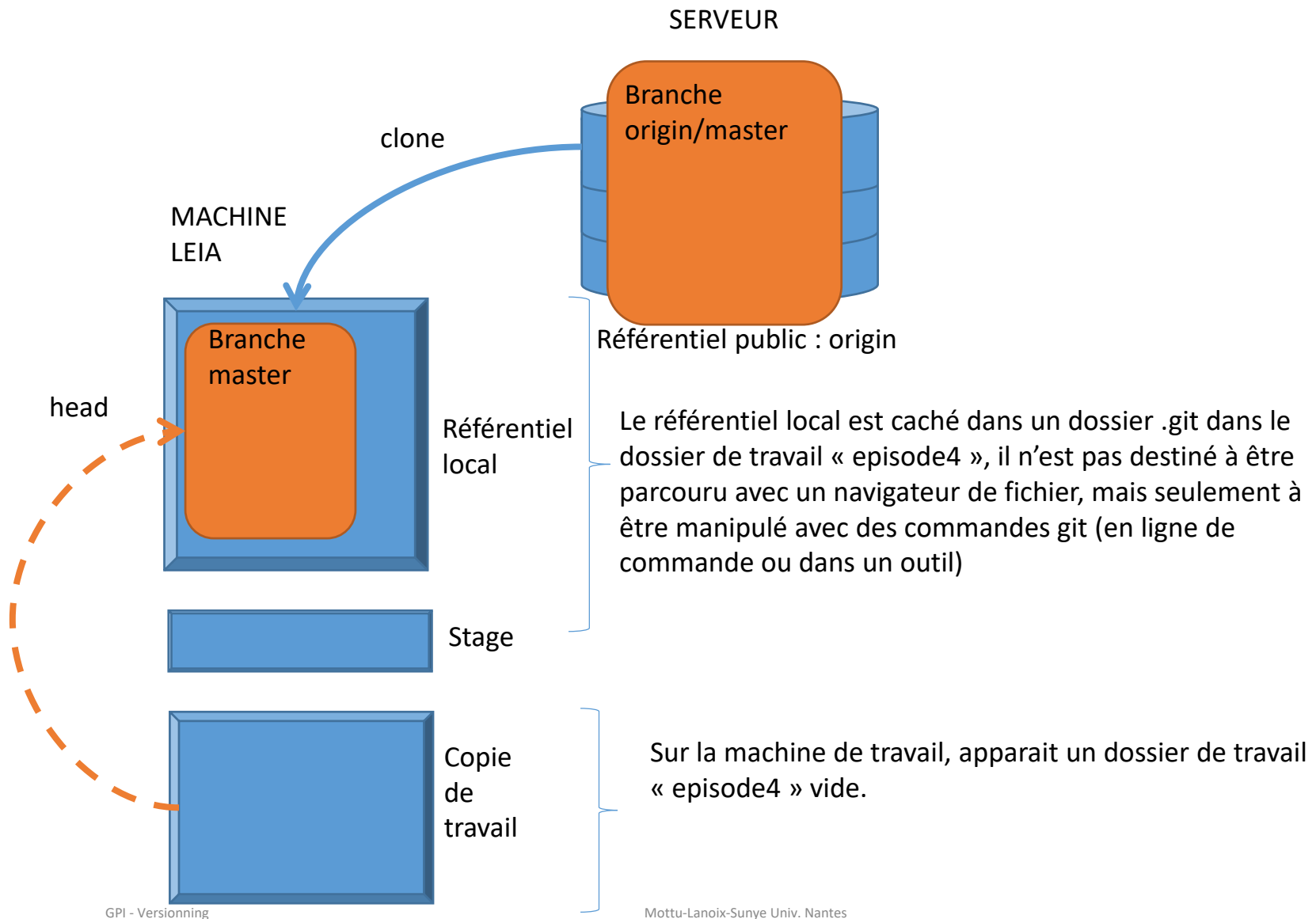
GIT rappel des notions (cf CM)

- Chacun versionne son travail sur sa machine puis le partage sur le serveur.
- La copie de travail (dans laquelle on écrit/travaille) pointe vers une version du référentiel local
- Quand on versionne son travail, on le committe vers le référentiel local (en passant par l'ajout/add en scène/stage)
- On pousse/push ensuite sa version sur le serveur pour le partage.
- Les autres acteurs tirent/pull (fetch-merge) ce qui est partagé sur le serveur vers leur référentiel local



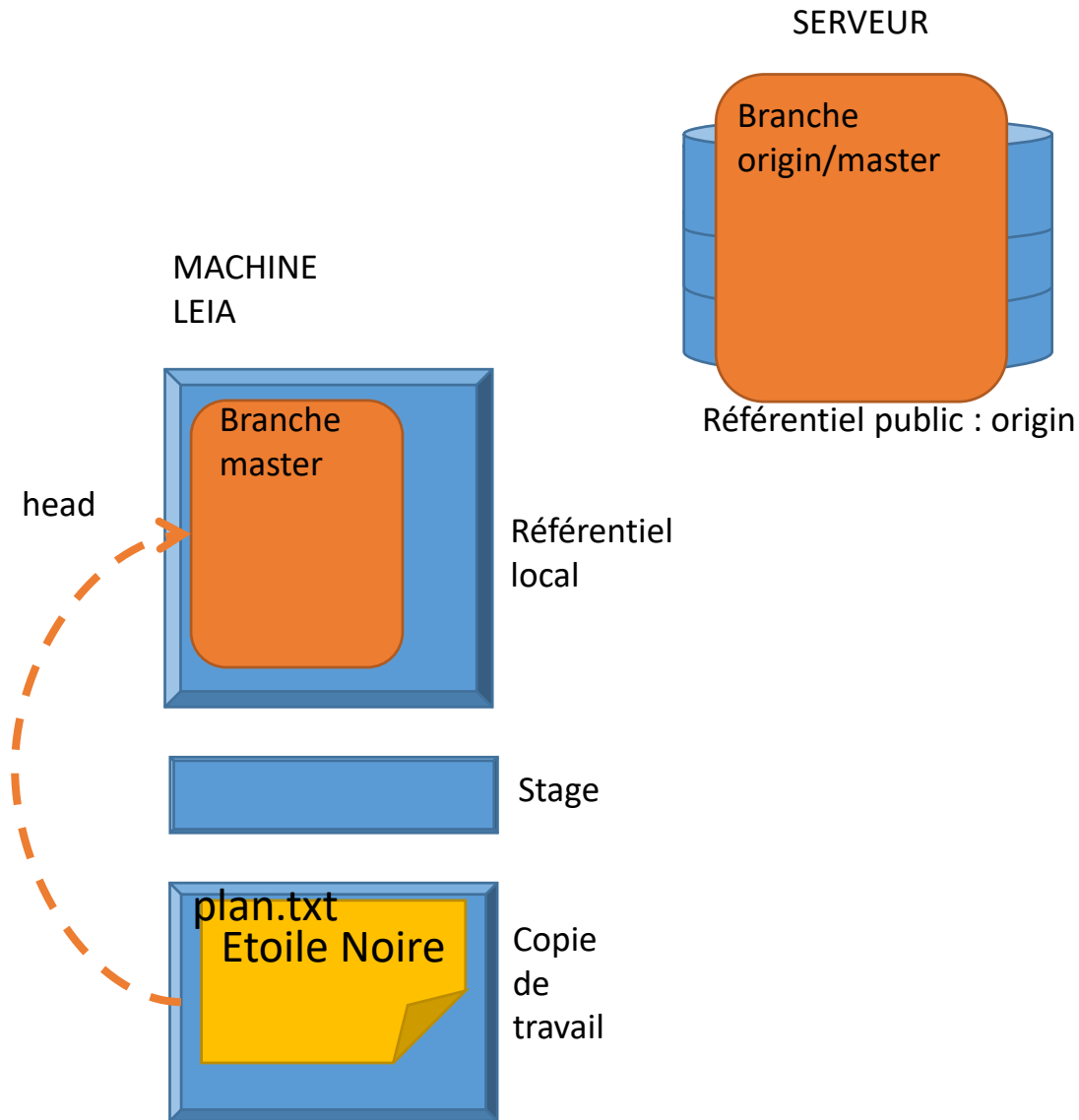
1 : git clone <https://gitlab.univ-nantes.fr/leila/episode4.git>

- On initialise sur la machine de Leila son référentiel local (repository) ayant la branche Master, et sa copie de travail.



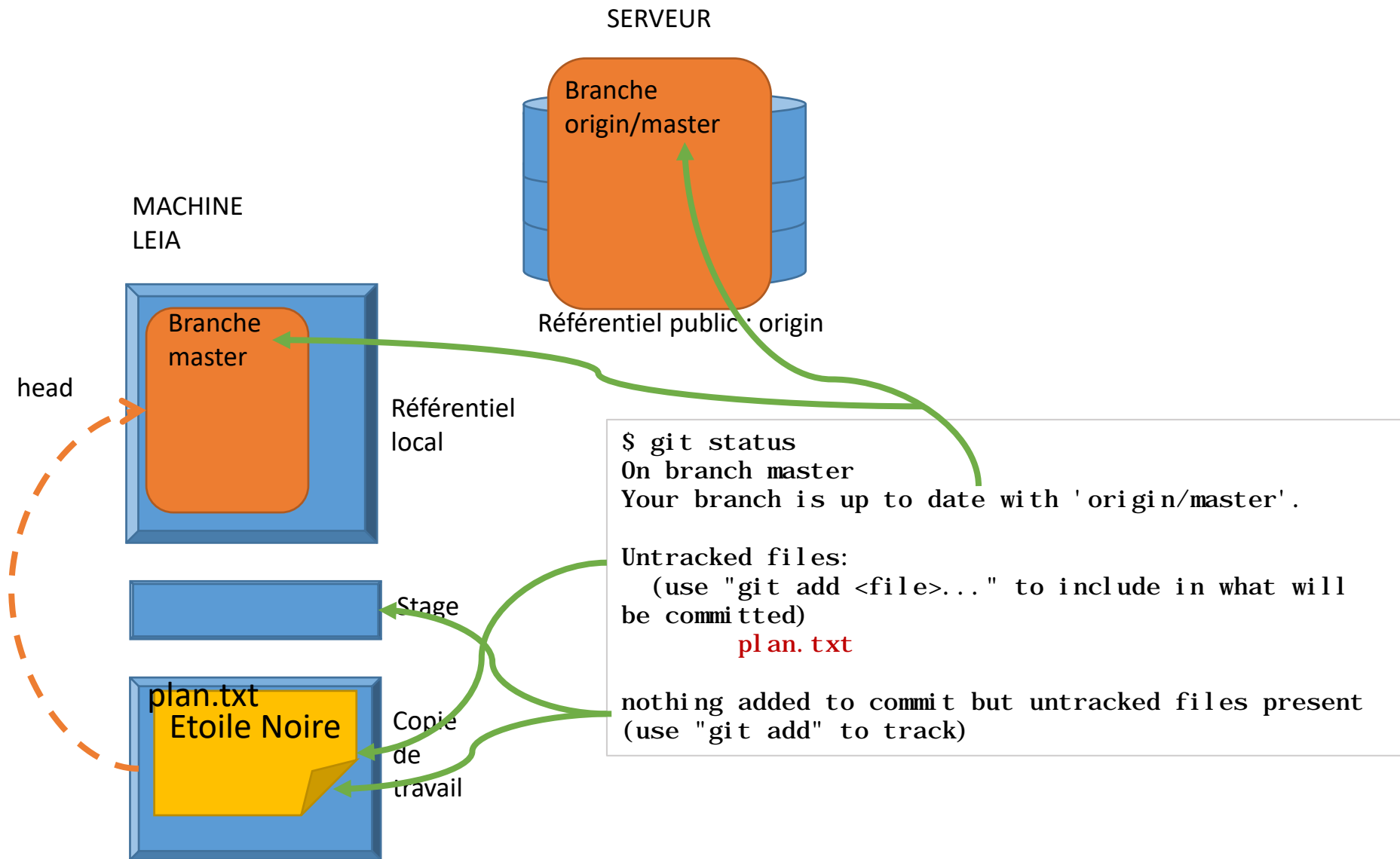
2. write(plan.txt, 1, Etoile Noire)

- Un fichier a été créé dans l'arborescence, localement, dans la copie de travail
 - C'est un travail effectué en local, avec n'importe quel outil, sans rapport avec git.



3. git status

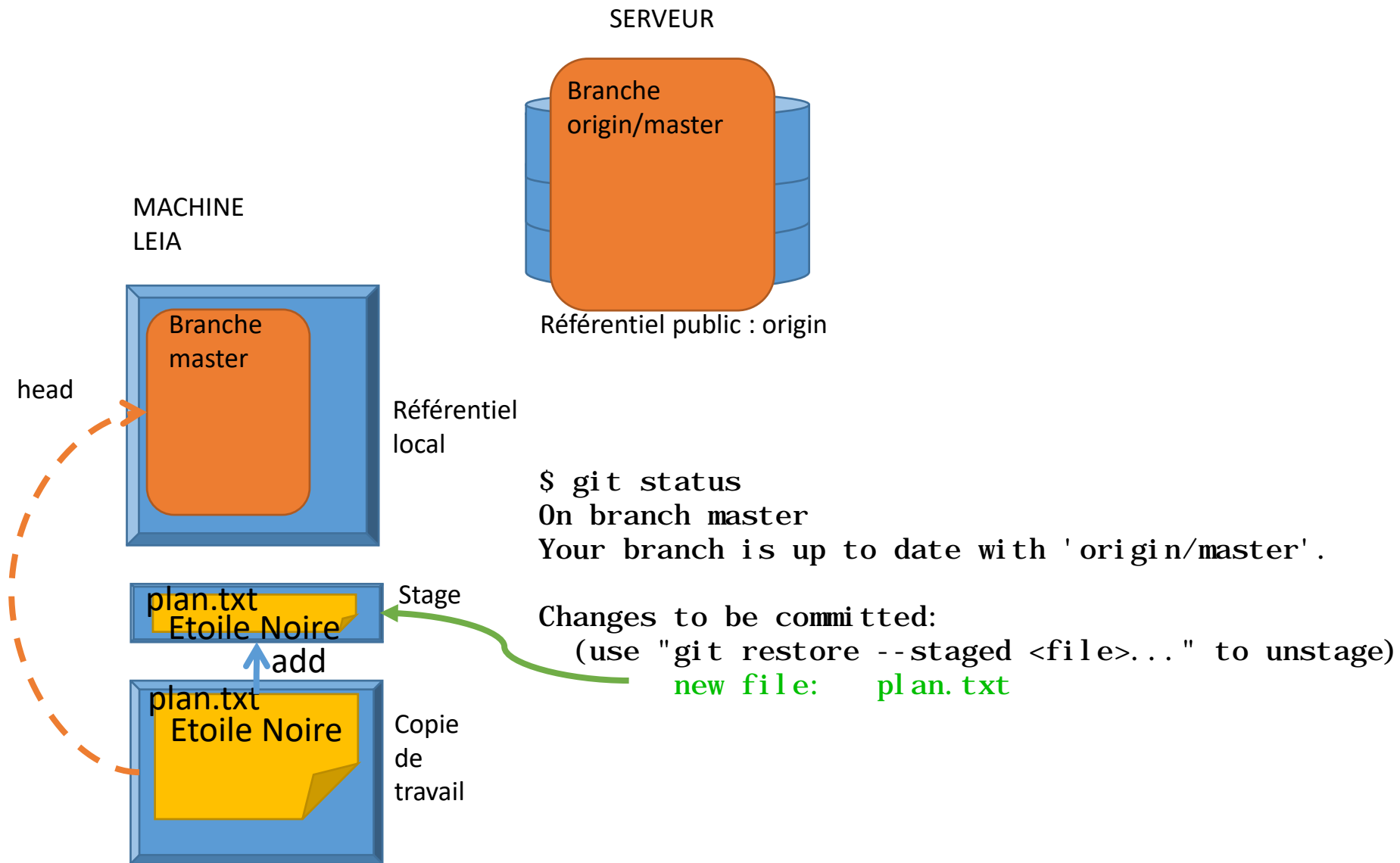
- Le référentiel local est à jour vis-à-vis du référentiel public
- Le fichier local n'est encore ni sauvegardé, ni versionné, ni partagé, ni même prêt à l'être



4. git add .

- On met le fichier en scène : on demande à ce que ces changements sont versionnés

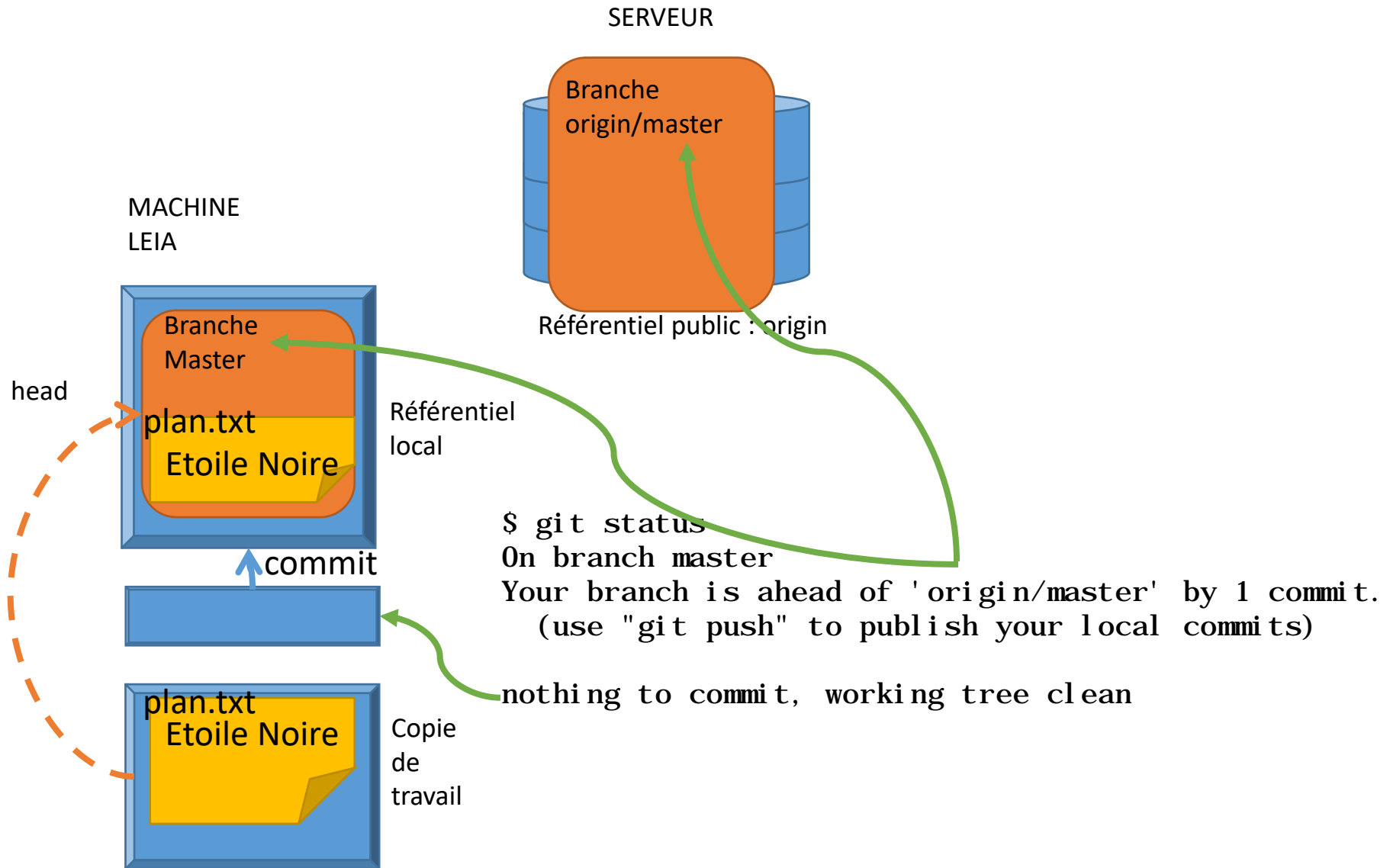
5. git status



6. git commit -m "creation du plan"

- Versionnage localement, les référentiels se distinguent

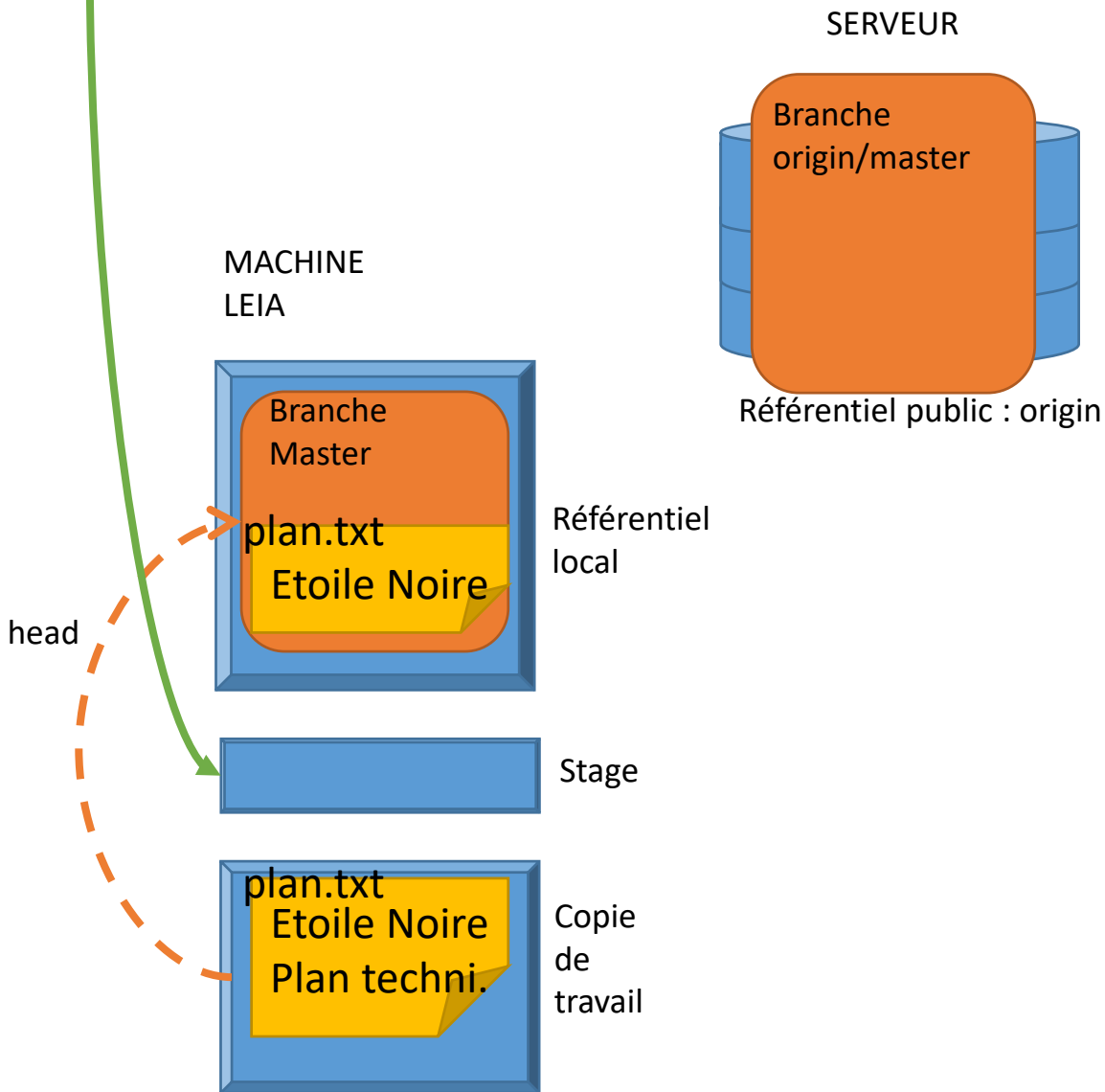
7. git status



8. Leia@linux : write(plan.txt, 2, Plan technique)

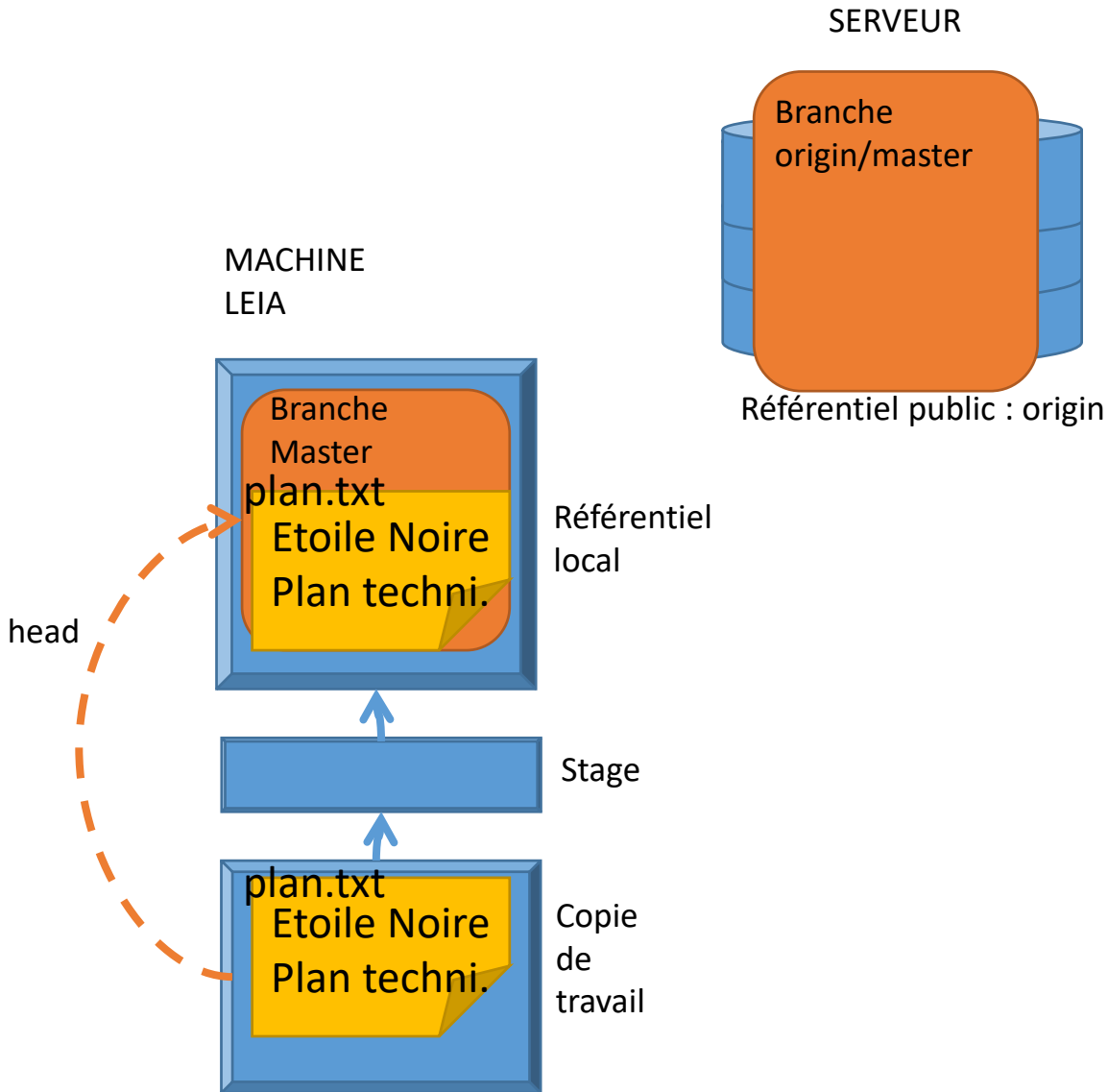
9. Leia@linux : git commit -m "Plan"

- Il n'y a rien en scène à commiter, car la scène est vide



10. Leia@linux : git add .
11. Leia@linux : git commit

- Ajout à l'index (la liste de ce qui est en scène) et commit (en ouvrant une fenêtre pour écrire le message obligatoire : bon courage avec l'éditeur vim (:wq pour quitter)

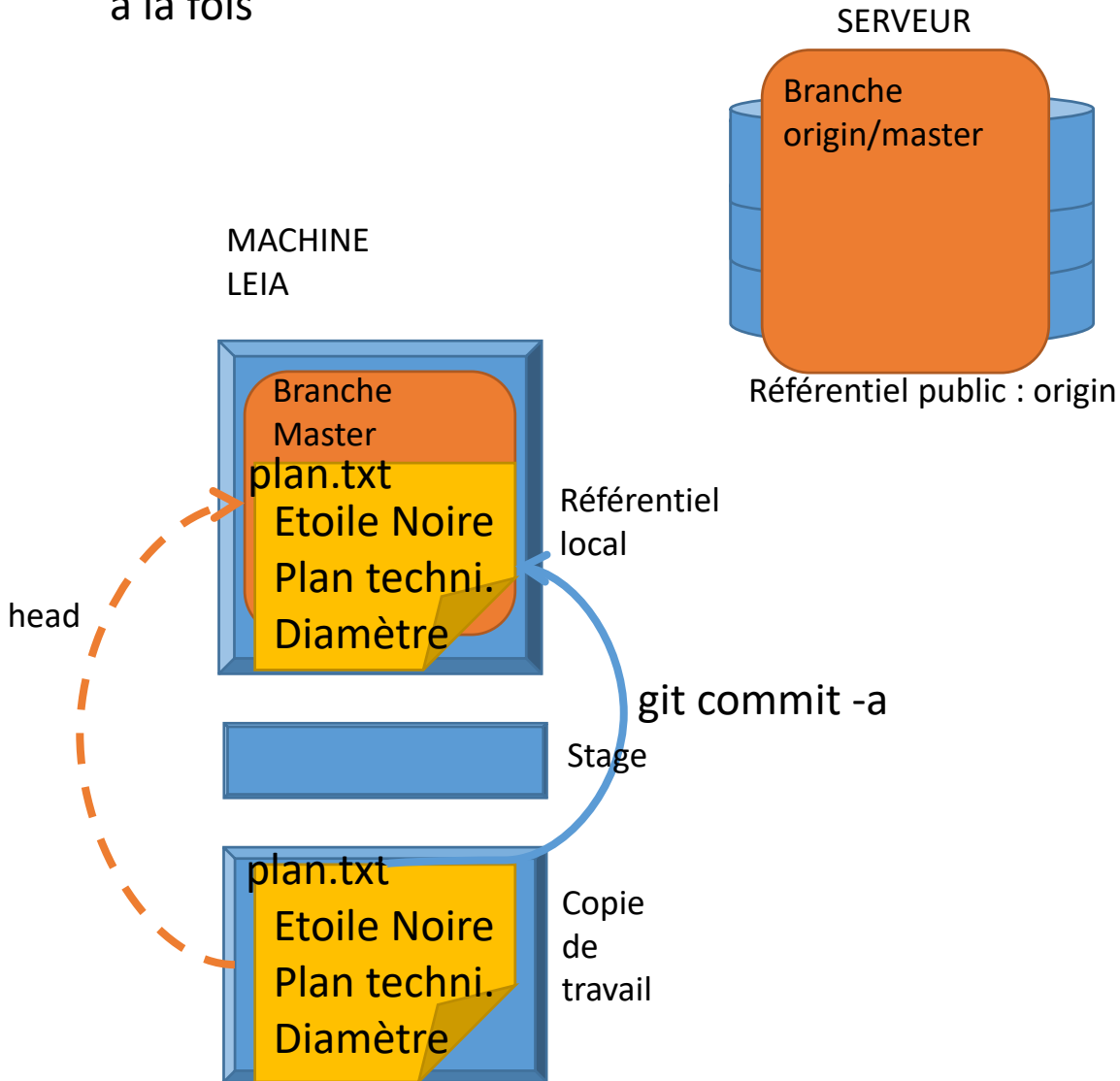


12. Leia@linux : write(plan.txt, 3, Diamètre : 120 kilomètres)

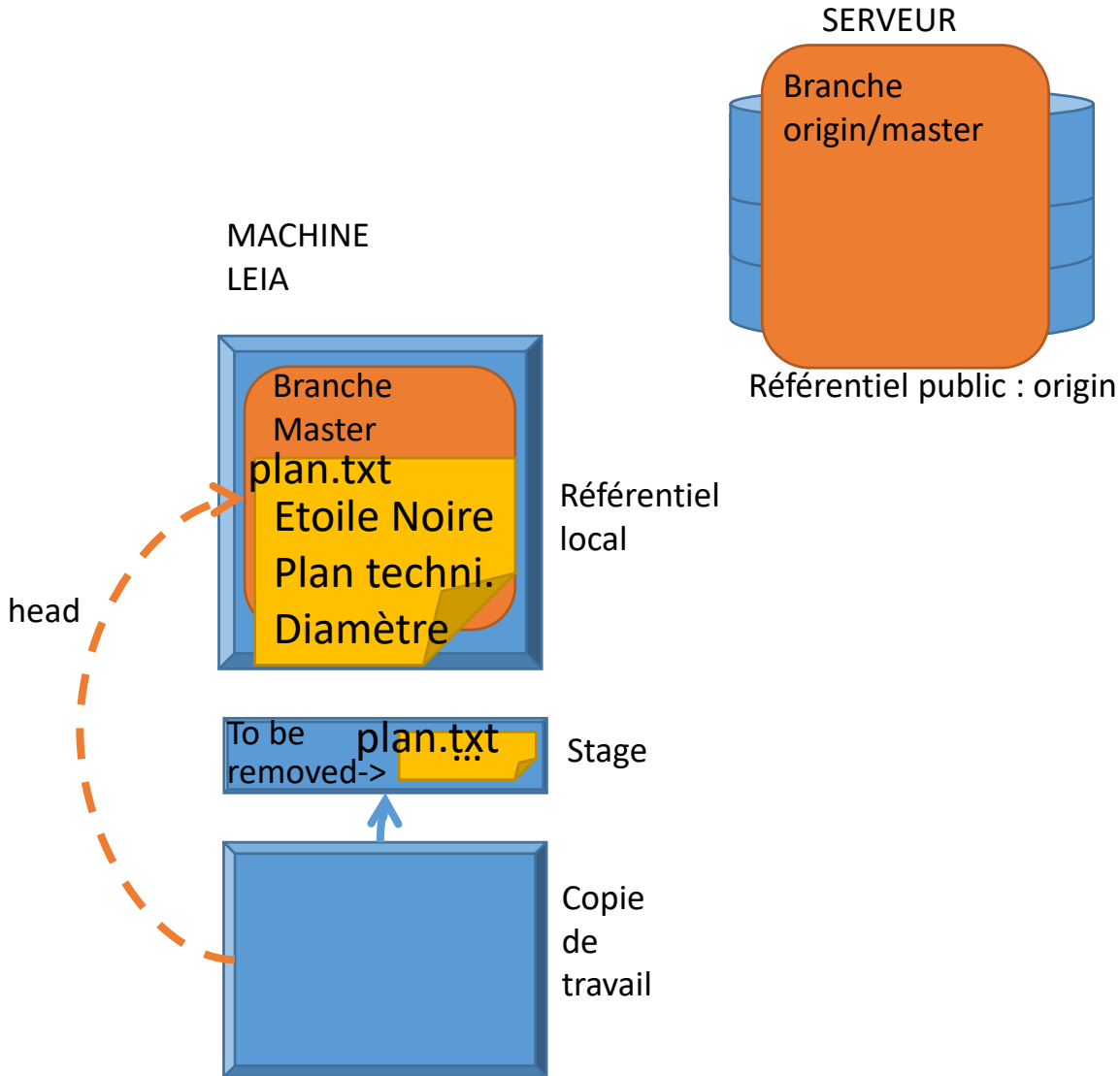
13. Leia@linux : git commit -a -m "diametre"

- Fait l'ajout et le commit

La scène est utile pour faire des « commits atomiques » : pas avec tous les fichiers modifiés à la fois

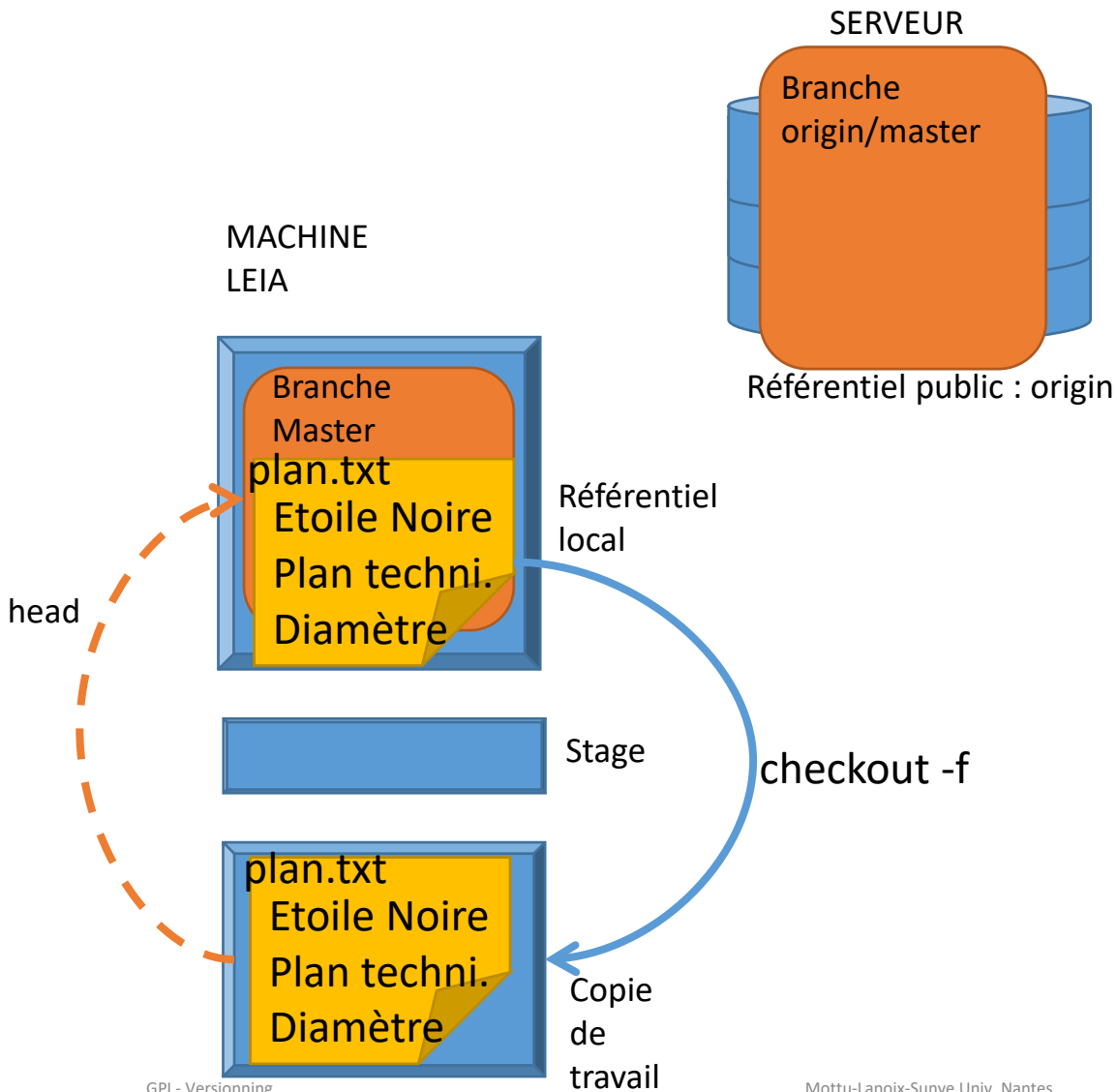


14. Leia@linux : git rm plan.txt -f
- Supprime le fichier de la copie de travail et met en scène sa suppression
15. Leia@linux : ls
- Pas une commande git, juste pour confirmer que le fichier est supprimé localement
16. Leia@linux : git status
- Il y a un fichier prêt à être supprimer, etc.

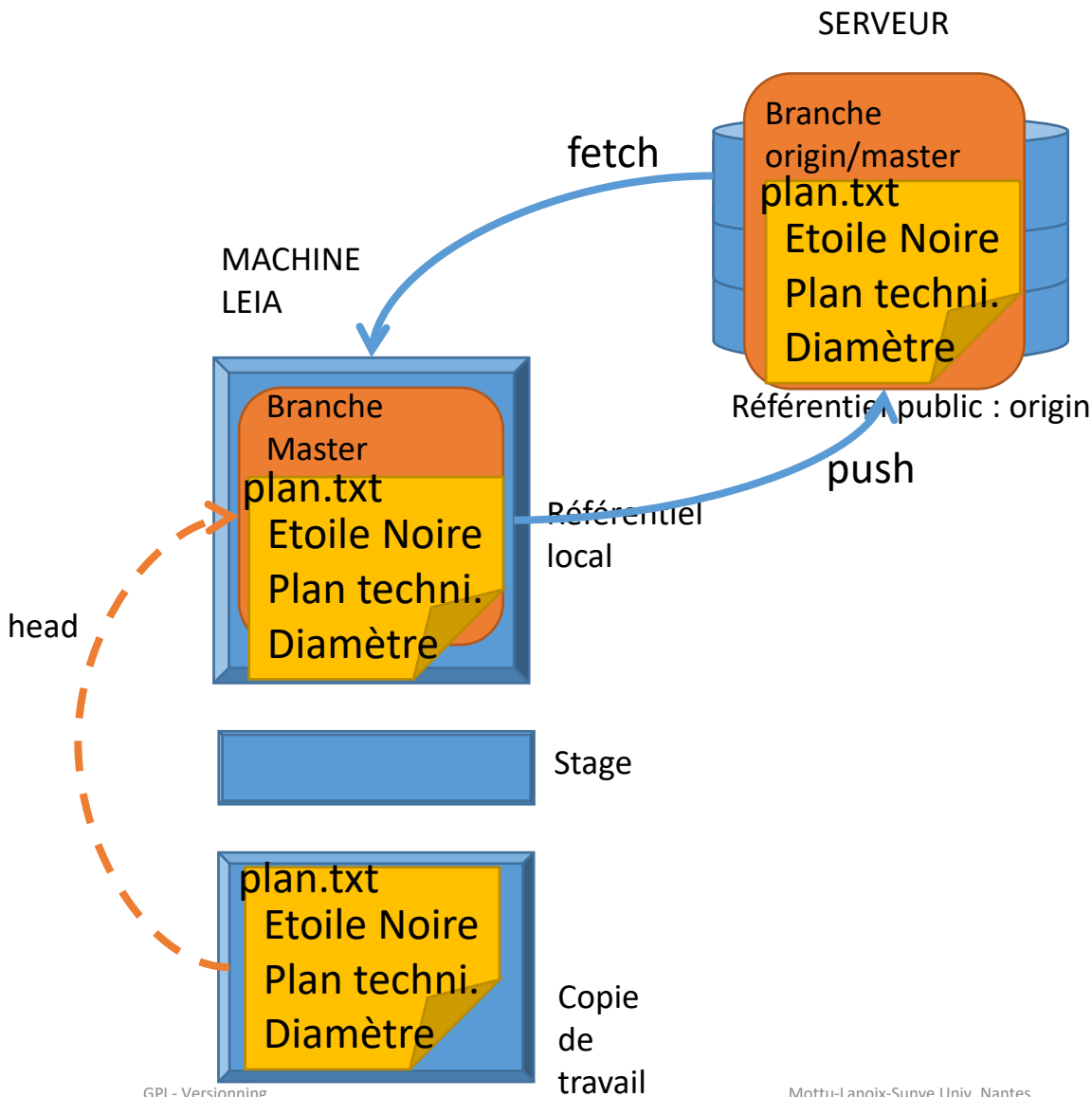


17. Leia@linux : git checkout -f

- Force de récupérer sur la copie de travail le contenu d'une branche (la branche pointée par head par défaut) : restaure la suppression
- Pour valider la suppression il aurait fallu commiter

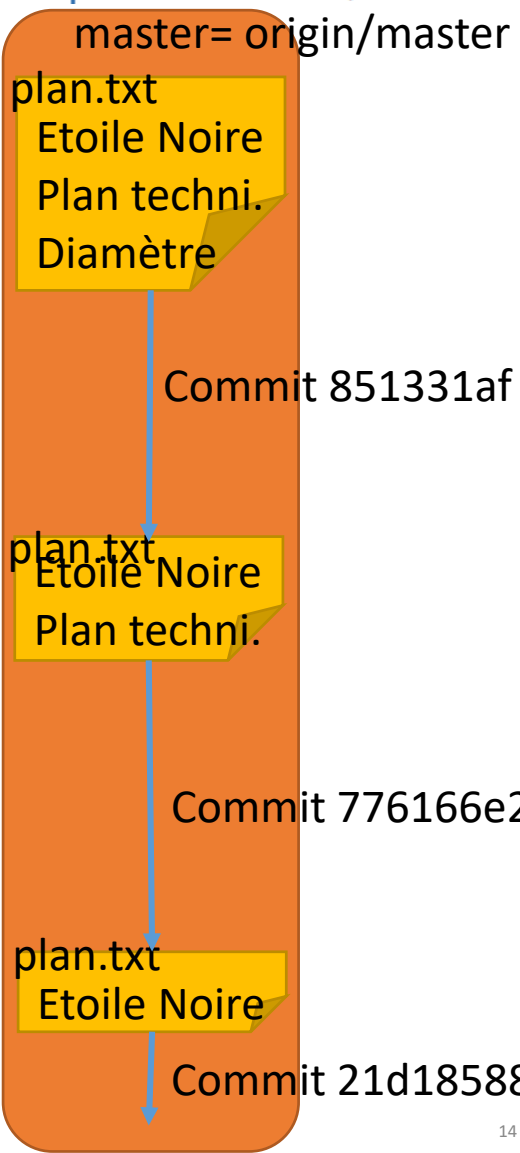


18. Leia@linux : git fetch • On vérifie qu'il n'y a rien à récupérer de nouveau du référentiel public
19. Leia@linux : git status. • Your branch is up to date with 'origin/master'.
20. Leia@linux : git push • On pousse son travail sur le référentiel public pour le partager



Mise au point sur les versions et le contenu des branches (les 2 branches étant maintenant les mêmes après cette première phase de travail)

Graph	Description	Date	Author	Commit
	diametre	11 mars 2021 22:52	Leia	851331a
	Plan	11 mars 2021 22:39	Leia	776166e
	creation du plan	11 mars 2021 22:19	Leia	21d1858



Commit 851331af

plan.txt		
1	1	Etoile Noire
2		- Plan technique
		\ No newline at end of file
	2	+ Plan technique
	3	+ Diamètre : 120 kilomètres

Commit 776166e2

plan.txt		
1		- Etoile Noire
		\ No newline at end of file
	1	+ Etoile Noire
	2	+ Plan technique

Commit 21d18588

plan.txt 0 → 100644		
	1	+ Etoile Noire

21. Luke@macosx : git clone https://gitlab.univ-nantes.fr/Leia/episode4.git

22. \$ git log -1 --abbrev-commit

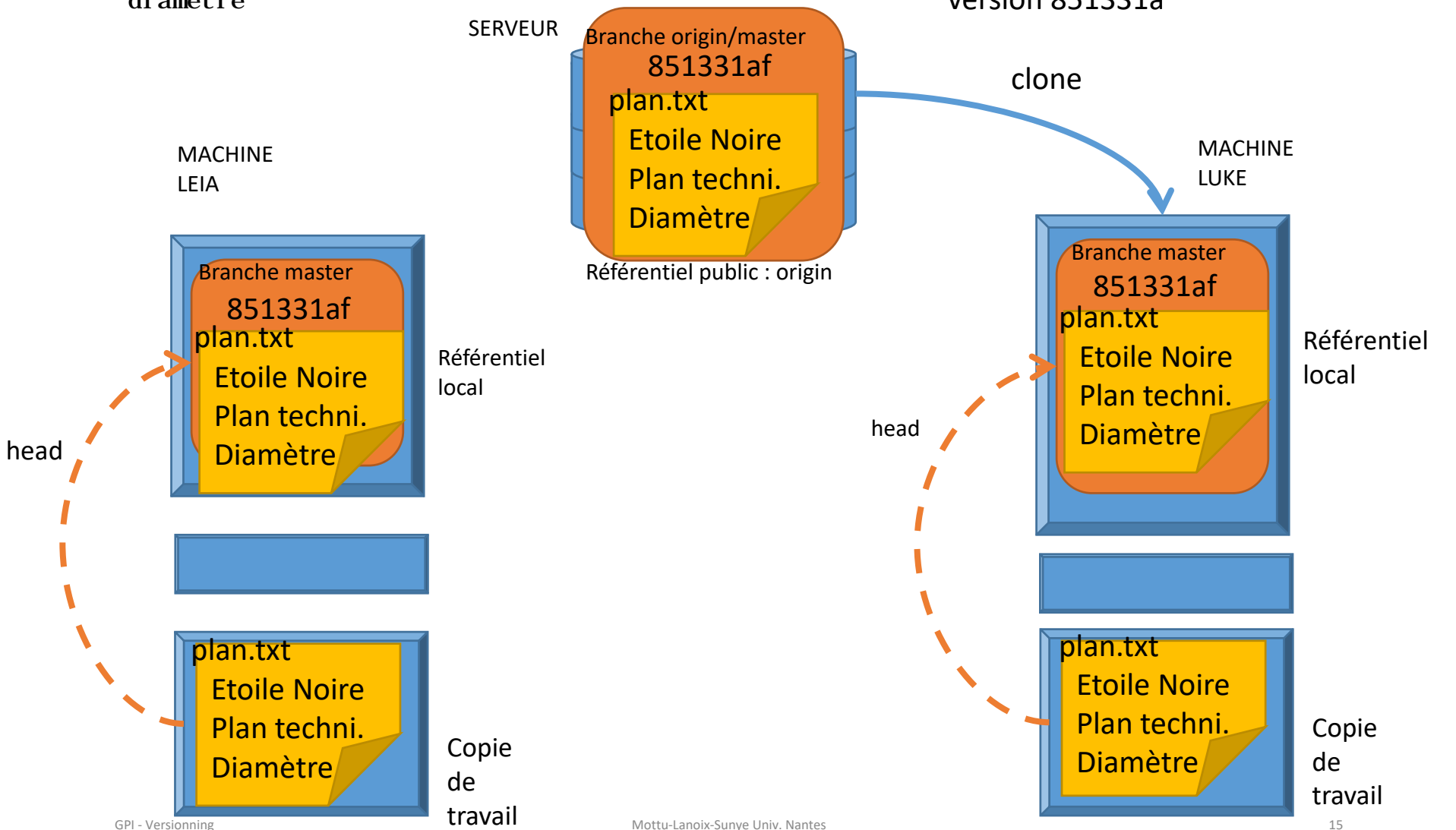
commit 851331a (HEAD -> master, origin/master, origin/HEAD)

Author: leia

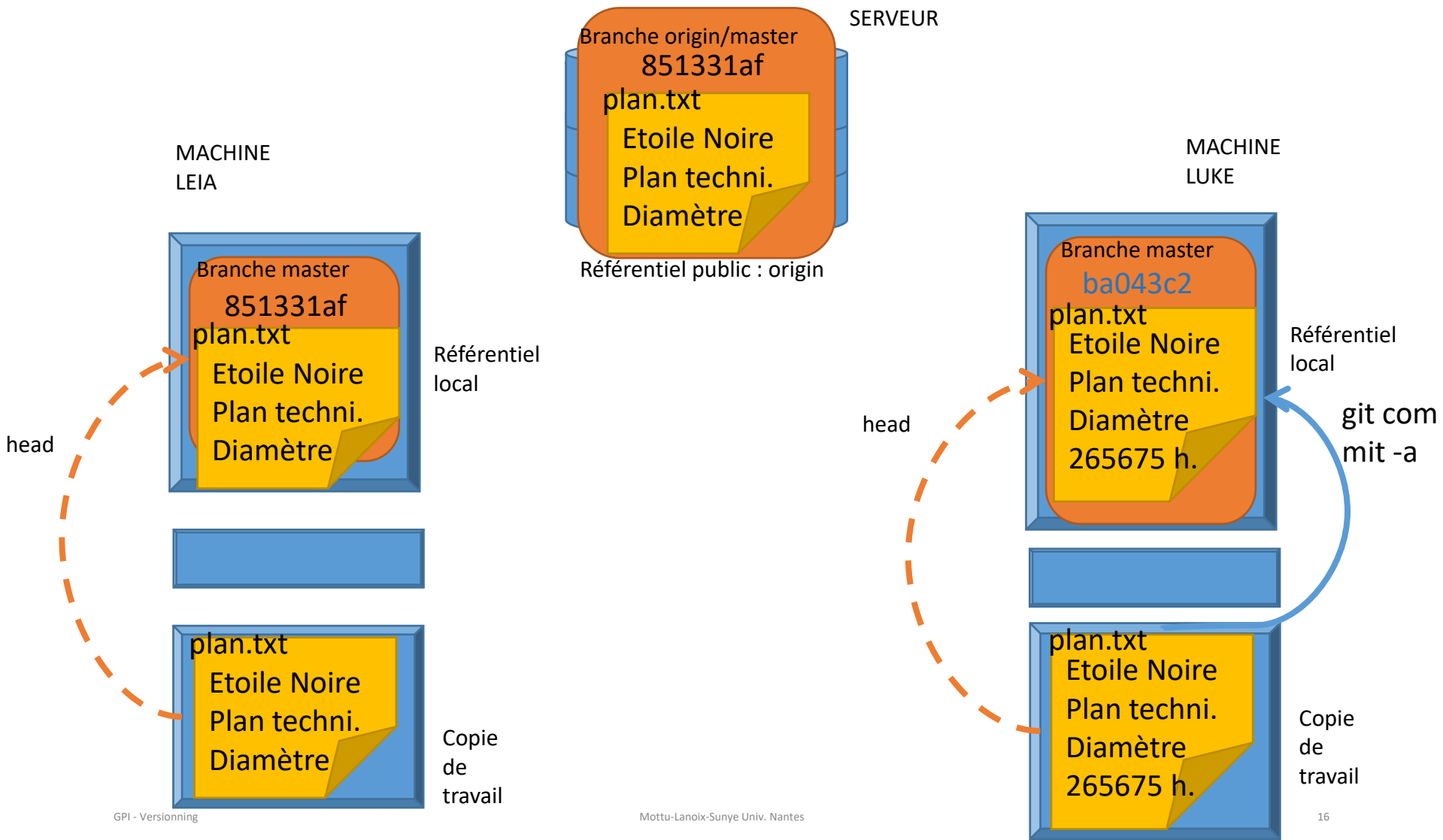
Date: Thu Mar 11 22:52:02 2021 +0100

Affiche 1 commit en abrégant le numéro
Le head pointe vers la branche master,
synchro avec origin, du commit de leia,
et toutes les branches sont à cette
version 851331a

di ametre

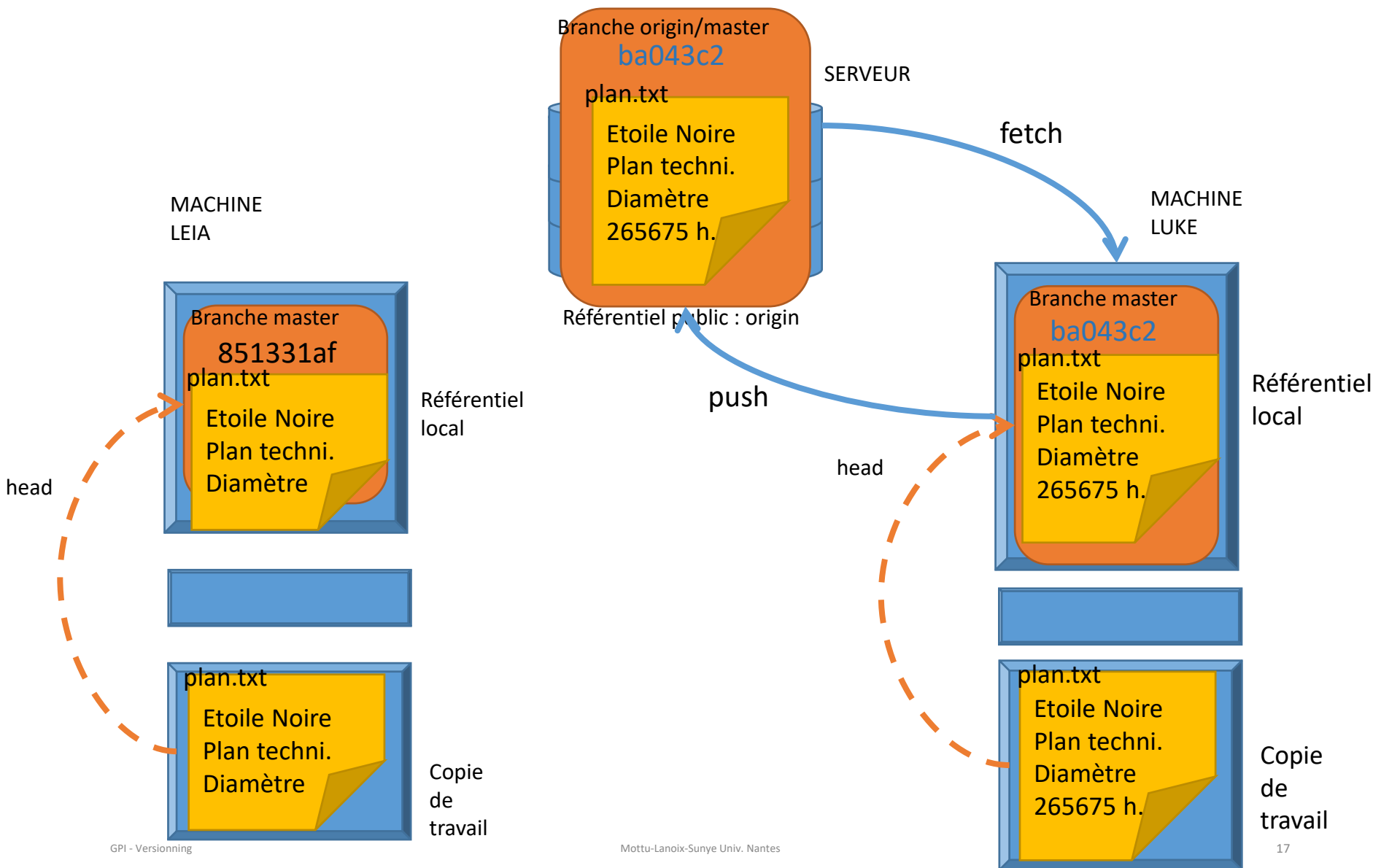


23. Luke@macosx : write(plan.txt, 4, Equipage : 265675 hommes)
24. Luke@macosx : git commit -a -m "equipage"
- Effectue le commit ba043c2

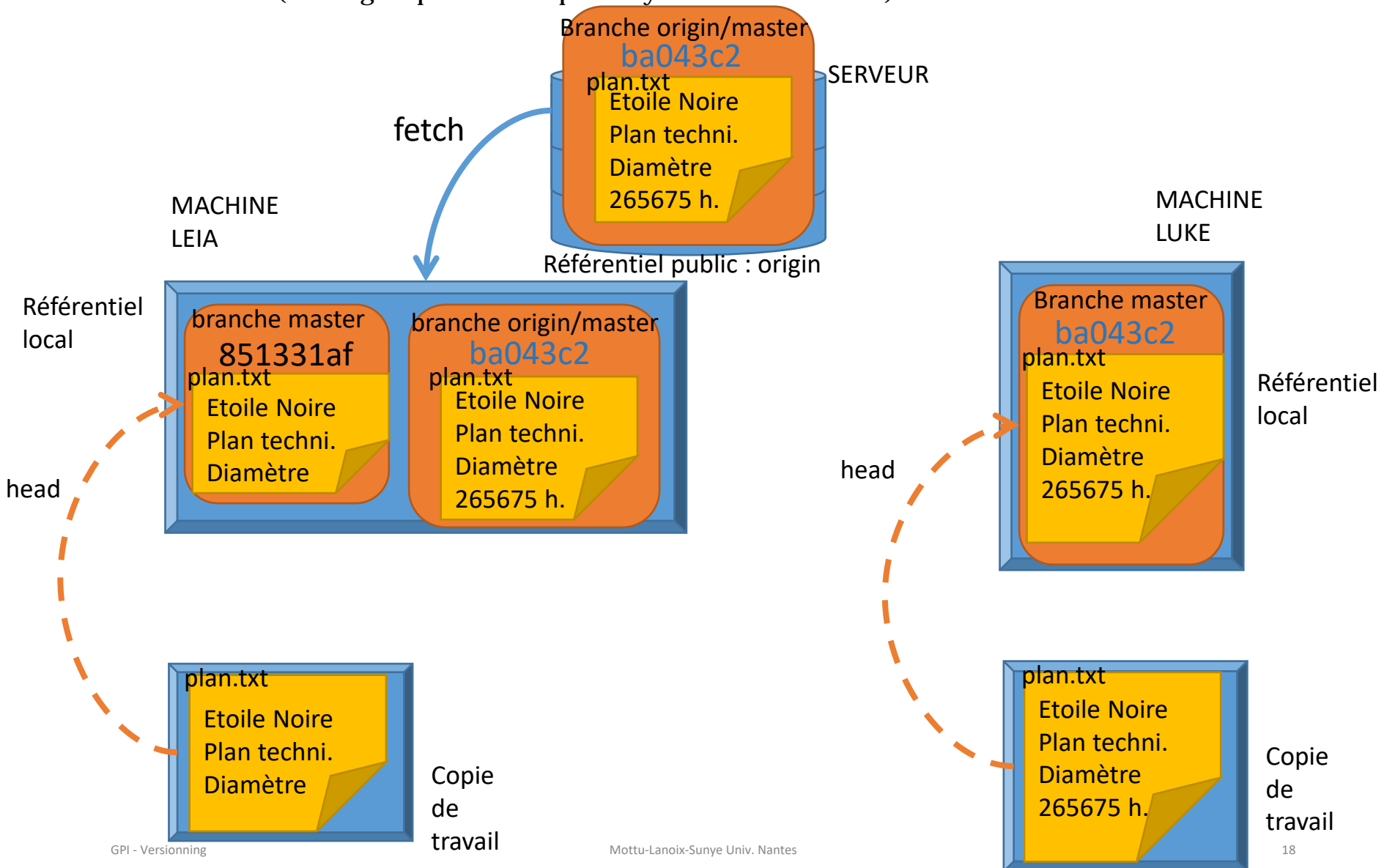


25. Luke@macosx : git fetch
26. Luke@macosx : git status
27. Luke@macosx : git push

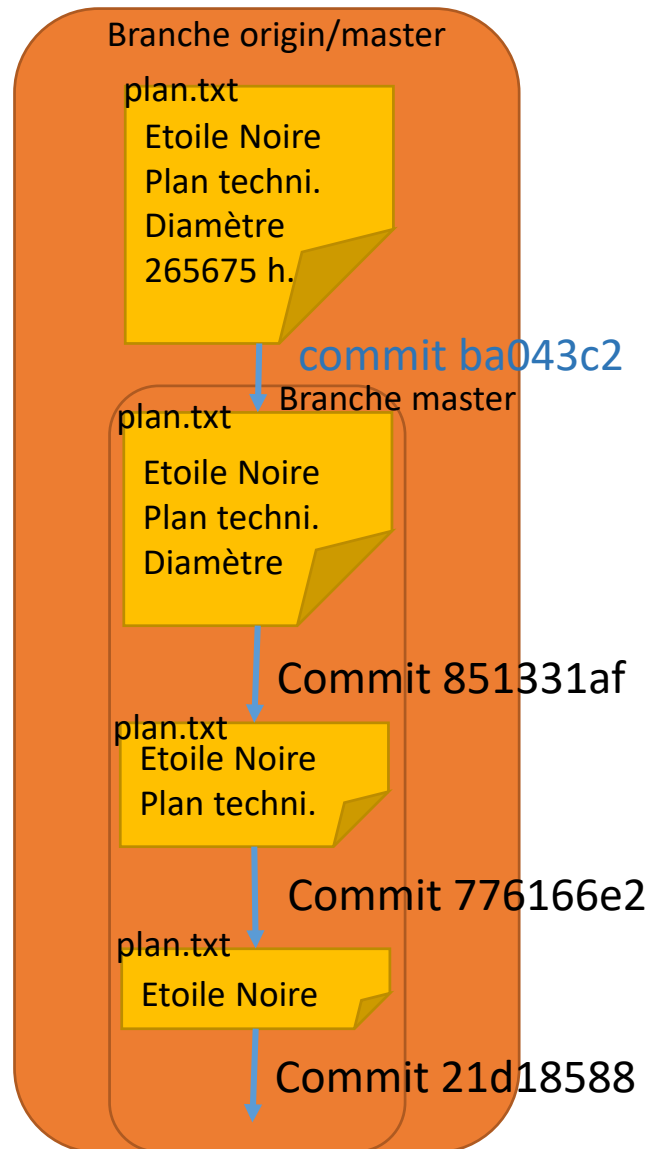
- On vérifie qu'il n'y a rien à récupérer de nouveau du référentiel public
- Your branch is up to date with 'origin/master'.
- On pousse son travail sur le référentiel public pour le partager



28. Leia@linux : git fetch
- Avant de travailler, on vérifie s'il y a du nouveau sur référentiel public
29. Leia@linux : \$ git status
- On branch master
- Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)



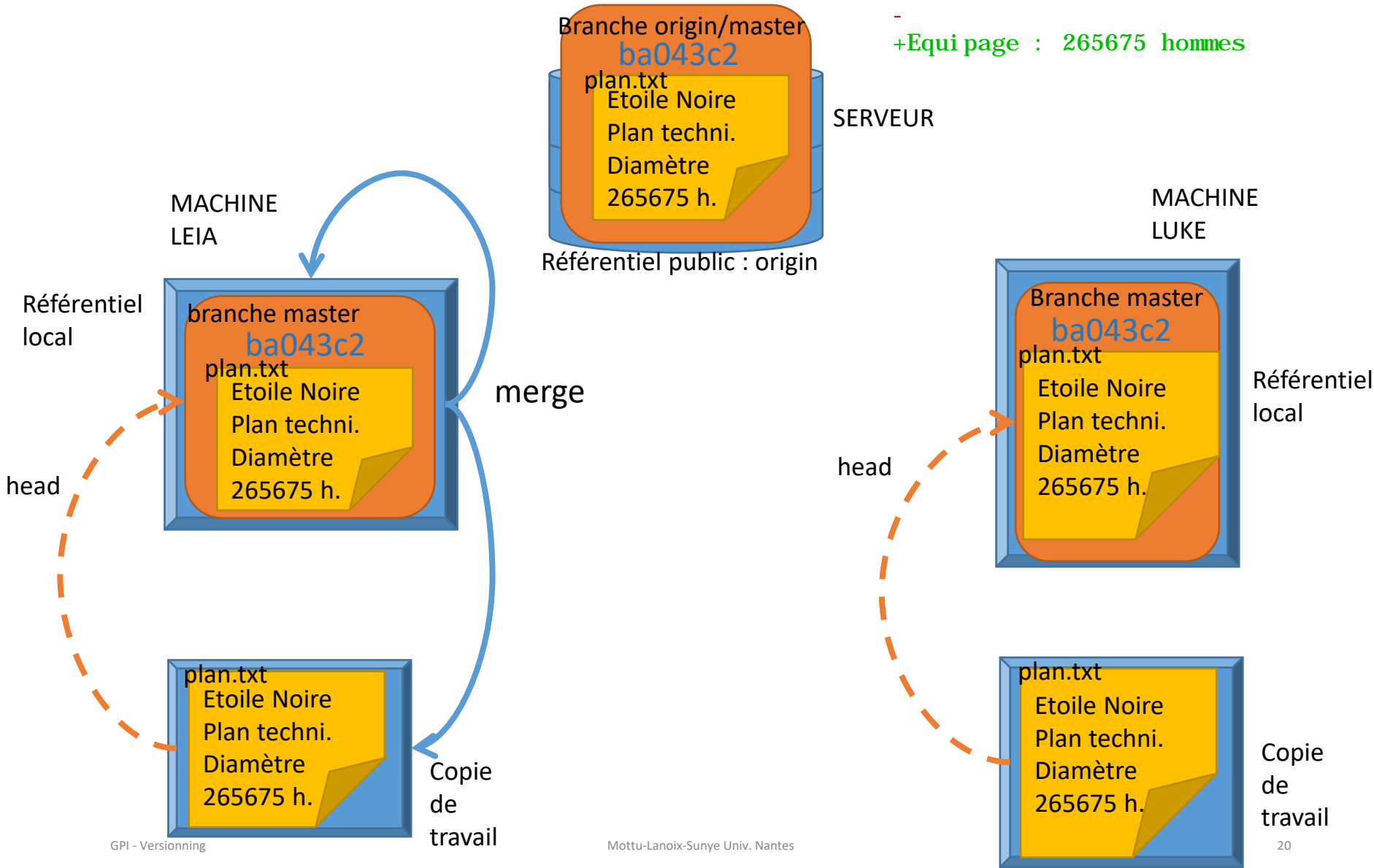
Plus en détail, ce qu'on a après un fetch sur le référentiel local de Leïa:
la branche origin/master est complète,
mais la branche master a un commit de décalage



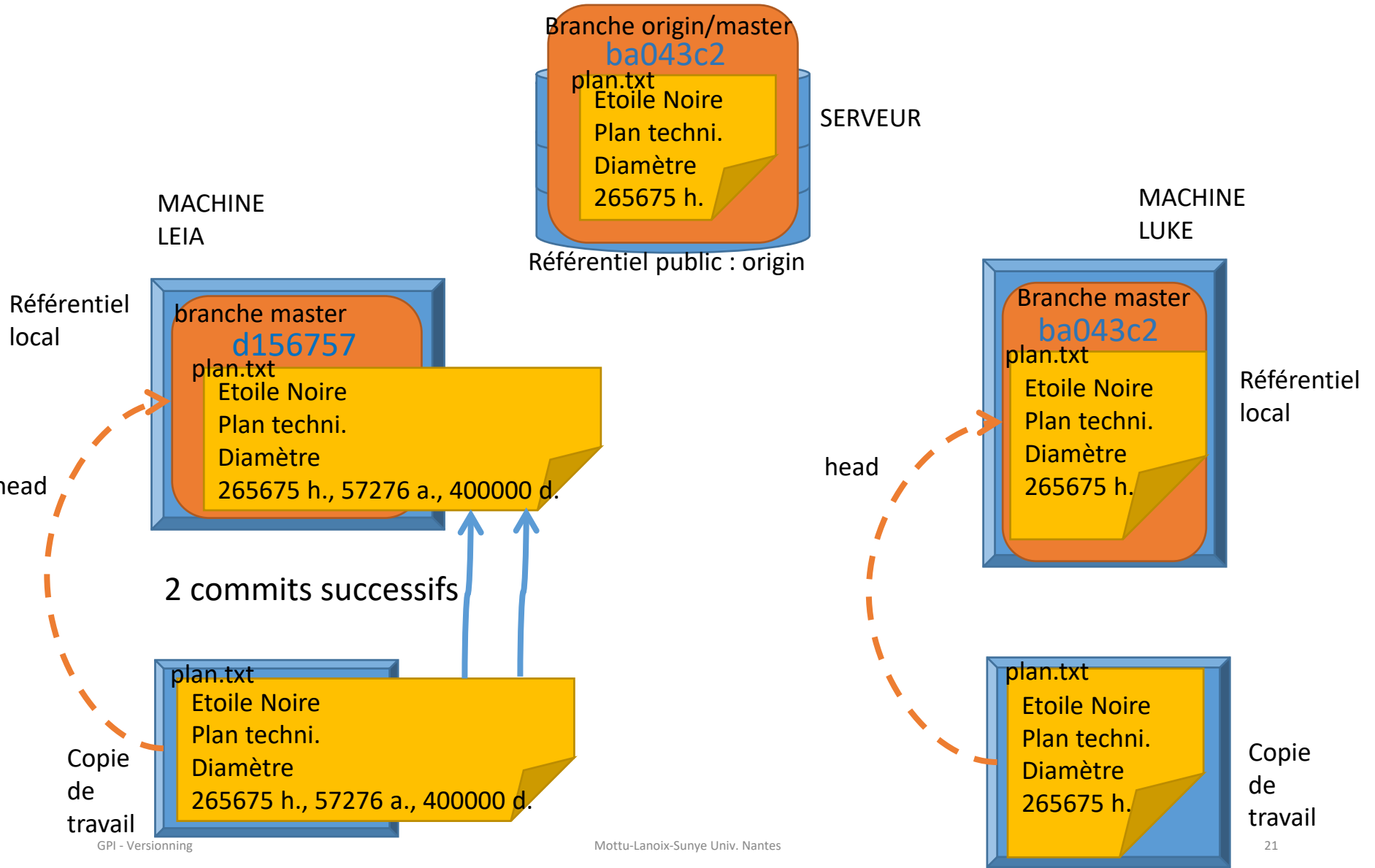
30. Leia@linux : git diff master origin/master • On compare pour vérifier si les nouveautés :
31. Leia@linux : git merge

- En local, la branche master est maintenant aussi complète que origin/master

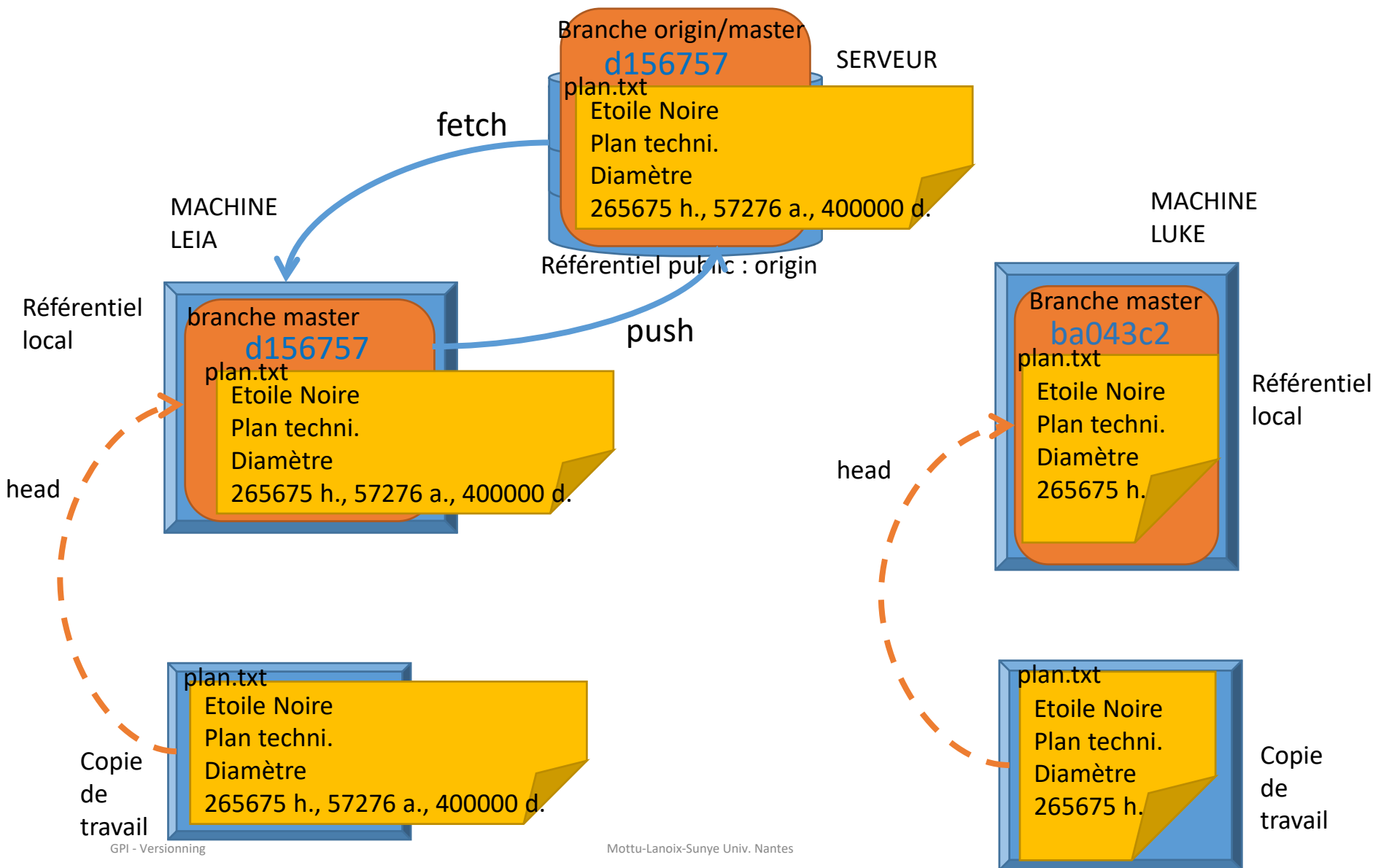
@@ -1,7 +1,7 @@
Etoile Noire
Plan technique
Diamètre : 120 kilomètres
-
+Equipage : 265675 hommes



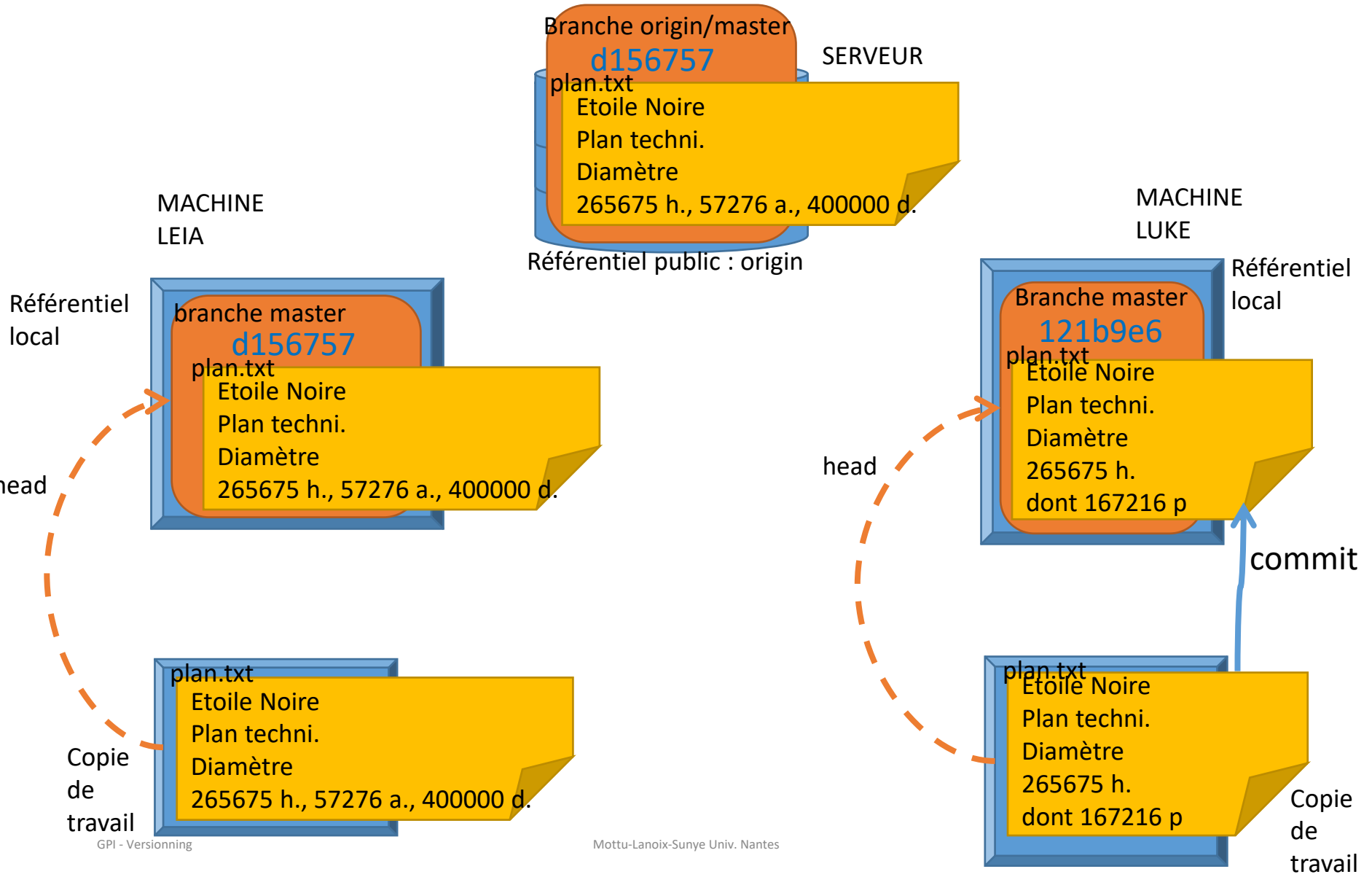
- 32. Leia@linux : write(plan.txt, 4, Equipage : 265675 hommes, 57276 artilleurs)
- 33. Leia@linux : git commit -a -m "MAJ Equipage"
- 34. Leia@linux : write(plan.txt, 4, Equipage : 265675 hommes, 57276 artilleurs, 400000 droides)
- 35. Leia@linux : git commit -a -m "MAJ Equipage"



36. Leia@linux : git fetch • On vérifie qu'il n'y a rien à récupérer de nouveau du référentiel public
37. Leia@linux : git status. • Your branch is up to date with 'origin/master'.
38. Leia@linux : git push • On pousse son travail sur le référentiel public pour le partager

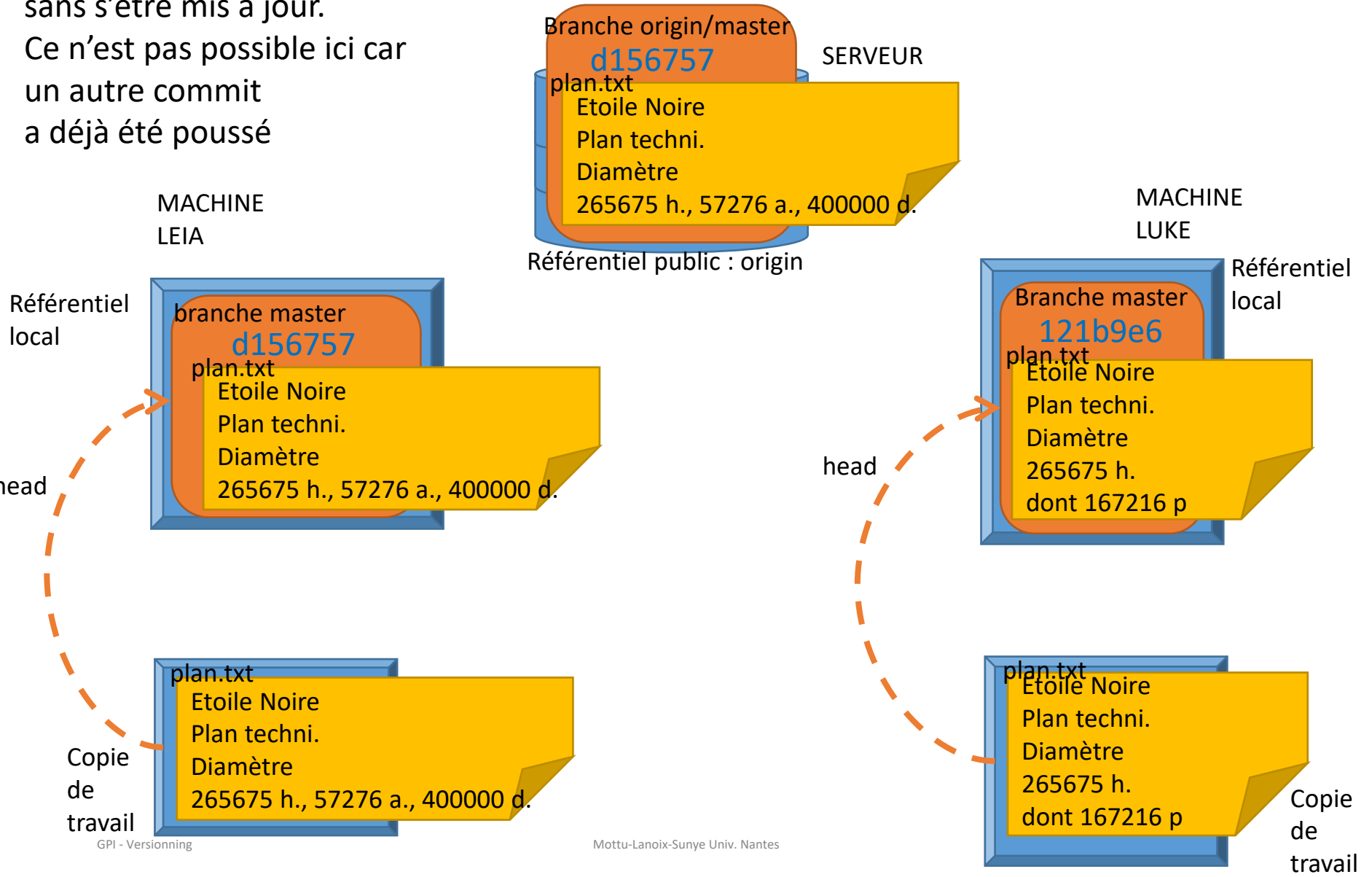


- 39. Luke@macosx : write(plan.txt, 5, dont 167216 pilotes)
- 40. Luke@macosx : git commit -a -m "MAJ equipage"



41. Luke@macosx : git push
! [rejected]

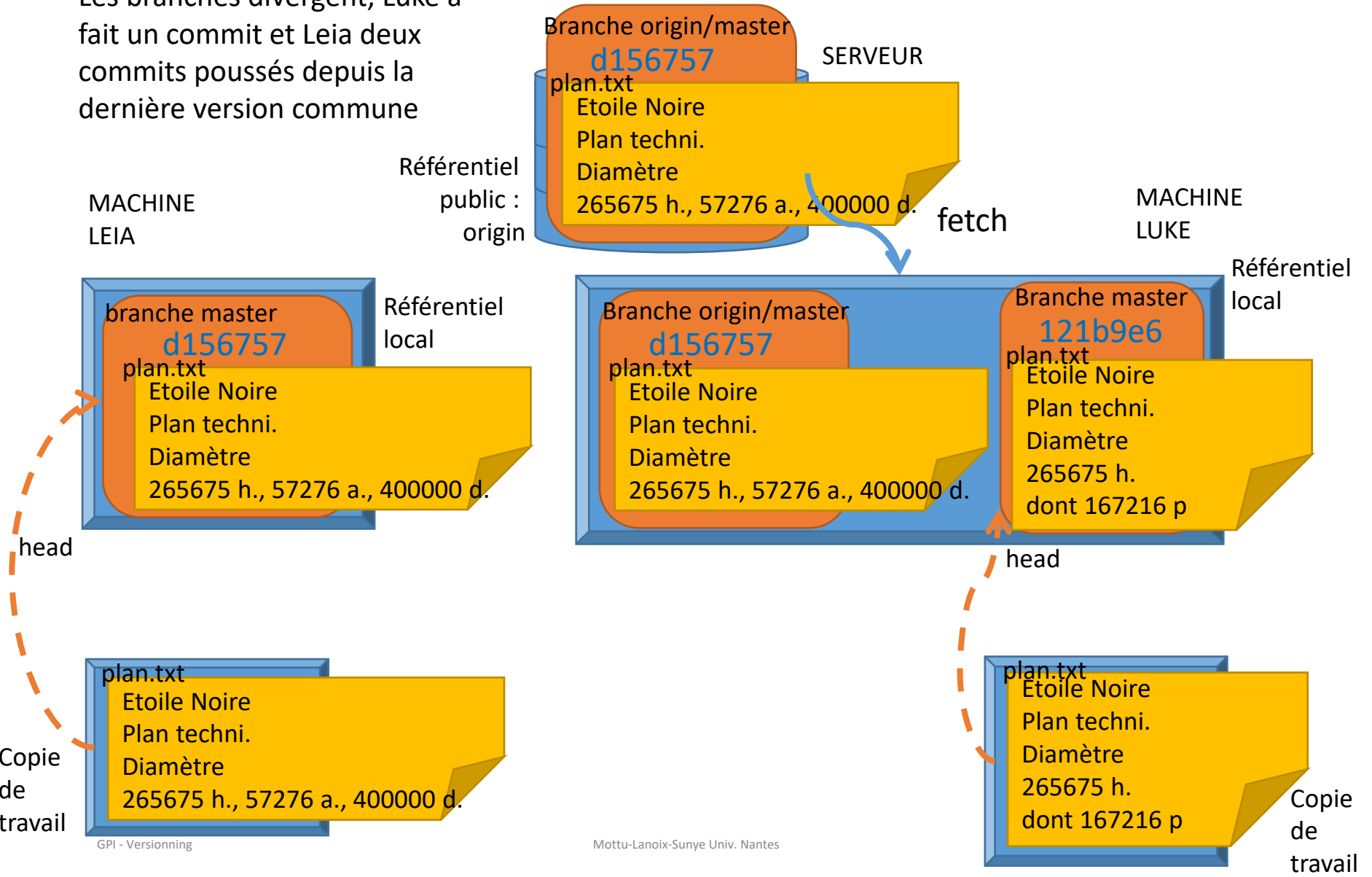
Il ne faut pas pousser
sans s'être mis à jour.
Ce n'est pas possible ici car
un autre commit
a déjà été poussé



42. Luke@macosx : git fetch
43. Luke@macosx : git status

- Les branches divergent, Luke a fait un commit et Leia deux commits poussés depuis la dernière version commune

```
$ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 2 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)
```



Plus en détail, ce qu'on a après un fetch sur le référentiel local de Luke : on a bien un graphe avec 2 branches qui se sont séparées.

Restent maintenant à les regrouper.

44. Luke@macosx :
git diff master origin/master

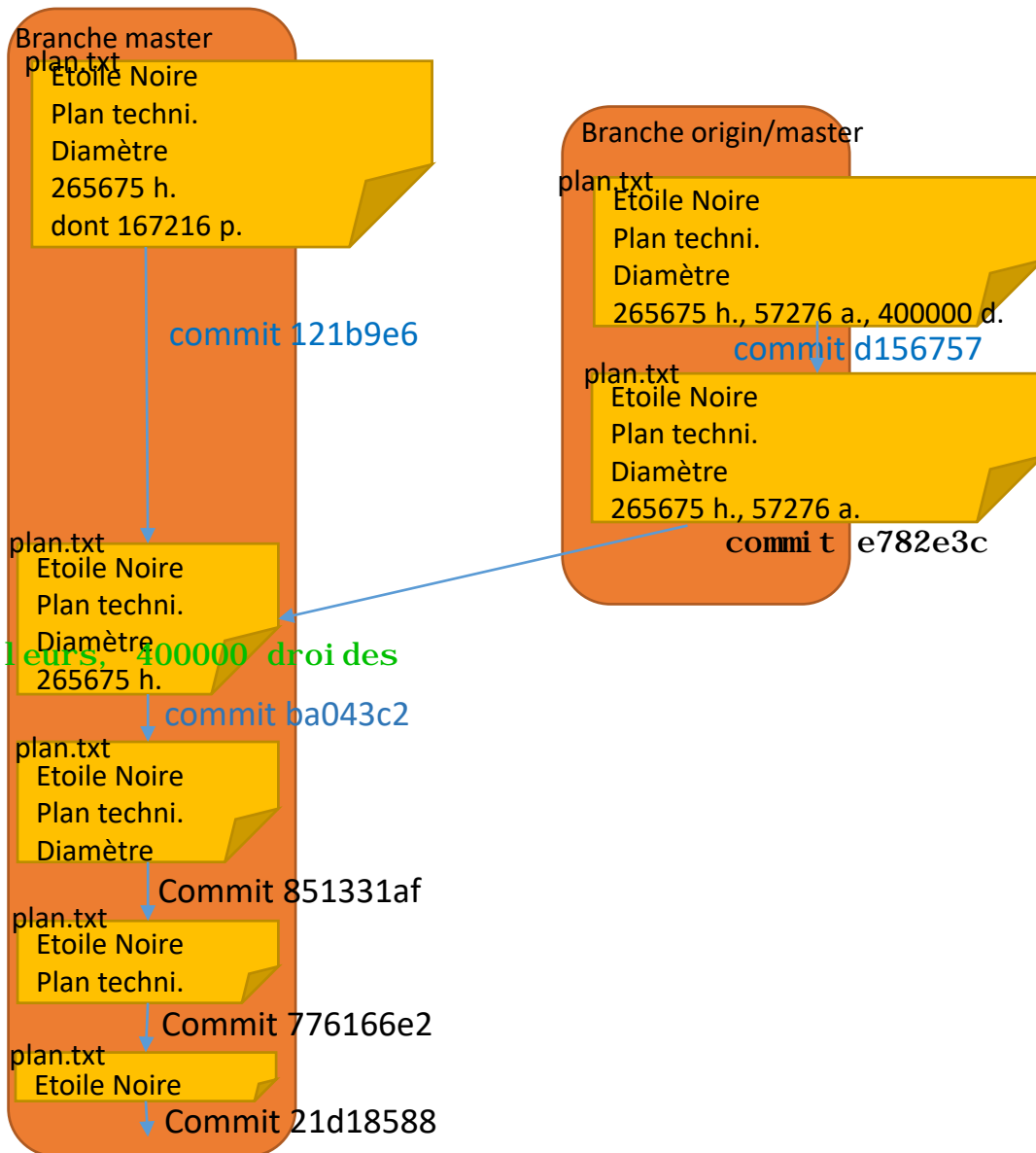
@@ -1,5 +1,4 @@

Etoile Noire
Plan technique
Diamètre : 120 kilomètres
-Equipage : 265675 hommes
-dont 167216 pilotes
+Equipage : 265675 hommes, 57276 artilleurs, 400000 droïdes

Cela peut étonner qu'on considère que la ligne « Equipage » est enlevée-ajoutée et pas juste modifiée, mais c'est une sécurité importante : attention en travaillant à plusieurs sur un même fichier.



Ca peut passer avec des changements éloignés dans un fichier.
Mais c'est une fausse sécurité : modifier un bout de code a de l'influence programmatiquement



45. Luke@macosx : git merge

Auto-merging plan.txt

CONFLICT (content): Merge conflict in plan.txt

Automatic merge failed; fix conflicts and then commit the result.

Puisqu'une ligne est considérée ajoutée-modifiée cela crée un conflit et produit un fichier avec les 2 versions:

46. Luke@macosx : résoudre le conflit dans un éditeur

```
Etoile Noire
Plan technique
Diamètre : 120 kilomètres
<<<<<<< HEAD
Equipage : 265675 hommes
dont 167216 pilotes
=====
Equipage : 265675 hommes, 57276 artilleurs, 400000 droides
>>>>>>> refs/remotes/origin/master
```

La version de Luke avec ce qu'il avait récupéré précédemment et ce qu'il a fait

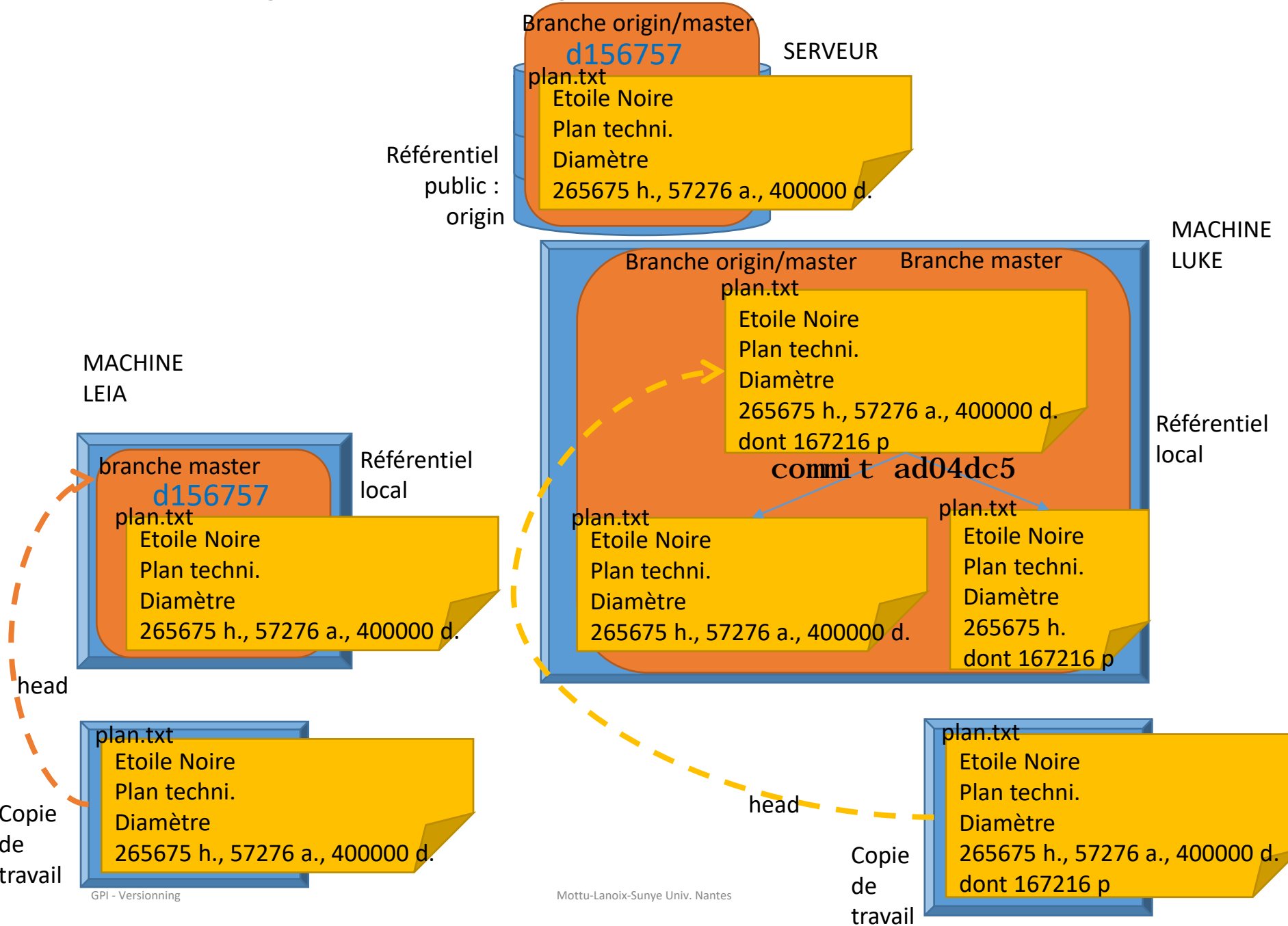
La version de Leia avec ce qu'elle a poussé depuis que Luke avait récupéré

Luke choisit à la main ce qu'il garde
(toutes les nouveautés par exemple ici)

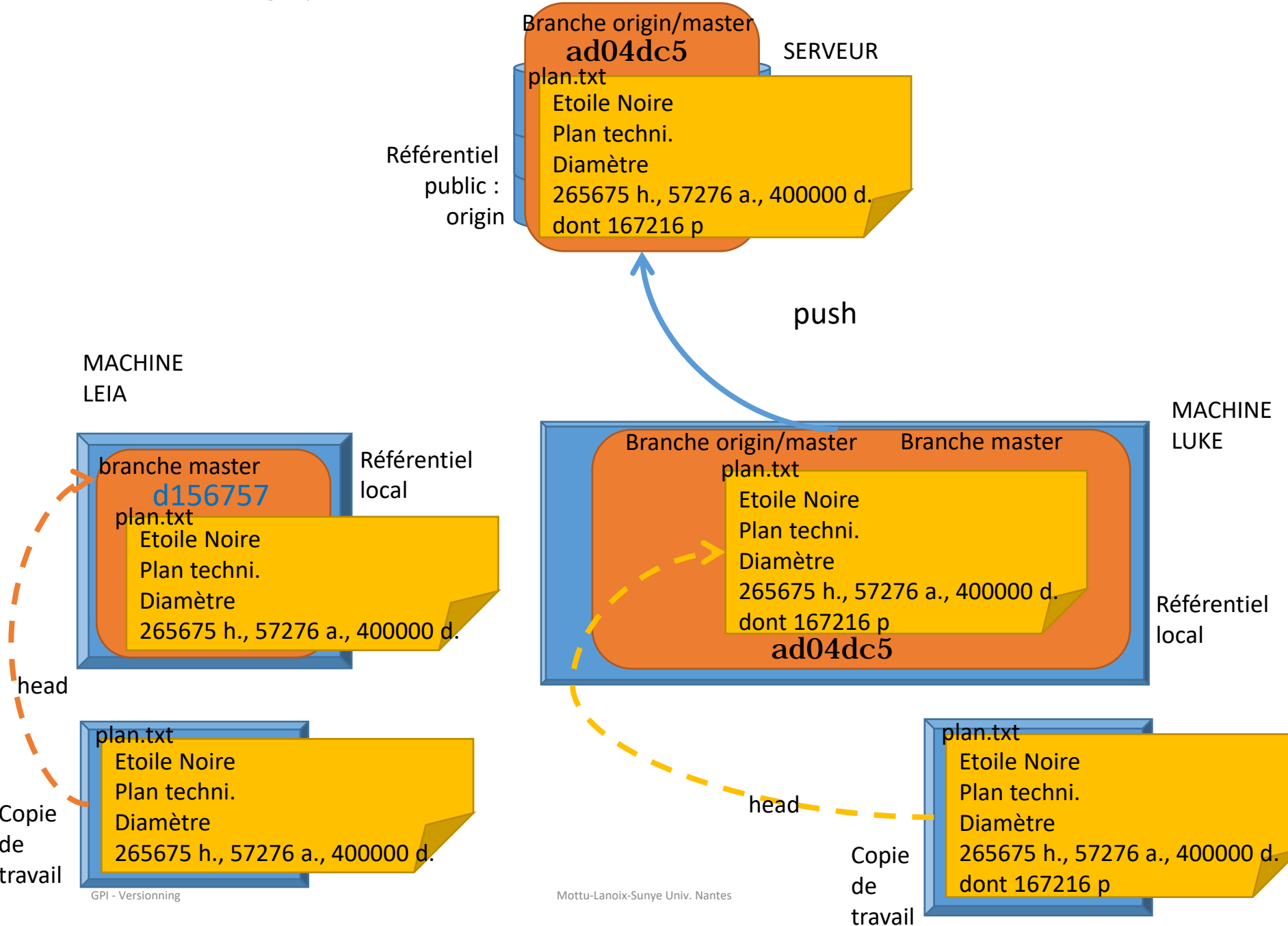
```
Etoile Noire
Plan technique
Diamètre : 120 kilomètres
Equipage : 265675 hommes, 57276 artilleurs, 400000 droides
dont 167216 pilotes
```

C'est son nouveau fichier de travail, résultat d'un « Merge-CONFLICT »

47. Luke@macosx : git commit -a -m "merge conflict"

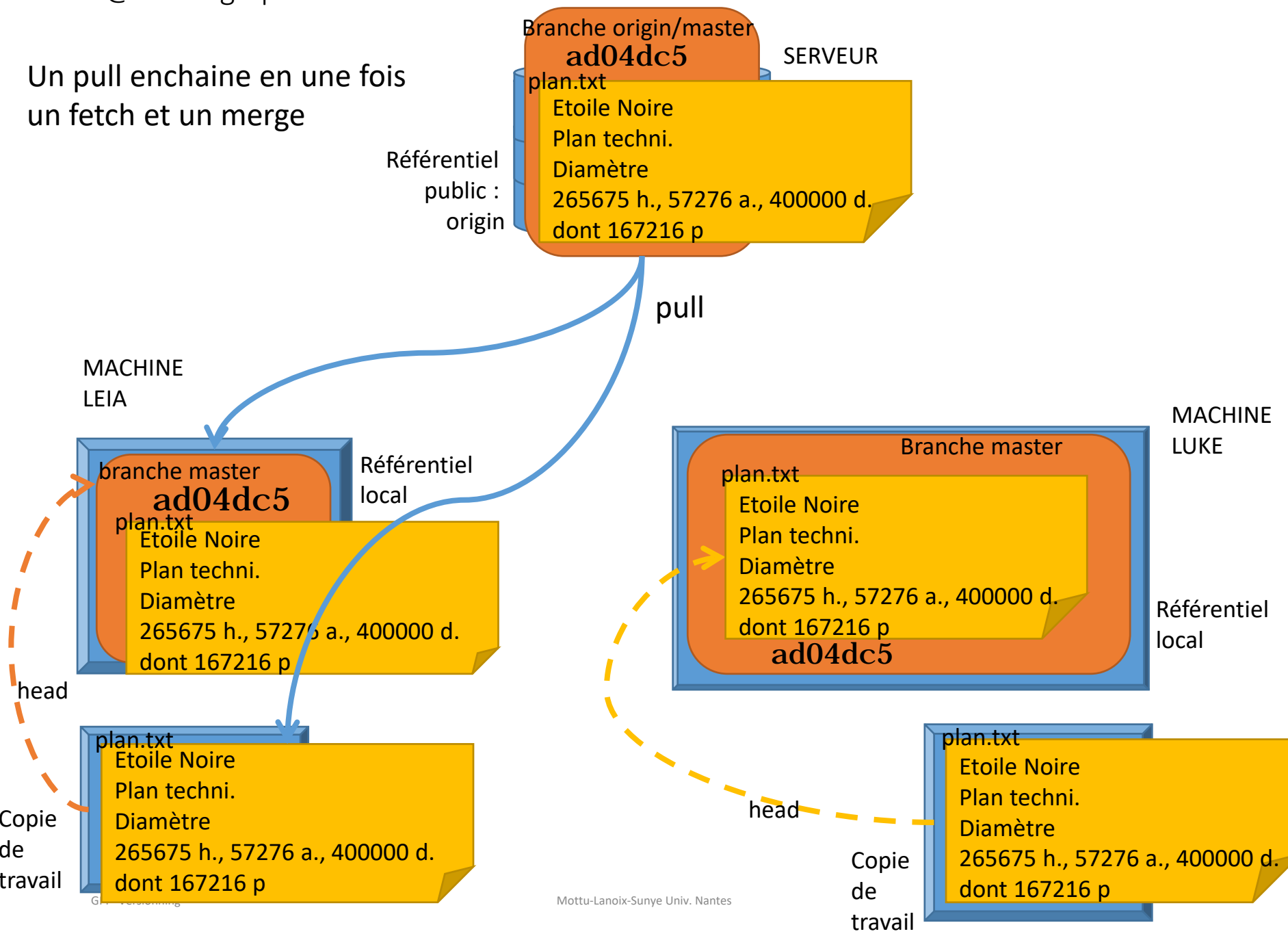


48. Luke@macosx : git push



49. Leia@linux : git pull

Un pull enchaîne en une fois
un fetch et un merge



A vous de jouer (exercice 2, 3, 4 du TD7)