Plan de la présentation

- **La gestion de version, pourquoi : introduction**

But, avantages/inconv de l'utilisation d'un SGV (Système de Gestion de Version)

- **➔ La gestion de version, comment : utilisation classique**

Cycle habituel : récupération d'un projet dans un working dir (WD), modifs, update (pour rester synchro), commit (seulement si tests ok), lien entre WD et repository

- **Les systèmes centralisés et décentralisés**
  - Comparatif général
  - GIT
- **==== PAUSE CAFE vers 15h30 =====**

- **Les clients**
  - Tortoise
  - Eclipse
  - CLI ou le mode console
- **Marquage (tags), Retour en arrière, Branching/Merging: comparatif CVS/SVN**
- **Installation/configuration d'un serveur**
- **Conclusion vers 17h**

Gestion de versions

Comment ça marche ?

Etienne PALLIER
e-mail: etienne.pallier@cesr.fr

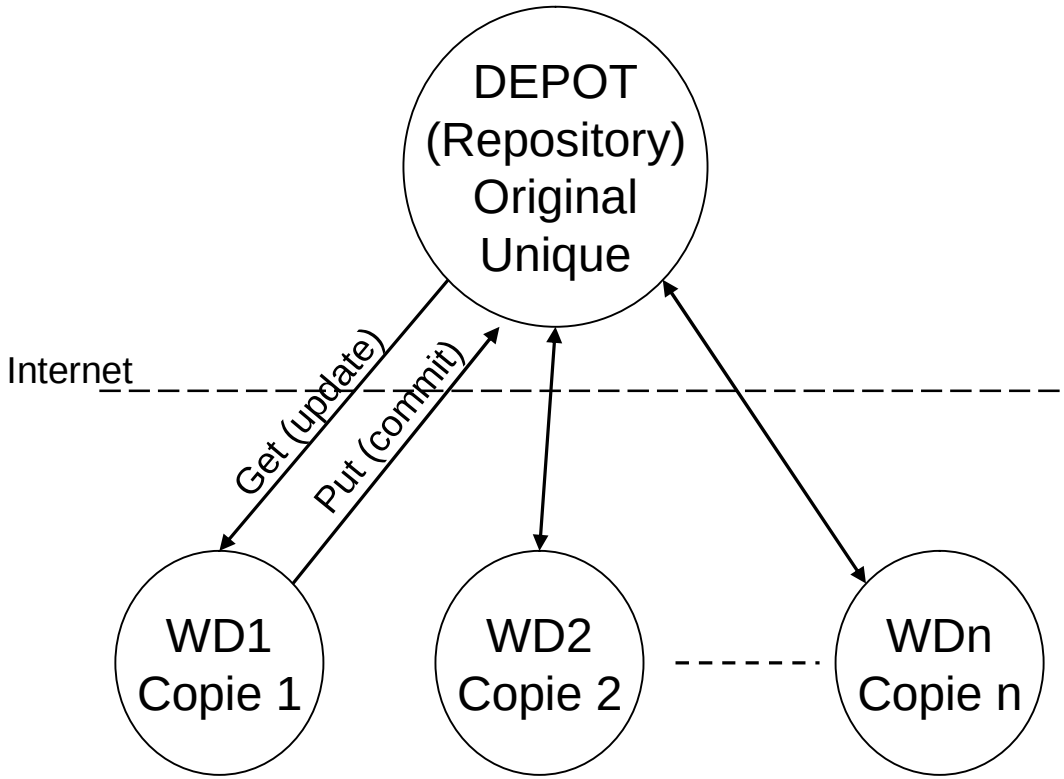
PLAN

- I - Architecture générale
- II - Processus classique

I – Architecture générale

2 gros avantages :

- Travail à plusieurs sur même projet en même temps
- Conservation de l'historique de l'évolution du projet



Les WD (Working Dir) sont totalement déconnectés du dépôt.

Ils peuvent se trouver n'importe où :

- sur le même serveur que le dépôt

- sur un pc de la même société

- sur un pc distant n'importe où dans le monde (OS quelconque)

On peut travailler sur son WD sans se connecter au dépôt pendant plusieurs semaines…

On ne se connecte au dépôt QUE dans 2 cas :

- quand on veut mettre à jour sa copie à partir de l'original (update)

- quand on veut mettre à jour l'original à partir de sa copie (commit)

Le dépôt est PASSIF. Ce sont les WD qui se connectent au dépôt.

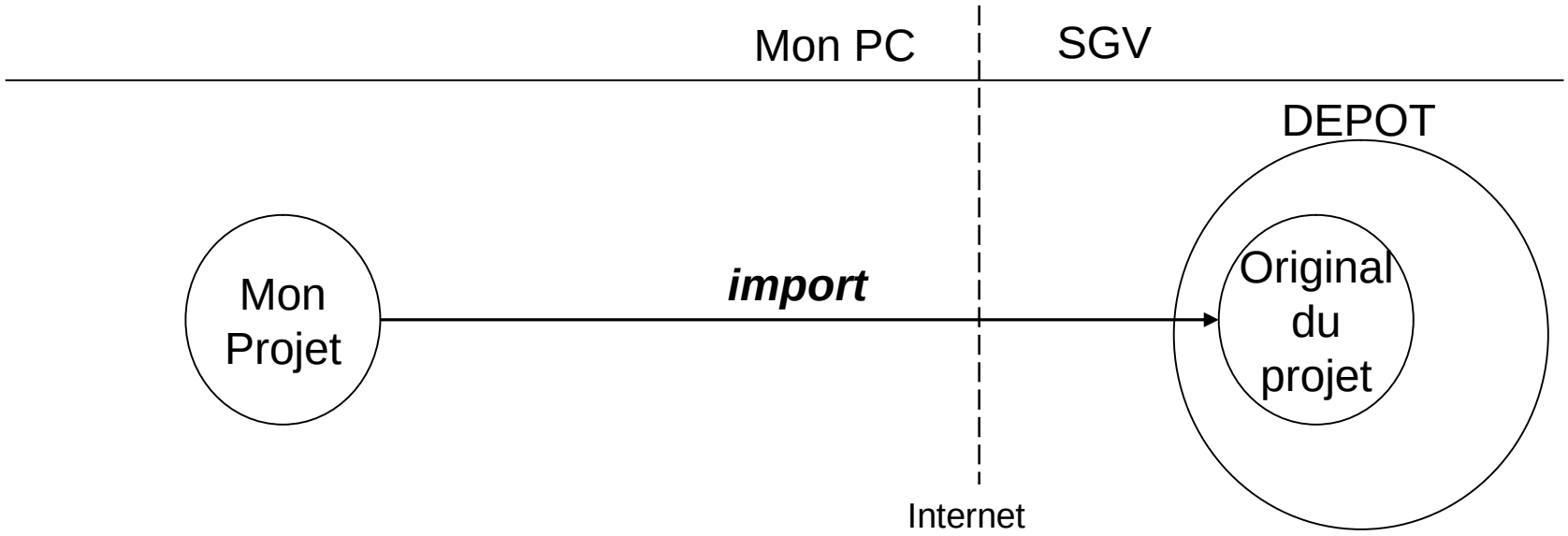
II – Processus classique

(Hypothèse : dépôt déjà existant)

- 1) Initialisation du projet
- 2) Récupération d'une copie du projet
- 3) Cycle de travail habituel
- 4) Livraison d'une version du projet

1) Initialisation du projet (**import**ation dans dépôt)

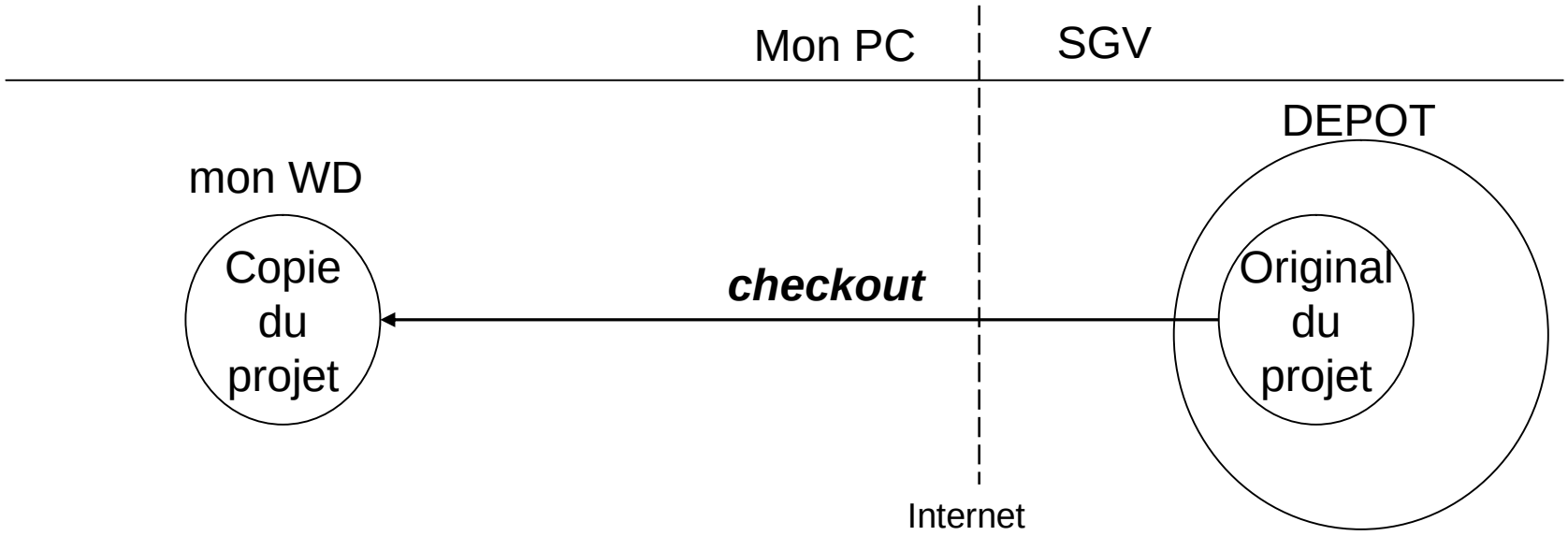
Cette étape est réalisée <u>une fois pour toutes</u> par l'initiateur du projet



Le projet est maintenant accessible au monde entier (gestion droits d'accès)

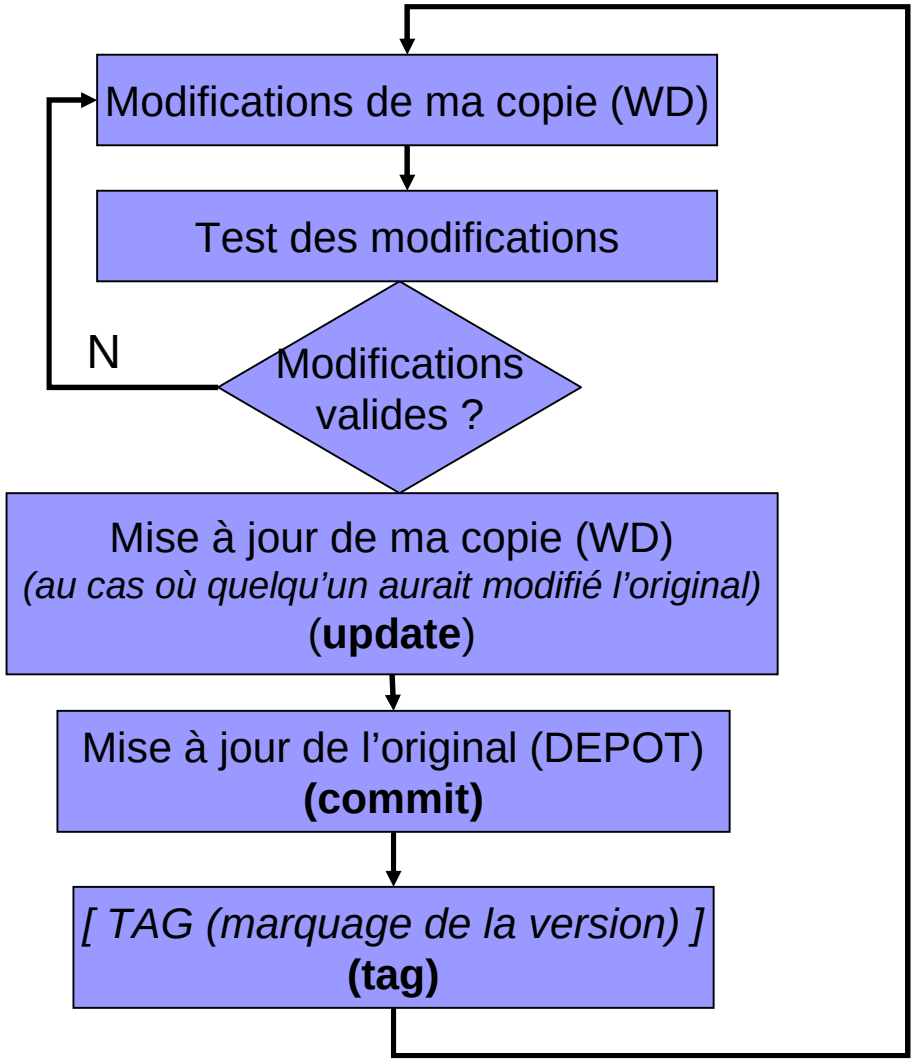
2) Récupération d'une copie du projet (**checkout**)

Cette étape est réalisée <u>une fois pour toutes</u> par tout développeur participant au projet

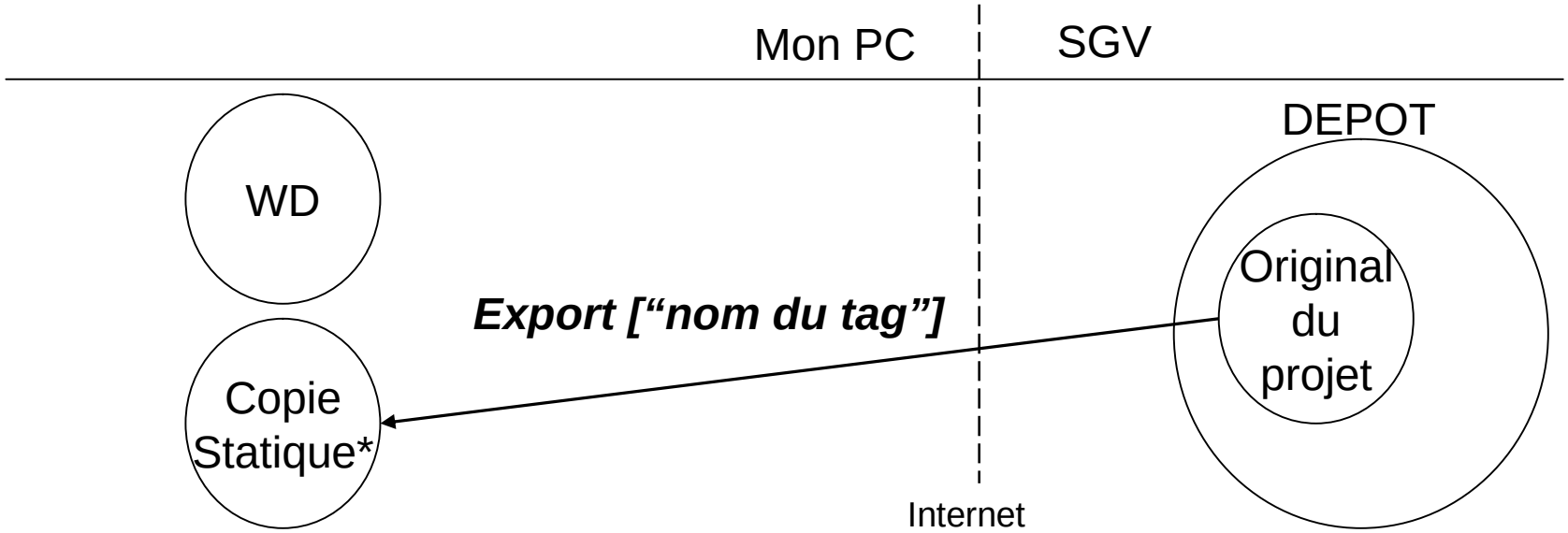


Je peux maintenant travailler sur ma copie du projet

3) Cycle de travail habituel sur le WD (la copie)



4) Livraison d'une version du projet (**export**)



** Non synchronisable avec le dépôt*

Ensuite, on envoie la copie statique sur le serveur de production (via ftp)

Plan de la présentation

- **La gestion de version, pourquoi : introduction**

But, avantages/inconv de l'utilisation d'un SGV (Système de Gestion de Version)

- **La gestion de version, comment : utilisation classique**

Cycle habituel : récupération d'un projet dans un working dir (WD), modifs, update (pour rester synchro), commit (seulement si tests ok), lien entre WD et repository

- **➔ Les systèmes centralisés et décentralisés**
  - Comparatif général
  - GIT
- **==== PAUSE CAFE vers 15h30 =====**

- **Les clients**
  - Tortoise
  - Eclipse
  - CLI ou le mode console
- **Marquage (tags), Retour en arrière, Branching/Merging: comparatif CVS/SVN**
- **Installation/configuration d'un serveur**
- **Conclusion vers 17h**