

Mercurial

pour le programmeur de garage

- Hg – *the chemical symbol for the element mercury*
- Mercurial (adj) relating to or having characteristics attributed to the god Mercury

Qui suis-je? Qui êtes-vous?

- Étudiant à l'UdeS, 2e année
- Mon passé avec PHP Québec
- Qui a déjà utilisé un SCV?
 - SCVD?

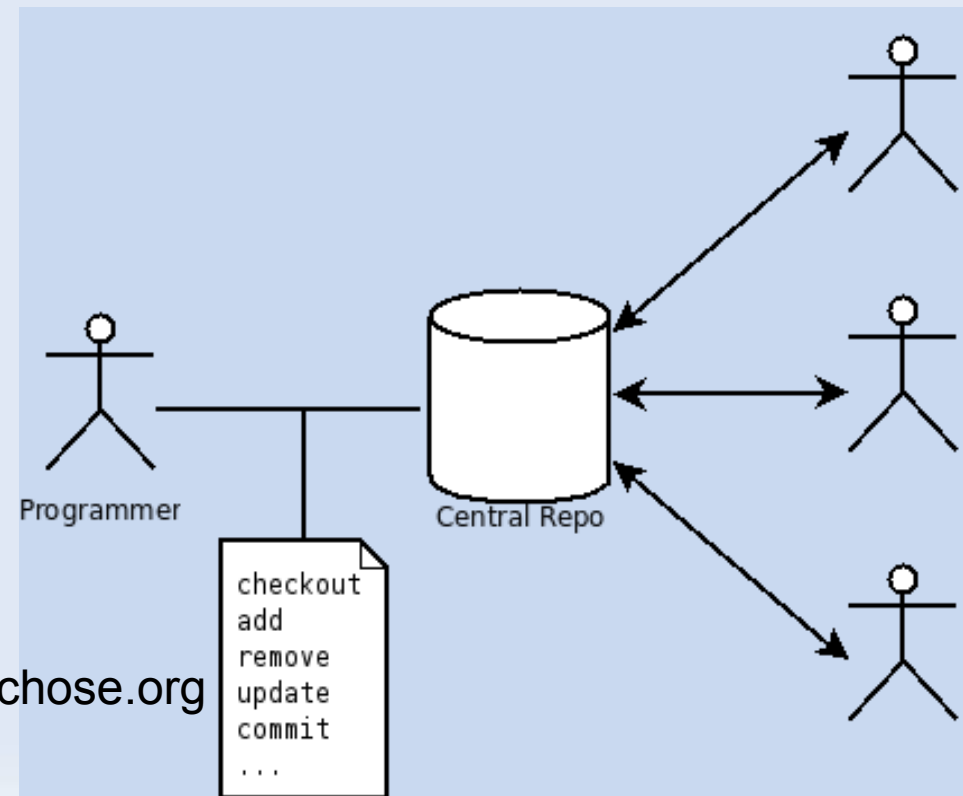
Plan de présentation

- SCV Centralisé
- SCV Distribué
 - Avantages et Inconvénients
- Différents SCVD
- Certains concepts de base
- Les commandes (avec exemples!)
- Outils (avec exemples?)
- Extensions (avec exemples?)

SCV Centralisé

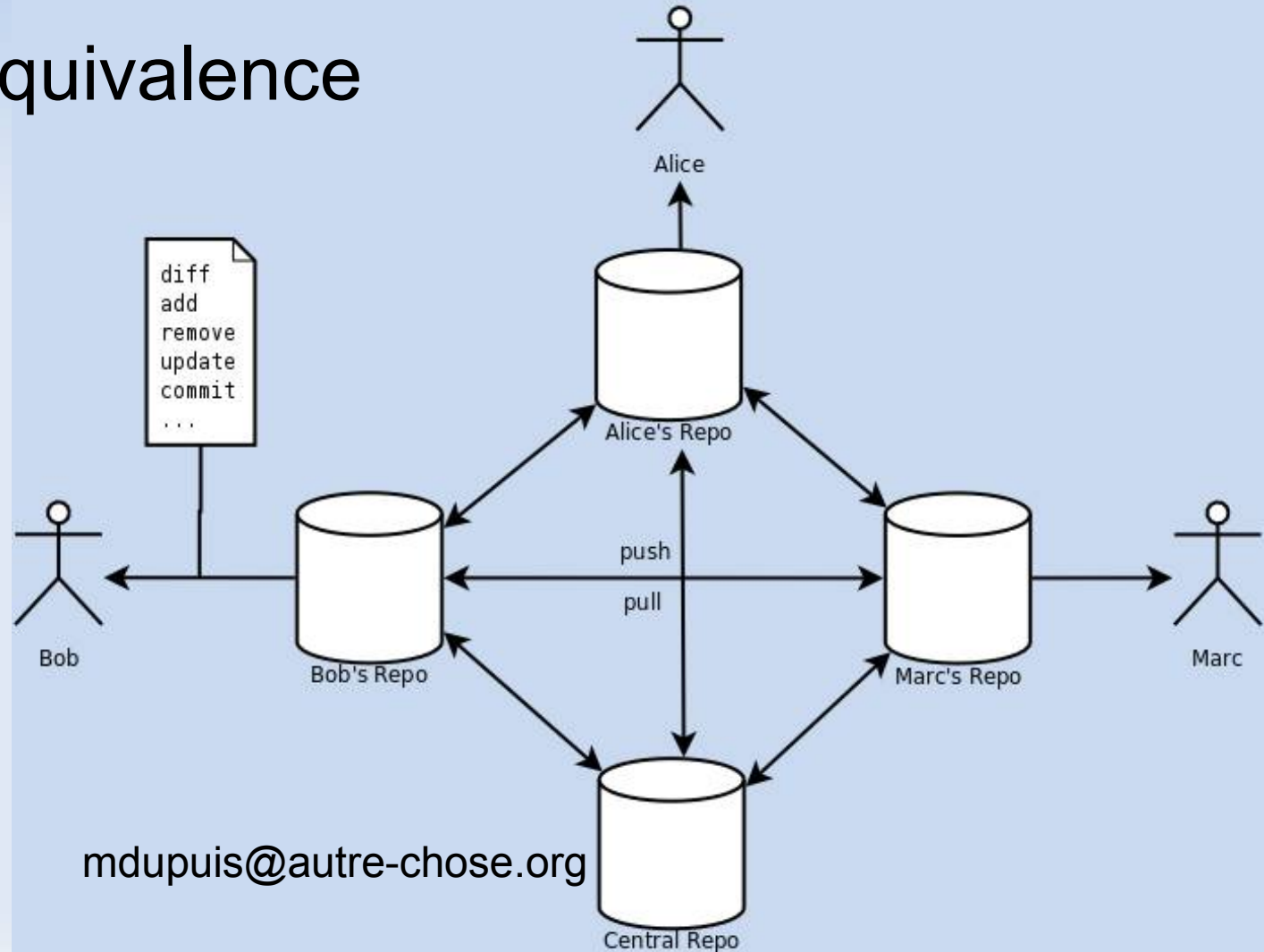
- Serveur central
- Sauvegarde = Check In (commit)
- Désavantages (branches/merges)

Exemples de SCV :
SVN, CVS



SCV Distribué

- Chaque repo connaît l'historique
- On parle d'équivalence



Avantages

- Facile de gérer les branches et les merges
- Repo local, toujours disponible
- Rapide et peu coûteux en ressources
- Accès à l'historique (ex.: renommer un fichier)
 - Dossier .hg à la racine

Inconvénients

- Façon différente de travailler
- Besoin de convention (qui est le maître?)
- Pas aussi développé (SCV, outils, etc.)
- Lequel choisir?

Différents SCVD

- Git -- Linus Torvalds
- Bazaar -- Canonical
- Mercurial -- Matt Mackall

Git

- Écrit en C
- le plus rapide
- Projets : Kernel linux, OLPC, RoR, projets Ruby
- Hébergement : [GitHub.com](https://github.com)
- Comple(t|xe)

Bazaar

- Écrit en Python
- le moins rapide
- Projets : Ubuntu, Drupal, Drizzle
- Hébergement : LaunchPad.net
- Similaire à SVN
- Beaucoup de révisions, change souvent

Mercurial

- Écrit en Python
- Relativement rapide, léger
- Projets : Mozilla, Netbeans, projets Python
- Hébergement : FreeHG.org, Intuxication (fr)
- Similaire à SVN

Concepts

- Révision : changements groupés
 - Consiste en un numéro et un ID unique
 - ex.: changeset: 0:8fc598890cdd4
- Branche : 2 têtes pouvant être combinées
 - Même révision parente à la base
 - Pas nécessairement 2 lignes de développement (2 repos)
- Un fichier de configuration par repo, un pour l'utilisateur, et un global à mercurial

Commandes

- clone : permet de cloner un repo
- add : ajoute un fichier au repo
- remove : supprime un fichier du repo
- log : affiche l'historique du repo
- status: visualisation globale des changements
- commit : sauvegarde les changements, crée une révision
- push/pull : échange des révisions avec un autre repo
- update/merge : mets à jour les fichiers locaux/combine 2 branches
- fetch : même chose que pull → update → merge

Commandes

Exemple!

Commandes supplémentaires

- diff: affiche les changements (format patch)
- revert : efface les changements effectués à un fichier
- rollback : même chose que `revert` *
- incoming/outgoing : différence entre 2 repos
- serve : dessert une interface web
- tag : attache un nom (ex.: "v1.0") à une révision
- tags : liste les "tags"

Commandes supplémentaires (2)

- `bundle` : exporte un groupe de révisions format binaire
- `unbundle` : importe un fichier de révisions binaire
- `export` : exporter **une** révision format *patch*
- `strip` : retire les révisions du repo
- `import` : importe un *patch* en tant que révision
- `annotate` : montre les lignes d'un fichier avec no de révision
 - Utile si combiné à la commande `log`

Commandes supplémentaires

Exemples?

Outils

- Plugiciel pour Eclipse
- TortoiseHG (Windows seulement)
- Conversion
 - Hgsvn : migrer ou gérer un dépôt SVN avec mercurial
 - Autres : GitConversion, extension Convert (Darcs, CVS), Tailor (bzd)
- hgk/hgview : GUI pour la commande log
- Meld : GUI pour faciliter la combinaison de 2 révisions lors de conflits

Hgsvn

- <http://pypi.python.org/pypi/hgsvn>
- hgimportsvn : importe un repo SVN depuis la révision 0 (loooooong...!)
 - ex.: hgimportsvn <http://domain.tld/repo> target-folder
 - Si migration : `find . -name '.svn' | xargs rm -r` dans "target-folder"
- hgpullsvn : importe les nouvelles révisions dans mercurial

Extensions

- glog, hgk/hgview : pour visualiser l'arbre de révisions
- mq : patch queue (Git) – gestionnaire de *patches*
- Forest : gérer un arborescence de repos
 - Utile pour gérer un repo comprenant mq
 - Maintenu par Simon Law

mq

- `qinit` : initialise la queue
- `qclone` : clone un repo mq
- `qnew` : crée un nouveau *patch*
- `qdiff` : même chose que `diff`
- `qrefresh` : met à jour le contenu de la *patch* courrante
- `qcommit` : soumet le *patch*
- `qtop`, `qseries` : affiche le *patch* courant, les *patches*
- `qpop`, `qpush` : applique ou dé-applique une *patch*
- `qgoto` : se déplacer dans la queue de *patches*

mq

Exemples?

Maxime Dupuis

mdupuis@autre-chose.org

Forest

- Configuration
 - Installation
 - paramétrisation

- Commandes
 - f* (où * est le nom d'une commande hg)

Forest

Exemple!

Questions? Commentaires?

- RTFM : <http://selenic.com/mercurial/wiki>
- Pour la présentation : Autre-Chose.org
- Me rejoindre : mdupuis@autre-chose.org

Merci!