



sshGate

Patrick Guiran
Chef de projet support
pguiran@linagora.com

Sommaire

1. Problématique des accès aux serveurs
 - Besoins Linagora
 - Recherche d'une solution
2. Présentation de sshGate
 - Bugs connus & Fonctionnalités demandées
 - Roadmap
3. Internals
 - Fonctionnements
 - Architecture logicielle
 - Industrialisation
4. Réutilisation de sshGate
5. Démo

Problématiques des accès aux serveurs

- Accès par mot de passe
 - Effet « post-it »
 - Partagé entre administrateurs
 - ... ou possédé par un seul administrateur (spof)
- Accès par clé ssh
 - A qui est la clé ?
 - Ajout des clés du « copain » -> solvable
- Accès à l'ensemble des serveurs
 - Même les plus critiques (mail, base de données)
 - ... Souvent de manière inconditionnelle

Gestion des accès et sécurité

- Gestion du départ d'un administrateur ?
- Qui a accès au serveur ? (simple)
- A quel serveur a accès cet administrateurs ? (compliqué)
 - « Simple » lorsque l'administrateur a accès à tous les serveurs :-)
 - Bon administrateur : « Mais si c'est simple ! »

```
for serveur in $( cat list-server.txt ) ; do
    ssh $serveur 'cat ~/.ssh/authorized_keys2?' \
        | grep user-sshkey >/dev/null
    [ $? -eq 0 ] && echo '${serveur}'
done
```

- Qui gère les accès ?

Besoin de Linagora

- Must have
 - Utilisation de ssh
 - Authentification par clés ssh
 - Que les clés des utilisateurs ne soient pas sur les serveurs
 - Gestion des accès centralisée (ACL)
- Nice to have
 - Enregistrement de l'activité des utilisateurs
 - Enregistrement des sessions des utilisateurs
 - Notification des actions d'administration

Contraintes et défis

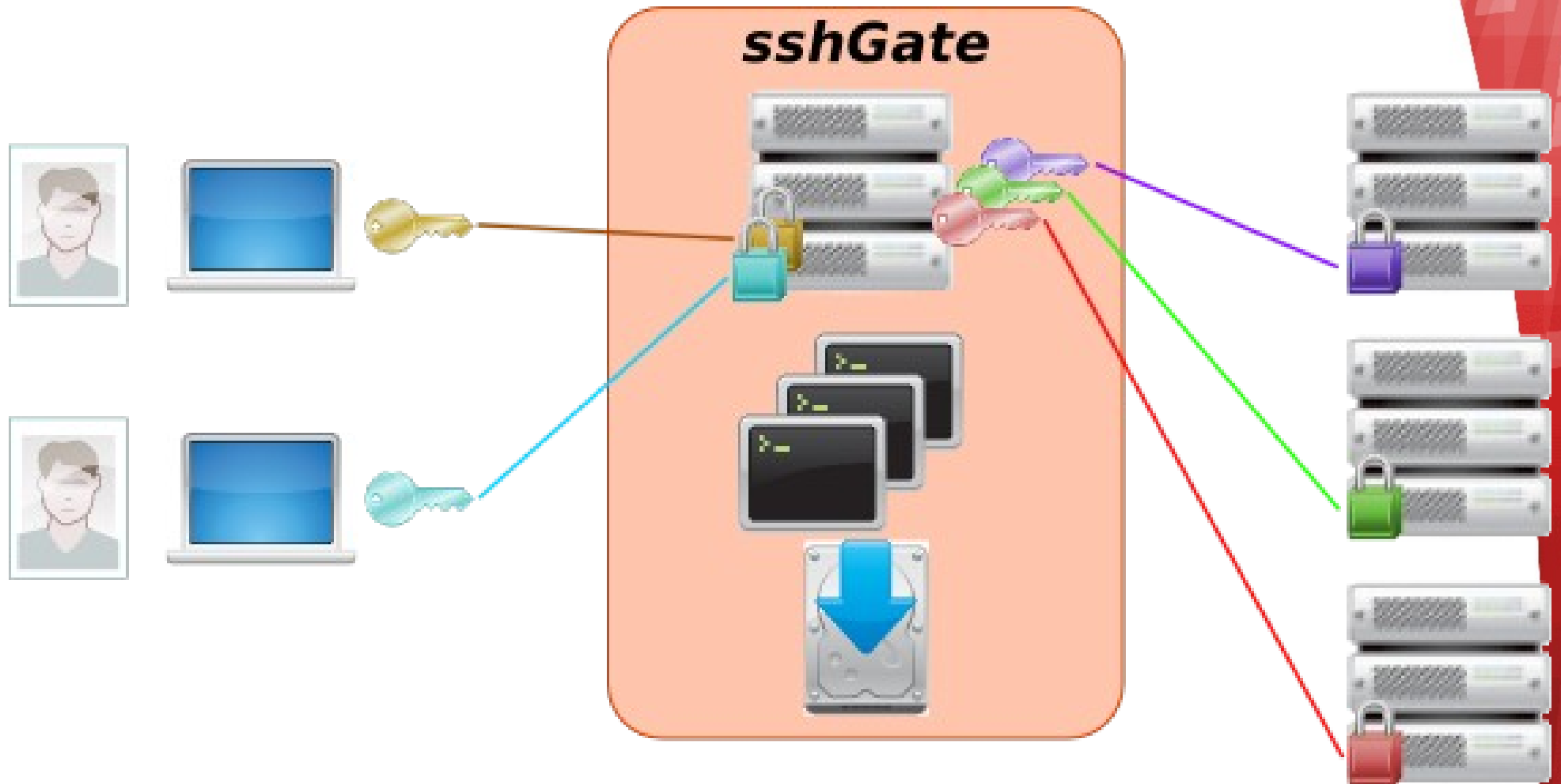
- Se reposer sur les outils existants : OpenSSH, putty
 - Aucune installation sur les serveurs administrés
 - Aucune installation sur les postes clients
- Multiplate-forme
 - Le serveur sshGate
 - Les serveurs administrés
 - Les postes clients
- Pas (ou peu) de patchs sur le serveur sshGate (patchs sshd)
- Léger, peu de dépendances logicielles (pas de BDD)

Recherche d'une solution

- Wallix AdminBastion
 - Solution française, propriétaire, ssh/telnet/rdp
- Observe-it
 - Solution américaine, propriétaire, ssh/telnet/rdp
- sshProxy
 - Projet Open-Source (GPLv2), python, client lourd
 - inactif depuis 2008(?), impossible à télécharger sur le site officiel
- AdminProxy
 - Projet Open-Source, commandé par l'État Français
 - Projet sur 2 ans. Fin prévu en 09/2010
 - Où sont les sources ? :-)



Présentation de sshGate



Fonctionnalités

- Sessions SSH & Transfert de fichier via SCP
- Gestion centralisée des accès (par utilisateur et groupe)
- Gestion des alias pour les serveurs administrés
- Support multi-login par serveur administré
- Configuration SSH globale, et spécifique par serveurs et login
- Enregistrement de l'activité des utilisateurs
 - Événement de connexion, de transfert de fichiers etc...
- Enregistrement des sessions SSH
 - Historique des commandes saisies, par utilisateur et serveur administré
- Interface d'administration en mode « CLI »

Caractéristiques

- Licence GPLv2+
- Développé en Shell Script (sh, dash, bash, zsh)
- multiplate-forme : Linux, Solaris, *BSD, MacOS (+ Windows/Putty)

- Naissance du projet : Août 2010
- Première mise en production : Septembre 2010
- Version actuelle (interne) :
 - En production : 0.1.192
 - Trunk : 0.2.38 (passage en production imminente)
 - Release 1.0 prévu pour juillet 2011

- Aujourd'hui : 55 utilisateurs, 161 systèmes administrés

Utilisation de sshGate par Linagora

- Quelques indicateurs
 - 57 utilisateurs
 - 162 machines administrées
 - 214 nom d'alias de machine administrées
- Historique du nombre de sessions gérées

```
gate.par.lng:/var/log/sshgate# find . -name "201103*" | wc -l  
2263  
  
gate.par.lng:/var/log/sshgate# find . -name "201104*" | wc -l  
1718  
  
gate.par.lng:/var/log/sshgate# find . -name "201105*" | wc -l  
758
```

Bugs connus & fonctionnalités demandées

- DDOS : remplissage des logs
 - Surveillance de l'évolution de la taille des logs
 - Et déconnexion des sessions abusives
- Saturation du nombre de fd ouvert
 - Limiter le nombre de connexion par utilisateur
 - Killer les sessions « idle » trop longue
- Possibilités de « cacher » certaines commandes

```
user@host $ read -s var      # rm -rf *  
user@host $ eval "${var}"   # AIE
```

Roadmap

- En cours
 - Packaging debian (début juin 2011)
 - Exposition de certaines commandes CLI (mai 2011)
 - Support du telnet (fin mai 2011)
 - Correction du DDOS + limitation des sessions (juillet 2011)
- A venir
 - Packaging fedora, arch, Solaris.
 - Authentification des utilisateurs par LDAP
 - Support des certificats
 - Interface web d'administration

Sommaire

1. Problématique des accès aux serveurs

- Besoins Linagora
- Recherche d'une solution

2. Présentation de sshGate

- Bugs connus & Fonctionnalités demandées
- Roadmap

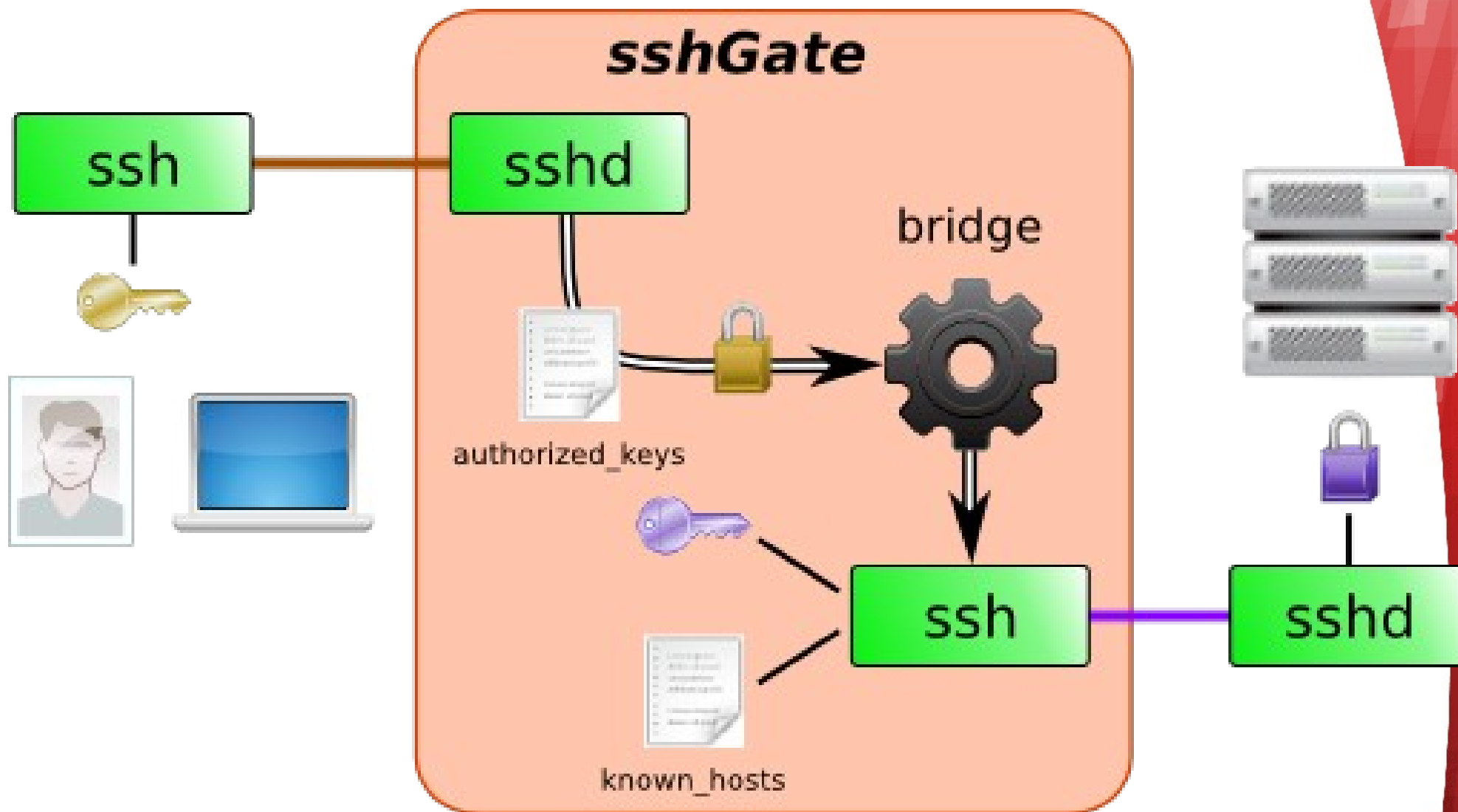
3. Internals

- Fonctionnements
- Architecture logicielle
- Industrialisation

4. Réutilisation de sshGate

5. Démo

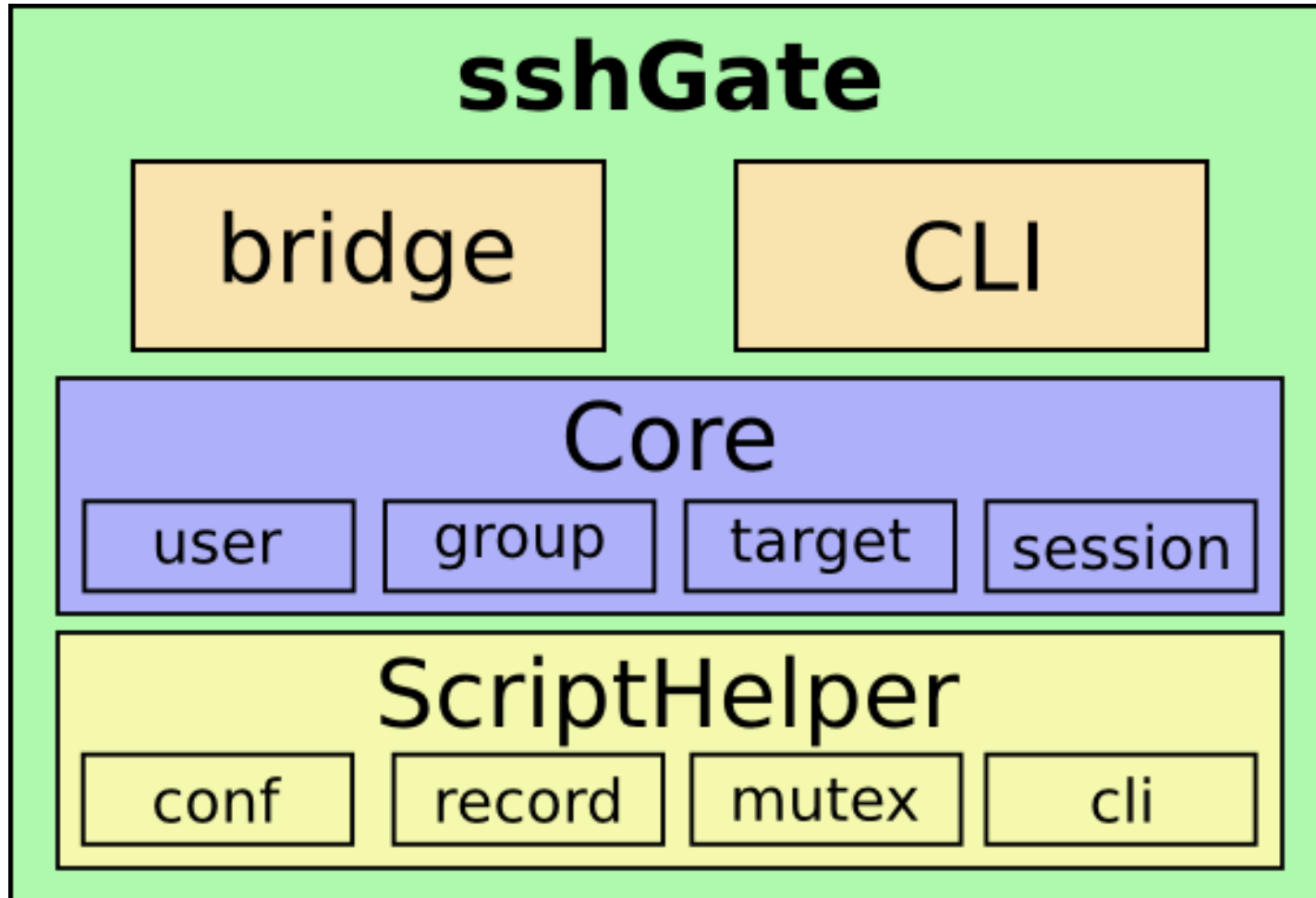
Fonctionnement d'une session



Déroulement d'une ouverture de session

- Connexion de l'utilisateur
 - Vérification de la présence de sa clé
 - Exécution du Bridge et passage du login + SSH_ORIGINAL_COMMAND
- Vérification de l'existant de l'utilisateur dans sshGate
- Parsing du SSH_ORIGINAL_COMMAND
 - Extraction du nom de la machine administrée cible
 - Extraction de la commande à effectuer (si présence)
 - Si aucun nom de machine cible n'est fournis, passage en mode interactif
- Si machine administrée cible existe
 - Lancement de l'action demandé (ssh ou scp)
 - Avec vérification de l'authenticité de la machine cible (known_hosts)

Architecture logicielle



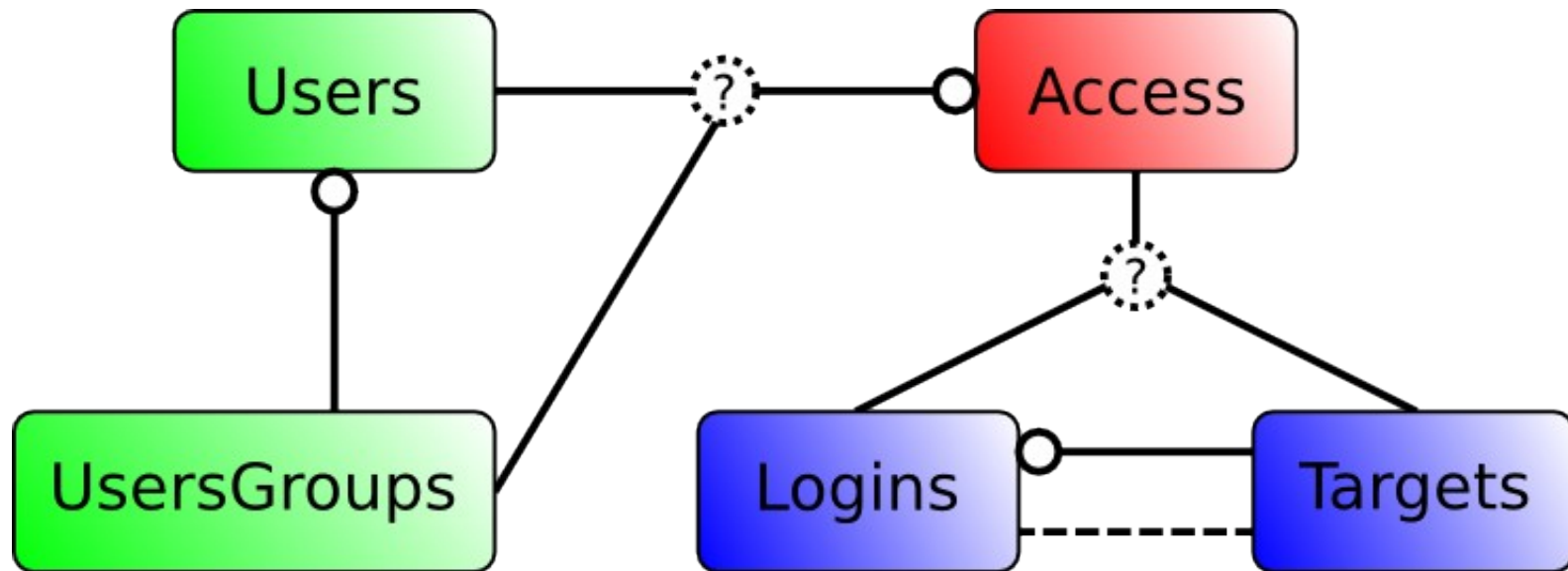
CLI d'administration

```
tauop : bash
Fichier  Édition  Affichage  Historique  Signets  Configuration  Aide
tauop@Tauopbox:~$ sshg cli
Enter passphrase for key '/home/tauop/.ssh/id_rsa':
sshGate administration Interface
By Patrick Guiran <pguiran@linagora.com>

Use 'help' command to list all avariable commands

sshGate > help
target access          -- Targets access related commands
target ssh             -- Targets ssh related commands
target                 -- Target related commands
usergroup              -- Usergroup related commands
user                   -- User related commands
sshGate > user list *sco*
scolson
scourtois
sshGate > user scolson access info
scolson --- usergroup(CORE_TEAM_OBM) ---> root@arathi.linagora.com
scolson --- usergroup(CORE_TEAM_OBM) ---> root@atlassian-obm.linagora.com
scolson --- usergroup(CORE_TEAM_OBM) ---> root@coruscant.par.lng
scolson --- usergroup(CORE_TEAM_OBM) ---> root@hudson-obm.linagora.com
scolson --- usergroup(CORE_TEAM_OBM) ---> root@obm-lng.virt.par.lng
scolson --- usergroup(CORE_TEAM_OBM) ---> root@obmonline.netaktiv.com
scolson --- usergroup(CORE_TEAM_OBM) ---> root@obm.org
sshGate > target list *arathi*
arathi.linagora.com
linagora-arathi.netaktiv.com
sshGate > exit
NOTICE: No modification noticed. Prepared e-mail will not be sent.
NOTICE: No modification noticed. Repport e-mail will not be sent
Connection to gate.par.lng closed.
tauop@Tauopbox:~$
```

Administration - modèle de données



Bibliothèque ScriptHelper

- Ensemble de bibliothèques
 - Permettant d'écrire plus rapidement les Scripts Shell
 - Se voulant le plus POSIX possible
- Quelques bibliothèques
 - `exec.lib.sh` : exécution avec log et vérification
 - `ask.lib.sh` : poser des questions
 - `cli.lib.sh` : permet de construire une CLI rapidement
 - `mutex.lib.sh` / `lock.lib.sh` : gestion des verrous
 - `record.lib.sh` : enregistrement de terminal
 - ...

Utilisation de ask.lib.sh

```
ASK SSHGATE_TARGETS_DEFAULT_SSH_LOGIN \  
  "What the default user account to use when connecting to target host ?" \  
  "${SSHGATE_TARGETS_DEFAULT_SSH_LOGIN}"  
  
CONF_SAVE SSHGATE_TARGETS_DEFAULT_SSH_LOGIN  
  
ASK --yesno SSHGATE_MAIL_SEND \  
  "Activate mail notification system [Yes] ?" \  
  "Y"  
[ "${SSHGATE_MAIL_SEND}" = 'N' ] && SSHGATE_MAIL_SEND='false'  
if [ "${SSHGATE_MAIL_SEND}" = 'Y' ]; then  
  SSHGATE_MAIL_SEND='true'  
  ASK SSHGATE_MAIL_TO \  
    "Who will receive mail notification (comma separated mails) ?" \  
    "${SSHGATE_MAIL_TO}"  
  [ -z "${SSHGATE_MAIL_TO}" ] && SSHGATE_MAIL_SEND='false'  
else  
  SSHGATE_MAIL_SEND='false'  
fi  
CONF_SAVE SSHGATE_MAIL_SEND  
CONF_SAVE SSHGATE_MAIL_TO
```

Utilisation de cli.lib.sh

```
# chargement de ScriptHelper
. ./lib/cli.lib.sh

# génération de l'aide et de son menu
CLI_REGISTER_HELP '/tmp/sshgate-cli-help.txt' \
    SSHGATE_GET_HELP \
    SSHGATE_DISPLAY_HELP \
    SSHGATE_DISPLAY_HELP_FOR

CLI_REGISTER_MENU 'user' 'User related commands'
CLI_REGISTER_COMMAND 'user list' 'USERS_LIST'
CLI_REGISTER_COMMAND 'user list <pattern>' 'USERS_LIST \1'
CLI_REGISTER_COMMAND 'user add <user> mail <email>' 'USER_ADD \1 \2'
CLI_REGISTER_COMMAND 'user del <user>' 'USER_DEL \1'
CLI_REGISTER_COMMAND 'user build auth_keys' 'USERS_AUTH_KEYS_BUILD'

# ....

# lancement ce la CLI
CLI_RUN
```

Industrialisation (1/2)

- SshGate et ScriptHelper possèdent
 - Construction d'une tarball de déploiement
 - Script de déploiement / installation
 - Des jeux de tests

```
tauop@Tauopbox:~/taff/.../sshGate$ ./build.sh server
sshgate version ? 0.2
sshGate build number ? 014
Include ScriptHelper in package ? y
- Build sshgate-server package ... OK
tauop@Tauopbox:~/taff/.../sshGate$
```

(oui c'est aussi possible avec le Shell Scripting :-)

Industrialisation (2/2)

```
root@tauop-netbook:/opt/sshgate/bin/tests# ./test.sh all
```

```
- Loading sshGate core ... OK
- Setup sshGate data directory ... OK
- Generate temporary test file ... OK
- Generate temporary sshkey test file ... OK
- Create and setup temporary Unix account ... OK
- Reset temporary test file ... OK
- Reset sshGate data directories ... OK
- Generate user tests ... OK
- Launch user tests ... OK
- Reset temporary test file ... OK
- Reset sshGate data directories ... OK
- Generate target tests ... OK
- Launch target tests ... OK
- Reset temporary test file ... OK
- Reset sshGate data directories ... OK
- Generate usergroup tests ... OK
- Launch usergroup tests ... OK
- Reset temporary test file ... OK
- Reset sshGate data directories ... OK
- Generate access tests ... OK
- Launch access tests ... OK
- Remove tests data ... OK
root@tauop-netbook:/opt/sshgate/bin/te
```

```
tauop@Tauopbox:/tmp/sshGate-server-0.2-014$ sudo ./install.sh
[sudo] password for tauop:
```

```
--- sshGate server configuration ---
      by Patrick Guiran
```

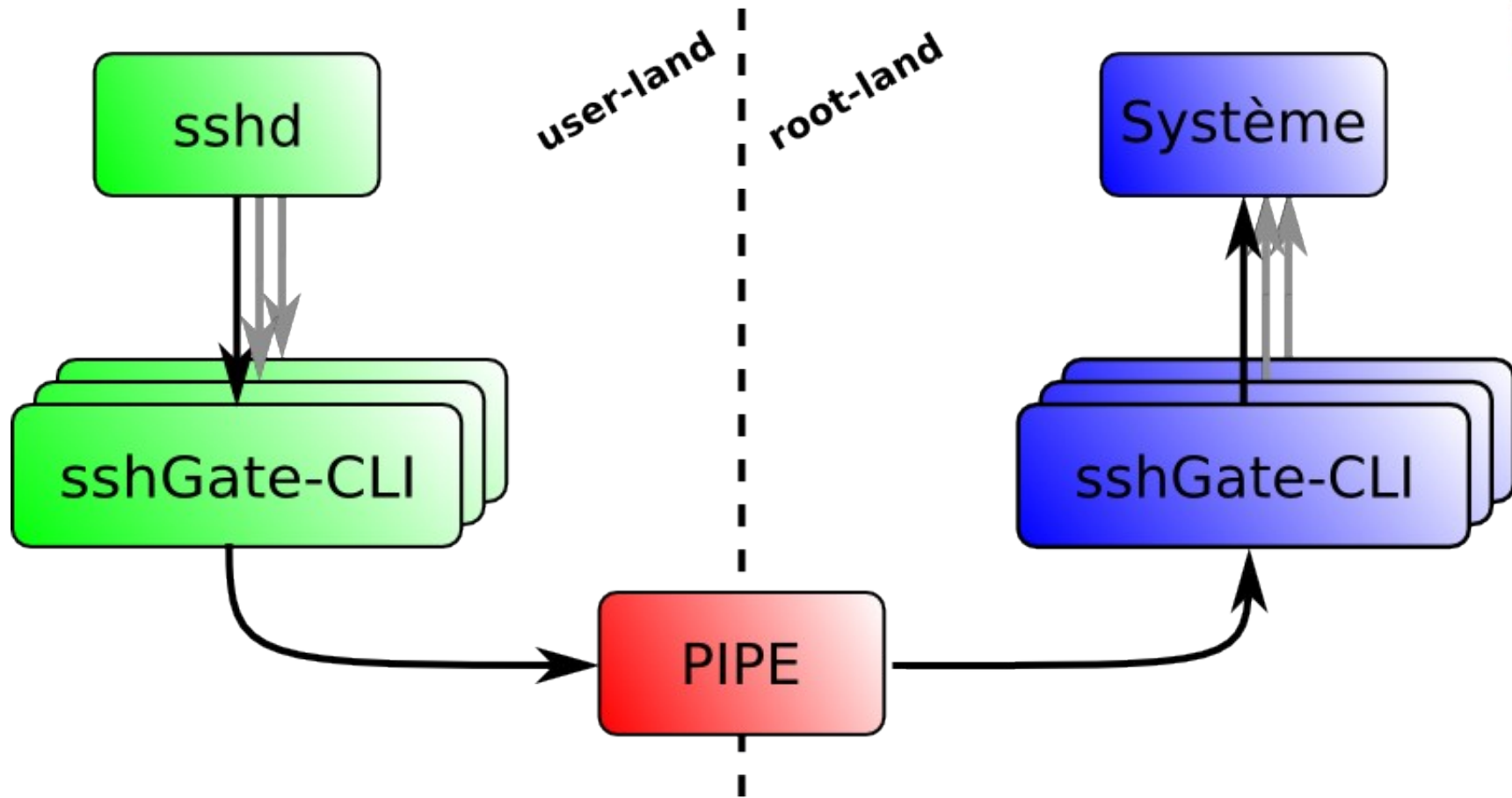
```
It seems that sshGate is already installed on your system.
Do you want to re-use the installed configuration [Y] ?
```

```
- Reload configuration ... OK
- Installing sshGate ... OK
- Generate default sshkey pair ... OK
- Setup files permissions ... OK
- Install archive cron ... OK
```

```
NOTICE: You may add /opt/sshgate/bin in your PATH variable
```

```
tauop@Tauopbox:/tmp/sshGate-server-0.2-014$
```


Recyclage du projet sshGate



Participer :-)

The screenshot shows the GitHub interface for the repository 'sshGate' by user 'Tauop'. At the top left is the GitHub logo. The top right navigation bar includes the user profile 'Tauop', 'Dashboard', 'Inbox 0', 'Account Settings', and 'Log Out'. Below this is a search bar and links for 'Explore GitHub', 'Gist', 'Blog', and 'Help'. The repository header shows 'Tauop / sshGate' with buttons for 'Admin', 'Unwatch', 'Pull Request', and view counts for '3' watches and '3' forks. A navigation bar contains 'Source', 'Commits', 'Network', 'Pull Requests (0)', 'Fork Queue', 'Issues (6)', 'Wiki (7)', and 'Graphs', with 'Branch: master' on the right. Below this are 'Switch Branches (1)', 'Switch Tags (0)', and 'Branch List'. The main content area features a description: 'Tools to configure and use a ssh proxy server — Read more' and a 'click here to add a homepage' link, with a 'Downloads' button. A URL bar shows 'SSH', 'HTTP', 'Git Read-Only', and the URL 'git@github.com:Tauop/sshGate.git', with a note 'This URL has Read+Write access'. A commit entry is shown: 'update README for sshg' by 'Tauop (author)' on 'April 19, 2011'. To the right of the commit are links for 'commit', 'tree', and 'parent' with their respective hashes. At the bottom, the repository name 'sshGate /' is followed by a table header with columns 'name', 'age', 'message', and 'history'.

github
SOCIAL CODING

Tauop | Dashboard | Inbox 0 | Account Settings | Log Out

Explore GitHub | Gist | Blog | Help | Search GitHub...

Tauop / sshGate

Admin | Unwatch | Pull Request | 3 | 3

Source | Commits | Network | Pull Requests (0) | Fork Queue | Issues (6) | Wiki (7) | Graphs | Branch: master

Switch Branches (1) | Switch Tags (0) | Branch List

Tools to configure and use a ssh proxy server — Read more
click here to add a homepage

Downloads

SSH | HTTP | Git Read-Only | git@github.com:Tauop/sshGate.git | This URL has Read+Write access

update README for sshg

Tauop (author)
April 19, 2011

commit 5b074cd4b24b47db005b
tree 2769c7bbd7454b2b9fd6
parent cbe9a615d6728ba3bc0f

sshGate /

name	age	message	history
------	-----	---------	---------

Participer :-)

The screenshot shows the GitHub interface for the repository 'ScriptHelper' by user 'Tauop'. The page includes the GitHub logo, user navigation links (Dashboard, Inbox, Account Settings, Log Out), and a search bar. The repository name 'ScriptHelper' is displayed with a smiley face icon. Action buttons for 'Admin', 'Unwatch', 'Pull Request', and view counts (3 eyes, 2 forks) are visible. A navigation bar shows 'Source' as the active tab, with other options like 'Commits', 'Network', 'Pull Requests (0)', 'Fork Queue', 'Issues (2)', 'Wiki (0)', and 'Graphs'. The current branch is 'master'. Below this, there are links for 'Switch Branches (1)', 'Switch Tags (0)', and 'Branch List'. A section titled 'Shell libraries to help writing shell script' includes a 'Read more' link and a 'click here to add a homepage' link. A 'Downloads' button is present. The 'SSH' tab is selected, showing the URL 'git@github.com:Tauop/ScriptHelper.git' with a note that this URL has 'Read+Write' access. A commit summary box shows the commit message 'Display command when it's unknown' by 'Tauop (author)' on 'April 04, 2011'. To the right of the commit message, the commit hash '29eadb9851a3584b74f' is shown, along with its parent hash '025538739a5000a26048'. At the bottom, a table header for commit history is visible with columns for 'name', 'age', 'message', and 'history'.

github SOCIAL CODING

Tauop | Dashboard | Inbox 0 | Account Settings | Log Out

Explore GitHub | Gist | Blog | Help | Search GitHub...

Tauop / ScriptHelper

Admin | Unwatch | Pull Request | 3 | 2

Source | Commits | Network | Pull Requests (0) | Fork Queue | Issues (2) | Wiki (0) | Graphs | Branch: master

Switch Branches (1) | Switch Tags (0) | Branch List

Shell libraries to help writing shell script — Read more
click here to add a homepage

Downloads

SSH | HTTP | Git Read-Only | git@github.com:Tauop/ScriptHelper.git | This URL has Read+Write access

Display command when it's unknown

Tauop (author)
April 04, 2011

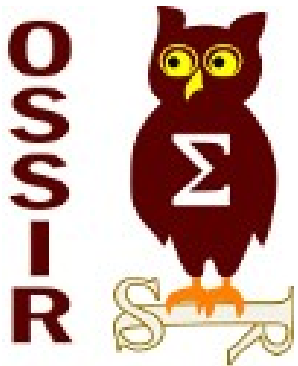
commit 29eadb9851a3584b74f
tree 77af569ed2c1baf211b3
parent 025538739a5000a26048

ScriptHelper /

name	age	message	history
------	-----	---------	---------

Participer au développement

- SshGate : <http://www.github.com/Tauop/sshGate>
- ScriptHelper : <http://www.github.com/Tauop/ScriptHelper>
- Venez nous voir sur IRC@Freenode #linagora
- Contact : pguiran@linagora.com



Merci de votre attention

Contact :

LINAGORA - Siège social
80, rue Roque de Fillol
92800 PUTEAUX
FRANCE

Tél. : 0 810 251 251 (tarif local)

Fax : +33 (0)1 46 96 63 64

Mail : info@linagora.com

Web : www.linagora.com

WWW.LINAGORA.COM