

Bastion SSH avec sshproxy

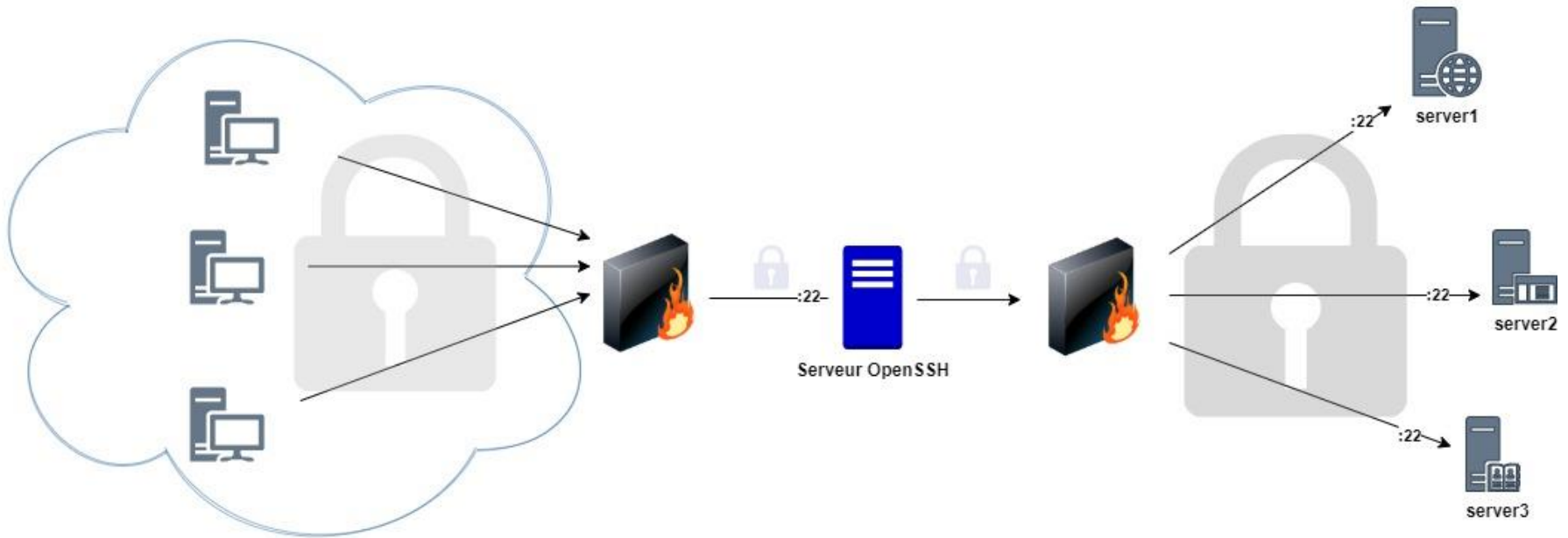


Capitoul – 11 Février 2021
rosalie.viala@ut-capitole.fr

SOMMAIRE

- Schéma de principe : Bastion SSH
- Avantage d'un bastion SSH avec sshproxy
- Schéma de principe : Bastion SSH avec sshproxy
- L'outil sshproxy en détail
- Fichier de configuration
- Mise en œuvre à l'UT1-C
- Intégration sshproxy
- RETEX après un an

Schéma de principe : Bastion SSH



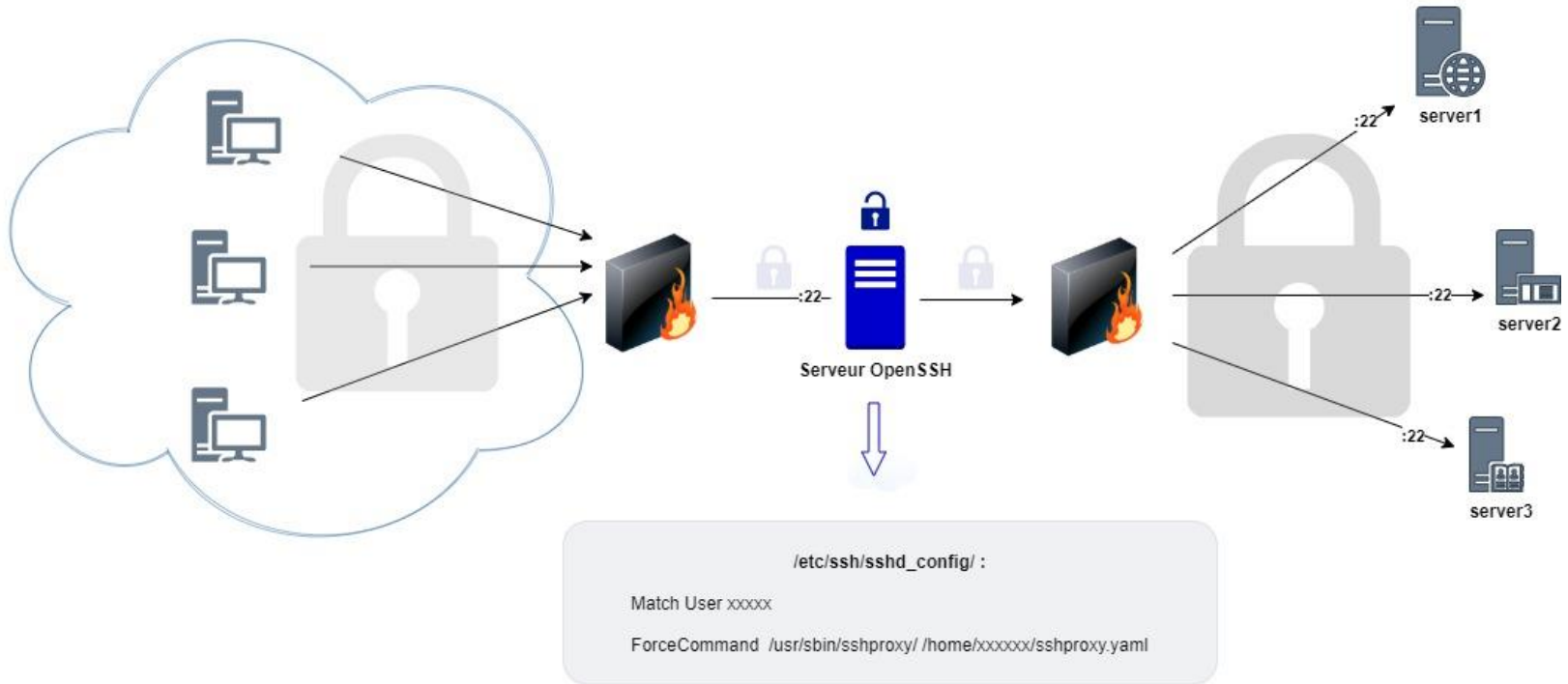
Avantage d'un bastion SSH avec sshproxy

- Augmenter la sécurité
- Rationaliser les accès SSH
- Qui a accès à quoi ?

Les + avec sshproxy :

- Connexion transparente
- Enregistrer une session SSH
- SFTP, SCP, RSYNC
- Suivi des connexions

Schéma de principe : Bastion SSH avec sshproxy



L'outil sshproxy en détail

- Disponible sur github : <https://github.com/cea-hpc/sshproxy>
- Développé en Go
- Prédéfinir des routes SSH

- Deux modes :
 - stateful via etcd → collecte statistiques + contrôle connexions
 - sshproxctl
 - stateless
- sshproxy-dumpd
- sshproxy-replay

- Installation : compilateur Go + make && make install
- Configuration : sshproxy.yaml

Fichier de configuration

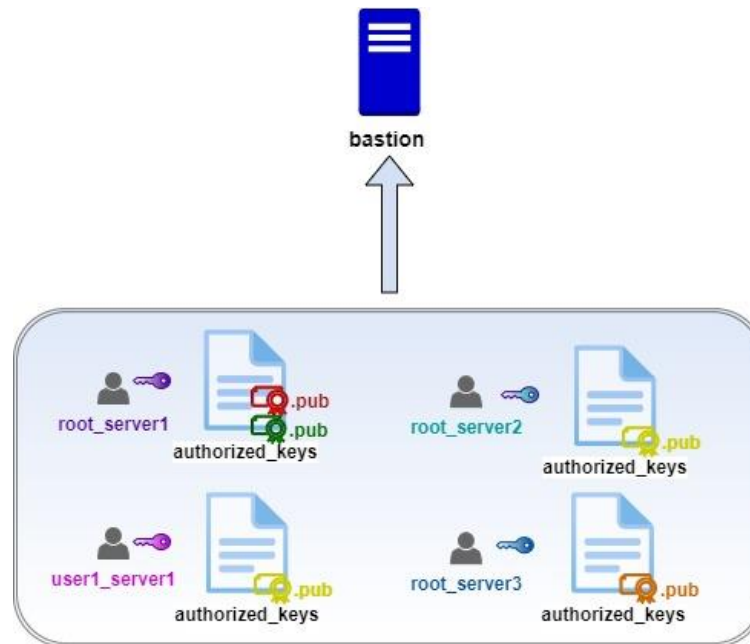
- Pour chaque utilisateur sur le serveur bastion, un fichier de configuration sshproxy.yaml :

```
---  
# Debug mode  
debug: true  
log: "/var/log/sshproxy/{user}.log"  
dump: "/var/log/sshproxy/dumps/{user}/{time}-{sid}.dump"  
dump_limit_size: 1048576  
environment:  
  XAUTHORITY: /home/{user}/.Xauthority  
ssh:  
  args: ["-l", "root", "-Y", "-q"]  
routes:  
  default:  
    dest: ["server1"]
```

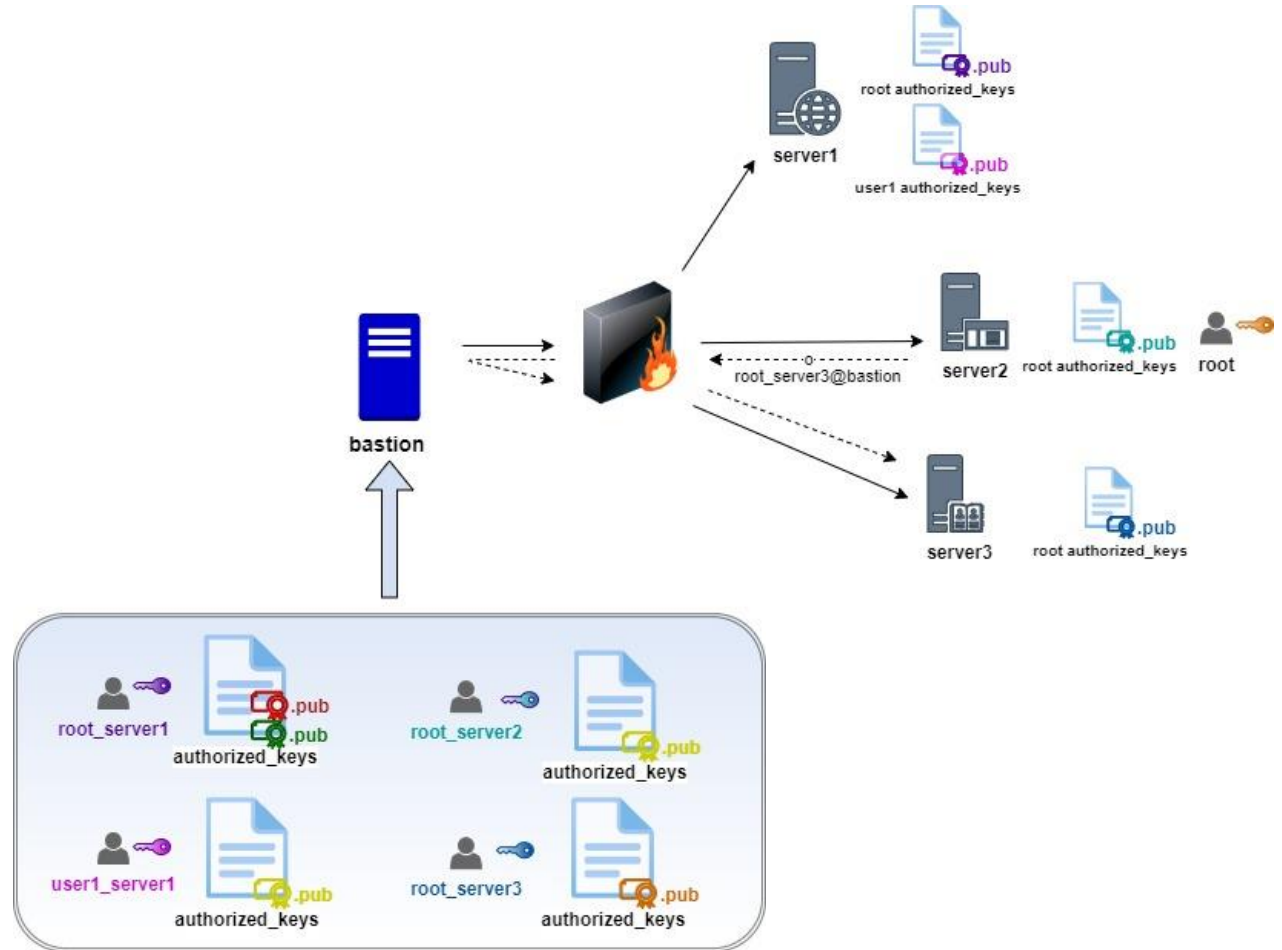
Mise en œuvre à l'UT1-C



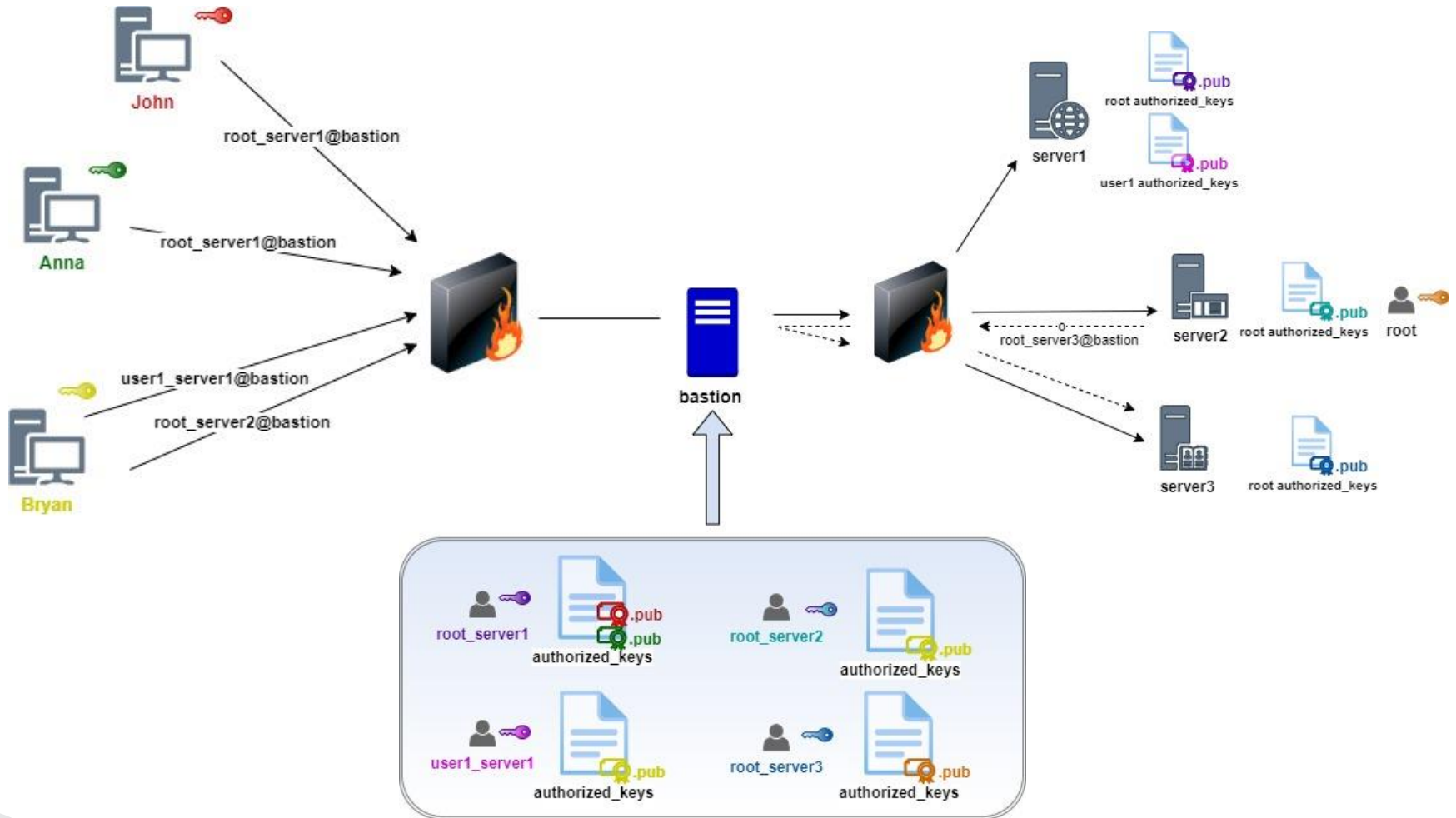
Mise en œuvre à l'UT1-C



Mise en œuvre à l'UT1-C



Mise en œuvre à l'UT1-C

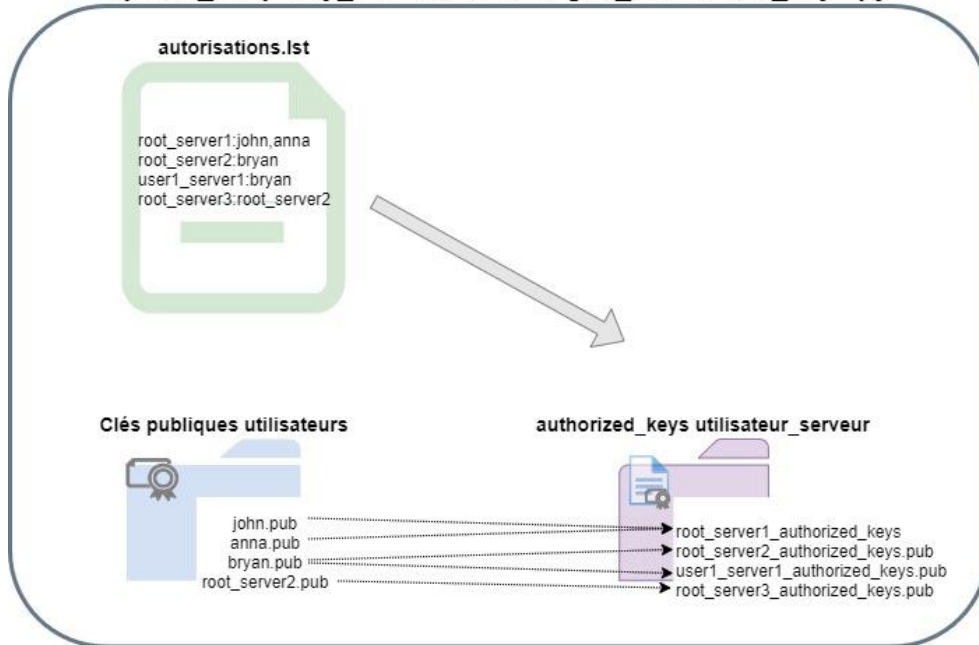


Intégration sshproxy



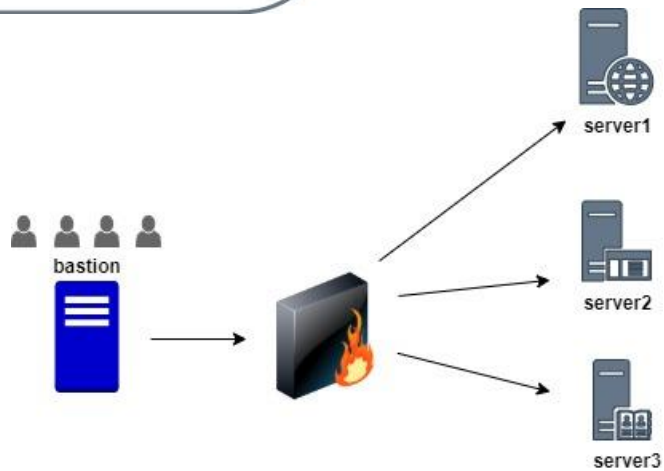
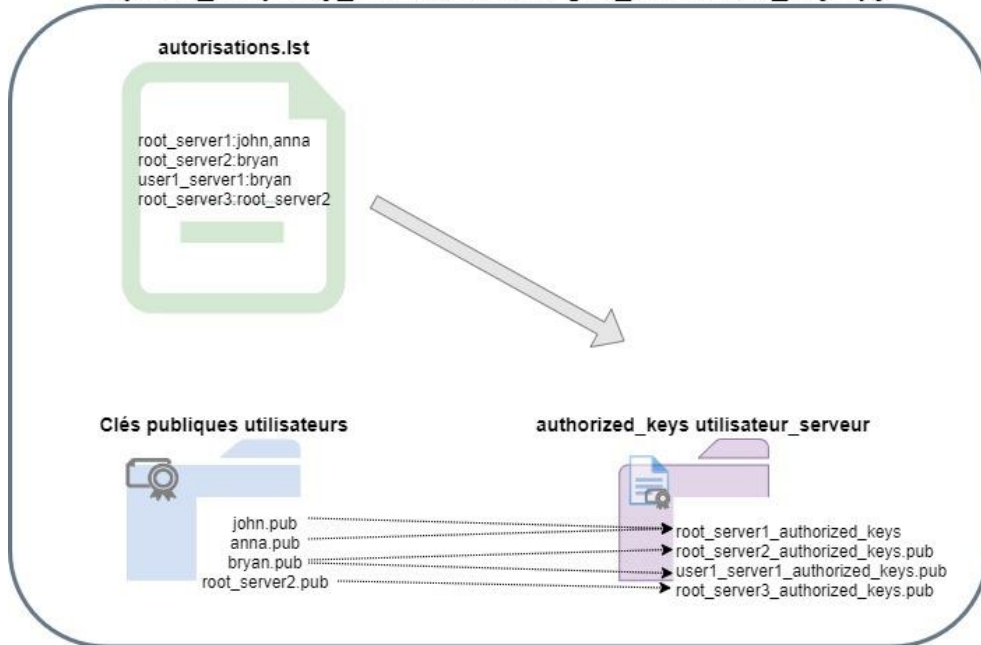
Intégration sshproxy

update_sshproxy_connection.sh / gen_authorized_keys.py



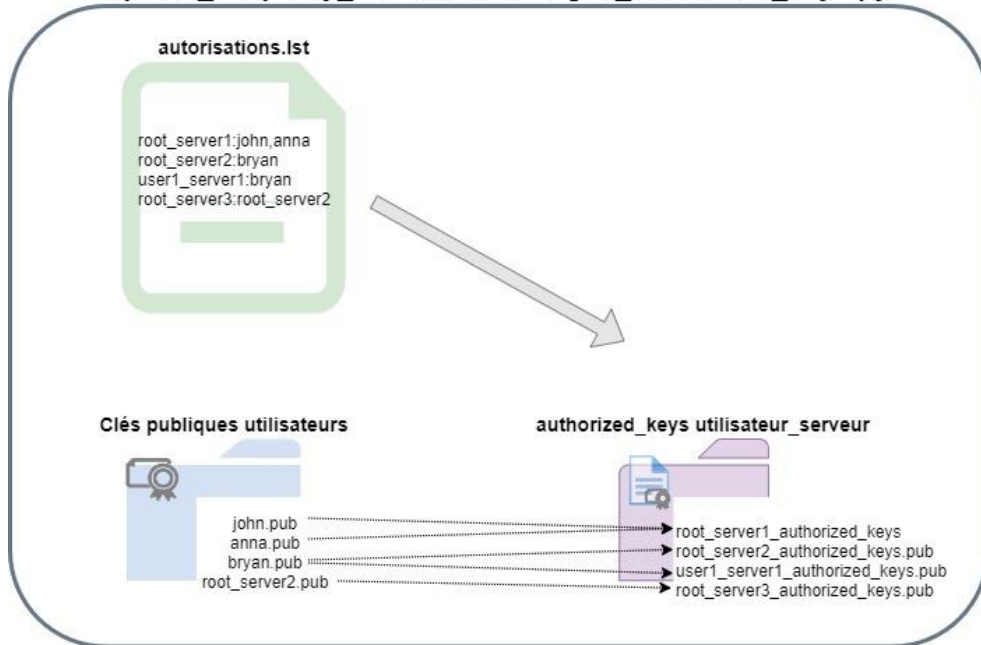
Intégration sshproxy

update_sshproxy_connection.sh / gen_authorized_keys.py

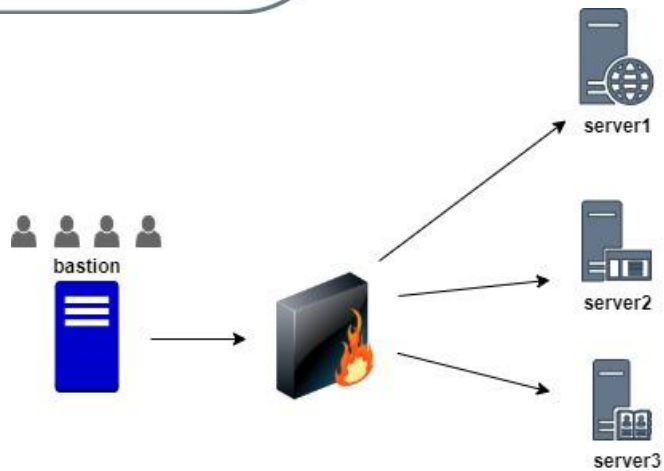


Intégration sshproxy

update_sshproxy_connection.sh / gen_authorized_keys.py

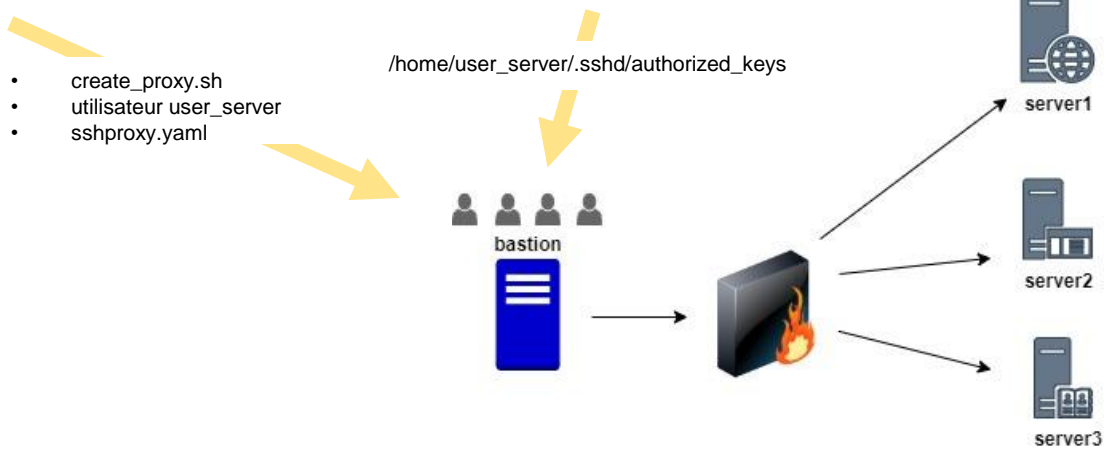
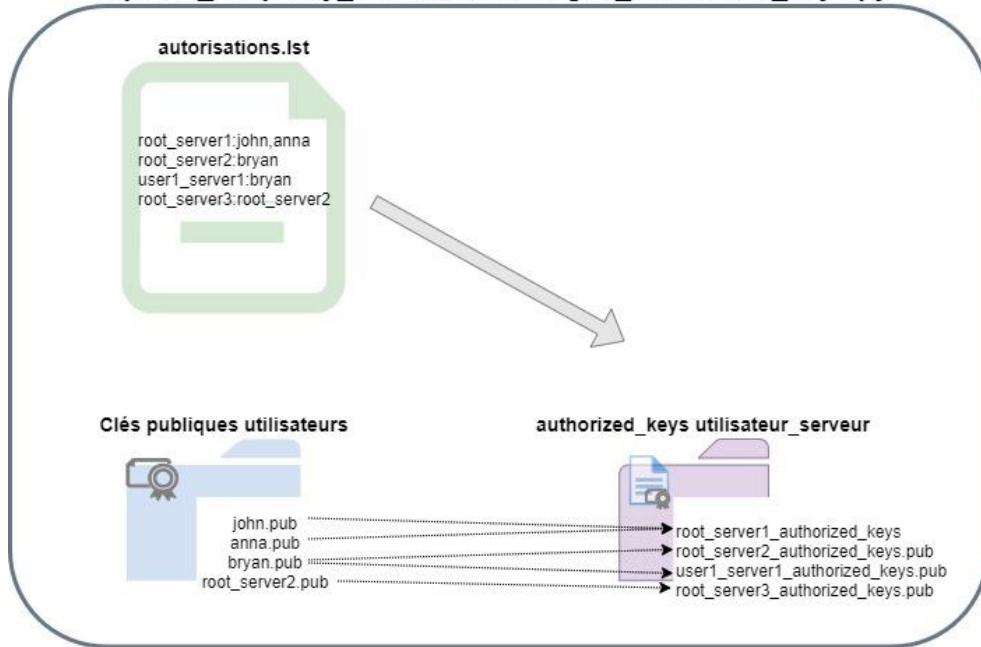


- create_proxy.sh
- utilisateur user_server
- sshproxy.yaml



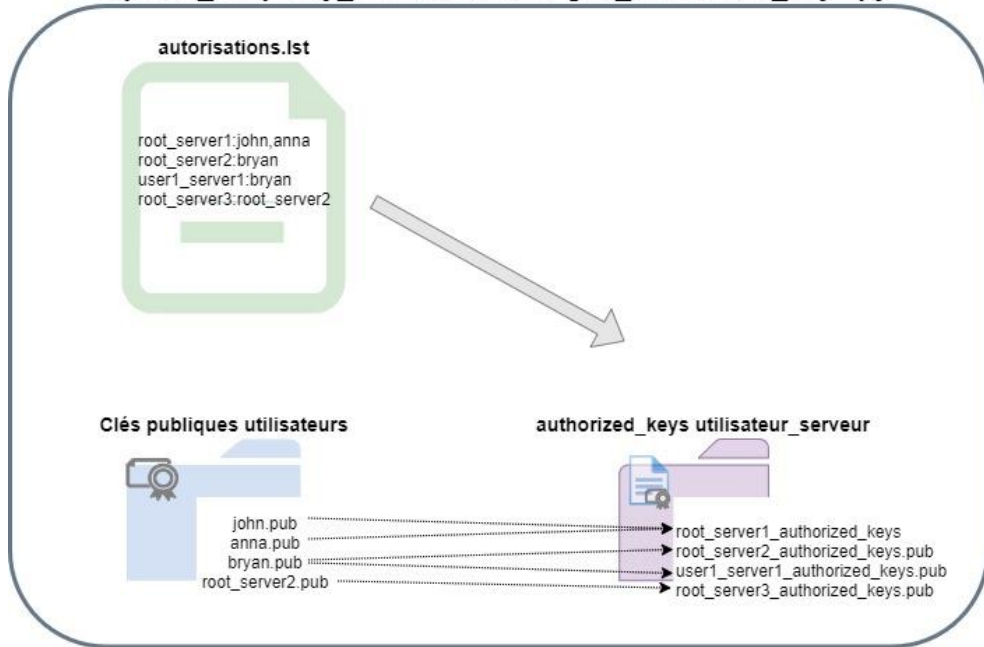
Intégration sshproxy

update_sshproxy_connection.sh / gen_authorized_keys.py



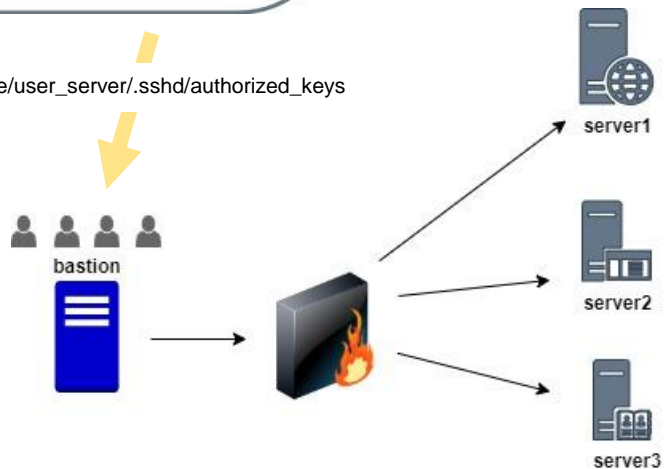
Intégration sshproxy

update_sshproxy_connection.sh / gen_authorized_keys.py



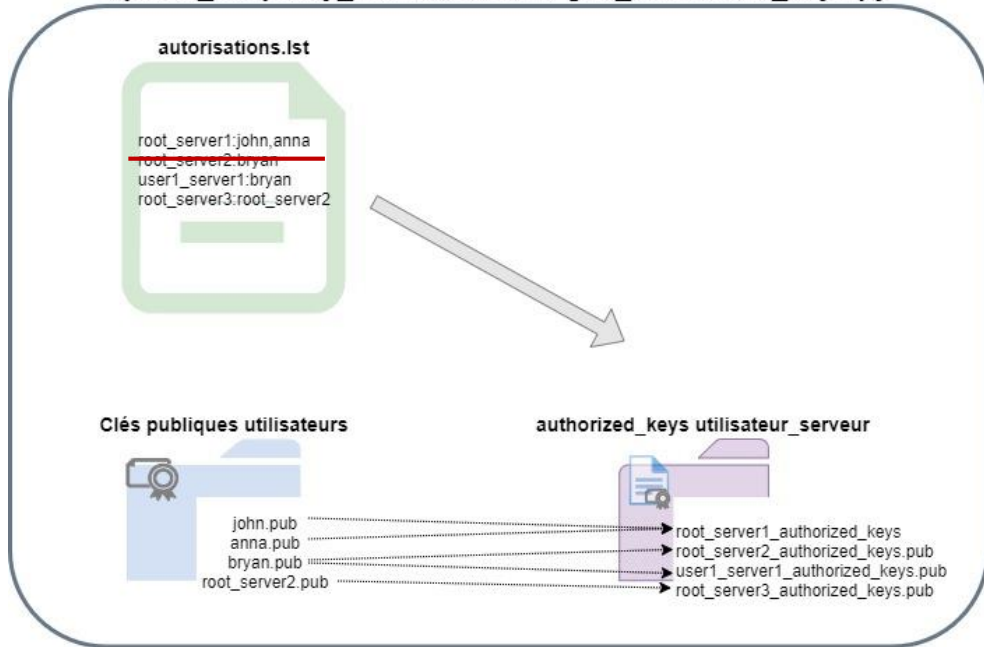
- create_proxy.sh
- utilisateur user_server
- sshproxy.yaml

/home/user_server/.ssh/authorized_keys



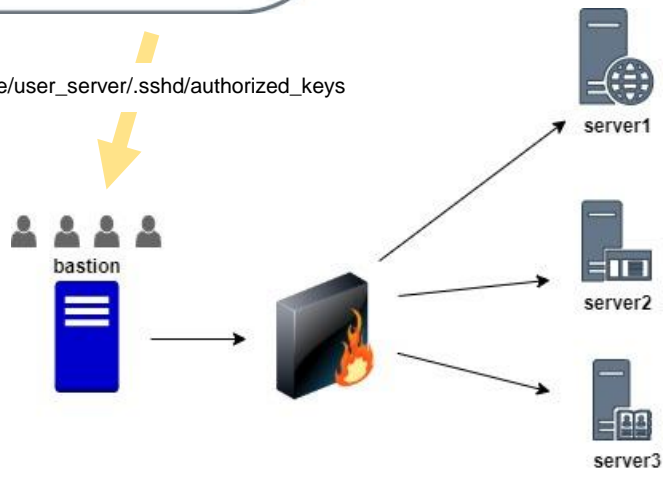
Intégration sshproxy

update_sshproxy_connection.sh / gen_authorized_keys.py



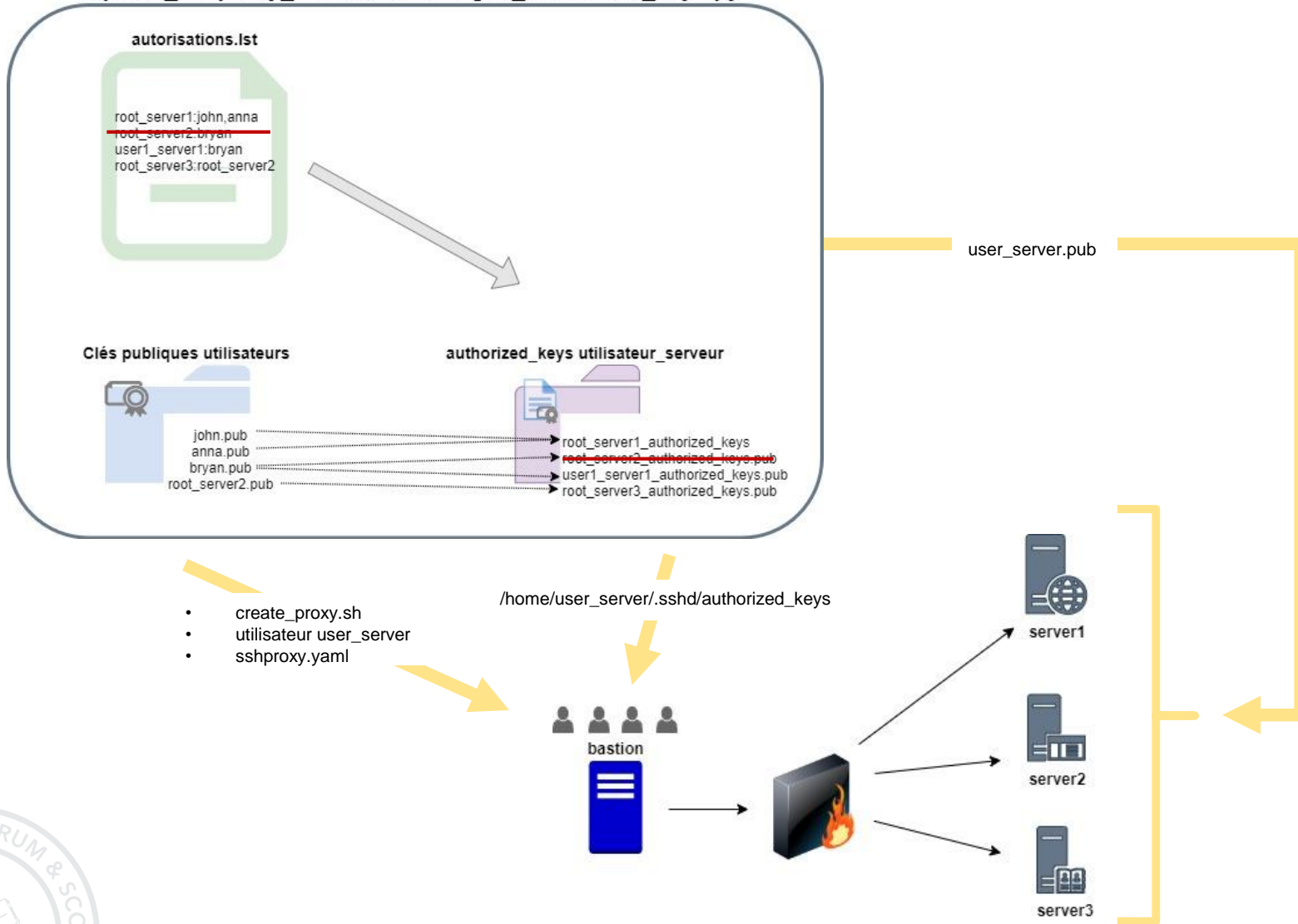
- create_proxy.sh
- utilisateur user_server
- sshproxy.yaml

/home/user_server/.ssh/authorized_keys



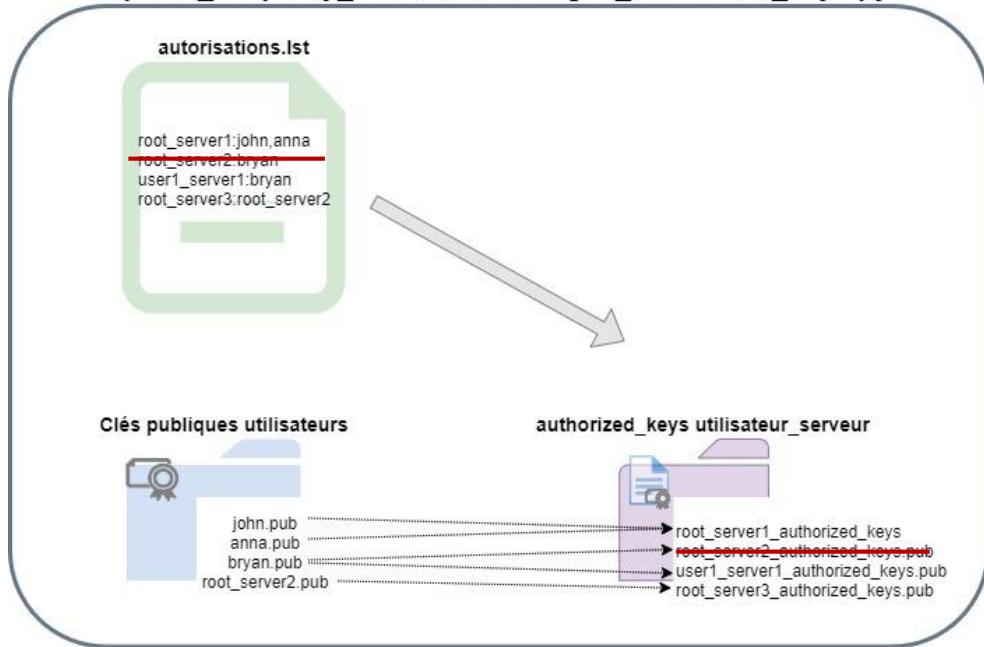
Intégration sshproxy

update_sshproxy_connection.sh / gen_authorized_keys.py



Intégration sshproxy

update_sshproxy_connection.sh / gen_authorized_keys.py



- create_proxy.sh
- utilisateur user_server
- sshproxy.yaml

/home/user_server/.ssh/authorized_keys

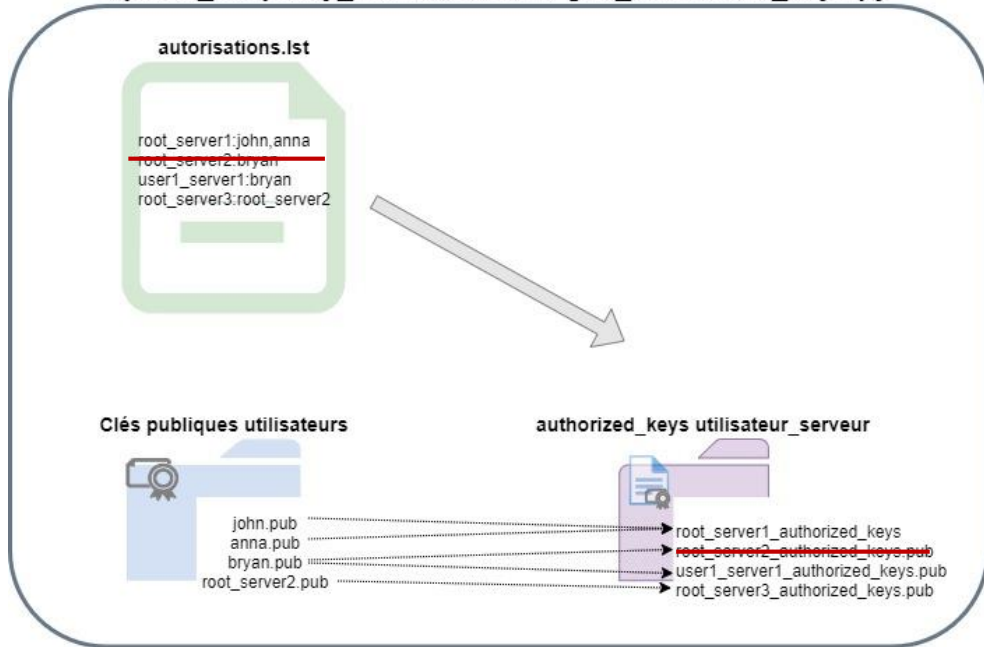


user_server.pub



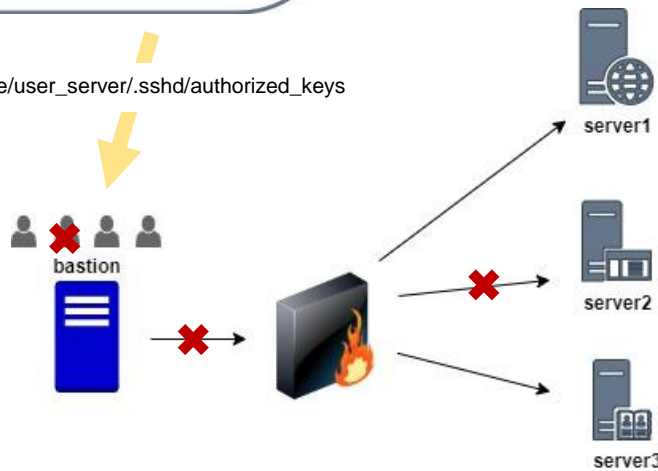
Intégration sshproxy

update_sshproxy_connection.sh / gen_authorized_keys.py



- create_proxy.sh
- utilisateur user_server
- sshproxy.yaml

/home/user_server/.ssh/authorized_keys



RETEX après un an

- Simplifier les procédures
- Utilisateurs intra DSI
- Fichier d'autorisation publié
- Prestataire extérieur
- Accès SFTP
- Problème taille fichier dump
- Bug rsync
- J'ai oublié ma passphrase



Questions ?

