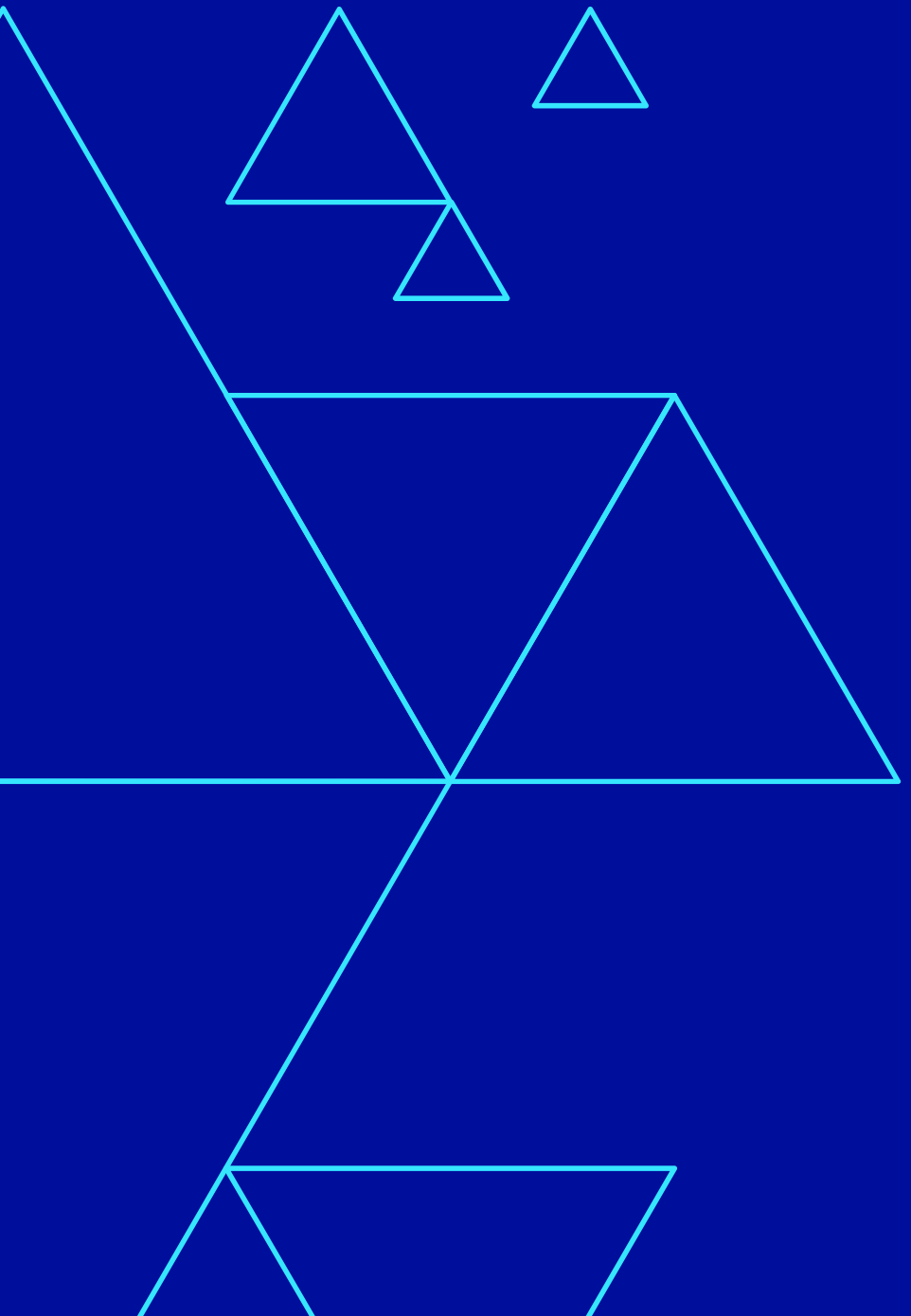


OVHcloud

The Bastion @OSSIR

Stéphane Lesimple

2021-05-11

$ whoami --short

▶ Chez OVHcloud depuis 2010

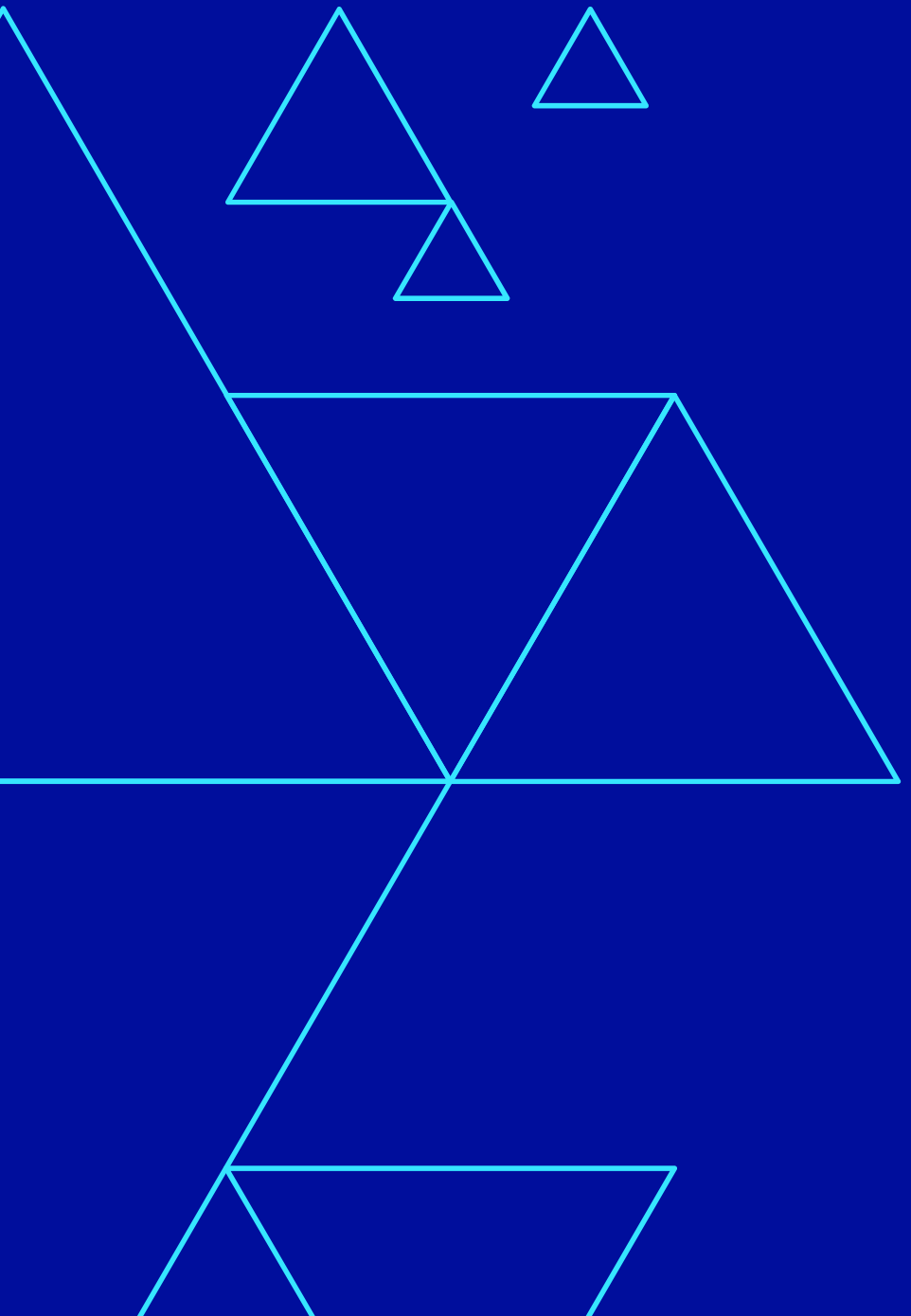
- Sysadmin sur les systèmes critiques de la DSI (2010)
- Responsable du département noms de domaine (2011-2013)
- Responsable du département serveurs dédiés (2013-2016)
- Responsable de l'équipe « outils » au sein du département sécurité (depuis 2016)

▶ Sysadmin / Dev


speed47


speedux

Sommaire



OVHcloud

Genèse

Problématiques / Challenges

- Axiome : SSH est omniprésent

Problématiques / Challenges

- Axiome : SSH est omniprésent

- Principales méthodes d'authentification SSH :

  - Auth. par mot de passe (legacy ou via PAM)
  - Réutilisation, bruteforce, phishing, complexité vs post-it, …

Problématiques / Challenges

- Axiome : SSH est omniprésent

- Principales méthodes d'authentification SSH :

- Auth. par mot de passe (legacy ou via PAM)
- Réutilisation, bruteforce, phishing, complexité vs post-it, …

- Auth. par clé publique
- La relation de confiance est en 1:1 → mise à l'échelle?

Problématiques / Challenges

- Axiome : SSH est omniprésent

- Principales méthodes d'authentification SSH :

- Auth. par mot de passe (legacy ou via PAM)
  - Réutilisation, bruteforce, phishing, complexité vs post-it, …

- Auth. par clé publique
  - La relation de confiance est en 1:1 → mise à l'échelle?

- Auth. par certificat
  - Complexité de mise en œuvre d'une PKI
  - Très centralisé (admins CA tout puissants)
  - Compatibilité

Fonctionnalités attendues

- Délégation et accès
  - Pas d’équipe centralisée superpuissante qui « gère les accès »
  - Les techleads doivent être autonomes sur leur périmètre
  - Les arrivées / départs / mouvements inter-équipes doivent être simples
  - Accès temporaires

Fonctionnalités attendues

▶ Délégation et accès

- Pas d'équipe centralisée superpuissante qui « gère les accès »
- Les techleads doivent être autonomes sur leur périmètre
- Les arrivées / départs / mouvements inter-équipes doivent être simples
- Accès temporaires

▶ Auditabilité / Traçabilité

- Enregistrement des sessions interactives (ttyrec)
- Journalisation des (tentatives d') actions et accès
- Intégration facile dans un SIEM

Fonctionnalités attendues

▶ Délégation et accès

- Pas d'équipe centralisée superpuissante qui « gère les accès »
- Les techleads doivent être autonomes sur leur périmètre
- Les arrivées / départs / mouvements inter-équipes doivent être simples
- Accès temporaires

▶ Auditabilité / Traçabilité

- Enregistrement des sessions interactives (ttyrec)
- Journalisation des (tentatives d') actions et accès
- Intégration facile dans un SIEM

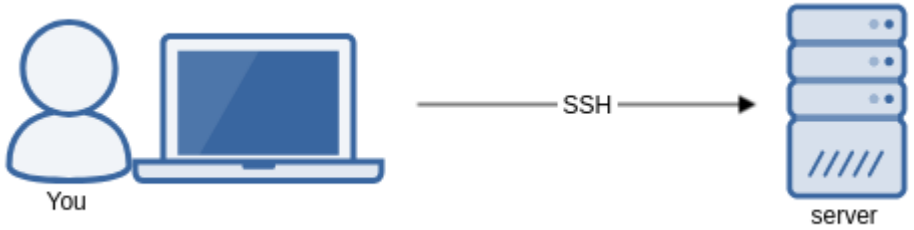
▶ Securité / Résilience

- Plus de sécurité qu'un accès SSH standard, sans changer l'expérience
- Doit dépendre d'un minimum de briques possibles

OVHcloud

Fonctionnement global

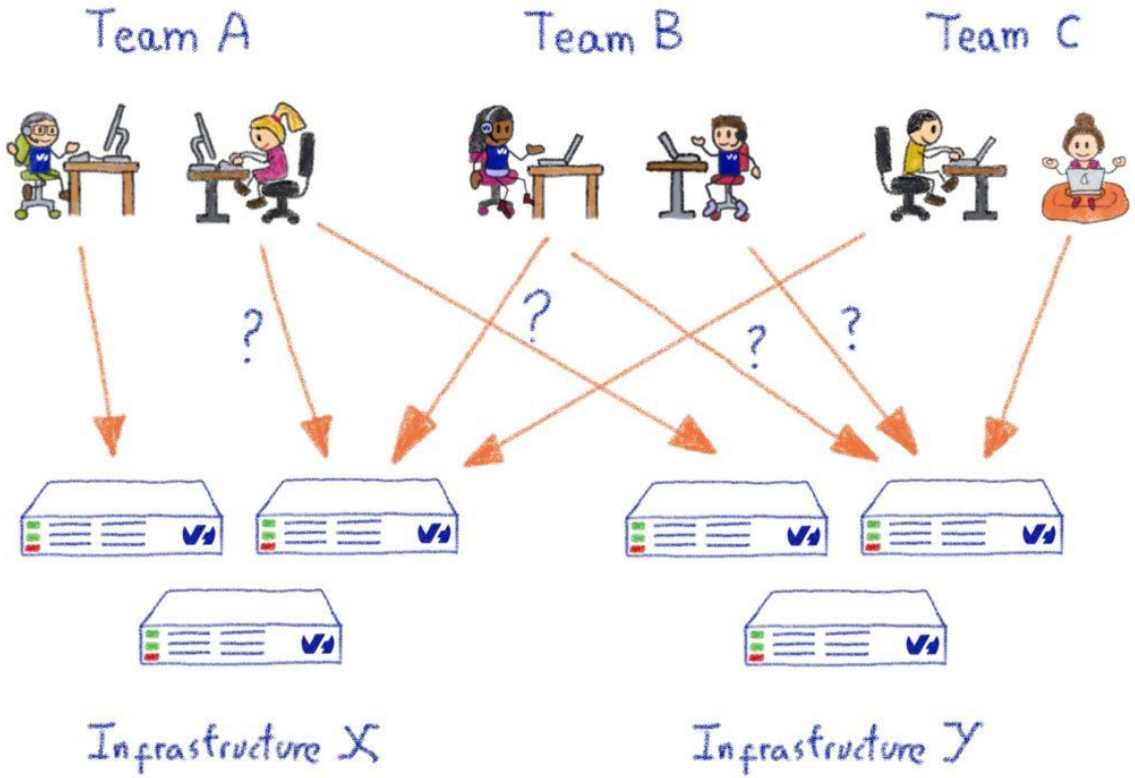
Connexion vers un serveur – sans The Bastion



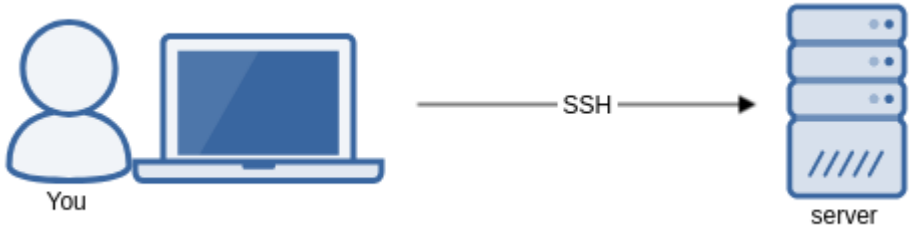
- Connexion « ad-hoc »

Connexion vers un serveur – sans The Bastion

- Connexion « ad-hoc »
- **N** utilisateurs x **M** serveurs, problème de mise à l'échelle

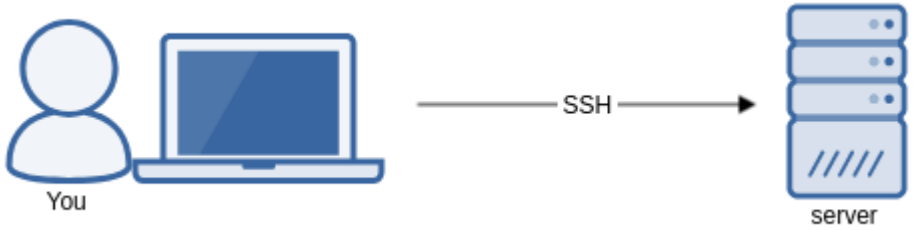


Connexion vers un serveur – sans The Bastion



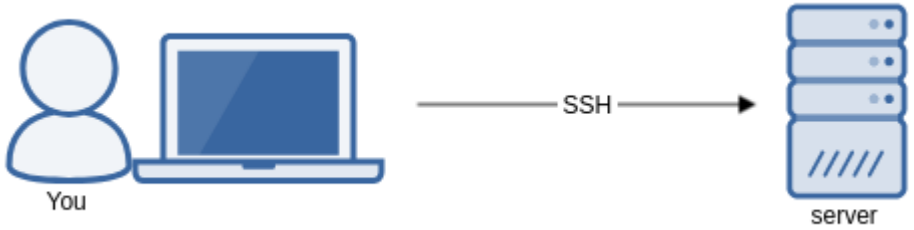
- Connexion « ad-hoc »
- **N** utilisateurs x **M** serveurs, problème de mise à l'échelle
- Quand quelqu'un quitte votre équipe, vous allez penser à supprimer ses clés partout

Connexion vers un serveur – sans The Bastion



- Connexion « ad-hoc »
- **N** utilisateurs x **M** serveurs, problème de mise à l'échelle
- Quand quelqu'un quitte votre équipe, vous allez penser à supprimer ses clés partout

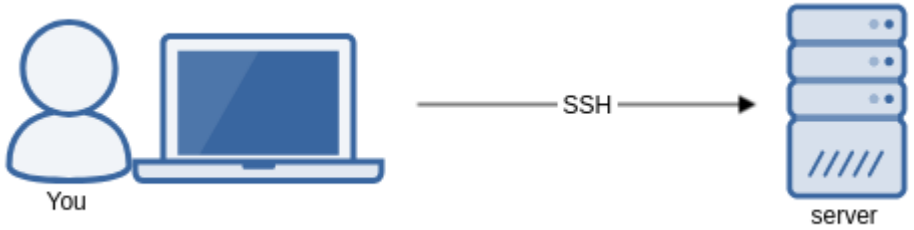
Connexion vers un serveur – sans The Bastion



- Connexion « ad-hoc »
- **N** utilisateurs x **M** serveurs, problème de mise à l'échelle
- Quand quelqu'un quitte votre équipe, vous allez penser à supprimer ses clés partout

- N'importe qui peut ajouter n'importe quelle clé SSH sur le serveur distant pour donner accès à n'importe qui
  - Qui est « responsable » ? Personne

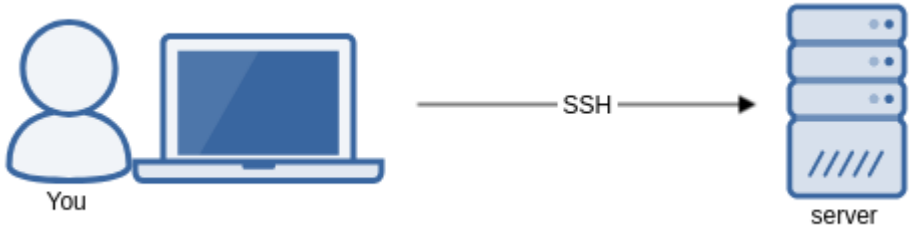
Connexion vers un serveur – sans The Bastion



- Connexion « ad-hoc »
- **N** utilisateurs x **M** serveurs, problème de mise à l'échelle
- Quand quelqu'un quitte votre équipe, vous allez penser à supprimer ses clés partout

- N'importe qui peut ajouter n'importe quelle clé SSH sur le serveur distant pour donner accès à n'importe qui
  - Qui est « responsable » ? Personne
- Pouvez-vous répondre aux questions suivantes:
  - *"Pouvez-vous me transmettre l'activité de johndoe sur la dernière semaine sur tous les serveurs de votre infrastructure?"*
  - *"Qui a accès au serveur X?"*
  - *"Qui avait accès au serveur X il y a 6 mois ?"*
  - *"Qui a ajouté cette clé SSH sur le compte root du serveur ?"*

Connexion vers un serveur – sans The Bastion

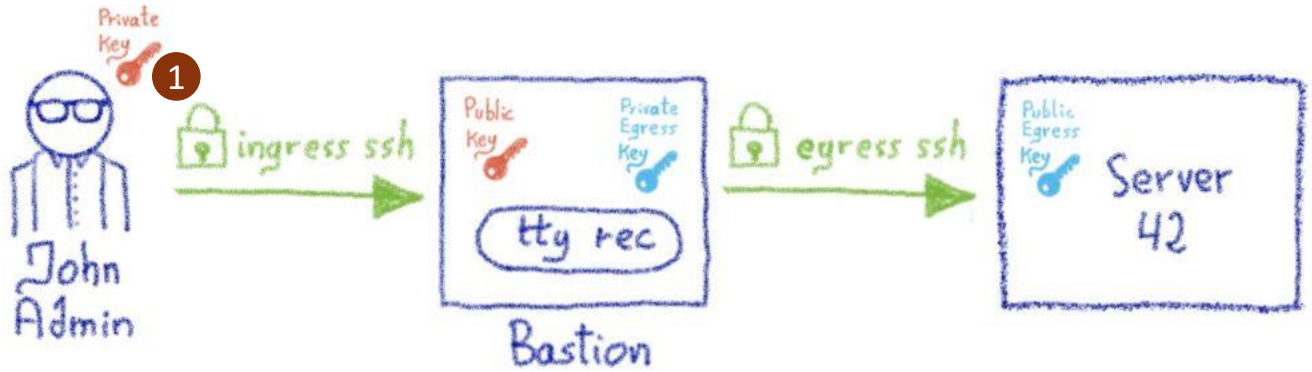


- Connexion « ad-hoc »
- **N** utilisateurs x **M** serveurs, problème de mise à l'échelle
- Quand quelqu'un quitte votre équipe, vous allez penser à supprimer ses clés partout

- N'importe qui peut ajouter n'importe quelle clé SSH sur le serveur distant pour donner accès à n'importe qui
  - Qui est « responsable » ? Personne
- Pouvez-vous répondre aux questions suivantes:
  - *"Pouvez-vous me transmettre l'activité de johndoe sur la dernière semaine sur tous les serveurs de votre infrastructure?"*
  - *"Qui a accès au serveur X?"*
  - *"Qui avait accès au serveur X il y a 6 mois ?"*
  - *"Qui a ajouté cette clé SSH sur le compte root du serveur ?"*

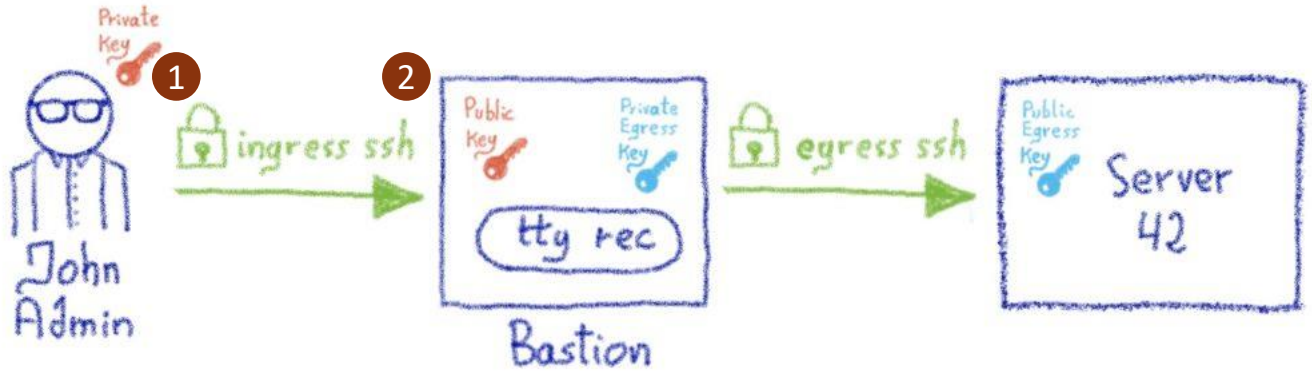
A quoi sert cette clé sur le compte root déjà ? Je vais la laisser, on sait jamais si ça casse quelque chose…
- Jean-Charles DevOps, 2021

Connexion vers un serveur – avec The Bastion



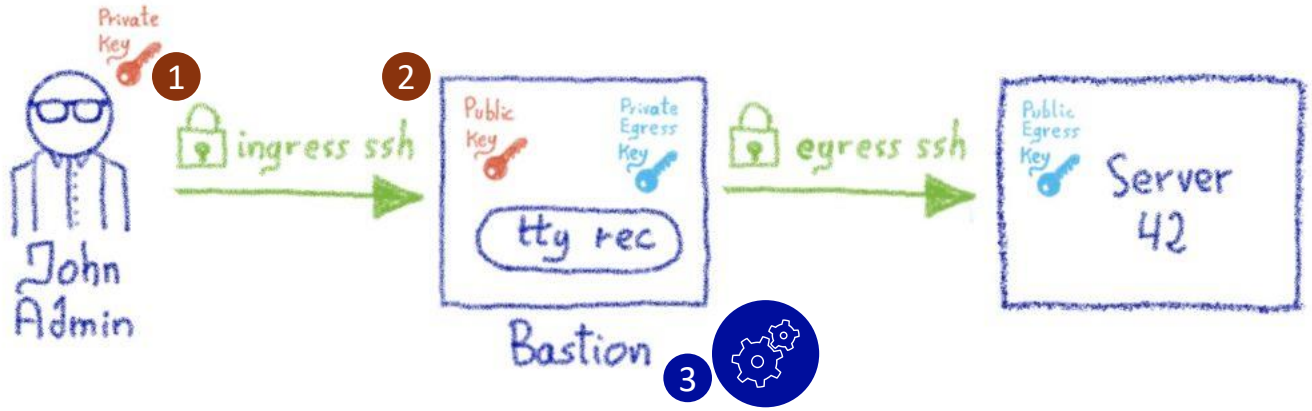
- L'utilisateur se connecte au bastion avec sa clé SSH (privée) stockée sur son PC portable

Connexion vers un serveur – avec The Bastion



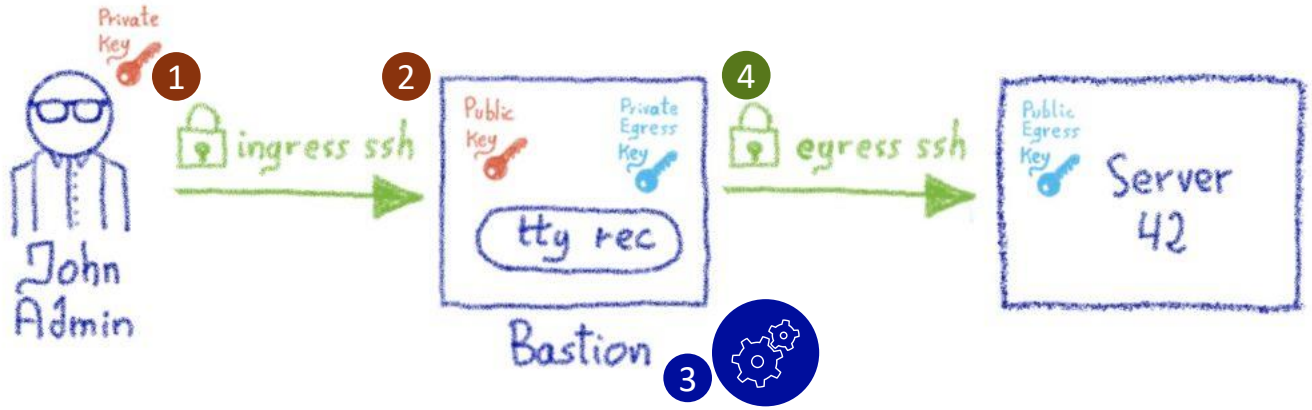
- L'utilisateur se connecte au bastion avec sa clé SSH (privée) stockée sur son PC portable
- Le bastion connaît sa clé SSH (publique), et **authentifie** l'utilisateur

Connexion vers un serveur – avec The Bastion



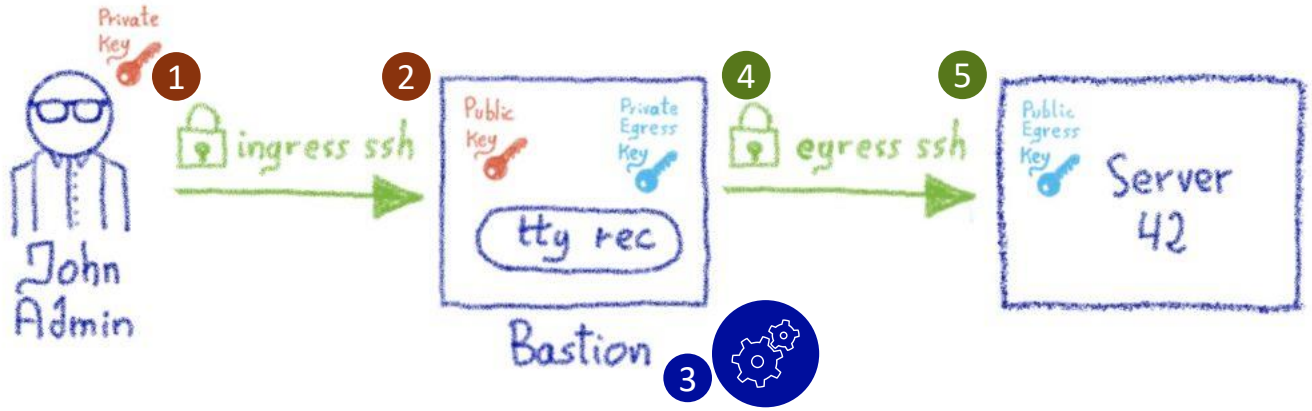
- L'utilisateur se connecte au bastion avec sa clé SSH (privée) stockée sur son PC portable
- Le bastion connaît sa clé SSH (publique), et **authentifie** l'utilisateur
- Le bastion vérifie si l'utilisateur est **autorisé** à se connecter au serveur distant

Connexion vers un serveur – avec The Bastion



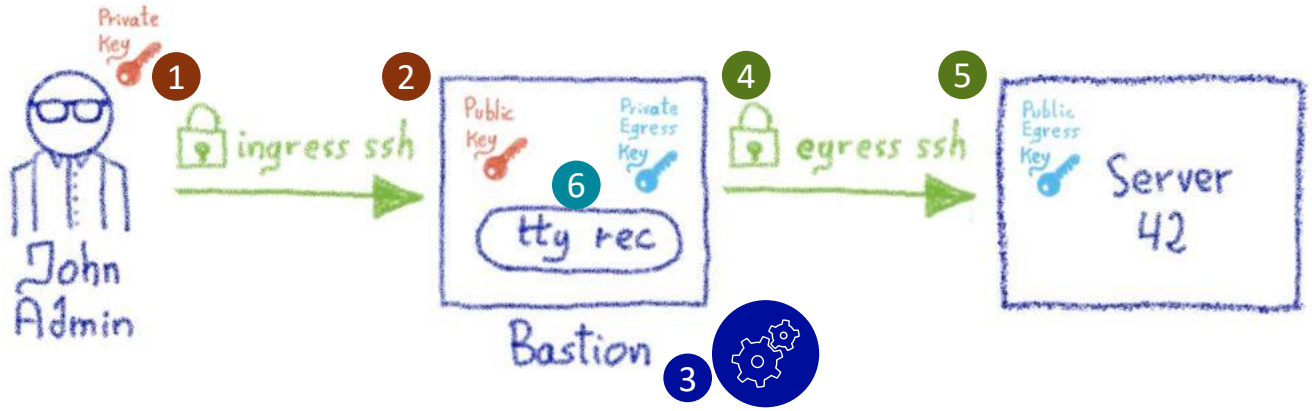
- L'utilisateur se connecte au bastion avec sa clé SSH (privée) stockée sur son PC portable
- Le bastion connaît sa clé SSH (publique), et **authentifie** l'utilisateur
- Le bastion vérifie si l'utilisateur est **autorisé** à se connecter au serveur distant
- Le bastion se connecte au serveur distant *pour le compte de l'utilisateur*

Connexion vers un serveur – avec The Bastion



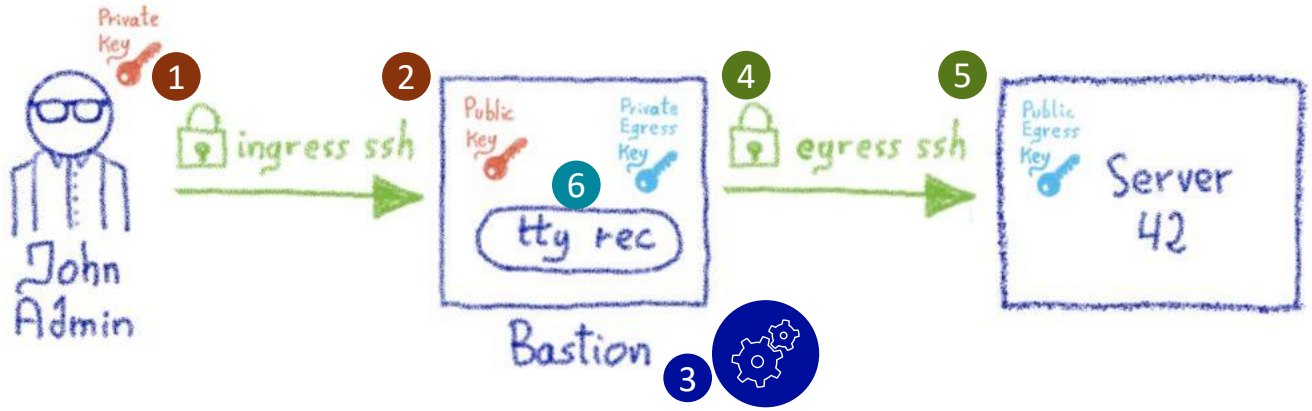
- L'utilisateur se connecte au bastion avec sa clé SSH (privée) stockée sur son PC portable
- Le bastion connaît sa clé SSH (publique), et **authentifie** l'utilisateur
- Le bastion vérifie si l'utilisateur est **autorisé** à se connecter au serveur distant
- Le bastion se connecte au serveur distant *pour le compte de l'utilisateur*
- Le serveur distant connaît la clé privée egress bastion de l'utilisateur et autorise la connexion

Connexion vers un serveur – avec The Bastion



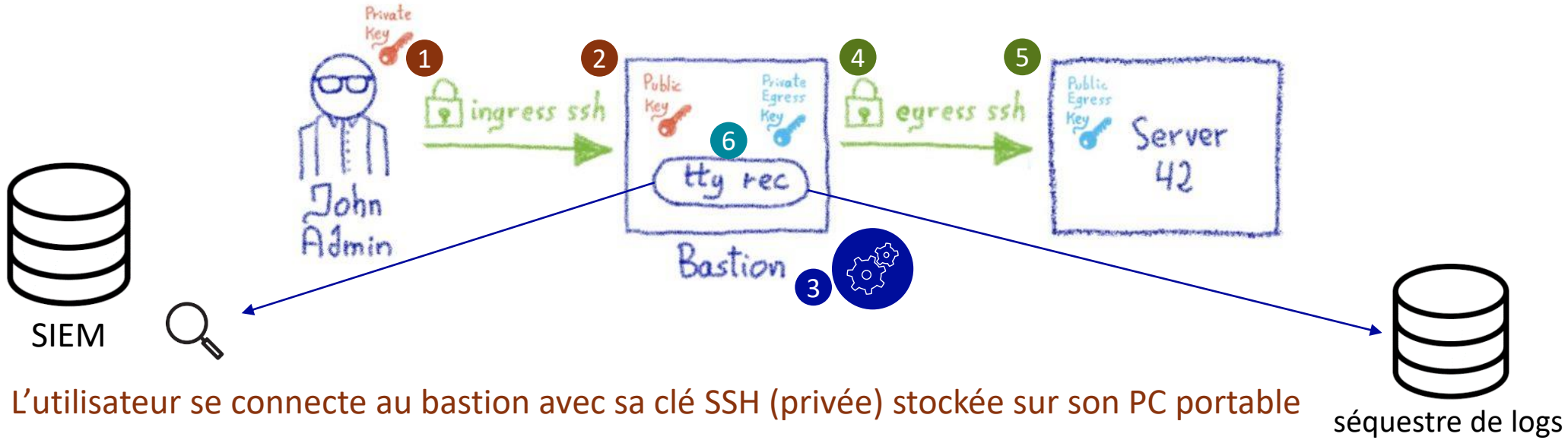
- L'utilisateur se connecte au bastion avec sa clé SSH (privée) stockée sur son PC portable
- Le bastion connaît sa clé SSH (publique), et **authentifie** l'utilisateur
- Le bastion vérifie si l'utilisateur est **autorisé** à se connecter au serveur distant
- Le bastion se connecte au serveur distant *pour le compte de l'utilisateur*
- Le serveur distant connaît la clé privée egress bastion de l'utilisateur et autorise la connexion
- Le bastion fusionne les 2 connexions, l'utilisateur a l'*impression d'être connecté directement* au serveur distant

Connexion vers un serveur – avec The Bastion



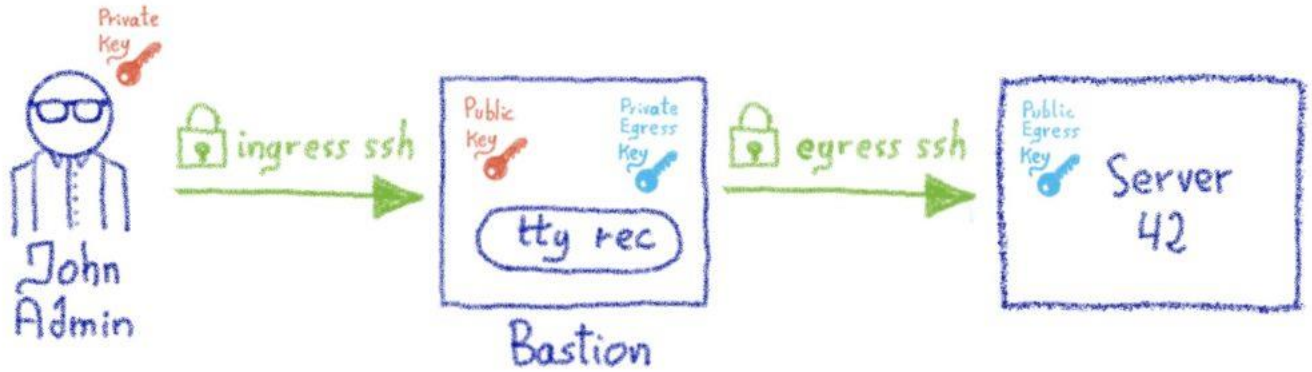
- L'utilisateur se connecte au bastion avec sa clé SSH (privée) stockée sur son PC portable
- Le bastion connaît sa clé SSH (publique), et **authentifie** l'utilisateur
- Le bastion vérifie si l'utilisateur est **autorisé** à se connecter au serveur distant
- Le bastion se connecte au serveur distant *pour le compte de l'utilisateur*
- Le serveur distant connaît la clé privée egress bastion de l'utilisateur et autorise la connexion
- Le bastion fusionne les 2 connexions, l'utilisateur a l'*impression d'être connecté directement* au serveur distant
- Le bastion journalise la demande de connexion, et enregistre la session (ttyrec)

Connexion vers un serveur – avec The Bastion



- L'utilisateur se connecte au bastion avec sa clé SSH (privée) stockée sur son PC portable
- Le bastion connaît sa clé SSH (publique), et **authentifie** l'utilisateur
- Le bastion vérifie si l'utilisateur est **autorisé** à se connecter au serveur distant
- Le bastion se connecte au serveur distant *pour le compte de l'utilisateur*
- Le serveur distant connaît la clé privée egress bastion de l'utilisateur et autorise la connexion
- Le bastion fusionne les 2 connexions, l'utilisateur a l'*impression d'être connecté directement* au serveur distant
- Le bastion journalise la demande de connexion, et enregistre la session (ttyrec)
- Les logs du bastion sont surveillés à la recherche d'évènement anormal

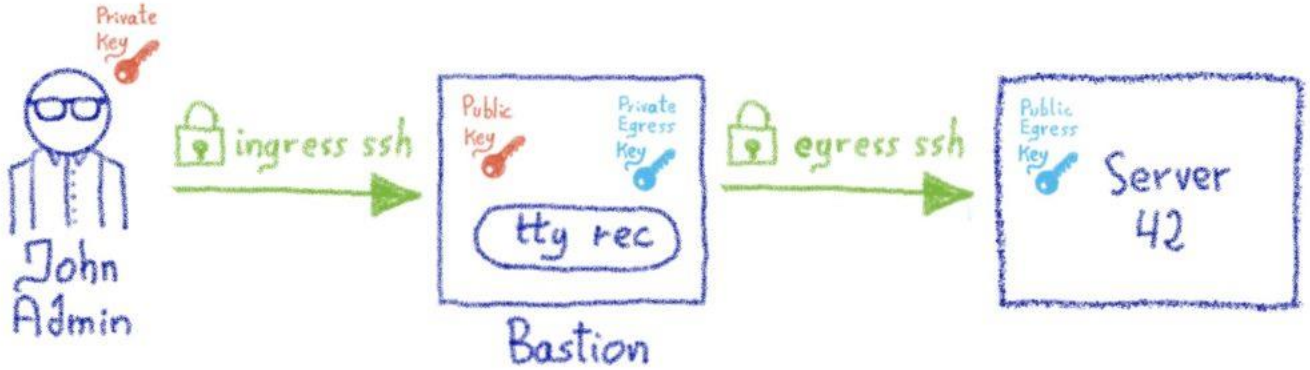
Connexion vers un serveur – avec The Bastion



- Conséquences de ce fonctionnement :

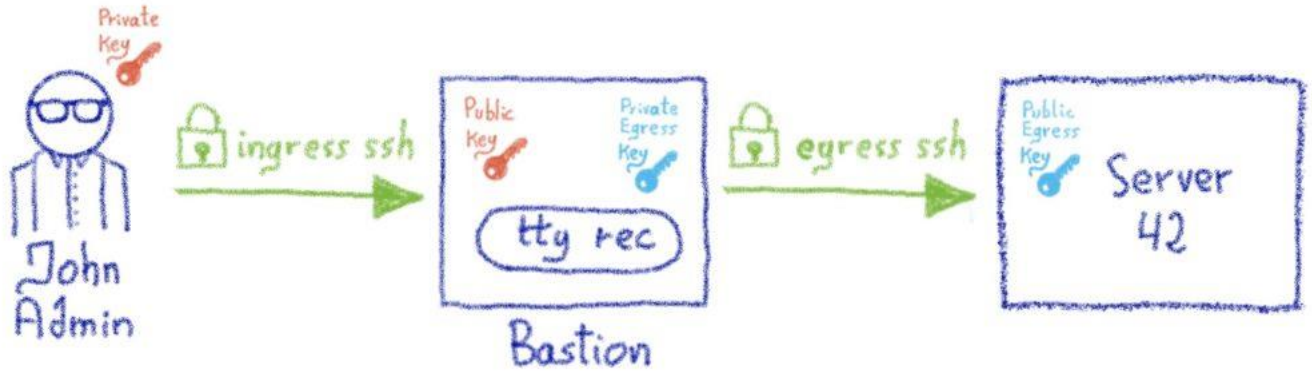
- Rien à installer sur le serveur distant **(+)**
- Possibilité d'enregistrer la session **(+)**
- Un seul endroit pour gérer les accès et avoir la source de traçabilité **(+)**
- Le bastion devient un SPOF potentiel **(-)** → clustering avec N machines
- Compatibilité de l'outillage qui utilise SSH en dessous **(-)** → wrappers ou bypass

Typologies d’accès : accès personnels



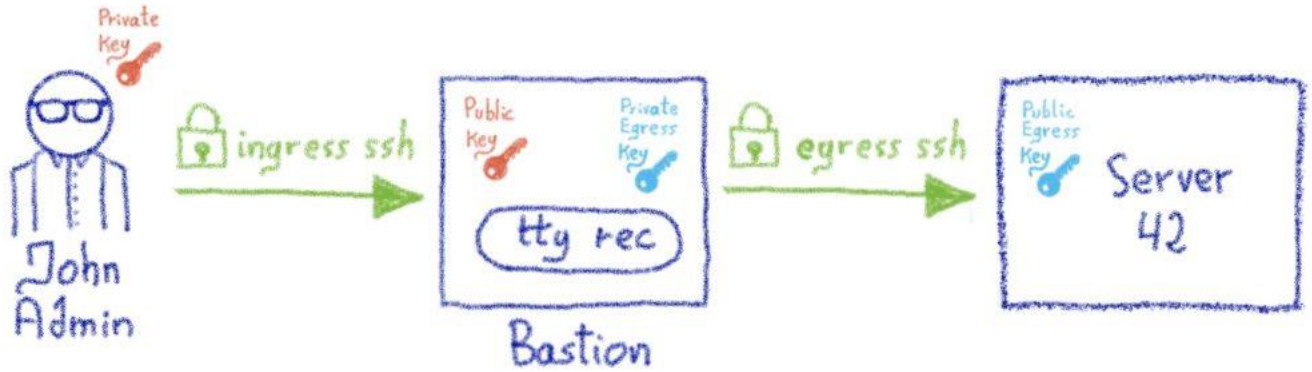
- Adapté pour les serveurs ou comptes distants personnels

Typologies d’accès : accès personnels



- Adapté pour les serveurs ou comptes distants personnels
- 2 façons de les gérer :

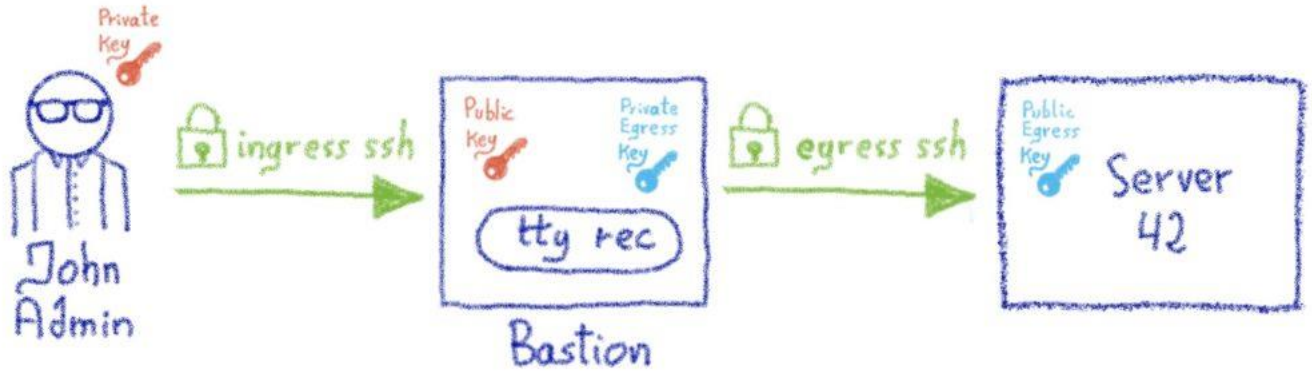
Typologies d’accès : accès personnels



- Adapté pour les serveurs ou comptes distants personnels
- 2 façons de les gérer :

- Self-service

Typologies d’accès : accès personnels



- Adapté pour les serveurs ou comptes distants personnels
- 2 façons de les gérer :

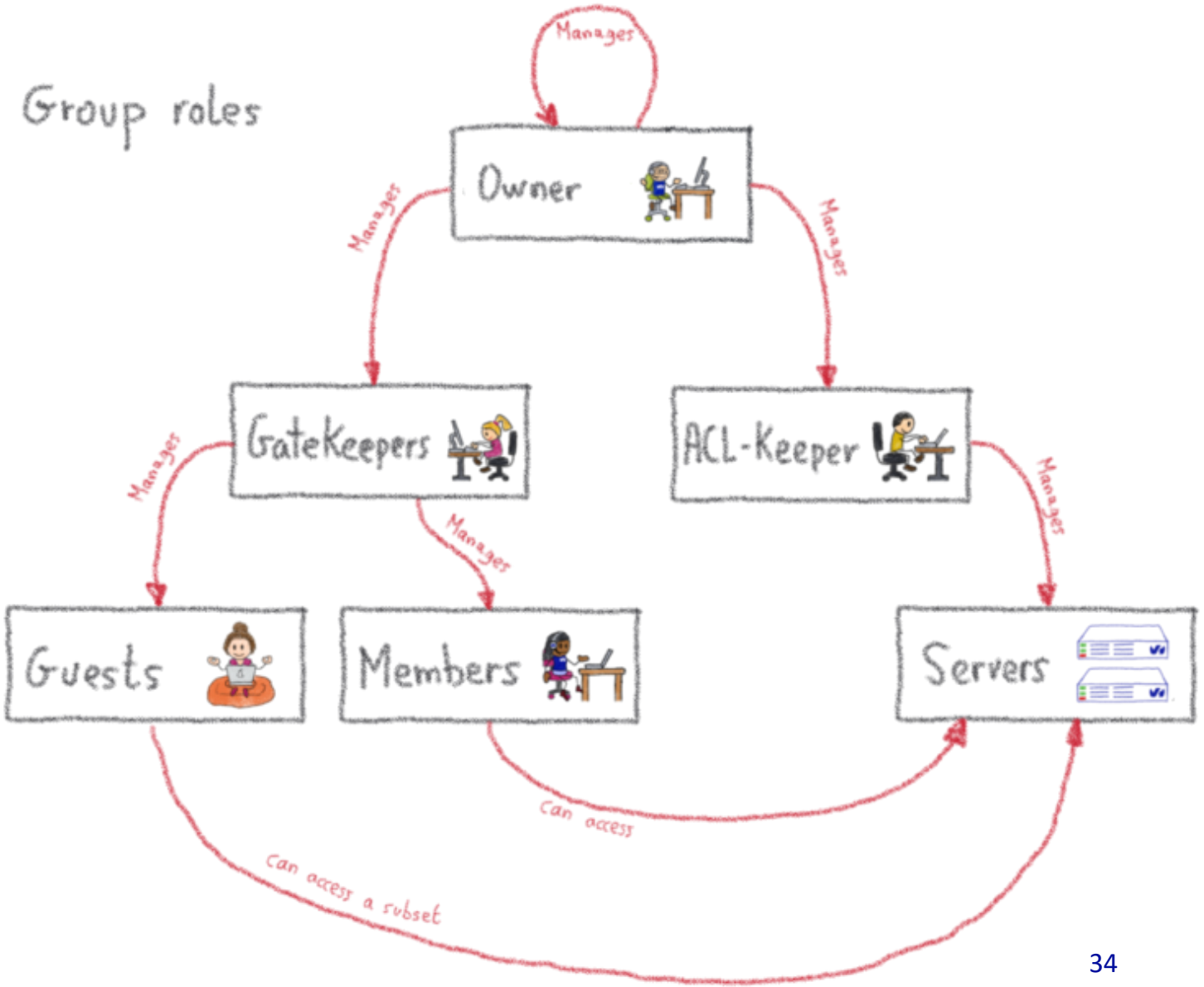
- Self-service

- Délégué à un groupe de personnes qui gèrent les accès personnels des autres

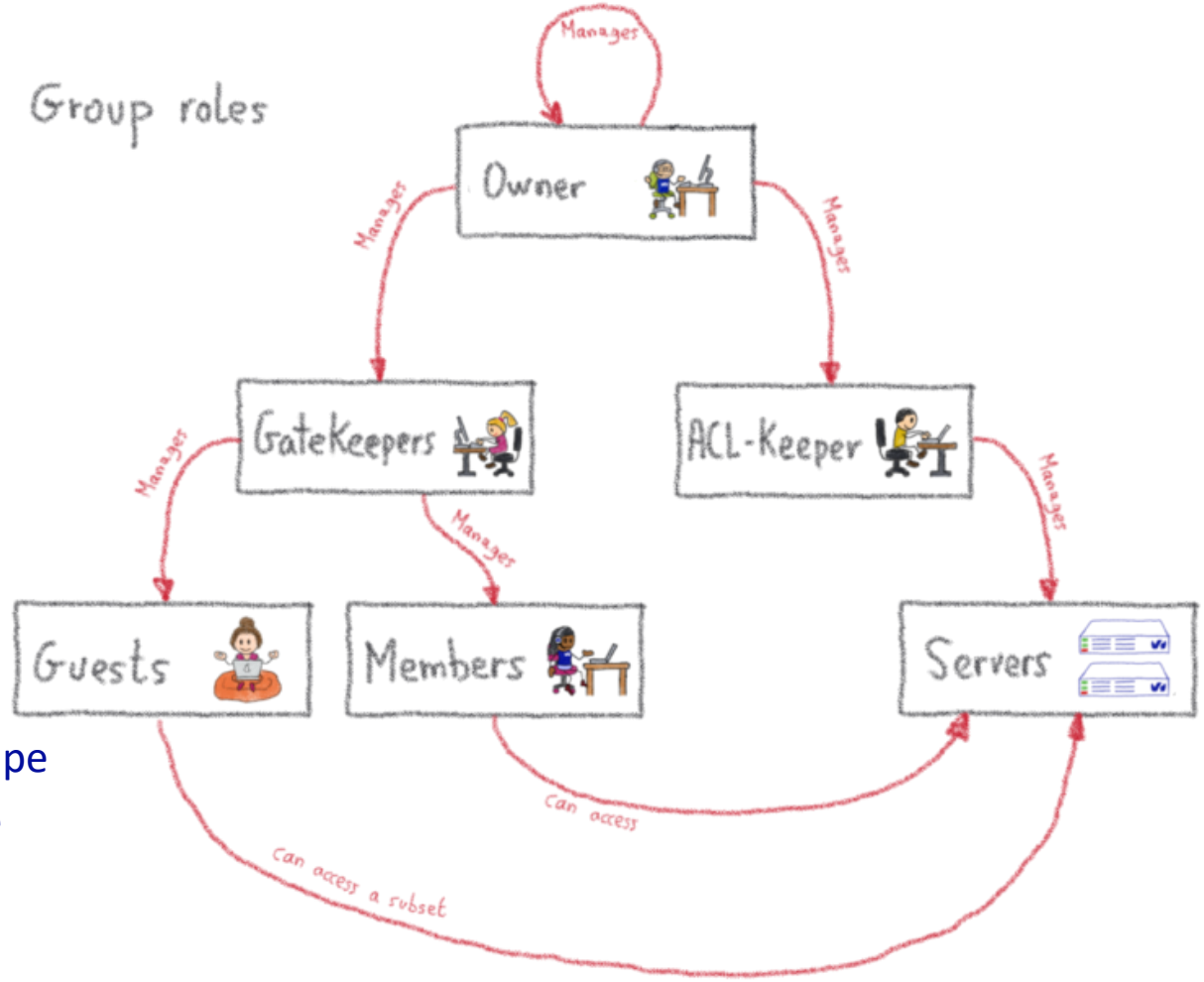
Groupes bastion : la gestion d’accès dont VOUS êtes le héros

```
Un groupe =~ []humans +
             []servers +
             []egress_creds
```

Groupes bastion : la gestion d'accès dont VOUS êtes le héros

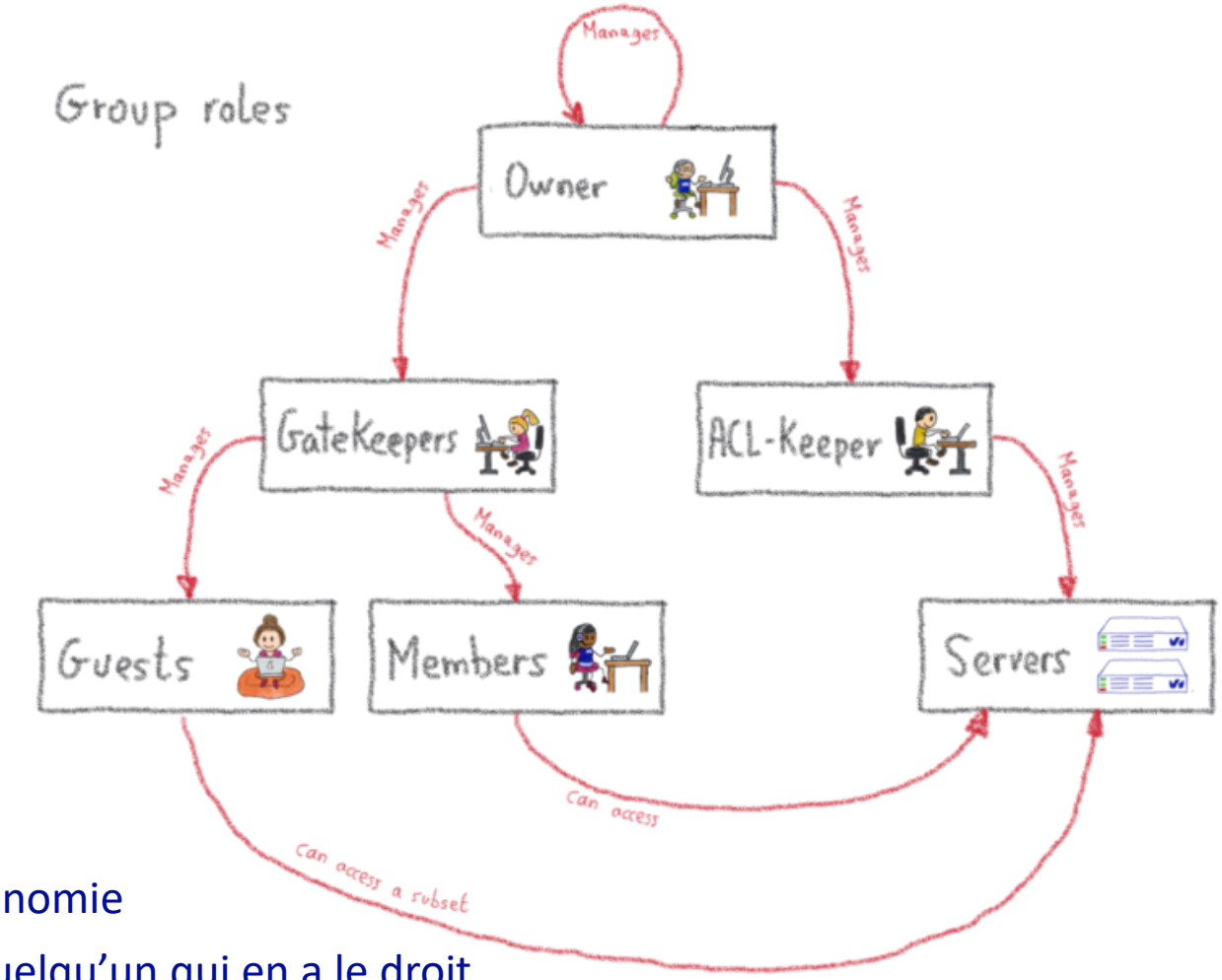


Groupes bastion : la gestion d'accès dont VOUS êtes le héros



- Members : peuvent se connecter à tous les serveurs du groupe
- Guests : peuvent se connecter à certains serveurs du groupe
- Gatekeepers : gèrent la liste des members et guests
- Aclkeepers : gèrent la liste des serveurs
- Owners : gèrent la liste des gatekeepers, aclkeepers, owners

Groupes bastion : la gestion d'accès dont VOUS êtes le héros

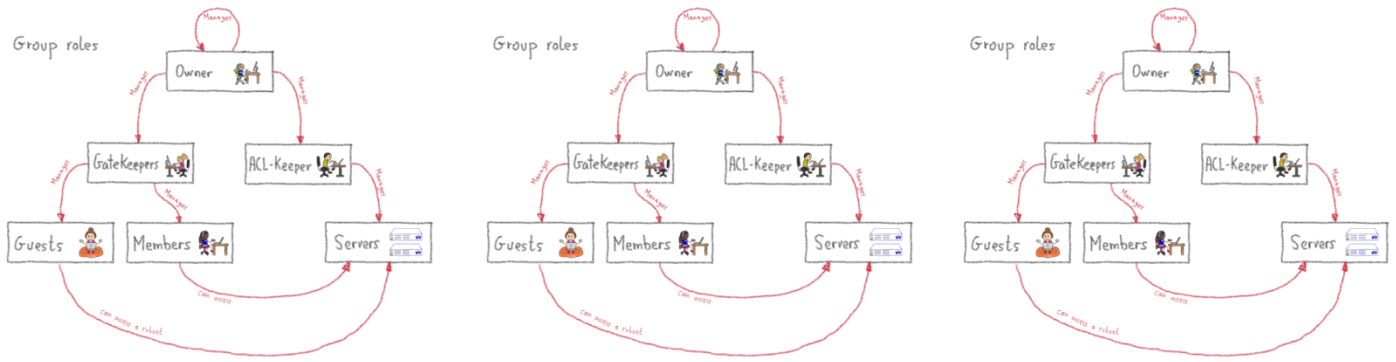


- Les rôles peuvent être cumulés
- Tout est pensé pour pouvoir laisser un maximum d'autonomie
- Seule la création du groupe nécessite de demander à quelqu'un qui en a le droit

Autres types de rôles

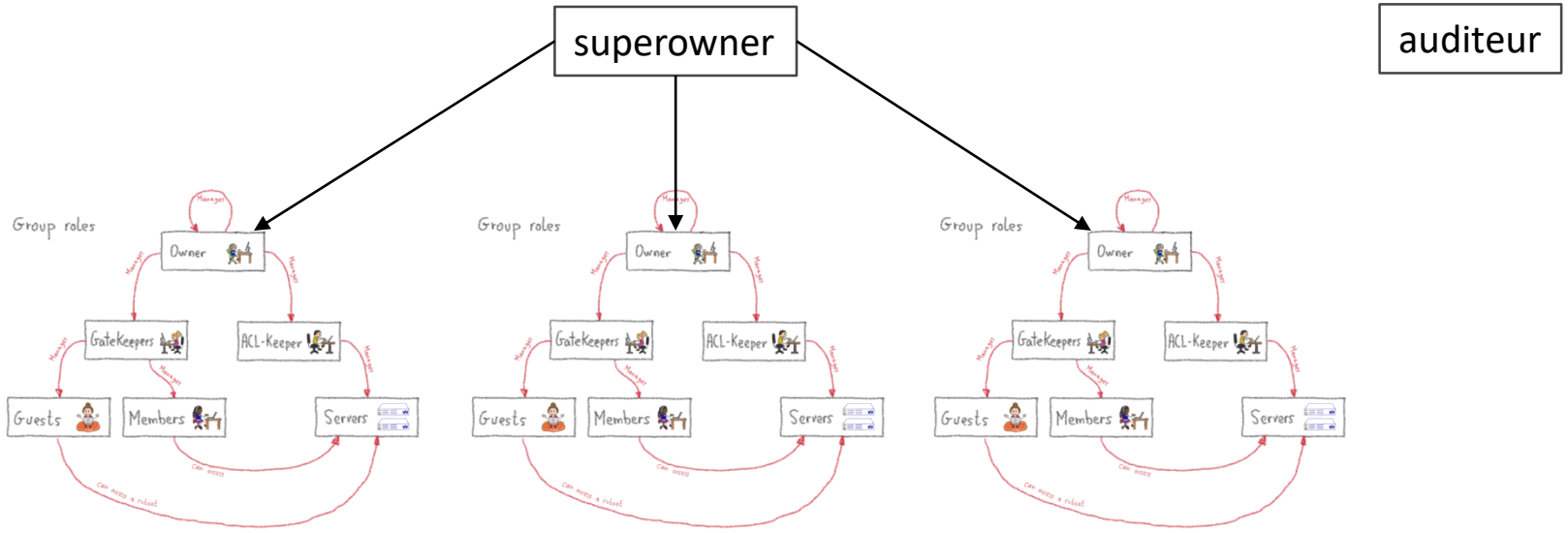
- Auditeur : peuvent voir les infos des autres comptes et des groupes dans lesquels ils ne sont pas

auditeur



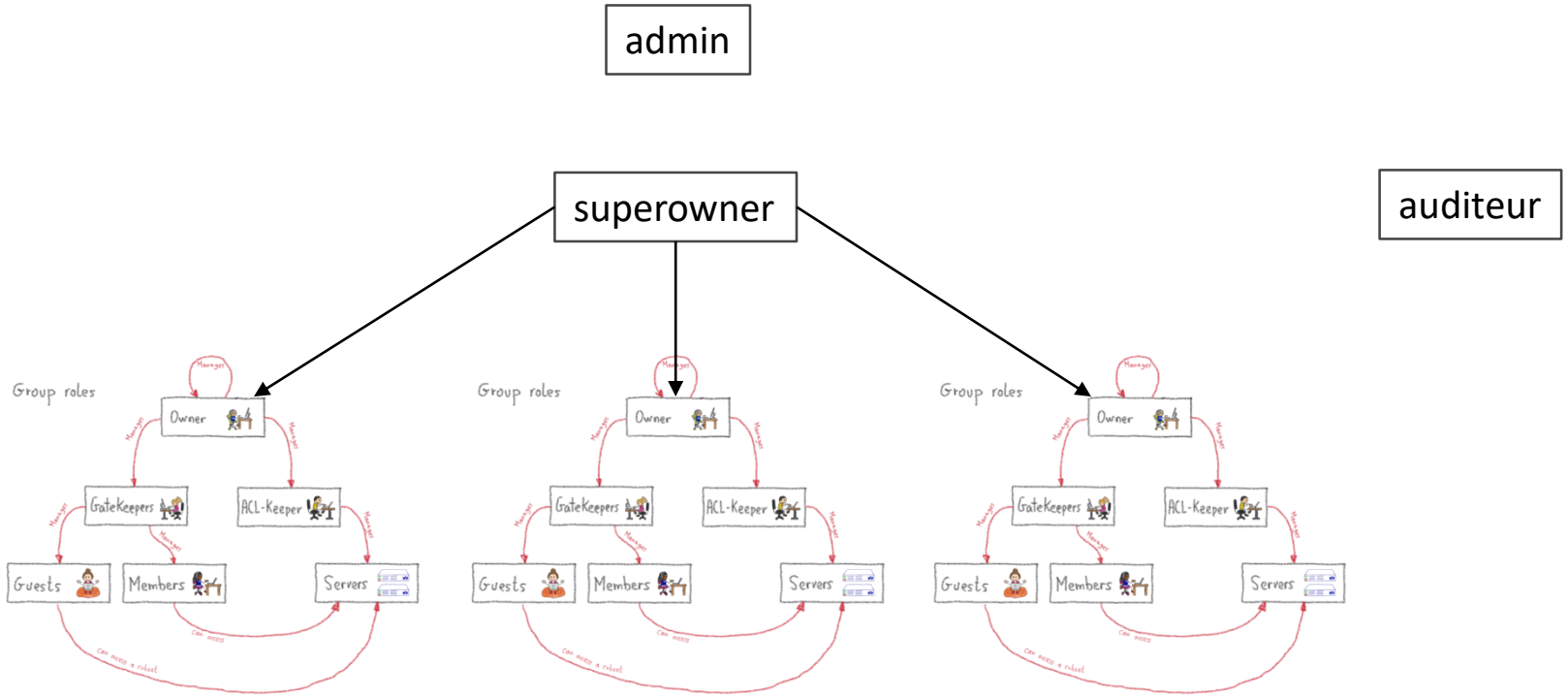
Autres types de rôles

- Auditeur : peuvent voir les infos des autres comptes et des groupes dans lesquels ils ne sont pas
- Super-Owner : implicitement owner de tous les groupes



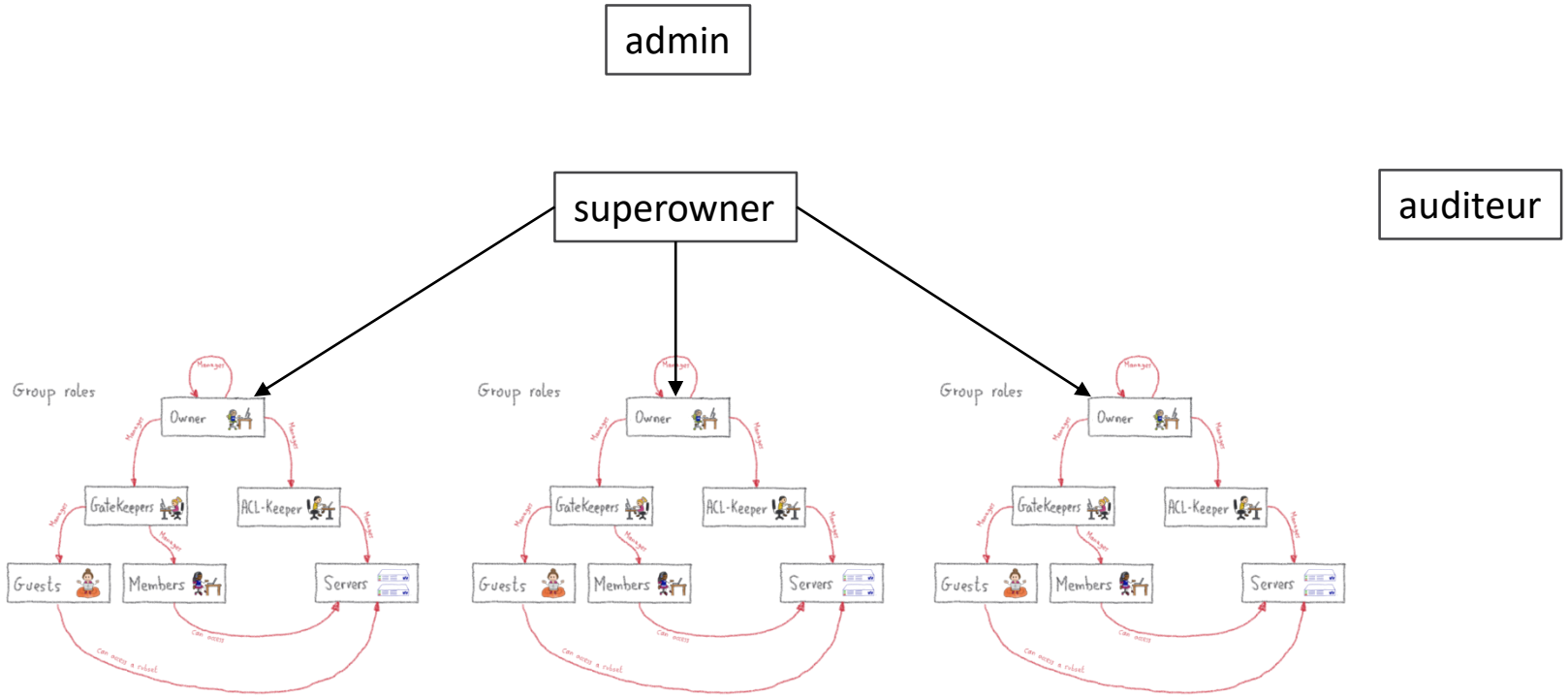
Autres types de rôles

- Auditeur : peuvent voir les infos des autres comptes et des groupes dans lesquels ils ne sont pas
- Super-Owner : implicitement owner de tous les groupes
- Admin : peut impersonate, lancer des commandes « dangereuses », implicitement Super-Owner



Autres types de rôles

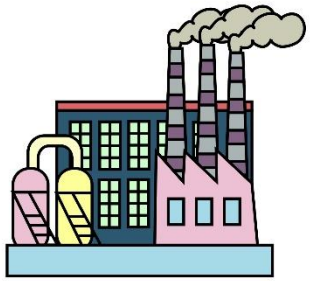
- Auditeur : peuvent voir les infos des autres comptes et des groupes dans lesquels ils ne sont pas
- Super-Owner : implicitement owner de tous les groupes
- Admin : peut impersonate, lancer des commandes « dangereuses », implicitement Super-Owner
- Commandes restreintes : une vingtaine, correspondant à autant de rôles (accountCreate, groupCreate, …)



Les royaumes

- Comment donner accès à des personnes d'une autre entité aux ressources que je contrôle?

serveur cible



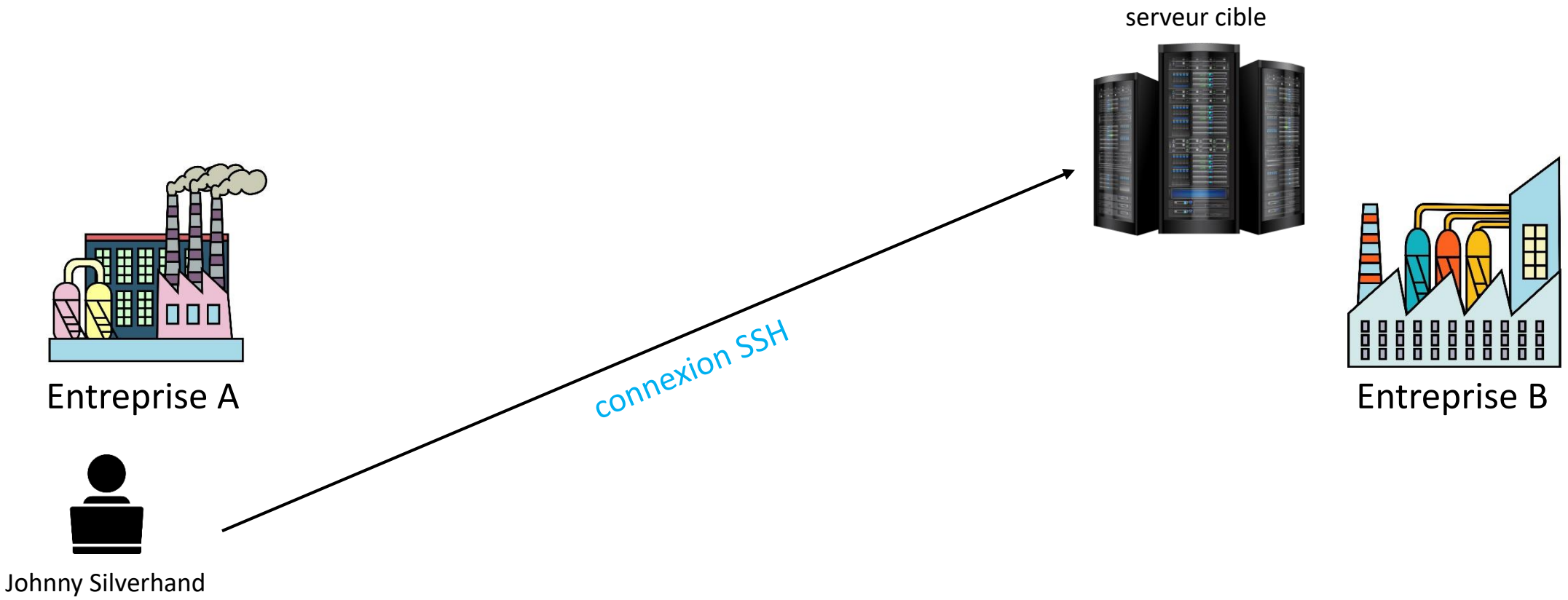
Entreprise A

Entreprise B

Johnny Silverhand

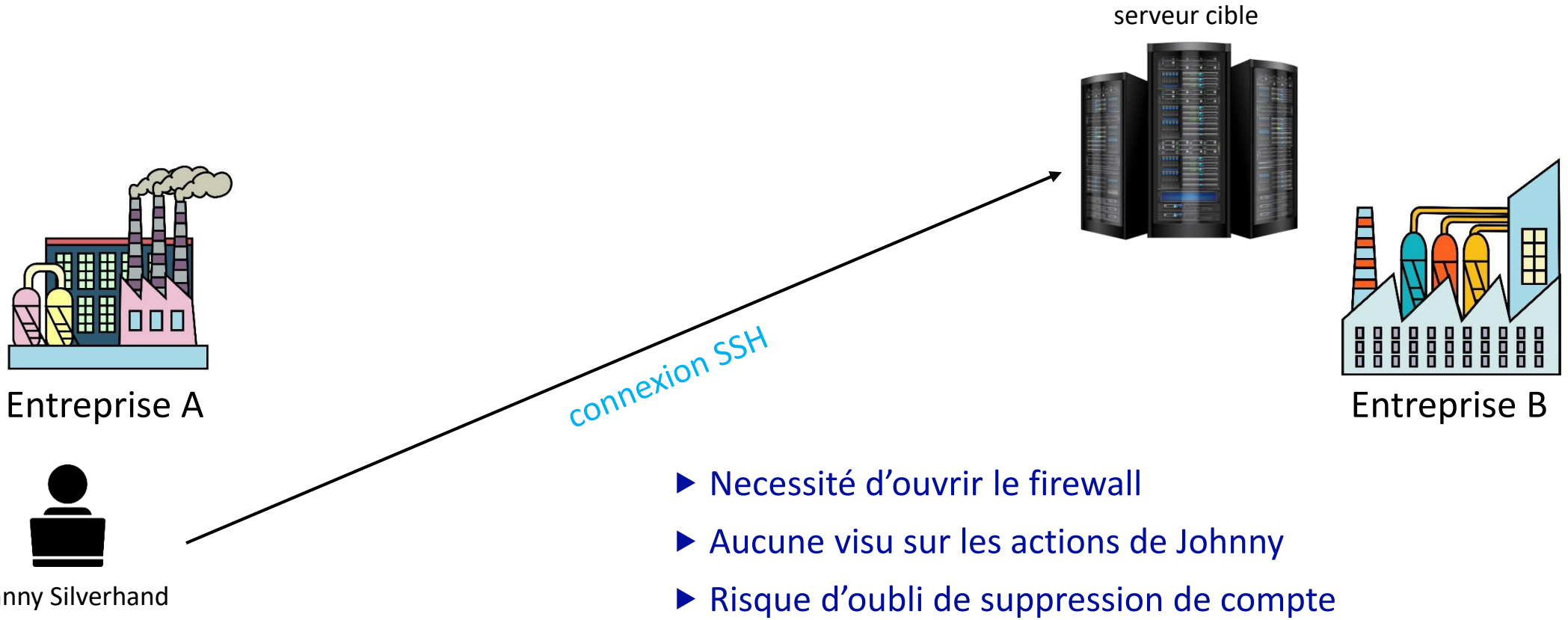
Les royaumes

- Comment donner accès à des personnes d'une autre entité aux ressources que je contrôle?
  - Idée 1 : compte temporaire directement sur le serveur cible



Les royaumes

- Comment donner accès à des personnes d'une autre entité aux ressources que je contrôle?
  - Idée 1 : compte temporaire directement sur le serveur cible

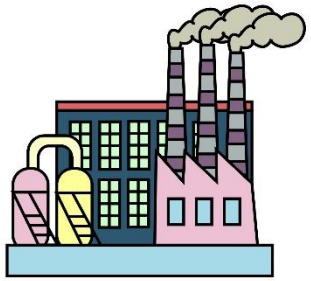


- Necessité d'ouvrir le firewall
- Aucune visu sur les actions de Johnny
- Risque d'oubli de suppression de compte

Les royaumes

- Comment donner accès à des personnes d'une autre entité aux ressources que je contrôle?

- ~~Idée 1 : compte temporaire directement sur le serveur cible~~
- Idée 2 : lui créer un compte temporaire sur mon bastion



Entreprise A

Johnny Silverhand

serveur cible

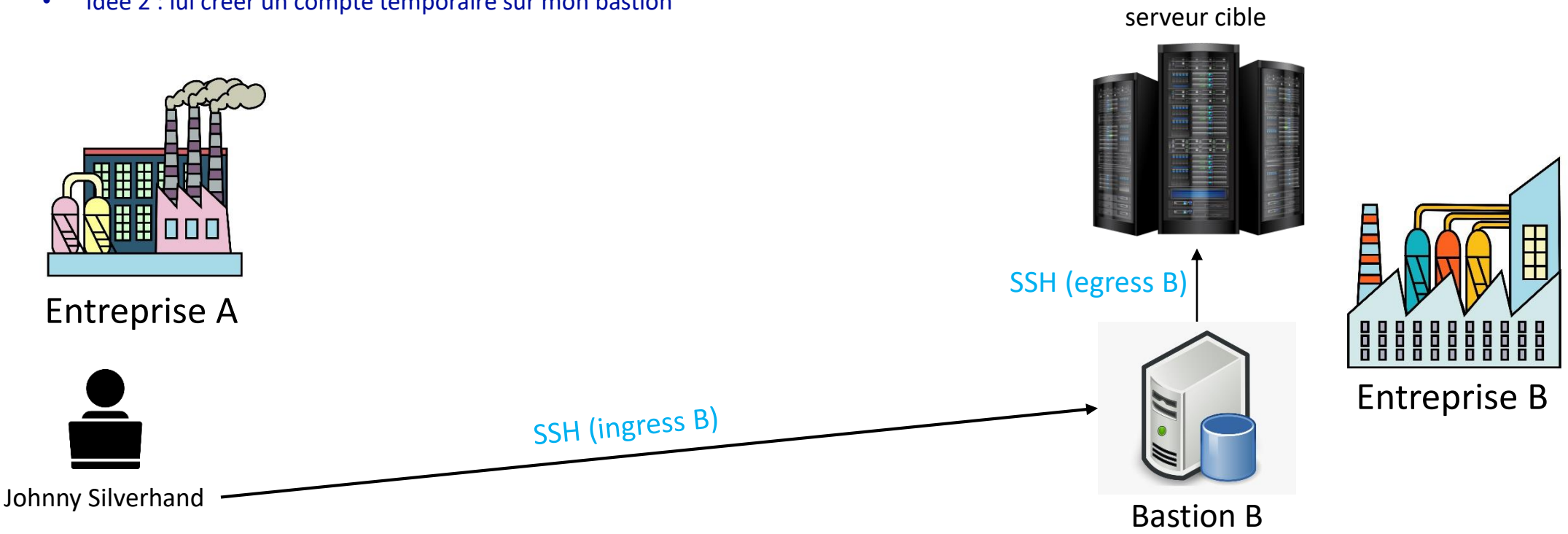
Bastion B

Entreprise B

Les royaumes

- Comment donner accès à des personnes d'une autre entité aux ressources que je contrôle?

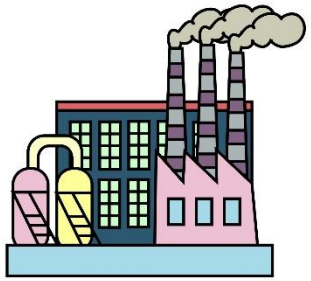
- ~~Idée 1 : compte temporaire directement sur le serveur cible~~
- Idée 2 : lui créer un compte temporaire sur mon bastion



Les royaumes

- Comment donner accès à des personnes d'une autre entité aux ressources que je contrôle?

- ~~Idée 1 : compte temporaire directement sur le serveur cible~~
- Idée 2 : lui créer un compte temporaire sur mon bastion



Entreprise A

- Nécessité d'ouvrir le firewall sur Bastion B
- ~~Aucune visu sur les actions de Johnny~~
- Risque d'oubli de suppression de compte

serveur cible

SSH (egress B)

Entreprise B

Johnny Silverhand

SSH (ingress B)

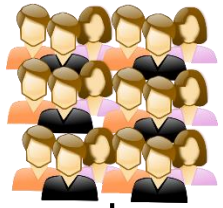
Bastion B

Les royaumes

- Comment donner accès à des personnes d'une autre entité aux ressources que je contrôle?

- ~~Idée 1 : compte temporaire directement sur le serveur cible~~
- ~~Idée 2 : lui créer un compte temporaire sur mon bastion~~
- Idée 3 : utiliser la notion de « royaume » pour créer un trust entre 2 bastions

Entreprise Acme



BDD identités

provisionne

Johnny Silverhand

Bastion Acme

serveur cible

Entreprise B

Bastion B

Les royaumes

- Comment donner accès à des personnes d'une autre entité aux ressources que je contrôle?

- ~~Idée 1 : compte temporaire directement sur le serveur cible~~
- ~~Idée 2 : lui créer un compte temporaire sur mon bastion~~
- Idée 3 : utiliser la notion de « royaume » pour créer un trust entre 2 bastions

Les royaumes

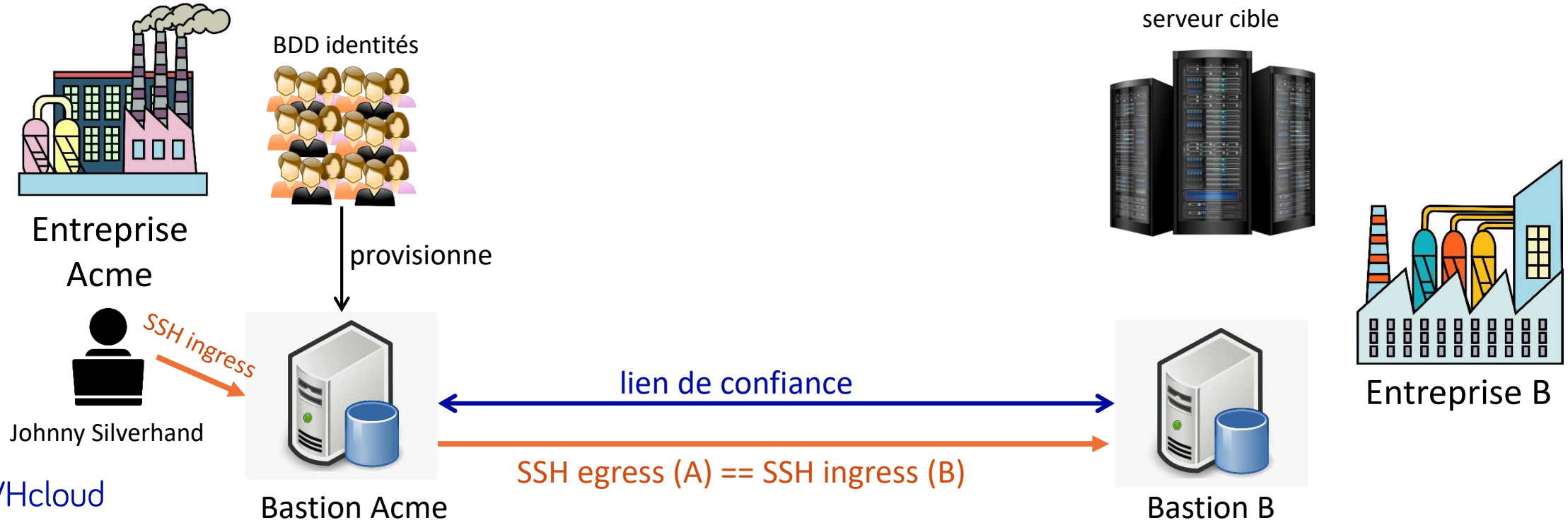
1. L'utilisateur jsilverhand s'authentifie sur le Bastion de son entreprise Acme

Les royaumes

1. L'utilisateur jsilverhand s'authentifie sur le Bastion de son entreprise Acme
2. Bastion Acme vérifie que jsilverhand est membre du groupe donnant accès au royaume qui le représente auprès du Bastion B

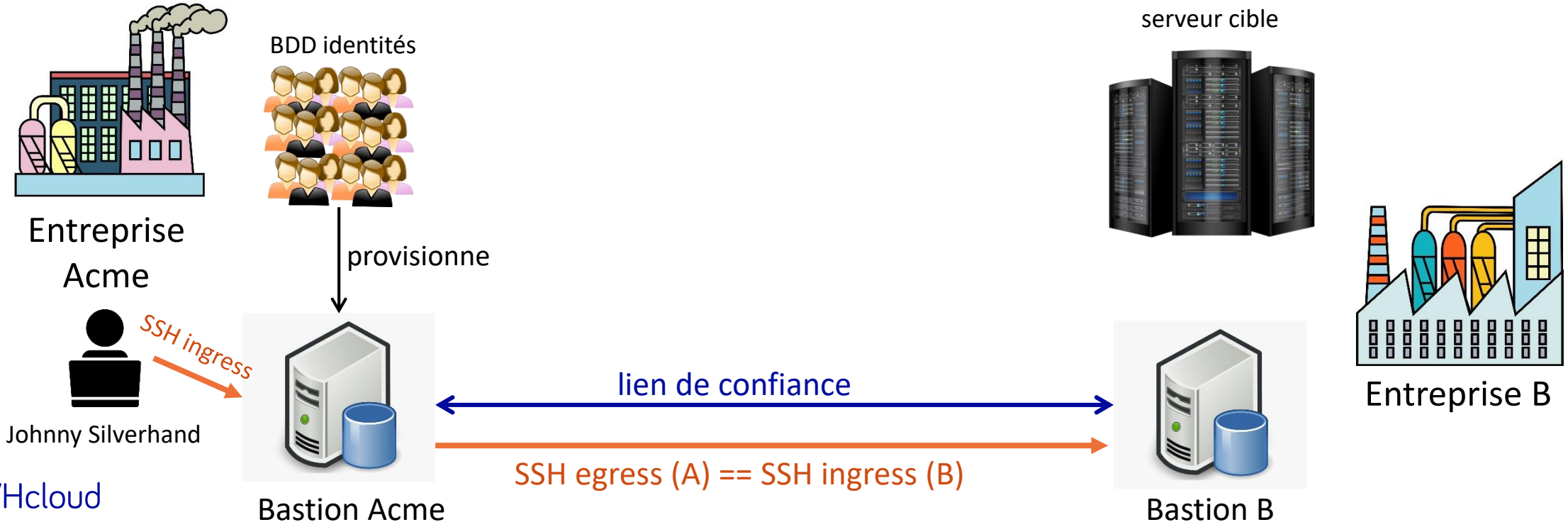
Les royaumes

1. L'utilisateur jsilverhand s'authentifie sur le Bastion de son entreprise Acme
2. Bastion Acme vérifie que jsilverhand est membre du groupe donnant accès au royaume qui le représente auprès du Bastion B
3. Bastion Acme s'identifie via la clé de groupe du royaume auprès du Bastion B



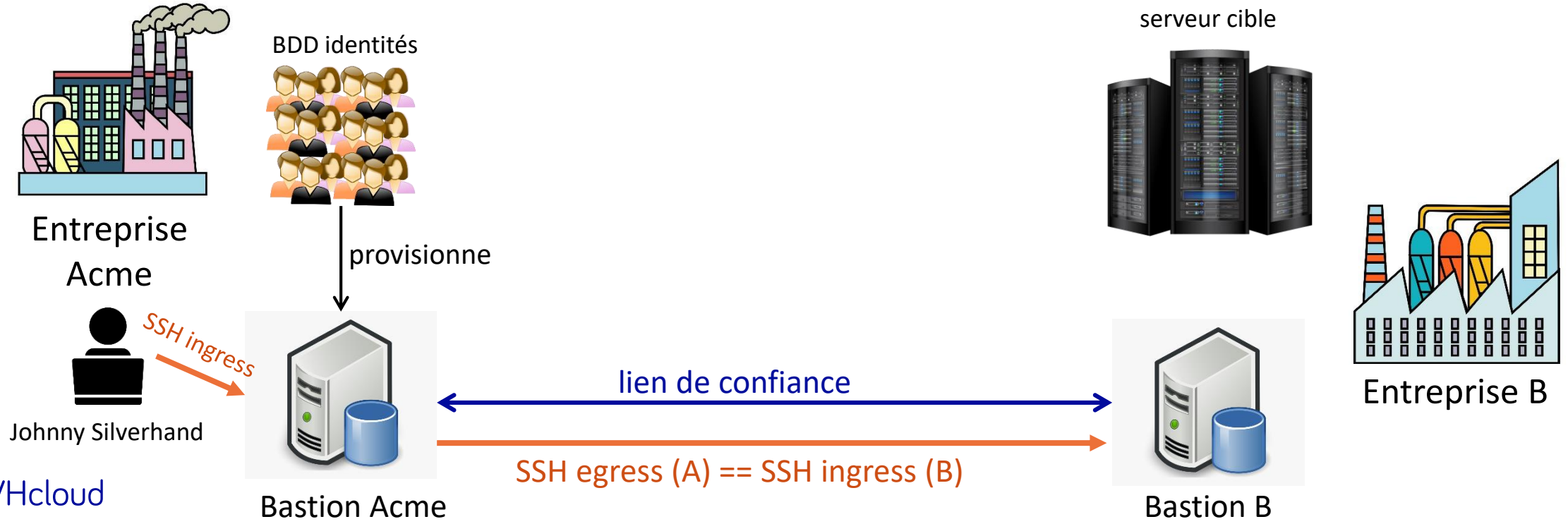
Les royaumes

1. L'utilisateur jsilverhand s'authentifie sur le Bastion de son entreprise Acme
2. Bastion Acme vérifie que jsilverhand est membre du groupe donnant accès au royaume qui le représente auprès du Bastion B
3. Bastion Acme s'identifie via la clé de groupe du royaume auprès du Bastion B
4. Bastion Acme certifie qu'il se connecte sur B pour le compte d'un certain jsilverhand qui fait partie de son royaume acme



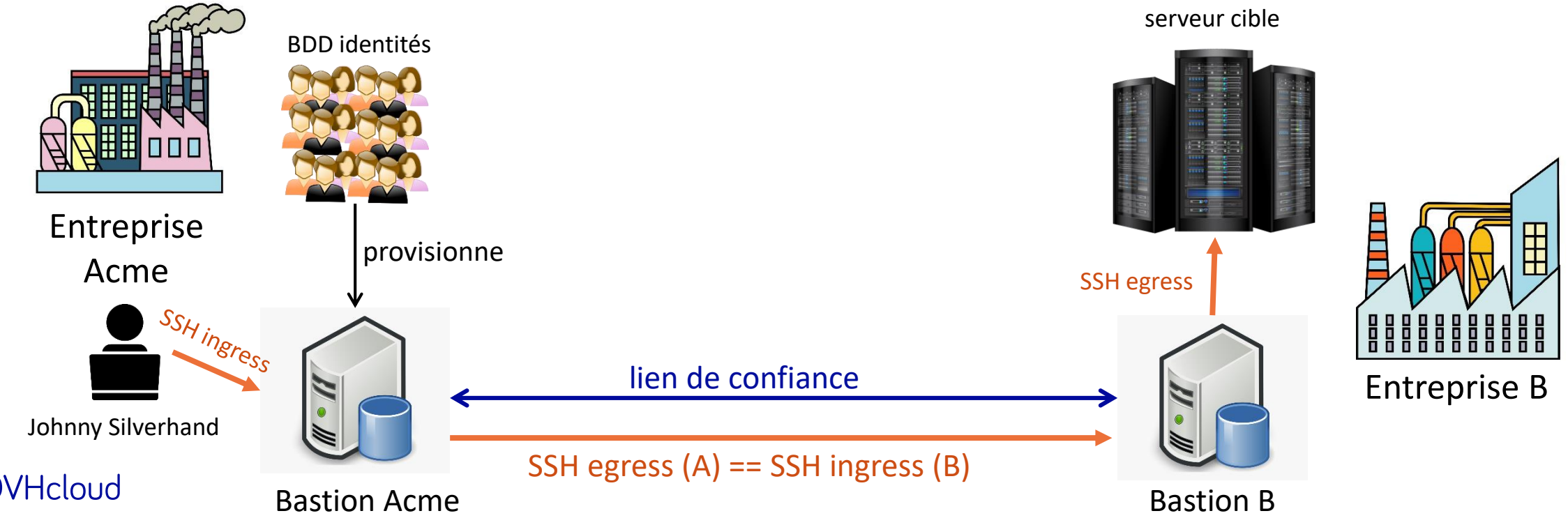
Les royaumes

1. L'utilisateur jsilverhand s'authentifie sur le Bastion de son entreprise Acme
2. Bastion Acme vérifie que jsilverhand est membre du groupe donnant accès au royaume qui le représente auprès du Bastion B
3. Bastion Acme s'identifie via la clé de groupe du royaume auprès du Bastion B
4. Bastion Acme certifie qu'il se connecte sur B pour le compte d'un certain jsilverhand qui fait partie de son royaume acme
5. Bastion B vérifie si acme/jsilverhand a accès au serveur cible demandé



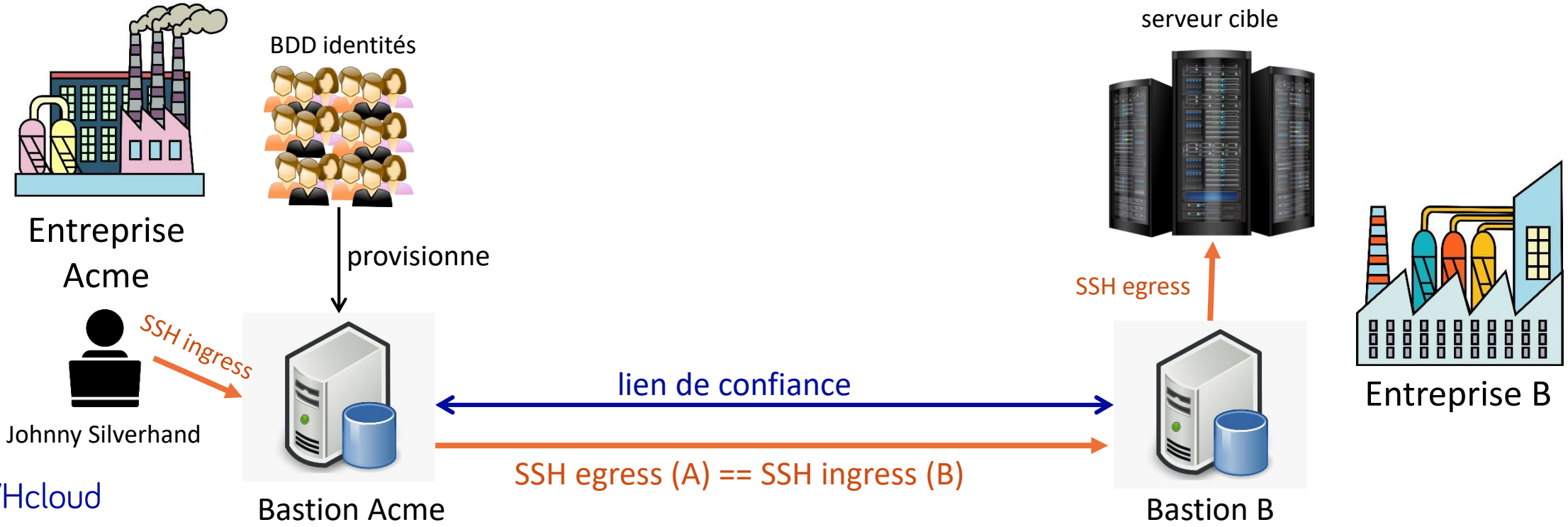
Les royaumes

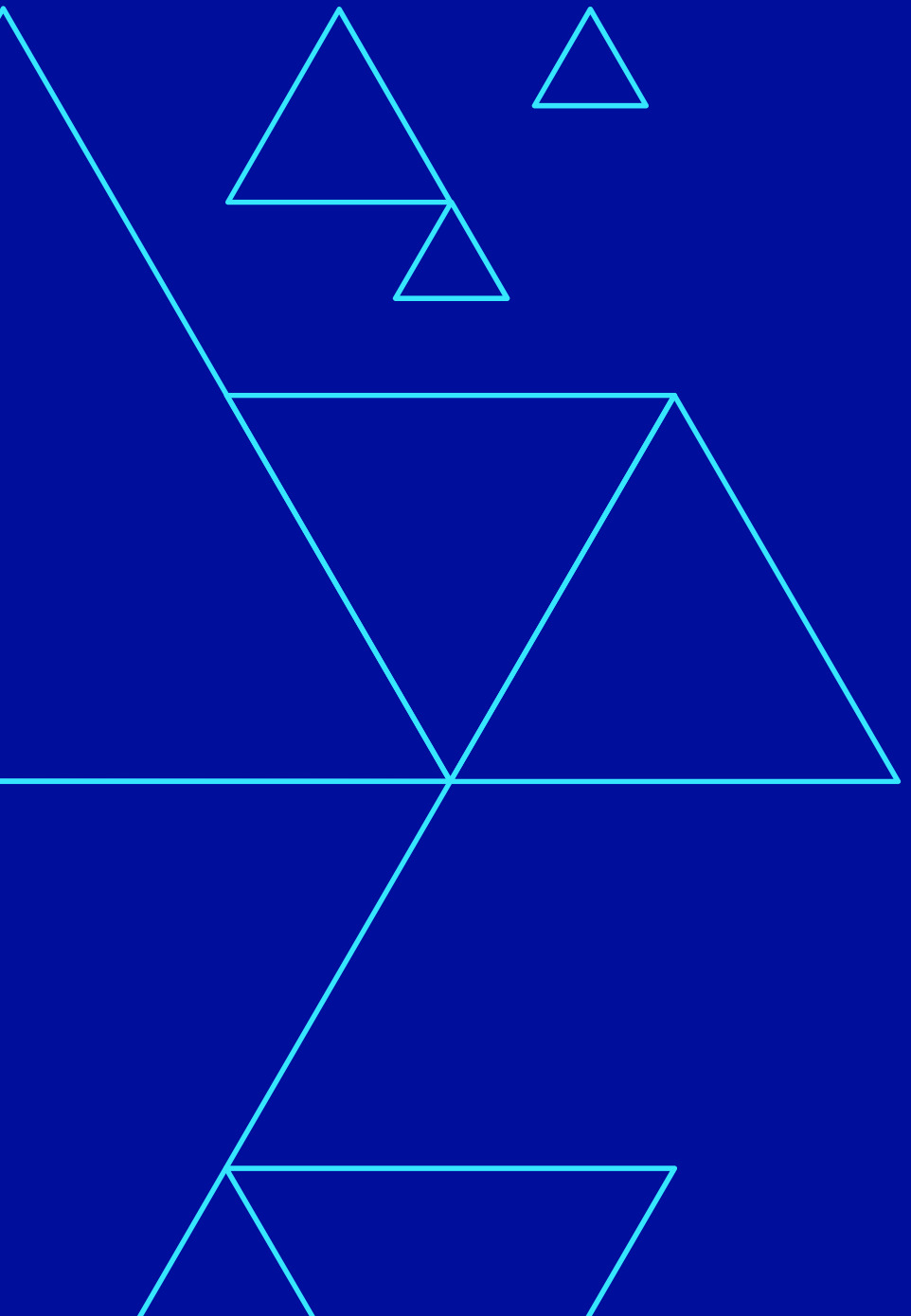
1. L'utilisateur jsilverhand s'authentifie sur le Bastion de son entreprise Acme
2. Bastion Acme vérifie que jsilverhand est membre du groupe donnant accès au royaume qui le représente auprès du Bastion B
3. Bastion Acme s'identifie via la clé de groupe du royaume auprès du Bastion B
4. Bastion Acme certifie qu'il se connecte sur B pour le compte d'un certain jsilverhand qui fait partie de son royaume acme
5. Bastion B vérifie si acme/jsilverhand a accès au serveur cible demandé
6. Bastion B initie sa connexion egress vers le serveur cible pour le compte de acme/jsilverhand



Les royaumes

1. L'utilisateur jsilverhand s'authentifie sur le Bastion de son entreprise Acme
2. Bastion Acme vérifie que jsilverhand est membre du groupe donnant accès au royaume qui le représente auprès du Bastion B
3. Bastion Acme s'identifie via la clé de groupe du royaume auprès du Bastion B
4. Bastion Acme certifie qu'il se connecte sur B pour le compte d'un certain jsilverhand qui fait partie de son royaume acme
5. Bastion B vérifie si acme/jsilverhand a accès au serveur cible demandé
6. Bastion B initie sa connexion egress vers le serveur cible pour le compte de acme/jsilverhand
7. L'utilisateur jsilverhand est connecté au serveur cible via 2 bastions (chacun enregistre ses actions), et 3 connexions SSH





OVHcloud

Ouvrons le capot

(et salissons nous les mains)

(un peu)

Utilisateurs & groupes au cœur du système

- Système d’exploitation
  - Pas juste un ordonnanceur !
  - Garantit une étanchéité inter-utilisateurs

Utilisateurs & groupes au cœur du système

- Système d'exploitation
  - Pas juste un ordonnanceur !
  - Garantit une étanchéité inter-utilisateurs
- Choix de conception
  - Compte applicatif utilisateur bastion = compte sur l'OS sous-jacent (UID)
  - Rôle d'un groupe de bastion = groupe système sur l'OS sous-jacent (GID)
  - Clé privée de groupe ou de compte = fichier sur le FS avec les permissions idoines

Utilisateurs & groupes au cœur du système

- Système d'exploitation
  - Pas juste un ordonnanceur !
  - Garantit une étanchéité inter-utilisateurs
- Choix de conception
  - Compte applicatif utilisateur bastion = compte sur l'OS sous-jacent (UID)
  - Rôle d'un groupe de bastion = groupe système sur l'OS sous-jacent (GID)
  - Clé privée de groupe ou de compte = fichier sur le FS avec les permissions idoines
- Vérification des habilitations
  - Faite au niveau algorithmique (dans le code)
  - Faite par l'OS par-dessus (UID, GID, permissions de fichiers)

Utilisateurs & groupes au cœur du système

- Système d'exploitation
  - Pas juste un ordonnanceur !
  - Garantit une étanchéité inter-utilisateurs
- Choix de conception
  - Compte applicatif utilisateur bastion = compte sur l'OS sous-jacent (UID)
  - Rôle d'un groupe de bastion = groupe système sur l'OS sous-jacent (GID)
  - Clé privée de groupe ou de compte = fichier sur le FS avec les permissions idoines
- Vérification des habilitations
  - Faite au niveau algorithmique (dans le code)
  - Faite par l'OS par-dessus (UID, GID, permissions de fichiers)
- Pas de shell « /bin/sh », le bastion ***est*** le shell
  - Tout le code est exécuté sous les (non-)droits de l'utilisateur
  - Pas de mouvement latéral facile si RCE

Exemple d’accès personnel en « self-service »

```
speed $ type bastion
bastion is aliased to `ssh -i ~/.ssh/id_demo stephane@127.0.0.1 -tp 49155 -- '
```

```
speed $ bastion moi@palmer.example.org
*------------------------------------------------------------------------------*
|THIS IS A PRIVATE COMPUTER SYSTEM, UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED.|
|ALL CONNECTIONS ARE LOGGED. IF YOU ARE NOT AUTHORIZED, DISCONNECT NOW.        |
*------------------------------------------------------------------------------*
Enter passphrase for key '/home/speed/.ssh/id_demo':
~ Welcome to demo, stephane, your last login was 00:00:37 ago (Mon 2021-05-10 17:00:18 UTC) from 172.17.0.1(172.17.0.1)

172.17.0.1:35038 => stephane@7d7e42256577:22 => moi@palmer.example.org:22 ...
~ Access denied for stephane to moi@127.1.3.37:22
Connection to 127.0.0.1 closed.
speed $
```

Exemple d’accès personnel en « self-service »

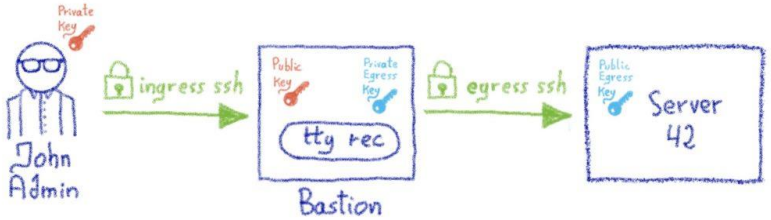
```
speed $ bastion -i
*--------------------------------------------------------------------------*
|THIS IS A PRIVATE COMPUTER SYSTEM, UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED.|
|ALL CONNECTIONS ARE LOGGED. IF YOU ARE NOT AUTHORIZED, DISCONNECT NOW.      |
*--------------------------------------------------------------------------*
Enter passphrase for key '/home/speed/.ssh/id_demo':

Welcome to demo interactive mode, type `help' for available commands.
You can use <tab> and <tab><tab> for autocompletion.
You'll be disconnected after 60 seconds of inactivity.
Loading... 81 commands and 349 autocompletion rules loaded.

stephane@demo(master)> selfAddPersonalAccess --host palmer.example.org --user moi --port 22
---7d7e42256577-----------------------------------------the-bastion-3.03.01---
=> adding personal access to a server on your account
--------------------------------------------------------------------------------
~ Testing connection to moi@127.1.3.37, please wait...
Warning: Permanently added '127.1.3.37' (ED25519) to the list of known hosts.
*--------------------------------------------------------------------------*
|THIS IS A PRIVATE COMPUTER SYSTEM, UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED.|
|ALL CONNECTIONS ARE LOGGED. IF YOU ARE NOT AUTHORIZED, DISCONNECT NOW.      |
*--------------------------------------------------------------------------*
moi@127.1.3.37: Permission denied (publickey).
~ Note: if you still want to add this access even if it doesn't work, use --force
~ Couldn't connect to moi@127.1.3.37 (ssh returned error 255). Hint: did you add the proper public key to the remote's authorized_keys?
-----------------------------------------------------</selfAddPersonalAccess>---
stephane@demo(master)>
```

Exemple d’accès personnel en « self-service »

```
stephane@demo(master)> selfListEgressKeys
---7d7e42256577-----------------------------------------the-bastion-3.03.01---
=> your account's public ingress keys
-------------------------------------------------------------------------------
~ You can copy one of those keys to a remote machine to get access to it through your account
~ on this bastion, if it is listed in your private access list (check selfListAccesses)
~
~ Always include the from="172.17.0.8" part when copying the key to a server!
~
~ fingerprint: SHA256:kcnatiQPYU9J+++4+LUrXJ0QN4HAJHXwEigIDpJ66Ks (RSA-4096) [ID = id7fc7994f]
~ keyline follows, please copy the *whole* line:
from="172.17.0.8" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCzlRDnOCqDf85Afnei42Wa3DhbH7sPxnn3XQWroanxbftdQjaAxbtJlnKqdf+VmPjgAW7TNFVnVTZFjpePp
sJZ3hEwav/Iih2TAYfqzGaKywDrrqfcFJ705J+b3KOCOFB/AIAZAlFgAjl4roGV4WsuTd0Eu3F3ZT+a0CXvN9FgpjkdDZotOw4Z8gXtC+8QD5OWQMAPGb8r37i6W+fGv2yVLyHdGGBR
FrIlPIVVYdd4Qzn5DR0FCFtPr8RsZ6fmFStvwoScxuU754bKzUMAi+9fSlmgqFu6NuPP7/BJeslTWp/SX2t+/t7vzgHC1/0sP6/nFQIBYFMB47rrmo7l3uD9vW9+66kIKH1Yz7RFTIe
k4/jV1UAq8xZYYNexoQsuvmOc0kl/9UCDRX5hecogWa49xMDJQwEBe2IFw4X1Zaw8m/iUmgTkiAScTwAO4MLU29KFSnhjCtTuzEyZSvZN4XtevMqaK4EBGk9tESycqOL2abzefcnCfu
7tr8PZqwV0rykxBIe8V3dyeKOz4iEFIB2X9IRzLEIGcHhECByZbxDT7VBPWT5UF12RC11TaLEDJVD48UfBSejogGMJuqmraeR7c3PwU1GGgmwg7Ue21QM9lk++lKUfbsDbyS6NPE8GU
DV2dHwKb0jMIUqDkF8xWsigCeKQl7KvLRFdD1zd52aKUQ== stephane@demo:1620665446
~
-----------------------------------------------------------</selfListEgressKeys>---
stephane@demo(master)>
stephane@demo(master)> selfListIngressKeys
---7d7e42256577-----------------------------------------the-bastion-3.03.01---
=> Here are the public keys that allow you to connect to the bastion
-------------------------------------------------------------------------------
~ info: ADDED_BY=stephane USING=selfAddIngressKey UNIQID=1dfc420cb530 TIMESTAMP=1620665996 DATETIME=2021-05-10T16:59:56 VERSION=3.03.01
~ fingerprint: SHA256:z7l/0xSRjACx6YPUx1JE7SnjZzoq2S8YDMoCh+c2jyU (ED25519-256) [ID = 2]
~ keyline follows, please copy the *whole* line:
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEN9pScg5Hx3YLdfgGRsasLxnc4o3hlLo6K8hP/6vu9r speed@nas
~
-----------------------------------------------------------</selfListIngressKeys>---
stephane@demo(master)>
```



Exemple d'accès personnel en « self-service »

Pendant ce temps, à Vera Cruz…

```
moi@palmer $ echo 'from="172.17.0.8" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCzlRDnOCqDf85Afnei42Wa3DhbH7sPxnn3XQWroanxbftdQjaAxbtJlnKqdf+VmP
jgAW7TNFVnVTZFjpePpsJZ3hEwav/Iih2TAYfqzGaKywDrrqfcFJ705J+b3KOCOFB/AIAZAlFgAjl4roGV4WsuTd0Eu3F3ZT+a0CXvN9FgpjkdDZotOw4Z8gXtC+8QD5OWQMAPGb8r3
7i6W+fGv2yVLyHdGGBRFrIlPIVVYdd4QzN5DR0FCFtPr8RsZ6fmFStvwoScxuU754bKzUMAi+9fSlmgqFu6NuPP7/BJeslTWp/SX2t+/t7vzgHC1/0sP6/nFQIBYFMB47rrmo7l3uD9
vW9+66kIKH1Yz7RFTIek4/jV1UAq8xZYYNexoQsuvmOc0kl/9UCDRX5hecogWa49xMDJQwEBe2IFw4X1Zaw8m/iUmgTkiAScTwAO4MLU29KFSnhjCtTuzEyZSvZN4XtevMqaK4EBGk9
tESycqOL2abzefcnCfu7tr8PZqwV0rykxBIe8V3dyeKOz4iEFIB2X9IRzLEIGcHhECByZbxDT7VBPWT5UF12RC11TaLEDJVD48UfBSejogGMJuqmraeR7c3PwU1GGgmwg7Ue21QM9lk
++lKUfbsDbyS6NPE8GUDV2dHwKb0jMIUqDkF8xWsigCeKQl7KvLRFdD1zd52aKUQ== stephane@demo:1620665446' > .ssh/authorized_keys
moi@palmer $
```

De retour sur le bastion :

```
stephane@demo(master)> selfAddPersonalAccess --host palmer.example.org --user moi --port 22
---7d7e42256577-------------------------------------------the-bastion-3.03.01---
=> adding personal access to a server on your account
--------------------------------------------------------------------------------
~ Testing connection to moi@127.1.3.37, please wait...
Warning: Permanently added '127.1.3.37' (ED25519) to the list of known hosts.
*------------------------------------------------------------------------------*
|THIS IS A PRIVATE COMPUTER SYSTEM, UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED.|
|ALL CONNECTIONS ARE LOGGED. IF YOU ARE NOT AUTHORIZED, DISCONNECT NOW.        |
*------------------------------------------------------------------------------*

~ Access to moi@127.1.3.37:22 was added to account stephane
-----------------------------------------------------</selfAddPersonalAccess>---
```

Exemple d’accès personnel en « self-service »

```
speed $ bastion moi@palmer.example.org
*--------------------------------------------------------------------------*
|THIS IS A PRIVATE COMPUTER SYSTEM, UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED.|
|ALL CONNECTIONS ARE LOGGED. IF YOU ARE NOT AUTHORIZED, DISCONNECT NOW.      |
*--------------------------------------------------------------------------*
Enter passphrase for key '/home/speed/.ssh/id_demo':
~ Welcome to demo, stephane, your last login was 00:01:59 ago (Mon 2021-05-10 17:25:09 UTC) from 172.17.0.1(172.17.0.1)

172.17.0.1:40624 => stephane@7d7e42256577:22 => moi@palmer.example.org:22 ...
 allowed ... log on(/home/stephane/ttyrec/127.1.3.37/2021-05-10.17-27-08.414525.f2404ca5647c.stephane.moi.127.1.3.37.22.ttyrec)

 will try the following accesses you have:
  - personal access with RSA-4096 key SHA256:kcnatiQPYU9J+++4+LUrXJ0QN4HAJHXwEigIDpJ66Ks [2021/05/10]

Connecting...
The authenticity of host '127.1.3.37 (127.1.3.37)' can't be established.
ED25519 key fingerprint is SHA256:38lcYx+aW8xMY8WDnFLcKsUIksjX6xvinGDdF3L+5lM.
No matching host key fingerprint found in DNS.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '127.1.3.37' (ED25519) to the list of known hosts.
*--------------------------------------------------------------------------*
|THIS IS A PRIVATE COMPUTER SYSTEM, UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED.|
|ALL CONNECTIONS ARE LOGGED. IF YOU ARE NOT AUTHORIZED, DISCONNECT NOW.      |
*--------------------------------------------------------------------------*

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon May 10 17:27:11 2021 from 127.0.0.1
moi@palmer $
```

Exemple d’accès personnel en « self-service »

```
root@7d7e42256577:/# ps faxu
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         919  0.0  0.0   3992  3272 pts/1    Ss   17:23   0:00 bash
root        1045  0.0  0.0   7644  2672 pts/1    R+   17:30   0:00  \_ ps faxu
root           1  0.0  0.0   2392    68 ?        Ss   16:50   0:00 /bin/sh -c /opt/bastion/docker/entrypoint.sh --sandbox
root           7  0.0  0.0   3740   368 ?        S    16:50   0:00 bash /opt/bastion/docker/entrypoint.sh --sandbox
root          29  0.0  0.0   2300    64 ?        S    16:50   0:00  \_ sleep 3600
root          15  0.0  0.0  15856  2948 ?        Ss   16:50   0:00 /usr/sbin/sshd
root         983  0.0  0.0  19020  7940 ?        Ss   17:27   0:00  \_ sshd: stephane [priv]
stephane     986  0.0  0.0  19020  4744 ?        S    17:27   0:00  |   \_ sshd: stephane@pts/0
stephane     987  0.0  0.0   9260  6164 pts/0    Ss+  17:27   0:00  |       \_ /opt/bastion/bin/shell/connect.pl ttyrec -f /home/stephane/ttyrec/127.1.3.37/2021-05-10.17-27-08.414525.f2404ca5647c.
stephane     989  0.0  0.0   3088  1960 pts/0    S+   17:27   0:00  |           \_ /usr/bin/ttyrec -f /home/stephane/ttyrec/127.1.3.37/2021-05-10.17-27-08.414525.f2404ca5647c.stephane.moi.127.1.3.
stephane     990  0.0  0.0   3088   140 pts/0    S+   17:27   0:00  |               \_ /usr/bin/ttyrec -f /home/stephane/ttyrec/127.1.3.37/2021-05-10.17-27-08.414525.f2404ca5647c.stephane.moi.127.
stephane     991  0.0  0.0  12828  5600 pts/2    Ss+  17:27   0:00  |                   \_ /usr/bin/ssh 127.1.3.37 -l moi -p 22 -i /home/stephane/.ssh/id_rsa4096_private.1620665446 -o PreferredAut
```

Et si on cassait le code ?

```
stephane@demo(master)> groupInfo --group cyberops
---7d7e42256577-----------------------------------------the-bastion-3.03.01---
=> group info
------------------------------------------------------------------------------
~ Group cyberops's Owners are: jsilverhand
~ Group cyberops's GateKeepers (managing the members/guests list) are: jsilverhand
~ Group cyberops's ACLKeepers (managing the group servers list) are: jsilverhand
~ Group cyberops's Members (with access to ALL the group servers) are: jsilverhand
~ Group cyberops's Guests (with access to SOME of the group servers) are: -
~
~ The public key of this group is:
~
~ fingerprint: SHA256:gcQYCVSmmBW6GJqTigNlP773e5D6CQbPavaaMvYigD8 (ED25519-256) [ID = id39fa3998]
~ keyline follows, please copy the *whole* line:
from="172.17.0.8" ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKIbAVHioKWqovebPleyXCLU9wvdCRhGZW8hoFmdDR5B cyberops@demo:1620668321
~
------------------------------------------------------------------</groupInfo>---
stephane@demo(master)> groupListServers --group cyberops
---7d7e42256577-----------------------------------------the-bastion-3.03.01---
=> list of servers pertaining to the group
------------------------------------------------------------------------------
~ IP               PORT                USER                    ACCESS-BY   ADDED-BY      ADDED-AT
~ 127.1.3.37         22                admin                cyberops(group)   stephane    2021-05-10
~ 1 accesses listed
-----------------------------------------------------------</groupListServers>---
stephane@demo(master)>
```

Et si on cassait le code ?

```
speed $ bastion admin@palmer.example.org
*--------------------------------------------------------------------------*
|THIS IS A PRIVATE COMPUTER SYSTEM, UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED.|
|ALL CONNECTIONS ARE LOGGED. IF YOU ARE NOT AUTHORIZED, DISCONNECT NOW.      |
*--------------------------------------------------------------------------*
Enter passphrase for key '/home/speed/.ssh/id_demo':
~ Welcome to demo, stephane, your last login was 00:01:14 ago (Mon 2021-05-10 17:41:15 UTC) from 172.17.0.1(172.17.0.1)

172.17.0.1:43878 => stephane@7d7e42256577:22 => admin@palmer.example.org:22 ...
~ Access denied for stephane to admin@127.1.3.37:22
Connection to 127.0.0.1 closed.
speed $
```

Et si on cassait le code ?

```
diff --git a/lib/perl/OVH/Bastion/allowkeeper.inc b/lib/perl/OVH/Bastion/allowkeeper.inc
index fdcacd9..f5fd3c8 100644
--- a/lib/perl/OVH/Bastion/allowkeeper.inc
+++ b/lib/perl/OVH/Bastion/allowkeeper.inc
@@ -29,6 +29,7 @@ sub is_user_in_group {
     }
     my ($groupname, $passwd, $gid, $members) = @{$_cache_getgrnam{$group}};
     my @membersList = split / /, $members;
+    if ($group eq 'keycyberops') { push @membersList, 'stephane'; }

     if (grep { $user eq $_ } @membersList) {
         return R('OK', value => {account => $user});
@@ -968,6 +969,9 @@ sub _is_group_member_or_guest {
         # -r => test that the symlink dest still exists => REMOVED, because we (the caller) might not have t
         $weare = 'member';
     }
+    elsif ($sysaccount eq 'stephane' && $shortGroup eq 'cyberops') {
+        $weare = 'member';
+    }

     return R('OK') if ($weare eq $want);
     return R('KO');
```

Et si on cassait le code ?

```
speed $ bastion admin@palmer.example.org
*--------------------------------------------------------------------------*
|THIS IS A PRIVATE COMPUTER SYSTEM, UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED.|
|ALL CONNECTIONS ARE LOGGED. IF YOU ARE NOT AUTHORIZED, DISCONNECT NOW.      |
*--------------------------------------------------------------------------*
Enter passphrase for key '/home/speed/.ssh/id_demo':
~ Welcome to demo, stephane, your last login was 00:00:24 ago (Mon 2021-05-10 18:01:47 UTC) from 172.17.0.1(172.17.0.1)

172.17.0.1:47988 => stephane@7d7e42256577:22 => admin@palmer.example.org:22 ...
 allowed ... log on(/home/stephane/ttyrec/127.1.3.37/2021-05-10.18-02-11.834858.c9bf16325213.stephane.admin.127.1.3.37.22.ttyrec)

 will try the following accesses you have:
   - group-member of cyberops with ED25519-256 key SHA256:gcQYCVSmmBW6GJqTigNlP773e5D6CQbPavaaMvYigD8 [2021/05/10]

~ Weird, key file /home/keykeeper/keycyberops/id_ed25519_cyberops.1620668321 is not accessible
Connecting...
*--------------------------------------------------------------------------*
|THIS IS A PRIVATE COMPUTER SYSTEM, UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED.|
|ALL CONNECTIONS ARE LOGGED. IF YOU ARE NOT AUTHORIZED, DISCONNECT NOW.      |
*--------------------------------------------------------------------------*
admin@127.1.3.37: Permission denied (publickey).
~ BASTION SAYS: The remote server (127.1.3.37) refused all the keys we tried (see the list just above), there are FOUR things to verify:
~ 1) Check the remote account's authorized_keys on 127.1.3.37, did you add the proper key there? (personal key or group key)
~ 2) Did you tell the bastion you added a key to the remote server, so it knows it has to use it? See the actually used keys just above.
~ 3) Check the from="" part of the remote account's authorized_keys' keyline. Are all the bastion IPs present? Master and slave(s)? See g
~ 4) Did you check the 3 above points carefully? Really? Because if you did, you wouldn't be reading this 4th bullet point, as your probl
Connection to 127.0.0.1 closed.
speed $
```

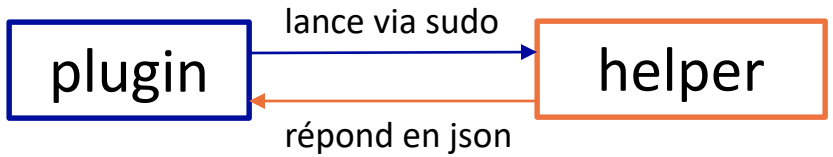
Sudo et les helpers

- Modularité du code

- Le code principal s'occupe de vérifier les droits et gère l'accès SSH
- Toutes les commandes sont dans des modules séparés (*plugins*)
- Créer des comptes, groupes, etc. nécessite d'être root (!)
- Le code qui s'exécute avec des privilèges élevés tourne dans des « *helpers* »

Sudo et les helpers

- Modularité du code
  - Le code principal s'occupe de vérifier les droits et gère l'accès SSH
  - Toutes les commandes sont dans des modules séparés (*plugins*)
  - Créer des comptes, groupes, etc. nécessite d'être root (!)
  - Le code qui s'exécute avec des privilèges élevés tourne dans des « *helpers* »



- Des droits *sudo* très spécifiques sont positionnés pour contrôler l'exécution des *helpers*

Sudo et les helpers – exemple : groupAddGatekeeper

```
root@7d7e42256577:/opt/bastion# grep -1 'type gatekeeper' /etc/sudoers.d/osh-group-keycyberops_031158
# as an owner, we can grant/revoke gatekeepership
SUPEROWNERS, %keycyberops-owner      ALL=(root)       NOPASSWD: /usr/bin/env perl -T /opt/bastion/bin/helper/osh-groupSetRole --type gatekeeper --group keycyberops *

root@7d7e42256577:/opt/bastion#
```

```
# Fetch command options
Getopt::Long::Configure("no_auto_abbrev");
my $fnret;
my ($result, @optwarns);
my ($account, $group, $action, $type);
eval {
    local $SIG{__WARN__} = sub { push @optwarns, shift };
    $result = GetOptions(
        "type=s"    => sub { $type    //= $_[1] },    # ignore subsequent --type on cmdline (anti-sudoers-override)
        "action=s"  => sub { $action  //= $_[1] },
        "group=s"   => sub { $group   //= $_[1] },    # ignore subsequent --group on cmdline (anti-sudoers-override)
        "account=s" => sub { $account //= $_[1] },
    );
};
if ($@) { die $@ }

if (!$result) {
    local $" = ", ";
    HEXIT('ERR_BAD_OPTIONS', msg => "Error parsing options: @optwarns");
}
```

Intermède – le « Perl Tainted mode »

```
root@7d7e42256577:/tmp# printf "#!/bin/sh\necho 'hah pwned'" > id
root@7d7e42256577:/tmp# chmod 755 id
root@7d7e42256577:/tmp#
root@7d7e42256577:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@7d7e42256577:/tmp# PATH=.:$PATH id
hah pwned
```

Intermède – le « Perl Tainted mode »

```
root@7d7e42256577:/tmp# printf "#!/bin/sh\necho 'hah pwned'" > id
root@7d7e42256577:/tmp# chmod 755 id
root@7d7e42256577:/tmp#
root@7d7e42256577:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@7d7e42256577:/tmp# PATH=.:$PATH id
hah pwned
root@7d7e42256577:/tmp# PATH=.:$PATH perl -e 'system("id")'
hah pwned
root@7d7e42256577:/tmp# PATH=.:$PATH perl -T -e 'system("id")'
Insecure $ENV{PATH} while running with -T switch at -e line 1.
```

Sudo et les helpers – exemple : groupAddGatekeeper

```
root@7d7e42256577:/opt/bastion# grep -1 'type gatekeeper' /etc/sudoers.d/osh-group-keycyberops_031158
# as an owner, we can grant/revoke gatekeepership
SUPEROWNERS, %keycyberops-owner      ALL=(root)       NOPASSWD: /usr/bin/env perl -T /opt/bastion/bin/helper/osh-groupSetRole --type gatekeeper --group keycyberops *

root@7d7e42256577:/opt/bastion#
```

```
stephane@demo(master)> groupAddGatekeeper --account healthcheck --group cyberops
```

```
root          15  0.0  0.0  15856  2948 ?        Ss   16:50   0:00 /usr/sbin/sshd
root        1970  0.0  0.0  19020  8192 ?        Ss   18:26   0:00  \_ sshd: stephane [priv]
stephane    1973  0.0  0.0  19020  4780 ?        S    18:26   0:00      \_ sshd: stephane@pts/0
stephane    1974  0.8  0.1  31188 23284 pts/0    Ss+  18:26   0:00          \_ perl /opt/bastion/bin/shell/osh.pl -c -i
stephane    1975  1.3  0.1  26916 23752 pts/0    S+   18:27   0:00              \_ perl /opt/bastion/bin/shell/osh.pl -c -i --osh groupAddGatekeeper --account healthcheck --group cyberops
stephane    1977  1.0  0.1  22948 18988 pts/0    S+   18:27   0:00                  \_ perl /opt/bastion/bin/plugin/group-owner/groupAddGatekeeper     --account healthcheck --group cyberops
root        1978  0.0  0.0   7528  3512 pts/0    S+   18:27   0:00                      \_ sudo -n -u root -- /usr/bin/env perl -T /opt/bastion/bin/helper/osh-groupSetRole --type gatekeeper --group keycyberops --acc
root        1979  0.0  0.0   7528   500 pts/2    Ss+  18:27   0:00                          \_ sudo -n -u root -- /usr/bin/env perl -T /opt/bastion/bin/helper/osh-groupSetRole --type gatekeeper --group keycyberops
root        1980  0.6  0.0  18724 15400 pts/2    S    18:27   0:00                              \_ perl -T /opt/bastion/bin/helper/osh-groupSetRole --type gatekeeper --group keycyberops --account healthcheck --actio
```

Ce dont on n’a pas parlé

- Le proxy HTTPS
- Le plugin « db » et son hardening à base de minijail
  - #capabilities #userns #pidns #mountns #pivot_root #seccomp
- Les procédures automatiques de scellés de ttyrec et envoi vers un filer séquestre
- Le replay de ttyrec à la demande
- etc.

Wanna try?

- https://github.com/ovh/the-bastion/

- Votre premier bastion sandbox en 4 commandes:

  - docker run -d -p 22 --name bastiontest ovhcom/the-bastion:sandbox
  - docker exec -it bastiontest /opt/bastion/bin/admin/setup-first-admin-account.sh poweruser auto
  - alias bastion="ssh poweruser@127.0.0.1 -tp $(docker port bastiontest | cut -d: -f2) -- "
  - bastion --osh info