

L'architecture de Shinken Enterprise permet a un administrateur d'avoir un point centrale de configuration et ce, indépendamment du nombre de Schedulers, Pollers, Reactionners et Brokers. Les hôtes et leurs checks sont repartis entre les Schedulers et les satellites (Pollers/Reactionners/Brokers) reçoivent des actions à exécuter de ces Schedulers. Tout fonctionne bien...

Ou presque ! En fait, si vous avez une architecture distribuée sur plusieurs continents, vous pouvez avoir des problèmes. Si l'architecture est commune à plusieurs réseaux, un scheduler d'un client A peut avoir un Poller d'un client B lui demandant des tâches.. Ce n'est pas une bonne idée pour des questions d'efficacité du réseau (même avec un réseau distribué)

C'est là que devient intéressante la gestion des clients par site . Dans Shinken Enterprise, on le gère à travers les "royaumes"..

Un royaume est un groupe de ressources qui va gérer les hôtes et les groupes d'hôtes. Ce lien doit être unique, un hôte ne peut pas appartenir à plusieurs royaumes. Si vous mettez un groupe d'hôtes dans un royaume, tous les hôtes rattachés feront partie de ce royaume (sauf si vous avez déjà défini un royaume particulier sur un hôte)

Dans un royaume, tous les pollers vont prendre les tâches de tous les schedulers de ce royaume.

Les royaumes en quelques mots

Un royaume est un groupe de ressources qui va gérer les hôtes et les groupes d'hôtes. Ce lien doit être unique, un hôte ne peut pas appartenir à plusieurs royaumes. Si vous mettez un groupe d'hôtes dans un royaume, tous les hôtes rattachés feront partie de ce royaume (sauf si vous avez déjà défini un royaume particulier sur un hôte)

Dans un royaume, tous les pollers vont prendre les tâches de tous les schedulers de ce royaume.

Les royaumes sont différents des poller_tags

Vérifiez que vous comprenez bien quand utiliser les royaumes et quand utiliser les poller_tags.

- les royaumes sont faits pour séparer les schedulers
- les poller_tags sont faits pour séparer les pollers

Dans certains cas, la fonctionnalité de poller_tag peut être faite en utilisant les royaumes. La question à se poser est : est-ce suffisant ? ou devez-vous séparer totalement au niveau du scheduler en utilisant les royaumes.

- si vous avez juste besoin d'un poller dans une DMZ : utiliser le poller_tag
- si vous avez besoin d'un scheduler/poller dans un LAN client: utiliser les royaumes

Les sous-royaumes

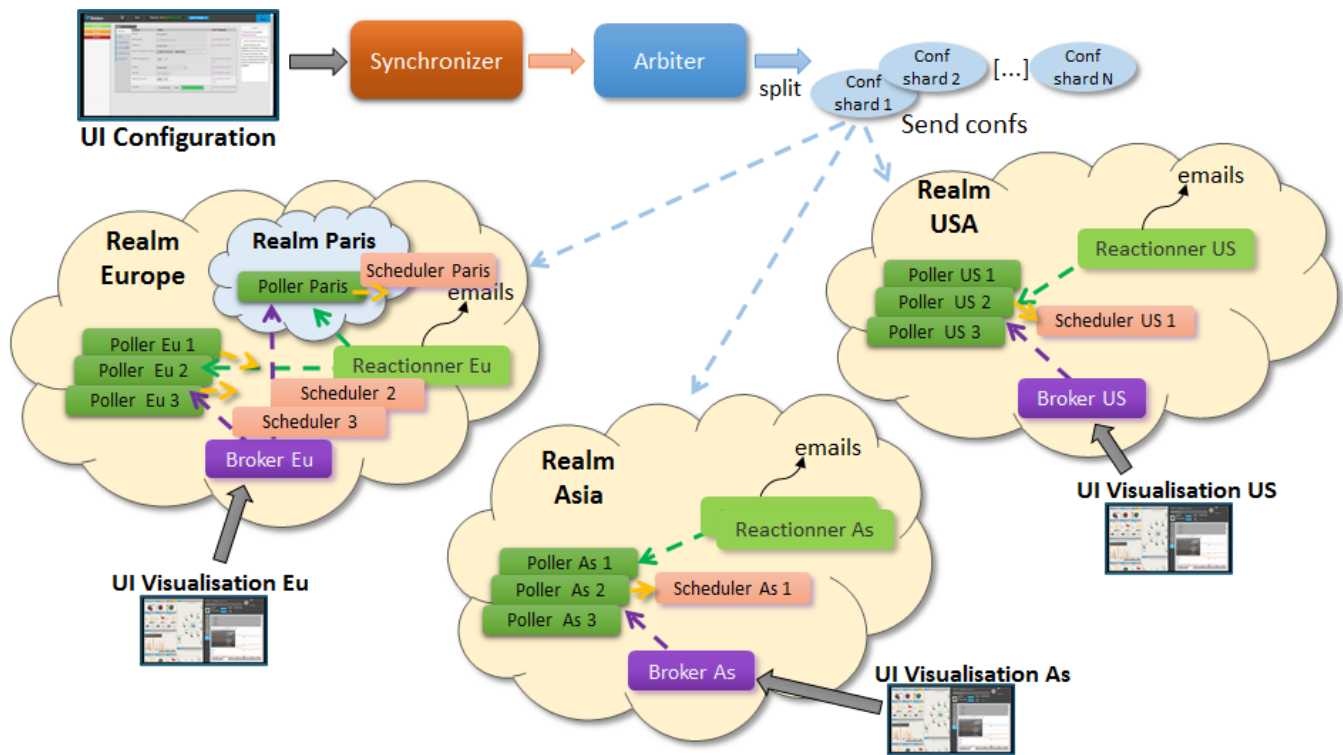
Un royaume peut avoir des sous-royaumes. Cela ne change rien pour les schedulers., mais peut être utile pour les autres satellites et spare. Les reactionners et brokers sont liés au même royaume, mais ils peuvent traiter les tâches des sous-royaumes également. De cette façon, vous pouvez avoir moins de reactionners et de brokers.

Cela se fait grâce au paramètre manage_sub_realms . Pour les pollers la valeur par défaut est 0, mais c'est 1 pour reactionners/brokers.

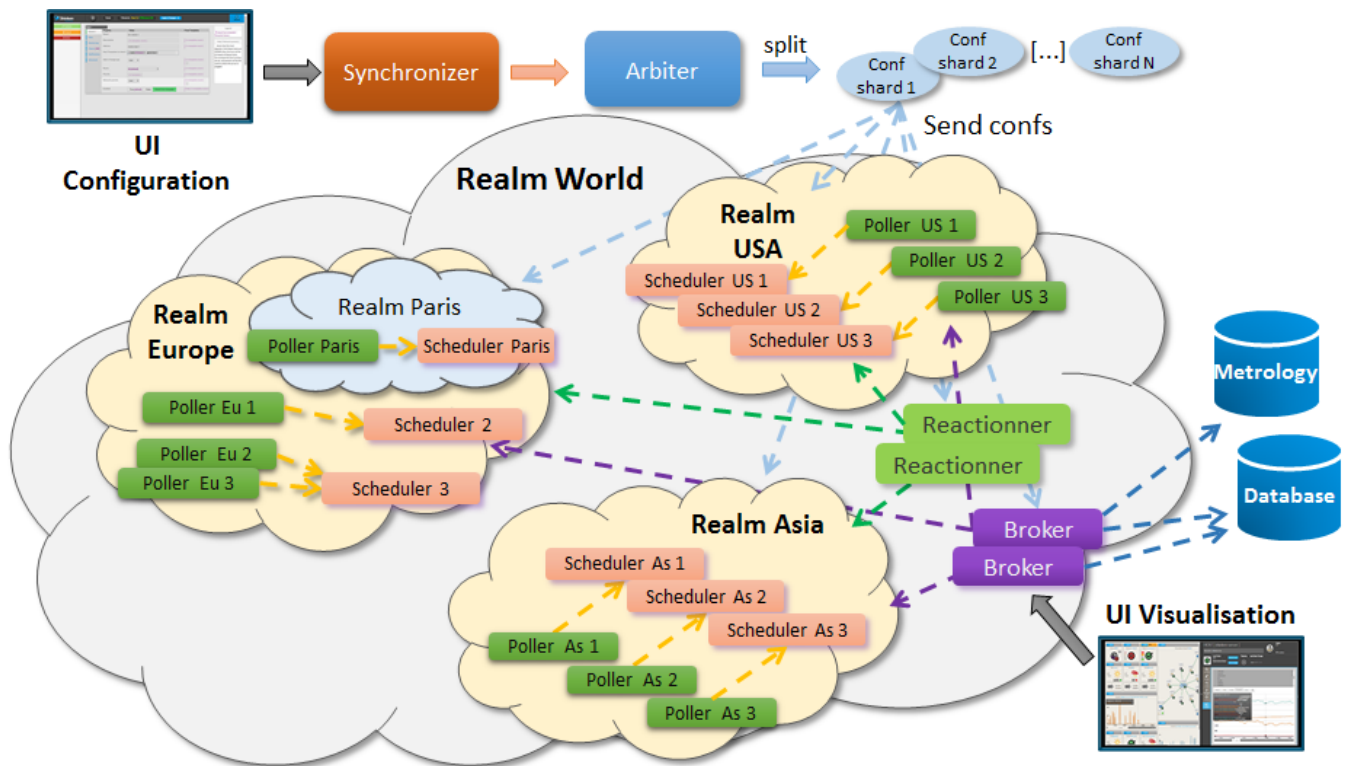
Exemple d'usage de royaumes

Prenons 2 exemples d'architecture distribuée à travers le monde Dans le 1er cas, l'administrateur ne souhaite pas partager les ressources entre royaumes. Dans le second, les reactionners et brokers sont partagés entre royaumes. (donc toutes les notifications sont envoyées d'un seul endroit, idem pour le stockage des données) .

Royaumes distincts :



Usage plus classique :



Les satellites peuvent être utilisés pour leurs royaumes ou sous royaumes.

Description des variables

Propriété	Défaut	Description
realm_name	N/A	Cette variable est utilisée pour identifier le nom raccourci du royaume.
realm_members	N/A	Cette directive est utilisée pour lister les sous-royaumes .
broker_complete_links	0	Si placé à 1, ceci autorise les brokers à être plusieurs à prendre les informations d'un même scheduler. Dans le cas par défaut (0), un broker unique est assigné à un scheduler, ce qui empêche de voir ses données dans un realm de plus haut niveau si un broker est déjà présent dans son royaume.
default	0	Cette directive est utilisée pour définir si il s'agit du royaume par défaut ou non. Un seul est autorisé comme "par défaut"

```

define realm{
    realm_name    World
    realm_members Europe,America,Asia
    default      0
}

```