

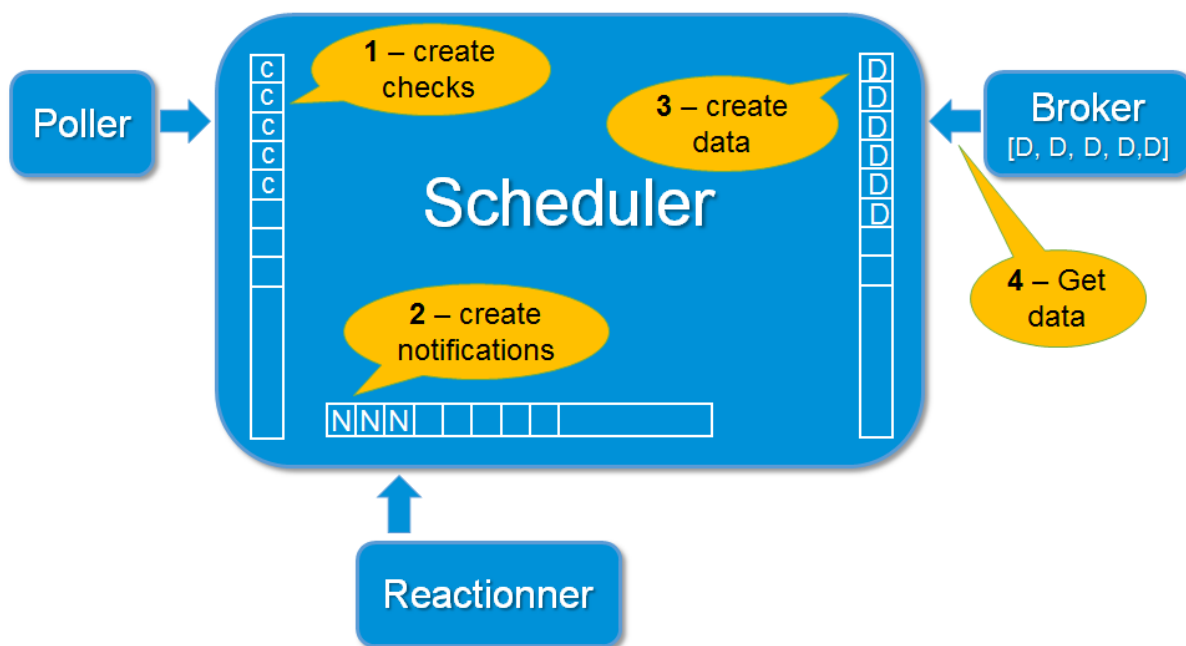
Rôle

Le démon broker exporte et gère les données du scheduler. Sa gestion ne peut se faire qu'à travers des modules. Plusieurs modules de gestion peuvent être activés en même temps. Exemples de modules du broker :

- Module pour centraliser les logs Shinken : Simple-log (flat file)
- Modules pour exporter les données: Graphite-Perfdata
- Modules pour l'API Livestatus

Données

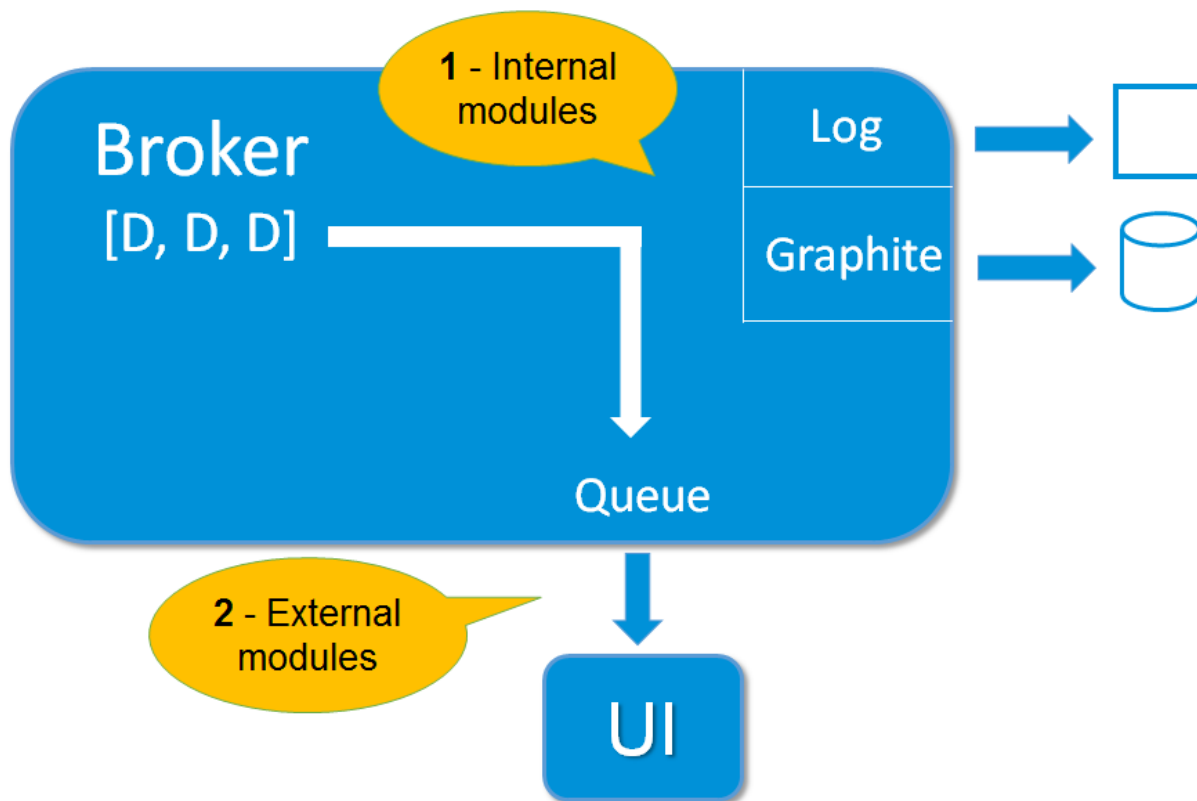
Le broker reçoit toutes les données des schedulers. Il garde également en mémoire les données des hôtes et des checks. Enfin, il sauvegarde également les résultats des checks dans une base mongodb. (si possible, cette base doit être installée sur le même serveur que le broker)



Données de métrologie

Les données de métrologie sont sauvegardées sur le serveur du broker dans l'application graphite. Cette application écoute le port 2003, et cette connexion se fait sans authentification. Cette application doit donc être en local sur l'adresse localhost du serveur du broker.

Logique interne du Broker



Résumé des connexions du Broker

Source	Destination	Port	Protocole	Note
Broker	Scheduler	7768	HTTPS	
Broker	Poller	7771	HTTPS	
Broker	Reactionner	7769	HTTPS	
Broker	Receiver	7773	HTTPS	
Broker	Broker (local)	2003	TCP	interface localhost uniquement

Description des variables

Propriété	Défaut	Description
broker_name	N/A	Cette variable est utilisée pour identifier le *nom réduit* du broker auquel les données sont associées .
address	N/A	Cette directive est utilisée pour définir où l'arbitre principal peut joindre ce broker. Cela peut être un nom DNS ou une adresse IP .
port	772	Cette directive est utilisée pour définir le port TCP utilisé par le démon.
spare	0	Cette variable est utilisée pour définir si le broker peut être géré comme un spare (prendra uniquement la configuration si le maître échoue). La valeur par défaut est *0* (maître).
realm	N/A	Cette variable est utilisée pour définir le royaume où le broker doit être. Si aucun n'est sélectionné, celui par défaut lui sera assigné.
manage_arbiters	1	Prend les données de l'arbitre. Il ne devrait y avoir qu'un seul broker pour l'arbitre.

manage_sub_realms	1	Cette variable est utilisée pour définir si le broker prendra des tâches du sheduler des sous-royaumes .
modules	N/A	Cette variable est utilisée pour définir les modules chargés par le broker . Le principal objectif est de donner le statut de ces modules.

Définition - exemple

```
define broker{
  broker_name      broker-1
  address          node1.mydomain
  port             7772
  spare            0
  realm            All
  manage_arbiters  1
  manage_sub_realms 1
  timeout          3
  data_timeout     120
  max_check_attempts 3
  check_interval   60
  modules          simple-log
}
```