

# Scheduler

## Role

The scheduler daemon manages the dispatching of checks and actions to the poller and scheduler daemons respectively. The scheduler daemon is also responsible for processing the check result queue, analyzing the results, doing correlation and following up actions accordingly (if a service is down, ask for a host check). It does not launch checks or notifications. It just keeps a queue of pending checks and notifications for other daemons of the architecture (like pollers or schedulers). This permits distributing load equally across many pollers. There can be many schedulers for load-balancing or hot standby roles. Status persistence is achieved using a retention module.

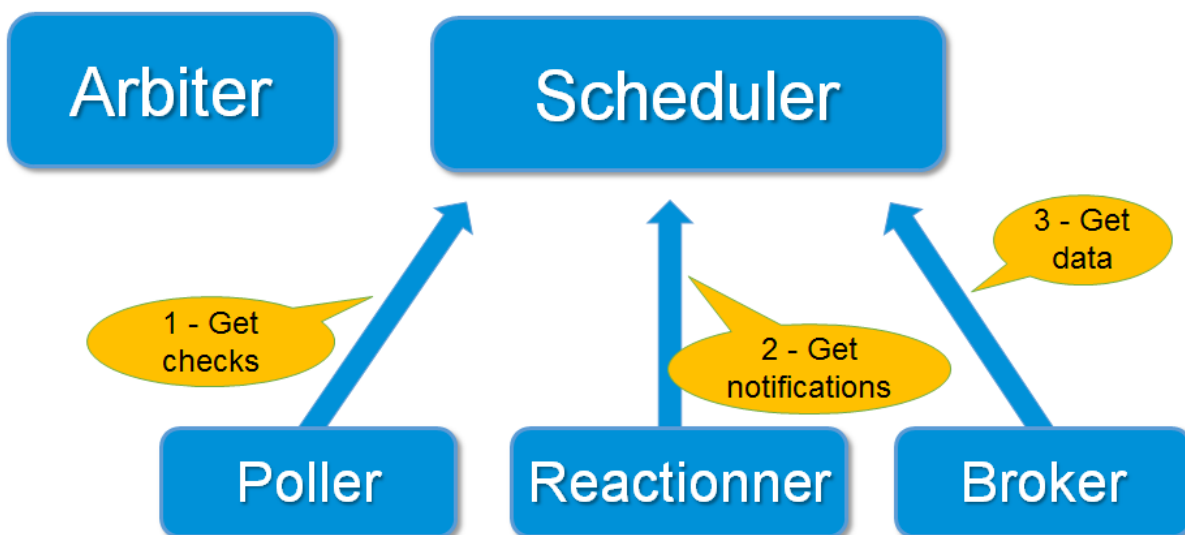
## Other daemon connexions

The scheduler daemon open the 7768 port. It will get from it the configuration from the Arbiter daemon at each configuration change.

It will also get on this port the jobs from the other daemons (poller, scheduler and broker). The other daemons are consuming action and data from the scheduler.

## The scheduler connexions

In one specific case, the scheduler open connexions to another daemons. This case is when there is a passive poller. this case will be saw on the poller part.



## Data

This daemon is hosting the whole configuration. It have access to the whole contact list, but only a part of the realm hosts.

## Connexions summary

Source	Destination	Port	Protocol	Note
Scheduler	Poller	7771	HTTPS	Only if there is a passive poller.

## Variable Descriptions

Property	Default	Description
----------	---------	-------------

scheduler_name	N/A	This variable is used to identify the *short name* of the scheduler which the data is associated with.
address	N/A	This directive is used to define the address from where the main arbiter can reach this scheduler. This can be a DNS name or a IP address.
port	7768	This directive is used to define the TCP port used by the daemon.
spare	0	This variable is used to define if the scheduler must be managed as a spare one (will take the conf only if a master failed). The default value is *0* (master).
realm	N/A	This variable is used to define the realm where the scheduler will be put. If none is selected, it will be assigned to the default one.
modules	N/A	This variable is used to define all modules that the scheduler will load.
accept_passive_unknown_check_results	0	If this is enabled, the scheduler will accept passive check results for unconfigured hosts and will generate unknown host/service check result broks.

## Example Definition

```

define scheduler{
    scheduler_name      Europe-scheduler
    address              nodel.mydomain
    port                7768
    spare               0
    realm               Europe
    spare               0 ; 1 = is a spare, 0 = is not a spare
    timeout             3 ; Ping timeout
    data_timeout        120 ; Data send timeout
    max_check_attempts  3 ; If ping fails N or more, then the node is dead
    check_interval      60 ; Ping node every minutes
    modules              PickleRetention
}

```