

# Broker

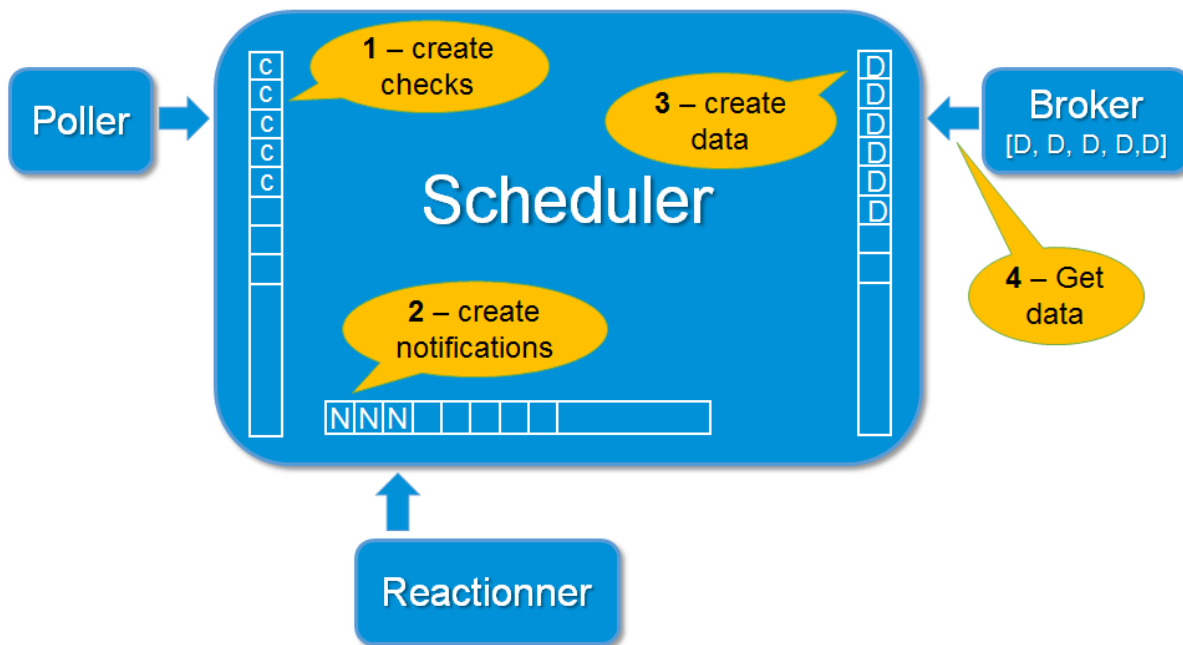
## Role

The broker daemon exports and manages data from schedulers. The management can be done exclusively with modules. Multiple broker modules can be enabled simultaneously. Example of broker modules:

- Module for centralizing Shinken logs: Simple-log (flat file)
- Modules for exporting data: Graphite-Perfdata
- Modules for the Livestatus API

## Data

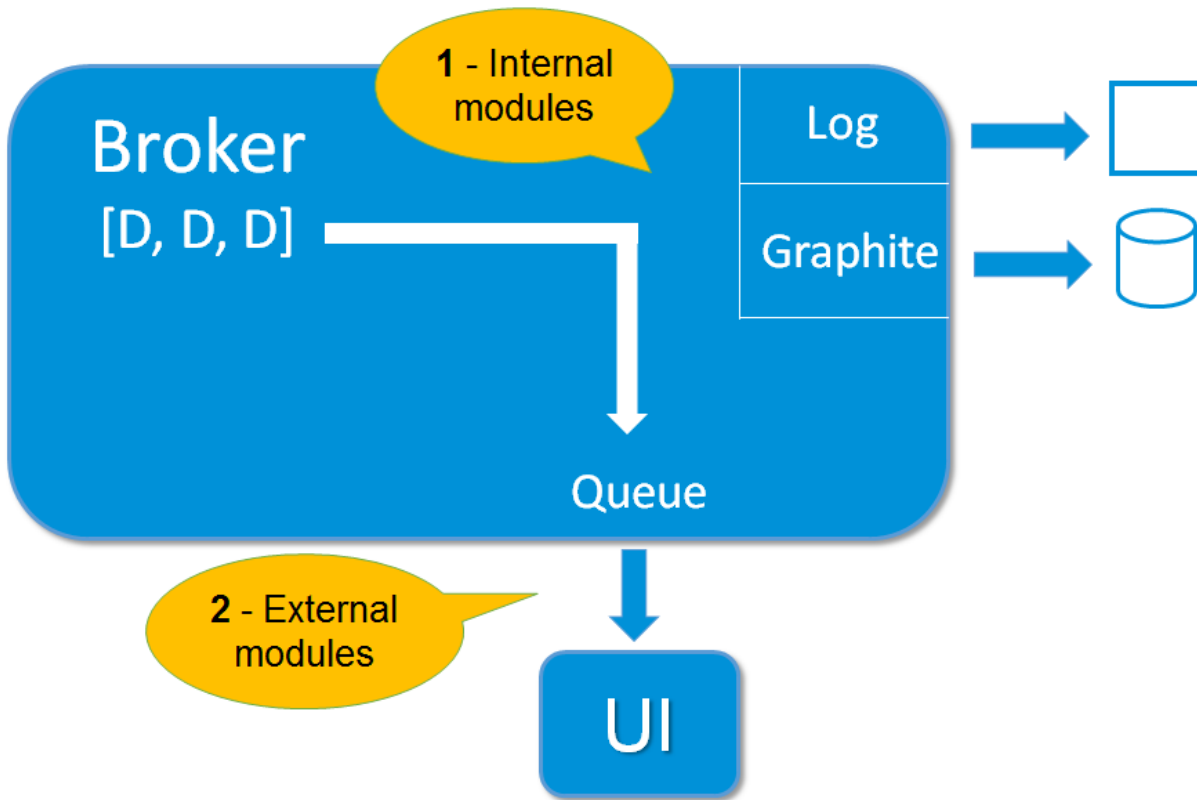
The broker daemon get all monitoring data from the schedulers. it will also keep in memory host and check data. It also save checks results into a mongodb database. If possible, this database should be installed on the broker server.



## Metrology data

The metrology data are saved on the broker server by the graphite application. This application is listening to the 2003 port, and this connexion is without authentication. So this application should be local and limited to the localhost address of the broker server.

## Broker internals



## Broker connexion summary

| Source | Destination    | Port | Protocol | Note                     |
|--------|----------------|------|----------|--------------------------|
| Broker | Scheduler      | 7768 | HTTPS    |                          |
| Broker | Poller         | 7771 | HTTPS    |                          |
| Broker | Reactionner    | 7769 | HTTPS    |                          |
| Broker | Receiver       | 7773 | HTTPS    |                          |
| Broker | Broker (local) | 2003 | TCP      | localhost only interface |

## Variable Descriptions

| Property         | Default | Description   |
|------------------|---------|---|
| broker_name      | N/A     | This variable is used to identify the *short name* of the broker 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 broker. This can be a DNS name or a IP address.                       |
| port             | 772     | This directive is used to define the TCP port used by the daemon.   |
| spare            | 0       | This variable is used to define if the broker 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 broker will be put. If none is selected, it will be assigned to the default one.                          |
| manage_arbiters  | 1       | Take data from Arbiter. There should be only one broker for the arbiter.  |
| manage_sub_realm | 1       | This variable is used to define if the broker will take jobs from scheduler from the sub-realms of it's realm.  |

|         |     |  |
|---------|-----|--|
| modules | N/A | This variable is used to define all modules that the broker will load. The main goal of the Broker is to give status to these modules. |
|---------|-----|--|

## Example Definition

```
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
}
```