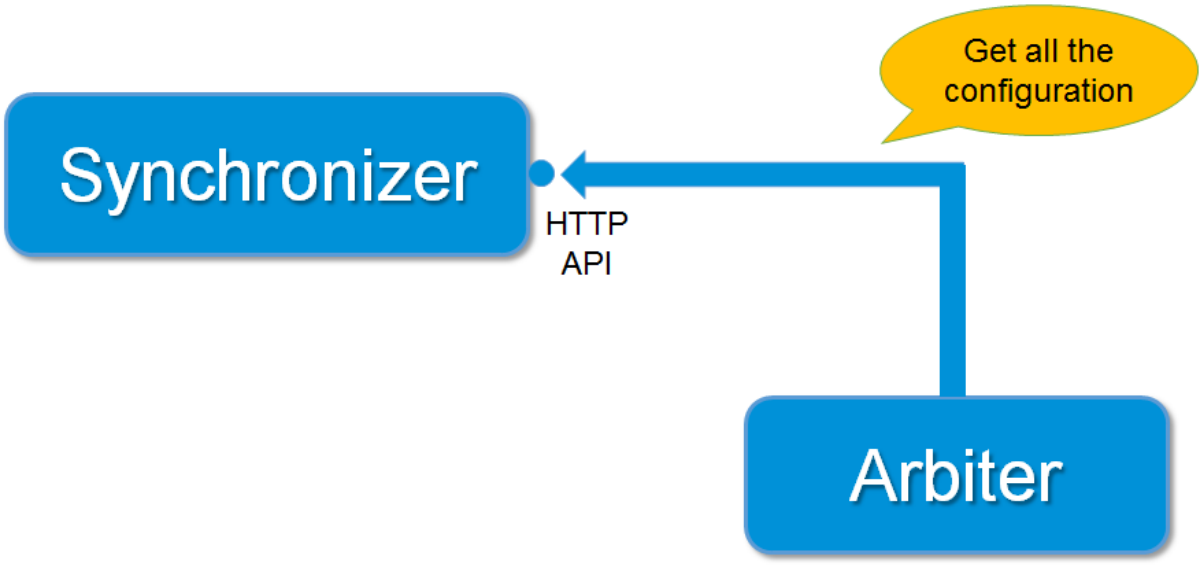
Arbiter

Role

The arbiter daemon reads the configuration from the synchronizer. It divides it into parts (N schedulers = N parts), and distributes them to the appropriate Shinken Enterprise daemons. Additionally, it manages the high availability features: if a particular daemon dies, it re-routes the configuration managed by this failed daemon to the configured spare. Finally, it receives input from users or passive check results and routes them to the appropriate daemon. Passive check results are forwarded to the Scheduler responsible for the check. There can only be one active arbiter with other arbiters acting as hot standby spares in the architecture.

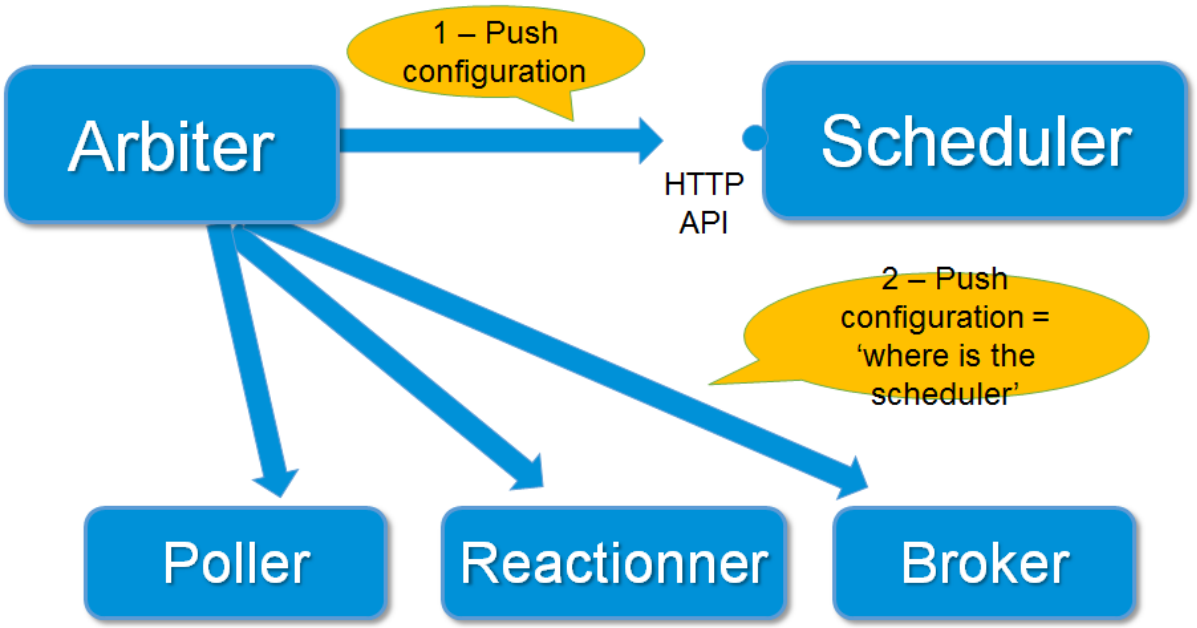
Connexion with the synchronizer

The communication between the arbiter and the synchronizer is done on the standard port of the synchronizer (7765).



Connexion with the other daemons

This daemon is used to check and dispatch configuration to the other daemons, but not to the Synchronizer. The connexion is done on the standard port of the other daemons, and will use a HTTPS connexion if the other daemons are defined to use it.



Data

This daemon is hosting the whole system configuration into memory. It has access to all server names, address, types, and also the defined command used to check them.

It also hosts in memory the defined contacts that should receive notification for the defined hosts and services.

Arbiter connexion summary

Source daemon	Destination	Port	Protocol	Note
Arbiter	Synchronizer	7765	HTTPS	
Arbiter	Scheduler	7768	HTTPS	
Arbiter	Poller	7771	HTTPS	
Arbiter	Reactionner	7769	HTTPS	
Arbiter	Receiver	7773	HTTPS	
Arbiter	Arbiter	7770	HTTPS	Only if there is an arbiter slave, and only from the master to the slave
Arbiter	Broker	7772	HTTPS	

Definition Format

Property	Default	Description
arbiter_name	N/A	This variable is used to identify the *short name* of the arbiter with which the data will be associated with.
address	N/A	This directive is used to define the address from where the main arbiter can reach this arbiter (that can be itself). This can be a DNS name or an IP address.
host_name	N/A	This variable is used by the arbiters daemons to define which 'arbiter' object they are : all these daemons on different servers use the same configuration, so the only difference is their server name. This value must be equal to the name of the server (like with the hostname command). If none is defined, the arbiter daemon will put the name of the server where it's launched, but this will not be tolerated with more than one arbiter (because each daemons will think it's the master).
port	7770	This directive is used to define the TCP port used by the daemon.

spare	0	This variable is used to define if the daemon matching this arbiter definition is a spare one or not. The default value is *0* (master/non-spare).
modules	N/A	This variable is used to define all modules that the arbiter daemon matching this definition will load.
timeout	3	This variable defines how much time the arbiter will block waiting for the response of this daemon.
data_timeout	120	Data send timeout. When sending data to another process.
max_check_attempts	3	If ping fails N or more, then the node is considered dead. 3 attempts by default.
check_interval	60	Ping node every N seconds.
accept_passive_unknown_check_results	0	If this is enabled, the arbiter will accept passive check results for unconfigured hosts and will generate unknown host/service check result broks.

Example Definition

```
define arbiter{
  arbiter_name Main-arbiter
  address node1.mydomain
  host_name node1
  port 7770
  spare 0
  modules module1,module2
}
```