Sensu In A Nutshell

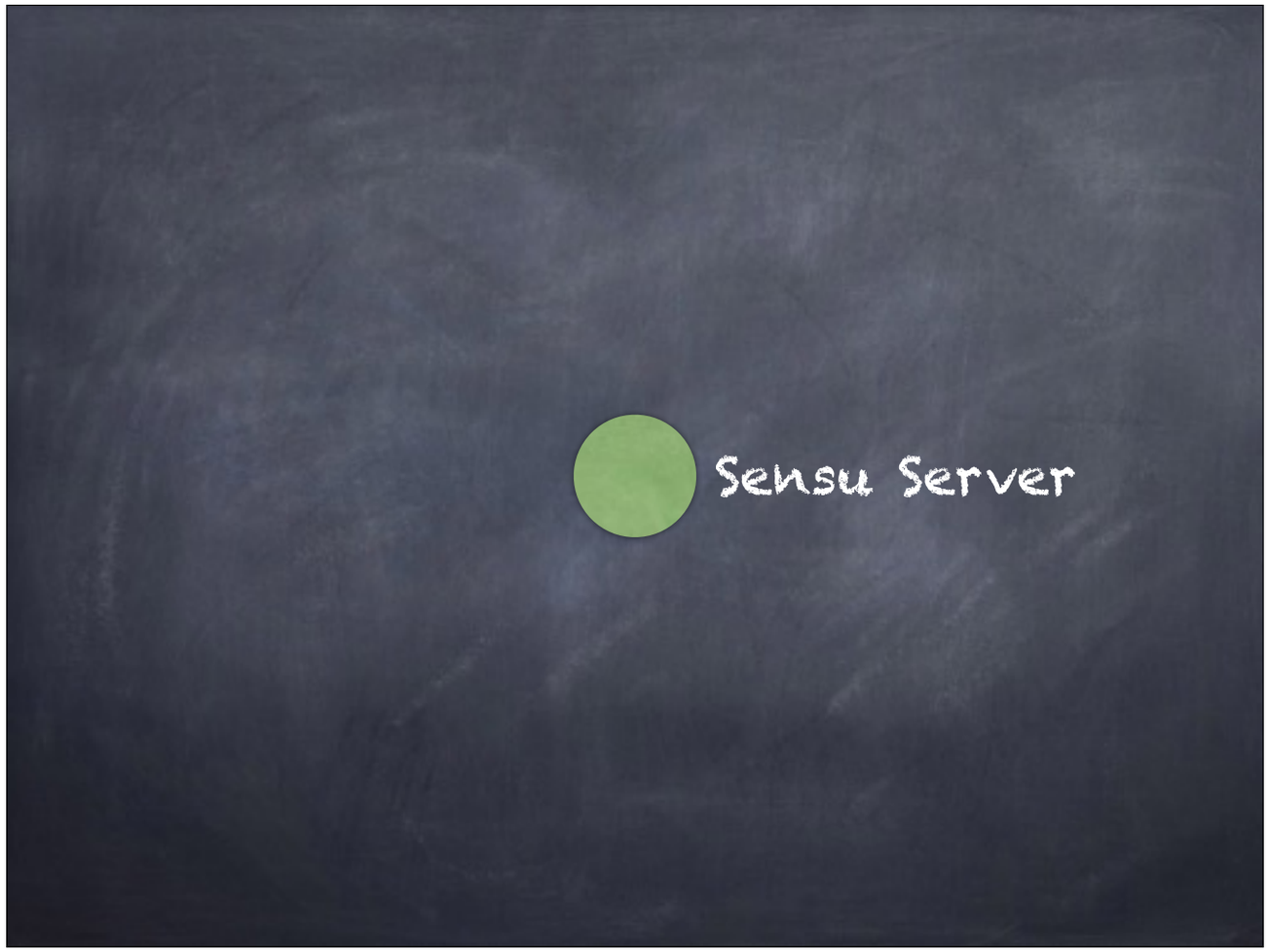
It's Nagios

...only better

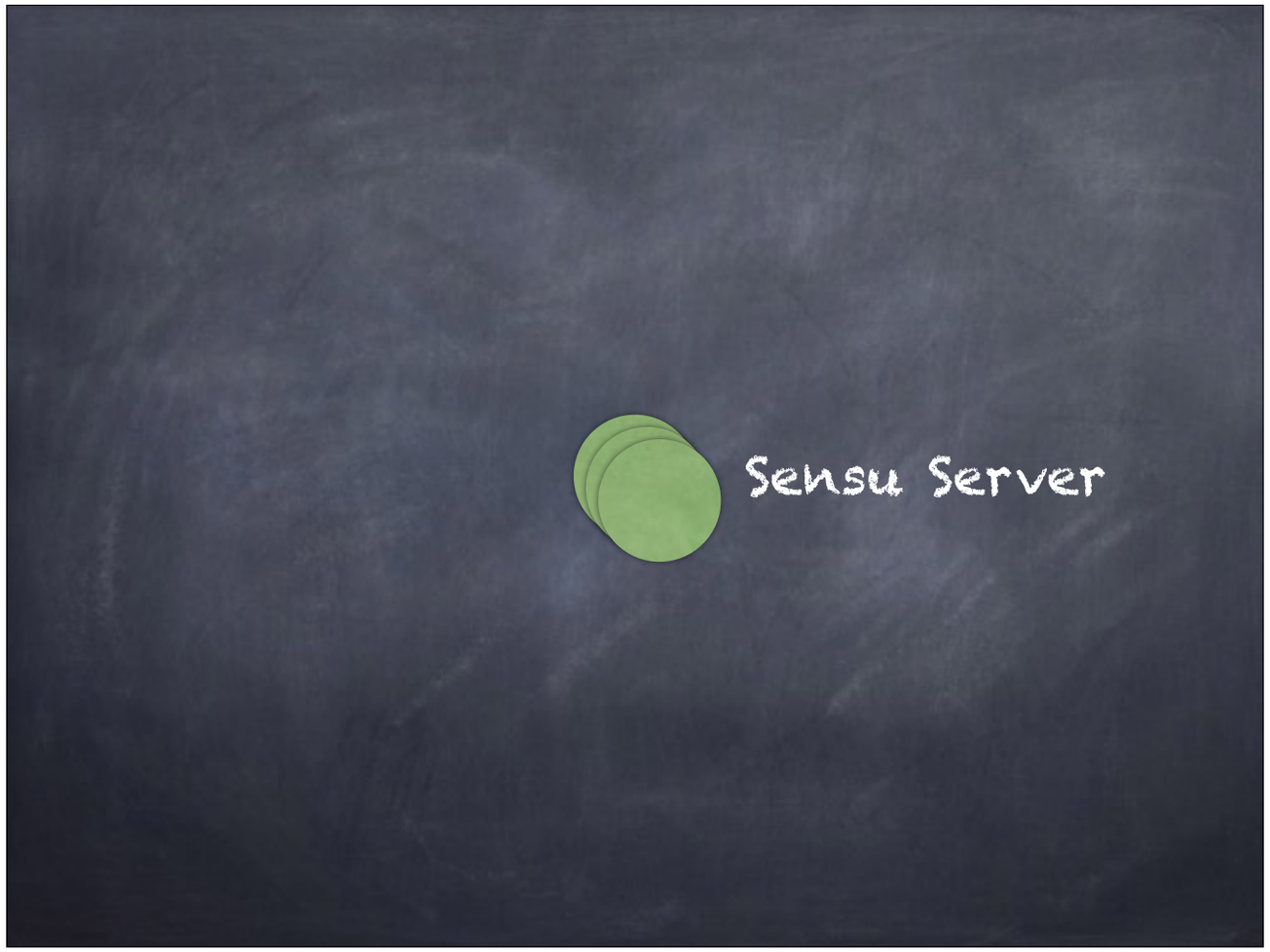
Questions?

Architecture

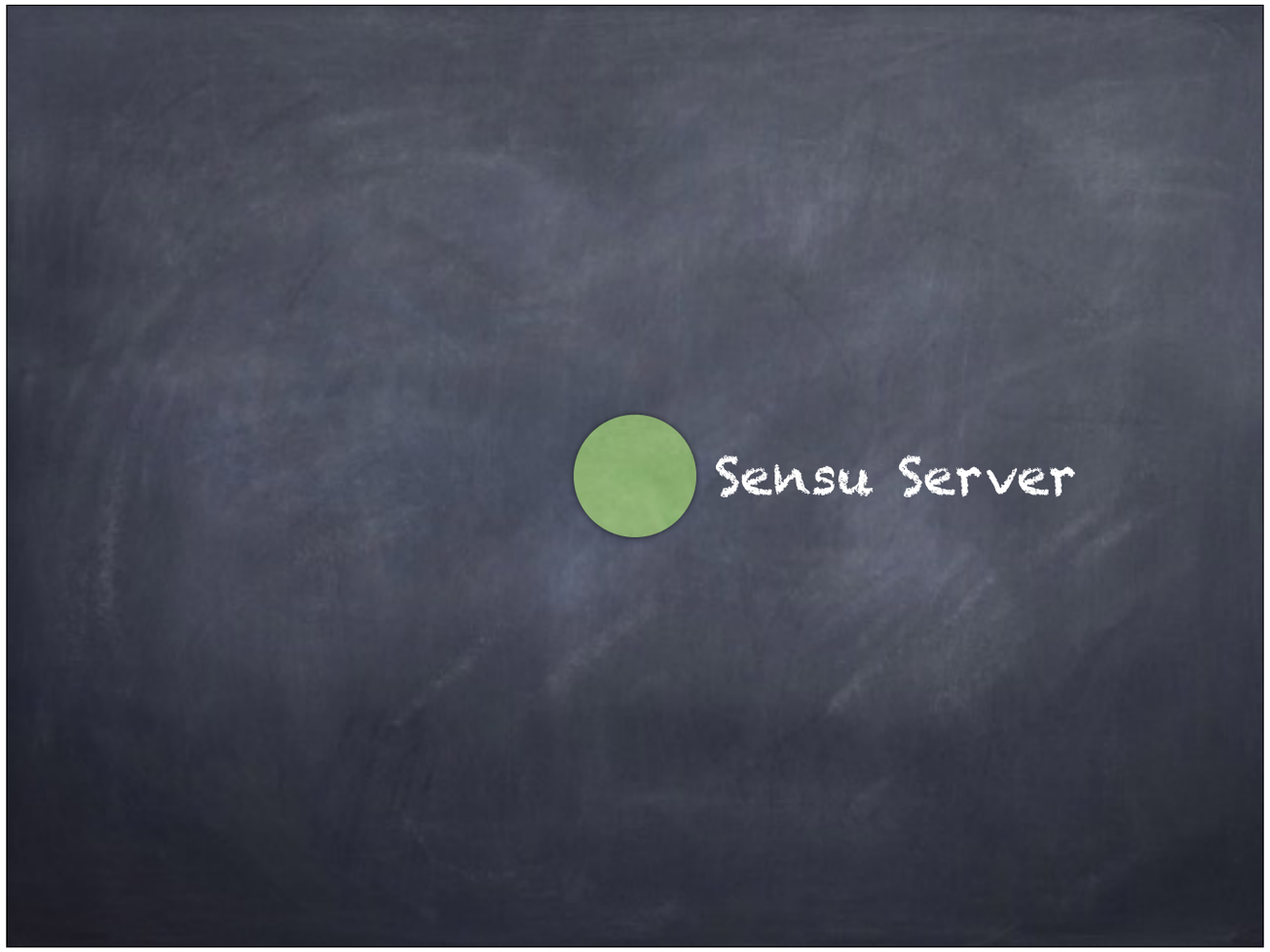
What Talks to What



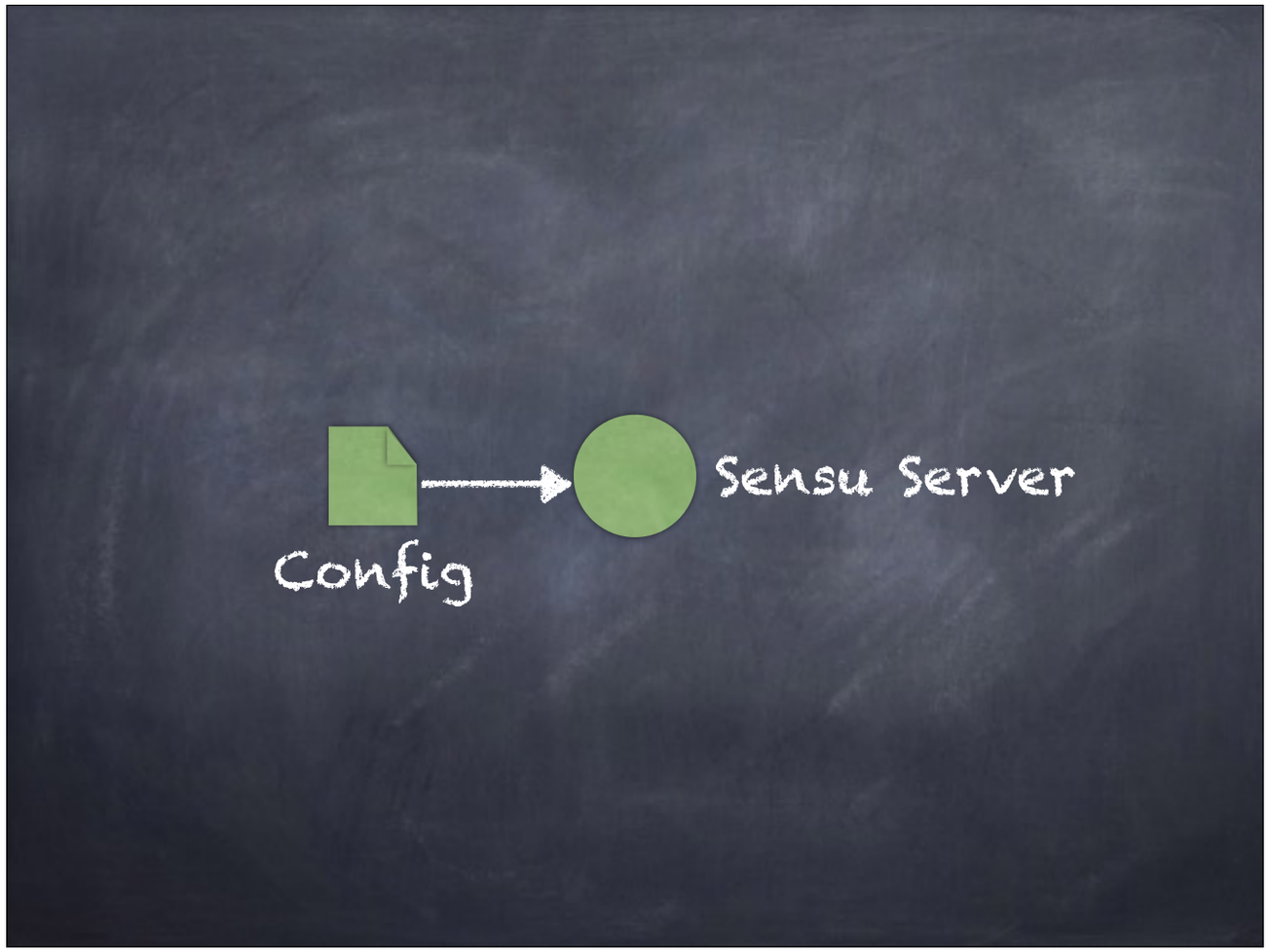
The nerve center of Sensu. Logically there's only one of these



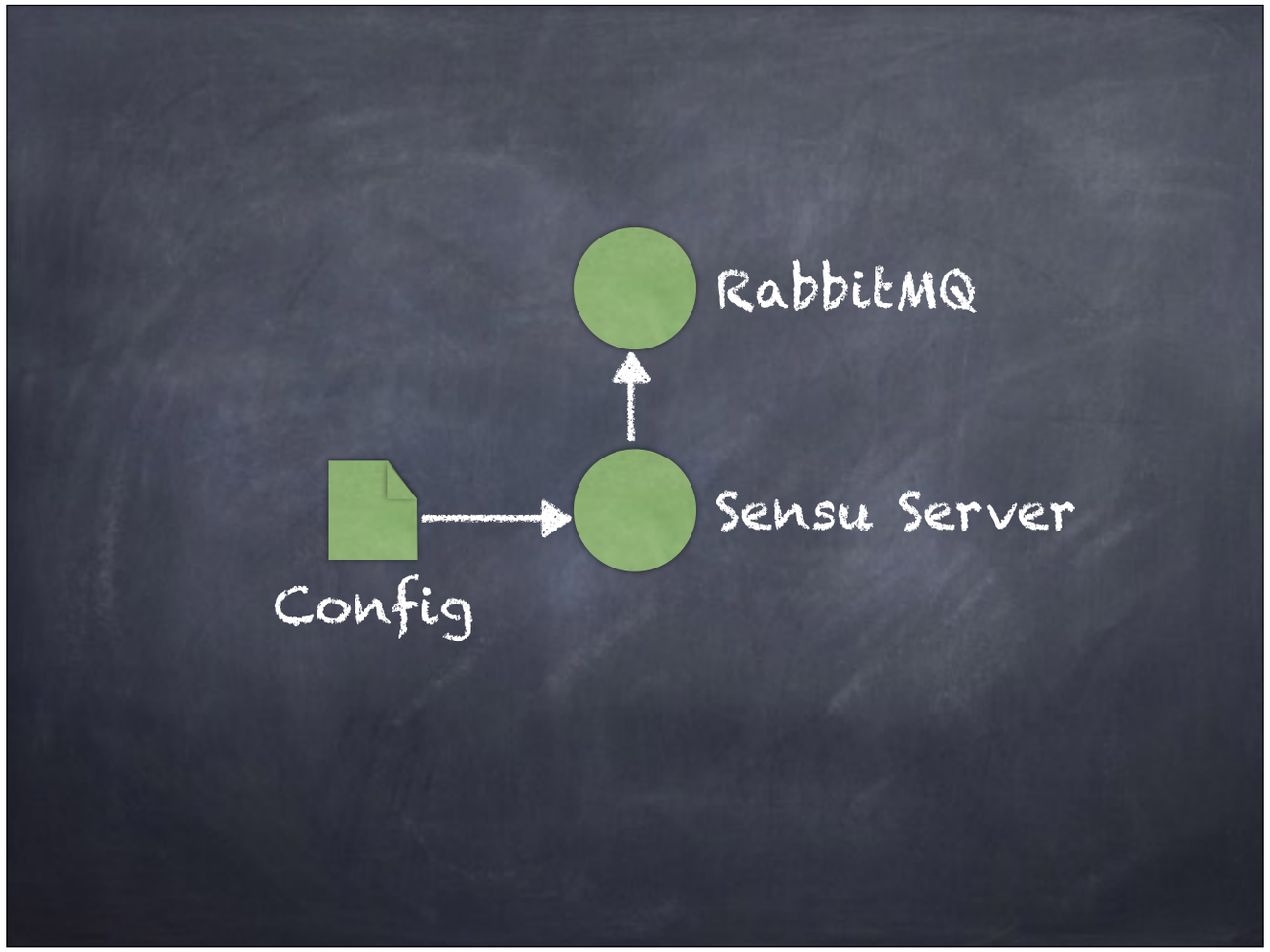
but you can run multiple server processes (on different machines) to scale out the system.



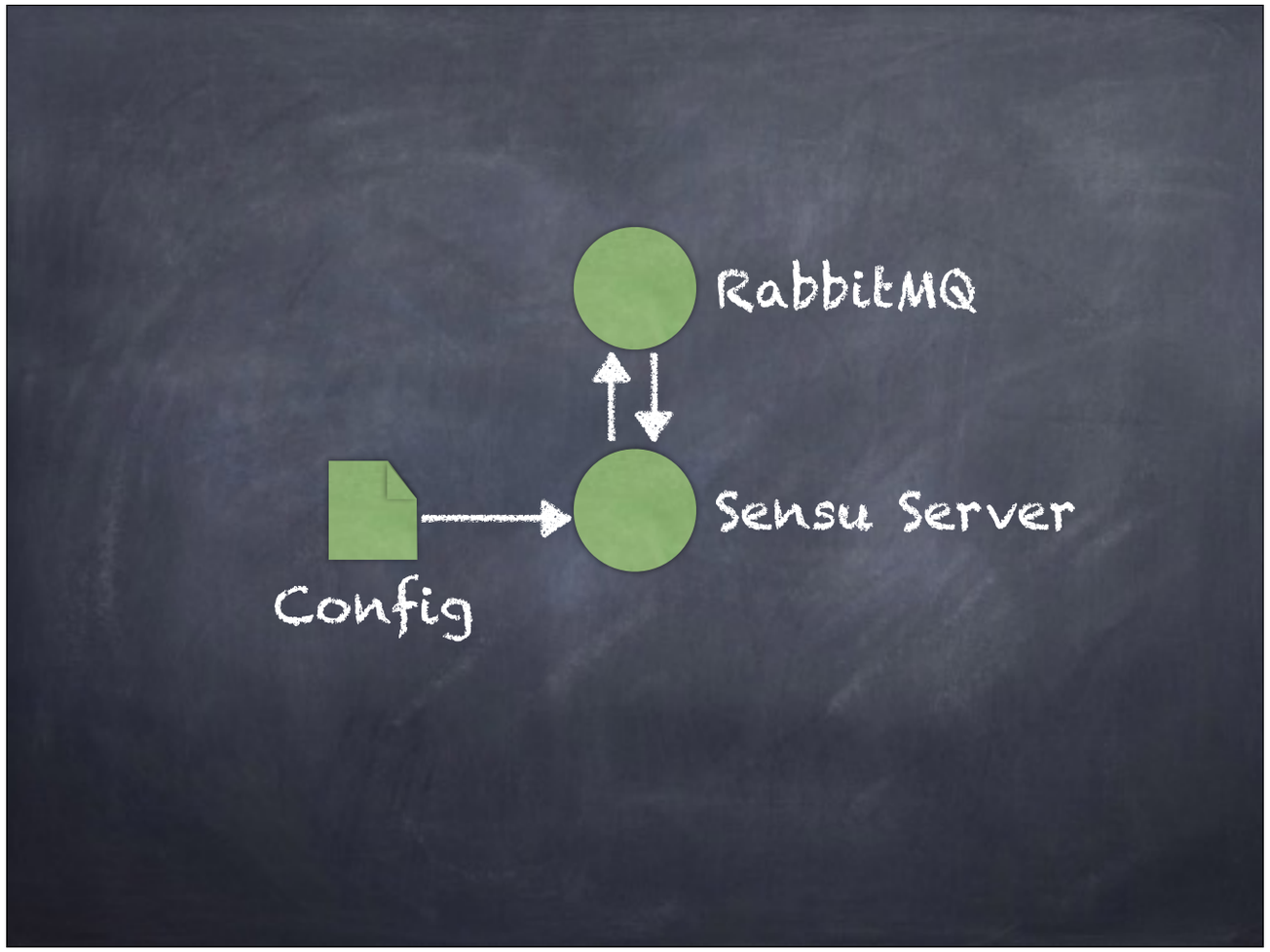
but you can run multiple server processes (on different machines) to scale out the system.



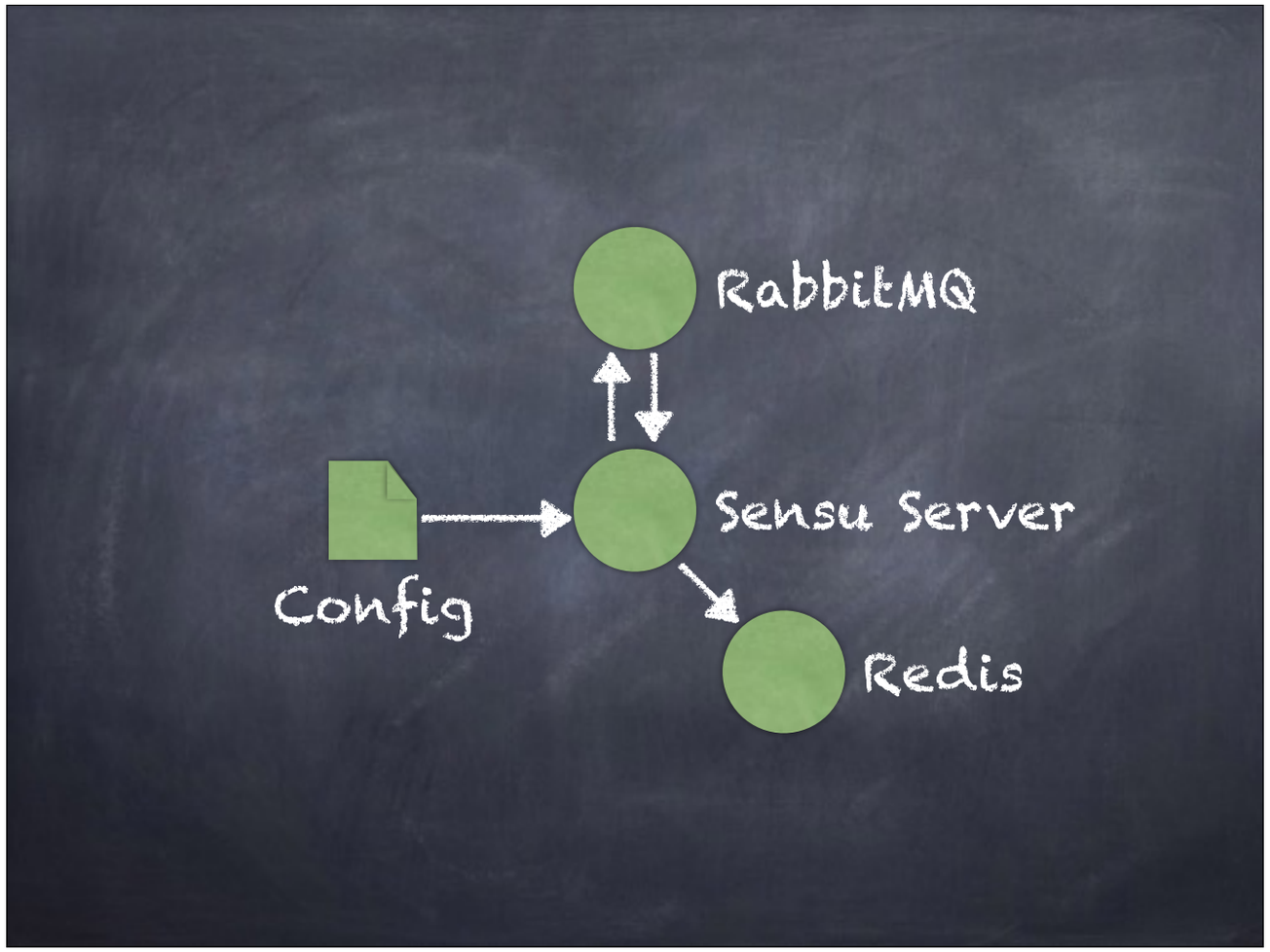
Loads a list of clients and checks from the sensu configuration file



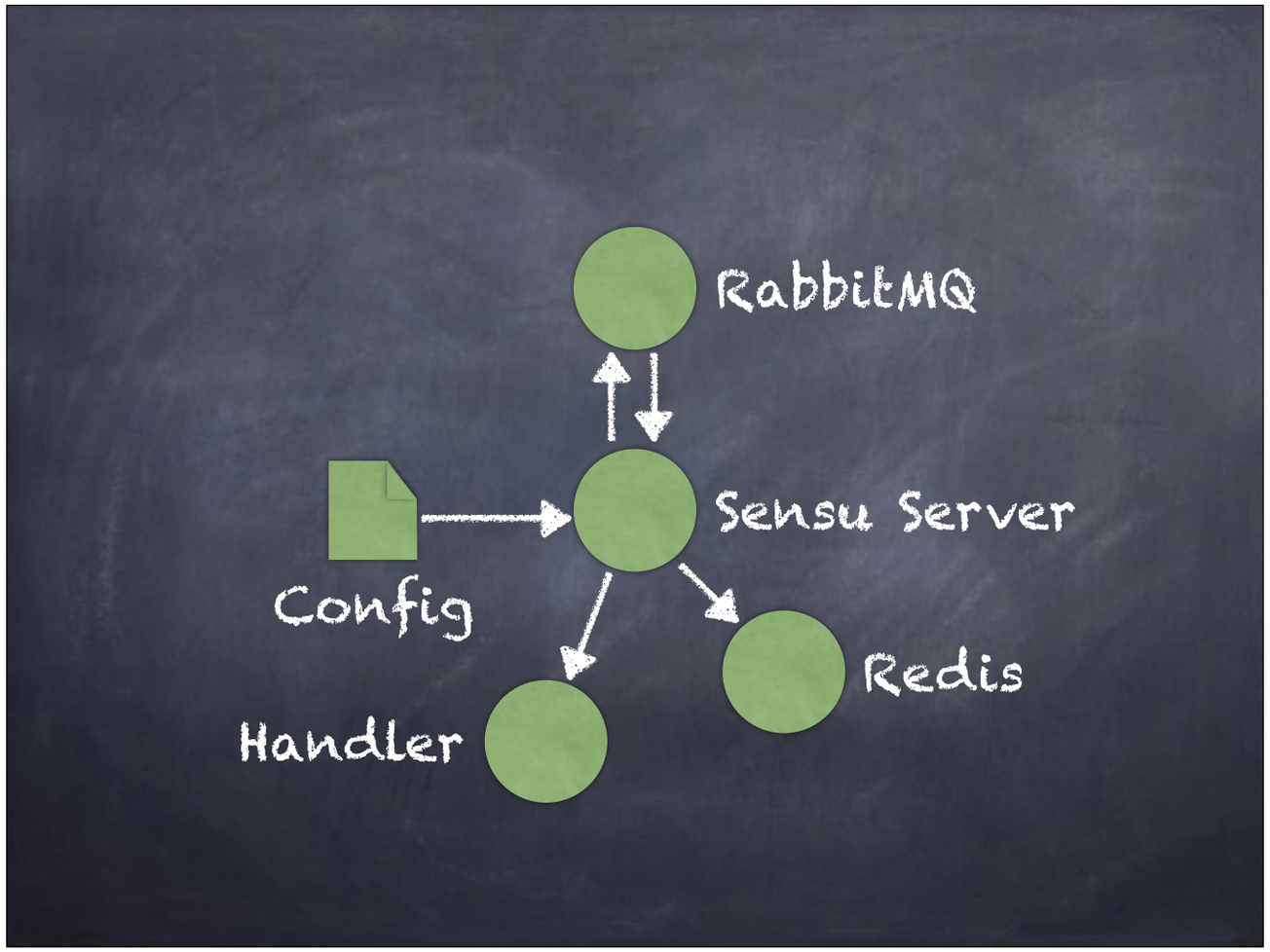
Sends a message over RabbitMQ to each client, telling it when to execute each check



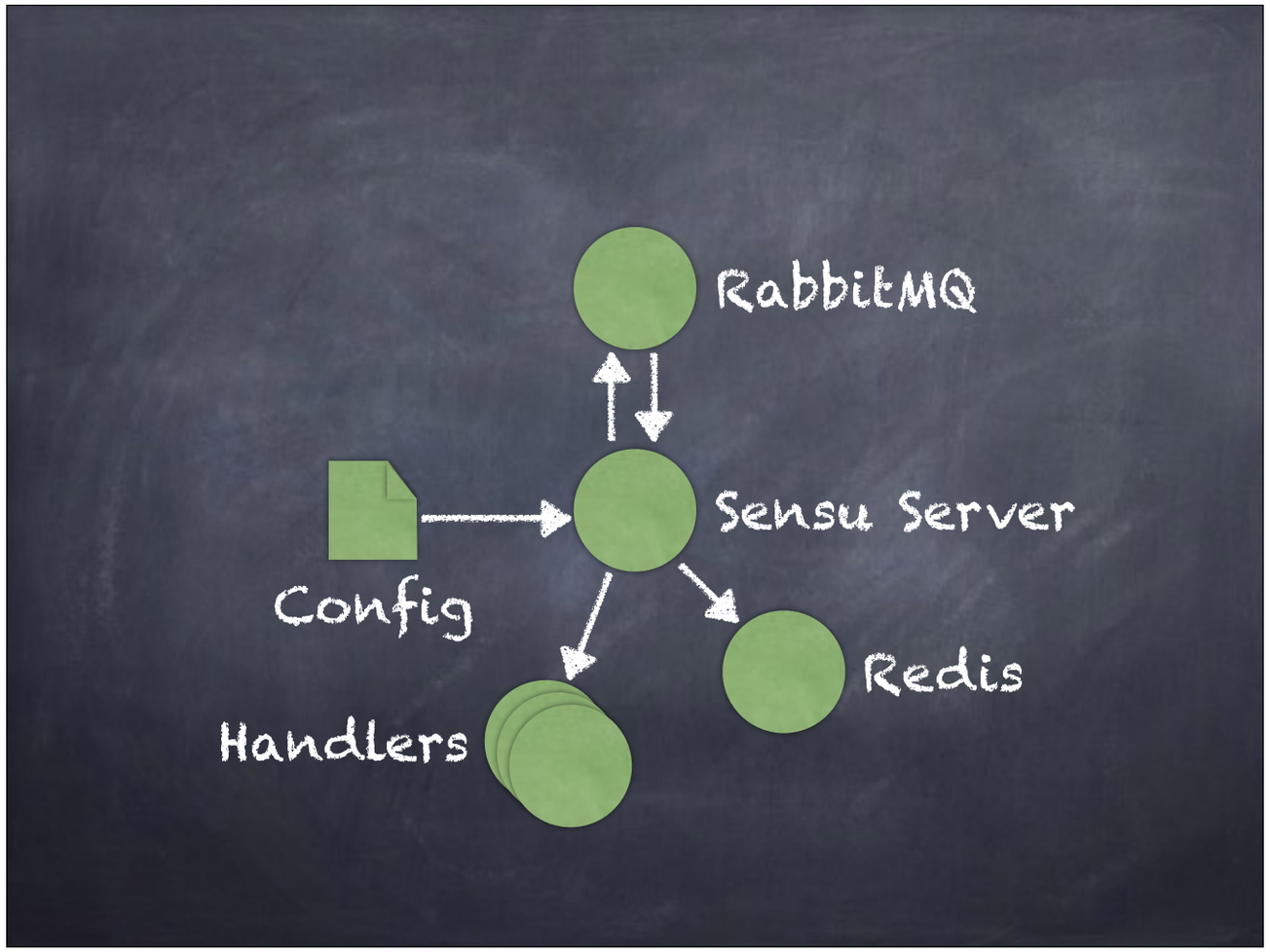
Receives events from each client over RabbitMQ



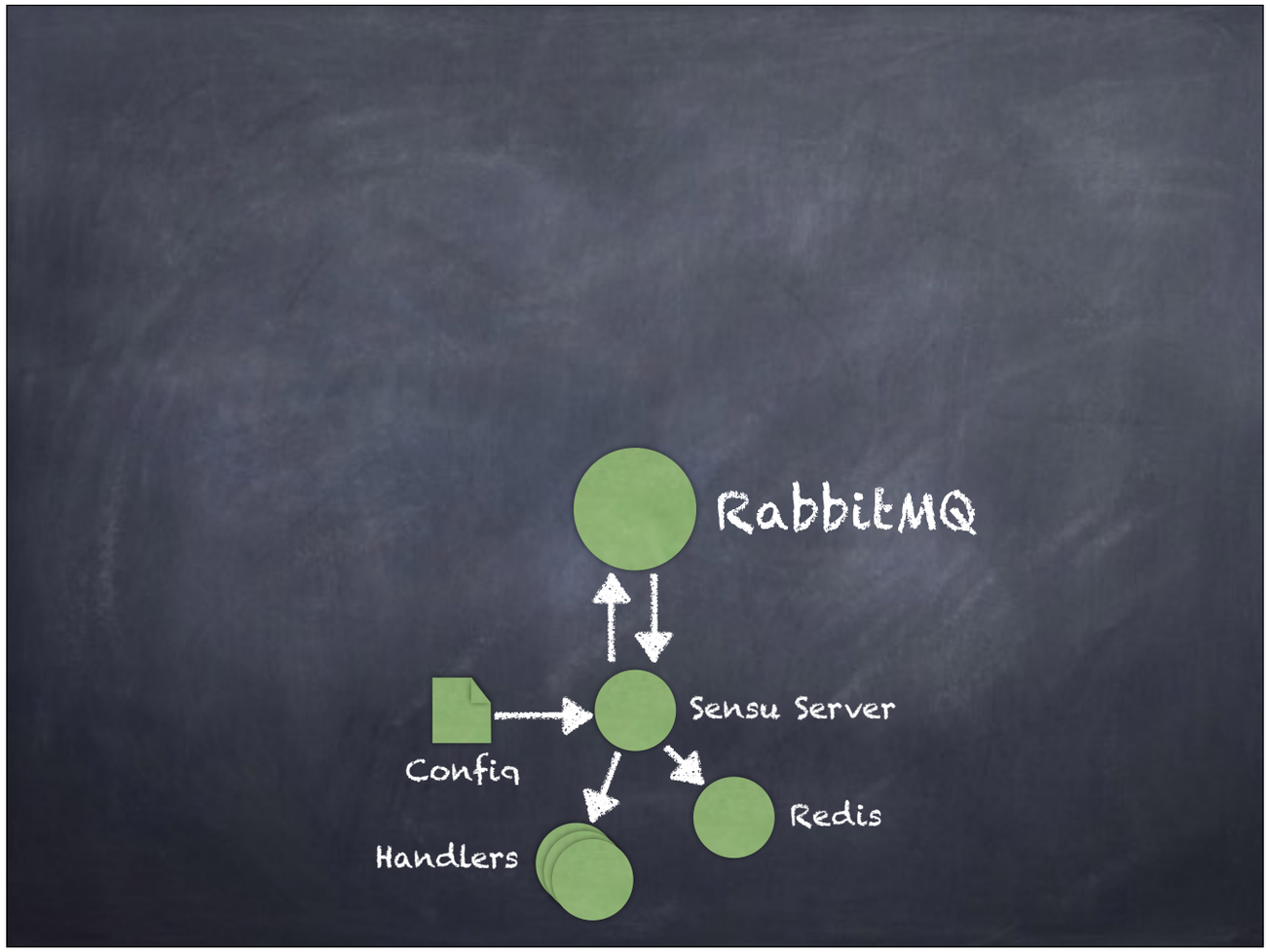
Stashes current state in Redis



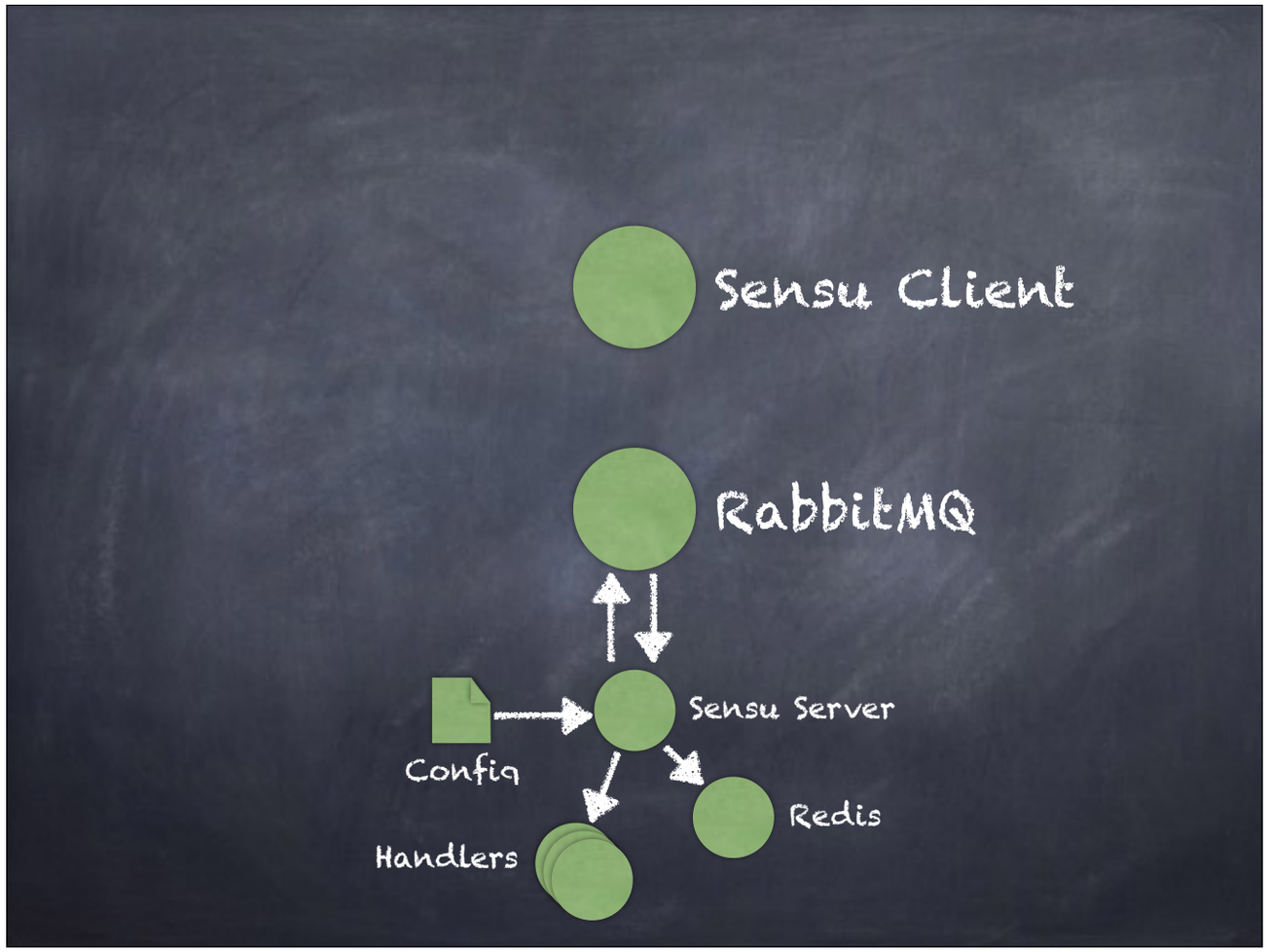
Executes handlers for each event



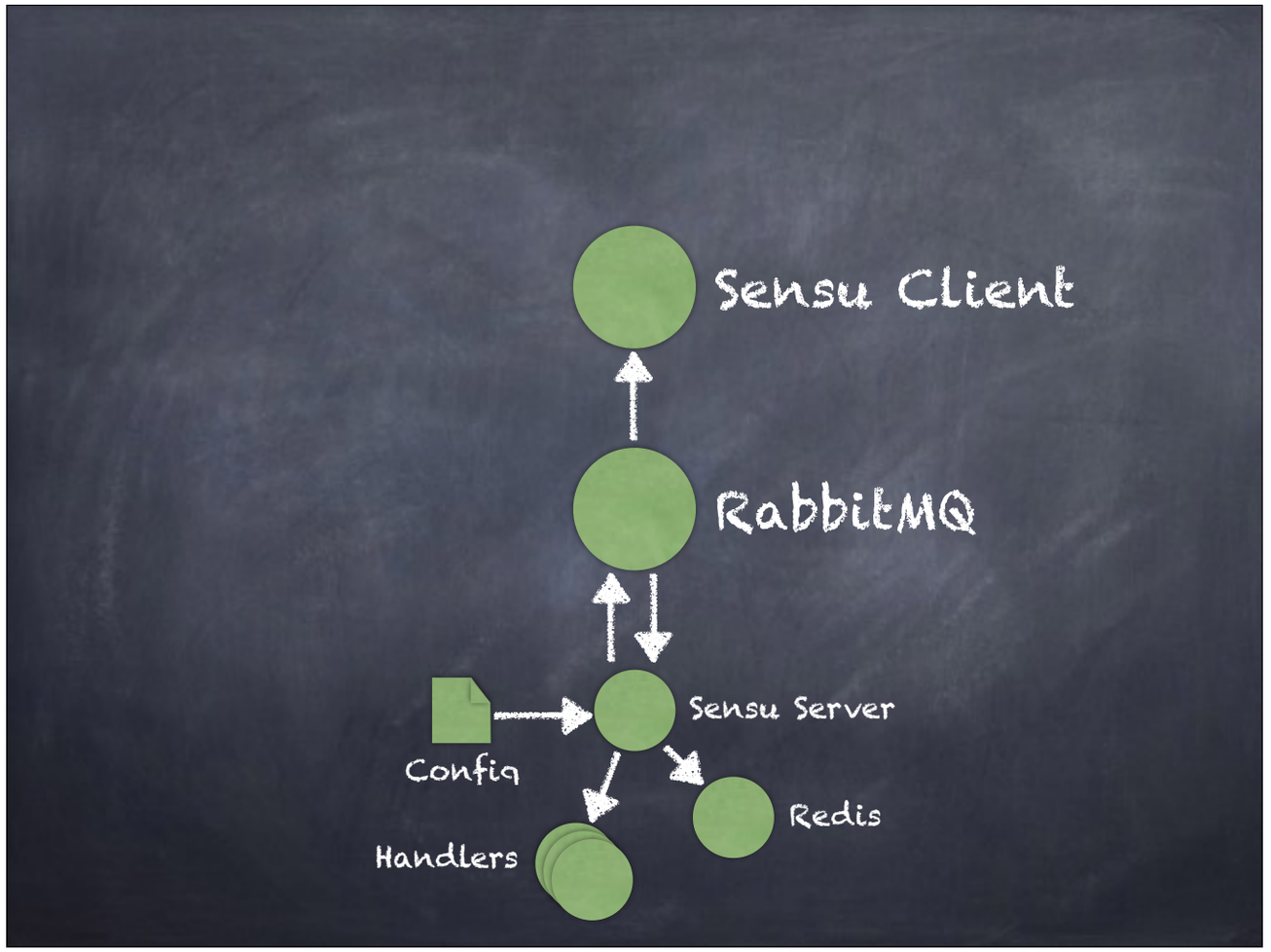
That’s the sensu server architecture



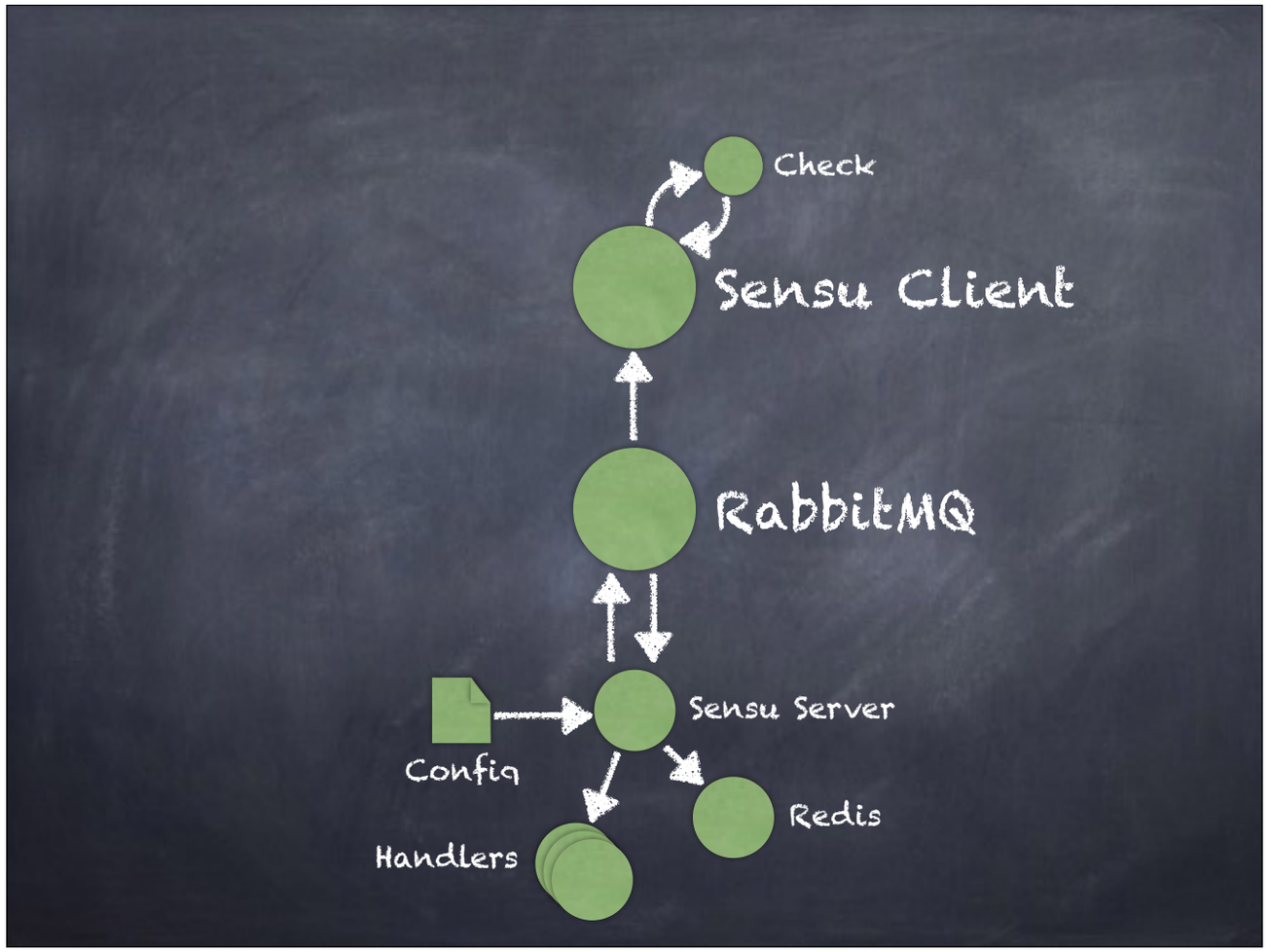
In order to actually check things, we need clients



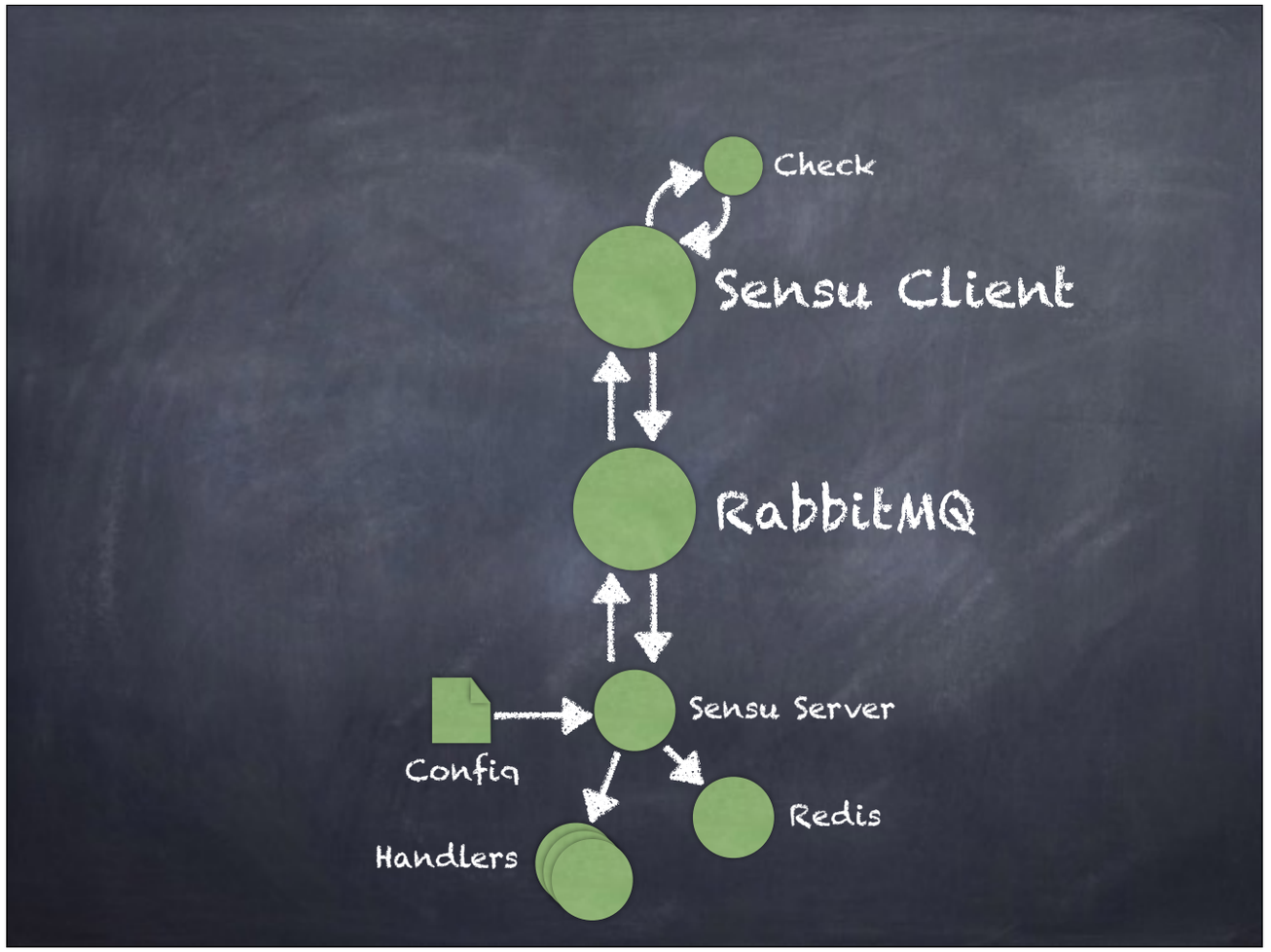
A check script orchestrator. You'll typically run one client on each machine you're monitoring, but you can also run "central" clients that execute remote checks.



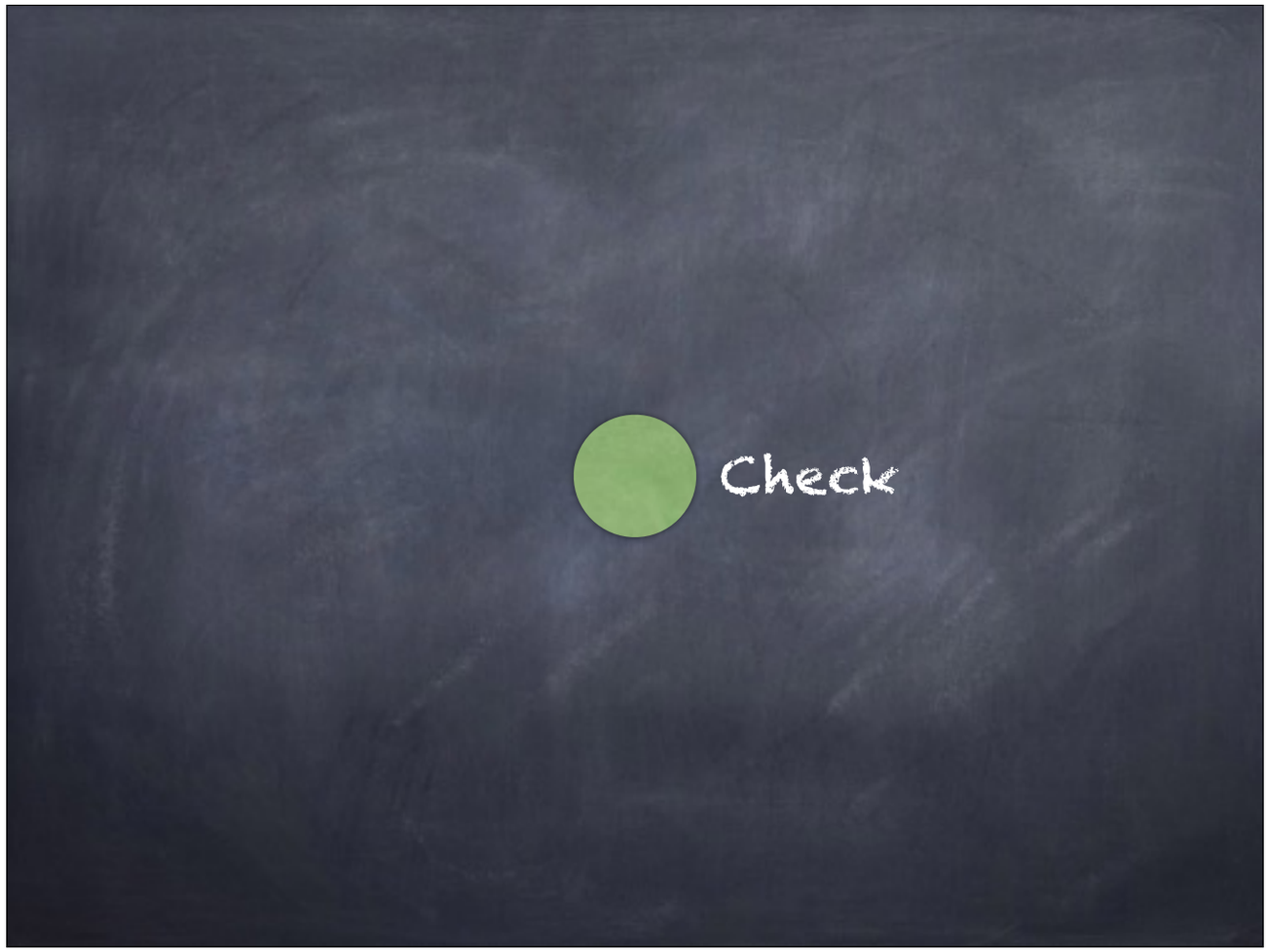
Receives checks to run over RabbitMQ



Runs the check

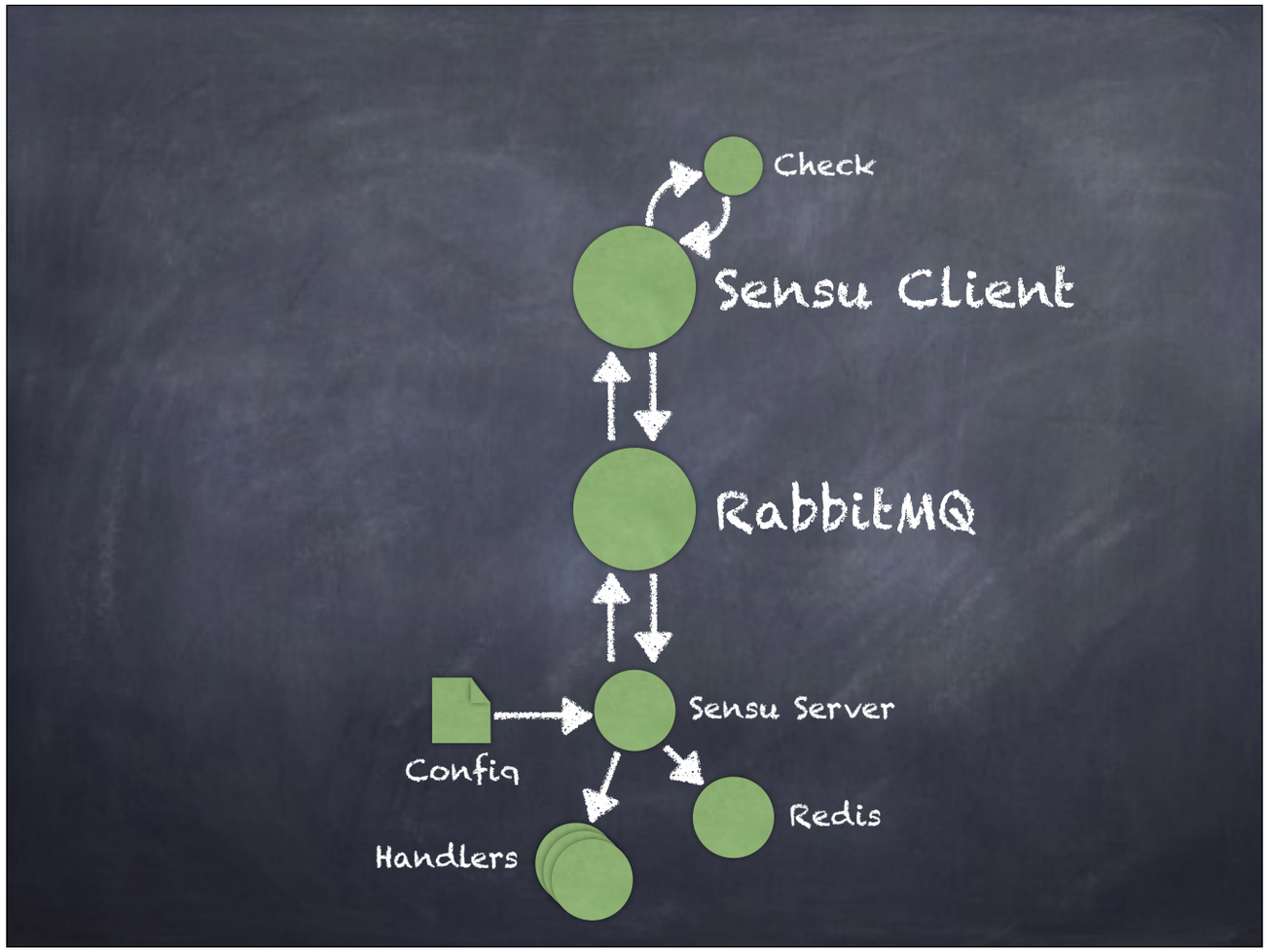


Sends script result, script output and metadata to RabbitMQ



A check is a bunch of metadata and a command line to execute.

* The command can be pretty much anything, as long as it exits with a 0 for success. This means you can use nagios plugins out of the box, and there are tons of those.
* Any output from the command is passed on to the Sensu server and can be processed by handlers (this allows for metrics in addition to status).
* The metadata defines the schedule to run the check, which handlers should process the events, which clients should run the check, etc.



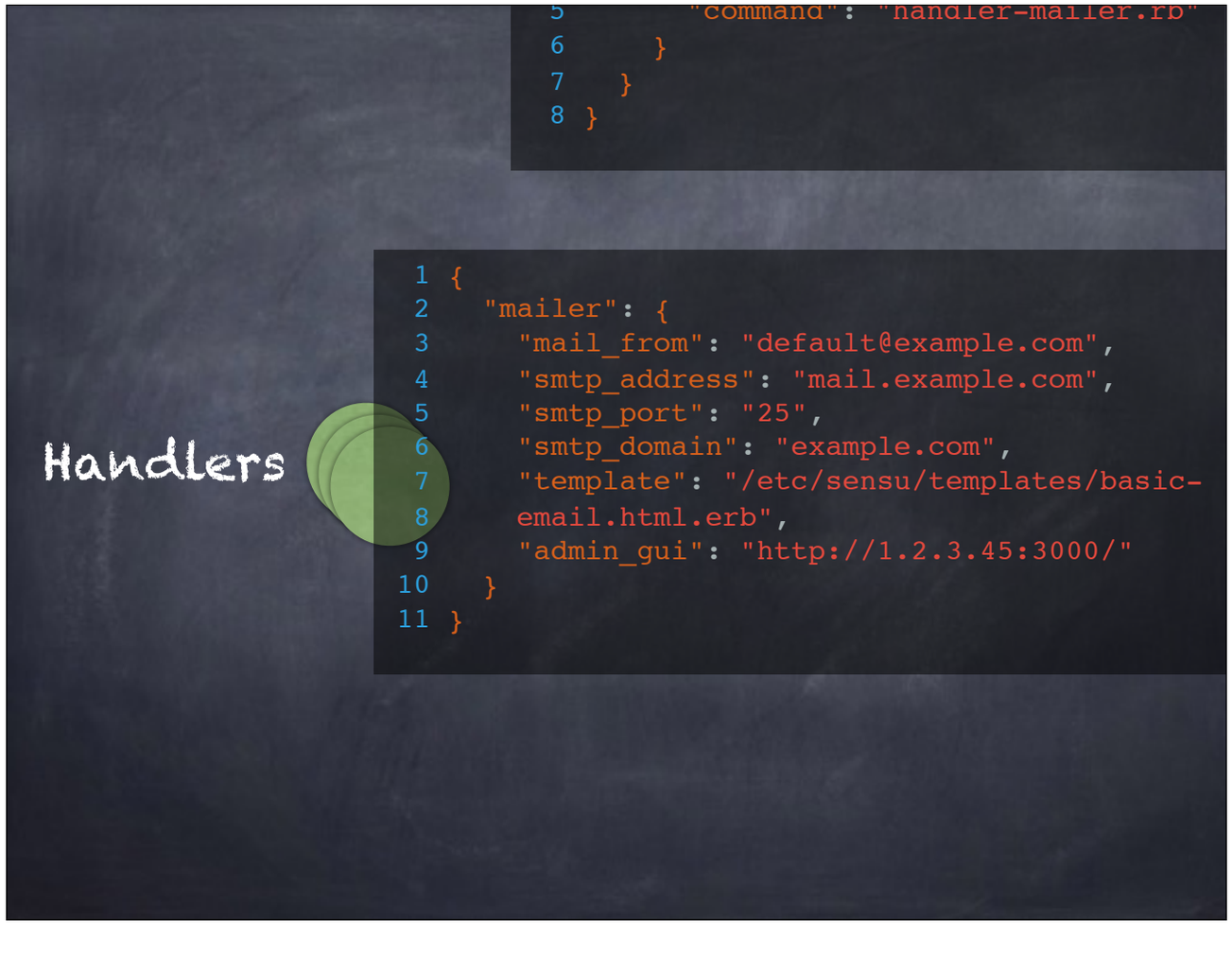
Now let’s look at Handlers

Handlers are where the magic happens (and were incredibly confusing for me). A handler is a process that receives event data (usually over STDIN or a socket) then "does stuff".

The most common handler just sends email. Since handlers are just scripts/processes, they can do just about anything (write metrics to ElasticSearch, restart servers, drive a robot, etc)

Configuration typically comes from three places:

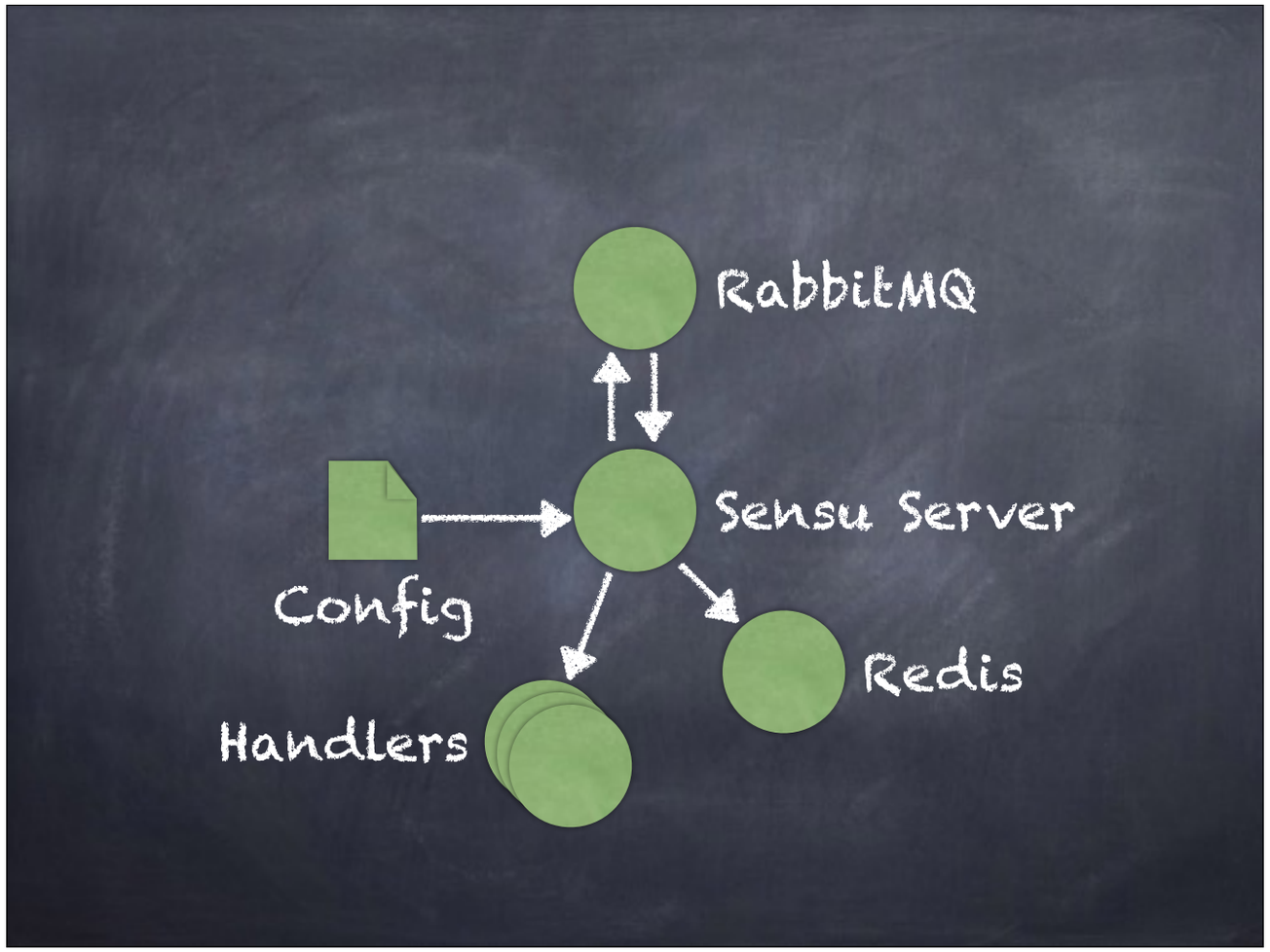
1. The handler definition provides the command to run, how to pass event data to the process, and a key to lookup additional configuration values



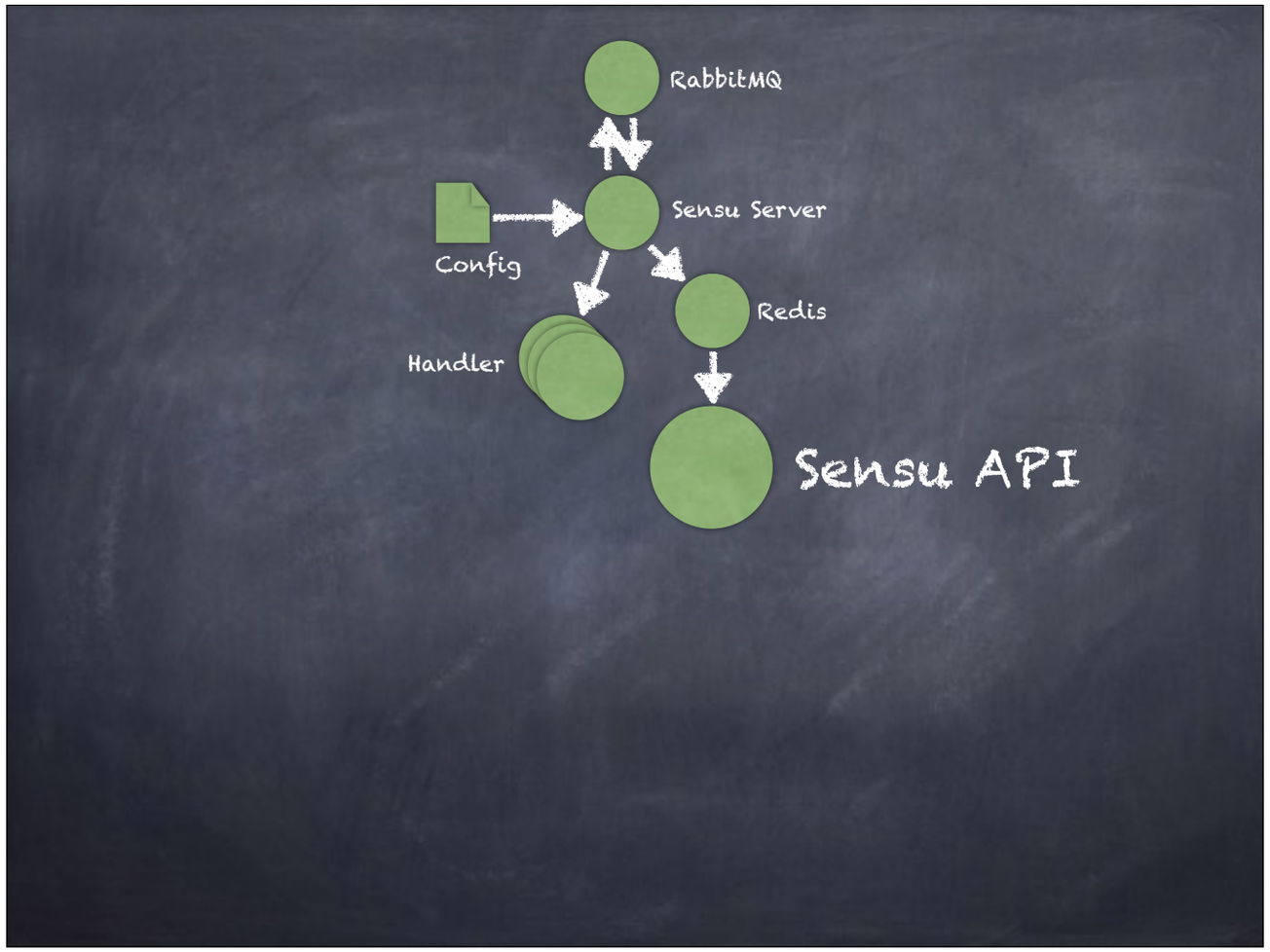
2. A custom top-level definition (that's configured in the handler definition) provides any "static" configuration values

3. The check definition provides check-specific configuration

* Each handler is responsible for *all* the event handling, including the basic stuff (ignoring dependent services when a parent is down, ignoring acknowledged alerts, flapping detection, etc)
* Sensu provides a ruby library you can extend that handles most of the basic stuff, but you're free to write your own if you want it to work differently

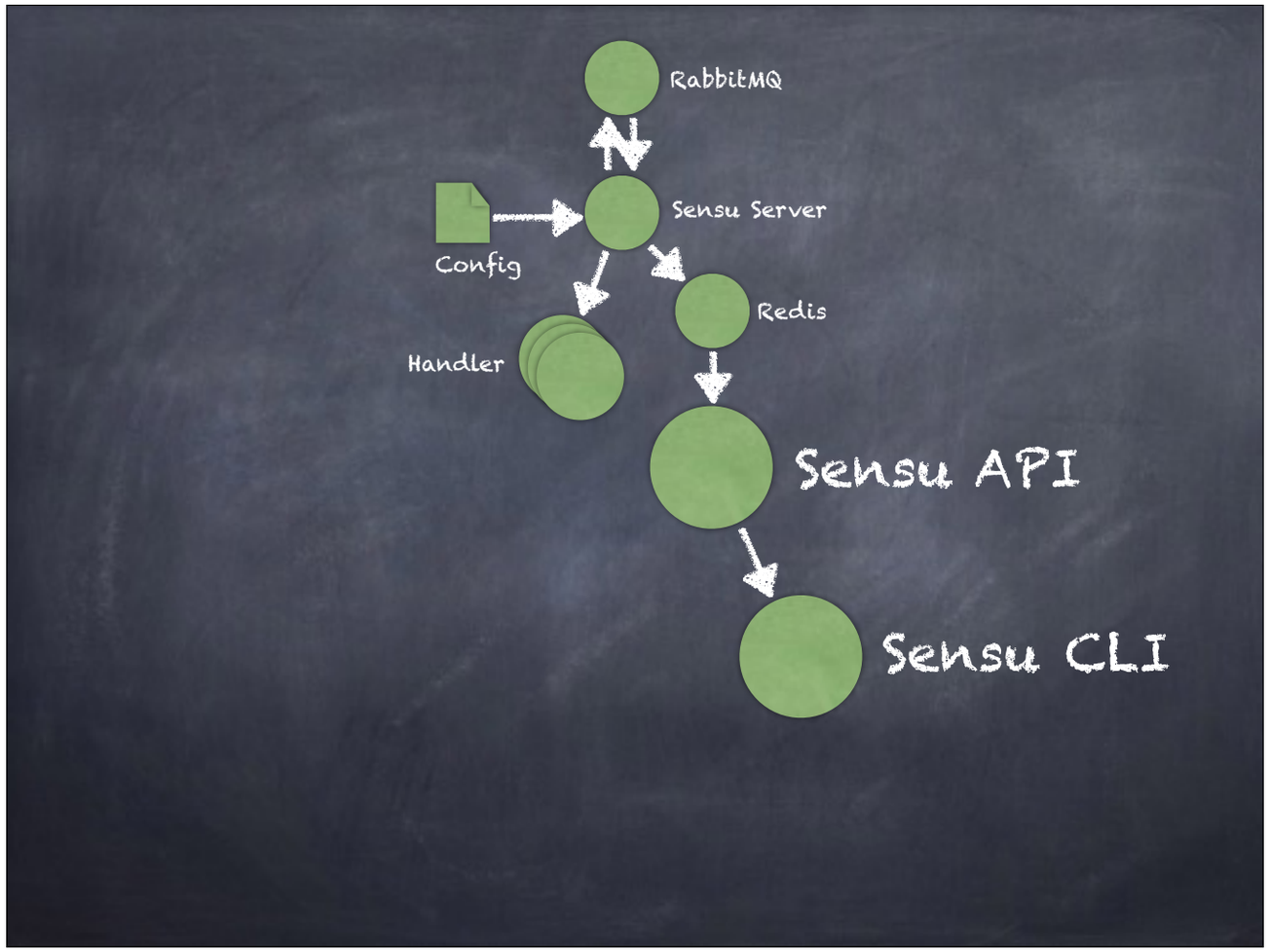


There are a couple more ancillary pieces to Sense

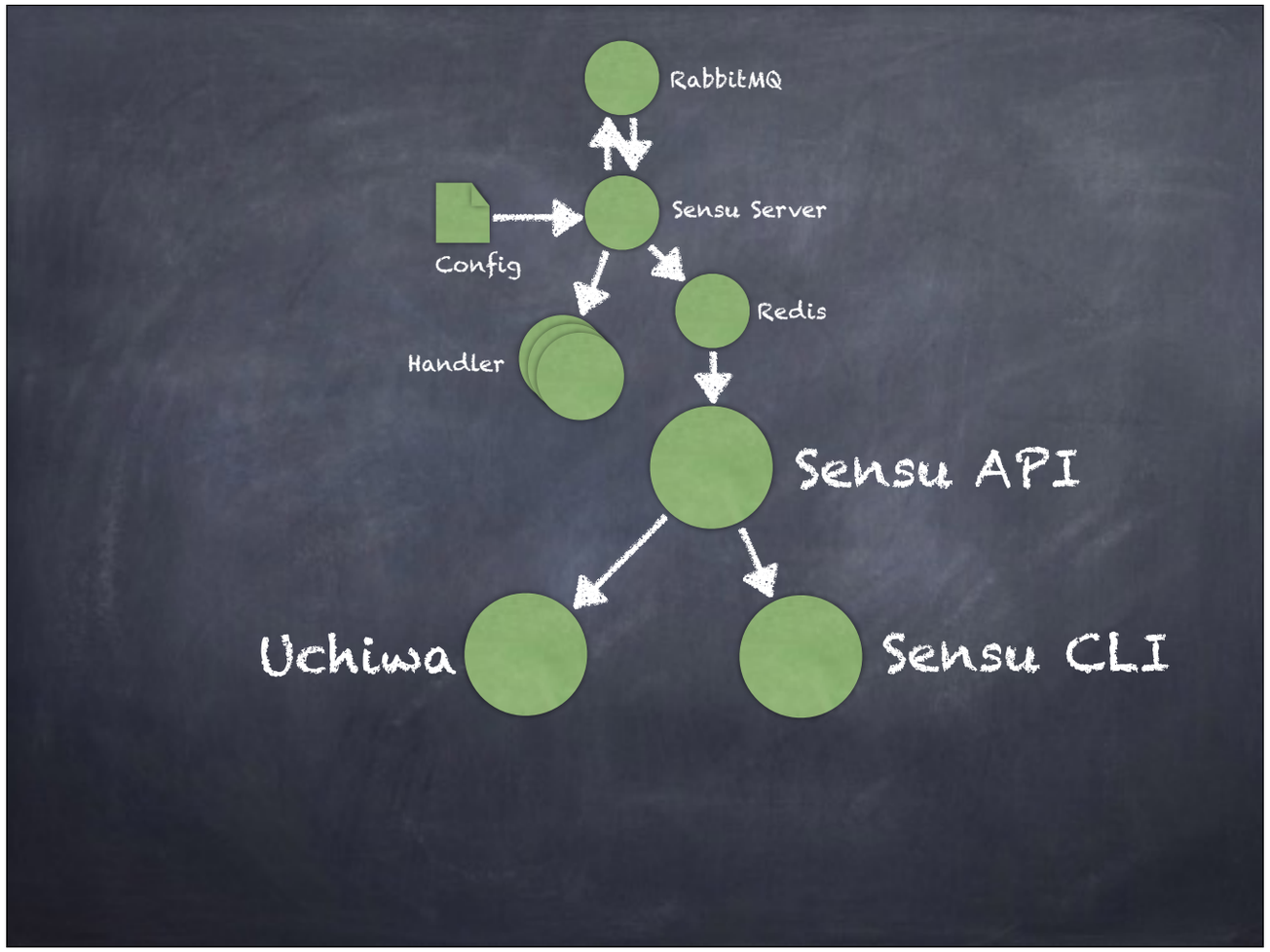


Sensu API - Exposes the data in Redis

* See which checks are defined
* See which checks currently have errors
* Store custom data (such as acknowledging an alert)

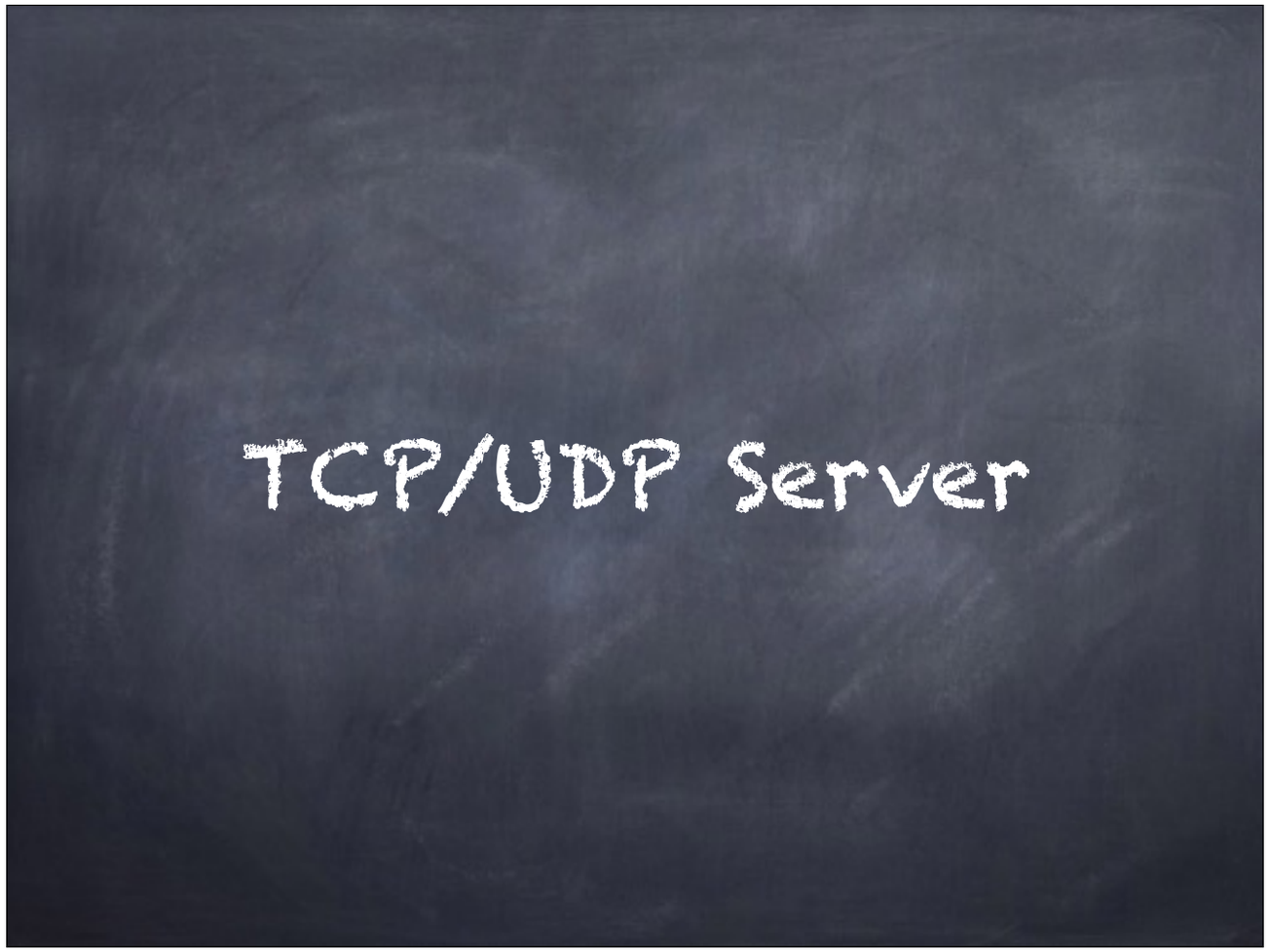


Sensu CLI - For using the API from a command line



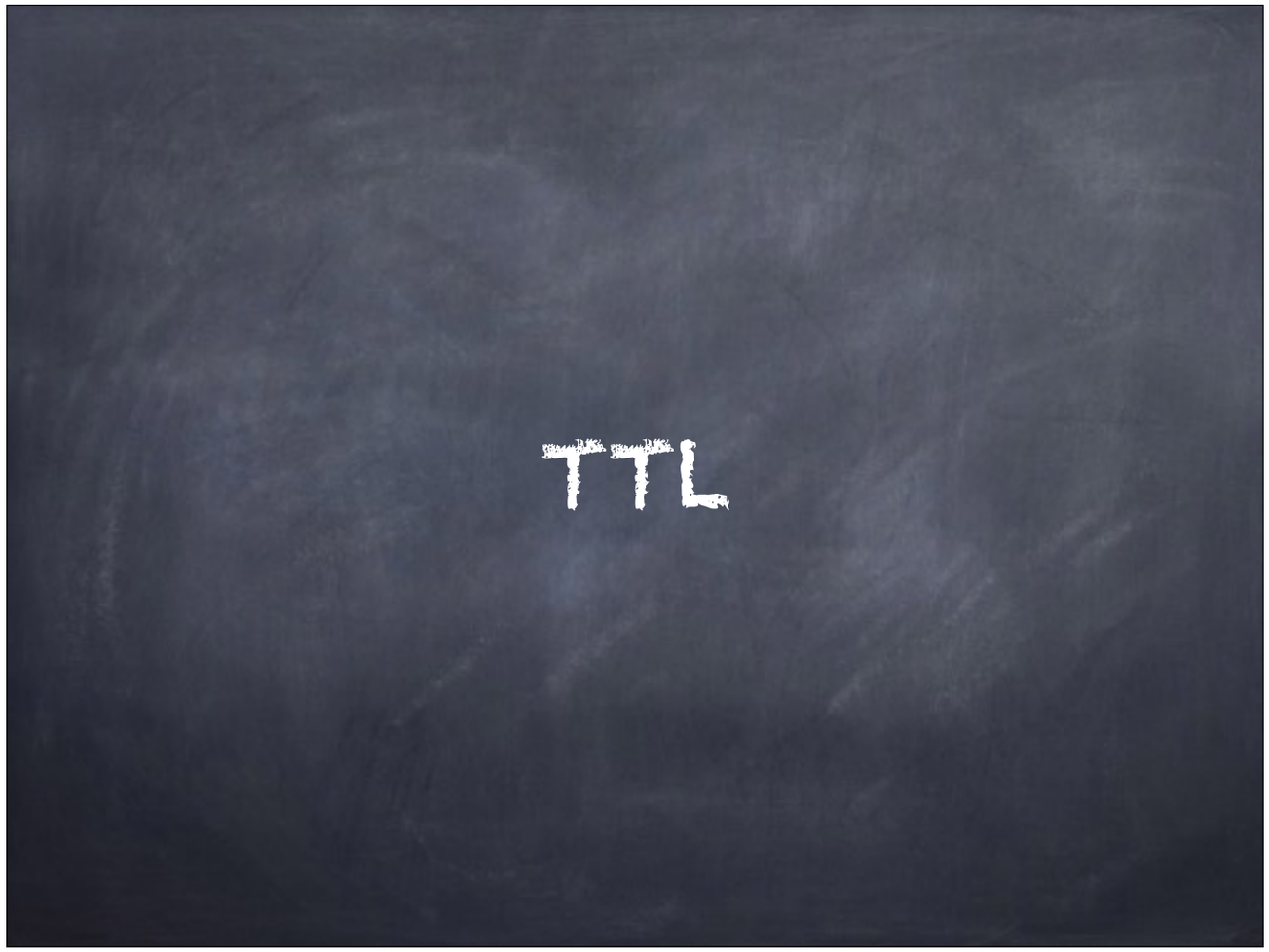
Uchiwa - Third-party web UI for the API

Cool Stuff



Each sensu client listens to a TCP and UDP socket for event data. You can send events directly from your own code and have them processed by Sensu's pipeline.

You don't have to predefine a check for the events you send this way, so it's very flexible.



Each event can have a TTL. The sensu server will fail the check if doesn't receive an event (any event) for this check before the TTL expires.

TTLs are super awesome when you combine them with the TCP Server. You can add a one-liner to the end of every cron job that sends an event to sensu, then sensu will alert if any cron job fails to run or finish on time.

This is just a handler plugin, but it's kinda awesome. Each check can define a sequence of escalating remediation steps to take (in addition to any other handlers, such as sending alerts). If you've got an annoying bug that crops up a lot, that's easy to resolve but hard to actually *fix*, this is perfect. Just have sensu run the command to resolve it for you when it happens.

Questions?

For reals this time