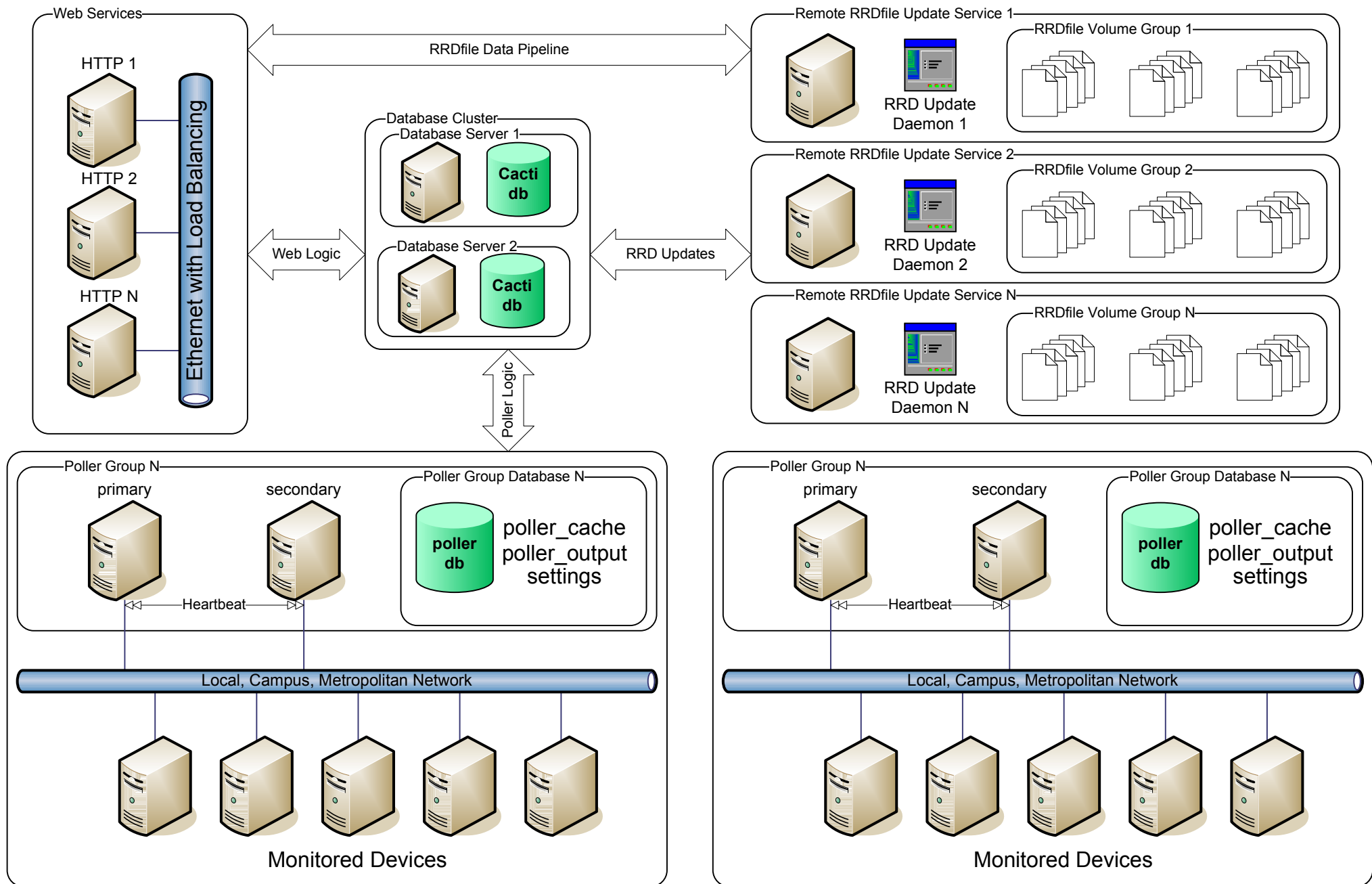


# Cacti Multiple Polling Interface

Copyright 2007, The Cacti Group

Sunday, December 16, 2007



# Cacti Multiple Polling Interface

Copyright 2007, The Cacti Group

Sunday, December 16, 2007

## Background

The Multiple Polling Infrastructure involves many components that have been created since the beginning of Cacti. It combines the idea of the Boost Plug-in, Structured RRDfile paths, and about 3 years of thinking on my part and feedback from you, the Cacti Community.

## Definitions

**Web Services:** A group of Web Servers (http/https) that work as a team. The Web Servers can be tied together with the load balancing of your choice.

*We could invent a load balancing scheme though. Your comments are welcome on this topic.*

**Database Services:** One to many Database Servers servicing Cacti Web Services, RRDfile Services, and Poller Groups. If it were many servers, they would have to be clustered in the format supported by the Database Vendor. *Right now, I am only concerned with MySQL, but could be swayed. I would need a good DBI programmer to step forward though in order to pull it off.*

**RRDfile Services:** A group of one to many daemons that provide real time RRDfile services. Almost identical to the Boost Server, these services include, but are not limited to the following methods: create, delete, update, graph, xport, dump, import, fetch, transfer, and tune. *If there are any other methods that we should be providing with this service, let me know.* Each service instance would accept requests on a known TCP port and in a multi-threaded fashion dispatch and respond to those requests. Security would be provided by an access list of known hosts maintained in the database. As noted by the diagram, each service could exist on a separate server.

**RRDfile Volume Groups:** This is similar to RRDfile structured paths in Cacti today. Each RRDtool Volume Group would have from one to many paths that it would be responsible for maintaining RRDfiles on. The alignment of RRDfiles in this scenario would typically lend it self to storage bandwidth limitations, and or memory limitations of the various servers. Each device would be associated with a RRDfile volume group. The movement and or distribution of RRDfiles in the various volume groups could be controlled by the special transfer and delete methods identified in the RRDfile services noted above. For new devices, each Volume Group would have an Active Path. In this case, if you had several paths available and one was at 90% and the other at 10% full, the second would be the default.

**Poller Groups:** A collection of servers that poll from one to many devices. In the early version of Cacti 0.8.7, then 0.9, the idea of poller resiliency was brought forward. The problem with poller resiliency was that if you lost a site, poller item failover would essentially fail. Regional polling could have been accomplished, but without no failover. Therefore, in the new design, each poller would be assigned to a poller group. For each poller group there would be a master and from one to many secondary masters. Each poller group would have a database instance that would contain a copy of the poller\_items from the master database, a poller\_output table and a settings table. The concept of disabling and enabling a remote poller or poller group would be supported. In addition to promotion and/or demotion of various pollers to and from master status. *I don't know of a scenario where a poller would be moved between poller groups, but would accept comments on that.* Also, in this design, device (aka poller\_item) resiliency would not be supported. However, a device could be moved between poller groups.