

PANDORA FMS

ARCHITECTURE





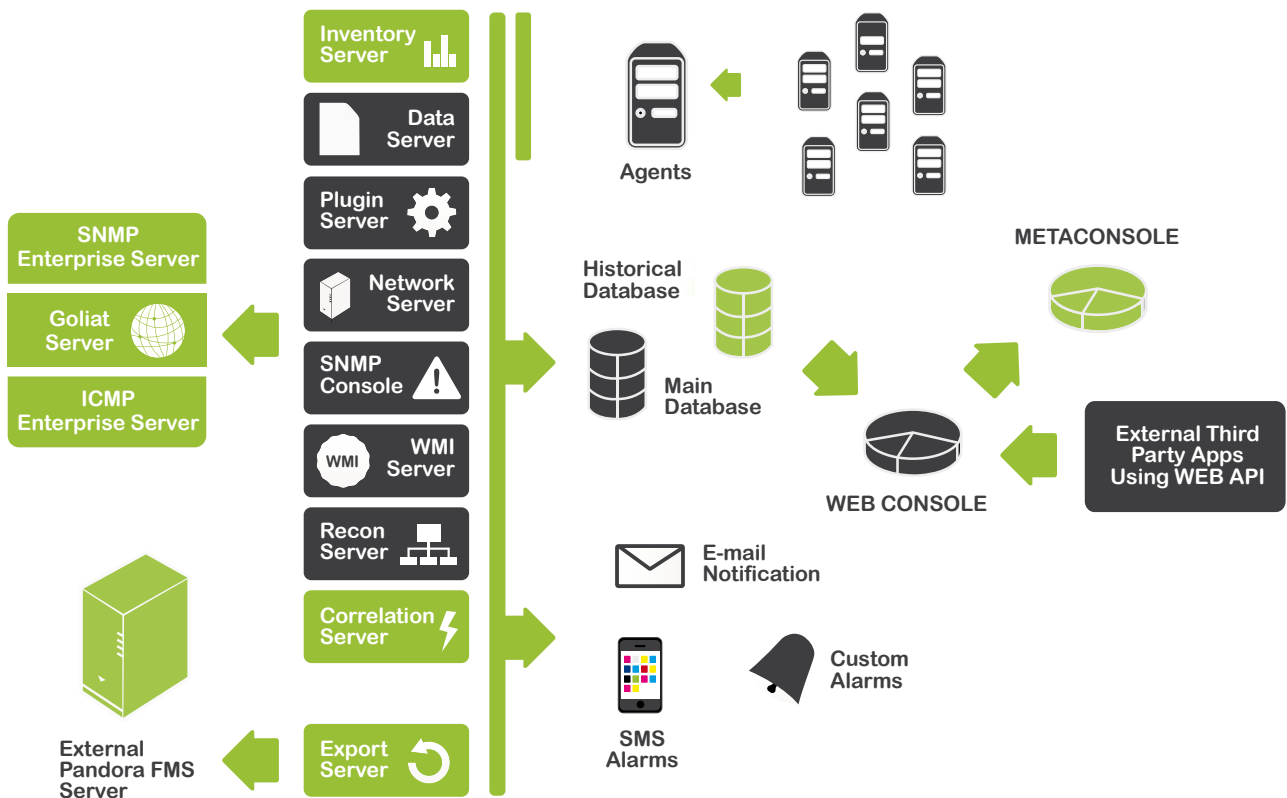
1. PANDORA FMS ARCHITECTURE

Pandora FMS has a very flexible and versatile design and allow to work in several different ways. Pandora FMS works with remote monitoring and with local agent-based monitoring, and of course, allow to combine both of them.

Pandora FMS is developed in several languages: C++,

Perl and VBS in the agents, Perl in the server and PHP5/Javascript in the Web Console.

Pandora FMS has a modular design, based on several specific servers for each kind of check. All its components are redundant and could work on HA Active/Active mode.

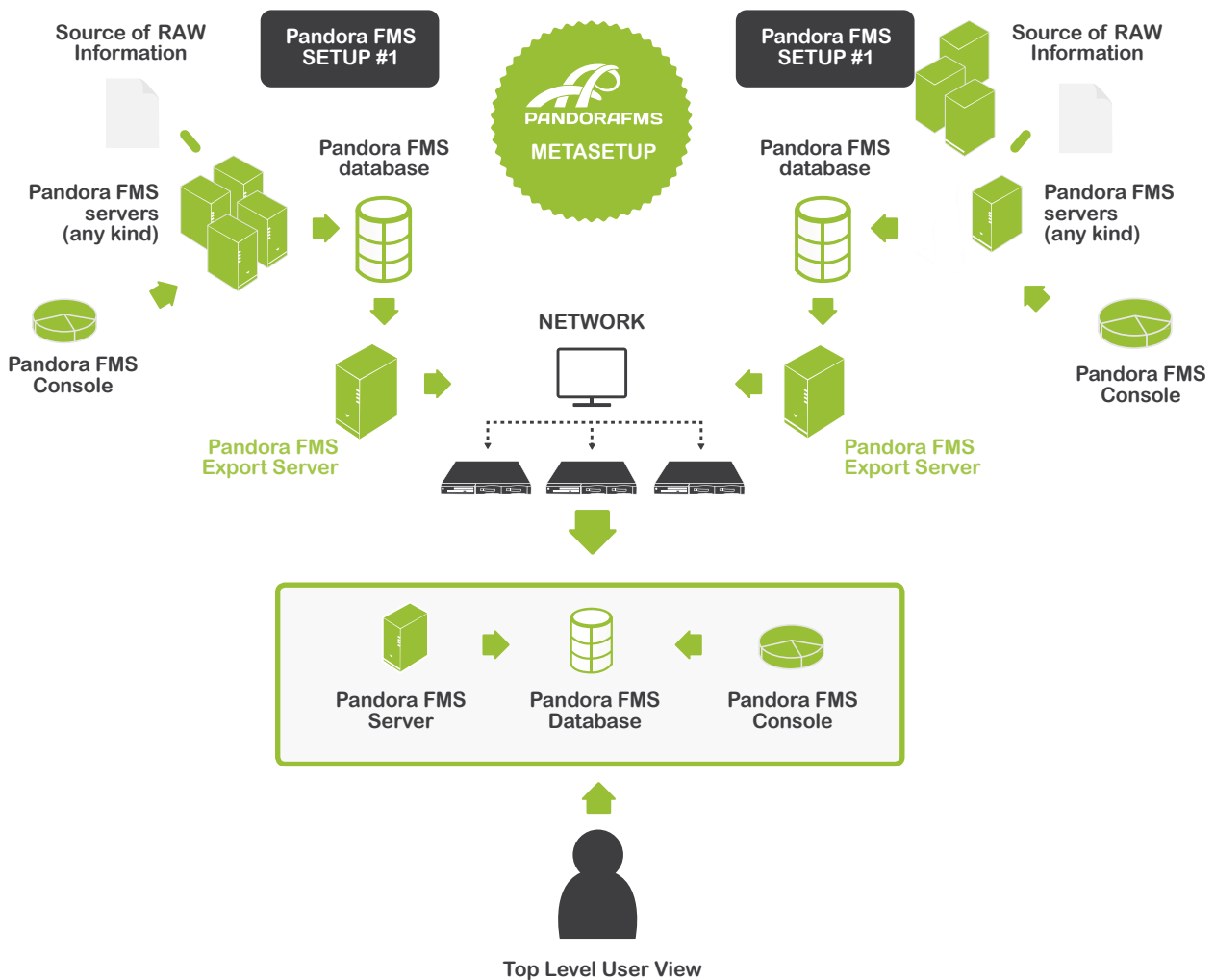




1.2 Export server

The main idea consist on creating a hierarchical structure that distribute the information from down to top, being the top point the more global one, that collect only an extract of basic information of Pandora FMS installations at the lowest level, instead of collecting a bigger number of information, and that allows to the Pandora FMS highest installation to have a «filter» vision and more information density.

The server that exports is hierarchically bellow the server that gets this information. In a different sketch of the filter vision, you could use this technology to do a reply of all data reported by a server, though it could affect to the server performance in an important way.



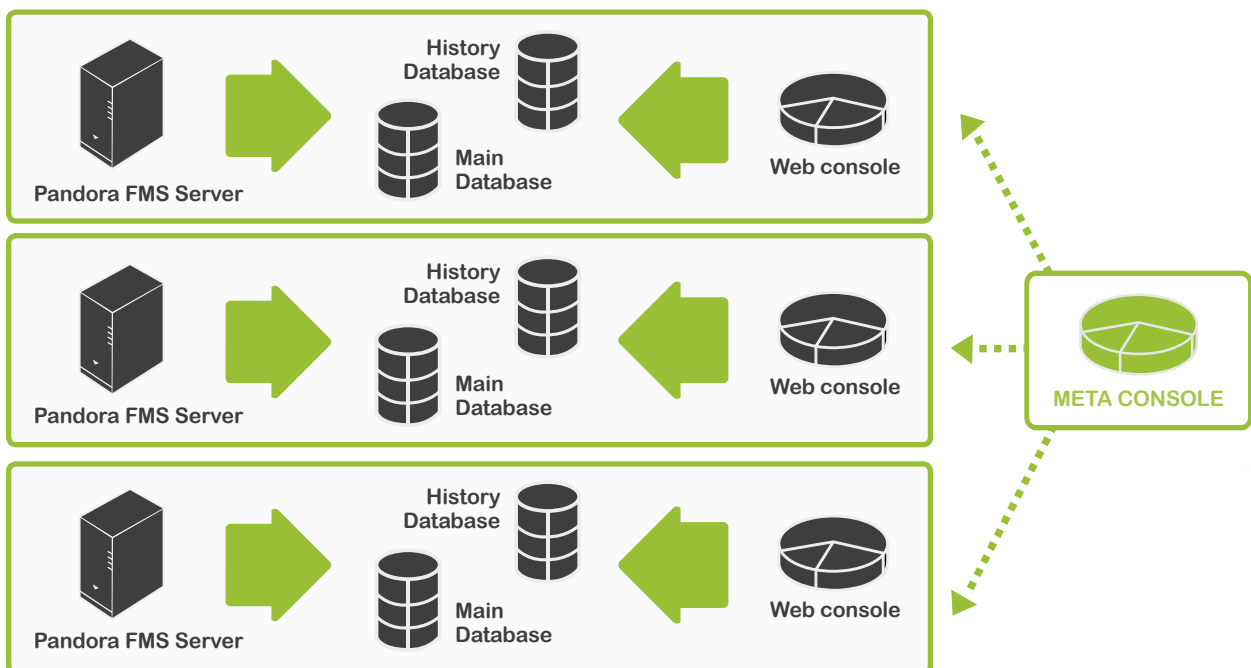


1.2 Metaconsole

Pandora FMS Enterprise version, thanks to the Metaconsole, implements a way for distributing the monitoring between different Pandora FMS servers that are physically independent. Each server has its own database, console and server. Besides, it also has its own agents, alerts and reports. And what is more: it has users, groups and policies.

The Metaconsole doesn't process information, it only "reads" the information from its original source, that is: from the Pandora's server, where the information is kept, only that the meta console can search an agent in all pandoras, and show the data views of each agent of each Pandora FMS, simply by linking automatically the views of "Local" data of each Pandora FMS.

This is possible through the delegated authentication (through hash) that implements Pandora FMS from version 2.1, that allows that an user previously authenticated in the meta console does not have to authenticate in one of the Pandoras associated to the metaconsole. This way, it doesn't exist a theoretical limit of maximum number of systems to monitor so we can keep adding Pandora FMS's servers in a linear way to get the scalability that we want, as you can see in the following example, where, if we suppose that each server processes 1.200 agents, we can see that we can easily exceed the 6.000 agents monitored adding 5 servers:





1.2 Tentacle proxy (Drone agents)

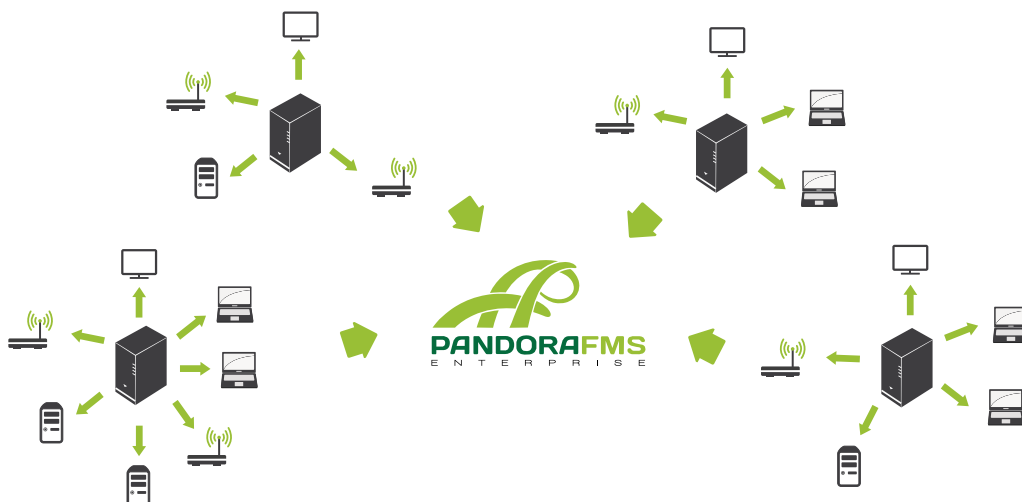
The new Tentacle version supports proxy usage (HTTP/Connect mode), so agents can contact directly with the server using an intermediate standard proxy.

You also can use a new tool called Tentacle Proxy Server, which as its name says, is used to centralize all communication between Pandora FMS and the agents, allowing the file management and remote configuration for policy based monitoring.

2. HIGH CAPACITY AND PERFORMANCE

Pandora FMS has been designed to work on enterprise systems: that means, systems able to grow, and grow to infinite. Our engineers have estimated an average of 2.500 agents per server, with 25 modules each, running 5 minutes interval checks. Using the metaconsole and the Export server you can expand these numbers using more servers, or try to allocate more agents in a single server.

We have customers with really big setups, using Pandora FMS in a different way. We have a customer using 6.000 agents in a four server setup, using the metaconsole, and another customer using a single setup with 160.000 modules in a single server.



info@artica.es

www.pandorafms.com

Ártica Soluciones Tecnológicas

Gran Vía 62 St.

28015, Madrid, Spain.

(+34) 91 559 72 22