

PANDORA FMS

ARCHITECTURE



V2 Version 2 | 二版
www.version-2.com





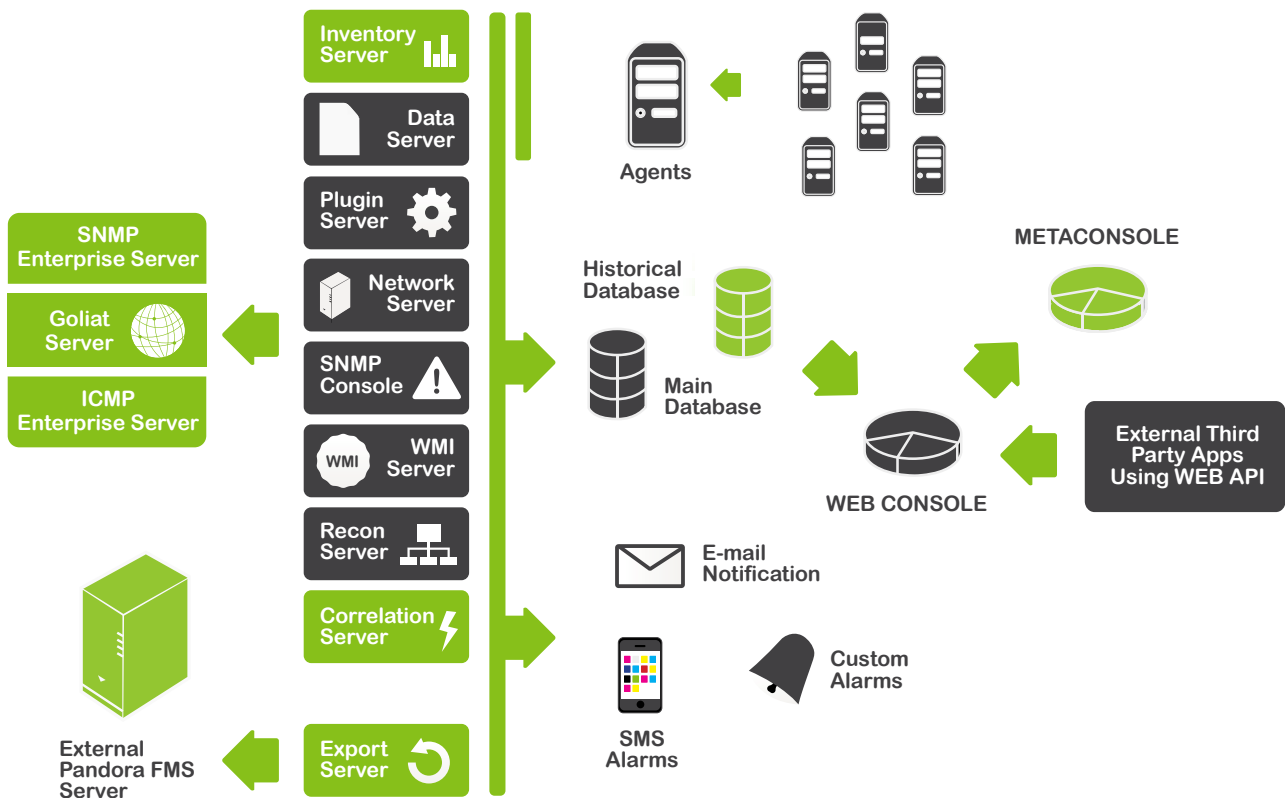
1. PANDORA FMS ARCHITECTURE

Pandora FMS is a very versatile and modular tool, which allows you to work in various ways. In a summarized manner, we could say that Pandora FMS works both with remote and agent-based monitoring, and can obviously combine both.

Pandora FMS is developed using a combination of

script languages: C++ and Perl for agents, Perl for servers, and PHP and JavaScript for the Web console.

Pandora FMS has a modular design based on a variety of specific subservers for each check type. All of its components are redundant and can function in HA Active/Active mode.

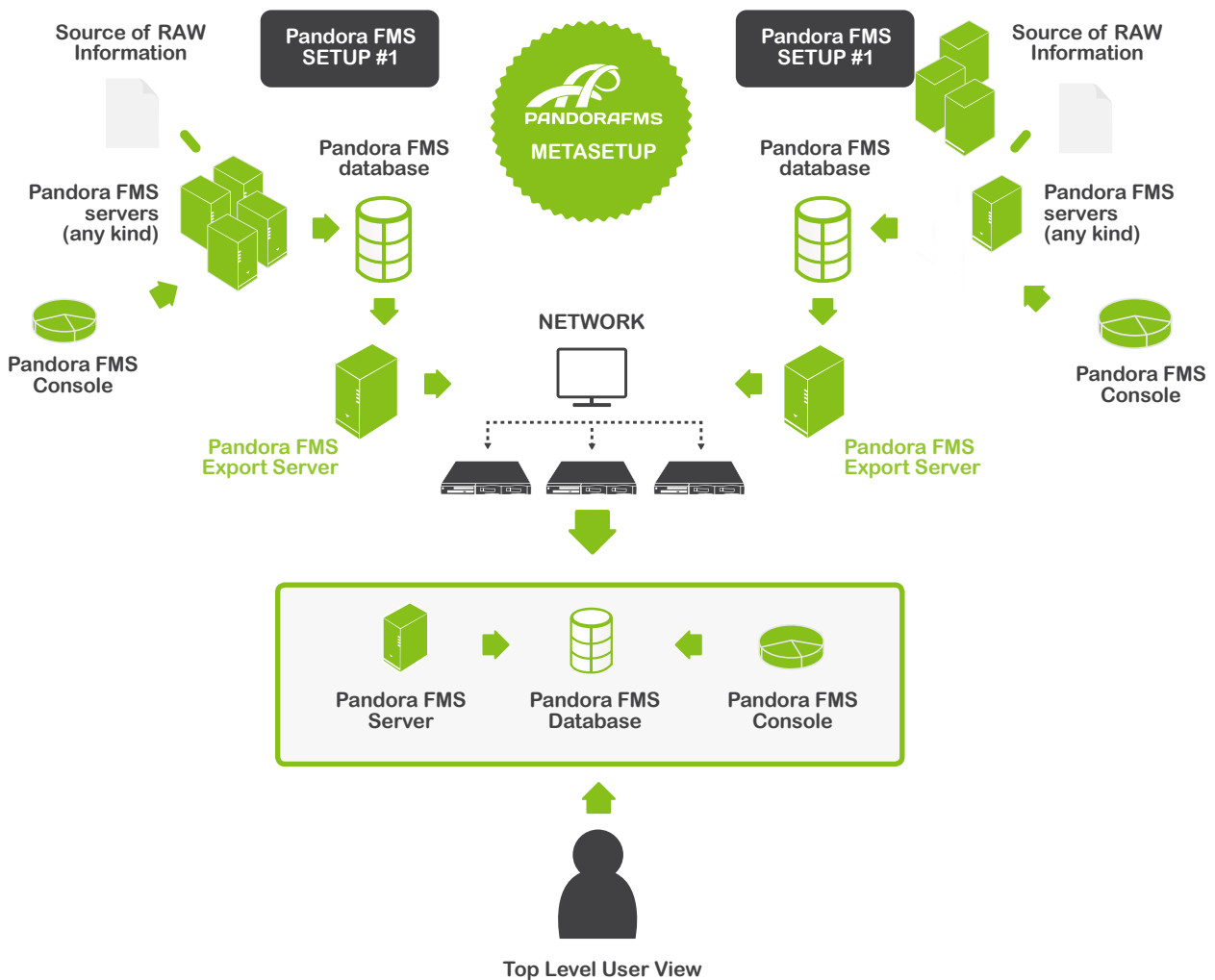




1.2 Export server

The export server allows certain data to escalate onto other instances of Pandora FMS. This means that data maybe transferred from one implantation (or site) to others on the Pandora FMS network. This way the other implantations receive server data as if it was a "copy" of the original data. All of the data that is meant to be exported can be marked, module by module, within the server sending the data.

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.



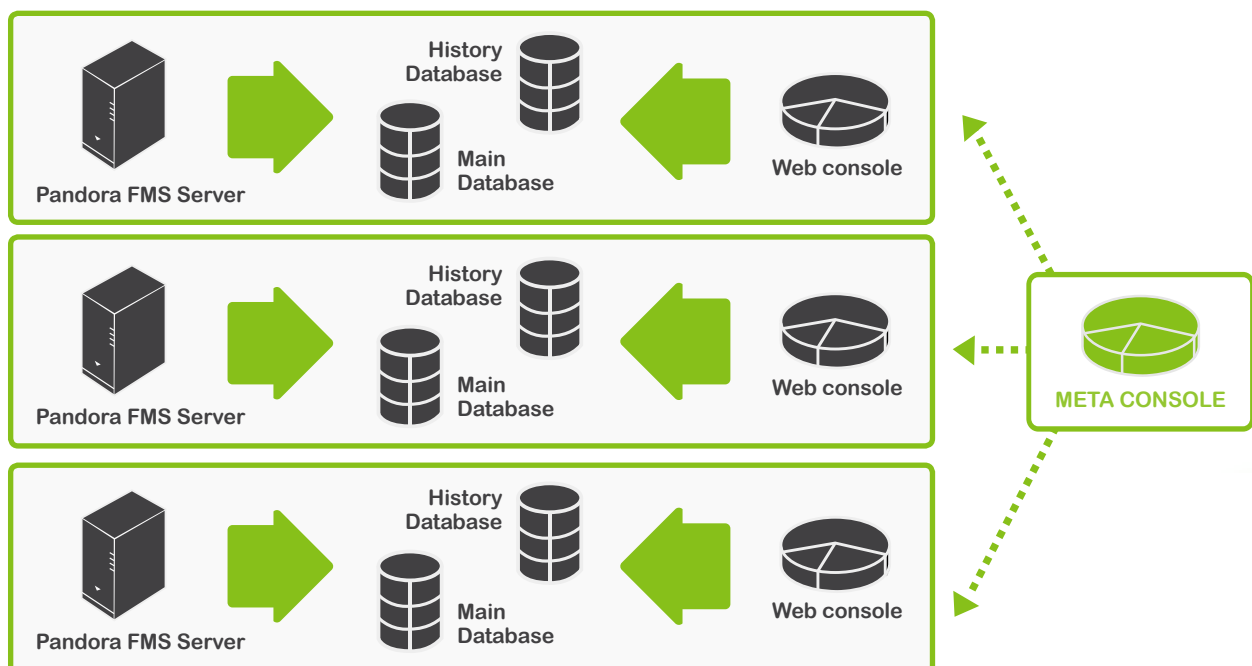


1.2 Metaconsole

Pandora FMS enterprise, thanks to its metaconsole, implements a method to distribute monitoring between different Pandora FMS servers that otherwise are physically independent. Each server has its own database, console and server. Of course, it also has its own agents, alerts, reports, and even users, groups and policies.

The metaconsole doesn't process information, it limits itself to "reading" the retrieved information from the original source; that is from Pandora's server, where the information is stored. The outstanding value of this is that the metaconsole can search for an isolated agent out of all the Pandora systems and can show the data overviews for each Pandora FMS agent. Simply by automatically linking the "local" data overview from each Pandora FMS.

This is possible using the delegate authentication (through hash) that Pandora FMS uses since its version 2.1, which allows users previously authenticated by the metaconsole to not have to authenticate themselves in any Pandora associated to the metaconsole. This way, there's no theoretical limit to the maximum number of monitored machines, since we can always continue to add Pandora servers linearly to achieve the scalability we desire. In the following example we suppose that each server processes 1,200 agents, which means we could easily surpass 6.000 monitored agents by adding five more servers.





1.2 Tentacle proxy (Drone agents)

The new version of Tentacle supports proxy use (in HTTP/connect mode) in a way that agents can connect directly to the server using a standard proxy address. Under the same principle we created a component called Tentacle Proxy Server, which

allows the use of an intermediate element that centralizes all communication with the destined server, apart from managing file collections (v3.2) and configurations.

2. HIGH PERFORMANCE CAPACITY

Pandora FMS has been designed to work in entrepreneurial environments. This means it's designed to work with groups of systems that can continue to grow infinitely. Our engineers have estimated an average of 2,500 agents per server, with up to 25 modules per agent, performing tests every 5 minutes. Using the metaconsole and the Export Server, these numbers can be expanded using more servers or by assigning more agents

per single server (which requires fine tuning). We have clients with really large environments, where Pandora FMS is used in an array of varied tasks. For example, we have a client with 6.000 agents and a 4 server setup with a single MetaConsole, but at the same time another client has created a monitoring environment based on a single server with 160.000 modules.

