SPOC: Spacecraft Operations Centre

Michalis Charalampakis[(1)] Jens Schmidt [(2)], Tim-Christin Hanschen [(3)], Matthias Tausche[(4)], Dr. Gilberto Arantes[(5)]

(1)
Manfred-Fuchs-Platz 2-4
28359 Bremen
GERMANY
Email: michalis.charalampakis@ohb.de

(2)
Manfred-Fuchs-Platz 2-4
28359 Bremen
GERMANY
Email: Jens.Schmidt@ohb.de

(3)
Manfred-Fuchs-Platz 2-4
28359 Bremen
GERMANY
Email: Tim-Christian.Hanschen@ohb.de

(4)
Manfred-Fuchs-Platz 2-4
28359 Bremen
GERMANY
Email: Matthias.Tausche@ohb.de

(5)
Manfred-Fuchs-Platz 2-4
28359 Bremen
GERMANY
Email: Gilberto.Arantes@ohb.de

INTRODUCTION

The idea of SPOC (Spacecraft Operations Centre) was driven by the occurrence of several problems. This was the starting point of several use-cases and room for improvement.

What problems ignite our engine?

"During a test the hard disk which holds the test results runs full of data so that a running test has to be ended and repeated after cleaning up the hard disk."

"It was not possible to send a command to a SCOE. The reason was a time skew between the SCOE and CCS server."

"It was not possible to start a CCS session on a client. The reason was that the TERMA CCS5[1] server software was not running."

These are just some examples. In all cases we had the impression that the problems could be avoided or solved much faster if we have a detailed overview about the test setups. We are flying blind.

So we need windows to see what is going on!

ABSTRACT
SPOC is the central point of administration and monitoring which will be used to get an overview of the health and status of all test setups. SPOC is fundamental of a central and cross-mission EGSE support.

All information which are required in case of troubleshooting is available in one central location. Due to a permanent monitoring we will be in the position to notice the possibility to run into a problem before the problem really occurs. This will increase the reliable and availability of the infrastructure.

WHY DO WE NEED MONITORING SYSTEMS?
AIT EGSE is a very complex environment consists from several different devices types and operating systems. The Project tight schedule, do not allow systems and services unplanned downtimes. Collecting manually information of the health status is a time-consuming and complex task.
Introducing a monitoring system in our environment allow us to have a proactive instead of reactive IT infrastructure support approach. A good monitoring system helps to increase productivity and this is shown through several aspects such:

- Fully automatic monitoring of the whole IT infrastructure,
- Real time network health overview that allow as to detect problems before they become incidents,
- Historical performance monitoring over time,
- Notification can be send by email or other communication channels,
- Users can concentrate on their tasks and not for troubleshooting,
- Costs reduction both in time and money.

WHY DO WE NEED REMOTE ACCESS?
AIT EGSE production and test environments are usually located in integration halls and laboratories - remote access allows us to access all systems from a central point, perform tasks faster **and provide better overall support experience.**

Secure remote access benefits include:

- Better work productivity and flexibility,
- Better end-to-end security,

- Minimizes response times,
- Improved business continuity.

SPOC IN GENERAL

The spacecraft operations centre is generally divided into two main parts. The first one is a central monitoring for all test setups. With the monitoring it is possible to monitor all devices and services. This can be done on a top level so on the EGSE or CCS LAN level, but also down to the SCOE LAN level. It is possible to monitor basic IT related services like a filesystem or the traffic on a switch. But it is also possible to monitor e.g. the time skew of NTP or if a required service is running on the CORTEX or IMBU. The monitoring is based on Icinga[2] which is an open-source monitoring application which is widely used for IT infrastructures but it is very flexible so that it can be adapted also to our needs. Icinga consist of two parts. A backend which connects to the devices which should be monitored. The backend contains plugins for the checks. It is also possible to create own plugins for special purposes. The second part is the frontend. There are several web-based frontends available which can be used to represent the check results in the best way. We are using one monitoring server for each test setup. The webpages can be reached via a secured (HTTPS) connection from each PC. The monitoring is configured in a way that at default nobody has access. Authentication is done via Active Directory so we are using the company authentication mechanism which means that there is no need for the users to get dedicated user accounts.

The second main function of SPOC is the ability to provide a remote access for all systems. The access is needed for administrational tasks and troubleshooting. Project networks are separated from the office network so the challenge was to provide a secure remote access without connecting the whole project network to the office network. The solution was again an open-source application… "GUACAMOLE"[3]. Guacamole is a clientless remote desktop gateway. What does this mean? This application makes it possible to provide a remote access (RDP, VNC, SSH) via HTML5 with a web browser. There are no plugins or add-ons required. The use is very easy and user-friendly and the performance is awesome. The guacamole server is placed between the project network and the office network. Connections to the server are also encrypted (HTTPS). Access is implemented in two steps. The first step is an authentication to the Guacamole server which also due via Active Directory. So again the user do not need a dedicated account for this. The default is no access to any system. In Guacamole we have a separate user database where we can explicitly grant access to a device.

As a single point of access we created a webpage which is the entry page to all web services, so all monitoring and remote access gateways servers are reachable from here. The users' just need to access this page and do not need to remind all servers.

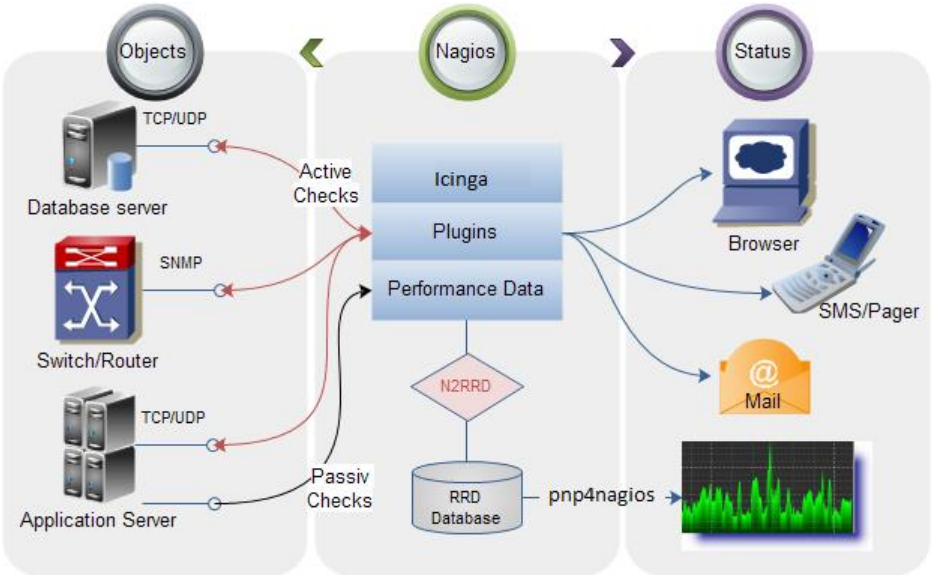
HOW IS THE MONITORING IMPLEMENTED

Icinga is a free and open source network monitoring application that allows us to check both private services of a system, such as CPU load, RAM/Disk usage etc., as well as public services like web, SSH, mail, and so on. Forked from Nagios, all configurations, plugins and add-ons working with Nagios can also be used. In addition, Icinga delivers improved database connectors and user-friendly web interface. Icinga's components perform the following tasks :

- Monitoring
    - Monitoring of network services (SMTP, POP3, HTTP, NTP, ping, etc.),
    - Monitoring of host resources (CPU load, disk usage, etc.),
    - Monitoring of any equipment connected on the network (switches, routers, temperature/humidity sensors, Time Servers etc.),
    - Simple plugin design that allows us to easily develop our own service checks.

- Alerting
    - Notification of contact persons when service or host problems occur via email, sms, or other user-defined methods such as pager, browser pop-up, etc.

- Visualization & Reporting
    - Interfaces for visualization of host and service status, network maps, reports, logs, etc.,
    - Report repository with varying access levels and automated report generation and distribution,
    - Performance graphing via add-ons such as PNP4Nagios.



Monitoring Server
Icinga monitoring Server can run on a Virtual Machine or a Physical Server meets the following minimum hardware requirements:

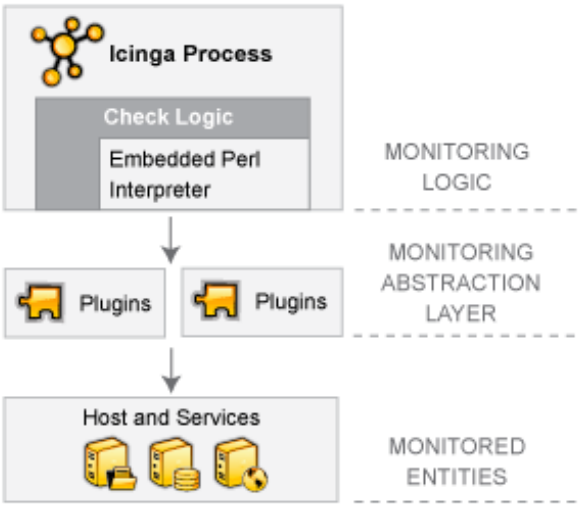
- ≥2 CPU Cores
- ≥ 20GB HDD
- ≥ 2GB RAM
- ≥ 1 Network Interface

Icinga monitoring Server can run on most Linux Distributions and needs the following software packages:

- Icinga
- Nagios Plugins
- SNMP
- NTP
- Postscript
- Apache
- PHP
- mk_livestatus

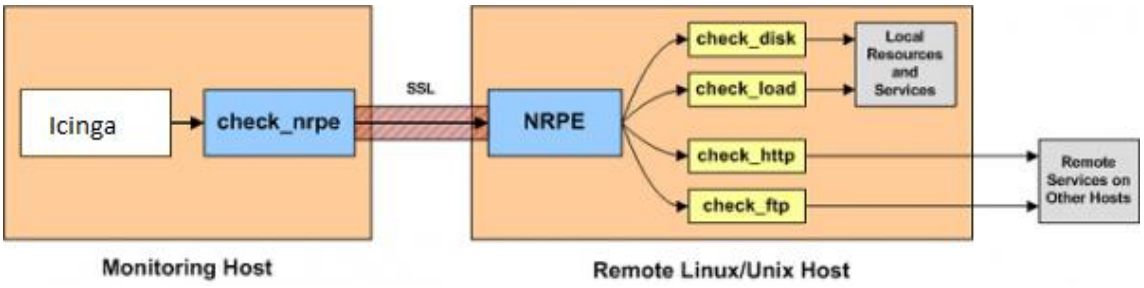
Nagios Plugins

Icinga does not include any mechanisms for checking the status of components. Plugins are compiled executables or scripts that can be run from a command line to check the status of a host or a service. Icinga uses the results from plugins to determine the current status. There are roughly 50 official plugins and thousands more that are developed from community members.



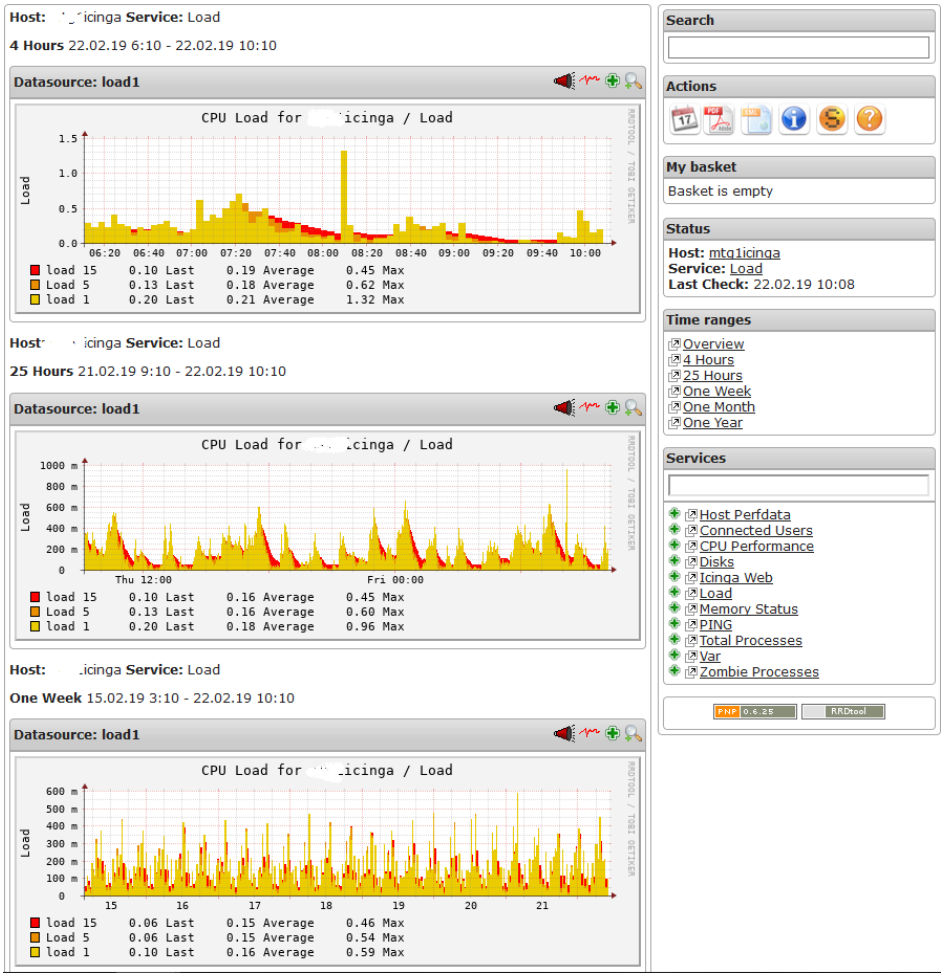
Nagios Remote Plugin Executor (NRPE)

Nagios Remote Plugin Executor (NRPE) consists from the plugin resides on the Icinga Server and a daemon run on the remote system - this allow us to execute Nagios Plugins on remote systems and monitor the local resources such as CPU, Memory, Disk usage etc.



Pnp4nagios

Pnp4nagios[4] is a performance data visualization tool, which analyses data provided by the plugins and stores them into RRDs (round-robin databases). That means that after some time the oldest data will be dropped at the “end” and it will be replaced by new values “at the beginning”. Using the defaults values allows us to store data with a resolution of one minute for the last two days, five minutes for ten days, 30 minutes for 90 days and 6 hours resolution for four years.

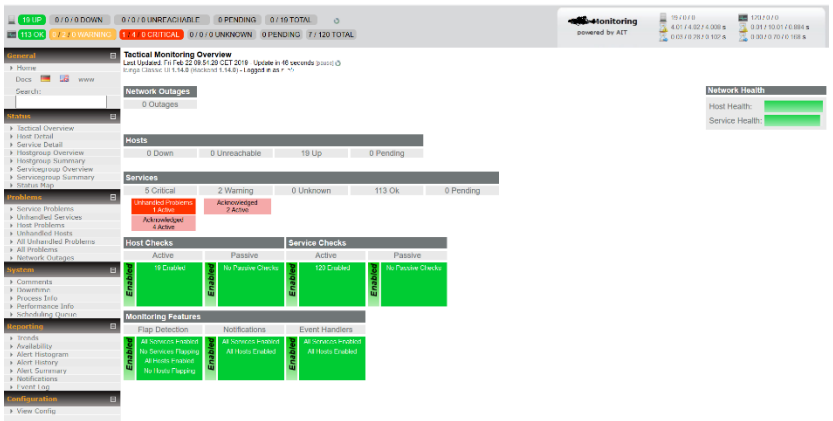


Alerting and Notification
Dashboards and mail notifications

VISUAL REPRESENTATION

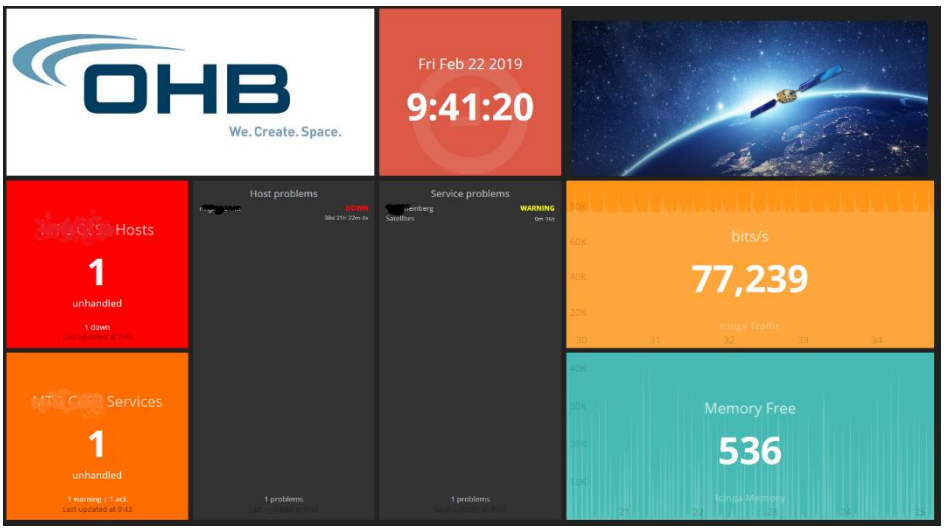
Icinga UI

The Icinga web interface present information on host and service status, history, notifications and status maps to show the health of a network in real-time.

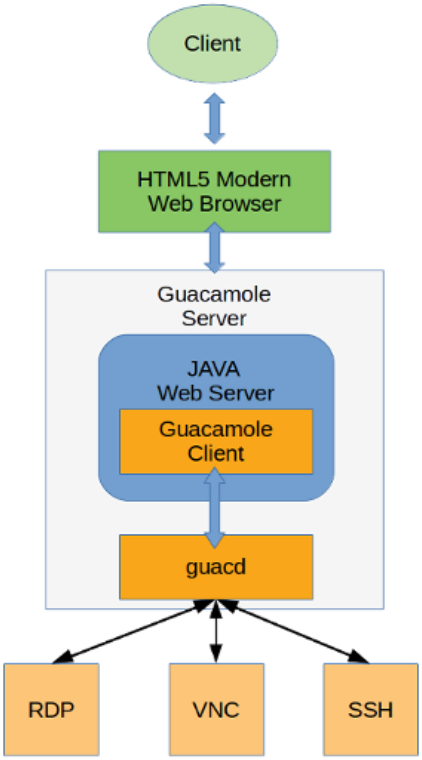


NagVis
NagVis[5] visualize the Icinga data so to display processes like system and service status in a "closer to real word" representation:

Icinga Dashing
Icinga Dashing[6] is a Sinatra based framework that allows us to build beautiful dashboards with real time information:



HOW IS THE REMOTE ACCESS IMPLEMENTED
Apache Guacamole is a free and open source clientless remote desktop gateway - as a true web application allows us to access remote machines from anywhere without violating the policy of the workplace, and without requiring the installation of special clients. Apache Guacamole allows us to centralize access to a group of machines, and specify on a per-user basis which machines are accessible. Rather than remember a list of machines and credentials, users need only log into the Apache Guacamole server and click on one of the connections listed.
Apache Guacamole supports multiple connection methods such as SSH, VNC, and RDP.



CONCLUSION

The implementation of SPOC was a big step ahead because it eliminated the need for barriers in front of our windows and we are now able to access the project networks in a very easy way… for troubleshooting, software uploads or configurations. The use of open-source products made this solution very cheap but also powerful because the used applications are well-known and due to this also very good supported in the community.

For us the ability to have a central point where we can get an overview of the status of all test setups and also a central point where we can access the setups was a big step ahead in terms of centralisation.

REFERENCES

[1] https://www.terma.com/
[2] https://icinga.com/
[3] https://guacamole.apache.org/
[4] https://docs.pnp4nagios.org/start
[5] http://www.nagvis.org/
[6] https://www.unixe.de/icinga2-dashing/