Haaga-Helia
University of Applied Sciences

Data Centre Monitoring System Change for Company X

Jan-Anders Wirén

Bachelor's Thesis
Degree Programme in
Business Information
Technology
2016

Haaga-Helia
University of Applied Sciences

Abstract

21.5.2016

Author(s)
Jan-Anders Wirén

Degree programme
Business Information Technology

Report/thesis title	**Number of pages and appendix pages**
Data Centre Monitoring System Change for Company X	**27 + 1**

Company X needs an up-to-date monitoring system to replace the current aging data centre monitoring system. More automation and less manual work is needed, as well as more visibility to present to customers.

There are several contenders, both open source and commercial, which have been introduced in this thesis report, their features have been compared to each other, after which one of them is recommended for implementation.

Licenses and prices, as well as hardware costs for the monitoring system, are beyond the scope and have not been considered and focus will be on suitability for the required tasks. The scope includes data centre and application monitoring whereas network monitoring is out of scope and though mentioned has not been considered.

The monitoring systems were compared to each other by several features, for which sufficient information was available in the documentation.

The result is that they are almost equal with regard to features.

In conclusion, within the narrow scope of this comparison, Nagios and CA UIM were found to have some immature parts, while Zabbix and BMC TrueSight seem quite mature and ready for large scale deployment.

It can also be seen that there is no inherent advantage or disadvantage to a system being developed commercially or as open source.

Keywords
Data centre monitoring system, data centre, commercial software, open source software

Table of Contents

1 Introduction 1
1.1 Background on company 1
1.2 Reason for change and goals of new monitoring system 1
1.3 Introduction to Monitoring 3
1.4 Background on current monitoring system 4
2 Author background 5
2.1 Courses completed during studies 5
2.1.1 Introduction to Business and Business Processes, Business Process Design and Modelling 5
2.1.2 Managing Business Information Systems Development 5
2.1.3 Information System Development Project 5
2.2 Personal experience 6
2.3 Work experience 7
2.3.1 Work placement 7
2.3.2 Current work 7
3 Plan of action 7
3.1 Constructing the requirements 7
3.2 Comparing features 8
3.3 Evaluate and Comment 8
4 Requirements for the new monitoring system 8
4.1 Environment 8
4.2 Required features 9
4.2.1 Heartbeat monitoring: Basic uptime (ping) monitoring 9
4.2.2 Basic metrics: CPU usage, Memory usage, Disk usage and Network usage, with history for usage pattern tracking 9
4.2.3 Open API for export of data and alarms into new monitoring system 10
4.2.4 High Availability and Redundancy 10
4.2.5 Web services monitoring 11
4.2.6 Database monitoring 11
4.2.7 Customized checks 11
4.2.8 Hardware and IPMI (Intelligent Platform Management Interface) monitoring 12
4.2.9 Application Server monitoring 12
4.2.10 Virtual Machine host monitoring 12
4.2.11 Distributed monitoring 12
4.2.12 IPv6 support 13

4.2.13 History of monitoring data 13
4.3 Preferred features 13
4.3.1 Monitoring agent 13
4.3.2 Easy deployment with automatically provisioned servers and integration into templates and existing systems 13
4.3.3 Automatic notification generation on pre-set events 14
4.4 Other things to consider 14
4.4.1 Hardware requirements 14
4.4.2 Device auto-discovery 14
4.4.3 SNMP monitoring capability 14
4.4.4 Cloud monitoring 14
5 Project scope 15
5.1 Defined in scope 15
5.2 Out of scope 15
6 Candidates for replacement 16
6.1 Candidates 16
6.2 BMC TrueSight Operations Management 16
6.3 CA UIM (Unified Infrastructure Management) 16
6.4 Zabbix 17
6.5 Nagios 17
7 Comparison and evaluation 18
7.1 Feature comparison chart 18
7.2 Comparison method 18
7.3 Comparison data 19
7.3.1 Database monitoring 19
7.3.2 RESTful API support 19
7.3.3 Virtualization monitoring 20
7.3.4 Monitoring agent 20
7.3.5 Process and service monitoring 21
7.3.6 IPv6 support 21
7.4 Comparison results 21
8 Discussion 23
8.1 Considering results 23
8.2 Thesis process and learning evaluation 24
9 Conclusions 24
References 26
Appendices 28
Appendix 1. Home lab layout 28

1 Introduction

1.1 Background on company

The fictional company X NOC (Network Operations Centre) is monitoring devices all over the world, and the company has servers primarily in company data centres. It is important for the NOC to immediately get notification if something is not working properly, so action can be taken to remedy the issue. Currently there is an alarm screen provided by a monitoring system that aggregates alerts and alarms from a wide array of sources. The basis is formed by basic heartbeat monitoring, to check if the host device is responding to pings at all. Further on, there are a large amount of different alarm and alert notifications sent as SNMP (Simple Network Management Protocol) traps. SNMP is a standard protocol for collecting information from managed devices, supported by most or all of them. This is integrated into the device software (firmware) by the manufacturer, and is configurable to the extent enabled by the manufacturer. This is what is used to monitor the wide variety of networking hardware from various manufacturers and the systems can rarely be modified by customers. Support from manufacturers is generally only available if the devices are running an approved and licensed operating system or firmware as provided by the manufacturer.

Company X has been created based on reading and real-world experience, as a non-trivial environment to explore a monitoring system change operation with feature comparison and suitability assessment.

1.2 Reason for change and goals of new monitoring system

The existing server monitoring solution, in use currently, is rather dated and is approaching the end of its support. There are a number of customized checks and settings, dating back to when monitoring systems did not include as many standard checks as today, such as web site checks, which would be preferable to replace with standard checks, as well as a lack of flexibility. The customized checks were created primarily by people no longer employed by the company, and use resources to maintain, while modern monitoring systems include equivalent checks without customized code. This saves manual work and maintenance time for the monitoring professionals enabling them to focus on development of new, rather than maintenance of old.

The monitoring system needs to be able to integrate with existing infrastructure using open APIs (Application Programming Interface) and monitored with more advanced SLA (Service Level Agreement) reporting. Customers are requesting more detailed statistics

and more detailed SLA-reporting, which the existing solution is not able to conveniently provide, while the intention is that the new monitoring system will be able to automate this. Currently the alarms from the existing monitoring system are shown on a different and separate alarm screen from the primary NOC alarm screen.

The new monitoring system is required to have output that can be processed and piped into the existing primary alarm screen, so that all alarms are available on one alarm screen. This will facilitate and clarify alarm processing and monitoring for NOC personnel, who may focus their attention on the primary screen, rather than two distinct screens, and avoid situations where some alarms may be missed entirely for some time. Human error is a factor to consider, and mitigate as much as possible.

Below is a flowchart showing the monitoring process as it is currently, and what it is to be with the new server monitoring system. All servers managed by Company X are hosted in Company X data centres, so network device monitoring at customer sites is not affected by this change and will continue as before.

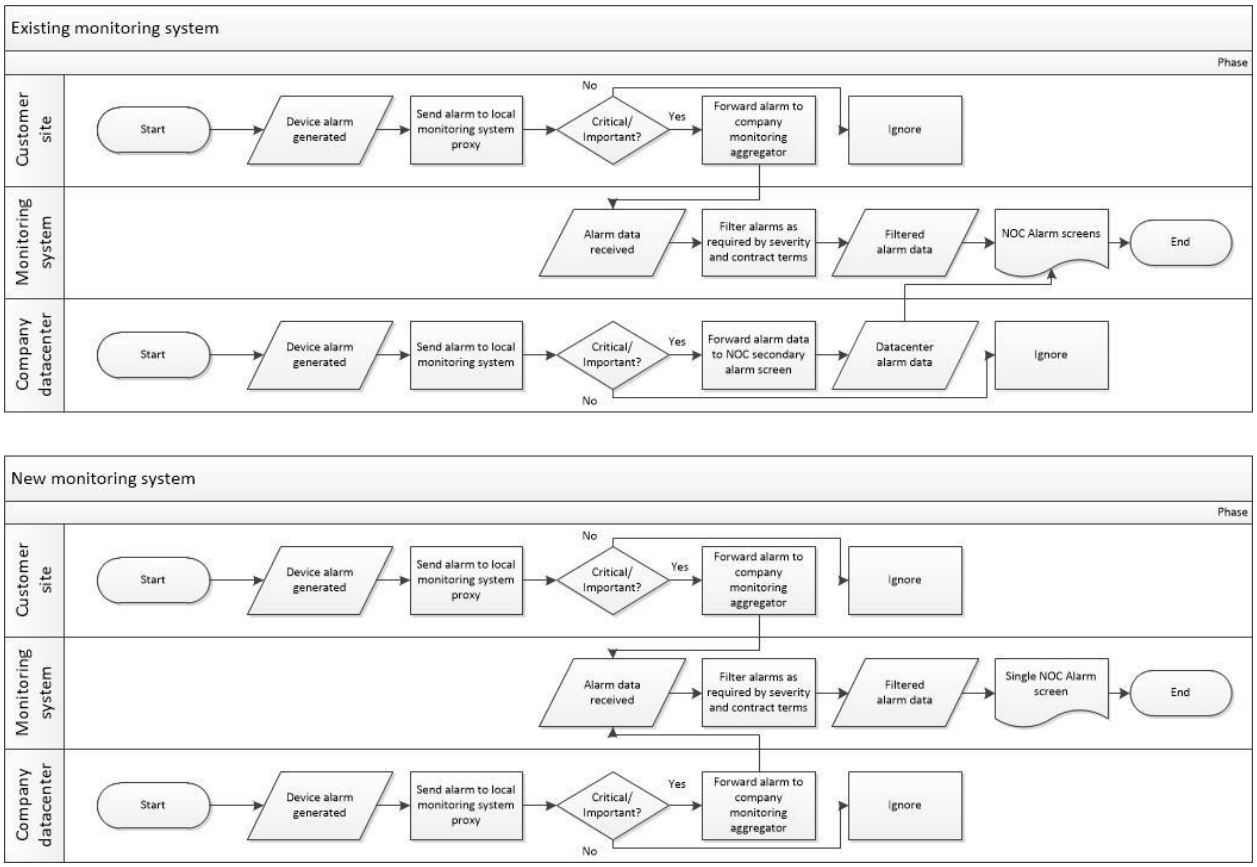

Figure 1. Flowcharts showing the process in the new and old systems

The flowchart shows the process broken down into steps, with differences between existing and new monitoring systems. One of the biggest issues currently, is having to

monitor two different and separate alarm screens. The new monitoring system is to solve this issue by sending all alarm data directly to the primary aggregator which puts the data directly on the correct monitoring screen, since this has become a viable option with the new monitoring system. This is expected to improve focus of NOC personnel and reduces the chance of mistakes happening.

1.3 Introduction to Monitoring

”In concept, monitoring systems are simple: an extra system or collection of systems whose job is to watch the other systems for problems.” (Josephsen 2007, xix)

Monitoring is collection of information. Useful information is collected about all servers and devices in the network, or networks, so the administrators will have the necessary information to manage and control them. The networks can be local or remote, with various ways of transferring the information to the monitoring system from remote networks, such as proxies. With the number of devices and servers, it becomes impossible to monitor them manually, especially with the commonly small IT teams of today.

This is why automated monitoring systems have been developed. The system will poll devices and systems in various ways, to see that they are alive, as well as record their CPU, memory loads, free disk space and network usage. This provides valuable information regarding future planning, such as disk space increases and adding of memory or CPU resources.

Ways to collect the information include monitoring agents on the servers, agentless monitoring that periodically logs in to collect data and SNMP monitoring.

“The problem is that monitoring systems are not turnkey solutions. They require a large amount of customization before they start solving problems...” (Josephsen 2007, xxi)

In practice, every company and entity needs a different monitoring set up, since no two environments are exactly alike, and many company networks can grow quite complex when including all partner connections and remote sites.

1.4 Background on current monitoring system

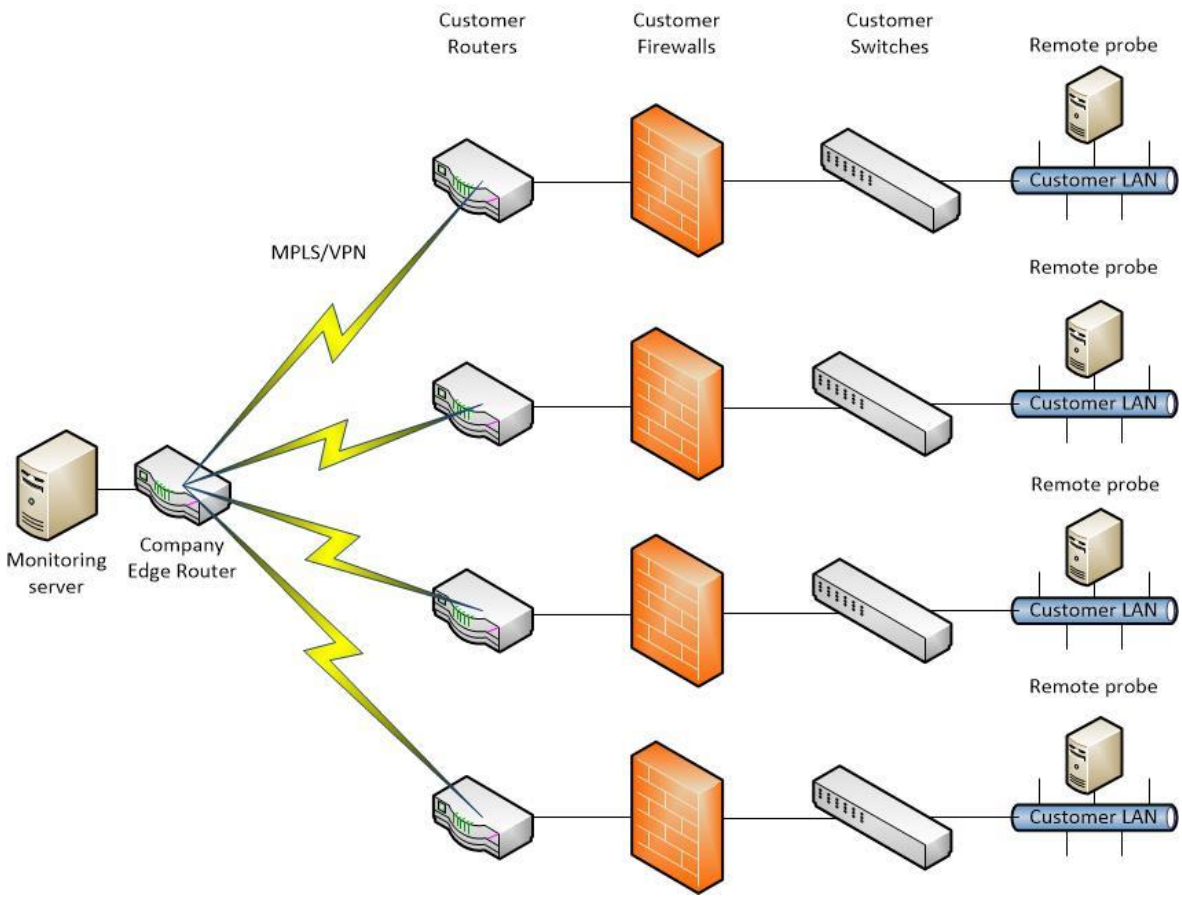

Figure 2. Basic monitoring configuration for customer networks, simplified view without special considerations. Does not take into account redundant links or servers, encrypted links used (MPLS (Multiprotocol Label Switching) / VPN (Virtual Private Network))

Figure 2 shows a picture of the common network monitoring implementation. A remote probe is placed inside the customer network, where it is able to monitor all customer devices and receive SNMP traps and data, and forward the aggregated data to the primary monitoring system at company premises.

For server monitoring, some monitoring systems use an agent daemon installed on the server to be monitored, that is connected to the monitoring server, which provides basic information, such as CPU, memory disk and network usage among others. The agents installed for monitoring the servers may provide any number of details on the running system and may even perform actions, such as restarting a service or process that has stopped. Monitoring clusters and applications are also supported by some monitoring systems. Servers may also have agentless monitoring.

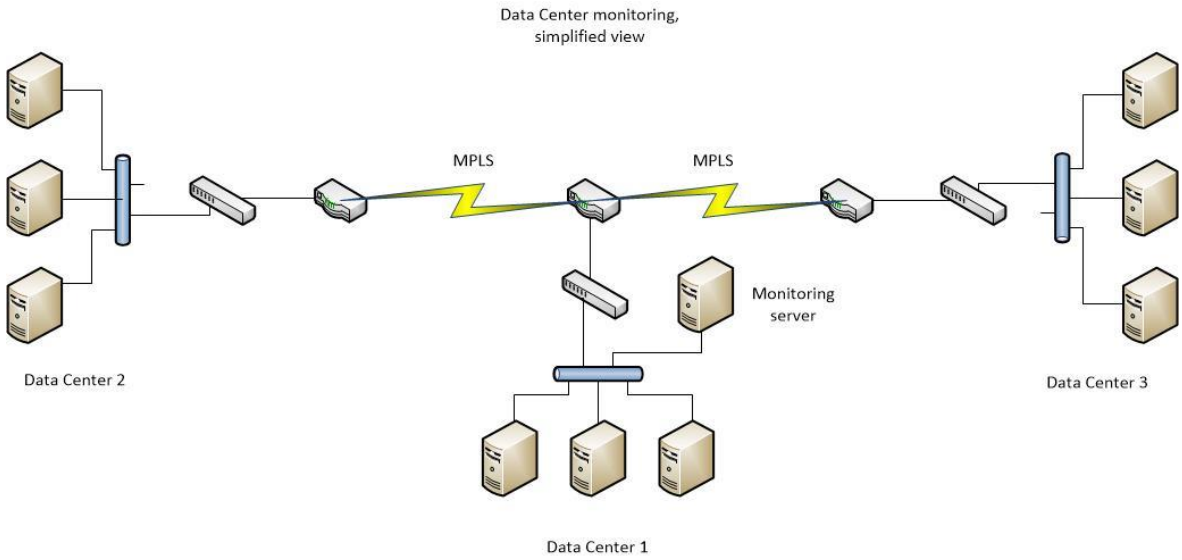

Figure 3. Company data centre monitoring configuration. There are encrypted communication links (MPLS/VPN) between data centres, connecting to the primary monitoring system.

2 Author background

2.1 Courses completed during studies

2.1.1 Introduction to Business and Business Processes, Business Process Design and Modelling

Provided an eye into corporate processes and procedure, plus also designing processes and modelling them, which is related to monitoring and monitoring processes. Helps to understand the monitoring processes.

2.1.2 Managing Business Information Systems Development

This taught processes and process development regarding Information systems in Business, which aligns quite well with the issue being considered, giving more knowledge on how to deal with processes.

2.1.3 Information System Development Project

This course placed us in a position of developing a project in a group. Project that our group picked was developing a dashboard for a Zabbix monitoring system, requiring familiarity with the system. Zabbix is also included in this review. Handling such a project was a very educational experience, since we have few opportunities during studies to

partake in larger projects in a group. It gives good experience and advice for the current project, making it easier to perform successfully.

2.2 Personal experience

I have used computers since I was 4 years old, and am self-taught, starting at the DOS command prompt, and later moving on to Windows. In the beginning, I was focusing mostly on games, but got into more and more things over the years.

Besides the courses during studies, I have experience with my home lab since between 1998 and 1999 I was able to piece together an extra computer from spare parts. This started my home computer lab, since it made me free to experiment with different Operating Systems and technologies as I could use my desktop for finding information online when it was needed. In the beginning, naturally, I was not familiar with Linux, FreeBSD, OpenBSD and NetBSD, but I started using them, experimenting with them, and setting up services and applications on them, and read instructions online, primarily the Operating System manuals.

Linux was not commonly known at the time, (w3schools.com, 2016.), leading to information not being abundant, which led to quite a large amount of troubleshooting and trying all the combinations until it worked, on issues where the instructions were not clear. This Linux and BSD use however, taught me about running systems and made me familiar with them, which has been very useful since then. At the moment, Linux is among the largest server platforms in use (Netcraft 2016.), and also the base for a number of network devices, such as Check Point Firewalls. (Wikipedia 2016.) FreeBSD is the base for many Juniper network devices. (Juniper Documentation 2016.) I have also monitored my systems over the years, with a variety of products, primarily the free and/or open source monitoring solutions.

Around 2003, my home lab had grown to 10 computers running, each hosting different services and running a variety of applications. Around this point, proper support for multiple CPUs was also becoming increasingly common in BSD or Linux systems. It was also around that time, that I was considering monitoring for the first time. Shortly afterwards, I set up monitors on the systems, showing the basic statistics.

Since then, I have spent over a decade using, installing, tweaking, troubleshooting and working with Windows, Linux and BSD systems. The last year I have been working with

systems and networks, doing troubleshooting and some network and firewall configuration as my daily job.

Currently, my home lab includes 4 host systems for Virtual Machines, as well as two storage systems with supporting switches and firewall, allowing me to build and use almost whichever systems I want. Additional switches are also available, as needed. More detailed description in appendix 1.

2.3 Work experience

2.3.1 Work placement

Work placement and subsequent working at company requires constantly working with monitoring systems. The position primarily deals with monitoring systems and servers, dealing with alarms and issues that arise, reported by both the monitoring systems and also issues reported by customers. Some issues are harder for monitoring systems to catch, and in many cases those issues are also harder to troubleshoot. One year of working closely with monitoring has taught a lot about enterprise monitoring system usage, and also about the inner workings of monitoring systems.

2.3.2 Current work

My current work continues within the same company where my work placement took place. Work description has not changed much, but with more of a focus on troubleshooting and customer-requested changes, rather than incidents. Incidents are still a large part of the job. Monitoring systems are at the core of the work.

3 Plan of action

3.1 Constructing the requirements

Building the requirements for the new data centre monitoring system will be done by researching monitoring systems done in practice, case studies, reviewing best practices, speaking to a developer of monitoring systems and also based on personal experience and working experience.

The monitoring systems considered in this study, have case studies available, which provide information on real world implementations, providing insight into the process of building a monitoring system. (Nagios 2016.), (CA 2016.), (BMC 2016.), (Zabbix 2016.)

Monitoring system best practices are freely and generally available from a number of sources, and a number of these have been reviewed and used during the process of building requirements. (MSDN 2016.)

The author also personally knows a monitoring system developer who has agreed to answer some questions regarding the process of building a monitoring system. (Boyanich 20 May 2016.)

The author has been working in a NOC for over a year, including the work placement period during studies, where the monitoring system is the primary tool that enables detection of issues to be resolved. Monitoring is the core of the position as all actions performed depend on the issues detected by the monitoring system. This has provided significant insight into the workings and complexities of monitoring systems.

The author has also been running a home lab for nearly 20 years, during which monitoring systems have also been implemented and used, both for learning and for notification of issues occurring.

3.2 Comparing features

Based on the requirements, compare the features available in the competing monitoring systems and confirm that the features are actually available. The features required are expected to be available, since only the most popular monitoring systems are considered.

Generally available information is used to compare the features, and as advertised by the providers.

3.3 Evaluate and Comment

Evaluating the results of comparison and based on findings evaluate and comment on the suitability of each system for replacing the current aging monitoring system.

4 Requirements for the new monitoring system

4.1 Environment

The environment to be monitored consists of several data centres with both physical servers which are primarily database hosts and important services, as well as physical servers that function as hosts for virtual machines. These include both company servers and customer servers, in various configurations. A number of customers have their own

virtual machine host clusters running and managed in one of the company data centres, while others use the company capacity cluster for their (virtualized) servers.

Everything needs to be monitored, to ensure services remain available at all times. Monitoring also provides indication of upgrade needs and performance issues. Current monitoring includes a number of undesirable customizations for common cases of today, due to limitations in the old monitoring system, which are to be replaced by standard solutions available in the selected monitoring system. Currently, the data centre SLA reporting is confined to monthly reports, but customers are requesting more and quicker visibility at any given time.

These data centres are connected to the primary data centre by MPLS links, over which the monitoring agent data is also transferred, so no proxies or remote probes are required for monitoring the company data centres since they are physically within the same metropolitan area. Monitoring system data will always be passing through encrypted communication links, with all points requiring minimum two-factor authentication, so encryption of monitoring data is not strictly required.

4.2 Required features

4.2.1 Heartbeat monitoring: Basic uptime (ping) monitoring

Heartbeat monitoring refers to sending the target host an ICMP Echo Request and waiting for an ICMP Echo Reply, which, when successfully received, indicates that the target host is alive.

4.2.2 Basic metrics: CPU usage, Memory usage, Disk usage and Network usage, with history for usage pattern tracking

Basic metrics collected from a system will provide insight into the usage of resources. Monitoring CPU usage and memory usage will show the load on the system, and in case usage is high, it suggests the application is utilizing all available capacity, or that bottlenecks exist, and that it may be time to consider growing the capacity or investigating performance issues, to keep the application functional. Increased amount of users may cause this, but high utilization may also suggest that something unplanned, or suspicious is going on, if load is deviating from normal usage patterns.

Disk usage tracking is needed, since most systems use disk space for logs and possibly other kinds of data, for example in the case of a file server. Disk usage alarms will alert

the administrator to increase the space allocation or remove old data or logs, before disk space runs out. In case disk space runs out, it will frequently cause issues with applications that use it, often causing the application to break. Monitoring network utilization is useful for finding bottlenecks and can also be viewed to assess traffic requirements for various applications and possible need of increased capacity.

Historical monitoring data is important for finding patterns and trends, troubleshooting and capacity planning.

Receiving metrics from a monitored system is also a way to confirm that the system is alive, eliminating the need for specific heartbeat monitoring.

4.2.3 Open API for export of data and alarms into new monitoring system

Current monitoring systems generally include a REST (Representational State Transfer) API for accessing data. REST uses simple HTTP calls, rather than complex mechanisms. It is used for machine to machine communication.This interface is required to get the monitoring data into the primary monitoring and alerting system and is one of the most important concerns for this comparison, as the intended function can not be accomplished unless available.

4.2.4 High Availability and Redundancy

High availability is a critical feature for monitoring large networks, since losing monitoring visibility may cause some systems or applications to remain unworkable and unnoticed, without any notifications or alarms alerting the administrators. This could potentially cause the customer large financial losses, and thus expose the company to large costs.

Monitoring systems also need to be monitored to detect issues and hardware failures, but there are ways to make the system more robust, primarily eliminating single points of failure.

For the system to be highly available, it needs to remain in operation in case any of its components cease to function. Solutions for handling this include load-balanced duplicated nodes and services, clustered databases and mirrored fault-tolerant storage, usually provided by a SAN (Storage Area Network) which tolerates broken components without failing or data loss. The fact is that all hardware will break at some point, and software frequently have issues due to bugs. This means highly available systems should be designed with breakage in mind.

This provides additional benefits, such as being able to upgrade or maintain individual devices and servers without downtime as there are always nodes that will handle operation while one is down for maintenance. When breakage occurs, applications will fail over to use the unaffected components, while the failed parts are being replaced, and load will be distributed across all the nodes while operational. Fail over may also be implemented by having an active node that services user requests, while another node is a passive stand-by node, which activates when failure in the active node occurs, swiftly restoring operation.

Another item worth consideration, especially in cases where multiple data centres are available, is placing the load-balanced nodes, or passive stand-by devices, in another data centre entirely. This will prevent a major outage even in cases where the whole data centre becomes inoperative. This will protect services and keep them running, especially during a large network operator outage or maintenance, or failure of the power grid and backup power in a data centre, however unlikely that may be.

4.2.5 Web services monitoring

Web services are used for communication between different services and applications using XML or JSON in most common cases. They are simpler to implement as they are machine to machine communication without presentation requirements.

4.2.6 Database monitoring

Databases are used for storing data, anything from assets and employee details to customer information and other business-critical data. They are commonly found in almost all companies.

Databases are often critical parts of infrastructure and monitoring will allow outages or failures to be detected and resolved in as short a time as possible. Monitoring also provides information on storage requirements, space usage and performance, simplifying their upkeep and space allocation as they grow, and if necessary, allocation of other additional resources. Modules would preferably available to handle each of the common databases, such as SAP, Oracle, MS SQL, MySQL/MariaDB and PostgreSQL.

4.2.7 Customized checks

The possibility of using customized checks, beyond common monitoring functions, is also a valuable addition, since it enables monitoring of customized systems and applications.

These are common in enterprises since business requirements vary from enterprise to enterprise, leading to large numbers of customized systems developed in-house, as well as a number of legacy systems. It is important to be able to monitor these special cases as well, since they may be critical to conducting business.

4.2.8 Hardware and IPMI (Intelligent Platform Management Interface) monitoring

Hardware and IPMI monitoring will provide information on the health of the underlying hardware, on top of which everything is running. In case physical parts become faulty, such as hard disk drives or power supplies, it is crucial to be notified of the occurrence, and have the faulty parts replaced as soon as possible. This is particularly important in case the hardware in question is hosting a large number of virtualized systems and applications, where the failure of a physical component could cripple a number of services and systems.

In case a server fails or shuts down, it is likely to be accessible through the IPMI interface. It is a small dedicated system within the server which provides management access independently of the host system. This will allow for some troubleshooting and potentially recovery of the system, without having a technician physically access the system.

4.2.9 Application Server monitoring

The monitoring system should also support monitoring of Application servers, which are very commonly used in enterprises for a variety of customized applications, as well as applications such as Content Management Systems.

4.2.10 Virtual Machine host monitoring

While Virtual Machines are monitored in the same way as physical servers, it is also necessary to monitor the Virtual Machine host environments. CPU, memory and storage capacity are monitored and administrators will be alerted to resource shortages, on a particular host or in the cluster, and will be able to take action in time, before Virtual Machines start getting migrated around due to capacity running out.

4.2.11 Distributed monitoring

Distributed monitoring is useful in larger environments, where the networks are located in a variety of locations. Remote probes can collect information and aggregate results to be sent to the primary monitoring system at headquarters.

While not strictly necessary at this time, it is likely that data centres will change and grow, so this possibility must be available, just as it is used for network monitoring within the company.

4.2.12 IPv6 support

While IPv6 is uncommon with regular users and home users, it will be increasingly necessary, making support for it a reasonable requirement, especially considering it is relatively mature. Due to IPv4 addresses running out, and actually having run out already, in parts of the world, IPv6 will become more and more important to consider, and as such, needs to be supported by the monitoring system. (Arin.net 2016)

4.2.13 History of monitoring data

Monitoring data and historical data needs to be available and easily accessible, so it can be used for troubleshooting and analysis. As customers are requiring more information and visibility into their service, data needs to be available and easily made available to customers for view as well. Preferably it will be automatically exported to customer portal for view. Monitoring data also needs to be available for SLA reporting.

4.3 Preferred features

4.3.1 Monitoring agent

Server monitoring systems normally include a monitoring agent, which provides more capabilities and deeper access to data versus agentless monitoring. Increased visibility is available into processes and services, and failures can be detected faster using agents. Agents also enable event handlers, enabling administrative actions such as restarting a service or application that has failed, or other actions based on predefined events.

4.3.2 Easy deployment with automatically provisioned servers and integration into templates and existing systems

Monitoring system configuration and agents are the most useful if they can be included in automatically provisioned servers, as well as easily installed on new systems through scripts in templates or through a configuration management system, such as Ansible or Puppet.

4.3.3 Automatic notification generation on pre-set events

Automatic alert processing is also desirable, in case there are special requirements, such as certain types of alarms being immediately reported to customers as per contract, or in case swift action is required, but requiring special skills such as databases, while outside working hours, or even during working hours.

4.4 Other things to consider

4.4.1 Hardware requirements

Reasonable hardware requirements are more difficult to define, as modern hardware is powerful. When monitoring thousands or even larger amounts of devices, performance may becomes a concern. There may be a significant difference in performance between different monitoring systems, depending on their efficiency. While a great deal of time and effort may be spent on the selection of suitable hardware, hardware requirements are out of scope for this review.

4.4.2 Device auto-discovery

Auto-discovery and auto-addition of new agents on newly installed and provisioned systems is desirable, avoiding manual intervention as much as possible, as long as security issues are addressed.

4.4.3 SNMP monitoring capability

SNMP monitoring is generally a critical requirement, in case this monitoring system is to be used as the sole monitoring system. For data centre monitoring, it is not required, as it is handled by the existing network monitoring system.

4.4.4 Cloud monitoring

Cloud monitoring is out of scope, but worth mentioning, since it is getting more commonplace that companies have their own public or private clouds, or even both. Many companies also use public clouds like AWS (Amazon Web Services) for testing or even data processing, since then their existing infrastructure can handle their primary business while cloud power can handle the extra testing needs without having to maintain extra infrastructure.

5 Project scope

5.1 Defined in scope

This project focuses on server monitoring, as there are a wide variety of available monitoring systems, both commercial and free/open source. The plan is to compare two commercial options, and two open source options, and select the better suited one for implementation in a service provider setting. The environment in which this monitoring system will be used is a datacentre service provider with various available capacity services.

Besides monitoring services, services like management for databases and email systems are also available as well as the services of the security operations centre. All the provided services and platforms also need to be monitored, so issues can be found and remedied swiftly. Monitoring by company is focused on the operating system, and the underlying hardware, as well as services for database servers, Active Directory and backup services. Customer applications are handled by customers or their partners, although some application checks may be implemented for monitoring.

The monitoring systems in this comparison are the most popular and common systems, so their features and capabilities do not differ much. Thus most issues will be considered only superficially, while several items will be examined in detail as basis for conclusions.

5.2 Out of scope

Monitoring system cost is out of scope for this review, and will not be considered. Licensing issues are also beyond the scope as all of the monitoring systems considered have paid support options available. SNMP monitoring is handled by the network monitoring system and thus out of scope for this review. Autodiscovery of devices is a feature worth considering, but out of scope for this review, as well as hardware requirements for the monitoring system.

The company also manages numerous customer sites and network devices, as well as network devices in customer data centres, but they are out of scope for this review, as the change affects only data centre monitoring.

6 Candidates for replacement

6.1 Candidates

The commercial offerings, CA UIM and BMC TrueSight were chosen after researching available commercial monitoring software solutions.

The open source monitoring systems, Zabbix and Nagios, have been selected since the author has run into them on numerous occasions and they are used by numerous large companies. (Zabbix 2016; Nagios 2016.)

6.2 BMC TrueSight Operations Management

BMC TrueSight covers a large number of things, from service availability to performance management of different parts of the IT environment, applications, infrastructure and middleware. It includes event management, application performance management, impact management, performance monitoring and data analytics.

In addition to basic monitoring, TrueSight provides automated detection and resolution of issues through event management as well as easy creation of application models that encompass a group of infrastructure components together into an application with automatic discovery included.

Performance metrics collected by TrueSight help measure and predict performance issues before they occur and also attempts to analyze probably cause for issues.
Data Analytics included with TrueSight learn baselines and helps detect anomalies. –log agent

Additional content packs are available, for instance an Oracle content pack, which comes preconfigured with best practices and domain knowledge.

TrueSight can be seamlessly integrated with other systems and services and reveal application performance issues from an end-user standpoint and with SLA reporting. (BMC 2016.)

6.3 CA UIM (Unified Infrastructure Management)

CA UIM is a service-centric monitoring platform designed to enable fast and easy identification of issues with alerting capabilities to enable reacting to issues before they affect users or service levels.

UIM includes comprehensive monitoring for leading database platforms out of the box, including end-user experiences, as well as storage monitoring to assist in preventing outages and pinpoint issues quickly. Capacity planning is also available.

Application monitoring is available for ensuring proper working of critical business applications such as SAP, MS Exchange as well as custom application monitoring through powerful APIs. Also available is easy monitoring for big data technologies like Hadoop, Cassandra and mongoDB. (CA 2016.)

6.4 Zabbix

Zabbix markets itself as “The Ultimate Enterprise-class Monitoring Platform”. Zabbix is open source and is available free of charge. Zabbix offers performance and scalability for monitoring and also a number of different kinds of visualizations and flexibly ways of analysing the data for the purpose of alerting.

Zabbix can be scaled up for very large environments with tens of thousands of devices, without vendor lock-in. Zabbix monitors resource usage trends for capacity planning purposes and also has commercial support available. (Zabbix 2016.)

6.5 Nagios

Nagios has a core open source version as well as a commercial version for enterprise customers. Nagios Core is the basic event scheduler, event processor and alert manager and can be extended using several APIs. Plugins are available for monitoring different kinds of devices, and for distributed monitoring, performance graphing and other features.

The commercial version of Nagios, called Nagios XI, provides monitoring for all mission-critical infrastructure and both the open source free version and commercial version have hundreds of plugins for monitoring virtually every kind of device. (Nagios 2016.)

7 Comparison and evaluation

7.1 Feature comparison chart

	BMC TrueSight	CA UIM	Zabbix	Nagios
Heartbeat monitoring	Yes	Yes	Yes	Yes
Basic metrics (CPU, Memory, Disk, Network usage)	Yes	Yes	Yes	Yes
Rest API for output	Yes	Yes	Yes	Yes
High Availability	Yes	Yes	Yes	Yes
Web Services Monitoring	Yes	Yes	Yes	Yes
Database Monitoring	Yes	Yes	Yes	Yes
Customized checks	Yes	Yes	Yes	Yes
Hardware and IPMI Monitoring	Yes	Yes	Yes	Yes
Monitoring Agent	Yes	Yes	Yes	Yes
Agent Event Handlers	Yes	Yes	Yes	Yes
Process and Service Monitoring	Yes	Yes	Yes	Yes
Easy deployment and Integration into Templates	Yes	Yes	Yes	Yes
Application Server Monitoring	Yes	Yes	Yes	Yes
Virtualization Monitoring	Yes	Yes	Yes	Yes
Automatic notification generation on pre-set events	Yes	Yes	Yes	Yes
Encrypted Agent Communications	Yes	Yes	Yes	Yes
Device Auto-Discovery	Yes	Yes	Yes	Yes
SNMP Monitoring capability	Yes	Yes	Yes	Yes
IPv6 Support	Yes	Yes	Yes	Yes
Data and History Export Capability	Yes	Yes	Yes	Yes
Distributed monitoring	Yes	Yes	Yes	Yes

Figure 4. Monitoring system feature comparison table

7.2 Comparison method

Due to the selected monitoring systems supporting all the requested features, several features with available in-depth information will be picked for a more detailed comparison. These features will be considered in-depth, and evaluated, and the comparison results will be based on these findings.

Comparison will be done based on publically available information provided by the monitoring system vendors.

Selected features are database monitoring, RESTful API support, virtualization monitoring, monitoring agent, process and service monitoring and ipv6 support due to availability of more detailed information.

7.3 Comparison data

7.3.1 Database monitoring

Database monitoring in Zabbix depends on ODBC (Open Database Connectivity) technology and drivers to collect data in RDBMS such as MySQL, PostgreSQL, Oracle and MS SQL Server. (Zabbix 2016.)

CA UIM has a wide variety of Database monitoring probes. Being a commercial monitoring system, support for commercial Databases seems to be emphasized, as shown by the probes available for IBM DB2, Oracle, Sybase and MS SQL, while also having a JDBC (Java Database Connectivity) probe for monitoring database servers (MS SQL, MySQL, Oracle, PostgreSQL, IBM DB2, IBM Informix). CA UIM also has monitoring probes for MongoDB and Cassandra which are NoSQL databases. (CA 2016.)

Nagios has fully supported monitoring for MS SQL, DB2, MySQL, PostgreSQL and Oracle. (Nagios 2016.) Plugins for other databases are also available on Nagios Exchange, but some do not have reviews or comments, making it difficult to assess their usefulness. (Nagios Exchange 2016.)

BMC TrueSight has monitoring solutions for IBM DB2, Oracle, MS SQL Server, and Sybase. On the NoSQL side, MongoDB monitoring is supported. (BMC Documentation 2016.)

7.3.2 RESTful API support

Zabbix comes with an API that provides access to almost all functions available. The API allows integration with any software able to make or accept external calls, such as ticketing systems or issue-tracking systems, such as JIRA. (Zabbix 2016.)

CA UIM provides a RESTful interface due to rising customer demand. The interface offers access to the UIM installation. Known issues are listed, such as performance slowdowns and communication errors without retries, suggesting that the RESTful interface is not very mature. (CA 2016.)

Nagios provides a RESTful API backend in the current version, allowing reading, writing, deleting and updating of data through commands that are authenticated. (Nagios 2016.)

BMC TrueSight provides a number of RESTful APIs for the purpose of easier integration with Infrastructure Management, which provides JSON output. (BMC Documentation 2016.)

7.3.3 Virtualization monitoring

Zabbix has a full set of VMWare monitoring capabilities. (Zabbix 2016.)

CA UIM has full sets of monitoring probes for Citrix Xenserver and XenApp, VMWare, Hyper-V and RHEV (Red Hat Enterprise Virtualization). In addition, Solaris Zones are supported, as well as a pre-release for Docker monitoring. (CA 2016.)

Nagios has good support for VMWare monitoring. (Nagios 2016.) Plugins for Citrix Xenserver monitoring exist on Nagios Exchange, but reviews are few. (Nagios Exchange 2016.)

BMC TrueSight has monitoring solutions for Citrix Xenserver and XenApp, VMWare, RHEV and System Centre VMM (Virtual Machine Manager). Docker monitoring is also supported. (BMC Documentation 2016.)

7.3.4 Monitoring agent

Zabbix agent runs natively on the host system and uses minimal CPU resources and memory and is able to run on devices with limited resources. It is available for Windows, AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris and Linux environments. The Zabbix agent supports both passive and active checks. (Zabbix 2016.)

CA UIM provides native agents that cause little or no stress to the devices being monitored. It allows data collection while disconnected, providing complete monitoring data. Supported OS are Windows, Linux, AIX, HP-UX and Solaris. (CA 2016.)

Nagios agent, NRPE (Nagios Remote Plugin Executor) is available. It has not received updates for some time but works. NRPE runs on Windows and Linux systems. (Nagios Exchange 2016.) A third party client, NSClient++, exists, which has more recent updates. It can be used to allow remote data collection. NSClient++ runs on Windows and Linux platforms. (NSClient 2016.)

BMC Patrol Agent runs natively on Windows, Linux, HP-UX, Solaris and AIX and provides monitoring data on resource usage and operating status. Windows version supports Windows-specific services. (BMC documentation 2016.)

7.3.5 Process and service monitoring

Zabbix agent handles monitoring of process status and memory usage, as well as service status for a number of services (ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap) as well as Windows services. (Zabbix 2016.)

CA UIM is monitoring process status, if it is running and should be, CPU and memory resource usage, as well as number of instances. Handle counts and services are monitored on Windows in addition. (CA 2016.)

Nagios official plugins include checks for processes and typical Linux/UNIX services. (Nagios Exchange 2016.) Nagios is capable of monitoring the state of all Windows and Linux services. (Nagios 2016.)

BMC Patrol Agent monitors process operational status and performance metrics. Windows services are monitored on Windows systems. (BMC documentation 2016.)

7.3.6 IPv6 support

Zabbix has full support for IPv6 for all components, allowing use in a mixed environment or pure IPv6 environment. (Zabbix 2016.)

CA UIM supports IPv6 for the most part, although for some probes the primary hub may only be IPv4. (CA 2016.)

Current version of Nagios supports IPv6. NRPE also supports IPv6. (Nagios 2016.)

BMC TrueSight Infrastructure Management and Patrol Agent all support IPv6 for communication. (BMC Documentation 2016.)

7.4 Comparison results

The competing systems will be compared with each other and placed in order according to suitability. Number 1 being best and number 4 being worst. Score will be added up and smallest score will represent the most suitable.

Within database monitoring, CA UIM has the most databases covered with monitoring probes, as well as generic probes, also including NoSQL databases which are gaining in popularity. (Couchbase 2016.) BMC TrueSight also has monitoring probes for the common databases, as well as support for one NoSQL database. Zabbix depends on generic ODBC access so is placed third and Nagios with a number of databases supported will be placed last due to some support being through plugins with little or no review.

RESTful API support is generally available in the monitoring systems being compared, so Zabbix, BMC TrueSight and Nagios will be placed equally. CA UIM support has listed issues and appears less mature, and is thus rated lower than the other three.

Virtualization monitoring is well covered by CA UIM and BMC TrueSight with UIM having an advantage with Solaris Zones, while Nagios and Zabbix properly support VMWare but little else.

Monitoring agents are available for all 4 systems, with the Zabbix agent running on the most systems. CA UIM and BMC TrueSight are the runners-up while Nagios depends on an agent that has not been updated recently, or alternately on a third party agent.

Process and Service monitoring is available in all four systems, and will be rated equal, due to similar features and capabilities.

IPv6 is equally supported by the monitoring systems, except for CA UIM which needs IPv4 to be used as the primary hub for some probes.

The results have been assembled in the table in Figure 5.

	BMC TrueSight	CA UIM	Zabbix	Nagios
Database Monitoring	2	1	3	4
RESTful API Support	1	2	1	1
Virtualization Monitoring	2	1	3	3
Monitoring agent	2	2	1	3
Process and Service Monitoring	1	1	1	1
IPv6 support	1	2	1	1
Score (Lower is better)	9	9	10	13

Figure 5. Detailed comparison results

The results suggest that BMC TrueSight and CA UIM would be equally suitable for consideration, feature-wise. There are issues to consider, however. CA UIM has a RESTful API implementation that appears less mature, which is an issue due to it being a central requirement for the replacement monitoring system. Zabbix may also be a suitable option, in case the lack of support for virtualization systems other than VMWare is not an issue. Nagios appears less suitable as a contender for the replacement monitoring system.

8 Discussion

8.1 Considering results

The result of this comparison shows that the common monitoring systems of today have numerous features and can be used for a large number of environments. The most important features are implemented by all of these systems, while the implementation details and level of maturity differ.

While researching the requirements, monitoring systems and their features and details, a large amount of other information was also discovered, which is not easily verifiable facts. For instance, usage experiences from a number of sources, on message boards, or blog posts, or comments sections. These provide some insight into the minds of real users of these systems, though this information is not directly useful in this comparison.

Based on this narrow review, differences are already starting to show up. Also, impression received from reading through research material and other material encountered during research is that the CA UIM product has some parts that are not quite mature. Even stronger impression is that Nagios has a strong core and the commercial offering has better supported parts, but that a lot of the monitoring plugins are immature and not quite ready for large-scale usage. The lack of a proper up-to-date agent provided by Nagios is also confusing, as one would expect this to be a core focus, and not something to be handled by a third party.

Availability of free software for performing complex tasks is a huge benefit, for people wanting to learn about system administration and for companies that have little resources but have skills. Zabbix in particular is an impressive system, usable by large companies, even though it is free and open source.

The findings of this comparison are based on vendor-provided information and as such should be verified in a proper testing environment, and in more depth, when selecting a monitoring system. Feature comparisons will rarely tell the whole story.

Due to the shortcomings in CA UIM RESTful API support, the monitoring systems to be considered here would be BMC TrueSight and Zabbix. Zabbix can be implemented at a lower price in case knowledgeable people exist in-house. Commercial support is also available for Zabbix if required. BMC TrueSight is a commercial product and is likely to be less flexible and the installation process may require more work and preparation.

8.2 Thesis process and learning evaluation

The scale of this project is the biggest I have undertaken by myself. The only thing somewhat comparable was the Information System Development Project-course during studies, which was a group project. It was useful in preparing for this experience. At times it was difficult to focus, with family and other things going on, but it also develops persistence, which is always useful.

The project plan was beneficial in managing the project, while timelines were subject to change, due to family or work interrupting planned schedules.

Monitoring is something I am interested in personally, and have implemented, and will re-implement in my home lab. I am also dealing with it nearly daily at my workplace. This project has made me dig deeply into monitoring and get a better understanding of it. It has given me new ideas on how to find information and use it for my own benefit in the future.

I have learned a lot and pushing myself to finish this project has been a useful and educational experience, which will serve me well, keeping me from giving up easily.

9 Conclusions

It is clear that monitoring systems are quite fully featured, whether they are free and open source, or commercial offerings. Several choices are available, which tends to be beneficial for further development, due to competition. The level that the open source monitoring systems that were considered here are at, indicates that there are people passionate about developing a better system behind it.

The findings of this review indicates that out of the four monitoring systems, one commercial monitoring system and one open source monitoring system, are quite mature and ready for large-scale usage. On the other hand, it also indicates that there are both commercial and open source monitoring systems that are in need of further development to bring them to maturity and stability.

References

Josephsen, D. 2007. Building a Monitoring Infrastructure with Nagios. Prentice Hall. Upper Saddle River.

Arin 2016. Arin IPv4 Free Pool Reaches Zero URL: https://www.arin.net/announcements/2015/20150924.html. Accessed: 21 May 2016.

Nagios 2016. Nagios – Network, Server and Log Monitoring Software. URL: https://www.nagios.com/. Accessed: 21 May 2016.

Nagios Exchange 2016. Nagios Exchange. URL: https://exchange.nagios.org/. Accessed: 21 May 2016.

W3schools 2016. OS Statistics. URL: http://www.w3schools.com/browsers/browsers_os.asp. Accessed: 21 May 2016.

BMC 2016. TrueSight Operations Management – BMC. URL: http://www.bmc.com/it-solutions/truesight-operations-management.html. Accessed: 21 May 2016.

BMC Documentation 2016. BMC TrueSight Operations Management 10.1 – BMC Documentation. URL: https://docs.bmc.com/docs/display/TSPS101/Home. Accessed: 21 May 2016.

CA 2016. CA Unified Infrastructure Management – CA Technologies. URL: http://www.ca.com/us/products/ca-unified-infrastructure-management.html. Accessed: 21 May 2016.

NSClient 2016. Welcome to NSClient++. URL: https://www.nsclient.org/. Accessed: 21 May 2016.

Couchbase 2016. Why NoSQL | Couchbase. URL: http://www.couchbase.com/nosql-resources/what-is-no-sql. Accessed: 21 May 2016.

Juniper Documentation 2016. Understanding Junos OS with Upgraded FreeBSD – Technical Documentation – Support – Juniper Networks. URL: http://www.juniper.net/documentation/en_US/junos15.1/topics/concept/understanding-junos-kernel-freedbsd10.html. Accessed: 21 May 2016.

Netcraft 2016. April 2016 Web Server Survey | Netcraft. URL: http://news.netcraft.com/archives/2016/04/21/april-2016-web-server-survey.html. Accessed: 21 May 2016.

Wikipedia 2016. Commercial products based on Red Hat Enterprise Linux. URL: https://en.wikipedia.org/wiki/Commercial_products_based_on_Red_Hat_Enterprise_Linux . Accessed: 21 May 2016.

MSDN 2016. Monitoring: Best practices. URL: https://msdn.microsoft.com/en-us/library/windows/desktop/bb226833(v=vs.85).aspx. Accessed: 21 May 2016.

Boyanich, A. 20 May 2016. Monitoring Software Development Engineer. Fujitsu Australia. Email interview.

Appendices

Appendix 1. Home lab layout

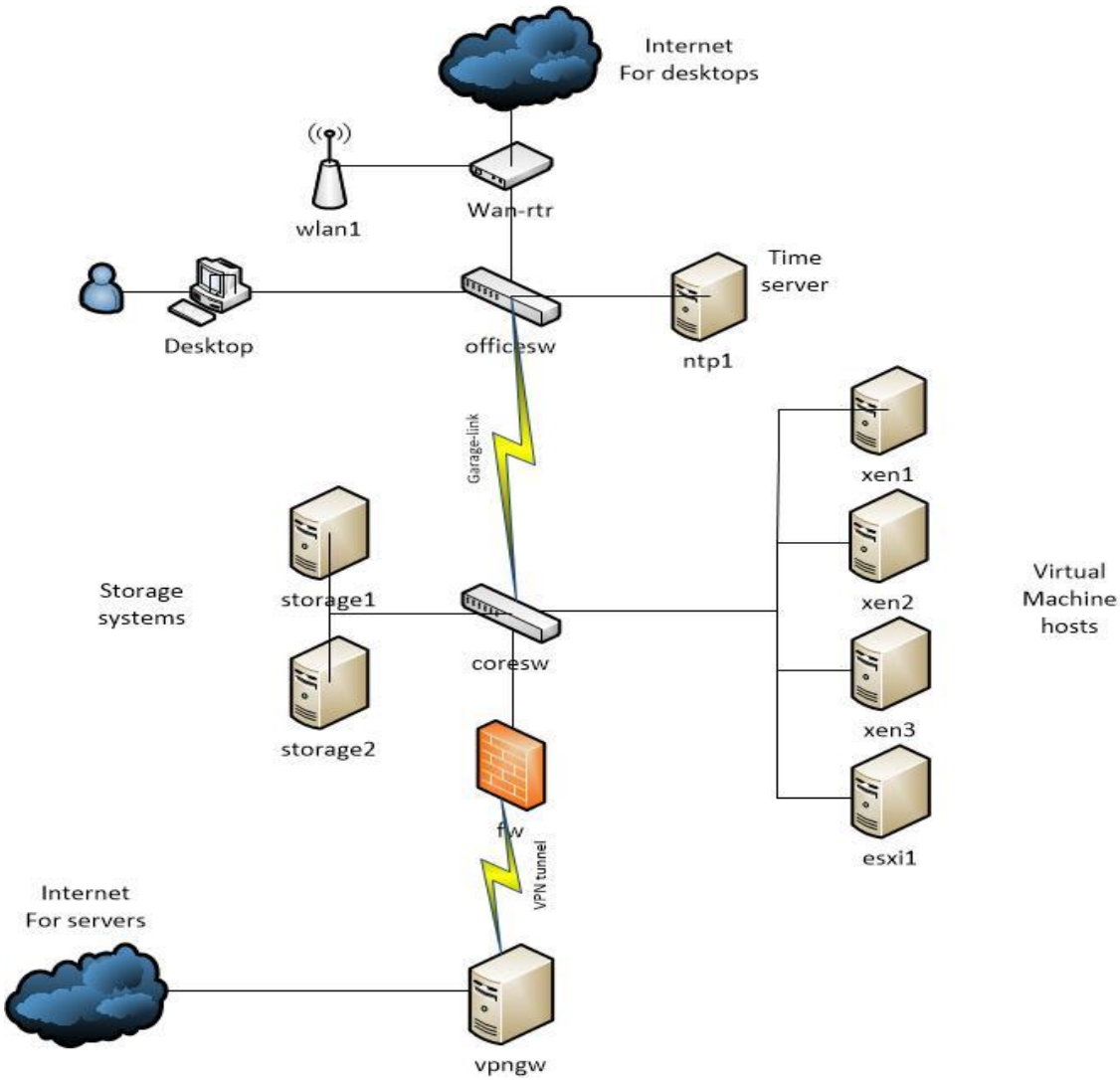

Figure 6. Home lab layout presented as a network diagram.

Figure 6 shows the home lab layout. Virtual Machine hosts and storage2 are rack servers, like servers in a data centre. Storage1 is a normal desktop filled with disks. All traffic between storage systems and Virtual Machine hosts goes through the core switch. Storage network is a separate virtual network (VLAN). Virtual Machine server network is also a separate virtual network that is connected to the internet through a pfSense firewall. All traffic from internet comes through a VPN tunnel and goes out the same way, due to a lack of public IP address provided by internet service provider. The VPN gateway is a server hosted in a data centre in France, where domain is pointed, and which forwards the traffic through the VPN tunnel to the appropriate server, passing though the firewall. A separate management network is also a separate virtual network and is connected to the office switch. It is accessible from desktops which do not go through firewall to internet.