Architecture

Allow web access to CMDB and monitoring data for on-call and SLA reporting

Sander van Dinten

Company: T-Systems
Requester: Stephan de Bot
Date, location: 20 February 2011, Tilburg
Version: 1.1

Title

Document history: Date	Version	Description	Author
21 December 2010	0.1	Initial draft	SvD
5 January 2011	0.2	Minor changes based on comments by René Zeestraten	SvD
6 January 2011	1.0	Finalized after comments from Stephan de Bot and some textual corrections	SvD
20 February 2011	1.1	Removed CMDB API, direct access to the CSV export is more efficient	SvD
22 February 2011	1.2	Added Active Directory query for the housekeeper	SvD

Table of contents

Terms

The following table describes the terms used throughout this document.

Term	Explanation
Content	The reports and information from the CMDB and Zabbix that are described in the requirements document
"CMDB/Zabbix"	The software component that correlates data from the CMDB and Zabbix, creates reports and displays in HTML format
Cron	Standard UNIX/Linux job scheduler

1. Introduction

This document provides a comprehensive architectural overview of the system that will be built during the "Allow web access to CMDB and monitoring data for on-call and SLA reporting" project. The scope of the document covers software architecture as well as the infrastructure architecture. Infrastructure architecture is included as it is closely related to the setup and design of the software.

The design of the system will be explained with the use of different figures. The realization of all requirements within the set limitations will be described in the last chapter.

This document was written during the design phase of the project and is based on the requirements document. The intended audience is the student, company mentor, school mentor and anyone who is interested.

2. Design

The design of the system is described in this chapter. To present the content to the user the system uses information from existing applications and data sources. The content itself is created by scripts and presented to the user through a web interface. The web interface is made available using existing components within the infrastructure.

2.1 Logical infrastructure

The logical infrastructure of the system and the communication protocols that are used are illustrated in Figure 1. The figure also shows the placement of the components in the different security zones.

The process flow is as follows:

- the user connects to the shared ISA service in the DMZ with an HTTPS connection;
- the ISA server will require the user to authenticate with the MGT domain;
- once authenticated, the ISA server will setup an HTTP connection to the web interface providing the content;
- depending on the user's request, "CMDB/Zabbix" will query the CMDB (via HTTP) and Zabbix (HTTPS/JSON) or will return a report to the user.

The CMDB is located in the PGN. In order to download information from the CMDB, "CMDB/Zabbix" has to use the shared proxy service located in the SSN.

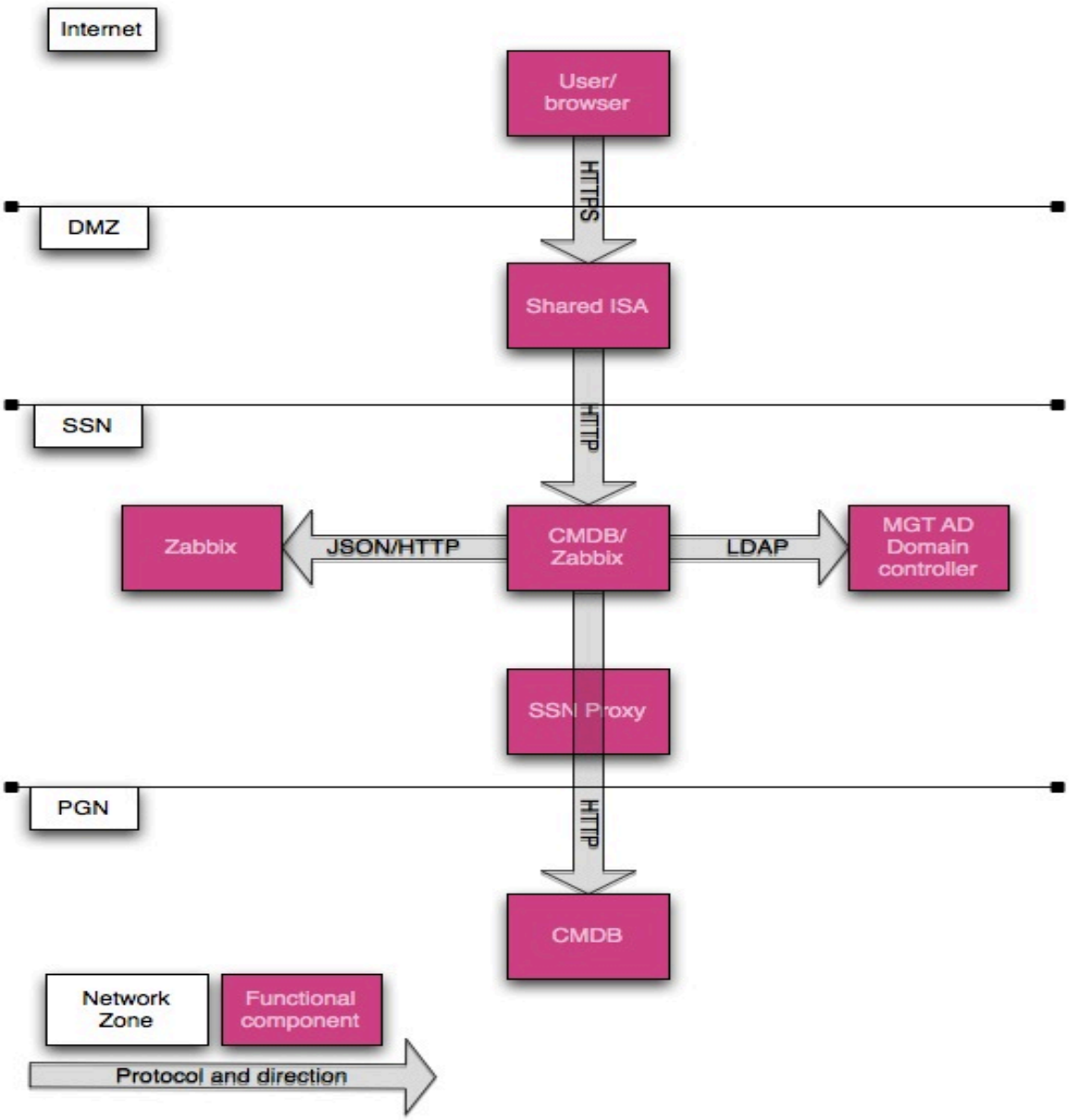

Figure 1

2.2 URLs

The following table describes the URLs that are used:

Application	URL
Shared ISA	https://161.85.120.187/cmdbzabbix/
CMDBZabbix	http://cmdbzabbix.ssn.philips.com/cmdbzabbix/
Zabbix	http://cmdbzabbix.ssn.philips.com/zabbix/
SSN Proxy	http://gdc-proxy.ssn.philips.com:8080/
CMDB	http://nlvg080.gdc1.ce.philips.com/CMDB/unix_output.txt
MGT AD	ldap://nlmehvdcs1dc000.mgt.philips.com

2.3 Physical infrastructure

Figure 2 illustrates the physical infrastructure design in combination with the placement of the logical components.

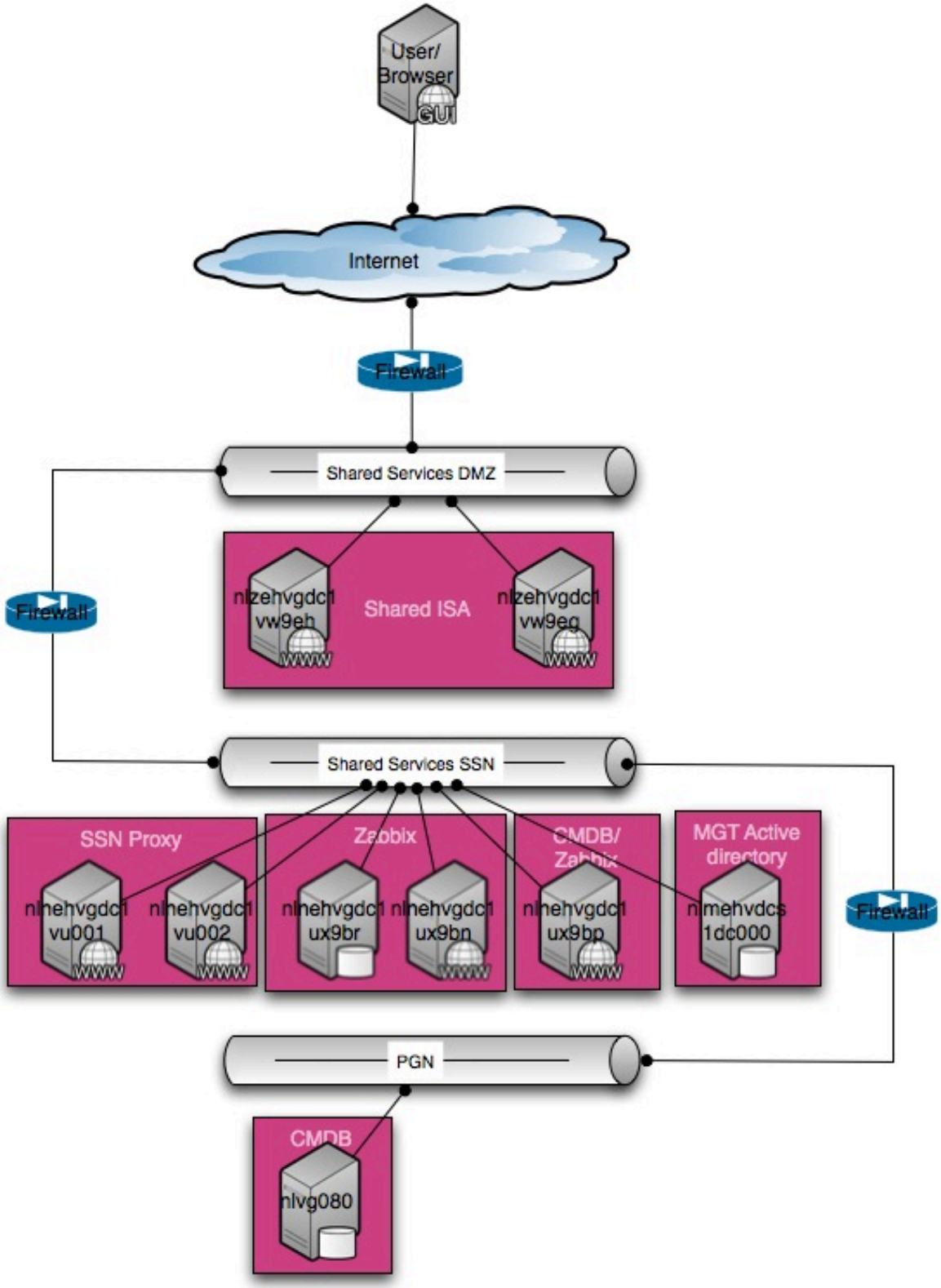

Figure 2

2.4 Existing infrastructure

The following components are part of the existing infrastructure and are either queried or provide access to "CMDB/Zabbix":

- Zabbix: monitoring application. Consists of a database server (nlnehvgdc1ux9br) and an application server (nlnehvgdc1ux9bn). "CMDB/Zabbix" only accesses the database server. The JSON connection over HTTP is local on nlnehvgdc1ux9bp.
- CMDB: Lotus Notes based application running on nlvg080. Provides a CSV export downloadable through HTTP.
- SSN Proxy: a load-balanced forward proxy service running on nlnehvgdc1vu001 and nlnehvgdc1vu002. Provides controlled access to the CMDB for "CMDB/Zabbix".
- Shared ISA: a load balanced reverse proxy service running on nlzehvgdc1vw9eh and nlzehvgdc1vw9eg in the DMZ. The load balancer also takes care of the SSL termination through SSL offloaders. The shared ISA component provides secure (HTTPS) and authenticated access to "CMDB/Zabbix" for the users from the Internet by authenticating a user to the MGT domain and forwarding the authentication information to eh "CMDB/Zabbix" component.
- nlnehvgdc1ux9bp: an existing server that is used as fallback server in case the Zabbix application server, nlnehvgdc1ux9bn, fails and is beyond repair.
- Nlmehvdcs1dc000 domain controller for the MGT active directory domain.

2.5 "CMDB/Zabbix"

Figure 3 illustrates the design of the "CMDB/Zabbix" component.

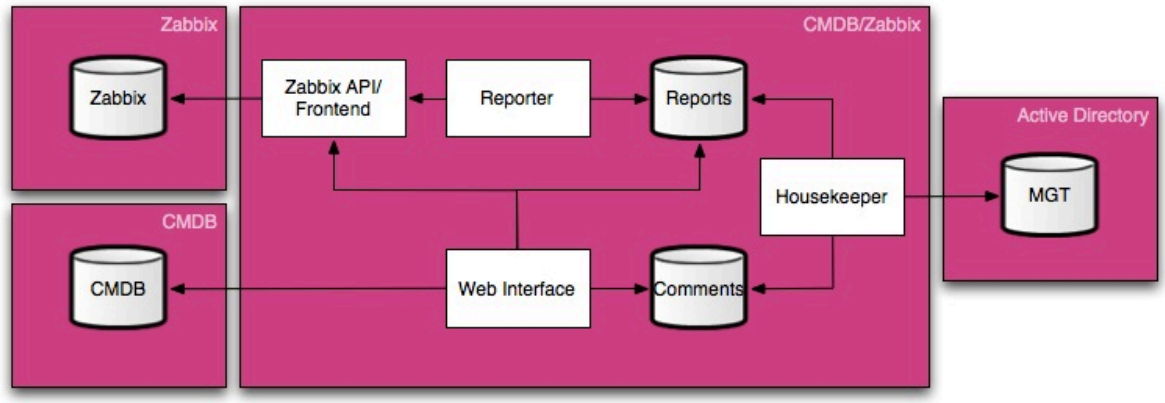

Figure 3

2.5.1 General

As shown in figure 2, the CMDB/Zabbix component runs on nlnehvgdc1ux9bp. The scripts, libraries and data for this component are located in /data/CMDBZabbix. Within this base directory, the following directories are available:

Subdirectory	Description
lib	Custom libraries; Zabbix API
bin	Scripts that are used command line or via cron; housekeeper and reporter
www	Scripts, CSS files and HTML pages for the web-interface
var	Variable data used throughout the CMDB/Zabbix component
etc	Directory to store the configuration file
Zabbix	The locally installed Zabbix web frontend

The CMDB/Zabbix component is written in Python.

The scheduled tasks, i.e. the housekeeper and the reporter, run from the UNIX cron daemon.

2.5.2 Zabbix API

The Zabbix API provides access to the external information sources in order to build the content. The Zabbix API is a publicly available Python library that provides a basic interface to Zabbix (logs into Zabbix, builds JSON calls, etc.). It can be downloaded via this link. The Zabbix API makes use of the Zabbix web frontend that is installed locally. The Zabbix frontend will be installed into an apache virtual host container. Access to the frontend will be limited to the "CMDB/Zabbix" component.

2.5.3 Reporter

The reporter is the script that builds the availability reports and places them in the reports directory. The reporter makes use of the Zabbix API to retrieve the required information from Zabbix. It runs every first day of every month at 7 AM and generates the availability report for the past month.

2.5.4 Housekeeper

The housekeeper is a script that runs every day at 6 AM. It:

- removes all comments older than 90 days and all reports older than 1 year.
- generates a list of users that is allowed access based on an Active Directory group in the MGT account

2.5.5 Web-interface

The web-interface is placed inside the document root of an Apache virtual host on the web server nlnehvgdc1ux9bp. Access to the web-interface is limited through the Apache virtual host configuration; it allows only the shared ISA servers (nlzehvgdc1vw9eh and nlzehvgdc1vw9eg) in the DMZ to connect. The web interface presents the content to the user.

3. Persistence

For the CMDB/Zabbix component no database is used, only flat files. To avoid possible simultaneous writes or locking problems, each file is only written during creation only.

3.1 Reports

The reports that are generated by the reporter are in CSV format. The filename describes the report type, report group and report period, e.g. "Availabilty-Unix-201012.csv" – availability report for the Unix servers for December 2010. The fields in the CSV output are the following:

Field	Description
HOSTNAME	The server
TOTAL SECS	Total seconds for the report period
OK	Total seconds the "Host unavailable / not reachable" trigger was in status OK meaning the server was up and running
PROBLEM	Total seconds the "Host unavailable / not reachable" trigger was in status PROBLEM meaning the server was unavailable
UKNOWN	Total seconds the "Host unavailable / not reachable" trigger was in status NOZABBIX meaning the server was not known to Zabbix (e.g. because it was added half way during the month)

Access to the reports is read-only for the web-interface and is read-write for the reporter and the housekeeper. The reports are created in the var/reports directory.

3.2 Comments

The comments that are entered by the user are stored as separate files in the var/comments directory. The naming convention for the comment files is <timestamp>-<server>, e.g. 1294136499-nlvu001. The timestamp is UNIX time. Each file has one line in the format: "Username#Comment". Both the housekeeper and the web-interface have read-write access to the var/comments directory