NetGlance NMS - An integrated network monitoring system

Aleksandr Ovcharov - a42648

Thesis presented to the School of Technology and Management in the scope of the Master in Sistemas de Informação.

Supervisors:

Rui Pedro Lopes

Natalia Efanova

This document does not include the suggestions made by the board.

Bragança

2019-2020

NetGlance NMS - An integrated network monitoring system

Aleksandr Ovcharov - a42648

Thesis presented to the School of Technology and Management in the scope of the Master in Sistemas de Informação.

Supervisors:

Rui Pedro Lopes

Natalia Efanova

This document does not include the suggestions made by the board.

Bragança

2019-2020

Dedication

To my graduate supervisors, friends and colleagues who helped me in this work.

Acknowledgment

This work was supported by Kuban State Agrarian University behalf I.T. Trubilin (KubSAU) and Instituto Politécnico de Bragança (IPB), as well as National Agency Erasmus+ Education and Training.

Abstract

This work is about IT infrastructure and, in particular, computer networks in KubSAU and IPB. Also, it is about a network monitoring system "NetGlance NMS" developed for KubSAU System Administration Department.

Work objective is to optimize the information structure for KubSAU and IPB.

During the work, following tasks were completed: Research the existing IPB information structure, Compare the information structure for KubSAU and IPB, Model the IPB computer network (topology, services), Research bottlenecks and potential pitfalls in the data-center and in the computer network of IPB, Research information security mechanisms in the computer network of IPB, Organize monitoring process for the computer network in KubSAU.

The most important impact of the work is an increasing network productivity and user experience as a result of creation and deploy a monitoring software.

Keywords: Information, Data, Computer network, Security, Data-center, Monitoring, Services, Network device, Link, ICMP, SNMP

Resumo

O trabalho descrito no âmbito desta dissertação incide sobre a infraestrutura TI e, em particular, sobre as redes de computadores da KubSAU e do IPB. Além disso, descreve-se um sistema de gestão integrada de redes, designada "NetGlance NMS", desenvolvido para o Departamento de Administração de Sistemas da KubSAU.

O objetivo do trabalho é desenvolver uma ferramenta para otimizar a gestão da estrutura de comunicações das duas instituições.

Durante o trabalho, as seguintes tarefas foram concluídas: levantamento da estrutura de comunicações do IPB, comparação da estrutura de comunicações entre a KubSAU e o IPB, modelação da rede de comunicações do IPB (topologia, serviços), estudo de possíveis estrangulamentos no datacenter e na rede de comunicações doIPB, estudo de mecanismos de segurança na rede de comunicações do IPB, organização do processo de monitorização da rede de comunicações da KubSAU.

O contributo mais relevante deste trabalho é o desenvolvimento de uma aplicação de gestão integrada de redes, de forma a contribuir para o aumento da produtividade da rede e da experiência dos utilizadores.

Palavras-chave: Redes de comunicações, Segurança de Rede, Data-center, Serviços de Monitorização, Dispositivos de Rede, Gestão Integrada de Redes.

Contents

List of Figures

Acronyms

AAA Authentication, Authorization and Accounting.

AJAX Asynchronous Javascript.

API Application Programming Interface.

ATM Asynchronous Transfer Mode.

CIM Common Information Model.

CLR Common Language Runtime.

CNA Cisco Network Assistant.

CSS Cascading Style Sheets.

CSV Comma-Separated Values.

DB Data Base.

DBMS Data Base Management System.

DHCP Dynamic Host Configuration Protocol.

DMTF Distributed Management Task Force, Inc..

DNS Domain Name System.

DOM Document Object Model.

EIGRP Enhanced Interior Gateway Routing Protocol.

ESTiG Escola Superior de Tecnologia e Gestão.

HTML HyperText Markup Language.

HTTP HyperText Transfer Protocol.

HTTPS HyperText Transfer Protocol Secure.

ICMP Internet Control Message Protocol.

IETF Internet Engineering Task Force.

IP Internet Protocol.

IPB Instituto Politécnico de Bragança.

IPMI Intelligent Platform Management Interface.

IS Information System.

JMAPI Java Management API.

JMX Java Management Extensions.

JSON JavaScript Object Notation.

KubSAU Kuban State Agrarian University behalf I.T. Trubilin.

LAN Local Area Network.

MAN Metropolitan Area Network.

MIB Management Information Database.

MVC Model-View-Controller.

NMC HP Network Management Center.

NMS Network Monitoring System.

NOM Micro Focus Network Operations Management.

OSI Open Systems Interconnection model.

OSPF Open Shortest Path First protocol.

OWIN Open Web Interface for .NET.

RMON Remote Network MONitoring.

SCOM Microsoft System Center Operations Manager.

SDN Software Defined Network.

SHA-2 Secure Hash Algorithm Version 2.

SMTP Simple Mail Transfer Protocol.

SNMP Simple Network Management Protocol.

SSH Secure Shell.

SSL Secure Sockets Layer.

STP Spanning Tree Protocol.

TCP Transmission Control Protocol.

TLS Transport Layer Security.

TME Tivoli Management Environment.

VLAN Virtual Local Area Network.

VPN Virtual Private Network.

WAN World Local Area Network.

WBEM Web-Based Enterprise Management.

WMI Windows Management Instrumentation.

WPA Wi-Fi Protected Access.

XSS Cross-Site Scripting.

Chapter 1

Introduction

Information technology is an important part of the modern world, which strives for quick and convenient access to information. In education institutions it is especially important because the educational process is built on that access. Maintaining the IT infrastructure of educational organizations in accordance with modern standards is the key to their success.

An information system is a system designed to store, search and process information. It includes related organizational resources (human, technical, financial, etc.) that provide and disseminate information. The information system is designed to provide the right information to the right people in a timely manner. Its main goal is to meet specific information needs within a specific subject area. Moreover, the functional result is information product — documents, data arrays, databases and information services. The totality of the organization's IS, as well as the relationships between them and external systems, forms the organization's IT infrastructure.

In order to understand the complex system of relationships in the IT environment, it is convenient to use the concept of services. IT service is a certain set of capabilities in combination with the corresponding information technologies that the provider grants to consumers. A service provides users with certain facilities. The essence of the service approach is to relieve the client on the use additional complexity or burden, while the service provider takes on the complexity, risks and costs of providing the service. As examples

of IT services are: e-mail, cloud storage, business application with long-term support. The options for IT services that are required for each organization can be completely different. It all depends on the direction of activity, the scale of the organization, the level of automation and development strategy. Often, IT services are divided into three main groups: support for IT infrastructure, support for business applications, support for users workflow. Each IT service has a number of parameters, such as time for service, availability, reliability, scale and cost, as well as performance and privacy. Users consume IT services through devices. It is important for the user that the service works correctly, fast and reliable.

Developers and administrators are on the other side of the service. Developers create service components, and administrators are involved with their implementation and support in a specific environment, for example, in an organization. It is important for administrators to have full and actual information about the systems that support the service. Monitoring of IT systems is used to achieve this goal.

1.1 Computer networks in education organizations

A computer network is a set of computers connected via communication channels and switching tools into a single system for messaging and user access to program, technical, information and organizational resources of the network. Computer network of the educational institution contains the organization's servers, work and personal devices of employees and students.

The purpose of creating a computer network in the organization is to provide fast and high-quality data exchange both within the enterprise and with the outside world (Internet). In educational institutions, networks are also used to disseminate and control knowledge for students.

A computer network supports different communication architectures, such as the client-server architecture, to provide services. In this case, users interact with the client part of the software on their devices, and the client part interacts with the server part

through the network. This model reduces the computing power and storage requirements of an end-user's device. A large amount of software cannot exist without such an architecture (for example, online conferences, cloud storage, e-mail and many others).

The computing network must meet the requirements of the services used. This primarily relates to supported network access technologies, network performance, reliability and security. Moreover, the network must meet not only current needs, but also potential needs that may arise as a result of an unplanned increase in load or urgent adaptation to new requirements. This must be taken into account, because the re-equipment of the network is a rather long and expensive process.

The educational process uses many services that require high-quality online access. Some examples include digital libraries, online lectures, online tests, a cloud-based storage of educational materials, collaborative work, communication with the teacher through the Internet. Also, in most educational organizations, a student's personal account is used to help organize their own training and perform most of the tasks online that previously required a physical presence (providing documents, payments, consultations with administrative staff). Thanks to these technologies, students can focus on gaining knowledge and working on their projects.

In the non-standard conditions that have developed in the world recently, e-learning technologies have become essential to cope with the requirements of maintaining physical distance during work and education. This is only possible with the widespread use of network technologies. Using the resources of educational institutions through the Internet, students can learn educational materials, pass knowledge control, keep in touch with teachers while staying at home. We can say that these organizations passed a tough exam to test their IT infrastructure readiness for modern challenges.

1.2 Network monitoring

The term network monitoring refers to the operation of a system that constantly monitors a computer network in search of slow or faulty systems and which, upon detection of

failures, reports them to the network administrator via email, messenger or other means of notification. These tasks are a subset of network management tasks. Monitoring is necessary in order to more effectively diagnose and solve problems when they arise, and thus reducing the duration of inoperative services. A network monitoring system monitors the network for problems caused by active and passive network equipment, end-user devices, or network connections. Failed requests (for example, if the connection cannot be established, it ends in timeout, or when the message was not delivered) usually cause a reaction from the monitoring system. As a reaction, an alarm can be sent to the system administrator, or a failure protection system can be activated automatically, which will temporarily decommission the problematic element until the problem is resolved. Active monitoring involves performing network requests and analyzing their results while passive monitoring collects and analyzes network traffic.

Active monitoring reports collected measurements via sending and receiving network packets. The active measurement system deals with metrics such as: utility, routers / routes, packet delay, packet retry, packet loss, unstable synchronization between arrivals, bandwidth measurement. Mostly it relies on the use of tools, such as the ping command, which measures the delay and packet loss, and traceroute, which helps to determine the network topology. Both of these tools send trial Internet Control Message Protocol (ICMP) packets to the destination and wait for that point to respond to the sender.

Passive monitoring, unlike active monitoring, does not add traffic to the network and does not change the traffic that already exists on the network. Passive measurements deal with information such as traffic and a mixture of protocols, the number of bits (bitrate), packet synchronization, and the time between arrivals. Passive monitoring can be better than active in that overhead data is not added to the network, but post-processing can cause a lot of time overhead. In real monitoring systems, a combination of these two methods of collecting information is often used.

Agent-managed monitoring reduces network load and Network Monitoring System (NMS) processing power. An agent is a system installed in another system for the purpose of monitoring and control. The agent can transmit statistical information to the NMS,

notify about events, provide an API for managing the target system. However, not in all cases it is possible to install an agent.

In some cases, administrators need to gather information about the network using some parameter. The most obvious is collecting information about a Virtual Local Area Network (VLAN) topology with a specific number. If the network controller is missing or does not support such functionality, the monitoring system collects information about all VLAN as part of the configuration of network devices. Data is collected at a specified time interval or at the initiative of a device sending an updated configuration to a monitoring system. Periodic data collection is poorly suited for operational work, and processing of a configuration change event is not supported by all devices, it has a different implementation (for example, it requires not only application, but also saving of a new configuration).

An alternative is to collect data on demand. Such data is not collected permanently, but only when a user needs it. For example, you can collect data on the topology of a specific VLAN on the network: existing on the device, status on the ports (none, tagged, untagged). This method is relatively simple and undemanding to network resources and computing power of devices. In the ideal case, only the data that is necessary for the user is requested and transmitted.

Chapter 2

Context and Technologies

The requirements for reliability and correct operation of IT services are growing every year. Almost every area of activity of organizations relies on the performance of the corresponding infrastructure. Accordingly, the cost of failures and the problems caused by them is high and growing.

Computer network is the most common cause of IT infrastructure failures. Joseph McKendrick, IOUG Research Analyst writes that 50% of unexpected IT infrastructure downtime happens due to network outages [1].

And yet, despite the fact that there are many solutions that allow you to monitor the network and network equipment, it is quite difficult to assess the effectiveness of its work. Even the search for the causes of the accident causes difficulties for untrained personnel, because they have to look through and analyze giant event logs containing an alarming amount of system messages, which often takes a lot of the working time.

For these reasons, monitoring is necessary in any corporate network, regardless of its topology, size and set of used technologies. The monitoring system allows you to quickly find out about existing and potential problems, conduct various types of analysis of data on the network, determine the priority of work. In large networks, special attention should be paid to the visibility of the output data and the convenience of the user, since the number of monitored devices can be very large.

The use of monitoring systems in the networks of educational institutions is very

important because of the specifics of these organizations. Often educational institutions have a large territory (campus), a mixture of technologies and decisions due to historical development (unplanned growth, partial updating of infrastructure) and the evolution of teaching methods. As a result, the network is a workable, but complex system built from equipment of various generations and manufacturers. Careful monitoring of such a network can be a non-trivial task.

2.1 Network Management

Network management is the process of administration and management of computer networks. The goal of this activity is to keep a network up to date and productive. It includes fault analysis, performance management, provisioning of networks and maintenance of the quality of service. Network management is carried out using various solutions and protocols and, like all IT technologies, it evolves and grows.

The Simple Network Management Protocol (SNMP) protocol is an old, but popular way for active and passive collecting information about network and for device management. SNMP is an application layer protocol that is part of the TCP/IP protocol. It allows administrators to manage network performance, find and fix network problems, and plan network growth. It collects statistics on traffic to the final host through passive sensors that are implemented with the router. The core of SNMP is a simple set of operations (and the information these operations gather) that gives administrators the ability to change the state of some SNMP based device. For example, you can use SNMP to shut down an interface on your router or check the speed at which your Ethernet interface is operating. SNMP can even monitor the temperature on your switch and warn you when it is too high [2].

There are three versions of this protocol. They differ in functionality, authentication capabilities and data encryption. SNMP has three key components: Managed Devices, Agents, and NMS.

Five PDUs (Protocol Data Units) are defined in SNMP; in other words, commands

supported by the protocol:

- GetRequest — a request to read the value of a given parameter of operation of the device, i.e., a Management Information Database (MIB) variable,
- GetNextRequest — a request to read the next variable of the MIB,
- GetResponse — transmit the MIB variable, it is a response to GetRequest,
- SetRequest — a request to set the value of an MIB variable, which allows controlling the functions of the device and configure the parameters of its operation,
- Trap — a trap, for example, a message about a problem [3].

Two more PDUs were introduced in SNMPv2:

- GetBulkrequest — a request to read an entire block of MIB objects,
- InformRequest — enabling exchange of information between various managers [3].

Remote Network MONitoring (RMON) is a standard MIB that is separate but closely related to SNMP. Like SNMP, RMON is an open standard administered by the Internet Engineering Task Force (IETF). The RMON standard is an SNMP MIB definition described in RFC 1757. RMON1, or simply RMON is defined in RFC 2819. An enhanced version, referred to as RMON2, is defined in RFC 2021 [4]. Unlike SNMP, which should send requests for information, RMON can configure signals that will "monitor" the network based on a specific criterion. RMON provides administrators with the ability to manage local networks as well as remote from one specific location or point. Its network layer monitors are shown below. RMON2 improves RMON1 by providing network-and-application level statistics[4]. It focuses on Internet Protocol (IP) traffic and application layer traffic. The two components of RMON are a sensor, also known as an agent or monitor, and a client, also known as a management station (management station). Unlike SNMP, a sensor or RMON agent collects and stores network information. A sensor is software embedded in a network device (such as a router or switch). The sensor can also

be run on a personal computer. The sensor must be placed for each different segment of the local or global network, since they are able to see traffic that passes only through their channels, but they are not aware of the traffic outside their borders. A client is usually a management station that is connected to a sensor that uses SNMP to receive and correct RMON data.

Netflow is an extension that was introduced on Cisco routers that provide the ability to collect IP network traffic, if specified in the interface. By analyzing the data provided by Netflow, the network administrator can determine such things as: the source and destination of traffic, class of service, and reasons for overcrowding. Netflow includes 3 components: FlowCaching (caching stream), FlowCollector (collector of information about flows) and Data Analyzer.

The Distributed Management Task Force, Inc. (DMTF) is an organization devoted to the development, unification and implementation of standards, initiatives and technologies for the Internet. DMTF creates open manageability standards spanning diverse emerging and traditional IT infrastructures including cloud, virtualization, network, servers and storage. Member companies and alliance partners worldwide collaborate on standards to improve the interoperable management of information technologies [5]. Several standards have evolved as a result of the activities of the DMTF. These include: Web-Based Enterprise Management (WBEM), Common Information Model (CIM) and others.

WBEM is a set of systems management technologies developed to unify the management of distributed computing environments. WBEM is based on Internet standards and DMTF open standards: CIM infrastructure and schema, CIM-XML, CIM operations over HyperText Transfer Protocol (HTTP), and WS-Management [6]. WBEM comprises a set of systems-management technologies developed to unify the management of distributed computing environments. The WBEM initiative, initially sponsored in 1996 by BMC Software, Cisco Systems, Compaq Computer, Intel, and Microsoft, is now widely adopted. WBEM is based on Internet standards and DMTF open standards. WBEM allows the management of any element in a standard and inter-operable manner. WBEM has many implementations and wide support in different operation systems. The data structure

in WBEM is based on the CIM, which implements an object-oriented approach to the presentation of system components. CIM is an extensible model that allows programs, systems and drivers to add their classes, objects, methods and properties to it.

Windows Management Instrumentation (WMI) is Microsoft's implementation of CIM that's produced by DMTF. The CIM (and WMI) defines a series of classes that supply information about Windows systems, and they may allow you to directly interact with aspects of local and remote systems. It is one of the core technologies for centralized management and monitoring of various parts of the computer infrastructure running the Windows platform [7].

Since WMI is built on an object-oriented basis, all data of the operating system is presented in the form of objects and their properties and methods. Class instances can generate events for subscription. When an event occurs, WMI automatically creates an instance of the class to which this event corresponds. It is convenient to use such a mechanism to execute a certain command when a certain event occurs, that is, monitor the state of objects of the operating system.

The Java Management API (JMAPI) is a collection of Java classes, provided by Sun Microsystems, that allow network management software developers to write management applications using a standardized platform-independent application programming interface. It is a toolkit for building network and application administration systems. JMAPI is not a network or system management product and by itself cannot provide any management functionality. Rather, it is designed to be used by developers of network management systems and also by network element hardware vendors [8]. SNMP support enables JMAPI to be integrated with other management systems, such as the Tivoli Management Environment (TME).

In a network composed of devices of various manufacturers and generations, the greatest difficulty is the integration of all elements into a single monitoring and control system. The reason is that different devices support different control protocols and their versions, provide different interfaces, or may not have any functionality at all. Therefore, when creating and upgrading a network, it is recommended to maintain uniformity of devices

on the control interface.

2.2 Network Topology and Options

The whole variety of computer networks can be classified according to the following four criteria: by the type of transmission medium, that is, the physical medium that is used to connect computers; by information transfer speed; by departmental affiliation; by territorial prevalence.

The transmission medium is also called a “communication line”. Information is transmitted over communication lines in the form of various signals, which, when facing the resistance of the medium, decay with distance. Therefore, one of the most important characteristics of a communication line is the maximum range to which information can be transmitted through it without distortion. As communication lines can be used: IR rays (provide the transfer of information between computers located within the same room); electrical wires (twisted pair cable provides communication between computers at a distance of up to 100m, coaxial cables — up to 500m); fiber optic cables (provide communication over a distance of several hundred kilometers); telephone lines, radio communications, satellite communications (allow you to connect computers located anywhere in the world).

By the speed of information transfer, computer networks are divided into low-speed (information transfer speed up to 10 Mbit/s), medium-speed (information transfer speed up to 100 Mbit/s), high-speed (information transfer speed over 100 Mbit/s).

By affiliation distinguish departmental and state networks. Departmental networks belong to one organization and are located on its territory. State networks are networks used in government structures.

By territorial distribution, networks can be local, global and regional. Local networks are networks located in one or more buildings. Regional networks are those located in the city or region. Global networks are networks located on the territory of a state or a group of states, for example, the Internet.

In the classification of networks, there are two main terms: a local network and a geographically distributed network.

2.2.1 Local and wide area networks

A local area network connects computers and printers, usually located in the same building (or complex of buildings). Each computer connected to the local network is called a workstation or network node. As a rule, in local networks, the use of high-speed channels is practiced. Local networks allow individual users to easily and quickly interact with each other. Here are just some of the tasks that a local network allows you to do: collaborate on documents, transfer files between computers, simplify workflow, save and archive your work on the server, easy access to applications on the server, facilitate the sharing of expensive technical resources.

Local area networks are divided into two radically different classes: peer-to-peer (single-level or Peer to Peer) networks and hierarchical (multi-level). A peer-to-peer network is a network of peer computers. Each node can use the resources of other nodes and provide them with access to their own resources. An example would be file sharing on client operating systems.

In hierarchical local networks there is one or several special computers — servers, on which information is shared by various users. A server in hierarchical networks is a permanent storage of shared resources. The server itself can only be a client of a server of a higher hierarchy level. Therefore, hierarchical networks are called networks with a dedicated server. Servers are usually high-performance computers, most often with several processors running in parallel, with high-capacity storage, and a high-speed network card. The computers from which information is accessed on the server are called clients.

A wide area network connects several local networks that are geographically distant from each other. Geographically-distributed networks provide the same advantages as local networks, but at the same time they cover a large territory. Usually, for this, the services of network providers are used — organizations involved in the creation, support

and development of commercial networks. The collection of all interconnected networks of the world is called the Internet. For data transmission within the WAN, fiber optic lines, radio communications and satellite communications are mainly used. To connect subscribers to the provider's network, in addition to the same technologies, coaxial cables, telephone lines, cellular communications, and twisted-pair cable are used.

2.2.2 Network topology

Network topology is the structure of connections in a network and may be depicted physically or logically. It is an application of graph theory where communicating devices are modeled as nodes and the connections between the devices are modeled as links or lines between the nodes. Physical topology is the placement of the various components of a network (e.g., device location and cable installation), while logical topology illustrates how data flows within a network. Distances between nodes, physical interconnections, transmission rates, or signal types may differ between two different networks, yet their topologies may be identical. A network's physical topology is a particular concern of the physical layer of the Open Systems Interconnection model (OSI) model. Favorite network topologies: Bus, Ring, Mesh, Star, Fully connected.

Virtual LANs, or VLANs, are virtual separations within a switch that provide distinct logical LANs that each behave as if they were configured on a separate physical switch. Before the introduction of VLANs, one switch could serve only one LAN. VLANs enabled a single switch to serve multiple LANs. Assuming no vulnerabilities exist in the switch's operating system, there should be no way for a frame that originates on one VLAN to make its way to another [9]. This increases network security, reduces the negative impact of broadcast traffic on throughput, and simplifies support and administration.

The VLAN topology typically includes one or more routers that have an interface in this VLAN and route traffic from devices in this VLAN and to them. In addition, VLAN includes connected end devices and intermediate devices configured for its use, as well as connections between them. For the devices to work correctly in VLAN, all connected

network components must be configured correctly. VLAN supports various types of traffic (tagged, untagged).

In spite of the wide prevalence, VLAN configuration remains a tedious and complex process. On one hand, VLAN configuration is complex, because of the size and complexity of today's enterprise networks (some of them even surpass those of carrier networks), and also because of the network-wide dependencies that are inherent to VLAN design. For example, a simple configuration such as adding a new host to a VLAN may require modifying the configuration of multiple switches in the network (a process called configuring the "trunk" links) [10].

Information about VLAN spread can be provided in different forms: textual as when outputting the configuration via the command line, in the form of a table, through an interactive interface that requires entering the VLAN number and selecting a device (and, possibly, port). However, presenting VLAN information on top of a general network scheme may be most useful for operational work and troubleshooting.

2.2.3 Network characteristics

The main requirement for networks is that they perform their main function — providing users with the potential access to shared resources of all computers connected to the network. All other requirements are related to the quality of this basic task.

The quality of the network is characterized by the following properties: performance, reliability, compatibility, controllability, security, extensibility, scalability and transparency. Further they will be considered in detail.

There are two main approaches to ensuring network performance. The first is that the network guarantees the user compliance with a certain numerical value of the service quality indicator. For example, frame relay and Asynchronous Transfer Mode (ATM) networks can guarantee the user a given level of throughput. In the second best effort, the network tries to provide the user with the highest quality possible, but does not guarantee anything. The main characteristics of network performance include: response

time, which is defined as the time between the occurrence of a request to a network service and receiving a response to it; bandwidth, which reflects the amount of data transmitted by the network per unit of time, and transmission delay, which is equal to the interval between the moment a packet arrives at the input of a network device and the moment it appears at the output of that device.

To assess the reliability of networks, various characteristics are used, including: availability, which means the fraction of time during which the system can be used; security, that is, the ability of the system to protect resources from unauthorized access; fault tolerance, the ability of a system to work under conditions of failure of some of its elements.

Compatibility means that the network is able to include a wide variety of software and hardware.

Network controllability implies the ability to centrally monitor the state of the main elements of the network, identify and solve problems that arise during the operation of the network, perform a performance analysis and plan the development of the network.

Security depends on its resistance to unauthorized access to the provided resources, as well as to the disconnection of the services provided by the network.

Extensibility means the ability to relatively easily add individual network elements (users, computers, applications, services), increase the length of network segments and replace existing equipment with more powerful one.

Scalability means that the network allows you to increase the number of nodes and the length of the connections in a very wide range, while network performance does not deteriorate.

Transparency — the ability of a network to hide from the user the details of its internal device, thereby simplifying its operation on the network.

2.3 Service Provisioning

Educational institutions provide many IT services for students and teachers. Services can be used directly by the consumer or provide the work of other, higher-level services.

Access to the organization's network is required for most services. Communication, file transfer, electronic document management, newsletters, the use of cloud resources and client-server software — all this is impossible without using a network. In some cases, long duration connections are required (online lecture). In others, short duration is enough (download a file with a description of the task, then download the completed work). Services differ in network bandwidth requirements, latency sensitivity and the need to retransmit lost data. Most modern services are resilient to loss of connection, but the user will receive little comfort due to frequent reconnections. Access to the network can be performed locally (from the territory of the organization) via wired and wireless technologies, or from the Internet using a Virtual Private Network (VPN), publishing services, or placing connection points at the network border.

Authentication, Authorization and Accounting (AAA) are required to control the use of resources provided by the organization. Since wireless technologies are widely used to connect user devices, and a large number of services are available via the Internet, the availability of reliable protection against misuse is especially important. Authentication is performed both when connecting to a network (802.1X, Wi-Fi Protected Access (WPA) or other protocols), and when trying to access a specific resource (for example, authentication on a website using a Google account).

The operation of supporting network services (glsDNS, glsDHCP) is necessary to access the organization's resources and allows the user to focus on the tasks performed instead of, for example, manually entering the IP address on his device when connecting to another segment of the local network.

E-mail is not the most modern, but the usual and universal way to exchange messages and small files. In addition, it is often used for official correspondence. The mail service is characterized by the supported protocols, the permissible size of the mailbox, the maximum size of the letter, the presence of spam protection, and the presence of a web interface.

Cloud storage allows the user to store files and share links to them. Such storage allows you to reliably store files (many times including version control), check for malicious

content, standardize the procedure for posting information, and manage access rights.

The organization's website provides access to relevant information and news, and it is the institution's visiting card on the Internet. Convenience, information and quality of the website seriously affect student decisions in the process of choosing an educational institution. A website is often used as an entry point to other online services of an organization.

The user's personal account helps to keep up to date with the latest information, to get convenient access to useful functions and reports. A personal account is the most convenient way to consolidate all the information intended personally for the user and a set of related options for action. Often, electronic document management and online payment systems are integrated into your personal account, which saves time and resources of both the user and the organization.

The modern educational process uses a large number of specialized software from subject areas. In many cases, you can provide network access to it, or the software is initially focused on network use. This reduces the requirements for the student's computer and helps him focus on studying the functionality of the program or the work performed, and not on the intricacies of installing and administering the program. For students of IT areas, universities often provide digital lab environments that allow you to use cloud computing, create a virtual IT infrastructure, etc. This allows you to significantly increase the useful experience gained and to consolidate the acquired knowledge in practice.

2.4 Tools and Applications

HP Network Management Center (NMC) is a suite of integrated HP software used by network managers in information technology departments. NMC allows network operators to see, catalog and monitor the routers, switches and other devices on their network. It alerts IT staff when a network device fails and predicts when a network node or connection point may go down. The solution is designed to simplify the management of complex, distributed, multi-vendor networks in large enterprise data centers. It provides

process-powered automation to automate the complete operational lifecycle of network devices from provisioning to policy-based change management, compliance and security administration.

IBM Tivoli Monitoring solutions provides a solid foundation for the development of management solutions addressing the complex needs of today's IT infrastructures. A set of modules built on top of IBM Tivoli Monitoring provide a comprehensive set of solutions for companies facing the challenge of monitoring composite application infrastructures [11]. IBM Tivoli Monitoring software tracks the performance and availability of distributed operating systems and applications. The system is built on an agent-server-client architecture. The data collected can be transferred to administrators or used by other Tivoli products (Network Manager, Composite Application Manager, etc). Network Manager topology viewer screenshot below (Figure 2.1).

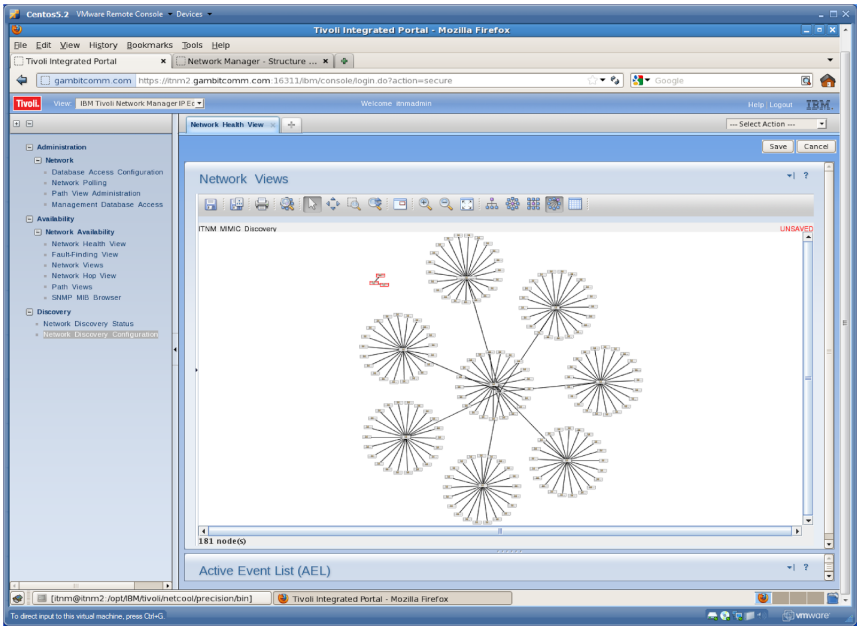

Figure 2.1: IBM Tivoli Network Manager topology viewer screenshot.

SCOM is a component of the Microsoft System Center suite of enterprise management software. SCOM allows data center administrators to deploy, configure, manage and monitor the operations, services, devices and applications of multiple enterprise IT systems through a single pane of glass. SCOM is a cross-platform tool and can work

with Windows, Mac OS and Unix-based operating systems, including Linux. Organizations using SCOM typically rely on management packs developed by third-party vendors to extend its monitoring capability beyond Microsoft workloads. Every enterprise relies on its underlying services and applications for everyday business and user productivity. SCOM is a monitoring and reporting tool that checks the status of various objects defined within the environment, such as server hardware, system services, operating systems, hypervisors and applications. Administrators set up and configure the objects. SCOM then checks the relative health such as packet loss and latency issues of each object and alerts administrators to potential problems. Additionally, SCOM offers possible root causes or corrective action to assist troubleshooting procedures. There is single-server deployment architecture on the picture (Figure 2.2).

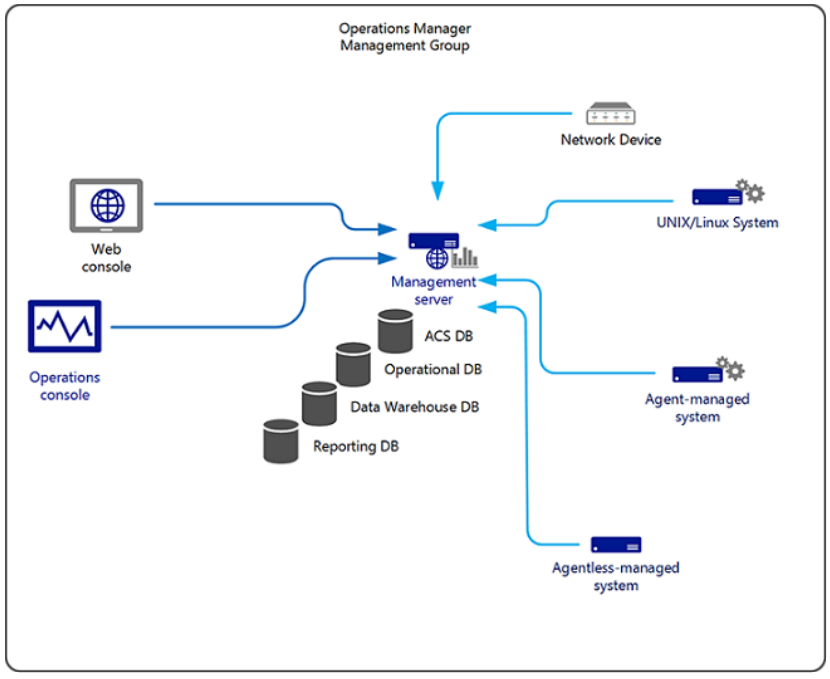

Figure 2.2: SCOM single-server deployment architecture

Micro Focus Network Operations Management (NOM) suite provides market leading management for enterprise networks, across public, managed and private clouds, integrating capabilities to monitor fault, performance, configuration, and compliance of physical, virtual, and Software Defined Network (SDN) infrastructure. The Suite has the broadest

and deepest multi-vendor support beyond simple SNMP/ICMP monitoring. It has the following features:

- Network Fault Management
- Network Performance Management
- Network Configuration Management
- Configuration Audit
- Configuration Policy Audit
- Network Auditing and Reporting
- On Demand Network Notification

OpenNMS is an enterprise-grade, integrated, open-source platform to build network monitoring solutions. Goals include accelerating time to production by supporting industry standard network management protocols, agents, and a programmable provisioning system [12]. The platform is developed and supported by a community of users and developers and by the OpenNMS Group, offering commercial services, training and support. The software architecture is on the scheme (Figure 2.3). The goal is for OpenNMS to be a truly distributed, scalable management application platform for all aspects of the FCAPS network management model while remaining 100% free and open source. All code associated with the project is available under the Affero General Public License.

Zabbix is an open-source monitoring software tool for diverse IT components, including networks, servers, virtual machines and cloud services. Zabbix provides monitoring metrics, among others network utilization, CPU load and disk space consumption. Zabbix monitoring configuration can be done using XML based templates which contain elements to monitor. Zabbix offers several monitoring options, such as simple checks can verify the availability and responsiveness of standard services such as Simple Mail Transfer Protocol (SMTP) or HTTP without installing any software on the monitored host (Figure 2.4).

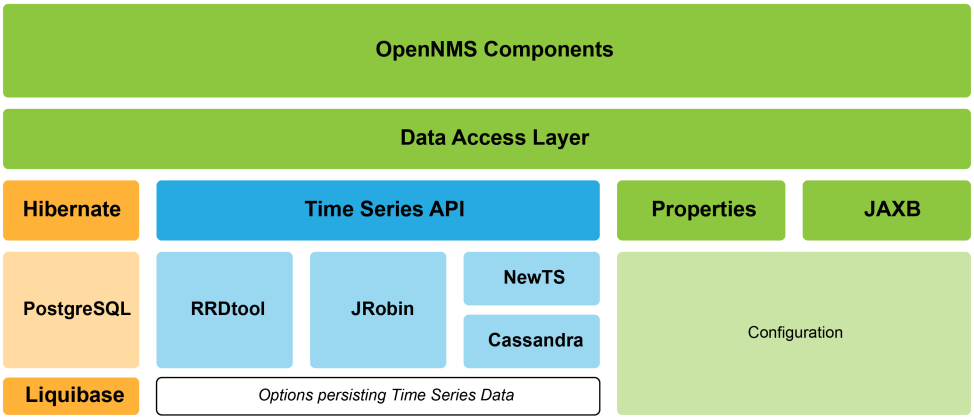

Figure 2.3: OpenNMS software architecture

It can be characterized as a semi-distributed monitoring system with centralized management. While many installations have a single central database, it is possible to use distributed monitoring with nodes and proxies, and most installations will use Zabbix agents [13].

A Zabbix agent can also be installed on Unix and Windows hosts to monitor statistics such as CPU load, network utilization, disk space, etc. As an alternative to installing an agent on hosts, Zabbix includes support for monitoring via SNMP, Transmission Control Protocol (TCP) and ICMP checks, as well as over Intelligent Platform Management Interface (IPMI), Java Management Extensions (JMX), Secure Shell (SSH), Telnet and using custom parameters. Zabbix supports a variety of near-real-time notification mechanisms.

Cisco Prime Infrastructure is a network management tool that supports lifecycle management of an entire network infrastructure from one graphical interface. Prime Infrastructure provides network administrators with a single solution for provisioning, monitoring, optimizing, and troubleshooting both wired and wireless devices. Robust graphical interfaces make device deployments and operations simple and cost-effective. It is a very suitable solution for Cisco-based network because it has wide functionality out-of-box, although it can be an expensive solution (Figure 2.5).

There are several integrated network management solutions available, both commercial and open source. Most are full featured and provide graphical or web based user interface.

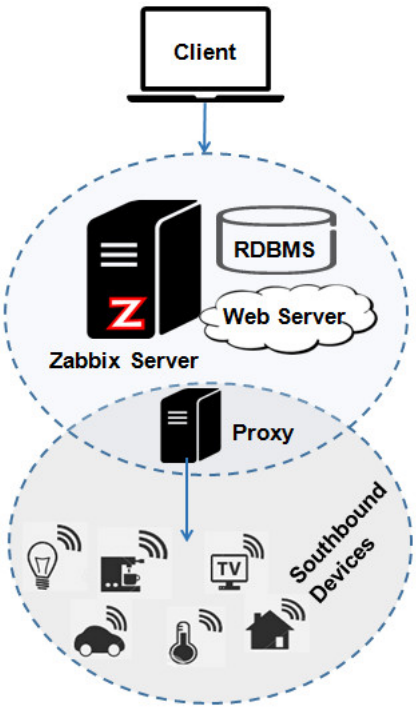

Figure 2.4: Zabbix architecture.

Figure 2.5: Cisco Prime Infrastructure user interface

Nevertheless, in a complex and a constantly updated field, it is important to be able to adjust, develop or complement the applications with other tools. For that, APIs and technolgies should be used, to facilitate the development and provide an adequate programming environment.

2.5 APIs and Technologies

ASP.NET (Active Server Pages for .NET) is a web application development platform that includes web services, software infrastructure, programming model from Microsoft. ASP.NET is part of the .NET Framework and is a development of older Microsoft ASP technology. Because ASP.NET is based on the Common Language Runtime (CLR), which is the foundation of all Microsoft .NET applications, developers can write code for ASP.NET using the programming languages included with the .NET Framework (C#, Visual Basic.NET, J# and JScript .NET). The ASP.NET programming model is based on the HTTP protocol and uses its rules of interaction between the server and the browser. ASP.NET supports several programming models for building web applications, including ASP.NET MVC.

ASP.NET Core is a free, open source, cross-platform framework for building open source web applications. This platform is being developed by Microsoft together with the community and has greater performance compared to ASP.NET. It has a modular structure and is compatible with operating systems such as Windows, Linux and Mac OS. Although this is a new framework built on the new web stack, it has a high degree of concept compatibility with ASP.NET.

Model-View-Controller (MVC) is a scheme for dividing application data, user interface and control logic into three separate components: model, view and controller, so that each component can be modified independently. The model provides data and responds to controller commands, changing its state. The view is responsible for displaying model data to the user, responding to model changes. The controller interprets user actions, notifying the model of the need for changes.

ASP.NET MVC Framework is a framework for creating web applications that implements the Model-view-controller template. The view engine is used to control markup and code insertion in a view. Starting with MVC 5, the only engine built in by default is Razor. The basis of Razor syntax is the "@" sign, after which the transition to code in C#/VB.NET languages is carried out. The use of third-party engines is also possible. Presentation files are not standard static pages with HyperText Markup Language (HTML) code, but when the controller generates a response using representations, they are compiled into classes, from which the HTML page is then generated. When processing requests, the ASP.NET MVC framework relies on a routing system that maps all incoming requests to routes defined in the system that indicate which controller and method should process this request. The default built-in route assumes a three-tier structure: controller / action / parameter.

Web API is a different way to build an ASP.NET application, somewhat different from ASP.NET MVC. Web API is a web service that can interact with various applications. At the same time, the application can be an ASP.NET web application, or it can be a mobile or regular desktop application. ASP.NET Web API is an extensible framework for creating HTTP-based services that can be accessed in various applications on different platforms, such as web, Windows, mobile devices, etc. It works more or less the same way an ASP.NET MVC web application, except that it sends data as a response instead of presenting HTML.

ASP.NET Core supports the Open Web Interface for .NET (OWIN) that allows web apps to be decoupled from web servers. It defines a standard way for middleware to be used in a pipeline to handle requests and associated responses. ASP.NET Core applications and middleware can interoperate with OWIN-based applications, servers, and middleware.

Bootstrap (also known as Twitter Bootstrap) is a free set of tools for creating websites and web applications. Includes HTML and Cascading Style Sheets (CSS) design templates for typography, web forms, buttons, tags, navigation blocks and other web interface components, including JavaScript extensions. Basic Bootstrap Tools:

- Grids — predefined column sizes
- Templates — a fixed or rubber document template
- Typography — descriptions of fonts, definition of some classes for fonts, such as code, quotes, etc
- Media — Provides some image and video management
- Tables — tables design tools, up to adding sorting functionality
- Forms — classes for the design of forms and some events that occur with them
- Navigation — design classes for panels, tabs, page navigation, menus and toolbars
- Alerts — the design of dialog boxes, tooltips and pop-ups

jQuery is a set of JavaScript functions that focuses on the interaction of JavaScript and HTML. The jQuery library helps to easily access and manipulate any Document Object Model (DOM) element, access attributes and contents of DOM elements. The jQuery library also provides a convenient Application Programming Interface (API) for working with Asynchronous Javascript (AJAX).

JavaScript Object Notation (JSON) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition — December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence

A Comma-Separated Values (CSV) file is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. A CSV file typically stores tabular data (numbers and text) in plain text, in which case each line will have the same number of fields.

Secure Hash Algorithm Version 2 (SHA-2) — a family of cryptographic algorithms — unidirectional hash functions, which includes the algorithms SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256 and SHA-512 / 224. Hash functions are designed to create "fingerprints" or "digests" for messages of arbitrary length. They are used in various applications or components related to information security. After the addition, the original message is divided into blocks, each block into 16 words. The algorithm passes each block of the message through a cycle with 64 or 80 iterations (rounds). At each iteration, 2 words are converted, the remaining words define the conversion function. The processing results of each block are added, the sum is the value of the hash function. However, the initialization of the internal state is performed by processing the previous block.

Microsoft Visual Studio is a Microsoft product line that includes an integrated software development environment and a number of other tools. These products allow you to develop both console applications and GUI applications, including those supporting Windows Forms technology, as well as websites, web applications, and web services in both native and managed codes for all platforms, supported by Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework and Silverlight. Visual Studio includes a source code editor that supports IntelliSense technology and the ability to easily refactor code. The built-in debugger can operate as a source level debugger or as a machine level debugger. Other built-in tools include a form editor to simplify the creation of the application GUI, a web editor, a class designer, and a database schema designer. Visual Studio allows you to create and connect third-party plugins to expand the functionality at almost every level.

Telegram Bot API is an HTTP-based interface created for developers keen on building bots for Telegram. Bots are third-party applications that run inside Telegram. Users can

interact with bots by sending them messages, commands and inline requests. At the core, Telegram Bots are special accounts that do not require an additional phone number to set up. Users can interact with bots in two ways:

- Send messages and commands to bots by opening a chat with them or by adding them to groups. This is useful for chat bots, news bots or agent bot that is used to communicate with another software
- username and a query. This allows sending content from inline bots directly into any chat, group or channel

Messages, commands and requests sent by users are passed to the software running on Telegram servers. An intermediary server handles all encryption and communication with the Telegram API.

WebSocket is a communication protocol over a TCP connection, designed for real-time messaging between the browser and the web server. WebSocket is designed to be implemented in web browsers and web servers, but it can be used for any client or server application. The WebSocket protocol is an independent protocol based on the TCP protocol. It enables closer interaction between the browser and the website, helping to spread interactive content and create real-time applications.

2.6 Summary

In this section, the main concepts related to networking, network management and service provisioning were described. In addition, an approach to the APIs and Technologies to be considered in the development of the NetGlance were also discussed. Next chapter will focus on the identification and description of the IPB and KubSAU networks.

Chapter 3

KubSAU and IPB Networks

This chapter is about KubSAU and IPB computer networks. It contains investigation results, key points comparison, characterization and best practices assessment for these networks.

3.1 KubSAU Network

KubSAU enterprise network is a campus network based on IP and Ethernet technologies. It covers most of the university territory of 174 hectares [14]. It is educational and administrative buildings, 21 dormitories, laboratories, a canteen, auxiliary and technical buildings — 43 buildings summary [15]. In addition, surveillance cameras located outside the buildings and some another devices are connected to the network. The network uses fiber and copper wires and Wi-Fi. The network has approximately 3500 active users in nominal mode and 5000 in peaks.

The network has Core&Distribution and Access level. It is built on mixed topology. The Core&Distribution level is designed as an IP failover ring. Some parts of the network are fully connected (Figure 3.1). The main ring has 2 Gbit/s throughput that is planned be increased to 20 Gbit/s this year. In other places, gigabit technology is mainly used, but some low-traffic parts are 100Mbit/s.

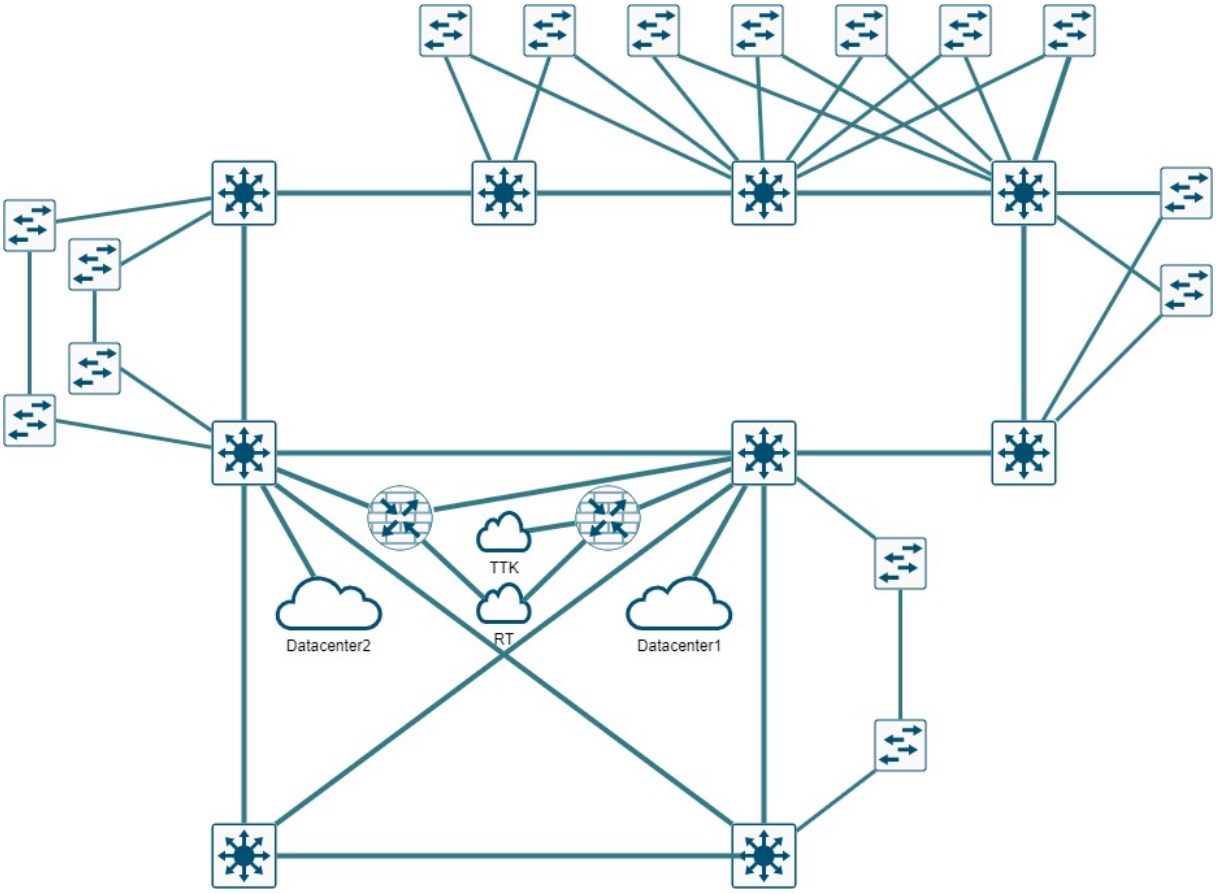

Figure 3.1: KubSAU network topology (Core&Distribution layer)

Desktop teacher workstations and computer labs are connected through a wired network. Students can use a special wireless network deployed in educational buildings and dormitories. There are also wireless networks for teachers and other workers. A third-party company provides paid high-speed wired internet access in dormitories.

The network is connected to the Internet through two Internet providers in two independent points in hot-standby mode. It means that Internet connection is high-available in the campus network. External channels have 700 Mbit/s summary throughput. Internet connection to the organization's internal network resources is implemented using a secure VPN. It is used to connect two branches with the main network and personally by employees for teleworking.

The network provides the following services for students: Internet access, an electronic library, a university website with a personal account, e-mail, as well as virtual machines and various web-based services for education and performing laboratory works. The main web portal is hosted at a commercial cloud outside the network. Other services are hosted at servers mostly in two data centers.

Enhanced Interior Gateway Routing Protocol (EIGRP) is used to provide dynamic routing. The network uses micro segmentation through the wide use of VLAN (more than 200 VLAN for 17 faculties and other departments).

The network uses devices from many manufacturers and a large number of models. At the Core&Distribution level, Cisco Layer 3 switches are used. At the access level, Cisco, HP, Telesis, SNR Layer 2 switches are used. In addition, there are some local SOHO access points in the offices.

LAN perimeter security is ensured by restricting physical access to network devices and connection points, wireless network policies, mac address checking. Students are authenticated in Wi-Fi network using domain personal accounts and Captive Portal.

Restrictions on access to prohibited Internet resources are implemented by the Internet providers in accordance with Russian law. The university further blocks torrent connections.

Main network bottlenecks are channels to Internet providers and some Access-level

switches with FastEthernet upstream ports.

The network management staff is small. The network is maintained by two system administrators and two network installers. There is no dedicated network admin. Large installation operations are carried out together with electricians and builders and, if necessary, third-party installers are involved.

Until March 2019, there was no centralized monitoring system for active network equipment. There was an attempt to deploy Cisco Network Assistant (CNA), but this turned out to be unproductive due to poor support of an equipment from other manufacturers. The implementation of Zabbix for the same purpose turned out to be ineffective due to the complexity of updating the network scheme and its low visibility. To solve this problem, a prototype monitoring system was developed, which allowed to form an accurate network monitoring strategy. Also added functionality for collecting information about VLAN, which turned out to be very useful. However, it was required to create a full-fledged multi-user web solution based on this prototype.

The main network monitoring needs stated by system administrators are checking the availability of intermediate network devices and port up-down status, as well as real-time distribution of VLAN on devices and ports. The main need for a network is updating the infrastructure to 10 Gbit/s equipment and replacing legacy FastEthernet segments with gigabit devices. Improving uninterruptible power supply for the main data center is also needed.

The KubSAU network is a complex system that evolved during a long time. It is structured to provide fail-over and it is productive. But it requires introduction 10 Gbit/s technologies on main traffic channels. Also, the use of long (two or more) chains of unmanaged devices for a permanent network reduces its controllability and can lead to unintended outages of a network branch. It is recommended to reduce the number of SOHO devices used outside of a single room, and to place active network equipment within network boxes when it is possible (use multi-pair cables to connect many devices in one room).

3.2 IPB Network

IPB enterprise network is IP and Ethernet based 3-Layer campus network. The network has approximately 3000 active users in nominal mode.

The network has Core, Distribution and Access level. It is built on mixed topology (Figure 3.2). The Core and Distribution levels have a fully connected topology and 20 Gbit/s throughput. Cluster links are 80 GBit/s. Other links use Gigabit technology.

Teacher workstations and computer labs are connected through wired or wireless network. Students use a special wireless network deployed in all educational buildings. In addition, some service wireless networks are used.

The network is connected to the Internet via two Internet providers through two channels with 2 Gbit/s and 200 Mbit/s throughput in hot-standby mode with load balancing, so Internet connection is high-available in the campus network and both connections can be used simultaneously. The 200 Mbit/s connection is planned to be upgraded to 1 Gbit/s this year. There is a secure VPN connection endpoint for teachers and students. A branch in Mirandela city is connected via dedicated physical channel and a backup tunnel through Internet.

There are many services for students in IPB network: Internet access, an electronic library, a university website with a personal account, e-mail, Learning Management System, cloud storage, virtual machines for education, online payments, academic services, ordering food in the dining room and rent a bike. For administrative tasks, there are also services for class registration, creditation procedures, and others. Web portals are hosted inside the network in a data center.

Dynamic routing is implemented using Open Shortest Path First protocol (OSPF). Approximately 50 VLAN are used for user segmentation.

The network uses few models of active network equipment, maintaining a homogeneous environment. At the Core and Distribution levels, Cisco Layer 3 switches are used. They are integrated into virtual switches to provide fault tolerance while maintaining ease of configuration. However, at the access level, equipment from other manufacturers is also

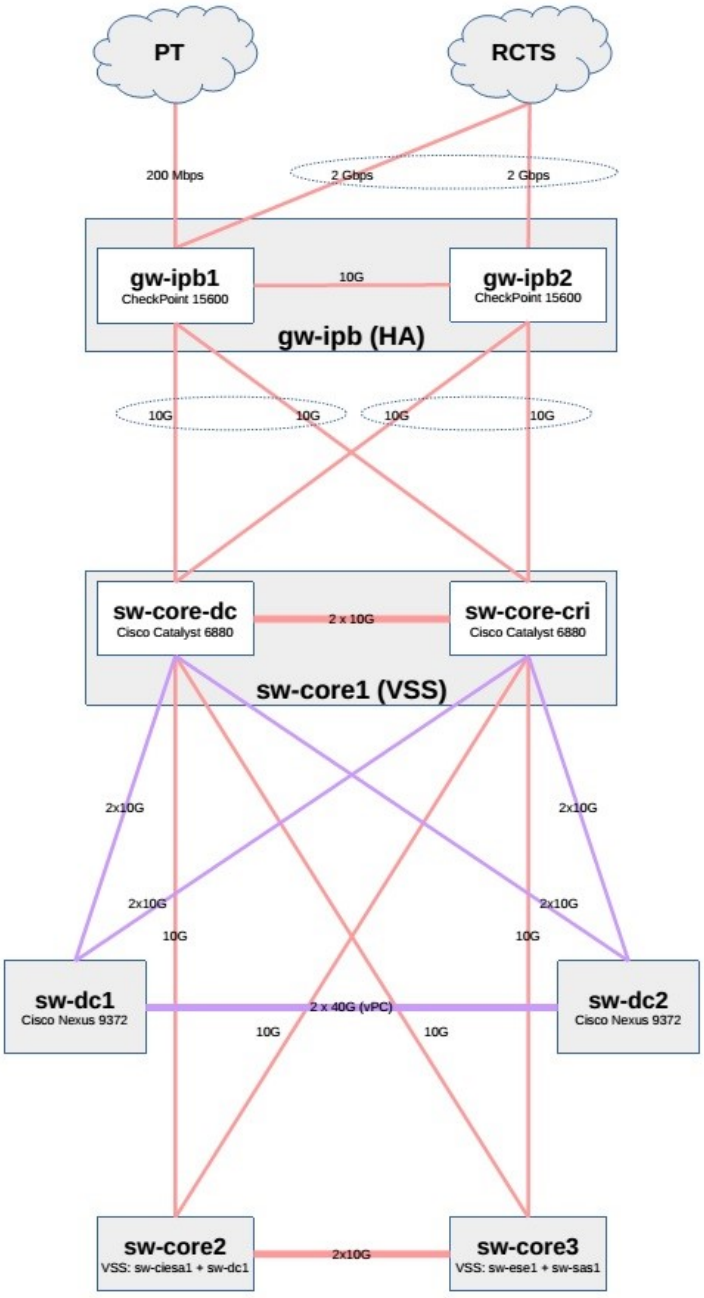

Figure 3.2: IPB network topology — Core layer

used, including unmanaged switches, but their number is small.

LAN perimeter security is ensured by restricting physical access to network devices and connection points, secure wireless networks and IEEE 802.1X. Students are authenticated in Wi-Fi network using their personal accounts in an unified European authentication system for education organizations (Eduroam).

IPB does not specifically restrict access to any sites in the Internet. Torrent connections for students are limited in bandwidth.

There is no known bottlenecks in the network, all channels are quite productive. This is achieved through the use of gigabit and 10 Gbit/s connections, fault-tolerant and productive equipment.

The network staff is very small. The network is maintained by system administrators and two network installers without a dedicated network admin. Large installation works are carried out together with other employees. Network monitoring is carried out using Zabbix and especially Cisco Prime. Many network measurements are monitored, including network devices reachability, port up-down status, security and topology events, channels bandwidth utilization, and more. Wide network statistics is available for analysis for the admins.

The main network monitoring needs are checking the availability of intermediate network devices and port up-down status. In the future IT specialists want to observe user experience for all wireless access points to find out potential bottlenecks. The main need for the network is upgrading some legacy Access-level switches.

IPB has a modern, with fail-over and productive network. But it has some need to upgrade equipment at the Access level, as well as to streamline the cabling system in some areas.

3.3 Discussion

IPB and KubSAU networks are similar in main points: both have big and complex campus network, thousands of users, a little network personnel.

Network in KubSAU is larger and more difficult in maintenance because of using active network equipment from many vendors. In addition, it has very complex VLAN infrastructure (micro-segmented network). It has insufficient bandwidth on Core&Distribution layer and will be updated this year.

The network in IPB is modern and has enough bandwidth. It is monitored using a solution from a vendor (Cisco) and an open-source system (Zabbix).

Both networks cope well with the current needs of organizations' IT infrastructure. However, how do they deal with unforeseen needs? The situation with COVID-19 created the following: the entire educational process was urgently transferred to a remote format. The load on the corresponding services has sharply increased, traffic between servers and the Internet has increased significantly. The IPB infrastructure quickly adapted using its education management system. At the KubSAU, the educational process initially faced the lack of a centralized education management platform. During the first month of quarantine, appropriate solutions were developed and implemented. In addition, this year a significant increase in the number of applications from applicants submitted online is forecasted. At KubSAU, admission documents will only be accepted online. This means a heavy load on the network, server and data storage. In addition, this places great demands on the reliability of the network — the systems unavailability during the work of the selection committee is unacceptable.

Bottlenecks with increasing activity in the use of services may arise in the KubSAU network. For channels on the Internet, this is not so critical within 1 Gbit/s per channel: agreements with providers provide for a temporary increase in bandwidth during peak loads, and to increase the channel width constantly, it is enough to renew the agreement on new conditions. To increase the bandwidth more, software and/or hardware changes on the equipment of the university and provider will be required. For the further development of the internal infrastructure of KubSAU (primarily, video surveillance systems), a transition to 10 gigabit channels at the core and distribution levels will be required. In the IPB network, such changes are not required, since the use of 10 and 20 Gbit/s channels with a margin covers the needs of the organization. The planned Internet channels

bandwidth increase will keep up to date and provide quick access to the Internet from the internal network, as well as to the organization's services from the Internet.

Both organizations are moving towards increasing the number of services for students and teachers. The responsibility placed on the availability and proper operation of these services is also growing — this is the way of the whole modern world. However, these factors lead to the fact that the well-being of the organization and individuals depends on the stability and security of the IT infrastructure. The more potentially vulnerable interfaces and various devices on the network there are, the more potential attack points for attackers. Therefore, it is so important to take care of the security of each element of the network, whether it is a server, an authorized client device, or network equipment.

3.4 Best practices in KubSAU and IPB

KubSAU uses physically remote server room for backups and backup Internet provider connection point. The network is micro-segmented using many VLAN that means greater security in case of an insecure resource or the spread of a network worm like WannaCry. All workstations are managed using Active Directory for remote monitoring, centralized security policies deployment, secure remote assistance from technical support.

The IPB network is built on 10 Gigabit and 40 Gigabit technologies on Core and Distribution layers and Gigabit on Access level. It means that the network is productive and modern. The number of device models used is small, which maintains the uniformity of the network infrastructure, simplifies management and monitoring. Staff authentication in the wireless network uses unified European system, which requires that the local authentication system integrates with the European counterpart, so administrators have an additional interface to keep — with the national scientific and computation foundation. IPB has more services for students (cloud file storage, web-based education management system). Deep network monitoring and statistics is performed using powerful network monitoring systems.

3.5 Summary

Both networks analyzed are interesting examples of the implementation of the campus network in educational organizations. They have similar goals: maintaining fault tolerance while minimizing the domain of failure, traffic distribution when connecting a large number of geographically distributed objects, various types of traffic both by source (administrative staff and teachers, students) and by purpose (within the network or outside). Both networks are designed to use the organization's resources from the Internet.

However, the solutions used are significantly different. IPB tries to use the minimum number of manufacturers and models of network devices to maintain network homogeneity. KubSAU, on the contrary, uses many different models of devices that are most suitable for specific purposes. IPB uses clusters as the primary way to ensure fail-over at the Core layer. KubSAU prefers to use stacking. IPB has the main site in its own data center, and KubSAU uses a commercial cloud.

Consideration of various approaches to solving the same problems (and, more importantly, the results of the implementation of these approaches) allows to choose the optimal network development strategies. KubSAU requires the implementation of a system for collecting information about a local network for continuous monitoring of the operation of network devices and connections, as well as for obtaining information about VLAN topology at the request of network administrators.

Chapter 4

NetGlance NMS

The main contribution of the work described in this document is the development of a Network Monitoring system, that would be able to be used in relatively large and heterogeneous networks as the ones described before. The application, called NetGlance NMS, is a web-based network management application, aiming at facilitating the task of system and network administrators.

4.1 General description

The solution is a multi-user web application that provides editable real-time network schemes and monitoring settings. It is developed as two-component information system and is better described by its functional and non-functional requirements.

4.1.1 Functional requirements

The solution should display a network scheme showing network devices with some information (name, address, model), as well as the connections between them. The user should be able to add elements to the scheme, delete and modify them, move devices on the diagram. Also, the scheme must support zooming and panning, like geographic online maps. Icons for different types of devices should vary. There could potentially be more

than one network scheme.

The system should display the current state of the network elements according to the specified parameters. Information should be updated approximately in real time. Different colors are used to display states. The user can select the type of information that he wants to display, as well as specify a query parameter, if necessary. In addition, the system should display a brief summary of the latest network events in text form.

The program should have the flexibility to support devices that differ in functionality, supported protocols and interface, including unmanaged devices, accessible only via ICMP, as well as SNMP of all versions. It should be possible to add other protocols. In addition, the program should be able to receive information about devices from a third-party data source (centralized controller). The device polling module code must be available through the user interface. Procedures for obtaining information should contain a minimum amount of infrastructure code. The program should support C# programming language with accessing third-party libraries from code. You also need the ability to cache service information for reuse, both for polling the same device and all devices of this type (for example, store a single connection to the controller to obtain information about all its devices).

Information obtained from the polling should be interpreted transparently for the user. The returned result is checked using a list of templates, each of which corresponds to a certain state. In total, the program should have 7 states: unknown, error, warning, normal, and 3 user-defined. Template lists should be editable by the user.

The system should be able to send notifications of network events to a messenger. This process should be flexible: a short-term change in the state of an object should not cause a message if it did not happen too often.

The system must support various groups of devices that are not based on their model. For these groups, permanent monitored parameters are configured. It is possible to specify whether to send status notifications for these parameters or not.

Network events should be logged. A log should also be kept of the system and user

actions (adding, modifying, deleting elements of network elements and other entities available to the user).

4.1.2 Non-functional requirements

The solution should be implemented as a web application and support the simultaneous operation of several users. The solution should not require special browser settings or plugins. The primary supported browser is the current version of Google Chrome for desktop, but compatibility with other desktop browsers is desired. There should be support for mobile devices.

All the basic actions in the program should be simple and understandable for the user. The convenience of working with a network scheme is especially important.

The system should provide only authenticated access. Authentication is carried out using login and password. Cookies are not used. Passwords should be stored in a secure format that does not allow restoration of their original form or hash selection. It is also necessary to prevent brute force password guessing.

The application must use HyperText Transfer Protocol Secure (HTTPS) to interact with clients. Network traffic should be small and resistant to transmission delays for the convenience of using the system via a potentially slow client's connection.

The program should be protected from errors while user code executed, such as uncaught exceptions and very long execution. These errors should not significantly reduce system performance. It is also necessary to provide protection against too frequent polling of devices as a result of incorrect configuration of the program.

The solution must be flexible, all important parameters should be easily accessible for configuration in an understandable way. Backing up network schemes and other program settings should be simple.

The deployment environment is the Windows Server 2012 R2 Standard operating system. However, tight binding to the operating system used is undesirable for portability.

The solution should not require the installation of additional server roles or other programs, and also require the presence of any services on the network like a Data Base Management System (DBMS). The system deployment format is one or more Windows services.

The solution must be deployed with minimal privileges both on the operating system and on the network. The program should not have administrative authority and access to files outside its own directory. For polling devices, read-only profiles must be used.

4.2 Analysis and Architecture

The solution is developed as two-component system. It contains an application part and a front-end server. Neither component require a web server and they are implemented as Windows services. The system is deployed on one of KubSAU management servers at Windows Server 2012 R2 operating system (Figure 4.1).

There is a sequence diagram for one user's action — open the main application page (Figure 4.2).

Some usage cases will be described below.

Use case 1: Routine Monitoring. The organization's system administrator aims to evaluate the network state at the beginning of the working day. The user opens the web application main page in a browser. A default network map is displayed and a predefined set of monitored parameters is displayed. In this case, the default set is the network devices availability and the link endpoints up-down state. The user examines the network scheme, paying attention to the colors of the elements and their signatures, if necessary, holding the cursor over them to display pop-up hints. It moves and scales the scheme for ease of use. He also reads a summary of recent events on the network. Based on all this information, he assessed the need for further network diagnostics. Using the high visibility of the information, he can determine the location of inaccessible network segments (if any) and make assumptions about the most likely reasons of the failure.

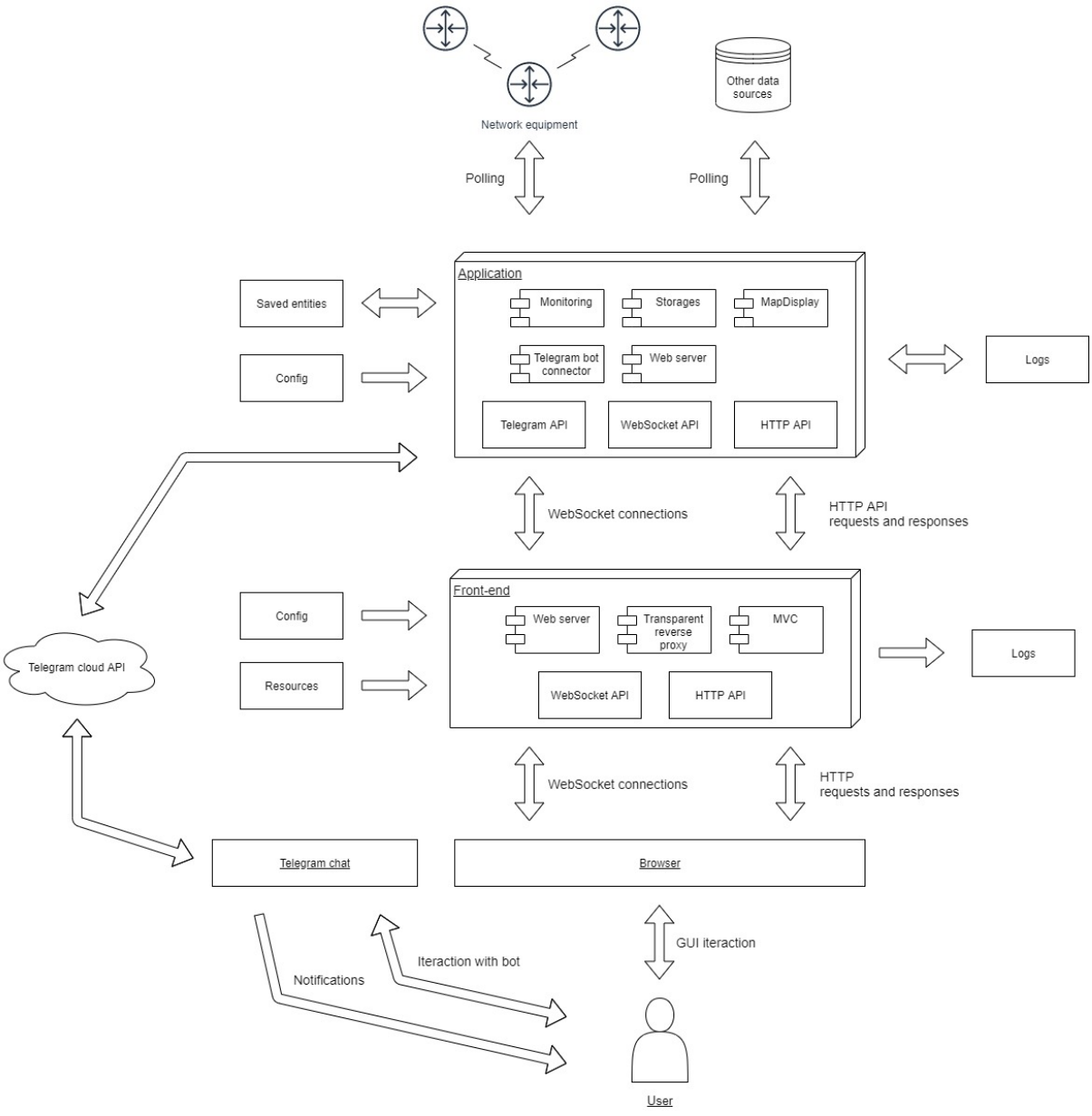

Figure 4.1: NetGlance architecture

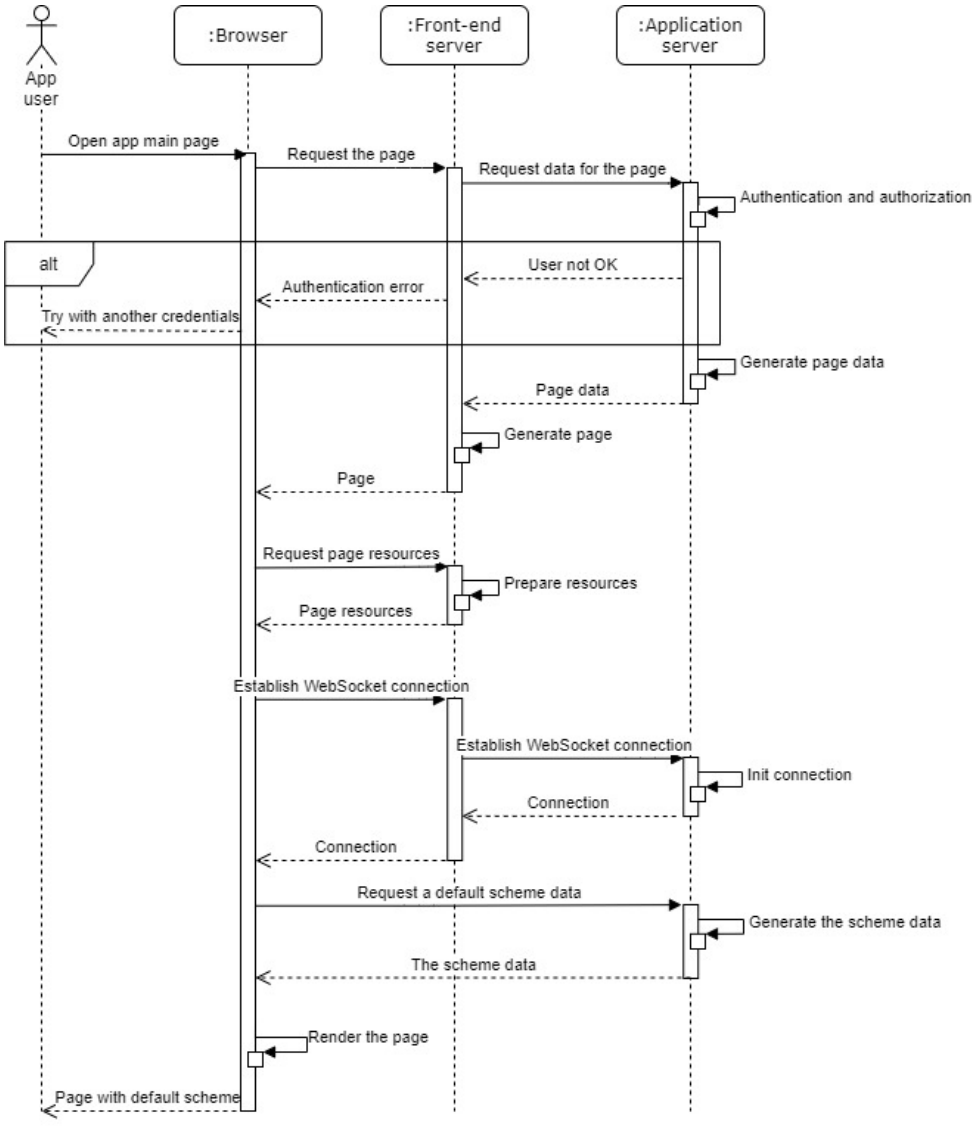

Figure 4.2: Sequence diagram — opening the main page

Use case 2: Unexpected failure. The organization's system administrator gets to work before the start of the working day. He receives a messenger notification on his smartphone that two important routers became unavailable 5 minutes ago, and that the ports on other devices connected to them went down. To better understand the extent of the problem, the administrator activates a VPN connection with the organization's network on the smartphone and opens the web application page. He sees more detailed information about the state of the network in almost real time and has the ability to plan actions to restore the network even before he reaches his office.

Use case 3: VLAN topology changes. The administrator needs to quickly forward a VLAN over the network from the distribution router through a series of switches to an end device. He must create the VLAN on those devices where it does not exist, and correctly configure upstream and downstream ports. However, after completing the configuration, the end device cannot communicate with the rest of the network. The administrator opens the web application, indicates the observed parameter type (VLAN) and its number. After 1 period of polling devices (approximately 20 seconds), he receives the VLAN status on devices (exists or not) and connections (tagged, untagged, absent, not applicable). The administrator saved time on finding a configuration error.

Use case 4: new devices. Several new managed switches were added to the network that are already in use on the network. This model is already in use on the network. They must be added to the monitoring system. The administrator in a few clicks tells the system to add a device. After specifying the connection address and authentication data, the system adds the device to the scheme, automatically receiving information about the name and ports. The device monitoring starts immediately according to the settings for this model and the selected device group. After that, the administrator adds the necessary connections to the device to the scheme, moves the devices for beautiful displaying.

4.3 NetGlance

The solution is a two-component system consisting of an application part and a front-end part. They communicate with each other using a web API. This approach was chosen to reduce the connectivity between task sets: directly monitoring and providing a web interface to the user. This allows to simplify the support of the solution, as well as to update the front-end without stopping monitoring. In addition, it allows to potentially place parts on different servers (now this is not used, but may be useful in the future).

4.3.1 Application

The main component is a monitoring application. It is built on ASP.NET WebAPI using .NET Framework 4.7 and self-hosted using Microsoft.Owin technology. It keeps various system entities (e.g. network schemes, monitoring groups, polling functions for different device types). Using all this information, it polls devices to collect all information that is necessary at the current moment. It discovers monitoring events for devices and device interfaces using compassion between current state and last saved state. If some monitoring event is discovered, the application records in a network log it and notifies subscribers if necessary.

The application also records its own logs to register internal events (start/stop, information about changes made by users, authorization events). It has various log levels for normal use and debug.

The program is managed using a configuration file that stores many settings to optimize its performance in a specific environment (polling settings, log level, data files location, buffering output information, etc). It is useful to tweak the application without re-compiling the assembly.

There is a reduced class diagram for Application entity storage system. All storage classes are inherited from base class — generic storage that implements “IAppModule” interface (Figure 4.3).

The application provides a web API to communicate with end clients and front-end

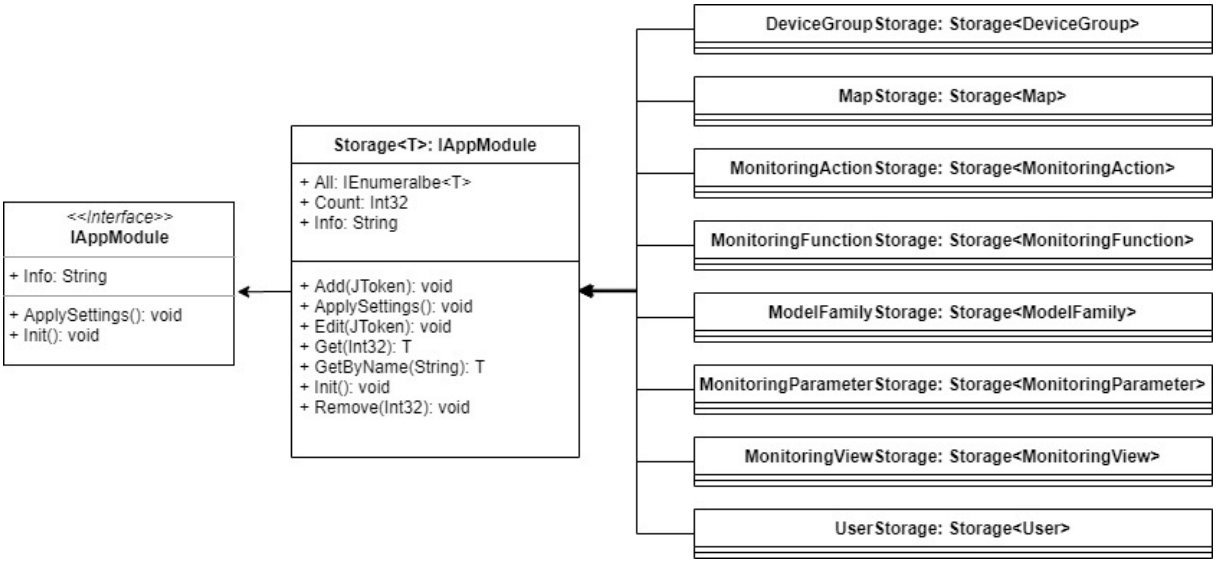

Figure 4.3: Class diagram — Application entity storage system

server. Also it provides some service information for user interface. Only authenticated and authorized queries are allowed. The API is provided using HTTP and WebSocket protocols. A HTTP API is stateless. It provides an access to application entities (list, read, add, edit, remove functionality). A WebSocket API is stateful. It implements subscribe pattern for network schemes. It distributes scheme updates and device/link state updates for connected clients.

4.3.2 Front-end

Another component is a web server that provides user interface. It provides web pages and a transparent reverse proxy for API requests and WebSocket connections. It uses Application web API as a data source for interaction with user including authentication and authorization.

The front-end server is built on ASP.NET Core MVC 3.1 and self-hosted with Kestrel web server implementation. Web pages are built using Bootstrap 4.1 CSS framework, jQuery JavaScript framework. All pages are dynamic and implemented as Views using ASP.NET Core MVC framework.

The web part is intended for using in latest versions of Chromium-based browsers,

but it also support latest Firefox browser with some insignificant interface defects. In addition, it supports mobile browsers Chrome for Android and Firefox Mobile, but devices in a network scheme cannot be moved. It is enough to view the network state outside an office using a mobile device.

4.4 Technical solutions

From the user's point of view, the main functionality is implemented as an editable real-time graphical network scheme. The solution uses active device polling for retrieving data from monitored network nodes. There is an ability to receive information on demand using parameterized queries. The application uses monitoring groups to manage sets of monitored parameters for every monitored device. An universality for different device types is reached using encapsulation of obtaining the same type of information from various types of devices — "MonitoringAction" application entity. It can have different implementations for different device types and retrieve the same information in different ways: using ICMP, SNMP, HTTP or other protocol, polling central network controller, watching syslog data, etc. Information about network interfaces is obtained by a single request to the device. A sequence diagram for requesting data from a source is presented on Figure 4.4.

4.5 Data store

The system uses file-based data store. Now this is enough to fulfill the user's requirements. In addition, it facilitates backup and other data operations.

The solution stores application entities in separate files using JSON format. File names are formatted to simplify human reading.

The system uses CSV format for network logs. This makes it easier for a human to read the file, but retains all the possibilities of automatic log analysis if necessary.

The application log has an informative role for human, so it is presented in plain

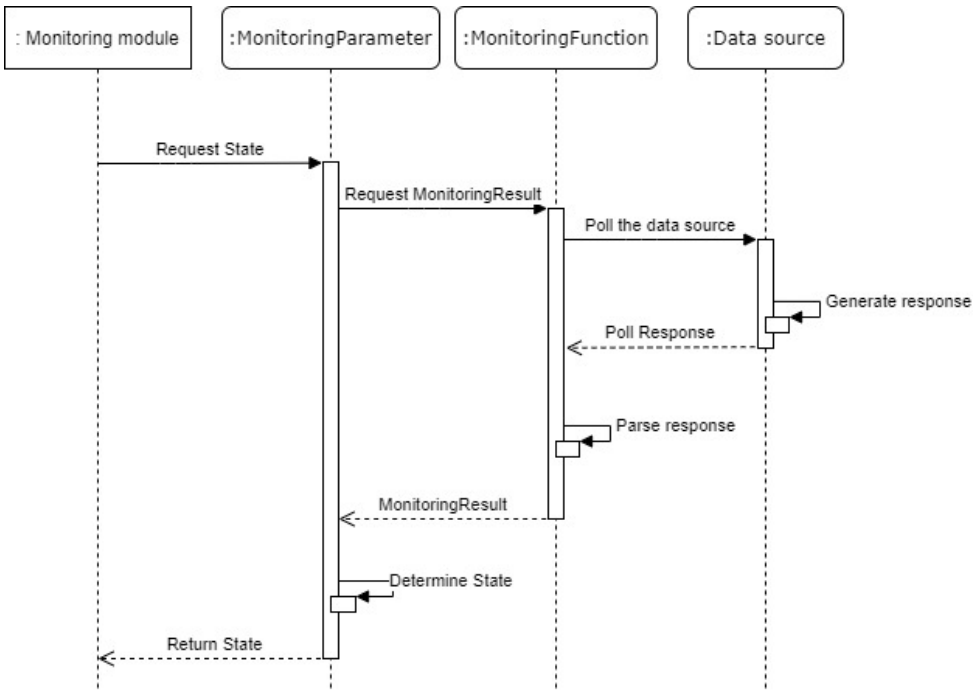

Figure 4.4: Sequence diagram — retrieving information about network element(s)

text. However, the log records are divided line by line and follow a certain format. The application log is intended for debugging and potential analysis of security incidents.

The front-end also stores resource files, such as a certificate, static web files (styles and scripts), as well as a file describing the changes in the current version.

The data store diagram is represented on Figure 4.5.

4.6 Libraries and plug-ins

These are server's side libraries and plugins that are not part of the base ASP.NET project:

- Lextm.SharpSnmpLib, Authors: Malcolm Crowe,Lex Li,and other contributors, License: MIT. Used for communication with devices that support SNMP.

- Newtonsoft.Json, Authors: James Newton-King, License: MIT. Standard library for JSON formatting.

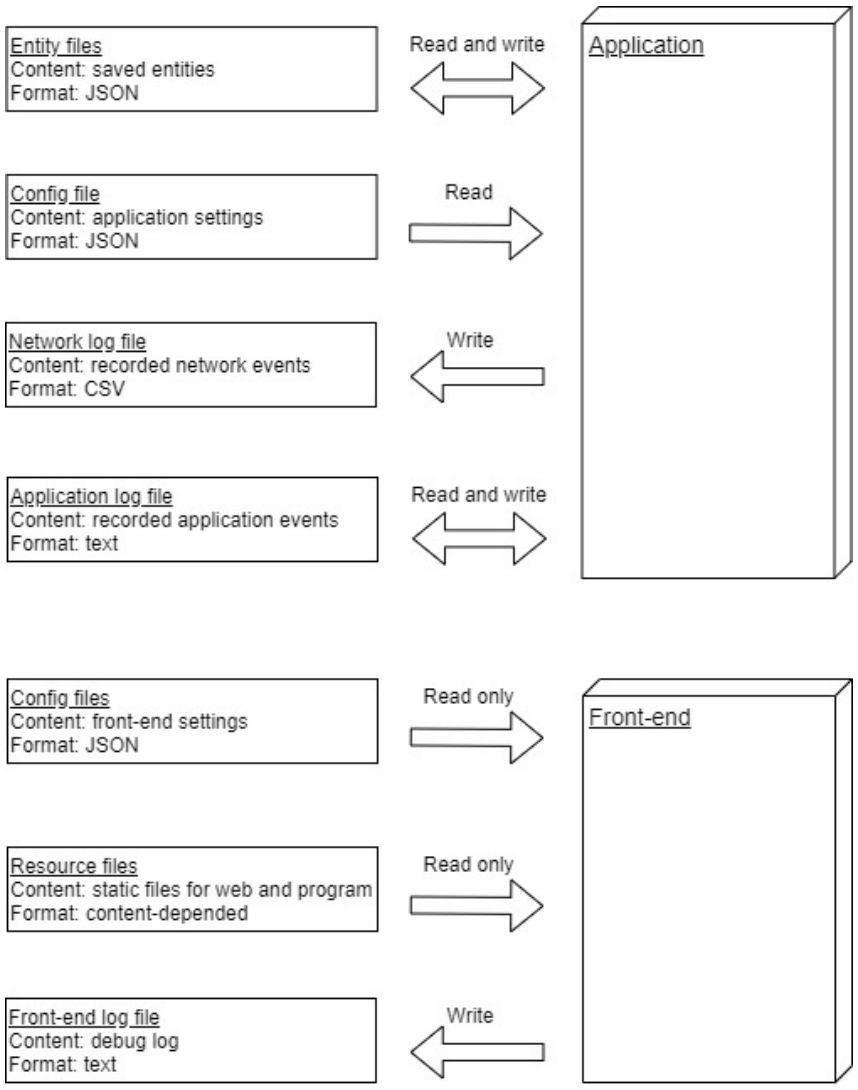

Figure 4.5: The solution data store

- DnsClient, Authors: MichaCo, License: Apache-2.0. It is smarter than standard DNS client and it have more important settings for performance optimization.
- Telegram.Bot, Authors: RoundRobin,Poulad, License: MIT. Easy-to-use Telegram bot API implementation for .NET.
- HttpToSocks5Proxy, Authors: MihaZupan, License: MIT. It is used to bypass Telegram prohibition in Russia.
- UnifiApi, Authors: Brent Howard, License: MIT. Provides API for interaction with Ubiquiti UniFi controller. It is not an obligatory part of the solution, it is added as en example of using user-added libraries for polling.

The solution also uses client's side libraries and other modules. They are listed below.

- Bootstrap-select v1.13.9, Copyright 2012-2019 SnapAppointments, LLC License: MIT. The jQuery plugin that brings select elements into the 21st century with intuitive multiselection, searching, and much more.
- Bootstrap Toggle, Copyright 2014 Min Hur, The New York Times Company, License: MIT. Switch button implementation for Bootstrap.
- dracula-theme, Copyright (c) 2016 Dracula Theme, License: MIT. Beautiful open-source dark color scheme. It is well suited for web pages that should not distract from the main content. Selected at the request of users.
- pako.js, Authors: Vitaly Puzrin and Andrei Tuputcyn, License: MIT. High speed "zlib" port to JavaScript. Used to decompress WebSocket server responses on client's side.

4.7 Summary

The developed solution was designed for a specific use environment based on a workable prototype. It does not aim to be a full replacement for the "heavy" corporate monitoring

systems that provide huge functionality. Instead, it offers an easy-to-use but fairly flexible approach, suitable for use where it is not possible to support more complex monitoring systems.

The main feature of the system is the ability to quickly collect information about VLAN using parameterized queries. This feature allows to automate monitoring in an area that relies heavily on vendor solutions, comprehensive network documentation, or resource-intensive device polling about all VLAN in the network.

In addition, the developed solution can be a transitional stage to more powerful systems, demonstrating the business effect of automatic network monitoring. The use of open storage formats for all data allows migration to another system, if the need arises.

In the meantime, users have a convenient, intuitive and flexible tool for observing network health, warning of failures and monitoring the structure of VLAN, without spending a lot of effort on updating data in the system.

Chapter 5

Tests and Discussion

This chapter presents and describes the tests that were developed to check if the project fulfills the objectives and solves the problem described in Chapter 3.

5.1 Application insight

The following are some screenshots of the pages of the finished web application.

Figure 5.1 is a network scheme page — the main user interface element. It consists of an application navigation element, a top toolbar, a selector for the diagram and view (a set of displayed parameters), a log of recent events, and, of course, a network diagram. The page displays a simple network of connected L2 and L3 switches, a router and a provider network (displayed as a cloud). Device "SW_2" has stopped responding, so it appears in "Error" state with red color. It is connected to "SWR_1" and "SWR_2", on their side the corresponding interfaces are down, therefore the interfaces' state is also "Error". Information about the transition to these states is displayed in the log. The connection interface of "SW_2" with the router is in "Warning" state.

Figure 5.2 is a web page for "MonitoringFunction" application entity. This page provides user access to entity properties, including the protocol used and the function code. It also selects the type of action that this function implements, such as checking device availability, obtaining a list of interfaces, etc.

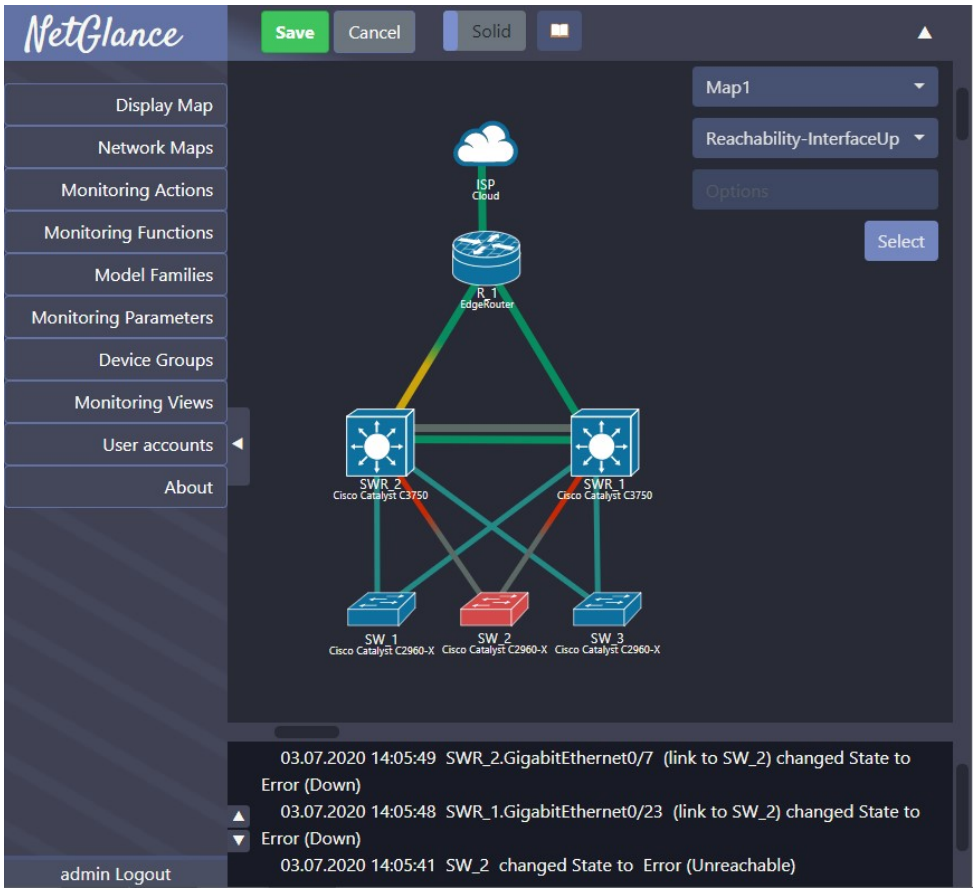

Figure 5.1: Home page with an example network scheme

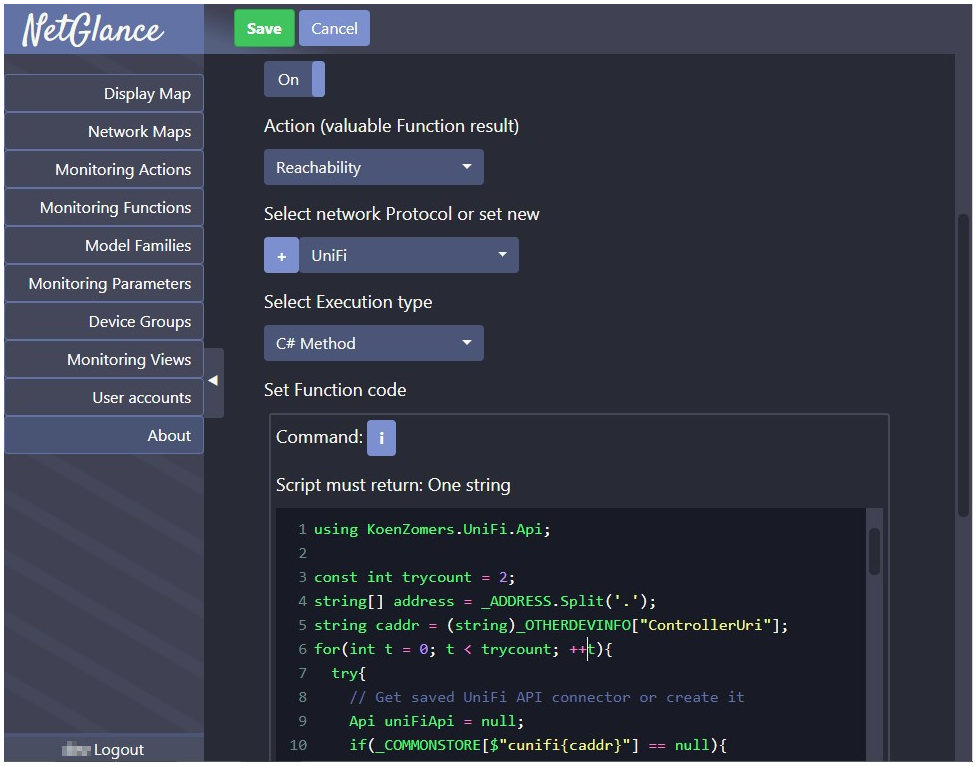

Figure 5.2: "MonitoringFunction" entity page

5.2 Tests

Various testing methods were used during the development, deployment and pilot implementation of the monitoring system. They allowed to create a solution that meets all the stated requirements, as well as clarify the requirements themselves in accordance with the future operating conditions of the program. Used unit, functional and user testing, as well as testing of performance, fault tolerance and security. Testing was done in a development environment and in the deployment environment, providing the necessary isolation from the production. The tests are described in more detail below.

During development, common testing methods were used to check the program code correctness. Application code was covered with unit tests approximately at 70%. Tests were created in parallel with the writing of program code. In total, there were 256 test scenarios (Figure 5.3). The high degree of test coverage provided a small number of problems at the integration stage. The tests are based on MSTest framework.

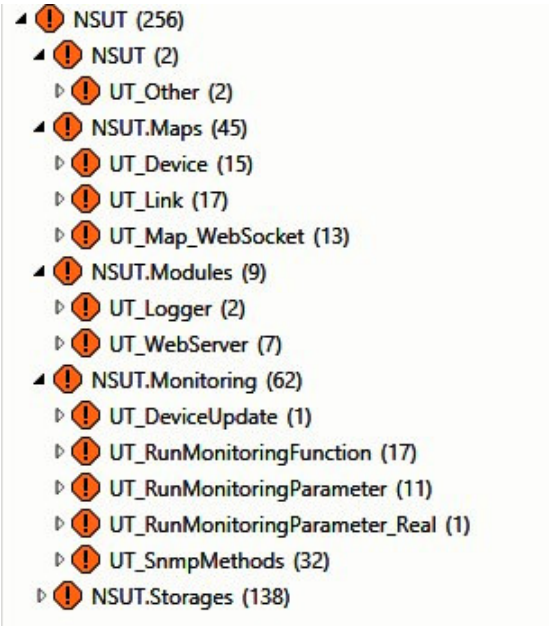

Figure 5.3: Application unit tests

After assembling the modules into a single solution, operational testing was carried out by performing all the actions available to the user and checking their results. Operations were performed manually using the user interface in a test environment. Using these tests, user interface errors and some errors in the interaction of solution components were detected and resolved.

In addition, a test network scheme and monitoring functions were made. It was used for modeling various monitoring situations in a test environment and checking monitoring correctness, sending messages mechanism. This test environment was implemented using fake network model with device and link states written in a file, and test monitoring functions that read variables in this file. To conduct this testing, the standard capabilities of the program were used to request information about network elements from a third-party source (Figure 4.4). Using this approach, it was possible to simulate a large number of monitoring cases to verify the correct response of the program to atypical cases.

Using network schemes containing a large number of elements, the performance of the solution was checked (load testing). The test schemes contain up to 500 devices and up to 1500 links. The test results show that recommended amount of devices and links is less

than 1500 elements summary due to user interface performance. But the program works correctly with a large number of elements, although the user interface response delays are getting longer.

The correct operation of processing possible internal errors was verified. Tests revealed that minor errors are logged and do not affect the overall system operation. Critical errors are also recorded and cause the system to stop working. They do not affect the safety of user data. In addition, the user input validation process has informative feedback using brief but meaningful notifications in the program. All debug modes record the corresponding information in the log, which helps in the development and refinement of the solution.

To determine the server resource consumption of the application and front-end parts, Visual Studio profiling tools were used. The results showed that approximate 120 MB memory is required to run both solution parts at the server. The profiling also confirmed that resource consumption increases linearly according to the number of network elements (see Figure 5.4).

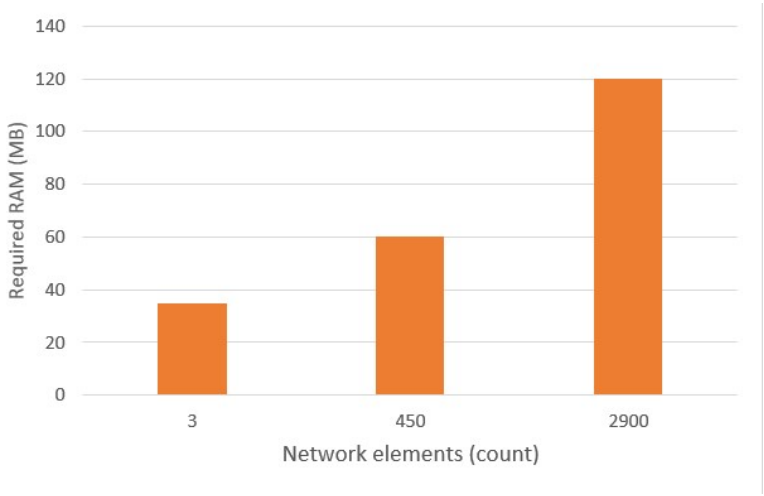

Figure 5.4: Application: RAM consumption depending summary amount of network elements

Some tests were performed with conditions that close to actual operating conditions (200 devices and 250 links). In particular, the network bandwidth requirements were tested this way. They are low: for checking network reachability and interface up-down

status with 20 sec cycle, less than 200 Kbit/s bandwidth has been used.

Also, load tests were done for multi-thread network polling system. The result is that the application successfully follows the defined maximum number of simultaneous monitoring tasks, and long tasks do not lead to an increase in the duration of the entire monitoring cycle.

User interface tests have included special tests for multi-user interaction. The result is that changes from one user are successfully distributed to other clients in real time, so the scheme keeps actual after every save operation. Also, for user interface tests, special network circuits were used, which included all the states of all available types of network elements to verify their correct displaying.

There were special tests for slow client connections. A poor connection was done using emulation in Google Chrome Developer tools, as well as a real slow VPN connection. These restrictions have not interrupted user experience significantly. This confirmed the justification for the widespread use of caching and compression of data transmitted over the network. Figure 5.5 is representing results of modeling network scheme page load duration for pure connection. It displays the duration from first browser query to full network scheme load. The conditions were: “Fast 3G” connection mode (90% of 1.6 Mbit/s for download, 90% of 750 Kbit/s for upload and 150 * 3.75 ms latency), CPU 2.20Ghz with 4 physical cores, Windows 8.1, Google Chrome 83, browser cache on HDD.

The program stability was tested using error prone user data for all input fields. All wrong data was detected and rejected during client side or server side validation. In addition, error prone polling scripts were added. They contain infinite loops and exception generators. They did not cause a negative effect, so that the program can be considered resistant to user input errors.

The network traffic between the client and server was analyzed using a traffic capture program “WireShark”. The goal was to make sure that the connection is encrypted, all the necessary data is compressed, and no extra data is transmitted. Traffic analysis has confirmed all these statements.

Security testing was done for authentication and authorization processes both for

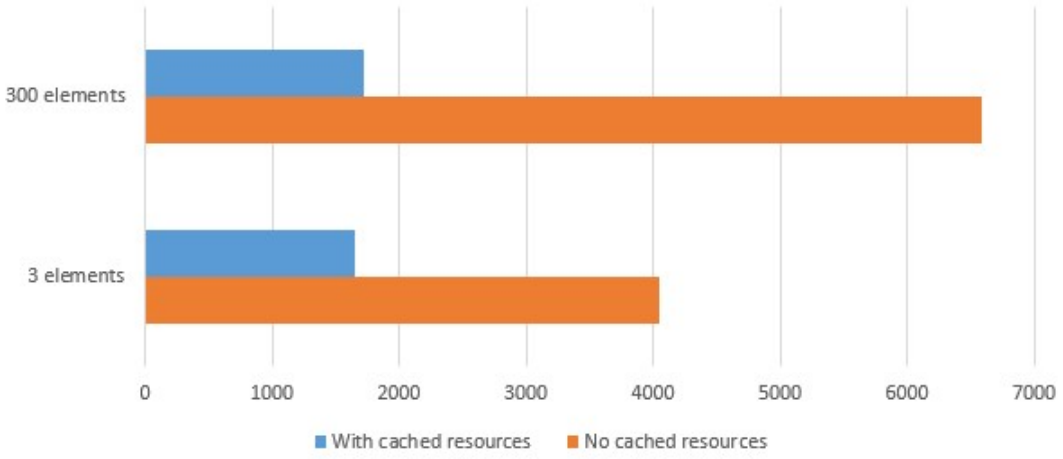

Figure 5.5: Network scheme page load duration through pure connection, ms

application part and front-end part. Standard tests showed that the solution checks an authority correctly. In addition, user-available requests were checked for Cross-Site Scripting (XSS) injection attempts. The result is that the program is XSS prone. Also, the inability to read information from the server outside the web root directory was verified.To summarize, the system can be considered resistant to the main attack methods.

To create a code for polling scripts (monitoring functions), a self-testing functionality was used that returns raw device response to user interface. It was implemented as an extra section in the entity page that contains fields to set destination properties (address, password, etc) like Device page, and output area that displays raw "MonitoringResult" content. This tool was so useful that it was added to the production version.

One of the requirements for the solution is to support various browsers and a mobile platform. To check the fulfillment of this requirement, the Firefox and Opera browsers of the current versions, the Google Chrome mobile emulation functional, a smartphone were used. Testing these environments showed that the necessary level of support is implemented in each case.

A workable environment from previous tests was provided for evaluation to end users. They tried the functionality of the program, made some minor clarifications of the requirements, made sure that this implementation met their expectations. The color scheme and

visual details of the user interface were clarified. The changes were applied and tested again. It took place along with parallel deployment to compare various implementations of the required UI elements under different URLs (as A-B testing).

5.3 Work results

"NetGlance NMS" was successfully deployed in KubSAU network. The deployed solution has covered all functional and non-functional requirements that were defined in Chapter 4.

5.3.1 Work results overview

The developed solution displays an informative, intuitive and easily editable network scheme. It shows the current network status of the selected parameters, which are monitored permanently or on demand. In addition, it represents a brief summary of the latest network events in text form.

The solution supports an easily expandable list of device models, including accessible through ICMP and SNMP, and even unmanaged devices as well as other data sources.

The program supports monitoring groups with various processing of network events. It can send alerts and can be controlled using a Telegram bot. It contains functionality to reduce the number of uninformative messages.

The solution has a log for network events and internal events. All detected network events, as well as important in-program events, are logged and can be subsequently analyzed for various purposes.

The system is web-based and designed for many simultaneous users. It does not require special browser settings or installing plug-ins. It has basic support for mobile devices. The access is secure and client web connections are encrypted. It is resistant to network connectivity failures and slow connections. It also checks the user input for validity, including the user code for polling the network.

Information about the device name and ports is collected automatically. This makes it easy to add devices and helps keep the network information up to date.

The program is highly resistant to monitoring configuration errors, including preventing too frequent requests, long requests, and the occurrence of uncaught exceptions in a script code.

Program entities, such as a network diagram, are backed up simply by storing data in files using the JSON format. Using this also facilitates the automation of data export and import if necessary.

The solution is deployed with minimal system privileges and does not use write-access accounts to poll the network. It does not require a web server or a database server for installation.

The system is suitable for use on user devices in all necessary scenarios, including mobile devices.

The solution is secure in terms of access control and preventing common attacks.

5.3.2 Goal achievement analysis

The main initial objective was to improve the quality of the network and reduce the load on staff by automating the collection of information about the network. This objective was planned to achieve using early problem detection and making work easier through VLAN troubleshooting automation.

The developed system was introduced and accepted for use. System users noted that all the necessary functionality has been implemented, and the interface is very convenient for work. User satisfaction level is high. Therefore, we can definitely assume that the goals are achieved.

5.3.3 Remarks and recommendations

However, some aspects of the program can be finalized and expanded. It is not part of the development requirements, but is required for enterprise network monitoring systems.

Logging should provide a wide functionality for obtaining statistics. In the current version, network and system logs are provided as files in CSV format and text format

without any tools to analyze it within the system.

In the further development of the product, it is recommended to use a database for storing program entities and logs. It is possible to keep the usability of files using the implementation of the mechanism for importing and exporting data to files.

It is necessary to view application logs through the interface. At the moment, receiving the tail of the application log is possible through the bot chat.

The solution lacks the functionality of passive data collection (SNMP traps). Although the program can be configured to use an external syslog server, this functionality is standard for network monitoring systems.

It is recommended that you implement full support for mobile platforms for the user interface, including moving elements on the network scheme.

In addition, unobtrusive validation for entity web forms should be used.

An important improvement would be the use of different polling frequencies for different device groups, as well as performing user queries outside the main polling cycle. This would allow users to receive relevant information faster, without overloading the network with too frequent requests.

Also, it is necessary to improve wireless links support to allow connect multiple devices to a wireless interface.

All these remarks and recommendations can be implemented in the future versions of the product.

Chapter 6

Conclusions

An investigation of KubSAU and IPB networks was carried out, a comparative analysis of the solutions used in them was made, and best practices were determined. The conclusions drawn from the study will be useful for the modernization of KubSAU network.

In the course of the work, a network monitoring system was created. It is functional, flexible and user friendly. It provides a fully editable visual displaying of a network topology, allows to monitor the status of network devices by the necessary parameters, and notify about important events using the messenger. It also makes it possible to fulfill requests with a parameter, such as VLAN number.

KubSAU has got a network monitoring system that successfully deployed and is used in IT department's everyday work to maintain and improve the network.

After the deployment of this system, the response to failures has accelerated.

The use of an easily editable network diagram facilitates its updating, including for temporary network segments (deployed at the time of events, the work of the selection committee, etc.).

Simplified work on changing the VLAN topology. Routine operations, prone to human error, were automated.

In the process of loading the network diagram into the program, information about the network was updated. The access level was initially not documented enough due to the large number of operational changes. Defects in the documentation were identified

and eliminated, such as invalid addresses and device models, port numbers.

Creating an up-to-date and easily updated scheme allowed us to look at the access level in a different way — keeping reliable information about it in the documentation, and not conducting problem-oriented investigations every time when necessary.

To create monitoring functions, a large amount of information on data collection from devices of various models was analyzed. Their features and shortcomings in the implementation of management interfaces were taken into account. In the work, a simple and flexible way of grouping devices was used — families of models identical in interaction from the point of view of the program.

When developing and deploying the program, special attention was paid to the security of the system, since it provides remote code execution and has access to network devices. The solution is quite safe due to the use of the concept of minimum privileges, the resistance of the program to data leakage and the use of reliable access control technologies provided by the development platform.

The created monitoring system is aimed at use in large networks with a high variety of network equipment used. It is well suited for day-to-day monitoring, as well as a place to consolidate network information.

Since Higher Education Institutions often have a large local network that has developed over time using different generations of network equipment, the solution should be considered for use in such organizations, not only in KubSAU.

6.1 Scientific features, innovation

Parametrized queries allow retrieve information on demand (e.g. about VLAN area). Similar features often are implemented in vendor's NMS, but we think it's a first system that combines permanent monitoring and on-demand queries with user-friendly network scheme viewer and editor.

6.2 Future work

Currently, two ways of developing a network monitoring system in KubSAU are being considered. The first option: continue to use NetGlance as primary NMS, but expand the logging functionality, add log analysis and network statistics tools.

Another way is to integrate the solution with the Zabbix monitoring system. Thus, it is possible to take advantage of the powerful functionality of the famous platform but keep performing parameterized queries using NetGlance. It is not necessary to support two independent network schemes — NetGlance will request it from Zabbix.

Bibliography

[1] J. McKendrick, "Enterprise data and the cost of downtime", *Information Today report*, Jul. 2012.

[2] D. Mauro and K. Schmidt, *Essential SNMP: Help for System and Network Administrators*. O'Reilly Media, 2005, ISBN: 9780596552770. [Online]. Available: https://books.google.pt/books?id=65%5C_0d25EpB4C.

[3] N. pl, *SNMP protocol: Network Basic. AL0-037*, ser. Network Basic. NOITE S.C. [Online]. Available: https://books.google.pt/books?id=z%5C_1zCwAAQBAJ.

[4] S. Karris, *Networks: Design and Management*. Orchard Publications, 2004, ISBN: 9780974423920. [Online]. Available: https://books.google.pt/books?id=7EQopK0Jh%5C_gC.

[5] DMTF, *Distributed management task force, inc*, https://www.dmtf.org/about, 2020.

[6] J. Russell and R. Cohn, *Web-Based Enterprise Management*. Book on Demand, 2012, ISBN: 9785512205655. [Online]. Available: https://books.google.pt/books?id=Pyf0MgEACAAJ.

[7] R. Siddaway, *PowerShell and WMI: Covers 150 Practical Techniques*. Manning Publications, 2012, ISBN: 9781617290114. [Online]. Available: https://books.google.pt/books?id=KC0ZygAACAAJ.

[8] R. Osso, *Handbook of Emerging Communications Technologies: The Next Decade*, ser. Advanced & Emerging Communications Technologies. CRC Press, 2018, ISBN: 9781420049626. [Online]. Available: https://books.google.pt/books?id=5fms2DW7mMUC.

[9] G. A. Donahue, *Network Warrior, 2nd Edition*. O'Reilly Media, Inc., 2011, ISBN: 9781449387860.

[10] S. D. Krothapalli, X. Sun, Y.-W. E. Sung, S. A. Yeo, and S. G. Rao, "A toolkit for automating and visualizing vlan configuration", in *Proceedings of the 2nd ACM Workshop on Assurable and Usable Security Configuration*, ser. SafeConfig '09, Chicago, Illinois, USA: Association for Computing Machinery, 2009, pp. 63–70, ISBN: 9781605587783. DOI: 10.1145/1655062.1655075. [Online]. Available: https://doi.org/10.1145/1655062.1655075.

[11] V. Gucer, A. Godoy, I. B. M. C. I. T. S. Organization, F. Salustri, G. Shah, J. Willis, and a. O. M. C. Safari, *Deployment Guide Series: IBM Tivoli Monitoring V6.2*, ser. IBM redbooks vol. 6. IBM, International Technical Support Organization, 2008. [Online]. Available: https://books.google.pt/books?id=KLZLzQEACAAJ.

[12] OpenNMS, *Opennms official website*, https://www.opennms.com/opennms-platform/.

[13] R. Olups, *Zabbix 1.8 Network Monitoring*, ser. From technologies to solutions. Packt Pub., 2010, ISBN: 9781847197696. [Online]. Available: https://books.google.pt/books?id=fsjNquHrQfYC.

[14] KubSAU, *Kubsau history*, https://kubsau.ru/university/history.

[15] ——, *Kubsau overview*, https://kubsau.ru/university.