TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

THE ANALYSIS AND IMPROVEMENT OF THE COMPANY'S MONITORING SYSTEM BUSINESS PROCESSES ON THE TELSET LTD. EXAMPLE

Bakalaureusetöö

Üliõpilane: Vladimir Gaidushin
Üliõpilaskood: 143011IABB
Juhendaja: Inna Švartsman

Tallinn 2018

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Vladimir Gaidushin

21.05.2018

Abstract

The thesis examines the problem of delivery of notification of incidents by the example of Telset Ltd. company. Long unavailability of the service cause the customer will cancel the subscription and the company will incur losses. The goal of the work is to correct the current situation in the company by developing a new notification system which will reduce alert time and reduce customer downtime. Reducing downtime helps to save the customer and avoid losses for the company

The work explores the current shortcomings and problems that need to be corrected in the new architecture of the system.

As a result of the work done, it will be seen that the system has made adjustments and corrected the company's existing problems.

This thesis is written in English and is 25 pages long, including 5 chapters, 14 figures and 2 tables.

Annotatsioon

Ettevõtte häire süsteemi äriprotsesside analüüs ja nende parendamine ettevõtte Telset AS näitel.

Bakalaureuse töö uurib juhtumite teate edastamise probleemi Telset AS'i näitel. Teenuse pikaajaline kättesaamine põhjustab kliendi tellimuse tühistamise ja ettevõte kannab kahjumit. Töö eesmärgiks on parandada praegust olukorda ettevõttes, töötades välja uue teavitussüsteemi, mis vähendab märguandeaega ja vähendab klientide seisakuid. Puudujäägi vähendamine aitab klienti päästa ja ettevõtte kahjumit vältida.

Töös uuritakse praeguseid puudusi ja probleeme, mida tuleb süsteemi uue arhitektuuri puhul korrigeerida.

Töö tulemusena näeme, et süsteem on parandanud ja parandanud ettevõtte olemasolevaid probleeme.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 25 leheküljel, 5 peatükki, 14 joonist, 2 tabelit.

List of abbreviations and terms

HTML	**HyperText Markup Language** The standard markup language for creating web pages and web applications.
REST	**REpresentational State Transfer** An architectural style that defines a set of constraints and properties based on HTTP.
HTTP	An application protocol for distributed, collaborative, and hypermedia information systems.
PHP	A server-side scripting language designed for web development.

Table of contents

List of drawings

List of tables

1. Introduction

Currently many companies are using different monitoring systems to monitor servers, devices or system events. All this monitoring systems send notifications when an event occurs or system state is near a trouble. Typically these notifications are sent as an email and this method is not so fast nowadays.

Companies can have multiple monitoring systems, because they are powerful in own way and each department of the company can have own monitoring system to cover all systems as much as possible.

By default, monitoring system sends notifications by email, but this causes a big problem with delivery time and it can take up to 15 minutes depends on client's email specification. This reaction time is totally unacceptable.

1.1. Background and problem

Telset AS is a telecommunication company that provides Internet, Television and home telephone services. Currently the company uses many monitoring systems to cover all available devices. Because this is a telecommunication company, the reaction time for events must be minimal and also access to these events/notifications must be from any device.

By default, systems provide ability to send notifications via email and some of these give opportunities to extend notification delivery methods. As email delivery time can be up to 15 minutes, it is unacceptable, because in critical situations every minute can cost money for the company. Broken infrastructure means that clients could not receive own services and the client's attitude towards the company deteriorates. In situation when service is not available for client, the company must receive a notice about incident as soon as possible. Long unavailability of the service leads to customer's cancellation of the subscription and as the result company incurs losses. Also a lot of emails in mail box from monitoring systems makes inbox like a "spam" box which cause search problem with searching among received notifications and important emails.

1.2. Setting up a task

The primary goal of the work is to improve today's company business logic of delivery notifications about incidents. Which in turn includes the analysis of today's system and its correction.

In addition to the delivery problems, it is also necessary to solve the problem with convenient access to these notifications through the mobile phone of the company's employees.

1.3. Methodology

The author of this work first of all needs to analyze current business logic of monitoring department to know what systems the company uses and also to know a little bit about the system from a specialist.

Second and third steps are the "Architecture and Building" steps. In this part of the work author will develop a new notification system logic. After the architecture of the system is thought out, the stage of its construction, in particular, Backend and API, will begin, after which a client application for iOS and Android devices will be developed.

The final step of the work is the "Integration and Validation" step. In this step the new system will be deployed to Live environment and the results will be evaluated. The results of evaluation will show how the whole process was improved.

2. Company background

Telset AS was founded in 1991 year and provides telecommunication services to clients. The clients of the company are both private clients and business clients. Most of the clients are private customers. [1]

The company provides widespread types of telecommunication services, such as: Digital television, Analog television, telephony and the Internet connection. [1] As it is a telecommunication company, it means that every of the service requires a lot of devices. As a result, the company has a large infrastructure consisting of many devices that need to be monitored and controlled. In case of failure, device is needed to be repaired or replaced for service resumption.

2.1. Internet connection

The Internet connection technically can be provided in many methods. The most popular ones are: Fiber, DOCSIS and ADSL/VDSL. Each option has its advantages and disadvantages. The popularity of some variants has developed historically, as updating to new technologies requires large financial investments.

2.1.1. ADSL and VDSL

ADSL or Asymmetric Digital Subscriber Line is a type of DSL connection that utilizes regular copper phone lines that aren't used by voice calls. This kind of connection don't require any special lines to be installed, so it's less expensive and more available than other forms of broadband. In poorly developed cities, this is the only way to give the Internet connection. [2]

Technical details [3]:

- Down-/Upstream Rate[1]: 24/1 Mbps
- Efficiency range[1]: 5 km

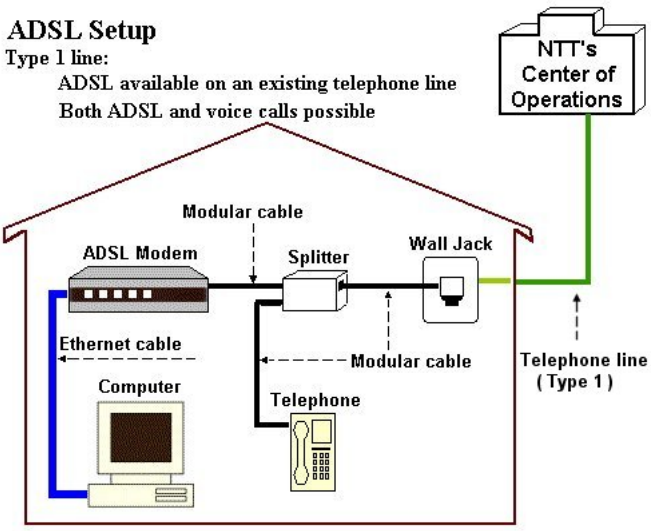
[1] Technical standard max.

Very-high-bitrate Digital Subscriber Line or VDSL service is closer to cable Internet in speed and behavior than ADSL. VDSL accomplishes this with more efficient use of phone lines achieved through a configuration that effectively shortens the distance that the signal has to travel. Short distances have less loss, which gives the most reliable connection. A higher amount of available bandwidth delivers better overall performance than ADSL can offer. Regardless of the improved technology, the distance and condition of the cable affects the quality of the connection. [2]

Technical details [3]:

- Down-/Upstream Rate[1]: 100/40 Mbps
- Efficiency range[1]: 1 km

In operator view it means that they have a main device which can accept 24-48 clients per device and each client has own ADSL/VDSL modem to accept the Internet connection. In this technology, the most important is the monitoring of the main device.



Picture 1. ADSL connection scheme [4]

2.1.2. DOCSIS 3.0

DOCSIS or Data Over Cable Service Interface Specification – an international telecommunications standard that permits the addition of high-bandwidth data transfer to an existing cable TV (CATV) system. [5]

Technical details [3]:

- Down-/Upstream Rate[1]: 200/100 Mbps[2]
- Efficiency range[1]: 1-100 km[3]

In operator view this technology is technically similar to ADSL in terms of the chain "Main Device - Client Device", but ADSL technology requires a separate line (cable) for each client, in case of DOCSIS one main cable can be used and split with other clients.

This technology requires to monitor main device and also client devices. Monitoring the client devices helps to see overall CATV network status.

2.1.3. UFB/Fiber connection

Fibre or UFB is the fastest growing Internet connection technology that gives speed 1 Gbps or more for both uploads and downloads. Speeds around 100 - 300 Mbps are more common for UFB connection and give most reliable Internet connection for customers. UFB bypasses phone lines and uses smaller, lighter fiber optic cables with glass conductors. These conductors transmit light signals rather than electricity, so they aren't subject to interference from electrical wires or damage from lightning strikes. Because of the high cost of cable laying, customers from less developed regions can't get this type of Internet connection. [6]

Technical details [3]:

- Down-/Upstream Rate[4]: 1/1 Gbps (and more)
- Efficiency range[1]: 10-60 km

[2] EuroDOCSIS
[3] Depends on amplification
[4] Technical standard max.

This technology is used in the backbone networks of the operator, since it has the least losses and allows the networks to grow as a spiderweb. At each intersection is placed a device called "Switch". Each “Switch” must be monitored by the provider. In general, switches occupy the largest number of devices that need to be monitored.

2.2. Television

The company provides television service upon coaxial (Digital television) and the Internet network (IPTV). Television network works over the Internet network.

2.2.1. Digital television

Operator monitors many aspects of the service including: sources availability, quality of the sources, transmission devices, etc.

A large amount of the devices works over the UFB network and mostly required to monitor the quality of the devices and input/output noises.

2.2.2. IPTV

Instead of receiving TV programs as broadcast signals that enter client’s home from a coaxial cable, client gets programs streamed (downloaded and played almost simultaneously) through the Internet connection. [7]

IPTV gives a big advantage over digital television because it provides two-way response. This gives ability to monitor popularity of the programs, provides cinema feature, show ads, buy content from the TV.

Technically system consist of many servers (Middleware, Databases, Transcoders) which are monitored by the company.

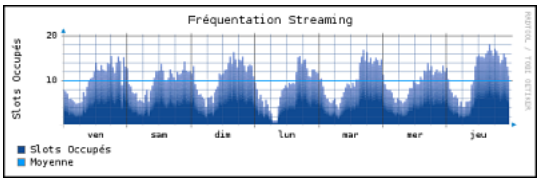
2.3. Monitoring systems

Monitoring software observes and tracks the operations and activities of applications and network services on a computer or enterprise systems. This type of software provides a way to supervise the overall processes that are performed on a computing system and provides reporting services to the system or network administrator. [8]

2.3.1. Cacti

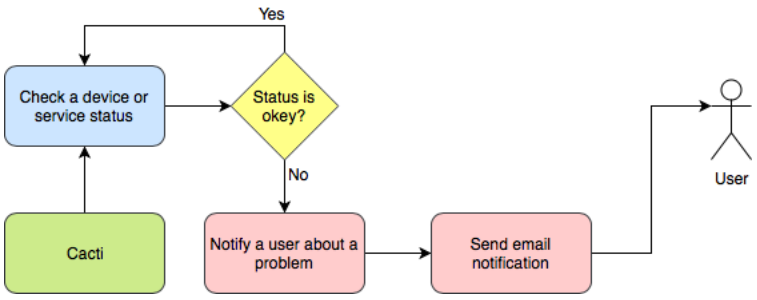
Cacti is a complete network graphing solution designed to harness the power of RRDTool's data storage and graphing functionality. Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box. All of this is wrapped in an intuitive, easy to use interface that makes sense for complex networks with thousands of devices. [9]

RRDtool is the Open Source industry standard, high performance data logging and graphing system for time series data. [10]



Picture 2. The graph generated by the RRDtool [10]

Cacti is primarily used by the company for logging and plotting traffic between devices and the state of client devices. The sub-routine graphs helps to find the cause of the problem, as well as the approximate time and behavior of the device in case of failure.



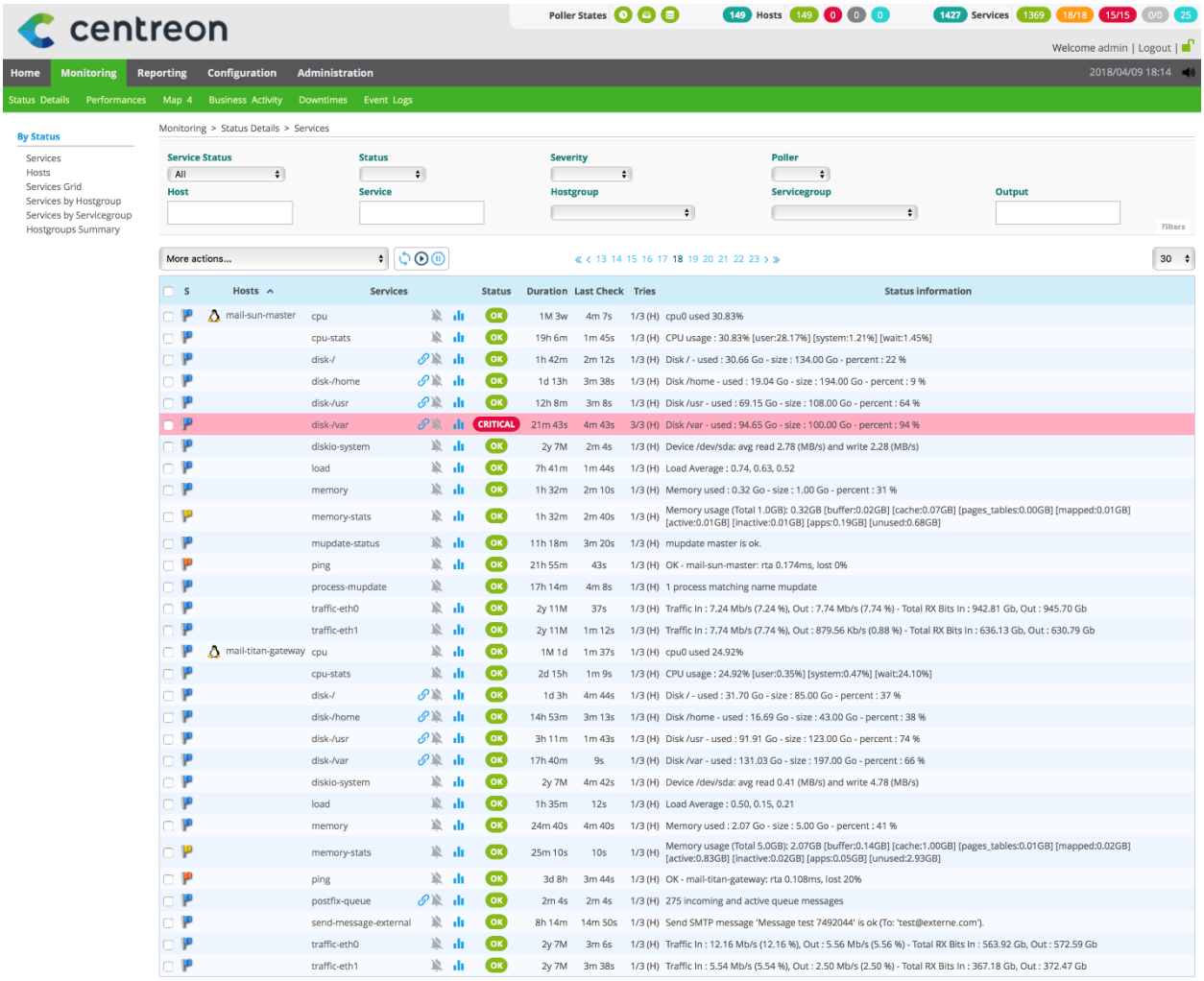
Picture 3. Current business logic of notifications delivery in "Cacti"

2.3.2. Centreon

Centreon is an Open Source software package that lets supervise all the infrastructure and applications comprising information system. [11]

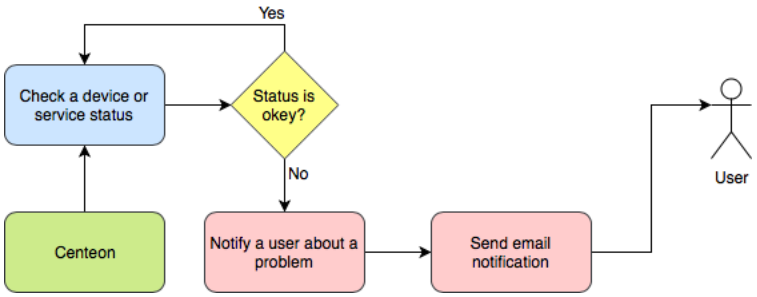
Centreon is the system and network monitoring software adapted to the needs of the company ISD (Information Services Department). It combines Open Source flexibility with the essential features of excellent supervision [12]:

- Monitoring servers, services and applications
- Real-time management console
- Expandable thanks to the open API and Open Source



Picture 4. Centreon management panel overview

In the company Centreon system is used as a main monitoring system. Centreon monitors all primary devices and services. The system allows to configure templates for checking servers, devices or applications by many criteria, for example: CPU load, RAM, disk usage, network traffic consumption, ping and much more. In the event that a deviation or error occurs in the service, Centreon sends an Email notification to the relevant addressees who are associated with the problem device.



Picture 5. Current business logic of notifications delivery in "Centreon"

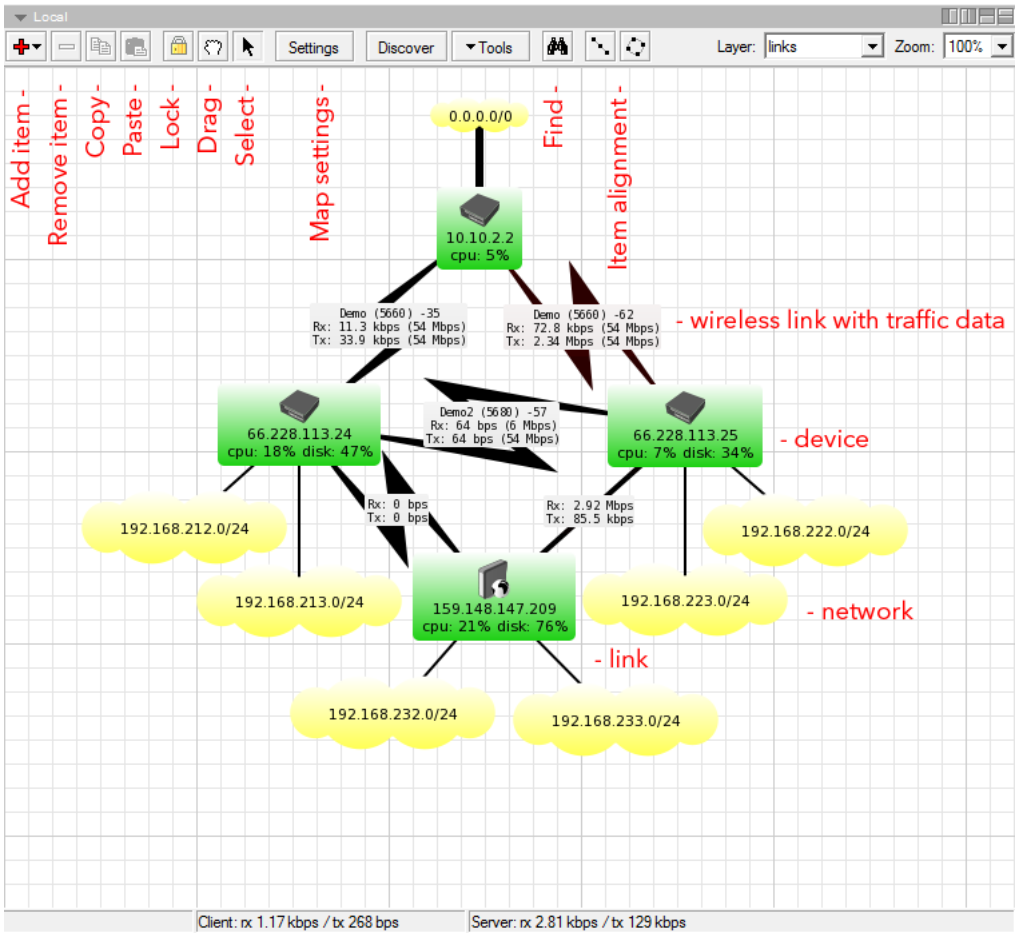
Business logic of notifications delivery in "Centreon" (*Picture 5*) is similar to Cacti. Centreon checks the status of a devices, service or application and if any problem occurs the system sends a notification via Email.

2.3.3. The Dude

The Dude is a network monitor which automatically scan all devices within specified subnets, draws and layouts a map of your networks, monitors services of your devices and alerts you in case some service has problems. [13]

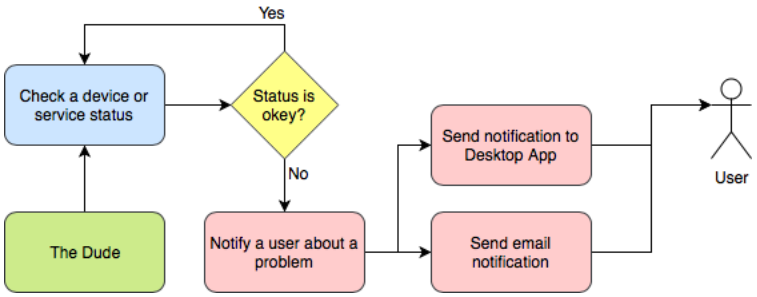
The program features [13]:

- Auto network discovery and layout
- Discovers any type of device
- Allows to draw custom maps
- Allows to add custom devices and mark them with icons
- Supports SNMP, ICMP, DNS and TCP monitoring
- Direct access to remote control tools



Picture 6. The Dude program screenshot. [8]

The Dude has a big advantage over the other monitoring systems, basically, because it is Desktop app which allows to draw a map and place devices on it. This system is very useful for system administrators as it shows a problem route and administrator can easily see what device was broken and which devices or part of the city was affected. However, the system is not so powerful to replace it with other services.



Picture 7. Current business logic of notifications delivery in "The Dude"

Business logic of notifications delivery in “The Dude” system (Illustrated on the *Picture 7*) is also similar to Cacti and Centreon, but as it is a desktop app it adds ability to show notifications in the app. It incredibly speeds reaction from system administrators, which is very important.

3. Creating a new business logic

In this chapter, the author creates a new business logic based on known data about the company.

3.1. Comparison of monitoring systems

The author compares the monitoring systems to identify the weaknesses of each. The main criteria for comparison are: the methods of delivery of notifications, as well as their speed.

According to GitHub pages of Cacti [14] and Centreon [11] as software is open source, the products are actively developed nowadays. “The Dude” also continues its development, which is confirmed in their changelog [15].

The author checks the capabilities of each of the systems based on their documentation or a page with the capabilities of the systems. The basis was used: “Cacti” - features page [16], Centreon – documentation [17] and The Dude - advantages page [13].

	Cacti	Centreon	The Dude
Product support	Yes.	Yes.	The product was frozen, but a year ago the development continued.
Drawing graphs	Yes.	Yes.	No.
Notification types	Email.	Email, Plugins.	Desktop app, Email.
Notification methods extendable	No.	Build own request or with plugin.	No.
Delivery time	Up to 15 minutes.	With Email up to 15 minutes.	Email – Up to 15 minutes, Desktop – app – Immediately.

Tabel 1. Comparison of monitoring systems

Based on the results of the comparison from *Tabel 1* the following conclusions are drawn:

- Primary included delivery method of notification is Email and only one system allows to extend delivery methods with custom ones.
- There are no delivery time problems when using "The Dude" with desktop app otherwise systems provides Email method which can take up to 15 minutes before user will receive an email message.
- Due to the fact, that not all systems allow to extend delivery methods, the new system must also include ability to receive notifications from monitoring system by email.
- Most systems draw graphics which can be attached to a notification.

3.2. System requirements

Based on problem described in chapter "1.1. Background and problem" and results of comparison in chapter "3.1. Comparison of monitoring systems" the author made a list of requirements for the new system:

1. Notifications must be accessed at any time – This means, that all notifications must be stored in a separate database.
2. Easy access for the company employees.
3. Employees must have ability to view notifications through mobile application.
4. Sending targeted notifications to users – Notification can be send to specific user, not all of company's employees are allowed get some kind of notifications.
5. Collecting notifications by API and Email – When it is possible would be used API method, otherwise Email method will be used. The API method will significantly speed up the delivery of notifications.
6. Notifications can include images.

3.3. System architecture

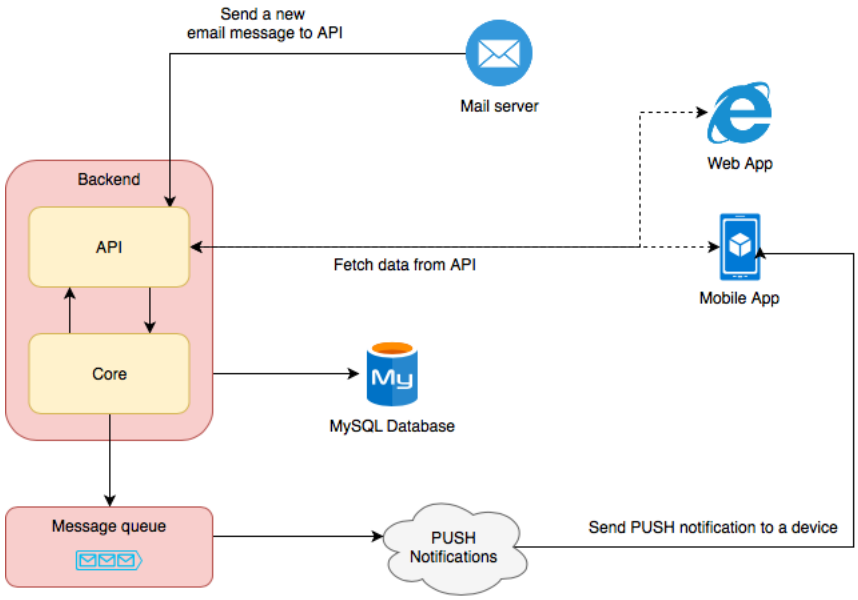
The requirements listed in chapter "3.2. System requirements" show that in order to implement this system, it is necessary to create a separate microservice.

System architecture is divided in 4 parts:

1. Backend – The core of the system.
2. API – The input methods for notifications, includes: REST API and Email inbound routing.
3. Database – For storing all data the MySQL database will be used.
4. Notifications delivery provider – The way between system and user device.

Frontend consists of two parts:

- Web version – Can be access from any device where browser is available.
- Mobile version – Version for iOS and Android devices. Also gives ability to receive PUSH notifications which can dramatically decrease delivery time to seconds.



Picture 8. System architecture

In the *Picture 8* is provided a diagram of the application flow.

A new notification can be created in two methods: Directly by the API route or by sending an email message to mail server which will redirect it to the API. In turn, the API stores the received notification in the MySQL database and sends a PUSH notification to the

phone via a special server. Delivering notifications to recipients can take seconds, but API must response as quick as possible. For this case is used "Message queue", system sends a new notification to queue and in a background processing sending notifications to recipients.

Frontend apps are also running over API. Authorization in the apps based on OAuth 2.0 technology. In mobile app a user can confirm receiving PUSH notifications what will give ability for backend to send PUSH notifications to the device.

3.4. Development of the system

The system consists of two parts: Backend and Frontend. The author started with developing backend part at the first step. Backend is better to develop as a first step, because frontend side work under API which is part of the Backend.

3.4.1. Backend

The core of the backend was developed on PHP language with using the Laravel framework. Laravel gives all required tools for modern web app developing. [18]

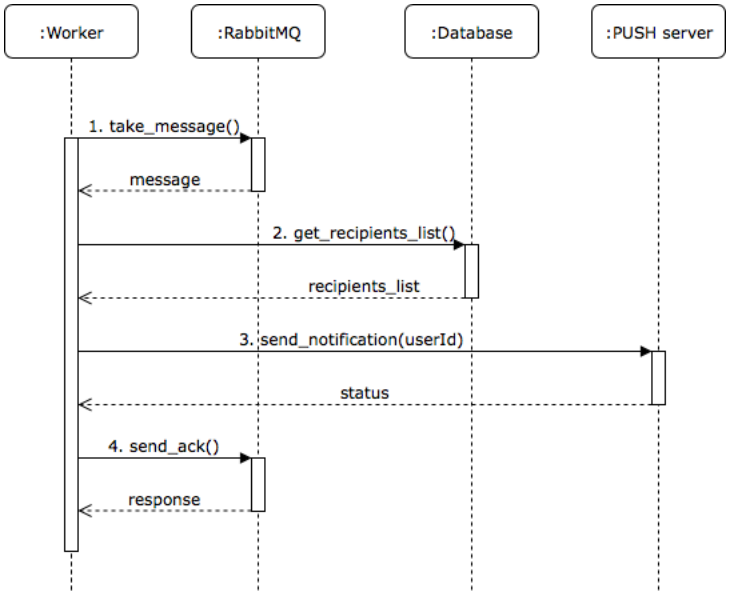
According to chapter “3.2. System requirements” and “3.3. System architecture” the backend must realize the following requirements:

- **Authorization** - Only approved employees should access the system. The system should have, as usual authorization through the browser, and implement OAuth 2.0 for authorization in applications.
- **Database connection** – All data must be stored in MySQL database.
- **API** – Is a single point to interacting with other services or apps.
- **Queue connection** – New notifications must be added to queue to improve speed of the system.

3.4.2. Message queue

Message queue is between backend and PUSH notifications. Message queue is running under RabbitMQ. Message queue work in pair of workers. Workers are small processes which tacking messages from the queue and make stuff required by the program.

The author created a queue worker, written on Node.js.



Picture 9. Sequence diagram of queue worker.

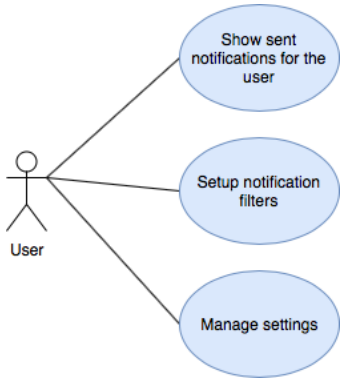
The queue worker works according to the scenario illustrated on the *Picture 9*:

1. Take a message from the queue.
2. Parse message and get a list of required recipients.
3. Send a PUSH notification to recipients.
4. Send *ACK* signal to the queue to release message from the queue.

The queue worker works on a server as a background task 24/7.

3.4.3. Applications

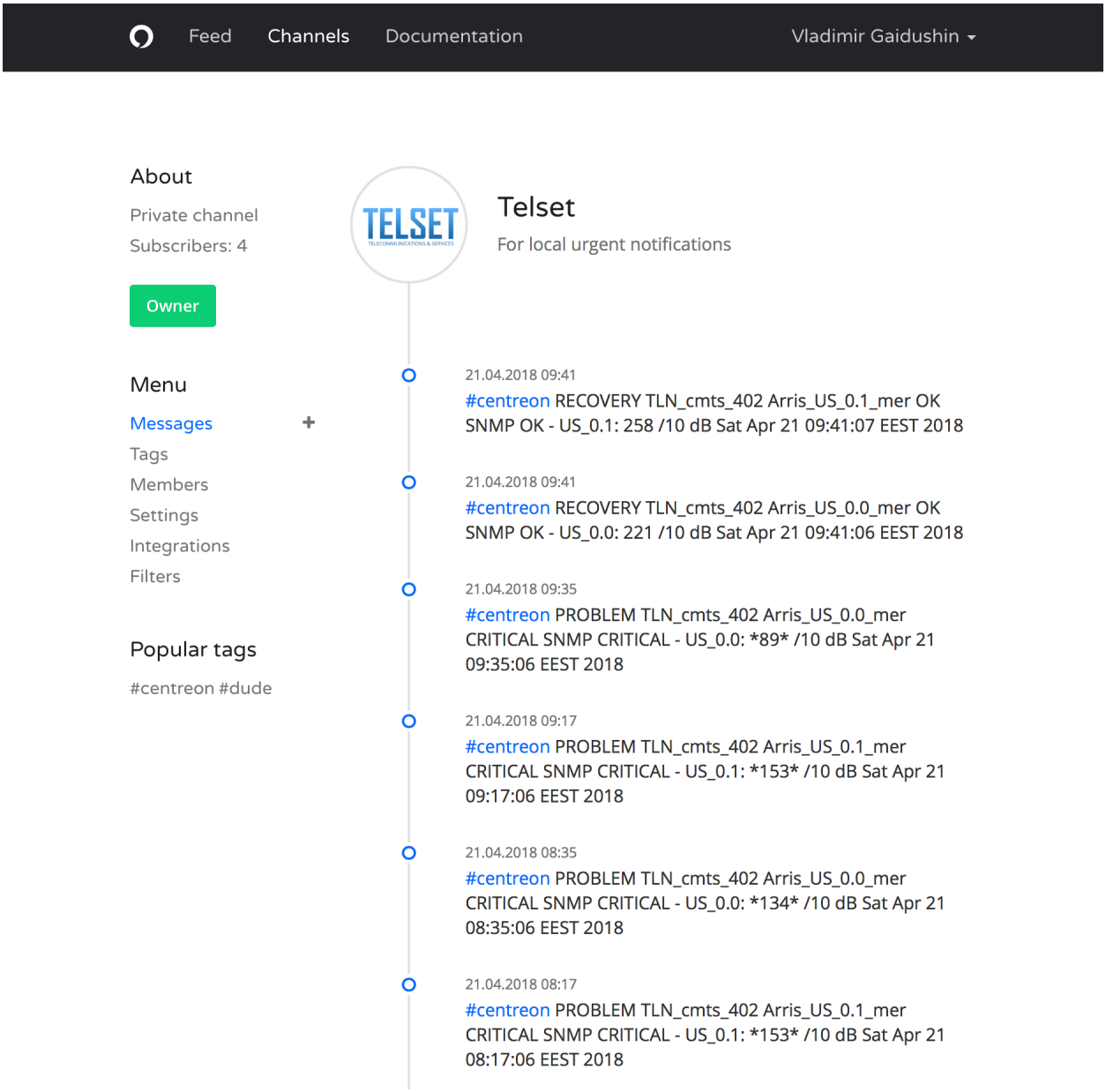
The system consists of two frontend apps – **web** and **mobile** app. The author decided to create a light version of mobile app with primary features and web version with all available settings.



Picture 10. Web application use case diagram.

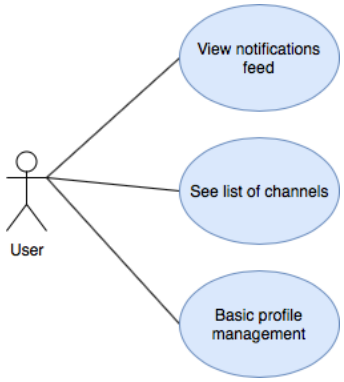
The primary functions of the **web application,** which are described on use case diagram in *Picture 10*:

- Show sent notifications for the user
- Setup notification filters
- Manage settings



Picture 11. Preview of the web application

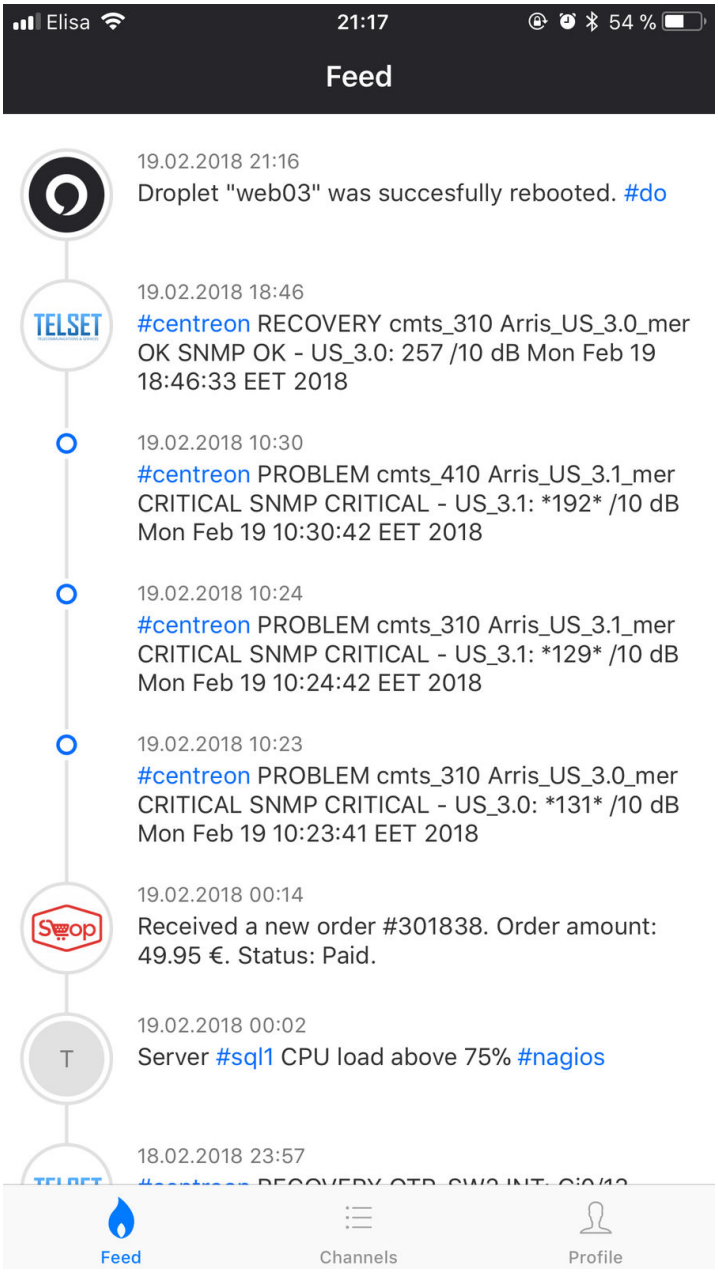
In the *Picture 11* is demonstrated view of the system. System allows to: see number of subscribers, notifications feed, list of employees, allow to change settings, setup inbound E-mail address and filter messages by tags.



Picture 12. Mobile application use case diagram.

The **mobile application** is lighter than the web application. To mobile application the author included the next functionality (As described on use case diagram in *Picture 12*):

- View notifications feed
- See list of channels
- Basic profile management



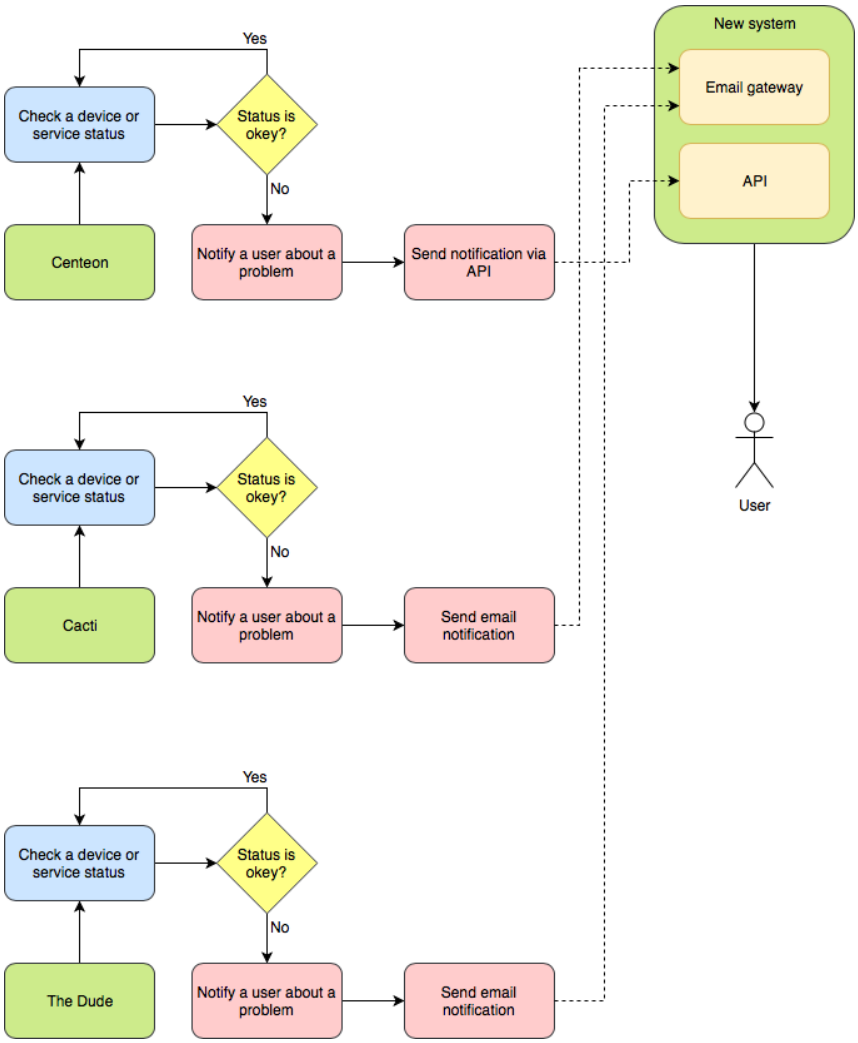
Picture 13. Preview of the mobile application

Picture 13 demonstrates mobile application. The concept of the app is a tab bar application with 3 tabs: Feed (A list of notifications), Channels (List of channels user subscribed) and Profile tab (User details).

4. Integration to the infrastructure

The developed system was integrated to the company infrastructure as a separate service. The system was configured to receive notifications from next systems:

- Cacti – Configured sending notification via E-mail to the system Email gateway.
- Centreon – Was created a new command in the monitoring system which sends direct API request to the system.
- The Dude - Configured sending notification via E-mail to the system Email gateway.



Picture 14. New business logic of delivering notifications.

One of the project goals was to decrease notifications delivery time. To verify the result, the author measured the average time of delivery of the notification to the employee comparing the new system with the old one. For the measurements was taken 25 last real notification from monitoring systems which. The results of measurements are presented in the *Tabel 2*. The average delivery time was **9,8 seconds** for new system and **590,6 seconds** for old system which is **60 times** faster. The maximum delivery time was **16,3 seconds** for new system and **1329,5 seconds** for old system which is **80 times** fewer. The results showed that the new system extremely improved the delivery speed.

Picture 14 shows that monitoring systems “Cacti” and “The Dude” are also continuing using email as a primary delivery method, but comparison results of monitoring systems in *Tabel 1* showed that this method takes up to 15 minutes, however, this method of delivery after integration of the system takes a few seconds. This is due to the fact that the system uses a special mail server that receives the message and instantly transfers it to the API. Unlike a regular email client on a mobile device, when the mailbox is checked every few minutes. This method contributes to such a large increase in the speed of delivery.

5. Conclusion

The purpose of this work was to optimize the method of delivery of notifications about incidents to employees. The optimization included: the increase in the speed of delivery, the convenience of their reading and accessibility.

In the course of the work, a new system architecture was developed and also implemented. As a result of the work done, the new system was integrated into the working environment. The results of the integration showed that the speed of delivery has increased up to several seconds. Employees got the opportunity to easily read their notifications with the help of their mobile phone.

The goals of the work were achieved on the basis of external sources of information, implementation of the application and the author's observations.

References

[1] Telset AS, "Firmast," 04 2018. [Online]. Available: http://telset.ee/rus/about/. [Accessed 4 April 2018].

[2] Craig, "WHAT'S THE DIFFERENCE BETWEEN ADSL, VDSL AND FIBRE INTERNET?," 09 01 2017. [Online]. Available: http://deucethemes.com/whats-the-difference-between-adsl-vdsl-and-fibre-internet/. [Accessed 13 April 2018].

[3] ateneKOM, "Comparison of different broadband technologies," 2014. [Online]. Available: http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?doc_id=4827. [Accessed 15 April 2018].

[4] "WD634GV ADSL modem configuration.," [Online]. Available: https://asahi-net.jp/en/support/guide/wd634gv.html. [Accessed 9 April 2018].

[5] "DOCSIS," 15 May 2018. [Online]. Available: https://en.wikipedia.org/wiki/DOCSIS. [Accessed 20 May 2018].

[6] haloweb, "WHAT'S THE DIFFERENCE BETWEEN ADSL, VDSL AND FIBRE INTERNET?," 13 March 2017. [Online]. Available: https://www.haloweb.co.za/whats-the-difference-between-adsl-vdsl-and-fibre-internet/. [Accessed 18 April 2018].

[7] "IPTV," 4 January 2018. [Online]. Available: http://www.explainthatstuff.com/how-iptv-works.html. [Accessed 2018 April 13].

[8] "Techopedia," [Online]. Available: https://www.techopedia.com/definition/4313/monitoring-software. [Accessed 8 April 2018].

[9] I. The Cacti Group, "What is Cacti?," [Online]. Available: https://www.cacti.net/what_is_cacti.php. [Accessed 10 April 2018].

[10] T. Oetiker, "About RRDtool," 2 February 2017. [Online]. Available: https://oss.oetiker.ch/rrdtool/. [Accessed 10 April 2018].

[11] Centreon, "centreon/centreon: Centreon is a network, system, applicative supervision and monitoring tool," September 2017. [Online]. Available: https://github.com/centreon/centreon. [Accessed 16 April 2018].

[12] anismaj, "Centreon: A Free Alternative To Nagios Xi Monitoring Platform," 27 January 2015. [Online]. Available: https://www.unixmen.com/centreon-free-alternative-nagios-xi-monitoring-platform/. [Accessed 2018 April 15].

[13] MikroTik. [Online]. Available: https://mikrotik.com/thedude. [Accessed 9 April 2018].

[14] "Cacti/cacti: Cacti ™," [Online]. Available: https://github.com/Cacti/cacti. [Accessed 14 April 2018].

[15] MikroTik, "All current and historical changelogs," 17 04 2018. [Online]. Available: https://mikrotik.com/download/changelogs. [Accessed 9 April 2018].

[16] I. The Cacti Group, "Features," [Online]. Available: https://www.cacti.net/features.php. [Accessed 17 April 2018].

[17] "Welcome on Centreon Documentation," [Online]. Available: https://documentation.centreon.com/?lang=gb. [Accessed 17 April 2018].

[18] "Laravel - The PHP Framework For Web Artisans," [Online]. Available: https://laravel.com. [Accessed 9 May 2018].

Appendix 1 - Measurements of notifications delivery times

Message	System	Delivery time with new system (sec.)	Delivery time with old system (sec.)
PROBLEM OTH_cmts_294 Arris_US_2.1_mer CRITICAL SNMP CRITICAL - US_2.1: *0* /10 dB	Centreon	7,5	499,1
PROBLEM OTH_cmts_294 Arris_US_2.1_mer CRITICAL SNMP CRITICAL - US_2.1: *0* /10 dB	Centreon	4,2	712,9
RECOVERY OTH_cmts_294 Arris_US_2.0_mer OK SNMP OK - US_2.0: 358 /10 dB	Centreon	6,4	884,0
PROBLEM OTH_cmts_302 Arris_US_0.0_mer CRITICAL SNMP CRITICAL - US_0.0: *77* /10 dB	Cacti	13,4	726,9
RECOVERY OTH_cmts_302 Arris_US_0.1_mer OK SNMP OK - US_0.1: 229 /10 dB	Centreon	15,6	818,1
Service ping on Joodre is now down (timeout)	The Dude	15,3	29,5
Service ping on Joodre is now up (ok)	The Dude	15,4	480,9
PROBLEM OTH_cmts_302 Arris_US_0.0_mer	Centreon	8,7	597,5

CRITICAL SNMP CRITICAL - US_0.0: *103* /10 dB			
Service ping on Roude is now down (timeout)	The Dude	16,3	44,8
Service ping on Roude is now up (ok)	The Dude	10,4	745,3
PROBLEM OTH_cmts_302 Arris_US_0.0_mer CRITICAL SNMP CRITICAL - US_0.0: *0* /10 dB	Cacti	6,6	749,1
PROBLEM OTH_cmts_302 Arris_US_0.0_mer CRITICAL SNMP CRITICAL - US_0.0: *0* /10 dB	Cacti	7,8	469,2
PROBLEM OTH_cmts_302 Arris_US_0.0_mer CRITICAL SNMP CRITICAL - US_0.0: *0* /10 dB	Centreon	14,3	868,0
RECOVERY OTH_cmts_302 Arris_US_0.1_mer OK SNMP OK - US_0.1: 229 /10 dB	Centreon	9,9	227,0
Service ping on Liikuri 155 is now down (timeout)	The Dude	6,8	609,7
Service ping on Liikuri 155 is now up (ok)	The Dude	10,2	984,3
Service ping on SCK_SWITCH-4 is now down (timeout)	Centreon	7,4	438,8

Service ping on SCK_SWITCH-5 is now down (timeout)	Centreon	6,9	315,5
Service ping on SCK_SWITCH-7 is now down (timeout)	Centreon	10,6	48,2
PROBLEM OTH_cmts_418 Arris_US_0.0_mer CRITICAL SNMP CRITICAL - US_0.0: *0* /10 dB	Centreon	7,9	1329,5
Service ping on SCK_SWITCH-7 is now up (ok)	The Dude	7,8	456,5
Service ping on SCK_SWITCH-5 is now up (ok)	The Dude	8,6	720,1
Service ping on SCK_SWITCH-4 is now up (ok)	The Dude	6,0	874,7
RECOVERY OTH_cmts_418 Arris_US_0.0_mer OK SNMP OK - US_0.0: 339 /10 dB	Centreon	14,2	503,1
Service ping on cham_X2_R2 is now up (ok)	The Dude	7,9	632,0

Tabel 2. Measurements of notifications delivery times.