Qosium Storage

Qosium Storage is the results system of Qosium. It comprises a QMCP receiver, a dedicated database server, and a web user interface for visualization and accessing the results. Results can also be accessed by using REST API. Storage is handy when results data grows large, and data access is wanted to be centralized. It is a key component in monitoring setups.

Table of Contents

1. Introduction

1.1. Basics

When Qosium is used for monitoring, statistics are generated continuously. Often, the monitoring setup consists of multiple simultaneous measurements, which can create a lot of monitoring statistics. Handling this by using separate results files becomes clumsy and, more importantly, lacks a way to give the user a complete picture of the status of the monitored system.

Qosium Storage is the answer to this need, providing permanent storage to the results and easy centralized access to them. Situation awareness rises to a new level as the QoS status of the monitored system can be seen as a whole.

Besides monitoring scenarios, Qosium Storage can also become handy in large-scale measurement setups.

In addition to Kaitotek's Qosium Storage, there is a possibility to use your own results system. If you are interested in this, check the QMCP integration possibilities.

The results stored in Qosium Storage can be accessed in various ways:

- QoS heatmap
- Customized overview reports
- List of measurements => raw results
- REST calls for precalculated overview statistics
- REST calls for raw results
- Direct database access

1.2. Structure

Qosium Storage is composed of the following main components:

- QMCP results receiver,
- Database, and
- Web server.

QMCP Receiver is a server listening to Qosium measurement results to be received in the defined port. The web server plays an essential role since it provides the web user interface to Qosium Storage. In addition, Storage supports direct REST calls for querying the results for integration with current applications and services. The results are stored in a database, also enabling direct access to the results with SQL queries.

All the components can be used and turned on and off separately. In the optimal case, they all are installed on the same platform, but it is not mandatory. The components can lie on different devices, making it possible, for example, to use your separate database with Qosium Storage.

1.3. Platforms

The database system of Qosium Storage is, by default PostgreSQL with TimescaleDB. Using other *SQL*-based solutions, however, are also possible. Please reach out to us for more information.

The recommended operating systems for Qosium Storage are **Ubuntu**, **Debian**, and **Redhat/CentOS**. Other operating systems are also possible, so if you have something in mind, ask Kaitotek support about it. In addition, if you have an existing supported database in your current environment, Qosium Storage can be configured to use that.

2. Preparative Work

2.1. Installation

2.1.1. From Installation Package (Debian-based)

In Debian-based systems, Qosium Storage installation is usually done from a deb installation file. The package is typically named `QosiumStorage_<version_details>.deb`, where the `<version_details>` part details depend on the version to be installed.

If you like to see what will be installed and where to, run:

```
dpkg --contents QosiumStorage_<version_details>.deb
```

To install a fresh or upgrade an existing Qosium Storage in a machine, open Terminal and run:

```
sudo dpkg -i  QosiumStorage_<version_details>.deb
```

Alternatively, you can use *apt* for installation:

```
sudo apt install ./QosiumStorage_<version_details>.deb
```

When running the installer, it checks for the dependencies and, if missing them, gives instructions on how to install them:

- Java,
- nginx (please reach out to us if you want to use another web server),
- PostgreSQL, and
- TimescaleDB.

The installation process asks some relevant details while installing:

- Local or remote database
- Web user username
- Web user password
- Database password

- Path for database (local database) or IP address to remote database

You can check which version of Qosium Storage is installed by running:

```
dpkg -l qosiumstorage
```

2.1.2. Manual Procedure

The installation package is unavailable for all platforms, but Qosium Storage will be installed manually. Kaitotek support will help you carry this out step-by-step.

2.2. Parameterization

After being installed, Qosium Storage is ready to be used, and parameterization is hardly needed. In case you do, see the instructions here.

2.3. Running

2.3.1. Control

Qosium Storage will be started automatically after the installation. It runs as a system service, controlled with *systemctl* commands:

Start:

```
sudo systemctl start QosiumStorage
```

Stop:

```
sudo systemctl stop QosiumStorage
```

Request status:

```
sudo systemctl status QosiumStorage
```

Disable Qosium Storage from starting automatically upon reboot:

```
sudo systemctl disable QosiumStorage
```

Enable Qosium Storage to start automatically upon reboot:

```
sudo systemctl enable QosiumStorage
```

2.3.2. License

Qosium Storage installation requires a license to operate. The License will be activated against Kaitotek's license server online. The process is fully automatic and is carried out when Qosium Storage is turned on. All you need to do is ensure that the device where Qosium Storage is installed has Internet access during

activation. Once done, Internet access is no longer required during the active License Period.

2.4. Feeding Storage with Results

2.4.1. The Procedure

Qosium Probes support direct results distribution to external systems such as Qosium Storage. Results sending is activated via a *measurement controller*.

When using Qosium Scope, the instructions to activate results sending are provided here. In the case of Scopemon, see how to enable results distribution. In addition, naturally, also Qosium Scope Lite can be used to activate the feature.

2.4.2. Identifying Qosium Measurements

Before you start pushing measurement results to Qosium Storage, think about how to organize measurement identifications so that it will be fluent in finding particular measurement results in the future. There are many ways to identify Qosium measurements, but three ways rise above others when dealing with Qosium Storage:

- **Service ID**,
- **User ID**, and
- **Measurement Description**.

Service ID is an identification of Qosium Probe that you can parameterize. Thus, it can be used in many ways. Often, it is used as a unique and static identifier of Qosium Probe. When used like this, it also identifies the device where the Probe is located. Further, if the device is, e.g., in a vehicle, and it is the only measurement point in that vehicle, **Service ID** identifies the vehicle. **Service ID** does not have to be unique depending on your use case. You can use it, for example, to identify a group of Qosium Probes. For instance, Probes in a specific network segment could use the same **Service ID** to identify that segment.

User ID is an identifier of the *measurement controller* that you can also set freely. For example, in Qosium Scope, it is set here. It provides another dimension in identification; again, there are various ways to use it. For example, you can use **User ID** to give all measurement controllers a unique ID. Another example is grouping. For instance, all VoIP QoS measurements could be grouped under the same **User ID**, while video QoS measurements could use another **User ID**.

Measurement Description is free text and can contain, besides some special characters, anything you wish. While **Service ID** and **User ID** are typically used as categorical/topological differentiators, **Measurement Description** is often used for details. It could contain, e.g., information on some nuances to separate otherwise similar measurements made with the same equipment. This parameter is defined in the measurement controller. For example, in Qosium Scope it is defined here.

All the presented measurement identifiers are stored with the measurement results. Thus, for a single measurement, you always have **Service ID**, **User ID**, and **Measurement Description**. All these can be used in results searches in Qosium Storage. Qosium provides different ways to identify measurements, and it is up to you how you use them.

3. Parameterization

There are a lot of parameters in Qosium Storage, but mostly you don’t need to touch them. We go through some important ones you might sometimes need to deal with.

3.1. Database

PostgreSQL database contains many parameters that affect performance. An automatic procedure will be run for optimizing them during installation, but you may also tweak them manually. The exact procedure of how to do this is out of the scope of this manual.

3.2. General Properties

Some general application parameters of Qosium Storage are found in the file `<path>/QosiumStorage/html/application.properties`. It uses the format:

`<parameter.name>=<parameter_value>`

Next, we go through some typically used parameters.

3.2.1. qosiumstorage.port

Defines the port (TCP) to which the results reception server is bound.

- Values: `<TCP port>` – Your specified port at range `1 - 65534`
- Default: `8888`

If Storage is behind a firewall and you wish to have access from outside, you have to open this port to the firewall.

3.2.2. qosiumstorage.averaging.qoe.selection

This parameter is used to define which QoE method is used in the general statistic calculation.

- Values:
  - `0` – PSQA is used
  - `1` – GQoSM is used
- Default: `1`

3.2.3. qosiumstorage.performance.report.interval

The *Status* tab in the Qosium Storage's UI shows internal status of the system. This parameter defines its reporting interval.

- Values: `1000 – 2147483647`
- Default: `10000`
- Unit: `milliseconds`

3.2.4. qosiumstorage.heatmap.area.left

This parameter is one of the four parameters defining the heatmap maximum visualization area. The parameter here defines the left side limit of the longitude axis.

- Values: `-180 – 180`
- Default: `20.61`
- Unit: `degrees`

3.2.5. qosiumstorage.heatmap.area.right

Defines the right side limit of the longitude axis.

- Values: `-180 – 180`
- Default: `31.65`
- Unit: `degrees`

3.2.6. qosiumstorage.heatmap.area.top

Defines the top side limit of the latitude axis.

- Values: `-90 – 90`
- Default: `70.22`
- Unit: `degrees`

3.2.7. qosiumstorage.heatmap.area.bottom

Defines the bottom side limit of the latitude axis.

- Values: `-90 – 90`
- Default: `59.65`
- Unit: `degrees`

3.3. Other Parameters

In addition to the ones presented above, there are many more parameters of Qosium Storage found in the file `<path>/QosiumStorage/html/application.properties`. Most of them, however, are such that you don't need to modify them. Reach out to Kaitotek support for more information on the parameters.

4. User Interface

This section guides you through how to interact with Qosium Storage.

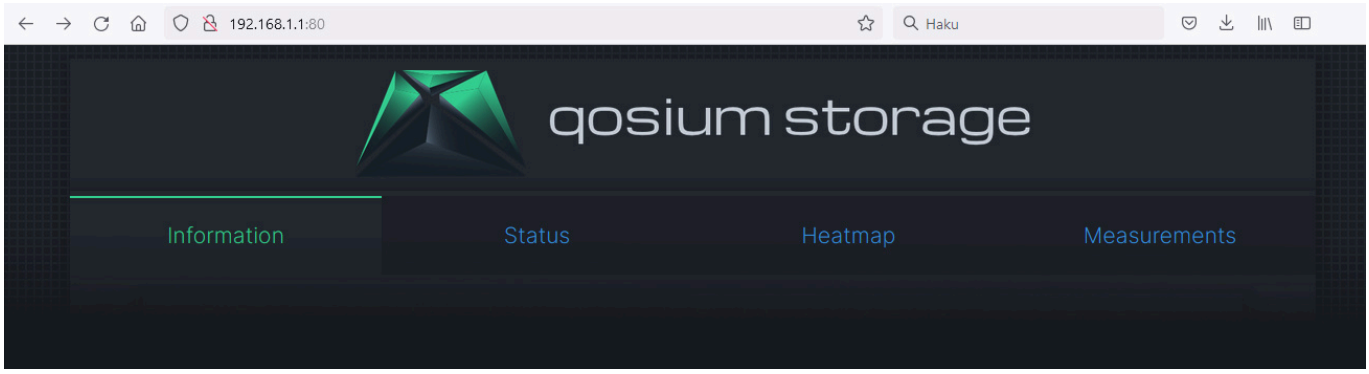
4.1. Accessing

Qosium Storage is used via a web browser. Thus, open your browser and type the address and port where your Storage's web server is bound. If you don't know Qosium Storage's port, check the parameters.

4.2. Tabs

4.2.1. Overview

When you enter Qosium Storage, you will land on the main page in the Information tab. There are typically three other tabs also visible.



Sometimes the Heatmap tab might be missing. If so, it has just been disabled. Via parameters, you can activate it. The next sections discuss the content of each tab.

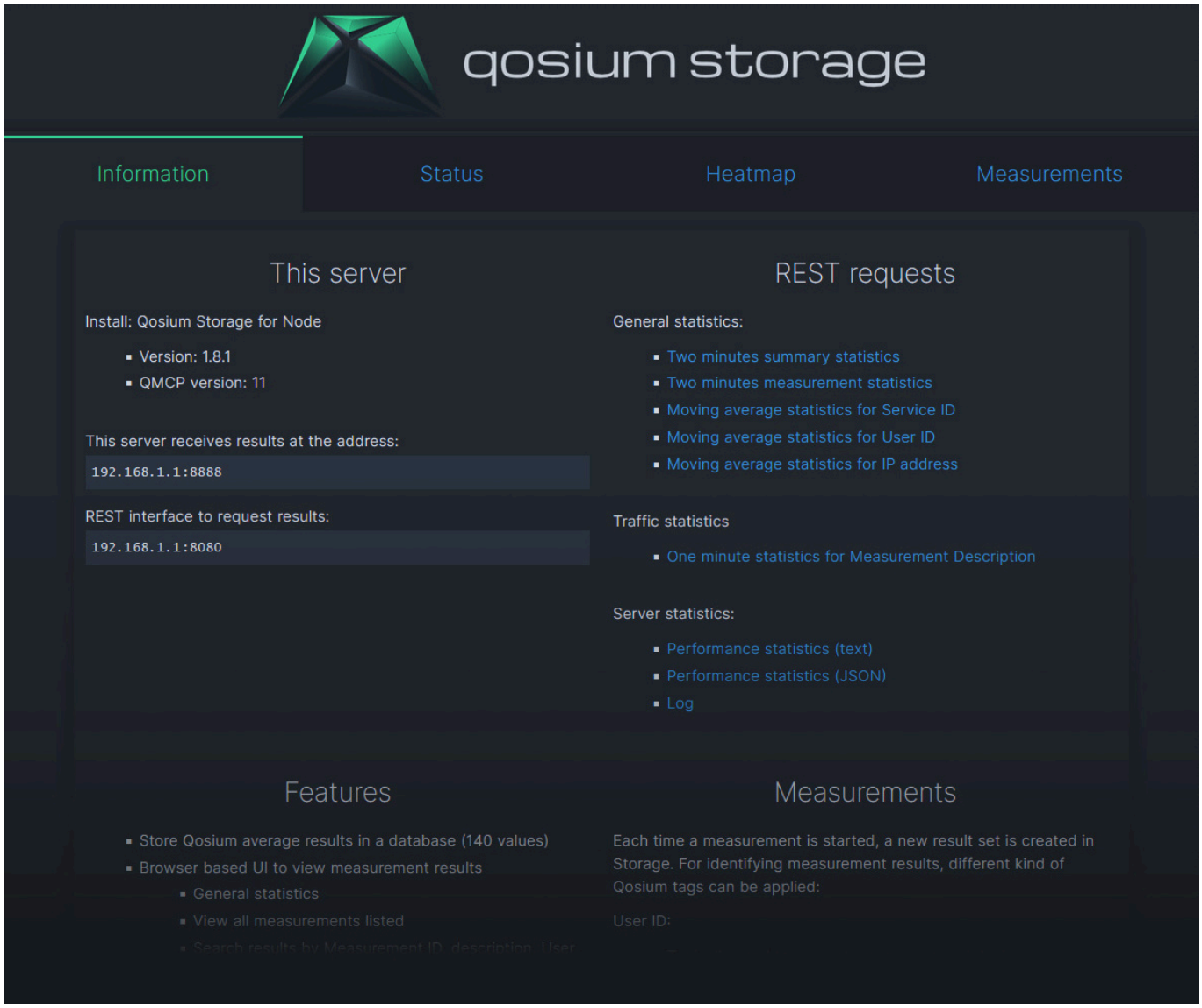
4.2.2. Information

Information tab is all about what the name says: information regarding the current installation of Qosium Storage. The information is split into four sections.

This server section tells the installation name, version, and relevant server ports. It also tells the QMCP main version, which is of importance since different main versions are typically non-interoperable. For instance, if you are running an outdated Qosium version of QMCP 10, you cannot send results to Qosium Storage with QMCP 11.

REST requests section shows some general REST requests available, and they can also be tested through the links. These are, however, not all the available REST calls, and more REST calls can be easily provided. In case you are interested, contact Kaitotek support to inquire about the *REST API documentation of Qosium Storage*.

Features section lists some of the main features of your version of Qosium Storage. And last, the *Measurements* section tells basic principles of how measurements and results in Storage correlate and how Qosium Storage uses the different identifications of Qosium.



4.2.3. Status

Status tab is the place to get up-to-date information on Qosium Storage itself. It is divided into three sections.

The first section shows ongoing measurements within the last two minutes period per Qosium's **Service ID**. As seen in the figure below, there are four ongoing measurements.

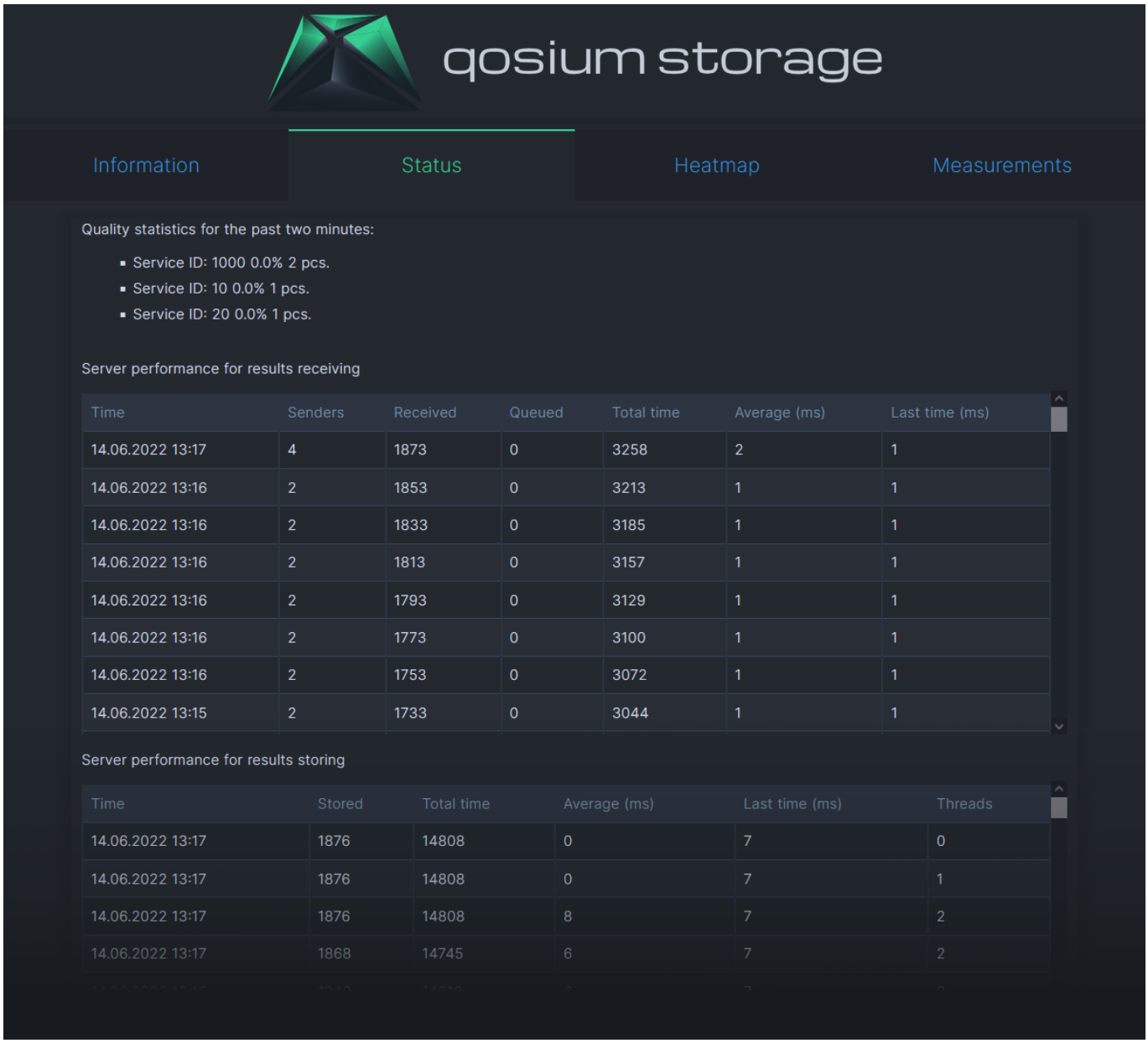
The second section shows the results reception performance. The observation interval is parameterized using this parameter. The columns are as follows:

- *Time* – When the performance calculation was made.
- *Senders* – The number of results senders (= measurements) that were ongoing
- *Received* – The total number of average results received
- *Queued* – The number of average results waiting to be stored
  - Regarding Qosium Storage's health, this is an important performance metric, as it indicates can Storage keep up with the pace the results are being received.
  - If the queue starts to pile up, it might be an indication that there are not enough computational resources for Qosium Storage to serve the current monitoring scenario.

- *Total time* – The total time spent for receiving the measurement results
  - Unit: ms
- *Average (ms)* – The time it took, on average, to receive a single result
  - This is yet another important performance metric.
  - A well-performing system should handle the reception procedures in one millisecond, and definitely < 10 ms.
  - Unit/resolution: ms
- *Last time (ms)* – The time it took to receive the last result
  - Comparing this with the previous performance metric gives an idea of how much there is deviation in the reception procedure durations between samples.
  - If the deviation is high, it is a potential indication of problems.
  - Unit/resolution: ms

The third section is about results storing performance. Most of the columns are analogous to those in the second section but now considering storing instead of receiving. Thus, these performance metrics tell how the database storing processes are functioning.

The only column having no corresponding one in the second section is *Threads*. It tells how many computational threads are currently active connecting and writing to the database. In heavy-load situations, there can be several tens of threads. This is mostly informative and typically not an indication of performance problems. Instead, the storing times (*Average (ms)* and *Last time (ms)*) are mostly the ones to follow.



4.2.4. Heatmap

Heatmap is handled in its own article because it is a larger entity.

4.2.5. Measurements

Measurements tab shows a list of all the stored measurements as a table, including some key elements of measurement identification.

The first column is the Storage's running number of the measurement. By clicking that, you can access the measurement data shown as a large table, potentially of several pages, depending on the length of the measurement. While this might be useful for quick evaluation, detailed analysis often requires downloading the measurement data, and the last column of the Measurements tab can do that.

Measurements tab also shows a quick glance at the overall quality in the *Stats* column. Starting and ending times of the measurement are shown in columns *Started* and *Ended*, respectively. If the *Ended* column shows *On-going*, the measurement has not yet finished. Sometimes it can happen that the network connection between Qosium Storage and the Qosium Probes carrying out the measurement has gone

down. As a result, no measurement end notification arrives at Qosium Storage. In this case, Qosium Storage shows the measurement as ongoing for some time until, because no new results come, a judgment is made that the measurement must have ended.

Each measurement has a unique Qosium Storage identifier string in the second column. **Service ID**, **User ID**, and Probe (Primary one) are related to measurement identification, shown in the respective columns. Those can also be clicked for instant search. For example, by clicking *Service ID 1001*, performs a search listing all measurements carried out with *Service ID 1001*. You can execute custom text searches in the upper left corner. In addition to the different IDs, the Description field is also included in the search.

4.3. Heatmap

Heatmap is the most visual element of Qosium Storage, raising performance monitoring of wireless networks to a new level. While the heatmap is at its best when dealing with wireless networks, it can also serve fixed network monitoring since Qosium Probes can be given a fixed location.

4.3.1. General

To use Storage's heatmap functionality, you need Internet access because the map is downloaded online dynamically.

The metric visualized is Qosium's QoE, i.e., GQoSM or PSQA. Main coloring is logical:

- No color: no results
- Green: Good performance
- Yellow: Average performance
- Red: Bad performance

In practice, there are countless shades between the main colors to indicate how bad or good the performance is.

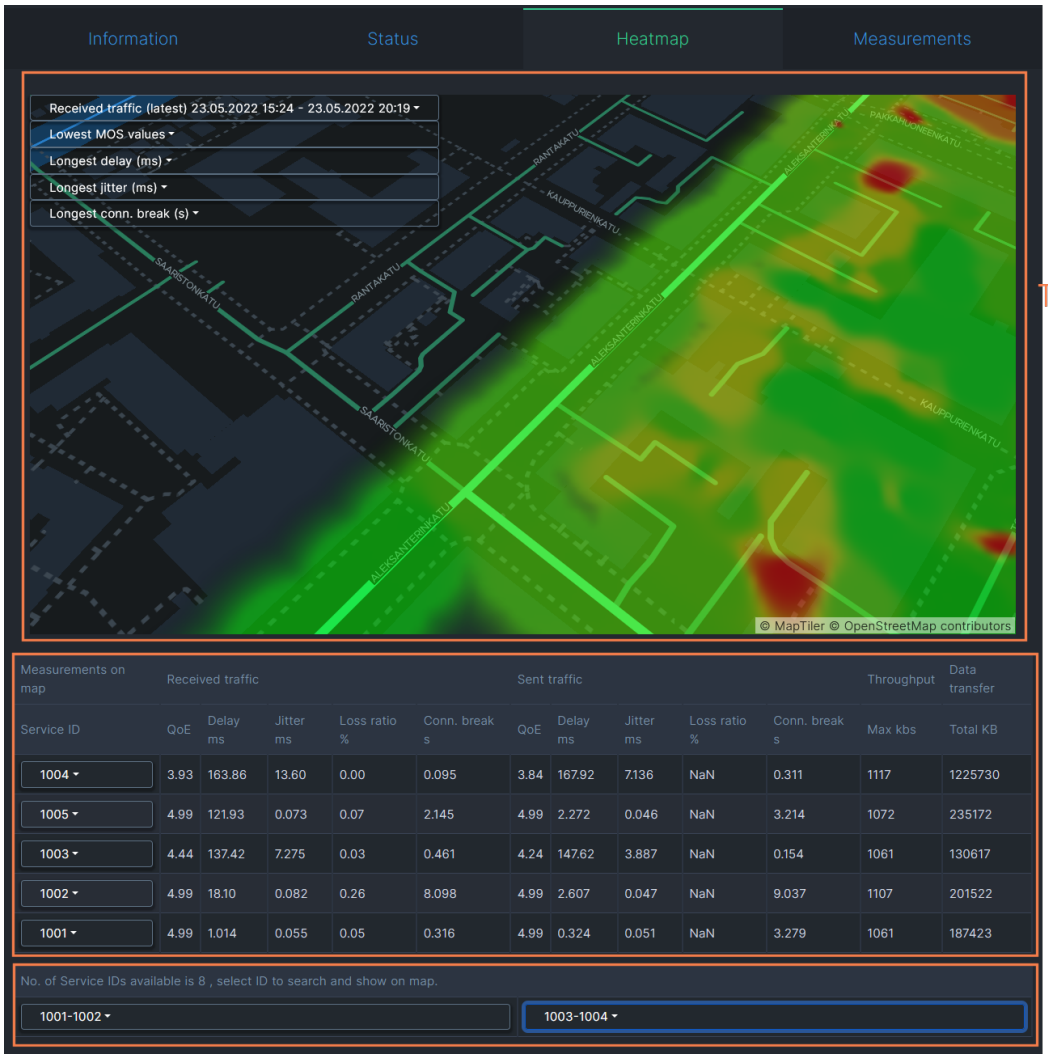
4.3.2. Control

Controlling the map is easy and similar to the heatmap of Qosium Scope:

- Zoom in/out:
  - Keyboard: +/-
  - Mouse: wheel
- Move:
  - Keyboard: the arrow keys
  - Mouse: left button + moving
- Rotate / tilt
  - Keyboard: shift + the arrow keys
  - Mouse: right button + moving
- Selection zoom
  - Mouse+keyboard: shift + left button + moving

4.3.3. Sections

The *Heatmap* tab is divided into three sections:



The map section

Service ID performance summary & visibility selection

Optional Service ID selection

Qosium Storage's heatmap is currently tightly bound to Qosium Probes' **Service ID**. That is because **Service ID** is often used to distinguish devices in the field, making it a device ID. Thus, **Service ID** is the natural identifier according to which you can select results in the map. In addition to selection possibilities, the middle section in the Heatmap tab shows QoS summaries per **Service ID**.

4.3.4. Select What to Show

4.3.4.1. General

There are many ways to select what to show on the map. The two main dimensions according to which you can select results are currently:

- Time and
- **Service ID**.

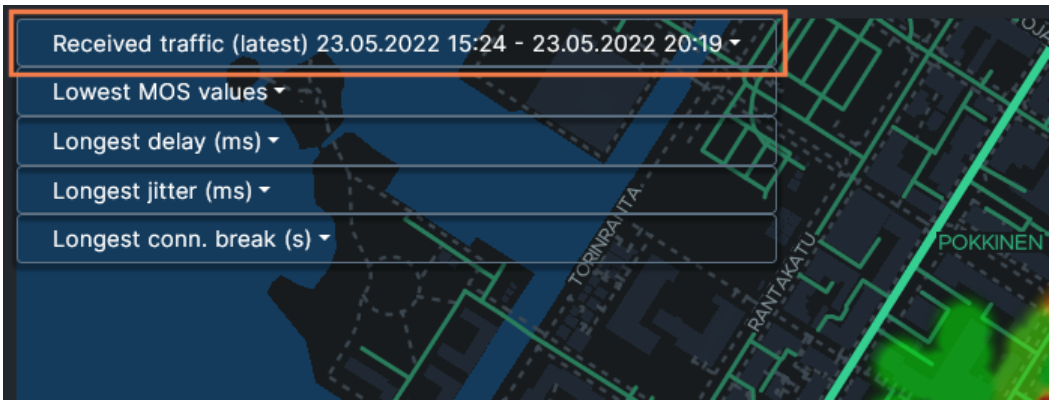
In addition, you can select that does the visualized quality consider sent or received traffic.

The typical workflow is to select the interesting time window and then filter the **Service ID's** you wish to observe.

4.3.4.2. Time Selection

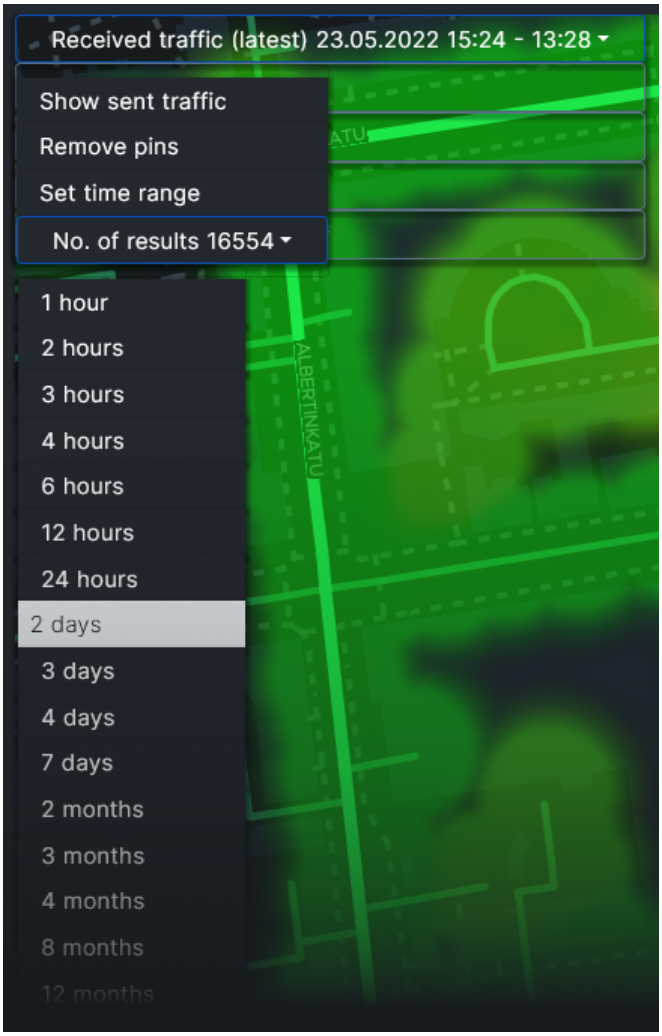
The time window from which to show results on the map can be a real-time window or your selection in the

past. The currently visible time window is shown in the top row of the control panel, located in the upper left corner of the map:

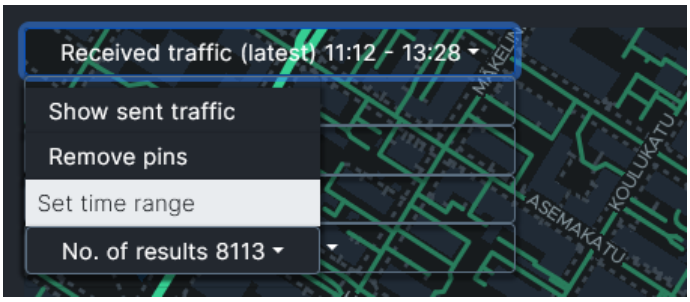


When the text ”latest” is visible, it means that the view is in real-time mode:

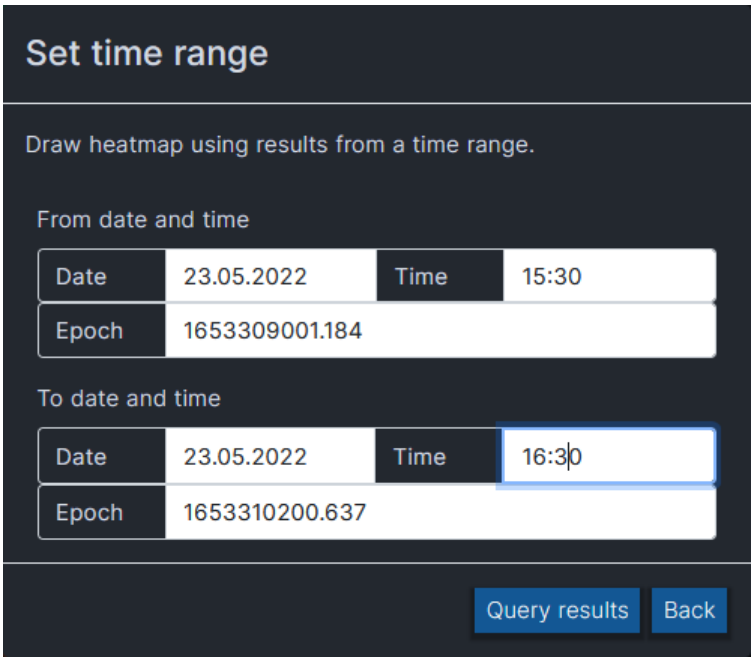
To set the time window in the real-time mode, click the top row of the control panel first. A smaller menu opens up. There, click the lowest item, Number of results. Now you see that a list opens where you can pick the size of the real-time window., e.g., two days:



To show a specific time range in the history, click the top row: the menu pops up again. Then select *Set time range*:



A separate box opens, letting you set the time range. Set the desired lower bound (From) date and time and then the desired upper bound (To) date and time. When done, press *Query results*.

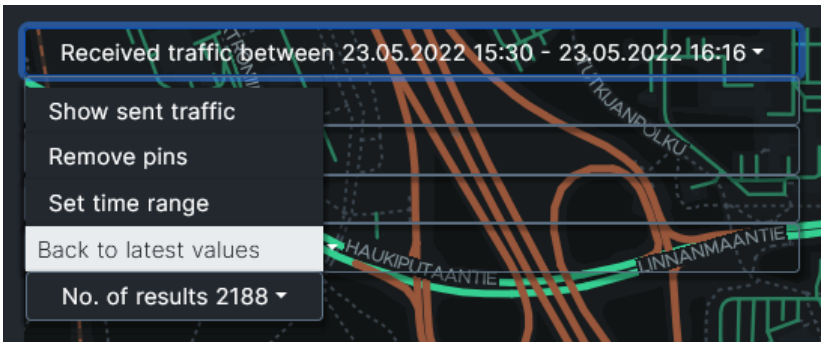


After fetching the results, a new heatmap is drawn with the newly selected time range.

After setting the time range like this, the map no longer updates in real-time.

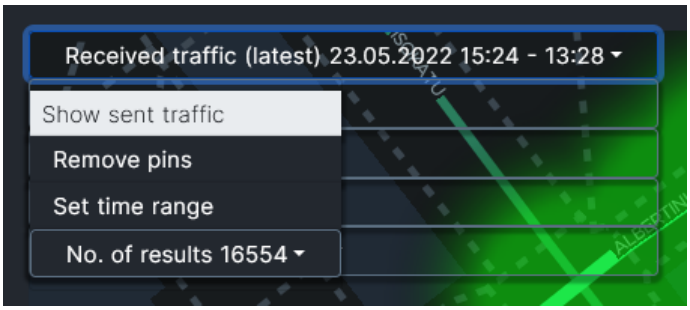
If the selected time range contains results only in a certain shorter interval, then that interval will be shown instead of the set time range.

If you want to get back to the real-time view, click the menu open again and select the new option: *Back to latest values*.



4.3.4.3. Direction

The control panel's top row tells the current direction of the visualized traffic. To change it, click the top row: the already familiar menu pops up. Then select the other direction, e.g., sent traffic:



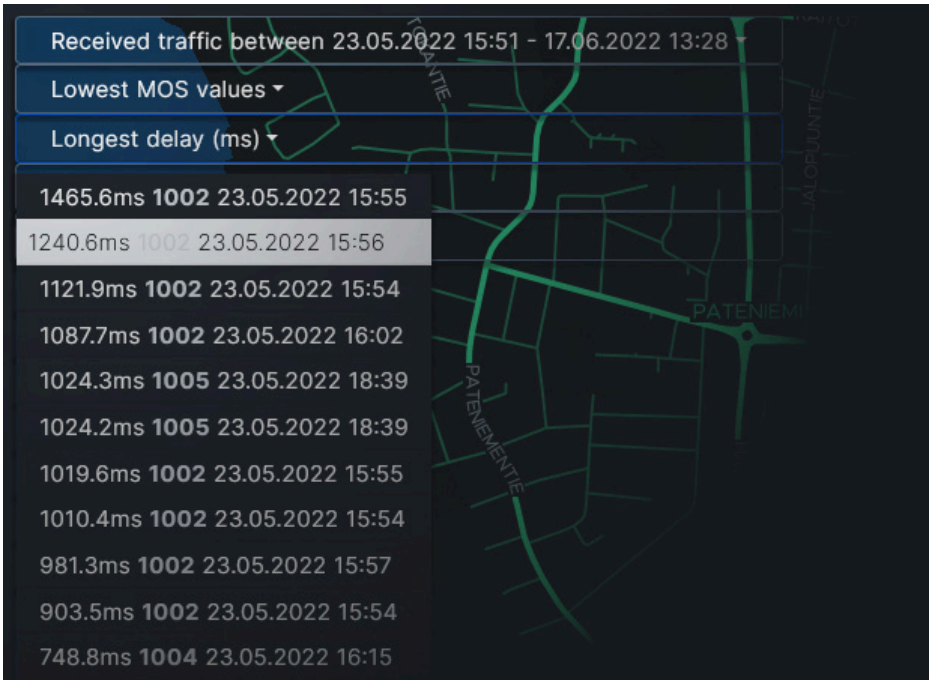
4.3.4.4. Service ID selection

By default, the heatmap shows all the results in the selected time interval. Results can, however, be filtered in and out according to **Service ID**. That is done in the *Measurements on map* table below the heatmap. The rows show key statistics averaged over the **Service IDs** shown in the first column. Click the desired **Service ID** to get a pop-up list of three items:

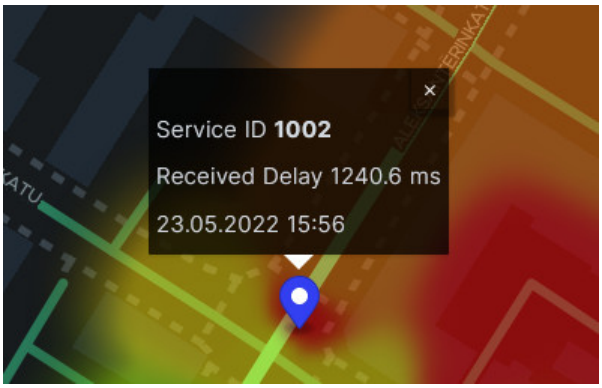
- *Show on map*: Adds a pin pointing the last location of the set on the heatmap.
- *Hide from map*:
  - Hides the results set from the heatmap.
  - The results set of this **Service ID** is transferred below in the list of hidden **Service IDs**.
  - By clicking the **Service ID** in the list below, you get the results back on the map.
- *Show this only*:
  - Hides all the other results and shows only results with this **Service ID**.
  - To restore all the results, click the **Service ID** and select *Show all*.

4.3.5. Pinpointing Values

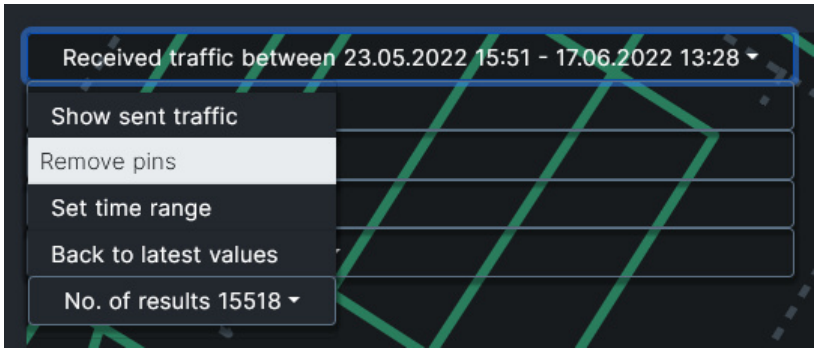
There is a nice feature in the heatmap that allows you to pinpoint interesting events. Just below the time range in the control panel, you can open drop-down selections of extreme values of different statistics. For example, in the *Longest delay (ms)* selection, you can see the longest individual delay values among the results currently shown. By selecting one of the values, you get a pin on the map revealing the location of this occurrence.



There can be multiple pins on the map at the same time. By clicking a pin, you get information about that particular measurement result.



To remove the pins on the map, open the drop-down menu of the control panel and select *Remove pins*.



5. Direct Access

You can integrate the results stored by Qosium Storage into your applications and services. You can access the data and specific reports via REST API or access Storage's database directly.

5.1. REST Interface

The *REST* interface of Qosium Storage contains multiple methods to fetch raw measurement results as well as averaged statistics directly without Qosium Storage's UI. The REST API documentation is available as a separate document. Please ask Kaitotek support about that if you find this interesting. Some REST calls are presented in the Information tab.

5.2. Database Access

The most efficient way to access Qosium measurement results raw data at Qosium Storage is to access the database directly. We recommend this approach, especially if you need to poll the raw results data often and, potentially, automatically in real time. Qosium Storage typically uses PostgreSQL with TimescaleDB extension, so *SQL* calls can be used to query the data. Read more about PostgreSQL here. Ask Kaitotek for assistance, if you are new to *SQL*.

6. Glossary

Structured Query Language

A language used in programming and designed for managing and processing data held in a relational database management system.

Representational State Transfer

A software architectural style that was created to guide the design and development of the architecture for the World Wide Web.