Tool selection

The purpose of this document is to provide an overview of the tool selection process. It will establish what different areas tools, what are the criteria for selecting a specific tool and what tools are available for each area.

# Area selection

The first step in our tool selection is determining where we can use external tools to improve our application. This is done by analyzing our use cases and the systems will be monitoring. After the analysis there has been determined that there are tree area where already existing tools can be used for our monitoring application.

## Data storage

To display the collected metrics of the application for 4 weeks there needs to be some way to store for the visualization to have data to pull from after the fact.

## Hardware monitoring

Some of the important metrics of the application are the remaining storage, ram and CPU usage. There are many tools that can be used to get the hardware metrics of multiple nodes easily.

## Visualization

Using an already preexisting tool to display the information in a web interface will be significantly easier

# Sources

In order to make sure that our search is as good as possible we have decide to use information only from the following sources in order to make sure our tool selection is not affected by misinformation and is as objective as possible.

* **Stack share** is the most popular website that specifically deals with finding different tools for applications. It is used by many companies and developers and shows what companies utilize which tools which is a useful metric to determine if a tool is widely used professionally.
* **The websites of the tools themselves** will be the primary source of information regarding the features and capabilities of each tool
* **Slant** a website designed for comparing different software tools and their upsides and downsides.
* **LibHunt** a website that provides curated lists, alternatives and comparisons for different tools. Is reliable doe to its size. Is a good indicator of a tools ecosystem since it includes GitHub metrics.

# Criteria

This chapter contains a list of criteria for each area and how each criteria is measured. Each non-knokout criteria will have a score and at the end we will chose the options with the highest score.

It is important to mention that a **knockout criteria** is a criteria that if false or below a certain score will automatically disqualify the tool form being used.

## General criteria

These are criteria that apply for any tool in any area.

**Name:** Ecosystem

**Explanation:** The more widely used a tool is the easier it is to find information on how to use it such as tutorial or blogs and the easier it is to solve problems since there is a higher chance that other people have had a similar problem and solved it. For this criteria we will prioritize projects that have been popular recently since this reduces the chance that the tool is outdated.

**How this criteria is measured:** This criteria’s score will be generated using two metrics the number of stars on GitHub and the activity score on LibHunt.

The number of stars on GitHub is a good indicator of how widely popular a project is. For this criteria the points will be added in the following way:

* 5 points 40.000+ stars
* 4 points 30.000-40.000 stars
* 3 points 20.000-30.000 stars
* 2 points 10.000-20.000 stars
* 1 point 5.000-10.000 stars
* 0 points 0-5.000 stars

The lib hunt activity score is based on recent metrics such as commit frequency, issue and pull request activity and community engagement (number of discussion and responses to issues). This makes the LibHunt activity score a good indicator of the how currently popular and up to date the tool is. The points for this criteria from this metric will be added in the following way:

* 5 points 9-10 score
* 4 points 7-9 score
* 3 points 5-7 score
* 2 points 3-5 score
* 1 point 1-3 score
* 0 points 0-1 score

The points of this metrics are added together to form the final score of the criteria which will be between 0-10.

**Name:** Free of cost

**Explanation:** The company does not want to pay for any additional tools that will be using in the creation of the monitoring application as a result we will only be using free and open-source options. This is a **knockout criteria**.

**How this criteria is measured:** True or false. False if it cost money to use.

**Name:** Environment compatibility

**Explanation:** This criteria is meant to represent if the tool selected works well with other tools that will be used. We define “works well” as having features or plugins that make it easier to use alongside other tools in the creation of the application.

**How this criteria is measured:** 3 points will be assigned for each other tool the selected tool “works well” with. These points will only be added if the tool it works well with another tool that is utilized.

**Name:** Wildcard

**Explanation:** This is meant to represent any unique features or problems a tool has that are not covered by other criteria.

**How this criteria is measured:** A score from -3 to +3 point depending on the specific tool.

## Monitoring Data Storage

These are the criteria for the data storage tools.

**Name:** 4 week storage

**Explanation:** The data storage must be capable of storing only the last 4 weeks of data and automatically deleting older data as to not occupy too much space. Is a **knockout criteria**.

**How this criteria is measured:** True or false. False if it can’t automatically delete data older than 4 weeks.

**Name:** Time series

**Explanation:** For this project we require any database used to be a time series database. This is because time series databases make it easier to keep track of the timing of different metrics making the excellent for a monitoring solution. This is a **knockout criteria**.

**How this criteria is measured:** True or false. True if the tool is a time series database.

**Name:**  Speed

**Explanation:** Because the database will not be handling particularly large amounts of data and the fact that measuring the exact performance for this project will be very difficult it was decided to not use speed as a criteria for this project.

**How this criteria is measured:** Not applicable

## Hardware Monitoring

**Name:**  Hardware Monitoring

**Explanation:** The tool used must be able to monitor the remaining ram, memory and the cpu usage. This is a **knockout criteria**.

**How this criteria is measured:** True or false. True if it can monitor all tree metrics.

## Visualization

**Name:**  Web Display

**Explanation:** For ease of use the tool must be able to display the information through a web interface to make it easy to access for the customers. This is a knockout criteria.

**How this criteria is measured:** True or false. True if it can display the information rough a web interface.

**Name:**  Real Time Display

**Explanation:** In order to quickly address problems the visual interface must be able to change in real time to reflect changes in the monitored application. This is a **knockout criteria**.

**How this criteria is measured:** True or false. True if it can modify itself in real time.

**Name:**  Display features

**Explanation:** The tool should have the features necessary to display the data in concise and easy to understand way.

**How this criteria is measured:** This criteria is measured based on the available feature of the tools. Each helpful feature will have a number of points assigned to it and the final score will be made of the sum of the features the tool has:

* Can display pop ups - 1 point
* Can display text messages – 3 points
* Can switch between multiple pages – 1 point
* Can display traces – 3 points
* Can display graphs – 2 points

# Possible Tools

## Data Storage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Good ecosystem | Works with other tools | Wildcard | Total |
| Prometheus | 10 | 6 | 2 | 18 |
| Timescale DB | 7 | 3 | 1 | 11 |
| Quest DB | 7 | 0 | 0 | 7 |
| Influx DB | 8 | 3 | 0 | 11 |

Explanation for scores:

**Tool Name:** Prometheus

**Good ecosystem:** 5 points for GitHub stars and 5 points for activity score

**Works with other tools:** 3 points for working with Grafana and 3 additional for working with Prometheus node explorer for a total of 6 points if both tools are picked.

**Wildcard:** 2 points thanks to the fact that Prometheus is considered easier to use than its alternatives

**Tool Name:** Timescale DB

**Good ecosystem:** 2 points for GitHub stars and 5 points for activity score

**Works with other tools:** 3 points for working with Telegraph and 3 points for working with Zabbix because Telegraph and Zabbix cannot be both selected Timescale DB receives the only tree points.

**Wildcard:** 1 point for the fact that Timescale DB uses the Postgres SQL as it’s query language that the developer of this project is already familiar with.

**Tool Name:** Quest DB

**Good ecosystem:** 2 points for GitHub stars and 5 points for activity score

**Works with other tools:** does not work particularly well with other tools

**Wildcard:** Nothing notable

**Tool Name:** Influx DB

**Good ecosystem:** 3 points for GitHub stars and 5 points for activity score

**Works with other tools:** 3 points for telegraph

**Wildcard:** Nothing notable

## Hardware Monitoring

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Good ecosystem | Works with other tools | Wildcard | Total |
| Prometheus Node Exporter | 6 | 3 | 0 | 9 |
| Telegraph | 7 | 3 | 0 | 10 |
| Zabbix | 6 | 3 | 0 | 9 |

Explanation for scores:

**Name:** Prometheus Node Exporter

**Good ecosystem:** 2 points for GitHub stars and 4 points for activity score

**Works with other tools:** 3 points for working with Prometheus

**Wildcard:** Nothing notable

**Name:** Telegraph

**Good ecosystem:** 2 points for GitHub stars and 5 points for activity score

**Works with other tools:** 3 points for working with Timescale DB

**Wildcard:** Nothing notable

**Name:** Zabbix

**Good ecosystem:** 1 point for GitHub stars and 5 points for activity score

**Works with other tools:** 3 points for working with Timescale DB

**Wildcard:** Nothing notable

## Visualization

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Good ecosystem | Works with other tools | Display features | Wildcard | Total |
| Grafana | 10 | 3 | 10 | 2 | 25 |
| Apache Superset | 10 | 3 | 7 | 0 | 21 |
| Metabase | 9 | 3 | 6 | 0 | 18 |

Explanation for scores:

**Name:** Grafana

**Good ecosystem:** 5 points for GitHub stars and 5 points for activity score

**Works with other tools:** 3 points for working with Prometheus

**Display features:** 10 points for having all display features

**Wildcard:** 2 points thanks to the fact that Grafana is considered easier to use than its alternatives

**Name:** Apache Superset

**Good ecosystem:**5 points for GitHub stars and 5 points for activity score

**Works with other tools:** 3 points for working with Timescale DB

**Display features:** 7 points for all display features except trace display

**Wildcard:** Nothing notable

**Name:** Metabase

**Good ecosystem:** 4 points for GitHub stars and 5 points for activity score

**Works with other tools:** 3 points for working with Timescale DB

**Display features:** 6 points for all display features except trace display and pop-ups

**Wildcard:** Nothing notable

# Tool selection

For each category the highest score that is held by Prometheus, Telegraph and Grafana. Prometheus and Grafana work well with each other and are easy to use, but Telgraf does not work with Prometheus. Because of this the actual option with the highest score is the one that works with Prometheus, the Prometheus node exporter.

The final selection is Prometheus node exporter, Prometheus and Grafana.