Heidi Ruhtinas

DEVELOPING THE METRICS AND STATISTICS MONITORING FOR A DIGITAL ARCHIVING SYSTEM

Bachelor's thesis

Bachelor of Engineering

Information Technology

2022

South-Eastern Finland
University of Applied Sciences

South-Eastern Finland University of Applied Sciences

Degree title	Bachelor of Engineering
Author (authors)	Heidi Ruhtinas
Thesis title	Developing the metrics and statistics monitoring for a digital archiving system
Commissioned by	Disec Oy
Year	2022
Pages	35 pages
Supervisor	Timo Mynttinen

ABSTRACT

The main objective of this thesis was to improve the current metrics and statistics monitoring methods for the digital archiving system Yksa by centralizing the separate data collection and management processes under one monitoring system. The secondary objective was to provide suggestions for future improvements and expansion of data collection, mostly to include general statistics and business metrics.

To achieve this, some background research about statistics, business metrics and data monitoring systems was carried out before choosing a monitoring system to migrate the current statistics and metrics collection to. The suggestions given were also based on the insight gained from the research. The practical implementations were mostly executed using the Java programming language and Flux query language, in addition to deploying local instances of InfluxDB and Grafana.

The main goal was achieved despite some complications arising due to the nature of the current data collection methods for specific data points. The secondary goal was achieved as well, but remained as a rather limited example of how to proceed from here, instead of providing a proper example of what the data collection could be in its full extent. Furthermore, despite the research done on business metrics, more business insight is needed to make the right choices when choosing to expand the data collection to business metrics.

Keywords: data, digital archiving system, metrics, statistics

CONTENTS

1 INTRODUCTION

Yksa, the digital archiving system of Disec Oy, is currently gathering and processing some elementary per-organization statistics, as well as Application Performance Monitoring (APM) metrics about the performance and status of the services and processes of Yksa. However, the process of collecting metrics and statistics is not centralized – currently, metrics are being collected using one system and statistics using another. Using different systems makes managing the metrics and statistics monitoring unnecessarily complicated. Also, the amount and type of data collected is very limited and has room for improvement.

The main goal of this thesis is to improve the collection and processing of metrics and statistics by centralizing the two processes under one monitoring system intended for such a task, thus making the methods more comprehensible for all Yksa's developers at Disec Oy.
The secondary goal is to provide some suggestions on how to further develop the data collection from the business perspective.

To achieve this, statistics and business metrics as a general concept will be studied first. The difference between metrics and statistics is defined, giving the task at hand a proper frame. A quick glance on the importance of monitoring and analysing metrics and statistics will provide motivation and justification for improving the current methods in Yksa, as well as insight for what needs to be considered in the Yksa case.

An overall description of Yksa and its current metrics and statistics monitoring methods will be given to have a clear idea of the current situation and what specifically needs to be improved. The metrics and statistics relevant to Yksa's functionalities will be defined, even if not all of it can be currently monitored.

Three different monitoring systems will be compared, keeping in mind the criteria defined in the previous step. After the choice has been made, the monitoring system will be set up and implemented in Yksa.

Finally, the current statistics and metrics collection will be migrated to the chosen system, refactoring the code as necessary.

Improving the business operations with the help of business metrics analysis right away is beyond the scope of this thesis, but the current system will be developed with scalability in mind, so that the improvements can be easily implemented in the future.

2 METRICS AND STATISTICS

Metrics and statistics have different meanings, methods and applications depending on the context. While the metrics collected in Yksa are APM metrics and therefore focus on technical details, in this thesis, metrics will be researched from the business perspective, as the data in question comes from a product (Yksa, its underlying processes and the actions and behaviour of its users) and could ultimately be monitored and analysed with the goal of improving Quality of Service as well as the profitability.

This chapter will describe the difference between metrics and statistics and take a closer look at both as well as the reason why metrics and statistics are important for a business in general.

2.1 Difference between metrics and statistics

To be able to properly define the scope of desired metrics and statistics, one must first learn the distinction between the two.

According to Gallagher (2019), statistics can be used to measure the performance of either an individual or a group of staff members, while metrics are an overall quantifiable measurement that is used to monitor whether the processes can be defined succeeded or failed. Campbell (2010) emphasizes that statistics capture the current situation and metrics tells how well the agent or group is performing.

Gallagher speaks from the business perspective in general, while Campbell's frame of reference is call centers, but the same idea remains; statistics focus on current data and details, while metrics show what is being done right and what needs to be improved.

Table 1 provides some simplified examples of statistics and metrics to further demonstrate the difference in their nature. However, the distinction between the two is not always as absolute. Some parties consider the amount of phone calls to be a statistic only, other ones say it's a metric by itself. In the end, it depends on the context and the goals of the business.

Table 1. Examples of statistics and metrics

Context	Statistics	Metrics
Hotel	# of rooms in a hotel	% of rooms booked in a hotel per night
Sales department	# of calls made a day	% of calls resulting in new customers
Business profitability	Total income	Total revenue (= income - costs)
Marketing	# of people clicking an online advertisement	% of people who made a purchase on the website after having clicked the advertisement
Customer satisfaction	# of feedback collected	% of people who rated a service 4 or 5 out of 5 stars

When Table 1 presents a metric with a percentage, it can be seen as a division between two statistics. For example, to get the percentage of rooms booked in a hotel for a night, we divide the number of booked rooms with the number of all existing rooms. Neither of those numbers tells much alone. 15 booked rooms is impressive for a small family-owned hotel with 16 rooms in total, but not so much for a massive chain hotel with over 200 rooms. However, when we inspect the ratio between these two numbers, it gives us insight about how well the hotel has managed to acquire customers for a given period of time.

With the differences roughly explained, both metrics and statistics will now be studied in more detail.

2.2 Metrics

There are several ways to divide the important performance metrics into categories. One of them contains four categories: business performance metrics, sales performance metrics, project management performance metrics and employee performance metrics (Indeed.com 2022). Each of them has a few important metrics, as listed in Figure 1.

As this thesis focuses on Yksa, not the overall business operations of Disec Oy, not all the metrics presented in Figure 1 will be explained in detail, but the four categories will be briefly described as they partially overlap with the goals of metrics and statistics monitoring in Yksa.

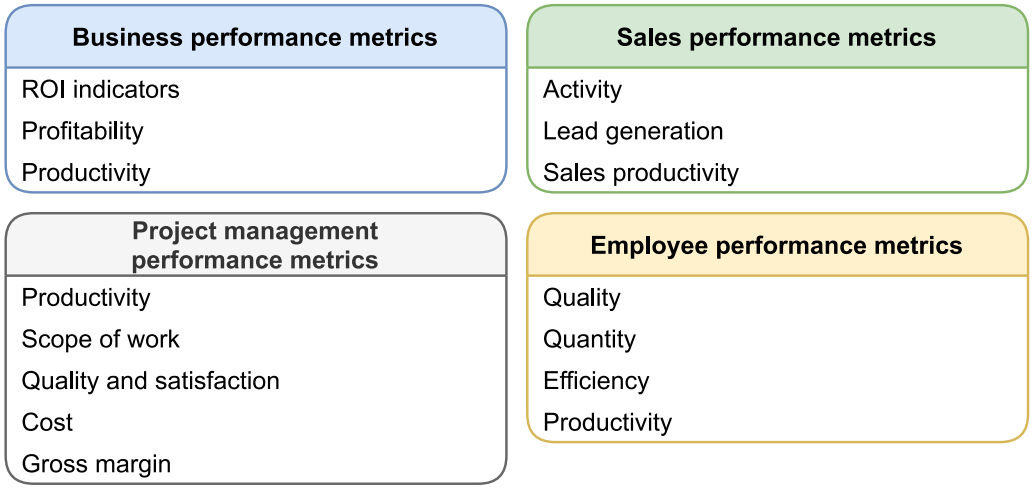

Figure 1. Four categories of performance metrics (Indeed.com 2022)

Business performance metrics track processes such as marketing, sales and profitability. ROI (Return on Investment) indicators are important, because they determine whether an investment is profitable or not. If a company is doing several investments, ROI indicators help define which ones are worth further pursuing and which ones are barely profitable or even result in a loss. Profitability, on the other hand, is not tied to investments. It tracks the profit margin on a more general level and can be analyzed to determine whether the sales methods require changes or not. (Indeed.com 2022.)

Sales performance metrics, as the name implies, focus on sales, lead generation and retention, as well as Key Performance Indicators (KPIs) such as total revenue or customer reach (Indeed.com 2022).

Project management performance metrics measure the profitability and effectiveness of a project throughout its lifecycle and can be utilized to make decisions about the completion of the project. They also track customer satisfaction, cost management and productivity. (Indeed.com 2022.)

Employee performance metrics focus on the productivity and efficiency of the employees and allow to monitor whether the employees meet their goals (Indeed.com 2022).

Metrics can be complicated to understand and very difficult to master. Sometimes they require data that may not be even available in a complete, exact form. Also, as no metric is likely to be perfect by itself, it's recommended to use a portfolio of metrics, so the metrics give each other some context and perspective. (Bendle et al. 2016, 3.)

From a business perspective, metrics give more important insight about the business operations, but as metrics are ultimately built on top of statistics, and the current statistics collection in Yksa could also be improved, they will also be briefly studied in more detail.

2.3 Statistics

As stated earlier, statistics have varying meanings depending on the context. Statistics is its own field of science by itself, but in this thesis, it refers to the plural of the word statistic – plain, numerical data that can be extracted from Yksa's operations, services and users. While mean values, for example, might be observed in addition to the raw data, the statistics used will not go into further details outside the scope of this thesis, such as making predictions based on the data or analyzing correlations and complicated distributions.

Figure 2 illustrates the primary types of statistical data.

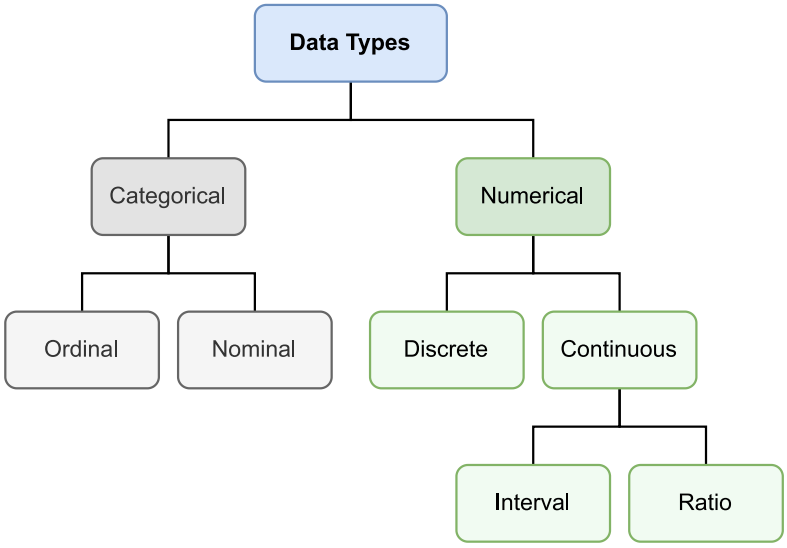

Figure 2. Data types (Donges 2019)

Categorical data constitutes characteristics, such as gender or language. The data may have numerical values, but the numbers don't have mathematical importance, as they are mapped to their true meanings (e.g., 0 for male, 1 for female). Categorical data is divided in two subcategories: ordinal and nominal data. Both can be considered as labels, but in the ordinal data, the order matters. For example, languages spoken by a person have no hierarchical order, making it nominal data, but customer satisfaction (very unhappy / unhappy / neutral / happy / very happy) has a clear order and therefore is ordinal data. However, sometimes the order isn't unambiguous, which is the main limitation with the ordinal data. (Donges 2019.)

Numerical data represents scalar values instead of labels. Discrete data can be counted, but not measured (for example, number of heads in a coin flip). Continuous data, on the contrary, can be measured but not counted (for example, a person's height). (Donges 2019.)

Continuous data consists of interval and ratio data, both of which represent ordered units that have the same difference, but the distinction between the two is the absence of "true zero". For example, temperature in Celsius or Fahrenheit

is considered to be interval data, as we can have a temperature of -10 °C or °F. Weight, on the other hand, does have a true zero, making it ratio data, as an object cannot weigh -10 kg. (Donges 2019.)

Yet another type of data exists – cyclic. It represents modulo data, such as angles or clock time. It has the unique property that two different points can be equal, for example, 0° and 360° fall in the same position on a circle. (Smith 2021, 35.)

2.4 The importance of monitoring and analyzing metrics and statistics in a business

In addition to the everyday responsibilities of running a business, management also needs numerical fluency to select, calculate and explain important business metrics in order to justify the financial risks and desired outcomes of their decisions (Bendle et al. 2016, 2).

However, metrics and statistics don’t concern management alone, as they are important tools to motivate the staff to take more responsibility and feel more equal and valuable in a company (Gallagher 2019). When the staff members can see the positive impact of their hard work, they feel incentivized to continue reaching their goals.

Often, metrics are monitored with cost management in mind, but depending on the company and its operations, they can also be utilized for various other goals, for example determining whether a marketing campaign was successful, finding new leads and opportunities, and monitoring the overall financial statistics of the company (Lutkevich & Ehrens 2022).

Whatever the goal is, metrics play a big role, as they provide invaluable insight to help make better decisions, and to determine which areas are performing well and which need more focus and support. However, while business metrics provide important quantifiable information, the data and numbers alone don’t tell

much without a context. To understand and interpret metrics, existing benchmarks, practices and objectives are used. (Lutkevich & Ehrens 2022.)

There is a risk of focusing on the wrong metrics, too. Years ago, a company published a YouTube video which went viral and received 1.5 million views, more than any other video the company had published. Looking at that metric alone, it could be concluded that the campaign was a success, as they gained a lot of visibility for a charity cause. However, when looking at another metric – how many people eventually ended up donating (zero), that impression is quickly corrected. One must know which metrics are important and consider them before being able to tell whether they succeeded or failed. (Bladt & Filbin 2013.)

Sometimes it is not that obvious which metrics are more relevant for the specific goals at the time, which is why Chapter 3.1 will also study and define the metrics and statistics that could be relevant to Yksa in particular.

However, before moving on from metrics and statistics in general, it is important to take a quick look at the concept of data quality.

2.5 Data quality

According to the ISO/IEC 25012 standard, data quality is defined as “the degree to which data satisfies the requirements of its intended purpose” (Ziad 2021). For different businesses it may mean different things, as each business values some data quality dimensions more than others. Regardless of which data quality dimensions are preferred, one rule applies for all businesses – if the stored data doesn’t satisfy the requirements of its intended purpose, it is of poor quality and worthless. Collecting, storing and managing data requires time, technology, knowledge and infrastructure, which will all be wasted if the data integrity is compromised. (Ziad 2021.)

Table 2 lists ten data quality dimensions and their descriptions.

Table 2. Data quality dimensions (Ziad 2021)

Data quality dimensions		
Intrinsic	**Accuracy**	How well do data values depict reality/correctness?
	Lineage	How trustworthy is the originating source of data?
	Semantic	Are data values true to their meaning?
	Structure	Do data values exist in the correct pattern and/or format?
Contextual	**Completeness**	Is your data as comprehensive as you need it to be?
	Consistency	Do same records at disparate sources have same data values?
	Currency	Is your data acceptably up to date?
	Timeliness	How quickly was the requested data made available?
	Reasonableness	Do data values have the correct data type and size?
	Identifiability	Does every record represent a unique identity / isn't a duplicate?

Intrinsic data quality dimensions assess and evaluate the data values directly, whereas contextual data quality dimensions consider the context, as the name implies. For example, *lineage* (whether the data comes from a trustworthy source or not) does not take into consideration the other data values of an attribute, but *timeliness* depends on other aspects, too. For example, problems in retrieving data in a reasonable time may be an indicator of data being poorly formatted or organized. (Ziad 2021.)

Data quality as a topic would have a lot more to offer than what fits within the scope of this thesis, but there are a few data quality dimensions that will be important to consider while migrating the metrics and statistics collection of Yksa to the chosen monitoring system: accuracy, lineage, structure and timeliness. The rest of the dimensions will be trusted to fulfil the data quality criteria. Chapter 5.3 further explains the chosen data quality dimensions and analyses the success of the migration based on these dimensions after the setup of the monitoring system.

3 YKSA

Yksa is a digital archiving system providing the means for managing the needs of both digital and traditional archiving and data storage. Currently, Yksa is being

used by several archives, cities, museums, businesses and learning institutions of higher education, each having their own, customized instance of Yksa.

Figure 3 describes some functionalities and purposes of Yksa.

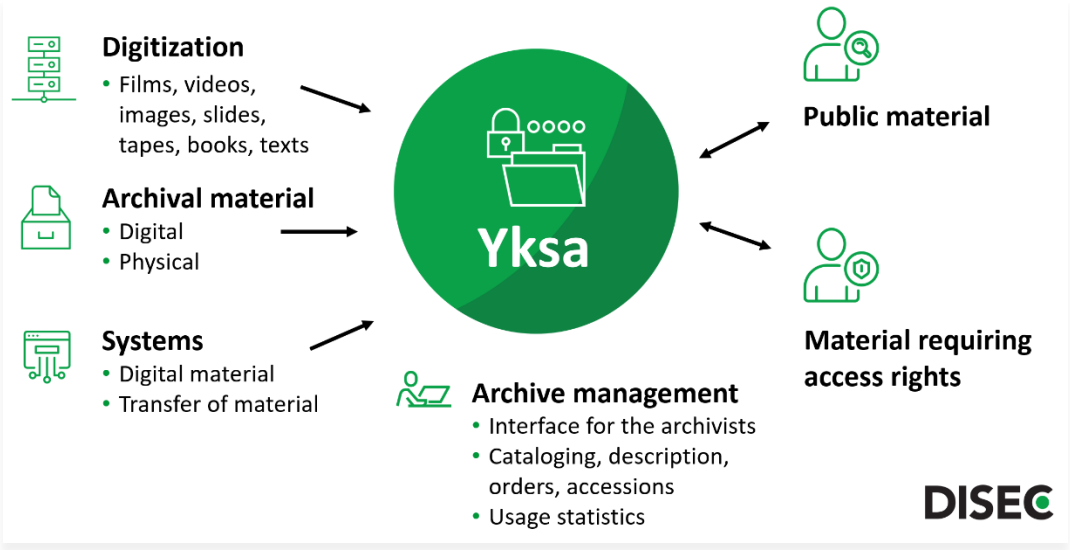

Figure 3. Description of Yksa (Disec Oy 2022)

Before comparing different monitoring systems, it is necessary to define which metrics and statistics are relevant to Yksa's functionalities. As the previous chapter showed, metrics alone don't tell enough – one also needs to understand which metrics are important for their current goals.

The monitoring methods currently existing in Yksa, and what possible advantages or disadvantages they have, will also be further examined before moving on to comparing and choosing the metrics and statistics monitoring system.

3.1 Metrics and statistics that are relevant to Yksa's functionalities

Right now, Yksa collects a rather limited number of statistics, such as how many file units an organization has in the system, how much disk space the files take or how many users the organization has configured. These are all numerical data – some of them discrete, some continuous (See Figure 2 and Chapter 2.3 for reference). However, some categorical data could also be beneficial to track, and

while some of it might require external surveys and other data collection methods, the possibilities of collecting such data might be worth consideration in the future.

As stated earlier, Yksa collects APM metrics. They focus on data, that is for example related to memory, data throughput, bandwidth or CPU utilization (Brush et al. 2022).

However, it would be beneficial to collect and process business metrics as well. As Yksa is a Software as a Service (SaaS), it could be seen to have the following important metrics, presented in Table 3.

Table 3. SaaS metrics (Balboni 2022; Lutkevich & Ehrens 2022)

SaaS metrics	
• Customer and loyalty retention	• Customer acquisition cost (CAC)
• Churn rate	• Session duration
• Activation rate	• Number of active users (NAU)
• Conversion rate	• Net promoter score (NPS)
• Average revenue per account (ARPA)	• Customer lifetime value (CLV or LTV)
• LTV-to-CAC ratio	• Monthly recurring revenue (MRR) / annual recurring revenue (ARR)

All metrics in Table 3 are important to track from a business perspective, but in order to stay within the scope of the thesis and an area that doesn't require company-level data outside Yksa's immediate reach, only few of these metrics will be suggested to be implemented in the near future to provide an example of how to include the business metrics collection in Yksa.

Session duration tracks the total amount of time a product or service was used by a user (Lutkevich & Ehrens 2022). It is something that would be possible to be monitored from inside Yksa.

Number of active users (NAU) shows, how many people use the app regularly. For this, one should define a threshold – how many sessions per week is

considered “power usage” – and monitor how many users use the service in quantities that exceeds the threshold. Within this pool of active users, it’s possible to focus on their actions and gain more insight about which parts of the service users prefer and use most frequently, and which features they ignore completely. (Balboni 2022.)

Since business metrics are ultimately built on top of statistics, in order to save the metric data for number of active users (NAU), we first need the statistics about the session durations and amounts per some pre-defined time interval for all users. For the expansion of metrics and statistics collection in Yksa, it would be a good start to save some statistics about the user behavior and actions, and the additional metrics can later be built using the existing statistics.

For example, saving the number of sessions per user, per day would be a valid statistic on its own. Alone it wouldn’t have much importance, but it would provide a valuable building block for some meaningful metrics in the future, for example when calculating NAU.

3.2 The current monitoring methods in Yksa

Right now, two different monitoring systems are being used in Yksa, as demonstrated in Figure 4.

Metrics are being monitored using Prometheus, which is one of the options that will be researched next, but statistics are being saved in a Couchbase database.

The organizations also have access to reports with statistical data, and that data is being queried from all the organization data in the database at the moment of the report being requested, which is less efficient than retrieving the data from a database if it had been stored there in the desired format beforehand.

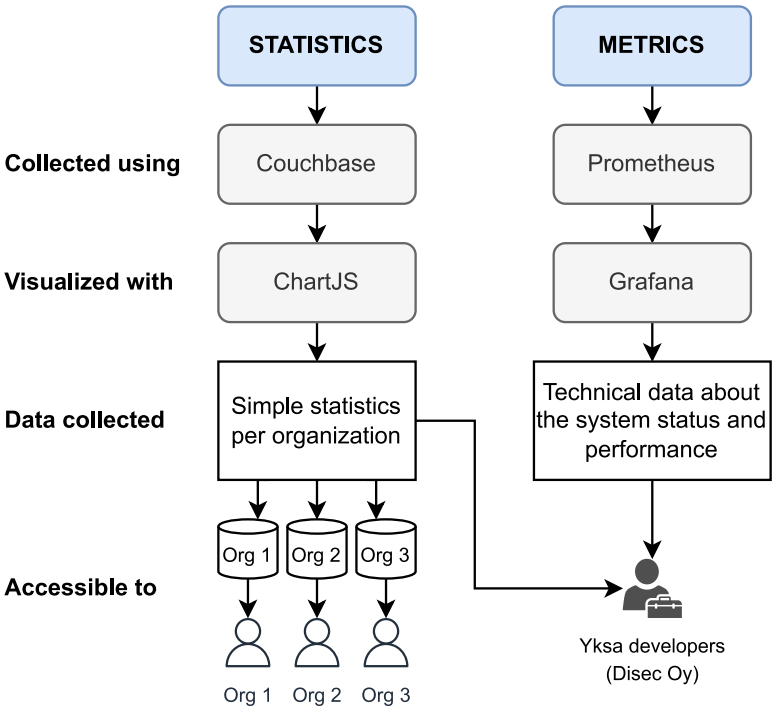

Figure 4. Current statistics and metrics monitoring methods in Yksa

Metrics are visible for Yksa's developers only, but as the statistics data is being collected for each organization, and is very limited and general by nature, the clients can view their own statistics inside their instance of Yksa and there has been no need to exclude some of the statistics from being shown to the clients.

In the future, in addition to the per-organization statistics and APM metrics, the suggestion is to also collect some general statistics (only visible to Yksa's developers) and business metrics. These two data types could be handled separately from the current statistics and metrics, but also be stored in the same location. Also, due to their similarities, it would be reasonable to bundle them together and treat them as the same data type, as neither will be accessible to customers, and both are relevant to the business perspective. The technicalities of this distinction will be further clarified in Chapter 5.4.

4 MONITORING SYSTEMS

Before starting to compare anything, one needs to know what matters – otherwise, the comparisons are irrelevant. In this chapter, the criteria for a monitoring system are defined, specifically with Yksa's best interests in mind.

Then, three monitoring systems will be compared, analyzing the findings while considering the criteria that was previously defined. Finally, one of the three systems will be chosen.

4.1 Criteria for a monitoring system

When choosing a system for monitoring metrics and statistics, it is advisable to first do some research and define the criteria the monitoring system should meet for the specific needs in question.

In general, a good monitoring system has the following features (Ellingwood 2022):

- Accepting and storing incoming and historical data
- Capability of managing data over periods of time, including older data
- Visualizing data by means of graphs and dashboards
- Pattern recognition
- Alerting

Other qualities to be considered could include:

- Computing resources used
- Processing speed

While Yksa's data collection and processing is still somewhat moderate, it is reasonable to choose a system that supports scalability in the future. Even if using Couchbase doesn't consume a significant number of resources or time right now, that might not be the case if the collection and processing of a myriad of statistics and metrics begins.

4.2 Comparison of some available systems

There are several noteworthy monitoring systems, three of which will be compared in this chapter: **Graphite**, **InfluxDB** and **Prometheus**. First, the technical information of the chosen systems will be analyzed using the comparison chart at DB-Engines.com (2022). Then, some additional online resources will be researched for further information and specifications.

While Prometheus has a slight front runner status, as it's already been deployed in Yksa, the other two will be given a fair chance during the comparison.

Table 4 illustrates the main features and differences of all three monitoring systems, according to DB-Engines.com (2022).

Table 4. Comparing three database management systems (DB-Engines.com 2022)

Comparing three database management systems			
	Graphite	**InfluxDB**	**Prometheus**
Primary database model	Time Series DBMS	Time Series DBMS	Time Series DBMS
DB-Engines ranking	Score: 6.03, Rank: #72 Overall, #4 Time Series DBMS	Score: 29.78, Rank: #29 Overall, #1 Time Series DBMS	Score: 6.62, Rank: #66 Overall, #3 Time Series DBMS
License	Open Source	Open Source	Open Source
Server operating systems	Linux Unix	Linux OS X (through Homebrew)	Linux Windows
Data scheme	yes	schema-free	yes
APIs and other access methods	HTTP API Sockets	HTTP API JSON over UDP	RESTful HTTP/JSON API
Supports Java	no	yes	Yes

For Table 4, most technical properties presented in DB-Engines.com (2022) were omitted due to not having any differing information that greatly affects the choice to be made.

While the DB-Engines ranking might appear to be a good comparator at first, the score and rank describe the performance in general, not specifically as a metrics or statistics monitoring system. Naturally, there might be some correlation, as it is possible that the most liked management system has the best performance in many or all aspects, but that is not guaranteed to be the case.

Before further comparisons, Graphite will be discarded from the consideration, as it has the weakest Java (the programming language) support among the three options. The metrics and statistics collection in Yksa is heavily built on Java, and

while some libraries for Graphite have been created, it seems that InfluxDB and Prometheus are more suitable for Yksa's needs in this matter.
InfluxDB is schema-free, but Prometheus uses a schema.

A schema can be described as a blueprint that defines how the data to be stored is formatted and mapped (Ogilvy 2020). Both Schema and Schemaless databases have their benefits and downsides, which are illustrated in Table 5.

Table 5. Advantages and disadvantages of Schema and Schemaless (Ogilvy 2020)

Schema vs Schemaless		
	Schema	**Schemaless**
Advantages	• The rules are clear for everyone when a structure is predefined • Less ambiguity, easier to migrate from one system to another • Possible data structure related issues will be resolved beforehand (which can also be a disadvantage) • Fewer bugs, easier to understand code	• No need for data modeling or upfront planning • No complex schema changes • Very fast iteration (quick for experimenting)
Disadvantages	• Data modeling and planning need to be done upfront • Changing the schema can be complicated • Slower iteration than schemaless, more difficult for experimenting	• Disorganized structure can cause problems especially with several people working on the data • Data type mismatches • Big performance cost for type conversion (garbage creation, extra CPU and RAM usage)

In the case of Yksa, the need for data modeling and upfront planning are not an issue. There is no need for fast iteration either, as the purpose is to carefully set up a robust system that can later be built upon.

Furthermore, setting up the data collection for statistic and metrics requires some planning either way, so having to consider the data storage aspect at the same time isn't very arduous.

More importantly, in this case, better performance and an organized structure are benefits that outweigh the general advantages of the schema-free data storage.

As these technical specifications don't necessarily tell the whole truth, more research was done to compare Prometheus and InfluxDB. Prometheus a pull-based system, but InfluxDB supports both pushing and pulling data (Elsmore 2020). In push-based systems, the application (such as Yksa) should actively push the data into the monitoring system, but pull-based systems fetch the data periodically from an endpoint the application has published.

InfluxDB is more suitable for event logging, Prometheus on the other hand for metrics recording – in fact, it was built with monitoring in mind (Elsmore 2020). While Prometheus sounds better for collecting metrics, collecting statistics has to be possible, too.

4.3 Choosing a monitoring system

Originally, based on some technical data in Chapter 4.2, Prometheus was chosen. For example, it's more suitable for metrics recording than InfluxDB, and due to Prometheus using a schema it was thought to perform better.

Upon closer investigation and experimenting, it turned out that it does not work well for the purpose of statistics collection in Yksa. The importance of support for pushing became clear when trying to migrate the statistics collection to Prometheus. In Prometheus, the data is pulled (scraped) at specific intervals, preferably very short ones. In this case, the statistics should be collected once a day, at a specific time. While Prometheus has an option called *Pushgateway*, which technically offers a solution for situations where pulling data isn't an option, its use cases differ greatly from what is needed for the statistics collection, meaning that it lacks many properties required for optimized pushing of the statistics. Furthermore, setting it up is a lot more complicated than simply using a system that properly supports pushing.

In InfluxDB pushing the data works the required way, and scraping is also supported, although right now the interval cannot be changed from the default

value, 10 seconds. Moreover, querying and parsing the data from InfluxDB proved to be a lot more straightforward. Therefore, InfluxDB was chosen.

5 IMPLEMENTING THE CHOSEN MONITORING SYSTEM

For possible demonstration purposes, a new instance of Yksa has been deployed for a demo museum. The data for it is artificially created and therefore does not compromise any real client's privacy.

5.1 Initial setup

First, InfluxDB was downloaded and installed following the instructions at Influxdata.com (n.d.b).

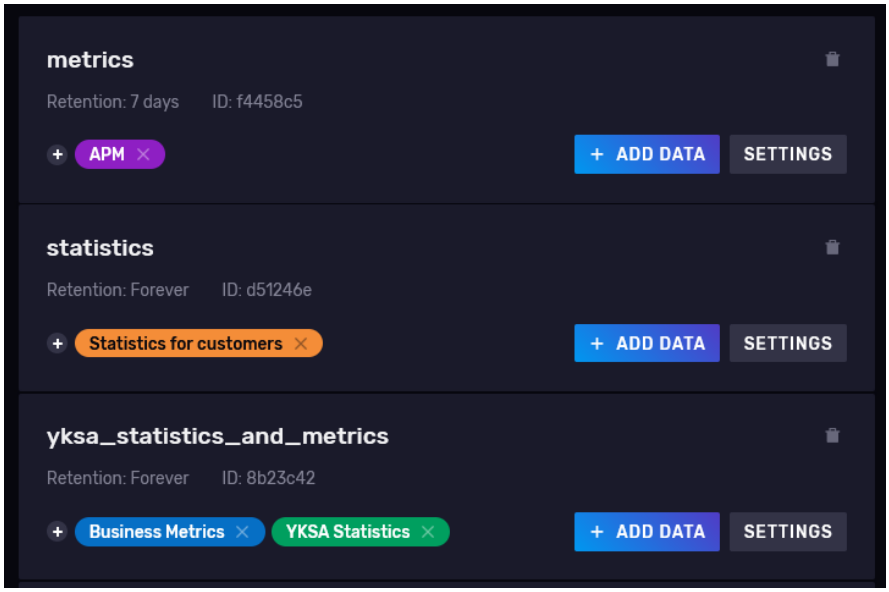

Figure 5. Suggested bucket division

A few buckets (containers for storing data) were created in the InfluxDB graphical user interface (GUI) – one for storing current metrics, one for storing the user-specific statistics and one for collecting general statistics and business metrics relevant to Yksa's functionalities (Figure 5). The suggested bucket names are simplified for demonstration purposes, and when InfluxDB is deployed on the company servers, it is advisable to rename them to adhere to the company security policy. The labels were also added purely for simplifying the distinction for the admins.

Separating these three different data groups in their respective buckets is not necessarily a required step, but makes the structure more comprehensible and makes the queries less prone to error as well. While the third bucket will not be used during the improvement of the current system, it was created to demonstrate how the expansion of data collection could be implemented.

It is up to the future implementer whether they want to keep the business metrics and general statistics together or not. It could also be justifiable to separate them, so that the general statistics bucket contains various data, whether it will end up having further use or not, and the business metrics bucket will have the carefully chosen data processed into meaningful numbers. As the expansion of the data collection remains a suggestion in this thesis, these two buckets and data types have been combined as one, for the sake of clarity.

```java
public class InfluxDBConfiguration {
    private @NotNull String url;
    private @NotNull char[] token;
    private @NotNull String org;

    @Override
    public String toString() { return ApplicationConfiguration.toString( o: this); }

    public String getOrg() { return org; }
    public void setOrg(final String org) { this.org = org; }
    public char[] getToken() { return token; }
    public void setToken(final char[] token) { this.token = token; }
    public String getUrl() { return url; }
    public void setUrl(final String url) { this.url = url; }
}
```

Figure 6. Class InfluxDBConfiguration.java for retrieving the InfluxDB credentials

Before creating new classes for the actual database related operations, the credentials needed for the database connection were saved in a separate configuration file and the format of the credentials was specified in a new class (Figure 6). The API token needed was also created in the InfluxDB GUI.
With the database connection set up, it was possible to move on and start the actual migration of the statistics and metrics.

5.2 Migrating the metrics and statistics data collection to InfluxDB

Figure 7 demonstrates the underlying structure for saving the statistics with a scheduled job, introducing the classes that will be referred to in the following steps.

Class *StatisticJob*, implementing class *Job*, receives a *Set* of *StatisticSuppliers* to forward the reactive stream of *Statistic* objects to the *StatisticsRepository.java*.

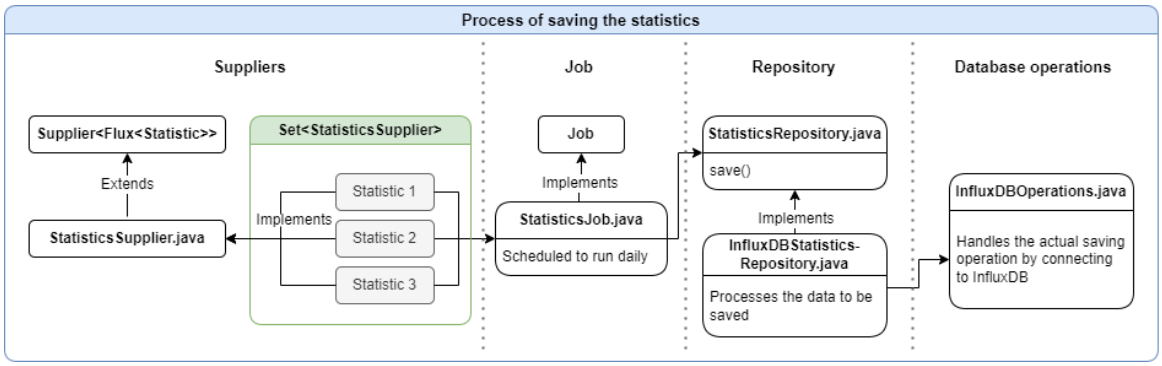

Figure 7. Process of saving the statistics

Earlier, the repository and operations classes for Couchbase were used, but for this migration, new classes were needed. A new class *InfluxDBStatisticsRepository.java* was created to handle the requests directed at InfluxDB. Similarly, a new class *InfluxDBOperations.java* was created to take care of the connection and requests to and from InfluxDB, such as querying or saving data.

5.2.1 Migrating the regular statistics

After creating the necessary classes, the next step was to recreate the current functionalities with InfluxDB.

The required steps were the following:

1. Create a Flux query to save the daily *Statistic* objects to InfluxDB (used by the Quartz Job, data itself comes from a set of Supplier classes and didn't need refactoring)
2. Create a script to migrate all existing *Statistic* objects from Couchbase to InfluxDB (using nested *for* loops and the *save()* method created above)

3. Create corresponding Flux queries for every N1QL query that is being used for retrieving *Statistic* objects or information related to them

First, the data to be saved was converted into a list of *Point* objects, which is a format that the *InfluxDBClient* library supports when saving data.

Figure 8 demonstrates how transforming the reactive stream of *Statistic* objects into a List of *Point* objects could work. The real *Point* objects created in Yksa have more tags, but they were omitted from the demonstration.

```
final List<Point> pointList = statistics.map(statistic ->
        Point.measurement(statistic.getType().name())
        .time(statistic.getTimestamp(), WritePrecision.S)
        .addField( field: "value", statistic.getValue())
        .addTag("key", statistic.getKey())
        .addTag("organization", statistic.getOrganization()))
    .collectList() Mono<List<Point>>
    .block();
```

Figure 8. Creating a list of Point objects

As all of the historical statistics data is currently saved in Couchbase, a migration script was created. It is meant to be executed once, and it will get all existing statistics and save them in InfluxDB.

For the query language, there were two options – InfluxQL and Flux. The syntax of the latter seemed very different from the more traditional querying languages, and due to its superiority over InfluxQL (Influxdata.com n.d.a), it was chosen for the queries used.

Figure 9 shows a rather simple Flux query that will fetch all distinct years from saved statistics of the requested organization. The result is used to populate a drop-down menu to let the user choose which year's statistics they wish to examine.

```
final String flux = """
    import "date"

    from(bucket: "%s")
            |> range(start: 0)
            |> filter(fn: (r) => r.organization == "%s")
            |> keep(columns: ["_time"])
            |> distinct(column: "_time")
            |> map(fn: (r) => ({r with _value: date.year(t: r._value)}))
            |> distinct(column: "_value")
    """;
```

Figure 9. A Flux query for retrieving distinct years for an organization

The query returns a list of *FluxTables,* that contain *FluxRecords* which can be mapped to a desired class using the *FluxResultMapper* class, as seen in Figure 10.

```
private <T> List<T> mapResultToType(final String flux, final Class<T> clazz) {
    final FluxResultMapper resultMapper = new FluxResultMapper();
    return influxDBOperations.findAll(flux) List<FluxTable>
        .stream() Stream<FluxTable>
        .flatMap(t -> t.getRecords() List<FluxRecord>
            .stream() Stream<FluxRecord>
            .map(r -> resultMapper.toPOJO(r, clazz))) Stream<T>
        .toList();
}
```

Figure 10. Private function to map the FluxTables into the desired class

While with the regular statistics the results need to be mapped to *Statistic* objects, in the case of the report statistics some variations exist.

5.2.2 Migrating the report statistics

Currently, the statistics are saved every night with a Quartz Job using a Supplier. In the future, the report statistics will be saved in the same way, so that building the reports will be more efficient.

The required steps were the following:

1. Create N1QL queries to retrieve the current information used by statistics reports from Couchbase
2. Create a script to retrieve all data used by the statistics reports and save it in InfluxDB
3. Create corresponding Flux queries for every N1QL or Solr query that is being used for retrieving report statistics or information related to them (considering the time intervals)

First, a new Java class implementing the *StatisticsSupplier* class was created. All of the N1QL queries were formatted to retrieve the current situation (instead of the one related to a specific time period), grouping the results by organization, so all the data can be saved as *Statistic* objects at once.
Some more fields were added to the *Statistic* class, so that *Statistic* objects can hold more information, needed by the report statistics.

While retrieving all the current *Statistic* objects from Couchbase was rather straightforward, getting all the data used by the statistics reports required some more work. As the reports can be built for any time range, the data also had to be saved for each day where any changes had happened. Sometimes the needed data doesn't have any unambiguous "created" date saved, which complicates the queries. This is one of the reasons why this information is preferred to be saved once a day in the future.

The resulting script was slightly heavyweight, but as it will be executed only once, there was no need to further optimize its speed or performance. Finally, for the different queries performed in the creation of the statistics report, the corresponding Flux queries were created to now fetch the same data from InfluxDB. The report statistics include data such as final counts of various types (e.g., archived images, acquisitions or documents) and the counts of new objects created during the given time period.

After the migration of the statistics, the InfluxDB database had the data points for each applicable day, so the calculations for the new objects were made based on the data entries at or before the given start and end date, instead of parsing through all existing documents for the same information. For the final counts, it

was necessary to query for the numbers for the end date, or the most recent data entry before that one.

Some of the report statistics were also included in the statistics charts visible to each organization, as they are very similar by nature. For this, some changes were made in the code to have their labels be translated correctly. Figure 11 demonstrates a graph populated with the data used in statistics reports, formerly unavailable on the statistics page. The graph shows the count of acquisitions and collection groups of the demo museum changing over time.

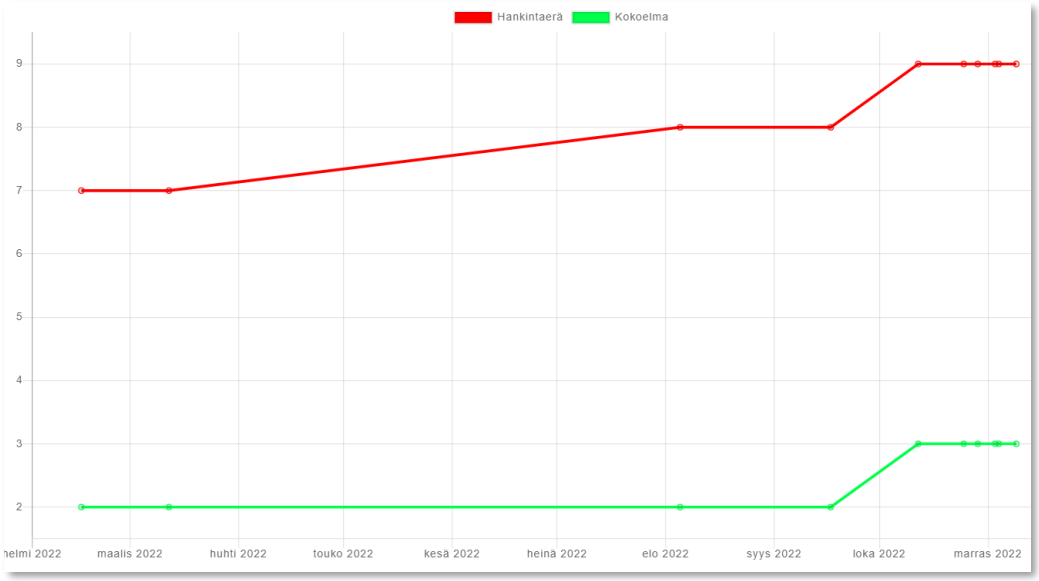

Figure 11. Example of an added graph

In addition to the queries causing some trouble, the timestamps also had to be carefully considered, which will be further described later.

5.2.3 Migrating metrics

Setting up a scraper for metrics monitoring was very simple, as it could implement the existing functionalities in Yksa. The scraper was given the same endpoint (the address where it pulls the data from) as the Prometheus scraper used, and no other changes were necessary. Yksa uses the Micrometer library, which processes and outputs APM metrics in a format that InfluxDB also supports.

Figure 12 shows the metrics scraper created in the InfluxDB GUI.

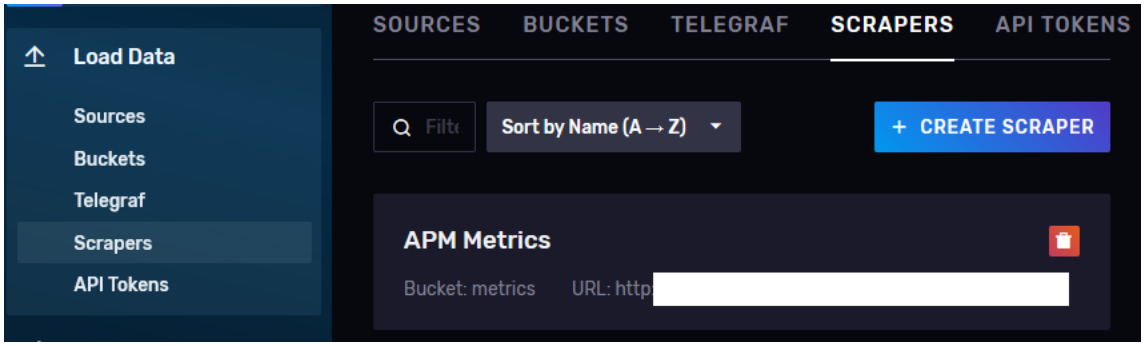

Figure 12. Creating a metrics scraper in the InfluxDB GUI

Next, following the instructions online (Grafana.com n.d), Grafana was set up, so that visualizing the migrated metrics collection could be verified to work as it used to. A new data source was added, configuring it with the InfluxDB information and credentials.

Next, a new dashboard was created, and in it a few replicated panels from the old system. The migration of the metrics visualization proved to be more time consuming than expected. The original panels used Prometheus as their data source, so the data was being queried with PromQL query language. For InfluxDB, both InfluxQL and Flux were an option, and the original queries would have to be converted to the chosen language.

Figure 13 shows the first few metrics visualized after having their PromQL queries converted to Flux.

For example, previously the process uptime data was requested with a rather simple query:

```
process_uptime_seconds{application=\"$application\", instance=\"$instance\"}
```

Now, to get the same information with Flux, it was replaced with the following query:

```
from(bucket: v.bucket)
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
```

```
|> filter(fn: (r) => r._measurement == "process_uptime_seconds")
|> last()
```

As some metrics use slightly more complicated logic, e.g., summing functions and multiple measurements combined in one table, some manual work is needed to visualize all current metrics again.

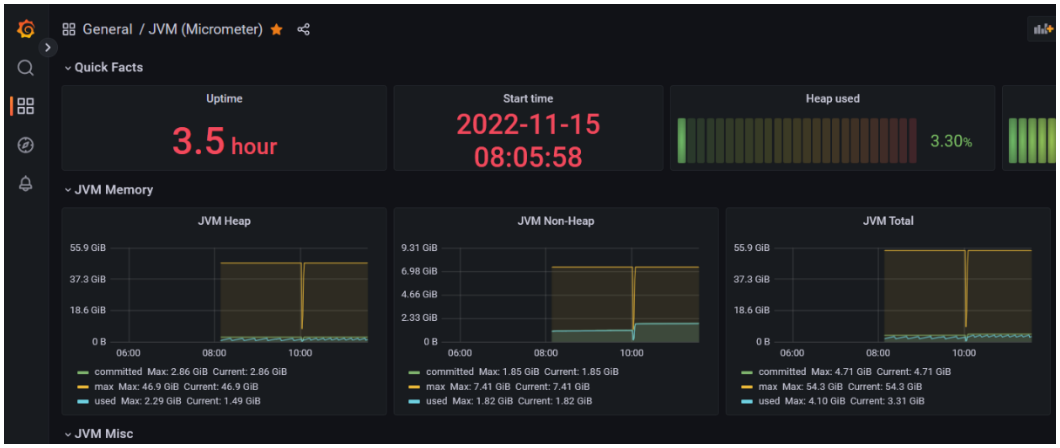

Figure 13. Creating a dashboard to visualize metrics from InfluxDB in Grafana

After finishing all query conversions, it is important to make sure the data corresponds to the original values. In the local environment, some numbers might differ greatly, which adds to the challenge of this verification. Before deploying the new system and discarding Prometheus, it will be necessary to do the final check with the real data on the Yksa server.

5.3 Current data quality dimension considerations in Yksa

As mentioned earlier, the data quality dimensions (Table 2, p.12), that were paid additional attention to were accuracy, lineage, structure, and timeliness.

Accuracy, or how well the data depicts reality or correctness (Ziad 2021) became a concern when querying for the report statistics. As the reports show accumulated data for a time period, it was crucial to make sure the dates are being handled correctly in the queries and while saving the historical data with the script.

At first, the script used truncated timestamps due to grouping, meaning that for example, when then examining the counts for day 05 November 2022, the

timestamp saved with the counts was in format 2022-11-05T00:00:00.000Z. This is counterintuitive, as the numbers reflect the situation at the end of the day, not at the beginning. Therefore, when querying for items created after day 5th of November, the query would have returned a result, according to which the count at the end of day was already the situation at the beginning of the day.

For unambiguity, the timestamp was changed to one minute before the midnight of the same day, which better indicates that the numbers in question are the situation after the day.

Lineage, or how trustworthy the source of the data is (Ziad 2021) was also relevant when processing the report statistics. At first, during the testing phase, there were some difficulties to make the data match between the report retrieving the data the old way and the report retrieving its data from InfluxDB. This was due to the reformatting of the N1QL queries in order to simplify them after the time period was no longer a concern in the query to be run daily. The query differences causing problems also applies to the metrics visualization, where the former PromQL queries had to be rewritten in Flux.

Structure, or whether the data exists in the correct pattern and format (Ziad 2021) caused no issues, as the statistics are tied to a Java class and its fields, which already enforces a clear structure. Naturally, it still had to be ensured that the data gets mapped to the fields correctly. The timestamp saved in the daily Quartz Job comes directly from the Instant class, which means it requires no further validation. Currently all statistics data is collected dynamically, so there is no room for user input errors.

Timeliness, or how quickly the data can be retrieved (Ziad 2021) is expected to be improved upon the migration of report statistics to InfluxDB, as it simplifies the queries needed for the report.

5.4 Suggestions for the future

For the expansion, the next step would be to create another model class, for example *YksaStatisticsAndMetrics.java* which contains the necessary fields for holding various information about Yksa. The metrics in question would be business metrics, and the statistics would be general in nature, and only accessible to the developers of Yksa.

Crucial fields for the new Java class would be *value*, *type* and *timestamp*, but additional fields would very likely be necessary when collecting more detailed data.

Then, another controller class would be needed. For this, the existing statistics controller can be used as an example. The new controller class would handle the data passing through the user interface and the repository.

A new Repository subclass would be required as well. When using a different model class and a different bucket, it helps keep the code simple and safer when these two different statistics would be handled separately.

However, the class for the database operations can be shared. Currently, the repository needs to specify the bucket when attempting to save, and queries have the bucket included in them, which allows a wider use of the database operations class.

For new statistics and metrics to be saved daily, they also need their own Supplier class, as the current one is being used by the statistics repository class that handles the per-organization statistics. The current *StatisticsJob* class could have another repository and set of suppliers injected and launch a separate save operation. A few additional Flux queries would be needed as well to retrieve the desired general statistics and business metrics.

Error! Reference source not found. sums up the suggestions for the metrics a nd statistics collection after developing the current methods. The suggested

metrics were explained in Chapter 3.1, while the suggested statistics are somewhat self-explanatory.

Suggested expansion: Statistics	Suggested expansion: Metrics

• Currently collected statistics
• Extra-organizational statistics only visible for Yksa's developers
  ○ # of users per language
  ○ # of all users
  ○ # of weekly users
  ○ # of organizations
	• Currently collected metrics (APM)
	• Session duration
	• Number of active users (NAU)
	• NAU / # of weekly users

Figure 14. Suggestions regarding Yksa's metrics and statistics to be collected

While the suggestions are also very limited by nature, they provide a rather simple start for the expansion of data collection in Yksa. After they have been set up along with their respective backend technologies, it will be easier to add more statistics and metrics to be collected in the future.

6 CONCLUSION

The main goal was to improve the current statistics and metrics collection methods by migrating away from Couchbase as the statistics storage system, and that goal was achieved by deploying InfluxDB to take over both statistics and metrics collection.

There were some problems along the process, first of them being the poorly interpreted research results when comparing the monitoring systems. Despite the research clearly stating that Prometheus is a pull-based system, the importance of this feature was not comprehended until an unsuccessful attempt to migrate the collection of statistics to Prometheus in the hope of the specious support for pushing being sufficient.

After switching to InfluxDB, migrating the regular statistics was very straightforward, but due to the problems arising with the report statistics, there was no time to implement some example business metrics to serve as the basis for future business metrics collection. Originally, the plan was to at the minimum

create the necessary Java classes and Flux queries for the new data to be collected, but those plans got truncated into suggestions and ideas existing in writing, only.

Before deploying InfluxDB on the company servers, more testing is needed with a bigger dataset consisting of real client data to ensure that the queries save and retrieve the data as they currently do with the existing methods.

Moreover, once the basics of the business metrics and general statistics collection have been implemented, it would be beneficial to consult someone with proper business insight. For example, additional research on KPIs and cost management might be worth consideration, as the service keeps growing. Even though business metrics were researched for this thesis, the findings that were documented barely scratch the surface of overall potential of business metrics, in order to stay within the scope of the thesis.

From the Yksa point of view, especially the SaaS metrics mentioned in Chapter 3.1 would deserve more attention. Observing a wider variety of business metrics is advisable, but they do not need to be tied to the monitoring system and methods of a software, such as Yksa. SaaS metrics, on the other hand, are directly related to Yksa and monitoring some of them is only possible by including data extracted from the operations of Yksa and its users.

For more user-related data, the developers of Yksa could consider creating some surveys they could ask the users to fill. The users are currently able to provide feedback in various ways, but such data cannot be quantified as easily as a questionnaire built for the very purpose of collecting user satisfaction data.

After creating a robust, scalable foundation, it is easy to continue building on top of the existing architecture and further improve the statistics and metrics collection methods, even far beyond the preliminary suggestions in this thesis.

REFERENCES

Balboni, K. 2022. 11 SaaS metrics you should be tracking. WWW document. Available at: https://www.appcues.com/blog/saas-growth-metrics [Accessed 30 August 2022]

Bendle, N. T., Farris, P. W., Pfeifer, P. E. & Reibstein, D. J. 2016. Marketing metrics. The manager's guide to measuring market performance. 3rd ed. New Jersey: Pearson Education, Inc. Available at: https://www.academia.edu/41617497/MARKETING_METRICS_THIRD_EDITION [Accessed 30 August 2022]

Bladt, J., Filbin, B. 2013. Know the Difference Between Your Data and Your Metrics. WWW document. Available at: https://hbr.org/2013/03/know-the-difference-between-yo [Accessed 19 August 2022].

Brush, K., Lockhart, E. & Demaitre, E. 2022. What is APM? Application performance monitoring guide. WWW document. Available at: https://www.techtarget.com/searchenterprisedesktop/definition/Application-monitoring-app-monitoring [Accessed 1 September 2022]

Campbell, S. 2010. Call Center Statistics and Metrics: What's the Difference? WWW document. Available at: https://technews.tmcnet.com/channels/call-center-reporting/articles/79957-call-center-statistics-metrics-whats-difference.htm. [Accessed 8 August 2022].

DB-Engines.com. System Properties Comparison Graphite vs. InfluxDB vs. Prometheus. WWW document. Available at: https://db-engines.com/en/system/Graphite%3BInfluxDB%3BPrometheus [Accessed 7 September 2022].

De Smith, M. J. 2021. Statistical Analysis Handbook. 2018-2021 ed. Winchelsea: The Winchelsea Press. Available at: https://www.statsref.com/StatsRefSample.pdf [Accessed 31 August 2022]

Disec Oy. 2022. Arkistonhallinta. WWW document. Available at: https://disec.fi/arkistopalvelut [Accessed 29 August 2022]

Donges, N. 2019. A Guide to Data Types in Statistics. WWW document. Available at: https://builtin.com/data-science/data-types-statistics [Accessed 31 August 2022]

Ellingwood, J. 2017. An Introduction to Metrics, Monitoring, and Alerting. WWW document. Available at: https://www.digitalocean.com/community/tutorials/an-introduction-to-metrics-monitoring-and-alerting [Accessed 1 September 2022]

Elsmore, M. 2020. Prometheus vs. InfluxDB: A Monitoring Comparison. WWW document. Available at: https://logz.io/blog/prometheus-influxdb [Accessed 8 September 2022]

Gallagher, B. 2019. Metrics Versus Statistics: What's the difference? WWW-document. Available at: https://www.megbusiness.com/metrics-versus-statistics-whats-the-difference [Accessed 8 August 2022].

Grafana.com. n.d. Get started. WWW-document. Available at: https://grafana.com/docs/grafana/latest/getting-started [Accessed 14 November 2022]

Indeed.com. 2022. 4 Examples of Key Performance Metrics To Track. WWW document. Available at: https://www.indeed.com/career-advice/career-development/key-performance-metrics [Accessed 29 August 2022]

Influxdata.com. n.d.a Flux vs InfluxQL. WWW-document. Available at: https://docs.influxdata.com/influxdb/v1.8/flux/flux-vs-influxql [Accessed 15 November 2022]

Influxdata.com. n.d.b. Get started with InfluxDB OSS 2.4. WWW document. Available at: https://docs.influxdata.com/influxdb/v2.4 [Accessed 25 September 2022]

Lutkevich, B., Ehrens, T. 2022. What is a business metric? WWW document. Available at: https://www.techtarget.com/searchcustomerexperience/definition/business-metric [Accessed 30 August 2022]

Ogilvy, H. 2020. What Are the Tradeoffs Between Schema And Schemaless? WWW document. Available at: https://www.search.io/blog/schema-vs-schemaless [Accessed 7 September 2022]

Ziad, Z. 2021. Data quality dimensions – 10 metrics you should be measuring. WWW document. Available at: https://dataladder.com/10-data-quality-metrics-you-should-measure [Accessed 30 August 2022]