MASTER THESIS

CREATING A GENERAL PROCESS TO MEASURE API PERFORMANCE

GENERATING A KPI-DRIVEN DASHBOARD AT AIR FRANCE KLM

AUTHOR

DESMOND VAN DEN BROEK

28-02-2023

UNIVERSITY OF TWENTE.

COLOPHON

MANAGEMENT
Department BMS
Industrial Engineering and Management

DATE
28-02-2023

PROJECT
Master thesis

AUTHOR(S)
Desmond van den Broek

TELEPHONE
+31651821087

EMAIL
d.vandenbroek@student.utwente.nl

POSTAL ADDRESS
P.O. Box 217
7500 AE Enschede

WEBSITE
www.utwente.nl

EXECUTIVE SUMMARY

This paper shows the research that is conducted to generate a generic process that can be used to create an API performance dashboard, where the procedure is being validated by carrying the process out in practise, namely at Air France-KLM(AFKL). The generalizability of this research ensures the scientific contribution. This process can be applied in different types of companies or instances and the process does not need to be altered. This process has value to any company that needs more insight on their API performance. At this moment, various teams within AFKL do measure numerous specific aspects on API performance, but all the information is scattered and the teams do not have the knowledge of the work that is carried out by other teams.

The general process that is created consists of eight steps. The first step is to make a clarification on the current situation at the firm the research is conducted. During this step, the stakeholders were also identified. Secondly, the objectives of the tool have been listed. The third step consists of finding the KPIs that are used on the dashboard. Within this phase, it is crucial to elaborate with the corperation what KPIs there were already thought of by the employees or stakeholders. Afterwards, the literature was checked to find KPIs that are proven to be useful in practise in preferably companies that are similar to the company that the research is conducted. When all KPIs were found, the most valuable KPIs were chosen. When the KPIs were listed, the requirements of the tool were conducted. The most important stakeholders need to give their input on what they find crucial on the dashboard. When the requirements were agreed upon, all necessary data had been retrieved. To ensure that all data that is necessary has been retrieved, the following steps need to be walked through. Firstly, it needs to be clear how the KPIs that are going to be visualized have to be calculated. If the company where the research is being conducted has multiple datasources, it needs to be evident what data has to be retrieved from which data source. When all the data sources are found, the necessary connections to the tool where the performance will be visualized on can be made. When this phase is processed, all the necessities for the visualization are generated and the tool can be produced. When the tool is created, an evaluation needs to be carried out to fix possible flaws or improvement options in the product. For the evaluation, the UTAUT model from Vankatesh can be used, where the UTAUT model focuses on four aspects, namely the performance expectancy, effort expectancy, social influence and facilitating conditions. From these four aspects, improvement options can easily be detected and a plan to improve the tool can be created. The UTAUT model needs to be conducted with the most important stakeholders. In the last step of the process, the tool is deployed within the company.

The validation of this general process is carried out within AFKL. The eventual tool is created within two programs. The data that is focused on the performance of the APIs is visualized within grafana, while the data that is focused on usage of the APIs is shown in a spotfire dashboard. The decision to have two dashboards instead of one, is done because not all data can be retrieved within AFKL at this moment. At this stage, there are five KPIs visualized. This means that at the end of this reaearch, AFKL acquired a dashboard with visualization of the five most important KPIs based on both their own expertise and the literature. With this dashboard, AFKL gains more knowledge on the API performance and this dashboard can be used to create more awareness throughout the different teams.

At the end of this paper the limitations of this research are stated, with the opportunities for further research and the recommendations for AFKL on how to implement the tool and make it as valuable as possible to AFKL. Some of the opportunities for AFKL include the improvement of the data connections and the addition of a few more KPIs to satisfy more needs of the stakeholders.

PREFACE

This thesis marks the end of my master's Industrial Engineering and Management. This report is the result of a long time of hard work within Air France-KLM and as a student at the University of Twente. I was very fortunate to get a great opportunity to experience a massive worldwide company, such as Air France-KLM. I want to thank Vahid Babaloo and Kevin Duijndam for this chance, it has been a pleasure. Moreover, I want to thank my close colleagues within Air France-KLM, who have supported me throughout my research, both personally and academically. The various discussions that were held and the feedback that was given was immensely appreciated and gave me guidance when necessary.

Furthermore, I want to than dr. ir. M.E. Iacob and second supervisor J.P.S. Piest with their clear and thorough feedback. The feedback not only helped me develop my research, but also my personal skills, which is very valuable.

Lastly, I want to dedicate this thesis to my dad. He was always supporting me throughout my studies and urged me to always work hard and do the best I can. Even on his deathbed, he tried to prolong his life to watch me graduate. Unfortunately, he is not here anymore, but I can proudly say I eventually fulfilled his biggest wish, to finish my master's.

Desmond van den Broek

Apeldoorn, February 2023

TABLE OF CONTENT

LIST OF FIGURES, TABLES AND EQUATIONS

1. INTRODUCTION

In this introductory chapter, an introduction is given on the company where this research will be conducted is given in section 1.1 and more precisely, on what teams it concerns within the company in section 1.2. The research motivation (section 1.3), action plan (section 1.4), research goal and questions (section 1.5) and research methodology (section 1.6) is elaborated on afterwards.

1.1 KLM

KLM stands for the Royal Dutch Airlines and was founded on the 7th of October 1919 and this makes KLM the oldest airline that still operates under its original name. KLM has an enormous reach throughout the world with 92 European and 70 intercontinental destinations. Since 2004, KLM has been merged with Air France, creating the Air France-KLM (AFKL) group where they work together on three key activities, passenger business, cargo and engineering and management from two main hubs in Amsterdam (Schiphol) and Paris (Charles de Gaulle) (KLM, 2022).

1.2 OPEN API COMMUNITY

The open Application Programming Interfaces (API) community includes all the teams that are working on APIs. This consists of all employees that work on APIs, which are the analysts, developers and designers, and its consumers. The open API community started in 2013 and has been expanding rapidly ever since. The growth of the community is driven by the increasing number of APIs and the willingness to improve their API performance. The number of APIs is currently at 50 and is still increasing. The vision of the open API community is that they want Air France-KLM to be successful in opening and profiting from new partnerships through APIs, enabling partnership managers, who is a manager of a partner whom concerns himself with the API performance within its own department, to act and adopt swiftly when opportunities arise and partners to connect easily.

1.3 RESEARCH MOTIVATION

Within the open API community at KLM there are four areas of focus. In all of these four areas KLM has the priority to improve. The first one is the API as a product. In this area, KLM wants to improve by making the APIs standard and marketable products. The second focus area is the API provider experience. KLM wants to improve the provider experience by improving the API delivery. Thirdly, the API consumer experience is a key focus point. KLM is improving the API consumer experience in various manners. One of them is by improving the onboarding process. To improve the onboarding process, KLM has identified many actions found by an extensive analysis. The last key aspect where the open API community has its main focus on is the API awareness. KLM wants to raise the awareness both internally and externally.

The open API community wants to create API awareness by creating a dashboard where multiple valuable KPIs are shown and continuously updated by the gathered data. The dashboard is necessary to both give KLM employees extra input on the value of the APIs within KLM as well as it gives external value to KLM by increasing the possibilities to provide API service to external clients. Indirectly, the creation of the dashboard also gives opportunities to give insights on the other three focus areas. Therefore, we see that creating a dashboard with valuable KPIs, can create enormous worth in all four focus areas of the open API community.

The generic process to visualize API performance can play an important role in addressing the issue of the lack of insight of the performance of APIs within companies. The generic process can give companies the tools to measure what is unknown and improve the problems that can be addressed. This generic process is suitable for most types of companies and can therefore be applied in several companies that need more insight on the API performance.

1.4 ACTION PLAN

The eventual goal of this research is to create a dashboard where the key success factors of the API performance within AFKL. To create this dashboard, numerous steps have to be taken. Firstly, a systematic literature research is carried out, simultaneously with an internal investigation on the already stated KPIs. The literature research is done to find out what other KPIs are commonly used in the industry and on API performance, while we want to reduce the stated number of KPIs to five to seven KPIs (Pillai, 2020). We need to find which KPIs are most valuable and which are more or less redundant, while we find out in the literature research, what KPIs are commonly used in the industry and on API performance.

The finalization of the KPIs is based on information acquired from the literature as well as expertise from the experts within KLM. Furthermore, an analysis of which stakeholders are currently influenced by API usage and their importance and value is crucial to eventually find the essential KPIs.

When the KPIs are stated, all necessary data that we need has to be retrieved. To do this, we need to know what data is already available, what data we need but do not have available yet and how we can obtain the missing data.

After we have our KPIs and the necessary data, the final dashboard is created. When the dashboard is generated, various key facets need to be exhibited. It has to be known who gets access to the dashboard/who will use the dashboard, what filters we need to add and more. The dashboard is created in Spotfire.

As soon as the dashboard is created, the evaluation is done. During the evaluation, various stakeholders are asked on the performance of the dashboard. This will include various facets, such as the usefulness, accessibility and the clearness of the dashboard.

1.5 RESEARCH GOALS AND QUESTIONS

Within AFKL there are three main goals regarding API performance. First of all, AFKL desires to visualize the value of its APIs. Next to that, AFKL wishes to accurately measure performances of the different APIs. Thirdly, wants to generate awareness within different departments of AFKL. To solve these problems, a dashboard with useful KPIs is essential. The goal of this research is to create a dashboard with the most valuable KPIs on API management for AFKL. To create the most useful dashboard, the focus will be on the following main research question, "***What is an effective KPI data driven approach to measure the performance of API management?***". The main research question is answered in chapter 3.

To start off, a research with stakeholders, combined with a literature review is performed to create a list of the most useful KPIs on API management. Within the literature, the focus will be on KPIs that have been used to measure API management in other companies, which are preferably similar to AFKL. This is done to create knowledge on how other companies and researchers have measured the performance on API management and how this gives insight on valuable indicators that have been proven to be effective. On the other hand, the ideas of the most useful indicators of the stakeholders are acknowledged to find out what the most important aspects are to AFKL. A combination of the KPIs from the literature and from the stakeholders are combined to find the most relevant KPIs to show on the dashboard. This leads to the following sub question, "***Which KPIs capture the performance of the API usage well?***". This sub question is answered in chapter 2.

When the KPIs are known, all necessary data needs to be retrieved to measure the KPIs. Because of this, we use the following research question, "***How can we retrieve all necessary data?***". This research question is elaborated on in chapter 4.

When all the necessary data is retrieved, the dashboard needs to be created. The different KPIs need to be shown in a clear way and in a way that the results are easy to see for the different users. To create this dashboard, the following sub question is asked, "***How can we design a dashboard for the listed KPIs?"***. This sub question is answered in chapter 5.

Furthermore, it is crucial to validate and evaluate the dashboard. This part of our process is essential to find out whether the dashboard is meeting up to the expectations of the stakeholders. That being the case, the second research question holds, "***Does the dashboard meet the expectation of the most crucial stakeholders?***". This research question is discussed in chapter 6.

1.6 RESEARCH METHODOLOGY

During this research, the guidelines of the Design Science Research Process (DSRP) by Peffers (2006) are used. For this particular situation, a problem centered approach is used. The DSRP proposes six steps, namely problem identification and motivation, objectives of a solution, design and development, demonstration, evaluation and communication. The DSRP is created by Peffers to incorporate the design science process into the field of information science research. Below, a figure is shown how Peffers' process is adjusted to adequately suit the creation of the generic process.

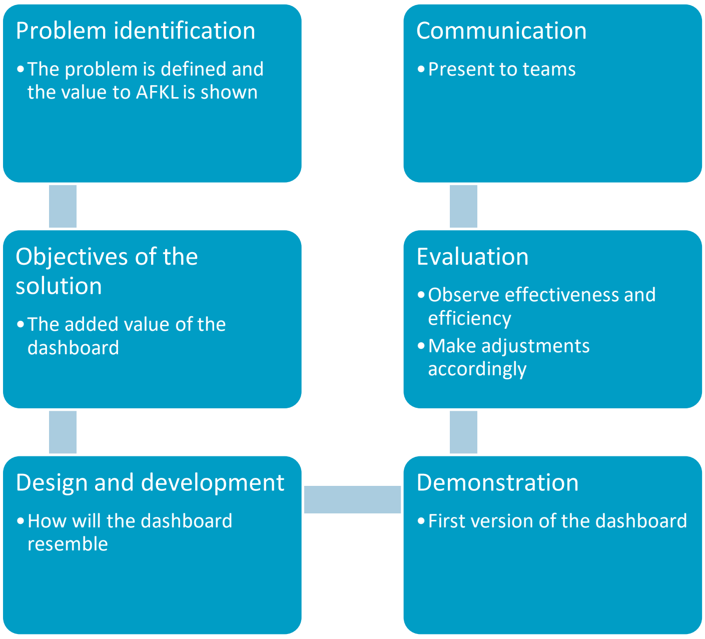

Problem identification
• The problem is defined and the value to AFKL is shown
Objectives of the solution
• The added value of the dashboard
Design and development
• How will the dashboard resemble
Demonstration
• First version of the dashboard
Evaluation
• Observe effectiveness and efficiency
• Make adjustments accordingly
Communication
• Present to teams

Figure 1: The Design Science Research Process. Figure adapted from Peffers (2006)

Problem identification and motivation

In this part of the research, where the research problem and the value of finding its solution is shown. The justification of the value of the solution has two major advantages, it motivates the researcher and the audience to pursue the solution and accept its results and it helps to understand the reasoning behind the researcher's understanding of the problem. The problem identification and its motivation is already stated above in section 1.3. At this moment, the API awareness throughout the company is too low, as low as that there are teams that are unaware of their existence. In chapter five this part will be further elaborated on, in the "current situation phase".

Objectives of a solution

During the objectives of a solution phase, a choice is made to a quantitative solution, a solution where the found solution outperforms the current solution, or a qualitative solution, where a new solution is found. In chapter two a qualitative solution will be found through a literature research, where KPIs are found in the literature and aligned with AFKL and their desires. In chapter three, this step is referred to as the "objectives" phase.

Design and development

In the design and development phase, an artificial solution is made and includes determining the desired functionality and its architecture. In this research, this includes the first version of the dashboard and this is shown in chapter five. This phase is also elaborated on in chapter three in the "dashboard" phase.

Demonstration

After the first design is created, a demonstration needs to be done to see the efficacy of the dashboard. This will involve experimentations and testing of internal stakeholders. There is further elaboration on the demonstration in chapter five.

Evaluation

During the evaluation phase, feedback from the stakeholders and other experts is consumed. It is essential to gain improvement options as well as positive feedback on features that are appreciated. This is crucial to gain a dashboard that provides valuable information, as well as a product that users are content to use. When useful data regarding its performance and design are obtained, a loop back to the design and development phase is made, to implement the required changes. In chapter three, this step is also considered, where it is also called the "evaluation" phase. The evaluation process is further elaborated on in chapter 6.

Communication

In the communication stage, the dashboard is finalized and communicated to all required recipients. At AFKL, the dashboard will become visible to all parties that need access and a page will be created where external parties that need insight, will have a location where they can use the dashboard. The communication step is referred to in chapter three as part of the "deployment" step.

1.7 STRUCTURE REPORT

In chapter two, the theoretical background is provided, combined with the KPIs that have been stated by AFKL and the elaboration which KPIs will be visualized on the dashboard. In chapter three, the generic process is explained step by step. In the fourth chapter the different ways of retrieving data are discussed. In chapter five, the later steps of the generic process are carried out and explained. In chapter six, the evaluation of the product is carried out. In chapter seven, the conclusion is given, combined with the limitations, recommendations and options for further research.

2. THEORETICAL BACKGROUND

In this chapter, the theoretical background is stated. This chapter answers the following research question, ***"Which KPIs capture the performance of the API usage well?"***. The chapter starts with a literature review where a theoretical background is given on APIs, KPIs and the combination of these two components. Furthermore, commonly used KPIs are listed with an elaboration on them, as well as KPIs listed by AFKL that the experts thought could be useful on measuring their performance. Lastly, a procedure is conducted on choosing the most useful KPIs for AFKL. This chapter is conducted to concretize the solution and its objectives.

2.1 STAKEHOLDERS

The stakeholders of the eventual dashboard are the members of the open API community. From these, the product owner of the developer portal, the product owner of service oriented architecture repository (SOAR or SOARepo), the product owner of API management teams (AMT) and the product manager of service centered integration (SCI) are the most important stakeholders.

Product owners

The purpose of product owners is optimizing the value of the product by setting a product vision and managing the development of the product. This is done by prioritizing, evaluating and steering new developments according to the needs of the customer and in coherence with the vision of Product Management (tasked with management of the product program) to contribute to the overall product improvement for the customer.

Product managers

The product manager is optimizing the value of a set of products (program) managed by an agile release plane by setting a vision for this program and managing the development of the products that are part of it. This is done by prioritizing, evaluating and steering new developments according to the needs of the customer and in coherence with the vision of Portfolio Management (tasked with management of the full product portfolio) to contribute to the overall product improvement for the customer.

Goals for stakeholders

1. To have the latest and the greatest documentation on the tools that we offer (SOA Repo, Developer Portal)
    a. To have a great dashboard means that we can keep track of which product team recently has updated their API compared to their documentation, this allows us to see how we can improve our workflow to accommodate the reasons of not also updating the information
    b. This should also decrease the amount of support required
2. To pinpoint what we can improve in terms of onboarding new partners
    a. Having a great dashboard also means that we can keep track of at what stage of the process, it gets stagnated, not able to move forward, and perhaps improve
    b. The amount of after support required from all members involved. Being able to keep track of the time taken means we can convert it into resources used, especially due to multiple teams being involved
3. To improve the automated process for providing teams
    a. Providing teams, especially new teams do not have the knowhow on how to deploy the services onto our environment. Having a dashboard means we can keep track of how long a team take from us being notified on an incoming new service to getting it to live to getting it to be consumed

b. Keep track if the technology used is on the latest version, as product teams owns their own product it is important that we collaborate together, as there is always a latest version, being able to keep track of which product team is using which version is really important

2.2 LITERATURE REVIEW

This main research goal of the systematic literature review is to answer the following research question, ***“Which KPIs capture the performance of the API usage well?”***. A systematic literature, based on the structure of Kitchenham (2007), review is carried out to gain more knowledge on indicators that are commonly used in businesses with API usage and proven to be successful in practise. Kitchenham explains the systematic literature review or systematic review as “*A form of secondary study that uses a well-defined methodology to identify, analyse and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable*”. One of the three main reasons to do a systematic literature review is according to Kitchenham is to provide a framework/background in order to appropriately position new research activity, which is what we are doing in this research. The indicators that are considered for the dashboard are given a value score between one and ten for the Weighted Shortest Job First (WSJF), which is elaborated on more in section 2.4.2.

The literature review below, the literature review process is visualized. The systematic literature review process is divided into six steps. To start off, the appropriate databases are selected. The data sources need to be complete and useful for the specific case study that needs to be executed. Secondly, a query is created and is ran through the different data bases to obtain the first list of articles. The query has to include the right keywords that will present the best results on the chosen topic. In the third step, a list of exclusion is used to reduce the number of articles that are found using the query. The exclusion criteria have the purpose of excluding as much data as possible that is selected by the query but is not useful for the specific research goal. Next, the abstract is used to remove more articles that do not seem to add value to the study. If the abstract seems to be about a completely different subject for instance, the article can be considered as non-valuable for this case. As a last filter mechanism, the complete articles are read to find out whether they are useful for the study. If the articles seem to be useful, they will be included in the research, however, if an article after reading is considered to be unusable for this research, the article is removed from the article list. When the previously mentioned steps are carried out carefully, a list of articles that are useful to answer the research question will be the outcome.

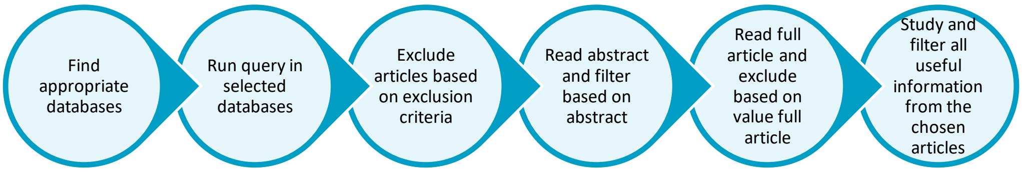

Find appropriate databases
Run query in selected databases
Exclude articles based on exclusion criteria
Read abstract and filter based on abstract
Read full article and exclude based on value full article
Study and filter all useful information from the chosen articles

Figure 2: Literature review process

For this review, three databases are used:

- Scopus
- Google Scholar
- ScienceDirect

These databases are chosen because of numerous reasons. Firstly, those databases are very complete and there papers and journals about a lot of topics, from various time periods. This helps to look into a research problem from different perspectives. For this research specifically, this means that we can get insights from different type of companies that use APIs and also companies that use similar software services to APIs. The different perspectives will lead to a more complete background, before creating the eventual product for AFKL.
Secondly, the availability of the documents is high in these databases. Within the previously named databases, a lot of the articles are accessible without having to go through a request process or a purchasing process. This ensures that the process of getting the required article is quicker than when one has to go through one of these processes for each article.
Furthermore, the three chosen databases are easy to use for such a systematic literature review. The search field is easy to use and filters are easily applied. This also speeds up the reviewing process.

The literature review protocol is projected in the table below

Academic databases	Scopus, Google Scholar, ScienceDirect
Search fields	Title, abstract, keywords
Language	English
Publication years	2011- present

Table 1: Literature review protocol

Inclusion criteria

- Includes at least one search term of the query
  This criterion speaks for itself, if there is no aspect from the query, or in other words, if there is no aspect included from what this research needs to achieve, the papers are not valuable for this research to include.
- Needs to be written in English
  This criterion is also fairly straightforward. The articles have to be understandable to be useful. Articles in English are the only articles that are desired.
- Needs to be published after 2010
  Because of the fact that this research is performed focusing on APIs, it is convenient to use recent data. APIs are relatively new, web APIs are only used for the first time in the year 2000(Bermbach, 2016). Using data published after 2011 will therefore ensure the significance to the current situation of AFKL.

Exclusion criteria

- Not related to the research
  If the paper is not related to the research, it does not add value to the research and is consequently excluded from the article list.
- Does not seem reliable
  If the article does not seem reliable, it is a risk to include it in the research. This could mean information that is invalid is being used, which would harm the validity of this research.
- Duplicates
  When using different data sources, duplicates may occur. Obviously, it is not valuable to use one article multiple times.
- Literature in any other language than English
  Articles in other languages may have a lot of value in them, but if the article is not in English, it does not have importance to this research.

- Published before 2011
  Articles published before 2011 may be outdated and there for have less significance. If the data is outdated, it could also harm the validity for this research. Because of this, articles before 2011 are outdated.

2.2.1 Literature review on API management

To obtain the most valuable articles, the following query has been used to search through the different databases. Note that the query is specified based on the inclusion and exclusion criteria mentioned in section above.

(“Key Success Factors” OR “KSF” OR “Critical Success Factors” OR “CSF” or “Key Performance Indicators” OR “KPI”) AND “Application Programming Interfaces” AND PUBYEAR > 2011 AND LANGUAGE ("ENGLISH")

There are different ways to formulate KPIs. Therefore different options are included within the query, separated with an “OR” operator. The “OR” operator makes sure that at least one of the search criteria is met. The first part regarding KPIs is connected to “Application Programming Interfaces” to make sure it is connected to the research problem. API as an abbreviation is not included, since this causes articles with words including -api- (e.g. capital) in their title, abstract or keywords to be added to the search results, while they are not interesting for our research. The titles and short summaries of the articles that are reviewed can be found in appendix A.

2.2.2 Literature review on service management

To gain more advanced knowledge on the usage of KPIs within services that are relatable to APIs, a different query is used to gain extra insights. Service management can be used as a replacement for API management.

Query
To obtain the most valuable articles, the following queries have been used to search through the different databases. For this search, only Scopus and Science direct are used.

(“Key Success Factors” OR “KSF” OR “Critical Success Factors” OR “CSF” or “Key Performance Indicators” OR “KPI”) AND “Service management” AND PUBYEAR > 2011 AND LANGUAGE ("ENGLISH")

The titles and short summaries can be found in appendix B.

These queries are created to gain knowledge on known KPIs in practise. The research is focused on finding KPIs that have been proven in similar circumstances as the situation of AFKL. If we loop this back to the research question as mentioned in chapter 1, ***Which KPIs capture the performance of the API usage well?***, this chapter focuses on all KPIs that have been proven to be successful in similar situation as for AFKL and we can therefore use from the literature on the dashboard of AFKL. Below the results of the literature review can be found

2.2.3 Application programming interfaces

An API is code that allows two software programs to communicate with each other. An API describes the correct way of how a developer requests services from an operating system. What APIs do is that they grant access to information requested by both users and other applications. The API also notes who has accessed what data at what time span (Kopestenski, 2019). APIs are made of two different elements (Medjaoui, 2019). The first element describes how information is communicated between two programs, which is done by a request from one party and then the request of processing from the second party. The second component is the software interface that is created.

Representational state transfer (REST) and simple object access tool protocol (SOAP) are commonly used architectures for APIs. Within AFKL, REST is used. The pros for REST is that it makes APIs easier to create and recreate, makes the implementation go smoother and the integration with websites and services is effortless.

AFKL makes use of three types of APIs based on the necessary security level. Private APIs are only used internally, open APIs are easily accessible and can be used by anyone who wants to use this API and partner APIs can only be used by partners of AFKL.

API maturity

APIs have their own lifecycles. The different stages of the maturity can indicate how much time and money still needs to be invested in the API. In the early stages of the development of the API, the main focus is on design, building aspects and reduce efforts on documentation for example. At the other stages more time is spent on monitoring the use of the API and securing it against misuse.

Within AFKL there are four stages of API maturity. The first stage is the explorer stage. In the explorer stage, the API needs to be developed and made ready for integration. In this stage, the main focus is on the administrative tasks and the following four questions,

- Is there a birth certificate?
- Is the API registered in SOARepo?
- Is there a product owner for the API?
- Did the API deliver business/functional content for the open innovation portal?

The second maturity level within AFKL is the silver level. To comply as a silver stage API, the following two criteria have to be met. Firstly, the API has to use the so called “Open API Specification” (OAS) format. This means that API has to follow the stated API design guidelines and the principles and rules stated by AFKL. The second criteria is that there cannot be any violations of the design guidelines. The next stage is the gold stage. For the gold stage, different criteria have to be met.

- API needs to be compliant with the AFKL API design guidelines
- More coverage of the previously mentioned OAS spec.
- Compliant to the relevant data models
- API design is reviewed
- API published on the DevPortal

Lastly, the platinum stage is obtained. When the API is in the platinum stage, it is in its optimal form and the main goal is to stay consistent and effective. The API needs to continue working to its causes and therefore stay valuable to AFKL.

2.2.4 API management

API management is the entire process, in which an organization develops, oversees and controls APIs (Kopetenski, 2019). API management is carried out to check whether the APIs are doing the right actions in the correct way. API management is essential because it ensures that all APIs are secure, scaled correctly, governed properly, analyzed continuously and, not to be forgotten, it creates revenue(Abbas, 2019).

API security

For all APIs, security is crucial. APIs have access to lots of data, which needs to be protected carefully. API security consists of multiple components, namely authenticating and authorizing user access to APIs and protection of sensitive data by usage of policies, to make sure there are no leaks or composure of sensitive data. To ensure the security, several measurements can be taken. To start off, it helps to identify the different problems. If the vulnerabilities are known, it is easier to adapt the system or program to prevent escapades within its data. Security tokens could also be an effective way to improve API security. Tokens create an authentication process that needs to be completed before the communication can continue. This will prevent unwanted users to use the data that is being used. Within AFKL this is also being used. Thirdly, encryption is an option. Encryption works in a way where the data is disguised where it can only be used when the correct encryption key is given. If the correct decryption key is not given, the data will be depicted as a bunch of random numbers and letters, which makes the data useless. Another option is to make use of OAuth, or Open Authorization. OAuth dictates the way of receiving access codes. Furthermore, a decision can be made to use throttling and quotas. Using throttling and quotas limits the accessibilities of the system. A throttle lowers the speed of the transition of data and quotas limit the amount of data that can be transferred. A way of attacking a server nowadays is a DDoS attack. A DDoS attack overwhelms the system to break it down. By using a throttle or quota, the risk for these kind of attacks is minimized. Moreover, an API gateway can be used. An API gateway is settled between the client and the backend and serves as a reverse proxy, where a reverse proxy is a server that sits in front of web servers and forwards clients to the web services (Sommerlad, 2003). Lastly, the zero trust approach can be applied. With the zero trust approach, the user always needs to authenticate itself. This ensures that there are no non-authenticated users using the server (Siriwardena, 2020).

API governance

Governance is the process of managing decision making and decision implementation (Medjaoui, 2019). API governance has several goals, but focuses on creating a pleasant user experience. Furthermore it also includes discoverability, the lifecycle management and documentation of the API, combined with its reusability.

API governance can be applied in numerous matters, for instance by making sure the different teams make decision in a standardized way, which is the same for all API teams and decision making processes (Fernandez, 2019). Applying API governance may bring challenges, especially when the business is growing. Most challenges occur in the edges of the company. With the edges, it is meant the parts of the company that is between two components of the company. The problems that occur are challenges when the API it regards is considered in two or more teams. It has to be clear which team or which managers make the decisions and the decisions have to be made in such a way it is convenient for all the different teams. This has to be clear before the decision and its actions are made, to prevent different teams doing the same task in their own way.

According to Kin (2020), there are some strategies to make API governance effective. Some of these include standardizing metadata across API to track analytics, deploying common schema across all API requests and responses and having a definition of success, while also measuring and reporting outcomes.

Within AFKL API governance could help fastening the processes of integrating APIs. The standardization of the integration would help all developers and the teams with following the same process of integrating the APIs. Furthermore, it also enables different employees to jump in different integrations, because all processes for integrating APIs are standardized.

API analytics

API analytics includes the real time monitoring of the APIs. This could include dashboards. API analytics is used to see how the performance is of the APIs. This is useful for both the end-users and the owners of the APIs. API analytics helps to gain real time data insight on numerous aspects, such as consumption, performance and response time (De, 2017). In addition to this, it can, as mentioned with the API analytics, it can help securing the data from attacks, because with analytics the consumption is being tracked and it might give indications that an attack is happening. The analytics can also help to improve the customer satisfaction by measuring their needs and looking into the improvement options. This research will generate a dashboard for AFKL, which is used as tool to give an insight on the performance of an APIs within the company.

2.2.5 Key performance indicators

Key performance indicators (KPIs) are indicators that show the performance on aspects that are essential for a business or organization. Every organization and part of organizations have their own desires and goals (Orta, 2019). Therefore, they also have their own suitable KPIs. KPIs focus on improving those aspects. There are several key focus areas when it comes to KPIs, such as sales, project management, marketing, customer service, operations and finance. Quantitative KPIs are data measurable on raw data, while qualitative KPIs need human input to become measurable. Furthermore, there is a difference between leading indicator and lagging indicators. Leading indicators look into the future and measure what might happen, while lagging indicators focus on the current performance (Blanco, 2018).

2.2.6 KPIs on APIs

As mentioned before, department and stakeholders might have different interests and therefore a need for different KPIs. Another thing to keep an eye on is the state of the API. If the API is in an early phase of its lifecycle, there might be a different focus than when the API is in its maintaining stage. An example is that you might be more focused on initial performance when it is in an earlier phase, while you want a reliable API when it is in the maintaining phase.

For API management, four main groups are interested in the KPIs of APIs. Firstly, the developers, who want to ensure that the APIs are running and the correct data sources are used. Secondly, the application engineers are interested in the correctness of the API logic. Furthermore, the product managers create the road mapping of the APIs and ensure that the needs of the customers are met as much as possible. Lastly, the business and growth managers are interested in successful usages of the APIs and niches in the market to expand to.

2.2.7 Commonly used KPIs on API management

For each of the four groups mentioned above, KPIs are listed. First the KPIs will be explained and afterwards the importance of the KPIs for AFKL is elaborated on. In appendix A and appendix B two tables are shown which visualize the KPIs that are used in the articles. The information found in the literature is adapted to be applicable in the context of AFKL.

Developers

- Uptime

  A fairly straightforward KPI is the time, expressed in %, that the API is working. Many organizations combine their uptime with a goal, their service level agreement (SLA). The trade-off between a high SLA and the costs to reach the set SLA needs to be considered. An SLA for 99.99% for example might be expensive to achieve, while 99.9% is still acceptable and induces lower costs (Stanik, 2014).

  The uptime is a crucial KPI in the API business for AFKL. There are numerous APIs that are essential for generating revenue. For instance the order API, which is an API that handles the ordering step within the booking process, which is thus crucial for creating the service for potential customers. This necessity makes a strong case to list this as an essential KPI for the dashboard, but there also arguments to debate differently. Because of the way the APIs are created, programmed and monitored, AFKL knows the uptime of the APIs is high, even though they do not have the exact percentages. Both the literature and AFKL consider the uptime as an essential KPI, therefore the given value is nine.

- CPU usage
  CPU usage is used to measure the responsiveness of the API. When the CPU usage is too high it means that there are too many users of the API or that there is a problem within the coding of the API (Giusti, 2019).
  The CPU usage of the APIs is a KPI that AFKL already tracks and anticipates on when APIs are expanding. AFKL makes sure to anticipate on time to make certain that there are no issues regarding the speed of responsiveness of the API.

Application

- Error rate
  The error rate is the number of errors the API gives when a call is made by a client per minute. Naturally, there are different types of errors, which can lead to different error rates (Ilieva, 2018; Stanik, 2014).
  The error rate is irrelevant to AFKL since the APIs are constantly monitored and tested thoroughly, which ensure there are no severe errors in the working of APIs. Another reason why this KPI is not suitable for AFKL is, is that there are many errors that are invisible for AFKL, but noticeable for the customer, which can easily be caused by the user and is therefore not an error of the API. For the dashboard, it is not a necessity to have both the error rate and the uptime. Because of that this KPI is redundant and considered together with uptime.
- Average latency
  The API latency or elapsed time is an essential KPI known to measure API performance. The average latency is the average elapsed time before an API is executed (Ruiz, 2018; Zhang, 2021).
  For AFKL the average latency is a good indicator because it gives an indication on how the API performs and how the user conceives the API. Certain APIs, such as the check in API, are used by huge amounts of customers. If the elapsed time is too high, this could lead to many complaints. Because of the value that is already shown in the literature and the necessity of a short elapsed time within AFKL, a value of eight is given for this KPI.

Product management

- API usage growth
  For most organizations, it is important to have insight on their growth and to keep on growing. API usage can be measured over any time period, but it is important to make a consistent timespan, for instance using four weeks instead of a month, because of the difference in number of days and that weekends might be less busy compared to weekdays(Suefert, 2015; Tsolkas, 2016).
  The API usage growth is an important KPI for AFKL, since it is a good indication of the maturity of the API and whether changes are necessary. Several changes can be required. If there is a rapid growth changes regarding CPU level for instance might need to change or more personnel needs to be focusing on that API. If the growth stabilizes, less personnel has to focus on this API and more focus can be put on creating new APIs or the support on growing APIs. This KPI is more specified in section 2.3.
- Unique API customers
  To get a more precise insight on your growth, the number of unique API customers is useful (Ilieva, 2018). When just measuring API usage, it might seem that there is a positive trend, while only one user is causing the increase of the usage, which might not the eventual goal of

the organization (Ravasan, 2018).
This KPI is considered very important to AFKL because it gives a good insight of the usefulness for different customers. Furthermore it can be useful to see whether new customers create their account and actually start using the APIs. Since this KPI gives a good insight on the growth and the usefulness, a score of eight is given regarding its value for the dashboard.

- Top customers per API
  If the top customers are identified, it can help you in several ways. First of all, if the organization desires to give an improved service to the biggest customer, it can be found out by using this KPI (Ravasan, 2018). Moreover, to know the branch of your main clients might give your organization opportunities to expand the business in.
  For AFKL it could be valuable to find out which customers use the APIs most. This information can be used to give promotions or extra service when necessary. Also the customers that might need extra help with using the APIs can be found.

Business and growth

The KPIs from product management could also be transformed to business and growth KPIs, by adding a revenue component, such as top customers based on revenue.

2.3 KPIS LISTED BY AFKL

AFKL was already planning on creating a dashboard with valuable KPIs, according to different stakeholders. Below are numerous KPIs explained that were listed, in combination with the possible usefulness for AFKL. Some KPIs are mentioned both in the literature as by the stakeholders regarding API consumption.

Monthly average new customers per month

The monthly average growth is a useful indicator too see whether the usage of APIs are still expanding. AFKL selected monthly because the period is long enough to see actual increasing or decreasing numbers and seeing the trend per month, which needs to be considered as well, since seasonal factors may have influence on the usage of the APIs. To make the KPI more concrete, AFKL wants to divide the KPI into three smaller KPIs, namely the new customers for open APIs, for private APIs and for the web services.
These three KPIs as mentioned above can give great value to AFKL, since it is important to keep track on when and which new customers are gained. From this, results from marketing strategies can be determined. Furthermore, the advantage of splitting this KPI into three smaller KPIs is that the differences between the different types of APIs can be seen with one eyesight. This could help AFKL to give better ideas on promotion or offers to certain customers, for instance to try to sell private APIs to customers that are only using open APIs. Both the stakeholders and the literature consider this KPI as important. Since this is the case, a value of eight is given to this KPI.

Revenue per API package

The revenue per API package is an important KPI for AFKL to show what packages add the most value to AFKL. One of the eventual goals within AFKL is to generate revenue, therefore it is essential to know how much revenue is generated and where this revenue is coming from. Moreover, it can help indicate which packages need extra marketing or changes to generate more revenue. The value of ths KPIs, combined with the further insight it can give to AFKL, this KPI gets a value score of seven. To calculate this KPI, three different components have to be determined. Firstly, all the packages have to be listed. Secondly, what is the value of each package that is sold and lastly, how many of each specific package is being sold. The problem of this calculation is, that the number of packages sold and the value of each of the package is not exactly known at this moment.

Monthly consumption trend

To have a better insight on the total usage of the APIs, AFKL suggests a KPI on the monthly consumption trend. Again, a monthly time period is seen as a suitable time period to measure the KPI. With this KPI, problems or other changes can easily be spotted by a peak or dent in the figure and therefore AFKL can easily adapt or find deficiencies if necessary. The consumption trend is important in many different aspects of a company, also for KPIs. Because of this a value of eight is set.

Conversion rate for open/closed APIs

To see whether new users also start using open or closed APIs, it is essential to gain knowledge on the conversion rate from the accounts created compared the number of accounts that gain access to an API and also the number of accounts that use the API after they have gained access. This is important to measure because this can give an indication when certain customers are experiencing difficulties with using the API. If conversion rates are low, AFKL could take action on extra clarifications on the gaining access process or interact with the accounts that are not using any APIs to increase the conversion rate.

The conversion rate can be a crucial KPI to AFKL. The conversion rate can give an implication on whether the customer knows how the accounts work and how to gain access to the different APIs. If the conversion rate is low, this could be an indication that there are no clear instruction how to retrieve the desired APIs. Since AFKL considers this KPI crucial a value of nine is awarded to this KPI.

Number of internal users/developers

The number of internal users and developers is a valuable KPI to gain more insight on the awareness within AFKL of APIs. AFKL desires to have awareness throughout the company. If the number is too low, either throughout the entire company or in certain departments, AFKL can take action on this by giving extra information or workshops to gain more internal users and therefore more awareness. To know the number of internal users is not essential for the dashboard. Obviously it has some value to the stakeholders, but the need is not high. Therefore, a value of five is specified for this KPI.

Percentage of failed access requests for open APIs

The open APIs are easily accessible and multiple tests have already shown that open APIs are always accessible if there are no errors in the complete system or server. This means that this API would only show a percentage of zero. Therefore, this KPI is not necessary to measure.

Average access request throughput time for open APIs

As mentioned before, for open APIs there is no need to ask for access in the current system. Hence, the access request throughput time would be zero, since there are no requests.

Average access request throughput time for closed APIs

For AFKL it would be beneficiary to gain more insight on the access throughput time of the closed API to have more information on whether this process needs improvements. The problem with this KPI is, there are different starting points for consumers. Zuev (2018), also discusses the importance of the quality of the IT service, where the throughput time for a request is an important component of the service. For example, an already known customer requests access to a different closed API has a different starting point to a customer that is new. Furthermore, new customers can also gain there access through different developers or departments. Because of all these different starting points, it is difficult to create a KPI that is relevant without any skewed information. The value of this KPI is significant, since the throughput time can give a clear insight on improvement options for the APIs. This leads to a value of eight, but, as mentioned, it is difficult to determine this KPI validly.

Customer satisfaction level

AFKL has a great need to create a better insight of the customer satisfaction level. At this moment, it does not have a big priority yet, since AFKL is currently migrating a new portal. For this reason, it is more useful to dig more into the customer satisfaction level when the new portal is completely migrated. A few essential components to measure customer satisfaction are quality of the product and the price of the product (Hallencreutz, 2019; Jin, 2014). This KPI has arguably the most value for a company, since the customers are the ones who give value to a product. Hence, a value of ten is given to this KPI.

Average response time

AFKL has different developers and specialists to handle problems, questions or deficiencies. For AFKL it could be very valuable to have insight on how long the response to a request takes. This could give an indication on whether the staff needs to be trained more or whether the number of staff members is sufficient. It could also be the case that there are too many employees working on this and that staff can be shortened. The importance of this KPI is also implied by Al-Hawari (2021). In Al-Hawari's study, it was found that the response rate on one department, ERP systems, was significantly lower than all other departments. Because of measuring this KPI thoroughly, this difference was noticed and extra training was performed and extra personnel was assigned to help solving issues with the ERP system.

2.4 NEW INSIGHTS AFKL

AFKL already had an extensive list of KPIs which were thought of and found valuable, but there also a few KPIs that were proven valuable in the literature and not thought of by AFKL. The uptime is one example that AFKL did not consider before. AFKL focuses mainly on the amount of errors (error rate) but not on the duration. As Stanik explained, the uptime can give a different insight on the necessity of enhancing a specific API or program compared to the error rate. Secondly, the number of unique customers is a KPI that was not considered by AFKL before. The number of unique customers give extra value to AFKL to see how diverse the API is and how many customers are actually using the API.

2.5 SELECTING KPIS

Due to time restrictions and convenience, not all KPIs that are listed will be considered in the creation of this dashboard. As mentioned by Pillai in 2020, five to seven KPIs are optimal. Because of this, a reduction of the list needs to be carried out. Furthermore, we can conclude from the literature review and the information from the stakeholders, we can exclude some KPIs that are either not important enough at this stage, too time consuming to consider or not suitable for AFKLs situation. Firstly, an elaboration is done on which KPIs are excluded and why they are not included. Afterwards the deliberation of the assessment of the KPIs is executed.

2.5.1 Selection criteria

To avoid selection of KPIs that are not possible to measure, some selection criteria are considered. First of all, the data that is necessary for measuring has to be available and obtainable. If AFKL does not measure certain information that is needed to create a valuable KPI, it does not make sense to try and measure it. This means that if the KPI cannot be measured, it is excluded from the list to decide which KPIs to measure. Furthermore, the data has to be valid. For instance, because of the integration of the new programs, customer satisfaction level is not worked on yet, because the programs are still under construction and developed. If there would be data obtained on the customer satisfaction, the results would be lower than the expected customer satisfaction, plus that customers that have used it later than others, might have different experiences because of updates. Thus, if the data is still receiving changes and is therefore invalid, the KPI cannot be measured at this stage and is therefore not included.

2.5.2 Exclusion KPIs

The CPU usage is a KPI that AFKL is already tracking accordingly and anticipated on when necessary. Because AFKL is already focusing on this KPI, it is not necessary to include this KPI on the dashboard. Secondly, the average response time of the developers is not considered in the WSJF. This is done because AFKL is already working on a way to correctly measure this KPI and show the results accordingly. Moreover, the revenue per package cannot be included on the dashboard, this is for the particular reason that this KPI cannot be determined with the data we have at hand. In the future, this KPI could have some abundant value, thus it could be valuable to determine all the necessary information for this KPI in a later stage. Furthermore, at this moment AFKL does not have any specific strategy for top customers regarding APIs, therefore the top customer KPI is redundant at this stage. This could change in the future if AFKL decides to create promotions or separate gateways for higher rated customers, but this is not the case at this moment.

2.5.3 WSJF

Within AFKL, the weighted shortest job first (WSJF) is a common procedure to find the most valuable tasks or in this case, KPIs. The WSJF is a relatively simple procedure that creates a sequence from the most valuable KPI to the KPI with the least value.

Figure 3: WSJF

To calculate the WSJF, the cost of delay is divided by the duration of the job (Emami 2020). In our case, the cost of delay is formulated as value for AFKL. This value is the value that AFKL experts find it would add to the different areas of focus, API as a product, API consumer experience, API awareness and API provider experience. The value is determined by AFKL experts, while the duration is estimated by the researcher. The WSJF gives the following values for the KPIs from the literature and from the experts in descending order.

KPI	Value	Duration	WSJF
Failure rate / uptime	9	4	2.25
Monthly average new customers	8	4	2
Response time of the API	8	4	2
Conversion rate	9	5	1.8
Consumption trend	8	5	1.6
Number of unique customers	9	7	1.29
Number of internal users/developers	5	4	1.25
Average access throughput time closed APIs	8	7	1.14

Customer satisfaction level	10	10	1
Revenue per API (package)	7	10	0.7
Percentage failed access requests open APIs	1	2	0.5
Average access throughput time open APIs	1	2	0.5

Table 2: WSJF

From the table above we can clearly see which KPIs are necessary to be measured at this point. As mentioned by Pillai (2020), ideally, five to seven KPIs are considered. Important is that there is at least one KPI is included for each of the four different aspects the APIs are divided in, as mentioned in 2.2. We can see that for the developer aspect, the uptime KPI is included. For the application facet, the KPIs error rate and response time of the API is incorporated. Thirdly, for the product management aspect, the monthly average new customers, conversion rate and consumption trend are included. Lastly, for the business and growth, there are no KPIs that are manageable and add enough value to the dashboard at this stage. This is important to note and should be improved in a later stage when there is more time in hand. For this dashboard, we chose to visualize five KPIs. Five KPIs are chosen because of the available data and the time limitations for this research. This means that the top five KPIs as mentioned in table 2 are picked to include in the dashboard.

2.6 CONCLUSION

This chapter started with the research question ***“Which KPIs capture the performance of the API usage well?”***. From the KPIs found in the literature review and the KPIs that were already listed by AFKL, the research question can be answered with the following five KPIs that are most crucial to AFKL and are manageable to measure:

- Failure rate / uptime
- Monthly average new customers
- Response time of the API
- Conversion rate
- Consumption trend

3. GENERAL PROCESS OF CREATING AN API PERFORMANCE DASHBOARD

Figure 12 shows the general process of the creation of a dashboard on API performance. There are eight different stages visualized together with the detailed steps for each stage, together with the deliverables in each stage. This chapter is part of the design and development phase of the DSRP formulated in chapter one.

Current situation

To start off, it is essential to know what kind of company the dashboard is being created for. The goals, number of KPIs and functionalities may differ for different types of companies. Secondly, it is crucial to know all the essential stakeholders. The stakeholders are the people who will benefit most from the completed dashboard. It is vital to gain their input on what their desires are, to make the dashboard useful for them and therefore the company. The deliverables in this phase are two documents, containing firstly information on the company itself, which includes information on the type of company, what the objectives of the company are and the teams that are included during the research and secondly a file with all the relevant stakeholders, combined with information about their functions and their desires. These files will be used to ensure that the right objectives of the solutions are stated in a later phase of the process.

When these deliverables are generated, a check is performed to ensure that there is enough information on the corporation and the stakeholders. This needs to be checked by both the business and the researcher. Only when the information is considered to be sufficient, the research can proceed to the next phase.

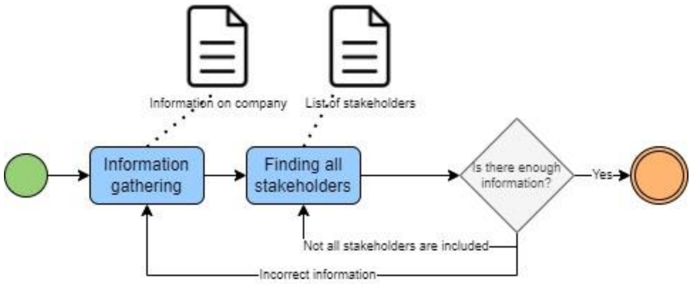


Figure 4: Current situation

Objectives

Firstly, it is important to know whether a quantative or a qualitative research needs to be carried out. This assessment will make a major difference in the types of KPIs that will eventually be listed and determined. Furthermore a list of the different objectives is necessary to find the goals of the dashboard. These objectives are needed to find the correct indicators later on in the process.

In this phase, an objective list is created as a deliverable. Again, this list needs to be checked by the stakeholders to ensure that all objectives are there and none are missing. If the list is not complete according to the stakeholders, the list needs to be updated to have a list that cover all aspects accordingly.

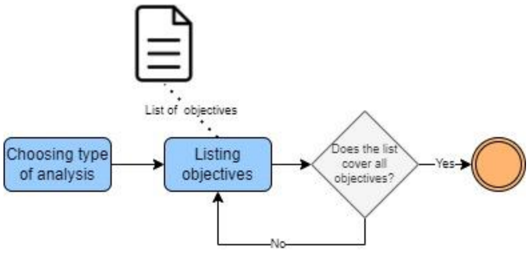

List of objectives
Choosing type of analysis
Listing objectives
Does the list cover all objectives?
Yes
No

Figure 5: Objectives

Indicators

To start with, it is convenient to look into the work that the company has already carried out. Of course, it could be the working instance could not have thought about KPIs before this research, so if there are no KPIs listed yet, there is are two options can be chosen, where the decision can be made with the stakeholders. The first option is that the researcher continues to the next step in this phase without any input from the company. If this is the case, all KPIs will be based on the research within the literature. The second option is to start from scratch and create a list with the KPIs that the stakeholders find necessary to determine, without further research. On the other hand, it may be the case that the company has already listed KPIs that they ought to be useful. If this is the case, the second step within this phase will have a starting point with a list that has already some input, which needs to be considered carefully.

When the first KPIs are listed, KPIs can be added that are proven to be useful in similar companies. Since these KPIs are proven to be effective, it is easier to convince the stakeholders of their value, compared to when they have not been proven useful or manageable before. During this step, it will most likely occur that at least one of the KPIs desired by the company is also used in the literature, either negatively or positively, since it might also be the case that the literature proves that a certain KPI does not work in the branch or on what you want to know the performance of, which is in this research APIs. If positive, the information in the literature can be used to enforce the importance of the KPIs already listed and enlighten the company on possible further advantages of using that KPI. The list that was already there with KPIs from the company, can be enlarged with the KPIs from theory. This list is then the complete list where the decision will be made on, on which KPIs will be visualized on the dashboard. For each KPI, it is important to state its value for the firm, as well as the stakeholders.

When the complete list is generated, a determination needs to be carried out on which KPIs are manageable to measure. To do this it is important to state various inclusion and exclusion criteria. This will shorten the list. After this step, the priority of each of the KPIs on the list needs to be determined, to find out on which KPIs the focus needs to be on. The top five to seven KPIs should be considered at first. Therefore, in this step it is crucial to make a clear division on how much value each possible KPI has and how much effort it would take to determine the measurement. The decision on which KPI will be visualized on the dashboard is concluded based on these scores.

When the finalized list is completed and the top KPIs are chosen, these need to be discussed with the stakeholders to confirm their value and whether they are indeed measurable and whether the stakeholders agree to the assigned value of each KPI. If this is the case, the top KPIs are known, based on the scores given before. The top KPIs are discussed with the stakeholders to ensure whether the stakeholders agree the top KPIs are the ones that are going to be visualized, they can have a decisive suggestion to make an alteration in the chosen KPIs.

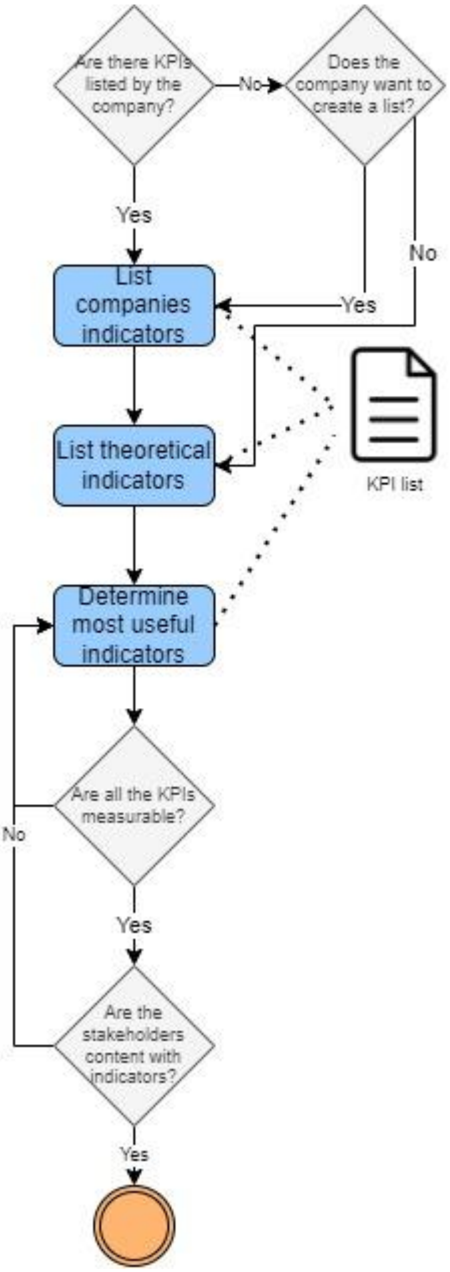

Are there KPIs listed by the company?
No
Does the company want to create a list?
Yes
No
Yes
List companies indicators
KPI list
List theoretical indicators
Determine most useful indicators
Are all the KPIs measurable?
No
Yes
Are the stakeholders content with indicators?
Yes

Figure 6: Indicators

Requirements

Before the dashboard is being built, it is a necessity to distinguish what features the dashboard needs to have to satisfy the desires of the different stakeholders. This can relate to what kind of figures the

dashboard needs to have, the timespan that is included or the colours, forms and position of the figures.
In this phase there is one deliverable, which is a document that includes all the requirements for the dashboard. It is important to have an inclusive elaboration with the stakeholders. During this discussion it is important for the researcher to know what is important for the visualization for each KPI. This can be different for each type of KPI (Few, 2006). Also, it is important to imply that these visualizations can be altered when necessary. This might be the case when priorities change.

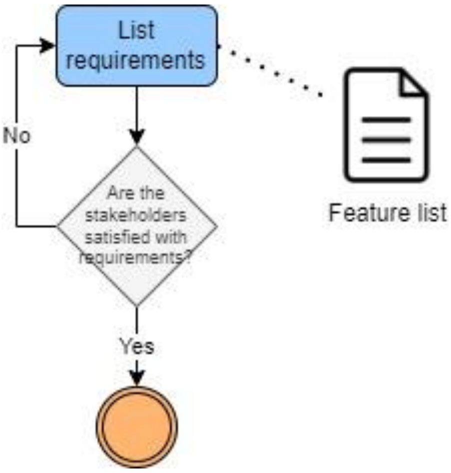


Figure 7: Requirements

Retrieving data

Before gathering the information and creating the connections between different databases and the tool, it is necessary to know how to calculate the different KPIs. There will most likely be some KPIs that consist of different components. Thus it is important to make a clarification with the different steps that need to be taken to measure the KPI. It is essential to know what information is needed to measure the KPI, to elucidate what data needs to be retrieved. This information is the first deliverable. It is important to verify the chosen manner to calculate the KPI. This can be done to take a small sample of the data and check whether the KPI can be calculated in the manner that is prescribed.
Secondly, the data sources have to be acquainted where all components of the KPIs are stored. The different components of the KPIs might have different locations of storages, which need to be combined to calculate one KPI. So in this step, the deliverable with the different parts of the KPIs is updated with the data source where that information is stored.
When all data sources are available, the connections between the data sources and the tool can be made. Note that there might be different ways of connecting to the tool. The connections need to be checked whether they work properly with the dashboard and the desired data can be loaded into the dashboard. Thus, when all necessary data is acknowledged, found in the data sources and can be loaded into the dashboard, the next phase can be started.

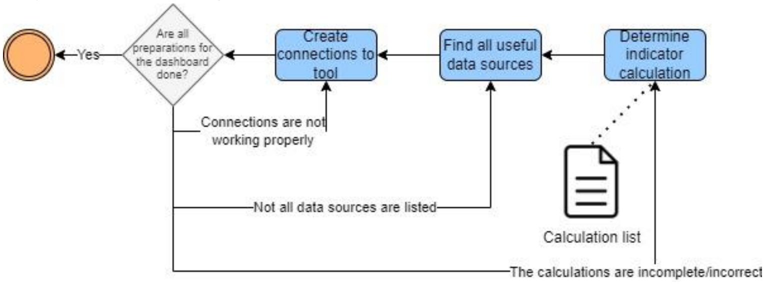

Yes
Are all preparations for the dashboard done?
Create connections to tool
Find all useful data sources
Determine indicator calculation
Connections are not working properly
Not all data sources are listed
Calculation list
The calculations are incomplete/incorrect

Figure 8: Retrieving data

Dashboard

In this phase, the database will be created, using all information gathered in the previous phases.

In the previous phases, the following deliverables were created.

- Information on the company
- Information on the stakeholders
- Objective list
- KPI list

- Requirement list
- Calculation of KPIs

All of these, except the specific information on the stakeholders themselves are used to create the dashboard. All of these are essential to make the dashboard complete and to ensure the dashboard fills the desires of the stakeholders.

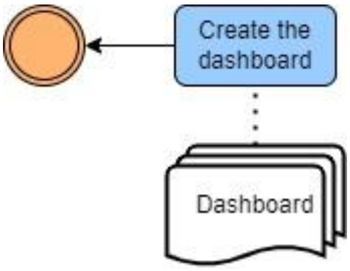


Figure 9: Dashboard

Evaluation

To find out what improvement options there are, a workshop with the most important stakeholders needs to be carried out. This workshop is used to give them a general understanding of the dashboard and the problems they come across with. These problems may be focused on numerous aspects, examples that might be included in the evaluation are questions on

- Functionality
- Visualization of KPIs
- Usability
- Credibility

In this phase there is a deliverable with the improvement options which can be improved by the designers of the dashboard. These improvement options are applied if possible to improve the dashboard.

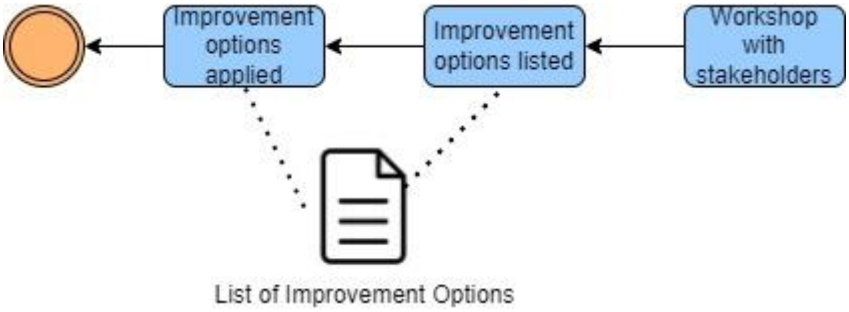

Improvement options applied
Improvement options listed
Workshop with stakeholders
List of Improvement Options

Figure 10: Evaluation

Deployment

In the last phase, the finalized dashboard is presented to the stakeholders. All functionalities need to be explained. Afterwards, the stakeholders may ask questions when necessary.

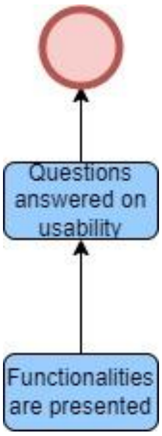

Questions answered on usability
Functionalities are presented

Figure 11: Deployment

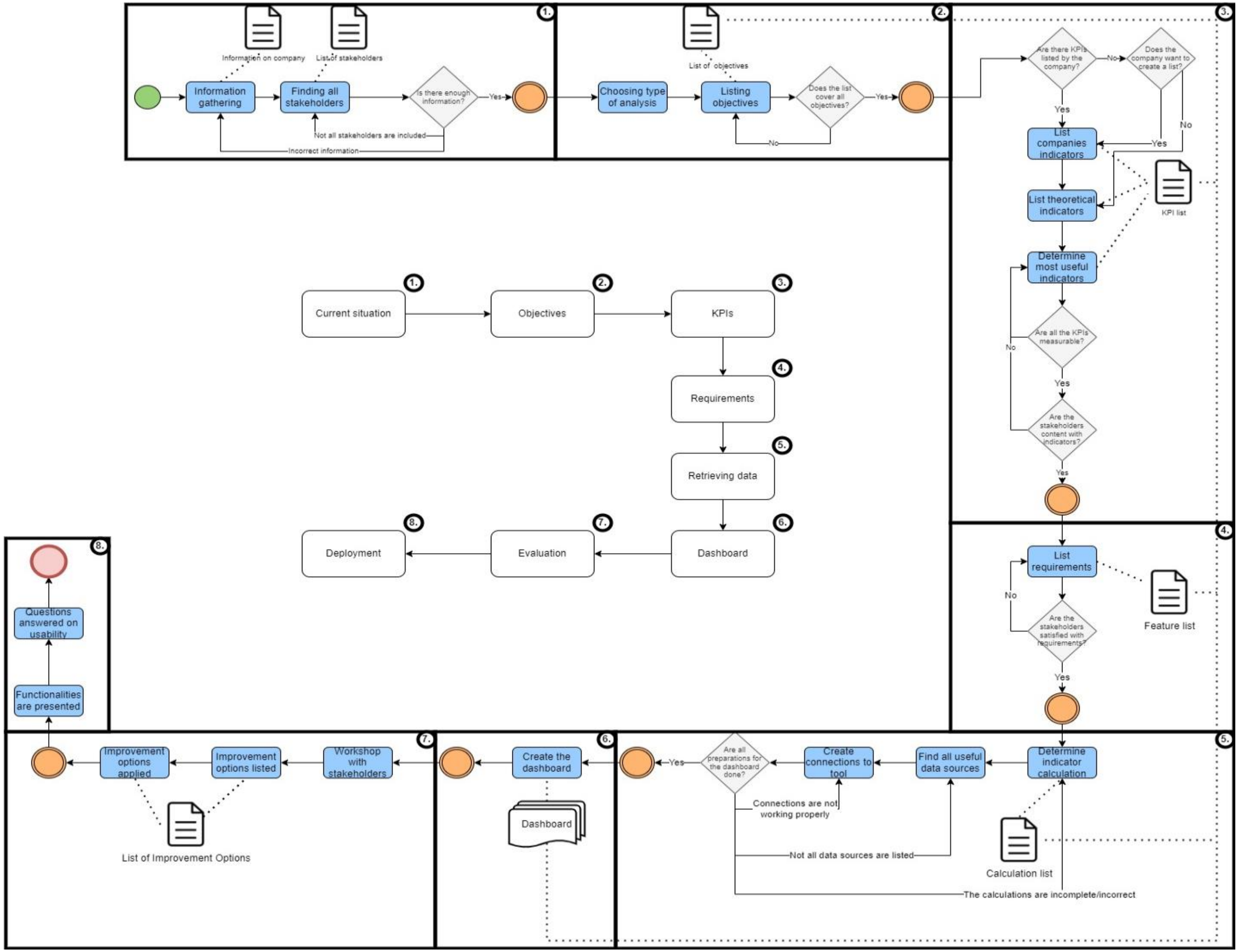

1.
Information on company
List of stakeholders
Information gathering
Finding all stakeholders
Is there enough information?
Yes
Not all stakeholders are included
Incorrect information
2.
List of objectives
Choosing type of analysis
Listing objectives
Does the list cover all objectives?
Yes
No
3.
Are there KPIs listed by the company?
No
Does the company want to create a list?
Yes
No
Yes
List companies indicators
List theoretical indicators
KPI list
Determine most useful indicators
Are all the KPIs measurable?
No
Yes
Are the stakeholders content with indicators?
Yes
Current situation
Objectives
KPIs
Requirements
Retrieving data
Dashboard
Evaluation
Deployment
4.
List requirements
Feature list
No
Are the stakeholders satisfied with requirements?
Yes
5.
Determine indicator calculation
Find all useful data sources
Create connections to tool
Are all preparations for the dashboard done?
Yes
Connections are not working properly
Not all data sources are listed
Calculation list
The calculations are incomplete/incorrect
6.
Create the dashboard
Dashboard
7.
Workshop with stakeholders
Improvement options listed
Improvement options applied
List of Improvement Options
8.
Functionalities are presented
Questions answered on usability

Figure 12: A general process of creating a dashboard on API performance

3.1 CONCLUSION

In this chapter, the generic process to create a dashboard is explained step by step. This general process is divided into eight different steps, namely the following:

1. Current situation
2. Objectives
3. KPIs
4. Requirements
5. Retrieving data
6. Dashboard
7. Evaluation
8. Deployment

Each of those steps are crucial to generate the eventual product. The finalized product is generated in such a way that it suits the situation of the company, the different objectives and requirements of the different stakeholders are met, that the most crucial KPIs are chosen and visualized, the product is evaluated and deployed within the company. This process also ensures that the product is easy to use for the different users and that bugs are fixed.

4. RETRIEVING DATA

In this chapter, the different data sources are elaborated on. It is explained what data is gathered from what data source and why this is necessary for the measurement of the KPI.

4.1 DATA SOURCES

Different types of information is stored at different locations. This is one of the reasons that the eventual dashboard will be very valuable. The dashboard will show the information of different data sources and give an overview in one eyesight, which was not available before this research.

Developer Portal

The developer portal is a self-service hub for API's providers and developers and functions as an interface for API initiative. Within the developer portal, you have the open innovation portal. The open innovation portal is a self-service hub for API's developer assets, and functions as a discovery and documentation interface for both internal and external businesses and IT stakeholders. The developer portal stores and processes the data on the accounts of consumers. The developer portal is the place where APIs and its packages can be accessed.

Common Repository

The common repository is an extensive data source within AFKL. Various performances regarding software programs is stored in this data source. COMO is based on the usage of ElasticSearch, which is an analytics engine for all sorts of data. Within AFKL, Kibana and Grafana are used for further analytics on the data, with some visualizations. Various queries can be created to form graphs in Kibana or Grafana from the ElasticSearch.

SOARepo

SOARepo is the location to find lots of different data. It is a standardized and governed collection of complementary services. The AFKL SOA repository contains all business domain's API and CORE Business Services, as well as the Software Services and Software APIs. It provides visibility on the service portfolio, and provides all information that is needed by anyone interested in making use of any of those services inside the AFKL enterprise. It enforces several SOA governance principles. Within SOARepo all the different software services are stored, together with some essential information regarding those services, such as what versions they are in, the maturity and who is the developer of the software service.

4.2 DATA TIER

The data tier, or database tier, is where the information processed by the application is stored and managed. Since there are more than one database used, there is also more than one management system used. Within the COMO database, a relational database management system is used behind the Elastic Search. A relational database is based on a relational model, which is a logical way of representing the data in tables. Each row represents a record with a unique ID, or key. The columns hold the value of the attributes (Magouryk, 2021). The relational model is designed to solve the problem of multiple arbitrary data structures. This is done by representing and querying the data in a standardized way. Because of this standardized way, the data can be used by all applications. Furthermore, the tables within the database are easy to work with and queries are easily applied. The queries could lead to an increased performance of the database. Not only the COMO database makes use of a relational database. SOARepo and the DevPortal make use of an Oracle database, which is also a relational database.

4.3 VISUALIZATION SOFTWARE

Within AFKL, TIBCO Spotfire is the main visualization software that is used. AFKL has its own licenses and also its own service employees if there are any problems regarding Spotfire. Because of the insight knowledge within AFKL, it is a wise decision to use the same visualization tool, instead of introducing a new tool. This makes it easier to create and also to use later by the AFKL employees. Secondly, AFKL also uses the tool within Grafana and Kibana to visualize information for the information that cannot be retrieved within spotfire and is stored in the elastic search.

4.4 DATA CONNECTION TO SPOTFIRE

Ideally, there will be a direct connection from Spotfire to the data source. This is most convenient because it would ensure real life data can be used automatically. If this is not the case, data needs to be loaded in manually, by for instance an Excel file.

Database
TIBCO Spotfire

Figure 13: Ideal data connection

Data architecture

The data architecture is visualized in figure 14 below. Within the business data, all the components are included that acquire data and send this to the data base. These sources are the mashery, which is the gateway of API, the API itself and third party data, such as customers of AFKL. At this stage, there is as little conversion to the data as possible. The data is sent to the data warehouses of AFKL, which are the Elastic Search and the SOARepo database. Here the data is processed and structured by a managed data model, reflecting the global direction of the final use of the data (Simon, 2020). The data is stored, before it is applied in the data marts. The used data marts are SOARepo, the developer portal and grafana. The data marts contain the aggregated data set for the specific usage of a particular business function. The data from the data marts can be used by consumers to clarify, or gain knowledge about the APIs.

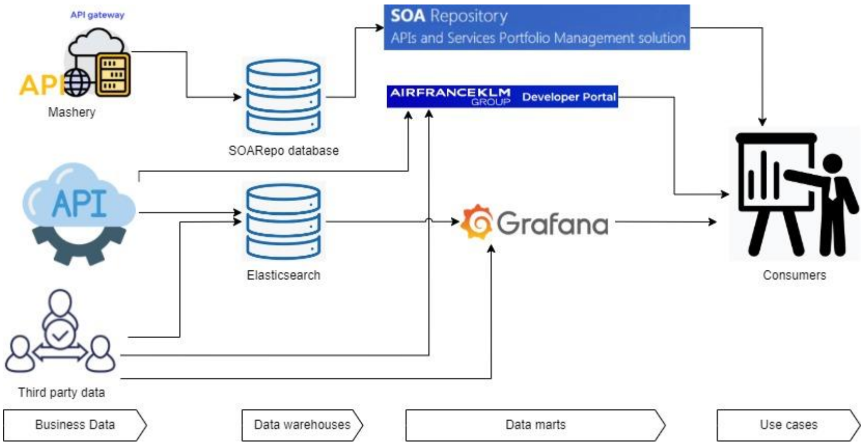


Figure 14: Data Architecture

Data connection developer portal

The developer portal cannot have a direct data connection with Spotfire at this moment. This is caused by the fact that the developer portal is still being migrated and developed. Because of this the data is not stored properly yet, which makes it impossible to gather through Spotfire. In practise, the data is stored at various locations within de developer portal, which is logical for the developers, but it is not stored in such a way, that a combination of data can easily be retrieved. Therefore, an Excel sheet is created with the necessary data from the developer portal.

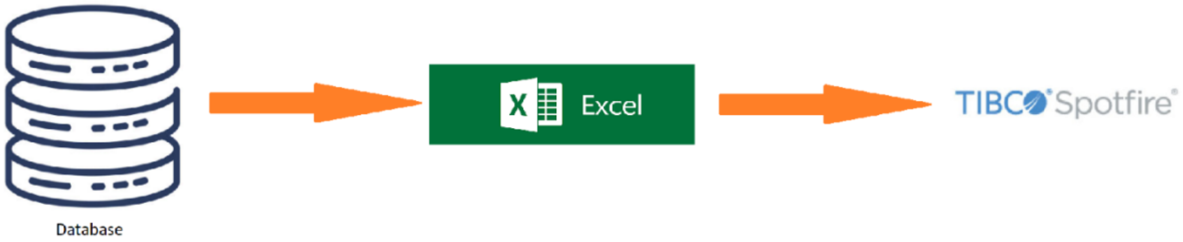


Figure 15: Connection developer portal

Data connection COMO

To create a direct connection between the Elastic Search of COMO, a Dremio connection to Spotfire needs to be established. Unfortunately, there is a problem within the Dremio software. At this moment, the newest version of Dremio is 0.9.1, but to create a Dremio connection within Spotfire, version 1.3 of Dremio is required. A case was made to TIBCO, but unfortunately, this update will only come July 2023. Because of this, there will be no direct connection from COMO to the Spotfire tool.

Data connection SOARepo

To create a direct connection from SOARepo to Spotfire, the Oracle ODAC is needed. Once this driver is installed, the data can be retrieved straight from SOARepo. It is important to note, that when a user wants to look into the Spotfire dashboard, the user will also need the Oracle driver on its device to see the visualizations that use the data from SOARepo.

4.5 CALCULATING KPIS

The KPIs that are chosen can be calculated from the data that can be retrieved from the different databases. But for a number of KPIs, some steps have to be taken to get to the correct value. These steps are explained below.

Failure rate

The failure rate is calculated by using the data within COMO. There are different types of failures, such as the 400 and the 500 error codes, where the 400 error codes give an indication that there went something wrong at the user side, while the 500 error codes give an indication there was a failure within the API or a program related to the API at the end of AFKL. To find out what the failure rate is, the following information is essential

- Number of calls
- Number of fails

Where the failure rate is calculated with the following formula, $\frac{number\ of\ fails}{number\ of\ calls} * 100\%$.

Equation 1: Failure rate

Monthly new customers

The monthly new customers can be examined by the validation date of the keys for API plans, given by the developers. This data is stored in SOARepo. The validation date is used as the starting date for the new customers and thus, the monthly new customers can be extracted from this. An important note on this KPI is, that not all the validation dates are known. Earlier, the developers of the API community did not have to add the validation date. This means that for the consumers that were validated before June 2021, the validation date is not available.

Response time of the API

The response times of all APIs are stored within COMO in seconds. These values will also be visualized in seconds. The stored data is ready to be visualized without any adjustments or further calculations on the data.

Conversion rate

The conversion rate is a KPI that consists of two different components. Firstly the number of accounts created on the DevPortal is compared to the number of accounts that have a key, where a key is acquired by an account when access to an API or API package is granted. From this we can see how many users of the DevPortal actually acquire an API. Secondly, also the number of keys that have been granted compared to the number of keys that have been used.
This means for this KPI we need to know the following four facets:

- What are the number of accounts on the DevPortal?
- How many of the accounts on the DevPortal have obtained at least one key?
- How many keys have been granted?
- How many keys are in use?

Consumption trend

The data regarding consumption of each API is kept within COMO. This KPI also does need any adjustments to be visualized.

4.6 CONCLUSION

In this chapter, the focus is on the following research question, "***How can we retrieve all necessary data?***". The data connections to SOARepo, the DevPortal and COMO have been explained. It is shown that there are no direct connections possible between the visualization software and the Devportal and COMO. Next to that, there is no connection possible between COMO and Spotfire either. This causes that there the data that needs to be retrieved from the DevPortal is downloaded to Excel and uploaded manually into Spotfire. Next to the retrieving of the data, the calculation of all the KPIs that will be visualized is shown.

5. EXECUTING THE PROCESS

In this chapter, the three tier architecture is elaborated on for this specific case within AFKL. The three tiers are data, application and presentation (Fernandez, 2008). In a three tier application, all information goes through the application tier and the presentation tier and the data tier do not have direct communication. Next to that, the process of chapter five is carried out. The current situation phase is described in chapter 1.1, 1.2 and 2.1. The objectives of the second phase, is also conducted in chapter 2.1. The third step, which is to find the desired KPIs, is performed in chapter 2.5.3, where we found the following KPIs, failure rate / uptime, monthly average new customers, response time of the API, conversion rate and consumption trend. The requirements phase and the retrieving data phase are discussed and elaborated on below, in chapter 5.1. The seventh step, or the evaluation step, is described in section 6.1.

5.1 APPLICATION TIER

Ideally, the dashboard would completely be created in spotfire and therefore be entirely visible in spotfire. Because COMO is still being migrated and improved, this is not possible at this moment. As mentioned in section 4.3.2, there is no direct connection possible between spotfire and the Elastic Search behind COMO. To work around this problem, four possibilities have been considered.

The first option would be the most logical one and is also used for the data from the developer portal, namely, creating a file that could be loaded into spotfire. When data is downloaded from Kibana or Grafana, a CSV file is created, which can be loaded directly in spotfire. Unfortunately, there is a problem when a larger timespan is asked within Kibana and Grafana. For graphs that need larger amounts of data, such as the error rate, Kibana and Grafana will give an error, a server time-out to be precise, when the timespan is longer than 24 hours. The developers are still working on tackling this problem. Since these errors occur, the data cannot be downloaded when using a proper timespan. Therefore, this option is off the table.

The second option that is considered is loading the data from the other data sources into COMO. Unfortunately, it is not possible at this stage to bring other data into the Elastic Search behind COMO. The developers of AFKL do not allow data from other sources to be loaded into Elastic Search. Also, even if the developers would allow the data to be loaded in, it has to be loaded in differently with modifications, since Elastic Search is not a classic database, but more a search and analytics engine. Because of these reasons, this option is also considered to be unfeasible.

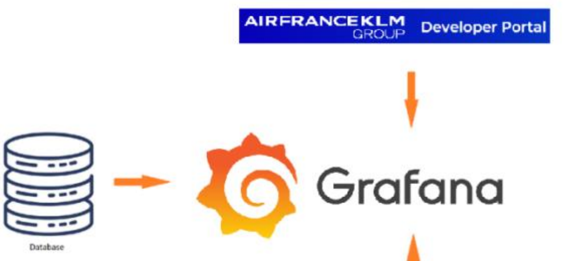


Figure 16: Connections if dashboard would be created in Grafana

Thirdly, to ensure all graphs are together in the same dashboard, screenshots of the graphs created in Grafana can be added into the spotfire dashboard. Even though this option adds the plus of having all graphs in one eyesight, it is considered to be not very useful, because when there is only a picture on the dashboard, it is impossible to look into the data, nor is it possible to focus on a specific aspect. Consequently, this option is also found to be an inconvenient option.

The fourth option is having a direct link to the COMO dashboard, within the Spotfire dashboard. The downside of this option is, that the users need to take another step, to see some of the data. The plus side from this option, is that the graphs and figures can be explored thoroughly, focusing on each aspect that one wants to deepen into.

All arguments considered, the fourth option is chosen to be the best option and is therefore used for the time being, until either a direct connection between Elastic Search and spotfire can be created or a proper file with the necessary data and a reasonable timespan can be downloaded.

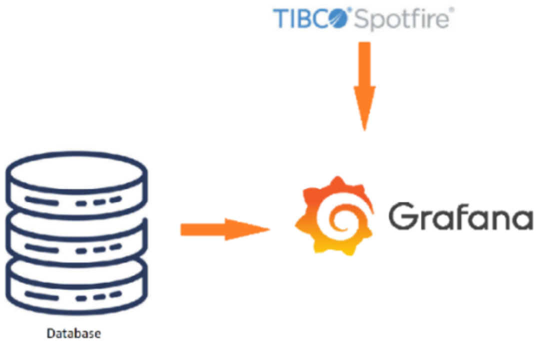


Figure 17: Connection COMO

5.2 PRESENTATION TIER

In section 5.2 the “dashboard phase”, mentioned in chapter 3 is being executed. This section will be divided into two components, one for the functionalities of the Spotfire dashboard and one for the Grafana dashboard.

5.2.1 Spotfire Dashboard

At the bottom, we can see the different pages of the spotfire dashboard. Each of the pages show their own KPI, except for the first page, which is the introduction page, where also the link to the grafana dashboard is shown, with a short text to elaborate on the differences between the two dashboards, why there are two dashboards and what is shown on the grafana dashboards. Next to that, specific values for that page are shown, such as the number of rows that are used, the marked values, and the number of columns.

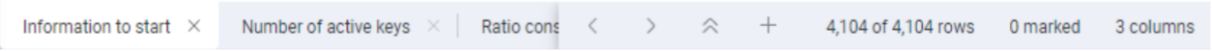
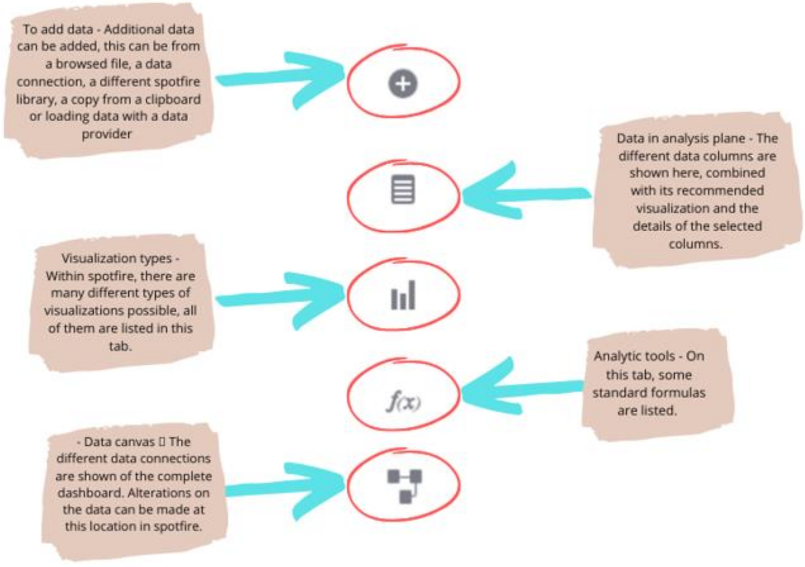


Figure 18: Bottom bar spotfire

At the left hand side, five different icons are shown and these are explained in the figure below.



To add data - Additional data can be added, this can be from a browsed file, a data connection, a different spotfire library, a copy from a clipboard or loading data with a data provider

Data in analysis plane - The different data columns are shown here, combined with its recommended visualization and the details of the selected columns.

Visualization types - Within spotfire, there are many different types of visualizations possible, all of them are listed in this tab.

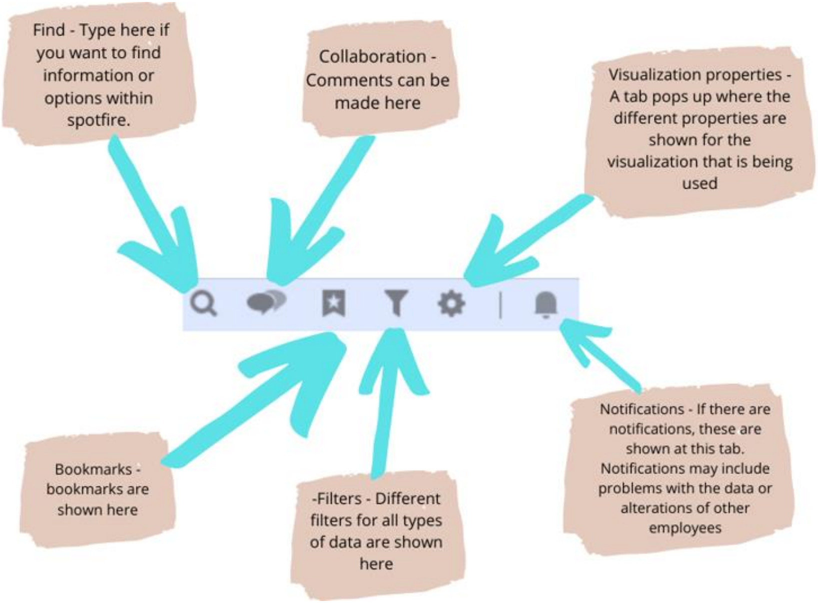
Analytic tools - On this tab, some standard formulas are listed.

- Data canvas ▯ The different data connections are shown of the complete dashboard. Alterations on the data can be made at this location in spotfire.

Figure 19: Left bar spotfire

At the top on the left hand side, we see the standard tabs when one is working on a document, namely the ‘file’, ‘edit’, ‘data’, ‘visualization’, ‘view’, ‘tools’, ‘help’ and ‘user’.

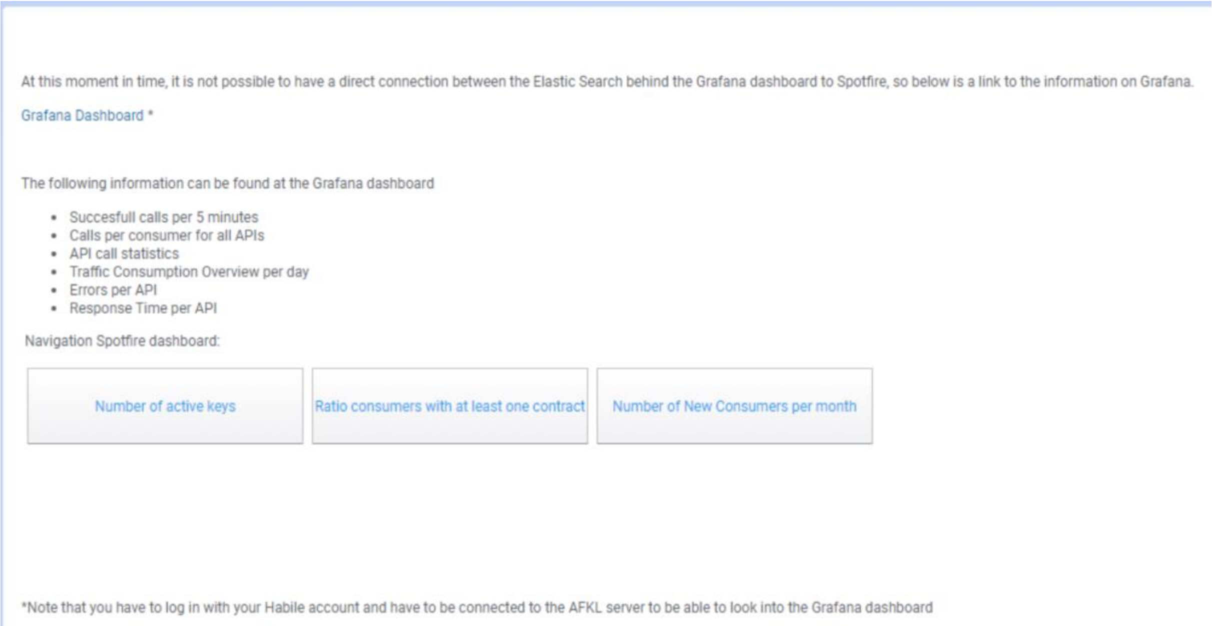
At the top on the right hand side, six buttons are shown.



Find - Type here if you want to find information or options within spotfire.
Collaboration - Comments can be made here
Visualization properties - A tab pops up where the different properties are shown for the visualization that is being used
Bookmarks - bookmarks are shown here
-Filters - Different filters for all types of data are shown here
Notifications - If there are notifications, these are shown at this tab. Notifications may include problems with the data or alterations of other employees

Figure 20: Functional buttons spotfire

The introductive page is depicted below. The page starts with a brief explanation on why there are two dashboards, with the link to the grafana dashboard. Also an enumeration of the KPIs shown in grafana is given, to inform the user already what KPIs are shown on the grafana dashboard, without havng to open it. Lastly, there are three buttons added that the user can press to be redirected to the linked page. There is one button for each of the KPIs that are shown in the spotfire dashboard.



At this moment in time, it is not possible to have a direct connection between the Elastic Search behind the Grafana dashboard to Spotfire, so below is a link to the information on Grafana.

Grafana Dashboard *

The following information can be found at the Grafana dashboard

- Succesfull calls per 5 minutes
- Calls per consumer for all APIs
- API call statistics
- Traffic Consumption Overview per day
- Errors per API
- Response Time per API

Navigation Spotfire dashboard:

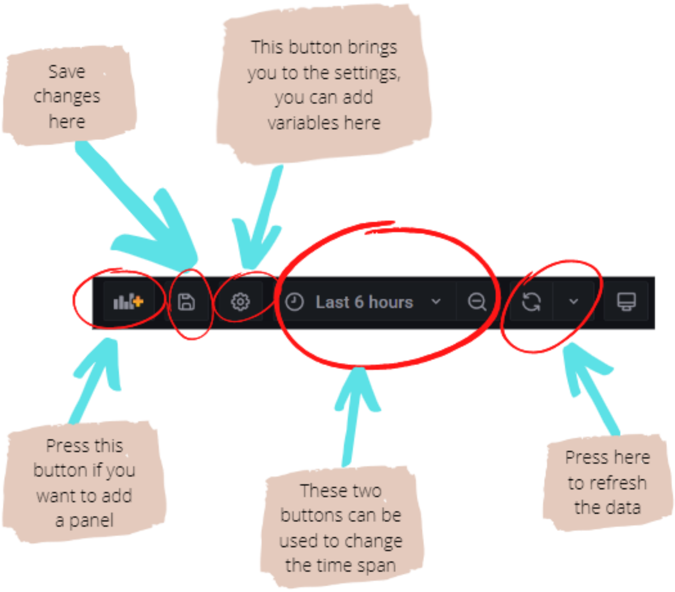
Number of active keys
Ratio consumers with at least one contract
Number of New Consumers per month

*Note that you have to log in with your Habile account and have to be connected to the AFKL server to be able to look into the Grafana dashboard

Figure 21: Introduction page spotfire dashboard

5.2.2 Grafana Dashboard

In figures 22, 23 and 24 a brief explanation is given on the functionalities within Grafana.



Save changes here

This button brings you to the settings, you can add variables here

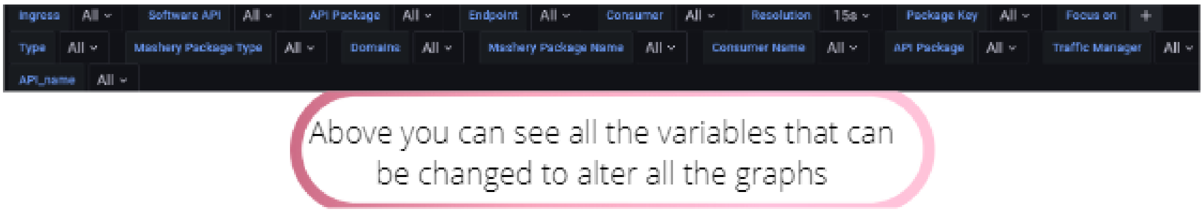
Press this button if you want to add a panel

These two buttons can be used to change the time span

Press here to refresh the data

Figure 22: Toolbar Grafana

In figure 22, the toolbar of Grafana is shown, together with its functionalities. An important note, which has been mentioned before, that it is not possible at this moment to increase the timespan massively. Developers are working on this problem, but it has not been fixed up to this day.

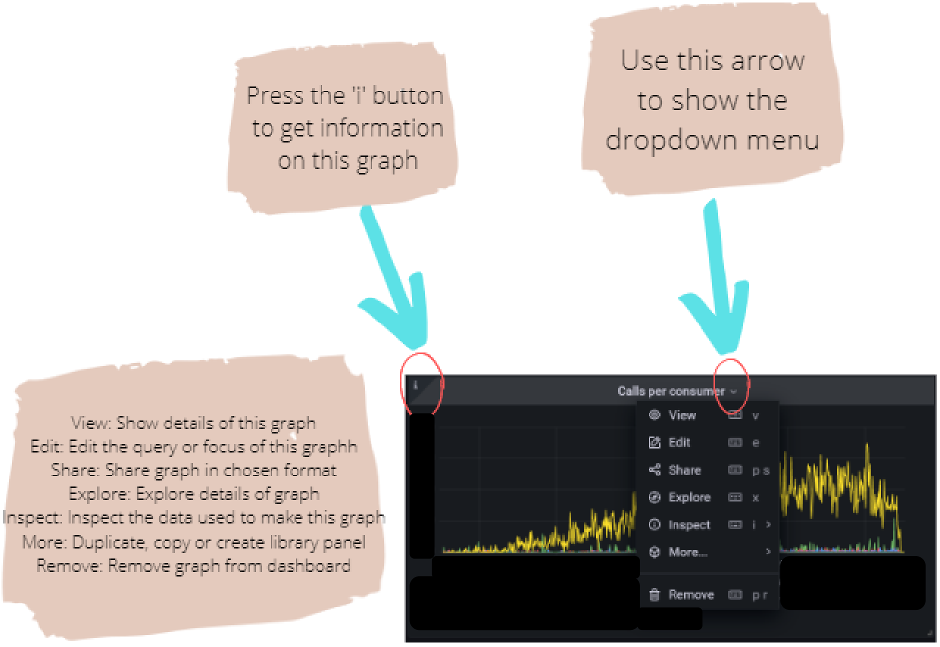


Ingress All · Software API All · API Package All · Endpoint All · Consumer All · Resolution 15s · Package Key All · Focus on +
Type All · Mashery Package Type All · Domains All · Mashery Package Name All · Consumer Name All · API Package All · Traffic Manager All
API_name All

Above you can see all the variables that can be changed to alter all the graphs

Figure 23: Variable list Grafana

In figure 23, shown above, a list of the variables can be found that are used in the various graphs. For each of the variables, it is possible to select one of the options. When selecting a specific outcome of a variable, such as an API, the graphs that have the variable API, will be altered to only the chosen API. One variable that is important is the resolution. The resolution is the maximum time that the dashboard may take to load in the data. If more data needs to be included in the graphs, a higher resolution time is needed.



Press the 'i' button to get information on this graph

Use this arrow to show the dropdown menu

View: Show details of this graph
Edit: Edit the query or focus of this graphh
Share: Share graph in chosen format
Explore: Explore details of graph
Inspect: Inspect the data used to make this graph
More: Duplicate, copy or create library panel
Remove: Remove graph from dashboard

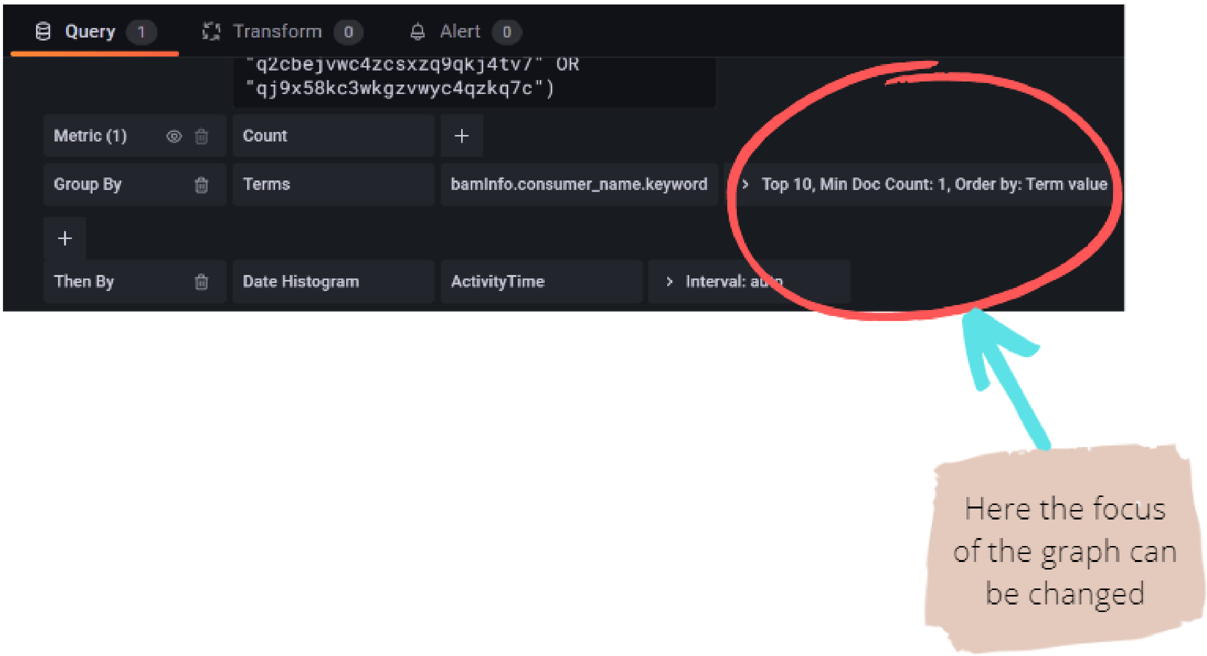
[redacted]

[redacted]

[redacted]

Figure 24: Grafana graph details

In figure 24 the possibilities for each graph is shown. Here are several options possible to change the graph or view. It is important to know you can also change the focus of the graph when you go to 'Edit'.



```
"q2cbejvwc4zcsxzq9qkj4tv7" OR
"qj9x58kc3wkgzvwyc4qzkq7c")
```

Metric (1) — Count
Group By — Terms — bamInfo.consumer_name.keyword — Top 10, Min Doc Count: 1, Order by: Term value
Then By — Date Histogram — ActivityTime — Interval: auto

Here the focus of the graph can be changed

Figure 25: 'Edit' section Grafana

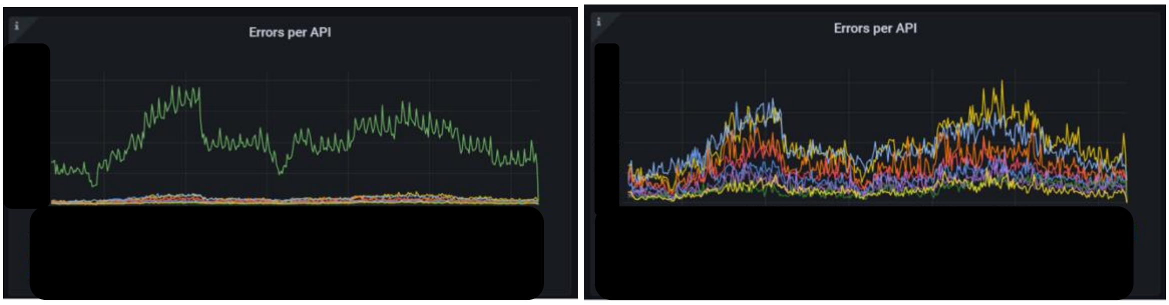
In the 'Edit' page, the focus of the graph can be altered. We can increase the size of the list, which was in this specific case the number of consumers. As it stands, only the top ten consumers are shown in the graph. We can for instance increase the list or show the bottom ten.

5.2.3 Visualizations KPIs

For each of the chosen KPIs a visualization is made. In appendix C the complete dashboard is shown.

Failure rate

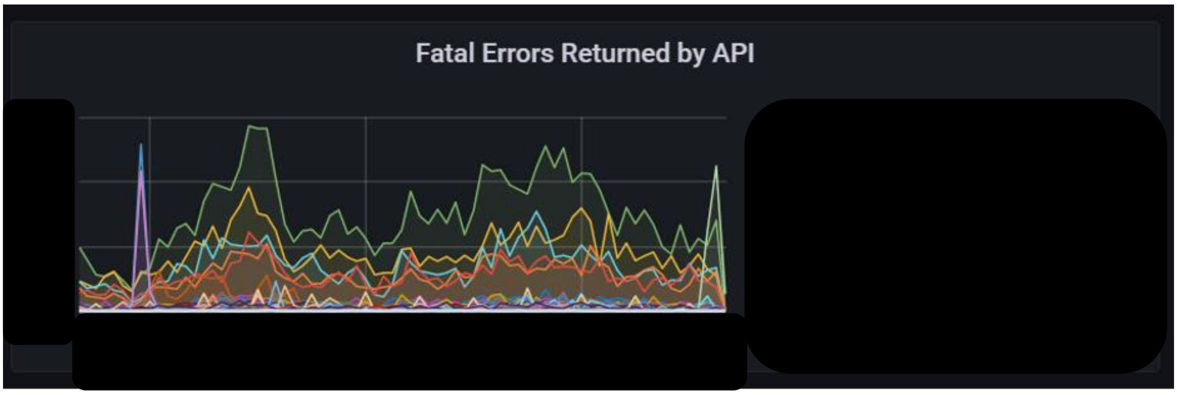
For the failure rate, three different graphs are shown to visualize the number of errors within grafana.



[redacted] [redacted] [redacted] [redacted]

Figures 26 and 27: Errors per API in grafana

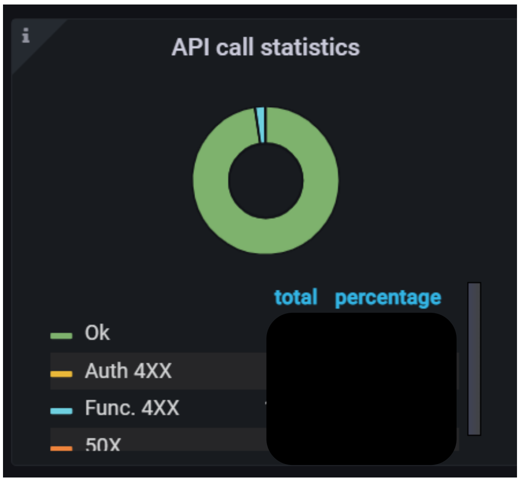
In these graphs, the number of errors per API are shown. In this case, the top ten values are given. In this graph it might be useful to make a distinction between the APIs, since the first graph is a bit out of proportion, caused by the "REST, other", which indicates that this is a combination of multiple APIs. This is stored this way by AFKL. In the second graph, all APIs are selected, except the previously mentioned "REST, other". In the second graph, it is clearer to see how the different values compare to each other. Line charts are useful to show different values when a scale is used, such as in this case a time scale is used, with different APIs that need to be displayed (Doerr, 2018).



[redacted] [redacted] [redacted]

Figure 28: Fatal errors returned by API

This graph shows the fatal errors per API. The fatal errors are the errors that are gained when a status code between 500 and 599 are received from the endpoint of the API. This graph adds value by making a distinction between errors that are caused by the API, compared to errors that have different causes.



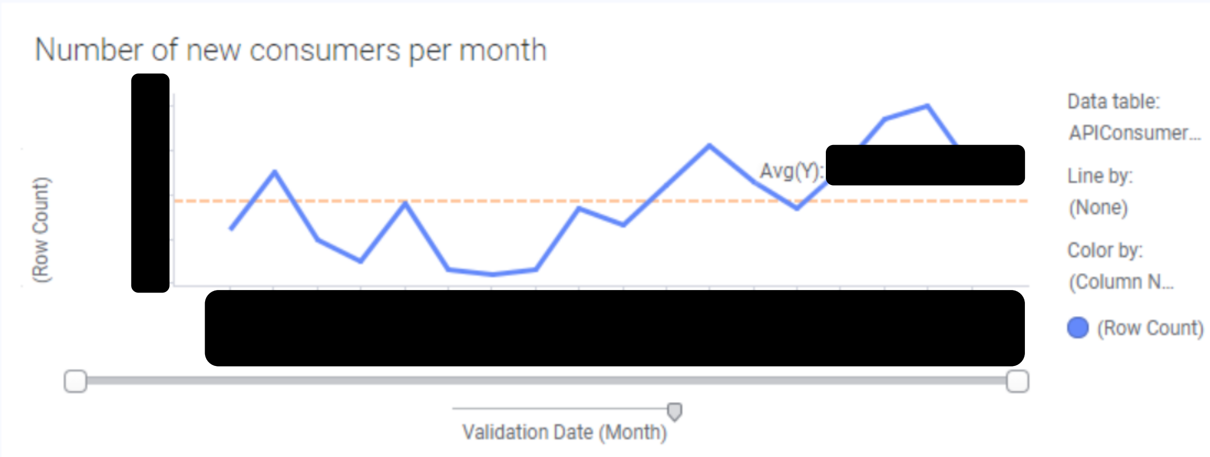
[redacted]

Figure 29: API call statistics

Lastly, we can see the number of calls, combined with the status that is received. This graph shows the ratio between the errors and the successful calls. This adds value by giving a different perspective to the number of errors, since it not only important to know how many errors occur, but also to see how this compares to the successful calls.

Monthly average new customers

The visualization of the new customers is done in spotfire. The validation date is used to determine the monthly new customers.



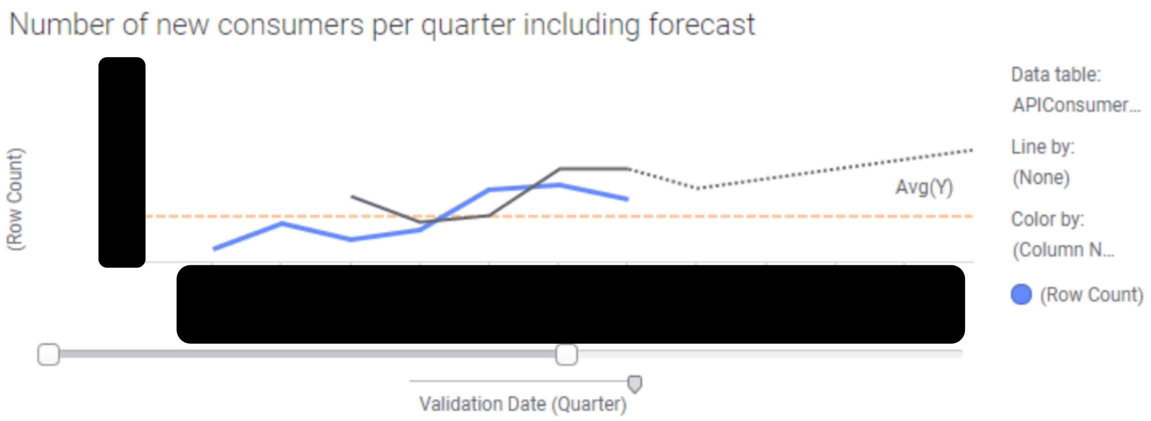
[redacted]

[redacted]

[redacted]

Figure 30: Number of new customers per month

In this graph, one can see the values that fluctuate a lot, with an average 99999. It is important to note, as mentioned earlier, that a lot of validation dates are unknown. This causes that only 46% of the rows are used to create the graph. This is important to note, since this could have a major impact on both the graph and the average that is mentioned earlier.



[redacted]

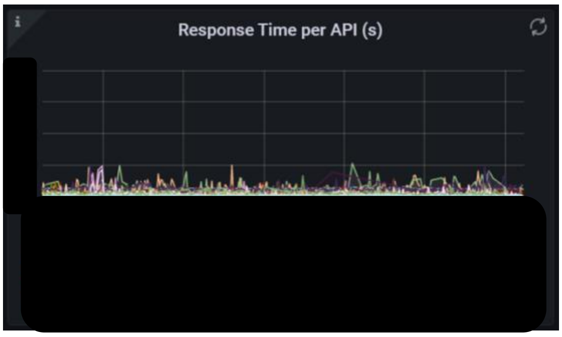
[redacted]

Figure 31: Number of new customers per quarter

In this graph, the data is more aggregated to quarters instead of months. This aggregation is done to carry out a forecast on the number of new customers for 2023. If a forecast would be carried out based on the monthly data, it is, mainly because the missing data, inaccurate compared to the more aggregated quarterly data.

Response time

The response time is shown in grafana and visualized in seconds.



[redacted]

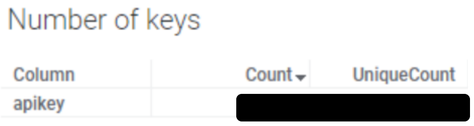
[redacted]

Figure 32: Response time

This graph shows all APIs with its respective response time. To create a more precise comparison, fewer APIs can be selected. Similarly to the visualization of the failure rate, there is one API that is out of proportion, namely the MARS REST API. Likewise, this API can be excluded to create a graph that is more proportional and a clearer comparison can be made.

Conversion rate

As mentioned in section 4.5, the conversion rate is determined based on a few different facets. The visualization is created within spotfire. It is important to note that the DevPortal, where some data regarding this KPI is coming from, is still being integrated and improved, which might have an influence on the completeness of the data.
To start off, a unique count is done to find the number of unique keys.



Number of keys

Column	Count ▾	UniqueCount
apikey	[redacted]	[redacted]

Figure 33: Number of keys

Afterwards, the keys are compared in status to find how many of the keys are in use. A pie chart is created to compare the three different statuses. The three different statuses are active, disabled and waiting. A key is active when the key is in use and there are no problems with that specific key. A key is disabled when the contract expires or there are fatal errors within that key. A key is in waiting when the contract is being processed or when the key is worked when there are problems.

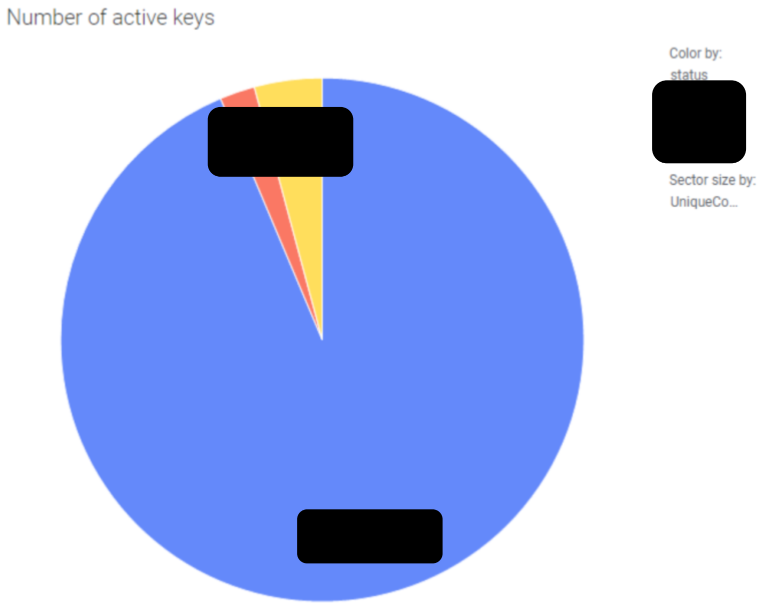
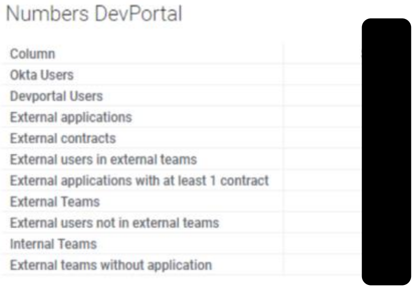


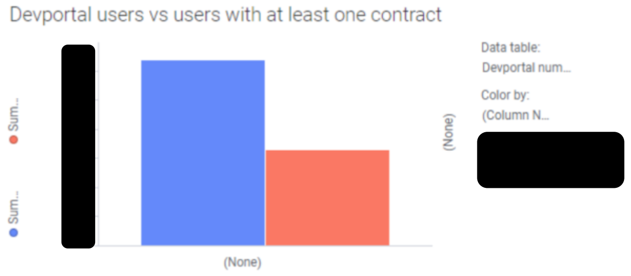
Figure 34: Comparison key statuses

In the piechart we can see that the majority, 99999% to be precise, of the keys are active. Only approximately four and two percent respectively are waiting and disabled. The absolute numbers are also shown in the chart. For this particular KPI, a pie chart is used. This is for the reason that a pie chart is beneficial when one wants to visualize part-to-whole information, which is what needs to be displayed in the graph related to the number of active keys.



Numbers DevPortal

Column	[redacted]
Okta Users	[redacted]
Devportal Users	[redacted]
External applications	[redacted]
External contracts	[redacted]
External users in external teams	[redacted]
External applications with at least 1 contract	[redacted]
External Teams	[redacted]
External users not in external teams	[redacted]
Internal Teams	[redacted]
External teams without application	[redacted]

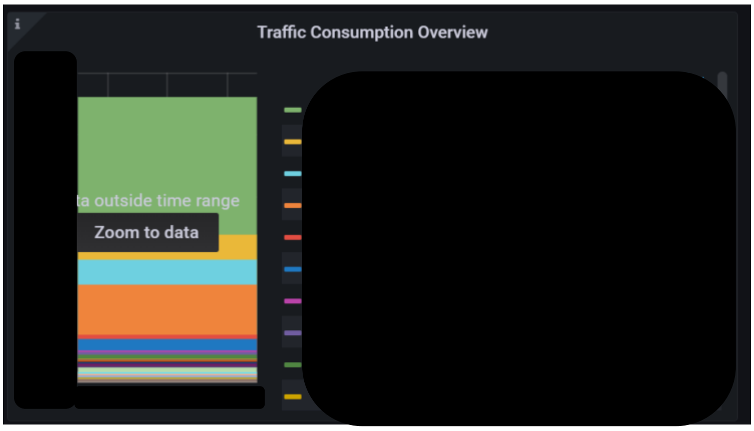


Figures 35 and 36: Information DevPortal and conversion rate

In the table, information is given on the contracts, the users and the teams that have created an account on the DevPortal. The data that is compared in the bar chart, are the total number of accounts on the DevPortal compared to the number of users with at least one contract. We can see that 99999% of the accounts also have a contract, which is a reasonable number, since there are numerous users that only create an account to see the possibilities of the DevPortal, without the desire of acquiring a contract. In this case, a bar chart is chosen as the visualization because there are two instances that need to be displayed and there is a clear distinction between the two values(Few, 2006).

Consumption trend

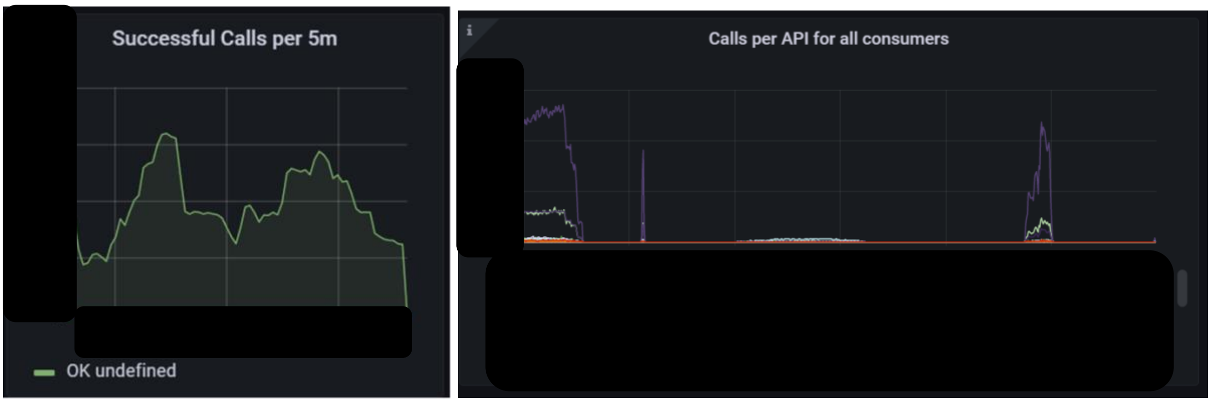
The consumption trend is visualized in grafana. In this case, it is useful to use a stacked bar chart, since it gives a good display on what the total amount of calls is, with a clear overview on what consumers have the biggest impact (Few, 2006).



[redacted] [redacted] [redacted]

Figure 37: Consumption trend

In the figure above, the total consumption trend for all consumers is shown, where at this stage, the ProductShop_Rest is the largest consumer by some margin. It is important to note that the time interval for this graph is one day. To elaborate further on the consumption trend, two graphs related to the number of calls are added to get more information on the consumption trend.



[redacted] [redacted] [redacted] [redacted]

Figures 38 and 39: Calls statistics

The graphs above show the number of successful calls and the number of calls per consumer, with a shorter time interval. These graphs give another insight on how much the APIs are being used and which consumers are using the APIs during which stage of the day.

5.3 CONCLUSION

This chapter focuses on the following research question, “***How can we design a dashboard for the listed KPIs?”***. In this chapter, the beginning shows an explanation on that there are two dashboards, with a link to the grafana dashboard within the spotfire dashboard, instead of one dashboard. This is done because of the inability within spotfire to create all necessary connections to the data sources. After this an explanation is given on both dashboard on how the tools work. Lastly, the different visualizations of each of the chosen KPIs are shown. Note that there are visualizations within both COMO and spotfire are shown.

6. EVALUATION

In this chapter, an evaluation is performed on the dashboard. The most important stakeholders gave their critiques and what is done to improve the dashboard, based on the UTAUT model, which is described in the beginning of this chapter. The questionnaire of appendix D is used to gain knowledge on the opinions of the stakeholders. There are two statement regarding the performance expectancy, five for the performance expectancy, three on the social influence and lastly, three that relate to the facilitating conditions. For each of the statements, the stakeholders are asked to give a score from one to five, with one being the lowest score and five being the highest score. When the statement is not of their interest or they do not have an opinion, the no opinion option can be used. Next to this, some notes are included when there are noteworthy points made on why a certain score is given to a statement. For this research, the statements focusing on the effort expectancy and the facilitating conditions are most crucial, because those aspects give the biggest insight of the value of the tool. An important note is, that during the questionnaires, the four variables, age, gender, experience and volitariness of use, are not taken into account. It is known that these variables do have their respective impact on the core components, but with the number of stakeholders and the essention of this research, these impacts can be neglected. The results of the evaluation can be found in appendix E, with more details in appendix F.

6.1 UTAUT MODEL

During the seventh phase, the Unified Theory of Acceptance and Use of Technology (UTAUT) process is carried out to do the evaluation, created by Vankatesh in 2003. The eventual goal of the evaluation is to create a continuous usage as visualized in the figure below, where the evaluation helps gaining acceptance from its users, where acceptance has a positive impact on the intention for continued usage.

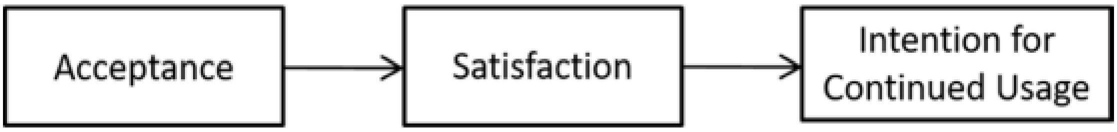


Figure 40: Influence of acceptance

The UTAUT model is shown in figure 40. The UTAUT model consists of eight different components. The first one is the performance expectancy, which is the degree to which an individual believes that using the system will help the user to attain gains in job performance. From a theoretical point of view, there are indications that the age and gender do have an impact on the performance expectancy. Researches, such as the research Minton and Schneider (1980), discovered that men are more task-oriented and therefore more focused on performance expectancy. In a similar way, is that younger people have a tendency to focus more on performance expectancy than older people.

Secondly, effort expectancy plays a role in the use behaviour. Effort expectancy is defined as the degree of ease associated with the use of the system. Vankatesh and Morris (2000) suggest that effort expectancy is more salient to women than to men. Similarly, when a user is older, it is expected to have more influence on effort expectancy. Thirdly, experience of the user has impact on the effort expectancy, with having little experience causing a bigger impact on the effort expectancy.

The third aspect in the UTAUT model is social influence. Social influence is explained as the degree one perceives others believe in the importance of the usage of the model. Social influence is influenced by all four variables we see at the bottom of figure 41. Vankatesh (2003) implies that older women, with lower experience and in mandatory usage have the strongest effect on the social influence component.

The facilitating conditions is the last core component of the UTAUT model. Facilitating conditions are defined as the degree one believes that an organizational and technical infrastructure exists to support use of the system. The influence of the facilitating conditions is impacted by the age and the experience of the user. In this case, the biggest impact is on users that are older and with increased experience.

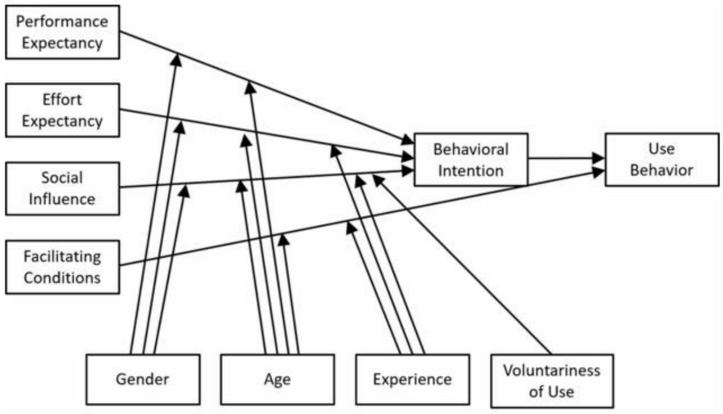


Figure 41: UTAUT model

To do the evaluation a questionnaire is created, focused on all four core components. Below all statements are shown for each aspect individually. On each statement, a score from one to five is given by a potential user, where one is the lowest score and five the highest. The form that is used for this is shown in appendix D. When all scores are obtained from the stakeholders, some details are determined, such as the average score for each statement.

Performance expectancy

- Using the dashboard increases my productivity
- Using the dashboard can help AFKL

Effort expectancy

- The dashboard is clear
- The visualizations of the graphs are useful
- The most important KPIs are visualized
- Mastering the dashboard will be easy for me
- Utilizing the dashboard is easy for me

Social influence

- People who influence my behaviour at work want me to use the dashboard
- This dashboard is desired by my manager(s)
- The dashboard suits me and my employees

Facilitating conditions

- I have all necessities to use the dashboard
- I have the knowledge to use the dashboard
- There are people available when there is a problem with the dashboard

6.2 EVALUATION PROCESS

To start off, the results from the questionnaire show that relatively often there was no opinion on the social influence statements, with twice only 50 percent and once 67 percent giving scoring votes. One of the reasons is that the teams that are working on creating the dashboards, might eventually not use the dashboard. When scores were given, there were a few low scores given. Some low scores are influenced by the fact that some managers are aloof to whether the dashboard is used to achieve goals. But the information that can be retrieved from the dashboard is also cherished to do more data driven decision making by the API teams.

The scores related to the performance expectancy are high. The average scores are 3.25 and 4.5 in chronological order. We can see that the productivity of the stakeholders themselves may not be impacted massively. There might not be a major impact because the productivity are not directly impacted with this tool. The second statement regards the helpfulness of the tool which the dashboard has more direct impact on. The average score is higher because the dashboard can already illustrate where improvement options occur or where performance is already on a high level. One important note on the performance expectancy, is that it is influenced by the lack of automation, the level of granularity and the availability of the data. These problems are known and already discussed in chapter four. If these aspects would be improved, the productivity and the value to AFKL will increase significantly. One worthy annotation is that the significance of the dashboard is currently high and that it also incenses the stakeholders to develop the dashboard further to get more extensive and precise results.

All votes on the statements regarding the effort expectancy are given, with all average scores coming out above 4. The scores regarding the easiness of use are 4.5 on average. This is a good indication on how simple it is to use. Because of this, the dashboard will, most probably, not give problems on the usage and it will be easy to utilize for all other users that will gain access after the final deployment. For the particular reason that the stakeholders find the instrument easy to use, may also help when the new users do have questions on the tool. A noteworthy comment was that there needs to be a more clear explanation on the difference between grafana and spotfire. This has been elaborated on thoroughly in the report, but not as clear on the dashboard itself. Also, the difference in time span should be elaborated on, because this is causing a bit of confusion. The data cannot be looked at together at this moment, since the grafana data has a much shorter time span than the data in the spotfire dashboard.

The statements on the subject of the facilitating conditions get mostly scoring votes, only two “No opinion” values are given. The knowledge and necessities are all sufficient and got an average score of 4.167 and 4.5 respectively. This indicates that the dashboard is ready to be used and that the users should have no problem gaining access, opening the dashboard and know the goals and value of the tool. The statement related to when a problem with the dashboard occurs has an average value of 3.75. A somewhat lower score, and the two times “No opinion” values, can be explained to the fact that it is not clear whom is going to actively develop and fix the dashboard when necessary. This might cause problems when the dashboard is giving problems, such as not depicting data correctly or when the dashboard gives errors. There are teams that work on issues on grafana or spotfire, thus those teams could help when certain errors occur, but they do not have the insights on what the dashboard has to depict, which can cause some problems.

7. CONCLUSION

In this chapter, the conclusions of the research are described, based on the used methodology and the dashboard presented to AFKL. The main research question of this research is

What is an effective KPI data driven approach to measure the performance of API management?

The research has been divided into several chapters, with a sub question answered in each chapter. Each section represents a separate step in the process of creating the dashboard. The process is visualized in figure 12. This figure shows what is carried out in each step, together with the steps in more detail. This is a general process and can be used for different cases.

The first sub question that has been answered in this report is "***Which KPIs capture the performance of the API usage well?***". To answer this sub question, a literature research has been performed to find KPIs that have been proven to be useful in API performance or software services comparable to API performance. Afterwards, the found KPIs are combined with the KPIs that were already listed by AFKL. All the KPIs have first been checked whether they fulfilled all selection criteria. When the KPIs that did not meet the selection criteria are excluded, a value of the duration of the task and the added value is given, which is utilized to make an order of most logical KPIs to add to the dashboard, making use of the WSJF principle. The eventual KPIs that are chosen to be on the dashboard are the following five

1. Failure rate / uptime
2. Monthly average new customers
3. Response time of the API
4. Conversion rate
5. Consumption trend

When the KPIs were chosen, the next sub question needed to be answered. The sub question that was answered in the next phase of the research was "***How can we retrieve all necessary data?***". In this section, the three data sources that are necessary to measure the data are highlighted, which are SOARepo, the DevPortal and Grafana. Afterwards, the possible connections between the data sources and the Spotfire tool are listed. Eventually the outcome was that the SOARepo can have a direct connection to Spotfire. The data from the DevPortal was extracted with an Excel file and then loaded into Spotfire. Unfortunately, the data from the Elastic Search to Grafana is still being integrated and improved, which makes it impossible to gather the desired data from either Elastic Search or Grafana. This means that there will be no connection between Spotfire and Grafana/Elastic Search. To work around this problem, the data is processed within Grafana itself, making a dashboard on Grafana. There is a link on the Spotfire dashboard to the Grafana dashboard, to still keep the dashboard as complete and closely together as possible.

Afterwards, the following sub-question was worked on, "***How can we design a dashboard for the listed KPIs?"***. With all the data retrieved, the different visualizations are created to show the chosen KPIs mentioned above. When interested employees or partners want to look into the dashboard, the spotfire dashboard is most logical to start at, where also the link to the grafana dashboard is. Because there are two dashboards, a detailed overview is created to show all the different functionalities of the dashboards to make it understandable for all users.

From the evaluation, which is explained in section 6.2, the last sub question, "***Does the dashboard meet the expectation of the most crucial stakeholders?***", is answered. The dashboard was in first instance already sufficient, but some additions have been made to improve the dashboard further, such as improving a number of graphs. The dashboard has the correct KPIs visualized in it and the visualizations are clear. Also, it is relatively easy to use, even though not all graphs are in one location and multiple log-ins are required to gain access to all graphs.

After all the sub questions are answered, the main research question can be looked at, which is stated below.

What is an effective KPI data driven approach to measure the performance of API management?

Within this research, a complete process is created to create a KPI data driven approach to measure the performance of API management. The entire process is visualized in figure 12. This step-by-step procedure is carried out within AFKL to validate the created process, with the dashboard as a result. The process takes the following steps. To start off, a current situation of the company is described, where the list of the stakeholders are crucial. In the second step, the objectives of the research are described and discussed with the stakeholders, to ensure the researcher starts on the right foot. Next, the KPIs that are going to be measured are listed, combined with a decision on what KPIs are most crucial to the company. Afterwards, all requirements of the product need to be defined and considered by the stakeholders. In the fifth step, all data that is necessary is listed, found and connected to the dashboard. If all of the previous steps are carried out in the approved manner, all information can be applied to create the product. When the product is created, an evaluation is executed to find improvement options and these improvement options are applied before the last step, which is the deployment of the product.

Since the process has been validated within AFKL, it is assumed that the approach that is desired in the main research question, is the step by step process shown in figure 12. This process is generalized and can therefore be used in different companies or instances that desire to measure API performance. One can assume the generalizability of the generic process since there is no distinction between the type of company, the measurements one wants to make, nor are the KPIs already defined before the process is carried out. This general process has scientific value because of this generalizability.

7.1 LIMITATIONS

To begin with, not all data connections are direct data connections, which means that not all data is live and has to be updated manually. In practise this is the case for COMO and the developer portal. Also, the fact that there is no direct connection possible to COMO, causes that there are still two different dashboards, instead of one that covers all KPIs, which would be ideal.

What is more, is that not all KPIs that are desired are included in the dashboard, this circumstance is mainly caused due to scope and time limitations. More KPIs could give extra value by broadening the information given, which could make the dashboard more advantageous to a larger extent of teams within AFKL.

In addition to that, the dashboard is created with a fairly one sided perspective. The stakeholders are very important and based on their opinions and ideas, the KPIs and dashboard is formed. Obviously, there could be more perspectives added, such as a consumer perspective or the developer perspective.

Furthermore, it has to be considered if one would want to use this method in another company, there can be structural differences for different companies. This can be related to the branch, but also to the structure within the company. There are decisions made, such as the manners of retrieving data, which might be significantly different to other companies.

7.2 RECOMMENDATIONS

This dashboard does satisfy the needs of AFKL, but there are also some opportunities found to improve the dashboard and add value to AFKL and its stakeholders.

Now the current data is available, it might be useful to extend the research into the future. The data can be used to find new possibilities or points that could need attention. For instance, from the data that is already there, a forecast can be made on the expected new customers of API packages. If the prediction is that there will be too many new customers that the developers can manage at that moment in time, it could give a reason to address more staff to a specific department.

As mentioned above, not all the connections are direct at this moment, but this is something that can be improved at a later stage. If this process is executed, the dashboard would be complete at one location and also updated in real time. This would mean that the most recent and, some would consider the most useful data, would be available all the time, without having to take additional steps, which is the case with the current dashboard.

Moreover, more KPIs can be added which focus more on specific stakeholders. The dashboard that is created has different information and is based on the needs of the most prolific stakeholders. If these, or other stakeholders, want to have a more thorough analysis on an aspect that is only narrowly touched in this dashboard, additional KPIs can be created and added in specific tabs to keep the dashboard clear and easy to use.

Next to this, it is important to ensure employees from all teams that could benefit from the dashboard are getting the necessary information to use the dashboard. It is essential that all teams are informed of the location of the dashboard. To start off it is important to inventorize which teams might benefit from the dashboard. Ensuring all teams are informed, can be achieved by giving introductive (online) meetings with managers of the teams, where the managers can inform their own teams with the necessary information. When the teams are informed on the dashboard, it is important to enlighten them on the value the dashboard it may add and the possibilities it gives to their team and AFKL as a company.

7.3 FURTHER RESEARCH

First of all, it can be useful to research on more perspectives than the focused perspective of the stakeholders. If all valuable perspectives are known, it could give a number of perspectives that are worth getting more expertise on. The dashboard could be enlargened based on these new perspective and gain value to more different goals of AFKL. Examples of the perspectives could be the consumer and developer perspectives.

Another aspect that is recommended is to gain extra knowledge on the implementation of the dashboard throughout the different teams. Before this research was conducted, a known problem at AFKL is that a lot of data was scattered and that teams with similar interests each concucted their own research and created their own products, while it could have been more valuable and/or convenient to either work together or use each other's knowledge. If this dashboard is implemented correctly and the teams that could benefit from the product are all included and given access, this would be profitable for more employees than otherwise.

Lastly, in the future, it might be useful to examine the created process on a different kind of company, compared to AFKL. Conceivably, changes need to be made to the current process, depending on what kind of company the process is being carried out. The size of the company, location or industry could have an impact on the process.

8. REFERENCES

Abbas, A. (2019). *API Integration of National Complaint Handling System in Indonesia: a State of the Art Review.* ResearchGate. Retrieved from https://www.researchgate.net/publication/338943952_API_Integration_of_National_Complaint_Handling_System_in_Indonesia_A_State_of_The_Art_Review

Al-Hawari, F. B. (2021). A machine learning based help desk system for IT service management. *Journal of King Saud University*, 702-718. Retrieved from https://www.sciencedirect.com/science/article/pii/S1319157819300515

Bermbach, D., & Wittern, E. (2016). *Benchmarking Web API Quality.* Springer. doi:https://doi.org/10.1007/978-3-319-38791-8_11

Blanco, B. L. (2018). *Energy vs. QoX Network- and Cloud Services Management.* Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-90415-3_10

De, B. (2017). *API Management.* Bangalore: Apress. doi:https://doi.org/10.1007/978-1-4842-1305-6_2

Doerr, J. (2018). *Measure What Matters.* Bennett Group.

Emami, H. (2020). *Feature prioritization in SAFe model using COCOMO II.* Retrieved from https://www.hessamemami.com/wp-content/uploads/2020/02/FP.pdf

Fernandez, E., Fonoage, M., Vanhilst, M., & Marta, M. (2008). *The Secure Three-Tier Architecture Pattern.* Barcelona: IEEE. doi:10.1109/CISIS.2008.51

Fernandez, P., Gamez-Diaz, A., & Ruiz-Cortez, A. (2019). *Governify for APIs: SLA-driven ecosystem for API governance.* doi:https://doi.org/10.1145/3338906.3341176

Few, S. (2006). *Information Dashboard Design, The Effective Visual Communication of Data.* O'Reilly.

Giusti, R. M. (2019). *Synchromodal logistics: An overview of critical success factors, enabling technologies, and open research issues.* ScienceDirect. Retrieved from https://www.sciencedirect.com/science/article/pii/S1366554519303928

Ilieva, S. P.-A. (2018). *Towards a Technological Platform for Transparent and Flexible Assessment of Smart Cities.* ResearchGate. Retrieved from https://www.researchgate.net/publication/327911143_Towards_a_Technological_Platform_for_Transparent_and_Flexible_Assessment_of_Smart_Cities

Jin, B. Z. (2016). *A study of key success factors of service enterprises in China.* Springer. Retrieved from https://link.springer.com/article/10.1007/s10551-014-2074-2

Kin, L. (2020). *API governance with Postman.*

Kitchenham, B. C. (2007). *Guidelines for Performing Systematic Literature Reviews in Software Engineering.* Researchgate.

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software.* Retrieved from https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering?enrichId=rgreq-0f8ee6bb19aa0766af0f5dbc9e2385be-

XXX&enrichSource=Y292ZXJQYWdlOzMwMjkyNDcyNDtBUzoxMDgxMjY5MTY1MjY4OTkyQDE2MzQ

KLM. (n.d.). *KLM Informatie*. Retrieved from KLM: https://www.klm.nl/information/corporate/company-profile

Kopetenski, I. (2019). *Erlang as an Enablement Technology for Resilient General-Purpose Application on Edge IoT Networks.* Louvain-La-Neuve. Retrieved from https://www.info.ucl.ac.be/~pvr/AchlysCameraReadyFinal.pdf

Magouryk, C. (2021). *What is a Relational Database (RDBMS)?* Retrieved from https://www.oracle.com/in/database/what-is-a-relational-database/

Medjaoui, M. W. (2019). *Continuous API Management.* Sebastopol: O'Reilly.

Minton, H., & Schneider, F. (1980). Differential Psychology. *Wavelend Press*.

Orta, E. R. (2019). *Met4ITIL: A process management and simulation-based method for implementing ITIL.* Elsevier. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0920548917303057

Peffers, K. (2007). *A design science research methodology for information systems research.* ResearchGate. Retrieved from https://www.researchgate.net/publication/284503626_A_design_science_research_methodology_for_information_systems_research

Pillai, S. (2020). *How to Use KPIs to Measure the Business Value of APIs.* Gartner. Retrieved from https://www.gartner.com/en/documents/3980238

Ravasan, A. M. (2018). *An investigation into the critical success factors of implementing information technology service management frameworks.* IGI Global. Retrieved from https://www.researchgate.net/publication/326410089_An_Investigation_Into_the_Critical_Success_Factors_of_Implementing_Information_Technology_Service_Management_Frameworks

Ruiz, M. M. (2018). *Using simulation-based optimization in the context of IT service management process.* Elsevier. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0167923618301015

Seufert, M. W. (2015). *Demo: On the Monitoring of YouTube QoE in Cellular Networks from End-Devices.* ResearchGate. Retrieved from https://www.researchgate.net/publication/305513137_Demo_On_the_Monitoring_of_YouTube_QoE_in_Cellular_Networks_from_End-devices

Simon, J. (2020). Fundamentals of Data Architecture to Help Data Scientists Understand Architectural Diagrams Better. *Towards Data Science*. Retrieved from https://towardsdatascience.com/fundamentals-of-data-architecture-to-help-data-scientists-understand-architectural-diagrams-better-7bd26de41c66

Siriwardena, P. (2014). *Advanced API security.* Apress Berkeley, CA. doi:https://doi.org/10.1007/978-1-4842-2050-4

Sommerlad, P. (2003). *Reverse Proxy Patters.* Wollerau. Retrieved from https://hillside.net/europlop/HillsideEurope/Papers/EuroPLoP2003/2003_Sommerlad_ReverseProxyPatterns.pdf

Stanik, A. K. (2014). *SLA-driven Federated Cloud Networking: Quality of Service for cloud-based Software Defined Networks.* Elsevier. Retrieved from https://www.sciencedirect.com/science/article/pii/S187705091400948X

Tsolkas, D. L. (2016). *A survey on parametric QoE estimation for popular services.* Elsevier. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S1084804516302533

Vankatesh, V., & Morris, M. (2000). G. "Why Don't Men Ever Stop to Ask For Directions? Gender, Social Influence, and Their Role in Technology Acceptance and Usage Behavior. *MIS Quarterly*, 115-139.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*. doi:https://doi.org/10.2307/30036540

Zhang, Y. X. (2021). *Big data and human resource management research: An integrative review and new directions for future research.* Elsevier. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0148296321002563

Zuev, D. K. (2018). Machine learning in IT service management. *Procedia Computer Science*, 675-679. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050918323494

9. APPENDICES

All appendices are listed below.

9.1 A. LITERATURE REVIEW ON API MANAGEMENT

9.1.1 Table XX: Filtering articles

Search filtering	Number of articles
Articles identified by searching query	23
Excluded based on exclusion criteria	-7
Number of articles still obtainable	16
Screened on abstract	16
Excluded based on abstract	-9
Articles still obtainable	7
Articles screened on full text	7
Articles excluded after reading full text	0
Articles added after reading full text	-
Valuable articles	7

9.1.2 Table XX: Summary of the papers selected

Title	Description
API Integration of National Complaint Handling System in Indonesia: A State of The Art Review (Abbas, 2019)	In this article, a review is done for API integration in Indonesia. They are integrating an API for a complaint handling system and research is done on the complaints within the integration of the API.
Note on article Abbas 2019: The focus of this article lays on one API only, whereas in AFKL the focus will be on many APIs. Also in the generation of the pattern, based on the historical data, duplicates were found. This could have a negative impact on the validity of the data.	
Erlang as an Enabling Technology for Resilient General-Purpose Applications on Edge IoT Networks (Kopestenski, 2019)	This paper gives insight on how an API is created to generate data for testing of edge computing, which is seen as a key success factor future internet solutions
Note on article Kopestenski 2019: Because the improvement of this research is not completed yet, the API works on an unreliable network. This is not a major problem, since they are still developing the system.	
Towards a Technological Platform for Transparent and Flexible Assessment of Smart Cities (Ilieva, 2018)	In this article, open APIs are used to improve data analysis. Within the APIs, three types of indicators are used, namely numbers, rate and value on a scale.
Note on article Ilieva 2018: The paper uses the city Sofia as a pilot city. Because this is the pilot city, the validation has not been done, which can give doubts to the outcome of the paper.	
Demo: On the Monitoring of YouTube QoE in Cellular Networks from End-devices (Suefert, 2015)	This article shows a demo on an API that measured several KPIs to learn more about the performance of YouTube.
Note on article Suefert, 2015: A video streaming service is very different to an air carrier, such as AFKL. Therefore, some KPIs that are used in this article are redundant for our studies. An example for this is the video quality level.	

<table>
<tr><td colspan="2">This KPI can still be transformed into a useful KPI for AFKL, but it will need to change its dynamics to be suitable.</td></tr>
<tr><td>SLA-driven Federated Cloud Networking: Quality of Service for cloud-based Software Defined Networks
(Stanik, 2014)</td><td>This paper proposes an architectural approach for a generic layered model and API based software to orchestrate and federate heterogeneous networks. In this article, various KPIs are measured to assure the service level agreement</td></tr>
<tr><td colspan="2">Note on article Stanik 2014:
This paper shows a system that works with the “get what you pay” principle. Even though AFKL uses open and closed APIs, which you could define as a “get what you pay” principle, it is not AFKL’s aim to make differences between customers. Furthermore, the validation and demonstration step is not carried out in this paper. Thus, the usefulness of this paper might be lower than expectation, because it might only work theoretically and not practically or their conception of the architecture might be different than reality.</td></tr>
<tr><td>Synchromodal logistics: An overview of critical success factors, enabling technologies, and open research issues
(Giusti, 2019)</td><td>In this article, it is discussed what critical success factors are on integrating new technologies in business. Projects are discussed on how APIs can help generating the correct data and give sufficient insight on the critical success factors</td></tr>
<tr><td colspan="2">Note on article Giusti 2019:
Because synchromodal logistics is still being researched extensively, it is still complex to implement it in a system for APIs.</td></tr>
<tr><td>Big data and human resource management research: An integrative review and new directions for future research
(Zhang, 2021)</td><td>This article gives brief insight on how KPIs are created and used from large data sets generated by APIs or other data collecting methods</td></tr>
<tr><td colspan="2">Note on article Zhang 2021:
This article works with data collection within one data storage system. Within AFKL, data is scattered, for instance in SOARepo and common monitoring (COMO).</td></tr>
</table>

	Explanation API management	Uptime	CPU usage	Error rate	Average Latency	API usage growth	Unique API customers	Top customers per API
Abbas, 2019	✓							
Kopestenski 2019,	✓		✓					
Ilieva, 2018		✓		✓		✓		
Suefert, 2015				✓		✓	✓	
Stanik, 2014		✓		✓				✓
Giusti, 2019			✓	✓				
Zhang, 2021					✓	✓		

9.2 B. LITERATURE REVIEW ON SERVICE MANAGEMENT

9.2.1 Table XX: Filtering articles

Search filtering	Number of articles
Articles identified by searching query	29
Excluded based on exclusion criteria	-11
Number of articles still obtainable	18
Screened on abstract	18
Excluded based on abstract	-5
Articles still obtainable	13
Articles screened on full text	13
Articles excluded after reading full text	-5
Articles added after reading full text	-
Valuable articles	8

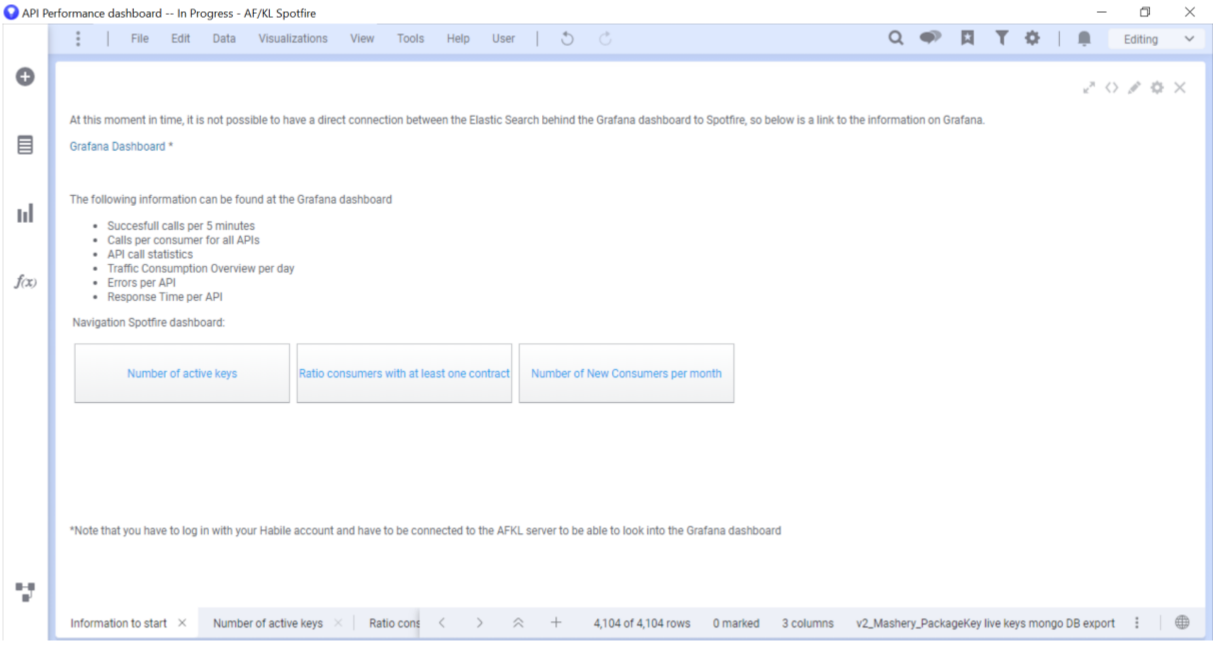
9.2.2 Table XX: Summary of the papers selected

<table>
<tr><td>Title</td><td>Description</td></tr>
<tr><td>A study of key success factors of service enterprises in China
(Jin, 2014)</td><td>In this paper, an explanation is given on the theory of how information was analyzed. It is focused on the enterprises customer-oriented service, service, service management service innovation and corporate social responsibility. The KSFs are customer satisfaction, perceived service quality and enterprise reputation.</td></tr>
<tr><td colspan="2">Note on article Jin 2014:
One aspect that is considered in this article is the corporate social responsibility (CSR) of China. If AFKL wants to use the key success factors of this article, it has to take into account that the CSR is different in China compared to the CSR within the AFKL staff, because of cultural differences.</td></tr>
<tr><td>A survey on parametric QoE estimation for popular services
(Tsolkas, 2016)</td><td>This article describes what changes occur in moving to a 5G era. One of those changes is the change from Quality of Service to Quality of Experience. It also mentions numerous important KPIs and major configuration parameters in the most popular service types.</td></tr>
<tr><td colspan="2">Note on article Tsolkas 2016:
Next to KPIs that are easily adapted to the AFKL structure, such as number of stalling events, are there multiple KPIs in this study that cannot be adapted to the AFKL landscape, such as the send loudness rating, which represents the loudness rate at the sender of the radio.</td></tr>
<tr><td>Energy vs. QoX Network- and Cloud Services Management
(Blanco, 2018)</td><td>This paper gives an overview of commonly used frameworks of multi-criteria optimization and are evaluated on different quality of service scenarios. Lastly, related networks tools have been described. This is done to identify possible application areas</td></tr>
<tr><td colspan="2">Note on article Blanco 2018
In the future research section it is stated that the formulation of the design should be altered to fit the multi-criteria optimization algorithm. This indicates that the work that is carried out is incomplete and the results might be influenced by this lacking components.</td></tr>
<tr><td>Using simulation-based optimization in the context of IT service management process</td><td>In this paper, an approach that uses simulation based multi-objective optimization is described</td></tr>
</table>

(Ruiz, 2018)	to achieve process efficiency, which is their critical success factor. This is done by simultaneously measuring multiple KPIs to achieve process efficiency.
Note on article Ruiz 2018: Very extensive simulation, with multiple experiments. The KPIs based on completed actions and usage numbers are easily adapted to the AFKL KPIs.	
Met4ITIL: A process management and simulation-based method for implementing ITIL (Orta, 2019)	This article reviews a method on the implementation of Information Technology Infrastructure Library (ITIL). The method is implemented in a Spanish company and reviewed by the stakeholders of the process, using a questionnaire. The questionnaire shows the usefulness and the CSFs.
Note on article Orta 2019: A questionnaire is a useful tool to gain information, but there are no insights on the number of people who took the questionnaire. A lack of insight on that, can lead to a reduced validity. For instance, if there are only two people who took the questionnaire, is this still a good representation of the company? This is at least doubtful.	
A machine learning based help desk system for IT service management (Al-Hawari, 2021)	This paper gives insight on the importance of an IT help desk system of companies that rely on IT services and resources. One aspect that is reviewed are the numerous KPIs that allow evaluation on the overall IT department performance.
Note on article Al-Hawari 2021: One of the main KPIs that is discussed in this article is customer satisfaction. The customer satisfaction level for the APIs of AFKL is not measurable yet, because of the configuration of a new system.	
Machine learning in IT service management (Zuev, 2018)	This paper describes how a dataset of a fast food chain is used to develop a predictive model that can improve the quality of the IT service. The main metrics that are focused on are the harmonic mean of the precision and recall rate.
Note on article Zuev 2018: The study of Zuev focuses on finding predictive methods to reduce the number of incidents of their IT service management. This could also be a possibility for AFKL when the dashboard is fully functional.	
An investigation into the critical success factors of implementing information technology service management frameworks (Ravasan, 2018)	In this article, critical success factors are identified that impact a successful implementation of the information technology service management frameworks. 22 critical success factors have been identified after an extensive research using questionnaires.
Note on article Ravasan 2018: This research uses a mixed method of a literature study and a questionnaire with 122 valid attendees. The 22 CSF that are found seem a bit too extensive to ensure all are necessary for a clear dashboard.	

	Explanation API management	Uptime	CPU usage	Error rate	Average Latency	API usage growth	Unique API customers	Top customers per API
Jin, 2014	✓						✓	
Tsolkas, 2016		✓		✓		✓		
Blanco, 2018	✓		✓					
Ruiz, 2018					✓	✓	✓	
Orta, 2019	✓			✓				
Al-Hawari, 2021		✓		✓	✓			
Zuev, 2018			✓	✓				
Ravasan, 2018		✓	✓	✓	✓	✓	✓	✓

9.3 C. FULL DASHBOARD



At this moment in time, it is not possible to have a direct connection between the Elastic Search behind the Grafana dashboard to Spotfire, so below is a link to the information on Grafana.

Grafana Dashboard *

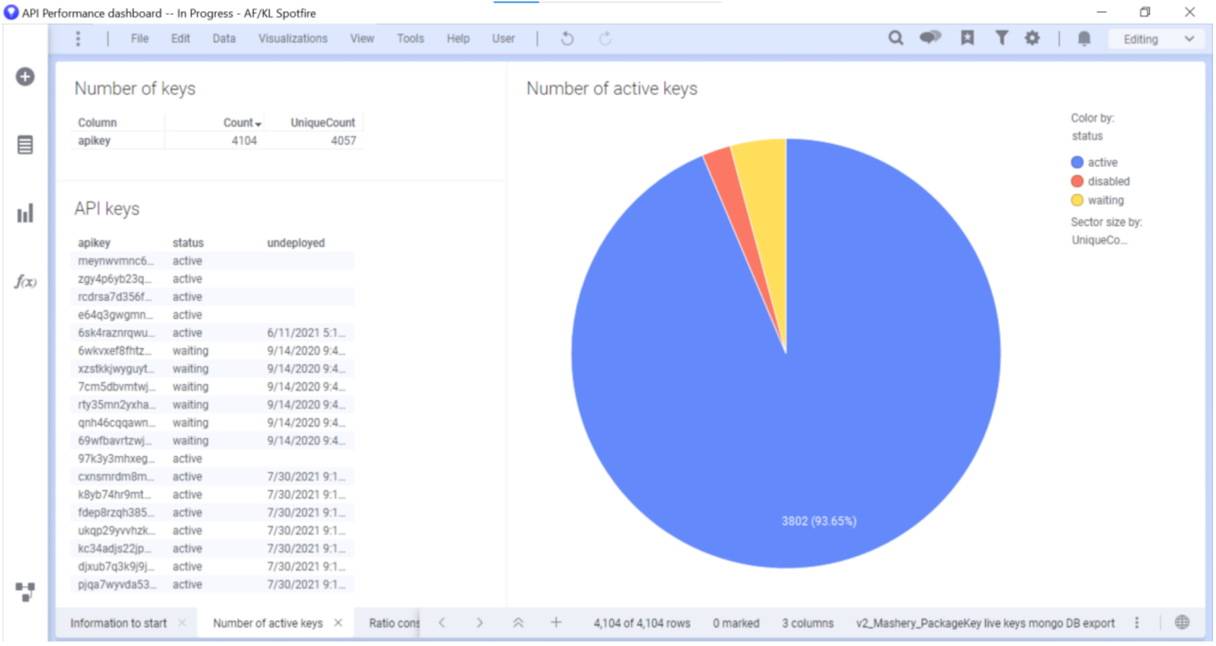
The following information can be found at the Grafana dashboard

- Succesfull calls per 5 minutes
- Calls per consumer for all APIs
- API call statistics
- Traffic Consumption Overview per day
- Errors per API
- Response Time per API

Navigation Spotfire dashboard:

Number of active keys | Ratio consumers with at least one contract | Number of New Consumers per month

*Note that you have to log in with your Habile account and have to be connected to the AFKL server to be able to look into the Grafana dashboard

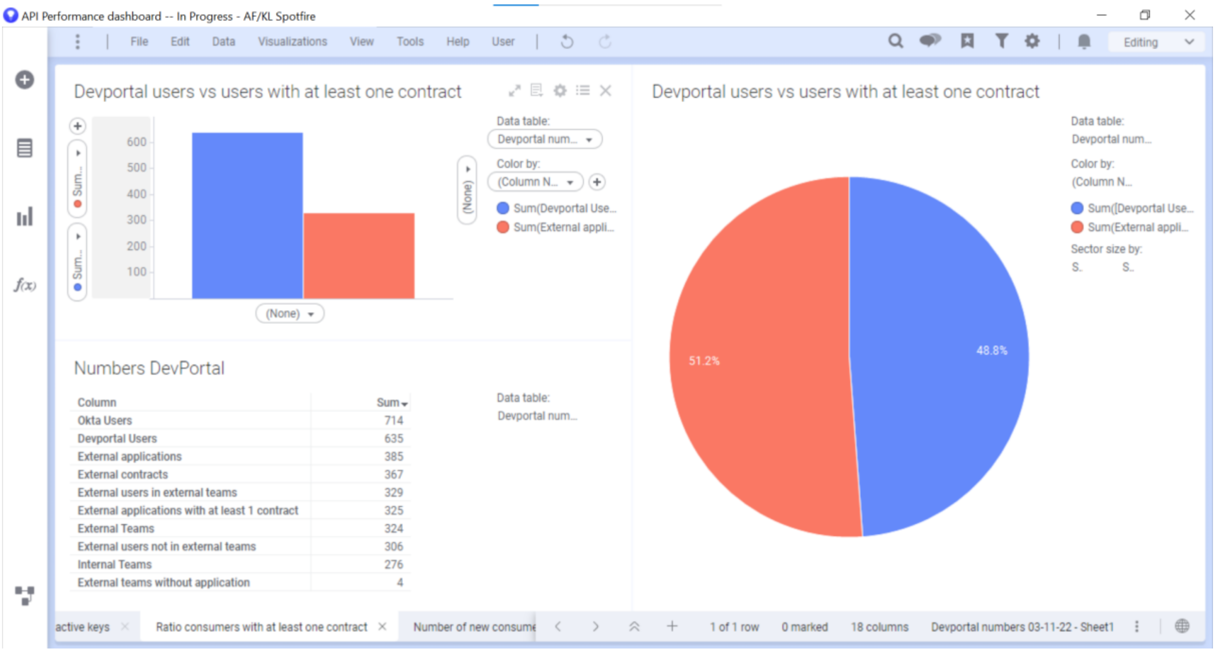


Number of keys

Column	Count	UniqueCount
apikey	4104	4057

API keys

apikey	status	undeployed
meynwvmnc6...	active	
zgy4p6yb23q...	active	
rcdrsa7d356f...	active	
e64q3gwgmn...	active	
6sk4raznrqwu...	active	6/11/2021 5:1...
6wkvxef8fhtz...	waiting	9/14/2020 9:4...
xzstkkjwyguyt...	waiting	9/14/2020 9:4...
7cm5dbvmtwj...	waiting	9/14/2020 9:4...
rty35mn2yxha...	waiting	9/14/2020 9:4...
qnh46cqqawn...	waiting	9/14/2020 9:4...
69wfbavrtzwj...	waiting	9/14/2020 9:4...
97k3y3mhxeg...	active	
cxnsmrdm8m...	active	7/30/2021 9:1...
k8yb74hr9mt...	active	7/30/2021 9:1...
fdep8rzqh385...	active	7/30/2021 9:1...
ukqp29yvvhzk...	active	7/30/2021 9:1...
kc34adjs22jp...	active	7/30/2021 9:1...
djxub7q3k9j9j...	active	7/30/2021 9:1...
pjqa7wyvda53...	active	7/30/2021 9:1...



Devportal users vs users with at least one contract

Data table: Devportal num...

Color by: (Column N...

● Sum(Devportal Use...
● Sum(External appli...

Devportal users vs users with at least one contract

Data table: Devportal num...

Color by: (Column N...

● Sum([Devportal Use...
● Sum(External appli...

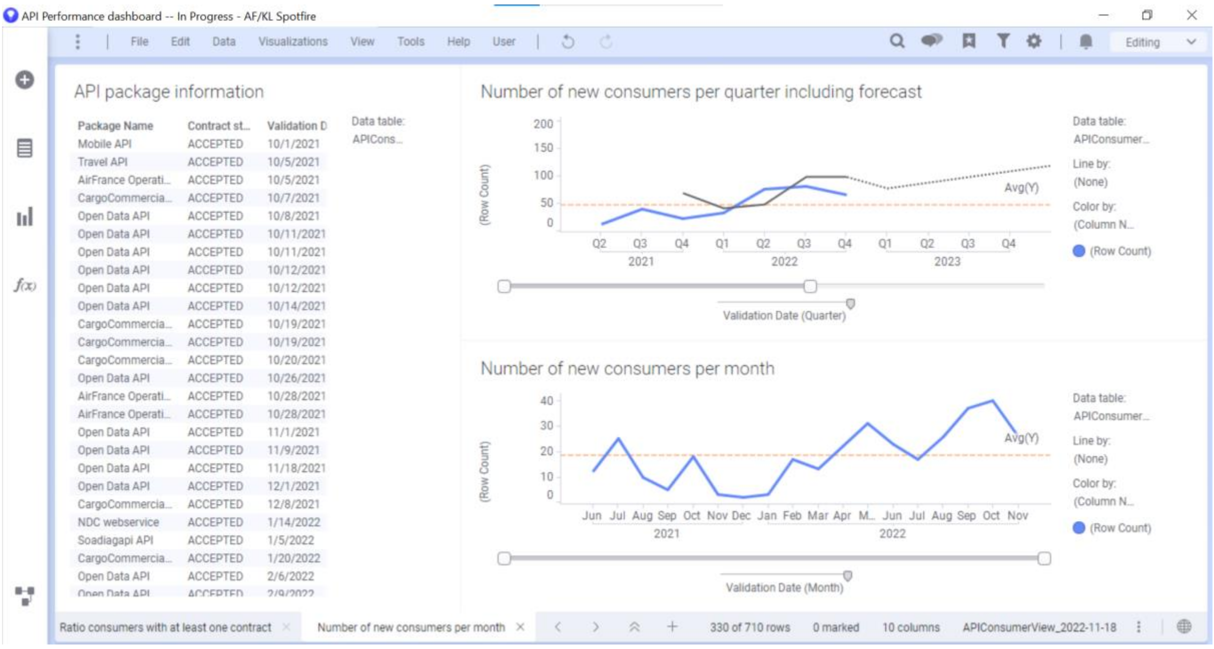
Sector size by: S... S...

Numbers DevPortal

Column	Sum
Okta Users	714
Devportal Users	635
External applications	385
External contracts	367
External users in external teams	329
External applications with at least 1 contract	325
External Teams	324
External users not in external teams	306
Internal Teams	276
External teams without application	4

Data table: Devportal num...

1 of 1 row 0 marked 18 columns Devportal numbers 03-11-22 - Sheet1



API package information

Package Name	Contract st...	Validation D
Mobile API	ACCEPTED	10/1/2021
Travel API	ACCEPTED	10/5/2021
AirFrance Operati...	ACCEPTED	10/5/2021
CargoCommercia...	ACCEPTED	10/7/2021
Open Data API	ACCEPTED	10/8/2021
Open Data API	ACCEPTED	10/11/2021
Open Data API	ACCEPTED	10/11/2021
Open Data API	ACCEPTED	10/12/2021
Open Data API	ACCEPTED	10/12/2021
Open Data API	ACCEPTED	10/14/2021
CargoCommercia...	ACCEPTED	10/19/2021
CargoCommercia...	ACCEPTED	10/19/2021
CargoCommercia...	ACCEPTED	10/20/2021
Open Data API	ACCEPTED	10/26/2021
AirFrance Operati...	ACCEPTED	10/28/2021
AirFrance Operati...	ACCEPTED	10/28/2021
Open Data API	ACCEPTED	11/1/2021
Open Data API	ACCEPTED	11/9/2021
Open Data API	ACCEPTED	11/18/2021
Open Data API	ACCEPTED	12/1/2021
CargoCommercia...	ACCEPTED	12/8/2021
NDC webservice	ACCEPTED	1/14/2022
Soadiagapi API	ACCEPTED	1/5/2022
CargoCommercia...	ACCEPTED	1/20/2022
Open Data API	ACCEPTED	2/6/2022
Open Data API	ACCEPTED	2/9/2022

Data table: APICons...

Number of new consumers per quarter including forecast

Data table: APIConsumer...

Line by: (None)

Color by: (Column N...

● (Row Count)

Number of new consumers per month

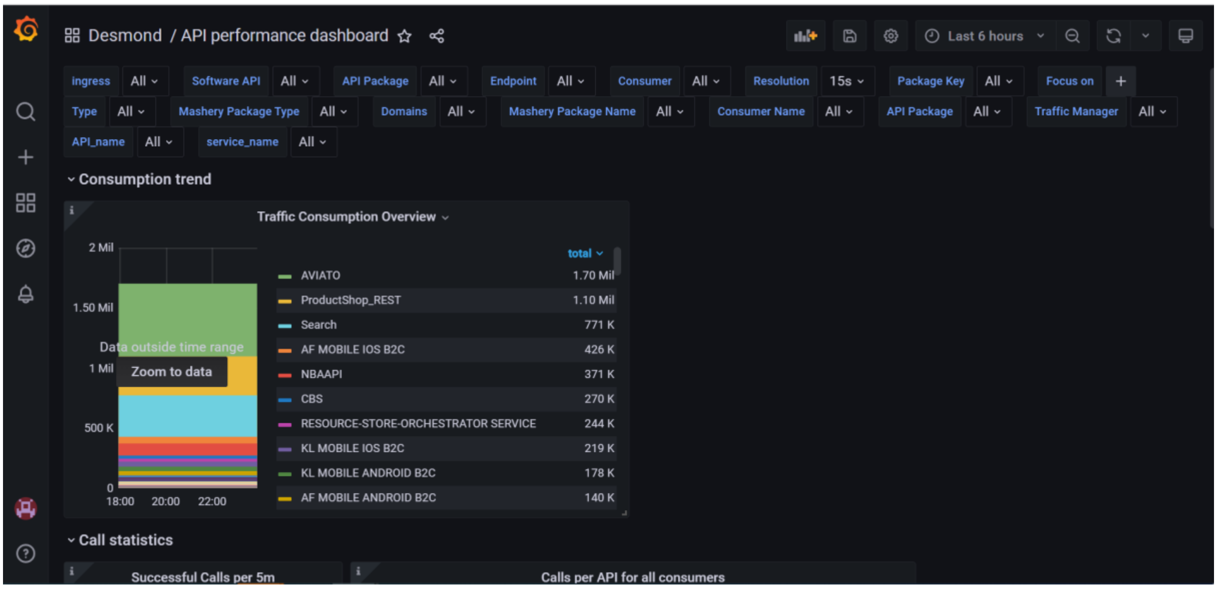
Data table: APIConsumer...

Line by: (None)

Color by: (Column N...

● (Row Count)

330 of 710 rows 0 marked 10 columns APIConsumerView_2022-11-18



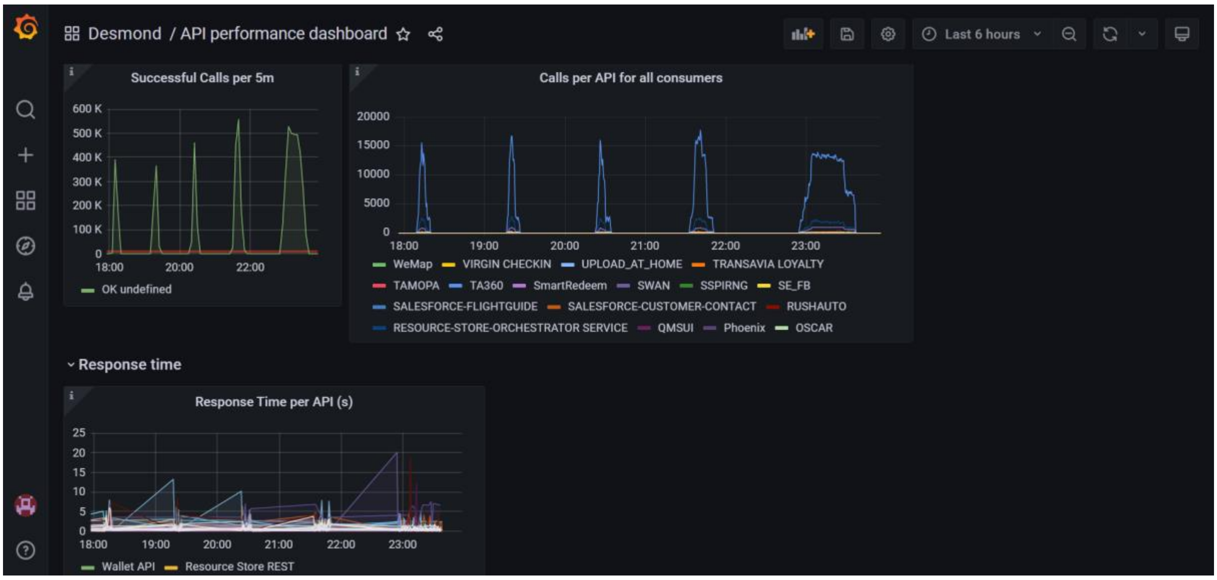
ingress All · Software API All · API Package All · Endpoint All · Consumer All · Resolution 15s · Package Key All · Focus on
Type All · Mashery Package Type All · Domains All · Mashery Package Name All · Consumer Name All · API Package All · Traffic Manager All
API_name All · service_name All

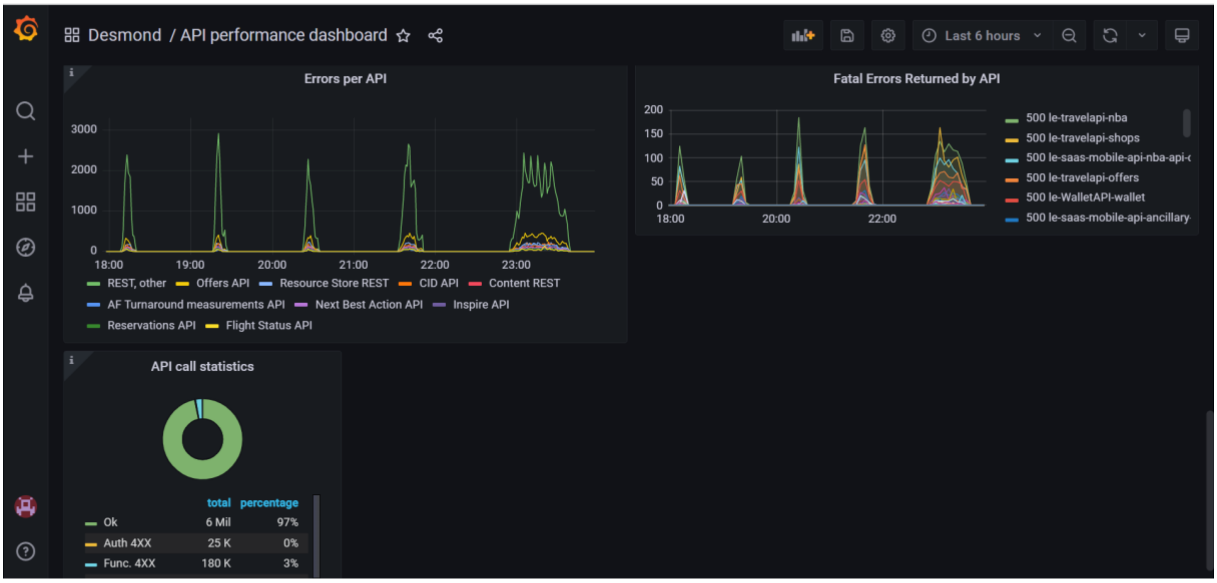
Consumption trend

Traffic Consumption Overview

	total
AVIATO	1.70 Mil
ProductShop_REST	1.10 Mil
Search	771 K
AF MOBILE IOS B2C	426 K
NBAAPI	371 K
CBS	270 K
RESOURCE-STORE-ORCHESTRATOR SERVICE	244 K
KL MOBILE IOS B2C	219 K
KL MOBILE ANDROID B2C	178 K
AF MOBILE ANDROID B2C	140 K

Call statistics





	total	percentage
Ok	6 Mil	97%
Auth 4XX	25 K	0%
Func. 4XX	180 K	3%

9.4 D. UTAUT MODEL QUESTIONNAIRE

PERFORMANCE EXPECTANCY

Using the dashboard increases my productivity					
1	2	3	4	5	No opinion

Using the dashboard can help AFKL					
1	2	3	4	5	No opinion

EFFORT EXPECTANCY

The dashboard is clear					
1	2	3	4	5	No opinion

The visualizations of the graphs are useful					
1	2	3	4	5	No opinion

The most important KPIs are visualized					
1	2	3	4	5	No opinion

Mastering the dashboard will be easy for me					
1	2	3	4	5	No opinion

Utilizing the dashboard is easy for me					
1	2	3	4	5	No opinion

SOCIAL INFLUENCE

People who influence my behaviour at work want me to use the dashboard					
1	2	3	4	5	No opinion

This dashboard is desired by my manager(s)					
1	2	3	4	5	No opinion

The dashboard suits me and my employees					
1	2	3	4	5	No opinion

FACILITATING CONDITIONS

I have all necessities to use the dashboard					
1	2	3	4	5	No opinion

I have the knowledge to use the dashboard					
1	2	3	4	5	No opinion

There are people available when there is a problem with the dashboard					
1	2	3	4	5	No opinion

9.5 E. EVALUATION SCORES

PERFORMANCE EXPECTANCY

Using the dashboard increases my productivity					
1	2	3	4	5	No opinion
		3	1		2

Using the dashboard can help AFKL					
1	2	3	4	5	No opinion
			3	3	

EFFORT EXPECTANCY

The dashboard is clear					
1	2	3	4	5	No opinion
	1		2	3	

The visualization of the graphs is useful					
1	2	3	4	5	No opinion
			5	1	

The most important KPIs are visualized					
1	2	3	4	5	No opinion
			4	2	

Mastering the dashboard will be easy for me					
1	2	3	4	5	No opinion
			3	3	

Utilizing the dashboard is easy for me					
1	2	3	4	5	No opinion
		1	1	4	

SOCIAL INFLUENCE

People who influence my behavior at work want me to use the dashboard					
1	2	3	4	5	No opinion
2			1		3

This dashboard is desired by my manager(s)					
1	2	3	4	5	No opinion
1			1	1	3

The dashboard suits me and my employees					
1	2	3	4	5	No opinion
		1	3		2

FACILITATING CONDITIONS

I have all necessities to use the dashboard					
1	2	3	4	5	No opinion
			5	1	

I have the knowledge to use the dashboard					
1	2	3	4	5	No opinion
			3	3	

There are people available when there is a problem with the dashboard					
1	2	3	4	5	No opinion
		2	1	1	2

9.6 F. DETAILS EVALUATION SCORES

AVG Score	Number of scoring votes	Total votes	% of scoring votes
Using the dashboard increases my productivity			
3.25	4	6	67%
Using the dashboard can help AFKL			
4.5	6	6	100%
The dashboard is clear			
4.166666667	6	6	100%
The visualization of the graphs is useful			
4.166666667	6	6	100%
The most important KPIs are visualized			
4.333333333	6	6	100%
Mastering the dashboard will be easy for me			
4.5	6	6	100%
Utilizing the dashboard is easy for me			
4.5	6	6	100%
People who influence my behavior at work want me to use the dashboard			
2	3	6	50%
This dashboard is desired by my manager(s)			
3.333333333	3	6	50%
The dashboard suits me and my employees			
3.75	4	6	67%
I have all necessities to use the dashboard			
4.166666667	6	6	100%
I have the knowledge to use the dashboard			
4.5	6	6	100%
There are people available when there is a problem with the dashboard			
3.75	4	6	67%

UNIVERSITY OF TWENTE
Drienerlolaan 5
7522 NB Enschede

P.O.Box 217
7500 AE Enschede

P +31 (0)53 489 9111

info@utwente.nl
www.utwente.nl