Mikko Salenius

Designing A Data Solution For A Game Company

Case InfluxDB and Grafana

Tietojenkäsittely
Tradenomi

Autumn 2017

KAJAANIN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

TIIVISTELMÄ

Tekijä(t): Salenius Mikko

Työn nimi: Designing A Data Solution For A Game Company – Case InfluxDB and Grafana

Tutkintonimike: Tradenomi (AMK), tietojenkäsittely

Asiasanat: data, Big data, datatiede, analyysimenetelmät

Opinnäytetyön tarkoituksena oli tutkia, miten data analytiikkaa voidaan käyttää hyväksi peliyhtiössä ja mitä pitää ottaa huomioon sen tarvittavia ohjelmistoja valitessa niiden yhteensopivuuden ja käytännöllisyyden takaamiseksi yhtiön tarpeita ajatellen. Työ antaa kokonaisvaltaisen kuvan siitä miten peliyhtiön olisi hyvä aloittaa suunnitellessaan analytiikan tuomista osaksi yhtiötään ja mitkä ovat ensimmäiset askeleet tavoitteen saavuttamiseksi. Tekninen työ tehtiin niin, että yhtiö voi aloittaa Influxin ja Grafanan yhdistelmän käytön nopeasti analytiikassaan.

Teoria osuus tehtiin kaikki tarvittavat askeleet sisältäväksi analytiikka suunnitelman suunnittelua varten alusta loppuun peliyhtiön perspektiivistä. Askeleet tehtiin järjestyksessä jossa lukija ensin saa hyvät perusteet peli analytiikassa ja ohjataan sitten rakentamaan ratkaisu käytännössä. Työ perustuu tutkimustyöhön useista lähteistä ja tekijän omiin kokemuksiin työskennellessään junior data analyytikkona.

Viimeiseksi tulokset esitetään yhdistäen kaikki yhteenvedoksi siitä mitä työllä saavutettiin. Opinnäytetyö antaa perusteet jonka päälle rakentaa kenelle tahansa alalla työskentelevälle, kun henkilö haluaa oppia data analytiikasta pelialan kontekstissa. Työ suunniteltiin puhtaasti katkeavaksi teoria ja käytännön osuuksiin, jotta riippumatta tiimistä ja teknisestä osaamisesta lukija voi saada opinnäytetyöstä jotain irti.

Jokaisen riippumatta tiimistä pitäisi ymmärtää datan potentiaali, sillä tämä tuo sekoitukseen useita perspektiivejä ja tämä voimistaa potentiaalisia datan analysoinnin tuloksia, kun kaikki on otettu huomioon, sillä eri tiimeillä on kuitenkin erilaisia tarpeita informaatiolle. Päätösten teko ei kuitenkin saisi ikinä perustua arvioihin ja arvailuun yhtiössä. Tämä on ongelma jonka data analytiikka auttaa ratkaisemaan täysin tehdessä sen oikein, koska numerot eivät koskaan valehtele, ellei niitä lasketa ja kerätä puutteellisesti.

ABSTRACT

Author(s): Mikko Salenius

Title of the Publication: Designing A Data Solution For A Game Company – Case InfluxDB and Grafana

Degree title: Bachelor of Business Administration, Business information Technology

Keywords: data, Big data, data science, analytical methods

The objective of this Bachelor's thesis was to research, how data analytics can be used for benefit in a game company and what should be taken in account when choosing necessary software's to guarantee their compatibility and practicality regarding the needs of the company. Work gives an overall view of what game company should start with when planning to introduce analytics to their company and what would be the first steps towards that goal. Technical work is done so that company can start using Grafana and Influx combination in their analytics fast.

Theory part was done to include all the steps required to plan analytics plan from start to finish from the perspective of a game company. Steps were made in such order that the reader first gets a solid foundation into the theory of game analytics and is then guided in building the solution in practice. Work is based on research from various sources and my own experiences working as a junior data analyst.

Finally, results are presented combining everything together to summarize what was achieved with the work. Thesis gives anyone in the industry foundation to build on when desiring to learn about data analysis in the context of game industry. Work was designed to be split cleanly to the theory and technical parts so that regardless of the team and technical skills, the reader can get something out of the thesis.

Data's potential should be understood by everyone regardless of teams, as this brings multiple perspectives into the mix and thus boosts the results data analysis can have when everything is accounted for, different teams have different kinds of needs for information after all. Decisions should after all never be based on assumptions and guessing in a company. This is a problem that data analysis helps to solve completely when done right, as numbers never lie, unless they are calculated and gathered inadequately.

TABLE OF CONTENTS

LIST OF SYMBOLS

InfluxDB = Influx Database software made by Influxdata

Grafana = Visualization software made by Grafana Labs

TSDB/Time Series Database = Database optimized for handling time series data, which in short is data that is indexed by time.

Timestamp = metadata that contains information on when the data was recorded

Metadata = data describing data’s basic information.

KPI = Key Performance Indicator

DAU = Daily Active Users

CCU = Current Concurrent Users

Retention = a time period where on X day look how many players do a specific action and after Y time period look how many of them are still doing it. E.g. 100 players start on Monday, how many of them still log in on next week’s Monday?

1 INTRODUCTION

This thesis is made in collaboration with Critical Force Ltd. Where the author is working as a Junior Data Analyst. The software’s used in the thesis are used in the company as main tools to analyse and visualize data. The purpose of this thesis is to go through how this system is built in the first place and to give insight to readers how data is used in the gaming industry to boost the revenue and game quality.

Big data has become one of the highly discussed topics in recent years due to its seemingly limitless potential in almost any industry. Data as information is also something that is growing exponentially due to the increasing tracking of everything we do as consumers to guide the company growth. Big data answers questions like what do the consumers buy, why they buy it, when they buy it, how often they buy it and what are the environmental variables which drive consumers to buy. To understand all these information data analysts and their software’s are needed.

Influx Database or InfluxDB for short is an open source database meant to be used as a Time-Series database. It’s the most popular TSDB according to db-engines.com ranking and is thought to be the market leader in Time Series category.

Grafana is an open source visualization software, that supports many different databases and gives the tools to represent the data in the databases in a visually pleasing and easy to understand way. Its main features consist of easy to use UI for making various graph types and flexibility due to its open source and focus on being the greatest visualization tool for time series data.

2 BIG DATA IN THE GAMING INDUSTRY

Big data has been making waves in IT industry for years now, it’s one of the or maybe even the most searched concept right now in this field. This is most likely due to how fast we’re moving with the size of data. In a Forbes article, Gil Press (Press, 2017) wrote:

> *“Data Monetization” will become a major source of revenues, as the world will create 180 zettabytes of data (or 180 trillion gigabytes) in 2025, up from less than 10 zettabytes in 2015, according to IDC.*

In this text, IDC stands for International Data Corporation, a child company owned by organization International Data Group. According to IDC revenues worldwide in big data and business analytics will increase dramatically from $130.1 billion in 2016 to more than $203 billion in 2020. (International Data Corporation, 2016).

All of this gives a good peek into how big of a deal Big data really is. Next, it’s time to dive into what it actually is, what makes it so valuable and how one can make use of it in a game company.

2.1 What is Big Data

Big Data is often described as a large quantity of data, but this isn’t the full definition of it. Big Data is data that has 3 defining V’s usually, but there is also a version with 5 V’s. The three most spoken V’s of Big Data are:

- Volume = How much data are we storing and what kind of data are we storing
- Velocity = How fast are we gathering the data, is it real-time data or near live time?

- Variety = Is the data structured, unstructured, semistructured or all of them?

This is the 3 V’s model. The 5 V’s model adds the following:

- Veracity = How trustworthy is the data we are getting? If the data is inaccurate, is there enough volume of it to compensate that?
- Value = How valuable is that data to business, what kind of information can be mined from it and how will it affect decision-making process? This is the most important V when designing a data strategy since all data has to have value to the company, otherwise, it’s just extra expenses for no real benefit.

Which model should then be used to determine the situation? 3 is enough to draw conclusions on whether one is dealing with big data yet, but when designing company’s own data strategy, all 5 V’s of Big Data should be accounted for. (Gerasimou, 2016) (Marr, Big Data: The 5 Vs Everyone Must know, 2014)

2.1.1 Why is it so important?

Companies have always gathered data about their clients to boost their revenue by increasing their understanding of their clientele. Why do customers buy our products? Do they return after one purchase? Are there similarities between clients? What conditions affect their purchase behaviour? These are all just a fraction of all the information data can give business owners, if done right. We live in an age, where it’s cheaper and cheaper to buy data storing capacity. Since 2009 the average cost per GB has been plummeting only to increase during 2012 Thailand floods crippled some of the biggest factories producing hard drives. In 2009 the price was 0.11$ per gigabyte, whereas in 2017 it’s close to 0.2$. This means price drop of close to 75% in just 8 years. This translates to businesses being able to increase their data storage about their customers tremendously in addition to the constant increase in techniques and methods in the industry. (Klein, 2017)

Many major companies have switched their tactic from tracking just data that they need to acquiring everything they are able to. This way of data analysis relies on storing the data as small fragments as possible for optimal sized storing. This data is then pulled only when needed to do analysis on. This method is known as data warehousing. It’s, however, an advanced method and doesn’t necessarily suit a starting company. For starting companies’ traditional relational databases are recommended, since it’s advised to first focus on the key KPI’s as later discussed. (Reddy, 2017)

How far can then data be used to analyse customer behaviour? A famous, or to some infamous example of this happened when Target, one of USA’s largest retailers predicted pregnancy of a girl in Highschool before her father knew about it. This was done by analysing what products customers bought and giving the customer “pregnancy score” if she matched certain criteria with purchases. Once the score was high enough Target would send vouchers for baby supplies appointed straight to the pregnant person. In this case, they were of course appointed to the daughter, sparking outrage from her father. In the end, however, data was right. (Hill, 2012)

This is a prime example of the power lying inside data. The case also showcases one of the two different situations with data. The first one is a situation where the company has a small amount of data, but it’s very accurate due to how detailed the process of gathering it is. The second situation and the one this case is related to is a situation where there is something potentially related happening, but the data is too inaccurate for it. The solution is to have so much of that data, that it replaces that need for detail while still giving just as trustworthy results. Of course, this doesn’t apply to every data measurement so treading carefully is advised and always redo the calculations when in doubt.

2.1.2 How is it used in game companies?

So far, we’ve talked about how in general data is used in businesses, but the bigger question is how it’s used in game companies. Game companies these days don’t make strictly products, but products as a service. What does this mean then? A product is something that is built by a company and released as a complete package most likely never again to be altered, old console games, for example, are an example of this. Games these days, however, are patched constantly, new items are added, dlc’s introduced and microtransactions have become a norm especially in mobile games. This is all so that companies can lengthen the life of a game keeping players playing that game for as long as possible. This has its benefits to players, but it’s also caused a lot of heated discussion due to differing approaches to monetization by companies. Some companies offer a game as a complete package just charging more if a player wants to buy additional content to the game whereas others use microtransactions or even both in their games. (Schreier, 2017)

All of this affects the monetary aspect of data analysis and it could be said to be the most important focus point for a game company for obvious reasons. This is combined with KPI’s portraying user statistics like Daily-, Weekly- and Monthly Active Users (DAU, WAU, MAU). Retention is for many companies the most important KPI to measure. It’s a formula of how many users begin a selected action and out of them how many remain after X amount of days. This gives a good picture of how well users are kept engaged in the game. A very basic retention example would be that out of 1000 users starting on day Y still return to the game after 30 days. Looking at this ratio through varying time periods gives a good insight on how long players think the game is worth coming back to. There are various KPI’s and various ways of calculating them, so it must be made sure, that everyone is on the same page on what methods are used for calculations inside the company. (Lovato, 2015)

These, however, are just the tip of the iceberg. Data analytics can be used to confirm decisions for virtually any team in the company. Let’s take for example the graphics team. In case the company is selling cosmetic items in the game, teams need to know which ones of them are popular and why. If the user is able to purchase the wanted cosmetic item straight away do they buy it instantly or do they keep coming to admire it before making a purchase decision? How long does this process take and is there anything marketing team can do to enhance it? On the other hand, when item drops are random like let’s say loot boxes in games like Hearthstone or Overwatch is there a correlation between releasing a cosmetic item batch and increased purchase rate? In the same way, other teams want to know statistics concerning their projects like tech team wants to get information on the number of errors and community team wants to figure out what the players actually want regarding media and to feel like they are part of some bigger community. A good way to figure out what could be done is to ask the team if they had all the information they needed, what would be the thing/facts they wanted to know more about regarding their job.

2.2 Important steps in designing a data solution for a game company

Now that we’ve established how important data analytics are for a game company comes the question what now? How to proceed after realizing this is something the company needs to develop as a part of their game development? The first plan must be made as with everything. There are various ways of tackling this problem, but this thesis is going to use the S.M.A.R.T approach where every letter corresponds to a step from start to finish in designing a data solution.

The first step is **S**tart with strategy. The key here is to start thinking what’s the minimum amount of data needed to make reliable decisions on key metrics. For a small company especially, it’s impossible to store all of the data indefinitely. Is there a way to calculate the numbers in advance and store only the result? Many databases have a functionality like this. An example of this will be discussed in the chapters concerning InfluxDB (3.2, 3.3). Balancing storing period of time and amount of data takes a startup company far already and learning it from the get-go helps in the future. A good start would be to only take the minimum amount of metrics and store it for 2 weeks / month. This gives enough time to actually see if a change in the game affected the statistics in some way. A good way to get a general view on data strategy is to use the SMART Strategy board in figure 1.

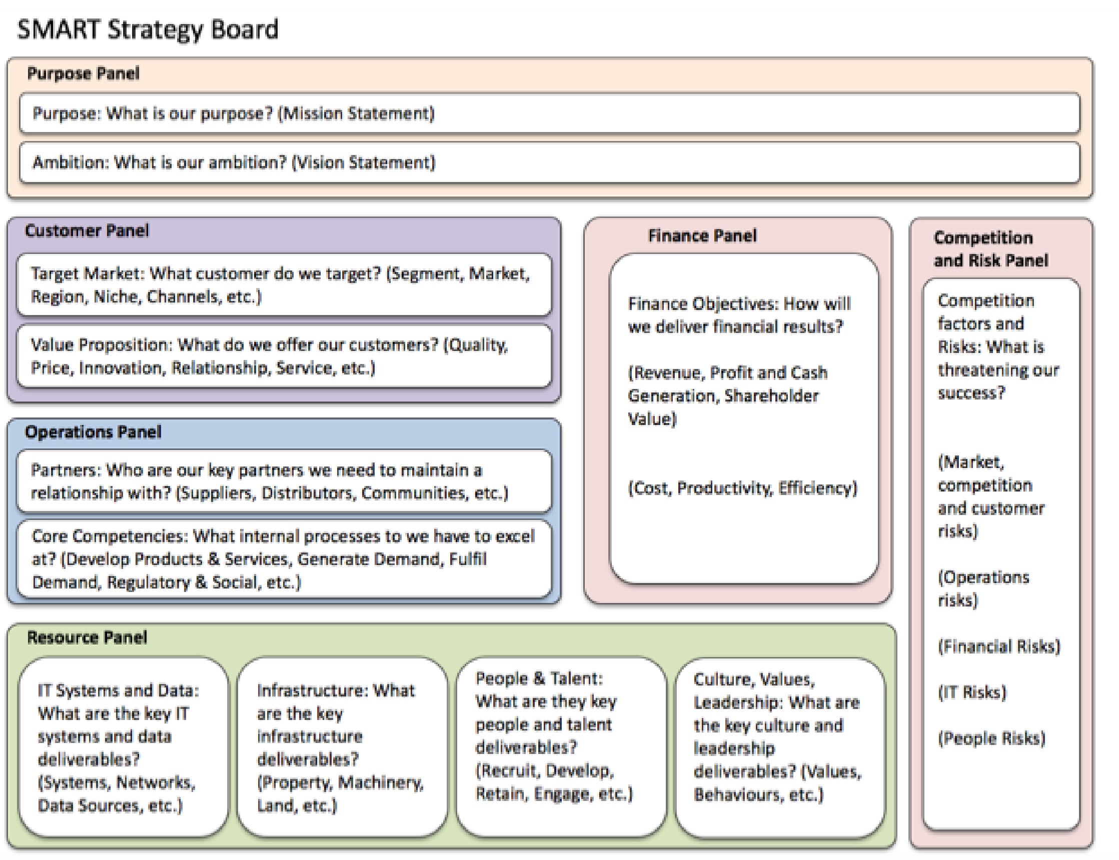

SMART Strategy Board
Purpose Panel
Purpose: What is our purpose? (Mission Statement)
Ambition: What is our ambition? (Vision Statement)
Customer Panel
Target Market: What customer do we target? (Segment, Market, Region, Niche, Channels, etc.)
Value Proposition: What do we offer our customers? (Quality, Price, Innovation, Relationship, Service, etc.)
Operations Panel
Partners: Who are our key partners we need to maintain a relationship with? (Suppliers, Distributors, Communities, etc.)
Core Competencies: What internal processes to we have to excel at? (Develop Products & Services, Generate Demand, Fulfil Demand, Regulatory & Social, etc.)
Finance Panel
Finance Objectives: How will we deliver financial results?
(Revenue, Profit and Cash Generation, Shareholder Value)
(Cost, Productivity, Efficiency)
Competition and Risk Panel
Competition factors and Risks: What is threatening our success?
(Market, competition and customer risks)
(Operations risks)
(Financial Risks)
(IT Risks)
(People Risks)
Resource Panel
IT Systems and Data: What are the key IT systems and data deliverables? (Systems, Networks, Data Sources, etc.)
Infrastructure: What are the key infrastructure deliverables? (Property, Machinery, Land, etc.)
People & Talent: What are they key people and talent deliverables? (Recruit, Develop, Retain, Engage, etc.)
Culture, Values, Leadership: What are the key culture and leadership deliverables? (Values, Behaviours, etc.)

Figure 1. SMART Strategy board to guide finding key questions

This might seem like a strategy targeted to starting a business, but it can also be used to figure out data needs for the company by finding the questions that the data can answer. Remember, it’s always about quality, not quantity. It doesn’t matter how much data there is if no one knows how to get value out of it. Ask a question or presume a hypothesis, then find out what data is needed to answer that question with certainty. (Marr, Big Data: The 5 Vs Everyone Must know, 2014, ss. 23-56)

Measure Metrics and Data is the second point. After finding out the questions the company wants answered it’s time to start thinking of laying foundations to get that data. This is done by figuring out the data model for the company. Data model is a plan about how the company stores, processes and accesses the data. It also entails what type of data will be used. The types are Internal, External, Structured, Unstructured and Semi-Structured. Internal data is all of the data generated by the company itself and owned by the company or data the company has access to, while external is everything that's gained from third-party companies. Examples of external data would be in the case of game company's data from Twitch or YouTube about the popularity of the game and their viewing statistics. Structured, Unstructured and Semi-Structured data types are covered in chapter 3.1 Typically, these are put into following hierarchy: Internal structured data, Internal semi-structured data, Internal unstructured data, External structured data and External unstructured data. This way of ordering ensures that process starts with the easiest to acquire and easiest to analyse data. So essentially from inside out and from easiest type of data to analyse to the hardest. (Marr, Big Data: The 5 Vs Everyone Must know, 2014, ss. 57-104)

Apply Analytics to find out what information gathered data contains. After finding out the questions that need to be answered and the data is gathered it's time to start analysing it. Here it's all about the software's talked about in this thesis and what they can do. This step also concerns the most important V of the 5, value. Data is worthless if value can't be dug from it using data analytics. Analysing the data should always start from the most critical questions that the company wants a certain answer for. Feel free to test the limits of what they can do and how flexible the system is, but one should not expect to use all of the functionalities for everything. Analytics is about answering complex questions as efficiently as simply as possible. Overcomplicating data analysis and trying to analyse everything leads to problems due to losing focus on what was the question that was meant to be answered. A good rule of thumb is to start from the most generic data found about a certain feature or problem and once something to research is found analyst can go deeper and deeper layer by layer until he finds the root cause that answers to the question of what's wrong or what can be improved. This way as little time and resources as possible are used while maintaining maximum efficiency in the analysis. (Marr, Smarter Business, 2015, pp. 105-154)

Report Results after data findings. After finding answers to the questions that were asked it’s time to visualize it. This doesn’t contain only choosing the right graph. The employee doing it also must think about the title, description, colours, value types, order etc. All of these points attribute to a successful visualization which communicates in a clear and concise way the results from analysing the data. This is a challenge where analyst must face two different worlds. On the other end we have tech team who know what’s going on under the hood and on the other hand, we have a business team who know about sales and user acquisition. The task is to make a visualization that satisfies the needs of both, so the visualization must be deep enough in information and shallow enough in simplicity. Remember that the main goal always with visualization is that anyone can glance at it and understand what it’s trying to achieve and how the results are read. (Marr, Smarter Business, 2015, pp. 155-198)

Transform Business is the final step of S.M.A.R.T. The questions have been asked, data has been gathered, it has been analysed and finally, it’s been visualized. Now it’s time to act on the results and transform the business by integrating the results into existing work outside analytics. This is also the time to start segmenting users into classes based on their attributes and behaviour. Things like geolocation, rank in the game and operating system used are some of the easier segmentations to start with and already offer a plethora of information. It’s also time to decide whether the result is something that needs to be done only once, every weeks/month or is it something to follow constantly. Analysing data is something to be constantly improved and followed up on because data never ends. It just gets bigger and bigger. (Marr, Smarter Business, 2015, pp. 199 - 230)

3 CHOOSING A DATABASE

Choosing the right database for a game company can be a challenging task, here matter to be discussed will be what should be thought of before getting overwhelmed by the data. First, we'll be looking at what kind of attributes should be accounted for when designing a database solution through 3.1 to 3.5 and after going through this it's time to talk about practical problems that might arise inside and outside the database software. Lastly, we'll go through the introduction of the selected database for this database and also talk about a potential contender to it, should one not be able to find what they want from a database in the first choice. Choosing a database is by no means a simple task for anyone, but due to limited resources and time sometimes it's hard to pinpoint a product that would be just right for the company as with game companies there are a lot of genres and specializations for different companies. Of course, the size of the company plays a big role in this, as bigger companies are able to broaden their selection and get better contacts to where they want with their influence and larger resources. The software used in this thesis is deemed compatible for game companies of varying sizes who want to know when do players do everything, thus time-series database was chosen.

3.1 Type of Data

First one must think about what kind of data does a game produce? What type of a game is it? Is it free to play, pay in advance, subscription-based or a mix? What does it monetize on and what kind of players are its target audience. There are tons of questions developer should ask about the data the game is producing. (Torres, n.d.)

One of the ways to handle player data is to use time series based data, where the main identifier for a data point is the timestamp. This way analyst is able to say when and how many users did thing X like buy something from the in-app store or kill an enemy boss during the special event.

In general, first thing regarding data type is to find out game's KPI's and design database prototype around those. What do company's DAU, CCU and Retention numbers look like? How long period of time of data is needed to measure for the decision maker to feel sure that the data is accurate? After feeling like the minimum viable amount of analytics has been grasped it's possible to start building the solution further. This will also help to vision what kind of database does the game company need.

Databases are usually divided into two different categories. Relational Databases (RDBMS) and Non-relational (NoSQL). Relational databases are the more common ones due to NoSQL being fairly new technology. Relational databases are structured databases where each table is connected to another via relationships using primary and foreign keys. This part is explained more in detail in chapter 3.4 talking about relativity in data. NoSQL, on the other hand, is unstructured data which is data that has no relation or structure to it. How is it possible to combine sound or video data to purchase data? Text messages to user data? These are just a few examples of data, that usually only has metadata to identify it. All of the previously mentioned examples are user made pieces of information that don't have any kind of set form or length to them. A Person can write whatever kind of text message he or she wants, same with audio and video data. The only factor that these might share is the geolocation and time. Because unstructured data is not indexed in any way it's very easy to store, but hard to search from. A good figure of speech differentiating structured and structured data is from professor Heinze Havinga, 2016 Enschede:

> *"Structured data is like a post office where you categorize everything in their respective lockers and you know exactly where everything is supposed to be, making it easy to find but takes a lot longer to organize. Unstructured data is a situation where you file everything very fast just throwing any paper into any container, but it will take ages for you to find what you're looking for later."*

The last datatype is semistructured data, which is a combination of the two previously mentioned. It contains data that's not fit for traditional databases (XML, JSON for example). Semistructured data uses a tree model instead of having a relationship and it’s used when dealing with different databases at the same time. Because of its nature, it’s very flexible, self-describing data that can be used in various manners. The hierarchy is very similar to normal folder tree where each folder branches out into subfolders. This way it’s easy to know where everything is, but it’s hard to combine two different branches together since essentially the search must move all the way from the other branch to the other without just going diagonally across like with a relational database. (Marr, Smarter Business, 2015, pp. 57-104)

So overall all datatypes have their uses for different things and must be carefully researched which is needed and where. (Foote, 2016) (Taylor, 2017) (Marr, Smarter Business, 2015)

3.2 Scalability

With great data, become great gigabytes. Scalability is something that should be thought about from the start since it’s ideal to have overhead in the system no matter how fast the company grows. Question to ask is: “What are the predictions for next year if currently there are 50,000 daily users?”. Questions like this help plan the database accordingly so that systems won’t get suffocated in case the game suddenly becomes a trend that makes everyone want to play it. A good example of this is how fast PlayerUnknown’s Battlegrounds grew in figure 2:

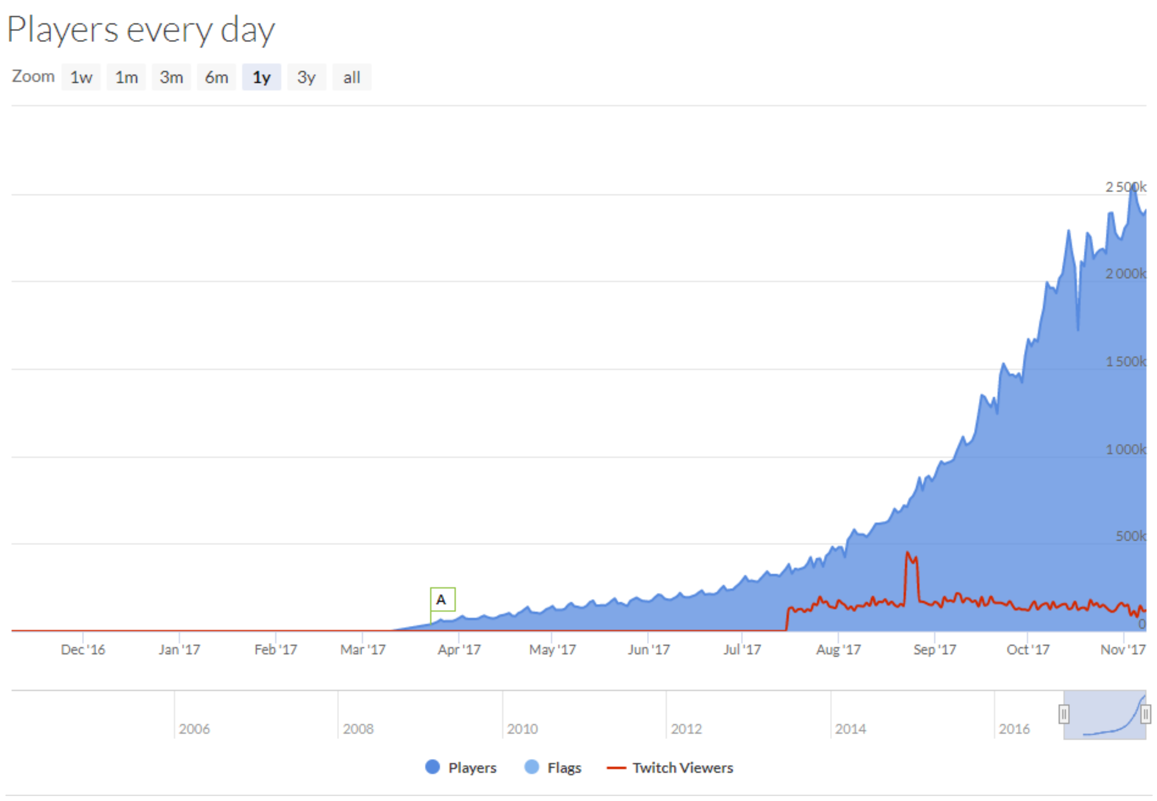


Figure 2. PlayerUnknown’s Battleground daily users in a year

Looking at halfway July when the first Twitch viewer trend begins, how at the same time players graph rises. It’s even easier to see that huge increase if one looks at the small line graph in the lower right corner. They started out very steadily rising, but then they got people engrossed in it and it blew out of hand so to speak. This is the ideal situation a company wants for a game and it must be prepared for it. In 1 month between August 17. And September 17 they almost doubled their daily players up from 500,000 to almost 1 million. This has required a lot of work to scale so, that everything gets through and is stored correctly. What if they wouldn’t be able to scale their database in time? Well, they would 1. Loose valuable data 2. Their numbers wouldn’t match 3. No one could trust the data so it’s useless and finally 4. It could clog the systems thus making even more problems. This would create a terrible situation where managers would have to answer to stakeholders and other similar key figures, that they can’t know for certain. Data is always about being certain.

Another important point is to think of how long is it possible to store the data. It’s important, that it’s possible to check on progress on a weekly, monthly, yearly basis. This way it’s easier to see how does, for example, a feature that’s been in development for months affect player acquirement and behaviour. Data should be kept so, that it’s stored as long period of time as possible while requiring a minimal amount of space. This is done via different methods of downsampling. It should also be possible to pull the data anytime it’s needed for analysis. Care is adviced, as too many different measurements stored infinitely can be a disaster waiting to happen unless talking about a big company with a huge budget. Optimize it early to reap the benefits in a long run. (xPaw, 2017)

3.3 Ease of use

What’s the point of having a state of the art system capable of scaling up to hundreds of thousands of players if no one knows how to use it? This brings us to the third point, ease of use which defines what kind of employee does the company need to do the database tasks. What language does it use? Is it easy to find support online (the more common a software is, the more google answers there are in case of problems)? How easy is it to modify and implement new metrics that the management wants to be measured? These are just few example questions to be thought of while pondering what kind of a system would be good for the ease of use aspect.

Ideally, the solution is easy to get into and all the data can be modified simply by just one person. Is the support of multiple people required to implement a measurement in the database due to infrastructure not being built for it? The more people needed to implement the data measurement, the more company must pay for each data point due to interruptions in other people’s tasks, communicating everything, delegating tasks etc. All of this adds up fast.

3.4 Relativity

Relativity describes how is it possible to combine different measurements together to gain further insight into player behaviour. A very common example of this is daily active users vs. action X, where action X could be entering a store inside the game. By dividing the store goers with the daily active users it's possible to count the percentage of how big portion of the player base actually visits the store on a specific day. In a relational database, relativity is handled by primary and foreign keys Which identify unique rows in a database. (Microsoft, 2017) (Murariu, n.d.)

Primary keys are in the original table and identify the rows in it, while foreign keys are "loaned" keys which identity which row in the foreign key table matches the row in the primary key table. This way we're able to combine the two tables on this key and more specific data. (Microsoft, 2017)

So, what is wanted in terms of relativity in a database? The answer is a universal relation. A situation where it's possible to combine pretty much any data together and tie it all up so that it's possible to get the complete path a player took during game sessions. This includes data like user ID, geolocation, rank, buying habits and preferred game mode. The more things that are possible to tie together in a reliable way a complete picture it's possible to acquire about players. (Marr, Big Data: The 5 Vs Everyone Must know, 2014)

3.5 Security

Security is always essential, but very much so when it comes to data. Data is so important to companies, that we're seeing an increasing number of hackers stealing it. Personal data is expensive and company data even more so. In Netherlands, in January 2017 author of this thesis was part of a project researching how to make cities smarter in the countryside. Data about how many people visited the city during 6 months based on phone company's data would have cost around 28,000€, so it wasn't possible to purchase it with the given budget. Here's a list is taken by Forbes about the estimated cost of personal data by Sophie Curtis (Curtis, 2015):

"Bank login credentials for a $2,200 balance bank account: $190

Bank login credentials plus stealth funds transfers to US banks: from $500 for a $6,000 account balance, to $1,200 for a $20,000 account balance

Bank login credentials and stealth funds transfers to UK banks: from $700 for a $10,000 account balance, to $900 for a $16,000 account balance

Login credentials for online payment services such as PayPal: between $20 and $50 for account balances from $400 to $1,000; between $200 and $300 for balances from $5,000 to $8,000

Login credentials to hotel loyalty programs and online auction accounts: $20 to $1,400

Login credentials for online premium content services such as Netflix: as little as $0.55"

Note that these are just prices for 1 single person. If these numbers are multiplied by 1000, 10,000 even 100,000 at a major company one gets a picture of just how tempting data can be to a cybercriminal. Caution is advised always. Backup, secure, monitor and limit access to the database always. (Layden, 2017).

By backing up the database company is safe from corruption crimes, where criminal sabotages the database making the data in it unusable. Backups are also the best defence against employee mistakes that could have irreversible consequences (Rubens, 2016)

Securing the data with a strong firewall and encryption against threats from outside works well, but what should be done against a physical threat? The database should always be locked up where only select few are able to physically access it because no matter how good the security software is, it doesn't do much against a physical threat. (Rubens, 2016)

Database usage should be monitored on a regular basis to see if there is anything out of the ordinary going on, like an admin logging on outside of work time and doing things not related to current tasks. This could inform the monitoring personnel that someone has gained access to admin’s account credentials. (Rubens, 2016)

Access should be limited in the database, so only the people responsible for making changes like data analysts and administrators should be allowed to access it. For other people, read-only access is enough. The more people there are who can do a modification to the database the more risks are taken protecting company’s data. (Rubens, 2016)

3.6 What problems might arise

In addition to the above-mentioned things to take care of when planning for the database, there are two kinds of problems databases could potentially face. These can be divided into problems inside the database and outside the database software. Problems outside of the database are things like poor documentation, badly designed server physical location and unclear responsibilities regarding the database which often goes unforeseen for a long time.

Problems inside the database are on the other hand more commonly noticed and attended to. This includes things like poor naming standards (always standardize everything in the database), not optimizing the data leads to performance issues, lack of testing will lead to unforeseen bugs which are hard to track later. Database software should be checked to be compatible with the visualization software unless the company wants to build everything from scratch. (Davidson, 2007)

3.7 Why InfluxDB was chosen?

InfluxDB is easy to set up and working (chapter 5). It’s also completely compatible with Grafana as both are meant to specialize in time series data. It’s an open source software which uses SQL-like query language called InfluxQL, with the major difference being the inability to do joins unless done via Continuous queries. Continuous queries are queries that loop at specific intervals and downsample data to single data points. This reduces the load on the database as one can calculate the results beforehand and instead of having all data points included in the calculation only the result is stored. Other ways of reducing the load with InfluxDB is done via sharding which breaks large databases into smaller more easily managed parts on different database server instances.

Another important tool to reduce the load is retention policy which dictates how long data is stored in the database until it’s automatically deleted. The best way to act is to store default data for a short amount of time via retention policy and have another database instance storing the data infinitely. The way this works is that the user makes a continuous query out of the default data and stores the resulting data point indefinitely in the infinity storage. As an example, let’s say data is coming every minute and it’s stored for 24 hours. The task is to count value X, which is sum of the data point values for the day. In case that analyst keeps using the minute values it results in 24 x 60 = 1440 data points. After making a continuous query which loops every 24 hours calculating the sum of those values and storing that result 1 data point is stored daily instead. So instead of 10,080 data points every week there would be 7. (solid IT, n.d.) (influxdata, 2017)

3.8 Second choice

The second choice for the database is Graphite, which is also compatible with Grafana. It’s an open source time series database written in Python and is ranked #3 on time series database rankings. It supports tags and has built-in processing backend named Carbon, which processes the data sent to Graphite before storing it in the database. Graphite supports many ways of sending data into carbon, main methods being Plaintext, Pickle and AMQP. To send a large amount of data it’s recommended to batch up the data and send it to Carbon’s pickle receiver so it should be user’s first choice when building Graphite solution for a company, as game’s need to send large amounts of data most of the time.

Graphite offers a lot of the same data analysis functionalities as InfluxDB disregarding continuous queries. However, Graphite's query language is far harder to learn due to not having a clear query language and the documentation of it is lacking, which can lead into problems. Graphite requires you to instead do the querying with functions that apply one metric at a time and can be pulled into files like CSV. To combat this difficulty Graphite has a web interface, which works similarly to Grafana where the user clicks on the data and functions he wants and the software builds the query and visualization for him. This however can limit the potential, since when queries get complex enough it becomes increasingly more difficult to make them in a web interface and instead manual querying is preferred, which Graphite has hard time answering to. Overall Graphite is a good tool to use, but requires a lot more technical knowledge to get going and to use. Also, being independent project free from company rule means that the documentation is sometimes very lacking and requires user to find a developer responsible for it and message him or try solving it on their own. (Graphite Project, 2017) (Solid IT, n.d.)

3.9 Hardware Requirements

Influxdata defines different kinds of hardware requirements based on whether user is wanting to use a single node or a cluster. Single node system works on a single machine and does all the calculations on itself. Cluster on the other hand is a stack of different machines which share the load between each other. This enables a company to buy many cheaper machines and make them into a cluster which rivals the power of the single node with potentially far lower price. We'll be focusing on single node service of Influx as their closed-source software needed for the clustering is not a free open source software, but a commercial product instead. Below in table 1. Are guidelines based on Influxdata's own recommendations. (InfluxData, 2017)

Table 1. Hardware requirement guidelines

Load	**Field Writes Per Second**	**Moderate queries per second**	**Unique Series**	**CPU Core**	**RAM GB**	**IOPS**
Low	< 5000	< 5	< 100k	2-4	2-4	500
Moderate	< 250,000	< 25	< 1mil	4-6	8-32	500-1000
High	> 250,000	> 25	>1mil	8+	+32	1000+

For storage, Influxdata recommends to always use SSD’s, as HDD’s might break easily under even moderate load. Storage capacity is dependent on the amount of data produced where non-string values are approx. 3 bytes. String values are dependent on string compression. To further reduce write load directories wal and data should be on separate storage devices. (InfluxData, 2017)

Graphite doesn’t have recommendation guidelines in their documentation, but in-stead give a vague recommendation of having good RAID array and/or SSD’s if working with high volumes of data, since every distinct measurement is stored in their own database file and this creates a lot of demand regarding I/O operations. Excess data is kept in memory until it can be processed, but thankfully Graphite allows writing of multiple datapoints at once. Since there aren’t any clear studies regarding hardware sizing it’s highly advised to make a prototype and see how it behaves with the company’s data. (Graphite Project, 2017)

4 CHOOSING A VISUALIZATION SOFTWARE

Choosing a visualization software isn't technically as challenging, but even more so when weighing how it will be used. A hard to use visualization software can halt the whole analytics process. Visualization software, likewise to database has some key attributes that should be thought of when making plans on what kind of visualization software serves company's objective in a most optimal way. In these chapters, first we'll be talking about the previously mentioned attributes in 4.1 to 4.3 and after that dive in the same manner as with choosing a database what problems should be considered. Lastly why Grafana was chosen and what would be the other potential data visualization software to consider. Data visualization is often thought to be simple and not so important, but in some ways, it's even more important than having a properly working database as not having or being able to visualize the results means that it doesn't serve the purpose of being easily understood by all people necessary, sure it might make sense to a technician doing the analytics, but to let's say a manager it doesn't make sense until the person doing the analytics explains where these numbers came from and how. It's been proven that human brain is able to understand pictures a lot better than words and numbers it's vital that the data used on a regular basis in the company is visualized in such a form that every person that needs that information is able to understand it. Visualization also helps the person to connect different data to each other giving a better picture of the overall situation rather than just a single detail. (Gillett, 2014)

4.1 Ease of use

Ease of use determines how much training and knowledge is needed to be successful in analysing the data. This is important, as that level can vary a great deal. Some visualization software's are designed so the user can just click ready-made functionalities in the user interface and choose the option he wants to use, kind of like answering a quiz with drop-down answers.

The other side of the spectrum, however, is a situation where such UI is non-existent, thus forcing the user to have a lot of programming knowledge in the language the software uses. This creates a poor situation where it doesn't end at the point of users requiring programming knowledge, but they most likely need to know one specific one for anything to work. This limits the options by a large margin in both possible software's and employees.

Ready-made UI with just clicking elements can however also have a problem where it's too constricted regarding analytics the software allows the user to do creating a bottleneck. Thus, a golden middle road is needed. A software which allows the users to learn it fast by allowing using simple ready-made elements for analysis, but at the same time has the functionality to program everything by hand. This way the software's scope of users becomes wider and more employees can use it for their tasks regardless of their competence. By having a software like this it's also possible to reduce human resource usage by a large margin, since people don't have to constantly communicate to get results, instead relying on each other only when data analysis gets complex enough. It's also crucial to make sure that the visualization software is able to create all of the graph types the company needs like pie charts, line graphs, histograms and heatmaps. A solution to gauge this would be to have both one experienced person and one complete rookie do tasks fit for their competence levels. Are they both able to complete their tasks or is the software too difficult/restricted? This way it's possible to test both ends of the spectrum and get a general idea of the balance between depth and user-friendliness.

4.2 Speed

Speed is the second important point to think of when choosing a visualization software. It's a must-have for both the visualization software and the database. These need to work together so, that every ran analysis is as fast as possible. This does two things. First one is that the person analysing doesn't have to wait on the results for a long time decreasing the number of possible distractions, that tasks are done while waiting for the query to finish might cause. The second point is that queries often need tuning when making them get the wanted results in a clear concise way and that requires multiple runs on the same query. If the query takes a long time to calculate every time it adds up fast and because of this, hogs up a lot of valuable work time and effort. In short, catching up and continuing the task employee started yesterday just because the analysis is slow is detrimental to the work quality and quantity.

4.3 Compatibility & Customisation

As with database to visualization software direction, other direction must work also. This ensures that unwanted bugs are minimized, and usage should be fairly straightforward. Customization is another key part of visualization software, as it determines how a deep and wide range of visualizations are possible to make with it. This also gives the company an ability to make the visualization look exactly like the company wants, even going as far as having the same theme. This makes it more visually pleasing and motivating for the employees. As they are looking at the graphs they are able to connect that data to the work they are doing. Customisation also contains the ability to make the dashboards and graphs to look just the way wanted, as this has a lot of importance on how easy it is to interpret the data shown. Dashboards end goal always is to show the data in such a way, that just by a quick glance everyone understands what it’s trying to achieve and how the different graphs connect to each other. A good point to start is to do the following: avoid 3D graphs at all cost since they often exaggerate the data, put the same graph types together and use the same colours for data that have relation to each other. (Matillion, n.d.)

4.4 What problems might arise

Problems regarding data visualization software’s are mainly about security in addition to the ones previously discussed. Visualization software has access to the database’s data and as such it’s a potential loophole for criminals to get in the database. Visualization software is also the easiest way for the criminal to see what kind of valuable data the company is storing, as these are put into easy to read graphs across the software. Just a picture of the graphs can have value to someone, like competitors in the same market. Another risk is the employees of the company. As visualization is the tool in analytics that has most users an employee who doesn't know what they are doing can cause serious harm if everyone is able to configure the graphs. Limiting access is because of this something every administrator should do in the visualization software. Some software's like Grafana offer a user type where the user is able to modify the graphs, but can't save them thus eliminating potential harm an employee can do.

4.5 Why Grafana was chosen?

Grafana was chosen because of its overall flexibility. It supports various backend data sources like previously mentioned InfluxDB and Graphite, it's open source and best of all it's easy to get into. Installation is very simple and what Grafana doesn't have in plugins is possible to make by programmers themselves. It has an easy to use graphical interface, which allows for smooth learning for anyone using Grafana, but at the same time having enough depth so that it doesn't restrict complex queries. It's possible to either use the clickable query editor or toggle manual mode where the user writes everything by hand. Other useful features are features like templating, alerting and changing time range on the fly. Overall Grafana is an excellent tool for time series data which can run on minimum hardware (250mb ram and single processor core). (Grafana Labs, 2017, p. Features) (Mostowfi, 2017)

4.6 Second choice

The second choice is Chronograf, which is made by Influxdata. It's their own answer to provide a competing visualization software for Grafana. It's part of TICK stack where it's part of a whole consisting of Telegraf, Kapacitor, InfluxDB and Chronograf made by Influxdata. It offers alerting, visualization, Grafana like easy to learn UI, templating and maximum support between InfluxDB and Chronograf due to being from the same company. What Chronograf lacks however, is the documentation quality of Grafana. Bare minimum is explained in their documentation which leads the user to either having to contact Influxdata for support or try find discussions regarding Chronograf in Influx user groups discussing features, bugs and helping each other. (influxdata, 2017, pp. Create a Dashboard, Dashboard Template Variables) (influxdata, 2017)

5 INFLUXDB INSTALLATION

Installation is done on an Ubuntu 16.04.02 LTS Server virtual machine, as Ubuntu is most familiar to Linux users. Ubuntu also has a lot of users and technical support, should anything go wrong. In this chapter we go through the installation of InfluxDB, required configurations, problems faced, other possible sources needed and what we need to do from InfluxDB perspective to make it compatible with Grafana. The server only contains InfluxDB installation with ip: 10.30.30.154

5.1 Installation of the Database

Firstly, adding the Influxdata repository is needed according to the documentation guide (Influxdata, 2017). So, we use the commands in the order:

curl -sL https://repos.Influxdata.com/InfluxDB.key | *sudo apt-key add – source /etc/lsb-release*

echo "deb https://repos.Influxdata.com/${DISTRIB_ID,,} ${DISTRIB_CODENAME} stable" | *sudo tee /etc/apt/sources.list.d/InfluxDB.list*

sudo apt-get update

sudo apt-get install InfluxDB

and then start InfluxDB with the command:

sudo service InfluxDB start

Please note the recommendations by Influxdata team found from running above commands:

> *“Hardware We recommend using two SSD volumes. One for the InfluxDB/wal and one for the InfluxDB/data. Depending on your load each volume should have around 1k-3k provisioned IOPS. The InfluxDB/data volume should have more disk space with lower IOPS and the*

InfluxDB/wal volume should have less disk space with higher IOPS. Each machine should have a minimum of 8G RAM. We've seen the best performance with the R4 class of machines, as they provide more memory than either of the C3/C4 class and the M4 class."

6 GRAFANA INSTALLATION

Installation for Grafana is also done on 16.04.02 LTS Server virtual machine for the above-mentioned reasons and to ensure the compatibility of the two servers. In this chapter, we'll go through the installation of Grafana and make it ready for the last part where we'll combine the two. Grafana resides on another virtual computer with ip address 10.30.30.153

6.1 Installation of the Grafana visualization software

First, we'll Install the Stable version by running commands below. This is the first way to just install a newest stable version of Grafana.

Wget https://s3-us-west-2.amazonaws.com/grafana-releases/re-lease/grafana_4.6.1_amd64.deb

sudo apt-get install -y adduser libfontconfig

sudo dpkg -i grafana_4.6.1_amd64.deb

The first wget downloads the Grafana files from Amazon AWS environment. Then

With the sudo apt-get install command we install the chosen package, where all questions are answered with yes due to -y on the command line. Finally, we add a user named libfontconfig.

With the last command sudo dpkg -i we install the chosen package, which in this case is, of course, Grafana 4.6.1 stable version.

The second way to install is via APT repository. It is done by first adding the following line to our sources list with the command. Sources list is a roadmap for the system to know where it can download packages for installation and updat-ing.

Sudo vim /etc/apt/sources.list

The line to add is:

deb https://packagecloud.io/grafana/stable/debian/ jessie main

This tells package manager where to get new Grafana versions of it in the future. Next, we’ll add the Package Cloud key to enable installing signed packages by running the command

curl https://packagecloud.io/gpg.key | sudo apt-key add -

Next to do is *sudo apt-get update* to refresh everything and finally install Grafana by running *sudo apt-get install Grafana*.

To start Grafana always at boot time use:

sudo update-rc.d grafana-server default

After all these steps a server reboot is required.

To start Grafana manually, type *sudo service grafana-server start*. (Grafana Labs, 2017)

6.2 Problems faced

Default address with port 3000 wasn’t working when trying to access Grafana via a web browser so port 80 had to be rerouted to port 3000 using the command:

Sudo iptables -t nat -A PREROUTING -p tcp –dport 80 -j REDIRECT –to-port 3000

This command only works for the current session. To make this permanent another series of commands is needed. First, the same command as before is needed and after it run commands:

apt-get install iptables-persistent

to install the needed software iptables that makes it possible to save port configurations. Make sure to click yes when asked to do the initial saves.

sudo vim /etc/network/interfaces

to go to network interfaces file and there add

pre-up iptables-restore < /etc/iptables/rules.v4

after dns-nameservers. Then write and quit using *:wq*

Now just make one additional save just to be sure everything was successfully saved.

iptables-save > /etc/iptables/rules.v4

7 CONNECTING THE TWO TOGETHER

Now that both services are running it’s possible to start configuring them to work together. The end result is portrayed below in Figure 3.

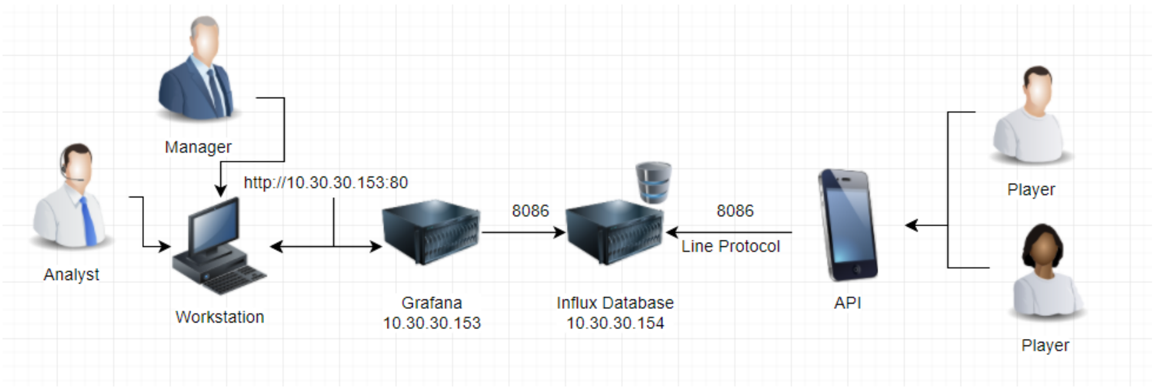


Figure 3. what the end result would look like in a mobile game company.

First, a database must be made and configured to have a place to store the data and query it through Grafana. This is done by commanding influx, while InfluxDB is running. This starts the CLI and connects to the InfluxDB instance. To create a database we use the command:

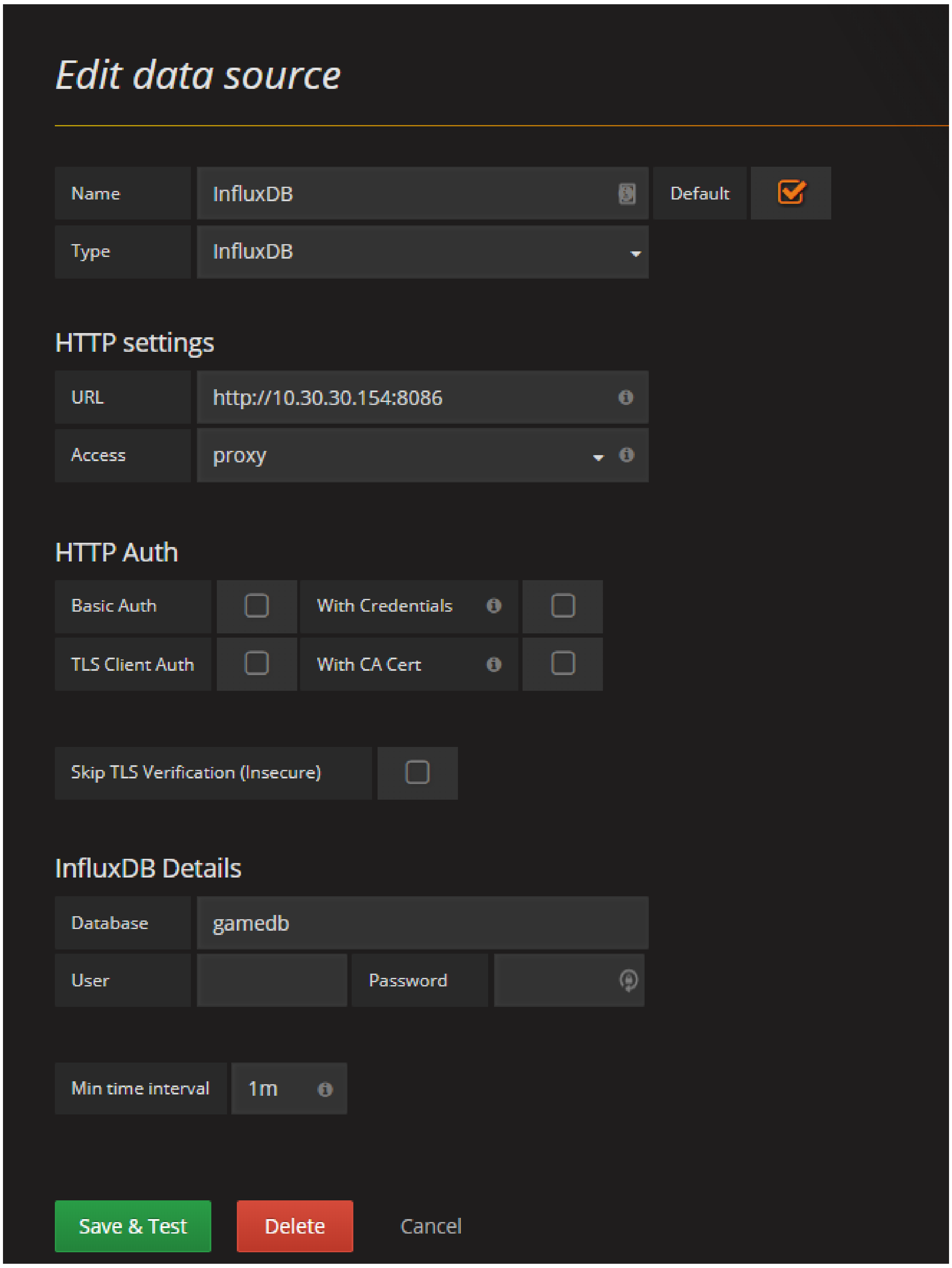
CREATE DATABASE gamedb

This creates a database called gamedb, which you can check by running the command:

SHOW DATABASES

It should list all databases which at this moment should be _internal and mydb. _internal is influx’s own database to store performance data. Now that a database is running it’s time to connect that database to Grafana. This happens by going to Grafana on the internet browser. Default login credentials are admin/admin. After logging in press Create data source from home dashboard or top left corner dropdown menu Data sources. Once in Edit data source screen type a name for the source and choose type: InfluxDB. Next, we need to set the URL and Access. For this work URL was: http://10.30.30.154:8086 where IP address is the address of the InfluxDB server and 8086 is the default port for

Grafana – Influx communication. Access is proxy so Grafana backend will be able to find data residing in InfluxDB. HTTP Auth can be left unchecked at the start since it’s not needed for the first installation. In InfluxDB Details, the name of the database is the previously created gamedb, which currently doesn’t have User or Password. Min time interval is set to the amount of time that is the minimum time interval of data points wanted to see (the bigger it is the less load there is due to forcing smaller datapoints as one. A good start is 1m, which shows the result of queries every minute as datapoints. Now if Save & Test was successful the two services have been connected together successfully (Figure 4. For example data source config).



Edit data source

Name	InfluxDB	Default	✓
Type	InfluxDB		

HTTP settings

URL	http://10.30.30.154:8086
Access	proxy

HTTP Auth

Basic Auth	☐	With Credentials	☐
TLS Client Auth	☐	With CA Cert	☐
Skip TLS Verification (Insecure)	☐		

InfluxDB Details

Database	gamedb		
User		Password	
Min time interval	1m		

Figure 4. an example of how data source should look like.

Now Grafana is ready to visualize the data if there is any. Just press drop-down arrow next to Home and + New Dashboard to start building visualizations. To test the system when no data is present it's possible to create data on influx command line. First, we have to use gamedb as the destination. This is done with command *use gamedb*. Now we need to insert data into the database. This is done with the INSERT command with the following format:

<measurement>[,<tag-key>=<tag-value>...] <field-key>=<field-value>[,<field2-key>=<field2-value>...] [unix-nano-timestamp]

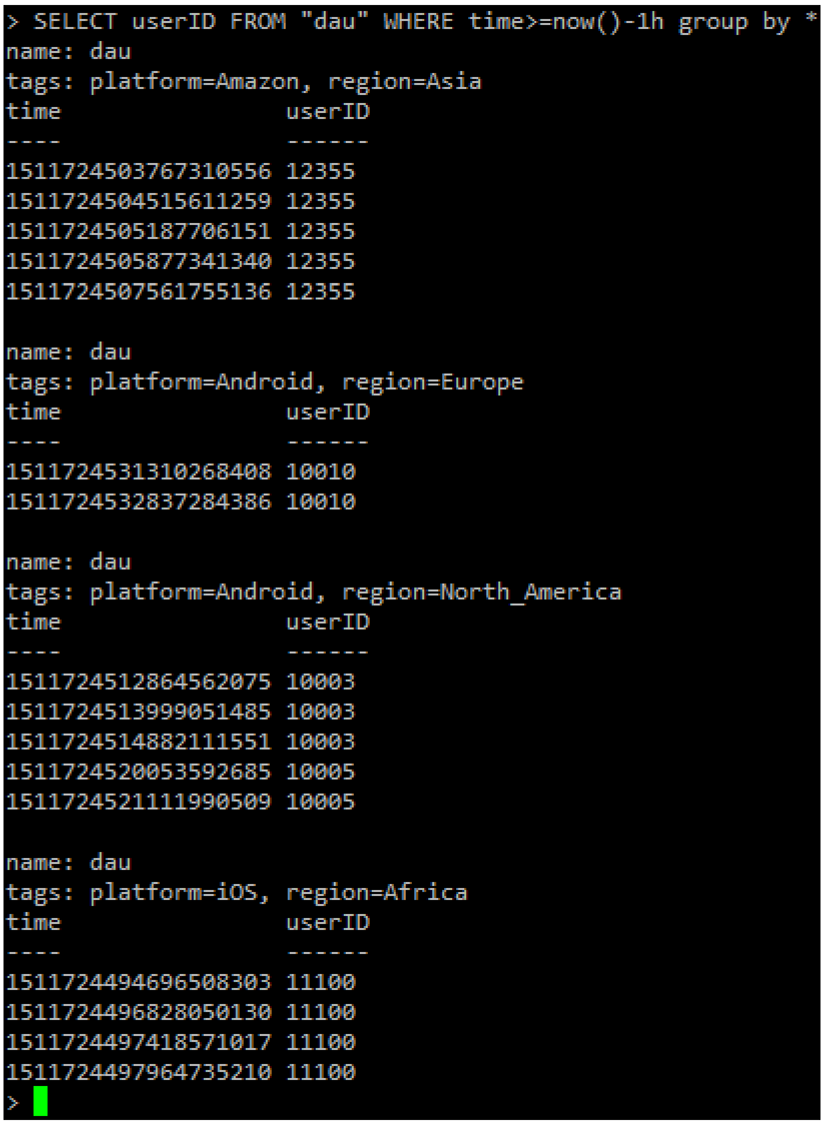
For this example, these queries were run.

```
INSERT dau,platform=iOS,region=Europe  userID=10002

INSERT dau,platform=Android,region=North_America  userID=10003

INSERT dau,platform=Android,region=North_America  userID=10004

INSERT dau,platform=Android,region=North_America  userID=10005

INSERT dau,platform=Android,region=Europe  userID=10010

INSERT dau,platform=iOS,region=Africa userID=11100

INSERT dau,platform=Amazon,region=Asia userID=12355
```

Here dau is the measurement name, platform and region are tags so they are indexed “primary keys” in the data and userID is a field value which is used to count the value using count(userID) and then segmenting the results based on the tags platform and region.

A visualization of the result in attachment 1.

To actually make a simple analysis of the data using the previous example's insert queries we will run them varying amounts.



```
> SELECT userID FROM "dau" WHERE time>=now()-1h group by *
name: dau
tags: platform=Amazon, region=Asia
time                userID
----                ------
1511724503767310556 12355
1511724504515611259 12355
1511724505187706151 12355
1511724505877341340 12355
1511724507561755136 12355

name: dau
tags: platform=Android, region=Europe
time                userID
----                ------
1511724531310268408 10010
1511724532837284386 10010

name: dau
tags: platform=Android, region=North_America
time                userID
----                ------
1511724512864562075 10003
1511724513999051485 10003
1511724514882111551 10003
1511724520053592685 10005
1511724521111990509 10005

name: dau
tags: platform=iOS, region=Africa
time                userID
----                ------
1511724494696508303 11100
1511724496828050130 11100
1511724497418571017 11100
1511724497964735210 11100
>
```

Figure 5. showing the example data from Influx command line

In figure 5, we run a query to check all userID’s from measurement named dau where time>=now()-1h so time period from last hour until this moment. Then we group by * which means grouping by all tags. This gives us all combinations of tags that the userID’s have in the last 1 hour. Now using Grafana it’s possible to count how many unique users regardless of region, but segmented by platform and counting only last 1 hour results. This is a basic daily active user calculation, just on a very small scale. Because Grafana doesn’t know how to count both distinct and count at the same time we need to use the manual input by pressing the first button on the right and selecting “toggle edit mode” which lets us make the query manually. Now we’ll just write the correct query using InfluxQL.

SELECT count(distinct(“userID”) FROM “dau” $timeFilter GROUP BY time(1h), “platform” fill(0)

This will make a graph with only 1 data point if the time period is zoomed to last 1 hour. Picture of the result can be found in the attachment 2. (Influxdata, 2017)

8 RESULTS

This thesis was meant to ease people in the game industry to get them started in using analytics in their own games. In that regard, the thesis successfully gives both theory and practical skills to start developing company's own system. I wanted to create a work that anyone with basic knowledge about IT can pick it up and find something to learn from it since analytics is necessary knowledge for many different employees in the company as discussed in the thesis. Practical work is possibly a bit too hard for a complete beginner using ubuntu server, but hopefully, it's at least easy to find out what actually happens during the installation.

Because of my work as a data analyst, the practical side of installing the system was hard for me since my work consists mostly of using ready-made systems to do my analysis work on. I wanted to challenge myself with this to better understand what's happening behind everything to help myself think why something is happening inside the software, especially when troubles arise. Now I'm more knowledgeable about InfluxDB and Grafana as software's and I can leverage it into my own work.

Overall, I'm very happy how this thesis turned out and I hope it will be useful to someone who wants to get into analytics, no matter if they are programmers, administrators, producers or just someone who wants to know more about it. I'm confident that after reading this the reader is ready to start his or her journey in game analytics. As last words, don't use approximates and gut feeling in your decision making, always use data because data is the truth.

References

Curtis, S. (2015, October 15). *How much is your stolen data worth on the dark web?* Retrieved from The Telegraph: http://www.telegraph.co.uk/technology/internet-security/11932927/How-much-is-your-stolen-data-worth-on-the-dark-web.html

Davidson, L. (2007, February 26). *Ten common Database Design Mistakes*. Retrieved from Redgate Hub: https://www.red-gate.com/simple-talk/sql/database-administration/ten-common-database-design-mistakes/

Foote, K. D. (2016, December 21). *A Review of Different Database Types: Relational versus Non-Relational*. Retrieved from Dataversity - Data Education for Business and IT professionals: http://www.dataversity.net/review-pros-cons-different-databases-relational-versus-non-relational/

Gerasimou, V. (2016, March 29). *Big Data and the 3Vs: What is the fourth 'V' and what are the implications for not embracing it?* Retrieved from ThinkBIG analytics, A Teradata company: https://www.thinkbiganalytics.com/2016/03/29/big-data-3vs-fourth-v-implications-not-embracing/

Gillett, R. (2014, September 18). *Why We're More Likely To Remember Content With Images And Video (Infographic)*. Retrieved from Fastcompany: https://www.fastcompany.com/3035856/why-were-more-likely-to-remember-content-with-images-and-video-infogr

Grafana Labs. (2017). *Installing on Debian / Ubuntu*. Retrieved from docs.grafana.org: http://docs.grafana.org/installation/debian/

Grafana Labs. (2017). *Welcome to the Grafana Documentation.* Retrieved from Grafana docs: http://docs.grafana.org/

Graphite Project. (2017, April 3). *FAQ.* Retrieved from graphite read the docs: https://graphite.readthedocs.io/en/latest/faq.html

Hill, K. (2012, February 16). *How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did*. Retrieved from Forbes: https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/#154123e06668

influxdata. (2017, June 20). *Chronograf Version 1.3 Documentation.* Retrieved from influxdata docs: https://docs.influxdata.com/chronograf/v1.3/

influxdata. (2017, June 21). *Continuous Queries.* Retrieved from influxdata docs: https://docs.influxdata.com/influxdb/v1.3/query_language/continuous_queries/

InfluxData. (2017, June 21). *Hardware Sizing Guidelines*. Retrieved from influxdata docs: https://docs.influxdata.com/influxdb/v1.3/guides/hardware_sizing/

Influxdata. (2017, October 2). *Installation*. Retrieved from InfluxDB 1.3: https://docs.influxdata.com/influxdb/v1.3/introduction/installation/

influxdata. (2017). *Open Source Time Series Platform*. Retrieved from influxdata: https://www.influxdata.com/time-series-platform/#kapacitor

Influxdata. (2017, October 23). *Query Language*. Retrieved from influxdata docs: https://docs.influxdata.com/influxdb/v1.3/query_language/

International Data Corporation. (2016, October 3). *Double-Digit Growth Forecast for the Worldwide Big Data and Business Analytics Market Through 2020 Led by Banking and Manufacturing Investments, According to IDC*. Retrieved from IDC Corporate Web Site: https://www.idc.com/getdoc.jsp?containerId=prUS41826116

Klein, A. (2017, July 11). *Hard Drive Cost Per Gigabyte*. Retrieved from Backblaze: https://www.backblaze.com/blog/hard-drive-cost-per-gigabyte/

Layden, J. (2017, September 20). *More data lost or stolen in the first half of 2017 than the whole of last year*. Retrieved from The Register: https://www.theregister.co.uk/2017/09/20/gemalto_breach_index/

Lovato, N. (2015, May 12). *Everything you Need to Know about Interpreting KPIs*. Retrieved from Gameanalytics: https://gameanalytics.com/blog/everything-need-know-interpreting-kpis.html

Marr, B. (2014, March 6). *Big Data: The 5 Vs Everyone Must know*. Retrieved from Linkedin: https://www.linkedin.com/pulse/20140306073407-64875646-big-data-the-5-vs-everyone-must-know/

Marr, B. (2015). Smarter Business. In B. Marr, *Big Data using smart big data analytics and metrics to make better decisions and improve performance* (pp. 9-10). Chichester: John Wiley & Sons Ltd.

Marr, B. (2017, March 29). *A simple way to prepare your business plan on a single page*. Retrieved from hiscox: https://www.hiscox.co.uk/business-blog/prepare-your-business-plan-on-a-single-page/

Matillion. (n.d.). *Dashboard examples: The good, the bad and the ugly*. Retrieved from Matillion: https://www.matillion.com/insights/dashboard-examples-the-good-the-bad-and-the-ugly/

Microsoft. (2017, July 25). *Primary and Foreign Key Constraints*. Retrieved from docs.microsoft.com: https://docs.microsoft.com/en-us/sql/relational-databases/tables/primary-and-foreign-key-constraints

Mostowfi, F. (2017, January 19). *Time Series Visualisations: Kibana or Grafana?* Retrieved from rittmanmead: https://www.rittmanmead.com/blog/2017/01/time-series-visualisations-kibana-or-grafana/

Murariu, C. (n.d.). *Active Users: Measure the Success of Your Business*. Retrieved from innertrends: https://blog.innertrends.com/active-users-2/

Press, G. (2017, January 20). 6 Predictions For The $203 Billion Big Data Analytics Market. *Forbes*.

Reddy, C. (2017, November 17). *Data Warehousing: Characteristics, Functions, Pros & Cons*. Retrieved from Wisestep: https://content.wisestep.com/data-warehousing-characteristics-functions-pros-cons/

Rubens, P. (2016, August 23). *7 Database Security Best Practises*. Retrieved from eSecurity Planet: https://www.esecurityplanet.com/network-security/6-database-security-best-practices.html

Schreier, J. (2017, May 30). *Top Video Game Companies Won't Stop Talking About 'Games As A Service'*. Retrieved from Kotaku: https://kotaku.com/top-video-game-companies-wont-stop-talking-about-games-1795663927

Solid IT. (n.d.). *Graphite System properties*. Retrieved from db-engines: https://db-engines.com/en/system/Graphite

solid IT. (n.d.). *InfluxDB System Properties*. Retrieved from db-engines: https://db-engines.com/en/system/InfluxDB

Taylor, C. (2017, August 3). *Structured vs. Unstructured data*. Retrieved from Datamation: https://www.datamation.com/big-data/structured-vs-unstructured-data.html

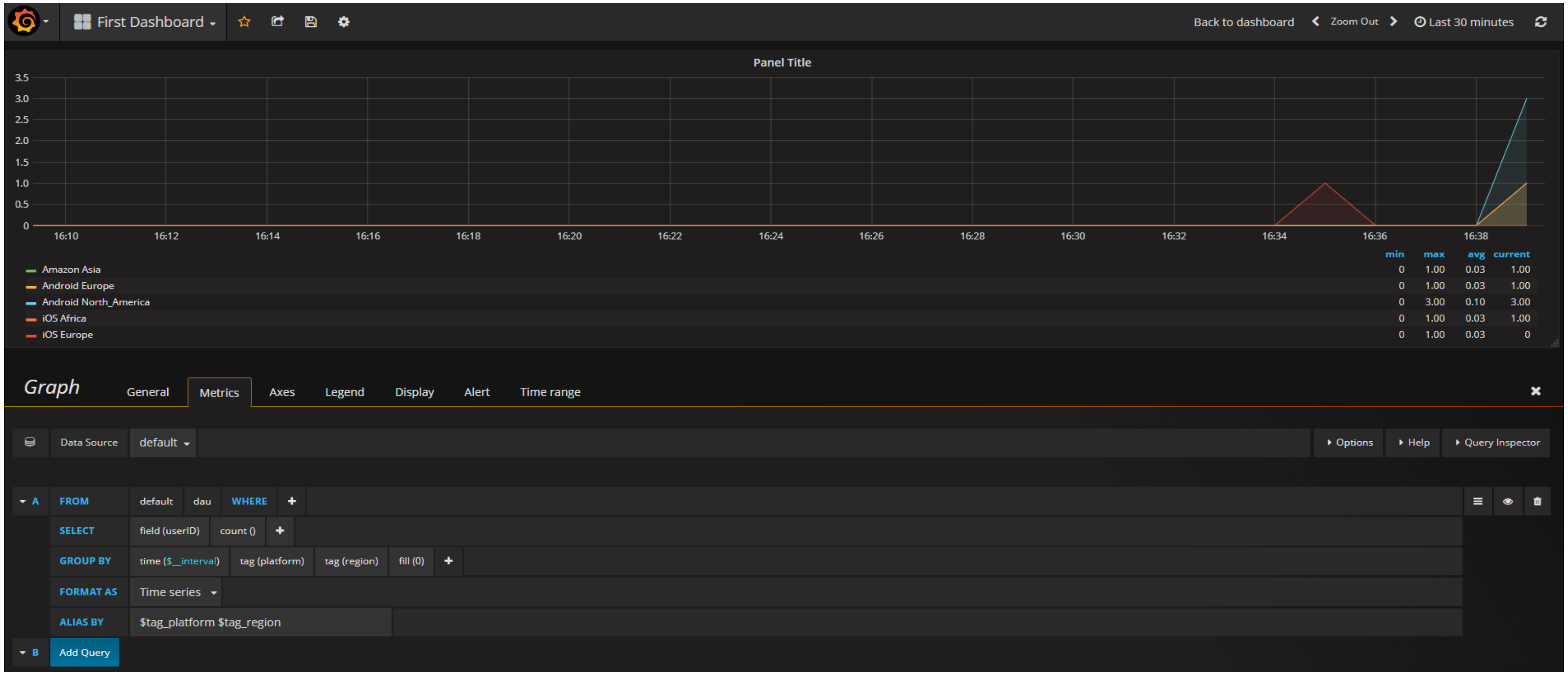
Torres, M. A. (n.d.). *App Monetization Statistics: Freemium vs Premium vs Paymium*. Retrieved from buildblog by thinkapps: http://thinkapps.com/blog/post-launch/paid-vs-freemium-app-monetization-statistics/

xPaw, M. (2017, November 9). *PlayerUnknown's battlegrounds*. Retrieved from SteamDB: https://steamdb.info/app/578080/graphs/

Figures:

1. (Marr, A simple way to prepare your business plan on a single page, 2017)
2. (xPaw, 2017)
3. Diagram from draw.io drawn by author.
4. Snippet taken from author’s own Grafana.
5. Snippet taken from InfluxDB virtual machine command line

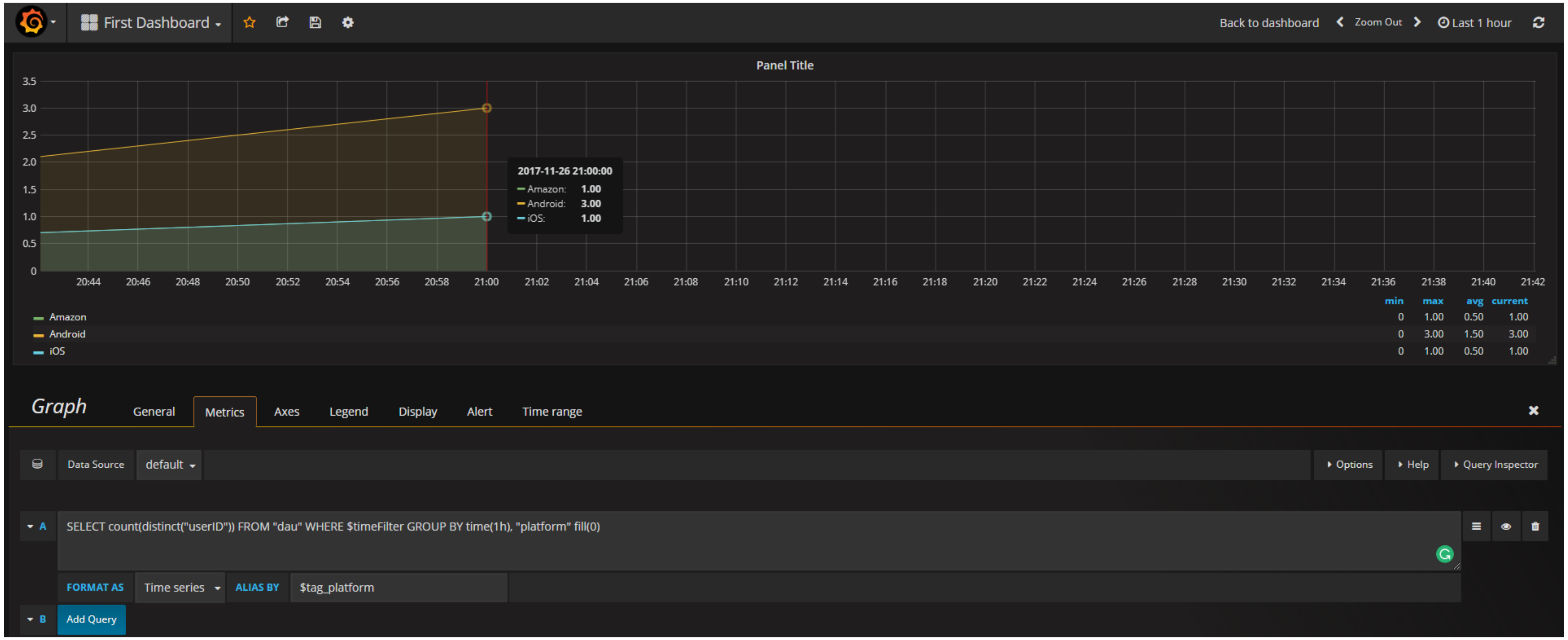
Attachments



	min	max	avg	current
Amazon Asia	0	1.00	0.03	1.00
Android Europe	0	1.00	0.03	1.00
Android North_America	0	3.00	0.10	3.00
iOS Africa	0	1.00	0.03	1.00
iOS Europe	0	1.00	0.03	0

```
Data Source   default

A  FROM        default  dau  WHERE
   SELECT      field (userID)  count ()
   GROUP BY    time ($__interval)  tag (platform)  tag (region)  fill (0)
   FORMAT AS   Time series
   ALIAS BY    $tag_platform $tag_region
```

1. Test data results in Grafana



```
2017-11-26 21:00:00
Amazon:   1.00
Android:  3.00
iOS:      1.00
```

```
                                          min   max   avg   current
Amazon                                    0     1.00  0.50  1.00
Android                                   0     3.00  1.50  3.00
iOS                                       0     1.00  0.50  1.00
```

```
A  SELECT count(distinct("userID")) FROM "dau" WHERE $timeFilter GROUP BY time(1h), "platform" fill(0)
   FORMAT AS  Time series   ALIAS BY  $tag_platform
```

2. Result of the dau example