Open Source Data Analytics platforms for smart-city-scale real-time data visualization and analytics

Artem Grukhal

Abstract—The development of city infrastructure is growing in both speed and quantity. The concept of making the city a “smart city” is needed, so that the infrastructure development can be efficiently adapted to the growing population, and further directions can be predicted better, with less redundant work from the city officials. However, there are numerous challenges that one has to overcome in order to apply a concept, and make it work regardless of the size of the city, its population or any other parameter. Moreover, to efficiently handle the data from applications, there needs to be some sort of system that will help to overview statuses, visualize output and process changes in data flow in real-time. This report will concentrate on features such a system should possess to maintain smart city application, and outline the most important requirements for a system to be functional. It will look into some existing systems for large-scale streaming data visualization and analytics platforms and try to evaluate the working models against a scenario of Tartu city, which may contain numerous devices with varying data supply intervals. The advantages and disadvantages will be assessed against the above mentioned scenario, sketch the solution to the scenario, and finally discuss some fruitful conclusions that we may make out of this research.

1 INTRODUCTION

The concept of smart city was introduced rather long ago, however, only recently with the surge of technological advancements related to network bandwidth and latency this concept is being talked about more and more. A smart city is such a network of sensors, actuators and devices that may be overlooked remotely, and data stored externally. This is especially useful, considering the fact that with each year the urbanisation is growing, and the need for efficient management of transportation, energy management, crisis response and business requests to city officials is growing proportionally. These systems may bring improvement to everyday quality of life of the population, reduced costs for maintenance of city infrastructure, faster response times to crisis events and more thoughtful planning of city development.

This is usually achieved with the help of Internet-Of-Things, which is also a developing topic in many different aspects like reducing costs for manufacturing, smart houses, etc. However, in such systems there have to be considered limitations of big-city-scale systems that may arise in the application. For instance, the distances that the data collected from smart city sensors will have to travel differ largely from the distances that the data in home automation will travel. Moreover, the size of the network may differ greatly, and more computational resources will be needed to process such data.

The key topic of this paper concerns the data analytics platforms. Essentially, they need to fit the criteria for a convenient work with such datasets, including visualization of real-time data. Mainly keeping in mind heterogeneous and non-uniform data, and considering limitations on the technical side as well. Later, these systems may be used to control data, process it, for example, with some prediction machine learning algorithms and store it elsewhere.

The research question that this report will try to answer is: what are the most important requirements for real-time data visualization and analytics platforms to be able to handle smart-city data and applications. It is done with an intention to prepare future works for developing such a system, having some background knowledge as supporting information to planning of such a system and guiding through the development cycle.

We have a Tartu city, where there are several systems for collecting different measurements for further processing and application. Among the ones that are particularly interesting to discuss, in terms of constantly updated data are: Location of public transportation, city border counters for passing cars, ticket validation statistics and information regarding electric bicycles location and usage. Since there also exists a requirement, that all the data should be collected to a single database, we also have a limitation that has to be taken into consideration when evaluating the current solutions.

This paper aims to focus primarily on work suggested by M. Farmanbar et al [1]. It will use the structural suggestions to such a system, findings related to the context of a scenario of several other European cities. It will try to apply the same system discussed to the context of the Tartu city scenario, discussing features and characteristics of such a design. Next, overview the system that was developed and extract some useful thoughts from the developed solution, that are relevant to our scenario, and finally, draw conclusions regarding the symbiosis of this work and our scenario.

2 BACKGROUND WORK

The first, and one of the most important background works[1] in this paper, is about developing a smart city dashboard Triangulum. It will be extensively referred to in this report, since the system that is being developed
has many hints on what the requirements should be for answering the research question. It is developed under EU Commission’s initiative, and it’s main feature is to provide adequate visualization from large streams of data, and using a regular web-browser be able to provide user-friendly interface for navigating around visualizations, and providing real-time response to changes.

The whole system is built in 3 layers: Data layer, Application layer and Presentation layer. Data layer is one of the main concerns in this case, since it has most challenges connected to it. It needs to collect data and store it efficiently, so authors suggest their way of implementing such manipulations.

Further we may see the system overview, it’s architecture from the software perspective, and the authors introduce main functionalities of the developed system: Filtering, Aggregation, Time series analysis and Visualization.

The datasets and scenarios are different, hence the the authors try to align the features to each scenario. The first in the list is the public transportation dataset, which contains GPS stamps together with timestamps, so there is a possibility to track buses on the map real-time. The second dataset is related to parking management, by showing empty parking spaces on the map. Third scenario is collecting household energy consumption, it contains only timestamps and power consumption, hence will only require plotting, without any map. Next scenario is a central power plant, to assess the efficiency of a power plant that works mainly on renewable energy with the data about the electricity consumption of the devices on a plant. Last scenario was regarding electric assist cargo bikes, to display all sorts of different information with the intention to further analyze it.

From the technical perspective: Apache Web Services, Kibana, Logstash and Elasticsearch were used for the server side technologies, together with Python being used as a primary backend language. Front-end was mostly operated by plain JavaScript and jQuery frameworks, together with Plotly and Leaflet libraries as visualization tools. The backend and frontend were communicating through Representational State Transfer(later refered to as REST/RESTful) requests, backed by the Flask framework.

Apart from the main research paper that this report will rely on, there are also supportive works that may help with evaluating and designing a potential system. For instance, work by L. T. De Paolis et al [2] suggests a simple IoT system, that is collecting data from sensors using the Thingsboard platform and uses Spark Streaming for later analysis and computing of the data. Even though the approach for using Thingsboard makes the research paper too specific for evaluation and comparison, the report could look into a possible way of utilizing Spark Streaming framework in order to process the data, that will be stored to a database.

Another point that the Thingsboard system may hint at is the usage of various protocols for data communication in systems. For instance, Kafka is fault tolerant messaging queue system that may help to efficiently organize data flow, and JSON files for data transfer make sense as well, since this is a universal format that many current frameworks understand.

Another work suggested by A. Protopsaltis et al [3] may hint on the presentation layer, as to how to present the data collected, what may be the proper charts to build, and help to identify which libraries one might use for developing. This paper gives a general understanding on how the data is going to later be used, and in which format would it be better to visualize.

Based on this work later we will discuss the application of this system to different scenarios from Tartu city, and analyze the features, if such a system is applicable universally, or if there is space for additional considerations.

3 Detailed description of Tartu city scenario
Tartu is a modern city, that has several opportunities for using the data generated in this city for improving the transportation infrastructure and for more efficient city resources planning. In order to further understand nuances of applying various existing platforms to the main scenario, there have to be specific requirements met, both from technical perspective and from the data variety. The next subchapters will talk more about some specific data examples and about some technical requirements that are placed onto the existing system provided by Telia for Tartu city on March 2018.

3.1 Smart Lighting
The current trend in the world is to reduce the electricity consumption when it is not necessary, and smart lighting can help in this aspect providing the illumination to some specific areas, where it is actually needed. This may include, but not limited to pedestrians passing by, cars driving nearby or in general the venues that would require light assistance.

This approach allows for controlling the light sources in different ways remotely, and due to such exposure, the possibility to assess the technical state of light sources also exists. The light sources can further signal in case of any malfunctioning and allow for timely response from the city officials.

3.2 Smart Bike Sharing
Smart bikes are already actively used in Tartu, and are popular among the locals. The opportunity to utilize the data gathered from the bikes and dock stations is promising, and can provide the authorities with a better overview on the mobility in the city and help making maintenance of such systems more efficient.

In particular, for example such bikes can be tracked in real-time for any malfunctioning, and data collected from dock stations may hint onto most busy places in the city, in order to improve the accessibility for such bicycles and promote preferred way of transportation, that is also a greener option, than conventional public transport.

3.3 Energy consumption
There are several pilot buildings in Tartu, that have energy consumption data collection devices, and provide accurate measurements, that may later be analyzed for if there can
be improved anything regarding the building, or show any leakage of energy and possibly point to the source of the problem.

Such information is valuable, since it may help to reduce the amount of energy wasted, and therefore reduce costs that may occur due to such errors in the energy flow systems inside the buildings.

### 3.4 Smart Public Transport

Buses in Tartu can also be utilized for a variety of useful analytics, and in general, be improved. Data from buses, for example, may hint at better scheduling options, so the buses are at the correct location on the most desired time slot. Another example where it may be used is to track the position of buses, so the applications for residents can be created, that help to estimate the bus arrival time and help with their commute.

Such data may also be helpful for private companies, to build their business around the population transportation, for example choosing the best locations for selling specific goods, or providing services, that may help with the commute.

### 3.5 Smart Open Data

The option of open smart data would allow companies to utilize other important information from the city, and incorporate it in their own services, or use it for analytics. Such a system would allow for saving costs on the stage of business planning and to make statistical research ready on request.

Besides being useful to developers and companies, the data contained within such a system is also trustworthy and follows the regulations provided by the city, so once it is available it may be used as it is.

Lastly, it is worth noting that there is also a system for counting vehicles that leave and enter the boundaries of Tartu, and it may be used for either general statistics, to estimate the amount of transport that is present within the city, or to estimate which roads are most busy, to help with effective planning of road infrastructure.

### 3.6 Smart Maintenance

On the other side of the system, as several already described scenarios would be able to provide timely notifications to the government regarding any service needs, or about any malfunctions that occurred in devices that the city uses, the system may also have a helpful tool for the maintenance workers, to enable them faster access to the information regarding the scope of the issue and thus will help to optimize the personnel reaction to such events.

In general, it may help to improve reaction time to different events, and therefore improve the living standards within the city.

### 3.7 City Dashboard

For Tartu City one of the main requirements is the general dashboard, that would allow orchestration of various services within one system, and having possibility to add them as the system develops. Dashboard would contain general information about the city state in real-time and provide possibility to control the devices that are used, without having the need to develop a custom application for each one of them.

This is the central part of the whole application, and therefore has to be as generalistic as possible, in order to provide possibility to incorporate other services, should they be developed. Most important part as well, since this is the starting point of the whole system.

### 3.8 Technical limitations

Currently, the data from many devices is collected to a single system - Cumulocity[4]. This system includes device management, with the possibility to have some basic analytics on devices, which would mostly include simplistic overview of device health, or the sensors values, that are sent to the system. There are several protocols that Cumulocity can work with, among the ones that are particularly interesting are REST and MQTT (abbreviation for Message Queue Telemetry Transport), that can be used for connecting devices. For exposing the data that is flowing in, only HTTP RESTful requests are supported.

Cumulocity allows to both see the data that is incoming, and to control the devices remotely, if such possibility exists. There are possibilities to manipulate the incoming data and transform it as it arrives in real-time. In general this is a very good tool, which can provide the analytics for the health of devices, and other important parameters, that are device-related.

Yet, there are limitations for using this platform, and one of the main ones is that this system is very hard to customize, especially, when the business related statistics is concerned. The platform is not user-intuitive to a non-developer, and since this article aims to discuss a system, that regular Tartu residents, or officials could use, this is not very suitable to use for real business analytics and planning.

Another limitation that is present, is that since only RESTful requests can be used to fetch the data present in the system, there is basically a single point of entry, and the platform that is to be developed should take into consideration this fact to develop it’s architecture accordingly.

### 4 Application of the Triangulum System to Tartu Scenario

Triangulum system is made to support the heterogeneity of data, and is expected to be able to accommodate many services to fit into the single dashboard. These services are also grouped by the city, where they belong, so with this chapter we may look into how the services could be integrated, and to see if any flaws can be found with these integrations. The architecture of Triangulum system can be seen on figure 1.

According to this schematic we can discuss now, how would we apply the given requirements and subsystems for Tartu city to an existing architecture.
4.1 Data layer
Starting with the data layer, Triangulum has a specific protocol of how to accept data. First the data intake form has to be submitted, that allows to agree on common data-format, that both the sending part and receiving part will operate with. Later the Triangulum developers will use this form to create a separate adapter for the system, that will be responsible for supplying this data to Python backend. Considering the fact, that by default Tartu city data is collected through Cumulocity, and is exported through its own API, that may provide RESTful responses, it has the possibility to export the data as JSON objects. This can be used to feed the data to Cloudstash as it is currently done in Triangulum, and follow it's architecture.

Essentially, key takeaway is that the Triangulum system does not provide flexibility to incorporate other services, as Cumulocity does for devices. Triangulum can only process data from external source after a specific development is done.

4.2 Application layer
This is the part where main processing is done. Since the data intake and transforming is done on the previous step, the main concern for application layer is processing the data and making objects that will be transferred to presentation layer for the graphical or statistical representation. At this step, it is worth noting that the data should already be unified during the data layer processing in such a way, that it can use the tools provided by Python and the frameworks that are used additionally. The data unification is mostly done during the development of an adapter, but since Cumulocity also has a possibility to export data in a distinct format from what it receives, there is also a possibility to send data already suitable for processing, so that there is less load logic-wise to an adapter.

4.3 Presentation layer
On this layer finally the libraries provided by the front-end are utilized to show the analytics that were pre-compiled by application layer, and to provide the internal navigation through various sections. Here Triangulum system can be utilized to have a separate tab for Tartu and later provide the information about services it provides with the cards, as it is displayed on figure 2. Since the presentation layer uses all the needed libraries for dropdowns, charts, time-series analysis and geospatial data, the development for services and datasets that the Cumulocity will provide seems to be sufficient in provided scenario. Moreover, on the presentation layer the data question does not play a big role, since it is mostly handled on data and application layers, so it is safe to say that this layer is scenario-agnostic.

However, there are some situations in this system in general, which may not be covered by the Triangulum system, but have to be somehow processed in Tartu city scenario. Particularly, Smart Maintenance and Smart Open Data seem to fall out of scope for a system like Triangulum. Hence, this will be looked into closely during the evaluation chapter of this report and on the discussion about the potential system to be developed for Tartu.

5 Evaluation of requirements met
Having described the possibilities of implementation of Tartu scenario on Triangulum platform, it is now important to evaluate feasibility of actual implementation of the features that would be available.

Starting off with the City Dashboard, by default its purpose is to organize the workflow in the system, and it is handled rather well, as it allows for having sub-sections for different services. The only concern, that is present in this case is the data protection, since some services have to only be accessible privately by the city government, and functionality for this was not clearly described in the Triangulum paper.

Similar issues would be present for Smart Maintenance, since this will be the service specifically for the workers of the Tartu maintenance companies, there should be a proper authorization flow, that only the users with the correct roles have access to the system. Otherwise, some developments for this system are still needed, since, for example, the functionalities of task assignment or the notification of a new faulty device are not described for Triangulum system, and need to be addressed separately.

Smart open data is another problematic point for Tartu city scenario applied to Triangulum system, since this feature does not exist in Triangulum. Mainly this service would require additional APIs that would be exposed upon request and give out specific sets of data, that can be further extended or modified accordingly by the users of the potential platform.

For Smart Public Transport, Energy Consumption metrics, Smart Bike Sharing and Smart Lighting most of the
features are already present within the system. Basically, these require visualization tools, which are present in abundance in the Triangulum system, and the data would be transformed accordingly so it fits the graphs already during the Data layer processing. Some examples for these functionalities are also present inside the Triangulum system, such as Public Transportation data, Smart Bike Sharing data and Energy Consumption metrics.

For the Smart Lighting, however, there may be issues, since this is a new system, which should signal the luminosity level and provide updates for bulbs in real-time. However, a similar solution regarding Parking Lot management in the Triangulum system is implemented. The capacity of parking lots is updated based on live information, and adapting it to Smart Lighting needs of displaying real-time luminosity level would not be a big issue.

The last concern about the system in general is the lack of actuating interfaces, which may be required for Tartu city. The Smart Lighting may get use of this functionality for controlling the light bulbs, so that they adapt to the lighting level automatically. Another example where this functionality would be utilized is manually disabling faulty dock stations in Bike Sharing system, or sending messages to Public Transportation systems, without the need to use third party services to transfer messages.

In general, Triangulum seems to provide only visual-analytical information about the system, however even though the Tartu city scenario is limited currently, there is a room for incorporating more services. And with this requirement in mind, the Triangulum system as it is would not be suitable for the chosen scenario. Adapting the Triangulum system for one’s needs may take too much effort from the developers side, and some major points are either unclear, or are non-existent whatsoever.

6 Discussion on possible optimal system for Tartu scenario

In theory, the developed system would have a rather flexible data ingestion as it is present in Triangulum, and more tools for customizing the dashboard. The main considerations that can be picked from the application of Triangulum system and its evaluation can be:

1) Data ingestion for heterogeneous datasets
2) Authorization and user access
3) API to allow for data to be processed before it gets to Presentation layer
4) Possibility to add actuating logic
5) API for further data export
6) Visualization and analytics

Data ingestion is currently operated by Cumulocity platform. This is raw data that is collected from devices, and it only provides some basic aggregation and analytics, which is not sufficient to be used in business logic; its own analytics engine is limited, and the export of data is either performed over REST requests, or by exporting CSV files. Hence, as the data arrives at the application layer, some interfaces can be developed in order to extract the most important parameters and organize each dataset according to its own sub-scenario.

Cumulocity platform typically exports information through HTTP/HTTPS requests. It is the easiest way provided by developers. Since this data is initially raw, there is a need to mark the datasets that would be unavailable to the public at this stage, but an even better way would be to isolate the channels that fetch and later expose this data. For this matter, a RESTful application seems to be the best suit. It is most commonly used in development of web applications, and it allows for easy and secure authorization flow through access tokens.

As the data is transformed and secured, some manipulations need to be done in order to aggregate it properly, sort and add supportive values to the final version. These may be certain divisions that would be displayed in charts, calculations, if any are needed, timestamps when the data was received etc. Since the Presentation layer is mostly used for already representing the final version, it will dictate the format in which it will expect the data, hence the choice of visualization libraries will play a big role in this case.

Currently the city of Tartu does not provide the actuating possibilities. Cumulocity is mostly used for collecting and visualizing the data, however, the control for at least Smart Lighting systems may appear in the future, hence to be prepared for this, there needs to be a certain consideration in the system’s architecture. Generally, this means allowing requests to come from the Presentation layer and transferring them to Cumulocity for specific endpoints. This also needs securing the incoming request, so the authorization is still in place and no external person may influence the system workflow.

As for the data export, the main interest lies in data that has already been processed for statistical purposes and is available to be used by a third party, which will have access to it. It should be further decided in which format the data is to be shared, but as this information is not available yet, the system should be ready for it to be in either JavaScript Object Notation (later refered to as JSON), or Comma-Separated Values (later refered to as CSV), or other widely-used format, that could be used to export data. Another consideration that might limit the data flow is the fact that it would be real-time, so there should be error-proofing present, that will help to avoid inconsistency in data, or data gap. For this reason, Spark Streaming platform may be utilized, to be able to emit data, that it would be provided. There is a work mentioned in the beginning of this report [2], that may help to build a system and it may be used for a reference.

Finally, visualization and analytics of the Presentation layer, that will be using already processed data from the backend need some specific requirements as well. Mostly the layer has to support rendering of the up-to-date results that will be fetched and aggregated. Eventually, there has to be an interface that would enable the system to send messages constantly to update the graphs that run on live data, and these graphs have to support such functionality. Specific graph types that would be taken can be looked upon from the technical implementation of the Triangulum system, but can also be found externally, in case the libraries to be used are not supported by the system to run with live data.
7 Conclusion

This paper has given the background work for development of a real-time visualization dashboard with specific requirements that will have to be fulfilled in order for the system to run smoothly and to provide with the analytics that would be interesting for the end-user. The scenario for Tartu city is described, together with its limitations and plans for future works, and the attempt to apply the Tartu city scenario over the existing solution of Triangulum is made.

As a result, even though Triangulum answers to most questions regarding how to handle data heterogeneity and the visualization tips, due to it being custom-tailored for specific applications, it may be difficult and in some places impossible to adapt it to the scenario that is outlined for this research, and therefore is not suitable in this case.

However, the main ideas help to signal and outline the most important requirements for a system are presented. The main takeaways from this research that are needed in an optimal system for smart city dashboard are:

1) Extensibility of the dashboard - meaning that more services can be added upon request
2) Support for actuating interfaces - at some point there will be a need to control the data, and not only to analyze
3) Visualization libraries that have support for charts, maps and graphs
4) Clear user authorization flow

These ideas are later evaluated against the scenario and explained in the finalizing chapter of this report. These requirements are also used to sketch the considerations for building a custom applications with its own architecture that would suit the Tartu city, and they are discussed to prepare the future developers for work that is to be done.

References