Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Master's Programme in Software Engineering and Digital Transformation

Kimmo Flykt

Data source integration to information credibility assessment system

Examiners : Professor Ajantha Dahanayake
Professor Naofumi Yoshida

Supervisors: Professor Ajantha Dahanayake
Researcher Naofumi Yoshida

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT
School of Engineering Science
Master's Programme in Software Engineering and Digital Transformation
Kimmo Flykt

Datalähteiden integraatio informaation uskottavuutta arvioivaan järjestelmään

Diplomityö
2019

59 sivua , 11 kuvaa, 4 liitettä

Työn tarkastajat: Professori Ajantha Dahanayake
Professori Naofumi Yoshida

Hakusanat: tietolähteet, sensorit, rajapinnat, integraatio

Nykypäivänä tiedon hakeminen ja tuottaminen on helpompaa kuin koskaan ennen. Samalla kuitenkin lieveilmiöt kuten virheellisen sekä valheellisen tiedon levittäminen on helpottunut. Tiedon todenperäisyyden tarkistaminen on entistä vaikeampaa sillä tiedon määrä on monesti liian suuri yksittäisen ihmisen hahmottaa kokonaiskuvaa tai ristiriitaista tietoa on liikaa johtopäätösten muodostamiseen. Tämän diplomityön tarkoitus on tutkia teknistä näkökulmaa järjestelmälle, jonka tarkoitus on liittää tiedon todenperäisyyttä arvioiva järjestelmä reaaliaikaiseen tietoon. Tutkimuksen pohjalta luotiin prototyyppi, joka kerää valituista lähteistä tietoa uutisista sekä kokoaa sensorien keräämää informaatiota tukemaan todenperäisyyden analyysiä. Järjestelmä esikäsittelee kerätyt tiedot ja luovuttaa ne eteenpäin tiedon todenperäisyyttä arvioivalle järjestelmälle.

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Master's Programme in Software Engineering and Digital Transformation
Kimmo Flykt

Data source integration to information credibility assessment system

Master's Thesis
2019

59 pages, 11 figures, 4 appendix

Examiners: Professor Ajantha Dahanayake
Professor Naofumi Yoshida

Keywords: Data sources, sensors, Interface, integration, credibility

The present day searching and finding information is easier that ever before. However at the same time side effect like misinformation and false information are more common to come across. Finding out authenticity of the information is more and more difficult because amount of information is too big or there is too much contradiction between information to find out what is the big picture. This purpose of this thesis is to provide technical viewpoint for the system that connects information credibility system to real time information flow. Based on the research, prototype of the system was created which collect news data from selected sources and data from selected sensors for supporting credibility analyse. System preprocesses collected data and gives it forward to credibility assessment system.

ACKNOWLEDGEMENTS

First I would like to thank prof. Ajantha Dahanayake and prof. Naofumi Yoshida-san for making this possible. I would also like to thank them for helping and guiding me trough this project. Additionally, I would thank all friends from Finland and from Japan for supporting and providing help while I was working in this project.

I had awesome five years in the Lappeenranta University of Technology and had great privilege to meet many great people. Some of these people became great friends that I will appreciate trough my life. I also had great opportunity to do my master's thesis and really warm welcome in Tokyo at Komazawa University that provided guidance and help during my stay there.

Lastly I want to say thanks to my family for supporting me trough all these year and helping me every time I was in need of it. Without stable relationship and trust that I won't be alone with my life this part of my life could have been much harder to live trough.

August 2019 in Shinjuku, Tokyo, Japan

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

IoT	Internet of Things
API	Application Programming Interface
WWW	World Wide Web
SOA	Service Oriented Architecture
SensorSA	Sensor Service Architecture
IT	Information Technology
SWE	Sensor Web Enablement
WSN	Wireless Sensor Architecture
pub/sub	publish/subscribe
RSS	Really Simple Syndication
QoS	Quality of Service
SDK	Software Development Kit
GUI	Graphical User Interface
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
CRUD	Create, Read, Update and Delete
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
XML	Extensible Markup Language
JSON	JavaScript Object Notation
RFID	Radio Frequency Identification
DSMS	Data Stream Management System
CEP	Complex Event Processing
NLP	Natural Language Processing
E-DP	Event Driven Programming
E-D	Event Driven
OOP	Object Oriented Programming
URL	Unique Resource Locator

1. INTRODUCTION

This thesis is part of the research article at Komazawa university at Tokyo that tries to find out how news information assessing systems should be operating and what kind of use cases it can be used. Main purpose of the article is to provide information about how this kind of credibility system could be implemented and how internet of things (IoT) devices can be used together with social sensors and news sources to assess credibility of distributed information. The two viewpoints of the article are data science and software engineering. Software engineering focuses on implementation and its challenges. Data science viewpoint tries to find ways how social sensors can be included in the credibility assessing process.

The purpose of this thesis is to research and provide software development viewpoint for the researched credibility assessing system. For example one of the major point of this side of the research is to find out how connection between data sources and designed system should operate and what components and techniques make it possible. These requirements not only affect to application programming interface (API) of the design but also how credibility functions can be implemented and what features they can have. A prototype application is made based on findings and signs of the article. This prototypes major purpose is to demonstrate how the design works in real world and show what are strong and weak points of proposed design.

The area of this research is continuity from earlier researches about news information credibility assessing at Komazawa university. Earlier researches from this field have resulted method for classifying intention behind news headline [1] and sensor selection for credibility calculation system [2]. Next step for this field of research from the software engineering viewpoints are to find out how to solve scalability of experiments and generation method of matrix node graph proposed for intention classifying.

1.1 Background

With the access of World Wide Web (WWW) gathering information is easier than ever. However quantity doesn't always mean quality. Easy access to information makes information distribution also easy. Uncertain information, rumours and half-truths can spread easily especially through social media platforms [3]. Because of this information credibility is more important than ever. However, true or false comparing is not only problem in misinformation but it is also time critical problem if something happens before information in confirmed like in Mexico 2018. In this event 2 people ended up dead after group of people reacted to rumour in WhatsApp messaging service, before the information was corrected [4]. To prevent this kind of tragedies it would be good to develop tools to help identify rumour and misinformation [5].

One solution for this problem is to use specialist-knowledge domain in the shape of crowdsourcing. In this approach different people evaluate web contents credibility and share it to common knowledge portal that can be used by other web users to check informations credibility [6]. However this works as long as only intention of the person evaluating information is to share correct information to other users. Because humans are individuals they don't do this exactly same way every time. This make it possible to change intention for information based on the topic of information [7]. In order to stay objective good way to find out credibility of information is to hand out the task to the computer.

Figure 1. represents system architecture for credibility calculation system. In this system sensors are chosen to be used calculating credibility of information by providing unbiased source of data from events related to evaluated information. If the sensor is not suitable for providing evaluation data then it is discriminated from being datasource for that assessing process. The purpose of this system is to provide automated method for selecting appropriate sensors for the purpose from multiple sources available at WWW [2].

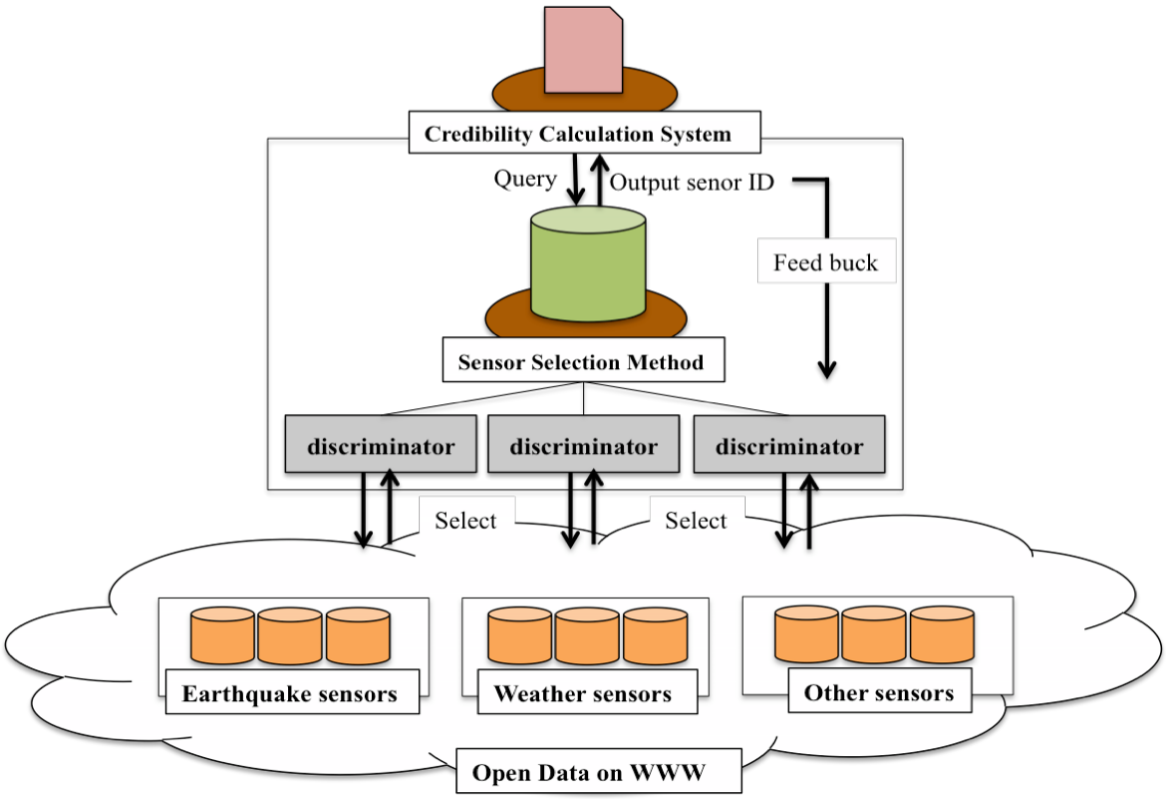

Figure 1. Selection and learning method for credibility assessing using sensor data [2]

In 2016 conference of "New trends in Databases and Information systems" new methods about information credibility calculation system for emergencies was introduced. The purpose of this method is to calculate information credibility by comparing target information with various information resources on World Wide Web and sensor data. This method can be used to calculate information for example related to natural disasters where there is reliable and objective sensor data available about event [8].

Important part of information credibility calculation is to know what kind of characteristics information has. One of these characteristics are intention. Reliable data from the source that doesn't involve human touch could be trusted to have only one intention that is share objective information about target events. However when humans are involved in information gathering process there could be secondary intention behind that is implicit. In order to find out what are the intentions behind information matrix node graph for credibility assessment system was proposed. This method will classify information and

show what are the intentions. With this information accuracy of credibility calculation can be increased. Figure 2. will show basic how to implement this method in credibility assessing system [1].

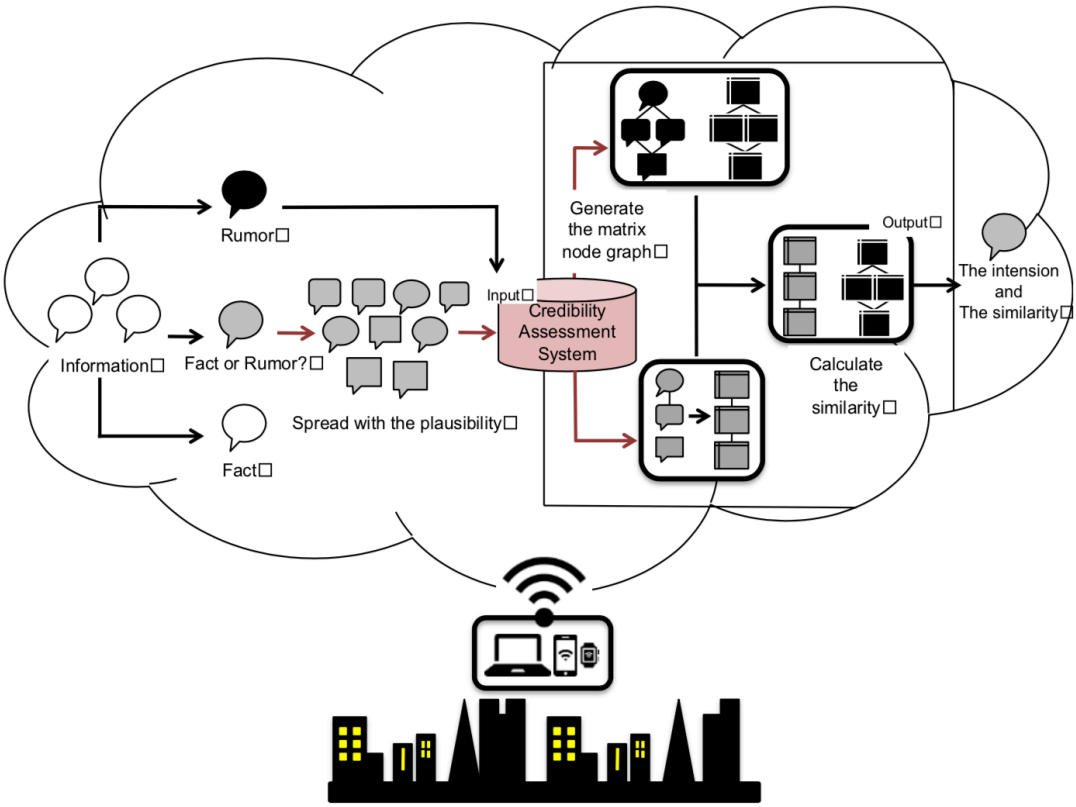

Figure 2. The Matrix Node Graph and Credibility Assessment System. [1]

1.2 Goals and delimitations

In this thesis attempt is to find out what are the best ways to provide data to target system from multiple sources. Literature review is made with the target system in mind to find best solutions for prototype system. The outcome from this thesis should be design and architecture plans for sensor data source integration to credibility assessing system and prototype system of based on the design. Outcomes from this thesis should lay groundwork

for the future where datasources with different types of output are required to communicate with credibility assessment system.

This thesis focuses only technical side of the credibility assessing system. This does not include possible completed modules fit for the system found at literature review. Also excluded are design of the social sensing module for the system and data modelling for it. Credibility functions necessary for information assessing are not included but are taken in account in design of data storing. In this part cooperation and consulting with other researcher included in same article is necessary in order to find out relations exceeded topics and researched topics.

Hypothesis of this thesis is that processing speed of the assessing system can be improved using data integration. This hypothesis is based on educated guess that with proper usage of tools and techniques process can be improved. To support this it is assumed that needed datasources can communicate via same infrastructure and protocols that assessing system uses. It is also assumed that the topic area of the news articles used as a data source is limited to only one or few.

The research :

1. What are best ways to integrate outside data sources to information credibility assessing system?
    i. How different kind of sensors can be integrated to system as data sources?
    ii. How different news sources can be integrated to system?
2. Can the information assessment process speed be improved by using data integration?

This thesis is part of the article and only focuses on technical side of the proposed system. Author works as a part of research project and is responsible of technical design and implementation of system in form of prototype with other authors of the article.

1.3 Research methodology

In this thesis theoretical part is literature review. In this part selected technologies are presented and explained. The purpose of this part is to gather knowledge what are the technologies, methods, models and other aspects in this field of study. This supports the empirical part of the thesis providing insight about researched topic and finding.

At the empirical part of this thesis presented theories and technologies are put in use to provide solution for presented problems. Solution in this thesis will be presented in form of prototype. The empirical part in this thesis is done in collaboration with other authors of the final article which this thesis is part of. This means that integration of different functionalities of prototype are done by different authors but are part of the same prototype. The parts of prototype presented in this thesis are from the datasource integration viewpoint.

1.4 Structure of the thesis

The first chapter of this thesis is introduction. This part gives overview of the thesis providing reasons behind and background information about the research. Goals and delimitations are described after background information in order to define the domain where this thesis will be. The hypothesis is also presented with goals and delimitations. Research methodology will be explained to provide transparency about results of the research. Last part of the first chapter is explanation about structure of the thesis.

Second chapter of the thesis is about theoretical background. In this chapter finding from the literature review of the topic is presented to the readers. In this chapter there is two major entities that explain different parts of the research. First part is sensor integration where frameworks and designs of integrating sensors to systems are presented. In this case sensors are mentioned because they are one major datasource target in this research but

also because they provide variety and that way similar problems than other datasources in the WWW. Second part of this chapter focuses more to introduce techniques and methodologies that could be used in solution.

Third chapter of the thesis is about implementation. In this chapter methods, designs and technologies presented in second chapter are implemented in credibility assessing system and reasons behind the choices are explained. Related closely to the third chapter is the fourth chapter where results of the research are presented to the reader. Results are also explained in order to get better understanding of the outcome.

Fifth chapter contains discussion about the research. Discussion creates dialog about how the research succeeded and what kind of problems there is for future researches. In this chapter context is provided in the bigger picture in order to help understanding what kind of impact this research has in real world.

Sixth chapter is about concluding the research. The conclusion part summaries the research and gives reason for the results. The answer to the research questions and result of the hypothesis can be found from this chapter. References and appendixes are listed after this chapter.

2. THEORETICAL BACKGROUND

This chapter is literature review of aspects related to this research. The purpose of this chapter is to give overview what kind of technologies and methods are to be used. In the first subsection target is to provide insight about designs used in implementation. Later parts the focus is on showing what kind of techniques and methods are needed to complete these designs.

2.1 Datasource integration

When looking data integrations basics form we can notice that it is matching problem of data. Outcome of this is a collection of equivalencies between different real-world concepts. Most of the time in data integration evaluation is performed with binary representation. This means that outcome of evaluation can be divided to mach or no mach. However, many times with big data it is required to have boarded evaluation because true/ false is not best way to represent the data differences. "As a concrete example, consider the evaluation measures of precision and recall, common in the area of information retrieval. These measures test the completeness and soundness of a matcher outcome with respect to some exact match (or a gold standard). Precision and recall are based on a binary correspondence comparison, which requires binary decision making from the matcher side and a binary exact match" [9].

According to Maurizio Lenzerini problem with the data integration is combining data that is from different sources and can be in different format from each other. This means that comparing data can be hard. Also providing unified and accurate view of the data to user can be challenging. These problems are important and have to be taken account when designing data integration system for the real world applications[10].

When reviewing data from academic sharing practices, it was revealed that concerning problem is fear of misinterpret and amount of work that is required to from receivers to address questions and concerns. Same review revealed that problem with the lack of structured metadata will also cause similar problems. When the metadata is not structured it is hard use widely and will cause hinderance to the usability [11]. When working with the raw data it is important to have heterogeneous structure and semantics between sources. When the data is easily comparable storing and retrieving data is easier. Also operations are more cost efficient and will help analysing process. When raw data form all sources can be easily compared will the analysing require less resources compared to analysing non aligning data [12].

2.1.1 Service Oriented Architecture

Nowadays, the amount of data is increasing. Data is generated for all kinds of sources and collected for analysing or for other purposes. When talking about the data integration problem the real name should be data combining problem. “To design a data integration framework, we need to address challenges, such as schema mapping, data cleaning, record linkage, and data fusion .” When these problems are solved, unified view of information can be provided for user. Information is result from combining heterogeneous data from different sources and combining them together for increasing information value [13]. With the success of web services the importance of it as a part of the service-oriented architecture is solidified. With the attributes like scalability and agility it is great choice for integration platform. “The basic principles of Web Services, and of service-oriented computing in general, consist in modularising functions and exposing them as services that are typically specified using (de jure or de facto) standard languages and interoperate through standard protocols.” [14]

In the book “Service oriented architecture” (SOA) Mr. Georgakopoulos pictures architecture being based on concepts of “messaging” and “services”. When application is designed with this architecture service can be pictured as the logical manifest of some

physical or logical resource. These resources can be databases, programs, devices or other components used in the program. Service can also be application logic opened for the network. Messages are used to integrate these service together. This way they will loosely form system which are scalable and reliable [14]. In his article from 2015 Mr. Shi continues about SOA by telling it being "infrastructure supporting communications between services, and some connecting services are required."[15] Figure 3 shows overview of the system that is designed with service oriented architecture.

Using fundamental abstractions of SOA, we can discuss the following [14]:

- Services are used as a base for the computing architecture
- Applications and protocols use document centric message system to distribute information.

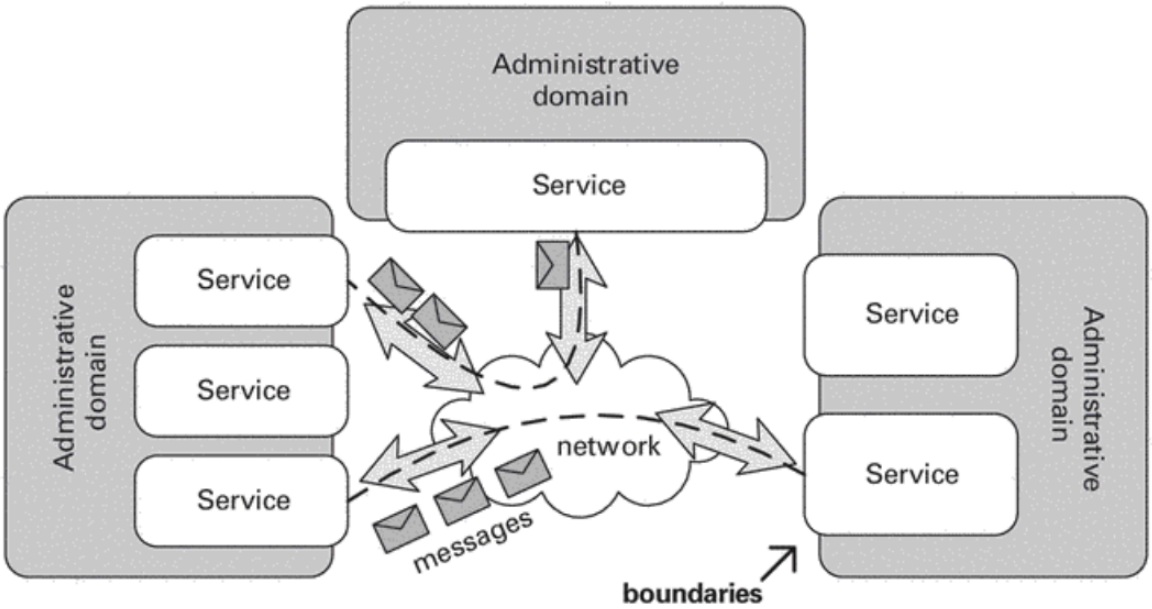

Figure 3. Service oriented architecture of a system. [14]

In order to have successful SOA based system standardising data is the most important factor based on several vendors. “The way in which data is formatted and stored —case sensitivity in names, use of dashes in credit-card numbers, etc. —needs to be fairly consistent for successful SOA implementations[16]." SOA itself is platform independent of programming languages. This means that there has to be understanding of used technologies to certain point. Requirement for this is that chosen technologies have to be able to communicate with each others. Standards can help this but first communication has to be possible overall [17].

2.1.2 Open Sensor Service Architecture

Sensor Service Architecture (SensorSA) is an open architecture which is designed for managing sensors in a network. Access management and information flow are main responsibilities of the architecture which provides platform neutral conceptual specifications and components for the management purposes[18]. If architecture is viewed from the information technology (IT) perspective the SensorSA operates like SOA by integrating different architectural styles together [19].

List of different architectural styles [19]:

- Remote invocation: Information transfers and function calls are requested by customer form trusted provider
- Event-driven: Information about events occurred for the sensor are send to broker services and from them distributed to registered consumers. Operating this way information about changes in the sensor network can be shared asynchronously. For example sensor is removed from network.
- Resource-oriented: Environmental information is from unique identifiable resources. This way operations can be limited per resource and encodings can be set to meet individual needs of network nodes.

Figure 4 shows the enhanced open sensor web architecture. Sensor web architecture follows Sensor Web Enablement (SWE) standard. Closer examination of the architecture reveals three basic layers. These layers are service layer, middleware layer and physical layer. Within the physical layer all the real world components and devices locate. Hardware, sensing devices and networking belong to this layer. Next layer from the physical layer is the middleware layer. In the middleware communication from hardware level is changed in the form that software can understand it. Layer itself is abstract but contains among other things drivers, gateways and other necessary technologies for communication. Main task for this layer is to make sensor deployment in the network easier. In this propose each gateway is designed to work with one type of sensing devices. With this gateway has to handle only homogenous devices and managing network becomes easy to manage [20].

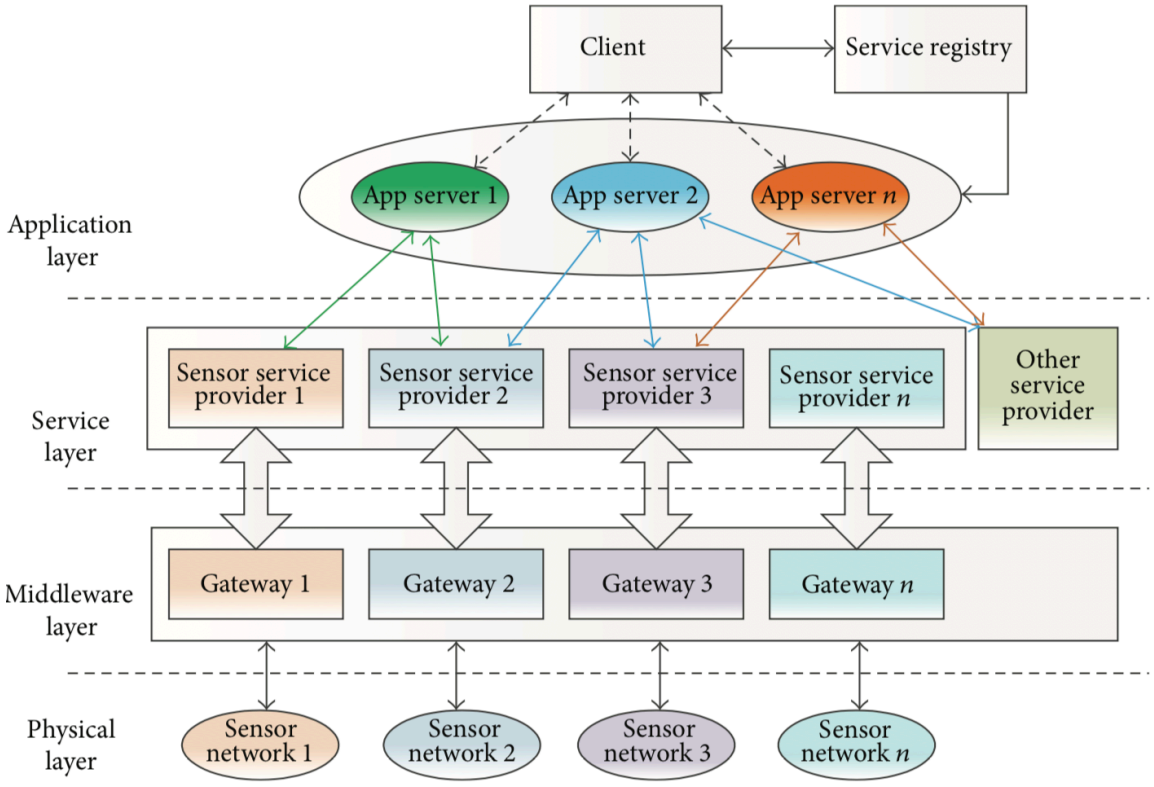

Figure 4. Enhanced open sensor web architecture based on application server. [19]

2.1.3 Wireless Sensor Network

A Wireless Sensor Network (WSN) could be network of autonomous wireless devices that are positioned spatially. Purpose of these devices is to observe physical or environmental conditions. With the WSN it is possible to have devices with different attributes in the same network. Combination of computation, sensing and communication devices are some of the possible attributes. Information from devices from the network is shared between other devices at the same network and information is used in calculations at distributed estimation system[21].

Wireless Sensor Network basic element is “node”. In every case of WSN with one or multiple nodes, every node is connected to at least one sensor. However typically there is multiple sensors connected to single node. In these kind of defector network there are multiple different parts that form the network. For example radio transceiver, a micro controller, associate antennas and support electronics. These components help to create wireless network infrastructure and they interface with the sensors for providing support. With the WSN there is also base station called sink which purpose is to communicate with the detector nodes in the network. Major part of the nodes communicate with each other wirelessly and can communicate with the sink directly or indirectly. Nodes also have capability to sense surroundings for information and store or pass it. Target for passed information can be other nodes in the network or the sink station. Also nodes can perform some computations of the information [21].

2.1.3 IoT Platforms

With the development of internet of things it is possible to create different applications that have ability to observe, sense and control physical environment. Currently typical case for the systems is to sense and actuate physical phenomena in relatively close proximity. For distributing gathered information systems rely on cloud based publish/subscribe (pub/sub)

infrastructure. Using this infrastructure it is also possible to control data and which users or external services have access to it. Even tough pub/sub system is popular choice for this kin d of use cases, there is still many questions about features necessary for specific system. For example middleware can have multiple different configuration but what is the best combination for IoT domain. Many system specific questions like number of connections, delays and throughput capacity need to have answers [22].

Current development of various networks and high diversity of information system has led to popularity with pub/sub systems. With the Pub/Sub middleware it is easy to collect and share information and this way provide value to system which it will be added. Pub/sub is also useful in many applications since it can be implemented not only in the web applications but also into enterprise applications. “For example, Pub/Sub middleware can be used for providing responsive stock information for users around the world, and it also can be used for Really Simple Syndication (RSS) aggregation while it is integrated into RSS readers.” In its basic form Pub/Sub middleware operation is based on events. Middleware reacts to changes in its environment and reacts automatically the way it is programmed. Usually this reaction starts data processing of the data from the publishers and provides it for subscribers. One of the attractive attributes of the Pub/Sub middleware is that it can “fully decouple the producers and the consumers in time, space and control flow”. With these features Pub/Sub middleware is gaining popularity in the cases where coordination and cooperation between distributed system is required in the integration [23]. Figure 5 shows overview of architecture using Pub/Sub principles [22].

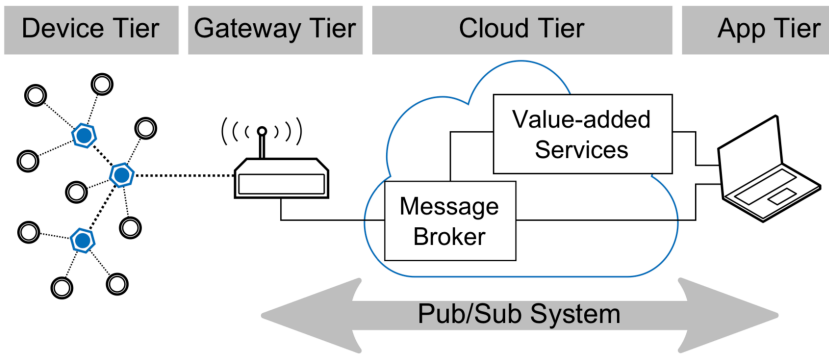

Figure 5. General architecture of a cloud-centric IoT platform. **[22]**

List of requirements for pub/sub in cloud-based IoT [22]:

- Message Pattern: Monitoring resource is required in all the cases. Symbolic addresses are used in matching producers and consumers together like done in the pub/sub pattern. Also point-to point messaging pattern should be supported in the system. This could be useful when matching address and contact information of particular object.

- Filtering: Many times target of interest is not the whole dataset but the part from the set. The capabilities of the filtering is determined by capabilities of the middleware. When filtering the results from the sensors best approach is to filter based on topics. However this is not optimal for every use case. Other desirable way for filtering the results is content-based filtering.

- Quality of Service (QoS) Semantics: When talking about messaging systems it is common event that sent message doesn't reach its target. In some cases lost message is tolerable but in other cases some guarantee of delivery is required. The middleware of the system should be able to support annotation of subscribers and messages while fulfilling set QoS standards.

- Topology: When looking at pub/sub systems in cloud that are IoT centric, the middleware should have support for some kind of centralised topology. With this based on the used filter broker nodes should be able to forward the messages. Brokers can be distributed to multiple virtual machine which should be taken into consideration in the context of this text. If the load to the broker needs to be reduced, producers and customers can directly communicate with the distributed topology. This however rises problem of finding particular sensors and actuators.

- Message format: When sensor hardware is heterogeneous it is really hard to find out what format for sensor data is going to be provided. Solutions for this made with pub/sub systems has to not take account the payload. The support for binary payloads should be provided on top of that it should be serialised so that frameworks sucks as buffers can be used.

2.2 Application Programming Interface

"Application Programming Interfaces, including libraries, frameworks, toolkits, and software development kits, are used by virtually all code." If we look at the internal APIs that are interfaces in the software project and public APIs that are for example Software Development Kits (SDK), Microsoft's .NET Framework, jQuery, Google Maps it can be interpreted that almost all code will be using API calls. API provides general tools and functionalities for programmer that can be used as a platform for custom functionalities. This makes programming more simple when at the start programmer already has some parts created for him. This also makes it possible to provide better compatibility and resource usage because resources are always accessed trough same protected APIs [24].

When designing usable APIs, designing process is similar to design process of Graphical User Interfaces (GUI). Characteristics of users are important to understand in order to be able to see how they are impacting to usage of the API. With the designs using principles from scenario at hand, API design is possible to create to respond its requirements. With more traditional way of reflecting implementation details produces less suitable results [25]. Good results with API designs are also possible to achieve if few guidelines are followed during design phase. Of course these are more general rules and do not guarantee success or best possible outcome but will help with the design. They also raise important questions that need to be solved in order to avoid problems in the future and make API more usable [26].

List contains some of the points for good API design [26]:

- An API must add more value for the caller. This could be functionality to achieve some task for caller.
- API design should not be too complicated. With minimal design it is possible to provide smallest inconvenience for the caller.

- With API design process context and its understanding are necessary for success.
- When designing API for general usage it should have less restrictions. On the other hand with specially designed API should have many restrictions.
- Design of the API should be done from callers perspective.
- Documentation for the API should be done before implementation.

2.2.1 RESTful Web Services

With the development of Web applications Representational State Transfer (REST) has become unofficial standard when operating resources. When talking about old Simple Object Access Protocol (SOAP) Web services used remote object. Functions in the SOAP are encapsulated and remote methods are utilised. With REST protocol target is only data structures and the state of transfer. Because of REST protocols great compatibility with Hypertext Transfer Protocol (HTTP) and simplicity has made it the choice when choosing technology for exposing data in Web applications [27].

"A great benefit of REST-based web design is the ability to use HTTP Headers to provide request context around each of the Create, Read, Update and Delete (CRUD) operations." When creating request to particular resource the answer could be HTTP, Extensive Markup Language (XML) or Javascript Object Notation (JSON) type. These answer types are chosen based on what kind of media is desired to transmit in the HTTP header. With this developers are able to create complex websites where programming API can be overplayed on top of the site. This will expose the API to users and can reduce the cost and complexity while at the same time providing method to have access to multi-format data related to site [27].

While REST looks to be dedicated technology but it really is not it. It is more architectural style for designing distributed network applications. It contains six principles that help it to create applications that are scalable, visible, portable and reliable. When looking at the theory of REST it is apparent that it is possible to do with almost any network infrastructure or transport protocol.

The six principles of REST [28]:

- Client-server: Client and server are separated and can be evolve and/or expanded independently
- Stateless: Communication between client and server should be stateless.
- Layered system: The can be multiple layers between client and server like middleware and gateways. Modifying and developing layers should be transparent.
- Cache: Clear declaration of cacheable or noncacheable should be made. Caching resources should improve performance.
- Uniform interface: All the interfaces should behave and look similar between client and server. This makes developing and designing easier.
- Code on demand: Allowing client to download code for executing functionality on demand. This can be for example Java applet.

2.2.2 Data Streaming

When looking at the current state of information science and technology in general it is apparent that the amount of data is growing. The volume of the data and its complexity are presenting new challenges to solve. It is more and more common to have sources of data that continuously produce it. Great example of this is Radio Frequency identification (RFID), sensor networks, telephone records, bank transactions, watching videos from internet and other similar technologies. All of these examples are called data streams.

When talking about data streaming the meaning is to describe “an ordered sequence of instances that can be read only once or a small number of times using limited computing and storage capabilities”. Characteristics of these sources are being open-ended and data moving in high speed [29].

When comparing data streaming to old ways of data transfer the question about differences arises. Compared to conventional relation models the key with data streaming is that data streaming model does not rule out the usage of data that is stored with conventional relations [29].

List of relevant differences between data streams and relational models [29]:

- Data elements in the stream arrive online.
- Data can arrive in any order across stream or multiple stream. System does not have tools for controlling this.
- Size of the streams can be limiting factor.
- Usually streamed data is not stored in the memory. After processing of arrived data it is normally discarded or saved. This is done because memory size is relatively small compared to data streams.

With the characteristics of traditional database management system they are geared towards one time queries run against finite stored data sets. However in the modern IT environment this is not the case every time. Many applications require support for queries run against continuous unbounded streams of data. These kind of application can be for example sensor networks, financial analysis, network monitoring and manufacturing [29]. For the data streaming purposes traditional data wearying and presentation of the results are not effective. If the traditional management would be used many problems would arise for example delays would grow or database would have to store large amounts of data that is no necessary for the purpose. In order to avoid these problems adequately Data

Streaming Management System (DSMS) is needed to handle the data flow. However there is still one question to be answered after need for DSMS is recognised. Does the system need support for storing data for later usage like analysing for decision making or is only real time data management enough. This question is called Complex Event Processing (CEP) [30].

List of different DSMS projects [30]:

- The Aurora
- Borealis
- STREAM
- Cougar
- TelegraphCQ
- Infosphere streams
- Microsoft StreamInsight
- Esper

2.2.3 Really Simple Syndication

"Really Simple Syndication (RSS) is an XML-based (Extensible Markup Language) content-syndication protocol." With the RSS websites have a way to share their content with others. As a format RSS allows information from the use to be aggregated from internet sources. Such sources can be for example e-mail, web logs, news feeds, etc [31].

When RSS technology is used the owners of the website are able to change online delivery method from PULL to PUSH. Between these two methods the key difference is the party that initialises the delivery. When websites are using PULL model they are in the passive role. They are just waiting for users to visit them after initialising search. The problem with this is to attract users interest from the big number of similar websites providing same or similar content and/or services. If the website is using PUSH model for the content delivery they are operating in the active role by sending information to potential users about updates. When user has selected RSS feed in his reader, there is possibility to create long lasting relationship between user and feed provider. This is great advantage in the competition between websites [32].

2.3 Natural Language Processing

"Natural language processing (NLP), is the attempt to extract a fuller meaning representation from free text." With this software tries to figure out from written language characteristics like what, whom, when, where, how and why. In order to figuring answers to these questions NPL uses linguistic concepts. These can be parts of speech like nouns, verbs or adjectives. Also grammar structures like noun phrase or dependency relations like subject of or object of can be used for NPL analyse. When using NPL the software has to deal with understanding context and relations from the text like anaphoras or ambiguities [33]. The main targets is for computer to understand written text in a "natural" way. This means that the closer computers understanding is to humans the better [34].

When taking closer look at NLP extracting semantic relationships from the text transpires to being important. Human language has many different ways to combine text into information and their relationships formulate complex structures in order to form context. In the natural language processing these king of structures and ways to write text work as a identifying elements that help processing to find out what kind of information text is likely to having inside. "For example, information extraction from newspaper articles is usually concerned with identifying mentions of people, organisations, locations, and extracting

useful relations between them." These information pieces can be great use when results are analysed later [33].

2.4 Event Driven Programming

When examining the characteristics of Event Driven Programming (E-DP) strongest attribute is being triggered by users input in arbitrary order. Comparing to procedural programming where event happen in the order that has been determined beforehand. This makes E-DP better in situations where flexibility is needed. E-DP also differs in the object structures from procedural programming using pre-defined visual object rather than non-visual user-defined object that are used greatly in Object Oriented Programming (OOP).

Event driven programming (E-DP) is characterised by programs that can be triggered by the user in an arbitrary order, rather than in a pre-determined order as in procedural programs. All event-driven (E-D) tools use pre-defined visual objects, compared to the non-visual, user-defined objects that are typically used in object-oriented programming [35].

From the experience implication can be formed that E-DP cannot apply OPP design structures and guidelines as they are due to difference in the approaches. Like previously presented the E-DP uses different procedures than procedural programming. Procedures are invoked by signals or messages from other procedures. On the other hand E-DP procedures are triggered or called by events. Of course this doesn't mean that procedures in procedural programming cannot be called by events but the important difference is that parameters cannot be passed to event based procedure, except rare fixed parameters. Other differences are forms and visual objects. Forms in the E-DP work as a containers for procedures and declarations. This means that forms will affect to the user interface and how to maintain it. In the case of las point visual objects are available for the tools of E-DP. On the other hand OOP tools use user defined non-visual object. The difference with these approaches is that like stated parameters cannot be passed for event procedures [35].

Event in the E-DP is usually something similar to these. Every event in the system has dedicated handler. An Event handler is function that will be operated when its related event has happened. After the dedicated function has ended its execution system returns to event dispatch loop and waits next event to happen. This will continue as long as there is no exit signal for the program. Example for exit signal could be user closing the program [36].

List of possible events [36]:

- A signal indication that disc operation has completed.
- A package has not arrived in the required time window.
- Connection to the network has beed disconnected.
- An action at the GUI element.
- Server has received and incoming message.

The great advantage of the E-DP is that it can process events in the parallel. This way resources are freed for other operations and are not bound to wait until the function execution has ended. In real world this means that system can clear list of waiting operations queue faster. Gainings from this is better performance and capability for higher load [37].

3. DESIGN AND IMPLEMENTATION OF THE CREDIBILITY ASSESSING SYSTEM

The implementation started with studying earlier prototypes and solutions. The study about sensor discrimination proposed method where sensors are in relationship with credibility assessing system. With this method only relevant sensors are used to calculate credibility and rest of the sensors are left alone. However this method does not say what is the best infrastructure for operating this way [2]. In order to solve this problem this paper propose method that will handle infrastructure of the data and information flow in credibility system. Other objective of this method is preprocess as much data beforehand in order to matrix node graph function could be faster.

The proposed method follow service oriented architectures principles. In order to achieve its design targets method integrates multiple datasources together under one service that serves matrix node graph service. Method also uses 3rd party services to get additional information to help credibility assessing process to be more accurate and unifying used data structures and models.

3.1 Component overview

Because this method is supposed to work with previously proposed solutions there where couple of constrains for tool requirements. Firstly used tools and techniques should be able to provide needed functionalities in order to achieve wanted end result. Secondly they should be as compatible as first constrain allows with the system that it will be integrated in.

3.1.1 Platform

Docker was chosen to be a platform for this prototype. Current days continuity is important and necessary for platforms growth. With docker platform compatibility issues are easy to avoid and it will allow easy integration of other functionalities to be integrated in the proposed method. The list at below shows that docker has good qualities for being

platform for this kind of system. Docker also supports all the necessary techniques used in this method.

List of benefits using docker [38]:

- Enables more efficient use of the system resources
- Enables faster software delivery
- Enables application portability
- Shines for micro services architecture

All the important functionalities from this methods point of view in previous systems where done with python programming language. Additionally to basic functionalities of python some of the additional libraries for calculation where used that where necessary to include this method in order to have data preprocessing done. This meant that python was chosen to be basic programming language for this prototype.

Because this method requires two services to be created docker compose was chosen to be used to handle service relations and configurations. This simplified system configuration and ensured that all needed local services where working together as planned. From the appendix 1 we can see the relation of the data integration service that is called “webcredibilityAPI” and database service which is called “dbmysql”. This separation follows the principles of service oriented architecture and ensures that the resource management is going to be easier. Appendix 2 shows configuration of the data integration service. The service is based on python image for docker that is in public circulation and used as a base for python applications on docker. Other lines are internal configuration parameters for docker container excluding three lines where polyglot is mentioned. Polyglot is text analysing part of this method and it needs outside files for operating. This is why they have to be manually added in docker container configuration step.

3.1.2 Framework

Choosing techniques necessary for the functionalities at code level where easy. Python was already locked as main programming language and time restrictions of the other participants of building the system meant that some simple solution for the framework was needed. After short research Flask framework was chosen to be main communication framework for this project. It offers easy REST API building tools and has relatively gentle learning curve.

Flask framework is mainly used for REST API functionalities needed for communicating with sensors. Flask also offers more complicated functionalities but in scope of this project those are unnecessary. However it is good to have some flexibility for the future. Other choice for this task was Django framework. It is more complete backend framework that offers more features and flexibility for managing data but at the same time is more complex and takes more time to learn. One good example of better coverage of features is build in database management in Django that has query builder functionality. This in many cases this can result better security for database because raw sql is rarely needed. With Flask framework database management is handled by with some other way and in this project default python-mysql-connector is used.

3.1.3 Essential tools

From the start some of the tools used where necessary in order to integrate preprocessing into this service. Because same coding language was chosen for this service, functions from previous works could be added without any modifications. In order for these functions to work, numpy and pandas function libraries had to be included in the service. These two libraries are widely used for the complicated calculations with python and are needed for the preprocessing calculations.

Polyglot and nltk libraries are also necessary for the calculation functions. One of the great feature of the matrix node graph calculations was classification of the intension from the news article. In order for the application to understand written language, natural language processing is needed. With these function libraries it is possible to analyse written language and process wanted information from it.

Apscheduler is python function that allows other functions to be scheduled as a background tasks from the main program. This is needed in order to influence asynchronous functions based on events that happen in main function. In this case main function is REST API and events are receiving message from sensor.

In order for this method to provide not only sensor data but also news information, RSS feed is used as a datasource for news. Because RSS is not readable format for humans it has to be parsed before it can be processed. For this job feedparser function was chosen. With it RSS feed can be parsed and important news data from the source can be extracted and analysed.

3.1.2 External services

In this method location information from the sensor is important part of finding relation between sensor data and news information. In order to get all the necessary data and have general data structure for the location outside service is used to generate geo data based on the coordinates from the sensors. For avoiding licensing problems open source service is used and best option with these parameters was Nominatim function of open street map. The location service uses similar REST API and gives response as a JSON package. As for a REST call, python request library is used to communicate with the service.

3.1.4 Database

Database service is second local service for this method. Because of the way docker can organise networking between containers main service and database service look to each others like they would be in same system. However in reality they are separate docker containers and only link between is docker network. This means that database locates in its own dedicated container and proposed method in other dedicated container. In order to achieve same functionality it is not necessary to have database on its own container but it is more future proof design for the system. This is done because database is the most crucial single component in the whole system. For the credibility assessing and information service to work asynchronously it is necessary to have them in their own processes. However some communication between processes is needed and this is done via database. This means that database has dual purpose in this method. First is to store all necessary information produced by the information infrastructure. Second is to serve that information to credibility assessing process.

Database structure for this method is designed in a way that the information loss in the data processing would be as minimal as possible. Also the design tries to achieve necessary functionality with simple structure so that it is easy to understand and allows good groundwork for future developments on top of the method. For all the simplicity only tow tables are needed for storing necessary information. At this point there is no need for relations. Figure 6 shows the complete table structure of the database.

Figure 6. ULM diagram of information service database

Sensor data table is responsible of storing the data received from the sensors. Primary key of this table is id. Id is only there to identify the row and is self generated, individual and not null. In order for sensor data being structured and easily analysed there has to be at least some identification that differentiates data rows from each others. For the received data itself Id is not important part and does not add more information additional to other columns. Datatype for ID field is int.

Second column in the sensor_data table is sensor type. This column contains classified information about datasource. For credibility assessment system to find out what kind of sensors should be used in the process there has to be some way to know what sensors do and if their data is relevant. Same thing is done with the next column that stores sensor purpose. With these two columns, necessary information about the sensor can be stored with the produced data. Data format for these fields is varchar.

Timestamp and from sensor and system are quite self explanatory. Sensor timestamp is time for measurement and system timestamp is for receiving time. For the credibility system these are important pieces of information because with time information algorithm can decide if the data is relevant for the assessment. Both of these are voluntary information but at least system timestamp is added to every entry by the system. Timestamp was made voluntary because it is not possible to get that information with every kind of resource that will send information to this system. However with the receiving time should give at least some rough idea where in the timeframe send information is related. Both columns in mysql database are date time types and use combined isoformat presentation where date and time are together and timezone information is included. Used iso standard is ISO 8601 and example of this can be seen in figure 7 Also all the time information in received packages should follow same representation.

YYYY-mm-ddThh:mm:ssZ

Figure 7. ISO 8601 standard time and date representation with timezone information [39]

Sensor data and sensor id columns store individual sensor id and boolean data from the sensor. Most of the time data from the sensors is boolean type data. For example earthquake sensor does not send data if there is nor earthquake but sends true message when there is something happening. Every other event from the sensors can be transformed in to similar datatype. If sensors nature is more streaming data there should be separate system which is responsible for analysing that information and sending true or false message to this solution. In the literature review part of this paper multiple good application choices for the streaming management are listed. However they are not implemented in this solution because of the operating nature and time restrictions. Datatype of the data column is boolean. Sensor id is individual id number that is given to each sensor or system that is sending data to this solution. It can be received from the root url address of the REST API and has to be implemented to the send package. With this id

number different datasources can be differentiated and used to help analysing in real time and in the future. Data type for id is varchar.

In the news_data table ID and time are similar that in the sensor_data table. Id is unique and for each entry and is auto incrementing value that is primary key in the table. Likewise time is date time object that tells when news article has been published but in this case value has to be parsed from the RSS feed and then transformed to datetime datatype. Otherwise time information is similar but timezone information is missing because source data does not have it.

Title and summary fields contain information from the news itself. Like naming suggests title field contains title of the news article and summary contains small summary of the articles text itself. Both of these fields are varchar data types. Especially title information is important because it is main source of data used to analyse credibility. Summary of the article is not as important but is included to the model in case that in the future there is some use for it.

Matrix field contains preprocessed information from credibility assessment function. This is one of the major features of this system. With preprocessing credibility assessment function can operate faster because one major analysing step is already done. Like name indicates matrix field contains result matrix from the matrix node graph function that is used to assess credibility. This matrix is numpy class object which means that it is not compatible with any field types in mysql. In order to change numpy class object to compatible format it has to be transformed to byte string. This operation can be done with build in functions found from numpy function library. For byte string to be stored safely in the database field type was chosen to be blob. MySQL documentation says that blob datatype is a binary large object that can hold a variable amount of data. Only difference is the length of the object [40]. This means that the chosen datatype for field is good fit for the purpose. For matrix data to be usable again after reading it from the database byte string has to be converted back to numpy class object. Fortunately like conversion to byte string there is build in function in numpy function library for this purpose.

Category and intension field store other results from the preprocessing function. Both of these results are dictionaries that have certain structure for calculated information. Like the matrix category and intension results are used to further calculate assessment for information. Also for both of the dictionaries have to be converted to other format in order to be compatible with Mysql datatype. In this case both dictionaries are converted to JSON format and stored to archer field. For these strings to be useful again after reading them from the database, like matrix data, they have to be converted back to dict datatype. For this operation pythons abstract syntax tree library was added to solution and used for conversion.

Link is quite self explanatory name for the field and its information. It contains url address of the article. Currently this information is not used by the credibility assessment function but it is important to have location of the original source of information for the case in the future need for it appears. Not only to add more transparency to the process but also information extracted from the RSS feed is only small part of the complete text. With the url address complete article can be accessed and possibly used to get more accurate credibility assessment.

3.2 Operating logic

Like mentioned earlier in this paper, designed system follows event driven programming. This means that system react to events in certain way based on its configuration. When system is started it enter to the state where it waits something to happen and starts series of reactions when configured trigger event happens. Because this system is build around REST API framework the trigger is package receiving event. Figure 8 shows component structure and how the operating logic works. Implementation of the proposed method is located inside docker container.

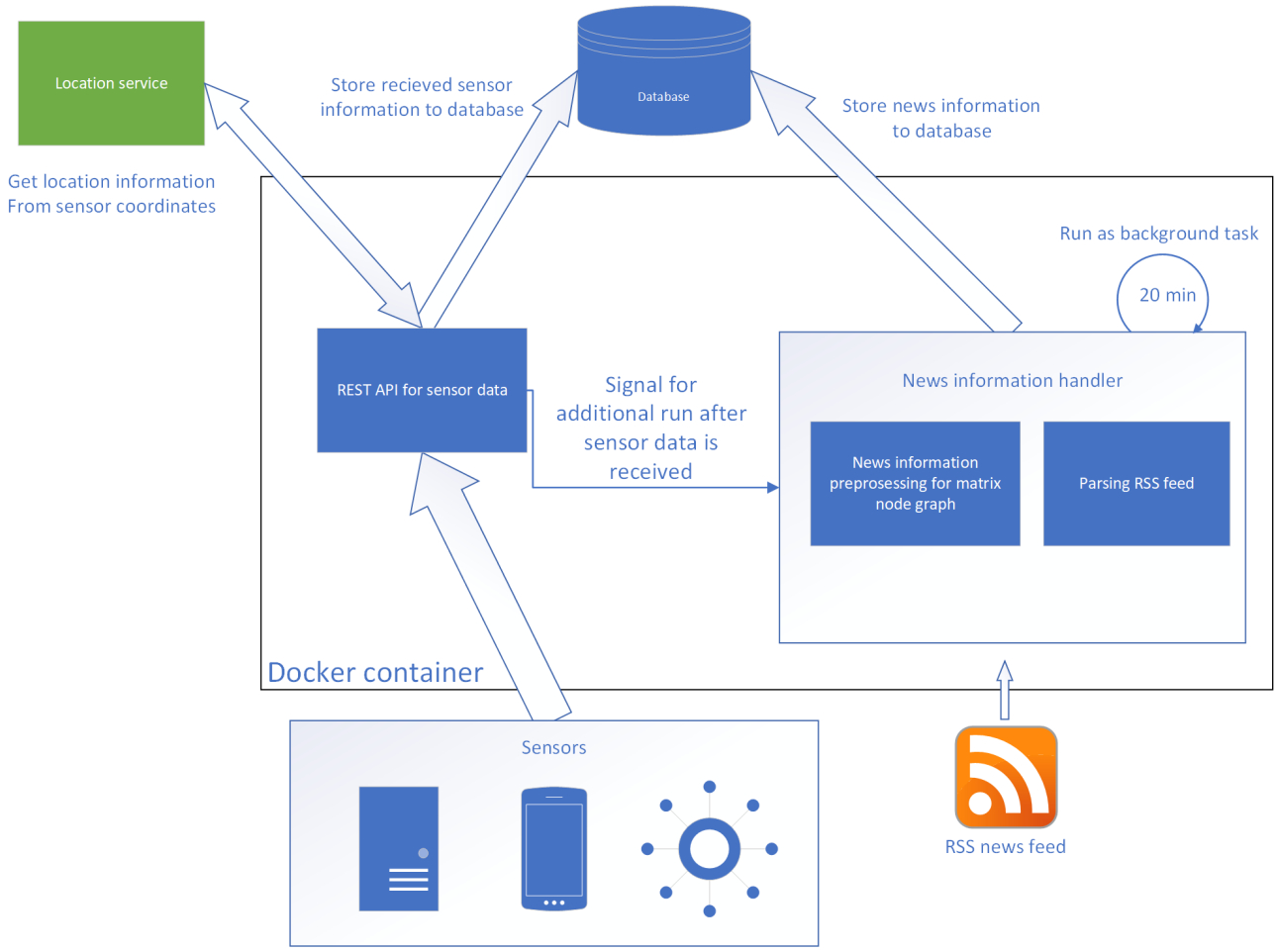

Figure 8. Visual representation of the solution

3.2.1 Component relations

Main component in the solution is the REST API part of the program. All the other parts start to move after package is received by API. Because main purpose for the system is to integrate datasources under one system for the credibility assessing function, it acts as a hub that receives information from sources. Sending packages is not as important and current version only gives http code 200 as response for the sensor source to confirm successful receiving. First thing when tis system is started is to run news information handler. This part of system receives RSS feed and parses it to correct format. After information is in the format that python can use it starts the preprocessing part of the component. First the function checks if article is already in the database. It if is function

moves to next article in parsed list. If article is not in the database will the function run in trough preprocessing. In this process matrix, intension and category are calculated based on news articles title. After this process articles information is stored in the database. After this function has run will the main process of the system set information handler function to background with the interval of 20 minutes. Like the figure 9 shows longest that this function takes is 0,024 second and shortest is 0,009 seconds. Based on this updating news information every second is not impossible. Interval is chosen to be 20 minutes because requesting RSS response every second from the service provider would not serve any purpose. With 20 minute interval system is still close enough to real time for this use and doesn't add burden to chosen news service.

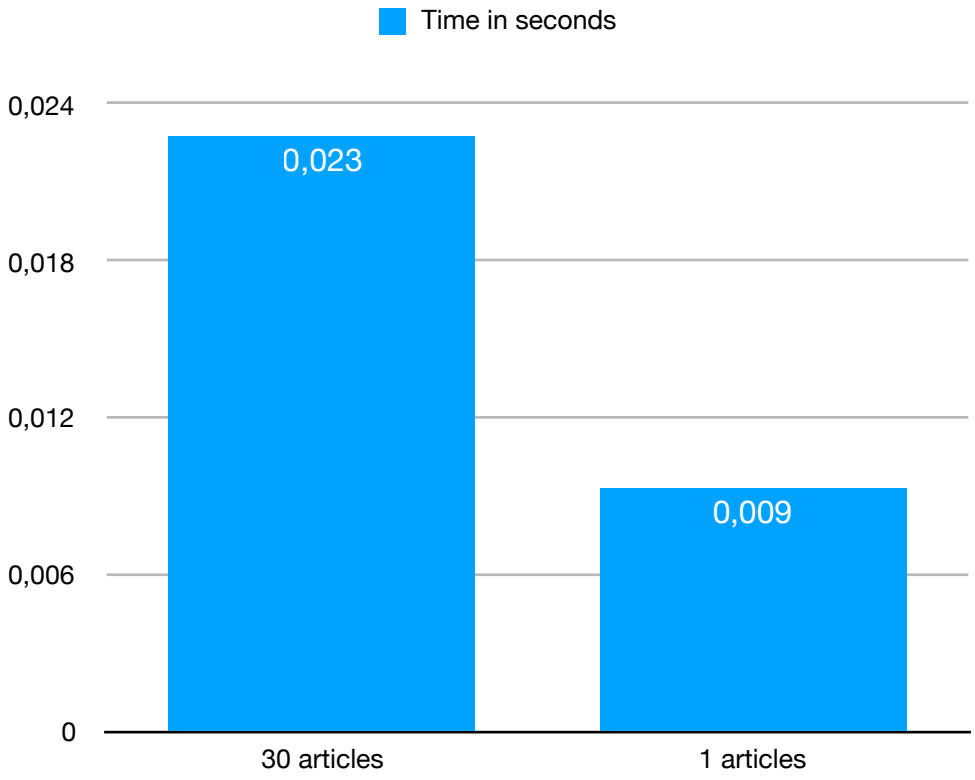

Figure 9. News processing and storing time in seconds with 30 new article and 0 old articles versus only one new article with 29 existing articles.

After the news information handler function has run the Flask process in started and left to wait messages from the sensors. If there is now messages from the sensors Flask process just continues staying alert. However when some sensor sends message chain reaction starts. First sensor data has to be saved in the database. However this cannot be done straight away because information from the given data is not complete. First thing is to get location information from the coordinated that came with the sensor data package. Like mentioned earlier in this paper location data is processed in third party service and result is saved. Also time from the sensor package has to be converted in suitable form. After data processing is done it is added to the sensor_data table. Once sensor data is stored safely will the logic remove news information handler task from the background tasks in order to avoid conflicts. News function is immediately run in order to confirm that all the recent news articles are preprocessed in the database. Right after forced run news function is added back to background tasks.

3.2.2 Sensor data classification

Storing data sent by the sensor is helpful only to the point. The more information can be extracted from the data, it is possible to have more accurate results with it. Previous work about credibility assessment system proposed method which would discriminate sensor sources depending on the article topic and choose only suitable ones for the assessing process. However in this solution there is no active discrimination or choosing of sensors. Choosing right data sources for the purpose is important but for this solution to work getting data stored is more important for this process. Reason behind this is that even tough received data is irrelevant currently there is no way to know if it will be useful in the future.

In order for this solution fulfil requirements of the discrimination we designed system to classify integrated sensors. For assessment system to be able to know what kind of data is useful for each of the news article assessing process it needs to know what kind of sensors where giving data in the time window of the assessing process. When we looked at the

attributes of the sensors we found out that two of those were important for finding out what kind sensor is in question.

First of the two attributes is purpose. This will tell about what kind of events is sensor measuring. It is important to know what is the method how the result was gotten from the source. Many times in the science world straight measurement is not possible but following events that are related to the one which is under research it is possible to prove its existence. Also having multiple different related events telling that hypothesis for that kind of event is happening helps generate correct conclusion. For example the case where tsunami is hitting the cost of the land it is really hard to create dedicated sensor for only that purpose. However if we take measurements from motion, water level, air pressure or from other related sensors it is possible to combine information and create accurate connection from the reason to the behaviour of the measurements. Next list shows up created choices for sensor purpose in this solution.

List of classifications for sensor purpose:

- motion
- air_pressure
- water_pressure
- water_level
- water_temperature
- air_temperature
- linear_acceleration
- angular_acceleration
- gyroscope

- magnetometer

Second of the attributes is functionality of the sensors. First attribute tells what kind of method sensor is using to sense but functionality will tell about what kind of event should that sensor measure. For example air temperature sensor measures temperature but that could also be used to find out information from other events. These events could be like thunderstorm where huge amount of moisture can change the air temperature quickly. If this information is combined with other kind of sensor information it is possible to rise accuracy of the event measurement and predictions. Next list contains classifications of this solution for sensor functionality.

List of classifications for sensor functionality:

- earthquake
- flood
- tsunami
- cyclone
- tornado
- heavy_rain
- heavy_wind

Like mentioned previously in this paper different events can be observed straight with correct method or indirectly by observing events related to the target event. Solution proposed is designed only to work with events related to nature like for example earth quake or tornado. This is done because some of these indirect relations are hard to find out and take time to implement in the assessment process. Also simulating more complex

events that are as easy to understand as natures are also quite hard to prove right. System could give answer and it can be really close to right one but knowing for sure that for example finding out if robbery happened in the store needs greater research between what kind of sensors should be used and how measuring algorithm should behave.

4. RESULTS

One of the target for this research is to find out if it is possible to make matrix node graph algorithm faster than previous implementations. The prototype system was created to prove that solution designed would work in the real word but also because it had to be measure in order to have comparison between old and new solution. Also it is important to provide basic metrics of the system in order to making future improvements easier. It is important to know what is the starting level of the system before improvements are implemented in order to know after measuring if the improvements where actually doing what their initial requirements where saying they should do.

4.1 Preprocessing

Preprocessing is one of the main components of the matrix node graph that works as a assessing algorithm in this solution. Without preprocessing it is impossible to have information in correct form for the analysing part of the algorithm. This means that the impact of the preprocessing is critical for the matrix node graph function event tough it only does couple of initial analyses from the news article.

The main functionality of the preprocessing is to analyse the news article and output data that is used for comparison in later stages of the process. Within this analyse process news articles title is run trough natural language tool that categories each word of the title. This is done in order to be able to allow classifying function to find out the attributes of the title with the logic made for matrix node graph method. With this preprocessing algorithm outputs matrix, intension and category datatypes that are also mentioned in the database section of this paper. These outputs are later used as a reference in the matrix node graph analyse.

When we started to design new solution for data infrastructure for the matrix node graphs system first step was to analyse old solution and understand how it operated and what parts could be improved without touching to the main logic of the system. Right away we found

out that preprocessing function could be separated from the old solution as long as its results where available when the main analyse function was running. We also found out that the nature of the sensors used was different from the analysing logic of matrix node graph. While matrix node graph relies on repeatedly running in interval the sensors operate more on standby and take action only when something is happening. With this I refer to the nature of the complete system with the sensor, not sensor itself. There is always little bit of logic with the sensor system for it to be able to communicate with outside systems. For example water level sensor measures the level of the water all the time. However the logic of sensor package should send message to the system proposed in this paper only when water level goes over the set level of the sensors logic.

Based on the qualitative part of this paper it was clear that this kind of integration solution should be REST based system and it should have its own database logic in order to be compatible to the matrix node graph system. REST API follows the design of event based programming and is main component used for communication. However not all parts used in this solution follow same design. Because news articles are made continuously without relation to this solution they have to be continuously parsed and stored to this solution. Fortunately it was possible to run parsing function as a background function and create logic to the system that allows parsing function to react to the REST API.

With these designs it was possible to move preprocessing function to this solution and remove it from the matrix node graph application. Target for this change was to allow preprocessing to be working with news information function and with that we have almost real time results with the news information. Also this change meant that if the database access would be faster than preprocessing function, the matrix node graph method would be faster that previous version. From the figure 10 we can see that the database access is clearly faster than old method where preprocessing was done as a part of matrix node graph method. With only few articles there is no real difference but from the 5 or more articles there starts to be growing gap between speed of the processes.

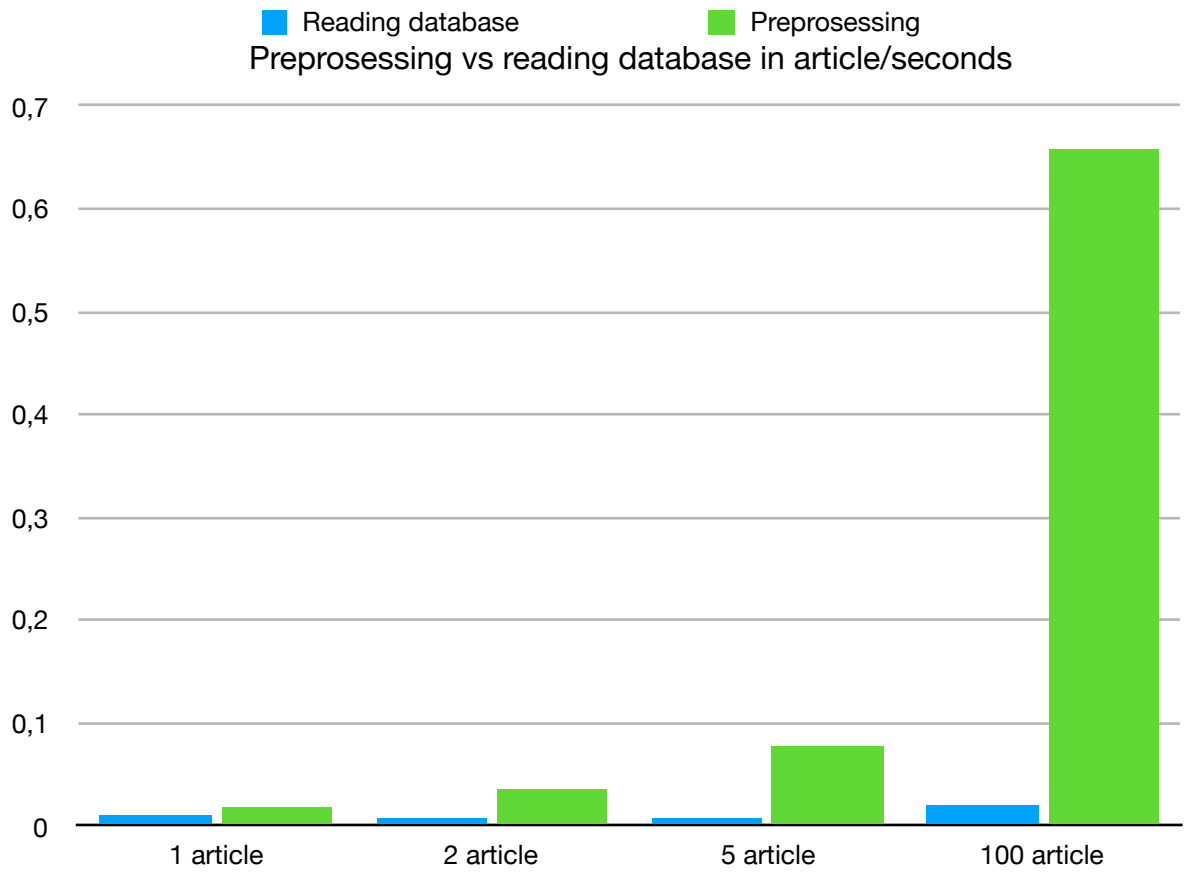

Figure 10. Result comparing speed difference between preprocessing and database access

The measuring process where preformed by taking timestamps between start and end of both methods and finding out the time difference. With the new solution measuring starts before database connection is made and ends when data is in same format that it would be after preprocessing function. This means that for example matrix fields data from database has to be converted back to numpy class object. With old solution measuring starts right before preprocessing function starts and ends at the same state that with new solution where data is preprocessed and ready for use. With current version RSS feed only contains 30 articles already stored articles where used in order to get higher count of data being processed. This was done by accessing database and taking correct amount of articles before closing connection in order to get accurate measurements.

4.2 General functionality

Like with preprocessing function analysing previous solution played great part when designing new version for the data integration. In old version there was no connectivity to outside world and all used source data was stored in JSON file in the system. Also when looking at the way processing works it was clear that algorithm was linear and there was not many possibilities to divide processes in order to run them as parallel to each other. This was understandable when looking at the presumptions of the old system which main purpose was to prove that proposed method is working as intended and gives correct result.

Design principles in this design where not only to create solution for the one use case but to ensure that there is good groundwork for future improvements. Decision to use docker system as platform was done not only because it is good tool and authors are familiar with it but also because it has great functionalities that cannot be found from normal web server or are hard to implement. First and obvious reason for this choice was to have great compatibility with different system without much difference on functionality. Sharing platform that could be build up with single command made integrating this service to original matrix node graph easier than with original web server platform.

Of course when creating solution to data integration one of the design principles is to create way to add new sources of data easily to the system. In this case problem is the amount of news data generated all the time. In this solution worst case scenario is that all 30 of the article in the only integrated RSS feed are not in the system and they have to be processed. Like previously presented in this paper this process takes about 0.023 seconds to complete. However like said this is only whit the one news source. If more sources are added in this solution as it is process is going to be linear and new news source is going to be processed after the one before it has completed processing. This means that the more news sources are added the more time preprocessing is going to take. Even tough with current solution scope is less than a second, little by little adding more sources the processing time could grow to be significant. From the figure 7 additionally to the comparison between preprocessing and database access we can also get rough idea that

with this solution database connection could handle huge amount of data in the same timeframe than the preprocessing of 30 news articles. With docker it is possible to initialise multiple instances of this solution to same web server. From the figure 11 we can see that there is plenty of resources left for one or two more instances of webcredibilityapi container. In this case memory is limiting resource and needs to be optimal in order to get good performance. With multiple instances of the solution it is possible to dedicate only few news sources for one instance. This means that processing per instance doesn't take too much time and tasks between instances are ran in parallel. When processes are run in parallel they don't have to wait each other and more data can be processed in smaller timeframe. Based on the assumption that database can handle more data in small amount of time there is still long way before database will become limiting factor in terms of speed. With this single database can be used and functionality will stay same from the matrix node graphs point of view.

NAME	CPU %	MEM USAGE / LIMIT	MEM %
di_project_webcedibilityapi_1	3.13%	251MiB / 992.8MiB	25.29%
di_project_dbmysql_1	0.22%	186.4MiB / 992.8MiB	18.77%

Figure 11. Docker container system resource usage

5. DISCUSSION

This section discusses about the whole research process from the qualitative review to the implementation and analyse of the results. In the subsection 5.1 design choices and results from qualitative research are analysed in the big picture and put in the context of whole system. Subsection 5.2 discusses about challenges and compromises made during the project. Last subsection 5.3 looks about what are the possible directions of future researches.

5.1 Analysing the Results

When the designing of the system started quite soon it was clear that the solution would be based on the event driven programming design and it would act as a service for the matrix node graph method. Important requirements for the system at the start where to have common data structure for the data send by sensors and try to help matrix node graph method to be faster in order to be more real time service. The problem that this solution tries to solve as a part of bigger solution is to reduce misinformation and that way prevent human action based on misinformation. With this problem time is important because action made by humans are placed in certain time and place. Like mentioned earlier in this paper it is critical to have to have correct information as soon as possible.

If we only look at the overall process faster than before then the answer would be yes. Measuring the time between old and new solutions the gap is bigger the more material it has to analyse. However like in the case of 100 articles to be analysed time used for processing is still under a second. Comparing this to the 3 hours interval between analyses with the matrix node graph these results seem to be minor and the impact probably won't show for enduser. Of course at this point the argument could be made that there is much more news data out there and with bigger amounts of data all the reduced operating time is better. Again this might be true but still the impact of this time reduction would be marginal compared to what gains could be possible when optimising the matrix node graph function itself.

Also the type of data used in this system has to be taken account as well as how fast data is processed. In this case integrated data sources are as simple as they can be and require nearly no additional processing to get desired information out from it. Comparing this to retrieving information from the social media or tracking more complex events with multiple different sensor which require more complex algorithm for information to be extracted. This means more time used for each news article and sensor event. However these problems are similar to all the other similar system but again when compared to the 3 hour interval of matrix node graph it is possible that this solution would still be in that timeframe and there would be no impact to the enduser.

Related to faster processing is chosen platform which allows multiple instances of the solution be initialised. With this it is possible to have more parallel calculations and this way reduce overall time usage. However in the current form of the solution this is not possible without changing the logic a little bit. In the current solution REST API and news information parsing function are located at the same container. This means that every new instance of this solutions container has same REST API with can cause problems when communicating with the sensors. By dividing these two components would resolve this problem.

5.2 Encountered challenges and Compromises

First big compromise made in this solution was to not include support for integrating streaming type sensors. In original plan it was decided that streaming support would be added alongside to the REST API. However time contains did not allow this to be done in the prototype. Fortunately architecture used for the solution still allows streaming management system to be added afterwards but it has to be added as its own service which communicates with matrix node graph trough the REST API like other sensors. Best way to implement this would be create its own infrastructure with the containers and if possible use shared local network build in to docker system to make it more robust solution.

During the creation of the prototype system one of the biggest problems was to get signal from on function to the other when it was necessary. The problem here was to run function independent from each other most of the time but on the certain moment have other function react to what is happening with the other. This was challenge because the nature of the REST API is to be active on the downtime and be ready to receive. This means that the process is running all the time and there is not much room for other process to be run automatically at the same time without disconnecting from the first process. Fortunately there is background task support in python and this problem could be solved but it is not the most elegant way to solve this problem. Other way to solve this would have been to use same solution that would have solved the problem with the two services being within same container. By dividing current background task and REST API to their own services would have made this solution easier to develop further in the future. However I was reluctant to do this in order to avoid networking overhead which would be greater because only way to service to communicate is to send packages via network to each other. In the current solution they are in the same container and no networking is needed.

5.3 Future Research

For the future research there are following topics. The first is to find out how to add nontraditional news sources to the system. Social media is gaining footprint in the news industry and is widely used for distributing information. These platforms provide information that is not accessible in the same way than many dedicated media corporations are providing their content.

Second area of research should be what kind of output and how this kind of assessing system should handle it. The solution proposed in this paper doesn't specify what the output should do. The matrix node graph with this service attached will give answer as a probability percentage but what king of interface it should have.

Third issue is to try make matrix node graph process more efficient from the used time viewpoint. Interval for the process is set on 3 hours in order to be sure that processing is

completed before the next round is started. For example with the amount of data this solution has gathered it is possible to teach machine learning algorithm to handle some parts or whole analysing process. Question at this point is to where to apply the machine learning and how much it will benefit the process.

6. CONCLUSION

This master's thesis project was based on expanding the research around information assessment with sensors. The first steps where to define the problem and define the scope of the research that would fit to the requirements of the master's thesis. This is explained in the first chapter of this paper in more detail. After the research and tasks were planned, knowledge about the problem and insight about related techniques where gathered by conducting literature research. This empirical part is documented in the second chapter of this paper. Based on the information gathered on during the literature research, architecture was designed for the datasource integration to the information assessing system. Architecture was implement in the form of prototype system in collaboration of authors from previous research of the subject. This process is documented in the third chapter of this research. Because this was complete new system that will be added to the original solution it was necessary to find out if there is improvements with the new solution. In order to explain findings it was necessary to provide context for the measurements. Test where made in the program and results recorder and visualised. This is done in the fourth chapter of this paper. Even tough results where presented in the fourth chapter it is necessary to provide more context to them and show their relation to the other parts of the system in order to understand the impact of the changes. This was discussed in the fifth chapter of this paper.

At the end it seems that overall the project was a success. Created prototype fulfils the requirements that where placed for it at the start. During the development there were some problems that took some time from the development but with the help of research team problems where solved. With the results of this research, Ken Honda's and Naofumi Yoshida's matrix node graph data structures application for credibility assessment could be tied to real world events with relatively real time responses. Also this research shows that with the data integration it is possible to improve performance of the system by dividing processing load if logic allows it.

REFERENCS

1. Honda, K. & Yoshida, N. 2019. A Matrix Node Graph Data Structure and Its Application for Credibility Assessment with Temporal Transition of Intension. The Proceedings of the 28th International Conference on Information Modelling and Knowledge Bases, EJC2019, June 3-7. Lappeenranta, Finland.
2. Honda, K. & Yoshida, N. 2019. A sensor selection and learning method for credibility assessment using sensor data. *Frontiers in Artificial Intelligence and Applications, 312*, pp. 400-414. doi:10.3233/978-1-61499-933-1-400
3. Zubiaga, A. 2016. Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. *PLoS ONE, 11*(3). doi:10.1371/journal.pone.0150989
4. BBC News, 2018, Burned to death because of a rumour on WhatsApp [online]. [Referenced 2.7.2019]. Available at https://www.bbc.com/news/world-latin-america-46145986
5. Zubiaga, A. 2015. Towards Detecting Rumours in Social Media [online], [Referenced 2.7.2019] Available at https://arxiv.org/abs/1504.04712
6. Wierzbicki, A. k. 2018. *Web Content Credibility*. Cham: Springer International Publishing [online]. [Referenced 2.7.2019]. Available from https://link-springer-com.ezproxy.cc.lut.fi/book/10.1007%2F978-3-319-77794-8
7. SHIBUTANI, TOMOTSU. (1966). Improvised News: A Sociological Study of Rumor [online]. [Referenced 2.7.2019]. Available at https://books.google.co.jp/books?id=zJypXrE2xqAC&pg=PA227&hl=ja&source=gbs_selected_pages&cad=2#v=onepage&q&f=false
8. Honda, K. 2016. An implementation method of an information credibility calculation system for emergency such as natural disasters. *Communications in Computer and Information Science, 637*, pp. 193-201. doi:10.1007/978-3-319-44066-8_20
9. Sagi, T. 2018. Non-binary evaluation measures for big data integration. *The VLDB Journal, 27*(1), pp. 105-126. doi:10.1007/s00778-017-0489-y
10. Lenzerini, M. 2002. Data integration: A theoretical perspective [online], [Referenced 2.7.2019] Available from https://dl.acm.org/citation.cfm?id=543644

11. Brown, K. S. 2019. Categorical data integration for computational science. *Computational Materials Science, 164*, pp. 127-132. doi:10.1016/j.commatsci.2019.04.002
12. Alrehamy, H. 2018. SemLinker: Automating big data integration for casual users. *Journal of Big Data, 5*(1), pp. 1-26. doi:10.1186/s40537-018-0123-x
13. Ma, B. 2017. A Novel Data Integration Framework Based on Unified Concept Model. IEEE Access, 5, pp. 5713-5722. doi:10.1109/ACCESS.2017.2672822
14. Georgakopoulos, D. & Papazoglou, M. 2009. *Service-oriented computing*. Cambridge, Mass.: MIT Press [online], [Referenced 5.7.2019] Available from https://app.knovel.com/web/toc.v/cid:kpSOC00001/viewerType:toc//root_slug:service-oriented-computing?kpromoter=marc
15. Shi, H. 2015. A service-oriented architecture for ensemble flood forecast from numerical weather prediction. *Journal of Hydrology, 527*(C), pp. 933-942. doi:10.1016/j.jhydrol.2015.05.056
16. Messenheimer, S. 2005. The Impact Of Service-Oriented Architectures On Data Access and Integration. *Software Development Times, 125*, pp. 31-32.
17. Hafner, M. & Breu, R. 2009. Security Engineering for Service-Oriented Architectures. Berlin, Heidelberg: Springer Berlin Heidelberg [online], [referenced 18.9.2019] Available at https://link-springer-com.ezproxy.cc.lut.fi/chapter/10.1007/978-3-540-79539-1_2
18. Usländer, T. (Ed.), 2009. Specification of the Sensor Service Architecture Version 3.0 (Rev. 3.1). OGC Discussion Paper 09-132r1. Deliverable D2.3.4 of the European Integrated Project SANY [online], [Referenced 2.7.2019] Available from https://portal.opengeospatial.org/files/?artifact_id=35888
19. Usländer, T. 2010. Designing environmental software applications based upon an open sensor service architecture. *Environmental Modelling and Software, 25*(9), pp. 977-987. doi:10.1016/j.envsoft.2010.03.013
20. Khan, M. S. 2014. Enhanced Service-Oriented Open Sensor Web Architecture with Application Server Based Mashup. *International Journal of Distributed Sensor Networks, 10*(6), p. 313981. doi:10.1155/2014/313981

21. K M, J. 2017. Wireless Sensor Network. International Journal of Advanced Research in Computer Science and Software Engineering, 7(6), pp. 588-596. doi:10.23956/ijarcsse/V7I6/0277
22. Happ, D. 2017. Meeting IoT platform requirements with open pub/sub solutions. *Annals of Telecommunications, 72*(1), pp. 41-52. doi:10.1007/s12243-016-0537-4
23. Fang, W. 2011. Design and evaluation of a Pub/Sub service in the cloud [online], [Referenced 3.7.2019] Available from https://ieeexplore-ieee-org.ezproxy.cc.lut.fi/document/6138542
24. Myers, B. 2016. Improving API usability. *Communications of the ACM, 59*(6), pp. 62-69. doi:10.1145/2896587
25. Clarke, S. 2004. Measuring API usability. *Dr. Dobb's Journal, 29*(5), pp. S6-S9.
26. Henning, M. 2009. API Design Matters. *Communications Of The Acm, 52*(5), pp. 46-56. doi:10.1145/1506409.1506424
27. Battle, R. 2008. Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST). *Web Semantics: Science, Services and Agents on the World Wide Web, 6*(1), pp. 61-69. doi:10.1016/j.websem.2007.11.002
28. Varanasi, B. 2015. *Spring REST*. Berkeley, CA: Apress, pp. 1-2. doi: 10.1007/978-1-4842-0823-6_1
29. Gama, J. & Gaber, M. M. 2007. Learning from Data Streams: Processing Techniques in Sensor Networks. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg [online], [Referenced 3.7.2019]. Available from http://ezproxy.cc.lut.fi:80/login?url=http://dx.doi.org/10.1007/3-540-73679-4
30. Jovanovic, Z. 2015. Data stream management system for moving sensor object data. *Serbian Journal of Electrical Engineering, 12*(1), pp. 117-127. doi:10.2298/SJEE1501117J
31. Cold, S. 2006. Using Really Simple Syndication (RSS) to enhance student research. *ACM SIGITE Newsletter, 3*(1), pp. 6-9. doi:10.1145/1113378.1113379
32. Ma, D. 2012. Use of RSS feeds to push online content to users. *Decision Support Systems, 54*(1), pp. 740-749. doi:10.1016/j.dss.2012.09.002

33. Kao, A. & Poteet, S. R. 2007. Natural Language Processing and Text Mining. London: Springer-Verlag London Limited [online 25.7.2019], [Referenced]. Available from http://ezproxy.cc.lut.fi:80/login?url=http://dx.doi.org/10.1007/978-1-84628-754-1
34. Beysolow II, T. k. 2018. Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing. Berkeley, CA: Apress [online], [Referenced 25.7.2019]. Available from https://link-springer-com.ezproxy.cc.lut.fi/book/10.1007%2F978-1-4842-3733-5
35. Philip, G. C. 1998. Software design guidelines for event-driven programming. *The Journal of Systems & Software, 41*(2), pp. 79-91. doi:10.1016/S0164-1212(97)10009-7
36. Lee, K. D. 2014. *Python Programming Fundamentals*. 2nd ed. 2014. London: Springer London [online], [Referenced 20.8.2019]. Available from http://ezproxy.cc.lut.fi:80/login?url=http://dx.doi.org/10.1007/978-1-84996-537-8
37. Rai, R. 2013. Socket.io Real-time Web Application Development. Birmingham: Packt Pub [online], [Referenced 20.8.2019] Available from https://app.knovel.com/web/toc.v/cid:kpSIORTWA6/viewerType:toc//root_slug:socketio-real-time?kpromoter=marc
38. Yegulalp, S. 2018. Why you should use Docker and containers [online], [Referenced 25.8.2019]. Available from ihttps://www.infoworld.com/article/3310941/why-you-should-use-docker-and-containers.html
39. International organisation of standardisation. 2019. ISO 8601 Date and Time Format [online]. [Referenced 31.8.2019]. https://www.iso.org/iso-8601-date-and-time-format.html
40. MySQL server documentation - 11.4.3 The BLOB and TEXT Types [online], [Referenced 29.8.2019]. Available at https://dev.mysql.com/doc/refman/8.0/en/blob.html

APPENDIXES

Appendix 1. Configuration file for docker-compose

```yaml
version: '3'

services:
  dbmysql:
    image: mysql:5.7.8
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: example
    ports:
      - "3306:3306"
  webcedibilityapi:
    build: .
    command: python3 main.py
    volumes:
      - .:/code
    ports:
      - "5000:5000"
    depends_on:
      - dbmysql
```

Appendix 2. Docker configuration file of data integration system

```
FROM python:3.7.3
ENV PYTHONUNBUFFERED 1
RUN mkdir /code
WORKDIR /code
RUN apt-get update
COPY requirements.txt /code/
RUN pip install -r requirements.txt
RUN polyglot download embeddings2.en
RUN polyglot download pos2.en
RUN polyglot download pos2.da
COPY . /code/
```

Appendix 3. JSON response from location service


```
{
    "place_id": 2622133,
    "licence": "Data © OpenStreetMap contributors, ODbL 1.0.
https://osm.org/copyright",
    "osm_type": "node",
    "osm_id": 356285216,
    "lat": "60.192033",
    "lon": "24.9455609",
    "display_name": "52-54, Aleksis Kiven katu, Vallila,
Helsinki, Helsingin seutukunta, Uusimaa, Etelä-Suomi,
Manner-Suomi, 00510, Suomi”,
```


(Continues)

Appendix 3. (continues)


```
"address": {
        "house_number": "52-54",
        "road": "Aleksis Kiven katu",
        "suburb": "Vallila",
        "city": "Helsinki",
        "county": "Helsingin seutukunta",
        "state_district": "Etelä-Suomi",
        "state": "Etelä-Suomi",
        "postcode": "00510",
        "country": "Suomi",
        "country_code": "fi"
    },
    "boundingbox": [
        "60.191933",
        "60.192133",
        "24.9454609",
        "24.9456609"
    ]
}
```


Appendix 4. Example POST package from sensor to REST API


```
{
    "lat":60.192059,
    "lon":24.945831,
    "sensor_purpose":"earthquacke",
    "sensor_timestamp":"12:00",
    "sensor_data":"true"
}
```