An Integrated Information Lifecycle Management Framework For Exploiting Social Network Data to Identify Dynamic Large Crowd Concentration Events in Smart Cities Applications

George Kousiouris[1], Adnan Akbar[2], Juan Sancho[3], Paula Ta-shma[4], Alexandros Psychas[1], Dimosthenis Kyriazis[1] and Theodora Varvarigou[1]

[1]Dept .of Electrical and Computer Engineering, National Technical University of Athens, 9, Heroon Polytechniou Str, 15773 Athens, Greece
[2] 5G Innovation centre (5GIC), Institute for Communication Systems, University of Surrey, Staghill Campus, Guildford, UK,
[3]ATOS Research & Innovation - Internet of Everything Lab - Av. Capuchinos Basurto, 6, Bilbao, Spain
[4]IBM Research, University of Haifa Campus, Mount Carmel, Haifa, 3498825, Israel

***Abstract*—With the current availability of an extreme diversity of data sources and services, emerging from the Internet of Things and Cloud domains, the challenge is shifted towards identifying intelligent, abstracted and adaptive ways of correlating and combining the various levels of information. The purpose of this work is to demonstrate such a combination, on one hand at the service level, through integrating smart cities platforms for user level data, and on the other hand at Complex Event Processing, Storage and Analytics capabilities together with Twitter data. The final goal is to identify events of interest to the user such as Large Crowd Concentration (LCC) in a given area, in order to enrich application level information with related event identification that can enable more sophisticated actions on behalf of that user. The identification is based on observation of Twitter activity peaks compared to historical data on a dynamic time and location of interest. The approach is validated through a two-month experiment in the city of Madrid, identifying LCCs in sporting events around two sports venues and analyzing various approaches with relation to the needed thresholds definition.**

Keywords—Internet of Things, Social Networks, Analytics, Cloud Computing, Event Identification, Smart Cities

I. INTRODUCTION

In the current technological landscape, the major challenge is to combine versatile data sources in an intelligent manner, integrating and reasoning in order to infer situational awareness. Thus one can transcend from the level of raw data to the level of knowledge and wisdom (according to the DIKW pyramid[1]). Especially with the advent of the Internet of Things, these data sources are expected to reach on an unprecedented scale[2], therefore enabling the optimization of multiple domains of an individual's day to day activities[24]. Among these domains, applications in

transport such as alerts to passengers, smart cities transport, crowd and traffic management are among the key identified aspects.

However, one of the major challenges is the ability to integrate these sources and to reason in order to exactly harvest the added value from this large availability of data ([2],[33]). On one hand a clear and repeatable process needs to be defined in order to link the different elements of such a system, catering for a decoupled integration approach, while a specific added value and context needs to be derived by each case specific analysis of the data that might prove useful to the recipients at the application layer. The need for a concise and multi-source big data and analytics framework for addressing current challenges is critical[21].

In order to design and implement such an approach, a set of further fine grained requirements need to be defined:

- Flexibility in terms of incorporating diverse interfaces and protocols in order to communicate between the different subsystems of different nature and scope, thus providing the combinatorial nature of data analysis and correlation from multiple sources (**Req. 1**)
- The produced information should be asynchronously sent in a notification manner, so that there is dynamicity in the receipt of information only for the case of interest (**Req. 2**)
- A fully automated and dynamic process should exist in order to be able to define arbitrary points of interest (**Req. 3**)
- The analysis performed based on the previous point should take under consideration the context of the given point (e.g. in terms of the specific location or timeslot of interest), adapting to each case in a fine-grained manner (**Req. 4**)
- The implementation should be offered as a service, API or other pluggable form for maximum flexibility and decoupling of implementation details (**Req. 5**)

The aim of this work is to present such an approach that includes the following main points:

- Integration between smart city data, coming from a specific passenger monitoring system in the city of Madrid (Reactivebox[8]) and social network data coming from Twitter, for alerting passengers with special needs (e.g. children, elderly, persons with cognitive restrictions etc.) and their caregivers about Large Crowd Concentrations (LCC) along their journey. This will aid them in avoiding confusion and reduced mobility circumstances when getting off at an area where an unexpected amount of people is concentrated. Novelty of the contribution in this case is inserted through the usage of a general purpose data source (such as Twitter data) for reasoning and inference on the state of a city region through an approach that does not require extensive knowledge of e.g. natural language processing or other complex approach. 5 different approaches of threshold investigation are identified and analyzed in terms of their accuracy and fit-for-purpose, based on the number and type of errors. Another aspect of novelty is the combination and fusion of information from one application domain to another in a cross-verticals integration, that demonstrates the value of diffusing knowledge externally to its main domain of usage, thus minimizing silos of data and exploiting multi-source data correlation. Thus a higher level of cognition and knowledge may be achieved, indicating the way towards numerous others combinations (and more importantly, state of mind to perform the combinations) that may be performed in the context of Smart City applications.
- Ingestion of the integrated data in scalable Cloud based solutions (Openstack Swift) integrated with analytics tools (Apache Spark) for directly working on the acquired datasets and Complex Event Processing tools in order to monitor and issue alerts from streaming data. The solution is able to adapt to heterogeneous data by adjusting the specified Apache AVRO template used for annotation. Novelty of the contribution in this case is represented through the use of specialized tools per case, which enables the exploitation of each tool's powerful features and focus on a specific domain, instead of general purpose tools that could ease integration aspects but would come in the cost of reduced functionality. Furthermore a contribution of this work is the coordination of a large part of the process (in the knowledge

extraction and on-line identification) through a graphical, web based tool like Node-RED which may enable multiple roles to interact with the produced system, since it minimizes the entry level knowledge needed for each layer and gives the ability to interact based on message formats and production/consumption of event information. Furthermore, the created flows can be easily copied and adapted to new cases of interaction.

- Connection of all these systems with an application based logic, achieved through a middleware layer based on Node-RED, in order to orchestrate the necessary actions in the foreseen data flow and provide the necessary adaptations in terms of protocols and data formats. Novelty of the contribution in this case is represented through the successful usage of Node-RED, a tool primarily used for interconnection in the IoT domain, as a general purpose integration and application logic mechanism, exploiting its powerful abstraction and intuitive usage features to speed up development and integration, while adapting to a multitude of protocols (indicatively DDP, MQTT, AMQP, REST, SQL), asynchronous, event-driven logic, push and pull acquisition models and different data formats (XML, JSON, and overall 4 different data schemas). Thus a contribution of this work is the proof of concept that this tool can be used also as a middleware layer in the context of Smart Cities, and not only at the level of Things.
- Fine grained analysis of a specific location needed by a specific user in order to identify a specific event such as a Large Crowd Concentration (LCC), through dynamic receipt and analysis of the respective data that can aid in this identification. The novelty of the contribution in this case is that there is no static division of the city in regions but in each case the adaptation is performed dynamically with relation to the specific user's route and the specific limited area of interest (such as the drop-off point of the journey). Such fine grained analysis enables enhanced accuracy and optimal adaptation at the user level, individualized focus and personal scope, a key feature of Web 3.0 applications.

The relation between the DIKW pyramid, the various levels of information, the involved technology enabling layers and the mapping on Twitter messages processing are presented in Figure 1. It is necessary to stress that while this LCC analysis is performed in the context of the specific application, its consumption may also be extended to other cases in which an LCC event could be the target of a specific action (e.g. police engagement and monitoring, marketing approach etc.), based each time on the scope under which the LCC event is consumed. Furthermore, the methodological aspects may be replicated in different types of events, by adjusting the baseline ingestion of data and extraction of knowledge.

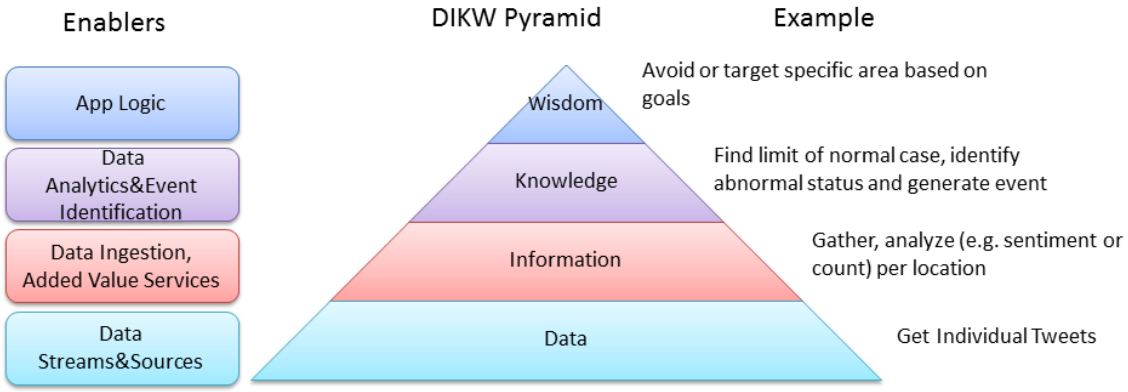


Figure 1: DIKW pyramid related to the specific analysis

The paper proceeds as follows. In Section II related work is presented with accordance to the usage of social networking data and the technologies used in this work, while Section III

describes the application context and the general structure of the implemented prototype. Section IV aims to validate the usage of the framework in an identification experiment including sports events as a source of undisputable gatherings and analysis of the related tweet patterns, while Section V concludes the paper.

II. RELATED WORK

Given that the matters investigated in this work are cross-domain, numerous technologies and approaches exist in each field. In this section the major points of interest will be investigated. What is common ground is the need for a repetitive and methodological approach with a clear focus on the middleware and modelling approach[22]. Citizen-centric approaches should be the main target of the respective applications ([25],[26]), in order to aid in the individualization of services, while multiple sources of social network data can be used together with analytics services for a variety of scenarios[27]. The ability to use social network data as a source of crowd concentration in order to guide tourism services (or in general smart platform services) has been reviewed with success in [40], while critical challenges of data ingestion, spatial analysis and visualization are included in [41].

Initially, in terms of what types of events are of interest, the results from end user surveys were taken into account[7], that demonstrated the need for identifying large crowds, especially in the case of passengers that have some form of limited mobility due to age (children or elderly) or other limitations (cognitive or mobility based ones). In this case, one of the major concerns of their caregivers was the possibility that these individuals, while traversing the public transport network on their own, will encounter such concentrations that may cause anxiety or limited mobility conditions due to the population density. This is the main reason for selecting to identify the specific LCC event. As the main source of data for this case, social networks (SNs) have been identified as the most optimal one due to their extended uptake and dynamic nature. Through these systems significant inference may be achieved with relation to circumstances affecting the societal status, given that almost in all cases these networks offer APIs (e.g. [3]) with which this information may be harvested and further processed. Following, a number of approaches for identifying events based on SN data are identified.

In [11], the approach is based on machine learning and the scope is to categorize incoming social data to identify events or newly defined ones, as a means to better organize information (thus making it more retrievable) without end user intervention (e.g. in the form of relevant event tags). The main issue from this point of interest is the fact that potentially these events may scale in number and differ in context, thus making supervised learning methods to a predefined number of categories insufficient. Initially the existing events are ranked and retrieved based on a set of similarity indices (e.g. upload time, geographic location and textual analysis etc.), that is used to limit the number of retrieved events. Then the SVM classification is performed on the same indices used for the initial retrieval, but in this case between the data item features and the event features. It is necessary to stress that the event notion is not clearly defined as to its granularity, which seems to be rather fine grained. With relation to the LCC case, the aspect seems to be too fine grained since the LCC aim is not to correctly identify the event but the concentration of people in an area. Thus a split between more fine-grained events might imply underestimated numbers for population concentration. Furthermore, in the LCC case the interest is also on how many tweets have appeared, which is not taken under consideration in this approach.

In [12], correlation and aggregation of content related to a planned or (more importantly) an unplanned event across different social media streams is investigated. A time decay function is used for the effect of time on the amount of information created as well as a set of indicators (like time, location, content clustering etc.) to reduce noise or inconsistencies in the way the information is reported. One interesting addition is the usage of URLs in the content to identify data that refer to the same event, investigating the URL differences based on the source and how

these may be helpful or introduce noise. In [10] an approach for planned events is described based on content aggregation from different social networks, focusing on query optimization for retrieving related data and matching them to the defined events. The goal is to retrieve the top-k relevant social documents related to an event from different sources and rank them with specific attributes per source. To this end, relevant queries are identified and diverse aspects of term frequency analysis are investigated, such as ratio of the documents in two given time intervals before and after the event. Again in these approaches the focus is on analyzing the actual event, without having an aspect of quantifying the number of people attending.

In [13] the authors utilize social network data for finding the city centers limits, that in modern polycentric metropolitan areas may be spatially distributed and interacting. This information is critical for urban planning, policymaking, resource allocation and traffic monitoring but also for understanding user mobility patterns across their prominent locations (e.g. home, office, supermarket etc.). Three different clustering algorithms are tried out, LGOG for high and low values, DBSCAN for density based clustering and Grivan-Newman community detection based on network or graph theory. One aspect that is recognized is the need to configure these methods in terms of a specific parameter as well as some knowledge of the city's landmark locations that may be used for the clusters. Another case is the representation of users, since young and more tech-prone users are more represented in the used samples. However, this can be seen as an advantage in some cases, since identifying different data sets for different demographics may prove useful to cases where specific target groups need to be identified (e.g. marketing campaigns against specific groups). While this is an interesting approach, the clusters can be used in a more static nature, not reflecting the current status and real time information. What is more, separating into static clusters does not aid the adaptation to the specific points of interest for a traveler (e.g. specific drop-off location monitoring) and can be considered too coarse-grained for the specific application context in this paper.

In [14] the approach focuses on geographic and time aspects to gather tweets around a specific area, for identifying events in the selected location. The presented method is based on the statistical significance of reoccurring terms in space and time and therefore the event is identified as a composition and aggregation of content of several tweets. More specifically, an event is defined "as a term, which occurs significantly higher in a certain local area than in other areas of the world and significantly higher in this local area for the current time frame than in a past time frame. Therefore, an event term has a specific location where it occurs and a specific time when it occurs". Initially a filtering is performed based on tweet location and then text processing is performed, exploiting nouns and hashtags. Frequency of term appearance is the main metric used in order to identify a local event, especially in the form of burst appearances. Furthermore, to strengthen the local nature of the event, this burst should not appear in areas outside the investigated one. While very interesting for fine grained event identification, the specific case does not attempt to evaluate the size of the event with relation to the normal case in the given area.

A more complex approach appears in [15], in which authors initially investigate neighborhoods of the social network (based on individual nodes attributes such as influence) in order to identify local anomalies by relying on e.g. time characteristics (hour of day etc.). Following, subgraphs are identified that maximize the anomalies in network clusters. Finally, for each such cluster, details on the event are extracted (e.g. location, time, participants etc.). The benefit in this case is the lack of parameters that need to be fine-tuned and the unsupervised learning methods applied. This aspect analysis can be used for an identification type such as LCC.

In [16] a very interesting approach is presented, in which social networks are not only used in order to identify events but to correlate the effect of a specific action to a situation or state. Machine understandable structured data are created through this approach in order to aid individuals achieve their goals. Timelines are investigated, as well as precedent and subsequent events, thus combining the actions with their effect and inferring positive or negative results. As

an example, correlations are performed between user tweets mentioning personal best records in running events and their previous tweets indicating details of their preparation. This approach could be used as an extension in this work in order to correlate the effect of an LCC event to another situation, e.g. traffic state.

In [17] identification of sub-events is performed during emergency situations, based on Flickr and Youtube data, by exploiting the metadata tags of the data items (photos and videos). Time and geolocation are not considered in this case as inputs given that they are usually not inserted by users. Sub events are identified through clustering and aid in the management of the crisis. Self-organizing maps are used for clustering, based on frequency-inverse document frequency (tf-idf) of relevant words based on all including documents. The metadata which is considered for each multimedia document are title, description, and the corresponding tags. User inserted keywords are used as a first filter and following the list of subevents is created (e.g. different buildings damaged during an earthquake). Analysis is performed via labelling and prioritization and a summary of the material is presented to the responsible person. The lack of time and location is a significant drawback for the case of LCC identification.

In [38] an approach is presented for identifying crowd sentiment together with monitoring data, in order to identify a specific crowd's state of mind and approach. This may be applied as a next step to the work in this paper (after large crowd concentration and in order to identify what type of crowd this is). The same approach on sentiment analysis through categories and classification, as well as evaluation of credibility for Twitter data is included in [39]. [42] initially investigates and finds a correlation between passenger flows in subway stations and social media posts, in an attempt to predict these flows through the metro system. Event detection is then based on hashtags identification, and feeds into a regression and time series model for predicting the next values of the flows. In [43] an analysis of spatio-temporal distribution of social media activity with relation to characterized areas (such as residential, commercial, mixed etc.) is performed in order to indicate differences in population sizes across time. However this analysis shows typical patterns that exist (except for the case of the mixed areas that indicate a stable behavior). Thus an analysis as the one presented in this paper for unexpected concentrations is an addition that can help indicate abnormal circumstances. [44] combines social network data with traffic network data in order to enhance information towards travelers regarding road conditions and state. Twitter data in this case are used in order to find individual trajectories, as well as cluster travelers to flows, indicating also abnormal flows with relation to the ones identified from the traffic data. Clustering is also performed around the content of the tweets to relate them to social activities. One key finding is that 46% of the abnormal behaviors is attributed to specific social events (an aspect of which can be considered to be the sporting events used in this work).

What can be concluded from the aforementioned works is that there is a strong trend of using social network data for more and more use cases outside their original purpose (traffic management, public transportation systems management as well as marketing and crisis investigation), thus empowering new and interesting features in event detection and city state analysis. From this work's scope, the need is not to go into the analysis of what type this event is, but primarily correlate it with relation to time and location for an adapted area of interest. However, an out of the box sentiment analysis is included in our case in order to be used for future work or for the need of more fine grained analysis. The type of processing of the retrieved data can be easily changed through the proposed approach with the alteration of the Spark script used for processing at the basis of our implementation.

In terms of analytics, numerous approaches exist in terms of databases and analytics frameworks. One of the most prominent ones is Apache Spark [6], which has gained significant interest in the Big Data community due to its in-memory capabilities and superior performance compared to Hadoop[23]. Like Hadoop, Spark supports Map Reduce style computations while developing the notion of a Resilient Distributed Dataset (RDD) a read-only dataset which is partitioned across multiple machines and can be rebuilt if some partitions are lost. RDDs can be

cached in memory and can be reused in multiple Map Reduce like operations. This can give significant performance improvements for certain classes of applications compared to Hadoop style Map Reduce, which writes all intermediate results to disk. For such applications Spark can outperform Hadoop by an order of magnitude. Spark has also included SparkSQL, a component which enables accessing Spark data with an SQL interface using a notion called DataFrames. DataFrames are essentially RDDs with schemas, and having the schema allows both the use of an SQL query language as well as query optimizations and improved memory management. Apache Flink is another approach that can be mentioned, especially due to the fact that it includes a variety of diverse functionalities. In this case there is a trade-off, such all-inclusive tools on one hand include easier integration into the application layer but on the other hand may not be optimized on specific features of a specialized tool for part of the functionality. For example, the CEP engine used in this work (μCEP) is more abstract in the sense of the expression language, while it focuses on a lightweight implementation for inclusion also in reduced capabilities hardware (such as the ones commonly found in the IoT domain). This of course comes at an extra integration cost, which in our case however is reduced through the inclusion of Node-RED as the intermediate integration and middleware framework. A comparison of various systems can be found in [5] and [4]. Spark shows a clear advantage in the case of the delay included in the machine learning libraries implementations, as well as extended maturity. On the other hand Flink, as a newer tool, is considered less mature but was originally designed to cover for the inefficiencies of Spark, therefore it may be expected to overcome it in the next years. In general Flink and Apache Storm (another similar stream processing system) appear to have lower latency but also lower throughput capabilities.

In terms of data delivery and message processing/distribution systems, a variety of options exist, including Apache Kafka, RabbitMQ (based on AMQP) and MQTT, each of them having specific strengths and weaknesses ([36],[37]). For example, Kafka is considered as the most throughput-oriented and commercial grade system (used by Twitter itself to cover large volumes in message delivery), while MQTT is considered as the most lightweight, commonly used in more IoT-focused solutions, with a very good payload-to-overhead ratio. RabbitMQ is in the middle, having a robust implementation focusing on guaranteed delivery and lower latency. In our work, all of the above are used, depending either on external dependencies or the position of the message processing task. For example, RabbitMQ clients are integrated for pushing relevant events to the Smart City platform, given that this is the input layer of the specific platform. Kafka on the other hand is used for the main raw data ingestion (horizontal line in the lambda architecture image of Figure 3), given the requirement in this position to handle large volumes and rates, while MQTT is used for the cross components data sharing path when data are more condensed (such as the case of identified events shared between components in the ILM, rules definitions etc.), coming from the vertical line of the lambda architecture, and thus message rates are much lower since they involve compressed, event-based information.

In terms of architecture, typical implementations ([28],[29]) follow a lambda architecture, the approach also followed in this work. The main difference is that in this work the incorporation of an abstracted middleware level based on Node-RED enables the abstraction, repetition and easy communication between the various components in terms of reusable flows, changing only the required templates of information (data formats and models and streaming rules templates).

Following, the overall system details will be provided as well as how the system is structured based on the specific requirements posed by the application and based on the scope of the event identification process.

III. SYSTEM DESIGN AND PROCESS

A. Application Use Case and Identified Functionalities Needed

In the context of the COSMOS project and in the framework of a Smart City/Mobility application, the public bus transport authority of Madrid (EMT Madrid) has enabled Route Monitoring of people with special needs (such as elderly, children, people with cognitive disabilities etc.) when travelling across the city using bus lines. Through this application their caregiver may design the planned journey across the city but also monitor in real time the realization of this journey through the respective platform (ReactiveBox[8]).

During the realization of the journey by the passenger, the caregiver needs to track and monitor the progress and be alerted of any abnormal situations that might affect the journey. One of the key disruptions is the existence of large crowds that may arise along the route and near the time of implementation of the journey. Thus the purpose of the LCC identification process is to monitor key locations across the user's journey (e.g. drop-off point etc) and notify caregivers in case an LCC event is identified (Figure 2). This asynchronous notification may be used by the latter in order to alter the planned journey of the user (e.g. get off one stop before or after the initial destination) in order to avoid congestion, confusion and limited mobility circumstances. Furthermore, this process should be adapted to each individual passenger, for their specific plan and the real-time implementation of this journey.

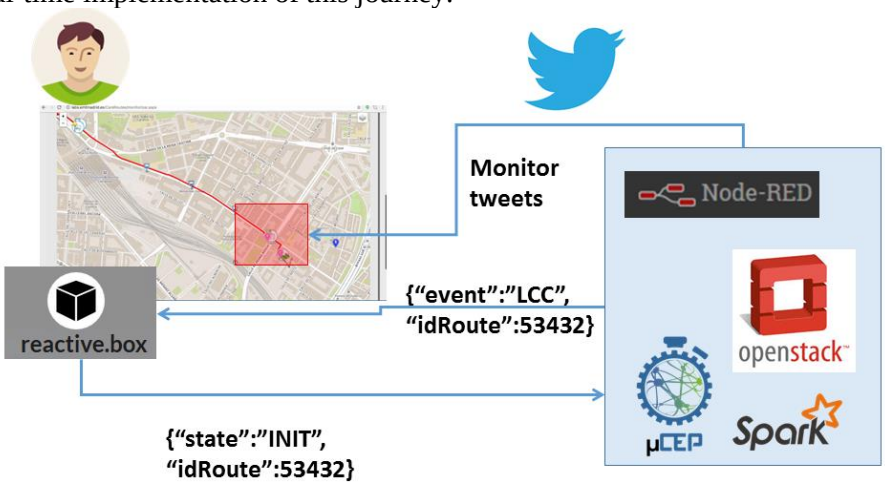


Figure 2: High Level View of Interaction

From this description, one can pinpoint the goals that exist for adaptation in this case and are:

- Goal to identify large, unexpected concentrations of people, which is one of the main needs, as indicated by user surveys[7], for which the specific user category needs to be alerted. Therefore the usage and validation of such a data source that can aid in the indirect identification of a Large Crowd Concentration in a dynamic and real time manner, enabling streaming and geo-localized filtering in order to monitor the location of interest to the specific user, is needed.. Given that Twitter covers these initial requirements, it was selected as the primary source, however a validation needs to take place in order to know if it can be used and with which threshold definition approach. Thus, historical data need to be captured and analyzed that will dictate the thresholds of the activity and indicate the difference between a normal and an abnormal case. Thus a suitable data ingestion and analytics process needs to be defined.
- Goal to integrate between the different systems and the way they expose/consume information. Given that especially in IoT a large number of diverse systems usually exist in a typical environment, with heterogeneous nature, protocols, scope etc., it is imperative that an

adaptation mechanism needs to be in place for obtaining and implementing the links to the systems. This relates initially to a link between the Application Layer (the aforementioned ReactiveBox system of EMT Madrid) in order to receive user level information such as planned journey route, start of journey etc. The specific subsystem is implemented in ReactiveBox via a Meteor server, thus a bridge to the Data Distribution Protocol (DDP) used by Meteor needs to be in place. Furthermore it relates to a link to the Application Layer (ReactiveBox system) in order to push notifications towards the user and caregiver with relation to the identified events related to this specific user. The specific subsystem is implemented in ReactiveBox via an AMQP endpoint, thus a bridge to this protocol needs to be in place. Furthermore, other adaptations need to be in place, related to how the results of an analysis mechanism are propagated to the online event identification mechanisms etc.

- Goal to implement the process of application logic definition through an adaptable, easy to use approach that can be extended and repeated at will with minimum information about the inner workings of each layer, while giving the ability to work directly at the data in order to meaningfully combine information from multiple sources to achieve the application goal. Thus a suitable middleware layer is used in order to link them, abstract from the specifics of each layer and glue all the information together so that it serves a unified and specific application purpose. The need for a sophisticated middleware is identified as one of the pillars of next generation smart city applications[22].

B. Information Lifecycle Management Process

Given the requirements set in the previous paragraphs, the main Information Lifecycle Management (ILM) Process is implemented via a lambda architecture ([20], [28], [29]) for large scale IoT applications (Figure 3). It provides an efficient way to store large IoT historical data in a Cloud Storage layer (based on Openstack Swift) and carry complex analytic tasks using Apache Spark and at the same time provides the functionality to process real-time data streams accurately and detect complex events in real-time, using a lightweight μCEP engine[32].

What is important is that it enables fine grained definition of the template CEP rule that is used to identify an event. Then the respective concretization of the thresholds can be based on individualized parameters that can be part of the respective queries towards the stored data in an automated manner. This enables application developers and city administrators to automatically detect them. An IoT application with rules and conditions set with static thresholds will suffer severe performance degradations due to the dynamic nature of the analyzed environment. Every area in the city has a different nature and pattern of Twitter activity (based e.g. on population density, type of suburb etc.) thus one needs to adjust accordingly the threshold values for when an LCC event alarm is triggered, as will be shown also in the experimentation in Section IV.

In order to perform this adaptation, historical data may be used combined with methods from statistics or Machine Learning in order to find optimized threshold values, which will be ingested in a specific μCEP Engine instantiation for a given location and user plan.
Following, details on the specific layers are portrayed.

1) Historical data ingestion

As mentioned in the previous paragraph, historical data are needed in order to analyze each location in terms of relevant tweet numbers for identifying a peak in activity that signals an abnormal LCC event. In order to ingest the data, and based on the generic ILM flow of Figure 3, initially the data need to be obtained. In this case the registration to the Twitter API is performed via a suitable Node-RED flow and the respective tweets are forwarded towards an intermediate messaging structure, the COSMOS Message Bus. Registration is performed for the entire bounding box of the city of Madrid and only for those tweets that have enabled geolocation. Given that tweets include a number of fields, not all of which are needed, in order to save storage

space, suitable cropping and filtering needs to be performed. Based on following stages needs for performing queries with geolocation and timeslot of appearance, the maintained fields include twitter userID, coordinates of the tweet, tweet text (potentially useful for sentiment analysis in the future) and timestamp of appearance.

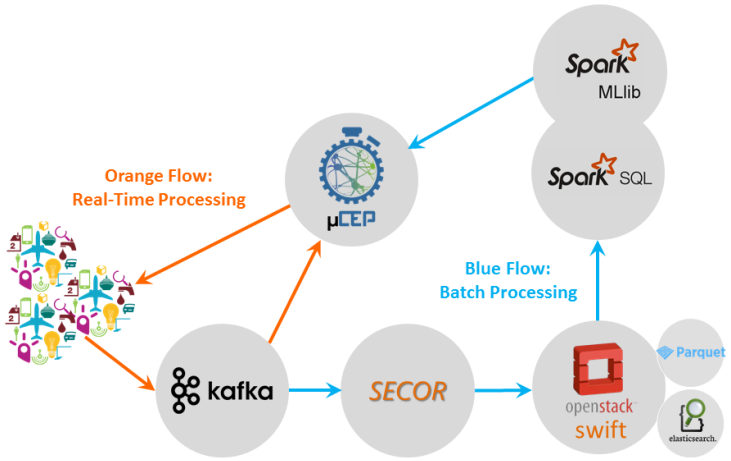


Figure 3: Information Lifecycle Management flow (orange is real-time flow, blue is ingestion and analysis flow)

2) On line usage

As mentioned in the general description of the ILM process, what is important is for the analytics process to be performed in an adapted manner to the specific user plan and points of interest (e.g. drop-off point of the journey). For this reason, during the online usage of the system and at the time the notification is sent for the start of the user journey, the sequence needs to be started in order to calculate thresholds for the drop-off point. For this reason, analytics job submission needs to be exposed through a REST interface, in order to be run for the given timeslot and user id. Then, historical data is accessed from the COSMOS object storage using Spark. Data is extracted and filtered using Spark SQL queries, through a Spark SQL driver which allows filtering the data close to the object storage, before it is sent to Spark. This significantly reduces the amount of data sent across the network. The driver uses metadata search to search for objects containing data relevant to a given Spark SQL query (in this case location coordinates and timeslot). Based on the implemented algorithm of choice, the Spark script returns the thresholds of the analysis to the caller. Once the calculation has finished, a message is sent to the Message Bus containing information regarding the threshold of the investigated box, along with the identifiers of user ID and route ID, in order to be received by the CEP engine listening to the endpoint.

Once the CEP engine receives the aforementioned threshold message through the MQTT feed, it populates a template rule available for the LCC event with the current values for this box and launches the respective rule instance. In order to retrieve the current count for that location, we also need a helper Twitter Counter Service per box, that will be described in the next section as part of the middleware logic. The CEP rule instance checks periodically this Twitter Counter Service for the specific box and generates the respective event. While in this specific case the rule is rather simple, the existence of the CEP syntax and approach enables its usage also in rules with more complex logic.

In this case there is a trade-off between the real-time nature and accuracy of event identification. Starting the analytics job when the journey starts implies that there will be a delay in calculating the threshold (in the experiments described in Section IV the delay was around 3-4 minutes for concluding on the thresholds), thus no event may be identified before this job has completed. However, given that a typical journey in the city public transport lasts approximately 1 hour, this initial delay is not considered important, whereas the complete adaptation to the timeslot of the journey achieved by this launch at start is considered more beneficial. Alternatively relevant pre-calculations may take place, but given that there is no fixed time on the implementation of the user journey, this should be performed for practically every possible timeslot. For speeding up the calculation of the thresholds, sizing/modelling techniques (e.g. similar to the ones in [34],[35]) could be used in order to investigate the resource needs of the specific subsystem.

3) Middleware Logic

Following the analysis of the previous paragraphs, we can identify the list of protocols that the middleware logic needs to support per case/component and cooperation (Table 1). For the main points extra details are given in the following paragraphs.

Table 1: List of Protocols involved for integration per layer

Layer	Protocol	Implementation
User Level System Feed Bridge	DDP to MQTT	Meteor and Node-RED
Analytics Job submission	REST	Spark and Node-RED
Analytics Link to Storage	SQL syntax	Spark SQL driver
Coordination between Spark and NodeRED	MQTT	Node-RED
Twitter Counter Services per box	REST	Node-RED
CEP and Node-RED	MQTT	Node-RED
Notification to User Level App (Reactive Box)	AMQP	Node-RED node with underlying script

a) Data I/O from Application System (Rbox)

In order to support Data I/O from the application system (ReactiveBox) two Node-RED flows have been implemented in order to adapt to the protocols of Rbox. Initially the bridge for receiving notifications (e.g. user start/end of journey etc) based on the DDP protocol includes a DDP implementation in javascript in a general function node, which receives the DDP notifications, filters and extracts the necessary information, pushing it to the Message Bus. Once an LCC event is detected and propagated to the same message bus, the output layer (Figure 4) receives the notification, adapts it to the needed output schema and forwards it to an AMQP client. Internally to our system all the information is forwarded in an MQTT message bus in order to be obtained by all the intermediate subsystems and components.

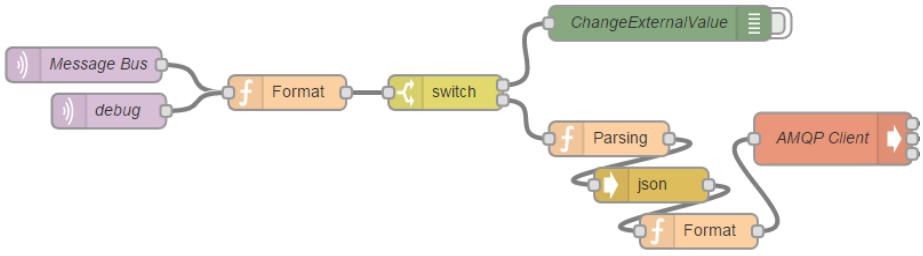


Figure 4: Notification output towards ReactiveBox application system (AMQP protocol)

b) Supporting Counter Services and Registers

For keeping individualized counter structures for each user journey (and the respective monitored location), a necessary service framework needs to be in place in order to register and monitor a given geographical box. This box is calculated from the drop off point of the user journey, including an area of 500 meters around it. The counter service keeps the count of tweets inside that box and offers this information to the interested components (in this case the constant checking of the rule thresholds against the real time count performed in the CEP rule instance for that location).

The service framework for retrieving and monitoring per location the number of tweets appears in Figure 6. The format for a monitored location information includes the boundaries of the geographical box (Southwest and Northeast corner coordinates), the necessary IDs for distinguishing between different routes and users, the current count (used by the LCC rule to get the state) and the registration time. The latter is useful in the case of linear calculation of the tweet evolution in the slot for issuing early warnings by the CEP engine.

c) Coordination/Synchronization Logic

As seen in Section III.B.2 during the online usage of the system, a series of actions needs to be coordinated in order to implement the functionality. In this case what is specifically needed is a coordination layer that will link between the application layer information (coming from the Reactive Box platform), residing at the user level, and the various subsystems mentioned in the ILM flow, coordinating their functionalities for a given user. The main nature of communications is asynchronous. The details of this cooperation (Figure 7) is described below:

- Upon user journey start, the ReactiveBox system sends the notification, that is received by the data I/O layer. These include initially the INIT message of user start and the registration of the points of interest from the user journey (e.g. drop-off location) in the respective Twitter counter filter service (after calculating a 500 meter bounding geographical box around that point). Thus tweets from that box are counted. Following, Spark is notified in order to dynamically calculate the threshold for that box. This is performed in order to enable variations based on the current timeslot.
- Once Spark has calculated the thresholds, these are annotated with the specific journey's ID and are forwarded to the CEP engine in order to launch a rule instance monitoring the progress of the counter against the threshold. Once the threshold is violated a relevant notification is pushed back to the ReactiveBox system in order to warn the caregiver about the identified LCC event. Early warnings can be sent in case the counter seems to be near violation. So for example if the threshold for one hour is set at 15 tweets, and the check during the half hour mark is above 15/2 (linear assumption of tweet distribution) based on the registration time of the box, then the according early warning is issued.
- Upon journey completion, another notification is sent by ReactiveBox that is used to clean up the monitoring process (stopping of the real-time location filter for Twitter and the CEP rule instance)

The overall sequence diagram for the detailed steps described in the previous sections appears in Figure 5.

The produced software is available at [30], including:

- The aforementioned middleware flows, for data ingestion, link with the application layer and Spark REST interface
- The μCEP engine
- The Spark script used for the batch analysis

Additions with relation to Openstack Swift in terms of metadata annotation have been contributed to the respective Swift repository.

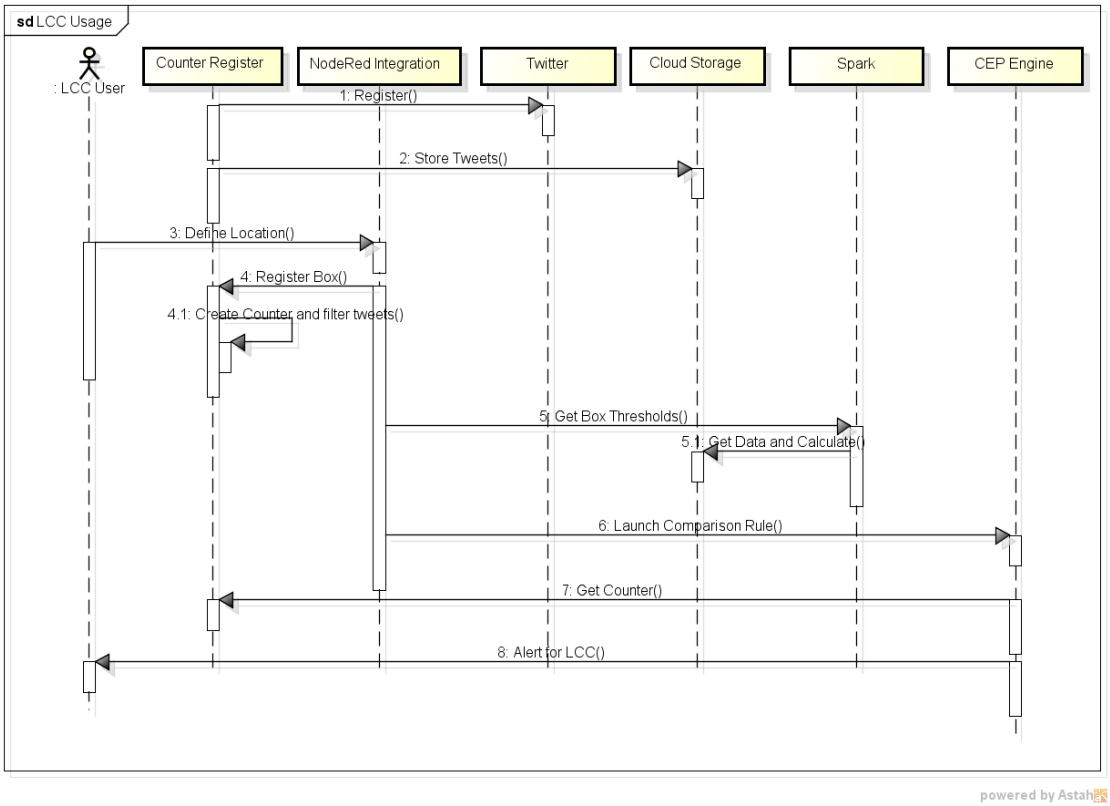


Figure 5: Overall Sequence Diagram for LCC Implementation

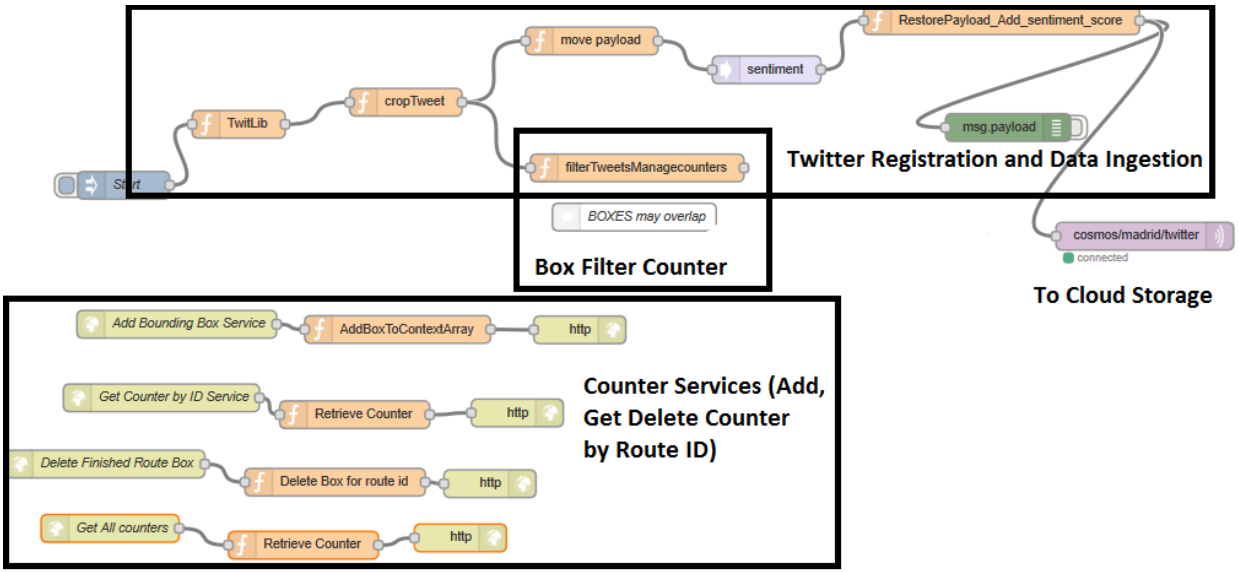


Figure 6: Node-RED implementation of Twitter Count Flows

IV. VALIDATION AND LCC EXPERIMENT IN MADRID

A. Selection of sporting events to identify LCC

In order to validate the operational nature of the system for a long running process and also the approach of investigating whether LCC events can be detected via social network feeds

(and for what thresholds), a 2-month experiment was performed in the city of Madrid (November to End of December 2016), by constantly monitoring the feeds around two sporting events locations. The reason for choosing sports events is that they represent **a well-known LCC event** (e.g. from ticket attendance), thus they can be used for validation. The game schedule is known, therefore one can be sure that the event has taken place at the specific time and how many people attended. Information on the chosen locations is depicted in Table 2.

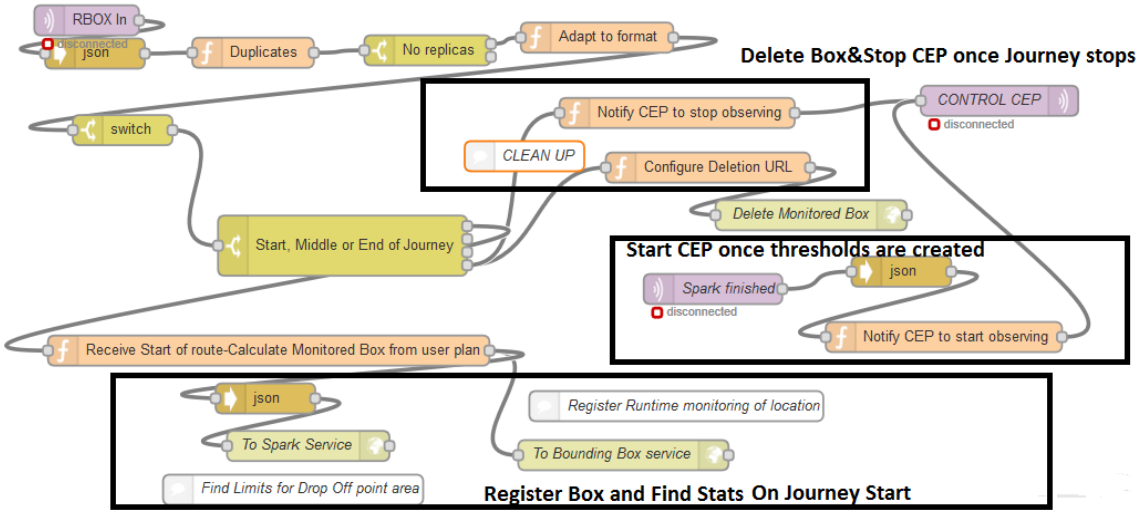


Figure 7: Middleware flow for connecting different layers

Table 2: Stadium Locations

Stadium Name	Location	Teams	Capacity
Santiago Bernabeu (Location 1)	lat : 40.4540469 lon : -3.687452	Real Madrid FC	85454
Vicente Calderon (Location 2)	lat : 40.402617 lon : -3.719667	Atletico Madrid FC	54907

In order to gather tweets, the Twitter Streaming API was used, registering for getting all of Madrid's tweets (based on a wide geographic location around the city). Only tweets with enabled geolocation were obtained. Twitter's policy in the specific API version is not to forward all tweets in some cases but a percentage of them (typically up to 1% of the overall current Twitter feed). However this limit was hardly reached. Indicatively, for a crowd concentration of about 80,000 attendants the number of tweets streamed by the Twitter API is about 40 in its peak (as will be shown in the next sections), which however is sufficient in order to define a peak.

In the monitored interval from November 2016 to end of December 2016, there were 10 scheduled events in these locations, relating to football games. The locations were constantly being monitored for a bounding box of 500 meters around the center of each stadium. Tweets with geolocation inside this box were counted for each hourly time slot, resulting in overall 2042 time slots.

B. Tweets evolution and definition of true LCC slots

Initially, in order to validate if tweet numbers evolution is sufficient in order to identify an event in the area, the respective plot was obtained throughout the previous day of the event. This appears in **Figure 8**, starting from the previous day (noon) and continuing until right after the

event. It is indicative that there is a large spike observed around the start time of the event (12:00 CET on 6/11/2016), which was an early clear indication that LCC could be identified through the observed number of tweets in the venue area.

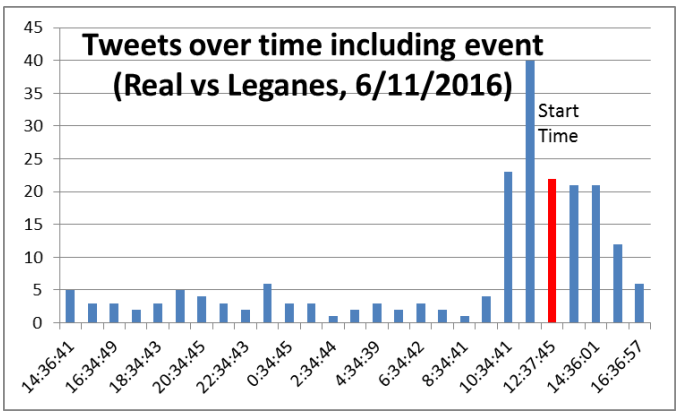


Figure 8: Evolution of tweets in a 24 hour duration before the game day@Bernabeu

However, in order to define which hourly slots around the start time of each event should be considered as true for LCC presence (for comparison purposes during the validation of the identification approach), an analysis of the 3 initial events was conducted, observing the time slots before, during and after the games. From this initial investigation, the aim was to check when the peaks appear and with which relation to the time of the event. The events monitored were "Atletico Madrid vs Malaga" (E1), "Atletico Madrid vs Rostov" (E2) and "Real Madrid vs Leganes" (E3). During this time, the hourly slots were divided and characterized with relation to the start time of the event (E#) in the following categories:

- Before slot (E#BEF)
- Approaching location (E#Appr)
- Final approach &1st half (E#LA1H)
- Half-time break % 2nd half (E#HTEM)
- Departure (E#DEP)
- After slot (E#AFT)

The results from this analysis appear in Figure 9. At this point the interesting aspect is the examined peak of the twitter activity, which is focused on the **Approaching** slot. This matches the needs of the LCC event identification, since it is that time in which traffic or pedestrians in the area will maximize, and therefore it is in the best interest of the user to avoid the specific location. Furthermore, the generic case is that **4 hour slots** are indicative of the event (1 prior to the event, 2 during the event and 1 after the event finalization). Differences in peaks comparing E1 and E2 with E3 are of no specific concern, since they relate to different locations and the adaptation is performed per location for the creation of the specific threshold, as detailed in the next step.

C. Investigation of thresholds and experiment results

As indicated in Figure 9, each location may have a different threshold with relation to the identified peak (or deviation from normal status). For this reason, an adaptation per box needs to be performed, which is done through the queries to get results from the same geographical box. However even in this case, one unified rule needs to be applied in order to handle historical data

and identify thresholds per location. For this reason, a number of analysis possibilities are investigated in this section.

Initially a threshold of mean+1σ is considered, with the mean extracted from the overall data of the experiment, for the given location and timeslot of day. In order to extract the mean, Spark was used, by filtering tweets based on their location (in order to be included in the monitored box), and their timestamp (to be inside the examined hourly slot). Average tweets per time slot of day appear in the following figures (Figure 10 and Figure 11) for the two venues. The large difference may be attributed to the fact that Atletico's venue includes a river and a park, therefore it is less populated normally than the area around Real Madrid's stadium. The slots that exhibit a large standard deviation are typical of game start times. One could further discriminate between week days and week end days for further refinement. Summing the mean and standard deviation gives the applied threshold per slot.

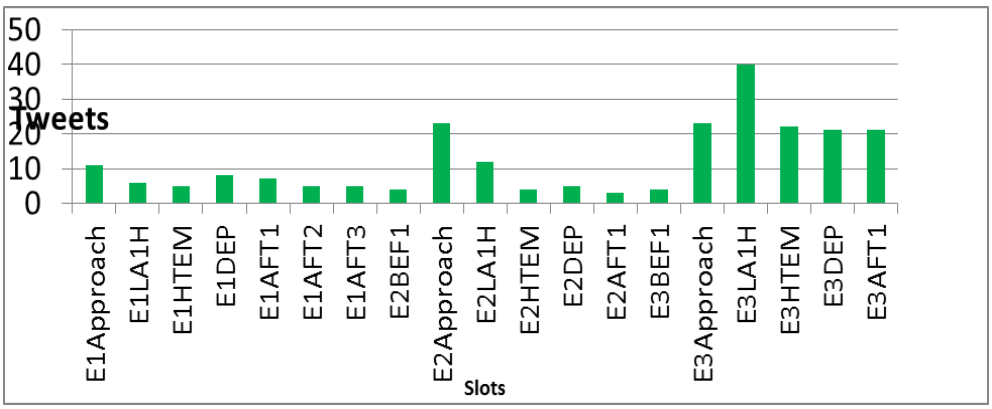


Figure 9: Evolution of tweets with relation to the division of relative slots to the event start time for the three initial events (E1,E2,E3)

The second option is a simpler one, requiring the assembly of all data (regardless of the time slot) for the same venue periodically and extracting the Cumulative Distribution Function (CDF) of the collection (Figure 12). Then the threshold is set to the value that indicates a higher than 95% (or other threshold) probability of appearance. From this analysis it is defined that the boundaries are set to be larger than 3 tweets per hour for Atletico Madrid and 10 for Real Madrid. Variations with 97.5% and 99% were also tried out (setting the boundaries to 5 and 9 respectively for Atletico Madrid and 13 and 28 for Real Madrid).

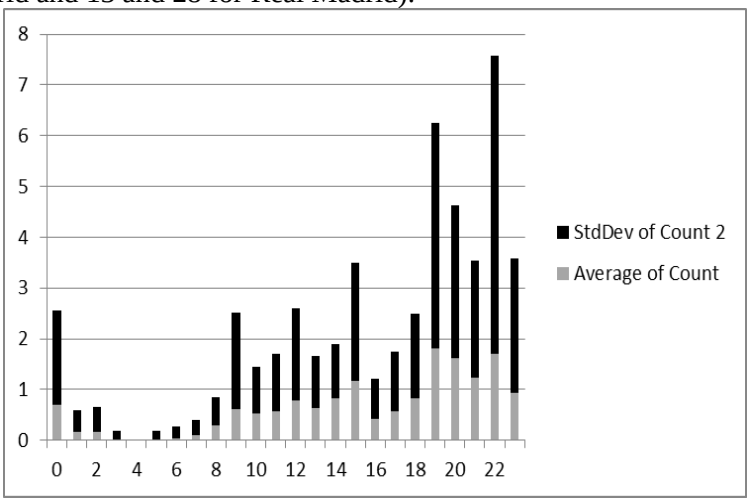


Figure 10: Mean and standard deviation of tweets per time slot of day (0 is 00:00-01:00 and so on) for the Vicente Calderon (Atletico) venue. Threshold for each slot is mean+1σ

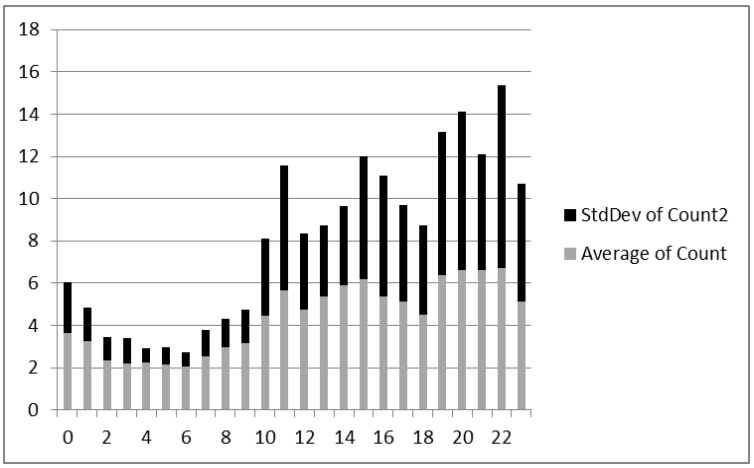


Figure 11: Mean and standard deviation of tweets per time slot of day (0 is 00:00-01:00 and so on) for the Santiago Bernabeu (Real Madrid) venue. Threshold for each slot is mean+1σ

From both approaches it is indicative that each location has its distinct characteristics and the thresholds need to be adapted per geographic location.

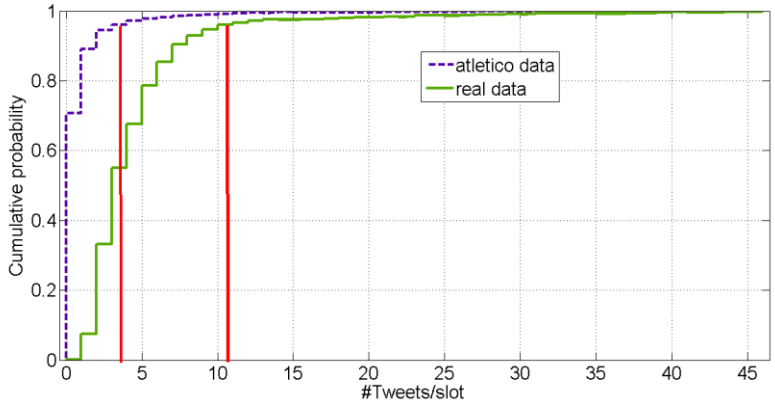


Figure 12: CDFs for number of tweets/slot appearances in the two venues and investigation of limits (red vertical lines) for 95% boundary

In order to evaluate the results of the identification process, the scheduled fixtures in the investigated period were considered (overall 10 sporting events, 5 for each location). For these, and based on the initial analysis of Section IV.B, 4 hourly slots around the start time of each game were considered as the cases that needed to be identified as LCC cases, thus constituting the LCC Positive slots (1 slot prior to the event, 2 during and 1 after). The remaining slots from the approximately 1020 gathered for each venue were considered as the cases in which no LCC should appear, constituting the LCC negative cases. If the approach identifies an LCC case correctly (i.e. the defined slots around the event), this indicates a True Positive, whereas if during these slots it concludes to a non-LCC state this result is characterized as a False Negative If on the other hand the approach identifies a non LCC case as such this constitutes a True Negative, while if it decides that an LCC appears in one of the non-LCC cases this results in a False Positive. The detailed results per variation, venue and category of identification appear in Table 3. For each location a separate row is used for each examined variation of the limit. In the last column the sum from the two locations is portrayed per limit variation used. Columns 2 and 3 indicate the ratio of True Negative/Actual Negative and True Positive/Actual Positive. The FP

and FN columns are complementary of the TN/AN and TP/AP ones respectively, based on the aforementioned analysis, but are maintained for better visualization of the results. Limit investigation variations include three different percentages of the CDF (95%, 97.5% and 99%) and two versions of the Mean+standard deviation limit (direct version and a rounded to the closest integer version).

Table 3: Results for the threshold variations in a 2042 samples dataset containing 40 event slots (based on the assumption that each sporting event relates to 4 slots, 1 hour pregame, 2 hours during and 1 hour after). Each slot has a duration of 1 hour

Variation	TN/ Actual Negative	TP/Actual Positive	FP	FN	Sum from two venues in False identifications
Limit>Mean+1s (Atletico) in1023 slots	932/1003	15/20	71	5	6 FN, 169 FP,
Limit>Mean+1s (Real Madrid) in 1019 slots	901/999	19/20	98	1	
Limit> Round(Mean+1s) (Atletico) in1023 slots	963/1003	14/20	40	6	7 FN, 89 FP
Limit> Round(Mean+1s) (Real Madrid) in1019 slots	950/999	19/20	49	1	
Limit 95% CDF (Atletico) in 1023 slots	980/1003	17/20	23	3	**4 FN, 45 FP**
Limit 95% CDF (Real Madrid) in 1019 slots	977/999	19/20	22	1	
Limit 97.5% CDF (Atletico) in 1023 slots	993/1003	13/20	10	7	9 FN, 17 FP
Limit 97.5% CDF (Real Madrid) in 1019 slots	992/999	18/20	7	2	
Limit 99% CDF (Atletico) in 1023 slots	999/1003	7/20	4	13	24 FN, 6 FP
Limit 99% CDF (Real Madrid) in 1019 slots	997/999	9/20	2	11	

The major criterion for selection is **the minimum false negatives (FN)**, since this is the most dangerous case, especially in the context of usage of the LCC event (as described in Section III.A, this is to warn people with special needs in order to avoid areas of LCC due to potential anxiety, frustration and reduced mobility reasons). In case the event is used in another context (e.g. usage by a marketing company to discover large crowds for promotional purposes), then

potentially the decision changes based on the specific need. For example, in this case large false positives (FP) may imply costly dispatching of the marketing teams for no benefit.
While the best overall performance is in the 97.5% and 99% CDF variations, with 26 and 30 accumulated false predictions, due to the minimum FN criterion it was considered that the 95% CDF variation is optimal. Figure 13 and Figure 14 portray all the counted slot samples in the overall experiment interval, highlighting their relation to the used CDF limits. While the 99% variation captures all the spikes in the events, this relates mainly to the main approaching spike and not the neighboring slots of e.g. during the game or while departing from the game. With relation to the timeslot incorporation in the case of the Mean+1s variations, this does not help in improving the results eventually. FPs in the majority of cases relate to slots not close to events that however have a small but violating number of tweets. This behavior is expected to be improved in a fully operational system with the incorporation of more of these rare occasions as data collection continues.

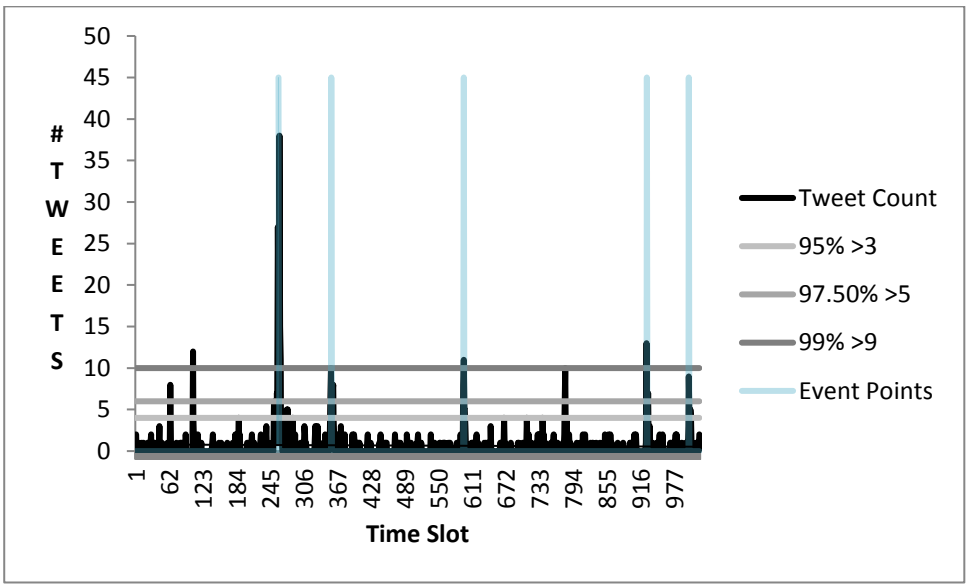


Figure 13: Timeslot counts compared to limits for CDF 95%, 97.5% and 99% for the Atletico case (vertical cyan lines indicate the time of the actual events)

D. Real Time requirements and and investigation of limitations

As mentioned in the Related Work section, various real time focused systems for complex event processing exist such as Spark streaming, Flink or Storm. Therefore the implementation could be based solely on one of these if one needed maximum throughput and scalability. However, one of the aspects targeted at this work was the ability for different roles to be involved in different parts of the value chain, including Data Science experts, application developers (for specific middleware development and combination of diverse data sources), easier definition of rules for event detection (e.g. combination of sensors that could indicate a certain event) etc. Therefore the ability to have abstracted means for them to combine the aforementioned technologies (through Node-RED) was considered more critical than the introduced processing overheads, which is a typical trade-off between ease of use and processing efficiency. This is strengthened by another aspect, the fact that the described approach, through the DIKW pyramid, aims eventually to transcend from the manipulation of large scale raw data to the level of more compressed information through registration for a specific information subset or specific events, whose message rates are considerably smaller. Thus the requirements for massive data processing are significantly reduced, enabling a trade-off of using abstracted environments (like Node-RED) to

make it easier for application developers to integrate different sources and design the business logic. However these requirements need to be pinpointed and the operational limits identified.

In the investigated application context (counting of Tweets per geolocation box), the data ingestion and analysis part, based on commercial grade tools such as Apache Kafka (used by Twitter), Swift and Spark, is considered as able to address wide range of needs in terms of scalability and real time aspects. What needs investigation is the middleware logic and specifically the Tweet counting flows in Node-RED, in the sense of being able to cope with the incoming tweets to be filtered and the number of registered users (users=number of geolocation boxes).

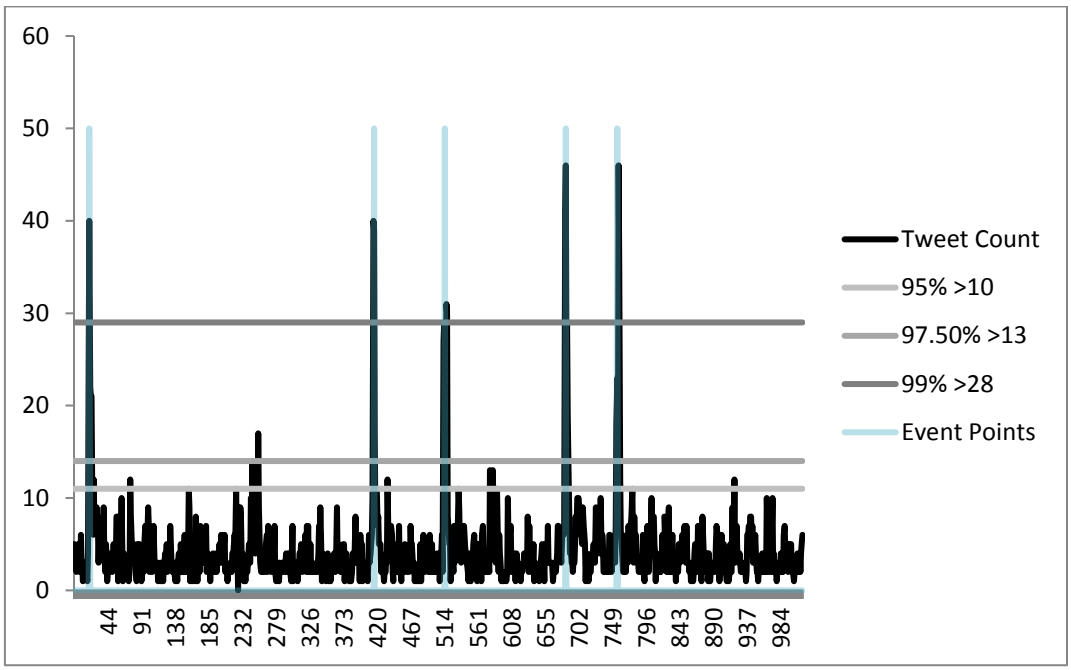
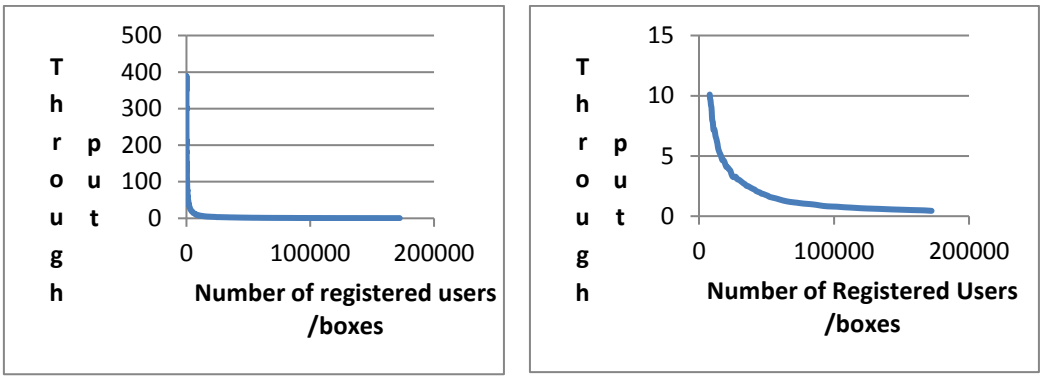


Figure 14: Timeslot counts compared to limits for CDF 95%, 97.5% and 99% for the Real Madrid case (cyan vertical lines indicate the time of the actual events)

In order to benchmark the specific part of the system, a benchmarking message count flow has been created and has been contributed in the software repository[30]. Running in a quad core i7 processor with 6 GB of RAM, Node-RED achieved a maximum 380 messages per second throughput, when measuring the ability to handle incoming messages without any type of filtering. The flow was then adapted to the overall actions that take place in the implemented application flow (including realistic size of tweet string input, processing to determine if the specific tweet is inside the various registered geolocation boxes of the users route etc.) and achieved throughput was measured for an increasing number of registered users/boxes. The graph produced by this experiment appears in figure 15, the overall data in (a) and the focused part on the last 10 messages/second in (b). In order to know to which point of the curve we should focus on, one needs to identify, based on the available data, what was the maximum tweet feed rate received per slot and needed to be filtered. The stream coming from the overall Madrid box included a maximum throughput of approximately 1600 geo-localized tweets per hour (0.44 per second) that need to be filtered by the flow. The number might seem low but it is also aided by the fact that the registration is very targeted (including only Madrid) and performed for the geo-localized tweets. Twitter's streaming API limits received tweets if the registration results in requesting more than 1% of the overall tweet feed, however this limit was not reached during the experimentation and it can be considered rather high given Twitter's amount of traffic per day (in the range of millions).

From the experiment it can be concluded that the highest number of registered users results to approximately 170,000 for Madrid, aided by the low message rate needed. One could also consider merging streams from multiple cities, through a relevant registration to the Twitter API, in which case the maximum needed throughputs per case could be added and a new user limit found (e.g merging 20 similar feeds would result in approximately 10 messages/sec, thus being able to support 8,000 users). Thus any related tradeoff might be achieved between number of cities and needed user support. Potential load balancing strategies may be taken under consideration, since the aforementioned analysis is for a single processing node. As an example, the solution of a RabbitMQ system in front of a load balanced, single-instance-per-city Node-RED environment could be inserted. The redirection may be performed based on a routing key with the city name and a routable RabbitMQ exchange (of type "direct" or "topic based").



(a) (b)

Figure 15: Achieved throughput (messages/sec) versus number of registered users/boxes: (a) includes all data (b) focus on throughputs of 10 messages/sec and lower

E. Visualization of monitoring mechanism

In order to observe the monitored locations, a relevant web GUI was created (Figure 16) using the worldmap node of Node-RED, in which the information is easily forwarded as a JSON message. Different color variations can be applied, e.g. by using red when the current count exceeds the threshold. Additional information is provided like the coordinates of the point and registration time in which the monitoring started, as well as the detailed values of threshold and current tweet count. This mechanism is needed at the system level in order to monitor all the ongoing points of monitoring and ensuring the operation of the system. At the user level, the notification for an individual threshold violation is sent to the ReactiveBox system of EMT Madrid (through the Data output layer described in Figure 4), in order to forward this information to the typical GUIs of the Smart Mobility platform (route monitoring web panel and mobile app).

V. CONCLUSIONS

Concluding, in order to exploit current advances across Cloud and IoT technologies, repetitive and abstracted methods should be created for bridging between sources of information and their processing, enabling the extraction of meaningful data at the application level for maximizing impact.

In this work, such an approach was presented, that links user data, in the form of planned routes and location, coming from a Smart Transportation platform, with social network activity peaks for identifying Large Crowd Concentration events that might affect user journey. This way application dynamicity and richness of information may be enhanced from multiple sources (Req 1 mentioned in the introduction). In order to exploit bulks of data, an Information Lifecycle Management approach based on a combination of Cloud Storage, CEP and Analytics services are

used (abiding by Reqs 3 and 5), that can be easily replicated in new use cases by changing the template of the consumed information. Adaptation is as simple as pushing the respective data in a messaging structure after defining a suitable schema for their structure and creating a relevant Spark script for the specific analysis. This schema is helpful at the later stages for optimized retrieval of the information based on structured queries. The respective implementation is available as a template on the IBM Bluemix Service[31]. Therefore the approach can be easily re-applied in other cases of data ingestion, event production and dissemination to interested parties.

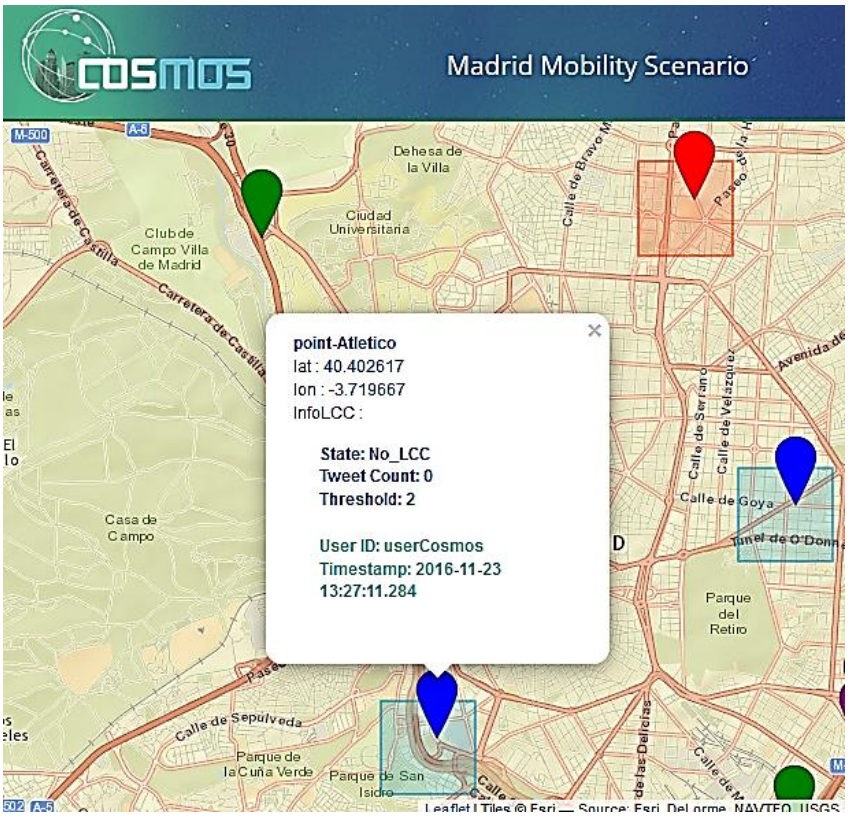


Figure 16: Visualization panel of LCC monitoring

One critical aspect is also the usage of an adaptable middleware layer in order to bridge and coordinate the various elements of the system, based on the Node-RED environment. Especially the latter has the benefit of faster integration, usage of ready-made plug-in nodes for the majority of the protocols used and asynchronous nature, enabling to focus directly on the application logic and the correlation needs, thus fitting to the Reqs 1,2 and 3 posed in the introduction of this work. The middleware flows are available from the Node-RED public repositories[30]. It is indicative that through this approach, 4 different protocols, 2 different platforms and 4 different APIs were integrated (Twitter, ReactiveBox platform, Spark service and internally required services), while 4 different templates of information (Tweet structure, user route and location structure, event description structure and internal monitoring counters structure) were defined, combined and used. Communication was performed through 3 separate messaging structures (DDP, MQTT, AMQP), each time adapting to the external systems and the used technology. Through the flexibility offered by Node-RED, another achieved aspect was the full parameterization of the analysis, including dynamic creation of thresholds and dynamic creation of monitored locations, without the need for division of the city in e.g. static geographical blocks. This aspect enables various entities to interact with the implementation with

a smaller knowledge barrier, without getting into the details of the internal systems and focusing only on the added value of data combination and knowledge extraction.

With relation to social data as an indirect way in order to decide whether a peak of activity is demonstrated in a specific location and based on the statistics and deviation from the norm in that location, the approach is validated in a nearly 2month experiment involving sporting events around two locations. This case study selection was made since the primary need was for guaranteed knowledge that an actual large crowd exists in the area for the experimentation cases and the definition of the actual positive and negative slots. Therefore, the existence of such information (e.g. through published ticket attendance from the sport venue) is a validation that an actual crowd was in the area (and how large this crowd really is). Furthermore, a key feature to sporting events is the existence of a large number of them (and a well-known program of when the event occurs), which is also necessary for gathering a sufficient data set (in comparison for example to other large crowd cases such as demonstrations, strikes etc.). But even in the sporting events case there are some significant challenges to address, such as the shifting time and day of the matches, as well as the fact that we need to distinguish also timeslots around the event (so for example, how long the approach and departure take or affect the nearby area) as well as the limits of the rules to apply. Different approaches have been investigated based on various statistical measures, indicating variable performance in the main categories of identification (TP, TN, FP, FN). Based on the event identified and the context under which it is used, the adopter may select the variation that fits best to their purpose. In our case, and due to the nature of the application and the need for minimal false negatives, the variation of the limit identification that is extracted from the 95% percentile of the tweets CDF is finally chosen, resulting in 49 errors in 2042 validation cases (and only 4 FNs). The overall accuracy can be enhanced by more strict limit specification (26 overall errors but with 9 FNs) The approach is adaptable to different locations via service registrations, thus abiding by Reqs 3,4 and 5. With relation to more generalized conclusions, social network data have been proven useful for identifying large crowd concentration indirectly, a use case that clearly extends their initial microblogging purpose and scope. Furthermore the message rates needed for this are considered low, thus minimizing the need for a large scale solution, while trade-offs between number of cities included in a stream and supported users can be achieved.

For the future one aspect that is worth pursuing is the further refinement of social data in order to distinguish from trends or topics of interest, e.g. by proceeding also at the Twitter user level of granularity. Another feature that would be of interest in order to further fine grain the results of event identification and thus enable more roles to take advantage of it is to utilize the sentiment analysis results directly in the type of events, for more fine grained analysis, extending the scope of the specific implementation to more domains (e.g. dynamic investigation of happy LCCs for use in promotional marketing activities).

ACKNOWLEDGMENT

. The research leading to these results is partially supported by the European Commission's Seventh Framework Program under grant agreement no 609043, in the context of the COSMOS Project.

REFERENCES

[1] Rowley, Jennifer E. "The wisdom hierarchy: representations of the DIKW hierarchy." Journal of information science (2007).

[2] Michael Bradley, Nick O'Leary, "The Internet Of Things", IBM, 2014

[3] Michael Armbrust, Reynold Xin, Cheng Lian, Yin Yuai, Davies Liu, Joseph Bradley, Xian-grui Meng, Tomer Kaftan, Michael Franklin, Ali Ghodsi, and Matei Zaharia. Spark SQL: Relational data processing in spark. In ACM Special Interest Group on Management of Data, 2015.

[4] Chintapalli, S., Dagit, D., Evans, B., Farivar, R., Graves, T., Holderbaugh, M., Liu, Z., Nusbaum, K., Patil, K., Peng, B.J. and Poulosky, P., 2016, May. Benchmarking streaming computation engines: Storm, Flink and Spark streaming. In *Parallel and Distributed Processing Symposium Workshops, 2016 IEEE International* (pp. 1789-1792). IEEE.

[5] García-Gil, Diego, Sergio Ramírez-Gallego, Salvador García, and Francisco Herrera. "A comparison on scalability for batch big data processing on Apache Spark and Apache Flink." *Big Data Analytics* 2, no. 1 (2017): 1

[6] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. (2012), Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (NSDI'12). USENIX Association, Berkeley, CA, USA, 2-2

[7] FP7 Cosmos Deliverable_D7.5.3_Smart events and protocols for smart public transport (Y3 Functionality), EMT Madrid and other partners, June 2016

[8] FP7 Cosmos_Deliverable D7.3.3_Smart Events and protocols for smart public transport (Year 3 Implementation), EMT Madrid and other partners, August 2016

[9] Twitter REST API details available at: https://dev.twitter.com/rest/public

[10] Hila Becker, Dan Iter, Mor Naaman, and Luis Gravano. 2012. Identifying content for planned events across social media sites. In Proceedings of the fifth ACM international conference on Web search and data mining (WSDM '12). ACM, New York, NY, USA, 533-542. DOI=http://dx.doi.org/10.1145/2124295.2124360

[11] Timo Reuter and Philipp Cimiano. 2012. Event-based classification of social media streams. In Proceedings of the 2nd ACM International Conference on Multimedia Retrieval (ICMR '12). ACM, New York, NY, USA, , Article 22 , 8 pages. DOI=10.1145/2324796.2324824 http://doi.acm.org/10.1145/2324796.2324824

[12] Fotis Psallidas, Hila Becker, Mor Naaman, Luis Gravano: Effective Event Identification in Social Media. IEEE Data Eng. Bull. 36(3): 42-50 (2013)

[13] Sun, Y., Fan, H., Li, M., Zipf, A. (2015): Identifying city center using human travel flows generated from location-based social networking data. Environment and Planning B: Planning and Design. (accepted).

[14] Andreas Weiler, Marc H. Scholl, Franz Wanner, and Christian Rohrdantz. 2013. Event identification for local areas using social media streaming data. In Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks (DBSocial '13). ACM, New York, NY, USA, 1-6. DOI=http://dx.doi.org/10.1145/2484702.2484703

[15] Feng Chen and Daniel B. Neill. 2014. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '14). ACM, New York, NY, USA, 1166-1175. DOI=http://dx.doi.org/10.1145/2623330.2623619

[16] Emre Kıcıman and Matthew Richardson. 2015. Towards Decision Support and Goal Achievement: Identifying Action-Outcome Relationships From Social Media. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15). ACM, New York, NY, USA, 547-556. DOI=http://dx.doi.org/10.1145/2783258.2783310

[17] Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. 2012. Automatic sub-event detection in emergency management using social media. In Proceedings of the 21st International Conference on World Wide Web (WWW '12 Companion). ACM, New York, NY, USA, 683-686. DOI=http://dx.doi.org/10.1145/2187980.2188180

[18] Dolce Language Specification v2.0, January 2015, available at: https://repository.atosresearch.eu/index.php/s/NfzIuQeC4e5F0ez

[19] Node-RED Wiring tool: http://nodered.org

[20] Omar Y. Al-Jarrah, Paul D. Yoo, Sami Muhaidat, George K. Karagiannidis, Kamal Taha, Efficient Machine Learning for Big Data: A Review, Big Data Research, Volume 2, Issue 3, September 2015, Pages 87-93, ISSN 2214-5796, http://dx.doi.org/10.1016/j.bdr.2015.04.001.

[21] Martin Strohbach , Holger Ziekow, Vangelis Gazis, Navot Akiva, "Towards a Big Data Analytics Framework for IoT and Smart City Applications", Modeling and Processing for Next-Generation Big-Data Technologies, Volume 4 of the series Modeling and Optimization in Science and Technologies pp 257-282, Springer International Publishing

[22] Johannes M. Schleicher, Michael Vögler, Christian Inzinger, and Schahram Dustdar. 2015. Towards the Internet of Cities: A Research Roadmap for Next-Generation Smart Cities. In Proceedings of the ACM First International Workshop on Understanding the City with Urban Informatics (UCUI '15). ACM, New York, NY, USA, 3-6. DOI=http://dx.doi.org/10.1145/2811271.2811274

[23] Zaheer KhanEmail author, Ashiq Anjum, Kamran Soomro and Muhammad Atif Tahir, "Towards cloud based big data analytics for smart future cities", Journal of Cloud ComputingAdvances, Systems and Applications 2015, 4:2, DOI: 10.1186/s13677-015-0026-8

[24] Nikolaos Panagiotou, Nikolas Zygouras , Ioannis Katakis, Dimitrios Gunopulos, Nikos Zacheilas, Ioannis Boutsis, Vana Kalogeraki, Stephen Lynch, Brendan O'Brien, "Intelligent Urban Data Monitoring for Smart Cities", Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part III, 2016,Springer International Publishing

[25] T. Yonezawa, I. Matranga, J. A. Galache, H. Maeomichi, L. Gurgen and T. Shibuya, "A citizen-centric approach towards global-scale smart city platform," 2015 International Conference on Recent Advances in Internet of Things (RIoT), Singapore, 2015, pp. 1-6.

[26] Delmastro, V. Arnaboldi and M. Conti, "People-centric computing and communications in smart cities," in IEEE Communications Magazine, vol. 54, no. 7, pp. 122-128, July 2016.

[27] Aisopos, Fotis, Antonios Litke, Magdalini Kardara, Konstantinos Tserpes, Pablo Martínez Campo, and Theodora Varvarigou. "Social Network services for innovative Smart Cities: the RADICAL platform approach." *Journal of Smart Cities* 2, no. 1 (2016).

[28] Kiran, Mariam, Peter Murphy, Inder Monga, Jon Dugan, and Sartaj Singh Baveja. "Lambda architecture for cost-effective batch and speed big data processing." In *Big Data (Big Data), 2015 IEEE International Conference on*, pp. 2785-2792. IEEE, 2015.

[29] Vögler, Michael, Johannes M. Schleicher, Christian Inzinger, and Schahram Dustdar. "Ahab: A cloud-based distributed big data analytics framework for the Internet of Things." *Software: Practice and Experience* (2016).

[30] COSMOS Middleware Flows repository, available at: https://github.com/COSMOSFP7/COSMOS-Platform-side

[31] COSMOS ILM flow, IBM Cloud Architecture Center - Real Time Traffic Analysis. [Online]. Available at: https://developer.ibm.com/architecture/gallery/transportationAnalytics

[32] ATOS Reference Architecture for distributed, scalable and "cloudified" complex event processing, available at: https://repository.atosresearch.eu/index.php/s/59vP3ATFl1JzyrJ

[33] Jeferry, Keith, George Kousiouris, Dimosthenis Kyriazis, Jörn Altmann, Augusto Ciuffoletti, Ilias Maglogiannis, Paolo Nesi, Bojan Suzic, and Zhiming Zhao. "Challenges emerging from future cloud application scenarios." Procedia Computer Science 68 (2015): 227-237.

[34] Athanasia Evangelinou, Michele Ciavotta, Danilo Ardagna, Aliki Kopaneli, George Kousiouris, Theodora Varvarigou, Enterprise applications cloud rightsizing through a joint benchmarking and optimization approach, Future Generation Computer Systems, Available online 11 November 2016, ISSN 0167-739X, http://dx.doi.org/10.1016/j.future.2016.11.002.

[35] Kousiouris, George, Andreas Menychtas, Dimosthenis Kyriazis, Spyridon Gogouvitis, and Theodora Varvarigou. "Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in Cloud platforms." *Future Generation Computer Systems* 32 (2014): 27-40.

[36] John, Vineet, and Xia Liu. "A Survey of Distributed Message Broker Queues." *arXiv preprint arXiv:1704.00411* (2017).

[37] Krawiec, Piotr, Maciej Sosnowski, Jordi Mongay Batalla, Constandinos X. Mavromoustakis, George Mastorakis, and Evangelos Pallis. "Survey on Technologies for Enabling Real-Time Communication in the Web of Things." In *Beyond the Internet of Things*, pp. 323-339. Springer International Publishing, 2017.

[38] Ngo, Minh Quan, Pari Delir Haghighi, and Frada Burstein. "A Crowd Monitoring Framework using Emotion Analysis of Social Media for Emergency Management in Mass Gatherings." *arXiv preprint arXiv:1606.00751* (2016).

[39] Andrews, Simon, Tony Day, Konstantinos Domdouzis, Laurence Hirsch, Raluca Lefticaru, and Constantinos Orphanides. "Analyzing Crowd-Sourced Information and Social Media for Crisis Management." In *Application of Social Media in Crisis Management*, pp. 77-96. Springer International Publishing, 2017.

[40] Cacho, Andrea, Mickael Figueredo, Arthur Cassio, Maria Valeria Araujo, Luiz Mendes, José Lucas, Hiarley Farias, Jazon Coelho, Nélio Cacho, and Carlos Prolo. "Social smart destination: a platform to analyze user generated content in smart tourism destinations." In *New Advances in Information Systems and Technologies*, pp. 817-826. Springer International Publishing, 2016.

[41] Croitoru, Arie, Andrew Crooks, Jacek Radzikowski, and Anthony Stefanidis. "Geovisualization of social media." *The International Encyclopedia of Geography* (2016).

[42] Ni, Ming, Qing He, and Jing Gao. "Forecasting the Subway Passenger Flow Under Event Occurrences With Social Media." *IEEE Transactions on Intelligent Transportation Systems* (2016).

[43] Li, Miaoyi, Zhenjiang Shen, and Xinhua Hao. "Revealing the relationship between spatio-temporal distribution of population and urban function with social media data." *GeoJournal* 81, no. 6 (2016): 919-935.

[44] Zhang, Zhenhua. "Fusing Social Media and Traditional Traffic Data for Advanced Traveler Information and Travel Behavior Analysis." PhD diss., State University of New York at Buffalo, 2017.