Towards Geo-Context Aware IoT Data Distribution

Jonathan Hasenburg and David Bermbach

TU Berlin & Einstein Center Digital Future
Mobile Cloud Computing Research Group, Berlin, Germany
{jh,db}@mcc.tu-berlin.de

Abstract. In the Internet of Things, the relevance of data often depends on the geographic context of data producers and consumers. Today's data distribution services, however, mostly focus on data content and not on geo-context, which would benefit many scenarios greatly. In this paper, we propose to use the geo-context information associated with devices to control data distribution. We define what geo-context dimensions exist and compare our definition with concepts from related work. By example, we discuss how geo-contexts enable new scenarios and evaluate how they also help to reduce unnecessary data distributions.

Keywords: Geo-Context, IoT, Data Distribution

1 Introduction

A long term vision of the Internet of Things (IoT) is to make sensor data available across applications and devices [13] to enable new and better services. For instance, exchanging information between cars, bikes, and other road users could improve road safety [15].

There are many data distribution services that are specifically tailored for IoT devices, e.g., AWS IoT[1] or Google Cloud IoT[2]. These services enable a selective distribution of messages as clients can define criteria [2,13] so that they only receive messages based on their respective interests. Compared to distributing data to all possible clients, this reduces bandwidth consumption and the amount of data processed by the clients which often operate in environments with constrained computational resources or bandwidth limitations.

Today's data distribution services, however, mostly focus on data content and not on the associated geo-context, which would benefit many IoT scenarios greatly. For example, a car that aims to avoid red traffic lights needs to process data from traffic lights within its current neighborhood only in order to determine an optimal route and velocity. Therefore, from the perspective of data consumers, it is often desirable not to receive data originating outside an area of interest

[1] https://aws.amazon.com/iot/
[2] https://cloud.google.com/solutions/iot/

to reduce computational efforts and cope with bandwidth limitations. A data producer, on the other hand, might already know that provided data is only relevant for data consumers in a certain area, and thus prevent others from receiving it. E.g., only drivers in the immediate vicinity of a particular car need to know when it brakes. Furthermore, if a data producer trusts the location provided by a data consumer, the geo-context can be used as an alternative to credentials-based authentication for data access control in some scenarios.

In the past, other researchers have been successful in using spatial data for various reasons (e.g., [5,6,7,10,11,12,14]). Every group of authors, however, has its own interpretation of the term geo-context and corresponding use cases. Thus, no standardized definition of geo-context exists, yet, and none of the related works consider the entire geo-context of data producers and consumers for their proposals. Therefore, we make the following contributions:

- We propose a definition of the geo-contexts associated with IoT devices.
- We compare our geo-context definition with concepts found in related work.
- We introduce three scenarios that benefit from using geo-context information and discuss how this reduces unnecessary data transmissions.

The remainder of this paper is structured as follows. We first present our motivation and three IoT scenarios that benefit from using geo-context information (section 2). We then discuss related work and present our definition for the geo-context of IoT devices (section 3). Next, we evaluate how using this additional information reduces unnecessary message transmissions (section 4). Finally, we draw a conclusion and present an outlook on future work (section 5).

2 Motivation

To better understand our motivation for using geo-contexts, we first need to highlight the difference between content and context. We do this by explaining these terms with the help of a topic-based pub/sub system. In such a system, publishers are the data producers and subscribers are the data consumers. Subscribers define which **content** they are interested in by subscribing to topics, e.g., when a subscriber creates a subscription to the topic *sensor/temperature*, he will receive temperature sensor measurements published to the same topic.

Dey defines **context** as "any information that can be used to characterize the situation of an entity" [8]. Thus, the context of IoT devices comprises many things such as other nearby devices, the type of power source, etc. In this paper, we only look at the geo-context which we consider to comprise (1) the location of the device and (2) special areas that are of interest/relevance to the device.

So why is it necessary to distinguish between content and geo-context? Both producers or consumers may have moved in between sending and receiving two data items. This, however, is not reflected in the content-related interests (e.g., the subscription) but affects the context-related interests. Hence, location information is not related to content.

Distinguishing content and geo-context information also has many practical benefits. For example, while it is possible to encode some geo-context information in topics, this requires clients to agree on such a structure and leads to very complicated and bloated topic trees. E.g., one could agree that the first topic level is always the country and the second topic level is always the city a given message refers to. Then, the topic *france/paris/sensor/temperature* would refer to all temperature sensors in Paris, while the topic *germany/berlin/sensor/temperature* would refer to all temperature sensors in Berlin. Besides the disadvantages mentioned above, this approach is very coarse-grained and it is not possible to distinguish between the location of a device and its area of interest.

We propose to consider the associated geo-context of IoT devices when distributing their messages for two reasons. First, with the geo-context additional information can be used to control data distribution. This can significantly reduce the amount of transmitted messages for scenarios where geo-context matters, thus reducing the load on data distribution services, the bandwidth consumption, and the amount of messages that need to be processed by clients. Especially in the IoT, such scenarios are quite common as IoT devices operate in a specific physical environment. Therefore, data collected by sensors such as temperature measurements or actions provided by actuators like moving a robotic arm are most relevant for other things in physical proximity. Such scenarios are the reason why Bellavista et. al [2] argue that geographical co-location should be taken into account. More domains with applications in which the value of information depends on the location of data producers and recipients include the Internet-of-Vehicle [10,20], Smart Cities [19] or Mobile Health [1].

Second, filtering data based on content and geo-context supports new (IoT) scenarios. In the following, we present three such scenarios from which two will also be used in our evaluation (section 4).

2.1 Scenario 1: Local Messaging and Information Sharing

In this scenario, clients travel on individual routes and send data to other clients in close proximity on a regular basis. Data can be of any kind, e.g., information concerning a client's current surroundings (e.g., the condition of the road), as well as simple text messages. Data should not be sent to clients too far away so that information is kept local; this prevents data from being mined by third parties. Furthermore, clients consume data based on their content interests, but also based on their individual geo-context. For example, a hiker might be interested in text messages (content) from clients in close proximity (geo-context), while a biker might be interested in road condition information (content) of a trapezoidal area in front of him (geo-context).

Other examples for the use of such a data distribution service are the real-time messaging service Jodel[3] or the location-based chats of Telegram[4].

[3] https://jodel.com

[4] https://telegram.org/blog/contacts-local-groups

2.2 Scenario 2: Open Environmental Data

Today's IoT sensor data is often not available directly to users, instead, it is common to create data dumps that are released once per day[5]. Such a procedure renders all IoT applications that require real time data impossible. Connecting the IoT sensors directly to a data distribution service, however, could easily lead to situations in which the service (and its potential clients) become overloaded, e.g., if a client accidentally consumes all data produced by temperature sensors at once by expressing interest for data labeled with *temperature*. While such a situation could be prevented by having the sensors use more diverse labels such as *temperature/regionA*, *temperature/regionB*, etc., considering the geo-context of data producers and consumers is more effective.

Data producers, for example, could restrict access based on arbitrarily shaped areas, e.g., only consumers in a certain geographic area can access data of said producer. On the other hand, data consumers often only have an interest in data of nearby sensors. For example, a tourist might want to receive weather data (content) only from the city he is visiting (geo-context) or a smart home application might only be interested in barometric pressure values (content) of sensors that are at most 20km away (geo-context) in order to identify approaching storms so that windows can be closed. Besides these more advanced application use cases, prohibiting the consumption of data from large areas at once can prevent accidental overload of services and clients.

2.3 Scenario 3: Context-based Data Distribution

Often, data needs to be distributed to clients in certain geographic areas. A prominent example for this is the Wireless Emergency Alerts system that is used to warn US citizens about dangerous weather or other critical situations[6]. The current system is not very accurate; only after November 30, 2019 it will reach an accuracy of below one-tenth of a mile overshoot [9] which is still rather imprecise.

A more accurate approach, however, in which messages are delivered based on the content interests of data consumers and the additional domain knowledge of data producers enables additional and better kinds of services. E.g., citizens interested in traffic information need to specify such a content interest only once and are then able to travel between districts (and even cities or states) while still receiving only relevant information as data producers know in which geographic area their messages are of relevance.

Many similar scenarios are possible in which data producers use their domain knowledge about the relevant geo-context to control the distribution of data, e.g., in the context of smart parking, advertisement, or smart buildings.

[5] E.g., this is done by the open data initiative of the German Meteorological Office: https://opendata.dwd.de/

[6] Wireless Emergency Alerts - https://www.fcc.gov/consumers/guides/wireless-emergency-alerts-wea

3 Geo-Context Dimensions

Previous work has already proposed to use geo-context information for a more advanced control of data distribution. Their focus, however, is not developing a general view on IoT device geo-contexts. Instead, the authors typically design a system for a very specific use case in which location-based data needs to be processed.

Chen et al. [7] propose a spatial middleware service that delivers messages to clients when they enter "zones" defined by data producers. While this allows data producers to control data distribution based on areas they consider as relevant, data consumers cannot control data distribution based on their own areas of interest.

Guo et al. [11,12] also propose a location-aware pub/sub service that delivers messages based on zones. In contrast to the service above, data consumers can control data distribution based on their areas of interest. The data producers, however, cannot use areas to control data distribution.

Frey and Roman [10] propose a protocol to bring context to a publish/subscribe system. They allow publishers to define a "context of relevance", and subscribers to define a "context of interest". When both contexts overlap, a message is delivered to the subscriber. While their context definition is very general, it can also be used for geo-context information, i.e., the (1) location of a device and (2) areas that are of interest/relevance to the device. However, they understand these two dimensions as one, so if a client moves he needs to update his subscriptions even if his area of interest did not change.

Li et al. [16] propose to use an R-tree index structure to efficiently identify which data producers are located in areas defined by data consumers. Again, this group of authors only looks at geo-context from one perspective so their approach does not work for areas defined by data producers and consumer locations.

Chapuis et al. [5,6] propose a horizontally scalable pub/sub architecture that supports matching based on a circular area around publishers and around subscribers. If the area of a publisher and subscriber overlap, messages are delivered. As these areas are not independent of client locations, this setup does not allow subscriptions to areas independently of the current location or subscriptions to multiple areas for different topics, e.g., as needed for the scenario in section 2.1.

Bryce et al. [3] propose MQTT-G, an extension of the MQTT protocol with Geolocation. While subscribers can define an area of interest to control message distribution, their area definitions are only created once per subscriber rather than for individual subscriptions. In addition, publishers cannot control the matching of messages based on areas defined by them.

Herle et al. [3] also propose to extend the MQTT protocol so that messages can be matched based on spatial geometries appended to published messages and subscriptions [14]. When both geometries overlap, messages are delivered. Their spatial matching, however, does not consider client locations.

Obviously, there is no general understanding of geo-contexts in IoT. Combining all these approaches allows us to identify four geo-context dimensions.

Both data producers and data consumers have a geographic location (**producer location** and **consumer location**), which consists of a latitude and a longitude value. Beyond this, data producers and data consumers have an area of interest, we propose to use geofences[7] to describe these areas. For our purposes, a geofence can have arbitrary shapes and may comprise non-adjacent subareas, e.g., Germany and Italy. The **consumer geofence** ensures that received data originates from an area of interest, i.e., producer locations are inside the consumer geofence. The **producer geofence**, on the other hand, ensures that only clients present in a certain area receive data, i.e., consumer locations are inside the producer geofence. Table 1 summarizes which of the four dimensions are considered by related work. Note, that Frey and Roman [10] only partly consider the location of consumers and producers as they mix it with the geofence.

	Location		Geofence	
Related Work	Consumer	Producer	Consumer	Producer
Bryce et al. [3]	✗	✓	✓	✗
Chapuis et al. [6]	✗	✗	✓	✓
Chen et al. [7]	✓	✗	✗	✓
Frey and Roman [10]	O	O	✓	✓
Guo et al. [12]	✗	✓	✓	✗
Herle et al. [14]	✗	✗	✓	✓
Li et al. [16]	✗	✓	✓	✗

Table 1. An Overview of the Geo-Context Dimensions in Related Work

As in the case of data content described above, producers and consumers can have multiple geo-contexts. For example, in a topic-based pub/sub system, a subscriber (consumer) can create individual subscriptions for different topics. Thus, when also using geo-context, subscribers might specify a geofence per subscription. Likewise, publishers might specify a geofence for every message.

Bringing geofences and locations together, two checks are necessary to decide whether data from a given producer should be sent to a given consumer (figure 1). First, from the consumer's perspective with the help of the consumer geofence and the producer location (Consumer GeoCheck) and, second, from the producer's perspective with the help of the producer geofence and the consumer location (Producer GeoCheck).

Figure 2 shows these two concepts by example. Here, a data consumer wants to receive all data from producers located in the northern part of a park (vertical stripes) by using the appropriate consumer geofence. For example, there could be a number of IoT sensors distributed across the park which collect and

[7] A Geofence is a virtual fences surrounding a defined geographical area. As a usage example, Reclus and Drouard describe a scenario in which such fences are used to notify factory workers about approaching trucks [18].

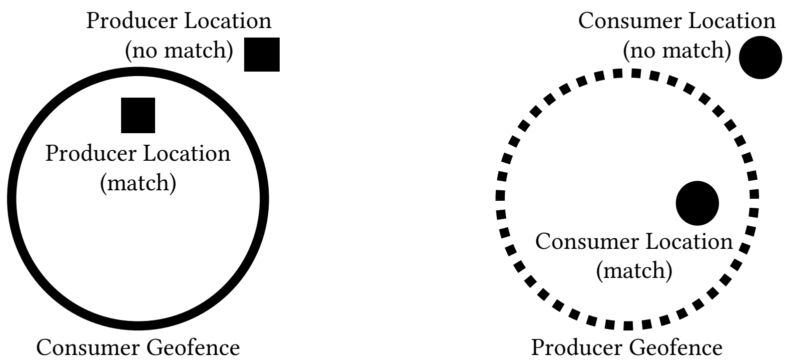

Fig. 1. Consumer GeoCheck (Left) and Producer GeoCheck (Right)

send information on humidity and other environmental parameters. Each data producer, however, wants to limit access to data consumers located inside an adjacent building (horizontal stripes), e.g., so that sensors do not accidentally expose information on botanical research experiments. Therefore, data producers use the appropriate producer geofence when transmitting data. The data should only be transmitted if the producer location is inside the consumer geofence (Consumer GeoCheck) and if the consumer location is inside the producer geofence (Producer GeoCheck). In the example, this is the case.

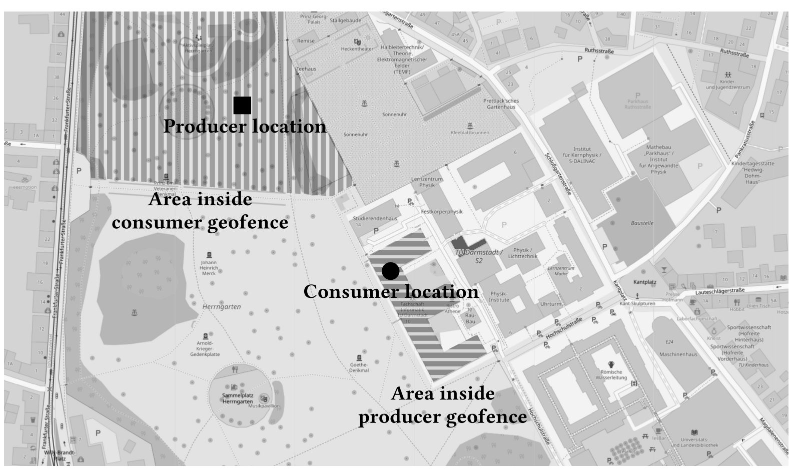

Fig. 2. All GeoChecks are Successful so the Data Consumer Receives Transmissions from the Data Producer.
Map data copyrighted by OpenStreetMap contributors and available from https://www.openstreetmap.org

4 Evaluation

In this section, we evaluate the impact of using geo-contexts to control data distribution. We describe a conceivable setup for two of the three scenarios that we introduced in section 2 and calculate the number of messages distributed to consumers when geo-contexts are used (GEO) and not used (NoGEO).

4.1 Local Messaging and Information Sharing

For this scenario, we assume a local messaging service for Central Park in New York City. The goal of this service is to provide a communication platform for visitors of the park while also preventing people outside the park from receiving messages. There is a multitude of different designs for such a service. We assume the following design:

- Clients can connect to the service without having to create an account, only their current location is required.
- Clients can act as data producers and send messages to the service (text, images, videos, etc.).
- Clients can act as data consumers and receive messages from the service.

For our analysis, we distinguish between the two approaches NoGEO and GEO. NoGEO does not consider the geo-context so messages are forwarded to all connected clients. GEO on the other hand, allows producers and consumer to specify geofences.

Central park spans an area of 3.4km^2 and had more than 42 million visitors in 2018 [4]. When assuming an even distribution of visitors across hours, there were about 5000 visitors per hour. Thus, we assume that our service is used by 5000 clients for one hour to demonstrate the effect of using geo-context information. All producer geofences span the whole park (as all visitors should receive messages). We assume, however, that consumer geofences only span 1% of the park each, as visitors are most interested in information that concerns their immediate environment (see figure 3).

For the evaluation, we assume that visitors send one message every two minutes on average. This leads to a total of 150k messages per hour. As subscription geofences only span 1% of the park, each visitor will on average receive only 1% of the messages with GEO, thus the total number of distributed messages is 7.5m/hour.
With NoGEO, every message is delivered to every visitor, so the total number of distributed messages is 750m/hour.

While this shows quite well how geo-contexts help to reduce the number of unnecessary message transmissions, it also shows how geo-context enable new application scenarios. Without geo-contexts, the application would not be useable as each visitor receives 2500 messages a minute, compared to 25 with geo-contexts. Furthermore, the requirement that only people inside the park are allowed to send messages can only be fulfilled with GEO (as long as no one

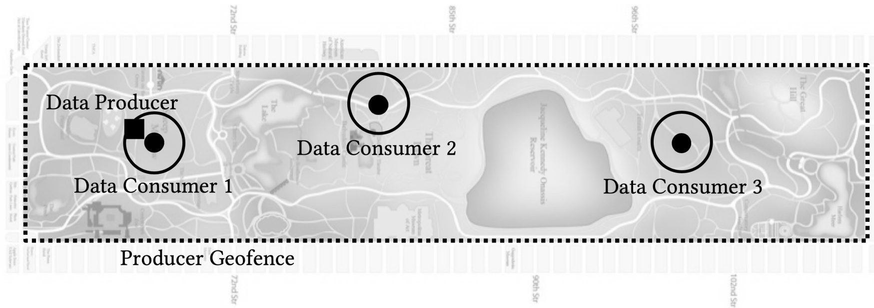

Fig. 3. The Producer Location is inside the Consumer Geofence of Data Consumer 1, so his Messages Get Delivered. All other Data Consumers do not Receive the Producer's Messages, even though their Locations are inside the Producer Geofence.
Central Park Map from https://biketourscentralpark.com/central-park-map

spoofs their location). With NoGEO, all users outside of the park would also receive every single message, and every message sent by that user would also be delivered to everyone else.

4.2 Context-based Data Distribution

For this scenario, we assume a context-based data distribution service for traffic information in the Netherlands. The goal of this service is to distribute measurement data from road side equipment to vehicles (data consumers) based on limitations put into place by the data producers.

In the Netherlands, more than 24k measurement sites exist which collect data every minute [17]. At the moment, the data is sent to a central database where it is processed before being distributed to data consumers. For this scenario, we propose a different architecture in which data is sent to cars directly so that they can make informed and individual decisions. This would also drastically reduce the latency which is 75 seconds with the current setup [17].

The Netherlands cover an area of about 42500km^2. For the GEO evaluation, we assume that each car uses a consumer geofence of the shape and size of the Netherlands. Producer geofences, however, only cover 1% of this area on average and can have very distinct shapes as they are based on the surrounding road network of each measurement site. This enables data producers to control data distribution very accurately and in real-time as geofences can be varied for different data transmissions, e.g., data on severe incidents needs to be send to cars further away while less important information is broadcasted only in the near vicinity.

With NoGEO, every car receives 24k measurements per minute as the data distribution is not limited. With GEO, every measurements will reach only 1% of the cars on average. Thus, every car only receives $24k * 0.01 = 240$ measurements a minute which greatly reduces resource consumption. Similarly, one can easily imagine that even smaller geofence sizes (both consumer and producer) can help to further reduce the number of messages.

5 Conclusion and Outlook

In this paper we proposed to use the geo-contexts associated with IoT devices to control data distribution. We showed that this can help to significantly reduce the amount of transmitted messages for scenarios where geo-context matters while also enabling new (IoT) scenarios that were not possible before. Our definition of geo-context comprises four dimensions: producer location, consumer location, producer geofence, and consumer geofence. We discussed which of these four have been considered by related work and explained why all dimensions are necessary with the help of three scenarios.

In future work, we plan to design a data distribution service based on the pub/sub paradigm that uses the geo-context of publishers and subscribers to control message distribution. For that, we want to use geo-contexts as an additional information input for the matching process which controls the distribution of published messages.

References

1. A serverless real-time data analytics platform for edge computing. IEEE Internet Computing **21**(4) (2017)
2. Bellavista, P., Corradi, A., Reale, A.: Quality of service in wide scale publish-subscribe systems. IEEE Communications Surveys & Tutorials **16**(3) (2014)
3. Bryce, R., Shaw, T., Srivastava, G.: MQTT-g: A publish/subscribe protocol with geolocation. In: 41st International Conference on Telecommunications and Signal Processing. IEEE (2018)
4. Central Park Conservancy Inc.: Central park conservancy annual report 2018 (Rev. 5). http://www.centralparknyc.org/about/annual-reports.html, accessed 09/08/2019 (2019)
5. Chapuis, B., Garbinato, B.: Scaling and load testing location-based publish and subscribe. In: IEEE 37th International Conference on Distributed Computing Systems. IEEE (2017)
6. Chapuis, B., Garbinato, B., Mourot, L.: A horizontally scalable and reliable architecture for location-based publish-subscribe. In: IEEE 36th Symposium on Reliable Distributed Systems. IEEE (2017)
7. Chen, X., Chen, Y., Rao, F.: An efficient spatial publish/subscribe system for intelligent location-based services. In: Proceedings of the 2nd International Workshop on Distributed Event-based Systems. ACM (2003)
8. Dey, A.K.: Understanding and using context. Personal Ubiquitous Computing **5**(1) (2001)
9. Federal Communications Commission: FCC improves wireless emergency alerts. https://www.fcc.gov/document/fcc-improves-wireless-emergency-alerts, accessed 09/08/2019 (2018)

10. Frey, D., Roman, G.C.: Context-aware publish subscribe in mobile ad hoc networks. In: Murphy, A.L., Vitek, J. (eds.) Coordination Models and Languages, vol. 4467, pp. 37–55. Springer (2007)
11. Guo, L., Chen, L., Zhang, D., Li, G., Tan, K.L., Bao, Z.: Elaps: An efficient location-aware pub/sub system. In: IEEE 31st International Conference on Data Engineering. IEEE (2015)
12. Guo, L., Zhang, D., Li, G., Tan, K.L., Bao, Z.: Location-aware pub/sub system: When continuous moving queries meet dynamic event streams. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. ACM (2015)
13. Happ, D., Karowski, N., Menzel, T., Handziski, V., Wolisz, A.: Meeting IoT platform requirements with open pub/sub solutions. Springer Annals of Telecommunications **72** (2017)
14. Herle, S., Becker, R., Blankenbach, J.: Bridging GeoMQTT and REST. In: Proceedings of the Geospatial Sensor Webs Conference (2016)
15. Khelil, A., Soldani, D.: On the suitability of device-to-device communications for road traffic safety. In: IEEE World Forum on Internet of Things. IEEE (2014)
16. Li, G., Wang, Y., Wang, T., Feng, J.: Location-aware publish/subscribe. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2013)
17. National Data Warehouse for Traffic Information: NDW real-time traffic data. https://www.ndw.nu/pagina/en/78/database/79/real-time_traffic_data/, accessed 09/08/2019 (2019)
18. Reclus, F., Drouard, K.: Geofencing for fleet & freight management. In: 9th International Conference on Intelligent Transport Systems Telecommunications. IEEE (2009)
19. Sanchez, L., Muñoz, L., Galache, J.A., Sotres, P., Santana, J.R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., et al.: Smartsantander: Iot experimentation over a smart city testbed. Computer Networks **61** (2014)
20. Sejin Shun, Sangjin Shin, Seungmin Seo, Sungkwang Eom, Jooik Jung, Kyong-Ho Lee: A pub/sub-based fog computing architecture for internet-of-vehicles. 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom) (2016)