TELKOMNIKA, Vol.16, No.6, December 2018, pp.2771~2781
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v16i6.11557

Towards Adaptive Sensor-cloud for Internet of Things

I Made Murwantara*[1], Hendra Tjahyadi[2], Pujianto Yugopuspito[3], Arnold Aribowo[4], Irene A. Lazarusli[5]
Faculty of Computer Science, Universitas Pelita Harapan Tangerang, Banten, Indonesia
*Corresponding author, e-mail: made.murwantara@uph.edu[1], hendra.tjahyadi@uph.edu[2], yugopuspito@uph.edu[3], arnold.aribowo@uph.edu[4], irene.lazarusli@uph.edu[5]

Abstract

The emerge of the Internet of Things (IoT) data as a commodity to optimize public services such as Fishing Locator has made sensor-cloud an important object. When sensors that are members of multiple IoT gateways can inter-operate at the same time for more than one application, it will reduce cost to deploy IoT infrastructure. However, reliability has also developed as the most important aspect for real-time data collection that should be streamed constantly. Due to uncertainty factors sensors failure is potentially occurred, then an adaptive approach should be addressed into this as to guarantee the flow of streaming data. This paper proposed an adaptive sensor-cloud mechanism to manage the reliability by using a runtime model approach where a transition model and dynamic software product line engineering will take place to weaving the system. Our technique is comparable to other approaches and can be implemented in many types of Cloud-based services.

Keywords*: sensor-cloud, virtual-sensor, adaptive, IoT, cloud computing, autonomous, dynamic software product line, models@runtime*

1. Introduction

Internet of Things (IoT) has been used and influenced almost all aspects of our lives. It can be found in a farming area [1], healthcare facility [2], education environment [3] and even in most modern vehicles [4]. The IoT gateway connects to some sensors to retrieve data, and then sending it over to a data collector who usually runs within a Cloud computing system. Each sensor provides information to make sure all operations or activities can be carried out to achieve its goals. For example, before we go to work we check the prediction of weather and traffic congestion which are based on many sensors reading, that located in many different places. In the last five years, the growth of IoT has increased drastically as the demand to predict businesses has driven a vast amount of data collection. According to Forrester [5], the design and operation scenarios which may include configuration of an IoT system in an easy way will create new demands. Other than that, the decision to capitalize the IoT data has already been agreed by the majority of European countries [5]. In a real-time data stream, we can make a prediction in a very tight time constraint and almost real situation [6]. This case rely on the stability of data stream to produce an accurate prediction, where several techniques are available and tested [7]. An interrupted data stream, for sure, will jeopardize a prediction which may lead to a fatal result. In a case of a failure traffic prediction, it may create a hectic and complicated congestion when drivers try to find alternative routes that utilize small roads.

Sensor-cloud provides data collection from many Things of IoT [8] that are retrieved from many different geolocations with varies reading [9]. Each data collector shared their sources that may have common and varies sensors. Those data, in common, share to many interests in a pay-as-you-go way where users may have flexibility to retrieve data from specific resources in terms of Sensor-as-a-Service (S2aaS) [10]. This kind of service enable some applications to have predictions without having an IoT infrastructure which, at the end, will reduce some costs. Most research to enhance the S2aaS services focus on scalability [11, 12], security [13], data caching [7], monetizing [14] the system and infrastructure [9] development. Xu et al. [11] argued that a sensor-cloud architecture should be in a form of open-end application that are scalable and energy efficient to cope the increase number of sensors. They proposed an Atlas Cloud-Edge-Beneath (CEB) that consists of four layers to bridging the data into an instance in a cloud computing system. Madria et al. [6] developed a middleware that

Received June 15, 2018; Revised October 3, 2018; Accepted November 2, 2018

configures physical sensor into virtual sensor as relationship between sensors, location and aggregate of location and sensors. Moreover, Madria et al. also coined a sharing data collection concept among different data consumers. In order to identifies important characteristic of sensors and data capture processes, Perera et al. [12] developed a framework that allows them to capture and model the sensor properties within a sensor-cloud. A combination of sensors based on its data type has been known as virtual-sensor. For example, a sensor temperature in a vessel is the same as a sensor temperature in the buoy are known as temperature virtual-sensor. In order to retrieve a virtual-sensor, end-user should configure of which sensors to retrieve in a pay-as-you-go S2aaS services. However, when a sensor that included into a configuration has failed to serve, the S2aaS should be able to provide a replacement source to avoid on breaking down the Service Level Agreement (SLA). In order to provide a reliable and a timely manner services, an adaptive technique that has a capability to find the virtual-sensors, seamlessly, replacement based on some parameters must be provided.

We address the virtual-sensor data collection using adaptive technique to guarantee the flow of data stream as stable as possible by reconfiguring the IoT infrastructure. In order to achieve our results, we make use modifications of models@runtime approach which originally designed for dynamic software product line [15, 16]. Our approach shows that the stability of data stream can be managed by using a scenario that reconfigured in an IoT infrastructure. In this paper, first, a sensor-cloud and adaptive approach will be explicated to make lights on the technology background. Then, an adaptive approach is proposed which is designed specifically for sensor-as-a-service. After that, we analyse the results of our adaptive approach. Finally, we make a conclusion at the end.

2. Research Method

In this chapter, we will provide the background of our research, motivation to do our work, overview on adaptive sensor-cloud services and how the result of this research can be monetized for future IoT product.

2.1. Background

Sensor-cloud is a collection of sharing IoT data which has connection to sensors in the field. Users may utilise the data in a pay-as-you-go without knowing the owners of the IoT devices. The ultimate benefit of sensor-cloud is the cost reduction of deploying thousands of sensors in a very short time and to maintain the reliability of output data. Sensor-cloud receives input, the sensor data, from IoT gateways with their data properties from sensor reading such as temperature, pressure, mechanical activities, and chemical where each sensor may have different characteristics and accuracy. A vast number of sensors may have commonality and variability in terms of data they have taken based on their location. As an example, an IoT gateway in location 1 collecting data from some sensors that measure outdoor temperature, air pollution and humidity. Meanwhile, IoT gateways in location 2 collecting data outdoor temperature, wind speed and humidity. If both gateways are located nearby, or let say less than 10 meter, then some of their data may always have not distinctive reading. However, if their location separated quite far, may be 10 kilometers away, then both similar data (from similar sensor, such as temperature) are distinctive.

As shown in Figure 1, end-user do not know where the data are coming from, but they know its property, such as blue colour indicated the temperature data and green colour for humidity data. The data are feeding by different sensors that can be identified based on the IoT gateway id. Moreover, the end-user able to create a configuration based on interest of the available data that are produced by some sensors. To manage a flexible system, models@runtime [15] create an adaptive environment by reconfiguring partially or the whole system which may include software and hardware which is based on the change of constraints. For example, a flooding sensor may have a communication technology management to efficient their battery with option to use bluetooth or Wi-Fi. Models@runtime method, usually, affiliated to dynamic software product line engineering that make use of reusability in an adaptive approach. We make use a modification of models@runtime to handle the adaptivity on an sensor-cloud system to advanced its flexibility based on the change of some parameters.

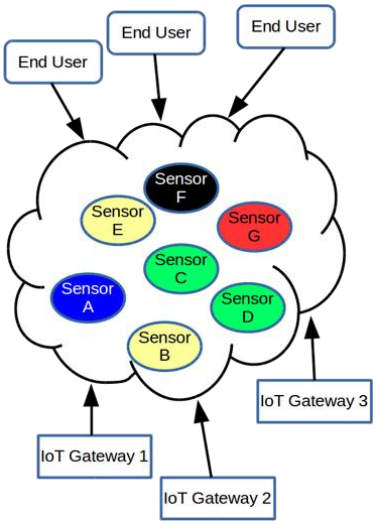

Figure 1. Sensor-cloud concept

2.2. Related Works

In the last decade, sensor-cloud has attracted many research groups. Xu et al. [11] focused on the scalability and energy efficient when numbers of sensor increase. Madria et al. [6] managed the relationship between physical and virtual sensors as a set of resources that has similar characteristics, which may reduce the complexity of sensor-cloud architecture. Both Xu [11] and Madria [6] only focused of how to integrate several elements of architecture to manage the data collection. While most research [7, 12, 13] pin point the method and technique to model the process of capturing data and secure the whole architecture. We found there is a gap in sensor-cloud research that is laid between architecture and services, namely the data availability. Our research makes use of adaptive approach to get insight into data availability that gain knowledge from a combination of data science and models@runtime.

2.3. Motivating Scenario

A fisherman wants to go fishing in an open sea. His fishing boat has specific catching based on fishing regulation that applied within certain country and location, which is designed to catch only tuna fish. Before off to the sea, he gathered some information regarding cost such as weather, water temperature, fuel price, bait location, the best fishing locations and market price. The information is coming into a fishing application that the fisherman use in his boat and mobile phone. This application also provides cost prediction regarding all the needs such as food and fuel supply during its fishing time.

Once he goes fishing, some input to his application has changed that pushed him to change the fishing vessel direction. For example, the sea water surface temperature has drop 10°C which changes the fishing location. If during his fishing hour, due to the failure of sensors that feeds the data, the application cannot get enough data so that he will waste a lot of money (for fuel, food and labour) to find the fishing points**.** This failure should be handled using a scenario that has a capability to take over the lost data by replacing the virtual sensor from the existing sensor-cloud configuration with some parameters such as sensor location and the cost to accesing the data. In our scenario, the fishing application should be able to get some data from different IoT providers, dynamically. It is possible that a single IoT service cannot provide a type of data, such as wind direction. In that case, the sensor-cloud should be able to provide alternative data from different IoT resources which is provided as a cluster geolocation data. Therefore, an adaptive sensor-cloud is used to have flexible data resources to avoid lack of application services such as cannot making a prediction of the fishing points.

We also think in advance about automatic steering motor vessel that manages its direction based on sensor-cloud guidance. The sensor-cloud includes geolocation which consists of longitude and latitude as a compulsory parameter on each IoT gateway. When an autonomous system, such as robot steering, installed in the fishing vessel attached to the

navigation system that has additional input from a sensor-cloud based application, then we can have an autonomous vessel steering system. This application will, absolutely, efficient the cost of fishing activity.

2.4. Adaptive Sensor-Cloud Services

There are three critical points to develop an adaptive sensor-cloud service. First, each sensor generates data with certain time line where the lost of data in certain sequence length of minutes may give impact to a prediction. Second, sensors with similar data type may be located in a very short distance (in almost the same geolocation) or far away which will become a stumbling block to provide an alternative data source when there is a failure during operation. The third, some sensors can or cannot be clustering into a certain group because of its uniqueness. A simple example is that the surface sea water temperature and under the sea temperature can change within 60 seconds in a spatial radius of 1 nautical mile. So that both sensors sea surface and deep sea temperature cannot be included into a group of data type because of their object of measurement. However, two sensors that measure the sea surface temperatures which have several miles of distance will be considered as similar type of sensor.

We model our sensor-cloud, as shown in Figure 2, where each sensor should have their geographical location (GeoLocation) identified. Our approach uses geolocation as the key point to group sensor based on their data type where IoT gateway feeds the virtual sensor. The reliability manager as a controller manages of which configuration should be implemented in a Sensor-Cloud system. A Feature Model (FM) provides the architecture of our adaptive sensor-cloud as features. As depicted in Figure 3, the adaptive IoT feature model consists of feature sensor with sensor types in it, Geolocation to identify the distance of sensor locations that are grouped into regional, island and country. To manage the data flow, we split the size of the sensor's data based on the 10 kilo-bytes into 100 mega-bytes which depends on the location that end-users prefer to have it. A method of data flow is also an option whether to use real-time or batch data retrieval, where both option can be included into our sensor-cloud configuration. Moreover, format of data to send has option to have json, csv or txt format. And, in reliablity, security and availability can be selected in partial or both.

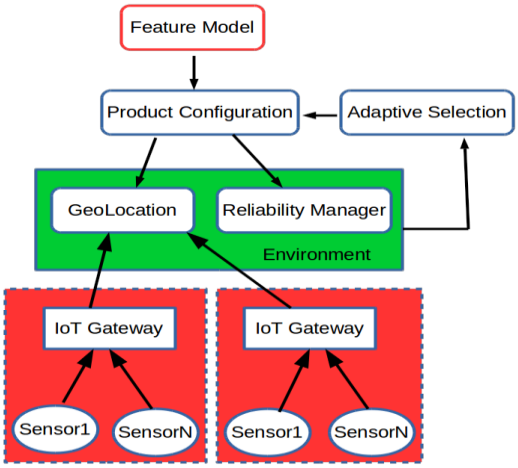

Figure 2. Adaptive sensor-cloud architecture

A relation between sensor-cloud and control system is described in FM, as shown in Figure 3, as robot feature has child actuator that has option of steering and throttle. An example is that when the information of fishing location will generate guidance to directing the vessel into a specific location that represented as latitude and longitude of geolocation. In order to have fuel efficient an automatic steering is needed. We modeled this automatic control as a feature robot that has an ability to perform an autonomous vessel steering system where other information from virtual-sensor might change the fishing location.

In our approach, we make use of dynamic software product line to provide an adaptive sensor-cloud inspired by Murwantara et al. [16]. In order to manage the transition between

Product Configuration, we implement the transition model of the model. A transition model comprises of product configurations which has combination of features and transition states with their rules as parameters to make transitions between product configurations. In Figure 4, we have geolocation as distance between a type of sensor and reliability as the status of sensor when it has a stable data flow. The transition illustration is as follow:

1) We assume the configuration as current running configuration.
2) An event occurs that one sensor in a particular location has failed to send its data which reduce the accuracy of prediction of several applications that depends on it.
3) The adaptive selection, as shown in Figure 2, reconfigures the sensor-cloud services - as a product configuration - by finding an IoT gateway nearby that has similar sensory data.
4) During searching sources that has alike sensor data type, the distance between the failed one and the available is measured.
5) When the distance is match to the constraint as defined in the feature model, see Figure 3 the feature Distance, then configured using the available one.

However, when a failed sensor has return to service, the transition will be reconfigured to have the original sensor location.

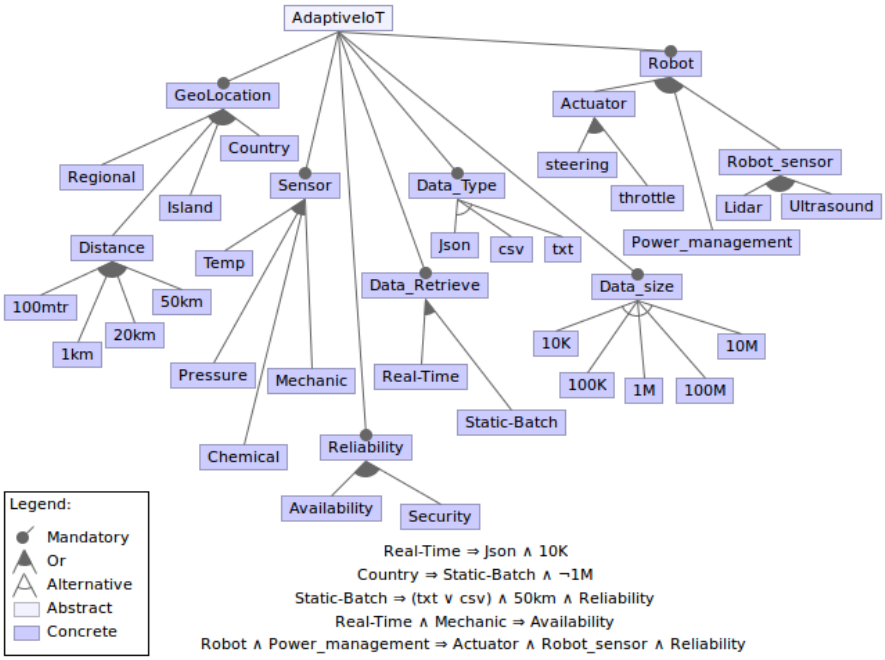

Figure 3. Excerpt feature model of sensor-cloud

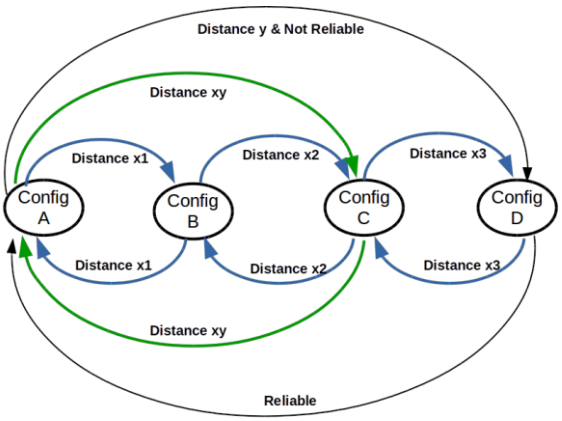

Figure 4. Transition diagram of sensor-cloud configuration

An application that heavily depends on the sensors' location, such as local temperature, might create an inaccurate prediction that at the end will influence the cost of end-user businesses. For example, a farming application uses temperature information from many sensors surrounding end-user location to have prediction when the best time to spread the fertilizer. If there is a wrong prediction, it may cost a lot to the farmer since it may influence their crops into lower harvest. In our adaptive sensor-cloud, end-user must configure which group of sensors they want to implement into their application. As shown in Figure 5, a class diagram of our sensor-cloud, class tactic manages which sensors will be included into client application by configuring the available sensors. Class tactic reconfigures the sensor-cloud by referring the information of the client configuration. The available sensor lists are gathered from the class IoT gateway which also provides information of their geolocation. So that, when an end-user configures their virtual sensors, the price of data are also presented to provide cost of operation planning. In order to manage of which sensor available for certain location, IoT gateway provides list of sensors and their status, whether it is on service or not. Then the collector collects the data that feeds into virtual sensor.

Class tactic, as presented in class diagram Figure 5, is the main configuration of virtual-sensor for specific end-user. In class Tactic, a fail-over mechanism is an option which will create an adaptive mechanism to configure combination of virtual-sensor. It is worth noting that a Virtual-sensor might have its input from many sensor-nodes if configured as an adaptive tactic. Otherwise, the virtual-sensor only linked to a specific sensor-node. In our model, end-user may configure the sensors to get the data as a group of virtual sensor. An adaptive management configures by tactic where sensors are grouped within a configuration. Then, we group similar sensors in a distance of geolocation, see Figure 3, and add the profile to have a fail-over handling. After that, a reconfiguration should take place based on which tactic ID has been chosen by the end-user. However, if the tactic failed to have alternative solutions based on user configuration, then tactic will stay running in their last configuration. In this paper we do not provide an intelligent reconfiguration such as an adaptive fail-over management using machine learning. We keep this problem for our next research. sensor-cloud may collect a vast amount of data. However, the storage capacity has its price. In order to cope with such condition, class collector will archiving the data that chunk into time lines, latitude and longitude of specific IoT gateway. As shown in Figure 5, collector collects the data in the cloud storage that can be pushed into virtual sensor when users need it.

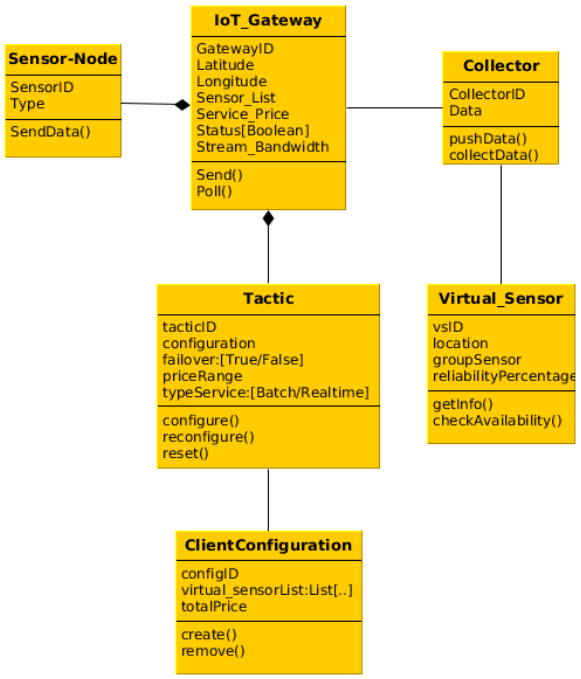

Figure 5. Class diagram of our adaptive sensor-cloud

2.5. Monetizing

We monetizing the sensor-cloud services based on the geolocation of IoT gateway. In our opinion, the more distinctive sensor data in a very difficult location may have its price high and low price for a common sensor data in a high-density sensor location. For example, the price of sensor data of a sea stream flow in arctic will be higher than the stream flow of a river in a big city in Europe. We include this information in the class of IoT gateway.

Types of data retrieval is also influence the price. In a real-time data retrieval, sensor-cloud should provide a high reliable services which may include low latency of traffic and high speed of data buffer to overcome the lack of data transmission to the end-users. On the other hand, batch data is an archive storage that uses less reliability should have lower price than real-time. Another monetizing in our sensor-cloud is the retrieval of data. Applications that rely on our system may feed data very intensively. For example, an application that predicts some businesses such as traffic in a megapolitan city, such as London, will feeding a lot amount of data each second. Since the most busiest road happens during working days, then the system will charge adaptively by metering the amount of data that has been transferred every hour.

2.6. Sensor-Cloud Semantic

In providing a reliable source management, we extend [17, 18, 19] IoT semantic to have geolocation and fail-over features. IoT gateway sends multiple sensors data into data collector, as depicted in Figure 2, where each transaction reveals its sources based on id and geolocation. Each IoT gateway pushes their data into a specific data collector predefined on the registration of the things. In our work, we assume that in one coordinate geolocation only an IoT gateway exist. It is worth to note that some location may have different altitude but has similar geolocation. For example, a building may have several IoT gateways in a stack able location.

The configuration of IoT gateway is represented in an XML file, as shown in Listing 1. The information of IoT gateway represent the geolocation with specific latitude, longitude and altitude. In IoT gateway XML, price is the amount of money to retrieve data that is included into the virtual sensor configuration. Each sensor in an IoT Gateway should have its name describes the specific devices attached to it. This sensor name will be categorized by the sensor-cloud as group of virtual sensors.

Listing 1. An Excerpt of Sensor-Cloud XML

```
<? xml version="1.0" encoding="UTF-8"?>
  <iotgtw>
     <geolocation>
          <id> 101 </id>
          <latitude> -6.12 </latitude>
          <longitude> 106.77 </longitude>
          <altitude> 1 </altitude>
          <price> $0.05 </price>
       </geolocation>
       <device>
          <sensor>
               <Type> Temperature </Type>
               <Property>
                   <P> getTemp </P>
               </Property>
          </sensor>
          <sensor>
            <Type>Humidity</Type>
               <Property>
                 <P> getHumid</P>
               </Property>
          </sensor>
        <sensor>
      <Type>WindDirection</Type>
               <Property>
                  <P> getWD</P>
               </Property>
        </sensor>
     </device>
  </iotgtw>
```

3. Results and Analysis

This section presents a running example that conveys us to an implementation of our method into a realistic product. In this section, we also explain of how our example may help us to manage an adaptive mechanism for sensor-cloud.

3.1. A Running Example

An application, we call it CSFishFinder, that runs in a mobile and desktop system provides a guidance for fishing vessel by tracking the history and current data to make a prediction of tuna fish position. This application retrieves its information from sensor-cloud [20, 21, 22] that pushed its data using an Application Programming Interface (API). CSFishFinder uses data from several sources such as VMS, globalfishingwatch, NOAA dan

some private data (including buoy). These data are analysed using an ensemble learning of machine learning to produce a prediction in a range of month or week. Moreover, in order to get a better accuracy, CSFishFinder also provides a real-time fish finder service that relies not only data from some static sensor but also from some mobile sources such as vessel and satellite. As shown in Figure 6, CSFinder uses many sources that are feed into a sensor-cloud services, such as vessel, weather station and Internet of Robotic Things (IoRT) [23, 24. 25]. Aforetime off to shore to catch tuna fish, the fisherman check the location that has the highest possibility of tuna fish. After several hours, a problem exist from one of the data feeding that the satellite connection has intermittent connection to the sensor-cloud. This problem, for sure, will influence the prediction results and might increase fishing operation costs.

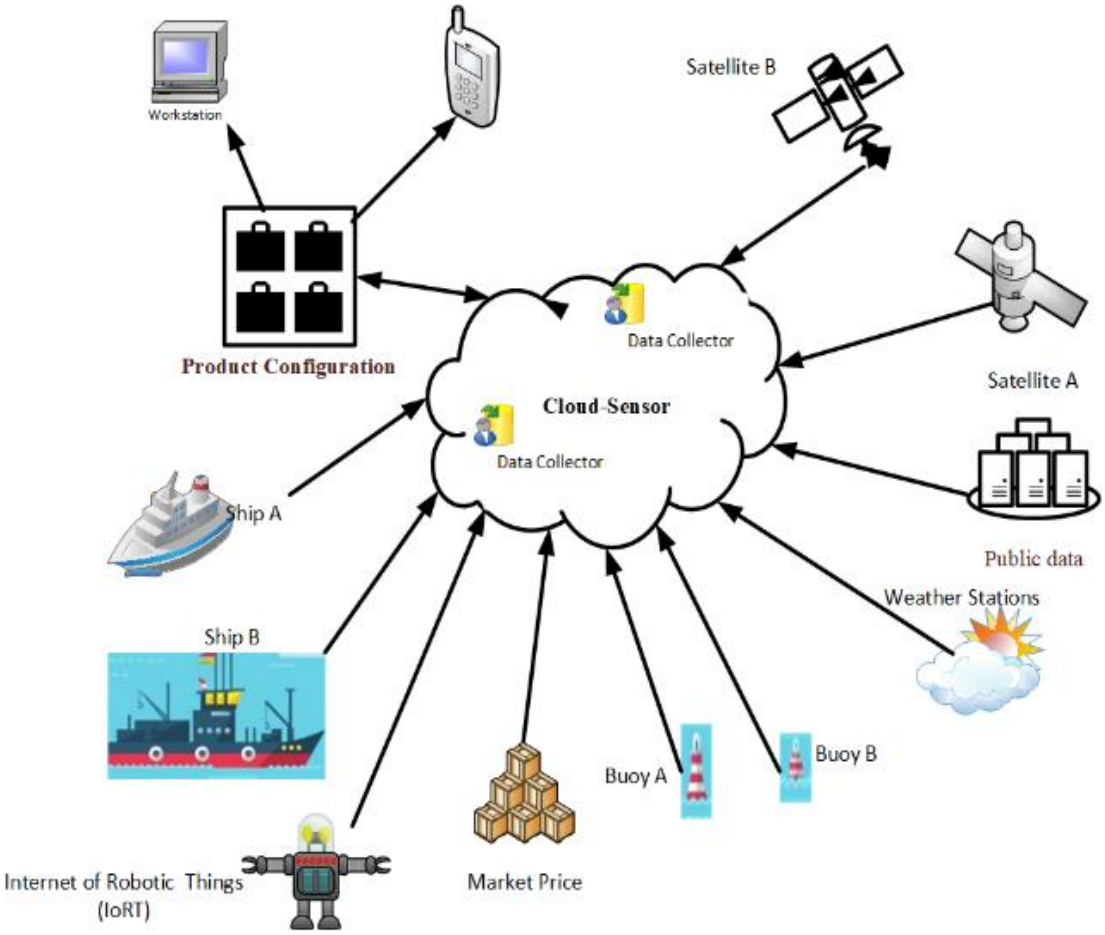

Figure 6. Illustration a running example of CSFishFinder

Sensor-Cloud provides feeding data into applications by providing some data as follows, temperature of sea water and in site location, wind direction, weather condition (rain, clear, mist), moisture and sea current direction as supporting data. To predict the tuna fish position, the history and current.

3.2. Go Fishing

A fisherman has checked the fishing location before he goes fishing. However, during his journey to the location, the information of the sea water temperature that feeds by some buoys surround the fishing location has been lost. This condition may give a bad prediction of the fishing location.

In this case, the sensor-cloud must provide a replacement to manage the reliability of data. Since buoy is the most reliable data of the sea temperature, it is very hard to replace such

accurate information. in order to cope with this condition, the adaptive mechanism in the sensor-cloud search for the nearest location which has similar data within range of location. Based on transition model that shown in Figure 4, the distance between primary sensor is ranged into several areas. For example, a range of 1, 10, and 100 meters. If there is a virtual sensor data type that match within the nearest location, the Cloud- sensor will change its source into that source automatically, as depicted in Figure 7.

Despite some replacement may overcome the loss of original data sources, the application might have inaccurate prediction because of the differences of data reading, which will, most likely, influence the prediction results. In order to address this problem, an analytic service should be able to create an imbalance learning for analytic services which is beyond the discussion within this paper.

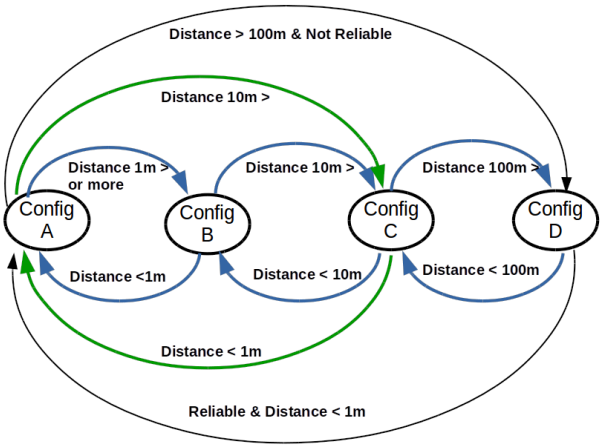

Figure 7. Adaptive virtual sensor configuration based on geolocation.

3.3. Load-Balancing

We use a load-balancing system to manage fail-over. We reversed the service direction of a normal load-balancing system to serve the Internet of Things, as shown in Figure 8. The sensor load-balancer receives input from IoT gateways that already registered into the sensor-cloud. Using security credential, each data flows to the system and are grouped into virtual sensor types such as sensor temperature, humidity and wind direction. If end-users would like to receive a specific type of file to stream, then the data type stream will format the data into it. The geolocation group provides reference of sensor location. This information creates an adaptive of sensor data stream, which autonomously change the source of data to similar data with nearby sources using the transition mechanism as illustrated in Figure 8.

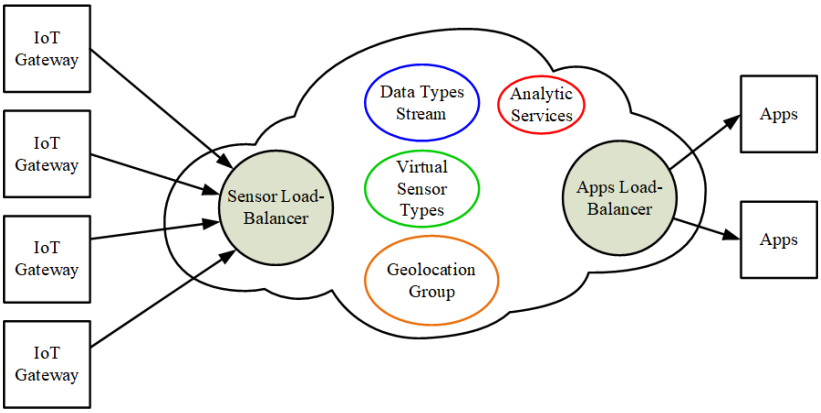

Figure 8. The Sensor-cloud fail-over management.

In our fail-over management, the sensor load-balancer receives input data from IoT gateways that have tags on their data stream that informs of their coordinate location. Then some treatment is done by grouping to the geolocation information and clustering into their data, as stream or batch, and virtual-sensor types. Another management entity is the analytic service that provides some analytic operations which include calculating the probability of IoT gateway reliability, making prediction of the incoming and outgoing sensor load-balancer and apps load-balancer, making decision on which scenario of fail-over should be addressed on specific symptoms or based on user pre-defined fail-over management, and the concurrent request management.

4. Conclusion

We have presented an adaptive sensor-cloud model. Our work modifies the dynamic software product line engineering to create adaptive services for the Internet of Things. We make use of models@runtime to enable adaptive mechanism by dynamically reconfigure the virtual sensor, which has its price on data availability and sensors' uniqueness. We have proposed a model to monetizing the sensor-cloud by classifying types of sensor, the amount of data transferred by time line and geolocation, which may attract the IoT data owner to share their data into this kind of system. Feature Modelling is used to provide the method of combination of components that related to the transition diagram.

In our future work, we would like to measure the performance of our adaptive sensor-cloud model in an integrated system such a fish finder system. Several machine learning and artificial intelligence approaches will be included to identify and predict predefined types of fish in different locations.

Acknowledgment

This work is supported by Research Grant No. 021/KM/PNT/2018 of the Ministry of Research, Technology and Higher Education of the Republic of Indonesia, managed by LPPM of Universitas Pelita Harapan, Research Contract No. 148/LPPM-UPH/IV/2018.

References

[1] J. Stewart, R. Stewart, and S. Kennedy. *Dynamic iot management system using k-means machine learning for precision agriculture applications*. ICC 2017. New York, NY, USA: ACM. 2017; 142:1– 142:8.

[2] R. Reda, F. Piccinini, and A. Carbonaro. *Towards consistent data representation in the IoT healthcare landscape*. Proceedings of the 2018 International Conference on Digital Health, ser. DH '18. New York, NY, USA: ACM. 2018: 5–10.

[3] D. Amaxilatis, I. Chatzigiannakis, G. Mylonas, L. Pocero, D. Zarras, and A. Koskeris. *Green mindset: Using IoT to promote energy efficiency and sustainability in Greek public schools*. Proceedings of the 19th Panhellenic Conference on Informatics, ser. PCI '15. New York, NY, USA: ACM. 2015: 297–302.

[4] A. Jime´nez, V. Garćıa Díaz, and J. Anzola. *Design of a system for vehicle traffic estimation for applications on IoT*. Proceedings of the 4th Multidisciplinary International Social Networks Conference, ser. MISNC '17. New York, NY, USA: ACM. 2017.

[5] "Predictions 2018: Iot will move from experimentation to business scale," https://go.forrester.com/blogs/predictions-2018-IoT-will-move-from-experimentation-to-business scale/, accessed: 2018-05-25.

[6] S. Madria, V. Kumar, and R. Dalvi. Sensor cloud: A cloud of virtual sensors. *IEEE Software*. 2014; 31(2): 70–77.

[7] S. Chatterjee and S. Misra. Adaptive data caching for provisioning sensors-as-a-service. *IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. *2016:* 1–5.

[8] A. Alamri, W.S.Ansari, M.M.Hassan, M.S.Hossain, A.Alelaiwi and M. A. Hossain. A survey on sensor-cloud: Architecture, applications, and approaches. *International Journal of Distributed Sensor Networks*. 2013; 9(2):917-923.

[9] Y.-H. Liu, K.-L. Ong, and A. Goscinski. Sensor-cloud computing: Novel applications and research problems in *Networked Digital Technologies*, R. Benlamri, Ed. Berlinn and Heidelberg: Springer Berlin Heidel- berg. 2012: 475–486.

[10] QQ. Xie and L. Wang. Privacy-preserving location-based service scheme for mobile sensing data. *Sensors*. 2016; 16(12).

[11] Xu and A. Helal. Scalable cloud x2013 “Sensor architecture for the internet of things. *IEEE Internet of Things Journal*. 2016; 3(3).
[12] C. Perera, A. Zaslavsky, C. H. Liu, M. Compton, P. Christen, and D. Georgakopoulos. Sensor search techniques for sensing as a service architecture for the internet of things. *IEEE Sensors Journal*. 2014; 14(2): 406–420.
[13] Poolsappasit, V.Kumar, S.Madria, and S.Chellappan. Challenges insecure sensor-cloud computing. in Secure Data Management, Springer Berlin Heidelberg. 2011.
[14] C. Zhu, V. C. M. Leung, E. C. H. Ngai, L. T. Yang, L. Shu, and X. Li. Pricing models for sensor-cloud. *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*. 2015: 454–457.
[15] G. Blair, N. Bencomo, and R. B. France. Models@ run.time*. Computer*. 2009; 42(10): 22–27.
[16] I. M. Murwantara, B. Bordbar, and J. B. F. Filho. A self-adaptive architecture with energy management in virtualized environments. *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT)*. 2017: 124–130.
[17] S. Ahmad, L. Hang, and D. H. Kim. Design and implementation of cloud-centric configuration repository for DIY iot applications. *Sensors*. 2018; 18(2): 474. DOI:10.3390/s18020474.
[18] T. Kushida and M. Yuriyama. *Sensor-Cloud Infrastructure - Physical Sensor Management with Virtualized Sensors on Cloud Computing*. 13th NBIS. Gifu Japan. 2010: 1-8.
[19] T. Kushida, M. Itakura and M. Yuriyama. A New Model of Accelerating Service Innovation with Sensor-Cloud Infrastructure. 2011 Annual, CA, 2011: 308-314. DOI:10.1109/SRII.2011.42
[20] S. Misra, S. Chatterjee and M. S. Obaidat. On Theoretical Modeling of Sensor Cloud: A Paradigm Shift from Wireless Sensor Network. *IEEE Systems Journal*. 2017; 11(2).
[21] S. M. A. Oteafy and H. S. Hassanein. Resilient IoT Architectures over Dynamic Sensor Networks with Adaptive Components. *IEEE Internet of Things Journal*. 2017; 4(2).
[22] N. Alhakbani, H. M. Mehedi, M. H. Anwar and A. Mohammed. A Framework of Adaptive Interaction Support in Cloud-Based Internet of Things (IoT) Environment. IDCS 2014: 136-146, doi: 10.1007/978-3-319-11692-1_12
[23] G. Keramidas, N. Voros, and M. Hbner. Components and Services for IoT Platforms: Paving the Way for IoT Standards (1st ed.). *Springer*. 2016.
[24] P. P. Ray, "Internet of Robotic Things: Concept, Technologies, and Challenges. *IEEE Access*. 2016; 4: 9489-9500. DOI: 10.1109/ACCESS.2017.2647747.
[25] P. Simoens et al. *Internet of Robotic Things: Context-Aware and Personalized Interventions of Assistive Social Robots*. 5th IEEE CLOUDNET. Pisa, Italy, 2017: 204-207.