A Cost Model for Data Discovery in Large-Scale IoT Networks of Smart Cities

Torbjørn Kirkevik Soltvedt[§], Amir Sinaeepourfard*, and Dirk Ahlers[¥]

* Department of Computer Science - [¥] Department of Architecture and Planning
*[§] Norwegian University of Science and Technology (NTNU)
*[§¥] Trondheim, Norway
*[¥]{a.sinaee, dirk.ahlers}@ntnu.no and [§]torbjoks@stud.ntnu.no

***Abstract*— A smart city with huge numbers of physical (e.g., sensors and actuators) and non-physical (e.g., external databases) data sources will continuously produce high amounts of massive city-data. Distributed data storage across the city may store the produced city-data. City managers through different update mechanisms may send the produced city-data from distributed data storage to centralized data storage (e.g., Cloud data storage). Hence, the data discovery issues are in a vital position in the smart city concepts because the produced city-data may exist in different data storage platforms from distributed to centralized data. In this paper, we will first present our proposed Distributed-to-Centralized Information and Communications Technology (D2C-ICT) architecture for the Zero Emission Neighborhoods (ZEN) center. This proposed D2C-ICT architecture can provide multiple facilities from the joined benefits of distributed and centralized technologies in smart cities. Second, we will show how the Multi-Attribute Utility Theory (MAUT) cost model can be beneficial to find the appropriate data for building city services across the different storage platforms on the city scale as well as can be applied in the ZEN center and its pilots.**

Keywords—Smart City; Cost Model; Data Discovery; Distributed-to-Centralized ICT Architecture

I. INTRODUCTION AND MOTIVATION

Data are fundamental feed components for services in smart cities and perform an essential role in the smart cities' improvement. Smart cities produce numerous city data from physical and non-physical data sources. Selecting among data is a crucial and complicated issue in this context, as city services are often dependent on appropriate data. To be able to choose relevant data for city services, there are many different data discovery solutions to estimate the performance measurement of the produced city-data and related their ICT system in smart cities. The performance measurement solution will be more complicated: i) once data can exist in the different data storage platforms from smallest to large scale of the city; ii) sometimes having the same data stored in two or more separate data storage platforms.

According to the complexities, as mentioned above, several cost models can be defined to measure the performance and efficiency of the produced city-data and related their ICT system through different ICT Key Performance Indicators (ICT-KPIs). The cost model may help to find an appropriate date in the large-scale ICT networks of smart cities across different local data storage from neighborhoods to city and Cloud data storage platform.

II. RELATED WORK

Several proposals about designing ICT architecture for smart cities exist, including Centralized [1], DC2C-ICT [2-5], and D2C-ICT [6] architecture. More details about ICT architectures are described in [34]. However, there are still open challenges through designing an efficient D2C-ICT architecture for smart cities through the different city and ICT requirements such as data management [1, 7, 33] and software services management [7].

With the focus on accessing the city-data, the Cloud storage platform as a centralized platform may provide facilities to store and access all city-data in the centralized platform [1]. However, D2C-ICT architecture provides benefits to store and access the produced city-data in the local data storage across the city [8]. If in case it is necessary concerning the data privacy issues and business requirements, all or some of the city-data will move from the local data storage in the city to the Cloud storage platform in most of the cases out of the city.

Due to data may store in different data storage platform from distributed to centralized ICT networks of smart cities, there is a need to define the performance measurement techniques. These techniques may be useful for data discovery in large-scale ICT networks of smart cities from distributed to centralized storage platform. Recently, several techniques are suggested, including quality measurement [9, 10], KPIs measurement [11-13], and cost models (cost estimation models) [14, 15]. In centralized ICT architecture, all performance measurement techniques acquire in the Cloud computing environment [1, 11, 12]. In the D2C-ICT architecture, there is a need to define how the performance measurement techniques can be applied across distributed to centralized ICT networks of smart cities. Recently, a set of KPIs as defined in [13] through Fog-to-Cloud ICT architecture. These KPIs can measure the most appropriate resources for data and service management.

Considering the reasons above, we use a MAUT cost model for data discovery through our proposed D2C-ICT architecture for smart cities [6, 7, 16].

III. PERFORMANCE MEASUREMENT MODELS FOR DATA DISCOVERY IN SMART CITIES

Quality measurement, ICT KPIs measurement, and cost (cost estimation) models are some examples of these data discovery solutions in the smart cities as described below.

A. Quality Measurement

Quality is described as the result of the judgment of the perceived composition of an entity concerning its desired composition [17]; in other words, how close it is to the ideal instance of it; and as fitness for use, with specific consideration for data integration tasks [35]. Data quality definition is still non-agreement on a single description. However, mainly data quality challenges imposed by mistaken data entry, missing information, or other invalid data [18]. If data value reaches an appropriate level, it will be stored in your storage platform. Oppositely, the data must be discarded or changed [19].

In Big Data environments perspective, the different quality level of collected data exists. Distinct types of data analysis techniques may process the collected data. Also, various applications/services may request data with different level qualities. Therefore, the paper [20] offered some data preprocessing techniques to improve data quality level in big data systems, as shown below:

- Integration: Data integration is a kind of combined technique for data residing in different sources. Traditional data integration approaches are used in a traditional database, as shown in two main categories as follows:
  - Data Warehouse (also known as ETL): This category has three main steps, as shown below:
    - The extraction step connects to the sources for choosing and collecting the relevant data for further analysis processing.
    - The transformation step aims at converting the obtained data into a united format through some designed rules of application.
    - The loading step receives the obtained data first. Then the obtained data will be sent to the storage platform for temporary or permanent storage.
  - Data Federation: This method provides a virtual database to make a query and aggregate data from different sources. In addition, the virtual database has a container of information or metadata, which includes the actual data and its location.
- Cleansing: The data cleansing techniques aim to recognize inaccurate, incomplete, or unreasonable data through a specific process. This process leads to the improvement of the quality level of data by discarding or changing inappropriate data. The cleansing techniques have five main steps, as shown below:
  - To describe and identify the type of errors;
  - To explore and distinguish examples of error;
  - To correct errors;
  - To register the type of errors;
  - To correct data entry procedures to minimize errors.
- Redundancy elimination: The data redundancy elimination techniques aim to remove redundant datasets. The redundant data makes an unpleasant effect on data transmission, data storage, data processing, etc.

IoT devices may generate data with different formats in smart cities. These data have a different level of quality. There are several views to measure data quality in smart cities. In one's view, applications and services are responsible for measuring data quality in the smart city [21]. There are two main views to measure the quality of applications, services, and products, including Quality of Service (QoS) and Quality of Experience (QoE), as shown below:

- The QoS refers to the characteristics of a service. QoS may satisfy the stated and implied needs of the use of the service [22]. These characteristics are quantitative measurements of the performance of the service.
- The QoE is based on the experience of the users and how they appraise the quality, e.g., user's feedback [23, 24].

B. ICT KPIs

Performance indicators or KPIs are a type of performance measurement. In general, KPIs evaluate the performance of an organization or a particular activity in which it engages [25]. In the smart city, if services are one of the main products of smart cities, the performance of the services can be measured through a set of KPIs in smart cities. Therefore, a characteristic that is deemed essential for the service and data will be referred to as KPIs. A KPI is a quantifiable metric that shows the performance of the service or data.

There are two main categories to measure KPIs through centralized and distributed-to-centralized technology in smart cities. Those KPIs are Cloud computing KPIs and distributed-to-centralized KPIs as discussed below:

1) Cloud Computing offers an extended context for service-oriented business and ICT [11]. Different KPIs [24, 26] are defined to measure success in Cloud computing environments [12]. Those KPIs [26] are based on important performance measurements and aspects of consideration when selecting a Cloud computing provider service, as well as definitions on how to measure them.

2) Edge Computing expands the Cloud computing facilities to the edge of the network, near to end-users, bringing the low level of latency and a large amount of bandwidth [27]. Different distributed technologies are used in Edge computing to extend Cloud technologies capabilities at the edge of the network. Edge computing aims to bring computation offloading toward the edge of networks [27]. Different KPIs [26] can be defined in the smart city literature for measuring the performance of the produced data to run the applications on them.

C. Cost Models

Cost estimation models used to estimate the costs of a product or project in general [26, 28]. The results of the models have approved the usage of a product or project. In addition, the results are also factored into business plans, budgets, and other financial planning and tracking mechanisms. There are several mathematical algorithms or parametric equations to estimate these costs [26].

In [26], we review ten different cost models. We realized that each cost models and selection methods have their strengths and weaknesses, and the best one depends on these properties and the users' demands. The properties that are used to compare different solutions are computational complexity, user input, adaptivity, etc. [26].

IV. Data Discovery through our Proposed D2C-ICT Architecture For Smart Cities

A. From Decentralized-to-Centralized ICT (DC2C-ICT) Architecture to D2C-ICT Architecture

In our previous studies, we first designed the Decentralized-to-Centralized ICT architecture based on Fog-to-Cloud technologies [3]. The proposed Decentralized-to-Centralized ICT architecture has facilities to manage data from sensors to Cloud technologies. Note that this architecture is only designed to manage physical data sources (sensors and actuators) in smart cities. Plus, the Fog layers can only communicate with each other through Cloud technology. Next, we designed a fully hierarchical D2C-ICT architecture for smart cities based on using Fog, cloudlet, and Cloud technologies [6]. This D2C-ICT architecture is able to organize non-physical data sources as well as physical data sources in smart cities. Some examples of the non-physical data sources in the city are external data consumer databases and third-party application databases. Therefore, D2C-ICT architecture can organize all physical and non-physical city data sources across different distributed-to-centralized technologies. In addition, D2C-ICT architecture can manage all city- and non-city data sources in centralized technologies. Finally, the distributed layers can communicate with each other through cloudlet technologies, and the distributed layers can communicate with Cloud and distributed technologies through Cloud technologies.

B. MAUT Cost Model

The stated criteria compare different cost model approaches that are presented in [26]. The basis of the comparison is divided into six columns, as described in [26], including approach name, computational complexity, user input, comparison, adaptivity, and strict constraints. Finally, we found that the MAUT [14] cost model is a suitable option for our performance measurement through our proposed D2C-ICT architecture for smart cities [26].

The MAUT cost model is an approach to solving multi-attribute/criteria decision or optimization problems using utility functions to rate the different KPIs [14]. It works almost the same way as the linear programming approach in the Simple Cloud Provider Selection (SCPS) algorithm [24], but it differs in how it calculates the scores of each attribute. It includes two extra steps, the first being the identification of the values of the KPIs themselves, and the second being the assigning of utility values to the KPI values identified. Like in the linear programming approach, preference weights are set, and the score for each solution is calculated by multiplying the utility value of each objective in the solution with its weight and summing up the results for each objective. An example of the MAUT function is shown in Formula 1, where each KPI and its measurement passed through a utility function ui(x) and multiplied by its weight wi. An example of a user-defined utility function is shown in Formula 2, which maps latency measurements to utility values. The utility functions could be any function that takes in measurement and returns a utility score between 0 and 100, where 100 is the optimal utility.

$$score = \sum_{i=1} u_i(x) * w_i \qquad \text{(1) taken from [14]}$$

$$u_{\text{latency}}(x) = \begin{cases} 100 & x \leq 25ms \\ 75 & 26ms \leq x \leq 50ms \\ 50 & 51ms \leq x \leq 75ms \\ 25 & 76ms \leq x \leq 100ms \\ 0 & 101ms \leq x \end{cases} \qquad \text{(2) taken from [14]}$$

As both services and data can have several KPIs, if the selection is dependent on more than one KPI, it makes the problem of selecting among these a multi-criteria decision problem. If you have a single criterion when selecting, the selection is trivial in many cases. For example, deciding to buy a car, if the only criterion is the price, then you pick the cheapest one. If you introduce two criteria, like price and comfort, it is suddenly not that easy. As there exists no value relation between the two criteria, how much more you are willing to spend on more comfort is highly individual. The difficulty arises because these problems give rise to a set of trade-off optimal solutions (also known as Pareto-optimal solutions), instead of a single optimal or a set of equally optimal solutions [29]. An optimal trade-off solution is a solution that is not dominated by another alternative. Being dominated means that there exists an alternative that is better on every criterion (strongly dominant). Fig. 1 shows an example of trade-off optimal solutions, and a single dominated alternative d, where d has a higher cost and lower comfort than solution A and B.

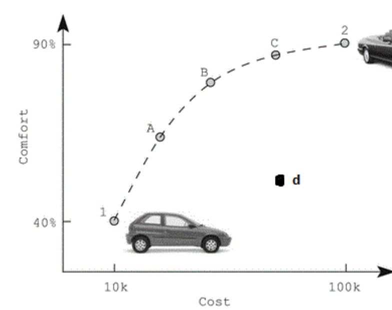

Fig. 1. Five trade-off optimal solutions [29]

There are two main approaches to selecting among these alternatives. The first involves finding all the trade-off optimal solutions and then selecting one in the set of trade-off optimal solutions. This is an approach fit for human decision-makers that use their intuition and knowledge to choose among a smaller set of alternatives. The other approach is called scalarization [29], which is an approach to convert the multi-criteria decision problem into a single-criteria decision problem, which is easy to solve and can be easily be automated. This approach, as the cost model, takes in a set of KPI measurements for the alternatives and produces a ranking of the alternatives. That ranking can then be used to select from, by picking the best-ranked alternative. Fig. 2 shows an example of a cost model used for three alternative cars with cost and comfort KPIs. After ranking and scoring the alternatives with a cost model, the one with the highest score is chosen.

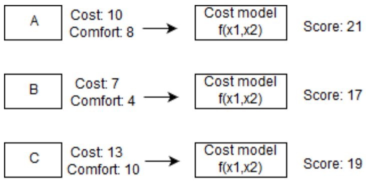

Fig. 2. Three alternatives with their KPIs [29]

V. USE CASE FOR DATA DISCOVERY: ZEN CENTER

The ZEN center contains eight different pilot projects in diverse cities in Norway [30]. The ZEN ICT architecture must be capable of including multiple ICT applications/services/tools for the researchers and partners concerning their business demands. Therefore, various data sources must cooperate to build ZEN services/tools.

A. KPIs and Toolbox for ZEN center

The ZEN center [30] defines a set of KPIs to test and analyze the performance of pilot projects [31]. ZEN KPIs are "a set of quantifiable performance measurements that defines sets of values based on measurement data from a project, making sure to measure and track the neighborhood's performance over time and against other similar projects." Therefore, there is a need for a clearly defined set of assessment criteria and KPIs that can be used to create methods and tools to assess the progress of the ZEN pilot project in terms of reaching their stated goals. This KPI data needs to be collected and made available from the pilots.

B. D2C-ICT Architecture for ZEN center

As we discussed in [6, 7, 16], we introduce a completely hierarchical distributed (from sensors and IoT devices) to centralized (Cloud computing technologies) ICT architecture for the ZEN center as a context of smart cities. Our D2C-ICT architecture for the ZEN center is based on Fog, cloudlet, and Cloud technologies [6, 16]. The preliminary basis of the recommended ICT architecture is proposed throughout the two main axes, Time and Location. Those axes demonstrate our idea about the ZEN ICT management in smart cities through the concept of the "ZEN center requirements," "data management architecture," "software management architecture," and "technology layers."

C. Using the MAUT cost model for data discovery in the ZEN center

The cost model design is based on the underlying ZEN decentralized-to-centralized ICT architecture [2, 32], as shown in Fig.3. There are two "Integrated and Intelligent Control and Monitoring of IoT" (I2CM-IoT) in Fog-Layer-2, which are in two different cities of Norway, Trondheim, and Bergen. Furthermore, there is also an I2CM-IoT is in the Cloud, which may be in a different country (in our test it is in Finland). The architecture supports adding more units in Fog-Layer-2, but for simplicity, only two are included in the design. The solid arrows represent the communication between I2CM-IoT s, while the stippled arrow represents communication between I2CM-IoT s and data repositories.

It is a distributed design, where different instances of what in this case is called "I2CM-IoT" are hosted in the Fog-Layer-2 and Cloud layer and can communicate with each other. The idea behind the design is that I2CM-IoT can be queried and get a response but being in a city where an I2CM-IoT in the Fog-Layer-2 layer is hosted. As a result, it is better to find data from that city. In this case, the I2CM-IoT is kept separate from the data repositories, but could just as easily be implemented as a part of the data storage. In the scenario outlined for the ZEN, there might be different actors within the same city, like an energy provider, a municipal entity, or a research team that they keep data from the city that might be accessed.

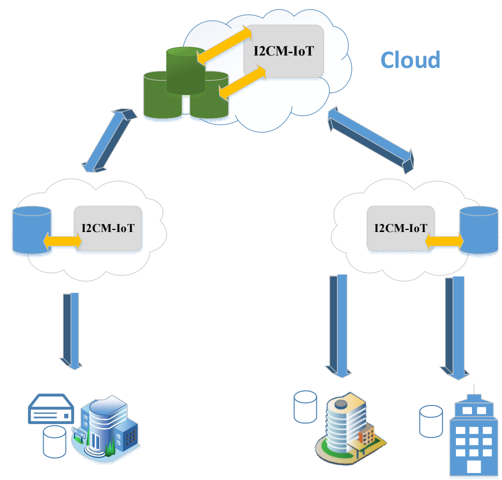

Fig. 3. The ZEN cost model based on F2C ICT architecture

The I2CM-IoT consists of three major components, as discussed below. Those three components are "request handler," "cost model," and "routing." In addition to the I2CM-IoT, a GUI is designed as the point where the user interacts with the system, consisting of a configuration component to specify criteria for the search, as well as a search component that searches for specific data and displays the results. In Fig.4, the internal architecture and how the different components interact is shown, where solid arrows show internal control flow of the I2CM-IoT and request and response to the I2CM-IoT, and the stippled arrows show the queries that the routing component might make. How the control flows from a query to the I2CM-IoT, to its response, is shown in the sequence diagram in Fig.5.

- *Request Handler:* The request handler organizes the incoming queries, from either a user or another I2CM-IoT and is responsible for forwarding the request to the other components and responding to the query once the other two components are used.
  The request handler acts as the interface to the I2CM-IoT. Two types of communications would access the request handler, the first being user requests in the form of a search query together with a description of the cost model

parameters, which returns a ranked list of all the suitable alternatives. The other is a request from another I2CM-IoT that needs information about data repositories that the I2CM-IoT knows about, in the form of a search query without a cost model configuration and returns all results that match the query. The search query consists of the location where the data was created, the time it was created, and the type of data it is, like temperature measurements.

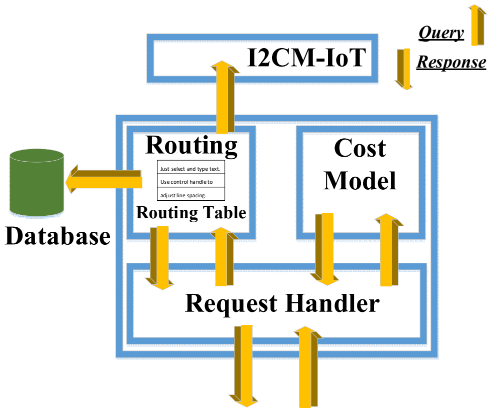

Fig. 4. Internal architecture of an I2CM-IoT [26]

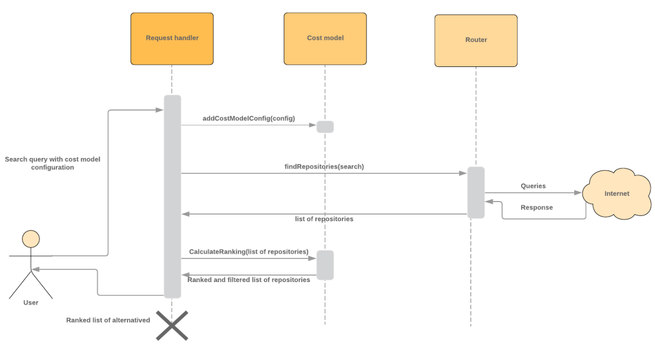

Fig. 5. Control flow for user query [26]

- *Cost Model:* The second component is the "cost model," which is responsible for filtering and calculating a ranking between all viable alternatives that are found based on the criteria given by the user in the query. It can either return the ranked results or the highest-ranked alternative.
  The cost model designed for the ZEN proposal is based on that MAUT approach [14]. The reason behind using this is that it is a fairly intuitive approach that allows creating a reasonably detailed ranking function depending on the number of intervals done. Being a utility-based comparison makes it easier to target specific performance values than by using comparison to the relative performance of all alternatives. It is not very computationally demanding, as it only needs one iteration through all alternatives to find the best score. It also seemed easy to make an intuitive interface for that as well. However, as it does not have strict constraints in it, filtering results on strict constraints will be added as part ranking and selection. This is something that should give a variety of ways for a user to specify their demands for the results. So, the cost model configuration consists of a set of significant constraints that filters out alternatives based on some KPIs, and utility functions in the form of utility intervals and their weights. The final result set consists of all the alternatives that are scored. A pseudo-code implementation of the cost model is presented in algorithm 1. After gathering all the alternatives in alternatives, it iterates through them. For each alternative, it first iterates through the strict constraints in the filter, if the alternative breaks a constraint, it fails, and it is nothing added to the final results. If it is not filtered out, it iterates through the criteria for the ranking, calculating a score with the weight and utility function in each criterion c, adding it to the total score, before it is appended to the list of results together with its score.

ALGORITHM 1: The cost model calculation [26]

```
filters: a list of strict constraints;
critera: a list of criteria to rank with;
alternatives: a list of alternatives;
result = [];
for a in alternatives do
    add = true;
    for f in filters do
        if a breaks constraint f then
            add = false;
        end
    end
    if add == true then
        score = 0;
        for c in criteria do
            score = score + (c ranking for a)
        end
        result.append({score: score, data: a})
    end
end
```

- *Routing:* The last component is the router. The router depends on some of the search parameters, queries the location that might hold the type of data for metadata about the data and their service properties.
  The routing component is responsible for querying all the places that might have the data the user is looking for, specified by the search query sent to the request handler, and passed to the routing. It consists of a routing table and a set of query algorithms. The idea behind this approach is based on the distributed hash tables containing information on where to find data or files presented in [26]. The approach allows the different nodes of the system to hold partial information about all data stored in the system and knowing where to forward the queries if it does not hold that information itself. This means that an I2CM-IoT in the Fog-Layer-2 layer only needs to hold information about the data stored for the Fog-Area, limiting the amount of information kept and managed by it and that communication can go to I2CM-IoT placed locally, instead of through the Cloud I2CM-IoT, lowering response times. Any query to it for data from another Fog-Area is forwarded to the Cloud I2CM-IoT that will forward it to the correct Fog.
  The main reason for this approach is to assume that data generated in a building can only be found at certain places, at the Fog-Layer near where it was generated as real-time

data, at Fog-Layer-2 layer as last recent data kept in the Fog-Layer-2 repositories in the same city, or the Cloud as historical data. This can also be influenced by the frequency of central data collection and aggregation of KPI data. This could be extended to be done together with "type" and "time," as part of the routing entries, but is not done yet, focusing solely on location.

VI. CONCLUSION AND FUTURE MAPS

The contributions of this paper include designing a cost model for D2C-ICT architecture and tailoring ICT KPIs with business KPIs for data discovery in the smart cities. Therefore, we suggested the MAUT cost model for data discovery within a D2C-ICT architecture in the smart city.

As a part of our future map, we will develop our proposed cost model for the ZEN center.

ACKNOWLEDGMENT

This paper has been written within the Zero Emission Neighbourhoods in smart cities (FME ZEN). The authors gratefully acknowledge the support from the ZEN partners and the Research Council of Norway. The first author would like to express his appreciation to NTNU University for their support through his master thesis studies [26].

REFERENCES

[1] J. Jin, et al., "An information framework for creating a smart city through internet of things," IEEE Internet of Things journal, vol. 1, no. 2, pp. 112-121, 2014.

[2] A. Sinaeepourfard, et al., "A Big Data Management Architecture for Smart Cities based on Fog-to-Cloud Data Management Architecture," in Proceedings of the 4th Norwegian Big Data Symposium (NOBIDS), Trondheim, Norway, 2018.

[3] A. Sinaeepourfard, et al., "Fog-to-Cloud (F2C) Data Management for Smart Cities," in Future Technologies Conference (FTC), 2017.

[4] B. Tang, et al., "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in The Fifth ASE International Conference on Big Data, 2015, p. 28: ACM.

[5] F. Bonomi, et al., "Fog computing and its role in the internet of things," in Proceedings of the first edition of the MCC workshop on Mobile cloud computing, 2012, pp. 13-16.

[6] A. Sinaeepourfard, et al., "F2c2C-DM: A Fog-to-cloudlet-to-Cloud Data Management Architecture in Smart City," presented at the IEEE 5th World Forum on Internet of Things (WF-IoT), Ireland, 2019.

[7] A. Sinaeepourfard, et al., "D2C-SM: Designing a Distributed-to-Centralized Software Management Architecture for Smart Cities," presented at the Proceedings of the 18th IFIP Conference on e-Business, e-Services, and e-Society (I3E), Norway, 2019.

[8] A. Sinaeepourfard, et al., "Data preservation through Fog-to-Cloud (F2C) data management in smart cities," in IEEE 2nd International Conference on Fog and Edge Computing (ICFEC), 2018, pp. 1-9.

[9] D. M. Strong, et al., "Data quality in context," Communications of the ACM, vol. 40, no. 5, pp. 103-110, 1997.

[10] S. R. Jeffery, et al., "Declarative support for sensor data cleaning," in International Conference on Pervasive Computing, 2006, Springer.

[11] Building Return on Investment from Cloud Computing – Cloud Computing Key Performance Indicators and Metrics. Available: http://www.opengroup.org/cloud/wp_cloud_roi/p5.htm

[12] B. Combest. (2016). The key performance indicators (KPIs) you should use to measure your success in the cloud. Available: https://blogs.partner.microsoft.com/mpn/the-key-performance-indicators-kpis-you-should-use-to-measure-your-success-in-the-cloud/

[13] S. Sengupta, et al., "An architecture for resource management in a fog-to-cloud framework," in European Conference on Parallel Processing, 2018, pp. 275-286: Springer.

[14] D. Von Winterfeldt, et al., "Multi-attribute utility theory: models and assessment procedures," in Utility, probability, and human decision making: Springer, 1975, pp. 47-85.

[15] M. Fiedler, et al., "A generic quantitative relationship between quality of experience and quality of service," IEEE Network, vol. 24, no. 2, pp. 36-41, 2010.

[16] A. Sinaeepourfard, et al., "A Distributed-to-Centralized Smart Technology Management (D2C-STM) Model for Smart Cities: a Use Case in the Zero Emission Neighborhoods'," presented at the IEEE International Smart Cities Conference (ISC2 2019), Morocco, 2019.

[17] P. Jekosch, "Voice and Speech Quality Perception: Assessment and Evaluation-Ute," 2005.

[18] F. Mehdipour, et al., "FOG-Engine: Towards big data analytics in the fog," in IEEE DASC/PiCom/DataCom/CyberSciTech, 2016, pp. 640-646: IEEE.

[19] A. V. Levitin, et al., "A model of the data (life) cycles with application to quality," Information and Software Technology, vol. 35, no. 4, pp. 217-223, 1993.

[20] H. Hu, et al., "Toward scalable systems for big data analytics: A technology tutorial," IEEE access, vol. 2, pp. 652-687, 2014.

[21] P. M. Barnaghi, et al., "Challenges for Quality of Data in Smart Cities," J. Data and Information Quality, 2015.

[22] I. Rec, "E. 800: Definitions of terms related to quality of service," International Telecommunications Union, Geneva, 1994.

[23] A. Lavignotte, et al., "Quality of Experience, a very personal experience!," in 2013 24th International Workshop on Database and Expert Systems Applications, 2013, pp. 231-235: IEEE.

[24] M. Souidi, et al., "An adaptive real time mechanism for IaaS cloud provider selection based on QoE aspects," in 2015 IEEE International Conference on Communications (ICC), 2015, pp. 6809-6814: IEEE.

[25] K. Bauer, "KPIs-The metrics that drive performance management," Information Management, vol. 14, no. 9, p. 63, 2004.

[26] T. Kirkevik Soltvedt, "A Distributed-to-Centralized Cost Model for Service Selection in Smart Cities," Master thesis, Computer Sceince, Norwegian University of Science and Technology (NTNU), 2019.

[27] L. Lin, et al., "Computation Offloading Toward Edge Computing," Proceedings of the IEEE, vol. 107, no. 8, pp. 1584-1607, 2019.

[28] M. Eklin, et al., "Model for cost estimation in a finite-capacity stochastic environment based on shop floor optimization combined with simulation," European journal of operational research, vol. 194, no. 1, pp. 294-306, 2009.

[29] K. Deb, "Multi-objective optimization," in Search methodologies: Springer, 2014, pp. 403-449.

[30] Research Center on zero emission neighbourhood. Available: https://fmezen.no/

[31] A. H. Wiberg et al., "ZEN TOOLBOX: First concept for the ZEN Toolbox for use in the development of Zero Emission Neighbourhoods Version 1.0," 2019, Available: https://fmezen.no/category/publications/reports/.

[32] A. Sinaeepourfard, et al., "A Zero Emission Neighbourhoods Data Management Architecture for Smart City Scenarios: Discussions toward 6Vs challenges," in 2018 International Conference on Information and Communication Technology Convergence (ICTC), 2018, pp. 658-663: IEEE.

[33] A. Sinaeepourfard, et al., " D2C-DM: Distributed-to-Centralized Data Management for Smart Cities Based on Two Ongoing Case Studies," in Proceedings of SAI Intelligent Systems Conference, 2019, pp. 619-632: Springer.

[34] A. Sinaeepourfard, et al., "Large-Scale Information and Communications Technology (ICT) Management in Smart Cities based on Edge to Cloud Orchestration," in IEEE Conference on Omni-layer Intelligent systems (COINS), 2020.

[35] J. Krogstie, "Evaluating Data Quality for Integration of Data Sources.," in The Practice of Enterprise Modeling, Springer, vol. 165, pp. 39-53, 2013.