A Novel Edge-to-Cloud-as-a-Service (E2CaaS) Model for Building Software Services in Smart Cities

Jaro Robberechts*, Amir Sinaeepourfard§, Tom Goethals‡, and Bruno Volckaert‡

Department of Information Technology*‡ - Department of Computer Science§,
Internet Technology and Data Science Lab (IDLab)‡ - imec‡,
Ghent University*‡ - Norwegian University of Science and Technology§,
Ghent, Belgium*‡ - Trondheim, Norway§
{jaro.robberechts, togoetha.goethals, bruno.volckaert}@ugent.be*‡, a.sinaee@ntnu.no§

***Abstract*—The main goal of a smart city is to enhance the quality of life of its inhabitants by providing services using Information and Communications Technology (ICT) components in a city. ICT components include not only Internet of Things (IoT) data sources spread across the city, but also traditional non-IoT data sources. Managing all ICT components in a smart city can be challenging and results in many complexities. Consequently, there is a need for ICT management architectures. Traditional solutions are often based on a centralized ICT architecture using Cloud technologies. Recently, the number of ICT components, services, and their corresponding complexities are growing, leading to large-scale ICT architectures. Centralized Cloud solutions cannot cope with the ever-expanding demands of this kind of architectures. The limitations of the centralized approaches necessitate the design of a new ICT architecture, using distributed technologies, for every layer and element of the city. Many solutions for management from Edge-to-Cloud (E2C) through distributed technologies are forthcoming, including Decentralized-to-Centralized ICT (DC2C-ICT) and Distributed-to-Centralized ICT (D2C-ICT) architectures. The DC2C-ICT architecture and its components work on their own tasks and are solely communicating with a centralized platform. On the other hand, components of the D2C-ICT architecture can work together to provide the services for the citizens across different layers from E2C. Therefore, the D2C-ICT architecture is less dependent on the central Cloud-based entity, but harder to design and manage. In this paper, an "Edge-to-Cloud-as-a-Service (E2CaaS)" model is proposed together with a model on how to build efficient software services in smart cities through different layers of E2C. The most important tasks for building these services are the management of "Data/Database," "Resources," and "Network Communication and Cybersecurity issues."**

Keywords-Smart City; IoT; Edge-to-Cloud; Distributed-to-Centralized ICT architecture; Edge-to-Cloud-as-a-Service;

I. INTRODUCTION

One of the main purposes of a smart city is to offer efficient software services to its citizens, companies, organizations, and government. The most important ingredient for building such software services is data, which is widely available in every city and originates from a variety of sources. When offering a service, some data will have to be processed on a computing platform to provide value. In a traditional centralized ICT architecture, the computing platform is located in the Cloud. Recently, a novel paradigm, called Edge computing, aims to move the computing resources and storage closer to the data sources. While this concept brings many advantages like reduced latency, improved privacy, and less pressure on network communication traffic and data centers [1]. It also causes extra complexities that have to be solved efficiently.

II. RELATED WORK

Building software services is one of the most crucial tasks in all ICT-based solutions (such as smart cities). The progression and classification of the "as-a-service (aaS)" concept have been ongoing from 1984 until the present, showing that "aaS" is a continuously evolving terminology. E.g., software is proposed as a service in 2000 and later redefined as a Cloud service such as "Infrastructure-as-a-Service (IaaS)," "Platform-as-a-Service (PaaS)," and "Software-as-a-Service (SaaS)" [2]. The classification of the Edge of smart city networks as a Service is a complex task involving different distributed to centralized technologies, such as Fog, and cloudlet. Some examples of "aaS" services through E2C orchestration are "Edge-as-a-Service (EaaS)," "Mobile Edge Computing-as-a-Service (MECaaS)," "Fog-as-a-Service (FaaS)," and "cloudlet-as-a-Service (caaS)." Fig. 1 shows the classification of the "aaS" model in a smart city through different layers of "E2C" technologies.

A. "Cloud-as-a-Service (CaaS)"

"CaaS" is a model that enables individuals or business companies to let specialized companies manage their Information System instead of hosting and managing the computing resources themselves [2]. Because the consumer hires the computing resources and does not have to buy and maintain expensive hardware, this model is advantageous for medium and small-sized companies who are not able to invest in this sort of equipment. The "CaaS" model is located as a top layer of the D2C-ICT architecture and is also visible as the top layer of Fig. 1. The main three models in "CaaS" are "IaaS," "PaaS," and "SaaS," as we described below.

• "IaaS" enables the provider to rent out the infrastructure used to run the software of the client, often in terms of Virtual Machines (VMs), processing or storage [3]. This gives a lot of freedom to the client because he/she can choose what kind of software stack to run on the

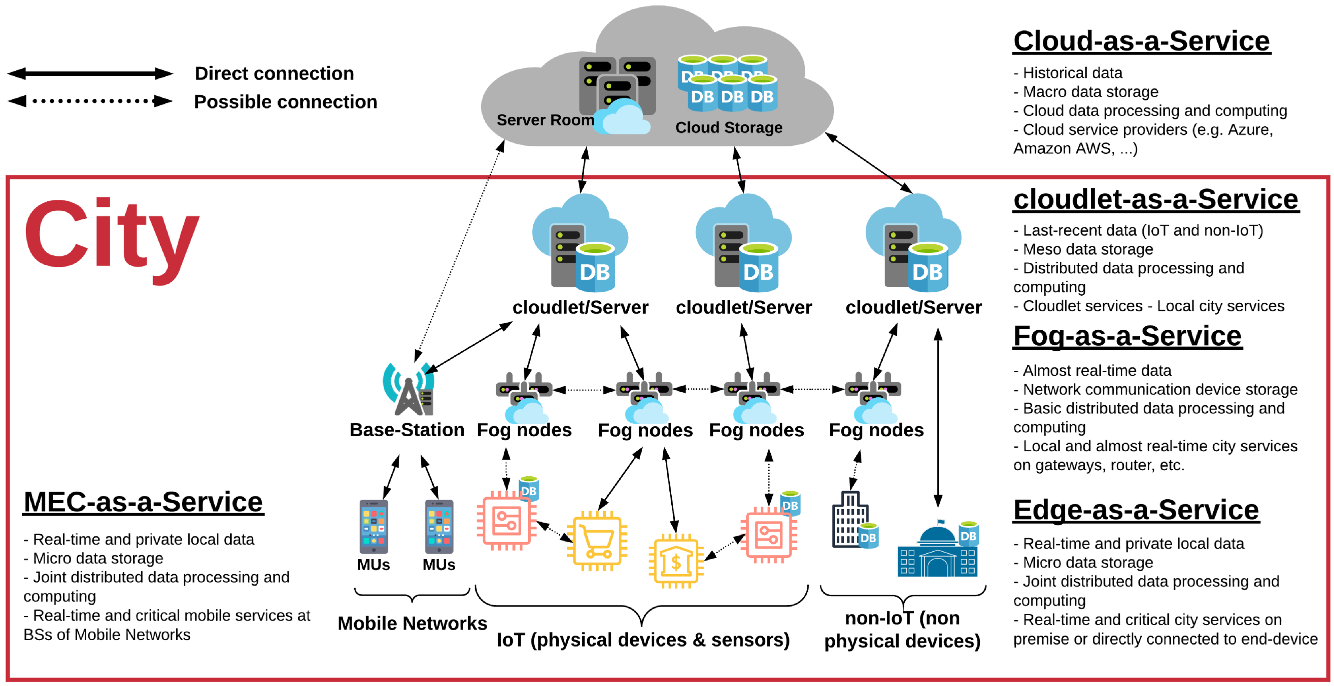

Figure 1. Classification of "aaS" model through the D2C-ICT architecture in a smart city

infrastructure components, which include computers, storage, networking, and networking services. This type of "Cloud as a Service" is ideal when having a limited hardware budget. Some use cases for IaaS are file backups, and product design [3].

• "PaaS" is a model where a service provider rents out the infrastructure with the operating system and databases. This model is used for quickly deploying applications [3].

• "SaaS" offers applications to clients via the web. The applications can mostly be accessed through a web browser, not requiring any software installation. The client has no control over the underlying infrastructure and limited control over the application settings [3].

B. *Edge of smart city networks as a Service*

Edge of smart city networks as a Service can be realized by different distributed technologies [4,5] and demonstrated by different service models, "EaaS", "MECaaS", "FaaS", and "caaS". The models can be applied through three layers of city/Edge networks as shown in Fig. 1.

1) *"EaaS"* includes the IoT and non-IoT data sources in the Edge layer as shown in Fig. 1 and is a service model that offers services to citizens through either executing the computation tasks on the devices of the citizens themselves or through computational devices that are directly connected to the citizens' devices [6]. The services are deployed either on-premise or on a device near the citizens. If the service is not deployed on-premise, the citizens have to be able to connect via LAN to other nearby Edge devices. The Edge device can forward IoT and non-IoT data or connect directly to the higher layers (i.e. Fog/cloudlet) for further processing and storage requirements. This model allows for building "critical," "real-time," and "private" services that require fast data processing with low latency, low operation cost, and low network traffic. The data that is processed in the Edge layer, is mostly "private data" from city consumers.

2) *"MECaaS"* is in the left-hand side of the lowest layer of Fig. 1 and is a computing model that tries to solve the limitations of Mobile Users (MUs) by offering computing services close to the MUs. The "critical" and "almost real-time" services are deployed at the base-stations of the mobile networks [7]. Other service types can also run on the higher layers (cloudlet/Cloud).

3) *"FaaS"* is located at the second-lowest layer (Fog layer) of Fig. 1. Fog nodes in this layer are often network devices such as "gateways." Due to the limited computation power of the "Fog nodes", this model is mostly used for building less demanding services or data management actions like data reduction, which decreases the network traffic towards the Cloud/cloudlets, or data analysis. Plus, this study [8] proposes a FaaS technology and architecture which is built on three layers: the "Infrastructure", "Platform", and "Software" layer, identical to the available service idea for "CaaS". These

three service layers only apply and execute services onto the "Fog nodes". The proposed architecture provides faster service responses and efficient use of resources.

4) "caaS" is located at the third-lowest layer (cloudlet layer) of Fig. 1 and uses cloudlet technologies. A cloudlet is a small data center/server in a box, computing device, or cluster of computers that brings the functionalities of the Cloud closer to the Edge of the network, thereby reducing the latency and network traffic. cloudlets not only include physical servers but also virtual servers. There are not many studies that specifically mention "caaS" model. However, [9] proposed a Cloud-based framework which uses cloudlets as service providers. In this framework, the main application runs on mobile devices. When a mobile device does not have the required computing power to execute a certain task, the device connects to a nearby cloudlet which will execute the task. The orchestration of the cloudlets and mobile devices is organized by a root server in the Cloud.

Focusing on the "FaaS" model, [10] defined how software services can be built at the Fog layer through a systematic literature review. The paper proposed a taxonomy that categorizes current work on Fog computing applications in smart cities. The main three categories are: "Service Objective," "Application Classification," and "Outcome Type." The "Application Classification" category seeks to place current work into groups based on their application domain (e.g. Smart Building), in a manner that the groups have minimum overlap. The "Service Objective" defines what the goals can be for a Fog application. The third and final main category of the taxonomy, the "Outcome Type" considers three main types of solutions: Architecture, Framework, and Platform.

The main limitation of the "aaS" model in smart cities is the following:

- How can we build software services in large-scale ICT networks of smart cities from smallest to largest scale? This solution must provide facilities to use different multilevel distributed and centralized technologies in combination with a different scope of ICT management strategies. Therefore, this paper presents a novel "E2CaaS" model for building software services in smart cities.

III. Software Services in large-scale ICT networks of smart cities: ZEN as a use case

This section goes into detail on how to build software services in the Zero Emission Neighborhoods (ZEN). ZEN is located in Norway and researches no emission of greenhouse gas in neighborhoods through eight different city pilots [11]. Subsection III-A goes into detail about the process of building software services. Subsection III-B discusses our proposed "E2CaaS" model.

A. Different steps for building Software Services

Deciding where and how the software services should be built in a smart city can be divided into four main steps. The first step is the classification of city services. This step classifies the service and defines the service objectives. The second step is to design the output. This step goes deeper into the design of the solution. The thirds step is the implementation of the design. The final step is the efficiency measurements. The different steps can be seen in Fig. 2 and will be discussed below.

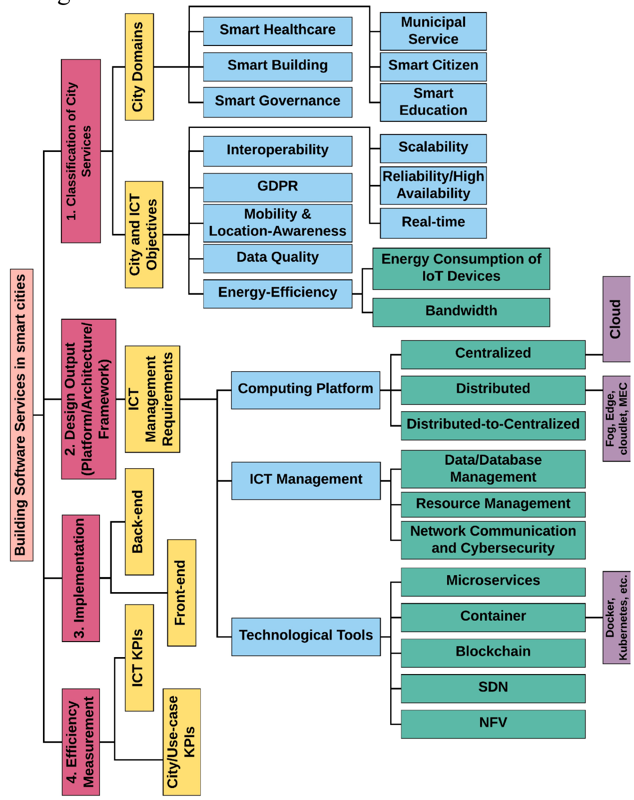

Figure 2. Building software services in smart cities

1) Classification of City Services can be organized in two steps. The first step is defining the city domains for the service, depending on the target audience and/or the type of organization. The second step is determining the city and ICT objectives for the service. These objectives mostly depend on the facilities of the domain.

We have defined some parameters, as discussed below.

- "Interoperability" means that the different computing and networking nodes from the different layers of the smart city must work together efficiently.
- "Mobility and Location-Awareness" means that services may have the possibility to handle the ability of citizen/device movement between different levels of the city, e.g., smartphone, and drones.
- "Real-time" means that the service must respond immediately, requiring a latency under a specified value. e.g., emergency health-care services.
- "Scalability" means that the services must be easily expandable when the amount of tasks increases.
- "Reliability/High Availability" means that the service should be able to withstand node failures and is often expressed in terms of uptime.

• "Data Quality" means that the services can provide certain requirements in terms of the quality of data.
• "Energy-Efficiency" has different domains in a smart city. ICT-based examples for the energy efficiency are: i) bandwidth management inside the city network communications; ii) energy consumption of IoT devices.

2) The *Design Output* will result in an Architecture, a Framework, or a Platform [10]. The main step for designing the output is exploring the ICT Management requirements. This step can be divided into three main categories: "Computing Platform," "ICT management," and "Technological Tools."

a) The *"Computing Platform"* can be either "Centralized", "Distributed" or "Distributed-to-Centralized." Centralized computing platforms are often realised using Cloud-based technologies, meaning that all technology resources for computing, storage, and other data management phases are located in the Cloud. Distributed computing platforms involve multiple devices that are spread across the city [12]. These devices can work together to combine their computational power and perform higher demanding tasks. Distributed-to-Centralized computing platforms use both potentials of distributed and centralized technologies for processing and storage at the same time, such as Fog-to-Cloud (F2C) [12,13] and Fog-to-cloudlet-to-Cloud (F2c2C) [4,5,14].

b) *"ICT Management"* is an important aspect in smart cities due to the wide variety of available ICT resources and components. Currently, authors [5] mention that there are three main categories for ICT management in large-scale ICT networks of smart cities: "Data/Database management," "Resource management," and "Network Communication and Cybersecurity issues management." These three concepts have to be managed efficiently and have to work together to meet the service objectives for citizens of smart cities.

• "Data/Database management" involves every activity in the life cycle of the data in a smart city, including "Data Acquisition," "Data Preservation," and "Data Processing." Further details are available in [1,15].
• "Resource management" involves the efficient organization of the various types of resources in a smart city. The main challenges are related to resource discovery, resource provisioning, and resource scheduling and load balancing [6]. Resource discovery handles the identification of the resources and services inside the city and is mandatory for the system to be able to find the best-suited computing nodes for a service. This selection procedure is called resource provisioning and involves selecting the optimal computing nodes and the placement of the services and VMs onto these nodes. Finally, resource management also takes care of resource scheduling and the offloading/load balancing.
• "Network Communication and Cybersecurity issues management" is related to the communication between the ICT components. The access from and to the network resources can be controlled with tools like Network Function Virtualization (NFV) or Software-Defined Networking (SDN) [16]. These tools will be explained briefly in paragraph III-A2c. The communication between the network nodes has to be secure, as often a lot of sensitive data is processed in a smart city. Many solutions have already been proposed related to network security. E.g., Blockchain [17] or SDN [18].

c) *Technological Tools:* Some common tools for ICT management are listed below.

• "Microservices" are small services that are often limited to one or a couple of subtasks, which are defined by decomposing a complete task. This enables the code of the task to be run on smaller devices and to distribute the processing, making the system more responsive and fault-tolerant [19].
• "Container" technologies are lightweight virtualization technologies, enabling the deployment and execution of large-scale distributed applications on Cloud, cloudlet, Edge/Fog, and IoT platforms [20]. Containers are useful for large-scale applications because of easy life-cycle management, negligible overhead, excellent start, restart, and stop times compared to VMs and application portability. The main components of container technologies are the containers itself, the container manager, and the container orchestrator. The type of container depends on the container technology of choice, i.e. Docker, LXC, OpenVZ, etc. The container manager provides an Application Programming Interface (API) to manage the life-cycle of the containers. The orchestrator enables the application provider to manage the deployment, monitoring, and the configuration of multi-container applications. Examples of well know orchestrators are "Kubernetes" and "Docker Swarm."
• "SDN" is an approach to networking where the data/forwarding plane and the control plane are separated by creating a virtualized control plane that manages network functions, thereby bridging the gap between service provisioning and network management. The network becomes directly programmable using Southbound Interfaces (e.g. "OpFlex", "OpenFlow", etc.). This type of network management results in a more flexible network that can adapt to changing network conditions, business, market and citizens' needs [16].
• "NFV" decouples Network Functions (NFs) from their specialized hardware, which enables them to be deployed on top of general-purpose hardware, greatly reducing the hardware cost for the service provider. Also, NFV enables the NFs to be easily deployed and dynamically allocated, resulting in a more scalable and flexible network [16].
• "Blockchain" originates from digital currency but is now being used in a variety of other technological domains. Due to its abilities in terms of consistency and integrity of data during transmission, its transparency, and its distributed nature, it is a viable option to protect network communication and the identity of the devices in distributed systems and IoT environments without the need for a trusted third party [17].

• "Machine Learning" can be used for the optimization of resource provisioning and improvement of the network security (e.g. anomaly detection methods).

3) "Implementation": The thirds step is implementing the output from the previous step. This process involves selecting the right frameworks and technologies for the front-end and the back-end of the service. The result will be a service that receives data from the city or Cloud in the back-end. The back-end of an application regulates how the application works behind the scenes. The citizen can access the result of the service via the front-end, which interacts with the back-end through an API or an SDK (Software Development Kit) [21]. An API is an interface which allows different layers of an application to communicate through a collection of definitions. An SDK is a package of software development tools such as APIs, libraries, code parts, etc.

4) "Efficiency Measurements": We consider two types of efficiency measurements in this study: ICT Key Performance Indicators (KPIs), and city/use-case KPIs. A KPI is a measurable value that indicates how well the service achieves key business objectives. KPI values are often obtained by performing simulations, measurements, or surveys about the service [14]. ICT KPIs [5] are based on the performance of "Data/Database management," "Resource management," and "Network Communication and Cybersecurity issues management." ICT KPIs are factors like bandwidth and latency which are defined by the city and the domain of the application. In our use case, ZEN has defined some KPIs and assessment criteria that have to be taken into account when building software services for the ZEN and its pilots.

B. Our proposed "E2CaaS" model

There are many technological resources available in a city/Cloud environment. The orchestration of these resources must be done through network communication between the different layers. So far, city-data is produced and sent to the services that are located in the Cloud through network communication in centralized orchestrations. On the other hand, distributed technology has the facility to manage data and provide services in their own layer, close to the data-sources in a city. Current orchestrations do not focus on solutions using multilevel distributed and centralized technologies for building software services. The "E2CaaS" model can cover this gap by providing facilities to generate software services in a city concerning the "Efficiency Measurements" and "Service Objectives" of the large-scale ICT networks.

In our D2C-ICT architecture [1,4,5], we already pointed out that the cloudlet is a good option to manage the large-scale ICT networks through different KPIs and requirements. In this paper [5], we designed "I2CM-IoT (Integrated and Intelligent Control and Monitoring of IoT)," where the cloudlets act as a control layer. The motivations for positioning the control in the cloudlet are listed below.

• The cloudlets can be seen as a middleware layer between a strong Cloud and the Edge of the network;

• The cloudlets are located inside the city, close to the data sources and citizens;

• Because of the city location of the cloudlet, they are suited for applying local city policies and data privacy and GDPR;

• A cloudlet can be like an external server with more computing and storage facilities in a city and somehow in transit between Edge to Cloud;

• cloudlets can provide efficient orchestration of all physical sources management in the city and Cloud.

By moving the control of the resources to the cloudlet layer, we can provide efficient services based on the Cloud service solutions, "IaaS", "PaaS", and "SaaS. These service models are described below for the E2C layers, with the "I2CM-IoT" applied in the cloudlet.

• "IaaS-E2C" enables the client to rent some possibly distributed infrastructure located inside the city on which it can deploy its containers. The client has, as in the Cloud, the option to choose whatever software stack is deployed on this infrastructure. As there is a wider variety of infrastructure available in the city, the customer has more freedom to choose the type of infrastructure that suits best.

• "PaaS-E2C" is similar as in the Cloud, the service provider rents the infrastructure and controls the type of operating system and databases. This allows the client to quickly deploy applications in the city.

• "SaaS-E2C" allows service providers to rent out software services tailored to certain business use cases. The software services, located in the city, benefit from improved privacy and security and, better connectivity to citizens.

To realize these facilities when moving the control to the cloudlets, the ICT components in the city must be managed efficiently across all the E2C layers. The categories of ICT management are discussed below in a general scenario. In the specific scenario of "real-time" or "critical" services, the services should be offered directly at the Edge of the network through e.g. Edge and MEC technologies to reduce latency and improve service performance for the citizens.

• *"Data/Database management":* There are many different databases and data types spread across the city and Cloud (e.g. IoT and non-IoT data). The cloudlet is in a position to manage all this data by connecting the Cloud and Edge database platforms. This allows for services to be built based on user requirements and capacity, e.g. "private" and "local" data is not shared outside the city if not necessary, and required data residing in clusters spread across the city and Cloud can be collected and processed efficiently.

• *"Resource management":* A city has many heterogeneous devices offering computing resources, which should be managed efficiently. The cloudlet can orchestrate and manage these resources due to its central position between the Edge and the Cloud. By monitoring the load on the resources, the cloudlet can schedule tasks and distribute the load based on the service requirements. This idea allows for resource management based on city capacity and user requirements.

• *"Network Communication and Cybersecurity issues management":* The cloudlet is in a good position to monitor traffic in the city as well as to the Cloud. The cloudlet orchestrates the network based on measurements, activities in the city (such as cyber-attacks), and user-requirements. This concept allows for better privacy solutions and better resource allocation.

These three categories should be managed together by the cloudlet in our "E2CaaS" model to meet the "Efficiency Measurements" and "Service Objectives." The concept of an integral solution for "E2CaaS" is explained using an example scenario. A citizen requests a service which requires a private connection, processing on computing resources inside the city, and data distributed over different clusters inside the city and the Cloud. First of all, the cloudlet retrieves the necessary data from the distributed Fog, Edge, cloudlet and Cloud clusters. Due to the central location of the cloudlet layer, the cloudlets can contact the city resources, as well as the Cloud to locate and/or retrieve data. Next, the cloudlet performs the "resource management" tasks. The communication for the previous steps should be performed efficiently by monitoring traffic over the city networks while respecting the privacy and the security of the citizen their data. The orchestration decisions made in the cloudlet are based on a multi-attributed cost model, composed of ICT and city/use case KPIs. In the case of the ZEN center, an integral solution must be directed towards particular services for their pilots and the specific requirements of those pilots.

Conclusion

In this paper, we proposed the "E2CaaS" model for building software services in large-scale ICT networks of smart cities. We envisage that the usage of different multilevel distributed and centralized technologies in combination with a different scope of ICT management strategies may allow us to obtain additional efficiency increment when creating efficient software cities in smart cities. These ICT technology management strategies include management of "Data/Database," "Resource," and "Network communication and Cybersecurity issues."

As part of our future studies, we will discover more options related to developing our "E2CaaS" by focusing on the ZEN center and its pilots' requirements. Our interests are to design, implement, and operate an integral solution for building Software Services in smart cities through "E2C" orchestration.

Acknowledgement

This paper has been written within the Zero Emission Neighborhoods in smart cities (FME ZEN). The authors gratefully acknowledge the support from the ZEN partners and the Research Council of Norway.

References

[1] A. Sinaeepourfard, et al., "D2C-DM: Distributed-to-Centralized data management for smart cities based on two ongoing case studies," Intelligent Systems and Applications, IntelliSys, Advances in Intelligent Systems and Computing, vol 1038, 2019.

[2] I. Adamu, et al., "Cloud computing as a service era: origins, current and future trends," IOSR Journal of Computer Engineering (IOSR-JCE), vol. 21, pp. 34-39, 2019.

[3] A. G. Prajapati, et al., "All About Cloud: A Systematic Survey," International Conference on Smart City and Emerging Technology (ICSCET), India, pp. 1-6, 2018.

[4] A. Sinaeepourfard, et al., "D2C-SM: Designing a Distributed-to-Centralized software management architecture for smart cities," Digital Transformation for a Sustainable Society in the 21st Century, Lecture Notes in Computer Science, vol 11701, 2019.

[5] 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," IEEE International Smart Cities Conference (ISC2), Morocco, pp. 760-765, 2019.

[6] A. Yousefpour, et al., "All one needs to know about fog computing and related edge computing paradigms: a complete survey," Journal of Systems Architecture, vol. 98, pp. 289--330, 2019.

[7] I. Elgendy, et al., "Resource allocation and computation offloading with data security for mobile edge computing," Future Generation Computer Systems, 2019.

[8] N. Chen, et al., ``Fog as a service technology," IEEE Communications Magazine, vol. 56, no. 11, pp. 95-101, 2018.

[9] R. Kumar, S.K. Yadav, "Scalable key parameter yield of resources model for performance enhancement in mobile cloud computing." Wireless Pers Commun 95, 3969–4000, 2017.

[10] G. Javadzadeh, G. Rahmani, "Fog computing applications in smart cities: a systematic survey," Wireless Networks, vol.26, pp. 1433-1457, 2020.

[11] Available: https://fmezen.no/about-us/.

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

[13] A. Sinaeepourfard, et al., "Fog-to-Cloud (F2C) data management for smart cities," Saiconference on FTC, Canada, 2017.

[14] A. Sinaeepourfard, et al., "F2c2C-DM: A Fog-to-cloudlet-to-Cloud Data Management architecture in smart city," IEEE 5th World Forum on Internet of Things (WF-IoT), Ireland, pp. 590-595, 2019.

[15] A.Sinaeepourfard, et al., "A data lifecycle model for smart cities," IEEE conference on ICTC, South of Korea, 2016.

[16] A. A. Barakabitze, et al., "5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges," Computer Networks, vol. 167, 2020.

[17] P. K. Sharma, et al., "A software defined fog node based distributed blockchain cloud architecture for IoT," IEEE Access, vol. 6, pp. 115-124, 2018.

[18] A.Sinaeepourfard, et al., "Cybersecurity in large-scale smart cities: novel proposals for anomaly detection from edge to cloud," The International Conference on Internet of Things, Embedded Systems and Communications, Tunisia, 2019.

[19] L. Wu, et al., "MicroRCA: root cause localization of performance issues in mircoservices," IEEE/IFIP Network Operations and Management Symposium, Hungary, Apr. 2020.

[20] E. Casalicchio, S. Iannucci, "The state-of-the-art in container technologies: application, orchestration and security," Concurrency and Computation: Practice and Experience, 2020.

[21] R. Tönjes, et al., "Real Time IoT Stream Processing and Large-scale Data Analytics for Smart City Applications," EU FP7 CityPulse Project.