information

Article

A New Edge Computing Architecture for IoT and Multimedia Data Management

Olivier Debauche [1,2,3,4,5,*,†], Saïd Mahmoudi [1,2,3,†] and Adriano Guttadauria [1,2,3]

1 Faculty of Engineering-ILIA, University of Mons, 7000 Mons, Belgium; said.mahmoudi@umons.ac.be (S.M.); adriano.guttadauria@umons.ac.be (A.G.)
2 Infortech, University of Mons, 7000 Mons, Belgium
3 Numediart, University of Mons, 7000 Mons, Belgium
4 GxABT-TERRA, University of Liège, 4000 Liège, Belgium
5 GxABT-BioDynE-DEAL, University of Liège, 4000 Liège, Belgium
* Correspondence: olivier.debauche@umons.ac.be or olivier.debauche@uliege.be; Tel.: +32-65-374-059
† These authors contributed equally to this work.

Abstract: The Internet of Things and multimedia devices generate a tremendous amount of data. The transfer of this data to the cloud is a challenging problem because of the congestion at the network level, and therefore processing time could be too long when we use a pure cloud computing strategy. On the other hand, new applications requiring the processing of large amounts of data in real time have gradually emerged, such as virtual reality and augmented reality. These new applications have gradually won over users and developed a demand for near real-time interaction of their applications, which has completely called into question the way we process and store data. To address these two problems of congestion and computing time, edge architecture has emerged with the goal of processing data as close as possible to users, and to ensure privacy protection and responsiveness in real-time. With the continuous increase in computing power, amounts of memory and data storage at the level of smartphone and connected objects, it is now possible to process data as close as possible to sensors or directly on users devices. The coupling of these two types of processing as close as possible to the data and to the user opens up new perspectives in terms of services. In this paper, we present a new distributed edge architecture aiming to process and store Internet of Things and multimedia data close to the data producer, offering fast response time (closer to real time) in order to meet the demands of modern applications. To do this, the processing at the level of the producers of data collaborate with the processing ready for the users, establishing a new paradigm of short supply circuit for data transmission inspired of short supply chains in agriculture. The removing of unnecessary intermediaries between the producer and the consumer of the data improves efficiency. We named this new paradigm the Short Supply Circuit Internet of Things (SSCIoT).

Keywords: Edge Computing; image analysis; Internet of Things; multimedia management; A2IoT

check for updates

Citation: Debauche, O.; Mahmoudi, S.; Guttadauria, A. A New Edge Computing Architecture for IoT and Multimedia Data Management. *Information* **2022**, *13*, 89. https://doi.org/10.3390/info13020089

Academic Editor: Antonio Jiménez-Martín

Received: 20 December 2021
Accepted: 10 February 2022
Published: 14 February 2022

1. Introduction

Multimedia data groups together different types of data such as sounds, videos, images and cartographic data. These types of data are characterized by large volumes and/or continuous flows of data. The processing of these types of data is generally carried out in the cloud out of convenience [1]. However, with the concomitant development of the Internet of Things (IoT), virtual reality, augmented reality, big data, and social networks have led to network congestion and an increase in processing times and the limits of the Central Cloud Computing (CCC) paradigm. Moreover, applications needs and user demands for near real-time reaction time have motivate the development of solutions to address congestion and reduce processing times.

Recent technologies have disrupted CCC to move storage and processing resources close to end-users and use also processing capabilities present in the neighborhood.

These ones have been motivated by the saturation of network and the need of response time to data queries in (quasi) real-time. Bringing processing closer to users at geographically distributed data centers (Cloudlets) and Micro Data Centers (MCD) have reduced latency for data transmission and querying. Subsequently, the increase in processing capacities at the level of routers and network equipment (Fog Computing) made it possible to offer new processing capacities closer to users. At the same time, the memory and processor capacities of the sensors have also evolved and it has been possible to offer processing capacities (Edge Computing) as close as possible to the sensors to verify, clean, encrypt, compress the data and lightly preprocess data in order to limit the amount of data that is sent to the cloud [1]. Processing and storage capacities and latency decrease the closer you get to the end user. The underlying question is where and when to perform the treatment. Different approaches have been developed in the academic and industrial world to try to provide a satisfactory answer to this question. **Osmotic Computing** (OC) uses micro virtual machines (MVM) that can be moved between the local level and the cloud depending on available resources. This paradigm is inspired by osmotic pressure to distribute tasks between Edge and Cloud level [1,2]. However, for the moment, security aspects in terms of privacy or data confidentiality are lowly or not managed. **Dew Computing** (DC) is a concept that aims to store data locally on the end user's device to enable them to have a local copy of data and allows them to consult them easily. However, this approach requires the implementation of a data synchronization protocol, and in the case of multimedia data generates significant traffic to store the data locally. **Mobile Edge Computing** (MEC) is born with the rapid development of the processing and storage capacities of mobile devices (mainly smartphones and tablets) and the need to follow devices in their displacements. It then expanded to also allow access via WiFi and is sometimes called Multi-access Edge Computing. In parallel, the deployment of 5G on a large scale will offer high throughput and ultra-low latency that will allow nomadic users to dispose of processing capacities that follow them on their displacements. However, this approach remains conditional on the effective deployment of 5G, the degree of coverage, as well as the compatibility of devices with 5G. A mixed approach called **Jungle Computing** (JC) seeks to federate the resources of all kinds available locally to carry out processing. The main difficulty with this approach is to federate widely heterogeneous resources in terms of processing capabilities, transmission without any guarantee on the availability in the time of these resources and processing of data on time. This way of operating based on opportunism does not ensure the Quality of Service (QoS).

Nevertheless, all these approaches suffer a lack of interoperability between future 5G-MEC new technologies and the wireless sensor networks (WSN) actually in place.

In this article, we propose to address the problem of data processing in situations where the access to the cloud is difficult or impossible. Few examples: An underground sensors network with a WSN that communicates in the ground [3,4], in the sewers [5] or in mountain areas [6,7] where sensors implanted on cannot be easily connected to a Cloud Computing service. However, the use of 5G and local processing capabilities can provide an answer to this type of problem. However, to achieve this type of approach, it is necessary to have interoperability between the WSN and the 5G-MEC. In other words, this approach is particularly important for the so-called white areas where 3rd Generation Partnership Project (3GPP) networks are not present or where the Internet network does not exist or with a speed too low to allow data transfer to the cloud.

Our motivation in this work is to propose an interoperable architecture integrating actual and still functional wireless sensor network with the 5G-MEC new paradigms. This have been achieved to take advantage of the low latency and high speed between the MEC and the end users on one hand and to continue to use alternative transmission solutions where 5G will not be available (white areas) or particularly harsh conditions on the other hand.

The rest of this article is organized as follows: In Section 2, we summarize alternatives to avoid to use Cloud Computing and processes data close to sensors. The Section 3 explains our architectural proposition. Afterwards, the Section 4, describe the implementation of the architecture. The Section 5 presents the experimentation. Finally, in Section 6, we conclude this article and outline future works.

Our contribution is a new paradigm aiming to improve exploitation of local resources at the sensors, networks and end-users devices level for an efficient data processing. The association of 5G-MEC improve the quality of the user experience while the association between Edge-Fog Computing process data close sensors, video cameras, Unmanned Aerial Vehicles (UAVs), robots or vehicles [1]. Our contribution is a coordination service which interconnects and makes MEC and Fog-Edge Computing interoperable.

2. Related Works

Different ways of local treatment have been developed to achieve the data processing closer to the sensors or users. It was therefore necessary to distinguish between the approaches achieving processing at user level such as the Mist Computing (MC), Social Internet of Things (SIoT) and those implemented on the user side such as the Jungle Computing (JC) or Dew Computing (DC).

The **Mist Computing** (MC) is a lightweight and rudimentary form of Edge Computing (EC) [8]. This paradigm extends elastic computation, storage and networking services from the edge to absolute edge/endpoints. Nowadays, smart end devices have evolved towards more efficient processors, more memory and a better networking stacks that allow them to run analysis and applications autonomously with real-time query and NoSQL like file-system support [9]. The MC rigs the computing at sensors and actuators level and is only used in case of a communication failure between the cloud and the IoT device to reduce the device power consumption [10]. When MC is insufficient Fog/Edge Computing will be of assistance [9]. The advantage of this paradigm is that the sensor data is close to its source and avoid sharing across the network except if required unequivocally. Barik et al. [11] have proposed a prototype development of SOA-Mist a Mist-based framework to provides an efficient and effective means of sharing geospatial health data resources. The framework integrates: (1) a security integration based on SSL which ensure integrity of service; (2) a database security, which ensures the availability of data for authenticated users.

The **Dew Computing** (DC) [12] allows to further improve response times by pushing from Central Cloud to End Users (EU), computing applications, data and low-level services. Client microcomputers are used to store a part of the data locally in background and to limit access to the cloud, reduce network dependency and drastically reduce processing cost [12]. DC is the additional piece of Cloud Computing. It is mainly based on on-premises computers composed of a wide range of heterogeneous devices and various equipment from smartphone to intelligent sensors; for instance, micro-services [13]. DC is highly and effectively capable in terms of scalability and ability to perform sophisticated operations and to process numerous applications and tools. Additionally, the equipment of DC is ad hoc programmable and self-adaptive [12]. They have the qualifications to running process within another process in a distributed way without a focal communication network [12]. Applications running in the on-premises computers provide services to users and/or devices independently of cloud, but collaborating with cloud services [13]. DC can provide access web fraction without an Internet connection (WiD), storage in dew has a cloud copy (STiD), local database has a cloud backup (DBiD), software ownership and settings have a cloud copy (SiD), Software Development Kit (SDK) and projects have a cloud copy (PiD), on-premises computer settings and data have a cloud copy (IaD), and other services (DiD) [13].

The **Jungle Computing** (JC) enables available computing elements present in the vicinity of users to achieve data treatment while DC synchronizes locally user data to improve the user Quality of Experience (QoE). The aim of JC is to exploit all available resources, which are diverse in terms of CPU architecture, number of cores, amount of memory, operating cost, and performance available to process data [14]. These heterogeneous, hierarchical, and distributed computing resources are, for example: Desktop Grids, Grids, isolated machines, mobile devices, clusters, and the cloud, but also specialized architectures such as GPUs and FPGAs [15]. Nowadays, the only usable Jungle Computing platform is the Ibis/Constellation [16,17]. Ibis is a high-performance distributed programming system written in Java and composed of a distributed deployment system allowing us to deploy an application in the Jungle and a high-performance programming system allowing us to write an application especially designed to run in the Jungle. While Constellation is a lightweight software platform designed for distributed, heterogeneous and hierarchical computing environments in which each application consists of multiple distinct, loosely coupled activities that communicate using events. Each activity represents distinct action, targeting small and homogeneous environment [16]. Zarrin et al. [18] have developed under the framework Service-oriented Operating System (S[o]OS), a Hybrid Adaptive Resource Discovery for Jungle Computing (HARD), an efficient and highly scalable resource-discovery approach applicable to large heterogeneous and highly dynamic distributed environment. HARD is self-adaptable and self-configurable to processing resources in the system.

The **Social Internet of Things** (SIoT) is a paradigm inspired by social networks, proposed by Atozi et al. [19], in which the privacy and protection technologies are used to enhance the security of the IoT [20]. Moreover, SIoT performs an effective discovery of things and services, services composition and improve the scalability [20,21]. The SIoT combines IoT and social networks. In this paradigm, each object establishes social relationships with other objects individually while respecting the heuristics set by the owner of the object [21]. Objects can interact following four basic relationship type: parent–child, co-location/co-work based, object ownership, and social object. These relations are used to discover and provide on-demand services. SIoT also includes service composition and trust management [22]. It is imperative for researchers to correctly identify from the outset where the data processing will take place because this has an impact on the choice of nodes, the amount of data to transfer and by consequence on communication protocols to use, but also on battery autonomy of devices. Kosmatos et al. [23] have proposed a unified architectural model for the IoT, which integrates Radio Frequency Identification (RFID) tags and smart things by using of social features of them. Ortiz et al. [24] have proposed an architecture combining humans, devices and services. Vouryras et al. [25] proposed an architecture using Virtual Entities that are the equivalent of smart things in the virtual world. Vouryras et al. [26] developed an architecture using the principles of relation model. Alam et al. [27] have proposed cyber physical architecture for the Social Internet of Vehicles (SIoV). SIoV is a vehicular instance of the SIoT, where vehicles are the key social entities in the machine to machine vehicular social networks. The architecture adopts the IoT-A reference model to design the domain models of the SIoV subsystems.

3. The Proposed Architecture

In this paper, we present a new paradigm that we name Short Supply Circuit Internet of Things (SSCIoT) illustrated in Figure 1. This new paradigm is inspired by short circuit supply in agriculture limiting intermediaries for more direct access to data. Although the deployment of 5G in association with MEC promises a major evolution in the IoT, it will nevertheless not be accessible everywhere. It is therefore important to be able to ensure interoperability between 5G-MEC, which most mobile users will benefit from using wireless protocols and low-speed protocols of Low-Power Wide-Area Network (LPWAN) that certain sensor networks will still use.

On the other hand, a collaboration 5G-MEC with the Dew Computing at user level allows to ensure service continuity in the event of a lack of network coverage. This association improves the user experience with a service that remains available even in the temporary absence of an internet connection. Moreover, the principle of OC can be diverted to be implemented between Fog Computing and MEC servers to balance workload in the function of the load of nodes and the priority or not character of the task. These tasks are also balanced inside and organized in mesh by means of Kubernetes.

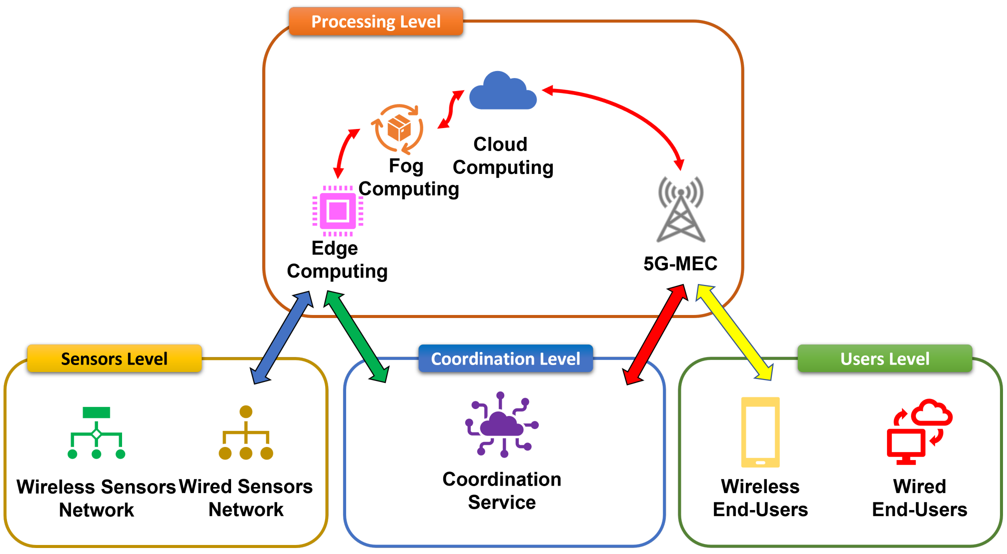

Figure 1. Overall Architecture.

3.1. Sensor Level

The Sensor Level contains various kinds of data producers categorized in wireless and wired sensors. These sensors connected to microcontrollers can be organized into networks of a few devices to tens of thousands, individually producing small amounts, from few bytes to few Kb, of data at regular intervals. Sensors can also equip vehicles, robots, drones, mobile devices such as smartphones and tablets also equipped with cameras producing large volumes of data up to several gigabytes of data per minute. These significant variations in terms of the number of sensors to be managed and the volumes of data implies that the data processing strategies are very dependent on the use cases implemented. Moreover, some data is critical and must be treated with priority over others. For example, it is easy to understand that data from fire sensors or cardiac monitoring devices must be treated as a priority. There is, therefore, a priority in the processing of data which depends on the use cases, but also according to the depreciation of the value of the data. These principles, but also of security and privacy, the desired processing time condition the level (Edge, Fog or Cloud) where the data will be treated preferentially.

3.2. Processing Level

This level groups three types of data processing (Edge, Fog and Cloud) with increasing capacities and latency from sensors to cloud. Edge Computing is achieved on microcontrollers, while Fog computing gathers network elements located between the edge of the network and the cloud where the most important capacities are available. The cloud offers possibilities to manage important amount of data but is limited in terms of latency, costly in terms of bandwidth due to data transfer. By opposition the Edge Computing offers very limited capacities of treatment close of sensors with reduced latency and a better level of privacy protection. The Fog Computing is an intermediate between Cloud Computing and Edge Computing with intermediate capacities of processing and storage allowing more important treatments than at the edge of the network [1]. MEC allows us to deploy services for mobile networks (Wi-Fi or 3GPP) [28]. It is obvious that these three levels of processing are brought to collaborate according to different modalities conditioned by the use cases such as cloud–edge, fog–edge, or cloud–fog–edge [29].

3.3. User Level

This level contains wired devices such as computers and wireless devices such as smartphones and tablets connected to Internet. In recent years, peripherals have seen their processing capacity, memory, and storage evolve significantly, allowing them to implement more complex algorithms. Wireless devices use generally WiFi or cellular networks (3G, 4G and 5G), while wired devices are connected to Internet by means of an Ethernet network. Wireless connection can be interrupted during the transmission depending on the coverage areas and the level of saturation of the networks. To overcome this problem, data caching systems have been put in place such as DC or other local caching mechanisms in order to continue to ensure the service when an Internet connection is not available.

3.4. Coordination Service

This service ensures the interoperability between MEC and LPWAN to achieve the collection and the quick treatment at fog level before their transmission to end user, critical or actuating systems. It allows the discovery of sensors and/or Edge/Fog Computing services providing data needed as input of applications hosted by MEC. The association 5G-MEC provides a connection to MEC with an ultra low latency and high throughput guaranteeing a quasi real-time processing of data.

4. Implementation

The coordination service is based on containerization that ensures the discovery of sensors, putting them in touch with end-users, coordinate processing between Fog computing/Cloud near sensors networks connected at low throughput with sensors and end-users connected with high bandwidth by means of 5G or xDSL.

Micro-services deployment can be done by using either virtual machines or containers. Virtual machines offer a better isolation thanks to the use of dedicated operating systems. On the other hand, containers are lighter because they only host the software layers necessary to run programs or services. Since containers are lighter than virtual machines, it is possible to deploy more of them with the same resource. Indeed, when the resources at network edge are limited, containerization technology is the preferred choice.

Among container orchestration systems, Kubernetes is the benchmark. Kubernetes was developed specifically for the deployment of clusters of several thousand nodes to form private, public or hybrid clouds. Micro Kubernetes (*micro k8s*) is a lightweight open source container orchestration platform that manages workloads (*worflow*) and containerized services at the edge with the ability to manage Nvidia GPU containers. Kubernetes and Micro Kubernetes are designed to work together and be deployed at the cloud and edge, respectively. Both use the notion of "pods" which are deployment units containing one or more containers. k3s is a lightweight version of k8s specifically designed for IoT as a project of the Cloud Native Computing Foundation. It takes the form of a 40 MB binary, requiring only 512 MB of memory and is adapted to run on ARMv7 and ARM64. k3s and micro k8s can run on Raspberry Pi, Nvidia Jetson Nx and Nano in theory. In practice, micro k8s has a large memory footprint for nodes with a small amount of memory, so we preferred k3s, which offers, in addition to its lightness, a sufficient level of security to implement an Edge-level cluster on constrained devices. Basically, k3s differs from k8s by replacing etcd (https://etcd.io, accessed on 24 January 2022) database with SQLite at the master node, adding a tunnel proxy that secures the connection between the master node and each worker node, and replacing the Container Network Interface (CNI) with a lighter component called “flannel”. The Figure 2 gives an overview of the interactions between software components.

Depending on the nature of the tasks to be performed, application cases and their constraints guide the choice and the number of computing nodes. For example, for image or video processing, we can be satisfied with computing nodes with many cores. If we want to use pre-trained AI algorithms, GPU-based nodes are preferred for this type of application.

A cluster with different types of nodes can address most of the needs at the Edge. In addition, if necessary, highly parallelized calculations can also be executed on the CPU nodes with an additional GPU and could be typically reserved for artificial intelligence algorithms. Similarly, artificial intelligence algorithms can also be executed on CPU-based nodes, but more slowly. The challenge is then to assign tasks according to their nature to the best suited resources according to their availability and load (CPU/GPU and memory resources used).

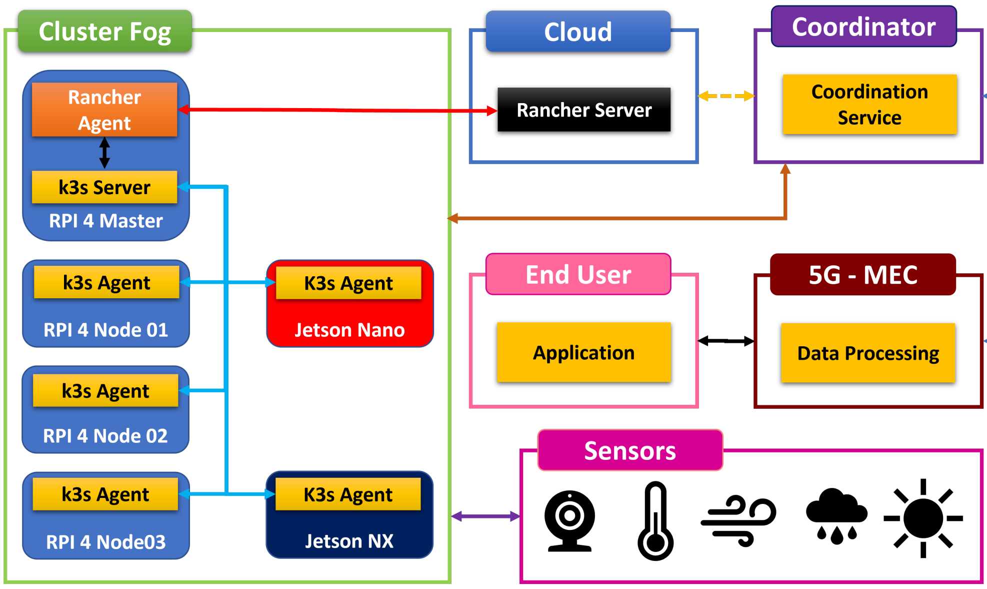

Figure 2. Interactions between components scheme.

The Rancher orchestrator is used to deploy containers at the edge level. At the cloud level, the Rancher server is deployed in a docker container on Apache Mesos. The Rancher agent deployed at the Edge cluster provides communication with the Rancher server deployed in the cloud. The Edge cluster is managed by a k3s master node, which communicates with the Rancher agent and the k3s agent installed on each k3s worker node.

Figure 3 shows the arrangement of the different components of k3s within the micro-cluster.

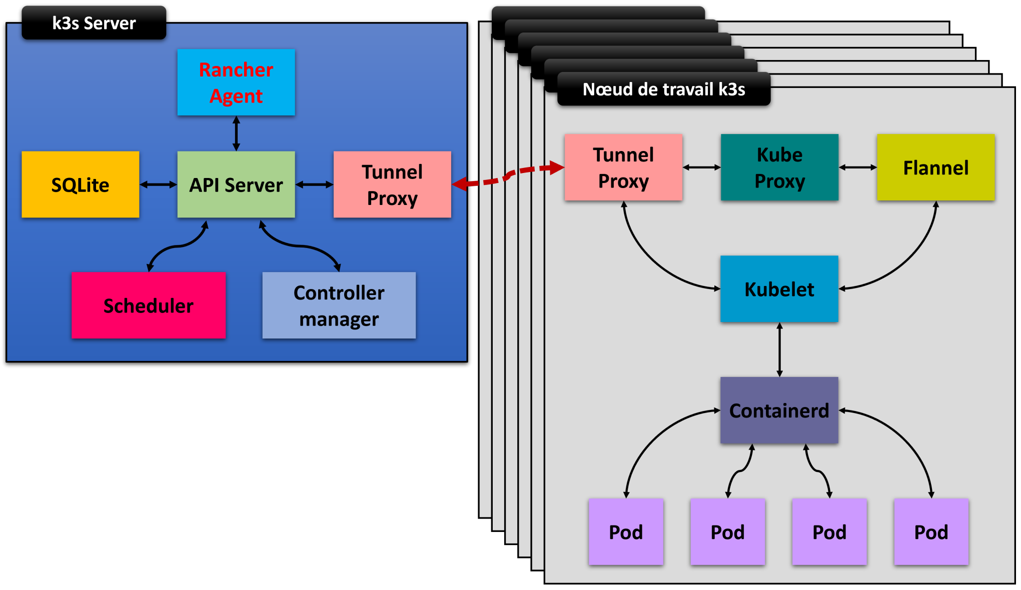

Figure 3. Components of k3s mini-cluster.

The **Master Node**, a Raspberry Pi 4, is responsible for application workload distribution, scheduling, and detecting and managing changes in the state of deployed applications. The master node is also responsible for assigning the application to a node chosen according to its needs. There are two ways used to configure our node: the first is to use the pod configuration file to describe the node that will be responsible for scheduling the application. The second way, consists of using the command line and specifying the label of the specific node.

The services installed on the master node are: (1) SQLite, for persistence and maintenance of statistical information about the various components of the k3s cluster. SQLite is used instead of etcd (https://etcd.io/, accessed on 24 January 2022) database, which is usually used by k8s, but is too memory-intensive to run in a memory constrained environment; (2) an API server exposes endpoints for all interactions with and within the k3s cluster; (3) a scheduler schedules based on application resource requirements and specific affinity constraints which application pod(s) should run on the selected working node(s) of the k3s cluster; (4) a controller manager; (5) a tunnel proxy that manages and maintains the connection tunnel between the master node and each of the worker nodes; and (6) the "Flannel" replaces the k8s Container Network Interface (CNI) and allows for the interconnection of worker nodes to each other.

The **worker nodes** mixing Raspberry Pi 4, Nvidia Jetson Nano and Nvidia Xavier NX are composed of: (1) Kubelet, an agent that runs on each worker (*Worker*) of k3s. It creates and starts an application module on the worker and monitors the health of the worker and all modules running on the master node via the API server; (2) Kube-proxy, a network proxy that is an entry point for accessing various application service endpoints and routes a request to the appropriate pods in the cluster; (3) Containerd manages the container lifecycle such as obtaining images, starting and stopping containers, etc.; and (4) the tunnel proxy maintains the connection between the master node and the worker node.

5. Experimentation

Our first use case is the analysis of cows' behaviors in field by means of a Solar WiFi camera 1080p connected to the mini-cluster previously partially described in [30] by an external Wi-Fi network. (See Figure 4).

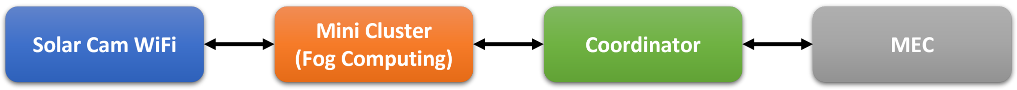

Figure 4. Global scheme of the first experiment.

The mini-cluster is composed of two Jetson NX equipped of 500 Gb SSD Samsung 980 Pro, and four Raspberry Pi 4 8GB equipped of 500 Gb Crucial MX500 SSDs interconnected by a 8 Port Gigabit Ethernet Network switch (See Figure 5). Raspberry PI 4 runs an adapted release of Ubuntu while Jetson runs Jetpack SDK 4.6 (https://developer.nvidia.com/embedded/jetpack, accessed on 25 January 2022). k3s has been used as lightweight container manager compatible with Kubernetes (k8s). It was preferred to other solutions such as micro k8s and k8s for its low memory footprint [31], its security level using centralized Role Based Access Control (RBAC), and finally for its CUDA support. One Raspberry Pi plays the role of master node, while three others are worker nodes. Jetson nodes (NX & Nano) are also worker nodes. k3s has been deployed on the mini cluster by means of Rancher 2.5 (https://rancher.com/, accessed on 25 January 2022), an open source software with zero vendor lock-in.

The cluster analyzes, on one hand, the camera video using a deep neural network, and on the other hand, collects and processes data from the environmental sensors.

The model based on a top-down approach and using YOLOv3 (https://pjreddie.com/darknet/yolo/, accessed on 23 January 2022) as object detection models to calculate a bounding box around each animal. Afterwards, the model computes for each one a digital twin based on a skeleton made of 20 key points by means of mmPose 0.21 [32], an open-source toolbox for pose estimation. The model was trained with the Animal Pose Dataset proposed by Cao et al. [33] providing animal pose annotations of 20 key points: Two eyes, Throat, Nose, Withers, Two Earbases, Tailbase, Four Elbows, Four Knees, and Four Paws for five categories: dog, cat, cow, horse, sheep. Finally, skeletons are classified animal behaviors: standing, rumination, grazing, walking, and lying [34,35]. Images are analyzed at a rate of 0.5 frame by second at Fog Level.

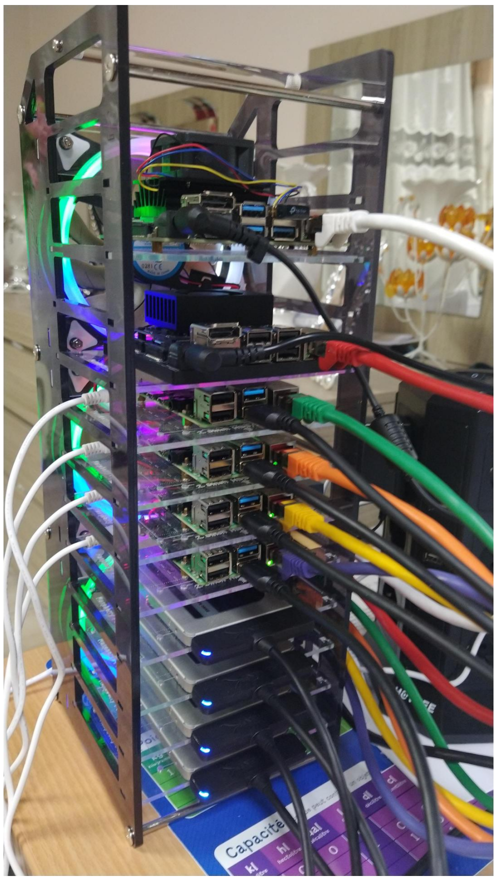

Figure 5. Experimental mini cluster.

A processed frame of a video is illustrated at Figure 6. The transmission of data coupling animal skeleton and environmental parameters is achieved by LoRa radio frequency protocol to the Coordinator (See Figure 4).

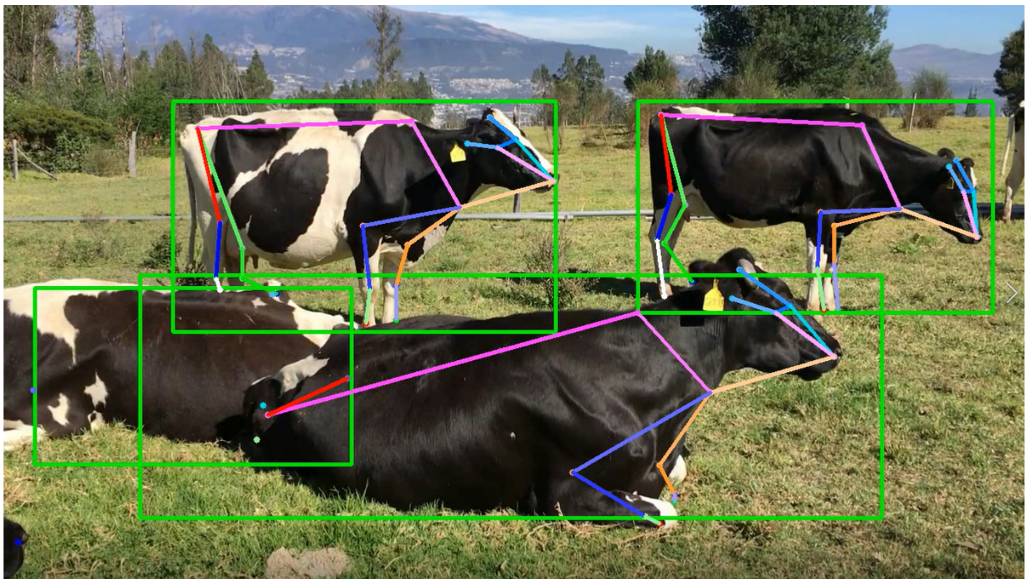

Figure 6. Processing result of a video frame.

Two kind of data packet are transmitted: The first packet of data contains the id of the cow (4 digits), and meteorological data: temperature (5 characters), relative humidity (4 characters), wind speed at 2 meter of the ground, expressed in meter per second (4 characters) transmitted once every 5 minutes. The second type of data packet contains a header with the batch index (4 digits), the id of the cow (4 digits) to witch relatives coordinates of the 10 of 20 key points (12 digits per point) of the skeleton are added. Hence, the 20 keypoints are transmitted in 2 packets each 2 s. Coordinator service merge environmental data and the two packages with data of skeleton keypoints are merged following the batch index before to be sent to MEC by 5G.

Data packets are forwarded by the coordinator to MEC by 5G where a model calculates the digital twin in the form of a skeleton as illustrated on the Figure 6 and determine behaviors from skeleton movement and deformation detection.

Our second use case is the monitoring of landslides in the mountain region of Tetouan in the North of Morocco (See Figure 7). The monitoring system previous described in [6] uses sensors to follow soil moisture and soil movements.

Figure 7. One of the monitored Landslides-Credit: Meryem Elmoulat.

The data is transmitted with LoRa protocol, but mountain harsh conditions limit the signal propagation to few hundreds meters. Then, we improved the system by using Edge Computing, but the facilities for data processing must be close to the data measurement locations which leads to high costs [36]. The SSCIoT now allows us to centralize the information from the WSNs and process one or more measurement areas at the 5G-MEC server level, allowing us to deploy the premises of our monitoring and early warning system [7]. As illustrated on the Figure 8, the coordinator is hosted on the LoRa-5G gateway which performs the dual roles of network gateway and coordination between the WSN and the 5G-MEC for data processing. The network gateway can perform both roles given the small amounts of data that are transmitted at a frequency much lower than 1 Hz.

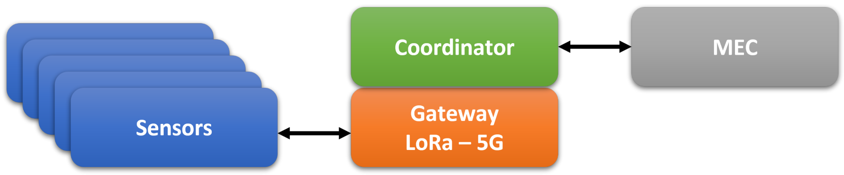

Figure 8. Global scheme of the second experiment.

This implementation of SSCIoT allows the gateway to be used for light pre-processing and hosting of the coordination service. The MEC is used for heavier processing instead of the cloud and provides ultra-low latency access to the processing results to the end user.

6. Conclusions and Perspectives

In this paper, we have proposed a new paradigm named the Short Supply Circuit Internet of Things (*SSCIoT*). This paradigm allows efficient data processing when the cloud access is difficult or impossible.

The availability of 5G is associated with the deployment of computing capacity associated with antennas that allow to offer services closer to users with ultra-low latency. The 5G allows to cover areas that were previously white, and the associated computing capacity can replace processing in the cloud. The use of sensors in harsh environments such as sewers, mountainous areas, or buried and communicating in the ground complicates the possibilities of local data processing and makes it impossible or almost impossible to transmit them to the cloud.

The coordination between MEC and Fog Computing where application processing is achieved and sensors data are treated respectively allows to process and exchange between us without or with a minimum use of cloud computing. This new paradigm that we named the SSCIoT aims to coordinate the processing of data at sensor and application level. The proposed SSCIoT paradigm avoids the use of cloud and costly fees of bandwidth on one hand, and provides a better user experience thanks to the ultra low latency and high throughput of 5G on the other hand.

We have demonstrated the feasibility of our paradigm on two concrete use cases of animal behavior analysis in field and landslides monitoring, that we have completely implemented and tested to validate the SSCIoT.

In our future work, we will implement the Osmotic Computing paradigm to balance the workload between Fog Computing and Mobile Edge Computing, we will finely measure the energy consumption and compare it with other paradigms, we will discuss the security issues.

Author Contributions: Conceptualization, O.D.; methodology, O.D.; resources, A.G.; writing—original draft preparation, O.D.; writing—review and editing, S.M.; visualization, A.G.; supervision, S.M.; project administration, A.G.; funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially funded by Infortech and Numediart Institutes.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not available.

Acknowledgments: The authors would like to express their gratitude to Meryem Elmoulat accepting to edit the writing of this paper.

Conflicts of Interest: The guest editor is one of co-authors. This paper is one the papers planned by the guest editor.

Abbreviations

The following abbreviations are used in this manuscript:

3GPP	3rd Generation Partnership Project
CCC	Central Cloud Computing
CNI	Container Network Interface
DC	Dew Computing
EU	End Users
FPGA	Field Programmable Gate Array
GPU	Graphics Processing Unit

HARD	Hybrid Adaptive Resource Discovery for Jungle Computing
JC	Jungle Computing
LPWAN	Low-Power Wide-Area Network
MC	Mist Computing
MEC	Mobile Edge Computing
MVM	Micro Virtual Machine
IoT	Internet of Things
OC	Osmotic Computing
QoE	Quality of Experience
QoS	Quality of Service
RBAC	Role Based Access Control
RFID	Radio Frequency Identification
SDK	Software Development Kit
SIoT	Social Internet of Things
SIoV	Social Internet of Vehicles
SOA	Service-Oriented Architecture
SSL	Secure Socket Layer
S[o]OS	Service-oriented Operating System
SSCIoT	Short Supply Circuit Internet of Things
UAV	Unmanned Aerial Vehicle
VM	Virtual Machine
WSN	Wireless Sensor Network

References

1. Debauche, O.; Trani, J.P.; Mahmoudi, S.; Manneback, P.; Bindelle, J.; Mahmoudi, S.; Lebeau, F. Data Management and Internet of Things : A Methodological Review in Smart Farming. *Internet Things* **2021**, *14*, 100378. [CrossRef]
2. Kaur, A.; Kumar, R.; Saxena, S. Osmotic Computing and Related Challenges: A Survey. In Proceedings of the 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 6–8 November 2020; pp. 378–383. [CrossRef]
3. Debauche, O.; El Moulat, M.; Mahmoudi, S.; Manneback, P.; Lebeau, F. Irrigation pivot-center connected at low cost for the reduction of crop water requirements. In Proceedings of the 2018 International Conference on Advanced Communication Technologies and Networking (CommNet), Marrakech, Morocco, 2–4 April 2018; pp. 1–9. [CrossRef]
4. Debauche, O.; Mahmoudi, S.; Elmoulat, M.; Mahmoudi, S.A.; Manneback, P.; Lebeau, F. Edge AI-IoT pivot irrigation, plant diseases, and pests identification. *Procedia Comput. Sci.* **2020**, *177*, 40–48. [CrossRef]
5. Tadrist, N.; Debauche, O.; Mahmoudi, S. Towards Low-Cost IoT and LPWAN-Based Flood Forecast and Monitoring System. *J. Ubiquitous Syst. Pervasive Netw.* 2022, *in press*.
6. El Moulat, M.; Debauche, O.; Mahmoudi, S.; Brahim, L.A.; Manneback, P.; Lebeau, F. Monitoring system using internet of things for potential landslides. *Procedia Comput. Sci.* **2018**, *134*, 26–34. [CrossRef]
7. Elmoulata, M.; Debaucheb, O.; Mahmoudib, S.; Mahmoudib, S.A.; Guttadauriab, A.; Mannebackb, P.; Lebeaud, F. Towards Landslides Early Warning System With Fog-Edge Computing And Artificial Intelligence. *J. Ubiquitous Syst. Pervasive Netw.* **2021**, *15*, 11–17. [CrossRef]
8. Iorga, M.; Feldman, L.; Barton, R.; Martin, M.J.; Goren, N.S.; Mahmoudi, C. Fog computing conceptual model. *NIST* **2018**. [CrossRef]
9. Yeow, K.; Gani, A.; Ahmad, R.W.; Rodrigues, J.J.; Ko, K. Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues. *IEEE Access* **2017**, *6*, 1513–1524. [CrossRef]
10. Yogi, M.K.; Chandrasekhar, K.; Kumar, G.V. Mist computing: Principles, trends and future direction. *Int. J. Comput. Sci. Eng.* **2017**, *4*, 19–21. [CrossRef]
11. Barik, R.K.; Dubey, A.C.; Tripathi, A.; Pratik, T.; Sasane, S.; Lenka, R.K.; Dubey, H.; Mankodiya, K.; Kumar, V. Mist data: Leveraging mist computing for secure and scalable architecture for smart and connected health. *Procedia Comput. Sci.* **2018**, *125*, 647–653. [CrossRef]
12. Skala, K.; Davidovic, D.; Afgan, E.; Sovic, I.; Sojat, Z. Scalable distributed computing hierarchy: Cloud, fog and dew computing. *Open J. Cloud Comput. (OJCC)* **2015**, *2*, 16–24. [CrossRef]
13. Wang, Y. Definition and categorization of dew computing. *Open J. Cloud Comput. (OJCC)* **2016**, *3*, 1–7. [CrossRef]
14. Seinstra, F.J.; Maassen, J.; Van Nieuwpoort, R.V.; Drost, N.; Van Kessel, T.; Van Werkhoven, B.; Urbani, J.; Jacobs, C.; Kielmann, T.; Bal, H.E. Jungle computing: Distributed supercomputing beyond clusters, grids, and clouds. In *Grids, Clouds and Virtualization*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 167–197. [CrossRef]
15. Tychalas, D.; Karatza, H. High performance system based on Cloud and beyond: Jungle Computing. *J. Comput. Sci.* **2017**, *22*, 131–147. [CrossRef]

16. Hajibaba, M.; Gorgin, S. A review on modern distributed computing paradigms: Cloud computing, jungle computing and fog computing. *J. Comput. Inf. Technol.* **2014**, *22*, 69–84. [CrossRef]
17. Maassen, J.; Drost, N.; Bal, H.E.; Seinstra, F.J. Towards jungle computing with Ibis/Constellation. In *Proceedings of the 2011 Workshop on Dynamic Distributed Data-Intensive Applications, Programming Abstractions, and Systems*; Association for Computing Machinery: New York, NY, USA, 2011; pp. 7–18. [CrossRef]
18. Zarrin, J.; Aguiar, R.L.; Barraca, J.P. HARD: Hybrid adaptive resource discovery for jungle computing. *J. Netw. Comput. Appl.* **2017**, *90*, 42–73. [CrossRef]
19. Atzori, L.; Iera, A.; Morabito, G.; Nitti, M. The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Comput. Netw.* **2012**, *56*, 3594–3608. [CrossRef]
20. Li, S.; Da Xu, L.; Zhao, S. The internet of things: A survey. *Inf. Syst. Front.* **2015**, *17*, 243–259. [CrossRef]
21. Roopa, M.; Pattar, S.; Buyya, R.; Venugopal, K.R.; Iyengar, S.; Patnaik, L. Social Internet of Things (SIoT): Foundations, thrust areas, systematic review and future directions. *Comput. Commun.* **2019**, *139*, 32–57. [CrossRef]
22. Afzal, B.; Umair, M.; Shah, G.A.; Ahmed, E. Enabling IoT platforms for social IoT applications: Vision, feature mapping, and challenges. *Future Gener. Comput. Syst.* **2019**, *92*, 718–731. [CrossRef]
23. Evangelos, A.K.; Nikolaos D.T.; Anthony, C.B. Integrating RFIDs and smart objects into a Unified Internet of Things architecture. *Adv. Internet Things* **2011**, *2011*, 4696. [CrossRef]
24. Ortiz, A.M.; Hussein, D.; Park, S.; Han, S.N.; Crespi, N. The cluster between internet of things and social networks: Review and research challenges. *IEEE Internet Things J.* **2014**, *1*, 206–215. [CrossRef]
25. Voutyras, O.; Bourelos, P.; Gogouvitis, S.; Kyriazis, D.; Varvarigou, T. Social monitoring and social analysis in internet of things virtual networks. In Proceedings of the 2015 18th International Conference on Intelligence in Next Generation Networks, Paris, France, 17–19 February 2015; pp. 244–251. [CrossRef]
26. Voutyras, O.; Bourelos, P.; Kyriazis, D.; Varvarigou, T. An architecture supporting knowledge flow in Social Internet of Things systems. In Proceedings of the 2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Larnaca, Cyprus, 8–10 October 2014; pp. 100–105. [CrossRef]
27. Alam, K.M.; Saini, M.; El Saddik, A. Toward social internet of vehicles: Concept, architecture, and applications. *IEEE Access* **2015**, *3*, 343–357. [CrossRef]
28. Wang, X.; Han, Y.; Leung, V.C.; Niyato, D.; Yan, X.; Chen, X. *Edge AI: Convergence of Edge Computing and Artificial Intelligence*; Springer Nature: Berlin/Heidelberg, Germany, 2020. [CrossRef]
29. Debauche, O.; Mahmoudi, S.; Manneback, P.; Lebeau, F. Cloud and Distributed Architectures for Data Management in Agriculture 4.0: Review and Future Trends. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**. [CrossRef]
30. Debauche, O.; Mahmoudi, S.; Mahmoudi, S.A.; Manneback, P.; Lebeau, F. A new edge architecture for ai-iot services deployment. *Procedia Comput. Sci.* **2020**, *175*, 10–19. [CrossRef]
31. Fathoni, H.; Yang, C.T.; Chang, C.H.; Huang, C.Y. Performance Comparison of Lightweight Kubernetes in Edge Devices. In *Pervasive Systems, Algorithms and Networks*; Esposito, C., Hong, J., Choo, K.K.R., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 304–309. [CrossRef]
32. Contributors, M. OpenMMLab Pose Estimation Toolbox and Benchmark. 2020. Available online: https://github.com/open-mmlab/mmpose (accessed on 1 February 2022).
33. Cao, J.; Tang, H.; Fang, H.S.; Shen, X.; Lu, C.; Tai, Y.W. Cross-Domain Adaptation for Animal Pose Estimation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019.
34. Debauche, O.; Mahmoudi, S.; Mahmoudi, S.A.; Manneback, P.; Bindelle, J.; Lebeau, F. Edge Computing for Cattle Behavior Analysis. In Proceedings of the 2020 Second International Conference on Embedded Distributed Systems (EDiS), Oran, Algeria, 3 November 2020; pp. 52–57. [CrossRef]
35. Debauche, O.; Elmoulat, M.; Mahmoudi, S.; Bindelle, J.; Lebeau, F. Farm animals' behaviors and welfare analysis with AI algorithms: A review. *Rev. D'Intelligence Artif.* **2021**, *35*, 243–253. [CrossRef]
36. Elmoulat, M.; Debauche, O.; Mahmoudi, S.; Mahmoudi, S.A.; Manneback, P.; Lebeau, F. Edge computing and artificial intelligence for landslides monitoring. *Procedia Comput. Sci.* **2020**, *177*, 480–487. [CrossRef]