2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation

Software-Defined Edge Computing (SDEC): Principles, Open System Architecture and Challenges

Pengfei Hu, Wai Chen
China Mobile Research Institute
Beijing, China
hupengfei@chinamobile.com, wai.w.chen@gmail.com

***Abstract*—Edge computing is a bridge for realizing the convergence between physical space and cyber space. Large numbers of physical objects produce a huge amount of data that needs to be efficiently processed in the edge side. This situation urgently requires novel ideas and framework in the design and management of edge computing to improve and enhance its performance. In this paper, we propose an approach and principles of Software-Defined Edge Computing (SDEC) from the perspective of cyber-physical mapping, where the ultimate goal is to achieve a highly automatic and autonomous edge computing system. And SDEC can also help realize flexible management and intelligent collaboration among various edge hardware resources and services by way of software. To this end, we design a SDEC-based open edge computing system architecture which decouples upper-level applications from the underlying physical edge resources and builds dynamically reconfigurable smart edge services. This approach can share, reuse and recombine edge resources and services so that overall service capability of the edge side is improved. Finally, we outline several challenges which are worthy of in-depth study and research.**

***Keywords*-Edge computing, Cyber-physical mapping, Software-Defined Edge Computing (SDEC), Software-Defined Edge Device (SDED), Software-Defined Edge Storage (SDESto), Software-Defined Edge Computing Resource (SDECR), Software-Defined Edge Service (SDESer)**

I. Introduction

The rapid development of Cyber-Physical System (CPS) and Internet of Things (IoT) makes the collaborative interaction and cross-border integration among people, machines and things more frequently. Physical space for human survival is developing toward the direction of networking, informatization and intelligentization. This also makes physical space full of data, information, knowledge and intelligent applications. More and more data and information need to be analyzed, processed and stored in the edge side to meet the demands for low latency and save bandwidth [1], [2]. Edge computing becomes a bridge of convergence between physical space and cyber space.

Edge computing moves the capacities of computing, communication, storage, services and control to the system edge in close proximity to the end devices and users. So that it can provide smart edge services with low latency, high reliability, high bandwidth efficient and high security [3]. Different from highly centralized cloud computing, edge computing uses the distributed computing and storage resources to improve the quality of service (QoS) [4].

In the current IoT and edge computing, more and more hardware devices (e.g., sensors, actuators, smart mobile phone, smart gateway, and computer terminals) are connected into cyber space. Generally, these edge devices are heterogeneous, which is one of the main challenges to realize the unified management and information sharing. Moreover, the sharing and collaboration of edge resources is critical for improving the utilization of resources and overall edge service capability, and reducing energy consumption [5]. In addition, the current IoT applications are mostly developed in the vertical industry to establish their own relatively independent and closed application system. This prevents the interoperability and cross-domain integration among edge resources.

The idea of software definition properly addresses the current limitations faced by edge computing. The core and essence of software definition are to separate the software and hardware by virtualizing, abstracting and pooling the hardware infrastructures to realize the on-demand segmentation and recombining of various resources. It manages and schedules these virtualized resources by control software to flexibly customize and expand the whole system function [6]. Like computer operating system (OS), it can be regarded as the software-definition for computer hardware. The software definition paradigm implements the interconnection and intercommunication of heterogeneous edge device resources in physical space and cyber space [7].

In this paper, we propose the idea of software-defined edge computing (SDEC) and the SDEC-based open edge computing system architecture in order to synergistically manage the heterogeneous edge resources. For different categories of edge resources and services, SDEC is subdivided into Software-Defined Edge Device (SDED), Software-Defined Edge Storage (SDESto), Software-Defined Edge Computing Resource (SDECR), and Software-Defined Edge Service (SDESer). They respectively abstracts and virtualizes the underlying edge device resources, edge storage

978-1-7281-4034-6/19/$31.00 ©2019 IEEE
DOI 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00047

resources, edge computing resources and edge services into cyber models, as well as forms corresponding resource pools. So that the abstraction, virtualization and pooling for edge resources and services are realized. Meanwhile, the control and management for edge resources are implemented by a logically centralized controller. In this way, the control and management functions are separated from hardware devices, and running it as software instead. This enables the customized control and simplifies the development and deployment of new edge services.

The SDEC-based open edge computing system architecture is designed by adding SDEC platform into the traditional edge computing architecture. The two main functions of this architecture are to downward manage all kinds of physical resources through the virtualization modeling technology, and upward provide various edge resources and public services through programmable functions of management. The SDEC-based scheme focuses on implementing a dynamic and reconfigurable smart edge computing system. The edge resources and services can be flexibly shared, reused and recombined to improve utilization and overall service capability of edge side.

The main contributions of this article are shown as follows:

- From the perspective of cyber-physical mapping, the principles of SDEC are proposed to realize the software definition of edge device resources, edge storage resources, edge computing resources and edge services, and decouple the software control and management functions from underlying physical devices.
- An SDEC-based open edge computing system architecture is designed to provide an intelligent, automated and autonomous edge framework and environment, and achieve the sharing and collaboration of edge resources and services.

The remainder of this paper is organized as follows. Section 2 reviews the related works on SDEC. Section 3 proposes the principles of SDEC. Section 4 proposes the SDEC-based open edge computing system architecture. Section 5 presents the challenges in the future research and application of SDEC. Section 6 draws a conclusion.

II. Related Works

In order to realize edge resources sharing, some researchers have adopted the virtualization technologies which are applied to cloud computing in edge computing, e.g., hypervisor, container, and so on. Wang et al. [8] have proposed a lightweight edge computing platform based on virtualization technology. The container is used to accomplish the virtualization of computing and storage resources. The container engine abstracts the computing and storage resources of physical infrastructure to implement resources sharing. This scheme abstract and virtualize edge resources from the container level, where resources are not modeled and described. It does not achieve the complete decoupling between software and hardware.

Network Functions Virtualization (NFV) technology has also been considered to realize the devices sharing. For the sharing of home multimedia devices, Faraci et al. [9] have proposed the concept of virtual presence as add-on of the Personal Network (PN) concept. It can export a real hardware or software resource, which is physically or virtually located in the same PN of a given user, towards another PN. And the architecture which allows to instantiate a virtual presence in a PN is proposed, so that it can be used as present locally in the visited PN.

Moreover, inspired by software-defined networking (S-DN), the software definition technologies have emerged to be used to manage various resources [10]. Buyya et al. [11] proposed an architecture of software-defined cloud computing (SDC) on data centers with emphasis on mobile cloud applications. It is used as an approach for automating the process of optimal cloud configuration by extending virtualization concept to various resources of data center. This provides a reference for edge computing although they have their own features.

Hu et al. [12] have proposed the concept and system model of software-defined device (SDD) for underlying sensing devices and actuating devices in IoT. Meanwhile, an open IoT system architecture based on SDD has been presented to realize the unified management, sharing and reusing of underlying devices. However, this paradigm only applies to underlying IoT devices, not to computing and storage resources.

In this work, we propose a software-defined edge computing (SDEC) paradigm from the perspective of cyber-physical mapping to realize the software definition of various heterogeneous edge resources and edge services, and decouple the software control and management functions from the underlying physical devices. In addition, the SDEC-based open edge computing system architecture will also be designed to provide an intelligent, automated and autonomous edge framework and environment, and achieve the sharing and collaboration of edge resources and services.

III. Principles of SDEC

A. Overview of SDEC

The concept of software definition is an emerging paradigm that has appeared to hide the management complexity of hardware infrastructures in traditional system architecture. The control plane and data plane are decoupled and separated. The management, controlling and scheduling functions of hardware resources are implemented by software. So this mechanism can flexibly expand and customize system functions [13].

Inspired by the software definition paradigm, we propose the idea of SDEC from the perspective of cyber-physical

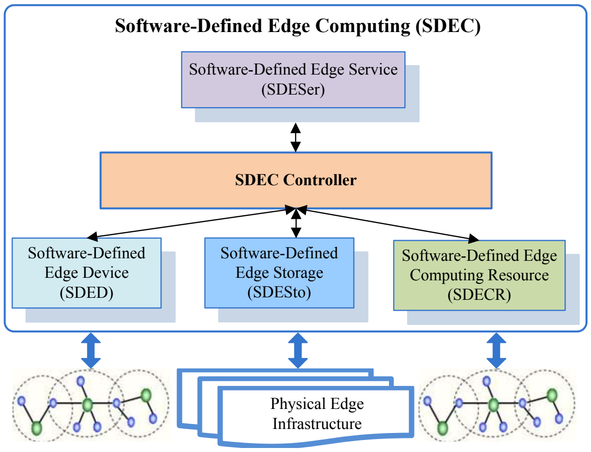

Figure 1. Relational view of SDEC.

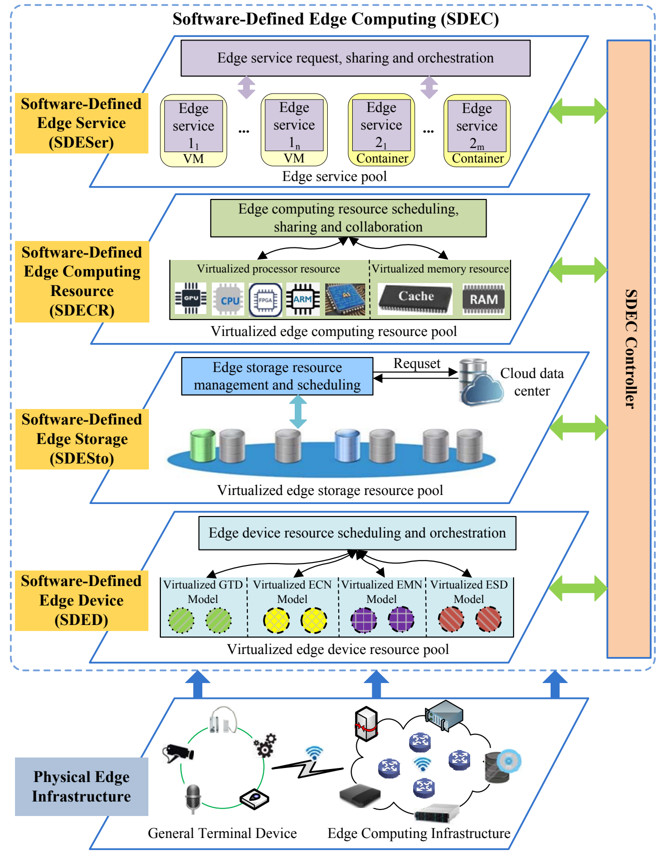

Figure 2. Detailed overview of SDEC.

mapping in this paper. It extends the concepts of software definition and virtualization modeling to the various resources (including terminal device resources, storage resources, and computing resources) and services in edge side. It includes not only software definition for edge resources, but also software definition for edge services. These physical resources and services are virtualized and abstracted to cyber models by cyber-physical mapping technologies, and pooled to form the resource pool and service sets. The SDEC controller schedules, manages and controls abstracted resources and services to implement various edge applications. In this way, the modularization of underlying edge resources and the customization of edge application services can be realized flexibly.

A relational view of SDEC is shown in Fig. 1. SDEC mainly consists of the following five core parts: SDED, SDESto, SDECR, SDEC controller and SDESer. The SDED, SDESto and SDECR are the abstraction and software definition for physical edge infrastructure. The SDEC controller is the management and control center. The SDESer is the software definition for edge service functions deployed on heterogeneous edge infrastructure. It depends on the resources and environments provided by SDED, SDESto, SDECR and SDEC controller.

The detailed overview of SDEC is shown in Fig. 2. The SDED, SDESto, SDECR, SDEC controller and SDESer cooperate together and realize software definition for edge computing system from the aspects of underlying edge hardwares, edge storage resources, edge computing resources, and edge application services, respectively. Leveraging the virtualization, modeling and control technologies, the edge resources and services can be flexibly managed, scheduled, shared and collaborated so that various smart edge applications are implemented.

The ultimate goal of SDEC is to achieve a highly automatic and autonomous edge computing system. Specially, this is mainly represented from the following aspects:

- Automatic collaboration among edge devices;
- Automatic update of device models and system models;
- Automatic matching between application requirements and edge resources;
- Automatic interaction and collaboration between physical space and cyber space.

B. Software-Defined Edge Device (SDED)

The development of current edge computing and IoT is industry-specific and domain-dependent, and fragmented in terms of standards and technologies. This results in the piecemeal "information island" and "application island". For different industries and domains, the interoperability, sharing and cross-domain composability of devices have become a challenge [12]. Moreover, due to the heterogeneity of different edge devices, it is difficult to directly share the capacities and data of devices, and uniformly schedule them [14]. Therefore, the efficient management, sharing and collaboration of heterogeneous edge devices play an important role for realizing smart edge services and applications.

According to the difference of function and hierarchic location in edge computing, edge devices are classified

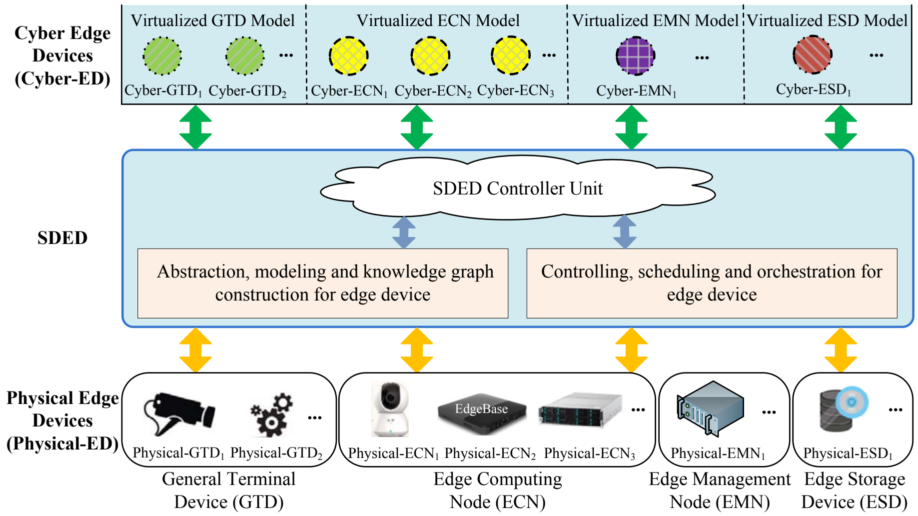

Figure 3. Software definition processes of SDED.

into four categories: general terminal device (GTD), edge computing node (ECN), edge management node (EMN) and edge storage device (ESD). Their functions are introduced as follows.

- GTD: It includes sensors and actuators. The main function is to sense physical objects and environment, and perform computing results.
- ECN: It includes some smart terminal devices, smart gateways, computing boxes, customized computing devices, and more. The main function is to provide edge computing capabilities and perform edge computing tasks.
- EMN: It includes edge servers or custom servers with slightly more computing and management capabilities. The main function is to host and run the SDEC platform, manage and coordinate ECNs, GTDs and ESD. It is also responsible for interacting with other EMNs or cloud data center. Furthermore, it can also be used as a computing node to perform some computing tasks.
- ESD: It includes dedicated and general storage devices. The main function is to store the data produced by edge devices, intermediate results, and processing results in edge side.

SDED is the software virtualization for edge hardwares. Its software definition processes is shown in Fig. 3. In this paradigm, it mainly consists of two aspects. On the one hand, underlying physical edge devices (Physical-ED) are abstracted and modeled into their virtualized cyber edge device (Cyber-ED) models by semantic modeling and knowledge graph technologies. Cyber-ED is the unique and comprehensive digital clone (or digital twins) of Physical-ED. The relationship between Cyber-ED and Physical-ED is a one-to-one correspondence. Physical-ED and Cyber-ED can synchronize each other's states by, for example, using Message Queuing Telemetry Transport (MQTT) technology to ensure their consistency. Based on these cyber models, a virtualized edge device resource pool is formed. On the other hand, application can obtain and handle Cyber-ED models by SDED controller unit to get the latest status of Physical-EDs or send control commands to Physical-EDs. In this process, SDED controller unit analyzes application requirements, and automatically converts them to the corresponding device operations by combining with Cyber-ED models. And controlling, scheduling and orchestration strategies are used to assign appropriate edge device resources for edge tasks. In this way, cyber space can control and act on Physical-EDs and physical space by Cyber-ED models to realize various smart edge applications.

Through this mechanism, hardware and software are completely decoupled. The applications programmers do not need to worry about the details of underlying edge devices, and can make full use of the programmability of hardware. In addition, these scattered edge devices can be uniformly managed, scheduled, shared, and dynamically reconfigured. Their service capacities can be fully utilized, and the overall edge service performance can also be greatly improved.

C. Software-Defined Edge Storage (SDESto)

In edge computing, the storage capacity is relatively weak compared with large-scale cloud data centers. And edge computing has strong real-time demands in data storage and processing. The utilization and scheduling of edge storage resources in an efficient manner is the key to improve overall edge storage capacity. Moreover, in the current diversified IoT environment, diverse data access types are required, for example, file, key-value, block, object, etc. A flexible storage resource management approach needs to be designed to meet the demands of diversified storage services [15].

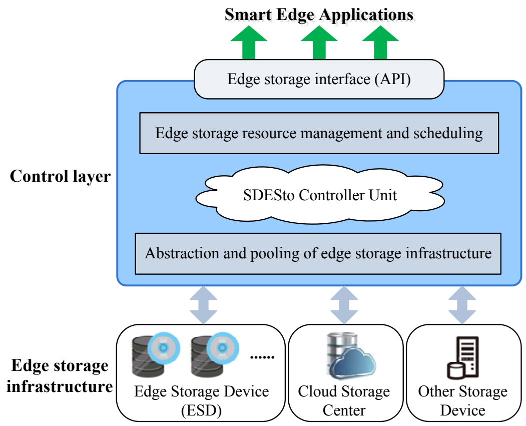

Figure 4. Architecture of SDESto.

SDESto is proposed to address the control and management challenges in edge storage. It manages and schedules edge storage resources by separating control layer from edge storage infrastructure. By central SDESto control unit, heterogeneous edge storage resources are managed uniformly rather than installing the control software on each storage resource.

Fig. 4 illustrates the architecture of SDESto. It consists of three layers: edge storage infrastructure layer, control layer, and smart edge application layer. The edge storage infrastructure layer includes various underlying ESDs, cloud storage center and other storage devices. In particular, as a complement to edge storage, cloud storage center is used to store the data that requires long-term storage and does not require real-time processing, or large volume data. The control layer is the bridge where smart edge applications interact with edge storage infrastructure. The SDESto control unit is considered to be the most critical element in SDESto mechanism, which is responsible for managing and controlling edge storage resources to enable the system automation. The edge storage infrastructure is abstracted and pooled into virtualized edge storage resource pool through sematic modeling technologies. By edge storage resource management and scheduling methods, appropriate resources can be matched and discovered to meet application demands. And storage services become dynamically reconfigurable. The smart edge application layer holds different applications and allows the end user to interact with edge storage devices by some APIs.

SDESto paradigm reduces the management complexity for edge storage resources. The resources allocation is realized dynamically on-demand. The flexible edge storage resources management manner improves the availability, reliability and efficiency of edge system, and meets the demands for diversified storage services.

D. Software-Defined Edge Computing Resource (SDECR)

There are a lot of edge computing devices or nodes designed and produced by different manufacturers in edge scenarios. According to the classification for edge devices in the above section, ECN is the main computing capabilities provider, and some capabilities of EMN are also used as computing. They have the characteristics of strong heterogeneity, different performance parameters and differentiated computing power. In addition, edge computing devices do not need to run intensive computing tasks anytime and anywhere generally, so idle computing resources should be shared to other applications through the network to improve utilization.

SDECR is proposed to solve the above problems by software definition mechanism. According to different processors, edge computing devices can be divided into CPU, GPU, FPGA, ARM, AI chip, etc. These computing devices are separately abstracted into different categories of edge computing resources. In above SDED paradigm, the ECN and EMN have been virtualized and abstracted into corresponding models. These models enable cyber space to fully understand edge computing resources. So edge applications can discover and find suitable resources to meet their computing demands.

Based on edge computing resource models, the SDECR controller unit can manage, control and schedule edge computing resources flexibly to meet the demands of computing tasks. The scheduling algorithms is designed and developed to ensure that computing resources are properly allocated to each smart edge application. Moreover, sharing and collaboration mechanism is also deployed to improve resource utilization and overall computing capability. By managing and sharing idle computing resources, the purpose of effective processing multi-source information can be achieved.

SDECR paradigm is envisioned to flexibly schedule and combine heterogeneous edge computing resources. The mechanism can dynamically assign suitable edge computing resources for computing tasks according to the computing power and free/busy states. This can prevent the computing tasks to be migrated to a computing node that is overloaded with system tasks, which will avoid impacting the overall performance of system.

E. SDEC Controller

For software definition mechanism, the centralized control unit is usually the core of overall system. As the central pivot of SDEC paradigm, SDEC controller mainly implements the management and control of abstracted edge resources. It serves as a sort of edge OS for underlying edge hardware. By taking the management and control functions off of the hardware and running it as software instead, the SDEC controller enables intelligent and automatic management for edge resources, and ensures the flexibility of overall edge system.

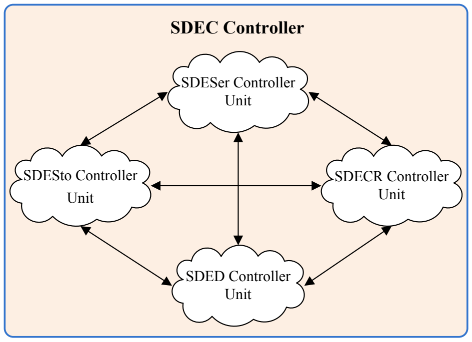

Figure 5. Composition of SDEC controller.

As is shown in Fig. 5, SDEC controller consists of four subunits, namely SDED controller unit, SDESto controller unit, SDECR controller unit, and SDESer controller unit. They are responsible for the management and control of SDED, SDESto, SDECR and SDESer, respectively. These controller units interconnect and interact with each other to implement the allocation and collaboration among various edge resources. The administrator can easily reconfigure these edge resources through the standard programmable interfaces.

The main functions of SDEC controller are as follows.

- Managing and controlling the access and exit, virtualized modeling, models update, resource allocation, sharing and collaboration;
- Obtaining the resource status information and providing the resource monitoring function;
- Maintaining a high level of load balance, and providing a fast responses for multi requests;
- Transforming the requests of application layer into the demands for edge resources;
- Providing the appropriate underlying edge resources for upper-level edge applications.

F. Software-Defined Edge Service (SDESer)

The edge device, storage and computing resources collaborate with each other closely, so that various edge services can be implemented to support smart edge applications. This includes not only edge services developed by the platform itself, but also edge services from other platforms, manufacturers, service providers and the third parties. In the traditional mode, edge services are tightly coupled with underlying dedicated hardwares where development environment, interface and configuration are different. Therefore, it is difficult for different edge services to share and cooperate with each other. For effective and seamless compatibility with various edge services, we propose the idea of SDESer in SDEC system.

SDESer is the software definition for edge service functions deployed in heterogeneous edge infrastructures. Different from SDED, SDECR and SDESto which are software definition for edge hardwares and infrastructures, SDESer is the software definition for functions. It adopts virtualization modeling technology to form edge service pool. This eliminates the dependency between edge service functions and hardwares. Then standardized execution environment and management interfaces are built for virtual edge service functions. In this way, multiple virtual edge service functions can share physical hardware. Finally, these virtual function modules are provided to upper-level applications in a uniform manner. In this paradigm, various edge service functions can be flexibly combined and separated according to different demands. It can shield the heterogeneity of underlying dedicated devices from the upper-level applications.

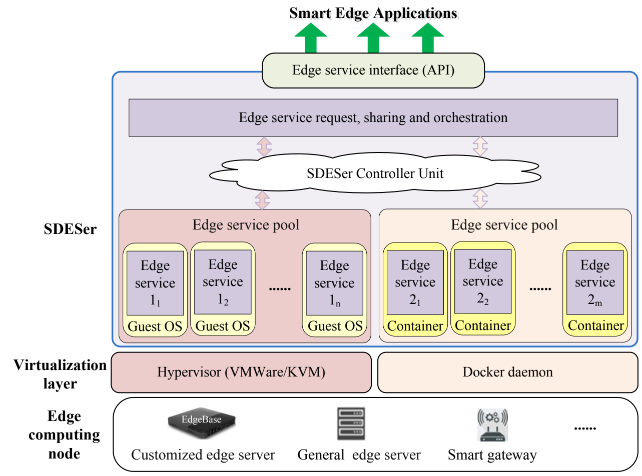

Figure 6. Architecture of SDESer.

The architecture of SDESer is shown in Fig. 6. SDESer is deployed on edge infrastructures, for example, customized edge server, general edge server, smart gateway, etc. These infrastructures run their own host OS, including Linux, Windows, Mac OS, Android, iOS, etc. Generally, virtualization technologies are adopted to build a virtualization layer on host infrastructure, so that heterogeneous applications can be developed and deployed uniformly. Virtual machine (VM) and docker are two major virtualization technologies. For VM, hypervisor is the management system, including VMWare, KVM, etc. Multiple different guest OSs need to be installed and deployed on hypervisor to run multiple edge services separately. According to the different requirements, multiple development environments and dependencies need to be installed on every guest OS. A guest OS can load and deploy only one edge application. For docker, the docker daemon replaces the hypervisor. It is a background process running on host OS and responsible for managing these docker containers. The source codes of edge services and its dependencies are packaged in docker image. Different applications require different docker images and run in different docker containers, which are isolated from each

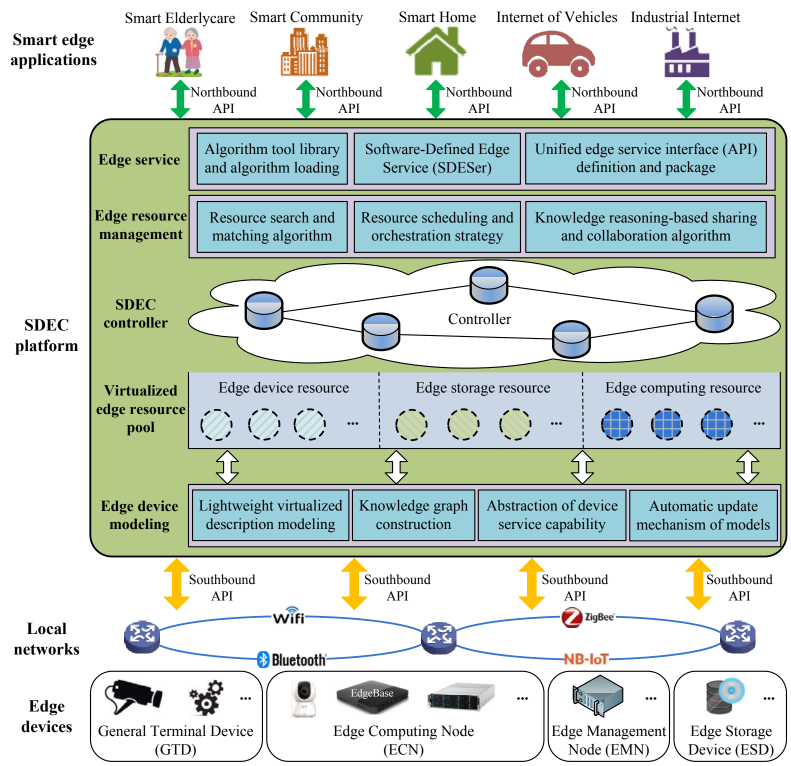

Figure 7. SDEC-based open edge computing system architecture.

other.

Based on edge infrastructures and virtualization environments, SDESer further abstracts and pools these edge services to form the edge service pool. SDESer control unit handles edge service request, and executes sharing and orchestration strategies. So various edge services can be provided to these authorized smart edge applications by way of API.

IV. SDEC-based Open Edge Computing System Architecture

Based on above SDEC paradigm, we propose a SDEC-based open edge computing system architecture shown in Fig. 7. Different from traditional edge computing architecture, the software-defined platform is added to achieve a highly automatic and autonomous edge computing system.

The edge system architecture contains four layers: edge devices, local network, SDEC platform and smart edge applications. Their detailed introductions are as follows.

1) Edge devices

This layer includes various edge devices, i.e., GTD, ECN, EMN and ESD. In particular, EdgeBase is a customized ECN developed by ourselves. It can be also used as an EMN to load the SDEC platform. In our future works, we will develop and deploy the proposed SDEC platform and smart edge computing application system mainly based on EdgeBase. These edge devices are interconnected and intercommunicated by local network to form the infrastructure of edge computing.

2) Local networks

Different from core networks, local network are closer to edge devices and end users. It is responsible for interconnecting various edge devices and connecting them into SDEC platform. Some communication protocols are in common use, for example, WiFi, BlueTooth, ZigBee, NB-IoT, NFC, etc.

3) SDEC platform

SDEC platform is the core of the proposed edge computing architecture. It is the concrete implementation of SD-ED, SDESto, SDECR and SDESer. This platform manages heterogeneous edge resources downward through the southbound API, and processes a large amount of heterogeneous data and multi-purpose application loads upward through the northbound API. It is responsible for the deployment, scheduling and migration of complex computing tasks on the edge computing nodes, so as to ensure the reliability of computing tasks and maximize the utilization of resources. Mainly, SDEC platform consists of the following five parts.

- **Edge device modeling**

Edge device modeling is the basis of software definition mechanism. For various Physical-EDs, their properties information and domain specific knowledge are abstracted and described to create Cyber-ED models through lightweight description modeling and knowledge graph construction methods. Some semantic technologies and knowledge engineering, are adopted to associate these devices and construct the knowledge graph of edge devices.

The device service capability abstraction module is responsible for transforming the edge devices into service capability resources, such as sensing capability, actuating capability, computing capability, and storage capability. Edge device resources are abstracted service components which will be discovered and accessed by upper-level edge applications. For example, a custom edge computing server with storage media can be abstracted not only as computing service capability, but also as storage service capability. Moreover, the unique identifiers based on Uniform Resource Identifier (URI) are assigned to every edge device resources in order to facilitate the discovery and addressing of service capabilities.

Because of the dynamic characteristics of edge scenario, the automatic update mechanism of models needs to be deployed to keep consistency between physical device resources and models. It includes not only the changes of device properties, but also the changes of domain knowledge and rules. Some related update strategies need to be designed and developed.

- **Virtualized edge resource pool**

The virtualized edge resource pool is responsible for storing and managing the models data of various edge resources in their entire life cycles. When a physical device accesses the system, or its status changes, or it is disconnected and deleted from the system, the corresponding virtualized resource model in the pool will be updated synchronously.

- **SDEC controller**

As mentioned above, SDEC controller is the logically central control center of SDEC platform. It controls and manages the various edge resources by corresponding controller units. All the operations on these edge resources have to go through the controller. On the one hand, controller transforms the user requests of application layer into the demands for edge resources. On the other hand, controller provides the abstracted models of underlying edge hardware for upper-level edge applications. The SDEC controller implements edge resources control function by specific programmable interfaces.

- **Edge resource management**

Edge resource management is responsible for resource search and scheduling, as well as sharing and collaboration among them. The corresponding algorithms and strategies need to be researched and developed. Resource search and matching algorithm realizes resource discovery according to application demands. By resource scheduling and orchestration strategies, all edge resources are properly assigned to each task according to priority. Using knowledge reasoning technologies, some knowledge and rules are defined to realize the edge resources sharing and collaboration. These algorithms and strategies enable efficient utilization of resources and intelligent collaboration.

- **Edge service**

The purpose of edge resource modeling and management is to provide flexible and efficient edge services. Some generic processing algorithms are packaged into algorithm tool library. According to different application demands, these algorithms can be loaded flexibly. Moreover, SDESer mechanism is deployed to realize the decoupling between edge services and heterogeneous edge resources. These smart edge services are defined and packaged into unified service interface (API) to support various upper-level smart edge applications.

4) Smart edge applications

This layer is oriented to the edge computing users to provide various smart edge applications, including smart elderlycare, smart community, smart home, etc. The demands of edge application scenarios are analyzed and understood by SDEC platform. And corresponding edge resources and services are matched and called to implement smart edge applications.

V. CHALLENGES

There are still some challenges and open issues for the implementation and further research of SDEC. In this section, we will highlight some of these challenges as follows.

1) High automation and autonomy of SDEC-based system

In edge environment, devices are usually unattended. Some operations, including software upgrading and update, device operation status monitoring, and collaboration among devices, will cost a large amount of manpower and money. Therefore, designing a highly automatic and autonomic edge computing system architecture, rather than arbitrary intervention, is vital to the development of smart edge computing ecology. The non-arbitrary operations will greatly improve the level of intelligence for the SDEC-based system.

2) Light-weight modeling, algorithms and platform

Compared with cloud computing, the resources and capacities of computing and storage in edge scenarios are limited. It is difficult to support the runs of large-scale software. Even though the processing speed of current processor is constantly improving and the power consumption is constantly reduced, it is still insufficient to support complex tasks processing in the current situation. Furthermore, S-DEC platform consists of a series of operations, including resources modeling, pooling, controlling, managing, and service support. The platform itself consumes a lot of resources. Therefore, light-weight is an important goal and challenge for SDEC platform optimization. This includes light-weight platform design, light-weight virtualization and abstraction modeling methods, and light-weight resources management and scheduling algorithms.

3) Automatic update mechanism of software definition models

In SDEC, each edge physical device always exists in pairs with its cyber model. In order to keep consistency between them, the cyber model needs to be updated with the changes of physical device, such as location change, function upgrades, battery depletion, and so on. Similarly, the software definition models in cyber space for various edge resources also need to be updated with the changes of physical space. The number and type of objects involved in SDEC paradigm is significant. Manual updates for these software definition models are almost impossible. Therefore, the automatic update mechanism and methods need to be explored.

4) Security and privacy

The processing capacities of edge computing nodes are usually weaker than cloud computing, which is unable to load rigorous security and privacy protection mechanism. So the security and privacy becomes one of the challenges for SDEC.

VI. Conclusions

In this paper, we have proposed an approach and principles of SDEC to synergistically manage different categories of edge resources and services. Accordingly, SDED, SDESto, SDECR and SDESer are also proposed to respectively realize the software definition for edge device resources, edge storage resources, edge computing resources and edge services. Furthermore, we have designed a SDEC-based open edge computing system architecture and introduced its software definition mechanism in detail. The proposed approach can share, reuse and recombine edge resources and services. The upper-level applications are decoupled from the underlying physical devices. It provides an automated and autonomous edge framework to improve the overall service capability of the edge side. In the future, there are some challenges and open issues that need to be explored.

References

[1] G. Premsankar, M. D. Francesco, and T. Taleb, “Edge computing for the internet of things: A case study,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1275–1284, 2018.

[2] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Luo, “Fog computing based face identification and resolution scheme in internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1910–1920, 2017.

[3] P. Hu, S. Dhelim, H. Ning, and T. Qiu, “Survey on fog computing: architecture, key technologies, applications and open issues,” *Journal of Network & Computer Applications*, vol. 98, pp. 27–42, 2017.

[4] W. Shi and S. Dustdar, “The promise of edge computing,” *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[5] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, “Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges,” *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.

[6] F. Hu, Q. Hao, and K. Bao, “A survey on software-defined network and openflow: From concept to implementation,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.

[7] H. Kim and N. Feamster, “Improving network management with software defined networking,” *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[8] J. Wang, Y. Hu, H. Li, and G. Shou, “A lightweight edge computing platform integration video services,” in *2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, 2018, pp. 183–187.

[9] G. Faraci and A. Lombardo, “An nfv approach to share home multimedia devices,” in *2017 IEEE Conference on Network Softwarization (NetSoft)*, 2017, pp. 1–6.

[10] B. N. Astuto, M. M. Ca, N. N. Xuan, K. Obraczka, and T. Turletti, “A survey of software-defined networking: Past, present, and future of programmable networks,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[11] R. Buyya, R. N. Calheiros, J. Son, A. V. Dastjerdi, and Y. Yoon, “Software-defined cloud computing: Architectural elements and open challenges,” in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2014, pp. 1–12.

[12] P. Hu, H. Ning, L. Chen, and M. Daneshmand, “An open internet of things system architecture based on software-defined device,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2583–2592, 2019.

[13] D. Kreutz, F. M. V. Ramos, P. E. Verłssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[14] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, “Secure data sharing and searching at the edge of cloud-assisted internet of things,” *IEEE Cloud Computing*, vol. 4, no. 1, pp. 34–42, 2017.

[15] A. Darabseh, M. Al-Ayyoub, Y. Jararweh, E. Benkhelifa, M. Vouk, and A. Rindos, “Sdstorage: A software defined storage experimental framework,” in *2015 IEEE International Conference on Cloud Engineering*, 2015, pp. 341–346.