

# Towards Metaverse with Diversified Applications: Service Native Architecture

Yuxiang Jiang <sup>a</sup>, Li Peng <sup>b</sup>, Xiaojie Zhu <sup>c</sup>, Jinlan Ma <sup>d</sup>, Xiaozhi Yuan <sup>e</sup>,

Zhan Liu <sup>f</sup>, Qingyang Wang <sup>g</sup>

Department of Mobile and Terminal Technology Research, Research Institute of China Telecom Co., Ltd, Guangzhou, China

<sup>a</sup> jiangyx8@chinatelecom.cn, <sup>b</sup> pengl2@chinatelecom.cn, <sup>c</sup> zhuxiaojie@chinatelecom.cn,

<sup>d</sup> majl6@chinatelecom.cn, <sup>e</sup> yuanxiao@chinatelecom.cn, <sup>f</sup> liuz22@chinatelecom.cn,

<sup>g</sup> wangqy7@chinatelecom.cn

**Abstract.** With the advancement of science and technology, Metaverse will empower the entity economy and social development, and stimulate the development potential of industries such as smart city. The current chimney-style application creation process is cumbersome, which cannot support the application diversification and efficient, flexible and open service provision methods, so it cannot meet the needs of the rapid development of the Metaverse. By introducing a service engine layer that encapsulates basic service capabilities, this research proposes a Metaverse application architecture based on service native, which has characteristics of function scalability, orchestration automation, system openness, full-type access, etc. The basic service capabilities based on this architecture can support diverse needs of Metaverse applications.

**Keywords:** Metaverse; Service Orchestration; Application Diversified; Service Native Architecture.

## 1. Introduction

In recent years, the Internet industry has entered a bottleneck period, and there has been a long-term lack of breakthroughs in content carriers, dissemination methods, interaction modes and profit models. At the same time, a variety of emerging technologies have developed, such as AI, XR, blockchain and cloud computing, and eager for new products to break the deadlock today[1]. The Metaverse came into being, its applications cover many fields, with the potential to become a general platform for digital social interaction in the future[2].

Metaverse crosses the boundary of virtuality and reality, providing users with a highly immersive interactive experience. At the same time, the superposition of reality and virtuality stimulates broad business potential and opens new exports for capital[1]. Countries are actively exerting the power of technology innovation, and "multiverses" are coming one after another, but it is accompanied by the following problems:

1. Currently, each application is provided by an independent service platform and cannot be reused with each other, resulting in a waste of resources. Therefore, Metaverse urgently needs efficient and flexible service provision methods to meet the development of its diversified applications.

2. With the increasingly diverse user needs, the problem of how to attract more creators to join the construction of Metaverse, and how to create Metaverse applications with low cost and high efficiency to meet the personalized needs of users has gradually become prominent [3].

By introducing a service engine layer that package basic service capabilities, this research proposes a Metaverse application architecture based on service native. This architecture adopts a native approach to provide basic service capabilities for Metaverse, orchestrates basic capability engines to meet the diverse needs of Metaverse service, and facilitates the development of Metaverse's diverse applications.

## 2. Metaverse Application Diversification Requirements and Challenges

Metaverse applications are spread across multiple industries in different fields, and there are various requirements in different industries. Among them, diversification of applications is a necessary requirement for the development of Metaverse. However, the current application is the chimney-type, so the process of creating new applications is complicated and cumbersome, which is not conducive to the rapid development of diversified applications of the Metaverse.

### 2.1 Diversification Requirements of Applications

With the joint efforts of enterprises and governments of various countries, Metaverse will empower various fields such as the entity economy and social. In terms of enterprises, Roblox opened Metaverse firstly, and then domestic and foreign companies joined the team to tap development potential of the Metaverse, such as Facebook, Microsoft, Tencent, NetEase and other companies, focusing mainly on social networks, games and other applications[4][5]. In terms of the governments, various countries add weight to support the research and construction of Metaverse[6]. The United States focuses on social privacy and data sensitivity issues. South Korea explores applying the Metaverse to tourism. Japan plans to improve the laws and development policies for XR equipment, digital content, etc.. And many local governments in China are exploring industrial applications, aiming to combine technology with industry and promote the in-depth application of Metaverse technology in many fields.

It can be seen from the above that enterprises pay attention to the development of the Metaverse in the social and entertainment of users, and the government tends to connect Metaverse with society and the country, focusing on stimulating the economy, protecting public privacy and improving laws. Therefore, driven by enterprises and governments, Metaverse applications are expected to spread across the smart cities, smart industries, social entertainment, digital finance and other physical industries. As Metaverse gradually matures, its applications will spread across entity industries such as smart city, smart industry, social entertainment, and digital finance. A mature Metaverse should have requirements of intelligence, ultra-high computing power, storage and network resources, breaking the boundary between virtuality and reality, and enabling users to obtain immersive experience[7]. In addition, Metaverse, which covers multiple industries, should continue to create new applications and provide new services to meet the changing needs of users. Therefore, supporting diversified applications of related industries as well as efficient, flexible and open service provision methods is the necessary requirement of Metaverse's development in multiple fields.

### 2.2 Challenges

There are two major difficulties in the diversification of Metaverse applications.

First, there are countless different applications developed to meet various services purposes currently, and most of the applications provide services to users based on the Over The Top (OTT) model[8]. Due to the lack of unified specification of data format, these applications cannot effectively coordinate with each other, forming data islands and becoming "chimney-type applications"[9]. This makes the application functions customized, and the cost of maintenance and expansion is high. There are many applications with the same function that cannot be integrated and reused between applications. In addition, it is time-consuming and labor-intensive to repeat previous work when developing new applications, which is difficult to support complex and fast-changing service. Therefore, the current mode of creating applications does not match the diverse needs of Metaverse applications, which leads to the fact that Metaverse product applications in various fields only create their own small Metaverse, and it is unable to provide users with the experience of free switching between multiple applications, whose mutual inclusiveness needs to be improved.

Another difficulty is that the talents invested in the construction of the Metaverse are limited, and the application development speed has not kept up with the growth rate of people's needs in life, culture, etc., and cannot meet the personalized needs of users. Therefore, the development of the

Metaverse needs to attract more creators to join, and it is best to make the public become the builders of Metaverse.

Based on the above problems, this research proposes a Metaverse application architecture based on service native. The basic service capabilities provided by this architecture can support diversified applications of Metaverse, while lowering the threshold of creating applications and encouraging the growth of the creator team.

### **3. Metaverse Application Architecture based on Service Native**

#### **3.1 Sketch**

By introducing a service engine layer that encapsulates basic service capabilities, this research proposes a Metaverse application architecture based on service native, where the native network provides service atomic capability engines, so that Metaverse users can orchestrate atomic capabilities according to their individual application requirements. The analysis in this chapter includes three aspects: design principles, architecture composition, and application examples. It shows that the architecture has many advantages such as high efficiency and flexibility, and lays the foundation for the innovative development of business models.

#### **3.2 Architecture Design Principles**

In different scenarios, users have different requirements for Metaverse applications, and each user can design and create the Metaverse[10]. In order to meet the needs of users for diversified applications of the Metaverse, this paper proposes a Metaverse application architecture. The design principles of this architecture can be attributed to the following terms:

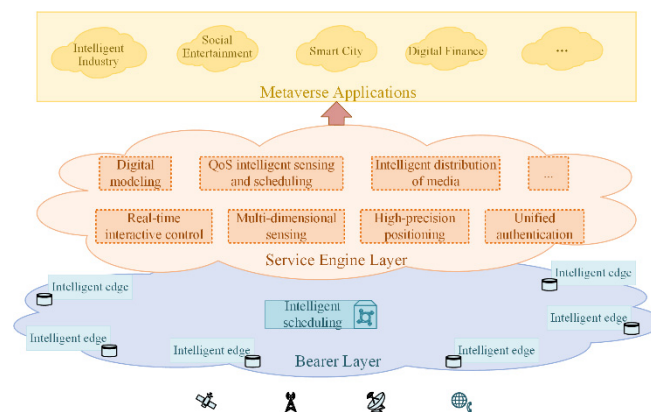
- Meet the common needs of Metaverse applications. Provide a public capability platform for the future network in the way of "service native", and continuously improve the ability of the future network to support the Metaverse.
- Possess the basic ability of efficient orchestration. The basic service capabilities are modularized into independent service engines, and the service can be orchestrated by invoking the service engine to quickly create new applications. Support user scenario customization, effectively improve user experience, and meet individual needs.
- Cloud-enabled and programmable. Cloud-based network architecture does not require dedicated hardware, which can speed up the deployment of infrastructure and expand the application of Metaverse conveniently and quickly.
- Basic service capabilities are directly opened to third parties[11]. An open service architecture can attract developers to create value and fully tap the potential of the platform. In addition, different platforms adopt a unified open interface to lay the foundation for the compatible development between platforms.
- Support cross-platform verification. Unified authentication can support collaboration between platforms and the integrity of cross-chain services. Users can log in quickly and securely without memorizing multiple passwords for different applications, so that account security is guaranteed.
- Has scalable capabilities. A system with flexible expansion ability can respond quickly with the changes of user needs, avoid system reconfiguration and minimize the impact on current applications.
- Be accessed in full type. Metaverse is developing based on future network, so the full-type access mode of the bearer layer is the premise for users to access the network anytime and anywhere and enjoy high-quality services.

Based on the above principles, the Metaverse application architecture with the characteristics of scalable function, automatic arrangement, open system and full-type access can provide basic service

capabilities for Metaverse. This basic service capabilities can be invoked repeatedly, which reduces the construction and maintenance costs, and effectively guarantees the effectiveness of the expansion and integration of Metaverse applications.

### 3.3 Metaverse Application Architecture

General overview of the architecture. The Metaverse application architecture based on service native proposed in this research is shown in Fig.1, which can support applications of Metaverse in smart city, intelligent industry, social entertainment, digital finance and other fields. The service native architecture includes two parts from top to bottom, namely, the service engine layer and the bearer layer. The service engine layer provides basic service capabilities to meet the common needs of Metaverse applications. This layer is based on cloud deployment and includes seven major engines, including Digital modeling, Intelligent distribution of media, and Real-time interactive control, etc. At the same time, new engines can be expanded to meet new needs. Each engine is independent of each other and open to third parties, so that third parties can efficiently orchestrate applications by invoking these engines, which lays the foundation for user cross-platform verification. The bearer layer supports different access modes, fixed or mobile, and serves as a bridge for signal transmission between devices in full-type access mode and service applications. This layer has independent intelligent scheduling capability and covers multiple intelligent edges.



**Fig 1.** Metaverse application architecture

The service engine layer. Service engine layer can cooperate with service providers upwards, and open basic service capabilities to them, which can meet the diverse needs of Metaverse services, and is tightly coupled with bearer layer downwards to fully mobilize the capabilities of the bearer layer. The service engine layer consists of the following seven engines, and the function of each engine are independent of each other.

- Digital modeling[12]: Has the mapping ability from the physical world (such as objects and spaces.) to the digital world. Use this engine to build a virtual twin, and perform efficient analysis, diagnosis, simulation and control of the physical world based on data and models, helping the physical world to achieve low-cost trial and error, intelligent decision-making and high-efficiency innovation.
- Real-time interactive control[13]: Provide real-time communication of various media streams such as holographic information, audio and video, between applications and terminals, terminals and terminals. The optional streaming media protocols include QUIC and WebXR. The services supported by this engine have the characteristics of ultra-low delay and ultra-large throughput, and are more sensitive to changes in network bandwidth, delay, jitter, and mobility, and have higher requirements for network quality.
- Quality of Service (QoS) intelligent sensing and scheduling: Sense and report QoS parameters and service status in real time, control QoS according to policies, intelligently predict changes

in QoS parameters for a period of time in the future, and optimize and adjust QoS policies in time.

- Multi-dimensional sensing: Provide environmental information around the target location. Through this engine, perception functions such as detection, positioning, recognition, and imaging of targets can be realized.
- Intelligent distribution of media[10]: Intelligently distribute massive multimedia data to appropriate multi-dimensional media processing units for processing according to the computing power requirements of media streams, such as encoding, compression, and rendering, as well as the state and capability of computing resources of multi-dimensional media processing units such as cloud, network, and terminal.
- High-precision positioning[14]: Provide accurate positioning information of terminals in various access methods. The engine utilizes advanced algorithms, edge computing and AI to generate ultra-high-resolution images, enabling high-precision real-time perception services.
- Unified authentication[7]: Realize single sign-on between application systems. The authentication function is implemented under a unified architecture, and transmits the identity authentication result securely to each application system, which not only improves the utilization efficiency of equipment, but also provides guarantee for the security of user accounts.

Bearer layer. The bearer layer guides each service information flow from the source end to the destination end according to the requirements of the upper layer, enables users to access the network anytime and anywhere, schedules network resources according to the attribute requirements of each service to ensure the function and performance of the service, and provides QoS guarantee for different types of services.

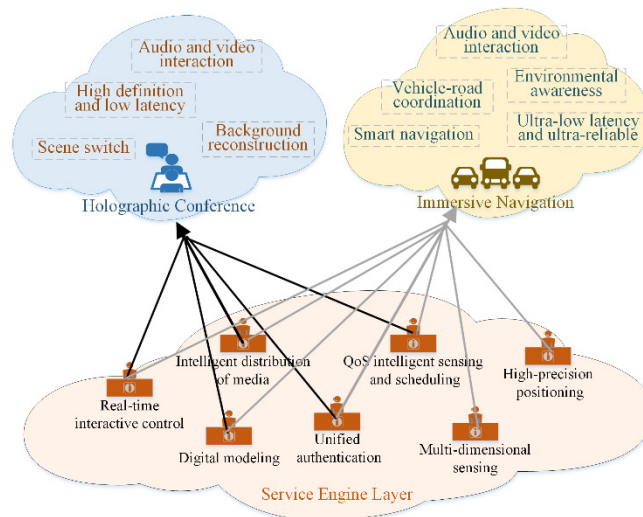
Based on IP and other bearer methods, the bearer layer introduces intelligent edge and intelligent scheduling function to fulfill the ultra-low-latency and ultra-high-bandwidth information transmission requirements of the Metaverse. Intelligent edge refers to the introduction of distributed computing and AI on edge devices to complete the access authentication of Metaverse users and the forwarding of media information. The function of intelligent scheduling is to complete the scheduling of network resources such as access, computing power, and bandwidth.

## 4. Use Case

### 4.1 Service Orchestration Example

Service orchestration is the core capability of the service native architecture. Based on cloud services, users can quickly complete the unified orchestration of applications and resources through invocation, migration, and reorganization, efficiently create scalable applications. This section takes the holographic conference and immersive navigation services in the smart city as examples to analyze the efficiency and importance of service orchestration for creating and expanding applications.

The process of creating applications through service orchestration is shown in the Fig 2. The upper independent cloud represents an application in certain scenarios such as medical treatment, education, conference or transportation. The needs of the application are listed in the cloud. The lower cloud represents the service engine layer, which provides basic service capabilities for applications. Applications meet their corresponding requirements by calling multiple capability engines of the service engine layer. In the holographic conference and immersive navigation scenarios, the invocation of the capability engine during the application creation process is represented by black and gray arrows respectively.



**Fig 2.** The example of service orchestration

In the holographic conference scenario, users need to communicate without obstacles in the conference room. Therefore, 1) by calling the Digital modeling engine to establish the background, the environment mapping from the physical to the virtual world is completed. 2) The Real-time interactive control engine and Intelligent distribution of media engine are called to meet the needs of audio and video interaction. 3) QoS intelligent sensing and scheduling engines optimize and adjust policies according to network status to ensure audio and video quality in conferences. 4) Unified authentication engine is used to ensure senseless switching between different scenarios. 5) Finally, the logical relationship between the called engines is defined, and the creation of the application in this scenario is completed.

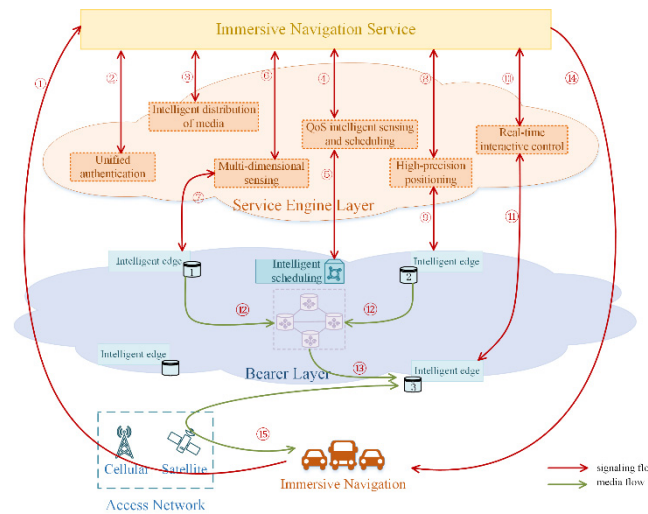
In the immersive navigation scenario, it is also necessary to call Digital modeling, Real-time interactive control, Intelligent distribution of media and QoS intelligent sensing and scheduling engines to meet the basic needs of navigation, audio and video interaction, ultra-low delay and ultra-high reliability during driving. The Unified authentication engine provides convenience for user ID verification. Therefore, these same engines in the holographic conference scenario can be reused. In addition, in the process of driving, it is necessary to achieve precise positioning in high-speed movement, as well as to perceive the obstacles near the vehicle, such as pedestrians and buildings. So, this scenario needs to add High-precision positioning engine and multi-dimensional sensing engine. Therefore, the progress of creating an immersive navigation application is, 1) calling the same engines as the holographic conference scene application, such as Real-time interactive control engine, 2) then adding High-precision positioning and multi-dimensional sensing engines to update the current road conditions and surrounding environment in real time, 3) defining the logical relationship between all engines finally. So far, the creation of this application is completed by extending the service engine layer of conference scenario.

To sum up, the service engine layer can provide basic service capabilities for applications. Applications in different scenarios have different requirements for service capabilities. Users can call the corresponding service engine according to their own needs to build applications. When the requirements change, the service engine layer can be extended to create new applications by adding corresponding engines. And service orchestration makes this process more efficient and flexible to meet the personalized needs of users.

## 4.2 Service Process Example

Taking the immersive navigation scenario as an example, once the immersive navigation service is triggered, the application layer will call the relevant engine of the service engine layer to meet the

service requirements. The specific process is shown in Fig 3. The red line in the figure represents the transmission of signaling flow, and the green line represents the transmission of media flow.



**Fig 3.** Service process of immersive navigation scenario.

- 1) The immersive navigation terminal service request is triggered to the application layer.
- 2) The application layer invokes the Unified authentication engine to query the user's "immersive navigation" service subscription profile, and obtains the service QoS indicators in the profile and the media resource requirements in this service request.
- 3) According to the service QoS index and the media resource requirements in this service request obtained in step 2, the application layer calls the Intelligent distribution of media engine to calculate the intelligent edge that can provide media processing for this service, and obtains the address list of the intelligent edge, such as Intelligent edge 1, Intelligent edge 2, Intelligent edge 3 in the figure.
- 4) The application layer calls the QoS intelligent sensing and scheduling engine to detect the network status and calculate the QoS policy according to the available intelligent edge and the QoS indicator requirements of this service obtained in step 3.
- 5) The QoS policy calculation obtained in step 4 is sent to the intelligent scheduling of the bearer layer. The intelligent scheduling of the bearer layer schedules the bearer paths that meet the requirements of the QoS indicators according to the QoS policy, and obtains the path from the Intelligent edge 1 and the Intelligent edge 2 to the Intelligent edge 3 as well as the path between the Intelligent edge 3 and the immersive navigation terminal. This path is used for the transmission and forwarding of media streams.
- 6-7) The application layer invokes the multi-dimensional sensing engine to obtain the user's current environment information, and transmits the environment information to the Intelligent edge 1 responsible for this service media processing according to the address of the Intelligent edge 1 obtained in step 3.
- 8-9) The application layer invokes the High-precision positioning engine to obtain the user's current positioning information, and transmits the positioning information to the Intelligent edge 2 responsible for this service media processing according to the address of the Intelligent edge 2 obtained in step 3.
- 10-11) The application layer calls the Real-time interactive control engine control Intelligent edge 3 obtained in step 3 to perform operations such as media integration and interactive control.
- 12-13) According to the bearer path planned by the intelligent scheduling in step 5, the Intelligent edge 1 and the Intelligent edge 2 send the processed media stream to the Intelligent edge 3 for integration processing.
- 14) The application layer sends the address of the Intelligent edge 3 obtained in step 3 to the immersive navigation terminal.
- 15) The immersive navigation terminal receives the media stream transmitted by the Intelligent edge 3 and presents it to the user.

## 5. Conclusion

By introducing a service engine layer that encapsulates basic service capabilities, building a service middle platform with service engines as the core, this research proposes a Metaverse application architecture based on service native, which has characteristics of function scalability, orchestration automation, system openness, and full-type access, etc. This paper takes holographic conference and immersive navigation service orchestration as examples to illustrate that service orchestration makes the creation and expansion of applications more efficient and flexible. At the same time, it takes the workflow of immersive navigation service as an example to show the end-to-end implementation process of this architecture. The advantages of using this architecture are as follows:

- Users can call the corresponding service engines for efficient orchestration and flexibly create new application according to their own wishes.
- When the requirements of users change, the application only needs to be added or deleted the corresponding service engine, then it can meet the user's personalized needs without rebuilding.
- The openness of the system provides a good foundation for the compatible development of different applications, so that users can not only complete social, commercial and other behaviors within a Metaverse application, but also shuttle freely among different Metaverses.
- The full-type access mode breaks the time and space constraints, enabling users to enter Metaverse whenever and wherever they want.

To sum up, the architecture provides Metaverse with an efficient, flexible and open service provision method to meet the changing needs of users. In addition, the simple and convenient way to create applications based on this architecture will make the threshold of application development lower, attract more talents to join the construction of Metaverse, and promote the vigorous development of Metaverse applications.

Furthermore, the application architecture based on service native has made innovative changes to the IMT system, and this paper takes the Metaverse application as an example to demonstrate the advantages of this architecture. The engine requirement of the architecture has been established as a project at ITU-T's SG16, and is being studied concurrently. Theoretically, all 6G basic service engines (such as Service control engine) can be built in the service engine layer, so this architecture can not only provide services for Metaverse applications, but also provide comprehensive services for 6G basic services such as intelligent interaction, communication perception, and holographic communication. Therefore, this architecture has a good reference value for future network development.

## References

- [1] M. A. I. Mozumder, M. M. Sheeraz, A. Athar, S. Aich and H. -C. Kim, "Overview: Technology Roadmap of the Future Trend of Metaverse based on IoT, Blockchain, AI Technique, and Medical Domain Metaverse Activity," 2022 24th International Conference on Advanced Communication Technology (ICACT), 2022, pp. 256-261.
- [2] Feng Tian, "SenseTime Intelligent Industry Research Institute: Intelligent Industry Metaverse Walker: Imagination is more important than knowledge," 2021. [online]. Available at: <[https:// www. sgpjbg. com/ info/30192.html](https://www.sgpjbg.com/info/30192.html)>.
- [3] J. Xing, G. Zhao. "Metaverse Token," Beijing, China: Chinese Translation Press, 2021.
- [4] Roblox Blog, Roblox Corp, 2022. Accessed: Nov. 1, 2021. [Online]. Available at: [https:// blog. roblox. Com/](https://blog.roblox.com/).
- [5] Jay David Bolter; Maria Engberg; Blair MacIntyre, "8 The Myth of Total VR: The Metaverse," in Reality Media: Augmented and Virtual Reality, MIT Press, 2021, pp.137-146.

- [6] Caichun Gong, "2022 China Metaverse White Paper," 2022. [online]. Available at:<[https:// 5gxt. com/ material/ data2023.htm](https://5gxt.com/material/data2023.htm)>.
- [7] S. -M. Park and Y. -G. Kim, "A Metaverse: Taxonomy, Components, Applications, and Open Challenges," in IEEE Access, vol. 10, pp. 4209-4251, 2022.
- [8] S. Oh and Y. Kim, "Interoperable OAuth 2.0 Framework," 2019 International Conference on Platform Technology and Service (PlatCon), 2019, pp. 1-5.
- [9] J. An, J. Hwang and J. Song, "Interworking technique and architecture for connecting LAN IoT devices towards standardized IoT service layer platform," 2016 IEEE 5th Global Conference on Consumer Electronics, 2016, pp. 1-2.
- [10] Focus Group on Technologies for Network 2030 (FG NET-2030), "Representative use cases and key network requirements for Network 2030," 2021.
- [11] Z. Fang, L. Cai and G. Wang, "MetaHuman Creator The starting point of the metaverse," 2021 International Symposium on Computer Technology and Information Science (ISCTIS), 2021, pp. 154-157.
- [12] B. R. Barricelli, E. Casiraghi and D. Fogli, "A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications," in IEEE Access, vol. 7, pp. 167653-167671, 2019.
- [13] Focus Group on Technologies for Network 2030 (FG NET-2030), "New Services and Capabilities for Network 2030: Description, Technical Gap and Performance Target Analysis," 2019.
- [14] Cui-Xia Li, Wei-Ming Liu and Jian-Wei Wu, "High-precision positioning algorithm based on GPS," 2010 2nd International Conference on Advanced Computer Control, 2010, pp. 364-368.