Design of an Intelligent IoT Platform Based on the Synergy of Edge Computing and Cloud Computing

Jinchuan Wei*, Ming Ni, Qiaojin Guo, Zhiwei Yu, Suhang Liu
Nanjing Research Institute of Electronic Engineering, Nanjing, Jiangsu, China
**Corresponding Author*

Abstract: This paper addresses the massive data processing challenges brought about by the rapid expansion of the Internet of Things (IoT), focusing on the increasingly prominent limitations of traditional cloud computing (CC) models in areas such as real-time response, network bandwidth, and data privacy. To systematically address these issues, this paper aims to design an intelligent IoT platform architecture based on the collaboration of edge computing (EC) and CC. The research employs a systematic design approach, proposing a three-layer overall architecture comprising a "device and edge layer, network transmission layer, and CC center layer," and elaborates on its core workflow of "edge-cloud" vertical collaboration and "cloud-edge" horizontal offloading. Through the design of key functional modules and the analysis of typical application scenarios, the paper demonstrates that the platform can effectively integrate the real-time processing capabilities of the edge side with the deep intelligence advantages of the cloud. The research results show that this collaborative architecture has significant value in ensuring low-latency response, optimizing bandwidth costs, enhancing system reliability, and achieving continuous evolution of global intelligence. This research provides a clear and feasible design reference for building efficient, reliable, and secure next-generation IoT systems, and has positive implications for promoting the practical application and industrial development of related technologies.

Keywords: Edge Computing; Cloud Computing; Internet of Things; Platform Architecture; Collaborative Design

1. Introduction

With the rapid development of information technology, the Internet of Things (IoT) has permeated from conceptualization to various aspects of the social economy, such as industrial production, urban management, and smart homes, realizing a wide connection between the physical world and the digital world [1]. In this process, the centralized cloud computing (CC) paradigm has become the mainstream mode supporting IoT applications due to its powerful elastic computing and massive storage capabilities. It aggregates all data to remote data centers for processing, providing a solid foundation for complex model training and global business analysis [2]. However, the exponential expansion of the IoT scale and the increasing refinement of application scenarios have gradually highlighted the inherent limitations of traditional cloud architecture. Uploading all raw data involving privacy or core production to the cloud has also brought significant security and compliance risks [3,4].

To address the aforementioned challenges, edge computing (EC), as a distributed computing paradigm that pushes computing, storage, and network resources down to the source of data, has received widespread attention from academia and industry in recent years [5]. It achieves local processing and real-time response of data by deploying computing nodes (such as smart gateways and edge servers) at the network edge, effectively reducing network transmission load and business latency. Researchers have explored the value of EC in the Internet of Things from multiple perspectives [6]; other studies are dedicated to deploying lightweight artificial intelligence models on resource-constrained edge devices to support local intelligence [7]. However, emphasizing EC in isolation also has limitations, such as the fact that edge nodes cannot match the cloud in terms of computing power, data breadth, and model training depth [8]. Therefore, the consensus in the field is

gradually becoming clear: architectures that rely solely on the cloud or the edge are not optimal solutions, and EC and CC are not substitutes for each other, but rather have significant complementary characteristics. Although existing research has recognized the importance of collaboration, most of the work either focuses on a specific technical detail (such as task scheduling algorithms) or only proposes a conceptual framework, lacking a coherent explanation of the overall platform design logic, modular functional composition, and collaborative workflow in typical scenarios [9,10].

Therefore, this paper aims to move beyond discussions of single technologies and, from the perspective of an integrated system, design an overall architecture for a smart IoT platform based on deep collaboration between EC and CC. Methodologically, this paper first analyzes the inherent logic and design goals of cloud-edge collaboration, and then proposes a three-layer overall architecture comprising a "device and edge layer, network transmission layer, and CC center layer." This paper aims to demonstrate how this design organically integrates the real-time responsiveness of the edge with the deep intelligence of the cloud, thereby systematically addressing core challenges such as latency, bandwidth, security, and cost.

2. Overview of Related Technologies

2.1 Core Features of CC

CC is a mode of accessing shared, configurable computing resources (such as networks, servers, storage, applications, and services) in an on-demand, easily scalable manner over a network (usually the Internet). It essentially provides computing power as a standardized, centralized service.

Its main characteristics are reflected in the following aspects:

(1) Centralization of resources. CC centralizes a large number of distributed compute, storage, and network resources in large data centers for unified management and scheduling. Users don't need to care about the physical location and specific details of the resource.

(2) Strong elasticity and scalability. CC platforms can dynamically and automatically allocate and release resources according to changes in users' business loads. This elastic scalability enables it to calmly respond to sudden data floods or computing needs that may occur in IoT scenarios.

(3) In-depth analysis and model training capabilities. Thanks to massive data aggregation and near-unlimited computing potential, CC centers are ideal places for complex data mining, machine learning model training, and global business intelligence analysis. It can extract universal laws and knowledge from massive historical data, and provide "brain" level intelligent support for the entire system.

2.2 Core Features of EC

EC is a new computing paradigm that moves compute, storage, and network functions from centralized cloud data centers to the edge side of the network closer to data sources or users.

EC is characterized by its proximity and distribution:

(1) Low latency and real-time response. Since the computing node is close to the end device, data can be processed without long-distance, multi-hop network transmission, and can achieve millisecond-level response. This is critical for applications that are extremely latency-sensitive, such as industrial control, autonomous driving, and interactive video analytics.

(2) Bandwidth saving and data burden reduction. Initial processing, filtering, and aggregation at the source of the data can greatly reduce the amount of data that needs to be uploaded to the cloud. This not only relieves the bandwidth pressure on the core network but also reduces the cost of data transmission and storage.

(3) Localized privacy and security enhancements. Sensitive or private data can be processed at local edge nodes without leaving the physical or administrative boundaries it generates. The raw data does not have to be uploaded to the cloud, reducing the risk of data leakage during transmission and cloud storage, while also helping to meet compliance requirements for localized data storage.

2.3 The Inevitability of Synergy

Looking at CC or EC in isolation does not fully meet the full range of needs of modern intelligent IoT applications. The two are not essentially a competition or substitution relationship, but present a clear and strong complementarity, which inherently determines

the inevitability of synergy.

In terms of task characteristics, CC is good at handling global, non-real-time, and computing-intensive tasks. EC, on the other hand, is adept at handling local, real-time, agile and responsive tasks. Therefore, an ideal intelligent IoT platform architecture must achieve functional decoupling and logical coordination between edge and cloud capabilities. As a distributed computing front-end, edge nodes mainly undertake real-time data processing and local decision-making tasks with high throughput and low latency, forming a preliminary closed loop of business response. The cloud data center is responsible for the aggregate storage of cross-domain heterogeneous data, the in-depth training of complex models, and the generation of global policies as a centralized intelligent backend. By continuously distributing cloud-optimized algorithm models and business rules to edge nodes, the system can realize the dynamic distribution of knowledge and experience and capability iteration. This collaborative mechanism based on "real-time execution at the edge and global optimization in the cloud" together constitutes an intelligent system with elastic responsiveness and continuous learning and evolution, which is the core paradigm of modern IoT platform architecture design.

3. Overall Design of the Intelligent Internet of Things Platform for Cloud-Edge Collaboration

3.1 Design Objectives

The platform is designed to meet the core needs of complex IoT application scenarios, and the following key goals are established:

Low latency and high real-time: For scenarios such as industrial control and video surveillance, it ensures that critical services can achieve millisecond-level response on the edge side and meet strict latency constraints.

Bandwidth optimization and cost control: By cleaning, aggregating, and preliminary analysis of raw data on the edge side, the amount of data that needs to be uploaded to the cloud is greatly reduced, thereby reducing the pressure on network transmission bandwidth and operating costs.

System reliability and resilience: Even if the network connection is temporarily interrupted or cloud services are unreachable, the edge side should be able to maintain the local autonomous operation of core services, ensure that basic system functions are not interrupted, and improve overall service availability.

Data security and privacy protection: Supports preprocessing such as desensitization and encryption at the source of data generation (edge side) to avoid sensitive raw data being transmitted in plaintext in the network or stored in the cloud to reduce the risk of data leakage.

Global intelligence and continuous evolution: Leverage the powerful computing power and data aggregation advantages of the cloud to conduct in-depth data analysis, model training, and knowledge mining, and dynamically distribute optimized algorithm models and strategies to the edge side to achieve continuous iteration and improvement of the overall intelligence level of the system.

3.2 Overall Architecture Design

To achieve the above goals, the platform adopts a hierarchical loosely coupled architecture, which is divided into the device and edge layer, the network transport layer, and the CC center layer from the bottom up (the overall architecture is shown in Figure 1). The three layers interact with the protocol through standardized interfaces and together form an organic synergistic whole.

The device and edge layer is the platform's physical sensing and real-time processing unit. This layer consists of a massive number of heterogeneous IoT terminal devices (such as sensors, controllers, and cameras) and their adjacent EC nodes (such as smart gateways, edge servers, and micro data centers). Terminal devices are responsible for collecting raw physical world data; edge nodes undertake core EC tasks, including data access, protocol parsing, real-time stream processing, local rule engine execution, lightweight AI model inference, and rapid local feedback control. This layer is crucial for the platform to achieve low-latency response and bandwidth optimization.

The network transport layer serves as the link for the flow of platform data and commands. It is responsible for providing a reliable, secure, and efficient bidirectional communication channel between the edge layer and the cloud. This layer needs to adapt to various network environments (such as wired, 4G/5G, LoRa, etc.) and support encryption, integrity verification,

and priority scheduling of transmitted data. Its design focuses on ensuring the quality of service for the uploading/downloading of collaborative commands and critical data, especially by providing fault tolerance and retransmission mechanisms when network conditions are unstable.

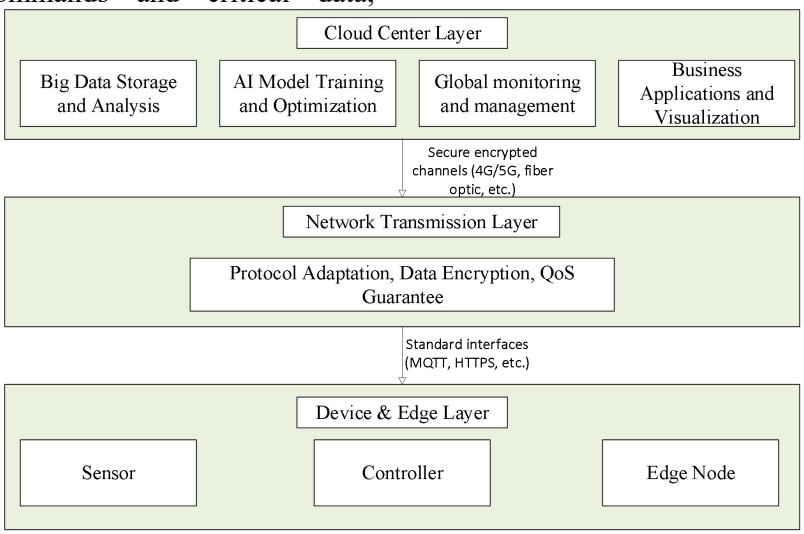

Figure 1. Overall Architecture Diagram of Cloud-Edge Collaborative Intelligent IoT Platform

The CC center layer serves as the platform's control and global resource management hub. It comprises centrally deployed cloud data centers, possessing near-limitless elastic storage and ultra-high computing power. This layer is primarily responsible for archiving and storing massive amounts of historical data, performing complex big data analysis and mining, training and optimizing large-scale machine learning models, unified monitoring and management of all network devices, and providing comprehensive business applications and visualizations for end-users. The cloud aggregates global information, generates superior business strategies and algorithm models, and distributes them to the edge layer to guide execution.

3.3 Collaborative Workflow Design

The core of the platform lies in the dynamic and intelligent task collaboration and data flow between the edge and the cloud. Its workflow design follows the core logic below (the collaboration process is shown in Figure 2):

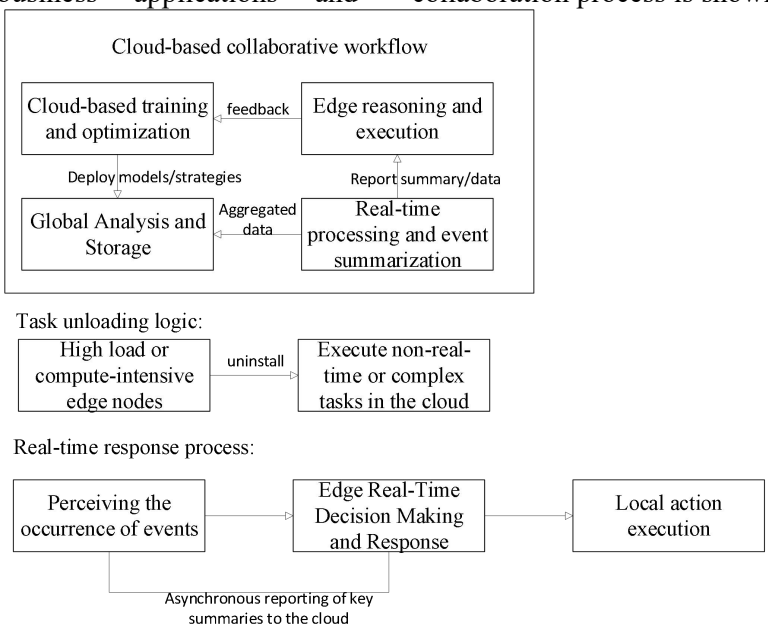

Figure 2. Cloud-Edge Collaboration Workflow Diagram

The "edge-cloud" vertical collaboration process is the key path. For time-sensitive tasks, the platform adopts a "real-time edge response, asynchronous cloud filing" process. For

example, when an industrial sensor detects that the vibration value of the equipment exceeds the safety threshold, the edge node immediately triggers a local alarm and executes an emergency shutdown command according to preset rules. At the same time, it asynchronously uploads key summary information of this event (rather than all the original waveform data) to the cloud for recording and subsequent analysis. For model-driven tasks, the process follows "cloud training, edge inference, and continuous iteration." The cloud uses aggregated historical data to train or optimize AI models, and then distributes the lightweight models to the edge nodes. The edge nodes use these models for local real-time inference. Meanwhile, new labeled data or model execution feedback generated at the edge are uploaded to the cloud for the next round of model retraining, forming a closed-loop optimization.

The "cloud-edge" horizontal collaboration logic is reflected in dynamic resource allocation and task offloading. The platform has a built-in resource status monitoring mechanism. When a single edge node faces a sudden surge in computing load, exceeding its processing capacity, the platform can intelligently "offload" some non-real-time or computationally intensive subtasks (such as batch analysis of historical data) to the cloud for execution, according to preset strategies. Conversely, for certain globally distributed analysis tasks, the cloud can also dynamically decompose and distribute the task to multiple edge nodes for parallel processing based on the real-time load and data relevance of each edge node. This dynamic task offloading and distribution logic aims to maximize the overall computing resource utilization efficiency and achieve load balancing of the system.

This collaborative workflow ensures that data processing is completed at the most appropriate level, meeting both real-time requirements and fully utilizing the powerful analytical capabilities of the cloud, thus forming the foundation for the platform's intelligent and efficient operation.

4. Platform Key Module Design

4.1 Resource Management and Scheduling Module

The resource management and scheduling module is the core of the platform's unified view and intelligent allocation of resources across different levels. Its design goal is to abstract heterogeneous computing, storage, and network resources and dynamically schedule them according to the needs of collaborative workflows. The module mainly consists of three parts: a resource abstraction layer, a status monitor, and a scheduling decision engine.

The resource abstraction layer is responsible for shielding the differences in underlying hardware and infrastructure. At the edge, this layer virtualizes the CPU, memory, storage, and dedicated accelerator (NPU) resources of various edge nodes into standardized computing units; at the cloud, it interfaces with the virtualized resource pools of cloud service providers. Through a unified resource description model, the platform can consistently perceive and manage the overall resource status from the edge to the cloud.

The status monitor continuously collects and summarizes real-time operational metrics for each resource unit, including but not limited to compute load, memory usage, network bandwidth consumption, task queue length, and node network connectivity. The monitoring data is used for real-time visualization, providing administrators with a comprehensive view of system operation; it also serves as input for scheduling decisions.

The scheduling decision engine executes scheduling based on preset strategies and real-time status. Its core scheduling strategies include two categories: first, collaborative task mapping, which intelligently decides whether to allocate tasks to edge nodes or offload them to the cloud based on their real-time requirements, computational complexity, and data locality; second, dynamic load balancing, where the scheduling engine can migrate some of its migrateable tasks to other idle edge nodes in the same region or to the cloud to prevent performance bottlenecks; conversely, it can also distribute some lightweight tasks from the cloud to the edge to reduce cloud pressure. The engine allows administrators to optimize strategies according to different business scenarios through a strategy configuration interface.

4.2 Security and Privacy Protection Module

The collaborative architecture introduces distributed data processing nodes, which expands the security perimeter but also brings

new privacy challenges. This module's design spans the entire "device-edge-cloud-pipe" chain, aiming to build a defense-in-depth system.

At the device and edge sides, module design emphasizes source protection. First, it provides hardware security anchors, such as integrating a Trusted Execution Environment (TEE) or security chip, to protect the integrity of the edge node's startup process, key storage, and critical code. Second, it implements local data anonymization and desensitization. When data leaves the terminal device or enters the edge node, sensitive fields (such as facial features and location trajectories) are perturbed, generalized, or encrypted according to policies to ensure that data uploaded to the network does not contain original information that can directly identify individuals or critical facilities.

At the network transport layer, the module enforces end-to-end communication security. All cross-layer data and control command transmissions must be conducted through a secure encrypted channel based on TLS/DTLS. Furthermore, the module supports integrity verification of transmitted data to prevent data tampering during transmission.

On the cloud side, the module focuses on centralized security management and auditing. A unified identity authentication and access control center is designed to perform strong authentication on all connected devices, edge nodes, and users, and implement fine-grained access authorization based on roles and attributes. Data stored in the cloud is encrypted by default, with keys managed by an independent key management service. Simultaneously, the module provides comprehensive security audit logging capabilities, recording all critical operations and data access behaviors for easy post-event traceability and analysis. Through a unified security policy management center, the module allows administrators to define security policies once and automatically distribute them to relevant edge nodes for execution, ensuring consistency of security policies across the cloud and edge.

4.3 Data Management and Service Module

This module is responsible for defining the lifecycle, flow rules, and service interfaces of data across different layers of the platform, and is the concrete manifestation of collaborative logic at the data level.

The module employs a tiered data storage strategy. At the edge, high-performance caches and temporary storage areas are designed to store frequently accessed data, supporting rapid read and write access for edge applications. For data requiring long-term storage or global analysis, preliminary cleaning, compression, and aggregation are performed at the edge, followed by asynchronous, tiered archiving to cloud object storage or time-series databases. Metadata is managed uniformly across both the edge and cloud to support efficient data discovery and retrieval.

Building upon data flow, the module provides standardized data service interfaces. It shields upper-layer applications from the complexity of data physical location, providing a unified data access view. For scenarios requiring batch data, such as model training, the module offers efficient data lake access services, supporting direct synchronization or retrieval of processed standardized datasets from edge nodes to the cloud.

In addition, the module features a specially designed model and knowledge distribution channel. After model training is completed in the cloud, this channel is responsible for securely and reliably distributing the model, its version information, and performance metrics to designated edge node groups. The channel supports differential updates and breakpoint resumption to adapt to potentially unstable network environments at the edge, ensuring that knowledge can be efficiently radiated from the center to the edge.

5. Conclusion

This paper addresses the limitations of traditional cloud-centric IoT architectures in terms of real-time performance, bandwidth costs, and data privacy. It designs a smart IoT platform architecture that integrates EC and CC. The design employs a three-layer architecture: a device and edge layer, a network transmission layer, and a CC center layer. It clearly defines the functional boundaries and collaborative interfaces of each layer and elaborates on the vertical collaborative process based on "instant edge response and asynchronous cloud filing" and "cloud training, edge inference, and continuous iteration," as well as the horizontal collaborative logic supporting dynamic task offloading. Furthermore, the design considerations for key modules such as resource

management and scheduling, security and privacy protection, and data management and services are explained, thus constructing a system model that is clearly layered, functionally decoupled, and organically collaborative. Future research can build upon this foundation to further explore AI-driven adaptive collaborative scheduling algorithms, lightweight unified security frameworks for heterogeneous edge environments, and standardized collaborative interface protocols, thereby promoting the deep evolution of cloud-edge collaboration from architectural design to large-scale stable deployment.

References

[1] Yi Zhang. (2025). Application of Internet of Things Technology in Electrical Engineering. New Engineering Technologies and Industrial Development, 1(1).

[2] Shi Jianfeng, Chen Xinyang, & Li Baolong. (2025). Research on task offloading and resource allocation algorithm in cloud-edge-end collaborative computing for the Internet of Things. Journal of Electronics & Information Technology, 47(2), 458-469.

[3] Wang Yibing, & Liu Yang. (2025). Blockchain Empowers E-commerce Transactions: The Dual Effect of Security Protection and Transparent Governance. E-Commerce Letters, 14, 621.

[4] Yin Jianan, Ding Hui, Qiao Peiran, et al. Feature extraction and similarity measurement of operation scenariost in airport surface. Command Information System and Technology, 2025,16(2):92-100.

[5] Modupe, O. T., Otitoola, A. A., Oladapo, O. J., Abiona, O. O., Oyeniran, O. C., Adewusi, A. O., ... & Obijuru, A. (2024). Reviewing the transformational impact of EC on real-time data processing and analytics. Computer Science & IT Research Journal, 5(3), 693-702.

[6] Sharma, M., Tomar, A., & Hazra, A. (2024). EC for industry 5.0: Fundamental, applications, and research challenges. IEEE Internet of Things Journal, 11(11), 19070-19093.

[7] Goriparthi, R. G. (2024). Hybrid AI Frameworks for EC: Balancing Efficiency and Scalability. International Journal of Advanced Engineering Technologies and Innovations, 2(1), 110-130.

[8] Peng Shaoliang, Bai Liang, Wang Li, Cheng Minxia, & Wang Shulin. (2024). Trusted EC for Smart Healthcare. Telecommunications Science, 36(6), 56-63.

[9] Yang Liu, & Wei Guang Liu. (2025). Discussion on the application status and future trend of IoT technology in smart cities. Smart City Applications, 8(1), 99-102.

[10] Zhang Jianwei, Chen Xu, Wang Shuyang, Jing Yongjun, & Song Jifei. (2025). A review of the application of spatiotemporal graph neural network in the Internet of Things. Journal of Computer Engineering & Applications, 61(5).

