International Journal of Research Publication and Reviews, Vol 4, no 12, pp 251-262 December 2023

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Edge-To-Cloud Continuum: Integration and Challenges

Jayprakash Patil

Institute: DKTE's Textile and Engineering College
Email: jayprakashpatil9096@gmail.com

ABSTRACT:

The integration of edge computing and cloud technologies forms a continuum that spans the distributed computing landscape, offering a seamless and comprehensive solution to address the diverse requirements of modern applications. This paper explores the challenges and opportunities inherent in the edge-to-cloud continuum, focusing on the synergy between edge devices and centralized cloud resources.

The integration of edge and cloud computing introduces a paradigm shift in data processing, storage, and application deployment. This paper delves into the architectural considerations, communication protocols, and data management strategies that facilitate efficient collaboration between edge devices and cloud infrastructure. It also addresses the dynamic nature of edge environments and the need for adaptive solutions to accommodate diverse use cases and application requirements.

Challenges in security, privacy, latency, and resource optimization are thoroughly examined, emphasizing the importance of a holistic approach to ensure the reliability and robustness of the edge-to-cloud continuum. The paper discusses the implications of data governance and compliance in this distributed ecosystem, acknowledging the need for standardized frameworks and protocols to foster interoperability and seamless integration.

In conclusion, the edge-to-cloud continuum presents a transformative approach to meet the evolving demands of applications in various domains. By addressing the integration challenges and embracing the opportunities offered by this continuum, organizations can unlock the full potential of distributed computing, delivering enhanced user experiences and unlocking new possibilities for innovation.

Keywords: Edge analytics, Edge processing , Edge intelligence, Cloud services, Cloud applications, Edge-to-cloud integration, Data synchronization, Edge security, Edge resource management, Cloud resource allocation

1. Introduction

The Edge-to-Cloud Continuum represents a paradigm shift in the field of computing, offering a seamless integration of edge computing and cloud computing resources. This approach recognizes the diverse and dynamic nature of modern applications, where data processing and storage are distributed across a spectrum ranging from the edge of the network to centralized cloud servers. The integration of edge and cloud computing creates a continuum that optimizes the benefits of both paradigms, addressing various challenges and unlocking new opportunities for businesses and industries.

The Edge-to-Cloud Continuum involves the orchestration and collaboration of computing resources at different points in the network. At the edge, devices and sensors process data locally, reducing latency and bandwidth requirements. Meanwhile, the cloud serves as a centralized hub for storage, advanced analytics, and resource-intensive tasks. The integration allows for efficient utilization of resources, ensuring that the right computational tasks are performed at the right location in the network.

This continuum relies on several key components and technologies to enable seamless integration. Edge devices equipped with sensors and actuators play a crucial role in collecting and processing data locally. Edge computing infrastructure facilitates real-time analytics, while cloud services provide scalability, storage, and advanced processing capabilities. Connectivity technologies, such as 5G, enhance the communication between edge devices and cloud servers, ensuring a reliable and high-speed network.

While the integration of edge and cloud computing brings numerous advantages, it also presents several challenges. One major challenge is ensuring the security and privacy of data across the continuum. With data distributed across diverse locations, robust security measures must be implemented to safeguard sensitive information. Additionally, managing the heterogeneity of devices, protocols, and data formats at the edge poses interoperability challenges that need to be addressed.

The Edge-to-Cloud Continuum opens up new opportunities for innovative applications across various domains. Industries such as healthcare, manufacturing, transportation, and smart cities can benefit from real-time data processing, predictive analytics, and improved decision-making. Autonomous systems, augmented reality applications, and Internet of Things (IoT) solutions stand to gain significantly from this integrated approach.

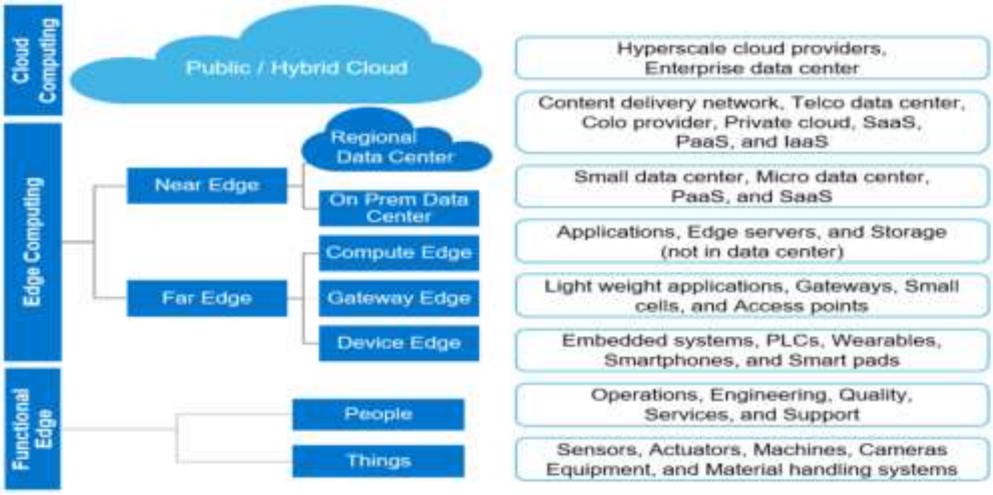


Figure 1 : Edge to cloud continuum overview

The Edge-to-Cloud Continuum represents a significant evolution in the way we design and deploy computing infrastructure. By seamlessly integrating edge and cloud resources, organizations can harness the strengths of paradigms, addressing challenges and unlocking the full potential of modern applications. As this continuum continues to evolve, it is essential to address issues related to security, interoperability, and scalability to fully realize the transformative impact on diverse industries.

2. Architecture and Models:

The Edge-to-Cloud Continuum refers to the seamless integration of computing resources and services across a spectrum ranging from edge devices to cloud infrastructure. This continuum enables the efficient processing and analysis of data at various points along the network, offering benefits such as reduced latency, improved bandwidth usage, and enhanced privacy.

The integration of edge and cloud computing presents a paradigm shift in how we process and manage data, offering opportunities for innovation across various industries. However, addressing the associated challenges is crucial to realizing the full potential of the Edge-to-Cloud Continuum.

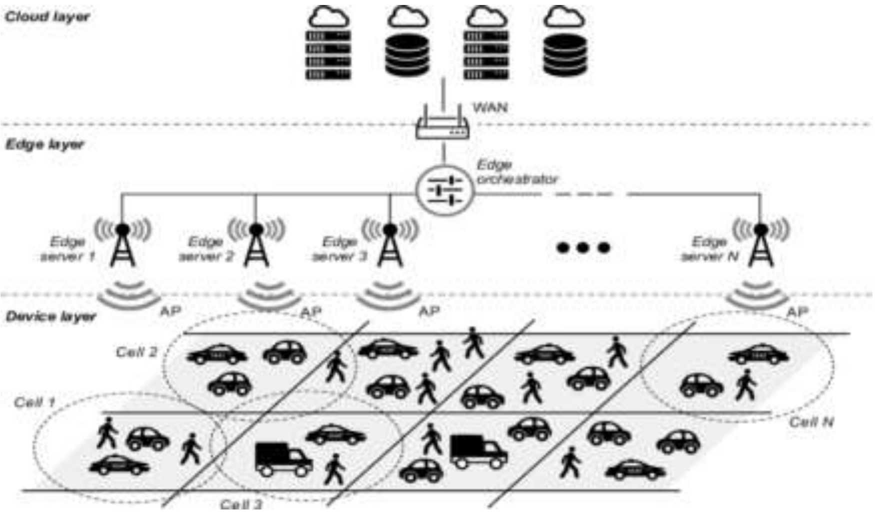


Figure 2 : The edge-cloud continuum architecture

2.1 Architecture:

1. Edge Devices:

These are the endpoints where data is generated or collected. Examples include sensors, IoT devices, and local computing devices. Edge devices perform initial data processing and filtering.

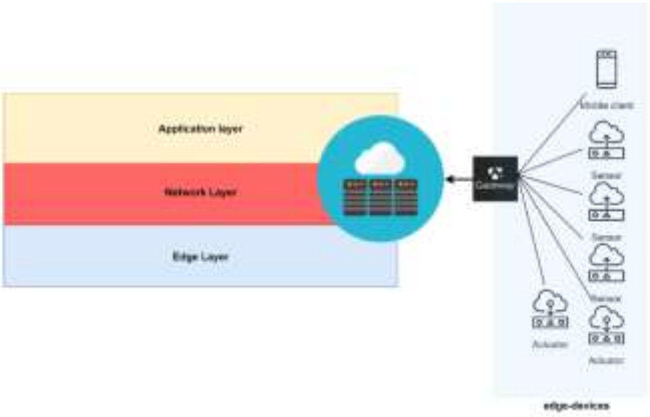


Figure 3 : Edge (device) layer components

2. Edge Computing:

Localized processing and storage that occurs near the data source. Reduces latency and bandwidth usage by handling data locally. Edge computing devices can include gateways, edge servers, and other infrastructure.

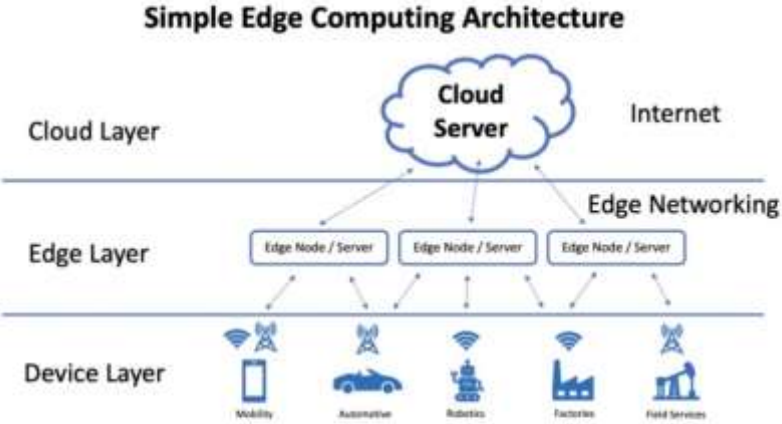


Figure 4 : Edge computing architecture

3. Fog Computing:

Extends edge computing by incorporating more powerful computing resources. Fog nodes are closer to the edge devices but can handle more complex tasks.- Enables real-time processing for applications like video analytics.

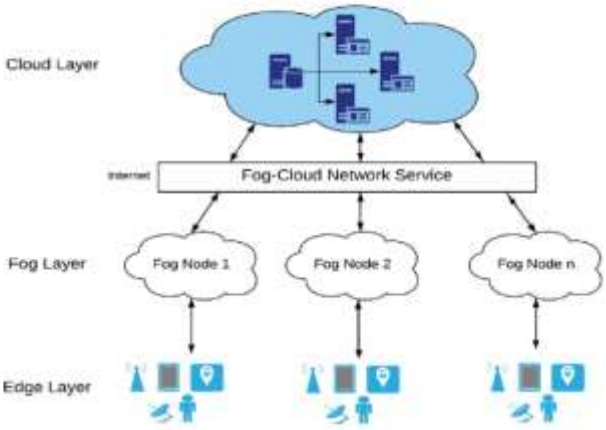


Figure 5 : Fog Computing Architecture

4. Cloud Computing:

Centralized computing resources hosted in data centers. Provides scalability, storage, and high-end processing capabilities.- Well-suited for applications that require extensive resources or historical data analysis.

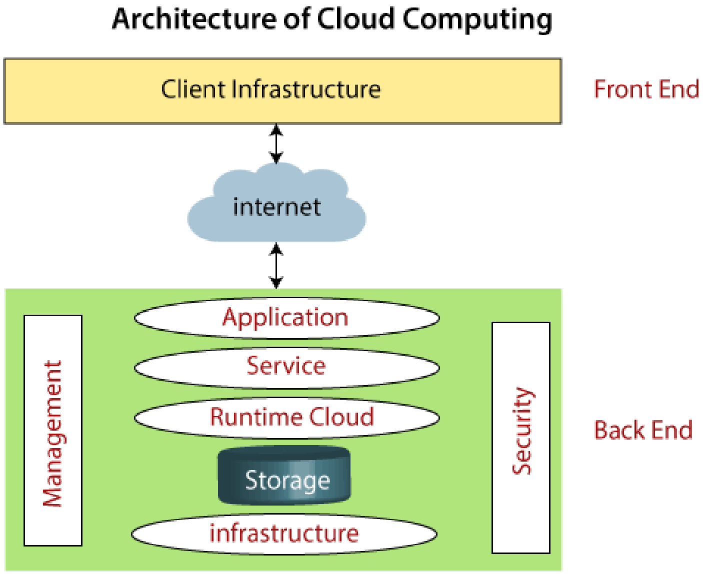


Figure 6 : Cloud Computing Architecture

5. Communication Infrastructure:

High-speed, reliable communication links between edge devices, edge computing nodes, fog nodes, and the cloud. Can include wired and wireless technologies.

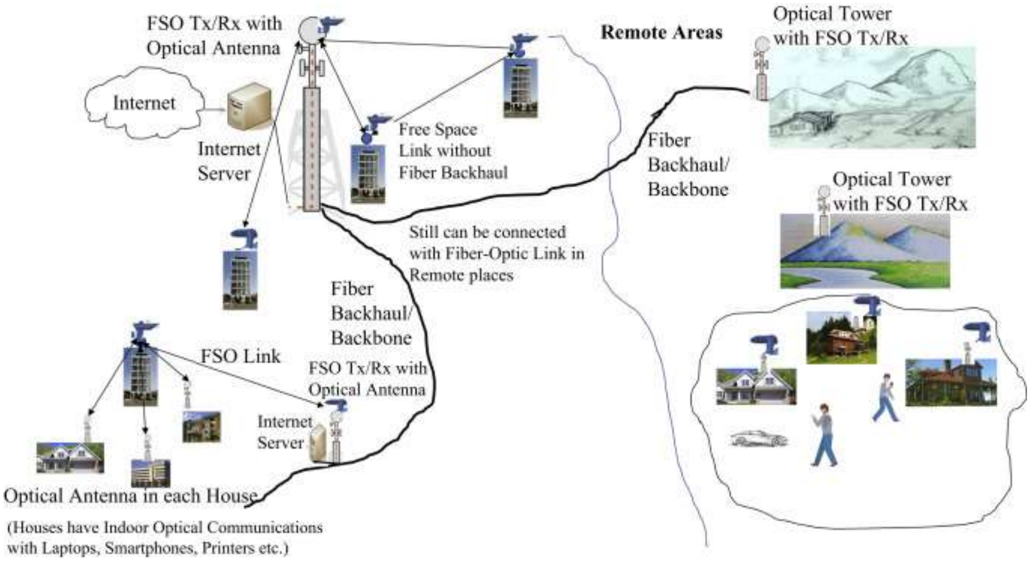


Figure 7 : Communication Architecture

2.2 Models:

1. Decentralized Model: Each layer (edge, fog, cloud) operates independently with minimal interaction.

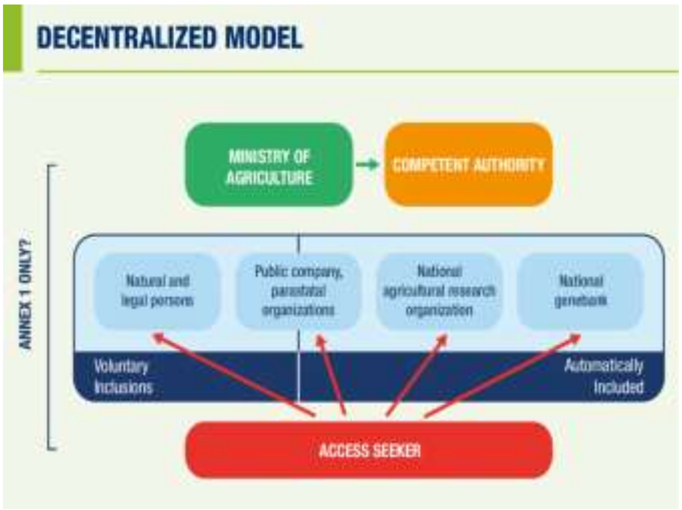
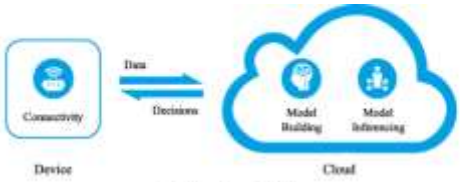


Figure 8 : Decentralized Model

2. Collaborative Model: Collaboration between edge, fog, and cloud for more efficient processing. Enables dynamic allocation of tasks based on the workload and resource availability.



(a) Cloud-based AIoT

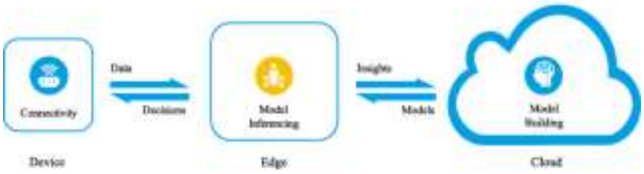


Figure 9 : An edge-cloud collaborative computing

3. Hierarchical Model: Organized in a hierarchical structure with clearly defined roles for each layer. Data processing occurs at different levels based on the complexity of the task.

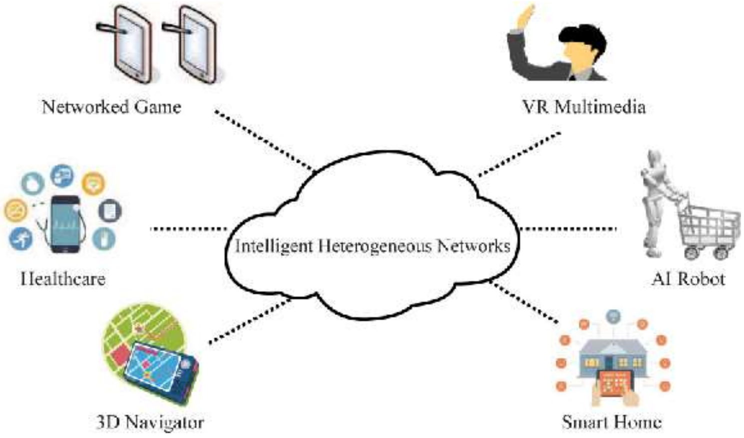


Figure 10 : Hierarchical Cloud Computing Architecture

4. Serverless Computing: Function-as-a-Service (FaaS) model where code is executed in response to events without the need for server management. Supports event-driven architectures and can be applied across the continuum.

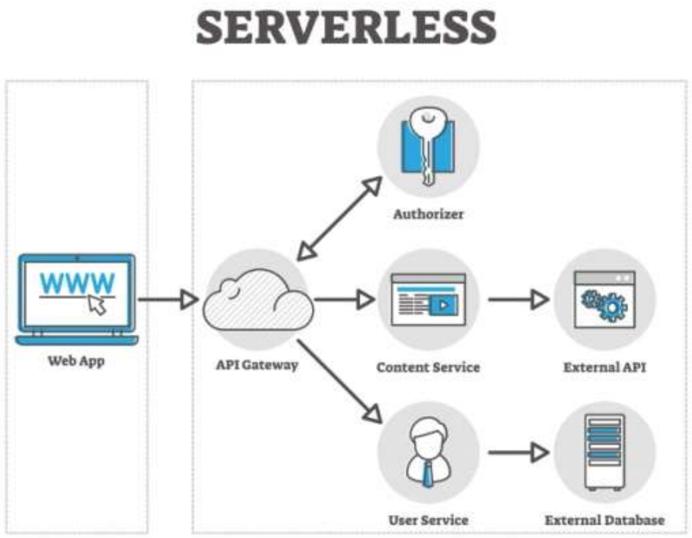


Figure 11 : Serverless Architecture

3. Establishing a Framework to Enable Continuum Computing

The scenarios outlined in the preceding section underscore the necessity of seamlessly integrating edge-to-cloud capabilities to facilitate analytics across both edge devices and distributed computing resources. Within this context, we introduce a conceptual framework in this section, embodying the concept of computing in the continuum. This framework complements the evolution of next-generation IoT systems and cyber infrastructures. Figure 12 illustrates our reference model, outlining the key research domains. The framework is structured as a four-layer architecture.

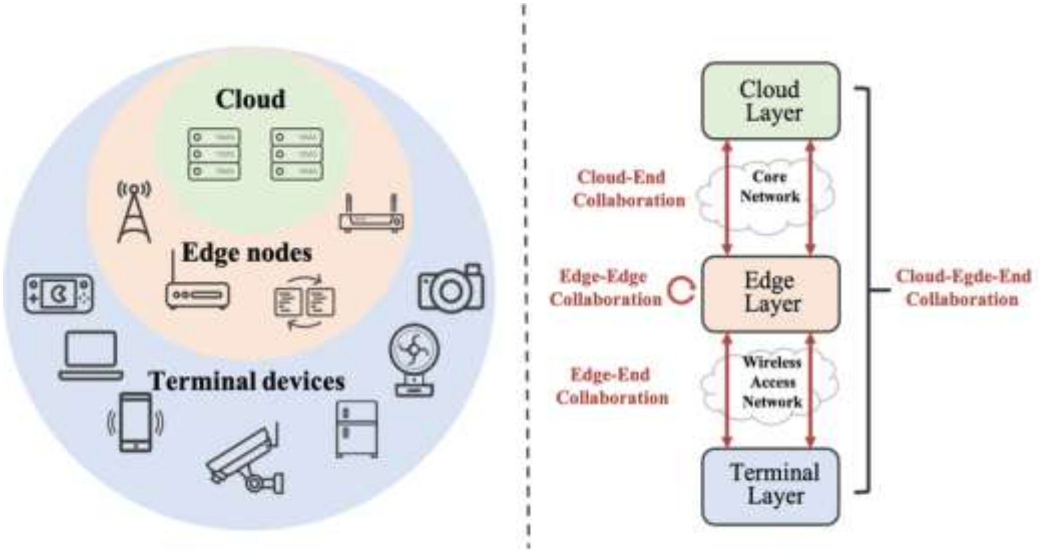


Figure 12. A comprehensive layered architecture of edge-based data-intensive IoT systems, designed to achieve computing in the continuum.

The initial layer delineates the Infrastructure, encompassing both physical elements such as sensors, specialized hardware, and data centers, as well as virtual components like virtual machines and containers. This layer facilitates the sensing and processing of data. The Federation layer coordinates the discovery, identification, and allocation of available resources across diverse hardware. Key design objectives for these layers include minimizing overhead and implementing distributed management for resource aggregation. Achieving low runtime overhead is particularly crucial when deploying on performance-constrained hardware platforms. Additionally, the distributed design of resource management components is essential for scalability with the number of applications and IoT/edge devices in the system.

The Application layer furnishes a set of operations from the application perspective, encapsulating logic for application control and resource management coordination. It encompasses the business logic of the application, overseeing transactions, and coordinating responses in the execution of its operations. The service layer comprises essential components: a rule engine, programming model, and workflow orchestrator. The rule engine empowers the creation of IoT applications capable of collecting, processing, analyzing, and acting on globally generated data without the need for infrastructure management. A rule engine applies a set of rules and facts to deduce conclusions, searching through the rules until it identifies one where the IF clause is known to be true.

The overarching goal of this framework is to simplify the utilization of heterogeneous and distributed resources through a unified programming abstraction. This abstraction enables developers to consider the 'what,' 'where,' and 'when' aspects of data collection and analysis. By bridging the gap between existing resources and science-driven problems, the framework facilitates the exploration of values across diverse research disciplines and applications. Additionally, it enables the examination of cross-cutting concerns such as quality, security, privacy, and energy consumption.

Within this framework, our research focuses on creating a data-driven programming model for analyzing requests from streaming-based workflows at runtime and determining the optimal data processing approach. We explore two primary approaches: (1) discovering and composing heterogeneous computational data pipelines by responding to the data's content and (2) conducting in-transit processing by leveraging resources at the edge, core, and along the data path, exploiting network resource capabilities. Together, these approaches dictate the required compute and network resources for efficiently executing a workflow and determine when they are needed. The subsequent sections provide a concise overview of these research endeavors.

3.1. R-Pulsar: Rule-Based Programming Abstraction for Data Pipelines

The majority of existing programming models are designed to efficiently handle batch and real-time data processing in a cloud environment. For example, MapReduce has become the standard for batch data processing in the Apache Hadoop framework, while Apache Spark (Zaharia et al., 2010) serves as a distributed batch processing framework that also supports stream processing through micro-batching. Additional frameworks such as Apache Storm (Toshniwal, 2014) for event-based stream processing and Apache Flink (Carbone et al., 2015) for unified batch and stream processing with APIs are tailored specifically for cloud environments. However, these frameworks lack control over computation placement.

In response to these challenges, R-Pulsar has been introduced to gather and analyze data for applications spanning both the cloud and the edge of the network. The goal of R-Pulsar is to establish a software stack that extends cloud capabilities to edge devices, enabling the collection and analysis of data closer to the source of information and autonomous response to local events.

Our initial investigations have centered on building upon and extending the Associative Rendezvous (AR) interaction model, adapting it to support data-driven IoT applications. The AR paradigm facilitates content-based decoupled interactions, defined in terms of semantic profiles rather than names. Programmable reactive behaviors form the core services, enabling data-driven workflow executions and decision-making. We have validated and evaluated our extensions using the previously introduced disaster recovery use case, leveraging the extended AR model to support triggered workflow topologies based on the content of data streams.

3.1.1. Content-Driven Rule-Based System

As the demand for processing larger volumes of streaming data within intricate workflows in a timely manner intensifies for IoT applications, the conventional approach becomes increasingly impractical. It becomes imperative to leverage resources closer to the edge, considering that although utilizing edge resources can mitigate data transfer costs, they are often limited and constrained in capabilities. Consequently, there is a need to strategically balance the "quality" of data processing with its immediacy and cost in a context-aware manner. To address these challenges, a rule-based system is employed, encapsulating all relevant knowledge within a set of If–Then rules.

Our content-driven rule-based system draws inspiration from the operational principles of OpenFlow (McKeown et al., 2008). It comprises a Storm data plane bolt and a singular rule table containing entries installed by the workflow developer, either during development or at runtime. Each flow-entry or rule-entry is associated with a straightforward action; our system currently supports the following actions:

1. Storing computation results at the Edge or the Cloud.

2. Triggering a new workflow topology across a set of participating nodes situated at the core or edge of the network, or routing data tuples to an already running topology.

3. Notify action, facilitating communication with any node within the overlay network and streaming the results to them.

A detailed evaluation of the scalability and overhead of the rule-based system is presented comprehensively in Gibert Renart et al. (2017). The results showcase minimal overhead, validating the feasibility of using R-Pulsar for real-time stream processing. Additionally, the system efficiently scales with the number of rules defined and evaluated, while retaining the capability to programmatically express trade-offs between data quality and computational performance. The latest version of R-Pulsar is accessible on GitHub under the Apache License 2.0.

These code snippets illustrate the utilization of R-Pulsar's API for basic resource actions, such as sensor registration and discovery (Listings 1 and 2), as well as the application of function actions for storing and triggering data-processing tasks (Listings 3 to 5).

In the first example, resource actions facilitate data exchange without prior knowledge between devices.

1) Listing 1 demonstrates a drone sensor specifying an advertising profile with the type of data it can produce, requesting notification when someone expresses interest in such data.
2) Listing 2 showcases a data consumer declaring the type of content of interest, specifying interest in any LiDAR sensor data matching the profile "Drone" and "Li*," within the specified range (40*, 70*). As this profile aligns with the previous one, the sensor from Listing 1 is notified of a consumer interested in its data, prompting the sensor to start streaming.
3) As previously mentioned, profiles can serve two purposes: discovering resources or subscribing to data publishers (resource actions) and deploying data-processing tasks across the edge and the cloud (function actions).
4) Listing 3 demonstrates the deployment of a function (post_processing_func) in the system, allowing the developer to specify where or on which set of resources it should be deployed.
5) Consequently, a profile and a decision (the IF–THEN rule) can be created to determine when to trigger the data-processing function (post_processing_func).
6) In Listing 4, the resulting action is created and attached to the function profile from Listing 5, sent when the rule's conditions are met. Listing 4 defines a rule continuously evaluated for every data element. If the condition of this rule is met, the function profile from Listing 5 is forwarded, resulting in the execution (trigger) of the data-processing task previously stored.

3.2. Comet Cloud—Federating Resources and Service Providers

CometCloud endeavors to furnish a programmable and dynamic framework capable of supporting data-driven applications. It employs software-defined environment (SDE) concepts to orchestrate the dynamic composition of infrastructure services from multiple providers. The resultant distributed software-defined environment (dSDE) autonomously evolves throughout the application life cycle, aligning with objectives and constraints established by users, applications, and/or resource providers.

The dSDE has the potential to redefine how scientists and end-users harness dynamic resources and services to construct data-driven applications and workflows. These applications can leverage the collective capabilities of a combined set of distributed and heterogeneous services, customizing execution based on the unique properties and availabilities of underlying resources. This necessitates two primary components: (1) the establishment of a control plane for the automated allocation and control of diverse services, and (2) empowering users and applications to specify criteria like performance, capacity, and Quality of Service (QoS) without in-depth knowledge of the underlying infrastructure.

To achieve these objectives, users and applications are allowed to express resource requirements programmatically. These requirements are then translated into a tailored perspective known as a "virtual slice"—a subset of services that meet the fundamental requirements specified by the user and/or application. A virtual slice is designed to evolve over time, adapting to variations in the service offering, such as changes in resource availability, as well as shifts in user and application interests.

4. Case Studies and Applications

The edge-to-cloud continuum refers to a seamless integration of computing resources and data processing capabilities across a spectrum ranging from edge devices to cloud infrastructure. This integration is crucial for optimizing the performance of various applications in different scenarios. Here are some real-world use cases and applications that benefit from the edge-to-cloud continuum, along with details on their performance in different scenarios:

The edge-to-cloud continuum refers to the seamless integration of computing resources from edge devices to cloud infrastructure, providing a unified and efficient system for processing and managing data. This continuum is crucial in various applications, offering benefits such as reduced latency, improved efficiency, and optimized resource utilization. Here are some real-world use cases and applications that leverage the edge-to-cloud continuum, along with details on their performance in different scenarios:

1. Smart Cities:

1) **Use Case:** Implementing smart city solutions involves deploying sensors and devices throughout the urban environment to collect and analyze data for various applications, such as traffic management, waste management, and environmental monitoring.
2) **Edge-to-Cloud Continuum**: Edge devices, such as traffic cameras and sensors, capture real-time data and perform initial processing locally. The processed data is then sent to the cloud for deeper analysis, long-term storage, and cross-city insights.
3) **Performance Evaluation:** The edge-to-cloud approach reduces latency for critical applications like traffic monitoring, ensuring quick response times. Cloud analytics can provide comprehensive insights for long-term urban planning.

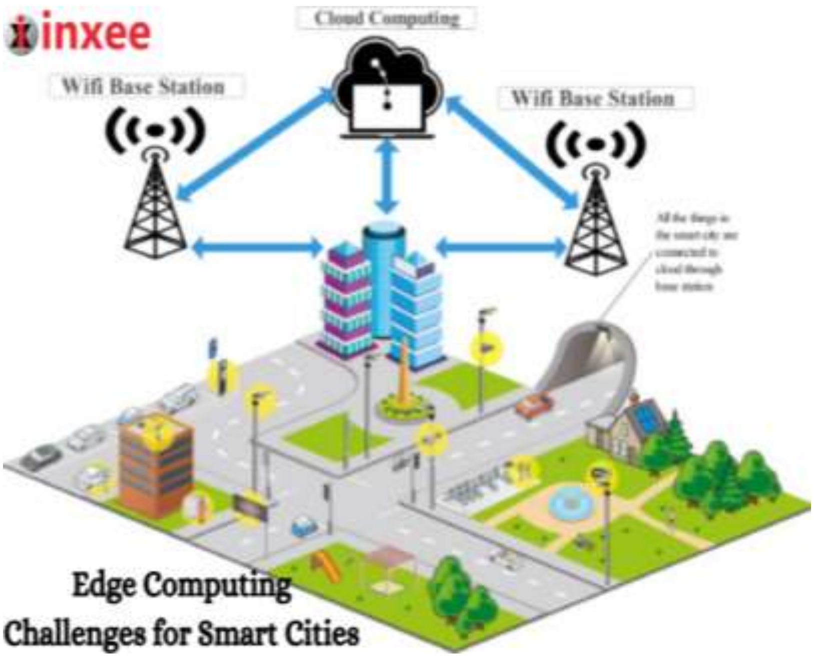


Figure 14 : Edge Computing Challenges for Smart

2. Industrial IoT (IIoT):

1) Use Case: In manufacturing, IIoT involves deploying sensors and connected devices on the factory floor to monitor equipment health, predict maintenance needs, and optimize production processes.
2) Edge-to-Cloud Continuum: Edge devices process real-time data for immediate decision-making, such as detecting equipment anomalies. The cloud is utilized for historical data analysis, predictive maintenance algorithms, and overall process optimization.
3) Performance Evaluation: Edge processing minimizes the time between data generation and action, improving response times for critical issues. Cloud analysis enables more accurate predictive maintenance models and long-term production optimization.

Figure 15 : Industrial Applications of Edge Computing

3. Healthcare:

1) Use Case: Remote patient monitoring involves wearable devices and sensors collecting health data for continuous monitoring and early detection of health issues.
2) Edge-to-Cloud Continuum: Edge devices process real-time health data locally, providing immediate feedback or triggering alerts. Cloud infrastructure is used for in-depth analysis, long-term trend monitoring, and integration with electronic health records.

3) Performance Evaluation: Edge processing ensures timely alerts for critical health events, while cloud analysis supports healthcare professionals with comprehensive insights into patient health trends.

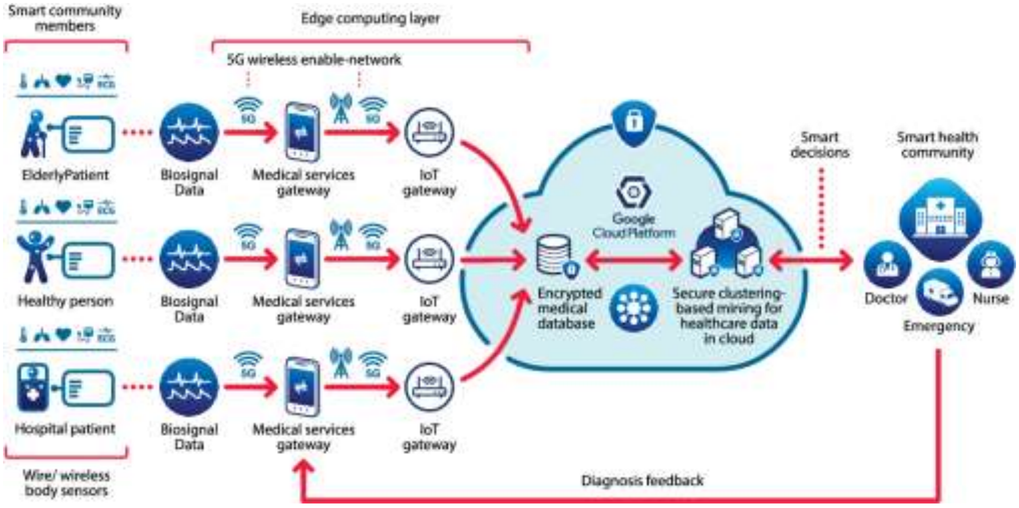


Figure 16 : Secure Edge of Things for Smart Healthcare Surveillance Framework

4. Autonomous Vehicles:

1) Use Case: Autonomous vehicles rely on a network of sensors and cameras to navigate, detect obstacles, and make real-time decisions.

2) Edge-to-Cloud Continuum: Edge computing on the vehicle processes sensor data for immediate decision-making, such as obstacle avoidance. Cloud infrastructure is utilized for high-level route planning, map updates, and collective learning from the entire fleet.

3) Performance Evaluation: Edge processing is crucial for quick response times in critical situations, while cloud analysis enhances overall fleet intelligence, contributing to better navigation and safety.

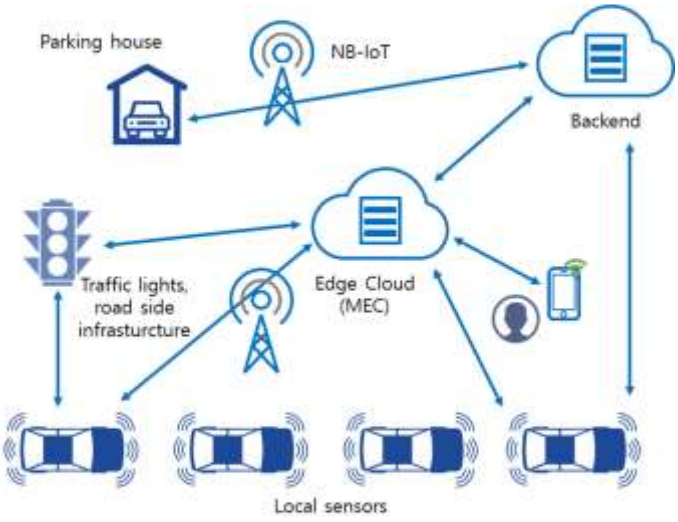


Figure 17 : Architecture of autonomous vehicles with edge computing

5. Challenges and Future Directions

The significant challenges in edge computing research, coupled with future directions and potential related categories, are detailed below:

1) Edge Computing System Service-Level Agreement (SLA): Current SLAs, tailored for cloud and network infrastructures, necessitate the exploration of new SLA designs and management techniques in edge computing. This research direction focuses on objectives such as Quality of Service (QoS) and cost optimization.

2) Bandwidth-Saving Edge Computing System: An essential aspect of edge computing involves diverting data traffic from the core. Currently, only a limited number of studies have delved into the design of bandwidth-aware edge computing systems and the measurement of actual bandwidth usage. Addressing this challenge may involve solutions like scheduling, load balancing, and resource analysis.
3) Green Edge Computing: Few works comprehensively address the overall energy efficiency of edge computing systems. Future exploration could focus on strategies to reduce energy consumption during computation offloading, mobility management, and the co-deployment of Internet of Things (IoT) and fog technologies.
4) Edge Computing Node Site Selection: Limited attention has been devoted to determining suitable locations for edge computing node deployments. Resolving this issue requires considering communication, storage, computation, and cost factors.
5) Security: Edge computing nodes face a higher likelihood of site attacks compared to cloud data centers. Future research could explore methods to guarantee the security of nodes against physical damage and disturbances. Additionally, designing robust access control protocols for edge computing nodes to implement isolation/sandboxing is a potential area of investigation.
6) Network Slicing: While prevalent network-slicing designs are primarily business-driven, creating end-to-end network slicing within the context of edge computing requires an understanding of the impact of radio transmission and edge computing characteristics. The integration of Network Function Virtualization (NFV), Software-Defined Networking (SDN), and edge computing is crucial in this future direction.
7) End-to-End Tradeoffs of Architectures: Designing improved end-to-end systems to implement greater tradeoffs among global centralized and local distributed architectures is another potential future direction. Establishing logical edge computing system topologies, either statically or dynamically based on the common physical network, can provide solutions across a spectrum of architectures ranging from fully centralized to fully distributed.

6. Conclusion

Fog computing aims to establish a system facilitating cloud-to-thing service connectivity and working collaboratively with the cloud. Simultaneously, Mobile Edge Computing (MEC) is recognized as a pivotal technology within the 5G system, while a cloudlet represents a micro-data center strategically deployed in close proximity. While fog computing primarily emphasizes IoT deployment scenarios, MEC predominantly offers innovative mobile Radio Access Network (RAN) applications for 5G systems. In contrast, cloudlets offload computing power at the network edge. Although each concept serves distinct purposes, the overarching objective of fog computing, MEC, and cloudlets is to create application-centric environments that streamline automated application deployment, management, and business provisioning.

In the current landscape, foundational hardware infrastructure and software technologies for edge computing have significantly matured. However, existing co-deployment solutions heavily depend on accurate modeling and prediction of service response times, network fluctuations, request arrival patterns, etc., posing challenges in achieving efficient scheduling decisions in real-world scenarios. Consequently, there is an immediate need to adapt or optimize hardware and software to meet the specific requirements of edge computing. This encompasses various aspects, including the computing capacity of edge nodes, performance optimization, reliability, and disaster recovery of edge nodes. Additionally, there is a demand for intelligent scheduling of edge computing tasks, unified management of heterogeneous edge nodes, development of data distribution mechanisms, and ensuring consistency.

Furthermore, areas requiring further research include the edge computing network architecture, large-scale edge applications and services, and edge functions and technologies (e.g., data granularity, video compression, and analytics). Addressing these aspects will contribute to advancing the field and enhancing the efficiency and effectiveness of edge computing systems in diverse real-world scenarios.

References:

[1]. Cisco, Cisco Annual Internet Report (2018–2023) White Paper, 2020. https://www.cisco.com/c/en/us/solutions/collateral/ executive-perspectives/annual-internet-report/ white-paper-c11-741490.pdf

[2]. S. Saleem et al., "Building Smart Cities Applications based on IoT Technologies: A Review", Technology Reports of Kansai University, vol. 62, no. 3, pp. 1083–1092, 2020.

[3]. ETSI, "Developing Software for Multi-access Edge Computing", white paper, 2019. https://www.etsi.org/images/files/ETSIWhitePapers/ etsi_wp20ed2_MEC_SoftwareDevelopment.pdf

[4]. ETSI GS MEC 026, "Multi-access Edge Computing (MEC)"; Support for regulatory requirements V2.1.1, whiter paper, 2019. https://www.etsi.org/deliver/etsi_gs/MEC/ 001_099/026/02.01.01_60/gs_MEC026v020 101p.pdf

[5]. ETSI, "Multi-access Edge Computing (MEC): Study on Inter-MEC systems and MEC-Cloud systems coordination", whiter paper, 2021. https://www.etsi.org/deliver/etsi_gr/MEC/ 001_099/035/03.01.01_60/gr_MEC035v03 0101p.pdf

[6]. K. Adlen and F. Pantelis, "Toward Slicing-Enabled Multi-Access Edge Computing in 5G", IEEE Network, vol. 34, no. 2, pp. 99–105, 2020. http://dx.doi.org/10.1109/MNET.001.1900261

[7]. Z. Kamran et al., "Service-Less Video Multicast in 5G: Enablers and Challenges", IEEE Network, vol. 34, no. 3, pp. 270–276, 2020. http://dx.doi.org/10.1109/MNET.001.1900435

[8]. M. Wojciech et al., "Challenges and Novel Solutions for 5G Network Security, Privacy and Trust", IEEE Wireless Communications, vol. 27, no. 4, pp. 6–7, 2020. http://dx.doi.org/10.1109/MWC.2020.9170261

[9]. J. Sunmi et al., "Ultra Low Latency Services in 5G Systems: A Perspective from 3GPP Standards", Etri Journal, vol. 42, no. 5, pp. 721–733, 2020. http://dx.doi.org/10.4218/etrij.2020-0200

[10]. M. Angel et al., "MEC for Fair, Reliable and Efficient Media Streaming in Mobile Networks", IEEE Transactions on Broadcasting, vol. 66, no. 2, pp. 264–278, 2020. http://dx.doi.org/10.1109/TBC.2019.2954097

[11]. A. Nadine and D. Nikos, "Mobility Management Solutions for 5G Networks: Architecture and Services", Computer Networks, vol. 169, 2020. http://dx.doi.org/10.1016/j.comnet.2019.107082

[12]. D. Ramraj et al., "Study and Investigation on 5G Technology: A Systematic Review", Sensors, vol. 22, no. 1, 2021. http://dx.doi.org/10.3390/s22010026

[13]. M. Ismael et al., "Design, Resource Management and Evaluation of Fog Computing Systems: A Survey", IEEE Internet of Things Journal, 2020. http://dx.doi.org/10.1109/JIOT.2020.3022699

[14]. M. Andras and K. Attila, "A Survey and Taxonomy of Simulation Environments Modelling Fog Computing. Simulation Modelling Practice and Theory", 2019. http://dx.doi.org/10.1016/j.simpat.2019.102042

[15]. P. Carlo et al., "Fog Computing for the Internet of Things: A Survey", ACM Transactions on Internet Technology, vol. 19, no. 2, 2019. http://dx.doi.org/10.1145/3301443

[16]. S. Svorobej et al., "Simulating Fog and Edge Computing Scenarios: An Overview and Research Challenges", Future Internet, vol. 11, 2019. http://dx.doi.org/10.3390/fi11030055

[17]. K. Tange et al., "A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities", IEEE Communications Surveys & Tutorials, vol. 22, pp. 2489–2520, 2020.http://dx.doi.org/10.1109/COMST.2020.3011208

[18]. G. Fersi, "Fog Computing and Internet of Things in one Building Block: A Survey and an Overview of Interacting Technologies", Cluster Computing, vol. 24, no. 2, 2021. http://dx.doi.org/10.1007/s10586-021-03286-4

[19]. M. Rashid and U. Wani, "Role of Fog Computing Platform in Analytics of Internet of Things - Issues, Challenges and Opportunities", John Wiley & Sons, Ltd , pp. 209–220, 2020. http://dx.doi.org/10.1002/9781119670087.ch12

[20]. K. M. Sadique et al., "Fog Computing for Trust in the Internet of Things (IoT): A Systematic Literature Review", in Proc. of the 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA). IEEE, 2020. http://dx.doi.org/10.1109/ICCSEA49143.2020.9132861

[21]. M. Kaur and R. Aron, "A Systematic Study of Load Balancing Approaches in the Fog Computing Environment", The Journal of Supercomputing, vol. 77, no. 6, 2021. http://dx.doi.org/10.1007/s11227-020-03600-8

[22]. K. Gasmi et al., "A Survey on Computation Offloading and Service Placement in Fog Computing-based IoT", The Journal of Supercomputing, vol. 78, no. 4, pp. 1–32, 2021. http://dx.doi.org/10.1007/s11227-021-03941-y

[23]. M. Ghobaei-Arani et al., "Resource Management Approaches in Fog Computing: A Comprehensive Review", Journal of Grid Computing, vol. 18, no. 1, 2020. http://dx.doi.org/10.1007/s10723-019-09491-1

[24]. M. Babar et al., "Cloudlet Computing: Recent Advances, Taxonomy, and Challenges", IEEE Access, vol. 9, pp. 29609–29622, 2021. http://dx.doi.org/10.1109/ACCESS.2021.3059072

[25]. L. Tian and X. Zhong [25] H. Elazhary, "Internet of Things (IoT), Mobile Cloud, Cloudlet, Mobile IoT, IoT Cloud, Fog, Mobile Edge, and Edge Emerging Computing Paradigms: Disambiguation and Research Directions", Journal of Network and Computer Applications, vol. 128, pp. 105–140, 2019. http://dx.doi.org/10.1016/j.jnca.2018.10.021