IoT Edge Challenges and Functions
draft-hong-t2trg-iot-edge-computing-02

Abstract

Many IoT applications have requirements that cannot be met by the traditional Cloud (aka Cloud computing). These include time sensitivity, data volume, uplink cost, operation in the face of intermittent services, privacy and security. As a result, the IoT is driving the Internet toward Edge computing. This document outlines the requirements of the emerging IoT Edge and its challenges. It presents a snapshot of the state-of-the-art, a general model, and major components of the IoT Edge, with the goal to provide a common base for future discussions in T2TRG and other IETF WGs and RGs.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 20 July 2020.
Table of Contents

1. Introduction ................................................. 3
2. Conventions and Terminology ................................. 4
3. Background .................................................... 4
  3.1. Internet of Things (IoT) ................................. 4
  3.2. Cloud Computing .......................................... 4
  3.3. Edge Computing ........................................... 5
  3.4. Example of IoT Edge Computing Use Cases ............... 6
    3.4.1. Smart Construction ................................. 6
    3.4.2. Smart Grid ........................................... 7
    3.4.3. Smart Water System ................................. 7
  3.5. Common Aspects of Current IoT Edge Computing Service
       Platforms .................................................. 8
4. Challenges for IoT and Impacts of Edge Computing ............ 9
  4.1. Time Sensitivity ......................................... 9
  4.2. Uplink Cost .............................................. 10
  4.3. Resilience to Intermittent Services ........................ 10
  4.4. Privacy and Security ..................................... 10
5. IoT Edge Computing Functions ................................... 11
  5.1. OAM Components ......................................... 13
    5.1.1. Resources Discovery ................................. 13
    5.1.2. Virtualization Management .......................... 13
    5.1.3. Authentication and Authorization .................... 13
    5.1.4. Edge Organization and Federation ................. 14
  5.2. Functional Components ................................... 14
    5.2.1. External APIs ....................................... 14
    5.2.2. Communication Brokering ............................ 14
    5.2.3. In-Network Computation .............................. 15
    5.2.4. Edge Caching ......................................... 15
    5.2.5. Other Services ....................................... 15
  5.3. Application Components .................................. 15
    5.3.1. IoT End Devices Management ........................ 15
    5.3.2. Data Management .................................... 16
Currently, many IoT services leverage the Cloud, since it can provide virtually unlimited storage and processing power. The reliance of IoT on back-end cloud computing brings additional advantages such as flexibility and efficiency. Today's IoT systems are fairly static with respect to integrating and supporting computation. It’s not that there is no computation, but systems are often limited to static configurations (edge gateways, the Cloud).

However, IoT devices are creating vast amounts of data at the network edge. To meet IoT use case requirements, that data increasingly is being stored, processed, analyzed, and acted upon close to the data producers. These requirements include time sensitivity, data volume, uplink cost, resiliency in the face of intermittent connectivity, privacy, and security, which cannot be addressed by today’s centralized cloud computing. These requirements suggest a more flexible way to distribute computing (and storage) and to integrate it in the edge-cloud continuum. We will refer to this integration of edge computing and IoT as "IoT edge computing". Our draft describes uses cases, challenges, a proposed system model and derived functional components.
2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Background

3.1. Internet of Things (IoT)

Since the term "Internet of Things" (IoT) was coined by Kevin Ashton in 1999 working on Radio-Frequency Identification (RFID) technology at the Auto-ID Center of the Massachusetts Institute of Technology (MIT) [Ashton], the concept of IoT had been broadened to reflect the vision of connecting the physical world to the virtual world of computers using (wireless) sensor networks with any kind of technology with which the Things can send and receive information without human intervention. Recently, the term has become more literal by actually connecting Things to the Internet and converging on Internet and also Web technology. Things are usually embedded systems of various kinds, such as home appliances, mobile equipment, wearable devices, etc. Things are widely distributed, but typically have limited storage and processing power, which raise concerns regarding reliability, performance, energy consumption, security, and privacy [Lin].

3.2. Cloud Computing

Cloud computing has been defined in [NIST]: "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". Cloud computing has been a predominant technology which has virtually unlimited capacity in terms of storage and processing power. The availability of virtually unlimited storage and processing capabilities at low cost enabled the realization of a new computing model, in which virtualized resources can be leased in an on-demand fashion, being provided as general utilities. Companies like Amazon, Google, Facebook, etc. widely adopted this paradigm for delivering services over the Internet, gaining both economical and technical benefits [Botta].

Now with IoT, we will reach the era of post-clouds where unprecedented volume and variety of data will be generated by things at edge/local networks and many applications will be deployed on the edge networks to consume these IoT data. Some of the applications may need very short response times, some may contain personal data,
and others may generate vast amounts of data. Today’s cloud-based service models are not suitable for these applications.

It is predicted that by 2019, 45% of the data created in IoT will be stored, processed, analyzed and acted close to, or at the edge of the network and about 50 billion devices will connect to the Internet by 2020 [Evans]. So, moving all data from edge/local networks to the cloud data center may not be an efficient way anymore to process vast amounts of data.

In cloud computing, users traditionally only consumed IoT data through cloud services. Now, however, users are also producing IoT data with their mobile devices. This change requires even more functionality at edge/local networks [Shi], to support mobile edge computing considerations.

3.3. Edge Computing

Edge computing, under certain aspects also referred to as fog computing, is a new paradigm in which substantial computing and storage resources are placed at the edge of the Internet, that is, in close proximity to mobile devices, sensors, actuators, or machines, so that computing happens near data sources [Mahadev], or closer to where decisions or interactions with the physical world are happening. It works on both downstream data on behalf of cloud services and upstream data on behalf of IoT services.

An edge device is any computing or networking resource residing between data sources and cloud-based datacenters. In edge computing, end devices not only consume data, but also produce data. And at the network edge, devices not only request services and information from the cloud, but also handle computing tasks including processing, storage, caching, and load balancing on data sent to and from the cloud [Shi].

The definition of edge computing from ISO is a "form of distributed computing in which significant processing and data storage takes place on nodes which are at the edge of the network" [ISO_TR]. ETSI’s definition of multi-access edge computing is a "system which provides an IT service environment and cloud-computing capabilities at the edge of an access network which contains one or more type of access technology, and in close proximity to its users" [ETSI_MEC_01].

The similar concept of fog computing from the Industrial Internet Consortium (formerly OpenFog) is "a horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing
continuum" Appendix A.3.3. The term fog computing usually represents the notion of a multi-tiered edge computing, that is, several layers of compute infrastructure between the end devices and the cloud.

Based on these definitions, we can summarize a general philosophy of edge computing as to distribute the required functions close to users and data, while the difference to classic local systems is the usage of management and orchestration features adopted from cloud computing.

Actors from various industries approach edge computing using different terms and reference models, although in practice these approaches are not incompatible and may integrate with each other:

* The telecommunication industry tends to use a model where edge computing services are deployed over NFV infrastructure at aggregation points, or in proximity to the user equipment (e.g., gNodeBs) Appendix A.3.1.

* Enterprise and campus solutions often interpret edge computing as an "edge cloud", that is, a smaller data center directly connected to the local network (often referred to as "on-premise").

* The automation industry defines the edge as the connection point between IT from OT (Operational Technology). Hence, here edge computing sometimes refers to applying IT solutions to OT problems such as analytics, more flexible user interfaces, or simply having more compute power than an automation controller.

It is clear that the combination of these models leads to a multi-tier edge computing solution as mentioned above.

3.4. Example of IoT Edge Computing Use Cases

IoT edge computing can be used in home, industry, grid, healthcare, city, transportation, agriculture, and/or education scenarios. We discuss here only a few examples of such use cases, to point out differentiating requirements.

3.4.1. Smart Construction

In traditional construction domain, heavy equipment and machinery pose risks to humans and property. Thus, there have been many attempts to deploy technology to protect lives and property in construction sites. For example, measurements of noise, vibration, and gas can be recorded on a remote server and reported to an inspector. Today, data produced by such measurements is collected by a local gateway and transferred to a remote server. This incurs
transmission costs, e.g., over a LTE connection, and storage costs, e.g., when using Amazon Web Services. When an inspector needs to investigate an incident, he checks the information stored on a server.

If we leverage IoT edge computing, sensor data can be processed and analyzed on a gateway located within or near a construction site. And with the help of statistical analysis or machine learning technologies, we can predict future incidents in advance and this prediction can trigger an on-site alarm and a notification to an inspector.

To determine the exact cause of an incident, sensor data including audio and video are transferred to a remote server. In this case, audio and video data volume is typically very large and the cost of transmission can be an issue. Edge computing can be leveraged to predict the time of an incident, which can reduce the volume of transmitted data; while during a normal time period audio and video data may be transmitted with a low resolution, during an emergency, this transmission may use a high resolution. This adjustment can reduce transmission cost significantly.

3.4.2. Smart Grid

In future smart city scenarios, the Smart Grid will be critical in ensuring highly available/efficient energy control in city-wide electricity management. Edge computing is expected to play a significant role in those systems to improve transmission efficiency of electricity; to react and restore power after a disturbance; to reduce operation costs and reuse renewable energy effectively, since these operations involve local decision making. In addition, edge computing can help monitoring power generation and power demand, and making electrical energy storage decisions in the smart grid system.

3.4.3. Smart Water System

The water system is one of the most important aspects of a city. Effective use of water, and cost-effective and environment-friendly water treatment are critical aspects of this system. They can be facilitated by edge computing in smart water systems, to help monitor water consumption, transportation and prediction of future water use. For example, water harvesting and ground water monitoring will be supported through edge computing. Edge computing will also enable locally analyzing collected information related to water control and management, and limit water losses.
3.5. Common Aspects of Current IoT Edge Computing Service Platforms

This section provides an overview of today’s IoT edge computing field, based on a limited review of standards, research, open-source and proprietary products in Appendix A.

IoT gateways (Appendix A.2.1, Appendix A.1.1) represent a common class of IoT edge computing products, where the gateway is providing a local service on customer premises, and is remotely managed through a cloud service. IoT communication protocols are typically used between IoT devices and the gateway, including CoAP, MQTT and many specialized IoT protocols (such as OPC UA and DDS in the Industrial IoT space), while the gateway communicates with the distant cloud typically using HTTPS. Virtualization platforms enable the deployment of virtual edge computing functions (as VMs, application containers, etc.), including IoT gateway software, on servers in the mobile network infrastructure (at base station and concentration points), in edge datacenters (in central offices) or regional datacenters located near central offices. End devices are envisioned to become computing devices in forward looking projects, but are not commonly used as such today.

Physical or virtual IoT gateways can host application programs, which are typically built using an SDK to access local services through a programmatic API. Edge cloud system operators host their customers’ applications VMs or containers on servers located in or near access networks, which can implement local edge services. For example, mobile networks can provide edge services for radio network information, location and bandwidth management.

Life cycle management of services and applications on physical IoT gateways is often cloud-based. Edge cloud management platforms and products (Appendix A.1.2, Appendix A.2.2) adapt cloud management technologies (e.g., Kubernetes) to the edge cloud, i.e., to smaller, distributed computing devices running outside a controlled data center. Services and application life-cycle is typically using a NFV-like management and orchestration model.

The platform typically includes services to advertise or consume APIs (e.g., Mp1 interface in ETSI MEC supports service discovery and communication), and enables communicating with local and remote endpoints (e.g., message routing function in IoT gateways). The service platform is typically extensible by edge applications, since they can advertise an API that other edge applications can consume. IoT communication services include protocols translation, analytics and transcoding. Communication between edge computing devices is enabled in tiered deployments or distributed deployments.
An edge cloud platform may enable pass-through without storage or local storage (e.g., on IoT gateways). Some edge cloud platforms use a distributed form of storage such as an ICN network (e.g., NFN nodes can store data in NDN, Appendix A.4.1) or a distributed storage platform (e.g., such as Ceph, Appendix A.1.2). External storage, e.g., on databases in distant or local IT cloud, is typically used for filtered data deemed worthy of long term storage, or in some cases for all data, for example when required for regulatory reasons.

Stateful computing is supported on platforms hosting native programs, VMs or containers. Stateless computing is supported on platforms providing a "serverless computing" service (a.k.a. function-as-a-service), or on systems based on named function networking.

In many IoT use cases, a typical network usage pattern is high volume uplink with some form of traffic reduction enabled by processing over edge computing devices. Alternatives to traffic reduction include deferred transmission (to off-peak hours or using physical shipping). Downlink traffic includes application control and software updates. Other, downlink-heavy traffic patterns are not excluded but are more often associated with non-IoT usage (e.g., video CDNs).

4. Challenges for IoT and Impacts of Edge Computing

As the IoT is maturing, systems are converging, deployments are growing, and IoT technology is used with more and more demanding applications such as industrial, automotive, or healthcare. This leads to new challenges for the network infrastructure. In particular, the amount of data created at the edge is expected to be vast. Industrial machines such as laser cutters already produce over 1 terabyte per hour, the same applies for autonomous cars [NVIDIA]. 90% of IoT data is expected to be stored, processed, analyzed, and acted upon close to the source [Kelly], as cloud computing models alone cannot address the new challenges [Chiang]. Below we discuss IoT use case requirements that are moving cloud capabilities to be more proximate and more distributed and disaggregated.

4.1. Time Sensitivity

Many industrial control systems, such as manufacturing systems, smart grids, oil and gas systems, etc., often require stringent end-to-end latency between the sensor and control node. While some IoT applications may require latency below a few tens of milliseconds [Weiner], industrial robots and motion control systems have use cases for cycle times in the order of microseconds [60802]. An important aspect for real-time communications is not only the latency, but also guarantees for jitter. This means control packets need to arrive with as little variation as possible with a strict deadline. Given
the best-effort characteristics of the Internet, this challenge is virtually impossible to address, without comprehending end-to-end guarantees for individual message delivery and continuous data flows.

4.2. Uplink Cost

Many IoT deployments are not challenged by a constrained network bandwidth to the cloud. The fifth generation mobile networks (5G) and Wi-Fi 6 both theoretically top out at 10 gigabits per second (i.e., 4.5 terabyte per hour), which enables high-bandwidth uplinks. However, the resulting cost for high-bandwidth connectivity to upload all data to the cloud is unjustifiable and impractical for most IoT applications. In some settings, e.g. in aeronautical communication, higher communication costs reduce the amount of data that can be practically uploaded even further.

4.3. Resilience to Intermittent Services

Many IoT devices such as sensors, data collectors, actuators, controllers, etc. have very limited hardware resources and cannot rely solely on their limited resources to meet all their computing and/or storage needs. They require reliable, uninterrupted or resilient services to augment their capabilities in order to fulfill their application tasks. This is hard and partly impossible to achieve with cloud services for systems such as vehicles, drones, or oil rigs that have intermittent network connectivity.

4.4. Privacy and Security

When IoT services are deployed at home, personal information can be learned from detected usage data. For example, one can extract information about employment, family status, age, and income by analyzing smart meter data \[\text{[ENERGY]}\]. Policy makers started to provide frameworks that limit the usage of personal data and put strict requirements on data controllers and processors. However, data stored indefinitely in the cloud also increases the risk of data leakage, for instance, through attacks on rich targets.

Industrial systems are often argued to not have privacy implications, as no personal data is gathered. Yet data from such systems is often highly sensitive, as one might be able to infer trade secrets such as the setup of production lines. Hence, the owner of these systems are generally reluctant to upload IoT data to the cloud.
5. IoT Edge Computing Functions

Edge computing is expected to play an important role in deploying new IoT services integrated with Big Data and AI, enabled by flexible in-network computing platforms. Although there are lots of approaches to edge computing, we attempt to lay out a general model and list associated logical functions in this section. In practice, this model can map to different architectures, such as:

* A single IoT gateway, or a hierarchy of IoT gateways, typically connected to the cloud (e.g., to extend the traditionally cloud-based management of IoT devices and data to the edge). A common role of an IoT Gateway is to provide access to an heterogeneous set of IoT devices/sensors; handle IoT data; and deliver IoT data to its final destination in a cloud network. Whereas an IoT gateway needs interactions with cloud like as conventional cloud computing, it can also operate independently.

* A set of distributed computing nodes, e.g., embedded in switches, routers, edge cloud servers or mobile devices. In the future, some IoT end devices may have enough computing capabilities to participate in such distributed systems. In this model, edge computing nodes can collaborate with each other to share their resources.
In this general model, the edge computing domain is interconnected with IoT end devices (southbound connectivity) and possibly with a remote/cloud network (northbound connectivity), and with a service operator’s system. Edge computing nodes provide multiple logical functions, or components, which may not all be present in a given system. They may be implemented in a centralized or distributed fashion, in the edge network, or through some interworking between the edge network and a remote cloud network.

We now attempt to enumerate major edge computing domain components. They are here loosely organized into OAM, functional and application components, with the understanding that the distinction between these
classes may not always be clear, depending on actual system architectures.

5.1. OAM Components

Edge computing OAM goes beyond the network-related OAM functions listed in [RFC6291]. Besides infrastructure (network, storage and computing resources), edge computing systems can also include computing environments (for VMs, software containers, functions), IoT end devices, data, and code.

Operation related functions include performance monitoring for service level agreement measurement; fault management and provisioning for links, nodes, compute and storage resources, platforms and services. Administration covers network/compute/storage resources, platforms and services discovery, configuration and planning. Management covers monitoring and diagnostics of failures, as well as means to minimize their occurrence and take corrective actions. This may include software updates management, high service availability through redundancy and multipath communication.

We further detail a few OAM components.

5.1.1. Resources Discovery

This component is about finding infrastructure resources, such as compute, network and storage, but also other resources such as IoT end devices, sensors, data, code, or services.

5.1.2. Virtualization Management

Some IoT edge computing systems make use of virtualized (compute, storage and networking) resources, which need to be allocated and configured.

5.1.3. Authentication and Authorization

This can cover authenticating platforms, end devices, data, code units and applications or users interacting with the system. Today, centralized gateway-based systems rely for device authentication on the installation of a secret on IoT end devices and on computing devices (e.g., a device certificate stored in a hardware security module).
5.1.4. Edge Organization and Federation

In a distributed system context, once edge devices have discovered and authenticated each other, they can be organized, or self-organize, into hierarchies or clusters. Such groups can also form federations with other edge or remote clouds.

5.2. Functional Components

5.2.1. External APIs

An IoT edge cloud may provide a northbound data plane or management plane interface to a remote network, e.g., a cloud, home or enterprise network. This interface does not exist in standalone (local-only) scenarios. To support such an interface when it exists, an edge computing component needs to expose an API, deal with authentication and authorization, support secure communication.

An IoT edge cloud may provide an API or interface to local users (e.g., to facilitate local management), or to mobile users (e.g., to provide access to services and applications, or to manage data published by the mobile device).

5.2.2. Communication Brokering

A typical function of IoT edge computing is to facilitate communication with IoT end devices: for example, enable clients to register as recipients for data from devices, as well as forwarding/routing of traffic to or from IoT end devices, enabling various data discovery and redistribution patterns, e.g., north-south with clouds, east-west with other edge devices [DATA-DISCOVERY]. Another aspect of a communication component is dispatching of alerts and notifications to interested consumers both inside and outside of the edge computing domain. Protocol translation, analytics and transcoding may also be performed when necessary.

Communication brokering may be centralized in some systems, e.g., using a hub-and-spoke message broker, or distributed like with message buses, possibly in a layered bus approach. Distributed systems may leverage direct communication between end devices and communication devices, such as device-to-device links. Brokers functions can include ensuring communication reliability, traceability, and in some cases transaction management.

QoS can be provided in some systems through the combination of network QoS (e.g., traffic engineering or wireless resource scheduling) and compute/storage resource allocations. In some systems a bandwidth manager service can be exposed to enable
allocation of bandwidth to/from an edge computing application instance.

5.2.3. In-Network Computation

A core function of IoT edge computing is to enable computation offloading, i.e., to perform computation on an edge node on behalf of a device or user. The support for in-network computation may vary in term of capability, e.g., computing nodes can host a virtual machine able run stateful or stateless code, or a rule engine providing an API to register actions in response to conditions such as IoT device ID, sensor values to check, thresholds, etc. Computation offloading includes orchestration or application lifecycle related aspects, such as: selecting an appropriate compute device based on available resources, compute node properties, etc., and with varying goals including for example load balancing and energy conservation; onboarding code on a platform or compute device; assisted or automatic partitioning of code; invoking remote code execution; relocating an instance from one compute node to another.

5.2.4. Edge Caching

A purpose of local caching may be to enable local data processing (e.g., pre-processing or analysis), or to enable delayed virtual or physical shipping. A responsibility of the edge caching component is to manage data persistence, e.g., to schedule removal of data when it is no longer needed. Another aspect of this component may be to authenticate and encrypt data. It can for example take the form of a distributed storage system, and deal with related issues, e.g., reaching and maintaining data consistency; enabling efficient access to data, for example using some form of sharding.

5.2.5. Other Services

Data generated by IoT devices and associated information obtained from the access network may be used to provide high level services such as end device location, radio network information and bandwidth management.

5.3. Application Components

5.3.1. IoT End Devices Management

IoT end device management includes managing information about the IoT devices, including their sensors, how to communicate with them, etc. Edge computing addresses the scalability challenges from the massive number of IoT end devices and IoT data value by separating the scalability domain into edge/local networks and remote network.
5.3.2. Data Management

With regard to the high level challenges listed in Section 4, data storage and processing at the edge is a major aspect of IoT edge computing. Data may therefore need to be classified (e.g., in terms of privacy, importance, validity, etc.). Data analysis such as performed in AI/ML tasks performed at the edge may benefit from specialized hardware support on computing nodes. IoT edge computing will face challenges in term of, for example, programmability, naming, data abstraction, data service management and data discovery (discussed in communication brokering). Furthermore, while edge computing can support IoT services independently of cloud computing, it can also be connected to cloud computing. Thus, the relationship of IoT edge computing to cloud computing, with regard to data management, is another potential challenge [ISO_TR].

5.4. Simulation and Emulation Environments

IoT Edge Computing brings new challenges to simulation and emulation tools used by researchers and developers. A varied set of applications, network and computing technologies can coexist in a distributed system, which make modelling difficult. Scale, mobility and resource management are additional challenges [SimulatingFog].

Tools include simulators, where simplified application logic runs on top of a fog network model, and emulators, where actual applications can be deployed, typically in software containers, over a cloud infrastructure (e.g. Docker, Kubernetes) itself running over a network emulating edge network conditions such as variable delays, throughput and mobility events. To gain in scale, emulated and simulated systems can be used together in hybrid federation-based approaches [PseudoDynamicTesting], while to gain in realism physical devices can be interconnected with emulated systems. Examples of related work and platforms include the publicly accessible MEC sandbox work recently initiated in ETSI [ETSI_Sandbox], and open source simulators and emulators ([AdvantEDGE] emulator and tools cited in [SimulatingFog]).

6. Security Considerations

T.B.D.

7. Acknowledgement

The authors would like to thank Joo-Sang Youn, Akbar Rahman, Michel Roy and Robert Gazda for their valuable comments and suggestions on this document.
8. References

8.1. Normative References


8.2. Informative References


<https://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/02.01.01_60/gs_MEC002v020101p.pdf>.


[Linux_Foundation_Edge]


Appendix A. Overview of the IoT Edge Computing

This list of initiatives, projects and products aim to provide an overview of the IoT edge computing.

Our goal is to be representative rather than exhaustive.

Please help us complete this overview by communicating with us about entries we have missed.

A.1. Open Source Projects

A.1.1. Gateway/CPE Platforms

EdgeX Foundry, Home Edge, Edge Virtualization Engine are Linux Foundation projects ([Linux_Foundation_Edge]) aiming to provide a platform for edge computing devices.
Such an open source platform can, for example, host proprietary programs currently run on IoT gateway products (Appendix A.2).

EdgeX Foundry develops an edge computing framework running on the IoT gateway.

Home Edge develops an edge computing framework especially dedicated to home computing devices, controlling home appliances, sensors, etc., and enabling AI applications, especially distributed and parallel machine learning.

The Edge Virtualization Engine (EVE) project develops a virtualization platform (for VMs and containers) designed to run outside of the datacenter, in an edge network; EVE is deployed on bare-metal hardware.

Computing devices: Hardware support for EdgeX and EVE is similar: they support x86 and ARM-based computing devices; A typical target can be a Linux Raspberry Pi with 1GB RAM, 64bit CPU, 32GB storage.

Service platform: EdgeX uses a micro-service architecture. Micro-services on the gateway are connected together, and to outside applications, through REST, or messaging technologies such as MQTT, AMQP and OMQ. The gateway can communicate with external backend applications or other gateways (north-south in tiered deployments or east-west in more distributed deployments). Gateway-device communication can use a wide range of IoT protocols. "Export services" enable on-gateway and off-gateway clients to register as recipient for data from devices. Core services are microservices that deal with persisting data from devices or alternatively "streaming" device data through, without persistence (core data service); managing information about the IoT devices, including their sensors, how to communicate with them, etc. (metadata service); and actual communication with IoT devices, on behalf of other on-gateway or off-gateway services (command service). A rule engine provides an API to register actions in response to conditions typically including an IoT device ID, sensor values to check, thresholds, etc. The scheduling micro service deals with organizing the removal of data persisted on the gateway. Alerts and notifications microservice can be used to dispatch alert/notifications from internal or external sources to interested consumers including backend servers, or human operators through email or SMS.

Edge cloud applications: Target applications for EdgeX include Industrial IoT (e.g., IoT sensor data and actuator control mixed with augmented reality application for technicians). Home Edge
focuses on smart home use cases, including using AI lifestyle and safety applications.

A.1.2. Edge Cloud Management Platforms

This set of open-source projects setup and manage clouds of individual edge computing devices.

StarlingX ([StarlingX]) extends OpenStack to provide virtualization platform management for edge clouds, which are distributed (in the range of 100 compute devices), secure and highly available.

Akraino Edge Stack, another project from the Linux Fundation Edge [Linux_Foundation_Edge], has a wider scope of developing a management platform adapted for the edge (e.g., covering 1000 plus locations), aiming for zero-touch provisioning, and zero-touch lifecycle management.

Computing devices: Compute devices are typically Linux-based application servers or more constrained devices.

Service platform: StarlingX adds new management services to OpenStack by leveraging building blocks such as Ceph for distributed storage, Kubernetes for orchestration. The new services are for management of configuration (enabling auto-discovery and configuration), faults, hosts (enabling host failure detection and auto-recovery), services (providing high availability through service redundancy and multi-path communication) and software (enabling updates).

Edge cloud applications: An edge computing platform may support a wide range of use cases. E.g., autonomous vehicles, industrial automation and robotics, cloud RAN, metering and monitoring, mobile HD video, content delivery, healthcare imaging and diagnostics, caching and surveillance, augmented/virtual reality, small cell services for high density locations (stadiums), universal CPE applications, retail.

A.1.3. Related Projects

Open Edge Computing ([OpenEdgeComputing]) is an initiative from universities, manufacturers, infrastructure providers and operators, enabling efficiently offloading cloudlets (VMs) to the edge. Computing devices are typically powerful, well-connected servers located in mobile networks (e.g., collocated with base stations or aggregation sites). The service platform is built on top of OpenStack++, an extension of OpenStack to support cloudlets. This project is mentioned here as a related project because of its edge
computing focus, and potential for some IoT use cases. Nevertheless, its primary use cases are typically non-IoT related, such as offloading processing-intensive applications from a mobile device to the edge.

A.2. Products

A.2.1. IoT Gateways

Multiple products are marketed as IoT gateways (Amazon Greengrass, Microsoft Azure IoT Edge, Google Cloud IoT Core, and gateway solutions from Bosh and Siemens). They are typically composed of a software framework that can run on a wide range of IoT gateway hardware devices to provide local support for cloud services, as well as some other local IoT gateway features such as relaying communication and caching content. Remote cloud is both used for management of the IoT gateways, and for hosting customer application components. Some IoT gateway products (Amazon Snowball) have a primary purpose of storing edge data on premises, to enable physically moving this data into the cloud without incurring digital data transfer cost.

Computing devices: Typical computing devices run Linux, Windows or a Real-Time OS over an ARM or x86 architecture. The level of service support on the computing device can range from low-level packages giving maximum control to embedded developers, to high-level SDKs. Typical requirements can start at 1GHz and 128MB RAM, e.g., ranging from Raspberry Pi to a server-level appliance.

Service platform: IoT gateways can provide a range of service including: running stateless functions; routing messages between connected IoT devices (using a wide range of IoT protocols); caching data; enabling some form of synchronization between IoT devices; authenticating and encrypting device data. Association between IoT devices and gateway based can require a device certificate.

Edge cloud applications: Pre-processing of IoT data for later processing in the cloud is a major driver. Use cases include industrial automation, farming, etc.

A.2.2. Edge Cloud Platforms

Services such as MobileEdgeX provide a platform for application developers to deploy software (e.g., as software containers) on edge networks.
Computing devices: Bare metal and virtual servers provided by mobile network operators are used as computing devices.

Service platform: The service platform provides end device location service, using GPS data obtained from platform software deployed in end devices, correlated with location information obtained from the mobile network. The service platform manages the deployment of application instances (containers) on servers close to end devices, using a declarative specification of optimal location from the application provider.

Edge cloud applications: Use cases include autonomous mobility, asset management, AI-based systems (e.g., quality inspection, assistance systems, safety and security cameras) and privacy-preserving video processing. There are also non-IoT use cases such as augmented reality and gaming.

A.3. Standards Initiatives

A.3.1. ETSI Multi-access Edge Computing

The ETSI MEC industry standardization group develops specifications that enable efficient and seamless integration of applications from vendors, service providers, and 3rd parties across multi-vendor MEC platforms ([ETSI_MEC_03]).

Basic principles followed include: leveraging NFV infrastructure; being compliant with 3GPP systems; focusing on orchestration, MEC services, applications and platforms.

Phase 1 (2015-2016) focused on basic platform services. Phase 2 (2017-2019) focuses on: supporting non-3GPP radio access technologies, especially WiFi; supporting a distributed, multi-operator and multi-vendor architecture; supporting non-VM based virtualization such as containers and PaaS.

Computing devices: Computing devices are typically application servers, attached to an eNodeB or at a higher level of aggregation point, and provide service to end users.

Service platform: The mobile edge platform offers an environment where the mobile edge applications can discover, advertise, consume and offer mobile edge services. The platform can provide certain native services such as radio network information, location, bandwidth management etc. The platform manager is responsible for managing the life cycle of applications including informing the mobile edge orchestrator of relevant application related events, managing the application rules and requirements.
including service authorizations, traffic rules, DNS configuration.

Edge cloud applications: Some of the use cases for MEC ([ETSI_MEC_02]) are IoT-related, including: security and safety (face recognition and monitoring), sensor data monitoring, active device location (e.g., crowd management), low latency vehicle-to-infrastructure and vehicle-to-vehicle (V2X, e.g., hazard warnings), video production and delivery, camera as a service.

A.3.2. Edge Computing Support in 3GPP

The 3GPP standards organization included edge computing support in 5G [3GPP.23.501]. Integration of MEC and 5G systems has been studied in ETSI as well [ETSI_MEC_WP_28].

Computing devices: From 3GPP standpoint, a mobile device may access any computing device located in a local data network, i.e., traffic is steered towards the local data network where the computing device is located.

Service platform: An external party may influence steering, QoS and charging of traffic towards the computing device. Session and service continuity can ensure that edge service is maintained when a client device moves. The network supports multiple-anchor connections, which makes it possible to connect a client device to both a local and a remote data network. The client device can be made aware of the availability of a local area data network, based on its location.

Edge cloud applications: Edge cloud applications in 3GPP can help support the major use cases envisioned for 5G, including massive IoT and V2X.

A.3.3. OpenFog and Industrial Internet Consortium

The OpenFog Consortium (now merged with the Industrial Internet Consortium) aims to standardize industrial IoT, fog, and edge computing. It produced a reference architecture for the fog ([OpenFog]), which has been published as IEEE standard P1934 in 2018. This work continues within the Industrial Internet Consortium.

Computing devices: Fog nodes include computational, networking, storage and acceleration elements. This includes nodes collocated with sensors and actuators, roadside or mobile nodes involved in V2X connectivity. Fog nodes should be programmable and may support multi-tenancy. Fog computing devices must employ a
hardware-based immutable root of trust, i.e., a trusted hardware component which receives control at power-on.

Service platform: The service platform is structured around "pillars" including: security end-to-end, scalability by adding internal components or adding more fog nodes, openness in term of discovery of/by other nodes and networks, autonomy from centralized clouds (for discovery, orchestration and management, security and operation) and hierarchical organization of fog nodes.

Edge cloud applications: Major use cases include smart cars and traffic control, visual security and surveillance, smart cities.

A.3.4. Related Standards


A.4. Research Projects

A.4.1. Named Function Networking

Named Function Networking ([Sifalakis]) is a research project that aims to extend ICN concepts (especially named data networking) to have the network orchestrate computation. Interests are sent for a combination of function and argument names, instead of using the content name in NDN.

Computing devices: NFN-capable switches are collocated with computing devices.

Service platform: NFN enables accessing static data and dynamic computation results in one data-oriented framework, thus benefiting from usual ICN features such as data authenticity and caching, as well as enabling the network to perform various optimizations, e.g., moving data, code or both closer to requesters. NFN also enables secure access to individual elements within Named Data Objects, e.g., for filtering or aggregation.

Edge cloud applications: Use cases include some form of MapReduce operations and service chaining. NDN, on which NFN is based, has been studied in the context of IoT, where it can provide local trust management and rendezvous service.
A.4.2. 5G-CORAL

The 5G-CORAL project ([_5G-CORAL]) aims to enable convergence of access across multiple radio access technologies using fog computing, using for this purpose an edge and fog computing system (EFS).

Computing devices: Computing devices used in 5G-CORAL include cloud and central data center servers, edge data center servers, and fixed or mobile "fog computing devices", which can be computing devices located in vehicles or factories, e.g., IoT gateways, mobile phones, cyber-physical devices, etc.

Service platform: 5G-CORAL architecture is based on an integrated virtualized edge and fog computing system (EFS), that aims to be flexible, scalable and interoperable with other domains including transport (fronthaul, backhaul), core and clouds. An Orchestration and Control System (OCS) enables automatic discovery of heterogeneous, multiple-owner resources, and federate them into a unified hosting environment. OCS monitors resource usage to guarantee service levels. Finally, OCS also includes orchestration and life cycle functions, including live migration and scaling. Applications (user and third-party) both inside and outside the EFS subscribe to EFS services through APIs, with emphasis on IoT and cyber-physical functionalities.

Edge cloud applications: EFS-hosted services include analytics obtained from IoT gateways (e.g., LORA or eNodeB gateways), context information services from RATs, transport (fronthaul and backhaul) and core networks. EFS-hosted functions include network performance acceleration functions, virtualized C-RAN functions for access nodes and possible end user devices.

A.4.3. FLAME

The FLAME project ([FLAME]) aims to improve performance of interactive media systems while keeping infrastructure costs low.

It builds over virtualization technologies such as XOS, OpenStack and ONOS/ODL to offer a programmable media service platform.

FLAME leverages IP-over-ICN technology developed through earlier projects including POINT ([POINT]).

Computing devices: The FLAME platform provides a service layer on top of an infrastructure platform, which can include cloud servers as well as computing devices collocated with WiFi access points.
Service platform: The FLAME platform can be seen as an "edge + cloud" computing platform with a use case focus on media dissemination, although the basic platform can be suitable for micro-services in general. The computing platform is comprised of: computing devices, an infrastructure platform (XOS, OpenStack, ONOS/ODL), NFV-MANO components (orchestrator, virtual infrastructure manager) and FLAME platform core services (PCE, network access point, surrogate manager).

Edge cloud applications: IoT use cases include public safety, such as supporting body-worn camera for police and social workers. As opposed to other multi-media applications that are also envisioned (pre-processing, user reporting, curation...), where a typical goal is to curate content early at the edge, to reduce expected high data volume, public safety use cases are typically about implementing triggers at the edge: everything needs to be kept anyway, to be available in case of an audit. Content is stored offline during off peak-hours delivery. For privacy and data volume concerns, triggers for, e.g., alerting police, cannot be performed in the cloud and should be performed as close to the data source as possible.

Authors’ Addresses

Jungha Hong
ETRI
218 Gajeong-ro, Yuseung-Gu
Daejeon

Email: jhong@etri.re.kr

Yong-Geun Hong
ETRI
218 Gajeong-ro, Yuseung-Gu
Daejeon

Email: yghong@etri.re.kr

Xavier de Foy
InterDigital Communications, LLC
1000 Sherbrooke West
Montreal H3A 3G4
Canada

Email: xavier.defoy@interdigital.com
Matthias Kovatsch  
Huawei Technologies Duesseldorf GmbH  
Riesstr. 25 C // 3.OG  
80992 Munich  
Germany  

Email: ietf@kovatsch.net

Eve Schooler  
Intel  
2200 Mission College Blvd.  
Santa Clara, CA, 95054-1537  
United States of America  

Email: eve.m.schooler@intel.com

Dirk Kutscher  
University of Applied Sciences Emden/Leer  
Constantiaplatz 4  
26723 Emden  
Germany  

Email: ietf@dkutscher.net