

Workgroup: Network Working Group
Internet-Draft: draft-irtf-t2trg-iot-edge-09
Published: 24 July 2023
Intended Status: Informational
Expires: 25 January 2024
Authors: J. Hong Y.-G. Hong
 ETRI Daejeon University
 X. de Foy
 InterDigital Communications, LLC
 M. Kovatsch E. Schooler
 Huawei Technologies Duesseldorf GmbH Intel
 D. Kutscher
 Hong Kong University of Science and Technology (Guangzhou)

IoT Edge Challenges and Functions

Abstract

Many IoT applications have requirements that cannot be met by the traditional Cloud (aka cloud computing). These include time sensitivity, data volume, connectivity 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 general model, and major components of the IoT Edge, to provide a common base for future discussions in T2TRG and other IRTF and IETF groups. This document is a product of the IRTF Thing-to-Thing Research Group (T2TRG).

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 25 January 2024.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

- [1. Introduction](#)
- [2. Background](#)
 - [2.1. Internet of Things \(IoT\)](#)
 - [2.2. Cloud Computing](#)
 - [2.3. Edge Computing](#)
 - [2.4. Examples of IoT Edge Computing Use Cases](#)
- [3. IoT Challenges Leading Towards Edge Computing](#)
 - [3.1. Time Sensitivity](#)
 - [3.2. Connectivity Cost](#)
 - [3.3. Resilience to Intermittent Services](#)
 - [3.4. Privacy and Security](#)
- [4. IoT Edge Computing Functions](#)
 - [4.1. Overview of IoT Edge Computing Today](#)
 - [4.2. General Model](#)
 - [4.3. OAM Components](#)
 - [4.3.1. Resource Discovery and Authentication](#)
 - [4.3.2. Edge Organization and Federation](#)
 - [4.3.3. Multi-Tenancy and Isolation](#)
 - [4.4. Functional Components](#)
 - [4.4.1. In-Network Computation](#)
 - [4.4.2. Edge Storage and Caching](#)
 - [4.4.3. Communication](#)
 - [4.5. Application Components](#)
 - [4.5.1. IoT Devices Management](#)
 - [4.5.2. Data Management and Analytics](#)
 - [4.6. Simulation and Emulation Environments](#)
- [5. Security Considerations](#)
- [6. Conclusion](#)
- [7. IANA Considerations](#)
- [8. Acknowledgements](#)
- [9. Informative References](#)
- [Authors' Addresses](#)

1. Introduction

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 scalability 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, cloud services).

However, IoT devices are generating vast amounts of data at the edge of the network. To meet IoT use case requirements, that data

increasingly is being stored, processed, analyzed, and acted upon close to the data sources. These requirements include time sensitivity, data volume, connectivity cost, resiliency in the face of intermittent connectivity, privacy, and security, which cannot be addressed by today's centralized cloud computing. To address these needs effectively, a more flexible approach is necessary. This involves distributing computing (and storage) and seamlessly integrating it into the edge-cloud continuum. We will refer to this integration of edge computing and IoT as "IoT edge computing". This draft describes related background, uses cases, challenges, system models, and functional components.

Due to the dynamic nature of the IoT edge computing landscape, this document does not list existing projects in this field. However, [Section 4.1](#) presents a high-level overview of the field, based on a limited review of standards, research, open-source and proprietary products in [[I-D.defoy-t2trg-iot-edge-computing-background](#)].

This document represents the consensus of the Thing-to-Thing Research Group (T2TRG). It has been reviewed extensively by the Research Group (RG) members who are actively involved in the research and development of the technology covered by this document. It is not an IETF product and is not a standard.

2. Background

2.1. 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 [[Ashton](#)], the concept of IoT has evolved. It now reflects a vision of connecting the physical world to the virtual world of computers using (wireless) networks over which 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 Web technology.

A Thing is a physical item that is made available in the Internet of Things, thereby enabling digital interaction with the physical world for humans, services, and/or other Things ([\[I-D.irtf-t2trg-rest-iot\]](#)). In this document we will use the term "IoT device" to designate the embedded system attached to the Thing.

Things are not necessarily constrained. Resource-constrained Things such as sensors, home appliances and wearable devices have limited storage and processing power, which raise concerns regarding reliability, performance, energy consumption, security, and privacy [[Lin](#)]. However, more generally Things, constrained or not, tend to generate a voluminous amount of data. This range of factors led to complementing IoT with cloud computing, at least initially.

2.2. 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". Low cost and massive availability of storage and processing power enabled the realization of another 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].

Today, an unprecedented volume and variety of data is generated by Things, and applications deployed at the network edge consume this data. In this context, cloud-based service models are not suitable for some classes of applications, which for example need very short response times, access to local personal data, or generate vast amounts of data. Those applications may instead leverage edge computing.

2.3. Edge Computing

Edge computing, also referred to as fog computing in some settings, is a new paradigm in which substantial computing and storage resources are placed at the edge of the Internet, that is, close to mobile devices, sensors, actuators, or machines. Edge computing happens near data sources [Mahadev], or closer (topologically, physically, in terms of latency, etc.) to where decisions or interactions with the physical world are happening. It processes both downstream data, e.g., originated from cloud services, and upstream data, e.g., originated from end devices or network elements. 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 cloud services.

An edge device is any computing or networking resource residing between end-devices' data sources and cloud-based data centers. 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]. This does not preclude end devices from hosting computation themselves when possible, independently or as part of a distributed edge computing platform (this is also referred to as Mist Computing).

Several standards developing organization (SDO) and industry forums have provided definitions of edge and fog computing:

*ISO defines edge computing as 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 defines multi-access edge computing as 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 Industry IoT Consortium (IIC, now incorporating what was formerly OpenFog) defines fog computing as "a horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-things continuum" [[OpenFog](#)].

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 Network Function Virtualization (NFV) infrastructure, at aggregation points or in proximity to the user equipment (e.g., gNodeBs) [[ETSI_MEC_03](#)].

*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 computing power than an automation controller.

2.4. Examples 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. These examples are followed with references to other use cases.

Smart Factory

As part of the 4th industrial revolution, smart factories run real-time processes based on IT technologies such as artificial intelligence and big data. In a smart factory, even a very small environmental change can lead to a situation in which production efficiency decreases or product quality problems occur. Therefore, simple but time-sensitive processing can be performed at the edge: for example, controlling temperature and humidity in the factory, or operating machines based on the real-time collection of the operational status of each machine. On the other hand, data requiring highly precise analysis, such as machine lifecycle management or accident risk prediction, can be transferred to a central data center for processing.

The use of edge computing in a smart factory can reduce the cost of network and storage resources by reducing the communication load to the central data center or server. It is also possible to improve process efficiency and facility asset productivity through the real-time prediction of failures, and to reduce the cost of failure through preliminary measures. In the existing manufacturing field, production facilities are manually run according to a program entered in advance, but edge computing in a smart factory enables tailoring solutions by analyzing data at each production facility and machine level. Digital twins [[Jones](#)] of IoT devices have been used jointly with edge computing in industrial IoT scenarios [[Chen](#)].

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 to, 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 to monitor power generation and power demand, and making local electrical energy storage decisions in the smart grid system.

Smart Agriculture

Smart agriculture integrates information and communication technology with farming technology. Intelligent farms use IoT technology to measure and analyze temperature, humidity, sunlight, carbon dioxide, soil, etc. in crop cultivation facilities. Depending on analysis results, control devices are used to set environmental parameters to an appropriate state. Remote management is also possible through mobile devices such as smartphones.

In existing farms, simple systems such as management according to temperature and humidity can easily and inexpensively be implemented with IoT technology. Sensors in fields are gathering data on field and crop condition. This data is then transmitted to cloud servers, which process data and recommend actions. Usage of edge computing

can reduce by a large amount data transmitted up and down the network, resulting in saving cost and bandwidth. Locally generated data can be processed at the edge, and local computing and analytics can drive local actions. With edge computing, it is also easy for farmers to select large amounts of data for processing, and data can be analyzed even in remote areas with poor access conditions. Other applications include enabling dashboarding, e.g., to visualize the farm status, as well as enhancing XR applications that require edge audio/video processing. As the number of people working on farming decreases over time, increasing automation enabled by edge computing can be a driving force for future smart agriculture.

Smart Construction

Safety is critical on a construction site. Every year, many construction workers lose their lives due to falls, collisions, electric shocks, and other accidents. Therefore, solutions have been developed in order to improve construction site safety, including real-time identification of workers, monitoring of equipment location, and predictive accident prevention. To deploy these solutions, many cameras and IoT sensors were installed on construction sites, measuring noise, vibration, gas concentration, etc. Typically, data generated from these measurements has been collected in an on-site gateway and sent to a remote cloud server for storage and analysis. Thus, an inspector can check the information stored on the cloud server to investigate an incident. However, this approach can be expensive, due to transmission costs, e.g., of video streams over an LTE connection, and due to usage fees of private cloud services such as Amazon Web Services.

Using edge computing, data generated on the construction site can be processed and analyzed on an edge server located within or near the site. Only the result of this processing needs to be transferred to a cloud server, thus saving transmission costs. It is also possible to locally generate warnings to prevent accident in real-time.

Self-Driving Car

Edge computing plays a crucial role in safety-focused self-driving car systems. With a multitude of sensors such as high-resolution cameras, radars, LIDAR, sonar sensors, and GPS systems, autonomous vehicles generate vast amounts of real-time data. Local processing utilizing edge computing nodes allows for efficient collection and analysis of this data to monitor vehicle distances, road conditions, and respond promptly to unexpected situations. Roadside computing nodes can also be leveraged to offload tasks when necessary, e.g., when the local processing capacity on the car is insufficient due to low-performing hardware or a large amount of data.

For instance, when the car ahead slows down, a self-driving car adjusts its speed to maintain a safe distance, or when a roadside signal changes, it adapts its behavior accordingly. In another example, cars equipped with self-parking features utilize local

processing to analyze sensor data, determine suitable parking spots, and execute precise parking maneuvers without relying on external processing or connectivity. It is also possible for in-cabin cameras, coupled with local processing, to monitor the driver's attention level, detecting signs of drowsiness or distraction. The system can issue warnings or take preventive measures to ensure driver safety.

Edge computing empowers self-driving cars by enabling real-time processing, reducing latency, enhancing data privacy, and optimizing bandwidth usage. By leveraging local processing capabilities, self-driving cars can make rapid decisions, adapt to changing environments, and ensure a safer and more efficient autonomous driving experience.

Digital Twin

A digital twin can simulate different scenarios and predict outcomes based on real-time data collected from the physical environment. This simulation capability empowers proactive maintenance, optimization of operations, and prediction of potential issues or failures. Decision-makers can use digital twins to test and validate different strategies, identify inefficiencies, and optimize performances.

With edge computing, real-time data is collected, processed, and analyzed directly at the edge, allowing for accurate monitoring and simulation of the physical asset. Moreover, edge computing effectively minimizes latency, enabling rapid responses to dynamic conditions, as computational resources are brought closer to the physical object. Running digital twin processing at the edge enables organizations to get timely insights and make informed decisions that maximize efficiency and performance.

Other Use Cases

AI/ML systems at the edge empower real-time analysis, faster decision-making, reduced latency, improved operational efficiency, and personalized experiences across various industries, by bringing artificial intelligence and machine learning capabilities closer to the edge devices.

Additionally, oneM2M has studied several IoT edge computing use cases, which are documented in [[oneM2M-TR0001](#)], [[oneM2M-TR0018](#)] and [[oneM2M-TR0026](#)]. The edge computing related requirements raised through the analysis of these use cases are captured in [[oneM2M-TS0002](#)].

3. IoT Challenges Leading Towards Edge Computing

This section describes challenges met by IoT, that are motivating the adoption of edge computing. Those are distinct from research

challenges applicable to IoT edge computing, some of which will be mentioned in [Section 4.3](#).

IoT technology is used with more and more demanding applications, e.g., in industrial, automotive or healthcare domains, leading to new challenges. For example, industrial machines such as laser cutters already produce over 1 terabyte per hour, and similar amounts can be generated in 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.

3.1. 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](#)]. In some cases speed-of-light limitations may simply prevent a solution based on remote cloud, however it is not the only challenge relative to time sensitivity. Guarantees for bounded latency and jitter ([[RFC8578](#)] section 7) are also important to those industrial IoT applications. This means control packets need to arrive with as little variation as possible and within a strict deadline. Given the best-effort characteristics of the Internet, this challenge is virtually impossible to address, without using end-to-end guarantees for individual message delivery and continuous data flows.

3.2. Connectivity Cost

Some IoT deployments may not face bandwidth constraints when uploading data 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 terabytes per hour), allowing for transferring uplink large amounts of data. However, the cost of maintaining continuous high-bandwidth connectivity for such usage can be 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. Minimizing reliance on high-bandwidth connectivity is therefore a requirement, e.g., by processing data at the edge and deriving summarized or actionable insights that can be transmitted to the Cloud.

3.3. Resilience to Intermittent Services

Many IoT devices such as sensors, actuators, controllers, etc. have very limited hardware resources and cannot rely solely on their own 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. The dual is also true, a cloud back-end might want to have a reading of the device even if it's currently asleep.

3.4. 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 [[ENERGY](#)]. Policy-makers started to provide frameworks that limit the usage of personal data and put strict requirements on data controllers and processors. 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 owners of these systems are generally reluctant to upload IoT data to the Cloud.

Furthermore, passive observers can perform traffic analysis on the device-to-cloud path. Hiding traffic patterns associated with sensor networks can therefore be another requirement for edge computing.

4. IoT Edge Computing Functions

We will first look at the current state of IoT edge computing ([Section 4.1](#)), and then define a general system model ([Section 4.2](#)). This provides context for IoT edge computing functions, which are listed in [Section 4.3](#).

4.1. Overview of IoT Edge Computing Today

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 [[I-D.defoy-t2trg-iot-edge-computing-background](#)].

IoT gateways, both open-source (such as EdgeX Foundry or Home Edge) and proprietary (such as Amazon Greengrass, Microsoft Azure IoT Edge, Google Cloud IoT Core, and gateways from Bosch, Siemens), 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 (using VMs, application containers, etc.), including IoT gateway software, on servers in the mobile network infrastructure (at base stations and concentration points), in edge data centers (in central offices) or regional data centers located near central offices. End devices are envisioned to become computing devices in forward-looking projects, but they are not commonly used as such today.

Besides open-source and proprietary solutions, a horizontal IoT service layer is standardized by the oneM2M standards body, to reduce fragmentation, increase interoperability and promote reuse in the IoT ecosystem. Furthermore, ETSI MEC developed an IoT API [[ETSI MEC 33](#)] that enables deploying heterogeneous IoT platforms and provides the means to configure the various components of an IoT system.

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' application 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.

Resilience in IoT can entail the ability to operate autonomously in periods of disconnectedness in order to preserve the integrity and safety of the controlled system, possibly in a degraded mode. IoT devices and gateways are often expected to operate in the always-on and unattended mode, using fault detection and unassisted recovery functions.

Life cycle management of services and applications on physical IoT gateways is generally cloud-based. Edge cloud management platforms and products (such as StarlingX, Akraino Edge Stack, or proprietary products from major Cloud providers) adapt cloud management technologies (e.g., Kubernetes) to the edge cloud, i.e., to smaller, distributed computing devices running outside a controlled data center. Service and application life-cycle is typically using an NFV-like management and orchestration model.

The platform typically enables advertising or consuming services hosted on the platform (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 platform is typically extensible by edge applications, since they can advertise a service 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 distributed storage such as provided by a distributed storage platform (e.g., IPFS, EdgeFS, Ceph), or, in more experimental settings, by an ICN network, e.g., Named Function Networking (NFN) nodes can store data in a Named Data Networking (NDN) system. External storage, e.g., on databases in distant or local IT cloud, is typically used for filtered data deemed worthy of long-term storage, although in some cases it may be 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, e.g., using stateless containers), 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).

4.2. General Model

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 traditional cloud-based management of IoT devices and data to the edge). A common role of an IoT Gateway is to provide access to a 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 the cloud, it can also operate independently in a disconnected mode.

- *A set of distributed computing nodes, e.g., embedded in switches, routers, edge cloud servers, or mobile devices. Some IoT devices can have enough computing capabilities to participate in such distributed systems due to advances in hardware technology. In

this model, edge computing nodes can collaborate to share their resources.

*A hybrid system involving both IoT gateways and supporting functions in distributed computing nodes.

In the general model described in [Figure 1](#), the edge computing domain is interconnected with IoT 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, at the network edge, or through some interworking between edge network and remote cloud network.

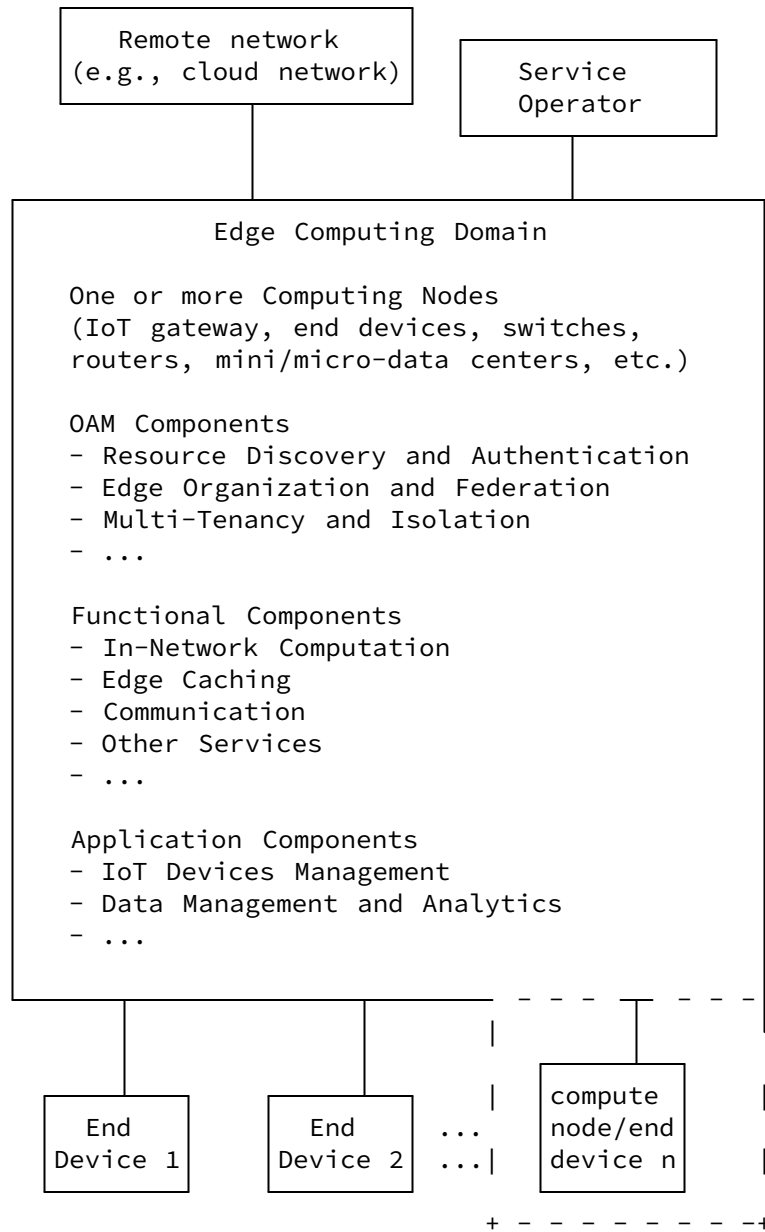


Figure 1: Model of IoT Edge Computing

In the distributed model described in [Figure 2](#), the edge computing domain is composed of IoT edge gateways and IoT devices which are also used as computing nodes. Edge computing domains are connected with a remote/cloud network, and with their respective service operator's system. IoT devices/computing nodes provide logical functions, for example as part of a distributed machine learning or distributed image processing application. The processing capabilities in IoT devices being limited, they require the support of other nodes: in a distributed machine learning application, the training process for AI services can be executed at IoT edge gateways or cloud networks and the prediction (inference) service is executed in the IoT devices; in a distributed image processing

application, some image processing functions can be similarly executed at the edge or in the cloud, while pre-processing, which helps limiting the amount of uploaded data, is performed by the IoT device.

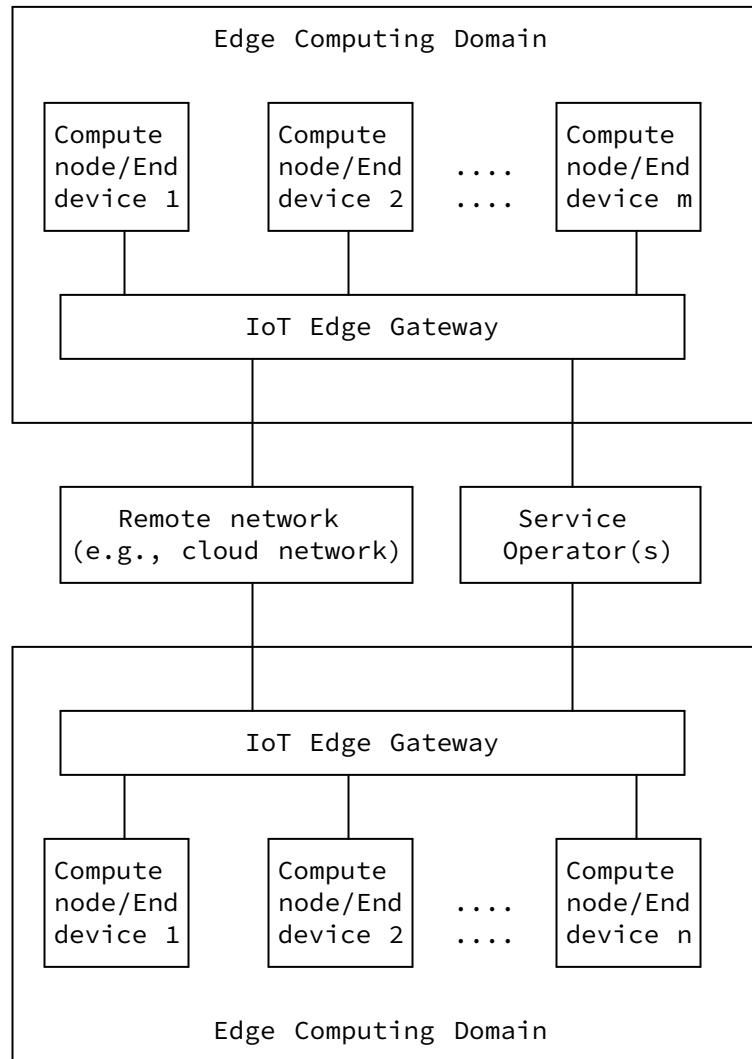


Figure 2: Example: Machine Learning over a Distributed IoT Edge Computing System

We now attempt to enumerate major edge computing domain components. They are here loosely organized into OAM (Operations, Administration, and Maintenance), functional and application components, with the understanding that the distinction between these classes may not always be clear, depending on actual system architectures. Some representative research challenges are associated with those functions. We used input from co-authors, IRTF attendees, and some comprehensive reviews of the field ([Yousefpour], [Zhang2], [Khan]).

4.3. 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 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. Discovery during normal operation (e.g., discovery of compute or storage nodes by endpoints) would typically not be included in OAM, however in this document we will not address it separately. 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. Centralized (e.g., SDN) and decentralized management systems can be used. Finally, we arbitrarily chose to address data management as an application component, however, in some systems, data management may be considered to be similar to a network management function.

We further detail a few OAM components.

4.3.1. Resource Discovery and Authentication

Discovery and authentication may target platforms, infrastructure resources, such as compute, network and storage, but also other resources such as IoT devices, sensors, data, code units, services, applications, or users interacting with the system. Broker-based solutions can be used, e.g., using an IoT gateway as a broker to discover IoT resources. More decentralized solutions can also be used in replacement or complement, e.g., CoAP enables multicast discovery of an IoT device, and CoAP service discovery enables obtaining a list of resources made available by this device [[RFC7252](#)]. Today, centralized gateway-based systems rely, for device authentication, on the installation of a secret on IoT devices and computing devices (e.g., a device certificate stored in a hardware security module, or a combination of code and data stored in a trusted execution environment).

Related challenges include:

- *Discovery, authentication, and trust establishment between IoT devices, compute nodes, and platforms, with regard to concerns such as mobility, heterogeneous devices and networks, scale, multiple trust domains, constrained devices, anonymity, and traceability.

*Intermittent connectivity to the Internet, preventing relying on a third-party authority [[Echeverria](#)].

*Resiliency to failures [[Harchol](#)], denial of service attacks, easier physical access for attackers.

4.3.2. 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. The organization structure may range from centralized to peer-to-peer, or it may be closely tied with other systems. Such groups can also form federations with other edge or remote clouds.

Related challenges include:

*Support for scaling, and enabling fault-tolerance or self-healing [[Jeong](#)]. Besides using hierarchical organization to cope with scaling, another available and possibly complementary mechanism is multicast ([[RFC7390](#)] [[I-D.ietf-core-groupcomm-bis](#)]). Other approaches include relying on blockchains [[Ali](#)].

*Integration of edge computing with virtualized Radio Access Networks (Fog RAN) [[I-D.bernardos-sfc-fog-ran](#)] and with 5G access networks.

*Sharing resources in multi-vendor/operator scenarios, to optimize criteria such as profit [[Anglano](#)], resource usage, latency, or energy consumption.

*Capacity planning, placement of infrastructure nodes to minimize delay [[Fan](#)], cost, energy, etc.

*Incentives for participation, e.g., in peer-to-peer federation schemes.

*Design of federated AI over IoT edge computing systems [[Brecko](#)], e.g., for anomaly detection.

4.3.3. Multi-Tenancy and Isolation

Some IoT edge computing systems make use of virtualized (compute, storage and networking) resources to address the need for secure multi-tenancy at the edge. This leads to "edge clouds" that share properties with the remote Cloud and can reuse some of its ecosystem. Virtualization function management is covered to a large extent by ETSI NFV and MEC standards activities. Projects such as [[LFEDGE-EVE](#)] further cover virtualization and its management into distributed edge computing settings.

Related challenges include:

- *Adapting cloud management platforms to the edge, to account for its distributed nature, e.g., using Conflict-free Replicated Data Types (CRDT) [[Jeffery](#)], heterogeneity and customization, e.g., using intent-based management mechanisms [[Cao](#)], and limited resources.

- *Minimizing virtual function instantiation time and resource usage.

4.4. Functional Components

4.4.1. In-Network Computation

A core function of IoT edge computing is to enable local computation on a node at the network edge, typically for application-layer processing such as, e.g., processing input data from sensors, making local decisions, preprocessing data, offloading computation on behalf of a device, service, or user. Related functions include orchestrating computation (in a centralized or distributed manner) and managing application lifecycles. Support for in-network computation may vary in terms of capability, e.g., computing nodes can host virtual machines, software containers, software actors or unikernels able to run stateful or stateless code, or a rules engine providing an API to register actions in response to conditions such as IoT device ID, sensor values to check, thresholds, etc.

Edge offloading includes offloading to and from an IoT device, and to and from a network node. [[Cloudlets](#)] offer an example of offloading from an end device to a network node. On the other side, oneM2M is an example of a system that allows a cloud-based IoT platform to transfer resources and tasks to a target edge node [[oneM2M-TR0052](#)]. Once transferred, the edge node can directly support IoT devices it serves with the service offloaded by the cloud (e.g., group management, location management, etc.)

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. For example, in some systems, a bandwidth manager service can be exposed to enable allocation of bandwidth to/from an edge computing application instance.

In-network computation may leverage underlying services, provided using data generated by IoT devices and access networks. Such services include IoT device location, radio network information, bandwidth management and congestion management (e.g., by the congestion management feature of oneM2M [[oneM2M-TR0052](#)]).

Related challenges include:

- *(Computation placement) Selecting, in a centralized or distributed/peer-to-peer manner, an appropriate compute device based on available resources, location of data input and data sinks, compute node properties, etc., and with varying goals including for example end-to-end latency, privacy, high availability, energy conservation, or network efficiency, e.g., using load balancing techniques to avoid congestion.
- *Onboarding code on a platform or computing device, and invoking remote code execution, possibly as part of a distributed programming model and with respect to similar concerns of latency, privacy, etc. For example, this can include offloading in a vehicular scenario [[Grewe](#)]. These operations should deal with heterogeneous compute nodes [[Schafer](#)], and may in some cases also support end devices, including IoT devices, as compute nodes [[Larrea](#)].
- *Adapting Quality of Results (QoR) for applications where a perfect result is not necessary [[Li](#)].
- *Assisted or automatic partitioning of code, for example for application programs [[I-D.sarathchandra-coin-appcentres](#)] or for network programs [[I-D.hsingh-coinrg-reqs-p4comp](#)].
- *Supporting computation across trust domains, e.g., verifying computation results.
- *Support for computation mobility: relocating an instance from one compute node to another, while maintaining a given service level; session continuity when communicating with end devices that are mobile, possibly at high speed (e.g., in vehicular scenarios); defining lightweight execution environments for secure code mobility, e.g., using WebAssembly [[Nieke](#)].
- *Defining, managing, and verifying Service Level Agreements (SLA) for edge computing systems. Pricing is a related challenge.

4.4.2. Edge Storage and Caching

Local storage or caching enable local data processing (e.g., pre-processing or analysis), as well as delayed data transfer to the cloud or delayed physical shipping. An edge node may offer local data storage (where persistence is subject to retention policies), caching, or both. Caching generally refers to temporary storage to improve performance with no persistence guarantees. An edge caching component manages data persistence, e.g., it schedules removal of data when it is no longer needed. Other related aspects include authenticating and encrypting data. Edge storage and caching can take the form of a distributed storage system.

Related challenges include:

- *(Cache and data placement) Using cache positioning and data placement strategies to minimize data retrieval delay [[Liu](#)], energy consumption. Caches may be positioned in the access network infrastructure, or on end devices.

- *Maintaining consistency, freshness, reliability, and privacy of stored/cached data in systems that are distributed, constrained, and dynamic (e.g., due to end devices and computing nodes churn or mobility), and which can have additional data governance constraints on data storage location. For example, [[Mortazavi](#)] exploits a hierarchical storage organization. Freshness-related metrics include the age of information [[Yates](#)], that captures the timeliness of information from a sender (e.g., an IoT device).

4.4.3. Communication

An 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, and support secure communication.

An edge cloud may provide an API or interface to local or mobile users, for example, to provide access to services and applications, or to manage data published by local/mobile devices.

Edge computing nodes communicate with IoT devices over a southbound interface, typically for data acquisition and IoT device management.

Communication brokering is a typical function of IoT edge computing, that facilitates communication with IoT devices: for enabling clients to register as recipients for data from devices, as well as forwarding/routing of traffic to or from IoT devices, enabling various data discovery and redistribution patterns, e.g., north-south with clouds, east-west with other edge devices [[I-D.mcbride-edge-data-discovery-overview](#)]. Another related aspect is dispatching alerts and notifications to interested consumers both inside and outside of the edge computing domain. Protocol translation, analytics, and video 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, over device-to-device links. A broker can ensure communication reliability, traceability, and in some cases transaction management.

Related challenges include:

- *Defining edge computing abstractions, such as PaaS [[Yangui](#)], suitable for users and cloud systems to interact with edge computing systems, and dealing with interoperability issues such as data models heterogeneity.

- *Enabling secure and resilient communication between IoT devices and remote cloud, e.g., through multipath support.

4.5. Application Components

IoT edge computing can host applications such as the ones mentioned in [Section 2.4](#). While describing components of individual applications is out of our scope, some of those applications share similar functions, such as IoT device management, data management, described below.

4.5.1. IoT Devices Management

IoT 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 devices by separating the scalability domain into edge/local networks and remote networks. For example, in the context of the oneM2M standard, the software campaign feature enables installing, deleting, activating, and deactivating software functions/services on a potentially large number of edge nodes [[oneM2M-TR0052](#)]. Using a dashboard or a management software, a service provider issues those requests through an IoT cloud platform supporting the software campaign functionality.

Challenges listed in [Section 4.3.1](#) may be applicable to IoT devices management as well.

4.5.2. Data Management and Analytics

Data storage and processing at the edge is a major aspect of IoT edge computing, directly addressing high-level IoT challenges listed in [Section 3](#). Data analysis such as performed in AI/ML tasks performed at the edge may benefit from specialized hardware support on computing nodes.

Related challenges include:

- *Addressing concerns on resource usage, security, and privacy when sharing, processing, discovering, or managing data. For example by presenting data in views composed of an aggregation of related data [[Zhang](#)]; protecting data communication between authenticated peers [[Basudan](#)]; classifying data (e.g., in terms of privacy, importance, validity, etc.); compressing and encrypting data, e.g., using homomorphic encryption to directly process encrypted data [[Stanciu](#)].

*Other concerns on edge data discovery (e.g., streaming data, metadata, events) include siloization and lack of standard in edge environments that can be dynamic (e.g., vehicular networks) and heterogeneous [[I-D.mcbride-edge-data-discovery-overview](#)].

*Data-driven programming models [[Renart](#)], e.g., event-based, including handling of naming and data abstractions.

*Data integration in an environment that do not have data standardization or where different sources use different ontologies [[Farnbauer-Schmidt](#)].

*Addressing concerns such as limited resources, privacy, dynamic and heterogeneous environment, to deploy machine learning at the edge. For example, making machine learning more lightweight and distributed (e.g., to enable distributed inference at the edge), supporting shorter training time and simplified models, and supporting models that can be compressed for efficient communication [[Murshed](#)].

*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](#)].

4.6. 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 makes modeling 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 network edge 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](#)]). EdgeNet [[Senel](#)] is a globally distributed edge cloud for Internet researchers, using nodes contributed by institutions, and based on Docker for containerization and Kubernetes for deployment and node management.

Digital twins are virtual instances of a physical system (twin) that is continually updated with the latter's performance, maintenance,

and health status data throughout the physical system's life cycle [[Madni](#)]. As opposed to a traditional emulation or simulated environment, digital twins, once generated, are maintained in sync by their physical twin, which can be, among many other instances, an IoT device, an edge device or an edge network. The benefits of digital twins go beyond those of emulation, and include accelerated business processes, enhanced productivity, and faster innovation with reduced costs [[I-D.irtf-nmrg-network-digital-twin-arch](#)].

5. Security Considerations

Privacy and security are drivers for the adoption of edge computing for IoT ([Section 3.4](#)). As discussed in [Section 4.3.1](#), authentication and trust (between computing nodes, management nodes, end devices) can be challenging as scale, mobility, and heterogeneity increase. The sometimes disconnected nature of edge resources can prevent relying on a third-party authority. Distributed edge computing is exposed to issues with reliability and denial of service attacks. Personal or proprietary IoT data leakage is also a major threat, especially due to the distributed nature of the systems ([Section 4.5.2](#)). Furthermore, blockchain-based distributed IoT edge computing need to be designed for privacy, since public blockchain addressing does not guarantee absolute anonymity [[Ali](#)].

However, edge computing also brings solutions in the security space: maintaining privacy by computing sensitive data closer to data generators is a major use case for IoT edge computing. An edge cloud can be used to take actions based on sensitive data, or to anonymize or aggregate data prior to transmitting to a remote cloud server. Edge computing communication brokering functions can also be used to secure communication between edge and cloud networks.

6. Conclusion

IoT edge computing plays an essential role, complementary to the cloud, to enable IoT systems in some situations. This document starts by presenting use cases and listing core challenges faced by IoT, that drive the need for IoT edge computing. The first part of this document may therefore help focusing future research efforts on the aspects of IoT edge computing where it is most useful. A second part of this document presents a general system model and a structured overview of the associated research challenges and related work. The structure, based on the system model, is not meant to be restrictive, and exists for the purpose of having a link between individual research areas and where they are applicable in an IoT edge computing system.

7. IANA Considerations

This document has no IANA actions.

8. Acknowledgements

The authors would like to thank Joo-Sang Youn, Akbar Rahman, Michel Roy, Robert Gazda, Rute Sofia, Thomas Fossati, Chonggang Wang, Marie-José Montpetit, Carlos J. Bernardos, Milan Milenkovic, Dale Seed, JaeSeung Song, Roberto Morabito, Carsten Bormann and Ari Keränen for their valuable comments and suggestions on this document.

9. Informative References

- [AdvantEDGE] "Mobile Edge Emulation Platform", Source Code Repository, 2020, <<https://github.com/InterDigitalInc/AdvantEDGE>>.
- [Ali] Ali, M. S., Vecchio, M., and F. Antonelli, "Enabling a Blockchain-Based IoT Edge", IEEE Internet of Things Magazine pp. 24-29, DOI 10.1109/IOTM.2019.1800024, 2018, <<https://doi.org/10.1109/IOTM.2019.1800024>>.
- [Anglano] Anglano, C., Canonico, M., Castagno, P., Guazzone, M., and M. Sereno, "A game-theoretic approach to coalition formation in fog provider federations", 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), DOI 10.1109/fmec.2018.8364054, April 2018, <<https://doi.org/10.1109/fmec.2018.8364054>>.
- [Ashton] Ashton, K., "That Internet of Things thing", RFID J. vol. 22, no. 7, pp. 97-114, 2009, <<http://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf>>.
- [Basudan] Basudan, S., Lin, X., and K. Sankaranarayanan, "A Privacy-Preserving Vehicular Crowdsensing-Based Road Surface Condition Monitoring System Using Fog Computing", IEEE Internet of Things Journal vol. 4, no. 3, pp. 772-782, DOI 10.1109/jiot.2017.2666783, June 2017, <<https://doi.org/10.1109/jiot.2017.2666783>>.
- [Botta] Botta, A., de Donato, W., Persico, V., and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey", Future Generation Computer Systems vol. 56, pp. 684-700, DOI 10.1016/j.future.2015.09.021, March 2016, <<https://doi.org/10.1016/j.future.2015.09.021>>.
- [Brecko] Brecko, A., Kajati, E., Koziorek, J., and I. Zolotova, "Federated Learning for Edge Computing: A Survey", Applied Sciences 12, no. 18 9124, DOI 10.3390/app12189124, 2022, <<https://doi.org/10.3390/app12189124>>.
- [Cao] Cao, L., Merican, A., Tootaghaj, D., Ahmed, F., Sharma, P., and V. Saxena, "eCaaS: A Management Framework of Edge Container as a Service for Business Workload",

Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, DOI 10.1145/3434770.3459741, April 2021, <<https://doi.org/10.1145/3434770.3459741>>.

[Chen] Chen, B., Wan, J., Celesti, A., Li, D., Abbas, H., and Q. Zhang, "Edge Computing in IoT-Based Manufacturing", IEEE Communications Magazine vol. 56, no. 9, pp. 103-109, DOI 10.1109/mcom.2018.1701231, September 2018, <<https://doi.org/10.1109/mcom.2018.1701231>>.

[Chiang] Chiang, M. and T. Zhang, "Fog and IoT: An Overview of Research Opportunities", IEEE Internet of Things Journal vol. 3, no. 6, pp. 854-864, DOI 10.1109/jiot.2016.2584538, December 2016, <<https://doi.org/10.1109/jiot.2016.2584538>>.

[Cloudlets] Satyanarayanan, M., Bahl, P., Caceres, R., and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing", IEEE Pervasive Computing vol. 8, no. 4, pp. 14-23, DOI 10.1109/mprv.2009.82, October 2009, <<https://doi.org/10.1109/mprv.2009.82>>.

[Echeverria] Echeverria, S., Klinedinst, D., Williams, K., and G. Lewis, "Establishing Trusted Identities in Disconnected Edge Environments", 2016 IEEE/ACM Symposium on Edge Computing (SEC), DOI 10.1109/sec.2016.27, October 2016, <<https://doi.org/10.1109/sec.2016.27>>.

[ENERGY] Beckel, C., Sadamori, L., Staake, T., and S. Santini, "Revealing household characteristics from smart meter data", Energy vol. 78, pp. 397-410, DOI 10.1016/j.energy.2014.10.025, December 2014, <<https://doi.org/10.1016/j.energy.2014.10.025>>.

[ETSI_MEC_01] ETSI, "Multi-access Edge Computing (MEC); Terminology", ETSI GS MEC 001, 2019, <https://www.etsi.org/deliver/etsi_gs/MEC/001_099/001/02.01.01_60/gs_MEC001v020101p.pdf>.

[ETSI_MEC_03] ETSI, "Mobile Edge Computing (MEC); Framework and Reference Architecture", ETSI GS MEC 003, 2019, <https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf>.

[ETSI_MEC_33] ETSI, "Multi-access Edge Computing (MEC); IoT API", ETSI GS MEC 033, 2022, <https://www.etsi.org/deliver/etsi_gs/MEC/001_099/033/03.01.01_60/gs_MEC033v030101p.pdf>.

[ETSI_Sandbox] "Multi-access Edge Computing (MEC) MEC Sandbox Work Item", Portal, 2020, <https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=57671>.

[Fan]

Fan, Q. and N. Ansari, "Cost Aware cloudlet Placement for big data processing at the edge", 2017 IEEE International Conference on Communications (ICC), DOI 10.1109/icc.2017.7996722, May 2017, <<https://doi.org/10.1109/icc.2017.7996722>>.

[Farnbauer-Schmidt] Farnbauer-Schmidt, M., Lindner, J., Kaffenberger, C., and J. Albrecht, "Combining the Concepts of Semantic Data Integration and Edge Computing", INFORMATIK 2019 50 Jahre Gesellschaft für Informatik - Informatik für Gesellschaft, pp. 139-152, DOI 10.18420/inf2019_19, 2019, <https://doi.org/10.18420/inf2019_19>.

[Grewe] Grewe, D., Wagner, M., Arumaithurai, M., Psaras, I., and D. Kutscher, "Information-Centric Mobile Edge Computing for Connected Vehicle Environments: Challenges and Research Directions", Proceedings of the Workshop on Mobile Edge Communications pp. 7-12, DOI 10.1145/3098208.3098210, 2017, <<https://doi.org/10.1145/3098208.3098210>>.

[Harchol] Harchol, Y., Mushtaq, A., McCauley, J., Panda, A., and S. Shenker, "CESSNA: Resilient Edge-Computing", Proceedings of the 2018 Workshop on Mobile Edge Communications, DOI 10.1145/3229556.3229558, August 2018, <<https://doi.org/10.1145/3229556.3229558>>.

[I-D.bernardos-sfc-fog-ran] Bernardos, C. J. and A. Mourad, "Service Function Chaining Use Cases in Fog RAN", Work in Progress, Internet-Draft, draft-bernardos-sfc-fog-ran-10, 22 October 2021, <<https://datatracker.ietf.org/doc/html/draft-bernardos-sfc-fog-ran-10>>.

[I-D.defoy-t2trg-iot-edge-computing-background] de Foy, X., Hong, J., Hong, Y., Kovatsch, M., Schooler, E., and D. Kutscher, "IoT Edge Computing: Initiatives, Projects and Products", Work in Progress, Internet-Draft, draft-defoy-t2trg-iot-edge-computing-background-00, 25 May 2020, <<https://datatracker.ietf.org/doc/html/draft-defoy-t2trg-iot-edge-computing-background-00>>.

[I-D.hsingh-coinrg-reqs-p4comp] Singh, H. and M. Montpetit, "Requirements for P4 Program Splitting for Heterogeneous Network Nodes", Work in Progress, Internet-Draft, draft-hsingh-coinrg-reqs-p4comp-03, 18 February 2021, <<https://datatracker.ietf.org/doc/html/draft-hsingh-coinrg-reqs-p4comp-03>>.

[I-D.ietf-core-groupcomm-bis] Dijk, E., Wang, C., and M. Tiloca, "Group Communication for the Constrained Application Protocol (CoAP)", Work in Progress, Internet-Draft,

draft-ietf-core-groupcomm-bis-09, 10 July 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-core-groupcomm-bis-09>>.

[I-D.irtf-nmrg-network-digital-twin-arch]

Zhou, C., Yang, H., Duan, X., Lopez, D., Pastor, A., Wu, Q., Boucadair, M., and C. Jacquenet, "Digital Twin Network: Concepts and Reference Architecture", Work in Progress, Internet-Draft, draft-irtf-nmrg-network-digital-twin-arch-03, 27 April 2023, <<https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-network-digital-twin-arch-03>>.

[I-D.irtf-t2trg-rest-iot] Keränen, A., Kovatsch, M., and K. Hartke, "Guidance on RESTful Design for Internet of Things Systems", Work in Progress, Internet-Draft, draft-irtf-t2trg-rest-iot-11, 11 January 2023, <<https://datatracker.ietf.org/doc/html/draft-irtf-t2trg-rest-iot-11>>.

[I-D.mcbride-edge-data-discovery-overview]

McBride, M., Kutscher, D., Schooler, E., Bernardos, C. J., Lopez, D., and X. de Foy, "Edge Data Discovery for COIN", Work in Progress, Internet-Draft, draft-mcbride-edge-data-discovery-overview-05, 1 November 2020, <<https://datatracker.ietf.org/doc/html/draft-mcbride-edge-data-discovery-overview-05>>.

[I-D.sarathchandra-coin-appcentres] Trossen, D., Sarathchandra, C., and M. Boniface, "In-Network Computing for App-Centric Micro-Services", Work in Progress, Internet-Draft, draft-sarathchandra-coin-appcentres-04, 26 January 2021, <<https://datatracker.ietf.org/doc/html/draft-sarathchandra-coin-appcentres-04>>.

[ISO_TR] "Internet of things (IoT) - Edge computing", ISO/IEC TR 30164, 2020, <<https://www.iso.org/standard/53284.html>>.

[Jeffery] Jeffery, A., Howard, H., and R. Mortier, "Rearchitecting Kubernetes for the Edge", Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, DOI 10.1145/3434770.3459730, April 2021, <<https://doi.org/10.1145/3434770.3459730>>.

[Jeong] Jeong, T., Chung, J., Hong, J., and S. Ha, "Towards a distributed computing framework for Fog", 2017 IEEE Fog World Congress (FWC), DOI 10.1109/fwc.2017.8368528, October 2017, <<https://doi.org/10.1109/fwc.2017.8368528>>.

[Jones] Jones, D., Snider, C., Nassehi, A., Yon, J., and B. Hicks, "Characterising the Digital Twin: A systematic literature review", CIRP Journal of Manufacturing Science and Technology vol. 29, pp. 36-52, DOI 10.1016/j.cirpj.

2020.02.002, May 2020, <<https://doi.org/10.1016/j.cirpj.2020.02.002>>.

[Kelly] Kelly, R., "Internet of Things Data to Top 1.6 Zettabytes by 2022", Retrieved on 2022-05-24, 2015, <<https://campustechnology.com/articles/2015/04/15/internet-of-things-data-to-top-1-6-zettabytes-by-2020.aspx>>.

[Khan] Khan, L., Yaqoob, I., Tran, N., Kazmi, S., Dang, T., and C. Hong, "Edge-Computing-Enabled Smart Cities: A Comprehensive Survey", IEEE Internet of Things Journal vol. 7, no. 10, pp. 10200-10232, DOI 10.1109/jiot.2020.2987070, October 2020, <<https://doi.org/10.1109/jiot.2020.2987070>>.

[Larrea] Larrea, J. and A. Barbalace, "The serverkernel operating system", Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking, DOI 10.1145/3378679.3394537, April 2020, <<https://doi.org/10.1145/3378679.3394537>>.

[LFEDGE-EVE] Linux Foundation, "Project Edge Virtualization Engine (EVE)", Portal, retrieved on 2022-05-24, 2020, <<https://www.lfedge.org/projects/eve>>.

[Li] Li, Y., Chen, Y., Lan, T., and G. Venkataramani, "MobiQoR: Pushing the Envelope of Mobile Edge Computing Via Quality-of-Result Optimization", 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), DOI 10.1109/icdcs.2017.54, June 2017, <<https://doi.org/10.1109/icdcs.2017.54>>.

[Lin] Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications", IEEE Internet of Things Journal vol. 4, no. 5, pp. 1125-1142, DOI 10.1109/jiot.2017.2683200, October 2017, <<https://doi.org/10.1109/jiot.2017.2683200>>.

[Liu] Liu, J., Bai, B., Zhang, J., and K. Letaief, "Cache Placement in Fog-RANs: From Centralized to Distributed Algorithms", IEEE Transactions on Wireless Communications vol. 16, no. 11, pp. 7039-7051, DOI 10.1109/twc.2017.2737015, November 2017, <<https://doi.org/10.1109/twc.2017.2737015>>.

[Madni] Madni, A. M., Madni, C., and S. D. Lucero, "Leveraging digital twin technology in model-based systems engineering", Systems 7, no. 17, DOI 10.3390/systems7010007, 2019, <<https://doi.org/10.3390/systems7010007>>.

[Mahadev]

Satyanarayanan, M., "The Emergence of Edge Computing", Computer vol. 50, no. 1, pp. 30-39, DOI 10.1109/mc.2017.9, January 2017, <<https://doi.org/10.1109/mc.2017.9>>.

[Mortazavi]

Hossein Mortazavi, S., Balasubramanian, B., de Lara, E., and S. P. Narayanan, "Toward Session Consistency for the Edge", USENIX, Workshop on Hot Topics in Edge Computing (HotEdge 18), 2018, <<https://www.usenix.org/conference/hotedge18/presentation/mortazavi>>.

[Murshed]

Murshed, M., Murphy, C., Hou, D., Khan, N., Ananthanarayanan, G., and F. Hussain, "Machine Learning at the Network Edge: A Survey", ACM Computing Surveys vol. 54, no. 8, pp. 1-37, DOI 10.1145/3469029, November 2022, <<https://doi.org/10.1145/3469029>>.

[Nieke]

Nieke, M., Almstedt, L., and R. Kapitza, "Edgedancer: Secure Mobile WebAssembly Services on the Edge", Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, DOI 10.1145/3434770.3459731, April 2021, <<https://doi.org/10.1145/3434770.3459731>>.

[NIST]

Mell, P. and T. Grance, "The NIST definition of cloud computing", National Institute of Standards and Technology report, DOI 10.6028/nist.sp.800-145, 2011, <<https://doi.org/10.6028/nist.sp.800-145>>.

[NVIDIA]

Grzywaczewski, A., "Training AI for Self-Driving Vehicles: the Challenge of Scale", NVIDIA Developer Blog, retrieved on 2022-05-24, 2017, <<https://devblogs.nvidia.com/training-self-driving-vehicles-challenge-scale/>>.

[oneM2M-TR0001]

Mladin, C., "TR 0001, Use Cases Collection", oneM2M, October 2018, <<https://member.onem2m.org/Application/documentapp/downloadLatestRevision/default.aspx?docID=28153>>.

[oneM2M-TR0018]

Lu, C. and M. Jiang, "TR 0018, Industrial Domain Enablement", oneM2M, February 2019, <<https://member.onem2m.org/Application/documentapp/downloadLatestRevision/default.aspx?docID=29334>>.

[oneM2M-TR0026]

Yamamoto, K., Mladin, C., and V. Kueh, "TR 0026, Vehicular Domain Enablement", oneM2M, January 2020, <<https://member.onem2m.org/Application/documentapp/downloadLatestRevision/default.aspx?docID=31410>>.

[oneM2M-TR0052]

Yamamoto, K. and C. Mladin, "TR 0052, Study on Edge and Fog Computing in oneM2M systems", oneM2M, September

2020, <<https://member.onem2m.org/Application/documentapp/downloadLatestRevision/default.aspx?docID=32633>>.

[oneM2M-TS0002] He, S., "TS 0002, Requirements", oneM2M, February 2019, <<https://member.onem2m.org/Application/documentapp/downloadLatestRevision/default.aspx?docID=29274>>.

[OpenFog] "OpenFog Reference Architecture for Fog Computing", OpenFog Consortium, 2017, <https://iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf>.

[PseudoDynamicTesting] Ficco, M., Esposito, C., Xiang, Y., and F. Palmieri, "Pseudo-Dynamic Testing of Realistic Edge-Fog Cloud Ecosystems", IEEE Communications Magazine vol. 55, no. 11, pp. 98-104, DOI 10.1109/mcom.2017.1700328, November 2017, <<https://doi.org/10.1109/mcom.2017.1700328>>.

[Renart] Renart, E., Diaz-Montes, J., and M. Parashar, "Data-Driven Stream Processing at the Edge", 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), DOI 10.1109/icfec.2017.18, May 2017, <<https://doi.org/10.1109/icfec.2017.18>>.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, DOI 10.17487/RFC6291, June 2011, <<https://www.rfc-editor.org/rfc/rfc6291>>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/rfc/rfc7252>>.

[RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, DOI 10.17487/RFC7390, October 2014, <<https://www.rfc-editor.org/rfc/rfc7390>>.

[RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <<https://www.rfc-editor.org/rfc/rfc8578>>.

[Schafer] Schafer, D., Edinger, J., VanSyckel, S., Paluska, J., and C. Becker, "Tasklets: Overcoming Heterogeneity in Distributed Computing Systems", 2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW), DOI 10.1109/icdcs.2016.22, June 2016, <<https://doi.org/10.1109/icdcs.2016.22>>.

[Senel] Senel, B., Mouchet, M., Cappos, J., Fourmaux, O., Friedman, T., and R. McGeer, "EdgeNet: A Multi-Tenant and

Multi-Provider Edge Cloud", Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, DOI 10.1145/3434770.3459737, April 2021, <<https://doi.org/10.1145/3434770.3459737>>.

[Shi] Shi, W., Cao, J., Zhang, Q., Li, Y., and L. Xu, "Edge Computing: Vision and Challenges", IEEE Internet of Things Journal vol. 3, no. 5, pp. 637-646, DOI 10.1109/jiot.2016.2579198, October 2016, <<https://doi.org/10.1109/jiot.2016.2579198>>.

[SimulatingFog]

Svorobej, S., Takako Endo, P., Bendeche, M., Filelis-Papadopoulos, C., Giannoutakis, K., Gravvanis, G., Tzovaras, D., Byrne, J., and T. Lynn, "Simulating Fog and Edge Computing Scenarios: An Overview and Research Challenges", Future Internet vol. 11, no. 3, pp. 55, DOI 10.3390/fi11030055, February 2019, <<https://doi.org/10.3390/fi11030055>>.

[Stanciu] Stanciu, V., Steen, M., Dobre, C., and A. Peter, "Privacy-Preserving Crowd-Monitoring Using Bloom Filters and Homomorphic Encryption", Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, DOI 10.1145/3434770.3459735, April 2021, <<https://doi.org/10.1145/3434770.3459735>>.

[Weiner] Weiner, M., Jorgovanovic, M., Sahai, A., and B. Nikolic, "Design of a low-latency, high-reliability wireless communication system for control applications", 2014 IEEE International Conference on Communications (ICC), DOI 10.1109/icc.2014.6883918, June 2014, <<https://doi.org/10.1109/icc.2014.6883918>>.

[Yangui] Yangui, S., Ravindran, P., Bibani, O., Glitho, R., Ben Hadj-Alouane, N., Morrow, M., and P. Polakos, "A platform as-a-service for hybrid cloud/fog environments", 2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), DOI 10.1109/lanman.2016.7548853, June 2016, <<https://doi.org/10.1109/lanman.2016.7548853>>.

[Yates] Yates, R. and S. Kaul, "The Age of Information: Real-Time Status Updating by Multiple Sources", IEEE Transactions on Information Theory vol. 65, no. 3, pp. 1807-1827, DOI 10.1109/tit.2018.2871079, March 2019, <<https://doi.org/10.1109/tit.2018.2871079>>.

[Yousefpour]

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., and J. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey", Journal of Systems Architecture vol. 98, pp. 289-330, DOI 10.1016/

j.sysarc.2019.02.009, September 2019, <<https://doi.org/10.1016/j.sysarc.2019.02.009>>.

[Zhang] Zhang, Q., Zhang, X., Zhang, Q., Shi, W., and H. Zhong, "Firework: Big Data Sharing and Processing in Collaborative Edge Environment", 2016 Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), DOI 10.1109/hotweb.2016.12, October 2016, <<https://doi.org/10.1109/hotweb.2016.12>>.

[Zhang2] Zhang, J., Chen, B., Zhao, Y., Cheng, X., and F. Hu, "Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues", IEEE Access vol. 6, pp. 18209-18237, DOI 10.1109/access.2018.2820162, 2018, <<https://doi.org/10.1109/access.2018.2820162>>.

[_60802] IEC/IEEE, "Use Cases IEC/IEEE 60802 V1.3", IEC/IEEE 60802, 2018, <<https://grouper.ieee.org/groups/802/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf>>.

Authors' Addresses

Jungha Hong
ETRI
218 Gajeong-ro, Yuseung-Gu
Daejeon
34129
Republic of Korea

Email: jhong@etri.re.kr

Yong-Geun Hong
Daejeon University
62 Daehak-ro, Dong-gu
Daejeon
300716
Republic of Korea

Email: yonggeun.hong@gmail.com

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 // 3.0G
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
Hong Kong University of Science and Technology (Guangzhou)
No.1 Du Xue Rd
Guangzhou
China

Email: ietf@dkutscher.net