

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 15 July 2022

J. Hong  
ETRI  
Y-G. Hong  
Daejeon University  
X. de Foy  
InterDigital Communications, LLC  
M. Kovatsch  
Huawei Technologies Duesseldorf GmbH  
E. Schooler  
Intel  
D. Kutscher  
University of Applied Sciences Emden/Leer  
11 January 2022

IoT Edge Challenges and Functions  
draft-irtf-t2trg-iot-edge-04

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.

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 15 July 2022.

Internet-Draft

IoT Edge Computing

January 2022

## Copyright Notice

Copyright (c) 2022 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. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Revised BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Background . . . . .	<a href="#">3</a>
<a href="#">2.1.</a>	Internet of Things (IoT) . . . . .	<a href="#">3</a>
<a href="#">2.2.</a>	Cloud Computing . . . . .	<a href="#">4</a>
<a href="#">2.3.</a>	Edge Computing . . . . .	<a href="#">4</a>
<a href="#">2.4.</a>	Examples of IoT Edge Computing Use Cases . . . . .	<a href="#">6</a>
<a href="#">3.</a>	IoT Challenges Leading Towards Edge Computing . . . . .	<a href="#">9</a>
<a href="#">3.1.</a>	Time Sensitivity . . . . .	<a href="#">9</a>
<a href="#">3.2.</a>	Connectivity Cost . . . . .	<a href="#">10</a>
<a href="#">3.3.</a>	Resilience to Intermittent Services . . . . .	<a href="#">10</a>
<a href="#">3.4.</a>	Privacy and Security . . . . .	<a href="#">10</a>
<a href="#">4.</a>	IoT Edge Computing Functions . . . . .	<a href="#">11</a>
<a href="#">4.1.</a>	Overview of IoT Edge Computing Today . . . . .	<a href="#">11</a>
<a href="#">4.2.</a>	General Model . . . . .	<a href="#">13</a>
<a href="#">4.3.</a>	OAM Components . . . . .	<a href="#">16</a>
<a href="#">4.3.1.</a>	Resource Discovery and Authentication . . . . .	<a href="#">16</a>
<a href="#">4.3.2.</a>	Edge Organization and Federation . . . . .	<a href="#">17</a>
<a href="#">4.3.3.</a>	Multi-Tenancy and Isolation . . . . .	<a href="#">18</a>
<a href="#">4.4.</a>	Functional Components . . . . .	<a href="#">18</a>
<a href="#">4.4.1.</a>	In-Network Computation . . . . .	<a href="#">18</a>
<a href="#">4.4.2.</a>	Edge Storage and Caching . . . . .	<a href="#">20</a>
<a href="#">4.4.3.</a>	Northbound/Southbound Communication . . . . .	<a href="#">20</a>
<a href="#">4.4.4.</a>	Communication Brokering . . . . .	<a href="#">21</a>
<a href="#">4.5.</a>	Application Components . . . . .	<a href="#">21</a>
<a href="#">4.5.1.</a>	IoT End Devices Management . . . . .	<a href="#">21</a>
<a href="#">4.5.2.</a>	Data Management and Analytics . . . . .	<a href="#">22</a>
<a href="#">4.6.</a>	Simulation and Emulation Environments . . . . .	<a href="#">23</a>

<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">23</a>
<a href="#">6.</a>	Acknowledgements . . . . .	<a href="#">24</a>
<a href="#">7.</a>	Informative References . . . . .	<a href="#">24</a>
	Authors' Addresses . . . . .	<a href="#">31</a>

## [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 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, cloud services).

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, connectivity 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 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](#)].

## [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.

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](#)]. This limited storage and processing power leads to complementing IoT with cloud computing.

## [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 Industrial IoT Consortium (IIC, 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-thing 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 use 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 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. 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, that measure 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\*

The self-driving car, with its focus on safety, is a system where edge computing has an essential role. Autonomous vehicles are equipped with high-resolution cameras, radars, laser scanners (LIDAR), sonar sensors, and GPS systems. Edge computing nodes collect and analyze vast amounts of data generated in real-time by these sensors to keep track of distances between vehicles in front, surrounding road conditions, vehicle flow, and to quickly respond to unexpected situations. For example, if the speed of the car running in front decreases, speed should be adjusted to maintain the distance between the cars, and when a roadside signal changes, a self-driving car should operate according to the new signal. If such processing is performed in a central data center, network delays or data transmission errors can lead to accidents. Applying edge computing can minimize these network delays and data transmission errors, thereby improving safety. In the shorter term we can expect edge computing nodes to be at the base station or in road-side units. However, to further reduce reaction times, some edge computing nodes should be located in the vehicle itself.

#### \*AR/VR\*

Augmented Reality (AR) and Virtual Reality (VR) are likely to strongly influence the Information and Communication Technology (ICT) market in the future, since they can support innovative products in most other use cases including smart factories, self-driving cars, etc. In AR/VR, due to large amounts of data generated at endpoints such as mobile devices and PCs, user immersion can be significantly decreased by a latency of only a few hundred milliseconds. Therefore, using an edge computing infrastructure built close to endpoints can not only reduce the cost and latency of data transmission but also maximize user immersion. For example, in AR using edge computing, streaming video can be displayed realistically in higher quality, giving users the best possible experience.

#### \*Other Use Cases\*

Additionally, oneM2M recently studied several use cases related to edge computing, including: smart factories, smart transportation, an accident notification service, a high-precision road map service, a vulnerable road user service and a vehicular data service. These use

---

cases are documented in [[oneM2M-TR0001](#)], [[oneM2M-TR0018](#)] and [[oneM2M-TR0026](#)]. 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 for IoT. 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 jitter 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 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 terabytes 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.

### [3.3.](#) 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. 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. 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 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

In this section, we 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 Bosh, 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.

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 (such as StarlingX, Akraino Edge Stack, Mobile EdgeX) 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 an NFV-like management and orchestration model.

Resilience in IoT often entails 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.

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., Named

Function Networking (NFN) nodes can store data in a Named Data Networking (NDN) system, or a distributed storage platform (e.g., IPFS, EdgeFS, Ceph). 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).

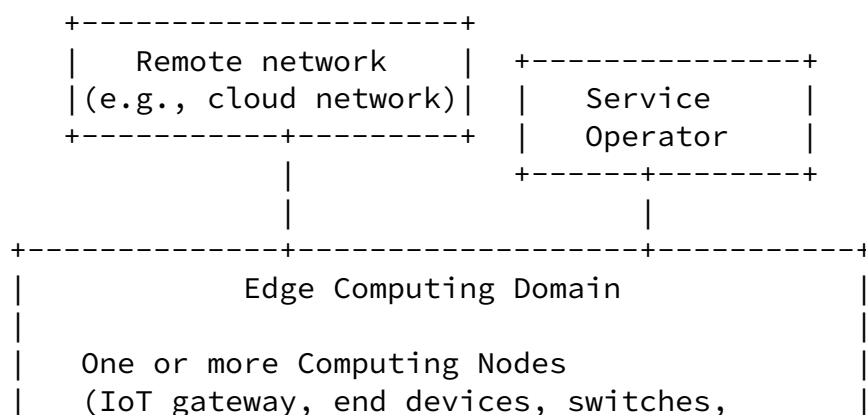
#### [4.2.](#) General Model

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 end 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.

In the general model described in Figure 1, 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, at the network edge, or through some interworking between edge network and remote cloud network.







prediction (inference) service is executed in the IoT end devices.

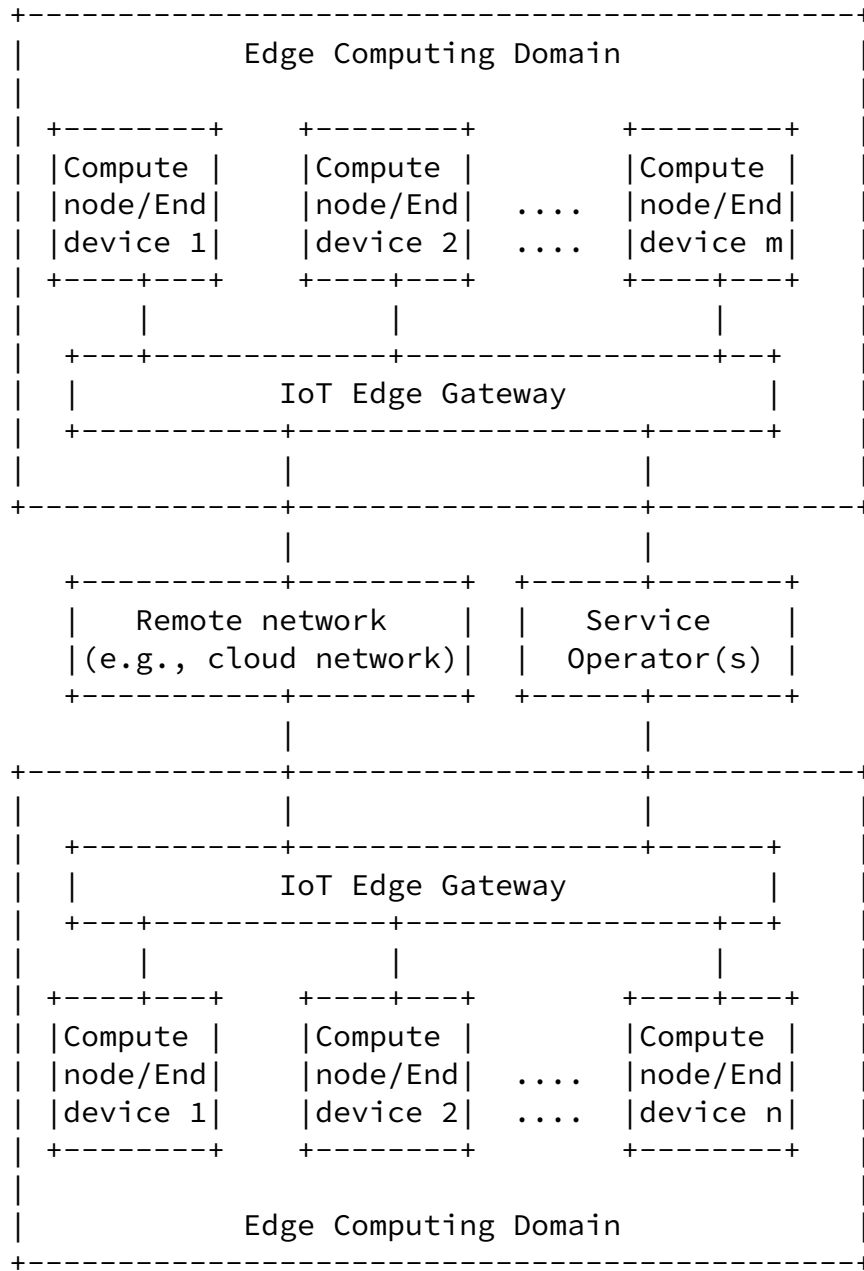


Figure 2: Example: Machine Learning over a Distributed IoT Edge Computing System

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 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. Discovery during normal operation (e.g., discovery of compute 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 end 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 end 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 end 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-oscore-groupcomm](#)])
- \* 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

#### [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, 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 applications lifecycle. Support for in-network computation may vary in term 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 include 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 end 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. 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 [[I-D.sarathchandra-coin-appcentres](#)]
- \* 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, 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). 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.](#) Northbound/Southbound 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, 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 end device 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.

#### [4.4.4.](#) 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

[[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:

- \* Enabling secure and resilient communication between IoT end 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 end device management, data management, described below.

##### [4.5.1.](#) 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 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 dash board 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 end 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
- \* 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.

#### [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](#)).

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 anonymizing, aggregating or compressing 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. Acknowledgements

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

## 7. Informative References

[AdvantEDGE]

"Mobile Edge Emulation Platform", Source Code Repository , 2020, <<https://github.com/InterDigitalInc/AdvantEDGE>>.

[Anglano] Anglano, C., Canonico, M., Castagno, P., Guazzone, M., and M. Sereno, "A game-theoretic approach to coalition formation in fog provider federations", IEEE Third International Conference on Fog and Mobile Edge Computing (FMEC), pages 123-130 , 2018.

[Ashton] Ashton, K., "That Internet of Things thing", RFID J. vol. 22, no. 7, pp. 97-114 , 2009.

[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, 4(3):772-782 , 2017.

[Botta] Botta, A., Donato, W., Persico, V., and A. Pescapé, "Integration of Cloud Computing and Internet of Things: A survey", Future Gener. Comput. Syst., vol. 56, pp. 684-700 , 2016.

[Cao] Cao, L., Merican, A., Zad Tootaghaj, D., Ahmed, F., Sharma, P., and V. Saxena, "ECaaS: A Management Framework of Edge Container as a Service for Business Workload", 4th International Workshop on Edge Systems, Analytics and Networking , 2021, <<https://doi.org/10.1145/3434770.3459741>>.

[Chen] Baotong Chen, ., Jiafu Wan, ., Antonio Celesti, ., Di Li, ., Haider Abbas, ., and . Qin Zhang, "Edge computing in

IoT-based manufacturing", IEEE Communications Magazine , 2018, <<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8466364>>.

[Chiang] Chiang, M. and T. Zhang, "Fog and IoT: An overview of research opportunities", IEEE Internet Things J., vol. 3, no. 6, pp. 854–864 , 2016.

[Cloudlets]

Mahadev Satyanarayanan, ., Paramvir Bahl, ., Ramón Cáceres, ., and . Nigel Davies, "The Case for VM-Based Cloudlets in Mobile Computing", IEEE Pervasive Computing , 2009, <<https://ieeexplore.ieee.org/document/5280678>>.

[Echeverria]

Echeverria, S., Klinedinst, D., Williams, K., and G. A Lewis, "Establishing trusted identities in disconnected edge environments", IEEE/ACM Symposium Edge Computing (SEC), pages 51–63. , 2016.

[ENERGY] Beckel, C., Sadamori, L., Staaake, T., and S. Santini, "Revealing Household Characteristics from Smart Meter Data", Energy, vol. 78, pp. 397–410 , 2014.

[ETSI\_MEC\_01]

ETSI, ., "Multi-access Edge Computing (MEC); Terminology", ETSI GS 001 , 2019, <[https://www.etsi.org/deliver/etsi\\_gs/MEC/001\\_099/001/02.01.01\\_60/gs\\_MEC001v020101p.pdf](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 003 , 2019, <[https://www.etsi.org/deliver/etsi\\_gs/MEC/001\\_099/003/02.01.01\\_60/gs\\_MEC003v020101p.pdf](https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.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", IEEE International Conference on Communications (ICC), pages 1-6 , 2017.
- [Harchol] Harchol, Y., Mushtaq, A., McCauley, J., Panda, A., and S. Shenker, "Cessna: Resilient edge-computing", Workshop on Mobile Edge Communications, pages 1-6. ACM , 2018.
- [I-D-defoy-t2trg-iot-edge-computing-background] de Foy, X., Hong, J., Hong, Y., Kovatsch, M., Schooler, E., and D. Kutscher, "Machine learning at the network

Hong, et al.

Expires 15 July 2022

[Page 25]

---

Internet-Draft

IoT Edge Computing

January 2022

edge: A survey", [draft-defoy-t2trg-iot-edge-computing-background-00](#) , 2020, <<http://www.ietf.org/internet-drafts/draft-defoy-t2trg-iot-edge-computing-background-00.txt>>.

- [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.ietf-core-oscore-groupcomm] Tiloca, M., Selander, G., Palombini, F., Mattsson, J. P., and J. Park, "Group OSCORE - Secure Group Communication for CoAP", Work in Progress, Internet-Draft, [draft-ietf-core-oscore-groupcomm-13](#), 25 October 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-core-oscore-groupcomm-13>>.
- [I-D.mcbride-edge-data-discovery-overview] McBride, M., Kutscher, D., Schooler, E., Bernardos, C. J., Lopez, D. R., and X. D. 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](https://datatracker.ietf.org/doc/html/draft-sarathchandra-coin-appcentres-04), 26 January 2021,  
<<https://datatracker.ietf.org/doc/html/draft-sarathchandra-coin-appcentres-04>>.
- [ISO\_TR] "Information Technology - Cloud Computing - Edge Computing  
Landscape", ISO/IEC TR 23188 , 2018.
- [Jeffery] Jeffery, A., Howard, H., and R. Mortier, "Rearchitecting  
Kubernetes for the Edge", 4th International Workshop on  
Edge Systems, Analytics and Networking , 2021,  
<<https://dl.acm.org/doi/10.1145/3434770.3459730>>.
- [Jeong] Jeong, T., Chung, J., Hong, J.W., and S. Ha, "Towards a  
distributed computing framework for fog", IEEE Fog World  
Congress (FWC), pages 1-6 , 2017.

Hong, et al.

Expires 15 July 2022

[Page 26]

Internet-Draft

IoT Edge Computing

January 2022

- [Jones] David Jones, ., Chris Snider, ., Aydin Nassehi, ., Jason  
Yon, ., and . Ben Hicks, "Characterising the Digital Twin:  
A systematic literature review", CIRP Journal of  
Manufacturing Science and Technology , 2020,  
<<https://www.sciencedirect.com/science/article/pii/S1755581720300110>>.
- [Kelly] Kelly, R., "Internet of Things Data to Top 1.6 Zettabytes  
by 2022", 2015,  
<<https://campustechnology.com/articles/2015/04/15/internet-of-things-data-to-top-1-6-zettabytes-by-2020.aspx>>.
- [Khan] Khan, L.U., Yaqoob, I., Tran, N.H., Kazmi, S.M.A., Dang,  
T.N., and C.S. Hong, "Edge Computing Enabled Smart Cities:  
A Comprehensive Survey", arXiv:1909.08747 , 2019.
- [Larrea] Larrea, J. and A. Barbalace, "The serverkernel operating  
system", Third ACM International Workshop on Edge Systems,  
Analytics and Networking , 2020,  
<<https://core.ac.uk/reader/327124532>>.

[LFEDGE-EVE]

Linux Foundation, ., "Project Edge Virtualization Engine (EVE)", Portal , 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", IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pages 1261-1270 , 2017.

[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 J., vol. 4, no. 5, pp. 1125-1142 , 2017.

[Liu]

Liu, J., Bai, B., Zhang, J., and K.B. Letaief, "Cache placement in fog-rans: From centralized to distributed algorithms", IEEE Transactions on Wireless Communications, 16(11):7039-7051 , 2017.

[Mahadev]

Satyanarayanan, M., "The Emergence of Edge Computing", Computer, vol. 50, no. 1, pp. 30-39 , 2017.

[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", arXiv:1908.00080 , 2019.

[Nieke]

Nieke, M., Almstedt, L., and R. Kapitza, "Edgedancer: Secure Mobile WebAssembly Services on the Edge", 4th

International Workshop on Edge Systems, Analytics and Networking , 2021,  
<<https://doi.org/10.1145/3434770.3459731>>.

[NIST] Mell, P. and T. Grance, "The NIST definition of Cloud Computing", Natl. Inst. Stand. Technol, vol. 53, no. 6, p. 50 , 2009.

[NVIDIA] Grzywaczewski, A., "Training AI for Self-Driving Vehicles: the Challenge of Scale", NVIDIA Developer Blog , 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.

[PseudoDynamicTesting]

Ficco, M., Esposito, C., Xiang, Y., and F. Palmieri, "Pseudo-Dynamic Testing of Realistic Edge-Fog Cloud Ecosystems", IEEE Communications Magazine, Nov. 2017 , 2017.

[Renart] Renart, E.G., Diaz-Montes, J., and M. Parashar, "Data-driven stream processing at the edge", IEEE 1st International Conference on Fog and Edge Computing (ICFEC), pages 31-40 , 2017.

[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://doi.org/10.17487/RFC6291>>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", [RFC 7252](#), DOI 10.17487/RFC7252, June 2014, <<https://doi.org/10.17487/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://doi.org/10.17487/RFC7390>>.

[Schafer] Schafer, D., Edinger, J., VanSyckel, S., Paluska, J.M., and C. Becker, "Tasklets: Overcoming Heterogeneity in Distributed Computing Systems", IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW), Nara, pp. 156-161 , 2016.

- [Senel] Senel, B., Mouchet, M., Cappos, J., Fourmaux, O., Friedman, T., and R. McGeer, "EdgeNet: A Multi-Tenant and Multi-Provider Edge Cloud", 4th International Workshop on Edge Systems, Analytics and Networking , 2021, <<https://dl.acm.org/doi/pdf/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 J., vol. 3, no. 5, pp. 637-646 , 2016.
- [SimulatingFog] Svorobej, S. and . al, "Simulating Fog and Edge Computing Scenarios: An Overview and Research Challenges", MPDI Future Internet 2019 , 2019.
- [Stanciu] Stanciu, V., van Steen, M., Dobre, C., and A. Peter, "Privacy-Preserving Crowd-Monitoring Using Bloom Filters and Homomorphic Encryption", 4th International Workshop on Edge Systems, Analytics and Networking , 2021, <<https://dl.acm.org/doi/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", IEEE Int. Conf. Commun. (ICC), Sydney, NSW, Australia, pp. 3829-3835 , 2014.
- [Yangui] Yangui, S., Ravindran, P., Bibani, O., H Glitho, R., Ben Hadj-Alouane, N., Morrow, M.J., and P.A. Polakos, "A platform as-a-service for hybrid cloud/fog environments", IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), pages 1-7 , 2016.
- [Yates] Yates, R.D. and S.K. Kaul, "The Age of Information: Real-Time Status Updating by Multiple Sources", IEEE Transactions on Information Theory, vol. 65, no. 3, pp. 1807-1827 , 2019.
- [Yousefpour] Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., and J.P. 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 , 2019.

Internet-Draft

IoT Edge Computing

January 2022

- [Zhang] Zhang, Q., Zhang, X., Zhang, Q., Shi, W., and H. Zhong, "Firework: Big data sharing and processing in collaborative edge environment", Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), pages 20-25 , 2016.
- [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 , 2018.
- [\_60802] IEC/IEEE, ., "Use Cases IEC/IEEE 60802 V1.3", IEC/IEEE 60802 , 2018, <<http://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0818-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

Hong, et al.

Expires 15 July 2022

[Page 31]

---

Internet-Draft

IoT Edge Computing

January 2022

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

Email: ietf@dkutscher.net

