Chapter 3

Types of Communications in IOT

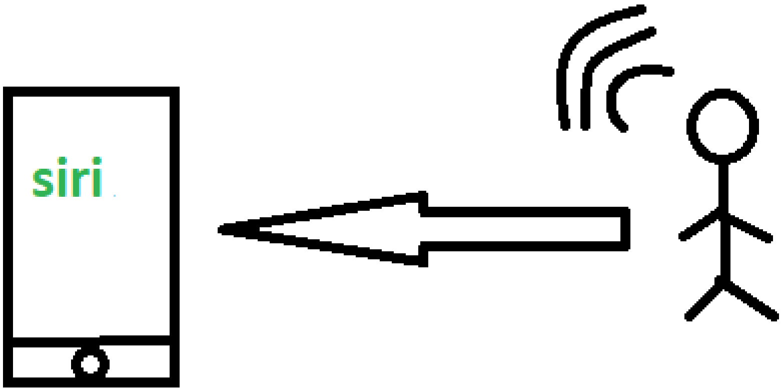
IoT Communication: IoT is the connection of devices over the internet, where these smart devices communicate with each other , exchange data , perform some tasks without any human involvement. These devices are embedded with electronics, software, network and sensors which help in communication. Communication between smart devices is very important in IOT as it enables these devices to gather, exchange data which contribute in success of that IOT product/project.

Types of Communications in IOT:

The following are some communication types in IoT:-

1. Human to Machine (H2M):

In this human gives input to IOT device i.e as speech/text/image etc. IOT device (Machine) like sensors and actuators then understands input, analyses it and responds back to human by means of text or Visual Display. This is very useful as these machines assist humans in every everyday tasks. It is a combo of software and hardware that includes human interaction with a machine to perform a task.



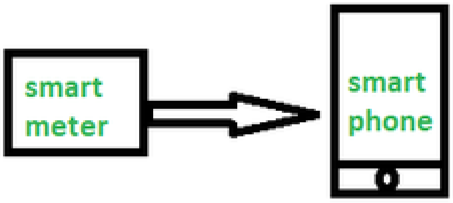
Merits: This H2M has a user-friendly interface that can be quickly accessed by following the instructions. It responds more quickly to any fault or failure. Its features and functions can be customized.

Examples:

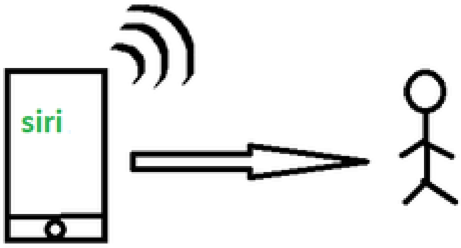
  - Facial recognition.
  - Bio-metric Attendance system.
  - Speech or voice recognition.
- **2. Machine to Machine (M2M):**
- The process of exchanging information or messages between two or more machines or devices is known as Machine to Machine (M2M) communication.

- It is the communication among the physical things which do not need human intervention

M2M communication is also named as Machine Type communication in 3GPP(3rd Generation Partnership Project).
In this the interaction or communication takes place between machines by automating data/programs. In this machine level instructions are required for communication. Here communication takes place without human interaction. The machines may be either connected through wires or by wireless connection. An M2M connection is a point-to-point connection between two network devices that helps in transmitting information using public networking technologies like Ethernet and cellular networks. IoT uses the basic concepts of M2M and expands by creating large “cloud” networks of devices that communicate with one another through cloud networking platforms.



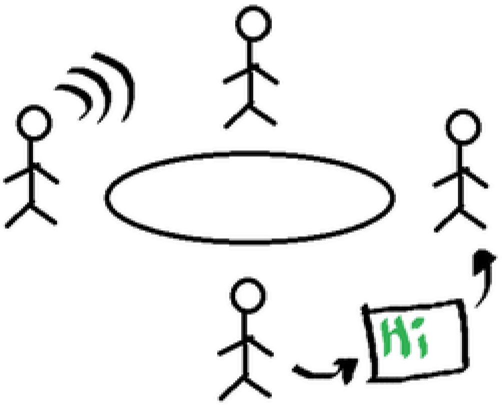
3. Machine to Human (M2H) :
In this machine interacts with Humans. Machine triggers information(text messages/images/voice/signals) respective / irrespective of any human presence. This type of communication is most commonly used where machines guide humans in their daily life. It is way of interaction in which humans co-work with smart systems and other machines by using tools or devices to finish a task.



Examples:

- Fire Alarms
- Traffic Light
- Fitness bands
- Health monitoring devices

- **4. Human to Human (H2H) :**
  This is generally how humans communicate with each other to exchange information by speech, writing, drawing, facial expressions, body language etc. Without H2H, M2M applications cannot produce the expected benefits unless humans can immediately fix issues, solve challenges, and manage scenarios.
- The process of exchanging information
  or messages between two or more people is known as human to human (H2H) communication. This can be done through various means such as verbal,
  non-verbal, or written communication.



For, communication of IoT devices many protocols are used. These IoT protocols are modes of communication which give security to the data being exchanged between IoT connected devices. Example bluetooth, wifi, zigbee etc.

Internet of Things Communication Models

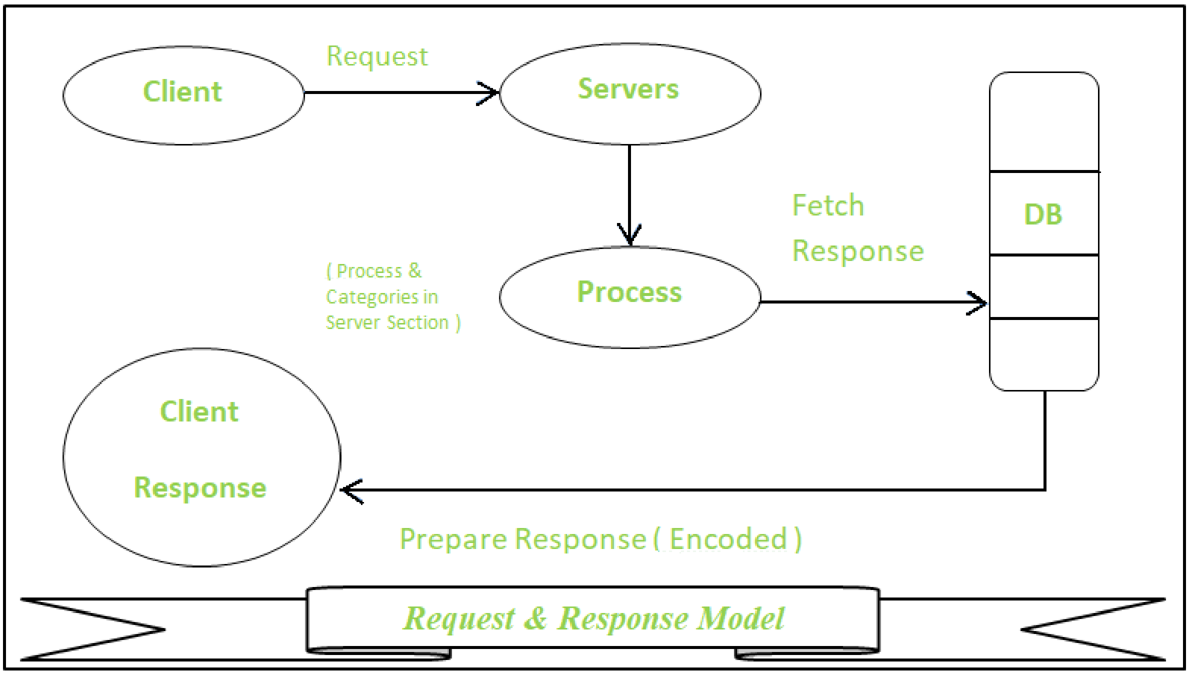
IoT devices are found everywhere and will enable circulatory intelligence in the future. For operational perception, it is important and useful to understand how various IoT devices communicate with each other. Communication models used in IoT have great value. The IoTs allow people and things to be connected any time, any space, with anything and anyone, using any network and any service.

Types of Communication Model :

1. Request & Response Model –

This model follows a client-server architecture.

- The **client**, when required, requests the information from the server. This request is usually in the encoded format.
- This model is stateless since the data between the requests is not retained and each request is independently handled.
- The server Categories the request, and fetches the data from the database and its resource representation. This data is converted to response and is transferred in an encoded format to the client. The client, in turn, receives the response.
- On the other hand — In **Request-Response** communication model client sends a request to the server and the server responds to the request. When the server receives the request it decides how to respond, fetches the data retrieves resources, and prepares the response, and sends it to the client.

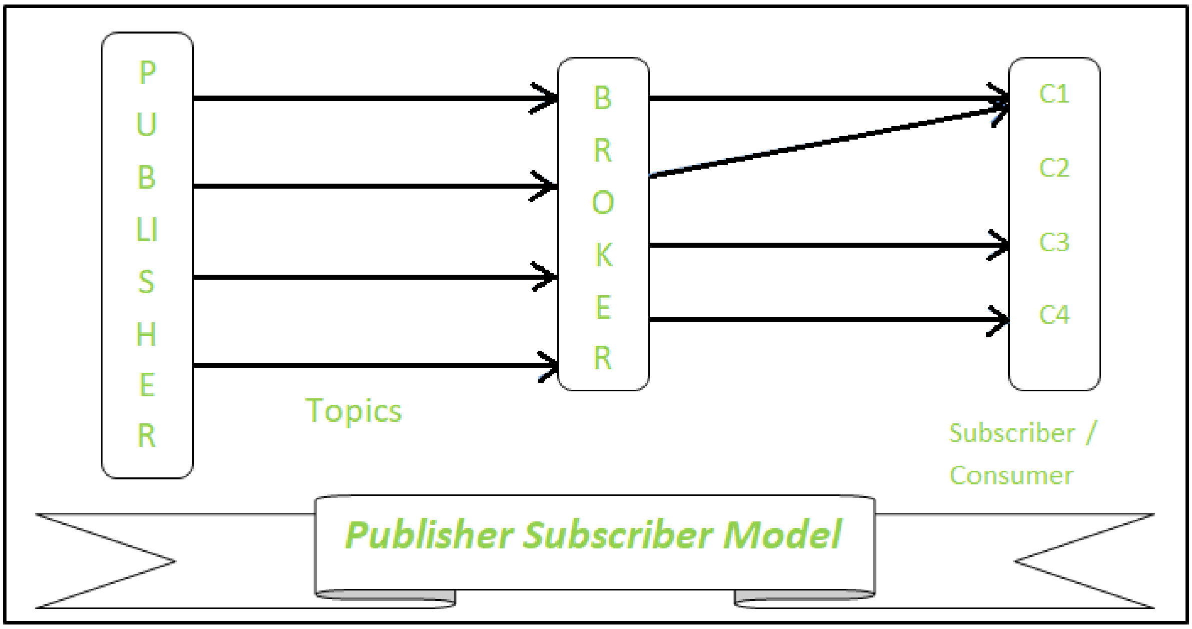


2. Publisher-Subscriber Model –

This model comprises three entities: Publishers, Brokers, and Consumers.

- **Publishers** are the source of data. It sends the data to the topic which are managed by the broker. They are
- not aware of consumers.

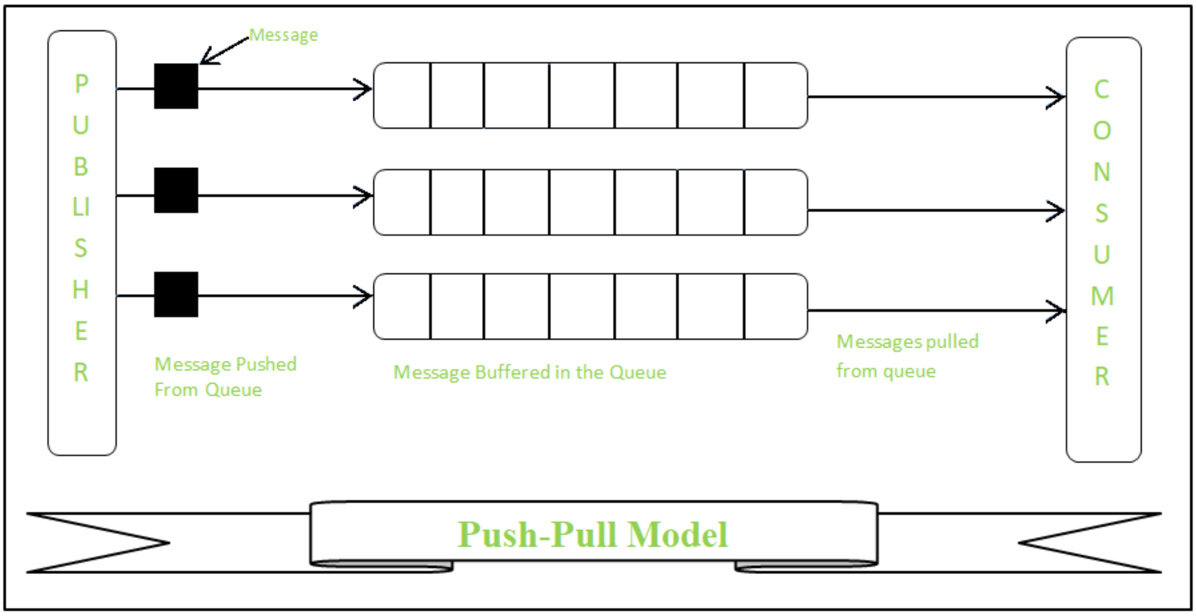
- **Consumers** subscribe to the topics which are managed by the broker.
- Hence, **Brokers** responsibility is to accept data from publishers and send it to the appropriate consumers. The broker only has the information regarding the consumer to which a particular topic belongs to which the publisher is unaware of.



3. Push-Pull Model –

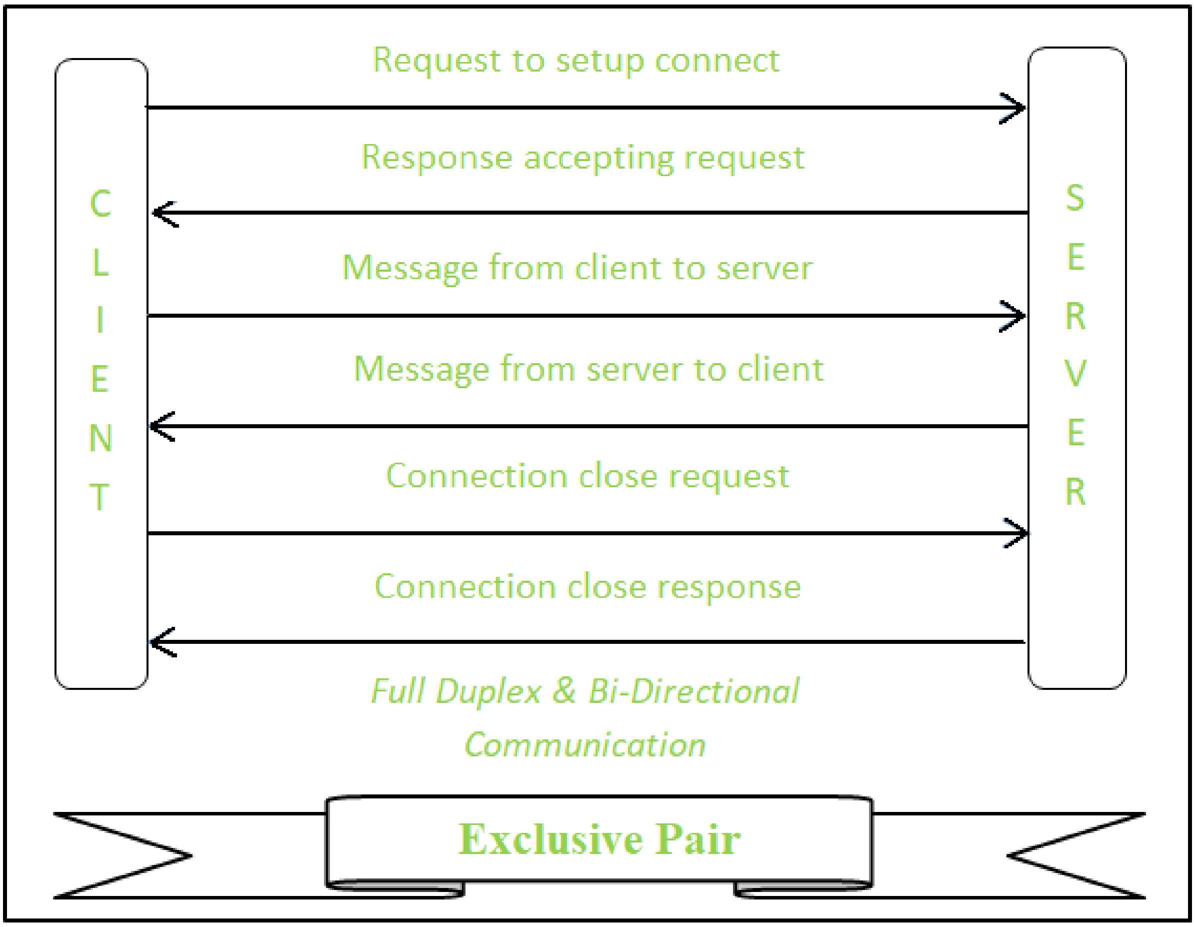
The push-pull model constitutes data publishers, data consumers, and data queues.

- **Publishers** and **Consumers** are not aware of each other.
- Publishers publish the message/data and push it into the queue. The consumers, present on the other side, pull the data out of the queue. Thus, the queue acts as the buffer for the message when the difference occurs in the rate of push or pull of data on the side of a publisher and consumer.
- **Queues** help in decoupling the messaging between the producer and consumer. Queues also act as a buffer which helps in situations where there is a mismatch between the rate at which the producers push the data and consumer
-
-
-
- s pull the data.


Message
PUBLISHER
CONSUMER
Message Pushed From Queue
Message Buffered in the Queue
Messages pulled from queue
Push-Pull Model

4. Exclusive Pair –

- **Exclusive Pair** is the bi-directional model, including full-duplex communication among client and server. The connection is constant and remains open till the client sends a request to close the connection.
- The **Server** has the record of all the connections which has been opened.
- This is a state-full connection model and the server is aware of all open connections.
- WebSocket based communication API is fully based on this model.

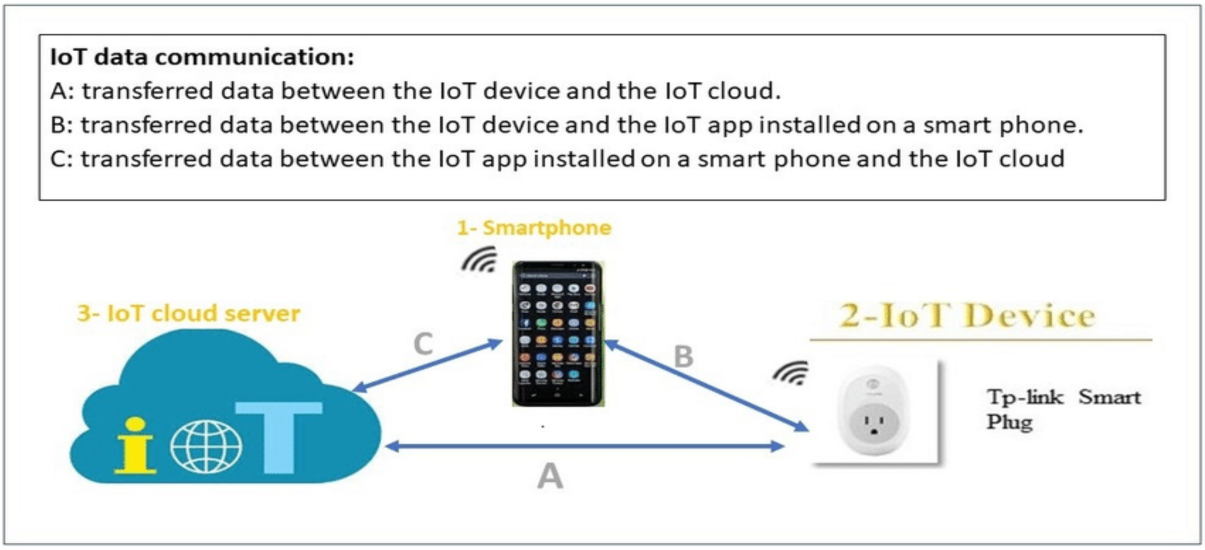

Request to setup connect
Response accepting request
Message from client to server
Message from server to client
Connection close request
Connection close response
CLIENT
SERVER
Full Duplex & Bi-Directional Communication
Exclusive Pair

The Four Internet of Things Connectivity Models Explained

The Four Internet of Things Connectivity Models Explained

At its most basic level, the Internet of Things is all about connecting various devices and sensors to the Internet, but it's not always obvious how to connect them.

Device-to-Device



IoT data communication:
A: transferred data between the IoT device and the IoT cloud.
B: transferred data between the IoT device and the IoT app installed on a smart phone.
C: transferred data between the IoT app installed on a smart phone and the IoT cloud

Device-to-device communication represents two or more devices that directly connect and communicate between one another. They can communicate over many types of networks, including IP networks or the Internet, but most often use protocols like Bluetooth, Z-Wave, and ZigBee.

This model is commonly used in home automation systems to transfer small data packets of information between devices at a relatively low data rate. This could be light bulbs, thermostats, and door locks sending small amounts of information to each other.

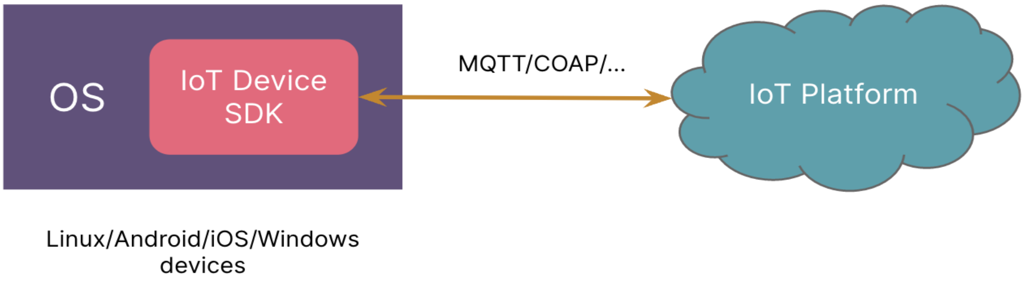
Each connectivity model has different characteristics, Tschofenig said. With Device-to-Device, he said "security is specifically simplified because you have these short-range radio technology [and a] one-to-one relationship between these two devices."

Device-to-device is popular among wearable IoT devices like a heart monitor paired to a smartwatch where data doesn't necessarily have be to shared with multiple people.

There are several standards being developed around Device-to-Device including Bluetooth Low Energy (also known as Bluetooth Smart or Bluetooth

Version 4.0+) which is popular among portable and wearable devices because its low power requirements could mean devices could operate for months or years on one battery. Its lower complexity can also reduce its size and cost.

Device-to-Cloud



Device-to-cloud communication involves an IoT device connecting directly to an Internet cloud service like an application service provider to exchange data and control message traffic. It often uses traditional wired Ethernet or Wi-Fi connections, but can also use cellular technology.

Cloud connectivity lets the user (and an application) to obtain remote access to a device. It also potentially supports pushing software updates to the device.

A use case for cellular-based Device-to-Cloud would be a smart tag that tracks your dog while you're not around, which would need wide-area cellular communication because you wouldn't know where the dog might be.

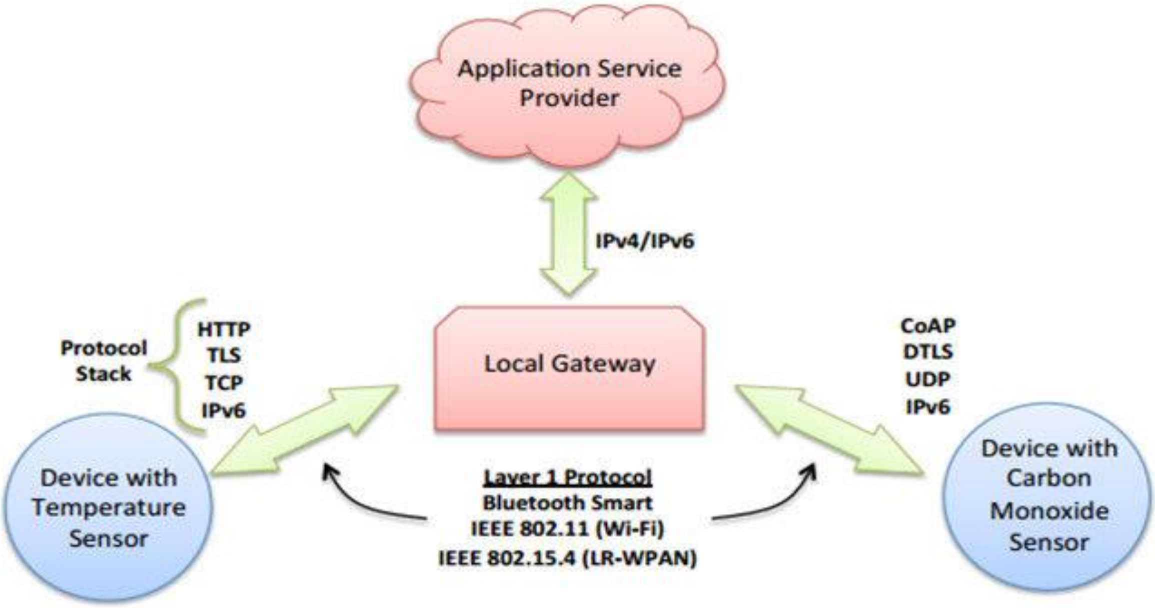
Another scenario, T schofenig said, would be remote monitoring with a product like the Drop cam, where you need the bandwidth provided by Wifi or Ethernet. But it also makes sense to push data into the cloud in this scenario because makes sense because it provides access to the user if they're away. "Specifically, if you're away and you want to see what's on your webcam at home. You contact the cloud infrastructure and then the cloud infrastructure relays to your IoT device."

From a security perspective, this gets more complicated than Device-to-Device because it involves two different types of credentials – the network access credentials (such as the mobile device's SIM card) and then the credentials for cloud access.

The IAB's report also mentioned that interoperability is also a factor with Device-to-Cloud when attempting to integrate devices made by different manufacturers given that the device and cloud service are typically from the same vendor. An example would be the Nest Labs Learning Thermostat, where the Learning Thermostat can only work with Nest's cloud service.

Tschofenig said there's work going into making Wifi devices that make cloud connections while consuming less power with standards such as LoRa, Sigfox, and Narrowband.

Device-to-Gateway



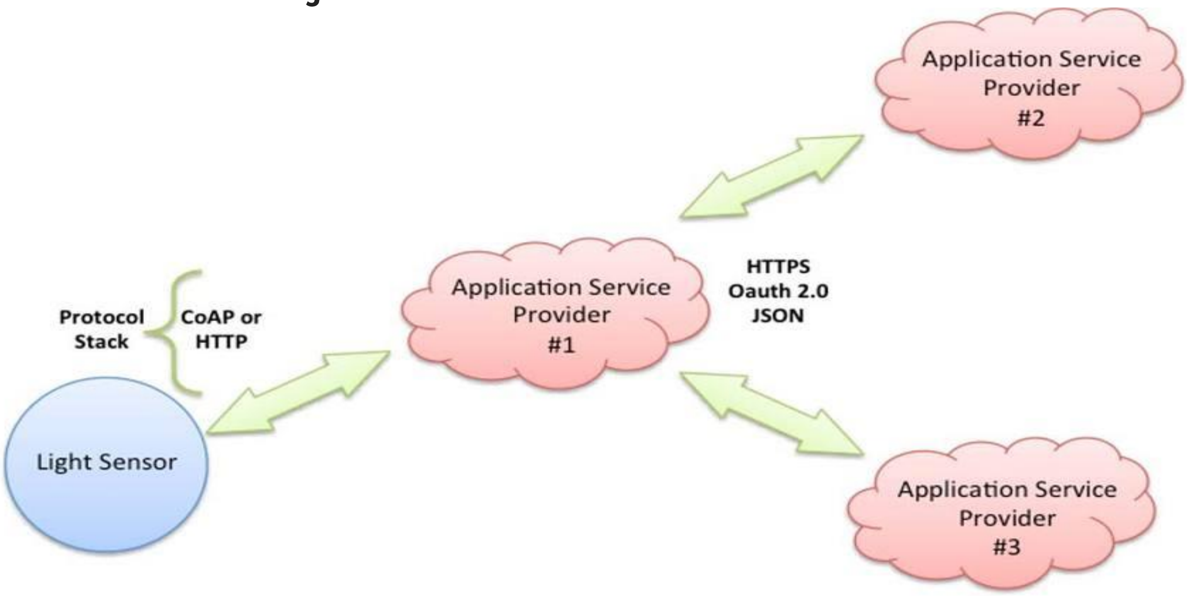
In the Device-to-Gateway model, IoT devices basically connect to an intermediary device to access a cloud service. This model often involves application software operating on a local gateway device (like a smartphone or a "hub") that acts as an intermediary between an IoT device and a cloud service.

This gateway could provide security and other functionality such as data or protocol translation. If the application-layer gateway is a smartphone, this application software might take the form of an app that pairs with the IoT device and communicates with a cloud service.

This might be a fitness device that connects to the cloud through a smartphone app like Nike+, or home automation applications that involve devices that connect to a hub like Samsung's SmartThings ecosystem. "Today, you more or less have to more or less buy a gateway from a dedicated vendor or use one of these mulit-purpose gateways," Tschofenig said. "You connect all your devices up to that gateway and it does something like data aggregation or transcoding, and it either hands [off the data] locally to the home or shuffles it off to the cloud, depending on the use case."

Gateway devices can also potentially bridge the interoperability gap between devices that communicate on different standards. For instance, SmartThings' Z-Wave and Zigbee transceivers can communicate with both families of devices.

4. **Backend Data Sharing**


Application Service Provider #2
Protocol Stack
CoAP or HTTP
Application Service Provider #1
HTTPS Oauth 2.0 JSON
Light Sensor
Application Service Provider #3

Back-End Data-Sharing essentially extends the single device-to-cloud communication model so that IoT devices and sensor data can be accessed by authorized third parties. Under this model, users can export and analyze smart object data from a cloud service in combination with data from other sources, and send it to other services for aggregation and analysis.

Tschofenig said the app Map My Fitness is a good example of this because it compiles fitness data from various devices ranging from the Fitbit to the Adidas miCoach to the Wahoo Bike Cadence Sensor. "They provide hooks, REST APIs to allow security and privacy-friendly data sharing to Map My Fitness." This means an exercise can be analyzed from the viewpoint of various sensors.

There's No Clear IoT Deployment Model; It All Depends on the Use Case

Tschofenig said that the decision process for IoT developers is quite complicated when considering how it will be integrated and how it will get connectivity to the internet working.

To further complicate things, newer technologies with lower power consumption, size and cost are often lacking in maturity compared to traditional Ethernet or Wi-Fi.

“The equation is not just what is most convenient for me, but what are the limitations of those radio technologies and how do I deal with factors like the size limitations, energy consumption, the cost – these aspects play a big