Internet Applications

Dr./ Ahmed Mohamed Rabie

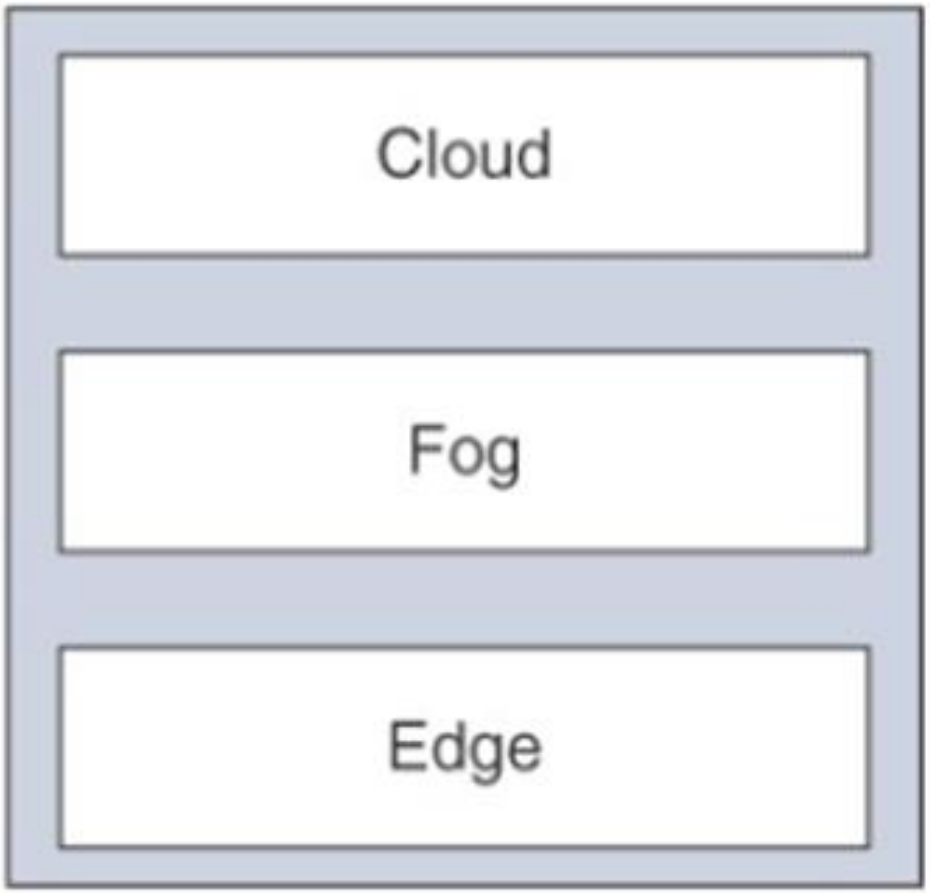
Chapter 1

Introduction

Internet ofThings

IoT Data Management and Compute Stack: Clearly, traditional IT networks are not prepared for this magnitude of network devices. However, beyond the network architecture itself, consider the data that is generated by these devices. If the number of devices is beyond conventional numbers, surely the data generated by these devices must also be of serious concern.

IoT Data Management and Compute Stack



In fact, the data generated by IoT sensors is one of the single biggest challenges in building an IoT system. In the case of modern IT networks, the data sourced by a computer or server is typically generated by the client/server communications model, and it serves the needs of the application. In sensor networks, the vast majority of data generated is unstructured and of very little use on its own.

For example, the majority of data generated by a smart meter is nothing more than polling data; the communications system simply determines whether a network connection to the meter is still active. This data on its own is of very little value. The real value of a smart meter is the metering data read by the meter management system (MMS). However, if you look at the raw polling data from a different perspective, the information can be very useful.


Smart Home
Advanced Metering Infrastructure
Utility Office
Smart Meter
Communication
Device

For example, a utility may have millions of meters covering its entire service area. If whole sections of the smart grid start to show an interruption of connectivity to the meters, this data can be analyzed and combined with other sources of data, such as weather reports and electrical demand in the grid, to provide a complete picture of what is happening. This information can help determine whether the loss of connection to the meters is truly a loss of power or whether some other problem has developed in the grid. Moreover, analytics of this data can help the utility quickly determine the extent of the service outage and repair the disruption in a timely fashion.

In most cases, the processing location is outside the smart object. A natural location for this processing activity is the cloud. Smart objects need to connect to the cloud, and data processing is centralized. One advantage of this model is simplicity. Objects just need to connect to a central cloud application. That application has visibility over all the IoT nodes and can process all the analytics needed today and in the future. However, this model also has limitations. As data volume, the variety of objects connecting to the network, and the need for more efficiency increase, new requirements appear, and those requirements tend to bring the need for data analysis closer to the IoT system.

These new requirements include the following:

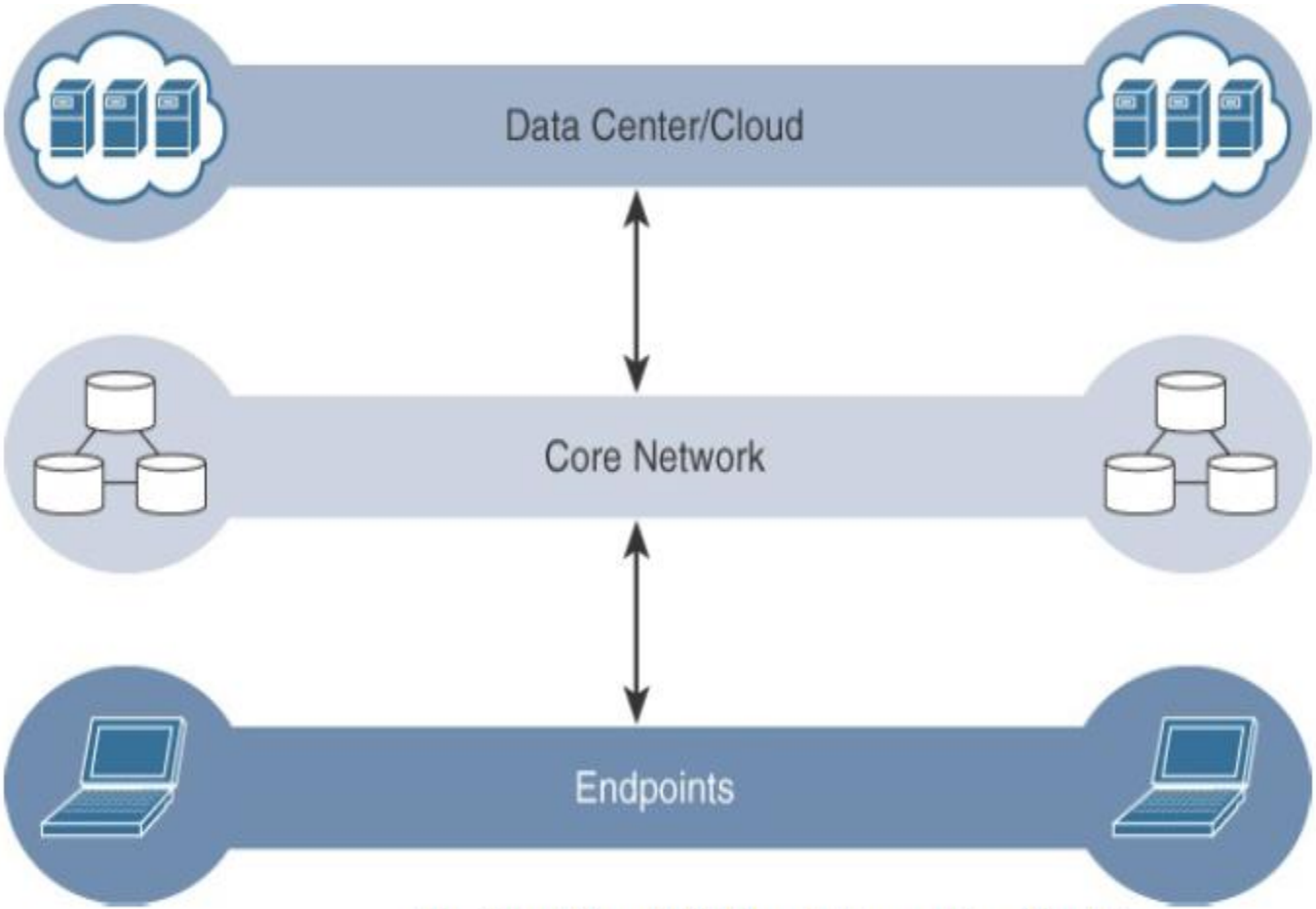
- Minimizing latency: Milliseconds matter for many types of industrial systems, such as when you are trying to prevent manufacturing line shutdowns or restore electrical service. Analyzing data close to the device that collected the data can make a difference between averting disaster and a cascading system failure.

- Conserving network bandwidth: Offshore oil rigs generate 500 GB of data weekly. Commercial jets generate 10 TB for every 30 minutes of flight. It is not practical to transport vast amounts of data from thousands or hundreds of thousands of edge devices to the cloud. Nor is it necessary because many critical analyses do not require cloud-scale processing and storage.

- Increasing local efficiency: Collecting and securing data across a wide geographic area with different environmental conditions may not be useful. The environmental conditions in one area will trigger a local response independent from the conditions of another site hundreds of miles away. Analyzing both areas in the same cloud system may not be necessary for immediate efficiency.

An important design consideration, therefore, is how to design an IoT network to manage this volume of data in an efficient way such that the data can be quickly analyzed and lead to business benefits. The volume of data generated by IoT devices can be so great that it can easily overrun the capabilities of the headend system in the data center or the cloud. For example, it has been observed that a moderately sized smart meter network of 1 million meters will generate close to 1 billion data points each day (including meter reads and other instrumentation data), resulting in 1 TB of data. For an IT organization that is not prepared to contend with this volume of data storage and real-time analysis, this creates a whole new challenge.

The volume of data also introduces questions about bandwidth management. As the massive amount of IoT data begins to funnel into the data center, does the network have the capacity to sustain this volume of traffic? Does the application server have the ability to ingest, store, and analyze the vast quantity of data that is coming in? This is sometimes referred to as the "impedance mismatch" of the data generated by the IoT system and the management application's ability to deal with that data.



The Traditional IT Cloud Computing Model

As illustrated data management in traditional IT systems is very simple. The endpoints (laptops, printers, IP phones, and so on) communicate over an IP core network to servers in the data center or cloud. Data is generally stored in the data center, and the physical links from access to core are typically high bandwidth, meaning access to IT data is quick IoT systems function differently.

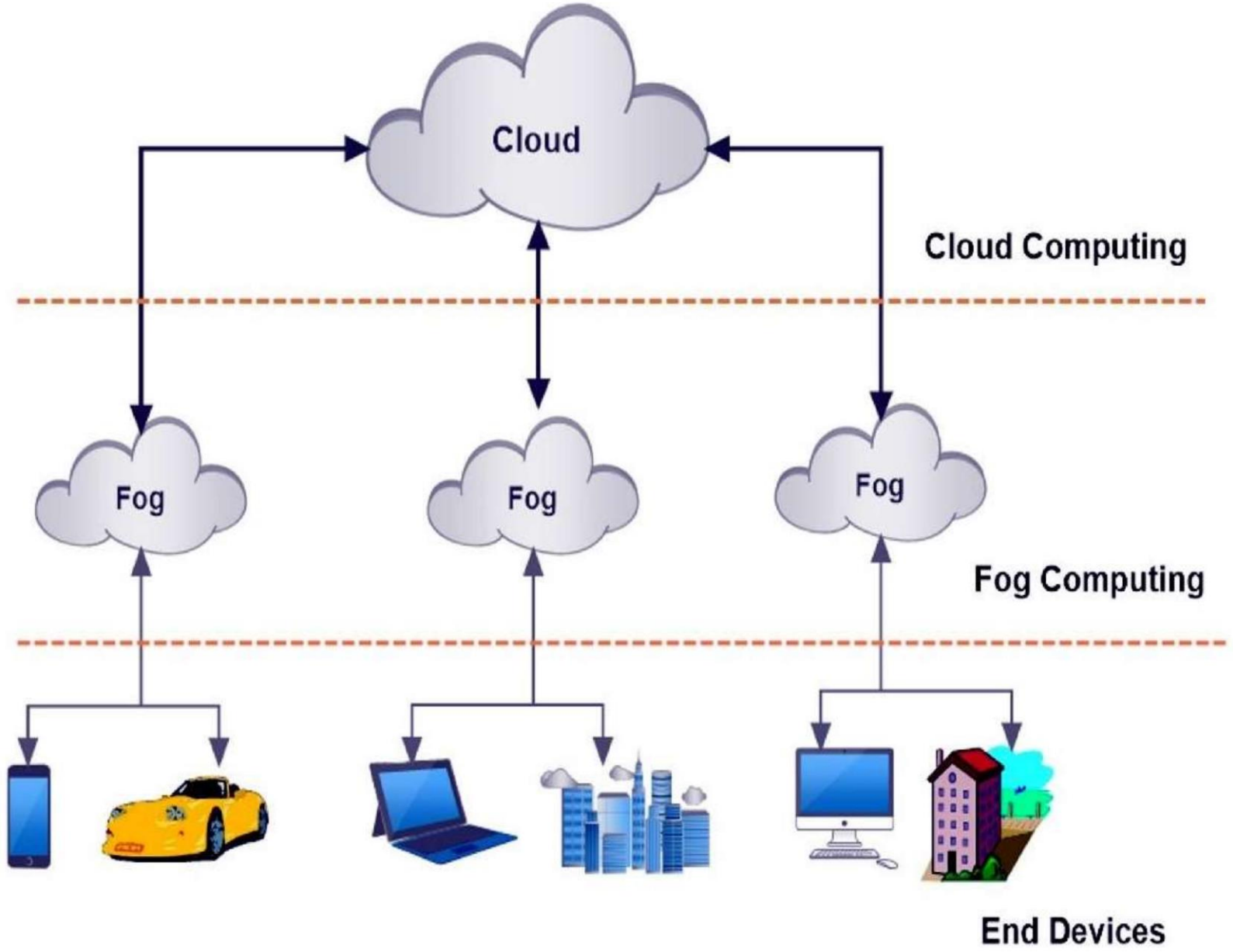
Several data-related problems need to be addressed:

- Bandwidth in last-mile IoT networks is very limited. When dealing with thousands/millions of devices, available bandwidth may be on order of tens of Kbps per device or even less.
- Latency can be very high. Instead of dealing with latency in the milliseconds range, large IoT networks often introduce latency of hundreds to thousands of milliseconds.

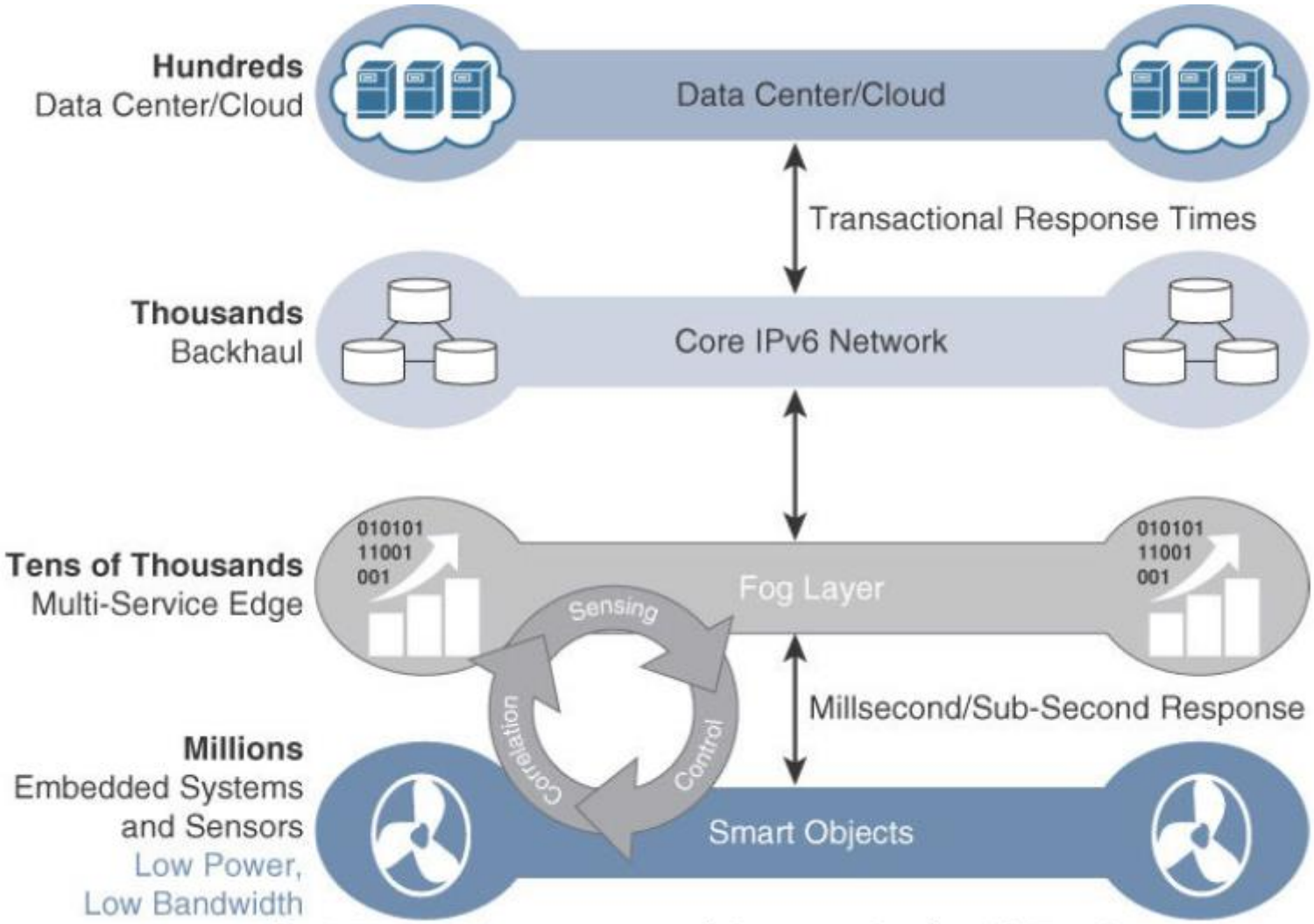
- Network backhaul from the gateway can be unreliable and often depends on 3G/LTE or even satellite links. Backhaul links can also be expensive if a per-byte data usage model is necessary.
- The volume of data transmitted over the backhaul can be high, and much of the data may not really be that interesting (such as simple polling messages).
- Big data is getting bigger. The concept of storing and analyzing all sensor data in the cloud is impractical. The sheer volume of data generated makes real-time analysis and response to the data almost impossible.

Fog Computing

The solution to the challenges mentioned in the previous section is to distribute data management throughout the IoT system, as close to the edge of the IP network as possible. The best-known embodiment of edge services in IoT is **fog computing**. Any device with computing, storage, and network connectivity can be a fog node. Examples include industrial controllers, switches, routers, embedded servers, and IoT gateways. Analyzing IoT data close to where it is collected minimizes latency, offloads gigabytes of network traffic from the core network, and keeps sensitive data inside the local network.


Cloud
Cloud Computing
Fog
Fog
Fog
Fog Computing
End Devices

An advantage of this structure is that the fog node allows intelligence gathering (such as analytics) and control from the closest possible point, and in doing so, it allows better performance over constrained networks. In one sense, this introduces a new layer to the traditional IT computing model, one that is often referred to as the "**fog layer**." The placement of the fog layer in the IoT Data Management and Compute Stack.



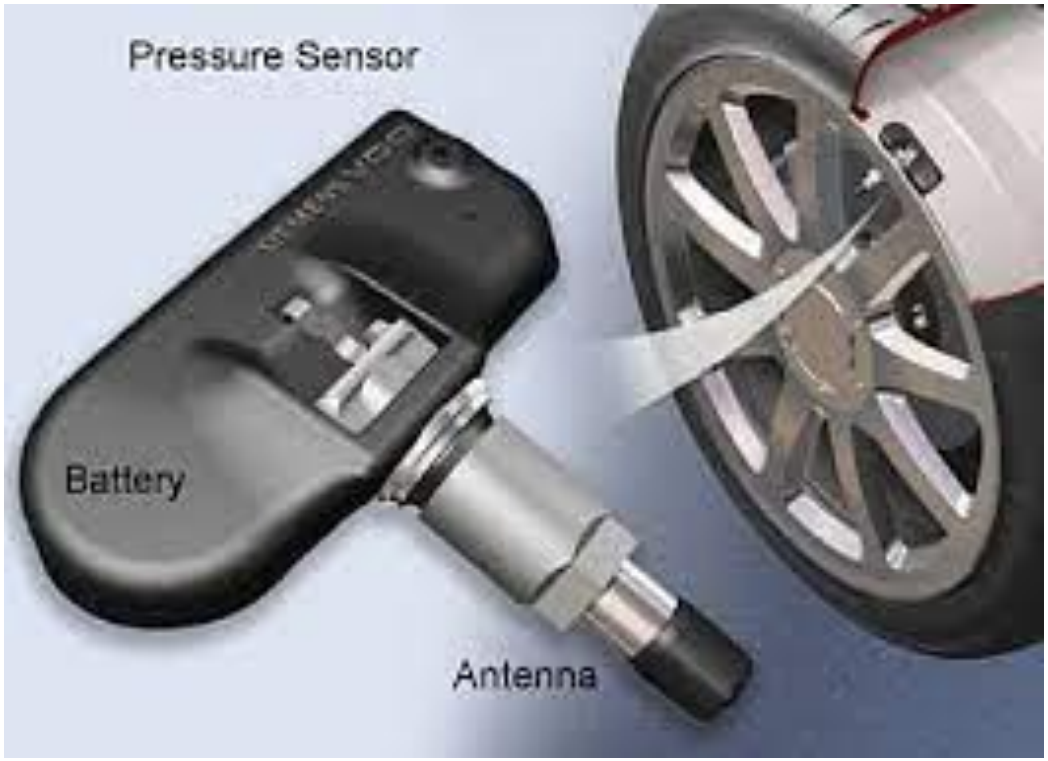
The IoT Data Management and Compute Stack with Fog Computing

Fog services are typically accomplished very close to the edge device, sitting as close to the IoT endpoints as possible. One significant advantage of this is that the fog node has contextual awareness of the sensors it is managing because of its geographic proximity to those sensors. For example, there might be a fog router on an oil derrick that is monitoring all the sensor activity at that location.

Because the fog node is able to analyze information from all the sensors on that derrick, it can provide contextual analysis of the messages it is receiving and may decide to send back only the relevant information over the backhaul network to the cloud. In this way, it is performing distributed analytics such that the volume of data sent upstream is greatly reduced and is much more useful to application and analytics servers residing in the cloud.

In addition, having contextual awareness gives fog nodes the ability to react to events in the IoT network much more quickly than in the traditional IT compute model, which would likely incur greater latency and have slower response times. The fog layer thus provides a distributed edge control loop capability, where devices can be monitored, controlled, and analyzed in real time without the need to wait for communication from the central analytics and application servers in the cloud.

The value of this model is clear. For example, tire pressure sensors on a large truck in an open-pit mine might continually report measurements all day long. There may be only minor pressure changes that are well within tolerance limits, making continual reporting to the cloud unnecessary. Is it really useful to continually send such data back to the cloud over a potentially expensive backhaul connection?


Pressure Sensor
Battery
Antenna

With a fog node on the truck, it is possible to not only measure the pressure of all tires at once but also combine this data with information coming from other sensors in the engine, hydraulics, and so on. With this approach, the fog node sends alert data upstream only if an actual problem is beginning to occur on the truck that affects operational efficiency.

IoT fog computing enables data to be preprocessed and correlated with other inputs to produce relevant information. This data can then be used as real-time, actionable knowledge by IoT-enabled applications. Longer term, this data can be used to gain a deeper understanding of network behavior and systems for the purpose of developing proactive policies, processes, and responses.

Fog applications are as diverse as the Internet of Things itself. What they have in common is data reduction—monitoring or analyzing real-time data from network-connected things and then initiating an action, such as locking a door, changing equipment settings, applying the brakes on a train, zooming a video camera, opening a valve in response to a pressure reading, creating a bar chart, or sending an alert to a technician to make a preventive repair.

The defining characteristic of fog computing are as follows:

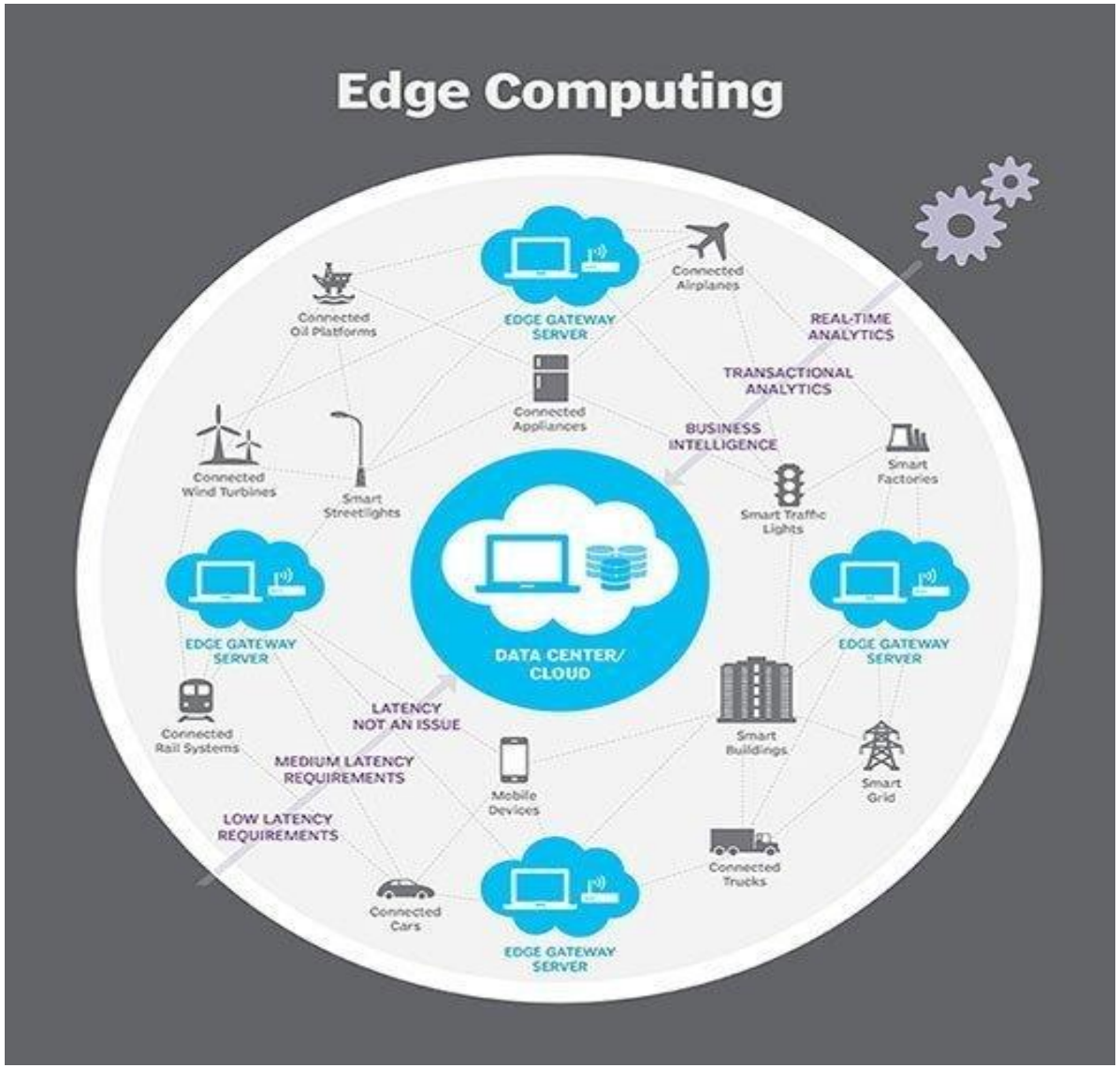
- Contextual location awareness and low latency: The fog node sits as close to the IoT endpoint as possible to deliver distributed computing.
- Geographic distribution: In sharp contrast to the more centralized cloud, the services and applications targeted by the fog nodes demand widely distributed deployments.

- Deployment near IoT endpoints: Fog nodes are typically deployed in the presence of a large number of IoT endpoints. For example, typical metering deployments often see 3000 to 4000 nodes per gateway router, which also functions as the fog computing node.
- Wireless communication between the fog and the IoT endpoint: Although it is possible to connect wired nodes, the advantages of fog are greatest when dealing with a large number of endpoints, and wireless access is the easiest way to achieve such scale.

- Use for real-time interactions: Important fog applications involve real-time interactions rather than batch processing. Preprocessing of data in the fog nodes allows upper-layer applications to perform batch processing on a subset of the data.

Edge Computing

Fog computing solutions are being adopted by many industries, and efforts to develop distributed applications and analytics tools are being introduced at an accelerating pace. The natural place for a fog node is in the network device that sits closest to the IoT endpoints, and these nodes are typically spread throughout an IoT network. However, in recent years, the concept of IoT computing has been pushed even further to the edge, and in some cases it now resides directly in the sensors and IoT devices.

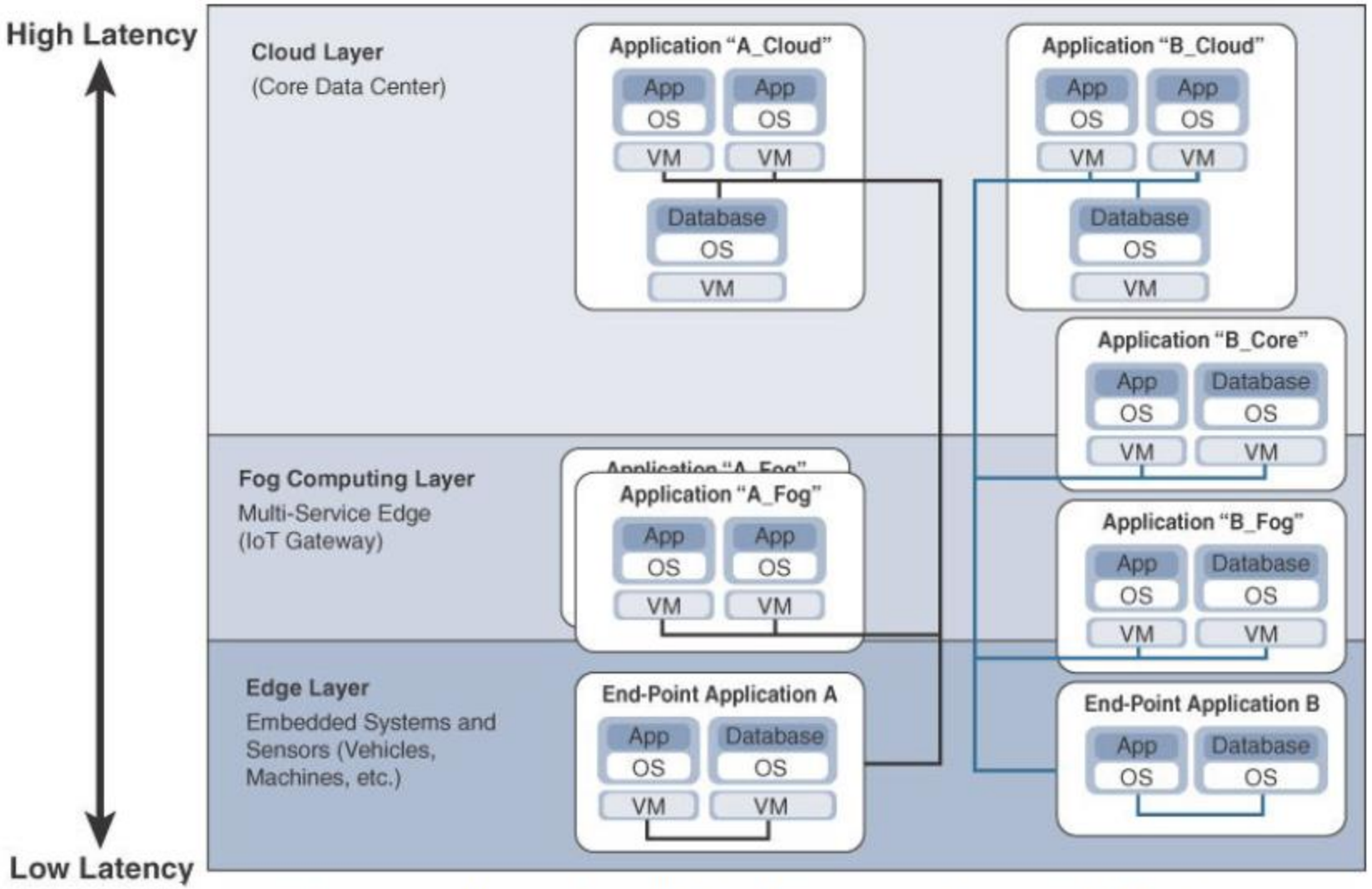

Edge Computing
Connected Oil Platforms
EDGE GATEWAY SERVER
Connected Airplanes
REAL-TIME ANALYTICS
TRANSACTIONAL ANALYTICS
Connected Appliances
BUSINESS INTELLIGENCE
Connected Wind Turbines
Smart Streetlights
Smart Factories
Smart Traffic Lights
EDGE GATEWAY SERVER
DATA CENTER/ CLOUD
EDGE GATEWAY SERVER
Connected Rail Systems
LATENCY NOT AN ISSUE
Smart Buildings
MEDIUM LATENCY REQUIREMENTS
Mobile Devices
Smart Grid
LOW LATENCY REQUIREMENTS
Connected Trucks
Connected Cars
EDGE GATEWAY SERVER

IoT devices and sensors often have constrained resources, however, as compute capabilities increase. Some new classes of IoT endpoints have enough compute capabilities to perform at least low-level analytics and filtering to make basic decisions. For example, consider a water sensor on a fire hydrant. While a fog node sitting on an electrical pole in the distribution network may have an excellent view of all the fire hydrants in a local neighborhood, a node on each hydrant would have clear view of a water pressure drop on its own line and would be able to quickly generate an alert of a localized problem.

The fog node, on the other hand, would have a wider view and would be able to ascertain whether the problem was more than just localized but was affecting the entire area. Another example is in the use of smart meters. Edge compute–capable meters are able to communicate with each other to share information on small subsets of the electrical distribution grid to monitor localized power quality and consumption, and they can inform a fog node of events that may pertain to only tiny sections of the grid. Models such as these help ensure the highest quality of power delivery to customers.

The Hierarchy of Edge, Fog, and Cloud: It is important to stress that edge or fog computing in no way replaces the cloud. Rather, they complement each other, and many use cases actually require strong cooperation between layers. In the same way that lower courts do not replace the supreme court of a country, edge and fog computing layers simply act as a first line of defense for filtering, analyzing, and otherwise managing data endpoints. This saves the cloud from being queried by each and every node for each event.

This model suggests a hierarchical organization of network, compute, and data storage resources. At each stage, data is collected, analyzed, and responded to when necessary, according to the capabilities of the resources at each layer. As data needs to be sent to the cloud, the latency becomes higher. The advantage of this hierarchy is that a response to events from resources close to the end device is fast and can result in immediate benefits, while still having deeper compute resources available in the cloud when necessary.



Distributed Compute and Data Management Across an IoT System

It is important to note that the heterogeneity of IoT devices also means a heterogeneity of edge and fog computing resources. While **cloud resources are expected to be homogenous,** it is fair to expect that in many cases both edge and fog resources will use different operating systems, have different CPU and data storage capabilities, and have different energy consumption profiles. Edge and fog thus require an abstraction layer that allows applications to communicate with one another.

The abstraction layer exposes a common set of APIs for monitoring, provisioning, and controlling the physical resources in a standardized way. The abstraction layer also requires a mechanism to support virtualization, with the ability to run multiple operating systems or service containers on physical devices to support multitenancy and application consistency across the IoT system. Definition of a common communications services framework is being addressed by groups such as oneM2M, discussed earlier.

From an architectural standpoint, fog nodes closest to the network edge receive the data from IoT devices. **The fog IoT application then directs different types of data to the optimal place for analysis**:

- The most time-sensitive data is analyzed on the edge or fog node closest to the things generating the data.
- Data that can wait seconds or minutes for action is passed along to an aggregation node for analysis and action.
- Data that is less time sensitive is sent to the cloud for historical analysis, big data analytics, and long-term storage.

For example, each of thousands or hundreds of thousands of fog nodes might send periodic summaries of data to the cloud for historical analysis and storage. In summary, **when architecting an IoT network,** you should consider the amount of data to be analyzed and the time sensitivity of this data. Understanding these factors will help you decide whether cloud computing is enough or whether edge or fog computing would improve your system efficiency. Fog computing accelerates awareness and response to events by eliminating a round trip to the cloud for analysis. It avoids the need for costly bandwidth additions by offloading gigabytes of network traffic from the core network. It also protects sensitive IoT data by analyzing it inside company walls.