Introduction to internet of things

Internet Of Things (University of Mumbai)

THE FLAVOUR OF THE INTERNET OF THINGS:

• (Examples of IOT)

• 1) The alarm rings. As you open your eyes , you see that it's five minutes later than your usual wake-up time.

• The clock has checked the train times online, and your train must be delayed, so it lets you sleep in a little longer.

• 2) In your kitchen, a blinking light reminds you it's time to take your tablets.

• 3) If you forget, the medicine bottle cap goes online and emails your doctor to let her know.

• 4) On your way out of the house, you catch a glow in the corner of your eye.

• Your umbrella handle is lit up, which means that it has checked the BBC weather reports and predicts rain.

• 5) As you pass the bus stop on the way to the station, you notice the large LCD display flash that the number 23 is due.

• When the bus company first installed those displays, they ran on the expected timetable information only, but now that every bus has GPS tracking its location, they simply connect to the bus company's online service and always give the updated information.

• 6) An ornament with a dial notices the change and starts to turn so that the text on it points to the word "Travelling".

• Your family will also see later that you've arrived at "Work" safely.

• 7) The wrist band's large display also makes it easy to glance down and see how fast you are running and how many calories you've burned.

• All the data is automatically uploaded to your sports tracking site, which also integrates with your online supermarket shopping account to make it easy to compare with how many calories you've eaten.

THE "INTERNET" OF "THINGS":

• All the cases we saw used the *Internet* to send, receive, or communicate information.

• And in each case, the gadget that was connected to the Internet wasn't a computer, tablet, or mobile phone but an object, a *Thing*.

• **These Things are designed for a purpose:**

• the umbrella has a retractable canopy and a handle to hold it.

• A bus display has to be readable to public transport users, including the elderly and partially sighted and be able to survive poor weather conditions.

• The sports bracelet is easy to wear while running, has a display that is large enough and bright enough to read even when you are moving, and will survive heat, cold, sweat, and rain.

• Unlike a calm light in the umbrella stand, gives piece of information to process subconsciously when you pass it on the way out of your home, an app requires you to perform several actions. (you have to take the phone out of your pocket

or bag, unlock it, navigate to the right website , you have to type the URL and read the data from a small screen.)

• Rather than having greater capabilities, the smart umbrella simply moves the same intelligence into your environment so that you don't have to change your routine.

• So the idea of the Internet of Things suggests that rather than having a small number of very powerful computing devices in your life (laptop, tablet, phone)

• you might have a large number of devices which are perhaps less powerful (umbrella, bracelet, mirror, fridge, shoes).

• The definition of ubicomp, however, would also include the air fresheners which release scent when they detect movement in the room as part of its domain.

• That is to say, such a device is an intelligently programmed computer processor, driven by sensors in the real world, and driving output in the real world, all embedded into an everyday object.

• These factors make this ubicomp, and it is only differentiated from the "Internet of Things" by the fact that these days most of the really interesting things done with computing also involve an Internet connection.

• But what does it mean to "connect an object to the Internet"?

• Clearly, sticking an Ethernet socket into a chair or a 3G modem into a sewing machine doesn't suddenly inspire the object with mysterious properties.

• Rather, there has to be some flow of information which connects the defining characteristics of the Thing with the world of data and processing represented by the Internet.

• The Thing is present, physically in the real world, in your home, your work, your car, or worn around your body.

• This means that it can receive inputs from your world and transform those into data which is sent onto the Internet for collection and processing.

• So your chair might collect information about how often you sit on it and for how long.

• The presence of the Thing also means that it can produce outputs into your world with what we call "actuators".

• Some of these outputs could be triggered by data that has been collected and processed on the Internet.

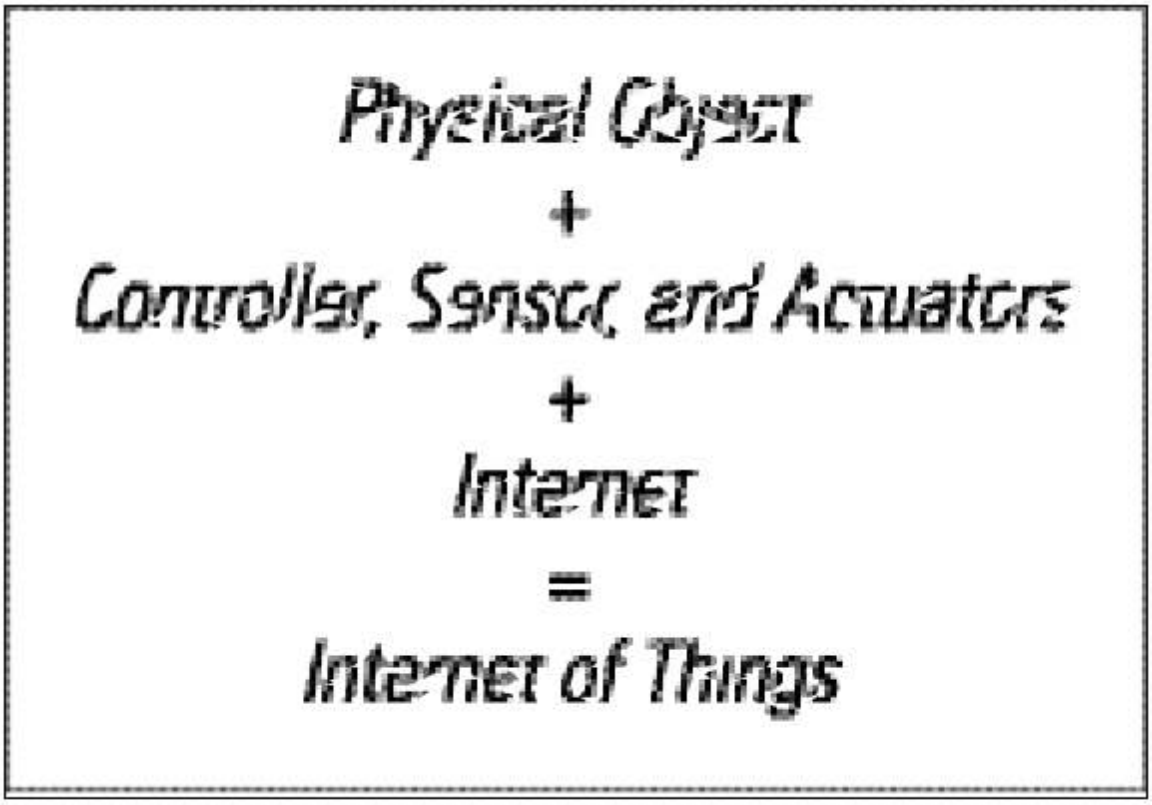
• So your chair might vibrate to tell you that you have received email.

• We could summarize these components in the following simple equation:

• Note that in all the cases we've looked at, the form of the object follows the function of the Thing:

• your chair is designed to sit on, the sewing machine to sew at, and so on.

• The fact of also being connected to the Internet and having general-purpose computing capabilities doesn't necessarily have an impact on the form of the object at all.



An equation for the Internet of Things.

THE TECHNOLOGY OF THE INTERNET OF THINGS:

• It is worth taking a little time to look at the Internet of Things through a lens of the history of technology to more clearly understand how and where it fits.

• Technology's great drivers have initially been fundamental needs, such as food and water, warmth, safety, and health.

• Hunting and foraging, building and medicine grow out of these needs.

• Then, because resources for these things are not always distributed where and when one might like, technological advances progress with enabling and controlling the movement of people.

• Trade develops as a movement of goods from a place where they are plentiful and cheap to one where they are rare and valuable.

• **Storage** is a form of movement in time—for example, from harvest time, when food is plentiful and cheap, to the following winter, when it is highly valued.

• Information becomes key, too—hence, the development of **language to communicate** technology to others.

• Travellers might pass on messages as well as goods and services, and an oral tradition allows this information to pass through time as well as space.

• From writing, via the telegraph, radio, and television, to digital information, more and more technology has been about enabling the movement of information or doing interesting things with that information.

• As technology has progressed, new categories of objects have been created:

• in the electronic age, they have included telephones, radios, televisions, computers, and smartphones.

• As with most new technology, these devices tended to start out very expensive and gradually come down in price.

• Demand drives down prices, and research leads to optimization and miniaturisation.

• Ultimately, it becomes not just possible but also feasible to include functionality that would previously have required its own dedicated device *inside* another one.

• mere computing power isn’t a sufficient precondition for the Internet of Things.

• Rather, we are looking at computing power linked on the one hand to electronic sensors and actuators which interact with the real world and on the other to the Internet.

• It turns out that the rapid sharing and processing of *information* with services or other consumers is a huge differentiator.

• Internet connectivity is also cheaper and more convenient than it used to be.

• Whereas in the past, we were tied to expensive and slow dial-up connections, nowadays we have broadband subscriptions, providing always-on connectivity to the Net.

• Wired Ethernet provides a fairly plug-and-play networking experience, but most home routers today also offer WiFi, which removes the need for running cables everywhere.

• For situations in which a fixed network connection isn’t readily available, mobile phone connectivity is widespread.

• Another factor at play is the maturity of online platforms.

• Whereas early web apps were designed to be used only from a web browser, programming using an Application Programming Interface (API), which allows other programs, rather than just users, to interact with and use the services on offer.

• As the online services mature, so too do the tools used to build and scale them.

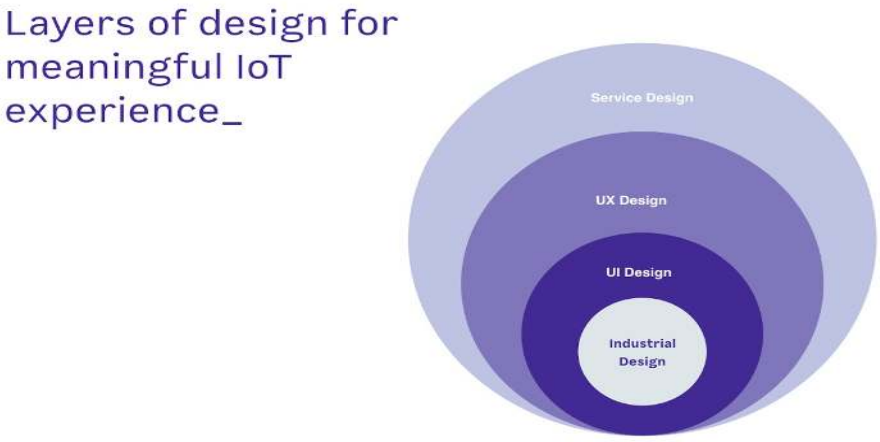
• Web services frameworks such as Python or Ruby on allow easy prototyping of the online component.

• Similarly, cloud services such as Amazon Web Services mean that such solutions can scale easily with use as they become more popular.

7 design principles for IoT:

In the near future, our everyday lives will be more and more filled with intelligent, connected objects. They will appear in our homes, in our working environments and in the cities we live in as well as travel with us everywhere we go in the form of wearables, smart clothing and things we cannot even imagine right now. This development is called the internet of things, IoT.

For designers focused on designing SW services and screen based interfaces or physical products, designing IoT solutions creates totally new design challenges. IoT solutions consist of multiple elements: physical devices like sensors, actuators and interactive devices, the network connecting these devices, the data gathered from these devices and analyzed to create a meaningful experience and last but definitely not least, the physical context in which user interacts with the solution. You need to do various types of design, from industrial product design to service and business design. All of these factors have their impact to the total UX of the IoT system and the task of designing in this context may feel quite overwhelming. To make it a little easier, here is the list of the 7 most important design principles for IoT.



1. Focus on value

In the world of IoT, user research and service design are more crucial than ever. While early adopters are eager to try out new technology, many others are reluctant to take new technology into use and cautious about using it, due to not feeling confident with it. For your IoT solution to become widely adopted, you need to dig deep into users' needs in order to find out where lies a problem truly worth solving and what is the real end user value of the solution. You also need to understand what might be the barriers of adopting the new technology in general and your solution specifically. For deciding on your feature set, you need research too. The features that might be valuable and highly relevant for the tech early adopters may be uninteresting for the majority of the users and vice versa, so you need to plan carefully what features to include and in which order.

2. Take a holistic view

IoT solutions typically consist of multiple devices with different capabilities and both physical and digital touchpoints. The solution may also be provided in co-operation with multiple different service providers. It is not enough to design one of the touchpoints well, instead you need to take a holistic look across the whole system, the role of each device and service, and the conceptual model of how user understands and perceives the system. The whole system needs to work seamlessly together in order to create a meaningful experience.

3. Put safety first

As the IoT solutions are placed in the real world context, the consequences can be serious, when something goes wrong. At the same time the users of the IoT solutions may be vary of using new technology, so building trust should be one of your main design drivers. Trust is built slowly and lost easily, so you really need to make sure that every interaction with the product/service builds the trust rather than breaks it. What it means in practise? First of all, it means understanding possible error situations related to context of use, HW, SW and network as well as to user interactions and trying to prevent them. Secondly, if the error situations still occur, it means appropriately informing the user about them and helping them to recover. Secondly, it means considering data security & privacy as key elements of your design. It is really important for users to feel, that their private data is safe, their home, working environment and everyday objects cannot be hacked and their loved ones are not put at risk. Thirdly, quality assurance is critical and it should not only focus on testing the SW, but on testing the end to end system, in a real-world context.

4. Consider the context

IoT solutions exist at the crossroads of the physical and digital worlds. Commands given through digital interfaces may produce real world effects, but unlike digital commands, the actions happening in the real-world cannot necessarily be undone. In the real world context lots of unexpected things can happen and at the same time user should be able to feel safe and in control. The context places also other kind of requirements to the design. Depending on the physical context, the goal might be to minimize distraction of the user or e.g. to design devices that hold up against changing weather conditions. IoT solutions in homes, workplaces and public areas are are typically multi-user systems and thus less personal than e.g. screen based solutions used in smartphones, which also brings into picture the social context where the solution is used and its' requirements for the design.

5. Build a strong brand

Due to the real world context of the IoT solutions, regardless of how carefully you design things and aim to build trust, something unexpected will happen at some point and your solution is somehow going to fail. In this kind of situations, it is of utmost importance, that you have built a strong brand that truly resonates with the end users. When they feel connected to your brand, they will be more forgiving about the system failures and will still keep on using your solution. While designing your brand, you must keep in mind, that trust should be a key element of the brand, one of the core brand values. This core value should also be reflected in the rest of the brand elements, like the choice of color, tone of voice, imagery etc.

6. Prototype early and often

Typically HW and SW have quite different lifespans, but as successful IoT solution needs both the HW and SW elements, the lifespans should be aligned. At the same time, IoT solutions are hard to upgrade, because once the connected object is placed somewhere, it is not so easy to replace it with a newer version, especially if the user would need to pay for the upgrade and even the software within the connected object may be hard to update due to security and privacy reasons. Due to these factors and to avoid costly hardware iterations, it's crucial to get the solution right, from the beginning of implementation. What this means from the design perspective is that prototyping and rapid iteration of both the HW and the whole solution are essential in the early stages of the project. New, more creative ways of prototyping and faking the solution are needed.

7. Use data responsibly

IoT solutions can easily generate tons of data. However, the idea is not to hoard as much data as possible, but instead to identify the data points that are needed to make the solution functional and useful. Still, the amount of data may be vast, so it's necessary for the designer to understand the possibilities of data science and how to make sense of the data. Data science provides a lot of opportunities to reduce user friction, i.e. reducing use of time, energy and attention or diminishing stress. It can be used to automate repeated context dependent decisions, to interpret intent from incomplete/inadequate input or to filter meaningful signals from noise. Understanding what data is available and how it can be used to help the user is a key element in designing successful IoT services.

The oneM2M IoT Standardized Architecture:

In an effort to standardize the rapidly growing field of machine-to-machine (M2M) communications, the European Telecommunications Standards Institute (ETSI) created the M2M Technical Committee in 2008. The goal of this committee was to create a common architecture that would help accelerate the adoption of M2M applications and devices. Over time, the scope has expanded to include the Internet of Things.

Other related bodies also began to create similar M2M architectures, and a common standard for M2M became necessary. Recognizing this need, in 2012 ETSI and 13 other founding members launched oneM2M as a global initiative designed to promote efficient M2M communication systems and IoT. The goal of oneM2M is to create a common services layer, which can be readily embedded in field devices to allow communication with application servers.oneM2M's framework focuses on IoT services, applications, and platforms. These include smart metering applications, smart grid, smart city automation, e-health, and connected vehicles.

One of the greatest challenges in designing an IoT architecture is dealing with the heterogeneity of devices, software, and access methods. By developing a horizontal platform architecture, oneM2M is developing standards that allow interoperability at all levels of the IoT stack. For example, you might want to automate your HVAC system by connecting it with wireless temperature sensors spread throughout your office. You decide to deploy sensors that use LoRaWAN technology. The problem is that the LoRaWAN network and the BACnet system that your HVAC and BMS run on are completely different systems and have no natural connection point. This is where the oneM2M common services architecture comes in. oneM2M's horizontal framework and RESTful APIs allow the LoRaWAN system to interface with the building management system over an IoT network, thus promoting end-to end IoT communications in a consistent way, no matter how heterogeneous the networks.

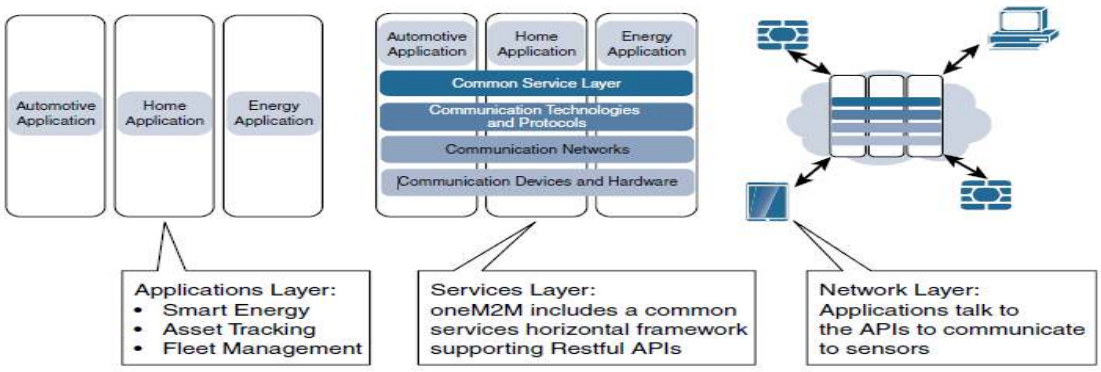


Figure 2-1 *The Main Elements of the oneM2M IoT Architecture*

The oneM2M architecture divides IoT functions into three major domains: the application layer, the services layer, and the network layer. While this architecture may seem simple and somewhat generic at first glance, it is very rich and promotes interoperability through IT-friendly APIs and supports a wide range of IoT technologies. Let's examine each of these domains in turn:

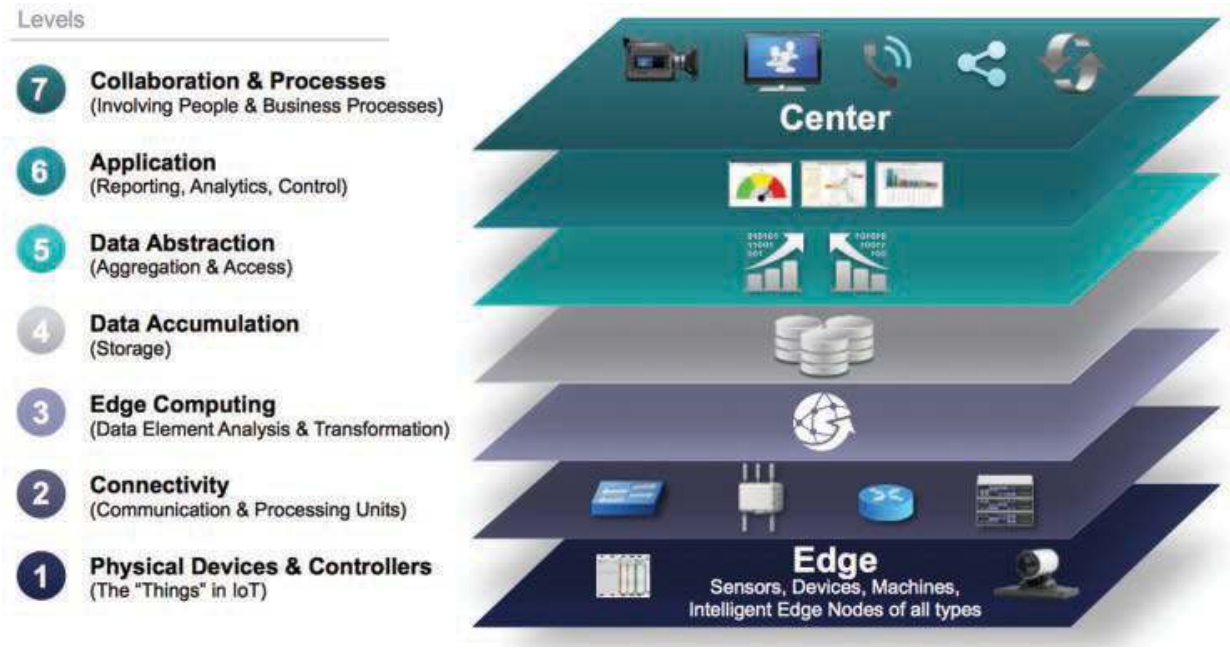
- **Applications layer:** The oneM2M architecture gives major attention to connectivity between devices and their applications. This domain includes the application-layer protocols and attempts to standardize northbound API definitions for interaction with business intelligence (BI) systems. Applications tend to be industry-specific and have their own sets of data models, and thus they are shown as vertical entities.

■ **Services layer:** This layer is shown as a horizontal framework across the vertical industry applications. At this layer, horizontal modules include the physical network that the IoT applications run on, the underlying management protocols, and the hardware. Examples include backhaul communications via cellular, MPLS networks, VPNs, and so on. Riding on top is the common services layer. This conceptual layer adds APIs and middleware supporting third-party services and applications. One of the stated goals of oneM2M is to "develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software nodes, and rely upon connecting the myriad of devices in the field area network to M2M application servers, which typically reside in a cloud or data center." A critical objective of oneM2M is to attract and actively involve organizations from M2M-related business domains, including telematics and intelligent transportation, healthcare, utility, industrial automation, and smart home applications, to name just a few.

■ **Network layer:** This is the communication domain for the IoT devices and endpoints. It includes the devices themselves and the communications network that links them. Embodiments of this communications infrastructure include wireless mesh technologies, such as IEEE 802.15.4, and wireless point-to-multipoint systems, such as IEEE 801.11ah. Also included are wired device connections, such as IEEE 1901 power line communications. Chapter 4 provides more details on these connectivity technologies.

The IoT World Forum (IoTWF) Standardized Architecture

In 2014 the IoTWF architectural committee (led by Cisco, IBM, Rockwell Automation, and others) published a seven-layer IoT architectural reference model. While various IoT reference models exist, the one put forth by the IoT World Forum offers a clean, simplified perspective on IoT and includes edge computing, data storage, and access. It provides a succinct way of visualizing IoT from a technical perspective. Each of the seven layers is broken down into specific functions, and security encompasses the entire model. Figure 2-2 details the IoT Reference Model published by the IoTWF.



As shown in Figure 2-2, the IoT Reference Model defines a set of levels with control flowing from the center (this could be either a cloud service or a dedicated data center), to the edge, which includes sensors, devices, machines, and other types of intelligent end nodes. In general, data travels up the stack, originating from the edge, and goes northbound to the center. Using this reference model, we are able to achieve the following:

- Decompose the IoT problem into smaller parts
- Identify different technologies at each layer and how they relate to one another
- Define a system in which different parts can be provided by different vendors
- Have a process of defining interfaces that leads to interoperability
- Define a tiered security model that is enforced at the transition points between levels

The following sections look more closely at each of the seven layers of the IoT Reference Model

Layer 1: Physical Devices and Controllers Layer

The first layer of the IoT Reference Model is the physical devices and controllers layer. This layer is home to the “things” in the Internet of Things, including the various

endpoint devices and sensors that send and receive information. The size of these "things" can range from almost microscopic sensors to giant machines in a factory. Their primary function is generating data and being capable of being queried and/or controlled over a network.

Layer 2: Connectivity Layer

In the second layer of the IoT Reference Model, the focus is on connectivity. The most important function of this IoT layer is the reliable and timely transmission of data. More specifically, this includes transmissions between Layer 1 devices and the network and between the network and information processing that occurs at Layer 3 (the edge computing layer).

As you may notice, the connectivity layer encompasses all networking elements of IoT and doesn't really distinguish between the last-mile network (the network between the sensor/endpoint and the IoT gateway, discussed later in this chapter), gateway, and backhaul networks. Functions of the connectivity layer are detailed in Figure 2-3.

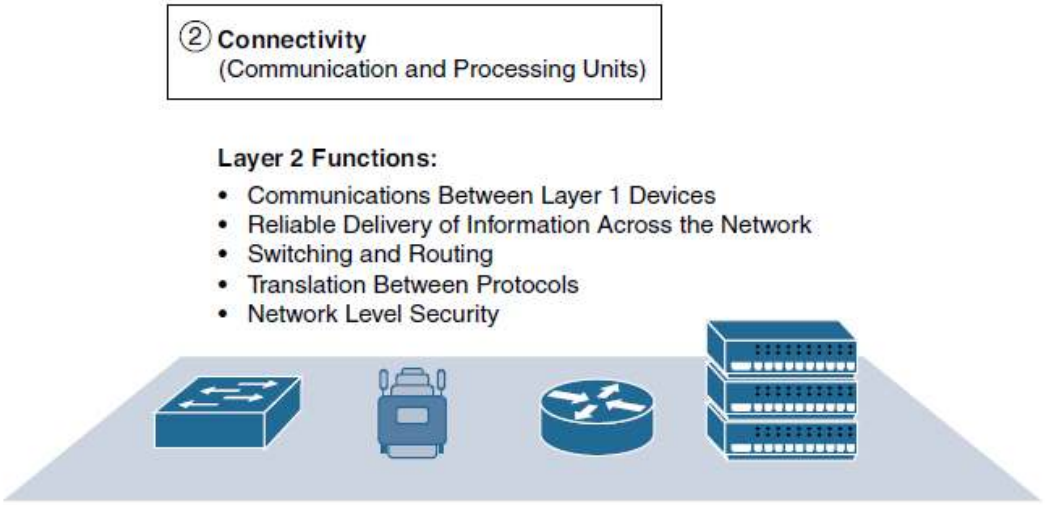


Figure 2-3 *IoT Reference Model Connectivity Layer Functions*

Layer 3: Edge Computing Layer

Edge computing is the role of Layer 3. Edge computing is often referred to as the "fog" layer and is discussed in the section "Fog Computing," later in this chapter. At this layer, the emphasis is on data reduction and converting network data flows into information that is ready for storage and processing by higher layers. One of the basic principles of this reference model is that information processing is initiated as early and as close to the edge of the network as possible. Figure 2-4 highlights the functions handled by Layer 3 of the IoT Reference Model.

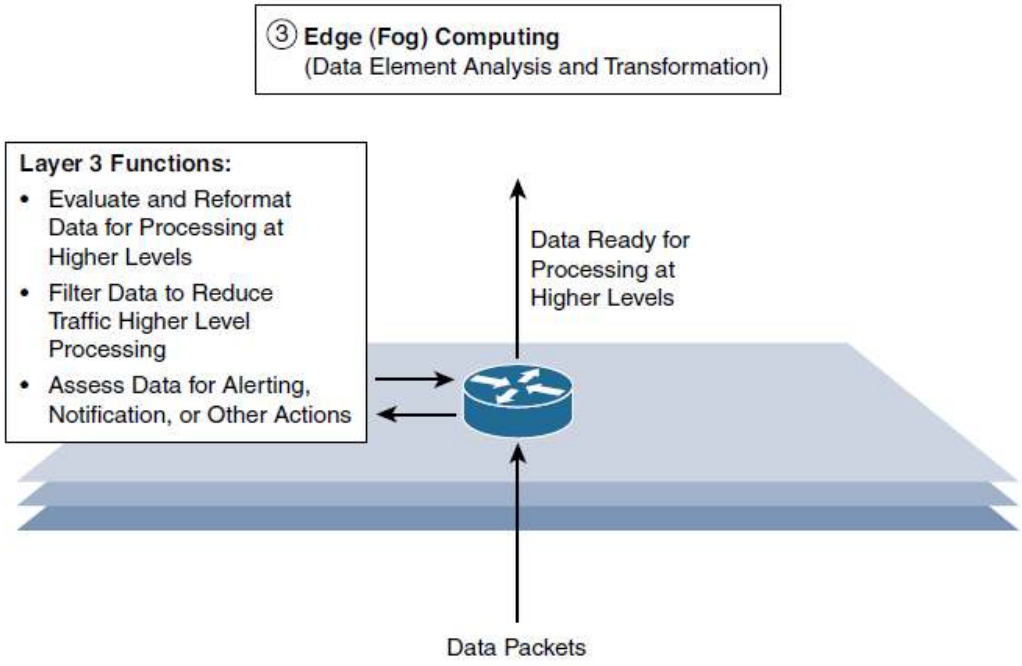


Figure 2-4 *IoT Reference Model Layer 3 Functions*

Another important function that occurs at Layer 3 is the evaluation of data to see if it can be filtered or aggregated before being sent to a higher layer. This also allows for data to be reformatted or decoded, making additional processing by other systems easier. Thus, a critical function is assessing the data to see if predefined thresholds are crossed and any action or alerts need to be sent.

Upper Layers: Layers 4–7

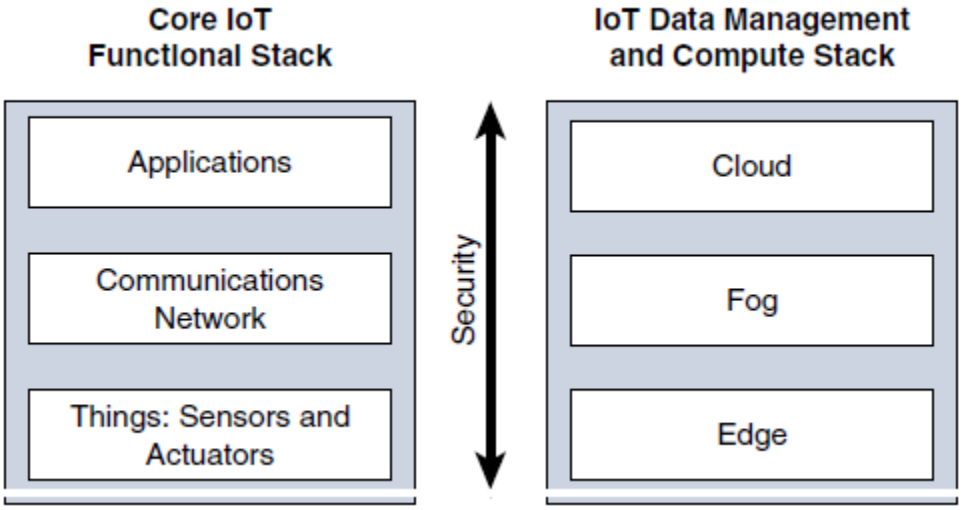
The upper layers deal with handling and processing the IoT data generated by the bottom layer. For the sake of completeness, Layers 4–7 of the IoT Reference Model are summarized in Table 2-2.

Table 2-2 *Summary of Layers 4–7 of the IoTWF Reference Model*

IoT Reference Model Layer	Functions
Layer 4: Data accumulation layer	Captures data and stores it so it is usable by applications when necessary. Converts event-based data to query-based processing.
Layer 5: Data abstraction layer	Reconciles multiple data formats and ensures consistent semantics from various sources. Confirms that the data set is complete and consolidates data into one place or multiple data stores using virtualization.
Layer 6: Applications layer	Interprets data using software applications. Applications may monitor, control, and provide reports based on the analysis of the data.
Layer 7: Collaboration and processes layer	Consumes and shares the application information. Collaborating on and communicating IoT information often requires multiple steps, and it is what makes IoT useful. This layer can change business processes and delivers the benefits of IoT.

IoT Architecture:

IoT architectures may differ somewhat depending on the industry use case or technology being deployed, and each has merit in solving the IoT heterogeneity problem. Figure illustrates the simplified IoT model. This framework is presented as two parallel stacks: The IoT Data Management and Compute Stack and the Core IoT Functional Stack.



The Core IoT Functional Stack

IoT networks are built around the concept of "things," or smart objects performing functions and delivering new connected services. These objects are "smart" because they use a combination of contextual information and configured goals to perform actions. These actions can be self-contained (that is, the smart object does not rely on external systems for its actions); however, in most cases, the "thing" interacts with an external system to report information that the smart object collects, to exchange with other objects, or to interact with a management platform. In this case, the management platform can be used to process data collected from the smart object and also guide the behavior of the smart object. From an architectural standpoint, several components have to work together for an IoT network to be operational:

- **"Things" layer:** At this layer, the physical devices need to fit the constraints of the environment in which they are deployed while still being able to provide the information needed.

- **Communications network layer:** When smart objects are not self-contained, they need to communicate with an external system. In many cases, this communication uses a wireless technology. This layer has four sublayers:

- **Access network sublayer:** The last mile of the IoT network is the access network. This is typically made up of wireless technologies such as 802.11ah, 802.15.4g, and LoRa. The sensors connected to the access network may also be wired.

- **Gateways and backhaul network sublayer:** A common communication system organizes multiple smart objects in a given area around a common gateway. The gateway communicates directly with the smart objects. The role of the gateway is to forward the collected information through a longer-range medium (called the backhaul) to a headend central station where the information is processed. This information exchange is a Layer 7 (application) function, which is the reason this object is called a gateway. On IP networks, this gateway also forwards packets from one IP network to another, and it therefore acts as a router.

- **Network transport sublayer:** For communication to be successful, network and transport layer protocols such as IP and UDP must be implemented to support the variety of devices to connect and media to use.

- **IoT network management sublayer:** Additional protocols must be in place to allow the headend applications to exchange data with the sensors. Examples include CoAP and MQTT.

- **Application and analytics layer:** At the upper layer, an application needs to process the collected data, not only to control the smart objects when necessary, but to make intelligent decision based on the information collected and, in turn, instruct the "things" or other systems to adapt to the analyzed conditions and change their behaviors or parameters.

Layer 1: Things: Sensors and Actuators Layer

Most IoT networks start from the object, or "thing," that needs to be connected. From an architectural standpoint, the variety of smart object types,
shapes, and needs drive the variety of IoT protocols and architectures. There are myriad ways to classify smart objects. One architectural classification could be:

■ **Battery-powered or power-connected:** This classification is based on whether the object carries its own energy supply or receives continuous power from an external power source. Battery-powered things can be moved more easily than line-powered objects. However, batteries limit the lifetime and amount of energy that the object is allowed to consume, thus driving transmission range and frequency.

■ **Mobile or static:** This classification is based on whether the "thing" should move or always stay at the same location. A sensor may be mobile because it is moved from one object to another (for example, a viscosity sensor moved from batch to batch in a chemical plant) or because it is attached to a moving object (for example, a location sensor on moving goods in a warehouse or factory floor). The frequency of the movement may also vary, from occasional to permanent. The range of mobility (from a few inches to miles away) often drives the possible power source.

■ **Low or high reporting frequency:** This classification is based on how often the object should report monitored parameters. A rust sensor may report values once a month. A motion sensor may report acceleration several hundred times per second. Higher frequencies drive higher energy consumption, which may create constraints on the possible power source (and therefore the object mobility) and the transmission range.

■ **Simple or rich data:** This classification is based on the quantity of data exchanged at each report cycle. A humidity sensor in a field may report a simple daily index value (on a binary scale from 0 to 255), while an engine sensor may report hundreds of parameters, from temperature to pressure, gas velocity, compression speed, carbon index, and many others. Richer data typically drives higher power consumption. This classification is often combined with the previous to determine the object data throughput (low throughput to high throughput). You may want to keep in mind that throughput is a combined metric. A medium-throughput object may send simple data at rather high frequency (in which case the flow structure looks continuous), or may send rich data at rather low frequency (in which case the flow structure looks bursty).

■ **Report range:** This classification is based on the distance at which the gateway is located. For example, for your fitness band to communicate with your phone, it needs to be located a few meters away at most. The assumption is that your phone needs to be at visual distance for you to consult the reported data on the phone screen. If the phone is far away, you typically do not use it, and reporting data from the band to the phone is not necessary. By contrast, a moisture sensor in the asphalt of a road may need to communicate with its reader several hundred meters or even kilometers away.

■ **Object density per cell:** This classification is based on the number of smart objects (with a similar need to communicate) over a given area, connected to the same gateway. An oil pipeline may utilize a single sensor at key locations every few miles. By contrast, telescopes like the SETI Colossus telescope at the Whipple Observatory deploy hundreds, and sometimes thousands, of mirrors over a small area, each with multiple gyroscopes, gravity, and vibration sensors.

Layer 2: Communications Network Layer: Once you have determined the influence of the smart object form factor over its transmission capabilities (transmission range, data volume and frequency, sensor density and mobility), you are ready to connect the object and communicate. Compute and network assets used in IoT can be very different from those in IT environments. The difference in the physical form factors between devices used by IT and OT is obvious even to the most casual of observers. What typically drives this is the physical environment in which the devices are deployed. What may not be as inherently obvious, however, is their operational differences. The operational differences must be understood in order to apply the correct handling to secure the target assets. Temperature variances are an easily understood metric. The cause for the variance is easily attributed to external weather forces and internal operating conditions. Remote external locations, such as those associated with mineral extraction or pipeline equipment can span from the heat of the Arabian Gulf to the cold of the Alaskan North Slope. Controls
near the furnaces of a steel mill obviously require heat tolerance, and controls for cold food storage require the opposite. In some cases, these controls must handle extreme fluctuations as well. These extremes can be seen within a single deployment. For example, portions of the Tehachapi, California, wind farms are located in the Mojave Desert, while others are at an altitude of 1800 m in the surrounding mountains. As you can imagine, the wide variance in temperature takes a special piece of hardware that is capable of withstanding such harsh environments.

Layer 3: Applications and Analytics Layer
Once connected to a network, your smart objects exchange information with other systems. As soon as your IoT network spans more than a few sensors, the power of the Internet of Things appears in the applications

- Analytics Versus Control Applications:

Multiple applications can help increase the efficiency of an IoT network. Each application collects data and provides a range of functions based on analyzing the collected data. From an architectural standpoint, one basic classification can be as follows:

■ **Analytics application:** This type of application collects data from multiple smart objects, processes the collected data, and displays information resulting from the data that was processed. The display can be about any aspect of the IoT network, from historical reports, statistics, or trends to individual system states. The important aspect is that the application processes the data to convey a view of the network that cannot be obtained from solely looking at the information displayed by a single smart object.

■ **Control application:** This type of application controls the behavior of the smart object or the behavior of an object related to the smart object. For example, a pressure

sensor may be connected to a pump. A control application increases the pump speed when the connected sensor detects a drop in pressure. Control applications are very useful for controlling complex aspects of an IoT network with a logic that cannot be programmed inside a single IoT object, either because the configured changes are too complex to fit into the local system or because the configured changes rely on parameters that include elements outside the IoT object.
An example of control system architecture is SCADA.

IoT Data Management and Compute Stack:

As illustrated in Figure, 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.

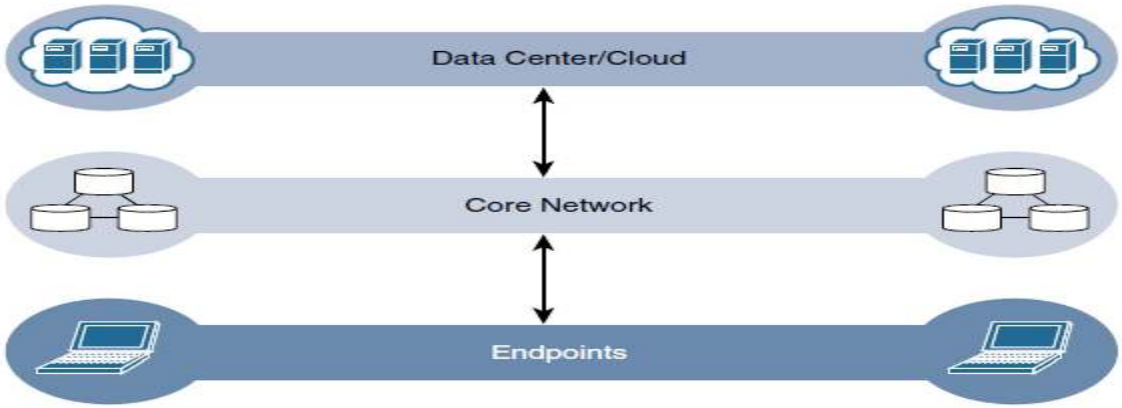


Figure 2-14 *The Traditional IT Cloud Computing Model*

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 above 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.

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." Figure 2-15 shows the placement of the fog layer in the IoT Data Management and Compute Stack.

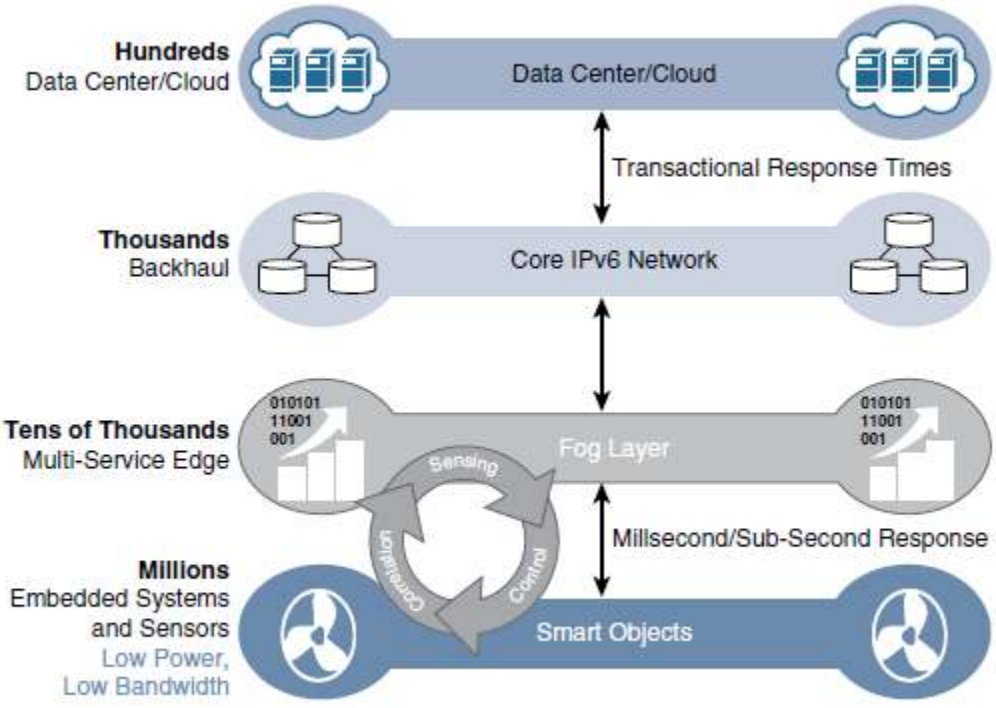


Figure 2-15 *The IoT Data Management and Compute Stack with Fog Computing*

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.

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.

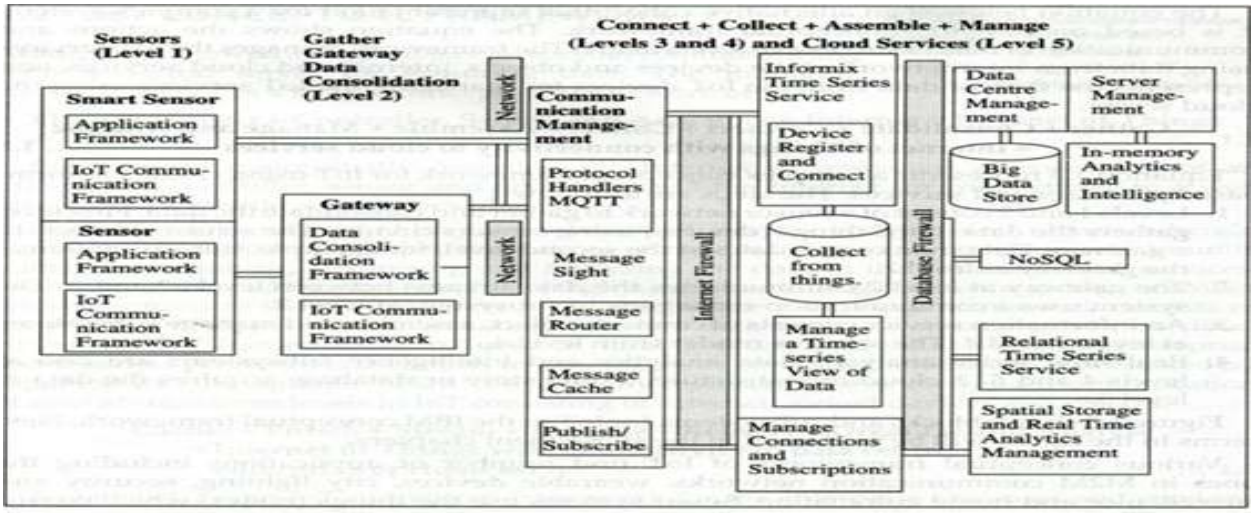
Edge Computing

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 is also sometimes called “mist” computing. If clouds exist in the sky, and fog sits near the ground, then mist is what actually sits on the ground. Thus, the concept of mist is to extend fog to the furthest point possible, right into the IoT endpoint device itself.

IoT Conceptual framework:



IBM IoT Conceptual Framwork

(Curtesy: TMH; IoT Architecture and Design Principles by Raj Kamal)

PRIVACY:

The Internet of Things devices that we own aren't the only ones that should concern us when it comes to matters of trust. With more sensors and devices watching us and reporting data to the Internet, the privacy of third parties who cross our sensors' paths (either by accident or design) is an important consideration. Designers of an Internet of Things service will need to balance these concerns carefully.

KEEPING SECRETS

For certain realms, such as health care, privacy concerns are an obvious issue. However, even seemingly innocuous applications can leak personal information, so you should be alert to the danger and take measures to avoid it. This advice is perfectly illustrated with an example from an early instrumented car park in a Westfield shopping mall in Australia. Each parking bay is overlooked by a small sensor from Park Assist, which uses a cheap camera to tell whether the space is occupied. The sensors are all networked and presumably can provide analytics to the owner of the car park as to its usage. A light on the sensor can help guide drivers to a free space. All useful and harmless stuff. The problem came with a more advanced feature of the system. The
shopping mall provided a smartphone app for visitors to download so that they could find out more information about the facilities. One of the features of the app was a Find My Car option. Choosing that, you were prompted to enter the first few characters of your licence plate, and the app would then return four small photos of potential matches—from optical character recognition software processing the sensor data on the mall's server. The returned images were only thumbnails—good enough to recognize which was your car, but not much else, and the licence plates were blurry
and hard to see. However, security professional Troy Hunt found that the implementation method left a lot to be desired (www.troyhunt.com/2011/09/find-my-car-find-your-car-find.html).
With a fairly simple, off-the-shelf bit of software, Troy was able to watch what information the app was requesting from the server and found that it was a simple unencrypted web request. The initial request URL had a number of parameters, including the search string, but also including information such as the number of results to return.

That request returned a chunk of data (in the easily interpreted, industry standard JSON format), which included the URLs for the four images to download, but also included a raft of additional pieces of information. Presumably, it was easier for the developer of the web service to just return
all the available data than to restrict it to just what was needed in this case. The extra data included, for example, the IP addresses of each of the sensor units, but more importantly, it also included the full licence plate for each vehicle and the length of time it had been parked in the space.
By altering the search parameters, Troy found that he could request many more than the four matches, and it was also possible to omit the licence plate search string. That meant he could download a full list of licence plates from all 2550 parking spaces in a single web request, whenever he liked. Obviously, all that data is already publicly available, but there's a pretty large

difference in ease of gathering it between staking out the entrance to the car park and watching cars come and go and setting up a script on a computer to check at regular intervals. Once alerted to the problem, Westfield and Park Assist were quick to disable the feature and then work with Troy to build a better solution. However, that situation came about only because Troy was generous enough to bring it to their attention.

Don't share more than you need to provide the service.

As founder of WikiLeaks, Julian Assange, has said, "The best way to keep a secret is to never have it" (www.pbs.org/wgbh/pages/frontline/wikileaks/interviews/julian-assange.html). If you can avoid gathering and/or storing the data in the first place, you need not worry about disclosing it accidentally. In this day and age, it is standard practice to never store passwords as clear text. You could also consider applying the standard mechanisms for password encryption, such as the one-way hash, to other pieces of data. This technique was suggested by Peter Wayner in his book *Translucent Databases* (CreateSpace Independent Publishing Platform, 2009). Rather than storing identifying data in the database, if you don't need to return it to its original form (that is, you just need it to be unique and associated with the same group of data), use a one-way hashed version of the information instead. Doing so still lets the originators of the data find their data (as they can provide it to be hashed again) and allows statistics gathering and reports, and the like, without storing the data in a recoverable form.

Hashes:

One-way hashing is a cryptographic technique used to condense an arbitrarily sized chunk of data into a fixed-sized piece, called the hash. It's called one-way hashing because there isn't an easy way, given the resultant hash, to work out what the original data was. Hashing algorithms are so designed such that even a small difference in the input data leads to a huge difference in the output hash. This makes them very useful for times when you want to verify that two pieces of data are identical without having to store them for comparison. That's useful when the data you want to compare is either very large or something you don't want to store in its original form. The most common use of cryptographic hashes is in password verification. Rather than store the user's password, the service provider stores a hash of the password. When the user wants to authenticate himself in the future, the hash can be recalculated; if it matches the stored one, the service can be reasonably sure that the user has provided the correct password. It is good practice to salt the password before applying the hash. This adds some random, non-secret extra text to the password before the hash is computed. The salt is then stored with the hash, so the service can concatenate the two again when it needs to verify a newly presented password. The salt prevents any attacker who ends up with a copy of the hash from easily comparing it to a dictionary of precompiled hashes to work out the password.

Genesis of IoT:

The age of IoT is often said to have started between the years 2008 and 2009. During this time period, the number of devices connected to the Internet eclipsed the world's population. With more "things" connected to the Internet than people in the world, a new age was upon us, and the Internet of Things was born. The person credited with the creation of the term "Internet of Things" is Kevin Ashton. While working for Procter & Gamble in 1999, Kevin used this phrase to explain a new idea related to linking the company's supply chain to the Internet.
Kevin has subsequently explained that IoT now involves the addition of senses to computers. He was quoted as saying: "In the twentieth century, computers were brains without senses—they only knew what we told them." Computers depended on humans to input data and knowledge through typing, bar codes, and so on. IoT is changing this paradigm; in the twenty-first century, computers are sensing things for themselves.
It is widely accepted that IoT is a major technology shift, but what is its scale and importance? Where does it fit in the evolution of the Internet?

As shown in Figure 1-1, the evolution of the Internet can be categorized into four phases. Each of these phases has had a profound impact on our society and our lives. These four phases are further defined in Table 1-1.

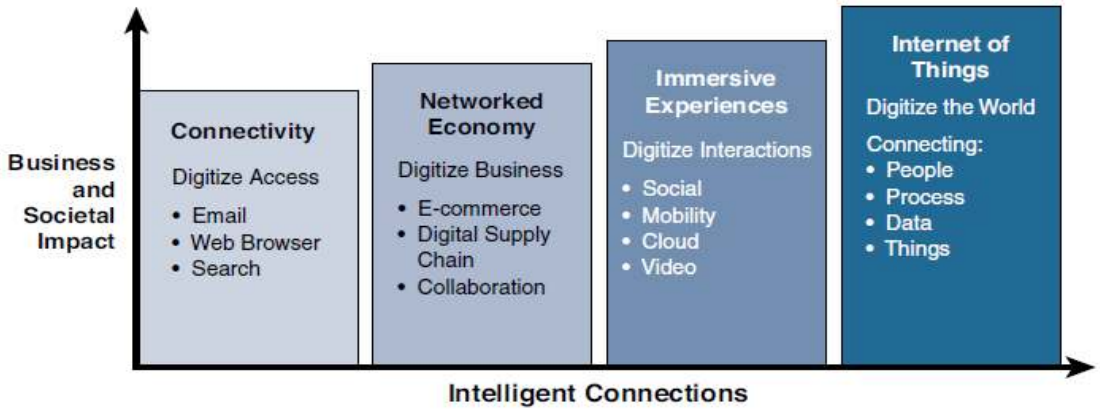


Figure 1-1 *Evolutionary Phases of the Internet*

Table 1-1 *Evolutionary Phases of the Internet*

Internet Phase	Definition
Connectivity (Digitize access)	This phase connected people to email, web services, and search so that information is easily accessed.
Networked Economy (Digitize business)	This phase enabled e-commerce and supply chain enhancements along with collaborative engagement to drive increased efficiency in business processes.
Immersive Experiences (Digitize interactions)	This phase extended the Internet experience to encompass widespread video and social media while always being connected through mobility. More and more applications are moved into the cloud.
Internet of Things (Digitize the world)	This phase is adding connectivity to objects and machines in the world around us to enable new services and experiences. It is connecting the unconnected.

Each of these evolutionary phases builds on the previous one. With each subsequent phase, more value becomes available for businesses, governments, and society in general. The first phase, Connectivity, began in the mid-1990s. Though it may be hard to remember, or even imagine if you are younger, the world was not always connected as it is today. In the beginning, email and getting on the Internet were luxuries for universities and corporations. Getting the average person online involved dial-up modems, and even basic connectivity often seemed like a small miracle. Even though connectivity and its speed continued to improve, a saturation point was reached where connectivity was no longer the major challenge. The focus was now on leveraging connectivity for efficiency and profit. This inflection point marked the beginning of the second phase of the Internet evolution, called the Networked Economy.

With the Networked Economy, e-commerce and digitally connected supply chains became the rage, and this caused one of the major disruptions of the past 100 years.

Vendors and suppliers became closely interlinked with producers, and online shopping experienced incredible growth. The victims of this shift were traditional brick-and-mortar retailers. The economy itself became more digitally intertwined as suppliers, vendors, and consumers all became more directly connected.

The third phase, Immersive Experiences, is characterized by the emergence of social media, collaboration, and widespread mobility on a variety of devices. Connectivity is now pervasive, using multiple platforms from mobile phones to tablets to laptops and desktop computers. This pervasive connectivity in turn enables communication and collaboration as well as social media across multiple channels, via email, texting, voice, and video. In essence, person-to-person interactions have become digitized.

The latest phase is the Internet of Things. Despite all the talk and media coverage of IoT, in many ways we are just at the beginning of this phase. When you think about the fact that 99% of "things" are still unconnected, you can better understand what this evolutionary phase is all about. Machines and objects in this phase connect with other machines and objects, along with humans. Business and society have already started down this path and are experiencing huge increases in data and knowledge. In turn, this is now leading to previously unrecognized insights, along with increased automation and new process efficiencies. IoT is poised to change our world in new and exciting ways, just as the past Internet phases already have.

IoT Challenges:

While an IoT-enabled future paints an impressive picture, it does not come without significant challenges. Many parts of IoT have become reality, but certain obstacles need to be overcome for IoT to become ubiquitous throughout industry and our everyday life. Table 1-4 highlights a few of the most significant challenges and problems that IoT is currently facing.

Table 1-4 *IoT Challenges*

Challenge	Description
Scale	While the scale of IT networks can be large, the scale of OT can be several orders of magnitude larger. For example, one large electrical utility in Asia recently began deploying IPv6-based smart meters on its electrical grid. While this utility company has tens of thousands of employees (which can be considered IP nodes in the network), the number of meters in the service area is tens of millions. This means the scale of the network the utility is managing has increased by more than 1,000-fold! Chapter 5, "IP as the IoT Network Layer," explores how new design approaches are being developed to scale IPv6 networks into the millions of devices.
Security	With more "things" becoming connected with other "things" and people, security is an increasingly complex issue for IoT. Your threat surface is now greatly expanded, and if a device gets hacked, its connectivity is a major concern. A compromised device can serve as a launching point to attack other devices and systems. IoT security is also pervasive across just about every facet of IoT. For more information on IoT security, see Chapter 8, "Securing IoT."
Privacy	As sensors become more prolific in our everyday lives, much of the data they gather will be specific to individuals and their activities. This data can range from health information to shopping patterns and transactions at a retail establishment. For businesses, this data has monetary value. Organizations are now discussing who owns this data and how individuals can control whether it is shared and with whom.
Big data and data analytics	IoT and its large number of sensors is going to trigger a deluge of data that must be handled. This data will provide critical information and insights if it can be processed in an efficient manner. The challenge, however, is evaluating massive amounts of data arriving from different sources in various forms and doing so in a timely manner. See Chapter 7 for more information on IoT and the challenges it faces from a big data perspective.
Interoperability	As with any other nascent technology, various protocols and architectures are jockeying for market share and standardization within IoT. Some of these protocols and architectures are based on proprietary elements, and others are open. Recent IoT standards are helping minimize this problem, but there are often various protocols and implementations available for IoT networks.

Convergence of IT and OT:

Until recently, information technology (IT) and operational technology (OT) have for the most part lived in separate worlds. IT supports connections to the Internet along with related data and technology systems and is focused on the secure flow of data across an organization. OT monitors and controls devices and processes on physical operational systems. These systems include assembly lines, utility distribution networks, production facilities, roadway systems, and many more. Typically, IT did not get involved with the production and logistics of OT environments.

Specifically, the IT organization is responsible for the information systems of a business, such as email, file and print services, databases, and so on. In comparison, OT is responsible for the devices and processes acting on industrial equipment, such as factory machines, meters, actuators, electrical distribution automation devices, SCADA (supervisory control and data acquisition) systems, and so on. Traditionally, OT has used dedicated networks with specialized communications protocols to connect these devices, and these networks have run completely separately from the IT networks.

Management of OT is tied to the lifeblood of a company. For example, if the network connecting the machines in a factory fails, the machines cannot function, and production may come to a standstill, negatively impacting business on the order of millions of dollars. On the other hand, if the email server (run by the IT department) fails for a few hours, it may irritate people, but it is unlikely to impact business at anywhere near the same level. Table 1-3 highlights some of the differences between IT and OT networks and their various challenges.

Table 1-3 *Comparing Operational Technology (OT) and Information Technology (IT)*

Criterion	Industrial OT Network	Enterprise IT Network
Operational focus	Keep the business operating 24x7	Manage the computers, data, and employee communication system in a secure way
Priorities	1. Availability 2. Integrity 3. Security	1. Security 2. Integrity 3. Availability
Types of data	Monitoring, control, and supervisory data	Voice, video, transactional, and bulk data
Security	Controlled physical access to devices	Devices and users authenticated to the network
Implication of failure	OT network disruption directly impacts business	Can be business impacting, depending on industry, but workarounds may be possible

Criterion	Industrial OT Network	Enterprise IT Network
Network upgrades (software or hardware)	Only during operational maintenance windows	Often requires an outage window when workers are not onsite; impact can be mitigated
Security vulnerability	Low: OT networks are isolated and often use proprietary protocols	High: continual patching of hosts is required, and the network is connected to Internet and requires vigilant protection

With the rise of IoT and standards-based protocols, such as IPv6, the IT and OT worlds are converging or, more accurately, OT is beginning to adopt the network protocols, technology, transport, and methods of the IT organization, and the IT organization is beginning to support the operational requirements used by OT. When IT and OT begin using the same networks, protocols, and processes, there are clear economies of scale. Not only does convergence reduce the amount of capital infrastructure needed but networks become easier to operate, and the flexibility of open standards allows faster growth and adaptability to new technologies.

However, as you can see from Table 1-3, the convergence of IT and OT to a single consolidated network poses several challenges. There are fundamental cultural and priority differences between these two organizations. IoT is forcing these groups to work together, when in the past they have operated rather autonomously. For example, the OT organization is baffled when IT schedules a weekend shutdown to update software without regard to production requirements. On the other hand, the IT group does not understand the prevalence of proprietary or specialized systems and solutions deployed by OT.

Take the case of deploying quality of service (QoS) in a network. When the IT team deploys QoS, voice and video traffic are almost universally treated with the highest level of service. However, when the OT system shares the same network, a very strong argument can be made that the real-time OT traffic should be given a higher priority than even voice because any disruption in the OT network could impact the business.

With the merging of OT and IT, improvements are being made to both systems. OT is looking more toward IT technologies with open standards, such as Ethernet and IP. At the same time, IT is becoming more of a business partner with OT by better understanding business outcomes and operational requirements.

The overall benefit of IT and OT working together is a more efficient and profitable business due to reduced downtime, lower costs through economy of scale, reduced inventory, and improved delivery times. When IT/OT convergence is managed correctly, IoT becomes fully supported by both groups. This provides a “best of both worlds” scenario, where solid industrial control systems reside on an open, integrated, and secure technology foundation.

DATA CONSOLIDATION:

What is Data Consolidation?

To the outside world, your business is a highly organized structure. But on the inside, it's a cauldron of raw material collected from databases, documents, and a multitude of other sources. This material — a.k.a. data — has all the potential in the world to help your business transform and grow, so long as you properly corral it all through a process called data consolidation.

Data consolidation definition:

Data is generated from many disparate sources and in many different formats. Data consolidation is the process that combines all of that data wherever it may live, removes any redundancies, and cleans up any errors before it gets stored in one location, like a data warehouse or data lake.

Data consolidation and database replication*. There's another kind of data consolidation, one that has to do with how changes to rows in the data (updated, inserted, or deleted records) in a database are merged with a data warehouse. This "consolidation" — beyond the scope of this post — is often performed on a regular schedule and is used to incorporate changes to data being replicated in order to ensure the "latest" version of data is reflected in the data warehouse*.

At a time when "information creation" is accelerating at exponential rates, data consolidation offers important benefits to organizations that are struggling to tackle today's business challenges. The process helps to ensure greater data quality and accuracy, making it much easier to access, manipulate, and analyze when you're ready. By eliminating the incongruencies that must first be addressed before

operationalizing the data in any way, you can achieve enormous time savings, improve efficiency, and add value to your organization's data operations as a whole.

How to consolidate data?

Data consolidation isn't standard across industries or organizations and there are a few different tools or methods you can use to do it:

- **Hand-coding or scripting.** This manual process custom builds scripting by data scientists to combine and consolidate data from a predetermined range of sources.
- **Open-source tools.** Open-source software helps organizations combine and consolidate data with relatively little cost and more flexibility, but requires a higher degree of expertise in coding and usually more manpower.
- **Cloud-based tools**. A modern approach to data consolidation, cloud-based tools automate many data consolidation tasks with speed, scalability, and security.

Challenges with data consolidation:

Even though data consolidation is a critical stepping-stone on the path to greater business intelligence and faster, more precise decisions, it isn't always realistic for organizations to do it themselves using existing teams and systems. On the upside, this more traditional approach may give the impression that your organization has full control of its data. On the downside, it can introduce a slew of other challenges that cancel out any control you may believe you have.

Here are four common roadblocks that can occur with traditional, on-site data consolidation:

Limited time. IT teams already have their hands full configuring, maintaining, and monitoring on-site hardware and other equipment, in addition to keeping up with the rest of their daily tasks. So spending the necessary hours to script, run, and manage error-free data consolidation may not always be feasible for your current team.

Limited resources. Any data integration process usually requires the help of skilled data scientists. Yet many organizations don't have the budget or internal buy-in to staff up with the right resources to get the job done. The truth is, acquiring specialized knowledge is time-consuming, and hiring it is a hefty investment.

Scattered locations. Many businesses operate with remote or branch locations, which means that data isn't available in a single physical place but has to be secured and managed in multiple locations. When you need to retrieve that outlying data and combine it with local data sources, it can take significantly more time (and a lot more bandwidth). Yet time is not a friend when quick decisions are on the line, since data can fast become outdated.

Security issues. Every place where data is stored opens up the potential for a hack or breach. And moving data to another place during the data consolidation process only increases that potential. As well, most businesses have to adhere to some level of regulatory standards. But patched equipment and having just one systems admin in charge of data management for the entire enterprise makes it much more difficult to maintain security and compliance to the degree necessary.

Data consolidation and ETL:

Data consolidation usually involves four layers of technology: data sources, an **ETL** (extract, transform and load) data pipeline, a **data warehouse** destination, and **business intelligence** (BI) tools.

ETL stands for "extract, transform, and load" — the process a data pipeline uses to replicate data from a source to a traditional data warehouse. In a variant of this process, **ELT**, the transformation step happens after the data is loaded on the target

system, because that's a better way to replicate data from a source system into a cloud data warehouse.

There are two ways to ETL:

1. **Hand coding** is a manual process in which an engineer builds a script to consolidate data from predetermined sources. Although hand-coding is time-consuming and requires a data engineer, it can be useful for smaller jobs with just a couple of sources. It also may be necessary when a source or destination is not supported by other tools.

2. **ETL tools**, both local and cloud-based, exist to expedite the data consolidation process. These tools automate the ETL process and can begin replicating data within minutes of implementation. Cloud-based ETL tools are tested, maintained, and updated constantly by the provider.

Data consolidation best practices

Organizations should plan and execute data consolidation projects carefully. These best practices promote effective data consolidation:

- **Check to see whether data types in your source and target are compatible**: If they're not, you'll have to transform data to address differences among data types.

- **Maintain copies of your data**: **Data lineage** allows an organization to understand exactly what was done to the data — and how — during the consolidation process. You may need information to demonstrate regulatory compliance, or for retracing steps to understand the results of analytics and any business decisions based on them.

- **Standardize character set conversions**: If you work with an application that allows you to store single-byte characters — such as Western languages — and double-byte characters — such as some Asian languages — in a database, the application can convert between these character types. However, when you move the data, the tools processing the data may be unaware that the data is stored in a different format. By standardizing character set conversions, you increase the likelihood of consolidating data for a reliable outcome.

DATA ENRICHMENT:

Data enrichment is a value adding process, where external data from multiple sources is added to the existing data set to enhance the quality and richness of the data. This process provides more information of the product to the customer.

Data enrichment is defined as merging third-party data from an external authoritative source with an existing database of first-party customer data. Brands do this to enhance the data they already possess so they can make more informed decisions. All customer data, no matter the source, begins in its raw form. When this collected data flows into a central data store, it often is ingested into the system in discrete datasets. What you often have when this happens is data being dumped into a data lake, or a data swamp, full of raw information that often isn't useful outside of narrow contexts.

Data enrichment makes this raw data more useful. By adding data from a third party, brands gain deeper insight into their customers' lives. The resulting enriched data is richer and more detailed, which enables brands to more easily personalize their messaging because they know more about their customers. Strong data enrichment processes are a key part of building the golden customer record. One dataset by itself, no matter how detailed, doesn't include every piece of behavioral or transactional data needed to build a comprehensive single view of the customer. This is why data enrichment practices are vital to marketing's long-term goal of delivering personalized experiences.

Two Kinds of Data Enrichment

There are as many types of data enrichment as there are sources to acquire data from, but two of the most common are:

- **Demographic Data Enrichment**: Demographic data enrichment involves acquiring new demographic data, such as marital status and income level, and adding that into an existing customer dataset. The types of demographic data are vast, as are the sources. You could receive a dataset that includes number of kids, type of car driven, median home value, and so on. What matters with demographic enrichment is what your end purpose is. If you want to provide credit card offers, for example, then you might acquire a database that provides the credit rating of a person. Data enriched in this way can be leveraged to improve targeting of marketing offers overall, which is vital in an age where personalized marketing holds sway.
- **Geographic Data Enrichment**: Geographic data enrichment involves adding postal data or latitude and longitude to an existing dataset that includes customer addresses. There are a number of providers that allow you to purchase this data, which can include ZIP codes, geographic boundaries between cities and towns, mapping insights, and so on. Adding this kind of insight into your data is useful in a few contexts. Retailers could use geographically enriched data to determine their next store location. If the retailer wants to capture the most customers within a specific radius, for example 30 miles, they can leverage their enriched data to make that decision. Marketers could also use geographic enrichment to save on bulk mailings of direct mail.

Every form of data enrichment is valid, depending on your business goals. What's important is identifying the kind of data you need to seek out and collect or acquire to get a positive solution. A word of caution, however. Whenever you acquire third-party data or attempt to match two first-party datasets, there must be a common factor that links the two datasets together.

For anonymous customers, this can be a device ID signifying a mobile device or desktop computer. For known customers, this could be a first name and last name or a mailing address. Even an email address can be used as an identifier to match and merge two distinct datasets. Otherwise, the original dataset won't be enriched because there's nothing to show that the two datasets refer to the same customers.

Data Enrichment Process involves:

Attributes Development	**:**	This process is also called as Schema development. Here Data experts having the core knowledge of the products respective to their Industry develop the attribute sets for each product which is complete, valid, consistent and unique.
Attribute Extraction	**:**	Attributes are captured from different sources
Web Research	**:**	Capture product information from manufacturer / vendor websites or catalogs. Part number validation is also involved in this process.
Hard / Soft Copy	**:**	Capture product information from the hard or soft copy catalogs, pdf's and other documents
Manual Sourcing	**:**	Contact manufacturer / vendor to get the product information or high resolution images
Quality Check	**:**	QC is done to reconfirm the accuracy of data in all stages and at the

end of the process by data experts

Benefits of Data Enrichment

Data enrichment has numerous benefits that make it a great infrastructure for companies.

1. Cost savings of Data Enrichment

   A report by Global Databerg contends that an organization with one petabyte of data spends around $650,000 annually to manage the data, yet these companies only use a fraction of their data for any true benefit. Data enrichment saves you money because you don't store information that is not useful to your business. Instead, you enhance the internal data with external sources of data for the benefit of your organization. The funds that would otherwise be used on databases are used on other activities that have a positive effect on the bottom line.

2. Data enrichment fosters meaningful customer relationships

   Enriched data promotes personalized communications and increases the likelihood of meaningful customer relationships and business opportunities. With relevant customer data, your business can develop communication strategies that meet customer preferences and needs. A customer is more likely to make a purchase when they feel that your company understands their needs.

3. Data enrichment maximizes customer nurturing

   Data enrichment maximizes customer nurturing by identifying segments of customers to be nurtured. A segment offers value-driven information that has the potential to evoke a purchase.

4. Data enrichment boost successful targeted marketing

   Targeted marketing is the future of marketing, and many businesses are realizing that a one-size-fits-all marketing approach does not work. They are turning to targeted marketing. For targeted marketing to be successful, an organization requires data enrichment to segment data effectively.

5. Get greater sales with data enrichment

   Imagine investing a huge sum of money on your contact list hoping to get customers and prospects only to discover that your contact list is outdated. Organizations cannot afford such losses. Data enrichment ensures you have a clean and accurate contact list to increase sales efficiency and boost ROI. Also, it offers opportunities for cross-sells and upsells because a business has the right data and knows its customers well.

6. Eradicate redundant data with data enrichment

   Redundant data costs a company significantly. It results in revenue loss, customer loss, and

damaged reputation. Redundant data is common in organizations because they are uncertain of the data to let go and data to keep. You can get rid of redundant data using data enrichment tools like Trifacta Wrangler. Data duplication is common in raw data and affects the quality of data. Data enrichment eliminates it and hence enhances data quality.

7. Data enrichment improves customer experience

   Customers have enormous expectations when it comes to their experience with brands. They expect companies to know them, anticipate their needs, and be relevant. Data enrichment enhances customer experiences by providing unique information on customers. Your business can anticipate customer needs and remain relevant through personalized marketing.

Five Use Cases for Data Enrichment:

1] Increased web form conversions

For most marketers, lead generation is a primary KPI. From that perspective, data enrichment tools can lower the barriers to web form conversions. For example, if you have data enrichment tools to fill up missing data, you can always reduce the number of entry fields on your web forms, thereby making it easier for the site visitors. This means, you can have the "must have" fields alone on your form and the rest is taken care by your data enrichment app.

2. **Lead Scoring**

   With better data and customer signals, you can qualify the leads at the top of your funnel and exponentially increase the lead-to-sales ratio. If you choose the right enrichment app, it can enrich and populate a number of additional fields of information about a lead, which in turn will massively help your lead scoring. You can thus prioritize and pursue the leads that have the highest buying intent and product fit. Enrichment apps are just useful for prospecting, but also understand your existing customers, thus helping you in account management, upselling and cross-selling.

3. **Enhanced Customer Segmentation**

   This is a step that extends from lead scoring and transpires into classifying your prospects and customer into segments according to their sales propensity. A data enrichment tool gives you enough information about your prospects and leads and more importantly ensures up-to-date information by refreshing the data from time to time. Therefore, regardless of the source of the leads, the enrichment app can help you segment your customers and thereby tailor your communications to them.

4. **Hyper-Personalization**

   Today's marketing boils down to the relevance of conversations. As we have discussed in some of our previous blog posts, we are no more in the era of mass marketing. In fact, even personalization and hyper-personalization has moved to mass personalization. Enriched data, helps you personalize on a scale. Rich contextual data about prospects and customers instantly

allows to establish meaningful conversations and improve the overall customer experience. Your communications need to go beyond understanding their demographic and firmographic information. You need to be relevant to their current pursuits, and that is why data enrichment is the way to go.

5. **Improving overall CX**
   Customer Experience (CX) makes all the difference for a prospect to choose between two vendors/partners. From a data standpoint, you need to be able to go beyond readily available data and identify the data that really matters, to enable a better experience for the customer. Enriched data means you have current and relevant data. It means when the insights derived from the updated data is more accurate. Therefore, your campaigns, targeting, and interactions with your customers and prospects are more meaningful than ever.