IoT, or the Internet of Things, describes the network of devices connected to the Internet and the technology that enables communication between devices and the cloud. Devices are often embedded with sensors, software, and other technologies that allow them to connect and exchange data over the internet. Increasingly, many everyday devices like kitchen appliances, televisions, lights, and thermostats are connected to the internet, allowing users more capabilities and customization. Other common examples of IoT include wearable devices like the Fitbit or Apple Watch, AI voice assistants like Alexa, smart cars, and even smart cities.

There are four main components used in IoT:

- **Low-power embedded systems:** Less battery consumption, high performance are the inverse factors that play a significant role during the design of electronic systems.
- **Cloud computing:** Data collected through IoT devices is massive and this data must be stored on a reliable storage server. This is where cloud computing comes into play. The data is processed and learned, giving more room for us to discover where things like electrical faults/errors are within the system.
- **Availability of big data:** We know that IoT relies heavily on sensors, especially in real-time. As these electronic devices spread throughout every field, their usage is going to trigger a massive flux of big data.
- **Networking connection:** In order to communicate, internet connectivity is a must where each physical object is represented by an IP address. However, there are only a limited number of addresses available according to the IP naming. Due to the growing number of devices, this naming system will not be feasible anymore. Therefore, researchers are looking for another alternative naming system to represent each physical object.

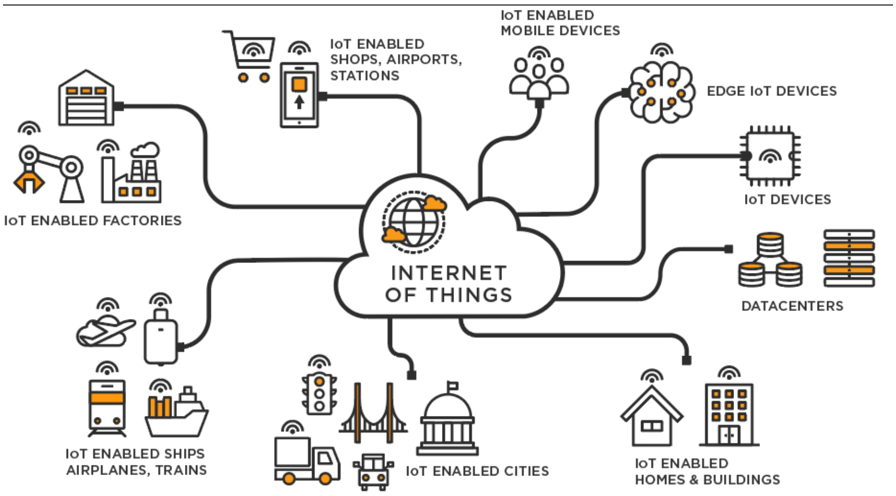


Fig. 1.1 IoT

Architecture Reference Model Introduction

A reference model is a division of functionality together with data flow between the pieces. A reference model is a standard decomposition of a known problem into parts that cooperatively solve the problem. Arising from experience, reference models are a characteristic of mature domains. Can you name the standard parts of a compiler or a database management system? Can you explain in broad terms how the parties work together to accomplish their collective purpose? If so, it is because you have been taught a reference model of these applications.

A reference architecture is a reference model mapped onto software elements (that cooperatively implement the functionality defined in the reference model) and the data flows between them. Whereas a reference model divides the functionality, a reference architecture is the mapping of that functionality onto a system decomposition. The mapping may be, but by no means necessarily is, one to one. A software element may implement part of a function or several functions. Reference models, architectural patterns, and reference architectures arenot architectures; they are useful concepts that capture elements of an architecture. Each is the outcome of early design decisions. The relationship among these design elements is shown in Figure 1

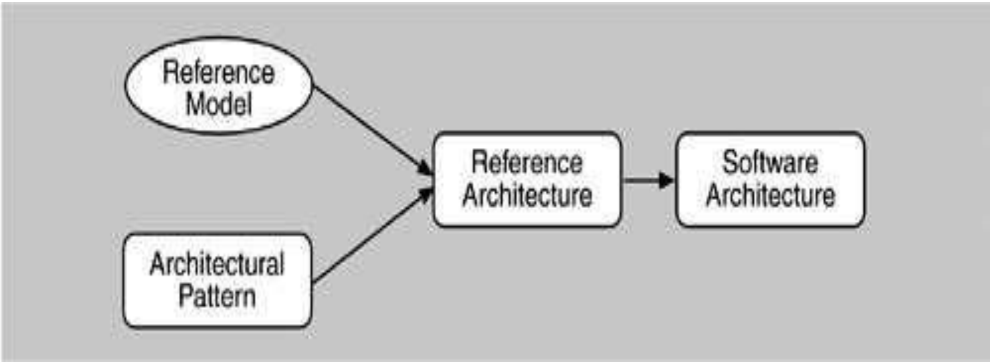


Fig : The relationships of reference models, architectural patterns, reference architectures and software architectures.

IoT Reference Architecture

The reference architecture consists of a set of components. Layers can be realized by means of specific technologies, and we will discuss options for realizing each component. There are also some cross- cutting/vertical layers such as access/identity management.

IoT architecture can comprise up to seven layers, which are known as the perception, transport, edge, processing, application, business, and security layers.

- **Perception Layer:** This layer comprises the physical devices or sensors that collect data from the environment or interact with the physical world. These devices can include temperature sensors, motion detectors, cameras, and other IoT−enabled devices.
- **Network Layer:** The network layer of an IoT system architecture transmits data from multiple devices (e.g., on-site sensors, cameras, actuators) to an on-premises or cloud data center.
    - As a first step, IoT gateways must convert the incoming input from analog to digital format. Next, the gateway may employ any one of a range of data transfer protocols (DTPs) to transmit the data to an on-premises or cloud data center.
        - Amount and type of data to be sent
        - Desired speed and interval of transmission
        - Reliability of network connection
        - Power consumption during data transmission
        - Data and network security
        - Communication among edge devices
    - The different DTPs used in IoT networks are characterized by varying advantages and disadvantages with respect to the above factors. Below are some of the most diverse and widely used IoT protocols:

MQTT (Message Queue Telemetry Transport)

MQTT is a lightweight protocol with publish/subscribe interaction schemes, originally designed by IBM. It has come to be the most widely used protocol in the IoT domain due to its open-source nature and suitability for devices located in remote areas with poor internet connectivity.

Modbus

Originally designed for use with the programmable logic controllers (PLCs) of Modicon (now Schneider Electric), the data communications protocol Modbus is a preferred method for connecting a supervisory computer to remote terminal units (RTUs) in IoT systems, following the supervisory control and data acquisition (SCADA) model.

AMQP (Advanced Message Queuing Protocol)

Spearheaded by JPMorgan Chase, one of the largest banks in the U.S., the AMQP protocol was primarily developed for use in transmitting data within the financial services sector. One of the strengths of the AMQP is its in-built security framework that uses components like transport layer security (TLS) and simple authentication and security layer (SASL).

PROFINET (Process Field Network)

Developed and supported by PROFIBUS & PROFINET International (PI), an automation community based in Germany, the Ethernet-compatible PROFINET protocol has been widely adopted in industrial automation systems requiring communication among multiple edge devices, machinery, and software systems.

CAN (Controller Area Network) bus

Originally designed by German engineering and technology company Bosch, CAN bus was designed for use in the automotive industry, where it enabled different devices and sensors within a vehicle to communicate directly, bypassing the requirement of an intermediary computer.

Subsequently, CAN bus has been adapted for a wide variety of two-way device communication uses, including in maritime vessels, construction equipment, lighting control systems, elevator and escalator controls, amongst others.

EtherCAT (Ethernet for Control Automation Technology)

The EtherCAT Ethernet-based protocol was initially developed by German industrial automation company Beckhoff, for systems requiring real-time updating of data. Supported by the EtherCAT Technology Group (ETG), an industrial collective with nearly 7,000 member organizations, EtherCAT is one of the most widely used IoT gateway protocols.

Other Data Transfer Protocols (DTPs)

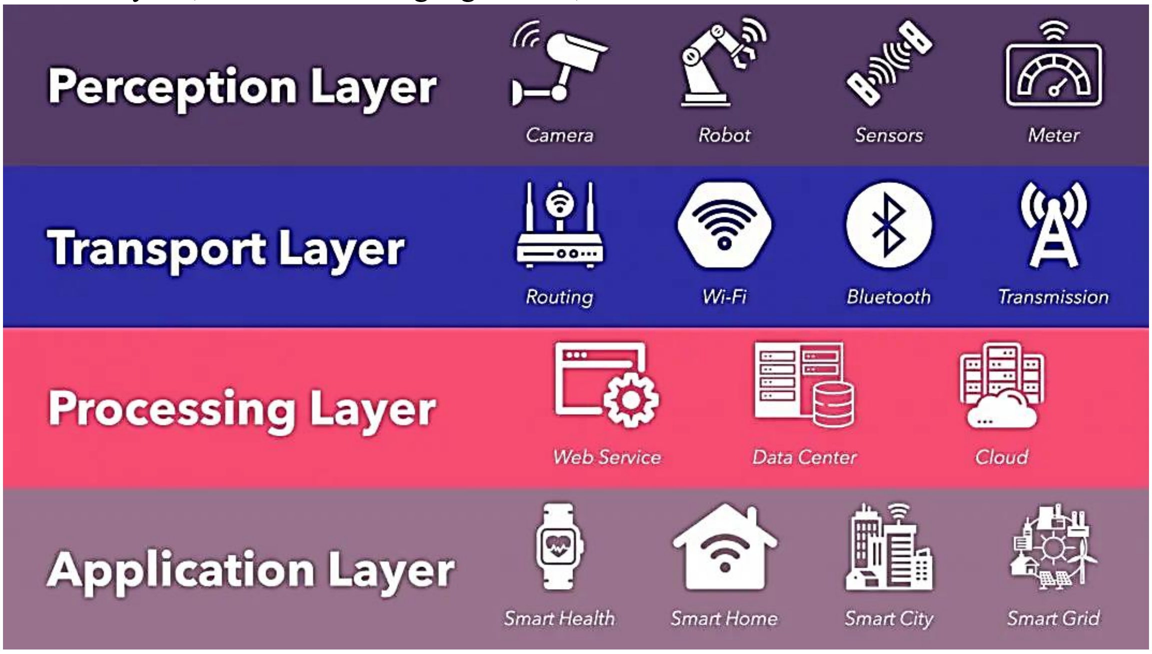
Several other data transfer protocols (DTPs) exist, such as Constrained Application Protocol (CoAP) and Data Distribution Service (DDS), that are also used extensively in industrial as well as non-industrial IoT applications, including domestic lighting, security, and smart healthcare devices.

- **Data Processing Layer:** This layer involves processing and analyzing the data collected from IoT devices. It may include edge computing devices or cloud−based platforms where data is aggregated, filtered, transformed, and analyzed to derive valuable insights.

An IoT system typically handles huge volumes of data, generated by numerous edge devices, at multiple sites on the edges of the network. The 'middleware' of the processing layer utilizes a three-stage approach to prepare this data for the application layer:

- **Data Accumulation:** middleware correctly identifies and assigns different data types to the appropriate storage. **Unstructured** data, such as audio and video streams and images, typically require more storage space and are housed in *data lakes*. Whereas **structured** data, comprising instrument readings, log values, and measurements (telemetry data) are more space-efficient and are stored in *data warehouses*
- **Data Abstraction:** involves aggregating data from multiple sources, as well as ensuring that data is converted into a format that can be "read" by the software of the application layer
- **Data Analysis:** employs machine learning (ML) or deep learning algorithms, which are specialized in detecting patterns within large and seemingly random data sets

- **Application Layer:** The application layer encompasses the software applications or services that utilize the processed IoT data to provide specific functionalities or address specific use cases. These applications can range from real−time monitoring and control systems to predictive analytics, machine learning algorithms, and automation.



IoT Reference Architecture

Q: What's edge and business layer?

IoT Reference Model

A reference model is a model that **describes the main conceptual entities and how they are related to each other**, while the **reference architecture** aims **at describing the main functional components of a system as well as how the system works**, how the system is deployed, what information the system processes, etc.

In an IoT system, data is generated by multiple kinds of devices, processed in different ways, transmitted to different locations, and acted upon by applications. The proposed IoT reference model is comprised of seven levels. Each level is defined with terminology that can be standardized to create a globally accepted frame of reference. The IoT Reference Model does not restrict the scope or locality of its components. For example, from a physical perspective, every element could reside in a single rack of equipment, or it could be distributed across the world. The IoT Reference Model also allows the processing occurring at each level to range from trivial to complex, depending on the situation. The model describes how tasks at each level should be handled to maintain simplicity, allow high scalability, and ensure supportability. Finally, the model defines the functions required for an IoT system to be completed. Figure illustrates the IoT Reference model and its levels. It is important to note that in the IoT, data flows in both directions. In a control pattern, control information flows from the top of the model (level 7) to the bottom (level 1). In a monitoring pattern, the flowof information is the reverse. In most systems, the flow will be bidirectional.

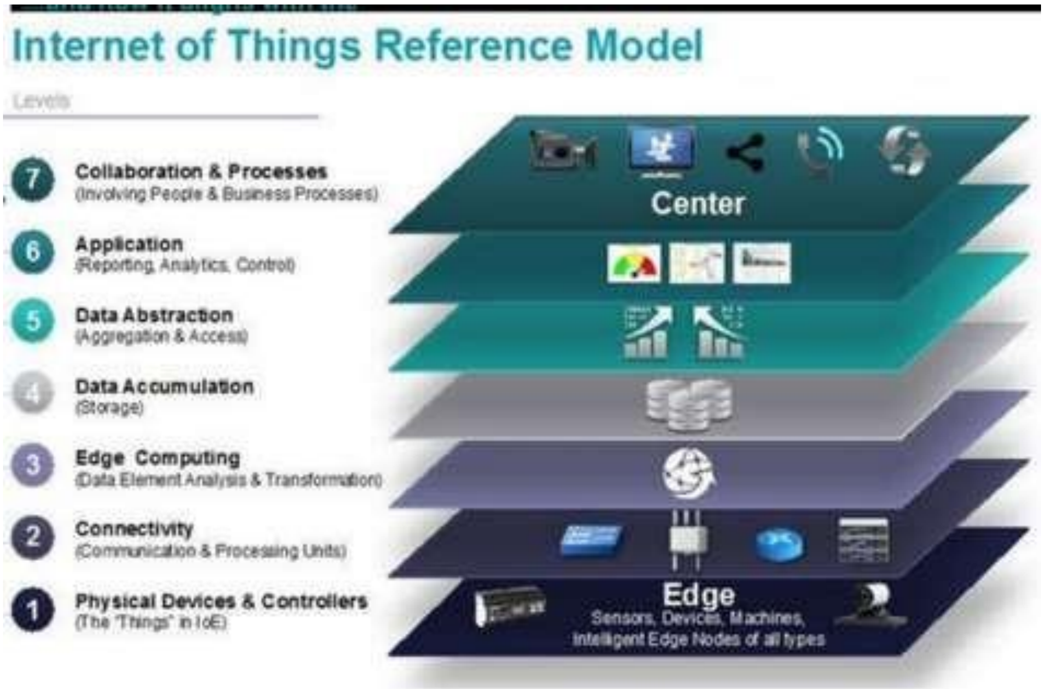


Fig 2.4 IoT Reference Model

Level 1: Physical Devices and Controllers

The IoT Reference Model starts with Level 1: physical devices and controllers that might control multiple devices. These are the "things" in the IoT, and they include a wide range of endpoint devices that send and receive information. Today, the list of devices is already extensive. It will become almost unlimited as more equipment is added to the IoT over time. Devices are diverse, and there are no rules about size, location, form factor, or origin. Some devices will be the size of a silicon chip. Some will be as large as vehicles. The IoT must support the entire range. Dozens or hundreds of equipment manufacturers will produce IoT devices. To simplify compatibility and support manufacturability, the IoT Reference Model generally describes the level of processing needed from Level 1 devices.

Level 2: Connectivity

Communications and connectivity are concentrated in one level—Level 2.

The most important function of Level 2 is reliable, timely information transmission. This includes transmissions: ● Between devices (Level 1) and the network ● Across networks (east-west) ● Between the network (Level 2) and low-level information processing occurring at Level 3

Traditional data communication networks have multiple functions, as evidenced by the International Organization for Standardization (ISO) 7-layer reference model. However, a complete IoT system contains many levels in addition to the communications network. One objective of the IoT Reference Model is for communications and processing to be executed by existing networks. The IoT Reference Model does not require or indicate creation of a different network—it relies on existing networks. However, some legacy devices aren't IP-enabled, which will require introducing communication gateways. Other devices will require proprietary controllers to serve the communication function. However, over time, standardization will increase. As Level 1 devices proliferate, the ways in which they interact with Level 2 connectivity equipment may change. Regardless of the details, Level 1 devices communicate through the IoT system by interacting with Level 2 connectivity equipment

Level 3: Edge (Fog) Computing

The functions of Level 3 are driven by the need to convert network data flows into information that is suitable for storage and higher-level processing at Level 4 (data accumulation). This means that Level 3 activities focus on high-volume data analysis and transformation. For example, a Level 1 sensor device might generate data samples multiple times per second, 24 hours a day, 365 days a year. A basic tenet of the IoT Reference Model is that the most intelligent system initiates information processing as early and as close to the edge of the network as possible. This is sometimes referred to as fog computing. Level 3 is where this occurs. Given that data is usually submitted to the connectivity level (Level 2) networking equipment by devices in small units, Level 3 processing is performed on a packet-by-packet basis. This processing is limited, because there is only awareness of data units—not "sessions" or "transactions." Level 3 processing can encompass many examples, such as:

- Evaluation: Evaluating data for criteria as to whether it should be processed at a higherlevel
- Formatting: Reformatting data for consistent higher-level processing
- Expanding/decoding: Handling cryptic data with additional context (such as the origin)
- Distillation/reduction: Reducing and/or summarizing data to minimize the impact of data and traffic on the network and higher-level processing systems
- Assessment: Determining whether data represents a threshold or alert; this could include redirecting data to additional destinations

Level 4: Data Accumulation

Networking systems are built to reliably move data. The data is "in motion." Prior to Level 4, data is moving through the network at the rate and organization determined by the devices generating the data. The model is event driven. As defined earlier, Level 1 devices do not include computing capabilities themselves. However, some computational activities could occur at Level 2, such as protocol translation or application of network security policy. Additional compute tasks can be performed at Level 3, such as packet inspection. Driving computational tasks as close to the edge of the IoT as possible, with heterogeneous systems distributed across multiple management domains represents an example of fog computing.

Fog computing and fog services will be a distinguishing characteristic of the IoT. Most applications cannot, or do not need to, process data at network wire speed. Applications typically assume that data is “at rest”—or unchanging—in memory or on disk. At Level 4, Data Accumulation, data in motion is converted to data at rest.

Level 4 determines:

- If data is of interest to higher levels: If so, Level 4 processing is the first level that is configured to serve the specific needs of a higher level.
- If data must be persisted: Should data be kept on disk in a non-volatile state or accumulated in memory for short-term use?
- The type of storage needed: Does persistency require a file system, big data system, or relational database?
- If data is organized properly: Is the data appropriately organized for the required storage system?
- If data must be recombined or recomputed: Data might be combined, recomputed, or aggregated with previously stored information, some of which may have come from non-IoT sources.

As Level 4 captures data and puts it at rest, it is now usable by applications on a non-real-time basis. Applications access the data when necessary. In short, Level 4 converts event-based data to query-based processing. This is a crucial step in bridging the differencesbetween the real-time networking world and the non-real-time application world.

Level 5: Data Abstraction

IoT systems will need to scale to a corporate—or even global—level and will require multiple storage systems to accommodate IoT device data and data from traditional enterprise ERP, HRMS, CRM, and other systems. The data abstraction functions of Level 5 are focused on rendering data and its storage in ways that enable developing simpler, performance- enhanced applications.

With multiple devices generating data, there are many reasons why this data may not land in the same data storage:

- There might be too much data to put in one place.
- Moving data into a database might consume too much processing power, so that retrieving it must be separated from the data generation process. This is done today with onlinetransaction processing (OLTP) databases and data warehouses.
- Devices might be geographically separated, and processing is optimized locally.
- Levels 3 and 4 might separate “continuous streams of raw data” from “data that represents an event.” Data storage for streaming data may be a big data system, such as Hadoop. Storagefor event data may be a relational database management system (RDBMS) with faster query times.
- Different kinds of data processing might be required.

For example, in-store processing will focus on different things than across-all-stores summary processing. For these reasons, the data abstraction level must process many different things. These include:

- Reconciling multiple data formats from different sources
- Assuring consistent semantics of data across sources
- Confirming that data is complete to the higher-level application

- Consolidating data into one place (with ETL, ELT, or data replication) or providing accessto multiple data stores through data virtualization
- Protecting data with appropriate authentication and authorization
- Normalizing or denormalizing and indexing data to provide fast application access Level

Application Level 6

It is the application level, where information interpretation occurs. Software at this level interacts with Level 5 and data at rest, so it does not have to operate at network speeds. The IoT Reference Model does not strictly define an application. Applications vary based on vertical markets, the nature of device data, and business needs. For example, some applications will focus on monitoring device data. Some will focus on controlling devices. Some will combine device and non-device data. Monitoring and control applications represent many different application models, programming patterns, and software stacks, leading to discussions of operating systems, mobility, application servers, hypervisors, multi- threading, multi-tenancy, etc. These topics are beyond the scope of the IoT Reference Model discussion. Suffice it to say that application complexity will vary widely. Examples include:

- Mission-critical business applications, such as generalized ERP or specialized industrysolutions
- Mobile applications that handle simple interactions
- Business intelligence reports, where the application is the BI server
- Analytic applications that interpret data for business decisions
- System management/control center applications that control the IoT system itself and don'tact on the data produced by it

If Levels 1-5 are architected properly, the amount of work required by Level 6 will be reduced. If Level 6 is designed properly, users will be able to do their jobs better.

Level 7: Collaboration and Processes

One of the main distinctions between the Internet of Things (IoT) and IoT is that IoT includes people and processes. This difference becomes particularly clear at Level 7: Collaboration and Processes. The IoT system, and the information it creates, is of little value unless it yieldsaction, which often requires people and processes. Applications execute business logic to empower people. People use applications and associated data for their specific needs. Often, multiple people use the same application for a range of different purposes. So, the objective isnot the application—it is to empower people to do their work better. Applications (Level 6) give business people the right data, at the right time, so they can do the right thing. But frequently, the action needed requires more than one person. People must be able to communicate and collaborate, sometimes using the traditional Internet, to make the IoT useful. Communication and collaboration often require multiple steps. And it usually transcends multiple applications. This is why Level 7, represents a higher level than a single application.

https://iotnotesbyparita.wordpress.com/architecture-reference-model/

Views: https://iotnotesbyparita.wordpress.com/iot-reference-architecture/