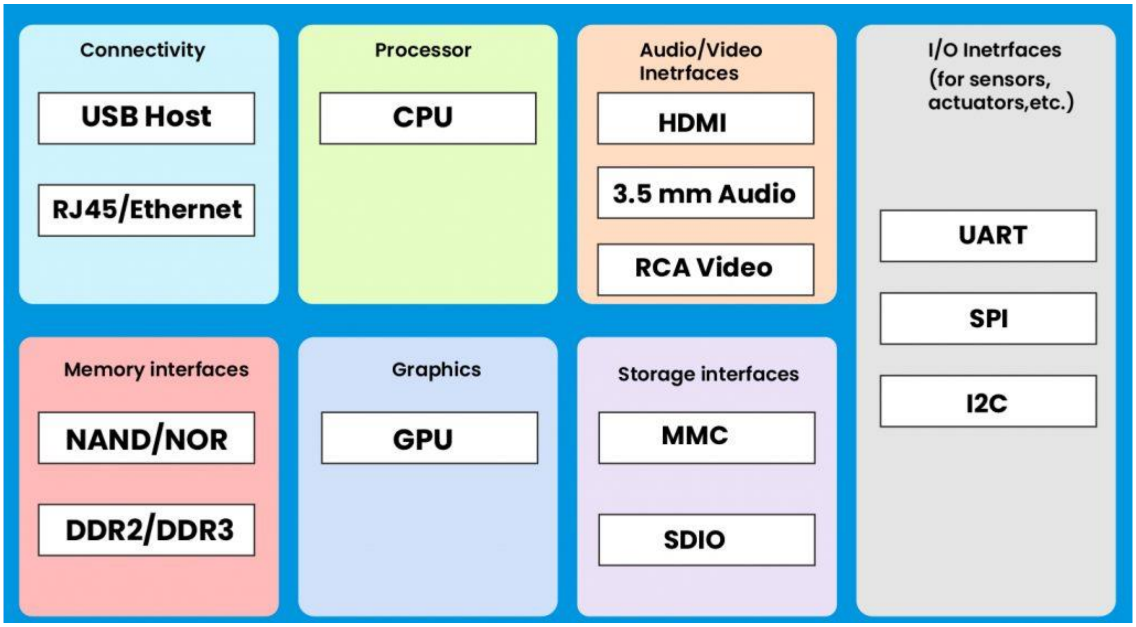
Logical and Physical Designs of IOT

The Internet of Things (IoT) is the physical devices that are connected to a network. These physical devices are called node devices. These are embedded with sensors, software, and other technologies to exchange data with other devices and systems over the Internet. With IoT, digital systems can record, monitor, and adjust each interaction between connected things. Hence now we can connect everyday objects like kitchen appliances, cars, thermostats, etc to the internet via embedded devices. This makes communication streamlined between people, processes, and things.to understand IoT properly we need to understand the Logical and Physical Designs of IOT in detail.

Physical Design of IoT



Now let us understand the physical design of IoT from the logical and physical design of IoT. The Physical design of IoT deals with the individual devices connected to the IoT network and the protocols used to create a functional IoT environment. Each IoT device can perform tasks of remote sensing, actuating, monitoring, etc due to the IoT network they are connected to. These can also transmit information through different types of wireless or wired connections. They can generate data, which is used to perform analysis and perform operations for improving the system.

Now let us understand the IoT node devices and the IoT protocols.

Node Devices

Node devices are used to build a connection, process data, and provide interfaces, and storage, in an IoT system. They generate data that can be analyzed by the IoT system and program to perform operations and improve the system.

Now let's understand which devices are used for which function in an IoT system.

- **Connectivity:** Devices like **USB hosts** and **ETHERNET** provides connectivity between the devices and the server.
- **Processor: Processors** like **CPU** and other units process the data. This is used to improve the decision quality of an IoT system.
- **Audio/Video Interfaces:** System interfaces like **HDMI** and **RCA** devices record audio and videos.
- **Input/Output interface:** Devices like **UART, SPI, CAN,** etc give input and output signals to sensors and actuators.
- **Storage Interfaces:** IoT devices like **SD, MMC,** and **SDIO** generate data**.** Storage interfaces store those data.
- **Controlling of activity:** Devices like **DDR** and **GPU** control the activity of an IoT system.

IoT Protocols

IoT protocols establish communication between a node device and a server over the internet by sending commands to an IoT device and receiving data from an IoT device. Both the server and client-side use different types of protocols. By network layers, they are managed. Some of the network layers are the application, transport, network, and link layers. It works as a building block for logical and physical design of IoT.

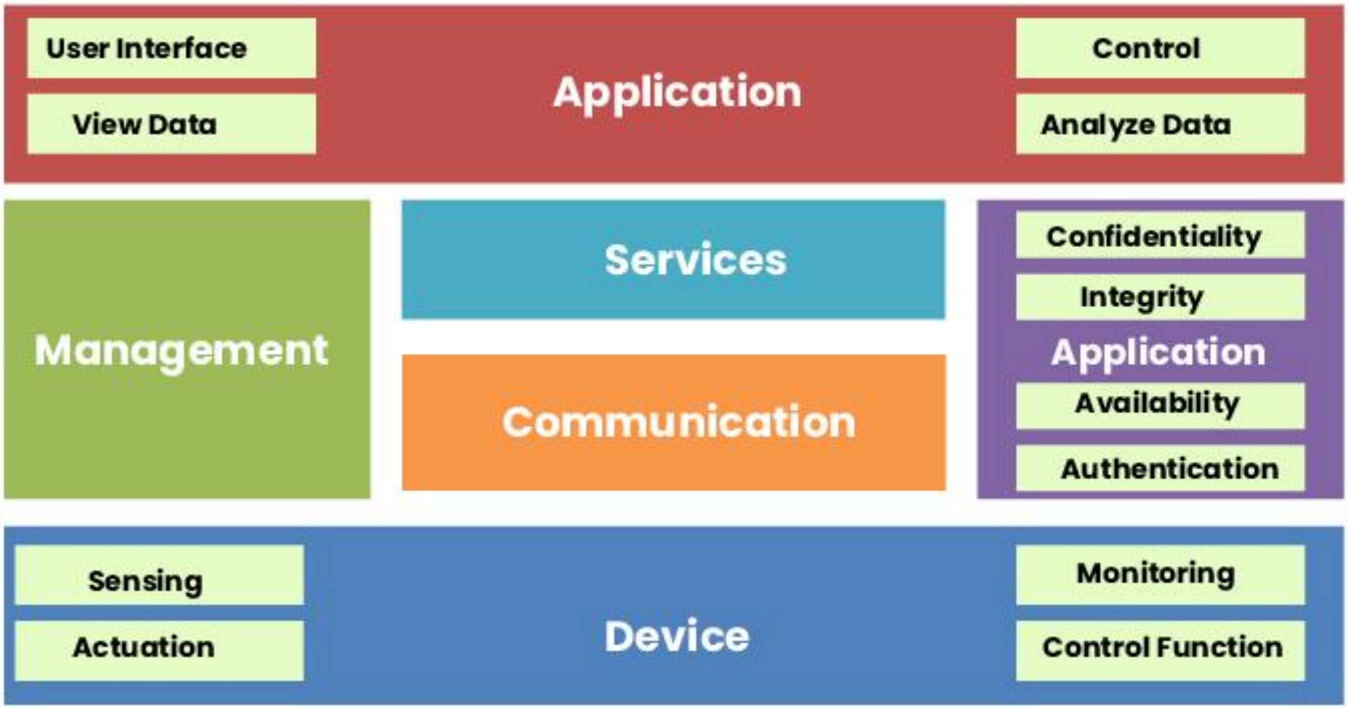
Some of the protocols are

- **Application Layer protocol-** Protocols in this layer define how the data is sent over the network. With the lower layer protocols, the application interface sends these data. Examples include HTTP, WebSocket, XMPP, MQTT, DDS, and AMQP protocols.
- **HTTP (HyperText Transfer Protocol)-** This protocol for transmitting media documents in an application layer by communicating between web browsers and servers.
- **WebSocket-** it enables two-way communication between a client and a host and is mostly web browsers use it.
- **Transport Layer-** it controls the flow of data segments. It also handles error control and provides end-to-end message transfer capability.
- **TCP (Transmission Control Protocol)-** It establishes and maintains a network that can exchange data using the internet protocol.

Some of the layers are

- **Network Layer-** it is used to send datagrams from the source network to the destination network. Some examples are IPv4 and IPv6.
- **Link Layer-** it is used to send data over the network's physical layer and determines how the packets are coded and signaled by the devices.
- **Ethernet-** It is a set of protocols used in LANs that defines the physical layer and the medium access control for LANs.
- **WiFi-** It is a set of LAN protocols. It specifies the set of media access controls and also the physical layer protocols that are used for implementing wireless LANs.

Logical Design of IoT



It is the actual design of the IoT system. It illustrates the assembling and configuration of the components i.e. computers, sensors, and actuators.

The logical design of IoT is composed of:

1. IoT functional blocks
2. IoT communications models
3. IoT communication APIs

1. IoT functional blocks

The functional blocks of IoT systems provide sensing, identification, actuation, management, and communication capabilities to the IoT ecosystem. The devices of the functional blocks handle the communication between the server and the host. This enables monitoring of control functions, managing the data transfer, securing the IoT system using authentication, and providing an interface for controlling and monitoring various functions. It is the most crucial part of the logical and physical designs of IoT.

The Functional blocks of IoT are:

- **Device**

Devices provide sensing, actuation, monitoring, and control functions to the IoT ecosystem.

- **Communication**

Manages communication for the IoT system.

- **Services**

Services perform the functions of device monitoring, device control, data publishing, and device discovery.

- **Management**

gives functions to govern the IoT system.

- **Security**

provides security to the IoT system. It performs authentication, authorization, message and content integrity, and data security functions.

- **Application**

An interface where the users can control and monitor various aspects of the IoT system. It allows the users to view the system status and analyze the processed data.

2. IoT Communication Models

The communication models of IoT are used for communicating between the system and the server. The types of IoT communication models are

- **Request-Response Model**

In this communication model, the client sends requests to the server and the server responds to their requests. After receiving a request, the server decides how to respond by fetching the data, retrieving resource representation, preparing the response, and then sending the response to the client. The request-response protocol between a client and a server is HTTP.

- **Publisher-Subscriber Model**
- Publisher-Subscriber Model

This model is made up of three entities: Publishers, Brokers, and Consumers.

- Publishers- it is the source of data that sends the data to the topic.
- Consumers- they subscribe to the topics.
- Brokers- they accept data from publishers and send it to the consumers.
- **Push-Pull Model**

This model is made up of data publishers, data consumers, and data queues.

- Publishers- they publish the message and push it into the queue.
- Consumers- they present on the other side and they pull the data out of the queue.
- Queues- it helps in decoupling the messages between the producer and consumer.
- **Exclusive Pair**

It is a bi-directional model that includes full-duplex communication between client and server. The client sends a request and the server keeps the record of all the connections. In this model, only WebSocket-based communication API is based.

3. IoT communication API

There are two types of communication APIs –

There are two types of communication APIs –

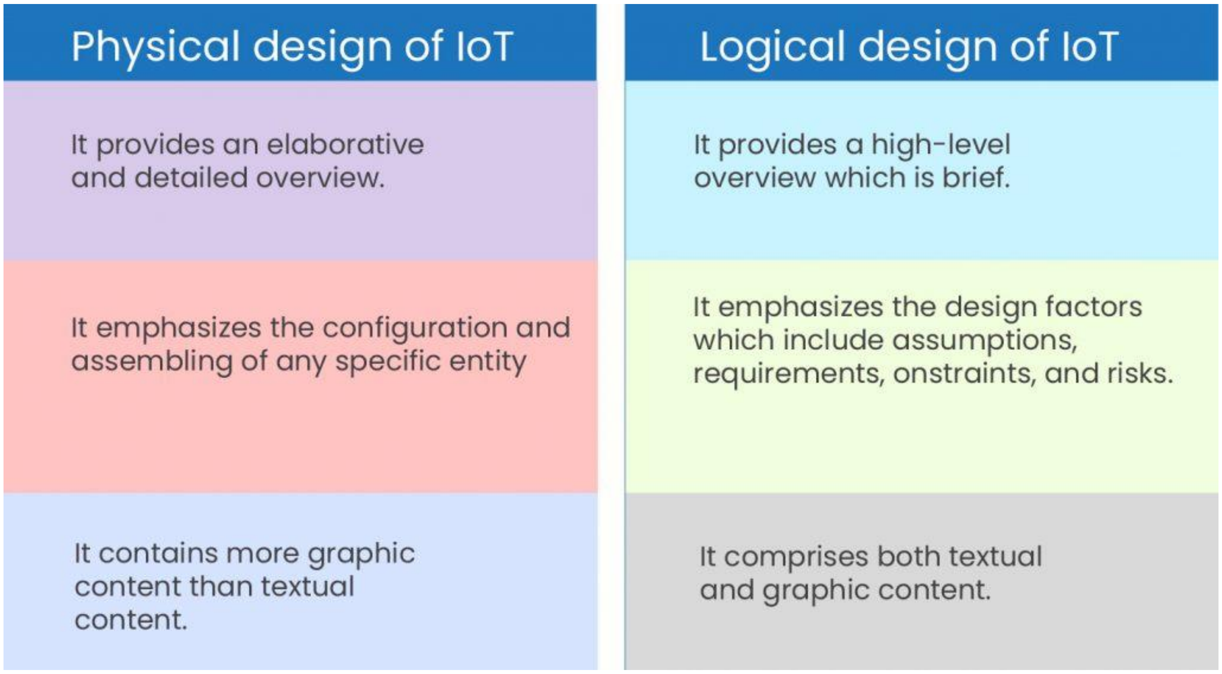
- **REST-based communication APIs**

REST stands for Representational State Transfer. It is a set of architectural protocols by which you can design web services and web APIs. These web services and APIs focus on a system's resources and how resource states are addressed and transferred. It has a request-response communication model. Its architectural constraints are components, connectors, and data elements. All of them are included within a distributed hypermedia system.

- **Web Socket-Based Communication APIs**

Web Socket API is the bi-directional, full-duplex communication model between clients and servers. It does not require a new connection to set up for each message between clients and servers. After the connection is set the messages can be sent and received continuously without any interruption. It is best for IoT Applications with low latency or high throughput requirements

Difference between the physical design of IoT & logical design of IOT



Physical design of IoT	Logical design of IoT
It provides an elaborative and detailed overview.	It provides a high-level overview which is brief.
It emphasizes the configuration and assembling of any specific entity	It emphasizes the design factors which include assumptions, requirements, onstraints, and risks.
It contains more graphic content than textual content.	It comprises both textual and graphic content.

Conclusion

IoT is the building block of many technological advancements these days. Some prominent examples of IoT are connected cars, smart appliances, connected security systems, smart agriculture equipment, connected retail, connected healthcare monitors, connected manufacturing equipment, and connected cities. Hence understanding IoT is crucial if you want to create a smart solution. This article gives an overview of the logical and physical design of IoT so that you know what makes the smart solutions we use nowadays.