



# SNS COLLEGE OF TECHNOLOGY COIMBATORE



AN AUTONOMOUS INSTITUTION

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A++’ Grade

Approved by AICTE New Delhi & affiliated to the Anna University, Chennai

## DEPARTMENT OF Computer Applications

**Course Name : 19CAE725- INTERNET OF THINGS**

**Class : I Year / II Semester**

**Unit V – Cases Studies and Advanced Topics**

**Topic 2 – Connecting cloud with IoT**



The development of smart things, sensors and telecommunication, IoT (Internet of Things) technology has greatly developed and become more and more standardized. But fragmented device-side communication connection problems often impede the project implementation process.

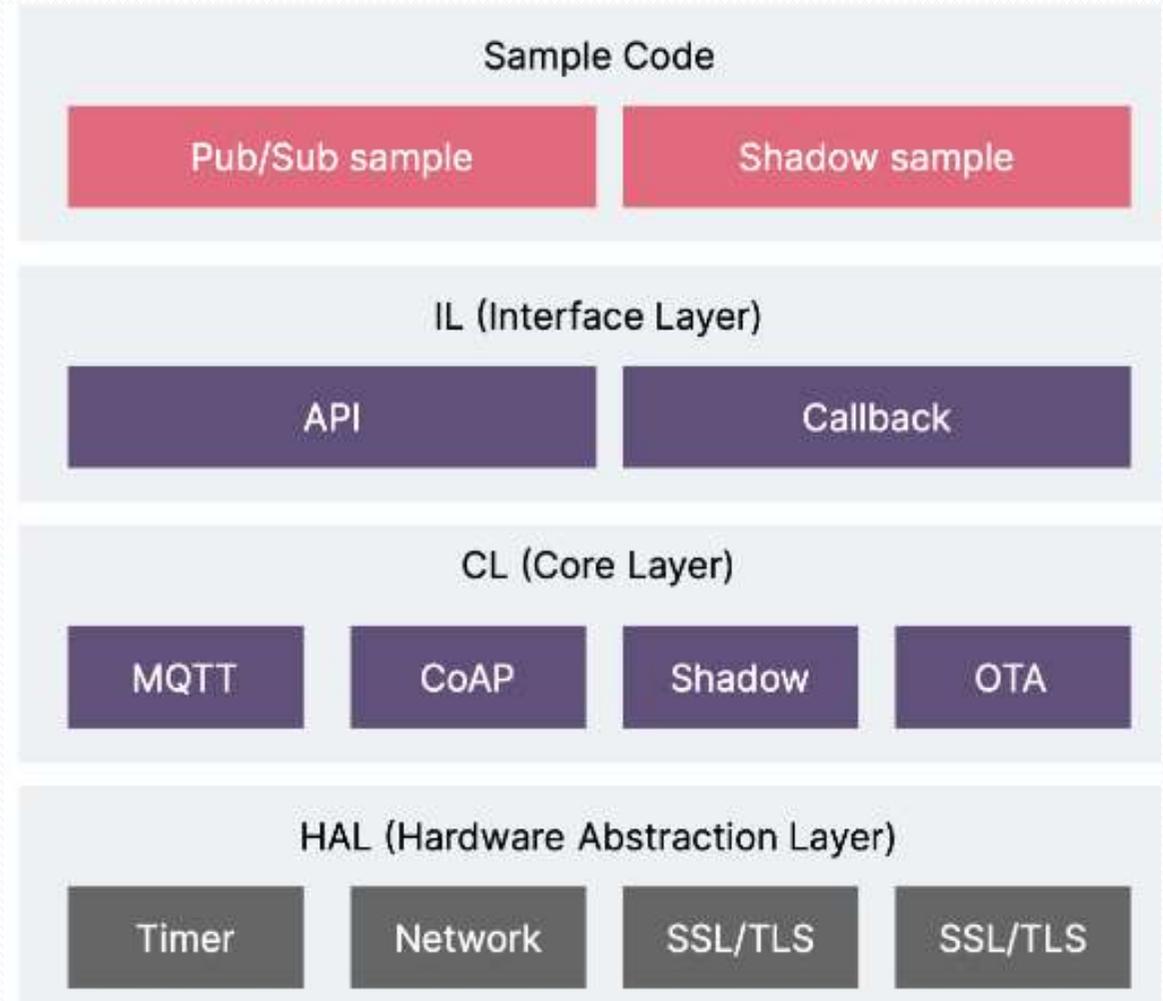
There are four best practices to connect different types of devices to the cloud:

- Directly integrate IoT SDK (Software Development Kit) for resource-rich devices
- Rely on a communication module for resource-constrained devices
- Use a local gateway for non-network devices
- Use a cloud gateway for private-protocol devices with network capabilities



First, we should introduce the IoT Device SDK.

IoT Device SDK is used to help us quickly connect hardware devices to the IoT platform. We can download the IoT Device SDK from the corresponding cloud platform, e.g. [Aws IoT device SDK](#)



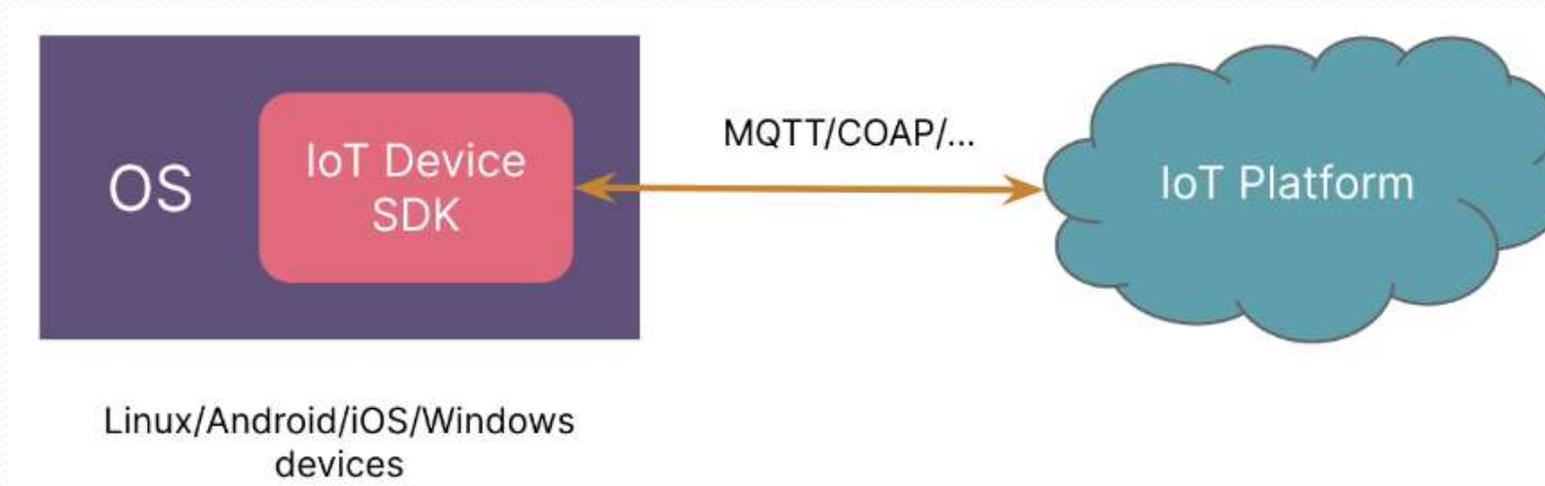


There are **4 layers of the IoT SDK**. From bottom to top:

1. The HAL (hardware abstraction layer) abstracts the support function interface of different OS (operating systems) to the SDK. This enables the SDK to be ported to different hardware environments, different OS, and even bare chip environments.
2. The core layer completes the function encapsulation of MQTT/CoAP communication based on the HAL layer interface, including MQTT connection establishment, message sending and receiving; CoAP connection establishment, message sending and receiving; shadow device operation; OTA firmware status query, download and upgrade.
3. Interface layer, providing API and callback function definitions, isolating the core layer and the application.
4. Provide sample programs so that developers can quickly learn how to use the SDK.



# Directly integrate IoT SDK for resource-rich devices



Examples of smart devices include smart phones, tablets, smart wearables, smart POS, computers, industrial gateways and development boards like the Raspberry Pi and ESP32.

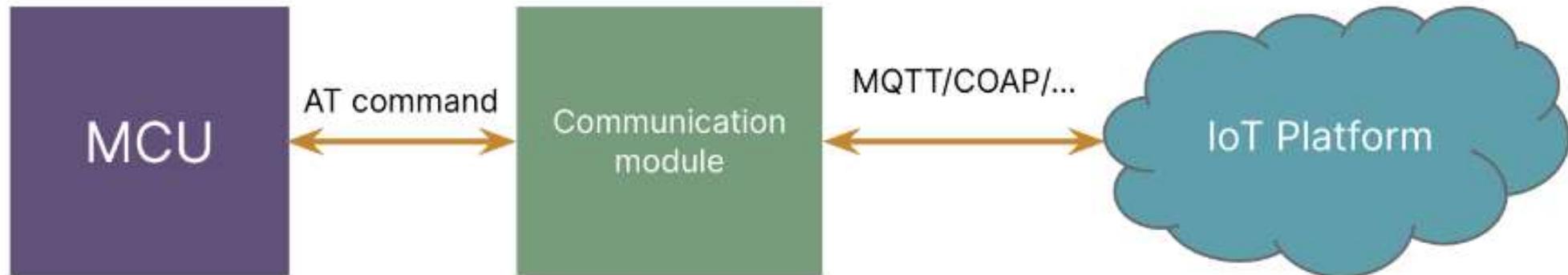


# Rely on a communication module for resource-constrained devices

In the IoT scenario, a large number of devices are resource-constrained, with RTOS, or even without an operating system, using MCU + communication modules to establish their link to the cloud.

There are many suppliers of cellular modules (NB-IoT/2G/3G/4G) on the market. The AT commands of each company are different, which makes developing device-side applications very difficult.

When we need to connect MCU to an IoT Platform, we should always carefully select the cellular modules and check whether they are suitable for a specific IoT platform.





Difference between DTU and industrial gateway:

DTU is a wireless terminal device used to convert serial data into IP data or IP data into serial data and transmit it through a wireless communication network. It has fast and flexible networking, a short construction period and low cost.

The industrial gateway has the functions of collecting data from field devices through serial port or network port. Data collection, protocol analysis, data standardization and uploading to the IoT platform through edge computing functions. Which is more flexible, powerful and customizable than DTU, but it is more expensive and needs more resources to maintain.

Examples of sensors / devices which can be connected to DTU or industrial gateway include sensors, industrial equipment PLCs (Programmable Logic Controller), Bluetooth bracelets





# Use a cloud gateway for private-protocol devices with network capabilities

*Devices that directly connect to the cloud gateway*

For some devices, they already have the ability to connect to the internet, but the protocols vary according to device manufacturers. We don't want the IoT platform layer to handle the parsing of these protocols directly an intermediate layer should satisfy the protocol conversion work to make the data meet the unified format of the IoT platform.

This intermediate layer is the cloud gateway. The cloud gateway is at the front of the platform. It receives data from the device side, completes message parsing, and then sends a message to the IoT platform with IoT device SDK.



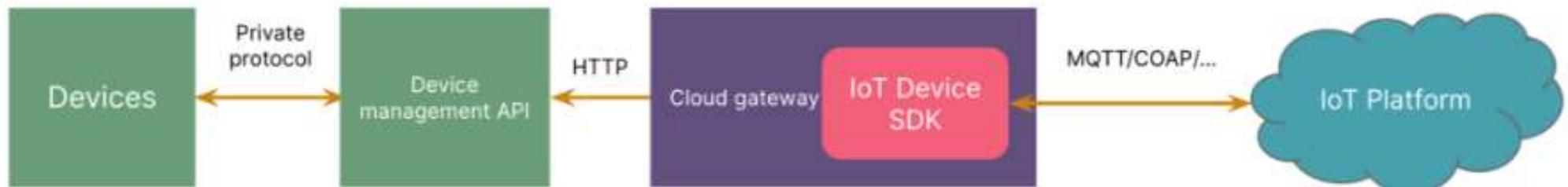
Examples of devices that connect to the internet through private protocol include vehicle GPS.



### *Devices that connect to the cloud gateway indirectly*

Some device vendors already provide a mature system which manages the devices and provides API. In this case, we can have another form of cloud gateway: Cloud to cloud connection. Making full use of the existing system will make the overall system more stable and give clear rights and responsibilities.

Using a mature system will bring us higher development efficiency, but it'll also introduce another middleware, which will increase communication time.



Examples of devices with mature systems include cameras/NVR systems.



## **Conclusion**

When we need to connect a device to the cloud, we should consider system maintainability, communication efficiency, development difficulty and budget. Think carefully about the most appropriate path for your project.

**Any Questions ?**