IoT Platform Secondary Development

HUAWEI

Foreword

- The HUAWEI CLOUD IoT platform enables southbound and northbound data exchange. Developers need to perform secondary development using this platform to implement end-to-end IoT services.

Objectives

- Upon completion of this course, you will:
  - Understand the main content of product development
  - Be able to describe development process on the device side
  - Master the process of calling APIs developed on the application side
  - Understand content and operations of routine cloud management

Contents

Introduction to Platform Secondary Development

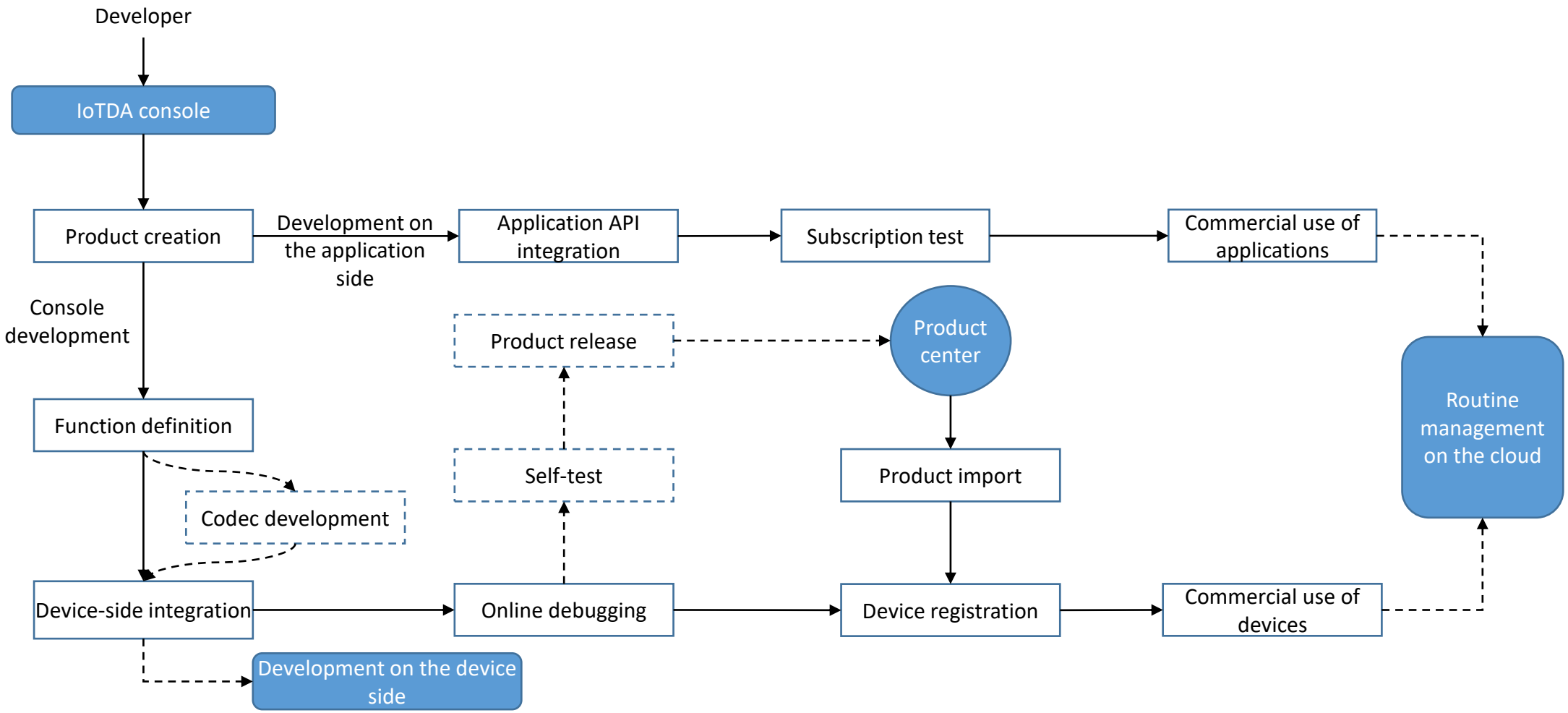
- To create an IoT solution based on the HUAWEI CLOUD IoT platform, you must perform the operations described in the table below.

Operation	Description
Product development	Manage products, develop product models and codecs, and perform online debugging on the IoT Device Access (IoTDA) console.
Development on the application side	Carry out development for interconnection between applications and the platform, including calling APIs, obtaining service data, and managing HTTPS certificates.
Development on the device side	Integrate and interconnect devices with the IoT platform, including connecting devices to the IoT platform, reporting service data to the platform, and processing commands delivered by the platform.

- The process of using IoTDA, including product, application, device, and routine management.
  - **Product development:** You can perform development operations on the IoTDA console. For example, you can create a product or device, develop a product model or codec online, perform online debugging, carry out self-service testing, and release products.
  - **Development on the application side:** The platform provides robust device management capabilities through APIs. You can develop applications based on the APIs to meet requirements in different industries such as smart city, smart campus, smart industry, and IoV.
  - **Development on the device side:** You can connect devices to the platform by integrating SDKs or modules, or using native protocols.
  - **Routine management:** After a physical device is connected to the platform, you can perform routine device management on the IoTDA console or by calling APIs.

Contents

Development Process: Product Development and Development on the Application Side

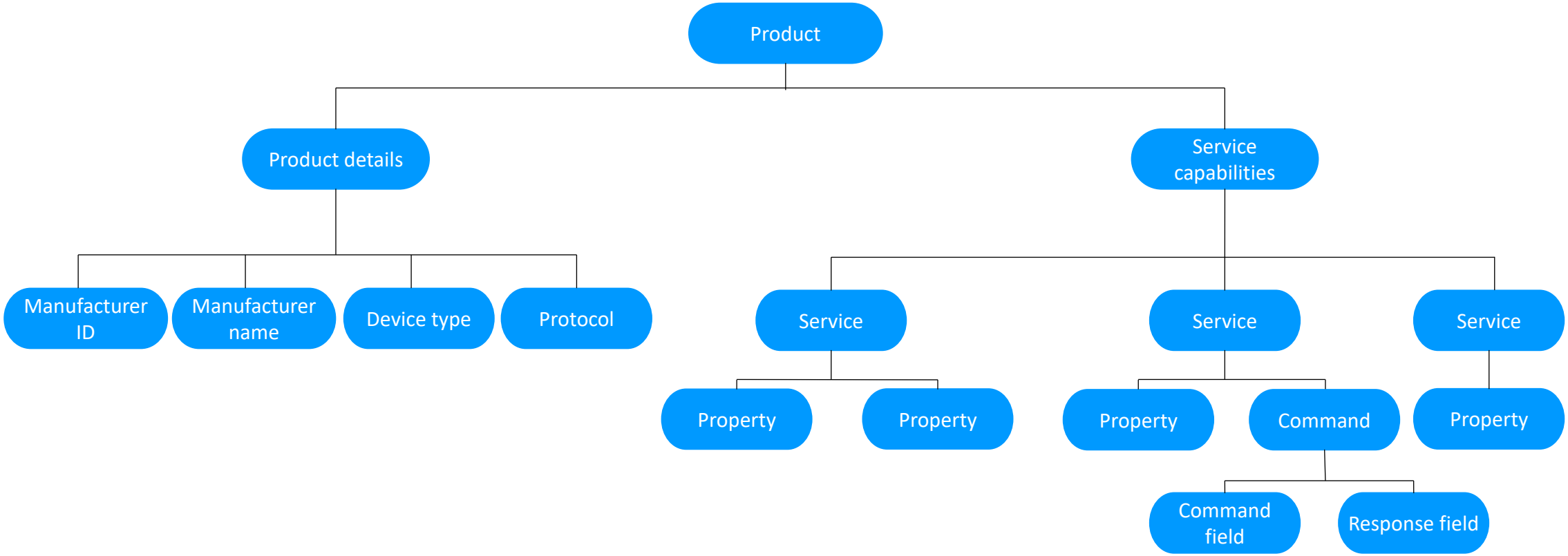


HUAWEI

Product Model (1)

- A product model, also known as a **profile**, defines the properties of a device, such as the color, size, collected data, identifiable commands, and reported events. The manufacturer, device type, and device model are used together to uniquely identify a product model. You can easily develop product models on the IoTDA console without writing any code.
- A profile (product model) is a file that describes **what a device is, what it can do, and how to control it**. You can build an abstract model of a device by defining a product model on the IoT platform so that the platform can know what services, properties, and commands are supported by the device, such as its color or any on/off switches. After defining a product model, you can use it for device registration.

Product Model (2)



- On the IoT platform, the product model is the key to device access. It contains the capabilities and services of a device and the data formats of upstream and downstream device messages. For example, when a device reports data to the IoT platform, the IoT platform matches the product model based on the keywords of the reported data and verifies the data format. **Only data that is matched is saved on the IoT platform.** If the reported data is not matched with the configuration in the product model, the data is considered invalid and dropped.

Product Model (3)

- Product Details
  - Product details describe basic information about a device, including the manufacturer ID, manufacturer name, device type, and protocol.
  - For example, the manufacturer name of a water meter could be ‘HZYB', the manufacturer ID 'TestUtf8ManuId', the device type 'WaterMeter', and the protocol 'CoAP'.
- Service Capabilities
  - Service capabilities of a device need to be defined. Device capabilities are divided into several services. The properties, commands, and command parameters of each service are defined in the product model.
  - Take a water meter as an example. It has multiple capabilities, such as reporting data about the water flow, alarms, power, and connections, and receiving commands from a server. When describing the capabilities of a water meter, the profile includes five services, each of which has its own properties or commands.

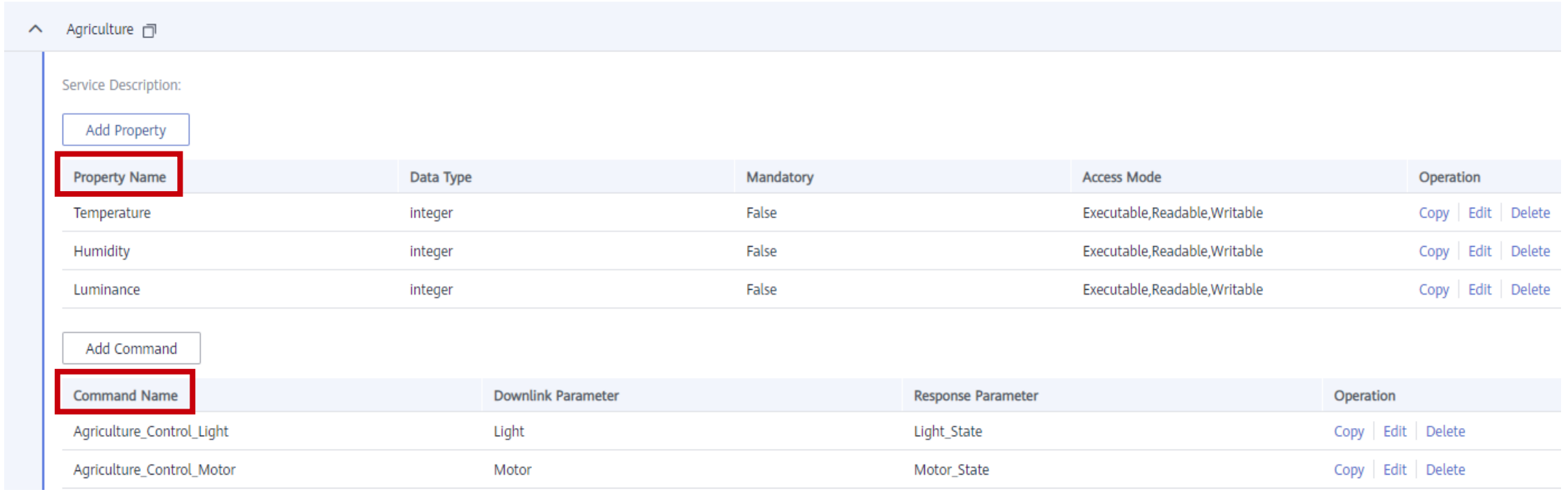
Service Capability - Water Meter

Service	Description
Basics (WaterMeterBasic)	Used to define parameters reported by the water meter, such as the water flow, temperature, and pressure. If these parameters need to be controlled or modified using commands, you also need to define parameters in the commands.
Alarm (WaterMeterAlarm)	Used to define data reported by the water meter in various alarm scenarios. Commands need to be defined if necessary.
Battery (Battery)	Used to define data including the voltage and current intensity of the water meter.
Transmission rule (DeliverySchedule)	Used to define transmission rules for the water meter. Commands need to be defined if necessary.
Connectivity (Connectivity)	Used to define connection parameters of the water meter.

- The HUAWEI CLOUD IoT platform provides multiple methods for developing product models. You can select one that suits your needs.
  - Importing models (preset product models on the platform)
  - Uploading a profile (offline development)
  - Importing models in an Excel file
  - User-defined functions (online development)

Profile Example

Properties/Commands



Contents

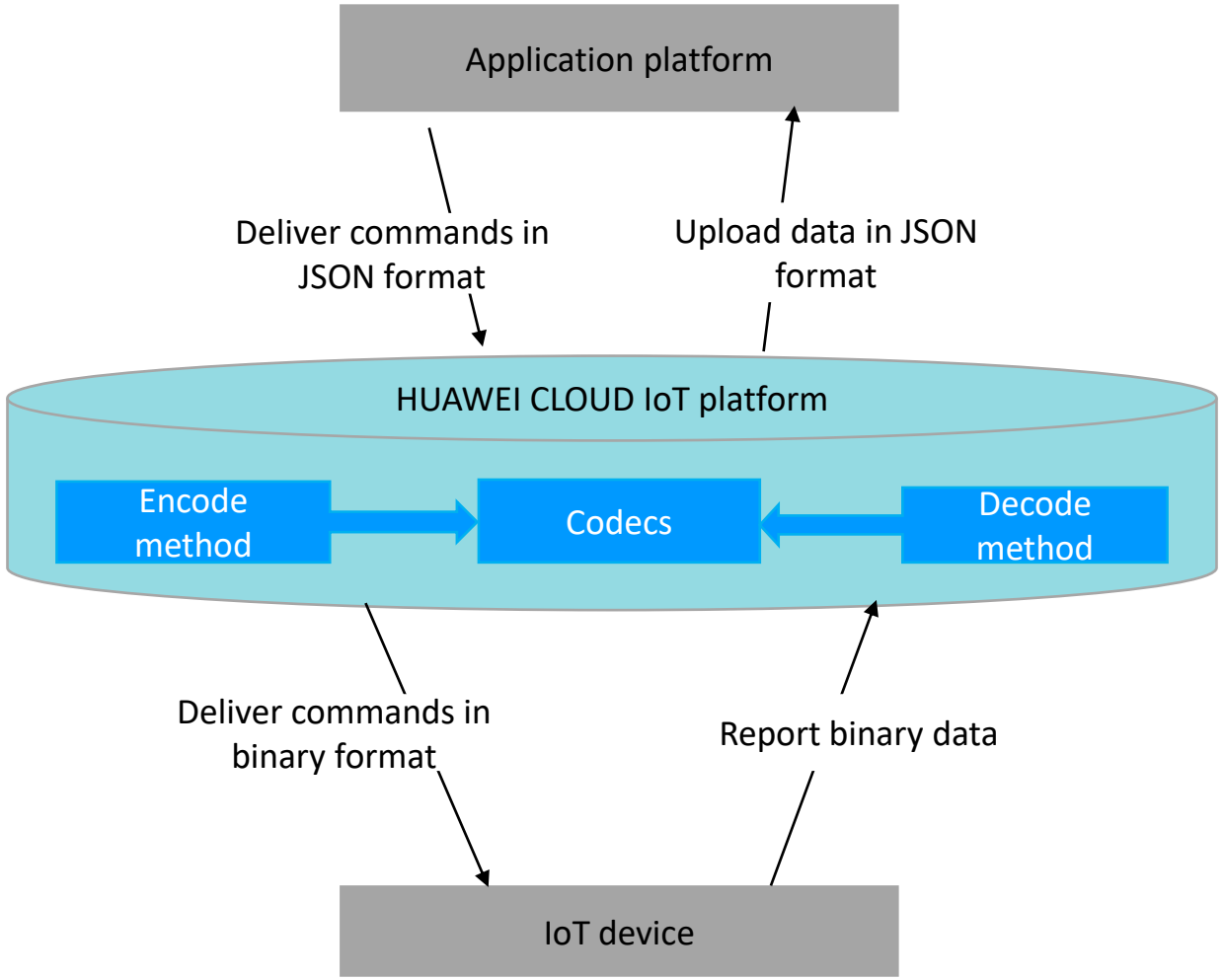
Codec (1)

- What Is a Codec?
  - The codec decodes binary data reported by devices into JSON data that can be read by the application and encodes downstream command data in JSON format of the application into binary data that can be executed by devices.
- Why Is the Codec Used?
  - NB-IoT devices use data in binary or TLV format.
    - CoAP is used for communication between NB-IoT devices and the IoT platform. The payload of CoAP messages carries data at the application layer, at which the data type is defined by the devices. Because NB-IoT devices have high requirements on power consumption, their application layer data is not in JSON format.
  - The application does not understand data in binary or TLV format.

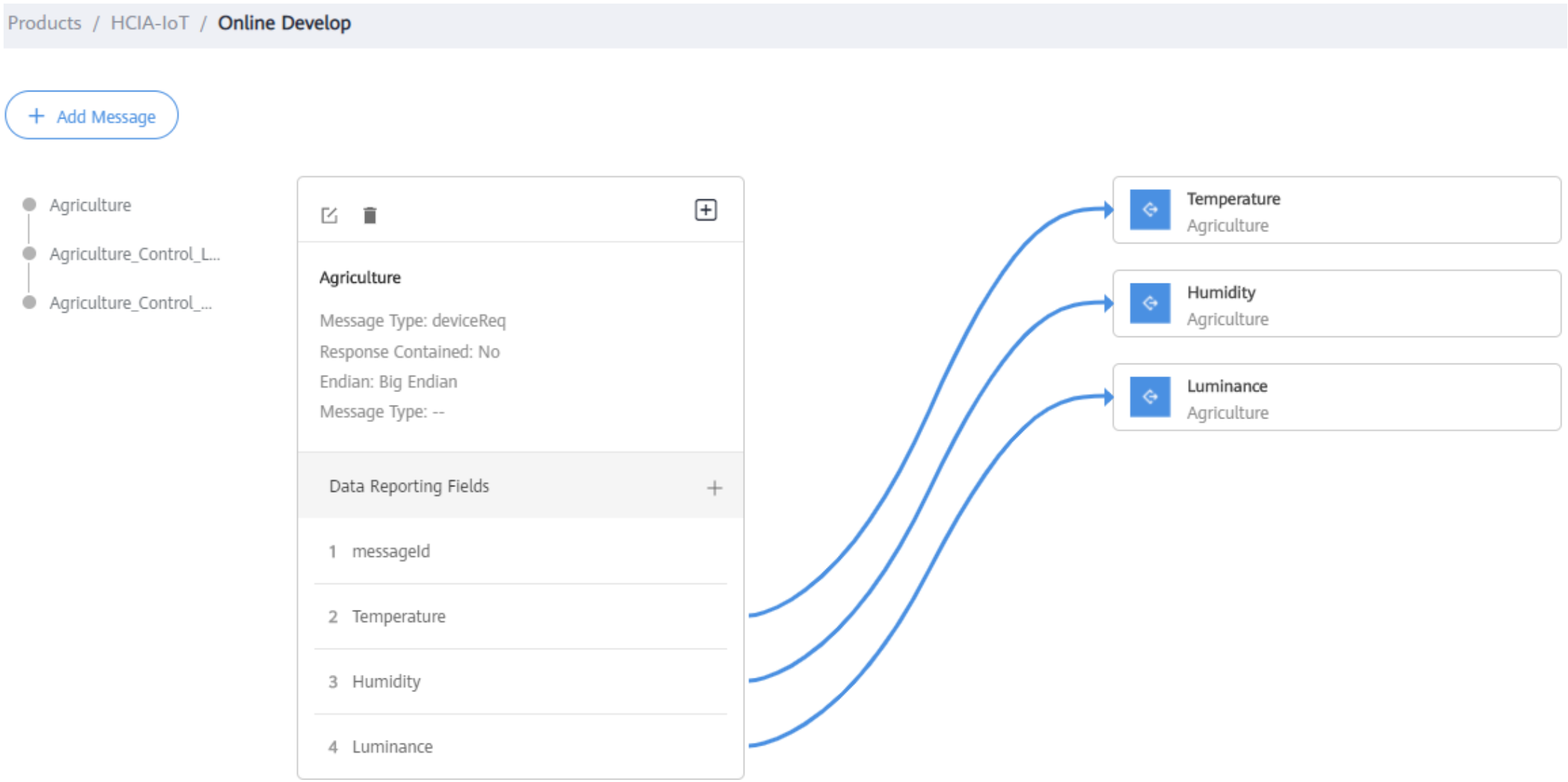
Codec (2)

- Developing a Codec
  - The platform provides three methods for developing codecs. Offline codec development is complex and time-consuming. **Graphical codec development** is recommended.
    - **Graphical development**: The codec of a product can be quickly developed in a visualized manner on the IoTDA console.
    - **Offline Development**: A codec is developed through secondary development based on the Java codec demo to implement encoding, decoding, packaging, and quality inspection.
    - **Script-based development**: JavaScript scripts are used to implement encoding and decoding.
  - The IoT platform abstracts and encapsulates the original codec development code. Therefore, developers can develop codecs simply by defining the format of code streams reported by devices and mapping the properties in the code streams and the profiles in a graphical way. When the development is complete, the codec is automatically generated and can be deployed on the IoT platform.

Codec (3)



Codec Example

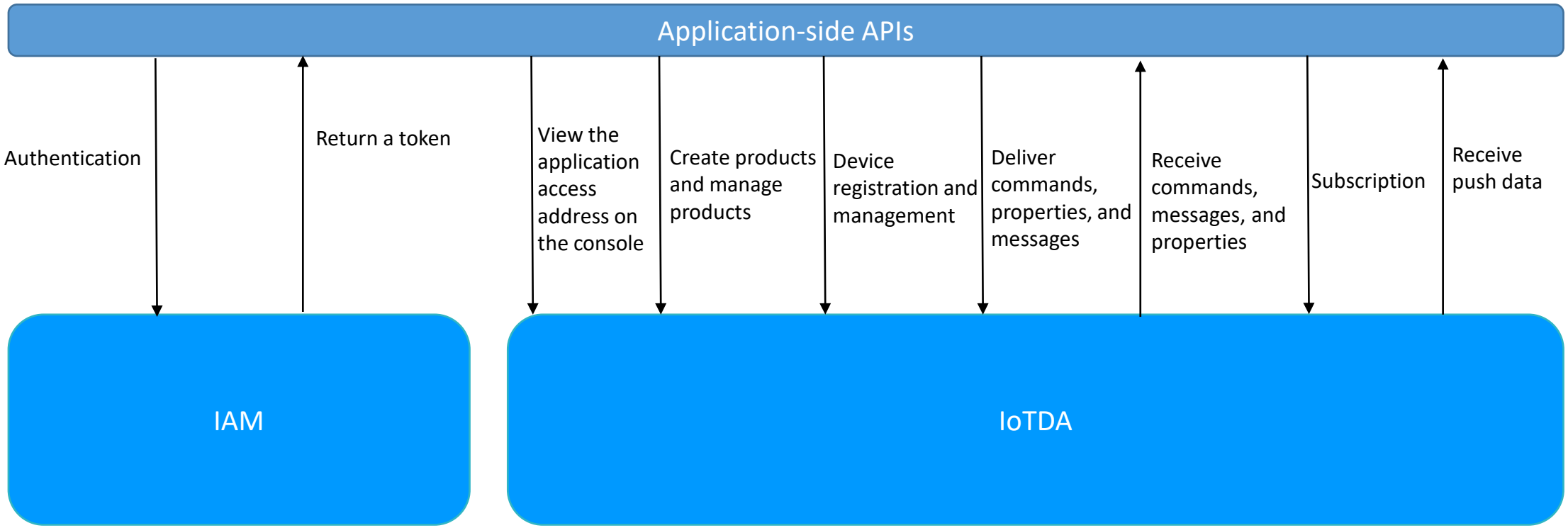


HUAWEI

Contents

Development on the Application Side

- The IoT platform provides APIs to make application development more easy and efficient. You can call these open APIs to quickly integrate platform functions, such as product, device, subscription, and rule management, as well as device command delivery.



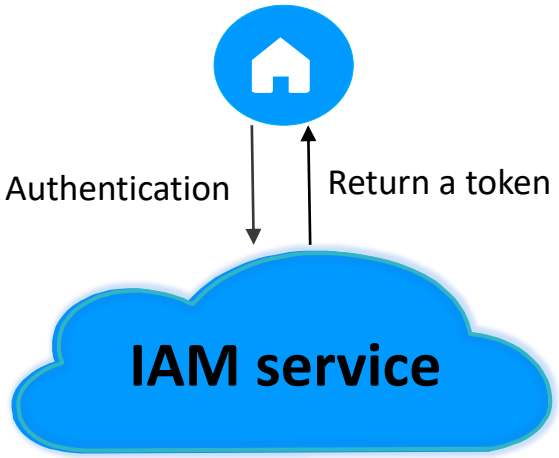
Northbound APIs of the IoT Platform

- The HUAWEI CLOUD IoT platform provides various northbound RESTful APIs for application developers to quickly develop IoT applications based on the capabilities provided by the platform.
- Northbound APIs provided by the HUAWEI CLOUD IoT platform include:
  - Subscription management
  - Product management
  - Device management
  - Device messages
  - Device commands
  - Device properties
  - Device shadows
  - Device group management
  - Tag management
  - Resource space management
  - Batch task
  - Batch task file management
  - Device CA certificate management
  - Rule management

Action	Description
GET	Obtains resources from the server.
POST	Creates a resource from the server.
PUT	Updates resources on the server.
DELETE	Deletes resources from the server.

Parameter	Description
header	Parameter of the HTTP message header.
path	Parameter of the path part in the URL.
query	Parameter behind the question mark (?) in the URL.
body	Parameter of the HTTP message body.

Application Access Authentication



Method: POST

Request:
https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens

Content-Type: application/json *//Body:*{

```
  "auth": {
    "identity": {
      "methods": [
        "password" ],
      "password": {
        "user": {
          "name": "username",
          "password": "********",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    ...
  }
```

}// username indicates the IAM username, and password indicates the password for logging in to HUAWEI CLOUD.

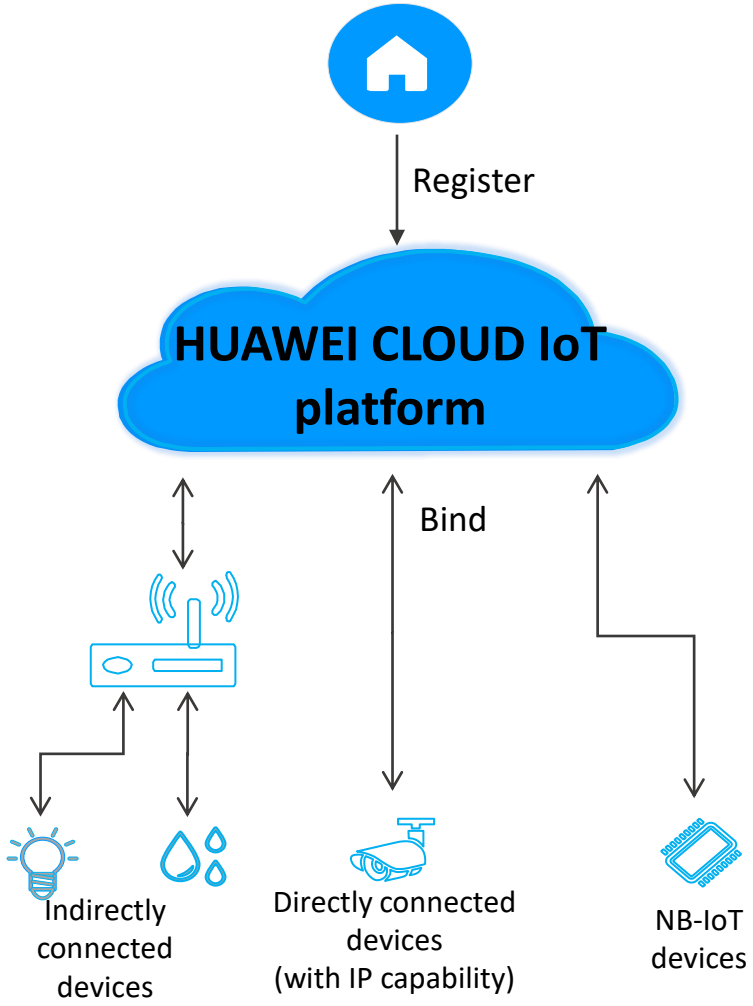
Response:
//Status Code:
Status Code: 201 Created
// Response header:

X-Auth-Token:MIIatAYJKoZIhvcNAQcCoIIapTCCGqECAQExDTALB...
//Body: {

```
  "token": {
    "catalog": [],
    "expires_at": "2020-01-04T09:05:22.701000Z",
    "issued_at": "2020-01-03T09:05:22.701000Z",
    "methods": [
      "password"
    ],
    "project": {
      "domain": {
        "id": "d78cbac186b744899480f25bd022f...",
        "name": "IAMDomain"
      },
...
```

} *//X-Auth-Token is the secret used for subsequent device and data operations.*

Creating a Device



Method: POST

Request:

https://{Endpoint}/v5/iot/{project_id}/devices
Content-Type: application/json
X-Auth-Token: ******
Instance-Id: ******

```
{
 "device_id" : "d4922d8a-6c8e-4396-852c-
164aefa6638f",
 "node_id" : "ABC123456789",
 "device_name" : "dianadevice",
 "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
 "auth_info" : {
  "auth_type" : "SECRET",
  "secure_access" : true,
  "fingerprint" :
"dc0f1016f495157344ac5f1296335cff725ef22f",
  "secret" : "3b935a250c50dc2c6d481d048cefdc3c",
  "timeout" : 300
 },
 ...
}
```

Response:

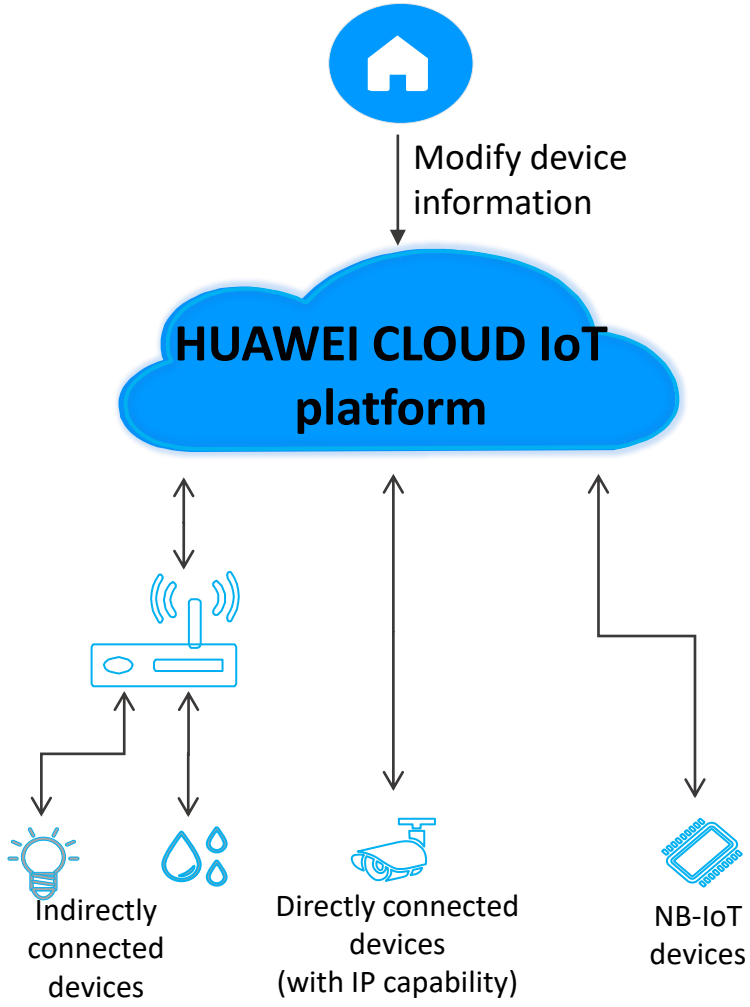
// Status Code:
Status Code: 201 Created

Content-Type: application/json
// Body:

```
{
 "app_id" : "***",
 "app_name" : "***",
 "device_id" : "***",
 "node_id" : "***",
 "gateway_id" : "***",
 "device_name" : "***",
 "node_type" : "***",
 "description" : "***",
 "fw_version" : "1.1.0",
 "sw_version" : "1.1.0",
 "auth_info" : {
  "auth_type" : "SECRET",
  "secret" : "***",
  "fingerprint" : "***",
  "secure_access" : true,
  "timeout" : 300
 },
 ...
 }
}
```


HUAWEI

Modifying Device Information



Method: PUT

Request:

https://{Endpoint}/v5/iot/{project_id}/devices/{device_id}
Content-Type: application/json
X-Auth-Token: ********
Instance-Id: ********

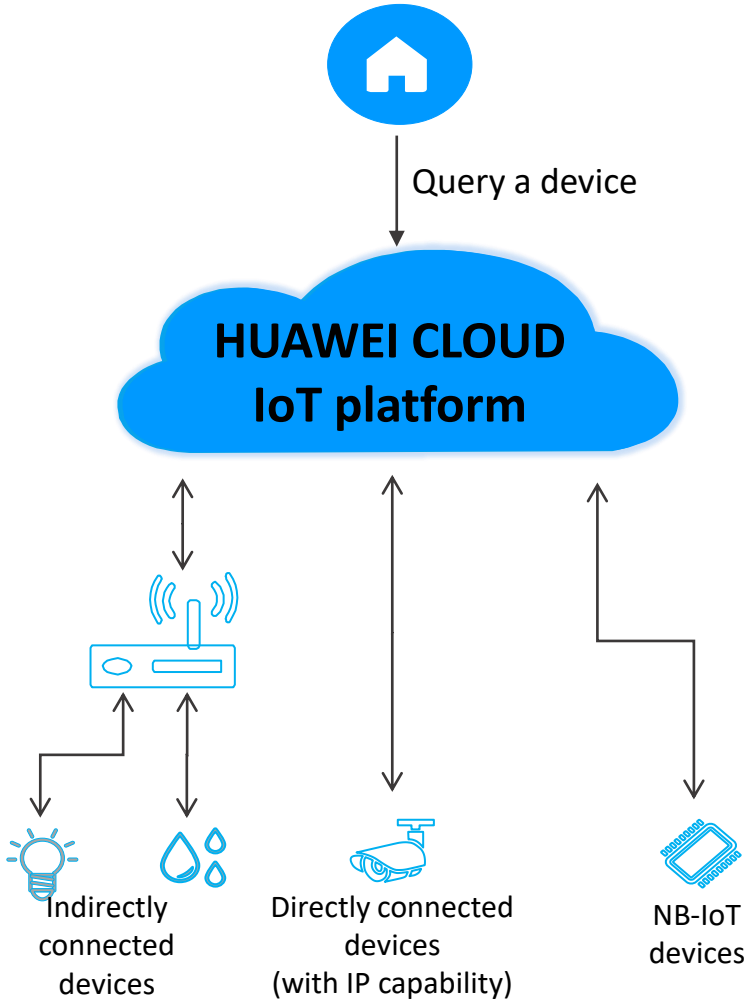
```
{
 "device_name" : "dianadevice",
 "description" : "watermeter device",
 "extension_info" : {
  "aaa" : "xxx",
  "bbb" : 0
 },
 "auth_info" : {
  "secure_access" : true,
  "timeout" : 300
 }
}
```

Response:

// Status Code:
Status Code: 200 OK

HUAWEI

Querying a Device



Method: GET

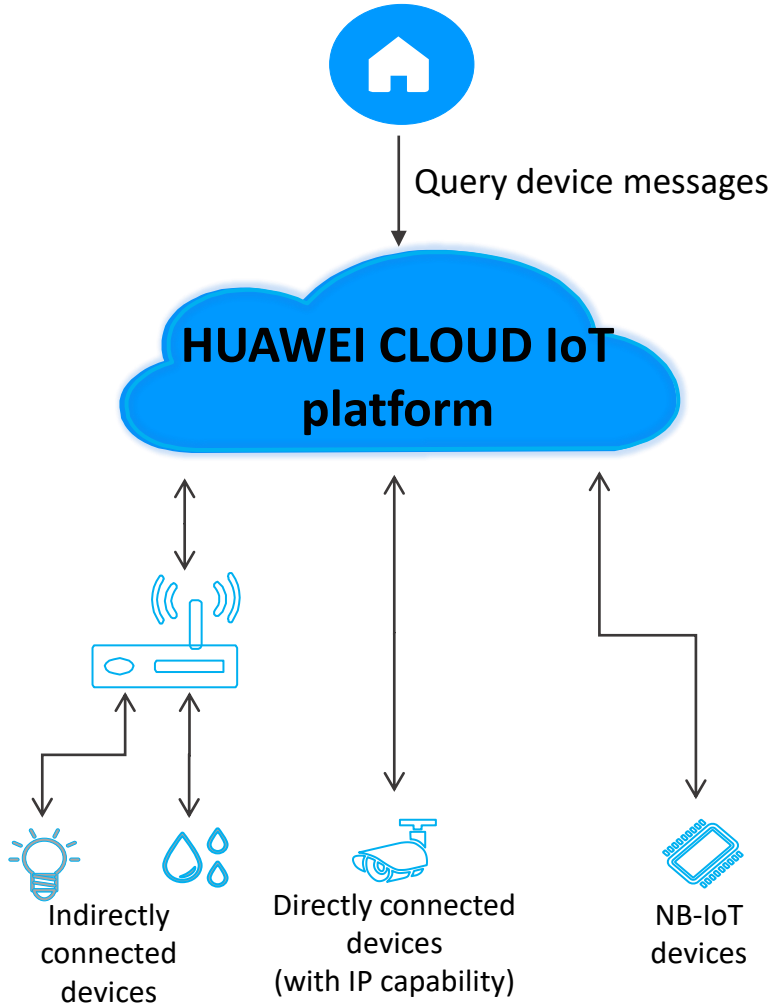
Request:
https://{Endpoint}/v5/iot/{project_id}/devices/{device_id}
Content-Type: application/json
X-Auth-Token: ********
Instance-Id: ********

Response:
// Status Code:
Status Code: 200 OK
Content-Type: application/json
//Body:

```
{
 "app_id" : "***",
  "app_name" : "***",
  "device_id" : "***",
  "node_id" : "***",
  "gateway_id" : "***",
  "device_name" : "***",
  "node_type" : "***",
  "description" : "***",
  "fw_version" : "1.1.0",
  "sw_version" : "1.1.0",
  "auth_info" : {
    "auth_type" : "***",
    "secret" : "***",
    "fingerprint" : "***",
    "secure_access" : true,
    "timeout" : 300
  },
...}
```


HUAWEI

Querying Device Messages



Method: GET

Request:

https://{Endpoint}/v5/iot/{project_id}/devices/{device_id}/messages
Content-Type: application/json
X-Auth-Token: ********
Instance-Id: ********

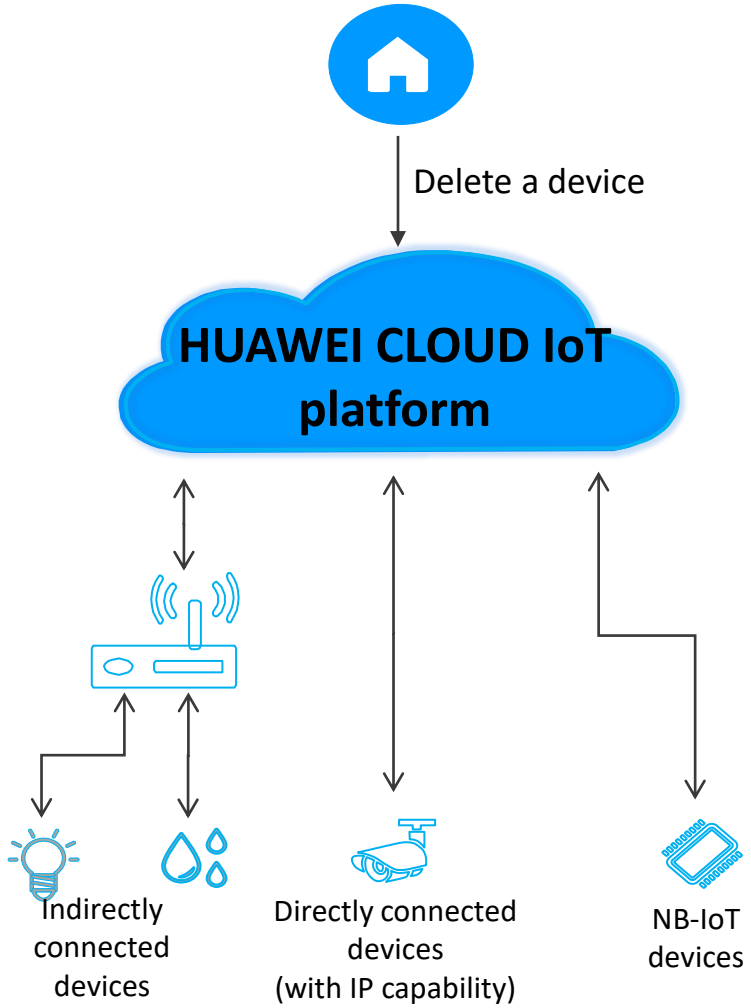
Response:

// Status Code:
Status Code: 200 OK
Content-Type: application/json


```
{
 "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
 "messages" : [ {
  "message_id" : "b1224afb-e9f0-4916-8220-
b6bab568e888",
  "name" : "message_name",
  "message" : "string",
  "topic" : "string",
  "status" : "PENDING",
  "created_time" : "20151212T121212Z",
  "finished_time" : "20151212T121212Z"
 } ]
}
```


HUAWEI

Deleting a Device



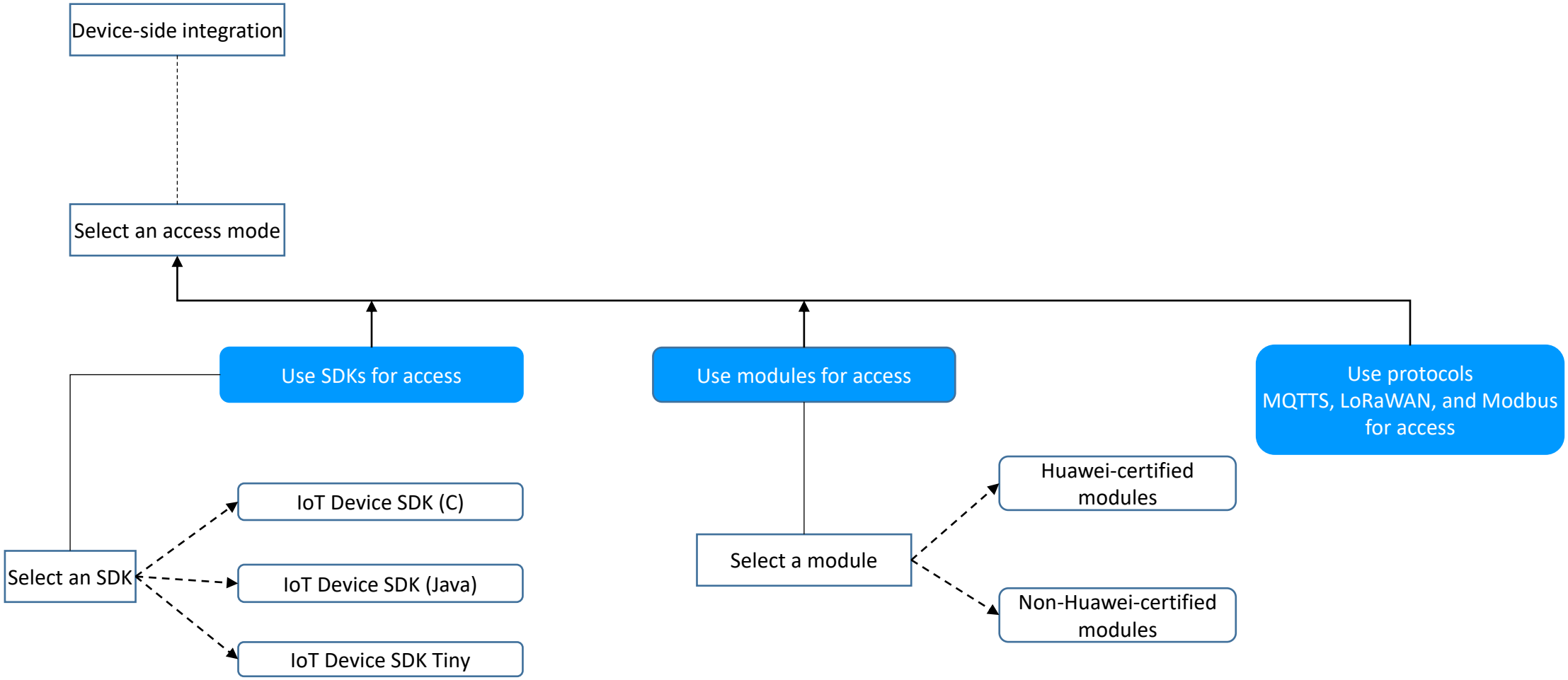
Method: DELETE

Request:
https://{Endpoint}/v5/iot/{project_id}/devices/{device_id}
Content-Type: application/json
X-Auth-Token: ********
Instance-Id: ********

Response:
Status Code: 204 No Content

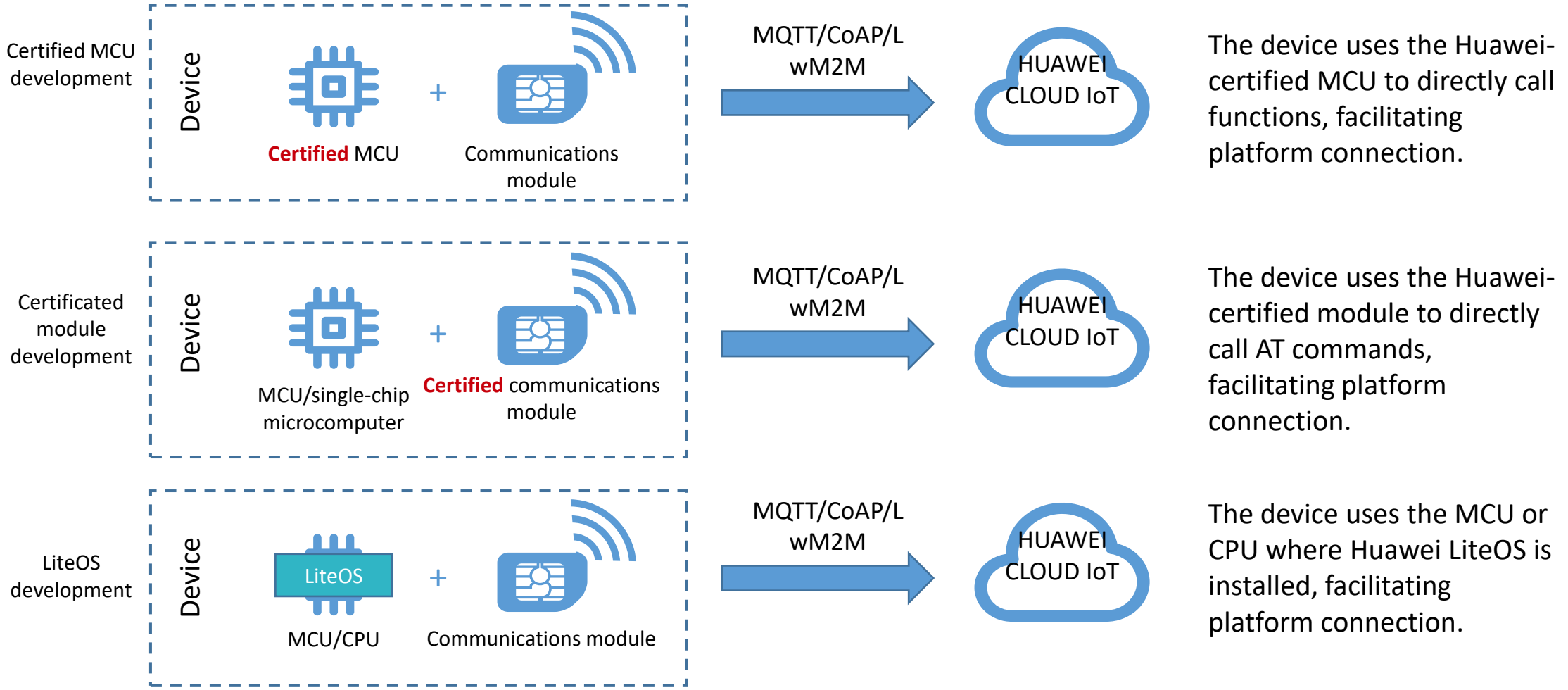
Contents

Platform Development Process: Development on the Device Side

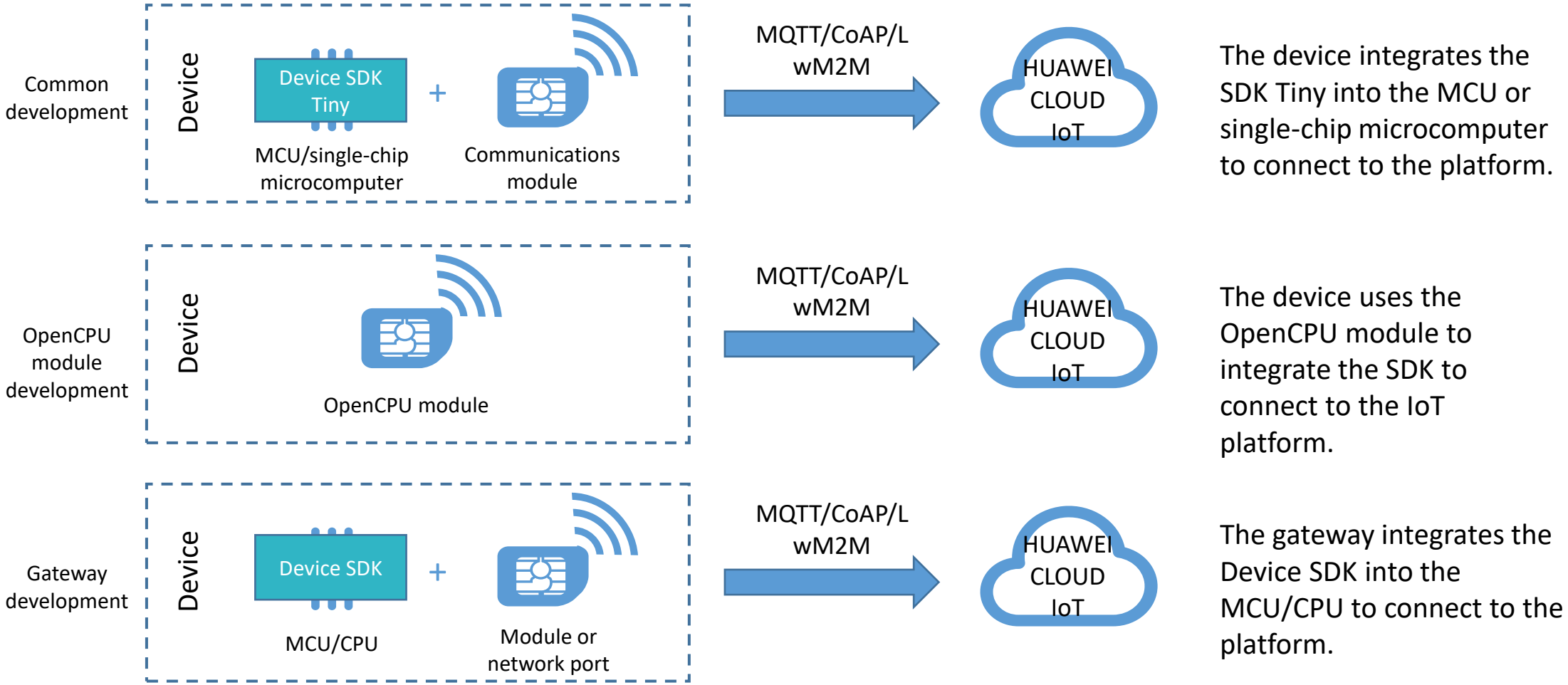


HUAWEI

Development on the Device Side (1)



Development on the Device Side (2)



HUAWEI

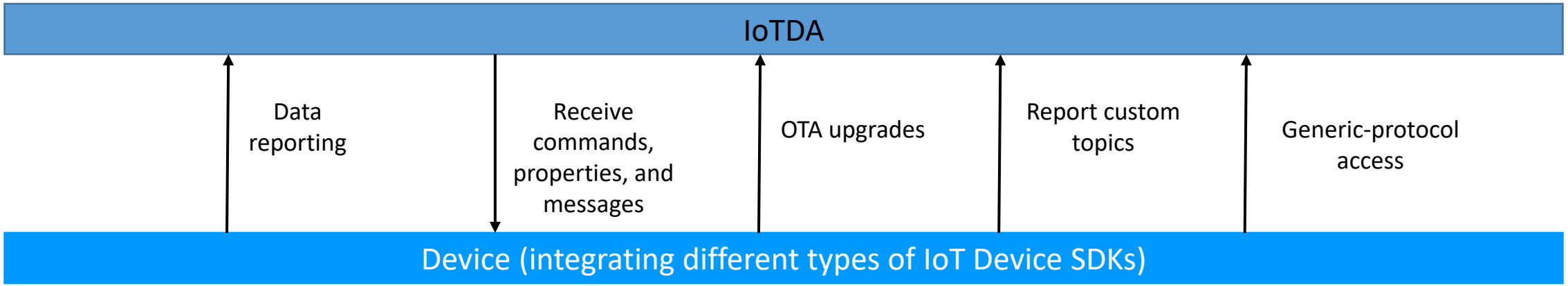
Using IoT Device SDKs for Access

- You can use Huawei IoT Device SDKs to quickly connect devices to the IoT platform. After being integrated with an IoT Device SDK, devices that support the TCP/IP protocol stack can communicate directly with the IoT platform. Devices that do not support the TCP/IP protocol stack, such as Bluetooth and Zigbee devices, need to use a gateway integrated with the IoT Device SDK to communicate with the platform.

1. Create a product on the IoTDA console or by calling the API **Creating a Product**.

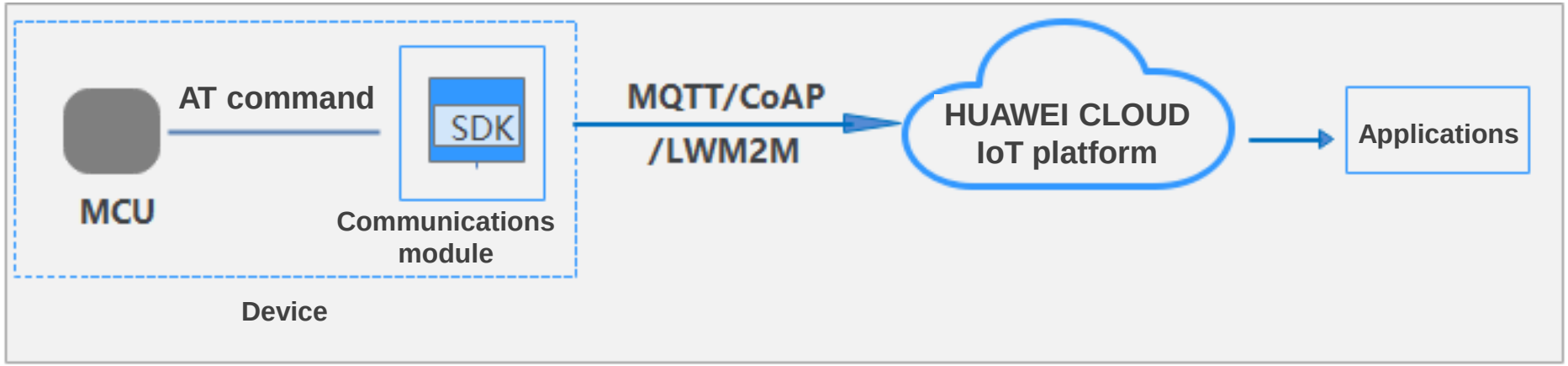
2. Register the device on the IoTDA console or by calling the API **Creating a Device**.

3. Implement the functions demonstrated in the figure, including reporting messages/properties, receiving commands/properties/messages, OTA upgrades, topic customization, and generic-protocol access.



Using Huawei - Certified Modules for Access

- Certified modules are pre-integrated with the IoT Device SDK Tiny. They have passed Huawei tests, and comply with Huawei's AT command specifications. The following benefits are available for Huawei-certified modules:
  - Device manufacturers do not need to worry about how to connect to the HUAWEI CLOUD IoT platform on the MCU (for example, how to set the secret encryption algorithm and clientID composition mode during MQTT connection setup). To connect their devices to the platform, they only need to invoke AT commands. This accelerates device interconnection and commissioning.
  - The MCU does not need to integrate the MQTT protocol stack or IoT Device SDK Tiny, greatly reducing MCU resource consumption.
  - Huawei releases certified modules on HUAWEI CLOUD Marketplace so that device manufacturers and service providers can purchase these certified modules to quickly connect to HUAWEI CLOUD IoT.
- The following figure shows how a certified module is used to connect a device to the platform.

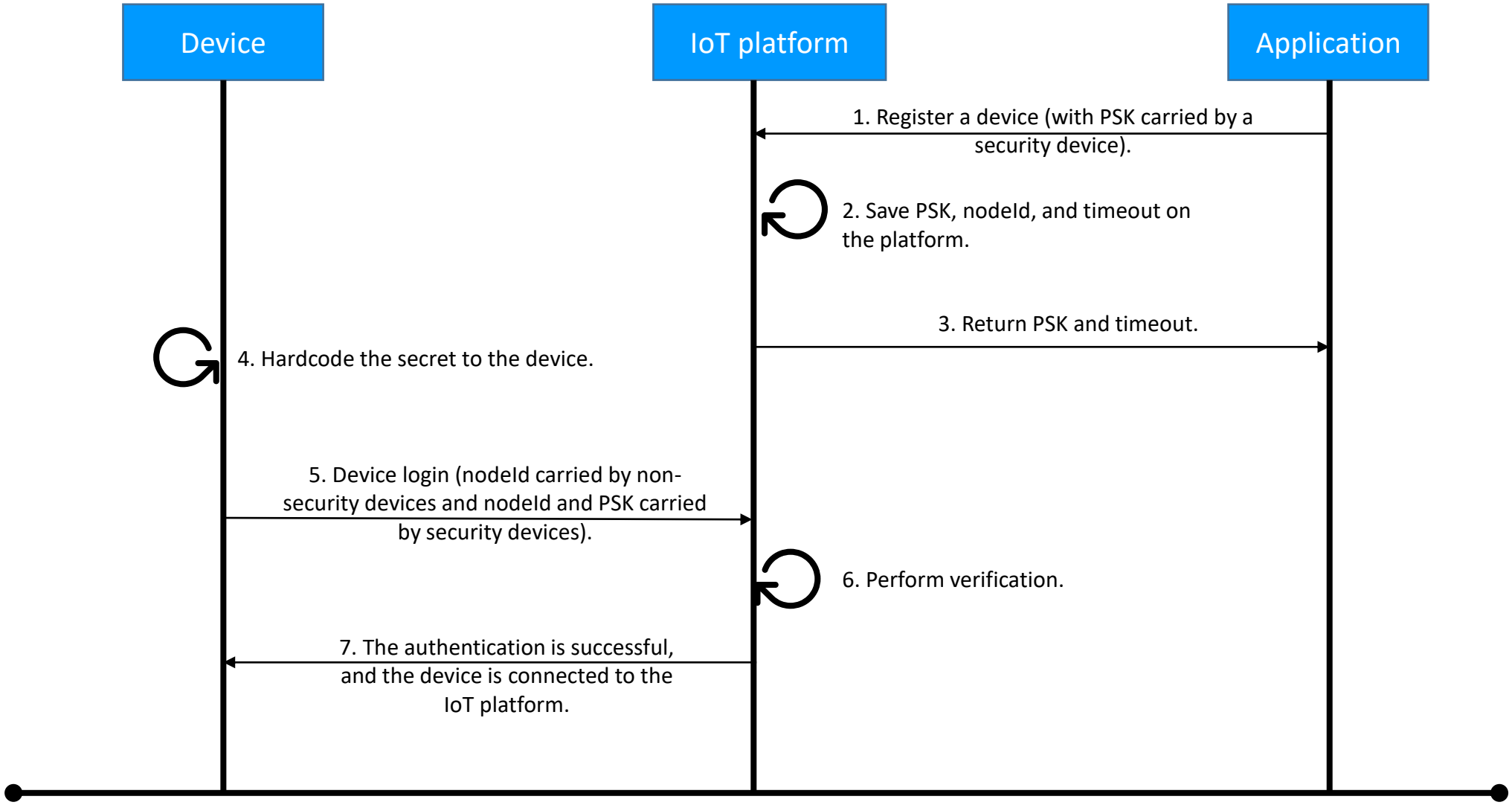


Device Authentication

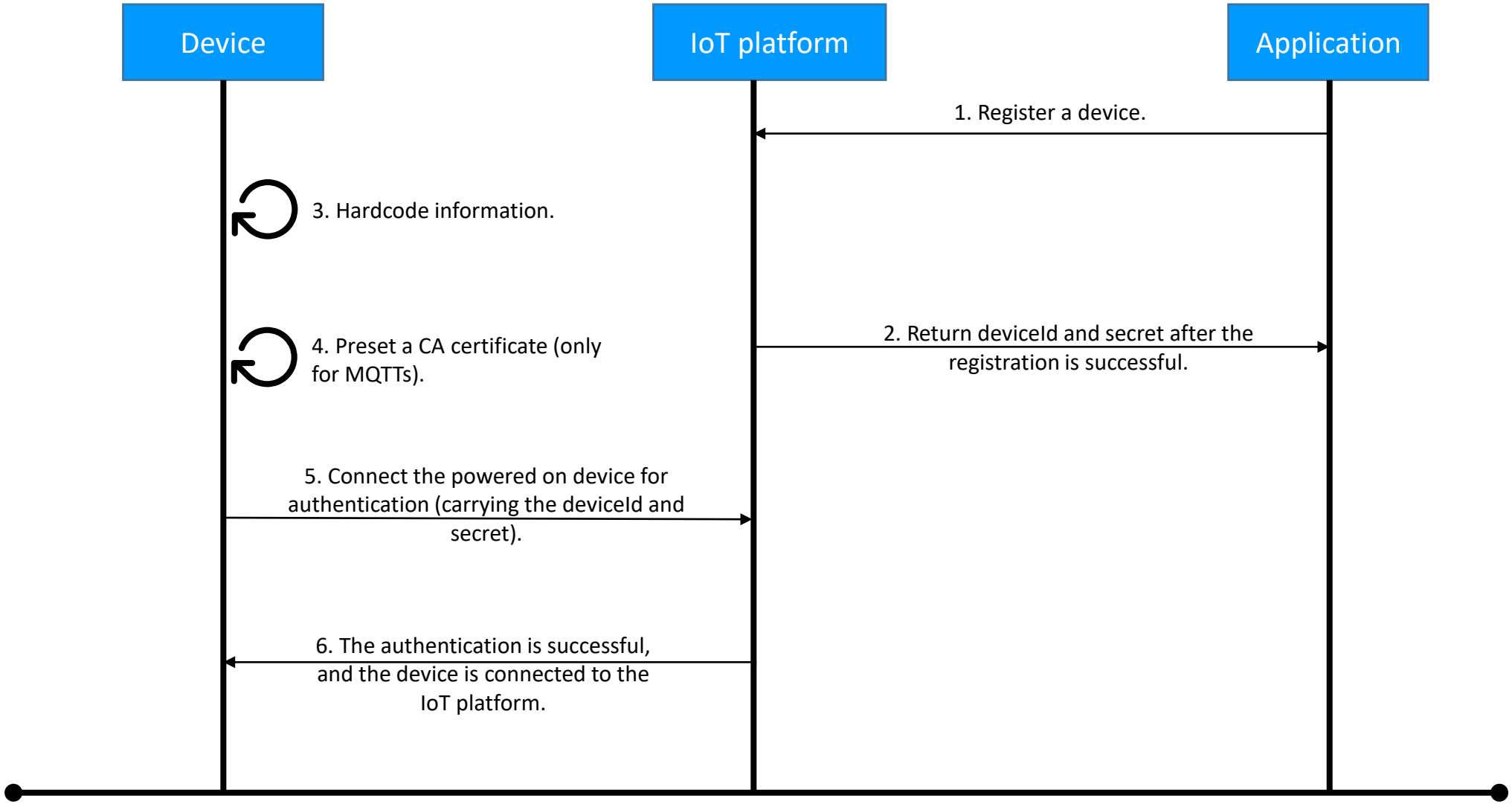
- The platform authenticates a device when the device attempts to access the platform. The authentication process depends on the access method.

Access Type	
Device connected using LWM2M over CoAP	Call the API **Creating a Device** or use the IoTDA console to register a device with the platform, and set the node ID (for example, the IMEI) as the verification code. The device can use the node ID to get authenticated and connect to the platform. When Datagram Transport Layer Security (DTLS) or DTLS+ is used, the transmission channel between the device and platform is encrypted by using a PSK.
Device using native MQTT or MQTTS	Call the API **Creating a Device** or use the IoTDA console to register a device with the platform, and hardcode the device ID and secret returned by the platform into the device. A CA certificate is preset on MQTTS devices, but not MQTT devices. The device uses the device ID and secret to get authenticated and connect to the platform.

Authentication for Devices Using LwM2M over CoAP

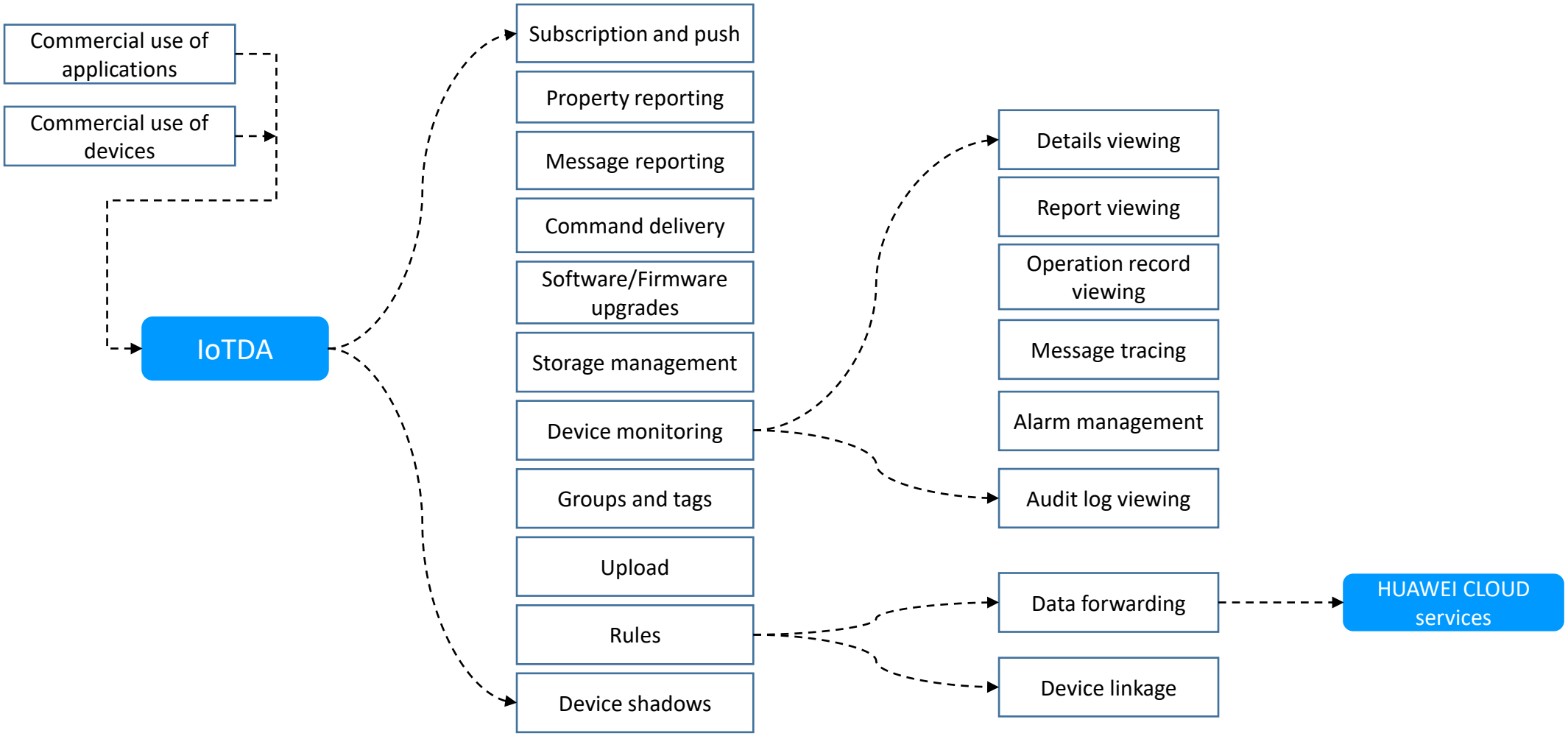


Authentication for Devices Using Native MQTT or MQTTS

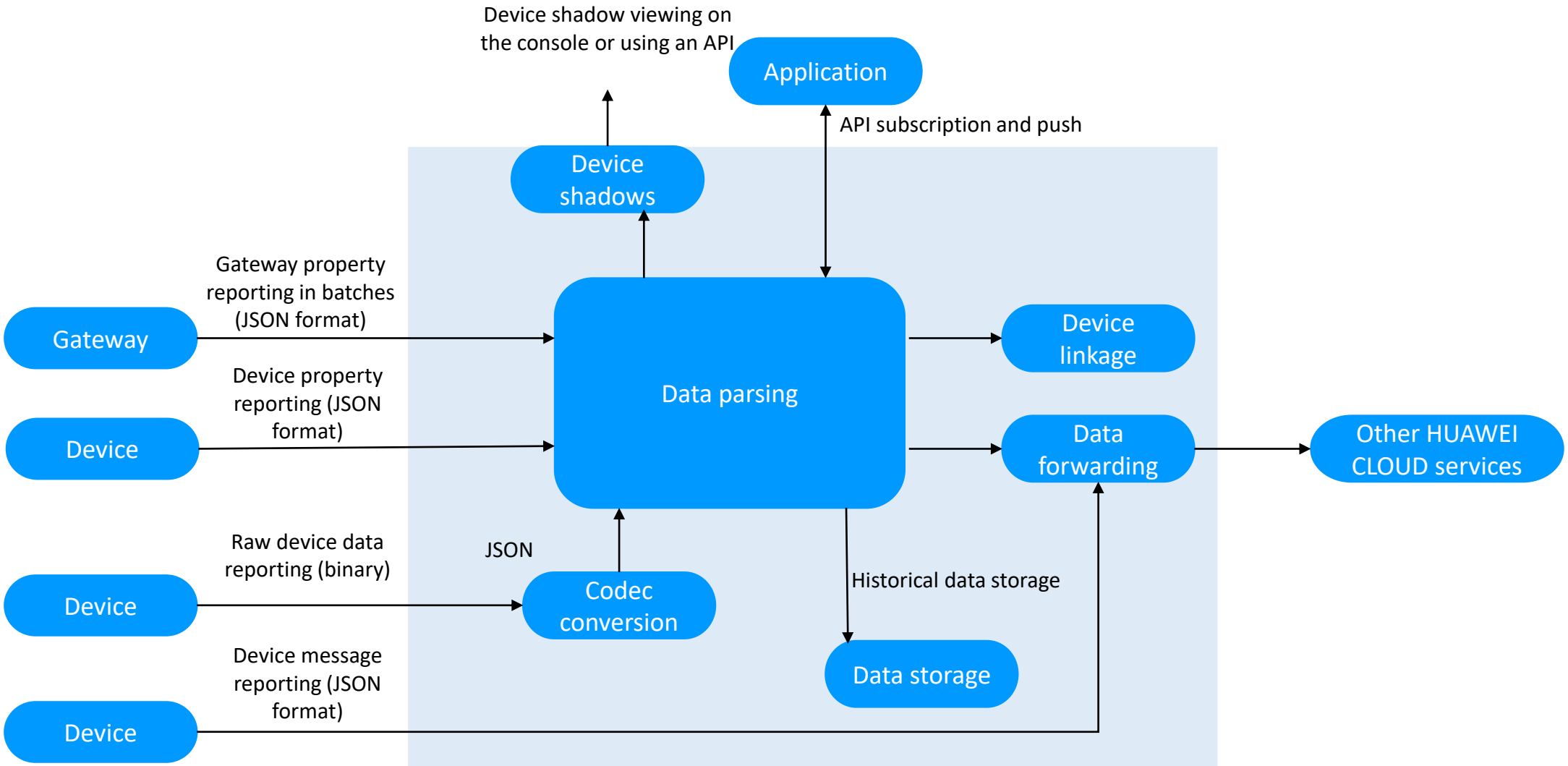


Contents

Platform Development Process: Cloud-based Routine Management

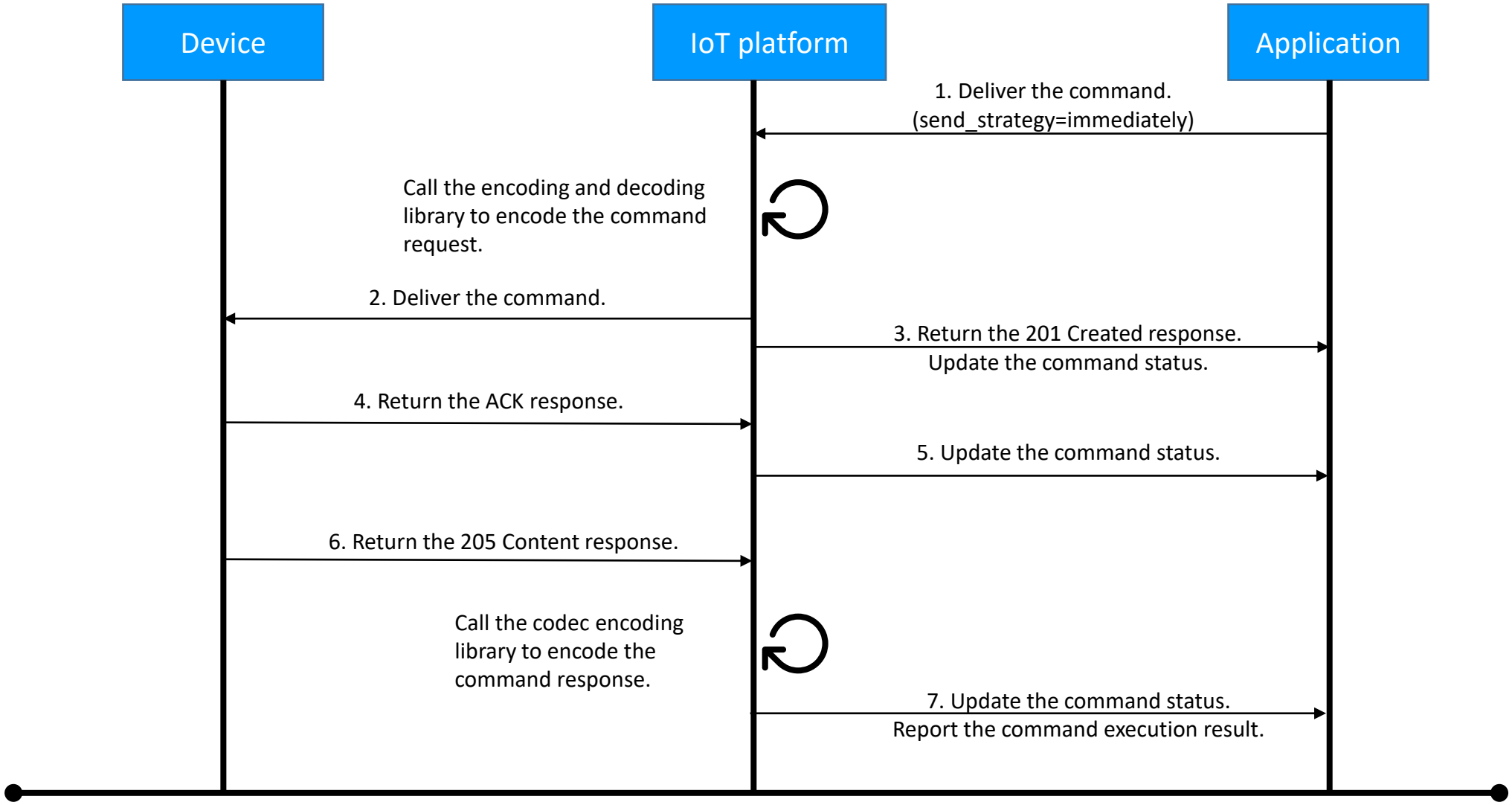


Data Reporting

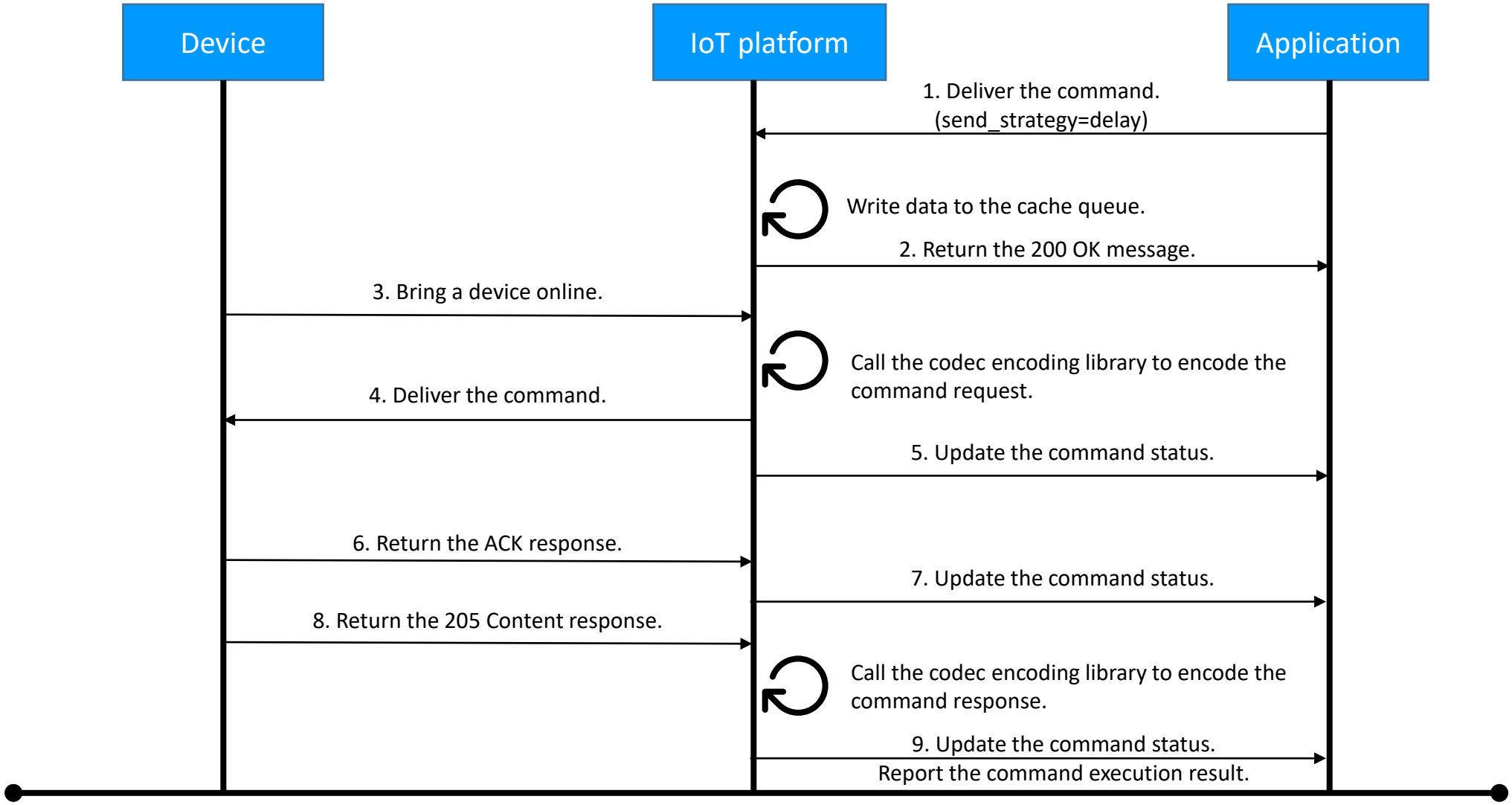


HUAWEI

Immediate Delivery of LwM2M/CoAP Device Commands

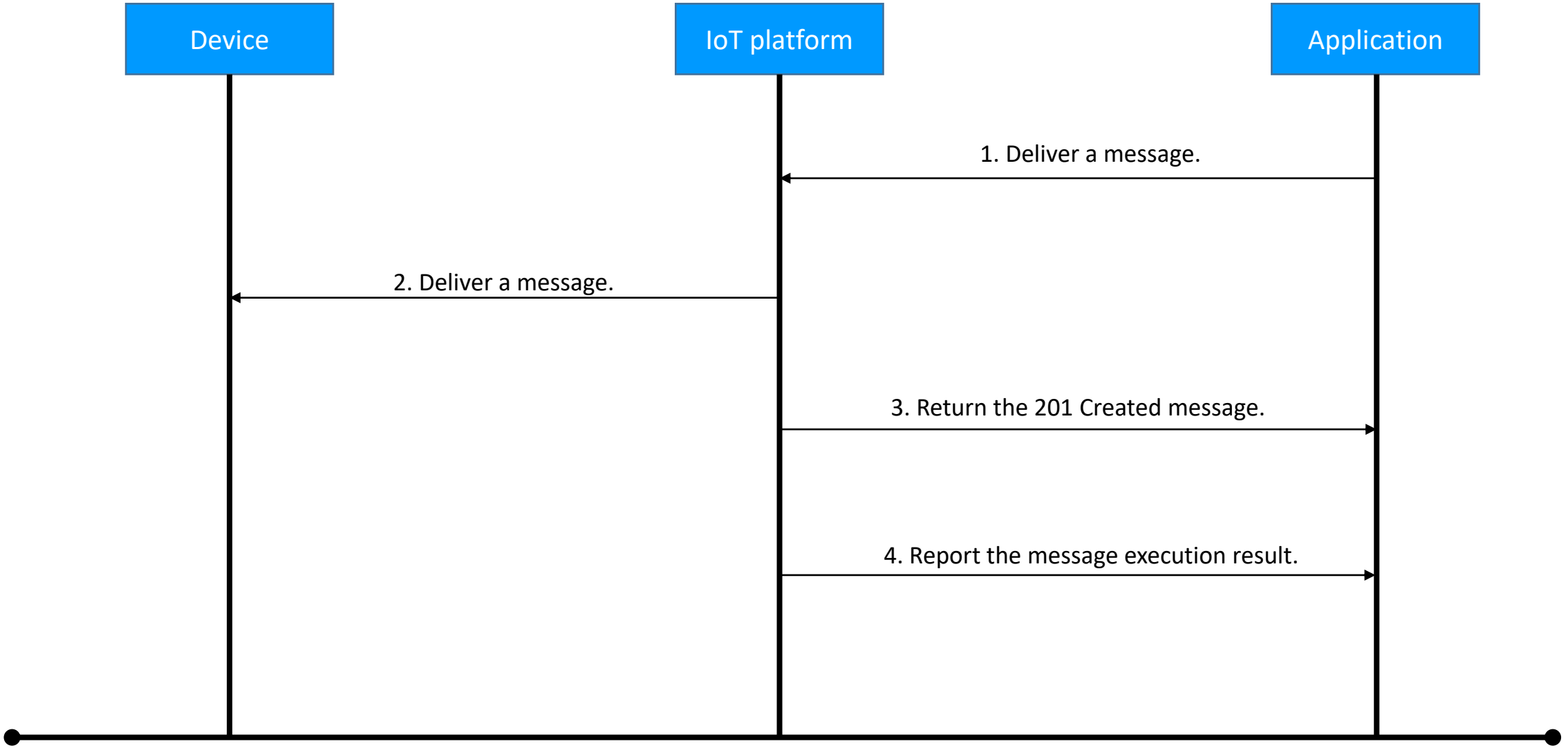


Delayed Delivery of LwM2M/CoAP Device Commands

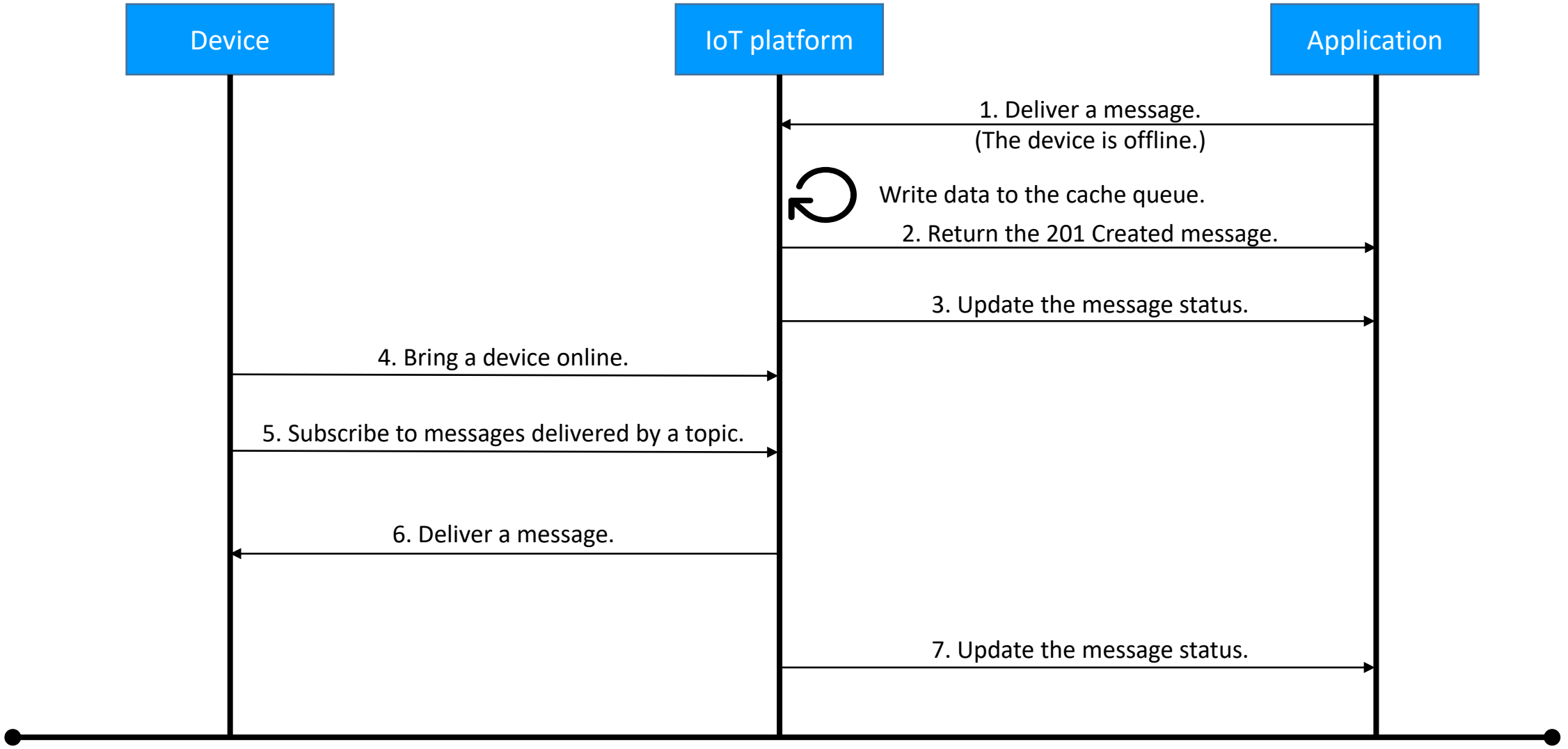


HUAWEI

Immediate Delivery of MQTT Device Messages



Delayed Delivery of MQTT Device Messages



Quiz

1. (Multiple Choice) Which of the following two formats are used by the IoT platform codec to convert data?

   A. Binary data

   B. Decimal data

   C. JSON data

   D. XML data

2. (True or false) An IoT application must be authenticated before being connected to the IoT platform.

Summary

- In this section, you learned how to perform secondary development on the IoT platform. Secondary development is classified into product development, development on the device side, development on the application side, and cloud-side routine management. Development on the product side includes product model development and codec development.

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

HUAWEI