# IoT Device Access

# Service Overview

**Issue** 1.0

**Date** 2024-12-31

**HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.**

## Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

## Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

# Contents



# 1 Overview

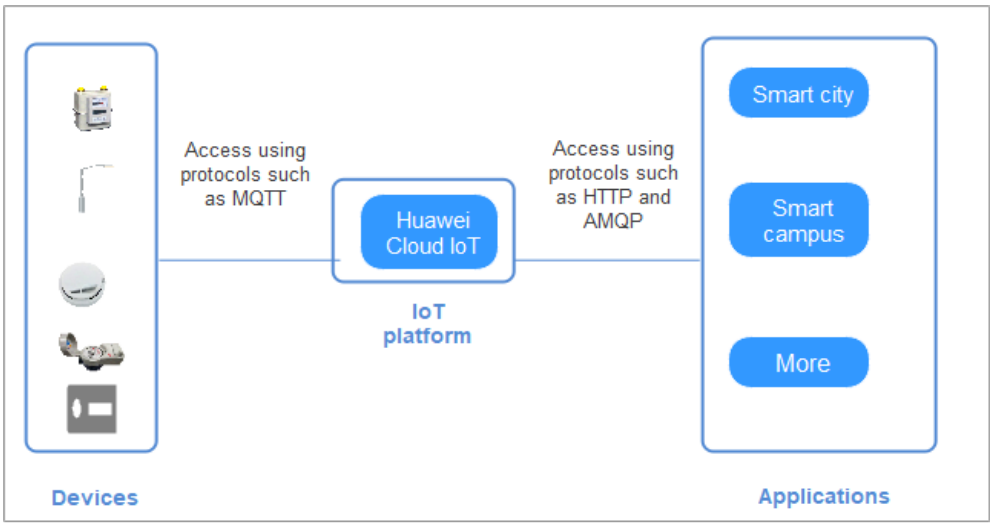
Huawei Cloud IoT Device Access (IoTDA) allows you to connect your physical devices to the cloud, where you can collect device data and deliver commands to devices for remote control. It can also work with other Huawei Cloud services to help you quickly develop IoT solutions.

A complete IoT solution consists of the IoT platform, devices, and applications.

- The IoT platform is located between applications and devices. It harmonizes differences between device interfaces to enable quick device access. It provides robust capabilities to help you develop diverse IoT solutions.
- Devices can access the platform via fixed broadband (FBB), 2G/3G/4G/5G, Narrowband Internet of Things (NB-IoT), and Wi-Fi networks. They can report service data to the platform using mainstream protocols or industry protocols, such as Message Queuing Telemetry Transport (MQTT), Lightweight Machine-to-Machine (LwM2M) over Constrained Application Protocol (CoAP), and Hypertext Transfer Protocol Secure (HTTPS). Devices can also receive commands from the platform.
- Applications call application programming interfaces (APIs) provided by the platform to implement services such as data collection, command delivery, and device management.

**Figure 1-1** IoT solutions




## Features

Devices can connect to the platform directly or through industrial gateways or home gateways. The platform supports multi-network access, multi-protocol access, and serialized Agent access, preventing issues caused by complex, diversified, and fragmented device access. It also provides comprehensive device management capabilities, simplifies management of device fleets, and improves management efficiency. The following table describes the IoTDA functions.

**Table 1-1** IoTDA functions

| Category | Function | Description |
| --- | --- | --- |
| Device access | Native protocol access | You can connect devices to the platform using MQTT, CoAP, LwM2M, and HTTPS protocols. |
| | Serial device SDKs | IoT Device SDK and IoT Device SDK Tiny in languages such as C and Java are supported. For details, see **Introduction to IoT Device SDKs**. |
| | Industry protocol access | You can connect devices to the platform through edge gateways using Modbus and OPC Unified Architecture (OPC UA) and using industry protocols based on plug-ins. |
| | Device access authentication | Authentication modes such as one-device-one-secret and X.509 certificates are supported. |
| | - | - |

| Category | Function | Description |
| --- | --- | --- |
| Device management | Device lifecycle management | You can add, delete, modify, and query devices, manage device status, freeze and unfreeze devices, and manage child devices. |
| | Groups and tags | You can group devices or add tags to them. For details, see **Groups**. |
| | Product models | You can define a product model (profile) for devices. For details, see **Product Model Definition**. |
| | Device shadows | You can configure and query device shadows. For details, see **Device Shadow**. |
| | OTA upgrade | You can upgrade device software and firmware. For details, see **OTA Upgrade**. |
| | Device file upload | Devices can upload files to OBS and request files from the cloud. For details, see **File Uploads**. |
| | Batch device operations | You can perform batch operations on devices, including device registration, software/firmware upgrades, and command delivery. For details, see **Create a Batch Task**. |
| Message communications | Bidirectional transparent transmission | Device messages can be pushed to applications using HTTP and AMQP. Applications can deliver messages to devices asynchronously. |
| | Product model topic communications | Applications and devices communicate with each other based on the properties, commands, and events defined in the product model in a decoupled mode. |
| | Custom topic communications | You can customize topics for bidirectional message communications. |
| | Data parsing and conversion | You can develop codecs online to parse and convert device data. |
| | Command delivery | Commands can be delivered to online devices in synchronous mode. In the NB-IoT scenario, commands can be delivered in asynchronous mode. For details, see **Command Delivery**. |

| Category | Function | Description |
| --- | --- | --- |
| Rules | Data forwarding | Data can be forwarded to Huawei Cloud Kafka, Object Storage Service (OBS), GaussDB, Data Ingestion Service (DIS), Distributed Message Service (DMS), and ROMA Connect. For details, see **Rules**. |
| | Device linkage | You can create rules for device linkage control. For details, see **Rules**. |
| | Data forwarding | The platform can forward data reported by devices to applications through HTTP or AMQP. |
| Monitoring and O&M | Logging | The console provides message tracing, integrates with Log Tank Service (LTS) for log analysis, and integrates with Cloud Trace Service (CTS) for log audit. For details, see **Monitoring and O&M**. |
| | Alarm reporting | The platform provides notifications and management of system alarms (such as threshold alarms) and alarms triggered by device rules by integrating with Application Operations Management (AOM). For details, see **Alarms**. |
| | Metric monitoring | The platform provides monitoring reports of tenant-level service metrics (such as device status, commands, subscription and push, and message transfer) by integrating with AOM. For details, see **Reports**. |

## Security and Data Protection

IoTDA established an end-to-end trustworthy security system. It is graded level-4 of China's Multi-Level Protection Scheme 2.0 and obtains international security certifications such as ISO27001, ISO27017, ISO27018, and CSA STAR. It complies with European Union's General Data Protection Regulations (GDPR).

- Device security: It provides a one-device-one-secret authentication mechanism to prevent unauthorized access and supports device security check.
- Data transmission: Secure transmission channels are provided based on Transport Layer Security (TLS), Datagram Transport Layer Security (DTLS), and DTLS+.
- Platform security: Threat defense is performed on the entire Huawei Cloud. Huawei Cloud security service products or components and security D&R department are fully leveraged to build a comprehensive security defense system that covers security analysis, design, coding, testing, and defense.
- Data protection: It complies with GDPR.

# 2 Advantages

With service development, an increasing number of enterprises choose to combine IoT technologies for business growth. Huawei Cloud IoT services have outstanding advantages in capabilities, costs, O&M, security, and ecosystem compared with MQTT clusters managed by enterprises.

**Table 2-1** Comparison

| Dimension | Item | Huawei Cloud IoT | MQTT Cluster Managed by Enterprises |
| --- | --- | --- | --- |
| Capabilities | Flexible protocols | • Supports mainstream IoT protocols and proprietary protocols to meet requirements of different devices and access scenarios.<br>• Provides the plug-in mechanism to parse custom protocols. | Supports only the MQTT protocol. The capability of supporting other protocols requires development. It is difficult, expensive, and inefficient to maintain multiple protocols. |
| | Access | • Provides series of multi-language, open-source IoT device SDKs.<br>• Pre-integrates SDKs in popular modules and chips for multi-network and multi-protocol access. This simplifies device access and shortens the access time to hours. | Developers are required to be familiar with different programming languages, causing heavy development workload. |

| Dimension | Item | Huawei Cloud IoT | MQTT Cluster Managed by Enterprises |
| --- | --- | --- | --- |
| | Performance stability | • Supports smooth and elastic expansion of service resources after purchase.<br>• Supports secure and stable connections of hundreds of millions of devices, reliable communications with 100,000 TPS concurrency, and devices going online concurrently with tens of thousands of TPS.<br>• Ensures 99.95% service availability. | R&D engineers need to perform tuning. To ensure 99.9% or higher availability, R&D engineers who are proficient in open-source MQTT and senior architecture personnel are required. |

| Dimension | Item | Huawei Cloud IoT | MQTT Cluster Managed by Enterprises |
| --- | --- | --- | --- |
| | Features | • Cell-based technologies control the fault scope.<br>• Supports message tracing for fault locating and cause analysis.<br>• Supports device shadows.<br>• Supports over-the-air (OTA) upgrades.<br>• Supports product models, which abstract and summarize product functions to decouple software and hardware development and improve system integration efficiency.<br>• Provides the plug-in mechanism to parse custom protocols.<br>• Supports data forwarding rules. Data can be forwarded to more than 10 cloud services.<br>• Supports device linkage rules. Rules can be customized based on time, conditions, and actions, to configure scenario linkage and implement automatic collaboration across applications, subsystems, and devices.<br>• Uses open architecture to leverage cutting-edge technologies and services of cloud computing.<br>• Extensive functions and solutions have served many customers in different industries. | Open-source MQTT provides basic functions. Developers need to develop a complete solution based on open-source capabilities. However, some open-source code left unmodified during intrusive modification by developers may cause accidents on the live network during open-source middleware upgrades. |
| | - | - | - |

| Dimension | Item | Huawei Cloud IoT | MQTT Cluster Managed by Enterprises |
|---|---|---|---|
| | Technical support | • Provides 24/7 professional support.<br>• The service ticket system responds within 10 minutes. | Open-source MQTT does not provide technical support and has a large number of default configuration parameters. Enterprises need to adjust the parameters based on service scenarios. If developers of enterprises are not familiar with the open-source code, their improper parameter adjustments pose huge risks to commercial systems. When problems occur, they have to solve problems by themselves. |
| Costs | Server cost | Servers do not need to be purchased. | Servers need to be purchased. |
| | Labor cost | No extra labor is required for cloud services. | Enterprises need to pay for professional development and O&M teams. |
| | Resource use | Resources are out-of-the-box and elastic for service growth and scale-out without interruption. | Enterprises need to develop the elastic resource scaling function by themselves. |
| | Architecture cost | The high-availability, high-performance, and secure architecture is built based on cloud native 2.0 and supports continuous evolution. | It is difficult for an enterprise to build the architecture that achieves high availability, high performance, and high security. |
| O&M | Infrastructure O&M | Provides unified O&M, quick response, scaling, upgrade, and troubleshooting based on professional teams. | Enterprises need to build their own O&M teams or use third-party O&M teams to solve scaling, upgrade, and O&M problems. Statistics show that most service faults are triggered by scaling and upgrade operations. The O&M cost is several times or even dozens of times the development cost. |

| Dimension | Item | Huawei Cloud IoT | MQTT Cluster Managed by Enterprises |
| --- | --- | --- | --- |
| | Service platform version | Provides unified update by public cloud service providers, fast version iteration. | It is managed by enterprises. |
| | O&M | • Provides full-link log analysis and message tracing.<br>• Provides real-time monitoring and sensing of device statuses.<br>• Supports custom service metric alarms. | It is managed by enterprises. |

| Dimension | Item | Huawei Cloud IoT | MQTT Cluster Managed by Enterprises |
| --- | --- | --- | --- |
| Security | System security | Establishes a trusted security system: It obtains international security certifications such as ISO27001, ISO27017, ISO27018, and CSA STAR. It complies with EU's GDPR.<br>• Transport network layer: Border security protection is provided based on web application firewall (WAF) and distributed denial of service (DDoS). Efficient, secure transmission protocols such as DTLS, TLS, HTTPS, CoAPS, and MQTTS are provided.<br>• Device side: A unique digital certificate is provisioned for each device for secure access. LiteOS-based security capabilities are provided.<br>• Platform side: Threat analysis is performed on entire Huawei Cloud network. Huawei Cloud security service products and public security services or components are fully reused to build a security defense system. | It is managed by enterprises. End-to-end security is a systematic project with high requirements. It is expensive and difficult to build and maintain system-level security capabilities. |
| | Data security | Provides a complete security protection system. Data is stored in redundant mode in the data center of the cloud service provider, ensuing data security. | Enterprises need to build data redundancy, backup, and recovery capabilities. |

| Dimension | Item | Huawei Cloud IoT | MQTT Cluster Managed by Enterprises |
| --- | --- | --- | --- |
| | Disaster recovery | Supports active-active service deployment, multi-data center DR, as well as high availability and DR capabilities based on multiple regions and availability zones (AZs). | Self-managed clusters usually do not have DR capabilities. Huge investment in active-active service deployment and DR devices often ends up with low return on investment (ROI). |
| | Vulnerability fixing | Establishes a comprehensive vulnerability management system and a dedicated security research department to detect, track, and fix vulnerabilities in a timely manner. | Most enterprises do not have a vulnerability management mechanism or fix vulnerabilities in a timely manner. As a result, they are prone to attacks and are not aware of attacks and data theft. |
| Ecosystem | Third-party access | Integrates upstream and downstream ecosystem resources and provides value-added services. | It is built by vendors. |
| | Scalability | • Supports fast scale-out of tens of thousands of devices to hundreds of millions of devices without service interruption.<br>• Seamlessly interconnects with other Huawei Cloud big data, EI, and middleware products to implement storage, computing, and intelligent analysis of device data at scale and integrate with other functions such as AI. In addition, cloud-based products support small-scale verification, facilitating fast, low-cost trial and error and service innovation. | The scaling period is long. Enterprises need to implement system or component interconnection. The labor and equipment costs are high. |

**Table 2-2** Expense comparison

| Item | Huawei Cloud IoT | MQTT Cluster Managed by Enterprises |
| --- | --- | --- |
| Cloud resource cost | One SU3 supports 40 million messages per day, and 1,000 TPS upstream and downstream messages. It costs about USD1,000 per month or USD12,000 per year.<br>Total: about USD12,000/year | • Server resources: Two ECSs (AP-Singapore region, x86 architecture, general computing, 4 vCPUs, 8 GB memory, 40 GB high I/O disk, and 5 Mbit/s shared bandwidth) cost USD2102.64 per year.<br>• Relational Database Service (RDS): A general-purpose DB instance (MySQL engine, 2 vCPUs, 4 GB memory, 40 GB cloud SSD disk, and primary/standby type) costs USD890.56 per year.<br>• Elastic Load Balance (ELB): An instance (AP-Singapore region, pay-per-use, public network, shared load balancer, and 1 Mbit/s bandwidth) costs USD262.8 per year.<br>Total: USD3,256/year |

| | | |
|---|---|---|
| Labor cost | None | Basic middleware implements basic functions.<br>• One engineer is required for routine O&M and R&D of the platform.<br>• Assume that the engineer devotes 50% of efforts and the monthly salary is USD10,000.<br>• Total: 10,000 x 12 x 50% = USD60,000/year<br>Special functions are added based on the basic middleware.<br>• Assume that only some functions are implemented without considering high availability, high performance, and high security of the platform.<br>• Two full-stack development engineers are required for the platform frontend and backend development and O&M of functions such as device management, message communications, and rules.<br>• One protocol professionals are required for device-side development, including implementing device access through native protocols, generic protocols industry protocols, and SDKs.<br>• Assume that all engineers devote 100% of efforts and the monthly salary is USD10,000.<br>• Total: 3 x 10,000 x 12 x 100% = USD360,000/year<br>Bonuses are not included. |
| Total | USD12,000/year | Basic functions: USD63,256/year<br>Basic and special functions: USD423,256/year |

# 3 Application Scenarios

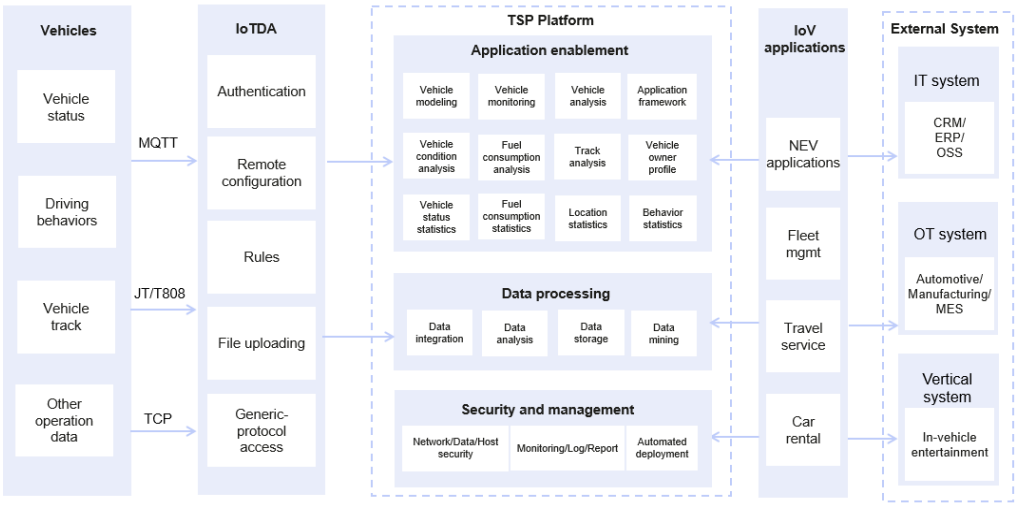
Huawei Cloud IoT Device Access (IoTDA) allows you to connect your physical devices to the cloud, where you can collect device data and deliver commands to devices.

## IoV

Requirements: For better management, automotive vendors access vehicles to the cloud platform. The platform needs to support data transmission using protocols such as JT/T808 and MQTT and can clean data for big data analysis and data mining.

Solutions: IoTDA provides secure and reliable connections at a low latency and supports multiple standard protocols. It uploads road, vehicle, and driving behavior data collected by vehicles to the cloud and processes the data with rules and FunctionGraph. In addition, IoTDA stores data in InfluxDB and DWS for big data analysis and uses ModelArts to perform machine learning to mine valuable data.

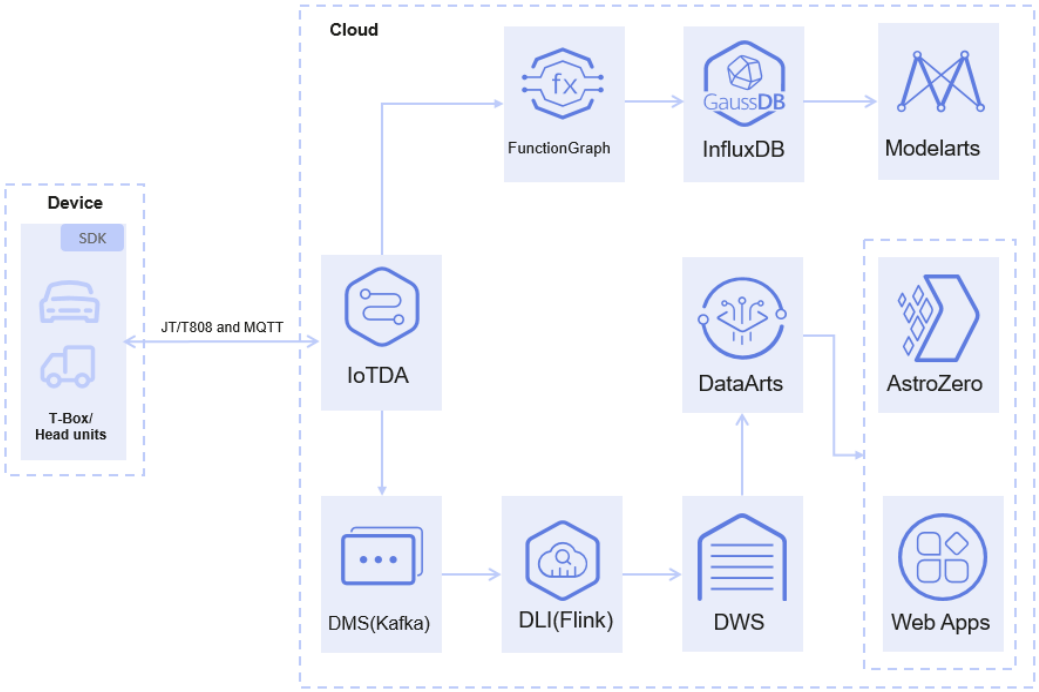
**Figure 3-1** Service architecture in the IoV scenario




Reference architecture of the IoV scenario:

- The T-Box on the device side reports data such as vehicle status, driving behavior, and vehicle track to the cloud through MQTT or JT/T808.
- IoTDA transfers data to different cloud services for data cleaning, storage, and analysis, building multiple IoV data applications.

**Figure 3-2** Reference architecture of the IoV scenario



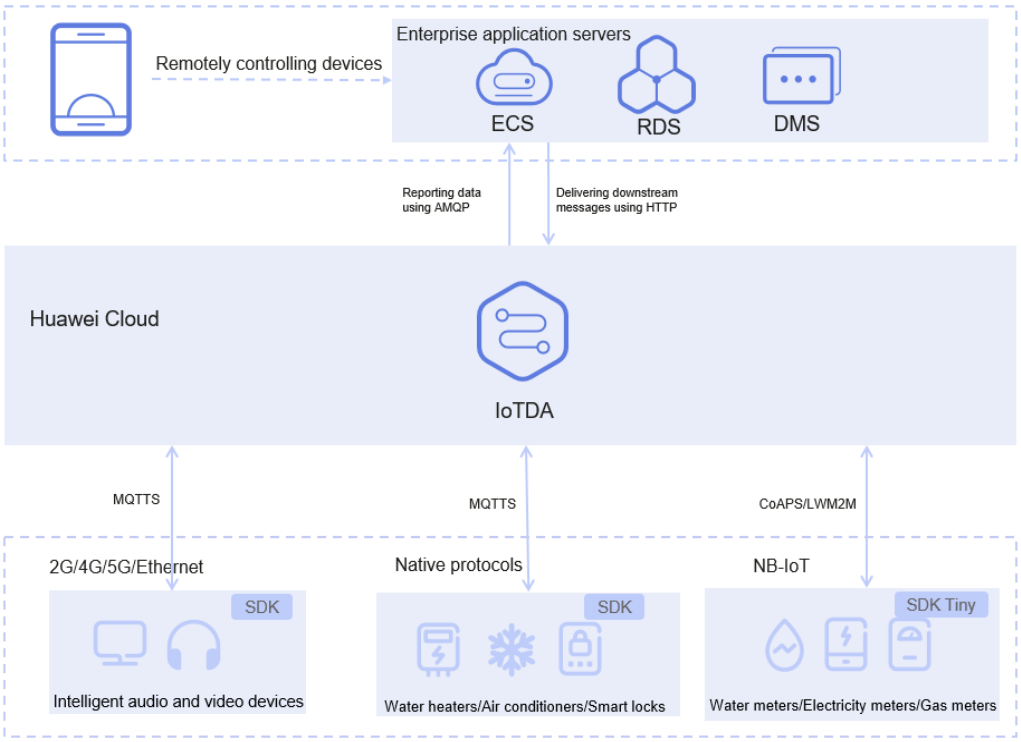

## Smart Home

Requirements: Home appliances, such as refrigerators, air conditioners, washing machines, sockets, TVs, and lighting devices, need to be connected to the cloud for data reporting and running status detection. Users can also run commands on applications provided by device vendors to remotely control devices.

Solutions: IoTDA provides a secure and reliable system to support massive device connections. It supports multiple protocols, such as MQTT, CoAP, HTTP, LwM2M, and WebSocket, and allows users to deliver messages and commands from the cloud to control devices in a timely manner.

**Figure 3-3** Reference architecture of the smart home scenario



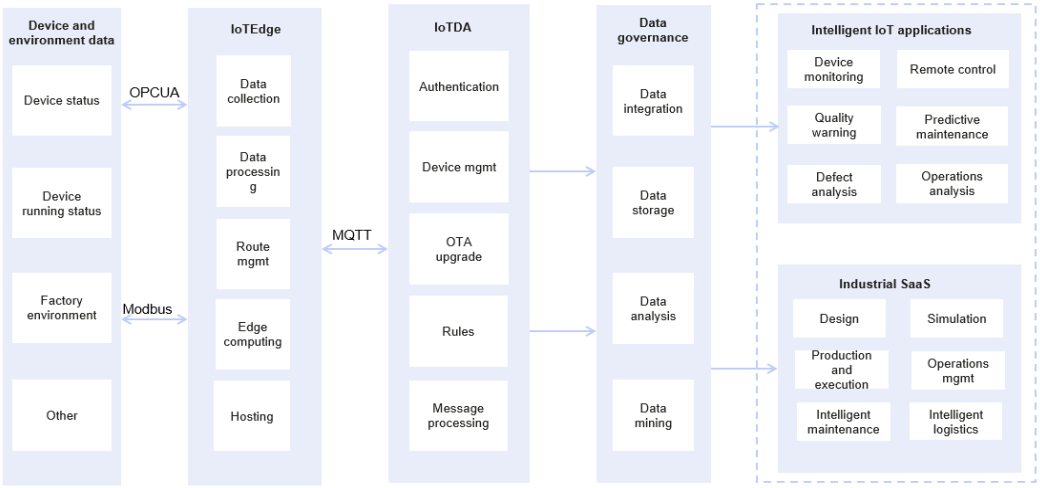

## Smart Manufacturing

Requirements: To manage mechanical devices of various brands and types on the assembly lines, factories need to collect device running and environment monitoring data with a cloud platform. The platform should calculate and analyze device running status in real time, provide device exception or fault prediction and alarm, and allow users to perform remote maintenance and upgrade.

Solutions: Factories can use IoT Edge to collect OT data such as device and environment data, report them to IoTDA through industrial gateways, and transfer them to other cloud services for data conversion and analysis. With this data, factories can monitor devices, receive alarms for abnormal devices, and upgrade and maintain devices remotely.

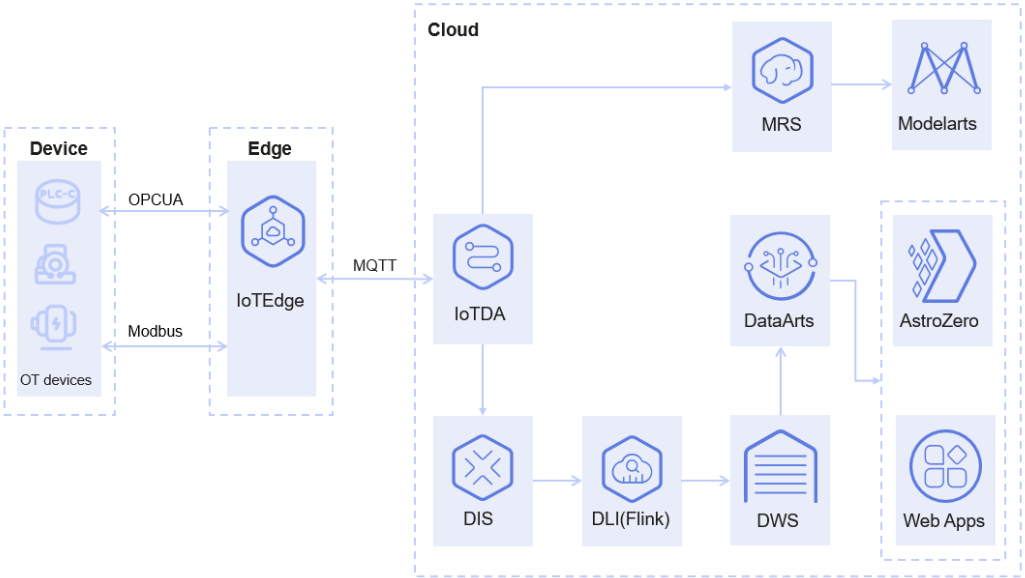
**Figure 3-4** Business architecture of the smart manufacturing scenario




Reference architecture of the smart manufacturing scenario:

- Edge nodes deployed on devices support protocols such as OPC UA and Modbus, collect running data, status data, and environment monitoring data of various OT devices, and report data from edge gateways to IoTDA.
- IoTDA uses rules to transfer data to Data Ingestion Service (DIS). After being processed by DLI-Flink, the data is written to DWS for subsequent data governance. Data can also be transferred to MapReduce Service (MRS) for big data cleaning and processing, facilitating subsequent AI analysis and data mining.

**Figure 3-5** Reference architecture of the smart manufacturing scenario




## Distributed Photovoltaics (PV)

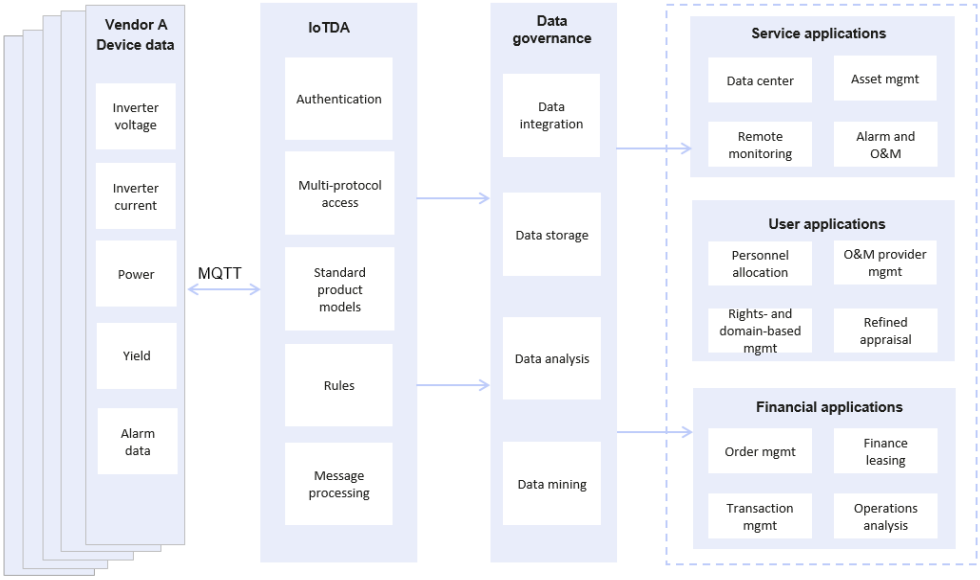
Requirements: New energy companies need to collect the voltage, current, power, yield, and alarm data of inverters produced by various vendors to the cloud for

further data processing and analysis, facilitating data center development, alarm O&M, and operation analysis.

Solutions: IoTDA provides standard object models and supports multi-protocol access. Shielding the format and protocol differences of data reported by devices from multiple PV device vendors, IoTDA uses rules to transfer data to Object Storage Service (OBS) for storage and MRS for further data processing.

**Figure 3-6** Service architecture in the distributed PV scenario

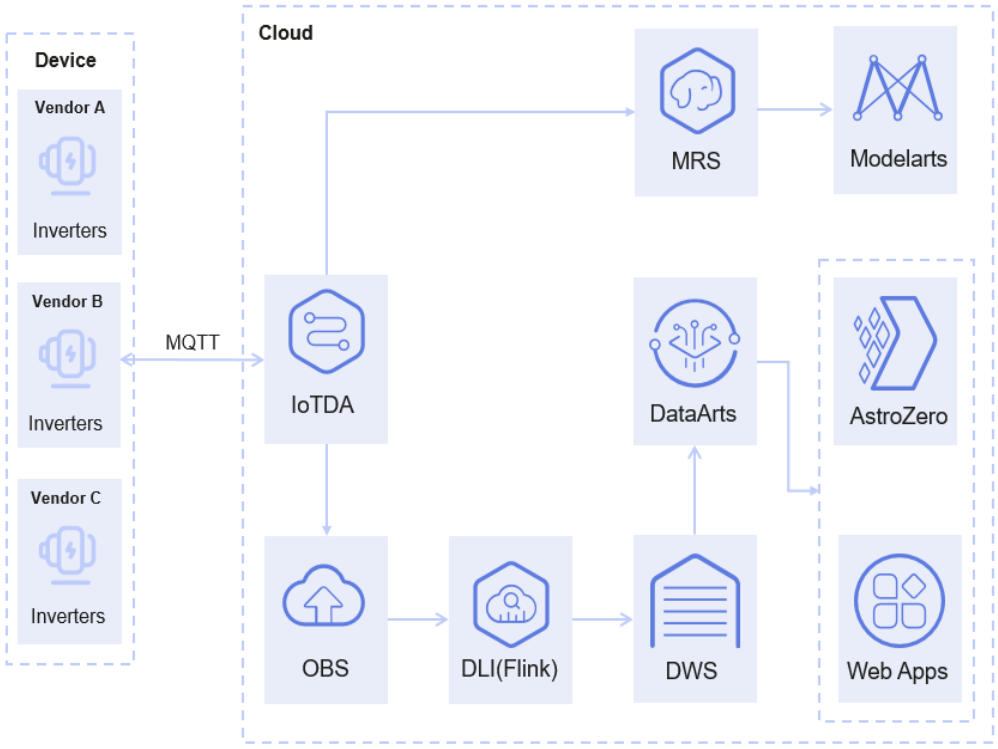



Reference architecture of the distributed PV scenario:

- Inverters from different vendors report data such as voltage, current, power, and yield to the cloud through MQTT.
- IoTDA uses rules to transfer data to OBS for storage. After being processed by DLI-Flink, the data is written to DWS for subsequent data governance. Data can also be transferred to MRS for big data cleaning and processing, facilitating subsequent AI analysis and data mining.

**Figure 3-7** Reference architecture of the distributed PV scenario




## Smart Charging Pile

Requirements: Charging pile operators need to collect charging data, meter information, and charging vehicle information of charging piles produced by different vendors to the cloud, so cloud applications can detect the status of user vehicles and charging piles in real time for fee calculation. Applications need to deliver commands to start and stop the charging.

- Solution 1: Charging pile devices from multiple vendors are directly connected to the IoTDA through MQTT. Generic-protocol plug-ins are deployed on the cloud for parsing and multi-protocol access. IoTDA can directly push data to customers' applications and allow applications to deliver commands to control the start and stop of the charging. This solution applies to urban areas or outdoor areas with good network environments.
- Solution 2: IoT Edge is used to collect charging pile data from multiple vendors. Protocol plug-ins can be deployed on edge nodes to shield proprietary protocols of multiple vendors. Some simple computing applications can also be deployed on edge nodes to reduce interaction with the cloud. The edge gateway reports data to the IoTDA through MQTT. IoTDA can directly push data to customers' applications and allow applications to deliver commands to control the start and stop of the charging. This solution applies to areas with poor network environments, such as highway service areas and underground parking lots.

**Figure 3-8** Service architecture in the smart charging pile scenario

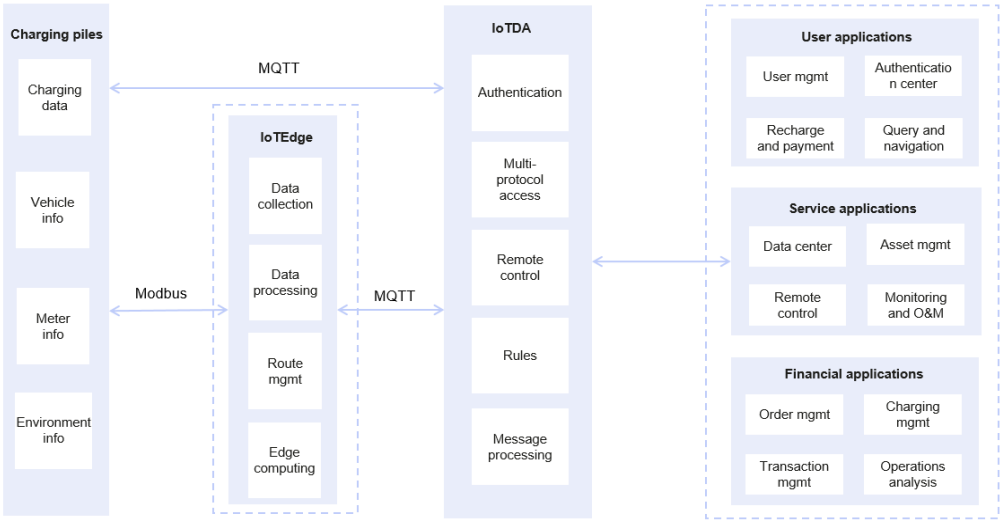
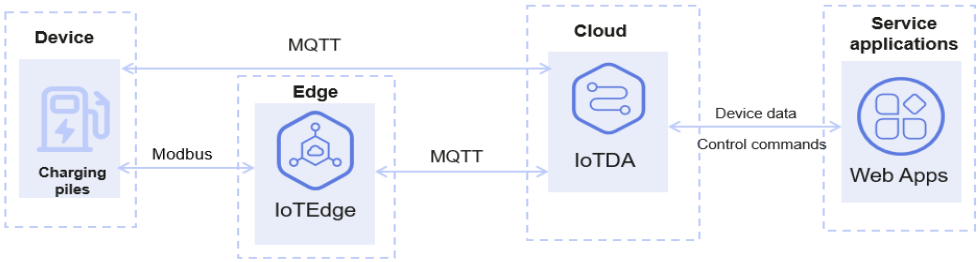



**Figure 3-9** Reference architecture of the smart charging pile scenario



# 4 Specifications

## Specifications of Standard Edition

IoTDA provides standard instances for device access and service processing. Each Huawei Cloud user can enable up to one standard instance in the same region. If it cannot meet your requirements, submit a service ticket. You can select the type and number of units in the standard instance to determine the total number of messages allowed per day between devices and the platform. If you increase the number of units of the same type, the total number of messages will increase and the function **limitations** will change as well.

After an instance is enabled, you are billed based on the usage duration (days) and unit type and quantity you select. For details about product prices, see **Price Calculator**. Select the required configuration and check the configuration fee at the bottom of the page. For details, see **Billing Overview**.

**Table 4-1** Unit specifications

| Specification | Unit Type | Messages/Day | Max. Message Size | Upstream/ Downstream Message TPS |
|---|---|---|---|---|
| iotda.standard.suf | SUF | 10,000 | 4 | 10 |
| iotda.standard.su1 | SU1 | 400,000 | 4 | 10 |
| iotda.standard.su2 | SU2 | 4 million | 4 | 100 |
| iotda.standard.su3 | SU3 | 40 million | 4 | 1,000 |
| iotda.standard.su4 | SU4 | 300 million | 4 | 6,000 |

NOTE

- A standard instance can be configured with multiple units of the same type, for example, five SU1 units, but cannot be configured with different types of units, for example, two SU1 units and three SU2 units. You can change the number and type of units at any time. For example, you can upgrade two SU1 units to five SU1 units or two SU1 units to two SU2 units. You can enable only one SUF unit but 100 SU1, SU2, SU3, or SU4 units. An SUF unit can be upgraded to an SU1, SU2, SU3, or SU4 units. After the upgrade, the original SUF unit is no longer retained.
- Number of messages: For details about the number of messages to be billed, see **Billing Items**. If the message limit is reached, IoTDA generates an alarm and rejects new messages. Upgrade the unit specifications or increase the number of units if necessary.
- Message size: Max. size for messages using MQTT: 1 MB. Max. size for messages using LwM2M over CoAP: 1 KB. Messages whose sizes exceed the limit will be rejected. Messages are counted as one unit up to 4 KB; any excess is counted as additional units.
- Number of devices: A standard instance supports up to two million registered devices. To increase the quota, **submit a service ticket**. All registered devices can go online concurrently. When only an SUF unit is enabled, up to 1,000 devices can be registered.
- TPS of upstream and downstream messages: Max. throughput of upstream and downstream messages per second, that is, the total number of messages exchanged between devices and the platform. The quotas of standard instances are determined by the type and number of units. **Submit a service ticket** for special requirements if necessary.
- For limitations on other than unit specifications and quota, see **Limitations**.

Calculating the required instance specifications:

- Scenario: A user enables the IoTDA standard instance and plans to register 100,000 devices. The average number of concurrent online devices per day is 10,000. Each online device sends a message (smaller than 4 KB) to the platform every 5 seconds on average. No API calls or push messages are involved. The daily working duration is 8 hours.
- The TPS of upstream and downstream messages is 2,000 (10,000 devices/5 seconds × 1 message per device). The total number of messages per day is 57,600,000 (8 hours × 60 × 60 seconds × 2,000 TPS). In this case, you can purchase two SU3 units, which support 2,000 TPS, 80 million messages per day, and up to 2 million registered devices, and cost about USD2,100 (USD1,050 x 2). Alternatively, you can purchase 20 SU2 units. Bills are generated and deduction is triggered every day.

# 5 Limitations

The following table lists the technical specifications of IoTDA. If the specifications cannot meet your service requirements, **submit a service ticket** to describe your requirements.

**Table 5-1** Resource restrictions

| Category | Object | Description | Limit |
| --- | --- | --- | --- |
| Instance management | Standard edition instance | Number of units purchased for a standard instance (see **Specifications of Standard Edition**) | 100 |
| Resource space management | **Resource space** | Number of resource spaces supported by an instance | 10 |
| Device access | MQTT | MQTT protocol standard | Supported: MQTT v3.1, v3.1.1, and v5.0; not supported: QoS 2, and will, and retained messages |
| | | Security levels supported by MQTT | TLS 1.1, TLS 1.2, and TLS 1.3 |

| Category | Object | Description | Limit |
| --- | --- | --- | --- |
| | | Heartbeat interval of MQTT connections (see **Establishing a Connection**) | Range: 30s to 1200s; recommended: 120s (If the heartbeat interval is not within the range, the server will reject the connection request.) |
| | | Maximum timeout interval = Heartbeat interval x 1.5 (If no device message is received within the maximum timeout interval, the server automatically disconnects from the device.) | |
| | | Number of connections established between a device and IoTDA | 1 |
| | | Length of a custom MQTT topic | 128 bytes |
| | | Size of a message reported by an MQTT device | 1 MB (A message larger than this size is rejected.) |
| | | Number of subscriptions for an MQTT connection | 50 |
| | | Bandwidth of an MQTT connection | 1 Mbit/s |
| | | Number of upstream messages per second for an MQTT connection | 50 |
| | | Number of connection requests created per second for an instance | **Specifications of Standard Edition** |

| Category | Object | Description | Limit |
| --- | --- | --- | --- |
| | | Number of upstream requests for an instance per second on the device side (assuming that the average payload of a message is 512 bytes) | **Specifications of Standard Edition** |
| | CoAP/LwM2M | CoAP version | RFC 7252 standard 3 |
| | | LwM2M version | V1.0.2 |
| | | Transport layer protocol | User Datagram Protocol (UDP) |
| | | Security level supported by CoAP | DTLS v1.2 |
| | | Size of a CoAP message packet | 1 KB |
| | | Number of messages per minute for a device | 300 |
| | HTTP | HTTP versions | HTTP 1.0 and HTTP 1.1 |
| | | TLS versions | TLS 1.1 and TLS 1.2 |
| | | Size of a message body | 1 MB |
| Device management | Product | Number of products in a resource space | 1,000 |
| | | Size of the service capability JSON file in a product | 500 KB |
| | | Number of services in a product | 500 |

| Category | Object | Description | Limit |
| --- | --- | --- | --- |
| | | Number of properties, events, or commands in a service capability | 500 |
| | Custom topic | Number of custom topics for a product | 50 |
| | Number of devices in a standard instance | Number of devices registered for a standard edition (SU1, SU2, SU3, and SU4) | 2,000,000 (If you want to increase the limit, submit a service ticket.) |
| | | Number of devices registered for an SUF | 1,000 |
| | Device | Number of child devices added to a gateway | 50,000 |
| | | Depth of the gateway structure | 2 levels |
| | Device tag | Number of tags added to a device | 10 |
| | Group | Depth of a group | 5 levels |
| | | Number of groups in a resource space | 1,000 |
| | | Number of devices in a group | 20,000 |
| | | Number of groups that a device can be added to | 10 |
| | Batch task | Number of devices registered at a time | 100,000 |
| | Codec | Number of codecs allowed in a product | 1 |
| | | Size of the codec package uploaded offline | 4 MB |

| Category | Object | Description | Limit |
| --- | --- | --- | --- |
| | | Length of the codec script | 1 MB |
| | | Timeout interval for a codec invoking request | 5s |
| | Device linkage rule | Number of rules added for an instance | Basic/Standard edition: 20 |
| | | | Enterprise edition: 200 |
| | | Number of actions supported by a rule | 10 |
| | | Number of rule actions executed by devices per second for an instance | Basic/Standard edition: 10 |
| | | | Enterprise edition: 100 |
| | | Number of waiting tasks for an instance | Basic/Standard edition: 100 |
| | | | Enterprise edition: 1,000 |
| | Batch task | Number of batch tasks concurrently processed in a resource space | 10 |
| | | Size of a batch task file | 2 MB |
| | | Number of lines in a batch task file | 100,000 |
| | | Number of batch task files supported by an instance | 10 |
| | OTA upgrade | Size of an upgrade package | Package uploaded to IoTDA: 20 MB; package uploaded to OBS: no limit |

| Category | Object | Description | Limit |
| --- | --- | --- | --- |
| | | Number of upgrade packages uploaded to a resource space | 200 |
| | | Size of a file uploaded to a resource space | File uploaded to IoTDA: 500 MB; file uploaded to OBS: no limit |
| Message communications | Synchronous command | Response time of a device to a synchronous command | 20 seconds |
| | Device message | Aging time of a message delivered to a device | 24 hours |
| | | Size of a message delivered to a device | 256 KB |
| | | Number of cached messages to be delivered to a device | 20 |
| | Device property | Number of child devices of which properties can be reported by a gateway at a time | 100 |
| | Asynchronous command | Aging time of an asynchronous command delivered to a device | 48 hours |
| | | Size of an asynchronous command delivered to a device | 256 KB |
| | | Number of cached asynchronous commands delivered to a device | 20 |

| Category | Object | Description | Limit |
| --- | --- | --- | --- |
| Message forwarding | Forwarding rule and action | Number of rules configured for an instance | 100 |
| | | Number of actions supported by a rule | 10 |
| | | Length of the **select** parameter in a forwarding rule | 0.5 KB |
| | | Length of the **where** parameter in a forwarding rule | 0.5 KB |
| | Cache policy for message forwarding | Message cache size | 1 GB |
| | | Message cache duration | 24 hours |
| | AMQP | AMQP version | AMQP 1.0 |
| | | TLS version | TLS 1.2 |
| | | Number of queues configured for an instance | 100 |
| | | Number of queues listened on for a connection | 10 |
| | | Number of connections allowed in an instance | 32 |
| | Flow control and stack policies | Number of data forwarding stack policies | 1 |
| | | Number of data forwarding flow control policies | 4 |
| Certificate management | Certificate configuration | Number of device CA certificates supported by an instance. | 100 |

| Category | Object | Description | Limit |
| --- | --- | --- | --- |
| | | Number of application CA certificates supported by an instance. | 10 |

Application API limitations:

Unless otherwise specified, an API can be called up to 100 times per second for an account (standard edition).

# 6 Security

## 6.1 Shared Responsibilities

Huawei guarantees that its commitment to cyber security will never be outweighed by the consideration of commercial interests. To cope with emerging cloud security challenges and pervasive cloud security threats and attacks, Huawei Cloud builds a comprehensive cloud service security assurance system for different regions and industries based on Huawei's unique software and hardware advantages, laws, regulations, industry standards, and security ecosystem.

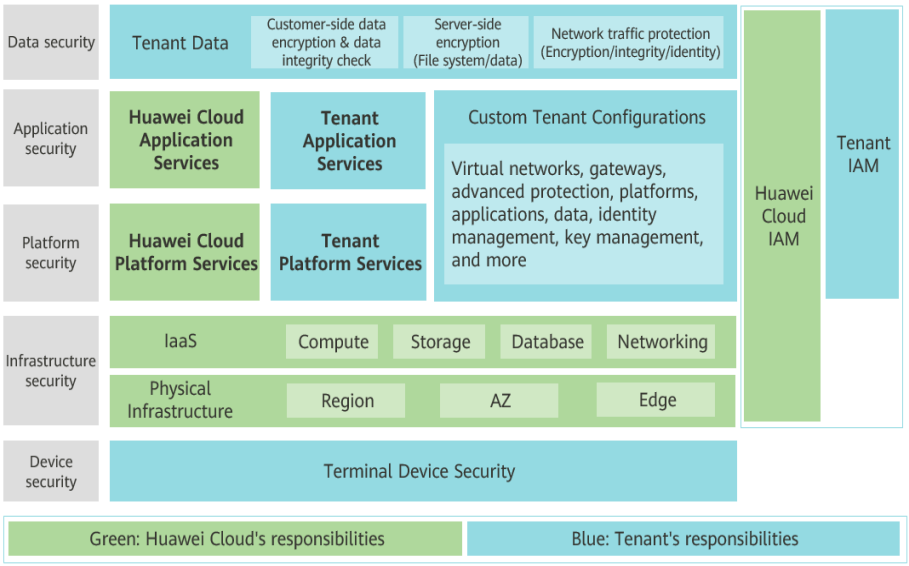
**Figure 6-1** illustrates the responsibilities shared by Huawei Cloud and users.

- **Huawei Cloud**: Ensure the security of cloud services and provide secure clouds. Huawei Cloud's security responsibilities include ensuring the security of our IaaS, PaaS, and SaaS services, as well as the physical environments of the Huawei Cloud data centers where our IaaS, PaaS, and SaaS services operate. Huawei Cloud is responsible for not only the security functions and performance of our infrastructure, cloud services, and technologies, but also for the overall cloud O&M security and, in the broader sense, the security and compliance of our infrastructure and services.
- **Tenant**: Use the cloud securely. Tenants of Huawei Cloud are responsible for the secure and effective management of the tenant-customized configurations of cloud services including IaaS, PaaS, and SaaS. This includes but is not limited to virtual networks, the OS of virtual machine hosts and guests, virtual firewalls, API Gateway, advanced security services, all types of cloud services, tenant data, identity accounts, and key management.

**Huawei Cloud Security White Paper** elaborates on the ideas and measures for building Huawei Cloud security, including cloud security strategies, the shared responsibility model, compliance and privacy, security organizations and personnel, infrastructure security, tenant service and security, engineering security, O&M security, and ecosystem security.

**Figure 6-1** Huawei Cloud shared security responsibility model




# 6.2 Identity Authentication and Access Control

## Identity Authentication

You are required to carry your identity credential and verify the identity validity when calling IoTDA APIs. Different identity credentials are used in the following IoTDA access scenarios:

- IoTDA application APIs support authentication using an IAM token or access key (AK/SK). For details, see **Authentication**.
- For MQTT device connection authentication, carry the client ID, device ID, and encrypted device secret. For details, see **Device Connection Authentication**.
- For HTTP device connection authentication, carry the device ID, password authentication mode, timestamp, and encrypted device secret. For details, see **Authenticating a Device**.
- For authenticating the connection between the AMQP client and IoTDA, carry **accessKey** and **accessCode**. For details, see **AMQP Client Access**.

## Access Control

IoTDA supports access control through IAM. IAM permissions define which actions on your cloud resources are allowed or denied. After creating an IAM user, the administrator needs to add it to a user group and grant the permissions required by IoTDA to the user group. Then, all users in this group automatically inherit the granted permissions.

IAM presets system permissions for each cloud service so that you can quickly configure basic permissions. The **following table** describes all system permissions of IoTDA.

**Table 6-1** All system permissions of IoTDA

| Role/Policy Name | Description | Type |
| --- | --- | --- |
| Tenant Administrator | Permissions to perform all operations on all services except IAM | System-defined role |
| Tenant Guest | Permissions to perform read-only operations on all services except IAM | System-defined role |
| IoTDA FullAccess | Permissions to perform all operations on IoTDA resources. | System-defined policy |
| IoTDA ReadOnlyAccess | Permissions to perform read-only operations on IoTDA resources. | System-defined policy |

# 6.3 Data Protection

The shared responsibility model applies to IoTDA data protection. IoTDA is responsible for the service security and provides a secure data protection mechanism. You are responsible for securely using IoTDA, including configuring security parameters and maintaining the control of permissions to use IoTDA and other dependent cloud services.

**Table 6-2** Data protection technologies

| Measure | Description | Reference |
| --- | --- | --- |
| Transmission encryption (HTTPS) | IoTDA supports HTTPS. To secure data transmission, use TLS 1.2 or later. | **Using HTTPS For Access** |
| Transmission encryption (MQTTS) | IoTDA supports MQTTS. To secure data transmission, use TLS 1.2 or later. TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 are recommended as cipher suites. | **MQTT Protocol Support** |

| Measure | Description | Reference |
| --- | --- | --- |
| Transmission encryption (AMQPS) | IoTDA supports AMQPS. To secure data transmission, the receiver must use TLS 1.2 or later for encryption. Non-encrypted TCP transmission is not supported. | AMQP Client Access |

# 6.4 Auditing and Logging

## Auditing

Cloud Trace Service (CTS) records operations on the cloud resources in your account. You can use the logs generated by CTS to perform security analysis, trace resource changes, audit compliance, and locate faults. For details about how to enable and configure CTS, see **Enabling CTS**.

When using IoTDA, you can use CTS to view your operations and results. For details about IoTDA audit records, see **Audit Logs**.

## Logging

IoTDA records connections with devices and applications and reports them as logs to LTS. LTS provides real-time query, mass storage, structured processing, and visualized chart analysis.

For details about IoTDA logs, see **Run Logs**.

# 6.5 Risk Monitoring

IoTDA provides comprehensive monitoring and O&M capabilities, including device message trace, report viewing, alarm management, and device anomaly detection, helping you obtain information about all devices connected to IoTDA in real time.

- **Message trace**: If a fault occurs during device authentication, command delivery, data reporting, data forwarding and other service scenarios, you can use message trace to quickly locate and analyze the fault.
- **Report Viewing**: You can view statistics in different dimensions, including device messages, device status, and the total number of devices.
- **Alarm management**: Based on the alarm notification capability of Application Operations Management (AOM), if a rule you set is triggered, IoTDA will send an alarm notification for you to pay attention to and handle the alarm in a timely manner.

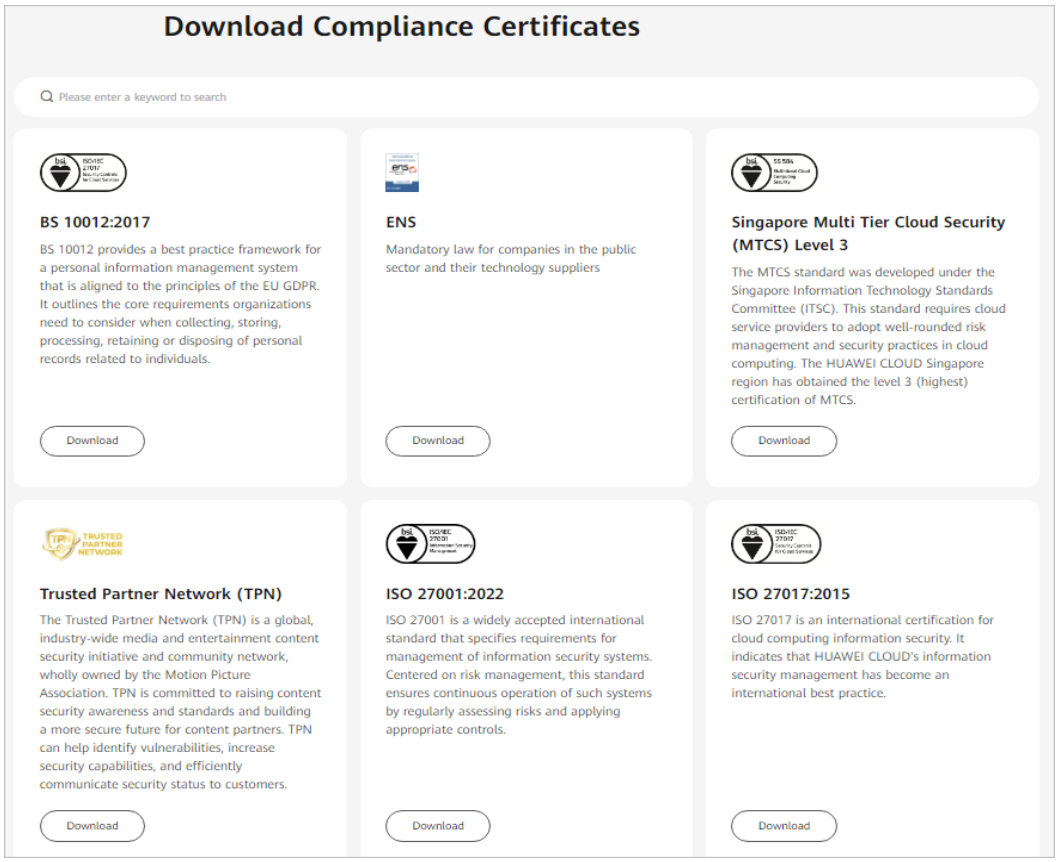
  

# 6.6 Certificates

## Compliance Certificates

Huawei Cloud services and platforms have obtained various security and compliance certifications from authoritative organizations, such as International Organization for Standardization (ISO). You can **download** them from the console.

**Figure 6-2** Downloading compliance certificates

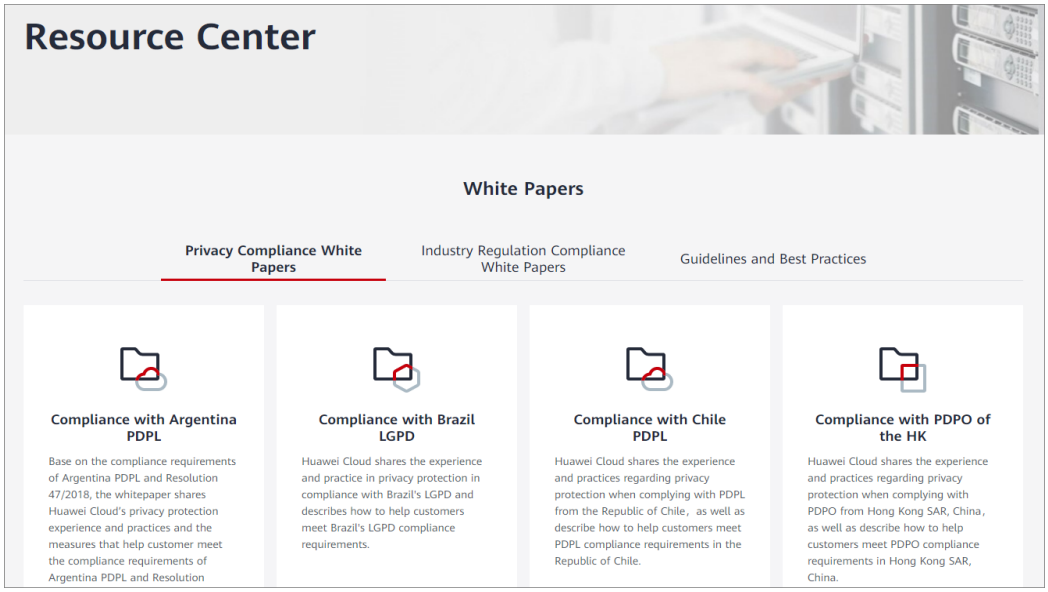



## Resource Center

Huawei Cloud also provides the following resources to help users meet compliance requirements. For details, see **Resource Center**.

**Figure 6-3** Resource center



# 7 Basic Concepts

## Basic Concepts

| Term | Description |
| --- | --- |
| IoTDA | Short for IoT Device Access, a basic service of the Huawei Cloud IoT platform. IoTDA provides functions such as device fleet access, bidirectional communications between devices and the cloud, batch device management, remote control and monitoring, OTA upgrade, and device linkage rules. It can flexibly transfer device data to other Huawei Cloud or third-party services. Using IoTDA, you can quickly connect devices to the platform and integrate your applications. |
| Resource space | A space allocated for your applications. Resources (such as products and devices) created on the platform must belong to a resource space. You can use the resource space for domain-based management as well as resource isolation and authorization management. |
| AppID | Resource space ID (the **app_id** parameter for API calling) used as the unique identifier of the resource space. |
| ProjectID | Unique identifier of a project. A default project is provided for each Huawei Cloud region to group and isolate resources (including compute, storage, and network resources) across physical regions. IAM users can be granted permissions to access all resources in a specific project. |
| Edge node | An open platform located close to devices or data sources. It integrates core capabilities of network, compute, storage, and applications to provide compute and intelligence services locally, making real-time services intelligent and ensuring data privacy and security. |
| Module | Also called a communications module, an independent display unit consisting of display modules, drive circuits, control circuits, chips, and mechanical parts. Devices communicate with the platform through modules. Currently, 2G/3G/4G/5G, NB-IoT, and Wi-Fi communications modules are provided. |

## Device Access

| Term | Description |
| --- | --- |
| Device | A device is a physical entity that belongs to a product. Each device has a unique ID. It can be a device directly connected to the platform, or a gateway through which child devices are connected to the platform. |
| Device ID | An ID that uniquely identifies a device. It is allocated by the platform during device registration and used for device access authentication and message transmission. |
| Node ID | A unique physical identifier for a device, such as its IMEI or MAC address. This parameter is used by the platform to authenticate the device during device registration. |
| CoAP/ CoAPS | A software protocol designed to enable simple devices to perform interactive communication on the Internet.<br>CoAPS refers to CoAP over DTLS. DTLS is used for encrypted transmission. |
| LwM2M | Short for Lightweight Machine to Machine, an IoT protocol defined by Open Mobile Alliance (OMA). It is mainly used for NB-IoT devices with limited resources (such as limited storage and power supply). |
| MQTT/ MQTTS | An IoT transmission protocol designed for lightweight publish/ subscription messaging. It provides reliable network services for IoT devices in low-bandwidth and unstable network environments.<br>MQTTS refers to the combination of MQTT and SSL/TLS. SSL and TLS are used for encrypted transmission. |
| Device CA certificate | A certificate issued by a certification authority (CA) such as VeriSign, Symantec, and GlobalSign and used to verify the identity between the server and client during HTTPS link establishment. |
| X.509 device certificate | A digital certificate used to authenticate communication entities. After a device with authentication mode set to X.509 certificate is created, the platform issues the X.509 certificate to the device. |
| Secret | Used for authentication when a device uses native MQTT to connect to the platform. |
| PSK | Used to encrypt the transmission channel between the platform and NB-IoT devices or devices integrated with the SDK. |

## Device Management

| Term | Description |
| --- | --- |
| IAM | Short for Identity and Access Management, a Huawei Cloud service that provides identity authentication and permission management. You can use IAM to manage user accounts (such as employees, systems, and applications) and control the operation permissions of these users on your resources. |
| Product | A collection of devices with the same capabilities or features. It helps developers quickly develop product models and codecs, and provides capabilities such as device integration, online commissioning, and topic customization, facilitating end-to-end IoT development and helping you improve integration development efficiency and shorten the construction period of IoT solutions. |
| Product model | A product model, also called a thing model, is used to describe the capabilities and features of a device. You can construct an abstract model of a device type by defining a product model on the platform, allowing it to understand the services, properties, and commands supported by the device. |
| Product ID | An ID that identifies the product to which a device belongs. This parameter is used to associate the product model of the device. |
| Service | Part of the product model that describes the capabilities of a device. Device capabilities are divided into several services. The properties, commands, and command parameters of each service are defined in the product model. |
| Property | Part of the product model that describes the running status of a device, such as the current ambient temperature read by an environment monitoring device. |
| Topic | A UTF-8 character string functioning as the transmission medium of publish/subscription messaging. You can publish messages to or subscribe to messages from a topic. |
| Command | A capability or method that can be invoked by external systems. |
| Event | A functional model of a device, which is an event during device running. Events can be subscribed to and pushed. |
| Codec | Plug-in used for format conversion. The platform communicates with applications using data in JSON format. For a device that reports binary data, you must develop codecs for the platform to convert binary data into JSON data. For a device that reports JSON data, you can develop codecs to convert data in different JSON formats. |
| Gateway | A physical entity that manages child devices and connects child devices to the platform. |

| Term | Description |
| --- | --- |
| Child device | A physical device that connects to the platform through a gateway. |
| Firmware | A driver underlying the device hardware. It is responsible for the underlying work of a system, for example, the basic input/output system (BIOS) on a computer mainboard.<br>Firmware upgrade, also called firmware over the air (FOTA), allows users to upgrade the firmware of LwM2M or MQTT devices in OTA mode. For example, an upgrade of an NB-IoT module is a firmware upgrade. |
| Software | Software is classified into system software and application software. The system software is for the basic device functions, such as the compilation tool and system file management. The application software provides functions such as data collection, analysis, and processing, depending on the features of the device.<br>Software upgrade, also called software over the air (SOTA), allows users to upgrade the software of LwM2M or MQTT devices in OTA mode. For example, an MCU upgrade is a software upgrade. |
| PCP | A protocol that defines the communications content and format between devices and the platform for device upgrades. |
| Group | A collection of devices. You can create groups for all the devices in a project space based on different rules, such as regions and types, and you can manage and operate the devices by group. |
| Tag | You can define tags and bind tags to devices. |
| Device shadow | A JSON file that stores the device status, latest device properties reported, and device configurations to be delivered. Each device has only one shadow. A device can retrieve and set its shadow to synchronize the status, either from the shadow to the device or from the device to the shadow. |

## Data forwarding

| Term | Description |
| --- | --- |
| Rule engine | A preset condition used by the platform to trigger actions. The device will report device data, which is checked against the rules. When a rule condition is met, the platform will trigger corresponding actions such as delivering a command to the device or forwarding data to other Huawei Cloud services. You can create device linkage and data forwarding rules. |

| Term | Description |
| --- | --- |
| Subscription and push | Subscription: An application calls a platform API to learn changes to device service details (such as device registration, data reporting, and device status) and management details (such as software or firmware upgrade statuses and upgrade results) from the platform.<br>Push: After a subscription is successful, the platform pushes the corresponding change to the specified callback URL or AMQP message queue based on the type of data subscribed. |
| token | An authentication parameter used to call platform APIs. When an application accesses the platform for the first time, it must call the authentication API to get authenticated and obtain a token. |
| AMQP | AMQP, an advanced message queue protocol at the application layer of the unified messaging service, is an open standard application layer protocol for message-oriented middleware. The platform can communicate with and transfer data to applications using AMQP. |