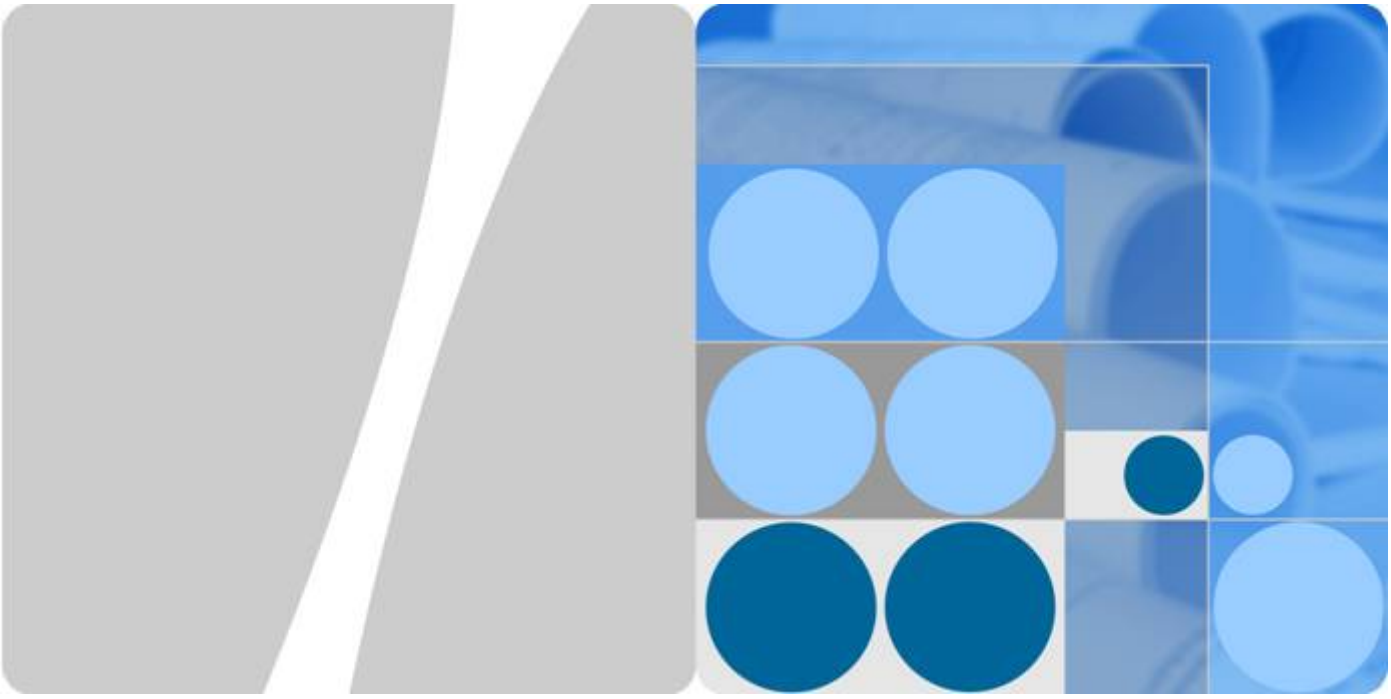


ROMA

Service Overview

Issue 01

Date 2019-08-26

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

Trademarks and Permissions


and other Huawei trademarks are trademarks 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 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.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.huawei.com

Email: support@huawei.com

Contents

1 What Is ROMA?

Enterprises face many difficulties in digital transformation:

- There is no unified way to integrate device information.
- Data formats are diversified, which makes it difficult to transmit and integrate data.
- There is no convenient way to share data and backend services with partners.
- There is no secure information channel for cloud and on-premises applications across different networks.

Relationship, Open, Multi-Ecosystem, and Any-Connect (ROMA) is a full-stack enterprise integration platform that is derived from Huawei's digital transformation integration practice. It focuses on application and data connections and applies to **multiple common scenarios of enterprises**. ROMA provides lightweight message, data, API, and device integration to simplify enterprise cloudification and support cross-regional integration for cloud and on-premises applications, helping enterprises achieve digital transformation.

ROMA consists of four components: device integration (LINK), message integration (MQS, short for Message Queue Service), data integration (FDI, short for Fast Data Integration), and business integration (APIC, short for API Connect).

- **LINK**

  LINK is a device integration service that uses the standard Message Queue Telemetry Transport (MQTT) protocol to connect devices, helping enterprises quickly and easily manage devices on the cloud.

  In industrial scenarios, device information and parameters involved in the production process are scattered. If a fault occurs in a production line, it takes a long time to manually collect information and parameters for each device. LINK connects devices to IT systems or big data platforms, and uploads information such as device running status to these platforms so that enterprise customers can see information about all devices graphically and therefore quickly locate faults. In addition, enterprise customers can configure the upper thresholds for device parameters to rule engines of LINK. If real-time parameters of a device are close to the upper thresholds, an alarm notification is sent to users to remind them to stop the device and perform maintenance.

- **MQS**

  Based on the real-time data processing platform Kafka, MQS sets up a message queue service that provides functions such as publishing and subscription, message tracking, resource statistics, and alarm monitoring. MQS provides enterprises with secure and standardized message channels for cross-network access.

For example, if an enterprise and its partners use different message systems, interconnection between the message systems is costly, and message transmission after the interconnection may not be reliable or secure. To address these issues, the Kafka protocol can be used for communication between the enterprise and its partners. In this way, MQS functions as a message transfer station to provide secure and reliable message transmission. Specifically, the enterprise can create multiple topics, set the permission for each partner to subscribe to these topics, and publish messages to the topics. Then, partners can subscribe to the topics to obtain messages in real time.

- **FDI**

  FDI supports flexible, fast, and non-intrusive data integration between multiple data sources, such as text, messages, APIs, and structured and unstructured data. It implements data integration across equipment rooms, data centers, and clouds, and supports automatic deployment, O&M, and monitoring of integrated data.

  For example, if an enterprise and its partners use different data sources, it is difficult to achieve effective information transmission. FDI provides multiple methods to convert mainstream data source formats such as MySQL, Kafka, and API. FDI can also cooperate with other services of HUAWEI CLOUD, such as **Data Ingestion Service (DIS)** and **Data Warehouse Service (DWS)**, to store, convert, and analyze big data.

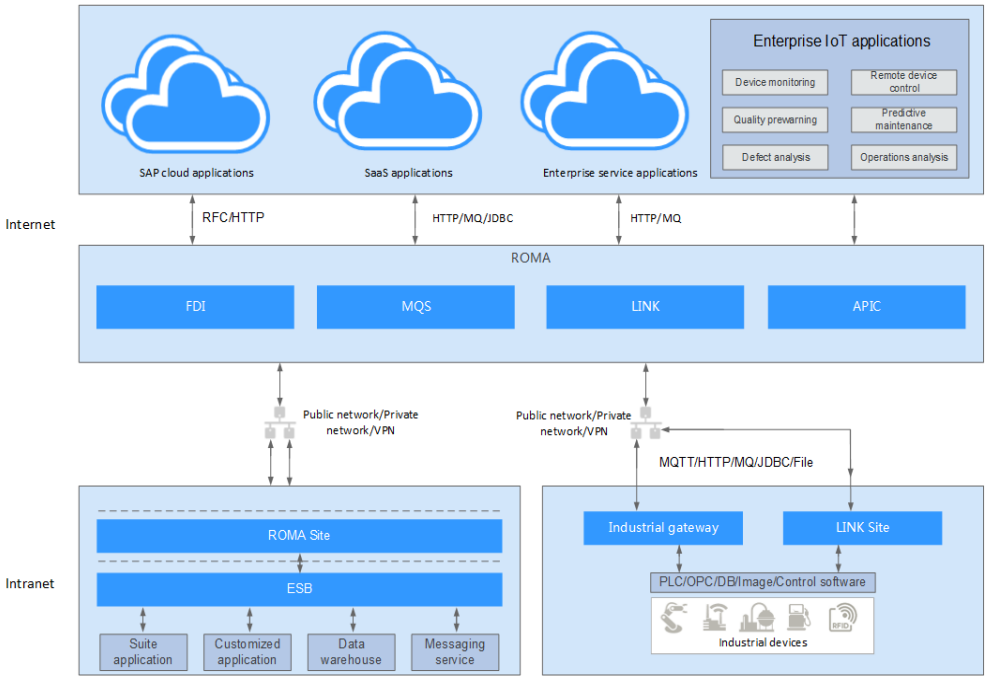
- **APIC**

  APIC provides data and backend services to an enterprise's branches or partners through APIs. This simplifies data sharing and service provisioning and reduces the interconnection costs for the enterprise. APIC provides SDKs in different programming languages and sample code, simplifying API calls.

  For example, if a company headquarters integrates its IT system with those of its branches in different regions, it is too complex to directly access each other's database and information disclosure may occur. If APIs are used to access databases and security is enhanced for API call, cross-network and cross-regional collaboration can be achieved.

During enterprise cloudification, multiple components need to be used together to integrate applications and data, as shown in **Figure 1-1**.

Figure 1-1 Overall ROMA architecture



2 Typical Application Scenarios

2.1 Smart Campus Integration

Many difficulties are encountered in smart campus management:

- Customized management systems hinder information collection and sharing.

  Buildings in a campus have different structures. Enterprises can customize subsystems for each building to collect all information on each one. However, after customization, the differences between subsystems hinder information collection and sharing, resulting in difficulty in information transmission. This reduces the "smart" level of a campus.
- Diversified devices and complex data collection make it difficult to implement system linkage.

  In scenarios such as vehicle entrance and exit management, visitor registration, facial recognition, and campus video surveillance, it is difficult to implement linkage management due to the complexity of data collection and centralization.
- The status of important devices cannot be monitored remotely in real time.

  For example, faulty street lamps cannot be alerted and must be manually repaired in a traditional campus, resulting in passive maintenance.

ROMA has a complete set of integration solutions involving devices, data, and services to help enterprises build smart campuses.

- **Efficient interconnection with various devices from different vendors**

  Information about devices from different vendors, such as cameras, turnstiles, and air conditioners, is sent to ROMA LINK using the standard MQTT protocol. In addition, LINK is connected to multiple IoT platforms, eliminating the need to collect data from each platform separately.
- **Data base construction for providing standard data services**

  FDI and MQS quickly integrate all data and open the data to different backend services of an enterprise. For example, facial data in turnstile systems, device status in video surveillance systems, and switch-on/switch-off and device information in street lamp

systems are transmitted to backend services in real time or in asynchronous mode for analysis and linkage management.

In addition, the high scalability design provided by ROMA supports huge data transmission and storage on the campus network, improving data transmission efficiency.

- **Integration of IT, OT, and AI for building an intelligent operation center**

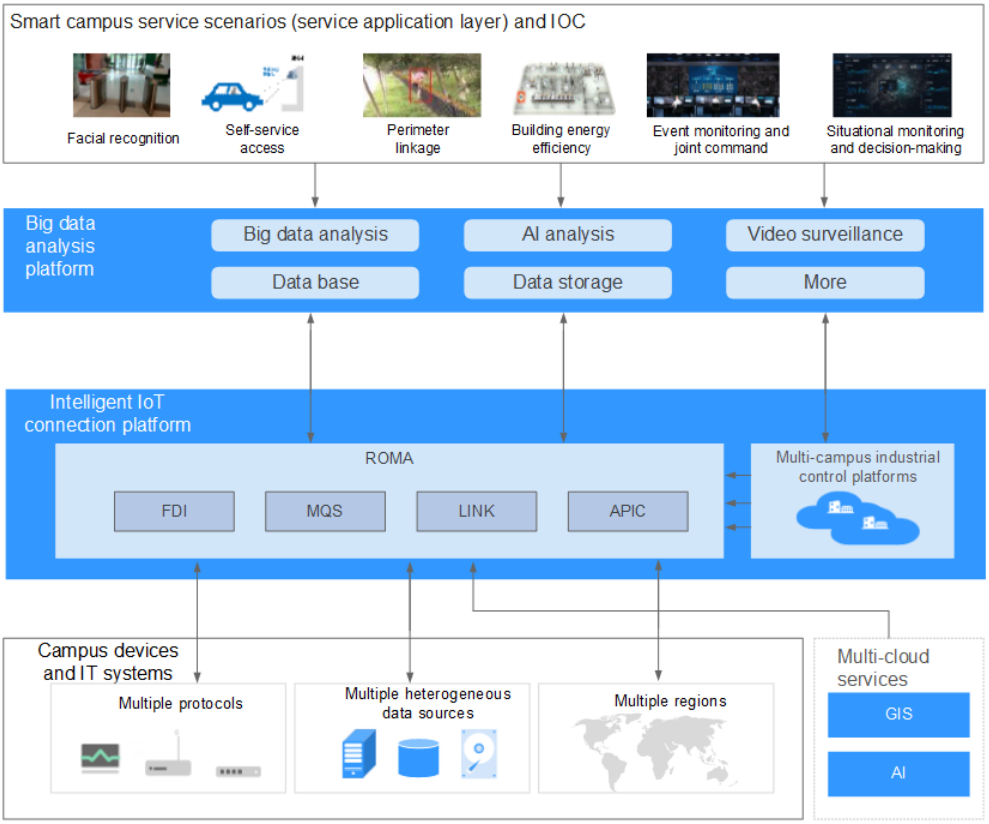
  ROMA provides a channel for data integration and sharing. Enterprises, then, can use the enterprise-grade AI, video analysis, and big data services provided by HUAWEI CLOUD to build a smart campus.

  For example, enterprises can track and analyze behavior of employees in a campus based on data such as vehicle entrance and exit, turnstile, and video surveillance.

- **Centralized and distributed architecture for supporting campus services**

  Enterprises holding large campuses often need to manage multiple campuses. The centralized and distributed architecture of ROMA helps these enterprises integrate data from multiple campuses onto the same platform and assists them in managing the distributed and centralized operations based on actual conditions.

Figure 2-1 Smart campus integration



2.2 Industrial Internet Integration

There are several typical problems in the digital transformation of the manufacturing industry:

- Difficulty in integrating device and environment data

  To monitor and manage production devices of various brands and types in real time, device and environment data need to be collected and uploaded. However, such devices use different data formats and database standards, which makes it difficult to integrate device and environment data.

- Difficulty in preventing device faults

  In a factory, any machine fault may have a huge impact on the entire assembly production line.

- Difficulty in optimizing production strategies and decision-making of enterprises

  Different formats of collected data result in difficult data analysis. Therefore, it is a challenge for enterprises to optimize existing production strategies based on the collected data and to determine whether to execute new production strategies.

Leveraging the HUAWEI CLOUD enterprise-grade big data analysis solution, ROMA helps the manufacturing industry transform to IoT integration through data collection and integration, and finally achieves the "smart" vision.

- **Device digitalization and integration**

  ROMA uses different methods, such as MQTT, gateway, and LINK Site, to connect various types of devices to enterprise backends, implementing bidirectional communication.

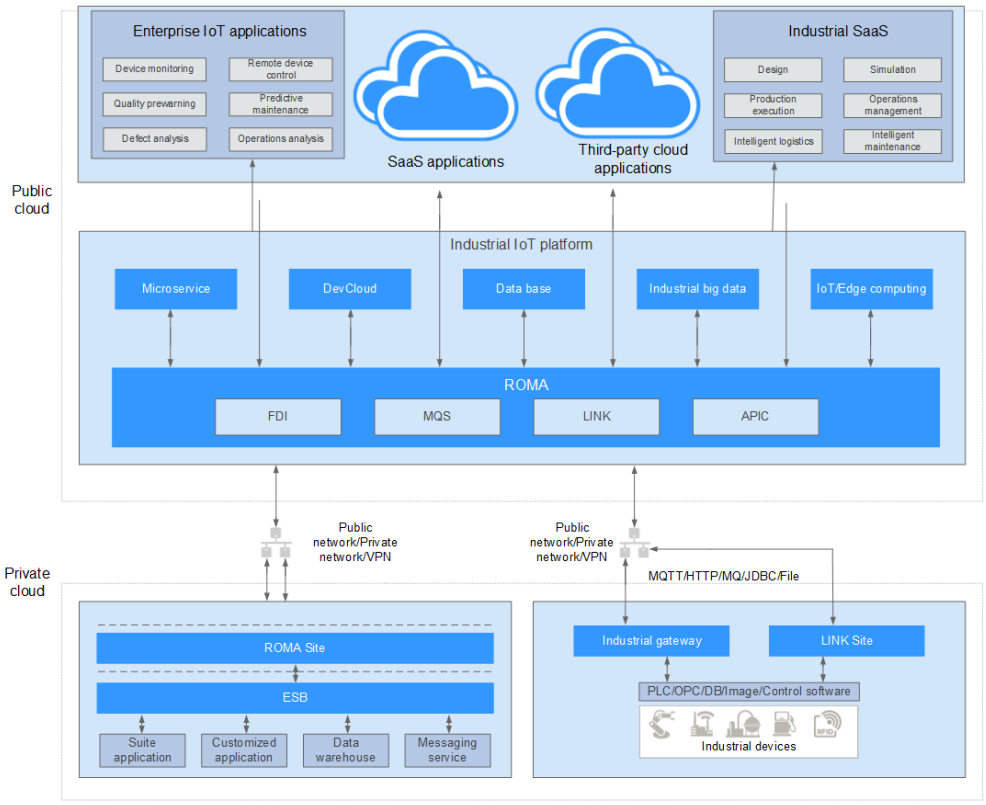
- **Fault prediction and alarming**

  Information about all devices is integrated on the ROMA console for real-time monitoring and prewarning analysis. Once parameters of a device become abnormal, ROMA generates an alarm on the console and notifies the owner of repairing the device. If the real-time status of a device deviates from the normal data range, a notification is sent to device maintenance personnel to repair the device in a timely manner.

- **Data storage and analysis**

  ROMA FDI imports the data generated by the industry SaaS to ROMA, collects and transmits the data to **MapReduce Service** of HUAWEI CLOUD, helping enterprises analyze big data and optimize production strategies.

Figure 2-2 Industrial Internet integration



2.3 Corporate Group Integration

The integration between the corporate headquarters and its branches and between the corporate group and its partners faces bottlenecks:

- Geographical differences

  The headquarters, branches, and partners are located in different regions and use different time zones. This reduces the timeliness and reliability of data.

- Different cloud services

  The cloud services used by the headquarters, branches, and partners are different. Therefore, it is difficult to invoke different cloud services.

- Network differences

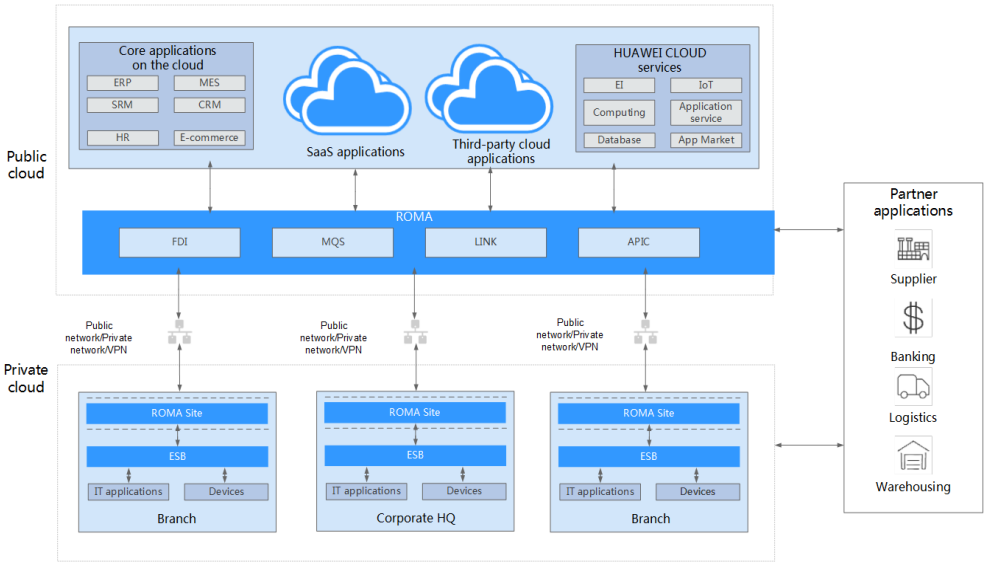
  The networks used by the headquarters, branches, and partners are different. Therefore, interconnection between the public networks, private networks, and VPNs is difficult.

ROMA helps corporation groups implement integration between the headquarters and branches and between the groups and their partners. As shown in **Figure 2-3**, ROMA supports the following scenarios:

- **Cross-regional integration**: The headquarters, branches, and partners located in different regions transmit their device information, data, and messages to ROMA. ROMA performs operations such as device information visualization, alarm monitoring, data conversion, and message transmission to streamline regional restrictions, implement integration and governance for regional businesses and share group information, ensuring the reliability of service integration.

- **Cross-cloud integration**: APIC converts SaaS applications and third-party cloud applications into API data. Then, enterprises call these APIs to integrate different cloud applications, ensuring seamless interconnection between services on the cloud.
- **Cross-network integration**: ROMA Site and inter-cloud ROMA are used to implement secure cross-network interconnection with partners' service systems. Enterprises upload data and information required by partners to ROMA. ROMA then converts the data formats and integrates data based on the partners' requirements. After an enterprise integrates data and messages, partners can access ROMA to obtain related information.

Figure 2-3 Corporate group integration



Enterprise integration through ROMA brings the following benefits to enterprises:

- Builds a unified platform for managing multiple cloud services and applications, simplifying management processes and helping enterprises achieve digital transformation.
- Enables information sharing between headquarters, branches, and partners.
- Supports large-scale integrated services, distributed deployment, automatic scaling, and low latency, ensuring service performance and reliability.

3 Functions

3.1 LINK

LINK is a device integration component of ROMA. It uses the standard MQTT protocol to connect devices, helping enterprises quickly and easily manage devices on the cloud. LINK provides the following functions:

- **Publishing and subscribing to messages**

  LINK supports the standard MQTT protocol. Enterprises can use open-source device SDKs based on this MQTT protocol to easily connect devices to the cloud for message publishing and subscription.

- **Message exchange between devices and backend applications**

  You can configure a rule engine on the LINK console to enable a device to communicate with other devices, backends, and other HUAWEI CLOUD products.

  LINK supports rule engines to forward data to MQS. Third-party services obtain data through MQS to implement asynchronous message communication between devices and third-party services.

- **Low-latency access for massive numbers of devices**

  LINK supports horizontal expansion of Broker and long connections of millions of devices.

- **Two-way synchronization between devices and applications**

  LINK supports TSL definition and binds the **TSL** to a **device shadow**. This allows users to implement two-way synchronization of configuration data and status data between devices and applications.

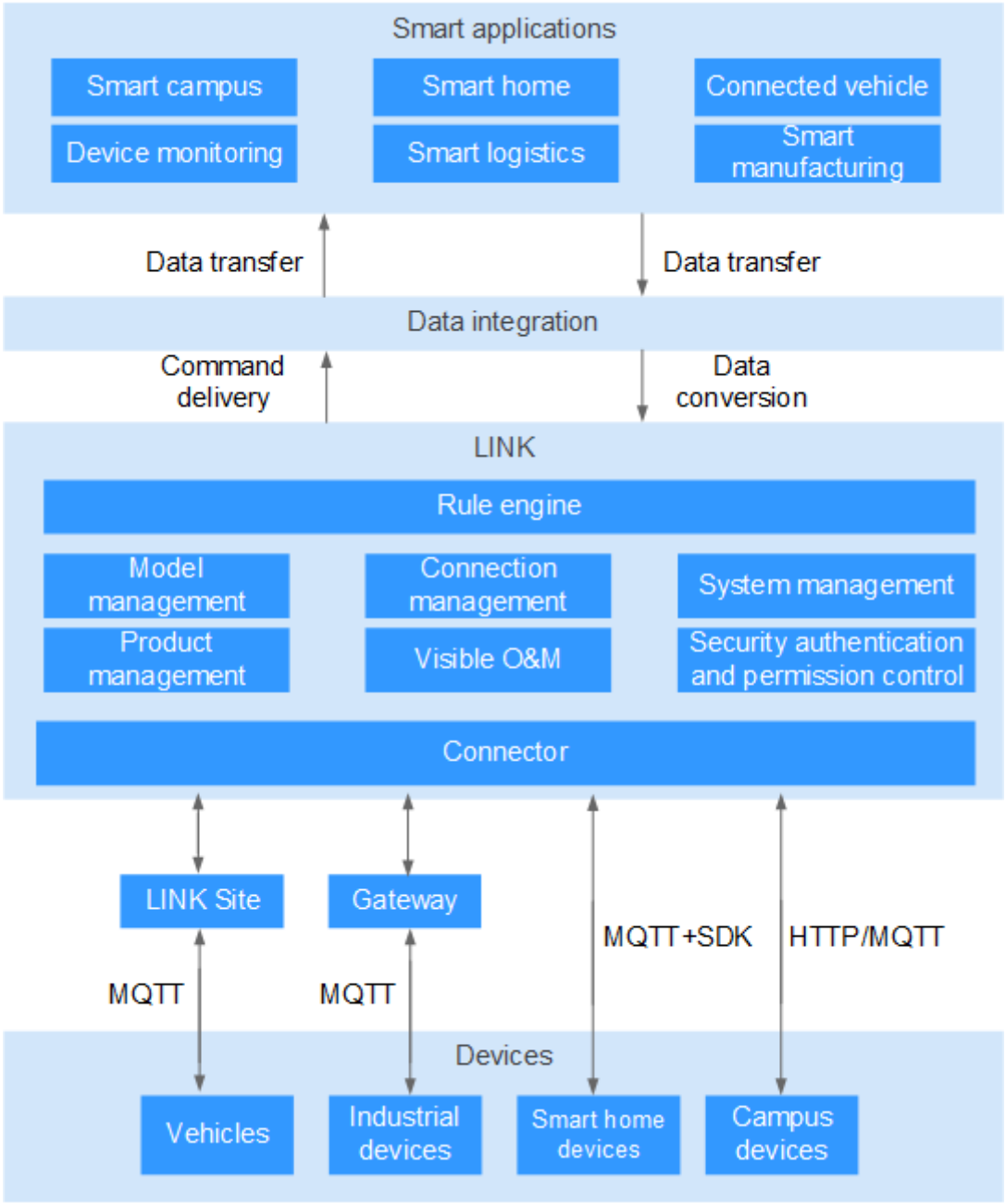
  On the one hand, users can set configuration parameters to the device shadow through APIs. When a device is online or goes online, the configuration parameters can be obtained from the device shadow.

On the other hand, devices can report their statuses to the device shadow. When a user queries the device status, the user only needs to query the device shadow instead of directly communicating with the device.

- **Secure information transmission**

  LINK provides authorization certification for devices and applications and bidirectional binding authorization for topics to ensure device security and uniqueness. It provides TLS-based data transmission channels for secure message transmission.

Figure 3-1 LINK process



3.2 MQS

Message Queue Service (MQS) is an enterprise-grade message middleware developed from Huawei's global IT scenarios. As an integration component of ROMA, MQS uses the unified message access mechanism based on the Kafka protocol to provide enterprises with secure, cross-network communication of cloud and on-premises applications.

MQS uses Kafka (community edition) as a message queue engine. It provides basic and advanced functions to offer a unified message channel for enterprise data management. The basic functions include message publishing and subscription, topic management, user permission management, resource statistics, monitoring and alarming. The advanced functions include message tracking, network isolation, and integration of cloud and on-premises applications.

- **Native Kafka basic functions**

  MQS supports topic management and message publishing and subscription after being connected to the client. It also supports visualized operations on the ROMA console, including topic creation and management, user management, permission configuration, and message query.

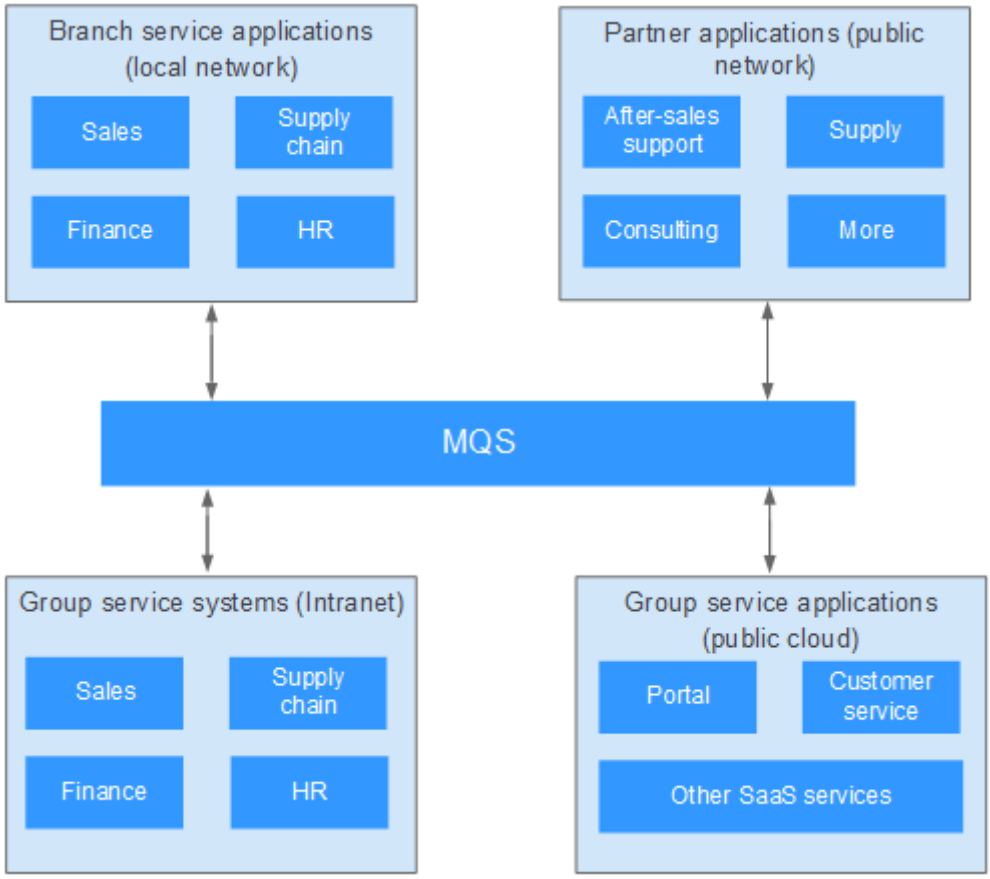
- **Monitoring and alarming**

  MQS allows you to configure monitoring metrics from multiple dimensions, such as instances, nodes, topics, and consumer groups. After monitoring data is reported to the Cloud Eye service, enterprises can view the monitoring data on the Cloud Eye console. In addition, MQS allows you to configure alarm rules so that alarms can be generated if an exception occurs.

- **Message tracking**

  MQS can trace the complete link information of message production and consumption, obtain the status of any message, and provide valid data for troubleshooting.

Figure 3-2 MQS architecture



3.3 FDI

Fast Data Integration (FDI) is a data integration component of ROMA. It supports flexible, fast, and non-intrusive data integration and conversion between multiple data sources. FDI can implement offline data synchronization (such as cross-equipment room, cross-DC, and cross-cloud data integration) and automatic deployment, O&M, and monitoring of integrated data. FDI provides the following functions:

- **Lifecycle management of data integration tasks**

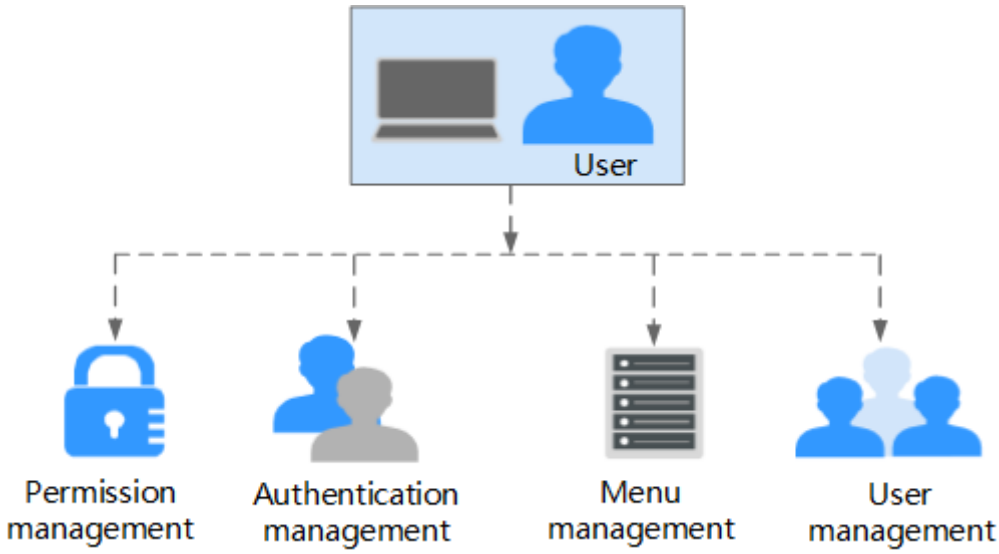
  FDI allows you to modify data integration task information and view running reports, run logs, and status of data integration tasks.

Figure 3-3 Task management

- **System management**

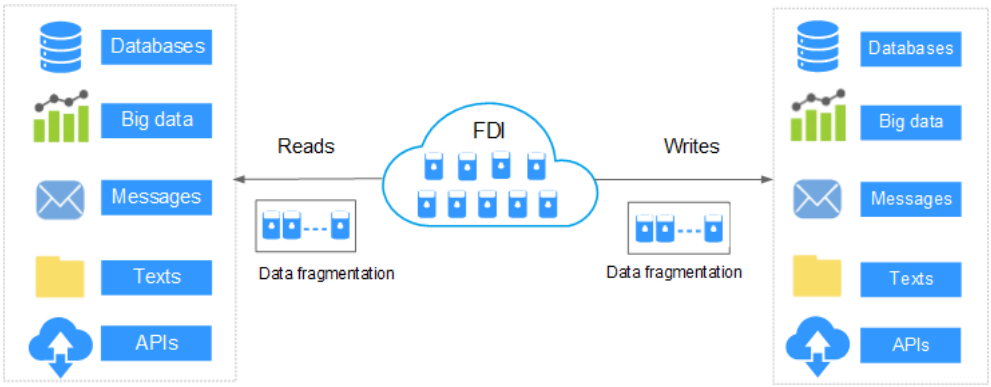
  FDI provides permission, authentication, menu, and user management.

Figure 3-4 System management



- **Flexible data reading and writing**
  - Reads and writes various types of data by fragment, such as MySQL data, text files, messages, and APIs.
  - Supports automatic recovery of tasks when the service is restored after an unexpected interruption occurs.
  - Supports task scheduling, monitoring, and resumable reading.

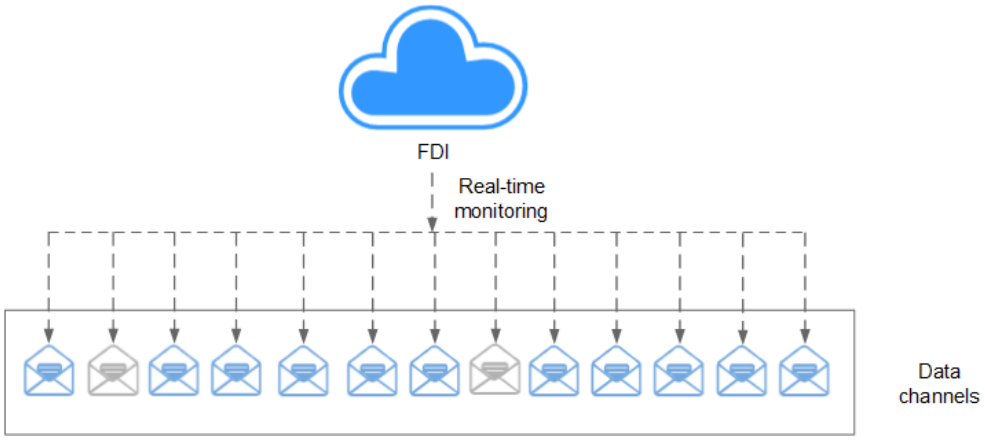
Figure 3-5 Reader and writer plug-ins



- **Reliable data transmission channel**

  FDI can continuously monitor data in data channels and supports concurrent execution of more than 100 threads. It monitors the message queue in real time and writes data to the target queue in real time.

Figure 3-6 Transmission channel for monitored data



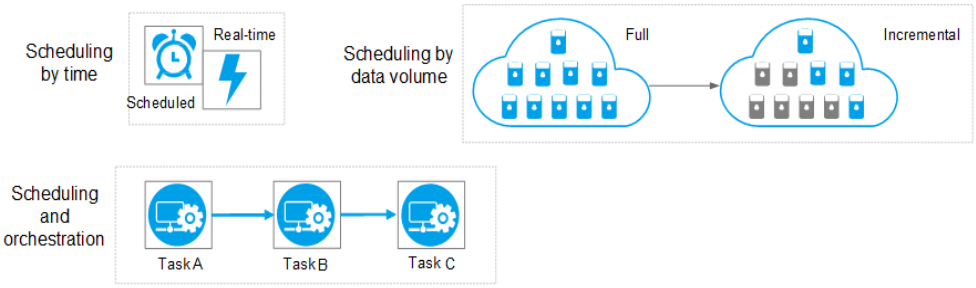
- **Task scheduling**

  FDI provides comprehensive, flexible, and highly available task scheduling services and supports data integration through APIs or messages. It schedules tasks based on time and data volume rules. FDI assigns tasks to the plug-ins based on the task configuration, and monitors and records the task execution status, as shown in **Figure 3-7**.

  Enterprises can select different data integration modes to suit their service requirements.

  - Incremental real-time integration is applicable to scenarios in which data changes need to be monitored in real time, for example, collecting real-time parameters of devices on the production line.
  - Full real-time integration is ideal for scenarios in which all historical data needs to be monitored in real time, for example, collecting statistics on the supplier shipments.
  - Incremental scheduled integration is ideal for scenarios in which data changes need to be monitored for a period of time. For example, enterprises use new production policies to verify whether production efficiency meets expectations.
  - Full scheduled integration is ideal for scenarios in which all historical data needs to be monitored for a period of time, for example, collecting statistics on the number of vehicles entering or leaving a campus during peak and off-peak hours.

Figure 3-7 Task scheduling



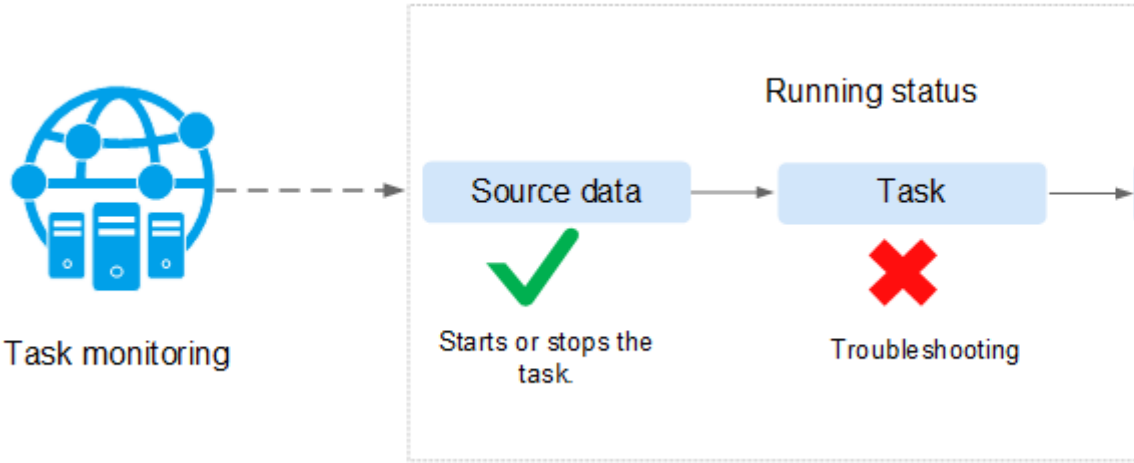
- **Simple debugging tool**

  FDI provides a convenient inline debugging tool to verify whether data sources can be accessed, simplifying data integration development and reducing maintenance costs.

- **Monitoring and alarming**

  FDI monitors the running status of data integration tasks and processes abnormal tasks to ensure normal service operation, as shown in **Figure 3-8**.

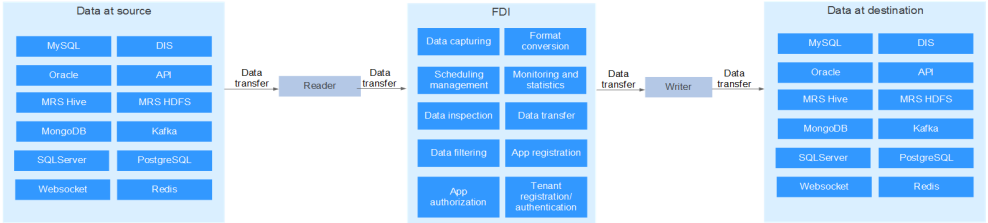
Figure 3-8 Task monitoring



- **Version management**

  An integration task can be published in different environments to meet scenario change and version upgrade requirements.

Figure 3-9 FDI process



3.4 API Connect

API Connect (APIC) is an API integration component of ROMA. It opens data and backend services as APIs to simplify data sharing and service provisioning and reduce the cost on

interconnection between enterprises. APIC provides SDKs and sample code in different programming languages to simplify the process of opening up backend services as APIs. APIC consists of Live Data and API Gateway (APIG).

- Live Data: After an enterprise connects a database to Live Data, Live Data converts the data in the database into APIs and publishes the APIs to APIG for users to call.
- APIG: An enterprise uses APIG to open and call APIs.
  - The enterprise opens APIs to connect backend services to APIG. APIG converts the backend services into APIs and publishes the APIs to the live network environment, gray environment, or customized test environment for users to call.
  - The enterprise calls an API in the specified authentication mode.

NOTE

APIG in this document is different from the **API Gateway** service. APIG is a component of ROMA APIC, whereas API Gateway is an independent cloud service. The APIG component cannot be replaced by the API Gateway service.

APIC provides the following functions:

- **API lifecycle management**

  The lifecycle of an API involves creating, publishing, removing, and deleting the API.
- **Simple debugging tool**

  APIC provides an inline debugging tool to simplify API development and reduce maintenance costs.
- **Version management**

  An API can be published in different environments to meet version upgrade requirements.
- **Request throttling**

  Request throttling controls the maximum number of times an API can be called by a user or an app within a time period.

  The throttling can be accurate to the second, minute, hour, or day.

  Special applications can be configured to be not affected by request throttling policies.
- **Monitoring statistics**

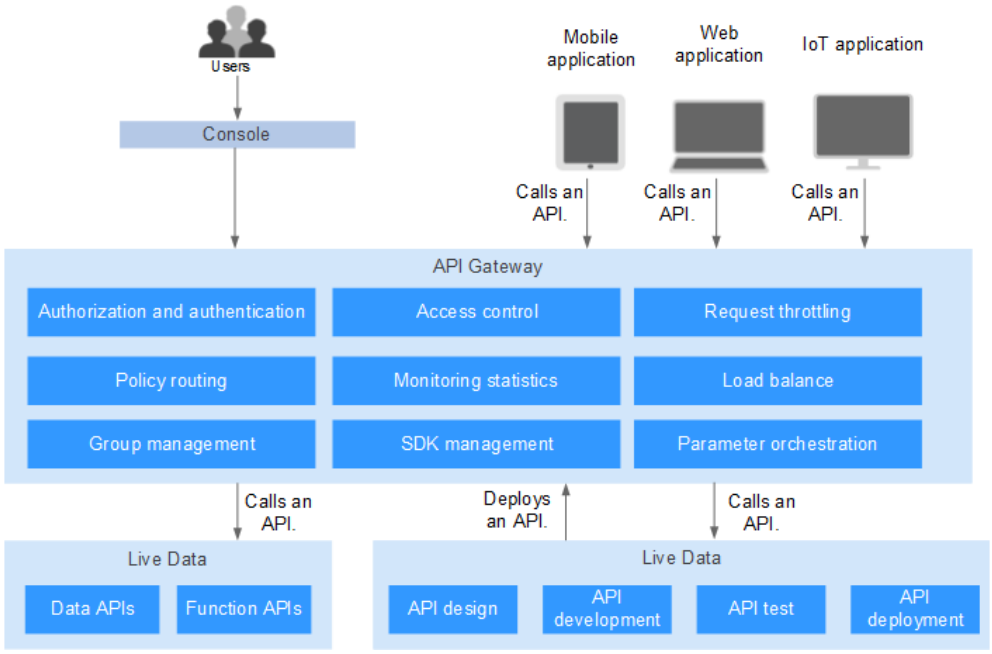
  APIC provides real-time, visualized API monitoring in terms of requests and errors.
- **Environment variable**

  When an API is published to different environments, the specified header parameters and special values are added to the API call request header to distinguish different environments.

  During publication, the variable is replaced with the environment variable value to ensure that the definition of the API does not change.
- **Accessing backend services using domain names**

  You can specify the domain name of a backend service, to which APIC will automatically forward requests.

Figure 3-10 APIC process



4 Basic Concepts

4.1 LINK-related

Model

A model is a set of attributes that abstract the functions of devices on the cloud. Model attributes can be inherited by multiple products. LINK uses the language in JSON format to describe a model.

TSL

A Thing Specification Language (TSL) is a JSON file that digitizes entities in the physical space, such as sensors, industrial devices, on the cloud. A TSL file describes an entity from three dimensions: property (what the entity is), service (what the entity can do), and event (what event information the entity reports). When the three dimensions are defined, the definition of product functions is completed.

Table 4-1 Three dimensions of TSL

Function Type	Description
Property	A property describes the running status of a device, such as the current ambient temperature read by an environment monitoring device. The property supports GET and SET request methods. An application system can initiate a request for reading and setting properties.

Function Type	Description
Service	A service refers to the capability or method that can be invoked by an external device. You can set input and output parameters. Compared with a property, a service can implement more complex business logic through an instruction, such as executing a particular task.
Event	An event is generated when the device is running. It generally contains notification information that needs to be sensed and processed by external systems. An event can contain multiple output parameters. For example, an event can be subscribed to or pushed when the information about a task is complete or the temperature of a device is faulty or an alarm is generated.

Product

A product is a fully functional collection of devices. Each product has a globally unique certificate. All devices under a product can use the same authentication.

Device

Devices constitute a product with certain functions. Each device has a unique certificate under the product and can be connected to ROMA LINK through separate authentication.

Device Shadow

A device shadow is a file in JSON format, which is used to store the current status of devices or applications. Each device has a unique device shadow on the cloud. Regardless of whether the device is connected to the network, enterprises can use the device shadow to obtain and set the device status using the MQTT protocol.

Message Queue

A message queue is a container that receives and stores message files. Different messages in one queue can be retrieved by multiple consumers at the same time as long as the consumers belong to different groups.

Devices use the MQTT protocol to connect to LINK, and the communication content is stored in a message queue.

Topic

A message subject based on which message production and consumption, and other management operations.

A topic is also a message queue model used for publishing and subscribing to messages. A producer publishes a message to a topic. Multiple consumers subscribe to the topic. The producer does not have a direct relationship with the consumers.

For details about topics, see **MQS-related**.

Rule Engine

A rule engine is used to set the rules for subscribing to device topics to obtain data reported by devices. Device data obtained by LINK can be sent to other services for consumption. For example, in the facial recognition scenario, face information obtained by the device is collected by LINK and then sent to the Face Recognition service for comparison and identification.

4.2 MQS-related

MQS uses Kafka as a message engine. The following concepts are described based on the Kafka.

Topic

A message subject based on which message production and consumption, and other management operations.

A topic is also a message queue model used for publishing and subscribing to messages. A producer publishes a message to a topic. Multiple consumers subscribe to the topic. The producer does not have a direct relationship with the consumers.

Producer

A producer is the party that publishes a message to a topic. The purpose of publishing a message is to transfer the message content to other systems or modules so that the other party can process the message as agreed. For example, in the device integration scenario, the device management platform publishes a management instruction as a message to the topic and waits for the device (consumer) to receive and process the instruction.

Consumer

A consumer is the party that subscribes to a message from a topic. Subscription messages are used to process message contents. For example, in the log integration scenario, the monitoring alarm platform (consumer) subscribes to log messages from a topic, identifies alarm logs, and sends an alarm by SMS or email.

Partition

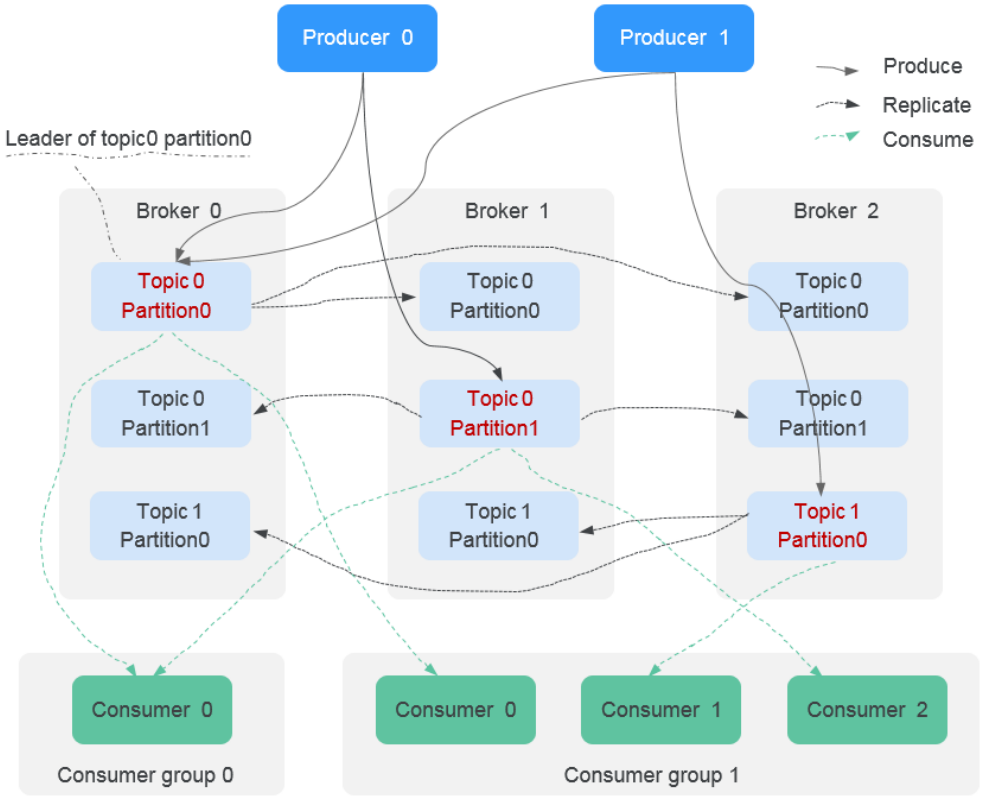
To achieve horizontal expansion and high availability, Kafka divides topics into multiple partitions. Messages are stored in partitions in a distributed manner.

Broker

A broker is a physical node in the Kafka cluster architecture design. Each broker stores all messages of a topic. A partition selects a broker as a leader. The production and consumption of all messages in the partition are completed on the leader. Messages are copied from the leader broker to other brokers (followers).

The topic is a logical concept, whereas the partition and broker are physical concepts. The following figure shows the relationship between partitions, brokers, and topics of Kafka based on the message production and consumption directions.

Figure 4-1 Kafka message flow



4.3 FDI-related

Controller

A controller is an FDI component that controls task and resource scheduling.

Relational Database

A relational database is a data organization consisting of two-dimensional tables and their relationships.

Reader

Reader is a plug-in that can read various types of data such as MySQL, text files, messages, and APIs.

Writer

Writer is a plug-in that can write various types of data such as MySQL, text files, and messages.

4.4 APIC-related

API

An API is a set of predefined functions that encapsulates application capabilities. You can create APIs and make them accessible to users. When creating an API, you need to configure the basic information and the frontend and backend request paths, parameters, and protocols.

API Group

An API group is a set of APIs used for the same service. An API developer can create an API group to manage all APIs in the group.

Data Source

A data source refers to the database or database server used by a database application.

Enterprise services can open up data to subsidiaries or partners for query using APIs instead of directly connecting to the enterprise database. In this scenario, enterprise services at the API backend can be considered as a data source.

Environment

To facilitate API lifecycle management, the usage scope is defined for APIs. This scope is called an environment, such as the API test environment and development environment.

An environment is a stage in the lifecycle of an API. The same APIs can be published in different environments.

To call APIs in different environments, you need to add a specified header parameter to the request header called by the API. The header parameter name is fixed as x-stage. The value of the header parameter is called an environment name, which is used to distinguish different environments.

Request Throttling

Request throttling controls the number of times APIs can be called by a user or an app over a period of time, protecting the backend services.

The throttling can be accurate to the minute and second.

Environment Variable

Environment variables are manageable and specific to environments. You can create variables in different environments to call different backend services using the same API.

Signature Key

A signature key consists of a key and secret. It is used by backend services to verify the APIC identity and ensure the security of the backend service when APIC requests backend services.

After the signature key is bound to an API, APIC sends an API request to the backend service. In this case, the backend service signs the API by using the same method and obtains

the signature result. If the signature is the same as that in the Authorization header sent by APIC, it indicates that the API request comes from APIC and is not a forged request.

Load Balance Channel

A load balance channel is a network security access channel established between APIC and services deployed in a VPC. After network access is enabled, APIC can open up the backend service capabilities deployed in the VPC to third-party users for calling.

Access Control

Access control restricts the API call source from the IP address and account name dimensions.

5 Usage Constraints

- In the open beta test (OBT) phase, only the basic edition is provided for ROMA. Only one instance can be created for each account. More editions will be provided after commercial use.
- In the OBT phase, ROMA can be used for free. If this service continues to be used after the OBT expires, it will be charged according to the price released on the HUAWEI CLOUD official website.

6 Related Services

- Virtual Private Cloud (VPC)

  VPC allows you to create private, isolated virtual networks on HUAWEI CLOUD. You can configure the CIDR block, subnets, and security groups, assign EIPs, and allocate bandwidth for a VPC.

  An instance runs in a VPC. When creating a ROMA instance, enterprises need to create or associate the VPC, subnet, and security group. To enable public network access for an instance, bind an elastic IP address to the instance.

- MapReduce Service (MRS)

  MRS is a HUAWEI CLOUD service that is used to deploy and manage the Hadoop system and enables one-click Hadoop cluster deployment. MRS provides enterprise-level big data clusters on the cloud. Tenants can fully control the clusters and easily run big data components such as Hadoop, Spark, HBase, Kafka, and Storm in the clusters.

  The massive data integrated by enterprises through FDI can be stored and analyzed in MRS of HUAWEI CLOUD.

- Data Ingestion Service (DIS)

  DIS addresses the challenge of transmitting data outside cloud services to cloud services. DIS builds data intake streams for custom applications capable of processing or analyzing streaming data.

  When you use FDI for real-time data integration, you can select the DIS service.

- Data Warehouse Service (DWS)

  DWS is an online data processing database based on the public cloud infrastructure and platform. It can help you mine and analyze massive sets of data.

  When you use FDI for data integration, you can use DWS to process and analyze huge amounts of data.

