Overview

Overview of Cisco Sensor Connect for IoT Services

Enterprise customers need easy ways to onboard, authorize, and control IoT devices onto a single unified infrastructure. The Cisco Sensor Connect addresses this by providing a set of interfaces to:

- Onboard devices into the infrastructure.
- Control and receive events from the devices.

To accomplish this, the Cisco Sensor Connect leverages the following interfaces between the network and applications:

1. **Onboarding Interface**: This interface is used to onboard a device to a network.
2. **Control Interface**: This interface is used for bi-directional communication with a non-IP device.
3. **Telemetry Interface**: This interface is used for streaming telemetry from a non-IP device.

Introduction to Cisco Sensor Connect APIs

The IoT Orchestrator is a Cisco IOx container that is an integral part of the Cisco Sensor Connect solution. You can deploy the IoT Orchestrator on the Cisco Catalyst 9800 Wireless Controllers.

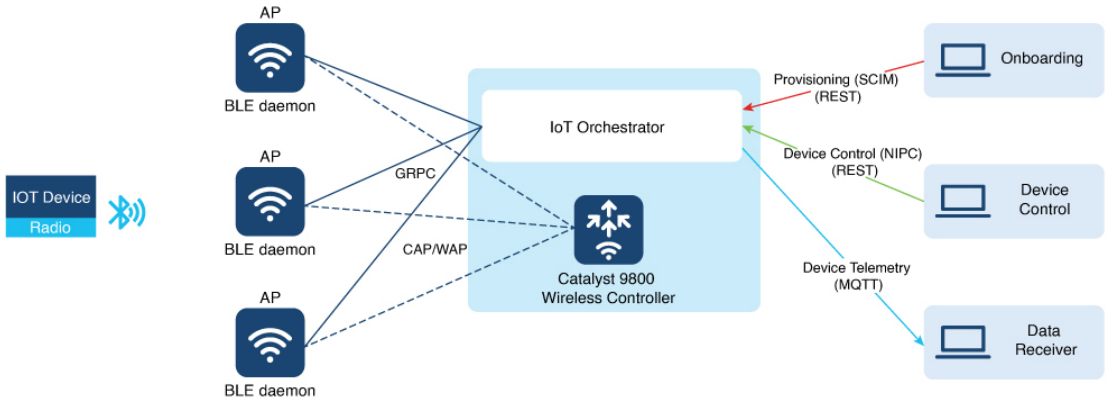
The IoT Orchestrator exposes APIs that allow applications to:

- Introduce devices to enterprise networks
- Control devices

- Receive telemetry from those devices

Note Device control and data reception are not required for IP-based end points because the devices can communicate directly using their IP address.

Figure 1: Cisco Sensor Connect APIs



The IoT Orchestrator exposes the following APIs:

- **Provisioning Interface:**
  - Leverages the IETF System for Cross-Domain Identity Management (SCIM version 2) API, specifically the SCIM for devices model.

    For information on the Device Schema Extensions to the SCIM model, see https://datatracker.ietf.org/doc/draft-ietf-scim-device-model/.

Note The SCIM has a hierarchical schema with a core device schema and extensions.

The SCIM allows an application to:

- Send a SCIM object to a SCIM server (gateway) to create, update, and delete devices in networks.

  For information on the System for Cross-domain Identity Management (SCIM) API, see RFC7643 and RFC7644.

- **Control Interface:**
  - Allows an application to connect to a non-IP device.
  - Enables data exchange with the device.
  - Register topics for streaming telemetry. The IETF draft for this protocol is called the Non-IP Control (NIPC).

- **Telemetry Interface:**

- A streaming data interface with publisher sub-topics of different types, such as:
  - Device broadcasts
  - Device streaming data, or
  - Device connection state.

API Security

The API security in the IoT Orchestrator is configured using either:

- API Key
- Certificate-Based Authentication

IoT Orchestrator in Cisco Catalyst 9800 Wireless Controller

The Cisco Catalyst 9800 Wireless Controller and APs support interfaces as mentioned in the Introduction to Cisco Sensor Connect APIs for BLE devices.

The IoT Orchestrator has two internal components:

- Gateway
- BLE Controller

These components are collocated in a single containerized application.

The IoT Orchestrator works in conjunction with a BLE daemon in the Cisco Catalyst APs. The controller unclears the AP infrastructure to the application and ensures that the device is connected to the most appropriate AP. This means that the application can communicate with a single Gateway and does not have to manage the network topology.

Gateway

The gateway supports the following functionalities:

- **Onboarding (using SCIM) Interfaces:** The gateway acts as the SCIM server.
- **Control (using NIPC) Interfaces:** The gateway provides NIPC interfaces as RESTful interfaces (Gateway is the server).
- **Telemetry (using MQTT) Interfaces:** The gateway functions as MQTT (Gateway is the MQTT broker).

Solution Overview

To communicate with or receive data from a BLE device, an application needs to first onboard the device by creating a SCIM object and send it to the SCIM server (the gateway). The SCIM object uses the device schema and has a BLE extension. If the SCIM call is successful, the gateway generates a unique Device ID and returns the complete SCIM object as registered with the Device ID.

Subsequently, the application uses the Device ID and leverages the control interfaces to communicate with the device or execute a specific NIPC operation.

The NIPC API covers the following types of APIs:

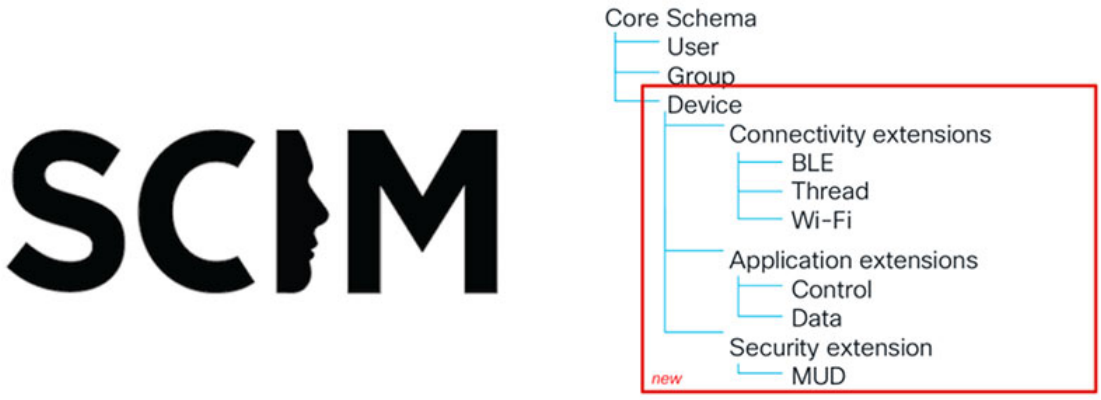
- **Connectivity**: Connect or disconnect, and pair BLE operations.
- **Data**: Read or write, start notifications or indications, discover.
- **Registrations**: Register data application, register topic (either advertisement or connection state).
- **Bulk**: API that allows for bulk operations (multiple operations in one API call).

About System for Cross-Domain Identity Management (SCIM)

The System for Cross-Domain Identity Management (SCIM) is an open standard designed to manage user identity information.

SCIM provides a defined schema to represent users and groups, and a RESTful API to run create, read, update, delete (CRUD) operations on the user and group resources.

Figure 2: SCIM



For more information, see SCIM.

About Non-IP Control (NIPC)

The use cases in building management, healthcare, workplaces, manufacturing, logistics, and hospitality have introduced low-power devices into these environments. These devices typically do not support IP-based interface. Hence, there is a need for gateway functions to allow these devices to communicate with the applications that manage them.

The control interface addresses the device and group objects as ID. Therefore, it is essential to declare a device to the gateway before addressing a NIPC operation in a device. You can address the NIPC operation in a device using SCIM.

Note

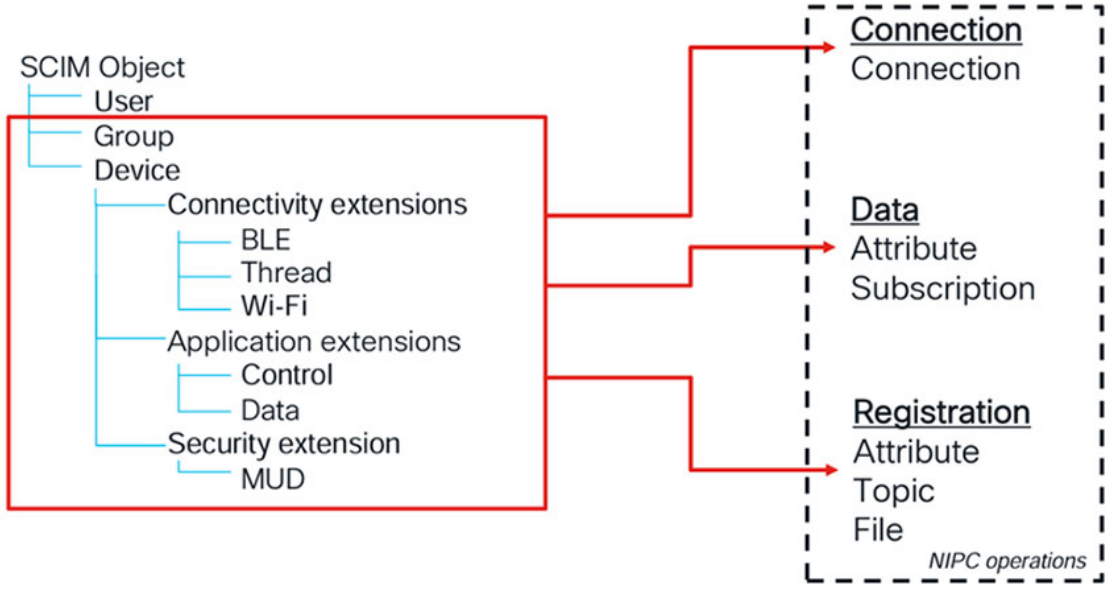
- The NIPC operation can be performed using a device or group ID.
- The existing NIPC APIs were developed while the standard was still evolving, resulting in some aspects being implemented differently. These APIs will be updated to align with the latest draft version of NIPC. Please note that backward compatibility with APIs from version 1.x will not be supported.

The gateway leverages information from the SCIM object to execute a specific NIPC operation. For instance, the keying information in SCIM object may be required to connect to a device.

Advantages of using NIPC

- Enables bi-directional communication with non-IP devices.
- Allows an application to register publisher or subscriber topics to support the data streaming interface.

Figure 3: Non-IP Control (NIPC)



For more information, see NIPC.

Note

SCIM and NIPC APIs have a 75-second timeout.

Each BLE device supports only one pending API request. Issuing another API for a BLE device before completion causes the request to fail.

About Message Queuing Telemetry Transport (MQTT)

The Message Queuing Telemetry Transport (MQTT) is a messaging protocol used to establish communication between multiple devices. This protocol relies on the publish-subscribe model widely used for communication in IoT framework.

Note The MQTT broker is Mosquitto, and the version is 2.0.20. The MQTT message format version is v3.1.

Publish-Subscribe Model

This model separates the client that sends messages (publisher) from the client that receives messages (subscriber). Publisher and subscribers do not need to establish a direct connection and MQTT broker is responsible to route and distribute all messages.

For example, consider the temperature sensor. It connects to the MQTT server as a client and publishes the temperature data to a topic (Temperature), and the server receives the message and forwards it to the client subscribed to the Temperature topic.

Topic

The MQTT protocol route messages based on topic. For information on the MQTT topics, see MQTT Topics and Wildcards: A Beginner's Guide.

About MQTT Message Batching

To improve network efficiency and achieve high MQTT throughput in scale scenarios, the IoT Orchestrator application supports MQTT message batching. As a result, a new protobuf message called **DataBatch** has been introduced. It acts as a wrapper for multiple data subscription messages, and each DataBatch message can contain multiple DataSubscription messages.

Note The partner application that consumes MQTT messages from the IoT Orchestrator must update the data application protocol buffer from the Pre-GA Release prototype to the GA Release prototype.

You need a GitHub account to access the following links:

- For information on the Data Application Protocol Buffer (Pre-GA Release), see the https://github.com/ietf-wg-asdf/asdf-nipc/blob/nipc-asdf-01/proto/data_app.proto.
- For information on the Data Application Protocol Buffer (GA Release), see the https://github.com/ietf-wg-asdf/asdf-nipc/blob/cisco-iot-orchestrator-1.1/proto/data_app.proto.

About Shared Subscriptions

A Shared Subscription is a feature in MQTT that allows multiple clients (subscribers) to share the workload of receiving messages from a single topic. Instead of each subscriber receiving all published messages (as in a normal subscription), MQTT brokers distribute messages among the subscribers in a round-robin or load-balanced manner. This strategy distributes messages among multiple subscribers, preventing any single

subscriber client from being overwhelmed. It also helps achieve high MQTT throughput by preventing a single subscriber client from becoming a bottleneck.

Note The NIPC API for registering the topic, registering the data application, and subscribing to the topic remains unchanged from the Pre-GA version of the IoT Orchestrator. No changes are required for the partner application. The expected MQTT subscriber throughput, based on the 9800 platform on which the IoT Orchestrator application is deployed, is presented in the following table:

Table 1: Platforms and MQTT Subscribers (Recommended)

Platforms	MQTT Subscribers (Recommended)
Cisco Catalyst 9800-L Wireless Controller	2
Cisco Catalyst 9800-40 Wireless Controller	5
Cisco Catalyst 9800-80 Wireless Controller	8
Cisco Catalyst CW9800M Wireless Controller	5
Cisco Catalyst CW9800H1 and CW9800H2 Wireless Controllers	8

About Onboarding BLE Devices

The Internet of Things present a management challenge in many dimensions. One of them being the ability to onboard and manage large number of devices. There are many models to bootstrap trust between devices and network deployments.

The System for Cross Identity Management (SCIM) defines a protocol and a schema to provision users. However, the SCIM is extended to provision devices. The protocol and core schema are designed to allow such extensions.

Why SCIM for devices

The SCIM considers only the information necessary to bootstrap trust so that the device can establish connectivity.

What's Next

For information about the API definition, see **Onboarding BLE Devices using SCIM**.

About Control Operations on BLE Devices

There is a need for non-IP devices to support processes in manufacturing, healthcare, retail, home, and office. Similarly, wireless access points are deployed nearly everywhere, many of which have radios that can transmit and receive different frame types (such as BLE). To integrate both these use cases to leverage a single wireless infrastructure and avoid the need of parallel infrastructure, you need a non-IP device gateway function.

The non-IP device gateway provides the following functions:

- Authenticate and authorize application clients to communicate with devices.
- APIs that allow an application to set up a connection with a device.
- APIs that allow an application to exchange data with a device.
- APIs that allow a device to create registrations in the network.

All these APIs along with the onboarding API (SCIM for devices) allows application to perform a complete set of operations in non-IP devices.

The following are the supported API tags:

Table 2: Supported API Tags

API Tags	Description
connectivity	APIs that allow applications to manage device connections.
data	APIs that allow applications to exchange data with non-IP devices.
registrations	APIs that allow applications to make registrations in the network for devices.
bulk	Compound API that allows applications to combine requests into a single call.

What's Next

For information about the API definition, see **Control Operations on BLE Devices**.

About Data Telemetry from BLE Devices

The NIPC is extensible in two ways:

- **Protocol Extensions**: This allows for extensions to the schema to integrate new non-IP communication protocols.
- **Interface Extensions**: This allows for defining extensions, such as publish/subscribe interface or data streaming interface.

Publish or Subscribe Interface

The publish or subscribe interface or data streaming interface is an MQTT publishing interface. Pub or sub topics can be created and managed by means of the **/register/topic** NIPC element.

What's Next

For information about the API definition, see **Data Telemetry from BLE Devices**.

About Security Model

All RESTful and MQTT interfaces in the Gateway are TLS secured. You will need to configure a server certificate in the Gateway.

Client authentication can be done either using an API key or a certificate. For more information, see the **Uploading Certificate and Key to Open HTTP Server and Listen for APIs** section in the *Cisco Sensor Connect for IoT Services Configuration Guide*.