Use Case 3 - BLE Notification-based Use Cases

Use Case 3 - BLE Notification-Based Use Cases

A BLE device can have multiple services and each service can have multiple characteristics. For BLE notification, you can register the topic and subscribe for a specific service or characteristic ID for which the notification needs to be received. In this case, the best example is to subscribe for BLE GATT notification.

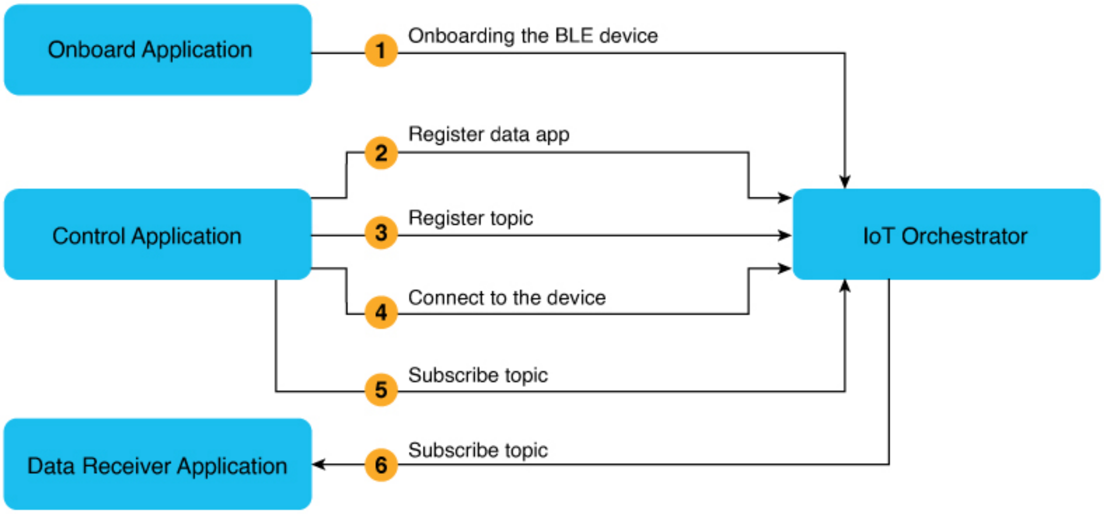
In this use case, you will be able to onboard a device, register a topic, connect to a device, and subscribe for notification.

Here, the sample use case is that of a Pulse Oximeter. A Pulse Oximeter is a medical device that non-invasively measures the oxygen saturation level (SpO2) of a person's blood and pulse rate. The Pulse Oximeter is a small portable device commonly used in hospitals, clinics, and home to monitor patients with respiratory or cardiovascular conditions.

The following are the GATT attributes associated with the Pulse Oximeter:

1. Oxygen level

2. Heart rate



The following workflow covers the sequence of operations:

1. Onboarding the BLE Device.
2. Registering the Data Receiver Application
3. Registering a Topic for Application Events.
4. Registering a Topic for GATT Notifications.
5. Connecting to a BLE Device from an AP.
6. Starting Notifications.
7. MQTT Subscription Messages.
8. Application Events.

Onboarding the BLE Device

API Definition

API	Description
`POST /scim/v2/Devices`	This API is used to onboard the BLE device.

Request Format for Onboarding the Device without Pairing

```
curl -k --location 'https://173.39.84.53:8081/scim/v2/Devices' \
--header 'Content-Type: application/json' \
--header 'x-api-key: 596f0eba1668e6afeb3b25669990b7d84b8bf70408f21465a05e227664763024' \
--data '{
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:Device",
        "urn:ietf:params:scim:schemas:extension:ble:2.0:Device",
        "urn:ietf:params:scim:schemas:extension:endpointapps:2.0:Device"
    ],
    "deviceDisplayName": "BLE Heart Monitor",
    "adminState": true,
    "urn:ietf:params:scim:schemas:extension:ble:2.0:Device": {
        "versionSupport": [
            "5.3"
        ],
        "deviceMacAddress": "CA:2B:5C:EC:95:46",
        "isRandom": false,
        "mobility": false,
        "pairingMethods": [
            "urn:ietf:params:scim:schemas:extension:pairingNull:2.0:Device"
        ],
        "urn:ietf:params:scim:schemas:extension:pairingNull:2.0:Device": null,
        "urn:ietf:params:scim:schemas:extension:pairingJustWorks:2.0:Device": {
            "key": null
        }
    },
    "urn:ietf:params:scim:schemas:extension:endpointAppsExt:2.0:Device": {
        "onboardingUrl": "onboardApplication",
        "deviceControlUrl": [
            "controlApplication"
        ],
        "dataReceiverUrl": []
    }
}'
```

Response Format for Onboarding the Device without Pairing

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Device",
    "urn:ietf:params:scim:schemas:extension:ble:2.0:Device",
    "urn:ietf:params:scim:schemas:extension:endpointapps:2.0:Device"
  ],
  "id": "79ca0f72-4ae2-436a-8bd4-24a2770faa83",
  "deviceDisplayName": "BLE Heart Monitor",
  "adminState": true,
  "urn:ietf:params:scim:schemas:extension:ble:2.0:Device": {
    "versionSupport": [
      "5.3"
    ],
    "deviceMacAddress": "CA:2B:5C:EC:95:46",
    "isRandom": false,
    "pairingMethods": [
      "urn:ietf:params:scim:schemas:extension:pairingNull:2.0:Device"
    ],
    "mobility": false,
    "urn:ietf:params:scim:schemas:extension:pairingNull:2.0:Device": {},
    "urn:ietf:params:scim:schemas:extension:pairingJustWorks:2.0:Device": {
      "key": 0
    },
    "urn:ietf:params:scim:schemas:extension:pairingPassKey:2.0:Device": {
      "key": 0
    },
    "urn:ietf:params:scim:schemas:extension:pairingOOB:2.0:Device": {
      "key": "",
      "randNumber": 0,
      "confirmationNumber": 0
    }
  },
  "urn:ietf:params:scim:schemas:extension:endpointAppsExt:2.0:Device": {
    "onboardingUrl": "onboardApplication",
    "deviceControlUrl": [
      "controlApplication"
    ],
    "dataReceiverUrl": []
  }
}
```

Request Format for Onboarding the Device with Pairing

```
curl -k --location 'https://173.39.84.53:8081/scim/v2/Devices' \
--header 'Content-Type: application/json' \
--header 'x-api-key: 596f0eba1668e6afeb3b25669990b7d84b8bf70408f21465a05e227664763024' \
--data '{
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:Device",
        "urn:ietf:params:scim:schemas:extension:ble:2.0:Device",
        "urn:ietf:params:scim:schemas:extension:endpointapps:2.0:Device"
    ],
    "deviceDisplayName": "BLE Heart Monitor",
    "adminState": true,
    "urn:ietf:params:scim:schemas:extension:ble:2.0:Device": {
        "versionSupport": [
            "5.3"
        ],
        "deviceMacAddress": "CA:2B:5C:EC:95:46",
        "isRandom": false,
        "mobility": true,
        "pairingMethods": [
            "urn:ietf:params:scim:schemas:extension:pairingJustWorks:2.0:Device"
        ],
        "urn:ietf:params:scim:schemas:extension:pairingNull:2.0:Device": null,
        "urn:ietf:params:scim:schemas:extension:pairingJustWorks:2.0:Device": {
            "key": null
        }
    },
    "urn:ietf:params:scim:schemas:extension:endpointAppsExt:2.0:Device": {
        "onboardingUrl": "onboardApplication",
        "deviceControlUrl": [
            "controlApplication"
        ],
        "dataReceiverUrl": []
    }
}'
```

Response Format for Onboarding the Device with Pairing

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Device",
    "urn:ietf:params:scim:schemas:extension:ble:2.0:Device",
    "urn:ietf:params:scim:schemas:extension:endpointapps:2.0:Device"
  ],
  "id": "79ca0f72-4ae2-436a-8bd4-24a2770faa83",
  "deviceDisplayName": "BLE Heart Monitor",
  "adminState": true,
  "urn:ietf:params:scim:schemas:extension:ble:2.0:Device": {
    "versionSupport": [
      "5.3"
    ],
    "deviceMacAddress": "CA:2B:5C:EC:95:46",
    "isRandom": false,
    "pairingMethods": [
      "urn:ietf:params:scim:schemas:extension:pairingJustWorks:2.0:Device"
    ],
    "mobility": true,
    "urn:ietf:params:scim:schemas:extension:pairingNull:2.0:Device": {},
    "urn:ietf:params:scim:schemas:extension:pairingJustWorks:2.0:Device": {
      "key": 0
    },
    "urn:ietf:params:scim:schemas:extension:pairingPassKey:2.0:Device": {
      "key": 0
    },
    "urn:ietf:params:scim:schemas:extension:pairingOOB:2.0:Device": {
      "key": "",
      "randNumber": 0,
      "confirmationNumber": 0
    }
  },
  "urn:ietf:params:scim:schemas:extension:endpointAppsExt:2.0:Device": {
    "onboardingUrl": "onboardApplication",
    "deviceControlUrl": [
      "controlApplication"
    ],
    "dataReceiverUrl": []
  }
}
```

Registering the Data Receiver Application

You need to register a data receiver application with the app ID “dataApplication” to subscribe to the topic “enterprise/hospital/pulse_oximeter”.

Request Format:

```
curl -k --location 'https://173.39.84.53:8081/control/registration/registerDataApp' \
--header 'Content-Type: application/json' \
--header 'x-api-key: b922958a7b030857b3e40f8d6a4db58f6a2bd65741f47fb120517ff8bb6fb0cb' \
--data '{
    "controlApp": "controlApplication",
    "topic": "enterprise/hospital/pulse_oximeter",
    "dataApps": [
        {
            "dataAppID": "dataApplication"
        }
    ]
}'
```

Response Format:

```
{
  "status" : "SUCCESS",
  "topic" : "enterprise/hospital/pulse_oximeter",
  "reason" : "Data applications successfully registered!"
}
```

Registering a Topic for Application Events

You can register the topic "enterprise/hospital/app_events" with the gateway to subscribe to application events.

API Definition

API	Description
/control/registration/registerTopic	This API is used to register a topic with the gateway to subscribe to GATT notifications from the device.

Request Format

```
{
"technology": "ble",
"controlApp": "controlApplication",
"topic": "enterprise/hospital/app_events",
"ble": {
"type": "application_events"
}
}
```

Response Format

```
{
"status" : "SUCCESS",
"topic" : "enterprise/hospital/app_events",
"reason" : "Application event topics successfully registered!"
}
```

Registering a Topic for GATT Notifications

You can register the topic "enterprise/hospital/pulse_oximeter" with the gateway to subscribe to GATT notifications from the device ID "df553860-975c-4d7f-8c1f-b2996f34e26f", service UUID "1800", and characteristic UUID "2A5E".

API Definition

API	Description
/control/registration/registerTopic	This API is used to register a topic with the gateway to subscribe to GATT notifications from the device.

Request Format

```
{
    "technology": "ble",
    "ids": [
        "79ca0f72-4ae2-436a-8bd4-24a2770faa83"
    ],
    "controlApp": "controlApplication",
    "topic": "enterprise/hospital/pulse_oximeter",
    "dataFormat": "default",
    "ble": {
        "type": "gatt",
        "serviceID": "1800",
        "characteristicID": "2A5E"
    }
}
```

Response Format

```
{
  "status" : "SUCCESS",
  "topic" : "enterprise/hospital/pulse_oximeter",
  "reason" : "Gatt topics successfully registered!",
  "registeredIDs" : [ "79ca0f72-4ae2-436a-8bd4-24a2770faa83" ]
}
```

Connecting to a BLE Device from an AP

API Definition

API	Description
/control/connectivity/connect	This API is used to connect to the BLE device from the AP and discover the services with service UUIDs 1800, 2022, 1822, and 180F.

Request Format

```
{
    "technology": "ble",
    "id": "79ca0f72-4ae2-436a-8bd4-24a2770faa83",
    "ble": {
        "services": [
            {
                "serviceID": "1800"
            },
            {
                "serviceID": "2022"
            },
            {
                "serviceID": "1822"
            },
            {
                "serviceID": "180F"
            }
        ]
    },
    "controlApp": "controlApplication"
}
```

Response Format

```
{
  "status": "SUCCESS",
  "id": "79ca0f72-4ae2-436a-8bd4-24a2770faa83",
  "services": [
    {
      "serviceID": "1800",
      "characteristics": [
        {
          "characteristicID": "6367",
          "descriptors": [
            {
              "descriptorID": "7751"
            }
          ],
          "flags": [
            "read",
            "write",
            "notify"
          ]
        }
      ]
    },
    {
      "serviceID": "2022",
      "handle": 1,
      "characteristics": [
        {
          "characteristicID": "3670",
          "descriptors": [
            {
              "descriptorID": "7929"
            }
          ],
          "flags": [
            "read",
            "write",
            "notify"
          ]
        }
      ]
    },
    {
      "serviceID": "1822",
      "handle": 2,
      "characteristics": [
        {
          "characteristicID": "2198",
          "descriptors": [
            {
              "descriptorID": "5622"
            }
          ],
          "flags": [
            "read",
            "write",
            "notify"
          ]
        }
      ]
    },
    {
      "serviceID": "180F",
      "handle": 2,
      "characteristics": [
        {
          "characteristicID": "2198",
          "descriptors": [
            {
              "descriptorID": "5622"
            }
          ],
          "flags": [
            "read",
            "write",
            "notify"
          ]
        }
      ]
    }
  ],
  "requestID": "251ea636-8708-43b0-b8f8-438332098c9d",
  "reason": "NO_INFO"
}
```

Starting Notifications

API Definition

API	Description
/control/data/subscribe	This API enables the GATT notifications in the device using the device ID, GATT service UUID, and characteristic UUID.

Request Format

```
{
    "technology": "ble",
    "id": "79ca0f72-4ae2-436a-8bd4-24a2770faa83",
    "ble": {
        "serviceID": "1800",
        "characteristicID": "2A5E"
    },
    "controlApp": "controlApplication"
}
```

Response Format

```
{
  "status" : "SUCCESS",
  "id" : "79ca0f72-4ae2-436a-8bd4-24a2770faa83",
  "requestID" : "5aa21160-24be-4bdb-8206-9de0f5c24898",
  "serviceID" : "1800",
  "characteristicID" : "2a5e"
}
```

MQTT Subscription Messages

Once the notification is enabled using the subscribe API, the data receiver application starts receiving the GATT notifications once it subscribes to the registered topic.

```
mosquitto_sub -h localhost -p 41883 -t '$share/group/enterprise/hospital/pulse_oximeter'
-u
dataApplication --pw d5a4c7b1afd862a070514528006f22d4964f6c61ec0e2e0b6c3ebd03c2fbb507

$dbdb70f8-b02d-4733-9d1e-8875fa74bda03baf13f
18002a5e

$dbdb70f8-b02d-4733-9d1e-8875fa74bda03baf13f
18002a5e

$dbdb70f8-b02d-4733-9d1e-8875fa74bda03baf13f
18002a5e

$dbdb70f8-b02d-4733-9d1e-8875fa74bda03baf13f
18002a5e

$dbdb70f8-b02d-4733-9d1e-8875fa74bda03baf13f
18002a5e
```

Note The partner application must run multiple instances of the **mosquitto_sub** command. The number of instances is based on the Cisco Catalyst 9800 Wireless Controller platform on which the IoT Orchestrator is deployed. For more information, see the table.

Application Events

Application events apply to use cases that require BLE device connection. If a BLE device is connected and any of the following scenarios occur:

- IoT Orchestrator application restarts
- Cisco Catalyst 9800 Wireless Controller reloads
- Cisco Catalyst 9800 Wireless Controller HA switchover

Then the partner application must register for the **application_events** topic to receive notifications such as *Application Restarted* or *Application Stopped*.

- *Application Restarted*: The IoT Orchestrator application has successfully restarted.
- *Application Down*: The IoT Orchestrator application has been gracefully shutdown.

Upon receiving an *Application Restarted* notification message, the partner application must issue the "NIPC Connect Request API" for devices that were connected prior to the IoT Orchestrator application restart. Without this action, the connection to those BLE devices will be lost.

Once the application events are enabled using the `registerTopic` API, the data receiver application starts receiving the application events once it subscribes to the registered topic.

```
mosquitto_sub -h localhost -p 41883 -t '$share/group/enterprise/hospital/app_events' -u
dataApplication --pw d5a4c7b1afd862a070514528006f22d4964f6c61ec0e2e0b6c3ebd03c2fbb507

APPLICATION RESTARTED
```