# Mandatory Capabilites for Vendor Device Certification

The device certification process requires the device to follow integration best practices. This means that more fields are mandatory for a certified device compared to platform minium requirements. In the following section, mandatory capabilities and behavior are described. The optional capabilities are part of a different document Optional Capabilities for Vendor Device Certification.

## Currently Testable Device Capabilities of Self-Service Certification Microservice

- [x] Foundation Capabilities [x] c8y_Agent
  - [x] Device Information
    - [x] c8y_IsDevice
    - [x] name
    - [x] type
    - [x] c8y_Hardware
    - [x] c8y_Firmware
    - [x] c8y_RequiredAvailability
    - [x] c8y_SupportedOperations
  - [x] External ID
  - [x] Sending Operational Data
    - [x] Measurements
    - [x] Events
    - [x] Alarms
- [ ] Optional Capabilities
  - [x] Gateways
    - [x] Child Device Types
  - [x] Log File Retrieval
  - [x] Device Configuration
    - [x] Text Based Configuration
    - [x] File Based Configuration
  - [x] Managing Device Software
  - [x] Managing Device Firmware
  - [x] Managing Device Firmware
  - [x] Device Profile
  - [x] Restart
  - [ ] Measurement Request
  - [ ] Shell
  - [ ] Cloud Remote Access
  - [x] Location & Tracking

# Mandatory Capabilities

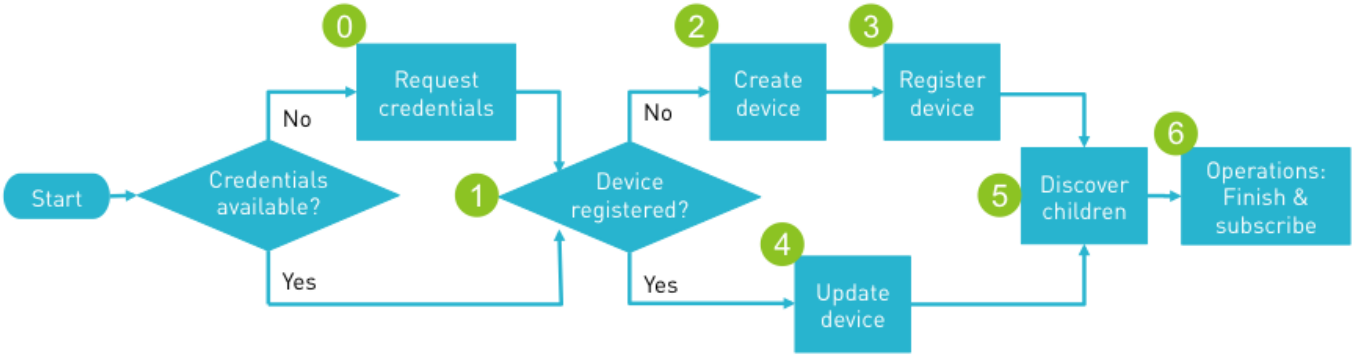
**Status: Reviewed and Ready** The chapter Device Behavior describes how a connector / agent that runs on a device registers to Cumulocity IoT. It must send a mandatory minimum of information to be certifiable covered in the section Mandatory Capability Fragments.

## Device Behavior

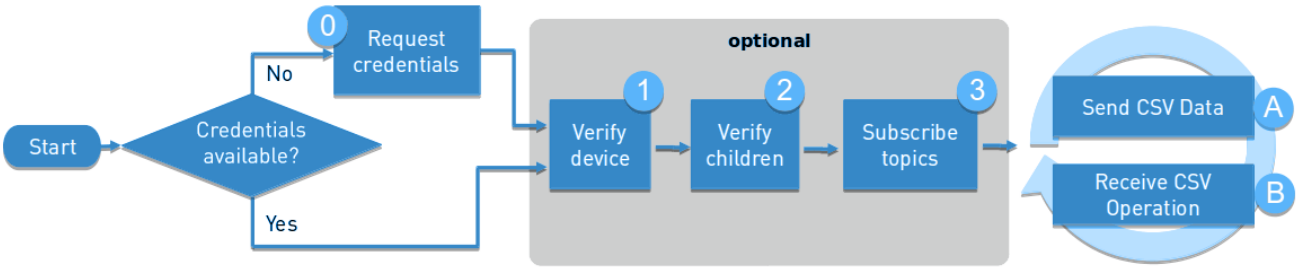
When started, the device follows the process flow defines for REST or MQTT based integration respectively.

**Please read and follow one of these guides:**

For REST based integrations:




For MQTT based integrations (using Smart-Rest 2 is recommended):




**Cypehr Suits:**

Cumulocity IoT fulfills SSL Labs A+ rating and therefor supports exclusively the following cypher suits from release Release 10.10:

- rsa_pkcs1_sha256
- dsa_sha256
- ecdsa_secp256r1_sha256
- rsa_pkcs1_sha384
- ecdsa_secp384r1_sha384
- rsa_pkcs1_sha512
- ecdsa_secp521r1_sha512
- rsa_pss_rsae_sha256
- rsa_pss_rsae_sha384

- rsa_pss_rsae_sha512
- ed25519 ed448
- rsa_pss_pss_sha256
- rsa_pss_pss_sha384
- rsa_pss_pss_sha512

## Mandatory Capability Fragments

For details and examples, compare metadata section of documentation as well as the detail sections below.

| Fragment | Meaning in Device Partner Portal | Mandatory |
|---|---|---|
| `c8y_Agent` | Information about the agent run on the device | Yes |
| `c8y_IsDevice` | Empty fragment. Declares a Managed Object as a Device | Yes |
| `name` | Sets the name of the device used e.g. in 'all devices' and 'device info' views | Yes |
| `type` | Functional type of device e.g. water meter, pump, Gateway, environmental sensor | Yes |
| `c8y_Firmware` | Firmware information about the device | Yes |
| `c8y_Hardware` | Hardware information about the device | Yes |
| `externalIds` | Used to identify a device with a unique information from the physical world | Yes |

Information about one physical device is stored within multiple managed objects. Cumulocity IoT stores all general device information as one managed object in its inventory. The following json structure represents a typical managed object of a device in the inventory (`GET {{url}}/inventory/managedObjects/{{deviceId}}`):

```
"c8y_IsDevice": {},
"com_cumulocity_model_Agent": {},
"name": "edge-agent-eabdcf5d344b4a52a4ffab13fe3d11cb",
"type": "c8y_EdgeAgent",
"c8y_Agent": {
    "name": "DeviceAgent",
    "version": "1.0",
    "url": ""
},
"c8y_Firmware": {
    "name": "raspberrypi-bootloader",
    "version": "1.20140107-1",
    "url": "31aab9856861b1a587e2094690c2f6e272712cb1"
},
"c8y_Hardware": {
    "model": "BCM2708",
    "revision": "000e",
```

```
    "serialNumber": "00000000e2f5ad4d"
}
```

## Agent Information

The fragments `c8y_Agent` must be present in the device managed object stored in the inventory.

**c8y_Agent**

The device certificate will be issued for device defined by: `c8y_Hardware.model`, `c8y_Hardware.revision`, `c8y_Firmware.name`, `c8y_Firmware.version`, `c8y_Agent.name`, and `c8y_Agent.version`.

| Fragment | Mandatory |
|---|---|
| `name` | Yes |
| `version` | Yes |
| `url` | No |

Example structure in device inventory:

```
"c8y_Agent": {
    "name": "myCustomAgent",
    "version": "1.2.34",
    "url": "https//link-to-agent-repo.url"
}
```

## Device Information

The fragments `c8y_IsDevice`, `name`, `type`, `c8y_Firmware`, and `c8y_Hardware` must be present in the managed object of the device stored in the inventory.

**name**

The Cumulocity IoT UI uses the device `name`. `name` sets the name of the device used e.g. in 'all devices' and 'device info' views.

| Fragment | Mandatory |
|---|---|
| `name` | Yes |

Example structure in device inventory:

```
"name": "ExampleDeviceName"
```

**c8y_IsDevice**

`c8y_IsDevice` is an empty fragment that declares a Managed Object as a device.

| Fragment | Mandatory |
|---|---|
| `c8y_IsDevice` | Yes |

Example structure in device inventory:

```
"c8y_IsDevice": {}
```

**type**

The fragment `type` can be interpreted as *device class*. Meaning, devices with the same `type` can receive the same types of configuration, software, firmware, and operations. The Cumulocity IoT UI uses the device `type` often for filtering purposes like sending a software package to all devices of one specific `type`. Based on the device `type` Cumulocity can assign Dashboards to the same `type`.

| Fragment | Mandatory |
|---|---|
| `type` | Yes |

Example structure in device inventory:

```
"type": "c8y_EdgeAgent"
```

**c8y_Hardware**

The device certificate will be issued for device defined by: `c8y_Hardware.model`, `c8y_Hardware.revision`, `c8y_Firmware.name`, `c8y_Firmware.version`, `c8y_Agent.name`, and `c8y_Agent.version`. These fragments will also be used in future versions of Device Partner Portal (display one "Device" entry in the overview device list per `c8y_Hardware.model` and a dropdown in the device detail view for each `c8y_Hardware.revision`).

| Fragment | Meaning in Device Partner Portal | Mandatory |
|---|---|---|
| `model` | Device in list view | Yes |
| `revision` | Dropdown inside device detail view to select device revision or version | Yes |
| `serialNumber` | Not used in Device Partner Portal | Yes |

Example structure in device inventory:

```
"c8y_Hardware": {
    "model": "BCM2708",
    "revision": "000e",
```

```
    "serialNumber": "00000000e2f5ad4d"
}
```

**c8y_Firmware**

The device certificate will be issued for device defined by: `c8y_Hardware.model`, `c8y_Hardware.revision`, `c8y_Firmware.name`, `c8y_Firmware.version`, `c8y_Agent.name`, and `c8y_Agent.version`.

| Fragment | Mandatory |
|---|---|
| `name` | Yes |
| `version` | Yes |
| `url` | No |

Example structure in device inventory:

```
"c8y_Firmware": {
    "name": "raspberrypi-bootloader",
    "version": "1.20140107-1",
    "url": "31aab9856861b1a587e2094690c2f6e272712cb1"
}
```

## External ID

The External ID is displayed by the UI in the tab "Identity". The fragments `externalId` and `type` must be present in the managed object of the device stored in the inventory.

**externalIds**

For details and examples, compare external id section of the documentation.

Used to identify the device in Cumulocity by its unique serial number, MAC, IMEI or similar unique identification string. If you don't want to specify a type, its recommend to use `c8y_Serial`.

| Fragment | Mandatory |
|---|---|
| `externalId` | Yes |
| `type` | Yes |

Example structure in external id managed object (this information is stored in the identity managed object, not in the inventory managed object):

```
"externalIds": [
        {
            "externalId": "edge-agent-eabdcf5d344b4a52a4ffab13fe3d11cb",
            "type": "c8y_Serial"
```

```
        }
    ]
```

## Sending Operational Data

Sending measurements, events, and alarms are basic capabilities of any IoT enabled device. Therefore vendors should aim to support all three. However, there might be some instance where devices only send events (e.g. basic switches) or only measurements (e.g. basic sensor). It is only mandatory to send either measurements, or events, or alarms in order to get certified while it is still recommended to implement all three capabilities.

| Functionality | Content | Mandatory |
|---|---|---|
| Measurements (M), Events (E), Alarms (A) | Information send from the device to the platform | Yes, at least one of the three |

### Measurements

For details and examples, compare measurements section of the documentation. It is only mandatory to send either measurements, or events, or alarms in order to get certified while it is still recommended to implement all three capabilities.

The device creates measurements with the following content:

| Fragment | Content | Mandatory for Measurements |
|---|---|---|
| `source` | Device ID | Yes |
| `type` | Type of measurement | Yes |
| `time` | Date and time when the measurement was made | Yes |
| Measurement Fragment | The category of measurement | Yes |
| Measurement Fragment Series | The name of the measurement series. Contains at least the `value` fragment, optionally the `unit` fragment | Yes |

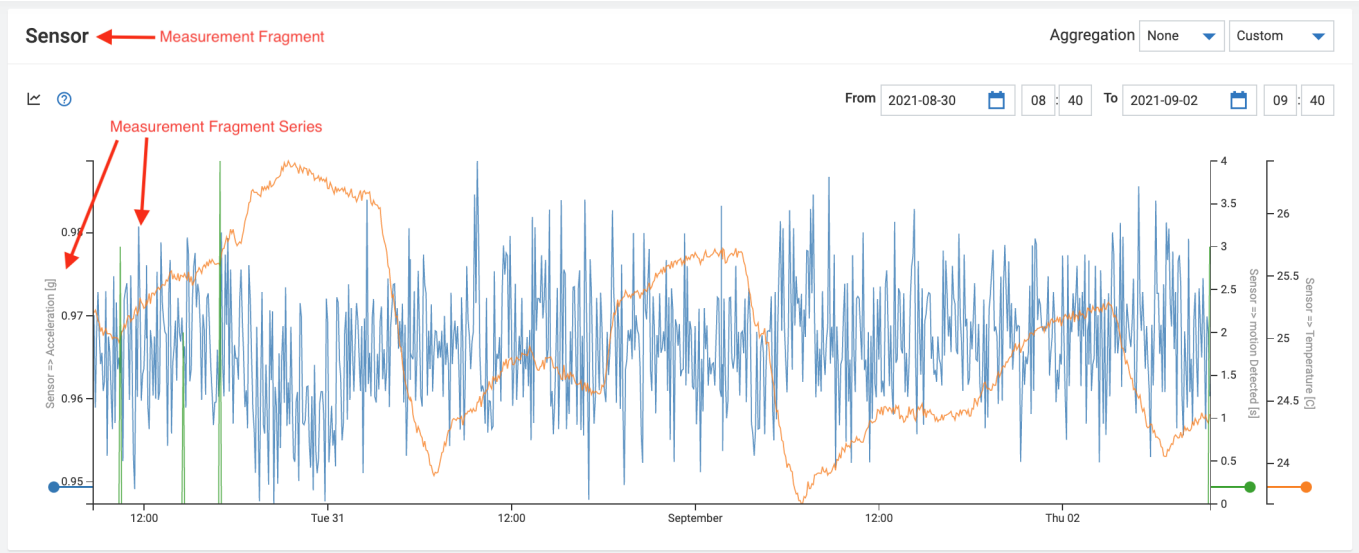
Measurement names should be written in camel-case. Cumulocity IoT UI inserts a blank space between a lower-case and an upper-case letter. Two or more consecutive upper-case letters are not separated with blank spaces. The UI also hides the prefix of a measurement name that is ending with a "_" (underline) symbol.

Example POST body:

```
{
  "source": {
    "id": "251982",
  },
  "type": "c8y_TemperatureMeasurement",
  "time": "2021-10-19T12:03:27.845Z",
```

```
  "c8y_Steam": {
    // Measurement Fragment
    "Temperature": {
      // Measurement Fragment Series
      "unit": "C",
      "value": "100",
    },
  },
}
```

The *Measurement Fragment* and *Measurement Fragment Series* are used in the Cumulocity IoT UI in the following way:




The following *Measurement Fragments* are standard measurement fragments in Cumulocity IoT: `c8y_AccelerationMeasurement`, `c8y_AccelerationSensor`, `c8y_Battery`, `c8y_CPUMeasurement`, `c8y_CurrentMeasurement`, `c8y_CurrentSensor`, `c8y_DistanceMeasurement`, `c8y_DistanceSensor`, `c8y_HumidityMeasurement`, `c8y_HumiditySensor`, `c8y_LightMeasurement`, `c8y_LightSensor`, `c8y_MeasurementPollFrequencyOperation`, `c8y_MeasurementRequestOperation`, `c8y_MemoryMeasurement`, `c8y_MotionMeasurement`, `c8y_MotionSensor`, `c8y_MoistureMeasurement`, `c8y_SignalStrength`, `c8y_SinglePhaseEnergyMeasurement`, `c8y_SinglePhaseEnergySensor`, `c8y_Steam`, `c8y_Temperature`, `c8y_TemperatureSensor`, `c8y_TemperatureMeasurement`, `c8y_ThreePhaseEnergyMeasurement`,`c8y_ThreePhaseElectricitySensor`, `c8y_VoltageMeasurement`,

**Events**

For details and examples, compare events section of the documentation. It is only mandatory to send either measurements, or events, or alarms in order to get certified while it is still recommended to implement all three capabilities.

The device creates events with the following content:

| Fragment | Content | Mandatory for Events |
|---|---|---|
| `source` | Device ID | Yes |
| `type` | Type of event | Yes |

| Fragment | Content | Mandatory for Events |
|---|---|---|
| `time` | Date and time when the event was created | Yes |
| `text` | Description of the event | Yes |

Example POST body:

```
{
  "source": {
    "id": "251982",
  },
  "type": "Intrusion detection",
  "text": "Door sensor was triggered",
  "time": "2021-10-19T12:03:27.845Z",
}
```

**Alarms**

For details and examples, compare alarms section of the documentation. It is only mandatory to send either measurements, or events, or alarms in order to get certified while it is still recommended to implement all three capabilities.

The device creates alarms with the following content:

| Fragment | Content | Mandatory for Alarms |
|---|---|---|
| `source` | Device ID | Yes |
| `type` | Type of alarm | Yes |
| `time` | Date and time when the alarm was created | Yes |
| `text` | Description of the alarm | Yes |
| `severity` | One of the following severities: `CRITICAL`, `MAJOR`, `MINOR`, `WARNING` | Yes |
| `status` | `ACTIVE` or `CLEARED`. If not specified, a new alarm will be created as `ACTIVE`. The state `ACKNOWLEDGED` is only set by the user, not by the device. | No |

Example POST body:

```
{
  "source": {
    "id": "251982",
  },
  "type": "Operational State Alarms",
  "text": "Machine stopped unexpectedly with exit reason 3",
  "severity": "MAJOR",
  "status": "ACTIVE",
```

```
  "time": "2021-10-19T12:03:27.845Z",
}
```

##MD file change Log

| Date | Chapter | Severity |
|---|---|---|
| 30/09/2021 | Added MD file change log | minor |
| 22/10/2021 | Added cypher suites information | medium |
| 01/11/2021 | measurements section: Naming convention added; sending operational data: table added with mandatory information; Device Information: com_cumulocity_model_agent mandatory rule changed and externalIds added | medium |
| 03/11/2021 | `com_cumulocity_model_agent` added as mandatory for each optional agent capability that relies on receiving operations; Moved supported child device types to optional capabilites; | major |
| 08/11/2021 | Updated broken links | minor |
| 09/11/2021 | Updated broken links, added Currently Supported Device Capabilities of Self-Service Certification Microservice | minor |
| 10/11/2021 | Added some common measurement names for reference | minor |
| 15/11/2021 | Changed structure | minor |