Demeter Server C2C API

V1.4

NOTICE: THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

Revision History		
Version	Release Date	Description
1.0	2019-12-16	Initial release.
1.1	2019-12-25	1. Add "Install App" API. 2. Add "error" notification when installing/patching IPK file of App to device failed. 3. Add "publisher" column to App information. 4. Modify desired arguments when requesting install/patch API.
1.2	2020-01-22	1. Update URI of "install/patch App" API 2. Add "Get containers information" API 3. Add "Get a specific container information" API 4. Add "Uninstall App" API
1.3	2020-06-11	1. Update device ID rule. 2. Add API to query all of the devices. 3. Add API to query the specific installed App information of specific device. 4. Add API to query all of the installed Apps information of specific device. 5. Update method to PUT when updating App for specific device.
1.4	2020-12-08	1. Update Apache Kafka relative descriptions and definitions

Table of contents

1. Introduction

The purpose of Demeter **A**pplication **P**rotocol **I**nterface is for partners' cloud services to integrate feature of Open-WRT compatible devices efficiently. Therefore, the partners' cloud services can obtain the necessary functions to setup the devices without considering the complex handshaking or communicating flow charts.

Generic Rule Declaration

Each of the API method follows rules listed as below:

1. The generic rules of Cloud2Cloud (C2C) API document were extended from Unified Message Exchange Interface (UMEi) Design Guideline defined by SERCOMM.

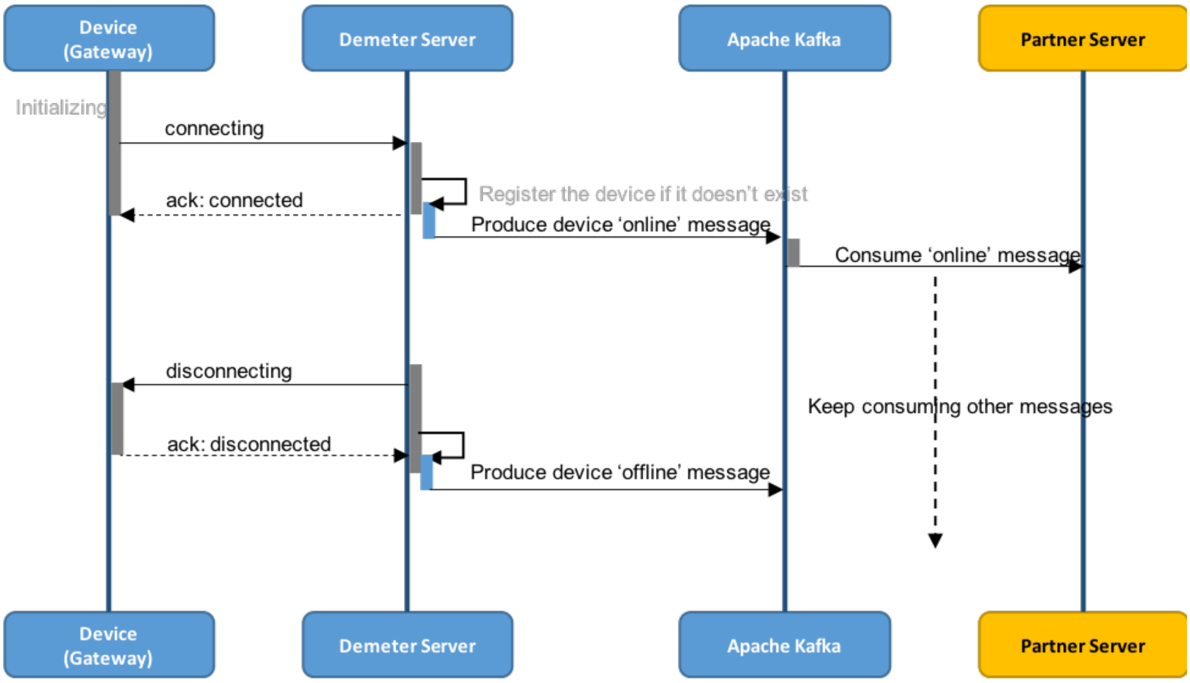
2. The "Request ID" and "Originator ID" are contained within each HTTP response header, named "**X-Request-ID**" and "**X-Originator-ID**".

3. "Receiver ID" is contained within each HTTP response header, named "**X-Receiver-ID**".

2. Demeter Server Notification

2.1. Device Online and Offline Notification

2.1.1. Flow Chart



2.1.2. Message Payload

- Action

  *Indicate the message type. The value will be "**online**" and "**offline**" when producing device online and offline messages.*

- Data Object

Attribute Name	Data Type	Mandatory	Description
serial	string	Yes	Serial number of the device.
mac	string	Yes	MAC address of the device.
model	string	Yes	Model name of the device

- Desire Object

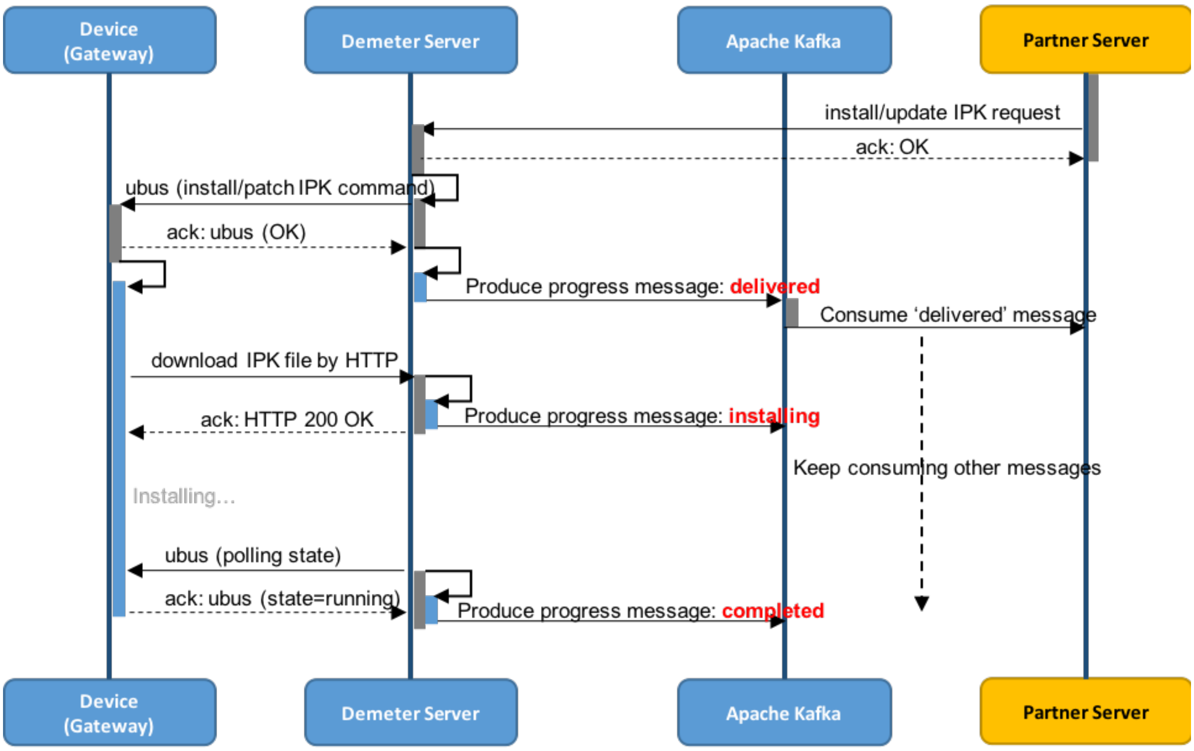
No desire object will be provided.

- An example of device offline notification:

```
{
    "action": "offline",
    "data": {
        "serial": "XX000XX00001",
        "mac": "AABBCCDDEEFF",
        "model": "HG4234B"
    }
}
```

2.2. App Installation Progress Notification

2.2.1. Flow Chart


Device (Gateway)
Demeter Server
Apache Kafka
Partner Server
install/update IPK request
ack: OK
ubus (install/patch IPK command)
ack: ubus (OK)
Produce progress message: delivered
Consume 'delivered' message
download IPK file by HTTP
ack: HTTP 200 OK
Produce progress message: installing
Keep consuming other messages
Installing…
ubus (polling state)
ack: ubus (state=running)
Produce progress message: completed
Device (Gateway)
Demeter Server
Apache Kafka
Partner Server

2.2.2. Message Payload

- Action

  *Indicate the message type. The value will be "**install**" when producing App installation messages.*

- Data Object

Attribute Name	Data Type	Mandatory	Description
serial	string	Yes	Serial number of the device.
mac	string	Yes	MAC address of the device.
taskId	string	Yes	ID of the patch task and was provided by partner server. Please refer to section 3.3.
progress	string	Yes	Progress will be the following values: - **delivered**: command was delivered - **installing**: IPK is downloading and installing - **completed**: upgrade completely - **timeout**: upgrade task timeout - **fail**: error occurred when interacting with device. The details of error will be described in "error" field as below.
error	string	No	Detail error message when installing/patching.

- Desire Object

No desire object will be provided.

- An example of App installation notification:

```
{
    "action": "install",
    "data": {
        "serial": "XX000XX00001",
        "mac": "AABBCCDDEEFF",
        "task": {
            "taskId": "1066943f-839c-4be8-9397-56284790edd4",
            "progress": "installing"
        }
    }
}
```

- An example of App installation notification:

```
{
    "action": "install",
    "data": {
        "serial": "XX000XX00001",
        "mac": "AABBCCDDEEFF",
        "task": {
            "taskId": "1066943f-839c-4be8-9397-56284790edd4",
            "progress": "fail"
        },
        "error": "DEVICE WAS OFFLINE"
    }
}
```

3. Demeter Server API

3.1. Get Devices

3.1.1. API Details

HTTP Verb	URI Pattern	Description
GET	/umei/v1/devices	Get devices' information from Demeter server.

- **{deviceId}**: Device ID consists of its serial number and MAC address.

Action Parameter(s):

Parameter Name	Mandatory	Description
from	No	Please refer to Action Parameter. Default value is **0**.
size	No	Please refer to Action Parameter. Default value is **100**. The maximum value is **500**.
sort	No	Please refer to Action Parameter. Attribute name can be: - **serial:** Sort by serial number of devices. - **mac:** Sort by MAC address of devices. Default is "**serial:asc**"
models	No	Expected model names of devices. Multiple model names need to be separated by comma symbols, e.g. "**HG4234B,HG5244B**". Default is all models.
states	No	Expected devices' current state. Multiple states need to be separated by comma symbol, e.g. "**online,offline**". Default is all states.

Request Payload:

- Data Object

No data object needs to be provided.

- Desire Object

No desire object needs to be provided.

- An example of valid UMEi HTTP Request:

```
GET /umei/v1/devices?from=0&size=100 HTTP/1.1/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-PARTNER /r/n
.../r/n
/r/n
```

Response Payload:

- Data Object

Device Object:

Attribute Name	Data Type	Mandatory	Description
serial	string	Yes	Serial number of the device.
mac	string	Yes	MAC address of the device.
model	string	Yes	Model name of the device.
state	string	Yes	The most current state of the device. State will be as the following values: - **offline** - **online**
firmware	string	Yes	The most current firmware version.
creationTime	string	Yes	Creation timestamp of device's registration.

➢ An example of Successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "meta": {
        "from": 0,
        "size": 2,
        "total": 2,
    },
    "data": [
        {
            "serial": "XX000XX00001",
            "mac": "AABBCCDDEEFF",
            "model": "HG4234B",
            "state": "online",
            "firmware": "3.2.03_T5",
            "creationTime": "2019-12-12T00:11:00+0000",
        },
        {
            "serial": "XX000XX00002",
            "mac": "AABBCCDDEEEF",
            "model": "HG4234B",
            "state": "online",
            "firmware": "3.2.03_T5",
            "creationTime": "2019-12-13T00:11:00+0000",
        },
    ]
}
```

➢ An example of UMEi HTTP response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 500,
        "detail": "INTERNAL SERVER ERROR"
    }]
}
```

3.2. Get Device Information

3.2.1. API Details

HTTP Verb	URI Pattern	Description
GET	/umei/v1/device/{deviceId}	Get device’s information from Demeter server.

- **{deviceId}**: Device ID consists of its serial number and MAC address.

Action Parameter(s):

No action parameter will be provided.

Request Payload:

- ➢ Data Object

No data object needs to be provided.

- ➢ Desire Object

No desire object needs to be provided.

- ➢ An example of valid UMEi HTTP Request:

```
GET /umei/v1/device/XX000XX00001-AABBCCDDEEFF HTTP/1.1/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-PARTNER /r/n
.../r/n
/r/n
```

Response Payload:

- ➢ Data Object

Device Object:

Attribute Name	Data Type	Mandatory	Description
serial	string	Yes	Serial number of the device.
mac	string	Yes	MAC address of the device.
model	string	Yes	Model name of the device.
state	string	Yes	The most current state of the device. State will be as the following values: - **offline** - **online**
firmware	string	Yes	The most current firmware version.
creationTime	string	Yes	Creation timestamp of device’s registration.

- ➢ An example of Successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
```

```
/r/n
{
    "data": {
        "serial": "XX000XX00001",
        "mac": "AABBCCDDEEFF",
        "model": "HG4234B",
        "state": "online",
        "firmware": "3.2.03_T5",
        "creationTime": "2019-12-12T00:11:00+0000",
    }
}
```

- An example of UMEi HTTP response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 404,
        "detail": "DEVICE CANNOT BE FOUND"
    }]
}
```

3.3. Get Installed Apps Information of Specific Device

3.3.1. API Details

HTTP Verb	URI Pattern	Description
GET	/umei/v1/device/{deviceId}/apps	Get device's installed Apps information from Demeter server.

- **{deviceId}**: Device ID consists of its serial number and MAC address.

Action Parameter(s):

No action parameter will be provided.

Request Payload:

- Data Object

No data object needs to be provided.

- Desire Object

No desire object needs to be provided.

- An example of valid UMEi HTTP Request:

```
GET /umei/v1/device/XX000XX00001-AABBCCDDEEFF/apps HTTP/1.1/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-PARTNER /r/n
.../r/n
/r/n
```

Response Payload:

- Data Object

Installed App Object:

Attribute Name	Data Type	Mandatory	Description
appName	string	Yes	Name of the App.
publisher	string	Yes	Publisher name of the App.
appId	string	Yes	ID of the App.
status	string	Yes	Status of the install App: - **installed**: The App was installed but is not executing. - **running**: The App was installed and is executing.
version	string	Yes	Installed version of the App. Please refer to Installed Version Object.

Installed Version Object:

Attribute Name	Data Type	Mandatory	Description
versionName	string	Yes	Name of the version.
versionId	string	Yes	ID of the version

- An example of Successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "data": [
        {
            "appName": "helloworld_1_arm",
            "publisher": "sercomm",
            "appId": "52CC495F818A5615289D582EEE7AC825",
            "status": "running",
            "version": {
                "versionName": "1.0",
                "versionId": "22B10A14A77A931AA1FD183D28230D17"
        },
        {
            "appName": "helloworld_2_arm",
            "publisher": "sercomm",
            "appId": "52CC495F818A5615289D582EEE7AC826",
            "status": "installed",
            "version": {
                "versionName": "1.0",
                "versionId": "22B10A14A77A931AA1FD183D28230D18"
        },
    ]
}
```

- ➢ An example of UMEi HTTP response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 404,
        "detail": "DEVICE CANNOT BE FOUND"
    }]
}
```

3.4. Get Installed App Information of Specific Device

3.4.1. API Details

HTTP Verb	URI Pattern	Description
GET	/umei/v1/device/{deviceId}/app/{appId}	Get device’s installed App information from Demeter server.

- **{deviceId}**: Device ID consists of its serial number and MAC address.
- **{appId}**: ID of the installed App.

Action Parameter(s):

No action parameter will be provided.

Request Payload:

- Data Object

No data object needs to be provided.

- Desire Object

No desire object needs to be provided.

- An example of valid UMEi HTTP Request:

```
GET /umei/v1/device/XX000XX00001-AABBCCDDEEFF/app/52CC495F818A5615289D582EEE7AC825 HTTP/1.1/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-PARTNER /r/n
.../r/n
/r/n
```

Response Payload:

- Data Object

Installed App Object:

Attribute Name	Data Type	Mandatory	Description
appName	string	Yes	Name of the App.
publisher	string	Yes	Publisher name of the App.
appId	string	Yes	ID of the App.
status	string	Yes	Status of the install App: - **installed**: The App was installed but is not executing. - **running**: The App was installed and is executing.
version	string	Yes	Installed version of the App. Please refer to Installed Version Object.

Installed Version Object:

Attribute Name	Data Type	Mandatory	Description
versionName	string	Yes	Name of the version.
versionId	string	Yes	ID of the version

- ➢ An example of Successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "data": {
        "appName": "helloworld_1_arm",
        "publisher": "sercomm",
        "appId": "52CC495F818A5615289D582EEE7AC825",
        "status": "running",
        "version": {
            "versionName": "1.0",
            "versionId": "22B10A14A77A931AA1FD183D28230D17",
        }
    }
}
```

- ➢ An example of UMEi HTTP response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 404,
        "detail": "APP WAS NOT INSTALLED"
    }]
}
```

3.5. Get Installable Apps Information of Device Model

3.5.1. API Details

HTTP Verb	URI Pattern	Description
GET	/umei/v1/apps/{model}	Get available Apps information.

- **{model}**: Model name of a device.

Action Parameter(s):

Parameter Name	Mandatory	Description
from	No	Please refer to Action Parameter. Default value is **0**.
size	No	Please refer to Action Parameter. Default value is **100**. The maximum value is **500**.
sort	No	Please refer to Action Parameter. Attribute name can be: - **name** : Sort by name of an Apps. - **creationTime** : Sort by App creation time. Default is "**name:asc**"

Request Payload:

- Data Object

No data object needs to be provided.

- Desire Object

No desire object needs to be provided.

- An example of valid UMEi HTTP request:

```
GET /umei/v1/apps/HG4234B HTTP/1.1/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
```

Response Payload:

- Data Object

Installable App Object:

Attribute Name	Data Type	Mandatory	Description
appName	string	Yes	Name of the App.
publisher	string	Yes	Publisher name of the App.
appId	string	Yes	ID of the App.
creationTime	string	Yes	Creation timestamp of the App.
versions	string	Yes	List of available versions. Please refer to Available Version Object.

Available Version Object:

Attribute Name	Data Type	Mandatory	Description
versionName	string	Yes	Name of the version.
versionId	string	Yes	ID of the version.
creationTime	string	Yes	Creation timestamp of the version.

- An example of successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "data": [
        {
            "appName": "helloworld_1_arm",
            "publisher": "sercomm",
            "appId": "52CC495F818A5615289D582EEE7AC825",
            "creationTime": "2019-12-12T00:01:00+0000",
            "versions": [
                {
                    "versionName": "1.0",
                    "versionId": "22B10A14A77A931AA1FD183D28230D17",
                    "creationTime": "2019-12-12T00:01:20+0000",
                },
                {
                    "versionName": "1.1",
                    "versionId": "219EF635A8DF081EF0505FD4C9F7A516",
                    "creationTime": "2019-12-12T00:01:30+0000",
                },
            ]
        },
    ]
}
```

- An example of UMEi HTTP response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 500,
        "detail": "INTERNAL SERVER ERROR"
    }]
}
```

3.6. Install or Update App

3.6.1. API Details

HTTP Verb	URI Pattern	Description
POST	/umei/v1/device/{deviceId}/app	Install App with specific version to specific device.
PUT	/umei/v1/device/{deviceId}/app	Update the installed App to specific version of specific device.

- **{deviceId}**: Device ID consists of its serial number and MAC address.

Action Parameter(s):

No action parameter will be provided.

Request Payload:

➢ Data Object

No data object needs to be provided.

➢ Desire Object

Attribute Name	Data Type	Mandatory	Description
appId	string	Yes	ID of the available App.
versionId	string	Yes	ID of the available version..
taskId	string	No	An unique ID for identifying the patch progress assigned by partner server.

➢ An example of valid UMEi HTTP Request:

```
POST /umei/v1/device/XX000XX00001-AABBCCDDEEFF/app HTTP/1.1/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "desire": {
        "appId": "2C0C0C020DB3712CDE1BEA0CD11FB822",
        "versionId": "17A29771640C794FB87F02F22664F249",
        "taskId": "fbec031d-59c0-436a-871e-e4cdc673f467"
    }
}
```

Response Payload:

➢ Data Object

No data object will be provided.

➢ An example of Successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
```

➢ An example of UMEi HTTP Response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER /r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 403,
        "detail": "APP CANNOT BE FOUND"
    }]
}
```

3.7. Uninstall (Delete) App

3.7.1. API Details

HTTP Verb	URI Pattern	Description
DELETE	/umei/v1/device/{deviceId}/app/{appId}	Uninstall (delete) App from specific device.

- **{deviceId}**: Device ID consists of its serial number and MAC address.
- **{appId}**: ID of the installed App.

Action Parameter(s):

No action parameter will be provided.

Request Payload:

➢ Data Object

No data object needs to be provided.

➢ Desire Object

No desire object needs to be provided.

➢ An example of valid UMEi HTTP request:

```
DELETE /umei/v1/device/XX000XX00001-AABBCCDDEEFF/app/2C0C0C020DB3712CDE1BEA0CD11FB822 HTTP/1.1/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-PARTNER /r/n
.../r/n
/r/n
```

Response Payload:

➢ Data Object

No data object will be provided.

➢ An example of successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
```

➢ An example of UMEi HTTP response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 403,
        "detail": "APP HAS NOT BEEN INSTALLED"
    }]
}
```

3.8. Start App

3.8.1. API Details

HTTP Verb	URI Pattern	Description
PUT	/umei/v1/device/{deviceId}/app/{appId}/start	Start the installed App of specific device.

- **{deviceId}**: Device ID consists of its serial number and MAC address.
- **{appId}**: ID of the installed App.

Action Parameter(s):

No action parameter will be provided.

Request Payload:

- Data Object

No data object needs to be provided.

- Desire Object

No desire object needs to be provided.

- An example of valid UMEi HTTP request:

```
PUT /umei/v1/device/XX000XX00001-AABBCCDDEEFF/app/2C0C0C020DB3712CDE1BEA0CD11FB822/start HTTP/1.1/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-PARTNER /r/n
.../r/n
/r/n
```

Response Payload:

- Data Object

No data object will be provided.

- An example of successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
```

- An example of UMEi HTTP response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 403,
        "detail": "APP HAS NOT BEEN INSTALLED"
    }]
}
```

3.9. Stop App

3.9.1. API Details

HTTP Verb	URI Pattern	Description
PUT	/umei/v1/device/{deviceId}/app/{appId}/stop	Stop the installed App of specific device.

- **{deviceId}**: Device ID consists of its serial number and MAC address.
- **{appId}**: ID of the installed App.

Action Parameter(s):

No action parameter will be provided.

Request Payload:

- Data Object

No data object needs to be provided.

- Desire Object

No desire object needs to be provided.

- An example of valid UMEi HTTP request:

```
PUT /umei/v1/device/XX000XX00001-AABBCCDDEEFF/app/2C0C0C020DB3712CDE1BEA0CD11FB822/stop HTTP/1.1/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-PARTNER /r/n
.../r/n
/r/n
```

Response Payload:

- Data Object

No data object will be provided.

- An example of successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
```

- An example of UMEi HTTP response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 403,
        "detail": "APP HAS NOT BEEN INSTALLED"
    }]
}
```

3.10. Get All Containers Information of Device

3.10.1. API Details

HTTP Verb	URI Pattern	Description
GET	/umei/v1/device/{deviceId}/lxcs	Get all containers information of the specific device.

- **{deviceId}**: Device ID consists of its serial number and MAC address.

Action Parameter(s):

No action parameter will be provided.

Request Payload:

➢ Data Object

No data object needs to be provided.

➢ Desire Object

No desire object needs to be provided.

➢ An example of valid UMEI HTTP request:

```
GET /umei/v1/device/XX000XX00001-AABBCCDDEEFF/lxcs HTTP/1.1/r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-PARTNER /r/n
.../r/n
/r/n
```

Response Payload:

➢ Data Object

List of Containers' Information:

Attribute Name	Data Type	Mandatory	Description
id	string	Yes	ID of the container.
name	string	Yes	Name of the container.
enabled	boolean	Yes	Container was enabled or not.
resources	string	Yes	Hardware resources usage. Please refer to Resource Object.

Resources Object:

Attribute Name	Data Type	Mandatory	Description
cpu	string	Yes	CPU usage status. Please refer to CPU Object.
memory	string	Yes	Memory usage status. Please refer to Memory Object.
storage	string	Yes	Storage usage status. Please refer to Storage Object.

CPU Object:

Attribute Name	Data Type	Mandatory	Description
usage	number	Yes	CPU usage status in percentage.

Memory Object:

Attribute Name	Data Type	Mandatory	Description
total	number	Yes	Total allocated memory size in byte unit.
free	number	Yes	Available memory size in byte unit.

Storage Object:

Attribute Name	Data Type	Mandatory	Description
total	number	Yes	Total allocated storage size in byte unit.
free	number	Yes	Available storage size in byte unit.

- An example of successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "data": [{
        "id": "security",
        "name": "security",
        "enabled": true,
        "resources": {
            "cpu": {
                "usage": 13.4,
            },
            "memory": {
                "total": 442280,
                "free": 142720
            },
            "storage": {
                "total": 117760,
                "free": 102656
            }
        }
    },
    {
        "id": "webgui",
        "name": "webgui",
        "enabled": true,
        "resources": {
            "cpu": {
                "usage": 2.5,
            },
            "memory": {
                "total": 442280,
                "free": 256360
            },
            "storage": {
                "total": 117760,
                "free": 101520
            }
        }
    }]
}
```

- ➢ An example of UMEi HTTP response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 404,
        "detail": "DEVICE CANNOT BE FOUND"
    }]
}
```

3.11. Get Specific Container Information of Device

3.11.1. API Details

HTTP Verb	URI Pattern	Description
GET	/umei/v1/device/{deviceId}/lxc/{containerId}	Get specific container information of the specific device.

- **{deviceId}**: Device ID consists of its serial number and MAC address.
- **{containerId}**: ID of target container.

Action Parameter(s):

No action parameter will be provided.

Request Payload:

- Data Object

No data object needs to be provided.

- Desire Object

No desire object needs to be provided.

- An example of valid request:

```
GET /rest/v1/device/XX000XX00001-AABBCCDDEEFF/lxc/security HTTP/1.1/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
.../r/n
/r/n
```

Response Payload:

- Data Object
- Containers Information:

Attribute Name	Data Type	Mandatory	Description
id	string	Yes	ID of the container.
name	string	Yes	Name of the container.
enabled	boolean	Yes	Container was enabled or not.
resources	string	Yes	Hardware resources usage. Please refer to Resource Object.

- Resources Object:

Attribute Name	Data Type	Mandatory	Description
cpu	string	Yes	CPU usage status. Please refer to CPU Object.
memory	string	Yes	Memory usage status. Please refer to Memory Object.
storage	string	Yes	Storage usage status. Please refer to Storage Object.

- CPU Object:

Attribute Name	Data Type	Mandatory	Description
usage	number	Yes	CPU usage status in percentage.

- Memory Object:

Attribute Name	Data Type	Mandatory	Description
total	number	Yes	Total allocated memory size in byte unit.
free	number	Yes	Available memory size in byte unit.

- Storage Object:

Attribute Name	Data Type	Mandatory	Description
total	number	Yes	Total allocated storage size in byte unit.
free	number	Yes	Available storage size in byte unit.

- An example of successful UMEi HTTP response without error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "data": {
        "id": "security",
        "name": "security",
        "enabled": true,
        "resources": {
            "cpu": {
                "usage": 13.4,
            },
            "memory": {
                "total": 442280,
                "free": 142720
            },
            "storage": {
                "total": 117760,
                "free": 102656
            }
        }
    }
}
```

- An example of UMEi HTTP response with error:

```
HTTP/1.1 200 OK/r/n
.../r/n
X-Request-ID: 7fe79306-366a-4098-bbb8-9ea5c77c09dc/r/n
X-Originator-ID: FQDN-OF-DEMETER-SERVER/r/n
X-Receiver-ID: FQDN-OF-PARTNER/r/n
.../r/n
/r/n
{
    "errors": [{
        "code": 404,
        "detail": "DEVICE CANNOT BE FOUND"
    }]
}
```

4. Appendix

4.1. Unified Message Exchange Interface (UMEi)

This chapter describe the guideline of SERCOMM cloud platforms API.

4.1.1. Objective

1. Define message format.
2. Keep rules be self-described and as simple as possible.
3. Rules defined here is not especially bound with any specific application layer protocol. How to adapt rules to specific protocol should be described in another documents.

4.1.2. Prerequisite

1. Default naming convention should be fully compliant with lower camel case throughout all UME related documents if does not specify explicitly.
2. Variables defined in this guideline are identified within namespace "http://www.sercomm.com/2018/ume/v1".

4.1.3. Terminology

- **Action Parameter:** Information which are sent with a CRUDN requests to specify execution criteria.
- **Actor:** An actor who is either sending requests or replying response to corresponding requests.
- **Attribute:** An attribute is a piece of information to represent a data or controllable attributes which are applied to resource.
- **CRUDN:** CRUDN actions specify how a resource will be processed, actions manipulate resource in different ways with specified parameters, controllable and data attributes.
- **Entity:** An entity is either a physical device or a software component.
- **Resource:** A resource represents a type of object which is able to be managed by actors which invoke APIs on resources with a specific CRUDN action. Resources are described by data attributes which associates with a kind of object.

4.1.4. Definition

4.1.4.1. Resource

Resource is composed of a set of attributes which represents a physical or logical entity. Resources could be created, read, updated, deleted or control by CRUDN actions with corresponding action parameters and action payload.

Attributes of resource may represent state of device element or controllable device elements. Attributes are categorized into three access authorities for different operation purposes.

4.1.4.1.1. Read-Only

Entities report resource with a set of read-only attributes to inform receiver current state of resource, it could be used as result of action or event notification.

4.1.4.1.2. Write-Only

A Write-only attribute should not appear in response as query result, but it is used in a request to alter state of resource or instruct resource on how to execute operation. An actor may control entities by altering the attributes of resources.

4.1.4.1.3. Read-Write

A Read-Write attribute has the features of both Read-Only and Write-Only.

Writable attributes are controllable elements, actor cannot retrieve state of resource from writable attributes but it is allowed to alter writable resource attributes to directly change the behavior of entity. A resource should contain following common attributes.

Attribute Name	Short Form	Date Type	Must	Read/Write	Description
resourceId	r	string	Yes	Read-Only	A system-wide unique identity to recognize a resource. The fully qualified expression represents a resource could be varied and depends on the application layer protocol. It should be discussed in separate document.
timestamp	t	datetime	Yes	Read-Only	The latest updated time of resource. The value expression should conform with "**YYYY-MM-DDThh:mm:ss.sss±hhmm**" defined by ISO8601, ±hhmm is the zone designator representing time offset from UTC.
version	v	positive number	No	Read-Only	Each resource is given a version number and version number is returned as part of resource information for READ action. It should be increased for Create/Update/Delete actions. It will be used to control the version of the resource the action is intended to be executed against when version attribute is specified.

Additional attributes for specific type of resource may be varied for representation or control purposes and it is beyond the scope of this guidline and should be defined in another documents.

4.1.4.2. Version System

The versioning system implements **O**ptimistic **C**oncurrency **C**ontrol transactions (aka. Optimistic locking) and involves resource manipualation in a distributed system.

By default, internal versioning is used that starts at 1 on created and increments with each update, deletes. The system checks to see if the version number passed to the request is identical to the version of the currently stored resource. If true, the document will be updated and the increase version number by 1. If the value provided is not identical to the stored resource’s version number, a version conflict error will occur and the operation will fail and status code 409 should be replied. **If no version is provided, then the operation is executed without any version checks**.

4.1.5. CRUDN Action

Action indicates a entity how to manipulate specified resources with specific parameters and attributes. A action operation is comppposed of a originator, a receiver, a request message and a optional response message. A originator invokes an action upon a receiver by sending a request message to the receiver, The receiver act on specified action and reply response message to receiver.

Five verbs are defined below for CRUDN actions.

CRUDN Action	Description
CREATE	Create a resource.
READ	Read one or multiple resources.
UPDATE	Update an existing resource or create a ne resource. Update attribute value of resource may imply a device control process.
DELETE	Delete one or multiple resources.
NOTIFY	Send a notification to receiver, no response is expected.

Following example represents a possibility to map HTTP verbs to CRUDN actions mentioned above.

HTTP Verb	CRUDN Action	Collection: /orders	Instance: /orders/{id}
GET	READ	Read a list of orders.	Read the detail of a single order.
POST	CREATE	Create a new order.	—
PUT	UPDATE/CREATE/NOTIFY	—	Full update or create a specific order.
PATCH	UPDATE	—	Partial update a order.
DELETE	DELETE	—	Delete a order.

Two actors will involved each action operation, originator and receiver. A **originator** is responsible to compose **requests**, with specific action assigned, and send it to receivers via specific application layer protocol. And a **recevier** should act on received request and reply originator either processing result or error within **response**.

4.1.5.1. Target URI

An **U**nified **R**esource **I**dentifier is designed for unambiguous identification of resources that specifies the means of acting upon. In order to easily differentiate between resoure URI and action parameters, the default naming convention for URI is compliant with lower snake case.

4.1.5.2. Action Parameter

Action parameters are used to specify the conditions to execute action, entity can specifies query parameters to filter or restrict the range of replied result or other operation instructions. For example, an originator invokes READ action to retrieve information from receiver according to specifiy execution criteria.

Some common parameters defined hereunder are shared among actions, additional parameters may be defined for specific operations should be discussed individually in another document.

Name	Date Type	Must	Description
begin	datetime	No	Specify the begin time of specific time range to restrict the number of queried resources. The value expression should conform with "**YYYY-MM-DDThh:mm:ss.sss±hhmm**" defined by ISO8601, ±hhmm is the zone designator representing time offset from UTC.
end	datetime	No	Specify the end time of specific time range to restrict the number of queried resources. The value expression should conform with "**YYYY-MM-DDThh:mm:ss.sss±hhmm**" defined by ISO8601, ±hhmm is the zone designator representing time offset from UTC.
size	number	No	The number of resources to be returned.
from	number	No	The offset of 1st resource to be returned.
sort	string	No	Sorting to perform, can be either a **attributeName** or **attributeName:asc**,**attributeName:desc** (sequence is important). default order is **asc**.
include	string	No	Set to false to disable retrieval of resource attributes. You can also specify attributes to retrieve part of the resource , e.g. **attributeName *("," attributeName)**.

4.1.5.3. Message Payload

Action payload is information which is mandatory for resource processing with specified action and parameters. Action payload may contain resources, metadata of resource and information to describe what kind data a payload is providing. Payload exists in both **request and response** and discussed in next sections.

4.1.6. Request Payload

Request payload is an object includes information which a originator sends it to receiver in order to instruct which operation should be prcessed accordingly. Request payload may be not required for action and is divided into several categories based on the characteristics of information it contains.

The following table represents the relationshiop between categories and actions.

Information Category	Short Form	Data Type	Create	Read	Update	Delete	Notify
data	d	object	+	—	+*	—	+
desire	ds	object	—	—	+*	—	—
requestId	ri	string	+*	+*	+*	+*	+*
originatorId	oi	string	+*	+*	+*	+*	+*

Legends: +: Mandatory, **+*:** Optional, **—:** Not including

- **data:** Data is a key-value map which includes at least one or multiple resources or arrays of resources to represent readable attributes of resources. Key is the name of resource in either singular or plural form. The data type of value should be array of resources if key is naming in plural and value type is object for singular.

- **desire:** Desire is a key-value map which includes at least one or multiple resource objects or arrays of resource to represent writable attributes of resources.
- **requestId:** An ID to identify a unique request sent from same originator.
- **originatorId:** An unique ID to identify the originator.

4.1.7. Response Payload

Response payload is anobject includes status code and information which a receiver replies originator to inform the result of execution of action. Response payload is divided into several categories based on the characteristics of information it contains.

The following table represents the relationshiop between categories and actions.

Information Category	Short Form	Data Type	Create	Read	Update	Delete	Notify
data	d	object	**+**	**—**	**+***	**—**	**—**
errors	e	array	**+***	**+***	**+***	**+***	**—**
meta	m	object	**—**	**+***	**—**	**—**	**—**
receiverId	rc	string	**+***	**+***	**+***	**+***	**—**
requestId	ri	string	**+***	**+***	**+***	**+***	**—**
originatorId	oi	string	**+***	**+***	**+***	**+***	**—**

Legends: +: Mandatory, **+*:** Optional, **—:** Not including

- **data:** Data is a key-value map which includes at least one or multiple resources or arrays of resources to represent readable attributes of resources. Key is the name of resource in either singular or plural form. The data type of value should be array of resources if key is naming in plural and value type is object for singular.
- **errors:** Errors is an array of error objects which represents the status of executions, an error object contains two attributes.
  - ***code**:* Status code indicates the result of execution (see status code definitions in following chapters).
  - ***detail**:* A description of error to make it more easy to read for human readers.

  In some implementations, e.g. API gateway in Microservice, a response message might be replied by receiver who invokes multiple upstream services and merge them into a single response message. In this case, there might be existing multiple error objects in errors to represent each status of execution.
- **meta:** Meta provides information about data that are encouraged to provided by CRUDN actions for different purposes and helps originators to understand how to manipulate data well.

The following table is the example case which defines the pagination metadata of READ action.

Attribute Name	Short Form	Data Type	Must	Read/Write	Description
from	fr	positive number	Yes	Read-Only	The offest between 1st data item in current response and whole matched data items.
size	sz	positive number	Yes	Read-Only	The number of data items.
total	tt	positive number	Yes	Read-Only	Total number of whole matched data items.

- **receiverId:** An unique ID to identify the receiver.
- **requestId:** Replicated from requestId of request if it is existed.
- **originatorId:** Replicated from originatorId of request if it is existed.

4.1.8. Status Code

Status code are returned in the Response payload to indicate the result of request processing. The status codes are categorized as one of 6 classes as the following table.

Status Class	Code Class	Description
Informational	1XX	The request is successfully received, but the request is still on process.
Success	2XX	The request is successfully received, understood, and accepted.
Redirection	3XX	(Not used in present release)
Originator Error	4XX	The request was malformed by the originator and is rejected.
Receiver Error	5XX	The requested operation cannot be performed due to an error condition at the Receiver.

4.1.8.1. Information Code

Numeric Code	Abbreviation
100	ACCEPTED

4.1.8.2. Success Code

Numeric Code	Abbreviation
200	OK
201	CREATED
202	DELETED
204	UPDATED

4.1.8.3. Originator Error Code

Numeric Code	Abbreviation
400	BAD_REQUEST
401	NOT_SUPPORTED
404	NOT_FOUND
405	ACTION_NOT_ALLOWED
408	REQUEST_TIMEOUT
409	VERSION_CONFLICT

4.1.8.4. Receiver Error Code

Numeric Code	Abbreviation
500	INTERNAL_SERVER_ERROR
501	NOT_IMPLEMENTED
502	BAD_GATEWAY
503	GATEWAY_TIMEOUT