Technical Report

Document Name	IoT mobile App develop project : monitoring presence of people in classroom
Date	2017-11-17
Author	Team : Wanderer 15011091 박혜선 15011095 이유리 15011113 이다솔 15011117 장세영

Contents

1. High Level Architecture and entities ············ 1p

2. Resource tree structure ············ 2p

3. Procedures and call flow ············ 2p
   3.1. Registration ············ 2p
   3.2. Initial resource creation ············ 3p
   3.3. Discovery and retrieval ············ 4p

4. Roles of entities ············ 6p
   4.1. Mobius (IN-CSE) ············ 6p
   4.2. &Cube Thyme (ADN-AE-Gwang / ADN-AE-Yul) ············ 6p
   4.3. Smart phone applications (ADN-AE-App) ············ 6p

5. Procedure ············ 6p
   5.1. Registration and resource creation ············ 6p
   5.2. Discovery and Retrieve ············ 9p

1. High Level Architecture and entities

This clause describes the high level architecture of Detecting the presence of People Project with components represented by the oneM2M entity roles.

In this project :

• The device RaspberryPi and Pi-cam is connected to &Cube Thyme by TAS. The AE which resides in the application and &Cube Dedicated Node is called ADN-AE.

• An IN-CSE (short for Infrastructure Node CSE) is hosted in the server.

For instance, the architecture is show in figure 1-1.

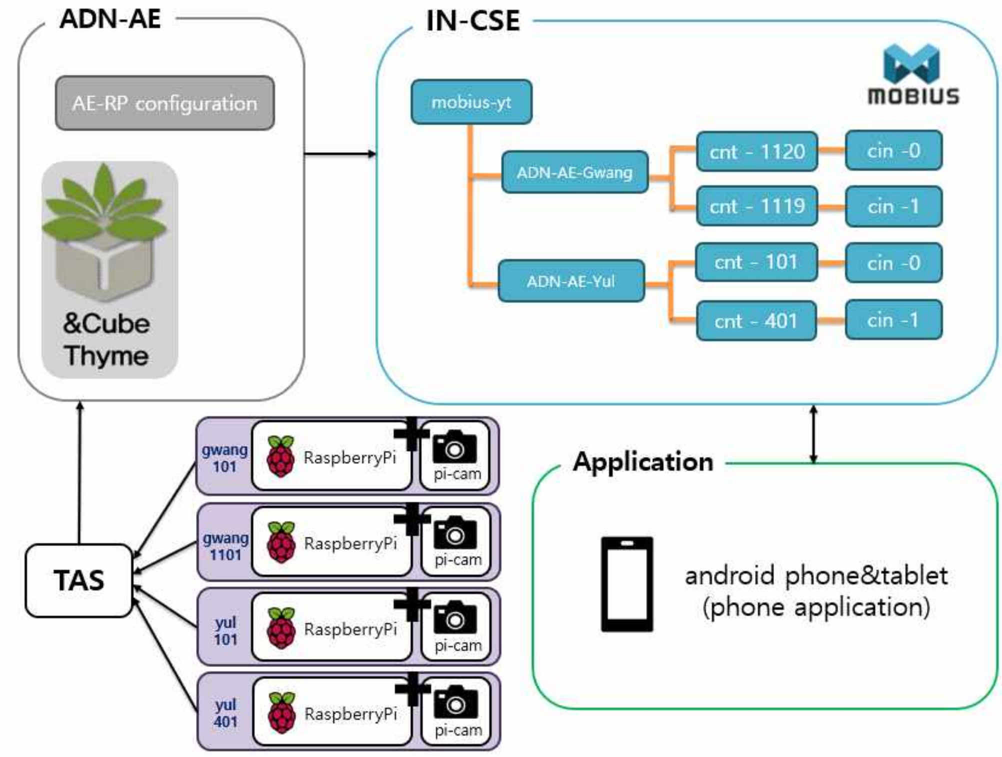

Figure 1-1 : high level architecture of Detecting the presence of people Project

Applications used in the current project are classified as follows :

1) ADN-AE-Gwang : an application that deals with devices(RaspberryPi + pi-cam) in Gwanggaeto through &Cube Thyme and interact with the IN-CSE.

2) ADN-AE-Yul : an application that deals with device(RaspberryPi + pi-cam) in Yulgok through &Cube Thyme and interact with the IN-CSE.

3) Application : a smart phone application embedded in the smart phone device with capabilities to interact directly with IN-CSE and thereby remotely monitor pi-cam's result.

2. Resource tree structure

All of resource tree of this project is show in figure 2-1 :

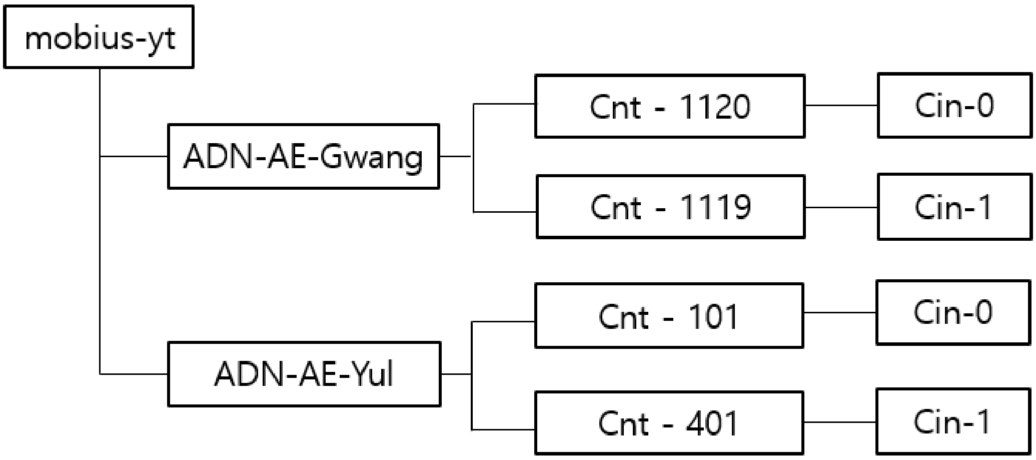

Figure 2-1 : Resouce tree of this project

3. Procedures and call flows

3.1. Registration

The first step is device(RaspberryPi + pi-cam) application registration and smart phone application registration. Device(RaspberryPi + pi-cam) will register applications with &Cube Thyme, and the &Cube Thyme will register with Mobius. The smart phone applications can register with Mobius anytime as needed.

Call flows regarding the registration phase depicted in figure 3.1-1 are ordered as follows :

1) &Cube Thyme configure devices through TAS(ADN-AE-Gwang / ADN-AE-Yul).

2) Device application(ADN-AE-Gwang / ADN-AE-Yul) registers with Mobius(IN-CSE) through &Cube Thyme.

3) Smartphone application(Application) registers with Mobius(IN-CSE).

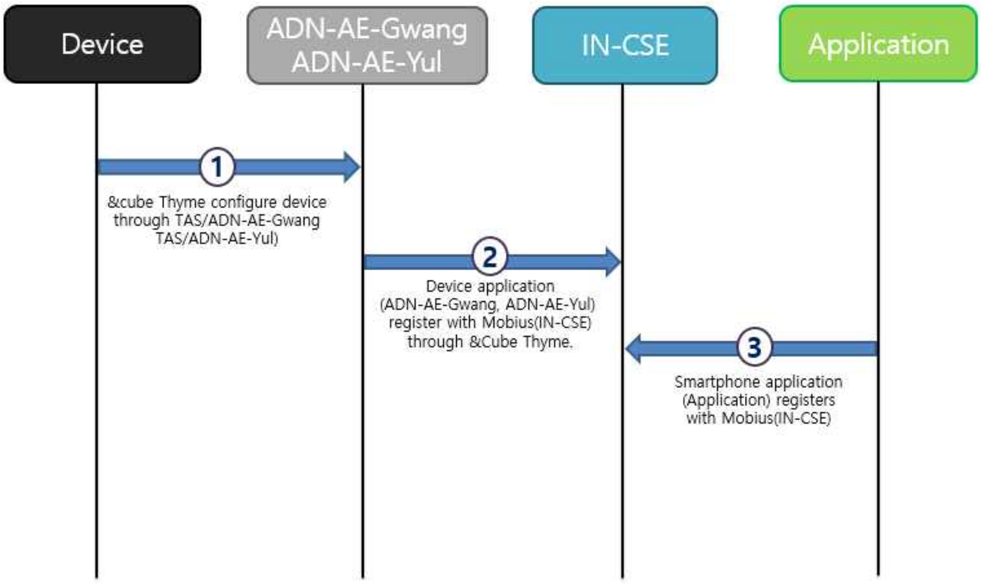
Device
ADN-AE-Gwang
ADN-AE-Yul
IN-CSE
Application
1
&cube Thyme configure device through TAS/ADN-AE-Gwang TAS/ADN-AE-Yul)
2
Device application (ADN-AE-Gwang, ADN-AE-Yul) register with Mobius(IN-CSE) through &Cube Thyme.
3
Smartphone application (Application) registers with Mobius(IN-CSE)

Figure 3.1-1 : Registration phase call flows

3.2. Initial resource creation

After registration, it is necessary to create container resources to store the data from device on the &Cube Thyme. Call flows regarding the initial resource creation phase depicted in figure 3.2-1 are ordered as follows :

Several container resources are created in the Mobius(IN-CSE) to store each of classroom's device data under the registered device application(ADN-AE-Gwang / ADN-AE-Yul).

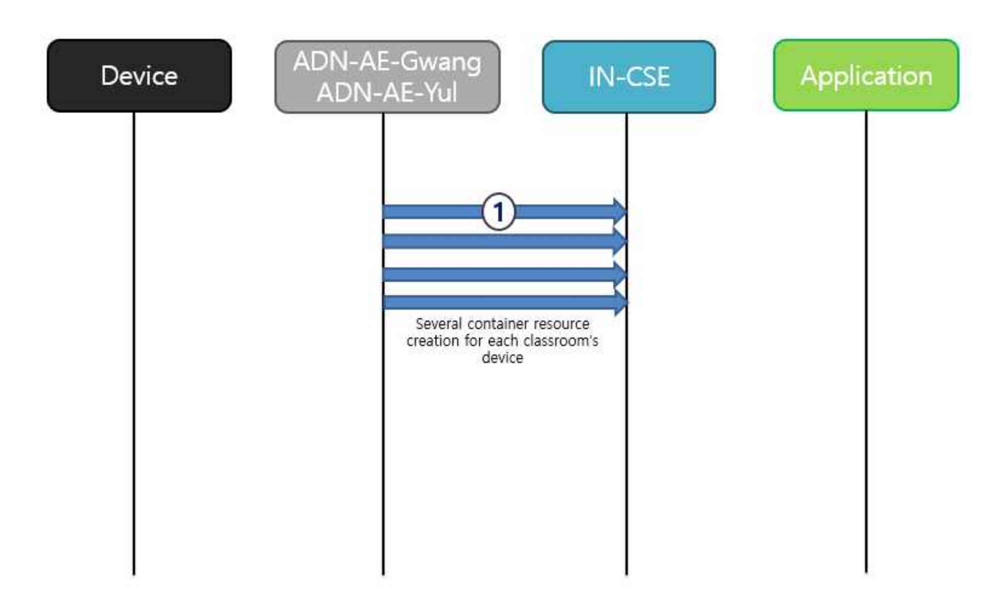
Device
ADN-AE-Gwang
ADN-AE-Yul
IN-CSE
Application
1
Several container resource creation for each classroom's device

Figure 3.2-1 : Initial resource creation phase call flows

After the initial resource creation process, the resource tree of IN-CSE is depicted in figure 3.2-2.

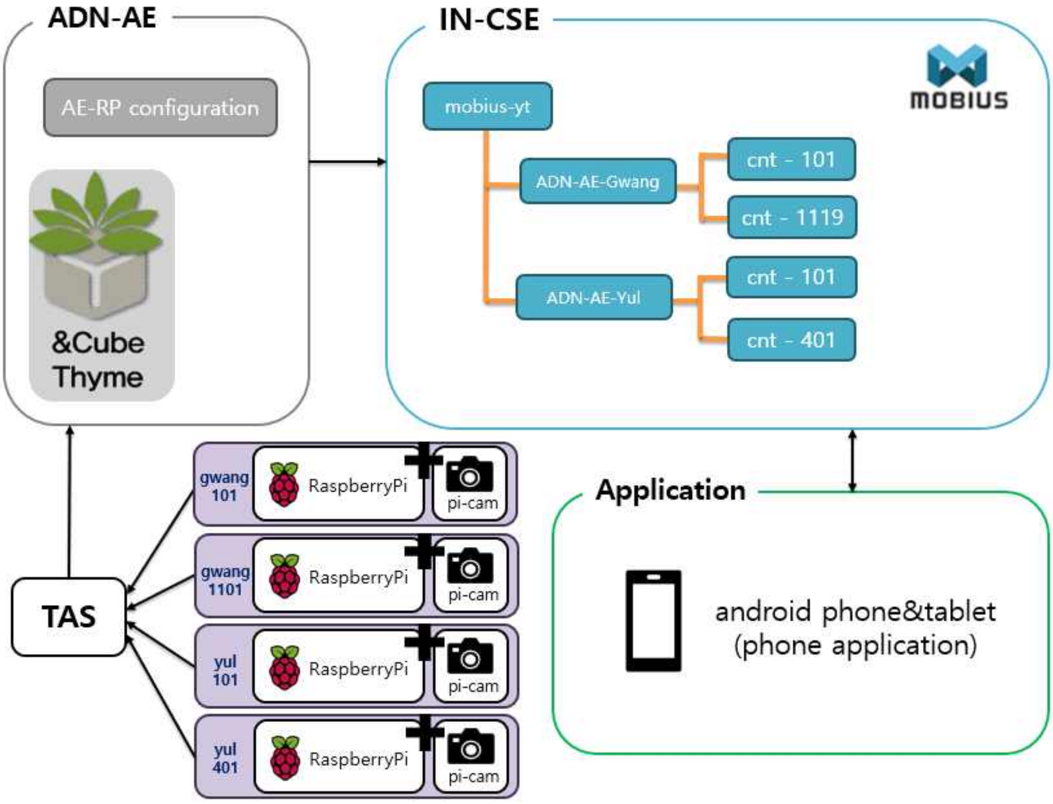

Figure 3.2-2 : State of after the initial creation process

3.3. Discovery and retrieval

Call flows regarding the discovery and retrieval of resources depicted in figure 3.3-1 are ordered as follows :

1) The smart phone application(Application) sends a RETRIEVE request including the parameter filterUsage and specific filter criteria condition(s) as a query string for discovery of resources stored under the ADN-AE-Gwang / ADN-AE-Yul.

2) The Mobius(IN-CSE) responds to the smart phone application(Application) with URIs of the discovered resources under ADN-AE-Gwang / ADN-AE-Yul.

3) The smart phone application(Application) sends RETRIEVE requests for retrieval of the latest data from discovered device resource(in the app, when user selects the floor), in this example, which is from the container-gwang1119 of ADN-AE-Gwang / ADN-AE-Yul(Figure 3.3-2).

4) The Mobius (IN-CSE) responds to the smart phone application (Application) with the latest data of device(contentInstance).

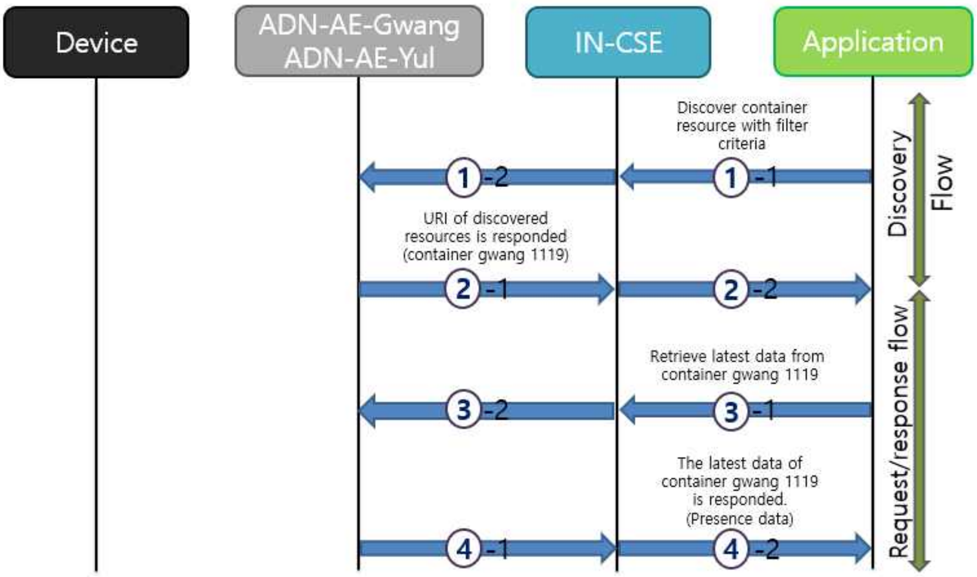
Device
ADN-AE-Gwang
ADN-AE-Yul
IN-CSE
Application
Discover container resource with filter criteria
1-2
1-1
Discovery Flow
URI of discovered resources is responded (container gwang 1119)
2-1
2-2
Retrieve latest data from container gwang 1119
3-2
3-1
The latest data of container gwang 1119 is responded. (Presence data)
4-1
4-2
Request/response flow

Figure 3.3-1 : Discovery and Retrieve phase call flows

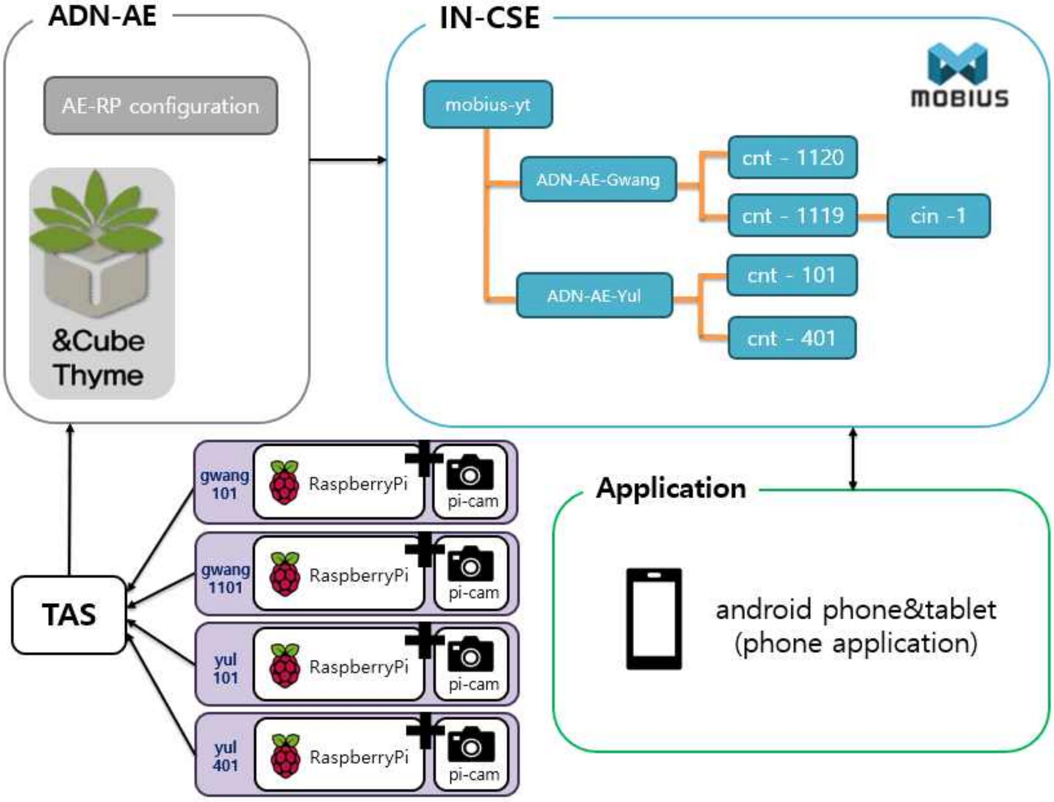

Figure 3.3-2 : Example of retrieve of latest data

4. Roles of entities

4.1. Mobius (IN-CSE)

The Mobius platform is modelled as an IN-CSE and is responsible for:

- handling the requests from smart phone Application and &Cube(ADN-AE-Gwang / ADN-AE-Yul)

4.2. &Cube Thyme (ADN-AE-Gwang / ADN-AE-Yul)

Each of the device applications are modelled as an &Cube(ADN-AE-Gwang / ADN-AE-Yul) and are responsible for :

- registering devices with the IN-CSE
- creating container resources in the IN-CSE
- creating content resources under containers cin-1 or cin-0 according to data

4.3. Smart phone applications (Appication)

The smart phone application which directly communicates with the Mobius(IN-CSE) and is responsible for :

- discovering and displaying the data from IN-CSE

5. Procedures

5.1. Registration and resource creation

The following example shows an device application ADN-AE-Gwang / ADN-AE-Yul registration request and response in clause 3.1 using Http with JSON serialization.

Http Request : (ADN-AE-Gwang)

```
Method : POST
Path : http://58.233.226.102:7579/mobius-yt?rcn=3
Headers
{
        Accept : application/xml
        X-M2M-RI : 4891
        X-M2M-Origin : S
        Content-Type : application/vnd.onem2m-res+xml; ty=2
}
Body
{
        <?xml version="1.0" encoding="UTF-8"?>
        <m2m:ae mlns:m2m="http://www.onem2m.org/xml/protocols"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        rn="adn-ae-Gwang">
                <api>4.2.801.916.0.130</api>
                <rr>true</rr>
        </m2m:ae>
}
```

Http Response :

```
{
        <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
        <m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols“
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <m2m:ae rn="adn-ae-Gwang">
                        <ty>2</ty>
                        <pi>rkR-qJGyf</pi>
                        <ri>HkbCvxBExf</ri>
                        <ct>20171123T123445</ct>
                        <et>20191123T123445</et>
                        <lt>20171123T123445</lt>
                        <api>4.2.801.916.0.130</api>
                        <aei>S20171123123445877p0AT</aei>
                </m2m:ae>
                <m2m:uri>/mobius-yt/adn-ae-Gwang</m2m:uri>
        </m2m:rce>
}
```

Http Request : (ADN-AE-Yul)

```
Method : POST
Path : http://58.233.226.102:7579/mobius-yt?rcn=3
Headers
{
        Accept : application/xml
        X-M2M-RI : 4891
        X-M2M-Origin : S
        Content-Type : application/vnd.onem2m-res+xml; ty=2
}
Body
{
    <?xml version="1.0" encoding="UTF-8"?>
    <m2m:ae mlns:m2m="http://www.onem2m.org/xml/protocols"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    rn="adn-ae-Gwang">
        <api>4.2.801.916.0.130.2</api>
        <rr>true</rr>
    </m2m:ae>
}
```

Http Response :

```
{
    <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <m2m:ae rn="adn-ae-Yul">
            <ty>2</ty>
            <pi>rkR-qJGyf</pi>
            <ri>BybLgeBVgz</ri>
            <ct>20171123T123245</ct>
            <et>20191123T123245</et>
            <lt>20171123T123245</lt>
            <api>4.2.801.916.0.130.2</api>
            <aei>S20171123123245569875z</aei>
        </m2m:ae>
        <m2m:uri>/mobius-yt/adn-ae-Yul</m2m:uri>
    </m2m:rce>
}
```

Then the following example shows a container(cnt-1119) create request and response in the porcedure of clause Figure 3.2-1 using Http with xml serialization.

Http Request :

```
Method : POST
Path : http://58.233.226.102:7579/mobius-yt/adn-ae-Gwang
Headers
{
    Accept : application/json
    X-M2M-RI : 4891
    X-M2M-Origin : SOrigin
```

```
          Content-Type : application/vnd.onem2m-res+xml; ty=3
}
Body
{
          <?xml version="1.0" encoding="UTF-8"?>
          <m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          rn="cnt-1119">
                    <lbl>gwang1119</lbl>
          </m2m:cnt>
}
```

Http Response :

```
{
          <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
          <m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          rcn="cnt-1119">
                    <pi>HkbCvxBExf</pi>
                    <ty>3</ty>
                    <ct>20171123T124347</ct>
                    <ri>S1GitzSElM</ri>
                    <lt>20171123T124850</lt>
                    <et>20191123T124347</et>
                    <lbl>gwang1119</lbl>
                    <st>2</st>
                    <mni>3153600000</mni>
                    <cr>SOrigin</cr>
                    <mbs>3153600000</mbs>
                    <mia>31536000</mia>
                    <cni>2</cni>
                    <cbs>2</cbs>
          </m2m:cnt>
}
```

Then the creation of a content instance resource under the container of ADN-AE-Gwang / ADN-AE-Yul with initial content is shown in the following procedure. The following example shows a contentInstance create request and response using Http with xml serialization :

Http Request :

```
Method : POST
```

```
Path : http://58.233.226.102:7579/mobius-yt/adn-ae-Gwang/cnt-1119
Headers
{
        Accept : application/xml
        X-M2M-RI : 4891
        X-M2M-Origin : SOrigin
        Content-Type : application/vnd.onem2m-res+xml; ty=4
}
Body
{
        <?xml version="1.0" encoding="UTF-8"?>
        <m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        rn="gwang1119cam">
                <con>1</con>
        </m2m:cin>
}
```

Http Response :

```
{
        <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
        <m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        rn="gwang1119cam">
                <ty>4</ty>
                <pi>S1GitzSElM</pi>
                <ri>B1Gs2QrEez</ri>
                <ct>20171123T124850</ct>
                <et>20191123T124850</et>
                <lt>20171123T124850</lt>
                <st>2</st>
                <cs>1</cs>
                <con>1</con>
                <cr>SOrigin</cr>
        </m2m:cin>
}
```

The smart phone applications can register with the Mobius anytime as needed.

5.2. Discovery and Retrieve

As mentioned in clause Figure 3.3-1, the smart phone application periodically sends a RETRIEVE request including the parameter filterUsage and specific filter criteria condition(s) as a query string for discovery of resources stored in the IN-CSE.

The discovery of containers for each devices registered with the ADN-AE-Gwang / ADN-AE-Yul by the smart phone AE is shown in the following procedure.

Http Request :

```
Method : GET
Path : http://58.233.226.102:7579/mobius-yt?fu=1&ty=3
Headers
{
        Accept : application/xml
        X-M2M-RI : 4891
        X-M2M-Origin : SOrigin
}
```

Http Response :

```
{
        <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
        <m2m:uril xmlns:m2m="http://www.onem2m.org/xml/protocols"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
        /mobius-yt/adn-ae-Gwang/cnt-1120 /mobius-yt/adn-ae-Yul/cnt-40
        1/mobius-yt/adn-ae-Yul/cnt-101 /mobius-yt/adn-ae-Gwang/cnt-
        1119</m2m:uril>
}
```

The smart phone application can retrieves the device data from ADN-AE-Gwang / ADN-AE-Yul. If the response is preferred to be returned with a xml representation, the following is a Http request message example :

Http Request :

```
Method : GET
Path : http://58.233.226.102:7579/mobius-yt/adn-ae-Gwang/cnt-1119/latest
Headers
{
        Accept : application/xml
        X-M2M-RI : 4891
        X-M2M-Origin : SOrigin
}
```

Http Response :

```
{
        <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
        <m2m:dbg xmlns:m2m="http://www.onem2m.org/xml/protocols"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        rn="gwang1119cam">
                <pi>S1GitzSElM</pi>
                <ty>4</ty>
                <ct>20171123T124850</ct>
                <ri>B1Gs2QrEez</ri>
                <lt>20171123T124850</lt>
                <et>20191123T124850</et>
                <st>2</st>
                <cs>1</cs>
                <cr>SOrigin</cr>
                <con>1</con>
        </m2m:cin>
}
```