Tizen/Artik IoT Lecture Chapter 14. IoTivity Easy Setup Manager

Sungkyunkwan University

Contents

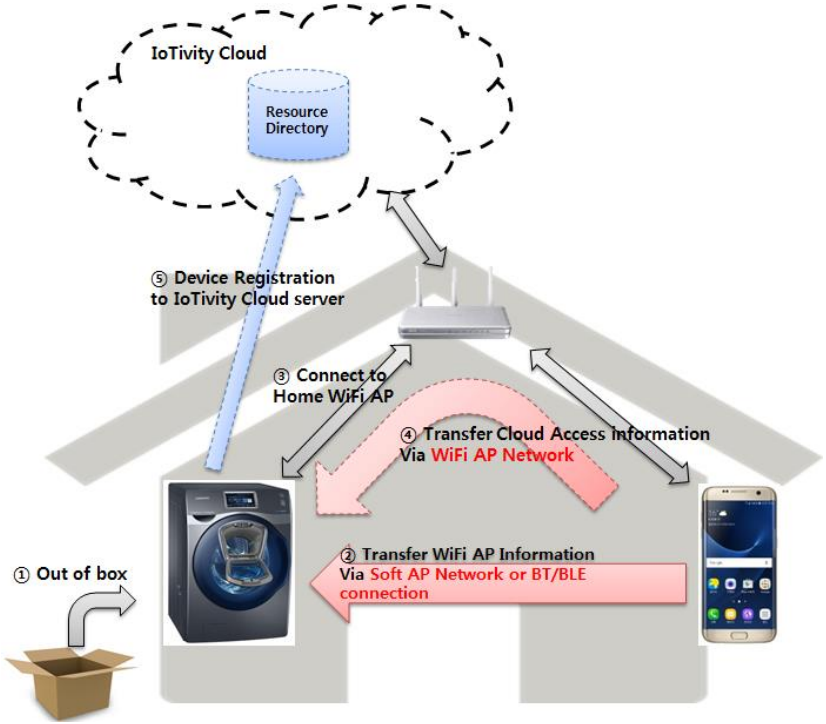
- **Easy Setup**
  - Use Cases
  - Device Roles
- **Easy Setup Resource Model**
  - Provisioning Resource
  - WiFi Resource
  - DevConf Resource
  - CloudServer Resource
- **Overall Workflow**
- **API Usage**

Easy Setup

- For making UI-less unboxed devices to be easily connected to the end user's IoTivity network seamlessly
- User can transfer a bunch of essential information to the unboxed devices in easy setup phase
  - **WiFi AP connection information**
    - The device to connect to Home AP
  - **Device configuration**
    - Like language and country settings
  - **Cloud access information**
    - Devices can register them to an IoTivity cloud server
    - User can access them via IoTivity cloud even from a distance.
- **Phase: Onboarding & Provisioning**

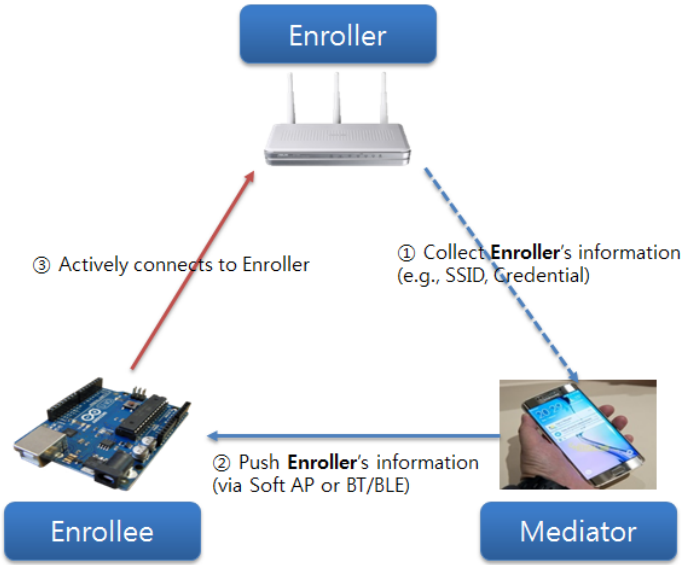
Use Cases

- **Mobile connects to the unboxed device.**
  - Using a Soft AP network or BT/BLE connection
- **Mobile transfers Home AP's information and other information**
  - SSID, password, security type of Home AP
  - Language, country used in the device.



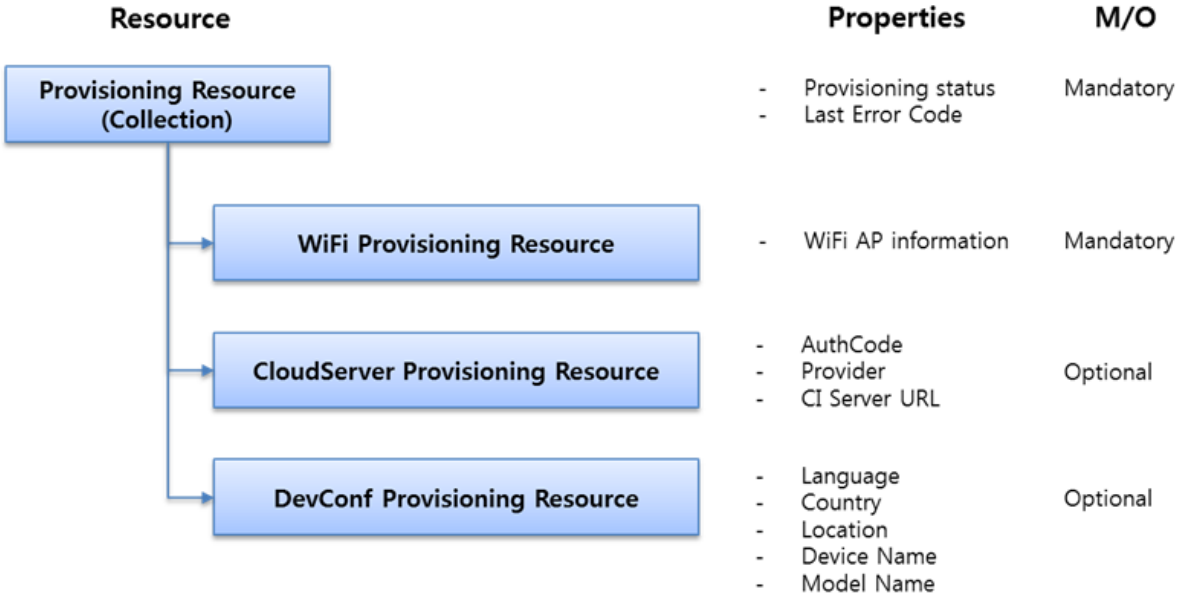
Device Roles

- **Enrollee**
  - Needs to be configured to join user's device network
- **Mediator**
  - To enable Enrollee devices to be easily connected to the target WiFi network and the target IoTivity cloud server
- **Enroller**
  - Target network entity to be connected by Enrollee



Resource Model

- **OCF resources for Easy Setup Service**
  - Provisioning, WiFi, DevConf, and CloudServer resources



Resource Model: Provisioning Resource

- A current status in easy setup process
- A last error code describing a reason of some failures occurred at the last time.

Resource Name	URI	Resource Type	CRUDN permission
Provisioning	/ProvisioningResURI	oic.wk.prov	RU

Property	Property Name(key)	Value Type	Access Mode	M / O	Description
Provisioning Status	ps	enum	R	M	Indicates the provisioning status of the device (TBD) (0: Need to provision, 1: Connecting to Enroller, 2: Connected to Enroller, 3:Failed to Connect to Enroller, 4: Registered to Cloud, 5: Failed to Register to Cloud)
Last Error Code	lec	enum	R	M	Indicates a failure reason if it fails to connect to Enroller (0: NO error, 1: A given SSID is not found, 2: WiFi's password is wrong, 3: IP address is not allocated, 4: NO internet connection, 5: Timeout, 6: Unknown error)
Links	links	array	R	M	Array of OIC web links that are wifi, cloudserver, and devConf resources

Resource Model: WiFi Resource

- WiFi SSID and password
- WiFi Security type (i.e. auth type and encription type)
- WiFi hardware capability (i.e. supported frequency and mode)

Resource Name	URI	Resource Type	CRUDN permission
WiFi	/WiFiProvisioningResURI	oic.wk.wifi	RU

Property	Property Name(key)	Value Type	Access Mode	M / O	Description
Supported WiFi Mode Type	swmt	array	R	M	Indicates supported WiFi mode types. It can be multiple. (i.e. 0: A, 1: B, 2: G, 3: N, 4: AC)
Supported WiFI Freq.	swf	enum	R	M	Indicates supported WiFi Frequency. (i.e. 0: 2.4G, 1: 5G, 2: both)
Target Network Name	tnn	string	RW	M	Indicates SSID of WiFi AP
Credential	cd	string	RW	O	Indicates credential information of WiFi AP
WiFi Auth Type	wat	enum	RW	O	Indicates WiFi Auth Type (i.e. 0: None, 1: WEP, 2: WPA-PSK, 3: WPA2-PSK)
WiFi Encryption Type	wet	enum	RW	O	Indicates WiFi Encryption Type (i.e. 0: None, 1: WEP-64, 2: WEP-128, 3: TKIP, 4: AES, 5: TKIP_AES)

Resource Model: DevConf Resource

- Device name (human-friendly)
- Model Number (defined by manufacturer)
- Lanugage (Follow IETF laungage tag using ISO 639X)
- Country (ISO 3166-1 Alpha-2 encoded)
- Location (GPS in json)

Resource Name	URI	Resource Type	CRUDN permission
DevConf	/DevConfProvisioningResURI	oic.wk.devconf	RU

Property	Property Name(key)	Value Type	Access Mode	M / O	Description
Device Name	dn	string	R	M	Device's human-friendly name like device model name
Model Name	mnmo	string	R	O	Model number as designated by manufacturer
Language	lang	string	RW	O	IETF language tag using ISO 639X
country	ctry	string	RW	O	ISO Country Code (ISO 3166-1 Alpha-2)
location	loc	string	RW	O	GPS information of device. Longitude, and latitude in json format

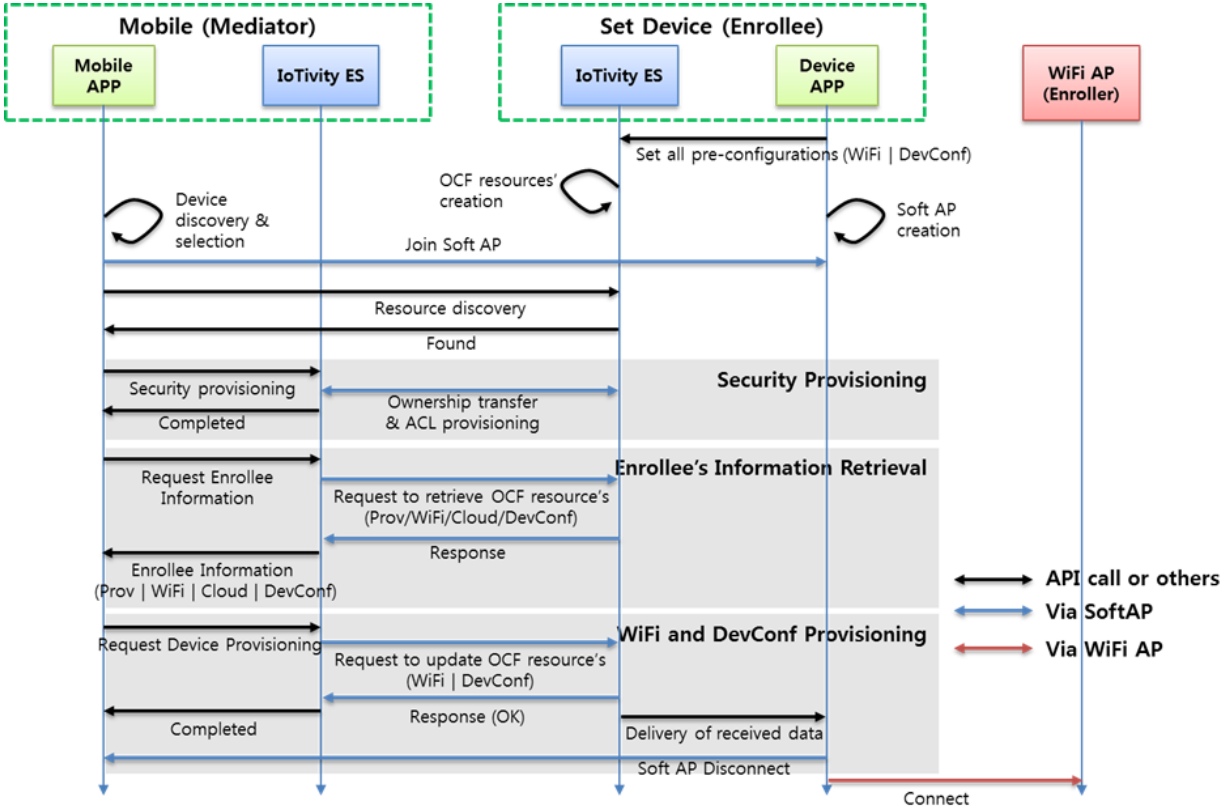
Resource Model: CloudServer Resource

- Auth code
- Authentication provider ID
- Cloud interface server URL

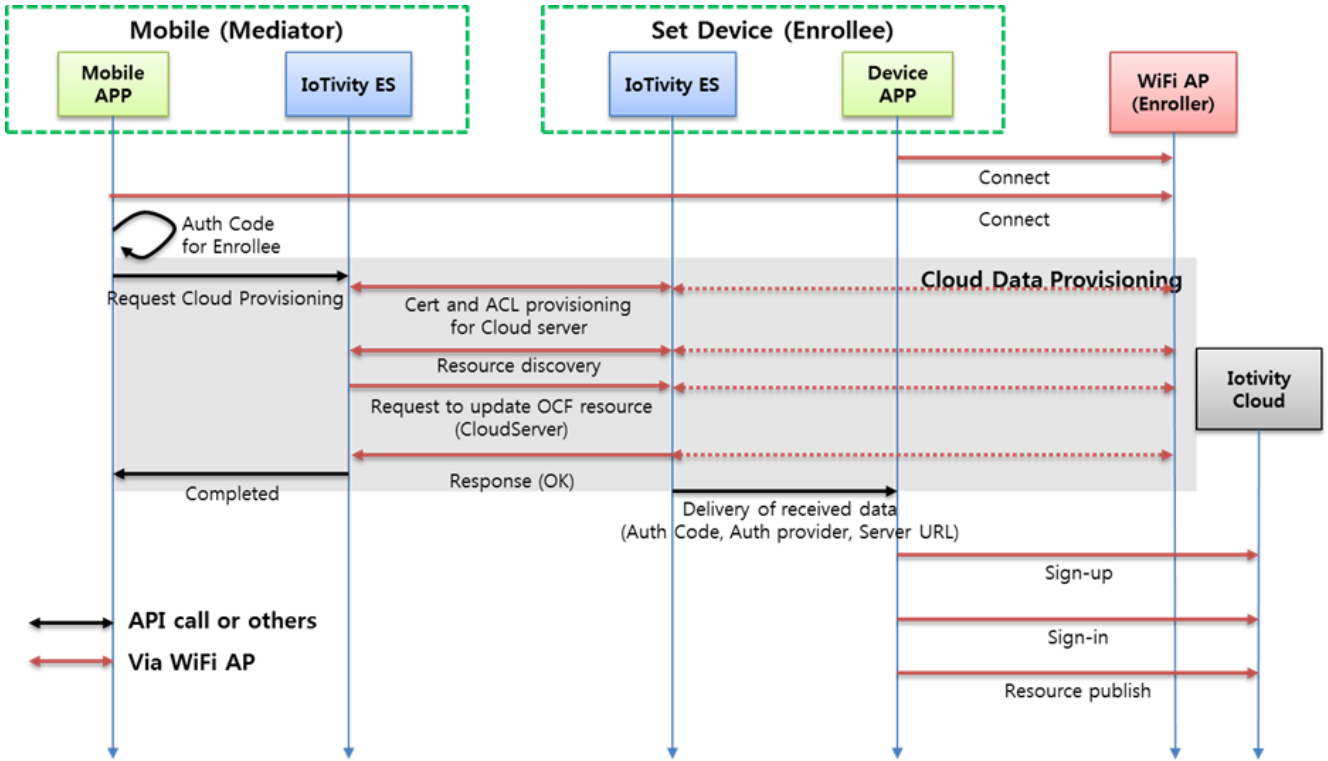
Resource Name	URI	Resource Type	CRUDN permission
Cloud Server Resource	/CloudServerProvisioningResURI	oic.wk.cloudserver	RU

Property	Property Name(key)	Value Type	Access Mode	M / O	Description
Auth Code	ac	string	RW	M	Indicates authCode provided by Authenticate server
Auth Provider Name	apn	string	RW	M	Indicates authenticate provider's name issuing an auth code
Cloud interface server	cis	string	RW	M	Indicates URL for IoTivity cloud interface server

Easy Setup Manager: Overall Workflow (1/2)



Easy Setup Manager: Overall Workflow (2/2)



API Usage: Mediator SDK

- **EasySetup class**
  - Create an instance of RemoteEnrollee object with a discovered resource which is represented as Provisioning resource in Enrollee
- **RemoteEnrollee class**
  - Provide all necessary APIs for communicating with the Enrollee
- **EnrolleeSecurity class**
  - Provide a functionality related to security provisioning
  - Perform an Ownership transfer, ACL provisioning, and Cert provisioning to Enrollee
- **EnrolleeResource class**
  - Perform a functionality to handle request/response related to WiFi and DevConf resource's properties
- **CloudResource class**
  - Perform a functionality to handle request/response related to CloudServer resource's properties

API Usage: Enrollee SDK

- **API includes**
  - Initialize Enrollee with resources' creation
    - **ESInitEnrollee()**
  - Set pre-configured properties in the resources
    - **ESSetDeviceProperty()**
  - Set provisioning status and last error code in Provisioning resource,
    which will let Mediator know by GET request
    - **ESSetState(), ESSetErrorCode()**
  - Terminate Enrollee with resources' destruction
    - **ESTerminateEnrollee()**

API Usage: Start Easy Setup

- **Application: Start Easy Setup**

```
ESProvisioningCallbacks gCallbacks = {
    .WiFiProvCb = &WiFiProvCbInApp,
    .DevConfProvCb = &DevConfProvCbInApp,
    .CloudDataProvCb = &CloudDataProvCbInApp
};
```

```
ESResourceMask resourcemMask = (ESResourceMask) (ES_WIFI_RESOURCE | ES_CLOUD_RESOURCE | ES_DEVCONF_RESOURCE);
cout << "resourcemMask : " << resourcemMask << endl;
if(ESInitEnrollee(gIsSecured, resourcemMask, gCallbacks) ! = ES_OK)
```

- **IoTivity: ESInitEnrollee**
  - Setup Callback
    - ES_WIFI_RESOURCE, ES_DEVCONF_RESOURCE, ES_CLOUD_RESOURCE
  - Creation Resource

```
if((resourceMask & ES_WIFI_RESOURCE) == ES_WIFI_RESOURCE)
{
    if(callbacks.WiFiProvCb ! = NULL)
    {
        gESProvisioningCb.WiFiProvCb = callbacks.WiFiProvCb;
        RegisterWifiRsrcEventCallBack(ESWiFiRsrcCallback);
    }
```

```
res = OCCreateResource(&gDevConfResource.handle, OC_RSRVD_ES_RES_TYPE_DEVCONF,
OC_RSRVD_INTERFACE_DEFAULT,
OC_RSRVD_ES_URI_DEVCONF, OCEntityHandlerCb,
NULL, OC_DISCOVERABLE | OC_OBSERVABLE);
```

API Usage: Mediator

- **Connect to Enrollee & Discovery a Provisioning resource**

```
PlatformConfig config
{
    OC::ServiceType::InProc, ModeType::Both, "0.0.0.0", 0, OC::QualityOfService::LowQos, &ps
};

OCPlatform::Configure(config);
```

```
std::ostringstream requestURI;
requestURI << OC_RSRVD_WELL_KNOWN_URI << "?rt=ocf.wk.prov";

OCPlatform::findResource("", requestURI.str(), CT_DEFAULT, &foundResource);
std::cout<< "Finding Resource... " <<std::endl;
```

```
remoteEnrollee = EasySetup::getInstance()->createRemoteEnrollee(resource);
```

- **foundResource Callback**
  - Creation remoteEnrollee

API Usage: Mediator

- **Get Configuration**

```
remoteEnrollee->getConfiguration(getConfigurationCallback);
```

  - Callback

```
void getConfigurationCallback(std::shared_ptr< GetConfigurationStatus > getConfigurationStatus)
{
    EnrolleeConf conf = getConfigurationStatus->getEnrolleeConf();
    cout << "========================================" << endl;
    cout << "\tDevice Name : " << conf.getDeviceName() << endl;
    cout << "\tModel Number : " << conf.getModelNumber() << endl;
```

- **Provisioning**

```
void provisionDeviceProperty()
{
    DeviceProp devProp;
    devProp.setWiFiProp("Iotivity_SSID", "Iotivity_PWD", WPA2_PSK, TKIP_AES);
    devProp.setDevConfProp("korean", "Korea", "Location");

    try
    {
        remoteEnrollee->provisionDeviceProperties(devProp, deviceProvisioningStatusCallback);
```