Tizen/Artik IoT Lecture Chapter 16. IoTivity Provisioning Manager

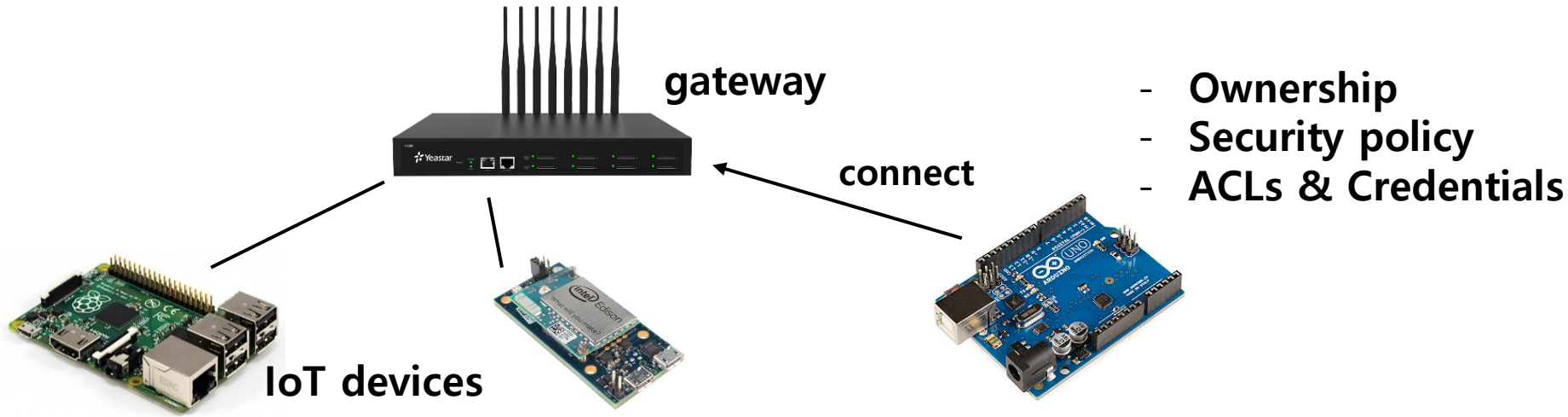
Sungkyunkwan University

Contents

- **Provisioning Manager**
  - Roles
  - Architecture
- **Ownership Transfer Method**
  - "Just Work"
  - Discovery and Set Ownership Transfer Method
- **Access Control List (ACL)**
- **Sequence Diagram of Direct Pairing**
- **Sample Applications**

What is Provisioning Manager?

- **Security administrator of IoT devices in its IP subnet**
  - Manages the ownership of the IoT devices and provides proper security policy

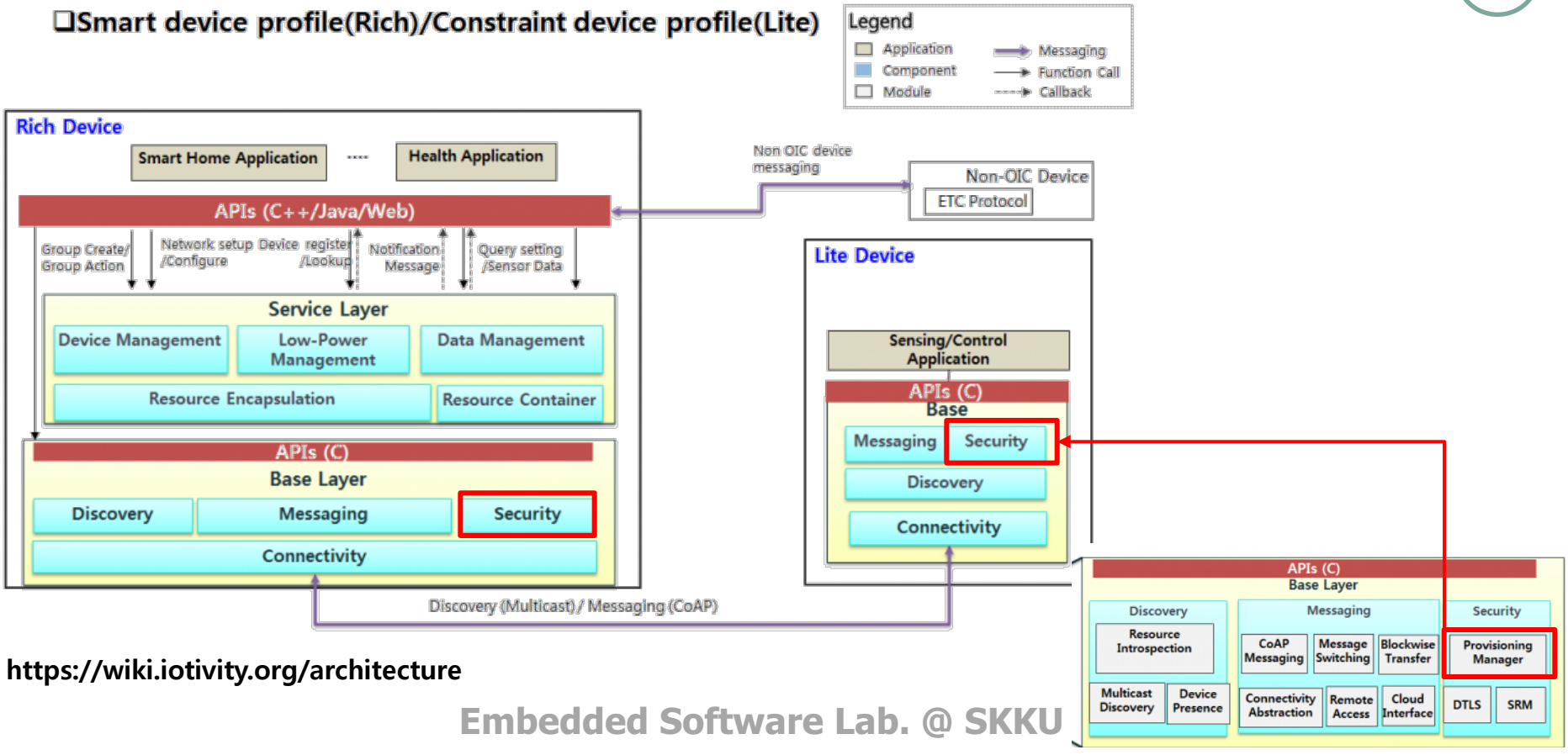


Roles of Provisioning Manager

- **Ownership transfer**
  - PM discovers new device and transfers the ownership to admin
  - Two methods: "Just works" and "Random PIN based"
- **Security management of owned devices**
  - PM provisions and revokes credentials and ACL to owned devices
  - PM keeps provisioned credential history to manage OIC network (Provisioning Database Manager, Secure Resource Provider)
- **Direct pairing**
  - Enables a security establishment between two IoT devices without any help of security tools and/or services
    - An immediate use of a new IoT device when provisioning tools/services are not available
    - Access and control to user IoT devices by a guest IoT device
    - Access and control among user IoT devices provisioned or owned by different provisioning tools/services

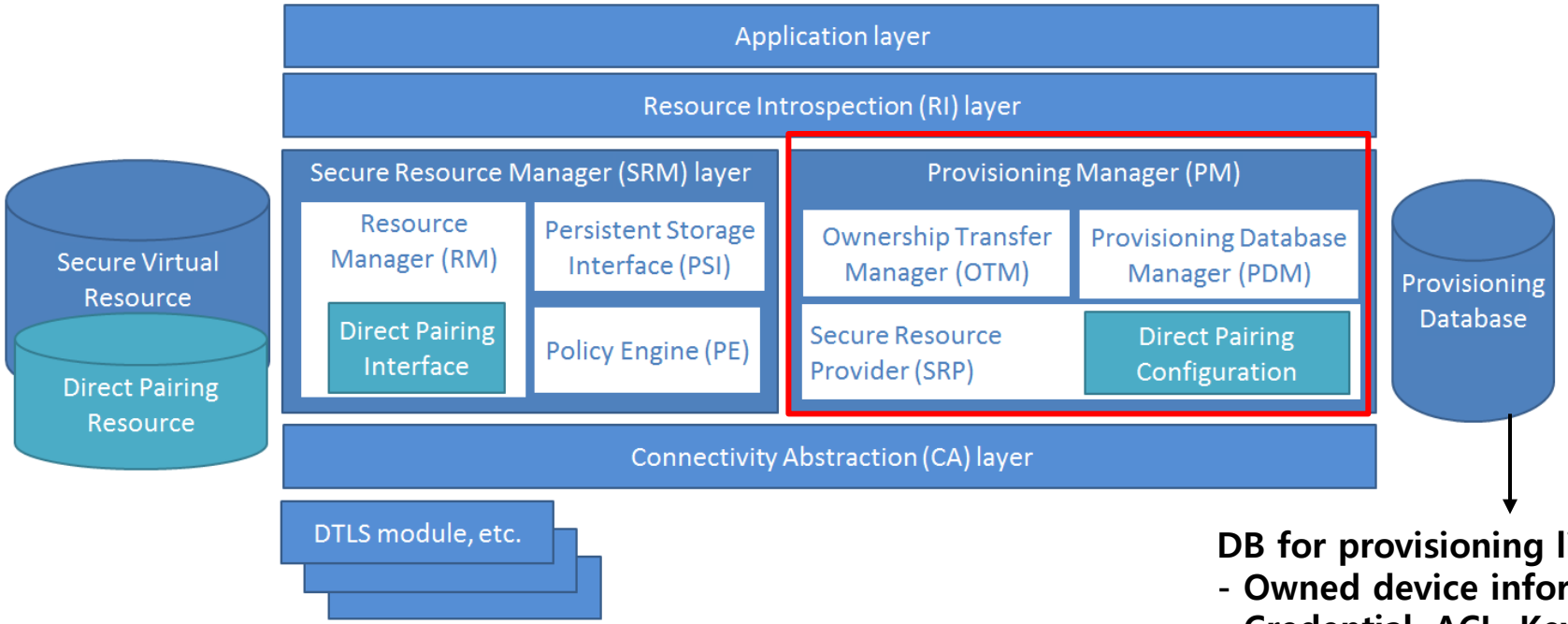
IoTivity Overall Architecture

❑Smart device profile(Rich)/Constraint device profile(Lite)



https://wiki.iotivity.org/architecture

Architecture of Provisioning Manager (PM)



DB for provisioning like,
- Owned device information
- Credential, ACL, Key

https://wiki.iotivity.org/provisioning

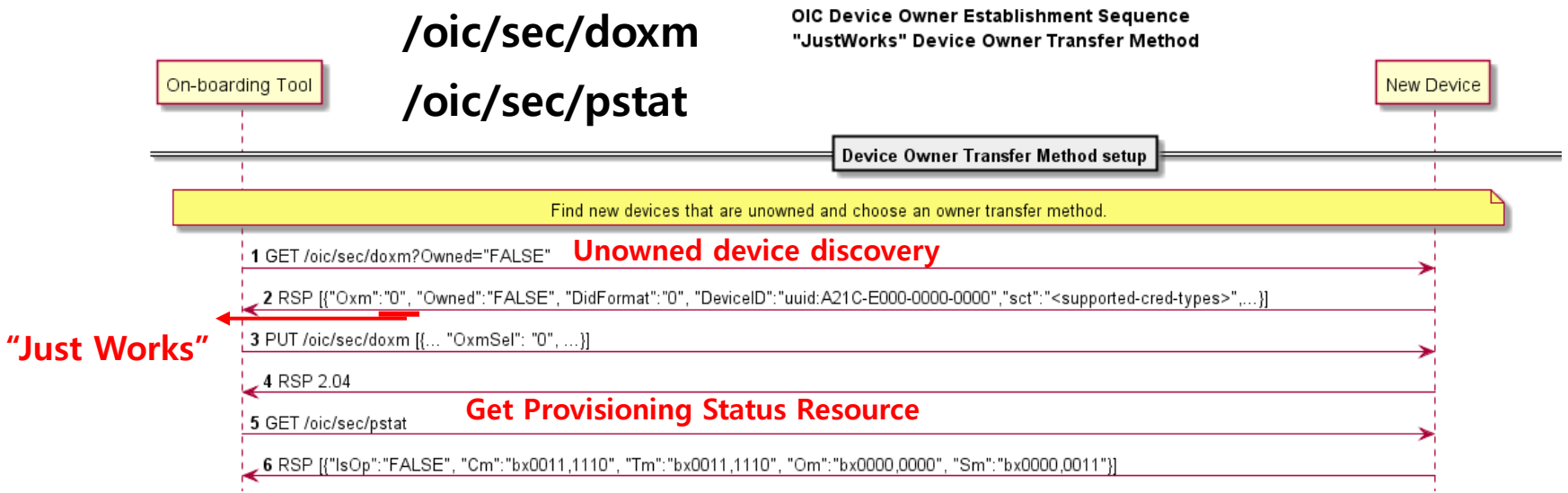
"Just Work" Ownership Transfer Method

- **"Just works" device owner transfer**
  - Find new devices that are unowned and choose an owner transfer method
  - On-boarding tools (OBT) tell new device how provisioning will be achieved (register device owner information)
  - The OBT decides which credential type will be used as owner credentials based on 'sct' of new device's doxm
    - Symmetric credential type, asymmetric credential type
  - Establish a secure session using owner credential

/oic/sec/doxm (device owner transfer method)
/oic/sec/pstat (provisioning status)

Discovery and Set Ownership Transfer Method

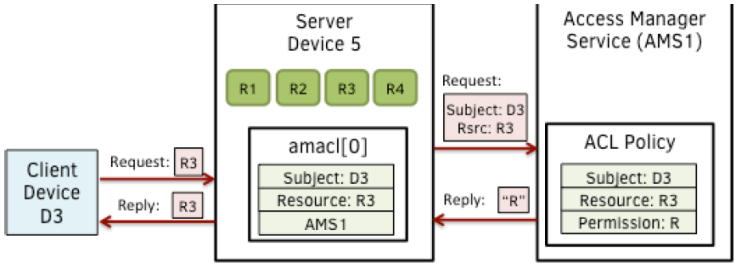
- **In "Just works" device owner transfer**



https://wiki.iotivity.org/provisioning

Access Control List (ACL)

- **Resources are hosted at OIC server and are made available to OIC clients subject to access control and authorization mechanisms**
- **Two types of access control mechanism**
  - Subject-based access control (SBAC)
  - Role-based access control (RBAC)



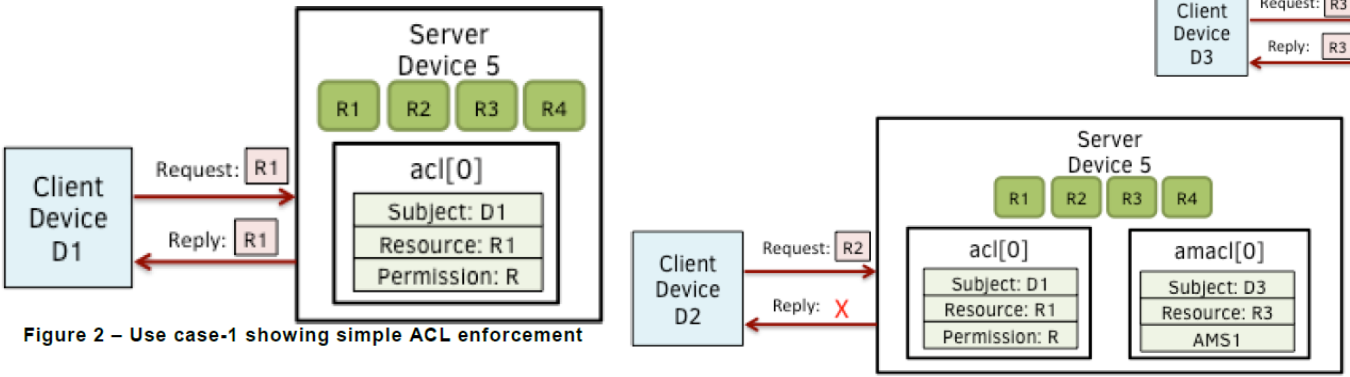
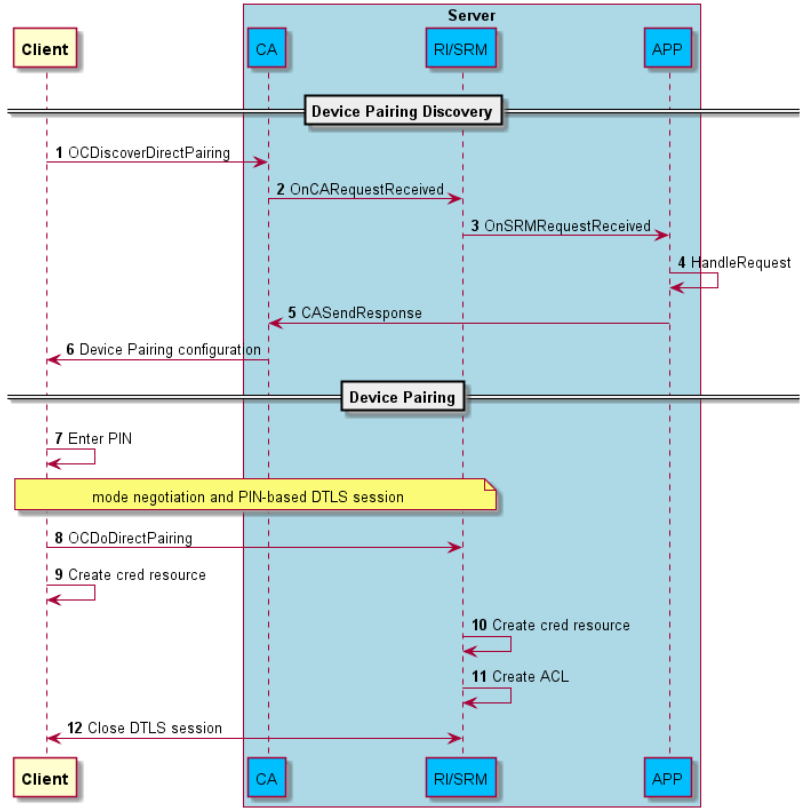


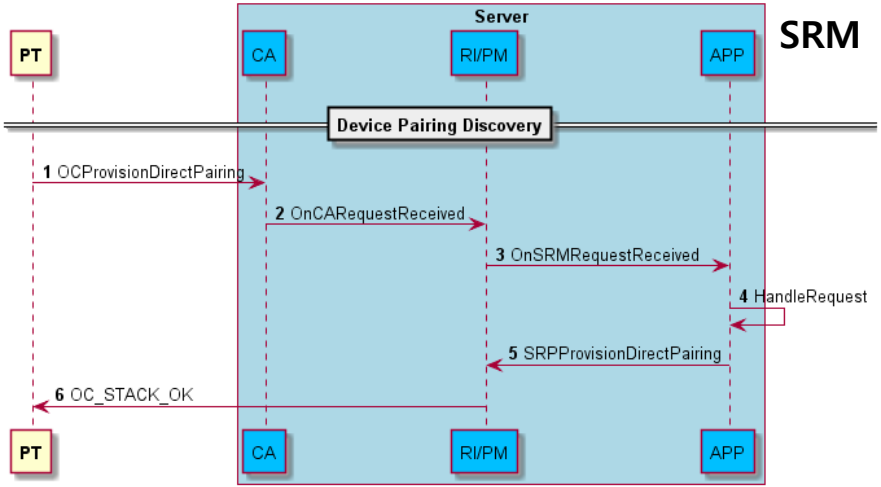
Figure 2 – Use case-1 showing simple ACL enforcement

Sequence Diagram of Direct Pairing

Device Pairing Sequence



Direct Pairing Configuration



SRM

/oic/sec/pconf

Sample Applications

- **How to Build**
  - $ cd ~/<IoTivity_DIR>
  - $ scons resource SECURED=1
    - Give security option when build the source
  - Directory
    - ~/<IoTivity_DIR>/resource/provisioning
    - ~/<IoTivity_DIR>/csdk/security/provisioning
- **Programs**
  - sampleserver_justworks
  - sampleserver_randompin
  - provisioningclient

Sample Application: provisioningclient

- **Ownership transfer**
- **Provision ACL and Credentials**
- **Provision direct-pairing configuration**

```
void printMenu()
{
    std::cout << "\nChoose an option:"<<std::endl;
    std::cout << "   1. UnOwned Device discovery"<<std::endl;
    std::cout << "   2. Owned Device discovery"<<std::endl;
    std::cout << "   3. Ownership transfer"<<std::endl;
    std::cout << "   4. Provision ACL"<<std::endl;
    std::cout << "   5. Provision Credentials"<<std::endl;
    std::cout << "   6. Credential & ACL provisioning b/w two devices"<<std::endl;
    std::cout << "   7. Unlink Devices"<<std::endl;
    std::cout << "   8. Remove Device"<<std::endl;
    std::cout << "   9. Remove Device using UUID"<<std::endl;
    std::cout << "  10. Get Linked Devices"<<std::endl;
    std::cout << "  11. Get Device Status"<<std::endl;
    std::cout << "  12. Provision Direct-Pairing Configuration"<<std::endl;
#if defined(__WITH_X509__) || defined(__WITH_TLS__)
    std::cout << "  13. Save the Trust Cert. Chain into Cred of SVR"<<std::endl;
    std::cout << "  14. Provision the Trust Cert. Chain"<<std::endl;
#endif // __WITH_X509__ || __WITH_TLS__
    std::cout << "  99. Exit loop"<<std::endl;
} ? end printMenu ?
```

Discovery of Devices

- **Discovery of owned and unowned devices**
  - Unowned devices should transfer ownership to provisioning manager

Discovery of unowned device

```
sinban@eslab03:~/iotivity/out/linux/x86_64/release/resource/provisioning/examples$ ./provisioningclient

Choose an option:
   1. UnOwned Device discovery
   2. Owned Device discovery
   3. Ownership transfer
   4. Provision ACL
   5. Provision Credentials
   6. Credential & ACL provisioning b/w two devices
   7. Unlink Devices
   8. Remove Device
   9. Remove Device using UUID
  10. Get Linked Devices
  11. Get Device Status
  12. Provision Direct-Pairing Configuration
  99. Exit loop

1
Started discovery...
Found secure devices, count = 1
Device 1 ID: 72616e64-5069-6e44-6576-557569643030 From IP: fe80::a62:66ff:fe7f:9282%em1
```

Discovery of owned device

```
Choose an option:
   1. UnOwned Device discovery
   2. Owned Device discovery
   3. Ownership transfer
   4. Provision ACL
   5. Provision Credentials
   6. Credential & ACL provisioning b/w two devices
   7. Unlink Devices
   8. Remove Device
   9. Remove Device using UUID
  10. Get Linked Devices
  11. Get Device Status
  12. Provision Direct-Pairing Configuration
  99. Exit loop
2
Started discovery...
Found owned devices, count = 2
Device 1 ID: 72616e64-5069-6e44-6576-557569643030 From IP: fe80::a62:66ff:fe7f:9282%em1
Device 2 ID: 6a757374-776f-726b-4465-765575696430 From IP: fe80::a62:66ff:fe7f:9282%em1
```

Ownership Transferring

- **Transfer ownership of unowned device to provisioning manager**
  - "Just works" server is registered without key
  - "Random Pin" server is registered throughout the PIN code

```
Choose an option:
   1. UnOwned Device discovery
   2. Owned Device discovery
   3. Ownership transfer
   4. Provision ACL
   5. Provision Credentials
   6. Credential & ACL provisioning b/w two devices
   7. Unlink Devices
   8. Remove Device
   9. Remove Device using UUID
  10. Get Linked Devices
  11. Get Device Status
  12. Provision Direct-Pairing Configuration
  99. Exit loop
3
1: 72616e64-5069-6e44-6576-557569643030 From IP:fe80::a62:66ff:fe7f:9282%em1
Select device number:
1
Registering OTM Methods: 1. JUST WORKS and 2. PIN
Transfering ownership for : 72616e64-5069-6e44-6576-557569643030
INPUT PIN : 08417517

Transferred Ownership successfuly for device : r a n d P i n D e v U u i d 0 0
```

Transfer ownership

"Just Works" server

```
sinban@eslab03:~/iotivity/out/linux/x86_64/release/resource/csdk/security/provis
ioning/sample$ ./sampleserver_justworks
53:20.272 DEBUG: SAMPLE_JUSTWORKS: OCServer is starting...
53:22.276 INFO: SAMPLE_JUSTWORKS: Created LED resource with result: OC_STACK_OK
53:22.276 INFO: SAMPLE_JUSTWORKS: Entering ocserver main loop...
```

"Random PIN" server

```
sinban@eslab03:~/iotivity/out/linux/x86_64/release/resource/csdk/security/provisioning/sample$
./sampleserver_randompin
56:55.737 DEBUG: SAMPLE_RANDOMPIN: OCServer is starting...
56:57.741 INFO: SAMPLE_RANDOMPIN: Created LED resource with result: OC_STACK_OK
56:57.741 INFO: SAMPLE_RANDOMPIN: Entering ocserver main loop...
58:17.622 INFO: SAMPLE_RANDOMPIN: ==========================
58:17.622 INFO: SAMPLE_RANDOMPIN:    PIN CODE : 08417517
58:17.622 INFO: SAMPLE_RANDOMPIN: ==========================
```

PIN code generated

Provisioning ACL and Credential


```
Choose an option:
   1. UnOwned Device discovery
   2. Owned Device discovery
   3. Ownership transfer
   4. Provision ACL
   5. Provision Credentials
   6. Credential & ACL provisioning b/w two devices
   7. Unlink Devices
   8. Remove Device
   9. Remove Device using UUID
  10. Get Linked Devices
  11. Get Device Status
  12. Provision Direct-Pairing Configuration
  99. Exit loop
4
Owned devices, count = 2
Device 1 ID: 72616e64-5069-6e44-6576-557569643030 From IP: fe80::a62:66ff:fe7f:9282%em1
Device 2 ID: 6a757374-776f-726b-4465-765575696430 From IP: fe80::a62:66ff:fe7f:9282%em1
Select 1 device(s) for provisioning
Device 1 : 1
Provision ACL for : 72616e64-5069-6e44-6576-557569643030
Please input ACL for selected device:
```

Provision ACL to deivce1

```
*********************************************************************************
-Set ACL policy for target device
*********************************************************************************
-URN identifying the subject
ex) 1111-1111-1111-1111 (16 Numbers except to '-')
Subject : 1111-2222-3333-4444
Num. of Resource : 1
-URI of resource
ex)/oic/sh/temp/0 (Max_URI_Length: 256 Byte )
[1]Resource : /oic/eslab
        Enter Number of resource type for [/oic/eslab]: 1
        Enter ResourceType[1] Name (e.g. core.led): core.eslab
        Enter Number of interface name for [/oic/eslab]: 1
        Enter interfnace[1] Name (e.g. oic.if.baseline): oic.if.baseline
-Set the permission(C,R,U,D,N)
ex) CRUDN, CRU_N,..(5 Charaters)
Permission : CRUDN
-URN identifying the rowner
ex) 1111-1111-1111-1111 (16 Numbers except to '-')
Rowner : 5555-4444-3333-2222

Received provisioning results: Result is = 4 for device r a n d P i n D e v U u i d 0 0
```

Set ACL policy

Provision Direct-pairing Configuration


```
Choose an option:
   1. UnOwned Device discovery
   2. Owned Device discovery
   3. Ownership transfer
   4. Provision ACL
   5. Provision Credentials
   6. Credential & ACL provisioning b/w two devices
   7. Unlink Devices
   8. Remove Device
   9. Remove Device using UUID
  10. Get Linked Devices
  11. Get Device Status
  12. Provision Direct-Pairing Configuration
  99. Exit loop


12
1: 72616e64-5069-6e44-6576-557569643030 From IP:fe80::a62:66ff:fe7f:9282%em1
2: 6a757374-776f-726b-4465-765575696430 From IP:fe80::a62:66ff:fe7f:9282%em1
Select device number:
1
Enter PIN to be configured: 08417517
*********************************************************************************
-Set ACL policy for target DP device
*********************************************************************************
Num. of Resource : 1
-URI of resource
ex)/oic/sh/temp/0 (Max_URI_Length: 256 Byte )
[1]Resource : /oic/eslab
-Set the permission(C,R,U,D,N)
ex) CRUDN, CRU_N,..(5 Charaters)
Permission : CRUDN

Received provisioning results: Direct Pairing is successful Result is = 0 for device r a n d P i n D e v U u i d 0 0
```

PIN number of device 2