Gateway模块与接口设计

[文档副标题]

罗永恒

2015

目录

[1. 介绍 2](#_Toc413512015)

[2. Gateway模块设计 2](#_Toc413512016)

[2.1. 近场通信模块（ProximalCommEngine） 2](#_Toc413512017)

[2.1.1. 设计原则 2](#_Toc413512018)

[2.1.2. 功能说明 2](#_Toc413512019)

[2.2. 云端通信模块（CloudCommEngine） 3](#_Toc413512020)

[2.2.1. 设计原则 3](#_Toc413512021)

[2.2.2. 功能说明 3](#_Toc413512022)

[2.3. 调度模块（ConnectorEngine） 3](#_Toc413512023)

[2.3.1. 设计原则 3](#_Toc413512024)

[2.3.2. 功能说明 3](#_Toc413512025)

[3. Gateway模块的接口设计 4](#_Toc413512026)

[3.1. 近场通信模块（ProximalCommEngine） 4](#_Toc413512027)

[3.1.1. 面向AllJoyn网络的接口 4](#_Toc413512028)

[3.1.2. 面向IoTivity网络的接口（待计划） 7](#_Toc413512029)

[3.1.3. 面向MQTT网络的接口（待计划） 7](#_Toc413512030)

[3.2. 云端通信模块（CloudCommEngine） 7](#_Toc413512031)

[3.2.1. 面向AllJoyn网络的接口 7](#_Toc413512032)

[3.2.2. 面向IoTivity网络的接口（待计划） 11](#_Toc413512033)

[3.2.3. 面向MQTT网络的接口（待计划） 11](#_Toc413512034)

[3.3. 调度模块（ConnectorEngine） 11](#_Toc413512035)

# 介绍

Gateway是指运行于Gateway Agent中的通信中间件应用，它来源于AllSeen Alliance Gateway Agent Working Group的定义（参见：<https://wiki.allseenalliance.org/gateway/gatewayagent>）。在IoT Cloud框架中，此App将用作连接近场网络（不仅是AllJoyn设备，还包括其他IoT近场标准的设备，如IoTivity，MQTT等）与IMS网络的通信转换。它有两大功能：一是在近场网络中协调各种不同的IoT近场通信标准，让他们之间按照统一的方式进行互通；二是将IoT近场标准扩展至一个标准的云端网络框架，使“近场”的概念扩展至“云端”。

# Gateway模块设计

## 近场通信模块（ProximalCommEngine）

近场通信模块要向其他模块开放接口并提供服务，用什么技术来提供服务呢？用AllJoyn兼容技术（即DBus技术）来提供服务，还是用广义的面向对象的模块来提供服务？如果用AllJoyn兼容技术，则其实现较为标准和灵活，即，可用单进程实现，也可用多进程实现，甚至可实现分布式，但缺点是可能会损失一些性能和效率（可忽略不计？），且其缺乏面向对象的一些特性，如继承与多态；用广义的面向对象的模块来实现，即用紧耦合的形式实现各个功能模块，优点是性能好（因为不用IPC/RPC的调用开销），缺点是实现和部署不够灵活。

### 设计原则

这个模块负责与近场网络中的设备进行通信，设计的原则是：

* 包装所有的近场通信与交互，其他模块与调用者不知道任何近场通信标准的知识均可正常使用，这包括提供三大标准的支持：AllJoyn，IoTivity，MQTT；
* 与业务层（Services）松耦合，最好是完全无关。因为Gateway与业务层（Services）尽量保持无关性；
* 接口与近场标准无关，这包括统一三大近场标准的一些接口调用方式，如调用地址的设计。

### 功能说明

这个模块实现的功能：

* 对于云端进入的调用（包括调用云端接口或服务的返回），ConnectorEngine模块会分析其调用地址（如lyh@nane.cn/BusName/ControlPanel/MyBedroomTV/rootContainer/mute，表示要调用卧室电视的静音接口），然后传递到本模块。本模块将首先寻找BusName对应的BusAttachment，然后生成接口对应的BusProxyObject，填充参数并调用；
* 对于云端接口或服务的调用（或云端进入的调用的返回），本模块会在此之前管理云端接口或服务在本地的映射（如通过服务订阅，或第三方应用进行映射），准备好代理的BusObject并注册到BusAttachment（并且Announce）。当调用时，直接生成相应的调用地址（如weather\_server\_A@nane.cn/CurrentWeather/CurrentTemperature，表示要调用某天气服务商的当前温度获取接口），传递到ConnectorEngine模块。ConnectEngine模块会将这个调用转换后向云端发送。

## 云端通信模块（CloudCommEngine）

### 设计原则

CloudCommEngine是负责与云端（IMS网络）进行通信的模块，设计的原则是：

* 包装所有与云端（IMS网络）通信的细节；
* 与业务层（Services）松耦合，最好是完全无关；
* 一旦云端改变或升级，此模块可被无缝替代，其他模块与程序不用改变或升级，这要求其接口具有通用性，在升级换代过程中不用对接口做升级。

### 功能说明

## 调度模块（ConnectorEngine）

### 设计原则

### 功能说明

ConnectorEngine是核心调度模块，它要实现的功能是：

* 所有通信模块（包括近场通信模块与云端通信模块）的创建、维护、删除，即生命周期控制；
* 对调用地址（类似于：lyh@nane.cn/BusName/ControlPanel/MyBedroomTV/rootContainer/mute）的解析、生成、调用管理等；对于调用地址的管理非常重要，这涉及到一套灵活的机制，将所有可能的近场通信的标准全部纳入；
* 根据AllSeen Gateway Agent Working Group的标准建议，此模块还将负责Gateway中的其他功能，如与Management App的交互，权限及规则的管理等；

# Gateway模块的接口设计

## 近场通信模块（ProximalCommEngine）

### 面向AllJoyn网络的接口

#### 接口名

|  |  |  |
| --- | --- | --- |
| Interface name | Version | Secured |
| cn.nane.IoTCloud.Gateway.ProximalCommEngine.AllJoynEngine | 1 | no |

#### 属性

#### 方法

##### AJLocalMethodCall

**输入**

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
| addr | s |  | 调用的AllJoyn接口的地址，由四部分组成，即BusName/ObjectPath/InterfaceName/MethodName |
| para | av |  | 调用参数数组 |
| caller | s |  | 唯一标识云端/远端调用者的IMS Account，如controlpanel\_server@nane.cn。传入caller地址的目的是让本模块保存调用信息，在某些情况下，如需要返回信息给调用者，则可直接使用此调用者信息 |
| cloudSessionID | s |  | 代表云端到Gateway的Session唯一ID |
| reply | av | out | 返回值的数组 |
| localSessionID | x | out | 代表本地近场网络的Session唯一ID，由AllJoyn网络返回 |

**输出**

| Return Signature | Description |
| --- | --- |
| QStatus | 返回状态值 |

**描述**

云端调用近场AllJoyn设备的某接口。传入的调用地址为统一编码的调用地址，适用于所有匹配的近场标准（AllJoyn，IoTivity，MQTT）。

##### AJSubscribeCloudServiceToLocal

**输入**

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
| serviceAddr | s |  | 所要订阅的服务的地址，即服务对应的account，如某云服务地址为weather\_server\_A@nane.cn；某人家的AllJoyn智能设备向其他人开放接口，此地址为其account：thierry\_luo@nane.cn |
| introspectionXML | s |  | 接口描述的XML描述字符串，用来让ProximalCommEngine生成此接口的本地代理BusObject（及其方法、属性与信号处理机制）。此XML将包含云端服务（有时是其他近场网络的device或app）的基本信息（AllJoyn中的AboutData），以及所有根BusObject描述信息 |

introspectionXML以https://allseenalliance.org/schemas/introspect.xsd为蓝本，增加一部分数据，示例如下：

<service name="/BusName/RootObjPath">

<about>

<aboutData key="" value=""/>

<aboutData key="" value=""/>

</about>

<node name="BusObject1">

<interface name="">

<signal name="">

</signal>

<property name="" type="a{qv}" access="read"/>

<property name="" type="u" access="read"/>

<property name="" type="q" access="read"/>

</interface>

<interface name="org.freedesktop.DBus.Introspectable">

<method name="Introspect">

<arg name="data" type="s" direction="out"/>

</method>

</interface>

<interface name="org.allseen.Introspectable">

<method name="GetDescriptionLanguages">

<arg name="languageTags" type="as" direction="out"/>

</method>

<method name="IntrospectWithDescription">

<arg name="languageTag" type="s" direction="in"/>

<arg name="data" type="s" direction="out"/>

</method>

<annotation name="org.alljoyn.Bus.Secure" value="off"/>

</interface>

</node>

<node name="BusObject2">

<interface name="">

</interface>

</node>

</service>

其中：

* service节点是固定根节点，表示一个服务
* 每个服务包含且仅包含一个信息节点，以about节点表示，用来表示该服务的基本信息，如App ID，Device ID，Description等，以key/value键值对表示
* 每个服务包含一个或多个服务实例（AllJoyn中既是BusObject），以node节点表示
* 每个服务实例包含一个或多个接口，以interface节点表示
* node节点及其下的节点遵循https://allseenalliance.org/schemas/introspect.xsd，但service、about节点不遵循，需要增加这些节点的xsd

**输出**

| Return Signature | Description |
| --- | --- |
| QStatus | 返回状态值 |

**描述**

将云端的接口/服务或其他近场网络的设备的接口/服务开放订阅到本地。在订阅后，本地设备调用这些远端的接口/服务与调用本地接口/服务没有不同。

将近场服务映射到云端时，只需将所有Announce的服务映射过去即可（调用CloudCommEngine::AJPublishLocalServiceToCloud方法）。但是这里稍有不同，因为不可能所有的云端服务都要映射到本地，这中间还需要有一个**订阅**过程，所以我们称作Subscribe。在云端进行必要的安全策略检查后，再调用本方法来实施最终的服务映射。

##### AJUnsubscribeCloudServiceFromLocal

**输入**

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
| serviceAddr | s |  | 所要取消订阅的服务的地址，即服务对应的account，如某云服务地址为weather\_server\_A@nane.cn；某人家的AllJoyn智能设备向其他人开放接口，此地址为其account：thierry\_luo@nane.cn |
| introspectionXML | s |  | 接口描述的XML描述字符串，用来让ProximalCommEngine生成此接口的本地代理BusObject（及其方法、属性与信号处理机制）。此XML将包含云端服务（有时是其他近场网络的device或app）的基本信息（AllJoyn中的AboutData），以及所有根BusObject描述信息 |
|  |  |  |  |

**输出**

| Return Signature | Description |
| --- | --- |
| QStatus | 返回状态值 |

**描述**

将订阅到本地的云端服务/接口从本地删除掉。

#### 信号

| Signal name | Parameters | | Sessionless | Description |
| --- | --- | --- | --- | --- |
| Announce | Name | Signature | yes | This signal is used to announce the application information and the service interfaces that it supports. The following information is provided in the signal:   * version -- Version number of the About interface. * port -- Session port. The app will listen on this port for incoming sessions. * objectDescription --Array of object paths and the list of supported interfaces provided by each object. * metaData -- Metadata. All the fields listed in Table 2 with a yes value in the Announced column are provided in this signal. |
| version | q |
| port | q |
| objectDescription | a(oas) |
| metaData | a{sv} |

#### Introspection XML

### 面向IoTivity网络的接口（待计划）

### 面向MQTT网络的接口（待计划）

## 云端通信模块（CloudCommEngine）

### 面向AllJoyn网络的接口

#### 接口名

|  |  |  |
| --- | --- | --- |
| Interface name | Version | Secured |
| cn.nane.IoTCloud.Gateway.CloudCommEngine.AllJoynEngine | 1 | no |

#### 属性

#### 方法

##### AJCloudMethodCall

**输入**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter name | Signature | List of values | Description |
| addr | s |  | 调用的云端接口的地址：  如果代表云端接口/服务，则由云端服务AS地址+接口地址组成，如weather\_server\_A@nane.cn/CurrentWeather  /CurrentTemperature  如果代表其他近场网络终端设备的接口/服务，则由其Gateway IMS Account+AllJoyn地址（四部分）组成，如thierry\_luo@nane.cn/BusName/ObjectPath  /InterfaceName/MethodName |
| para | av |  | 调用参数数组 |
| callee | s |  | 唯一标识云端/远端调用者的IMS Account，如controlpanel\_server@nane.cn。这个callee有可能是某个IMS AS的地址，如controlpanel\_server@nane.cn，也有可能是其他近场网络Gateway的IMS Account，如thierry\_luo@nane.cn（IMS Account是sip:thierry\_luo@nane.cn的Gateway Agent所管辖的近场网络）。 |
| localSessionID | x |  | 代表本地近场网络的Session唯一ID |
| reply | av | out | 返回值的数组 |
| cloudSessionID | s | out | 代表云端到Gateway的Session唯一ID，由云端返回 |

调用此方法的调用者（一般来说是ProximalCommEngine）是如何知道此调用的被调用者（callee）是谁呢：

* 如果这个调用是AllSeen标准Service Framework中的调用（包括标准Service如Control Panel，Notification，Configuration，以及其他AllSeen Working Group所扩展的Service Framework），则可直接从初始化的配置文件中得到，如Control Panel服务的调用则送往controlpanel\_server@nane.cn。如何知道此调用是AllSeen标准Service Framework调用呢？可从接口名或Session Port中得到；
* 如果这个调用不是AllSeen标准Service Framework中的调用，而是第三方应用的接口/服务调用，或其他近场网络Gateway的接口调用，则这个callee的地址需要事先在这些接口订阅到本地时存储起来。这些云端接口在被调用前需要映射到本地（如通过订阅服务，或通过第三方应用进行映射，参见ProximalCommEngine中面向AllJoyn网络的接口中的AJSubscribeCloudServiceToLocal方法），ProximalCommEngine则负责管理这个映射，并代理这些接口在本地提供服务。

**输出**

| Return Signature | Description |
| --- | --- |
| QStatus | 返回状态值 |

**描述**

近场网络的设备调用云端服务时调用此接口。一般来说，云端服务需要首先通过ProximalCommEngine::AJSubscribeCloudServiceToLocal方法来映射到近场网络。对于近场网络的设备来说，在映射过后，这些服务跟其他近场设备提供的服务没有区别。在近场设备调用这些服务时，ProximalCommEngine只需要通过映射找到相应的callee地址，然后将调用通过AJCloudMethodCall转向云端。

##### AJPublishLocalServiceToCloud

**输入**

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
| serviceAddr | s |  | 所要映射的服务的地址，即服务对应的account地址，如某人家的AllJoyn智能设备向其他人开放接口，则此地址为gateway的account：thierry\_luo@nane.cn；某云服务的account：weather\_server\_A@nane.cn |
| introspectionXML | s |  | 服务描述的XML描述字符串，将直接存储到云端（Prosence Server/XDM Server）。此XML将包含服务（其实是device或app）的基本信息（AllJoyn中的AboutData），以及所有根BusObject描述信息 |

introspectionXML的设计请参见3.1.1.3.2。

**输出**

| Return Signature | Description |
| --- | --- |
| QStatus | 返回状态值 |

**描述**

将本地的接口/服务开放映射到云端。在映射后，将可被云端或其他近场网络设备调用（在某种安全策略的控制下）。

一般来说，在AllJoyn网络中，某个服务被About::Announce后，即可被映射到云端。如果云端希望调用这些服务，有可能还需要一些云端的安全策略检查。

##### AJDeleteLocalServiceFromCloud

**输入**

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
| serviceAddr | s |  | 所要映射的接口/服务的地址，由接口/服务对应的account和接口地址组成，如某人家的AllJoyn智能设备向其他人开放接口thierry\_luo@nane.cn/BusName/ObjectPath |
| introspectionXML | s |  | 服务描述的XML描述字符串，将直接存储到云端（Prosence Server/XDM Server）。此XML将包含服务（其实是device或app）的基本信息（AllJoyn中的AboutData），以及所有根BusObject描述信息 |
|  |  |  |  |

**输出**

| Return Signature | Description |
| --- | --- |
| QStatus | 返回状态值 |

**描述**

将映射到云端的本地服务/接口从云端删除。

##### AJSubscribe

**输入**

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
| remoteAccount | s |  | 所要订阅的远端用户帐号，如某人家的AllJoyn智能设备向其他人开放接口thierry\_luo@nane.cn |
|  |  |  |  |

**输出**

| Return Signature | Description |
| --- | --- |
| QStatus | 返回状态值 |

**描述**

订阅远端用户的所有接口/服务。

##### AJUnsubscribe

**输入**

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
| remoteAccount | s |  | 所要取消订阅的远端用户帐号，如某人家的AllJoyn智能设备向其他人开放接口thierry\_luo@nane.cn |
|  |  |  |  |

**输出**

| Return Signature | Description |
| --- | --- |
| QStatus | 返回状态值 |

**描述**

取消订阅远端用户的所有接口/服务。

#### 信号

| Signal name | Parameters | | Sessionless | Description |
| --- | --- | --- | --- | --- |
| Announce | Name | Signature | yes | This signal is used to announce the application information and the service interfaces that it supports. The following information is provided in the signal:   * version -- Version number of the About interface. * port -- Session port. The app will listen on this port for incoming sessions. * objectDescription --Array of object paths and the list of supported interfaces provided by each object. * metaData -- Metadata. All the fields listed in Table 2 with a yes value in the Announced column are provided in this signal. |
| version | q |
| port | q |
| objectDescription | a(oas) |
| metaData | a{sv} |

#### Introspection XML

### 面向IoTivity网络的接口（待计划）

### 面向MQTT网络的接口（待计划）

## 调度模块（ConnectorEngine）