Gateway Agent HLD

Letv

Junjie Zhao

1.1. 介绍

1.2. 文件范围

文件体现了 AllJoyn Gateway Agent 的系统级设计信息。相关的接口和 API 设计体现到了功能级。接口和 API 实际的定义不在这份文档范围内

1.3. 文档架构

文档是按如下章节安排的：

章节 1——介绍：提供介绍信息包括：范围，文档架构，定义和缩写，参考文献，文档版本历史和设计范围。

章节 2——Gateway Agent 设计：体现了 AllJoyn Gateway Agent 的系统设计包括系统体系结构，调用流程和功能设计。

章节 3——App Management：体现了 Gateway Agent 第三方应用的管理设计。

章节 4——性能：体现了 Gateway Agent 的性能需求。

章节 5——需求：体现了 AllJoyn Gateway Agent 的系统需求。

章节 6——配置参数：体现了 Gateway Management 应用的配置参数。

1.4. 参考文件

Gateway Agent 1.0 PRD

1.5. 缩写和词汇

缩写/词汇	描述
AllJoyn frame	开源 P2P 框架允许底层网络概念和 API 的抽象
AllJoyn service framework	AllJoyn 平台提供的一个服务框架
Consumer	通过 AllJoyn 平台消费服务的 AllJoyn 应用
GM App	Gateway Management App
GUID	Globally Unique Identifier. 通过一种冲突概率可以忽略的随机方法产生的一个 128 位的标识符
GW	Gateway
IoE	Internet of Everything
PM App	Package Manager App
Producer	通过 AllJoyn 平台提供服务的 AllJoyn 应用
SLS	Sessionless Signal
WKN	Well-Know Name

1.6. 产品需求

这个 HLD 只专注于 Gateway Agent 1.0 PRD 里的 UC1。主要设计产出是 Gateway Management 应用和 AllJoyn 总线的政策执行。任何第三方应用的 PRD 需求不再考虑范围内。

1.7. 设计范围

1.7.1. 设计假定

Gateway Management 应用和第三方是分开的软件运行在 Gateway Agent 上，通过 AllJoyn router 通信。这样的结构体系使得这两个应用已不同的权限分开运行，也就是说，Gateway Management 应用以更高的权限运行。因为第三方应用是被服务提供者（或其他第三方实体）提供的，还因为它连接着云端，所以它被认为是不可信的。

1.7.2. 当前依赖关系

Gateway Agent 设计假设依赖于如下依赖关系。

表 1-1 Gateway Agent 设计依赖关系

依赖关系	版本
Security feature	1.0
Notification service framework	1.1
About feature	1.0

1.7.3. 持续依赖关系

这里有一些持续的依赖关系（列在下面）可能影响 Gateway Agent 设计。在设计成熟且相关的产品需求被定义了以后，HLD 将被按需补充。

- Security 2.0

1.7.4. 超出范围

当前的 HLD 不考虑以下项目。

表 1-2 超出范围的项目

超出范围项目	解决方案
Gateway Management 应用的升级	应用安装/升级设计会包括第三方应用的安装/升级/卸载。体系结构在将来会被设计的更灵活来适应 Gateway management 应用。
Gateway Management 应用相关的接口的错误回应代码	将会作为实际接口定义的一部分处理
Gateway Management 应用和 Package Manger 应用间的接口	将会作为实施阶段的一部分处理

2. Gateway Agent 设计

2.1. Overview

在临近 AllJoyn IoE 网络，producer 设备暴露像 notification，控制其他临近网络里的 consumer 设备这样的功能。这使得 consumer 设备接收 producer 设备端关于事件和状态改变的通知，并且展示给用户。这也使得用户发起的或机器发起的控制动作可以在 producer 设备端被执行，例如开关设备，更新设备设置等。宜有一套机制去远程控制这样的设备功能使得用户在临近网络之外可以无缝的体验接收通知，然后或控制设备。也适用于家庭自动化用例。Gateway Agent 是被设计来提供这样一套机制的。

图 2-1 展示了远程访问临近网络里的设备提供的服务的环境体系结构。

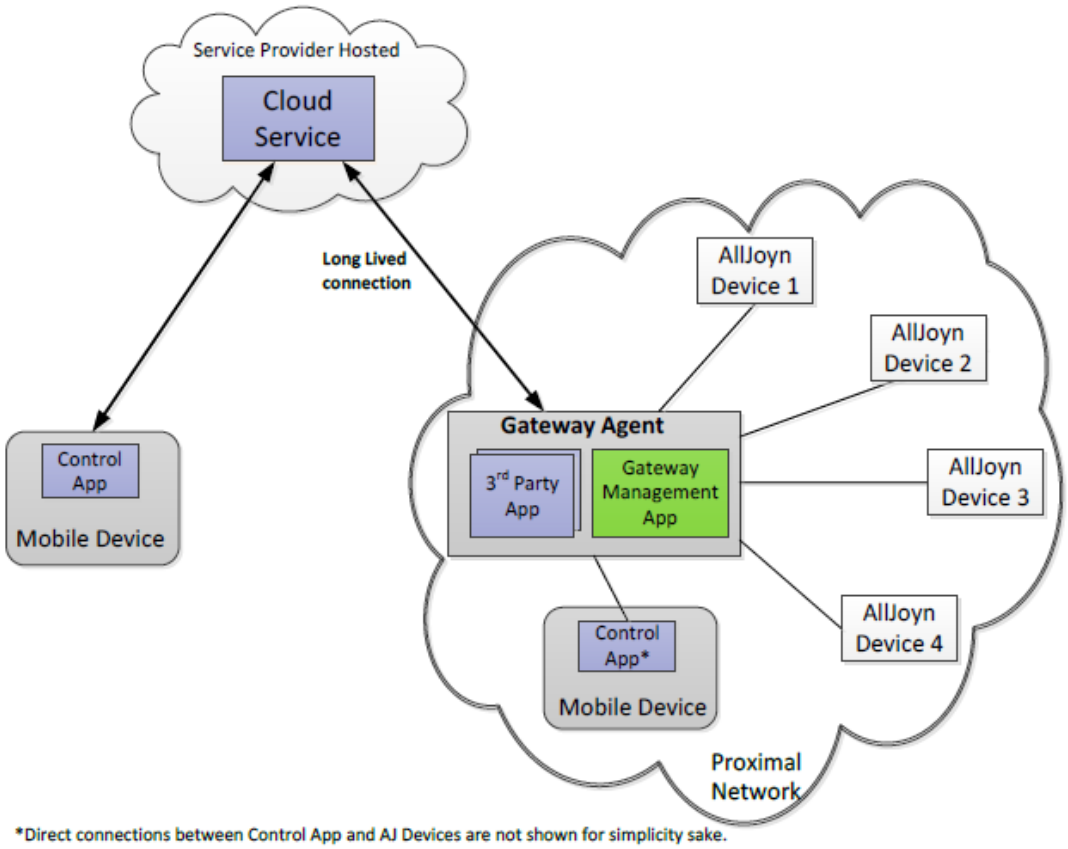


Figure 2-1 Gateway Agent context architecture

用户登录一个服务提供商来远程的访问 AllJoyn 设备服务。服务提供商将通过能用来控制设备或发送/接送通知的一个 Control app 实现远程服务访问。当在临近网络中的时候，Control app 将很可能通过 AllJoyn 点对点通信完成设备间的交互。当在临近网络之外的时候，Control app 将于一个托管于服务提供商的云服务交互。

一个新的 Gateway Agent 组件被加进了临近网络来实现远程访问。Gateway Agent 有一个单独的 Gateway Management app（GM app）和一个或多个第三方应用。所有这些应用都是 AllJoyn 有效的，并且与一个单独预装在 Gateway Agent 的 AllJoyn 总线交互。Gateway Management app 实现让用户去通过 Control app 定义和管理 remote profile。

一个 remote profile 列出了一系列用户可能会远程访问的设备/应用/接口。对 AllJoyn 设备的远程访问是通过 AllJoyn 总线里的 config file 控制的。Gateway Management app 的职责是适当的更新 config file，只允许可以被远程控制的 AllJoyn 设备被远程访问。Remote profile 信息通过 Gateway Management app 提供的一个 AllJoyn 接口暴露给第三方应用。

第三方应用维护一个与服务提供商云服务的长期连接。这个持续的连接时用来接收远程用 Control app 的用户或云服务本身（比如家庭自动化的情况）发起的控制行为。 第三方应用和云服务之间的通信是通过服务提供商定义的网页协议。在临近网络内，第三方应用直接与可远程控制的设备/应用/接口交互来通过预装的 AllJoyn 总线用 AllJoyn framework 控制行为。AllJoyn 总线通过 Gateway Management app 基于 remote profile 设置更新的 config file 来执行允许的规则。第三方应用也作为从临近网络到云服务传播通知的代理，反之亦然。

在临近网络之外，Control app 将与服务提供商托管的云服务交互。云服务之后会与 Gateway Agent 上的第三方应用交互。第三方应用会提供协议将服务提供商的协议转换成基于 AllJoyn 的协议，并且像如上描述的一样调用设备上的远程接口来交互。

2.2. Gateway 概念

2.2.1. Service provider（服务提供商）

服务提供商提供基于云的服务给在用户临近网络里的 IoE 设备，比如家庭自动化，家庭安保，远程控制等。

2.2.2. Remote profile

Gateway Management app 维护一个或多个用户通过 Control app 创建的 remote profile。一个 remote profile 定义了需要被远程访问的 IoE 设备/应用/接口。一个 remote profile 关联一个单独的服务提供商。

Remote profile 只能在临近网络内被通过 Control app 配置。因为安全原因，profile 配置不支持远程。

2.2.3. 远程 AllJoyn 设备/应用/接口

被列为 Gateway Management app 上 remote profile 一部分的 AllJoyn 设备/应用/接口被叫做可以远程控制的。

2.3. 系统体系结构

图 2-2 展示 Gateway Agent 详细的系统构架。每个 Gateway Agent 和一个单独的服务提供商。一个服务提供商能为远程访问 IoE 设备提供一个或多个云服务。Gateway Agent 将有一个第三方应用与给定云服务进行交互。Gateway Agent 支持多个服务提供商应用提供 IoE 设备远程访问的不同的云服务。一个临近网络里也可以有多个 Gateway Agent，每个通过不同的服务提供商支持 IoE 设备的远程访问。

为了保证服务提供商功能（不一定可信）不能操纵被 remote profile 定义的远程访问策略，第三方应用和 Gateway Management app 应该被分别安装在两个用户账户里。这两个 app 会与同一个预装在 Gateway Agent 里的 AllJoyn 总线交流。远程访问 AllJoyn 设备在 AllJoyn 总线里通过 config file 策略被限制。Gateway Management app 基于 profile 上的激活和失效来更新 config file 策略。

Gateway Management 应用像图 2-2 里描述的一样暴露详细的 AllJoyn 接口。包括：

- Control app 使用的 Profile Management 接口
- 第三方应用使用的 app 访问接口

第三方应用访问可远程访问的 AllJoyn 设备是通过执行在 AllJoyn 总线里的 config file 控制的。

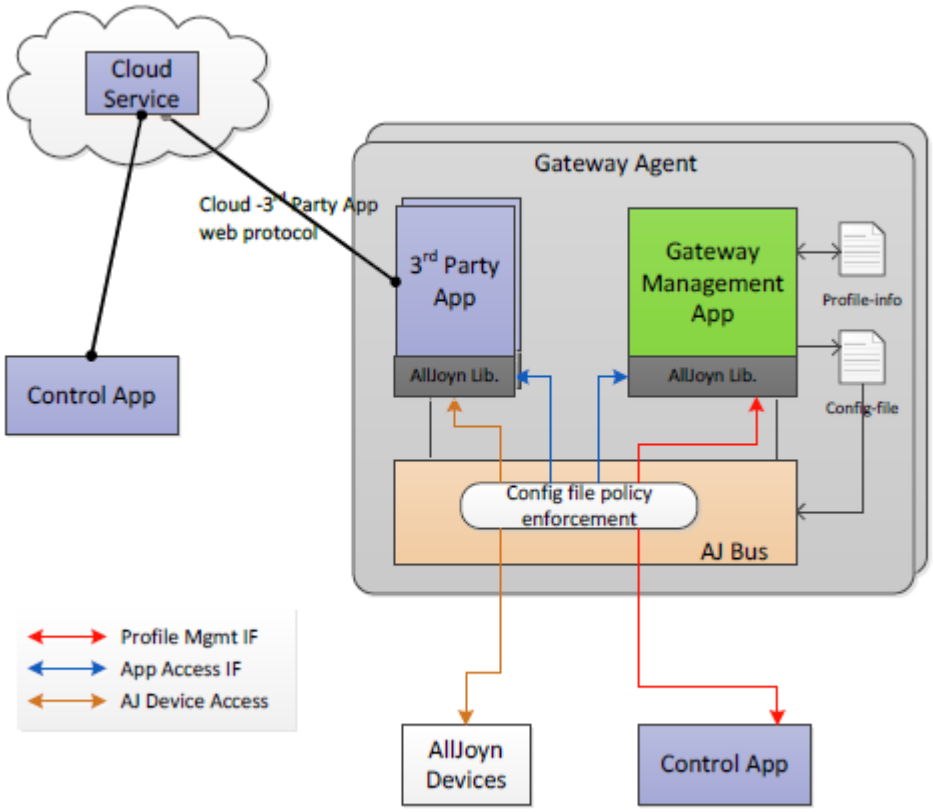


Figure 2-2: Gateway Agent architecture

2.3.1. Gateway Management 应用

Gateway Management app 提供以下只要功能：

- **AllJoyn Profile management:** Gateway Management app 应用一个安全 AllJoyn Profile Management 接口 org.alljoyn.GWAgent.ProfileMgmt。这是个通用接口，独立于服务提供商并且暴露 profile 管理功能。它通过 Gateway Management app 发出的 About announcement（声明）信号里被广播，而且被 Control app 用来发现 Gateway Management app。声明信号里‘App Name’项可以被设置来显示服务提供商-详细的 Gateway Agent，例如“Service Provide 1 AllJoyn Gateway App”。

  Profile Management 接口支持 Control app 为了远程访问，去配置作为 profile 一部分的设备/应用/接口。Gateway Management app 也负责永久储存 profile data。可以有多个同时激活的 profile 在 Gateway Management app 上。在 Control app 激活一个 profile 的基础上，Gateway Management app 也负责更改 config-file 的策略（AllJoyn 总线用来认证第三方应用发起的设备访问）。

Gateway Management app 为了创建 config file 策略规则利用关联了第三方应用的 UNIX 用户 ID。对于 AllJoyn 设备，Gateway Management app 在 config file 策略规则里使用他们的 unique name。为了获取可远程操作的 AllJoyn app 的最新的 unique name，Gateway Management app 监听这些设备的 AllJoyn 声明。每当可远程控制的设备应用的 unique name 改变，Gateway Management app 负责更新 config file。

注意：

一个给定的设备/应用/接口可以作为多个 profile 的部分被配置。在这种情况下，如果那个设备/应用/接口上的服务被通过两个 profile 不同的 file 调用，最后的运作会被保留。

- **App access for third-party apps：** Gateway Management app 暴露一个应用访问接口给第三方应用来取得 profile 数据从而激活分配给那个第三方应用的 profile。用例是提供已经为远程访问配置过的设备/应用/接口的详细信息给第三方应用，然后最终给云服务。这也接口也支持 profile 更新指示（从 Gateway Management app 到第三方应用）和连接状态更新（从第三方应用到 Gateway Management app）。

  应用访问接口包含 Gateway Management app 和第三方应用之间的双向通信。因为 Gateway Management app 和第三方应用是和同一个 AllJoyn 总线交流的，所以没有必要为了建立通信路径在他们之间建立 AllJoyn 会议。只要这些应用知道彼此的终端地址（well-known name，或 WKN），他们就能彼此通信。Gateway Management app 和第三方应用在总线上注册如下 WKN，并用这些 WKN 和其他 app 通信。

  Gateway Management app WKN:”org.alljoyn.GWAnent.GMApp”

  第三方应用 WKN:”org.alljoyn.GWAgent.ThirdPartyApp.<third-party app id>”

  出于安全原因，一个给定的第三方应用应该只能访问跟他相关联的 profile 数据。这是通过把为每个第三方应用维护的 profile 作为 Gateway Management app 上不同服务对象的部分来实施的。服务对象有一个采用格式/Profiles/<third-party app id>的 well know 对象路径。通过执行 Gateway Management app 写的 config file 策略，每个第三方应用只被允许访问相应的服务对象。

2.3.2. 第三方应用

第三方应用负责如下功能：

- **和云服务的连接管理：**第三方应用基于云服务支持的协议与云服务建立持续的连接。他维护自己一系列的连接相关数据，例如云服务 URL，传输类型（例如 WebSocket，TCP socket）etc。他也维护任何与云服务连接需要的授权证书。
- **和云服务的数据交换：**第三方应用用服务提供商特定的网络协议让云服务和临近网络交换数据来连入输出通信量。这个协议可以是基于标准的，比例 HTTP/REST 使用 XML/REST 或者可以是云服务支持的专有协议。
- **协议转换：**第三方应用在云服务与临近网络 AllJoyn 设备的中间位置上。它将通过基于网络的协议从云服务接收到的数据转换成 AllJoyn 形式来调用设备接口。在相反方向上，它将从 AllJoyn 设备收到的数据转换成基于网络的协议来送到云服务。
- **AllJoyn 设备访问：**第三方应用负责通过 AllJoyn framework 访问 AllJoyn 设备。注意只有在被 config file 认证可以执行的情况下，第三方应用才能对特定的设备/

应用/接口收发 AllJoyn 信息。这个 config file 是被 GM app 写的，被 AllJoyn 总线用来执行策略。在相反的方向上，只有在 AllJoyn 总线被每个 config file 策略过滤过，AllJoyn 设备发出的 AllJoyn 信号才会被第三方应用接收。

作为这功能的一部分，第三方应用还负责监听已经配置成 profile 一部分的设备/应用的 discover 声明信号，从而获取这些终端的最新的连接信息。

- **Notification service framework：**第三方应用支持 consumer 和 producer 端的 Notification service framework。Consumer Notification service framework 监听在临近网络里的 IoE 设备/应用产生的通知。这些通知在 AllJoyn 总线根据 config file 策略设置被过滤之后送去适当的第三方应用。第三方应用也能接收从云服务发来的要传到临近网络内部的通知。第三方应用将进来的 notification 构造成 notification message 然后发给连接的 AllJoyn 总线。如果被每个 config file 策略设定允许的话，AllJoyn 总线会把它发送出去。

  AllJoyn 总线里的 config file 对通知的策略设置对接收发出的通知支持基于通知类型的过滤。通知过滤策略规则是基于 Notification service framework 支持不同通知类型对应不同对象路径的。

  当在临近网络里的时候，Control app 可能会收到两次从设备发来的通知——一次是设备直接发来的，一次是从第三方应用作为外来信息通过云服务发来的。对于这种情况，Control app 负责基于 notification message Id 为 notification message 执行重复剔除。推荐服务提供商云服务不修改 notification message Id。

  在部署 Super-Agent 的情况下，Super-Agent 的声明信号会被 AllJoyn 总线每个 config filecelve 规则封锁，因此到不了第三方应用的 Consumer Notification service framework。因此，即使部署了 Super-Agent，第三方应用一直会从 producer 设备直接接收通知。

- **在 Gateway Management app 的 app 访问：**第三方应用能用 Gateway Management app 暴露出来的 App Access 接口取得相关 profile 数据并且接收 profile 更新/删除信号。Gateway Management app 将每个第三方应用的 profile 作为有一个用/Profiles/<third-party app id>格式的 well know 对象路径的独立的服务对象维护。当获取 profile 数据的时候，第三方应用应该用跟它相关的 well-known 对象路径。

  通过 App Access 接口，第三方应用提供它和云服务的连接状态给 Gateway Management app。

- **在临近网络暴露服务：**第三方应用也可以暴露一个或多个服务给临近网络里的 AllJoyn 设备。这样的服务可以完成通过 AllJoyn framework 暴露一个基于云的服务给设备（例如天气更新服务）。作为 profile 的一部分，用户一定要明确的选择哪些第三方应用里支持的服务应该被暴露来从临近 AllJoyn 网络访问。

  如果一个第三方应用会在临近网络里产生通知的话，Produce Notification service

framework 将会是其中一个被暴露的服务。访问这些接口是是被每个

2.3.3. AllJoyn bus（总线）

AllJoyn 总线负责通过 config file 给第三方软件执行访问策略。

Config file 策略执行：第三方应用直接访问临近设备上可远程操控的 AllJoyn 接口。访问这些接口是被 config file 上的允许策略授权的。这包括调用方法调用，属性的获取/设置和从服务接口接收信号，配置成 profile 的部分那样。AllJoyn 总线通过 config file 也执行仅给第三方应用访问适用的 profile 数据。就像上面描述的那样，这些策略都是被 GM App 写的。

2.3.4. Control app

Control app 是被服务提供商或附属实体开发的支持 AllJoyn 的应用。这个 app 能在临近网络里也能远程的访问它们的 IoE 设备。 Control app 应该支持如下与 Gateway Agent 相关的功能。

注意

这里提供的是我们现在看 Control app 的功能描述。由于 Control app 是在服务提供商的领域里，可能只有这些功能的一个子集被支持。

- **和服务提供商云端的用户认证：**用户登录服务提供商的 IoE 云服务。 Control app 应该提供一个方法输入用户证书然后用来认证用户和服务提供商。这将确保只有认证的用户能用 Control app 配置 Gateway Agent。
- **AllJoyn Gateway Agent discover:** Control app 应该使用基于 AllJoyn 声明的发现机制去发现 Gateway Management app。邻近网络里可以布置多个 Gateway Agent。声明信号里的 App Name 项可以被设置指示任何适用的特定的服务提供商 Gateway Agent。Control app 应该用 App Name 项来发现任何他在寻找的特定的服务提供商。
- **AllJoyn 设备发现和过滤：**Control app 应该使用基于 AllJoyn 声明的发现来发现临近网络里的设备/应用。当给一个特定的第三方应用创建 remote profile 的时候 Control app 应该基于 manifest file 内容为第三方应用过滤设备/应用/接口。假设 Control app 访问的 manifest file 是作为第三方应用安装的一部分或给已经安装的第三方应用单独下载。目的是只允许这些设备/应用/接口被作为第三方应用对应的 remote profile 的一部分预分配。
- **Profile 功能 UI：**Control app 应该提供一套 UI 给用户让用户可以在 Gateway Agent 选择 profile 配置的设备/应用/接口。
- **Profile 配置：**在发现一个想要的 Gateway Agent 之后，Control app 调用 AllJoyn profile 管理接口提供的 profile 管理方法调用来创建 remote profile。Profile name 和其他的设备/应用/接口数据一起作为 profile creation 的一部分。Profile ID 返回来响应 Gateway Management app。 Control app 应该为了 profile 将来的更新维护这个 profile ID。Control app 通过一次明确的调用来激活一个先前创建的 profile 从而支持相关的设备/应用/接口的远程访问。

Control app 能对存在的 profile（激活的或未激活的）进行更新。如果是已经激活的 profile，更新会立即生效。Control app 也能使一个以及获得 profile 失效或完全删除一个 profile。

2.3.5. 实体关系

一个 Gateway Agent 有一个单独的 Gateway Management app 和一个或多个第三方应用。Gateway Management app 支持一个或多个 remote profile。每个 remote profile 与单独的第三方应用相关。体系结构支持一个在 remote profile 和第三方应用件 1:1 的关系。图 2-3 展示了 Gateway Management app，第三方应用和 remote profiles 之间的实体关系。

每个第三方应用连接一个云服务。典型的，每个第三方应用将连接一个提供不同远程访问类型的不同的云服务。然而，一个服务提供商可以部署一个支持多后端服务的前端云服务。在这种情况下，多个第三方应用可以链接到同一个云服务。

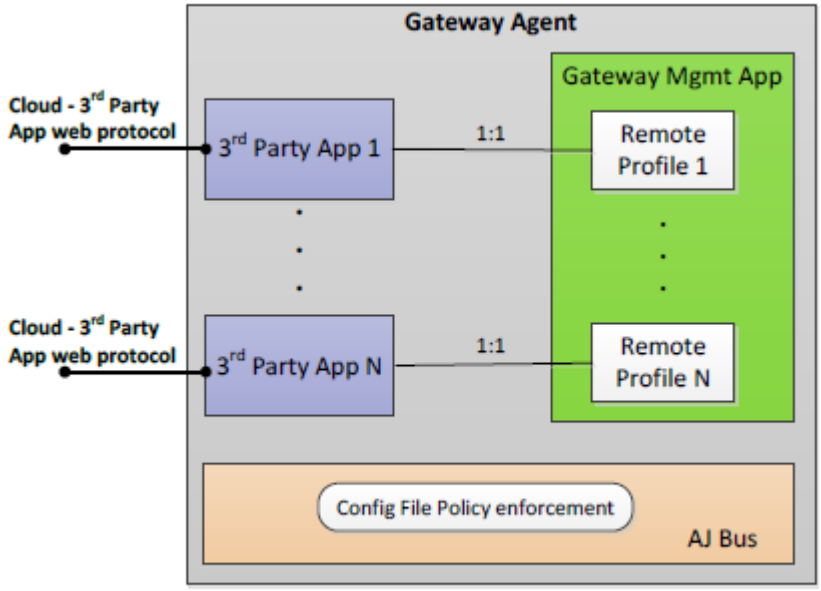


Figure 2-3 Gateway Agent Entity Relationship

2.3.6. 访问安全接口

由于显而易见的安全原因，被通过 profile 远程控制的 AllJoyn 接口是典型的安全接口。并且，所有 Gateway Management app 暴露给 Control app 的所有 AllJoyn 接口将是安全的。访问这样的安全接口是通过使用密码，PIN 或其他在 Security 1.0 里的密钥材料（基于使用的 AllJoyn 身份认证机制），描述如下。

注意

在 Security 2.0 中，安全接口访问是通过用户授权的。Security2.0 现在还在设计阶段，更详细的内容会之后加上。在 Security2.0 领域里一种介绍安全接口被 Gateway Agent 访问的可能的体系结构见附录 A。

2.3.6.1. Security 1.0

Security 1.0 使用密码，PIN 或其它密钥材料（基于使用的 AllJoyn 身份认证机制）来访问安全接口。密码，PIN 或其它密钥材料使用户在 Control app 提供的，可以作为 profile creation 的一部分或独立的。Control app 查询需要被远程控制的 AllJoyn 设备/应用支持的认证方式的类型。然后它向用户索要适当的基于从这些设备/应用收到的认证类型信息的密钥材料。Control app 负责为可远程控制的设备/应用发送密钥材料和认证类型给服务提供商的云服务。每当一个设备访问通过云服务被调用，密钥材料都会被作为请求的一部分被提供。出于安全原因，第三方应用不应该为了

设备访问存储密钥材料。

Control app 需要密钥材料访问 Gateway Management app 暴露的安全的 Profile Management 和 App Management 接口。这个密钥材料（如密码或 PIN）是被用户在 Control app 输入的，可以被 app 存储以备将来使用。

注意

一个用户可以有一个单独的密码设置给临近网络里所有的设备/应用或者可以是多个密码给 AllJoyn 设备/应用。

图 2-4 展示了输入密码和它在 Security 1.0 里关于 Gateway Agent 的用法的流程图。

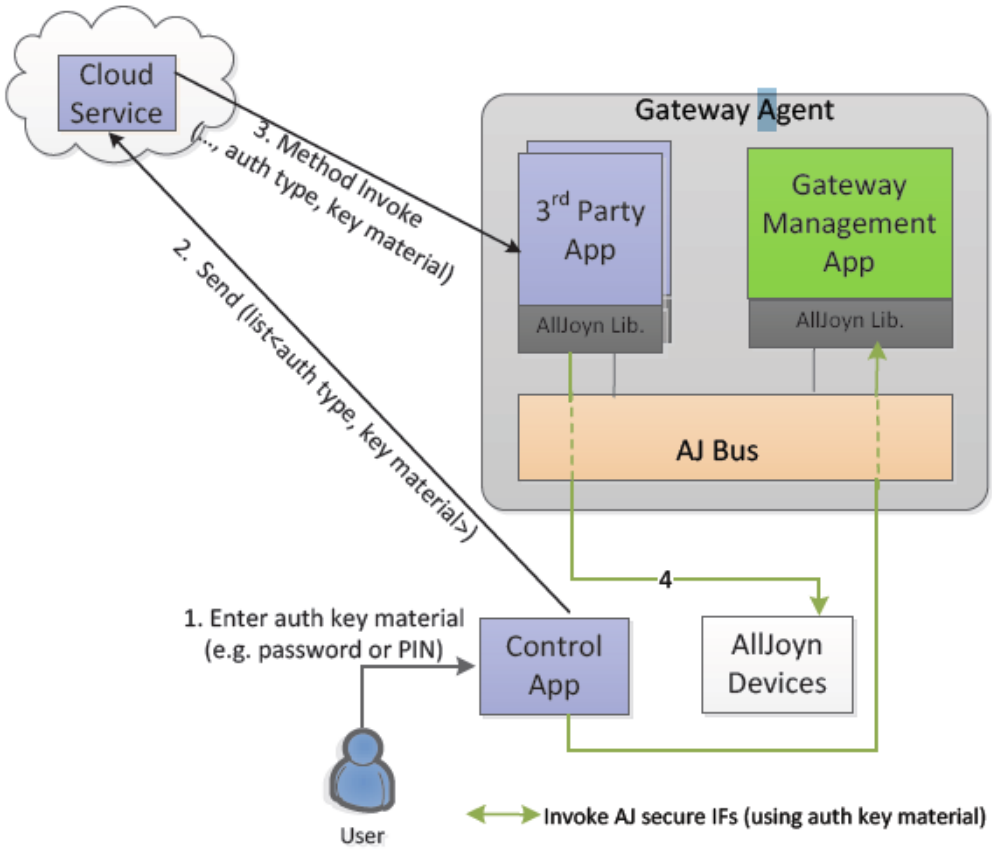


Figure 2-4 Gateway Agent Security 1.0

2.3.7. 安全 Gateway Management 应用接口

Gateway Management app 提供安全的 AllJoyn 接口给第三方应用管理和 profile 管理。Gateway Management app 为了 Security 1.0 范围里的这些接口应该支持 ALLJOYN_SRP_KEYX 身份验证机制。一个默认的密码应该被配置在 Gateway Management app 上，而且 Configuration service frame 应该在 Gateway Management app 上被支持，使得用户可以在 Control app 上修改这个密码。

2.4. 调用流程

这部分摆阔了各种 Gateway Agent 使用情况的调用流程。

2.4.1. Gateway Agent discover

Control app 会寻找一个有支持 Gateway Management 接口的 Gateway Management app 的 Gateway Agent。图 2-5 阐明了如下使用情况

- Control app 寻找 Gateway Management app 发出的声明信号里的 AppName 识别

的指定 Gateway Agent，比如 Service Provider 1 GW Agent

- Control app 想要发现任何 Gateway app

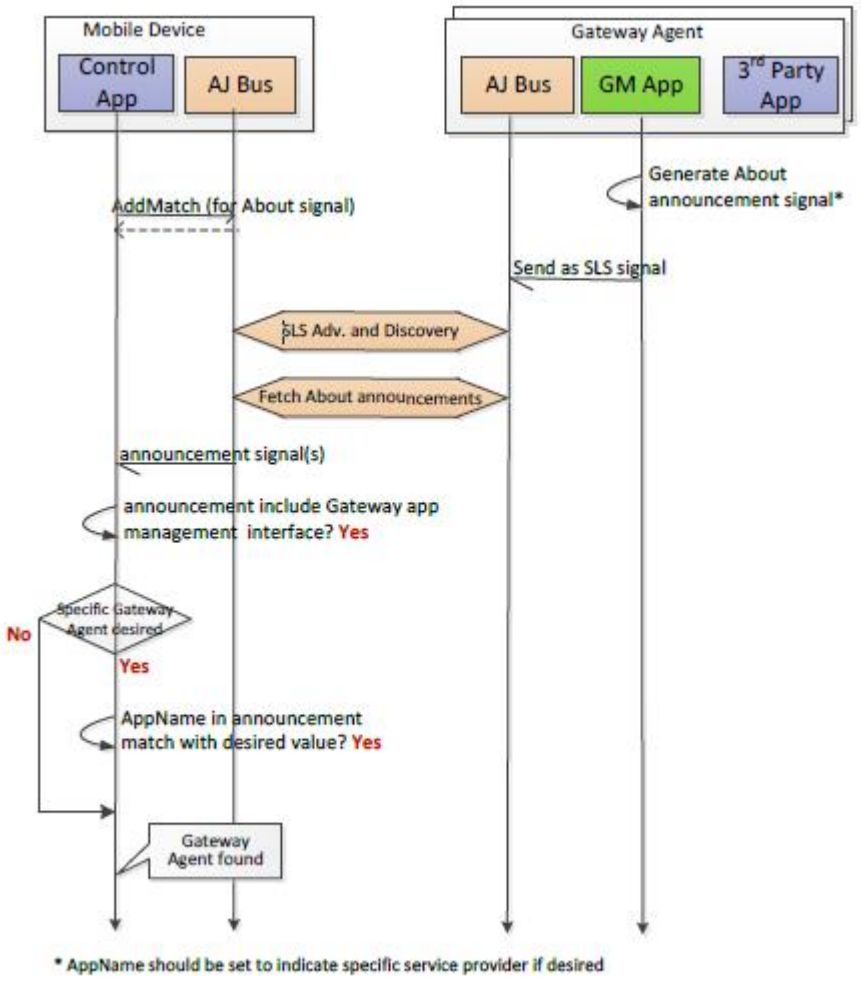


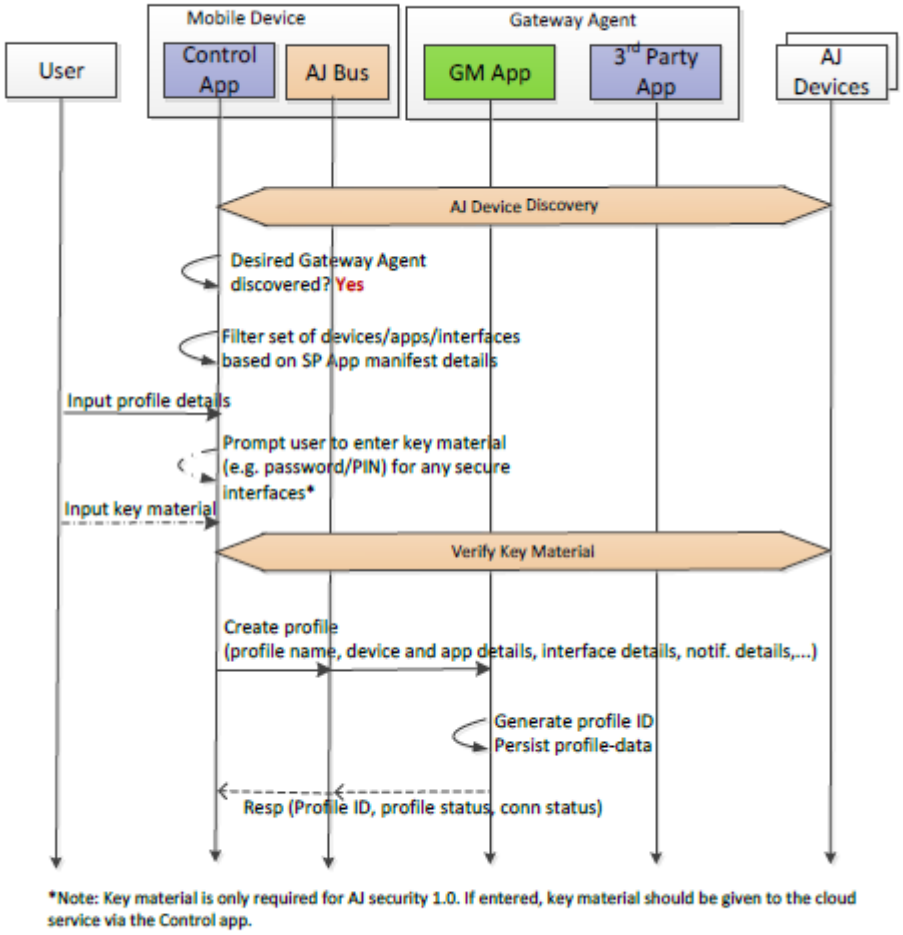
Figure 2-5 Gateway Agent discovery

2.4.2. Profile 管理

Profile 管理师被 AllJoyn Profile Management 接口提供的，支持功能如表 2-1 里所示。

2.4.2.1. 创建 Profile

2.4.2.1 Create profile



*Note: Key material is only required for AJ security 1.0. If entered, key material should be given to the cloud service via the Control app.

Figure 2-6 Create profile

2.4.2.2. 激活 Profile

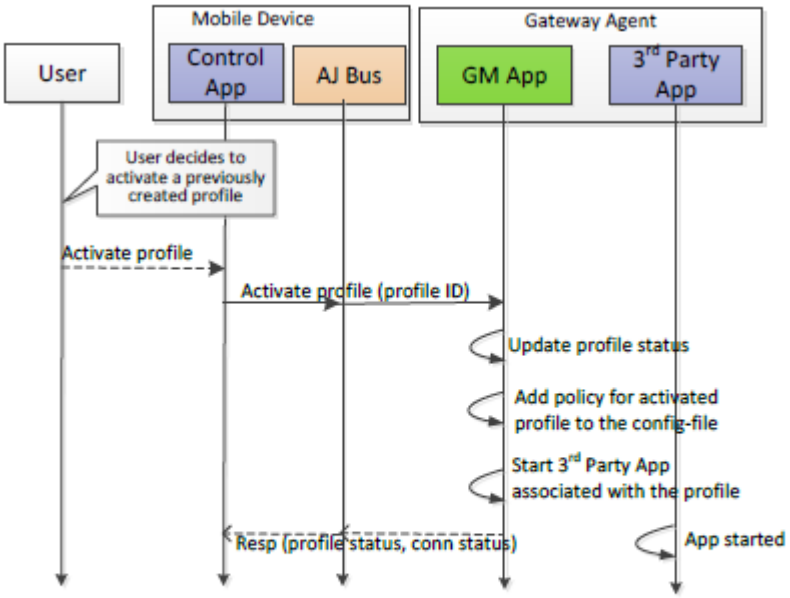


Figure 2-7 Activate profile

2.4.2.3. 第三方应用启动

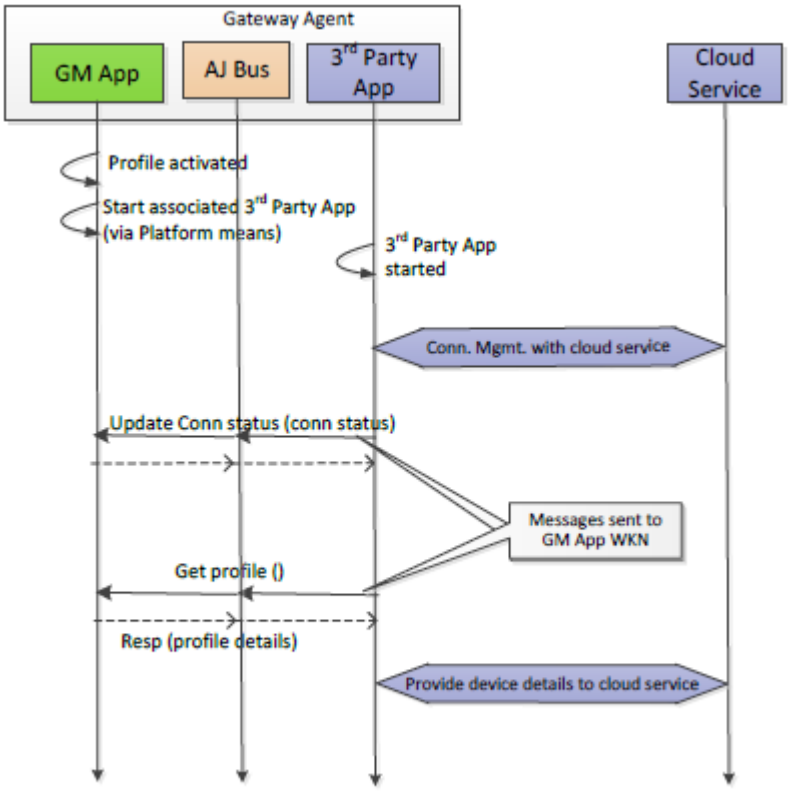


Figure 2-8 Third-party app startup

2.4.2.4. 更新 Profile

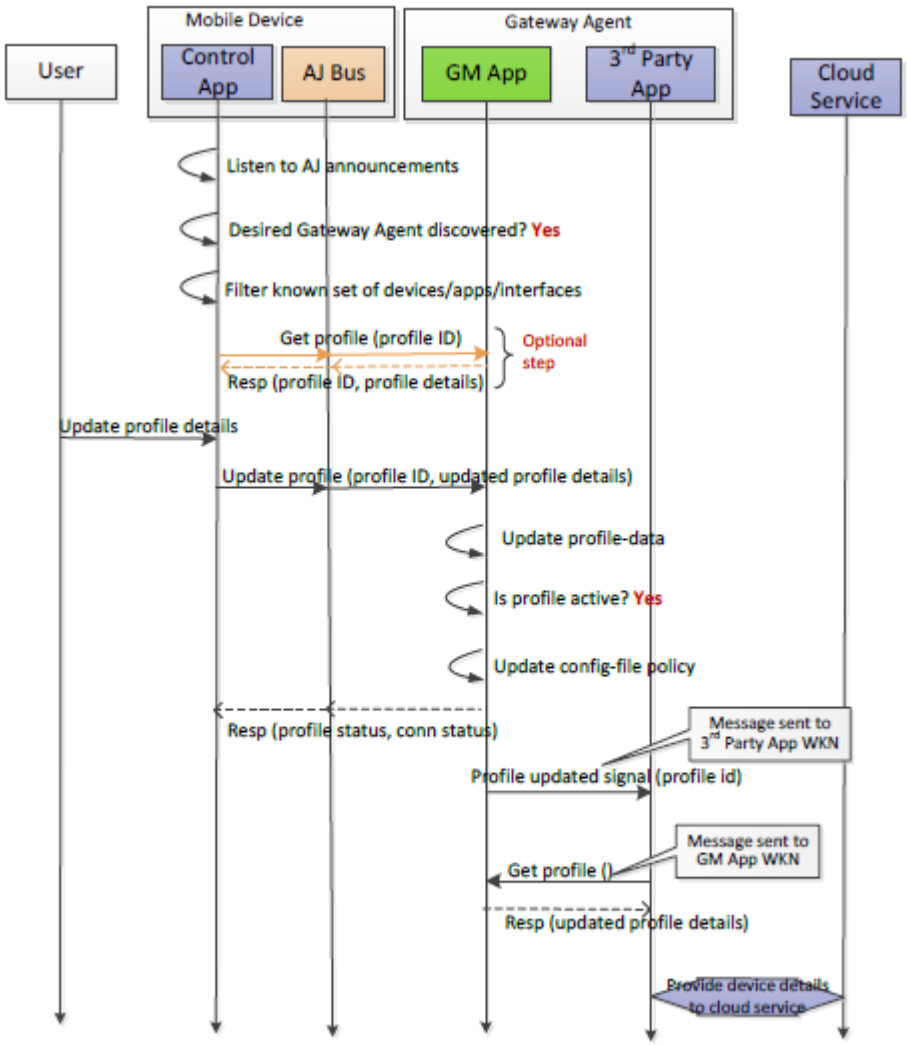


Figure 2-9 Update profile

2.4.2.5. 删除 Profile

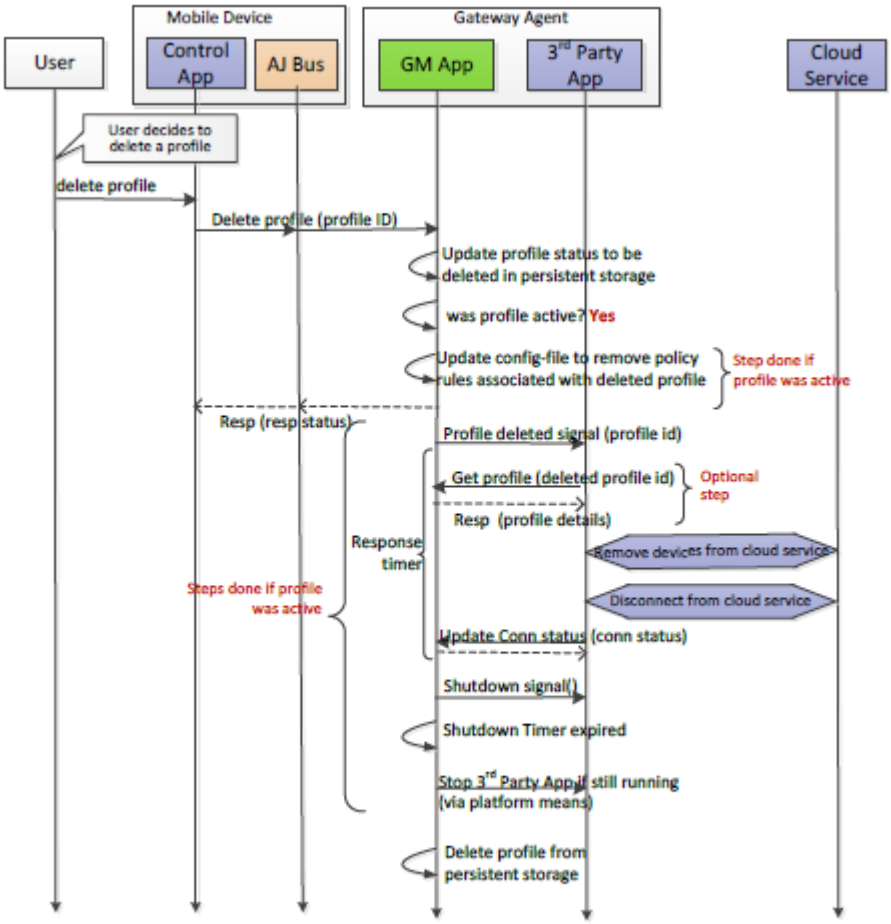


Figure 2-10 Delete profile

2.4.2.6. 删除所有 Profile

如果用户要删除 Gateway Management app 上的所有 profile，Control app 必须调用每个存在的 profile 上的 delete profile。通过 Gateway management 可以调用 Get profile list 功能获取 profile 列表。

2.4.2.7. 使 Profile 无效

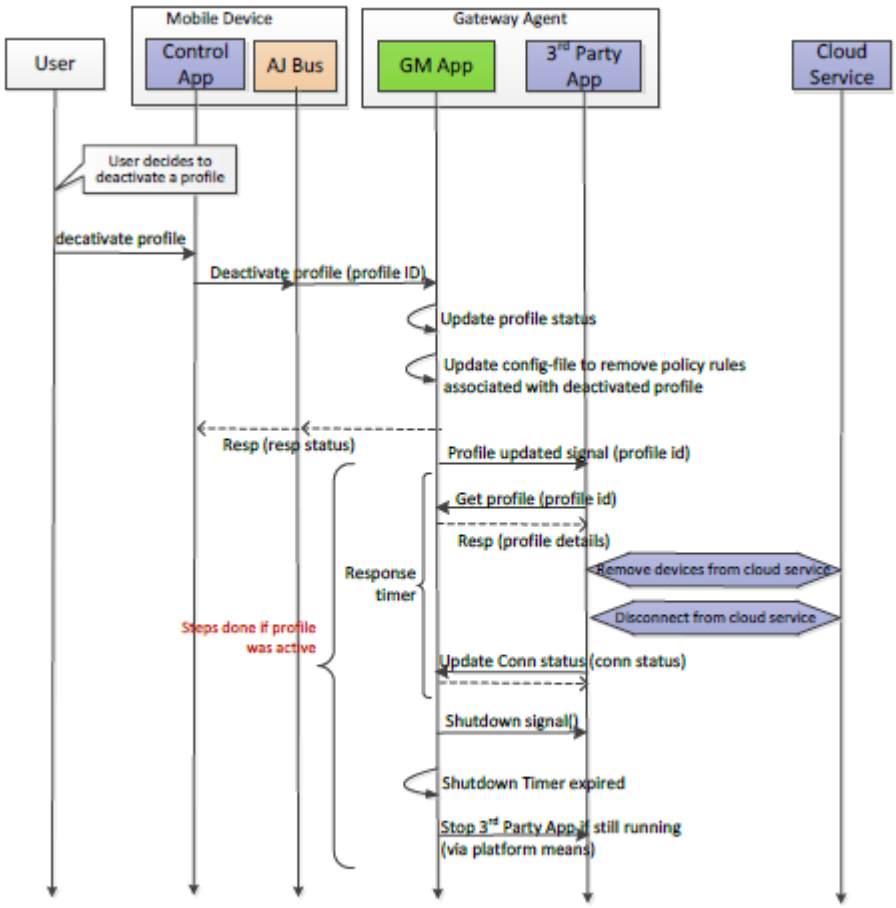


Figure 2-11 Deactivate profile

2.4.2.8. 使所有 Profile 无效

如果用户要使 Gateway Management app 上的所有 profile 失效，Control app 必须调用每个存在的 profile 上的 delete profile。通过 Gateway management 可以调用 Get profile list 功能获取 profile 列表。

2.4.3. Notification

注意

Notification 1.1 的解除（dismissal）信号是非远程控制的。

2.4.3.1. 发出通知

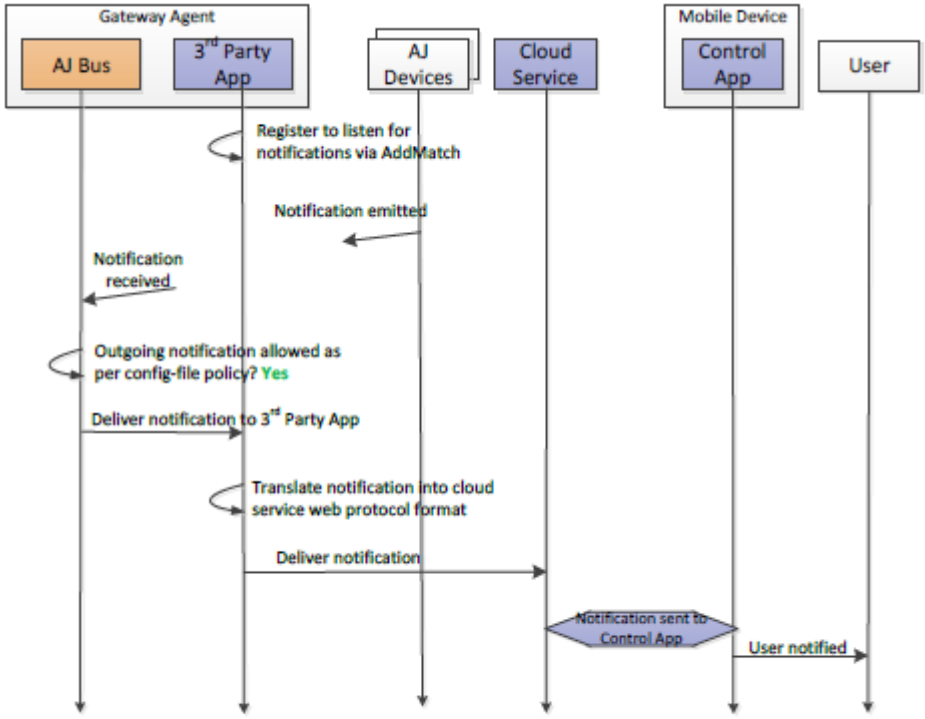


Figure 2-12 Outgoing notification

2.4.3.2. 收到通知

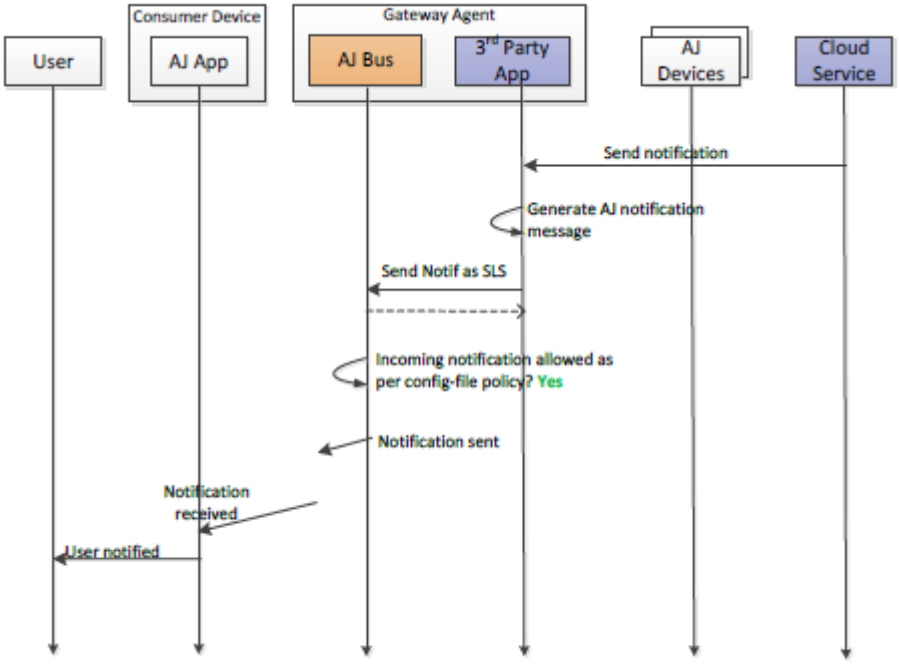


Figure 2-13 Incoming notification

2.4.4. AllJoyn 设备访问

2.4.4.1. 方法调用

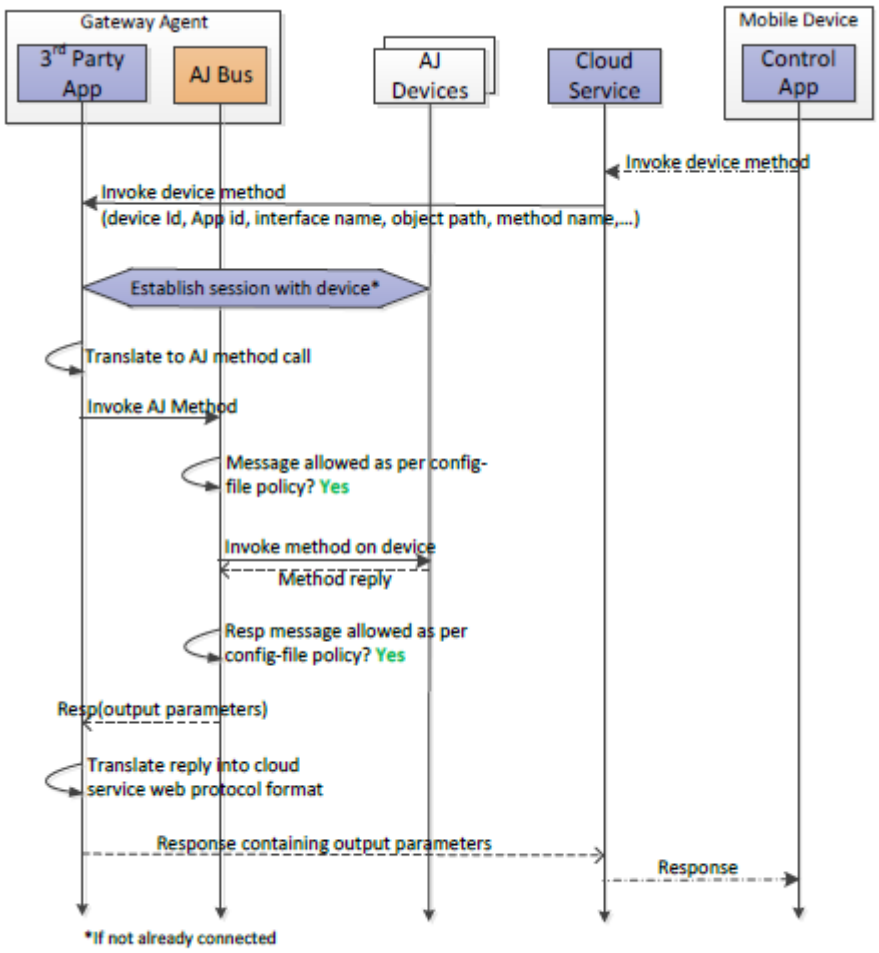


Figure 2-14 Method invocation

2.4.4.2. 获取属性

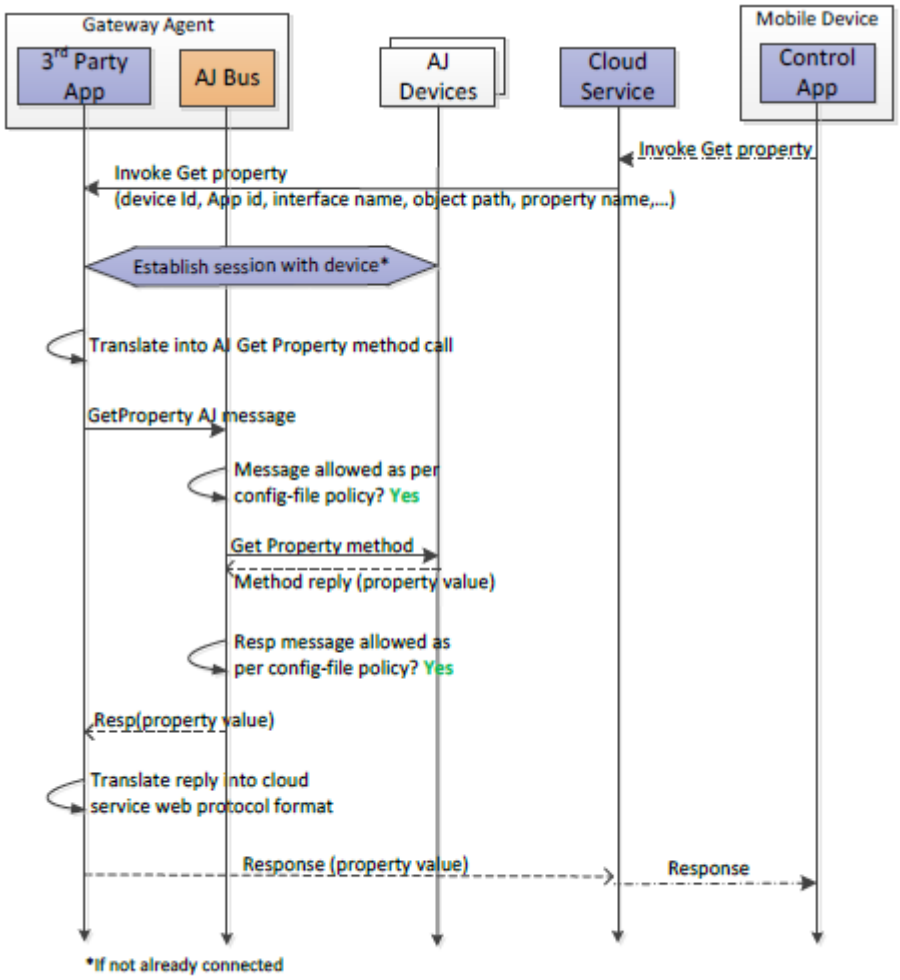


Figure 2-15 Get property

2.4.4.3. 设置属性

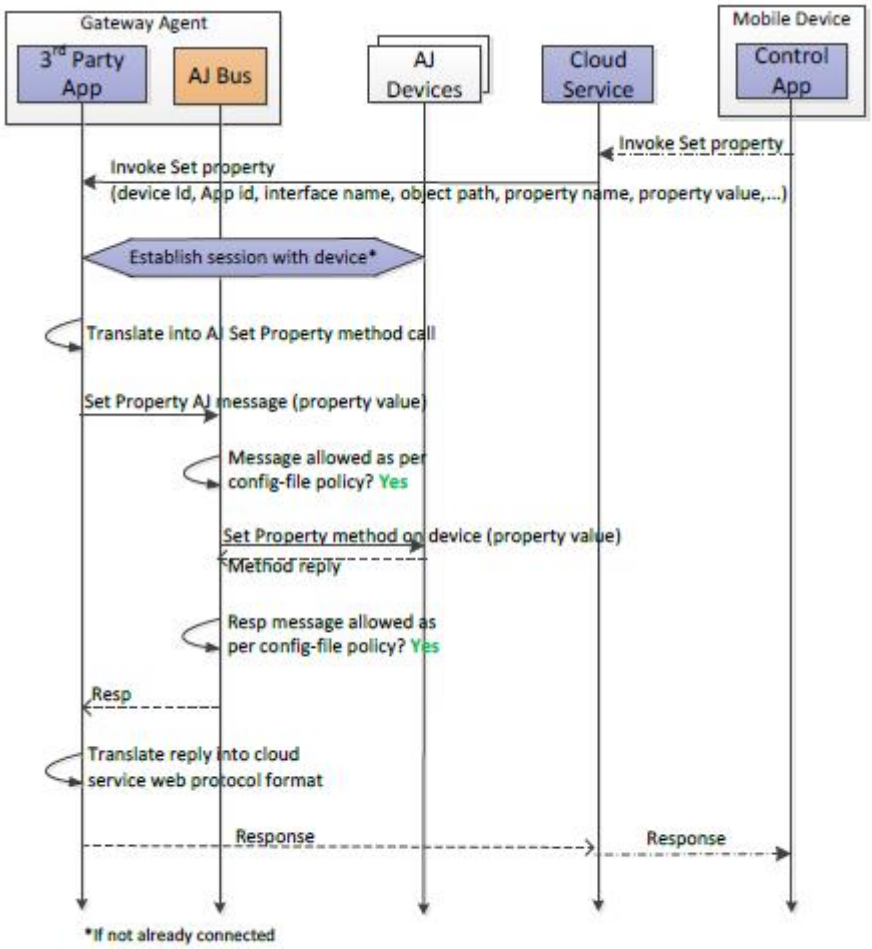


Figure 2-16 Set property

2.4.4.4. 遥控基于会议的信号

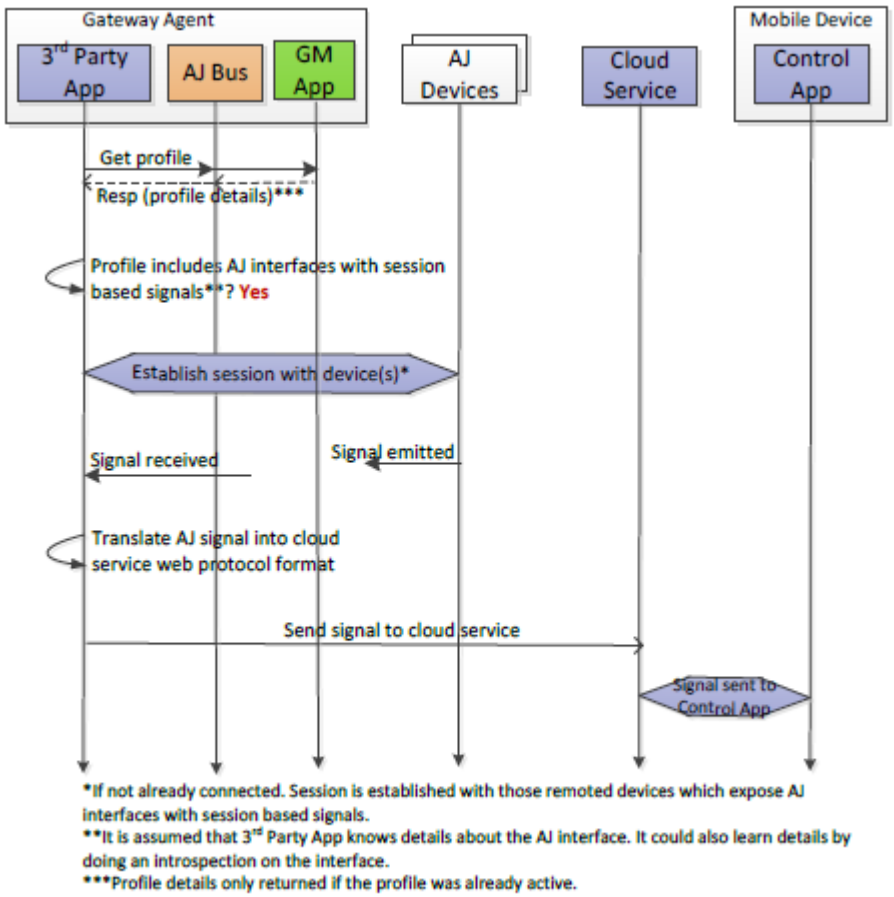


Figure 2-17 Remoting session-based signals

2.5. 功能设计

2.5.1. Profile management design

Profile 管理接口是一个支持表 2-1 里所列的 profile 管理方法的 AllJoyn 接口。

表 2-1Profile management interface functions

Functional method	Description
Create profile	创建一个新的服务提供商 profile 在 Gateway Management app 上。在创建之后，profile 保持在一个非激活状态直到明确地被激活。
Active profile	在 Gateway Management app 上激活一个当前的非激活状态的 remote profile。激活一个 profile 会导致在 AllJoyn 总线 config file 上添加相关的访问策略规则和启动相关的的第三方 app。
Get profile	从 Gateway Management app 上取得一个存在的 remote profile 信息。
GetProfileStatus	返回给定 profile 的状态
GetProfileList	取得与一个给定第三方 app 相关的一个或多个 remote profile 的列表
Update profile	在 Gateway Management app 上更新一个现存的 remote profile。对已经激活的 profile 的更新即时生效。对已激活 profile 的更新会导致 config file 按需要适当的改变。
Delete profile	在 Gateway Management app 上删除一个存在的 remote profile。激

	活的和未激活的 profile 都能被删除。删除一个 profile 会导致在 AllJoyn 总线 config file 上移除相关的访问策略规则和停止相关的的第三方 app。删除的 profile 被从永久存储中移除。
Deactive profile	在 Gateway Management app 使一个当前激活的 remote profile 失效。使一个 profile 失效会导致在 AllJoyn 总线 config file 上移除相关的访问策略规则和停止相关的的第三方 app。删除的 profile 被从永久存储中移除。

2.5.1.1. Create Profile

表 2-2 提供一个 Create profile 方法的功能定义。

注意

Gateway app 将存储设备 unique name（从接收这些 app 的声明了解到）作为 profile 数据的一部分。这需要在 config file 去创建访问策略给设备 app。对第三方应用来说，相关的用户 Id 在创建 config file 策略规则的时候被使用。

表 2-2 Create profile method

Method	Parameter		
	Parameter Name	In/Out	Description
Create profile	Profile name	In	给服务提供商 profile 的名字（用户）
	Third-Party App Name	In	与 profile 相关的第三方应用的应用名
	Third-Party App Id	In	与 profile 相关的第三方应用的 ID
	List<Exposed services>		被暴露出来让临近网络里设备访问的第三方应用提供的 AllJoyn 服务列表
	Object path	In	对象应用被列出的接口的对象路径。“*”表示列出的接口能被任何对象路径访问
	isPrefix	In	指示对象路径参数是一个对象路径前缀或完整的对象路径
	List<Interface name>	In	第三方应用暴露出的 AllJoyn 接口列表。“*”表示所有对象路径应用的接口都暴露了。
	List<Remoted Apps>		Profile 包括可远程访问的 AllJoyn 应用列表信息。
	Device Id	In	About 特征里唯一的设备 Id。
	App Id	In	About 特征里唯一的应用 Id。
	List<Interface info>		对每个应用，profile 详细列出可以远程访问的 AllJoyn 接口列表。
	Object path	In	对象应用被列出的接口的对象路径。“*”表示列出的接口能被任何对象路径访问
	isPrefix	In	指示对象路径参数是一个对象路径前缀或完整的对象路径

Method	Parameters		
	List<Interface name>	In	可远程访问的 AllJoyn 接口列表。“*”表示所有对象路径应用的接口都可远程访问。
	List<Metadata>	In(optional)	可远程访问的应用的一系列详细的元数据。
	Key	In	主要描述元数据。
	Value	In	元数据的值。
	Profile Id	Out	Gateway Management app 给 remote profile 产生的唯一的标示符。
	Profile Status	Out	表明 profile 当前状态。像表 2-16 一样设置。在 Create profile 方法的响应里应该被设为非激活。

2.5.1.2. Activate Profile

表 2-3 提供了激活 profile 方法的功能定义。与 profile 关联的第三方应用在 profile 激活时启动。启动后，第三方应用试图建立一个和云服务的连接。

表 2-3 激活 profile method

Method	Parameter		
	Parameter Name	In/Out	Description
Active profile	Profile Id	In	Remote profile 被激活的唯一标识符
	Profile Status	Out	表明 profile 当前状态。像表 2-16 一样设置。在 Active profile 方法的响应里应该被设为’active’。

2.5.1.3. Get Profile

表 2-4 提供了 Get profile 方法的功能定义。

表 2-4 Get profile method

Method	Parameter		
	Parameter Name	In/Out	Description
Get profile	Profile Id	In	Remote profile 被获取的唯一标识符
	Profile name	Out	服务提供商的给定名字。
	Third-Party App name	Out	与 profile 相关的第三方应用的应用名
	Third-Party App Id	Out	与 profile 相关的第三方应用的 ID（从声明里重取回）
	List<Exposed services>		被暴露出来让临近网络里设备访问的第三方应用提供的 AllJoyn 服务列表
	Object path	In	对象应用被列出的接口的对象路径。“*”表示列出的接口能被任意对象路径访问
	isPrefix	In	指示对象路径参数是一个对象路径前

			缀或完整的对象路径
	List<Interface name>	In	第三方应用暴露出的 AllJoyn 接口列表。“*”表示所有对象路径应用的接口都暴露了。
	List<Remoted Apps>		Profile 包括可远程访问的 AllJoyn 应用列表信息。
	Device Id	Out	About 特征里唯一的设备 Id。
	App Id	Out	About 特征里唯一的应用 Id。
	List<Interface info>		对每个应用，profile 详细列出可以远程访问的 AllJoyn 接口列表。
	Object path	Out	对象应用被列出的接口的对象路径。“*”表示列出的接口能被任何对象路径访问
	isPrefix	In	指示对象路径参数是一个对象路径前缀或完整的对象路径
	List<Metadata>	In(optional)	可远程访问的应用的一系列详细的元数据。
	Key	In	主要描述元数据。
	Value	In	元数据的值。
	Profile Status	Out	表明 profile 当前状态。像表 2-16 一样设置。在 Create profile 方法的响应里应该被设为非激活。

2.5.1.4. Get Profile status

表 2-5 提供了 Get profile status 方法的功能定义。

表 2-5 Get profile status method

Method	Parameter		
	Parameter Name	In/Out	Description
Get profile status	Profile Id	In	Remote profile 被激活的唯一标识符
	Profile name	Out	被安排给 profile 的名字。
	Profile Status	Out	表明 profile 当前状态。像表 2-16 一样设置。

2.5.1.5. Get profile list

表 2-6 提供了 Get profile list 方法的功能定义。

表 2-6 Get profile list method

Method	Parameter		
	Parameter Name	In/Out	Description
Get profile list	Third-Party App Id	In	Profile list 取回了的第三方应用的标识符
	List<Profile Info>		提供与特定第三方应用相关的 profile list

	Profile Id	Out	Remote profile 被激活的唯一标识符
	Profile name	Out	被安排给 profile 的名字。
	Profile Status	Out	表明 profile 当前状态。像表 2-16 一样设置。

2.5.1.6. Update Profile

表 2-7 提供了 Update profile 方法的功能定义。这个调用提供了完整的一系列更新的 profile 信息。

表 2-7 Update profile method

Method	Parameter		
	Parameter Name	In/Out	Description
Update profile	Profile Id	In	Remote profile 被激活的唯一标识符
	Profile name	In	被安排给 profile 的名字。
	Third-Party App name	In	与 profile 相关的第三方应用的应用名
	Third-Party App Id	In	与 profile 相关的第三方应用的 ID
	List<Exposed services>		被暴露出来让临近网络里设备访问的第三方应用提供的 AllJoyn 服务列表
	Object path	In	对象应用被列出的接口的对象路径。“*”表示列出的接口能被任意对象路径访问
	isPrefix	In	指示对象路径参数是一个对象路径前缀或完整的对象路径
	List<Interface name>	In	第三方应用暴露出的 AllJoyn 接口列表。“*”表示所有对象路径应用的接口都暴露了。
	List<Remoted Apps>		Profile 包括可远程访问的 AllJoyn 应用列表信息。
	Device Id	In	About 特征里唯一的设备 Id。
	App Id	In	About 特征里唯一的应用 Id。
	List<Interface info>		对每个应用，profile 详细列出可以远程访问的 AllJoyn 接口列表。
	Object path	In	对象应用被列出的接口的对象路径。“*”表示列出的接口能被任何对象路径访问
	isPrefix	In	指示对象路径参数是一个对象路径前缀或完整的对象路径
	List<Interface name>	In	第三方应用暴露出的 AllJoyn 接口列表。“*”表示所有对象路径应用的接口都暴露了。
	List<Metadata>	In(optional)	可远程访问的应用的一系列详细的元数据。

	Key	In	主要描述元数据。
	Value	In	元数据的值。
	Profile Status	Out	表明 profile 当前状态。像表 2-16 一样设置。

2.5.1.7. Delete Profile

表 2-8 提供了 Delete profile 方法的功能定义。

表 2-8 Delete profile method

Method	Parameter		
	Parameter Name	In/Out	Description
Delete profile	Profile Id	In	服务提供商 profile 被删除的唯一标识符
	Resp Status	Out	表明 profile 删除的成功或失败。

2.5.1.8. Deactivate Profile

表 2-9 提供了 Deactive profile 方法的功能定义。

表 2-9 Deactive profile method

Method	Parameter		
	Parameter Name	In/Out	Description
Deactive profile	Profile Id	In	服务提供商 profile 被删除的唯一标识符
	Profile Status	Out	表明 profile 当前状态。像表 2-16 一样设置。在 Deactive profile 方法的响应里应该被设为非激活。

2.5.2. App Access interface

第三方应用和 Gateway Management 应用通过 AllJoyn App Access 接口交流。由于两个应用都连接在相同的 AllJoyn 总线上，所以不需要 AllJoyn 会议。两个应用之间使用 WKNs（Well-known name）交流：

- Gateway Management app WKN: org.alljoyn.GWAgent.GMApp
- Third-party app WKN: org.alljoyn.GWAgent.ThirdPartyApp.<third-party app id>

App Access 接口允许第三方应用获取 profile 数据和提供连接状态更新给 Gateway Management app。Gateway Management app 利用这个接口通知第三方应用关于 profile 更新和删除。Gateway Management app 也支持关闭信号作为接口的一部分去关闭第三方应用。

App Access 接口支持的功能方法/信号总结在表 2-10 里。

表 2-10 App Access interface functions

Functional method	Description
Get profile	返回与索要 profile 数据的第三方应用相关的现有已激活 remote profile 的数据。如果没有激活的 profile，返回一个错误响应。
Update	被第三方应用调用来更新它在 Gateway Management app 的云服务

connection status	连接状态。这个方法不需要发送响应。
Functional Signal	Description
Profile updated	提供与第三方设备相关的特定 profile 被更新的提示。每有一个激活的 profile 被更新，指示被 Gateway Management app 发送。这是一个单播信号仅被递送给相关的第三方应用。
Profile deleted	提供与第三方设备相关的特定 profile 被更新的提示。每有一个激活的 profile 被删除，指示被 Gateway Management app 发送。这是一个单播信号仅被递送给相关的第三方应用。
Shutdown App	提供一个让第三方应用执行正常关闭的触发器。

2.5.2.1. Get Profile

表 2-11 提供了 Get profile method 的功能定义。第三方应用能请求一个特定的 Profile（通过提供 profile Id），或他能请求一系列与这个第三方应用相关的所有 profile。这个方法应该只返回和第三方应用相关的激活的 profile。如果被请求的 profile 不是激活的（profile Id 被提供的情况下）或没有与第三方应用相关的激活的 profile，返回一个错误响应。

Gateway Management app 会将每个第三方应用对应的 profile 作为单独的服务对象进行维护。正如早先解释的，由于安全原因，要确保每个第三方应用只能访问自己的 profile 数据。这些服务对象有一个格式为/Profiles/<third-party app id>的 well-known 对象路径。当发送 Get profile message 给 Gateway app 的时候，第三方应用应该使用 well-known 对象路径。

表 2-11 Get profile method

Method	Parameter		
	Parameter Name	In/Out	Description
Get profile	Profile Id（optional）	In	服务提供商 profile 去取得的唯一标识符
	List<Profile Info>		回应包括与第三方应用相关的一个或多个激活的 profile 详细资料。
	Profile name	Out	给服务提供商 profile 的名字（用户）
	Third-Party App Name	Out	与 profile 相关的第三方应用的应用名
	Third-Party App Id	Out	与 profile 相关的第三方应用的 ID（从声明里取回）。
	List<Exposed services>		被暴露出来让临近网络里设备访问的第三方应用提供的 AllJoyn 服务列表
	Object path	Out	对象应用被列出的接口的对象路径。“*”表示列出的接口能被任何对象路径访问
	isPrefix	In	指示对象路径参数是一个对象路径前缀或完整的对象路径
	List<Interface name>	In	第三方应用暴露出的 AllJoyn 接口列表。“*”表示所有对象路径应用的接

			口都暴露了。
	List<Remoted Apps>		Profile 包括可远程访问的 AllJoyn 应用列表信息。
	Device Id	Out	About 特征里唯一的设备 Id。
	App Id	Out	About 特征里唯一的应用 Id。
	List<Interface info>		对每个应用，profile 详细列出可以远程访问的 AllJoyn 接口列表。
	Object path	Out	对象应用被列出的接口的对象路径。“*”表示列出的接口能被任何对象路径访问
	isPrefix	In	指示对象路径参数是一个对象路径前缀或完整的对象路径
	List<Interface name>	In	第三方应用暴露出的 AllJoyn 接口列表。“*”表示所有对象路径应用的接口都暴露了。
	List<Metadata>	In(optional)	可远程访问的应用的一系列详细的元数据。
	Key	In	主要描述元数据。
	Value	In	元数据的值。
	Profile Status	Out	表明 profile 当前状态。像表 2-16 一样设置。

2.5.2.2. Update connection status

表 2-12 提供 Update connection status 方法的功能定义。

表 2-12 Update connection status 方法

Method	Parameter		
	Parameter Name	In/Out	Description
Update connection status	Third-Party App Id	In	和云服务的连接状态被更新了的第三方应用的 Id
	Connection Status	In	指示云服务与特定第三方应用的连接状态。设置如表 3-15.

2.5.2.3. Profile updated signal

表 2-13 提供 profile updated signal 的功能定义。

表 2-13 Profile updated signal

Method	Parameter		
	Parameter Name	In/Out	Description
Profile Updated	Third-Party App Id	In	与 profile 相关的第三方应用 Id。
	Profile Id	In	被更新的 Profile Id。

2.5.2.4. Profile deleted signal

表 2-14 提供 profile deleted signal 的功能定义。

表 2-14 Profile deleted signal

Method	Parameter		
	Parameter Name	In/Out	Description
Profile Updated	Third-Party App Id	In	与 profile 相关的第三方应用 Id。
	Profile Id	In	被删除的 Profile Id。

2.5.2.5. Shutdown app signal

2.5.2.6. 表 2-15 提供被发送给第三方应用来正常关闭的 shutdown app signal 的功能定义。

2.5.2.7. 表 2-15 Shutdown app signal

Method	Parameter		
	Parameter Name	In/Out	Description
Shutdown app	Third-Party App Id	In	被关闭的第三方应用的 App Id。

2.5.3. Fields

2.5.3.1. Profile status

表 2-16 Profile status

Value	Description
Inactive	Profile 没有激活
Active	Profile 激活
To be deleted	Profile 被标记马上要被删除

2.5.4. Config file policy enforcement

Gateway Management app 将 AllJoyn 总线执行策略规则写在 config file 里来控制访问可远程控制的设备接口。当 profile 被激活以后，Gateway Management app 更新 config file 允许第三方应用和可远程控制的设备之间的通信。Gateway Management app 也负责在 profile 失效后移除允许策略规则来终止设备接口的远程访问。

为了给 profile 创建允许规则，Gateway Management app 使用与第三方应用相关的用户 ID 和远程控制的 AllJoyn 应用的 unique name。Gateway Management app 应该更新 config file 策略来完成如下功能：

1. 策略应该允许 Gateway Management app 接收临近网络里多有设备应用的声明。这项将默认实现。
2. 启动时，Gateway Management app 应该基于预装的第三方应用的用户 ID 给它们全部设一个 deny-all 规则。
3. 当一个新的第三方应用安装了之后，Gateway Management app 应该基于它的用户 ID 给它加一个 deny-all 规则。
4. 在 profile 激活之后，Gateway Management app 应该更新 config file 添加允许策略规则，如表 2-17 所述。
5. 在激活的 profile 被删除或失效之后，Gateway Management app 应该更新 config file 移除所有与此 profile 相关的允许策略规则。

表 2-17 Config file allow rules

Profile 激活后添加的 config file 规则
1. 只有当服务对象为一个第三方应用维护 profile 的时候，才允许这个第三方应用访问 Gateway Managemnet app 上的 Profile Retrieval Interface。
2. 允许第三方应用通过临近网络发送它的声明（如果有）。
3. 允许第三方应用通过临近网络发送它的声明（如果有）。

4. 允许第三方应用从 profile 上可远程控制的设备/应用/接口/对象路径调用方法调用和接收方法响应。
5. 允许第三方应用从 profile 上可远程控制的设备/应用/接口/对象路径接收信号。
6. 允许第三方应用从 profile 里允许发送通知的的可远程控制的设备应用接收通知。注意这里每个通知优先级有独立的通知对象路径。
7. 允许第三方应用在临近网络里发送通知，通知类型如 profile 里一样配置。
8. 允许第三方应用向在 profile 里暴露应用接口的第三方应用接受方法调用和发送方法回应。
9. 允许给在 profile 里暴露应用接口的第三方应用发送信号。
10. 允许第三方应用内省任何可远程控制的接口。

这些 config file 策略规则的精确的语法会在应用阶段被定义。

2.5.4.1. Unique name retrieval

Gateway Management app 需要知道设备应用的唯一名来给激活的 profile 上的设备/应用在 config file 上创建允许策略规则。Gateway Management app 从 profile 激活开始就应该开始监听声明，并持续接受声明。由于不能重新取得声明，Gateway Management app 需要储存所有接收到的声明并且用相应的唯一名给 AllJoyn 应用创建允许策略。

在各种情境下，如重启，设备应用也能获得新的唯一名。在这种情况下，Gateway Management app 会受到带有新的唯一名的新声明。Gateway Management app 会使用（设备 Id，应用 Id）对来确定一个从特定设备发来的更新的声明并弄清楚唯一名是否改变过。如果是的，Gateway Management app 更新 config file 里的允许策略规则反应新唯一名。Gateway Management app 于是会触发 AllJoyn 总线去加载更新的 config file。

2.5.4.2. Config file reload

更新 config file 之后，Gateway Management app 通知 AllJoyn 总线使它可以重载更新过的 config file。AllJoyn 总线支持如下两个方法接收 config file 更新的触发。

- Platform-based signal:这个方式使用设备平台信令（如 OpenWRT kill 信令）发信号给 AllJoyn 总线来完成 config file 的重载。
- AllJoyn Method trigger：一个 AllJoyn 方法调用可以在总线调用来触发 config file 的重载。支持这个方法调用的详细情况会在应用过阶段被定义。

Gateway Management app 应该在 AllJoyn 总线上调用 AjjJoyn 方法来触发 config file 重载。

2.6. Versioning

AllJoyn 接口的版本控制应该通过一个版本属性完成呢过。如下每个 Gateway Management app 支持的 AllJoyn 接口应该被独立的控制版本。

- Profile Management interface
- App Access interface
- App Management interface

AllJoyn 网管相关接口当前的版本是‘1’。

3. 应用管理

3.1. 体系结构

图 3-1 展示了第三方应用管理功能的 Gateway Agent 体系结构。

- 一个 App Download 服务器（服务提供商提供）掌管第三方应用包。

- Control app 给初始化应用安装/升级提供 UI 和给 Gateway Management app 提供下载链接来从 App Download 服务器下载应用包。
- Gateway Management app 暴露一个 App Management 接口来使 Control app 管理 Gateway Agent 上第三方应用，包括应用的安装、升级、卸载和重启。

出于安全考虑，决定不使用 OpenWRT opkg 包管理器来安装第三方应用。Gateway Agent 上支持一个独立的 Package Management（PM）应用来安装/升级/卸载第三方应用。Package Manager app 在安装安装包之前，执行应用包的下载和签名认证。Package Manager app 的确切性质被当成应用细节，不在此 HLD 范围内。

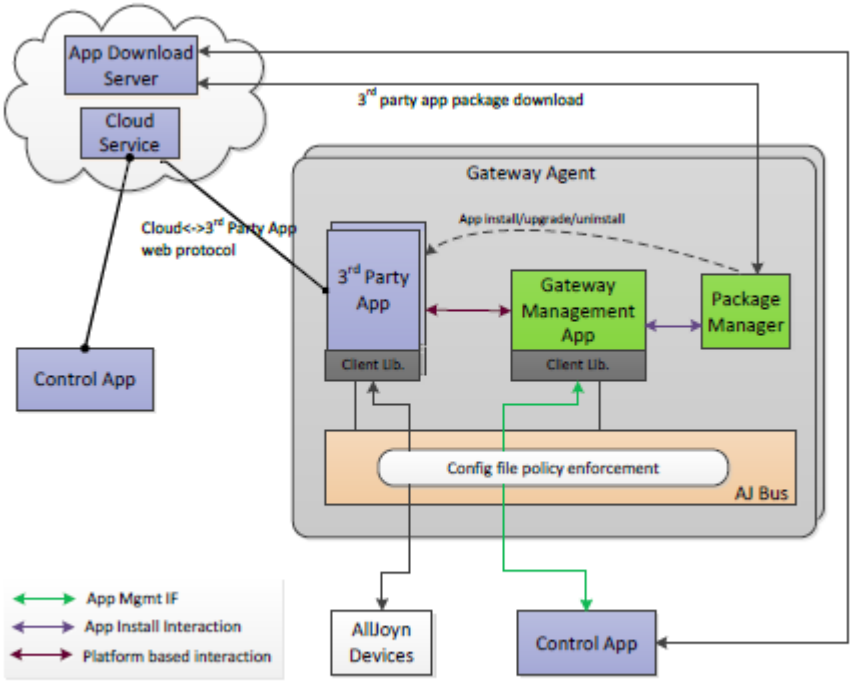


Figure 3-1 Gateway Agent app management architecture

Gateway Management app 和 Package Manager app 将被打包成 OpenWRT 安装包并且用 OpenWRT 包管理器（opkg）安装。一个 Gateway Agent 设备可能只配备 Gateway Management app 和 Package Manager app，没有出示安装任何第三方应用。在这种情况下，所有第三方应用将通过 App Management 接口安装。

3.1.1. Gateway Management app 应用管理

Gateway Management app 暴露一个安全的 App Management 接口(org.alljoyn.GWAgent.AppMgmt)来使 Control app 管理 Gateway Agent 上的第三方应用。这个接口给第三方应用提供安装，升级和卸载功能。它可以被用来在 Gateway Agent 上动态的安装新的第三方应用和升级已安装的第三方应用。Control app 给 Gateway Management app 提供应用包的下载链接。

Gateway Management app 和 Package Manager app 交互来执行应用包的下载和安装。每个第三方应用被安装在一个独立的 UNIX 用户 Id 里。Gateway Management app 分配唯一的用户 Id 给第三方应用。所有第三方应用被分配一样的 UNIX 组 Id。

App Management 接口也给第三方应用提供重启功能。Control app 也通过这个接口取回特定第三方应用的状态信息（包括安装状态、运作状态和云服务连接状态）。App Management 接口在 Gateway Management app 发出的 About 声明里被广告，且被 Control app 用来发现 Gateway Management app。

Gateway Management app应该有一个逻辑应用表并维护一个应用入口使每个第三方应用包含表 3-1 里列出的详细内容。

表 3-1Gateway Management app 维护的应用详细内容

App Detail	Description
App Id	第三方应用的 AllJoyn 标识符。
App name	第三方应用的应用名。
App Package Name	第三方应用包的包名。
App version	第三方应用的版本的最新安装版本。
App Package File URL	第三方应用包的文件位置。
App User Id	第三方应用相关的 UNIX 用户 Id。
App Group Id	第三方应用相关的 UNIX 组 Id。
Install Status	提供第三方应用的安装状态如表 3-16。
Install Description	提供应用的最后一个应用安装/升级/卸载运作完成结果的描述。
Connection Status	提供第三方应用的云服务连接状态如表 3-15。
Operation Status	提供第三方应用的运作状态如表 3-17。

对于安装失败的第三方应用，会有一个表明‘安装失败’状态的入口被维护着。这样的入口会在几个时间周期后被清空。对于预安装的第三方应用，这些应用详细信息作为离线安装进程的一部分。

所有第三方应用将有一个 manifest file 提供应用的元数据。应用的 manifest file 应该存在应用包目录里的预定义位置（例如应用包的根目录）。

3.1.2. Package Manager（包管理器）应用

Package Manager 应用提供使 Gateway Management app 能够安装/升级/卸载第三方应用的功能。Gateway Management app 传递下载链接给 Package Manager app, Package Manager app 从 App Download 服务器下载应用包，执行签名认证并把应用包安装到一个新的用户账户里。应用升级时，升级包被安装在旧包相关的用户账户里。Package Manager app 应该支持 CLI(command-line-interface)为了它提供的应用安装功能。

如上面提到的，Package Manager app（AllJoyn vs. nonAllJoyn app）详细的性能和 Gateway Management app 与 Package Manager app 通信频道的详细内容作为应用细节不在此 HLD 范围内。

3.1.3. Control app 应用管理

Control app 使用户能在 Gateway Agent 上安装，卸载和升级第三方应用。Control app 决定信的第三方应用和已安装应用的升级是否在 Gateway Agent 上可用。对此，Control app 可用 Gateway Management app 暴露的 About 数据包括（制造商、型号、平台、平台版本、软件版本、AllJoyn 软件版本和硬件版本）来给 Gateway Agent 设备过滤出可用的应用。

注意

这些字段只有部分会从 About 声明信号里接收。其它的需要从和 Gateway Management app 暴露的 About 对象的连接里取回。

Control app 也能从 Gateway Management app 取得已安装应用的 manifest 文件，从 App Download 服务器取回新应用的 manifest 文件，然后使用这些信息决定升级应用的可用性。Control app 提供安装/升级选项给用户并基于用户动作调用 App Management 接口。

Control app 也能使用户可以重启一个应用并检查应用的最新状态，包括安装状态，运作状态和云服务连接状态。

3.2. 调用流

这部分讲的是各种各样第三方应用管理用例的可能出现的场景的调用流。

3.2.1. 第三方应用安装

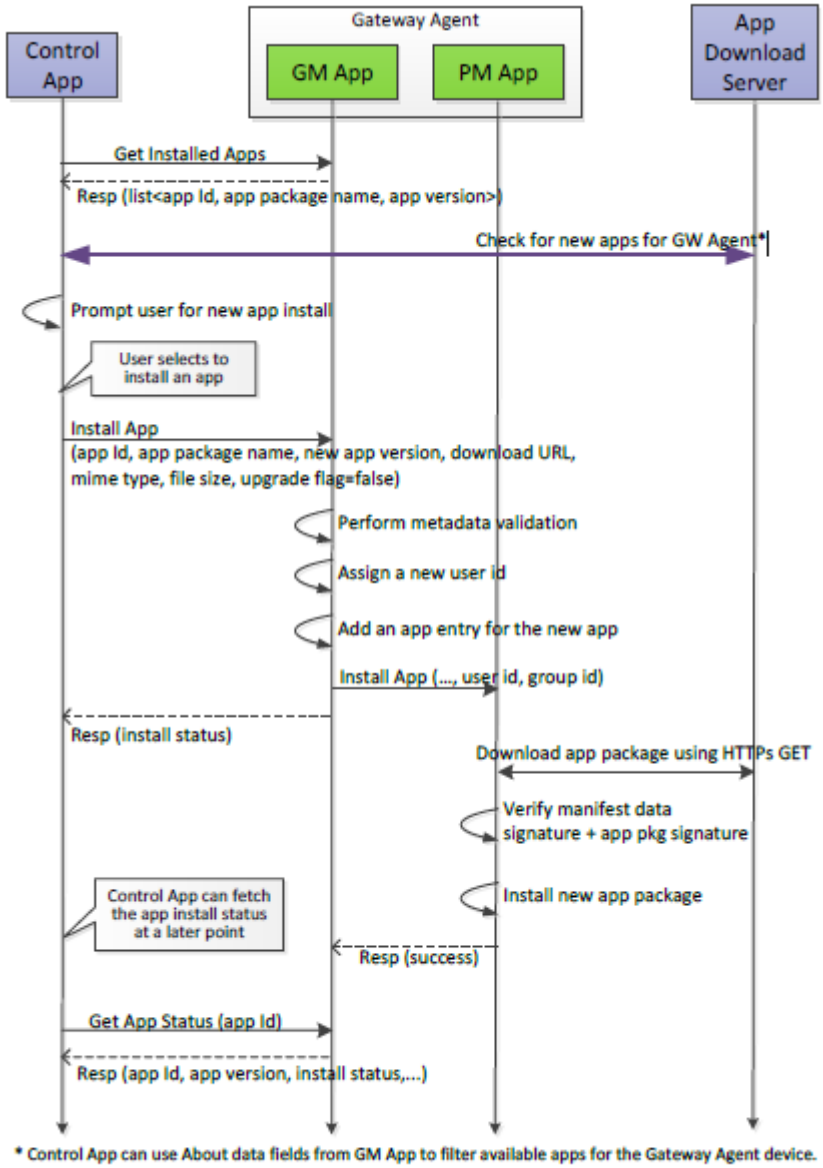


Figure 3-2 Third-party app install

3.2.2. 第三方应用升级

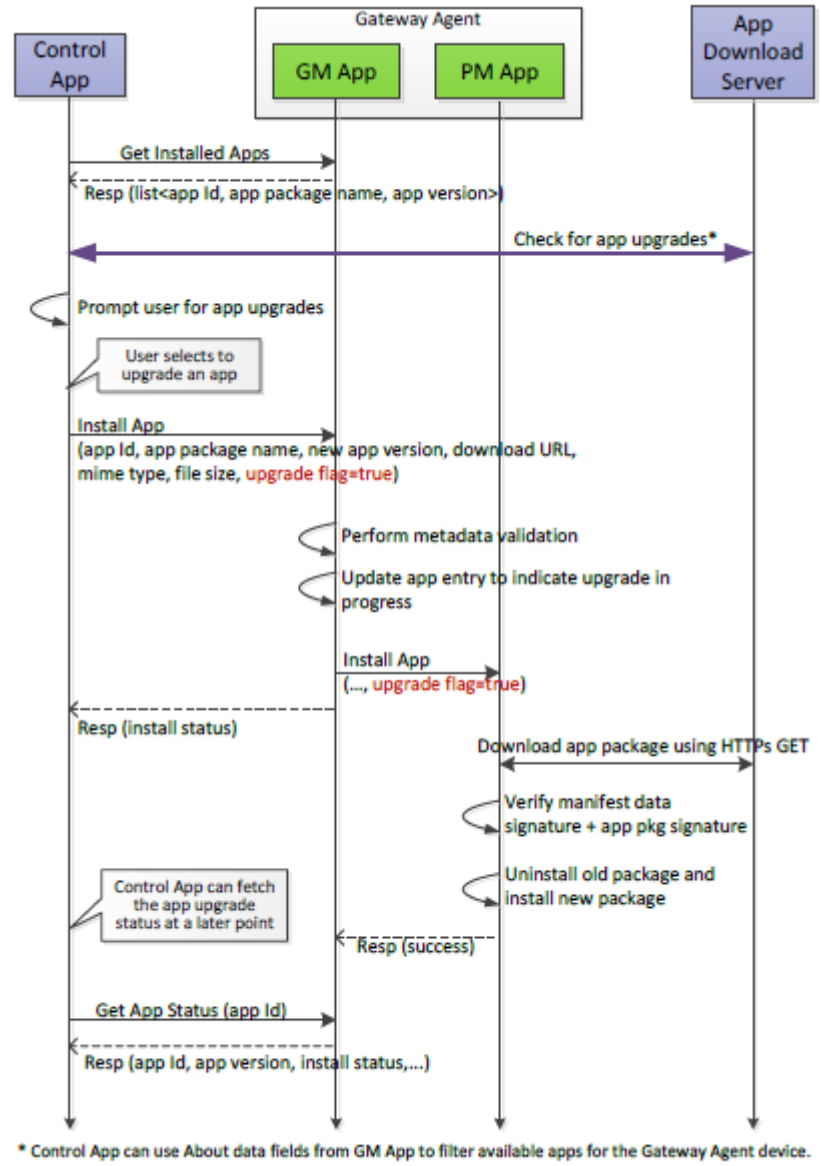


Figure 3-3 Third-party app upgrade

3.2.3. 第三方应用卸载

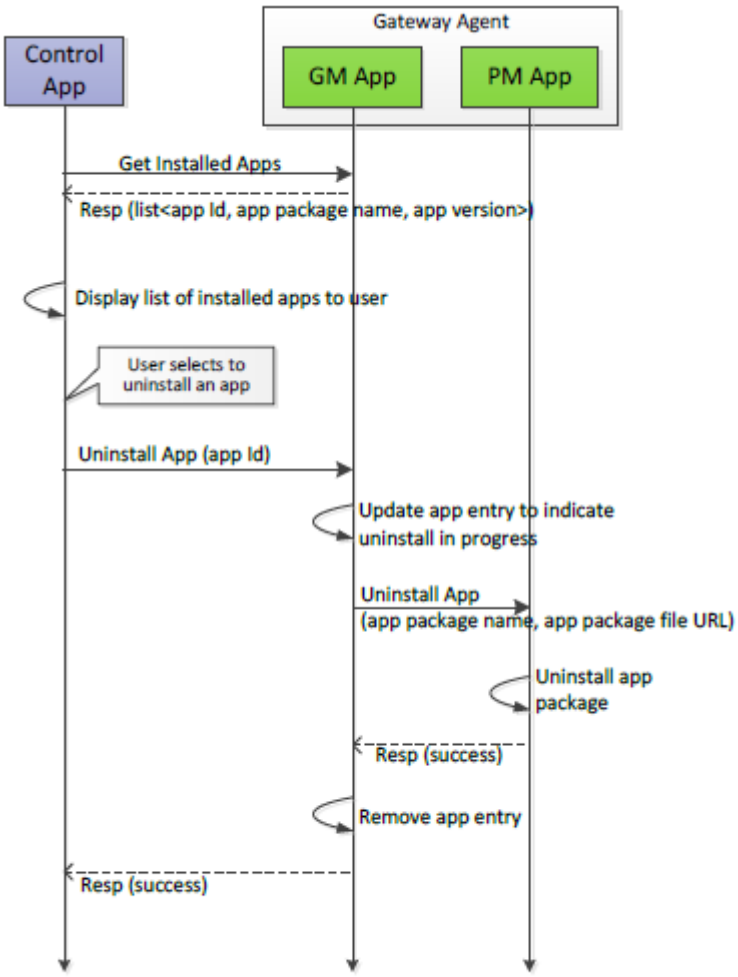


Figure 3-4 Third-party app uninstall

3.2.4. 第三方应用重启

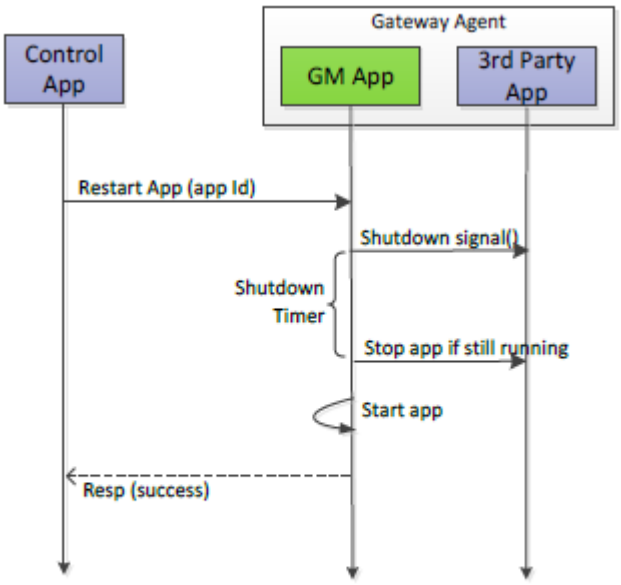


Figure 3-5 Third-party app restart

3.2.5. 添加新设备/应用

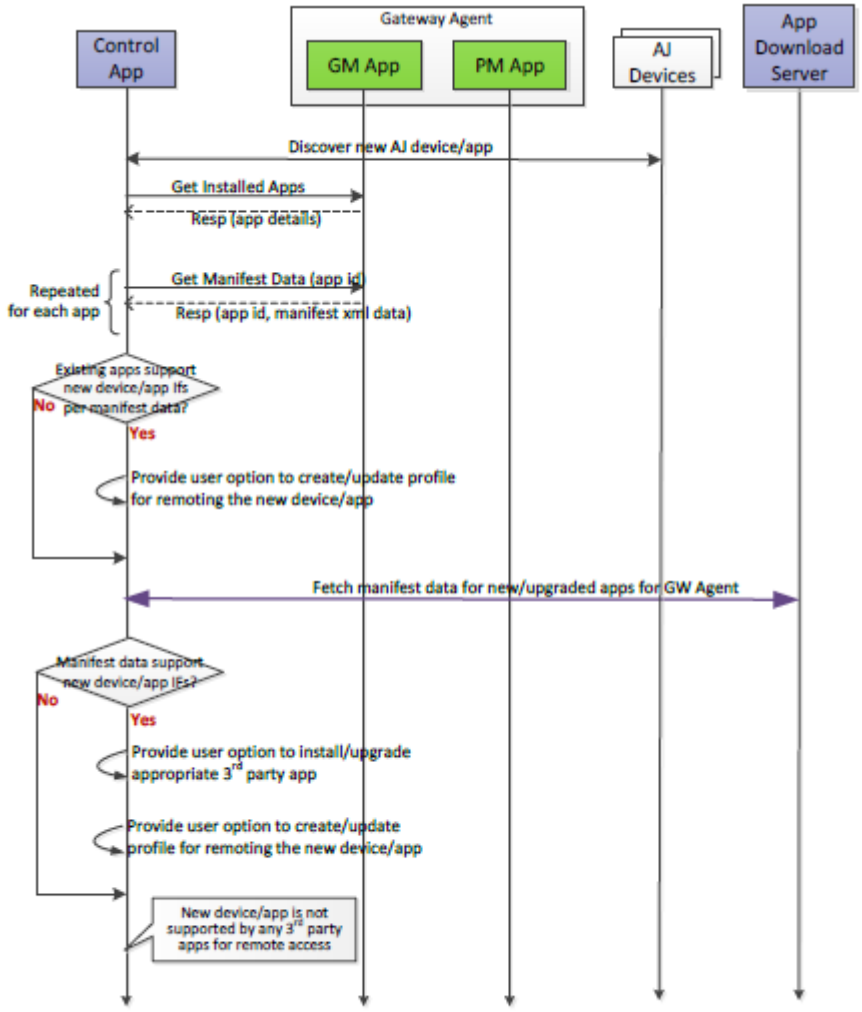


Figure 3-6 Add a new device/app

3.2.6. 移除设备

如果一个远程可访问的 IoE 设备永久的从临近网络里移除，第三方设备和云服务在访问那个设备的时候会依次失败。用户可能会也可能不会把设备从 remote profile 中移除。当用户把那个设备从 profile 里移除的时候，设备将不再被远程访问可见。

一个远程可访问的 IoE 设备可能暂时的从邻近网络里移除，过段时间再回来。第三方设备和云服务在在它移除的这段时间访问那个设备的时候会依次失败。当设备回到临近网络的时候，远程访问会自动被存储。

3.3. 功能设计

3.3.1. 应用管理接口

App Management 接口是 Gateway Management app 提供的 AllJoyn 接口，支持表 3-2 总结的功能方法。

表 3-2 App Management 接口功能

Functional method	Description
Install App	导致从提供的连接下载第三方应用包并在 Gateway Agent 上的新用户账户上安装包。包签名在安装前认证。也用于升级一个已存在的第三方应用到新版本。
Uninstall App	导致从 Gateway Agent 卸载特定的第三方应用。

Restart App	导致特定的第三方应用重启。如果与这个第三方应用相关的 profile 未激活的话，调用会失败。
Get App Status	返回特定第三方应用的状态信息包括安装状态、运作状态和云服务连接状态。
Get Installed Apps	返回 Gateway Agent 上已安装第三方应用的列表。
Get Manifest Data	返回特定第三方应用的 manifest 文件。
Get Manifest Interfaces	翻译特定第三方应用的 manifest 文件的 AllJoyn 接口的详细信息。
Functional Signal	Description
App Status Changed	当第三方应用状态信息改变的时候发送给 Control app 的信号。这是个基于会话的信号。

3.3.1.1. Install App

表 3-3 提供 Install App method 的功能定义

表 3-3 Install App method

Method	Parameter		
	Parameter	In/ Out	Description
Install App	App Id	In	安装的第三方应用的 App Id。
	App Name	In	安装的第三方应用的 App name。
	App package name	In	安装的第三方应用的 Package name。
	App version	In	安装的第三方应用的版本。
	Download URL	In	下载应用包的网页连接。
	App package file size	In	包括 manifest 数据的整个应用包的文件大小。
	Upgrade Flag	In	标记设为 false 表示新应用安装，设为 true 表示应用升级。
	Install Status	Out	指示第三方应用的安装状态。将被设为 'Install in progress' 或 'upgrade in progress'。

3.3.1.2. Uninstall App

表 3-4 提供了 Uninstall App method 的功能定义。

表 3-4 Uninstall App method

Method	Parameter		
	Parameter	In/Out	Description
Uninstall App	App Id	In	需要被卸载的第三方应用的 App Id。
	Resp Status	Out	指示卸载第三方应用的成功/失败状态。
	Install Status	Out(condition)	指示第三方应用的卸载状态。只有当 Resp Status 为 failure 时

			才包括。这将被设为'installed'。
	Install Description	Out(condition)	提供卸载失败的相关描述。只有当 Resp Status 为 failure 时才包括。

3.3.1.3. Restart App

表 3-5 提供了 Restart App method 的功能定义。

表 3-5 Restart App method

Method	Parameter		
	Parameter	In/Out	Description
Restart App	App Id	In	需要被卸载的第三方应用的 App Id。
	Resp Status	Out	指示重启第三方应用的成功/失败状态。

3.3.1.4. Get App Status

表 3-6 提供了 Get App Status method 的功能定义。

表 3-6 Get App Status method

Method	Parameter		
	Parameter	In/Out	Description
Get App Status	App Id	In	需要被卸载的第三方应用的 App Id。
	Install Status	Out	指示第三方应用的安装状态。
	Install Description	Out (optional)	提供描述指示最后一个第三方应用的最后一个完成的应用安装/升级/卸载运作。
	Connection Status	Out	指示第三方应用和云服务的连接状态。
	Operational Status	Out	指示第三方应用的运作状态。

3.3.1.5. Get Installed App

表 3-7 提供了 Get Installed App method 的功能定义。

表 3-7 Get Installed App method

Method	Parameter		
	Parameter	In/ Out	Description
Get Installed App	List<Installed App Info>		
	App Id	Out	安装的第三方应用的 App Id。
	App name	Out	安装的第三方应用的 App name。
	App version	Out	安装的第三方应用的版本。
	App object path	Out	访问应用详细信息的对象路径。

3.3.1.6. Get Manifest Data

表 3-8 提供了 Get Manifest Data method 的功能定义。

表 3-8 Get Manifest Data method

Method	Parameter		
	Parameter	In/Out	Description
Get Manifest Data	App Id	In	安装的第三方应用的 App Id。
	Manifest Data	Out	提供 manifest 数据 blob。

3.3.1.7. Get Manifest Interfaces

表 3-9 提供了 Get Manifest Interfaces method 的功能定义。

表 3-9 Get Manifest Interfaces method

Method	Parameter		
	Parameter	In/Out	Description
Get Manifest Interfaces	App Id	In	安装的第三方应用的 App Id。
	List<Supported interfaces>		
	Interface name	Out	第三方应用支持的 AllJoyn 接口的接口名。
	Interface friendly name	Out	AllJoyn 接口的用户可读名。
	List<Desired Interfaces>		
	Interface name	Out	第三方应用可能访问的 AllJoyn 接口的接口名。
	Interface friendly name	Out	AllJoyn 接口的用户可读名。

3.3.1.8. App Status Changed signal

表 3-10 提供了 App Status Changed signal 的功能定义。

表 3-10 App Status Changed signal

Signal	Parameter		
	Parameter	In/Out	Description
App Status Changed signal	Third-Party App Id	In	与 profile 相关的第三方应用 Id。
	Install Status	In	指示第三方应用的安装状态。
	Install Description	In (optional)	提供描述指示最后一个第三方应用的最后一个完成的应用安装/升级/卸载运作。
	Connection Status	In	指示第三方应用和云服务的连接状态。
	Operational Status	In	指示第三方应用的运作状态。

3.3.2. Package Manager 应用功能

Package Manager 应用提供安装和卸载功能如下

表 3-11 Package Manager 应用功能

Logical function	Description
InstallApp	导致从被提供的链接下载第三方应用包并在 Gateway Agent 的新账户上安装包。安装前验证包签名。也用于升级现存的应用到新版本。
UninstallApp	导致从 Gateway Agent 特定第三方应用的卸载。

3.3.2.1. InstallApp

表 3-12 提供了 Package Management app 提供的 InstallApp 逻辑功能的输入输出变量表。

表 3-12 InstallApp function

Logical function	Parameter		
	Parameter	In/ Out	Description
InstallApp	App Id	In	安装的第三方应用的 App Id。
	App package name	In	安装的第三方应用的包名。
	App version	In	安装的第三方应用的版本。
	Download URL	In	下载应用包的网页链接。
	App package file size	In	包括 manifest 数据的整个应用包的文件大小。
	Upgrade Flag	In	标记设为 false 表示新应用安装，设为 true 表示应用升级。
	App user Id	In	安装/升级应用时用到的 UNIX 用户 Id。安装新应用时设为新唯一用户 Id。升级时设为当前应用用户 Id。
	Resp Status	Out	指示第三方应用安装/升级的成功/失败状态。

3.3.2.2. UninstallApp

表 3-13 提供了 Package Management app 提供的 UninstallApp 逻辑功能的输入输出变量表。

表 3-13 UninstallApp function

Logical function	Parameter		
	Parameter	In/ Out	Description
InstallApp	App package name	In	被卸载的第三方应用的包名。
	App package file URL	In	被卸载的第三方应用包的文件位置。
	Resp Status	Out	指示第三方应用安装/升级的成功/失败状态。

3.3.3. AllJoyn 包格式

第三方应用包应该被做成符合 AllJoyn 包格式。应用包文件在一个二进制文件里应

该包括如下内容：

- 给应用提供元数据的 manifest 文件
- 应用包文件

每个应用包文件将要访问预定义的一系列标准 OpenWRT 函数库。如果一个应用要访问更多的函数库，这些附加的函数库需要与应用包绑定。

3.3.3.1. Manifest（清单）文件

每个第三方应用将有一个提供与其相关的元数据的 manifest file。Manifest file 将放在应用包目录下预定义位置（如，应用包更目录）。Manifest file 应该包括如表 3-14 所描述的信息。

表 3-14 Manifest file content

Fields	Description
App Id	第三方应用的 AllJoyn 标识符。
App package name	第三方应用包的包名。
App version	第三方应用版本。
Min AJ SDK version	第三方应用需要的最低 AllJoyn SDK 版本。
List<Desire AllJoyn Interfaces for access>	第三方应用会访问到的 AllJoyn 接口列表。
List<AllJoyn interfaces app support>	可以被设置使得临近网络内 AllJoyn 设备能访问的第三方应用支持的 AllJoyn 接口列表。

3.3.3.2. 签名认证

应用包文件将用一个应用证书签名发给第三方应用开发者。应用签名证书应该和一个在 Package Management app 维持的根证书绑定。

注意

自签名证书不使用。

作为应用包签名过程的一部分，manifest file 内容和应用文件都被签名。签名和应用证书（X.509 格式）被附到应用包二进制文件里。

Package Management app 将认证接收到的应用证书绑定到它维护的相同的根证书上。Package Management app 接着用从应用证书里的接收到的公钥认证接收到的签名。

3.3.4. 字段

这部分提供了第三方应用管理功能相关的一些字段的定义。

3.3.4.1. 连接状态

表 3-15 Connection status

Value	Description
Not initialized	由于 profile 未激活，与云服务连接未初始化。
In-progress	连接在被建立的过程中。
Connected	和云服务的连接被建立和运作了。
Not connected	和云服务的连接断了。
Error	第三方应用有错误情况。

3.3.4.2. 安装状态

表 3-16 Install status

Value	Description
Installed	第三方应用安装在 Gateway Agent 了。
Install in progress	第三方应用当前被安装在 Gateway Agent。
Upgrade in progress	第三方应用当前被升级过在 Gateway Agent。
Uninstall in progress	第三方应用当前被卸载在 Gateway Agent 了。
Install failed	第三方应用的应用安装失败。有这个状态的应用入口会维持一段时间让 Control app 能取得这个状态。过了这段时间，应用入口会被移除。

3.3.4.3. 安装描述

这个字段提供应用的最后一个应用安装/升级/卸载运作结果的相关文本描述。这可以指示‘应用升级失败’或‘应用卸载失败’。在这些情况下，Install Status 字段将被设为’Installed’。

3.3.4.4. 运作状态

表 3-17 Operational status

Value	Description
Running	第三方应用正在运行。
Stopped	第三方应用停止了。

4. 性能

[GWA-HLD-0001]Gateway Management 应用将支持 Gateway Agent 1.0 PRD 里定义的准出条件的性能和可扩展性需求。

5. 要求

这部分描述 Gateway Agent 的系统需求。

5.1. 总体要求

表 5-1 Gateway management application requirements

Requirement	Description
GWA-HLD-0002	Gateway Management Application 将支持 org.alljoyn.GWAgent.ProfileMgmt 接口。
GWA-HLD-0003	Gateway Management Application 将支持 org.alljoyn.GWAgent.AppAccess 接口。
GWA-HLD-0004	Gateway Management Application 将支持 org.alljoyn.GWAgent.AppMgmt 接口。
GWA-HLD-0005	org.alljoyn.GWAgent.ProfileMgmt 接口将是一个安全接口。
GWA-HLD-0006	org.alljoyn.GWAgent.AppMgmt 接口将是一个安全接口。
GWA-HLD-0007	Gateway Management Application 将持久的储存 profile 数据。
GWA-HLD-0008	对一个特定的第三方应用，Gateway Management Application 将只支持一个 profile。
GWA-HLD-0009	Gateway Management Application 将为多个第三方应用支持 profile。

GWA-HLD-0010	Gateway Management Application 将支持更新 config file 策略规则来控制第三方应用和 AllJoyn 设备之间的通信。
GWA-HLD-0011	AllJoyn 总线在第三方应用间传递消息的时候将执行 config file 策略规则。
GWA-HLD-0012	Gateway Management Application 将创建防火墙规则来阻拦第三方应用和临近 wi-fi 网络里其他 AllJoyn 或非 AllJoyn 设备间的开放 IP 连接。
GWA-HLD-0081	Gateway Management Application 将为安全 AllJoyn 接口支持 ALLJOYN_SRP_KEYX 认证机制。
GWA-HLD-0082	Gateway Management Application 将被配置一个默认密码用于安全 AllJoyn 接口。
GWA-HLD-0083	Gateway Management Application 将支持 Config 服务来为安全 AllJoyn 接口更新密码。

5.2. 网关发现

表 5-2 Gateway discover requirements

Requirement	Description
GWA-HLD-0014	Gateway Management Application 将在 About 声明信号里声明 Profile Management 接口。
GWA-HLD-0015	Gateway Management Application 将在 About 声明信号里声明 Application Management 接口。
GWA-HLD-0016	Gateway Management Application 将把'App Name'参数的配置值填写到声明信号的 AppName 字段。

5.3. Profile 管理

表 5-3 Profile management requirements

Requirement	Description
GWA-HLD-0017	Gateway Management Application 将为 Profile Management 接口支持依据表 2-1 的功能方法。
GWA-HLD-0018	Gateway Management Application 将支持依据图 2-6 的行为来创建 profile。
GWA-HLD-0019	Gateway Management Application 将支持依据图 2-7 的行为来激活 profile。
GWA-HLD-0020	Gateway Management Application 将支持依据图 2-9 的行为来更新 profile。
GWA-HLD-0021	Gateway Management Application 将支持依据图 2-10 的行为来删除 profile。
GWA-HLD-0023	Gateway Management Application 将支持依据图 2-11 的行为来使 profile 失效。
GWA-HLD-0025	Gateway Management Application 将依据表 2-2 支持 Create Profile 方法的功能定义。
GWA-HLD-0026	Gateway Management Application 将依据表 2-3 支持 Activate Profile 方法的功能定义。
GWA-HLD-0027	Gateway Management Application 将依据表 2-4 支持 Get Profile 方法的功能定义。

GWA-HLD-0084	Gateway Management Application 将依据表 2-5 支持 Get Profile Status 方法的功能定义。
GWA-HLD-0085	Gateway Management Application 将依据表 2-6 支持 Get Profile list 方法的功能定义。
GWA-HLD-0028	Gateway Management Application 将依据表 2-7 支持 Update Profile 方法的功能定义。
GWA-HLD-0029	Gateway Management Application 将依据表 2-8 支持 Delete Profile 方法的功能定义。
GWA-HLD-0031	Gateway Management Application 将依据表 2-9 支持 Deactivate Profile 方法的功能定义。

5.4. 第三方应用应用访问

表 5-4 App access for third-party apps requirement

Requirement	Description
GWA-HLD-0033	Gateway Management Application 将在 AllJoyn 总线上注册”org.alljoyn.GWAgent.GWApp”的 well-known name。
GWA-HLD-0034	每个第三方应用将在 AllJoyn 总线上注册”org.alljoyn.GWAgent.ThirdPartyApp.<third-party app id>”的 well-known name。
GWA-HLD-0035	Gateway Management Application 将为每个第三方应用把 profile 数据作为有着”/Profile/<third-party app id>”格式 well-known 对象路径的独立服务对象的一部分维护。
GWA-HLD-0036	Gateway Management Application 将依据表 2-10 支持 App Access 接口的功能方法。
GWA-HLD-0037	Gateway Management Application 将依据表 2-11 支持 App Access 接口里 Get Profile 方法的功能定义。
GWA-HLD-0038	只在相关的 profile 激活时，Gateway Management Application 将在 Get Profile 回应提供 profile 信息给第三方应用。
GWA-HLD-0039	Gateway Management Application 将依据表 2-12 支持 Update Connection Status 方法的功能定义。
GWA-HLD-0040	Gateway Management Application 将依据表 2-13 支持 Profile Updated signal 的功能定义。
GWA-HLD-0041	每当与一个第三方应用相关的激活的 profile 更新了，Gateway Management Application 将给那个应用发送 Profile Updated signal。
GWA-HLD-0042	Gateway Management Application 将依据表 2-14 支持 Profile Deleted signal 的功能定义。
GWA-HLD-0043	每当与一个第三方应用相关的激活的 profile 被删除了，Gateway Management Application 将给那个应用发送 Profile Deleted signal。
GWA-HLD-0044	Gateway Management Application 将依据表 2-15 支持 Shutdown App signal 的功能定义。
GWA-HLD-0045	Gateway Management Application 在一个第三方应用需要被停止的时候将给那个应用发送 Shutdown App signal。

5.5. Config file 策略

表 5-5 Config file policy requirements

Requirement	Description
GWA-HLD-0046	在 config file 里为一个特定的第三方应用创建允许策略规则的时候，Gateway Management Application 将使用与那个应用相关的 UNIX 用户 Id。
GWA-HLD-0047	在 config file 里创建允许策略规则的时候，Gateway Management Application 将使用可远程操控的 AllJoyn 应用的最新的唯一名。
GWA-HLD-0048	在启动时，Gateway Management Application 将更新 config file 策略规则来允许所有在临近网络里的设备为自己的声明。
GWA-HLD-0049	在启动时，Gateway Management Application 将更新 config file 策略规则来给所有预装的第三方应用加一条拒绝所有的策略规则。
GWA-HLD-0050	在一个新的第三方应用安装之后，Gateway Management Application 将更新 config file 策略规则来给那个第三方应用加一条拒绝所有的策略规则。
GWA-HLD-0051	在一个 profile 激活之后，Gateway Management Application 将更新 config file 策略规则来依据表 2-17 添加允许策略规则。
GWA-HLD-0052	在一个 profile 被删除或失效之后，Gateway Management Application 将更新 config file 策略规则来移除所有与那个 profile 相关的允许策略规则。
GWA-HLD-0053	一旦 profile 激活了，Gateway Management Application 就将开始从那 AllJoyn 应用监听声明并应该在之后一直从那个 AllJoyn 应用接受声明。
GWA-HLD-0054	Gateway Management Application 将储存所有从 AllJoyn 应用接受的声明。
GWA-HLD-0055	当 Gateway Management Application 确定了一个可远程操作的 AllJoyn 应用相关的唯一名改变了，它将更新 config file 策略规则来反映新的唯一名。
GWA-HLD-0056	在更新 Config file 之后，Gateway Management Application 将通过一个 AllJoyn 方法触发 config file 的重载的方式通知 AllJoyn 总线。

5.6. 应用管理

Requirement	Description
GWA-HLD-0057	Gateway Management Application 将依据表 3-1 为每个第三方应用维护详细信息。
GWA-HLD-0058	Gateway Management Application 将未安装失败的第三方应用维护至少一段清空时间的应用详细信息。
GWA-HLD-0059	作为预装应用的离线安装过程的一部分，Gateway Management Application 将用应用详细信息进行配置。
GWA-HLD-0060	Package Manager Application 将为第三方应用提供应用安装，升级和卸载功能。
GWA-HLD-0061	Package Manager Application 将为应用安装，升级和卸载提供命令行接口。
GWA-HLD-0062	Gateway Management Application 将依据表 3-2 支持 App Management 接口的功能方法。

GWA-HLD-0063	Gateway Management Application 将依据图 3-2 支持第三方应用安装行为。
GWA-HLD-0064	Gateway Management Application 将依据图 3-3 支持第三方应用升级行为。
GWA-HLD-0065	Gateway Management Application 将依据图 3-4 支持第三方应用卸载行为。
GWA-HLD-0066	Gateway Management Application 将依据图 3-5 支持第三方应用重启行为。
GWA-HLD-0067	Gateway Management Application 将依据表 3-3 支持 Install App 方法的功能定义。
GWA-HLD-0068	Gateway Management Application 将依据表 3-4 支持 Uninstall App 方法的功能定义。
GWA-HLD-0069	Gateway Management Application 将依据表 3-5 支持 Restart App 方法的功能定义。
GWA-HLD-0070	Gateway Management Application 将依据表 3-6 支持 Get App Status 方法的功能定义。
GWA-HLD-0071	Gateway Management Application 将依据表 3-7 支持 Get Installed Apps 方法的功能定义。
GWA-HLD-0072	Gateway Management Application 将依据表 3-9 支持 Get Manifest Data 方法的功能定义。
GWA-HLD-0073	Package Management Application 将依据表 3-11 提供 InstallApp 和 Unistallapp 功能。
GWA-HLD-0074	Package Management Application 将依据表 3-12 支持 InstallApp 功能的参数。
GWA-HLD-0075	Package Management Application 将依据表 3-13 支持 UninstallApp 功能的参数。
GWA-HLD-0076	Package Management Application 将支持通过 HTTPs 从提供的下载链接下载应用包。
GWA-HLD-0077	Package Management Application 将支持 manifest file 和应用包文件包含在 AllJoyn 包格式里。
GWA-HLD-0078	第三方应用的 manifest file 依据表 3-14 将包含元数据。
GWA-HLD-0079	Package Management Application 将为第三方应用签名认证维护一个根证书并且将接收到的应用证书与同一个跟证书绑定。
GWA-HLD-0080	作为包二进制文件，Package Management Application 将认证接收到的应用包签名。

6. 配置参数

Gateway Management app 将支持如表 6-1 的配置参数

表 6-1 Gateway Management app app config parameters

Parameter	Default Value	Description
App name	“GM App”	可以被设置来指示明确的 Gateway Management app 服务提供商名，在声明信号里被广告。
Response	TBD	定义等待第三方应用回应的定时器的值。 Gateway

timer		Management app 需要从第三方应用的回应的所有情况使用一个简单的可配置的定时器。
Shutdown timer	200ms	给 Gateway Management app 设定定时器值在发送关闭信号后等待第三方应用。如果在时间超出的情况下，第三方应用依然运行着，则强制关闭之。
Purge timer	TBD	定义定时器值清空 Gateway Management app 上安装失败的应用接口。

附录 A　Gateway Agent Architecture with Security 2.0

这部分描述可能的 Security 2.0 下的 Gateway Agent 体系结构。

A．1 Accessing secure interfaces with Security 2.0

Security 2.0 访问 AllJoyn 网络上的应用和设备需要设置许可。图 6-1 展示了设定这些许可和用这些许可访问设备/应用与 Gateway Agent 相关的流程。详细内容请参见 *AllJoyn™ Security 2.0 HLD*。

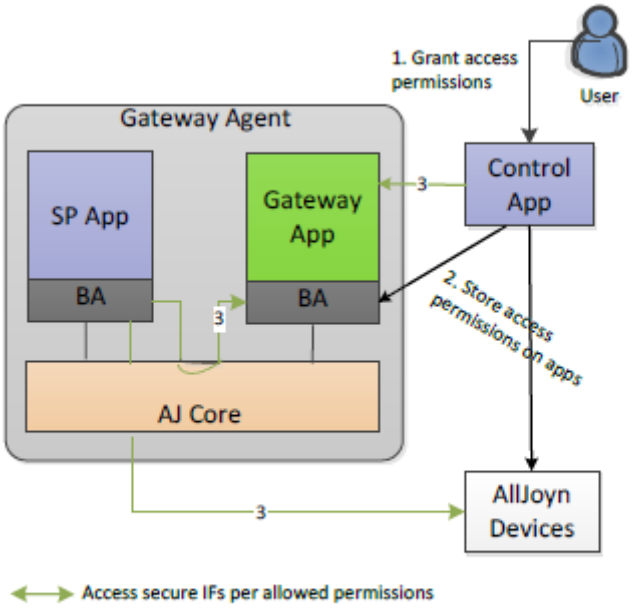


Figure 6-1 Gateway Agent Security 2.0