



(12)发明专利申请

(10)申请公布号 CN 106797392 A

(43)申请公布日 2017.05.31

(21)申请号 201580045182.7

(22)申请日 2015.07.22

(30)优先权数据

62/027,398 2014.07.22 US

(85)PCT国际申请进入国家阶段日

2017.02.22

(86)PCT国际申请的申请数据

PCT/US2015/041493 2015.07.22

(87)PCT国际申请的公布数据

W02016/014642 EN 2016.01.28

(71)申请人 康维达无线有限责任公司

地址 美国特拉华州

(72)发明人 李旭 李庆光 董丽君 路广

沙米姆·阿克巴尔·拉赫曼 陈卓

王重钢

(74)专利代理机构 中原信达知识产权代理有限
责任公司 11219

代理人 穆森 戚传江

(51)Int.Cl.

H04L 29/08(2006.01)

H04W 4/00(2009.01)

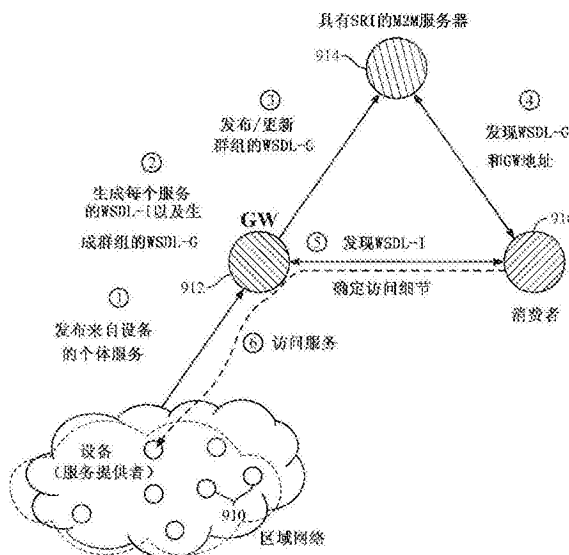
权利要求书5页 说明书33页 附图29页

(54)发明名称

M2M-IoT服务的发布和发现

(57)摘要

公开了用于M2M/IoT服务的发布和发现的系统。M2M网关系统从M2M/IoT设备接收资源描述。网关系统对于每个接收到的资源描述创建记录关于该资源的信息的个体web服务描述语言文件(WSDL-I文件)。网关识别具有相似特性的服务群组,并对于每个所识别的群组生成记录关于该群组的信息的web服务描述文件(WSDL-G文件)。WSDL-G文件被传输至服务注册基础架构(SRI)。消费者系统对于满足特定准则的服务查询SRI,并接收满足该准则的服务群组的WSDL-G文件。消费者系统从网关请求与所选WSDL-G文件相对应的WSDL-I文件。消费者系统使用所接收的WSDL-I文件来选择特定服务,并且使用WSDL-I文件中所选择服务的信息来从相对应的M2M/IoT设备请求服务。



1. 一种计算机实现的方法,包括:
第一计算系统接收多个资源描述,每个资源描述描述由设备提供的资源;
所述第一计算系统对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件;
所述第一计算系统生成与所描述的服务的群组相对应的web服务描述语言文件;以及
所述第一计算系统传送与所描述的服务的群组相对应的所述web服务描述语言文件。
2. 根据权利要求1所述的计算机实现的方法,
其中,第一计算系统接收多个资源描述包括第一计算系统从多个IoT设备接收多个资源描述。
3. 根据权利要求1所述的计算机实现的方法,
其中,第一计算系统对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件包括:第一计算系统生成描述如何访问特定服务的web服务描述语言文件。
4. 根据权利要求3所述的计算机实现的方法,
其中,所述第一计算系统生成描述如何访问特定服务的web服务描述语言文件进一步包括所述第一计算系统生成描述以下中的一个或多个的web服务描述语言文件:服务何时可用、服务在何处可用、以及与服务相关联的成本。
5. 根据权利要求4所述的计算机实现的方法,
其中,第一计算系统生成web服务描述语言文件包括第一计算系统生成WSDL-I文件。
6. 根据权利要求1所述的计算机实现的方法,
其中,第一计算系统生成与所描述的服务的群组相对应的web服务描述语言文件包括:第一计算系统生成包括描述多个相似服务的信息的web服务描述语言文件。
7. 根据权利要求6所述的计算机实现的方法,
其中,第一计算系统生成与群组相对应的web服务描述语言文件包括生成WSDL-G文件。
8. 根据权利要求1所述的计算机实现的方法,
其中,所述第一计算系统是网关计算系统。
9. 根据权利要求1所述的计算机实现的方法,进一步包括:在服务器系统处接收所传输的与所描述的服务的群组相对应的web服务描述语言文件。
10. 根据权利要求9所述的计算机实现的方法,进一步包括:所述服务器系统从服务消费者系统接收发现服务的请求。
11. 根据权利要求10所述的计算机实现的方法,进一步包括:
响应于所述请求,所述服务器系统将关于与所描述的服务的群组相对应的所述web服务描述语言文件的信息传输至所述服务消费者系统。
12. 根据权利要求11所述的计算机实现的方法,进一步包括:
所述服务消费者系统参照与所描述的服务的群组相对应的所述web服务描述语言文件中的信息,并且将所述第一计算系统识别为包含关于在与所描述的服务的群组相对应的所述web服务描述语言文件中识别的服务的信息;以及
所述服务消费者系统生成并向所述第一计算系统传输对于关于在与所描述的服务的群组相对应的所述web服务描述语言文件中识别的服务的信息的请求。

13. 根据权利要求12所述的计算机实现的方法,进一步包括:所述第一计算系统对于多个服务中的每一个来向所述服务消费者系统传送描述服务的web服务描述语言文件。

14. 根据权利要求13所述的计算机实现的方法,进一步包括:

所述第一计算系统对于多个服务中的每一个来接收描述服务的web服务描述语言文件;

所述第一计算系统解析所接收的web服务描述语言文件以识别服务和关于访问所识别的服务的过程的参数;以及

所述第一计算系统采用所识别的参数来生成和传送执行所识别的服务的请求。

15. 一种系统,包括:

网关计算系统,包括:

计算处理器;以及

计算存储器,所述计算存储器与所述计算处理器通信地耦合,所述计算存储器中存储有用于执行操作的可执行计算指令,所述操作包括:

接收多个资源描述,每个资源描述描述由设备提供的资源;

对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件;

生成与所描述的服务的群组相对应的web服务描述语言文件;以及

传送与所描述的服务的群组相对应的所述web服务描述语言文件。

16. 根据权利要求15所述的系统,

其中,接收多个资源描述包括从多个IoT设备接收多个资源描述。

17. 根据权利要求15所述的系统,

其中,对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件包括:生成描述如何访问特定服务的web服务描述语言文件。

18. 根据权利要求17所述的系统,

其中,生成描述如何访问特定服务的web服务描述语言文件进一步包括生成描述以下中的一个或多个的web服务描述语言文件:服务何时可用、服务在何处可用、以及与服务相关联的成本。

19. 根据权利要求18所述的系统,

其中,生成web服务描述语言文件包括生成WSDL-I文件。

20. 根据权利要求15所述的系统,

其中,生成与所描述的服务的群组相对应的web服务描述语言文件包括:生成包括描述多个相似服务的信息的web服务描述语言文件。

21. 根据权利要求20所述的系统,

其中,生成与群组相对应的web服务描述语言文件包括生成WSDL-G文件。

22. 根据权利要求15所述的系统,进一步包括通信地耦合至所述网关计算系统的服务器系统,所述服务器系统中存储有可执行计算指令以用于执行操作,所述操作包括:接收所传输的与所描述的服务的群组相对应的web服务描述语言文件。

23. 根据权利要求22所述的系统,进一步包括服务消费者系统,所述服务消费者系统与所述服务器系统和所述网关计算系统通信地耦合,其中,所述服务器系统中存储有可执行

计算指令以用于执行进一步的操作,所述进一步的操作包括:从所述服务消费者系统接收发现服务的请求。

24. 根据权利要求23所述的系统,

其中,所述服务器系统中存储有可执行计算指令以用于执行进一步的操作,所述进一步的操作包括:响应于所述请求,将关于与所描述的服务的群组相对应的所述web服务描述语言文件的信息传输至所述服务消费者系统。

25. 根据权利要求24所述的系统

其中,所述服务消费者系统中存储有可执行计算指令以用于执行进一步的操作,所述进一步的操作包括:

参照与所描述的服务的群组相对应的所述web服务描述语言文件中的信息,并且将所述网关系统识别为包含关于在与所描述的服务的群组相对应的所述web服务描述语言文件中识别的服务的信息,以及

生成并向所述网关系统传输对于关于在与所描述的服务的群组相对应的所述web服务描述语言文件中识别的服务的信息的请求。

26. 根据权利要求25所述的系统,

其中,所述网关计算系统中存储有可执行计算指令以用于执行进一步的操作,所述进一步的操作包括:对于多个服务中的每一个来向所述服务消费者系统传送描述服务的web服务描述语言文件。

27. 根据权利要求26所述的系统,

其中,所述网关计算系统中存储有可执行指令以用于执行进一步的操作,所述进一步的操作包括:

对于多个服务中的每一个来接收描述服务的web服务描述语言文件;

解析所接收的web服务描述语言文件以识别服务和关于访问所识别的服务的过程的参数;以及

采用所识别的参数来生成和传送执行所识别的服务的请求。

28. 一种计算机实现的方法,包括:

第一计算系统接收多个资源描述,每个资源描述描述由设备提供的资源;以及

所述第一计算系统对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件;

其中,生成描述服务的web服务描述语言文件包括:

并入指示所述服务何时可用的信息;

并入指示所述服务在何处可用的信息;以及

并入指示与使用所述服务相关联的成本的信息。

29. 一种系统,包括:

计算处理器;以及

计算存储器,所述计算存储器与所述计算处理器通信地耦合,所述计算存储器中存储有用于执行操作的可执行计算指令,所述操作包括:

接收多个资源描述,每个资源描述描述由设备提供的资源;以及对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件;

其中,生成描述服务的web服务描述语言文件包括:

并入指示所述服务何时可用的信息;

并入指示所述服务在何处可用的信息;以及

并入指示与使用所述服务相关联的成本的信息。

30.一种计算机实现的方法,包括:

第一计算系统接收多个资源描述,每个资源描述描述由设备提供的资源;

所述第一计算系统对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件;以及

所述第一计算系统生成与所描述的服务的群组相对应的web服务描述语言文件,

其中,所述第一计算系统生成与所描述的服务的群组相对应的web服务描述语言文件包括:编译概述所描述的服务的所述群组中的服务的信息。

31.根据权利要求30所述的计算机实现的方法

其中,编译概述所描述的服务的所述群组中的服务的信息包括至少编译以下内容中的多个:群组标识符;群组功能;群组成员的数目;群组统计;所述群组中的服务的区域;对所述群组中的服务预期的输入;以及从所述群组中的服务预期的输出。

32.一种系统,包括:

计算处理器;以及

计算存储器,所述计算存储器与所述计算处理器通信地耦合,所述计算存储器中存储有用于执行操作的可执行计算指令,所述操作包括:

接收多个资源描述,每个资源描述描述由设备提供的资源;

对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件;以及

生成与所描述的服务的群组相对应的web服务描述语言文件,其中,生成与所描述的服务的群组相对应的web服务描述语言文件包括编译概述所描述的服务的所述群组中的服务的信息。

33.根据权利要求32所述的系统,

其中,编译概述所描述服务的所述群组中的服务的信息包括至少编译以下内容中的多个:群组标识符;群组功能;群组成员的数目;群组统计;所述群组中的服务的区域;对所述群组中的服务预期的输入;以及从所述群组中的服务预期的输出。

34.一种计算机实现的方法,包括:

第一计算系统接收多个资源描述,每个资源描述描述由设备提供的资源;以及

所述第一计算系统对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件,

其中,生成web服务描述语言文件包括:解析CoRE资源目录文件以及生成与所述CoRE资源目录文件的内容相对应的内容和web服务描述语言标签。

35.一种系统,包括:

计算处理器;以及

计算存储器,所述计算存储器与所述计算处理器通信地耦合,所述计算存储器中存储有用于执行操作的可执行计算指令,所述操作包括:

接收多个资源描述,每个资源描述描述由设备提供的资源;以及
对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件,

其中,生成web服务描述语言文件包括:解析CoRE资源目录文件以及生成与所述CoRE资源目录文件的内容相对应的内容和web服务描述语言标签。

36.一种计算机实现的方法,包括:

第一计算系统接收多个资源描述,每个资源描述描述由设备提供的资源;

所述第一计算系统对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件;

所述第一计算系统对于每个服务来确定所述服务是否对应于现有群组;

在确定服务对应于现有群组时,

更新与所描述的服务的群组相对应的现有web服务描述语言文件,以及

向服务器系统传送所更新的与所描述的服务的群组相对应的web服务描述语言文件;

以及

在确定服务不对应于现有群组时,

创建与所描述的服务的新群组相对应的新web服务描述语言文件,所述所描述的服务的新群组包括被确定为不与现有群组相对应的服务,以及

向服务器系统传送新创建的与所描述服务的新群组相对应的web服务描述语言文件。

37.一种系统,包括:

计算处理器;以及

计算存储器,所述计算存储器与所述计算处理器通信地耦合,所述计算存储器中存储有用于执行操作的可执行计算指令,所述操作包括:

接收多个资源描述,每个资源描述描述由设备提供的资源;

对于每个所接收的资源描述生成描述与由设备提供的资源相对应的服务的web服务描述语言文件;

对于每个服务来确定所述服务是否对应于现有群组;

在确定服务对应于现有群组时,

更新与所描述的服务群组相对应的现有web服务描述语言文件,以及

向服务器系统传送所更新的与所描述的服务的群组相对应的web服务描述语言文件;

以及

在确定服务不对应于现有群组时,

创建与所描述的服务的新群组相对应的新web服务描述语言文件,所述所描述的服务的新群组包括被确定为不与现有群组相对应的服务,以及

向服务器系统传送新创建的与所描述服务的新群组相对应的web服务描述语言文件。

M2M-IOT服务的发布和发现

[0001] 相关申请的交叉引用

[0002] 本申请根据35U.S.C. §119 (e) 要求于2014年7月22日提交的美国临时专利申请号62/027,398的优先权,其内容通过引用整体并入本申请。

背景技术

[0003] 服务指代使其在网络环境中可用和可访问的功能或能力的所限定的集合。设备和/或软件应用向执行特定设备或应用的期望的功能的服务传送请求。

[0004] 服务在各种各样的技术环境中采用。例如,服务被广泛地用于Web相关的技术中。具有特定功能的Web服务可以在特定网络地址处可用。需要特定功能的系统可以经由Web来向web服务传送请求,然后该web服务执行期望的功能。在示例场景中,一个系统可以采用Web服务来与其他方式下不兼容的系统交换数据。

[0005] 服务还被广泛用于机器对机器 (M2M) 和物联网 (IoT) 技术的场境 (context) 中。机器对机器 (M2M) 是一种广泛的标签,其被用于描述使得连网设备能够在没有人类的手动辅助的情况下交换信息和执行动作的技术。物联网 (IoT) 指代这样的场景,其中包括机器和机器的组件的对象被提供有唯一的标识符以及通过网络在对象之间自动传递数据的能力,而不需要人对人或人对计算机交互。有时被称为M2M/IoT场境中的资源的服务被用于提供M2M能力,其包括例如,与安全、计费、和数据管理有关。

[0006] 最近,存在对允许Web服务访问M2M/IoT服务的兴趣。换句话说,存在对允许Web应用来利用由M2M/IoT服务提供的功能的不断增长的兴趣。现有技术已经被证明不足以支持期望的整合。

发明内容

[0007] 申请人在本文公开了用于发布和发现M2M/IoT资源或服务的系统和方法。所公开的系统和方法使得消费者系统——其可以是基于Web的应用——能够发现和访问由M2M/IoT设备提供的功能。

[0008] 在一个示例实施例中,可以是例如M2M网关系统的第一计算系统从M2M/IoT设备接收资源描述。资源描述中的每一个描述特定资源,其可以被视为是在从其接收该描述的设备或系统处可访问的服务。例如,资源描述可以包括描述特定设备可用于在特定设备附近提供测量或环境条件的信息。资源描述可以是任何合适的格式,诸如对资源目录 (RD) 的引用。

[0009] 在一个示例实施例中,对于接收到的每个资源描述,网关创建描述该资源的条目。在一个示例实施例中,网关可以为每个资源创建单独的或个体的文件,其可以被称为服务。被包括在文件中的信息包括关于如何访问特定资源的信息。在一个示例场景中,文件可以包括指定资源何时可用、资源在何处可被访问、以及与访问资源相关联的成本的信息。可以用任何合适的方式格式化个体文件。在一个示例实施例中,可以使用web服务描述语言 (WSDL) 来格式化个体文件。使用web服务描述语言格式化的个体文件可以被称为WSDL-I文

件,并且可以被存储在网关上。

[0010] 在一个示例实施例中,网关可以被进一步编程为根据资源或服务的特性来对资源进行分组。例如,网关可以适于将新接收的资源与先前创建的群组进行比较。网关基于该比较来确定资源是否可以用现有的资源分组来归类,或者该特定资源是否属于新的分组。在资源与用于识别温度的功能有关的示例场景中,网关可以确定该资源可以与类似地提供温度测量的资源的现有分组配对。在一个替选场景中,网关可以确定该资源不类似于任何现有群组。

[0011] 在网关确定该资源与现有群组有关的情况下,网关将该特定资源添加到现有群组。例如,这可以涉及更新存储关于与该群组相关联的资源的文件的信息。该信息可以包括适合于概述与特定群组相关联的资源的任何信息。可以以任何合适的方式格式化包含该概述信息的文件。在一个示例场景中,文件可以被格式化为web服务描述语言文件。在这样的场景下,该文件可以被称为WSDL-G文件。

[0012] 在网关确定该资源不与现有群组有关的情况下,网关可以生成新群组。例如,网关可以生成存储有关新群组的概述信息的新文件。在首次创建群组和文件时,该群组中可以只有一个资源。随着时间的推移,该文件将随着新资源被添加到该群组中而更新。在一个示例实施例中,文件可以被格式化为web服务描述语言文件。

[0013] 根据所公开的实施例的另一方面,网关被编程为公布关于其信息已经被接收到的资源的信息,使得设备和系统可以发现资源的存在并请求对该资源的使用。在一个示例实施例中,网关将关于所定义的资源的信息传输至维护概述群组信息的服务器计算系统。例如,网关可以将信息传输至服务注册基础架构(SRI),该服务注册基础架构作为可用资源或服务的注册表来操作。在一个示例实施例中,网关可以为每个群组传送包含关于该特定资源群组的概述信息的文件。因此,当网关响应于接收到不适合现有群组的资源而创建新群组时,网关可以将对应的文件(WSDL-G)传输至SRI。在群组先前存在并且资源被添加到群组的群组文件中的实例下,网关可以向群组文件先前被传输至其的SRI传输更新。

[0014] 一旦关于资源的群组的信息已经被发布,服务器系统——即SRI——就可以接收发现关于资源群组的信息的请求。例如,可以从服务消费者系统接收请求,该服务消费者系统可以是可能需要使用服务的任何系统。在一个示例场景中,可以从Web应用接收请求。该请求可以提供关于系统正在搜索的服务或资源的类型的各种参数。例如,请求可以指定对在特定区域中以及在特定时间段期间,与特定类型的功能或操作有关的服务的查询。在一个示例场景中,该请求可以指定对于在制造设施的特定部分中提供温度的资源的查询。

[0015] 响应于该请求,资源服务器系统——即SRI——使用该查询中的信息来识别可能响应的资源群组。SRI可以在多个WSDL-G文件中搜索以识别与响应该特定请求的资源群组有关的一个或多个文件。SRI向消费者系统返回该SRI确定响应于该查询的一个或多个WSDL-G文件。

[0016] 消费者系统解析与所识别的服务群组相关联的信息,并确定群组中的哪些与消费者系统可以使用的服务或资源相关联。消费者系统从WSDL-G文件中识别包含有关在特定WSDL-G文件中概述的服务的附加信息的特定网关。然后,消费者系统生成关于消费者系统已经确定其可以使用的特定服务群组的附加信息的请求或查询,并将其传送至所识别的网关。

[0017] 网关处理查询并识别与请求有关的特定服务群组。网关识别包括在请求中识别的特定群组中的特定资源，并检索与每个所识别的资源相对应的个体文件(WSDL-I)。网关将个体文件——其可以是例如WSDL-I文件——传输至消费者系统。

[0018] 消费者系统解析所接收的WSDL-I文件并确定它将访问哪个或哪些服务。已经确定了它希望访问的特定服务的消费者系统从相对应的WSDL-I文件中检索关于访问该特定服务的特定参数。例如，消费者系统可以从WSDL-I文件识别可以访问服务/资源的特定系统以及用于访问资源的特定方式和时间。消费者系统生成并向所识别的系统传送使用该资源的请求。

[0019] 可以是M2M/IoT系统的所识别的系统，根据请求来提供服务。例如，在服务与提供温度读数有关的情况下，所识别的系统可以以预定间隔向用户系统传输温度读数。

[0020] 提供本发明内容来以简化的形式介绍下面在说明性实施例的具体实施方式中进一步描述的一些概念。本发明内容不旨在标识所要求保护的的主题的关键特征或必要特征，也不旨在被用于限制所要求保护的的主题的范围。其他特征如下所述。

附图说明

[0021] 说明性实施例的前述说明内容和以下附加描述在结合附图来阅读时可以被更好地理解。应当理解，所公开的系统和方法的潜在实施例不限于所描绘的那些。

[0022] 图1描绘了描绘服务层的相对定位的示例协议栈。

[0023] 图2描绘了网络中的示例服务层部署。

[0024] 图3是描绘示例oneM2M架构的图。

[0025] 图4是描绘示例oneM2M架构的图。

[0026] 图5是描绘示例WSDL文件的图。

[0027] 图6是描绘服务发布和发现的流程的图。

[0028] 图7是描绘M2M设备和Web服务之间的通信路径的图。

[0029] 图8是描述使用CoAP来记录的操作的图。

[0030] 图9是适于服务发布和发现的系统架构的图。

[0031] 图10A和10B描绘了用于定制WSDL-I和WSDL-G模板的示例用户界面。

[0032] 图11是描绘示例WSDL-G模板的图。

[0033] 图12是将资源映射到WSDL文件中的过程的流程图。

[0034] 图13是描绘资源描述和WSDL文件之间的映射的图。

[0035] 图14是描绘所接收资源描述的处理的图。

[0036] 图15是描绘对资源的更新的处理的图。

[0037] 图16是图示被采用来从个体资源/服务生成群组的示例逻辑的图。

[0038] 图17是描绘示例WSDL-G模板的图。

[0039] 图18是描绘在服务发现期间的示例处理的图。

[0040] 图19是描绘在服务发现期间的示例处理的图。

[0041] 图20是描绘在服务发现期间的示例处理的图。

[0042] 图21是描绘在服务发现期间的示例处理的图。

[0043] 图22是描绘用于实现服务发布和发现的M2M资源配置的图。

- [0044] 图23是从M2M服务层角度描绘服务发布的图。
- [0045] 图24是描绘M2M系统组件之间的服务发布和发现的调用流程的图。
- [0046] 图25是用于服务发布和发现的备选架构的图。
- [0047] 图26A是示例机器对机器 (M2M)、物联网 (IoT) 或 (WoT) 通信系统的图。
- [0048] 图26B是描绘在为M2M应用、M2M网关设备、M2M终端设备、和通信网络提供服务时M2M服务层的操作的示例系统的图。
- [0049] 图26C是适合用作与本文所描述的系统和方法结合的基站的系统的图。
- [0050] 图26D是可被用于实现本文所描述的系统和方法的示例性计算系统的图。

具体实施方式

[0051] 申请人在本文中公开了用于M2M/IoT资源或服务的发布和发现的系统和方法。所公开的系统和方法使得服务消费者系统——可以是Web服务系统或Web应用——能够发现和访问由M2M/IoT设备提供的功能。

[0052] 示例服务架构

[0053] 从网络和架构的角度来看,服务可以被认为存在于“服务层”中。图1提供了图示服务层的相对位置的示例协议栈的图。如图所示,服务层110位于各个传输层和网络层112的顶部。这允许服务层中的服务利用网络层中的能力来提供服务。服务层可以被认为存在于客户端应用层114下方。客户端应用向服务层中的服务作出请求以访问由服务实现的功能。因此,服务层作为提供对由服务呈现的功能和能力的访问的中间件来操作。

[0054] 图2图示了具有被部署在其中的服务层的实例的示例网络拓扑。参照图2,存在于网络应用域210中的用户应用可能需要执行特定功能。用户应用可以在设备30上执行,例如设备30可以是诸如下面结合图26A-D所描述的系统。作为示例,专用于管理网络中的设备的用户应用可能需要关于设备的当前状态的信息。在另一示例中,用户应用可以是Web应用,并且可能需要将数据传输至使用另一数据格式化标准的另一应用。

[0055] 应用向网络服务域212传输对于特定服务的请求。如图所示,网络服务域212包括各个网络节点220。网络节点220是网络内的网络可寻址实体。网络节点220可以是物理项目,诸如:例如设备、网关、或服务器;或者可以是虚拟实体,诸如:例如使用例如VMware的虚拟化软件创建的虚拟机。在图2的示例实施例中,网络节点220包括各个服务器类型,包括例如目录服务器、应用服务器、存储服务器、管理服务器、和服务服务器。网络节点220可以由任何合适的计算系统来实现,诸如:例如下面结合图26D描述的计算系统。在图2的实施例中,经由网关226提供对各个网络222的访问。网关可以由任何合适的计算系统来实现,诸如:例如下面结合图26C或26D所描述的。

[0056] 服务器220和网关226在其上具有服务层230。服务层230是通过应用编程接口(API)的集合和底层网络接口支持增值服务能力的软件中间件层。来自网络应用域210中的应用的、执行特定服务功能的请求被路由到特定服务层230并在该特定服务层230处接收。服务层230处理该请求并将所请求的服务的结果返回给请求应用。托管支持一个或多个服务能力的服务层的网络节点被称为服务节点。

[0057] 参照图2,服务层230中的服务也可以由设备应用——诸如存在于设备应用域214中的那些设备应用——来请求。因此,在诸如例如传感器或致动器的特定设备需要执行特

定功能的情况下,设备可以向网络服务域212中的服务层之一中存在的适当服务传送请求。服务层230处理该请求并将所请求的服务的结果返回给请求设备。设备应用域214中的设备可以使用任何合适的计算系统来实现,例如下面结合图26A-D所描述的那些计算系统。

[0058] 在一些实例下,网络服务域内的服务可能需要由其他服务中的一个提供的功能。因此,服务层230中的服务可以向存在于另一服务层230中的服务传送请求。当接收到该请求的服务完成其处理时,它传送对作出请求的服务的响应。

[0059] 服务层230可以是任何类型的服务层。例如,服务层230中的一个或多个可以是专门面向为移动网络设备提供多媒体服务的IP多媒体子系统(IMS)服务层。作为另一示例,服务层230中的一个或多个可以是M2M/IoT服务层。M2M/IoT服务层是专门面向为M2M/IoT类型设备和应用提供增值服务的一种类型的服务层的示例。近来,若干工业标准团体(例如,ETSI M2M,oneM2M和OMA LWM2M)一直在开发M2M/IoT服务层,以应对与将M2M/IoT类型的设备和应用整合到部署(诸如互联网/Web、蜂窝、企业、和家庭网络)中相关联的挑战。

[0060] M2M服务层可以向应用和设备提供对由服务层支持的M2M中心能力的集合的访问。几个例子包括安全、计费、数据管理、设备管理、发现、配备(provisioning)、和连接管理。这些能力经由API而对应用可用,API利用由M2M服务层定义的消息格式、资源结构、和资源表示。

[0061] oneM2M的目的和目标是开发技术规范,其应对对于公共M2M服务层的需要,该公共M2M服务层可以容易地被嵌入在各个硬件和软件中,并且依赖于将该领域中的广泛多种设备连接到全世界的M2M应用服务器。

[0062] 图3图示了示例基本oneM2M架构。如图所示,oneM2M架构包括应用实体(AE) 310,其在oneM2M方案中提供应用逻辑的。oneM2M公共服务层支持一组公共服务功能(CSF)(即,服务能力)。一个或多个特定类型的CSF的集合的实例化被称为公共服务实体(CSE) 312,其可以在不同类型的网络节点(例如,基础架构节点、中间节点、专用节点)上被托管。服务功能通过参考点Mca 320和Mcc 322呈现给其他实体。参照点Mcn 330被用于访问底层网络服务实体。网络服务实体(NSE) 340向CSE提供服务。M2M服务的示例包括设备管理、位置服务、和设备触发。例如,底层NSE可以是基于3GPP的网络。

[0063] 通常使用表述性状态传递(RESTful)架构来实现oneM2M应用。在这样的实例下,CSF被表示为RESTful“资源”集。资源是架构中唯一可寻址实体,其具有可以经由诸如创建、检索、更新、和删除的RESTful方法操纵的表示。这些资源使用统一资源标识符(URI)而可寻址。资源可以包含子资源和属性。子资源是与父资源具有包含关系的资源。父资源表示包含对其子资源的引用。子资源的寿命受父资源寿命的限制。每个资源支持存储关于该资源的信息的“属性”集。申请人在本文中公开了发布关于M2M/IoT资源/服务的信息的系统和方法以便使它们对诸如Web应用的系统可见。

[0064] 图4描绘了包括示例M2M服务架构的组件的图。M2M服务架构通过向AE提供对M2M服务和M2M服务提供者的访问来增强上面结合图3所描述的oneM2M功能架构。如图4所示,AE 410可以与已经在CSE 414中分组在一起的服务呈现组件412通信地耦合。服务呈现组件412与网络服务利用组件420和远程服务呈现组件422通信地耦合。网络服务利用组件412提供对经由网络服务实体430提供的服务的访问。远程服务呈现组件422提供对来自远程M2M环境432的服务的访问。

[0065] Web服务技术

[0066] 网络服务 (WS) 指代实现软件系统的方法, 该软件系统被设计为支持通过网络计算机之间的可互操作交互。WS提供了在各种平台和框架上运行的软件应用之间的互操作的标准方式。例如, WS可以采用远程过程调用 (RPC), 使得第一程序可以从位于网络中的不同计算机上的第二程序请求服务。WS可以使用可扩展标记语言 (XML) 来实现, 并且可以通过它们的强大可互操作性和可扩缩性以及它们的机器可读描述来表征。WS可以以松散耦合的方式组合来实现复杂操作以使得提供简单服务的程序可以彼此交互来递送精细的增值服务。

[0067] Web服务描述语言 (WSDL)

[0068] Web服务描述语言 (WSDL) 是被用于描述由WS提供的功能的基于XML的接口描述语言。WS的WSDL描述 (也被称为WSDL文件) 提供对以下的机器可读描述: 如何调用服务, 它期望什么输入参数, 它递送什么输出、以及它使用什么传输协议等的。根据所公开的系统的一个方面, WSDL文件可以由服务提供者发布到网络, 使得可以被称为服务消费者的其他软件应用可以发现可用的WS并且容易地确定如何基于WSDL文件中所包括的信息来访问那些WS。

[0069] 采用现有WSDL版本2.0格式化的示例WSDL模板如下所示。如所图示的, 现有的WSDL格式化提供用于识别与服务有关的信息, 包括例如服务使用的数据、服务提供的功能、提供服务的方式、以及可以访问服务的位置。

```

<description...">
    <!-- What data a WS uses (WS 使用什么数据) ->
</types/>
    .....
</types>
<!-- What the WS can do (WS 能够做什么) ->
<interface>
    <operation>
        <input>
            .....
        </input>
        <output>
            .....
[0070]    </output>
        </operation>
    </interface>
<!-- How the WS is provided (如何提供 WS) —>
<binding/>
    .....
</binding>

<!-- Where to access the WS (在何处访问 WS) —>
<service/>
    <endpoint >
        .....
    </service/>
</description>

```

[0071] 如上面的示例所图示, WSDL版本2.0支持被用于描述Web服务的若干元素。更具体地, 现有的WSDL模板支持被用于描述Web服务的以下元素:

- [0072] • <Types>元素描述由特定服务使用的数据。XML范式定义 (XSD) 用于此目的。
- [0073] • <interface>元素定义可以使用特定服务执行的操作 (如作为<interface>的一

部分的<operation>元素中所定义)。操作信息可以包括可以结合服务传递的特定输入、输出、和故障消息。与服务的数据交换的这种格式化在WSDL 2.0中可以被称为消息交换模式(MEP)。

[0074] • <Binding>元素定义WS客户端或消费者如何与WS进行通信。绑定传输协议可以是若干不同协议中的任何协议,包括例如用于REST WS的HTTP或用于传统WS的SOAP (XML+HTTP)。

[0075] • <service>元素将WS的地址(在<endpoint>元素中指定)与特定接口和绑定相关联。

[0076] 通常,为每个Web服务生成WSDL文件。图5图示了与酒店预订服务相对应的示例WSDL文件。如图5所图示,<types>、<interface>、<binding>、和<service>WSDL元素被用于定义由预订服务执行的操作、由预订服务使用的数据、如何与预订服务通信、以及可以访问预约服务的位置。

[0077] WSDL中的消息交换模式(MEP)

[0078] WSDL可以被用于在面向服务的架构(SOA)中定义WS。SOA是一种架构样式,其中提供多件功能的软件组件被封装为不可知的服务并且经由消息来彼此通信。在WSDL被用于支持SOA的情况下,WSDL定义:1)服务能够发送和接收的消息;和2)消息如何通过将消息分组到操作中而相关。WSDL 2.0引入了一种通用机制来描述操作类型,被称为消息交换模式(MEP)。MEP通过采用预定义模板来定义WSDL操作的操作类型,该预定义模板定义了交换属于相同操作的消息的顺序。在WSDL 2.0中定义了以下八个MEP。

[0079] • In-Only:服务接收消息。

[0080] • Robust In-Only:服务接收消息,并在故障的情况下返回故障消息。

[0081] • In-Out:服务接收消息并返回响应消息。

[0082] • In-Optional-Out:服务接收消息并可选地返回响应消息。

[0083] • Out-Only:服务发送消息。

[0084] • Robust-Out-Only:服务发送消息,并且在合作伙伴服务发生故障的情况下接收故障消息。

[0085] • Out-In:服务发送消息并接收响应消息。

[0086] • Out-Optional-In:服务发送消息并可选地接收响应消息。

[0087] 所有这些模式描述了服务和远程调用服务的合作伙伴(即,服务消费者)之间的消息交换。

[0088] 现有服务发布和发现过程

[0089] 图6描绘了现有服务发布和发现处理的逻辑流程。现有的WS发布和发现处理涉及由三个主要系统进行的处理及其之间的通信:服务提供者系统610;服务注册系统612;和服务消费者系统614。服务提供者系统610操作以提供对特定服务的访问。如图6中的流程1所图示,服务提供者系统610发布对于可经由系统610访问的服务中的每一个的描述。发布处理可以涉及将对服务的描述传输至服务注册系统612。服务描述可以被格式化为WSDL文件。

[0090] 服务注册系统612可以充分利用服务注册基础架构(SRI),其可以是例如通用描述、发现、和集成(UDDI)。服务注册系统或SRI 612存储服务描述信息。在一个示例实施例中,SRI 612存储用于被发布到SRI 612的每个服务的WSDL文件。

[0091] 如图6的流程2所图示,服务消费者系统614将请求传输至SRI 612以便发现Web服务。请求可以包括指定所请求的服务的类型的参数。响应于该请求,SRI 612将与在该请求中所指定的参数相对应的服务的WSDL文件转发到服务消费者系统614。

[0092] 服务消费者系统614使用来自接收到的WSDL文件的信息来格式化请求以访问由服务提供者610提供的服务。例如,服务消费者系统614使用WSDL文件中的信息来识别服务提供者系统610提供对特定服务的访问并且将对应的请求传送到系统610。

[0093] 已经尝试了服务发布和注册的替选方法。例如,可以以纯分布式方式来使用通用搜索引擎、特殊WS广告网站、和/或在web上传播WSDL文件来执行服务发布和发现。

[0094] WS语义

[0095] WSDL提供了用于描述服务连同相关输入和输出、端点、和与服务有关的其他信息的手段。即使利用该信息,处理WSDL文件的计算系统也可能需要人类协助来完全处理所有的相关联信息,包括例如接口描述中的有意义的标识符和文本信息。因此,一些现有的努力诸如OWL-S已经集中在向WSDL文件添加语义注释。

[0096] Web应用描述语言 (WADL) 和RESTful服务描述语言 (RSDL)

[0097] 现有的发布和发现系统可以采用与WSDL不同的技术。例如,可以使用被可以用于描述表述性状态传递 (REST) 样式的web应用的web应用描述语言 (WADL) 来描述服务。WADL是通常被采用来描述基于HTTP的Web应用的描述语言,该基于HTTP的Web应用在“资源”作为中心概念的意义上遵循REST或面向资源架构 (ROA) 风格。WADL由Sun Microsystems于2009年8月31日提交万维网联盟 (W3C), 尽管它尚未标准化且尚未得到广泛采用。

[0098] 通过比较,WSDL版本2.0是正式规范,并且已经成为用于描述和定义WS的事实上的标准。WSDL是一种面向服务的描述语言。WS已经成为实现面向服务架构 (SOA) 的最流行的技术之一,其中“服务”是中心概念。现有的WS可以分为两种类型:1) 大WS,它是传统类型,2) RESTful WS。这两种类型的不同之处在于大WS使用SOAP (HTTP+XML) 作为传输协议,并且不遵循REST设计原则。例如,“add”操作绑定到HTTP中的“GET”操作。相比之下,RESTful WS遵循REST设计原则,包括:1) 显式地使用HTTP方法,2) 是无状态的,3) 呈现目录结构类的URI,以及4) 传输XML,JavaScript对象表示法 (JSON)。值得注意的是,基于RESTful WS的系统仍然可以是SOA而不是ROA (其中“资源”是中心概念)。

[0099] RESTful服务描述语言 (RSDL) 也是RESTful WS的机器可读XML描述。RSDL旨在简化基于Web的HTTP架构的WS的重用。但是,RSDL专门设计用于描述RESTful WS,而最新的WSDL 2.0已经实现了这一目的。此外,与WADL类似,与正在被广泛采用的WSDL 2.0相比,RSDL不是用于描述WS的标准化语言。

[0100] M2M系统和M2M服务

[0101] 图7是描绘Web服务系统和M2M系统资源之间的通信路径的网络图。如图所示,M2M区域网络710提供M2M端设备712和M2M网关 (GW) 714之间的连接。M2M区域网络710可以是任何合适的网络,包括例如采用个域网技术的网络,诸如IEEE 802.15、Zigbee、蓝牙等。M2M设备712可以是资源受限的设备,其提供诸如报告诸如湿度或温度的感测信息的服务,或者充当诸如照明开关的控制器。M2M端设备712与M2M GW 714通信,该M2M GW 714通过核心网络720与M2M服务器722通信。M2M服务器722与外部网络和应用730对接。外部应用730可以是服务的消费者,并且可以经由M2M服务器722访问由设备712提供的服务。

[0102] CoAP协议中的资源观察

[0103] 受限应用协议 (CoAP) 是专门开发用于与诸如无线传感器网络的受限节点/网络一起使用的的应用协议。CoAP已经吸引了越来越多的关注,并已成为IoT系统的有前途的消息传送协议。除了支持CoAP服务器和CoAP客户端之间的RESTful资源操作之外,CoAP协议还定义了被称为“观察 (Observe)”的订阅/通知机制,其允许从CoAP服务器(即,服务提供者)向感兴趣的CoAP客户端(即,服务消费者)自动推送资源表示。

[0104] 图8图示了使用CoAP来记录的观察操作的示例。CoAP客户端首先将GET请求与观察选项一起发送至CoAP服务器,以注册感兴趣的资源。客户端从CoAP服务器接收三个通知,其中第一通知在注册时发出,并且另外两个通知在资源的状态改变时发出。通知返回来自CoAP客户端的原始GET请求中包含的相同令牌,因此CoAP客户端可以轻松地将它们与原始GET请求相关联。

[0105] 现有发布和发现处理的限制

[0106] 许多现有的软件系统(诸如企业应用系统)已经通过使用WS技术来以分布式和松散耦合的方式实现。现在有兴趣来扩展现有的方法以便充分利用和/或整合新兴的M2M/IoT服务。因此,开发将这些新兴的M2M服务平滑地整合到现有的基于WS的软件系统中的系统和方法是待解决的根本问题。通过在新兴系统中使用的相异的技术来开发这样的系统是复杂的。例如,许多现有系统采用SOA,而新兴系统通常利用ROA。

[0107] 如上述讨论中所指出的,存在可被用于描述针对服务发布和发现的的目的的服务的若干不同的服务/资源描述语言(例如,WSDL、WADL或RSDL)。这些语言中的每一个都具有阻碍基于Web的系统作出M2M服务的有效发布和发现的限制。对本文的讨论集中于WSDL,因为它是最广泛使用的描述语言。然而,应当理解,其他描述语言可以具有类似的限制,并且可以类似于下面结合WSDL所描述的那样被增强以提供所公开的系统和方法。

[0108] 现有的WSDL发布/发现架构不适用于IoT系统

[0109] 在IoT场境中,通常存在提供资源(诸如温度,湿度等)的大量设备(例如,传感器)。据预计到2020年为止将存在500亿个IoT设备。此外,在IoT网络中,即使对于类似类型的设备,设备也可能有很多变化。例如,提供类似功能的设备可以使用由不同制造商创建的不同软件等。由于设备可能变得不活动——即睡眠——的时间段,IoT设备可能不总是可访问的。IoT设备还经历随时间的动态配置变化(例如,调整睡眠时间表,改变输入和输出格式等),包括可以如何访问设备。

[0110] 如上面结合图6所指出,发布M2M和IoT资源/服务的现有方法涉及每个服务提供者生成与资源中的每一个相对应的WSDL文件,并将资源中的每一个的WSDL文件中的每一个传送到服务注册。但是,这种现有的处理在IoT场境中不是最优的。例如,对于资源受限的设备和网络来说,服务提供者(即IoT设备)和服务注册表之间的直接交互是具有挑战性的,该直接交互是生成、发布、和更新WSDL文件所需的。许多IoT设备具有有限的计算处理能力和有限的功率预算,这两者都限制了与SRI的通信。

[0111] 作为另一示例,来自IoT设备的大量WSDL文件可以容易地使SRI超负荷。如上所指出,很可能存在非常多的IoT设备,每个设备具有一个或多个资源或服务。根据当前实践,每个资源导致生成单独的WSDL文件并将该WSDL文件传输至SRI。传输至SRI并存储在SRI上的大量WSDL文件可能容易地使SRI超负荷并导致长的查询延迟。

[0112] 使用当前的发布和发现程序,表示设备配置中的改变的频繁更新可能使SRI性能劣化。IoT设备适用于动态调整其配置。例如,IoT设备有时调整其配置以保持能量。最佳地,配置中这样的改变将在与设备上的资源相对应的WSDL文件中表示。最佳地,配置中这样的改变应该在位于SRI上的对应WSDL文件中表示。不幸的是,为了更新具有配置上的变化的SRI所需的频繁更新显着增加了SRI的处理开销和效能。

[0113] 用于服务发布和发现的现有方法没有考虑IoT设备可用性。如上所指出,由于例如特定设备进入睡眠,一些IoT设备经历不可用性的时间段。因此,根据当前实践,即使服务消费者系统可以经由SIR来定位服务,消费者系统可能也无法访问特定的服务提供者设备,因为设备可能不可用,即正在睡眠中。

[0114] 申请人还注意到,除了现有的资源发布和发现过程的不足之外,当前被用于描述资源的web服务描述语言(WSDL)不能适应IoT设备的特性。例如,现有的WSDL模板缺乏用于充分描述M2M服务的特性的特征或元素。通常情况是,对于给定的服务,已知三个数据项:可以访问该服务的地方;可以访问该服务的时间;以及与访问该服务相关联的成本。在某些方面,由M2M/IoT设备提供的服务与由披萨餐馆提供的服务没有不同,因为餐馆可以具有相关联的三个数据项:何处-披萨餐馆可以位于第三大道781号;何时-披萨餐厅可能在周一至周五上午9点至下午6点之间营业;和多少钱-一片披萨是每片3.5美元。正如它在广告披萨餐馆以通知潜在客户何处、何时、以及多少钱是有用的,关于M2M和IoT以及发布服务信息,申请人已经注意到,发布指定何时、何处、以及多少钱的信息是有用的。但是,现有的WSDL模板元素无法充分支持此信息的发布。

[0115] 关于服务位于“何处”,在特定位置附近利用服务的概念在IoT系统中是常见的。在一个示例场景中,可能期望将温度传感器设备定位在光/烟雾传感器附近以便检测可能的火灾事件。在另一示例场景中,可能期望将服务定位在非常接近的位置,以便减少在访问特定服务时涉及的网络流量。现有的WSDL模板缺少被标记用于发布和发现位置信息的元素。

[0116] 关于“何时”可以访问服务,IoT设备和它们提供的服务/资源经历不可用性的时间段。例如,IoT设备经历设备睡眠的时间段不是反常的。在不可用性时间段期间尝试访问服务很可能导致服务故障。现有的WSDL模板缺少被标记用于发布和发现与服务有关的时间信息的元素。现有的WSDL模板和处理假设该服务始终可用。

[0117] 关于服务可能花费“多少钱”,关于使用哪个服务的决策可以至少部分地基于与访问该服务的相关联的成本来作出。当前WSDL版本2.0模板不支持服务提供者在服务描述中指定使用其服务的价格(其不一定是真实的钱,例如,使用虚拟货币)。

[0118] 现有的WSDL模板还缺乏用于描述M2M系统中的新服务供应或交互模式的特征。现有的WSDL 2.0规范定义了八个消息交换模式(MEP)。所定义的MEP中的许多描述了单个交易中服务提供者和服务消费者之间的交互。所定义的MEP缺乏在一段时间内描述多个交易或交互的能力。

[0119] 现有的WSDL模板也不支持描述新的交互模式,诸如例如CoAP协议中的“观察”操作。如上结合图8所述,结合观察操作,CoAP客户端在向CoAP服务器注册之后的指定时间段内接收多个通知。现有的WSDL缺乏充分描述这样的特征的结构。

[0120] 强化的M2M/IoT服务的发布和发现

[0121] 申请人已经开发了由M2M/IoT设备提供的服务的发布和发现的新系统和方法。所

公开的系统和方法对于受限的M2M/IoT设备特别有用。根据所公开的实施例的一个方面，IoT设备向M2M GW传输资源描述。GW作为中间节点来操作。对于每个接收到的资源，GW在被称为WSDL-I文件的“个体”WSDL文件中包装该资源，其中“I”标示该文件与个体服务有关。GW还创建共享一些特性的服务群组，并记录可被称为WSDL-G文件的WSDL文件中的分组，其中“G”标示与服务群组有关的文件。GW将经分组的WSDL文件（即，WSDL-G文件）发布到SRI。GW通常不发布个体的WSDL-I文件，并且从而减少由WSDL过程导致的开销。服务消费者系统——其可以是例如Web服务应用——查询满足特定参数的服务的SRI。该SRI返回与查询参数相对应的WSDL-G文件。服务消费者系统使用WSDL-G文件中的信息来定位适当的GW并请求关于在特定WSDL-G文件中指定的服务的特定信息。作为响应，GW返回与特定WSDL-G文件相对应的WSDL-I文件。服务消费者系统使用接收到的WSDL-I文件中的信息来选择一个或多个服务并格式化请求以接收所选择的服务。

[0122] 将GW引入作为发布和发现处理中的中介，具有将可能频繁的WSDL更新预留到M2M区域网络内的益处。GW仅需要向SRI发布WSDL-G文件，这减少了与SRI的通信开销和在SRI内的计算开销。

[0123] 图9A是描绘用于M2M/IoT服务的发布和发现的系统架构和高级流程的图。参照图9A，多个M2M/IoT设备910与GW 912通信地耦合。设备910可以是任何M2M/IoT设备，诸如例如提供感测信息的传感器。例如，设备910可以通过M2M区域网络与GW 912通信。参照图9A，在步骤1，设备910向GW 912传输资源描述。该资源描述描述了由特定设备提供的资源或服务。例如，该信息可以包括与服务做什么、服务位于何处、以及与服务相关联的成本有关的信息。在一个示例实施例中，可以使用诸如下面结合图26A-D所描述的系统来实现设备910和网关912。

[0124] 在步骤2，GW912通过为每个服务生成WSDL-I文件来将个体资源中的每一个包装到服务中。在一个替换示例实施例中，设备910可以生成WSDL-I文件并将该文件传输至GW 912。在一些实例下，设备910可以适于提供多个资源，诸如例如温度、湿度、光等。在一个示例场景中，每个资源可以被包装到单独的WSDL-I文件中。然而，通常，每个设备910仅提供一个资源。因此，外部消费者系统仅需要如之前那样通过WSDL-I文件发现M2M服务，这符合他们现有的SOA风格。

[0125] M2M/IoT设备910可以动态地改变它们的配置或者出于能量效率目的而频繁地变得不可用。当这样的改变发生在设备910处时，对应的WSDL-I文件需要相应地更新。考虑到可能涉及的大量设备，这样的更新可能频繁发生。在所公开的系统，WSDL-I文件被存储在GW 912处，而不是直接发布到SRI 914。因此，所公开的系统和方法避免了跨网络传送大量文件的需要，并且导致改进的可扩缩性。不同于将每个WSDL-I文件发布到SRI，GW 912基于功能上的相似性来对资源进行分组。在一个示例场景中，GW912可以定义记录特定地理位置中的温度读数的设备群组。GW 912创建WSDL格式化文件，被称为记录作为特定群组的成员的服务的WSDL-G文件。

[0126] 参照图9A，在步骤3，GW 912通过将文件传输至SRI 914来公布WSDL-G文件。在一个示例实施例中，SRI 914可以在与如图7所描绘的典型M2M系统架构一致的M2M服务器处实现，并且可以在诸如下面结合图26D描述的设备上实现。GW 912随后可以从设备910接收更新信息。例如，设备可以改变其配置或进入睡眠，这可以导致更新的资源信息被传输至GW

912.GW 912响应于这样的更新可以为适当的服务改变对应的WSDL-I。如果由接收到的资源更新所实现的改变影响WSDL-G文件中所反映的群组信息,则GW912也可以更新对应的WSDL-G。一般来说,WSDL-G不需要被频繁更新,因为它反映了不常受底层WSDL-I文件的变化影响的高级公共信息。因为WSDL-G文件不频繁变化,所以GW 912需要将修改的WSDL-G文件转发到SRI 914的实例相对较少。结果,专用于在GW 912和SRI 914之间传输更新的资源相对于现有方法而减少。

[0127] 对于任何一个WSDL-G文件,可以存在多个WSDL-I文件。因此,WSDL-I和WSDL-G文件可以被视为是在层级上相关的。这种关系反映在涉及多个步骤的发现处理中。参照图9A,如步骤4所示,可以作为利用M2M/IoT服务的任何系统的消费者系统916生成请求并将其传送到SRI 914。在一个示例实施例中,服务消费者系统916可以是使用由web服务提供的信息的基于Web的应用。在一个示例场景中,消费者系统916可以是草坪维护系统,其访问来自IoT/M2M系统的天气报告服务,并使用接收到的信息来实时计划劳动力的分布。在另一示例场景中,消费者系统916可以是交通管理系统,其访问来自路侧IoT/M2M系统的交通状况服务并且使用所接收的信息来动态地指导交通流以便减轻阻塞。在一个示例实施例中,服务消费者系统916可以使用诸如结合图26A-D所描述的系统来实现。该请求可以指定具有定义服务特性的特定参数的查询。例如,该请求可以指定定义在特定仓库设施于6PM和6AM时间之间提供温度读数的服务的参数。SRI 914查询存储在其上的WSDL-G文件,以识别与响应于查询参数的服务群组相关的特定WSDL-G文件。响应的WSDL-G文件被传输至消费者系统916。

[0128] 参照图9A的步骤5,消费者系统916解析所接收的WSDL-G文件中的信息,并识别与特定文件相关联的网关。消费者系统916生成对于与所接收的WSDL-G文件有关的详细信息的请求并将其传送至所识别的网关。GW 912接收该请求,识别与该请求相对应的WSDL-G文件,收集与WSDL-G文件相对应的WSDL-I文件,并且将所收集的WSDL-I文件传输至该请求起源的消费者系统916。

[0129] 在步骤6,消费者系统916处理从GW 912接收到的所接收的WSDL-I文件。消费者系统916使用在WSDL-I文件中提供的信息选择一个或多个服务。使用用于所选择的服务的WSDL-I文件中的信息,消费者系统916识别与所选择的服务相对应的特定设备910,生成访问服务的请求,并将该请求传送到特定设备910。

[0130] 应当理解,关于图9A,步骤(1)-(3)与服务发布有关,步骤(4)-(5)与服务发现有关,步骤(6)与服务访问有关。

[0131] 在所公开的方法中,与在消费者系统联系SRI时传输WSDL文件的现有过程相比,被包括在WSDL-I文件中的服务访问细节在稍后的时间被传输至消费者系统,即在联系GW时传输。在所公开的系统和方法中的处理中稍后提供服务访问细节允许关于动态服务提供者绑定的更大灵活性,动态服务提供者绑定涉及确定特定服务提供者来访问和获得与输入/输出和端点地址信息有关的确切访问细节。这种增强的灵活性使动态改变适应可以在M2M设备内发生的服务状态和异构特性。相比之下,在现有的WSDL发布和发现处理中,服务消费者直接从SRI获得服务访问信息。使用这种现有技术,消费者有可能将无法访问设备上的服务,因为设备的状态可能已经改变并且可能不再可用。此外,可能已经提供类似服务的设备,如果已经被选择则将不被使用。

[0132] 尽管根据本申请的服务的发现涉及首先查询SRI并随后查询GW的两步过程,但是

在一些情形下,发现可以继续对GW的一步请求。例如,如果消费者系统先前已经获得了关于GW和存储在其上的群组和WSDL-I文件的信息,则消费者系统可以直接联系GW以选择适当的WSDL-I文件。

[0133] WSDL-I模板

[0134] WSDL-I模板是通用模板,其被用于为每个个体服务生成WSDL-I文件。如前所提及,由于现有的WSDL 2.0模板仅包括与如何访问服务有关的信息,在M2M场景中使用它不是最优的。申请人提出向现有WSDL模板添加新属性,使得WSDL-I不仅描述如何访问服务,而且还描述场境或社会信息,诸如:例如服务何时可用、服务提供者位于何处、以及使用服务的成本。这些中的每一个都是对M2M场景中成功服务配备具有潜在影响的不可忽略的因素。

[0135] 关于使用服务的成本,通用术语“虚拟价格(virtual price)”在本文中使用并且可以指代任何合适的价值度量,诸如:例如信用、分值、或虚拟钱币,其可以被用于M2M系统以支持或促进社会相关的合作。例如,设备可以赚取信用或赚取虚拟钱币或分值,以用于允许其他系统访问其资源。虚拟价格的价值可以取决于不同的应用场景。虚拟价格可以由网络运营商、设备自身、或网络载体来设置。

[0136] 在一个示例实施例中,WSDL模板已经适用于包括用于接收描述服务何时可用、服务提供者位于何处、以及服务的使用成本的信息的新属性。在一个示例实施例中,该属性已被添加到当前WSDL 2.0模板的<endpoint>区段中,如以下示例中所示。

<!-- Adding new attributes to endpoint (向端点添加新属性) -->

<!-- Where to access WS (在何处访问 WS) -->

<service/>

<endpoint..... >

<!-- when the service is available (WS 何时可用) -->

<availability .../>

<!-- where is the service provider (服务提供者在何处) -->

<geoLocation .../>

[0137]

<!-- How much cost to use the service (do not have to be real dollars) (使用该服务要花费多少钱 (不一定要真实美元)) ~>

<virtualPrice .../>

<endpoint/>

<service/>

<!-- A example <type> and <endpoint> section of a WSDL-I file based on above WSDL-I template (基于以上 WSDL-I 模板的 WSDL-I 文件的示例<type>和<endpoint>区段) >

<xs: schema

```

xmlns :xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://greath.example
om/2004/schemas/resSvc"
xmlns="http://greath.example.com/2004/schemas/resSvc"
"> <xs:schema/>

```

```
<types/>
```

```

<!-- Define three new data elements, stored in
<types> section (定义存储在<type>区段中的三个新
数据元素) -->

```

```

<xs:element name="availability" type="xs:string"
"/>

```

```

<xs:element name="location" type="xs:string"/>

```

```

<xs:element name="price" type="xs:string"/>

```

```
<types/>
```

[0138]

```
<service/>
```

```
<endpoint address="59.169.36.79: 12345">
```

```
<availability element=" availability " >
```

```
Available
```

```
<availability/>
```

```
<geoLocation element="location">
```

```
780 3rd Ave., King of Prussia
```

```
<geoLocation/>
```

```
<virtualPrice element="price">
```

```
0.15 unit per time
```

```
<virtualPrice />
```

```
<endpoint/>
```

```
<service/>
```

[0139] 对于给定的WSDL-I文件,<operation>元素描述了用于输入和输出的消息交换模式和数据类型。输入/输出的实际值不被包括在WSDL文件中。相反,输入将来自服务消费者,

并且输出将在服务提供者处理输入之后来自该服务提供者。关于服务描述信息,诸如本文提出的新属性,该值被包括在WSDL文件中。这类似于<endpoint>部分,其中包括特定的URI以指示可以访问服务的位置。不同的度量系统可以被用于设置该属性的值,而不会降低M2M服务被广泛部署和与其他服务可互操作的能力。根据潜在实施例的一个方面,语义信息被添加到属性以帮助消费者正确地理解那些属性中的值的含义。

[0140] 将理解,在一个示例实施例中,个人可以定制将对于服务指定的属性。例如,个人可以采用由计算系统生成和显示的用户界面来指定什么属性适用于服务并且针对服务被记录。图10A描绘了可以由系统生成并由个人使用来指定要存储在服务的WSDL-I文件中的属性的示例用户界面。如图所示,该用户界面提供复选框,用户可以通过该复选框来选择该个人希望将哪些属性应用于服务,以及将哪些属性包括在服务的WSDL-I文件中。在图10A的示例实施例中,该用户界面提供用户界面特征,其允许用户选择以在WSDL-I文件中包括指定何时服务可用、服务提供者位于何处、以及使用服务的虚拟成本的属性。一旦用户选择了与服务有关的特定属性并选择“确认(confirm)”按钮,则对应的WSDL-I文件可以被配置为与用户输入一致。应当理解,尽管图10A的示例实施例说明了用于选择特定属性的特征,然而取决于要被包括在对应的WSDL-I文件中的属性,用户界面可以被更改和/或扩展以具有用于选择附加和/或不同的属性的附加特征。

[0141] 使得WSDL-I能够在CoAP中描述“观察者(Observer)”操作

[0142] 可以使得WSDL-I文件能够描述在CoAP协议中使用的“观察”操作。在CoAP协议中,当客户端意图订阅针对给定资源的观察操作时,客户端向资源提供者传送特定信息。例如,客户端可以传送以下:要使用的令牌,其允许客户端理解从提供者发送的后续通知;以及观察持续时间,其指示客户端应当接收通知多久。

[0143] 描述WSDL-I模板中的观察操作涉及执行如下所述的特定流程。在一个示例情形中,所涉及的资源是设备上的温度资源。因此,参数名称可以与“温度”有关以便于理解。

[0144] 为了描述观察操作,可以在WSDL-I中定义用于观察注册的新数据元素(即, `xs:element name="observeRegist"`)。观察注册数据元素可以包括要由服务消费者使用的服务令牌和观察持续时间,如以下示例所示。随后在CoAP层内自动生成不同的令牌,CoAP客户端使用该令牌来辨识随后的通知。相比之下,此处定义的observeServiceToken是使得服务消费者(其位于CoAP层上方)能够辨识来自服务提供者的通知。如下面的示例所示,新的数据元素可以在WSDL-I文件的<type>区段中定义。

```
<xs:element name="observeRegist" type="tObserveRegist"/>
```

```
<xs:complexType name="tObserveRegist">
```

```
<xs:sequence>
```

```
[0145] <xs:element name="observeTimeDuration" type="xs:string"/>
```

```
<xs:element name="observeServiceToken" type="xs:string"/>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

[0146] 实现观察操作还涉及定义新的数据元素(即 `xs:element name="`

observeResponse”)以用于通知。observeResponse通知数据元素包括指示例如资源表示(例如,温度)和令牌的元素,如以下示例列表所示。

```
[0147] <xs:element name="observeResponse"
type="tCheckTemperatureResponse"/>
<xs:complexType name="tCheckTemperatureResponse">
<xs:sequence>
[0148] <xs:element name="temperature" type="xs:double"/>
<xs:element name="observeServiceToken" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

[0149] 实现观察者操作还涉及为WSDL-I文件中的<operation>区段中的模式属性定义新值(即,“Observe-In-Out”)。通常,WSDL 2.0允许任何人定义新模式,其可以由可以辨识和理解新模式的WSDL处理器处理。事实上,如果observeTimeDuration被设置为0,则Observe-In-Out可以成为传统的In-Out模式。因此,WSDL-I文件中的<operation>可以被描述为“观察”操作,如以下示例所示。如下所示,操作的模式被设置为“Observe-In-Out”(指示消费者可以在下一时间段中接收一批通知),并且输入/输出消息分别使用如上定义的新数据元素,即observeRegist(被用于注册)和observeResponse(被用于后续通知)。

```
<operation name="opCheckTemperature"
pattern="http://www.w3.org/ns/wsd/Observe-In-Out"
style="http://www.w3.org/ns/wsd/style/iri" wsdbcsafe =
[0150] "true">
<input messageLabel="In" element="observeRegist" />
<output messageLabel="Out" element="observeResponse" />
</operation>
```

[0151] WSDL-G模板

[0152] 除了通过生成WSDL-I文件来将接收到的设备资源中的每一个包装为服务之外,GW还将类似服务归类为群组。GW通过概述在WSDL-I文件中所包括的信息来为群组中的每一个生成WSDL-G文件。替换地或另外地,GW可以从在从M2M/IoT设备接收的资源描述中所包括的信息直接生成WSDL-G文件。示例WSDL-G模板在图11中描述。WSDL-G模板是通用的模板,其被用于生成特定的WSDL-G文件。如图11的示例WSDL-G模板所图示,WSDL-G文件通常包括与服务群组有关的高级或经汇总统计信息。图11的WSDL-G模板被呈现以用于说明。应当理解,在替换实施例中可以使用与图11中描绘的那些属性不同的属性或附加属性。

[0153] 参照图11,示例WSDL-G模板可以包括<groupIdentifier>属性。<groupIdentifier>属性意图接收定义该群组ID的数据。当向GW请求关于在该群组中定义的特定服务群组的信息时,群组ID可以由消费者服务使用。

[0154] WSDL-G模板可以包括<groupFunction>属性。<groupFunction>属性接收描述特定群组的功能的信息。在一个示例场景中,该信息可以包括明确语义(express semantic)信息。

[0155] <numberOfMembers>属性接收指示当前在群组中的成员的数目的信息。在一个示例实施例中,成员数目的值可以不是确切数目,而是以比较性术语表示的更一般化的度量。例如,成员的数目可以被表示为大于或小于特定值,例如大于(>)20。

[0156] <groupStatistics>属性接收与对于服务群组的统计有关的信息。例如,该属性可以包括指示在访问该群组中的服务的成功概率的信息。此外,QoS相关的参数也可以被添加到该模板。在评估是否使用与特定群组相关联的服务时,服务消费者系统可以使用该信息。

[0157] <locationRegion>属性接收指示从其提供群组中的服务的区域的信息。

[0158] <inputAllowed>属性描述了需要从消费者得到的参数以执行服务。<outputExpected>属性描述了群组中服务所期待的输出。<inputAllowed>和<outputExpected>属性通常不为各服务提供具体描述,而是包括群组中服务的输入和输出的概要列表,以供客户端具有初步或高级的理解。

[0159] <GWAddress>属性包括指示如何联系创建特定WSDL-G模板的GW的信息。例如,该信息可以包括可以触达GW的IP地址和/或特定端口。

[0160] 应当理解,在一个示例实施例中,个人可以定制为服务分组所指定的信息。例如,个人可以采用由计算系统生成和显示的用户界面来指定在为特定群组中的每个服务所存储的信息中所包括的属性。图10B描绘了可以由系统生成并由个人使用以指定服务的属性的示例用户界面。如图所示,该用户界面提供复选框,用户可以通过该复选框来选择个人希望在特定服务群组的对应WSDL-G文件中所包括的特定属性。在图10B的示例实施例中,用户界面提供用户界面特征,其允许用户选择是否在WSDL-G文件中包括指定由服务中的每一个提供的服务质量的属性,和/或指定经由其访问服务的网关的特性的属性。一旦用户选择了与服务群组有关的特定属性并选择“确认”按钮,则WSDL-G文件被配置为与用户选择一致。应当理解,尽管图10B的示例实施例图示了用于选择特定属性的特征,然而取决于要被包括在对应的WSDL-G文件中的属性,用户界面可以被更改和/或扩展以具有用于选择附加和/或不同的属性的附加特征。

[0161] 资源描述和WSDL-I服务描述之间的映射

[0162] 在M2M系统中,对其他系统可用的设备操作被称为使用特定描述性句法和语言来描述的资源。在Web服务操作中,提供给其他系统的功能被称为使用特定描述性句法和语言来描述的服务。因此,当GW从设备接收资源描述时,GW需要将来自资源描述的信息格式化为对于期待Web服务描述的系统是可理解的。此过程有时被称为通过生成WSDL-I文件将资源包装到服务中。该处理通常涉及将元素从资源描述映射到WSDL-I文件内的相应元素。

[0163] 有许多不同的方法来描述资源。相应地,没有通用流程可以在所有实例中被使用以供处理资源描述来创建WSDL-I文件。CoRE资源目录(RD)是IETF域中流行的概念,并且通常被用作存储资源描述的存储库。RD充当关于在设备上托管的资源的web链接的存储库。RD实现一组REST接口以供设备来注册和维护被称为资源目录条目的Web链接集。资源目录条目可以充当资源描述。在一个示例实施例中,RD可以与GW位于同一位置,使得M2M/IoT设备向RD进行注册并将其资源描述传输至GW。

[0164] 图12描绘了用于将使用RD描述的资源描述映射到WSDL-I文件的过程。如图12的左侧所示,设备-1可以初始化POST请求以将其温度资源提交给GW,所述温度资源由post的有效载荷部分中的链接格式来描述。Post请求包括用于描述该资源的所有相关信息,该信息被用于生成WSDL-I文件。应当理解,所提供的信息可以包括指定例如资源做什么、资源位于何处、资源何时可用、以及与资源相关联的成本的信息。在接收到该请求后,GW将资源描述存储在RD中,并且执行WSDL相关的流程以生成WSDL-I文件。

[0165] 通常,GW通过解析资源描述中的信息并将所解析的信息映射到WSDL模板的适当元素中,从资源描述生成WSDL-I文件。该映射取决于对所接收的资源描述的格式化而变化。然而,通常,该映射要求GW包括标识来自资源描述的哪些信息与WSDL模板中的哪些元素相对应的信息。

[0166] 图13是描绘所接收的资源描述和所生成的WSDL-I文件之间的示例映射的图。参看图13,参考字符A标志了资源描述中指定传输协议的信息。如对应箭头所指出的,资源描述信息可以在WSDL-I文件中的<binding>元素中使用。

[0167] 参考字符B标志了资源描述中指定特定服务提供者的信息。如对应箭头所指出的,该特定资源描述信息可以在WSDL-I文件中的<service>元素中使用。

[0168] 参考字符C标志了指示设备/资源何时可能不可用的信息。如对应箭头所示,来自资源描述的这个特定信息可以在WSDL-I文件中的<availability>元素中使用。

[0169] 参考字符D标志了指示与特定资源相关联的虚拟价格的信息。如对应的箭头所示,该特定信息可以被映射到WSDL-I文件中的<virtualPrice>元素。

[0170] 参考字符F标志了指示在何处可以访问资源或服务的信息。如对应箭头所示,该特定信息可以被映射到WSDL-I文件中的<geoLoacation>元素。

[0171] 参考字符E标志了指定资源或服务的类型的信息。如对应的箭头所示,信息可以被映射到WSDL-I文件中的<type>元素。

[0172] 参考字符G标志了指定特定资源的接口信息的信息。如对应的箭头所示,信息可以被映射到WSDL-I文件中的<Interface>元素。

[0173] 服务分组

[0174] GW识别具有类似特性的服务群组,并在WSDL-G文件中记录所识别的分组。GW可以基于任何有用的准则对服务进行分组。在一个示例场景中,GW可以对具有类似功能的服务进行分组。通常,当GW从设备接收资源描述时,存在两个典型的场景。在第一场景下,资源属于新类型,在这种情况下,GW为资源定义新群组,并为该群组生成WSDL-G文件。在第二种场景下,资源描述定义对现有资源描述的更新,在这种情况下,GW更新该群组和与特定资源相关联的WSDL-G文件。

[0175] 图14描绘了当在GW处接收到与新资源有关的资源描述时执行的处理的流程图。参看图14,在步骤1,GW从设备接收新的资源描述。资源描述可以处于适合于传输与资源有关的相关细节的任何格式,包括例如指定所提供的功能的信息、资源可用的时间、资源的位置、以及使用资源的成本。在一个示例场景中,使用RD来传输资源。

[0176] 再次参看图14,在步骤2,GW解析资源信息,生成新的WSDL-I文件,并将该文件存储在存储器中。GW可以处理所接收的资源描述以执行如上面结合图13所描述的映射。

[0177] 在步骤3,GW将接收到的资源与由GW定义的现有群组进行比较。GW评估资源是否可

以归类到现有群组。如果可以,则处理在步骤4继续,其中更新现有WSDL-G文件以反映特定资源的接收。如果资源无法被归类至现有群组,则处理在步骤5继续,其中创建新的WSDL-G文件。

[0178] 在步骤4,GW将新识别的资源添加到现有群组。更具体地,GW更新现有群组的WSDL-G文件以反映新的资源/服务。在更新WSDL-G文件时,GW评估向群组中添加新成员是否需要更新先前被传输至一个或多个SRI的任何WSDL-G文件。例如,如果新资源可以提供比群组中现有成员更丰富的信息,则WSDL-G文件中的<Output Expect>区段(参见图13)可能需要被更新以反映此改变。如果GW确定先前分发的WSDL-G文件需要更新,则GW向SRI发起WSDL-G更新请求。

[0179] 在步骤5,在GW确定新资源不属于任何现有群组的实例下,GW定义新群组并为该群组创建新的WSDL-G文件。当首次定义群组时,经只存在该群组的一个成员。

[0180] 在步骤6,GW向一个或多个SRI发布新的WSDL-G文件。

[0181] 图15描绘了当GW接收到向现有资源提供更新的资源描述时执行的处理的流程图。如图所示,在步骤1,GW从设备接收资源更新。典型地,M2M/IoT设备可以经历反映设备的输入/输出、服务可用性、或设备的操作的任何其他方面的变化的动态变化。结果,当设备的状态改变时,GW可以频繁地从设备接收资源更新请求。

[0182] 在步骤2,取决于更新请求中所包括的信息,GW更新已经被更新的特定资源的对应WSDL-I文件。应当理解,在本文中所公开的改进处理下,WSDL-I文件通常仅存储在GW处而不存储在SRI设备。因此,M2M/IoT设备典型所具有的服务的频繁更新不需要花费资源来更新SRI。

[0183] 在步骤3,除了更新WSDL-I文件之外,GW还评估资源更新是否可以影响其相应服务所属的群组。如果GW确定资源更新不需要更新群组和对应的WSDL-G文件,则不需要进一步的动作,并且处理在步骤4处终止。如果GW确定资源更新确实需要更新群组,则处理在步骤5处继续。

[0184] 在步骤5,GW更新群组以反映接收到的服务更新。例如,如果接收到的更新指示设备将由于电池能量耗尽而变得不可达,则GW将从相应群组中删除服务。GW更新群组的WSDL-G文件。

[0185] 在步骤6,GW确定除了更新WSDL-G文件的本地副本之外,GW是否还需要更新存储在一个或多个SRI上的WSDL-G文件的任何副本。例如,如果具有特定输出接口的所有设备变得不可用,则GW将修改WSDL-G文件中的<outputExpect>区段,因为不再支持该特定输出。在这种场景下,GW可以确定该更新需要对群组和WSDL-G进行这样重大的改变,使得该改变应当被推送到当前维护WSDL-G的副本的任何SRI。在其他实例中。

[0186] 上面结合图13和14的讨论指涉生成WSDL-G文件以记录所识别的服务群组。GW识别具有类似特性的服务群组,并在WSDL-G文件中记录所识别的分组。GW可以基于任何有用的准则来对服务进行分组。在一个示例场景中,GW可以对具有类似服务功能的服务进行分组。可以以任何合适的方式执行对哪些服务具有类似功能的确定。在一个示例场景中,其中链接格式被用于描述资源并且“ct”参数用于指示资源的功能,GW可以创建包括具有相同“ct”的所有资源/服务的群组。

[0187] 图16提供了被采用来从个体的资源/服务生成群组的示例逻辑的图示。在图16所

描绘的示例中,两个资源描述已经提交给GW并且二者都与温度有关。因此,GW可以定义被称为“温度”的新群组,并且可以将该两个资源指派给该群组。在一些实例下,资源可以被表征为属于相同群组,但是在个体资源/服务之间可以具有变型。例如,从可以与链接格式规范中所定义的结果内容相对应的“rt”参数,资源/服务中的一个以摄氏(Celsius)温度提供读数,而另一个以华氏(Fahrenheit)温度提供读数。此外,两个资源关于它们的“if参数(接口描述)”是不同的。“if”参数指示资源/服务两者之一能够提供丰富的信息,而另一些仅提供基本信息。这可能产生这样的场景,其中提供资源/服务的设备中的一个支持“传感器丰富”接口,由此其不仅返回温度读数,而且返回相关联的场景信息,诸如感测/采样时间戳、最近的趋势数据(变得更冷或更暖)等。在相同的场景下,具有“基于传感器的”接口的另一设备可以仅单单返回温度值。

[0188] 关于图16的示例,在定义“温度”群组之后,GW为生成该群组的WSDL-G文件。与图16的示例相对应的示例WSDL-G文件在图17示出。如图17所示,GW使用<groupId>参数向该群组指派了特定群组标识符,其在该实例中具有值“TEM0909011”。GW已经定义了<groupFunction>参数的值,在这种示例下,其已经被指派值“温度(Temperature)”。使用<numberOfMembers>参数来识别群组中的成员数目,其在该特定示例中被指派值20。成员数目的值可以表示已经与该群组相关联的资源的数目(其可以对应于多个设备)。<groupStatistics>参数指示访问该群组中的服务的尝试中有90%是成功的。使用<locationRegion>参数识别群组的位置,该参数在该特定示例中指示该群组的成员被分布在阿默斯特(Amherst)镇区域中。WSDL-G文件包括<inputAllowed>参数,以用于识别群组成员通常接受的查询输入的类型。在此特定示例中,<inputAllowed>参数指示此群组中的服务接受邮政编码或地理坐标的查询/输入值。换句话说,服务响应通过邮政编码或地理坐标作出的对温度读数的查询。最后,WSDL-G文件包括<gwAddress>参数,该参数识别与群组中定义的服务相关联的网关的地址,并且通过该网关地址可以联系群组中的服务。在图17的特定示例中,<gwAddress>参数识别可以访问GW的特定互联网协议地址。

[0189] 服务发现和访问处理

[0190] 结合图13至16的讨论主要应对描述用于服务发布的示例实施例。一旦已经向GW注册了服务并且服务群组已经被发布到SRI,则该服务可以被服务消费者系统发现和访问。图18描绘了服务发现的示例过程的流程图。参照图18,在步骤1,可以是例如Web服务的服务消费者系统消费者-1——Consumer-1——初始化对于由M2M系统提供的服务的服务发现请求。换句话说,服务消费者系统确定其需要使用服务以便满足应用或业务需求。

[0191] 在步骤2,服务消费者系统消费者-1生成请求并将其传输至SRI。请求可以被格式化为查询。例如,消费者-1可以生成并传送指定期望的服务的特性的查询。在一个示例场景中,请求可以指定期望的功能、功能在其期间被需要的时间、以及提供服务的优选位置。应当理解,上面结合图17讨论的值中的任何一个可以是服务消费者系统提交的查询的基础。可以使用任何合适的技术来格式化和传输请求/查询。

[0192] 在步骤3,SRI使用请求/查询中的信息来搜索其上存储的群组信息,以识别满足由服务消费者系统提交的搜索准则的服务群组。更具体地,SRI搜索在先前已被转发到SRI的各个WSDL-G文件中指定的数据。在识别满足指定准则的服务群组——即WSDL-G文件——时,SRI将响应的WSDL-G文件的列表传输至消费者-1系统。

[0193] 在一些场景下,可能不存在与由消费者-1提交的选择准则相对应的WSDL-G文件。在这样的场景下,SRI可以不返回群组信息。在这样的实例下,消费者-1系统可以生成具有一组新的选择准则的新查询。

[0194] 在步骤4,消费者-1系统解析所识别的WSDL-G文件的内容,并选择应当获得其进一步信息的一个或多个群组。换句话说,消费者-1系统确定哪些群组是用于提供期望服务的期望候选。消费者-1系统一旦选择了特定群组,它就识别存储与该群组有关的各文件的网关。从与所选择的群组相对应的WSDL-G文件中检索标识该网关的信息。

[0195] 在图18的步骤5,消费者-1生成查询并将其传送至特定网关—GW-1,其在与所选择的服务群组相对应的WSDL-G文件中被识别。查询请求关于在所选服务分组中识别的特定服务的信息。在一个示例场景中,该通信请求与在特定WSDL-G文件中识别的服务相对应的WSDL-I文件。

[0196] 在步骤6,GW-1解析来自消费者-1的查询,并且识别所请求的信息。在一个示例场景中,GW-1搜索其存储器以识别与在请求中识别的WSDL-G文件相对应的WSDL-I文件。换言之,GW-1针对WSDL-I文件来搜索其存储器并且针对在所识别的群组中所包括的服务来检索该WSDL-I文件。根据所公开的实施例的方面,如果所识别的群组中的服务的数目非常大,则GW-1可以限制返回列表中的条目的数目,以便加快搜索时间。GW-1将识别的WSDL-I文件传输至消费者-1系统。

[0197] 在步骤7,消费者-1系统解析从GW-1返回的WSDL-I文件。该群组中的服务中的每一个可用的所有信息对消费者-1可用。消费者-1基于在WSDL-I文件中所包括的信息来选择一个或多个服务。

[0198] 在步骤8,消费者-1生成请求并将其传送到所选择的一个或多个服务。消费者-1系统使用针对所选择的服务的WSDL-I文件中的信息以确定如何格式化该请求。例如,消费者-1从WSDL-I文件确定用于向提供服务的设备传送请求的具体方式和格式。

[0199] 在一个示例场景中,消费者-1系统将WSDL-I请求传送到提供所选择的服务的设备,其在图18中被标识为设备-1。在一个备选场景中,可能有必要在消费者-1系统不适于以其进行通信的协议中与设备-1进行通信。在这样的场景下,可以通过GW-1来路由与设备-1的通信。GW可以在服务访问过程期间执行转换或代理功能。在一个示例场景中,在消费者-1系统适于使用HTTP进行通信但是设备-1适于使用CoAP协议的情况下,可能需要这样的转换。

[0200] 最后,在步骤9,设备-1执行消费者-1所请求的服务。

[0201] 可用性-感知服务发现和访问

[0202] 如上所讨论,根据所公开的实施例的一个方面,WSDL模板可以被修改以支持记录关于服务的附加信息,包括服务可用性、服务位置、和服务价格。作为已经捕获该信息的结果,可以在服务发现处理和服务访问期间使用该信息。

[0203] 图19图示了在服务发现期间的示例处理,其中已经为每个服务记录了与服务可用性有关的信息。参照图19,消费者-1遵循以上结合图18所述的过程以发现和访问设备-1上的服务-1。作为该过程的一部分,消费者-1接收关于服务-1的服务/设备可用性信息。在一个示例场景中,服务/设备可用性信息可以指示服务-1被调度为在特定日期和时间变得不可用。在这样的场景下,并且如图198的步骤4所示,消费者-1服务识别服务-1将变得不可

用,并且在步骤5主动地联系GW-1以搜索可以提供与由设备-1提供的服务相同类型的服务的备选设备。应当理解,在大多数实例下,消费者-1服务不需要通过联系SRI来发现新服务,因为GW-1能够直接返回在与服务-1相同的群组中的其它可用设备的备选的WSDL-I文件。在步骤6,GW-1在存储器中搜索提供与服务-1类似的服务并且在所需时间期间可用的其他服务。GW-1向消费者-1通信,设备-2上的服务-2是替换服务-1的候选。该通信包括服务2的WSDL-I文件。在步骤7,消费者-1系统使用服务-2的WSDL-I文件中的信息以生成请求并将其传送至服务-2。

[0204] 应当理解,存在利用可用性信息的其他方法。例如,支持可用性-感知服务发现和访问的一个备选方式是使消费者系统向GW指定其需求,并且GW返回一批服务,使得消费者系统可以在一个服务变得不可用时按顺序访问所述服务。

[0205] 位置-感知服务发现和访问

[0206] 图20图示了在服务发现期间的示例处理,其中已经对于每个服务记录了与服务的地理位置有关的信息并将其存储在对应的WSDL-I文件中。参照图20,消费者-1遵循如上结合图18所概述的过程以发现和访问设备-1上的服务1。作为该过程的一部分,消费者-1系统接收关于服务-1的服务/设备地理位置信息。基于应用或业务需求,在步骤4,消费者-1随后可以要求接近服务-1的一个或多个附加服务。在一个示例场景中,其中服务-1涉及报告温度并且消费者-1是公司的企业系统中的火灾警示应用,消费者-1可以确定其需要在服务-1附近的光或烟雾的附加读数。在步骤5,消费者-1系统生成并向GW-1传送对于具有特定特性的服务的请求。因为该请求是对于接近服务-1的服务的,所以GW-1很可能还在其上存储与所请求的服务有关的WSDL-I文件。在步骤6,GW-1搜索满足所请求准则的服务并且识别设备2上的服务2。GW-1向消费者-1系统传送识别服务2并且包括服务2的WSDL-I文件的响应。在步骤7,消费者-1系统传送访问服务-2的请求。

[0207] 虚拟价格-感知服务发现和访问

[0208] 图21图示了在服务发现期间的示例处理,其中已经为每个服务记录了与虚拟价格有关的信息。参照图21,在步骤1,消费者-1系统使用从SRI检索的WSDL-G文件来将GW-1识别为具有关于与WSDL-G相对应的服务的信息。在步骤2,消费者-1向GW-1传送对于进一步的服务发现的请求。如步骤3所示,利用在WSDL-I中所包括的虚拟价格信息,设备自身或网络运营商可以出于不同目的而动态地更新该价格。例如,如果设备已经长时间过载并且与其他对等体(peer)相比消耗了太多的能量,则由该设备提供的服务的虚拟价格可以被设置为更高的值,使得服务访问请求中的一些可以被卸载到相同群组中的其他对等体。在WSDL-I文件中包括价格信息的能力使得服务提供者能够具有取决于任何数目的变量的更灵活的服务方案,所述变量诸如:例如请求者的位置、作出请求的组织的身分、或作出请求的组织的成员资格,其中的任何变量可能导致将收取的不同价格。

[0209] 再次参照图21,在步骤4,GW生成包括WSDL-I文件的候选者列表并将其传送到消费者-1。WSDL-I文件中的信息包含与每个服务相关联的虚拟价格信息。在步骤5,消费者-1评估WSDL-I文件中的每一个以便选择服务。作为评估的一部分,消费者-1考虑使用服务中的每一个的相对价格。在一个示例场景中,消费者-1系统至少部分地由于服务的相对价格而确定使用服务-2。在步骤6,消费者-1向服务2传送请求。在步骤7,服务2返回来自该服务的结果。

[0210] oneM2M实施例

[0211] 应当理解,存在本文所公开的系统和方法的许多不同的可能实施方式。在一个示例实施例中,所公开的用于资源发布和发现的方法可以使用根据oneM2M规范的、基于RESTful资源的编程接口来实现。

[0212] 通常,并且如上面结合图3所述,oneM2M公共服务层支持一组公共服务功能(CSF)。一个或多个特定类型的CSF的集合的实例化被称为公共服务实体(CSE),其可以被托管在若干不同类型的网络节点中的任何节点上(例如,基础架构节点、中间节点等,并且它们对应的CSE分别被称为IN-CSE,MN-CSE)。oneM2M应用通常被称为应用实体(AE)。

[0213] 在用于服务发布和发现的系统的示例M2M实施例中,定义了两个新资源<WSDL-I>和<WSDL-G>。图22图示了两个资源的示例实施方式。该图使用以下oneM2M约定来描述资源:方框被用于表示资源和子资源;具有圆角的方框用于表示属性;定义了每个属性和子资源的多样性;并且用“<”和“>”划界的资源名称指示在创建资源期间指派的名词。

[0214] 参照图22,资源<WSDL-I>和<WSDL-G>中的每一个具有一批属性和子资源。例如,如前所述,<WSDL-I>资源中的属性对应于如WSDL-I文件中定义的不同数据元素,诸如类型、接口、绑定等。类似地,<WSDL-G>资源中的属性对应于如上文结合图11所述的不同数据元素。

[0215] 应当理解,在M2M系统内使用M2M资源<WSDL-I>和<WSDL-G>。M2M GW生成基于xml的WSDL-I和WSDL-G文件,这些文件由外部服务消费者发现和访问。如图所示,基于xml的WSDL-I和WSDL-G文件可以被存储在<container>类型的资源中,如图中由批注<XML-based WSDL-I file>和<XML-based WSDL-I file>指示。

[0216] M2M GW作为边功能/组件操作。M2M GW不仅充分利用当前的oneM2M架构以用于在M2M系统内的不同节点之间通信,而且使得使用常规的基于WS的协议的外部服务消费者系统能够访问M2M服务。

[0217] 图23图示了从M2M服务层角度的服务发布。参照图23,若干oneM2M服务层实体分别位于M2M设备、M2M GW、和M2M服务器上。更具体地,AE在M2M设备上操作,MN-CSE在M2M GW上操作,并且IN-CSE在M2M服务器上操作。因此,可以是例如温度读取应用的AE能够通过向MN-CSE注册来提交其资源描述。MN-CSE生成<temperature-App>资源和子<WSDL-I>资源。在创建<WSDL-G>资源作为<IN-CSEBase>/<MN-CSEBase>的子资源的情况下,MN-CSE通过向IN-CSE注册来向SRI发布WSDL-G。

[0218] 图24图示了与上面的过程相关联的详细调用流程。参照图24,步骤1-7与报告/生成资源描述有关,而步骤8-15与更新资源描述有关。

[0219] 在图24的步骤1,AE在受限M2M设备上执行。在一个示例场景中,AE读取并报告由设备测量的温度信息。在发现在M2M GW节点上运行的MN-CSE之后,AE将其自身注册到MN-CSE。注册消息包括M2M设备提供的资源的描述。资源描述包括诸如上面结合对WSDL-I文件的讨论所描述那些的属性。

[0220] 在步骤2,MN-CSE生成与AE相对应的<application>资源。使用在注册消息中提供的资源描述,MN-CSE生成可以被放置在<application>资源下的<WSDL-I>资源。

[0221] 在步骤3,在成功完成注册和生成资源之后,MN-CSE生成确认并向AE传送确认。

[0222] 在步骤4,MN-CSE识别AE及其资源是否可以被指派给现有的服务群组。如果MN-CSE确定该资源无法被指派给新群组,则它生成新群组。

[0223] 在步骤5, MN-CSE生成并向在M2M服务器上操作的IN-CSE传送资源创建请求。该创建请求包括与先前讨论的WSDL-G有关的所有属性。

[0224] 在步骤6, 在从MN-CSE接收到请求之后, IN-CSE基于在步骤5的请求中转发的群组描述信息来生成<WSDL-G>资源。

[0225] 在步骤7, 在创建<WSDL-G>资源之后, IN-CSE生成确认该动作的、对MN-CSE的响应。

[0226] 在处理的此点处, 新资源的发布完成。如上所述, 资源/服务可用于由消费者系统发现, 并且个体资源可用于被请求。

[0227] M2M设备经历状态的改变。例如, M2M设备可以经历他们睡眠的时间段以便保持资源。因此, M2M设备的操作状态频繁地改变。参照图24, 在步骤8, 当M2M设备上的AE改变状态或者具有即将到来的状态改变时, AE生成更新请求并将其传输至MN-CSE。该请求包括关于状态改变的信息以及应当进行改变以更新<WSDL-I>资源的一个或多个属性的信号。

[0228] 在步骤9, MN-CSE更新<WSDL-I>资源以反映状态的改变。更新的类型可以变化。例如, 更新可能与资源中的属性有关, 诸如, 例如修改资源中“可用性”属性以反映节点睡眠的请求。在另一示例场景中, 由于故障的硬件, 更新可能需要删除该<WSDL-I>资源。

[0229] 在步骤10, 在更新<WSDL-I>资源之后, MN-CSE生成确认并将其传送至AE。

[0230] 在步骤11, MN-CSE确定需要对<WSDL-I>所属的群组的更新, 以便将该更新反映到<WSDL-I>资源。

[0231] 在步骤12, MN-CSE生成资源更新请求并向IN-CSE传送该资源更新请求。该请求指定更新IN-CSE上的对应<WSDL-G>资源。

[0232] 在步骤13, 在从MN-CSE接收到请求之后, IN-CSE基于被包括在请求中的群组更新描述来更新<WSDL-G>资源。

[0233] 在步骤14, 在更新<WSDL-G>资源之后, IN-CSE生成并向MN-CSE传送对更新的确认。

[0234] 备选服务发布和发现

[0235] 应当理解, 存在与本文所公开的系统和方法一致的服务发布和发现的附加实施例。在上面结合图9和22描述的实施例中, GW在帮助资源受限设备方面起着重要作用。在诸如图25所描绘的备选实施例中, 独立设备(例如, 蜂窝电话)可以直接与M2M服务器通话而不经GW。因此, 在一个备选实施例中, 由GW执行的上述处理可以扩展到M2M服务器。另外, SRI可以被实现为如图25所示的单独的实体, 而不是被实现在M2M服务器上。

[0236] 示例计算环境

[0237] 图26A是可以实现一个或多个所公开的实施例的、示例机器对机器(M2M)、物联网(IoT)或(WoT)通信系统10的图。通常, M2M技术为IoT/WoT提供构建块, 并且任何M2M设备、M2M网关、M2M服务器、或M2M服务平台可以是IoT/WoT以及IoT/WoT服务层等的组件或节点。通信系统10可以被用于实现所公开实施例的功能, 并且可以包括功能和逻辑实体, 诸如AE 310、CSE 312、NSE340、GW 226、GW 912、SCS 916、和UE 916以及产生用户界面的逻辑实体。

[0238] 如图26A所示, M2M/IoT/WoT通信系统10包括通信网络12。通信网络12可以是固定网络(例如, 以太网、光纤、ISDN、PLC等)或者无线网络(例如, WLAN、蜂窝等)或异构网络的网络。例如, 通信网络12可以包括向多个用户提供诸如语音、数据、视频、消息传送、广播等的内容的多个接入网络。例如, 通信网络12可以采用一种或多种信道接入方法, 诸如码分多址

(CDMA)、时分多址(TDMA)、频分多址(FDMA)、正交FDMA(OFDMA)、单载波FDMA(SC-FDMA)等。此外,通信网络12可以包括诸如:例如核心网络、互联网、传感器网络、工业控制网络、个域网络、融合个人网络、卫星网络、家庭网络、或企业网络的其他网络。

[0239] 如图26A所示,M2M/IoT/WoT通信系统10可以包括基础架构域和场域(Field Domain)。基础架构域指代端到端M2M部署的网络侧,而场域指代通常位于M2M网关后的区域网络。场域和基础架构域可以都包括各种不同的网络节点(例如,服务器、网关、设备等)。例如,场域可以包括M2M网关14和终端设备18。应当理解,根据需要,任何数目的M2M网关设备14和M2M终端设备18可以被包括在M2M/IoT/WoT通信系统10中。M2M网关设备14和M2M终端设备18中的每一个被配置为使用通信电路经由通信网络12或直接无线电链路来传送和接收信号。M2M网关14允许无线M2M设备(例如,蜂窝和非蜂窝设备)以及固定网络M2M设备(例如,PLC)来通过诸如通信网络12的运营商网络或直接无线电链路进行通信。例如,M2M终端设备18可以收集数据并且经由通信网络12或直接无线电链接将该数据发送至M2M应用20或其他M2M设备18。M2M终端设备18还可以从M2M应用20或M2M终端设备18接收数据。此外,如下所述,可以经由M2M服务层22向M2M应用20发送数据和信号以及从M2M应用20接收数据和信号。M2M终端设备18和网关14可以经由各种网络进行通信,所述网络包括例如蜂窝、WLAN、WPAN(例如,Zigbee、6LoWPAN、蓝牙)、直接无线电链路和有线线路。

[0240] 示例性M2M终端设备18包括但不限于平板电脑、智能电话、医疗设备、温度和天气监视器、连接的汽车、智能仪表、游戏控制台、个人数字助理、健康和健身监视器、灯、恒温器、电器、车库门和其他基于致动器的设备、安全设备、和智能插座。

[0241] 参照图26B,场域中图示的M2M服务层22为M2M应用20、M2M网关设备14、和M2M终端设备18以及通信网络12提供服务。通信网络12可以被用于实现所公开的实施例的功能,并且可以包括例如AE 310、CSE 312、NSE 340、GW 226、GW 912、SCS 916、和UE 916的功能和逻辑实体。M2M服务层22可以由一个或多个服务器、计算机、设备、虚拟机(例如,云/存储群等)等来实现,包括例如下面描述的图20C和20D中所图示的设备。将理解,M2M服务层22可以根据需要与任何数目的M2M应用、M2M网关14、M2M终端设备18、和通信网络12通信。M2M服务层22可以由网络的一个或多个节点来实现,其可以包括服务器、计算机、设备等。M2M服务层22提供应用于M2M终端设备18、M2M网关14、和M2M应用20的服务能力。M2M服务层22的功能可以以各种方式实现,例如作为web服务器,在蜂窝核心网络中,在云中等。

[0242] 与所图示的M2M服务层22类似,在基础架构域中存在M2M服务层22'。M2M服务层22'为基础架构域中的M2M应用20'和底层通信网络12'提供服务。M2M服务层22'还为场域中的M2M网关14和M2M终端设备18提供服务。将理解,M2M服务层22'可以与任何数目的M2M应用、M2M网关和M2M设备通信。M2M服务层22'可以通过不同服务提供者来与服务层交互。M2M服务层22'由网络的一个或多个节点实现,其可以包括服务器、计算机、设备、虚拟机(例如,云计算/存储群等)等。

[0243] 也参照图26B,M2M服务层22和22'提供多样的应用和垂直元(vertical)可以充分利用的服务交付能力的核心集合。这些服务能力使M2M应用20和20'能够与设备交互并且执行诸如数据收集、数据分析、设备管理、安全性、计费、服务/设备发现等功能。本质上,这些服务能力免除了应用实现这些功能的负担,从而简化应用开发并且降低了投放市场的成本和时间。服务层22和22'还使得M2M应用20和20'能够通过与服务层22和22'提供的服务结合

的各个网络12和12'进行通信。

[0244] 本申请的方法可以被实现为服务层22和22'的一部分。服务层22和22'是通过一组应用编程接口 (API) 和底层网络接口来支持增值服务能力的软件中间件层。ETSI M2M和oneM2M二者都使用可以包含本申请的连接方法的服务层。ETSI M2M的服务层被称为服务能力层 (SCL)。SCL可以在M2M设备 (其中它被称为设备SCL (DSCL))、网关 (其中它被称为网关SCL (GSCL)) 和/或网络节点 (其中它被称为网络SCL (NSCL)) 内实现。oneM2M服务层支持一组公共服务功能 (CSF) (即服务能力)。一个或多个特定类型的CSF的集合的实例化被称为可以被托管在不同类型的网络节点 (例如,基础架构节点、中间节点、专用节点) 上的公共服务实体 (CSE)。此外,本申请的连接方法可以被实现为M2M网络的一部分,其使用面向服务的架构 (SOA) 和/或面向资源的架构 (ROA) 来访问诸如本申请的连接方法的服务。

[0245] 在一些实施例中,M2M应用20和20'可以与所公开的系统和方法结合使用。M2M应用20和20'可以包括与UE或网关交互并且还可以结合其他公开的系统和方法使用的应用。

[0246] 在一个实施例中,诸如例如AE 310、CSE 312、NSE 340、GW 226、GW 912、SCS 916、和UE 916的逻辑实体可以被托管在M2M节点所托管的M2M服务层实例内,M2M节点诸如图20B中所示的M2M服务器、M2M网关、或M2M设备的。例如,诸如AE 310、CSE 312、NSE 340、GW 226、GW 912、SCS 916、和UE 916的逻辑实体可以包括M2M服务层实例内的个体服务能力或者作为现有服务能力内的子功能。

[0247] M2M应用20和20'可以包括在各种行业中的应用,诸如但不限于交通、健康和保健、互连家庭、能量管理、资产跟踪、以及安全和监视。如上所提及,跨设备、网关、服务器和系统的其他节点运行的M2M服务层支持多个功能,诸如例如数据收集、设备管理、安全、计费、位置跟踪/地理围栏 (geo-fencing)、设备/服务发现、和遗留系统整合,并且将这些功能作为服务来提供给M2M应用20和20'。

[0248] 通常,服务层22和22'定义通过一组应用编程接口 (API) 和底层网络接口来支持增值服务能力的软件中间件层。ETSI M2M和oneM2M架构二者都定义了服务层。ETSI M2M的服务层被称为服务能力层 (SCL)。SCL可以在ETSI M2M架构的各种不同节点中实现。例如,服务层的实例可以在M2M设备 (其中它被称为设备SCL (DSCL))、网关 (其中它被称为网关SCL (GSCL)) 和/或网络节点 (其中它被称为网络SCL (NSCL)) 内实现。oneM2M服务层支持一组公共服务功能 (CSF) (即服务能力)。一个或多个特定类型的CSF的集合的实例化被称为可以被托管在不同类型的网络节点 (例如,基础架构节点、中间节点、专用节点) 上的公共服务实体 (CSE)。第三代合作伙伴计划 (3GPP) 还定义了用于机器类型通信 (MTC) 的架构。在该架构中,服务层及其提供的服务能力被实现为服务能力服务器 (SCS) 的一部分。无论是体现在ETSI M2M架构的DSCL、GSCL、或NSCL中,在3GPP MTC架构的服务能力服务器 (SCS) 中,在oneM2M架构的CSF或CSE中,还是在网络的一些其他节点中,服务层的实例都可以被实现为逻辑实体 (例如,软件、计算机可执行指令等),其或则在网络中的一个或多个独立节点——包括服务器、计算机和其他计算设备或节点——上执行,或则作为一个或多个现有节点的一部分来被执行。作为示例,服务层或其组件的实例可以以在网络节点 (例如,服务器、计算机、网关、设备等) 上运行的软件的形式来实现,该网络节点具有图26C或图26D所图示的一般架构。

[0249] 此外,本申请的逻辑实体——诸如AE 310、CSE 312、NSE 340、GW 226、GW 912、SCS 916和UE 916——可以被实现为M2M网络的一部分,该M2M网络使用面向服务的架构 (SOA)

和/或面向资源架构 (ROA) 来访问本申请的服务。

[0250] 图26C是M2M网络节点30的示例硬件/软件架构的框图,例如eNB 2100、M2M设备18、M2M网关14、M2M服务器等。节点30可以执行或包括诸如AE 310、CSE 312、NSE 340、GW 226、GW 912、SCS 916、和UE 916的逻辑实体。设备30可以是如图26A-B所示的M2M网络的部分或者是非M2M网络的部分。如图26D所示,M2M节点30可以包括处理器32、不可移动存储器44、可移动存储器46、扬声器/麦克风38、小键盘40、显示器、触摸板和/或指示器42、电源48、全球定位系统 (GPS) 芯片组50、以及其他外围设备52。节点30还可以包括通信电路,诸如收发器34和传送/接收元件36。应当理解,M2M节点30可以包括前述元件的任何子组合,同时保持与实施例一致。该节点可以是实现本文本所描述的SMSF功能的节点。

[0251] 处理器32可以是通用处理器、专用处理器、常规处理器、数字信号处理器 (DSP)、多个微处理器、与DSP核心相关联的一个或多个微处理器、控制器、微控制器、专用集成电路 (ASIC)、现场可编程门阵列 (FPGA) 电路、任何其他类型的集成电路 (IC)、状态机等。一般来说,处理器32可执行存储在节点的存储器(例如,存储器44和/或存储器46)中的计算机可执行指令,以便执行该节点的各个所需功能。例如,处理器32可以执行信号编码、数据处理、功率控制、输入/输出处理、和/或使得M2M节点30能够在无线或有线环境中操作的任何其他功能。处理器32可以运行应用层程序(例如,浏览器)和/或无线电接入层 (RAN) 程序和/或其他通信程序。处理器32还可以执行安全操作,诸如认证、安全密钥协定、和/或加密操作,诸如例如在接入层和/或应用层处的操作。

[0252] 如图26C所示,处理器32耦合到其通信电路(例如,收发器34和传送/接收元件36)。通过执行计算机可执行指令处理器32可以控制通信电路以便使得节点30经由其所连接的网络与其他节点通信。具体地,处理器32可以控制通信电路,以便执行本文和权利要求书中描述的传送和接收步骤。尽管图26C将处理器32和收发器34描绘为单独的组件,但是应当理解,处理器32和收发器34可以一起整合在电子封装或芯片中。

[0253] 传送/接收元件36可以被配置为向包括M2M服务器、网关、设备等的其它M2M节点传送信号或从其接收信号。例如,在一个实施例中,传送/接收元件36可以是配置为传送和/或接收RF信号的天线。传送/接收元件36可以支持各种网络和空中接口,诸如WLAN、WPAN、蜂窝等。在一个实施例中,传送/接收元件36可以是配置成传送和/或接收例如IR、UV、或可见光信号的发射器/检测器。在另一个实施例中,传送/接收元件36可以被配置为传送和接收RF和光信号。应当理解,传送/接收元件36可以被配置为传送和/或接收无线或有线信号的任何组合。

[0254] 另外,虽然传送/接收元件36在图20D中被描绘为单个元件,但是M2M节点30可以包括任何数目的传送/接收元件36。更具体地,M2M节点30可以采用MIMO技术。因此,在一个实施例中,M2M节点30可以包括用于传送和接收无线信号的两个或更多个传送/接收元件36(例如,多个天线)。

[0255] 收发器34可以被配置为调制要由传送/接收元件36传送的信号,并解调由传送/接收元件36接收的信号。如上所述,M2M节点30可以具有多模能力。因此,收发器34可以包括多个收发器,以用于使得M2M节点30能够经由多个RAT——诸如例如UTRA和IEEE802.11——进行通信。

[0256] 处理器32可以从诸如不可移动存储器44和/或可移动存储器46的任何类型的合适

的存储器访问信息并将数据存储在其中。例如,处理器32可以在其存储器中存储会话上下文,如上所述。不可移动存储器44可以包括随机存取存储器(RAM)、只读存储器(ROM)、硬盘、或任何其他类型的存储器存储设备。可移动存储器46可以包括订户身份模块(SIM)卡、记忆棒、安全数字(SD)存储卡等。在其他实施例中,处理器32可以从没有在物理上位于M2M节点30上——诸如在服务器或家庭计算机上——的存储器访问信息,并且将数据存储在其中。处理器32可以被配置为控制显示器或指示器42上的照明模式、图像、或颜色以反映M2M服务层会话迁移或共享的状态,或者从用户获得输入或向用户显示关于节点的会话迁移或共享能力或设置的信息。在另一示例中,显示器可以示出关于会话状态的信息。当前公开定义了oneM2M实施例中的RESTful用户/应用API。可以在显示器上示出的图形用户界面可以被分层在API的顶部,以允许用户经由本文描述的底层服务层会话功能交互地建立和管理E2E会话或其迁移或共享。

[0257] 处理器32可以从电源48接收电力,并且可以被配置为分发和/或控制M2M节点30中的其他组件的电力。电源48可以是用于为M2M节点30供电的任何合适的设备。例如,电源48可以包括一个或多个干电池(例如,镍镉(NiCd)、镍锌(NiZn)、镍金属氢化物(NiMH)、锂离子(Li离子)等),太阳能电池,燃料电池等。

[0258] 处理器32还可以被耦合到GPS芯片组50,该GPS芯片组50被配置为提供关于M2M节点30的当前位置的位置信息(例如,经度和纬度)。应当理解,M2M节点30可以在保持与实施例一致的同时通过任何合适的位置确定方法获取位置信息。

[0259] 处理器32还可以被耦合到其它外围设备52,其可以包括提供附加特征、功能和/或有线或无线连接的一个或多个软件和/或硬件模块。例如,外围设备52可以包括加速度计、电子罗盘、卫星收发器、传感器、数字照相机(用于照片或视频)、通用串行总线(USB)端口、振动设备、电视收发器、免提耳机、蓝牙®模块、调频(FM)无线电单元、数字音乐播放器、媒体播放器、视频游戏播放器模块、互联网浏览器等。

[0260] 图26D是示例性计算系统90的框图,其还可以被用于实现诸如M2M服务器、网关、设备、或其它节点的M2M网络的一个或多个节点。计算系统90可以包括计算机或服务器,并且可以主要由计算机可读指令来控制,计算机可读指令可以处于软件形式,而无论这样的软件在何处或者以何种方式来被存储或访问。计算系统90可以执行或包括逻辑实体,诸如AE 310、CSE 312、NSE 340、GW 226、GW 912、SCS 916、和UE 916。计算系统90可以是M2M设备、用户设备、网关、UE/GW或任何其他节点,包括例如移动护理网络、服务层网络应用提供者、终端设备18或M2M网关设备14的节点。这样的计算机可读指令可以在诸如中央处理单元(CPU) 91的处理器内执行,以使得计算系统90进行工作。在许多已知的工作站、服务器、和个人计算机中,中央处理单元91由称为微处理器的单芯片CPU实现。在其他机器中,中央处理单元91可以包括多个处理器。协处理器81是与主CPU不同的可选的处理器91,其执行附加功能或辅助CPU91。CPU91和/或协处理器81可以接收、生成、和处理与所公开的用于E2E M2M服务层会话的系统和方法相关的数据,诸如接收会话凭证或基于会话凭证来认证。

[0261] 在操作中,CPU 91抓取、解码、和执行指令,并经由计算机的主数据传输路径——系统总线80——向其它资源传送信息和从其他资源传送信息。这样的系统总线连接计算系统90中的组件,并且定义介质以供数据交换。系统总线80通常包括用于发送数据的数据线、用于发送地址的地址线,以及用于发送中断和用于操作系统总线的控制线。这样的系统总

线80的示例是PCI(外围组件互连)总线。

[0262] 耦合到系统总线80的存储器包括随机存取存储器(RAM)82和只读存储器(ROM)93。这样的存储器包括允许存储和检索信息的电路。ROM 93通常包含不易修改的所存储数据。存储在RAM 82中的数据可以由CPU 91或其他硬件设备读取或改变。对RAM 82和/或ROM 93的访问可以由存储器控制器92控制。存储器控制器92可以提供地址转换功能,其在执行指令时将虚拟地址转换为物理地址。存储器控制器92还可以提供存储器保护功能,其隔离系统内的进程并将系统进程与用户进程隔离。因此,以第一模式运行的程序可以仅访问由其本身的进程虚拟地址空间映射的存储器;它无法访问另一进程的虚拟地址空间内的存储器,除非已经建立了进程之间的存储器共享。

[0263] 另外,计算系统90可以包含外围设备控制器83,外围设备控制器83负责将指令从CPU 91传输至外围设备,诸如打印机94、键盘84、鼠标95、和盘驱动器85。

[0264] 由显示控制器96控制的显示器86用于显示由计算系统90生成的视觉输出。这样的视觉输出可以包括文本、图形、动画图形、和视频。显示器86可以利用基于CRT的视频显示器、基于LCD的平板显示器、基于气体等离子体的平板显示器、或触摸面板来实现。显示控制器96包括生成向显示器86发送的视频信号所需的电子组件。

[0265] 此外,计算系统90可以包含通信电路,诸如例如网络适配器97,其可以被用于将计算系统90连接到外部通信网络,诸如图26A和图26B的网络12,以使得计算系统90能够与网络的其他节点通信。

[0266] 应当理解,本申请所描述的系统、方法和过程中的任何一个或全部可以以存储在计算机可读存储介质上的计算机可执行指令(即,程序代码)的形式体现,所述指令当由诸如包括例如M2M服务器、网关、设备等的M2M网络的节点的机器执行时,执行和/或实现本文所描述的系统、方法、和过程。具体地,上述的任何步骤、操作或功能中的任何一个——包括网关、UE、UE/GW、或者移动核心网的任何节点、服务层或网络应用提供者的操作——可以以这样的计算机可执行指令的形式实现。诸如AE 310、CSE 312、NSE 340、GW 226、GW 912、SCS 916、和UE 916的逻辑实体可以以存储在计算机可读存储介质上的计算机可执行指令的形式体现。计算机可读存储介质包括以用于存储信息的任何非易失性(即,有形或物理)方法或技术实现的易失性和非易失性、可移动和不可移动介质,但是这样的计算机可读存储介质不包括信号。计算机可读存储介质包括但不限于RAM、ROM、EEPROM、闪存或其他存储器技术,CD-ROM、数字多功能盘(DVD)或其他光盘存储器,盒式磁带、磁带、磁盘存储或其它磁性存储设备,或可被用于存储所需信息并可由计算机访问的任何其它有形或物理介质。

[0267] 因此,申请人已经公开了用于M2M/IoT服务的发布和发现的示例系统和方法。在示例系统和方法中,网关系统从M2M/IoT设备接收资源描述。网关系统为每个接收到的资源描述创建记录关于该资源的信息的个体web服务描述语言文件(WSDL-I文件)。网关识别具有类似特性的服务群组,并为每个所识别的群组生成记录关于该群组的信息的web服务描述文件(WSDL-G文件)。WSDL-G文件被传输至服务注册表基础架构(SRI)。消费者系统查询满足特定准则的服务的SRI,并接收满足该准则的服务群组的WSDL-G文件。消费者系统从网关请求与所选择的WSDL-G文件相对应的WSDL-I文件。消费者系统使用所接收的WSDL-I文件来选择特定服务,并且使用WSDL-I文件中用于所选择的的服务的信息来从对应的M2M/IoT设备请求服务。

[0268] 应当理解,虽然已经公开了说明性实施例,但是潜在实施例的范围不限于明确阐述的那些。例如,虽然已经主要参照M2M网关和SRI来描述了系统,但是可以另外地或替选地使用其他计算布置。设想的实施例使用特定架构或技术扩展到实施方式之外。例如,潜在实施例可以采用除WSDL之外的计算机可读描述性语言。潜在的实施方式扩展到所有类型的服务层架构、系统、和实施例。

[0269] 应当理解,本文所描述的各个技术可以结合硬件或软件或在适当时利用两者的组合来实现。因此,本文描述的主题的方法和装置或其某些方面或部分可以采取体现在有形介质中的程序代码(即指令)的形式,所述有形介质诸如软盘、CD-ROM、硬盘驱动器、或任何其他机器可读存储介质,其中当程序代码被加载到诸如计算机的机器中并由机器执行时,该机器变成用于实践本文所述主题的装置。在程序代码被存储在介质上的情况下,可能的情况是,所讨论的程序代码被存储在共同执行所讨论的动作的一个或多个介质上,也就是说被一起采取的一个或多个介质包含代码以执行动作,但是-在存在多于一个单一介质的情况下-不要求代码的任何特定部分被存储在任意特定介质上。在可编程计算机上执行程序代码的情况下,计算设备通常包括处理器、处理器可读的存储介质(包括易失性和非易失性存储器和/或存储元件)、至少一个输入设备、以及至少一个输出设备。可以实现或利用结合本文所述的主题来描述的过程的一个或多个程序,例如通过使用API、可重用控件等。这样的程序优选地以高级面向过程或面向对象的编程语言实现来与计算机系统通信。但是,如果需要,程序可以以汇编或机器语言实现。在任何情况下,语言可以是编译或解释语言,并且与硬件实现结合。

[0270] 虽然示例实施例可以指涉在一个或多个独立计算机系统或设备的场境中利用本文描述的主题的方面,但是本文描述的主题不限于此,而是可以结合任何计算环境——诸如网络或分布式计算环境——来实现。此外,本申请描述的主题的方面可以在多个处理芯片或设备中或跨多个处理芯片或设备来实现,并且存储可以类似地跨多个设备被影响。这样的设备可以包括个人计算机、网络服务器、手持设备、超级计算机、或被整合到诸如汽车和飞机的其他系统中的计算机。

[0271] 以下是可能出现在上述描述中的与服务级别技术有关的缩略语的列表。

[0272]	AE	应用实体
[0273]	CoAP	受限应用协议
[0274]	CoRE	受限RESTful环境
[0275]	CSE	公共服务实体
[0276]	CSF	公共服务功能
[0277]	GW	网关
[0278]	HTTP	超文本传输协议
[0279]	IETF	互联网工程任务组
[0280]	IoT	物联网
[0281]	JSON	JavaScript对象表示法
[0282]	M2M	机器对机器
[0283]	MEP	消息交换模式
[0284]	OWL-S	Web服务的Web本体语言

[0285]	RD	资源目录
[0286]	REST	表述性状态转移
[0287]	ROA	面向资源的架构
[0288]	RPC	远程过程调用
[0289]	RSDL	RESTful服务描述语言
[0290]	SOA	面向服务的架构
[0291]	SOAP	简单对象访问协议
[0292]	SRI	服务注册基础架构
[0293]	UDDI	通用描述发现和集成
[0294]	WADL	Web应用描述语言
[0295]	WS	Web服务
[0296]	WSDL	Web服务描述语言
[0297]	WSDL-G	相似服务群组的WSDL文件
[0298]	WSDL-I	个体服务的WSDL文件
[0299]	W3C	万维网联盟
[0300]	XML	可扩展标记语言
[0301]	XSD	XML范式定义

[0302] 尽管已经用特定于结构特征和/或方法论动作的语言描述了主题,但是应当理解,所附权利要求书中定义的主题不一定限于上述具体特征或动作。相反,上述具体特征和动作被公开为实现权利要求书的示例形式。

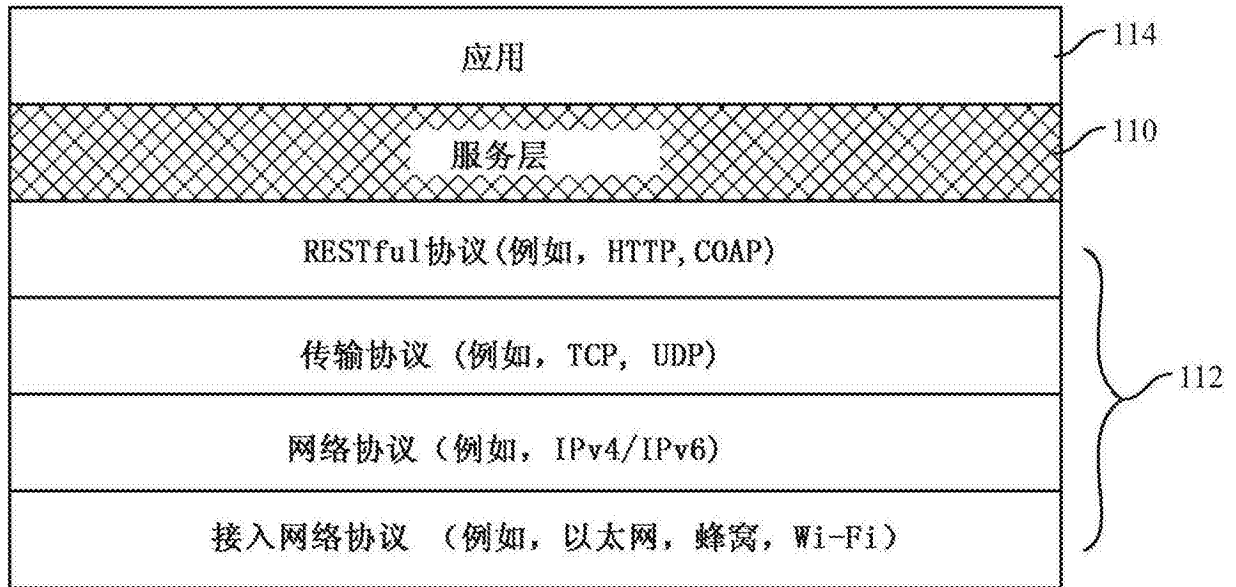


图1

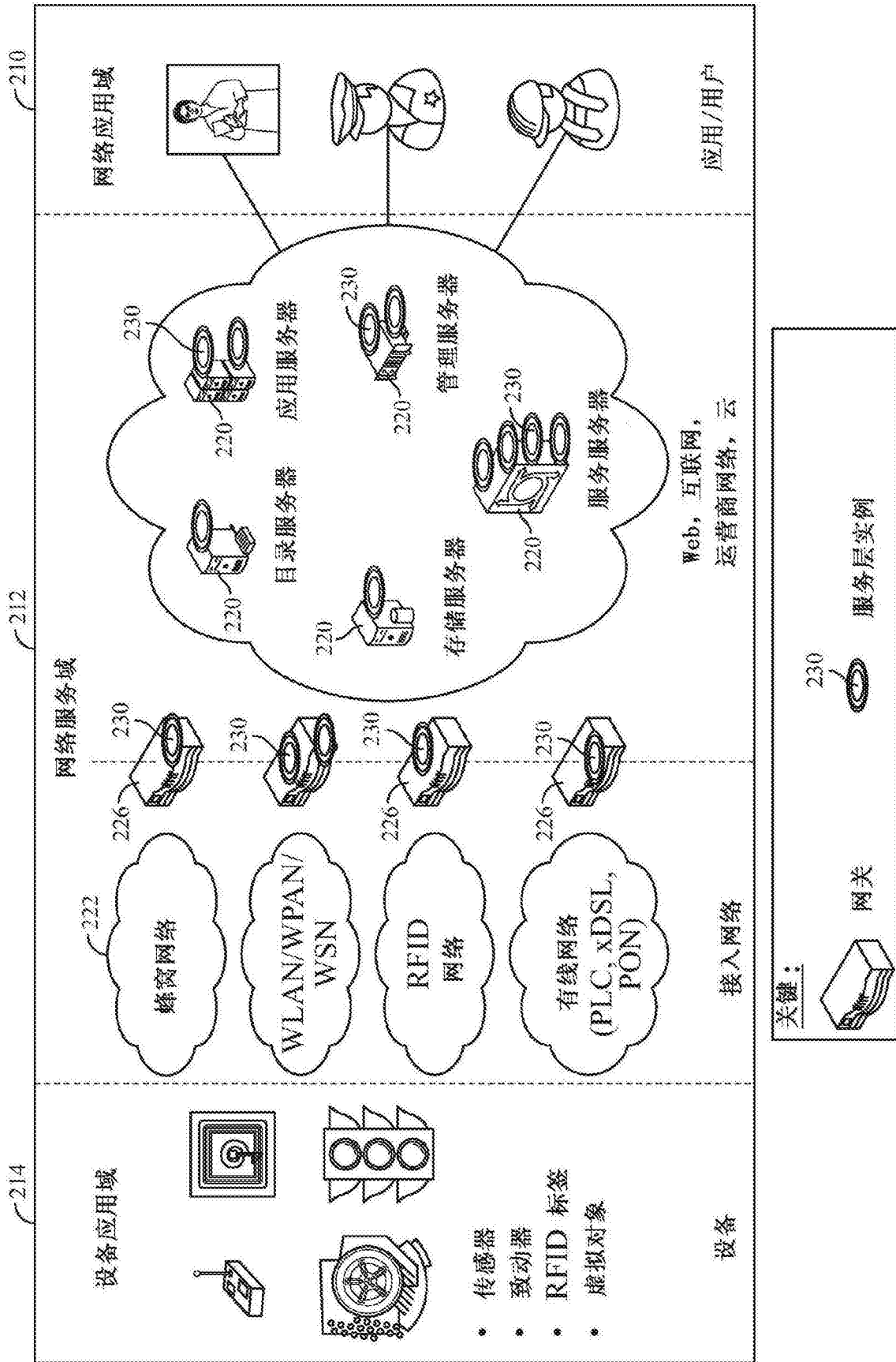


图2

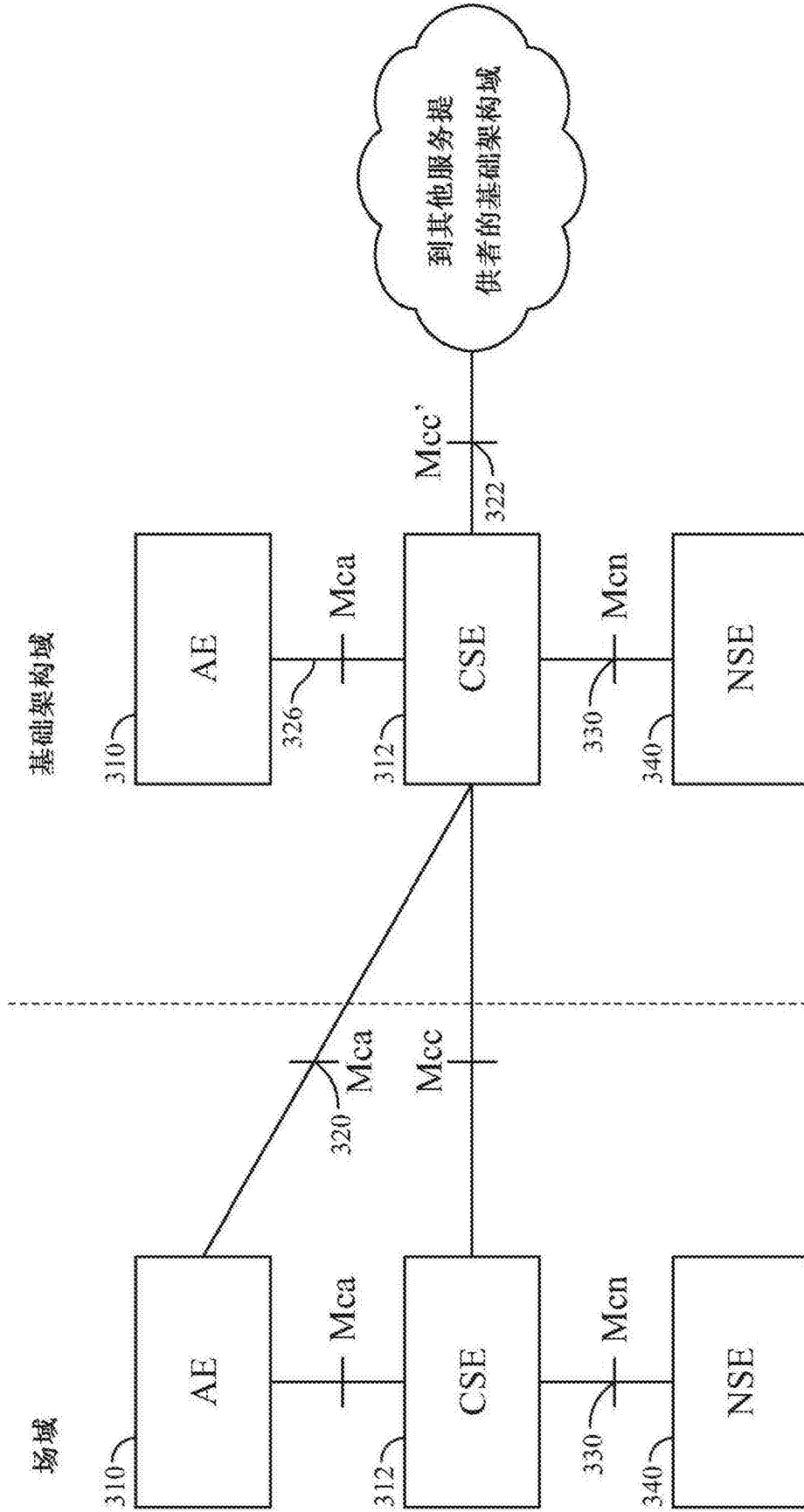


图3

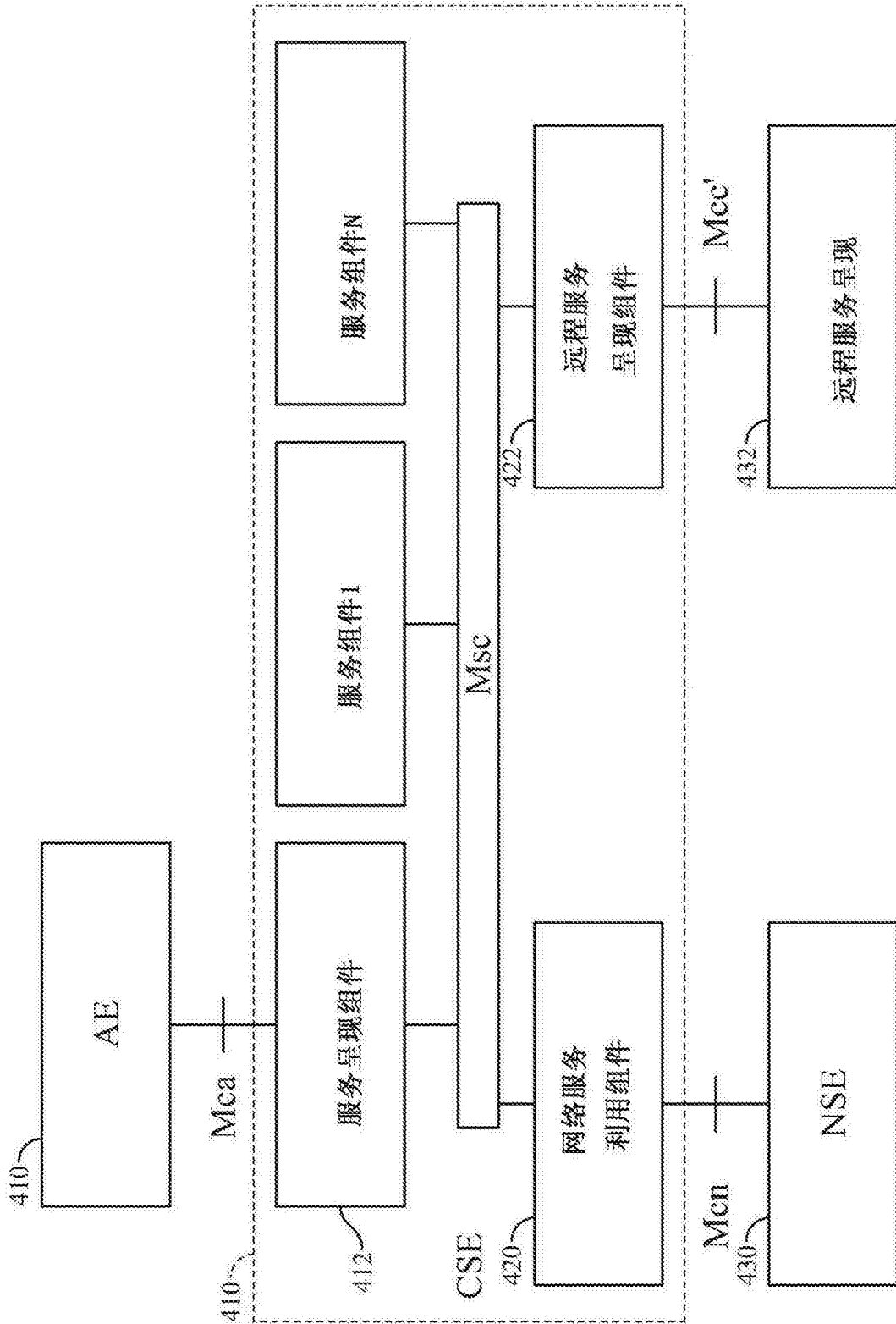


图4

酒店预订
WSDL示例

```

<documentation>
  本文档描述了GreatH Web服务。对于
  使用此服务的附加应用层级的要求—
  超出WSDL2.0能够描述的范围—in
  http://greatH.example.com/2004/reservation-documentation.html 可用
</documentation>

```

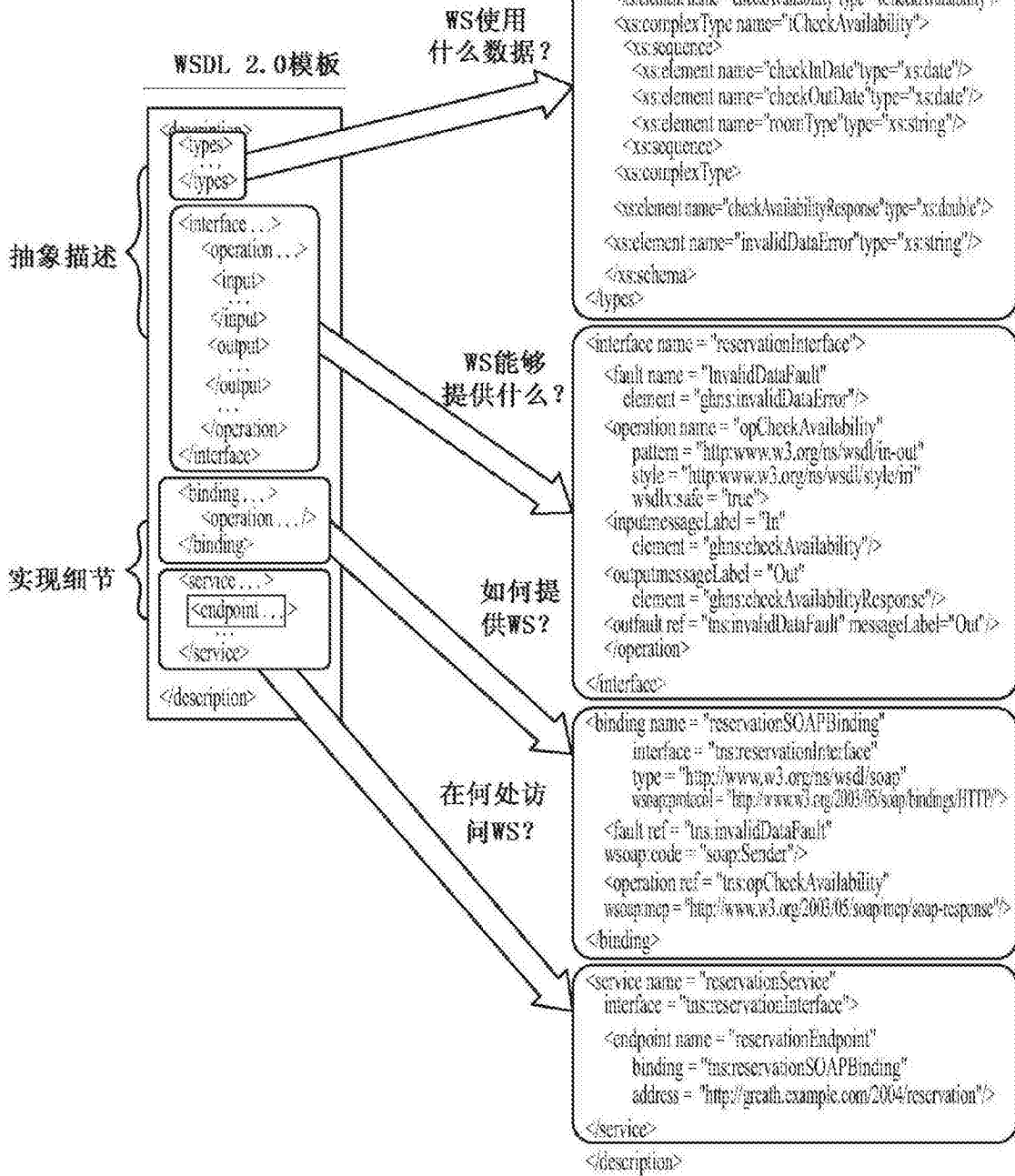


图5

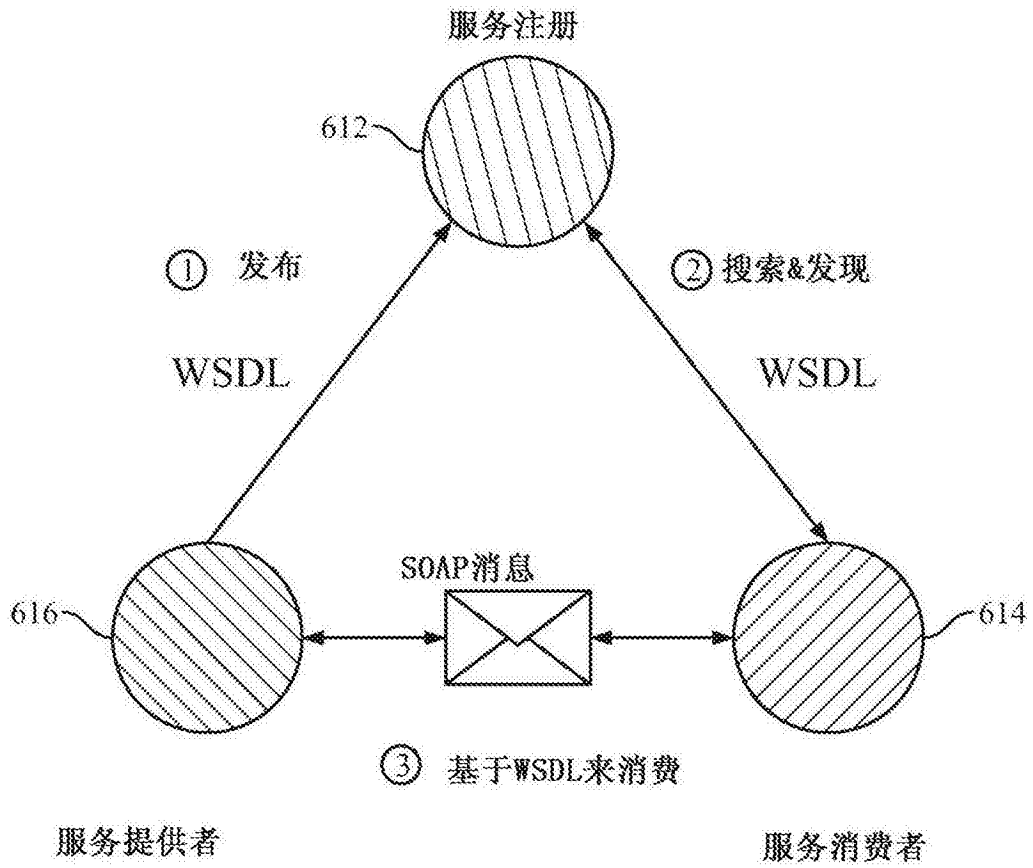


图6

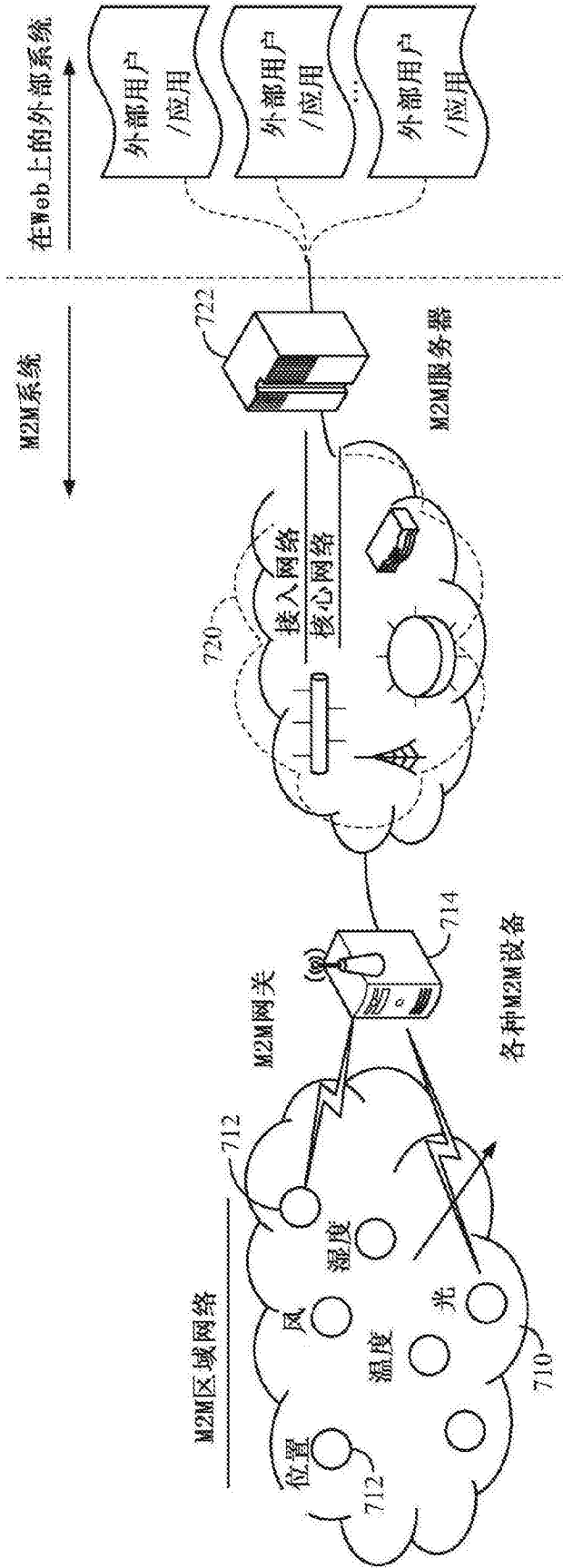


图7

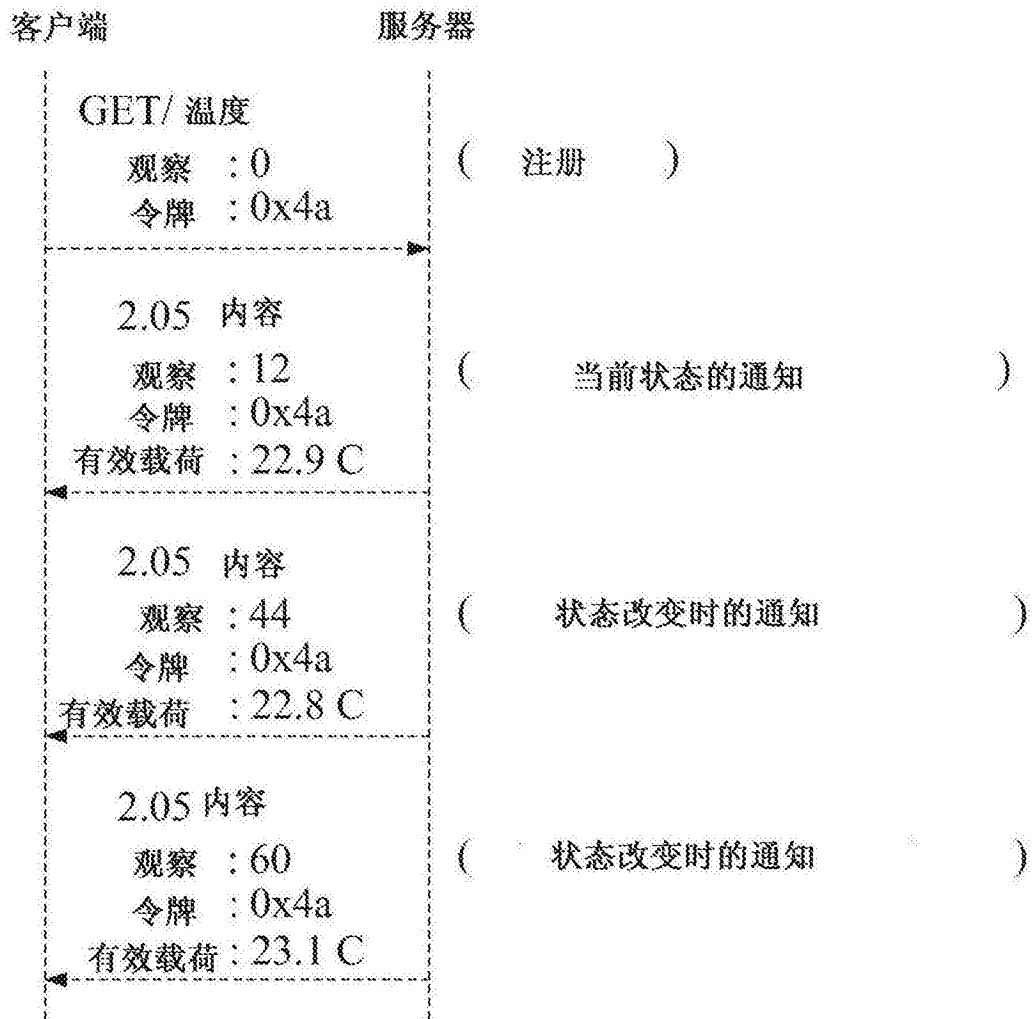


图8

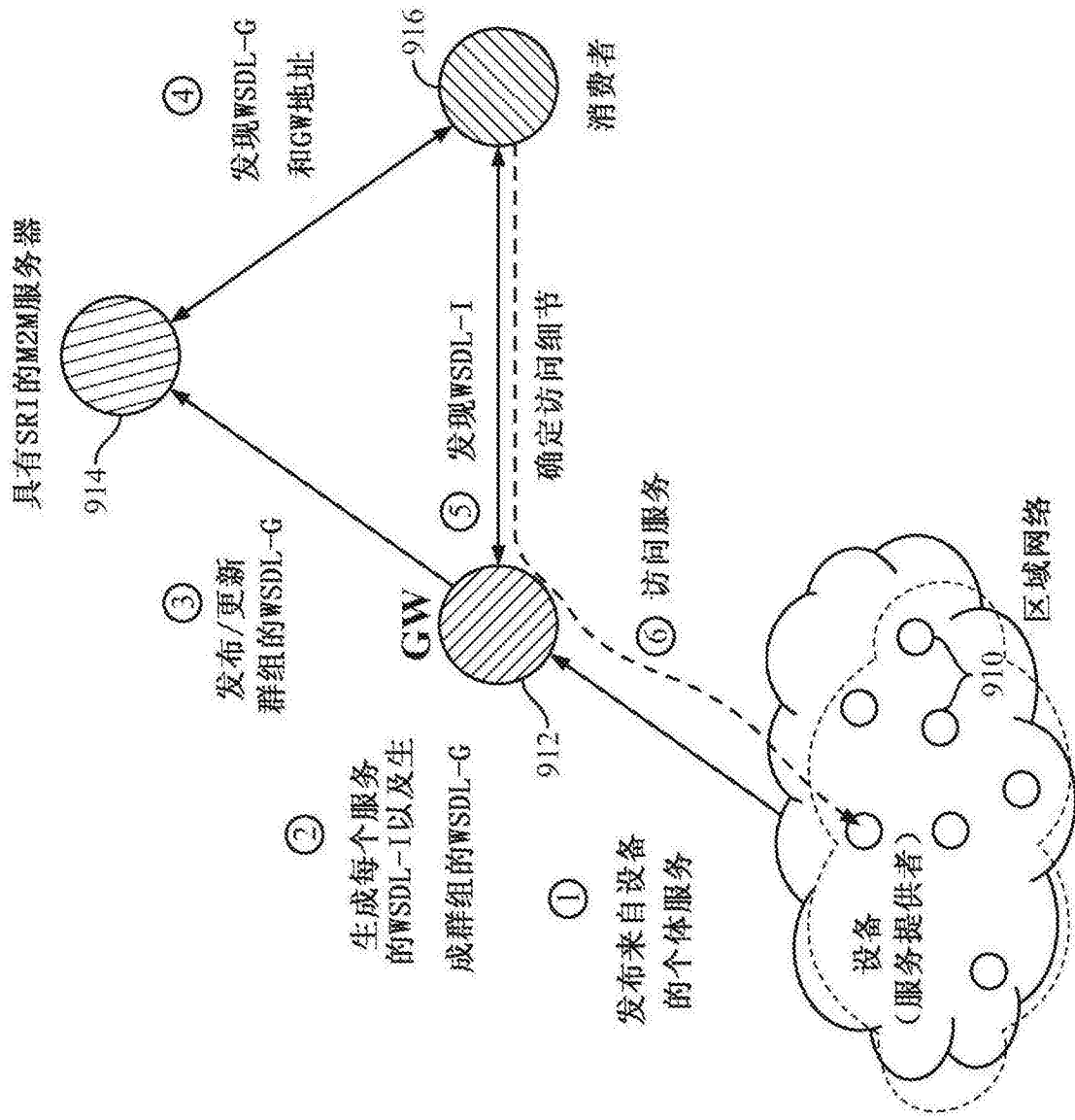


图9

WSDL-I Template Configuration Panel

Please select which data items you would like to include in the WSDL-I template (only optional items are listed):

<availability.../>

<geoLocation.../>

<virtualPrice.../>

Confirm

图10A

WSDL-G Template Configuration Panel

Please select which data items you would like to include in the WSDL-G template (only optional items are listed):

<group QoS.../>

<GW PoC.../>

Confirm

图10B

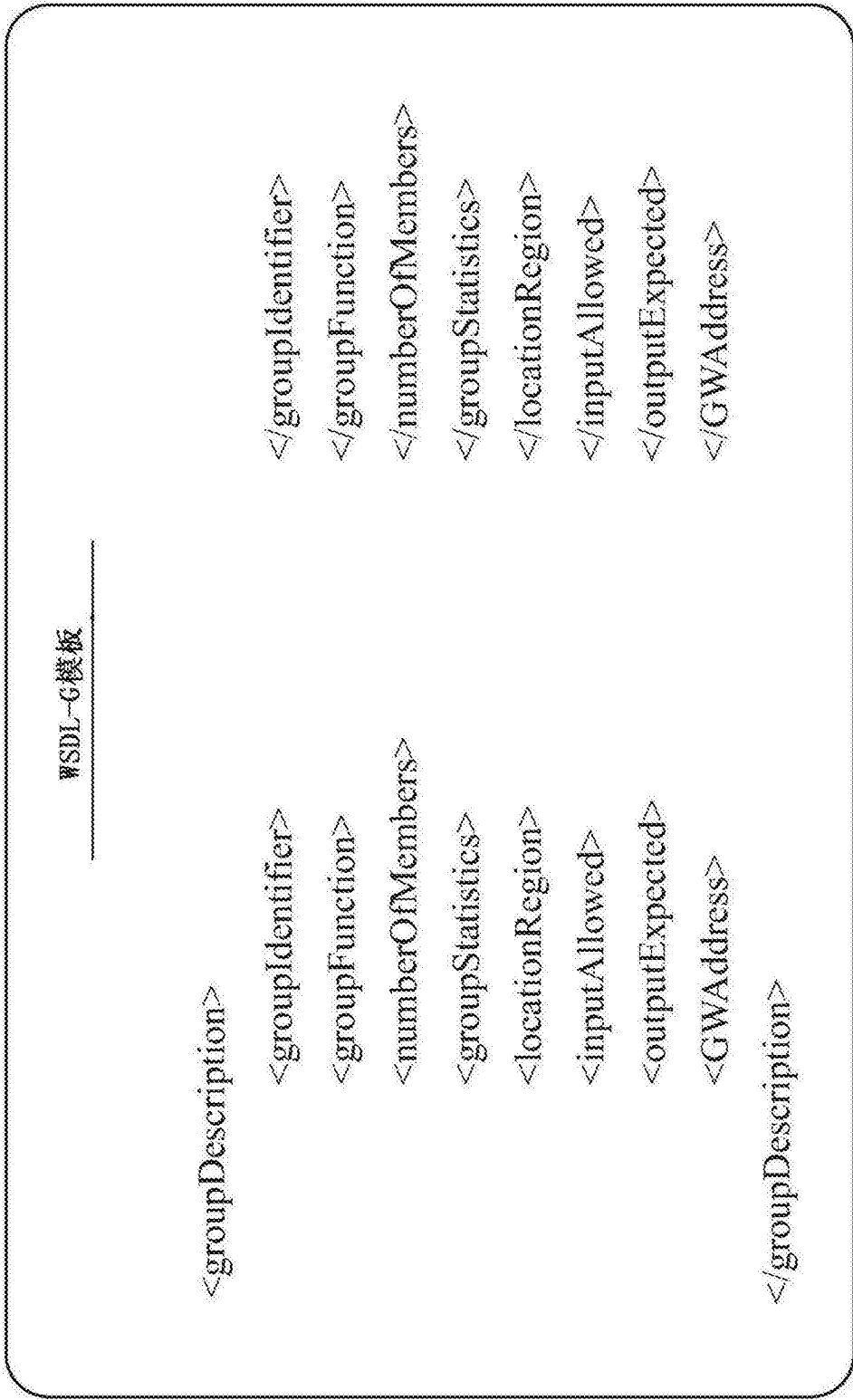


图11

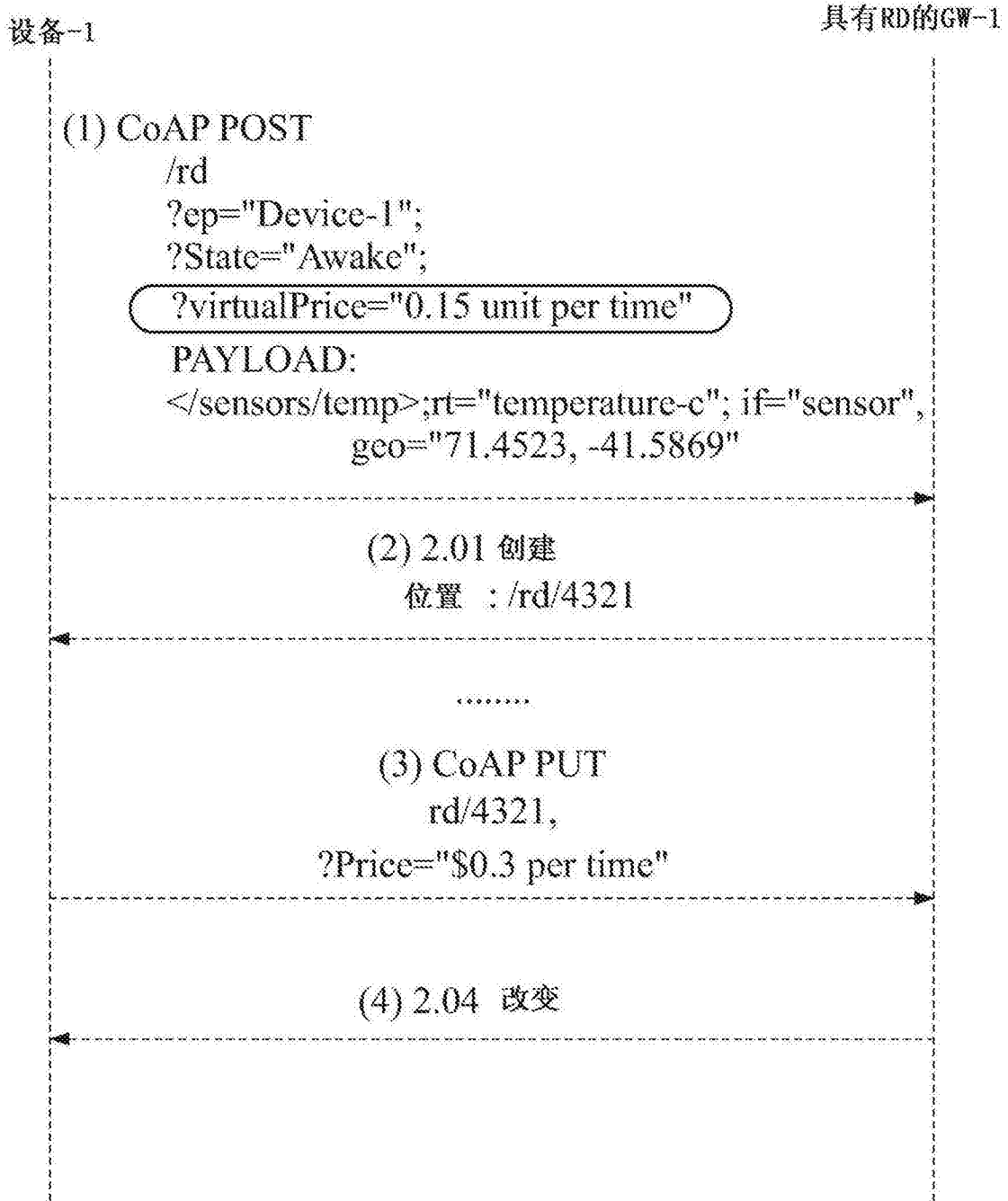


图12

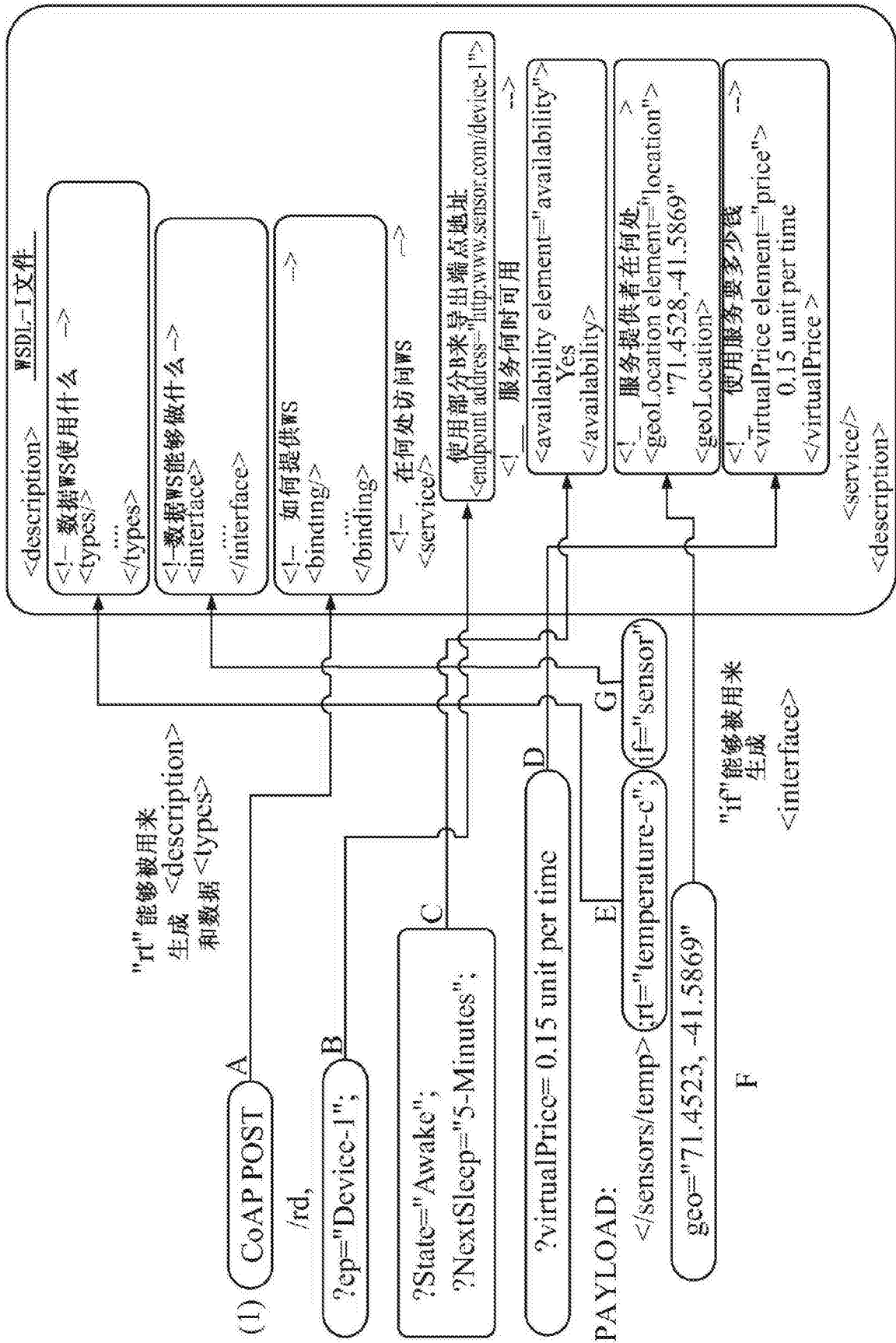


图13

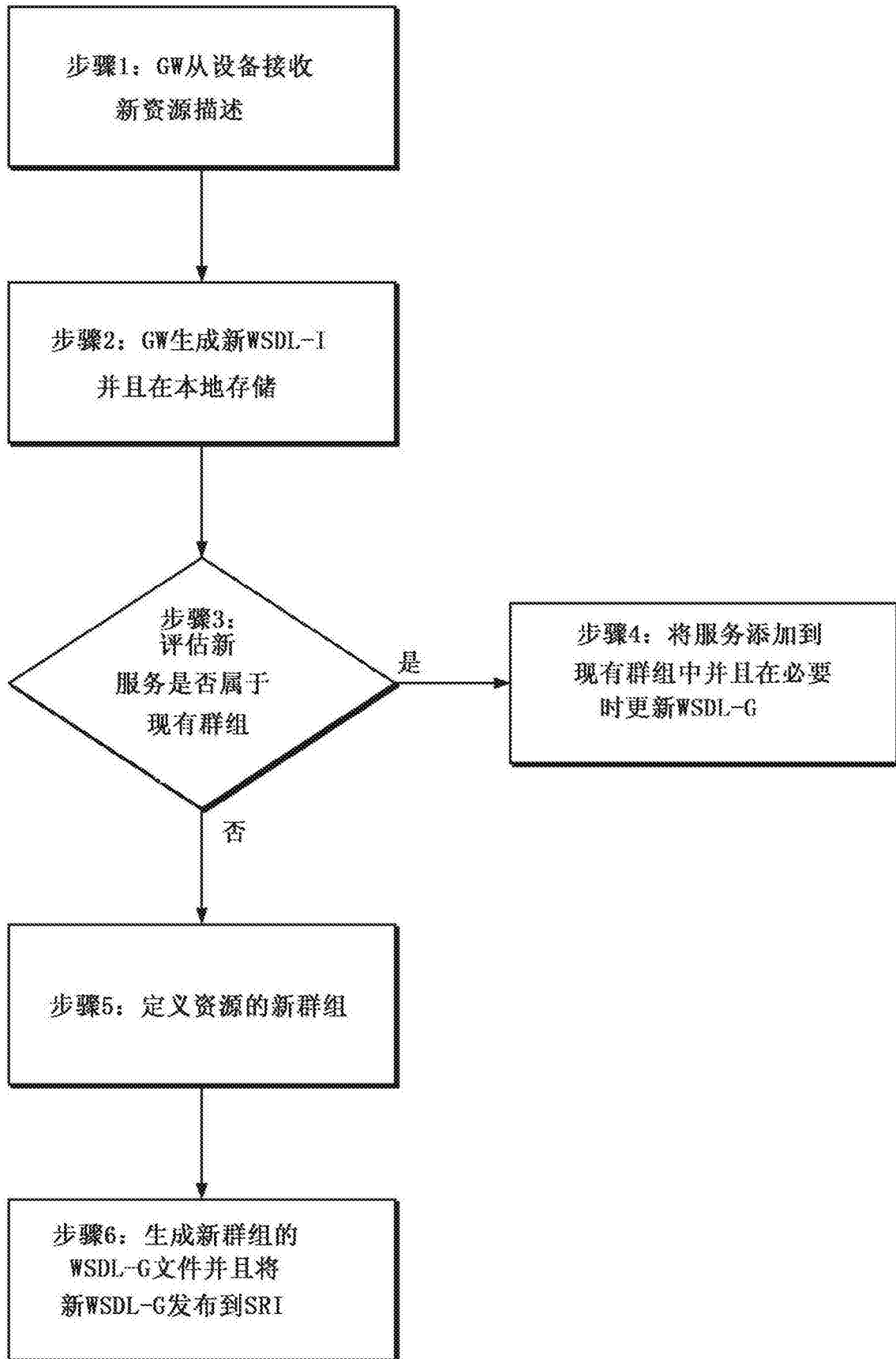


图14

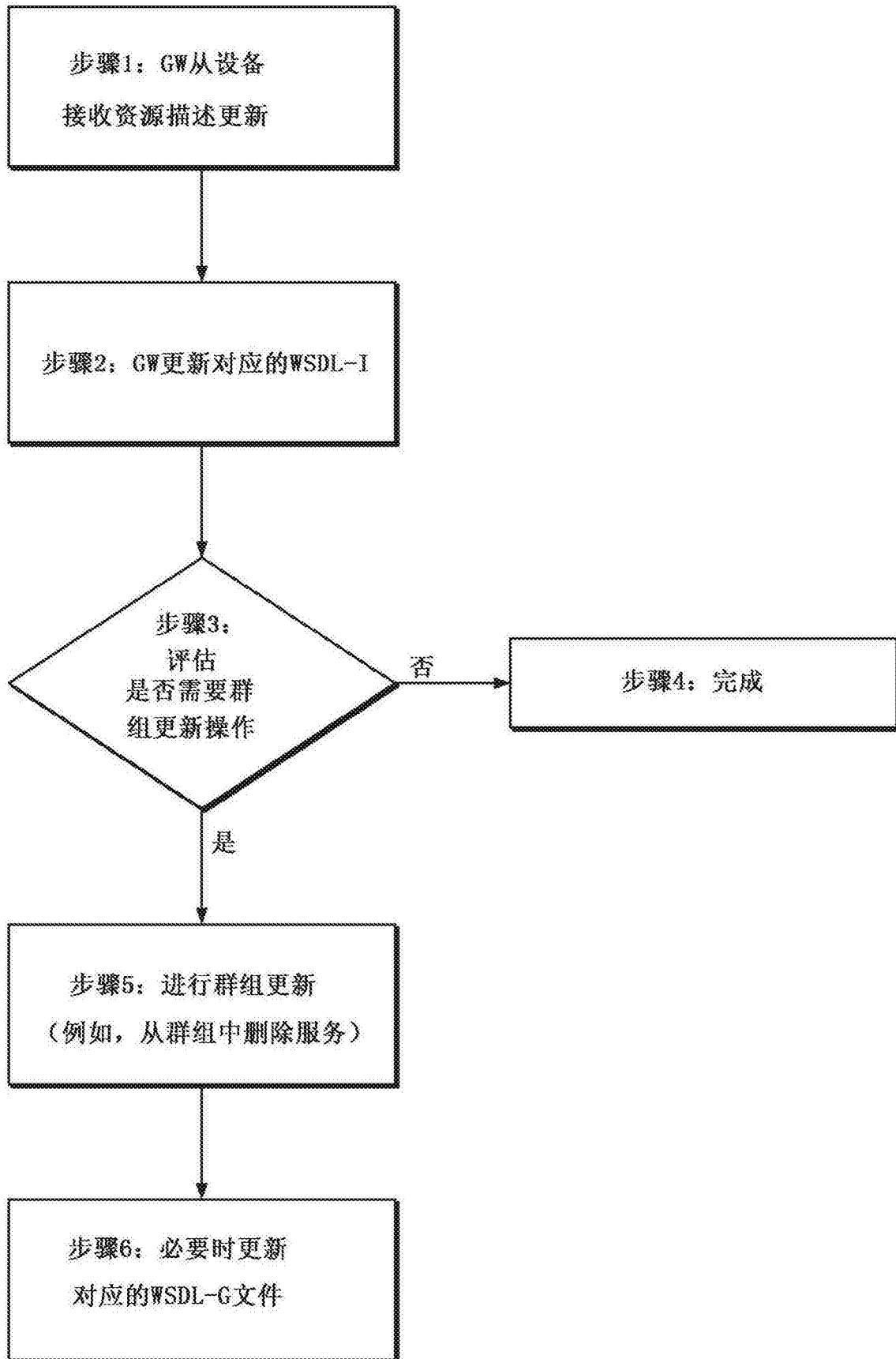


图15

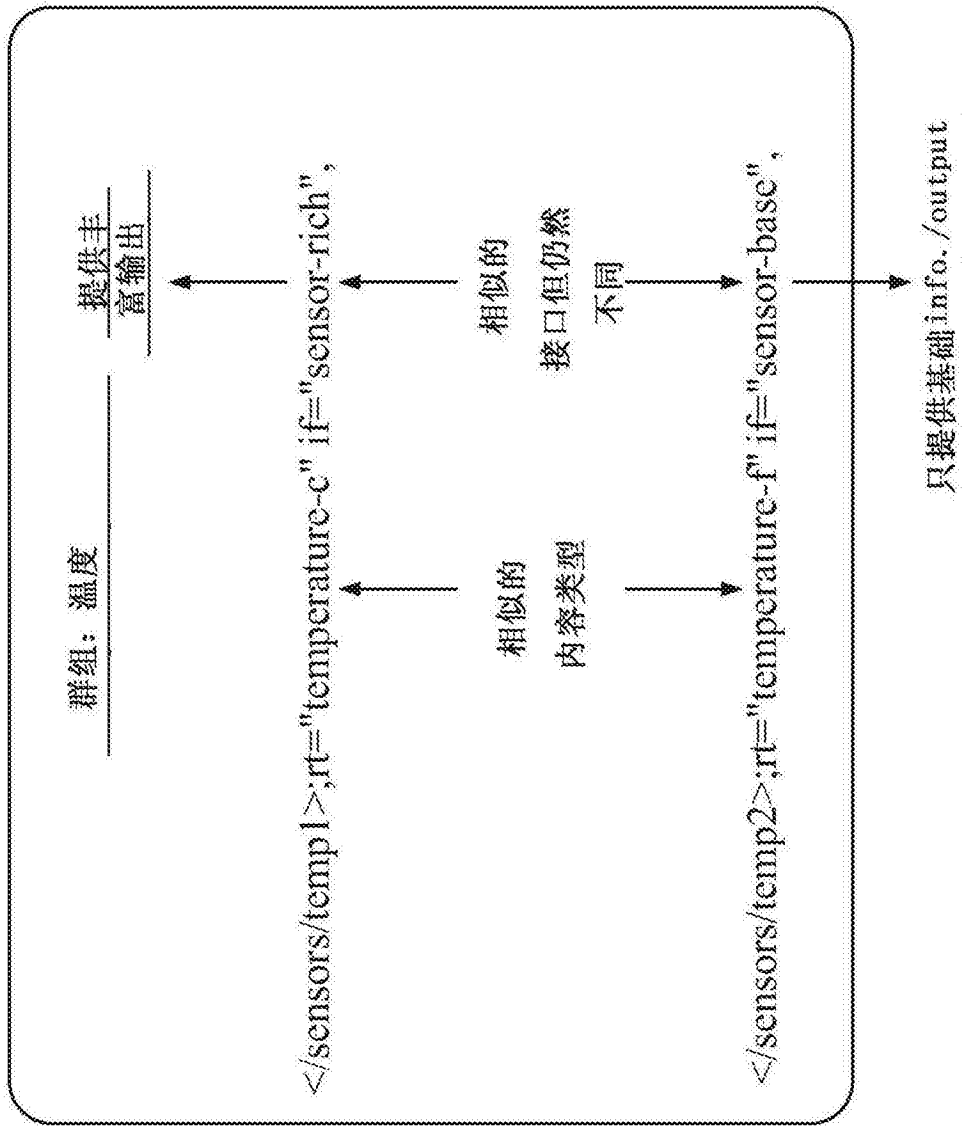


图16

温度的WSDL-G

```
<groupDescription>  
  <groupIdentifier> TEM0909011 </groupIdentifier>  
  <groupFunction> Temperature </groupFunction>  
  <numberOfMembers> >20 </numberOfMembers>  
  <groupStatistics> 90% successful </groupStatistics>  
  <locationRegion> Amherst Town </locationRegion>  
  <inputAllowed> ZipCode, Coordinates </inputAllowed>  
  <outputExpected> temp-c, temp-f </outputExpected>  
  <gwAddress> 58.197.46.5:12345 </gwAddress>  
</groupDescription>
```

图17

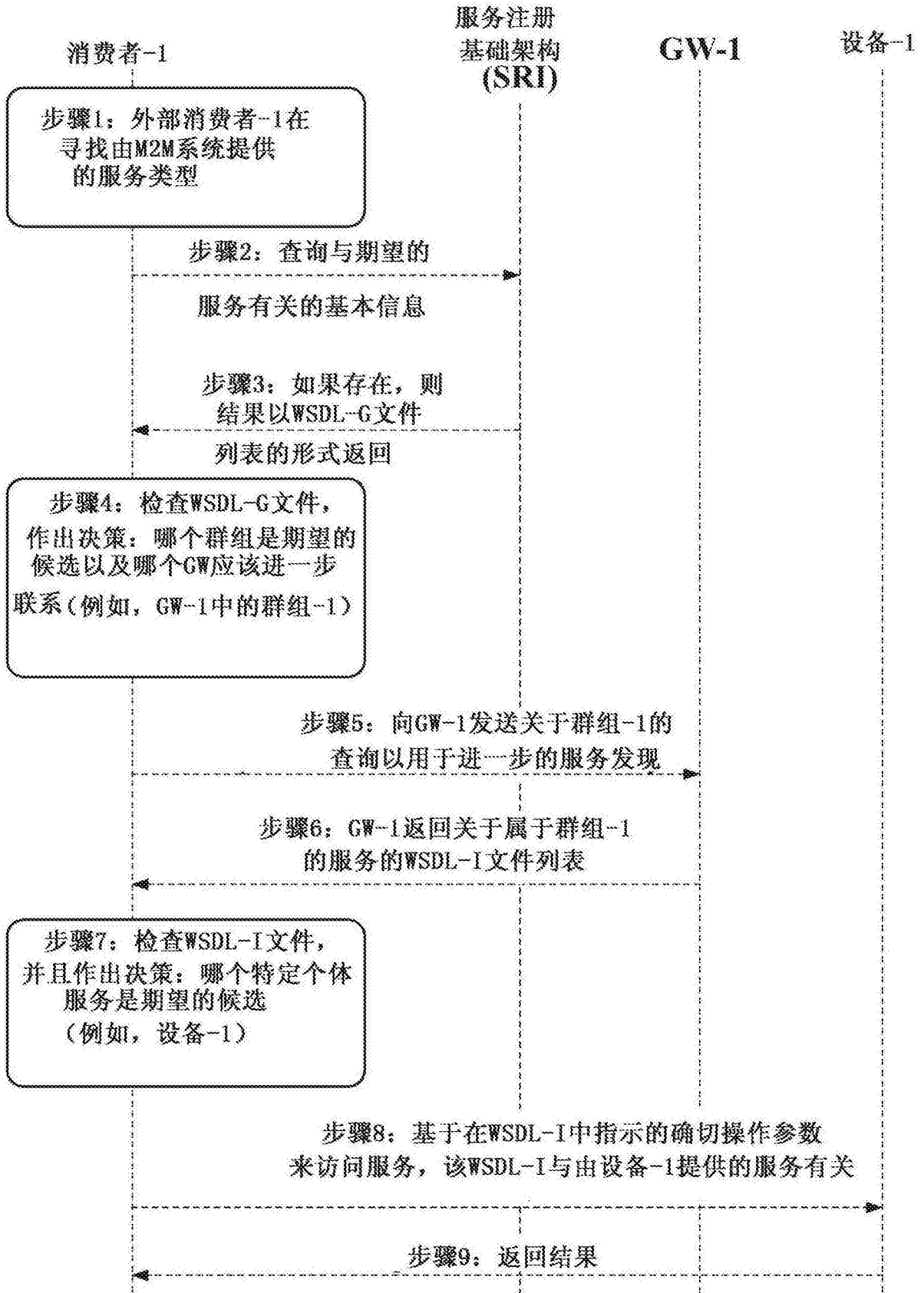


图18

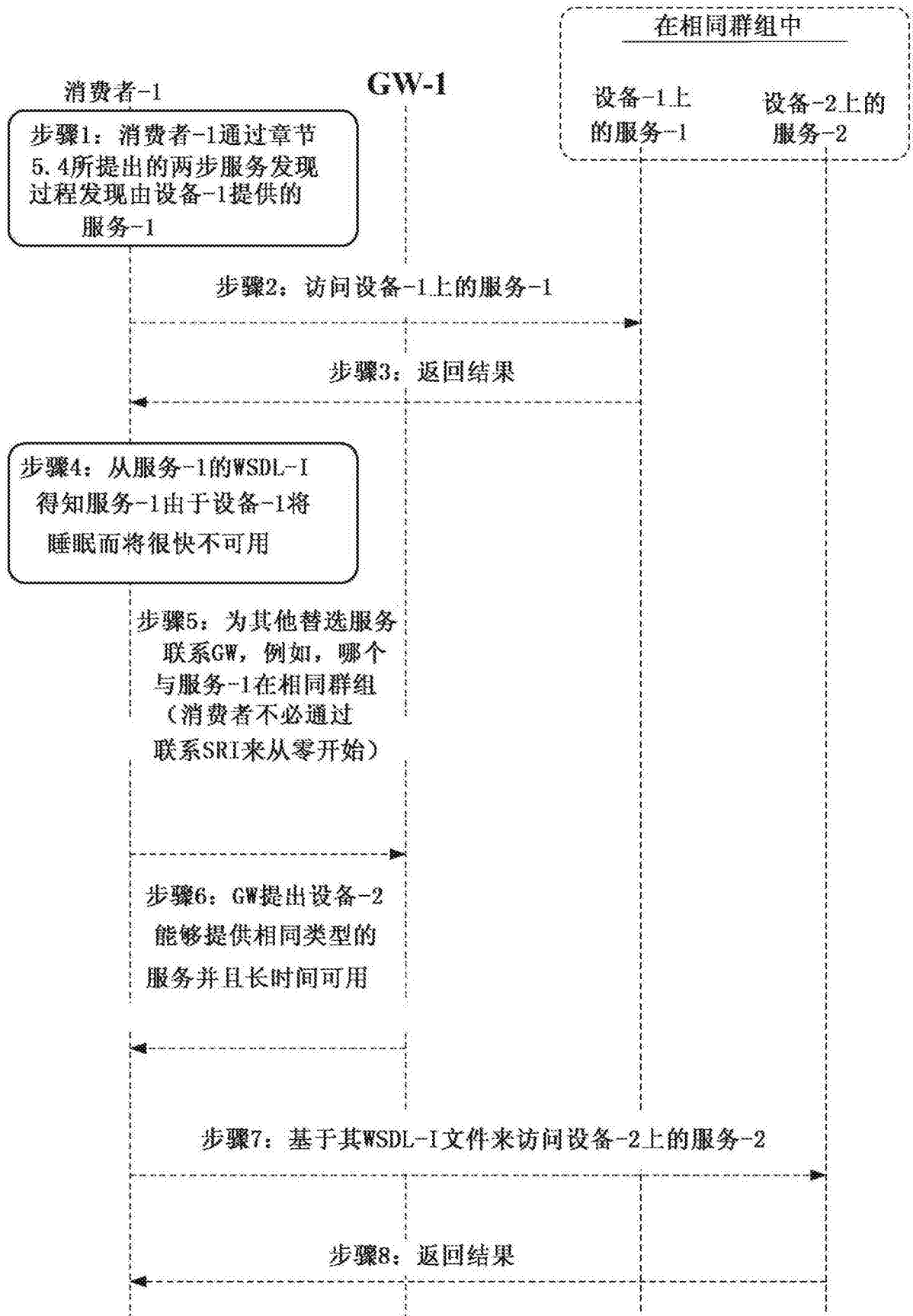


图19

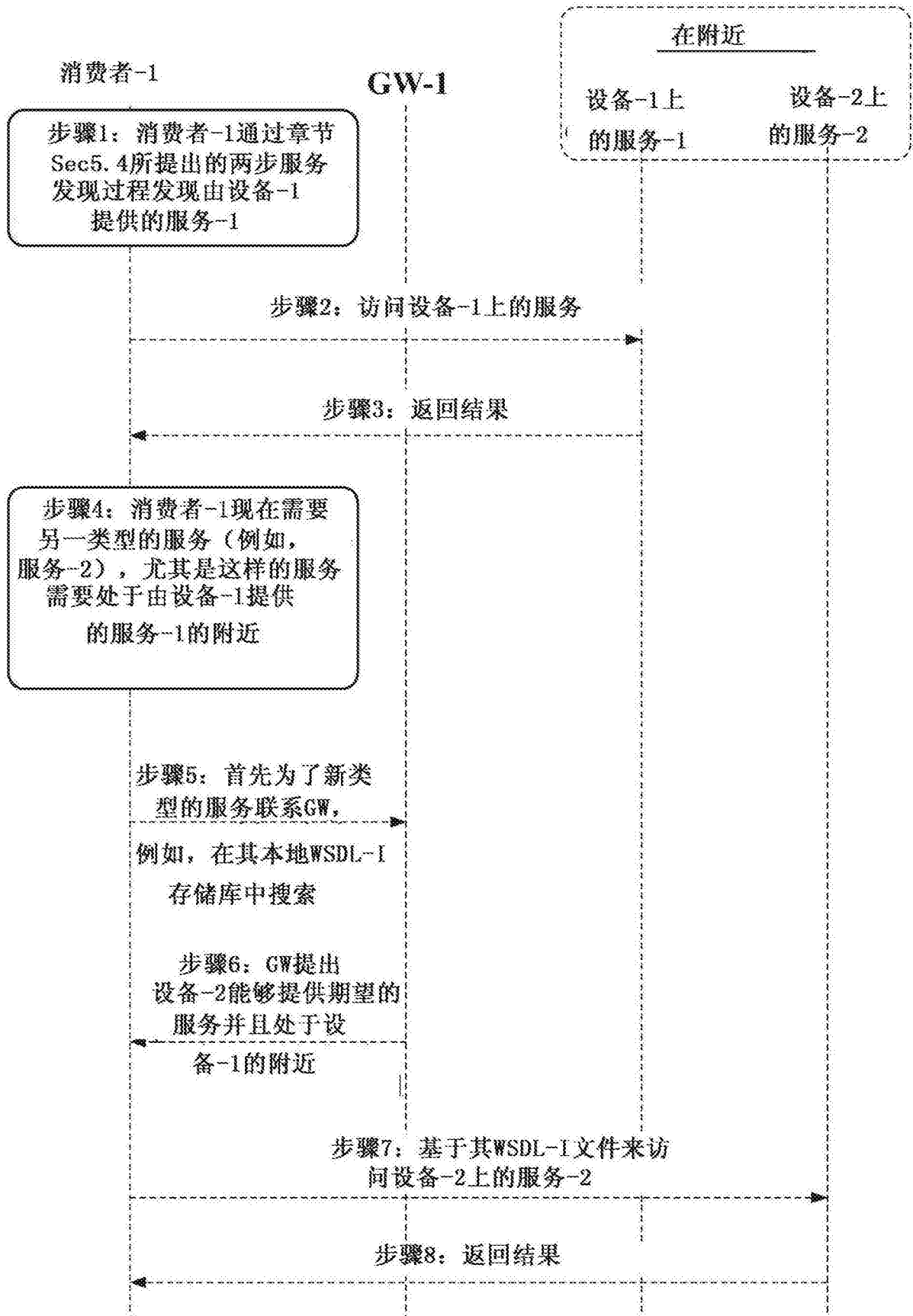


图20

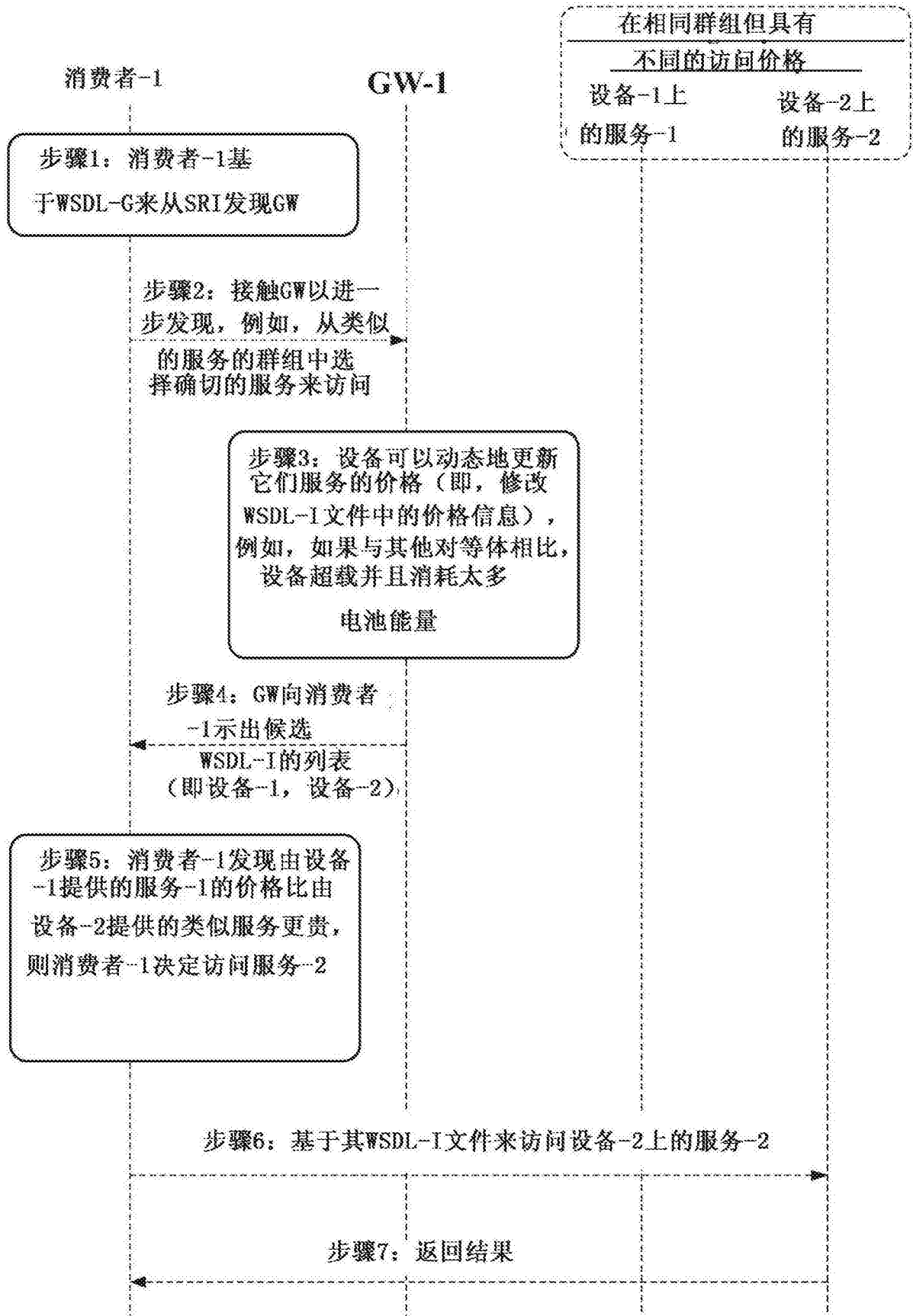


图21

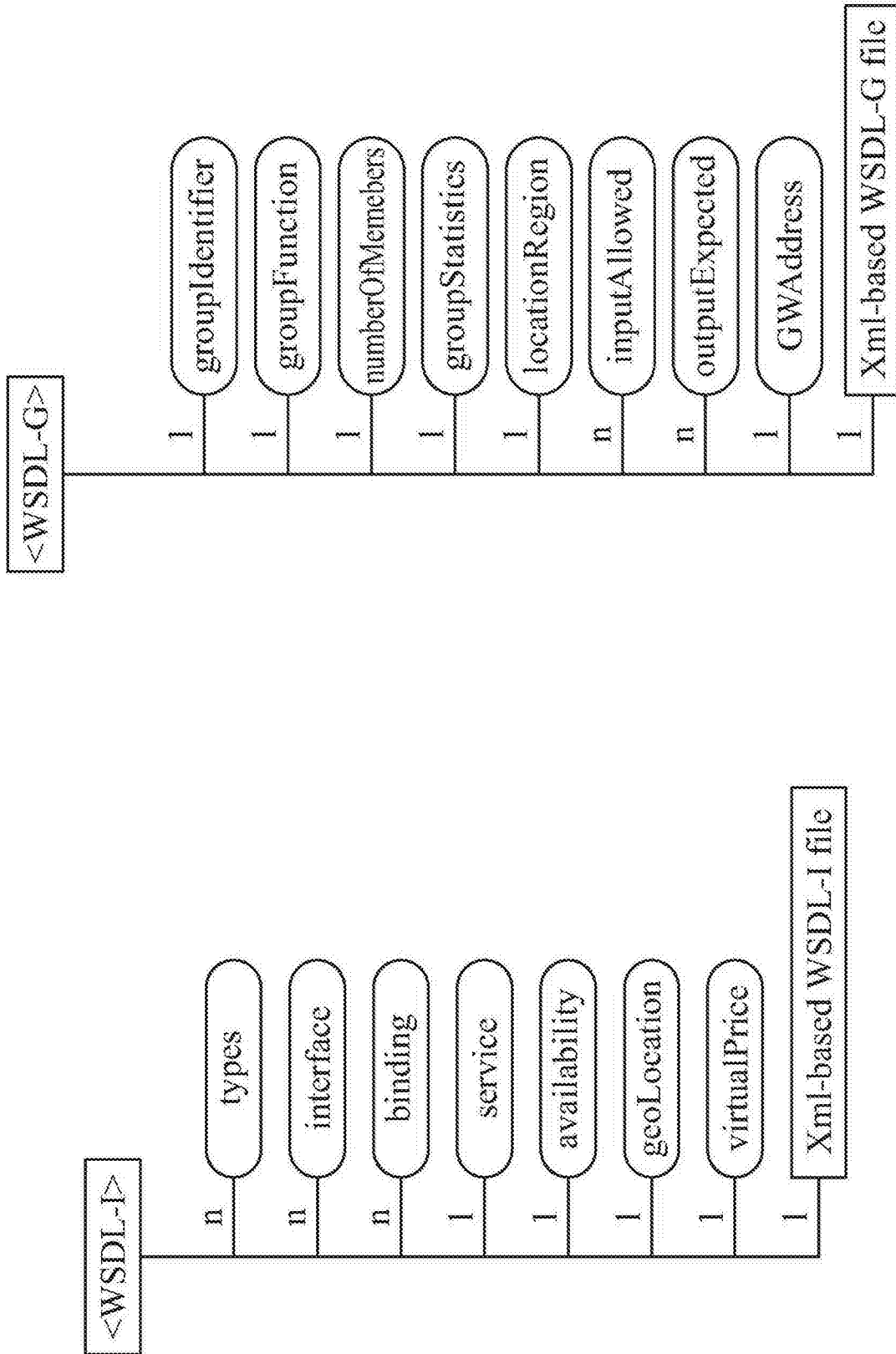


图22

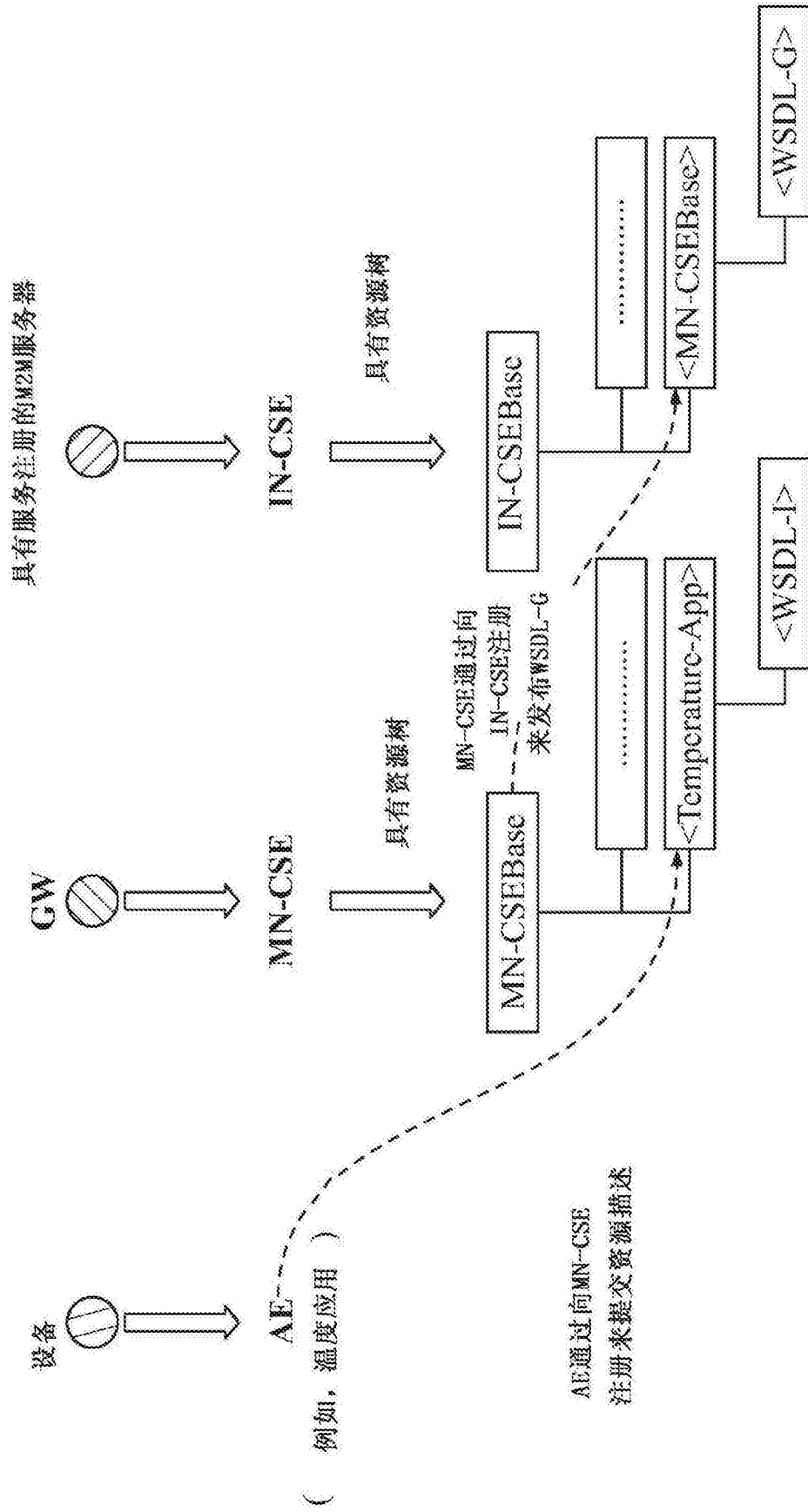


图23

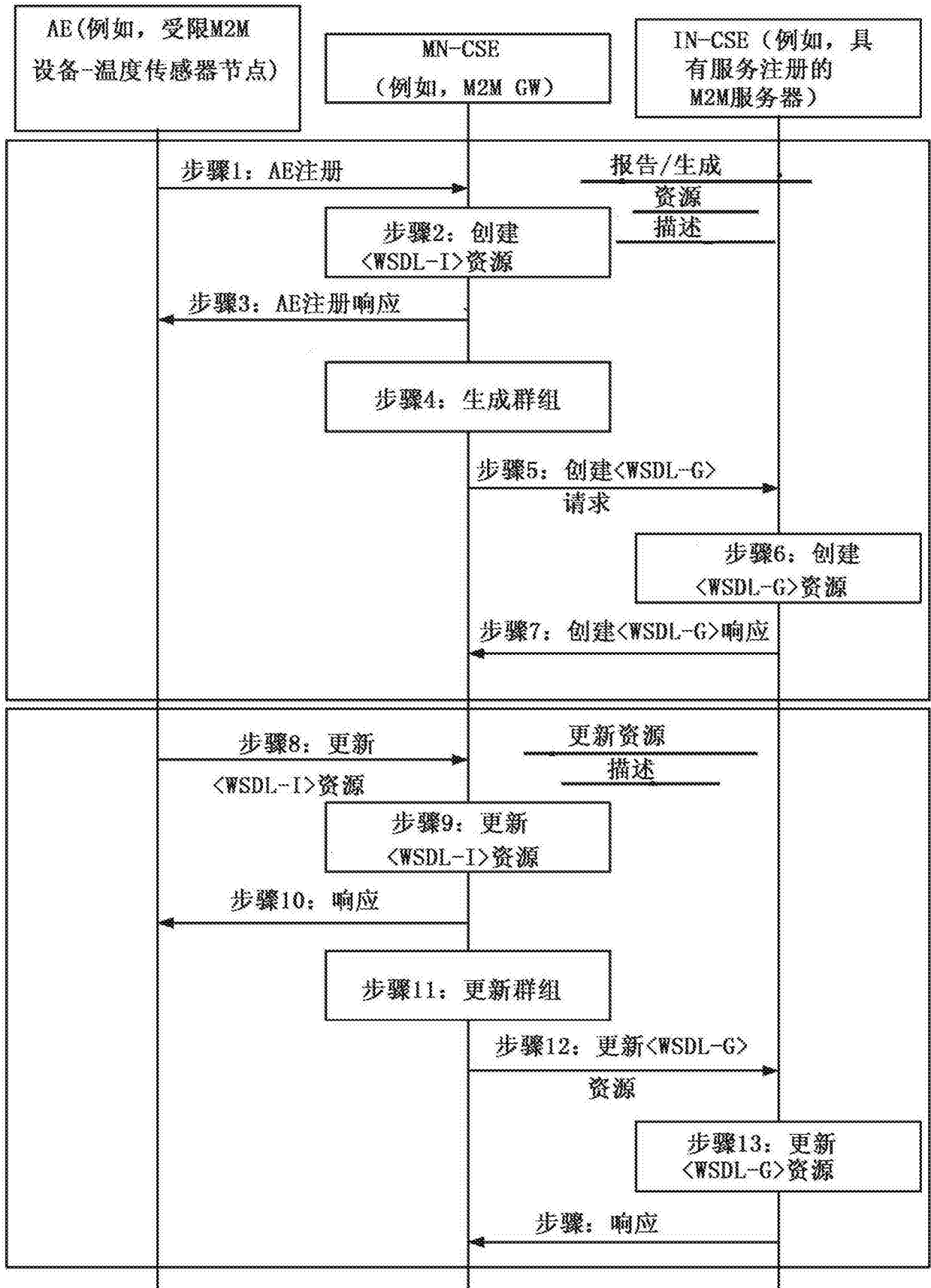


图24

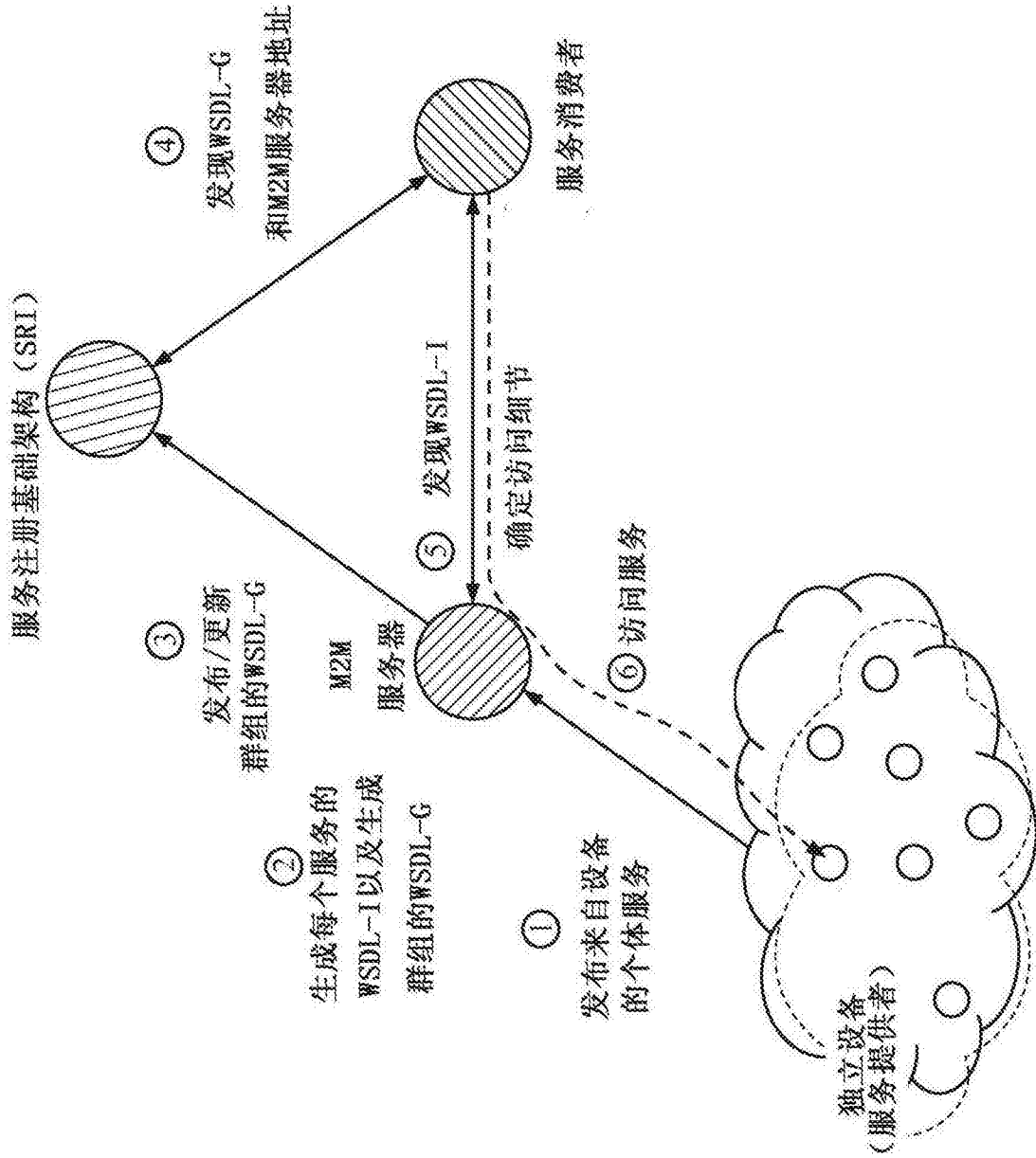


图25

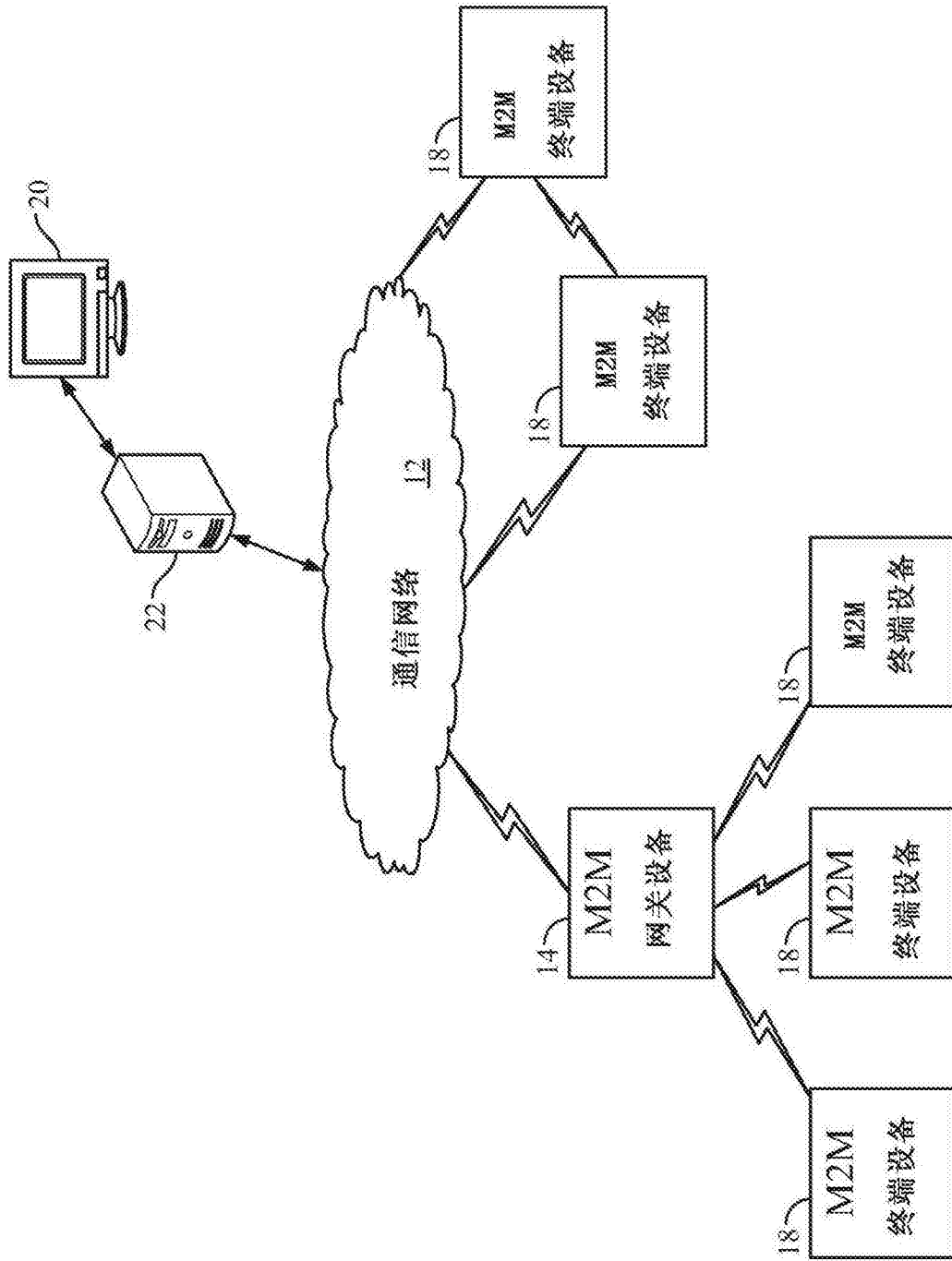


图26A

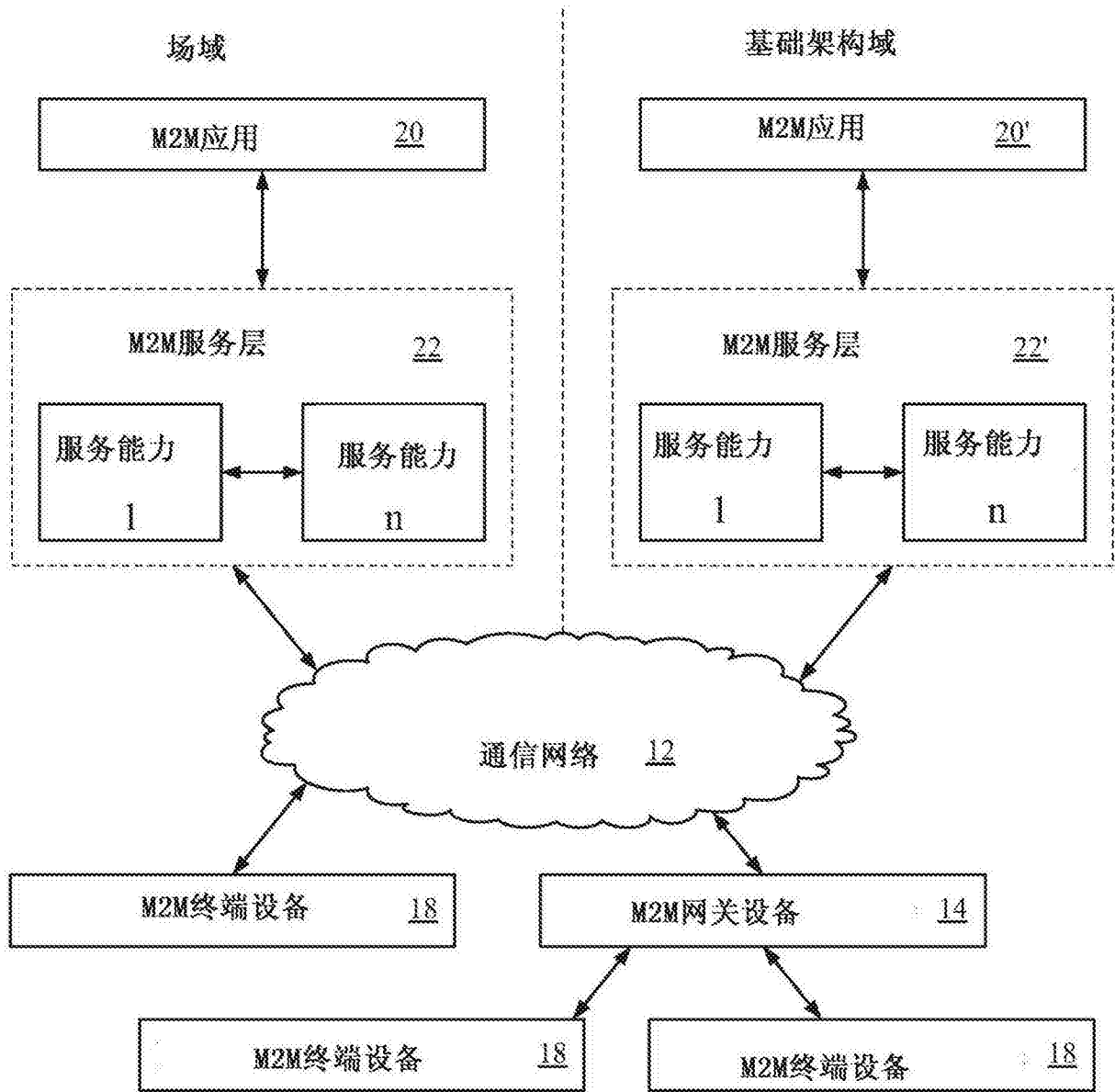


图26B

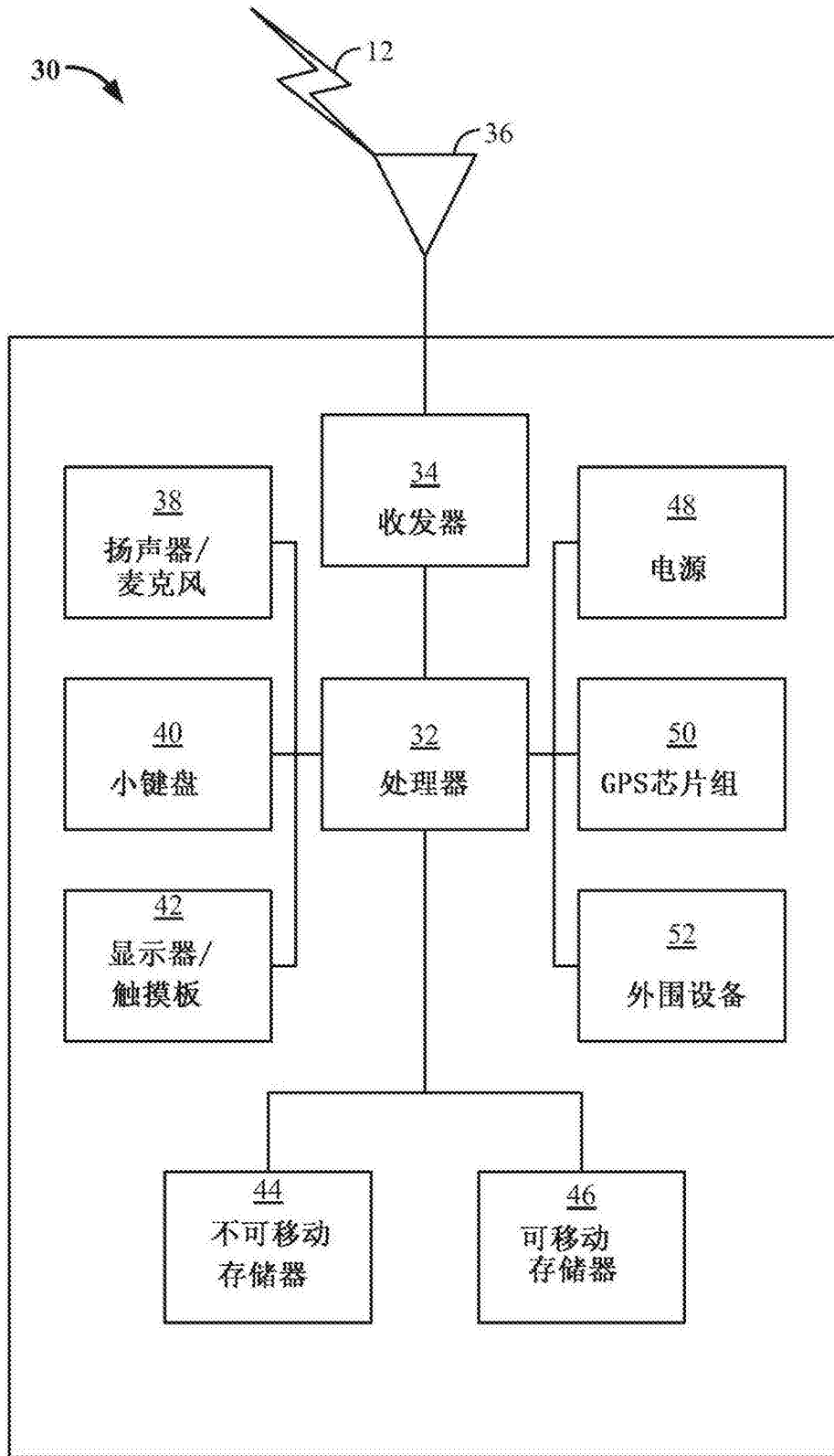


图26C

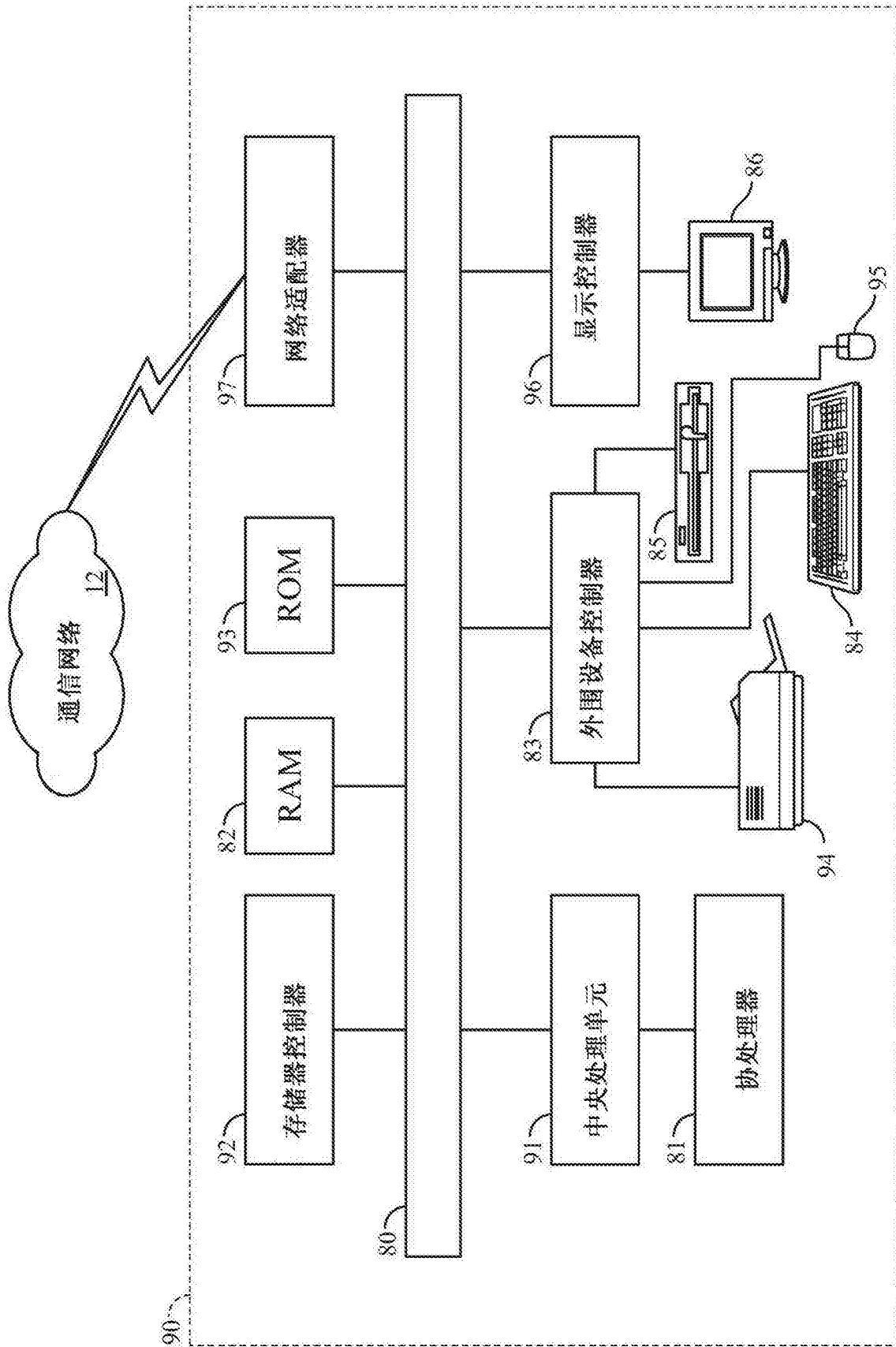


图26D