

(12) 发明专利

(10) 授权公告号 CN 101083586 B

(45) 授权公告日 2012.05.30

(21) 申请号 200710108722.9

(56) 对比文件

(22) 申请日 2007.05.31

US 2004/0024768 A1, 2004.02.05,

(30) 优先权数据

CN 1620653 A, 2005.05.25,

11/444,279 2006.05.31 US

US 2006/0106581 A1, 2006.05.18,

(73) 专利权人 SAP 股份公司

审查员 费聿辉

地址 德国瓦尔多夫

(72) 发明人 克里斯托弗·博恩霍夫德

布赖恩·S·莫 马赛厄斯·M·韦曼

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 邵亚丽 李晓舒

(51) Int. Cl.

H04L 12/28 (2006.01)

H04L 29/06 (2006.01)

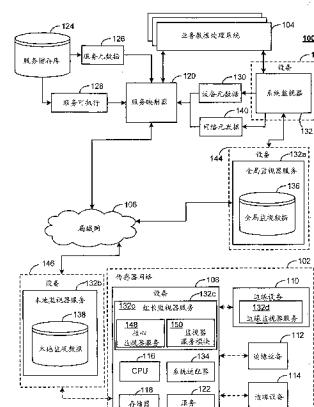
权利要求书 2 页 说明书 34 页 附图 17 页

(54) 发明名称

用于智能物件监视的模块化监视器服务系统
和方法

(57) 摘要

可以使用模块化的方法来实现可部署在设备网络上的监视器服务，其中，核心监视器服务被映射到包括在所述设备网络或与该设备网络相关联的一个或多个设备。可以使用与所述核心监视器服务交互的插件、附加服务或服务组件、或者其它服务模块向这些设备提供附加的与监视相关的功能性。可以根据例如其它服务的要求和 / 或所述设备的相关设备元数据（例如能力）来将所述核心监视器服务和任何监视器服务模块映射到所述设备中的特定设备上。在附加和或替代的实现方式中，可以使用各种协议来以快速、安全、能源有效以及可靠的方式向所述分布式监视服务注册新的设备和所部署的监视器服务，即使在设备加入或离开所述设备网络时也是如此。



1. 一种用于智能物件监视的模块化监视器服务的系统,包括:

至少第一设备(108、110),其被配置成使用位于分等级的、多级监视器架构的第一逻辑层(206、208)的核心监视器服务(148)的第一实例以及至少第一监视器服务模块(150、150c)来收集与至少一个设备网络(102)相关联的监视数据(136、138),其中所述核心监视器服务(148)被配置成实施基本功能体,而所述至少第一监视器服务模块(150、150c)被实施为根据需要添加到所述核心监视器服务(148)的插件组件;以及

至少第二设备(108、144、146),其被配置成使用位于所述分等级的、多级监视器架构的第二逻辑层(204、206)的所述核心监视器服务(148)的第二实例以及至少第二监视器服务模块(150a、150b、150c)来将所述监视数据(136、138)从所述第一设备(108、110)的至少一部分向上传播通过所述分等级的、多级监视器架构,

其中,所述第二监视器服务模块(150、150b)包括数据预处理器模块(606),其被配置成处理所述监视数据以获得所述监视数据的至少一部分,所述处理包括过滤或聚集所述监视数据中的一个或多个,

其中,所述预处理器模块(606)包括:过滤器(1406),其被配置成去除特定类型的监视数据,或者特定量的监视数据;聚集器(1408),其被配置成组合接收到的消息,以便发送组合的消息;以及更新通知发送器(1410),其被配置成提供关于监视数据的更新。

2. 如权利要求1所述的系统:

其中,所述至少第一设备(110)包括与所述至少一个设备网络(102)相关联的智能物件设备(110),并且,该智能物件设备(110)被配置成使用边缘监视器服务(132d)实现所述核心监视器服务(148)的第一实例以及所述第一监视器服务模块(150),其中,在所述边缘监视器服务(132d)收集所述监视数据;并且

其中,所述至少第二设备(108、144、146)包括与所述设备网络(102)相关联的组长设备(108),并且所述组长设备(108)被配置成使用组长监视器服务(132c)实现所述核心监视器服务(148)的第二实例和所述第二监视器服务模块(150c),其中,在所述组长监视器服务(132c)处对所述监视数据进行处理以用于传输。

3. 如权利要求2所述的系统,其中,所述至少第二设备(108、144、146)包括与所述组长监视器服务(132c)相关联的本地设备(146),并且所述本地设备(146)被配置成使用本地监视器服务(132b)实现所述核心监视器服务(148)的第三实例以及至少第三监视器服务模块(150b),其中,经过处理的监视数据(138)在所述本地监视器服务(132b)处被存储。

4. 如权利要求3所述的系统,其中,所述至少第二设备(108、144、146)包括与所述本地监视器服务(132b)相关联的全局设备(144),并且所述全局设备(144)被配置成使用全局监视器服务(132a)实现所述核心监视器服务(148)的第四实例以及至少第四监视器服务模块(150a),其中,在所述全局监视器服务(132a)处被存储的监视数据(138)被用于更新全局监视数据(136)。

5. 如权利要求1所述的系统,其中,所述核心监视器服务(148)包括下列中的一个或多个:

系统全景数据库(806),其被配置成存储已知的与所述至少一个设备网络(102)相关联的监视器数据;

信跳发送器(810),其被配置成从所述第一设备向所述第二设备发送信跳信号;

查验请求器 (808), 其被配置成测试较低层级的设备的连接;

查询路由器 (804), 其被配置成将对监视数据的查询路由到所述核心监视器服务的较低层级的实例;以及

更新通知发送器 (812), 其被配置成发送所述分等级的、多级监视器架构的全景中的变化的通知。

6. 如权利要求 1 所述的系统, 其中, 所述核心监视器服务 (148) 与模块管理器 (608) 相关联, 该模块管理器 (608) 被配置成从多个监视器服务模块中选择并实现所述第一监视器服务模块 (150、150c), 并且其中, 所述多个监视器服务模块中的每一个包括用于与所述核心监视器服务 (148) 通信的公共接口。

7. 如权利要求 1 所述的系统, 其中, 所述第一监视器服务模块 (150、150c) 包括系统适配器 (134), 其被配置成收集与在所述第一设备 (108、110) 上的服务相关联的服务元数据 (126) 和 / 或与所述第一设备 (108、110) 相关联的设备元数据 (130)。

8. 如权利要求 1 所述的系统, 其中, 所述第一监视器服务模块 (150、150c) 包括通信适配器 (602), 其被配置成进行用于所述核心监视器服务 (148) 的第一实例的通信。

9. 如权利要求 1 所述的系统, 其中, 所述第二监视器服务模块 (150、150b) 包括数据存储模块 (604), 其被配置成向所述核心监视器服务 (148) 的第二实例提供对所述监视数据的至少一部分的存储。

用于智能物件监视的模块化监视器服务系统和方法

技术领域

[0001] 本发明涉及普适 (ubiquitous) 计算技术。

[0002] 背景技术

[0003] 智能物件技术可以包括例如射频识别 (RFID) 系统、嵌入式系统、传感器微片 (mote)、和 / 或传感器网络，并且可以用于例如为业务软件应用提供对真实世界数据的快速访问。例如，可以使用智能物件技术来支持对 RFID 标签的检测、读或写，以及支持与无线传感器网络和嵌入式系统的通信和对无线传感器网络和嵌入式系统的控制。在很多实例中，智能物件可以包括下面这样的设备：所述设备具有本地处理能力、存储器和 / 或通信能力，并且能够提供关于设备及其属性的数据，或者提供关于智能物件设备的当前状态或环境的信息。因此，一些这样的设备可以用于执行后端或底层 (underlying) 业务应用的服务组件，并且，更具体来说，一些这样的设备可以通过形成自组织 (ad-hoc) 网络来以协作的方式执行后端或底层业务应用的服务组件，用以收集、处理或发送业务数据。

[0004] 智能物件设备的例子包括 RFID 标签，RFID 标签可以是无源的或有源的，并且它可以被附加到真实世界对象，且用来提供与所述对象相关的产品或处理信息。智能物件设备的其他例子包括各种传感器，诸如例如环境传感器（例如温度、湿度或振动传感器），刚刚提到的传感器可能能够进行通信，以形成一个或多个传感器网络。这些以及其他类型的智能物件设备还可以包括嵌入式系统，一般来说嵌入式系统可以指其中包括有专用处理器和 / 或程序的任何系统，并且 / 或者其中所述系统封装在正被控制的设备中。

[0005] 通过自动实时的对象跟踪，智能物件技术可以为业务提供准确及时的与业务运作有关的数据，并且还可有助于业务运作的合理化和自动化。因此，可以实现成本降低和附加商业利益（例如，增加资产可见性、提高响应性以及扩展业务机会）。

[0006] 发明内容

[0007] 根据一个总体方面，一种用于智能物件监视的模块化监视器服务的系统包括：至少第一设备，其被配置成使用位于分等级的、多级监视器架构的第一逻辑层的核心监视器服务的第一实例以及至少第一监视器服务模块来收集与至少一个设备网络相关联的监视数据，其中所述核心监视器服务被配置成实施基本功能体，而所述至少第一监视器服务模块被实施为根据需要添加到所述核心监视器服务的插件组件；以及至少第二设备，其被配置成使用位于所述分等级的、多级监视器架构的第二逻辑层的所述核心监视器服务的第二实例以及至少第二监视器服务模块来将所述监视数据的至少一部分从所述第一设备向上传播通过所述分等级的、多级监视器架构，其中，所述第二监视器服务模块包括数据预处理器模块，其被配置成处理所述监视数据以获得所述监视数据的至少一部分，所述处理包括过滤或聚集所述监视数据中的一个或多个，其中，所述预处理器模块包括：过滤器，其被配置成去除特定类型的监视数据，或者特定量的监视数据；聚集器，其被配置成组合接收到的消息，以便发送组合的消息；以及更新通知发送器，其被配置成提供关于监视数据的更新。

[0008] 实现方式可以包括下列特征中的一个或多个。例如，所述至少第一设备可以包括与所述至少一个设备网络相关联的智能物件设备，并且，该智能物件设备可以被配置成使

用边缘监视器服务实现所述核心监视器服务的第一实例以及所述第一监视器服务模块，其中，在所述边缘监视器服务收集所述监视数据。所述至少第二设备可以包括与所述设备网络相关联的组长设备，并且所述组长设备可以被配置成使用组长监视器服务实现所述核心监视器服务的第二实例和所述第二监视器服务模块，其中，在所述组长监视器服务处对所述监视数据进行处理以用于传输。

[0009] 所述至少第二设备可以包括与所述组长监视器服务相关联的本地设备，并且所述本地设备可以被配置成使用本地监视器服务实现所述核心监视器服务的第三实例以及至少第三监视器服务模块，其中，经过处理的监视数据在所述本地监视器服务处被存储。所述至少第二设备可以包括与所述本地监视器服务相关联的全局设备，并且所述全局设备可以被配置成使用全局监视器服务实现所述核心监视器服务的第四实例以及至少第四监视器服务模块，其中，在所述全局监视器服务处被存储的监视数据被用于更新全局监视数据。

[0010] 所述核心监视器服务可以包括下列中的一个或多个：系统全景数据库，其被配置成存储已知的与所述至少一个设备网络相关联的监视器数据；信跳发送器，其被配置成从所述第一设备向所述第二设备发送信跳信号；查验请求器，其被配置成测试较低层级的设备的连接；查询路由器，其被配置成将对监视数据的查询路由到所述核心监视器服务的较低层级；以及更新通知发送器，其被配置成发送所述分等级的、多级监视器架构的全景中的变化的通知。所述核心监视器服务可以与模块管理器相关联，该模块管理器被配置成从多个监视器服务模块中选择并实现所述第一监视器服务模块，并且其中，所述多个监视器服务模块中的每一个可以包括用于与所述核心监视器服务通信的公共接口。

[0011] 所述第一监视器服务模块可以包括系统适配器，其可以被配置成收集与在所述第一设备上的服务相关联的服务元数据和/或与所述第一设备相关联的设备元数据。所述第一监视器服务模块可以包括通信适配器，其被配置成进行用于所述核心监视器服务的第一实例的通信。

[0012] 所述第二监视器服务模块可以包括数据存储模块，其被配置成向所述核心监视器服务的第二实例提供对所述监视数据的所述至少一部分的存储。所述第二监视器服务模块可以包括数据预处理器模块，其可以被配置成处理所述监视数据以获得所述监视数据的至少一部分，所述处理包括过滤或聚集所述监视数据中的一个或多个。

[0013] 根据另一个总体方面，一种用于智能物件监视的模块化监视器服务的方法包括：在与至少一个设备网络相关联的分等级的、多级监视器架构的多个级中的每一级上提供被配置成实施基本功能体的核心监视器服务的实例，该核心监视器服务与收集与所述至少一个设备网络相关联的监视数据相关联。在所述多个级的至少一个上提供被实施为根据需要添加到所述核心监视器服务的插件组件的至少一个监视器服务模块，该至少一个监视器服务模块被配置成与所述核心监视器服务通信，以便将所述监视数据的至少一部分从所述至少一个设备网络向上传播通过所述分等级的、多级监视器架构。

[0014] 实现方式可以包括下列特征中的一个或多个。例如，所述至少一个监视器服务模块可以包括系统适配器，其被配置成收集与服务相关联的服务元数据和/或与设备相关联的设备元数据。可以轮询在所述至少一个设备网络的多个设备中的每一个上部署的边缘监视器服务，该边缘监视器服务包括所述核心监视器服务和所述系统适配器。可以使用至少一个组长监视器服务处理从所述多个设备接收的信跳消息，该组长监视器服务包括所述核

心监视器服务和第一数据预处理监视器服务模块。

[0015] 可以使用至少一个本地监视器服务处理从所述至少一个组长监视器服务接收到的信跳消息，该本地监视器服务包括所述核心监视器服务、第二数据预处理监视器服务模块、以及数据存储服务模块。可以使用全局监视器服务处理从所述至少一个本地监视器服务接收到的信跳信息，其中，该本地监视器服务被配置成向所述全局监视器服务发送增量同步消息以更新与该全局监视器服务相关联地存储的全局监视数据。

[0016] 根据另一个总体方面，一种用于智能物件监视的模块化监视器服务的系统包括：服务储存库，其被配置成存储核心监视器服务和多个监视器服务模块，其中，所述核心监视器服务和所述多个监视器服务模块与从至少一个设备网络获得监视数据相关联，其中所述核心监视器服务被配置成实施基本功能体，而所述多个监视器服务模中的每一个监视器服务模块被实施为根据需要添加到所述核心监视器服务的插件组件；系统映射器，其被配置成将所述核心监视器服务的实例部署到与获得所述监视数据相关联的分等级的、多级架构的至少两级上，并且其还被配置成将至少一个监视器服务模块部署到所述分等级的、多级架构的至少一级上；以及系统监视器，其包括所述核心监视器服务的实例以及所述至少一个监视器服务模块，所述系统监视器被配置成将所述监视数据的至少一部分从所述设备网络传播通过所述分等级的、多级架构。

[0017] 实现方式可以包括下列特征中的一个或多个。例如，所述系统映射器可以被配置成根据与实现至少一级的设备相关联的设备元数据，将所述至少一个监视器服务模块部署到所述分等级的、多级架构的所述至少一级上。

[0018] 所述至少一个监视器服务模块可以包括多个监视器服务模块，它们共享用于与所述核心监视器服务通信的公共接口。所述至少一个监视器服务模块可以包括系统适配器模块、通信适配器模块、数据存储模块、或数据预处理模块中的一个或多个。

[0019] 所述系统映射器可以被配置成将所述核心监视器服务部署到与所述设备网络相关联的设备，并且所述系统监视器可以被配置成使用所述核心监视器服务向所述系统映射器提供与所述设备相关联的静态元数据，并且，所述系统映射器可以被配置成根据所述静态元数据将所述监视器服务模块部署到所述设备。所述系统监视器可以包括跨越所述分等级的、多级架构部署的合成监视器服务，该合成监视器服务在所述架构的相应级包括：边缘监视器服务，其被配置成收集与所述设备网络相关联的原始监视数据；组长监视器服务，其被配置成接收和处理所述原始监视数据；本地监视器服务，其被配置成接收、进一步处理和存储经过处理的监视器数据；以及全局监视器服务，其被配置成从所述本地监视器服务接收增量同步消息并在其基础上更新全局监视数据。

[0020] 在下面的附图和说明中将阐述一个和多个实现方式的细节。通过说明书、附图和权利要求书，其它特征将变得明显。

[0021] 附图说明

[0022] 图 1 是用于监视智能物件的系统的框图；

[0023] 图 2 是图示图 1 的系统的示例部署的框图；

[0024] 图 3 是图示图 1 的系统的第二示例部署的框图；

[0025] 图 4 是用于实现图 1、图 2 和图 3 的系统的智能物件基础设施的框图；

[0026] 图 5 是图示用于使用图 1 的系统采集监视数据的示例算法的时序图；

- [0027] 图 6 是可以代表图 1 的一个或多个监视器服务 132a-132d 的模块化监视器服务的框图；
- [0028] 图 7A 是图示使用图 1 和图 6 的模块化监视器服务的图 1 的系统操作的流程图；
- [0029] 图 7B 是图示图 1 的系统的附加示例操作的流程图；
- [0030] 图 8 是图 1 和图 6 的核心监视器服务的示例实现方式的框图；
- [0031] 图 9 是图示与图 6 的模块化监视器服务一起使用的系统适配器管理器和 / 或图 8 图示的核心监视器服务的实现方式的框图；
- [0032] 图 10 是图 1、图 6 和图 9 的系统适配器的示例实现方式的框图；
- [0033] 图 11 是图 1、图 6 和图 9 的系统适配器的另一个示例实现方式的框图；
- [0034] 图 12 是图 6 的模块化监视器服务的通信适配器的示例实现方式的框图；
- [0035] 图 13 是图 6 的通信适配器的另一个示例实现方式的框图；
- [0036] 图 14 是图 6 的预处理器模块的示例实现方式的框图；
- [0037] 图 15 是图示与图 1 的系统 100 一起使用的注册协议的系统的框图；
- [0038] 图 16 是图示图 15 的注册协议的示例实现方式的顺序图 (sequencediagram)；
- [0039] 图 17 是图示图 15 的系统的示例操作的流程图。

具体实施方式

[0040] 图 1 是用于监视智能物件设备的系统 100 的框图。在图 1 的例子中，包括各种智能物件设备的传感器网络 102 使用局域网 106(尽管也可以使用其它规模或类型的网络)以及时和准确的方式向一个或多个业务数据处理系统 104 提供真实世界数据。例如，传感器网络 102 可以包括在这里称为“智能物件设备”或简单地称为“设备”(或类似的术语)的智能物件设备 108、110、112 和 114，它们可以包括例如 RFID 读取器(用于读取与具有 RFID 标签的真实世界对象相关联的 RFID 标签)、各种嵌入式系统、和 / 或各种类型的传感器和 / 或传感器微片。

[0041] 尽管作为示例的实现方式，下面的描述主要涉及传感器网络 102 或其它传感器网络，但是应当理解，这种传感器网络可以指任何采集或获得信息的网络，即使并非传感器网络的所有设备都不具有与其相关联的特定的传感器功能性(例如，传感器网络 102 的设备之一可以包括执行器 (actuator))。而且，应当理解，可以单独的或相互结合地使用几乎任何设备的网络。此外，这种设备网络的设备可以相互通信(例如以对等的方式)，并且 / 或者可以通过适当的中间件与业务数据处理系统 104 或其它系统通信。

[0042] 在图 1 中，设备 108 被图示为包括中央处理单元 (CPU) 116 和存储器 118。因此，应当将设备 108 理解为具有各种水平的计算能力，包括例如处理或发送所感测的数据(例如，在设备 108 包括传感器或设备 108 与传感器相关联的情况下)。尽管出于清楚的目的在图 1 中没有特别示出，但应当理解，设备 110、112、和 114 也可以包括相同或不同的计算能力，包括例如形成传感器网络 102 或参与到传感器网络 102 中的能力，传感器网络 102 诸如无线网络和 / 或对等网络。

[0043] 因此，传感器网络 102 可以用于收集、处理、过滤、聚集、或发送可能对于业务数据处理系统 104 有用的数据。例如，业务数据处理系统 104 可以包括库存管理系统、供应链管理系统、零售商店管理系统、仓库管理系统、和 / 或可以用来针对真实世界对象执行业务处

理的任何其它系统,其中,所述真实世界对象可以包括例如待售产品、货盘或其它装运单位 (shipment element)、病人 (patients)、或制造材料 / 设备。通过跟踪和分析这些真实世界对象,业务数据处理系统 104 可以例如用于确定库存量、设置价格层级、评估市场策略、评估制造和生产技术、减少偷盗、和维护安全性。

[0044] 通过包括智能物件设备以作为传感器网络 102 的设备 108、110、112 和 114 或者包括与传感器网络 102 的设备 108、110、112 和 114 相关联的智能物件设备,处理可以在数据收集过程的非常早期执行,从而可以减小或消除业务数据处理应用 104 上的负担。例如,业务数据处理应用 104 可以位于公司总部,而传感器网络 102 可以表示由广域网 106 连接的可以散布在大地理区域内的许多 (类型的) 网络之一。这样,例如,业务数据处理应用 104 可能仅仅需要网络 102(以及相关网络) 所收集数据的特定子集或特性,并且可以不需要或不想要所有被收集的数据。

[0045] 在一些实现方式中,业务数据处理应用 104 可以包括复合应用或合成应用,其由被设计成执行某些定义明确的任务的可重用软件组件或服务构成。并且,在这些或其它实现方式中,业务数据处理应用 104 可能包括不能与数据收集设备 (或其它业务数据处理系统) 容易地进行通信的遗留 (legacy) 应用,在这种情况下,可以提供服务或服务组件以作为遗留应用与数据收集设备和 / 或其它系统之间的接口。系统 100 允许这些和其它应用和服务直接部署到设备 108、110、112 和 114 上,从而使得例如可以以及时、高效、可靠、自动、节省成本和可缩放的方式在设备上运行服务 (以及收集和 / 或处理数据)。因此,例如,业务处理可以分解成单个服务,并且部署在不同的设备。

[0046] 如图所示,系统 100 包括服务映射器 120,其用于从传感器网络 102 的多个设备 108、110、112 和 114 中选择设备 108,以作为被选设备,用于在其上部署服务 122。在进行选择的过程中,服务映射器 120 访问服务储存库 124,服务储存库 124 用于存储适合在传感器网络 102 和 / 或其它网络 (图 1 中未示出) 内执行的多个服务。除了实际的服务可执行 (executable) 128 之外,服务映射器 120 还确定服务元数据 126,并且将服务元数据 126 与多个设备 108、110、112 和 114 中的每一个所关联的设备元数据 130 进行比较。至少基于服务元数据 126 和设备元数据 130,服务映射器 120 就可以选择设备 108 作为特别适合在其上部署服务 122(包括服务可执行 128) 的设备。

[0047] 例如,设备元数据 130 可以包括对每个设备的描述,所述描述根据服务映射器 120 已知的、且不同的设备 108、110、112 和 114 公用的本体 (ontology) 和 / 或模式 (schema) 来构造。此外,或可替代地,设备元数据 130 可以例如由服务监视器 132 以特定于设备的格式或结构针对设备 108、110、112 和 114 中的每一个来收集,并且之后可被转换成公有的模式以供服务映射器 120 使用。例如,设备元数据可以包括对设备 108、110、112 和 114 的各种技术能力的描述,该描述例如通过使用可扩展标记语言 (XML) 模式以基于 XML 的语言提供,这将在下面更具体地描述。当然,也可以使用其他格式、语言和 / 或结构。

[0048] 更一般地,设备元数据 130 可以包括例如设备描述、软件描述、硬件描述和设备状态。例如,设备描述可以包括设备名称、标识符或类型,或者可以包括销售商信息,所述销售商信息包括销售商名称或销售商网站。软件描述可以包括操作系统描述,所述操作系统描述包括版本和 / 或销售商,或者,软件描述可以包括在设备平台上运行或允许在设备平台上运行的服务的描述。硬件描述可以包括关于 CPU 116 的属性 (例如名称或速度)、存储

器 118 的属性（例如存储器总量和 / 或空闲量）或设备的连接能力的属性（例如连接速度或者连接类型）的信息。设备状态可以包括更易变的信息，包括设备位置、当前的 CPU 使用率，或者剩余的功率或存储器。当然，如下面将描述和 / 或将会明晰的那样，其他的设备方面或信息也可以包括在设备元数据 130 中。例如，设备元数据 130 可以包括关于其他设备的信息，例如如果设备 108 包括 RFID 读取器，则设备元数据 130 可以包括对可被该 RFID 读取器读和 / 或写的 RFID 标签的类型的描述。

[0049] 服务元数据 126 可以略微近似地包含与服务是否可以且如何在一个或多个设备上执行相关的各种服务描述和 / 或要求。例如，服务元数据可以包括服务行为描述、服务的技术约束、或与服务的输入、输出、前置条件或作用 (IOPE) 相关的信息。例如，技术约束可以包括所要求的 CPU 类型或速度、所需的（空余）存储器量、所要求或首选的连接类型或速度、操作系统版本 / 名称 / 描述、或电池或其他设备电源的类型或状态。

[0050] 因此，如同设备元数据 130 的情况一样，可以区分静态服务要求和动态服务要求，诸如硬件要求。例如，可以包括静态值，诸如服务所要求的总存储器或最大处理速度，以及动态值，诸如可用存储器 / 处理 / 功率、和 / 或在服务执行时可以允许与所讨论的服务一起在设备上并发运行的其他服务的数目或类型。

[0051] 至少使用服务元数据 126 和设备元数据 130，服务映射器 126 可以将给定的服务映射到传感器网络 102 的设备 108、110、112 和 114 上。这种映射被设计成不仅在必要时对服务元数据 126 和设备元数据 130 的不同方面进行值匹配 (value-match)（例如，使在服务元数据 126 中指定的需要的存储器与在设备元数据 130 中指定的设备存储器匹配），而且还使能和最优化所讨论的服务的部署和执行。例如，情况可能是设备 108 和设备 110 两者名义上或表面上都能够运行服务 122（例如设备 108 和 110 两者可能都具有一些对存储器、处理能力或功率的最低要求的值）。但是，情况可能是服务 122 对功率的需求更甚于对存储器的需求（或者反过来），所以如果设备 108 目前能够相对于另一候选设备 110 提供更多的功率，则即使设备 110 目前提供比设备 108 更多的空闲存储器，服务映射器 120 也可能将服务 122 映射到设备 108。

[0052] 一旦已经执行了适当的服务映射，就可以使用服务注入器（在图 1 中未 示出）来安装或开始设备 108 上所映射的服务（例如服务 122）。这种服务映射器还可以例如在需要时执行服务更新或停止服务，从而用来例如管理服务的生命周期。

[0053] 如上面提到的，在系统 100 的运行时间期间，系统监视器 132 被配置成确定、跟踪、和 / 或提供系统 100 的当前状态。这种状态信息可以包括例如设备 108、110、112、114 中的哪些当前可用于服务部署、或者在设备 108、110、112、114 中的哪些设备上当前运行了什么服务。更一般地，系统监视器 132 可以检测和收集设备元数据 130，包括将在下面更具体地描述的静态、离散动态、和连续动态的设备元数据以及也将会在下面更具体描述的与传感器网络 102 相关的网络元数据 140。很显然，这样就使得诸如设备元数据 130 和 / 或网络元数据 140 的监视数据对于服务映射器 120、业务数据处理系统 104 和 / 或系统管理员 (administrator) 来说是可获得的。

[0054] 在图 1 的例子中，使用四级架构 (four-tier architecture) 来实现系统监视器 132，以提供可缩放的 (scalable)、分布式的监视服务。也就是说，系统监视器 132 可以使用一个或多个监视服务来实现，其中，所述监视服务可以存储在服务储存库 124 中，并且可以

由服务映射器 120 全部或部分地映射到传感器网络 102 (以及与传感器网络 102 相关联的设备)。这种四级架构可以结合这里所描述的算法来使用,从而在诸如系统 100 的分布式系统全景 (landscape) 下有效地收集监视数据。

[0055] 这种监视数据可以包括例如设备元数据 130 (其可以包括例如如这里所述的设备 108、110、112、114 的连接性信息、电池寿命或 CPU/ 存储器使用率) 和 / 或网络元数据 140。网络元数据 140 可以包括例如各种网络参数,特别是在这些参数是动态的并且不必要与关于任何单个设备的信息可区分的情况下。网络元数据 140 的一个这样的例子可以包括传感器网络 102 上可用的带宽。其它的例子包括网络拓扑信息、网络作为整体的移动性 (mobility) 特性、以及网络连接的可靠性。

[0056] 在一些实现方式中,如这里所描述的,这些监视器服务可以使用模块方式来实现,其中,核心监视器服务被映射到包括在传感器网络 102 中或与传感器网络 102 相关联的一个或多个设备中。然后,可以使用与核心监视器服务进行交互的插件 (plug-in)、附加 (add-on) 服务或服务组件或其它服务模块来向这些设备提供附加的与监视相关的功能性。应当理解,核心监视器服务和任何补充的监视器服务可以使用服务映射器 120 并且基于例如特定于应用的要求和相关的设备元数据 (例如能力) 来映射。在附加的或可替代的实现方式中,即使是在设备加入或离开传感器网络 102 时,也可以使用各种协议以快速、安全、能源有效和可靠的方式向分布式的监视服务注册新的设备和所部署的监视器服务。

[0057] 然后,在图 1 的例子中,上面提到的四级架构包括四个类型或类别 (class) 的监视器服务,它们可以是层级分层 (hierarchically) 的。具体来说,最上层 (这里也称为全局 (global) 层) 包括全局监视器服务 (GMS) 132a。全局监视器服务 132a 提供对监视数据的最高级别的存储和查看 (view),不仅仅是针对传感器网络 102,而是潜在地针对多个传感器网络 (例如见图 2)。因此,例如,全局监视器服务 132a 可以提供用于其它系统组件 (例如服务映射器 120)、业务数据处理系统 104、系统管理员和 / 或系统管理图形用户接口 (GUI) 的系统监视信息的中心接触点 (central contact point)。全局级别的监视数据可以存储在图 1 中由全局监视数据 136 表示的存储器中,因此可以认为该全局监视数据 136 表示或包括至少在高级别描述传感器网络 102 和其它网络的设备的全局设备元数据、以及关于例如在哪些网络上正在运行哪些服务的当前服务信息。

[0058] 在概念上全局监视器服务 132a 在第二级或层之上,或由第二级或层更新,在这里,第二级或层被称为本地层或本地级,包括本地监视器服务 132b。本地监视器服务 132b 被配置成向全局监视器服务 132a 提供关于底层传感器网络 102 中的变化的信息。因此,本地监视器服务 132b 可以被配置成使用在图 1 中被表示成本地监视数据 138 的存储器来存储关于传感器网络 102 中的设备 108、110、112、114 的监视数据,这些监视数据比实际可能与全局监视器服务 132a (和全局监视数据 136) 相关联地存储的数据要更加具体。因此,应当理解,本地监视器服务 132b 可以负责一个或多个传感器网络或其它设备集群 (cluster),并且这些网络 / 设备可以包括彼此相对紧密地物理邻近地部署的潜在异类设备。因此,本地监视数据 138 可以被看作是表示关于在本地监视设备 132 的范围内的网络 (诸如传感器网络 102) 的本地设备元数据,其中,应当理解,一般来说,对于传感器网络 102 和其它被监督的网络的具体设备来说,与全局监视数据 126 相比,本地监视数据 138 可能更加具体。

[0059] 第三层或级包括组长 (group leader) 监视器服务 132c, 其可以被配置成 用于收集、过滤和聚集针对设备 108、110、112、114 的监视数据。在这里所述的各种实现方式中, 组长监视器服务 132c 可以被配置成使用例如轮询或信跳 (heartbeat) 技术对从相关的设备 108、110、112、114 收集监视数据进行管理。于是, 组长监视器服务 132c 负责将得到的监视数据报回其各自本地监视器服务 132b。在图 1 的例子中, 组长监视器服务 132c 不存储监视数据, 因为在设备 108 的资源有限, 或者因为由组长监视器服务 132c 处理的监视数据的变化速率使得存储不现实。因此, 组长监视器服务 132c 可以被看作是用来提供关于其组内的具体设备 108-114 的组级别的设备元数据, 包括例如动态或静态设备特性、或关于在设备 108-114 中的哪个设备上正在运行哪些服务的信息。

[0060] 图 1 中示例的四级架构的第四层或级包括边缘监视器服务 132d。在图 1 的例子中, 边缘监视器服务 132d 收集针对其相关联的设备 (这里是设备 110) 和该相关联设备的本地服务的监视数据。也就是说, 应当理解, 就像设备 108 被图示为执行服务 122 那样, 设备 110 也可以执行一个或多个服务, 并且边缘监视器服务 132d 可以被配置成例如监视关于这些服务的信息, 并且监视设备 110 本身所特有的静态或动态数据 (例如总存储器、当前可用的存储器、设备类型和 / 或其它设备特性)。边缘监视器服务 132d 还被配置成将所收集的监视数据报告给组长监视器服务 132c。这样, 边缘监视器服务 132d 提供关于具体的 (通常是单个的) 设备, 诸如设备 110 的边缘级别的设备元数据。

[0061] 在上述架构中, 监视器服务 132a-132d 可以被配置成负责不同的任务, 并且一般来说, 可以朝着系统 100 的架构的边缘方向越来越轻便 (lightweight)。结果, 例如, 边缘监视器服务 132d 可以在具有非常有限的计算能力的设备 110 上实现, 而全局监视器服务 132a 可以在设备 142 和 / 或 144, 诸如例如个人计算机 (PC) 或类似强大的服务器机器, 上实现。同时, 本地监视器服务 132b 可以例如在设备 146, 诸如例如智能 RFID 读取器或个人数字助理 (PDA) 上实现。如图 1 所示, 组长监视器服务 132c 可以被部署到传感器网络 102 的设备之一上 (尽管应当理解设备 108、110、112、114 可以是具有不同计算能力的异类设备, 从而可以根据例如设备 108 的相对较高级的资源而选择其作为组长), 或者, 在另一个例子中, 组长监视器服务 132c 可以被部署到与设备 110、112、114 通信的无线接入点上。下面, 针对图 2-3 提供几个不同的关于如何将监视器服务 132a-132d 在物理上部署到不同的设备 108-114、142、144 和 / 或 146 的一些或所有设备以及其它设备上的例子。

[0062] 可以使用用于特定基本监视功能性的核心监视器服务 148, 结合图 1 中由监视器服务模块 150 表示的多个潜在的插件、附件、服务组件或其它模块, 来实现监视器服务 132a-132d 中的一些或全部。通过这种方式, 例如, 监视器服务模块 150 可以与设备 108 中的 (并且, 类似地, 尽管在图 1 中未示出, 与设备 142、144、146 和 110-114 中的) 核心监视器服务 148 相关联, 从而使得可以根据特定于应用的需要和 / 或各个设备的能力来专门化 (specialize) / 扩展核心监视器服务 148, 该核心监视器服务 148 可能非常微小 (slim) 从而可以在小型或资源贫乏的设备上运行。因此, 例如, 可以在不改变核心监视器服务 148 的情况下将与通信、数据存储或数据处理相关的功能性添加或替换到给定设备上。尽管在图 1 中没有特别示出, 但是将参照图 6-14 更详细地讨论核心监视器服务和相关监视器服务模块的使用的例子。

[0063] 在操作期间, 系统 100 可以使用下面的示例算法来采集监视数据, 包括例如设备

元数据 130 或网络元数据 140。具体来说，监视数据可以在组长监视器服务 132c 处被采集，从那里其被传播到本地监视器服务 132b，然后到全局监视器服务 132a。为了采集数据，组长监视器服务 132c 可以执行系统适配器 134，系统适配器 134 本身可以是例如由服务映射器 120 从服务储存库 134 映射到设备 108 的服务。系统适配器 134 可以在概念上表示多个不同类型的适配器，并且还可以被看作是监视器服务模块 150 的附加或可替代的示例。系统适配器 134 可以被配置成与服务 122 接口连接 (interface)，从而可以从服务 122 采集数据，或者采集与服务 122 相关的数据。在其它的例子中，系统适配器 134 可以表示被配置成与设备 108 的组件，例如 CPU 116 或存储器 118 接口连接的设备适配器，从而提供与设备 108 相关的例如特定于硬件的信息。

[0064] 因此，为了采集数据，组长监视器服务 132c 可以执行系统适配器 134，从而系统适配器 134 用作数据源，以提供关于设备和服务的监视数据。例如，在图 1 的例子中，监视数据可以包括关于设备 108 的信息，和 / 或关于服务 122 的信息。监视数据通过例如信跳消息被发送到上级，例如，经由本地监视器服务 132b 到全局监视器服务 132a。在示例实现方式中，组长监视器服务 132c 从边缘设备 110、112、114 收集监视数据。特别地，边缘设备 110、112、114 中的每一个可以与其自身的监视器服务，例如边缘设备 110 的边缘监视器服务 132d，相关联。正如应当理解的那样，并且下面将更具体的描述，监视器服务 132a-132d 中的任何一个都可以使用核心监视器服务 148 (的实例) 以及不同示例和类型的监视器服务模块 150 来实现。

[0065] 组长监视器服务 132d 和本地监视器服务 132b 可以被配置成执行消息聚集，其中来自不同设备 (例如来自不同的组长设备、和 / 或来自不同的边缘设备) 的消息被概括成单个消息。在组长监视器服务 132c、本地监视器服务 132b 和全局监视器服务 132a 中的每一个处，监视数据可以被预处理 (例如过滤或聚集)，并且这种预处理可以由监视器服务模块 150 或类似的模块来执行。因此，可以减少传送到全局监视器服务 132a 的监视数据量。

[0066] 于是，在图 1 的系统 100 的架构中，(更新过的) 监视数据被从较低级上行传播到较高级，例如从边缘监视器服务 132d 传播到全局监视器服务 132a。如图所示，监视数据可以以不同的形式和以不同的程度存储在不同的级 / 层内。例如，如图所示，数据可以存储在全局监视数据 136，和 / 或在本地监视数据 138，其中，应当理解，本地监视器服务 132b 和本地监视数据 138 (和设备 146) 可以表示与系统 100 相关联的多个服务、存储器和设备 (例如参见图 2-4)。

[0067] 可以使用不同的技术来确定将存储 (或不存储) 在图 1 的系统 100 的架构的给定一级的监视数据的类型和程度。例如，对于是否以及在哪里存储监视数据的判定可以基于电池寿命、存储器可用性、设备可靠性、网络利用率、响应时间、和过时信息 (例如，可能存在保持特定类型的监视数据在特定程度上为最新的要求)。例如，与本地监视器服务 132b 相关联的设备 146 可能具有特定量的可用存储器，而另一个设备 (未示出) 可能实现在层级上与全局监视器服务 132a 相关联的另一个监视器服务，并且可以具有相对更大量的空闲存储器。

[0068] 是否在给定地点 (例如在本地监视数据 138 或全局监视数据 136) 存储监视数据的一个方面涉及监视数据的优先级。例如，诸如设备 ID、设备类别、以及有效状态 (alive-status) 的信息可以具有被存储的高优先级。另一个方面涉及监视数据的本质。例

如,来自组长监视器服务 132c 的静态和离散动态数据可以在查询时被存储,从而不频繁读取经常查询的监视数据(例如,被有效地缓存(cache)在本地监视数据 138 处)。作为另一个例子,必需管理(潜在地)过时数据,从而根据需要更新该过时数据。同时,可以不存储连续动态数据,因为它可能在存储时就过时了。因此,可以简单地根据需要从相关设备读取(例如,查询)连续动态数据。

[0069] 可以通过评估结合了诸如例如存储所需的硬件能力、存储位置与传感器的接近程度、或要存储的信息的类型等各方面的表达式来动态地进行对于是否以及如何在给定位置存储监视数据的判定。例如,每个设备(例如设备 146)可以根据相关的硬件简档、对于查询的响应时间、到边缘设备 110-114 的距离、数据优先级和 / 或数据的动态类型(例如静态、离散动态或连续动态)来计算一个值。可以使用数学表达式(例如在注册具体服务或设备期间实现的数学表达式,在下面将会参照图 7B 和图 15-17 具体描述这种注册的例子)来定义得到的值,其中,所考虑的方面被数字表示和加权,以用于后续的组合以及与定义的阈值的比较。在公式(1)中示出了这种表达式的例子:

[0070] 设备能力描述值 = (剩余寿命) $\times w_1 + (\text{CPU 速度}) \times w_2 + (\text{CPU 利用率}) \times w_3 + (\text{存储器大小}) \times w_4 + (\text{存储器利用率}) \times w_5 + (\text{网络带宽}) \times w_6 + (\text{网络利用率}) \times w_7 + (\text{设备类别}) \times w_8 + (\text{信息种类}) \times w_9$

[0071] 等式(1)

[0072] 除了刚才描述的动态 / 自动方法之外,也可以进行手动存储判定。例如,系统管理员可能想要定义应当或不应当存储在指定设备上的监视数据的类型或实例。例如,管理员可以具有否决(overrule)上面的自动判定并定义附加的或替代的存储规则的权限。这种手动判定可以例如针对每个特定的设备进行,或者针对每个设备类型进行,或者基于每个监视器服务的类型 / 角色进行(例如可以要求所有本地监视器服务 132b 存储特定类型的监视数据)。

[0073] 最后,参照图 1,并且如这里将参照例如图 7B 和 15-17 再次更详细地描述的,应当理解,监视器服务(以及相关的设备)可以离开或加入网络。例如,设备 110 可以最初不是传感器网络 102 的一部分,并且可以是可以从一个位置移动到另一个位置的移动设备或便携式设备。这样,设备 110 可以被运送到传感器网络 102 的位置,并且设备 110(例如边缘监视设备 132d)可以尝试在组长监视器服务 132c 处注册,或者在本身连接到全局监视器服务 132a 的任何设备处注册。因此,可以使用专用协议来进行这种注册,从而使得设备和监视器服务可以容易地或自动地被配置成离开一个传感器网络并加入另一个。这种专用协议的例子将在下面参照图 15-17 来具体讨论。

[0074] 图 2 是图示图 1 的系统 100 的示例部署的框图 200。图 2 示出了,如上所述地,系统 100 可以跨越四个概念级或层来部署,这四个概念级或层在图 2 中被表示为全局层 202、本地层 204、组层 206 和边缘层 208。从对图 1 的描述应当理解,层 202-208 可以具有不同的任务,并且每个与它们各自类型的监视器服务(例如全局监视器服务 132a、本地监视器服务 132b、组长监视器服务 132c、和边缘监视器服务 132d)相关联。每个类型或类别的监视器服务可以根据设备的能力而与物理设备(例如图 1 的设备 144、146、108 和 110)相关联。结果,例如,相对轻便的边缘监视器服务 132d 可以在具有非常有限的计算能力的例如传感器微片的设备上运行,而全局监视器服务 132a 可以在具有相当多的计算资源的设备,

例如 PC 或服务器上运行。

[0075] 在一些示例的实现方式中,如图 1 所示,上面提到的监视器服务 132a-132d 的变化可以通过监视器服务 132a-132d 的模块化部署的方式来实现。例如,可以仅使用核心监视器服务 148(在图 2 中示为核心监视器服务 148d) 的实例在边缘层 208 部署边缘监视器服务 132d。

[0076] 同时,可以使用核心监视器服务 148c 的实例以及监视器服务模块 150(图 2 中示为监视器服务模块 150c) 的实例来部署组长监视器服务 132c。即,应当理解,监视器服务模块 150c 可以表示一个或多个监视器服务模块,其中所述监视器服务模块被配置成实现上面参照图 1 的组长设备 108 所描述的监视功能性。例如,监视器服务模块 150c 可以与聚集从边缘层 208 的所有边缘监视器服务 132d 接收的监视数据(例如消息)相关联,用于将它们报告给本地监视器服务 132b。

[0077] 沿着类似的线,图 2 的本地监视器服务 132b 可以包括核心监视器服务 148b 以及监视器服务模块 150b,其中监视器服务模块 150b 可以被配置成实现与本地层 204 相关联的功能性(例如在本地监视数据 138 中存储监视数据,或者更新全局监视数据 136)。最后,类似地,全局监视器服务 132a 可以包括核心监视器服务 148a 以及特定于全局层的监视器服务模块 150a。

[0078] 应当理解,图 2 仅仅是一个例子,许多变化都是可能的。例如,在一些例子中,核心监视服务 148a-148d 可以是相同的或基本相同的,而在另一些例子中,它们可能有本质的不同。例如,核心监视器服务 148a 和 148b 可以不同于核心监视器服务 148c 和 148d,并且例如可以包括与核心监视器服务 148c 和 148d 不要求的数据存储相关联的功能性,而不是在这一功能性方面依赖与监视器服务模块 150a 和 150b。在另一些例子中,全局监视器服务 132a 可以以集成的、非模块化的方式来实现,而剩下的监视器服务 132b-132d 可以使用这里所描述的模块化结构来实现。核心监视器服务 148a-148d 的模板可以存储在服务储存库 124 中,以便设备 144、146、108、110 可以例如通过服务映射器 120 接收被实例化以用于部署的核心监视器服务 148a-148d 的实例。

[0079] 图 2 还示出了系统 100 的层级本质。即,在图 2 中,在多个本地监视器服务 132b 之上维持(maintain)全局监视器服务 132a,并且全局监视器服务 132a 与所述多个本地监视器服务 132b 通信;并且,在多个组长监视器服务 108 之上维持所述多个本地监视器服务 132b 中的每一个本身,并且所述多个本地监视器服务 132b 中的每一个本身与该所述多个组长监视器服务 108 通信。最后,如在图 1 中已经示出的,可以在多个边缘设备 132d 之上维持每个组长监视器服务 132c,并且每个组长监视器服务 132c 与所述多个边缘设备 132d 通信。

[0080] 在图 2 中,如在图 1 的例子中所预期的,可以是每个监视器服务 132a-132d 被部署在单个物理设备上的情况。然而,更一般地,情况也可以是多个(多个类型的)物理系统或设备与逻辑级或层 202-208 中之一相关联。相反,情况也可以是,多个层 202-208 可以与单个(类型的)物理系统或设备相关联。

[0081] 图 3 是示出图 1 的系统 100 的第二示例部署的框图 300,其中,层 202-208 可以与多个(类型的)物理系统或设备相关联。在图 3 的例子中,公司服务器 302(例如全公司的(company-wide)中央服务器)可以位于第一地理位置(例如欧洲),并且可以实现全局服

务监视 132a。

[0082] 同时,生产工厂服务器 304(例如在该公司的生产工厂实现的或位于该公司的生产工厂的服务器计算机)可以位于例如在旧金山的该公司的生产工厂。生产工厂服务器 304 以及 Stargate 计算机 306(例如具有通信和传感信号处理能力的单板计算机)和 WiFi 接入点 308 可以在本地层 204 实现。换言之,可以使用物理设备 304-308 中的一个或多个来实现一个或多个本地监视器服务 132b。

[0083] 在图 3 的例子中,例如(至少)对于 Stargate 306 和 WiFi 接入点 308,应当理解为至少包括两个本地监视器服务 132b。即,在至少一个组长监视器服务之上维持每个这种本地监视器服务,而该组长监视器服务应当被理解为 在传感器微片 310 和 PDA 312 上运行,其中,传感器微片 310 是多个传感器微片 314-318 的组长(因此传感器 314-318 表示潜在地(边缘)设备 110-114 的例子)。

[0084] 于是,在图 3 的例子中,存在多于四个物理层,并且这些物理层可以被映射到图 2 的四个逻辑级或层 202-208。因此,多个物理系统层可以担任一个概念上的级的角色。在使用多于四个物理层的例子中,设备可以在概念级之间路由(route)信息。例如,边缘设备可以将信息从本地监视器服务 132b 路由到组长监视器服务 132c,即使在所讨论的边缘设备可能属于不同的传感器网络、但相对于组长监视器服务 132c 来说在物理上更接近本地监视器服务 132b 的情况下也是如此。

[0085] 然而,如上面所提到的,物理系统 / 设备还可以具有少于四级。例如,可以只有两个物理级,在这种情况下,全局监视器服务 132a 可以被部署在第一设备上,同时本地监视器服务 132b、组长监视器服务 132c 和边缘监视器服务 132d 可以部署在第二设备上,有可能伴随有监视器服务 132b-132d 的单个的、组合实例。

[0086] 在图 1-3 的例子中,设备的层级可以是概念上的树型结构,例如,每个设备可以具有恰好一个上(或相同)层监视器服务以向其报告(其可以被称为“父代(parent)监视器”)。包含全局监视器服务 132a 的全局层设备,例如设备 144,可以是唯一没有这种父代监视器的监视器服务。出于收集监视数据的目的,边缘设备(例如边缘设备 110-114)可能不需要相互之间直接通信,而是可以简单地将监视数据转发给它们的父代设备。当然,在这些或其它上下文中,设备也可以直接相互通信(例如,以对等的方式,可能用来计算房间内的平均温度或用来执行其它协作的计算)。

[0087] 更一般地,设备可以在物理上连接到同一网络中的其它设备,或者可以连接到几个网络。因此,在其它实现方式中,监视器服务 132b-132d 可以具有几个父代设备。在一些附加的或替代的实现方式中,如上面所提到的,可以存在恰好一个主网络连接,其可以被定义为通过最短路径(在跳数(hop)方面)到全局层设备 144 的连接。其它的连接(例如对等连接和 / 或多个网络连接)可以被看作是辅助连接或“捷径”(shortcut)连接,从而例如使得第一传感器网络上具有本地监视器服务 132b 的本地层设备无需通过全局层来与第二传感器网络的边缘设备通信,而是可以使用居间的组长设备与边缘设备通信。

[0088] 图 4 是用于实现图 1-3 的系统的智能物件基础设施 400 的框图。智能物件基础设施 400 包括五层:设备层 402、设备级(level)服务层 404、业务处理桥接层 406、系统连通(connectivity)层 408 以及企业应用层 410。层 402 可以被认为包含图 1 的设备 108、110、112 和 114 中的不同设备,或者跨越多个组、本地网络和 / 或物理位置的类似的设备。同时,

层 406、408 和 410 可以被看作是图 1 的业务数据处理系统 104 的一部分,或者与该业务数据处理系统相关联。因此,层 404 可以被看作是表示图 1 的系统 100 的剩余的组件,例如服务映射器 12、系统监视器 132 和 / 或监视器服务 132a-132d、以及服务储存库 124,如图 4 所示。

[0089] 因此,设备层 402 包括实际的智能物件设备,以及它们之间的任何通信。设备层 402 还负责向临近的最高层,设备级服务层 404 呈现任何被提供的 (offered) 的硬件服务。所述设备可以包括例如 RFID 设备 412、嵌入式系统 414、传感器网络 416 和任何其它适当的新颖或新兴的技术 418。应当理解,设备层 402 的例子是非限制性的,并且可以重叠。例如,传感器网络可以包括 RFID 设备 (读取器 / 标签)、嵌入式系统、和 / 或其它新兴的技术。

[0090] 对于 RFID 设备 412 来说,移动标签可以被附加到真实时间对象,然后由 RFID 读取器读取 (并且可选地向其写入)。在使用有源标签的实现方式中,有源标签也可以提供附加的传感器数据 (例如当前值或过去的值)。在 RFID 中,通信一般由读取器启动,同时标签可以直接相互通信,或者可以不直接相互通信。这种 RFID 读取器可以被配置到处理标签数据的程度,例如,可以被配置成执行对所写数据的验证,或者被配置成避免在表面上丢失的标签实际上在给定时间窗口内重新出现的情况下报告标签消失。

[0091] 用于与嵌入式系统 414 通信的技术可以随嵌入式系统的设备的类型而变化。例如,嵌入式系统可以表示从小规模单片微计算机一直到成熟的 PC 硬件的任何系统。因此,例如,对于具有移动电话或更多 (例如能够运行 JavaVirtual MachineTM(Java 虚拟机)) 的能力的设备来说,可以用 JavaTM 来进行实现,或者可以基于 OSGi 来实现 (后者表示用于实现对应用和 / 或应用组件进行远程安装和管理的组件模块的已知框架)。如同样已经讨论过的,传感器网络 416 可以包括任何数目的类型的传感器,所述传感器可以包括集成的处理能力并且可以执行对等通信。

[0092] 服务储存库 124 可以存储至少两种服务,复合服务和原子服务。复合服务一般依靠其它设备来完成其任务,并且可以不具有它们自己的直接可执行代码;然而复合服务可以包括存储在相应服务描述中的可执行服务合成描述。因此,复合服务可以具有一个服务可执行,即,服务合成描述。相反,原子服务一般不使用其它服务,并且具有它们自己的直接可执行代码。并且,如上面所提到的,由于原子服务可以被部署在不同的平台上,因此原子服务可以具有多于一个的服务可执行,例如,可以具有用于不同平台中的每一个的服务可执行。

[0093] 服务储存库 124 还可以存储服务元数据 126,其中的所述服务元数据 126 已经在上面具体描述过,其可以包括服务名称、标识符、版本、或厂商,或者可以描述服务的运行时要求,包括例如技术部署要求 (例如高带宽、或所需的最小处理能力)、语义 (semantic) 要求 (例如接收设备具有串行连接和 / 或许多设备邻居)、以及空间要求 (例如接收设备在地下室中、或者在特定建筑物的南侧)。

[0094] 最后,在设备级服务层 404 可以包括设备存储库 420。从上面的描述应当理解,设备存储库 420 可以以类似于服务储存库 124 维护关于服务的信息 (例如服务元数据) 的方式包括例如关于设备的信息 (例如设备元数据)。例如,设备元数据可以基于与系统监视器 132 的通信 (例如与监视器服务 132a-132d 中的一个或多个的通信) 而存储在设备存储库 420 中。在其它的示例实现方式中,设备存储库可以包括由管理员根据外部可获得的关于设

备的信息而存储的设备元数据。例如,如已经提到的,设备元数据可以包括设备名称、功率容量、存储器容量、处理能力、或可能与将服务映射(以及最终部署)到相关设备有关的其它信息。当然,在这些例子中,设备存储库 420 一般可以包括静态或高级的设备元数据、和 / 或可以包括全局监视数据 136 和 / 或本地监视数据 138 的一些或全部。

[0095] 在运行时,如已经描述的,系统监视器 132 使用监视器服务 132a-132d 监视当前系统状态。服务的状态的任何部分是否以及如何暴露给系统监视器 132 可以由服务的开发者在设计时设置。随后,这种状态可获得性(state-availability)信息可以对于系统管理员和服务映射器 120 两者都是可获得的。如在上面也已描述过的,服务映射器 120 接收部署请求,然后例如通过将服务元数据与设备元数据进行匹配来确定相应的服务应当部署到哪个设备,所述设备元数据可以包括智能物件设备和相关的本地网络的当前状态。如在这里已经描述过的,服务映射器 120 还可以对包括(由系统监视器 132 认识到的)网络状态改变的特定事件或条件做出反应,并且可以在之后决定重新映射服务或者添加或去除服务的实例,以更好的满足给定的部署请求 / 要求。

[0096] 业务处理桥接层 406 包括这样的服务:该服务被设计为聚集经由设备级服务层 404 提供的来自在设备层 402 的设备的数据,并且将来自设备层的数据转换成业务相关信息。通过这样做,可以减少发送到后端企业应用系统的数据量,并且可以针对不同的企业应用系统来执行业务逻辑。

[0097] 例如,可以使用一个或多个规则处理器 422 来分析到来的消息,支持基本操作服务(例如物件移动、关联、去关联或设备读 / 写)以及支持信息查询。规则处理器 422 处理定义或引用(reference)了应当执行或参考(consulted)的任何其它基本操作服务的用户定义的业务规则。通过使用这样的规则和基本操作服务,提供了灵活的框架来使系统 400 适应不同的业务场境。

[0098] 规则处理器 422 可以使用数据储存库(data repository)424 以用于保持对所有感兴趣的物理对象的跟踪,例如,用于保持对给定被跟踪对象的当前状态、位置、时间戳或相关业务事务的跟踪,以及用于保持对预期将来会有什么动作的跟踪。可以在有规则的基础上,例如每天或每月,报告从数据储存库 424 聚集的信息。

[0099] 层 402、404 和 406 的操作的一个例子包括“货物接收”场境(scenario)。例如,向接收者交付对象的供应商可以发送预先装运通知(Advanced ShippingNotice,“ASN”)以及诸如电子产品代码(EPC)的对象标识符,所述 ASN 包含装运的货物中的所有对象的列表。ASN 可以存储在数据储存库 424 中。当装运的货物到达并通过例如位于接收平台(dock)大门处的设备层 402 的 RFID 读取器时,RFID 读取器读取该 EPC,并将其发送到规则处理器 422。规则处理器 422 查找该消息所来自的读取器的 ID,确定该读取器的位置和角色,然后调用负责处理所接收的装运货物的适当的基本操作服务。这个操作服务将所获得的 EPC 与来自之前的 ASN 的期望的 EPC 进行比较,并且,如果发现匹配,则向企业应用 432 报告交付已经到达并已完成。然后,所执行的操作服务还可以更新数据储存库 424 中的数据。上面描述的服务以及用于接收和发送所涉及的消息的服务可以由服务管理器 426 来管理。

[0100] 系统连通层 408 中的组件可以用来连接不同的应用系统,并支持系统和数据整合(integrate)。例如,通过信息交换和变换模块 428,消息和数据可以被路由到正确的后端系统,还可以被转换,从而使能语义正确的整合。除了消息路由和转换服务之外,系统连通性

层 408 还提供服务储存库 430 中能够由企业应用 432 用来访问由业务桥接层 406 提供的功能性的服务。

[0101] 企业应用层 410 包括例如负责控制和管理企业业务应用的传统的企业 IT 系统。覆盖特定业务过程的企业应用 432 可以不是单个程序,而是可以由不同的服务组成,所述不同的服务一起工作以实现期望的功能性。可以由同一企业系统、企业应用层 410 内的另一个企业系统(可能位于业务伙伴所在地)、或来自下层的系统(例如在设备层 402 的智能物件设备)来提供这样的服务。

[0102] 最后,在图 4 中,开发工具 434 可以指用于创建企业应用 432 和其它应用 / 服务的工具。使用与基础设备 400 集成的开发环境可以支持以与企业应用空间中已知的开发工具类似的方式实现基本服务。并且,开发工具 434 可以允许创建要求的服务元数据 126,并且允许将现有的服务包括到新的应用中。此外,开发工具 434 允许开发者指定特定服务应当在哪里运行、配置各个服务实例、以及以期望的方式部署服务。也就是说,开发者可以使用开发工具 434 开发服务元数据 / 可执行 436,然后可以提供期望的服务元数据 / 可执行 436,以便存储在服务储存库 124 中,并且 / 或者在同一时间或稍后的时间由服务映射器 120 进行映射 / 重新映射。

[0103] 以上已经描述了与图 1 的系统 100 相关联的示例实现方式和设置,图 5 是示出用于使用图 1 的系统 100 采集监视数据的示例算法的时序图。图 5 的算法使用全局监视器服务 132a 来实现,如已经描述过的,全局监视器服务 132a 被配置成采集关于其下面的层级内的所有设备和服务的信息,从而提供全局的系统全景视图。在图 4 的上下文中,全局监视器服务 132a 可以作为图 4 的基础设施 400 的设备级服务层 404 和 / 或业务处理桥接层 406 的一部分来运行。全局监视器服务可以由同样在设备级服务层 404 或业务处理桥接层 406 运行的高级服务请求 / 调用,或者通过分离的设备和服务监视图形用户接口 (GUI) 请求 / 调用。

[0104] 如上面提到的,本地层可以包括多个本地监视服务 132b,其每个被配置成使用例如轮询和信跳技术从边缘设备采集监视数据,这将在下面具体描述。本地监视器服务 132b 被配置成管理所收集的监视数据,包括例如过滤、聚集、和存储所述监视数据。组层包括多个位于或靠近基础设施(例如图 4 的基础设施 400)的边缘的设备,它们担任组长的角色,诸如图 1 的设备 108。组长采集关于在它们附近安装了边缘监视服务 132d 的边缘设备(例如设备 110-114)的信息。在图 5 的例子中,由组长(例如 108)获得的监视数据不存储在组长处,而是可以在转发给本地监视器服务 132b 以存储之前,对其进行过滤、聚集或其它处理。最后,设备(例如设备 110-114)运行边缘监视器服务 132d 并形成边缘层。在一些实现方式中,组长监视器服务 132c 和边缘监视器服务 132d 可能在提供、预处理和不存储监视数据方面本质上相同,尽管边缘监视器服务 132d 不负责管理其它设备。在图 5 的例子中,边缘监视器服务 132d 被配置成提供监视数据并使用信跳消息(例如以某些预定义的间隔使用主动发送队列发送的消息)将其发送到它的组长。然而,更一般地,从边缘层设备采集监视数据可以由发送信跳消息的边缘设备的定时器触发,或者由在边缘设备从上级 / 层接收的轮询消息触发。

[0105] 在图 5 的例子中,定时器触发本地监视器服务 132b 检查本地监视数据 138 中过时的数据,这导致轮询(502)几个边缘层设备。本地监视器服务 132b 不直接与边缘层设备通

信,而是使用位于网关处的组长设备 108,因此向组长监视器服务发送轮询消息 (504)。组长监视器服务 132c 将单个轮询消息分成不同的轮询消息(尽管图 5 为了简单仅示出了一个轮询消息),这些不同的轮询消息被发送到各个边缘层设备 110-114(506)。

[0106] 每个边缘层设备 110-114 读取本地系统适配器 (508) 以获得最新的 (up-to-date) 监视数据,该本地系统适配器应当被理解成与图 1 的系统适配器 134 相同或相似。系统适配器和相关元件的例子将在下面参照图 9-11 更详细地描述。但是,一般来说,应当理解这种系统适配器用作数据源,以提供关于它们的各个边缘设备的监视数据以及关于任何在它们的各个边缘设备上运行的服务的监视数据。

[0107] 然后,边缘监视器服务 132d 执行对监视数据的预处理 (510)(例如过滤和 / 或聚集)。然后,不存储经过预处理的监视数据,而是向组长监视器服务 132c 发送带有经预处理的监视数据的信跳消息 (512)。组长监视器服务 132c 接收来自多个边缘设备的带有监视数据的消息,并提供进一步的预处理 (514),包括例如计算所有被管理的边缘设备的 CPU 利用率的平均值。然后,组长监视器服务 132c 执行消息聚集 (516),其中,来自不同设备的消息被概 括成单个信跳消息,以便发送到本地监视器服务 132b (518)。

[0108] 本地监视器服务 132b 从多个组长监视器服务 132c 接收这样的信跳消息,并执行预处理 (520),包括例如过滤非临界 (non-critical) 数据,和 / 或聚集非临界数据。然后存储经预处理的消息 (522),例如,存储在本地监视数据 138。然后,本地监视器服务 132b 的定时器触发信跳指示器 (indicator) (524),其导致本地监视器服务 132b 计算本地监视数据 138 与全局监视数据 136 之间的增量同步 (delta synchronization) (526)。即,本地监视器服务 132b 确定作为执行图 5 的监视算法的结果,本地监视数据 138 中的哪些相关信息已经被改变或更新,并且只有所述被改变的数据被例如作为增量同步消息的一部分提交给全局监视数据 136 (528)。通过这种方式,减少了传输和处理的数据量。在附加和替代的实现方式中,增量同步可以由从全局监视器服务 132a 到本地监视器服务 132b 的消息来启动。

[0109] 最后,在图 5 中,更新的数据由全局监视器服务 132a 进行预处理 (530),并且被存储在全局监视数据 136 中。然后,业务数据处理系统 104,例如供应链管理应用,可以从全局监视数据 136 访问数据。从上面的描述应当理解,在从边缘设备本身到全局监视器服务 132a 的路径上的全景树中存储了来自给定边缘设备的监视数据。结果,可以容易的定位监视数据以便使用或对其进行删除。

[0110] 图 6 是可以表示图 1 的监视器服务 132a-132d 中的一个或多个的模块化监视器服务 600 的框图。也就是说,模块化监视器服务 600 包括核心监视器服务 148,在图 1 中,在组长监视器服务 132c 的上下文中示出了该核心监视器服务 148,但是如上面所解释的,该核心监视器服务 148 也可以在剩余的监视器服务 132a、132b 和 / 或 132d 中的一些或全部的上下文中实现。模块化监视器服务 600 还包括监视器服务模块 150,同样,在图 1 中,该监视器服务模块 150 在组长监视器服务 132c 的上下文中示出,但是其也可以在剩余的监视器服务 132a、132b 和 / 或 132d 中的一些或全部的上下文中实现。

[0111] 因此,核心监视器服务 148 可以被配置成实现可以由监视器服务 132a-132d 中的大多数或全部使用的基本功能性(例如,查询路由、基于角色的数据处理、以及协调与在其它设备上运行的其它监视器服务的通信),而监视器服务模块 150 可以被配置成实现可能

更具体的、针对监视器服务 132a-132d 中的给定的一个的功能性。因此，监视器服务模块 150 可以被实现 为根据需要添加到核心监视器服务 148 的插件组件，从而核心监视器服务 148 可以根据每个监视器服务和 / 或设备的个体需要和能力而专门化。

[0112] 例如，在图 6 的例子中，监视器服务模块 150 包括系统适配器 134、通信适配器 602、数据存储模块 604 和数据预处理器 606。结果，无需改变核心监视器服务 148 就可以将这些插件的相关功能性或其它功能性添加或替换到模块化监视器服务 600。而且，如在下面将具体描述的，相同种类 (category) 的监视器服务模块 150 (例如通信适配器 602 的不同实例或数据存储模块 604 的不同实例) 可以共享公共的接口，从而使得核心监视器服务 148 无需知道不同实例 / 插件的使用率。

[0113] 此外，在图 6 中，模块管理器 608 可以用于管理相同种类的模块。例如，模块管理器 608 可以包括用于系统适配器 134 的管理器，并且可以被配置成根据核心监视器服务 148 的当前情况和 / 或需要选择和实现系统适配器 134 中适当的一个。下面将会参照图 9 和 10 提供这种系统适配器管理器的特定例子，还会讨论模块管理器 608 的其它例子。应当理解，在一些实现方式中，模块管理器可以用作在核心监视器服务 148 与给定的监视器服务模块 150 之间的消息的实际中介 (intermediary)，例如，将消息从核心监视器服务 148 转发到适当的监视器服务模块 150，或者相反。在其它的实现方式中，模块管理器 608 可以被配置成加载或选择监视器服务模块 150 中适当的一个，然后可以将其自己从进一步的通信中去除。

[0114] 图 7A 是示出使用模块化监视器服务 600 的系统 100 的操作的流程图。在图 7A 中，核心监视器服务 148 的实例被部署到例如系统 100 的架构级中的一个或多个 (702)。例如，服务映射器 120 可以被配置成可能在系统 100 的最初建立 (set-up) 期间将核心监视器服务 148 部署到系统 100 的设备 108-114、146 和 144 上。在另外的例子中，将在下面参照图 7B 和图 15-17 具体讨论的，服务映射器 120 可以将核心监视器服务 148 的实例映射到进入传感器网络 102 的区域且希望与设备 108-114、146 和 / 或 144 中的一个或多个通信的新设备 (图 1 中未示出)。

[0115] 应当理解，核心监视器服务 148 可以作为模板、表格 (form)、或其它类属结构存储在服务储存库 124 中，并且可以在系统 100 内部署核心监视器服务 148 的不同实例。此外，实例不需要精确地相同。例如，可以取决于实例被部署到系统监视器 132 的哪一层而部署包括有或使能了不同功能的核心监视器服务 148 的不同实例。服务映射器 120 可以例如根据设备元数据 130 来执行上述核心监视器服务 148 的实例的服务映射。

[0116] 一旦在系统 100 内安装了核心监视器服务 148，不同的监视器服务模块就可以被安装 (704)。例如，可以例如取决于与所讨论的设备相关联的设备元数据 130、和 / 或取决于监视器服务模块被部署到监视器架构的哪一层级上，来安装监视器服务模块 134、602、604 和 / 或 606。同样，正如下面将更加具体地讨论的，这种设备元数据可以使用已经安装的核心监视器服务 148 来获得，并且 / 或者可以从存储器获得。

[0117] 利用核心监视器服务 148 和位于适当位置的监视器服务模块，可以例如通过使用图 6 的模块化监视器系统 600 实现图 5 的算法来收集监视数据。例如，从上面对于图 5 的讨论应当理解，最初，可以使用核心监视器服务 148 和 / 和系统适配器 134 来收集原始监视数据 (例如在边缘监视器服务 132d) (706) (如例如此处参照图 9-11 更具体地讨论的，其可

以包括服务和设备适配器两者)。通过这种方式,(例如关于相关服务和 / 或设备的)原始监视数据可以隐藏了(异类)服务 / 设备的不同接口、并且可扩展以用于新的服务 / 设备的方式来提供。此外,可以实现诸如例如 OSGi 或 JSR82 的系统标准。

[0118] 可以在预处理器模块 606 对原始监视数据进行预处理 (708)。结合图 5 并且进一步参照图 14 来提供这种预处理的例子。然而,一般来说,这种预处理作用于到来的数据,以便通过使用例如过滤或聚集来减少将要通过网络连接发送的数据量。

[0119] 可以使用数据存储模块 604 存储(经预处理的)监视数据 (710)。例如,数据存储模块 604 可以被配置成隐藏可获得的不同类型的存储器(例如文件系统、数据库或存储器内 (in-memory) 数据结构)。如上所述,监视数据的存储可以被刚好限制在本地监视器服务 132b 和全局监视器服务 132a,从而由此使得数据存储模块 604 可以仅仅被部署到那些监视器服务上,并且可以节约资源。

[0120] 可以传播经处理和存储的数据,并且可以使用通信适配器 602 来执行对于新数据的查询(例如通过轮询)(712)。例如,应当理解到,通信适配器 602 可以被配置成保证服务 / 设备之间的通信,可以隐藏不可靠的或无次序 (out-of-order) 的数据传输,并且可以隐藏核心监视器服务 148 使其免受不同通信协议 / 硬件的影响。在一些实现方式中,通信适配器 602 可以被看作是与监视服务 132a-132d 分离的,并且例如可以与核心监视器服务 148 或监视器服务模块 134、602 或 606 分离地(即在分离的时间)部署。下面会参照图 8、12 和 13 的例子来提供通信适配器 602 的不同操作和使用。

[0121] 图 7B 是示出系统 100 的附加实例操作的流程图。在图 7B 中,假设最初没有安装核心监视器服务 148。然而,在许多情况中,可能需要核心监视器服务 148,以便进行系统 100 的进一步操作。例如,在系统 100 内新设备的注册过程期间,注册协议可能要求核心监视器服务 148 的安装和可用性(见例如讨论用于向系统 100 注册新设备的示例注册协议的图 15-17 的例子)。在这个以及其它场境中,可以实现自举 (bootstrapping) 技术,其中,首先安装核心监视器服务 148,然后来自被安装的(执行的)核心监视器服务 148 的信息本身可用来注册新设备,和 / 或部署监视器服务模块 134 和 / 或 602-606。

[0122] 例如,可以首先确定所讨论的设备是否使用支持现有的用于部署服务的标准 (714)(例如 OSGi、Jini、Services and Networks for Domotic Applications (SENDA)、服务管理模块 (SMM) 或 JSR232)。如果所讨论的设备不支持这些标准,则可以采取适当的动作。例如,在所述设备是移动电话或 PDA 层级时,可以指示其用户执行用于建立核心监视器服务 148 的安装向导(例如,按照链接来下载并启动安装程序)(716)。在这种情况下,和 / 或在使用已知的部署标准的情况下,并且如果核心监视器服务 148 仍未安装 (718),则可以例如使用标准和 / 或从存储器确定静态服务元数据 (720)。

[0123] 此时,可以安装核心监视器服务 (722),并且可以从其获得附加的监视数据 (724)。如下面将例如参照图 15-17 更加具体地描述的,如果需要,全面的 (full-scale) 向系统监视器 132 注册设备也可以在此时执行。使用附加的监视数据,可以部署 (726) 适当的监视器服务模块 150,从而可以进行对附加监视数据的收集和 / 或传输。

[0124] 图 8 是核心监视器服务 148 的示例实现方式的框图。在图 8 的例子中,核心监视器服务 148 可以使用具有到来的 802a 或输出的 (outgoing) 802b 的消息的通信抽象 (abstraction) 层来例如建立到不同设备上的监视器服务的连接,或者建立到 GUI 接口和

其它类型的服务的连接。下面会参照图 12 更具体地讨论通信抽象层 802a 和 802b。从图 6 和 7 应当很明显地看到,其它模块管理器 150(例如参照图 9-11 描述的统适配器管理器)也可以与核心监视器服务 148 通信,并且,取决于所述管理器是如何定义和实现的,其也可以与 通信抽象层 802a、802b 的一些功能性重叠。

[0125] 图 8 的核心监视器服务 148 包括查询路由器 804。查询路由器 804 可以负责例如采集和返回被查询的监视数据,可以由在其它设备上运行的(其它)监视器服务 132a-132d 或者由其它服务(例如温度检测服务、物件跟踪服务或其它数据收集服务)来查询监视数据。

[0126] 在核心监视器服务 148 的图 8 的例子中包括并激活系统全景数据库 806(例如,核心监视器服务 148 可以与全局监视器服务 132a 或本地监视器服务 132b 相关联,在这种情况下系统全景数据库 806 可以表示、包括、或者以其它方式分别与全局监视数据 136 或本地监视数据 138 相关联)。在这种情况下,查询路由器 804 可以尝试从系统全景数据库 806 获得所请求的监视数据。

[0127] 如果期望的监视数据或其它期望的数据过时或没有找到,则查询监视器 806 可以从在父代设备上运行的监视器服务查询期望的监视数据,所述父代设备例如用于管理实现核心监视器服务 148 的设备的不同的设备(例如,组长监视器服务 132c 可以查询本地监视器服务 132b,或者本地监视器服务 132b 可以查询全局监视器服务 132a)。在另一种情况下,当前监视器服务 148 可以管理被查询的系统(即,是被查询系统的父代),在这种情况下,将寻址较低层级的监视器服务以收集期望的监视数据。在另一个场境中,对于图 8 的核心监视器服务 148 来说被查询设备可以是未知的,在这种情况下,查询可以被转发到管理更多系统的上层,并且最终可以前进到全局监视器服务 132a,然后全局监视器服务 132a 可以将该查询转发到适当的本地监视器服务 132b。

[0128] 如果可以直接寻址到目标系统而无需居间的系统,则查询路由可以是直接的(straightforward)。在不存在到目标系统的直接连接的情况下,则要确定在路由路径上的居间跳(hop)。描述这种居间跳的路由信息可以从所存储的拓扑信息(其可以存储在系统全景数据库 806 中,或者存储在分离的路由表中)获取。

[0129] 例如,每个监视器服务 132b-132d 都配备有在监视器服务的注册期间分配的父代监视器服务(例如见下面的图 15-17)。这种子代 - 父代信息可以存储在系统全景数据库 806 中,和 / 或存储在其它存储器中,以作为属性,根据所述属性可以生成树型拓扑。可以遍历该树型拓扑以确定下一跳。在其它的实现方式中,下一跳 / 居间跳可以使用适当的表查找从路由表获得。一般来说,这种分离的路由表可能比从树型拓扑确定路由信息使用更多的存储器,但是 其可能更加有效和 / 或快速。

[0130] 只有在设备与系统 100 连接或断开时才可以改变路由表中的条目,以便减少需要改变的次数。为了减少路由表中条目的数目,只有在下一跳不是父代监视器服务时才可以添加下一跳,这是因为可以通过默认的方式假设下一跳是父代监视器服务。通过这种方式,可以将路由表中条目的数目减少为由监视器服务所管理 / 监视的、以及直接连接到监视器服务的系统的数目。

[0131] 下面的代码段 1 示出了用于查询路由的算法,其实现了刚刚描述的查询路由方针。

```
[0132] Query landscape :get the queried data
[0133] If data not found
[0134]     if information is requested about the current device
[0135]         get system adaptor to service or device
[0136]         get information from adaptor
[0137] else(information is not about the current device)
[0138]     Query landscape :is device managed by this device ?
[0139]     If device is managed by this device
[0140]         get child device to ask
[0141]         get data from child device by asking the monitor on lower
tier
[0142]     else(device is not managed by this device)
[0143]         get parent device to ask
[0144]         get data from parent device by asking the monitor on
higher tier
[0145] if data still not found
[0146]     return error
[0147] else(data found)
[0148]     return data
[0149] 代码段 1
```

[0150] 一般来说,如已经描述过的,代码段 1 是与这样的查询算法一致的:其中,在(当前)设备处接收到对于监视数据的请求。如果监视数据与当前设备相关,但却不在系统全景数据库 806 中,则如这里所描述的,可以使用系统适配器 134 来获得监视数据。否则,监视数据与另一个服务 / 设备有关,并且不在系统全景数据库 806 中,则会适当地(as appropriate) 查询父代或子代监视器服务。如果在系统全景数据库 806 中或父代 / 子代监视器服务中发现所述监视数据,则可以检查该监视数据,看其是否是最新的。如果不是,则可以执行附加的轮询 / 查询来获得更新的监视数据。

[0151] 核心监视器服务 148 还包括查验(ping) 请求器 808,其被配置成由全局监视器服务 132a 和 / 或本地监视器服务 132b 使用,或者由任何在某些类型的监视数据存储器(例如系统全景数据库 806) 中存储监视数据的监视器服务使用。因此,查验请求器 808 可以被配置成查验系统全景中的设备,由此帮助保持子代系统的系统全景数据是最新的,并帮助检测断开的设备(并将数据标记为“怀疑”(in doubt))。如果监视器服务不在本地监视数据存储器中存储状态信息,例如组长监视器服务 132c 或边缘监视器服务 132d,则可以省略或不激活查验请求器 808。

[0152] 查验请求器 808 可以由定时器激活,作为响应,其可以从监视数据存储器(例如从系统全景数据库 806) 确定所有连接的和被管理的子代系统。例如,对于所有系统,查验请求器 808 将接收到上一个生命信号(life-signal) 的时间与当前时间进行比较,以便检测过时的监视数据(和其它被存储的信息)。如果过时,则可以将系统的状态改变成指示这种过时,例如,系统的状态可以从“有效”(alive) 变成“过时”。如果时间戳比特定切断

(cut-off) 点久，则状态可以从“过时”变成“怀疑”，从而指示所讨论的设备有可能断开。对于所有“过时”的系统，查验请求器 808 都发送查验请求。查验请求器 808 接收查验响应并由此更新“上次见到的”(last-seen) 时间戳，然后在系统全景数据库 806 中将相关状态改回“有效”。

[0153] 在下面的代码段 2 的上下文中提供了与上面对查验请求器 808 的解释相一致的算法：

```
[0154] Landscape query :get all systems and their status
[0155] For all systems do
[0156]     If the timestamp is old enough to mark the system as "in-doubt"
[0157]         then
[0158]             Mark all related data as "in doubt"
[0159]     Else if it is old enough to mark the system as "outdated" then do so
[0160]     If the status is "outdated" then
[0161]         Send a ping signal to this system
```

[0162] 代码段 2

[0163] 尽管就查验请求器 808 对图 8 进行了讨论，但是应当理解，也可以使用其它的技术来确定系统 100 的系统全景的要素的状态，例如，使用其它的回送 (echo) 请求来测试系统的可到达性和 / 或测试网络延迟。此外，应当理解，如图所示，核心监视器服务 148 也可以接收然后响应于来自其它监视器服务的查验请求。如将在下面参照图 12 所讨论的那样，保证对查验（或其它）状态请求的可靠数据传输的传输协议的实际实现方式可以被实现为通信适配器 602 的任务。

[0164] 核心监视器服务 148 还包括信跳发送器 810，其可以被实现为这样的服务：其被配置成向父代监视器服务发送信跳消息以指示（在其上部署了核心监视器服务 148 的）设备仍在运行。信跳发送器 810 可以被定时器触发，其中，可以在相关联的信跳设置中定义两个消息之间的时间间隔。可以考虑资源约束来选择所述间隔，以便具有有限电源和网络能力的小型设备可以使用比更强大的设备（例如连接良好 (well-connected) 的服务器）更长的时间间隔。可以使用公制 (metric) 来定义为每个设备类型（例如为传感器设备、PDA 和膝上型计算机）定制的默认间隔。此外，可以由用户，例如图 4 的基础设施的系统管理员来调整该间隔的值以及其它相关设置。

[0165] 除了所讨论的设备有效且被连接的信息之外，信跳发送器 810 还可以被配置成也发送至少一些监视数据，如果在核心监视器服务 148 接收到来自另一个监视器服务的所述监视数据，该监视数据可以用来更新系统全景数据库 806。信跳设置可以用来定义应当与信跳消息一起发送所述监视数据中的哪些。通过这样做，可以节约网络流量、功率和其它资源。此外，通过接收信跳消息中的监视数据，相对较高级的监视器服务可以避免执行至少一些对较低级服务的轮询，因为所述较高级的服务可能已经具有期望的监视数据。通过执行对监视数据存储器的查询或者通过读取系统适配器 134（见图 6 和 9-11），在其上部署了核心监视器服务 148 的设备可以包括附加监视数据以及信跳消息。

[0166] 例如如果两个连续的查验都丢失了，则查验请求器 808 可以响应于没有 到达的期望的信跳消息而发送附加的请求，从这方面来说，信跳消息并非必需通过可靠的通信信

道来发送。因此,即使通信信道不可靠,并且信跳消息丢失,核心监视器服务 148 也可以继续工作。因此,可以使用诸如例如轻便且无连接 (connection-less) 的用户数据报协议 (UDP) 的协议。

[0167] 在下面的代码段 3 的上下文中提供了与上面对于信跳发送器 810 的解释相一致的算法 :

```
[0168] Read heartbeat settings  
[0169] If time interval has changed then  
[0170]     Update timer component  
[0171]     Get all data that should be included in the heartbeat message  
[0172]     Send heartbeat message to parent with all queried data  
[0173]     Sleep according to heartbeat settings  
[0174] 代码段 3
```

[0175] 查验请求器 808 可以被理解为信跳发送器 810 的附加部件。例如,如上面所提到的,在一些连续丢失信跳消息的事件中,查验请求器 808 可以构建可靠的连接以保证查验被接收和处理。此外,如果子代设备不具有 (在起作用的) 信跳发送器 810,则父代监视器的查验请求器 808 可以完全取代信跳发送器 810。结果,子代系统并非必需具有已安装的信跳发送器,从而可以使用更少的硬件能力。

[0176] 核心监视器服务 148 还包括更新通知发送器 812,其可以被配置成向服务提供可以例如从系统全景数据库 806 确定的关于系统全景的更新,诸如例如监视数据的变化、或者新设备的注册。因此,更新通知发送器 812 可以向存储在接收者注册表 (registry) 814 中的所有已注册的接收者发出通知消息。

[0177] 注册到更新通知发送器 812 的服务可以明确地指定期望的数据,以便仅获得期望的数据 (并最小化其它传输)。对于发送什么 (类型的) 监视数据更新的约束可以基于例如以下属性 :数据提供服务的名称、或者设备和数据字段的名称 (例如有效状态、CPU 使用率、RAM 大小、或连通性值)。因此,可以用下面的信息来存储用于接收更新的订阅 (subscription) 规则 :接收系统的标识符、以及约束规则 (例如源系统标识符和 / 或数据字段名称)。在一些实现方式中,可以用 AND (与) 连接两个限制条件,从而例如为了发送更新,系统的名称与数据字段的名称两者都必需匹配。在其它实现方式中,可以通过定义多个订阅规则来实现 OR (或) 规则。

[0178] 在下面的代码段 4 的上下文中提供了与上面对于更新通知发送器 812 的解释相一致的算法 :

```
[0179] Query recipient database :get all recipients that are registered for  
this service
```

```
[0180]             or device and data field  
[0181] For all found recipients do  
[0182]     Send update trigger message  
[0183] 代码段 4
```

[0184] 图 9 是示出用于与图 6 的模块化监视器服务 600 和 / 或图 8 示出的核心监视器服务 148 的实现一起使用的系统适配器管理器 902 的框图。具体来说,系统适配器管理器

902 可以用于管理系统适配器 134。如上面提到过的, 系统适配器 134 可以被配置成提供核心监视器服务 148 与提供监视数据的各种服务和设备 (例如服务 122 和 / 或设备 108) 之间的连接。因此, 应当理解, 由于系统全景中异质性 (heterogeneity) 的可能性, 服务、设备和监视数据可能非常显著地变化。因此, 系统适配器 134 可以实现用于与核心监视器服务 148 通信的公用接口, 同时实现监视 (并且, 潜在地控制) 可能存在的各种异质的服务和 / 或设备所需的任何其它接口, 从而使核心监视器服务 148 为了与大量的类型的系统适配器 134 通信 (以及由此与大量的类型的服务和设备通信), 仅仅需要知道该公共接口。

[0185] 由于核心监视器服务 148 可能能够 (仅仅使用公共接口) 与大量的系统适配器 134 通信, 因此核心监视器服务 148 不确定或不能确定哪个系统适配器 134 适合给定的任务。于是, 在确定用于给定任务或设置的特定的 / 正确的系统适配器 134 时, 使用系统适配器管理器 902。这样, 系统适配器管理器 902 代表模块管理器 150 的例子, 并且在核心监视器服务 148 与系统适配器 134 之间提供抽象层。因此, 可以免除核心监视器服务 148 跟踪、选择或以其它方式管理系统适配器 134 的不同实例或可能性的职责, 从而大大减少了服务或设备本身。

[0186] 如上面所提到的, 系统适配器 134 可以包括例如服务适配器和 / 或设备适配器 (见例如图 10), 从而可以执行核心监视器服务 148 (或其它监视器服务模块 150) 与给定服务 (例如服务 122) 或设备 (例如设备 108) 之间的通信, 并且可以获得相应的监视数据。例如, 系统适配器 134 可以被认为是提供到环境 (例如到服务 122 和 / 或到设备 108) 的接口的模块化监视器服务 600 的一部分。在设备 108 的情况中, 系统适配器 134 中给定的一个可以确定例如该设备的 CPU 利用率、存储器使用率或存储器大小。在服务 122 的情况中, 系统适配器 134 中给定的一个可以确定例如与服务 122 相关联的软件版本或软件厂商。

[0187] 因此, 应当理解, 可以不存在到环境 (例如到各种服务 / 设备) 的定义的标准接口, 因为每个这样的服务和设备可以具有其自己的特定接口。于是, 系统适配器 134 可以被设计和实现为适合所讨论的被监视系统 (例如系统 100)。这样, 监视服务 132a-132d 可以读取关于设备和服务的信息, 由系统适配器 134 和系统适配器管理器 902 向核心监视器服务 148 隐藏了所述设备和服务的不同接口, 从而系统适配器管理器 902 的组件提供了供核心监视器服务 148 使用的抽象层。

[0188] 系统适配器 134 的各种类型和实例可以在适配器注册表 904 中存储 (或索引 (referenced)), 从而使系统适配器管理器 902 可以确定 (例如基于核心监视器服务 148 的请求或基于管理员的设计选择) 在给定情况中使用哪个系统适配器 134。

[0189] 因此, 系统适配器管理器 902 可以被配置成加载和管理实现公共接口的多个系统适配器 134。在启动时, 系统适配器 134 可以在适配器管理器 902 处注册。系统适配器管理器 902 可以执行所有已注册适配器 134 的维护功能, 其中, 除了读取和返回监视数据 (例如关于当前运行的服务的信息、特定于设备的信息、或者由在其上运行了模块化监视器服务 600 的设备所收集的传感器数据) 之外, 系统适配器 134 的维护功能还可以实现内部管理活动。

[0190] 在图 9 的例子中, 系统适配器 134 包括基本信息适配器 906 和 OSGi 适配器 908。基本信息适配器 906 一般指可以例如在系统 100 中收集和使用的服务和 / 或设备信息, 诸如上面提供的设备元数据 130 的许多和不同的例子。OSGi 适配器 908 一般指这样的事

实 : 系统适配器 134 可以支持诸如 OSGi 的标准化协议, 以获得关于没有安装监视器服务 132a-132d 之一的设备的信息。这种协议标准的其它的例子包括 Java API 平台 JSR232(其包括 OSGi) 以及 Java API for Bluetooth, JSR82。

[0191] 图 10 是图 1、6 和 9 的系统适配器的示例实现方式的框图。在图 10 的例子中, 示出了上级 (例如全局或本地层) 设备 144、146, 其可以包括例如个人计算机或服务器计算机。设备 144、146 包括核心监视器服务 148a 和 / 或 148b, 以及两个系统适配器 : 服务适配器 1006(用于与多个服务 1004 接口连接) 和设备适配器 1006(用于与多个设备硬件 1008 的实例 / 类型接口连接)。如图所示, 核心监视器服务 148a、148b 还可以与预先注册的设备 1010 通信, 该设备 1010 可以被配置成直接与核心监视器服务 148 接口连接。

[0192] 进一步, 在图 10 中, 低级 (例如组长或边缘) 设备 108、110、112 和 / 或 114 可以表示资源相对更加贫乏的设备, 并且可以包括核心监视器服务 148c、148d。类似于设备 144、146, 设备 108、110、112、114 包括服务适配器 1012(用于与多个服务 1014 接口连接) 和设备适配器 1016(用于与设备硬件 1018 的多个实例 / 类型接口连接)。设备 108、110、112、114 还包括分离的设备适配器 1020, 其可以用于与实现了标准 1022 的例如设备 1024 的设备通信。例如, 设备 1024 可以表示运行了管理标准 1022(例如 OSGi) 的传感器微片或蜂窝电话, 从而使得核心监视器服务 148 可以使用标准 1022 与其通信。

[0193] 图 11 是图 1、6 和 9 的系统适配器的另一个示例实现方式的框图。在图 11 的例子中, 系统适配器 134 被图示为与核心监视器服务 148 通信。如上所述, 系统适配器 134 可以使用核心监视器服务 148 的 API 1102 来与核心监视器服务 148 通信, 其中所述 API 1102 对于所有系统适配器 134 来说是公共的。系统适配器 134 还使用 API 1106 与服务和 / 或设备 1104 通信, 该 API 1106 特定于服务 / 设备 1104, 并且特定于与图 11 中的系统适配器 134 的特定的、相关的实例。

[0194] 在图 11 中, 系统适配器 134 包括通知系统 1108, 其被配置成从核心监视器服务 148(例如从图 8 的更新通知发送器 812) 接收通知。例如, 如在图 8 的上下文中解释过的, 核心监视器服务 148 可以发送关于监视数据的变化或者新服务或新设备的注册的更新。通知系统 1108(其因此可被理解为包括在接收者注册表 814 中) 可以订阅特定的通知或者通知类型, 从而可以以周期性的间隔或预定的间隔接收通知。

[0195] 通知系统 1008 还可以从其它服务, 诸如例如业务逻辑规则引擎服务 1110(其实现业务规则并且可以触发对特定真实事件的反应) 或记录器 (logger) 或监视定时器 (watchdog) 服务 (图 11 中未示出) 的其它服务接收通知。取决于系统适配器 134 的实现方式, 通知可以被转发或处理, 并且系统适配器 134 定义和实现对每个可能的通知的适当的反应。系统适配器 134 还包括系统读取器 1112, 其被配置成从服务 / 设备 1104 读取监视数据, 诸如例如 CPU 利用率、电池寿命和服务的状态 (例如服务是正在运行还是被停止)。然后, 系统读取器 1112 可以将监视数据提供给正在轮询系统适配器 134 的核心监视器服务 148。

[0196] 在这些以及其它实现方式中, 系统读取器 1112 可以向信跳发送器 1114 提供监视数据。一般来说, 信跳发送器 1114 被配置成周期性地向核心监视器服务 148 发送生命信号。正如刚刚提到的, 并且在上面参照图 8 提到过的, 信跳发送器 1014 还可以接收和发送来自服务 / 设备 1104 的监视数据 (例如状态值), 这允许减少消息发送 (messaging) 开销。如

参照图 8 描述过的,信跳设置 1116 可以包括与信跳发送器 1114 相关联的参数,诸如例如信跳消息之间的时间间隔。

[0197] 最后,在图 11 中,初始化器 (initializer) 1118 被配置成建立信跳设置 1116 并且在监视数据存储器 (例如图 8 的系统全景数据库 806) 注册监视数据。初始化器 1118 还可以被配置成针对在更新通知发送器 812 处的更新通知消息向接收者注册表 814 进行注册。

[0198] 在这里描述的例子中,处理线程可被用于各种任务,包括监视器服务器维护线程和适配器线程。这些线程 (用于例如上下文切换、线程调度和线程间通信) 可以根据需要被创建和管理。监视器服务线程可以用于保持存储在系统全景数据库 806 和 / 或全局 / 本地监视数据 136/138 中的监视数据是最新的,并且还可以用于管理低级服务和设备的状态。

[0199] 可以与监视器服务维护线程分离地实现系统适配器线程,从而使得系统适配器 134 不会由于执行复杂或时间密集的任务而阻塞 (block) 监视器服务维护线程。系统适配器 134 可以作为分离的线程运行,或者可以由相关联的监视器维护服务的线程来执行。在后一种情况下,监视器服务维护线程可能在执行系统适配器 134 的维护功能的时间内被阻塞。可以被监视器服务调用的系统适配器 134 仍然可以创建在后台进行耗时的工作的临时线程,从而使 监视器服务线程不被阻塞。

[0200] 一般使用其自己的线程的系统适配器可能更加强大。并且,这种系统适配器不阻塞监视器服务,因此可以执行耗时的任务和 / 或等待 I/O 中断。此外,这种系统适配器在决定何时被激活方面具有更多的自由。因此,这种系统适配器不依赖于在不确定的时间点由监视器服务进行的调用。定时器可被用来主动地发送信跳消息和更新全景值。

[0201] 尽管如此,对于有限的设备,将系统适配器实现为使用监视器服务线程的无源组件可能更加适当。在这种情况下,可以避免创建和管理线程的开销,其代价就是在执行所讨论的系统适配器期间阻塞监视器服务。

[0202] 图 12 是图 6 的模块化监视器服务 600 的通信适配器 602 的实例实现方式的框图。如图所示,通信适配器管理器 802a、802b 可以用作图 8 中与核心监视器服务 148 通信的通信抽象层。类似于系统适配器管理器 902,通信适配器管理器 802a、802b 管理用于设备的通信适配器 1202 并且与其通信 (以便与设备 1024 通信),并且,还与通信适配器 1206 通信 (以便与服务 1208 通信)。从上面对于系统适配器管理器 902 的描述应当理解,通信适配器管理器 802a、802b (通信抽象层) 可以被配置成加载 / 卸载通信适配器 1202、1206,以及注册 (或取消注册) 通信适配器 1202、1206。

[0203] 图 13 是图 6 的通信适配器 602 的另一个示例实现方式的框图。在图 13 中 (如图 12 那样),通信适配器管理器 (抽象层) 802a、802b 可以支持本地方法调用和网络通信两者。此外,通信适配器管理器 (抽象层) 802a、802b 可以操作以用于使通信协议和网络设备对于核心监视器服务 148 不透明。

[0204] 例如,消息处理和套接字 (socket) 通信可用于网络通信。此外,本地通信可以使用消息、API、套接字、管道 (pipe) 和 / 或共享存储器来实现。

[0205] 因此,通信适配器管理器 802a、802b (通信抽象层) 隐藏当前使用的可以在不同 ISO/OSI 层工作的、并且 / 或者可以组合的协议 (例如传输控制协议 (TCP)、统一数据报协议 (Uniform Datagram Protocol, UDP)、网际协议 (IP)、超文本传输协议 (HTTP) 或文件传输协议 (FTP))。通信适配器管理器 802a、802b (通信抽象层) 还可以隐藏网络设备 1204 (例

如蓝牙、调制解调器或 WiFi 设备)。通信适配器管理器 802a、802b(通信抽象层)还可以隐藏特定的通信概念(例如消息传递、套接字、共享存储器或 API 调用)。

[0206] 换言之,如已经针对系统适配器 134 描述过的,通信适配器管理器 802a、802b 可以加载或卸载例如 TCP 适配器 1302、UDP 适配器 1304、或 API 调用适配器 1306,它们每个都可以提供到核心监视器服务 148 的公共接口。因此,通过使得通信对于核心监视器服务 148 和其它服务不透明,图 12 和 13 的例子可以允许避免应付(deal with) 协议、网络设备、序列号或传输错误。

[0207] 在图 13 中,适配器 1302-1304 中的每一个都被标识为与网络(1302、1304)或本地(1306)通信相关联。此外,适配器 1302-1304 中的每一个都被标识为与可靠的(1302、1306)或不可靠的(1306)通信相关联。适配器 1302-1306 中的每一个还可以与唯一的标识符(适配器 ID)相关联,以供通信适配器管理器 802a、802b 在选择或加载各个适配器时使用。

[0208] 一般来说,不可靠的数据传输可以比可靠的传输简单得多,也因此比可靠的传输耗费的资源少。因此,在可能的情况下可以首选不可靠的数据传输。然而,一些功能性,诸如查验系统,可能要求可靠的数据传输(以及相应的通信适配器)。

[0209] 因此,一般来说,应当从图 8 的例子理解到,核心监视器服务 148 可以发送和接收不同类型的消息,并且通信适配器管理器 802a、802b 可以用于选择相应的、适当的通信适配器来执行所述发送和接收。例如,核心监视器服务 148 可能需要使用轻便的、不可靠的传输协议通过网络发送消息。在这个例子中,通信适配器管理器 802b 可以从所有加载的适配器中定位(locate)实现不可靠网络通信的 UDP 适配器 1304。然后,通信适配器管理器 802b 将消息转发给 UDP 适配器 1304,由其来执行实际的通信。

[0210] 图 14 是图 6 的预处理器模块 606 的示例实现方式的方框图。例如,各种监视器服务 132b-132d 中的一个或多个可以使用预处理器模块 606,以便在将经处理的监视数据转发给图 1 的系统 100 的架构中的下一个较高级之前在本地对监视数据进行处理。

[0211] 在图 14 中,预处理器管理器 1402 被配置为与上述的适配器管理器 802a/802b 和 902 相似地或类似地操作。例如,预处理器管理器 1402 可以与预处理器注册表 1040 接口连接,以确定可能的预处理器模块 606 中适当的一个,以便与核心监视器服务 148 一起使用。如图所示,预处理器管理器 1402 可以被配置成诸如从本地服务或设备、并且可能经由核心监视器服务 148 和适当的系统适配器 902 和 / 或通信适配器 802b 接收到来的监视数据。在由预处理器模块 606 预处理之后,预处理器管理器 1402 可以输出经预处理的监视数据,使其返回到核心监视器服务 148,以便发送到系统监视器架构中下一个较高级。

[0212] 例如,预处理器模块 606 可以包括过滤器 1406、聚集器(aggregator) 1408、以及更新通知发送器 1410。例如,过滤器 1406 可以被配置成去除特定类型的监视数据,或者特定量的监视数据。聚集器 1408 可以被配置成组合接收到的消息,以便发送组合的消息。例如,组长监视器服务 132c 可以从每个边缘设备 110-110 接收消息以将它们聚集,或者可以从单个边缘设备(例如边缘设备 110)接收一系列消息。

[0213] 如上所述,与更新通知发送器 812 相似,更新通知发送器 1410 可以被配置成提供更新。在图 14 的例子中,更新通知发送器 1410 将更新传输(communicate)给 GUI 1412,以供例如系统管理员查看。当然,GUI 1412 的例子可能在监视器架构的较高级最为适当。此外,除了聚集和过滤之外,记录、统计计算以及其它数据预处理器也可用于监视数据。

[0214] 例如,系统适配器 134 可以出于确定过去两分钟内的平均利用率的要求而每秒钟递送一次 CPU 利用率。为了支持可缩放性,监视的数据值可以在进行测量的系统适配器附近被预处理。通过这种方式可以节约通信资源,因为可以避免将每个数据值都发送(每秒一次)到上级的需要。相反,例如,根据需要,可以仅发送聚集的值。

[0215] 在操作中,可以在核心监视器服务 148 注册不同的预处理器模块。当监视数据(例如状态值)到来时,执行每个注册过的预处理器。例如,第一预处理器模块可以将原始监视数据作为参数接收,并且可以返回经预处理的数据组。随后,下一个预处理器被启动,并且从第一预处理器接收经预处理的输出数据。由此,每个预处理器都可以执行,以针对其前任(predecessor)的输出数据进行工作。

[0216] 在这些例子中,当使用这种模块(例如插件)的顺序执行时,可能需要对插件的执行进行适当的排序。例如,可能向核心监视器服务 148 和 / 或预处理器管理器 1402 注册了两个模块。在这个例子中,第一模块可以删除特定服务或设备的所有数据,而第二模块可以计算所有被监视设备的平均 CPU 利用率。显然,如果这两个模块的次序改变,则结果将会不同,因为那样的话所有被监视服务和设备都会被考虑,包括要删除的特定服务或设备。

[0217] 因此,预处理器管理器 1402 可以例如在注册期间设置适当的执行次序。在设计时可能不知道预处理器模块 606 的数目和类型,从这方面来说,可能很难分配绝对的执行次序,特别是当预处理器模块可以在运行时被动态加载的情况下。结果,一个选择是创建预处理器模块 606 的组,以便根据组来对预处理器模块 606 排序。每个预处理器模块 606 可以与组标识符相关联,以便在注册期间可以被放倒正确的组中。然后,如果不是在各个预处理器模块层级上依次执行组的话,可以在组层级上依次执行组。可以根据例如执行的预处理的优先级和 / 或类型来对组进行排序。例如,第一组可以包括必需针对原始数据进行工作的预处理器模块,第二组可以对已经经过预处理的数据进行工作,而第三组可以处理最终的数据。因此,用于预处理器模块 606(例如 1406-1410)的模块化或插件概念可以增加图 1 的系统 100 的可缩放性,并且可以解决资源约束的问题。例如,可以通过免除全局监视器服务 132a 和本地监视器服务 132b 的一些或全部预处理任务来增加可缩放性。此外,模块化架构允许监视器服务 132a-132b 可根据每个设备的需要和能力来配置。

[0218] 也可以实现其它类型的监视器服务模块 150。例如,如图 6 所示,可以使用数据存储模块 604。例如,取决于在其上安装了全局监视器服务 132a 和本地监视器服务 132b 的设备的能力,可以选择与所选择的用于在本地监视数据 136 中存储监视数据的技术不同的技术来用于在全局监视数据 138 中存储监视数据,例如,任何一个都可以将数据存储在数据库、文本文件、存储器内数据结构或几乎任何其它类型的数据存储器中的一个或多个中。数据存储模块 604 的使用可以使得这种差异对于使用了以上参照图 6-14 描述的各种技术的监视服务来说是不透明的。例如,可以用定义明确的接口来封装监视数据存储器,并且所述接口可以用作来自实际数据存储系统的抽象层。

[0219] 图 15 是图示用于与图 1 的系统 100 一起使用的注册协议的系统 1500 的方框图。在图 15 中,可以是便携式设备(例如 PDA 或膝上型计算机)的组长设备 1502 进入组长设备 1504 的区域。如图所示,组长设备 1504 是传感器网络的多个边缘设备 1506a、1506b、1506c 的组长。因此,例如,组长设备 1504 和边缘设备 1506a、1506b、1506c 可以包括在许多传感器网络应用中,包括例如库存管理或危险材料跟踪。

[0220] 因此,例如,组长设备 1504(如上所述,其可以例如与边缘设备 1506a、1506b、1506c 相同,或者可以是 WiFi 接入点或基站)可以处于收集与仓库中的库存相关联的信息(即监视数据)的过程中。例如,这种监视数据可以包括待销售的存货的类型和数量、以及 / 或者存货附近的当前温度信息(例如以检测事物产品变质的可能性)。因此,组长设备 1502 可以进入组长设备 1504 的区域,例如当雇员带着 PDA 或其它计算设备进入该区域并想要获得期望的监视数据时。

[0221] 在上面的例子以及类似的例子中,组长设备 1502 可能希望收集监视数据以供雇员在本地使用,和 / 或可能自己收集 / 发送监视数据到全局监视器服务 1508。举例来说,在后一种情况中,组长设备 1502 可能具有在组长设备 1504 中不可用(或者至少是当前不可用)的特定计算资源或能力。应当理解,在图 15 的例子中,组长设备 1502 可能实际上是当前用作多个边缘设备(在图 15 中未示出)的组长,或者可能在这样的意义上被称为组长,即,存在充足的计算资源以在此处描述的基础设施的组长级处动作(例如,分裂消息以分发到边缘设备、或者过滤 / 聚集来自边缘设备的消息以用于后续的传输)。在另一个例子中,情况可以是组长设备 1504 失效,或者即将失效(例如由于低电池功率),或者被要求重新定位,从而出于承担组长设备 1504 的责任的目的而重新定位组长设备 1502。

[0222] 在图 15 的例子中,全局监视器服务 1508 包括全局监视数据 1510,并且在设备 1512(如这里所描述的,其可以包括 PC 或类似强大的计算机)上运行。如已经描述过的,全局监视数据 1510 可以包括与系统 1500 整体相关的全局全景数据,包括图 15 中未示出的其它要素。例如,全局监视数据 1510 可以包括描述组长 1504 和边缘设备 1506a、1506b、1506c 的设备元数据(包括例如静态和动态设备能力两者)。

[0223] 此外,全局监视器服务 1508 可以包括与组长设备 1502 相关的设备元数据。例如,组长设备 1502 可能在之前已经注册到全局监视器服务 1508 以作为分离的网络(图 15 中未示出)的一部分。当操作组长设备 1502 的雇员从分离的网络重新定位到图 15 所示的设备的区域时,组长设备 1502 可以断开作为分离的网络的一部分(并且如已经描述过的,在全局监视数据 1510 中做出这样的标记),尽管关于该组长设备的信息可以那个在全局监视数据 1510 中保存某一段时间。

[0224] 系统 1500 还包括本地监视器服务 1514,其具有本地监视数据 1516 并且在设备 1518 上运行。如上所述,设备 1518 可以包括例如便携式计算机或 Stargate 计算机。此外,系统 1500 可以包括本地监视器服务 1520,其具有本地监视数据 1522,并且在设备 1524 上运行。

[0225] 然后,在图 15 的例子中,组长设备 1502 可能需要注册到全局监视器服务 1508 以便例如建立与组长 1504 的连接 1526。例如,可以是这样的情况:组长设备 1502 和组长设备 1504 可以具有重叠的能力来相互通信;例如,两者可以都具有 Bluetooth(蓝牙)能力。即便如此,组长设备 1502 可能不知道组长设备 1504 具有蓝牙能力,而且,组长服务 1502 可能不具有与蓝牙和任何正在组长设备 1504 上运行的服务(例如上面提到的温度报告服务)两者都兼容的本地通信服务(例如适配器 / 模块)。

[0226] 然而,相反,全局监视器服务 1508 可能非常清楚存储在全局监视数据 1510 中的所有这些信息(例如适当的服务元数据和 / 或设备元数据)。当然,从上面的描述应当理解,关于组长设备 1504 的信息也可以存储在本地监视数据 1516 中。因此,全局监视器服务

1508(间接地)能够建立连接 1526,因为例如全局监视器服务 1508 可能能够确定需要哪个通信服务(适配器和模块),然后可以将所述服务下载和 / 或注入到组长设备 1502 上。全局监视数据 1510 还可以与组长设备 1504 通信,以便保证(如果需要的话通过注入)组长设备 1504 不仅具有蓝牙(或其它通信协议 / 标准),而且还具有与蓝牙和组长设备 1502 上的相应服务两者都兼容的本地服务。

[0227] 通过这种方式可用建立连接 1526。在类似的情况下,组长 1502 可以用其它方式与系统 100 的其它设备 / 服务通信。因此,在系统 1500 中组长设备 1502 的注册可以例如为组长设备 1502 的用户提供多个优点。

[0228] 然而,这种注册可能与多个困难相关联。例如,这种注册可能表现出安全问题,因为未注册的以及可能恶意的设备一般不应当被监视并被整合到系统 1500 中。并且,频繁交换注册消息可能消耗宝贵的带宽、电池功率和其它通信资源。然而,如果设备没有注册,则系统 1500 的其它部分可能就不能识别且不能访问这样的设备。

[0229] 因此,下面的描述与图 15-17一起提供了组长设备 1502 可以如何注册到全局监视器服务 1508 的例子。当然,应当理解,这仅仅是一个例子,并且很明显地,许多其它的例子也可以用来实现所描述的注册协议或所描述的注册协议的变体。

[0230] 然后,在图 15 中,对于所述例子假设组长设备 1502 具有安装在其上的监视器服务(尽管在图 15 中未示出)。这个监视器服务可以包括此处描述的各种实现方式,包括图 6 的模块化监视器服务,或者可以包括非模块化的、例如集成的监视器服务。因此,组长设备 1502 及其安装的监视器服务尝试加入系统 1500 的网络,包括在全局监视器服务 1508 注册。

[0231] 组长设备 1502 包括注册系统 1528 和注册数据 1530。注册系统 1528 被配置成与本地监视器服务 1514、1520 中的一个或多个(和 / 或与全局监视器服务 1508)交换注册消息 1532、1534、1536 和 1538。注册数据包括关于组长设备 1502 的正在连接(connecting)的监视器服务的信息,诸如例如关于组长设备 1502 的当前状态的元数据、或者关于组长设备 1502 的设备元数据。注册数据还可以包括初始的传感器状态 / 值信息(例如,以避免会在注册之后立刻发生的针对这些值的初始轮询所造成的开销,因为数据是过时的或者不存在的。在注册数据中会包括过量的值的情况下,则可以至少发送静态数据值)。

[0232] 尽管在图 15 中注册数据 1530 被概念性地图示为存储器 / 在存储器中,但是应当理解,注册数据 1530 也可以简单地代表在注册发生期间或在注册发生之前由组长设备 1502(例如由注册系统 1528 和 / 或由组长设备的监视器服务)收集的值。当然,在一些示例实现方式中,组长设备 1502 可以具有存储器。此外,在本地监视器服务尝试注册到新的网络的情况下,从上面的描述应当理解,可以使用本地存储器(例如本地监视数据 1516、1522)。更一般地,应当理解注册系统 1528 和注册数据 1530 或者它们的变体可以在图 15 的其它设备和监视器服务中实现。

[0233] 所描述的注册协议可以被实现为简单的,例如无连接的健壮的协议。此外,注册协议可以被配置成支持主动注册(例如由正在连接的设备发起)和被动注册(例如由被连接的网络设备发起)两者。结果,如所描述的,注册过程可以由任何设备启动。

[0234] 然后,在图 15 中,组长设备 1502 作为未注册但现在正在注册的设备发出征求(solicitation)消息 1532。可用作为广播消息发送的征求消息 1532 可以包含对于同一层级或下一个较高层级的监视器服务的特定类型或其标识的请求,并且所述请求可以部分地

基于正在连接的设备的设备能力。例如,如果(正在注册的(registering))组长设备1502具有特定的能力(例如是移动电话),则它可以尝试定位其它的组长设备(例如组长设备1504,即它的监视器服务)和/或本地监视器服务(例如1514、1520),而不是全局监视器服务1508。

[0235] 如果没有发现首选的监视器服务类型的适当实例,则注册系统1528可以重新发送具有下一个较高层级的标识的征求消息1532(从较低层级监视器服务到较高层级监视器服务上传(go up))。此外,如果征求消息1532丢失,则注册系统1528可以简单地例如在超时之后发送另一个征求消息。被接收两次的征求消息不是问题,因为这是所述协议的初始步骤。因此,轻便的、无连接的传输协议。例如UDP就可以是足够的。

[0236] 因此,如果没有接收到应答,则可以周期性地发出征求消息1532。在一些重复之后,组长(即正在注册的或正在连接的)设备假定不存在监视器服务,并至少在特定的时间间隔内停止发送消息,这节省了功率和计算时间。

[0237] 同时,系统1500的已经连接的和注册的服务(例如本地监视器服务1520)可以发出广告(advertisement)消息1534。可以例如响应于定时器或征求消息1532来发送广告消息1534。广告消息1534可以包括发送广告消息1534的监视器服务(例如在图15的例子中的本地监视器服务1520)的类型的标识。通过接收广告消息1534,网络中的其它监视器服务发觉存在当前正在接受子代监视器服务的指定类型的监视器服务,并且意识到如何与该可用的监视器服务通信。可以将广告消息1534作为广播消息来发送,并且可以由这里所描述的任何或所有设备(一般来说,除了各种边缘设备,因为边缘设备一般不具有任何连接到其上的子代监视器服务)来发送广告消息1534。

[0238] 广告消息1534提供在征求消息1532丢失的情况下自动失效保护(fail-safe)机制,还可以用作例如用于不发送征求消息的设备的注册过程的发起者(initiator)。例如,如上面提到的,组长设备1502可以在指定时间段之后,或者如果设备不能检测网络变化的情况下停止发送征求。使用广告消息1530,这种设备可以被“重新激活”。并且,丢失的广告消息不是破坏性的,因为它们会被一次次地重发。

[0239] 正在连接的/正在注册的设备,诸如组长设备1502可以通过发送注册消息1536而对广告消息1534做出一次响应。如上面提到的,注册消息1536可以包含例如关于被监视的数据的信息、全景元数据、设备元数据和/或实际被监视的(例如传感器)值。

[0240] 一旦在广告消息1534的发送者(即本地监视器服务1520)处成功接收到注册消息1536,注册数据就可以被保存在从接收的父代监视器到全局监视器服务1508的路径上的所有设备中(例如,在图15的例子中,可以被存储在设备1524和1512以及任何居间的设备)。换句话说,注册消息1536被转发到上层监视器服务。

[0241] 上层监视器服务的存储器(例如1510、1522)中的过时的数据可以被标记为“怀疑”,并且/或者可以被删除以避免不一致。例如,全局监视器数据1510可以具有来自之前的注册事件的信息:组长设备1502当前连接到另一个网络(图15中未示出)。然而,由于组长设备1502已经从以前的网络的设备移开,因此以前的连接丢失,从而使全局监视器数据1510中的数据变得过时。一旦组长设备1502通过本地监视器服务1520再次注册,如图15所示,则可以由此更新全局监视器数据1510中的当前注册状态、位置和其它信息/数据。

[0242] 注册系统 1536 可以通过在注册过程中仅允许一个消息触发的注册消息 1536 来保证注册消息 1536 的副本不是问题。也就是，注册系统 1536 可以被配置成一旦发送了注册消息 1536，就忽略附加的广告消息 1534。

[0243] 然后，注册消息 1536 获得确认消息 1538，以保证注册被成功处理。如果在预定时间之后没有确认到达，则可以重发注册消息 1536。在这样的实现方式中，重复的注册消息不是问题，因为可以允许一个注册覆盖 (override) 另一个而不引起任何损害。此外，或者可替代地，如果相关服务或设备已经注册，则可以简单地忽略重复的注册消息。

[0244] 确认消息 1538 可以由正在管理 (managing) 的监视器服务（例如本地监视器服务 1520）来发送。当正在注册的设备（组长设备 1502）接收到确认消息 1538 时，注册过程结束（如果没接收到确认消息 1538，例如，确认消息 1538 丢失，则组长设备 1502 可以简单地重新发送注册消息 1536）。然后，组长设备 1502 的现在已注册的监视器服务可以开始其分配的监视任务。因此，图 15 的注册协议允许无连接和不可靠的数据传输，从而可以使用简单且健壮的协议，例如 UDP。

[0245] 图 16 是图示图 15 的注册协议的示例实现方式的顺序图。在图 16 的例子中，出现了各种异常，以便图示该协议的健壮本质。

[0246] 具体来说，从组长设备 1502（例如从在其上安装的组长监视器服务）向本地监视器服务 1520 发送征求消息，但该征求消息未被接收 (1602)。即，该征求消息包括期望本地监视器服务以作为父代设备的标识，但是没有这样的潜在的父代设备接收到所述消息或对其做出响应。

[0247] 在指定的时间之后，发送第二征求消息 (1604)，其在本地监视器服务 1520 和全局监视器服务 1508 两者处都被接收。如上所述，注册系统 1528 可以确定由于第一征求消息的发送 (1602) 不成功，因此有必要发送第二征求消息。

[0248] 在这种情况下，本地监视器服务 1520 和全局监视器服务 1508 两者都接收到所述征求消息 (1604)。然而，全局监视器服务 1508 简单地忽略 (1606) 该征求，这是因为该征求指定了本地监视器服务 1520。

[0249] 然后，本地监视器服务 1520 发出广告消息 (1608a 和 1608b)。可以响应于征求消息的发送 (1604) 来发送该广告消息 (1608a)，或者可以根据定时器值来发送该广告消息。因此，附加广告消息的发送 (1608b) 作为广播的一部分到达全局监视器服务 1508，但是没有被忽略。

[0250] 然后，组长设备 1502 发送注册消息 (1612)，导致本地监视器服务注册相关元数据 (1614)。在这点上，尽管未示出，已注册的元数据可以被转发到全局监视器服务 1508（例如使用增量同步）。然后，可以发送确认消息，但是在图 16 的例子中，在注册系统 1528 处没有接收到该确认消息 (1616)。因此，组长设备 1502 的注册系统 1528 简单地重新发送注册消息 (1618)，然后，在本地监视器服务 1520 处再次接收到该注册消息，并且该注册消息被重新注册 (1620) 或被忽略。再次发送确认消息，这次在组长设备 1502 接收到该确认消息 (1622)，因此注册过程结束。

[0251] 图 17 是图示图 15 的系统 1500 的示例操作的流程图 1700。在图 17 的例子中，正在连接的监视器服务（例如组长设备 1502 的监视器服务，该组长设备 1502 可以包括注册系统 1528 或与其相关联）广播征求消息（例如征求消息 1532），其包括来自在系统 1500 的

分层级的监视器架构中的期望的父代监视器的标识 (1702)。

[0252] 如果在父代服务处 (例如在本地监视器服务 1520 处) 没有接收到该征求消息 1532 (1704), 则该父代监视器服务可以继续等待预定义的定时器 (1706)。因此, 如果定时器到时 (reach), 或者如果在该父代监视器服务处接收到了征求消息, 则该父代监视器服务可以发送作为结果的广告消息, 例如广告消息 1534 (1708)。然后, 应当理解, 无论是否接收到征求消息 1532, 都可以发送广告消息 1534, 因此增加了正在连接的设备连接 / 注册的可能性。

[0253] 如果在正在连接的设备没有接收到广告消息 1534 (1710), 则正在连接的设备可用重发征求消息 1532 (1712), 可能标识分层级的监视器架构的下一个较高层级作为期望的父代服务 (例如全局监视器服务 1508, 以代替本地监视器服务 1520、1514)。在这种情况下, 只要没有超时 (1714), 处理就可以继续, 直到交换了注册和 / 或广告消息。否则, 处理停止 (1716)。在这一点上应当理解, 超时条件照字面意思可以指经过的时间段, 或者可以指发送预定义数目的 (特定类型的) 消息。

[0254] 当然, 应当理解, 可以根据其相关联的定时器 (1706) 重新发送广告消息 (1708), 即使过程在之前停止或失败。而且, 图 17 中的几乎任何“停止”条件, 包括停止条件 (1716), 都可以是临时的, 并且过程可以在某个进一步定义的时间段到达之后继续。

[0255] 如果接收到广告消息 (1710), 则正在连接的设备可以向在广告消息中标识的父代监视器服务发送注册消息, 例如注册消息 1536 (1718)。如果没有接收到注册消息 (1720), 并且定时器没有到时 (1722), 则正在连接的设备可以重新发送注册消息 (1718)。如已经描述过的, 注册消息可以包括监视器数据, 从而, 如果接收到注册消息 (1720), 则可以沿等级向上传播该注册消息 (1726), 例如将其传播到全局监视器服务 1508。在这种情况下, 包括关于正在连接的设备的元数据以及实际的传感器值的监视器数据可以保存在一些或全部居间的设备处 (1728), 例如, 保存在本地监视器服务 1520 处。通过以这种方式传播注册 / 监视数据, 全局监视器服务 1508 可以保持对正在连接的服务 / 设备的存在的跟踪, 并且可以最后剩余的被标记为“怀疑”的数据, 所述被标记为“怀疑”的数据如果不删除的话可能会导致不一致 (例如同一设备在不同的网络同时注册)。

[0256] 然后, 父代监视器服务可以发送确认消息 (1730), 例如确认消息 1538。如果没有接收到确认消息 (1732), 并且还没有超时 (1734), 则可以重新发送注册消息 (1718), 并且上述过程可以全部地或部分地重复。如果超过了超时时间 (1734), 或者如果接收到了确认消息, 则注册过程也结束 (1724)。

[0257] 刚刚描述的注册协议可以在正在连接的设备 (例如组长设备 1502) 已经在其上安装了监视器服务时实现。然而, 如上面参照图 7B 所描述的, 如果正在连接的设备没有在其上安装监视器服务, 则可以进行预注册过程, 其中, 如上所述, 可以使用这样的自举技术: 其中, 核心监视器服务 148 被安装, 然后被用于执行图 15-17 的剩余的注册过程。

[0258] 如刚刚所描述的, 图 15-17 的健壮的、可靠的注册协议允许使用不可靠的 / 无连接的通信协议, 同时最小化注册过程所需的网络带宽的量。例如, 可以通过在特定时间段之后停止重新发送一个或多个消息来节约网络带宽。在这种情况下, 每次注册协议在超时之后要求重新发送消息时, 都存在这样的可能性, 即, 正在连接的设备或父代设备可能丢失连接, 并且这种可能性可能导致系统 1500 的全部或一些的“卡塞”(stuck) 状态。为了避免这

种状态，并且如上面所提到的，每个重新发送的消息仅可以被重新发送有限的次数，并且如果没有实现成功的注册，则整个注册过程可以重新开始。

[0259] 应当理解，以上描述仅仅旨在提供分等级地分层的监视器架构的例子，并且可以使用更多或更少层的这种等级。通过实现这种分等级地分层架构，例如，如上面更加具体地描述的，可以获得各种特征和优点。例如，图上面提到过的，此处描述的系统 100 可以容易地被扩大到包括大量智能物件设备，同时仍然能够执行对服务和 / 或设备的有效监视。

[0260] 这里描述的各种技术的实现方式可以被实施在数字电子电路中，或者实施在计算机硬件、固件、软件或它们的组合中。实现方式可以实施为计算机程序产品，即实实在在地具体实施在信息载体中，例如在机器可读存储设备中或者在传播的信号中，的计算机程序，以供数据处理装置执行，或者控制数据处理装置的操作，所述数据处理装置例如可编程处理器、计算机、多个计算机。计算机程序，例如上面描述的计算机程序，可以用任何形式的编程语言编写，包括汇编语言或解释语言，并且，它可以被以任何形式部署，包括作为独立的程序或者作为模块、组件、子程序或其他适于在计算环境中使用的单元。计算机程序可以被部署成在一个计算机上或在位于一个地点或跨过多个地点分布并被通信网络互连起来的多个计算机上执行。

[0261] 方法步骤可以被一个或更多个可编程处理器执行，所述可编程处理器执行计算机程序，通过对输入数据操作和产生输出来执行功能。方法步骤还可以被专用逻辑电路执行，或者装置可以被实施为专用逻辑电路，所述专用逻辑电路例如 FPGA (现场可编程门阵列) 或 ASIC (专用集成电路)。

[0262] 作为例子，适于执行计算机程序的处理器包括通用和专用微处理器，以及任何类型的数字计算机的任意一个或更多个处理器。一般来说，处理器将从只读存储器或随机访问存储器接收指令和数据，或者从两者都接收指令和 数据。计算机的要素可以包括至少一个用于执行指令的处理器，和用于储存指令和数据的一个或更多个存储器设备。一般来说，计算机还可以包括，或者被可操作地连接，以从一个或更多个用于存储数据的海量储存设备接收数据，或把数据传送到海量储存设备，或者二者皆有，所述海量储存设备例如：磁盘、磁光盘或光盘。适于具体实施计算机程序指令和数据的信息载体包括所有形式的非易失性存储器，作为例子，包括半导体存储器器件，例如：EPROM、EEPROM 和快闪存储器设备、磁盘，例如内置硬盘或可移动磁盘、磁光盘和 CD-ROM 以及 DVD-ROM 盘。处理器和存储器可以被专用逻辑电路补充，或被包含在专用逻辑电路中。

[0263] 为了提供和用户的交互，实现方式可以在具有显示设备和键盘以及指示设备 (pointing device) 的计算机上实施，显示设备例如阴极射线管 (CRT) 或液晶显示器 (LCD) 监视器，用于向用户显示信息，键盘和指示设备例如鼠标或跟踪球，用户利用它们可以提供到计算机的输入。其他种类的设备也可以被用来提供和用户的交互；例如，提供给用户的反馈可以是任何形式的感觉反馈，例如视觉反馈、听觉反馈或触觉反馈，并且，来自用户的输入可以被以任何形式接收，包括声音、语音或触觉输入。

[0264] 实现方式可以被在包括后端组件或包括中间件组件或包括前端组件的计算系统中实施，或者在这些后端、中间件、前端组件的任意组合中实施，后端组件例如数据服务器，中间件组件例如应用服务器，前端组件例如具有图形用户接口，或 Web 浏览器的客户端计算机，通过图形用户界面或 Web 浏览器，用户可以和实现方式进行交互。可以利用数字数据

通信的任何形式或介质互连组件，数字数据通信介质例如通信网络。通信网络的例子包括：局域网（LAN）和广域网（WAN），例如因特网。

[0265] 虽然如这里所描述的那样已经示出了所描述的实现方式的某些特征，但是本领域普通技术人员现在将想到很多修改、替换，变化或等同物。因此要理解，所附权利要求旨在覆盖所有这些修改和变化。

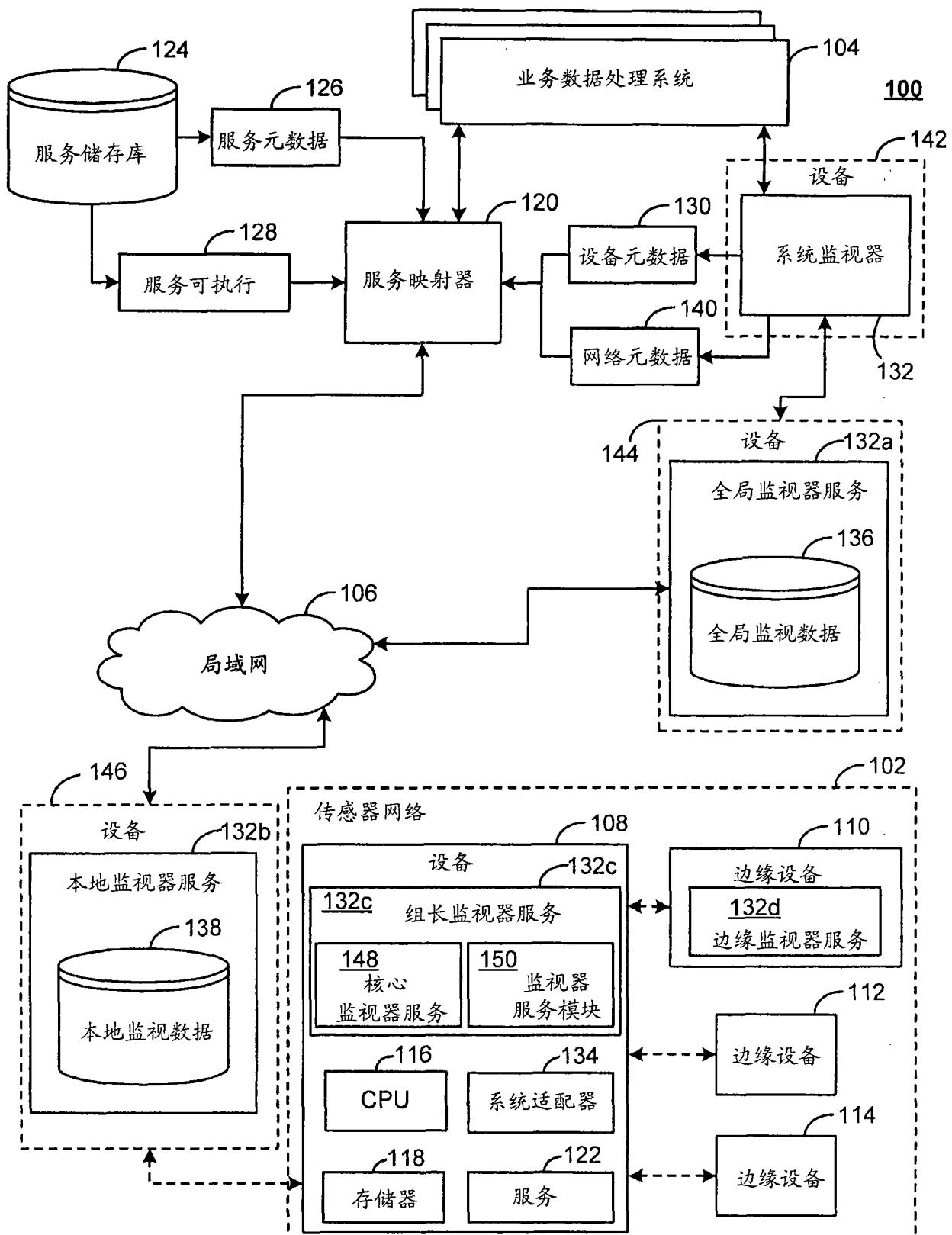


图 1

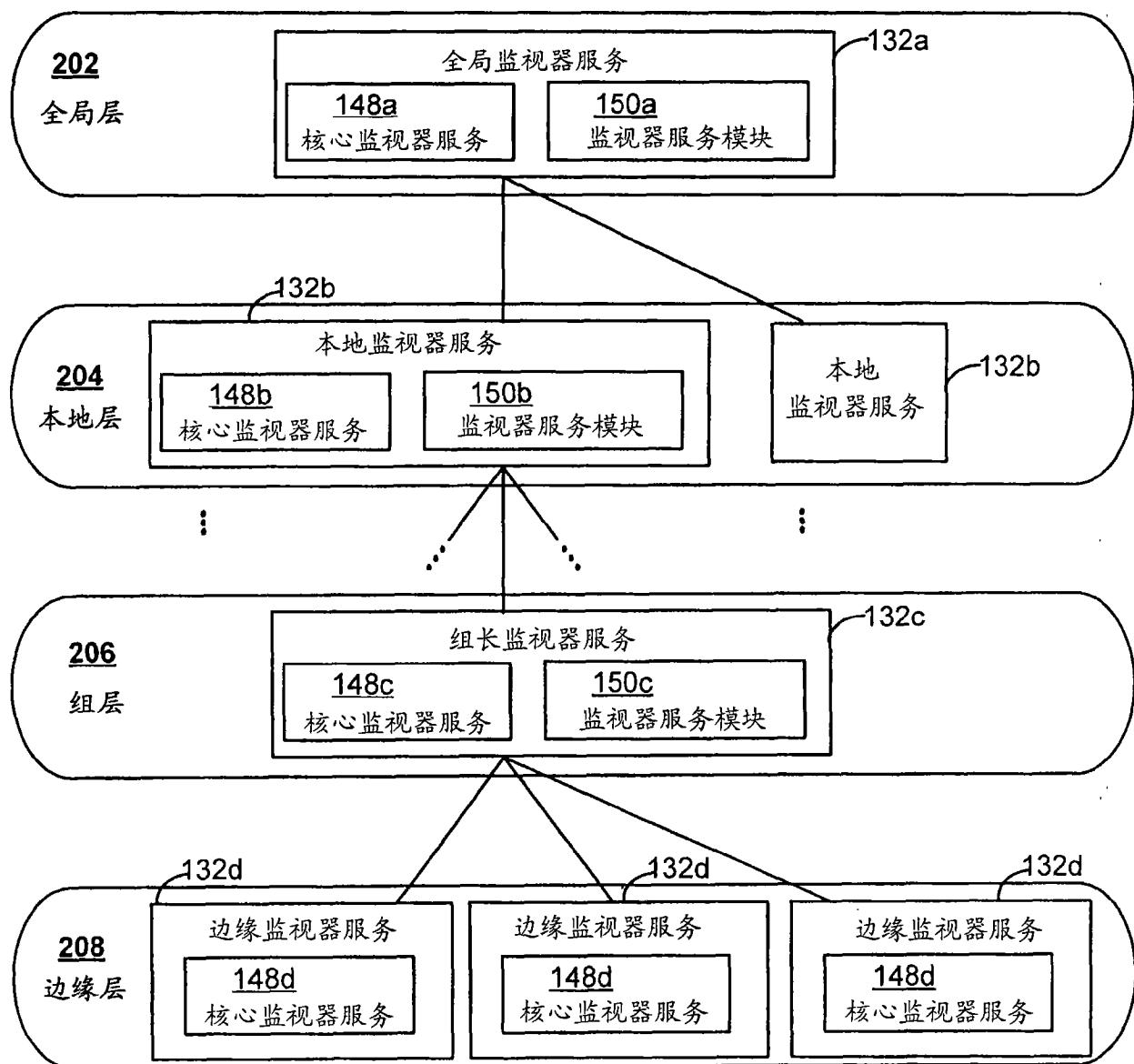
200

图 2

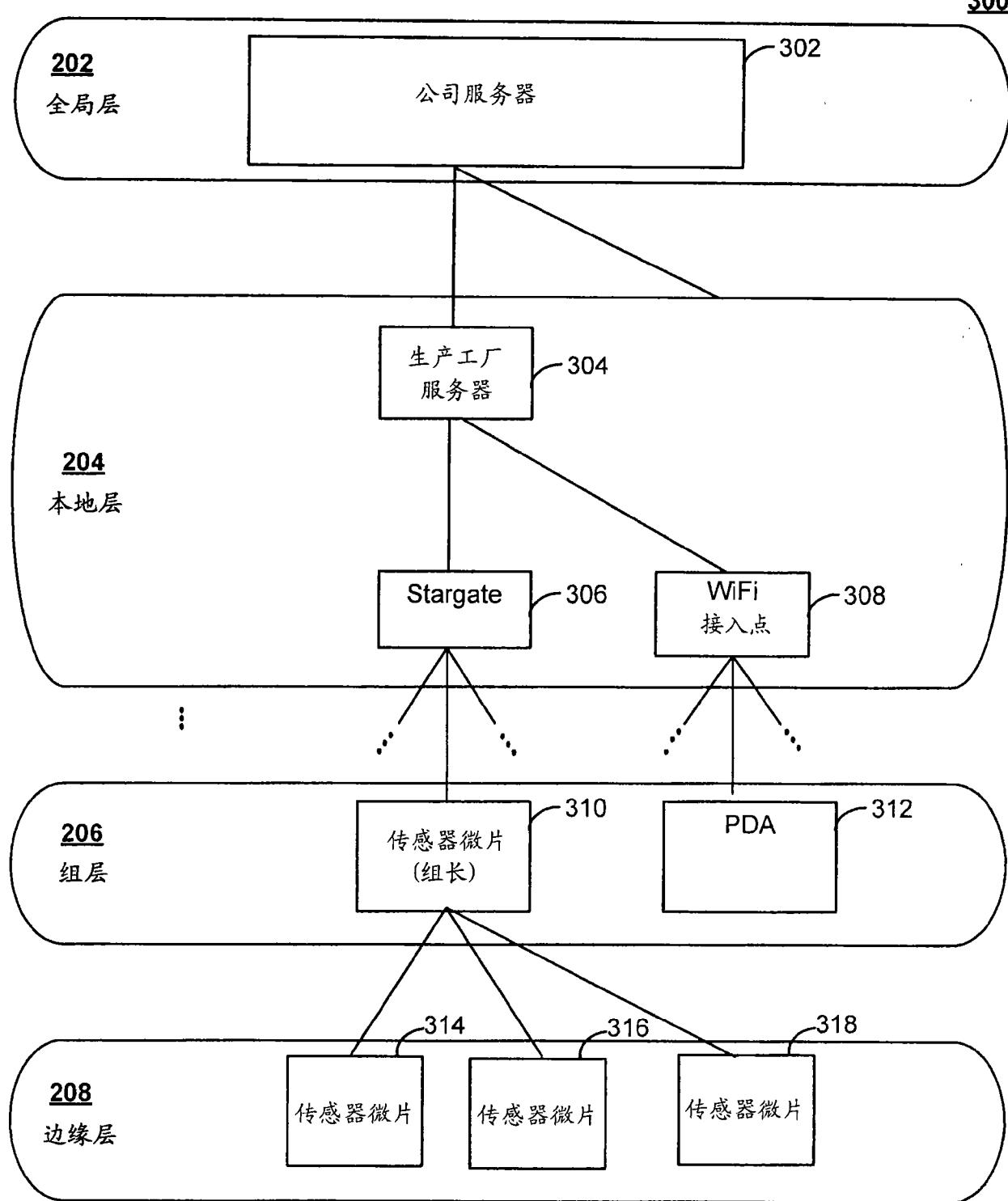
300

图 3

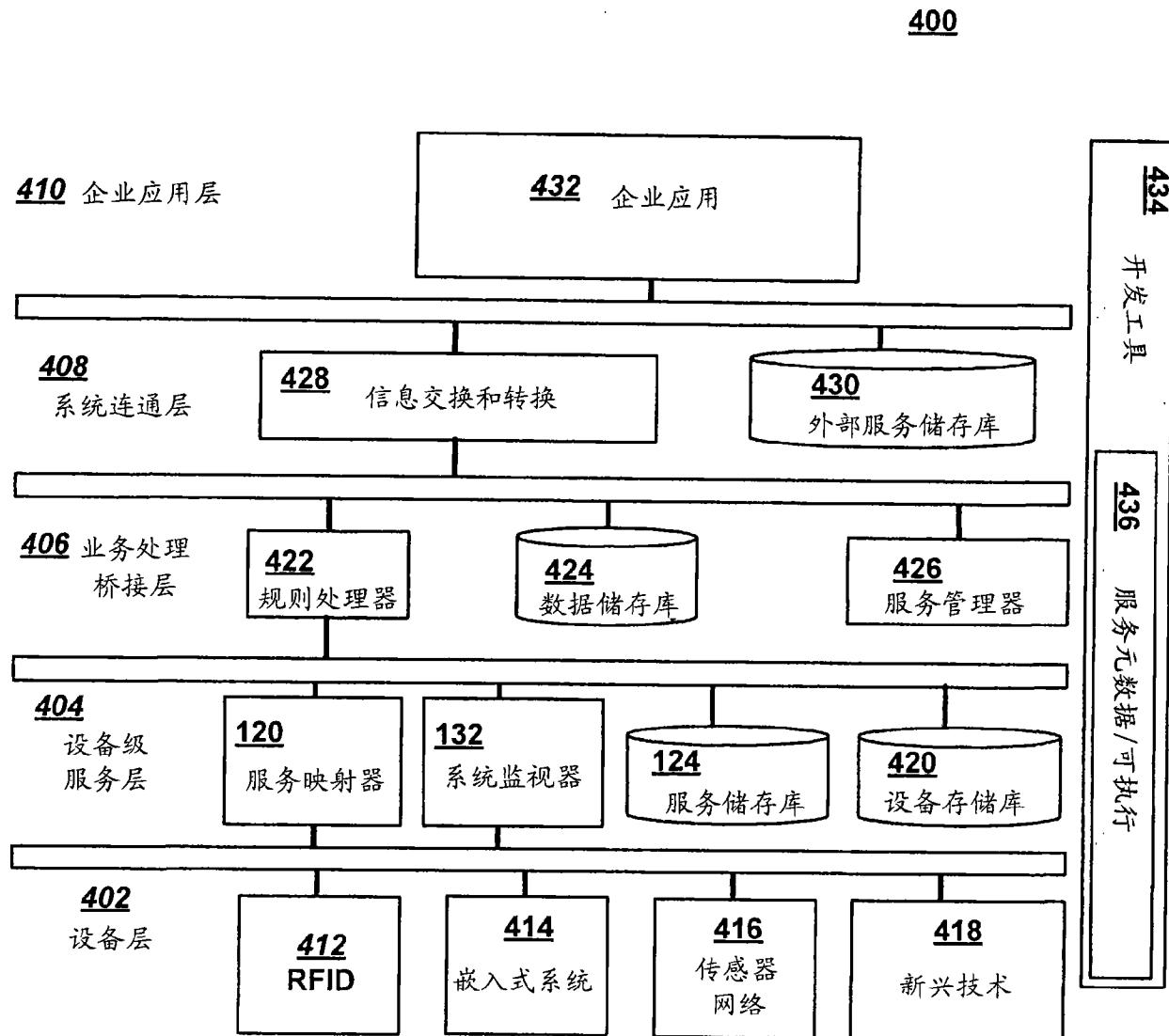


图 4

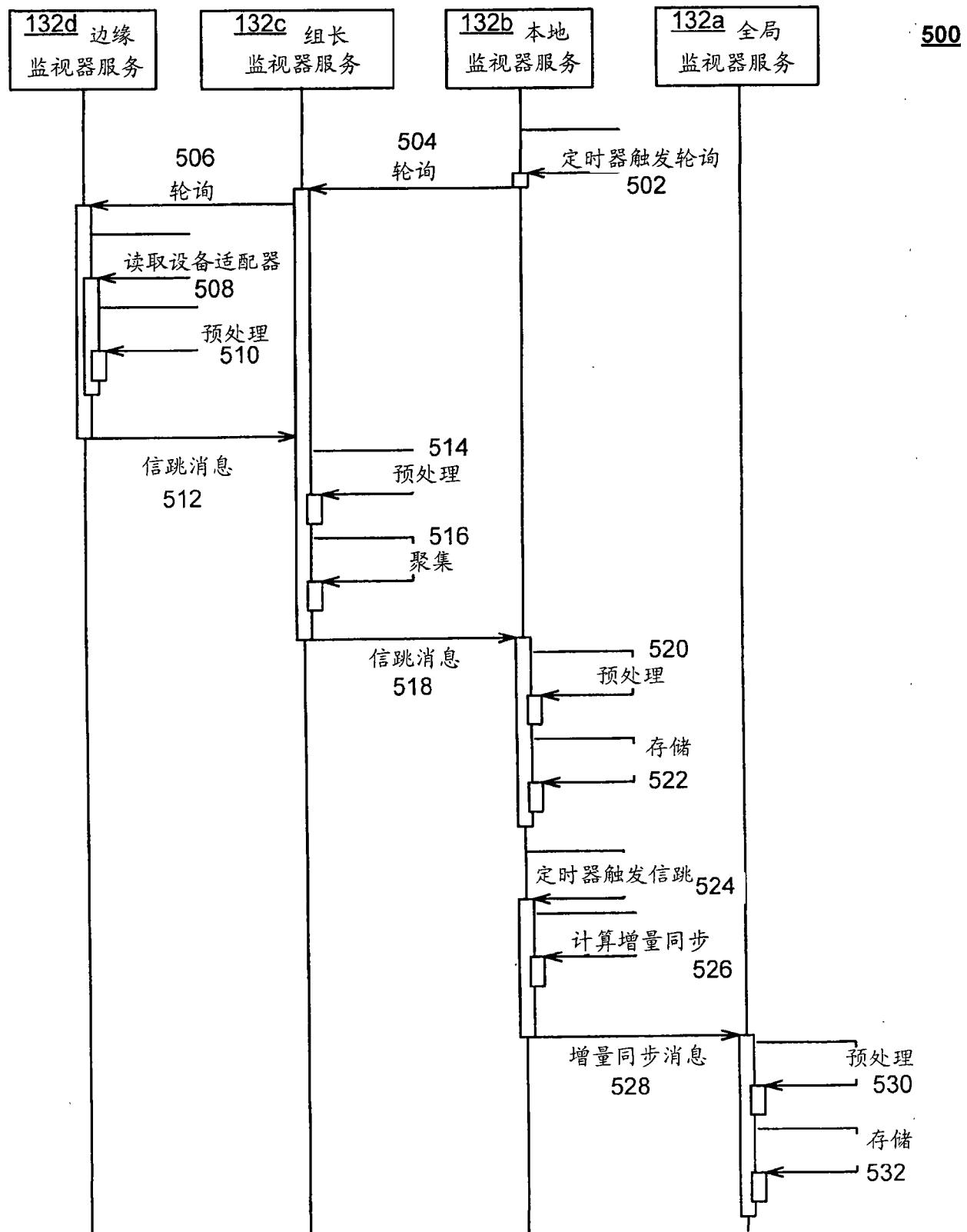


图 5

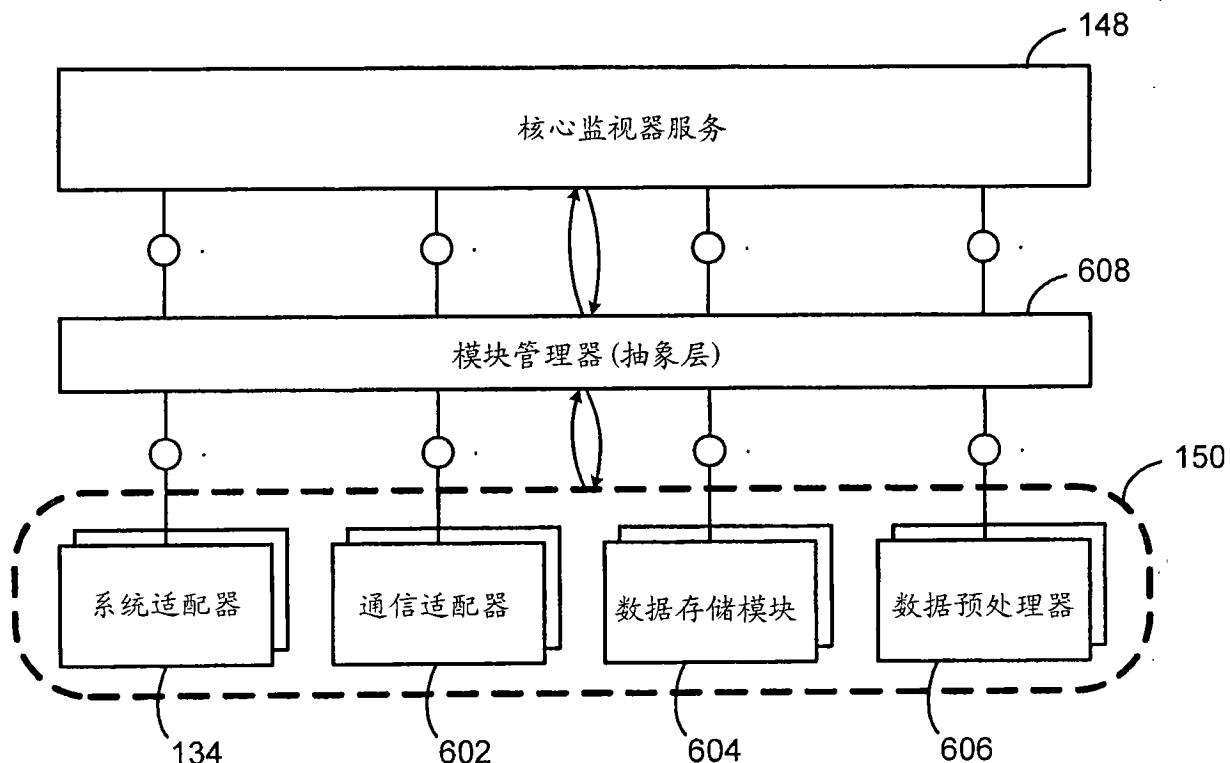
600

图 6

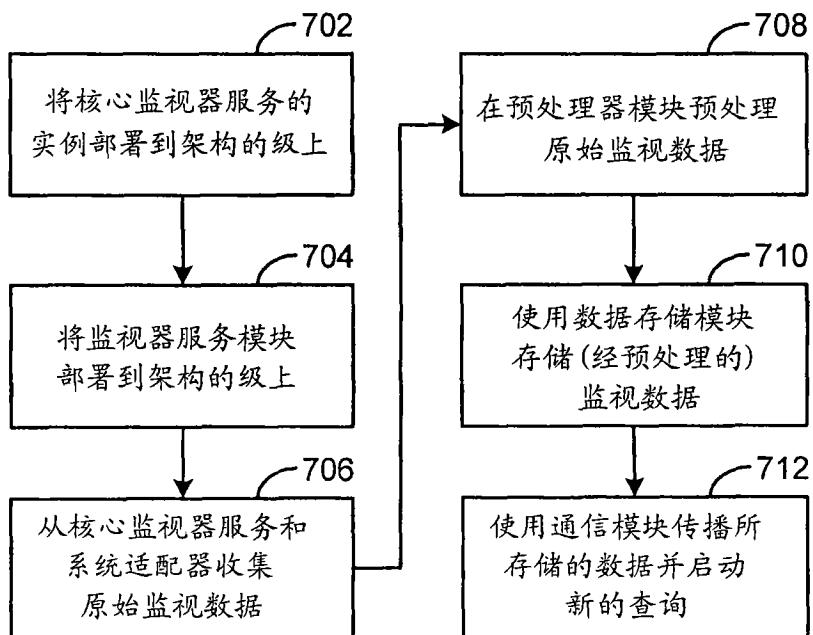


图 7A

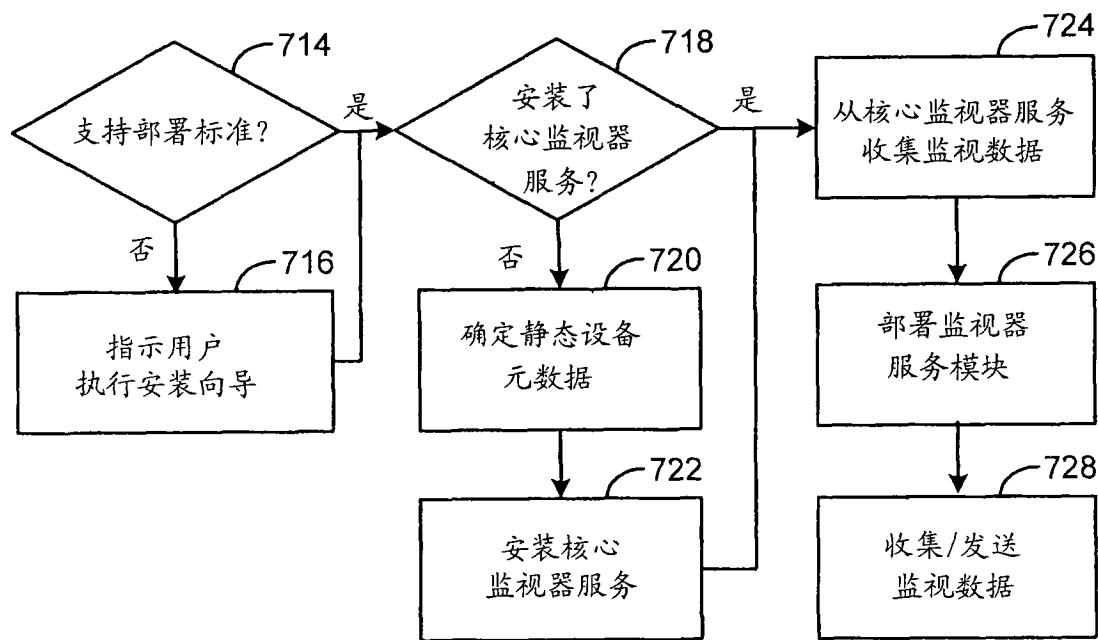


图 7B

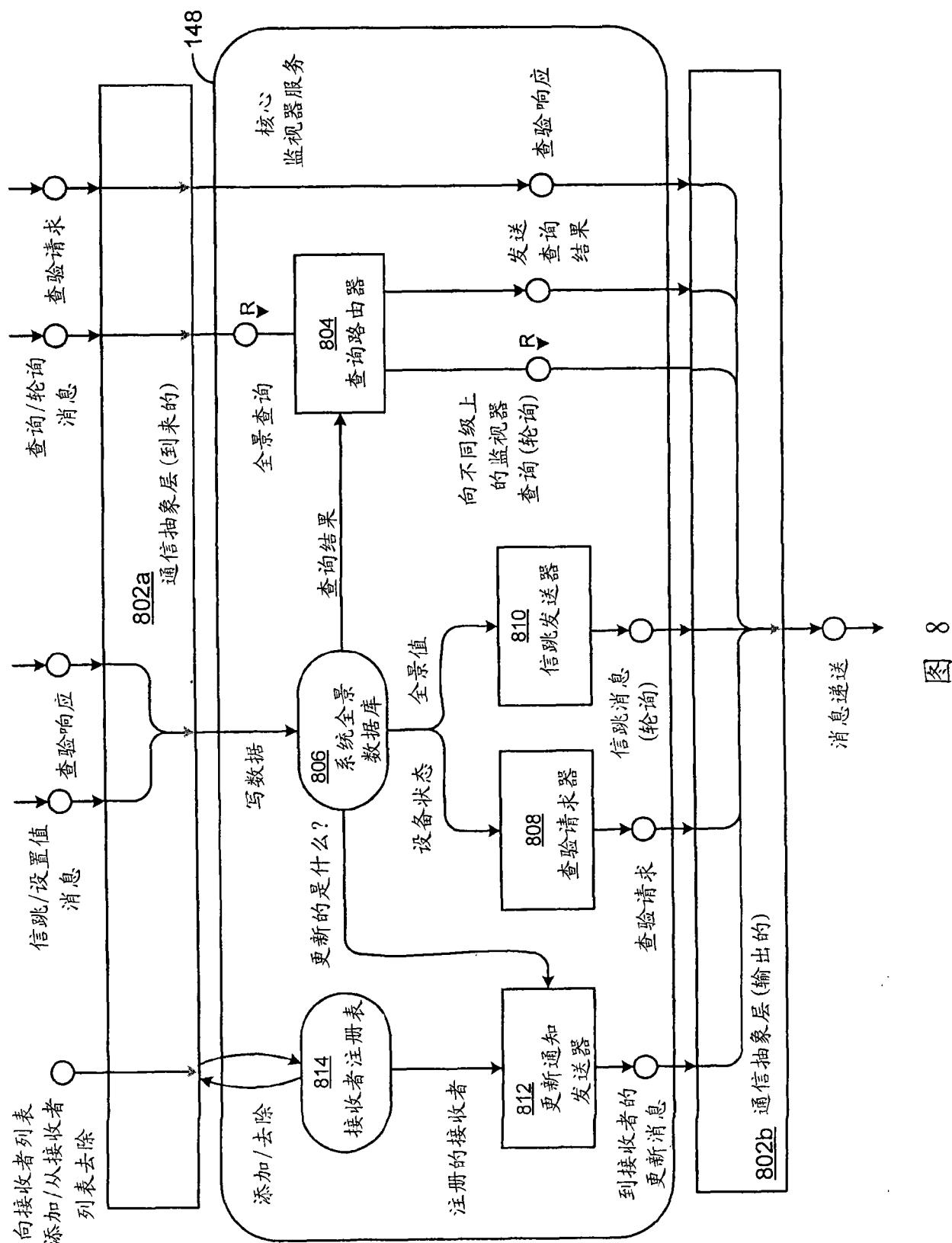


图 8

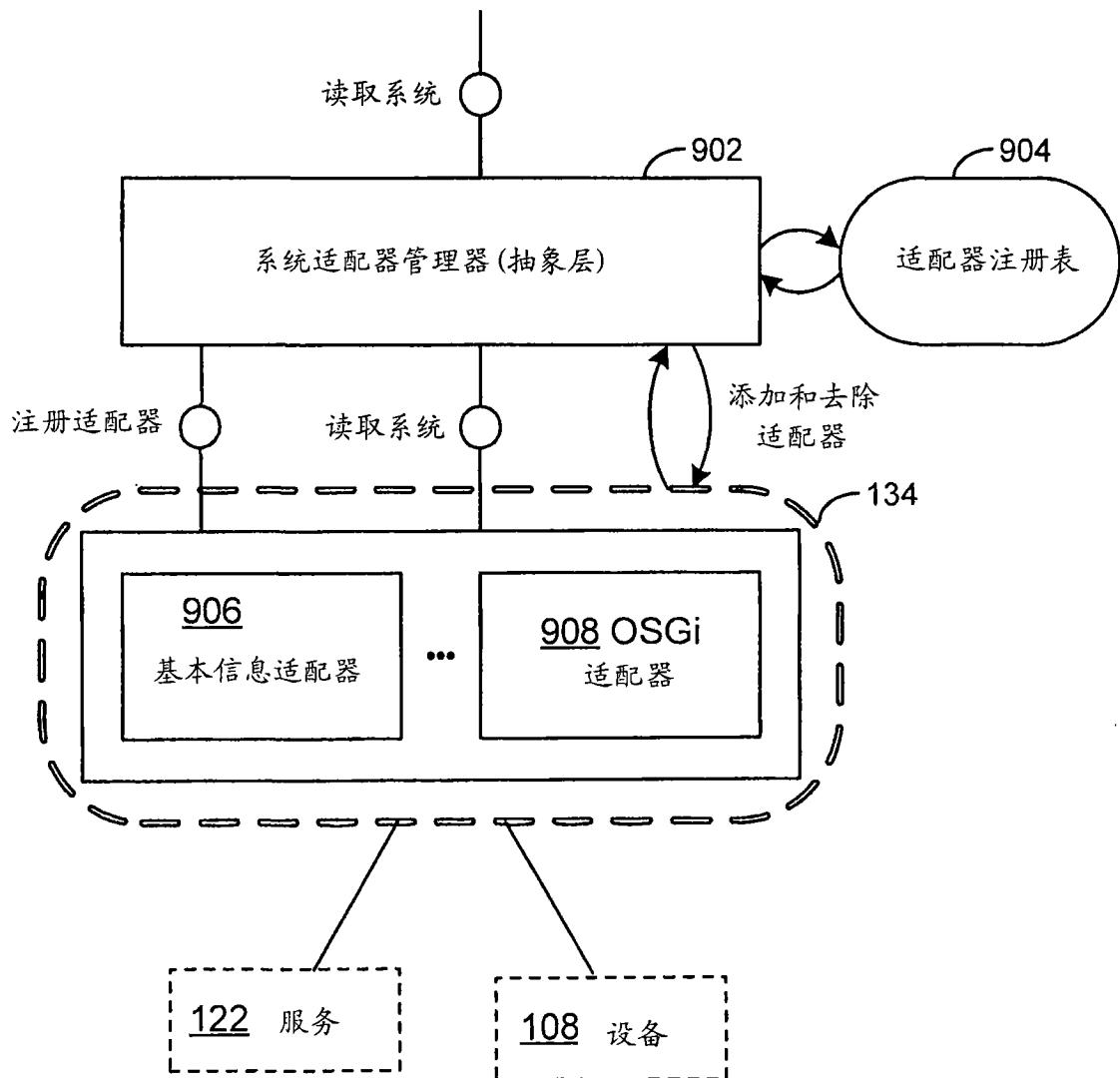


图 9

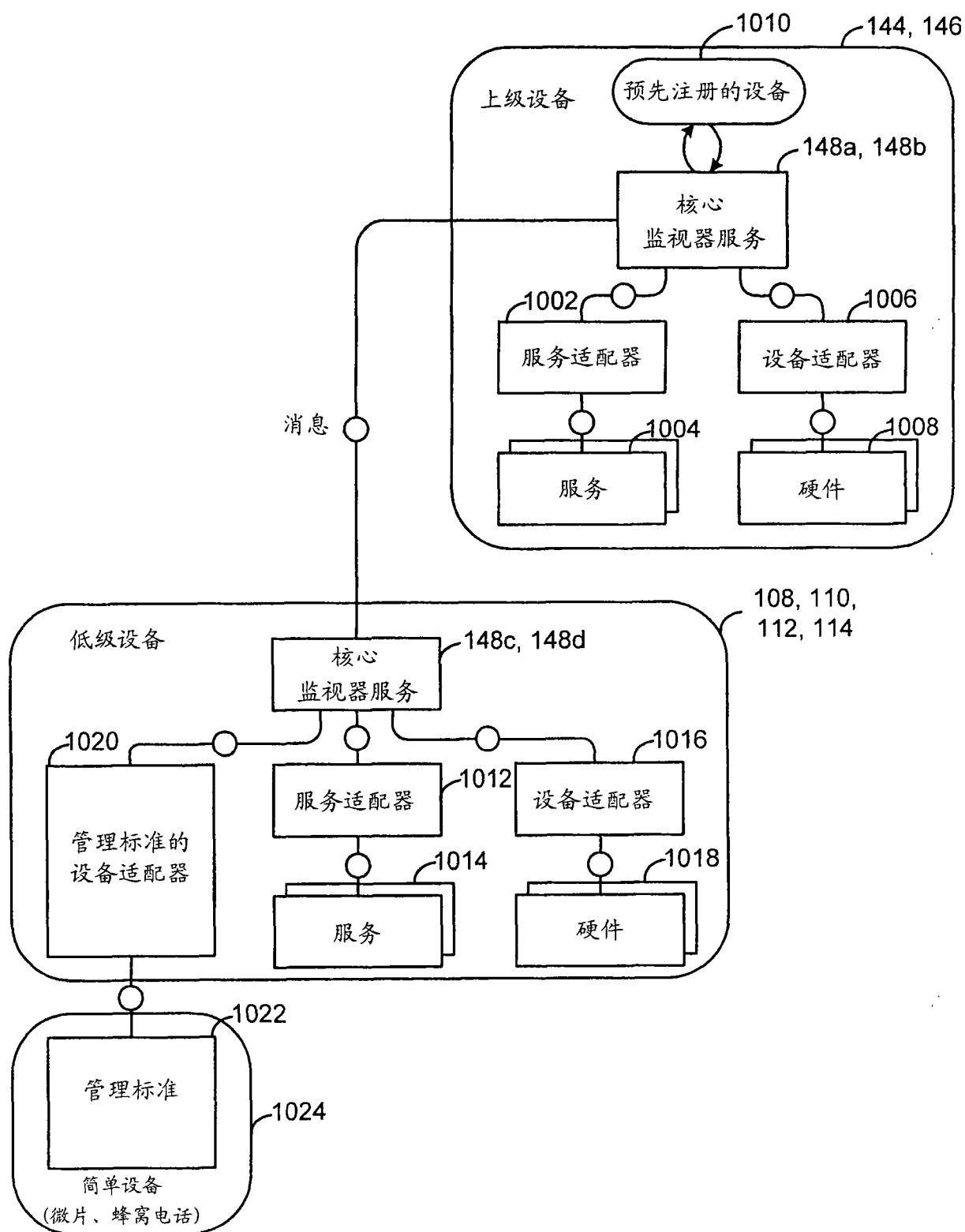


图 10

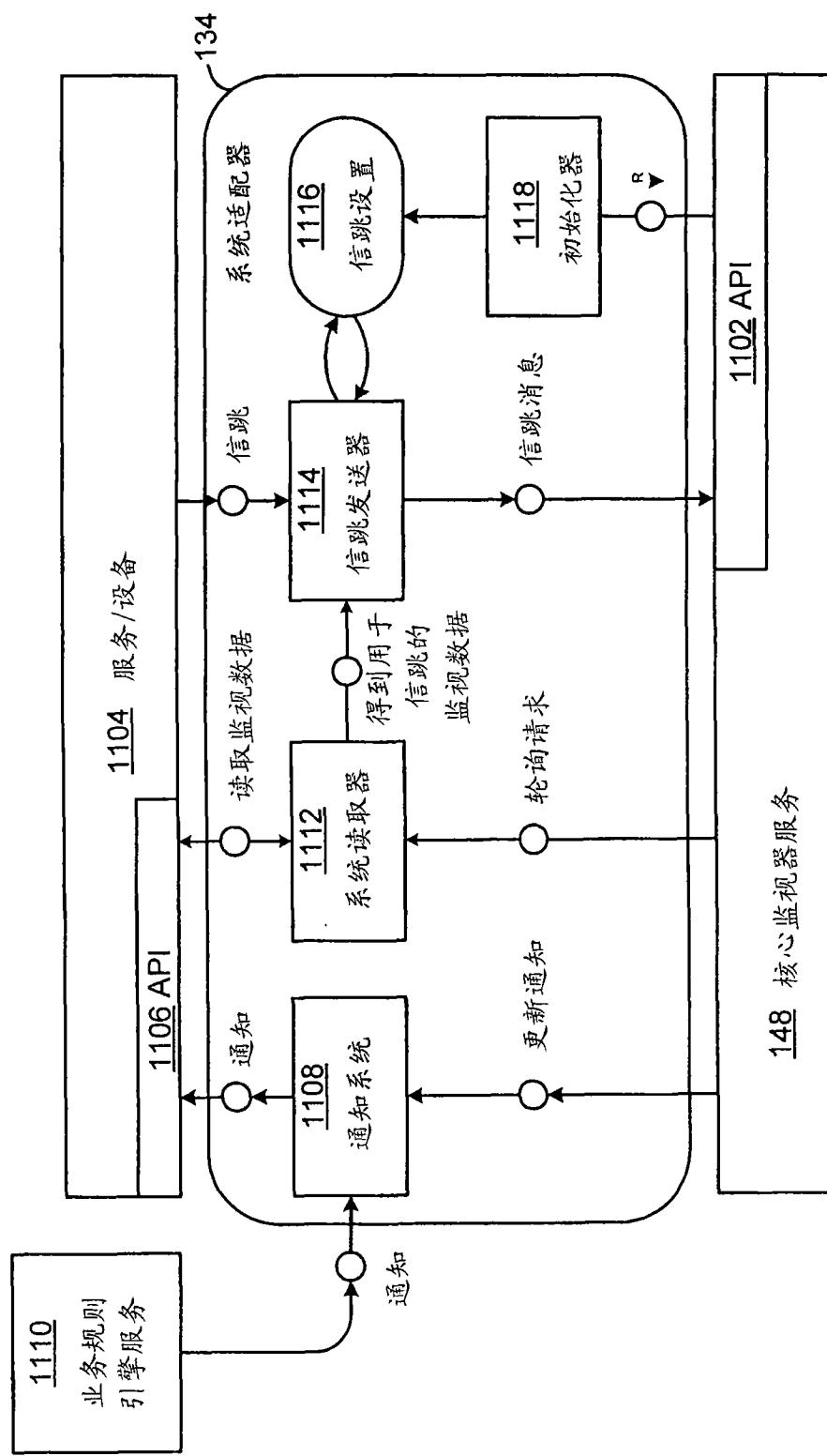


图 1.1

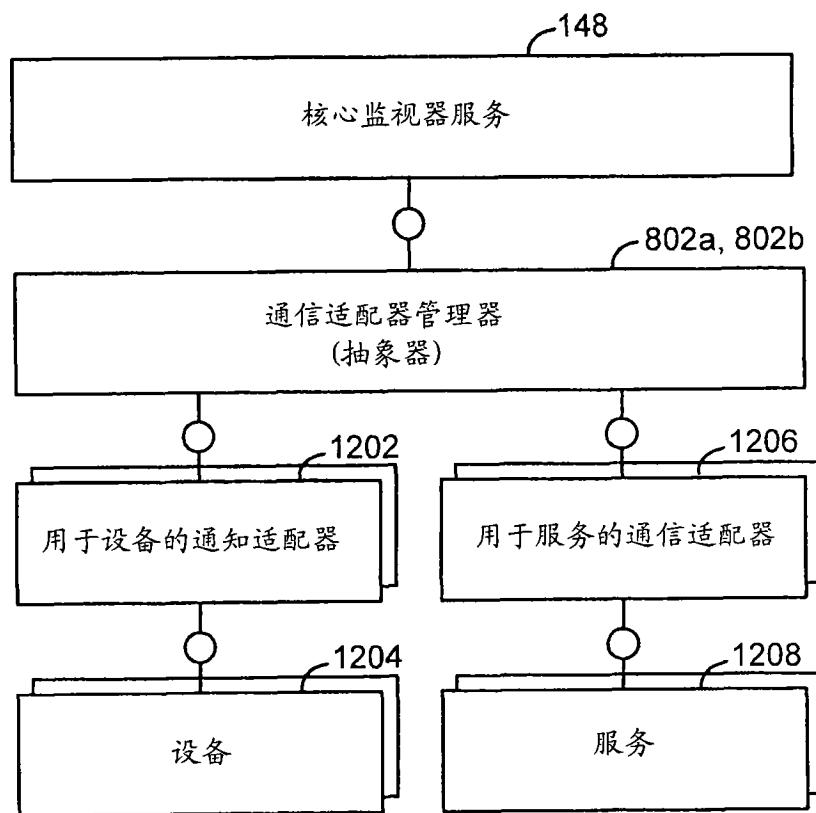


图 12

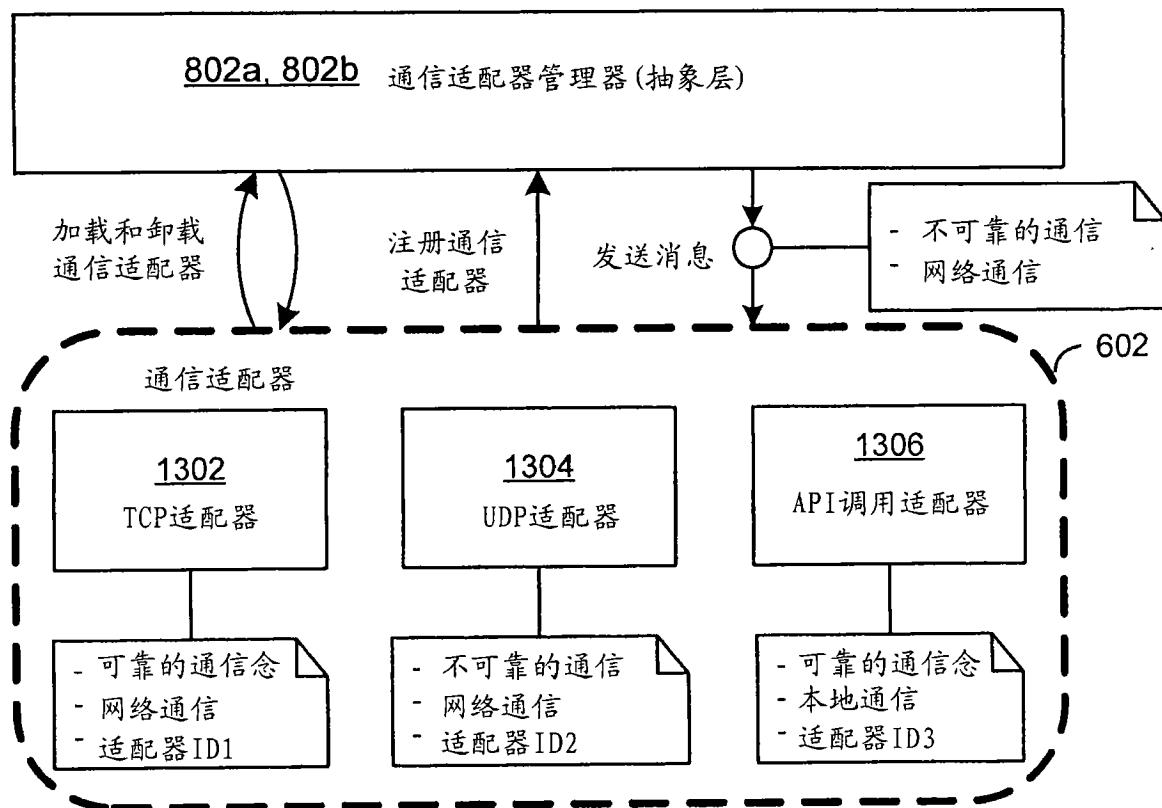


图 13

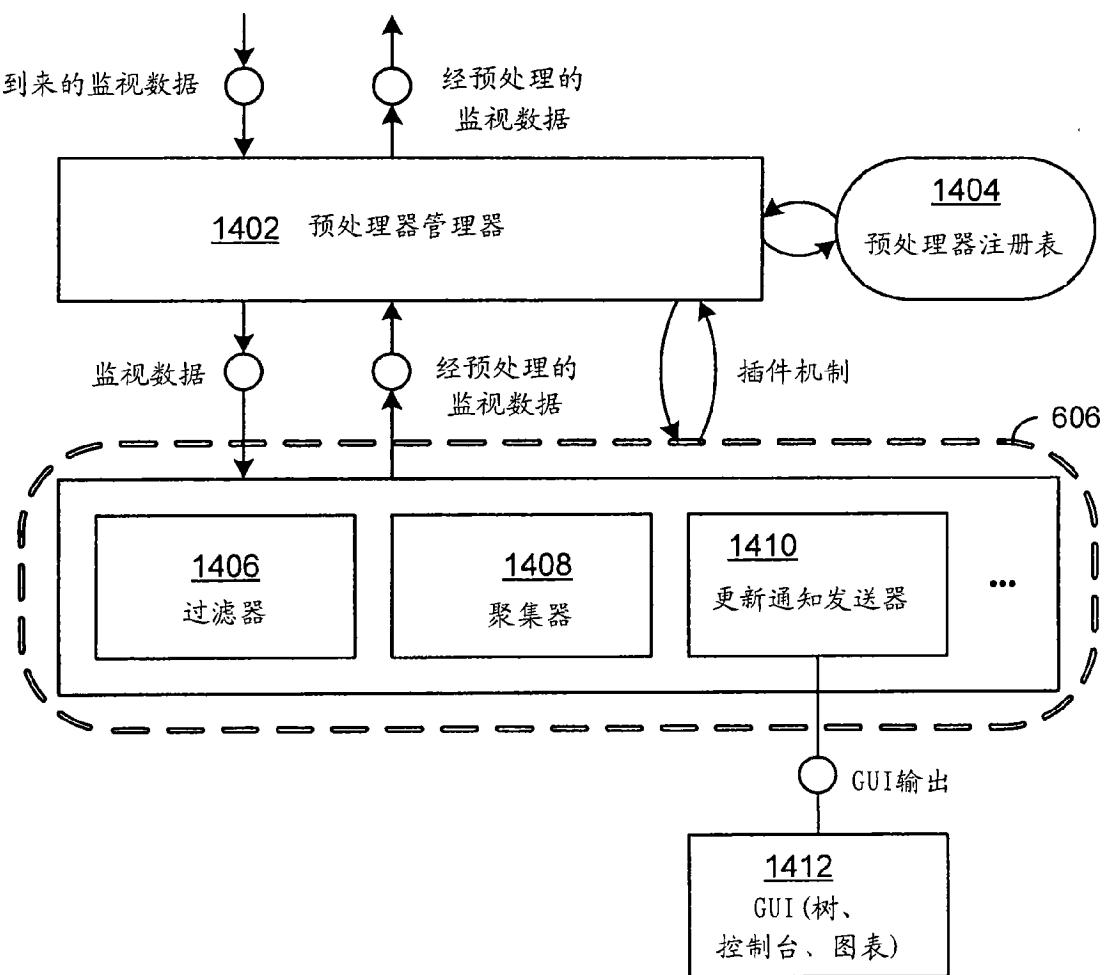


图 14

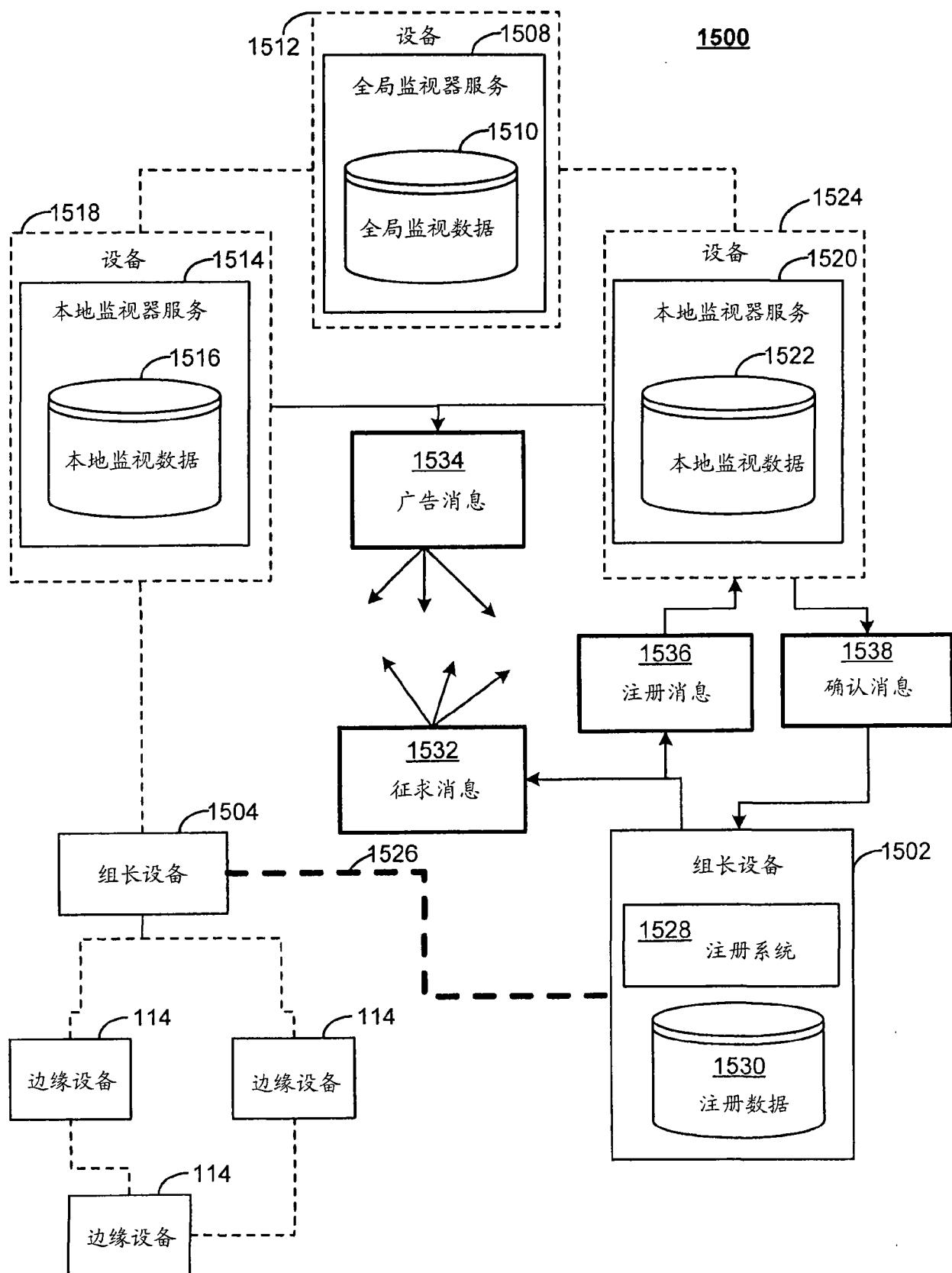


图 15

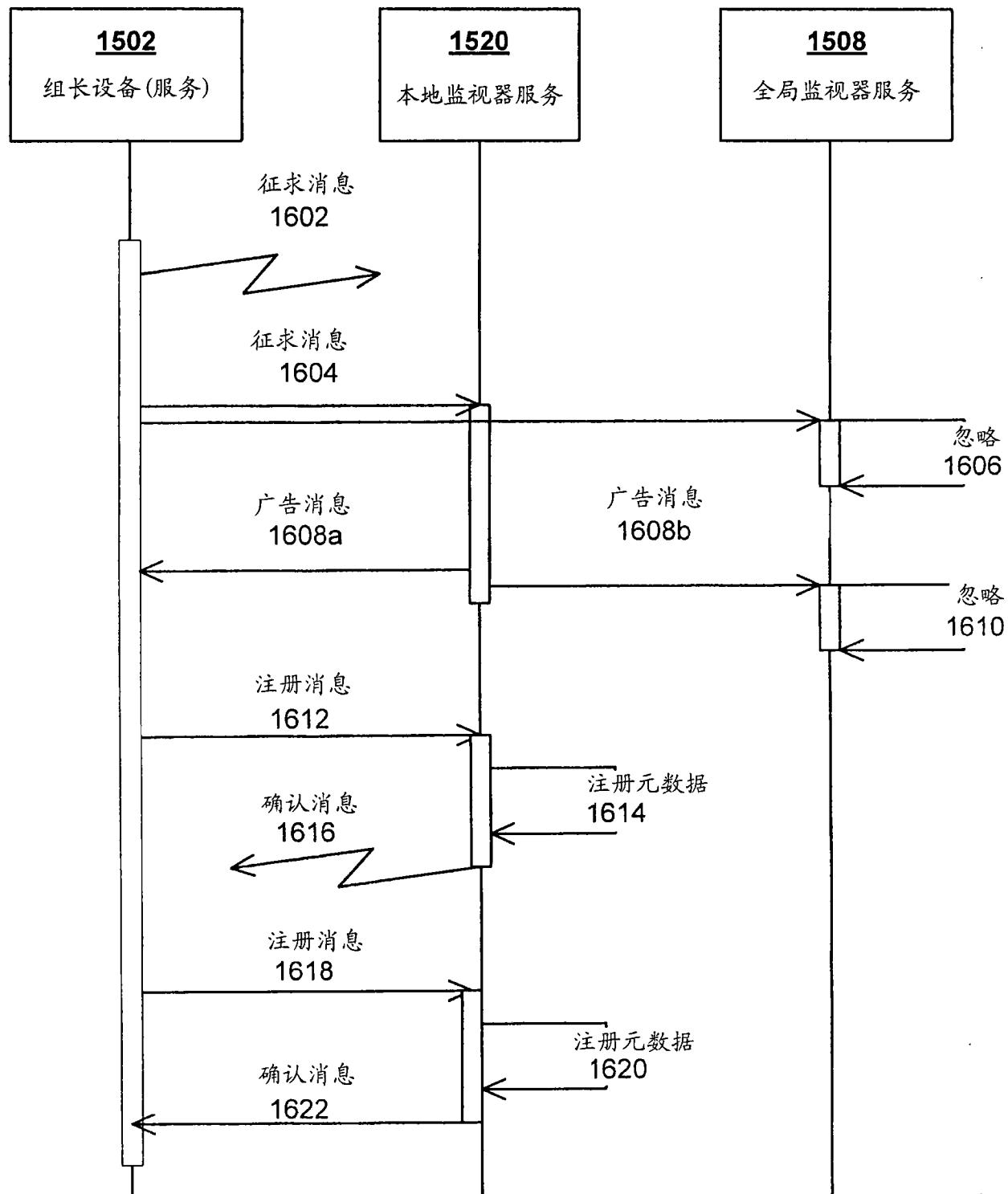


图 16

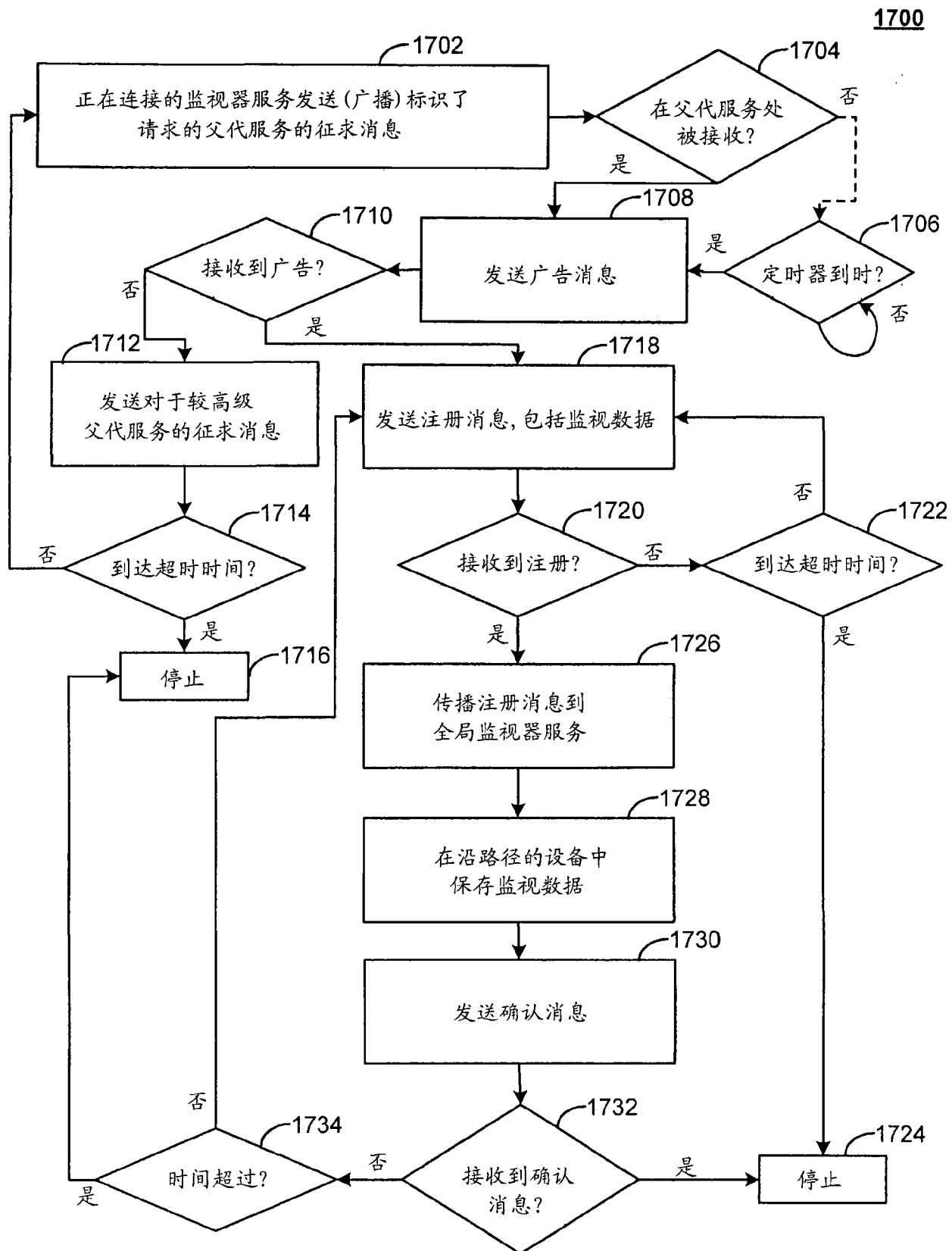


图 17