



## (12) 发明专利

(10) 授权公告号 CN 106796526 B

(45) 授权公告日 2021.03.16

(21) 申请号 201580052625.5

(22) 申请日 2015.10.30

(65) 同一申请的已公布的文献号

申请公布号 CN 106796526 A

(43) 申请公布日 2017.05.31

(30) 优先权数据

14/529,546 2014.10.31 US

(85) PCT国际申请进入国家阶段日

2017.03.29

(86) PCT国际申请的申请数据

PCT/US2015/058374 2015.10.30

(87) PCT国际申请的公布数据

W02016/070081 EN 2016.05.06

(73) 专利权人 甲骨文国际公司

地址 美国加利福尼亚

(72) 发明人 A·马修 N·辛诃

(74) 专利代理机构 中国贸促会专利商标事务所  
有限公司 11038

代理人 边海梅

(51) Int.Cl.

G06F 8/51 (2018.01)

G06F 8/52 (2018.01)

G06F 8/20 (2018.01)

G06F 8/10 (2018.01)

审查员 董显彬

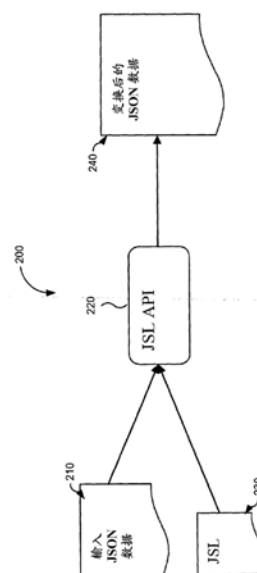
权利要求书3页 说明书24页 附图15页

(54) 发明名称

JSON样式表语言变换

(57) 摘要

提供了用于使用其它JSON对象指定JSON对象的变换的系统和方法。接收使用JavaScript对象标记指定的第一对象。第一对象包括一个或多个属性的集合,其中每个属性具有预定的JSON数据类型并且具有至少一个值。还接收使用JavaScript对象标记指定的第二对象。第二对象包括一个或多个属性的集合,每个属性与第一对象的属性的集合中的至少一个属性对应并且具有定义一个或多个变换的至少一个值。基于使用第二对象变换第一对象而生成使用JavaScript对象标记指定的第三对象。



1. 一种用于变换JSON数据的方法,所述方法包括:

在计算机处接收使用JavaScript对象标记指定的第一对象,第一对象具有一个或多个属性的集合,每个属性具有预定的JSON数据类型并且具有至少一个值;

在所述计算机处接收使用JavaScript对象标记指定的第二对象,第二对象具有一个或多个属性的集合,每个属性与第一对象的属性的集合中的至少一个属性对应并且具有定义一个或多个变换的至少一个值,所述至少一个值包括要在第一对象的属性的集合中的对应的至少一个属性的值上执行的表达式;

由所述计算机基于第二对象将第一对象变换为使用JavaScript对象标记指定的第三对象;及

将第三对象存储在与所述计算机关联的存储设备中。

2. 如权利要求1所述的方法,其中在所述计算机处接收第二对象包括:接收使用JavaScript对象标记指定的、具有名称与第一对象的属性的名称对应的属性的对象。

3. 如权利要求2所述的方法,其中接收具有名称与第一对象的属性的名称对应的属性的对象包括:接收具有与第一对象的属性相同名称的属性。

4. 如权利要求2所述的方法,其中接收具有名称与第一对象的属性的名称对应的属性的对象包括:接收具有对第一对象的属性的名称进行引用的值的属性。

5. 如权利要求1所述的方法,其中在所述计算机处接收第二对象包括:接收使用JavaScript对象标记指定的对象的具有如下值的属性,该值指定要应用到第一对象的属性的名称或第一对象的属性的值的操作。

6. 如权利要求1所述的方法,其中在所述计算机处接收第二对象包括:接收使用JavaScript对象标记指定的对象的具有如下值的属性,该值指定与第一对象的属性的变换对应的第三对象的属性的数据类型。

7. 如权利要求1所述的方法,其中在所述计算机处接收第二对象包括:接收使用JavaScript对象标记指定的对象的具有如下值的属性,该值指定对第一对象的一个或多个属性的修改。

8. 如权利要求1所述的方法,其中在所述计算机处接收第二对象包括:接收使用JavaScript对象标记指定的对象的具有如下值的属性,该值定义一个或多个测试,该一个或多个测试定义所述一个或多个变换是否能够应用到第一对象。

9. 如权利要求8所述的方法,其中所述一个或多个测试是关于第一对象的一个或多个属性定义的。

10. 如权利要求8所述的方法,还包括:

由所述计算机确定所述一个或多个测试的肯定评估;及

由所述计算机基于所述一个或多个测试的肯定评估来执行所述一个或多个变换。

11. 如权利要求10所述的方法,其中由所述计算机执行所述一个或多个变换包括:重命名第一对象的属性、关于第一对象的属性的值执行一个或多个数值操作、关于第一对象的属性的值执行一个或多个字符串操作、将一个或多个属性添加到第一对象的也作为对象的属性、或者从第一对象的也作为对象的属性中移除一个或多个属性。

12. 一种非临时性计算机可读介质,存储能够由计算机系统的处理器执行来变换JSON数据的计算机程序产品,所述非临时性计算机可读介质包括:

用于接收使用JavaScript对象标记指定的第一对象的代码,第一对象具有一个或多个属性的集合,每个属性具有预定的JSON数据类型并且具有至少一个值;

用于接收使用JavaScript对象标记指定的第二对象的代码,第二对象具有一个或多个属性的集合,每个属性与第一对象的属性的集合中的至少一个属性对应并且具有定义一个或多个变换的至少一个值,所述至少一个值包括要在第一对象的属性的集合中的对应的至少一个属性的值上执行的表达式;及

用于基于第二对象将第一对象变换为使用JavaScript对象标记指定的第三对象的代码。

13. 如权利要求12所述的非临时性计算机可读介质,其中用于接收第二对象的代码包括:用于接收使用JavaScript对象标记指定的、具有名称与第一对象的属性的名称对应的属性的对象的代码。

14. 如权利要求13所述的非临时性计算机可读介质,其中用于接收具有名称与第一对象的属性的名称对应的属性的对象的代码包括:用于接收具有与第一对象的属性相同名称的属性的代码。

15. 如权利要求13所述的非临时性计算机可读介质,其中用于接收具有名称与第一对象的属性的名称对应的属性的对象的代码包括:用于接收具有对第一对象的属性的名称进行引用的值的属性的代码。

16. 如权利要求12所述的非临时性计算机可读介质,其中用于接收第二对象的代码包括:用于接收使用JavaScript对象标记指定的对象的具有如下值的属性的代码,该值指定要应用到第一对象的属性的名称或第一对象的属性的值的操作。

17. 如权利要求12所述的非临时性计算机可读介质,其中用于接收第二对象的代码包括:用于接收使用JavaScript对象标记指定的对象的具有如下值的属性的代码,该值指定与第一对象的属性的变换对应的第三对象的属性的数据类型。

18. 如权利要求12所述的非临时性计算机可读介质,其中用于接收第二对象的代码包括:用于接收使用JavaScript对象标记指定的对象的具有如下值的属性的代码,该值指定对第一对象的一个或多个属性的修改。

19. 如权利要求12所述的非临时性计算机可读介质,其中用于接收第二对象的代码包括:用于接收使用JavaScript对象标记指定的对象的具有如下值的属性的代码,该值定义一个或多个测试,该一个或多个测试定义所述一个或多个变换是否能够应用到第一对象。

20. 如权利要求19所述的非临时性计算机可读介质,其中所述一个或多个测试是关于第一对象的一个或多个属性定义的。

21. 如权利要求19所述的非临时性计算机可读介质,还包括:

用于确定所述一个或多个测试的肯定评估的代码;及

用于基于所述一个或多个测试的肯定评估来执行所述一个或多个变换的代码。

22. 如权利要求21所述的非临时性计算机可读介质,其中用于执行所述一个或多个变换的代码包括:用于重命名第一对象的属性、关于第一对象的属性的值执行一个或多个数值操作、关于第一对象的属性的值执行一个或多个字符串操作、将一个或多个属性添加到第一对象的也作为对象的属性、或者从第一对象的也作为对象的属性移除一个或多个属性的代码。

23. 一种用于变换JSON数据的系统,所述系统包括:  
硬件处理器;及  
存储器,存储指令集合,当所述指令集合由所述处理器执行时,使所述处理器:  
接收使用JavaScript对象标记指定的第一对象,第一对象具有一个或多个属性的集合,每个属性具有预定的JSON数据类型并且具有至少一个值;  
接收使用JavaScript对象标记指定的第二对象,第二对象具有一个或多个属性的集合,每个属性与第一对象的属性的集合中的至少一个属性对应并且具有定义一个或多个变换的至少一个值,所述至少一个值包括要在第一对象的属性的集合中的对应的至少一个属性的值上执行的表达式;及  
基于第二对象将第一对象变换为使用JavaScript对象标记指定的第三对象。
24. 如权利要求23所述的系统,其中接收第二对象包括:接收使用JavaScript对象标记指定的、具有名称与第一对象的属性的名称对应的属性的对象。
25. 如权利要求24所述的系统,其中接收具有名称与第一对象的属性的名称对应的属性的对象包括:接收具有与第一对象的属性相同名称的属性。
26. 如权利要求24所述的系统,其中接收具有名称与第一对象的属性的名称对应的属性的对象包括:接收具有对第一对象的属性的名称进行引用的值的属性。
27. 如权利要求23所述的系统,其中接收第二对象包括:接收使用JavaScript对象标记指定的对象的具有如下值的属性,该值指定要应用到第一对象的属性的名称或第一对象的属性的值的操作。
28. 如权利要求23所述的系统,其中接收第二对象包括:接收使用JavaScript对象标记指定的对象的具有如下值的属性,该值指定与第一对象的属性的变换对应的第三对象的属性的数据类型。
29. 如权利要求23所述的系统,其中接收第二对象包括:接收使用JavaScript对象标记指定的对象的具有如下值的属性,该值指定对第一对象的一个或多个属性的修改。
30. 如权利要求23所述的系统,其中接收第二对象包括:接收使用JavaScript对象标记指定的对象的具有如下值的属性,该值定义一个或多个测试,该一个或多个测试定义所述一个或多个变换是否能够应用到第一对象。
31. 如权利要求30所述的系统,其中所述一个或多个测试是关于第一对象的一个或多个属性定义的。
32. 如权利要求30所述的系统,其中所述指令集合在由所述处理器执行时,还使所述处理器:  
确定所述一个或多个测试的肯定评估;及  
基于所述一个或多个测试的肯定评估来执行所述一个或多个变换。
33. 如权利要求32所述的系统,其中执行所述一个或多个变换包括:重命名第一对象的属性、关于第一对象的属性的值执行一个或多个数值操作、关于第一对象的属性的值执行一个或多个字符串操作、将一个或多个属性添加到第一对象的也作为对象的属性、或者从第一对象的也作为对象的属性中移除一个或多个属性。
34. 一种包括用于执行如权利要求1至11中任一项所述的方法的构件的装置。

## JSON样式表语言变换

### 背景技术

[0001] JSON是一种使用人类可读的文本来传输由属性-值对组成的数据对象的开放标准格式。它主要用来在服务器和Web应用之间传输数据,作为XML的替代方案。尽管最初是从JavaScript脚本语言派生的,但JSON是一种与语言无关(language-independent)的数据格式。用于解析和生成JSON数据的代码很容易以各种编程语言提供。

[0002] 目前在企业级,JSON数据格式已经获得了很大的关注,因为它比其它基于web的数据格式更灵活和更容易使用。它是移动世界中高度优选的数据格式。JSON被推广作为对其它数据格式的低开销替代方案,这些其它格式在其通常使用的现实世界情景中广泛支持创建、读取和解码。除了XML以外,其它示例还可以包括OGDL、YAML和CSV。

[0003] 为了变换JSON数据,在市场上可以获得许多转换API。然而,这些API不是用户友好的,因为它们可能需要用户学习另一种语言来定义变换。通常,用户最终编写自己的语法来这样做。一些API甚至将JSON转换为XML、应用变换、并且然后转换回JSON。这导致大量的开销处理。

[0004] 相应地,所期望的是解决与JSON变换相关的问题,其中一些问题可能在本文中讨论。此外,所期望的是减少与JSON变换语言相关的缺点,其中一些缺点可能在本文中讨论。

### 发明内容

[0005] 本公开的以下部分至少为了提供对主题的基本理解的目的而呈现在本公开中发现的一个或多个创新、实施例和/或示例的简化概要。本概要不试图提供任何特定实施例或示例的详尽概述。此外,本概要不意在识别实施例或示例的关键/紧要元件或描绘本公开的主题的范围。相应地,本概要的一个目的可以是以简化形式呈现在本公开内找到的一些创新、实施例和/或示例,作为稍后呈现的更详细描述的前言。

[0006] 提供了用于使用其它JSON对象来指定JSON对象的变换的系统和方法。接收使用JavaScript对象标记(JavaScript Object Notation)指定的第一对象。第一对象包括一个或多个属性的集合,其中每个属性具有预定的JSON数据类型并具有至少一个值。还接收使用JavaScript对象标记指定的第二对象。第二对象包括一个或多个属性的集合,每个属性与第一对象的属性的集合中的至少一个属性对应并具有定义一个或多个变换的至少一个值。基于使用第二对象变换第一对象而生成使用JavaScript对象标记指定的第三对象。

[0007] 在一个实施例中,一种用于变换JSON数据的方法包括在计算机处接收使用JavaScript对象标记指定的第一对象。第一对象可以具有一个或多个属性的集合,其中每个属性具有预定的JSON数据类型并具有至少一个值。接收使用JavaScript对象标记指定的第二对象,第二对象具有一个或多个属性的集合,每个属性与第一对象的属性的集合中的至少一个属性对应并且具有定义一个或多个变换的至少一个值。第一对象基于第二对象被变换为使用JavaScript对象标记指定的第三对象。

[0008] 在某些实施例中,在计算机处接收第二对象包括接收使用JavaScript对象标记指定的、具有名称与第一对象的属性的名称对应的属性的对象。接收具有名称与第一对象的

属性的名称对应的属性的对象可以包括接收与第一对象的属性具有相同名称的属性。接收具有名称与第一对象的属性的名称对应的属性的对象可以包括接收具有对第一对象的属性的名称进行引用的值的属性。

[0009] 在进一步的实施例中,在计算机处接收第二对象可以包括接收使用JavaScript对象标记指定的对象的属性,其中该属性具有指定要应用到第一对象的属性的名称或第一对象的属性的值的操作的值。在计算机处接收第二对象可以包括接收使用JavaScript对象标记指定的对象的属性,其中该属性具有指定第三对象的与第一对象的属性的变换对应的属性的数据类型的值。在计算机处接收第二对象可以包括接收使用JavaScript对象标记指定的对象的属性,其中该属性具有指定对第一对象的一个或多个属性的修改的值。

[0010] 在各种实施例中,在计算机处接收第二对象可以包括接收使用JavaScript对象标记指定的对象的属性,其中该属性具有定义一个或多个测试的值,该一个或多个测试定义一个或多个变换是否可应用到第一对象。可以关于第一对象的一个或多个属性来定义一个或多个测试。一个或多个测试的肯定评估(positive evaluation)可以被确定并用来执行一个或多个变换。由计算机执行一个或多个变换可以包括重命名第一对象的属性、关于第一对象的属性的值执行一个或多个数值操作、关于第一对象的属性的值执行一个或多个字符串操作、将一个或多个属性添加到第一对象的属性(该属性也是对象),或者从第一对象的属性(该属性也是对象)中移除一个或多个属性。

[0011] 在一个实施例中,非临时性计算机可读介质存储或嵌有可由计算机系统的处理器执行的、用于变换JSON数据的计算机程序产品。非临时性计算机可读介质可以包括:用于接收使用JavaScript对象标记指定的第一对象的代码,第一对象具有一个或多个属性的集合,每个属性具有预定的JSON数据类型并具有至少一个值;用于接收使用JavaScript对象标记指定的第二对象的代码,第二对象具有一个或多个属性的集合,每个属性与第一对象的属性的集合中的至少一个属性对应并具有定义一个或多个变换的至少一个值;以及用于基于第二对象将第一对象变换为使用JavaScript对象标记指定的第三对象的代码。

[0012] 在一个实施例中,用于变换JSON数据的系统可以包括硬件处理器;以及存储指令集合的存储器,当指令由处理器执行时,使处理器:接收使用JavaScript对象标记指定的第一对象,第一对象具有一个或多个属性的集合,每个属性具有预定的JSON数据类型并具有至少一个值;接收使用JavaScript对象标记指定的第二对象,第二对象具有一个或多个属性的集合,每个属性与第一对象的属性的集合中的至少一个属性对应并具有定义一个或多个变换的至少一个值;以及基于第二对象将第一对象变换为使用JavaScript对象标记指定的第三对象。

[0013] 除了上述部分,还通过参考本公开的其余部分、任何附图和权利要求,应当实现对本公开的主题的本质和等同物(以及所提供的任何固有或明确优点和改进)的进一步理解。

## 附图说明

[0014] 为了合理地描述和示出在本公开内找到的那些创新、实施例和/或示例,可以参考一个或多个附图。用来描述一个或多个附图的附加细节或示例不应当被认为是对本公开内呈现的任何要求保护的发明、任何当前描述的实施例和/或示例或者当前所理解的任何创新的最佳模式的范围的限制。

[0015] 图1是根据本公开实施例的系统环境的组件的简化框图,通过该系统环境,由实施例系统的组件提供的服务可以被提供作为移动云服务。

[0016] 图2是一个实施例中的用于变换JSON数据的工作流的简化框图。

[0017] 图3是一个实施例中的用于使用定义一个或多个变换的JSON对象来执行JSON数据变换的方法的流程图。

[0018] 图4是一个实施例中的用于确定如何使用一个对象的JSON数据属性语言来变换另一个对象的JSON数据的方法的流程图。

[0019] 图5是一个实施例中的用于确定如何使用JSON样式表(stylesheets)语言来变换JSON数据的方法的流程图。

[0020] 图6是一个实施例中的用于变换与JSON样式表语言对象的属性具有相同名称的JSON数据属性的方法的流程图。

[0021] 图7是一个实施例中的输入JSON字符串、JSL字符串以及通过使用JSL字符串对输入JSON字符串的属性进行变换而产生的输出字符串的图示。

[0022] 图8是一个实施例中的用于变换由JSON样式表语言对象的属性引用的JSON数据属性的方法的流程图。

[0023] 图9是一个实施例中的输入JSON字符串、JSL字符串以及通过使用JSL字符串对输入JSON字符串的属性值进行变换而产生的输出字符串的图示。

[0024] 图10是一个实施例中的用于对满足JSON样式表语言对象的属性的表达式的JSON数据属性进行变换的方法的流程图。

[0025] 图11是一个实施例中的输入JSON字符串、JSL字符串以及当满足表达式时通过使用JSL字符串对输入JSON字符串进行变换而产生的输出字符串的图示。

[0026] 图12是一个实施例中的输入JSON字符串、JSL字符串以及当满足表达式时通过使用JSL字符串对输入JSON字符串进行变换而产生的输出字符串的图示。

[0027] 图13是一个实施例中的被配置为使用JSON样式表语言对象执行JSON对象变换的硬件单元的框图。

[0028] 图14绘出了用于实现实施例之一的分布式系统的简化图。

[0029] 图15示出了在其中可以实现本发明的各种实施例的示例性计算机系统。

## 具体实施方式

### [0030] 引言

[0031] 在下面的描述中,为了解释的目的,阐述了具体的细节,以便提供对本发明实施例的透彻理解。然而将清楚的是,实施例可以在没有这些具体细节的情况下实践。例如,电路、系统、网络、过程和其它组件可以被示为框图形式的组件,以便不以不必要的细节模糊实施例。在其它情况下,众所周知的电路、过程、算法、结构和技术可以被示出而没有不必要的细节,以便避免模糊实施例。附图和描述不旨在是限制性的。相反,示例性实施例的随后的描述将为本领域技术人员提供用于实现示例性实施例的实现性描述(enabling description)。应当理解,在不脱离如所附权利要求中阐述的本发明的精神和范围的情况下,可以对元件的功能和布置进行各种改变。

[0032] 而且应当指出,各个实施例可以被描述为被绘制为流程图、流图、数据流图、结构

图或框图的过程。虽然流程图可以将操作描述为顺序过程,但是这些操作当中的许多操作可以并行或并发执行。此外,操作的次序可以被重新布置。过程在其操作完成时终止,但是可以具有不包括在图中的附加步骤。过程可以对应于方法、函数、过程、子例程,子程序等。当过程对应于函数时,其终止可以对应于该函数返回到调用函数或主函数。

[0033] 术语“机器可读介质”或“计算机可读介质”包括但不限于便携式或非便携式存储设备、光学存储设备、无线信道,以及能够存储、包含或携带(一个或多个)指令和/或数据的各种其它介质。代码段或机器可执行指令可以表示过程、函数、子程序、程序、例程、子例程、模块、软件包、类、或者指令、数据结构或程序语句的任意组合。代码段可以通过传递和/或接收信息、数据、自变量、参数或存储器内容而耦合到另一代码段或者硬件电路。信息、自变量、参数、数据等可以经由任何合适的手段(包括存储器共享、消息传递、令牌传递、网络传输等)被传递、转发或传输。

[0034] 此外,实施例可以由硬件、软件、固件、中间件、微代码、硬件描述语言或者其任意组合来实现。当在软件、固件、中间件或微代码中实现时,执行必要任务的程序代码或代码段可以被存储在机器可读介质或计算机可读介质中。一个或多个处理器可以执行这些必要任务。

[0035] 在一些图中绘出的系统可以以各种配置提供。在一些实施例中,系统可以被配置为分布式系统,其中系统的一个或多个组件跨云计算系统中的一个或多个网络分布。在进一步的实施例中,系统可以被配置为单个系统,其中系统的一个或多个组件结合到单个结构或包中。

[0036] 图1是根据本公开的实施例的系统环境100的一个或多个组件的简化框图,通过该系统环境100,由实施例系统的一个或多个组件提供的服务可以被提供作为云服务。在所示出的实施例中,系统环境100包括向一个或多个客户端计算设备104、106和108提供云服务的云基础设施系统102。客户端计算设备104、106和108可以被用户用来与云基础设施系统102交互。客户端计算设备104、106和108可以被配置为操作诸如web浏览器、专用客户端应用(例如,Oracle Forms)、或可以被客户端计算设备的用户用来与云基础设施系统102交互以使用由云基础设施系统102提供的服务的某种其它应用之类的客户端应用。

[0037] 应当认识到,图1中所绘出的云基础设施系统102可以具有除绘出的那些之外的其它组件。另外,图1中所示出的实施例只是可以结合本发明的实施例的云基础设施系统的一个示例。在一些其它的实施例中,云基础设施系统102可以具有比图1中所示更多或更少的组件、可以组合两个或更多个组件、或者可以具有组件的不同配置或布置。

[0038] 客户端计算设备104、106和108可以是便携式手持设备(例如,**iPhone®**、蜂窝电话、**iPad®**、计算平板、个人数字助理(PDA))或可穿戴设备(例如,Google**Glass®**头戴式显示器),这些设备运行诸如Microsoft Windows**Mobile®**的软件和/或诸如iOS、Windows Phone、Android、BlackBerry 10、Palm OS等的各种移动操作系统,并且这些设备启用了互联网、电子邮件、短消息服务(SMS)、**Blackberry®**或其它通信协议。客户端计算设备104、106和108可以是通用个人计算机,例如包括运行各种版本的Microsoft **Windows®**、Apple **Macintosh®**和/或Linux操作系统的个人计算机和/或膝上型计算机。客户端计算设备104、106和108可以是运行任何各种商业上可用的 **UNIX®**或类



UNIX操作系统(包括但不限于各种GNU/Linux操作系统,诸如Google Chrome OS)的工作站计算机。可替代地或附加地,客户端计算设备104、106和108可以是能够经(一个或多个)网络110进行通信的任何其它电子设备,诸如瘦客户端计算机、启用了互联网的游戏系统(例如,具有或不具有**Kinect®**手势输入设备的Microsoft Xbox游戏控制台)和/或个人消息传送设备。

[0039] 虽然示例性系统环境100被示为具有三个客户端计算设备,但是任何数量的客户端计算设备都可以被支持。诸如具有传感器的设备等的其它设备可以与云基础设施系统102交互。

[0040] (一个或多个)网络110可以便于客户端104、106和108与云基础设施系统102之间的通信和数据交换。(一个或多个)网络110可以是本领域技术人员熟悉的任何类型的网络,这些网络可以支持使用任何各种商业上可用的协议的数据通信,包括但不限于TCP/IP(传输控制协议/互联网协议)、SNA(系统网络体系架构)、IPX(互联网分组交换)、AppleTalk等。仅仅作为示例,(一个或多个)网络110可以是局域网(LAN),诸如基于以太网、令牌环等的局域网。(一个或多个)网络110可以是广域网和互联网。它可以包括虚拟网络,包括但不限于虚拟专用网络(VPN)、内联网、外联网、公共交换电话网络(PSTN)、红外网络、无线网络(例如,在电气和电子工程师协会(IEEE) 802.11协议套件、**Bluetooth®**和/或任何其它无线协议当中的任何协议下操作的网络);和/或这些和/或其它网络的任意组合。

[0041] 云基础设施系统102可以包括一个或多个计算机和/或服务器。这些计算机系统或服务器可以包括一个或多个通用计算机、专用服务器计算机(例如,包括PC(个人计算机)服务器、**UNIX®**服务器、中档服务器、大型计算机、机架式服务器等)、服务器场、服务器集群或任何其它适当的布置和/或组合。在各种实施例中,与云基础设施系统102关联的一个或多个计算机系统或服务器可以适于运行前述公开中描述的一个或多个服务或软件应用。例如,与云基础设施系统102关联的一个或多个计算机系统或服务器可以对应于用于执行本文所述的根据本公开实施例的处理的服务器。

[0042] 与云基础设施系统102关联的一个或多个计算机系统或服务器可以运行包括上面讨论的任何操作系统的操作系统,以及任何商业上可用的服务器操作系统。与云基础设施系统102关联的一个或多个计算机系统或服务器还可以运行各种附加的服务器应用和/或中间层应用中的任何应用,包括HTTP(超文本传输协议)服务器、FTP(文件传输协议)服务器、CGI(公共网关接口)服务器、**JAVA®**服务器、数据库服务器等。示例性数据库服务器包括但不限于可从Oracle、Microsoft、Sybase、IBM(国际商业机器)等商业获得的数据库服务器。

[0043] 在某些实施例中,由云基础设施系统102提供的服务可以包括可以由云基础设施系统102的用户按需使用的许多服务,诸如在线数据存储和备份解决方案、基于web的电子邮件服务、托管的办公套件和文档协作服务、数据库处理、受管理的技术支持服务等。由云基础设施系统102提供的服务可以动态地缩放,以满足其用户的需求。由云基础设施系统102提供的服务的具体实例化在本文中称作为“服务实例”。一般而言,来自云服务提供商的系统的、经由诸如互联网的通信网络对用户可用的任何服务被称为“云服务”。典型地,在公共云环境中,构成云服务提供商的系统的服务器和系统与用户自己的办公场所(on-premises)服务器和系统不同。例如,云服务提供商的系统可以托管应用,并且用户可以经

由诸如互联网之类的通信网络按需订阅和使用该应用。

[0044] 在一些示例中,由云基础设施102实例化的服务实例可以包括对由云供应商提供给用户或者以本领域中已知的其它方式提供的存储、托管数据库、托管web服务器、软件应用或其它服务的受保护的计算机网络访问。例如,由云基础设施102实例化的服务实例可以包括通过互联网对云上远程存储的密码保护的访问。作为另一个示例,由云基础设施102实例化的服务实例可以包括用于联网开发人员私人使用的基于web服务的托管关系数据库和脚本语言中间件引擎。作为另一个示例,由云基础设施102实例化的服务实例可以包括对在云供应商的web站点上托管的电子邮件软件应用的访问。

[0045] 在某些实施例中,云基础设施系统102可以包括以自助服务、基于订阅、弹性可缩放、可靠、高可用性并且安全的方式交付给客户的一套应用、中间件、开发服务和数据库服务提供物(offerings)。具体化为云基础设施服务102的这种云基础设施系统的示例是由本受让人提供的Oracle Public Cloud(Oracle公共云)。

[0046] 云基础设施系统102可以经由不同的部署模型提供云服务。例如,服务可以在公共云模型下提供,在公共云模型中云基础设施系统102由销售云服务的组织拥有(例如,被Oracle拥有)并且使得服务对一般公众和不同行业的企业可用。作为另一个示例,服务可以在私有云模型下提供,在私有云模型中云基础设施系统102只为单个组织运营并且可以为该组织内的一个或多个实体提供服务。云服务还可以在社区云模型下提供,在社区云模型中云基础设施系统102和由云基础设施系统102提供的服务由相关社区中的若干个组织共享。云服务也可以在混合云模型下提供,该混合云模型是两种或更多种不同模型的组合。

[0047] 在一些实施例中,由云基础设施系统102提供的服务可以包括在软件即服务(SaaS)类别、平台即服务(PaaS)类别、基础设施即服务(IaaS)类别、或包括混合服务的其它服务类别下提供的一个或多个服务。在一些实施例中,由云基础设施系统102提供的服务可以包括但不限于应用服务、平台服务和基础设施服务。在一些示例中,应用服务可以经由SaaS平台由云基础设施系统102提供。SaaS平台可以被配置为提供落在SaaS类别下的云服务。例如,SaaS平台可以提供在集成开发和部署平台上构建和交付一套按需应用的能力。SaaS平台可以管理和控制用于提供SaaS服务的底层软件和基础设施。通过使用由SaaS平台提供的服务,客户可以利用在云基础设施系统上执行的应用。客户可以获取应用服务,而无需客户购买单独的许可和支持。可以提供各种不同的SaaS服务。示例包括但不限于,提供用于大型组织的销售绩效管理、企业集成和业务灵活性的解决方案的服务。

[0048] 在一些实施例中,平台服务可以经由PaaS平台由云基础设施系统102提供。PaaS平台可以被配置为提供落在PaaS类别下的云服务。平台服务的示例可以包括但不限于使组织(诸如Oracle)能够在共享、公共的体系架构上整合现有应用、以及整合构建利用由平台提供的共享服务的新应用的能力的服务。PaaS平台可以管理和控制用于提供PaaS服务的底层软件和基础设施。客户可以获取由云基础设施系统102提供的PaaS服务,而无需客户购买单独的许可和支持。平台服务的示例包括但不限于,Oracle Java云服务(Oracle Java Cloud Service,JCS)、Oracle数据库云服务(Oracle Database Cloud Service,DBCS)以及其它服务。

[0049] 通过利用由PaaS平台提供的服务,客户可以采用由云基础设施系统102支持的编程语言和工具并且还可以控制所部署的服务。在一些实施例中,由云基础设施系统102提供

的平台服务可以包括数据库云服务、中间件云服务 (例如, Oracle Fusion Middleware 服务) 和 Java 云服务。在一个实施例中, 数据库云服务可以支持使组织能够汇集数据库资源并且以数据库云的形式为客户提供数据库即服务的共享服务部署模型。在云基础设施系统中, 中间件云服务可以为客户提供平台来开发和部署各种业务应用, 并且 Java 云服务可以为客户提供部署 Java 应用的平台。

[0050] 各种不同的基础设施服务可以由云基础设施系统102中的IaaS平台提供。基础设施服务便于对诸如存储、网络和其它基础计算资源之类的底层计算资源的管理和控制, 以便让客户利用由SaaS平台和PaaS平台提供的服务。

[0051] 在某些实施例中, 云基础设施系统102可以提供对云基础设施系统中的云服务 (例如, SaaS、PaaS和IaaS服务) 的综合管理。在一个实施例中, 云管理功能可以包括用于供应、管理和跟踪由云基础设施系统102接收到的客户的订阅等的功能。在各种实施例中, 云基础设施系统102可以适于自动供应、管理和跟踪客户对由云基础设施系统102提供的服务的订阅。客户经由订阅订单可以订购由云基础设施系统102提供的一个或多个服务云基础设施系统102然后执行处理, 以提供客户的订阅订单中的服务。

[0052] 在一个实施例中, 如图1中所绘出的, 云管理功能可以由诸如订单管理和监视模块114之类的一个或多个模块提供。这些模块可以包括一个或多个计算机和/或服务器或者使用一个或多个计算机和/或服务器来提供, 该一个或多个计算机和/或服务器可以是通用计算机、专用服务器计算机、服务器场、服务器集群、或任何其它适当的布置和/或组合。

[0053] 在示例性操作中, 使用客户端设备 (诸如客户端设备104、106或108中的一个或多个) 的客户可以通过请求由云基础设施系统102提供的一个或多个服务来与云基础设施系统102交互。客户可以使用各种方式向云基础设施系统102发布服务请求134。服务请求134可以包括对由云基础设施系统102提供的一个或多个服务等下订阅订单、访问由云基础设施系统102提供的一个或多个服务等。在某些实施例中, 客户可以访问云用户接口 (UI) (云UI 132、云UI 134和/或云UI 138) 并且经由这些UI下订阅订单。响应于客户下订单而被云基础设施系统102接收到的订单信息可以包括识别客户的信息以及识别客户想要订阅的由云基础设施系统102提供的一个或多个服务的信息。在客户已经下订单之后, 订单信息经由云UI 132、134和/或138被接收。

[0054] 在这个示例中, 订单管理和监视模块114将从客户接收的信息发送到订单数据库, 以使客户下达的订单存储在记录中。订单数据库可以是由云基础设施系统102操作并与其它系统元件结合操作的若干数据库之一。订单管理和监视模块114可以将包括存储在订单数据库中的订单信息的全部或部分的信息转发到订单管理模块。在一些情况下, 订单管理模块可以被配置为执行与订单相关的计费 and 记帐功能, 诸如验证订单并且在通过验证时订立 (book) 订单。

[0055] 订单管理和监视模块114可以将订单信息的全部或部分传送到订单编排模块, 订单编排模块利用订单信息来为客户下达的订单编排服务和资源的供应。在一些情况下, 订单编排模块可以使用订单供应模块的服务来编排资源的供应, 以支持所订阅的服务。

[0056] 在某些实施例中, 订单编排模块使得能够管理与每个订单关联的业务过程并且应用业务逻辑以确定订单是否应该继续供应。当接收到针对新订阅的订单时, 订单编排模块发送请求到订单供应模块, 以分配资源和配置履行订阅订单所需的那些资源。订单供应模

块使得能够分配用于客户所订阅的服务的资源。订单供应模块在由云基础设施系统102提供的云服务 and 用于提供所请求的服务供应资源的物理实现层之间提供一层抽象。订单编排模块可以因此与实现细节隔离,诸如服务和资源是实际上被即时 (on the fly) 供应还是预先被供应并且只在请求时被分配/指派。

[0057] 在某些实施例中,订单管理和监视模块114管理和跟踪客户的订阅订单。在一些情况下,订单管理和监视模块114可以接收指示与客户关联的任何所供应的服务和/或资源的信息。订单管理和监视模块114可以被配置为收集对于订阅订单中的服务的使用统计,诸如所使用的存储量、传送的数据量、用户数量以及系统正常运行时间和系统停机时间的量。

[0058] 一旦在上面的示例中供应了服务和资源,服务结果138就可以被发送到客户端计算设备104、106和/或108上的客户,从而通知客户所提供的服务和/或资源。在服务请求130包括访问服务或者让服务执行一个或多个操作的请求的情况下,服务结果138可以被发送到客户端计算设备104、106和/或108上的客户,从而提供所请求的访问或任何操作的结果、所执行的服务或所请求的数据。

[0059] 在某些实施例中,云基础设施系统100可以包括身份管理模块114。身份管理模块114可以被配置为提供身份服务,诸如在云基础设施系统102中的访问管理和授权服务。在一些实施例中,身份管理模块114可以控制关于希望利用由云基础设施系统102提供的服务的客户的信息。这种信息可以包括验证这种客户的身份的信息和描述这些客户被授权相对于各种系统资源(例如,文件、目录、应用、通信端口、存储器段、等等)执行哪些动作的信息。身份管理模块114还可以包括对关于每个客户的描述性信息和关于该描述性信息可以如何访问和修改以及由谁来访问和修改的信息的管理。

[0060] 在某些实施例中,云基础设施系统102还可以包括基础设施资源116,用于提供被用来向云基础设施系统102的客户提供各种服务的资源。在一个实施例中,基础设施资源116可以包括硬件(诸如服务器、储存器和联网资源)的预集成和优化的组合,以执行PaaS平台和SaaS平台提供的服务。

[0061] 在一些实施例中,云基础设施系统102中的资源可以由多个用户共享并且根据需求动态地重新分配。此外,可以向不同时区中的用户分配资源。例如,云基础设施系统102可以使第一时区中的第一用户集合能够在指定的小时数内利用云基础设施系统的资源,然后使得能够将相同的资源重新分配给位于不同时区中的另一用户集合,由此最大化资源的利用。

[0062] 在某些实施例中,可以提供由云基础设施系统102的不同组件或模块以及由云基础设施系统102提供的服务共享的多个内部共享服务118。这些内部共享服务118可以包括但不限于安全和身份服务、集成服务、企业储存库服务、企业管理器服务、病毒扫描和白名单服务、高可用性、备份和恢复服务、用于启用云支持的服务、电子邮件服务、通知服务、文件传输服务等。

[0063] 在某些实施例中,可以提供由云基础设施系统102的不同组件或模块以及由云基础设施系统102提供的服务共享的多个外部共享服务120。这些外部共享服务120可以包括但不限于安全和身份服务、集成服务、企业储存库服务、企业管理器服务、病毒扫描和白名单服务、高可用性、备份和恢复服务、用于启用云支持的服务、电子邮件服务、通知服务、文件传输服务等。

[0064] 在各种实施例中,外部共享服务120可以包括向(一个或多个)企业计算机系统126提供访问、数据变换、自动化等的一个或多个组件。对(一个或多个)企业计算机系统126的访问可以由云基础设施系统102的不同组件或模块以及由云基础设施系统102提供的服务共享。在一些实施例中,对(一个或多个)企业计算机系统126的访问可以由云基础设施系统102提供的受限一个或多个订户的服务实例共享。

[0065] 在进一步的实施例中,外部共享服务120可以包括由云基础设施系统102的不同组件或模块以及由云基础设施系统102提供的服务共享的外部应用编程接口(API)服务128。这些外部API服务128可以包括但不限于由其它第三方服务或实体提供的API。

[0066] 各种不同的移动云服务可以由云基础设施系统102中的移动云服务(MSC)122提供。根据本发明的一些实施例,MCS 122便于移动计算设备与企业计算机系统(例如,企业计算机系统124和126)之间的通信。MCS 122可以包括被用来存储企业数据和认证信息的一个或多个存储器存储设备(“本地储存器”)。企业数据可以从企业计算机系统126或从客户端计算设备104、106或108接收、或者可以包括由云基础设施系统102转换的企业数据,或者是它们的组合。认证信息可以从身份管理系统116接收和/或由云基础设施系统102生成。在一些实施例中,认证信息可以包括关于对服务的请求指示用户的安全认证的信息。

[0067] 诸如企业计算机系统126之类的企业计算机系统可以在与云基础设施系统102不同的地理位置(例如,远程地理位置)处在物理上位于云基础设施系统102的防火墙之外。在一些实施例中,企业计算机系统126可以包括一个或多个不同的计算机或服务器。在一些实施例中,企业计算机系统126可以是单个计算机系统的一部分。

[0068] 在某些实施例中,企业计算机系统126可以使用一个或多个不同的协议与云基础设施系统102通信。每个企业计算机系统126可以使用不同的通信协议与云基础设施系统102通信。企业计算机系统126可以支持相同或不同的安全协议。在一些实施例中,MSC 1112可以包括代理系统,以处置与企业计算机系统126的通信。

[0069] 协议可以包括通信协议,诸如SPDY。协议可以包括应用协议,诸如基于HTTP的协议。在一些实施例中,企业计算机系统126可以使用REST或SOAP通信协议与云基础设施系统102通信。例如,REST协议可以支持包括URI或URL的格式。为了使用REST协议进行通信而格式化的企业数据可以容易地转换成诸如JSON、逗号分隔值(CSV)和真正简单的聚合(RSS)的数据格式。企业计算机系统126和云基础设施系统102可以使用诸如远程过程调用(RPC)(例如,XML RPC)的其它协议进行通信。

[0070] 在一些实施例中,MCS 122可以包括被配置为支持与由云基础设施服务102提供的一个或多个服务的通信的适配器接口,其中一些可以支持不同协议或技术以进行通信。在一些实施例中,MCS 122可以包括被配置为支持与企业计算机系统126进行通信的适配器接口,其中一些可以支持不同协议或技术以进行通信。MCS 122可以包括一个或多个适配器,每个适配器可以被配置为根据通信协议、企业计算机系统的类型、应用的类型、服务的类型或其组合进行通信。适配器支持的通信协议可以特定于服务或企业计算机系统126中的一个或多个。

[0071] 在某些实施例中,客户端计算设备104、106和108可以各自实现可以提供具体用户接口以与MCS 122通信的应用。具体UI可以被配置为使用具体通信协议进行通信。在一些实施例中,具体UI可以包括可以被调取以与MCS 122通信的可调用接口、函数、例程、方法和/

或操作。具体UI可以接受用于企业数据和/或请求服务与云基础设施服务102提供的服务或者与企业计算机系统126通信的参数,作为输入。在一些实施例中,通过MCS 122的通信可以被转换以使用定制通信协议进行通信。在一些实施例中,具体UI可以对应于应用中的定制客户端。

[0072] MCS 122可以包括一个或多个可调用接口,例如应用编程接口(API)。与MCS 122关联的可调用接口可以使移动计算设备上的应用能够向MCS 122传送请求。与MCS 122关联的可调用接口可以支持公共或标准接口,这可以允许请求包括根据标准化协议、体系架构风格和/或格式(例如,REST协议)从应用接收的其参数。与MCS 122关联的可调用接口可以由计算设备104、106或108中任何一个的用户配置。与MCS 122关联的可调用接口可以根据通信协议接收对服务的请求。设备应用开发人员为了他们的定制应用可以连接到MCS 122。在一些实施例中,与MCS 122关联的可调用接口可以由开发应用的同一个人配置,使得该人可以实现定制应用以与MCS 122通信。

[0073] 与MCS 122关联的可调用接口还可以使企业计算机系统126能够根据标准化的协议或格式与MCS 122通信。与应用开发人员类似,管理企业计算机系统的人可以实现被配置为经由一个或多个可调用接口与MCS 122通信的代码(例如,代理系统)。与MCS 122关联的可调用接口可以基于计算设备的类型、企业计算机系统的类型、应用、代理系统、服务、协议或其它标准来实现。在一些实施例中,与MCS 122关联的可调用接口可以支持对服务的请求,这些服务包括认证、压缩、加密、使用游标的分页、基于客户端的节流(throttling)、不可抵赖性(non-repudiation)、日志记录和度量数据收集。在一些实施例中,与MCS 122关联的可调用接口可以被实现用于与定制的业务相关的服务,诸如认证、策略实施、响应的高速缓存、对MCS 122的调用的节流、在异步和同步模式之间的转换、对底层服务的调用的日志记录,或者它们的组合。在一些实施例中,与MCS 122关联的可调用接口可以使用户能够加载用于由云基础设施系统102实现的定制代码。定制代码可以实现与用于云基础设施系统102的MCS 122关联的一个或多个可调用接口,这可以使用户能够访问定制服务或其它企业计算机系统。

[0074] 与MCS 122关联的协议翻译器可以处理消息,以确定用于消息的通信协议和/或将消息转换为用于目的地的通信协议。与MCS 122关联的协议翻译器可以转换从客户端计算设备104、106或108接收的请求。该请求可以从客户端计算设备104、106或108所支持的通信协议的格式转换为由云基础设施服务102提供的服务或企业计算机系统126所支持的通信协议的格式。与MCS 122关联的协议翻译器可以转换从由云基础设施服务102提供的服务或企业计算机系统126接收的响应。响应可以从由云基础设施服务102提供的服务或企业计算机系统126所支持的通信协议的格式转换成客户端计算设备104、106或108所支持的通信协议的格式。

[0075] 与MCS 122关联的安全服务可以管理对从客户端计算设备104、106或108当中任一个接收的请求的安全认证。与MCS 122关联的安全服务可以保护客户过程和企业数据的完整性。为了防止系统或数据被破坏,当从客户端计算设备104、106或108接收到请求时,可以发生安全认证。安全认证可以在调度请求以由云基础设施系统102处理之前执行。为用户确定的安全认证可以使得与移动计算设备关联的用户能够具有经由MCS 122请求服务的授权。安全认证可以针对用户减少对不同请求和/或经由MCS 122请求的服务进行认证的工

作。与MCS 122关联的安全服务可以被实现为被配置为执行认证请求的安全性的各种操作的一个或多个功能块或模块。

[0076] 与MCS 122关联的认证服务可以管理对从客户端计算设备104、106或108接收的请求的安全认证。与MCS 122关联的认证服务可以确定与向MCS 122发送请求的计算设备关联的用户的安全认证。安全认证可以基于时间段来确定,该安全认证可以绑定到应用的操作(例如,启动应用)、请求、计算设备、企业计算机系统、与请求相关的其它标准或其组合。安全认证可以针对以下任何一个进行验证和准许,诸如单独的请求、一个或多个企业计算机系统、特定服务、服务类型、用户、计算设备、用于确定安全认证的其它标准,或者它们的组合。在一些实施例中,云基础设施系统102可以存储从企业计算机系统或支持企业计算机系统的认证系统接收的用户的认证信息。云基础设施系统102可以通过执行查找功能以确定与请求关联的用户的身份是否具有做出这种请求的授权来确定认证。所存储的认证信息可以包括用户可以被授权访问的诸如请求的类型、功能、企业计算机系统、企业数据等的信息。在一些实施例中,基础设施系统102可以发起与发出请求的计算设备的通信以确定认证。

[0077] 在一些实施例中,安全认证可以基于与请求服务的用户关联的角色来确定。角色可以与请求访问MCS 122的用户关联。在一些实施例中,用户可以作为MCS 122的订户或租户来请求服务,其中这种MCS 122的订户或租户可以被准许访问由MCS 122提供的资源和/或服务。认证可以对应于用户对MCS 122的订阅,使得用户可以被授权作为订户经由MCS 122请求服务。在一些实施例中,订阅可以限于由MCS 122提供的特定资源集合。安全认证可以基于MCS 122的用户可访问的资源和/或服务。在一些实施例中,可以在被称作“运行时环境”的执行期间为请求供应模板。运行时环境可以与为请求、用户或设备分配的资源关联。

[0078] 在一些实施例中,与MCS 122关联的认证服务可以请求身份管理系统确定用于用户的安全认证。身份管理系统可以由云基础设施系统102(例如,作为身份管理114)或由在云基础设施系统102外部的另一个计算机系统实现。身份管理116可以基于用户的角色或者对访问MCS 122的订阅来确定用户的安全认证。可以关于企业计算机系统、企业计算机系统提供的服务、企业计算机系统的功能或特征、用于控制对企业计算机系统的访问的其它标准或者它们的组合向角色或订阅指派特权和/或权利。

[0079] 各种不同的应用开发框架(ADF)可以由云基础设施系统102中的应用开发框架(ADF) 124提供。ADF 124提供基础设施代码,以实现基于灵活(agile) SOA的应用。ADF 124还通过一个或多个开发工具(例如,Oracle JDeveloper 11g开发工具)为开发提供可视和声明性方法。由ADF 124提供的一个或多个框架可以实现模型-视图-控制器设计模式。这种框架提供集成的解决方案,通过对诸如对象/关系映射、数据持久性、可重用控制器层、丰富的Web用户界面框架、数据到UI的绑定、安全性和定制之类的区域的解决方案而覆盖MVC体系架构的所有层。除了核心的基于Web的MVC方法之外,这种框架还与Oracle SOA和WebCenter Portal(网络中心门户)框架集成,从而简化了完整复合应用的创建。

[0080] 在某些实施例中,ADF 124使得容易开发通过将服务接口耦合到由云基础设施系统102提供的内置业务服务来将数据暴露作为服务的灵活应用。业务服务实现细节的这种分离在ADF 124中经由元数据执行。这种元数据驱动的体系架构的使用使应用开发人员能够专注于业务逻辑和用户体验,而不是服务被如何访问的细节。在某些实施例中,ADF 124



在模型层中的元数据中存储服务的实现细节。这使开发人员能够在不修改用户接口的情况下交换服务,从而使应用非常灵活。此外,创建用户接口的开发人员不需要操心业务服务访问细节。相反,开发人员可以专注于开发应用接口和交互逻辑。创建用户体验可以像将期望的业务服务拖放到可视页面设计器上并指示应当由哪种类型的组件表示那个数据一样简单。

[0081] 在各种实施例中,开发人员与ADF 124交互,以创建形成企业应用的模块。企业应用可以在云基础设施系统102的上下文内执行。在各种实施例中,开发人员与ADF 124交互,以创建形成移动应用的模块。移动应用可以在云基础设施系统102的上下文内执行。下面描述的本发明的特征可以使用编程语言和应用开发框架的任何期望的组合实现,通过阅读本文提供的公开,这对于相关领域的技术人员将是清楚的。

[0082] 在一个示例中,由ADF 124提供的一个或多个框架可以体现为Oracle ADF。相应地,ADF 124中的框架可以基于模型-视图-控制器(MVC)设计模式。MVC应用被分为:1) 处置与数据源的交互并运行业务逻辑的模型层,2) 处置应用用户接口的视图层,以及3) 管理应用流并充当模型层和视图层之间的接口的控制器。将应用分为这三个层简化了组件跨应用的维护和重用。每个层相对于其它层的独立性导致松散耦合的面向服务的体系架构(SOA)。

[0083] 在各种实施例中,ADF 124提供允许开发人员创建多层的形式的应用的工具和资源,每层包含根据预定义的规范实现期望逻辑的代码模块/文件。因此,在一个实施例中,ADF 124使应用能够被开发为四层:包含提供应用的用户接口的代码模块/文件的视图层,包含控制应用的流的代码模块的控制器层,包含为底层数据提供抽象层的数据/代码模块的模型层,以及包含提供对来自各种源的数据的访问并处置业务逻辑的代码模块的业务服务层。

[0084] 在某些实施例中,ADF 124让开发人员选择当实现这些层中的每个层时他们偏好使用的技术。EJB、Web服务、JavaBeans、JPA/EclipseLink/TopLink对象以及许多其它的技术全都可以被用作作用于ADF 124的业务服务。视图层可以包括使用JSF、Desktop Swing应用和MS Office前端实现的基于Web的接口,以及用于移动设备的接口。

[0085] 在一方面,视图层表示正在被开发的应用的接口。视图层可以包括桌面视图、移动视图和基于浏览器的视图,其中每个视图提供接口的全部或部分,并且以对应于视图类型的各种方式可访问。例如,网页可以由应用响应于接收到包含对应URL的客户端请求而被发送。然后,网页可以由与发出请求的客户端系统关联的显示单元(未示出)上的浏览器显示,由此使发出请求的客户端系统的用户能够与企业应用交互。ADF 124支持对业务服务的多信道访问,从而允许业务服务的重用和从Web客户端、基于客户端-服务器swing桌面的应用、Microsoft Excel电子表格、诸如智能电话的移动设备等进行访问。

[0086] 可以使用超文本标记语言(HTML)、Java服务器页面(JSP)和Java服务器面(JSF)中的一个或多个来实现形成视图层(诸如网页)的代码文件/模块。可替代地,接口可以使用诸如Swing之类的Java组件和/或可扩展标记语言(XML)来实现。如进一步提及的,接口可以充分利用用户对桌面应用(诸如Microsoft的Word和Excel)的经验和熟悉性。

[0087] 如上面所提及的,在每个层中提供相关的用户开发的代码/数据模块。然而,每个层通常包含由ADF 124提供的其它预定义的代码/数据模块。这些预定义的模块当中的一些可以在开发期间使用,例如,作为用于开发网页的模板,用于在所开发的代码中包括期望的



功能,等等。其它预定义模块(诸如URL重写模块)可以与所开发的应用一起部署,并且可以在企业应用执行期间向用户提供附加的功能(将所请求的URL映射到内部名称)。

[0088] 控制器层包含控制应用的流的代码模块/文件。每个控制器对象包含根据在视图层中呈现信息的期望方式实现的软件指令和/或数据。期望的方式可以包括当用户点击/选择另一个网页中的链接时要显示的特定网页、当执行期间发生错误时要显示的页面、指示要存储/检索的具体数据,等等。

[0089] 在一方面,控制器层管理应用流并处置用户输入。例如,当在页面上点击搜索按钮时,控制器确定要执行的动作(进行搜索)和导航到的位置(结果页面)。在JDeveloper中对于基于Web的应用有两个控制器选项:标准JSF控制器或扩展JSF控制器功能的ADF控制器。无论使用哪种控制器,应用流通常都是通过通过在图表上布置页面和导航规则来设计的。应用的流可以被分解为更小的、可重用的任务流;包括非可视组件,诸如流中的方法调用和决策点;并创建在单个包含页面的区域内部运行的“页面片段(page fragment)”流。

[0090] 形成控制器层的代码模块/文件常常被实现为接收客户端请求并将期望的网页作为对应响应发送的Java伺服小程序(servlet)。控制器对象也可以被实现为例如Apache Jakarta Struts控制器,或根据JSF标准实现。

[0091] 模型层包含将各种业务服务连接到在其它层中使用它们的对象(诸如连接到上面讨论的控制器对象或直接连接到如所示的桌面应用)的数据/代码模块。模型层的每个抽象数据对象提供可以被用来访问在底层业务服务层中执行的任何类型的业务服务的对应接口。数据对象可以从客户端抽象服务的业务服务实现细节和/或向视图组件暴露数据控制方法/属性,从而提供视图层和数据层的分离。

[0092] 在一方面,模型层由利用元数据文件来定义接口的两个组件(数据控件和数据绑定)组成。数据控件从客户端抽象业务服务实现细节。数据绑定向UI组件暴露数据控件方法和属性,从而提供视图和模型的干净分离。由于模型层的元数据体系架构,当将任何类型的业务服务层实现绑定到视图层和控制器层时,开发人员获得相同的开发体验。

[0093] 在某些实施例中,ADF 124强调贯穿整个开发过程使用声明性编程范例,以允许用户专注于应用创建的逻辑,而不必涉及实现细节。在高级别,用于Fusion web应用的开发过程通常涉及创建应用工作空间。使用向导,由开发人员选择的技术所需的库和配置被自动添加,并且应用被结构化为具有包和目录的项目。

[0094] 通过对数据库对象进行建模,可以创建任何数据库的在线数据库或离线副本、编辑定义、以及更新模式(schema)。使用UML建模器,然后可以为应用创建用例。还可以设计应用控制和导航。图形绘制器可以被用来可视确定应用控制和导航的流。然后,可以自动创建描述流的底层XML文件。资源库可以被用来允许开发人员通过简单地将导入的库拖放到应用中来查看和使用库。从数据库表,可以使用向导或对话框创建实体对象。从这些实体对象,创建视图对象以由应用中的页面使用。可以实现验证规则和其它类型的业务逻辑。

[0095] 在这个示例中,业务服务层管理与数据持久化层的交互。它提供诸如数据持久性、对象/关系映射、事务管理和业务逻辑执行之类的服务。业务服务层可以以以下任何选项实现:作为简单的Java类、EJB、Web服务、JPA对象和Oracle ADF业务组件。此外,数据可以直接从文件(XML或CSV)以及REST中消费。因此,每个业务服务管理与对应的数据持久化层的交互,并且还提供诸如对象/关系映射、事务管理、业务逻辑执行等服务。业务服务层可以使用

简单的Java类、企业Java Beans、web服务等其中的一个或多个来实现。

[0096] 业务组件表示使用例如Oracle ADF业务组件实现的业务服务,以提供与数据库、web服务、遗留系统、应用服务器等的交互。在一个实施例中,业务服务层的业务组件包含合作以提供业务服务实现的应用模块、视图/查询对象和实体对象的混合。应用模块可以是UI客户端为了与应用/事务数据一起工作而与其通信的事务组件/代码模块。应用模块可以提供可更新的数据模型以及与用户事务相关的过程/函数(通常被称为服务方法)。

[0097] 实体对象可以表示数据库表中的对应行并且简化对存储在对应行中的数据的操纵(更新、删除等)。实体对象常常封装用于对应行的业务逻辑,以确保始终实施期望的业务规则。实体对象还可以与其它实体对象关联,以反映存储在底层数据库中的行之间存在的关系。

[0098] JSON样式表语言变换

[0099] 图2是一个实施例中的用于变换JSON数据的工作流200的简化框图。在工作流200中,在JSL API 220处接收输入JSON数据210。在JSL API 220处接收JSL 230(也是JSON文档)。JSL API 220基于输入JSON数据210和JSL 230生成变换后的JSON数据240。

[0100] 在这个示例性实施例中,一般而言,输入JSON数据210是指定数据的一部分或对象的逻辑结构的JSON文档。输入JSON数据210的逻辑结构可以根据一个或多个JSON模式(schema)来定义。

[0101] JSL 230也是JSON文档。处理器(例如,JSL API 220)可以理解JSL 230的JSL(JSON样式表语言)并且对输入JSON数据210应用一个或多个指定的变换。这类似于用来对XML应用变换的XSLT。JSL的主要优点之一是用户不需要熟悉其它语言的规范。这是因为JSL本身是以JSON数据格式编写的,类似于定义XML变换的XSL。

[0102] 在市场上,存在许多变换JSON数据的变换工具。但是这些工具中的大多数是用单独的语言编写的,这使得用户难以处置。例如,JsonT是一种用于变换类JSON结构的语言,类似于XML世界中的XSLT。这种实现不像JSL那么直接。在另一个示例中,json2json是用CoffeeScript编写的并被设计为在Node.js环境中运行。这种实现的缺点是用户必须学习Coffee语言来创建用于变换的模板文件。然而,在JSL中,用户为变换指定的模板是以JSON编写的。

[0103] 在各种实施例中,JSL 230是根据JSL规范定义的。JSL可以包含配置处理器以实现输入JSON数据210的变换的关键字。在一方面,在JSL文件中提供“value(值)”关键字以命令处理器如何表示输入JSON数据中的指定键的值。“value”关键字可以用值来定义。在一个实施例中,值可以是指示输入JSON数据210中的键的词汇路径(lexical path)的词汇字符串。这个路径可以以“root(根)”关键字(例如,root.employee.id)开始,其中employee(雇员)是对象,id是字符串值属性。

[0104] 在另一个实施例中,值可以是文本字符串,该文本字符串指示在单引号之间定义以被视为值的任何文本。在还有另一个实施例中,值可以是运算字符串,该运算字符串指示词汇/文本字符串与在它们上定义的算术运算/字符串运算/逻辑运算的组合。例如,“concat(root.employee.id,'abc')”指示当插入到变换后的JSON数据240中时,id字段的值将与字符串‘abc’联接(concatenate)。

[0105] 在另一方面,在JSL文件中提供“type(类型)”关键字,以命令处理器如何表示具体

键的值的数据类型。在各种实施例中，“type”关键字可以用值来定义。在某些实施例中，值可以是整数(integer)、字符串(string)、双精度(double)、布尔型(Boolean)、空(null)等。对于JSONArray(JSON数组)元素，值可以被指定为objectarray(对象数组)、<Primitive Data Type>(基本数据类型)数组——例如Stringarray(字符串数组)等。

[0106] 在还有另一方面，在JSL文件中提供“source(源)”关键字，以命令处理器如何表示实体的父代的词汇路径。在各种实施例中，仅为数组元素定义该关键字。在另一方面，在JSL文件中提供“testexpression(文本表达式)”关键字，以命令处理器如何相对于一个或多个指定的条件评估输入JSON数据210。这个关键字可以被用来评估将基于其来处理具体JSON对象或JSON数组的表达式。该表达式可以返回布尔值。在对象级，基于“testexpression”结果，处理器可以决定对象是被排除还是被包括在变换后的JSON数据240中。在节点级，基于“testexpression”结果，处理器可以决定节点是被排除还是被包括在变换后的JSON数据240中。

[0107] 在某些实施例中，支持各种变换操作。例如，字符串操作可以包括联接、比较、替换、提取子串(substring)等。算术运算可以包括加法、乘法、除法、减法等。布尔运算可以包括true()、false()、not()等。

[0108] 图3是一个实施例中的使用定义一个或多个变换的JSON对象来执行JSON数据变换的方法300的流程图。图3中绘出的方法300中的处理或方法300的实现可以在由逻辑机器(诸如计算机系统或信息处理设备)的中央处理单元(CPU或处理器)执行时由软件(例如，指令或代码模块)执行、可以由电子设备或专用集成电路的硬件组件执行、或者可以由软件和硬件元件的组合执行。图3中绘出的方法300开始于步骤310。

[0109] 在步骤310中，接收第一JSON对象。第一JSON对象可以作为字符串形式的序列化对象被接收。在步骤320中，接收定义一个或多个变换的第二JSON对象。如上面所讨论的，像用户熟悉的其它JSON数据那样，使用指示将应用于其的过滤器、源节点、源对象的变换的一个或多个已定义的关键字将第二JSON对象编写为用户熟悉的其他JSON对象。

[0110] 在步骤330中，基于第一JSON对象和第二JSON对象生成第三JSON对象。相应地，不需要变换成任何中间形式或语言。此外，用户不需要为了定义变换而学习新语言。

[0111] 图4是一个实施例中的用于确定如何使用一个对象的JSON数据属性语言来变换另一个对象的JSON数据的方法400的流程图。图4中绘出的方法400中的处理或方法400的实现可以在由逻辑机器(诸如计算机系统或信息处理设备)的中央处理单元(CPU或处理器)执行时由软件(例如，指令或代码模块)执行、可以由电子设备或专用集成电路的硬件组件执行、或者可以由软件和硬件元件的组合执行。图4中绘出的方法400开始于步骤410。

[0112] 在步骤410中，确定输入对象具有由变换对象的属性引用的属性。在某些实施例中，可以通过在输入对象与变换对象之间匹配属性名称来进行引用。相应地，输入对象的属性可以具有与变换对象的属性完全相同的名称。在一些实施例中，输入对象的属性可以由变换对象的属性的值引用。例如，作为变换对象的属性的值而提供的名称关键字可以被用来对输入对象的属性的名称进行引用。

[0113] 在步骤420中，确定如何基于变换对象的属性来变换输入对象的属性。在某些实施例中，所确定的变换可以包括输入对象的属性的重命名、对输入对象的属性的值的修改、对于是否将输入对象的属性包括在输出对象中的确定，等等。

[0114] 在步骤430中,基于使用变换对象变换输入对象来生成输出对象。

[0115] 图5是一个实施例中的用于确定如何使用JSON样式表语言来变换JSON数据的方法的流程图。图5中绘出的方法500中的处理或方法500的实现可以在由逻辑机器(诸如计算机系统或信息处理设备)的中央处理单元(CPU或处理器)执行时由软件(例如,指令或代码模块)执行、可以由电子设备或专用集成电路的硬件组件执行、或者可以由软件和硬件元件的组合执行。图5中绘出的方法500开始于步骤510。

[0116] 在步骤510中,接收输入属性。输入属性由键-值对表示。键提供属性的名称,并且值可以是一个或多个基本数据类型或对象。在步骤520中,确定是否存在输入属性的对应变换。如上面所讨论的,可以通过变换对象的与输入属性具有相同名称的属性、通过变换对象的具有对输入属性的引用的属性等来找到对应的变换。

[0117] 如果确定不存在输入属性的对应变换,则在步骤530中,确定在输出对象中是包括还是排除该输入属性。规则可以指示变换的不存在使属性按原样流向输出对象。另一个规则可以指示变换的不存在使属性从输出对象中被过滤掉。

[0118] 如果确定存在输入属性的对应变换,则在步骤540中,基于变换对象确定条件集合。在各种实施例中,变换对象的对应属性可以包括定义一个或多个条件的表达式。条件可以表达需要被满足以针对该条件评估为真的准则(criteria)。准则可以在以下方面被表达:属性的名称、属性的值、其组合或者需要被评估以便确定它们是否满足预定准则的其它条件、规则、阈值、限制等。

[0119] 在步骤550中,确定输入属性是否满足所确定的准则。可以发现输入属性满足或不满足准则的空集。可以发现输入属性满足或不满足预定阈值、评估、计算、操作集合等。如果确定输入属性未能满足所确定的准则,则在步骤530中,确定在输出对象中是包括还是排除该输入属性,如上面所讨论的那样。

[0120] 如果确定输入属性满足所确定的准则,则在步骤560中,基于变换对象来确定要执行的操作集合,这些操作确定输出对象中的对应属性的值。在某些实施例中,所确定的操作集合可以包括:输入对象的属性的重命名、对输入对象的属性的值的修改、对于在输出对象中是包括输入对象的属性的值的全部或部分的确定,等等。

[0121] 在步骤570中,针对输出对象中的对应属性基于变换对象确定数据类型。数据类型可以由变换对象显式地定义、从输入对象推断、基于执行操作集合的结果确定,等等。如上面所讨论的,所确定的数据类型可以包括基本JSON对象数据类型。

[0122] 在步骤580中,基于执行操作集合所得到的值以及所确定的数据类型来生成输出对象的对应属性。在某些实施例中,操作的空集可以使输入属性的值流经到输出对象的对应属性。在其它实施例中,操作集合可以重命名值、计算新值、修改值等。

[0123] 图6是一个实施例中的用于变换与JSON样式表语言对象的属性具有相同名称的JSON数据属性的方法600的流程图。图6中绘出的方法600中的处理或方法600的实现可以在由逻辑机器(诸如计算机系统或信息处理设备)的中央处理单元(CPU或处理器)执行时由软件(例如,指令或代码模块)执行、可以由电子设备或专用集成电路的硬件组件执行,或者可以由软件和硬件元件的组合执行。图6中绘出的方法600开始于步骤610。

[0124] 在步骤610中,确定输入对象具有名称与变换对象的属性的名称匹配的属性。图7是一个实施例中的输入JSON字符串710、JSL字符串720以及通过使用JSL字符串720对输入

JSON字符串710的属性进行变换而产生的输出JSON字符串730的图示。如所示的,输入JSON字符串710和JSL字符串720两者都具有名称为“id”的属性。

[0125] 在步骤620中,根据变换对象的具有相同名称的属性确定操作和数据类型。参考图7,JSL字符串720的属性“id”指定如何使用表达式“`root.id*2`”来变换输入JSON字符串710的“id”属性。换句话说,输入JSON字符串710的“id”属性的值将乘以2。JSL字符串720的属性“id”还指定从表达式“`root.id*2`”得到的值将是整数。

[0126] 在步骤630中,对输入对象的属性的值执行所确定的由变换对象的匹配属性指定的操作。在步骤640中,基于所确定的数据类型和操作的结果为输出对象创建具有相同名称的属性。如图7中所示,输出JSON字符串730包括具有使用表达式“`root.id*2`”确定的值的“id”属性。输出JSON字符串730的“id”属性的数据类型是整数。

[0127] 图8是一个实施例中的用于变换JSON样式表语言对象的属性所引用的JSON数据属性的方法800的流程图。图8中绘出的方法800中的处理或方法800的实现可以在由逻辑机器(诸如计算机系统或信息处理设备)的中央处理单元(CPU或处理器)执行时由软件(例如,指令或代码模块)执行、可以由电子设备或专用集成电路的硬件组件执行,或者可以由软件和硬件元件的组合执行。图8中绘出的方法800开始于步骤810。

[0128] 在步骤810中,确定输入对象具有名称由变换对象的属性引用的属性。如图7中所示,输入JSON字符串710具有名称为“name”的属性,并且JSL字符串720具有名称为“new\_name”的属性,该名称为“new\_name”的属性通过作为值的表达式(例如“`value`”:“`root.name`”)来引用输入JSON字符串710的名称为“name”的属性。

[0129] 在步骤820中,根据变换对象的属性确定属性名称。在步骤830中,根据变换对象的属性确定数据类型。在步骤840中,为输出对象创建具有所确定的名称和所确定的数据类型的属性。如图7中所示,输出JSON字符串730包括具有与输入JSON字符串710的“name”属性相同值(例如,“A green door”)和数据类型(例如,字符串)的“new\_name”属性。

[0130] 图9是一个实施例中的输入JSON字符串910、JSL字符串920以及通过使用JSL字符串920对输入JSON字符串910的属性值进行变换而产生的输出字符串930的图示。在这个示例中,输入JSON字符串910和JSL字符串920具有匹配的属性“tags”。JSL字符串920指定变换,该变换不是针对如上所示的属性名称,而是针对对应属性的值。已经指定了字符串操作“`concat(root.tags, '_new')`”,以向“tags”数组中的每个元素添加后缀“\_new”。相应地,输出JSON串930包括名称为“tags”的属性,其中数组的每个元素都添加了“\_new”后缀。

[0131] 图10是一个实施例中的用于变换满足JSON样式表语言对象的属性的表达式的JSON数据属性的方法1000的流程图。图10中绘出的方法1000中的处理或方法1000的实现可以在由逻辑机器(诸如计算机系统或信息处理设备)的中央处理单元(CPU或处理器)执行时由软件(例如,指令或代码模块)执行、可以由电子设备或专用集成电路的硬件组件执行,或者可以由软件和硬件元件的组合执行。图10中绘出的方法1000开始于步骤1010。

[0132] 在步骤1010中,确定输入对象具有满足变换对象的属性的表达式的属性或属性值。可以在变换对象的具有相同名称的属性的值中、引用输入属性的名称的属性的值中、或者引用输入属性的值的属性的值中指定表达式。图11是一个实施例中的输入JSON字符串1110、JSL字符串1120以及在满足表达式时通过使用JSL字符串1120对输入JSON字符串1110进行变换而产生的输出字符串1130的图示。如所示的,输入JSON字符串1110具有名称为

“emp”的属性,并且JSL字符串1120具有名称为“emp”的属性。JSL字符串1120的名称为“emp”的属性在“textexpression”中包括表达式,该表达式被这样的对象满足:该对象的empname(雇员姓名)以“B”开头。因此,“emp”数组中只有一个对象满足条件(例如,“empname”：“Bob”)。

[0133] 在步骤1020中,根据变换对象的属性确定操作。参考图11,JSL字符串1120的属性“emp”指定如何变换输入JSON字符串1110的“emp”属性的满足“textexpression”的每个对象。在步骤1030中,根据变换对象的属性确定数据类型。在步骤1040中,基于所确定的数据类型和操作的结果为输出对象创建属性。如图11中所示,输出JSON字符串1130包括具有带“empname”：“Bob”的单个对象的“emp”属性。相应地,可以指定过滤掉属性或属性值从而仅在输出对象中留下期望的属性或属性值的条件。

[0134] 图12是一个实施例中的输入JSON字符串1210、JSL字符串1220以及当满足表达式时通过使用JSL字符串1220对输入JSON字符串1210进行变换而产生的输出字符串1230的图示。在这个示例中,过滤掉不期望的属性或属性值。相应地,“textexpression”从对象中移除“Discount”:20属性,因为CustomerType不是“Gold”,而是“Silver”。

[0135] 图13是一个实施例中的被配置为使用JSON样式表语言对象执行JSON对象变换的硬件单元1300的框图。硬件单元1300包括JSON数据接收单元1310、预处理器单元1320、JSON变换接收单元1330、变换单元1340、后处理单元1350和JSON数据生成单元1360。每个单元可以由计算机处理器根据存储在存储器中的计算机可执行代码执行。

[0136] JSON数据接收单元1310包括接收JSON数据的硬件和/或软件元件。JSON数据接收单元1310可以从用户输入或其它计算机进程接收JSON数据。预处理器单元1320包括预处理JSON数据的硬件和/或软件元件。一个或多个预处理步骤可以包括数据操纵操作、数据过滤操作、数据扩充(augmentation)操作等。JSON变换接收单元1330包括接收JSON数据的硬件和/或软件元件。JSON变换接收单元1330可以从用户输入或其它计算机进程接收JSON数据。在这个示例中,在JSON变换接收单元1330处接收的JSON数据根据上面讨论的JSL规范被格式化。

[0137] 变换单元1340包括接收对象JSON对象和JSL变换的形式的JSON数据并应用任何指定的变换的硬件和/或软件元件。变换单元1340不需要任何中间语言(诸如XML)来实现变换。

[0138] 后处理单元1350包括处理JSON数据的硬件和/或软件元件。一个或多个处理步骤可以包括数据操纵操作、数据过滤操作、数据扩充操作等。JSON数据生成单元1360包括生成JSON数据的硬件和/或软件元件。虽然已经示出了多个单元,但是将理解的是,不是全部这些单元都是需要的,并且由一些单元执行的一些功能可以结合到其它单元中。

#### [0139] 结论

[0140] 图14绘出了用于实现实施例之一的分布式系统1400的简化图。在所示实施例中,分布式系统1400包括一个或多个客户端计算设备1402、1404、1406和1408,这些客户端计算设备被配置为经一个或多个网络1410执行并操作诸如web浏览器、专属客户端(例如,Oracle Forms)等的客户端应用。服务器1412可以经由网络1410与远程客户端计算设备1402、1404、1406和1408在通信上耦合。

[0141] 在各种实施例中,服务器1412可以适于运行由系统的组件其中的一个或多个提供

的一个或多个服务或软件应用。在一些实施例中,这些服务可以作为基于web的服务或云服务、或者在软件即服务(SaaS)模型或平台即服务(PaaS)模型下向客户计算设备1402、1404、1406和/或1408的用户提供。操作客户端计算设备1402、1404、1406和/或1408的用户继而可以利用一个或多个客户端应用与服务器1412交互,以利用由这些组件提供的服务。

[0142] 在图14中所绘出的配置中,系统1400的软件组件1418、1420和1422被示为在服务器1412上实现。在其它实施例中,系统1400的一个或多个组件和/或由这些组件提供的服务也可以由客户端计算设备1402、1404、1406和/或1408当中的一个或多个实现。操作客户端计算设备的用户然后可以利用一个或多个客户端应用来使用由这些组件提供的服务。这些组件可以在硬件、固件、软件或其组合中实现。应当认识到的是,各种不同的系统配置是可能的,这些配置可以与分布式系统1400不同。因此,该图中所示的实施例是用于实现实施例系统的分布式系统的一个例子并且不旨在进行限制。

[0143] 客户端计算设备1402、1404、1406和/或1408可以是便携式手持设备(例如,**iPhone®**、蜂窝电话、**iPad®**、计算平板、个人数字助理(PDA))或可穿戴设备(例如,Google**Glass®**头戴式显示器),这些设备运行诸如Microsoft Windows**Mobile®**的软件和/或诸如iOS、Windows Phone、Android、BlackBerry 10、Palm OS等各种移动操作系统,并且启用了互联网、电子邮件、短消息服务(SMS)、**BlackBerry®**或其它的通信协议。客户端计算设备1402、1404、1406和/或1408可以是通用个人计算机,包括例如运行各种版本的Microsoft**Windows®**、Apple**Macintosh®**和/或Linux操作系统的个人计算机和/或膝上型计算机。客户端计算设备1402、1404、1406和/或1408可以是运行任何各种商业可用的**UNIX®**或类UNIX操作系统(包括但不限于各种GNU/Linux操作系统,诸如例如Google Chrome OS)中的任意操作系统的工作站计算机。替代地或附加地,客户端计算设备1402、1404、1406和1408可以是能够经(一个或多个)网络1410通信的任何其它电子设备,诸如瘦客户端计算机、启用互联网的游戏系统(例如,具有或不具有**Kinect®**手势输入设备的Microsoft Xbox游戏控制台),和/或个人消息传送设备。

[0144] 虽然示例性分布式系统1400被示为具有四个客户端计算设备,但任何数量的客户端计算设备可以被支持。其它设备(诸如具有传感器的设备等)可以与服务器1412交互。

[0145] 分布式系统1400中的(一个或多个)网络1410可以是本领域技术人员熟悉的、可以支持使用各种商业可用的协议当中的任意协议的数据通信的任何类型的网络,其中这些协议包括但不限于TCP/IP(传输控制协议/互联网协议)、SNA(系统网络体系架构)、IPX(互联网数据包交换)、AppleTalk,等等。仅仅作为示例,(一个或多个)网络1410可以是局域网(LAN),诸如基于以太网、令牌环等的LAN。(一个或多个)网络1410可以是广域网和互联网。它可以包括虚拟网络,包括但不限于虚拟专用网(VPN)、内联网、外联网、公共交换电话网(PSTN)、红外网络、无线网络(例如,在电子电气工程师协会(IEEE)802.11协议套件、**Bluetooth®**和/或任何其它无线协议当中的任意协议下操作的网络)和/或这些和/或其它网络的任意组合。

[0146] 服务器1412可以由一个或多个通用计算机、专用服务器计算机(包括例如PC(个人计算机)服务器、**UNIX®**服务器、中档服务器、大型计算机、机架式服务器,等等)、服务器场、服务器集群,或者任何其它适当的布置和/或组合组成。在各种实施例中,服务器1412可



以适于运行在前面公开中所描述的一个或多个服务或软件应用。例如,服务器1412可以对应于用于执行上面根据本公开的实施例所描述的处理的服务器。

[0147] 服务器1412可以运行包括上面所讨论的操作系统当中的任意操作系统以及任何商业可用的服务器操作系统的操作系统。服务器1412还可以运行各种附加的服务器应用和/或中间层应用当中的任意应用,包括HTTP(超文本传输协议)服务器、FTP(文件传输协议)服务器、CGI(公共网关接口)服务器、**JAVA®**服务器、数据库服务器,等等。示例性数据库服务器包括但不限于从Oracle、Microsoft、Sybase、IBM(国际商业机器)等等可商业获得的那些。

[0148] 在一些实现中,服务器1412可以包括一个或多个应用,以分析和整合从客户端计算设备1402、1404、1406和1408的用户接收的数据馈送和/或事件更新。作为示例,数据馈送和/或事件更新可以包括但不限于**Twitter®**馈送、**Facebook®**更新或者从一个或多个第三方信息源接收到的实时更新以及连续数据流,这些连续数据流可以包括与传感器数据应用、金融报价机、网络性能测量工具(例如,网络监视和流量管理应用)、点击流分析工具、汽车流量监视等相关的实时事件。服务器1412还可以包括一个或多个应用,以经由客户端计算设备1402、1404、1406和1408的一个或多个显示设备显示数据馈送和/或实时事件。

[0149] 分布式系统1400还可以包括一个或多个数据库1414和1416。数据库1414和1416可以驻留在各个位置中。作为示例,数据库1414和1416当中的一个或多个可以驻留在服务器1412本地(和/或驻留在服务器1412中)的非临时性存储介质上。作为替代,数据库1414和1416可以远离服务器1412,并且经基于网络的连接或专用的连接与服务器1412通信。在一组实施例中,数据库1414和1416可以驻留在本领域技术人员熟悉的存储区域网络(SAN)中。类似地,用于执行服务器1412所具有的功能的任何必要的文件都可以适当地本地存储在服务器1412上和/或远程存储。在一组实施例中,数据库1414和1416可以包括适于响应于SQL格式的命令而存储、更新和检索数据的关系数据库,诸如由Oracle提供的数据库。

[0150] 图15示出了在其中可以实现本发明的各种实施例的示例性计算机系统1500。系统1500可以被用来实现上述任何计算机系统。如图15中所示,计算机系统1500包括总线子系统1502和经由总线子系统1502与多个外围子系统通信的处理单元1504。这些外围子系统可以包括处理加速单元1506、I/O子系统1508、存储子系统1518和通信子系统1524。存储子系统1518包括有形计算机可读存储介质1522和系统存储器1510。

[0151] 总线子系统1502提供了用于让计算机系统1500的各个组件和子系统按意图彼此通信的机制。虽然总线子系统1502被示意性地示为单个总线,但是总线子系统的替代实施例可以利用多个总线。总线子系统1502可以是若干种类型的总线结构中的任何一种,包括使用任何各种总线体系结构的存储器总线或存储器控制器、外围总线、以及局部总线。例如,这种体系结构可以包括工业标准体系结构(ISA)总线、微通道体系结构(MCA)总线、增强型ISA(EISA)总线、视频电子标准协会(VESA)局部总线和外围组件互连(PCI)总线,这些可以被实现为按IEEE P1386.1标准制造的Mezzanine(夹层)总线。

[0152] 可以被实现为一个或多个集成电路(例如,常规微处理器或微控制器)的处理单元1504控制计算机系统1500的操作。一个或多个处理器可以被包括在处理单元1504中。这些处理器可以包括单核处理器或多核处理器。在某些实施例中,处理单元1504可以被实现为一个或多个独立的处理单元1532和/或1534,其中在每个处理单元中包括单个或多核处理



器。在其它实施例中,处理单元1504也可以被实现为通过将两个双核处理器集成到单个芯片中而形成的四核处理单元。

[0153] 在各种实施例中,处理单元1504可以响应于程序代码执行各种程序并且可以维护多个并发执行的程序或进程。在任何给定的时间,要被执行的程序代码中的一些或全部代码可以驻留在(一个或多个)处理器1504中和/或存储子系统1518中。通过适当的编程,(一个或多个)处理器1504可以提供上述各种功能。计算机系统1500可以附加地包括处理加速单元1506,处理加速单元1506可以包括数字信号处理器(DSP)、专用处理器,等等。

[0154] I/O子系统1508可以包括用户接口输入设备和用户接口输出设备。用户接口输入设备可以包括键盘、诸如鼠标或轨迹球的指点设备、结合到显示器中的触摸板或触摸屏、滚动轮、点击轮、拨盘、按钮、开关、键盘、具有语音命令识别系统的音频输入设备、麦克风以及其它类型的输入设备。用户接口输入设备可以包括例如诸如Microsoft **Kinect**<sup>®</sup>运动传感器之类的运动感测和/或手势识别设备,Microsoft **Kinect**<sup>®</sup>运动传感器使得用户能够使用手势和语音命令通过自然用户接口控制诸如Microsoft **Xbox**<sup>®</sup> 360游戏控制器的输入设备并与之交互。用户接口输入设备还可以包括眼睛姿势识别设备,诸如从用户检测眼睛活动(例如,当拍摄照片和/或做出菜单选择时的“眨眼”)并且将眼睛姿势变换为到输入设备(例如,Google **Glass**<sup>®</sup>)中的输入的Google **Glass**<sup>®</sup>眨眼检测器。此外,用户接口输入设备可以包括使用户能够通过语音命令与语音识别系统(例如,**Siri**<sup>®</sup>导航器)交互的语音识别感测设备。

[0155] 用户接口输入设备还可以包括但不限于三维(3D)鼠标、操纵杆或指向棒、游戏面板和绘图板,以及音频/视频设备,诸如扬声器、数码相机、数码摄像机、便携式媒体播放器、网络摄像头、图像扫描仪、指纹扫描仪、条形码阅读器3D扫描仪、3D打印机、激光测距仪和视线跟踪设备。此外,用户接口输入设备可以包括例如医学成像输入设备,诸如计算机断层扫描、磁共振成像、正电子发射断层扫描、医疗超声设备。用户接口输入设备还可以包括例如音频输入设备,诸如MIDI键盘、数字乐器等。

[0156] 用户接口输出设备可以包括显示子系统、指示灯,或者诸如音频输出设备的非视觉显示器,等等。显示子系统可以是阴极射线管(CRT)、诸如使用液晶显示器(LCD)或等离子显示器的平板设备、投影设备、触摸屏,等等。一般而言,术语“输出设备”的使用旨在包括用于从计算机系统1500向用户或其它计算机输出信息的所有可能类型的设备和机制。例如,用户接口输出设备可以包括但不限于在视觉上传达文本、图形和音频/视频信息的各种显示设备,诸如监视器、打印机、扬声器、耳机、汽车导航系统、绘图仪、语音输出设备,以及调制解调器。

[0157] 计算机系统1500可以包括包含软件元素的存储子系统1518,其中这些软件元素被示为当前位于系统存储器1510中。系统存储器1510可以存储可加载并且可在处理单元1504上执行的程序指令,以及这些程序执行期间所产生的数据。

[0158] 取决于计算机系统1500的配置和类型,系统存储器1510可以是易失性的(诸如随机存取存储器(RAM))和/或非易失性的(诸如只读存储器(ROM)、闪存存储器,等等)。RAM通常包含可以被处理单元1504立即访问和/或目前正被处理单元1504操作和执行的数据和/或程序模块。在一些实现中,系统存储器1510可以包括多种不同类型的存储器,例如静态随

机存取存储器 (SRAM) 或动态随机存取存储器 (DRAM)。在一些实现中,基本输入/输出系统 (BIOS) 通常可以被存储在ROM中,基本输入/输出系统 (BIOS) 包含诸如在启动期间有助于在计算机系统1500内的元件之间传送信息的基本例程。作为示例而不是限制,系统存储器1510也示出了应用程序1512、程序数据1514、以及操作系统1516,其中应用程序1512可以包括客户端应用、web浏览器、中间层应用、关系数据库管理系统 (RDBMS) 等。作为示例,操作系统1516可以包括各种版本的Microsoft **Windows®**, Apple **Macintosh®**和/或Linux操作系统、各种商业可用的**UNIX®**或类UNIX操作系统 (包括但不限于各种GNU/Linux操作系统、Google **Chrome®**操作系统,等等) 和/或诸如iOS、**Windows®** Phone、**Android®** OS、**BlackBerry®** 100S和**Palm®** OS操作系统之类的移动操作系统。

[0159] 存储子系统1518也可以提供用于存储提供一些实施例的功能的基本编程和数据构造的有形计算机可读存储介质。当被处理器执行时提供上述功能的软件 (程序、代码模块、指令) 可以被存储在存储子系统1518中。这些软件模块或指令可以由处理单元1504执行。存储子系统1518还可以提供用于存储根据本发明被使用的数据的储存库。

[0160] 存储子系统1500还可以包括可以被进一步连接到计算机可读存储介质1522的计算机可读存储介质读取器1520。计算机可读存储介质1522可以与系统存储器1510一起以及可选地与系统存储器1510组合来全面地表示用于临时和/或更持久地包含、存储、传输和检索计算机可读信息的远程、本地、固定和/或可移除的存储设备加存储介质。

[0161] 包含代码或代码的部分的计算机可读存储介质1522还可以包括本领域已知或使用的任何适当的介质,包括存储介质和通信介质,诸如但不限于以用于信息的存储和/或传输的任何方法或技术实现的易失性和非易失性、可移除和不可移除介质。这可以包括有形的计算机可读存储介质,诸如RAM、ROM、电可擦除可编程ROM (EEPROM)、闪存存储器或其它存储器技术、CD-ROM、数字多功能盘 (DVD) 或其它光学存储器、磁卡带、磁带、磁盘存储器或其它磁存储设备,或者其它有形的计算机可读介质。这也可以包括非有形的计算机可读介质,诸如数据信号、数据传输,或者可以被用来传输期望的信息并且可以被计算系统1500访问的任何其它介质。

[0162] 作为示例,计算机可读存储介质1522可以包括从不可移除的非易失性磁介质读取或写到其的硬盘驱动器、从可移除的非易失性磁盘读取或写到其的磁盘驱动器、以及从可移除的非易失性光盘 (诸如CD ROM、DVD和Blu-**Ray®**盘或其它光学介质) 读取或写到其的光盘驱动器。计算机可读存储介质1522可以包括但不限于**Zip®**驱动器、闪存存储器卡、通用串行总线 (USB) 闪存驱动器、安全数字 (SD) 卡、DVD盘、数字音频带,等等。计算机可读存储介质1522还可以包括基于非易失性存储器的固态驱动器 (SSD) (诸如基于闪存存储器的SSD、企业闪存驱动器、固态ROM等)、基于易失性存储器的SSD (诸如固态RAM、动态RAM、静态RAM、基于DRAM的SSD、磁阻RAM (MRAM) SSD)、以及使用基于DRAM的SSD和基于闪存存储器的SSD的混合SSD。盘驱动器及其关联的计算机可读介质可以为计算机系统1500提供计算机可读指令、数据结构、程序模块及其它数据的非易失性存储。

[0163] 通信子系统1524提供到其它计算机系统和网络的接口。通信子系统1524充当用于从其它系统接收数据和从计算机系统1500向其它系统传输数据的接口。例如,通信子系统1524可以使计算机系统1500能够经由互联网连接到一个或多个设备。在一些实施例中,通

信子系统1524可以包括用于访问无线语音和/或数据网络的射频(RF)收发器组件(例如,使用蜂窝电话技术,诸如3G、4G或EDGE(用于全球演进的增强型数据速率)的先进数据网络技术,WiFi(IEEE 802.11系列标准)、或其它移动通信技术、或其任意组合)、全球定位系统(GPS)接收器组件和/或其它组件。在一些实施例中,作为无线接口的附加或者替代,通信子系统1524还可以提供有线网络连接(例如,以太网)。

[0164] 在一些实施例中,通信子系统1524也可以代表可以使用计算机系统1500的一个或多个用户接收结构化和/或非结构化数据馈送1526、事件流1528、事件更新1530等形式的输入通信。

[0165] 作为示例,通信子系统1524可以被配置为实时地从社交媒体网络和/或其它通信服务的用户接收数据馈送1526,诸如**Twitter®**馈送、**Facebook®**更新、诸如丰富站点摘要(RSS)馈送的web馈送,和/或,接收来自一个或多个第三方信息源的实时更新。

[0166] 此外,通信子系统1524还可以被配置为接收连续数据流形式的数据,连续数据流可以包括实时事件的事件流1528和/或事件更新1530,连续数据流形式的数据本质上可以是连续的或无界的而没有明确的终止。产生连续数据的应用的示例可以包括例如传感器数据应用、金融报价机、网络性能测量工具(例如,网络监视和流量管理应用)、点击流分析工具、汽车流量监视,等等。

[0167] 通信子系统1524还可以被配置为向一个或多个数据库输出结构化和/或非结构化数据馈送1526、事件流1528、事件更新1530等等,这一个或多个数据库可以与耦合到计算机系统1500的一个或多个流传输数据源计算机通信。

[0168] 计算机系统1500可以是各种类型之一,包括手持便携式设备(例如,**iPhone®**蜂窝电话、**iPad®**计算平板、PDA)、可穿戴设备(例如,Google**Glass®**头戴式显示器)、PC、工作站、大型机、信息站、服务器机架、或任何其它数据处理系统。

[0169] 由于计算机和网络的不断变化的本质,在图中绘出的计算机系统1500的描述旨在仅仅作为具体的示例。具有比图中绘出的系统更多或更少组件的许多其它配置是可能的。例如,也可以使用定制的硬件,和/或特定的元件可以用硬件、固件、软件(包括applets)或它们的组合来实现。另外,也可以采用到诸如网络输入/输出设备之类的其它计算设备的连接。基于本文提供的公开和教导,本领域普通技术人员将认识到实现各种实施例的其它方式和/或方法。

[0170] 在前述的说明书中,本发明的各方面参考其具体实施例进行了描述,但本领域技术人员将认识到,本发明不限于此。上述发明的各个特征和各方面可以被单独或联合使用。另外,在不背离本说明书的更广泛精神和范围的情况下,实施例可以在除本文所述的那些之外的任何数目的环境 and 应用中被利用。相应地,本说明书和附图应当被认为是说明性而不是限制性的。

[0171] 其教导可以在本公开内呈现的一个或多个发明中任何一个的各种实施例可以以软件、固件、硬件或其组合中的逻辑的形式实现。逻辑可以存储在机器可访问存储器、机器可读物品、有形计算机可读介质、计算机可读存储介质或其它计算机/机器可读介质之中或之上,作为适于指导逻辑机器的中央处理单元(CPU或处理器)执行可在本公开内呈现的发明的各种实施例中公开的一组步骤的指令集合。逻辑可以形成软件程序或计算机程序产品的一部分,因为代码模块在被执行时变得与计算机系统或信息处理设备的处理器一起操

作,以执行本公开内呈现的发明的各种实施例中的方法或过程。基于本公开和本文提供的教导,本领域普通技术人员将认识到用于在软件、固件、硬件或其组合中实现所呈现的一个或多个发明的各种实施例的所公开的任何操作或功能的其它方式、变型、修改、替代和/或方法。

[0172] 其教导可以在本公开中呈现的那些发明中任何一个的所公开的示例、实现和各种实施例仅仅是说明性的,以便以合理的清晰度向本领域技术人员传达本公开的教导。由于这些实现和实施例可以参考示例性说明或具体附图来描述,因此对于所描述的方法和/或具体结构的各种修改或调整可以对本领域技术人员变得清楚。依赖于本公开和本文中找到的一些教导并且所述教导已通过其使本领域进步的所有此类修改、调整或变化都将被认为在其教导可以在本公开中呈现的一个或多个发明的范围内。由此,本描述和附图不应当在限制的意义上考虑,因为应当理解,在公开中呈现的发明决不限于具体示出的那些实施例。

[0173] 相应地,以上描述和任何附图、说明和图示旨在是说明性的而不是限制性的。因此,本公开中所呈现的任何发明的范围不应当简单地参考以上描述和图中所示的那些实施例来确定,而是应当参考所附权利要求及其全部范围或等同物来确定。

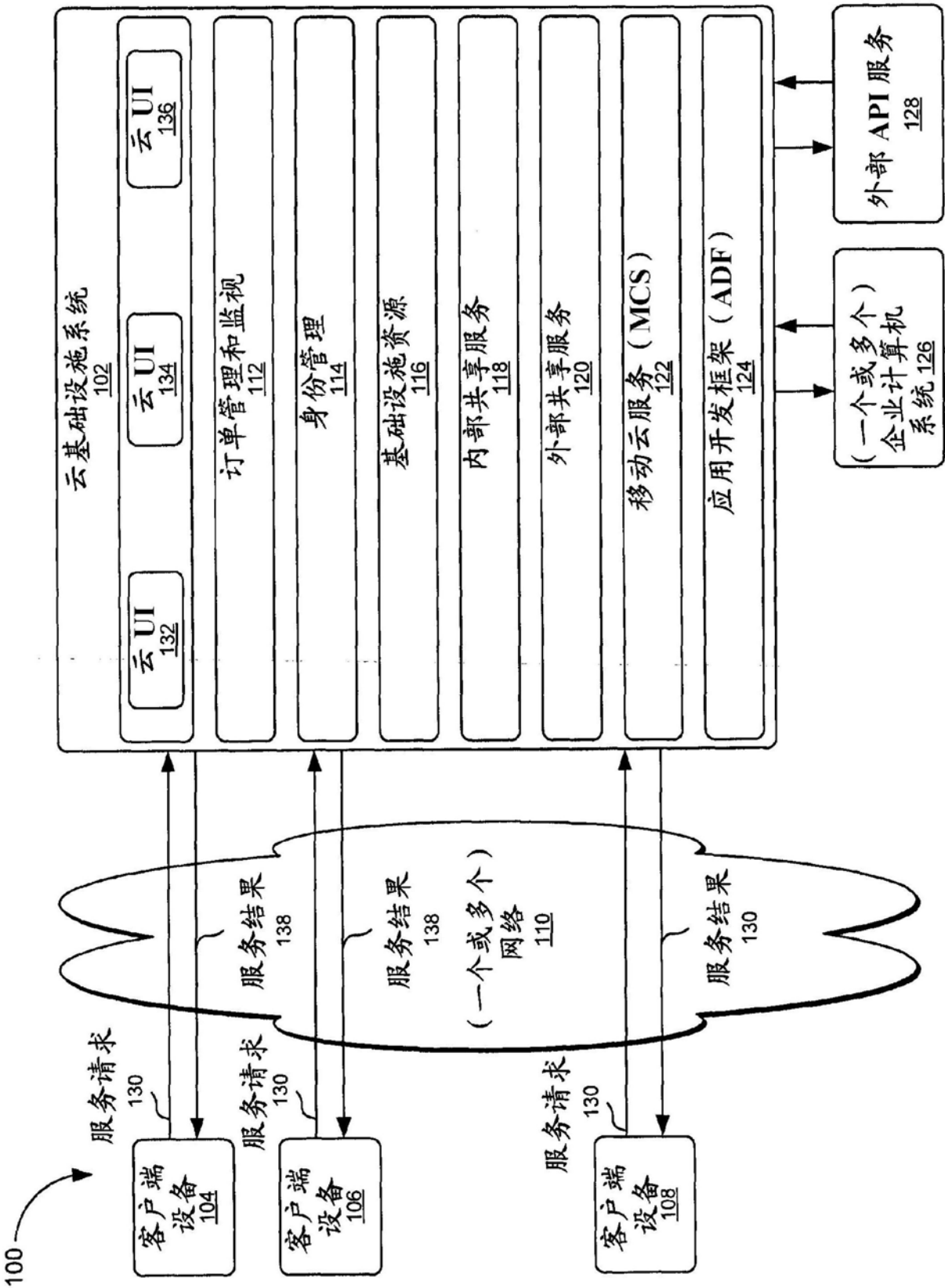


图1

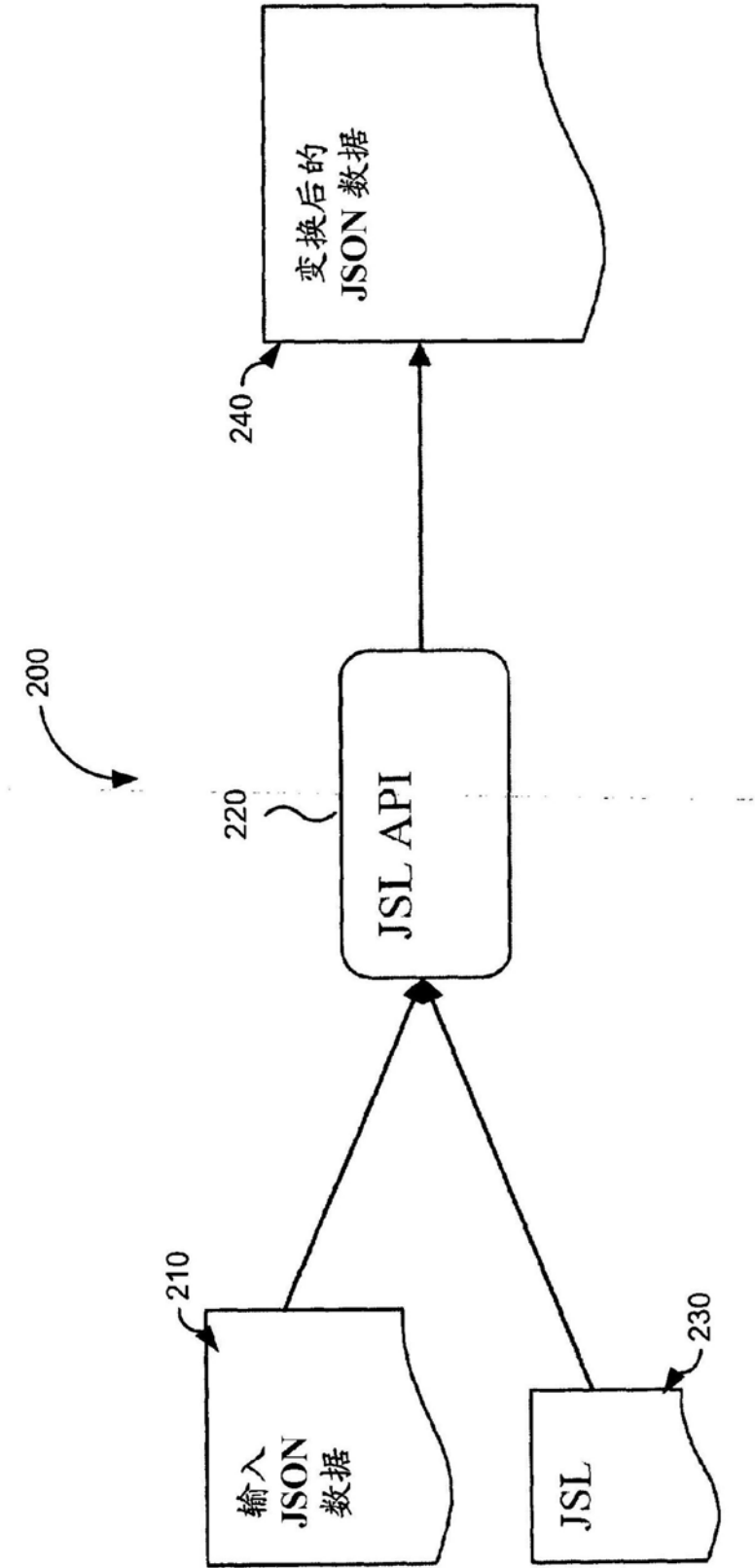


图2

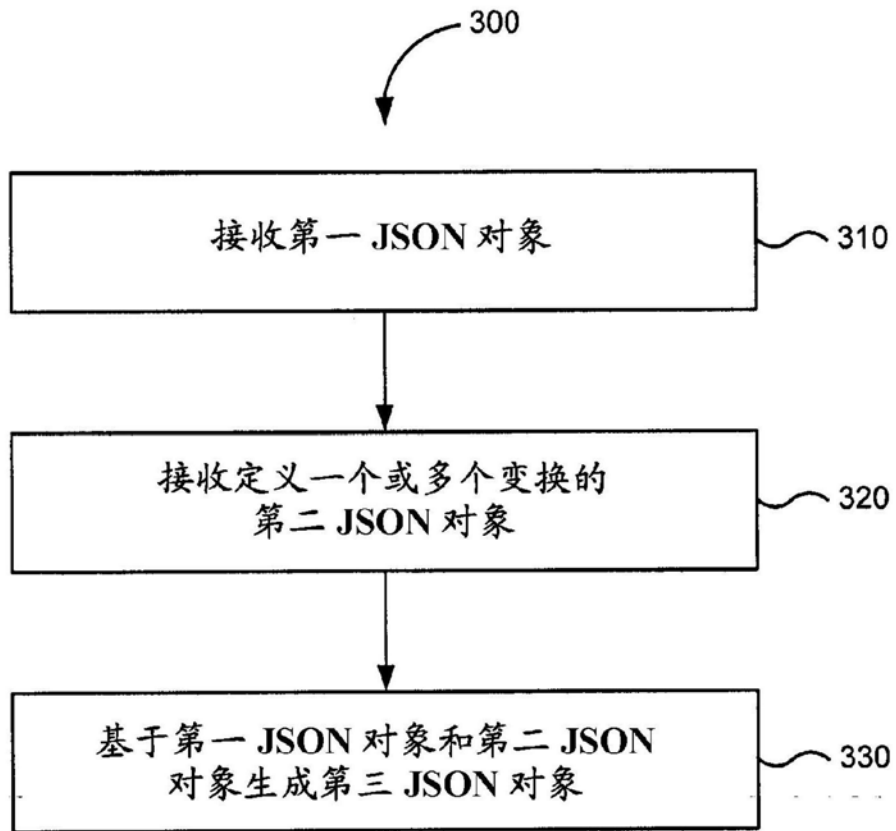


图3

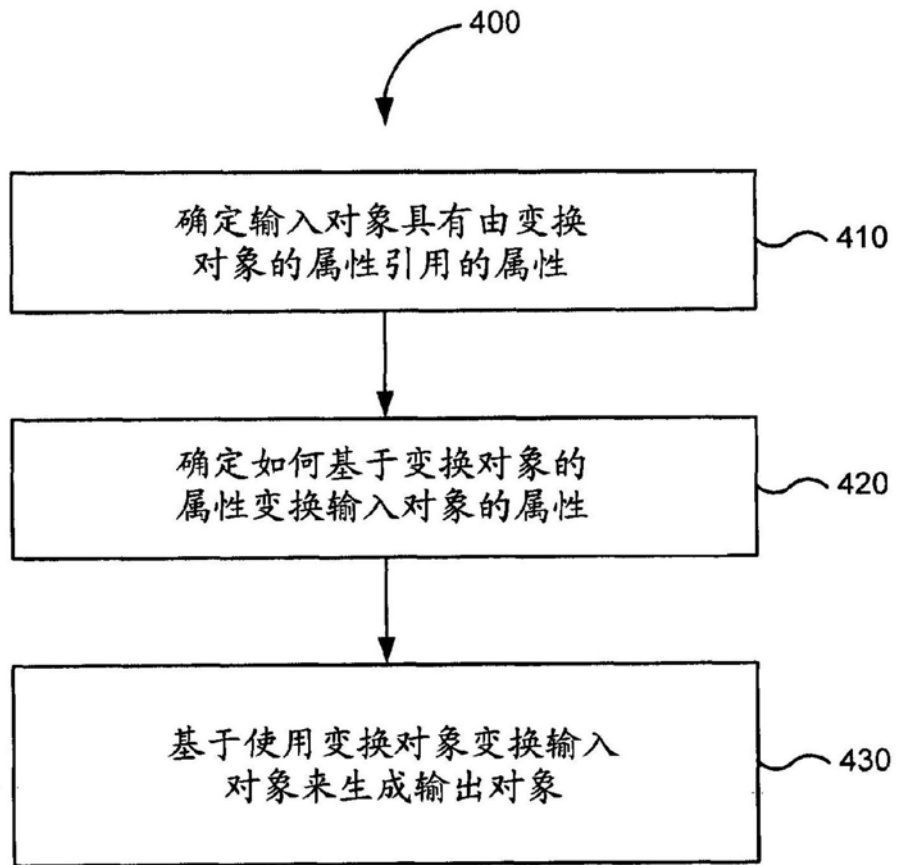


图4



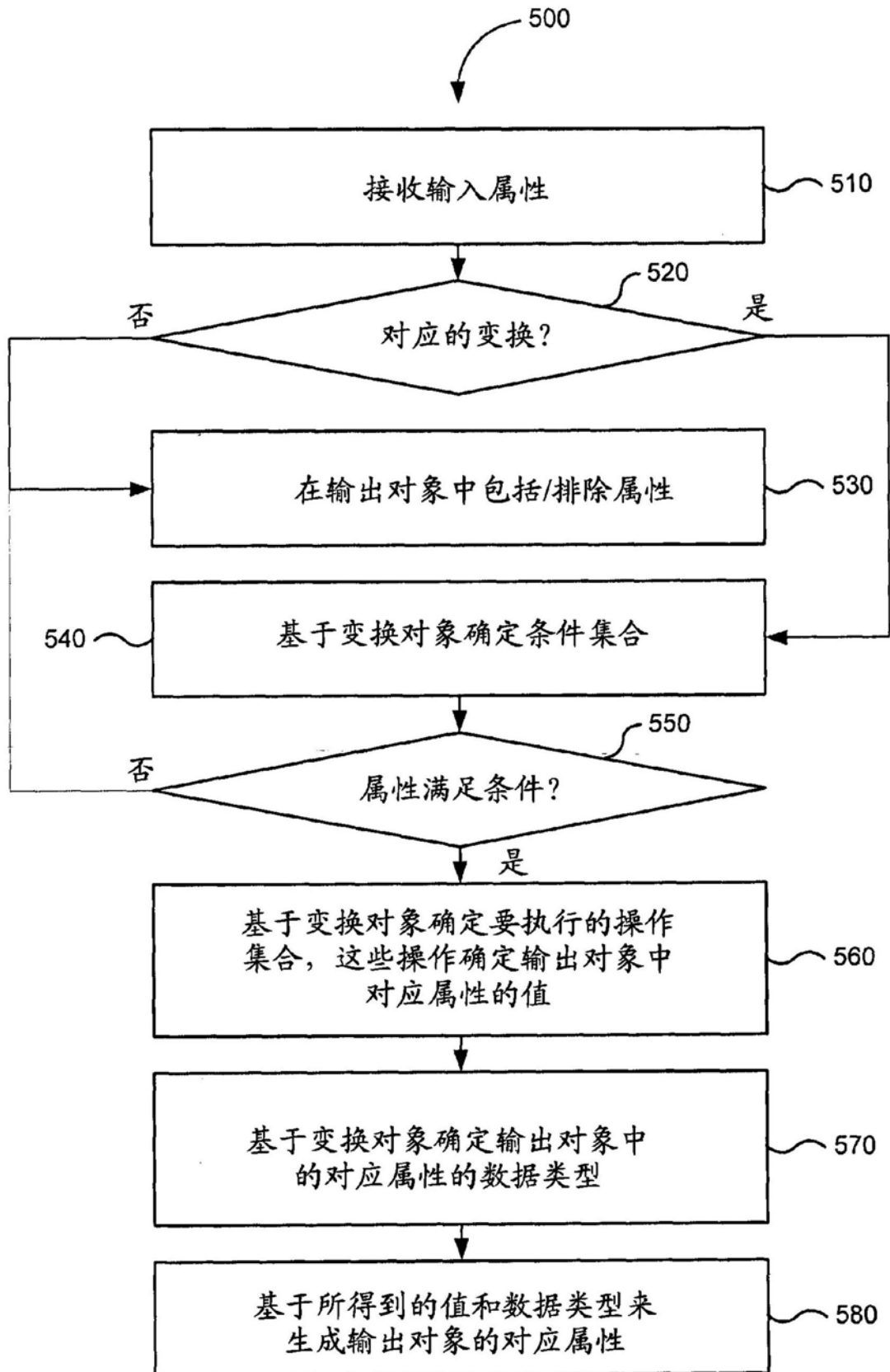


图5

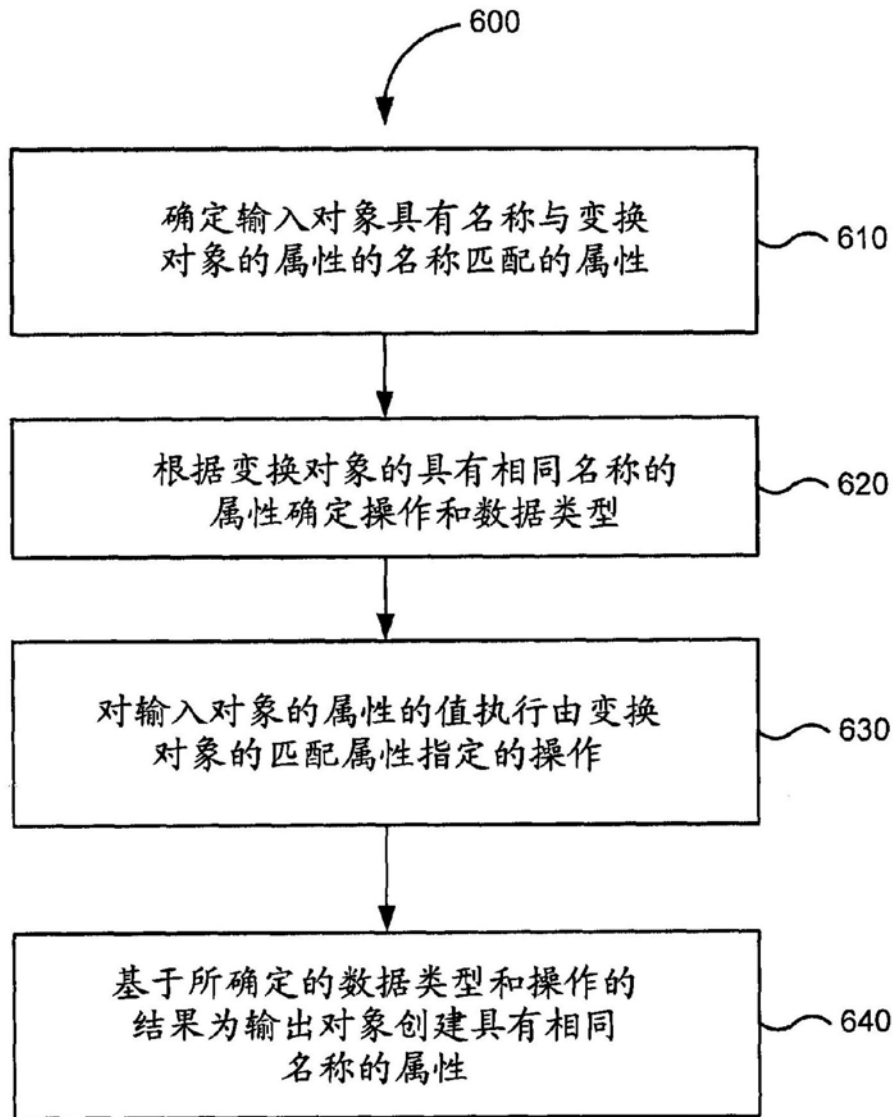


图6

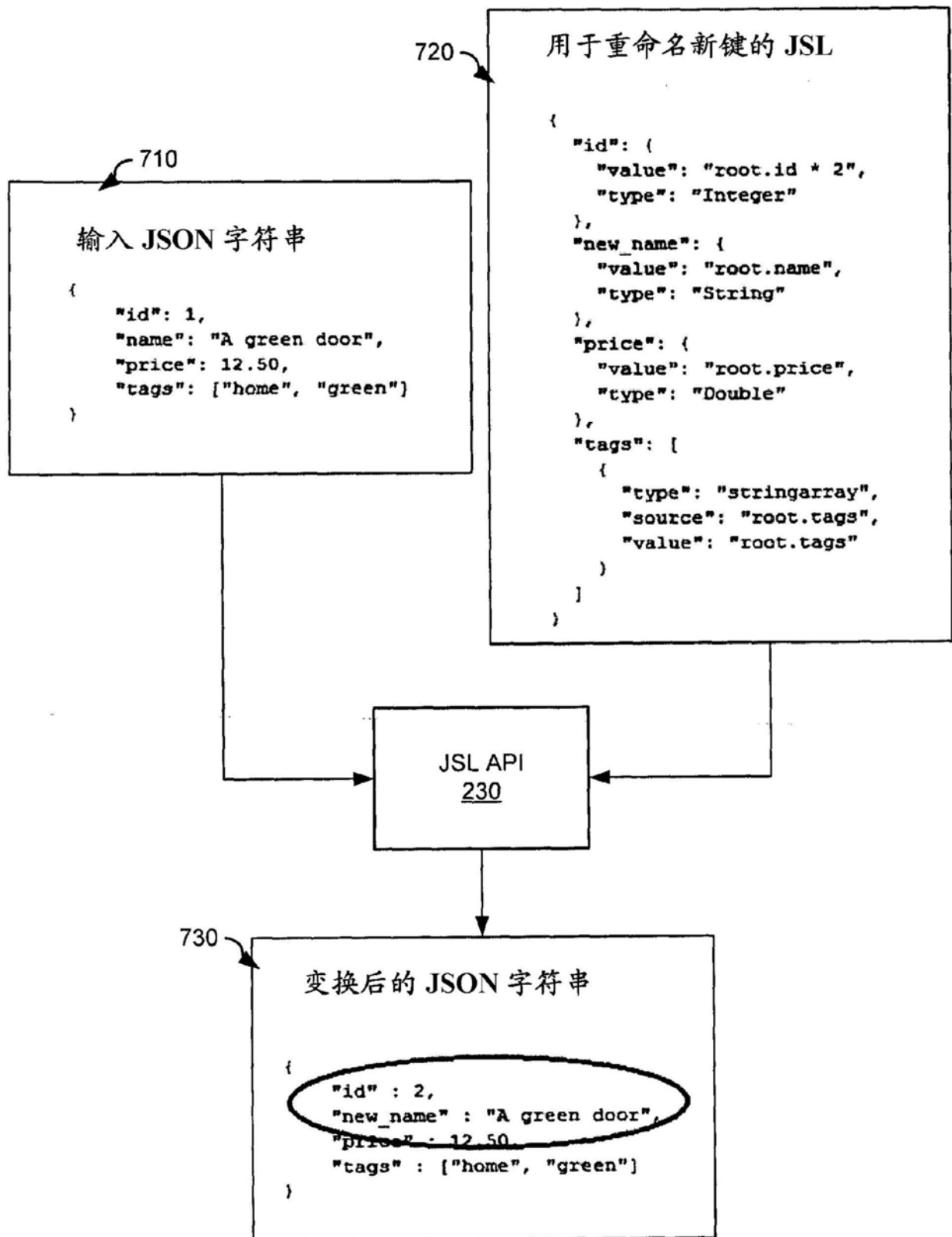


图7

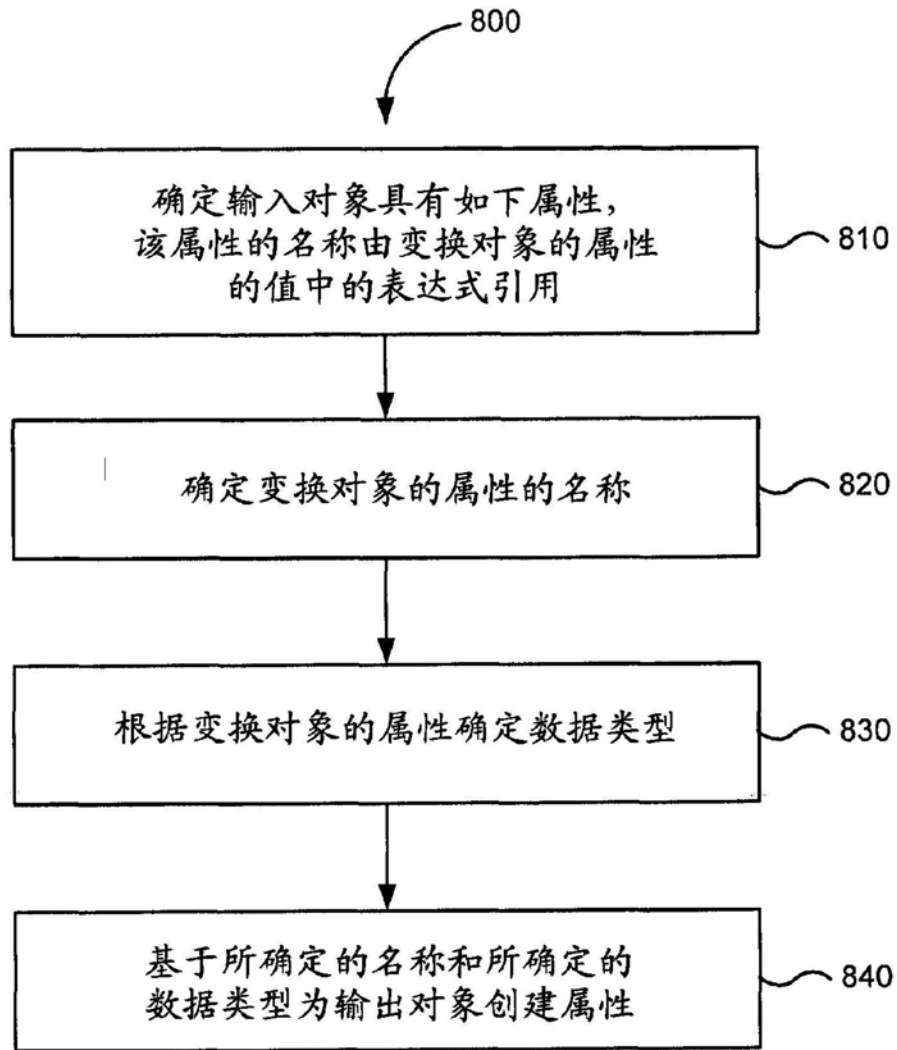


图8

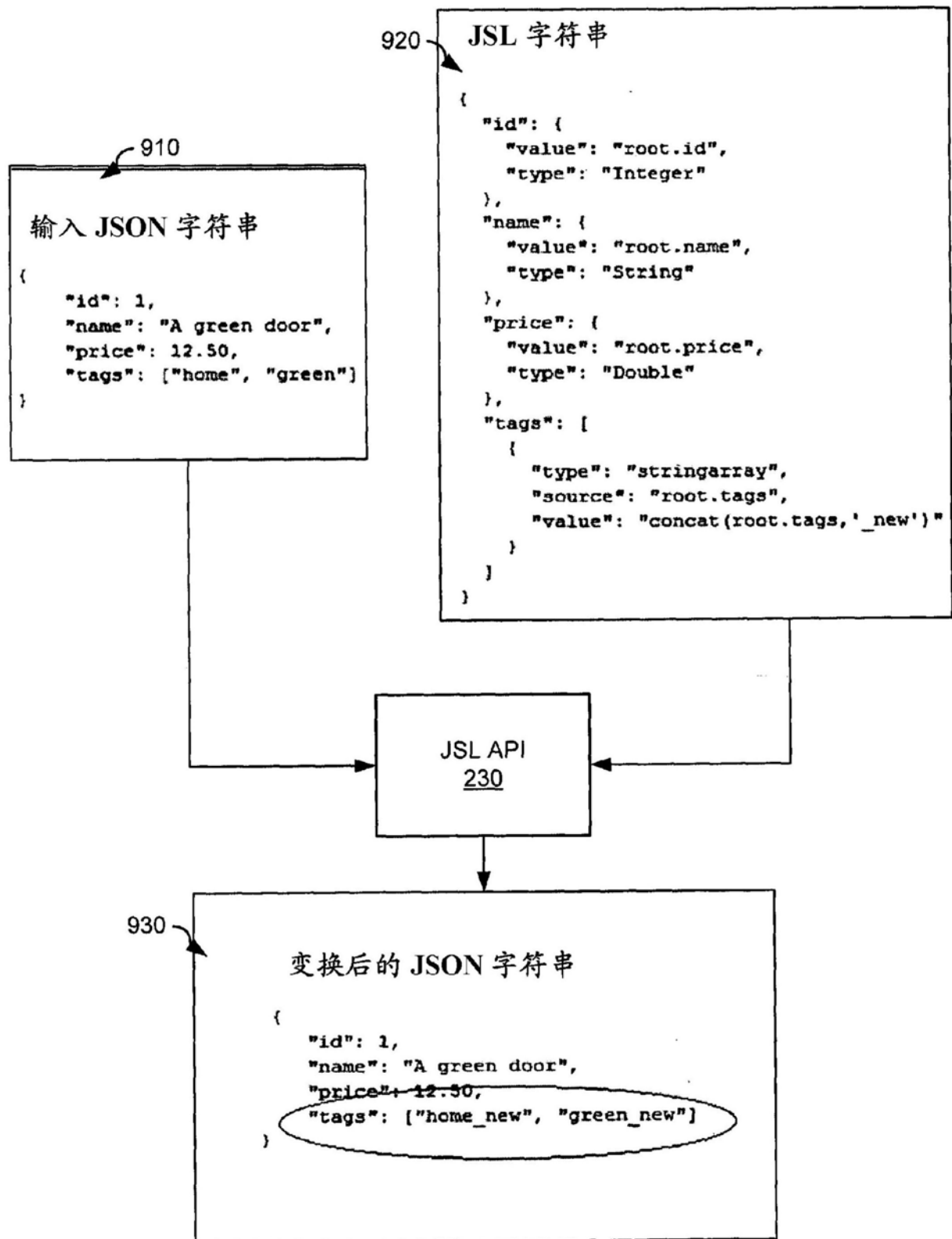


图9

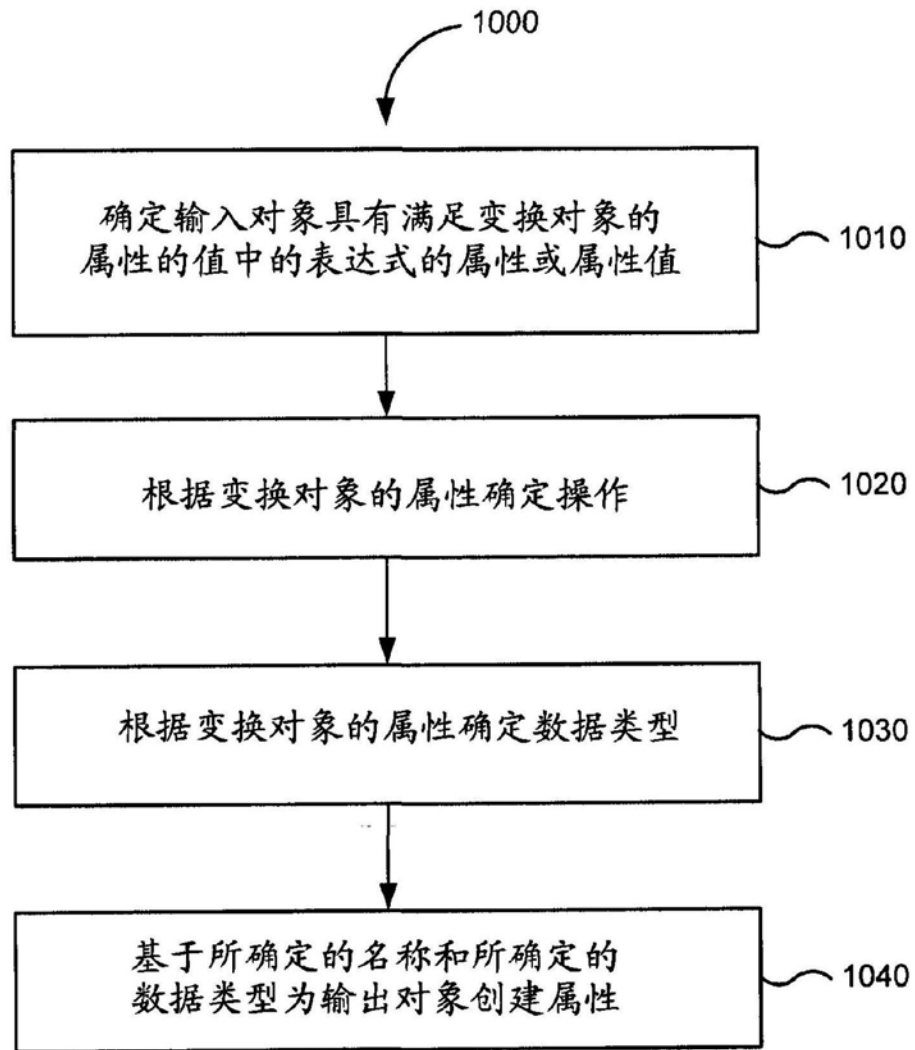


图10

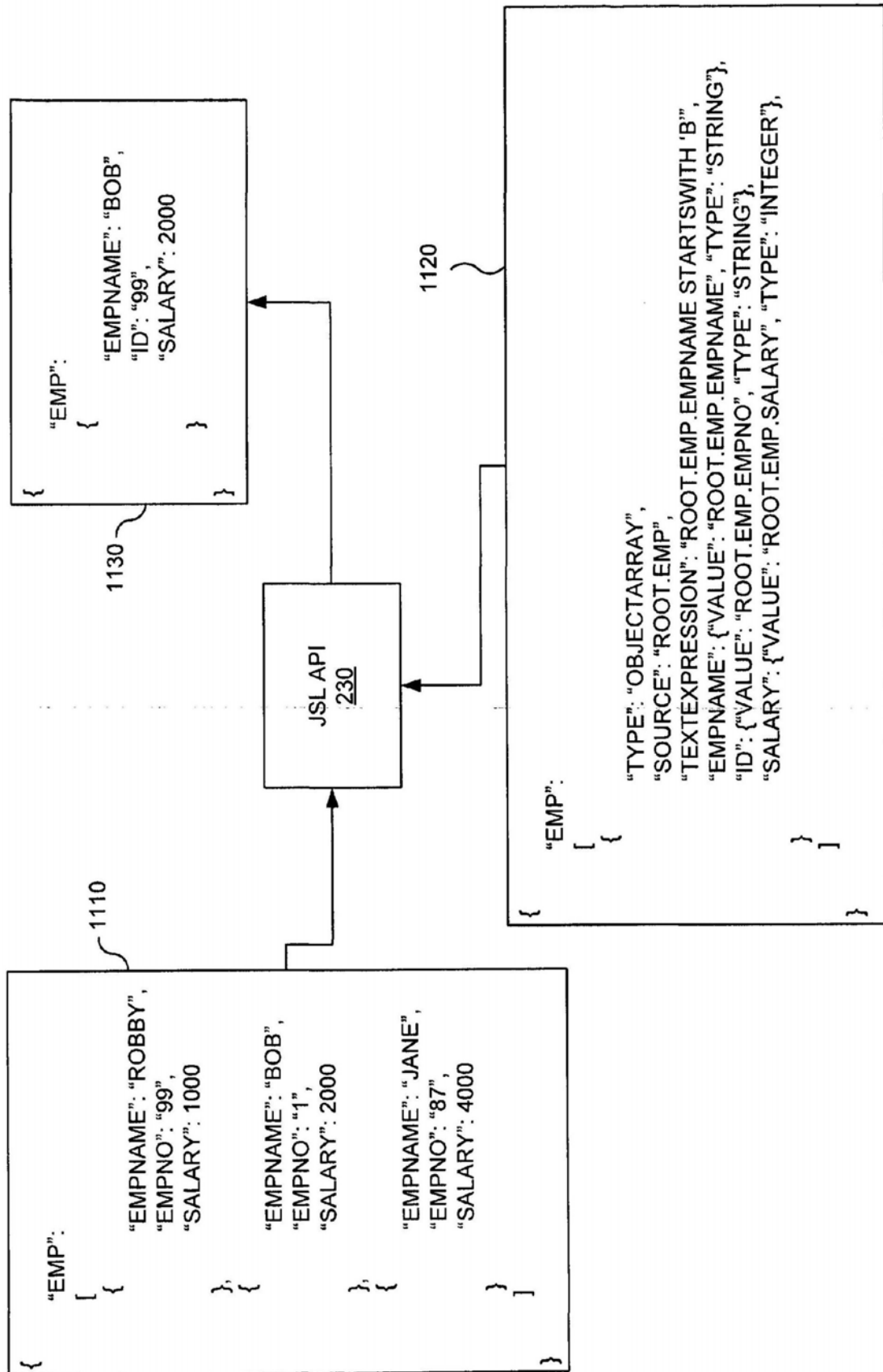


图11

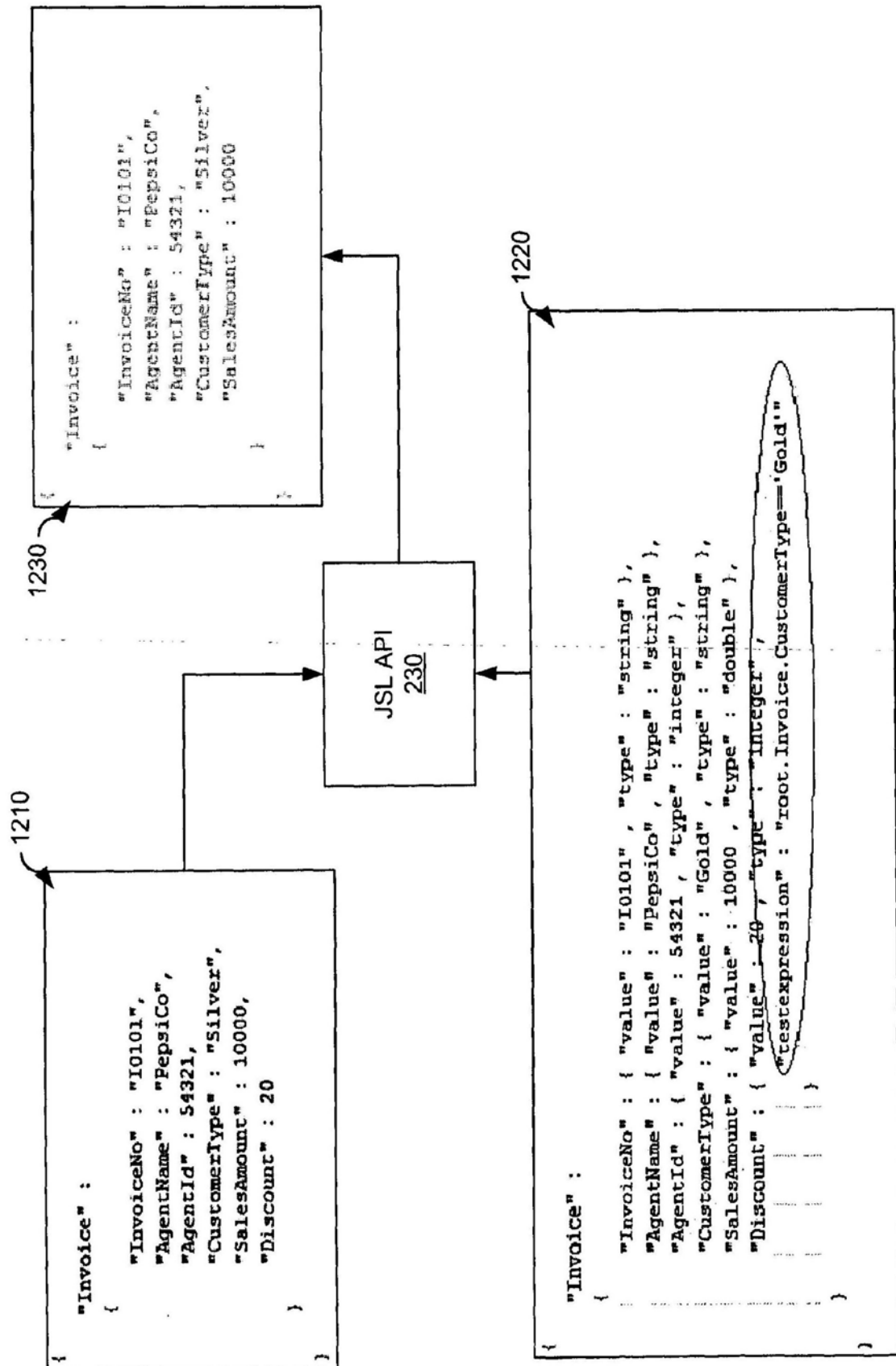


图12



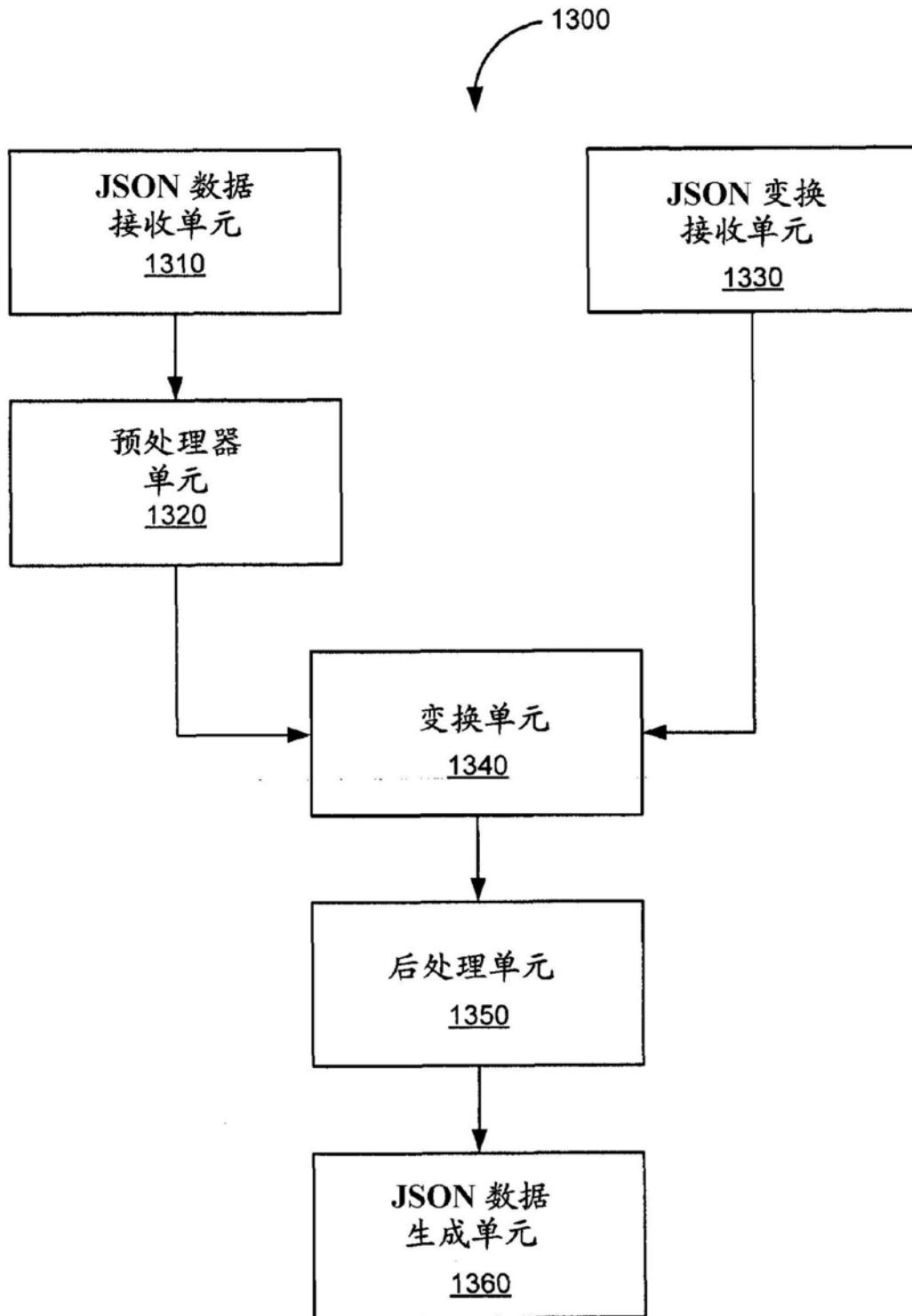


图13

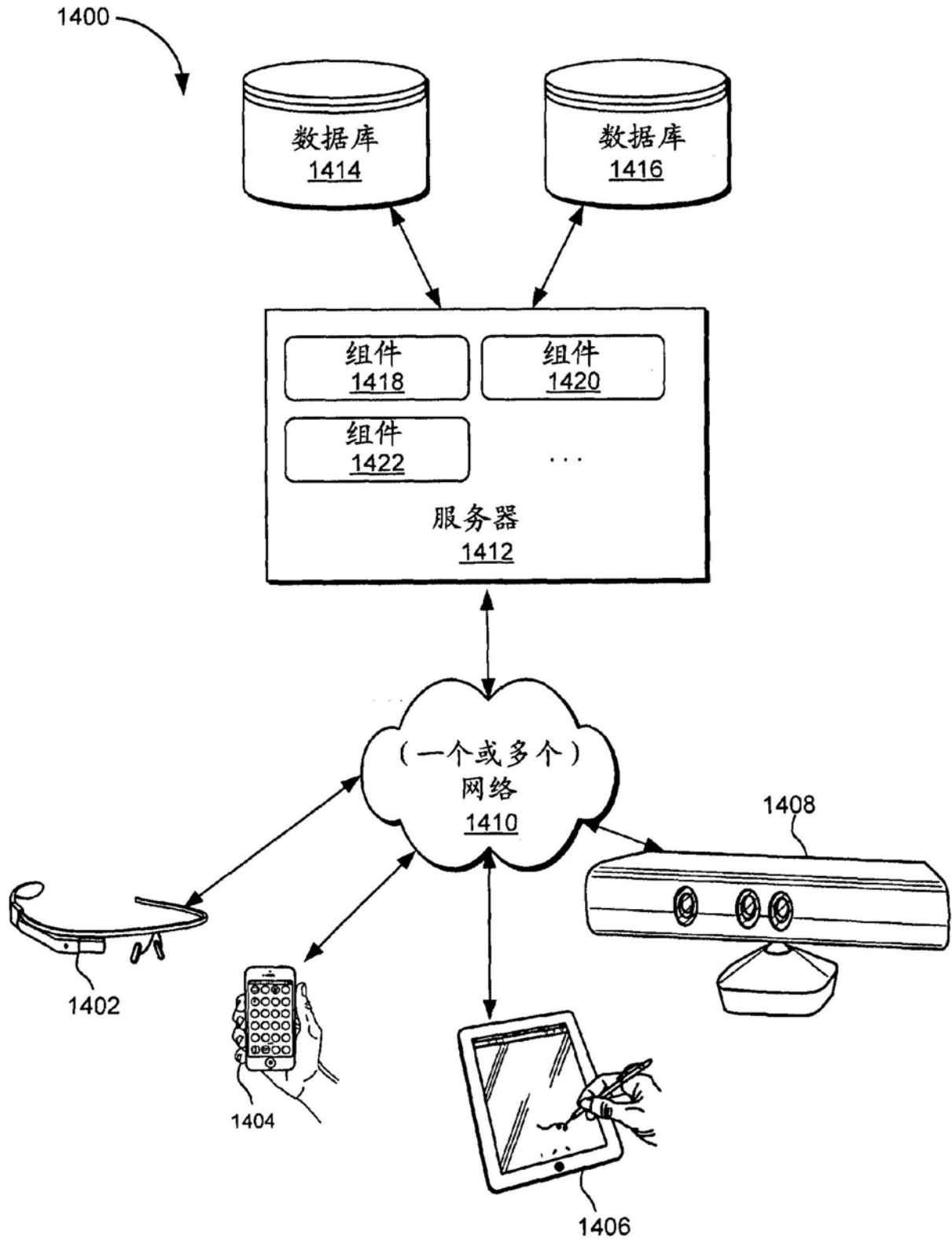


图14

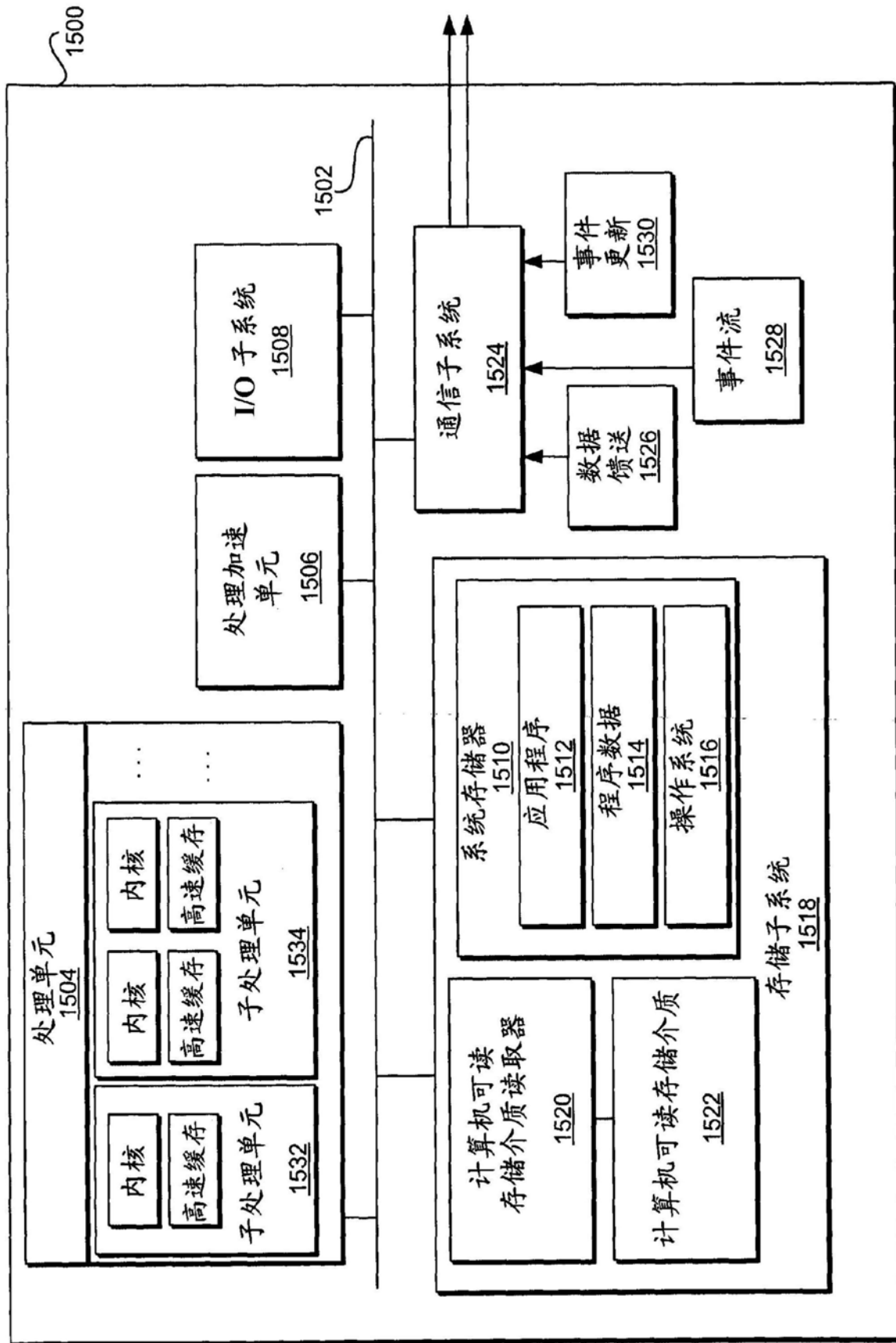


图15