(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104662547 A
(43)申请公布日 2015.05.27

(21)申请号 201380048869.7
(22)申请日 2013.10.18
(30)优先权数据
1215/KOL/2012 2012.10.19 IN
(85)PCT国际申请进入国家阶段日
2015.03.19
(86)PCT国际申请的申请数据
PCT/US2013/065799 2013.10.18
(87)PCT国际申请的公布数据
WO2014/063124 EN 2014.04.24
(71)申请人 迈克菲股份有限公司
地址 美国加利福尼亚州
(72)发明人 S·纳卢瑞 D·库尔卡尼 R·辛哈 V·科瑞斯纳普 V·K·纳加拉贾 K·K·度若 K·霍尔德
(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 毛力
(51)Int.Cl.
G06F 21/10(2006.01)
G06F 9/44(2006.01)

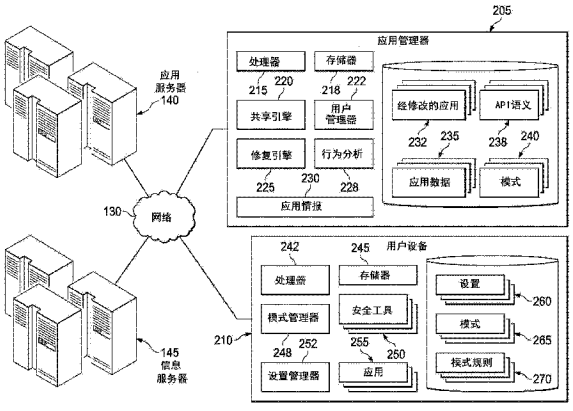
权利要求书6页 说明书21页 附图23页

(54)发明名称

移动应用管理

(57)摘要

对照特定平台的软件开发包的语义模型分析特定应用的代码。该语义模型将多个应用行为与该特定平台各自的应用编程接口(API)调用相关联。基于对该代码的分析,标识该特定应用的一组行为,并且将这组行为中特定的一个行为标识为不期望的行为。可自动地修改该特定的应用以补救该不期望的行为。可将该特定应用分配到多个设备模式中的一个模式,并且对用户设备上的该特定应用的访问可基于多个用户模式中的哪一个模式在该用户设备上是活动的。



CN 104662547 A

1. 至少一种机器可访问存储介质，具有存储于其上的多条指令，当在机器上执行所述多条指令时，所述多条指令使所述机器用于：

对照特定平台的软件开发包的语义模型分析特定应用的代码，其中，所述语义模型将多个应用行为与所述特定平台各自的应用编程接口（API）调用相关联；

基于对所述代码的所述分析，标识所述特定应用的一组行为；以及

标识所述一组行为中的特定行为是不期望的行为。

2. 如权利要求 1 所述的存储介质，其特征在于，标识所述特定行为是不期望的行为包括：确定一个或多个行为违反一条或多条规则。

3. 如权利要求 2 所述的存储介质，其特征在于，所述规则与特定的用户相关联。

4. 如权利要求 3 所述的存储介质，其特征在于，所述规则的至少一部分包括由所述特定的用户定义的多条规则。

5. 如权利要求 2 所述的存储介质，其特征在于，所述多条规则与网络服务提供商相关联。

6. 如权利要求 1 所述的存储介质，其特征在于，用户输入标识所述特定行为是不期望的。

7. 如权利要求 6 所述的存储介质，其特征在于，结合显示所标识的所述一组行为的人类可读描述的用户界面，接收所述用户输入。

8. 如权利要求 7 所述的存储介质，其特征在于，使用用于生成所述描述和所述语义模型的模板，生成所述人类可读描述。

9. 如权利要求 1 所述的存储介质，其特征在于，所述特定的用户设备是智能电话和平板计算设备中的一个。

10. 一种方法，所述方法包括：当执行存储在如权利要求 1-9 中任意一项所述的至少一种机器可读存储介质上的多条指令时由机器执行的多个动作。

11. 一种方法，包括：

对照特定平台的软件开发包的语义模型分析特定应用的代码，其中，所述语义模型将多个应用行为与所述特定平台各自的应用编程接口（API）调用相关联；

基于对所述代码的所述分析，标识所述特定应用的一组行为；以及

标识所述一组行为中的特定行为是不期望的行为。

12. 如权利要求 11 所述的方法，其特征在于，进一步包括：至少部分地基于所述语义模型，将所述特定应用的代码反汇编成控制流，并且生成针对所述特定应用的应用逻辑的模型。

13. 如权利要求 12 所述的方法，其特征在于，应用逻辑的所述模型进一步至少部分地基于周边应用知识。

14. 如权利要求 11 所述的方法，其特征在于，进一步包括：基于所述一组行为中的一个或多个行为是不期望的行为的指示，执行补救动作。

15. 如权利要求 11 所述的方法，其中，结合在特定的用户设备上实现所述特定应用的尝试，分析所述特定应用的所述代码。

16. 如权利要求 15 所述的方法，其特征在于，进一步包括：基于所述一组行为中的一个或多个行为是不期望的行为的标识，限制在所述特定用户设备上的所述特定应用的实现。

17. 如权利要求 16 所述的方法，其特征在于，限制实现包括 ：阻止在所述特定的用户设备上安装所述特定应用。

18. 如权利要求 16 所述的方法，其特征在于，限制实现包括 ：将所述特定应用分配到将限制对所述特定应用的访问的设备模式中。

19. 如权利要求 16 所述的方法，其特征在于，进一步包括，修改所述特定应用的代码以补救所述不期望的行为。

20. 一种系统，包括用于执行如权利要求 11-19 中任意一项所述的方法的装置。

21. 一种系统，包括 ：

至少一个处理器设备 ；

至少一个存储器元件 ；以及

应用行为分析引擎，当由所述至少一个处理器设备执行所述应用行为分析引擎时，所述应用行为分析引擎适用于 ：

对照特定平台的软件开发包的语义模型分析特定应用的代码，其中，所述语义模型将多个应用行为与所述特定平台各自的应用编程接口 (API) 调用相关联 ；

基于对所述代码的所述分析，标识所述特定应用的一组行为 ；以及

标识所述一组行为中的特定行为是不期望的行为。

22. 如权利要求 21 所述的系统，其特征在于，进一步包括应用修复器引擎，所述应用修复器引擎用于 ：

标识所述特定应用的、对应于所述特定行为的一部分代码 ；以及

对所述一部分代码执行补救动作，以补救所述特定行为，并生成所述特定应用的经修复的版本。

23. 如权利要求 21 所述的系统，其特征在于，进一步包括模式管理器，所述模式管理器用于 ：

激活针对用户设备所定义的多个模式中的特定的一个模式 ；以及

根据所激活的特定模式，限制对所述特定应用的访问，其中，当激活所述多个模式中的另一模式时，使所述特定应用可被访问。

24. 如权利要求 21 所述的系统，其特征在于，进一步包括用户设备，其中，所述应用行为分析引擎用于 ：基于由所述用户设备将所述特定应用安装在所述用户设备上的尝试，将对所述代码的所述分析的结果传递到所述用户设备。

25. 至少一种机器可访问存储介质，具有存储于其上的多条指令，当在机器上执行所述多条指令时，所述多条指令使所述机器用于 ：

标识检测到的被包括在特定应用中的一组行为中的特定行为 ；

标识所述特定应用的、对应于所述特定行为的一部分代码 ；以及

对所述一部分代码执行补救动作，以补救所述特定行为，并生成所述特定应用的经修复的版本。

26. 如权利要求 25 所述的存储介质，其特征在于，所述补救动作保留除所述特定行为之外的所述特定应用的其他行为。

27. 如权利要求 25 所述的存储介质，其特征在于，所述特定行为被标识为不期望的行为。

28. 如权利要求 25 所述的存储介质，其特征在于，所述补救动作包括删除所述一部分代码。

29. 如权利要求 25 所述的存储介质，其特征在于，所述补救动作包括重写所述一部分代码。

30. 如权利要求 29 所述的存储介质，其特征在于，基于对应于所述特定行为的代码的已知的替代代码，将重写所述一部分代码。

31. 如权利要求 30 所述的存储介质，其特征在于，所述特定行为包括尝试与不受信任的系统通信，并且将重写所述一部分代码以将通信重定向到受信任的系统。

32. 如权利要求 25 所述的存储介质，其特征在于，所述补救动作包括将附加的代码添加到所述应用中以使所述特定行为无效。

33. 如权利要求 25 所述的存储介质，其特征在于，从策略中标识出所述补救动作，所述策略标识被确定为适用于补救所述特定行为的补救型式。

34. 如权利要求 25 所述的存储介质，其特征在于，所述补救动作包括插入应用逻辑，所述应用逻辑允许用户在启动用户设备上经修复的应用时，能够选择性地启用所述特定行为的经修复的版本。

35. 如权利要求 34 所述的存储介质，其特征在于，所述逻辑进一步允许所述用户能够选择性地启用所述特定行为的未经修复的版本来代替所述经修复的版本。

36. 一种方法，所述方法包括：当执行存储在如权利要求 25-35 中任意一项所述的至少一种机器可读存储介质上的多条指令时由机器执行的多个动作。

37. 一种方法，包括：

标识检测到的被包括在特定应用中的一组行为中的特定行为；

标识所述特定应用的、对应于所述特定行为的一部分代码；以及

对所述一部分代码执行补救动作，以补救所述特定行为，并生成所述特定应用的经修复的版本。

38. 如权利要求 37 所述的方法，其特征在于，所述补救动作保留除所述特定行为之外的所述特定应用的其他行为。

39. 如权利要求 37 所述的方法，其特征在于，进一步包括，通过对所述特定应用的代码的分析，检测所述特定应用的所述一组行为。

40. 如权利要求 37 所述的方法，其特征在于，通过用户请求来触发所述补救动作。

41. 如权利要求 37 所述的方法，其特征在于，基于检测到所述特定行为，自动地继续进行所述补救动作，而没有用户的干预。

42. 如权利要求 37 所述的方法，其特征在于，进一步包括：

对照特定平台的软件开发包的语义模型分析特定应用的代码，其中，所述语义模型将多个应用行为与所述特定平台各自的应用编程接口（API）调用相关联；

基于对所述代码的所述分析，标识所述特定应用的所述一组行为；以及

标识所述特定行为是不期望的行为。

43. 一种系统，包括用于执行如权利要求 37-42 中任意一项所述的方法的装置。

44. 一种系统，包括：

至少一个处理器设备；

至少一个存储器元件；以及

应用修复器引擎，当由所述至少一个处理器设备执行所述应用修复器引擎时，所述修复器引擎适用于：

标识检测到的被包括在特定应用中的一组行为中的特定行为；

标识所述特定应用的、对应于所述特定行为的一部分代码；以及

对所述一部分代码执行补救动作，以补救所述特定行为，并生成所述特定应用的经修复的版本。

45. 如权利要求 44 所述的系统，其特征在于，所述补救动作保留除所述特定行为之外的所述特定应用的其他行为。

46. 如权利要求 44 所述的系统，其特征在于，进一步包括应用行为分析引擎，所述应用行为分析引擎用于：

对照特定平台的软件开发包的语义模型分析特定应用的代码，其中，所述语义模型将多个应用行为与所述特定平台各自的应用编程接口（API）调用相关联；

基于对所述代码的所述分析，标识所述特定应用的所述一组行为；以及

标识所述特定行为是不期望的行为。

47. 如权利要求 44 所述的系统，其特征在于，进一步包括模式管理器，所述模式管理器用于：

激活针对用户设备所定义的多个模式中的特定的一个模式，其中，当激活所述特定模式时，启用对安装在所述用户设备上的所述特定应用的所述经修复的版本的访问。

48. 如权利要求 44 所述的系统，其特征在于，进一步包括用户设备，其中，所述应用行为分析引擎用于将所述特定应用的所述经修复的版本传递到所述用户设备，用于在所述用户设备上安装所述特定应用的所述经修复的版本。

49. 至少一种机器可访问存储介质，具有存储于其上的多条指令，当在机器上执行所述多条指令时，所述多条指令使所述机器用于：

标识所接收的用户输入；

基于所述所接收的用户输入，激活针对特定用户设备所定义的多个模式中的特定模式；以及

在所述特定用户设备处，根据所激活的特定模式，将访问限制于安装在所述特定用户设备上的一个或多个应用，其中，当激活所述多个模式中的另一模式时，一个或多个受限的应用是可访问的。

50. 如权利要求 49 所述的存储介质，其特征在于，响应于由所述特定用户设备的用户输入的特定通行码，激活所述特定模式，其中，所述多个模式中的每一个模式与对应的通行码相关联。

51. 如权利要求 50 所述的存储介质，其特征在于，所述特定模式包括：

基于所述特定通行码的输入，从所述多个模式中标识所述特定模式；以及

基于所述通行码的输入，认证对所述特定模式的访问。

52. 如权利要求 49 所述的存储介质，其特征在于，所述多个模式是用户定义的模式。

53. 如权利要求 49 所述的存储介质，其特征在于，所述多个模式包括允许修改所述多个模式的管理模式。

54. 如权利要求 49 所述的存储介质，其特征在于，所述多个模式中的至少一个模式是能从远离所述特定的用户设备的模式共享服务中下载的模式的实例。

55. 如权利要求 49 所述的存储介质，其特征在于，至少部分地基于检测到使用所述特定用户设备的功能的特定情境，自动地激活所述特定模式。

56. 如权利要求 55 所述的存储介质，其特征在于，由所述特定用户设备的传感器检测所述特定情境。

57. 如权利要求 56 所述的存储介质，其特征在于，所述传感器包括加速度计、相机、陀螺仪、全球定位系统、电池状态传感器和生物测定传感器中的一个或多个。

58. 如权利要求 49 所述的存储介质，其特征在于，基于针对所述特定模式所定义的规则，限制所述应用中的至少特定的一个应用。

59. 如权利要求 58 所述的存储介质，其特征在于，所述所定义的规则与所述特定应用的检测到的行为有关。

60. 如权利要求 49 所述的存储介质，其特征在于，所述多个模式包括被指定为针对等待行为分析或补救的应用的隔离模式的模式。

61. 一种方法，所述方法包括 ：当执行存储在如权利要求 49-60 中任意一项所述的至少一种机器可读存储介质上的多条指令时由机器执行的多个动作。

62. 一种方法，包括 ：

标识用户输入 ；

基于所述所接收的用户输入，激活针对特定用户设备所定义的多个模式中的特定模式 ；以及

在所述特定的用户设备处，根据所激活的特定模式，将访问限制于安装在所述特定用户设备上的一个或多个应用，其中，当激活所述多个模式中的另一模式时，一个或多个受限的应用是可访问的。

63. 如权利要求 62 所述的方法，其特征在于，进一步包括 ：在所述特定模式是活动的时候，将替代的设备配置应用于所述特定用户设备。

64. 如权利要求 63 所述的方法，其特征在于，所述替代的设备配置限制对所述特定用户设备的一个或多个子系统的访问。

65. 如权利要求 64 所述的方法，其特征在于，所述替代的设备配置是多个替代的设备配置中的、能够在激活所述多个模式中各自的每一个模式期间应用的特定设备配置。

66. 如权利要求 64 所述的方法，其特征在于，所述一个或多个子系统包括相机、email 客户端、Wifi 适配器、电话模块、USB 端口和 SMS 模块中的一个或多个。

67. 如权利要求 62 所述的方法，其特征在于，在远离所述特定用户设备的设备处接受所述用户输入。

68. 一种系统，包括用于执行如权利要求 62-67 中任意一项所述的方法的装置。

69. 一种系统，包括 ：

至少一个处理器设备 ；

至少一个存储器元件 ；以及

模式管理器，当由所述至少一个处理器设备执行所述模式管理器时，所述模式管理器适用于 ：

标识用户输入；

基于所述所接收的用户输入，激活针对特定用户设备所定义的多个模式中的特定模式；以及

在所述特定的用户设备处，根据所激活的特定模式，将访问限制于安装在所述特定用户设备上的一个或多个应用，其中，当激活所述多个模式中的另一模式时，一个或多个受限的应用是可访问的。

70. 如权利要求 69 所述的系统，其特征在于，进一步包括应用行为分析引擎，所述应用行为分析引擎用于：

对照特定平台的软件开发包的语义模型分析一个或多个特定应用中的特定应用的代码，其中，所述语义模型将多个应用行为与所述特定平台各自的应用编程接口 (API) 调用相关联；

基于对所述代码的所述分析，检测所述特定应用的所述一组行为；以及

标识所述特定行为是不期望的行为。

71. 如权利要求 70 所述的系统，其特征在于，至少部分地基于所述不期望的行为，使用所述特定模式来限制对所述特定应用的访问。

72. 如权利要求 70 所述的系统，其特征在于，进一步包括应用修复器引擎，所述应用修复器引擎用于：

标识所述特定应用的、对应于所述特定行为的一部分代码；以及

对所述一部分代码执行补救动作，以补救所述特定行为，并生成所述特定应用的经修复的版本。

73. 如权利要求 69 所述的系统，其特征在于，至少部分地远离所述特定用户设备实现所述模式管理。

74. 如权利要求 69 所述的系统，其特征在于，所述系统包括所述特定用户设备。

移动应用管理

技术领域

[0001] 本公开总体涉及计算机安全领域，更具体而言，涉及移动设备的安全。

背景技术

[0002] 诸如智能电话、PDA、膝上型计算机、上网本和平板之类的移动设备的分布和使用已迅速地增长。此外，在一些发达市场中，此类设备的采用也在扩展，并且此类设备的数量赶超了台式计算机和功能电话的数量。移动设备的操作系统和硬件能力的复杂度也在增加，并且在一些情况下，其复杂度超过了传统计算机的特征集和功能的复杂度。例如，现代的移动设备可能拥有一般不包括在传统设备上的诸如位置传感器（像全球定位系统(GPS)、加速度计、陀螺仪、近场通信设备(NFC)等）之类的多样的传感器和子系统。除此之外，一些移动设备的总是连接的性质以及设备的拥有者持续地携带设备的趋势使得这些移动设备已成为恶意软件开发者、黑客和其他恶意行为人感兴趣的目标。此外，“app商店”和其他开放市场已使已经为此类设备（包括诸如谷歌安卓™(Google Android™)、iOS、Windows™等的设备平台）所开发的数以万计的应用（或“app”）的发展成为可能，而这些应用中的一些应用的质量和意图是可疑的。

附图说明

[0003] 图1是根据一个实施例的包括应用管理系统的示例系统的简化示意图；

[0004] 图2是根据一个实施例的包括示例应用管理器和用户设备的示例系统的简化框图；

[0005] 图3是表示根据一个实施例的用户设备的应用的分析和修复的简化框图；

[0006] 图4是表示根据一个实施例的应用的示例行为评估的简化框图；

[0007] 图5A-5B是根据一些实施例的示例应用之内的控制流的简化表示；

[0008] 图6是表示根据一些实施例的可由示例用户设备访问的示例子系统的简化框图；

[0009] 图7是表示根据一些实施例的表示使用规则来确定应用行为的简化框图；

[0010] 图8是表示根据一个实施例的应用行为的评估和不期望的行为的修复的简化流程图；

[0011] 图9是表示根据一个实施例的结合基于应用的行为分析被确定为包括不期望的行为的应用的管理和补救所作出的决定的简化流程图；

[0012] 图10是表示根据一个实施例的应用的示例修复的简化流程图；

[0013] 图11是表示根据一个实施例的应用的示例修复的简化框图；

[0014] 图12A-12E表示根据一些实施例的应用的不期望的行为的检测和补救的示例；

[0015] 图13是表示根据一个实施例的应用的示例修复的简化流程图；

[0016] 图14A-14B是表示根据一些实施例的示例模式管理器的特征的简化框图；

[0017] 图15A-15B表示根据一些实施例的用于管理用户设备中的模式的示例算法的多个部分；

[0018] 图 16 是根据一个实施例的用于在多个设备之间共享设备模式的简化框图；

[0019] 图 17 是示出根据一个实施例的在管理设备的多个模式时情境的使用的简化框图；

[0020] 图 18 是示出根据一些实施例的用户设备上的多个模式的远程提供和 / 或激活的简化流程图；

[0021] 图 19 是表示根据一些实施例的所收集的应用信息的简化框图；

[0022] 图 20A-20D 是根据一些实施例的结合用户设备的模式管理所提供的示例用户界面的屏幕截图；

[0023] 图 21A-21C 是表示根据一些实施例的涉及示例应用管理系统的示例操作的流程图。

[0024] 各附图中同样的附图标记指示同样的元素。

具体实施方式

[0025] 图 1 示出示例系统 100，其包括例如示例应用管理服务器 105 和诸如智能电话、移动游戏系统、平板计算机、膝上型计算机、上网本等等此类示例之类的一个或多个移动用户设备 110、115、120 和 125。应用管理服务器 105 能够向这些用户设备提供一项或多项服务，从而在所下载、安装、使用或以其他方式提供给用户设备 110、115、120 和 125 的多个应用的管理中进行协助。用户设备 110、115、120 和 125 能够访问诸如集中式应用店面 (storefront) 之类的应用服务器 140，诸如例如，安卓市场 ™，等等此类示例。在一些示例中，应用服务器 140 可进一步包括可下载并安装在用户设备 110、115、120 和 125 上的软件应用的其他源。用户设备 110、115、120 和 125 能够在包括局域网和广域网（诸如，因特网）的一个或多个网络 130 上与应用管理服务器 105 通信，并且消费该应用管理服务器 105 的数据和服务。在示例应用管理服务器 105 的多项服务中，至少部分地可由通过该应用管理服务器 105 所提供的功能来分析、评估并修复可用于用户设备 110、115、120 和 125 的多个应用。此外，结合可用于用户设备 110、115、120 和 125 的多项服务，应用管理服务器 105 能够与诸如信息服务器 145 之类的其他外部系统和服务器进行交互，并且消费该其他外部系统和服务器的资源、数据和服务。例如，此类信息服务器 145 能够主管提供关于可用于用户设备 110、115、120 和 125 的多个应用的附加情报和情境的服务和数据，等等此类示例。

[0026] 通常，包括示例计算环境 100 中的系统设备（例如，105、110、115、120、125、140、145 等）的“服务器”、“客户机”、“客户机设备”、“用户设备”、“移动设备”、“计算设备”、“网络元件”、“主机”、“系统型系统实体”和“系统”可包括可用于接收、发送、处理、存储或管理与计算环境 100 相关联的数据和信息的电子计算设备。如在本文档中所使用的，术语“计算机”、“处理器”、“处理器设备”或“处理设备”旨在涵盖任何合适的处理设备。例如，在计算环境 100 之内示出为单个设备的元件可使用诸如包括多个服务器计算机的服务器池之类的多个计算设备和处理器来实现。此外，计算设备中的任何、全部或一些可适用于执行任何操作系统以及虚拟机，操作系统包括 Linux™、UNIX™、微软 Windows™、苹果 OS™、苹果 iOS™、谷歌安卓 ™、Windows Server™等，并且虚拟机可适用于虚拟化对包括自定义和专属操作系统的特定操作系统的执行。

[0027] 此外，服务器、用户设备、网络元件、系统和其他计算设备可各自包括一个或多个

处理器、计算机可读存储器以及一个或多个接口，等等此类特征和硬件。服务器可包括任何合适的软件组件或模块，或包括能够主管和 / 或服务于包括分布式、企业或基于云的软件应用、数据和服务的软件应用和服务（例如，服务器 105 的个人安全系统、服务和应用等）的计算设备。例如，在一些实现方案中，通过云实现的系统可至少部分地包括计算系统 100 的应用管理服务器 105、应用服务器 140、信息服务器 145 或其他子系统，该云实现的系统配置成用于远程地主管、服务或以其他方式管理数据、软件服务和应用，这些数据、软件服务和应用与系统 100 中的其他服务和设备对接、相协调，依赖于系统 100 中的其他服务和设备，或者以其他方式由系统 100 中的其他服务和设备使用。在一些实例中，服务器、系统、子系统、或计算设备可实现为可在共同的计算系统、服务器、服务器池或云计算环境上受主管的、并且共享包括共享存储器、处理器和接口的计算资源的多个设备的一些组合。

[0028] 用户设备、端点设备或客户机计算设备（如，110、115、120 和 125 等）可包括传统计算设备和移动计算设备，包括，设计成用于与人类用户交互、并且能够在一个或多个网络（如，130）上与其他设备通信的个人计算机、膝上型计算机、平板计算机、智能电话、个人数字助理、功能电话、手持式视频游戏控制台、台式计算机、启用互联网的电视机和其他设备。计算机辅助的或“智能的”装置可包括家用或工业设备和机器，这些家用或工业设备和机器包括计算机处理器，并且由该计算机处理器、该计算机处理器的其他硬件和 / 或该计算机处理器所执行的一个或多个软件程序控制、监测、辅助、补充，或者通过该计算机处理器、该计算机处理器的其他硬件和 / 或该计算机处理器所执行的一个或多个软件程序以其他方式增强这些设备的功能。计算机辅助的装置可包括各种计算机辅助的机器和产品，其包括冰箱、洗衣机、汽车、HVAC 系统、工业机械、炉、安全系统等。

[0029] 用户计算设备、计算机辅助装置、服务器和计算设备的属性一般在设备之间差异很大，这些属性的差异包括各自的操作系统以及所加载、安装、执行、操作或以其他方式可由每个设备访问的软件程序的集合。例如，计算设备可运行、执行、已安装或以其他方式包括程序的各种集合，包括能够由各自的设备运行、执行或以其他方式使用的操作系统、应用、插件、小应用程序、虚拟机、机器镜像、驱动程序、可执行文件和其他基于软件的程序的各种组合。

[0030] 一些系统设备可进一步包括由这些系统设备的多个计算机处理器所支持的至少一个图形显示设备和多个用户界面，该至少一个图形显示设备和多个用户界面允许用户看到系统中所提供的应用和其他程序的图形用户界面并与之交互，系统中所提供的应用和其他程序的图形用户界面包括程序和图形用户界面的用户界面和图形表示，其中，上述程序与在系统设备之内受主管的应用进行交互，上述图形用户界面与应用管理服务器服务和其他应用等相关联。此外，虽然可按照由一个用户使用来描述系统设备，但是本公开构想了许多用户可使用一个计算机或一个用户可使用多个计算机。

[0031] 虽然图 1 被描述为包含多个元件或与多个元件相关联，但是并非图 1 中的计算环境 100 之内所示出的所有元件可被用于本公开的每一个替代实现方案中。此外，结合图 1 中的多个示例所描述的一个或多个元件可位于计算环境 100 的外部，而在其他实例中，某些元件可被包括在一个或多个其他所描述的元件以及未在所示出的实现方案中描述的其他元件之中，或者是一个或多个其他所描述的元件以及未在所示出的实现方案中描述的其他元件的部分。另外，图 1 中所示出的某些元件可以与其他组件相组合，并且可用于除本文

所描述的那些目的之外的替代目的或附加目的。

[0032] 现在转向图 2 中的示例框图，示出了示例系统，该示例系统包括在一个或多个网络 130 上进行通信的应用管理器 205、用户系统 210 以及包括例如应用服务器 140 和信息服务器 145 的其他计算设备和网络元件等。在一个示例实现方案中，应用管理器 205 可包括一个或多个处理器设备 215、存储器元件 218 以及一个或多个其他软件和 / 或硬件实现的组件。例如，在一个示例实现方案中，应用管理器 205 可包括共享引擎 220、用户管理器 222、修复引擎 225、行为分析引擎 228、应用情报引擎 230 以及包括上述各项的多个组合的其他潜在的机器可执行逻辑、组件和功能等。

[0033] 在一个示例中，共享引擎 220 可配置成用于提供管理关于应用（例如，通过应用服务器 140 而变得可用）的信息的众包 (crowdsourcing) 以及共享此类信息和资源的功能，此类资源包括至少部分地由应用管理器 205 生成的或由应用管理器 205 收集的资源。例如，示例共享引擎 220 可允许为特定用户和相关联的用户设备（如，210）所开发的经修改的应用 232 以及所定义的应用模式 240 横跨多个用户设备（如，210）被共享，等等此类示例。示例用户管理器 222 可提供管理消费或以其他方式使用应用管理器 205 的多项服务的各种用户设备（如，210）的用户账户的功能。示例用户管理器 222 可将各种经修改的应用 232、应用数据和反馈数据（如，235）和包括由特定用户所开发或修改的应用模式的应用模式 240 与系统中的一个或多个用户账户和用户设备（如，210）相关联，等等此类示例。

[0034] 在一些实现方案中，应用管理器 205 可附加地包括能够结合用户设备下载、安装、激活或以其他方式使用或获得各种应用（包括通过一个或多个应用服务器（如，140）所提供的多个应用）的尝试，向一个或多个用户设备（如，210）提供应用管理、安全和诊断服务的多个组件、引擎和模块。例如，在一个示例实现方案中，应用管理器 205 可包括示例行为分析引擎 228，该示例行为分析引擎 228 适用于分析并标识可用于系统上的多个用户设备的各种应用的功能。此外，可例如通过行为分析引擎 228 来标识用户或管理员可能希望阻止、限制、修复或修改的应用的功能，等等此类示例。相应地，在一些实施方案中，示例应用管理器 205 可包括示例修复引擎 225，该示例修复引擎 225 配置成用于代表用户来修改应用，从而消除例如由行为分析引擎 228 检测到的不期望的应用特征，并且进而生成经修改的应用 232。在一些示例中，可基于对应用户的请求、规则、设置和偏好来具体地修改并配置经修改的应用 232。此外，应用管理器 205 可包括应用情报引擎 230，该应用情报引擎 230 配置成用于收集例如来自信息服务器 145 以及在应用管理器 205 及其客户机用户设备（如，210）的内部和外部的其他源的应用数据（如，235）。应用情报引擎 230 可用于收集关于例如由应用服务器 144 服务的一个或多个应用的情报。可结合应用管理器 205 所提供的多项服务（诸如由应用管理器 205 进行的对应用的行为分析和评估，等等此类示例）来使用该情报。

[0035] 在一些实现方案中，用户设备（如，210）可包括一个或多个处理器设备 242 和一个或多个存储器元件 245 以及一个或多个软件和 / 或硬件实现的组件，这些软件和 / 或硬件实现的组件包括例如，模式管理器 248、设置管理器 252、安全工具 250 以及一个或多个应用 255（如，通过应用服务器 140 所获取的应用）。在一个示例实现方案中，用户设备 210 可包括模式管理器 248，该模式管理器 248 配备有用于对用户设备 210 上的多个应用访问模式 265 进行定义、实施以及以其他方式进行管理的功能。模式规则 270 可另外由模式管理器

248 管理，该模式规则 270 定义例如在用户设备 210 上自动地发起或实施各种模式 265 的多个特定条件。此外，可由用户例如通过示例设置管理器 252 来定义一个或多个设置 260，该设置对应于设备 210 的多种模式 265，并且在一些情况下结合设备 210 的多种模式 265 被使用，等等此类示例。

[0036] 转至图 3 的示例，示出简化框图 300，其示出示例应用管理器的功能和流。例如，行为监测器 228 可评估多个应用，以标识应用的一个或多个功能和 / 或内容是好的、差的、可疑的还是质量未知的，等等此类示例。该评估可基于从诸如信息服务器、用户反馈和其他源之类的多种源（如，145）中获取的信息。在标识了“差的”应用功能和 / 或内容的实例中，应用修复引擎 225 可致力于修改该应用并补救所标识的不期望的功能，以生成对应于该应用的修复版本的经修改的应用文件 232。此外，例如可由模式管理器 248 将可疑的或未知的应用指定为专用于用户设备 210 的特定的受限的访问模式，从而实际上隔离该可疑的应用，直到获取关于该应用的更多情报为止。在确定应用符合用户、网络、管理员等的规则、要求或偏好的实例中，反而可允许该应用继续在用户设备上进行安装。此外，已经被修复而生成了经修改的应用文件的应用可允许该经修改的应用继续去往用户设备，从而安装在此设备上，等等此类示例。

[0037] 图 4 包括示出通过示例应用行为分析引擎实现的示例原理和活动的框图 400。可由反汇编器和数据 / 控制流分析器 410 访问或接收应用二进制文件 405，该反汇编器和数据 / 控制流分析器 410 结合诸如应用描述、审核、评论和其他结构化和非结构化的数据之类的周边 (ambient) 应用知识 415（例如，从外部信息源以及用户、评论者等处收集到的），可针对每一个应用二进制文件 405 开发应用逻辑的模型 420。反汇编器和控制流分析器 410 可基于例如将代码或应用逻辑模型与软件开发包和 / 或公共 API 中所定义的或从软件开发包和 / 或公共 API 中识别出的已知功能进行比较来标识给定的应用的行为 425，该软件开发包和 / 或公共 API 是由对应的客户机设备操作系统和与该客户机设备兼容的大多数或所有的应用利用的。一些示例包括，谷歌安卓软件开发包、苹果 iOS 软件开发包、Windows 软件开发包，等等此类示例。

[0038] 通常，平台软件开发包（或“SDK”）可提供文档、头部文件、库、命令、接口等，这些文档、头部文件、库、命令、接口等定义并提供对可由与该平台兼容的多个应用访问的各种平台子系统的访问。在一个示例实现方案中，可在可由例如应用行为引擎用于确定与平台兼容的多个应用的行为和功能的模型中表示平台 SDK 和对应的 API 和 API 调用（即，对 API 的函数和例程的调用）。以程序可读形式并且利用导出应用行为所必需的关键信息来表示常用的 API 的语义。平台 SDK 的语义可表示为使得示例应用行为引擎能够使用该语义模型理解并标识使用该 API 调用的给定应用的操作和行为。例如，在一个示例实现方案中，例如通过 API 情报 430，可通过用描述 API 调用各自在平台上做了什么的行为标签以及这些 API 的操作和行为的对应参数来标注每个 API 调用各自的名称来表示该平台的所有潜在的 API 调用。作为示例，此类语义表示的模板可被模型化，例如：

[0039]

<APIName: name>

[0040]

<Category: read/write/process/transform/.../...>

<CategoryDetail>

<Reads: sensitivity>

<Writes: sensitivity>

<Transform: sensitivity>

<Senitivity: red:5/orange:4/yellow:3/green:1>

<Parameters: No of paramerer>

<ParameterIndex:Index>

<Type: integer/object/string/../..>

<Operation: input/output/transformative>

<return value: void/integer/object/string/>

<Dependency>

<True/False>

<Description>

<APIDescription: description of the API>

<Verbs:xxx>

<Nouns:xxx>

[0041] 在上述示例中，“category”（“类别”）可指定 API 调用的类型，并且可用于标识此类 API 调用的总体功能，诸如，该 API 调用从特定的子系统、盘等中读取信息；生成各种消息；发起各种网络行为；尝试与各种外部服务器通信；以及触发特定的设备功能或元件（例如，相机、SMS 控制器等）。“Sensitivity”（“敏感度”）可表示在结合子系统的恶意行为的潜在可能性的情境中，该子系统受 API 影响或者由该 API 被关联的各自的敏感度，例如，在该子系统潜在地允许恶意软件的引入、未经授权的跟踪或数据收集、对 SMS 或 email 消息的未经授权或不期望的读取或发送等等许多此类示例的情况下，读取特定的存储器位置是否引入间谍活动的潜在可能。此外，“dependency”（“依赖关系”）可表示该 API 的输出是否可直接地对该程序的其他部分存在影响。例如，sendTextMessage()（发送文本消息 ()）API 可被标识为不具有依赖关系，其中，该 API 仅发送出消息而不返回任何消息，等等此类示例。

[0042] 行为启发式 (heuristics)/ 规则引擎 435（如，示例分析引擎（如，228）的行为启发式 / 规则引擎）可将其他信息用于确定受评估的应用的行为，其他信息诸如是聚合来自源社群 445、规则 450 和其他信息的全球威胁情报 (GTI)440。

[0043] 如上文所述，示例应用行为分析引擎（如，228）可拥有基于例如兼容的应用所基于的标准平台 SDK 的语义表示来标识给定的应用的控制流、操作、功能和行为的功能。在图 5A 中，示出针对示例游戏应用的简化的应用控制流的表示 500。当该游戏的功能与平台

SDK 的语义标识和针对该游戏应用的周边应用情报相比之下处于该游戏应用二进制文件的代码的主要期望的、安全的以及良性的、更深层次的监测之中的时候，该功能也识别不能由用户立即地或以其他方式识别、理解或领会的其他功能，例如，在用户明确知晓或允许的情况下或者用户不明确知晓或允许的情况下发送 SMS 消息的应用。在图 5B 中所示的另一示例中，监测应用二进制文件的特定对象可揭示给定应用的全部功能和控制流，并且揭示用户原本可能无法意识到、理解或赞成的不同的程序、程序单元或应用之间的依赖关系。作为示例，在一些实现方案中，可在标识数据流和数据调用的特定型式 (pattern) 的 XML 文件中外部地表示所标识的行为启发，可从这些特定型式中可识别该行为。例如：

[0044]

```
<Pattern>
    < Call to API1(): mandatory>
    < Call to API2()/API3()/….: mandatory>
    < Call to API5()/API6()/….: optional>
    < Call to API10(): mandatory>
</Pattern>
```

[0045] 在一些实现方案中，基于例如平台 SDK 的语义表示的模型，应用逻辑可被模型化，并且可将多个规则应用于解释该应用逻辑以及标识该应用的对应的二进制文件之内的、与恶意行为、侵犯隐私行为、违反策略行为或其他不期望的行为对应的多个指令与调用。应用的功能的逻辑模型可包括通过数据流结构和控制流结构等等此类示例进行的应用逻辑的表示（如，505）。数据流结构可表示数据对象经过应用逻辑（如，510）并且到达包括外部程序单元的其他程序单元（如，515）之上的生命期。数据流结构（如，505）可用于在来自应用程序的一个部分的数据的流移动时标识该流，并且潜在地被应用逻辑变换。例如，数据流模型可用于推断特定的数据正由该应用通过因特网通信投递操作被泄露，等等此类示例。此外，控制流结构可表示用于标识被确定为敏感的或不期望的应用调用的原始源的不同的函数调用（如，520、525）的控制流。作为说明性的示例，可由用户将由应用进行发送 SMS 消息的调用（或者甚至是该应用后台处理中的匿名事件，等等潜在的许多此类示例）回溯至例如与所交互的应用的 UI 元件。

[0046] 转向图 6 中的示例，示出了表示例如可由多个应用通过平台 SDK 中所定义的一个或多个 API 来访问的各种子系统、设备和功能的简化框图。在一些实现方案中，在可以结合恶意行为或其他方式的不期望的行为操纵或利用子系统的潜在可能的情境中，基于各个子系统的敏感度，可对所有的平台子系统进行分类和权重分配。此类权重和敏感度可基于包括例如隐私侵犯、数据泄露、金融敏感度等等此类示例的潜在可能的各种因素。这些因素可形成对平台的各种子系统进行分类的基础。此类子系统可包括例如，通讯录、相册、email 客户端、日历、因特网连接与浏览、图形、视频功能、相机、音频、安全工具和引擎、通话、Wi-Fi 能力、蓝牙能力、数据端口、电池电源、触屏、全球定位系统以及潜在的许多其他的功能和子系统，包括可被集成在移动设备中的未来功能。

[0047] 如图 7 的示例中所表示的，应用行为分析引擎的规则引擎可例如从规则数据库中访问规则，这些规则包括根据例如用户偏好以及适用于这些用户的策略（例如，因特网服

务提供商、企业网络、宽带数据提供商等的策略）为特定用户或用户群和/或由特定用户或用户群自定义的规则。该规则引擎可将应用逻辑模型（例如，基于对应于该应用的平台 SDK 的语义表示所开发的应用逻辑模型）作为附加输入以评估应用逻辑模型中所标识的应用的各种操作和功能。根据被标识为适用于应用的特定实例（例如，已尝试被下载或安装在与所标识的规则相关联的用户的特定用户计算设备上的应用的实例）的多条规则，该规则引擎可评估应用的各种操作和功能。可由该规则引擎来标识应用行为，包括将这些应用行为标识为违反一条或多条规则（如，禁止某些行为或动作的规则）且（在某些情况下）提示补救所标识的应用行为和/或将该应用分配到目的地用户设备上的、诸如隔离操作模式或管理操作模式之类的一个或多个操作模式，等等此类示例。

[0048] 在一些实现方案中，可建立标识行为的并且基于 API 语义的描述的人类可读描述。在一个示例中，人类相关的动词和名词可与语义表示中的模板消息相关联，并且可被映射至可用于 API 的功能和操作的特定的人类可理解的描述。此外，结合根据例如由应用行为分析引擎所执行的语义模型对应用进行的评估，可从上述映射中生成描述各种功能以及在一些实现方案中描述经分析的应用的控制流数据流的行为分析结果的人类可读概要，并且将该人类可读概要呈现给用户。此类结果可利用该人类可读描述以生成在对该应用进行分析期间所揭示的功能的描述，包括原本可能对用户不可见或难以被用户检测到的功能。例如，在一个实现方案中，可利用并填充该模板，从而标识并描述用于从该用户的设备中读取 SMS 数据的示例应用的功能。作为说明性示例，对应的描述可生成为例如："此应用从 SMS 收件箱中读取您的 SMS 数据并将其发送到网站上。"可例如通过填充基于诸如下列的平台 SDK 和 API 的语义表示的示例模板来建立此类描述："此应用从 <noun:SMS inbox>(< 名词 :SMS 收件箱 >)<verb :reads>(< 动词 :读取 >) 您的 <noun :SMS data>(< 名词 :SMS 数据 >) 并且将其 <verb:sends>(动词 :发送) 到 <noun :website>(< 名词 :网站 >) 上"，等等此类示例。

[0049] 在一些示例中，经分析的应用行为可揭示由该经分析的应用对其他应用、程序或服务的使用。例如在其他经调用的应用被标识为不安全的、不受信任的或未知的等等此类示例的时候，一些实例（由经分析的应用对本地应用、远程服务或其他程序的调用）可能是不期望的。在其他实例中，由经分析的应用调用或使用的程序可被标识为受信任的程序。相应地，在一些实现方案中，应用行为分析引擎可利用、生成、修改或以其他方式管理白名单和/或黑名单，这些白名单和/或黑名单标识还不为各种经分析的应用所知的或可能潜在地由各种经分析的应用调用的各种程序的状态和信誉。在一些实现方案中，由远程服务器主管的应用和服务可附加地由对应于各自的主机服务器的各自的 URL 或其他地址信息等等此类示例在此类白名单和/或黑名单中进行标识。

[0050] 在一些实现方案中，该行为分析引擎可标识由受评估的应用执行特定活动、访问平台 API 或使用功能的情境。作为示例，可基于尝试的原因或情境来评估经分析的应用访问平台通话子系统的尝试。例如，在一些情境中，特定的 API 调用可能是完全可接受的，而在另一些情境中，该 API 调用可能是不期望的。例如，可能以不同于由应用自主地访问通话子系统并且不响应于用户提供的指示等等此类示例的尝试的方式来评估响应于诸如按下按钮之类的用户界面交互访问通话子系统的所标识的应用功能。

[0051] 如上文所述，在一些实现方案中，可定义能用于应用行为的评估中的规则。此类规

则可被表示为或配置为用于执行应用逻辑或应用行为分析引擎所标识的潜在的恶意行为（包括该行为要被确定为是恶意的情境）的启发式分析。例如，规则引擎可将一条或多条规则应用于应用逻辑模型以标识存在于该应用中的一个或多个潜在恶意的或以其他方式不期望的行为。在一些实现方案中，规则可表示为：

[0052] <Rule>

[0053] <Run><Dataflow><ReadOperation>of<red sub system>to a<WriteOperation>of<write sub system>

[0054] 这些规则可以是通用的，或者可以是专用于特定的子系统的等等，诸如是检测存储个人通讯录数据的存储器元件的数据泄露的规则，等等此类示例。可基于单条规则或多条规则的应用来导出专用的应用行为。

[0055] 在一些实现方案中，可在远离为其执行分析的移动用户设备的一个或多个服务器计算设备上主管应用行为分析引擎。在其他示例中，可向应用行为分析引擎的至少部分替代地或冗余地提供服务器侧的应用行为分析引擎组件的功能。例如，在一个示例实现方案中，可向用户计算设备提供应用行为分析引擎功能，该应用行为分析引擎功能允许对要在该用户设备处执行的应用进行的至少部分的或快速的初步评估，进而向用户提供快速反馈，并且评估是否应当隔离应用、拒绝下载或安装应用和/或将该应用转发到诸如云系统中所提供的远程应用行为分析引擎之类的远程应用行为分析引擎，随后允许对该应用进行更稳健的行为分析（那可能将增加的等待时间引入行为分析评估中）。

[0056] 在一些实现方案中，在分析应用期间，可阻止或延迟下载、隔离或启动经分析的应用，直到该分析完成为止。在一些实例中，可向用户提供提示，该提示标识该应用的分析，并且向该用户提供用于处理该经分析的应用的安装、下载或启动的各种选项。例如，可向用户提供跳过分析、延迟安装该经分析的应用、将该经分析的应用分配到特定模式等等此类示例的选项。此外，在一些实现方案中，结合评估呈现给该用户的提示可以与诸如从行为分析引擎评估和/或关于该经分析的应用的外部情报中收集到的初步信息之类的信息一起被呈现出来。此类情报可包括例如由该行为分析引擎在对该经分析的应用进行的先前的评估中收集到的情报，等等此类示例。实际上，在一些实现方案中，该行为分析引擎可向针对该应用所发现的用户行为指示用户多久响应一次从该行为分析引擎接收到的、关于该特定的经分析的应用的反馈，等等此类示例。

[0057] 在一些实现方案中，行为分析引擎可保持为其所知和/或先前经其分析的多个应用的黑名单、灰名单和/或白名单。此类黑名单、灰名单和/或白名单可基于从先前的行为分析中收集到的历史情报和来自其他源以及其他用户的外部情报。该行为分析引擎可利用此类信息来执行对应用的初步评估，并且利用从先前的分析中收集到的信息。进而可向用户提供初步的过滤或反馈，以辅助该用户确定如何处理特定的应用，以及是否使用该行为分析引擎对该应用发起进一步的行为分析。

[0058] 应用的行为分析和/或黑名单/白名单可进一步包含或考虑开发者或被标识为对各种应用负责的其他方的总体信誉信息，等等此类示例和考量。可定义考虑应用的开发者、经销商等的可信度或不可信度的多条规则。例如，可由该行为分析引擎基于对开发者的应用的综合分析来计算应用开发的得分评级。例如，此类评级可导出如下：App 开发者评级＝f(app 的总数，app 中不期望的行为的加权平均，app 的流行度，低评级的平均比率)，等等

此类示例。例如，在一个说明性示例中，可针对开发者的一组应用生成不期望的行为的加权平均：

[0059]

行为	权重（最高为 10）	发生次数	总计权重
通讯录泄露	9	2	18
设备 ID 泄露	2	5	10
消息泄露 (SMS)	8	3	24
位置泄露	5	4	20
非必需的许可	2	1	2

[0060] 并且平均权重可通过“平均权重＝总计权重 /App 总数”来导出，等等此类示例实现方案。

[0061] 诸如情报数据库（如，全球威胁情报 (GTI) 馈送）之类的外部源可用于标识已横跨由经行为分析引擎评估的多个应用所采用的一个或多个网络检测到的恶意行为。例如，可标识先前已被确定为要与其他恶意攻击、恶意软件或可疑系统相关联的或要用于其他恶意攻击、恶意软件或可疑系统的各种 URL、IP 地址、电话号码和文件。此外，行为分析引擎可与情报数据库对接，以提供从由该行为分析引擎自身所执行的应用的行为分析中收集到的附加情报，等等此类示例。

[0062] 另外，在一些系统和平台中，由一个或多个应用服务器或店面所提供的应用可向用户提供针对给定的应用所收集的基本描述、评级、用户反馈等。不幸的是，在许多实例中，应用开发者可能会自身来提供、操纵或以其他方式影响此类评级、应用描述、内容评级等，进而潜在地减弱了提供给用户的、关于一些应用的信息的可信度或正当性。相应地，在一些实现方案中，从对由示例行为分析引擎执行的多个应用的行为分析中收集到的情报（如，行为描述）可用于补充、纠正或以其他方式修改结合用户浏览、购买以及下载平台上可得到的应用提供给这些用户的描述。此外，在一些实现方案中，行为分析引擎可将这些默认的应用描述、内容评级、用户反馈等用作结合行为分析而考虑的外部情报。在又一示例中，行为分析引擎可用于标识多个应用之间的、可用作根据行为对这些应用进行分类的基础的共同的行为特点。随后，可向用户提供此类类别，以辅助用户更好地理解各种应用的质量和行为以及潜在的风险，等等此类示例。

[0063] 转向图 8，示出示例流的简化的示意图 800，该示例流用于在一些示例中，执行对应用行为的深度分析（例如，使用行为分析引擎）以及试图补救被确定为在应用中是不期望的那些行为来执行应用修复，同时仍然保持该应用的其他核心功能。如所示出的那样，可将应用二进制文件提交到（例如，行为分析引擎的）反汇编器和数据控制流分析器 410，以便在一些示例中附加地基于周边应用知识 415、情报等来开发应用逻辑模型（如，420）。如上文所述，可通过行为启发式 / 规则引擎 435，基于所定义的规则、平台 API 情报和行为启发来评估应用逻辑的模型 420 以标识各自应用的应用行为。此外，在评估期间，可将多部分的

应用的代码标识为对所呈现的不期望的行为是负有责任的。可标记该代码以进行补救。此外，在由用户、管理员或预定义的规则将应用行为标识为不期望的，并且请求或规定修复这些应用行为的实例中，可进一步处理这些应用二进制文件以去除、阻止或以其他方式补救这些冒犯性行为以及对应的代码，从而生成用户随后可将其下载、安装到该用户的设备上并且在该用户的设备上执行的应用二进制文件的经修复的版本 232。此外，如上文所述，全球威胁情报馈送 440 或其他情报数据库可提供用于考虑、行为分析以及应用修复的情报。此外，从行为分析中收集到的情报可与附加地从用户和系统社群 445 接收输入、数据和情报的外部情报数据库共享。

[0064] 现在转向图 9 中的示例，所示出的附加的流程图 900 表示结合对基于应用的行为分析被确定为包括不期望的行为的应用进行的管理和补救所作出的决定。例如，可例如由用户或系统或网络管理员定义规则或策略，以定义将如何处理以及在什么条件下处理已被确定为包括一个或多个不期望的行为的应用。此类策略可例如标识特定类型的不期望的行为，并且将此类行为映射至预定义的动作过程，诸如，修复或补救应用，将应用加入黑名单或白名单，隔离应用，等等此类示例。此外，用户输入可驱动对在用户计算设备上应用的部署的管理。可结合呈现给用户的提示来接收此类输入，并且此类输入可包括例如，补救一个或多个所标识的不期望的行为的请求、用于将经分析的应用分配到特定的操作模式或隔离区的指令，等等此类示例。

[0065] 如上文所述，可由修复引擎执行对应用行为的静态修复和个性化操作，该修复引擎允许修改该应用的代码并生成允许用户在去除不期望的行为的同时保留该应用的安全的或正当的功能的该应用的“安全”版本。在一些情况下，可将此类修复个性化或定制到驱动该修复的具体定义的策略，进而允许用户、服务提供商、设备制造商等能控制并个性化要安装在对应的用户设备上的应用的功能。在图 10 中，示出简化的图 1000，该简化的图 1000 示出原始应用 1005 的示例修复的流。在如 1010 处所示标识了不期望的行为以及应用二进制文件的代码的冒犯性部分之后，可提供修复引擎，用于标识、去除、替换或阻止该冒犯性代码和对应的行为，从而生成经修改的应用二进制文件 1015。作为示例，修复引擎 228 可包括逻辑，该逻辑用于去除或阻止各种类型的不期望的行为，例如在此示例中，通过去除在原始的应用二进制文件中发现的冒犯性指令来去除对 SMS 功能的未经授权的读取或访问。在诸如此示例中所示的其他实例中，修复引擎可诸如通过重写代码以将 API 调用重定向到受信任的系统、目的地、地址等来修改该冒犯性代码。修复引擎 228 可通过作出最小改变来修改原始代码，以避免影响该应用的核心的所期望的功能。此外，修复策略可标识针对标识用于修复的应用代码所考虑的型式。这可在例如标识对应于冒犯性行为的代码的启发型式的 XML 文件中被表示出来。可由诸如根据对应的策略的修复方法之类的特定修复方法来修复每种类型的所定义或所标识的代码的型式。能以修复不影响该应用的功能的其余部分的方式标识和定义此类方法。

[0066] 可由应用修复器引擎来采用各种修复方法。例如，特定的冒犯性的代码功能行可被标识为控制链中的最终节点或叶节点。在此类实例中，该冒犯性的代码可被确定为能够被抑制或去除，而不影响应用中的其他依赖关系，等等此类示例。在另一示例中，如果去除特定的 API 调用被确定为可能对周围的代码没有影响，则可以应用该去除修复方法。可例如从语义平台 SDK 表示等等此类示例中学习 API 的性质和特性。在其他实例中，该冒犯性行

为可能来自代码的一个或多个部分，并且可导致应用于补救该行为的多个修复方法，例如，通过替换寄存器中的数据以更改该 API 的行为，或者通过用新的 API 代码替换该冒犯性的 API 代码，利用相同的界面将该 API 调用重定向到该 API 的新版本，等等此类示例。在引入 API 的新版本的实例中，新的 API 可例如什么都不做并且设置寄存器状态，从而不影响该程序的其他部分，以不同的方法来处理输入以避免不期望的行为，或者进行输入 / 输出参数的预处理和 / 或后处理以及调用原始的 API，等等此类解决不期望的行为的示例技术。

[0067] 转到图 11，示出简化的框图，该简化的框图示出与特定的不期望的行为有关的代码的标识。例如，应用代码的部分 1a 和 1b 可被标识为对应于第一检测到的不期望的行为，而部分 2a 和 2b 可被标识为对应于应用的第二不期望的行为。相应地，修复该应用可包括利用修改或抑制不期望的行为的代码来修改或替换所标识的冒犯性部分的代码。此外，可标识对应于所标识的代码或 API 调用的修复策略以标识用于修改经冒犯的代码以及补救不期望的行为的修复技术。

[0068] 在图 12A-12E 中，示出了检测不期望的行为和补救这些不期望的行为的附加示例。例如，在图 12A 中，允许应用向外部服务器发送纬度和经度信息的示例代码片段被示出为经处理以例如利用行为分析引擎来填充 API 模板。如图 12B 中所示，可标识该应用代码的、对应于收集地理位置数据并且将该地理位置数据发送到外部服务器的行为的多个部分。根据一个示例，可例如利用对地理位置数据的发送进行屏蔽或重定向的代码来替换冒犯性的代码行以防止该应用跟踪用户位置，等等此类示例。在另一示例中，图 12C 中示出可在应用中标识控制流以及对应的应用代码。如图 12D-12E 中的示例中所示，补救特定的不期望的行为可包括删除冒犯性的代码行，等等此类示例。

[0069] 图 13 示出结合应用的一个或多个检测到的不期望的行为的补救的示例流 1300。例如，可标识与针对特定用户的应用行为的动态个性化、应用的复合行为和对应的代码片段的联系。可结合对允许用户选择特定的经标识的行为以进行补救或修改的应用进行的修复或定制来呈现用户界面。在一个示例实现方案中，可结合应用修复引擎，向该用户界面提供指示该应用修复引擎将如何（例如，哪些经标识的行为）修改该应用的用户输入。在另一示例中，应用修复引擎可将一个或多个用户界面控制插入到该应用原始的二进制文件中，从而允许用户在启动经修改的应用时动态地启用、禁用或以其他方式补救或自定义该应用的行为。例如，基于该用户的选择，可利用对应于可接受行为的代码的原始部分来代替相同代码的经修复的版本，等等此类示例。实际上，可基于用户偏好和输入，选择性地关闭或打开展示出行为的代码的片段中的每一个片段。此外，该用户界面可向用户提供保存应用设置的选项，以便对应用行为的特定子集的选择得以保存并且该选择在该应用下一次在该用户设备上启动时是可用的。

[0070] 在一些实现方案中，可提供功能以定义、启用和采用用户设备上所定义的使用模式 (mode)。从传统意义上说，诸如智能电话和平板计算机等等此类示例之类的用户设备被设计成用于支持单个用户和应用简档。然而，对于该设备的所有实际用户或者对于使用该设备的多个情景而言，单个操作简档和模式可能不是合适的。例如，用户可能期望在某个短时期内将其设备借给朋友，但是想要保留对访问该设备上的敏感应用和数据、email 应用、通讯录、日历、消息收发功能等中的一些的控制。在其他实例中，该用户可能期望允许儿童临时地使用该设备（例如，玩游戏），但是更希望将其他应用（如，web 浏览器）以及对某些

设备设置和数据的访问与该儿童隔离开。此外，用户可能期望将该设备上应用的一些子集的使用控制在特定的时间、地点和情景中。例如，在上课时间期间可能期望禁用游戏和社交联网应用，等等此类示例。

[0071] 图 14A 示出模式管理器的示例实现方案的简化框图 1400a。例如，可基于从该用户设备和外部服务收集到的情报来定义各种模式。用户可通过用户界面定义一个或多个模式，该设备上的模式管理器可例如使用分配给每一个模式的专用证书来管理对各种模式的访问。此外，如上文所述，应用监测服务或应用行为分析引擎可对于该用户设备上可用的隔离模式或高安全性模式推荐多个特定的应用。相应地，用户可定义此类模式以将对潜在地有风险的或当前正在被分析的应用的访问限制于管理用户、成人用户或其他受信任的用户，等等此类示例。

[0072] 图 14B 示出另一简化框图 1400b，其示出应用模式管理器的原理。在一些实现方案中，应用模式管理器 248 可包括各种模块和功能，例如，模式设置管理器 1405、锁定服务 1410、锁定管理器 1415、证书管理器 1420、应用访问管理器 1425、应用保护服务 1430、密码引擎 1435，等等此类示例。例如，在所示出的示例中，具有管理权限的用户可例如使用模式设置管理器来设置密码或 PIN，并且将这些证书分配到该用户所定义的多个模式中。访问管理器可利用证书管理器来验证是否接收到了允许该设备的当前用户访问针对该设备所定义的一组模式中的一个模式的有效证书。如果输入了错误的证书，则锁定管理器调用锁定服务，以通过向该用户分配受限模式或完全将该用户锁在外面来将该用户锁在一个或多个应用之外。

[0073] 在一些实现方案中，设备模式可由排除列表或包含列表组成。就应用的使用被禁止或限制的意义而言，设备模式可被定义为应用在该模式中或被允许或以某种方式受保护的应用的各个集合。在一些实例中，可定义针对指示在对应的模式下可访问的设备的应用和 / 或子系统的特定子集（即，在该模式中，其余的应用受保护或被锁定）的模式的排除列表。例如，模式可根据如下方式定义：<ModeName, Inclusion/Exclusion, Access PIN, App1, App2, App3…..App N>(< 模式名称，包含 / 排除，访问 PIN，App1，App2，App3…..App N>)。在一些实例中，每一个设备模式可受保护，并且可与特定的密码相关联。可定义允许对设备的功能和应用的全部进行访问的主模式。相应地，可提供允许访问该主模式的主密码。在该主模式之内，可向用户提供对用于管理在该设备处可用的或经定义的该组模式的管理控制台的访问。相应地，该用户可通过该管理控制台编辑或定义多个模式，以及激活或删除预定义的多个模式。示例管理控制台可允许用户从多个应用的列举中选出该用户希望指定为在任何给定的模式中受保护或可访问的那些应用。在一些情况中，可在多个不同的模式下允许或保护单个应用。

[0074] 在一些实现方案中，可将模式密码存储在经加密的存储器中。例如，可使用相同密码生成的密钥对每个模式的密码进行加密。然后，可通过利用由用户输入的密码所生成的密钥对该密码进行解密来验证经加密的密码。然后，可将该经解密的数据与用户输入的密码进行比较。基于用户所提供的该密码，可标识并认证对应的模式以允许该用户对该模式的访问。在一些实现方案中，例如在持续很久的不活动时期之后，该用户可手动锁定该设备或者该设备可锁定其自身。当尝试解锁或唤醒该设备时，可再次向用户呈现请求针对该设备的可用的和所定义的多个模式中的一个模式的密码的登录提示。

[0075] 在一些实现方案中，多个模式可以是分层的。例如，登录到较高级模式（即，提供相对更重要的访问等级的模式）的用户可能能够自由地移动到另一模式，而不需要提供那个较低级模式的证书。另一方面，已被认证到较低级模式的用户在尝试访问分层结构中比该用户先前得到认证的较低级模式更高层级处的另一模式时，不得不输入附加的证书。例如，在一个实例中，可定义四个设备模式，其中：

[0076] 模式 1 是管理员级模式；

[0077] 模式 2 访客级模式；

[0078] 模式 3 是访客级模式；以及

[0079] 模式 4 是低权限模式

[0080] 并且该分层结构定义为：模式 1>（模式 2 和模式 3）> 模式 4，其中，模式 2 是与模式 3 相同的等级，等等此类示例实现方案。

[0081] 在一些实现方案中，当某些模式是活动的时候，可更改、自定义或至少部分地限制该设备的配置。例如，特定的模式可激活或停用 GPS 功能、数据访问、通话和某些应用。此外，在一些示例中，可提供在模式时确保特定应用的数据的多个设备模式。例如，一旦已创建了新的模式并且已向应用的集合分配了对应的访问等级，则可由通过分开的加密密钥的加密来保护这些应用的数据。这可例如通过使用用于对文件和文件夹进行加密的加密文件系统来实现，等等此类示例。

[0082] 在一些实现方案中，可保全应用的可执行代码以防护在不允许访问和 / 或使用应用的行为或特征中的一个或多个的多个模式中所使用的多个应用。例如，在一个实现方案中，可将应用可执行文件存储在经加密的次级存储设备中。在一些示例中，仅仅在如果在针对尝试对设备的访问的活动的设备模式的可允许的应用列表中找到该应用时，该用户设备的操作系统加载器才可获得对可执行代码的有条件的、未经加密的访问，等等此类潜在的实现方案。

[0083] 在一些示例中，定义针对用户设备的多个设备模式可进一步导致提供在对应的多个模式中的每一个模式中所呈现的多个唯一的主屏。结果，在此类实现方案中，给定的主屏的外观可向用户指示在该设备上活动的模式以及在该模式中可用的访问权限。在一些实例中，主屏可包括在该对应的模式中可用的应用的图标，并且隐匿或隐藏在该模式中受保护的其他应用的图标，等等此类示例。

[0084] 此外，在一些实例中，可例如基于在用户设备上检测到的或加载的应用的经标识的多个行为和安全简档来自动地创建多个设备模式。例如，模式管理器可利用例如由示例应用行为分析引擎所执行的行为分析来标识展示行为的公共类别或安全简档的类别的多个应用。例如，被标识为允许访问在线资源的多个应用可被动态地分组或分配到已被定义为允许此类访问的一个或多个模式中。诸如专用于未成年用户的模式之类的其他模式可拒绝对允许用户访问因特网等此类示例的多个应用的访问。其他示例类别可包括：启用通话或移动消息收发功能的应用；使用利用敏感数据、收集潜在私人信息的子系统（如，相机、录音机、GPS 系统等）的应用，等等此类示例。在一些实现方案中，诸如年龄评级（如，7 岁 +、12 岁 +、18 岁 + 等）、用户评论或其他信息之类的关于应用的周边情报可用于对多个应用进行分类并将它们分组到各种模式中。例如，对应用的描述可包括年龄或成熟度评级以及针对该成熟度评级的原因。相应地，在一个示例中，可定义例如阻止由儿童用户访问具有更

高成熟度评级的多个应用的一个或多个模式，等等此类示例。

[0085] 其他全局的或分布式情报也可用于开发针对给定应用的信息，诸如图 19 中的简化框图 1900 中所示。例如，可从关于来自全球威胁情报 440、发行方 / 开发者信誉信息 1905、app 商店反馈和评论 1910、行为分析结果 1915 等等此类示例的应用的行为的安全信息中建立应用信息。可结合多个应用的行为评估 1915（例如，应用是否潜在地泄露数据，提供位置信息，启用 SMS 消息收发等）来使用此类信息（如，440、1905、1910 等）以将某些应用分配到诸如隔离模式或管理模式等等此类示例之类的特定的设备模式中。用户可进一步将自定义的类别或行为指定为或将预定义的类别或行为选为将多个应用分配到各自的模式的基础，而不是单独地选择多个应用以按部就班地将其包括在一个或多个模式中，等等此类示例。

[0086] 转到图 15A 中的示例，表示出用于存储与特定的模式相关联的密码信息的示例算法。图 15B 表示用于验证密码并标识对应于所输入的密码的要激活的模式的示例算法。应当领会，图 15A-15B 中的算法是仅出于说明目的而呈现出的非限制性示例，并且在其他实例中可利用其他替代性算法和实现方案。

[0087] 转到图 16 中的示例，在一些实现方案中，可将给定的用户所定义的多个模式提供给例如允许一个或多个模式以及与这些模式相关联的多条规则的应用管理服务、云服务或其他服务（如，1600）以聚集这些模式并且与其他用户共享这些模式。此外，可浏览并选择由模式共享服务 1600 维持的多个共享的模式，用于在用户设备 110、120 上的下载和利用，从而允许用户向他们自己的设备提供由其他用户创建的、并且使用模式共享服务被共享的多个模式。此外，在一些示例中，该用户可下载并安装来自模式共享服务的共享模式的定义，并向该新安装的模式分配唯一的密码来提供该共享模式。在另一些示例中，可例如通过像蓝牙、近场通信 (NFC)、WiFi 等的无线对等技术，直接地在多个设备之间共享多个模式配置，其中，一个设备从共享该模式的另一设备中获取新的模式。

[0088] 在诸如图 17 中的示例中所示那样的一些实现方案中，可基于例如由该设备自身检测到的情境信息自动地激活多个模式。在一些示例中，用户可配置（例如，在管理控制台上）用于自动地激活多个特定模式的多条规则。例如，可响应于在该用户设备处检测到特定的情境，自动地激活特定的模式。此类情境可包括例如：在所定义的地理围栏之内检测到该设备的位置或接近；检测到该设备邻近其他设备；在特定的数据网络的范围内检测到该设备；检测到该设备的用户（例如，基于该设备所收集的生物测定信息）；检测到的一天中的时间；设备电池状态；使用活动度（例如，防止特定的用户在该设备上花费太多时间等）；该设备是否正在行进中或在运动中（例如，通过该设备上的 GPS 功能、加速度计或其他功能所检测的）等等此类潜在的许多示例。

[0089] 现在转到图 18 中的示例，在一些实现方案中，可通过诸如云服务之类的远程服务来提供并配置多个模式，从而允许用户能够远程地激活 / 停用或定义模式。使用此类服务，用户能够远程地创建模式（例如，使用不同于目标移动用户设备的计算机），向目标用户设备提供一个或多个模式，并且也能从远程位置激活和停用该用户设备上的模式。此外，管理员也能够使用该服务在多个移动用户设备上提供此类模式，并且定义针对自动地激活、应用或停用给定模式等此类示例的多条规则和多个情境。

[0090] 图 20A-20D 示出多个用户界面的示例屏幕截图，这些用户界面示出移动用户设备

上的模式管理的一些示例实现方案的多个特定特征。例如，图 20A 中的屏幕截图示出用于定义新模式和模式密码的用户界面。可提供类似的用户界面以允许用户能够选择并激活在该设备上多个可用的模式中的一个，以及 / 或者提供针对所选择模式的证书。在一些实现方案中，用户设备可包括本机登录证书或本机登录管理器。模式管理器可被实现为覆盖本机登录管理器并用模式专用登录提示（例如，允许该用户设备的多模式功能的提示）来替换本机登录屏幕的应用自身。在一些实例中，由于该登录屏幕能够接受多个不同的登录代码中的一个（每一个登录代码对应于用户设备上所提供的所支持的模式（包括多个隐藏模式）），因此，用户可能无法在视觉上识别出向该用户设备提供了多个模式。

[0091] 图 20B 中的屏幕截图示出针对特定模式的主屏的视图。如此示例中所示，可指定一组受限的应用，仅可通过向更高等级的模式（例如，允许对这些受限的应用的访问的模式）提供证书并激活该更高的等级来访问这组受限的应用。此外，“我的 App”文件夹可提供对已在当前活动的模式中启用的那些应用的访问。图 20C 中的屏幕截图提供允许用户激活、编辑或创建新模式的示例管理屏幕的另一视图。此外，图 20D 中的屏幕截图示出可在模式管理器的一些实现方案中提供的用户界面，该用户界面允许用户从该设备上的多个应用的列表中指定在给定的模式中将包括或保护哪些应用，等等。应当领会，仅出于说明某些原理的目的提供上述示例，并且不应当将上述示例解释为限制性的示例。实际上，各种不同的实现方案、用户界面、程序结构、操作系统、SDK 平台和方法序列可替换上述那些示例，而不背离本说明书中所示或所述的一般原理。

[0092] 图 21A-21C 是流程图 2100a-c，其示出管理移动用户计算设备上的应用中所用的示例技术。例如，在图 21A 中的示例中，可如 2105 处所示，例如对照诸如平台 SDK 和 / 或 API 的表示之类的平台的语义表示来分析特定的应用的代码。如 2110 处所示，可标识该特定应用的一组行为。如 2115 处所示，可基于用户从经标识的一组行为中选择一个或自动地根据针对要在特定的移动计算设备上下载、安装、启动或以其他方式使用的多个应用所定义（如，由用户或管理员定义）的多条规则和 / 或多个策略，在这组行为中标识至少一个不期望的行为。

[0093] 在图 21B 中的示例中，如 2120 处所示，可标识行为，并且针对特定的应用检测一组行为（例如，根据图 21A 中的示例的原则）。然后，如 2125 处所示，可标识对应于所标识行为的特定应用的一部分代码。如 2130 处所示，可例如响应于所标识的行为是不期望的行为等的指示，对代码的经标识的部分执行补救动作以自动地补救该行为。该补救动作可导致动态地生成特定应用的“修复”版本，该“修复”版本保留该特定应用的原始功能的至少部分，而阻止不期望的功能或者将不期望的功能从经修复的版本中剥离出去。

[0094] 在图 21C 中的示例中，如 2140 处所示，可激活多个模式中的特定的一个。可针对特定的用户计算设备定义这些模式，并且这些模式可规定该计算设备及其软件的功能中的哪个子集可由具有访问多个模式中的各个模式的证书的特定用户访问。如 2145 处所示，可根据 2140 处的特定模式的激活，在 2145 处将访问限制到被安装在该用户计算设备上的一个或多个应用中。此外，在一些实现方案中，激活该特定的模式可产生要应用的该计算设备的受限的或替代的配置，该配置进而限制用户对一个或多个子系统和功能的访问，包括该用户计算设备的硬件功能、设置和数据，等等此类示例。

[0095] 虽然已按照某些实现方案以及一般相关联的方法描述了本公开，但是这些实现方

案和方法的变更和置换对本领域技术人员而言将是显而易见的。例如，本文中所描述的动作可按不同于所描述的顺序来执行，并且仍然实现期望的结果。作为一个示例，所附附图中描绘的过程并不一定要求所示的特定顺序、或顺序地来实现所期望的结果。在某些实现方案中，多任务处理和并行处理可能是有利的。此外，也可支持不同的用户界面布局和功能。其他变型落在所附权利要求的范围之内。

[0096] 可在数字电子电路中，或在计算机软件、固件、或硬件（包括在本说明书中公开的结构和它们的等效结构）中，或在上述各项中的一个或多个的组合中实现本说明书中所描述的主题和操作的各实施例。本说明书中所描述的主题的各实施例可被实现为一个或多个计算机程序，即计算机程序指令的一个或多个模块，这些计算机程序指令被编码在计算机存储介质上，用于由数据处理装置执行或控制数据处理装置的操作。替代地或附加地，这些程序指令可被编码在人工生成的传播信号上（例如，机器生成的电、光、或电磁信号），生成该人工生成的信号以编码信息，用于向合适的接收机装置进行的传输，供数据处理装置执行。计算机存储介质可以是计算机可读存储设备、计算机可读存储基板、随机或顺序存取存储器阵列或设备、或它们中的一个或多个的组合，或者计算机存储介质可被包括在上述各项中。此外，尽管计算机存储介质本身不是传播信号，但是计算机存储介质可以是被编码在人工生成的传播信号中的计算机程序指令的源或目的地。计算机存储介质也可以是包括分布式软件环境或云计算环境的一个或多个单独的物理组件或介质（例如，多个 CD、盘、或其他存储设备），或者计算机存储介质可以被包括在其中。

[0097] 包括核心网和接入网（包括无线接入网）的网络可包括一个或多个网络元件。“网络元件”可涵盖各种类型的路由器、交换机、网关、桥接器、负载平衡器、防火墙、服务器、内联服务节点、代理、处理器、模块、或可用于在网络环境中交换信息的任何其他合适的设备、组件、元件、或对象。网络元件可包括支持（或以其他方式执行）与将处理器用于本文中所概括的屏幕管理功能相关联的活动的合适的处理器、存储器元件、硬件、和 / 或软件。此外，网络元件可包括便于其操作的任何合适的组件、模块、接口、或对象。这可包括允许有效地交换数据或信息的合适的算法和通信协议。

[0098] 本说明书中所描述的各操作可被实现为由数据处理装置对储存在一个或多个计算机可读存储设备上的或从其他源接收到的数据执行的多个操作。术语“数据处理装置”、“处理器”、“处理设备”以及“计算设备”可涵盖用于处理数据的所有种类的装置、设备和机器，作为示例，包括可编程处理器、计算机、芯片上系统、或上述各项中的多个或组合。该装置可包括通用或专用逻辑电路，例如，中央处理单元 (CPU)、刀片 (blade)、专用集成电路 (ASIC) 或现场可编程门阵列 (FPGA)，等等此类合适的选项。尽管一些处理器和计算设备已描述和 / 或示出为单个处理器，但是可根据相关联的服务器的特定需求使用多个处理器。引用单个处理器的目的在于，在适用时包括多个处理器。一般而言，处理器执行指令并操纵数据以执行某些操作。除硬件之外，装置也可包括代码，其创建用于所讨论的计算机程序的执行环境，例如，构成处理器固件、协议栈、数据库管理系统、操作系统、跨平台运行时环境、虚拟机或它们中的一个或多个的组合的代码。该装置和执行环境可实现各种不同的计算模型基础结构，如 web 服务、分布式计算和网格计算基础结构。

[0099] 计算机程序（也称为程序、软件、软件应用、脚本、模块、（软件）工具、（软件）引擎、或代码）可以用任何形式的编程语言来编写，包括编译的或解释的语言、声明性或程序

性语言，并且它可以按任何形式来部署，包括作为适合在计算环境中使用的独立程序或模块、组件、子例程、对象、或其他单元。例如，计算机程序可包括有形介质上的、当被执行时可用于执行至少本文中所描述的过程和操作的计算机可读指令、固件、有线或编程硬件或上述各项的任何组合。计算机程序可以但不必对应于文件系统中的文件。程序可被存储在将其他程序或数据（例如，存储在标记语言文档中的一个或多个脚本）保持在专用于所谈论的该程序的单个文件中，或保持在多个协调的文件（例如，存储一个或多个模块、子程序、或代码的多个部分的多个文件）中的文件的一部分中。可将计算机程序部署为在一个计算机上执行，或者在位于一处或横跨多个地点分布并通过通信网络互连的多个计算机上执行。

[0100] 程序可被实现为通过各种对象、方法、或其他过程实现各种特征和功能的多个单个的模块，或在适当时可反而包括数个子模块、第三方服务、组件、库，诸如此类。反之，在适当时，各组件的特征和功能可被组合到单个组件中。在某些情况下，程序和软件系统可被实现为复合的经主管的应用。例如，该复合应用的各部分可被实现为企业 Java Bean(EJB)，或者设计时组件可具有将运行时实现生成到不同的平台中的能力，例如，J2EE(Java 2 平台，企业版)、ABAP(高级业务应用编程)对象或微软的 .NET，等等。此外，应用可表示经由网络(例如，通过因特网)被访问并执行的、基于 web 的应用。此外，可远程地存储、引用或执行与特定的受主管的应用或服务相关联的一个或多个过程。例如，特定的受主管的应用或服务的一部分可以是与该应用相关联的、被远程地调用的 web 服务，而该受主管的应用的另一部分可以是被打包以供在远程客户机处进行处理的接口对象或代理。此外，经主管的应用和软件服务中的任一个或全部可以是另一软件模块或企业应用（未示出）的子代 (child) 或子模块，而不背离本公开的范围。另外，受主管的应用的各部分可由在主管该应用的服务器处直接工作的用户以及远程地在客户机处工作的用户来执行。

[0101] 本说明书中所描述的过程和逻辑流可由一个或多个可编程处理器执行，这些可编程处理器执行一个或多个计算机程序以通过操作输入数据并生成结果来执行多个动作。这些过程和逻辑流程也可由专用逻辑电路（例如，FPGA(现场可编程门阵列）或 ASIC(专用集成电路)）来执行，并且装置也可被实现为专用逻辑电路（例如，FPGA(现场可编程门阵列）或 ASIC(专用集成电路))。

[0102] 作为示例，适于执行计算机程序的处理器包括通用微处理器和专用微处理器两者，以及任何类型的数字计算机中的任何一个或多个处理器。一般而言，处理器将从只读存储器或随机存取存储器或两者中接收指令和数据。计算机的基本元件是用于根据指令执行多个动作的处理器以及用于存储指令和数据的一个或多个存储设备。一般而言，计算机还将包括用于存储数据的一个或多个大容量存储设备（例如，磁盘、磁光盘或光盘），并且 / 或者计算机将被操作地耦合以从或向用于存储数据的一个或多个大容量存储设备（例如，磁盘、磁光盘或光盘）接收或传输数据。然而，计算机不需要具有此类设备。此外，计算机可以被嵌入到另一设备中，例如，移动电话、个人数字助理 (PDA)、平板计算机、移动音频或视频播放器、游戏控制台、全球定位系统 (GPS) 接收器、或便携式存储设备（如，通用串行总线 (USB) 闪存驱动器）等等。适用于存储计算机程序指令和数据的设备包括所有形式的非易失性存储器、介质和存储器设备，通过示例的方式，其包括，半导体储存设备（例如，EPROM、EEPROM) 和闪存设备；磁盘（例如，内部硬盘或可移动盘）；磁光盘；以及 CD-ROM 和 DVD-ROM

盘。处理器和存储器可由专用逻辑电路补充和/或被结合在专用逻辑电路中。

[0103] 为了提供与用户的交互，本说明书中所描述的主题的各实施例可被实现在具有用于向用户显示信息的显示设备（如，CRT（阴极射线管）或LCD（液晶显示）监视器）以及用户可通过其向计算机提供输入的键盘和指点设备（如，鼠标或跟踪球）的计算机上。其他类型的设备可用于提供与用户的交互；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按任何形式接收，包括声音、言语或触觉输入。此外，计算机可通过向用户所使用的设备（包括远程设备）发送文档和从该设备接收文档与该用户交互。

[0104] 可在计算系统中实现本说明书中所描述的主题的各实施例，该计算系统包括后端组件，如作为数据服务器的后端组件；或者包括中间件组件，如应用服务器；或者包括前端组件，如具有图形用户界面或Web浏览器的客户机计算机，客户可通过该图形用户界面或Web浏览器与本说明书中所描述的主题的实现进行交互；或者包括这些后端、中间件或前端组件中的一个或多个组件的任意组合。系统的组件可通过任何形式或任何介质的数字数据通信（例如，通信网络）来互连。通信网络的示例包括用于促进系统中的各计算组件之间的通信的任何内部或外部的网络、多个网络、子网、或上述各项的组合。网络可在多个网络地址之间传递例如网际协议（IP）分组、帧中继帧、异步传输模式（ATM）单元、语音、视频、数据、以及其他合适的信息。网络也可包括一个或多个局域网（LAN）、无线电接入网（RAN）、城域网（MAN）、广域网（WAN）、因特网的全部或部分、对等网络（例如，自组织对等网络）、和/或一个或多个位置处的一个或多个任何其他通信系统。

[0105] 计算系统可包括客户机和服务器。客户机和服务器一般彼此远离，并且通常通过通信网络进行交互。客户机和服务器的关系凭借在各自的计算机上运行并且具有相对于彼此的客户机-服务器关系的计算机程序来产生。在一些实施例中，服务器将数据（如，HTML页面）发送到客户机设备（如，为了向与该客户机设备交互的用户显示数据并且从该用户处接收用户输入）。可从服务器处的客户机设备中接收在客户机设备处生成的数据（如，用户交互的结果）。

[0106] 虽然本说明书包含许多具体实现细节，但这些具体实现细节不应当被解释为对任何发明或可主张权利的范围的限制，而是被解释为针对特定发明的特定实施例的特征的描述。在本说明书中多个单独的实施例的情境中所描述的某些特征也可被组合地实现在单个实施例中。反之，在单个实施例的情境中所描述的各种特征也可以单独地或以任何合适的子组合的形式被实现在多个实施例中。此外，虽然多个特征在上文中可能被描述为以某些组合的方式起作用，并且甚至最初是如此要求保护的，但是，来自所要求保护的组合的一个或多个特征在一些情况下可从该组合被删去，并且所要求保护的组合可以针对子组合、或子组合的变型。

[0107] 以下示例涉及根据本说明书所述的各实施例。一个或多个实施例可提供装置、系统、机器可读介质和方法，其用于：对照特定平台的软件开发包的语义模型分析特定应用的代码；基于对该代码的分析，标识该特定应用的一组行为；以及标识一组行为中的一个或多个行为是不期望的行为。语义模型可将潜在的应用行为与该特定平台的一个或多个API相关联。

[0108] 在一个示例中，标识一组行为中的一个或多个行为是不期望的行为包括确定这一

个或多个行为违反了一条或多条规则。这些规则可与特定的用户相关联。

[0109] 在一个示例中，用户输入将一组行为中的一个或多个行为标识为不期望的行为。可结合显示所标识的这组行为的人类可读描述的用户界面来接收该用户输入。

[0110] 在一个示例中，可至少部分地基于语义模型将特定应用的代码反汇编成控制流，并且生成针对该特定应用的应用逻辑的模型。该应用逻辑的模型可进一步至少部分地基于周边应用知识。

[0111] 在一个示例中，可基于一组行为中的一个或多个行为是不期望的行为的指示来执行补救动作。

[0112] 在一个示例中，结合在特定的用户设备上实现特定应用的尝试，分析该特定应用的代码。

[0113] 一个或多个实施例可提供装置、系统、机器可读介质和方法，其用于：标识检测到的被包括在特定应用中的一组行为中的特定行为；标识该特定应用的、对应于该特定行为的一部分代码；以及对这部分代码执行补救动作以补救该行为，并且生成该特定应用的经修复的版本。

[0114] 在一个示例中，该补救动作保留除特定行为之外的特定应用的其他行为。

[0115] 在一个示例中，该补救动作包括删除这部分代码。

[0116] 在一个示例中，该补救动作包括重写这部分代码。

[0117] 在一个示例中，该补救动作包括将附加的代码添加到该应用中以使特定行为无效。

[0118] 在一个示例中，从策略中标识该补救动作，该策略标识被确定为适用于补救特定行为的补救型式。

[0119] 在一个示例中，该补救动作包括插入应用逻辑，该应用逻辑允许用户当在用户设备上启动经修复的应用时，能够选择性地启用特定行为的经修复的版本。可进一步允许该用户选择性地启用特定行为的未经修复的版本来代替该经修复的版本。

[0120] 在一个示例中，可通过对特定应用的代码的分析检测特定应用的这组行为。

[0121] 在一个示例中，通过用户请求来触发该补救动作。

[0122] 一个或多个实施例可提供装置、系统、机器可读介质和方法，其用于：激活针对特定用户设备所定义的多个模式中的特定的一个模式；以及根据所激活的该特定的模式，限制对安装在特定用户设备上的一个或多个应用的访问。当激活多个模式中的另一模式时，该受限的应用可以是可被访问的。

[0123] 在一个示例中，响应于由特定用户设备的用户输入的特定通行码，激活特定的模式，其中，多个模式中的每一个模式与对应的通行码相关联。激活特定模式可包括：基于特定通行码的输入，从多个模式中标识特定的模式；并且基于通行码的输入，认证对特定模式的访问。

[0124] 在一个示例中，多个模式中的一个或多个模式是用户定义的模式。

[0125] 在一个示例中，可基于特定模式的激活将替代的设备配置应用于特定的用户设备。该替代的设备配置可限制对特定用户设备的一个或多个子系统的访问。

[0126] 在一个示例中，多个模式中的一个模式是允许修改这多个模式的管理模式。

[0127] 在一个示例中，多个模式中的至少一个模式是可远离特定的用户设备、从模式共

享服务中下载的模式的实例。

[0128] 在一个示例中，至少部分地基于检测到使用特定用户设备的功能的特定情境，自动地激活特定模式。

[0129] 在一个示例中，基于针对特定模式所定义的规则，限制应用中的至少特定的一个应用。

[0130] 在一个示例中，所定义的规则与特定应用的检测到的行为有关。

[0131] 在一个示例中，多个模式包括被指定为针对等待行为分析或补救的应用的隔离模式的模式。

[0132] 在一个示例中，响应于在远离特定用户设备的设备处接收到的用户命令来激活特定模式。

[0133] 类似地，虽然在附图中以特定顺序描绘了多个操作，但这不应当被理解为：要求按所示的特定顺序或顺序地执行此类操作，或者要求要执行所有示出的操作才能达成期望的结果。在某些情况下，多任务处理和并行处理可能是有利的。此外，将上文所描述的各实施例中的各种系统组件分开不应当被理解为在所有实施例中都要求这样分开，并且应当理解，所描述的程序组件和系统一般可以一起被集成在单个软件产品中或被封装进多个软件产品中。

[0134] 因而，已描述了本主题的特定实施例。其他实施例落在所附权利要求的范围之内。在一些情况下，权利要求中叙述的动作可按不同顺序来执行，并且仍实现期望的结果。另外，在所附附图中描绘的多个过程不一定要求所示出的特定顺序或要求顺序地来实现期望的结果。

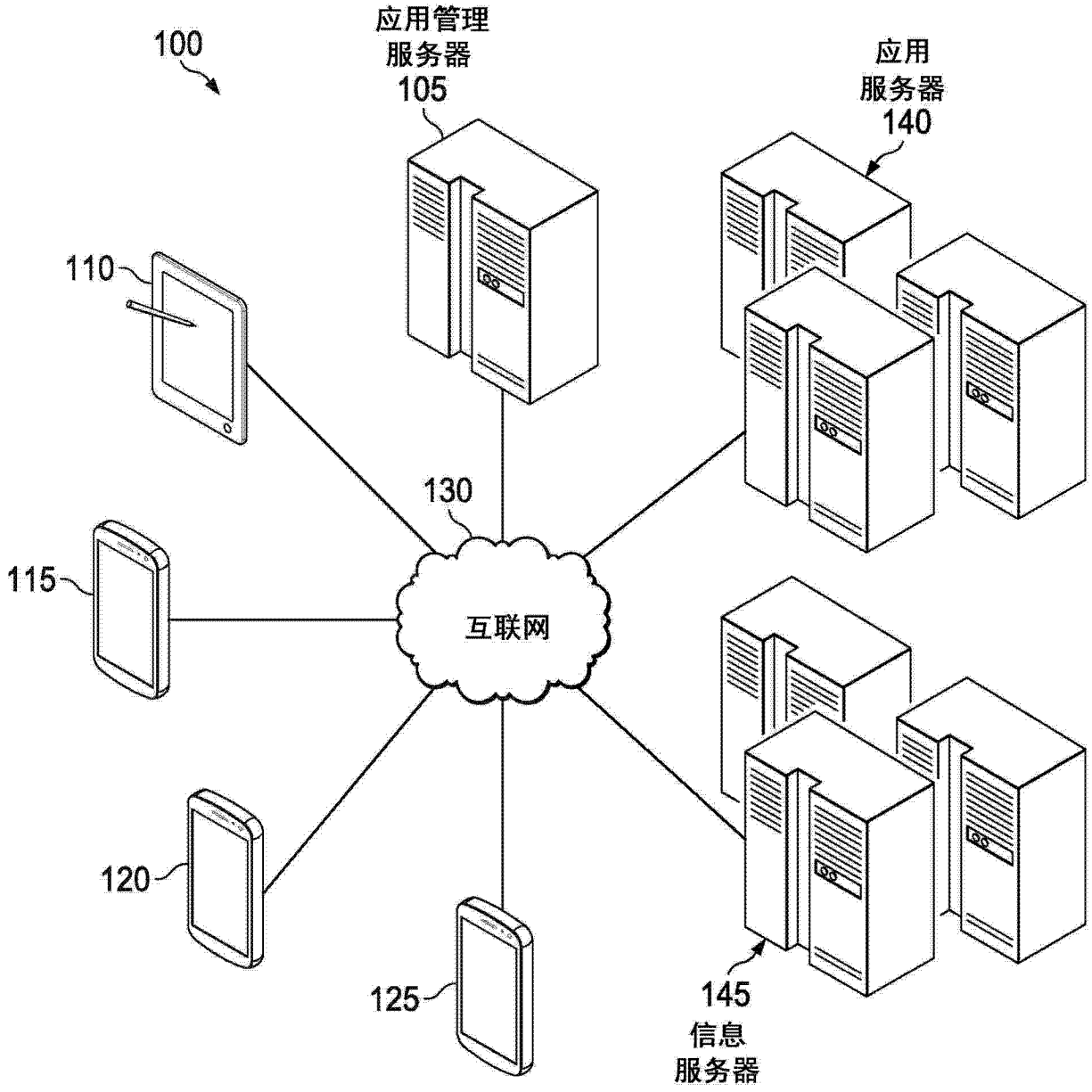


图 1

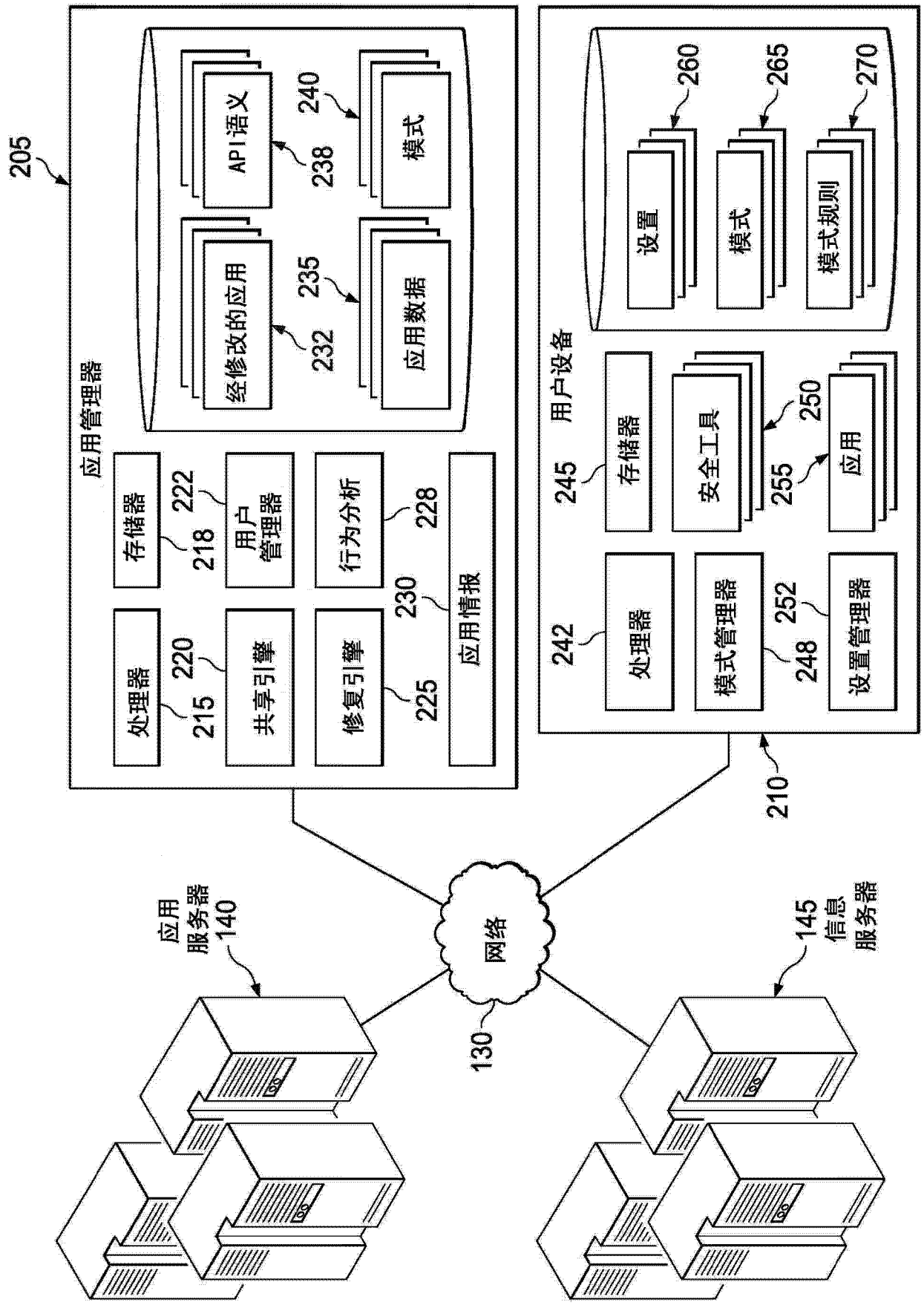

205
应用管理器
存储器
218
222
用户管理器
行为分析
228
处理器
215
220
共享引擎
修复引擎
225
230
应用情报
经修改的应用
232
235
应用数据
API语义
238
240
模式
用户设备
245
存储器
安全工具
255
250
应用
242
处理器
模式管理器
248
252
设置管理器
260
设置
265
模式
270
模式规则
210
网络
130
应用服务器
140
145
信息服务器

图 2

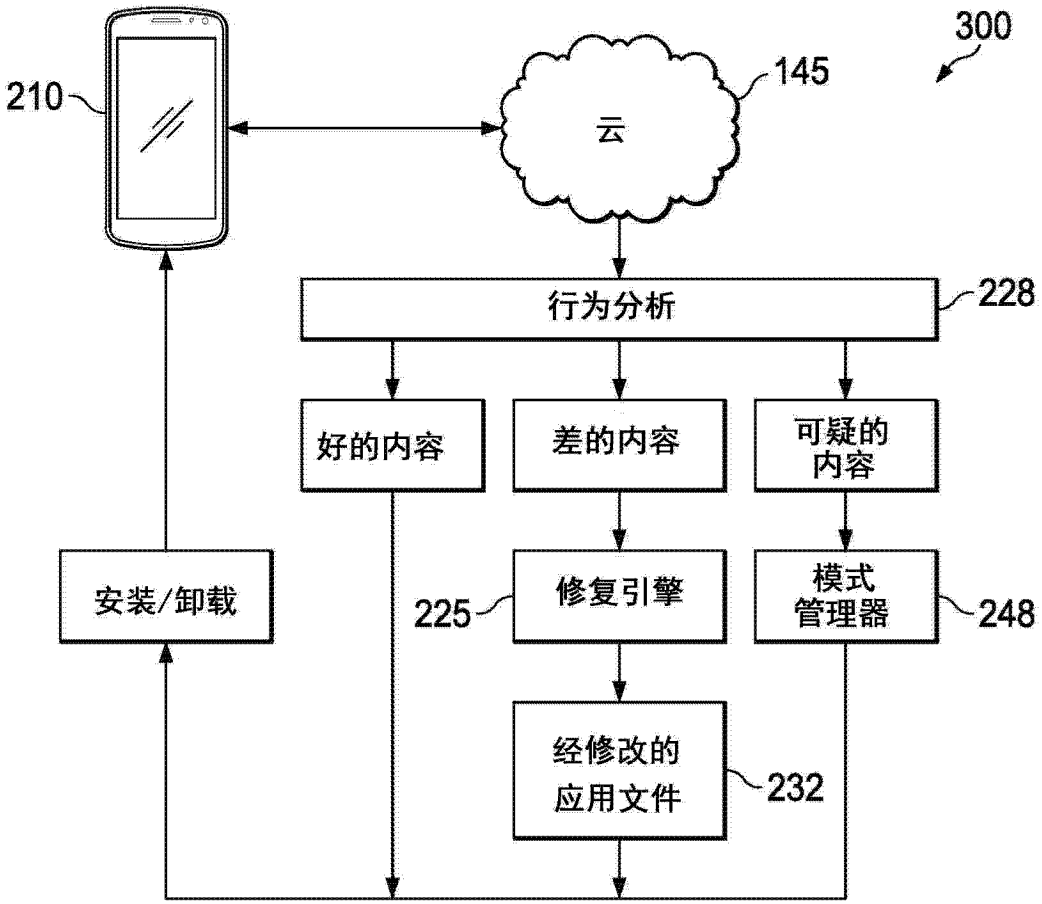

300
210
145
云
行为分析
228
好的内容
差的内容
可疑的
内容
安装/卸载
225
修复引擎
模式
管理器
248
经修改的
应用文件
232

图 3

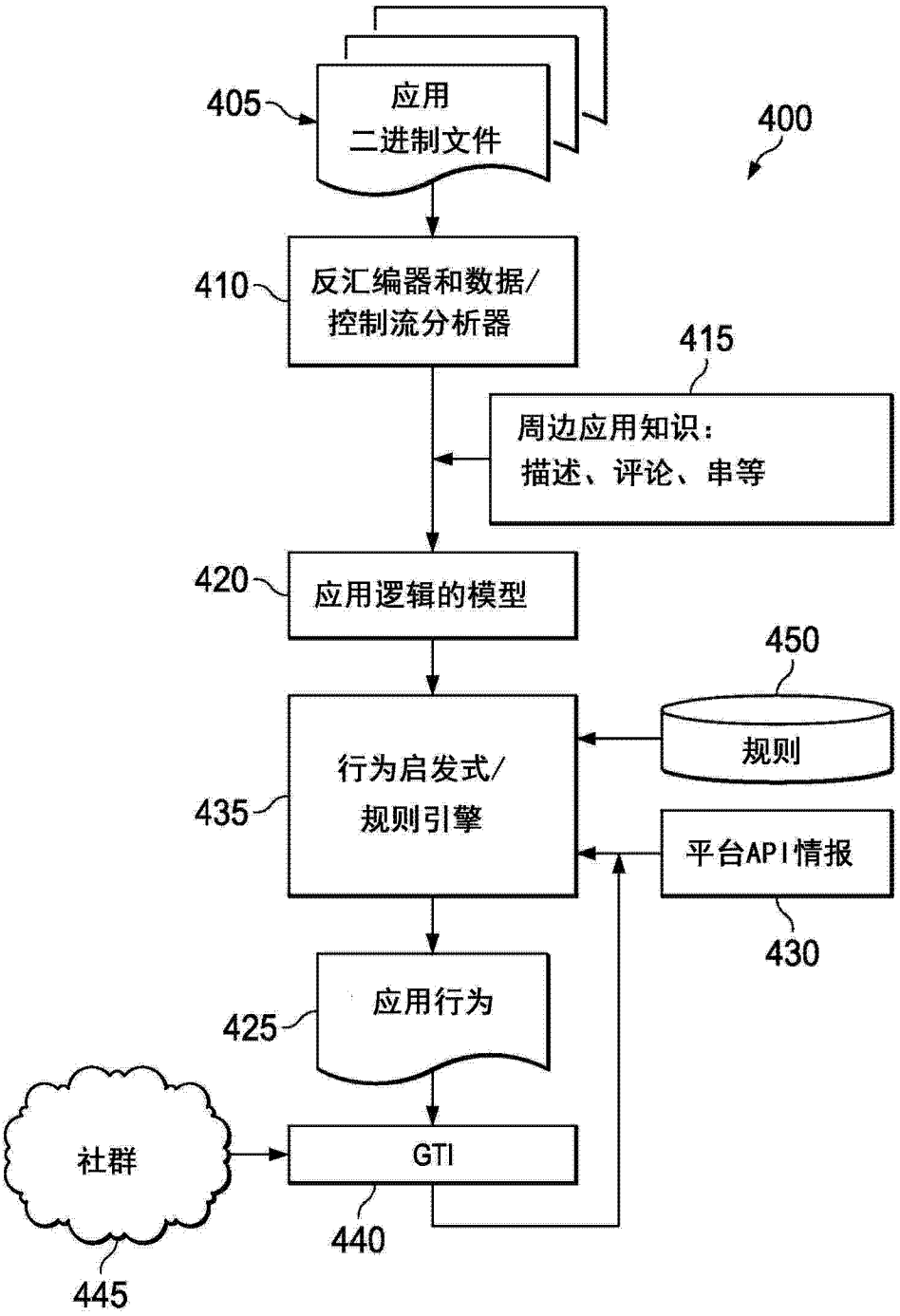

405
应用
二进制文件
400
410
反汇编器和数据/
控制流分析器
415
周边应用知识：
描述、评论、串等
420
应用逻辑的模型
450
规则
行为启发式/
规则引擎
435
平台API情报
430
应用行为
425
社群
GTI
440
445

图 4

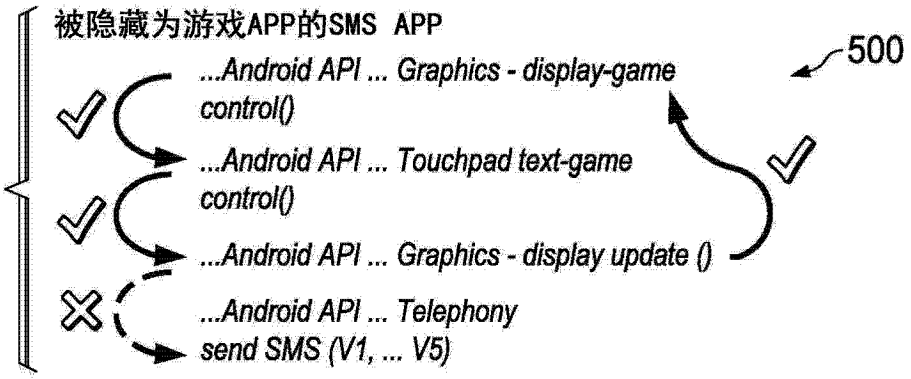

被隐藏为游戏APP的SMS APP
500
...Android API ... Graphics - display-game control()
...Android API ... Touchpad text-game control()
...Android API ... Graphics - display update ()
...Android API ... Telephony send SMS (V1, ... V5)

图 5A

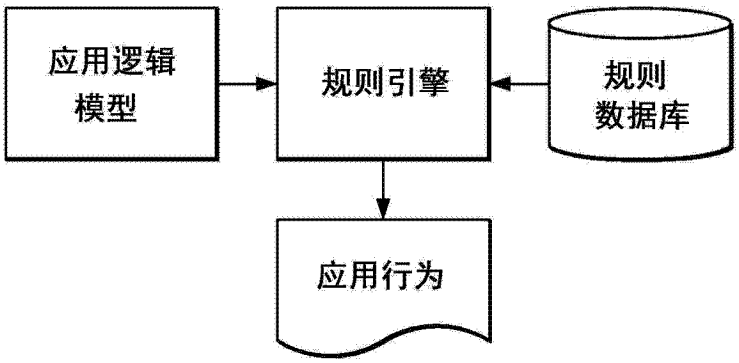

应用逻辑模型
规则引擎
规则数据库
应用行为

图 7

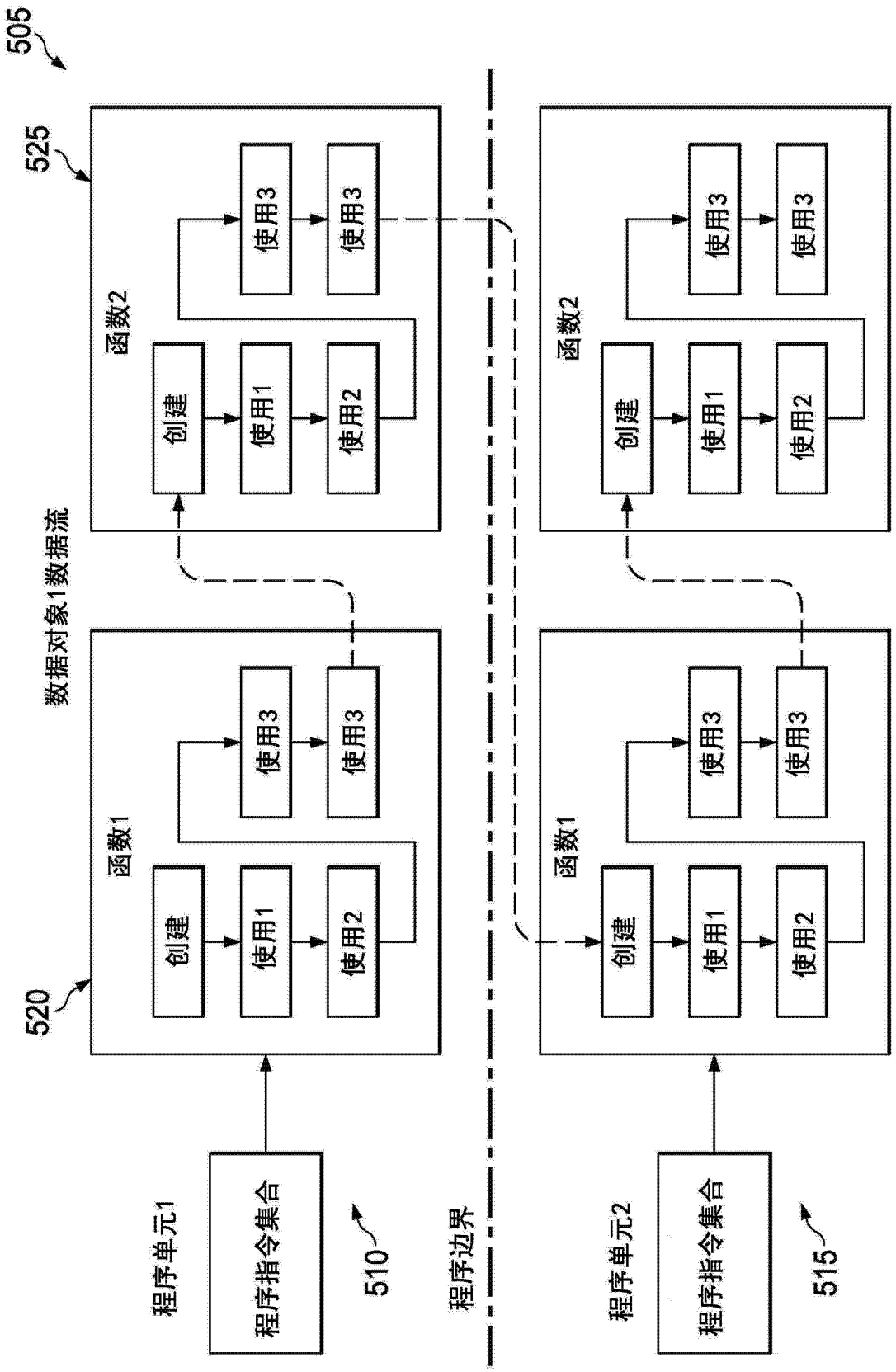

505
525
函数2
创建
使用1
使用2
使用3
使用3
数据对象1数据流
520
函数1
创建
使用1
使用2
使用3
使用3
程序单元1
程序指令集合
510
程序边界
函数2
创建
使用1
使用2
使用3
使用3
函数1
创建
使用1
使用2
使用3
使用3
程序单元2
程序指令集合
515

图 5B

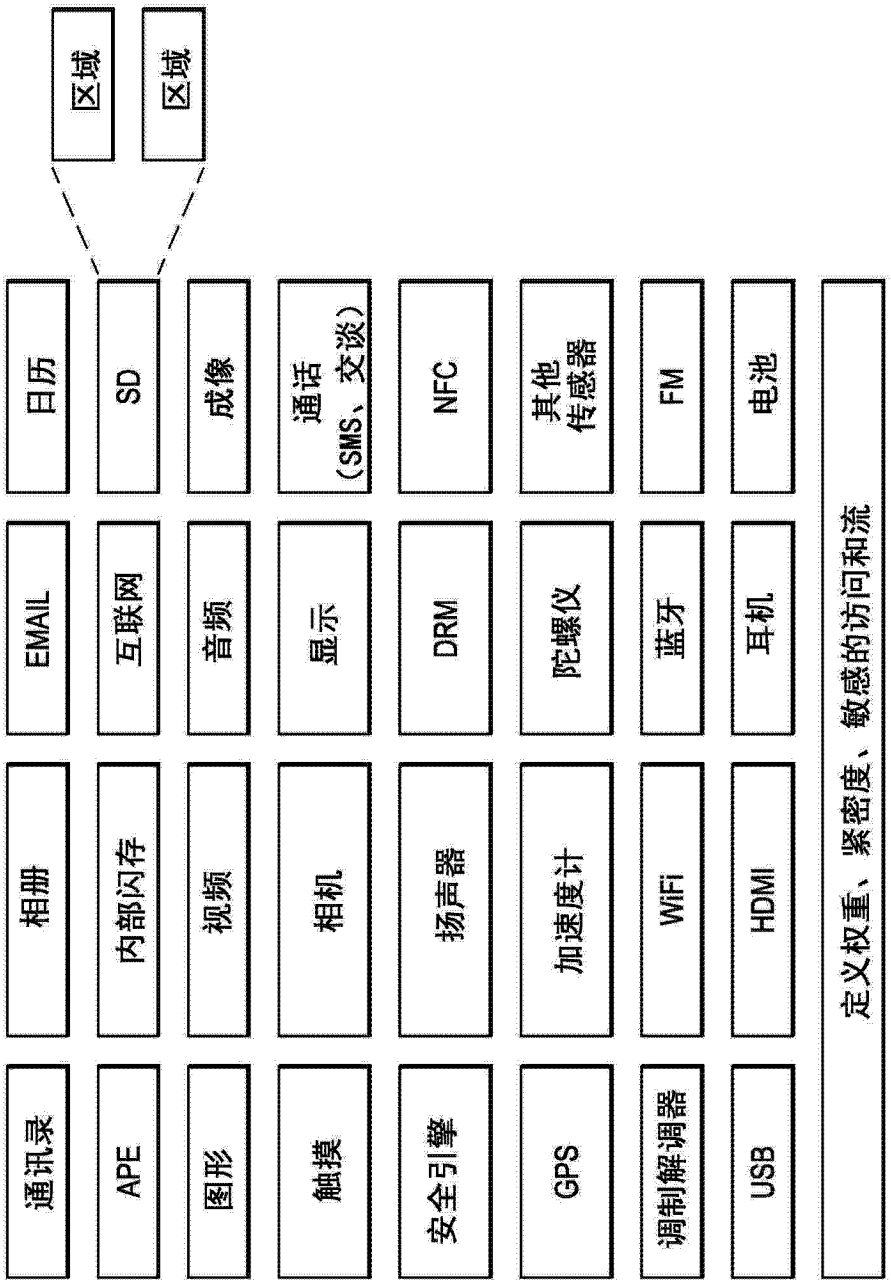

区域
区域
日历
SD
成像
通话（SMS、交谈）
NFC
其他传感器
FM
电池
EMAIL
互联网
音频
显示
DRM
陀螺仪
蓝牙
耳机
相册
内部闪存
视频
相机
扬声器
加速度计
WiFi
HDMI
通讯录
APE
图形
触摸
安全引擎
GPS
调制解调器
USB
定义权重、紧密度、敏感的访问和流

图 6

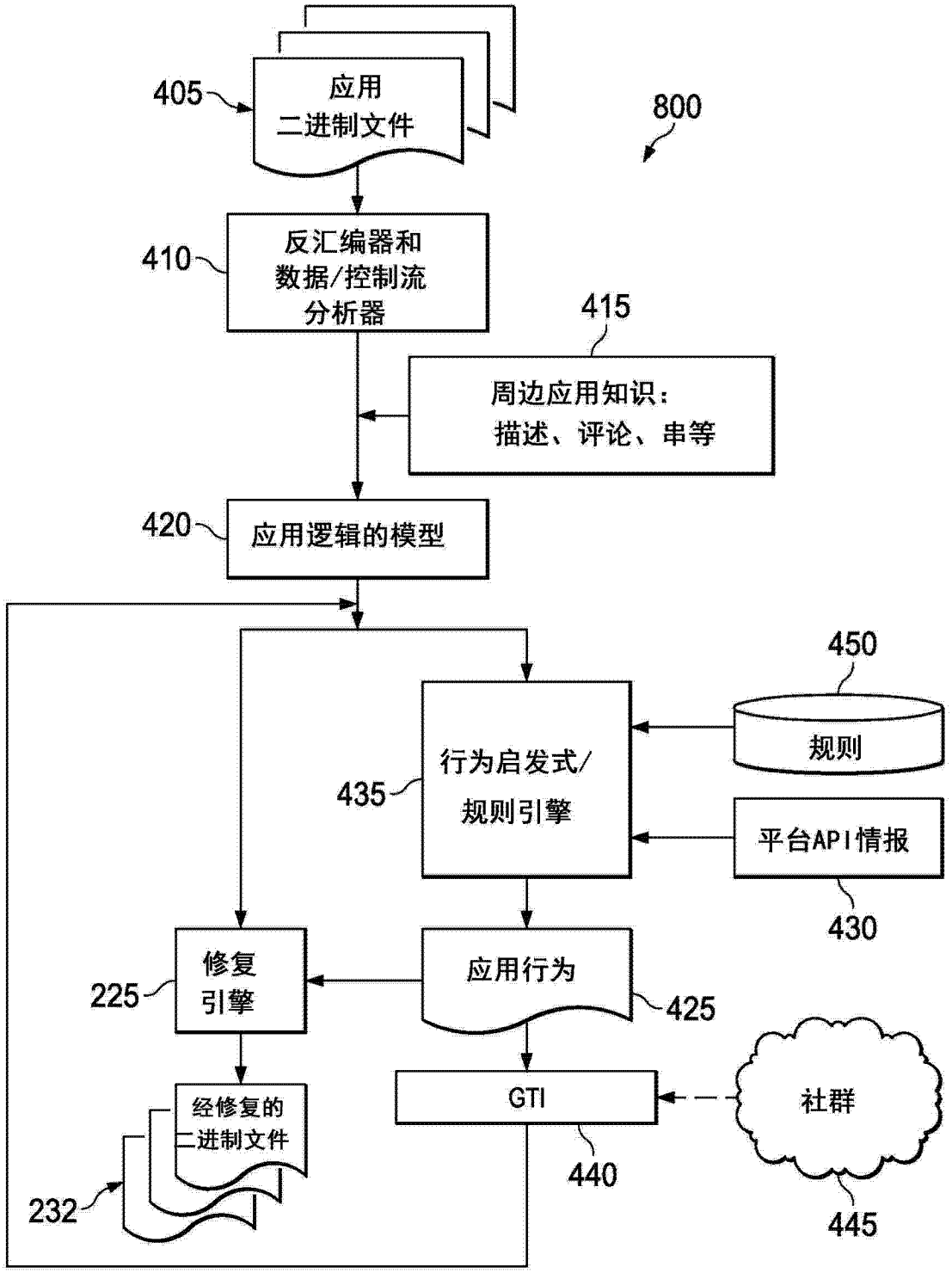

405
应用
二进制文件
800
410
反汇编器和
数据/控制流
分析器
415
周边应用知识：
描述、评论、串等
420
应用逻辑的模型
450
规则
435
行为启发式/
规则引擎
平台API情报
430
225
修复
引擎
应用行为
425
GTI
社群
经修复的
二进制文件
232
440
445

图 8

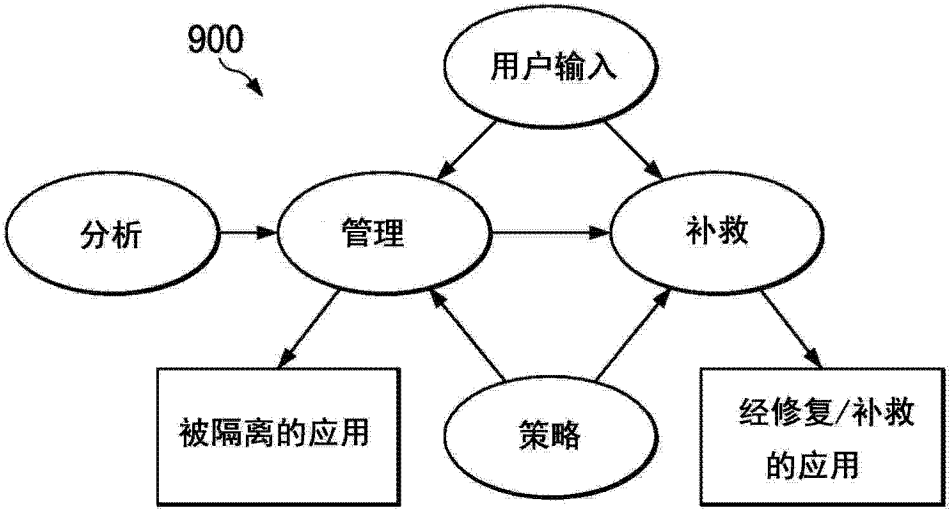

900
用户输入
分析
管理
补救
被隔离的应用
策略
经修复/补救
的应用

图 9

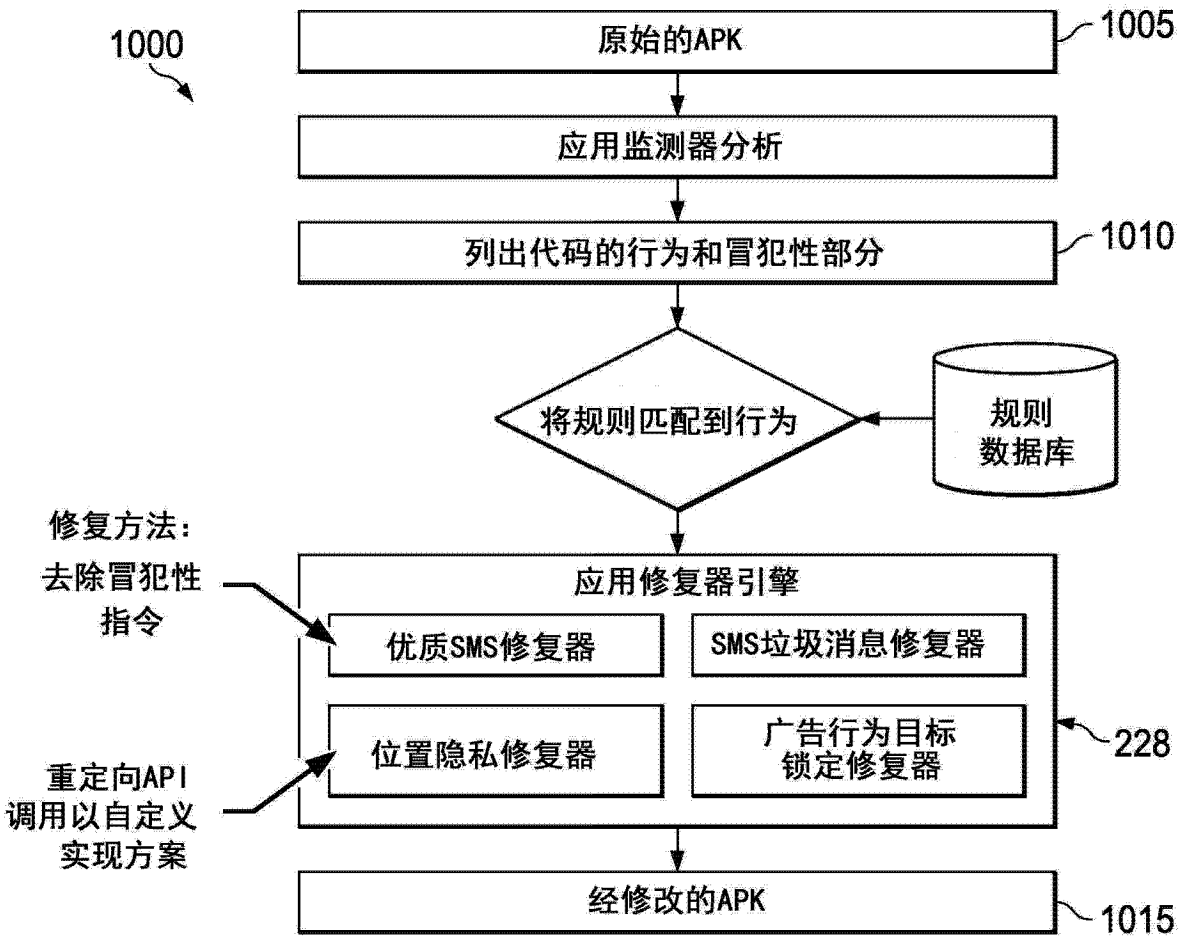

1000
原始的APK
1005
应用监测器分析
列出代码的行为和冒犯性部分
1010
将规则匹配到行为
规则
数据库
修复方法：
去除冒犯性
指令
应用修复器引擎
优质SMS修复器
SMS垃圾消息修复器
位置隐私修复器
广告行为目标
锁定修复器
228
重定向API
调用以自定义
实现方案
经修改的APK
1015

图 10

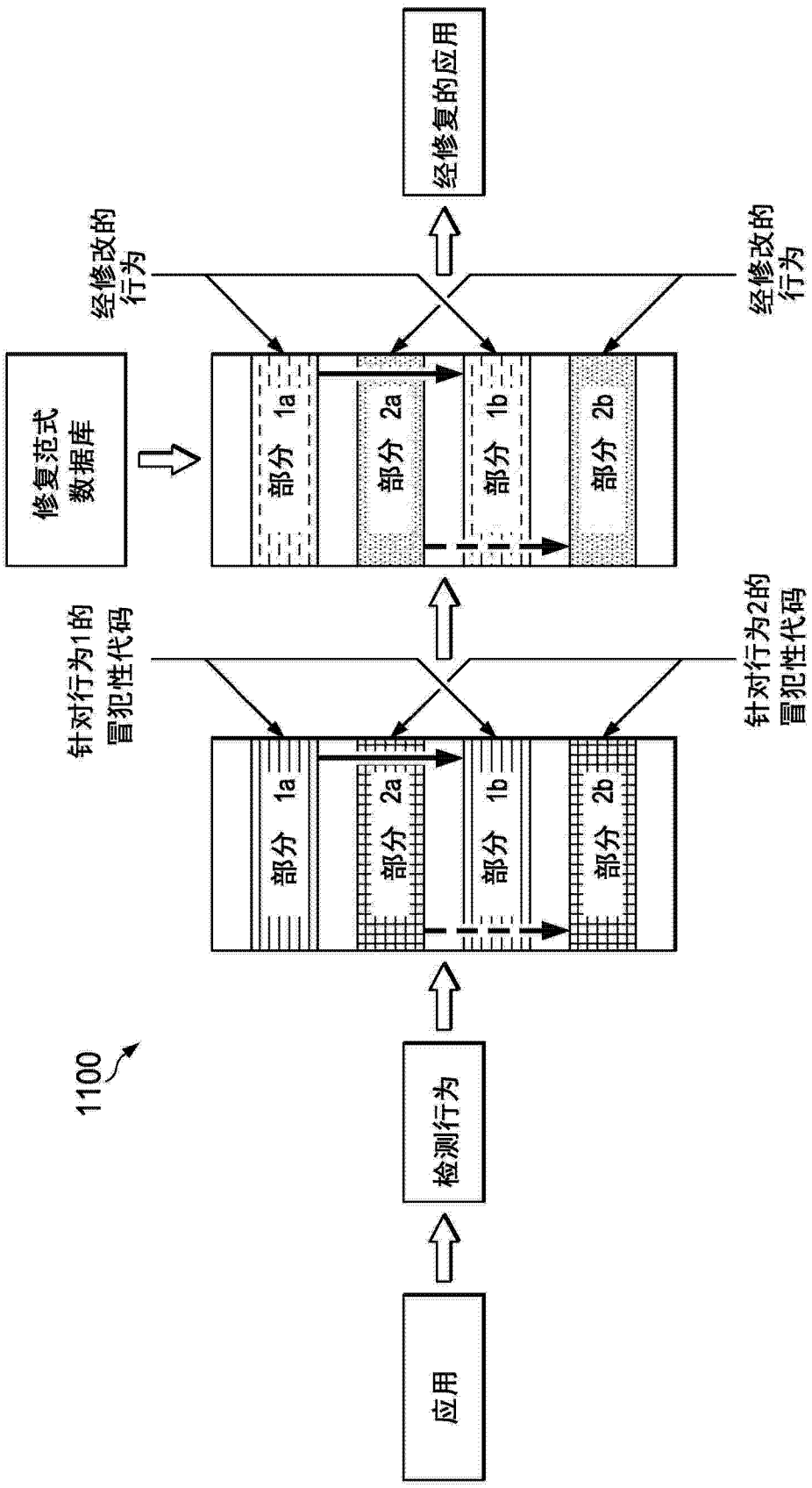

经修复的应用
经修改的行为
经修改的行为
修复范式数据库
部分 1a
部分 2a
部分 1b
部分 2b
针对行为1的冒犯性代码
针对行为2的冒犯性代码
部分 1a
部分 2a
部分 1b
部分 2b
1100
检测行为
应用

图 11

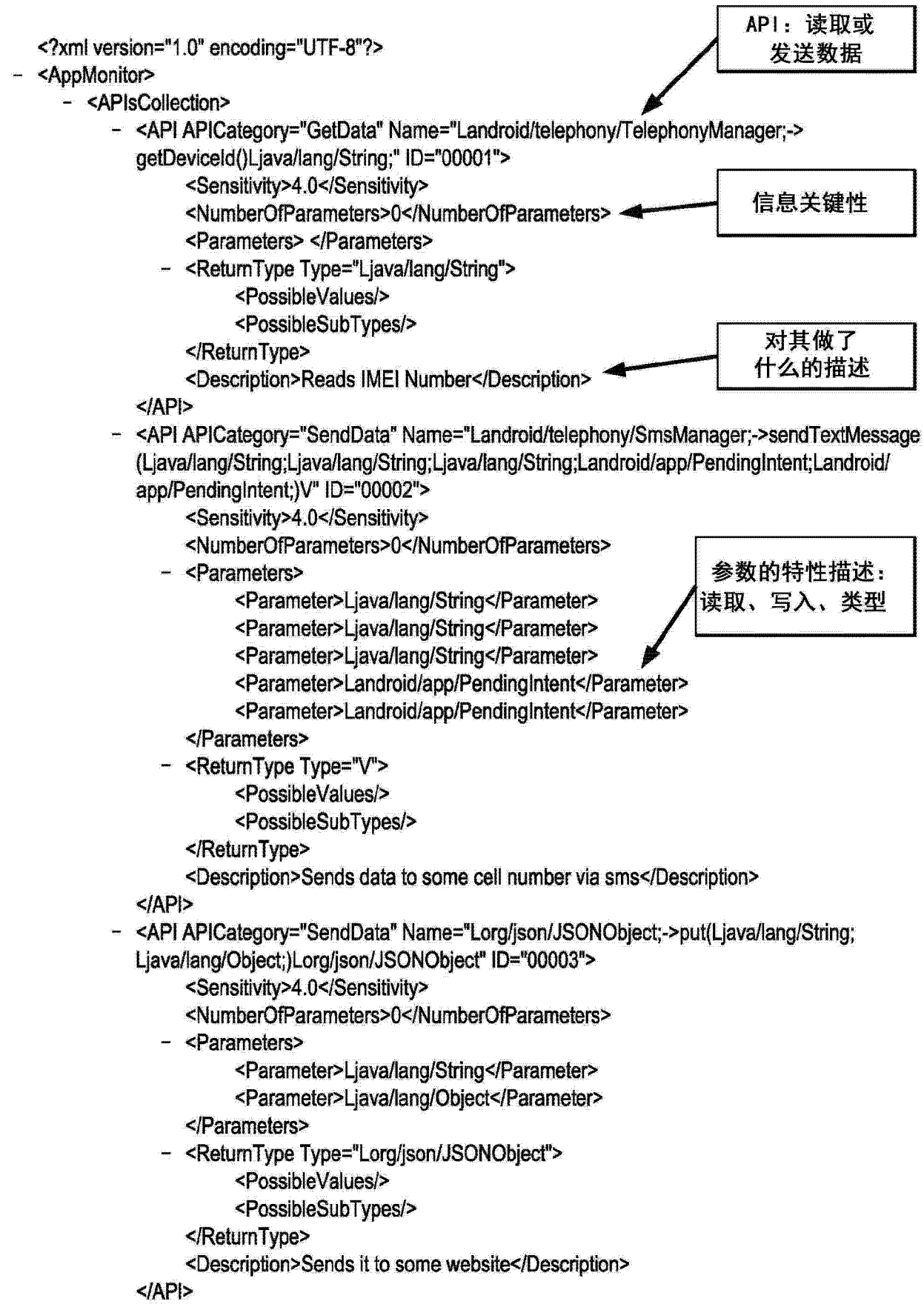

<?xml version="1.0" encoding="UTF-8"?>
- <AppMonitor>
- <APIsCollection>
- <API APICategory="GetData" Name="Landroid/telephony/TelephonyManager;->getDeviceId()Ljava/lang/String;" ID="00001">
<Sensitivity>4.0</Sensitivity>
<NumberOfParameters>0</NumberOfParameters>
<Parameters> </Parameters>
- <ReturnType Type="Ljava/lang/String">
<PossibleValues/>
<PossibleSubTypes/>
</ReturnType>
<Description>Reads IMEI Number</Description>
</API>
- <API APICategory="SendData" Name="Landroid/telephony/SmsManager;->sendTextMessage(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;Landroid/app/PendingIntent;Landroid/app/PendingIntent;)V" ID="00002">
<Sensitivity>4.0</Sensitivity>
<NumberOfParameters>0</NumberOfParameters>
- <Parameters>
<Parameter>Ljava/lang/String</Parameter>
<Parameter>Ljava/lang/String</Parameter>
<Parameter>Ljava/lang/String</Parameter>
<Parameter>Landroid/app/PendingIntent</Parameter>
<Parameter>Landroid/app/PendingIntent</Parameter>
</Parameters>
- <ReturnType Type="V">
<PossibleValues/>
<PossibleSubTypes/>
</ReturnType>
<Description>Sends data to some cell number via sms</Description>
</API>
- <API APICategory="SendData" Name="Lorg/json/JSONObject;->put(Ljava/lang/String;Ljava/lang/Object;)Lorg/json/JSONObject" ID="00003">
<Sensitivity>4.0</Sensitivity>
<NumberOfParameters>0</NumberOfParameters>
- <Parameters>
<Parameter>Ljava/lang/String</Parameter>
<Parameter>Ljava/lang/Object</Parameter>
</Parameters>
- <ReturnType Type="Lorg/json/JSONObject">
<PossibleValues/>
<PossibleSubTypes/>
</ReturnType>
<Description>Sends it to some website</Description>
</API>
API：读取或发送数据
信息关键性
对其做了什么的描述
参数的特性描述：读取、写入、类型

图 12A

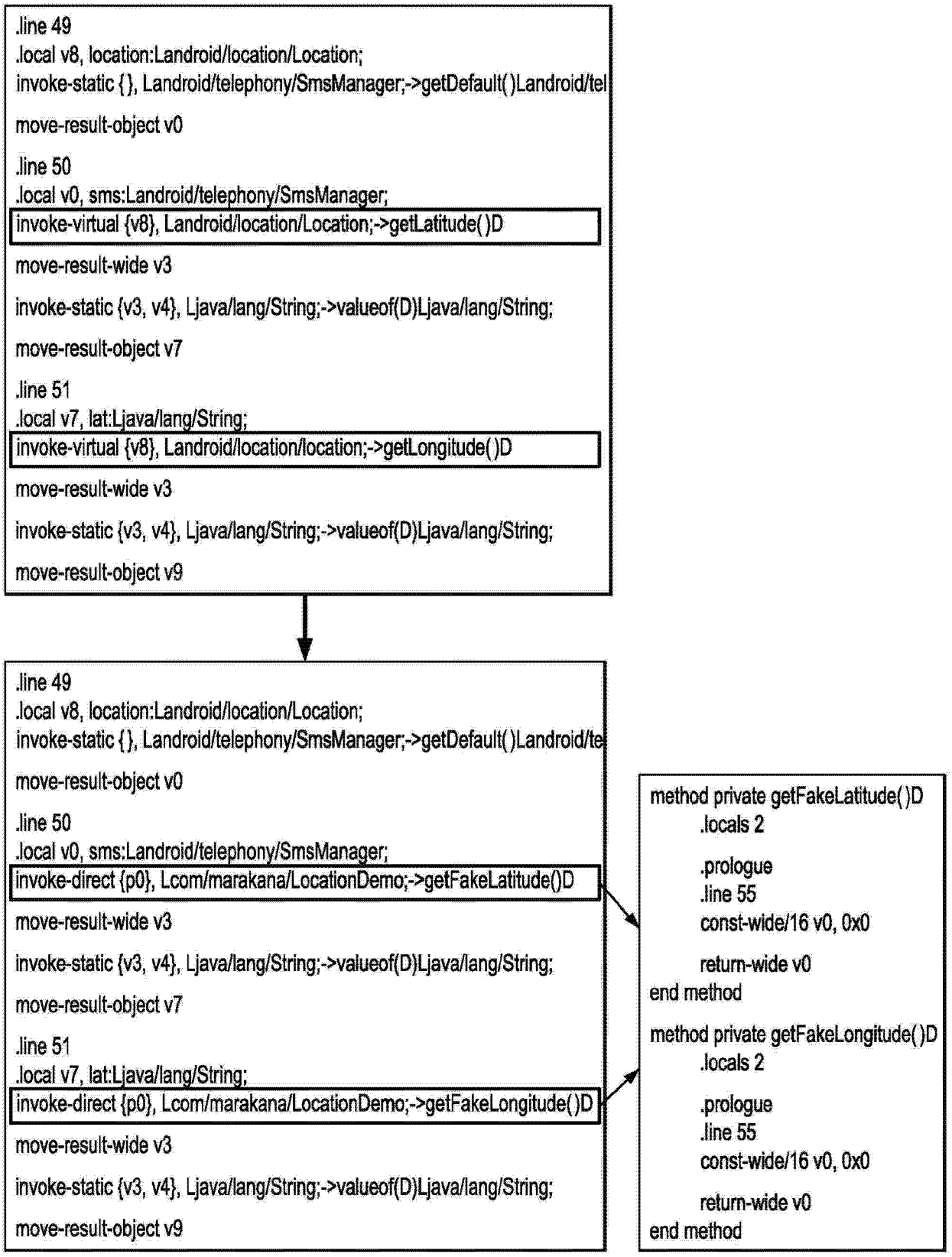

.line 49
.local v8, location:Landroid/location/Location;
invoke-static {}, Landroid/telephony/SmsManager;->getDefault()Landroid/tel
move-result-object v0
.line 50
.local v0, sms:Landroid/telephony/SmsManager;
invoke-virtual {v8}, Landroid/location/Location;->getLatitude()D
move-result-wide v3
invoke-static {v3, v4}, Ljava/lang/String;->valueof(D)Ljava/lang/String;
move-result-object v7
.line 51
.local v7, lat:Ljava/lang/String;
invoke-virtual {v8}, Landroid/location/location;->getLongitude()D
move-result-wide v3
invoke-static {v3, v4}, Ljava/lang/String;->valueof(D)Ljava/lang/String;
move-result-object v9
.line 49
.local v8, location:Landroid/location/Location;
invoke-static {}, Landroid/telephony/SmsManager;->getDefault()Landroid/te
move-result-object v0
.line 50
.local v0, sms:Landroid/telephony/SmsManager;
invoke-direct {p0}, Lcom/marakana/LocationDemo;->getFakeLatitude()D
move-result-wide v3
invoke-static {v3, v4}, Ljava/lang/String;->valueof(D)Ljava/lang/String;
move-result-object v7
.line 51
.local v7, lat:Ljava/lang/String;
invoke-direct {p0}, Lcom/marakana/LocationDemo;->getFakeLongitude()D
move-result-wide v3
invoke-static {v3, v4}, Ljava/lang/String;->valueof(D)Ljava/lang/String;
move-result-object v9
method private getFakeLatitude()D
.locals 2
.prologue
.line 55
const-wide/16 v0, 0x0
return-wide v0
end method
method private getFakeLongitude()D
.locals 2
.prologue
.line 55
const-wide/16 v0, 0x0
return-wide v0
end method

图 12B

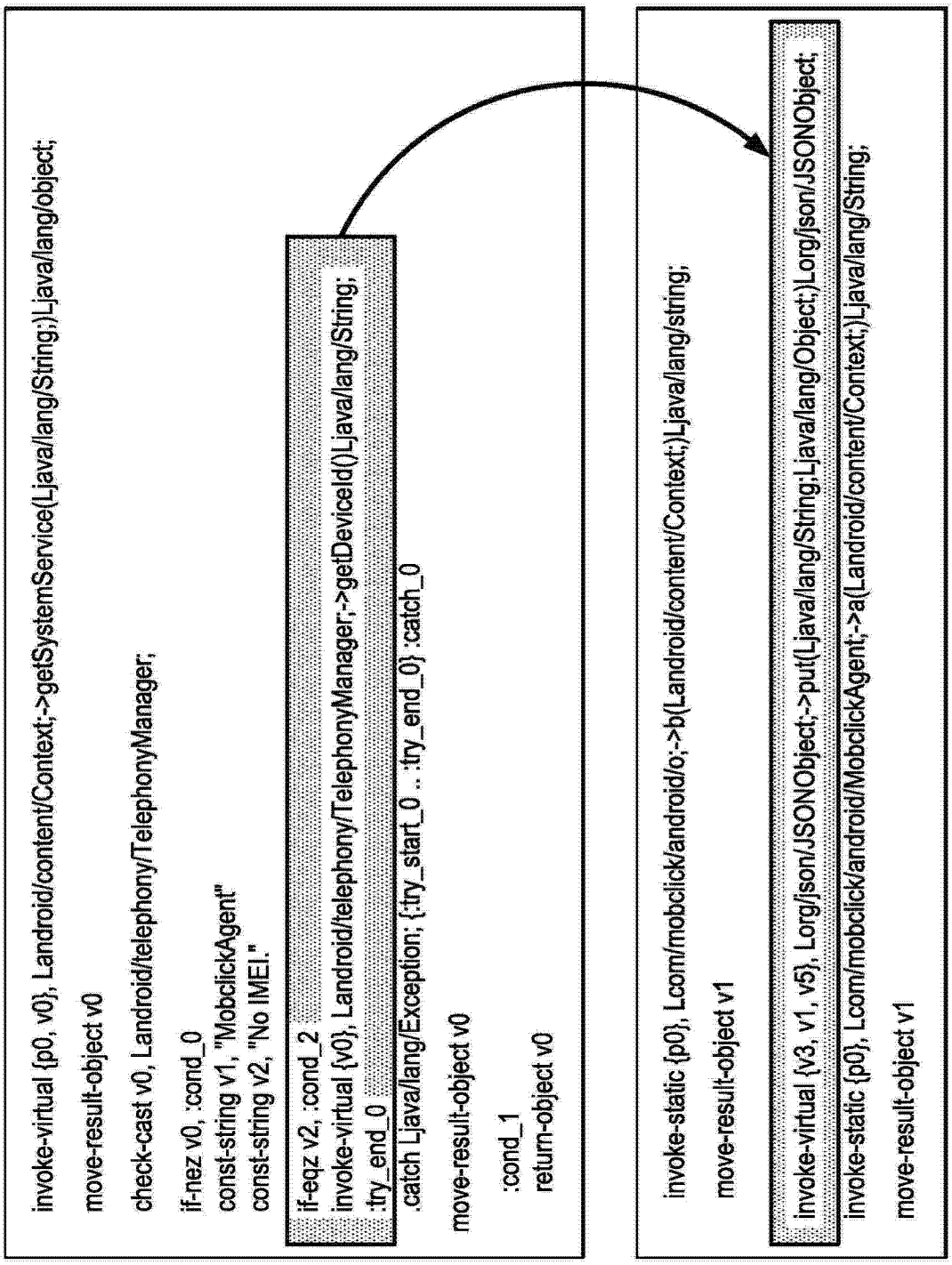

invoke-virtual {p0, v0}, Landroid/content/Context;->getSystemService(Ljava/lang/String;)Ljava/lang/object;
move-result-object v0
check-cast v0, Landroid/telephony/TelephonyManager;
if-nez v0, :cond_0
const-string v1, "MobclickAgent"
const-string v2, "No IMEI."
if-eqz v2, :cond_2
invoke-virtual {v0}, Landroid/telephony/TelephonyManager;->getDeviceId()Ljava/lang/String;
:try_end_0
.catch Ljava/lang/Exception; {:try_start_0 .. :try_end_0} :catch_0
move-result-object v0
:cond_1
return-object v0
invoke-static {p0}, Lcom/mobclick/android/o;->b(Landroid/content/Context;)Ljava/lang/string;
move-result-object v1
invoke-virtual {v3, v1, v5}, Lorg/json/JSONObject;->put(Ljava/lang/String;Ljava/lang/Object;)Lorg/json/JSONObject;
invoke-static {p0}, Lcom/mobclick/android/MobclickAgent;->a(Landroid/content/Context;)Ljava/lang/String;
move-result-object v1

图 12C

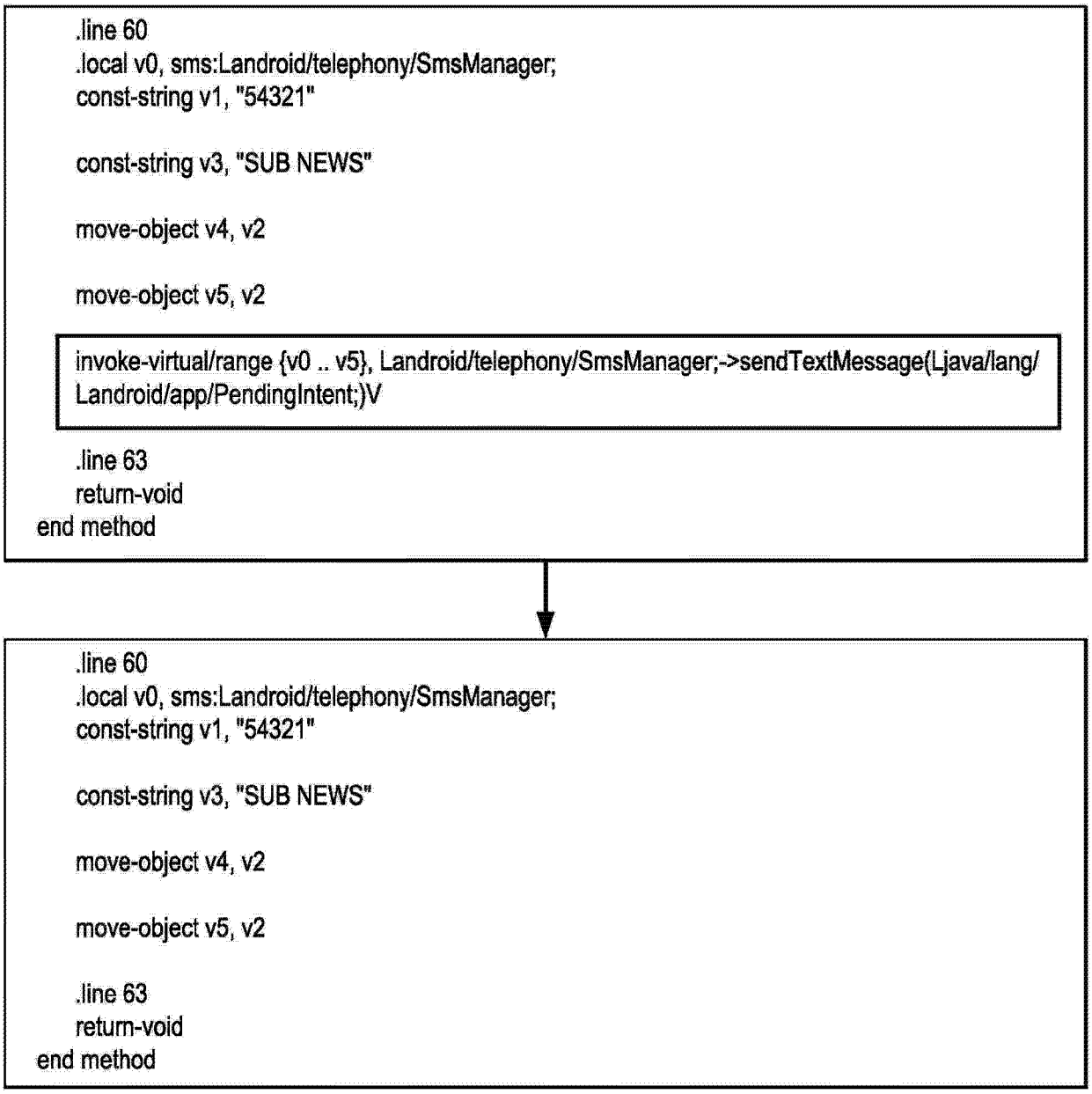

.line 60
.local v0, sms:Landroid/telephony/SmsManager;
const-string v1, "54321"
const-string v3, "SUB NEWS"
move-object v4, v2
move-object v5, v2
invoke-virtual/range {v0 .. v5}, Landroid/telephony/SmsManager;->sendTextMessage(Ljava/lang/
Landroid/app/PendingIntent;)V
.line 63
return-void
end method
.line 60
.local v0, sms:Landroid/telephony/SmsManager;
const-string v1, "54321"
const-string v3, "SUB NEWS"
move-object v4, v2
move-object v5, v2
.line 63
return-void
end method

图 12D

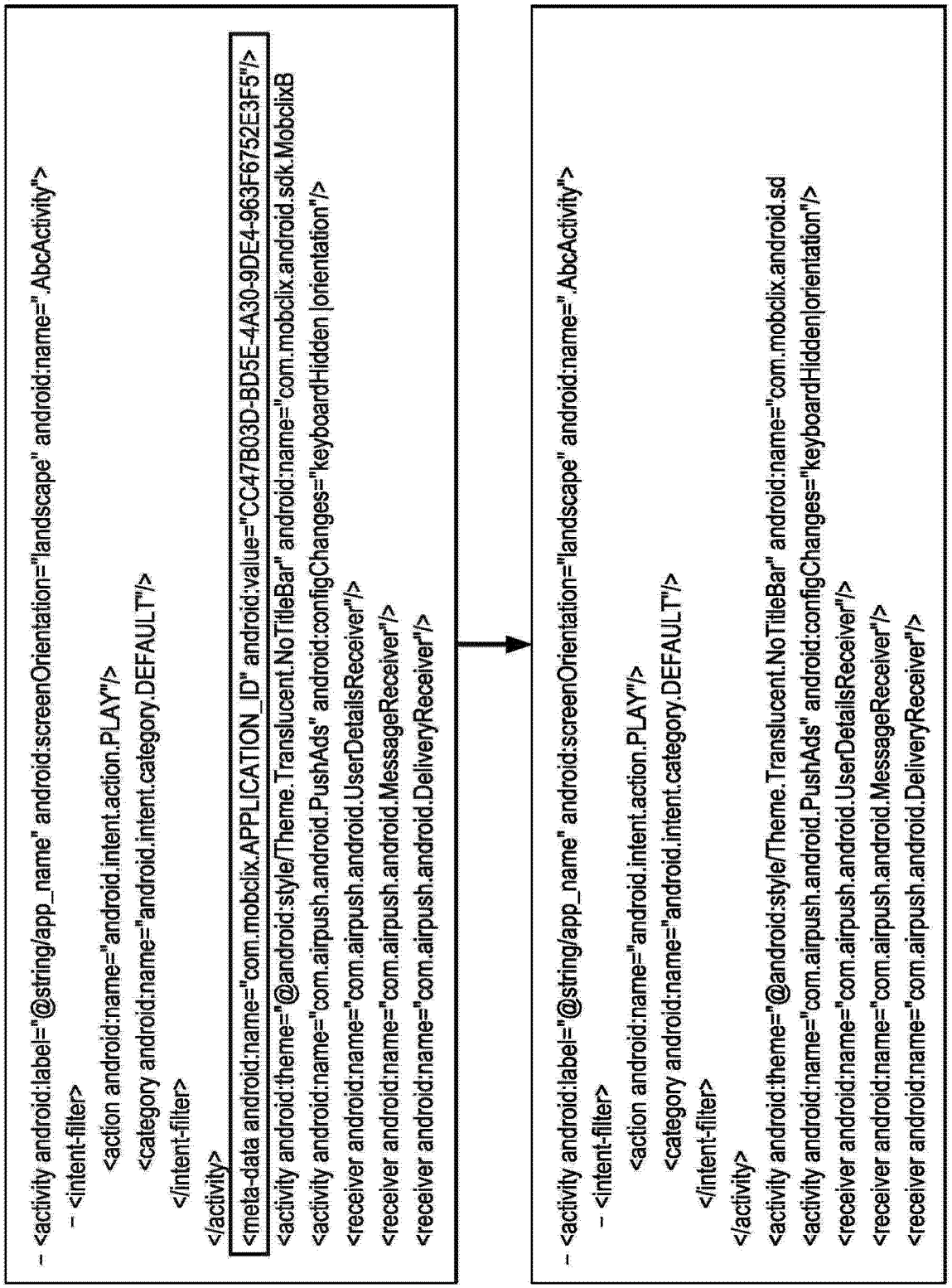

- <activity android:label="@string/app_name" android:screenOrientation="landscape" android:name=".AbcActivity">
- <intent-filter>
<action android:name="android.intent.action.PLAY"/>
<category android:name="android.intent.category.DEFAULT"/>
</intent-filter>
</activity>
<meta-data android:name="com.mobclix.APPLICATION_ID" android:value="CC47B03D-BD5E-4A30-9DE4-963F6752E3F5"/>
<activity android:theme="@android:style/Theme.Translucent.NoTitleBar" android:name="com.mobclix.android.sdk.MobclixB
<activity android:name="com.airpush.android.PushAds" android:configChanges="keyboardHidden |orientation"/>
<receiver android:name="com.airpush.android.UserDetailsReceiver"/>
<receiver android:name="com.airpush.android.MessageReceiver"/>
<receiver android:name="com.airpush.android.DeliveryReceiver"/>
- <activity android:label="@string/app_name" android:screenOrientation="landscape" android:name=".AbcActivity">
- <intent-filter>
<action android:name="android.intent.action.PLAY"/>
<category android:name="android.intent.category.DEFAULT"/>
</intent-filter>
</activity>
<activity android:theme="@android:style/Theme.Translucent.NoTitleBar" android:name="com.mobclix.android.sd
<activity android:name="com.airpush.android.PushAds" android:configChanges="keyboardHidden|orientation"/>
<receiver android:name="com.airpush.android.UserDetailsReceiver"/>
<receiver android:name="com.airpush.android.MessageReceiver"/>
<receiver android:name="com.airpush.android.DeliveryReceiver"/>

图 12E

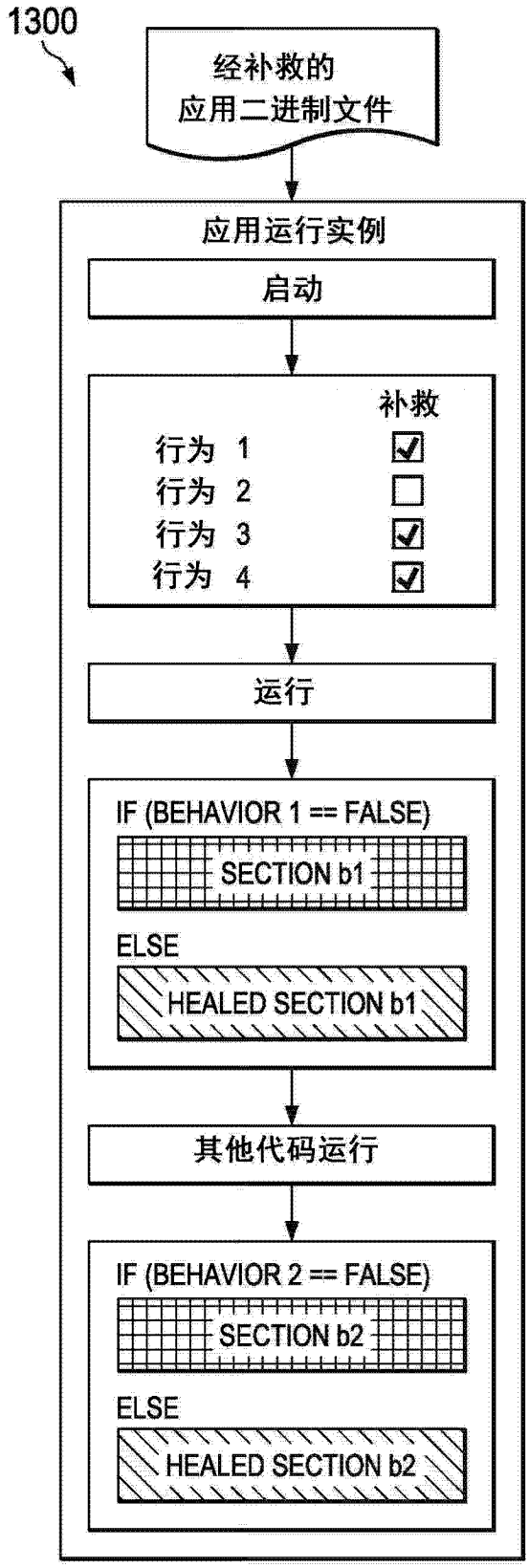

1300
经补救的
应用二进制文件
应用运行实例
启动
补救
行为 1
行为 2
行为 3
行为 4
运行
IF (BEHAVIOR 1 == FALSE)
SECTION b1
ELSE
HEALED SECTION b1
其他代码运行
IF (BEHAVIOR 2 == FALSE)
SECTION b2
ELSE
HEALED SECTION b2

图 13

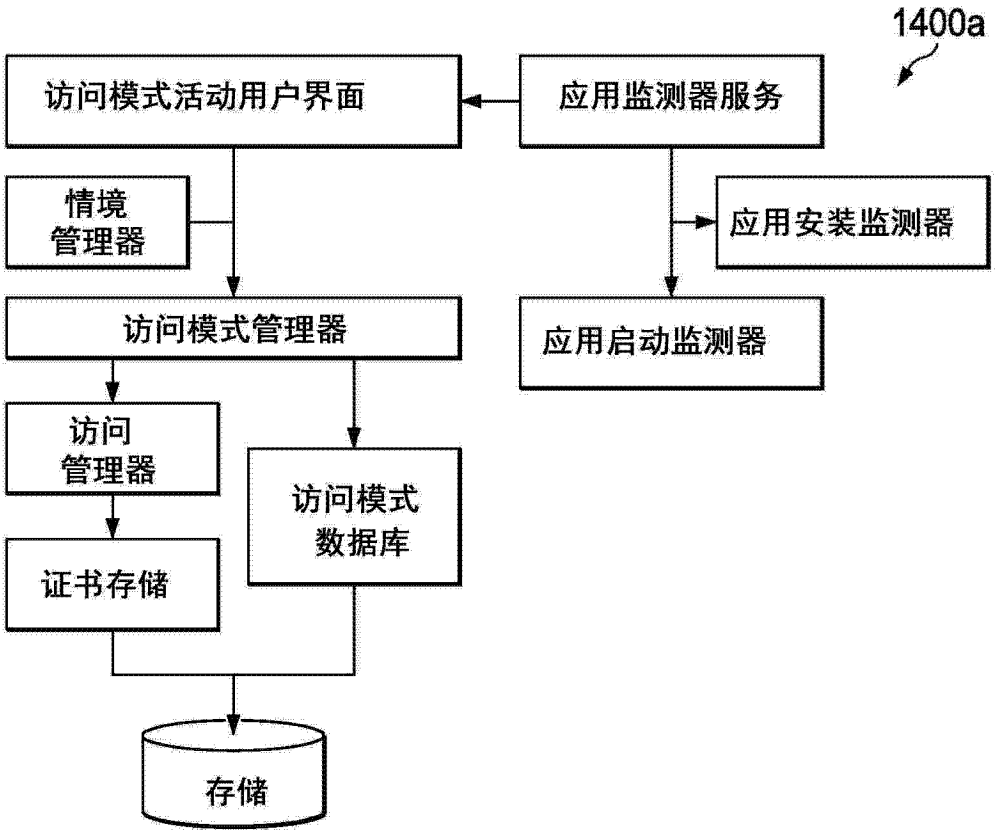

1400a
访问模式活动用户界面
应用监测器服务
情境
管理器
应用安装监测器
访问模式管理器
应用启动监测器
访问
管理器
访问模式
数据库
证书存储
存储

图 14A

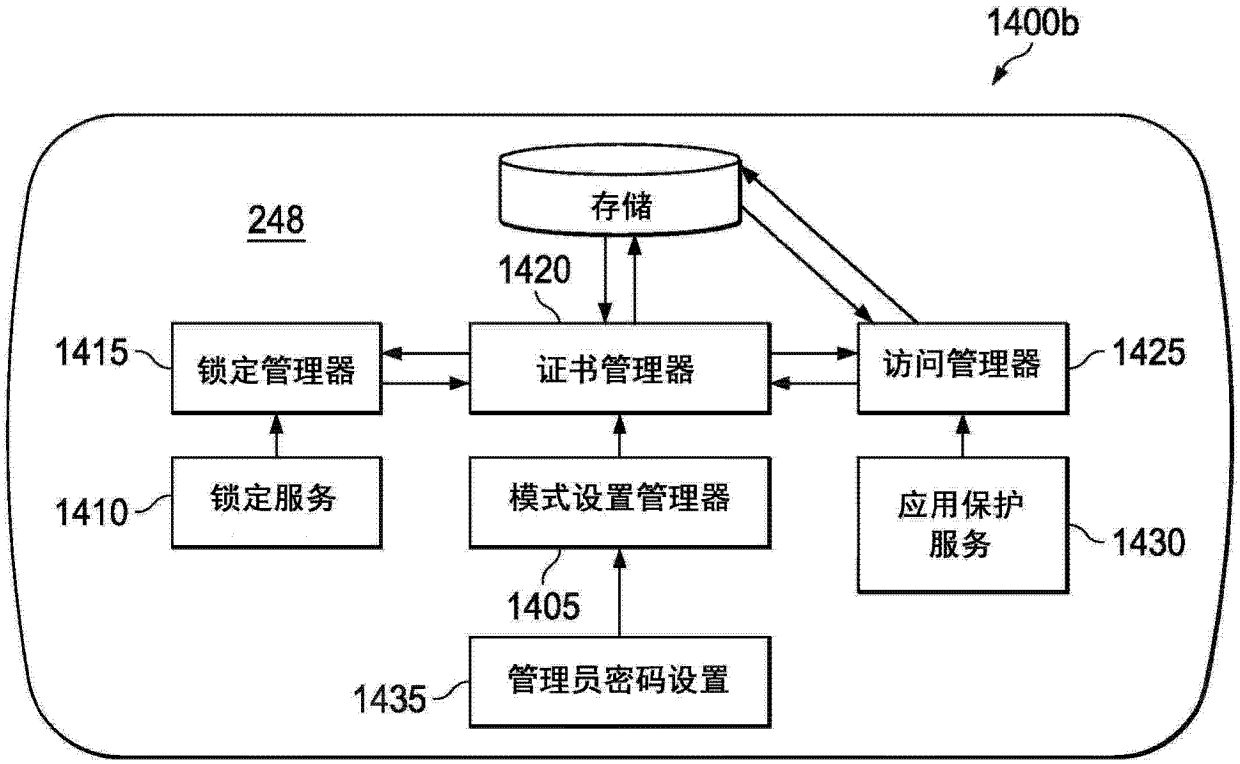

1400b
248
存储
1420
1415
锁定管理器
证书管理器
访问管理器
1425
1410
锁定服务
模式设置管理器
应用保护
服务
1430
1405
1435
管理员密码设置

图 14B

密码存储

从用户p处获取密码

```
key = generatekey(P+ salt1)
ep = encryptwithkey( key, p)
store touple <ModeName, ep >
```

密码验证

从用户p处获取密码

```
key = generatekey(p+ salt1)
dp = decryptwithkey(key, ep;
if ( dp equals p)
        return success;
else
        return failure;
```

图 15A

密码验证以及确定要激活的模式

在设备锁屏上呈现用于输入密码的用户UI元素

从用户p处获取密码

```
key = generatekey(p+ salt1)
for each mode<ModeName, ep > on device
        dp = decryptwithkey( key, ep)
        if ( dp equals p)
                return ModeName; //entered password matched with this modes
        password
        else
                continue;
        return failure; // no modes matched the password provided
```

图 15B

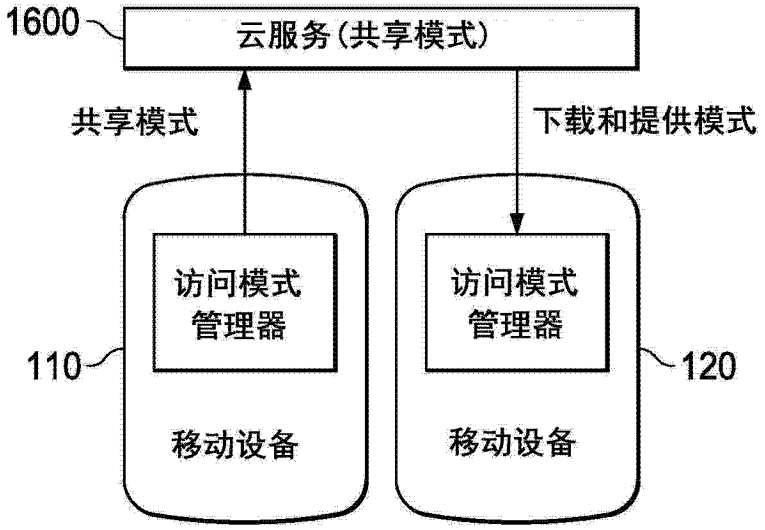

1600
云服务(共享模式)
共享模式
下载和提供模式
访问模式
管理器
访问模式
管理器
110
120
移动设备
移动设备

图 16

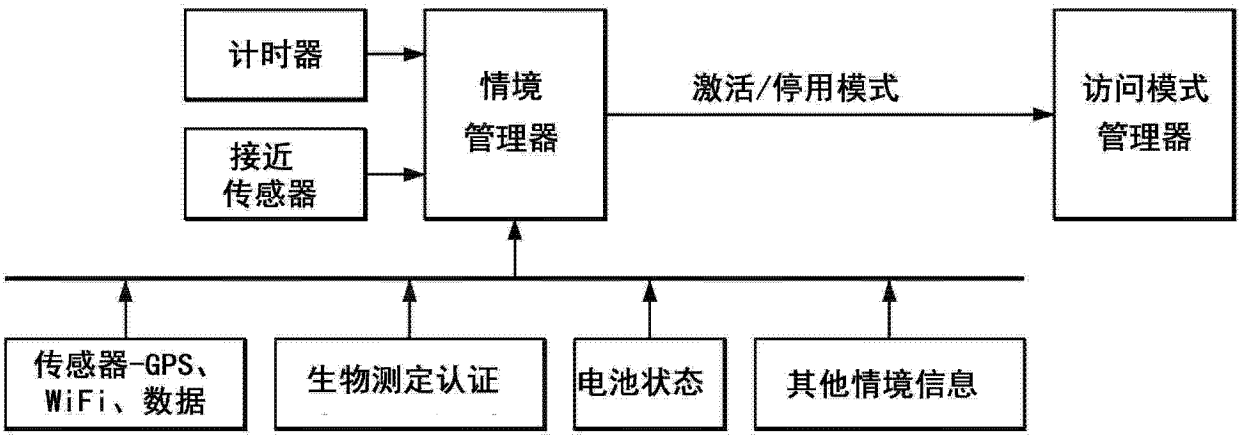

计时器
接近
传感器
情境
管理器
激活/停用模式
访问模式
管理器
传感器-GPS、
WiFi、数据
生物测定认证
电池状态
其他情境信息

图 17

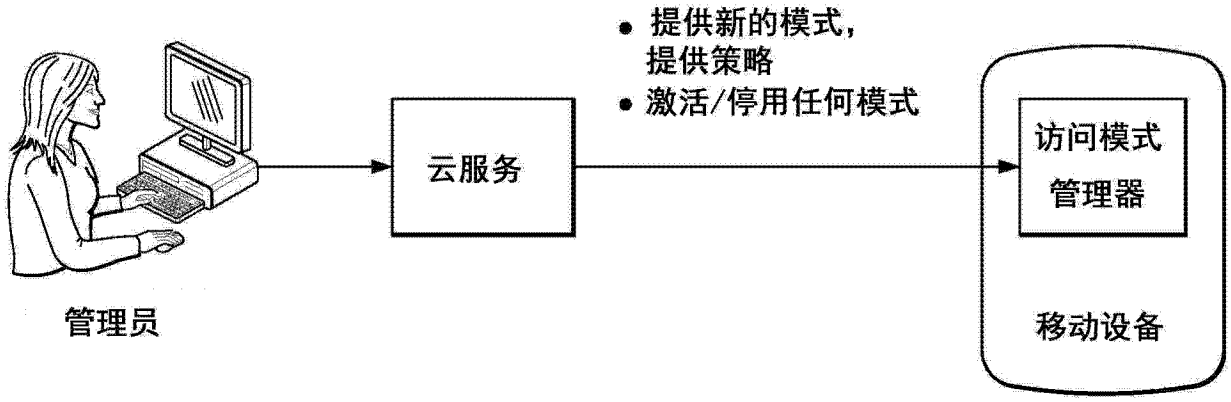

• 提供新的模式，
提供策略
• 激活/停用任何模式
云服务
访问模式
管理器
移动设备
管理员

图 18

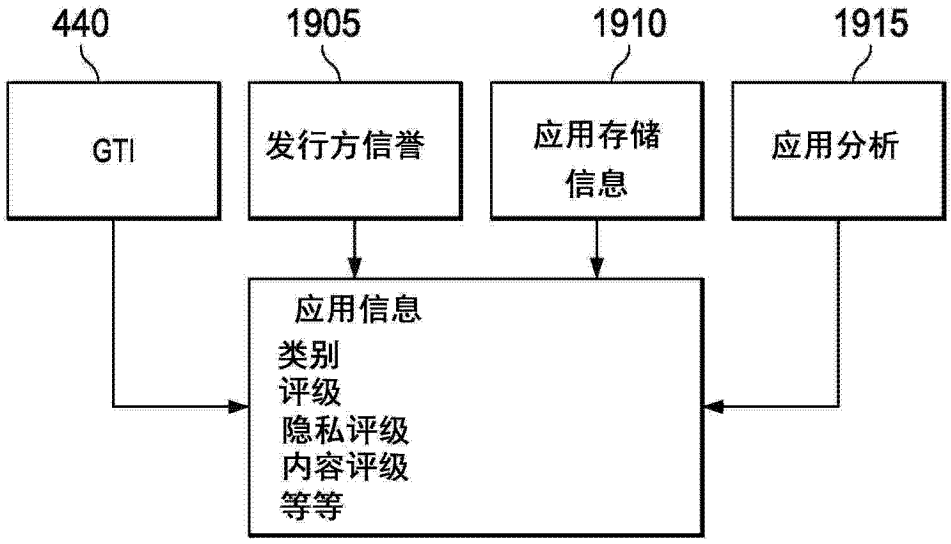

440
1905
1910
1915
GTI
发行方信誉
应用存储信息
应用分析
应用信息
类别
评级
隐私评级
内容评级
等等

图 19

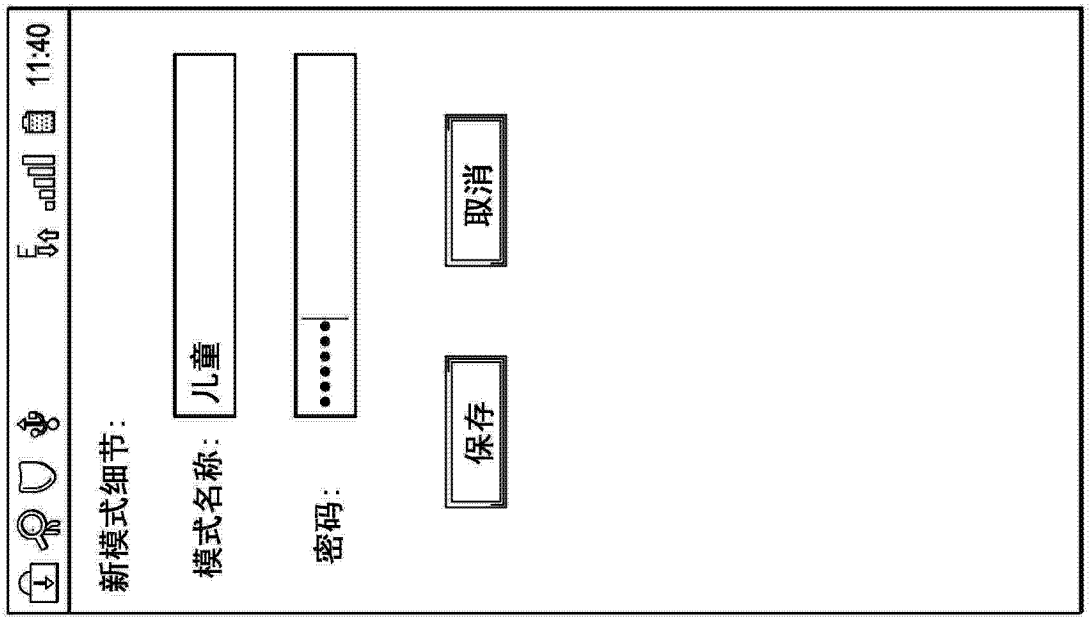

11:40
新模式细节:
模式名称:
儿童
密码:
取消
保存

图 20A

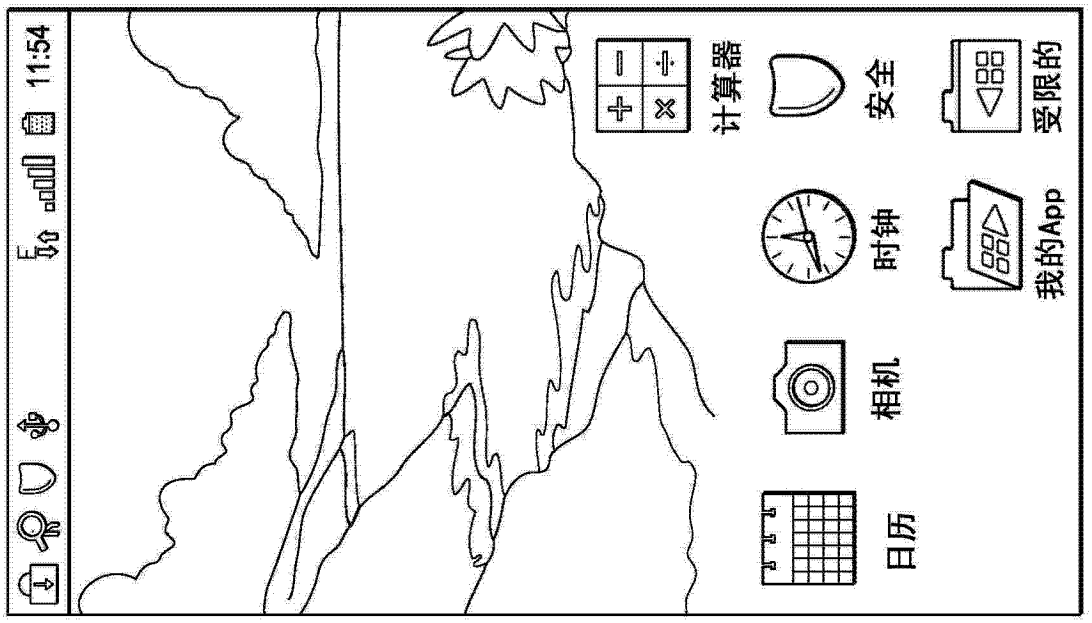

11:54
计算器
安全
受限的
时钟
我的App
相机
日历

图 20B

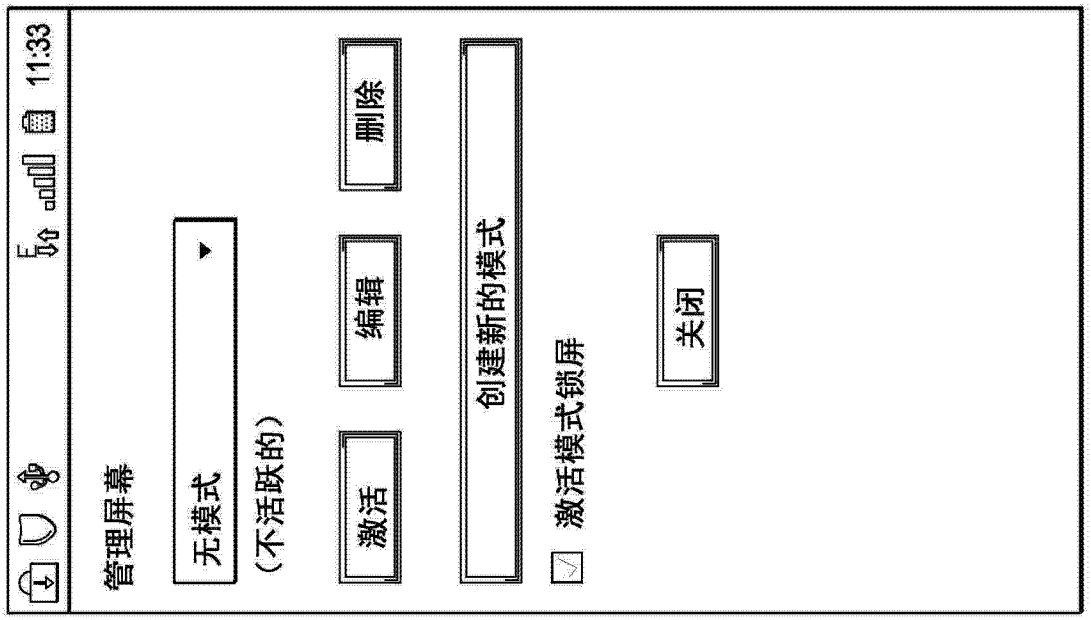

11:33
管理屏幕
无模式
(不活跃的)
激活
编辑
删除
创建新的模式
激活模式锁屏
关闭

图 20C

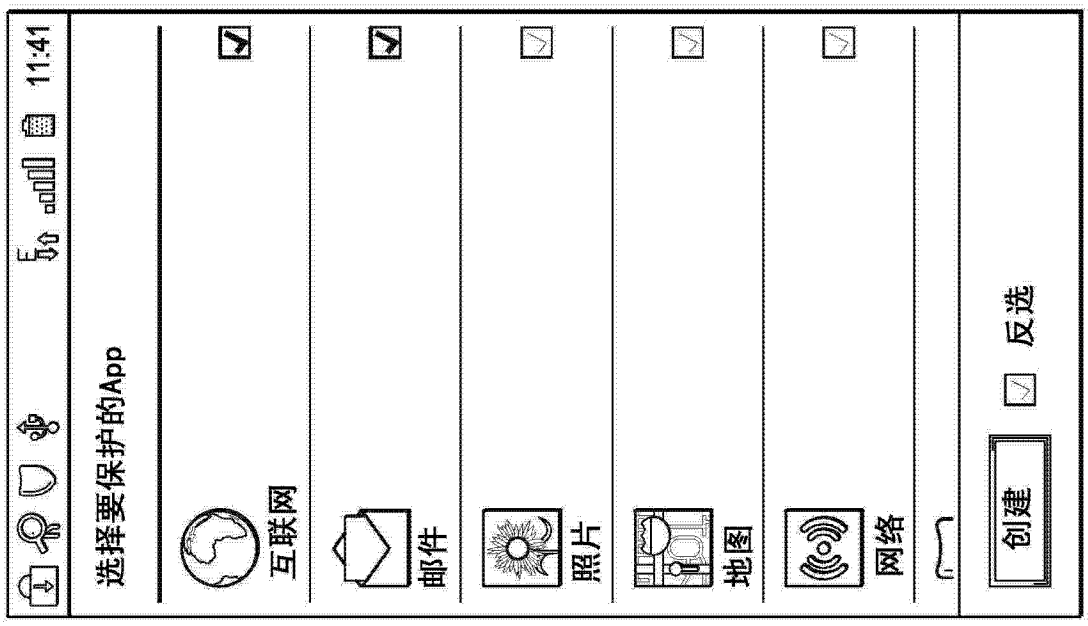


图 20D

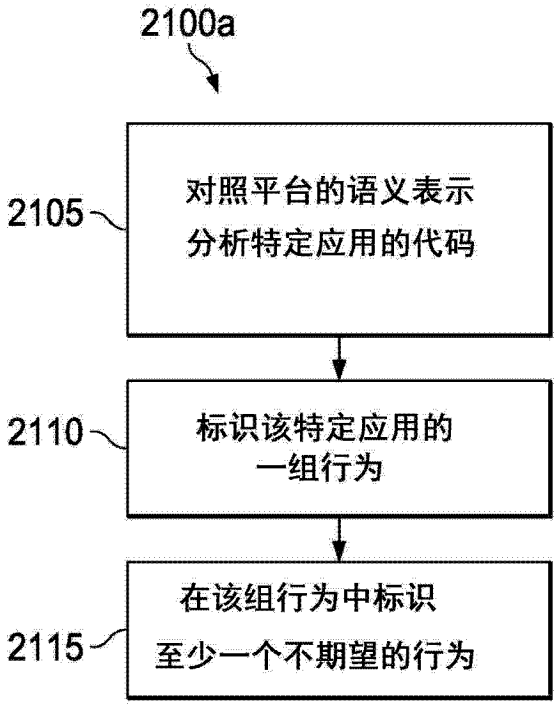


图 21A

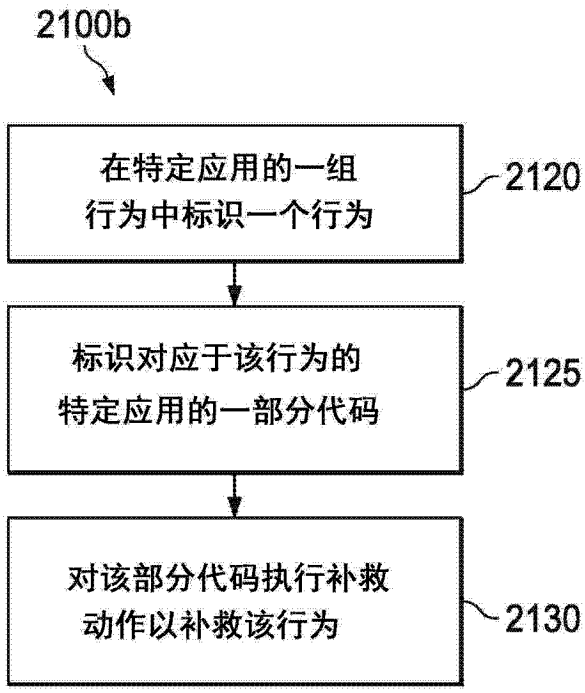


图 21B

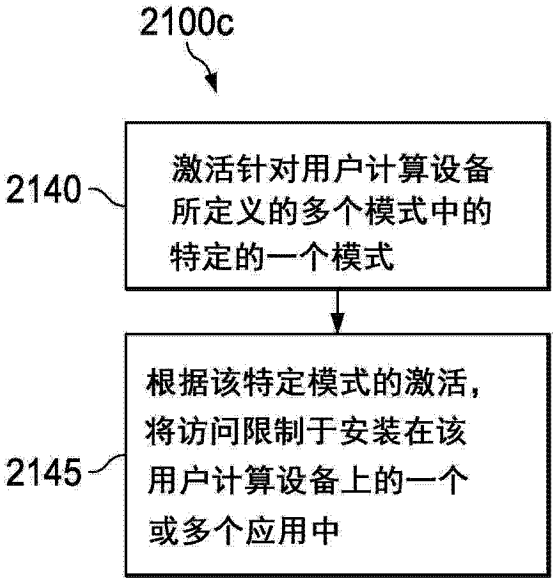

2100c
2140
激活针对用户计算设备所定义的多个模式中的特定的一个模式
2145
根据该特定模式的激活，将访问限制于安装在该用户计算设备上的一个或多个应用中

图 21C