

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2020 年 1 月 9 日 (09.01.2020)



(10) 国际公布号
WO 2020/006911 A1

(51) 国际专利分类号:
G06F 8/60 (2018.01)

(21) 国际申请号: PCT/CN2018/108781

(22) 国际申请日: 2018 年 9 月 29 日 (29.09.2018)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201810710034.8 2018 年 7 月 2 日 (02.07.2018) CN

(71) 申请人: 平安科技(深圳)有限公司 (PING AN TECHNOLOGY(SHENZHEN)CO., LTD.) [CN/CN]; 中国广东省深圳市福田区福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518000 (CN)。

(72) 发明人: 邹国忠(ZOU, Guozhong); 中国广东省深圳市福田区福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518000 (CN)。

(74) 代理人: 北京英特普罗知识产权代理有限公司 (INTELLECPRO CHINA LIMITED); 中国北京市西城区车公庄大街 9 号五栋大楼 C 座 11 层专利信息部, 李莎莎, Beijing 100044 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL,

(54) **Title:** HYBRID APP INTERFACE CALLING METHOD AND DEVICE, AND COMPUTER-READABLE STORAGE MEDIUM

(54) 发明名称: 混合型应用的接口调用方法、装置及计算机可读存储介质

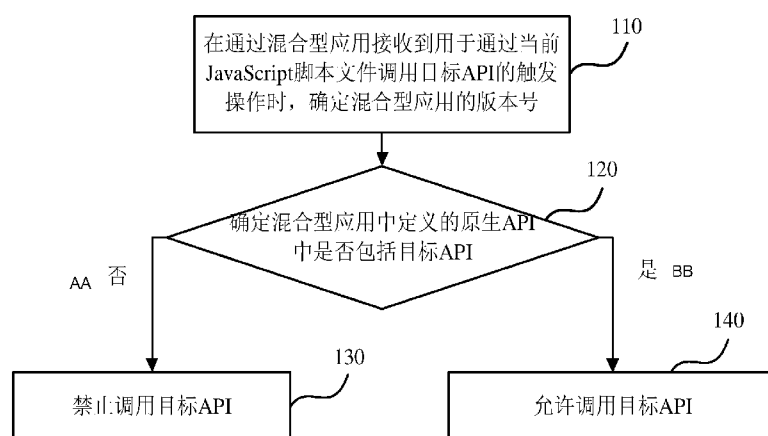


图 1

110 WHEN A TRIGGERING OPERATION USED TO CALL A TARGET API VIA A CURRENT JAVASCRIPT SCRIPT FILE IS RECEIVED VIA A HYBRID APP, DETERMINE A VERSION NUMBER OF THE HYBRID APP
120 DOES A NATIVE API DEFINED IN THE HYBRID APP COMPRISE THE TARGET API?
130 FORBID CALLING OF THE TARGET API
140 ALLOW CALLING OF THE TARGET API
AA NO
BB YES

(57) **Abstract:** The present application provides a hybrid app interface calling method, comprising the following steps: when a triggering operation used to call a target API via a current JavaScript script file is received by means of a hybrid app, determining a version number of the hybrid app; and determining, according to the version number and the current JavaScript script file, whether a native API defined in the hybrid app comprises the target API, so as to forbid or allow calling of the target API. The present application achieves version control of a pseudo-protocol for interaction between a JavaScript and a native script file. In a situation in which a current JavaScript script file does not match a native script file because JavaScript has performed hot updating but an application has not been upgraded,

PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,
SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, ZA, ZM, ZW。

- (84) 指定国(除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM,
AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,
CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 包括国际检索报告(条约第21条(3))。

if it is detected that a current version cannot complete calling, then calling of a target API is forbidden when the current JavaScript script file is used to respond to a triggering operation, and the calling instance is directly skipped, so as to ensure that main functions of a hybrid app operate normally.

(57) 摘要: 本申请提供一种混合型应用的接口调用方法, 包括如下步骤: 在通过混合型应用接收到用于通过当前JavaScript脚本文件调用目标API的触发操作时, 确定混合型应用的版本号, 根据版本号与当前JavaScript脚本文件, 确定混合型应用中定义的原生API中是否包括目标API, 从而禁止调用或允许调用目标API; 本申请实现了对JavaScript与原生交互的伪协议的版本控制, 在JavaScript进行热更新但应用未升级而导致的当前JavaScript脚本文件与原生不匹配的情况下, 若检测到当前版本下无法完成调用时, 则在通过当前JavaScript脚本文件响应上述触发操作时, 禁止调用目标API并直接跳过此次调用, 保证混合型应用主体功能的正常使用。

混合型应用的接口调用方法、装置及计算机可读存储介质

本申请申明享有 2018 年 7 月 2 日递交的申请号为 CN2018107100348、名称为“混合型应用的接口调用方法、装置及计算机可读存储介质”的中国专利申请的优先权，该中国专利申请的整体内容以参考的方式结合在本申请中。

技术领域

本申请涉及计算机领域，尤其涉及一种混合型应用的接口调用方法、装置及计算机可读存储介质。

背景技术

混合型应用（Hybrid App）是一种介于网页应用(Web App)和原生应用（Native App）之间的应用，指的是应用的原生（Native）代码中部分功能采用动态网页（HTML5+JavaScript）来实现，兼具原生应用良好的用户交互体验以及网页应用跨平台开发的优势。

在混合型应用的使用过程中，原生代码和 JavaScript（一种直译式脚本语言）代码互相调用并传递参数，而且能回调数据，但原生与 JavaScript 在相互调用过程中存在以下问题：在混合型应用的开发过程中，原生代码和 JavaScript 代码通常是分开开发的，当需要为混合型应用增加一种新的功能时，需要在 JavaScript 代码中增加对应于该新功能的调用方法，同时对原生代码进行更新使原生代码增加一个与该新功能对应的 API（Application Programming Interface，应用程序编程接口）；JavaScript 代码热更新后的新版本直接被发布到线上，因此可以直接使用更新后的 JavaScript 代码，但原生代码更新后的新版本只有在用户更新该混合型应用后才能被使用，而实际使用时，用户通常不会及时更新混合型应用，也就不会及时获取到新版本的原

生代码，则当新版本的 JavaScript 执行新增加的调用来调用原生时，由于旧版本的原生代码中并不包括对应的 API，因此 JavaScript 中的新的调用方法无法完成调用，混合型应用程序无法完成相应功能的启动，甚至无法继续使用，兼容性和使用体验都较差。

发明内容

本申请的目的在于提供一种混合型应用的接口调用方法、装置及计算机可读存储介质，通过版本控制提高了混合型应用的兼容性，同时保证了 JavaScript 异步调用时的安全性。

本申请是通过下述技术方案来解决上述技术问题：

一种混合型应用的接口调用方法，该方法包括：

若通过混合型应用接收到用于通过当前 JavaScript 脚本文件调用目标 API 的触发操作，则确定混合型应用的版本号；

根据版本号与当前 JavaScript 脚本文件，确定混合型应用中定义的原生 API 中是否包括目标 API；

若原生 API 中不包括目标 API，则在通过当前 JavaScript 脚本文件响应触发操作时，禁止调用目标 API；

若原生 API 中包括目标 API，则在通过当前 JavaScript 脚本文件响应触发操作时，允许调用目标 API。

为了实现上述目的，本申请还提供了一种混合型应用的接口调用装置，该装置包括：

版本确定模块，用于在通过混合型应用接收到用于通过当前 JavaScript 脚本文件调用目标 API 的触发操作时，确定混合型应用的版本号；

检测模块，用于根据版本号与当前 JavaScript 脚本文件，确定混合型应用中定义的原生 API 中是否包括目标 API；

操作模块，用于在原生 API 中不包括目标 API 时，通过当前 JavaScript 脚本文件响应触发操作时，禁止调用目标 API；

调用模块，用于在原生 API 中包括目标 API 时，通过当前 JavaScript 脚本文件响应触发操作时，允许调用目标 API。

为了实现上述目的，本申请还提供了一种计算机设备，该计算机设备包括存储器、处理器以及存储在存储器上并可在处理器上运行的计算机程序，处理器执行计算机程序时实现混合型应用的接口调用方法的以下步骤：

若通过混合型应用接收到用于通过当前 JavaScript 脚本文件调用目标 API 的触发操作，则确定混合型应用的版本号；

根据版本号与当前 JavaScript 脚本文件，确定混合型应用中定义的原生 API 中是否包括目标 API；

若原生 API 中不包括目标 API，则在通过当前 JavaScript 脚本文件响应触发操作时，禁止调用目标 API；

若原生 API 中包括目标 API，则在通过当前 JavaScript 脚本文件响应触发操作时，允许调用目标 API。

为了实现上述目的，本申请还提供了一种计算机可读存储介质，计算机可读存储介质内存储有至少一条指令，至少一条指令由至少一个处理器加载并执行实现混合型应用的接口调用方法的以下步骤：

若通过混合型应用接收到用于通过当前 JavaScript 脚本文件调用目标 API 的触发操作，则确定混合型应用的版本号；

根据版本号与当前 JavaScript 脚本文件，确定混合型应用中定义的原生 API 中是否包括目标 API；

若原生 API 中不包括目标 API，则在通过当前 JavaScript 脚本文件响应触发操作时，禁止调用目标 API；

若原生 API 中包括目标 API，则在通过当前 JavaScript 脚本文件响应触发操作时，允许调用目标 API。

本申请的积极进步效果在于：

1、本申请实现了对 JavaScript 与原生交互的伪协议的版本控制，在 JavaScript 进行热更新但应用未升级而导致的当前 JavaScript 脚本文件与当前

版本的原生不匹配的情况下，在通过当前 JavaScript 脚本文件响应调用目标 API 的触发操作时，禁止调用目标 API，直接跳过此次调用，保证混合型应用主体功能的正常使用。

2、JavaScript 在异步调用多个原生方法时，原生按调用顺序依次执行并依次回调，保证了调用过程中的安全性；同时在多次调用符合预设调用顺序时，原生可以对多次调用进行合并处理，全部执行完成后在最后进行一次回调，提高了 JavaScript 与原生之间互相调用的效率。

附图说明

图 1 示出了本申请混合型应用的接口调用方法一实施例的流程图。

图 2 示出了本申请混合型应用的接口调用方法又一实施例的流程图。

图 3 示出了本申请混合型应用的接口调用装置一实施例的程序模块示意图。

图 4 示出了本申请混合型应用的接口调用装置又一实施例的程序模块示意图。

图 5 示出了本申请混合型应用的接口调用装置又一实施例的程序模块示意图。

图 6 示出了本申请混合型应用的接口调用装置又一实施例的程序模块示意图。

图 7 示出了本申请计算机设备一实施例的硬件结构示意图。

具体实施方式

为了使本申请的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本申请进行进一步详细说明。应当理解，此处所描述的具体实施例仅用以解释本申请，并不用于限定本申请。基于本申请中的实施例，本领

域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

实施例一

请参考图 1，示出了混合型应用的接口调用方法的具体步骤，该混合型应用同时包括原生以及 JavaScript 脚本文件，原生和 JavaScript 脚本文件互相调用以实现整个混合型应用的功能。其中，原生基于硬件平台支持的开发工具和语言来开发，其被安装于基于该类硬件平台的计算机设备上并可以调用计算机设备的硬件功能（如扬声器、加速度传感器和摄像头等），例如，原生可以基于 IOS 平台支持的 Xcode 或 Objective-c 来开发并被安装于基于 IOS 平台的计算机设备上，或者，原生基于安卓平台支持的 Eclipse 或 Java 来开发并被安装于基于安卓平台的计算机设备上。JavaScript 脚本文件基于 Web 技术开发。本申请公开的接口调用方法具体如下：

步骤 110，若通过混合型应用接收到用于通过当前 JavaScript 脚本文件调用目标 API 的触发操作，则确定混合型应用的版本号，其中，当前 JavaScript 脚本文件为经过热更新后的最新版本的 JavaScript 脚本文件，该当前 JavaScript 脚本文件可能与当前版本的混合型应用的原生匹配，也可能不匹配。目标 API 为不影响混合型应用正常使用的 API，比如混合型应用为银行类应用，目标 API 为网络信息读取 API，没有该网络信息读取 API 虽然无法读取网络信息但不会影响银行类应用的正常使用，实际目标 API 具体的含义可以自定义。JavaScript 可以通过调用原生提供的相应接口就可以获取混合型应用的版本号，比如调用 `navigator.userAgent()`。

步骤 120，根据版本号与当前 JavaScript 脚本文件，确定混合型应用中定义的原生 API 中是否包括目标 API：由于各个版本的混合型应用定义的原生 API 的列表会维护在服务器端，因此当前 JavaScript 脚本文件根据混合型应用的版本号就能获取到服务器端维护的该混合型应用定义的所有的原生 API。

步骤 130, 若原生 API 中不包括目标 API, 则说明当前 JavaScript 脚本文件与目前原生的版本不匹配, JavaScript 热更新后原生尚未进行更新, 因此此次调用不能完成, 则在通过当前 JavaScript 脚本文件响应触发操作时, 禁止调用目标 API, 通过做法是当前 JavaScript 脚本文件直接跳过此次对目标 API 的调用, 同时还可以在混合型应用中显示提示框提示用户更新该混合型应用。

比如, 目标 API 为网络信息读取 API, 混合型应用维护有 1.0 版本和 1.1 版本, 1.0 版本的混合型应用的原生不提供网络信息读取 API, 1.1 版本的混合型应用的原生提供该网络信息读取 API, JavaScript 已经进行了热更新并增加了调用网络信息读取 API 的调用方法。假设用户安装的混合型应用仍然是 1.0 版本的, 则当前 JavaScript 脚本文件在需要调用该网络信息读取 API 时, 会检测到 1.0 版本的混合型应用的原生不提供相应接口, 目前的做法可能是在用户不更新的状态下强制退出混合型应用不允许用户继续使用, 但网络信息读取不是混合型应用的主要功能, 无法读取网络信息不会影响混合型应用主体功能的正常使用, 因此在本申请的做法中, 当前 JavaScript 脚本文件可以直接跳过对该网络信息读取 API 的调用, 使得用户能够继续使用该混合型应用。

步骤 140, 若原生 API 中包括目标 API, 则说明当前 JavaScript 脚本文件与目前原生的版本相匹配, 此次调用可以完成, 则在通过当前 JavaScript 脚本文件响应触发操作时, 允许调用目标 API。

例如, 在上述例子中, 假设用户安装的混合型应用已经是 1.1 版本的, 则当前 JavaScript 脚本文件在需要调用该网络信息读取 API 时, 会检测到 1.1 版本的混合型应用的原生提供相应接口, 则执行调用读取到当前的网络信息。

实施例二

请参考图 2, 示出了混合型应用在接口调用过程中允许调用目标 API 时的具体步骤, 也即上述实施例一中步骤 140 的具体实施方法的步骤, 包括:

步骤 141, 若原生 API 中包括目标 API, 则说明当前 JavaScript 脚本文件与目前原生的版本相匹配, 此次调用可以完成, 则通过当前 JavaScript 脚本文件执行对目标 API 的调用, 生成本次调用对应的回调标识, 该回调标识用于标识本次调用的回调函数, 生成方法通常为: 通过当前 JavaScript 脚本文件获取调用触发参数, 调用触发参数是此次调用触发操作的各项参数, 调用触发参数包括但不限于接收到调用触发操作的时间戳信息、预定时间内连续接收到调用触发操作的次数以及目标 API 的接口编号等, 由调用触发参数生成字符串, 然后利用字符串转换方法将该字符串转换为预定格式即为回调标识, 比如可以利用字符串转换方法将生成的该字符串转换为 20 进制的短字符串。

步骤 142, 通过当前 JavaScript 脚本文件将回调标识作为参数传送给原生, 并对目标 API 进行调用。当前 JavaScript 脚本文件采用异步调用的方法进行调用, 也即当前 JavaScript 脚本文件在执行一次调用后, 不需要等待原生完成回调就可以按照步骤 141 和步骤 142 的方法继续执行下一次调用。则同一时刻可能会存在一个或多个 JavaScript 的调用。

步骤 143, 若当前 JavaScript 脚本文件在预设时长内仅执行一次调用, 则原生执行本次调用的业务逻辑, 其中预设时长可以自定义。在执行完成本次调用的业务逻辑后, 原生调用此次调用的回调标识对应的回调函数完成本次调用的回调, 在回调时, 原生可以将业务逻辑的执行结果数据作为参数传送给当前 JavaScript 脚本文件以向当前 JavaScript 脚本文件返回此次调用相应的数据, 然后原生清除回调标识, 保证回调的唯一性。

步骤 144, 若当前 JavaScript 脚本文件在预设时长内执行 N 次调用, $N \geq 2$ 且 N 为正整数, 则原生检测 N 次调用的调用顺序是否符合预设调用顺序, 该预设调用顺序可以是一个自定义的顺序规则, 比如比较常用的一种预设调用顺序为: 在调用数据修改类 API 之前和之后分别调用了数据读取 API, 数据修改类 API 包括数据写入 API、数据删除 API 和数据更新 API 中的至少一种, 比如 4 次调用依次为: 调用数据读取 API (getData())、调用数据删除

API (deletedData())、调用数据写入 API (saveData())、调用数据读取 API。

步骤 145，当检测到 N 次调用的调用顺序符合预设调用顺序时，原生按照调用顺序依次执行 N 次调用，并在执行完成所有 N 次调用的业务逻辑后，调用最后一次调用的回调标识对应的回调函数完成回调，并清除 N 次调用的 N 个回调标识，也即原生只执行一次回调和清除标识的操作，对于第 1 至 N-1 次调用，原生执行完成业务逻辑后并不进行回调和清除标识。

例如，当前 JavaScript 脚本文件执行的 4 次调用依次为：调用数据读取 API (对应回调 ID_1)、调用数据删除 API (对应回调 ID_2)、调用数据写入 API (对应回调 ID_3)、调用数据读取 API (对应回调 ID_4)，符合预设调用顺序。则原生执行第一次调用读取数据后，不执行回调，也不清除回调 ID_1，而是继续执行第二次回调删除数据，然后也不执行回调，也不清除回调 ID_2，继续执行第三次回调写入数据，然后也不执行回调，也不清除回调 ID_3，继续执行第四次回调读取数据，然后调用回调 ID_4 对应的回调函数将最终数据返回给当前 JavaScript 脚本文件，并清除回调 ID_1、回调 ID_2、回调 ID_3 和回调 ID_4。

步骤 146，当检测到 N 次调用的调用顺序不符合预设调用顺序时，原生按照调用顺序依次执行 N 次调用，且在执行完成每一次调用后，调用当前调用的回调标识对应的回调函数完成回调并清除回调标识。

例如，当前 JavaScript 脚本文件执行的 2 次调用依次为：调用相机 API (对应回调 ID_5) 和调用数据读取 API (对应回调 ID_6)，则原生执行第一次调用打开相机应用后，调用回调 ID_5 对应的回调函数完成回调并清除回调 ID_5，然后原生执行第二次调用读取数据后，调用回调 ID_6 对应的回调函数完成回调并清除回调 ID_6。

实施例三

请参考图 3，示出了一种混合型应用的接口调用装置 300，该装置被分割成一个或多个程序模块，每一个程序模块可以进一步被分割成一个或多个

程序子模块及程序单元，该一个或多个程序模块被存储在存储介质中，并由一个或多个处理器所执行，以完成本申请。本申请中的程序模块、程序子模块和程序单元是指能够完成特定功能的一系列计算机程序指令段，比程序本身更适合描述接口调用装置在存储介质中能够的执行过程，以下描述将具体介绍本实施例各模块、子模块和单元等功能：

版本确定模块 310，用于在通过混合型应用接收到用于通过当前 JavaScript 脚本文件调用目标 API 的触发操作时，确定混合型应用的版本号；

检测模块 320，用于根据版本号与当前 JavaScript 脚本文件，确定混合型应用中定义的原生 API 中是否包括目标 API；

操作模块 330，用于在原生 API 中不包括目标 API 时，通过当前 JavaScript 脚本文件响应触发操作时，禁止调用目标 API；

调用模块 340，用于在原生 API 中包括目标 API 时，通过当前 JavaScript 脚本文件响应触发操作时，允许调用目标 API。调用模块 340 进一步包括调用子模块 341、标识生成子模块 342、传送子模块 343 和回调子模块 344，请参考图 4。调用子模块 341，用于通过当前 JavaScript 脚本文件执行对目标 API 的调用；标识生成子模块 342，用于通过当前 JavaScript 脚本文件生成本次调用对应的回调标识，回调标识用于标识本次调用的回调函数；传送子模块 343，用于通过当前 JavaScript 脚本文件将回调标识传送给原生；回调子模块 344，用于通过原生执行本次调用的业务逻辑，并在执行完成本次调用的业务逻辑后，调用回调标识对应的回调函数完成本次调用的回调，并清除回调标识。

请参考图 5，进一步的，标识生成子模块 342 包括参数获取单元 342-1 和标识生成单元 342-2。参数获取单元 342-1，用于获取调用触发参数，调用触发参数包括但不限于接收到调用触发操作的时间戳信息、预定时间内连续接收到调用触发操作的次数以及目标 API 的接口编号；标识生成单元 342-2，用于根据调用触发参数生成回调标识。

请参考图 6，进一步的，回调子模块 344 包括检测单元 344-1、第一执

行单元 344-2 以及第二执行单元 344-3。检测单元 344-1，用于通过原生检测 N 次调用的调用顺序是否符合预设调用顺序；第一执行单元 344-2，用于在检测到 N 次调用的调用顺序符合预设调用顺序时，通过原生按照调用顺序依次执行完成 N 次调用的业务逻辑后，调用最后一次调用的回调标识对应的回调函数完成回调，并清除 N 次调用的 N 个回调标识；第二执行单元 344-3，用于在检测到 N 次调用的调用顺序不符合预设调用顺序时，通过原生按照调用顺序依次执行 N 次调用，且在执行完成每一次调用后，调用调用的回调标识对应的回调函数完成回调并清除回调标识。

实施例四

本实施例提供一种计算机设备，如可以执行程序的智能手机、平板电脑、笔记本电脑、台式计算机、机架式服务器、刀片式服务器、塔式服务器或机柜式服务器（包括独立的服务器，或者多个服务器所组成的服务器集群）等。本实施例的计算机设备 70 至少包括但不限于：可通过系统总线相互通信连接的存储器 71 和处理器 72，如图 7 所示。需要指出的是，图 7 仅示出了具有存储器 71 和处理器 72 的计算机设备 70，但是应理解的是，并不要求实施所有示出的组件，可以替代的实施更多或者更少的组件。

本实施例中，存储器 71（即可读存储介质）包括闪存、硬盘、多媒体卡、卡型存储器（例如，SD 或 DX 存储器等）、随机访问存储器（RAM）、静态随机访问存储器（SRAM）、只读存储器（ROM）、电可擦除可编程只读存储器（EEPROM）、可编程只读存储器（PROM）、磁性存储器、磁盘、光盘等。在一些实施例中，存储器 71 可以是计算机设备 70 的内部存储单元，例如该计算机设备 70 的硬盘或内存。在另一些实施例中，存储器 71 也可以是计算机设备 70 的外部存储设备，例如该计算机设备 70 上配备的插接式硬盘，智能存储卡（Smart Media Card, SMC），安全数字（Secure Digital, SD）卡，闪存卡（Flash Card）等。当然，存储器 71 还可以既包括计算机设备 70 的内部存储单元也包括其外部存储设备。本实施例中，存储器 71 通常用于存储

安装于计算机设备 70 的操作系统和各类应用软件，例如实施例三的混合型应用的接口调用装置 300 的程序代码等。此外，存储器 71 还可以用于暂时地存储已经输出或者将要输出的各类数据。

处理器 72 在一些实施例中可以是中央处理器（Central Processing Unit, CPU）、控制器、微控制器、微处理器、或其他数据处理芯片。该处理器 72 通常用于控制计算机设备 70 的总体操作。本实施例中，处理器 72 用于运行存储器 71 中存储的程序代码或者处理数据，例如运行实施例三的混合型应用的接口调用装置 300 的程序代码，以实现实施例一和实施例二的混合型应用的接口调用方法。

实施例五

本实施例提供一种计算机可读存储介质，该计算机可读存储介质内存储有至少一条指令，该至少一条指令由至少一个处理器加载并执行以实现实施例一和二的接口调用方法的操作。

通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。

以上仅为本申请的优选实施例，并非因此限制本申请的专利范围，凡是利用本申请说明书及附图内容所作的等效结构或等效流程变换，或直接或间接运用在其他相关的技术领域，均同理包括在本申请的专利保护范围内。

权利要求

1、一种混合型应用的接口调用方法，其特征在于，所述方法包括：

若通过混合型应用接收到用于通过当前 JavaScript 脚本文件调用目标 API 的触发操作，则确定所述混合型应用的版本号；

根据所述版本号与所述当前 JavaScript 脚本文件，确定所述混合型应用中定义的原生 API 中是否包括所述目标 API；

若所述原生 API 中不包括所述目标 API，则在通过所述当前 JavaScript 脚本文件响应所述触发操作时，禁止调用所述目标 API；

若所述原生 API 中包括所述目标 API，则在通过所述当前 JavaScript 脚本文件响应所述触发操作时，允许调用所述目标 API。

2、根据权利要求 1 所述的方法，其特征在于，所述在通过所述当前 JavaScript 脚本文件响应所述触发操作时，允许调用所述目标 API，包括：

通过所述当前 JavaScript 脚本文件执行对所述目标 API 的调用，生成本次调用对应的回调标识并将所述回调标识传送给原生，所述回调标识用于标识本次调用的回调函数；

通过所述原生执行本次调用的业务逻辑，并在执行完成本次调用的业务逻辑后，调用所述回调标识对应的回调函数完成本次调用的回调，并清除所述回调标识。

3、根据权利要求 2 所述的方法，其特征在于，所述生成本次调用对应的回调标识，包括：

获取调用触发参数，所述调用触发参数包括但不限于接收到所述调用触发操作的时间戳信息、预定时间内连续接收到所述调用触发操作的次数以及所述目标 API 的接口编号；

根据所述调用触发参数生成所述回调标识。

4、根据权利要求 2 所述的方法，其特征在于，所述当前 JavaScript 脚本文件在预设时长内依次执行 N 次调用， $N \geq 2$ ，则通过所述原生执行本次调

用的业务逻辑，并在执行完成本次调用的业务逻辑后，调用所述回调标识对应的回调函数完成本次调用的回调，并清除所述回调标识，包括：

通过所述原生检测所述 N 次调用的调用顺序是否符合预设调用顺序；

当检测到所述 N 次调用的调用顺序符合所述预设调用顺序时，通过所述原生按照所述调用顺序依次执行完成 N 次调用的业务逻辑后，调用最后一次调用的回调标识对应的回调函数完成回调，并清除所述 N 次调用的 N 个回调标识；

当检测到所述 N 次调用的调用顺序不符合所述预设调用顺序时，原生按照所述调用顺序依次执行所述 N 次调用，并在执行完成每一次调用后，调用所述调用的回调标识对应的回调函数完成回调并清除所述回调标识。

5、根据权利要求 4 所述的方法，其特征在于，所述预设调用顺序包括：在调用数据修改类 API 之前和之后分别调用了数据读取 API，所述数据修改类 API 包括数据写入 API、数据删除 API 和数据更新 API 中的至少一种。

6、一种混合型应用的接口调用装置，其特征在于，所述装置包括：

版本确定模块，用于在通过混合型应用接收到用于通过当前 JavaScript 脚本文件调用目标 API 的触发操作时，确定所述混合型应用的版本号；

检测模块，用于根据所述版本号与所述当前 JavaScript 脚本文件，确定所述混合型应用中定义的原生 API 中是否包括所述目标 API；

操作模块，用于在所述原生 API 中不包括所述目标 API 时，通过所述当前 JavaScript 脚本文件响应所述触发操作时，禁止调用所述目标 API；

调用模块，用于在所述原生 API 中包括所述目标 API 时，通过所述当前 JavaScript 脚本文件响应所述触发操作时，允许调用所述目标 API。

7、根据权利要求 6 所述的装置，其特征在于，所述调用模块包括：

调用子模块，用于通过所述当前 JavaScript 脚本文件执行对所述目标 API 的调用；

标识生成子模块，用于通过所述当前 JavaScript 脚本文件生成本次调用对应的回调标识，所述回调标识用于标识本次调用的回调函数；

传送子模块，用于通过所述当前 JavaScript 脚本文件将所述回调标识传送给原生；

回调子模块，用于通过所述原生执行本次调用的业务逻辑，并在执行完成本次调用的业务逻辑后，调用所述回调标识对应的回调函数完成本次调用的回调，并清除所述回调标识。

8、根据权利要求 7 所述的装置，其特征在于，所述标识生成子模块包括：

参数获取单元，用于获取调用触发参数，调用触发参数包括但不限于接收到调用触发操作的时间戳信息、预定时间内连续接收到调用触发操作的次数以及目标 API 的接口编号；

标识生成单元，用于根据调用触发参数生成回调标识。

9、根据权利要求 7 所述的装置，其特征在于，所述回调子模块包括：

检测单元，用于通过所述原生检测所述 N 次调用的调用顺序是否符合预设调用顺序；

第一执行单元，用于在检测到所述 N 次调用的调用顺序符合所述预设调用顺序时，通过所述原生按照所述调用顺序依次执行完成 N 次调用的业务逻辑后，调用最后一次调用的回调标识对应的回调函数完成回调，并清除所述 N 次调用的 N 个回调标识；

第二执行单元，用于在检测到所述 N 次调用的调用顺序不符合所述预设调用顺序时，通过所述原生按照所述调用顺序依次执行所述 N 次调用，并在执行完成每一次调用后，调用所述调用的回调标识对应的回调函数完成回调并清除所述回调标识。

10、根据权利要求 8 所述的装置，其特征在于，所述预设调用顺序包括：在调用数据修改类 API 之前和之后分别调用了数据读取 API，所述数据修改类 API 包括数据写入 API、数据删除 API 和数据更新 API 中的至少一种。11、一种计算机设备，其特征在于，所述计算机设备包括存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的计算机程序，所述处理器执行

所述计算机程序时实现混合型应用的接口调用方法的以下步骤：

若通过混合型应用接收到用于通过当前 JavaScript 脚本文件调用目标 API 的触发操作，则确定所述混合型应用的版本号；

根据所述版本号与所述当前 JavaScript 脚本文件，确定所述混合型应用中定义的原生 API 中是否包括所述目标 API；

若所述原生 API 中不包括所述目标 API，则在通过所述当前 JavaScript 脚本文件响应所述触发操作时，禁止调用所述目标 API；

若所述原生 API 中包括所述目标 API，则在通过所述当前 JavaScript 脚本文件响应所述触发操作时，允许调用所述目标 API。

12、根据权利要求 11 所述的计算机设备，其特征在于，所述在通过所述当前 JavaScript 脚本文件响应所述触发操作时，允许调用所述目标 API，包括：

通过所述当前 JavaScript 脚本文件执行对所述目标 API 的调用，生成本次调用对应的回调标识并将所述回调标识传送给原生，所述回调标识用于标识本次调用的回调函数；

通过所述原生执行本次调用的业务逻辑，并在执行完成本次调用的业务逻辑后，调用所述回调标识对应的回调函数完成本次调用的回调，并清除所述回调标识。

13、根据权利要求 12 所述的计算机设备，其特征在于，所述生成本次调用对应的回调标识，包括：

获取调用触发参数，所述调用触发参数包括但不限于接收到所述调用触发操作的时间戳信息、预定时间内连续接收到所述调用触发操作的次数以及所述目标 API 的接口编号；

根据所述调用触发参数生成所述回调标识。

14、根据权利要求 12 所述的计算机设备，其特征在于，所述当前 JavaScript 脚本文件在预设时长内依次执行 N 次调用， $N \geq 2$ ，则通过所述原生执行本次调用的业务逻辑，并在执行完成本次调用的业务逻辑后，调用所

述回调标识对应的回调函数完成本次调用的回调，并清除所述回调标识，包括：

通过所述原生检测所述 N 次调用的调用顺序是否符合预设调用顺序；

当检测到所述 N 次调用的调用顺序符合所述预设调用顺序时，通过所述原生按照所述调用顺序依次执行完成 N 次调用的业务逻辑后，调用最后一次调用的回调标识对应的回调函数完成回调，并清除所述 N 次调用的 N 个回调标识；

当检测到所述 N 次调用的调用顺序不符合所述预设调用顺序时，原生按照所述调用顺序依次执行所述 N 次调用，并在执行完成每一次调用后，调用所述调用的回调标识对应的回调函数完成回调并清除所述回调标识。

15、根据权利要求 14 所述的计算机设备，其特征在于，所述预设调用顺序包括：在调用数据修改类 API 之前和之后分别调用了数据读取 API，所述数据修改类 API 包括数据写入 API、数据删除 API 和数据更新 API 中的至少一种。

16、一种计算机可读存储介质，其特征在于，所述计算机可读存储介质内存储有至少一条指令，所述至少一条指令由至少一个处理器加载并执行实现混合型应用的接口调用方法的以下步骤：

若通过混合型应用接收到用于通过当前 JavaScript 脚本文件调用目标 API 的触发操作，则确定所述混合型应用的版本号；

根据所述版本号与所述当前 JavaScript 脚本文件，确定所述混合型应用中定义的原生 API 中是否包括所述目标 API；

若所述原生 API 中不包括所述目标 API，则在通过所述当前 JavaScript 脚本文件响应所述触发操作时，禁止调用所述目标 API；

若所述原生 API 中包括所述目标 API，则在通过所述当前 JavaScript 脚本文件响应所述触发操作时，允许调用所述目标 API。

17、根据权利要求 16 所述的计算机可读存储介质，其特征在于，所述在通过所述当前 JavaScript 脚本文件响应所述触发操作时，允许调用所述目

标 API，包括：

通过所述当前 JavaScript 脚本文件执行对所述目标 API 的调用，生成本次调用对应的回调标识并将所述回调标识传送给原生，所述回调标识用于标识本次调用的回调函数；

通过所述原生执行本次调用的业务逻辑，并在执行完成本次调用的业务逻辑后，调用所述回调标识对应的回调函数完成本次调用的回调，并清除所述回调标识。

18、根据权利要求 17 所述的计算机可读存储介质，其特征在于，所述生成本次调用对应的回调标识，包括：

获取调用触发参数，所述调用触发参数包括但不限于接收到所述调用触发操作的时间戳信息、预定时间内连续接收到所述调用触发操作的次数以及所述目标 API 的接口编号；

根据所述调用触发参数生成所述回调标识。

19、根据权利要求 17 所述的计算机可读存储介质，其特征在于，所述当前 JavaScript 脚本文件在预设时长内依次执行 N 次调用， $N \geq 2$ ，则通过所述原生执行本次调用的业务逻辑，并在执行完成本次调用的业务逻辑后，调用所述回调标识对应的回调函数完成本次调用的回调，并清除所述回调标识，包括：

通过所述原生检测所述 N 次调用的调用顺序是否符合预设调用顺序；

当检测到所述 N 次调用的调用顺序符合所述预设调用顺序时，通过所述原生按照所述调用顺序依次执行完成 N 次调用的业务逻辑后，调用最后一次调用的回调标识对应的回调函数完成回调，并清除所述 N 次调用的 N 个回调标识；

当检测到所述 N 次调用的调用顺序不符合所述预设调用顺序时，原生按照所述调用顺序依次执行所述 N 次调用，并在执行完成每一次调用后，调用所述调用的回调标识对应的回调函数完成回调并清除所述回调标识。

20、根据权利要求 19 所述的计算机可读存储介质，其特征在于，所述

预设调用顺序包括：在调用数据修改类 API 之前和之后分别调用了数据读取 API，所述数据修改类 API 包括数据写入 API、数据删除 API 和数据更新 API 中的至少一种。

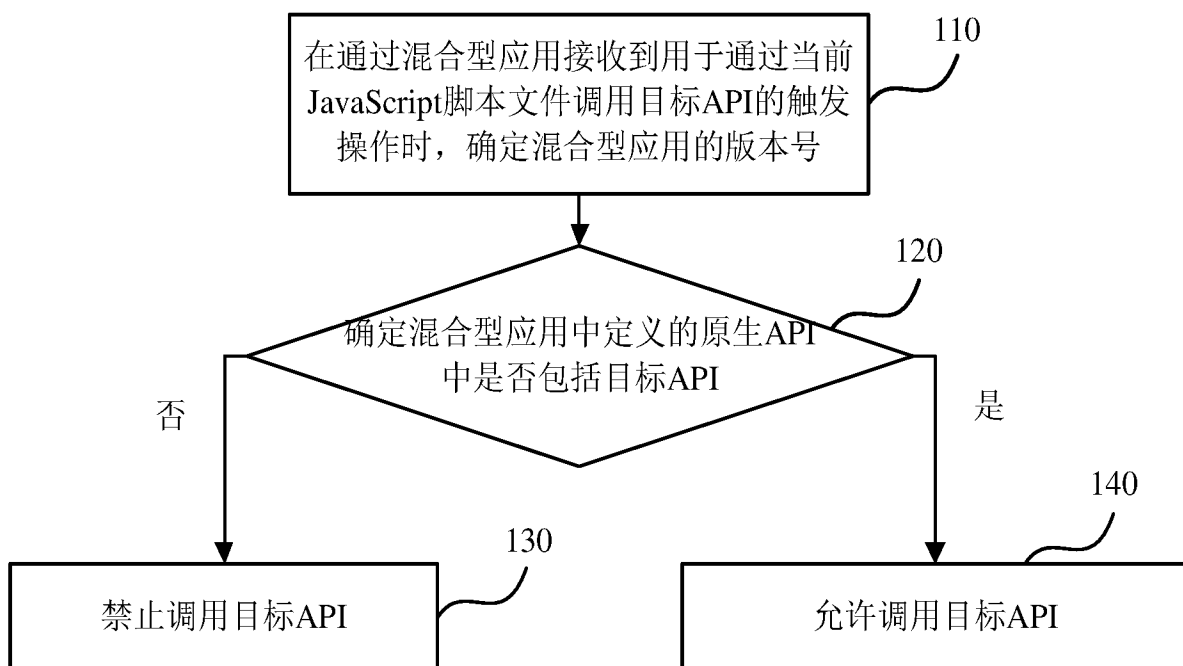


图 1

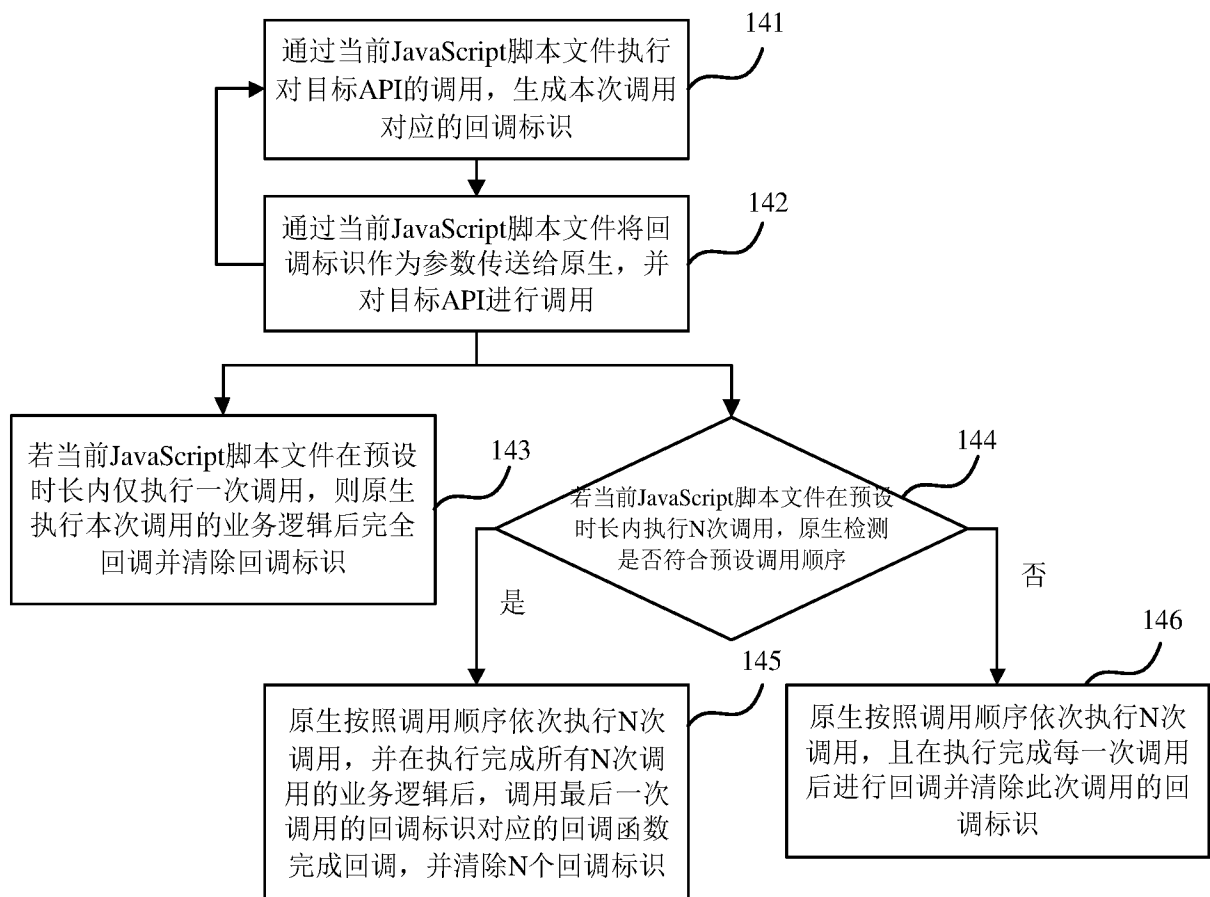


图 2



图 3

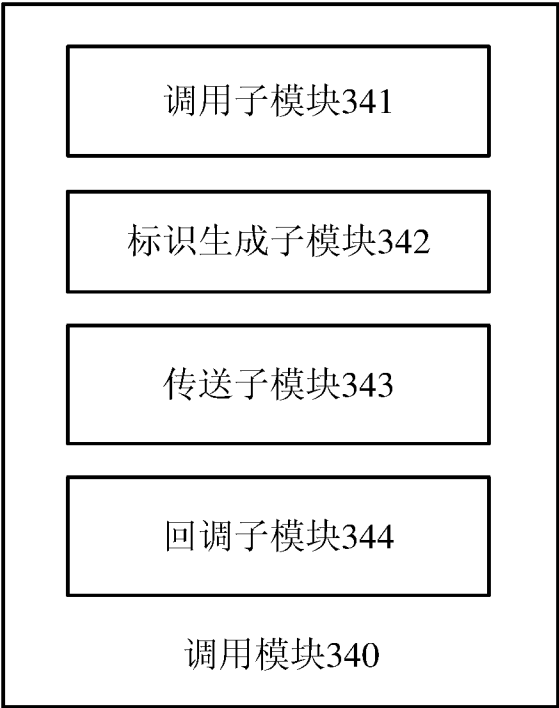


图 4

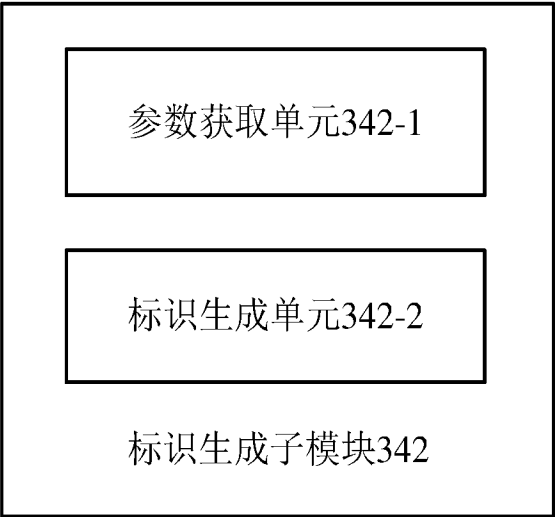


图 5

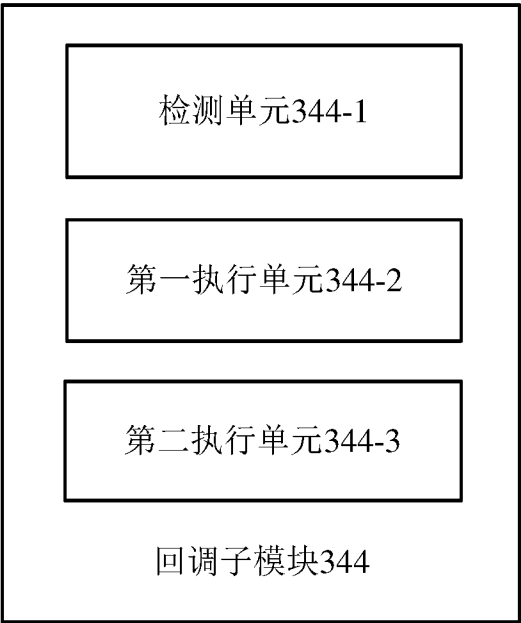


图 6

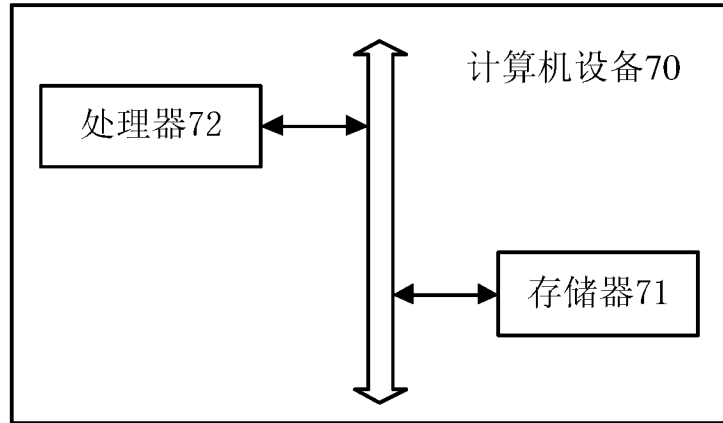


图 7

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/108781

A. CLASSIFICATION OF SUBJECT MATTER

G06F 8/60(2018.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F8/-;9/-

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS; DWPI; CNKI; SIPOABS: API, 接口, native, 脚本, 调用, 版本, 原生, version, interface, script, application, invoke

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 104881318 A (BEIJING KINGSOFT SECURITY SOFTWARE CO., LTD.) 02 September 2015 (2015-09-02) entire document	1-20
A	CN 104978176 A (TENCENT TECHNOLOGY (BEIJING) CO., LTD.) 14 October 2015 (2015-10-14) entire document	1-20
A	CN 102622211 A (TENCENT TECHNOLOGY (SHENZHEN) CO., LTD.) 01 August 2012 (2012-08-01) entire document	1-20

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

07 March 2019

Date of mailing of the international search report

15 March 2019

Name and mailing address of the ISA/CN

**National Intellectual Property Administration, PRC (ISA/
CN)**
**No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing
100088
China**

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2018/108781

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	104881318	A	02 September 2015	WO	2016177341	A1	10 November 2016
CN	104978176	A	14 October 2015	US	2016314219	A1	27 October 2016
				WO	2015154670	A1	15 October 2015
CN	102622211	A	01 August 2012	CN	102622211	B	25 May 2016

国际检索报告

国际申请号

PCT/CN2018/108781

A. 主题的分类

G06F 8/60(2018.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F8/-;9/-

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNABS;DWPI;CNKI;SIPOABS:API, 接口, native, 脚本, 调用, 版本, 原生, version, interface, script, application, invoke

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
A	CN 104881318 A (北京金山安全软件有限公司) 2015年 9月 2日 (2015 - 09 - 02) 全文	1-20
A	CN 104978176 A (腾讯科技北京有限公司) 2015年 10月 14日 (2015 - 10 - 14) 全文	1-20
A	CN 102622211 A (腾讯科技深圳有限公司) 2012年 8月 1日 (2012 - 08 - 01) 全文	1-20

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2019年 3月 7日

国际检索报告邮寄日期

2019年 3月 15日

ISA/CN的名称和邮寄地址

中国国家知识产权局(ISA/CN)
中国北京市海淀区蓟门桥西土城路6号 100088

传真号 (86-10)62019451

受权官员

马驰

电话号码 62412168

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2018/108781

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	104881318	A	2015年 9月 2日	WO	2016177341	A1	2016年 11月 10日
CN	104978176	A	2015年 10月 14日	US	2016314219	A1	2016年 10月 27日
				WO	2015154670	A1	2015年 10月 15日
CN	102622211	A	2012年 8月 1日	CN	102622211	B	2016年 5月 25日

表 PCT/ISA/210 (同族专利附件) (2015年1月)