

(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
2015 年 9 月 17 日 (17.09.2015)



(10) 国际公布号
WO 2015/135221 A1

- (51) 国际专利分类号:
H04L 12/18 (2006.01)
- (21) 国际申请号: PCT/CN2014/073496
- (22) 国际申请日: 2014 年 3 月 17 日 (17.03.2014)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201410090696.1 2014 年 3 月 12 日 (12.03.2014) CN
- (71) 申请人: 深圳创维-RGB 电子有限公司 (SHENZHEN CHUANGWEI-RGB ELECTRONIC CO., LTD.) [CN/CN]; 中国广东省深圳市南山区深南大道创维大厦 A 座 13-16 楼, Guangdong 518057 (CN)。
- (72) 发明人: 郭尚锋 (GUO, Shangfeng); 中国广东省深圳市南山区深南大道创维大厦 A 座 13-16 楼, Guangdong 518057 (CN)。 吕伟龙 (LV, Weilong); 中国广东省深圳市南山区深南大道创维大厦 A 座 13-16 楼, Guangdong 518057 (CN)。
- (74) 代理人: 深圳鼎合诚知识产权代理有限公司 (DHC IP ATTORNEYS); 中国广东省深圳市福田区金田路与福华路交汇处现代商务大厦 2201, Guangdong 518048 (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, 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, 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,

[见续页]

(54) Title: INTER-PROCESS COMMUNICATION METHOD BASED ON APPLICATION LAYER OF ANDROID AND BASIC APPLICATION COMMUNICATION SYSTEM

(54) 发明名称: 基于 Android 的应用层的进程间通信方法及基础应用通信系统

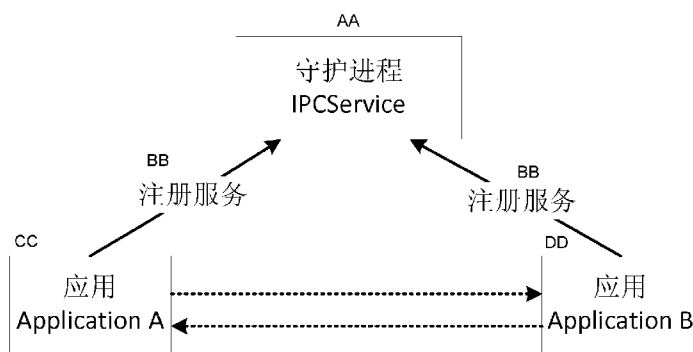


图 1 / FIG.1

AA DAEMON PROCESS IPC SERVICE
BB REGISTRATION SERVICE

CC APPLICATION A
DD APPLICATION B

(57) Abstract: Disclosed in the present invention are an inter-process communication method based on an application layer and a basic application communication system. In accordance with a client/server communication mode, frame encapsulation applications for the mutual communication between system-level applications and user applications are created and virtualized as a client side; a daemon process IPC Service is created and virtualized as a server side; registration and binding of the IPC Service are completed through the applications, so as to achieve mutual communication among every application. Through the adoption of the technical solution of the present invention, a special communication mechanism can be rapidly and safely built during the inter-process communication of every application, which not only brings more convenience to the communication of system applications, but also imparts the system with simplicity and openness so that third-party applications can be easily and conveniently integrated and free developers are also given a great convenience in developing applications on the system platform; in this way, not only are the applications of the system platform enriched, but the cost of application development can also be lowered.

(57) 摘要:

[见续页]

WO 2015/135221 A1



IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, **本国际公布:**

RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD,
TG)。

— 包括国际检索报告(条约第 21 条(3))。

本发明公开了一种基于 Android 的应用层的进程间通信方法及基础应用通信系统，按照客户机/服务器的通信模式，创建系统级应用和用户应用相互通信的框架封装 Application，并虚拟为客户机端；创建守护进程 IPC-Service，并虚拟为服务器端；通过 Application 向 IPCService 注册绑定，实现各个 Application 的互相通信。采用本发明的方案，在各个应用进行进程间通信时，可以快速安全地建立起自己的一套通信机制，不仅各系统应用的通信更加方便，而且让系统具有了简单性和开放性，可以简便地融合第三方应用，也极大方便了自由开发者在系统平台上开发应用，既丰富了系统平台的应用，也可以降低应用开发成本。

基于 Android 的应用层的进程间通信方法及基础应用通信系统

技术领域

本发明涉及数据通信领域，特别涉及一种基于 Android 的应用通信技术。

背景技术

随着科技的迅猛发展,以智能手机为代表的 Android 设备如雨后春笋般迅速发展壮大。Android 系统自推出以来,就以明显的优势逐渐扩大市场份额,在国内,Android 系统的应用已不仅局限于手机产业,几年来其迅速扩张到相邻领域,例如平板电脑、车载系统、电视 STB、智能电器、智能会议系统等。

但是对于采用 Android 系统的各个开发厂商和互联网企业而言,不能仅仅满足于在 Android 的原生系统上开发出各种各样的应用,还需要深层次地定制带有自己特色的 Android 系统;尤其对于各大电视厂商来说,独具特色的系统显得非常重要,而且也可以快速适应芯片的迭代和更新。那么,如何定制和创建一套上层开发可用的系统,成了各大厂商开发的首要任务和难题。

要创建一套基于 Android 的系统级应用,或者说需要定制一套有自己特色的 Android 系统,那么不可避免要用到进程间通信等通信方法。众所周知,在 Android 系统上,各个应用的通信方法无非两种, Binder 机制和 SOCKET 通信机制。当前,大多数电视厂商都采用了单一的 Binder 机制或 SOCKET 机制,甚至广播机制。但是这些做法都有一定的缺陷,不能完全满足一个基于可开放平台的系统要求。

采用 SOCKET 机制之后,可以降低各个资源模块的耦合性,独立的模块可以实现独立的资源接口,这样可以在某个模块崩溃之后,系统仍然可以正常运行。但是,由于 SOCKET 接口的通用性,导致其传输效率低,开销大,不适合用来多进程之间的相互实时通信,更适合不同机器间的数据传输和通信。同时,采用了 SOCKET 机制之后,由于每个模块都需要一个端口号,才能互相通信,这样在第三方应用集成进来的时候,需要分配端口号,各个系统应用如果需要和第三方应用通信,还得先后去各个应用的端口号之后才能通信,很大程度上阻碍了软件平台的开发和丰富的第三方应用的接入。所以采用 SOCKET 机制来建立系统

级应用的机制存在明显缺陷。

而采用 Binder 机制之后,由于单一的采用 Binder 机制和 AIDL 的调用方式,上层应用为了可以正常调用到所有可用的资源接口,将所有的模块功能捆绑在了一个 service 进程上,这样就导致了一个问题,系统的耦合性太强,如果核心提供资源接口的 Service 进程挂掉,那么整个系统将不能运作,给用户带来很大的不便,用户体验较差。

发明内容

本发明的主要目的是:提供一种基于Android的应用通信方案。

为此,首先,本发明提出一种基于Android的应用层的进程间通信方法,创建系统级应用和用户应用相互通信的框架封装Application,并虚拟为客户机端;创建守护进程IPCService,用于进程间通信服务,并虚拟为服务器端;所述Application向IPCService注册绑定,各个所述Application通过所述IPCService实现各个应用的互相通信。

其次,本发明提出了一种基于Android的基础应用通信系统,包括:

Application,为系统级应用和用户应用相互通信的框架封装,虚拟为客户机端;

守护进程IPCService,用于进程间通信服务,虚拟为服务器端;

所述Application向IPCService注册绑定,各个所述Application通过所述IPCService实现各个应用的互相通信。

再次,本发明提出了一种智能终端,应用上述的基于 Android 的应用层的进程间通信方法或基础应用通信系统,实现各个 Application 的互相通信。

同时,本发明提出一种智能电视终端,固化有上述的基于 Android 的基础应用通信系统。

最后,本发明提出了一种电子数据载体,存储有用于实现上述的基于 Android 的应用层的进程间通信方法的计算机程序。

本发明 Android 应用的通信方案,由具体的应用和一个守护进程组成,在 Android 系统各个应用进行进程间通信时,可以快速安全地建立起自己的一套通信机制,不仅各系统应用的通信更加方便,而且让系统具有了简单性和开放性,可以简单方便的融合第三方应用,也极大方便了自由开发者在系统平台上开发应

用，这样既丰富了系统平台的应用，也可以降低应用开发成本。

本发明基于 Android 应用的通信方案，在 Android 系统上各个应用间快速安全地建立起自己的一套通信机制，并且实现了不仅各个应用进程间的通信，更可以精准到应用的某个 Activity 或者 Service，让进程间通信不再那么神秘和应用困难。而且，基于本方案实现的通信系统上开发应用会显得非常简单，并且可以实现应用不再是独立的一个 Android 应用，可以形成一个体系，一个套件的形式展现给用户，更简洁有序，用户体验不会感觉到应用很多、很杂乱。

【附图说明】

图1 本发明的基础应用通信系统的系统通信概要图；

图2 本发明中Binder机制示意图；

图3 本发明实施例中Application绑定IPCService服务的流程；

图4 本发明实施例的注册机制示意图；

图5 本发明实施例的数据包结构示意图；

图6 本发明实施例的接收数据流程示意图；

图7 本发明实施例的发送数据流程示意图；

图8 本发明实施例的Activity和Service的通信流程图。

【具体实施方式】

下面通过具体的实施例并结合附图对本发明作进一步详细的描述。

技术名词解释：Application 主要指某个具体的应用或者 Android 的 Application 对象，本申请中，主要指系统级（即能在其上开发用户应用的基础应用）应用和用户应用相互通信的框架封装。

Application 层，泛指应用层，包含 Application 和 IPCService。

IPCService，守护进程，用于进程间通信服务。

Activity Android 应用程序组件，提供一个屏幕，用户可以用来交互。

Service 服务，Android 组件之一，没有用户界面的在后台运行执行耗时操作的应用组件。

ServiceManager 服务管理，Android 的重要进程，用于管理系统中的 Service。

本发明采用 C/S (Client/Server, 客户机/服务器) 的通信模式, 总体框架请参阅图 1 的系统通信概要图, 由 Application 和 IPCService 两部分组成, 通过 Application 向 IPCService 注册, 然后实现各个 Application 互相通信的过程。

实施例一:

本例的基于 Android 的应用层的进程间通信方法, 通过创建一个基于 Android 的基础应用 (泛指所有系统应用和用户应用) 通信系统来实现, 包括:

首先, 创建系统级 (即能在其上开发用户应用的基础应用) 应用和用户应用相互通信的框架封装应用 Application, 实现绑定服务、保存对象、发送数据、接收数据等主要重要步骤。

1. 绑定服务, 主要指 Application 层在 Activity 和 Service 在 oncreated() (7 个与生命周期有关的函数之一, 是 activity 第一次启动时 being 执行的, 主要是初始化一些变量) 的时候, 向 ServiceManager 发送申请 IPCService 服务的请求, 请结合图 2 和图 3 所示, 利用了 Android 的 Binder 机制;

S11. IPCService 申请创建 Binder 的实体: 首先, IPCService 在自己的进程中向 Binder 驱动申请创建一个 IPCService 的 Binder 的实体;

S12. Binder 驱动创建 IPCService 的服务: Binder 驱动为 IPCService 创建为了内核中的 IPCService 的 Binder 实体节点以及 Binder 的引用, 实际上是通知 ServiceManager 注册一个叫做 IPCService 的服务, 用于进程间通信服务;

S13. 关联查找表: ServiceManager 收到数据包之后, 从中取出 IPCService 的名字和引用, 填入一张查找表中;

S14. Application 申请并获得 IPCService 的绑定服务: Application 向 ServiceManager 发起申请 IPCService 服务的请求, 那么 ServiceManager 就可以在查找表中找到 IPCService 的引用, IPCService 把 Binder 引用返回给 Application。

2. 保存对象, 如图 4 注册机制中的虚线框中所示, Application 层在绑定服务成功之后, 将各个继承自 SkyActivity 和 SkyService 的实例对象保存在一张对应表中, 以便接收数据的时候可以分发到各个具体的 Activity 和 Service 对象。通过保存具体的实例对象, 就可以准确传输数据到指定应用的指定 Activity 和指定 Service 中, 实现了精准传输通信的逻辑结构。

本例采用的两张表：具体来说就是，每个 Application 都保存了一张对应表，该表是自身 Service 和 Activity 的集合；而 IPCService 的查找表，一个通信平台一个，保存该平台的所有 Application 的引用等。这在现有技术的方案中是完全没有的。现有技术中，Android 自身虽然可以实现 Activity 到 Activity 或者 Service 的通信，但是其缺点过于明显，实现的是进程对进程的通信，采用进程对进程的一张表，如果一个进程中有多个 service 和 activity，就没法实现具体到哪个 service 或者 activity 的通信，故只能实现点对点之间的通信，通信的数据只能是基本类型的数据结构（int,String,List 等）；而本例保存了具体的实例对象后，相当于将所有的应用置于了一个通信平台，不仅仅是点对点，还可以多个点之间互相通信。

现有技术中，利用 Binder 在进程间通信，限定了返回的引用必须是 IBinder，应用 Application A 如果要和应用 Application B 通信，就得保存 Application B 的 IBinder，Application A 如果还要同 Application C 通信，那么 Application A 也得保存 Application C 的 IBinder，Application A 要通信的点越多，导致 Application A 需要绑定的次数也越多，保存的引用也越多，并且这些 Application 之间如果要互相通信，那么每个应用都得保存一份除自身之外的所有引用，非常繁琐复杂。并且受 Binder 机制的接口约束，通信数据的结构也局限于 Android 通信的基本数据结构，基本都是采用 String 进行通信。

Binder 机制的核心就是两个进程或者两个组件的通信，采用本案的封装和利用后，可以面向一个平台，并且大大减少了上层应用开发的复杂性，但同时又实现了上层应用的多点互相通信。本案直接采用字节流之后，用户就可采用任何数据结构，只需要确保通信的双方可以正常序列化和反序列化即可，并且 Java 的很多复杂数据结构也支持字节序列化和反序列化，大大减少了数据结构向 String 转换的性能消耗。

3.发送数据，由 Application 将要通信的数据封装成如图 5 所示的数据结构中，主要由包头 header 和数据体 body 组成，header 主要包括了发送方身份信息 from、发送目的地 to、通信命令 cmd、优先级 priority、回复 needack 等字段；body 主要是消息的字节流；header 的各个字段的含义如下：

3.1.from，主要包含发送数据方的包名，Activity 或者 Service 名，由 Application 层自动组装和获取，具体的使用者可以不用关心；

3.2.to, 主要指数据要发送到某个包中的某个 Activity 或者 Service, 由具体实现发送命令的使用者传递参数;

3.3.cmd, 主要是双方用来通信的协议, 代表某一协商好的含义, 具体为一个字符串;

3.4.priority, 主要是指命令的优先级, 在 IPCService 处理命令的时候区分优先级;

3.5.needack, 是否需要立即回复的标识, 表示该命令是否需要立即返回一个 result(通信的一个标签);

在 Application 组装好数据之后, 由于绑定服务已经获取到了 IPCService 的 Binder 的引用, 就可以直接将组装好的数据包发送给 IPCService, 然后由 IPCService 处理和转发。而且, 本案采用了字节流的形式对数据进行再次编码和压缩, 在 Application 对外开放的通信接口上, 应用如果需要通信, 直接传递字节流即可; 而现有技术大多对外接口都是字符串, 或者一种协商好的数据结构, 不够通用和灵活。字节是基本的传输单位, 用户 (也就是采用本通信的使用者) 有权决定自身的数据结构和序列化方式, 如果采用一种固定的数据结构限制了用户对于序列化等操作的要求; 面向用户开放字节流之后, 用户就可以自己根据自己的需求决定采用哪种方式序列化, 采用哪种方式反序列化, 不必限定于非要采用 Java 的序列化, 或者 Android 的序列化, 或者某种自定义的序列化等, 让使用者可以采用的数据结构更加多样化, 不再局限于基础的数据结构, 打破了固定化数据结构的通信模式。

4.接收数据, 如图 6 接收数据机制所示, 在 Application 层接收到数据包以后, 会将接收到的数据包丢入一个命令池里面, 该命令池是对 Java 现有线程池的封装, 主要实现对于请求的控制处理; 该命令池专用来处理接收到的数据包的, 有一个线程会不断的从该命令池中取数据包, 解析数据包的头部, 然后根据具体的 Activity 名或者 Service 名, 从一开始保存的对应表中查询出具体的 Activity 实例, 然后交给具体的 Activity 实例去处理具体的数据包。

其次, 创建一个守护进程 IPCService, 作为处理和转发数据包的核心服务, 虽然从 Android 的架构角度来讲处于用户空间, 但是对于用户来说是不可见的, 用户和使用者不必关心 IPCService 需要具体处理哪些事务。其主要包含两个功

能：

一方面，IPCService 管理所有 Application 提供的服务，即所有 Application 接收数据的 Binder 引用；IPCService 在管理所有 Application 服务的同时，保存了所有通信应用的包名，将应用的包名和 Binder 引用一一对应起来形成一查找表，实现了任何包对包的通信。这样极大方便了第三方应用的集成和开发，因为此时不再局限于已经存在的系统应用的通信，同时将范围大大扩展到了任何采用本框架的应用，不管是自行开发的，还是第三方集成的。

另一方面，IPCService 响应所有 Application 的请求并为之分配对应的服务。IPCService 在接收到 Application 的请求时，根据包名取出所请求的 Binder 引用，然后调用该 Binder 的接口，转发数据。

这样就实现了一个发送数据的流程，如图 7 所示：

假设 Application A 向 Application B 发送数据，当 IPCService 接收到 Application A 的请求时，处理发送到的数据包的头，解析出 to 的对象，也就是 Application B，然后从保存的查找表中取出 Application B 的 Binder 引用，然后调用 Application B 的引用，这样数据就发送到了 Application B。如果查找表中没有发送到的对象，说明 Application B 没有启动，就可以直接丢弃数据包，因为所有继承自该套通信机制的应用都会在启动的时候向 IPCService 发起注册和绑定服务。

采用本方案提出的通信机制之后，Application 的请求便于管理，另一方面，只需要 Application 启动的时候，向 IPCService 建立连接即可，就可以花费非常少的时间去实现 IPCService 的功能。

综上所述，本例的基于 Android 系统级应用的通信方案，由具体的应用和一个守护进程组成，解决了在 Android 系统上各个应用进行进程间通信的一些弊端，可以快速安全地建立起自己的一套通信机制，并且不仅实现了各个应用进程间的通信，更可以精准到应用的某个 Activity 或者 Service，让进程间通信不再那么神秘和应用困难。最重要的是，在基于本方案实现的通信系统上开发应用会显得非常简单，并且可以实现应用不再是独立的一个 Android 应用，可以形成一个体系，一个套件的形式展现给用户，让用户不会感觉到应用很多、很杂乱。基于本方案实现的通信系统上，可以让一个独具特色的系统更加具有了简单性、开放性

以及非常好的可扩展性，不再是一个单一的系统，而是一个可开放的平台。

实施例二：

依据实施例一的方案，将其具体应用在创维的智能电视操作系统-天赐系统的重构版本中，详细说明如下：

首先，实现一个全局的 Application，包含在框架层，并且封装成 jar 包。编写一个全局的单例模式的 MyApplication 继承自 Application。由于在应用层无法直接对原生 Application 进行更改，更改也违背了 Android 开放性和通用性的要求；创建一个继承自 Application 的 MyApplication 是为了实现对事物的普通性和特殊性的需求，再次创建的 MyApplication 既是一个普通的 Android 应用，但同时又实现了自身对于通信的需求。MyApplication 实现通信框架层的封装，主要包含绑定服务、保存对象、发送数据、接收数据等重要步骤。各个步骤的具体实施方案同实施例一所述。如图 4，完整地表示了整个 Application 层通信的处理流程：Activity 和 Service 在 onCreate() 的时候，被初始化并被 Application 保存在一张对应表。假如应用 A 中有 2 个 Activity，3 个 Service，那么保存的表对应关系如下所示，该应用所有的 Activity 和 Service 均被保存在两个 HashMap 中，形成一个 Application 的对应表，包含有该应用所有基础组件的包名和类名 (key 和 value 的顺序可以调换)。

Value	key	Application
Activity 1	Activity 1 的类名	Application
Activity 2	Activity 2 的类名	
Service 1	Service 1 的类名	

该表记录了该进程所有创建的需要通信的 Activity 和 Service；在 Activity 和 Service 初始化的时候，Application 将自身的实例包名作为参数，向 ServiceManager 发送申请 IPCService 服务的请求；ServiceManager 在接收到请求之后，将 IPCService 的引用返回给 Application；并且与此同时，IPCService 会将绑定者（也就是 Application）的引用保存在一张查找表中，该表包括绑定者的包名、提供的服务等信息。如下，一个由包名对应 binder 应用返回的 ICmdReceiver 接口实例组成的 hashmap<Strng, ICmdReceiver>，形成 IPCService 中的查找表。

Key	Value
com.test.test1	Test1 应用的 ICmdReceiver

	实现
com.test.test2	Test2 应用的 ICmdReceiver 实现
com.test.test3	Test3 应用的 ICmdReceiver 实现

之所以在应用层实现通信框架的封装，是因为在 Android 基础上，应用程序第一个组件运行时将会为应用创建一个进程，并且首先实例化所指定的 Application，此时就会执行绑定 IPCService 的服务，从而建立 Application 和 IPCService 的通信；

然后，实现 IPCService 的守护进程：

创建一个守护进程的 service，提供注册、发送数据的 Binder 引用，再创建一张保存所有已经注册的 Application 的 Binder 查找表，最后根据数据包的包头，从查找表查找获取对应 Binder 的引用来实现数据转发。

上述步骤实现了基本通信的框架系统，那么对于具体的各个基础应用或者用户应用来说，要做的事情就更简单了，只需要三个简单的步骤，即可实现整个系统各个 Activity 和 Service 的通信。

S1，所创建的应用依赖系统框架 jar 包；

S2，所有需要通信的 Activity 和 Service 继承自 SkyActivity 和 SkyService；

S3，在 androidmanifest.xml 中配置 Application 选项，其中 name 后面为框架 jar 包中 Application 的路径，如下所示：

```
<application android:name="a.b.c.MyApplication" ></application>
```

通过以上三个简单的步骤，就可以在任意继承自 SkyActivity 和 SkyService 的类中处理和收发数据包，并且可以实现阻塞和非阻塞两种调用方式。阻塞调用方式，主要流程是将命令和数据发送到接收方，同时同步等待接收方处理的返回，此时发送方的线程被阻塞，直到调用的返回。非阻塞调用的方式，将命令和数据发送到接收方时，命令先由框架中的线程处理，发送方不会等待命令的返回；框架中的线程池会根据命令的优先级和时间将线程池中的所有命令按照优先顺序依次交给接收方处理。

综上所述，本例在天赐系统的重构和优化过程中得到了应用，同时本发明提出的方案不仅仅可以应用于电视上，任何一个基于 Android 的系统都可以采纳本

方案进行进程间通信以及系统的深入开发与体系开发。而且，本方案在 Android 系统通信机制的基础上进行了再次封装和创新，让应用的开发更加简单，也让针对不同平台的应用差异化工作更容易开展，做到了让平台简单化，统一化，应用多元化的目标。总之，本方案实现了系统应用通信的简单性、开放性。

实施例三

本例是一种智能终端，其中内置了实现实施例一中的基于 Android 的应用层的进程间通信方法或基础应用通信系统的程序，通过启用该程序，可实现智能终端中各个 Application 的互相通信。实现了系统应用通信的简单性、开放性，易于第三方进行应用开发，实现应用多元化的目标。智能终端包括但不限于智能电视、智能手机、个人电脑、平板电脑或云系统个人终端等。

实施例四

本例是一种智能电视终端，其中固化有实现实施例一中的基于 Android 的应用层的进程间通信方法或基础应用通信系统的程序，通过启用该程序，可实现智能终端中各个 Application 的互相通信。在此基础应用通信系统的基础上，易于第三方进行用户应用开发，实现应用多元化的目标。比如，第三方开发者甚至不用理解该框架，只需要两个步骤即可调用系统的一些接口。比如目前创维天赐系统中的爱奇艺、有朋等第三方应用，可以非常容易和快速地实现不同的应用的同一个云端收藏等功能。

实施例五

本领域技术人员可以理解，上述实施方式中各种方法的全部或部分步骤可以通过计算机程序来指令相关硬件完成。本例是一种计算机可读的电子数据载体，包括但不限于光盘、闪存盘、硬盘和网盘等，其中存储了用于实现实施例一中的基于 Android 的应用层的进程间通信方法或基础应用通信系统的计算机程序。可供基于 Android 的智能终端进行操作系统的个性化升级，也可供第三方进行应用开发。智能终端包括但不限于智能电视、智能手机、个人电脑、平板电脑或云系统个人终端等。

以上内容是结合具体的优选实施方式对本发明所作的进一步详细说明，不能

认定本发明的具体实施只局限于这些说明。对于本发明所属技术领域的普通技术人员来说，在不脱离本发明构思的前提下，还可以做出若干简单推演或替换，都应当视为属于本发明的保护范围。

1.一种基于Android的应用层的进程间通信方法，创建系统级应用和用户应用相互通信的框架封装Application，并虚拟为客户机端；创建守护进程IPCService，用于进程间通信服务，并虚拟为服务器端；所述Application向守护进程IPCService注册绑定，各个所述Application通过所述守护进程IPCService实现各个应用的互相通信。

2.如权利要求1所述的应用层的进程间通信方法，其特征是：具体包括：

所述框架封装Application，实现绑定服务、保存对象、发送数据和接收数据步骤；

所述守护进程 IPCService，管理所有所述 Application 服务，保存所有通信应用的包名，将应用的包名和 Binder 引用一一对应，实现任何包对包的通信；响应所有所述 Application 的请求并为之分配对应的服务，在接收到所述 Application 的请求时，根据包名取出所请求的 Binder 引用，然后调用该 Binder 的接口，转发数据。

3.如权利要求2所述的应用层的进程间通信方法，其特征是：

所述绑定服务步骤中，所述Application在Activity和Service在oncreated()时，向ServiceManager发送申请守护进程IPCService服务的请求，利用Binder机制，由ServiceManager绑定所述Application和守护进程IPCService之间的服务机制。

4. 如权利要求3所述的应用层的进程间通信方法，其特征是：所述绑定服务的具体流程包括：

S11. 守护进程IPCService在自己的进程中向Binder驱动申请创建一个守护进程IPCService的Binder的实体；

S12. Binder驱动为守护进程IPCService创建成为内核中的守护进程IPCService的Binder实体节点以及Binder的引用，即通知ServiceManager注册一个用于进程间通信服务的守护进程IPCService；

S13. ServiceManager收到数据包之后，从中取出IPCService的名字和引用，填入查找表中；

S14. Application向ServiceManager发起申请守护进程IPCService服务的请求，ServiceManager在所述查找表中找到守护进程IPCService的引用，并把Binder引用返回给Application。

5. 如权利要求3或4所述的应用层的进程间通信方法，其特征是：所述保存对象步骤中，Application在与守护进程IPCService绑定服务成功之后，将各个继承自某一Activity和某一Service的实例对象保存在对应表中，以便接收数据的时候可以分发到各个具体的Activity和Service对象。

6. 如权利要求3或4所述的应用层的进程间通信方法，其特征是：所述发送数据的具体流程包括：由Application将要通信的数据封装成包括包头header和数据体body的数据结构中；将组装好的数据包发送给守护进程IPCService，然后由守护进程IPCService处理和转发。

7. 如权利要求5所述的应用层的进程间通信方法，其特征是：所述接收数据步骤中，Application接收到数据包以后，将接收到的数据包放入一个用于处理接收到的数据包的命令池里，由一个线程从该命令池中取数据包，解析数据包的头部，根据具体的Activity名或者Service名，从所述对应表中查询出具体的Activity实例，然后交给具体的Activity实例去处理具体的数据包。

8. 如权利要求5所述的应用层的进程间通信方法，其特征是：Application A向Application B发送数据的具体流程包括：Application A向守护进程IPCService发出服务请求，守护进程IPCService接收到Application A的请求时，处理发送到的数据包的头部，解析出被发送的对象Application B，然后从保存的调用表中取出Application B的Binder引用，调用Application B的引用，数据被发送到了Application B。

9. 如权利要求6所述的应用层的进程间通信方法，其特征是：所述包头header包括发送方身份信息、发送目的地、通信命令、优先级、回复字段，

发送方身份信息，包含发送数据方的包名，Activity或者Service名，由应用层自动组装和获取；

发送目的地，指数据要发送到某个包中的某个 Activity 或者 Service，由具体实现发送命令的使用者传递参数；

通信命令，代表某一协商好的、双方用来通信的协议；

优先级，命令的优先级，区分守护进程 IPCService 处理命令的优先级；

回复，是否需要立即回复的标识。

10. 一种基于Android的基础应用通信系统，包括：

Application，为系统级应用和用户应用相互通信的框架封装，虚拟为客户机端；

守护进程IPCService，用于进程间通信服务，虚拟为服务器端；

所述Application向守护进程IPCService注册绑定，各个所述Application通过所述守护进程IPCService实现各个应用的互相通信。

11. 如权利要求10所述的基础应用通信系统，其特征是：所述框架封装Application，用于实现绑定服务、保存对象、发送数据和接收数据步骤；

所述守护进程 IPCService，管理所有 Application 服务，保存所有通信应用的包名，将应用的包名和 Binder 引用一一对应，实现任何包对包的通信；响应所有 Application 的请求并为之分配对应的服务，在接收到 Application 的请求时，根据包名取出所请求的 Binder 引用，然后调用该 Binder 的接口，转发数据。

12. 如权利要求11所述的基础应用通信系统，其特征是：

所述绑定服务步骤中，Application在Activity和Service在oncreated()时，向ServiceManager

发送申请IPCService服务的请求，利用Binder机制，由ServiceManager绑定所述Application和IPCService之间的服务机制。

13. 如权利要求12所述的基础应用通信系统，其特征是：所述绑定服务的具体流程包括：
S11.守护进程IPCService在自己的进程中向Binder驱动申请创建一个守护进程IPCService的Binder的实体；

S12.Binder驱动为守护进程IPCService创建成为内核中的守护进程IPCService的Binder实体节点以及Binder的引用，即通知ServiceManager注册一个用于进程间通信服务、称为守护进程IPCService的服务；

S13.ServiceManager收到数据包之后，从中取出守护进程IPCService的名字和引用，填入查找表中；

S14.Application向ServiceManager发起申请守护进程IPCService服务的请求，ServiceManager在所述查找表中找到守护进程IPCService的引用，并把Binder引用返回给Application。

14. 如权利要求12或13所述的基础应用通信系统，其特征是：所述保存对象步骤中，Application在与守护进程IPCService绑定服务成功之后，将各个继承自某一Activity和某一Service的实例对象保存在对应表中，以便接收数据的时候可以分发到各个具体的Activity和Service对象。

15. 如权利要求12或13所述的基础应用通信系统，其特征是：所述发送数据的具体流程包括：由Application将要通信的数据封装成由包头header和数据体body组成的数据结构中；将组装好的数据包发送给守护进程IPCService，然后由守护进程IPCService处理和转发。

16. 如权利要求14所述的基础应用通信系统，其特征是：所述接收数据步骤中，Application接收到数据包以后，将接收到的数据包放入一个用于处理接收到的数据包的命令池里，由一个线程从该命令池中取数据包，解析数据包的头部，根据具体的Activity名或者Service名，从所述对应表中查询出具体的Activity实例，然后交给具体的Activity实例去处理具体的数据包。

17. 如权利要求14所述的应用通信系统，其特征是：Application A向 Application B发送数据的具体流程包括：Application A向守护进程IPCService发出服务请求，守护进程IPCService接收到Application A的请求时，处理发送到的数据包的头部，解析出被发送的对象Application B，然后从保存的调用表中取出Application B的Binder引用，调用Application B的引用，数据被发送到了Application B。

18. 如权利要求15所述的基础应用通信系统，其特征是：所述数据头header包括发送方身份信息、发送目的地、通信命令、优先级、回复字段，

发送方身份信息，包含发送数据方的包名，Activity或者Service名，由Application层自动组装和获取；

发送目的地，指数据要发送到某个包中的某个 Activity 或者 Service，由具体实现发送命令的使用者传递参数；

通信命令，代表某一协商好的、双方用来通信的协议；

优先级，命令的优先级，区分守护进程 IPCService 处理命令的优先级；

回复，是否需要立即回复的标识。

19.一种智能终端，应用如权利要求 1-9 中任一项所述的基于 Android 的应用层的进程间通信方法，实现各个 Application 的互相通信。

20.一种智能电视终端，固化有如权利要求 10-18 中任一项所述的基于 Android 的基础应用通信系统。

21. 一种电子数据载体，存储有用于实现如权利要求 1-9 中任一项所述的基于 Android 的应用层的进程间通信方法的计算机程序。

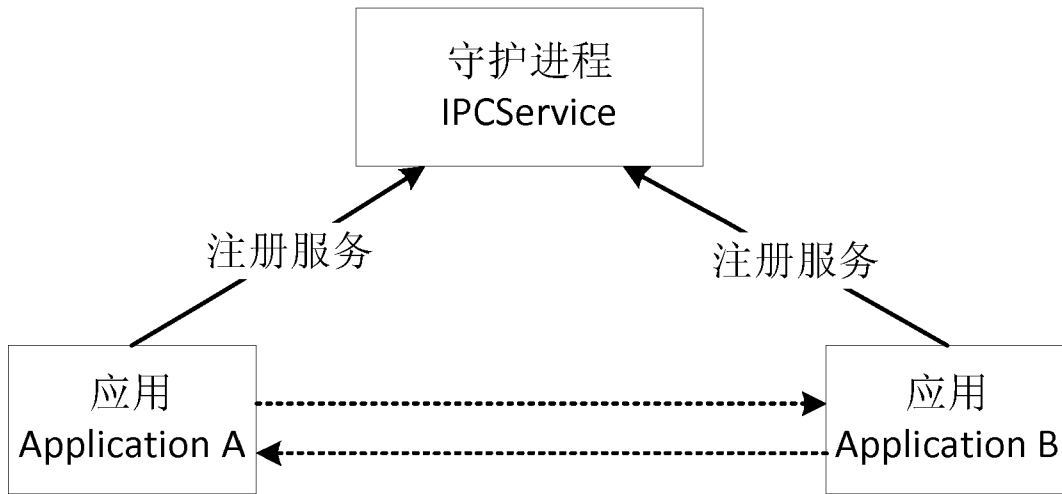


图 1

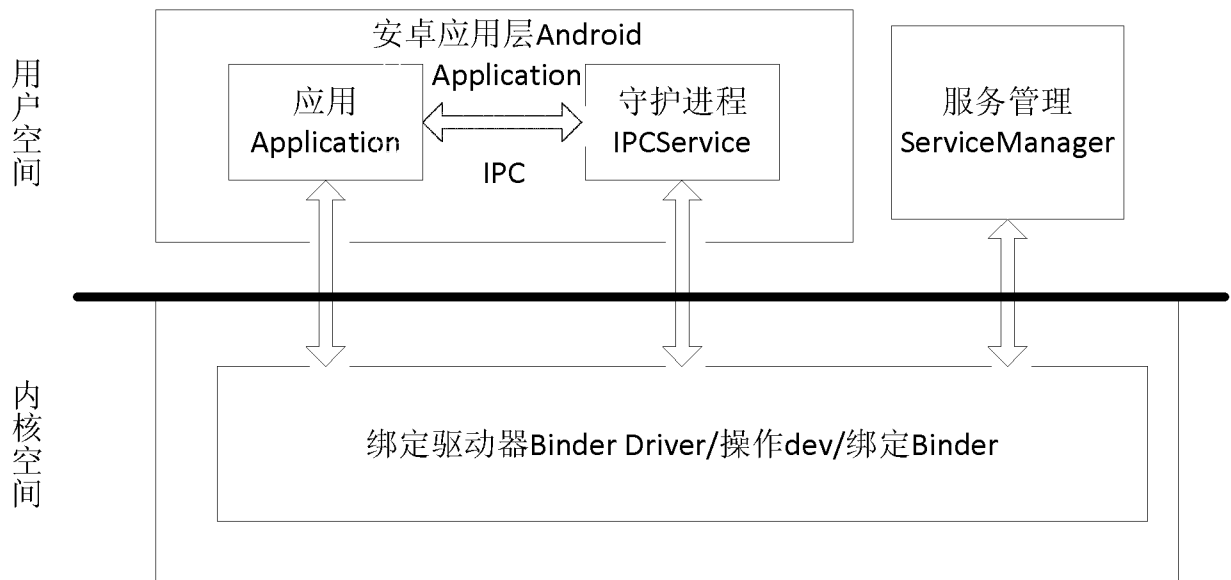


图 2

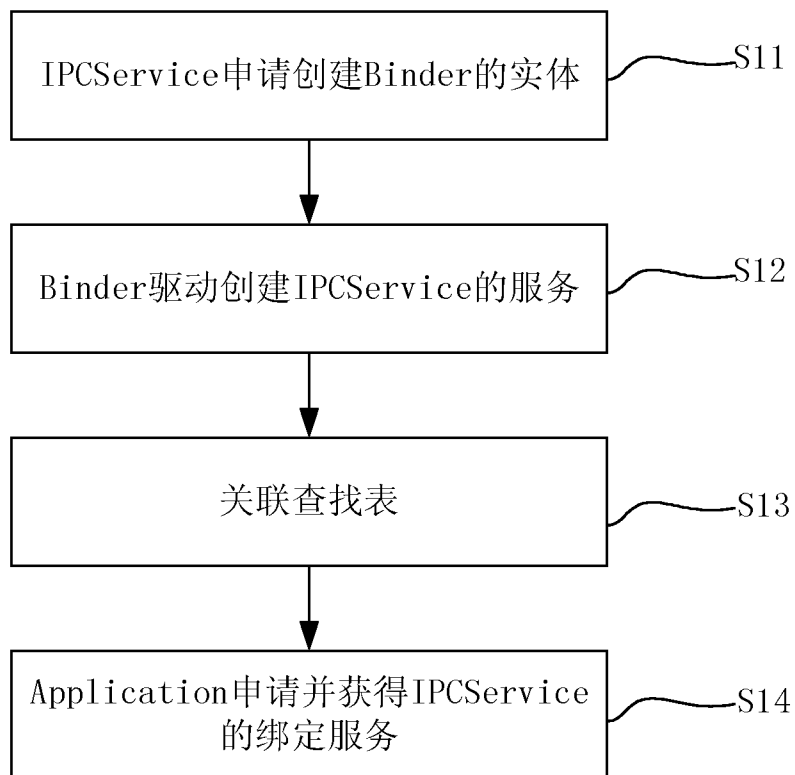


图 3

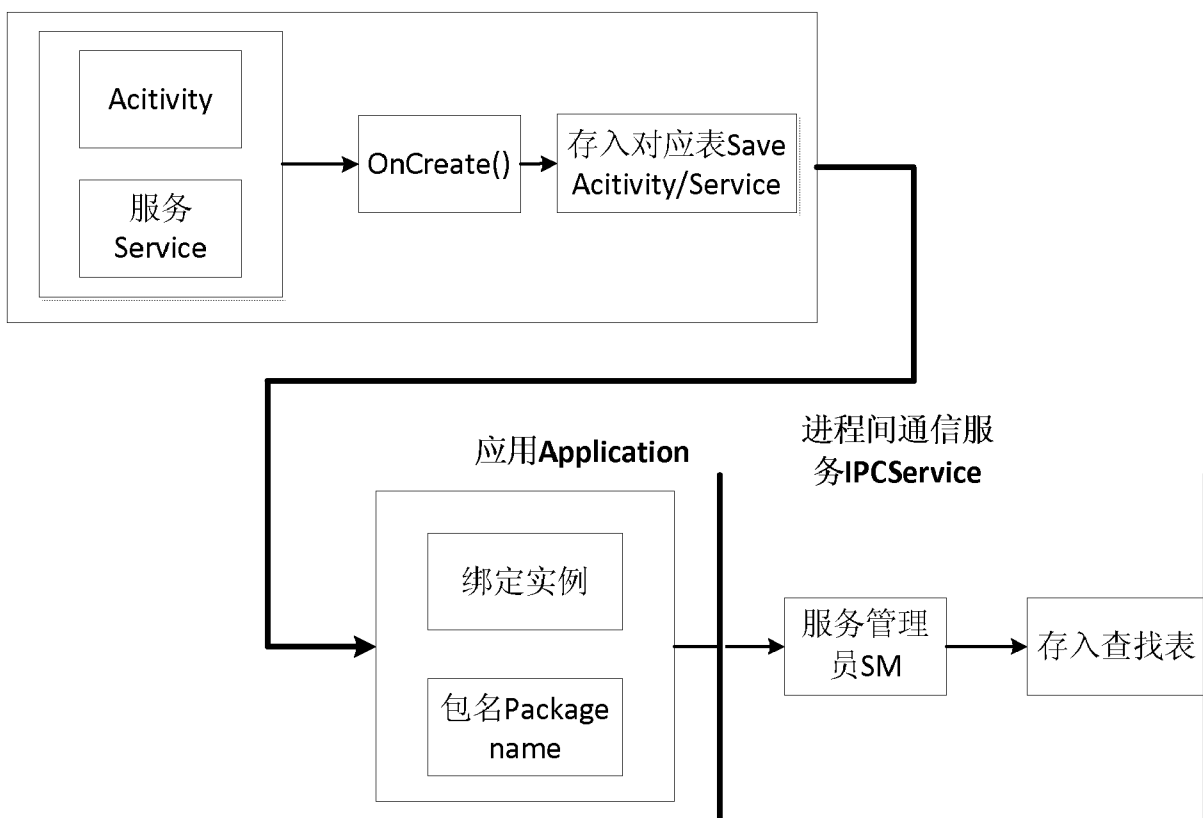


图 4

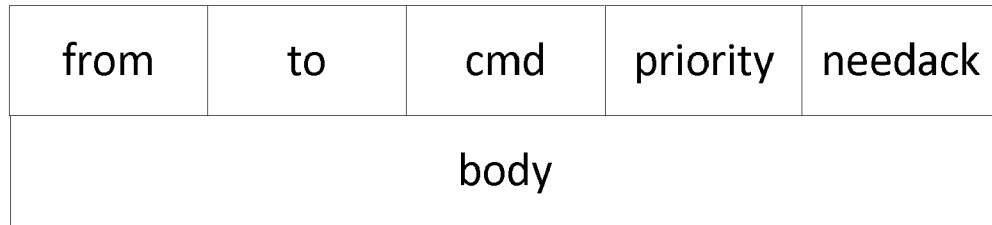


图 5

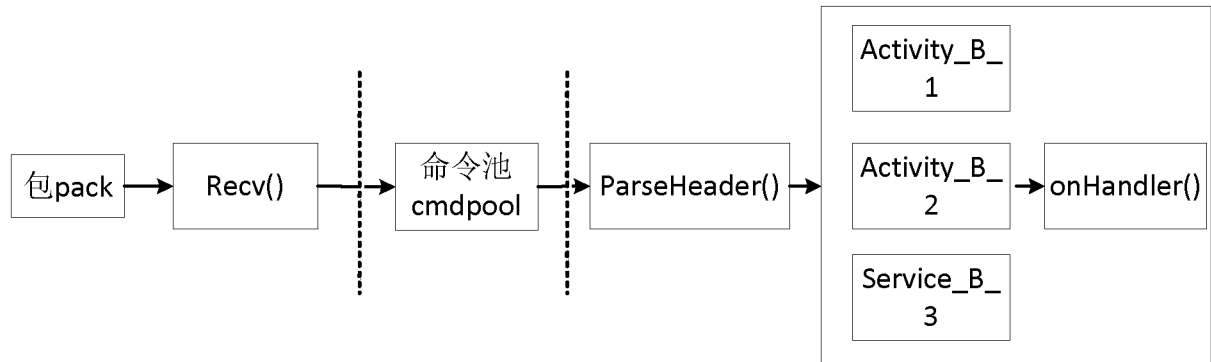


图 6

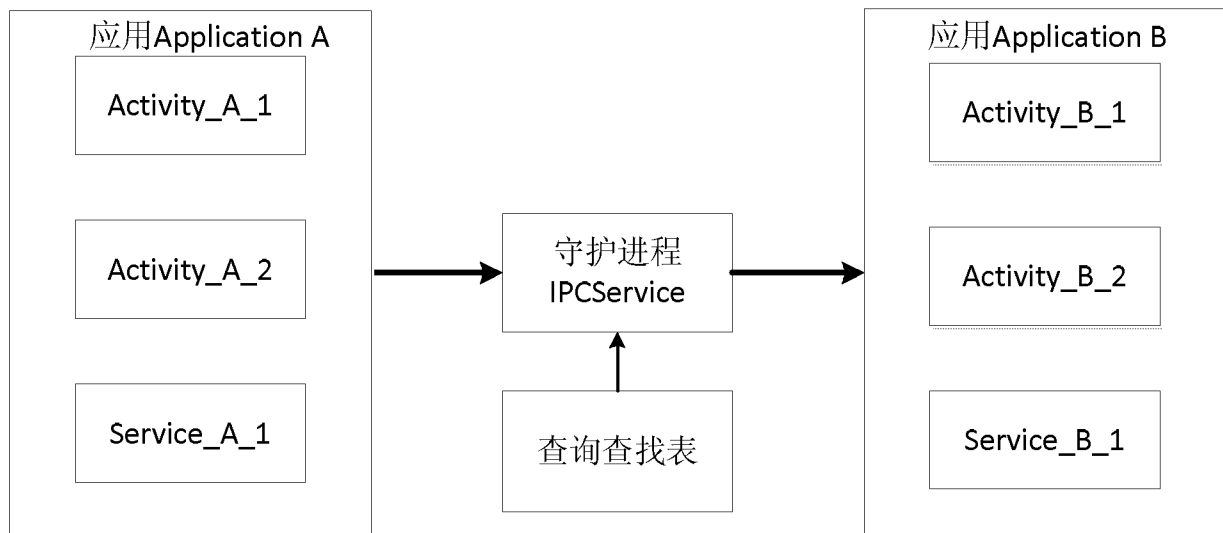


图 7

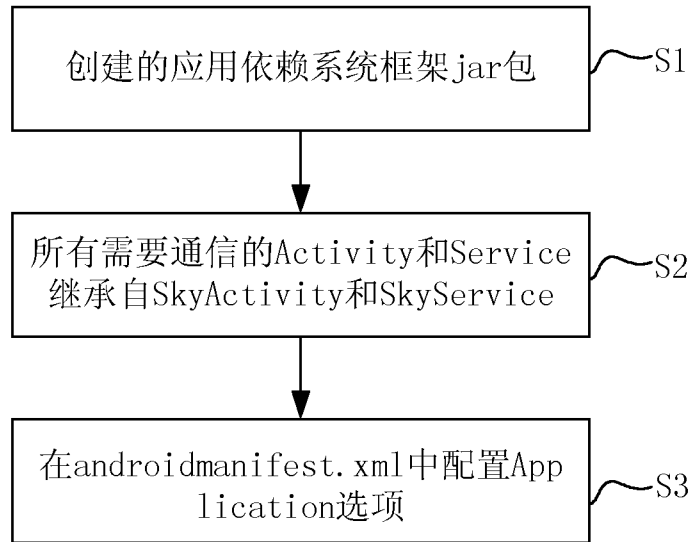


图 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2014/073496**A. CLASSIFICATION OF SUBJECT MATTER**

H04L 12/18 (2006.01) 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)

H04L; H04W

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)

WPI; EPODOC; CNKI; CNPAT: packaging, daemon, register, IPCService, client, server, virtual, binding

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 102833080 A (KONKA GROUP CO., LTD.), 19 December 2012 (19.12.2012), description, paragraphs [0005]-[0014] and [0023]-[0045]	1-21
A	CN 101141742 A (ZTE CORP.), 12 March 2008 (12.03.2008), the whole document	1-21
A	WO 2011032453 A1 (EVOC INTELLIGENT TECHNOLOGY CO., LTD.), 24 March 2011 (24.03.2011), the whole document	1-21

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

* Special categories of cited documents:	“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
“A” document defining the general state of the art which is not considered to be of particular relevance	“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
“E” earlier application or patent but published on or after the international filing date	“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
“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)	“&” document member of the same patent family
“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	

Date of the actual completion of the international search 15 November 2014 (15.11.2014)	Date of mailing of the international search report 22 December 2014 (22.12.2014)
Name and mailing address of the ISA/CN: State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No.: (86-10) 62019451	Authorized officer LV, Yuan Telephone No.: (86-10) 62413388

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2014/073496

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 102833080 A	19 December 2012	None	
CN 101141742 A	12 March 2008	CN 101141742 B	10 August 2011
		WO 2009046593 A1	16 April 2009
WO 2011032453 A1	24 March 2011	CN 102023940 A	20 April 2011
		CN 102023940 B	10 July 2013

A. 主题的分类 H04L 12/18 (2006.01) 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) H04L; H04W 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)) WPI; EPDOC; CNKI; CNPAT: 封装, 注册, 守护进程, 客户机, 服务器, 虚拟, 绑定, register, IPCService, client, server, virtual, binding		
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
X	CN 102833080 A (康佳集团股份有限公司) 2012年 12月 19日 (2012 - 12 - 19) 说明书第[0005]-[0014], [0023]-[0045]段	1-21
A	CN 101141742 A (中兴通讯股份有限公司) 2008年 3月 12日 (2008 - 03 - 12) 全文	1-21
A	WO 2011032453 A1 (EVOC INTELLIGENT TECHNOLOGY CO., LTD) 2011年 3月 24日 (2011 - 03 - 24) 全文	1-21
☐ 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 2014年 11月 15日		国际检索报告邮寄日期 2014年 12月 22日
ISA/CN的名称和邮寄地址 中华人民共和国国家知识产权局 (ISA/CN) 北京市海淀区蓟门桥西土城路6号 100088 中国 传真号 (86-10)62019451		授权官员 吕源 电话号码 (86-10)62413388

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2014/073496

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	102833080	A	2012年 12月 19日	无			
CN	101141742	A	2008年 3月 12日	CN	101141742	B	2011年 8月 10日
				WO	2009046593	A1	2009年 4月 16日
WO	2011032453	A1	2011年 3月 24日	CN	102023940	A	2011年 4月 20日
				CN	102023940	B	2013年 7月 10日