

基于卡片的信息展示方法、信息展示业务的处理方法及装置

本申请要求 2016 年 08 月 03 日递交的申请号为 201610630286.0、发明名称为“基于卡片的信息展示方法、信息展示业务的处理方法及装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5

技术领域

本申请涉及计算机技术领域，尤其涉及基于卡片的信息展示方法、信息展示业务的处理方法及装置。

10 背景技术

随着互联网的蓬勃发展，越来越多的应用 APP 采用卡片流（Feed 流）的形式向用户进行信息展示。在通过 Feed 流进行信息展示的过程中，根据所要展示的信息类型的不同，通常将所要展示的信息配置为不同的类型的卡片。

然而，随着应用 APP 版本的不断更新，在新版本中通常会出现新的卡片类型。当用户所使用的应用 APP 版本未更新时，这些新类型的卡片将无法在该用户所使用的应用 APP 中展示。

针对应用 APP 无法对新类型的卡片进行展示的情况，通常通过引导用户执行应用 APP 版本更新的操作进行处理。当用户通过引导进行应用 APP 的版本更新后，将在该用户所使用的应用 APP 中展示该新类型的卡片。

20 然而，通过更新应用 APP 的版本实现新类型的卡片在应用 APP 中的展示这一方法存在：在应用 APP 服务器出现繁忙等情况导致应用 APP 更新失败时，依然无法对该新类型的卡片进行展示，使得用户对该应用 APP 的使用体验降低。

发明内容

25 本申请实施例提供基于卡片的信息展示方法、信息展示业务的处理方法及装置，用于解决现有技术中的问题。

本申请实施例提供了一种基于卡片的信息展示方法，该方法包括：

确定待展示的目标信息和用于展示所述目标信息的第一卡片的卡片类型；

30 当确定客户端的配置版本不支持所述第一卡片的卡片类型时，确定所述第一卡片对应的关联标识，并根据所述关联标识关联所述客户端的配置版本所支持的第二卡片；

利用所述第二卡片展示所述目标信息和所述客户端的版本升级提示信息。

优选的，若所述客户端包含白名单，所述白名单中包含所述客户端的配置版本所支持的卡片类型；则

确定客户端的配置版本是否支持所述第一卡片的卡片类型，具体包括：

5 将所述第一卡片的卡片类型与所述白名单中包含的卡片类型进行匹配；

根据匹配结果，确定所述客户端的配置版本是否支持所述第一卡片的卡片类型。

优选的，若所述客户端中包含所述客户端的配置版本的版本标识与所述配置版本所支持的卡片类型之间的对应关系；

确定客户端的配置版本是否支持所述第一卡片的卡片类型，具体包括：

10 确定所述客户端的配置版本的版本标识，并根据所述对应关系确定所述版本标识对应的配置版本所支持的卡片类型；

根据所述第一卡片的卡片类型和所确定的所述客户端的配置版本所支持的卡片类型，确定所述客户端的配置版本是否支持所述第一卡片的卡片类型。

优选的，根据所述关联标识关联所述客户端的配置版本所支持的第二卡片，包括：

15 根据所述关联标识，确定与所述第一卡片关联的第二卡片的卡片数据；

根据所述卡片数据，确定所述客户端的配置版本所支持的第二卡片。

优选的，确定待展示的目标信息和用于展示所述目标信息的第一卡片的卡片类型具体包括：从客户端的缓存数据获取待展示的目标信息和用于展示所述目标信息的第一卡片的卡片类型；或

20 从服务端获取待展示的目标信息和用于展示所述目标信息的第一卡片的卡片类型。

优选的，当从服务端获取待展示的目标信息和用于展示所述目标信息的第一卡片的卡片类型时，所述方法还包括：从服务端获取并缓存所述客户端的配置版本所支持的第二卡片；则，

25 根据所述关联标识关联所述客户端的配置版本所支持的第二卡片，具体包括：根据所述关联标识关联已缓存的所述第二卡片。

本申请实施例还提供一种信息展示业务的处理方法，该方法包括：

接收客户端发送的获取目标信息的业务请求，所述业务请求中包含用于确定所述客户端的配置版本的版本标识，所述版本标识与所述客户端的配置版本所支持的卡片类型之间具有对应关系；

30 当根据所述版本标识对应的卡片类型确定所述客户端的配置版本不支持用于展示所

述目标信息的第一卡片的卡片类型时，向所述客户端返回目标信息以及指示信息，所述指示信息用于指示所述客户端利用所述客户端的配置版本所支持的第二卡片展示所述目标信息和所述客户端的版本升级提示信息。

5 优选的，所述向所述客户端返回目标信息以及指示信息具体包括：以 json 字符串的形式向所述客户端返回目标信息以及指示信息。

本申请实施例还提供一种基于卡片的信息展示装置，所述装置位于客户端，包括：确定单元、关联单元以及展示单元，其中：

确定单元，确定待展示的目标信息和用于展示所述目标信息的第一卡片的卡片类型；

10 关联单元，当确定客户端的配置版本不支持所述第一卡片的卡片类型时，确定所述第一卡片对应的关联标识，并根据所述关联标识关联所述客户端的配置版本所支持的第二卡片；

展示单元，利用所述第二卡片展示所述目标信息和所述客户端的版本升级提示信息。

本申请实施例还提供一种信息展示业务的处理装置，所述装置位于服务端，包括：接收单元以及返回单元，其中：

15 接收单元，接收客户端发送的获取目标信息的业务请求，所述业务请求中包含用于确定所述客户端的配置版本的版本标识，所述版本标识与所述客户端的配置版本所支持的卡片类型之间具有对应关系；

20 返回单元，当根据所述版本标识对应的卡片类型确定所述客户端的配置版本不支持用于展示所述目标信息的第一卡片的卡片类型时，向所述客户端返回目标信息以及指示信息，所述指示信息用于指示所述客户端利用所述客户端的配置版本所支持的第二卡片展示所述目标信息和所述客户端的版本升级提示信息。

本申请实施例采用的上述至少一个技术方案能够达到以下有益效果：

25 当确定客户端的配置版本不支持所述第一卡片的卡片类型时，通过第一卡片中的关联标识关联该配置版本所支持的第二卡片，利用该第二卡片展示目标信息和客户端的版本升级提示信息，从而解决了现有技术中不升级客户端的配置版本，无法展示新的卡片类型的问题。另外，还可以将第二卡片的卡片类型统一配置为多个配置版本均支持的卡片类型，这样当确定客户端不支持某个类型的卡片后，通过该统一类型的卡片进行展示，使得所展示的卡片更加易于管理。

30 附图说明

此处所说明的附图用来提供对本申请的进一步理解，构成本申请的一部分，本申请的示意性实施例及其说明用于解释本申请，并不构成对本申请的不当限定。在附图中：

图 1 为本申请实施例 1 提供的一种基于卡片的信息展示方法的具体实现流程示意图；

图 2 为本申请实施例 1 提供的实际应用中的一种奇点卡片的示意图；

5 图 3 为本申请实施例 2 提供的一种信息展示业务的处理方法的具体实现流程示意图；

图 4 为本申请实施例 2 提供的实际应用中通过卡片基类生成卡片类型的示意图；

图 5 为本申请实施例 2 提供的实际应用中的一种信息展示系统；

图 6 为本申请实施例 3 提供的一种基于卡片的信息展示装置的具体结构示意图；

图 7 为本申请实施例 4 提供的一种信息展示业务的处理装置的具体结构示意图。

10

具体实施方式

如上所述，随着互联网的蓬勃发展，越来越多的应用 APP 采用卡片流（Feed 流）的形式向用户进行信息展示。在通过 Feed 流进行信息展示的过程中，根据所要展示的信息类型的不同，通常将所要展示的信息配置为不同的类型的卡片。

15 例如，微信朋友圈通过 Feed 流，将好友的状态信息（好友评论、状态更新等）配置为卡片进行展示；微博通过 Feed 流，将新闻配置为卡片进行展示。由于不同展示信息通常需要的展示形式不同，因此卡片对应有多种不同的类型。例如，股票信息在展示时，通常需要能够显示股票的变化趋势；观点类信息在展示时，通常需要能够评论等，因此，在实际应用中，通常需要配置出多种不同类型的卡片。

20 随着应用 APP 版本的不断更新，在新版本中通常会出现新的卡片类型，这些新的卡片类型通常在旧版本中无法进行展示。在现有技术中，当应用 APP 遇到无法展示的卡片类型时，通常会通过引导用户进行版本更新的形式进行处理。在用户通过引导进行应用 APP 的版本更新后，能够显示该类型的卡片。

然而，通过更新应用 APP 的版本实现新类型的卡片在应用 APP 中的展示这一方法
25 存在：在应用 APP 服务器出现繁忙等情况导致应用 APP 更新失败时，依然无法对该新类型的卡片进行展示，使得用户对该应用 APP 的使用体验降低。

为使本申请的目的、技术方案和优点更加清楚，下面将结合本申请具体实施例及相应的附图对本申请技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在
30 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。

以下结合附图，详细说明本申请各实施例提供的技术方案。

实施例 1

实施例 1 提供了一种基于卡片的信息展示方法，该方法用于客户端，能够解决现有技术中的问题。该方法的具体流程示意图如图 1 所示，包括下述步骤：

步骤 S11：接收展示指令。

该客户端通常可以是应用 APP 的客户端，也可以是网页应用（Web APP）的客户端。

在特定条件下，客户端会接收到展示指令。例如，当指定时间点（上午 9 点或者其它时间点）到来时，客户端可以接收到展示指令；或者，当服务器需要客户端展示某些信息（广告、资讯信息等）时，会向客户端发送展示指令；或者，根据用户的个人设置，在一定条件下（例如，好友有状态更新），客户端会接收到展示指令；或者，用户通过触发，使得客户端接收到展示指令，并且通常用户触发的方式可以为鼠标点击的方式、点击触控触摸屏按钮的方式或拉取触控触摸屏的方式等。

以用户刷新微信朋友圈为例，当用户触控触摸屏上的朋友圈按钮时，客户端会接收到展示指令，根据该展示指令能够展示微信好友的状态更新和/或服务端推送的广告等；在微博中，当用户通过拉取等方式刷新微博时，微博的客户端会接收到展示指令，用以展示所关注好友的状态更新、新闻资讯等。

步骤 S12：确定待展示的目标信息和用于展示所述目标信息的第一卡片的卡片类型。

目标信息通常可以是新闻资讯、广告资讯、天气预报消息、交通路况消息、基金或股票消息、评论消息、朋友圈的状态消息等。并且，目标信息的展现元素可以为文字、图片、音频和/或视频等。

例如，当用户刷新微信朋友圈时，好友更新后的新的状态消息为待展示的目标信息，该目标信息的展示元素可以包括文字、图片、音频和/或视频等。

通常来说，用于展示目标信息的卡片都有对应的卡片类型，卡片类型反映了卡片的分类类别。例如，观点类的卡片对应观点类的卡片类型（简称，观点卡片）、股票类的卡片对应股票类的卡片类型（简称，股票卡片），基金类的卡片对应基金类的卡片类型（简称，基金卡片）等，这些不同类型的卡片对应着不同的分类类别。另外，对于客户端来说，不同配置版本的客户端通常能够展示的卡片类型通常不同，客户端的配置版本与能够展示的卡片类型通常具有对应关系。

通常来说，客户端确定目标信息以及第一卡片的卡片类型，可以是获取目标信息以

及第一卡片的卡片类型；并且，通常还可以是从客户端的缓存数据中获取该目标信息以及第一卡片的卡片类型，这种从缓存数据中获取的方式，能够加快卡片展示的响应速度；另外，还可以是从服务端获取该目标信息以及第一卡片的卡片类型。

步骤 S13：判断客户端的配置版本是否支持所述第一卡片的卡片类型，若是，则执行步骤 S14，若否，则执行步骤 S15。

通常来说，在实际应用中，判断客户端的配置版本是否支持第一卡片的卡片类型的方式可以有多种，这里列举几种优选方案：

优选方案一：可以在客户端中配置白名单，在该白名单中记录该客户端的配置版本所支持的卡片类型；则，在判断客户端的配置版本是否支持第一卡片的卡片类型时，可以将第一卡片的卡片类型与该白名单中包含的卡片类型进行匹配，根据匹配结果，判断该客户端的配置版本是否支持所述第一卡片的卡片类型，当匹配成功时，确定该客户端的配置版本支持第一卡片的卡片类型，当匹配失败时，确定该客户端的配置版本不支持第一卡片的卡片类型。

这种通过白名单进行判断的方式，当新增加了该配置版本的客户端支持的卡片类型时，可以对白名单中所记录的卡片类型进行更新，在实际应用中易于实现。

优选方案二：客户端通常会包括配置版本的版本标识，该版本标识能够唯一标识该配置版本的客户端。客户端的配置版本与所支持的卡片类型之间的对应关系，可以体现为版本标识与所支持的卡片类型之间的对应关系。因此，当客户端中包含配置版本的版本标识与该配置版本所支持的卡片类型之间的对应关系时，判断客户端的配置版本是否支持第一卡片的卡片类型，可以先确定该客户端的配置版本的版本标识，并根据该对应关系确定该版本标识对应的配置版本所支持的卡片类型，然后根据第一卡片的卡片类型和所确定的所述客户端的配置版本所支持的卡片类型，确定所述客户端的配置版本是否支持所述第一卡片的卡片类型。

步骤 S14：利用所述第一卡片展示所述目标信息。

当确定客户端的配置版本支持所述第一卡片的卡片类型时，可以通过第一卡片展示该目标信息。

步骤 S15：确定所述第一卡片对应的关联标识，并根据所述关联标识关联所述客户端的配置版本所支持的第二卡片，利用所述第二卡片展示所述目标信息和所述客户端的版本升级提示信息。

当确定客户端的配置版本不支持所述第一卡片的卡片类型时，可以通过第一卡片对

应的关联标识，关联客户端的配置版本所支持的第二卡片。在这里，所说的第二卡片的卡片类型为该配置版本的客户端支持的卡片类型。

例如，客户端的配置版本为“XYZ”，客户支持的卡片类型为 A、B 和 C，第一卡片的卡片类型为 D，当确定该配置版本的客户端不支持第一卡片的卡片类型（D）时，
5 可以通过第一卡片的关联标识，关联卡片类型 A、B 或 C，然后利用 A、B 或 C 类型的卡片，展示目标信息以及客户端的版本升级提示信息，这里的 A、B 或 C 类型的卡片称之为第二卡片。

需要说明的是，在实际应用中，由于客户端的配置版本经常需要更新，为了使得客户端在确定不支持某个卡片类型时，能够通过同一个卡片类型的卡片进行展示，通常可以
10 将第二卡片的卡片类型设置为多个配置版本均支持的卡片类型。这样可以使得，当客户端确定不支持某个卡片类型的卡片时，通过同一种指定卡片类型的卡片进行展示，这种方式可以方便的对所展示的卡片进行管理。

另外，在步骤 S12 中提到，当从服务端获取目标信息和第一卡片的卡片类型时，还可以同时从服务端获取第二卡片，并且在获取第二卡片后可以将该第二卡片在客户端进行缓存。这样可以使得，在根据所述关联标识关联所述客户端的配置版本所支持的第二
15 卡片时，可以根据所述关联标识关联已缓存在客户端的第二卡片，从而减少获取第二卡片的时间，提高响应速度。

采用实施例 1 的该方法，当确定客户端的配置版本不支持述第一卡片的卡片类型时，通过第一卡片中的关联标识关联该配置版本所支持的第二卡片，利用该第二卡片展示目标信息和客户端的版本升级提示信息，从而解决了现有技术中不升级客户端的配置版本，
20 无法展示新的卡片类型的问题。另外，还可以将第二卡片的卡片类型统一配置为多个配置版本均支持的卡片类型，这样当确定客户端不支持某个类型的卡片后，通过该统一类型的卡片进行展示，使得所展示的卡片更加易于管理。

另外，需要说明的是，在通过第二卡片展示目标信息时，还需要同时展示客户端的版本升级提示信息，通过该版本升级提示信息提示用户进行版本升级。在实际应用中，
25 由于用户通常并不清楚客户端出现了新的配置版本，在通过第二卡片展示目标信息时，如果没有同时展示客户端的版本升级提示信息，用户通常并不会对客户端的配置版本进行升级，从而使得用户不能及时体验新配置版本的客户端所带来的体验提升，并且对于提供版本升级服务的公司也会造成业务量损失。

30 特别的，第一卡片能够展示目标信息（这里称为第一目标信息），第二卡片能够展

示与第一目标信息对应的第二目标信息，这里的第二目标信息可以与第一目标信息相同，也可以为调整第一目标信息中的对应展示元素后得到的目标信息。可以根据实际需要由第一目标信息确定第二目标信息。

例如，第一目标信息包括文字 A 以及图片 B 和图片 C，第二目标信息可以为文字 A 以及图片 B 和图片 C，也可以只为文字 A，也可以为文字 A 以及图片 A（或图片 B），也可以为文字 A 的一部分等。

在实际应用中，客户端从服务端获取的目标信息、第一卡片的卡片类型或关联标识等，通常是以固定的数据格式进行获取。该固定的数据格式，通常由客户端和服务端双方事先定义的协议格式所规定。例如，该协议格式可以如下所示。

10 将第一卡片的数据抽象成一个 FeedViewItem 类，并在其中添加一个关联字段（可以形象地称为“奇点字段”或“singular”），FeedViewItem 类图可以如表 1 所示。

表 1：实际应用中的 FeedViewItem 类

属性	数据类型	说明
feedId	String	第一卡片的卡片 id
type	String	第一卡片的卡片类型
content	json	第一卡片的目标信息
create time	Long	第一卡片的时间戳
singular	json	关联标识

在 FeedViewItem 类中，通过“type”字段的数据（即第一卡片的卡片类型），当确定客户端的配置版本不支持第一卡片的卡片类型时，通过“singular”字段的数据（即关联标识）关联第二卡片，以便通过第二卡片（可以形象的称为“奇点卡片”，奇点 card）展示目标信息以及客户端的版本升级提示信息。

在实际应用中，也可以将第二卡片（奇点卡片）抽象成一个数据类，该数据类可以如表 2 所示。

表 2：实际应用中奇点卡片的数据类

属性	数据类型	说明
----	------	----

content	String	奇点卡片中的目标信息
referenceMap	json	目标信息中包含的链接
tip	String	奇点卡片中的版本升级提示信息
tipReferenceMap	json	版本升级提示信息中包含的链接
clickUrl	String	点击奇点卡片跳转的链接

其中，“clickUrl”字段定义了，点击整个奇点卡片时，实现页面跳转的链接；“content”字段定义了，奇点卡片中展示的目标信息；“referenceMap”字段定义了，目标信息中包含的链接，例如下载新的配置版本的链接和/或关联其它用户（例如，@其它用户）的链接等；“tip”字段定义了，奇点卡片中的版本升级提示信息，可以通过该版本升级提示信息提醒用户进行版本升级；“tipReferenceMap”字段定义了，版本升级提示信息中包含的链接，通过该链接通常可以下载新的配置版本。

另外，在程序编写阶段，还可以为各个字段均增加占位符，以便于编写程序，下面是对各个占位符说明：

“U”代表用户名称；“G”代表图片；“S”代表股票；“F”代表基金；“T”代表话题；“V”代表视频；“L”代表链接。

需要说明的是，在将奇点卡片进行展示时，通常还需要根据预设规则约定该奇点卡片中各展示元素（文字、图片、音频、视频等）的大小、颜色、样式、链接地址等。在实际应用中，可以通过构建 ReferenceData 类来进行约定。该 ReferenceData 类可以如表 3 所示。

表 3：实际应用中的 ReferenceData 类

属性	类型	说明
referString	String	定义了链接的显示样式（高亮显示、斜体加粗等，下面以高亮显示为例）
matchStart	int	定义了高亮显示起始字符的位置
matchEnd	int	定义了高亮显示终止字符的位置
placeholderKey	String	用于代替占位符中的内容

如图 2 所示为实际应用中的奇点卡片，在该奇点卡片中，A 为奇点卡片中的目标信息（包括文本内容和视频），由表 2 中“content”字段确定，并且在 A 中，有斜体加粗（实际应用中也可以为高亮显示等）部分为目标信息中包含的链接，该链接的地址由表 2 中“referenceMap”字段确定，另外，斜体加粗的显示样式，斜体加粗显示的起始字符、终止字符分别由表 3 中“referString”、“matchStart”以及“matchEnd”字段确定；B 为奇点卡片中的客户端版本升级提示信息，由表 2 中“tip”字段确定，并且在 B 中，有斜体加粗（实际应用中可以为高亮显示等）部分为 B 中包含的链接，该链接的地址由表 2 中“tipReferenceMap”字段确定，该链接斜体加粗的显示样式，斜体加粗显示的起始字符、终止字符也分别由表 3 中“referString”、“matchStart”以及“matchEnd”字段确定；C 为整个奇点卡片界面，点击奇点卡片界面进行页面跳转，该页面跳转链接的地址由表 2 中“clickUrl”字段确定。

实施例 2

实施例 1 所提供的方法用于客户端，在实际应用中，当需要展示目标信息时，通常需要即时地从服务端获取数据。因此，本申请的实施例 2 将提供一种信息展示业务的处理方法，该方法用于服务端，能够解决现有技术中的问题。该方法的具体流程示意图如图 3 所示，包括下述步骤：

步骤 S21：接收客户端发送的获取目标信息的业务请求，所述业务请求中包含用于确定所述客户端的配置版本的版本标识，所述版本标识与所述客户端的配置版本所支持的卡片类型之间具有对应关系。

这里的目标信息与步骤 S12 中的目标信息相同，这里就不再赘述。

该业务请求中包含发送该业务请求的客户端的配置版本的版本标识，该版本标识与该配置版本的客户端所支持的卡片类型具有对应关系。

步骤 S22：当根据所述版本标识对应的卡片类型确定所述客户端的配置版本不支持用于展示所述目标信息的第一卡片的卡片类型时，向所述客户端返回目标信息以及指示信息，所述指示信息用于指示所述客户端利用所述客户端的配置版本所支持的第二卡片展示所述目标信息和所述客户端的版本升级提示信息。

在实际应用中，通常可以通过服务端来判断客户端是否支持第一卡片的卡片类型，当不支持时，向该客户端返回目标信息以及指示信息，通过该指示信息指示客户端利用第二卡片展示目标信息。通常，该第一卡片的卡片类型可以是新的卡片类型，第二卡片

的卡片类型是该客户端的配置版本支持的卡片类型。也就是说，服务端判断客户端是否支持新的卡片类型，当不支持时，通过指示信息指示客户端利用能够支持的卡片类型展示目标信息。

另外，服务端向客户端返回目标信息以及指示信息时，可以通过 json(JavaScript Object Notation)字符串的形式向所述客户端返回目标信息以及指示信息。

采用本申请实施例 2 所提供的该方法，通过服务端判断客户端的配置版本不支持第一卡片的卡片类型时，通过指示信息指示客户端通过第二卡片展示目标信息以及客户端的版本升级提示信息，从而解决了现有技术中的问题。另外，与实时例 1 相比，这种通过服务端判断客户端的配置版本是否支持第一卡片的卡片类型的方式，由于服务端的运算能力通常比客户端强，因此能够提高目标信息的展示速度。

需要说明的是，在实际应用中，当服务端的运算压力较大时，通常为了减轻服务端的运算压力，也可以通过实施例 1 的方法，通过客户端进行判断后展示目标信息，这样减轻服务端的运算压力。

实际应用中，服务端在生成新的卡片类型时，通常先生成一个卡片基类 (FeedBaseModel)，再通过该卡片基类来生成新的卡片类型。该 FeedBaseModel 可以包括 feedID、type、content、create time 四个基本字段，当需要生成新的卡片类型时，在四个基本字段的基础上增加对应的新的字段，如图 4 所示。

例如，当需要生成奇点卡片时，在 feedID、type、content、create time 四个基本字段的基础上增加 singular 字段；当需要生成观点卡片 (CommentFeedModel) 时，在 feedID、type、content、create time 四个基本字段的基础上增加 “Comment 基字段；当需要生成股票卡片时，在 feedID、type、content、create time 四个基本字段的基础上增加 “Stock 段字段；此外，还可以通过对应的方式生成基金卡片 (FundFeedModel)、SpecialFeedModel、FootmarkFeedModel 等卡片类型。

如图 5 所示为包括客户端 401 和服务端 402 的一种信息展示系统，该系统中客户端 401 还可以分为三个层级，即数据解析层 4011、卡片数据分发层 4012 以及卡片渲染层 4013，其中：

服务端 402：根据业务请求，生成用于展示目标信息的卡片的卡片数据，并以 Json 字符串的形式将卡片数据下发给客户端 401，并且在该服务端 402 通常还可以维护一张可更新的白名单，使得可以通过服务端 402 来确定客户端 401 能够支持的卡片类型，并且通常还可以将客户端 401 中的白名单与服务端 402 中的白名单进行同步；

数据解析层 4011: 从服务端 402 下发的 json 字符串形式的卡片数据中解析出对应的卡片类型;

卡片数据分发层 4012: 依照客户端 401 中的白名单, 把能够识别的卡片类型解析成对应的类型的卡片, 把不在该白名单内的卡片, 统一解析成奇点卡片。并且, 在实际应用中, 当客户端 401 刚启动时 (或者在指定时间点), 还可以将客户端 401 中的白名单与服务端 402 中维护的白名单进行同步。

卡片渲染层 4013: 用于将卡片数据分发层 4012 的结果, 渲染出对应类型的卡片, 进行展示。

10 实施例 3

基于与实施例 1 相同的发明构思, 本申请实施例 3 提供了一种基于卡片的信息展示装置, 所述装置位于客户端。如图 6 所示, 该装置 50 包括: 确定单元 501、关联单元 502 以及展示单元 503, 其中:

确定单元 501, 确定待展示的目标信息和用于展示所述目标信息的第一卡片的卡片类型;

关联单元 502, 当确定客户端的配置版本不支持所述第一卡片的卡片类型时, 确定所述第一卡片对应的关联标识, 并根据所述关联标识关联所述客户端的配置版本所支持的第二卡片;

展示单元 503, 利用所述第二卡片展示所述目标信息和所述客户端的版本升级提示信息。

采用本申请实施例 3 所提供的该装置 50, 由于该装置 50 基于与实时例 1 相同的发明构思, 因此能够解决现有技术中的问题。并且, 在实际应用中, 该装置 50 还可以通过结合具体的硬件设备取得其它的实施效果。例如, 将该装置 50 的客户端安装在移动终端上, 由于客户端能够通过第二卡片展示现有技术中通过第一卡片无法展示的目标信息, 因此能够提高该移动终端的用户体验, 从而提升该移动终端的商业价值。

实施例 4

基于与实施例 2 相同的发明构思, 本申请实施例 4 提供了一种信息展示业务的处理装置, 所述装置位于服务端。如图 7 所示, 该装置 60 包括: 接收单元 601 以及返回单元 602, 其中:

接收单元 601，接收客户端发送的获取目标信息的业务请求，所述业务请求中包含用于确定所述客户端的配置版本的版本标识，所述版本标识与所述客户端的配置版本所支持的卡片类型之间具有对应关系；

5 返回单元 602，当根据所述版本标识对应的卡片类型确定所述客户端的配置版本不支持用于展示所述目标信息的第一卡片的卡片类型时，向所述客户端返回目标信息以及指示信息，所述指示信息用于指示所述客户端利用所述客户端的配置版本所支持的第二卡片展示所述目标信息和所述客户端的版本升级提示信息。

采用本申请实施例 4 所提供的该装置 60，由于该装置 60 采用与实施例 2 相同的发明构思，因此能够解决现有技术中的问题。另外，该装置 60 位于服务端，通过服务端判定客户端是否能识别第一卡片，由于服务端的运算能力强于客户端，因此能够提高目标信息的展示速度。

在实际应用中，通过返回单元 602 向客户端返回目标信息以及指示信息时，可以以 json 字符串的形式向所述客户端返回目标信息以及指示信息。

15 本领域内的技术人员应明白，本申请的实施例可提供为方法、系统、或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

20 本申请是参照根据本申请实施例的方法、设备（系统）、和计算机程序产品的流程图和 / 或方框图来描述的。应理解可由计算机程序指令实现流程图和 / 或方框图中的每一流程和 / 或方框、以及流程图和 / 或方框图中的流程和 / 或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的装置。

25 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的步骤。

- 5 在一个典型的配置中，计算设备包括一个或多个处理器 (CPU)、输入/输出接口、网络接口和内存。

内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器 (RAM) 和/或非易失性内存等形式，如只读存储器 (ROM) 或闪存(flash RAM)。内存是计算机可读介质的示例。

- 10 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、
15 只读光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体 (transitory media)，如调制的数据信号和载波。

- 20 还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括要素的过程、方法、商品或者设备中还存在另外的相同要素。

- 25 本领域技术人员应明白，本申请的实施例可提供为方法、系统或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

- 30 以上仅为本申请的实施例而已，并不用于限制本申请。对于本领域技术人员来说，本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替

换、改进等，均应包含在本申请的权利要求范围之内。