

# **Card-Based Information Displaying Method and Apparatus, and Information Displaying Service Processing Method and Apparatus**

The present application claims priority to the Chinese patent application No.  
5 201610630286.0, filed on August 03, 2016 and entitled “Card-Based Information  
Displaying Method and Apparatus, and Information Displaying Service Processing  
Method and Apparatus”, the entirety of which is hereby incorporated by reference into  
this application.

## **10 Technical Field**

The present application relates to computer technology, particularly to a  
card-based information displaying method and apparatus, and an information  
displaying service processing method and apparatus.

## **15 Background Technologies**

Along with the rapid development of the Internet, more and more applications  
(APPS) adopt the form of Feed stream to display information to users. In the process  
of displaying information through Feed stream, the information to be displayed is  
typically configured to be different types of cards according to the types of the  
20 information to be displayed.

However, with the continuous update of the version of an APP, new card types  
typically will appear in a new version. When the version of the APP used by a user  
has not been updated, these new types of cards cannot be displayed in the APP used  
by the user.

25 With respect to the situation where an APP cannot display a new type of card,  
the user is typically guided to proceed to an APP version update operation. After the  
user completes version update of the APP under the guidance, the new type of card  
will be displayed in the APP used by the user.

30 However, the method for achieving display of a new type of card in an APP by  
updating the APP version has the following problem: when the APP version update

fails because the APP server is busy or due to other circumstances, the new type of card still cannot be displayed, causing a lower user experience in the APP.

### **Summary**

5           The embodiments of the present application provide a card-based information displaying method and apparatus as well as an information displaying service processing method and apparatus that solve the problem in the current technologies.

          The embodiments of the present application provide a card-based information displaying method, comprising:

10           determining target information to be displayed and a card type of a first card for displaying the target information,

          when it is determined that a configuration version of a client does not support the card type of the first card, determining an association identifier corresponding to the first card, and associating with a second card supported by the configuration  
15           version of the client according to the association identifier; and

          displaying the target information and version update prompt information of the client using the second card.

          Preferably, if the client contains a white list, and the white list contains card types that are supported by the configuration version of the client, then

20           determining whether the configuration version of the client supports the card type of the first card, which comprises:

          matching the card type of the first card with the card types contained in the white list; and

          according to a match result, determining whether the configuration version of  
25           the client supports the card type of the first card.

          Preferably, if the client contains a corresponding relation between a version identifier of the configuration version of the client and the card type supported by the configuration version,

          determining whether the configuration version of the client supports the card  
30           type of the first card, comprises:

          determining the version identifier of the configuration version of the client, and

based on the corresponding relation, determining the card type supported by the configuration version corresponding to the version identifier; and

according to the card type of the first card and the determined card type supported by the configuration version of the client, determining whether the configuration version of the client supports the card type of the first card.

Preferably, associating with a second card supported by the configuration version of the client according to the association identifier, comprises:

according to the association identifier, determining card data of a second card associated with the first card; and

according to the card data, determining a second card supported by the configuration version of the client.

Preferably, determining target information to be displayed and a card type of a first card for displaying the target information comprises: obtaining target information to be displayed and a card type of a first card for displaying the target information from cached data of a client; or

obtaining target information to be displayed and a card type of a first card for displaying the target information from a server.

Preferably, when target information to be displayed and a card type of a first card for displaying the target information are obtained from a server, the method further comprises: obtaining from a server a second card supported by the configuration version of the client and caching the second card; then,

associating with a second card supported by the configuration version of the client according to the association identifier, comprises: associating with the cached second card according to the association identifier.

The embodiments of the present application further provide an information displaying service processing method, comprising:

receiving a service request sent by a client for obtaining target information, wherein the service request contains a version identifier for determining a configuration version of the client, and the version identifier has a corresponding relation with a card type supported by the configuration version of the client; and

when it is determined according to the card type corresponding to the version

identifier that the configuration version of the client does not support a card type of a first card for displaying the target information, returning to the client the target information and instruction information that is used to instruct the client to display the target information and version update prompt information of the client using a second  
5 card supported by the configuration version of the client.

Preferably, returning to the client the target information and instruction information comprises: returning the target information and instruction information to the client in a form of a json character string.

The embodiments of the present application further provide a card-based  
10 information displaying apparatus, which is located on a client and comprises: a determining unit, an associating unit and a displaying unit, wherein:

the determining unit, determining target information to be displayed and a card type of a first card for displaying the target information;

the associating unit, when it is determined that a configuration version of a  
15 client does not support the card type of the first card, determining an association identifier corresponding to the first card and associating with a second card supported by the configuration version of the client according to the association identifier; and

the displaying unit, displaying the target information and version update prompt information of the client using the second card.

20 The embodiments of the present application further provide an information displaying service processing apparatus, which is located on a server and comprises: a receiving unit and a returning unit, wherein:

the receiving unit, receiving a service request sent by a client for obtaining target information, wherein the service request contains a version identifier for  
25 determining a configuration version of the client, and the version identifier has a corresponding relation with the card type supported by the configuration version of the client; and

the returning unit, when it is determined according to the card type corresponding to the version identifier that the configuration version of the client does  
30 not support a card type of a first card for displaying the target information, returning to the client the target information and instruction information used to instruct the client to display the target information and version update prompt information of the

client using a second card supported by the configuration version of the client.

At least one of the foregoing technical solutions adopted by the embodiments of the present application can achieve the following beneficial effect:

When it is determined that the configuration version of a client does not support the card type of a first card, a second card supported by the configuration version is associated through the association identifier in the first card, and the second card is used to display the target information and version update prompt information of the client, thus solving the problem in the current technologies that a new card type cannot be displayed if the configuration version of the client has not been updated. Furthermore, the card type of the second card may also be uniformly configured as a card type that is supported by a plurality of configuration versions. In this way, when it is determined that a client does not support a type of card, displaying can be conducted through this unified type of card, facilitating the management of the displayed cards.

#### **Brief Description of the Drawings**

The drawings described herein are intended to provide a further understanding of the present application and constitute a part of this application, and the exemplary embodiments of the present application and their descriptions are intended to explain the present application and not to limit the present application. In the drawings:

FIG. 1 is a schematic diagram of the implementation flow of a card-based information displaying method according to Embodiment 1 of the present application.

FIG. 2 is a schematic diagram of a singular card in an actual application according to the Embodiment 1 of the present application.

FIG. 3 is a schematic diagram of the implementation flow of an information displaying service processing method according to Embodiment 2 of the present application.

FIG. 4 is a schematic diagram of generation of a card type through a card base class in an actual application according to the Embodiment 2 of the present application.

FIG. 5 shows an information displaying system in an actual application according to the Embodiment 2 of the present application.

FIG. 6 is a structural diagram of a card-based information displaying apparatus according to Embodiment 3 of the present application.

FIG. 7 is a structural diagram of an information displaying service processing apparatus according to Embodiment 4 of the present application.

5

### **Detailed Description**

As described above, along with the rapid development of the Internet, more and more APPS adopt the form of Feed stream to display information to users. In the process of displaying information through Feed stream, the information to be  
10 displayed is typically configured to be different types of cards according to the types of the information to be displayed.

For example, a friend circle of WeChat displays status information of friends (such as friend comments and status updates) by configuring it as cards through Feed stream, and microblog displays news by configuring it as cards through Feed stream.  
15 As different information to be displayed typically needs different displaying forms, cards accordingly have a plurality of types. For example, display of stock information typically needs to indicate variation trends of stocks, and display of view information typically needs to allow comments. Therefore, in an actual application, typically different types of cards need to be configured.

20 With the continuous update of the version of an APP, new card types typically will appear in a new version. Typically, these new types of cards cannot be displayed in an old version. In the current technologies, when an APP is unable to display a card type, the user is typically guided to update the version of the APP. After the user completes version update of the APP under the guidance, this type of card can be  
25 displayed.

However, the method for achieving display of a new type of card in an APP by updating the APP version has the following problem: when APP version update fails because the APP server is busy or due to other circumstances, the new type of cards still cannot be displayed, causing a lower user experience of the APP.

30 The technical solutions of the present application will now be clearly and completely described with reference to the embodiments and accompanying drawings of the present application from which its objectives, technical solutions and

advantages will be evident. Obviously, the described embodiments are only some and not all of the embodiments of the present application. Based on the embodiments in the present application, all other embodiments obtained by those of ordinary skill in the art without creative effort shall fall within the scope of the present application.

5           Below by referring to the accompanying drawings, the technical solutions according to the embodiments of the present application are described in detail.

### **Embodiment 1**

Embodiment 1 provides a card-based information displaying method. This  
10   method is used on a client and can solve the problem in current technologies. The schematic diagram of its flow is shown in FIG. 1, including the following steps:

Step S11: receiving a display instruction.

The client typically may be a client of an APP, or a client of a Web APP.

Under some conditions, the client will receive a display instruction. For  
15   example, when a designated time point (9 AM or any other time point) comes, the client may receive a display instruction; or, when a server needs the client to display some information (advertisement, news information, etc.), it will send a display instruction to the client; or, according to a personal setting of a user, under certain conditions (for example, the status of a friend is updated), the client will receive a  
20   display instruction; or, through triggering, a user causes the client to receive a display instruction, and a typical way of triggering by a user may be clicking the mouse, clicking the touch screen button or pulling the touch screen.

Taking a user's refreshing of a WeChat friend circle for example, when a user  
25   touches a friend circle button on the touch screen, the client will receive a display instruction. Based on this display instruction, status updates of WeChat friends and/or advertisements pushed by the server, etc. can be displayed. In a microblog, when a user refreshes the microblog by pulling or other methods, the client of the microblog will receive a display instruction to display status updates of the user's friends, news information, etc..

30           Step S12: determining target information to be displayed and a card type of a first card for displaying the target information.

Target information typically may be news information, advertisement

information, weather forecast messages, traffic messages, fund or stock messages, comment messages, friend circle status messages, etc. Further, elements for presenting the target information may be text, picture, audio and/or video.

For example, when a user refreshes a WeChat friend circle, new status messages  
5 after friends' update are target information to be displayed. The elements for presenting this target information may be text, picture, audio and/or video.

Generally speaking, every card for displaying the target information has a corresponding card type that indicates a category of the card. For example, a comment type of card corresponds to a card type of comment (for short, comment card), a stock  
10 type of card corresponds to a card type of stock (for short, stock card), and a fund type of card corresponds to a card type of fund (for short, fund card). These different types of cards correspond to different categories. Further, as far as clients are concerned, clients of different configuration versions typically can display different card types, and a configuration version of a client typically has a corresponding relation with the  
15 card type that can be displayed.

Generally speaking, determining target information and a card type of a first card by a client may be: obtaining target information and a card type of a first card, and typically may further be: obtaining the target information and the card type of the first card from cached data of the client, where this method can raise the response  
20 speed of card display. In addition, it may further be: obtaining the target information and the card type of the first card from a server.

Step S13: determining whether the configuration version of the client supports the card type of the first card, and if yes, then proceeding to Step S14; if no, then proceeding to Step S15.

25 Generally speaking, in an actual application, whether the configuration version of client supports the card type of a first card may be determined by a plurality of methods. Here, a few preferred solutions are enumerated:

Preferred solution 1: A white list may be configured in the client. In the white list, the card types supported by the configuration version of the client are recorded.  
30 Then, when it is determined whether the configuration version of the client supports the card type of the first card, the card type of the first card may be matched with the card types contained in the white list and, according to a result of the match, it is

determined whether the configuration version of the client supports the card type of the first card. When the match is successful, it is determined that the configuration version of the client supports the card type of the first card. When the match fails, it is determined that the configuration version of the client does not support the card type of the first card.

This determination method through a white list can be easily achieved in actual applications. When a new card type that is supported by the client of the configuration version is added, the card types recorded in the white list can be updated.

Preferred solution 2: A client typically can include a version identifier of a configuration version. The version identifier can uniquely identify the client of the configuration version. A corresponding relation between the configuration version of the client and its supported card type can be manifested as a corresponding relation between the version identifier and the supported card type. Therefore, when a client contains a corresponding relation between a version identifier of a configuration version and a card type that is supported by the configuration version, firstly the version identifier of the configuration version of the client may be determined, then the card type that is supported by the configuration version corresponding to the version identifier is determined according to this corresponding relation, and then it is determined whether the configuration version of the client supports the card type of the first card according to the card type of the first card and the determined card type that is supported by the configuration version of the client.

Step S14: displaying the target information using the first card.

When it is determined that the configuration version of a client supports the card type of the first card, the target information may be displayed by using the first card.

Step S15: determining an association identifier corresponding to the first card, associating with a second card supported by the configuration version of the client according to the association identifier, and displaying the target information and version update prompt information of the client using the second card.

When it is determined that the configuration version of a client does not support the card type of the first card, a second card supported by the configuration version of the client may be associated through the association identifier corresponding to the first card. Here, a card type of a second card is a card type that is supported by the

client of the configuration version.

For example, the configuration version of a client is "XYZ", the card types supported by the client are A, B and C, and the card type of a first card is D. When it is determined that the client of the configuration version does not support the card type (D) of the first card, the card type A, B or C may be associated with through the association identifier of the first card, and then a card of the type A, B or C is used to display target information and version update prompt information of the client. Here, the card of the type A, B or C is called a second card.

It needs to be noted that, in an actual application, as the configuration version of a client often needs to be updated, in order that the client can display target information through the same card type of card when it is determined that the client does not support a card type, typically the card type of a second card may be set as a card type that is supported by a plurality of configuration versions. In this way, when it is determined that a client does not support a type of card, displaying can be conducted through the same designated card type of card, facilitating the management of the displayed cards.

Further, it is mentioned in Step S12 that, when target information and the card type of a first card are obtained from a server, a second card may also be obtained from the server at the same time, and after the second card is obtained, the second card may be cached on a client. In this way, when a second card supported by the configuration version of the client is associated with according to the association identifier, the second card that has been cached on the client may be associated with according to the association identifier, thus reducing the time spent in obtaining the second card and raising response speed.

In the method of Embodiment 1, when it is determined that the configuration version of a client does not support the card type of the first card, a second card supported by the configuration version is associated with through the association identifier in the first card, and the second card is used to display the target information and version update prompt information of the client, thus solving the problem in current technologies that a new type of card cannot be displayed if the configuration version of the client is not updated. Further, the card type of the second card may also be uniformly configured as a card type that is supported by a plurality of configuration versions. In this way, when it is determined that a client does not

support a type of card, displaying can be conducted through this unified type of card, facilitating the management of the displayed cards.

Further, it needs to be noted that when target information is displayed through a second card, version update prompt information of a client also needs to be displayed at the same time. Through this version update prompt information, the user is prompted to update the version. In an actual application, as the user typically does not know the client has a new configuration version, and if version update prompt information of a client is not displayed at the same time as the target information is displayed through a second card, the user typically will not update the configuration version of a client, thus being unable to experience improvement brought by the new configuration version of the client in a timely manner and causing loss of service volume of the company providing the version update service.

In particular, a first card can display target information (referred to here as first target information), and a second card can display second target information corresponding to the first target information. Here, the second target information may be same as the first target information and may also be target information obtained after adjustment of corresponding elements for presenting the first target information. The second target information may, according to the actual need, be determined according to the first target information.

For example, first target information comprises text A, picture B and picture C, and second target information may be text A, picture B and picture C, or may be only text A, or may be text A and picture A (or picture B), or may be a part of text A, and so on.

In an actual application, target information of a client obtained from a server, a card type of a first card or an association identifier, typically is obtained in a fixed data format. This fixed data format typically is specified by a protocol format predefined by the client and the server. For example, the protocol format may be as shown below.

Data of a first card is abstracted into a FeedViewItem class and an association field (it may be vividly referred to as a “singular field” or “singular”) is added to the FeedViewItem class. A FeedViewItem class picture may be as shown in Table 1.

**Table 1: FeedViewItem class in an actual application**

| Attribute   | Data type | Description                          |
|-------------|-----------|--------------------------------------|
| feedId      | String    | Card id of the first card            |
| type        | String    | Card type of the first card          |
| content     | json      | Target information of the first card |
| create time | Long      | Timestamp of the first card          |
| singular    | json      | Association identifier               |

In FeedViewItem class, through the data in the “type” field (i.e., a card type of a first card), when it is determined that the configuration version of a client does not support the card type of the first card, a second card is associated through the data in the “singular” field (i.e., association identifier) so as to display target information and version update prompt information of a client through a second card (it may be vividly referred to as a “singular card” or “singular”).

In an actual application, a second card (singular card) may also be abstracted into a data class. This data class may be as shown in Table 2.

**Table 2: Data class of a singular card in an actual application**

| Attribute       | Data type | Description                                            |
|-----------------|-----------|--------------------------------------------------------|
| content         | String    | Target information in the singular card                |
| referenceMap    | json      | Link contained in target information                   |
| tip             | String    | Version update prompt information in the singular card |
| tipReferenceMap | json      | Link contained in version update prompt information    |
| clickUrl        | String    | Link redirected to when the singular card is clicked   |

Here, the “clickUrl” field defines a link which achieves page redirect when the entire singular card is clicked on; the “content” field defines the target information displayed in the singular card; the “referenceMap” field defines a link contained in the target information, e.g., a link for downloading a new configuration version and/or a link for associating with other users (for example, @other users); the “tip” field defines version update prompt information in the singular card, and through this version update prompt information, a user may be reminded to update the version; and the “tipReferenceMap” field defines a link contained in the version update prompt information, and through this link, a new configuration version typically may be

downloaded.

Further, in the programming stage, placeholders may be added to each of the fields to facilitate the programming. Below the placeholders are described one by one:

“U” represents username, “G” represents picture, “S” represents stock, “F” represents fund, “T” represents topic, “V” represents video, and “L” represents link.

It should be noted that when a singular card is displayed, typically the size, color, pattern, link address and other information of every presenting element (text, picture, audio, video, etc.) in the singular card need to be specified according to preset rules. In an actual application, they may be specified by establishing ReferenceData class. The ReferenceData class may be as shown in Table 3.

**Table 3: ReferenceData class in an actual application**

| Attribute      | Data type | Description                                                                                                      |
|----------------|-----------|------------------------------------------------------------------------------------------------------------------|
| referString    | String    | Define the display pattern of a link (highlight, italic, bold, etc., and highlight is taken as an example below) |
| matchStart     | int       | Define the position of the starting character of highlight                                                       |
| matchEnd       | int       | Define the position of the ending character of highlight                                                         |
| placeholderKey | String    | Content for replacing a placeholder                                                                              |

FIG. 2 shows a singular card in an actual application. In the singular card, A is target information (including text content and video) in the singular card and determined by the “content” field in Table 2. Further, in the A, the italic and bold part (in an actual application, it may be highlighted or in other display patterns) is a link contained in the target information. The address of this link is determined by the “referenceMap” field in Table 2. Further, the italic and bold display pattern, as well as the starting character and ending character with the italic and bold display pattern are determined by the “referString”, “matchStart” and “matchEnd” fields in Table 3 respectively. B is version update prompt information of a client in the singular card and determined by the “tip” field in Table 2. Further, in the B, the italic and bold part (in an actual application, it may be highlighted or in other display patterns) is a link contained in the B. The address of this link is determined by the “tipReferenceMap” field in Table 2. The italic and bold display pattern of this link, and the starting character and ending character with the italic and bold display pattern are also

determined by the “referString”, “matchStart” and “matchEnd” fields in Table 3 respectively. C is the entire singular card interface. Clicking the singular card interface can achieve page redirect. The address of this page redirect link is determined by the “clickUrl” field in Table 2.

5

## **Embodiment 2**

The method provided in Embodiment 1 is used on a client. In an actual application, when target information needs to be displayed, typically data needs to be instantaneously obtained from a server. Therefore, Embodiment 2 of the present  
10 application will provide an information displaying service processing method. This method is used on a server and able to solve the problem in current technologies. The schematic diagram of its flow is shown in FIG. 3, including the following steps:

Step S21: receiving a service request sent by a client for obtaining target information, wherein the service request contains a version identifier for determining  
15 the configuration version of the client, and the version identifier has a corresponding relation with the card type that is supported by the configuration version of the client.

The target information here is the same as the target information in Step S12, so it will not be described again.

The service request contains a version identifier of the configuration version of  
20 the client which sends the service request. The version identifier has a corresponding relation with the card type that is supported by the configuration version of the client.

Step S22: when it is determined according to the card type corresponding to the version identifier that the configuration version of the client does not support a card type of a first card for displaying the target information, returning to the client the  
25 target information and instruction information that instructs the client to display the target information and version update prompt information of the client using the second card supported by the configuration version of the client.

In actual application, whether a client supports the card type of the first card typically may be determined on a server. When the client does not support the card  
30 type, the target information and instruction information are returned to the client, and through the instruction information the client is instructed to use a second card to display target information. Typically, the card type of the first card may be a new card

type, and the card type of a second card is a card type that is supported by the configuration version of the client. That is to say, the server determines whether a client supports a new card type. When the client does not support the card type, the client is instructed through the instruction information to use a supported card type to display target information.

Further, when a server returns the target information and instruction information to a client, it may return the target information and instruction information to the client in a form of json (JavaScript Object Notation) character string.

In the method provided by Embodiment 2 of the present application, when a server determines that the configuration version of a client does not support the card type of a first card, the server will instruct the client through instruction information to use a second card to display target information and version update prompt information of the client, thus solving the problem in current technologies. Further, compared with Embodiment 1, this method, which determines on a server whether the configuration version of a client supports the card type of a first card, can raise the speed for displaying target information because a server typically has stronger computing power than a client.

It should be noted that in an actual application, when the computing load of the server is large, in order to alleviate the computing load of the server, the method provided in Embodiment 1 may be adopted and target information can be displayed after determination by the client.

In an actual application, when a new card type is generated on the server, typically a card base class (FeedBaseModel) is first generated, and then a new card type is generated through the FeedBaseModel. The card base class may comprise four basic fields: feedID, type, content and create time. When a new card type needs to be generated, a corresponding new field is added on the basis of the four basic fields, as shown in FIG. 4.

For example, when a singular card needs to be generated, a singular field is added on the basis of the four basic fields: feedID, type, content and create time. When a comment card (CommentFeedModel) needs to be generated, a "Comment" field is added on the basis of the four basic fields: feedID, type, content and create time. When a stock card needs to be generated, a "Stock" field is added on the basis of the four basic fields: feedID, type, content and create time. Further, fund card

(FundFeedModel), SpecialFeedModel, FootmarkFeedModel and other card types may also be generated correspondingly.

FIG. 5 shows an information display system comprising a client 401 and a server 402. The client 401 in the system may further comprise three tiers, namely: a data analysis tier 4011, a card data distribution tier 4012 and a card rendering tier 4013, here:

Server 402: It generates card data of a card for displaying target information according to a service request and sends the card data to the client 401 in a form of Json character string. The server 402 typically may maintain a white list that can be updated so that through the server 402, the card type that can be supported by the client 401 may be determined. Further, typically the white list in the client 401 may also be synchronized with the white list in the server 402.

Data analysis tier 4011: It analyzes the corresponding card type from the card data issued by the server 402 in the form of json character string.

Card data distribution tier 4012: According to the white list in the client 401, it parses the card type that can be recognized into a corresponding type of card and parses the cards not in the white list into singular cards uniformly. Further, in an actual application, when the client 401 is just started (or at the designated time point), the white list in the client 401 may also be synchronized with the white list maintained in the server 402.

Card rendering tier 4013: It is used to render a corresponding type of card from the result of the card data distribution tier 4012 and to display it.

### **Embodiment 3**

Based on the same inventive concept as that of Embodiment 1, Embodiment 3 of the present application provides a card-based information displaying apparatus. The apparatus is located on a client. As shown in FIG. 6, the apparatus 50 comprises: a determining unit 501, an associating unit 502 and a displaying unit 503, here:

determining unit 501, determining target information to be displayed and a card type of a first card for displaying the target information,

associating unit 502, when it is determined that a configuration version of a client does not support the card type of the first card, determining an association

identifier corresponding to the first card, and associating with a second card supported by the configuration version of the client according to the association identifier,

displaying unit 503, displaying the target information and version update prompt information of the client using the second card.

5        When the apparatus 50 provided by Embodiment 3 of the present application is adopted, as the apparatus 50 is based on the same inventive concept as that of Embodiment 1, it can solve the problem in current technologies. Further, in an actual application, the apparatus 50 may also achieve other implementation effects in combination with other hardware devices. For example, the client of the apparatus 50  
10        is mounted on a mobile terminal. As the client can display through a second card the target information that cannot be displayed through a first card in current technologies, the user experience of the mobile terminal can be raised, thus increasing the business value of the mobile terminal.

#### 15        **Embodiment 4**

Based on the inventive concept as that of Embodiment 2, Embodiment 4 of the present application provides an information displaying service processing apparatus. The apparatus is located on a server. As shown in FIG. 7, the apparatus 60 comprises: a receiving unit 601 and a returning unit 602, here:

20        receiving unit 601, receiving a service request sent by a client for obtaining target information, wherein the service request contains a version identifier for determining a configuration version of the client, and the version identifier has a corresponding relation with the card type supported by the configuration version of the client,

25        returning unit 602, when it is determined according to the card type corresponding to the version identifier that the configuration version of the client does not support the card type of a first card for displaying the target information, returning to the client the target information and instruction information that is used to instruct the client to display the target information and version update prompt information of  
30        the client using the second card that is supported by the configuration version of the client.

When the apparatus 60 provided by Embodiment 4 of the present application is

adopted, as the apparatus 60 is based on the same inventive concept as that of Embodiment 2, it can solve the problem in current technologies. Further, the apparatus 60 is located on a server, and determining through a server whether a client can recognize a first card can raise the speed of displaying target information because  
5 a server has a stronger computing power than a client does.

In an actual application, when target information and instruction information are returned to a client via a returning unit 602, the target information and instruction information may be returned to the client in a form of json character string.

Those skilled in the art should understand that the embodiments of the present application may be provided as methods, systems or computer program products, so  
10 the present application may adopt embodiments in a form of complete software, or complete hardware, or combination of software and hardware. Further, the present application may adopt a form of computer program product implemented on one or a plurality of computer-usable storage media (including but not limited to disk memory, CD-ROM and optical memory) that contain computer-usable program codes.  
15

The present application is described by referring to the flow charts and/or block diagrams of the methods, devices (systems) and computer program products according to the embodiments of the present application. It should be understood that every flow and/or block in the flow charts and/or block diagrams, and the  
20 combination of the flows and/or blocks in the flow charts and/or block diagrams may be achieved through computer program instructions. These computer program instructions may be provided to a general-purpose computer, a special-purpose computer, an embedded processor, or the processor of any other programmable data processing equipment to generate a machine so that, through the instructions executed  
25 by a computer or a processor of any other programmable data processing equipment, an apparatus for achieving the functions designated in one or a plurality of flows in the flow charts and/or one or a plurality of blocks in the block diagrams is generated.

These computer program instructions may also be stored in a computer readable memory that can guide a computer or other programmable data processing equipment  
30 to work in a specific way so that the instructions stored in the computer readable memory generate a product comprising the instruction apparatus. The instruction apparatus achieves the functions designated in one or a plurality of flows in the flow charts and/or one or a plurality of blocks in the block diagrams.

These computer program instructions may also be loaded to a computer or other programmable data processing equipment so that a series of operating steps are executed on the computer or other programmable data processing equipment to generate the processing achieved by computer, thus the instructions executed on the computer or other programmable data processing equipment provide steps for achieving the functions designated in one or a plurality of flows in the flow charts and/or one or a plurality of blocks in the block diagrams.

In a typical configuration, the computing equipment comprises one or a plurality of processors (CPU), I/O interfaces, network interfaces and internal memories.

The internal memory may be in a form of volatile memory, random access memory (RAM) and/or non-volatile memory such as read-only memory (ROM) or flash memory (flash RAM) in the computer readable media. Internal memory is an example of computer readable media.

Computer readable media include non-volatile and volatile, movable and non-movable media and may achieve information storage by any method or technology. Information may be computer readable instructions, data structure, program modules or other data. The examples of computer storage media include without limitation phase change random access memory (PRAM), static random access memory (SRAM), dynamic random access memory (DRAM) and other types of random access memory (RAM), read only memory (ROM), electrically erasable programmable read only memory (EEPROM), flash memory or other memory technologies, compact disc – read only memory (CD-ROM), digital video disk (DVD) or other optical memory, cassette type magnetic tape, tape disk memory or other magnetic storage devices or any other non-transfer media. They may be used to store information that is accessible to computer devices. According to the definition in this document, computer readable media do not include transitory media, such as modulated data signals and carriers.

It should be further noted that terms “include”, “contain” or their variants are intended to cover non-exclusive inclusion so that a process, method, commodity or equipment, including a series of elements not only includes these elements but also includes other elements not clearly set out, or also includes the elements inherent to such process, method, commodity or equipment. Unless with more limitations, the elements defined by sentence “include one.....” do not exclude that the process,

method, commodity or equipment including the elements also have other same elements.

Those skilled in the art should understand that the embodiments of the present application may be provided as methods, systems or computer program products, so  
5 the present application may adopt embodiments in a form of complete software, or complete hardware, or combination of software and hardware. Further, the present application may adopt a form of computer program product implemented on one or a plurality of computer-usable storage media (including but not limited to disk memory, CD-ROM and optical memory) that contain computer-usable program codes.

10 The foregoing descriptions are embodiments of the present application and are not intended to limit the present application. For those skilled in the art, the present application may have various changes and modifications. All modifications, identical replacements and improvements made without departing from the spirit and principle of the present application shall fall within the scope of the present application.

15

### Claims

1. A card-based information displaying method for a client, wherein the method comprises:

in response to a request for target information to be displayed on the client, obtaining,  
5 from a server, the target information to be displayed and a card type of a first card for displaying the target information, wherein the client contains a list which contains card types that are supported by a configuration version of the client;

obtaining, from the server, a second card supported by the configuration version of the client at the same time as obtaining the target information and the card type of the first card;

10 caching the second card;

matching the card type of the first card with the card types contained in the list;

according to a result of the matching, determining whether the configuration version of the client supports the card type of the first card;

when it is determined that the configuration version of the client does not support the  
15 card type of the first card, determining an association identifier corresponding to the first card, and associating the cached second card supported by the configuration version of the client with the association identifier; and

responsive to associating the cached second card with the association identifier,  
displaying the target information on the client using the cached second card.

20

2. The method according to claim 1, wherein the client contains a corresponding relation between a version identifier of the configuration version of the client and the card type supported by the configuration version, and wherein determining whether the configuration version of the client supports the card type of the first card comprises:

25 determining the version identifier of the configuration version of the client, and based on the corresponding relation, determining the card type supported by the configuration version corresponding to the version identifier; and

according to the card type of the first card and the determined card type supported by the configuration version of the client, determining whether the configuration version of the client  
30 supports the card type of the first card.

3. The method according to claim 1, wherein associating the cached second card supported by the configuration version of the client with the association identifier comprises:

according to the association identifier, determining card data of the cached second card associated with the first card; and

5 according to the card data, determining the cached second card is supported by the configuration version of the client.

4. The method according to claim 1, wherein the method further comprises:

displaying version update prompt information of the client using the cached second card.

10 5. A card-based information displaying apparatus, wherein the apparatus is located on a client and comprises: one or more processors and one or more non-transitory computer-readable memories coupled to the one or more processors and configured with instructions executable by the one or more processors to cause the apparatus to perform operations  
15 comprising:

in response to a request for target information to be displayed on the client, obtaining, from a server, the target information to be displayed and a card type of a first card for displaying the target information, wherein the client contains a list which contains card types that are supported by a configuration version of the client;

20 obtaining, from the server, a second card supported by the configuration version of the client at the same time as obtaining the target information and the card type of the first card; caching the second card;

matching the card type of the first card with the card types contained in the list;

25 according to a result of the matching, determining whether configuration version of the client supports the card type of the first card;

when it is determined that a configuration version of a client does not support the card type of the first card, determining an association identifier corresponding to the first card, and associating the cached second card supported by the configuration version of the client with the association identifier; and

30 responsive to associating the cached second card with the association identifier, displaying the target information on the client using the cached second card.

6. The apparatus according to claim 5, wherein the client contains a corresponding relation between a version identifier of the configuration version of the client and the card type supported by the configuration version, and wherein determining whether the configuration version of the client supports the card type of the first card comprises:

5 determining the version identifier of the configuration version of the client, and based on the corresponding relation, determining the card type supported by the configuration version corresponding to the version identifier; and

according to the card type of the first card and the determined card type supported by the configuration version of the client, determining whether the configuration version of the client  
10 supports the card type of the first card.

7. The apparatus according to claim 5, wherein associating the cached second card supported by the configuration version of the client with the association identifier comprises:

according to the association identifier, determining card data of the cached second card  
15 associated with the first card; and

according to the card data, determining the cached second card is supported by the configuration version of the client.

8. The apparatus according to claim 5, wherein the operations further comprise:

20 displaying version update prompt information of the client using the cached second card.

9. One or more computer-readable media configured with instructions executable by one or more processors to perform operations for a client comprising:

in response to a request for target information to be displayed on the client, obtaining,  
25 from a server, the target information to be displayed and a card type of a first card for displaying the target information, wherein the client contains a list which contains card types that are supported by a configuration version of the client;

obtaining, from the server, a second card supported by the configuration version of a client at the same time as obtaining the target information and the card type of the first card;

30 caching the second card;

matching the card type of the first card with the card types contained in the list;

according to a result of the matching, determining whether configuration version of the client supports the card type of the first card;

when it is determined that a configuration version of a client does not support the card type of the first card, determining an association identifier corresponding to the first card, and

5 associating the cached second card supported by the configuration version of the client with the association identifier; and

responsive to associating the cached second card with the association identifier, displaying the target information on the client using the cached second card.

10 10. The one or more computer-readable media according to claim 9, wherein the client contains a corresponding relation between a version identifier of the configuration version of the client and the card type supported by the configuration version, and wherein determining whether the configuration version of the client supports the card type of the first card comprises:

15 determining the version identifier of the configuration version of the client, and based on the corresponding relation, determining the card type supported by the configuration version corresponding to the version identifier; and

according to the card type of the first card and the determined card type supported by the configuration version of the client, determining whether the configuration version of the client supports the card type of the first card.

20

11. The one or more computer-readable media according to claim 9, wherein associating the cached second card supported by the configuration version of the client with the association identifier comprises:

25 according to the association identifier, determining card data of the cached second card associated with the first card; and

according to the card data, determining the cached second card is supported by the configuration version of the client.

30 12. The one or more computer-readable media according to claim 9, wherein the operations further comprise:

displaying version update prompt information of the client using the cached second card.