



US 20170010790A1

(19) **United States**

(12) **Patent Application Publication**
GLOVER et al.

(10) **Pub. No.: US 2017/0010790 A1**

(43) **Pub. Date: Jan. 12, 2017**

(54) **MULTI-STATE CARD SHARING BUTTON**

(71) Applicant: **Quixey, Inc.**, Mountain View, CA (US)

(72) Inventors: **Eric GLOVER**, Palo Alto, CA (US);
Tomer KAGAN, Sunnyvale, CA (US)

(21) Appl. No.: **14/981,778**

(22) Filed: **Dec. 28, 2015**

Related U.S. Application Data

(60) Provisional application No. 62/190,436, filed on Jul. 9, 2015.

Publication Classification

(51) **Int. Cl.**

G06F 3/0484	(2006.01)
G06F 3/14	(2006.01)
G06N 99/00	(2006.01)
G06F 3/0482	(2006.01)

(52) **U.S. Cl.**

CPC **G06F 3/04842** (2013.01); **G06F 3/0482**
(2013.01); **G06F 3/14** (2013.01); **G06N**
99/005 (2013.01)

(57)

ABSTRACT

According to some implementations of the disclosure, a method is disclosed. The method includes transmitting state indication data corresponding to a current state of first application to a card system. The method further includes receiving a card system response from the card system indicating whether the card system can provide a card corresponding to the current state and determining a sharing score of the card corresponding to the current state. The sharing score is indicative of an estimated probability that a user of the device will share the card corresponding to the current state. The method includes displaying a first selection element overlaying a graphical user interface of the first application when the sharing score is above a threshold. The method includes displaying a second selection element overlaying the graphical user interface when the sharing score is below the threshold and above a second threshold.



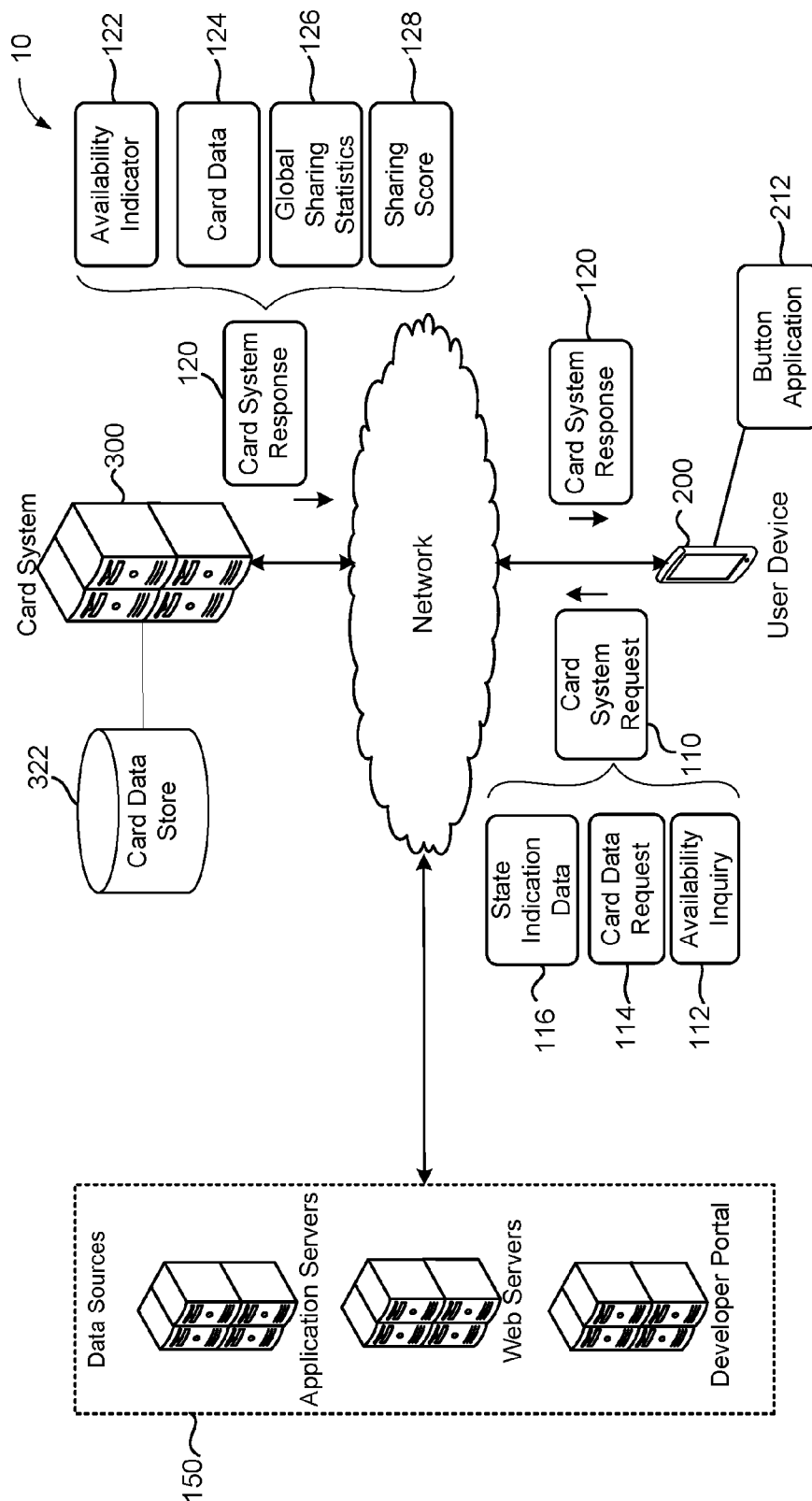


FIG. 1A

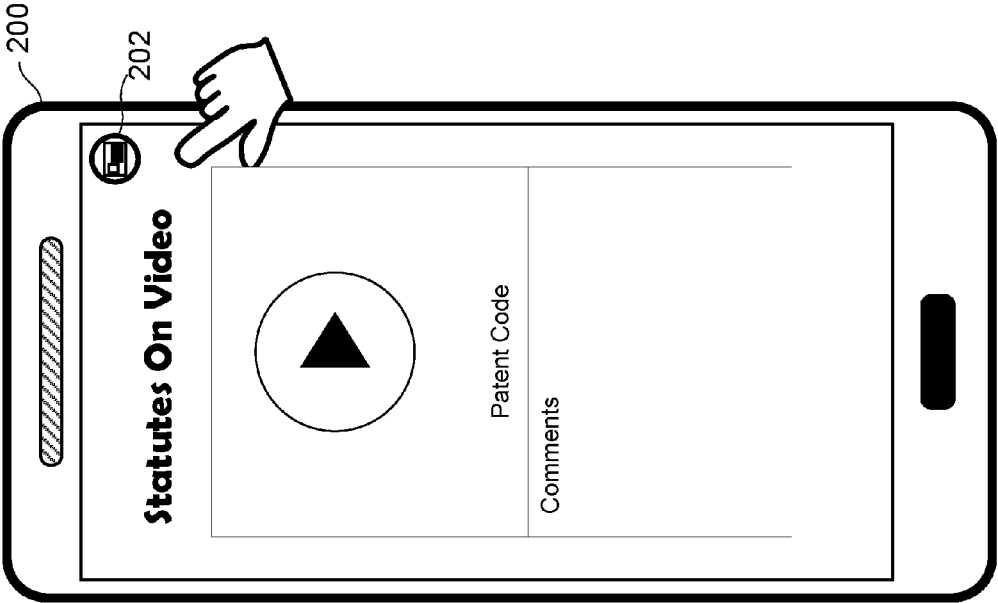


FIG. 1C

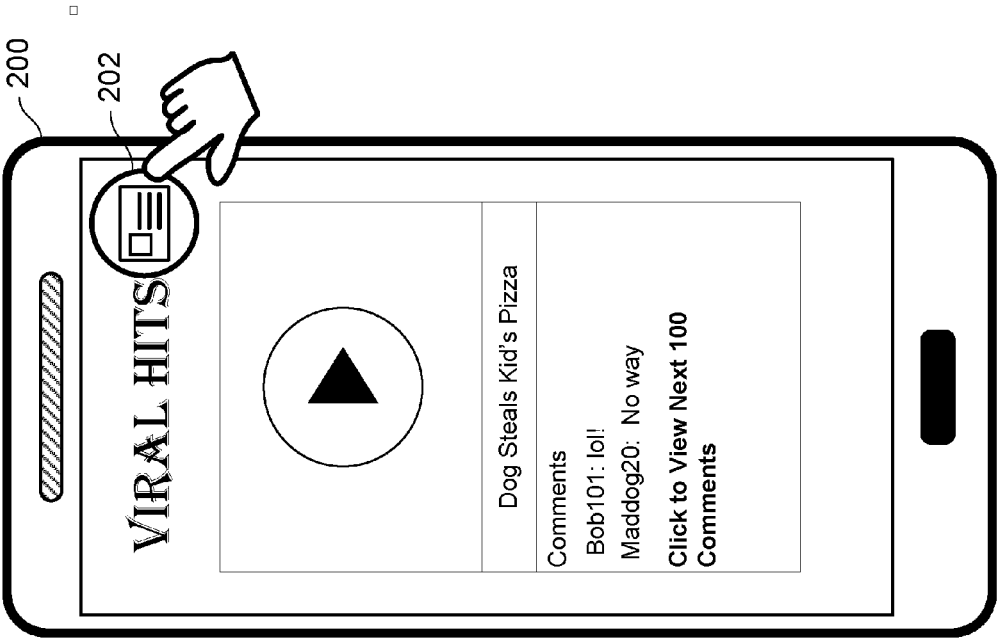


FIG. 1B

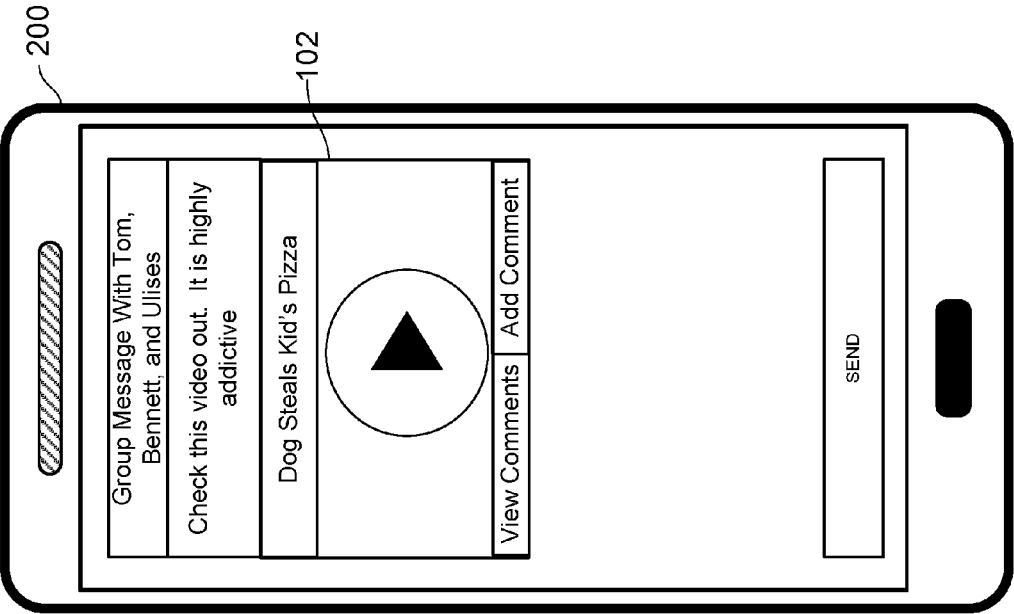


FIG. 1E

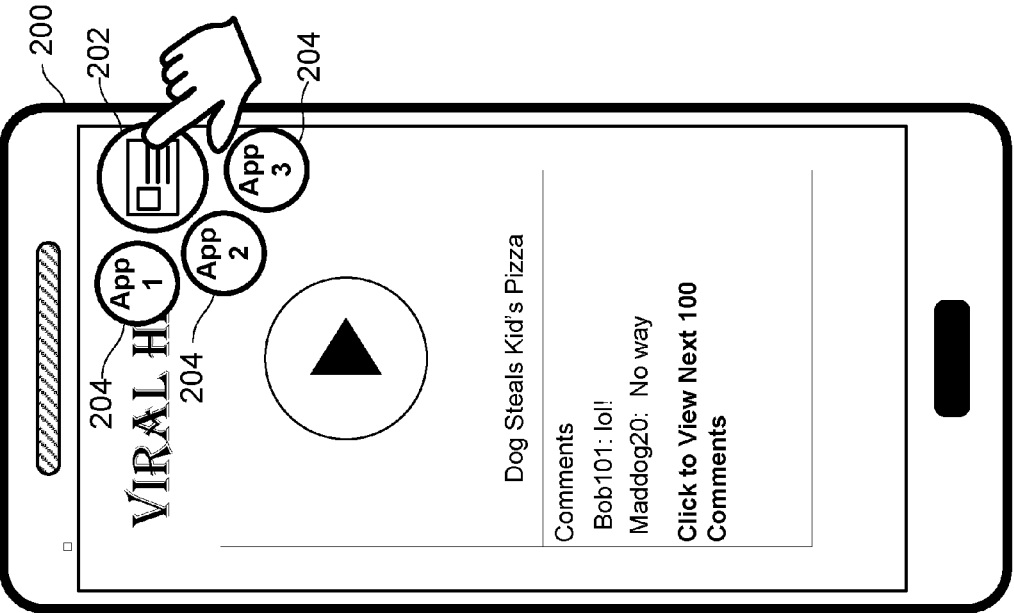


FIG. 1D

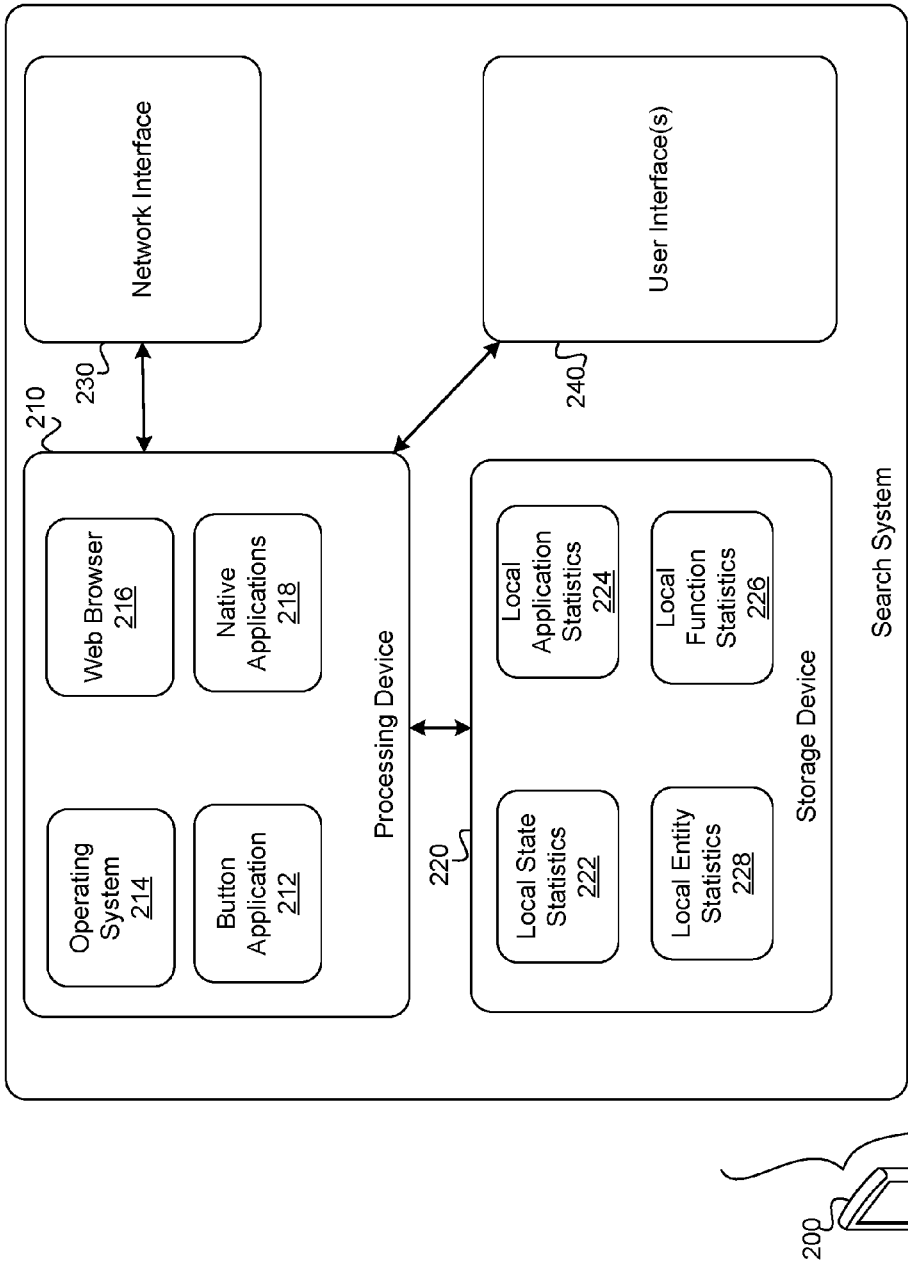


FIG. 2

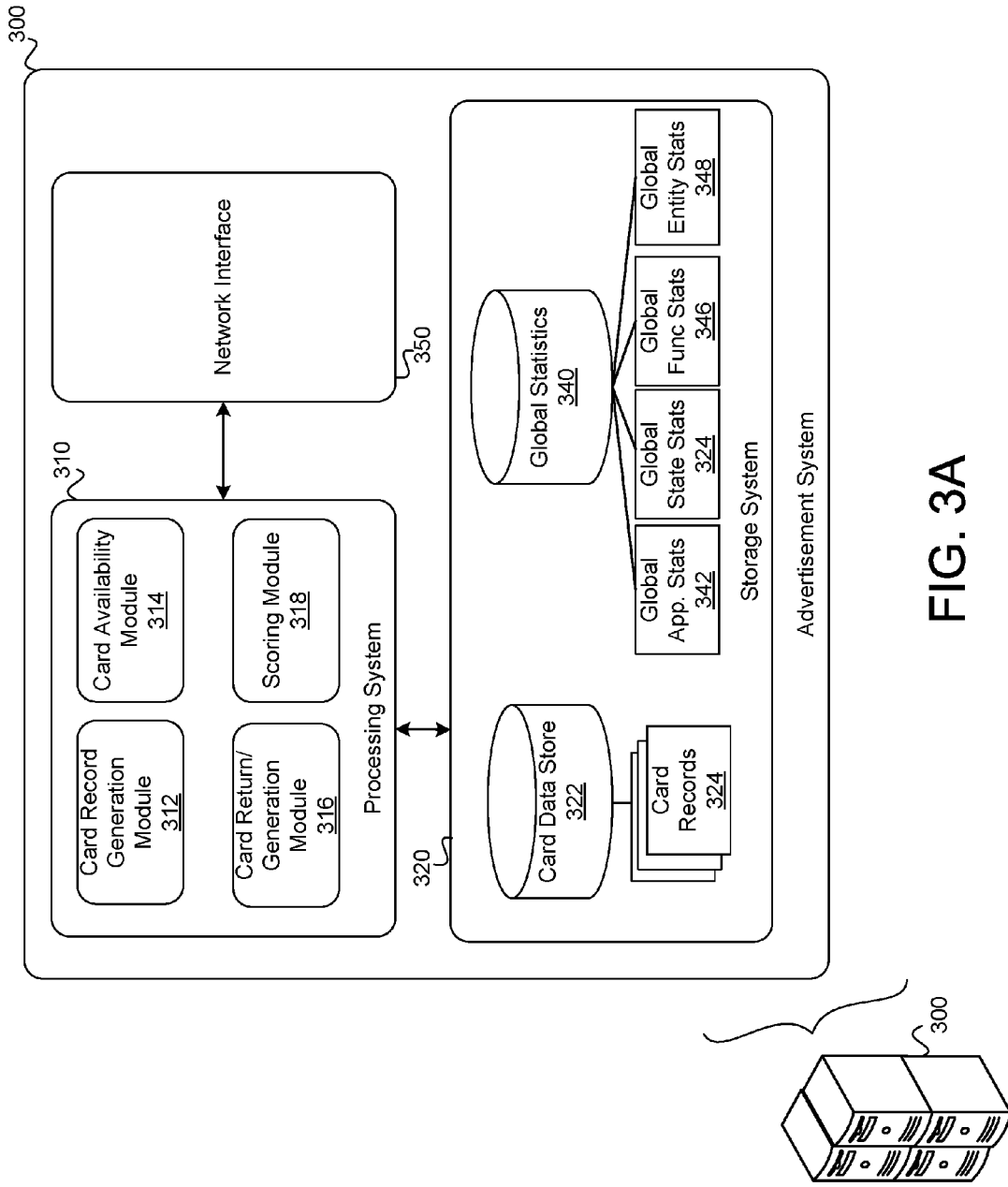


FIG. 3A

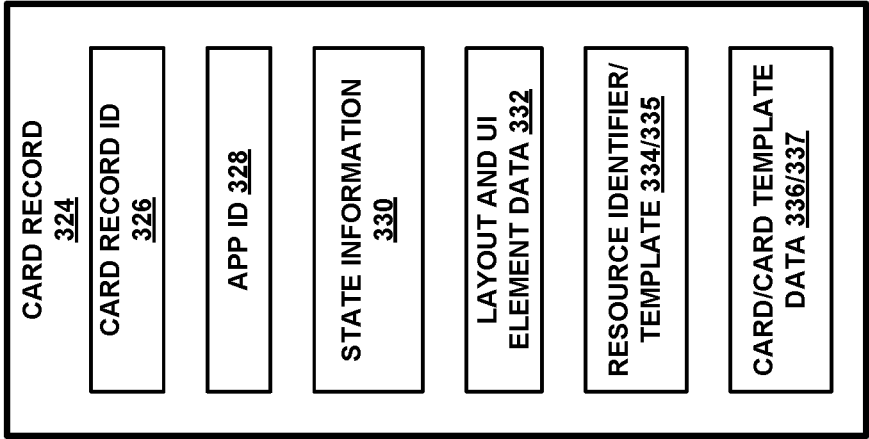
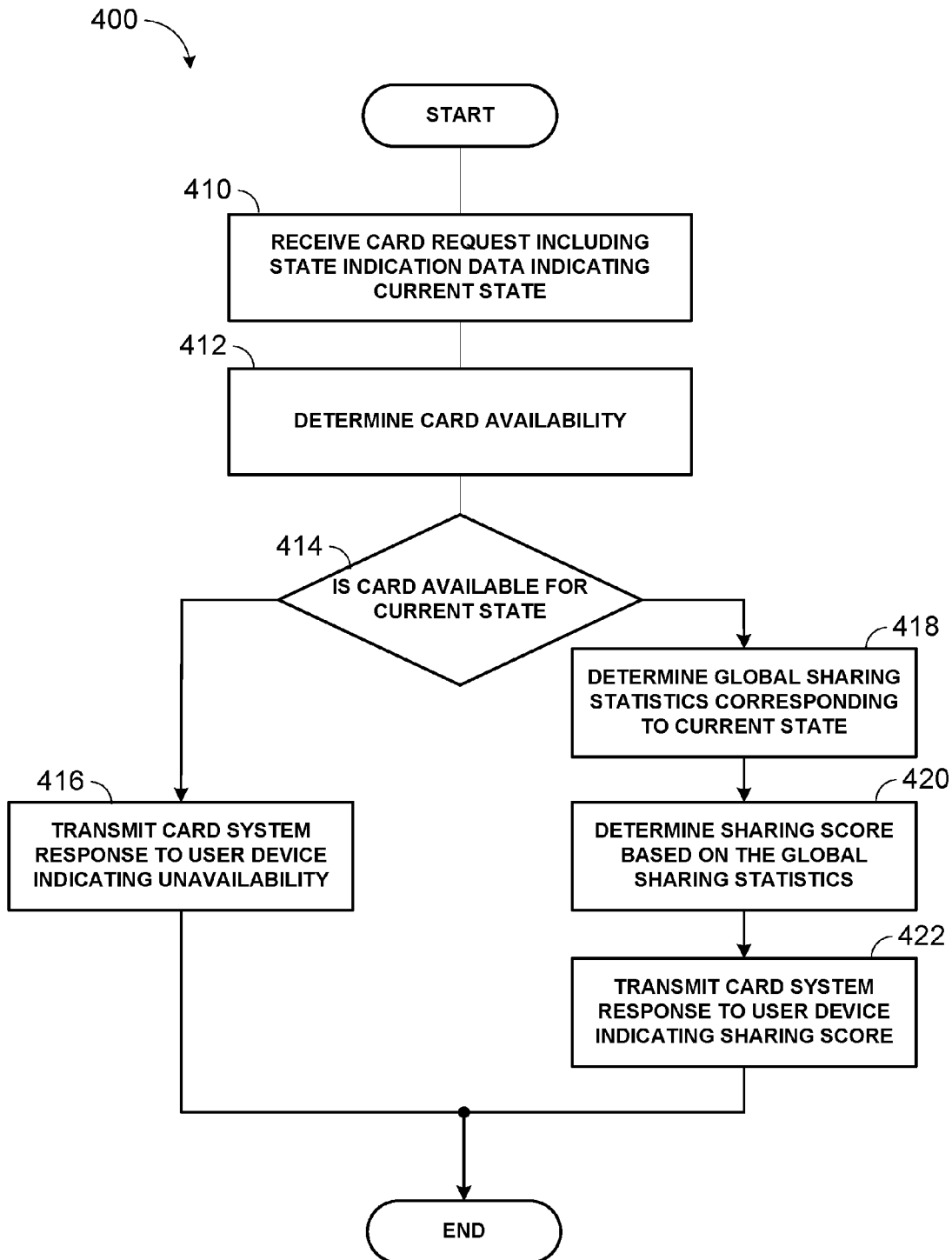


FIG. 3B

**FIG. 4**

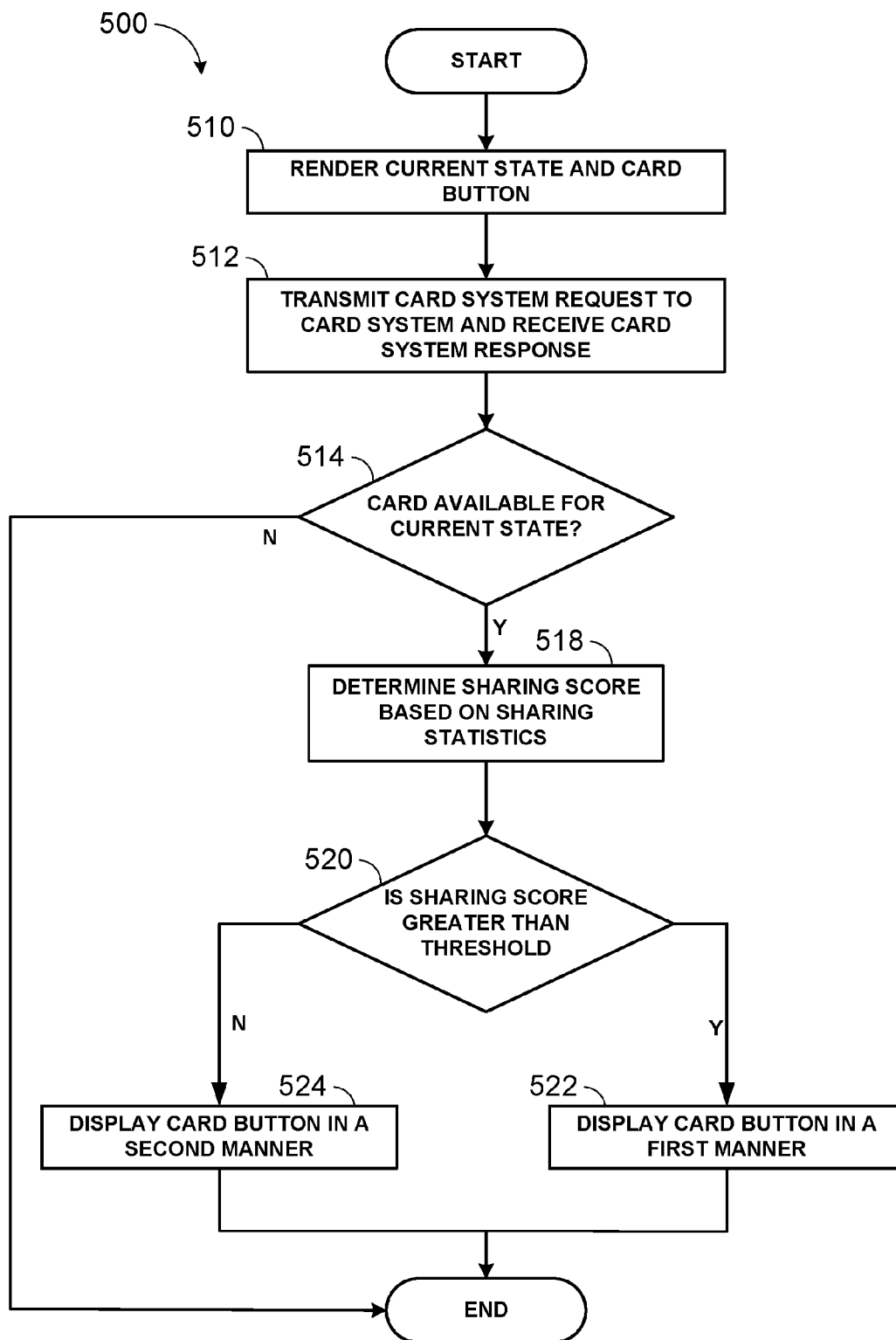


FIG. 5

MULTI-STATE CARD SHARING BUTTON

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. provisional patent application 62/190,436 filed on Jul. 9, 2015, the contents of which are hereby incorporated by reference in their entirety.

TECHNICAL FIELD

[0002] The present disclosure is directed to techniques for displaying a card button used to share an active application state into a target application.

BACKGROUND

[0003] The wide-spread adoption of mobile user devices has led to a sharp increase in the consumption of electronic content, including videos, on-line articles, music, and the like. Different software applications provide different content and functionality. One trend that differentiates electronic content from traditional forms of content is the amount of content that is shared between people. A funny video may be made available on one day, and through “sharing” via social media outlets, the video may be viewed by a million people in a few hours. Typically a user will share a URL with others users to share the content.

SUMMARY

[0004] According to some implementations of the disclosure, a method is disclosed. The method includes transitioning to a current state of a first application, obtaining state indication data corresponding to the current state of the first application, and transmitting the state indication data to a card system. The method further includes receiving a card system response from the card system indicating whether the card system can provide a card corresponding to the current state and determining a sharing score of the card corresponding to the current state. The sharing score is indicative of an estimated probability that a user of the device will share the card corresponding to the current state. The method includes displaying a first selection element overlaying a graphical user interface of the first application when the sharing score corresponding to the current state is above a threshold. The method includes displaying a second selection element overlaying the graphical user interface of the first application when the availability indicator indicates that the card system can provide the card and the sharing score is below the threshold and above a second threshold.

[0005] According to some implementations of the present disclosure, a method is disclosed. The method includes receiving, by a card system, a card system request from a user device including state indication data indicating a state of an active application of the user device and identifying a card record based on the state indication data. The card record corresponding to the state indicated by the state indication data and defining card data of a card corresponding to the state. The method includes retrieving a set of global share statistics based on the card record from a global sharing statistics data store. The global sharing statistics correspond to a group of users. The method further includes determining a sharing score corresponding to the card based on the global sharing statistics. The sharing score is indicative of an estimated probability that a user of the user device

will share the card corresponding to the state. The method further includes transmitting the sharing score to the user device.

[0006] The details of one or more implementations of the disclosure are set forth in the accompanying drawings and the description below. Other aspects, features, and advantages will be apparent from the description, drawings, and claims.

DESCRIPTION OF DRAWINGS

[0007] FIG. 1A illustrates an example environment that includes a card system in communication with user devices and data sources.

[0008] FIGS. 1B-1C illustrate example manners by which a card button may be rendered by a user device.

[0009] FIGS. 1D-1E illustrate an example of insertion of a card representative of a current state of an active application into a target application via interaction with a card button.

[0010] FIG. 2 illustrates an example user device and components thereof.

[0011] FIG. 3A illustrates an example card system and example components thereof.

[0012] FIG. 3B illustrates an example of a card record.

[0013] FIG. 4 illustrates a set of operations of a method for displaying a card button.

[0014] FIG. 5 illustrates a set of operations of a method for inserting a card in a target application.

[0015] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0016] Systems and techniques directed to inserting a card corresponding to an application state of an active application into another application (referred to as a “target application”) are herein disclosed. A card may refer to a graphical user interface (GUI) element that represents an application state. A card may include text, images, icons, and/or user interface input elements. A card may be a link that, when selected, links to the represented state of the application. Additionally or alternatively, a card may access functions of the represented application. For example, the card may include computer-readable instructions, such that when the card is selected by the user, cause the user device to make a request to the application or to an API request to a third-party server. The third-party server may return content that is displayed within the card. In some implementations, a card may present information from the active application state in a manner that is consistent with the look and feel of the active application. In other implementations, the card may express the functionality in a specific, consistent user-friendly layout. The techniques of the present disclosure may provide a streamlined user experience for bookmarking application information and/or sharing application information with other users.

[0017] According to some implementations of the present disclosure, a user device may execute a button application and a plurality of native applications. The button application may be in communication with a remote card system. The remote card system may serve cards representing states of various applications. The button application may display an interactive button (referred to as a “card button”) that overlays (or is otherwise integrated in) a GUI of an appli-

cation that is currently accessed by the user device (referred to as the “active application”). The appearance of the card button may be varied based on a sharing score associated with the card. A sharing score can indicate an estimated probability or likelihood of a user sharing the card. The card button may be displayed in a different manner (e.g., larger/smaller or bolded/faded) depending on whether the sharing score is above or below a threshold. To the extent a state of an application cannot be represented by a card (e.g., the card system does not support the application), the button application will not display the card button or may display the card button in a third manner. For instance, if the user is viewing a popular movie trailer that has been shared many times by other users, the card button may be displayed in a prominent manner, while if the user is viewing a page that shows the “cast and crew” of the movie, the card button may be displayed in a less prominent manner assuming that the user, or users in general, are less likely to share “cast and crew” application states. If, however, the user is playing a game that is not recognized by the card system, the button may be faded or not displayed all together, thereby indicating that the card system cannot generate a card representing the video.

[0018] As mentioned, the sharing score of a card represents a likelihood (e.g., estimated probability value) that a user will share the card. The sharing score may be calculated using global sharing statistics or local sharing statistics (e.g., user-specific statistics) relating to the state represented by the card, the application, the function performed at the state, and the entities relating to the state. Global sharing statistics relating to the state may indicate how many times a group of users have visited the state and how many times users have shared a card representing the state. Global sharing statistics relating to the application may indicate how many times the group of users accesses the application and how many times those users share cards representing any state of the application. Global sharing statistics relating to a function performed at a current state (e.g., show a restaurant menu) may indicate how often the group of users access a state of any application that performs the function and how many times those users share a card representing a state that performs the function. Global sharing statistics relating to an entity indicate how many times the group of users were presented with a state of any application that relates to an entity defined in the current state and how many times users shared cards representing the states relating to the entity.

[0019] Local sharing statistics relating to a state may indicate how many times the current user has visited the state and how many times the user has shared a card representing the state. Local sharing statistics relating to the application may indicate how many times the user has accessed the application and how many times the user has shared a card representing any state of the application. Local sharing statistics relating to a function performed at a current state (e.g., view movie trailer) may indicate how often the user accesses a state of any application that performs the function and how many times the user shares a card representing a state that performs the function. Local sharing statistics relating to an entity indicate how many times the user is presented with a state of any application that relates to an entity defined in the current state and how many times the user shared cards representing the states relating to the entity. Furthermore, user-specific statistics may also include

how many times a user accesses any sharable state and how many times the user shares any cards.

[0020] The card system may track the global sharing statistics. Individual user devices may track local sharing statistics of the user of the respective device, and may report the local sharing statistics for inclusion in the global sharing statistics. In these scenarios, users may be required to expressly agree to allow such data to be monitored. Furthermore, this data may be anonymized to protect user privacy.

[0021] In some implementations, the card system may determine the sharing score of a state of an application. The card system may determine the sharing score based on the global sharing statistics. Furthermore, in some implementations, the card system may take into account context surrounding the user (e.g., location, time of day, demographics, etc.). The card system may utilize rules-based scoring, a machine learned scoring model, and/or a scoring function to calculate the sharing score.

[0022] In some implementations, the user device may determine the sharing score of a state of an application. The button application may determine the sharing score based on the local sharing statistics and/or the global sharing statistics. Furthermore, the user device may take into account context surrounding the user (e.g., location, time of day, demographics, etc.). The button application may utilize rules-based scoring, a machine learned scoring model, and/or a scoring function to calculate the sharing score.

[0023] If a card is available for the current underlying application state, the card system may provide an availability indicator to the user device indicating that a card is available for a given state. Depending on the sharing score of the application state, the card button may be displayed prominently or less conspicuously. In response to a user selecting the card button, the user device can request the card (which may be encoded in a card object) for the current. Alternatively, the card system can transmit the card when available with or in lieu of the availability indicator. The button application can receive the card from the card system and embed the received card into a target application, such as a messenger application, a social networking application, or other application. In some scenarios, the user may select the target application from a set of target applications that may receive the card.

[0024] FIG. 1A illustrates an example environment that includes a card system 300 in communication with user devices 200 and data sources 150. A card system 300 is a set of one or more computing devices that provides cards to requesting user devices 200. As described herein, the user device 200 can execute a button application 212 that requests a card from the card system 300. The card is representative of a current state of the active application on the user device 200. If the current state can be represented by a card, the card system 300 can transmit the card to the requesting user device 200. The user device 200 may insert a received card into a target application.

[0025] In some implementations, the card system 300 includes a card data store 322. The data stored in the card data store 322 may be represented by card records 324 (FIGS. 3A and 3B). The card system 300 obtains the data that is stored in the card records 324 from the data sources 150. In some implementations, the card system 300 generates card records based on data retrieved from the data sources 150. The data that is collected may be representative

of various states of applications or collections of states. The data that is collected may include a name of an application, content presented at a state of an application, a layout of an application at a given state (or information regarding the layout) and access mechanisms or templates to generate access mechanisms with which to request the content presented at the state on a user device. The data that is collected may also include data representing a layout of the card and/or a template to generate a card. Data sources 150 can include web servers, application servers, and/or developer portals, whereby a developer portal may be associated with the card system 300.

[0026] The card system 300 can crawl and scrape documents collected from web servers. For example, the card system 300 may crawl websites (web applications) and scrape the content from the crawled websites. Additionally or alternatively, the card system 300 may perform static and/or dynamic analysis of a native application to identify the content and/or the layout of an application. Additionally or alternatively, the card system 300 may generate a card dynamically in real-time from the native application or API representing the native application to get up-to-date data from the application. Additionally or alternatively, an application developer or another party may upload documents, templates, layout files, and the like via a developer portal corresponding to the card system 300. In these scenarios, the card system 300 may extract features of the crawled states, including the function performed at the state and the entities (and entity types) referenced in the state.

[0027] The user device 200 can transmit a card system request 110 to the card system 300. The card system request 110 may include an availability inquiry 112, a card data request 114, and/or state indication data 116. An availability inquiry 112 may be a value that requests the card system 200 to determine whether the card system 300 can return a card object corresponding to a state indicated by the state indication data 116. The state indication data 116 is data that is indicative of a state of an active application being executed by the user device 200. The data included in the state indication data 116 may vary for different implementations or applications. In some scenarios, the state indication data 116 can explicitly indicate the current state of the application. In these scenarios, the state indication data 116 may include a resource identifier that indicates the current state. For example, the state indication data 116 may include a resource identifier, such as a URL or an application resource identifier. In some implementations, the state indication data 116 may further include a function performed at the state and/or the entities referenced at the state. The function performed at the state and the entities may be scraped from the metadata of the current state of the active application (e.g., from tagged data in HTML or XML code).

[0028] The card data request 114 may be a request for the card data (as opposed to the availability of the card). The card data request 114 may be transmitted with the state indication data 116. The card data request 114 may be transmitted subsequent to the availability inquiry 112 or in lieu of the availability inquiry 112.

[0029] The card system 300 operates to handle card system requests 110 from the user devices 200 and to transmit card system responses 120 in response to the card system requests 110. The card system response 120 can include an availability indicator 122, card data 124, global sharing statistics 126 associated with the state indication data, and/or

a sharing score 128 associated with the state indication data. In some implementations, an availability indicator is a flag that indicates whether the card system 300 can provide a card corresponding to the state indication data 116 provided in the card system request 110. Card data 124 is data that is used to present a card. Card data 124 may include the content to be displayed in the card. Additionally, the card data 124 may be bound to a layout (e.g., a layout file) that defines the look and feel of the card. The card data 124 may be transmitted with one or more access mechanisms used to access a state of an application represented by the card. An access mechanism is a string or address that an operating system or application can use to set an application to a specific state. Examples of access mechanisms include web resource identifiers (e.g., URLs), application resource identifiers, and scripts.

[0030] In response to the card system request 110, the card system 300 may determine whether a card corresponding to the state indicated by the state indication data 116 can be delivered to the user device 200. Furthermore, either automatically or in response to the card system request 110, the card system 300 can determine the card data 124 that corresponds to the state indication data 116. The card system 300 may include the card data 124, the layout file, and the access mechanisms used to access an application state corresponding to the card in a card object. A card object is any data structure that stores the card data 124 that the user device 200 uses to generate a card. The card object may be a file, such as a .json file or an XML file. The card system 300 may communicate the card object to the user device in the card system response 120. The user device 200 may then render a card 102 in the target application in response user interactions with the card button.

[0031] The global sharing statistics 126 correspond to the state indication data 116. The global sharing statistics 126 can define state-specific sharing statistics (or “state statistics”), application-specific sharing statistics (or “application statistics”), function-specific sharing statistics (or “function statistics”), and/or entity-specific sharing statistics (or “entity statistics”). Global sharing state statistics can define how often users access the state indicated by the state indication data and how often users shared a card representing the state. Global application statistics can define how often users accessed any sharable states of the application indicated by the state indication data 116 and how often users shared cards corresponding to those sharable states. Global function statistics can define how often users accessed sharable states that perform the function performed at the state indicated by the state indication data and how often users shared those cards. The global function statistics can be across multiple applications. For instance, if a function corresponds to displaying a menu of a selected restaurant, the global function statistics may apply to any sharable state of any application that performs the displaying a menu function. The global entity statistics can define how often users access states that pertain to a particular entity (or entity type) relating to the state indicated by the state indication data and how often users shared a card corresponding to any sharable state of any application relating to the particular entity (or entity type). For instance, if the global state statistics corresponds to an article about a particular movie, the global entity statistics may correspond to states of any applications that reference the movie,

including a state of another application that shows the movie trailer or a state of yet another application that describes the plot.

[0032] In some implementations, the card system 300 includes the global sharing statistics 126 in the card system response 120. In these implementations, the user device 200 utilizes the global sharing statistics 126 to calculate a sharing score of the card, assuming a card corresponding to the current application state can be delivered. The user device 200 utilizes the global sharing statistics and/or local sharing statistics to calculate the sharing score. The local sharing statistics are specific to the user (or potentially users) of the user device 200. The local sharing statistics may include state statistics, application statistics, function statistics, and/or entity statistics. These may be substantially similar to the global sharing statistics, but based on observations at the user device. In the scenario where the user device 200 collects local sharing statistics, the user device 200 can periodically report such statistics to the card server 300 so that the card server 300 may aggregate the global sharing statistics. In such implementations, user device 200 may anonymize the local sharing statistics, so as to protect the privacy of the user.

[0033] In some implementations, the card system 300 includes the sharing score 128 in the card system response 120. In these implementations, the card system 300 may utilize the global sharing statistics 126 to calculate the sharing score 128. The card system 300 may utilize any number of scoring techniques to determine the sharing score 128, including rules-based scoring, weighted scoring functions, and machine learned scoring models.

[0034] In some implementations, the user device 200 may send the single card system request 110 and the card system 300 can respond with a single card system response 120. In these implementations, if a card is available for a given state, the card system 300 can include the card data 124 in lieu of the availability indicator 122. In this way, the user device 200 can treat the card data 124 as an affirmative availability indicator 122. In other implementations, the card system request 110 and/or the card system response 120 can be broken into multiple separate communications. For example, initially, the user device 200 may transmit the card system request 110 containing an availability inquiry 112 to the card system 300 with corresponding state indication data 116. The card system 300 may respond to the availability request 112 with the availability indicator 122 (e.g., a flag) that indicates whether the card 102 corresponding to the state is available.

[0035] The user device 200 may visually indicate to the user that the card 102 is available by displaying a card button (or analogous interface element). The user device 200 checks the sharing score against a threshold to determine whether sharing a card is relatively likely or unlikely. If the sharing score is greater than the threshold, the user device 200 displays the card button in a relatively more prominent or conspicuous manner. For example, the user device 200 may display the card button 202 in a larger format and/or a colorful or bold format. If the sharing score is below the threshold, the user device 200 displays the card button in a less prominent or conspicuous manner. For example, the user device 200 may display the card button in a relatively smaller format and/or in a semi-transparent or faded format. In this way, the user is not bothered by the card button if he/she is unlikely to use the card button.

[0036] FIGS. 1B-1C illustrate example manners by which the user device 200 can render the card button 202 based on a sharing score of the card. It is noted that in the implementations of FIGS. 1B and 1C it is assumed that the card system 300 can deliver a sharable card to the user device 200, should the user request a card from the card system 300. In the illustrated examples, the user device 200 can render the card button 202 in a conspicuous state (e.g., card button 202a in FIG. 1B) or can render the card button 202 in an inconspicuous state (e.g., card button 202b in FIG. 1C).

[0037] In the example of FIG. 1B, the active application (e.g., Viral Videos) is displaying a video that appears to be very popular, and thus, may be shared by many users or may have been shared by the user. Furthermore, the active application itself may have many videos that are shared, being that it shows viral videos. Thus, the sharing score associated with the state of the active application exceeds a threshold. In this way, the user device 200 displays the card button 202 in a conspicuous manner, so as to make sharing the card button 202 easier for the user.

[0038] In the example of FIG. 1C, the active application (Statute Videos) is displaying a video relating to the United States Patent Code. While this may be of interest to some people, this video, and videos on this application in general (e.g., relating to statutes on video), and/or the entities discussed (e.g., "Patent Code") may have low instances of being shared, despite being sharable. Thus, the sharing score of the application state shown in FIG. 1C is likely to be below the threshold. Accordingly, the user device 200 displays the card button 202b in a less conspicuous manner, as compared to the card button 202a displayed in FIG. 1B.

[0039] Even though the card button 202 is displayed in a less conspicuous manner, the user may nonetheless select the card button 202b if he or she decides they would like to share a card corresponding to the state of the active application. Although the example of FIGS. 1B and 1C alter the appearance of the card button 202 by changing the size of the card button 202, the user device 200 can alter the appearance in different manners, including darkened/lightened graphical renderings, using different colors in the card button 202 (e.g., brighter colors when more likely), or different UI experiences (e.g., a vibration or sound when a card is likely to be shared). The user may press on the card button 202 to request the card 102 or to display the card 102 (in the event that a card object is communicated in the card system response 120).

[0040] In response to a user selection of the card button 202, the user device 200 inserts a card into a target application. In some implementations, the user device 200 may prompt the user to select a target application in which the user device 200 is to insert a card. FIG. 1D illustrates an example GUI that allows a user to select a target application. In this example, the user device 200 displays three options in the form of target buttons 204. The user can select a target application, for example, by sliding the card button 202 to one of the target buttons 204 representing the target application or by pressing on one of the target button 204. For instance, the user device may allow the user to select from a messaging application (e.g., the SNAPCHAT application or the FACEBOOK MESSENGER application or the device's MMS service or EMAIL application or other near-field communications (NFC) application) or a social networking application (e.g., the TWITTER application, the FACEBOOK application, or the PINTREST application).

The target applications (e.g., target buttons **204**) that the user can select from may be a curated list or may be adjusted according to local sharing statistics and/or global sharing statistics. Furthermore, in some implementations, the target buttons **204** may indicate deeper states of target applications. For instance, the target applications may allow a user to send a card to a specific contact or to store the card in a specific bookmark folder. Further, in some implementations, the list of available applications to select/launch may be based on the current context of the user and the current state. For example, if the current application is a sports application, the list of target applications can change to include specific social messaging apps, e.g., the FACEBOOK MESSENGER application, while if the current context is a technology article, the set of target applications may change to include a different set of apps, such as the LINKEDIN application. Once a user selects a target application, the user device **200** can insert the card in the user device **200**.

[0041] FIG. 1E illustrates an example of the card **102** being inserted into a target application. In the example of FIG. 1E, the user has selected the card button **202** shown in FIGS. 1B and 1D. In response to the selection of the card button **202**, the user device **200** may request the card data **124** of the card from the card system **300**. In some implementations, the user device **200** transmits a request for the card data **124** from the card system **300** upon the user selection of the card button **202**. In other implementations, the card system **300** transmits the card data **124** automatically upon determining that there is a card available for the current state. In these implementations, the user device **200** may temporarily store the card data **124**, such that if the user selects the card button **202** the user device **200** does not need to make an additional card system request **110**. In response to the selection of the target application, the user device **200** inserts the card **102** corresponding to the current state of the active application in a target application. The target application is configured to handle how the card is displayed to the user. Launching an application by the user device **200** may include the button application **212** instructing an operating system **214** to launch the target application. This may include launching the target application and passing the card data **124** representing the card to the target application. In FIGS. 1D and 1E, the target application is a messaging application and the card **102** is inserted in a group message.

[0042] FIG. 2A illustrates an example user device **200** and an example set of components of the user device **200**. In the illustrated example, the user device **200** includes a processing device **210**, a storage device **220**, a network interface **230**, and a user interface **240**. The user device **200** may include additional components not shown in FIG. 2. The components of the user device **200** may be interconnected by, for example, a bus or other communication circuitry.

[0043] The processing device **210** can include one or more processors that execute computer-executable instructions and associated memory (e.g., RAM and/or ROM) that stores the computer-executable instructions. In implementations where the processing device **210** includes more than one processor, the processors can execute in a distributed or individual manner. The processing device **210** can execute the operating system **214**, a button application **212**, a web browser **216**, and one or more native applications **218**.

[0044] The storage device **220** can include one or more computer-readable mediums (e.g., hard disk drives, solid state memory drives, and/or flash memory drives). The

storage device **220** can store any suitable data that is utilized by the operating system of the user device **200**. The storage device **220** can be in communication with the processing device **210**, such that the processing device **210** can retrieve any needed data therefrom.

[0045] In some implementations, the storage device **220** stores local sharing statistics, such as local state statistics **222**, local application statistics **224**, local function statistics **226**, and/or local entity statistics **228**. As previously mentioned, the local sharing statistics identify statistics relating to the users of the user device **200** and his/her propensity to share cards given a condition (e.g., state, application, function, entity). Local state statistics **222** can define how often the user has accessed the state indicated by the state indication data and how often the user has shared a card representing the state. Local application statistics **224** can define how often the user has accessed any sharable states of the application indicated by the state indication data **116** and how often the user has shared cards corresponding to those sharable states. Local function statistics **226** can define how often the user has accessed sharable states that perform the function performed at the state indicated by the state indication data **116** and how often the user has shared those cards. As is the case with global function statistics, local function statistics **226** can be across multiple applications. The local entity statistics **228** can define how often the user has accessed states that pertain to a particular entity (or entity type) relating to the state indicated by the state indication data **116** and how often the user has shared a card corresponding to any sharable state of any application relating to the particular entity (or entity type). The storage device **220** may store additional data as well, including other statistics pertaining to the user's propensity to share cards. For example, the storage device **220** may store statistics relating to the user's propensity to share cards given certain context/conditions (e.g., time of day, location, etc.).

[0046] The network interface **230** includes one or more devices that are configured to communicate with the network **150**. The network interface **230** can include one or more transceivers for performing wired or wireless communication. Examples of the network interface **230** can include, but are not limited to, a transceiver configured to perform communications using the IEEE 802.11 wireless standard, an Ethernet port, a wireless transmitter, and a universal serial bus (USB) port.

[0047] The user interface **240** includes one or more devices that receive input from and/or provide output to a user. The user interface **240** can include, but is not limited to, a touchscreen, a display, a QWERTY keyboard, a numeric keypad, a touchpad, a microphone, and/or speakers.

[0048] The button application **212** may be embodied as a set of computer-readable instructions. The button application **212** can be a stand-alone application that is downloaded (e.g., from the digital distribution platform) and installed on the user device **200**. Additionally or alternatively, the button application **212** can be preloaded onto the user device **200** before a user purchases the device (e.g., preloaded by an OEM, such as a telecommunications company). In some implementations, the functionality attributed to the button application **212** herein may be included in one or more native applications **218** included on the user device **200**, instead of, or in addition to, being included as a stand-alone application. For example, the functionality of the button application **212** may be provided in a software developer's

kit (SDK), whereby application developers can utilize the SDK to include the button application 212 functionality in their respective applications. In some implementations, the button application 212 may be included in the operating system, as a feature included by the operating system provider and/or device manufacturer.

[0049] The button application 212 can determine a state of an active application (either native application or a web application). Put another way, the button application 212 captures the state indication data 116 indicating the state of the active application. An active application represents an application (native or web) currently being accessed on the user device 200. The button application 212 can then determine whether the card system 300 can provide a card representing the state of the active application. In some implementations, the button application 212 identifies a resource identifier (e.g., URL or application resource identifier) corresponding to the state of the active application. For example, if the content is provided in an electronic document such as an HTML document, the document may contain a web resource identifier or an application resource identifier tagged in the document. The button application 212 can transmit the resource identifier to the card system 300 in the card system request 110.

[0050] In some implementations, the button application 212 can capture information such as a function performed at a state and/or entities corresponding to the state being accessed by the user device 200. In these implementations, the button application 212 can determine the entities and/or functions by analyzing the current state of the application. In some of these implementations, the button application 212 may reference the data used to render the current state. Often this data is provided in a tagged document such as an XML document or an HTML document. The data may be tagged according to known schemas. The button application 212 can parse the data to identify tags associated with entity types and/or function types. The button application 212 can then read in the information following the identified tags to determine the entities and/or function performed by the state. In other implementations, the card server 300 maintains this information and returns the function and/or entity information to the button application 212 in response to receiving the state indication data 116 in the card request 110.

[0051] The button application 212 can transmit the state indication data 116 to the card system 300 in the card system request 110. In response to the card system request 110, the card system 300 returns the card system response 120. The card system response 120 can include the availability indicator and/or card data 124 to render the card. Additionally, when a card corresponding to the state indication data is available, the card system response 120 includes the global sharing statistics 126 corresponding to the available card or the sharing score 128 corresponding to the available card.

[0052] In implementations where the card system response 120 includes the global sharing statistics 126, the button application 212 calculates a sharing score based on the global sharing statistics and/or local sharing statistics. The button application 212 determines the sharing score 128 if the availability indicator 122 indicates that the card system 300 can deliver a card corresponding to the state indication data 116 or if the card system response 120 includes the card data 124 corresponding to the state indication data 116. The

button application 212 can calculate the sharing score 128 in a number of different manners.

[0053] In some implementations, the button application 212 can determine a sharing score using a machine-learned scoring model. The machine-learned scoring model may include supervised and/or unsupervised data that tracks the decisions of users to share a card when presented with different applications states as well as the sharing statistics (global and/or local) corresponding to the different states. In some implementations, the machine-learned scoring model can be a gradient-boosted decision tree. In these implementations, a machine-learned scoring model receives a feature vector relating to the card and outputs a sharing score of the card. The feature vector may include one or more scoring features. The scoring features may include any or all of the global sharing statistics 126, any or all of the local sharing statistics relating to the state indication data, and/or any context values. The global sharing statistics 126 may be represented in ratios and/or as raw numbers. Context values may include a location of the user device 200 or a time of day. Over time, the machine-learned scoring model may learn that the user is less likely to share a card if he or she is at the office and it is during work hours (e.g., 9 AM-5 PM). The button application 212 can generate the feature vector based on the global and local state statistics relating to the current state of the active application, global and local application statistics relating to the active application, global and local function statistics relating to the function performed by the active application at the current state, global and local entity statistics relating to the entities corresponding to the current state of the active application, and/or any context parameters known to the user device 200. The machine-learned scoring model receives the feature vector and outputs the sharing score of the card.

[0054] In some implementations, the button application 212 can calculate the sharing score based on a set of scoring rules. The scoring rules may be conditional statements that define an effect on the sharing score when the condition is satisfied. The conditional statements may be based on the global sharing statistics, the local sharing statistics, and/or context parameters. For example, one rule may state if the current application is "A", then increase the sharing score by a certain value, and if the local application statistics of the application "A" are much greater (e.g., the ratio is greater than 0.10) than the global application statistics of application A, then multiply the sharing score by another value (e.g., multiply by 1.25). The scoring rules can be hand curated and/or adjusted based on experimental data. The rules may be indexed by application identifiers, state identifiers, entity identifiers, and/or function identifiers. The button application 212 can retrieve the relevant rules based on the application identifier of the active application, a function identifier of the current function of the active application, entity identifiers of any entities relevant to the current state of the active application, and/or a state identifier of the current state (e.g., a URL). The button application 212 can then determine the sharing score based on the retrieved rules.

[0055] In some implementations, the button application 212 can calculate the sharing score using a scoring function. The scoring function may be a weighted linear combination that is based on the global and local sharing statistics. For example the scoring function may take the form:

$$\text{Score(Available Card)} = w_1 \frac{GS_{State}}{GV_{State}} + w_2 \frac{GS_{App}}{GV_{App}} + w_3 \frac{GS_{Func}}{GV_{Func}} + w_4 \frac{LS_{Entity}}{LV_{Entity}} + w_5 \frac{LS_{State}}{LV_{State}} + w_6 \frac{LS_{App}}{LV_{App}} + w_7 \frac{LS_{Func}}{LV_{Func}} + w_8 \frac{LS_{Entity}}{LV_{Entity}}$$

Where $w_1, w_2, \dots, w_8$ are weights, the GS values are global shares of the state, the application, the function, and the entity respectively, the GV values are global views of the state, the application, the function, and the entity, respectively, the LS values are local shares of the state, the application, the function, and the entities, respectively, and the LV values are local views of the state, the application, the function, and the entities. The weights may be hard coded and/or learned from experimental data. Furthermore, while the equation above assumes at least one view of a state, other scoring functions may be used. For example, a score of an available card may be calculated according to:

$$w_1 f_1(GS_{State}, GV_{State}) + w_2 f_2(GS_{App}, GV_{App}) + w_3 f_3(GS_{Func}, GV_{Func}) + w_4 f_4(GS_{Entity}, GV_{Entity}) + w_5 f_5(LS_{State}, LV_{State}) + w_6 f_6(LS_{App}, LV_{App}) + w_7 f_7(LS_{Func}, LV_{Func}) + w_8 f_8(LS_{Entity}, LV_{Entity})$$

Where $f_1 \dots f_8$ are simple functions that map to a value (e.g., $f_n(x, y)$ map to $(x+c_1)/(y+c_2)$, where c_1 and c_2 are constants to smooth the value and ensure no division by zero. The scoring function may be provided to the user device from the card system 300 and may be periodically updated as the values of the weights change.

[0056] In implementations where the card system response 120 includes neither the sharing score 128 or the global sharing statistics 126, the button application 212 calculates the sharing score based on the local-statistics. In these implementations, the button application 212 can calculate the sharing score in a number of different manners. The button application 212 can calculate the sharing score using a machine-learned scoring model, a set of scoring rules, or a scoring function. In these implementations, the sharing score may be based on the local sharing statistics and/or the context data.

[0057] The button application 212 determines a manner by which to display the card button 202 based on the sharing score. The button application 212 can compare the sharing score to a threshold to determine the manner by which to display the card button 202. The value of the threshold may be hard coded into the button application 212 and/or may be periodically updated by the card system 300. The value of the threshold may be determined experimentally or may be assigned to either promote or discourage sharing. For instance, if the card system 300 wants to encourage more people to share, the threshold may be lowered (e.g., for scores normalized between 0 and 1, the threshold is lowered from 0.1 to 0.01). When the sharing score exceeds a threshold, the button application 212 displays the card button 202 in a conspicuous manner (e.g., card button 212a in FIG. 1B). When the sharing score is below the threshold and a card is available for the current state, the button application 212 displays the card button 202 in an inconspicuous manner (e.g., card button 212b in FIG. 1C).

[0058] The button application 212 may display the card button 202 (also referred to as a “button”) over the graphical user interface of the active application (i.e., overlaid upon the current state). If the user selects the button 202, the button application 212 can obtain and insert the card in a

target application. A target application represents another application, which may or may not be running, in which a card can be inserted in response to a user interacting with the card button 202. The button application 212 may insert a card into a default target application or may allow the user to select a target application from a plurality of potential target applications (e.g., native applications 218). In some cases, a plurality of different ones of the additional native applications 218 can be active/target applications depending on what applications the user is using at the current time and what applications the user selects to use as target applications or depending on the current context of what the user is doing, e.g., reading an article, playing a game, listening to music or shopping.

[0059] In some implementations, the button application 212 maintains the local sharing statistics. Each time the user accesses a sharable state, the button application 212 can increment the local sharing statistics corresponding to the sharable state. In operation, the user accesses a state of an active application. In response, the button application 212 captures the state indication data 116 and communicates the state indication data 116 to the card system 300. The button application 212 receives the card system response 120, which indicates whether the current state of the active application is a sharable state. If the current state is a shareable state, the button application 212 can update the local state statistics 222 defining the number of times the user has accessed the current state. Similarly, the button application can update the local application statistics 224 defining the number of times the user accessed sharable states of the active application. The button application 212 may also update the local function statistics 226 defining the number of times the user accessed a state performing the function of the current state on any application. The button application 212 can also update the local entity statistics 228 defining the number of times the user accessed states of applications corresponding to the entities represented or relating to the current state using any application. Should the user opt to share a card 202 representing the current state, the button application 212 can update the local sharing statistics accordingly. In such a scenario, the button application 212 can update the local state statistics 222, the local application statistics 224, the local function statistics 226, and/or the local entity statistics 228 to indicate that the user shared the card. Furthermore, the button application 212 can periodically report the local sharing statistics to the card system 300 for inclusion in the global sharing statistics. In these implementations, the button application 212 only reports the local sharing statistics should the user consent to such reporting. Furthermore, the button application 212 can anonymize the local sharing statistics so as to protect the privacy of the user.

[0060] FIG. 3A illustrates the card system 300 and example components thereof. In the illustrated example, for requesting and inserting the cards 102 in a target application. In the example of FIG. 3, the card system 300 includes a processing system 310, a storage system 320, and a network interface 340. The card system 300 may include additional components not explicitly shown.

[0061] The processing system 310 is a collection of one or more processors that execute computer readable instructions. In implementations having two or more processors, the two or more processors can operate in an individual or distributed manner. In these implementations, the processors may be connected via a bus and/or a network. The proces-

sors may be located in the same physical device or may be located in different physical devices. Moreover, the processors may be located at different physical locations. The processing system may execute a card record generation module 312, a card availability module 314, a card retrieval/generation module 316 (hereinafter “card generation module”), and a card scoring module 318.

[0062] The network interface device 240 includes one or more devices that perform wired or wireless (e.g., Wi-Fi or cellular) communication. Examples of the network interface devices include, but are not limited to, a transceiver configured to perform communications using the IEEE 802.11 wireless standard, an Ethernet port, a wireless transmitter, and a universal serial bus (USB) port.

[0063] The storage system 320 includes one or more storage mediums. The storage mediums may be any suitable type of computer readable mediums, including but not limited to read-only memory, solid state memory devices, hard disk memory devices, and optical disk drives. The storage devices may be connected via a bus and/or a network. Storage devices may be located at the same physical location (e.g., in the same device and/or the same data center) or may be distributed across multiple physical locations (e.g., across multiple data centers). The storage system 320 stores a card data store 322 and a global shared statistics data store 340, example contents of which are discussed in greater detail below.

[0064] The card data store 322 can store a plurality of card records 324, each card record 324 corresponding to a different deliverable card 102 or group of cards 102. The card data store 322 may include one or more indexes (e.g., inverted indexes) that index the card records 324.

[0065] FIG. 3B illustrates an example of the card record 324. The card record 324 can represent data for one or more potential cards. In some examples, the card record 324 can include data related to a single card 102 (e.g., a single state of an app). In other examples, the card record 324 can include data related to the group of cards 102 (e.g., a card template for a group of different states of a specific application). For example, one card template can be used for a variety of different related states of an application. In a specific example, one card may have the same layout with different content for different states (e.g., one card layout can be reused for different restaurants in a single restaurant app). In such examples, the card may be populated with content from the different states to represent the respective states.

[0066] The card record of FIG. 3B includes a card record identifier (card record ID) 326, an application identifier (application ID) 328, state information 330, layout and UI element data 332, an access mechanism 334/template 335, and card data 336/card template data 337. The card record ID 326 may be any value or string that uniquely identifies the card record 324 in the card data store 322. The application ID 328 indicates the application with which the record is associated. The application ID 328 may include the name of the application or a numerical value that maps to the application.

[0067] The state information 330 can include text, images, or other data related to an application state (i.e., the state associated with the card). The state information 330 includes text and/or images that are presented to the user when the user is in the application state. The state information 330 may be obtained from a document representing a state (e.g., an HTML document) by scraping the document. The state

information 330 may be structured or unstructured data. The state information 330 may further include a title of the state. For instance, if the state corresponds to a news article, the state information 330 may include the title of the application. In this way, a card corresponding to the news article, may include the title of the application state. The state information 330 may be scraped from the application (e.g., from a document representing the state) or may be uploaded to the card system 300 by the developer of the application via a portal.

[0068] In some implementations, the state information 330 further stores metadata that is relevant to the state. The metadata may include entity data relation to entities referenced or corresponding to the state or states represented by the card record 324. An entity may be any logical object (e.g., a person, place, or thing). Entities can include tangible and non-tangible objects. An entity identifier is a value that represents the entity. An entity identifier may be a unique numerical value, the name of the entity, or a unique alphanumeric string. For example, if the card record 324 corresponds to a particular state of a restaurant review application (e.g., a page of a particular restaurant), the state information 330 may define an entity identifier of the restaurant. Similarly, if the card record 324 can represent all of the pages that review restaurants on the application, the state information 330 may define the entity identifiers of all of the restaurants that are reviewed in the application. Any entity is also of one or more entity types. An entity type defines a category of the entity.

[0069] The metadata may further define a function identifier that identifies that corresponds to the state or set of states represented by the card record 324. A function may define a type of the state. Put another way, the function of a state defines what the application does when put at that state. Examples of functions include, but are not limited to, “display menu,” “book a flight,” “make a reservation,” “watch a video,” “present article,” and etc. The function identifier may be a unique value, a name of the function (e.g., “display menu” or “watch video”) or alphanumeric string that represents the function.

[0070] The card record 324 can include layout and UI element data 332. The layout and UI element data 332 can describe the arrangement of the graphical user interface of the application to which the card record 324 corresponds. In particular, the layout and UI element data 332 can define the location of objects (e.g., where text or images appear) when presented in the state as well as the type and location of the UI input elements when presented in the state. The layout and UI element data 332 can also include text and other data associated with UI elements, such as text associated with buttons, labels, text boxes, etc. Before receiving card system requests, the card system 300 (e.g., the card record generation module 312) can retrieve (e.g., crawl/scrape) the data for the card records 324 from the applications. The card record generation module 312 may also analyze the applications (e.g., using static/dynamic analysis) in order to determine the layout and UI elements for the states of the applications. The combinations and arrangements of the UI elements and other data related to the UI elements make a fingerprint for identifying the current state. In some implementations, the card record generation module 312 includes layout and UI element data 332 in the card record 324, whereby the layout and UI element data 332 can be used to match the respective card record 324 to the state indication

data **116** received with the card system request **110**. In some implementations, the layout and UI element data **332** may be a value that is indicative of arrangement of the graphical user interface. For example, the layout and UI element data **332** may be a hash value that is the output of a low collision hash function that receives the layout and UI element data **332** and outputs a unique (or likely unique) value based thereon. In these implementations, the card record **324** may be indexed by the value.

[0071] The card record **324** may include one or more access mechanisms **334** and/or one or more access mechanism templates **335**. As previously discussed, the access mechanisms **334** may include web resource identifiers, application resource identifiers, and/or scripts. The access mechanism **334** can be used (e.g., by the user device) to open the application state. In cases where the card record **324** is for a single application state, the card record **324** may include the single access mechanism **334** (or multiple access mechanisms for the same state for different platforms). In cases where the card record **324** is for multiple application states, the card record **324** may include the access mechanism template **335** that can be used to generate different access mechanisms **334**. For example, the access mechanism template **335** may have parameter fields that receive values in order to form a complete access mechanism **334**. In a specific example, a URL template for the IMDB application may be `http://www.imdb.com/title/{movieID}`. In this specific example, inserting a movieID may complete the access mechanism template **335**, thereby forming the access mechanism **334**. An example access URL for the particular feature film on IMDB application is: `http://www.imdb.com/title/tt0096895/`, where `tt0096895` is the movieID. In this example, the value `tt0096895` may be used to generate the URL given the template. Although HTTP web URLs are given as examples above, other types of access URLs that are app/platform specific (e.g., Android/iOS specific) can be used. In some implementations, the card data store **322** can include template rules for filling out the access mechanism templates **335** based on data included in the card system request (e.g., in the state indication data).

[0072] The card record **324** includes card data **336** and/or card template data **337**. The card data **336** includes the content used to render the card **102** in the target application, such as text and images. The card data **336** may include content that is presented at a state of an application. For example, the card data **336** may include a name of the application, a title of the state, a text snippet from the state of the application, images presented at the state of the application, and/or an icon of the application. As opposed to the state information, which may include content from the state of the application that is not presented, the card data **336** includes the content that is presented in the rendered card **102**.

[0073] The card template **337**, when present, may be filled in to generate a card. Put another way, the card template **337** defines the layout of the card. The layout may be defined in a layout file and binding. The layout file defines where each element appears in a rendered card. For example, the layout file may define the location of the title of a state, the location of an icon of the application, and locations of text and/or images presented in the card. In this way, the card template **337** defines the look and feel of the card **102**. The binding can define the data types that are received by the layout file. The card system **300** can utilize the binding to bind the text,

images, and access mechanisms that correspond to a state to the card template **337**, thereby generating a card object containing the card data **124**. In some implementations, the layout file may further define UI input elements appearing in the card, whereby the cards served by the card system **300** can include input elements. In this way, the card system **300** can serve cards that link to dynamic application states. The card generation module **316** can generate a card object based on the card data **336** and/or the card template data **337**.

[0074] In some implementations, the card record generation module **312** generates the card records **324** based on data obtained from the data sources **150**. Data sources can include web servers, application servers, and/or developer portals, whereby a developer portal may be associated with the card system **300**. The card record generation module **312** can actively collect data from the data source **150** (e.g., from a web server or applications server) and/or can passively receive data from a data source (e.g., from application developers via a developers portal). For example, the card record generation module **312** may employ a collection of web crawlers that crawl applications. In some implementations, the crawlers can systematically identify documents (e.g., states) to request from applications (e.g., web servers or application servers). The applications (e.g., web servers or application servers) return the requested documents. In these implementations, the card record generation module **312** identifies the access mechanism corresponding to requested document with the requested document.

[0075] Furthermore, the card record generation module **312** scrapes the document, thereby obtaining the data comprising the state information **330** and/or the card data **336**. The card record generation module **312** may utilize a known schema to parse and scrape the document, whereby the scraped data may be used to structure the state information **330** and/or the card data **336** in the card. For example, the card record generation module **312** can determine the title, the images, icons, and other content from the tags in the requested document and the content following the tags. In the event the document corresponds to a newly discovered application state, the card record generation module **312** can create a new card record **324** and include the scraped data in the card data **336** and/or the state information of the new card record **324** and the corresponding access mechanisms in the new record **324**. In the event the application state is a previously known application state, the card record generation module **312** can update a previously generated card record **324** with the scraped data. The card record generation module **312** can utilize default card templates when the content is obtained by crawlers.

[0076] Additionally or alternatively, the card record generation module **312** may connect to a portal where developers may upload their content to the portal. In doing so, the developers can provide one or more documents and access mechanism(s) corresponding to each document, whereby an access mechanism defines the location from which the document may be retrieved. In this way, the card record generation module **312** can determine the state information **330** and the access mechanisms **334**. Furthermore, the developer can define which content is to be included in the card data **336**. Application developer may also provide layout templates and/or access mechanism templates. In this way, the application developer may have control over the content displayed in the card and/or the look and feel of the card.

[0077] The card record generation module 312 may implement additional or alternative techniques to obtain content to populate the card data store 322. For example, the card record generation module 312 may perform static and/or dynamic analysis of a native application to identify the content and/or the layout of an application. The card record generation module 312 generates card records 324 based on the collected data.

[0078] Referring back to FIG. 3A, the storage system 320 may also store a global sharing statistics data store 340. The global sharing statistics data store 340 may include any global sharing statistics, including but not limited to: global state statistics 342, global application statistics 344, global function statistics 346, and/or global entity statistics 348. In some implementations, the global sharing statistics data store is indexed by state identifiers, application identifiers, function identifiers, and/or entity identifiers.

[0079] Each known sharable state (i.e., a state that can be represented by a card) may have a set of global state statistics 342 associated therewith. The global sharing statistics 344 of a particular state may include a state identifier of the state (e.g., a URL or application resource identifier), a number of views of the state by users and a number shares of cards representing the state. In some implementations, the number of views and the number of shares are combined into a ratio

$$\left(\text{e.g., } \frac{\text{Views}}{\text{Shares}} \right).$$

Furthermore, in some implementations, a smoothing function may smooth the global sharing statistics 342 such as

$$\left(\frac{\text{views} + c_1}{\text{shares} + c_2} \right)$$

where c_1 is a value that is less than or equal to c_2 and a default score of a state with no shares or views is

$$\frac{c_1}{c_2}.$$

[0080] Each application known to have sharable states may have a set of global application statistics 344 associated therewith. The global application statistics 346 of an application may include an application identifier of the application, a number of times any sharable states of the application were viewed by users and a number of times cards corresponding to any sharable states of the application were shared by users. In some implementations, the number of views and the number of shares are combined into a ratio

$$\left(\text{e.g., } \frac{\text{Views}}{\text{Shares}} \right).$$

[0081] Each function known to be performed by at least one sharable state may have a set of global function statistics 346 associated therewith. The global function statistics 342 of a function may include a function identifier of the

function, a number of times any sharable states that perform the function (regardless of application) were viewed by users and a number of times cards corresponding to an of those sharable states were shared by users. In some implementations, the number of views and the number of shares are combined into a ratio

$$\left(\text{e.g., } \frac{\text{Views}}{\text{Shares}} \right).$$

[0082] Each known referenced by at least one sharable state may have a set of global entity statistics 348 associated therewith. The global entity statistics 348 of an entity may include an entity identifier of the entity, a number of times any sharable states that relate to the entity (regardless of application) were viewed by users and a number of times cards corresponding to any of those sharable states were shared by users. In some implementations, the number of views and the number of shares are combined into a ratio

$$\left(\text{e.g., } \frac{\text{Views}}{\text{Shares}} \right).$$

[0083] The card availability module 314 and the card generation module 316 can operate to handle card system responses 120 from the user devices 200 and to transmit the availability indicators 122 and/or card data 124 of available cards 102 to the requesting user devices 200. The card availability module 314 determines whether the card data 124 corresponding to the state indication data 116 is available in response to the card system request 110 received from the user device 200. The card generation module 316 can generate a card object in response to the card data request 114. A card object is any data structure that stores the card data 124 that the user device 200 uses to generate the card 102. A card object can include instructions and/or data for rendering the card 102 (e.g., a layout file with the card data 124 bound thereto), content that is to be displayed in the card 102, and/or one or more access mechanisms for accessing a represented state of an application. The card generation module 316 can transmit the card object to the user device 200. The user device 200 may then render the card 102 in the target application in response user interactions with the card button.

[0084] In operation, the card availability module 314 receives the card system request 110 from the user device 200. The card system request 110 may include an availability inquiry 112 and the state indication data 116. The availability inquiry 112 may be a value that indicates that the request is for a determination as to whether the card 102 can be delivered corresponding to a state indicated by the state indication data 116. The state indication data 116 may be a resource identifier (e.g., a URL or an application resource identifier).

[0085] In response to receiving an availability inquiry 112 and the state indication data 116 in the card system request 110, the card availability module 314 determines whether the card generation module 316 can provide a card 102 corresponding to the state indicated by the state indication data 116. The card availability module 314 searches the card record data store 322 using the state indication data 116.

[0086] The inverted index outputs the card record ID 326 of the card record 324 if the state indication data 116 corresponds to a known state. In such a scenario, the card availability module 312 can provide the affirmative availability indicator 122 in the card system response 120 to the user device 200. In this way, the card system response 120 indicates to the user device 200 that a card is available for the state indicated by the state indication data 116. In some implementations, the availability indicator 122 may be a flag in the card system response 120 that is set to a value (e.g., TRUE) if a card is available. As discussed, the button application 212 can utilize the availability indicator 112 to adjust the appearance of the button 202 displayed at the user device 200. If the inverted index does not output a card record ID, the card availability module 316 determines that the card system 300 cannot output a card to the user device 200. In this scenario, the card availability module 314 can include an availability indicator that indicates that the card system 300 cannot transmit a card corresponding to the state indication data 116.

[0087] When the card system 300 can provide a card corresponding to the state indication data 116, the card system 300 can either include either the global sharing statistics 126 corresponding to the card or the sharing score 128 corresponding to the card in the card system response 120. In implementations where the card system 300 includes the global sharing statistics 126, the card generation module 316 can retrieve the global sharing statistics 126 from the global sharing statistics data store 340. In some implementations, the card generation module 316 can retrieve the global sharing statistics from the global sharing statistics data store 340 using a state identifier, an application identifier, a function identifier, and/or one or more entity identifiers corresponding to the available card. The state identifier, the application identifier, the function identifier, and any entity identifiers may be obtained from the card record 324 corresponding to the available card and/or the state indication data 116.

[0088] In implementations where the card system 300 provides a sharing score of an available card, the scoring module 318 can calculate a sharing score based on the global sharing statistics 126 associated with the available card. In some implementations, the scoring module 318 can further consider context signals received from/associated with the requesting user device 200, such as location or a time of day. The scoring module 318 may score an available card in a number of different manners. The scoring module 318 may implement rules-based scoring, a machine learned scoring model, and/or a scoring function to calculate the sharing score of an available card. The scoring module 318 may output the determined sharing score to the card generation module 316, which includes the sharing score in the card system response 120.

[0089] In some implementations, the scoring module 318 utilizes a machine-learned scoring model to score an available card. In these implementations, the scoring module 318 retrieves the global sharing statistics relating to the available card from the global sharing statistics data store 340. The scoring module 318 can generate a feature vector based on the retrieved statistics. For instance, for an available card, the scoring module 318 can include the global state statistics 342, global application statistics 344, the global function statistics 346, and the global function statistics 348 pertaining to the available card in the feature vector. Furthermore,

in some implementations the scoring module 318 can include one or more context signals in the feature vector, such as a time of day or location of the user device 200 that provided the card system request 110. The scoring module 318 feeds the feature vector into the machine-learned scoring model. The machine-learned scoring model outputs the sharing score of the available card.

[0090] In some implementations, when a card is available the card generation module 316 generates a card object corresponding to the state indication data 116 and transmits the card object in the card system response message 120 that is responsive to an availability inquiry 112. In other implementations, the card generation module 316 receives a secondary card system request 110 including the card data request 114 when the user selects a button 202 displayed by button application 212 executing on the user device 200. The secondary card system request 110 includes the state indication data 116 and the card data request 114. In response to the card data request 114, the card generation module 316 generates a card object corresponding to the state indication data 116 and transmits the card object to the user device 200. As previously discussed the machine-learned scoring model may be a gradient-boosted decision tree, a Bayesian model, or a neural network.

[0091] The scoring module 316 can calculate the sharing score in other manners. The scoring module 316 may utilize a set of scoring rules to score an available card. In these implementations, the rules can take the global sharing statistics and/or context parameters into consideration to determine the sharing score of the available card. In other implementations, the scoring module 318 utilizes a scoring function to score the available card. For instance, the scoring module 318 may calculate a weighted linear combination of the global sharing statistics.

[0092] In generating a card object, the card generation module 316 can instantiate a new card object from a card object template. The card object may be a container such as a .JSON file or an XML file. The card generation module 316 retrieves the card record 324 corresponding to the state indication data 116 from the card record data store 322. In some implementations, the card generation module 316 can obtain the card data 336 from the card record 324. The card generation module 316 also obtains the card template (e.g., a layout file and binding) from the card record 324. The card generation module 316 may bind the card data 336 to the card template 337. For example, the card generation module 316 may bind the card data 336 to a layout file using a binding. The card data 336 may define the content that is presented in the card and the layout of the card, which defines where and how the content is presented. The card generation module 316 encodes the card data 336 and the card template into the new card object. Additionally, the card generation module 316 may obtain the one or more access mechanisms 334 defined in the card record 324. The card generation module 316 can encode the access mechanism(s) into the new card object. The card generation module 316 can provide the card object to the user device 200. The card generation module 316 may include the card object in a card system response 120. The card generation module 316 may also include the global sharing statistics 126 or the sharing score 128 in the card system response 120. The card system 300 may transmit the card system response 120 to the user

device 200 or to an intermediate device, which in turn transmits the card system response 120 to the user device 200.

[0093] FIG. 4 illustrates an example set of operations for a method 400 for responding to the card system request 110. The method 400 is described with respect to the card system 300. The method 400 may be executed by other suitable devices as well.

[0094] At 410, the card availability module 314 receives the card system request 110. The card system request 110 may include the state indication data 116. As previously indicated, the state indication data 116 may include a resource identifier or other data that is indicative of the state. In the case that the user device 200 does not transmit a resource identifier, the state indication data 116 may include alternate information to identify the state, such as data relating to the layout of the application and/or the content defined at the current state. The card availability module may further include additional information such as an application identifier, a function identifier, and/or one or more entity identifiers.

[0095] At 412, the card availability module 314 determines whether the card generation module 316 can provide the card 102 to the user device 200 given the state indication data 116. In some implementations, the card availability module 314 can search the card data store 322 for the card record 324 indexed by a matching resource identifier. The card data store 322 may be indexed such that the card records 324 are indexed in an inverted index that relates resource identifiers or other state indication data 116 to the card records 324. In this way, the card data store 322 may be searched by querying the inverted index.

[0096] If the card availability module 314 determines that the system 300 cannot deliver a card for the current state, the card availability module 314 transmits the availability indicator 122 that indicates that no card is available, as shown at 414. As a result, the user device 200 does not display the card button 202.

[0097] If the card availability module 314 determines that a card can be delivered at 412, the card system 300 delivers the availability indicator 122 and/or a card object to the user device 200. At 416, the scoring module 318 retrieves the global sharing statistics from the global sharing statistics data store 340. As previously indicated, the global sharing statistics may include global state statistics 342, global application statistics 344, global function statistics 346, and/or global entity statistics 348. The scoring module 318 can utilize a state identifier of the current state, an application identifier of the current state, a function identifier corresponding to the current state, and/or entity identifiers corresponding to the current state to retrieve the global sharing statistics. In some implementations, the scoring module 318 can encode the global sharing statistics into the card system response 120. In these implementations, the user device 200 is tasked with determining the sharing score.

[0098] In implementations where the card system 300 scores available cards, the scoring module 318 determines the sharing score based on the global sharing statistics corresponding to the available card, as shown at 418. In some of these implementations, the card system 300 may additionally take context parameters received from or associated with the user device into account when scoring the available card. As previously discussed, the scoring module

318 may implement rules-based scoring, a machine-learned scoring model, or a scoring function to score the available card.

[0099] At 420, the card generation module 316 transmits the card system response 120 to the user device 200. The card generation module 316 can include the sharing score or the global sharing statistics in the card system response 120. The card generation module 316 can further include the availability indicator 122 and/or card data 124 encoded in a card object in the card system response 120. In some implementations, the card generation module 316 utilizes the card data 336, the access mechanisms 334 (e.g., web resource identifier and/or application resource identifier), and the card layout 337 of the card record 324 to generate the card object. In these implementations, the card records 324 define single states of applications. The card generation module 316 includes the card data 336 and the resource identifier in the card layout 337 to generate the card object. In other implementations, a card record 322 can be used to generate multiple card objects 124 corresponding to different states of an application. In these implementations, the card generation module 316 can obtain the access mechanism template 335 and may generate the access mechanism 334 for the state indicated by the state indication data 116 based on the content of the indicated state and the accessed state. For instance, if the current state is a state of a flight booking application showing different flight options for a given airport code pair and date pair, the card generation module 316 can insert values corresponding to the airport codes and the dates indicated at the state into the resource identifier template 337 to generate an access mechanism that links to the dynamic state that corresponds to the provided values. The card generation module 316 may utilize a schema of the application to generate the values. For instance, the format of dates may be defined by the schema of the flight booking application (e.g., “dept_date[day].[month].[year]” vs. “dept_datel day[month]-[year]”). The card generation module 316 may include the access mechanism in the card object 122. The card generation module 316 may include multiple access mechanisms if the state is accessible by different application editions of the software application. The card generation module 316 may transmit the card object to the user device 200.

[0100] FIG. 5 illustrates an example set of operations of a method 500 for displaying the card button 202. The method 500 is discussed with respect to the user device 200 and the card system 300 described above. The method 500 may be performed by other systems as well.

[0101] At 510, the user device 200 renders a current state of an active application. Typically, the user device 200 renders the current state in response to a user command. For instance, a user may command the user device 200 to transition states within an application or to switch applications altogether. In one example, a user may be reading a first article on a news application and may switch to another article. In another example, a user may be searching for flights on a travel application. In response to entering the search parameters (e.g., airport codes and dates) and entering “search,” the travel application may switch from a search state to a result page state, where the user can see the various flight options. In another example, the user may select a search result and may be linked to a state of an application.

[0102] At 512, the button application 212 sends the card system request 110 to the card system 300. The card system

request **110** may include the state identification data **116**. The state identification data may be a resource identifier corresponding to the current state. Alternatively, the state identification data **116** may be information that defines a layout and/or the content defined in the current state. Additionally, in some implementations the button application **212** may include an application identifier, a function identifier, and/or entity identifiers corresponding to the current state of the active application in the card request **110**. The button application **212** may obtain this data by scraping a document representing the current state.

[0103] At **514**, the button application **212** determines whether the card **102** for the current state of the active application is available. The button application **212** may receive the card system response **120** that includes an availability indicator **122**. If the availability indicator **122** indicates that a card is available or if the card system response **120** includes the card data **124**, the button application **212** determines a sharing score of the available card, as shown at **516**. In some implementations, the card system response **120** includes the sharing score of the available score. In other implementations, the button application **212** determines the sharing score based on local sharing statistics and/or global sharing statistics. The button application **212** can obtain local-sharing statistics using a state identifier, an application identifier, a function identifier, and/or one or more entity identifiers corresponding current state of the target application/the available card. The state identifier, the application identifier, the function identifier, and/or the one or more entity identifiers can be transmitted in the card system response **120** or can be scraped from the current state by the button application **212**. The global sharing statistics may be received in the card system response **120**. The button application **212** can utilize the retrieved local sharing statistics and/or the received global sharing statistics to determine the sharing score of the available card. As discussed above, the button application **212** can calculate the sharing score using rules-based scoring, a machine-learned scoring model, or a scoring function.

[0104] At **518**, the button application **212** determines whether the sharing score exceeds a threshold value. For example, if the sharing scores are normalized between the value of 0 and 1, the threshold value may be set at 0.1 or 0.2, depending on the desired amount of sharing (lower thresholds may correlate to more sharing). By exceeding the threshold value, the sharing score corresponds to an available card that is more likely to be shared by the user. At **520**, the button application **212** displays the card button **202** in a first manner when the share button exceeds the threshold value. In this situation, the button application **212** displays the card button **202** in a more conspicuous manner (e.g., in a relatively larger format). At **520**, the button application **212** displays the card button **202** in a second manner if the sharing score does not exceed the threshold, but is greater than zero. In this situation, the button application **212** displays the card button **202** in a relatively less conspicuous manner (e.g., in a relatively smaller format).

[0105] If the user selects the card button **202**, the button application **212** generates the card **102** based on the card object. The button application **212** may prompt the user to select a target application from a group of target applications or may utilize a default target application. In response to determining a target application, the button application **212** inserts the card **102** in the target application. For example,

the generated card may be inserted in a social media application, a bookmarking application, and/or a messaging application. In some implementations, the button application **212** can access an API of the target application to insert the card **102** to the target application.

[0106] The methods **400**, **400** of FIGS. **4** and **5** are provided for example only. Variations of the methods **400**, **500** are within the scope of the disclosure.

[0107] Various implementations of the systems and techniques described here can be realized in digital electronic and/or optical circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations can include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device.

[0108] These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and can be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the terms “machine-readable medium” and “computer-readable medium” refer to any computer program product, non-transitory computer readable medium, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term “machine-readable signal” refers to any signal used to provide machine instructions and/or data to a programmable processor.

[0109] Implementations of the subject matter and the functional operations described in this specification can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Moreover, subject matter described in this specification can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter affecting a machine-readable propagated signal, or a combination of one or more of them. The terms “data processing apparatus,” “computing device” and “computing processor” encompass all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated

electrical, optical, or electromagnetic signal that is generated to encode information for transmission to suitable receiver apparatus.

[0110] A computer program (also known as an application, program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[0111] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

[0112] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio player, a Global Positioning System (GPS) receiver, to name just a few. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[0113] To provide for interaction with a user, one or more aspects of the disclosure can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube), LCD (liquid crystal display) monitor, or touch screen for displaying information to the user and optionally a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide interaction with a user as well; for example, feedback provided to the user can be any

form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's client device in response to requests received from the web browser.

[0114] One or more aspects of the disclosure can be implemented in a computing system that includes a backend component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a frontend component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such backend, middleware, or frontend components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), an inter-network (e.g., the Internet), and peer-to-peer networks (e.g., ad hoc peer-to-peer networks).

[0115] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other. In some implementations, a server transmits data (e.g., an HTML page) to a client device (e.g., for purposes of displaying data to and receiving user input from a user interacting with the client device). Data generated at the client device (e.g., a result of the user interaction) can be received from the client device at the server.

[0116] While this specification contains many specifics, these should not be construed as limitations on the scope of the disclosure or of what may be claimed, but rather as descriptions of features specific to particular implementations of the disclosure. Certain features that are described in this specification in the context of separate implementations can also be implemented in combination in a single implementation. Conversely, various features that are described in the context of a single implementation can also be implemented in multiple implementations separately or in any suitable sub-combination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a sub-combination or variation of a sub-combination.

[0117] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multi-tasking and parallel processing may be advantageous. Moreover, the separation of various system components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems can generally

be integrated together in a single software product or packaged into multiple software products.

[0118] A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the disclosure. Accordingly, other implementations are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results.

What is claimed is:

1. A method comprising:
 - transitioning, by a processing device of a user device, to a current state of a first application;
 - obtaining, by the processing device of the user device, state indication data corresponding to the current state of the first application;
 - transmitting, by the processing device, the state indication data to a card system;
 - receiving, by the processing device, a card system response from the card system indicating whether the card system can provide a card corresponding to the current state;
 - determining, by the processing device, a sharing score of the card corresponding to the current state, the sharing score being indicative of an estimated probability that a user of the user device will share the card corresponding to the current state;
 - when the sharing score corresponding to the current state is above a threshold, displaying a first selection element overlaying a graphical user interface of the first application; and
 - when an availability indicator indicates that the card system can provide the card and the sharing score is below the threshold and above a second threshold, displaying a second selection element overlaying the graphical user interface of the first application.
2. The method of claim 1, wherein determining the sharing score includes:
 - receiving global sharing statistics from a card server corresponding to the card; and
 - determining the sharing score based on the global sharing statistics.
3. The method of claim 2, wherein determining the sharing score based on the global sharing statistics includes feeding the global sharing statistics to a machine learned scoring model.
4. The method of claim 2, wherein determining the sharing score based on the global sharing statistics includes feeding the global sharing statistics to a machine learned scoring model.
5. The method of claim 2, wherein the global sharing statistics include global state statistics indicating a first amount of instances where a group of users viewed the current state and a second amount of instances where the users shared cards corresponding to the current state.
6. The method of claim 2, wherein the global sharing statistics include global application statistics indicating a first amount of instances where a group of users viewed any sharable state of the first application and a second amount of instances where the group of users shared cards corresponding to any of the sharable state.
7. The method of claim 2, wherein the global sharing statistics include global function statistics indicating a first amount of instances where a group of users viewed any

sharable state corresponding to a function performed at the current state across a set of known applications and a second amount of instances where the group of users shared cards corresponding to any of the sharable states.

8. The method of claim 2, wherein the global sharing statistics include global entity statistics indicating a first amount of instances where a group of users viewed any sharable state corresponding to an entity relating to the current state across a set of known applications and a second amount of instances where the group of users shared cards corresponding to any of the sharable states.

9. The method of claim 2, wherein determining the sharing score further includes:

- retrieving local sharing statistics corresponding to the card; and

- determining the sharing score based on the local sharing statistics and the global sharing statistics.

10. The method of claim 9, wherein the local sharing statistics include local state statistics indicating a first amount of instances where a user of the user device viewed the current state and a second amount of instances where the user shared the card corresponding to the current state.

11. The method of claim 9, wherein the local sharing statistics include local application statistics indicating a first amount of instances where a user of the user device viewed any sharable state of the first application and a second amount of instances where the user shared cards corresponding to any of the sharable state.

12. The method of claim 9, wherein the local sharing statistics include local function statistics indicating a first amount of instances where a user of the user device viewed any sharable state corresponding to a function performed at the current state across a set of known applications and a second amount of instances where the user shared cards corresponding to any of the sharable state.

13. The method of claim 9, wherein the global sharing statistics include global entity statistics indicating a first amount of instances where a user of the user device viewed any sharable state corresponding to an entity relating to the current state across a set of known applications and a second amount of instances where the user shared cards corresponding to any of the sharable state.

14. A method comprising:

- receiving, by a processing system of a card system, a card system request from a user device including state indication data indicating a state of an active application of the user device;

- identifying, by the processing system, a card record based on the state indication data, the card record corresponding to the state indicated by the state indication data and defining card data of a card corresponding to the state;

- retrieving, by the processing system, global sharing statistics based on the card record from a global sharing statistics data store, the global sharing statistics corresponding to a group of users;

- determining, by the processing system, a sharing score corresponding to the card based on the global sharing statistics, the sharing score being indicative of an estimated probability that a user of the user device will share the card corresponding to the state; and

- transmitting, by the processing system, the sharing score to the user device.

15. The method of claim 14, wherein determining the sharing score based on the global sharing statistics includes feeding the global sharing statistics to a machine learned scoring model.

16. The method of claim 14, wherein the global sharing statistics include global state statistics indicating a first amount of instances where a group of users viewed the state and a second amount of instances where the users shared cards corresponding to the state.

17. The method of claim 14, wherein the global sharing statistics include global application statistics indicating a first amount of instances where a group of users viewed any sharable state of the active application and a second amount of instances where the group of users shared cards corresponding to any of the sharable state.

18. The method of claim 14, wherein the global sharing statistics include global function statistics indicating a first amount of instances where a group of users viewed any sharable state corresponding to a function performed at a current state across a set of known applications and a second amount of instances where the group of users shared cards corresponding to any of the sharable states.

19. The method of claim 14, wherein the global sharing statistics include global entity statistics indicating a first amount of instances where a group of users viewed any sharable state corresponding to an entity relating to a current state across a set of known applications and a second amount of instances where the group of users shared cards corresponding to any of the sharable states.

* * * * *