



(11) **EP 4 343 569 A2**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
27.03.2024 Bulletin 2024/13

(51) International Patent Classification (IPC):
G06F 16/215^(2019.01)

(21) Application number: **23208140.6**

(52) Cooperative Patent Classification (CPC):
G06F 9/451; G06F 9/543; G06F 16/215; G06F 8/36

(22) Date of filing: **25.08.2016**

(84) Designated Contracting States:
**AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR**

- **Hsu, Tiffany**
Denver, 80202 (US)
- **McCardel, Barry**
Denver, 80202 (US)
- **Schafer, Mark**
Denver, 80202 (US)
- **Bonventre, Andrew**
Denver, 80202 (US)

(30) Priority: **26.08.2015 US 201562210296 P**
12.08.2016 US 201615235558

(62) Document number(s) of the earlier application(s) in
accordance with Art. 76 EPC:
16185581.2 / 3 144 810

(74) Representative: **Dendorfer, Claus et al**
Dendorfer & Herrmann
Patentanwälte Partnerschaft mbB
Neuhauser Straße 47
80331 München (DE)

(71) Applicant: **Palantir Technologies Inc.**
Denver, CO 80202 (US)

(72) Inventors:

- **Falter, Kelsey**
Denver, 80202 (US)
- **Tao, Joy**
Denver, 80202 (US)
- **Stearns, Taylor**
Denver, 80202 (US)

Remarks:

- This application was filed on 07-11-2023 as a
divisional application to the application mentioned
under INID code 62.
- Claims filed after the date of filing of the application
/ after the date of receipt of the divisional application
(Rule 68(4) EPC).

(54) **SYSTEM FOR DATA AGGREGATION AND ANALYSIS OF DATA FROM A PLURALITY OF DATA SOURCES**

(57) A method, storage medium and computing system involve operations of: receiving artifact data from a source software application via an API; formatting the artifact data to generate an interactive graphical unit; generating user interface data for rendering an interactive user interface on a computing device, the interactive user interface including the interactive graphical unit; determining if a user has write permission with respect to

at least the interactive graphical unit; if it is determined that the user has write permission with respect to at least the interactive graphical unit, and in response to an indication from the user that the interactive graphical unit or a group of interactive graphical units comprising the interactive graphical unit is to be edited, performing at least one editing, adding, changing, formatting or removing action.

EP 4 343 569 A2

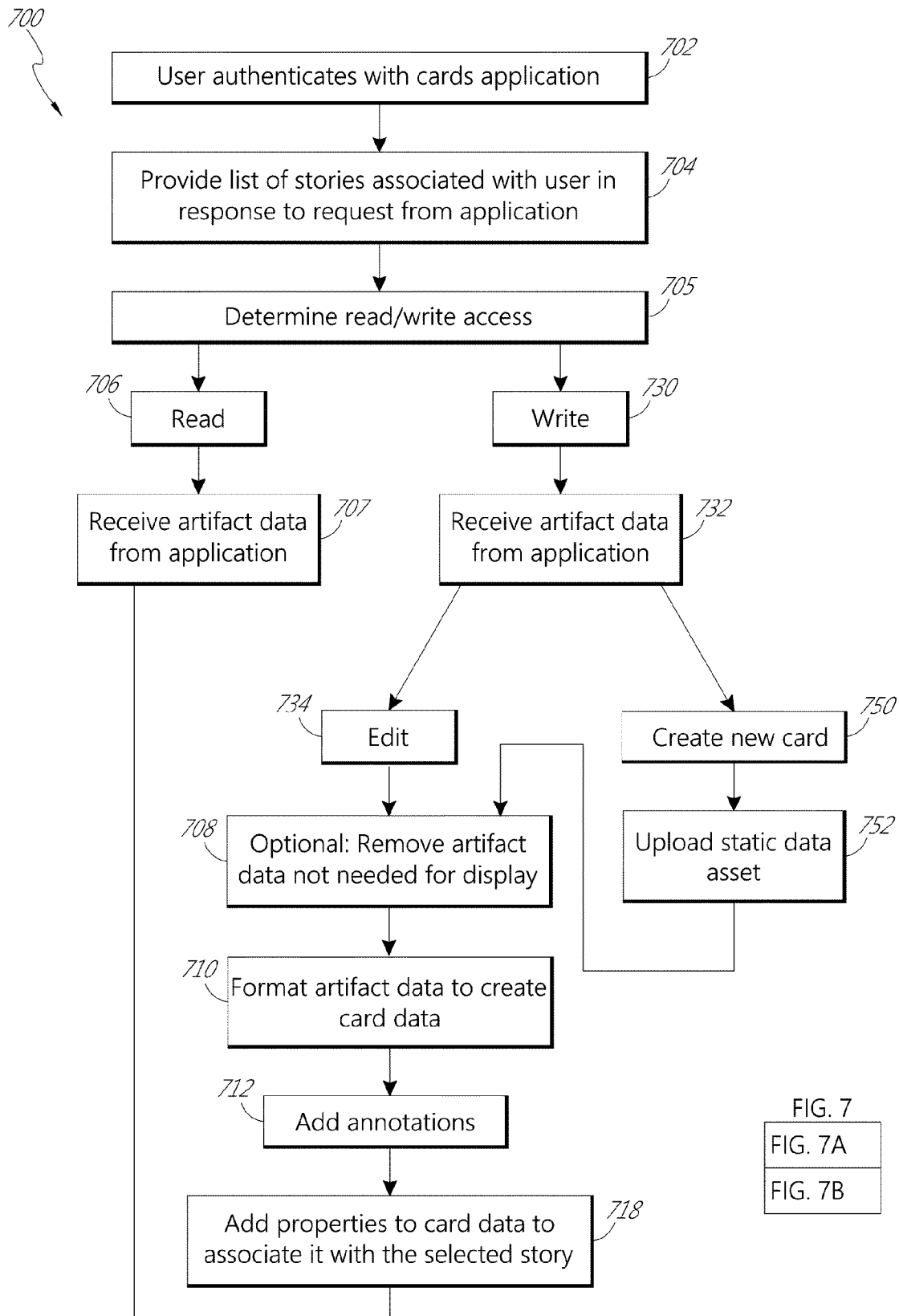


FIG. 7
FIG. 7A
FIG. 7B

FIG. 7A

Description

TECHNICAL FIELD

[0001] The present disclosure relates to systems and techniques for aggregating, modifying, and displaying data in an interactive user interface.

BACKGROUND

[0002] A variety of analytical platforms exist for creating and/or editing representations of data. Such representations of data may be referred to as "artifacts." For example, an artifact may be a histogram, line chart (e.g., time series), heatmap, bar chart, etc. Such artifacts may be useful for providing graphical representations of data such as statistics, readouts from a computer, etc.

[0003] In some cases, a user may wish to output artifacts from an analytical platform. For example, the user may wish to present analytical findings to consumers, colleagues, etc. The user may also wish to present the artifact together with other data, including other artifacts. However, it may be difficult to output artifacts in a dynamic, interactive format that can be made available for consumption on a variety of platforms.

SUMMARY

[0004] The systems, methods, and devices described herein each have several aspects, no single one of which is solely responsible for its desirable attributes. Without limiting the scope of this disclosure, several non-limiting features will now be discussed briefly.

[0005] Disclosed herein are various systems and methods for receiving and displaying data in interactive user interfaces. In some embodiments, the interactive user interfaces may display data sets from a plurality of external applications and/or data sources. The data corresponding to the data sets may include graphs (as described below), interactive objects, images, documents, and/or other types of data. A collection of data sets to be displayed on the interactive user interface may be referred to as "card data." External applications may also be referred to herein as "source applications."

[0006] In some embodiments, card data may comprise one or more data items to be displayed from the plurality of external applications and/or data sources. A displayed unit based on card data may be referred to as a "card." In some embodiments, card data may comprise display parameters specifying the data to be displayed by a card, and also how the data is to be displayed. Card data may be comprised of one or more "card data items" that may specify individual components of the card data. For example, for a card displaying a bar graph, one card data item may define a color scheme for the card, another may define a title for the card, etc. Card data may also include information about a card, such as tracked modifications to a card, date of creation, etc.

[0007] Card data may be retrieved from external applications. In some embodiments, retrieved card data may be displayed as a card in a format based upon that of their native external application. That is, the displayed card may look like the artifact which the card data was derived from. However, in an embodiment, the derived card data may not include all of the data related to the artifact used by the native external application. For example, card data may include only data necessary for displaying a card and may not include data that is only necessary for manipulating or changing card data. That is, the card may look like the artifact, but may not be manipulated or changed in the way the artifact could be at the external application. The card may include a link which allows users to access the external application to make any desired modifications or changes. In alternative embodiments, data related to manipulating or changing the data may be included in the card data. For example, the card data may be displayed in an interactive format, allowing the user to change or manipulate the data in a manner that would be possible in the data's native application. For example, the card may correspond to an artifact, wherein data for the artifact comprises one or more time-series graphs retrieved from a graph application. The user may be able to perform manipulations on the displayed graphs using the interactive user interface that correspond to manipulations that the user would have been able to perform using the graph application (e.g., zooming or scrolling through a graph, setting markers on a graph, associating events with data points in a graph, and/or the like).

[0008] The systems and methods described herein may provide several benefits. For example, the systems and methods described herein may allow the user to more easily gather and aggregate data from various different sources while maintaining an ability to access the various sources. By allowing the user to specify, view, and interact with different types of data from different sources in a single interface, the user may be able to better gather information, identify trends or correlations, and form a more complete picture of a topic or entity of interest. For example, a user may specify for display one or more graphs containing sensor data associated with an entity of interest during a time period of interest and associate it with a display of a similar graph from a different entity, thus providing an improved ability to compare the two. In addition, the user may be able to easily export and/or share such data and access the sources of the data to modify and/or build on the analytical findings. As such, the user may be able to more quickly and accurately access, navigate, assess, and digest the data than previous systems, potentially providing reduced mental workloads, improved decision-making, reduced work stress, and/or the like, for the user.

[0009] It has been noted that design of computer user interfaces "that are useable and easily learned by humans is a non-trivial problem for software developers." (Dillon, A. (2003) User Interface Design. MacMillan En-

cyclopedia of Cognitive Science, Vol. 4, London: Mac-Millan, 453-458.) The present disclosure describes various embodiments of interactive and dynamic user interfaces that are the result of significant development. This non-trivial development has resulted in the user interfaces described herein which may provide significant cognitive and ergonomic efficiencies and advantages over previous systems. The interactive and dynamic user interfaces include improved human-computer interactions that may provide reduced mental workloads, improved decision-making, reduced work stress, and/or the like, for a user. For example, user interaction with the interactive user interface via the inputs described herein may provide an optimized display of, and interaction with, graph data, image data, and/or other data, and may enable a user to more quickly and accurately access, navigate, assess, and digest the data than previous systems.

[0010] Further, the interactive and dynamic user interfaces described herein are enabled by innovations in efficient interactions between the user interfaces and underlying systems and components. For example, disclosed herein are improved methods of receiving user inputs (including methods of interacting with, and selecting, images, graphs, and other types of data), translation and delivery of those inputs to various system components, automatic and dynamic execution of complex processes in response to the input delivery, automatic interaction among various components and processes of the system, and automatic and dynamic updating of the user interfaces (to, for example, display the relevant data from various different applications and/or data sources). The interactions and presentation of data via the interactive user interfaces described herein may accordingly provide cognitive and ergonomic efficiencies and advantages over previous systems.

[0011] Various embodiments of the present disclosure provide improvements to various technologies and technological fields. For example, existing data aggregation and analysis technology is limited in various ways (e.g., limited in the types of applications or data sources the data may be drawn from, loss of data interactivity, etc.), and various embodiments of the disclosure provide significant improvements over such technology. Additionally, various embodiments of the present disclosure are inextricably tied to computer technology. In particular, various embodiments rely on detection of user inputs via graphical user interfaces, aggregation of data from different applications and data sources, and automatic processing, formatting, and display of the aggregated data via interactive graphical user interfaces. Such features and others (e.g., automatically determining an application or data source an inputted link is directed to, accessing the application or data source to retrieve and display the requested data, implementing interactivity of displayed data reflective of how the data would be displayed in its native application) are intimately tied to, and enabled by, computer technology, and would not exist except for computer technology. For example, the interactions with

displayed data described below in reference to various embodiments cannot reasonably be performed by humans alone, without the computer technology upon which they are implemented. Further, the implementation of the various embodiments of the present disclosure via computer technology enables many of the advantages described herein, including more efficient interaction with, and presentation of, various types of electronic data.

[0012] Additional embodiments of the disclosure are described below in reference to the appended claims, which may serve as an additional summary of the disclosure.

[0013] In various embodiments, computer-implemented methods are disclosed in which, under control of one or more hardware computing devices configured with specific computer executable instructions, one or more aspects of the above-described embodiments (including one or more aspects of the appended claims) are implemented and/or performed.

[0014] In various embodiments, transitory or non-transitory computer-readable storage mediums storing software instructions are disclosed, wherein, in response to execution by a computing system having one or more hardware processors, the software instructions configure the computing system to perform operations comprising one or more aspects of the above-described embodiments (including one or more aspects of the appended claims).

[0015] Further, as described herein, various embodiments of the system may be configured and/or designed to generate user interface data useable for rendering the various interactive user interfaces described. The user interface data may be used by the system, and/or another computer system, device, and/or software program (for example, a browser program), to render the interactive user interfaces. The interactive user interfaces may be displayed on, for example, electronic displays (including, for example, touch-enabled displays).

BRIEF DESCRIPTION OF THE DRAWINGS

[0016]

Figure 1 illustrates a system diagram illustrating how a cards application communicates with other applications and/or data sources to retrieve data.

Figure 2 illustrates an example interactive graphical user interface of the cards application, including multiple user-selectable "stories."

Figures 3A-3C illustrate example interactive graphical user interfaces of the cards application, including a display of "cards" within a selected "story."

Figure 4 illustrates an example interactive graphical user interface of an external application 108 that may communicate with the cards application 102 of Figure 1.

Figure 5 illustrates an example interactive graphical user interface of the cards application for applying

permissions.

Figures 6A and 6B show a flowchart depicting an illustrative operation of an external application in which the external application interacts with a user and the cards application.

Figures 7A and 7B show a flowchart depicting an illustrative operation of the cards application in which the cards application interacts with a user and the external application of Figures 6A and 6B.

Figure 8 is a flowchart depicting an illustrative operation of the cards application for determining whether a user has access permissions.

Figure 9 illustrates a computer system with which certain methods discussed herein may be implemented, according to various embodiments.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

Overview

[0017] As described above, it may be difficult to output analytical representations of data (e.g., histograms, line charts, heatmaps, bar charts, etc.) in a dynamic, interactive format that can be made available for consumption on a variety of platforms. In other words, the ability to share analysis may be inhibited by a lack of a canonical, self-contained unit of insight capable of effectively sharing analysis results. For example, analysis results may typically be presented using PowerPoint/Keynote presentations, e-mails, and/or hardcopy documents passed around at meetings. However, such presentation methods may be ineffective for busy, on-the-go consumers and/or may not provide abilities for interaction with the presentation materials.

[0018] Accordingly, disclosed herein are various systems and methods for providing a platform-agnostic, self-contained unit of content for sharing analysis results and insights in an interactive user interface. Such systems and methods may provide a flexible Application Programming Interface (API) that allows users to develop a format for mapping analytical findings to an output that is effective for consumption as well as for providing attribution and/or other information for the analytical findings. For example, a computing device (e.g., the computing system 900 of Figure 9 described below) may be configured to retrieve data from one or more external applications and display the retrieved data in one or more interactive graphical unit, which may be referred to herein as "cards." The set of data associated with a card, and upon which a card is based (e.g., the retrieved data from the one or more applications) may be referred to herein as a set of "card data" and/or "card data items." A set of card data items may be composed of various data items (including various properties, and/or metadata) which define the features of the card. For example, such card data may include a unique identifier (ID), Access Control List (ACL) or other permissions, title, author, time/date of creation

and/or most recent edit, time/date of data update, tags, creation path, source of data, type of card, link to other card or application, card details, footnote, etc. The card may be interactive in that a user may manipulate individual cards data items associated with the card, as well as be able to access the original source (e.g., the external application) of the data displayed in the card. A card may be created by a card software application, hereinafter referred to as a "cards application," or "target application."

[0019] Figure 1 illustrates a system diagram illustrating how a cards application communicates with other applications and/or data sources to retrieve data, in accordance with some embodiments. A user may access a cards application 102 and a variety of other (e.g., external) applications 104, 106, 108 via a user device 110. The user device 110 may be any computing device capable of accessing software applications, such as a smartphone, tablet, laptop computer, etc.

[0020] One or more applications may be run on various computing devices, such as user device 110. For example, cards application 102 and external applications 104, 106, and 108 may be run on the user device 110. Applications may also be run on different devices. For example, cards application 102 and each of external applications 104, 106, and 108 may each be run on separate computing devices. Figure 9 illustrates an exemplary computing device (e.g., user device 110) which may be used to run one or more applications. While Figure 1 illustrates three external applications 104, 106, and 108, this is only exemplary and more or fewer applications may be used.

[0021] Cards application 102 and/or external applications 104, 106, and 108 may generate user interface data that may be transmitted to the user device 110. The user interface data may be transmitted in a form suitable for display by the user device 110 and/or may be suitable for being rendered (e.g., via a web browser) at the user device 110. The user device 110 may communicate with the cards application 102 and/or external applications 104, 106, and 108 over wired or wireless connections and/or over the internet. The various devices in Figure 1 may be locally or remotely connected.

[0022] Cards application 102 may communicate with external applications 104, 106, and 108 (e.g., analytical platforms) via an Application Programming Interface (API) 112 (referred to herein as "cards API 112"). The cards API 112 may be a flexible API. For example, a user may access an external application 108 to create a graphical representation of data (e.g., an artifact), such as a bar graph). To create the bar graph, the external application 108 may access data stored in a database 114. The database 114 may contain data such as graphical representations (e.g., tools for creating a bar graph) and data to be represented (e.g., in the bar graph). After completing the bar graph, the user may wish to output the graph to a platform to make it available to consumers, colleagues, etc. The user may select a link that instructs

the external application 108 to perform an API call to the cards API 112. In one example, the external application 108 may access the API 112 via a plug-in or extension. The cards API 112 may receive data representative of the graph and may perform a variety of functions to make the received data suitable for use by the cards application 102 and/or add features to the received data for improved consumption. The cards application 102 may receive the data representative of the graph and may create card data representative of the modified bar graph and any additional features added by the cards API. The cards application 102 may store the card data in a database 116 and may display the card data as a card. As further illustrated in Figure 1, the cards API 112 may communicate direct with the database 116 to, for example, service API calls via the cards API 112.

[0023] Figure 1 further illustrates a cards job server 122. The cards job server 122 may communicate with the database 116 and the cards API 112 to perform, for example, various backend functions. For example, as described here, in some implementations cards (e.g., data underlying a card) may be automatically updated (e.g., periodically, on a schedule, upon a user's request, or otherwise). In these implementations, the cards job server 122 may run a job that, for example, may access and update data via API calls of the cards API 112. For example, a user may specify an update job as follows: "I would like the data in this card to be updated on X frequency." In this example, the jobs server 122 will execute a job with the specified frequency to update the data related to the specified card, for example, accessing data from a data-provider application (e.g., any external data source; external applications 104, 106, 108; databases 114, 118, 120; etc.) if the application/data is not available on the user's local machine.

[0024] As used herein, the term "database" may include, without limitation, other data structures such as comma separated values (CSV) files, eXtensible markup language (XML) files, TeXT (TXT) files, flat files, spreadsheet files, and/or any other widely used or proprietary format for data storage. Furthermore, a database may be any data store or data structure (and/or combination of multiple data stores, data structures, and/or databases) for storing data. For example, suitable databases may include a relational database management system (RDBMS), a structured query language (SQL) or non SQL (NoSQL) database, etc.

[0025] Communication between the cards application 102 and the external applications 104, 106, 108 may be bi-directional. For example, a user viewing the modified bar graph card via the cards application 102 may wish to access the original bar graph. The user may select an attribution derivation link embedded in the card that performs a call to the cards API 112. The cards API 112 may receive the card data representative of the bar graph from the cards application 102 and route the user device 110 to the external application 108 where the graph was created. The external application 108 may access data cor-

responding to the original graph from the database 114 and present it to the user.

[0026] Card data and/or card data items may be associated with permissions. For example, a card data item may provide an Access Control List (ACL) that defines which users may access the card data and/or view the card associated with the card data. The defined users may be on an individual basis or may be based on a class (e.g., employees of a company). Moreover, individual card data items may be associated with permissions. For example, a card data item may define a card's title. The author of the card may associate a permission with the title to allow only certain users accessing the card to be able to change the title.

[0027] The cards application 102 may organize cards into different analysis narratives, referred to herein as "stories" or groups of interactive graphical units. Each card in a story may provide an atomic unit of content that may be combined with other related cards in a larger presentation. For example, a story directed to consumption of a certain product may include multiple cards, with each card illustrating consumption of the product by a different demographic group and/or over a different time period.

[0028] Figure 2 illustrates an example interactive graphical user interface of the cards application, including multiple user-selectable stories, in accordance with some embodiments. As shown in Figure 2, each story 202 may have one or more identifiers, such as a title 204, an owner 206 (if any), and a date/time of a most recent modification 208. Each story 202 may also have a card number indicator 210 to indicate how many cards are contained in the story 202. A user may perform a search 212 of the stories using any story identifier. Stories 202 may also be sorted by any identifier (e.g., owner 206) and/or other data item (e.g., number of cards). The interface 200 also includes a new story link 214 to allow a user to create a new story. Each story 202 in the interface may be selected to direct a user to an interface providing further detail regarding the selected story 202. A story 202 may have an associated ACL to control which users are able to access the story 202.

[0029] Figure 3A illustrates an example interactive graphical user interface of the cards application, including a display of cards within a selected story, in accordance with some embodiments. As shown in Figure 3A, cards (e.g., cards 302) may be grouped into sections (e.g., section 304) within a story. The interface 300 includes a story title 306 and section titles (e.g., section title 308). The story and section titles may be added and/or edited by the user so as to enable the user to organize the cards in any way desired. Each card (e.g., see one of cards 302) may have an identifying number 310 and/or a title/description 312. A user may perform a variety of functions for each card, including deleting (e.g., by selection of icon 314) and expanding (e.g., by selection of button 316) a card. In an embodiment, icon 314 and/or button 316 may be displayed with respect to a

card when the user hovers a mouse cursor over the card. A user may also move a card to a different section and/or position within a section, e.g., by selecting and dragging the card to the desired section. The user may, via the example interactive graphical user interface of Figure 3A, add or delete sections from the story, and/or otherwise organize the story as desired. After making any changes to a card (e.g., changing card data associated with one of cards 302) and/or section, the user may save changes made to the story by selection of button 318. Alternatively, changes to the story, sections within the story, and/or cards may be automatically saved. The user interface of Figure 3A also includes user-selectable elements (e.g., elements 320 and 322) by which the user may preview and/or share the story, and/or cards of the story. For example, a user may share the story and/or a card of the story with an external platform, such as Facebook, Twitter, etc., and/or may share a card with, and/or transfer a card to, a different story.

[0030] The user may also export (e.g., by selection of user-selectable element 323) the story and/or one or more cards of the story to a different platform and/or format for convenience in sharing the story with others. For example, a user may export the story to a format such as a presentation format (e.g., PowerPoint format). In such an export, the cards of the story may each be automatically converted into slides of a presentation format, and compiled into a single presentation file. In other examples, the story may be converted into any other format suitable for a different platform. Exporting a story may also cause various card data items associated with a card and/or story to be embedded in the exported file and/or format. For example, an exported card (e.g., a card exported to a slide of a presentation file, may include attribution information at the bottom of the slide to indicate that the card was created in the cards application 102, may provide an identity of the author of the card and/or story, and/or may provide other similar information associated with the card and/or story. The exported card may also include a link back to the cards application 102 to allow the user to view the card associated with the exported card within the cards application 102.

[0031] Figure 3B illustrates the example interactive graphical user interface of Figure 3A with an expanded view of a card, in accordance with some embodiments. The card 370 illustrated in Figure 3B is merely one example card composed of multiple example aspects, and other cards may have any number of similar or different aspects, including aspects different from those illustrated in Figure 3B. As shown in Figure 3B, the example card 370 may include a display of multiple aspects, each of which may be based on a card data item, including a bar chart 324, a key 326, statistical data 328, and a title/description 329. Additionally, the card 370 may further include a display of other card data items, such as a title 330, an attribution derivation link 332, and options to edit and remove (e.g., by selection of user-selectable elements 334 and 336, respectively) the bar chart 324. Fur-

ther properties may include an author 338 and card type 340. One or more of these aspects may be modified by a user. For example, a user may change the author 338 identified for the card 302. In some embodiments, various aspects may have associated permissions. For example, only users from a particular group may be allowed to modify the title/description 330 of the card 370, while user from a different (e.g., larger) group may be allowed to modify the bar chart 324. A user may also perform further actions, such as duplicating, copying, and moving (e.g., by selection of user-selectable elements 341, 342 and 344, respectively) the card 370.

[0032] As discussed, cards may be responsive to user interaction. For example, hovering or clicking on aspects of card 370 may cause more data to be shown. In an example of a line graph, such data may include values of selected data points. For pie or bar charts, such data may include breaking out the parts of a selected data point.

[0033] Additionally, a card may have an associated card ID. The ID for each card may be unique and may be associated with card data items of the card, including, for example, a date of creation, author, attribution history, ACL, artifact, and/or any other data items used in displaying the card. Card data items associated with a card, as described herein, may be stored in the database 116, and may be associated with the card ID of the card, which may also be stored in the database 116. Each component of the card ID may individually map to the card, such that a user can search card data items associated with a card ID to find a desired card. For example, a user searching cards having a particular date of creation may be provided a list of cards having an ID that includes the selected date of creation. In another example, a user may search for cards added to a particular story that were created in a particular time period selected by the user. In this way, users can effectively search stories, cards, and card data items using varying levels of specificity. As shown, such searching may apply not only to display data related to a card and/or story, but also the underlying metadata (e.g., data tracking modifications to a card).

[0034] A card ID may provide a reference point for a card. That is, by associating each card with a card ID, each card is itself an atomic unit independent of association with any particular story. Each card may be a member of multiple stories.

[0035] In an embodiment, the cards application 102 may track changes made to a card and/or card data items associated with the card. For example, if the title of the card is changed, the change may be recorded in the database 116 for the card ID associated with the card. In an embodiment, the database 116 may save multiple versions of a card. Users may also search for changes made to a card. For example, a user may search for any changes made to a card in a selected time period. For each change to a card, the user may have the option to make the update across all instances of the card ID associated with the card or generate a new card ID.

[0036] Although cards are illustrated in Figures 3A and 3B as having an expanded (e.g., Figure 3B) or contracted (e.g., Figure 3A) state viewed in a vertical sequence, cards may be presented in a variety of ways. For example, cards may be displayed in a grid having a one or more cards in one or more columns and/or rows. In the present example, cards in a grid format may be expanded and contracted. In another example, cards may be viewed in a horizontal sequence. Other methods of displaying cards are also contemplated.

[0037] In the example embodiment, the bar chart 324 is composed of five bars 346, each representing a sum of projected dollars 348 for a given year 350. The data represented in the bar chart 324 may be populated automatically. For example, the bar chart 324 may have been created by use of software that automatically populated the bar chart 324 with stored data. In another example, the bar chart 324 may be automatically updated in real-time from a source providing real-time data, such as a website. In addition to the bar chart 324, any other card data items associated with the card 370 may be updated (i.e., refreshed) automatically. Alternatively, the data represented in the bar chart 324 may be populated manually by users. For example, a user may adjust the values indicated in the statistical data 328 to adjust the values in the bar chart 324. The user may also edit (e.g., by selection of user-selectable element 334) the bar chart 324. In some embodiment, users may make adjustments directly to various card data items, including the bar chart 324.

[0038] In an embodiment, a user may not edit the bar chart 324 via the cards application 102, but may select the attribution derivation link 332 to edit the bar chart 324 using an external application. The attribution derivation link 332 may provide a link to an external application where the bar chart 324 was created. That is, by selecting the attribution derivation link 332, the user may be directed to an external application 108 (e.g., an analytical platform) and may view the analytical path for creation of the bar chart 324. For example, the link 332 may launch the external application 108 and direct the user to a view of the application 108 with pre-populated data from the database 114 related to creation of the bar chart 324, including data represented by the bar chart 324, format used for the bar chart 324, etc.

[0039] The attribution derivation link 332 is illustrated in Figure 3B having text providing a file name of the file used to create the bar chart 324. However, the attribution derivation link 332 may alternatively comprise text providing different information. For example, the link 332 may comprise text describing how the bar chart 324 was created, such as "created having # bars, narrowed to # bars," etc.

[0040] Figure 3C illustrates the example interactive graphical user interface of Figure 3B with an annotation 352, in accordance with some embodiments. The annotation 352 may comprise visual and/or textual cues that may be added to content to help explain, give context,

and/or provide commentary. Annotations 352 may have a variety of uses, including making portions of a card 302 intelligible or interesting, and flagging parts of a card 302 as a communication to other users. For example, an annotation 352 may comprise highlighting of a portion of a card 302 or a text box, as shown in Figure 3C.

[0041] An annotation 352 may have an "anchor" 354 that connects a cue 356 to the card 302. An anchor 354 may be a selection of data of the card 302 that the annotation 352 references. For example, in Figure 3C, the anchor is a bar 346 of the bar graph 324. Annotations 352, anchors 354, and cues 356 may each be searchable by users. In one embodiment, when a user selects an attribution derivation link 332 to access a different application 108, annotations 352 are not transferred between applications, but may be accessible whenever a card 302 is viewed via the cards application 102. Alternatively, annotations may be transferred between applications when the user selects an attribution link and/or creates a card. For example, annotations associated with an artifact created at an external application 108 may be transferred, via the cards API 112, as card data items to be associated with a card created from the artifact.

[0042] Users may also provide comments on a card or story. For example, when viewing a card, a user may be able to view comments left by previous viewers. Comments may be associated with a card ID and with the commenter's user ID.

[0043] Any of the described data items associated with stories (e.g., owner, number of cards, etc.) and cards (e.g., title, annotations, etc.) may be searchable by users. In this way, the cards application 102 and associated databases 116 may provide a convenient platform for accessing various analytical information, including information from a variety of applications. Various databases may also be suitable for indexing to provide improved search efficiency. This may be possible because related (and/or all) card data items may be stored in a database (e.g., database 116) associated with the cards application 102, rather than in disparate databases associated with various external applications. Thus, related (and/or all) card data items may be efficiently indexed and searched by the application.

[0044] Figure 4 illustrates an example interactive graphical user interface of an example external application 108 that may communicate with the cards application 102 of Figure 1, in accordance with some embodiments. As shown in Figure 4, an application 400 may be suitable for creating and/or editing an artifact, such as a bar chart 402. The application 400 may be configured to communicate with the cards application 102 via the cards API 112. In an embodiment, the application 400 includes a "cardify" icon 404 which may facilitate communication with the cards application 102. A user may select the cardify icon 404 to open a list of stories 406 and/or an option to create a new story 408. The user may select a story 406 to view a card list 410 showing options to select a card 412 within the selected story 406 or create a new

card 414 By selecting a story 406, 408 and/or a card 412, 414, the application 400 may implement the cards API 112 and transmit data indicative of the bar chart 402 created in the application 400 to the cards API 112. The cards API 112 may modify the bar chart 402 data to create a card for use by the cards application 102 and route the user to the cards application 102. The card created based on the bar chart 402 data may be added to the story 406, 408 selected by the user. Additionally, if a particular card 412 was selected, the card data items associated with the selected card 412 may be modified based on the created card 302. For example, a previous artifact at the selected card 412 may be replaced by the bar chart 402.

[0045] In an embodiment, the cards API 112 and/or cards application 102 may apply logic to data received from an external application 108. For example, a story may have settings that are applied to each card in the story. In the present example, a story may have an associated key that defines a coloring scheme for cards in the story. That is, each card may have a matching key that coordinates the color scheme of artifacts added to cards in the story. Logic may also be applied based on a type of card. For example, logic may be applied such that negative values of bar charts 324 are represented in red. Thus, any received data indicative of a bar chart 324 artifact may be adjusted to illustrate negative values in red. Such logic may be individualized for specific users, external applications 108, artifact type, etc.

[0046] Figure 5 illustrates an example interactive graphical user interface of the cards application 102 for applying permissions, in accordance with some embodiments. A user may apply permissions (e.g., ACLs) to stories, cards, and/or any other individual or group of card data items of a card. For each ACL, the user may select an access level 502 for all users. For example, an ACL may allow all users to access a given card or restrict access to all users. Individual users may have access 504 to a given card, and other users may be added to an access list (e.g., by selection of user-selectable element 506). The level of access for each individual user may also be adjusted. For example, a user may only be allowed to view a card or may also be allowed to edit the card (e.g., by selection of user-selectable elements 508 and/or 510, respectively). In an embodiment, providing access to a user for a given story may allow the user to access all of the cards included in the story. Similarly, providing access for a given card may allow the user to access all of the card data items associated with the card. However, access for individual cards and/or card data items associated with a card (to the exclusion of others) may also be provided.

[0047] Figures 6A and 6B (herein collectively referred to as Figure 6) show a flowchart depicting an illustrative operation of an external application in which the external application interacts with a user and the cards application 102. In various implementations, the described blocks of the flowchart of Figure 6 may be performed in an order other than that specifically disclosed, or multiple blocks

may be combined in a single block. The example blocks may be performed in serial, in parallel, or in some other manner. Further, blocks may be added to or removed from the disclosed example embodiments. In an implementation, blocks 622-629 may be performed in an order different from that shown in, and described in reference to, Figure 6.

[0048] Additionally, as described above, and also below in reference to Figure 7, each data item and property of cards/stories may individually be associated with access controls. Accordingly, while not explicitly depicted in Figure 6, certain functions may depend on a user having access privileges to perform the action. For example, the user would only be able to select a story to add card data to if they have write access with respect to the card and the associated properties.

[0049] At block 602, the process 600 may involve implementing the cards API 112. For example, a user may select a link 404 at an external application 108 that implements the cards API 112.

[0050] The cards API 112 may include various standardized function calls that may be implemented in various applications including the external application 108. The external application 108 may call the standardized function calls of the cards API 112 to transfer data to the cards application 102 and other external applications (e.g., 104, 106) may perform the same actions by calling the standardized function calls. In this way, a variety of applications may communicate with the cards application 102 (e.g., send data to the cards application 102) via the cards API 112.

[0051] At block 604, the process 600 may involve receiving a selection of an artifact to add as a card. For example, the external application 108 may transmit data indicative of a selected artifact. The selection may be received at the cards API 112.

[0052] At optional block 606, a determination is made as to whether the identity of the user is known. For example, if a user has previously implemented the cards API 112, the identity of the user may be known. In an embodiment, if the user's identity is known by the external application, information regarding the user's identity may be provided, via the cards API 112, to the cards application 102. If the identity of the user is not known, then at block 608, the identity of the user may be determined by the external application. For example, the user may log into the external application and/or otherwise provide identifying information to the external application, which may be provided to the cards application 102 via the cards API 112.

[0053] At optional block 610, a determination is made as to whether the user is authenticated with the cards application 102. Such a determination may be made based on the identifying information described above. Authentication of the user is useful, as described below, to allow the user to easily add to, and interact with, existing cards and/or stories associated with the user in the cards application 102. For example, if there are permis-

sions associated with the cards application 102 and/or a particular card or story, user authentication may be required to allow the user to add a card to an existing story. However, in an embodiment, user authentication may not be required. If the user is not authenticated with the cards application 102, then at block 612, the user may be allowed to authenticate with the cards application 102. For example, the user may register an account with the cards application 102. Alternatively, if the user is not authenticated with the cards application 102, then at block 614 the user may be allowed to add a card to a story (e.g., a new or existing story). That is, the user may be able to create a new story that does not have associated permissions.

[0054] At block 616, the process 600 may involve performing a call to the cards API 112 to access the cards application 102 and retrieve stories associated with the user. For example, the cards application 102 may determine which stories the user has permission to access or that the user has previously worked on.

[0055] At block 618, the process 600 may involve allowing the user to select a story (e.g., a story which the user has write access to) to which the card will be added, or to create a new story to which the card may be added. For example, the selected story may be one of the stories retrieved by the cards application 102. The selected story may alternatively be a new story. At block 618, the user, for example, may select a story he or she has write access to, or may create a new story.

[0056] At block 619, the user may select to create a new card, which may be added to an existing story or a new story.

[0057] At block 620, the process 600 may involve determining a card and/or artifact type. For example, if the artifact is a bar chart, the external application 108 may determine that it is a bar chart type. The external application 108 may then transmit information indicative of the bar chart type to the cards API 112. The external application 108 may also transmit data items associated with bar charts, such as a key for labeling the bars in the bar chart. Further, in an implementation, the user may select the card type (e.g., chart card, image card, etc.)

[0058] Alternatively, at block 621, the user may select to add an artifact to an existing card in a story. In an implementation, the user may select the artifact/card type, or the system may determine the artifact/card type by virtue of the type of data of the artifact or the type of card the artifact is being added to.

[0059] Thus, for example, the user may either select an existing card to add to (e.g., the card being added could be the first data artifact added, or could be accretive to an existing data artifact within an existing card) or create a new card, as described above.

[0060] At block 622, the process 600 may optionally involve determining a set of artifact data needed for displaying the card 302. For example, not all of the data indicative of the artifact transmitted to the cards API 112 may be necessary for displaying data representative of

the artifact in the cards application 102. For instance, the external application 108 may use such data for editing and/or other capabilities involving the artifact that are not used by the cards application 102. As such, the cards application 108 may trim excess data from the artifact for improved efficiency at the cards application 102. In an implementation, the user may provide the data needed to create the card, for example, by uploading a static tabular data file, an image, or the like (e.g., where the data input is the user's local machine and the derivation link (as described below in reference to Figure 7) is a file, stored in the cards database).

[0061] At block 624, the process 600 may involve performing a cards API 112 call to transfer object data to the cards application 102. For example, the external application 108 may perform the call to the cards API 112.

[0062] At block 626, the process 600 may optionally involve automatically transferring the user (i.e., user device 110) to the cards application 102. For example, if the user was accessing the external application 108, the user may be transferred to the cards application 102 to view the card.

[0063] At block 628, the process 600 may involve receiving an acknowledgement that the card was added. For example, the external application 108 receives an acknowledgement from the cards application 102, which acknowledgement may be provided to the user.

[0064] At block 629, the process 600 may involve refreshing the data of cards from data sources associated with the cards. As described herein, cards are associated with data and a source for the data. Cards may further be associated with a frequency that the data in the card should be updated or refreshed. Thus, in block 629, the system may make a request for updated data (e.g., via an API call of the cards API 112, or by a request to a local data source), may receive the updated data, and may update the card. As described above in reference to Figure 1, such refreshing may be automatic or user initiated (e.g., the user may indicate the data is to be updated immediately via a user interface, such as a web interface, by selecting a "refresh" button). In an implementation, the system or the user may indicate a frequency or a schedule for updating the data (e.g., the user may input a frequency or schedule via a user interface, such as a web interface). Such frequency of updating may be, for example, never, daily, weekly, or some other frequency.

[0065] At block 630, the process 600 ends.

[0066] Figures 7A and 7B (herein collectively referred to as Figure 7) show a flowchart depicting an illustrative operation of the cards application 102 in which the cards application 102 interacts with a user and the external application of Figure 6, in accordance with some embodiments. In various implementations, the described blocks of the flowchart of Figure 7 may be performed in an order other than that specifically disclosed, or multiple blocks may be combined in a single block. The example blocks may be performed in serial, in parallel, or in some other

manner. Further, blocks may be added to or removed from the disclosed example embodiments.

[0067] Additionally, in various implementations, access controls (e.g., ACLs) may be associated with every item of data of a card/story, every property of a card/story. Accordingly, a user's access to card/story data may be finely defined and controlled. For example, a user may have read access to a story and simultaneously have partial write access to certain cards of the story. In another example, a user could have read/write control over a title of a card, but not design and visualization of the card, as each of these items comprises a different property of the card. In another example, a card ID can have a different access level than properties of the card, or they may all have the same access levels. Accordingly, ACLs may be defined for every item of data, every property every property type, etc., and for each user, user group, type of user, etc. Access controls, as described below in reference to Figure 7 and elsewhere herein, may be implemented in association with API calls that grab the card/artifact data, such that user actions (and data viewable by users) are restricted in accordance with their permission levels.

[0068] At block 702, the process 700 may involve the user authenticating with the cards application 302. For example, the cards application 102 may store user information in a database 116. The user information may be indicative of stories and/or cards associated with the user, including user permissions for accessing stories and/or cards.

[0069] At block 704, the process 700 may involve providing a list of stories associated with the user in response to a request from an external application 108. For example, the cards application may provide any stories associated with the user based on information stored in the database 116. The stories may be provided in response to the request from the external application 108 at block 616 of Figure 6.

[0070] At block 705, the process 700 may involve determine assess permissions of the user. For example, the system may determine that the user has one or more of read or write permission with respect to a card, story, etc.

[0071] If the user has read access, the process 700 may continue down the path indicated by block 706. As described above, read access may apply at the property level, the card level, the story level, etc. Accordingly, read access may mean that the user can only read the story, and that they cannot perform any other action beyond interacting with the read only graphical interface of the card. However, the user may still be able to comment on a story/card. Other combinations of access privileges may also be implemented.

[0072] At block 707, the process 700 may involve receiving artifact and/or card data from the external application 108. For example, the cards application 102 and/or the cards API 112 may receive data indicative of the artifact and/or card. The received data may be data trans-

ferred by the external application 108 at block 624 of Figure 6.

[0073] At block 714, the process 700 may involve accessing card data associated with the story. Further, at block 714, the system may display the card data to the user, as the user has read access permissions for the data.

[0074] At block 716, the process 700 may involve adding comments to the card. For example, users may add comments to a card.

[0075] At block 717, the process 700 may involve downloading artifact/card data.

[0076] The process 700 ends at block 720.

[0077] As mentioned above, in various embodiments, one or more of blocks 707, 714, 716, 717, or 720 may be optional, and/or these blocks may be performed in an order different from that shown in, and described in reference to, Figure 7 above.

[0078] Returning to block 705, if the user has write access, the process 700 may continue down the path indicated by block 730. As described above, write access may apply at the property level, the card level, the story level, etc. As user herein, the term "write" access generally includes read access also. Accordingly, write access may mean that the user can read the story, edit the story (e.g., reorder cards, add cards, etc.), etc. Other combinations of access privileges may also be implemented.

[0079] At block 732, as described above in reference to block 707, the process 700 may involve receiving artifact and/or card data from the external application 108. For example, the cards application 102 and/or the cards API 112 may receive data indicative of the artifact and/or card. The received data may be data transferred by the external application 108 at block 624 of Figure 6.

[0080] If the user indicates that a card/story is to be edited (e.g., by adding or changing data and artifacts, adding or changing cards, rearranging cards, editing the content of cards, etc.), the process 700 proceeds to block 734.

[0081] At block 708, the process 700 may optionally involve removing artifact data not needed for display. As discussed above, the external application 108 may provide data which the external application 108 uses in creating the artifact but may not be necessary for displaying the artifact in a card form. The cards application 102 and/or cards API 112 may therefore trim the excessive data for improved efficiency.

[0082] At block 710, the process 700 may involve formatting the received artifact data to create card 302 data. For example, the cards application 102 may require a slightly different format than that used by the external application 108.

[0083] At block 712, the process 700 may involve adding annotations to the card. For example, a user and/or the cards application 102 may add annotations to the card. Annotations may be added to a card 302 to communicate information to users about the card in general and/or about specific parts of the card. Annotations may

also be received from the external application 108. For example, the external application 108 may transmit annotations associated with the artifact used in creating the card. Such annotations may be added to the card.

[0084] At block 718, the process 700 may involve adding properties to the card data to associate the card with the selected story. For example, the user may select a story to associate the card with, and the card may be embedded with a card data item which creates a link between the card and the selected story.

[0085] At block 714, as described above, the process 700 may involve accessing card data associated with the story. The cards application 102 may associate certain data with each card of a given story. For example, each card in a story may include as a property the title of the story. Further, at block 714, the system may display the card data to the user.

[0086] At block 716, as described above, the process 700 may involve adding comments to the card. For example, users may add comments to a card.

[0087] At block 717, as described above, the process 700 may involve downloading artifact/card data.

[0088] The process 700 ends at block 720.

[0089] As mentioned above, in various embodiments, one or more of blocks 708, 710, 712, 718, 714, 716, 717, or 720 may be optional, and/or these blocks may be performed in an order different from that shown in, and described in reference to, Figure 7 above.

[0090] Returning to block 732, if the user indicates that a new card is to be added, the process 700 proceeds to block 752.

[0091] At block 752, the user may optionally upload a static data asset (e.g., locally provide a data file, such as a CSV file, image file, etc.) as described above.

[0092] The process 700 then proceeds as generally described above in reference to blocks 708, 710, 712, 718, 714, 716, 717, or 720 following the edit block 734.

[0093] Accordingly, the process 700 illustrates that certain blocks may be executed depending on the access permissions and indicated actions of the user. Further, in various implementations, various functionality of the blocks may vary depending on the access permission level of the user.

[0094] Figure 8 is a flowchart depicting an illustrative operation of the cards application 102 for determining whether a user has access permissions, in accordance with some embodiments. At block 802, the process 800 may involve creating or loading card data associated with a card 302. For example, a card may be created and/or edited at the cards application 102.

[0095] At block 804, the process 800 may involve selecting an attribution derivation link 332 for the card 302. For example, the attribution derivation link 332 may be configured to link the cards application 102 to an external application 108. The attribution derivation link 332 may transfer the user device 110 to the external application 108 to view the artifact used in creating the card 302.

[0096] At block 806, a determination is made as to

whether the user has access permissions for the attribution derivation link 332. For example, only one or more users of a group of users (e.g., the user who created the card) may be allowed to access the artifact used in creating the card 302. Other permissions may also be associated with other card data items associated with a card and/or of the card itself. If the user does not have access permissions, then at block 808, the external application 108 may not be launched and/or the user may not be transferred to the external application 108. If the user does have access permissions, then at block 810, the process 800 may involve launching the external application 108 to allow the user to access the artifact used in creating the card 302. A similar determination may be made for other card data items. For example, if a user attempts to view a particular card data item, a determination may be made as to whether the user has access permissions for the selected item.

[0097] In an embodiment, the cards application 102 may be configured to track viewing history of consumers. For example, when a user views a card and/or story, the cards application 102 may record data in the database 116 indicating the viewing by the user. The recorded data may not only indicate the user's identity, but may also indicate any groups associated with the user. For example, if a user from a marketing department group views a card, a number of views for that card from the marketing department group may automatically increment. This information may be viewed, for example, by an administrator and/or other users. This information may be useful in a variety of ways. For example, such data may indicate that a particular group is focusing too heavily on a particular card, author, type of card, etc. The information may also be useful in tracking what kinds of information interests a particular group for directing further analytics.

[0098] Cards 302 may have multiple "frames." A frame may be created by a user to focus on a single insight which may be derived from the data shown in the card 302 and may comprise a combination of textual and graphical cues. For example, a user may create a first frame that, when selected, highlights a portion of a card 302 to emphasize the highlighted portion. An individual frame may also have associated annotations.

[0099] The systems and methods described herein may provide several benefits. For example, the systems and methods described herein may allow users to more easily gather and aggregate data from various different sources while maintaining an ability to access the various sources. This may be possible through use of a flexible API that is configured to facilitate communication between multiple applications. The systems and methods described herein may also allow users to quickly and efficiently search, locate, and access various analytical data. This may be possible because related (and/or all) card data items may be stored in a database (e.g., database 116) associated with the cards application 102, rather than in disparate databases associated with various external applications. Thus, related (and/or all) card

data items may be efficiently indexed and searched by the application. Moreover, the systems and methods described herein provide an ability to effectively organize data for consumption, thus further improving accessibility of desired information.

Implementation Mechanisms

[0100] According to one embodiment, the techniques described herein are implemented by one or more special-purpose computing devices. The special-purpose computing devices may be hard-wired to perform the techniques, or may include digital electronic devices such as one or more application-specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs) that are persistently programmed to perform the techniques, or may include one or more general purpose hardware processors programmed to perform the techniques pursuant to program instructions in firmware, memory, other storage, or a combination. Such special-purpose computing devices may also combine custom hard-wired logic, ASICs, or FPGAs with custom programming to accomplish the techniques. The special-purpose computing devices may be desktop computer systems, server computer systems, portable computer systems, handheld devices, networking devices or any other device or combination of devices that incorporate hard-wired and/or program logic to implement the techniques.

[0101] Computing device(s) are generally controlled and coordinated by operating system software, such as iOS, Android, Chrome OS, Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10, Windows Server, Windows CE, Unix, Linux, SunOS, Solaris, iOS, BlackBerry OS, VxWorks, or other compatible operating systems. In other embodiments, the computing device may be controlled by a proprietary operating system. Conventional operating systems control and schedule computer processes for execution, perform memory management, provide file system, networking, I/O services, and provide a user interface functionality, such as a graphical user interface ("GUI"), among other things.

[0102] For example, Figure 9 is a block diagram that illustrates a computer system 900 upon which various embodiments may be implemented. For example, any of the computing devices discussed herein may include some or all of the components and/or functionality of the computer system 900.

[0103] Computer system 900 includes a bus 902 or other communication mechanism for communicating information, and a hardware processor, or multiple processors, 904 coupled with bus 902 for processing information. Hardware processor(s) 904 may be, for example, one or more general purpose microprocessors.

[0104] Computer system 900 also includes a main memory 906, such as a random access memory (RAM), cache and/or other dynamic storage devices, coupled to bus 902 for storing information and instructions to be executed by processor 904. Main memory 906 also may

be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 904. Such instructions, when stored in storage media accessible to processor 904, render computer system 900 into a special-purpose machine that is customized to perform the operations specified in the instructions. Main memory 906 may also store cached data, such as zoom levels and maximum and minimum sensor values at each zoom level.

[0105] Computer system 900 further includes a read only memory (ROM) 908 or other static storage device coupled to bus 902 for storing static information and instructions for processor 904. A storage device 910, such as a magnetic disk, optical disk, or USB thumb drive (Flash drive), etc., is provided and coupled to bus 902 for storing information and instructions. For example, the storage device 910 may store measurement data obtained from a plurality of sensors.

[0106] Computer system 900 may be coupled via bus 902 to a display 912, such as a cathode ray tube (CRT) or LCD display (or touch screen), for displaying information to a computer user. For example, the display 912 can be used to display any of the user interfaces described herein with respect to Figures 1 through 11D. An input device 914, including alphanumeric and other keys, is coupled to bus 902 for communicating information and command selections to processor 904. Another type of user input device is cursor control 416, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 904 and for controlling cursor movement on display 912. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane. In some embodiments, the same direction information and command selections as cursor control may be implemented via receiving touches on a touch screen without a cursor.

[0107] Computing system 900 may include a user interface module to implement a GUI that may be stored in a mass storage device as executable software codes that are executed by the computing device(s). This and other modules may include, by way of example, components, such as software components, object-oriented software components, class components and task components, processes, functions, attributes, procedures, subroutines, segments of program code, drivers, firmware, microcode, circuitry, data, databases, data structures, tables, arrays, and variables.

[0108] In general, the word "module," as used herein, refers to logic embodied in hardware or firmware, or to a collection of software instructions, possibly having entry and exit points, written in a programming language, such as, for example, Java, Lua, C or C++. A software module may be compiled and linked into an executable program, installed in a dynamic link library, or may be written in an interpreted programming language such as, for example, BASIC, Perl, or Python. It will be appreciated that soft-

ware modules may be callable from other modules or from themselves, and/or may be invoked in response to detected events or interrupts. Software modules configured for execution on computing devices may be provided on a computer readable medium, such as a compact disc, digital video disc, flash drive, magnetic disc, or any other tangible medium, or as a digital download (and may be originally stored in a compressed or installable format that requires installation, decompression or decryption prior to execution). Such software code may be stored, partially or fully, on a memory device of the executing computing device, for execution by the computing device. Software instructions may be embedded in firmware, such as an EPROM. It will be further appreciated that hardware modules may be comprised of connected logic units, such as gates and flip-flops, and/or may be comprised of programmable units, such as programmable gate arrays or processors. The modules or computing device functionality described herein are preferably implemented as software modules, but may be represented in hardware or firmware. Generally, the modules described herein refer to logical modules that may be combined with other modules or divided into sub-modules despite their physical organization or storage

[0109] Computer system 900 may implement the techniques described herein using customized hard-wired logic, one or more ASICs or FPGAs, firmware and/or program logic which in combination with the computer system causes or programs computer system 900 to be a special-purpose machine. According to one embodiment, the techniques herein are performed by computer system 900 in response to processor(s) 904 executing one or more sequences of one or more instructions contained in main memory 906. Such instructions may be read into main memory 906 from another storage medium, such as storage device 910. Execution of the sequences of instructions contained in main memory 906 causes processor(s) 904 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions.

[0110] The term "non-transitory media," and similar terms, as used herein refers to any media that store data and/or instructions that cause a machine to operate in a specific fashion. Such non-transitory media may comprise non-volatile media and/or volatile media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 910. Volatile media includes dynamic memory, such as main memory 906. Common forms of non-transitory media include, for example, a floppy disk, a flexible disk, hard disk, solid state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage medium, any physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, NVRAM, any other memory chip or cartridge, and networked versions of the same.

[0111] Non-transitory media is distinct from but may

be used in conjunction with transmission media. Transmission media participates in transferring information between non-transitory media. For example, transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 902. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0112] Various forms of media may be involved in carrying one or more sequences of one or more instructions to processor 904 for execution. For example, the instructions may initially be carried on a magnetic disk or solid state drive of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 900 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infrared detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 902. Bus 902 carries the data to main memory 906, from which processor 904 retrieves and executes the instructions. The instructions received by main memory 906 may retrieve and execute the instructions. The instructions received by main memory 906 may optionally be stored on storage device 910 either before or after execution by processor 904.

[0113] Computer system 900 also includes a communication interface 918 coupled to bus 902. Communication interface 918 provides a two-way data communication coupling to a network link 920 that is connected to a local network 922. For example, communication interface 918 may be an integrated services digital network (ISDN) card, cable modem, satellite modem, or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 918 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN (or WAN component to communicate with a WAN). Wireless links may also be implemented. In any such implementation, communication interface 918 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0114] Network link 920 typically provides data communication through one or more networks to other data devices. For example, network link 920 may provide a connection through local network 922 to a host computer 924 or to data equipment operated by an Internet Service Provider (ISP) 926. ISP 926 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 928. Local network 922 and Internet 928 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 920 and through communication interface 918, which carry the digital data to and from computer system 900, are exam-

ple forms of transmission media.

[0115] Computer system 900 can send messages and receive data, including program code, through the network(s), network link 920 and communication interface 918. In the Internet example, a server 930 might transmit a requested code for an application program through Internet 928, ISP 926, local network 922 and communication interface 918.

[0116] The received code may be executed by processor 904 as it is received, and/or stored in storage device 910, or other non-volatile storage for later execution.

Terminology

[0117] Each of the processes, methods, and algorithms described in the preceding sections may be embodied in, and fully or partially automated by, code modules executed by one or more computer systems or computer processors comprising computer hardware. The processes and algorithms may be implemented partially or wholly in application-specific circuitry.

[0118] The various features and processes described above may be used independently of one another, or may be combined in various ways. All possible combinations and subcombinations are intended to fall within the scope of this disclosure. In addition, certain method or process blocks may be omitted in some implementations. The methods and processes described herein are also not limited to any particular sequence, and the blocks or states relating thereto can be performed in other sequences that are appropriate. For example, described blocks or states may be performed in an order other than that specifically disclosed, or multiple blocks or states may be combined in a single block or state. The example blocks or states may be performed in serial, in parallel, or in some other manner. Blocks or states may be added to or removed from the disclosed example embodiments. The example systems and components described herein may be configured differently than described. For example, elements may be added to, removed from, or rearranged compared to the disclosed example embodiments.

[0119] Conditional language, such as, among others, "can," "could," "might," or "may," unless specifically stated otherwise, or otherwise understood within the context as used, is generally intended to convey that certain embodiments include, while other embodiments do not include, certain features, elements and/or steps. Thus, such conditional language is not generally intended to imply that features, elements and/or steps are in any way required for one or more embodiments or that one or more embodiments necessarily include logic for deciding, with or without user input or prompting, whether these features, elements and/or steps are included or are to be performed in any particular embodiment.

[0120] The term "comprising" as used herein should be given an inclusive rather than exclusive interpretation. For example, a general purpose computer comprising

one or more processors should not be interpreted as excluding other computer components, and may possibly include such components as memory, input/output devices, and/or network interfaces, among others.

[0121] Any process descriptions, elements, or blocks in the flow diagrams described herein and/or depicted in the attached figures should be understood as potentially representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process. Alternate implementations are included within the scope of the embodiments described herein in which elements or functions may be deleted, executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those skilled in the art.

[0122] It should be emphasized that many variations and modifications may be made to the above-described embodiments, the elements of which are to be understood as being among other acceptable examples. All such modifications and variations are intended to be included herein within the scope of this disclosure. The foregoing description details certain embodiments of the invention. It will be appreciated, however, that no matter how detailed the foregoing appears in text, the invention can be practiced in many ways. As is also stated above, it should be noted that the use of particular terminology when describing certain features or aspects of the invention should not be taken to imply that the terminology is being re-defined herein to be restricted to including any specific characteristics of the features or aspects of the invention with which that terminology is associated. The scope of the invention should therefore be construed in accordance with the appended claims and any equivalents thereof.

EXAMPLE ENUMERATED EMBODIMENTS

[0123] Further examples of embodiments according to the present disclosure are defined, without limitation, by the following Example Enumerated Embodiments (EEEs):

EEE 1: A computing system configured to access and display data from a source software application, the computing system comprising:

one or more transitory or non-transitory computer-readable storage devices configured to store:

an application programming interface (API) configured to be called by a source software application and facilitate transfer of data from the source software application to a target software application in a standardized way; and
a target software application; and

one or more hardware computer processors in communication with one or more transitory or non-transitory computer readable storage devices and configured to execute the target software application to cause the computer system to:

- receive the data, via the API, from the source software application;
- generate a link configured to perform a call to the source software application to cause the source software application to display a user interface associated with the data;
- format the data to generate an interactive graphical unit; and
- generate user interface data for rendering an interactive user interface on a computing device, the interactive user interface including the interactive graphical unit and the link.

EEE 2: The computing system of EEE 1, wherein the one or more hardware computer processors are further configured to execute the target software application to cause the computer system to receive user identifying information from the source software application and transmit a list of group identifiers associated with the user to the source software application in response to the user identifying information, the group identifiers identifying groups of interactive graphical units which the interactive graphical unit may be added to.

EEE 3: The computing system of EEE 1 or EEE 2, wherein formatting the data comprises modifying the data to a form that is suitable for use by the target software application.

EEE 4: The computing system of EEE 3, wherein modifying the data comprises removing data that is not necessary for displaying the data.

EEE 5: The computing system of any of EEEs 1-4, wherein the one or more hardware computer processors is further configured to execute the target software application to cause the computer system to associate the interactive graphical unit with a unique identifier.

EEE 6: The computing system of EEE 5, wherein the unique identifier comprises at least one of an access control list (ACL), title, author, date of creation, type of data, and source of data.

EEE 7: The computing system of any of EEEs 1-6, wherein the one or more hardware computer processors is further configured to execute the target software application to cause the computer system to modify the unique identifier based on user input.

EEE 8: A computing system configured to transmit data to a target software application, the computing system comprising:

one or more transitory or non-transitory compu-

ter readable storage devices configured to store a source software application; and one or more hardware computer processors in communication with one or more transitory or non-transitory computer readable storage devices and configured to execute the source software application to cause the computer system to:

- perform a call to an Application Programming Interface (API);
- receive a selection of data from a user;
- determine a type of data associated with the selected data;
- transmit the selected data and the determined type of data, via the API, to the target software application;
- display a user interface associated with the selected data in response to a call from the target software application.

EEE 9: The computing system of EEE 8, wherein the one or more hardware computer processors are further configured to execute the source software application to cause the computer system to determine a set of the selected data needed to display an interactive graphical unit.

EEE 10: The computing system of EEE 8 or EEE 9, wherein the one or more hardware computer processors are further configured to execute the source software application to cause the computer system to receive identification information for the user.

EEE 11: The computing system of EEE 10, wherein the one or more hardware computer processors are further configured to execute the source software application to cause the computer system to transmit a request for a list of group identifiers associated with the user to the target software application and receive the list of group identifiers from the target software application.

EEE 12: The computing system of EEE 11, wherein the one or more hardware computer processors are further configured to execute the source software application to cause the computer system to receive a selection of a group identifier from the user and transmit the selected group identifier, via the API, to the target software application.

EEE 13: The computing system of any of EEEs 8-12, wherein the one or more hardware computer processors are further configured to execute the source software application to cause the computer system to receive acknowledgement from the target software application that an interactive graphical unit was created.

Claims

1. A computing system configured to access and display data from a source software application (104, 106, 108, 400), the computing system comprising:

one or more transitory or non-transitory computer readable storage devices configured to store:

an application programming interface - API - (112) configured to be called by a source software application (104, 106, 108, 400) to transfer artifact data from the source software application (104, 106, 108, 400) to a target software application (102); and a target software application (102); and

one or more hardware computer processors in communication with one or more transitory or non-transitory computer readable storage devices and configured to execute the target software application (102) to cause the computer system to:

receive the artifact data, via the API (112), from the source software application (104, 106, 108, 400); format the artifact data to generate an interactive graphical unit (302, 370); generate user interface data for rendering an interactive user interface (300) on a computing device (110), the interactive user interface (300) including the interactive graphical unit (302, 370); determine (705) if a user has write permission with respect to at least the interactive graphical unit (302, 307); if it is determined (705) that the user has write permission with respect to at least the interactive graphical unit (302, 307), and in response to an indication from the user that the interactive graphical unit (302, 370) or a group (202) of interactive graphical units (302, 370) comprising the interactive graphical unit (302, 370) is to be edited, performing at least one action of:

adding data to the interactive graphical unit (302, 307), changing data in the interactive graphical unit (302, 307), adding an artifact to the interactive graphical unit (302, 307), changing an artifact in the interactive graphical unit (302, 307), removing an artifact from the interactive graphical unit (302, 307), changing the interactive graphical unit

(302, 307), editing content of the interactive graphical unit (302, 307), formatting (710) the artifact data to create data for the interactive graphical unit (302, 370), adding (712) at least one annotation to the interactive graphical unit (302, 370), and/or adding (718) properties to associate the interactive graphical unit (302, 370) with a group (202) of interactive graphical units (302, 370).

5

10

15

20

25

30

35

40

45

50

55

2. The computing system of claim 1, wherein said performing said at least one action comprises adding data to, or changing data in, the interactive graphical unit (302, 370).

3. The computing system of claim 1, wherein said performing said at least one action comprises adding an artifact to, or changing an artifact in, or removing an artifact from, the interactive graphical unit (302, 370).

4. The computing system of claim 1, wherein said performing said at least one action comprises formatting (710) the artifact data to create data for the interactive graphical unit (302, 370).

5. The computing system of claim 1, wherein said performing said at least one action comprises adding (712) at least one annotation to the interactive graphical unit (302, 370).

6. The computing system of claim 1, wherein said performing said at least one action comprises adding (718) properties to associate the interactive graphical unit (302, 370) with the group (202) of interactive graphical units (302, 370).

7. The computing system of any of claims 1-6, wherein the one or more hardware computer processors are further configured to execute the target software application (102) to cause the computer system to generate a link (332), the link (332) being configured to perform a call to the source software application (104, 106, 108, 400) to cause the source software application (104, 106, 108, 400) to display a user interface associated with the artifact data, and wherein the interactive user interface (300) further includes the link (332).

8. The computing system of any of claims 1-7, wherein the one or more hardware computer processors are further configured to execute the target software application (102) to cause the computer system to receive user identifying information from the source

software application (104, 106, 108, 400) and transmit a list of group identifiers associated with the user to the source software application (104, 106, 108, 400) in response to the user identifying information, the group identifiers identifying groups (202) of interactive graphical units (302, 370) which the interactive graphical unit (302, 370) may be added to. 5

9. The computing system of any of claims 1-8, wherein formatting the artifact data comprises modifying the artifact data to a form that is suitable for use by the target software application (102). 10

10. The computing system of claim 4, wherein modifying the artifact data comprises removing artifact data that is not necessary for displaying the artifact data. 15

11. The computing system of any of claims 1-10, wherein the one or more hardware computer processors are further configured to execute the target software application (102) to cause the computer system to associate the interactive graphical unit (302, 370) with a unique identifier. 20

12. The computing system of claim 11, wherein the unique identifier comprises at least one of an access control list - ACL -, title, author, date of creation, type of data, and source of data. 25

13. The computing system of any of claims 1-12, wherein the one or more hardware computer processors is further configured to execute the target software application (102) to cause the computer system to modify the unique identifier based on user input. 30

14. A computer-implemented method performed under control of one or more hardware computing devices configured with specific computer executable instructions, the method comprising operations which include the operations recited in any of claims 1-13. 35 40

15. A computer-readable storage medium storing software instructions, wherein, in response to execution by a computing system having one or more hardware processors, the software instructions configure the computing system to perform operations comprising the operations recited in any of claims 1-13. 45

50

55

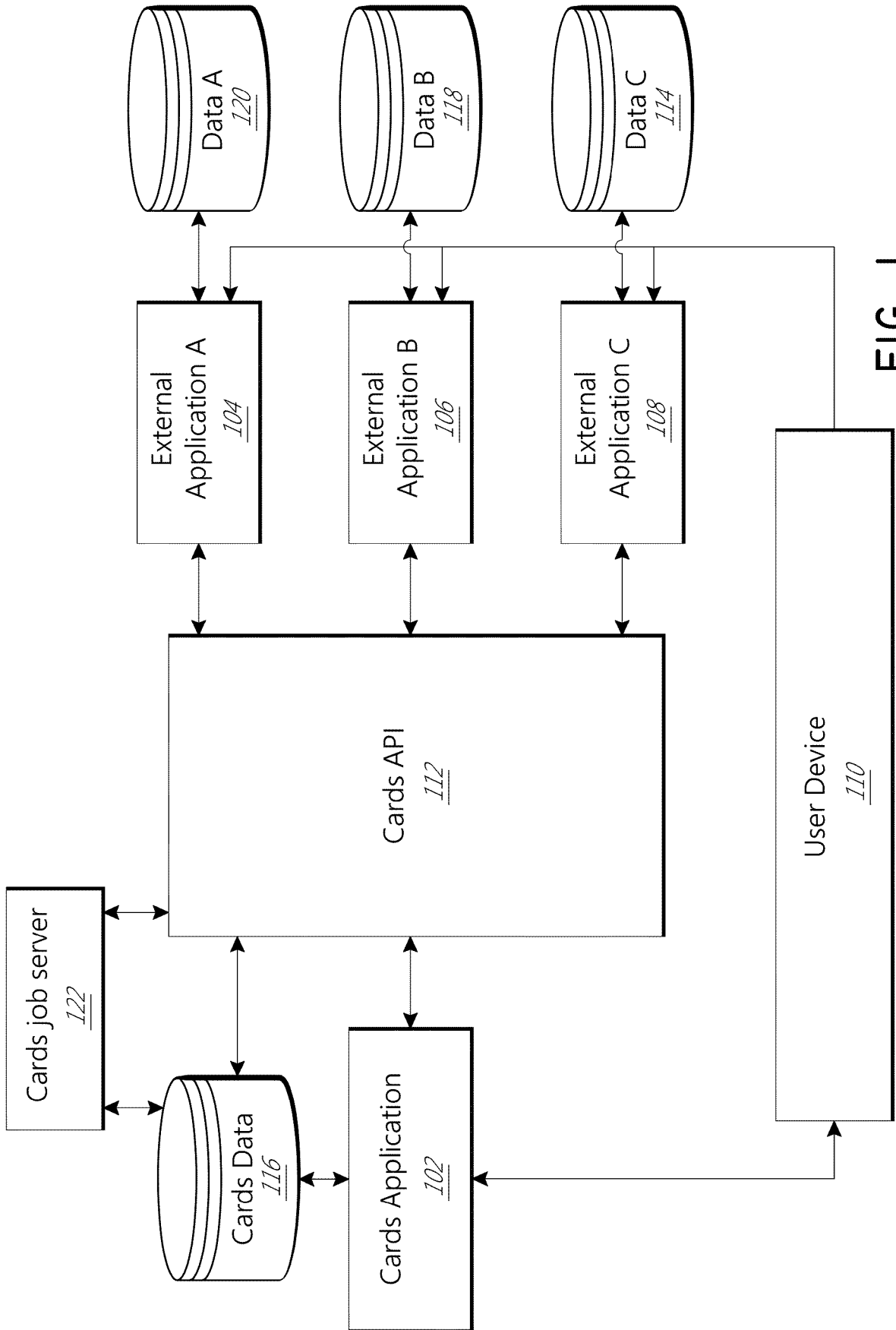


FIG. 1

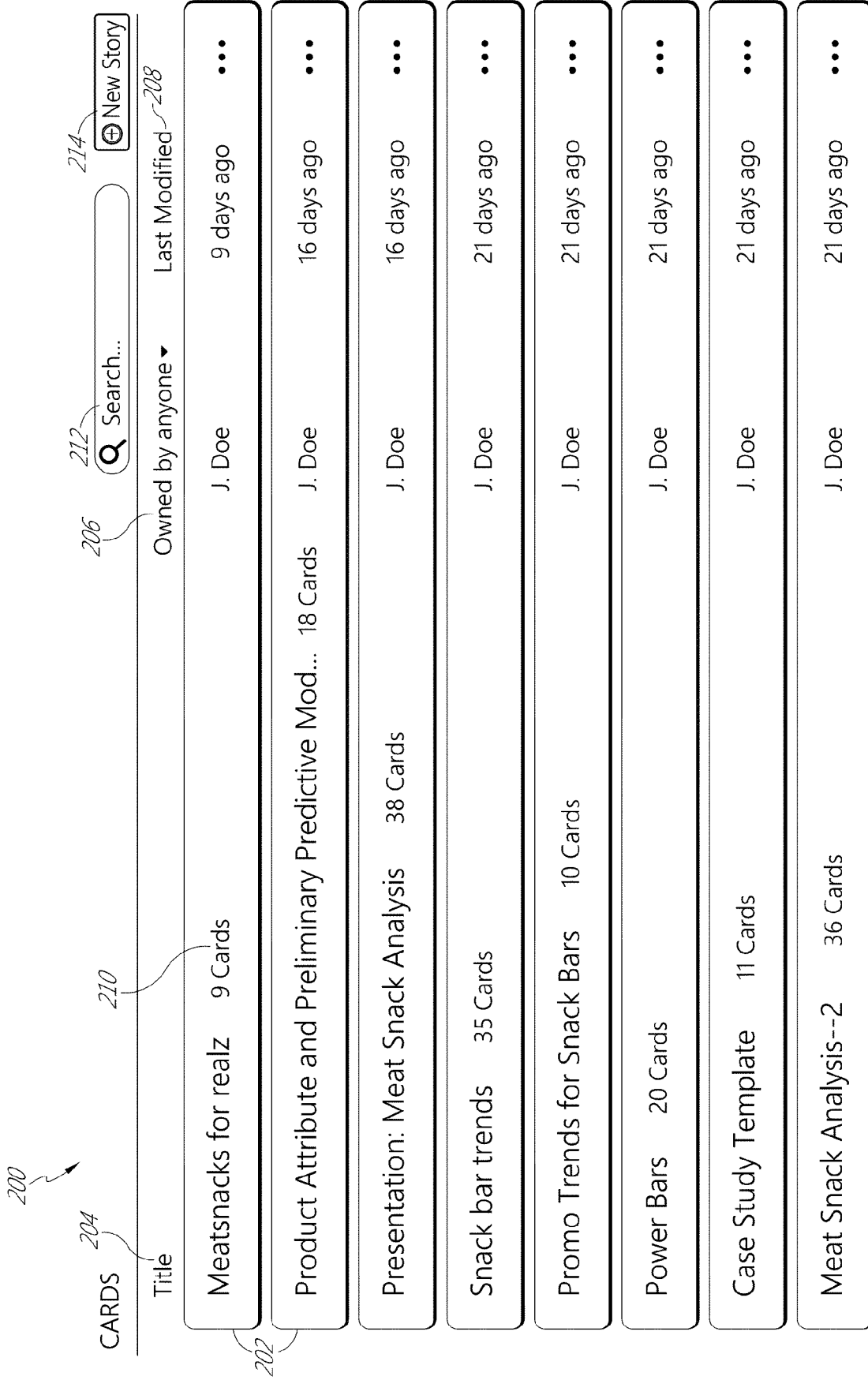


FIG. 2



FIG. 3A

Overall Sales by Region

329

370

1-Meat snack sales were highest in the South and West and grew overall from 2010 to 2014

328

CAGR:
 Midwest = 11%
 Northeast = 13%
 South = 13%
 West = 7%

326

- Midwest (N Central)
- Northeast
- South
- West

338

AUTHOR

Assign

ACTIONS

Duplicate

Copy To

Move

CARD TYPE

Chart

330

SUM of Projected Dol by YEAR vs Date

Derived from [ontology.panel-trip detail](#)

336 Remove

334 Edit

341 342 344 340

324

600.0m

500.0m

400.0m

300.0m

200.0m

100.0m

2010 2011 2012 2013 2014

Date

346

348

SUM of Projected Dol

2 comments

Favorite

Delete Card

Collapse

2-The Midwest and Northeast look similar to the South with XYZ dominating the market

3-In the West smaller brands, like ABC, are more competitive

FIG. 3B

Overall Sales by Region

1-Meat snack sales were highest in the South and West and grew overall from 2010 to 2014

CAGR:
 Midwest = 11%
 Northeast = 13%
 South = 13%
 West = 7%

- Midwest (N Central)
- Northeast
- South
- West

AUTHOR

Assign

ACTIONS

Duplicate

Copy To

Move

CARD TYPE

Chart

SUM of Projected Dol by YEAR vs Date

Derived from [ontology_panel-trip detail](#) Edit | Remove 352

Year	Total SUM of Projected Dol
2010	346
2011	346
2012	354
2013	354
2014	356

2 comments
 Favorite

Delete Card
 Collapse

2

3

2-The Midwest and Northeast look similar to the South with XYZ dominating the market

3-In the West smaller brands, like ABC, are more competitive

FIG. 3C

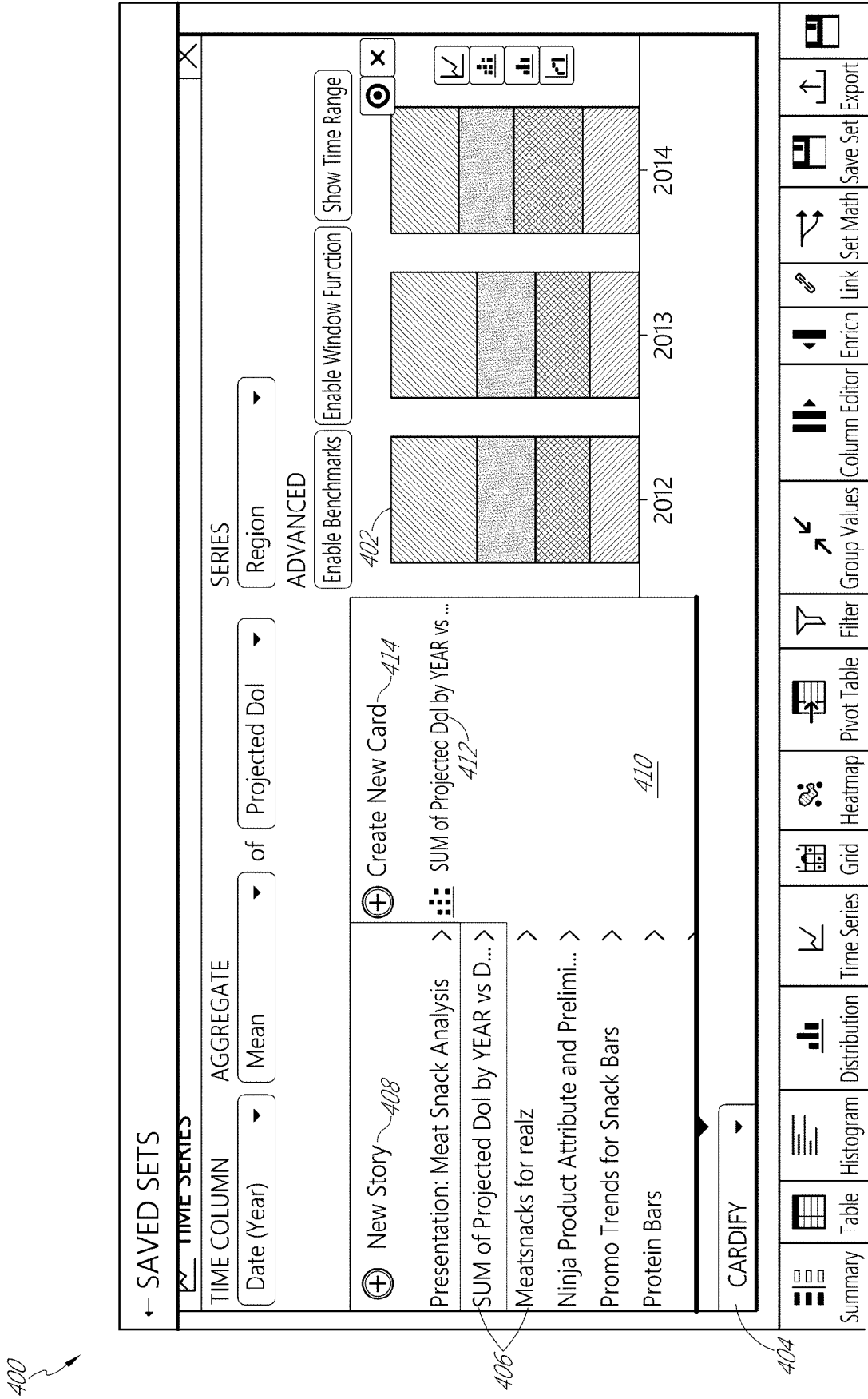


FIG. 4

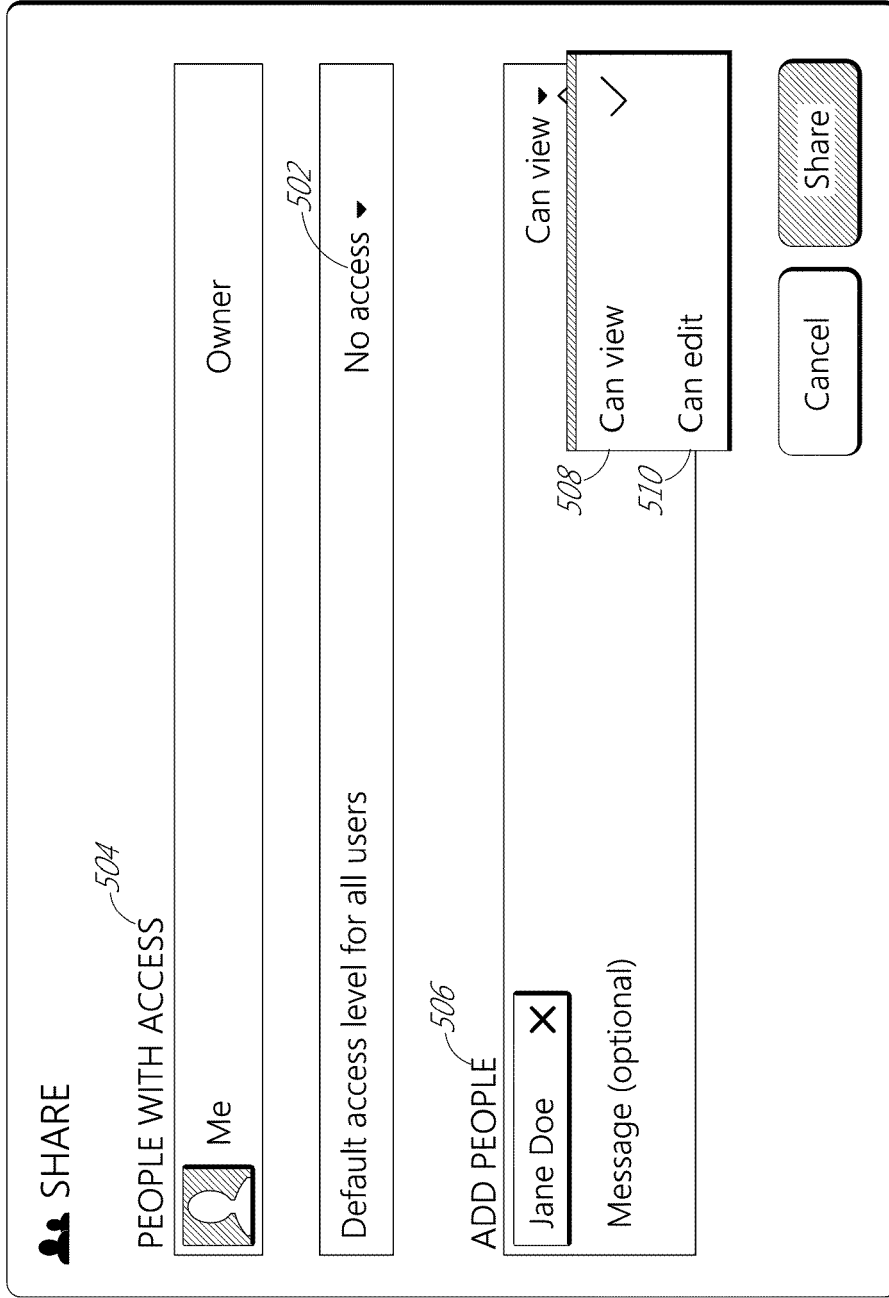


FIG. 5

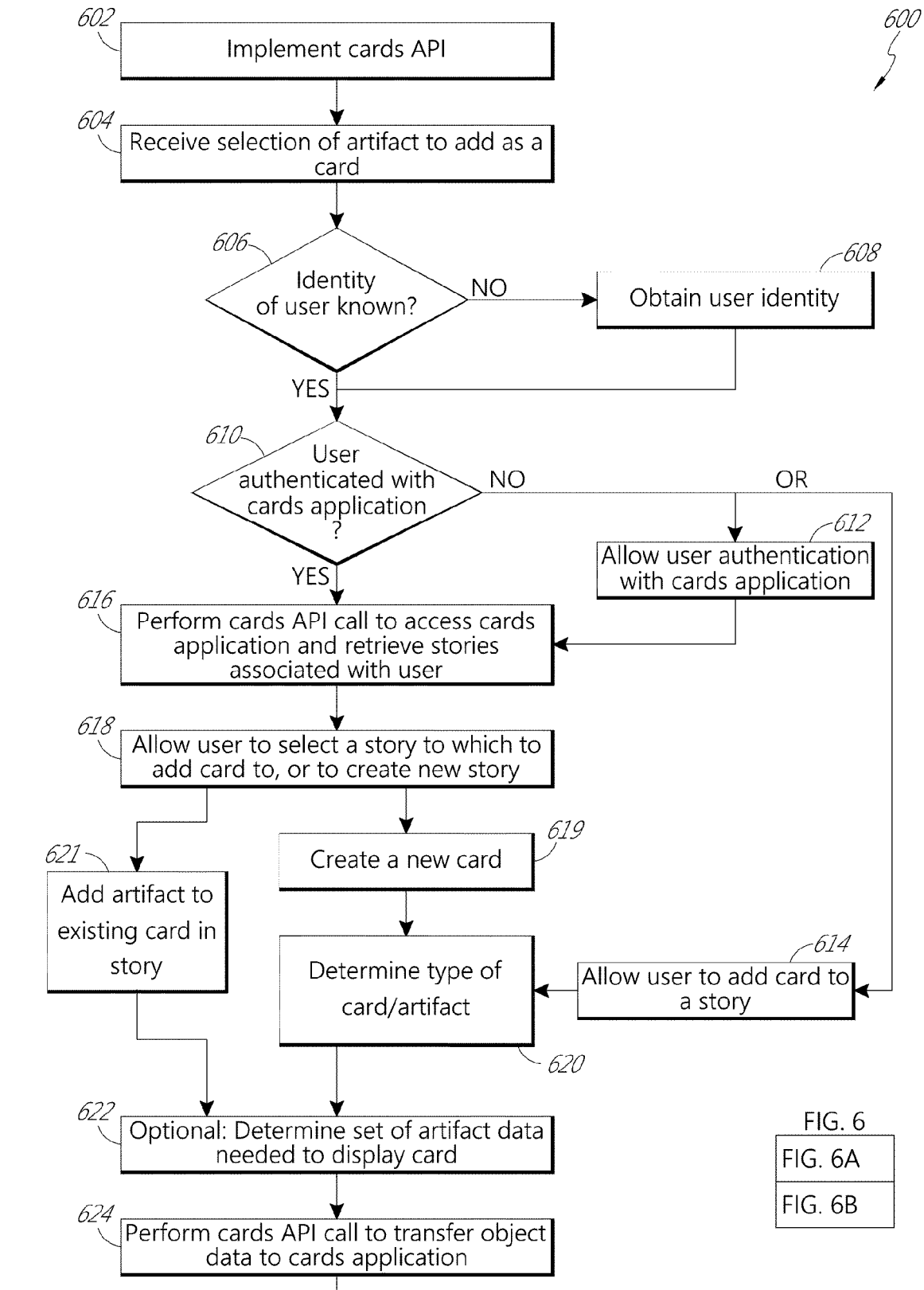


FIG. 6
FIG. 6A
FIG. 6B

FIG. 6A

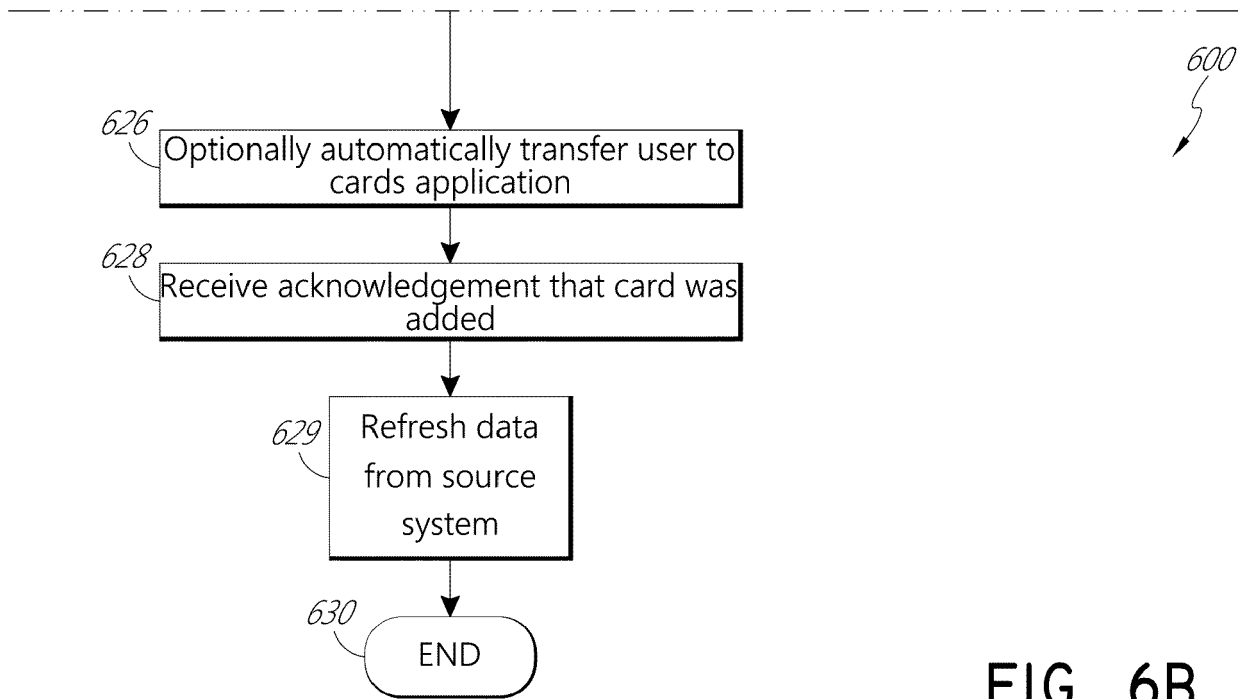


FIG. 6B

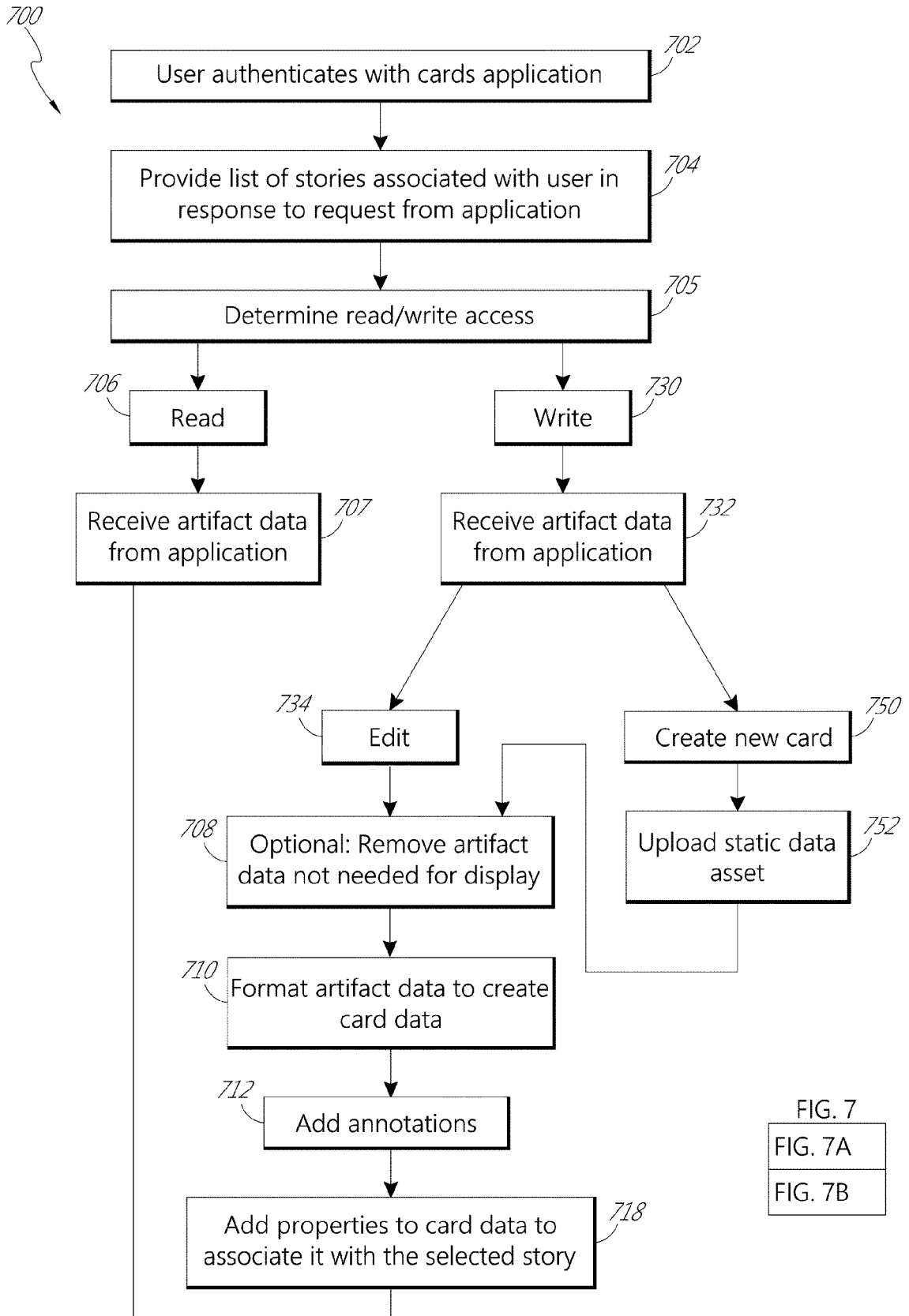


FIG. 7
 FIG. 7A
 FIG. 7B

FIG. 7A

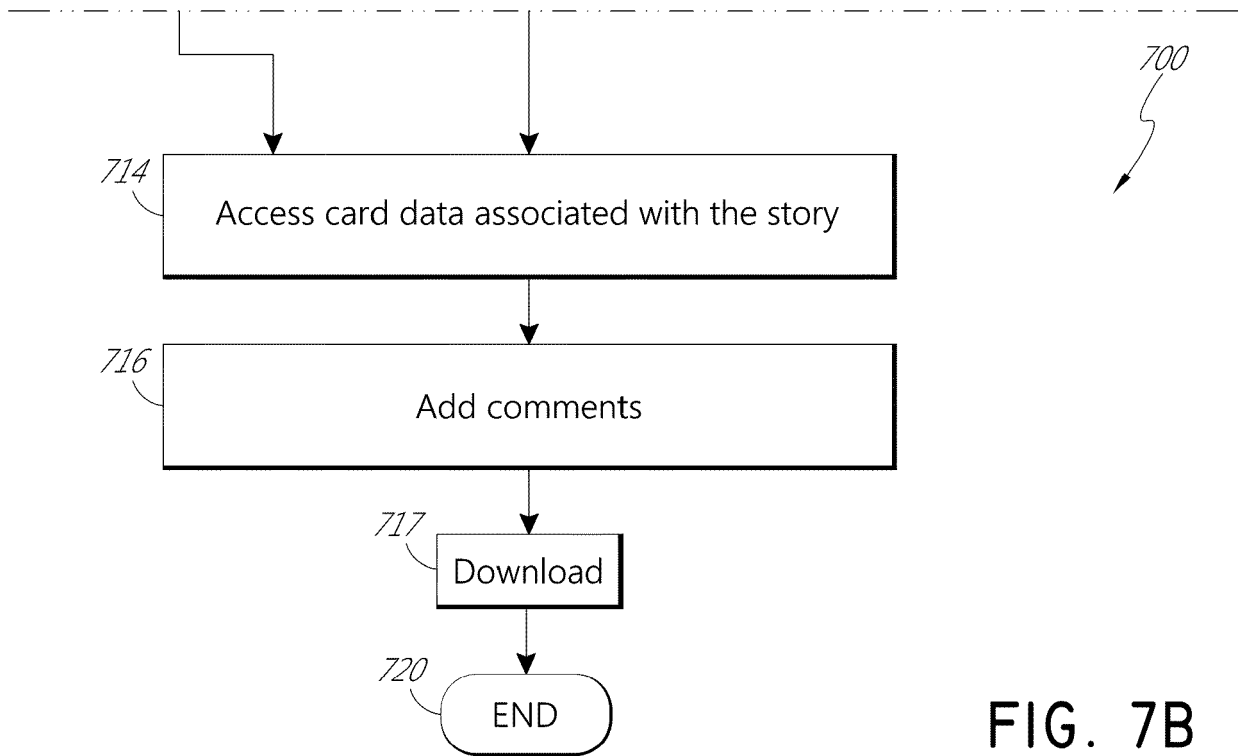


FIG. 7B

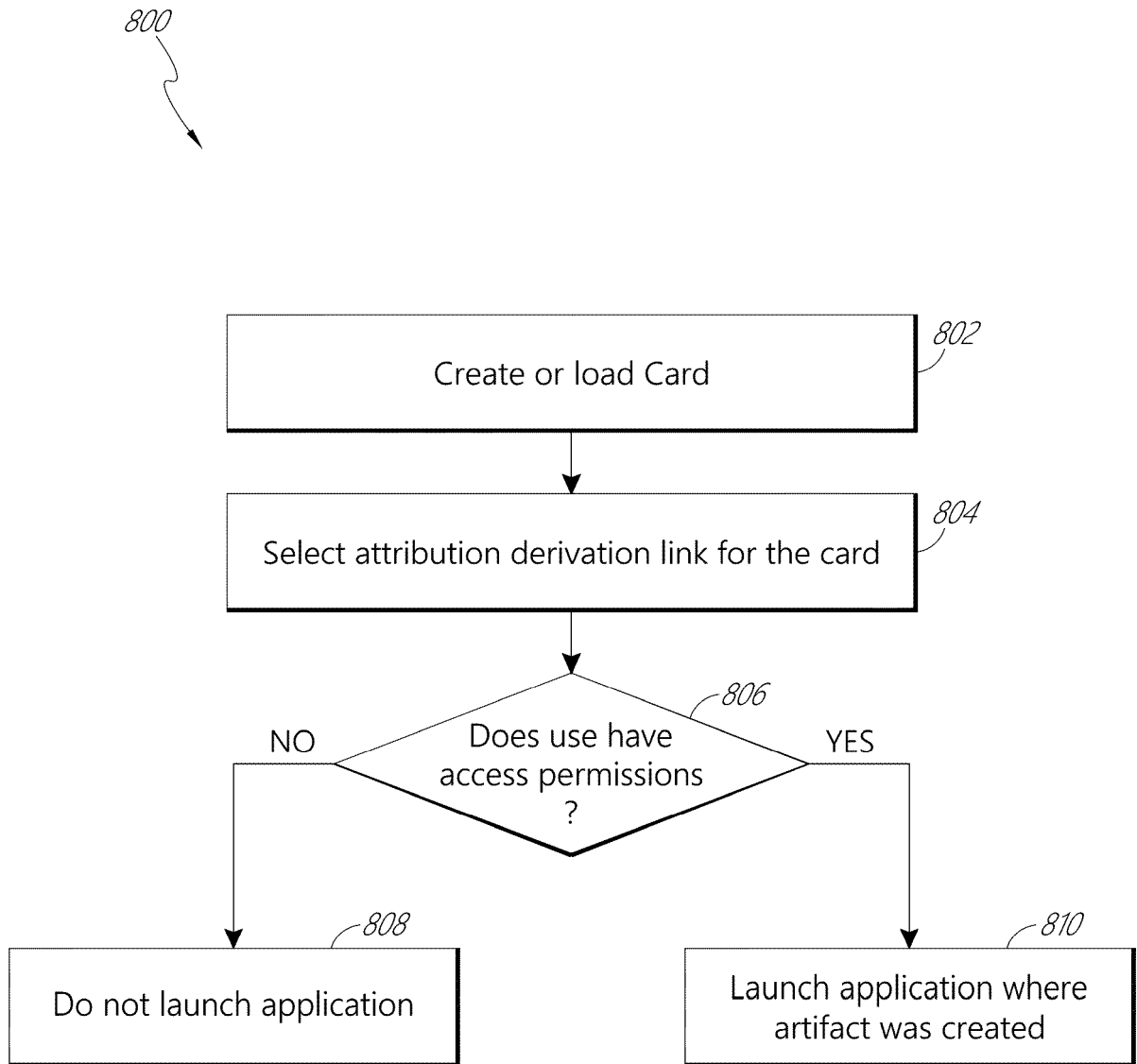


FIG. 8

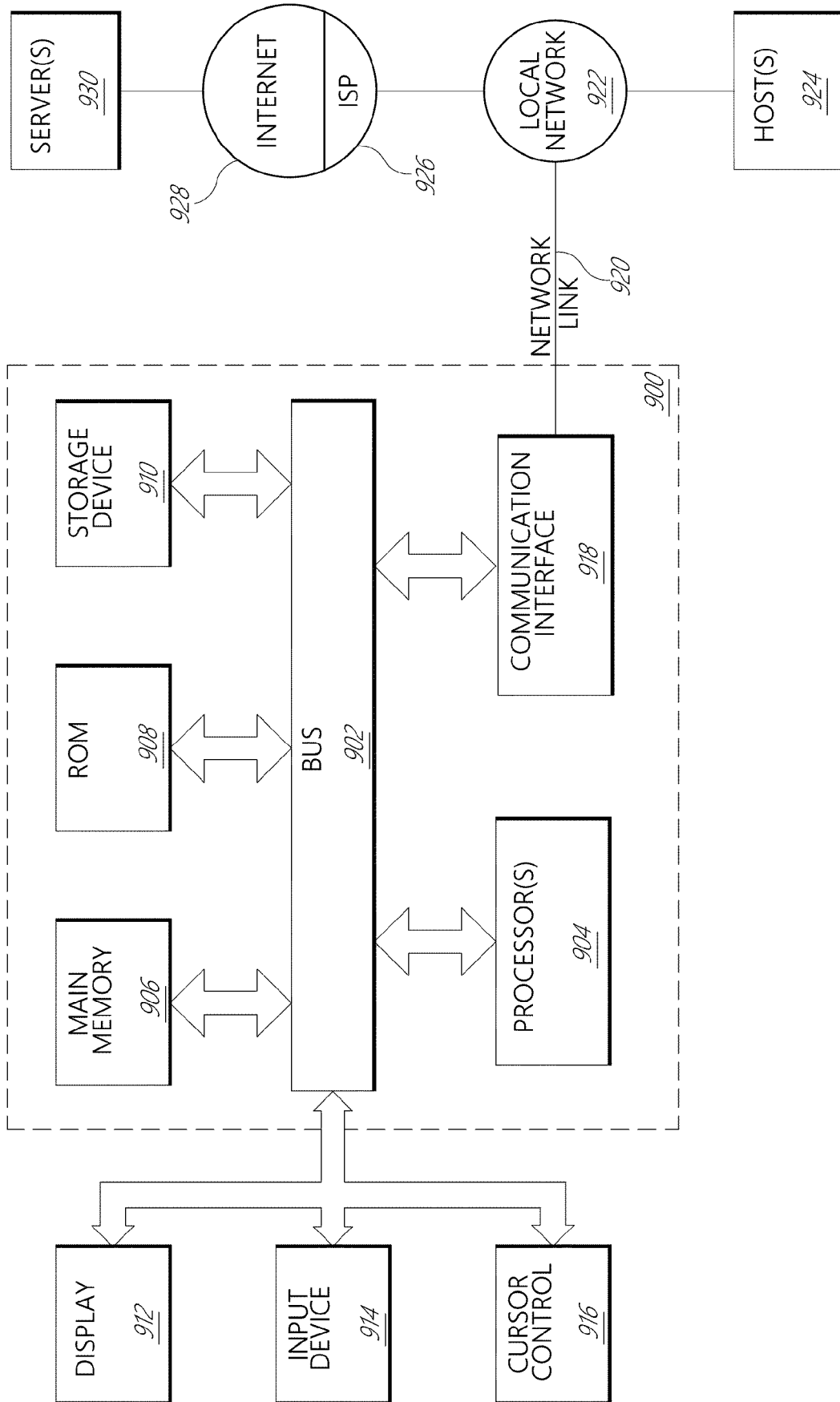


FIG. 9

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Non-patent literature cited in the description

- User Interface Design. **DILLON, A.** MacMillan Encyclopedia of Cognitive Science. MacMillan, 2003, vol. 4, 453-458 [0009]