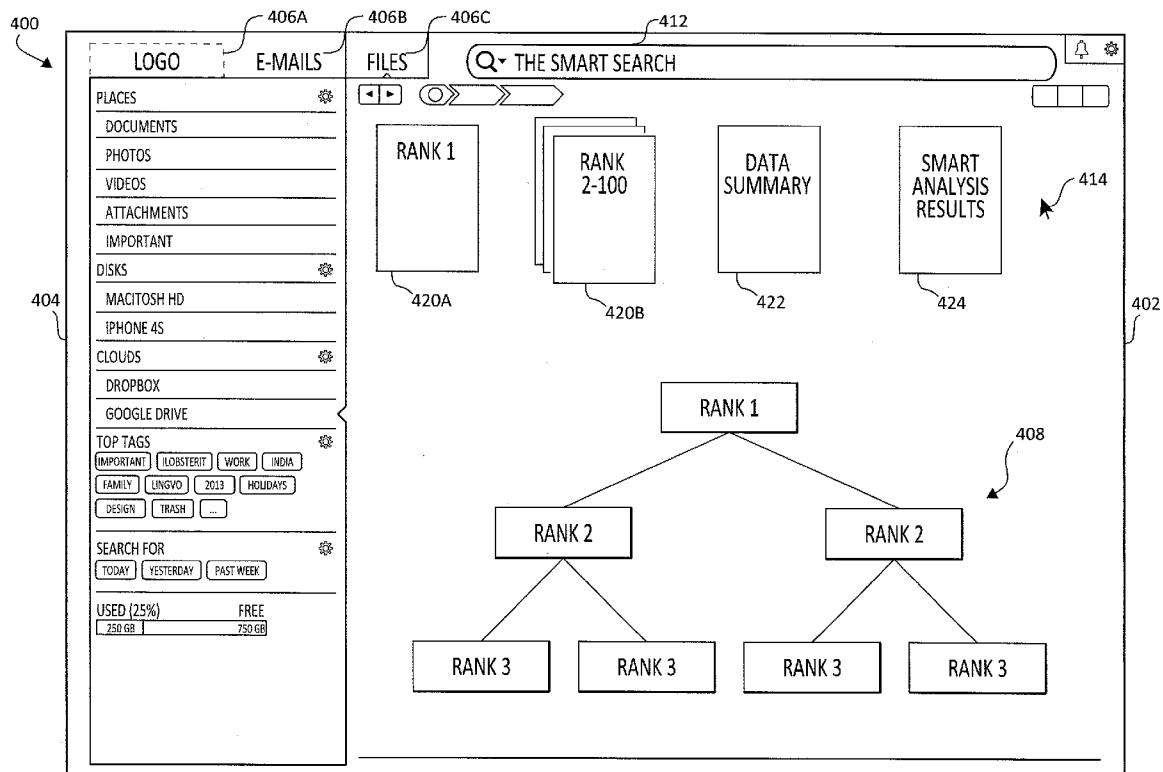


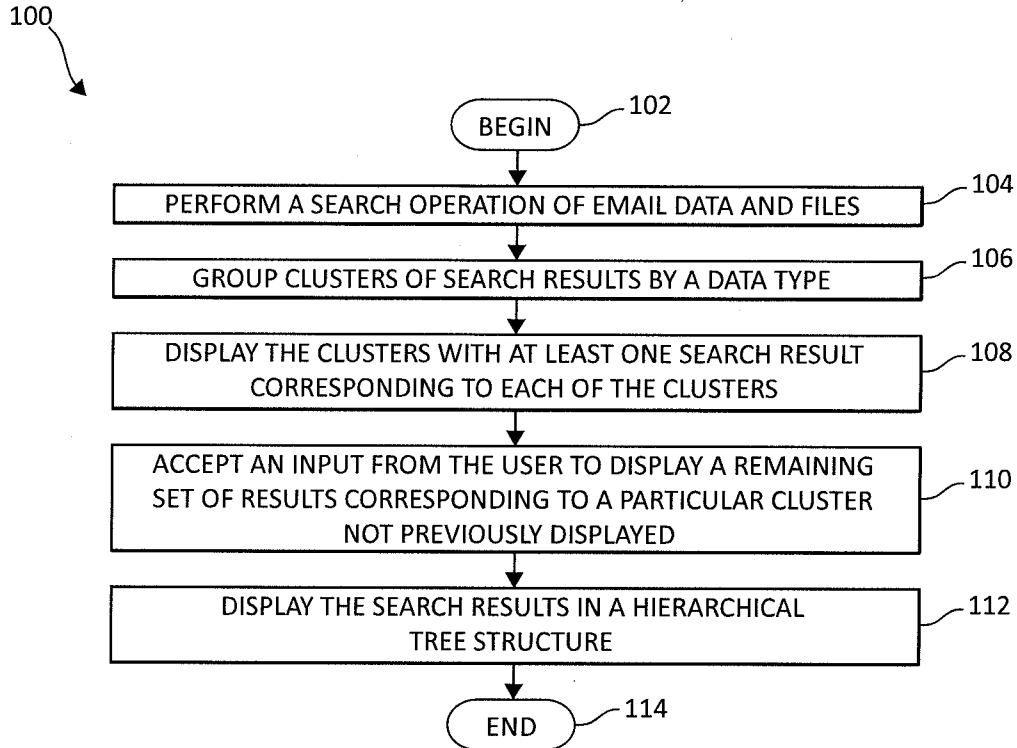
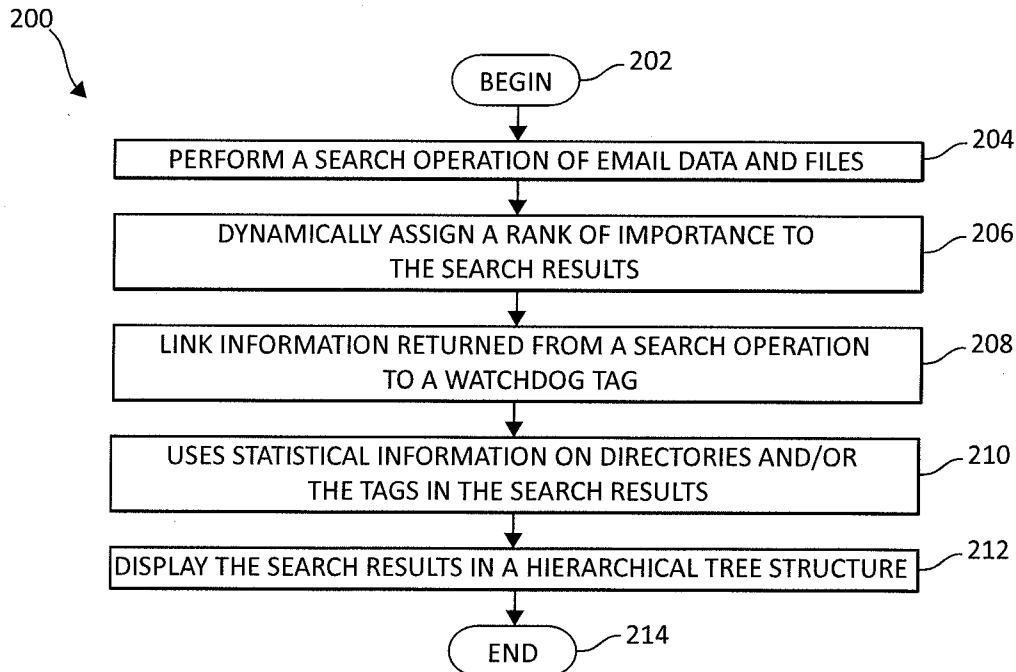


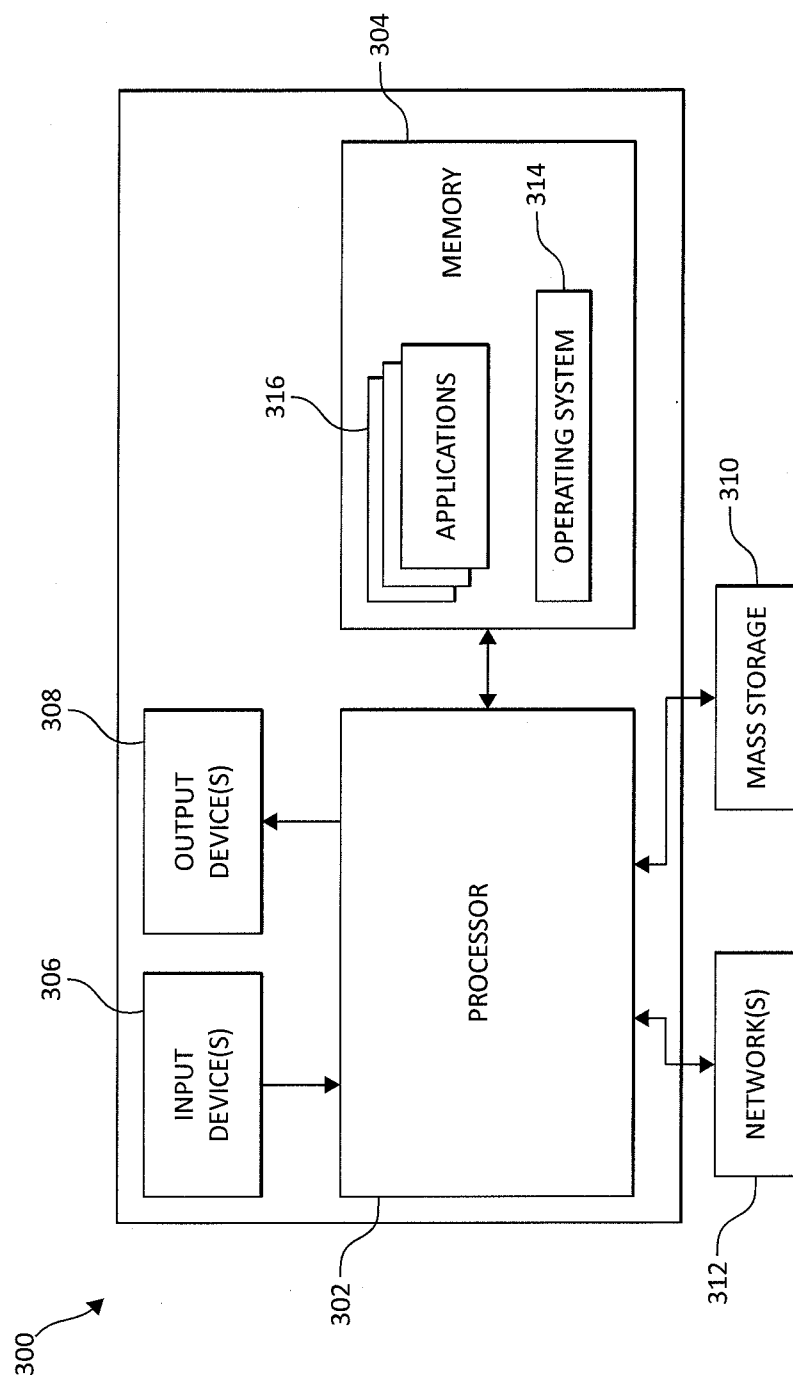
US 20150186478A1

(19) **United States**(12) **Patent Application Publication**  
**Yan et al.**(10) **Pub. No.: US 2015/0186478 A1**(43) **Pub. Date: Jul. 2, 2015**(54) **METHOD AND SYSTEM FOR TREE  
REPRESENTATION OF SEARCH RESULTS****Publication Classification**(71) Applicant: **ABBYY Development LLC**, Moscow  
(RU)(51) **Int. Cl.**  
**G06F 17/30** (2006.01)(72) Inventors: **David Yan**, Portola Valley, CA (US);  
**Konstantin Anisimovich**, Moscow (RU)(52) **U.S. Cl.**  
CPC .... **G06F 17/30554** (2013.01); **G06F 17/30327**  
(2013.01)(21) Appl. No.: **14/579,204**(22) Filed: **Dec. 22, 2014****Related U.S. Application Data**(60) Provisional application No. 61/922,597, filed on Dec.  
31, 2013.(57) **ABSTRACT**

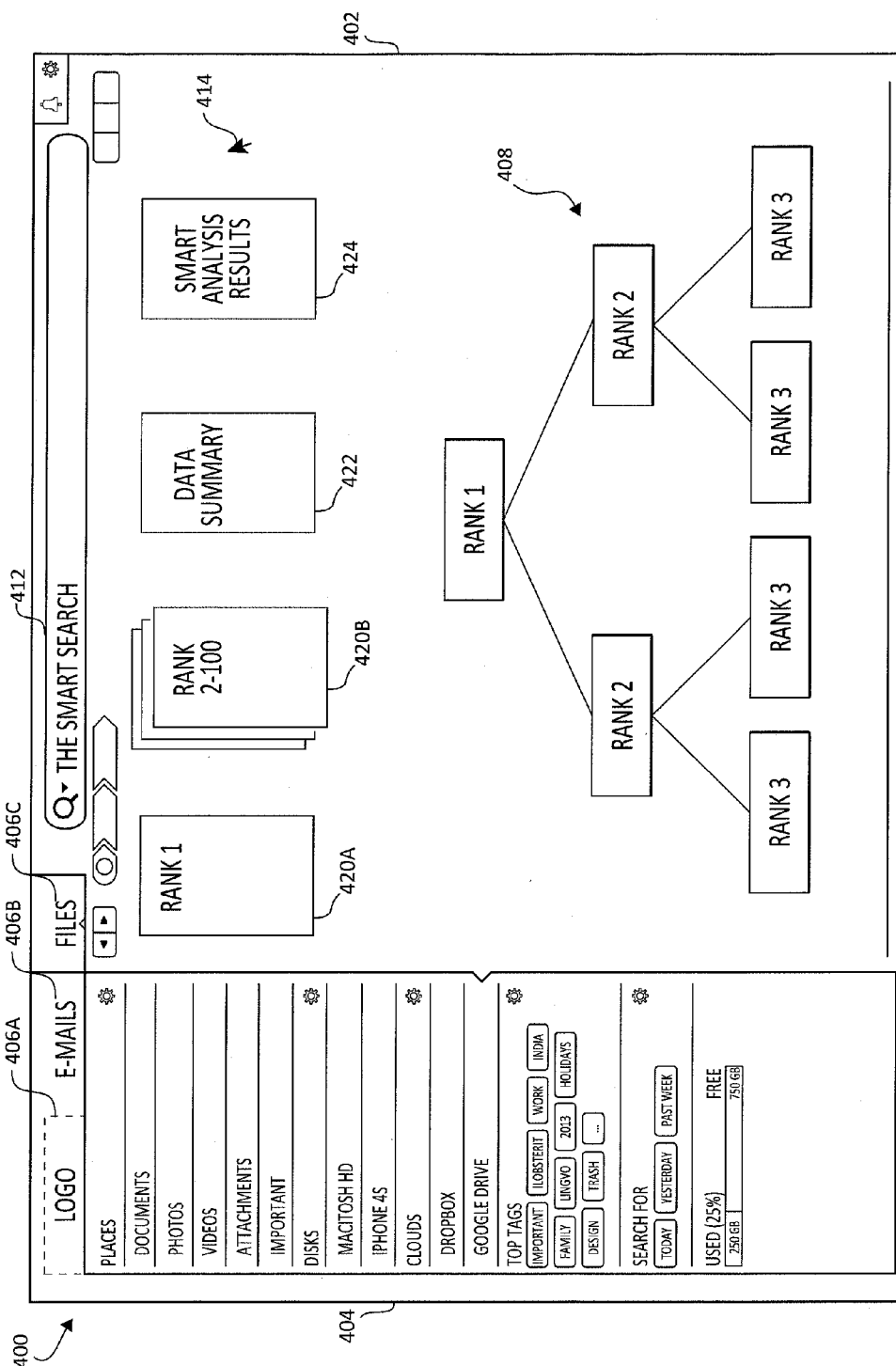
An illustrative method according to a set of instructions stored on a memory of a computing device includes receiving, by a processor of the computing device, a search input. The method further includes determining, by the processor, the presence of a plurality of electronic files related to the search input. The method further includes displaying, by the processor, on a graphical user interface (GUI), representative information of the plurality of electronic files related to the search input, where the representative information comprises a hierarchical tree structure.



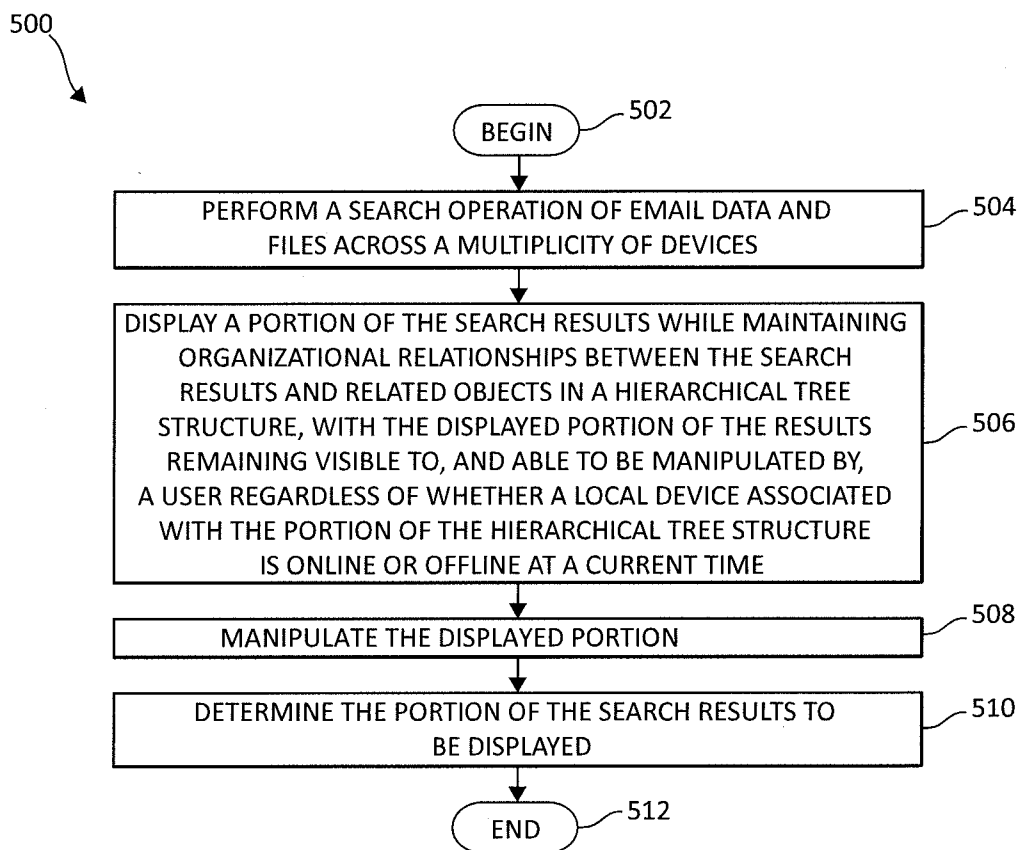
**FIG. 1****FIG. 2**



**FIG. 3**



**FIG. 4**

**FIG. 5**

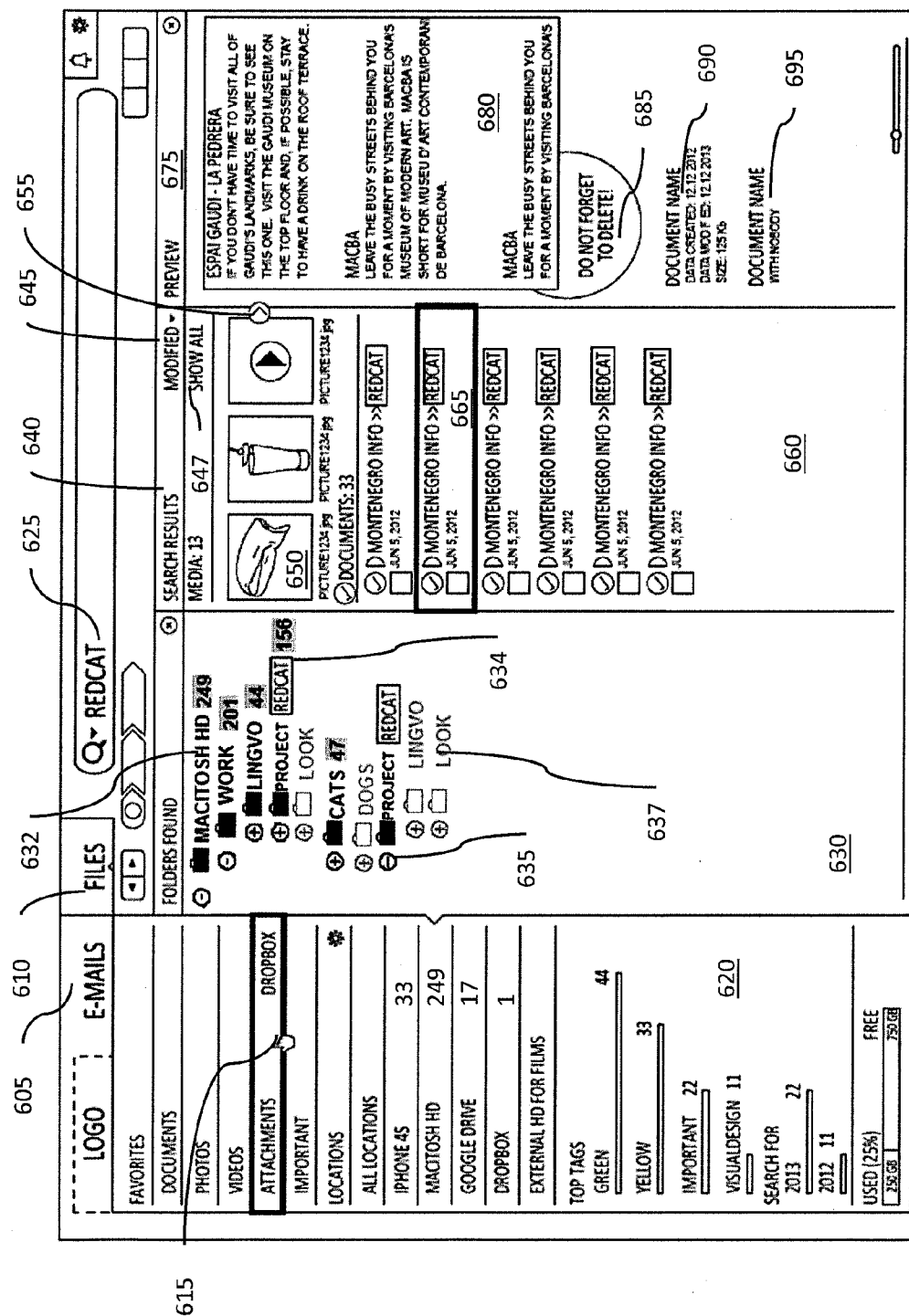


Fig. 6

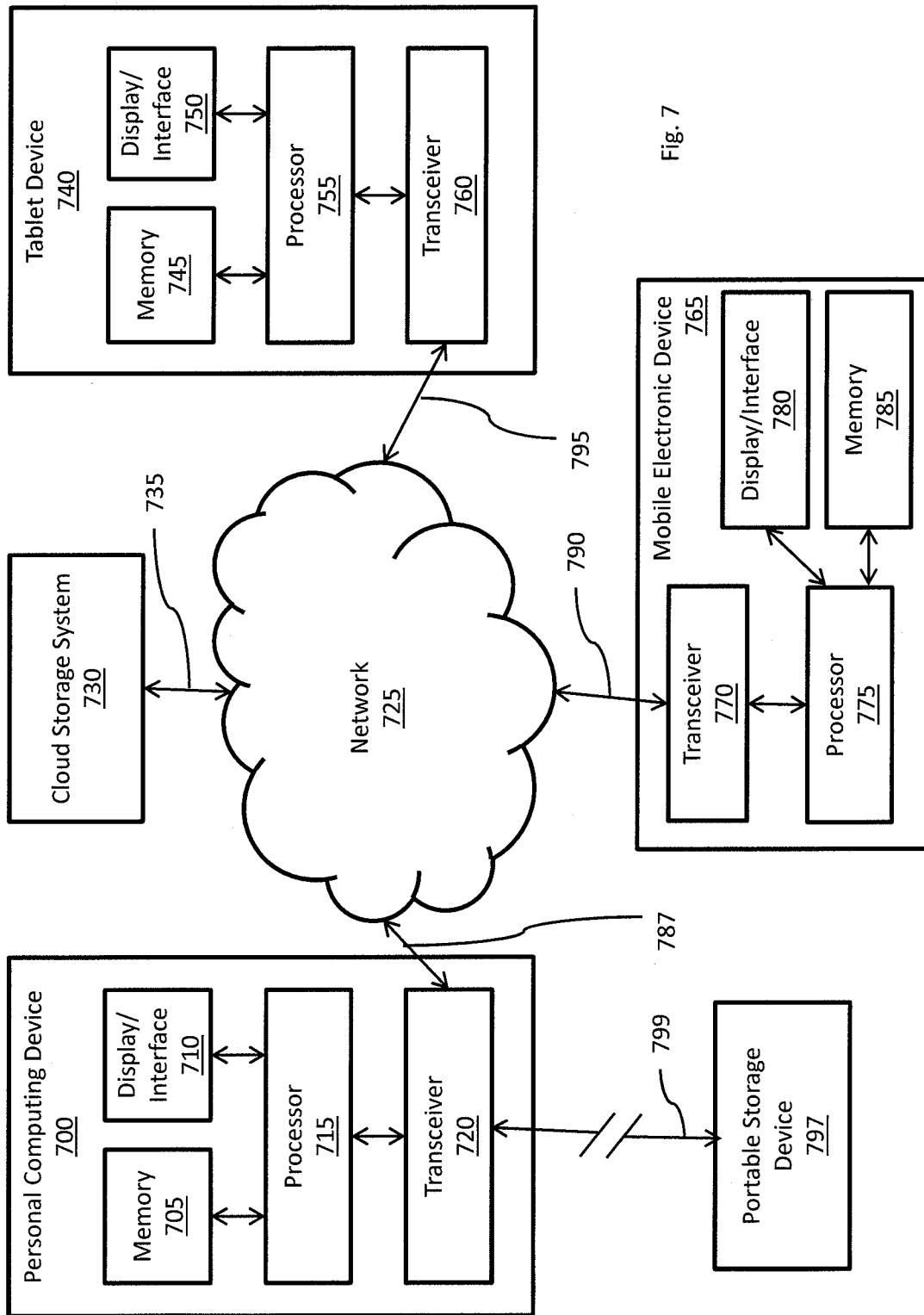


Fig. 7

Fig. 8

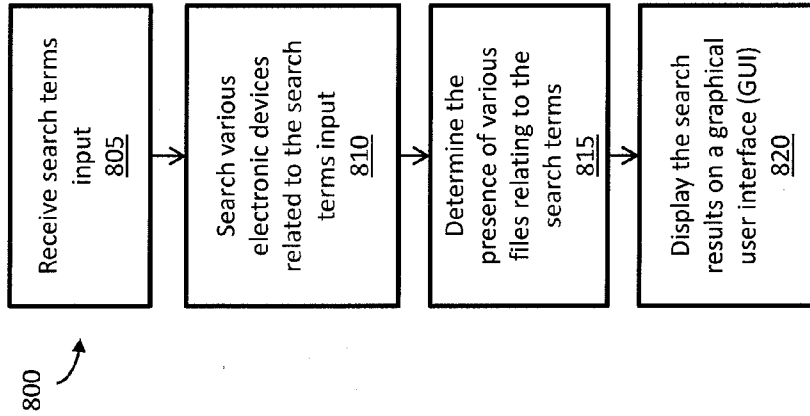


Fig. 9

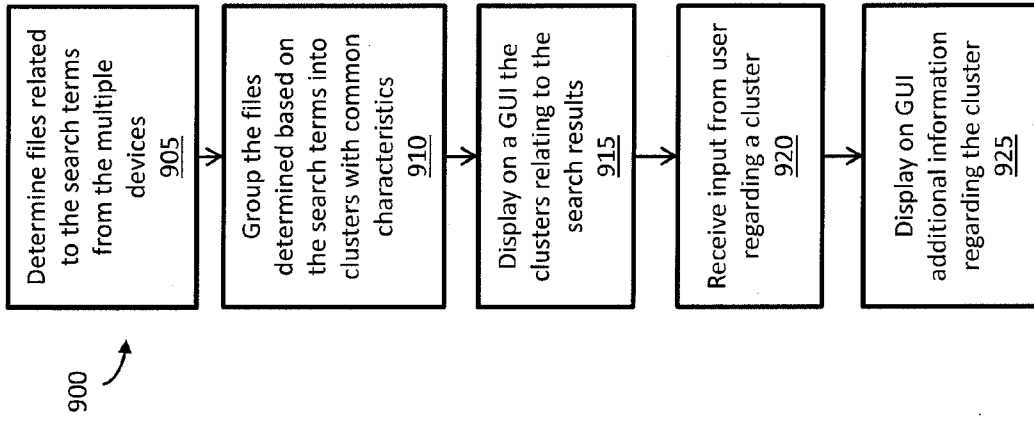
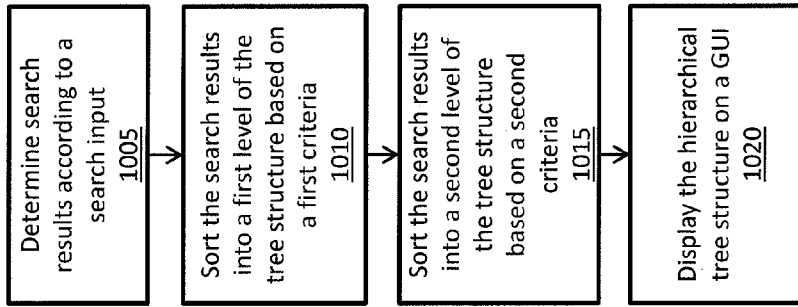
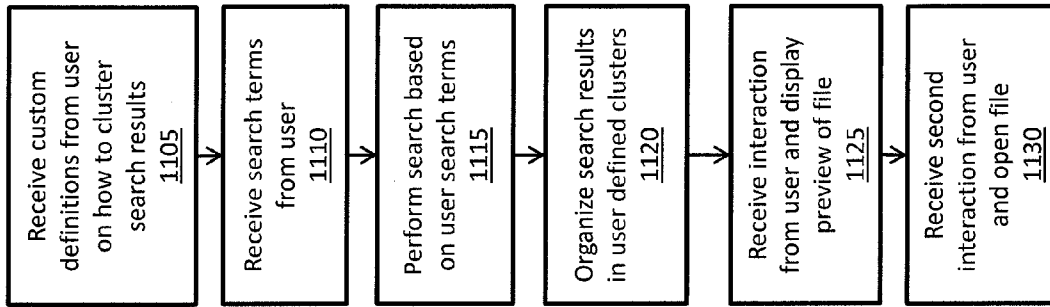


Fig. 10



1000

Fig. 11



1100

## METHOD AND SYSTEM FOR TREE REPRESENTATION OF SEARCH RESULTS

### CROSS-REFERENCE TO RELATED PATENT APPLICATION

[0001] This non-provisional application claims priority to U.S. Provisional Application 61/922,597 filed on Dec. 31, 2013, the disclosure of which is incorporated herein by reference in its entirety.

### BACKGROUND

[0002] Many people today use multiple devices and media in their daily activities. For example, a typical person may use laptops, smart phones, tablets, home and work desktops, external hard drives and memory cards, media centers, smart TVs, cameras, DVRs, cloud storage, as well as occasional use of the devices of family, friends, coworkers, or roommates. Across these various devices, people store large quantities of data of varying importance. Additionally, people receive and store files, programs, messages, and other media in external servers, such as e-mail or cloud computing systems.

### SUMMARY

[0003] An illustrative method according to a set of instructions stored on a memory of a computing device includes receiving, by a processor of the computing device, a search input. The method further includes determining, by the processor, the presence of a plurality of electronic files related to the search input. The method further includes displaying, by the processor, on a graphical user interface (GUI), representative information of the plurality of electronic files related to the search input, where the representative information comprises a hierarchical tree structure.

[0004] An illustrative apparatus includes a memory, a processor coupled to the memory, and a first set of instructions stored on the memory and configured to be executed by the processor. The processor is configured to receive a search input. The processor is further configured to determine the presence of a plurality of electronic files related to the search input. The processor is further configured to display, on a graphical user interface (GUI), representative information of the plurality of electronic files related to the search input, where the representative information comprises a hierarchical tree structure.

[0005] A non-transitory computer readable medium has instructions stored thereon that, upon execution by a computing device, cause the computing device to perform the operations including instructions to receive a search input. The instructions also include instructions to determine the presence of a plurality of electronic files related to the search input. The instructions also include instructions to display, on a graphical user interface (GUI), representative information of the plurality of electronic files related to the search input, where the representative information comprises a hierarchical tree structure.

### BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Illustrative embodiments will hereafter be described with reference to the accompanying drawings.

[0007] FIG. 1 is a flow chart illustrating a method of displaying a tree representation of search results in accordance with an illustrative embodiment.

[0008] FIG. 2 is a flow chart illustrating a method of displaying a tree representation of search results using ranking of search results in accordance with an illustrative embodiment.

[0009] FIG. 3 is a block diagram illustrating a computer system that may be used in accordance with an illustrative embodiment.

[0010] FIG. 4 is a representation of a graphical user interface (GUI) demonstrating a tree representation of search results in accordance with an illustrative embodiment.

[0011] FIG. 5 is a flow chart illustrating a method of displaying a tree representation of search results across a multiplicity of devices in accordance with an illustrative embodiment.

[0012] FIG. 6 is a representation of a GUI demonstrating a hierarchical tree structure search results in accordance with an illustrative embodiment.

[0013] FIG. 7 is a block diagram illustrating various computing and electronic storage devices that may be used in accordance with an illustrative embodiment.

[0014] FIG. 8 is a flow diagram illustrating a method of performing a cross-platform search in accordance with an illustrative embodiment.

[0015] FIG. 9 is a flow diagram illustrating a method of grouping search results into clusters in accordance with an illustrative embodiment.

[0016] FIG. 10 is a flow diagram illustrating a method of displaying search results in a hierarchical tree structure in accordance with an illustrative embodiment.

[0017] FIG. 11 is a flow diagram illustrating a method of customizing search result groupings according to user defined groups in accordance with an illustrative embodiment.

### DETAILED DESCRIPTION

[0018] Described herein are methods and systems for displaying on a graphical user interface (GUI) search results in a hierarchical tree structure or tree representation. A hierarchical tree structure organizes search results in a hierarchy to make viewing and navigating through search results easier for a user. A hierarchical tree structure can be displayed using folders and sub-folders. For example, a folder may represent an electronic device where various electronic files are stored. A hierarchical tree structure can also display sub-folders that are located on the electronic device. Examples of sub-folders may be "My Documents," "My Music," "Photos from my vacation to Monaco," and countless other possibilities. Sub-folders themselves may also contain additional sub-folders for organizing electronic files. There is no limit as to how many levels of sub-folders may exist or be displayed on the hierarchical tree structure. In an alternative embodiment, even though additional sub-folders may exist on an electronic device, the hierarchical tree structure may only display a predetermined number of levels of sub-folders. For example, a hierarchical tree structure may only show a predetermined number of six levels of sub-folders. In this embodiment, if an electronic device only shows six levels of sub-folders, a selected sixth level will cause the display of all of the files (or a representation of the files) that are in the sixth level or any of the sixth level's sub-folders. Sub-folders may also be subordinate to their higher level folders or sub-folders. That is, any action taken with respect to a higher level folder will also be applied to any sub-folders of that higher level folder.

**[0019]** Search results displayed in a hierarchical tree structure can be generated using cross-platform searching of multiple information sources and devices. The methods and systems disclosed herein allow a user to perform a single search across multiple different devices or a single device to search for different types of data. Such data may include messages, e-mails, documents, presentations, photos, videos, attachments to e-mails, contact information or other data types. This different data may reside on several different electronic devices, such as a tablet, mobile smart phone, laptop computer, desktop computer, digital camera, cloud storage system, smart television device, portable digital media player, wearable electronic device, or other types of electronic devices that are capable of storing electronic files and data. Furthermore, the search may be performed across the various devices regardless of the platforms and operating systems that the electronic devices utilize. For example, various electronic devices may use various operating systems like Microsoft Windows™, iPhone™ Operating System (iOS™), Android™, Unix™, Linux™, Mac OS X™, OS X™, Symbian™ OS, Chrome™ OS, Chromium™ OS, BlackBerry™ OS, or other operating systems.

**[0020]** A hierarchical tree structure as disclosed herein can also be manipulated to expand and contract based on a user's inputs. For example, folders and sub-folders may be collapsible and expandable to show or hide the contents of a folder or sub-folder. In other words, the hierarchical tree structure can be manipulated to show more than one level of the hierarchical tree structure. Folders and/or sub-folders may also be selected in order to open or preview the contents of the folders and/or sub-folders. Similarly, if an electronic file is represented on a GUI, it too can be selected to show a preview of the electronic file itself. By receiving and reacting to such manipulation commands, the system can display only parts of the search results that a user deems most relevant or interesting without over whelming the user with a long list of search results.

**[0021]** After an initial search is made, the system may only display a certain number of folders and sub-folders. That is, the system automatically shows search results at a high level. After the initial display of search results, it may be up to the user to use manipulation commands to expand folders and sub-folders, display previews of electronic files in the search results, and open electronic files. A system may determine what folders and sub-folders to initially display after a search in a variety of ways. For example, if a user is only searching a single electronic device, the system may display several sub-folders that exist on the memory of the electronic device. In another example, if a user is searching several electronic devices, the system may only initially display one folder for each electronic device and corresponding to each electronic device. The user can then use manipulation commands to expand the folder for an electronic device to find out more about the search results located thereon. In another embodiment, the folders and/or sub-folders displayed as search results may relate to a virtual location as opposed to a physical storage location such as a particular electronic device. For example, the system may display a folder in the hierarchical tree structure that contains all spreadsheets. In this example, the spreadsheets in the spreadsheets folder may be actually stored on any number of electronic devices, but are organized into a virtual location for all spreadsheets to make it easier to find spreadsheets. In other words, a virtual location folder could relate to types of electronic files. In some embodiments,

a hierarchical tree structure may include both physical location folders and virtual folders.

**[0022]** In other embodiments, virtual folders may be related to things other than file type. Virtual folders may be user defined, such as "Favorite Pictures." Such a user defined folder may include photos from a variety of electronic devices, social network websites, etc. In other words, a virtual folder may not actually contain files, but may instead merely contain location information about the files in the virtual folder such as links, routing instructions, address, folder location, or the like so that the system can find the actual file in the actual physical location from the location information. In another embodiment, virtual folders can be automatically generated. Automatically generated virtual folders may include, for example, a "Recently Opened" folder, a "Top 10 Most Used in the Last Month" folder, etc. In another embodiment, the system may display virtual folders related to data types or other factors. For example, virtual folders for presentations, word processing documents, movies, and pictures may all exist. In another example, virtual folders that relate to a date files were created, last modified, last opened, etc. may all exist and be displayed in search results, either automatically or as designated by a user.

**[0023]** Other folders and/or sub-folders may be displayed automatically by the system because the folder and/or sub-folder or a search result within the folder and/or sub-folder is particularly relevant or high ranked. A folder and/or file may be highly ranked if it has high relevance to the search terms or search input. A folder and/or file may be highly ranked if the quantity of hits for the search terms is higher than other folders/files, or for a folder the number of files in the folder/sub-folder that are relevant to the search input is high relative to other folders. A folder and/or file may be highly ranked if it is stored on a device with a positive access pattern. In other words, if a user over time has a demonstrated preference for data stored on a particular electronic device, the system may highlight search results from the device by showing more sub-folders with search results from that device than other devices. A ranking based on where a file or folder is stored may also relate to an availability or accessibility of the file or folder. For example, the system may determine if an indexed file or folder on a tablet device is actually accessible (for example connected to a wireless network) before highlighting and ranking the file or folder high. As a result, the system can display and prioritize search results that are actually accessible at the time the search is performed. Another way a folder and/or file may be ranked is by a frequency of use metric. That is, if a file or folder is more often used by a user, the system may display that automatically because the chances are higher the user is seeking it again.

**[0024]** File access patterns may also be used to rank files and/or folders. The system may have statistical data on file types that the user opens or uses more often. Accordingly, those file types or folders containing those file types may be ranked higher when ranking search results. For example, a particular user may have a fondness for music and listen to it almost constantly. Accordingly, if the user performs a search, the system may rank music and audio files (or folders where such files are contained) higher than those that are not audio files (or contain audio files). In another example, the system may record and use statistical file access patterns that are device specific. For example, the system may note that on a user's smart phone, the user most often sends SMS messages and has telephone conversations with her friends. The system

may also note that when the user writes and checks e-mails, the user most often uses a tablet device and a laptop device. Accordingly, if a user enters a search input of “e-mail confirmation for flight,” the system may rank results from the tablet and laptop devices higher than any results from the smart phone device. In a simpler example, a system may note that a user spends more time on their smart phone than any other device, thus a higher ranking may be assigned to results from the smart phone based on a device-based frequency of use (as opposed to a frequency of use of files and/or folders). In another example, a user may designate rankings, preferences, etc. for how search results should be prioritized and displayed.

**[0025]** In another embodiment, data to or from a contact (such as a message) can be also be assigned a rank of importance. The rank of importance could be represented by a number, such as a number from 1 to 100. The rank of importance could also be a high, medium, or low categorization. Other ways of ranking may also be used. The user may change the ranking system manually, and may also change the rank of particular data manually. The ranks of importance for data may be used to determine how to visualize the incoming messages and e-mails from different contacts. For example, messages from higher ranked contacts or related to higher ranked conversations may be displayed by the system more prominently than other messages. In one example, the system may automatically rank correspondence and contacts by how frequently they communicate with the user. By analyzing the data in social networking channels, e-mail addresses, message structures, and/or signatures in messages, the system may extract a contact’s possible past and current places of employment. In this way, the system may build an ontology of the people and companies with whom the user has been corresponding and contacting. Such factors of files and messages may be designated as indicating a higher relevance or ranking for search results.

**[0026]** After a system has performed the search based on a search input, the system displays the hierarchical tree structure on the GUI. Next to each folder and/or sub-folder displayed, the system displays a number of how many search results are present in the folder or sub-folder. If the folder or sub-folder does not have any search results, no number is displayed. In an alternative embodiment, a number (or other alphanumeric sequence) may refer to a relevance or ranking of search results contained within the folder or sub-folder. In an embodiment where files are also shown in the hierarchical tree structure, the number may be displayed next to the name of the file to directly refer to a relevance or ranking of that file. In this embodiment, folders and sub-folders that do not have any relevant search results are displayed differently than folders and sub-folders that contain search results. For example, icons and folder names without search results may be a different color, shade, font, size, etc. than those with search results. Additionally, a different icon or folder name itself may be used to indicate that it has no search results contained therein. In this way, a user can easily visually differentiate between folders with and without search results. In an alternative embodiment, the system may omit and not display folders or sub-folders that do not contain any search results.

**[0027]** Search results can be grouped into clusters in some embodiments. Clusters may be according to various factors such as time, file type, size, etc. Clustering into groups can occur in various ways. A system may create or already have established virtual folders directed to certain clusters, such as

file types. Accordingly, when a search is performed the system can display results according to those clusters using the virtual folders. In one example, the system may create new virtual folders as a result of the search in order to display the clusters. Advantageously, the clusters (and formed virtual folders) can then be utilized in the future, for example for future searches. In a related example, a virtual folder related to file clustering may already exist, but the folder may not contain certain of the search results returned in a search that are returned and relevant to the virtual folder. In such an embodiment, the system may add those relevant search results to the virtual folder. In one example, clusters may be shown as search results based on file type. If a user enters a manipulation command to expand a cluster, the system may show other actual physical and/or virtual sub-folders. For example, if the user selects a “Documents” cluster and expands it, sub-folders representing different electronic devices may then be displayed. In this way, the user can then have access to documents that are relevant to the search input and organized by device. Accordingly, if a user was looking for a document that he thought might be on his laptop or desktop computer, such a document may be easier to locate in this embodiment.

**[0028]** In a different example, the system may cluster search results based on various criteria without using virtual folders. In this embodiment, a temporary cluster (or something akin to a temporary virtual folder) is created after a search. For example, a user may enter the search term “Vacation.” The system may automatically cluster search results, or may cluster results for electronic devices or folders that exceed a predetermined threshold of results. That is, if a folder or device has a lot of results, the results may be clustered. In other embodiments, results may always be clustered. In this example, the system may identify a folder on the user’s laptop called “Vacation to Chile.” The folder may have numerous files in it and in sub-folders within it. The system may cluster the files in the folder automatically, or may do so if the folder is expanded by a manipulation command from the user. In this example, the system may cluster the files in the “Vacation to Chile” into various clusters, such as “Photos,” “Planning and Visa Documents,” “Flight, Hotel, and Rental Car Reservation Documents,” “Videos,” and “Correspondence With Lisa re: Chile Trip.” After the system has clustered the contents of the folder, the system may display the clusters on the GUI for the user, similar to how sub-folders would be displayed. Such clustering and temporary display of results may help the user locate the object of her search more easily and quickly.

**[0029]** A user may perform a search input by entering text regarding the subject of their search. The system may receive the search input from the user and perform the search by looking for electronic files related to the search input throughout several electronic storage locations. As noted above, the electronic storage locations may several different electronic devices and/or types of electronic devices. Additionally, the devices may be in different locations or physically remote from each other. As long as the devices can be communicated with by the device doing the searching, the searched devices can be anywhere. For example, the devices may be connected to the internet and the searching device may perform the search by accessing the memory of the other devices through the internet. Through these communications, the system can determine whether electronic files matching the search inputs are located on any of the searched electronic devices.

**[0030]** In another illustrative embodiment, a user's social media networks may also be searched for electronic files relating to a search. For example, a social networking website may allow or utilize file sharing. Accordingly, the system may search the electronic files a user has access too on social networks when performing a search. The system may also be able to search for contacts from the relationships a user has on a social networking website. Additionally, the system may be able to locate information related to users on a social network that the user does not have a relationship with. For example, a user may be trying to locate the contact information of a person who has previously helped the user do landscaping and yard work. By entering the name of the individual, the system may be able to locate the individual and an e-mail address listed on a social media profile for the individual. Such searching and results may take place irrespective of a formal relationship between the user and the individual on the social media website. However, such searches may be limited by the amount of information that can be accessed and searched for by the user. So in this respect, if the user has an official relationship with someone over a social networking website, such a user may accordingly have more information accessible to the user based on the official relationship. As such, more information may be available to the system for searching and search results.

**[0031]** If the system does locate relevant electronic files, the system can display the results to a user on a graphical user interface (GUI). The results may be displayed using representative information of an electronic file. That is, the entire electronic file may not be displayed on the GUI, but rather certain information relating to a relevant file may be displayed. Representative information may also include a preview of the electronic file. Representative information could also include any sort of metadata about the electronic file, such as the physical location where the electronic file is stored, the size of the electronic file, a date and/or time the electronic file was created or last modified, a file type of the electronic file, a preview of the electronic file, or any other information relating to the electronic file.

**[0032]** In some embodiments, representative information of particular electronic files may be displayed in the hierarchical tree structure search results. In other embodiments, the hierarchical tree structure may only display folders, sub-folders, and/or clusters and individual files, representative information of files, and previews of files may be displayed in a different part of the GUI or a different window on the GUI altogether. In some embodiments, a preview of one or more of the electronic files may be automatically displayed on the GUI with the search results. In another embodiment, the user may position a mouse or pointer over representative information of an electronic file to get a preview to appear. In another embodiment, the user may have to click on the representative information or otherwise interact with it through the GUI (such as touching the representative information through a touchscreen) to get a preview of an electronic file to appear.

**[0033]** In another embodiment, the system may display on the GUI search results in a way that does not immediately display information directly related to individual electronic files. Instead, the system may display the information in representative groups (also referred to herein as clusters or an amalgamation). The system may display groups of search results based on the various characteristics or traits of the results. For example, electronic files that are found as part of the search results may be sorted according to data type. For

example, the results may be sorted and grouped into clusters based on whether each result is a document, presentation, photo or other image, video, e-mail, contact information, message, or other sort of file or data type. The clusters could also contain more than one data type in the same cluster. A cluster could also represent all search results from a particular storage location. A cluster could also represent search results from a certain period of time (when the files were modified or created, for example). The clusters may be automatically determined by the system, or may be configured and defined by the user.

**[0034]** A group or cluster displayed on the GUI may also indicate how many electronic files of each type of files were found during the searching. In another illustrative embodiment, the system may display certain groups or clusters more prominently than others. For example, the system may automatically display a cluster representing documents that were saved on a hard drive more prominently than a cluster representing documents that were attached to e-mail messages. Although the system may automatically display certain groups and/or clusters with more prominence, the system may also be configured by the user to show certain groups or clusters more prominently than others.

**[0035]** A user may also expand a cluster to view representative information of the files that are represented in the cluster. The user may accomplish this through an interaction on the GUI with a particular cluster or group. Representative information of individual files within a cluster or group may also be displayed more prominently than others depending on system defaults or preferences established by the user.

**[0036]** In one illustrative embodiment, the clusters or groups displayed by the system may be displayed as a hierarchical tree structure. In a hierarchical tree structure, results may be shown in groups that are expandable and collapsible to demonstrate more detail regarding the search results. The system may also indicate in a hierarchical tree structure, how many results regarding each level of the hierarchical tree structure there are. For example, a search results may be displayed in clusters as a hierarchical tree structure. The levels initially displayed in the hierarchical tree structure may sort the search results according to a first criteria. For example, the search results may be sorted into clusters depending on a data type. In other words, the search results are sorted based on what type of electronic file each file is. Each level may also display how many relevant files are represented by the cluster or group.

**[0037]** Upon interacting with one of the levels with, for example, a touch screen selection or pointer point and click operation, a cluster or group is expanded on the GUI. In one embodiment, the expanded level may display representative information of each electronic file represented by the cluster or group. In one embodiment, the system may determine to do this where the number of files represented by a cluster are relatively low (for example, five electronic files or fewer). In other embodiments, further sub-levels of the hierarchical tree structure that represent sub-groups of the clusters or groups may be displayed. For example, if a cluster representing e-mail messages is interacted with, further sub-groups sorted by a second data type may be displayed on the GUI. For example, the sub-groups may be sorted by date, such as month and year the e-mail messages were received. In this way, other sub-groups may also exist, so that a set of search

results is organized viewable based on several different categorizations based on data type, metadata, file location, and the like.

**[0038]** In order to determine relevant search results, various types of searching may be performed by the system. For example, morphological searching, metadata searching, tag searching, full text searching, fuzzy searching, or ontology-based semantic searching may all be used, alone or in different combinations, in various illustrative embodiments. For example, morphological searching may automatically utilize searching for words or forms of words that are similar to the words entered as a search term by the user. In another example, the system may utilize metadata and tag searching. In this embodiment, the system can search information that is associated with an electronic file as metadata or tags for information relevant to the search terms entered by the user. In full text searching, the system may search the entire contents of an electronic file for relevance to the search terms entered by the user.

**[0039]** Fuzzy searching, also known as approximate string matching, may also be utilized by the system. Fuzzy searching takes a string of characters entered by the user as a search term and searches for a similar string of characters that approximately match the string of characters entered. In this way, if a search term that is entered has been misspelled, the system may be able to automatically search for the correctly spelled word rather than the misspelled word.

**[0040]** In ontology-based semantic searching, the system may utilize context and interrelated information from electronic files to perform searches. For example, system may examine the content of electronic files on the system and utilize the examination to determine certain relationships or contexts. For example, an e-mail may contain a signature line with the name of a company that the sender works for. As a result, the system could return results for the company broadly during a search where the search terms entered was the name of the sender himself.

**[0041]** In another illustrative embodiment, a user may attempt to perform a search when one or more electronic devices of the user is not online. That is, the electronic device performing the searching may not be in communication with certain of the user's electronic devices when a search is performed. In such an embodiment, the search may only locate electronic files that are located on the available and online electronic devices and cloud storage systems. In an alternative embodiment, however, the system may still be able to search for documents that are stored on an offline electronic device.

**[0042]** For example, the system may maintain an index of electronic files on the various devices of the user that includes information such as metadata and tags about files. The system may search this index when a user performs a search rather than search the memory of each electronic device of the user's in the system. This is particularly advantageous where one or more of the electronic devices is offline while the user is attempting to run a search. Where a device is offline during a search, the system may search the physical memory of online devices while searching the index regarding the offline devices. In another embodiment, the system may only search the index, regardless of whether a device is online or not. In yet another embodiment, the system may start with a search of the index, and may use the results of the index search to determine specific electronic files to further search that are located on the various user devices.

**[0043]** In an alternative embodiment, the index may be a full text index. In this embodiment, the full texts of electronic files are stored on the index. In this way, the system may perform searches on all of the electronic files and their content, even if the electronic files themselves are not currently stored on devices that are online. In another alternative embodiment, only some of the total electronic files on a user's electronic devices may be full text indexed. For example, the system may automatically full text index electronic files from electronic devices that are more likely to be offline than others. For example, the system may full text index electronic files from external hard drives and digital cameras, while not full text indexing a desktop computer. In another embodiment, the user may be able to specify which devices' electronic files should be full text indexed or not. In another embodiment, the system may selectively full text index files based on other criteria. For example, the system may not full text index certain electronic files that are over a certain size. In another example, the system may not full text index certain types of electronic files, such as images and videos. The system may be configured to automatically make such determinations, or the user may configure what types of electronic files to full text index and which electronic files not to.

**[0044]** The user may also be able to set up rules regarding full text indexing. This may be advantageous where the user recognizes the importance of the electronic files selected for full text indexing, particularly where the user knows he or she will have a future desire to fully search said electronic files. For example, the user may set a rule that any e-mail messages and files attached therewith from contact John Smith should be full text indexed. The user may set up a rule based on an important relationship. For example, John Smith may be the user's financial advisor. Accordingly, the user wants to ensure that any correspondence or documentation sent to the user from Mr. Smith is well documented and accessible utilizing the system. In another embodiment, the system may determine these types of rules automatically. For example, the system may determine a user's most important and/or frequent contacts. Any correspondence between an important and/or frequent contact may be automatically fully indexed by the system so that the correspondence is easily searchable and locatable.

**[0045]** In an illustrative embodiment, the system can determine the presence of electronic files of a user among multiple devices and assemble data relating to the files into an index. For example, the user may utilize a tablet, smart phone, laptop, universal serial bus (USB) drive, cloud storage, set top boxes (smart TVs), and many other electronic storage and computing devices and methods. Further, software applications may include files that relate to a user or the user has access to. Such files may include e-mails, messages, chats, documents, pictures, videos, contacts, or other electronic data. The system can determine all of these types of data from a plurality of devices and index the data. Even if a user was not the creator or owner of the document, the system can index the data as long as the user has access to it. The index may be stored on a particular device of the user's, for example, a laptop. The user may also designate which device will store the index. In another embodiment, the index will be stored on a cloud storage system. Advantageously, when the index is stored on a cloud storage system (e.g., a remote server), the index may be updated regarding changes to data or files on any device whenever the device is connected to the internet.

**[0046]** Described herein are systems and methods which allow users to disclose, access, distribute, synchronize, manage, and search different types of content and electronic files, including but not limited to file content, communication content, contact content, and other personal information across multiple devices and cloud servers. Also described herein is a graphical user interface (GUI) that displays representative information of different electronic files. The electronic files may be physically stored on multiple devices, but the representative information relating to the electronic files may be displayed on a single GUI. The representative information of an electronic file may include the physical location where the electronic file is stored, the size of the electronic file, a date and/or time the electronic file was created or last modified, a file type of the electronic file, a preview of the electronic file, or any other representative information relating to the electronic file. Furthermore, a user can interact with the GUI to view, open, access, organize, backup, synchronize, distribute, or locate different electronic files stored on various electronic devices. Representative information as defined herein may include, but is not limited to, various types of metadata.

**[0047]** After electronic files such as documents, pictures, e-mails, contact information, and the like have been indexed, representative information of the electronic files can be displayed on a graphical user interface (GUI) to a user. In one embodiment, the representative information may be displayed in a single window format. Representative information may include a file name, a file preview, date last modified, version number, date created, number of related files, number of files in a folder, and any other information relating to an electronic file. In some embodiments, the representative information may include information regarding the location of the electronic file. The location information can be indicated utilizing a symbol or icon. For example, if a file is on the user's smartphone, a smartphone icon may be used to indicate the location of the file. Different icons or symbols may have different shapes, colors, sizes, or text content. For example, the device where a file is stored may be part of the representative information. The system may also index a so-called virtual location of a file. For example, regardless of where an e-mail might be physically stored, it might be virtually located in a user's e-mail inbox within the system and displayed as such whenever the inbox is displayed on the GUI. In this case, the system may indicate either or both of the virtual and physical locations. In some embodiments, the representative information may be displayed in a single window, but may be divided within a multiple tab configuration. For example, a GUI may include tabs for files, photos, e-mails, and contacts. In another embodiment, the GUI may have tabs for files and e-mails, where photos are categorized under files and contacts are categorized under e-mails. The user may be able to switch between tabs by interacting with the GUI. For example, a user may utilize a motion translation device and input, such as a touchscreen in which the user may touch the location on the screen where one of the tabs appears, activating and displaying that tab. To switch to a different tab, the user may touch the different tab. In another example, the user may use a mouse or touchpad to locate a pointer over the desired tab on the GUI. The user may then make another input, such as a single mouse click, to activate and display the desired tab the pointer is located over.

**[0048]** Whether or not tabs are used, the representative information of electronic files displayed by the system may include information relating to where and how the electronic

files are stored. For example, the representative information may include an indication that the file is stored on a particular device or in a cloud storage system. The representative information may also include an indication of whether the file has been backed-up or not. The representative information may also include the location of a backed-up copy of the file. The representative information may also include an indication of whether the device where the file is stored is online or offline. In the case where a file is stored on two different devices, the system may indicate the status of both devices. In another embodiment, the system may only indicate enough about the status of the devices to indicate whether the file is available for opening and/or editing. For example, if a file is on two different devices, and one is online and one is not, the system may indicate on a GUI that the device where the file is stored is online, because the user's actual access to the file is not hindered by one of the devices being offline. Other status's than online and offline may also be indicated by the system. For example, the connections speed or status of a device may be indicated. In another example, a passive or active status may be indicated. Such a status may be relevant if a computing device has a standby, sleep, hibernate, or screen saver mode that restricts remote access to the memory of the device. In another example, instead of online or offline which indicates communication between two devices over a network, a device status may be connected or unconnected. This may be useful for a secure digital (SD) card, portable hard drive, or USB storage device, where the devices are connected directly to a computing device rather than communicating with devices over a network. Other potential statuses that may be indicated by the system about a computing device may include a currently operational status, a currently non-operational status, or a set in a lower functional mode status. Despite the status of a device however, if a file has been previously indexed, the representative information relating to the file can be displayed.

**[0049]** In an illustrative embodiment, the representative information for a file may be manipulated or changed by the user. For example, if the user receives a file through a file sharing website or attached to an e-mail, the file may have a computer generated name or a name that is otherwise undesirable by the user. Accordingly, the user may then change the name of the file that is used in the representative information. Any other of the representative information may also be changed by the user.

**[0050]** The system may also display a preview of electronic files on the GUI. This may occur automatically, or may occur based on some sort of interaction by the user with the GUI. For example, the system may display the preview only if the user clicks or taps on the file once. In another example, a preview may be displayed when the user moves a pointer over a file. In another embodiment, a preview may be automatically displayed, but a further preview may be displayed when a file is interacted with. For example, the first page of a document may be shown, and when a pointer is moved over the document, the second page of the document is shown. In another example, a thumbnail of a picture may be shown. When the user moves a pointer over the thumbnail, a larger and/or more high resolution version of the photo may be displayed by the system. In order to generate the preview of electronic files, the system may extract all or a portion of content from an electronic file to generate the preview. For example, where a first page of a document is used as a preview image, the first page of the document is extracted. Various

types of previews may be displayed by the system. For example, a preview may include a first group of words from a document or message (such as an e-mail). Another preview may be a thumbnail image of a larger image, or a screen shot of a video. The preview image could also include some or all of the contact information relating to an e-mail or other type of message. A preview image could also include a task information relating to an electronic file. For example, where the electronic file is a task list, reminder list, or calendar, such a task list, reminder list, or calendar may include information that can be used in the preview. For example, a reminder list may include an item reminding the user to pay their rent. Accordingly, a preview of the reminder list or of the item on the reminder list itself may include the text "Rent is Due!" In another embodiment, such a task or preview may be derived from an e-mail, such as an e-mail requesting information from the user. The preview of such an e-mail may include text indicating the request.

**[0051]** In another illustrative embodiment, context menus may be used. Context menus are menus that pop-up on the GUI relating to a particular part of the GUI. For example, a context menu may pop-up if a user moves their mouse over the representative information of an electronic file. In another embodiment, the user can move a pointer over the representative information and make another input, such as a mouse click, to get the context menu to display. In other embodiments, such as when a touch screen is in use, the user may only input with regard to the representative information, and a pointer may not be utilized. A context menu may include information related to a particular electronic file or actions that may be taken related to the electronic file. For example, a context menu may indicate that an electronic file is classified as read only. In another example, a context menu may present a user with options relating to the electronic file, such as download the file, delete the file, rename the file, move the file, back-up or copy the file, etc.

**[0052]** In another illustrative embodiment, various user inputs and interactions may be made to effectuate opening an electronic file by clicking on or otherwise interacting with the representative information of a file on the GUI. For example, if the user wishes to open a document stored on a laptop, even though the user is currently using a tablet, the user may activate the system disclosed herein. The user can then navigate to or search for the representative information of the document desired, and tap on it through the touch screen of the tablet. The system receives the input from the user, and the system then uses an application that matches the file type of the document to open the document and display it to the user on the tablet. The application may be located on the tablet or located elsewhere. As long as the system has access to the application, the document can be opened. In an alternative embodiment, the system may select an application to utilize when opening a document based on a number of factors. For example, the system may consider whether the user made any choices regarding what application to use (e.g., an open-with option) or how the file should be displayed (e.g., read-only). The system may also choose an application that is appropriate for the device the user is currently using. The system may also select an alternative application to use where a preferred application is not available. In an alternative embodiment, electronic files that have been opened may be displayed directly by the system and even edited within the GUI. If a document is not available (for example, the device where the document is stored is offline), the system may automatically

find another copy or replica of the document on another device and open that one. In this way, the user may never have to worry about how the document is actually opened. For example, there may be a copy or replica of the document on an external hard drive, a cloud storage server, or the user's home personal computer (PC). In another embodiment, the system may be able to locate a copy or a replica of the document on an e-mail server or otherwise find it among other message types that the user has sent or received.

**[0053]** In another illustrative embodiment, the GUI may display some representative information more prominently than other representative information. For example, if the representative information includes an icon such as a piece of paper type icon for a document. The piece of paper type icon for certain files may be displayed larger and/or further toward the top of the GUI than the icons for other files. Such a difference in prominence may be random, so as to make the display less monotonous and more pleasing to the eye. If groups of files are displayed together in a solitary representative information, different groups of files represented by an icon or symbol may also be displayed more prominently than other groups or individual files. In another embodiment, such differences in prominence may result from a decision made by the system to display the representative information of the files accordingly. For example, the system may determine to display more recently opened or edited files more prominently. In another example, the system may display messages that have been received but not read or responded to more prominently. In another example, the system may display photos or videos posted to the user's social networks that have a relatively greater number of social interactions with others on the social networks more prominently. If the system is displaying search results, the system may display results with more relevance to the search more prominently. Other criteria, such as number of times modified, date created, size of the file, or tag or metadata associated with the file, may also be used by the system to determine how to determine representative information to display more prominently on the GUI.

**[0054]** In another illustrative embodiment, the GUI may also display a representation of a current storage utilization of a device in which the GUI is being displayed. For example, the representation may include a graphical representation indicating a percentage or fixed amount of free storage space and used storage space. In another embodiment, such a representation may be used to indicate an amount of free and used storage space on a different device. In another embodiment, such a representation may be used to indicate an amount of free and used storage space on all of the user's devices combined. In another embodiment, such a representation may be used to indicate an amount of free and used storage space in a portion of a cloud storage space allotted to the user. In yet another embodiment, such a representation may be used to indicate an amount of free and used storage space in all of the user's devices and the portion of the cloud storage space allotted to the user.

**[0055]** In another illustrative embodiment, the representative information displayed on the GUI may include a hierarchical tree structure. Such a tree structure may indicate the various physical devices of the user and the files and folders stored thereon. The hierarchical tree structure may also indicate the files and folders stored on a user's allotted cloud storage space. In addition to showing where files are physically stored on various devices and/or cloud storage space, a hierarchical tree structure may alternatively or additionally

display files in their designated virtual locations. A virtual file location is defined herein as a location the system visualizes a file being in, even if the file is not physically stored there. For example, a virtual location of “My Music” may contain all of a user’s audio files, regardless of what device or folder the audio files are in. Thus, each audio file has a virtual location of the My Music folder. Other files and contacts may similarly have other virtual locations as determined by the system or the user. A virtual file is defined herein as a file that is a merged version of several files. The virtual file may physically exist on a storage device, or may exist locally when opened by the user and incorporates data from multiple physical files. A virtual file may be displayed as representative information on the GUI. The representative information may indicate that it is a virtual file, or the representative information may appear similar to the representative information for any other file, not indicating to the user that the file is a virtual file.

**[0056]** Electronic files indexed, displayed, used, etc. by the system can include many things. For example, files as defined herein may include things such as documents, presentations, spreadsheets, e-mails, messages, audio files, video files, contact information, web pages, information from webpages, tasks, calendars, and the like. Such files may be gleaned by the system from various sources such as local file storages powered with operating systems (e.g., laptops, phones, tablets, media centers, etc.), external passive file storages (e.g., external hard drives, memory sticks, CD/DVD’s, etc.), cloud storages (e.g., Dropbox™, Google Drive™, Evernote™, Flickr™, Facebook™ galleries, etc.), e-mail accounts (e.g., Gmail™, Hotmail™, Yahoo™ mail, exchange servers, etc.), messenger services (e.g., Twitter™, Skype™, SMS, Snapchat™, Viber™, Google Hangouts™, etc.), social media feeds (e.g., Facebook™, Google+™, LinkedIn™, etc.), contact lists (from e.g., e-mail accounts, messengers, telephone, personal address books, Outlook™ contacts, etc.), tasks, calendars, dictionaries, encyclopedias, internet sites, and any other services like FineReader™ online.

**[0057]** Advantageously, the systems and methods disclosed herein provide benefits over an exclusive utilization of cloud storage. Some people store information in a cloud storage in order to synchronize information on different personal devices. However, for other people, cloud storage is not necessary because personal devices have become very productive, and often have sufficient capabilities to handle the needs of many users without resorting to cloud storage strategies. In other cases, the amount of data produced by some people is so massive that transferring that data to cloud servers may be cumbersome and time consuming. For example, a digital camera owner may return from a vacation with hundreds, if not thousands, of photos or videos in high resolution. If these photos and videos were moved to cloud storage, the photos and videos could occupy multiple gigabytes of memory in the cloud storage servers, and may additionally severely tax data transfer resources to move all the photos and videos to a cloud server. Further, many devices have become wirelessly connected to each other (e.g., smart TVs, media centers, cameras, fitness bracelets, Google™-glasses, other wearable computers, etc.) so that it may be preferable to keep some data on local devices rather than remote servers. Additionally, much of the data that could be stored on a cloud storage may never be used again, thus unnecessarily taking up space on the cloud storage servers.

**[0058]** Accordingly, there may not be sufficient resources and/or desire to move vast quantities of generated data from

multiple devices to cloud storage and organize all of the data. Additionally, cloud storage systems or leaving data on multiple devices may not be satisfactorily searchable to make such methods for file storage useful.

**[0059]** The systems and methods discussed herein provide for a multiple device storage system where files can be organized across all personal devices, cloud locations, and applications (e.g., e-mail, word processing, videos, pictures, etc.). Further, people share computer files sending them as e-mail attachments or attaching them to instant messages, social media feeds, blogs, or publishing them in their galleries or cloud storages. Accordingly, the systems and methods herein provide for organization and searching not only across all personal devices and cloud locations but also through a user’s e-mails, instant messages, social media feeds, blogs, media galleries, and other communication content.

**[0060]** In one illustrative embodiment, a system can display, access, distribute, synchronize, manage, and search different types of content, including but not limited to file content, communication content, contact content and other personal information across multiple devices and cloud servers. The system may also include a server system that communicatively couples to an agent-type application on one or more client devices.

**[0061]** In one illustrative embodiment, a user may be able to perform a customizable search. For example, a user may wish to only search for a certain type of electronic files or content, such as contact information files. In this example, the user is able to specify that contact information files should be searched. In another example, the user may wish to only search among spreadsheet type electronic files. In another embodiment, the user may specify that he or she would only like to search among electronic files that are stored on particular electronic devices. For example, a user may specify that he or she would only like to search for electronic files on the user’s mobile phone and cloud storage system. In other examples, these functionalities may be combined and mixed by the user. For example, the user may specify in the system that only videos from the desktop computer are the subject of a search.

**[0062]** In another illustrative embodiment, a method provides multi-platform content access, distribution, and synchronization of files on different devices including using a client-side application to manage transfer and presentation of files on a device and across server systems.

**[0063]** In another illustrative embodiment, a method provides multi-platform content access including full-text indexing of the content, files, and metadata of files located on a device and searching of files using full text index located on a server system.

**[0064]** In another embodiment, a method is disclosed for displaying data from local, external, offline, and other sources in a single interface presented to a user. The display can also include information on where the data and/or files are actually stored.

**[0065]** In another embodiment, a computer-readable medium may include instructions executable by a processor to cause the processor to perform functions allowing multi-platform content access, distribution, and synchronization on different devices. Such instructions may be embodied in a client-side computer organization application. In addition to the foregoing embodiments, other systems and computer program product embodiments are provided and supply related advantages.

**[0066]** The number of devices that a user may implement in her day-to-day life has proliferated. While these individual devices are often-times interconnected via, for example, wireless networking, cellular networking, and other infrastructures, systems and methods as disclosed herein can effectively manage data shared between such devices. For example, the systems and methods disclosed herein address platform content access, distribution, and synchronization of data and files, while managing the data and files as well.

**[0067]** The systems and methods disclosed herein better effectuate such platform content access, distribution, and synchronization, while at the same time managing the data itself so as to make more effective use of available cloud-based solutions. Finally, the systems and methods disclosed herein provide for utilizing multiple, yet interconnected, devices while making file organization over such devices and systems easy for a user.

**[0068]** The embodiments disclosed herein provide a convenient one-stop (unified) access to multiple forms of personal data. The embodiments facilitate a unified catalog of personal files across all devices, as well as an ability to view or play files locally or remotely.

**[0069]** In addition, the embodiments disclosed herein provide file and/or data management functionality, such as opening, copying, moving, deleting, archiving, sharing, and editing. The disclosed embodiments promote technology to automatically smart move files in a user's family cloud, so that there is always space on the user's phone and laptop, and the most necessary files are always at hand.

**[0070]** The embodiments disclosed herein also provide search functionality. Advantageously, the search functionalities disclosed herein expand searching beyond file name searching. This may be helpful where, for example, files arrive in e-mail or are shared online or by other means and have a computer generated file name. Accordingly, the embodiments disclosed herein can examine the metadata for full-text content utilizing a fuzzy search, within the meaning according to different ontologies, etc.

**[0071]** In an illustrative embodiment, the system searches not only electronic files on the user's electronic devices and cloud storage systems, but also any electronic files that are shared with the user through file sharing websites and other mediums. For example, some cloud storage systems facilitate the sharing of electronic files amongst its user base. The system in this embodiment can seamlessly search such shared documents, whether or not they were authored by the user, as part of other searches for electronic files. That is, the user may not even notice that his or her searches are including shared documents. Of course, in an alternative embodiment, the system may indicate in the search results that certain results are shared documents. In another additional embodiment, the user may be able to specify that a shared document search be performed. In another embodiment, someone may share a contact list with the user. This may be through an e-mail exchange or other application. In this case, the shared contact list may also be searchable by the system. Other types of shared electronic files may also be searchable like the other examples given here.

**[0072]** The embodiments disclosed herein also provide privacy/security functionality. In some embodiments, a repository of data will not be stored in the cloud for users concerned about privacy and security issues. There may be temporary or compartmentalized storage for proxy caching, data manipulation, or convenience, but significant duplication or long-

term storage in the cloud can be avoided. Additionally, some embodiments may include no indexing or cataloging of data in the cloud to ensure data protection, theft protection, corruption, integrity of data, and related issues with cloud based repositories. Further, some embodiments may keep no file catalogue or full theft index in the cloud. Further still, the architecture envisioned can either utilize or not utilize a mirrored or master copy of the index/catalogue data in the cloud. In some embodiments, a mirrored or master copy of the index/catalogue data in the cloud may be stored and indexed locally on the client side to facilitate local control of the index. In some cases, such an arrangement may be considered a peer-to-peer index and data synchronization process.

**[0073]** As will be further described, the embodiments disclosed herein provide search and data management functionality over an entire available range of data storage mediums and systems of exchange between devices. For example, various embodiments disclosed herein promote (a) disclosure, (b) manage, and (c) search information functionality for (1) locally stored files, (2) offline files, (3) cloud-based files, (4) e-mail information (whether stored locally, offline, or remotely), (5) instant message/short message format information (again stored in a variety of locations), (6) social media feeds, (7) contact information, (8) other type of content including but not limited to dictionaries and encyclopedias, tasks, calendars, etc.

**[0074]** An illustrative embodiment includes a distributed platform and a client application that allows users to have very easy access to their files and e-mail. The embodiment brings together all user information sources available in the user's network, including local and cloud sources, external devices, social networks, e-mail, and telephone communications. Furthermore, the embodiment allows for data management, which may be configured to be seamless and automatic. For example, one of the user's devices may be running out of available storage space on a particular data medium. The system may make the decision to migrate data in order to free space, or perform backup operations.

**[0075]** Another illustrative embodiment includes a platform based on a back-end and a front-end, native client and external device architecture. Any product/service that generates data and files (documents or messages) can integrate with this platform as a source of data and files. Accordingly, the user can see these data and files in a uniform interface with an ability to search, tag, preview, manage, share, etc. The platform may include built-in smart rules and notification functionality. For example, a particular device may be wearable, such as a fitness bracelet. This device, through the embodiments disclosed herein, may trigger a rule or notification based on the activity of the user to provide an alert. Moreover, embodiments as disclosed herein may include infrastructure through which data may be displayed. For example, a user may again be wearing a device, through which the infrastructure facilitates the viewing of a video message or otherwise displays data.

**[0076]** In an illustrative embodiment and from the point of view of architecture—agent software is installed under Operating System (OS) software. The agent extracts text and metadata from all local files and e-mail and passes on the data to a server or other local or remote processing device (herein referred to as cloud service) for indexing. The agent monitors changes in files. The agent collects the actual aggregate of all the devices of the user code and creates a local copy of the relevant index (cache).

**[0077]** In an illustrative embodiment and from the point of view of the user interface, the cloud service provides a combination of a simple, but very dynamic, attractive, and convenient e-mail client and a simple, but very comfortable file manager. Such a program may function in ways used in a traditional e-mail client (e.g., Microsoft™ Outlook™), but also may include the ability to view not only an e-mail message but also files included in correspondence. This program, a client tied to the cloud service, or cloud service client, may be implemented for desktop operating systems (OSX™ and Windows™), and for mobile operating systems, such as iOS™ and Android™.

**[0078]** In one illustrative embodiment, the cloud service client interface for desktop OS is very fresh and light in the spirit of interfaces for mobile OS. This program is not intended to replace traditional e-mail clients or file managers, but aims to make working with mail and files easier. For example, a first type of functionality may serve to review incoming mail. A system may provide a radically more convenient way of presenting the incoming mail than traditional systems, such as Outlook™ and Mac Mail. Such a presentation may be based on the fact that important messages are displayed in a different way than un-important messages. Another feature of such a system may relate to the formation of responses to messages directly within the cloud service client.

**[0079]** The systems and methods disclosed herein incorporate an innovative end-to-end search (wider than a typical user search query) for all mail, contacts, devices, and locations where user information may be stored. In such a search functionality, precision and recall may be superior due to the following example implementations. Searching as disclosed herein may be performed to generate a dossier. Such searches may be performed automatically by the system, or the system may create dossiers based on user searches and search terms. A search may be based on natural language and meaningful information may be extracted from the email and metadata. Further, the search may be based on the involvement of background information, which does not exist in the actual desired letter or document, but may be extracted from the previous history of correspondence, from external sources (e.g., Facebook™, LinkedIn™, Wikipedia™, etc.), or from a dossier—a company's ontology, geographies, etc. which are built during the historical correspondence.

**[0080]** For example, a user may have forgotten the name of an architect with whom she is in sporadic contact. Using systems and methods disclosed herein, the user may make a search query: "looking for a sample wall finishing, which a designer sent to me." An example system may find such a document even if the picture has been sent as a file attachment with the name P7272812.jpg (a title with no obvious meaning) and the letter in which the file was received contained a single word "Stucco" and no signature. Sender is indicated as "a5067834@hotmail.com" (again an ambiguous title). The system may identify that a5067834@hotmail.com—is associated with a "designer," because in a previous correspondence they discussed drawings and floor plans.

**[0081]** Search terms can also be extracted from common knowledge (synonymous/homonymous narrowing/broadening of the search). In other embodiments, fuzzy search, phonetic search, and/or search with the transliteration may be used. Indexed links that are sent in a message body may also be included. Sometimes in traditional systems, a user may not find a desired file or set of files because the user thinks that

they sent files as email attachments, when the files were actually sent as a link to a file sharing site like Dropbox™. As another example, when a user is looking for a New York Times™ article, they may not find it because the article came as a reference (e.g., an HTML address or remote storage server address, etc.). However, the current systems and methods could locate the link based on context. Further, a search may be run on letters and attachments, and also through the files on all of a user's devices.

**[0082]** In further embodiments, a tag classifier's probability may be taken into account when ranking search results. If an auto-tagging system is not sure about a classifier, the probability for the classifier is indicated. Search results may also contain statistical information on directories. When switching to directories, selection in search results may be narrowed (or expanded). Search results may contain statistical information on tags. Search results may also be visualized to the user in a hierarchical tree structure, for example, where there is information on how many files are relevant to the search query, and the associated folder for those files.

**[0083]** Often, in order for a user to search through photos, the user might first perform tedious manual tagging operations, such as tagging names of his relatives and friends on faces and objects (e.g., sea, beach, house, tree, etc.) so that those faces and objects will be recognized and associated with the remainder of the user's photos and tagged in the future. In many cases, the user may not have time to perform this manual tagging. The systems and methods disclosed herein alleviate the burden of manual tagging in a number of ways. The system performs "social tagging" automatically. For example, when a user is posting some of his photos in social networks, other people are tagging the faces, objects, and places on the photos and commenting on the photos themselves. The system can track these tags and comments and use them to create metadata for the particular photos which have already been posted, and use these tags and comments for initial tagging; after which the same people, objects, and places on other photos which have not yet been posted are recognized and categorized. Accordingly, the system (without requiring initial user tagging) builds a database of searchable metadata for both photos which have previously been posted in social networks and those which have not yet been posted.

**[0084]** In an illustrative embodiment, a user may have received an email from "Pete." The user may wish to find the email he recollects, however, he is unsure from which account or location the email may have originated. The system addresses this issue by merging all available information in the user's network, including emails, social accounts, and other related information. For example, dossiers may be used to merge information. For every contact, a unique ID may be initiated that corresponds to the contact in the system. The system parses through all of the available information, and performs such merging operations on predefined rules, such as names similarity, email similarity and other information. Following this merger, the system may then query the user to verify that each of the instances found in the user's network should be associated with the unique ID. Optionally, the user may be able to specify differing ID's for portions of the identified information. Such predefined rules may vary according to the information being parsed. Such functionalities may be applied in a variety of scenarios. In addition, the system may utilize additional information such as logos, product descriptions, employee information, and the like to

perform merger operations. In the case of project tags, the system may group e-mail communications, analyze the time that the communications were received/sent, and other information to organize information based on a particular project tag. A unique ID and/or project tags may be associated with a particular dossier for a particular data unit, theme, or ontological object.

**[0085]** In an illustrative embodiment, search results may also be grouped together, particularly as they are displayed on a GUI. For example, search results may be grouped together by type of electronic file, date of creation, date of last modification, identity of author, identity of last modifier, and/or other various criteria. A cluster or group may also be organized based on the relative calculated ranks of the electronic files. The groups or clusters of search results may be displayed as groups on the GUI and as will be further disclosed herein. When clusters of search results are displayed, little information about the individual electronic files may be displayed in the cluster. For example, a cluster may be based on the year in which the electronic files were created. In this example, the year an number of search results represented by the cluster or group may be displayed. The user may interact with the cluster or group that is displayed, and the system will then display more information related to the search results in that cluster. Such information may include representative information and the information may be presented based on ranks as disclosed herein.

**[0086]** In one embodiment where clusters and groups are used to display search results, the system may display a general cluster, like the one described previously with regard to files that were created in the same year. When the user interacts with the cluster to display more information relating to the files represented by the cluster, ranks are used to determine how that information is displayed. Electronic files that have higher ranks may have representative information regarding the files displayed individually, so that the user can view details about each of the most highly ranked files. For lower ranked files, little or no representative information may be displayed initially. Instead, the system may display a sub-cluster or subgroup that represents several of the lower ranked files. The user may further interact with the sub-cluster or subgroup to display representative information relating to the lower ranked files.

**[0087]** In another illustrative embodiment, the system may assign higher ranks to electronic files that have previously been designated as important by the user. For example, if a user has backed up a particular file by copying it to a cloud storage device, that file may be considered more important to the user, and thus would be ranked higher, than a file that has never been backed up by the user. Similarly, a file that has been moved by a user from one device to another may be considered either more or less important to a user for ranking purposes. In another example, if a file has ever been attached to an e-mail or other message, the system may consider it more important to a user than a file that has never been attached to a message.

**[0088]** The ranking system disclosed herein may also be utilized to determine an optimal storage strategy for electronic files. For example, if a particular device has a memory that is full or has reached a certain threshold, the system may determine particular files on that device that should be relocated. The system may use the ranking system to determine which files to relocate. In one embodiment, the system may

relocate file associated with relatively high rankings. In another embodiment, the system may relocate files associated with relatively low rankings.

**[0089]** Herein, use of the terminology contact may refer to an individual or an organization. Herein, an organization may be an ontological object. Dossiers as disclosed herein may be created for ontological objects. In one embodiment, the methods and system disclosed herein perform ranking operations to automatically rank contacts. This ranking may be performed using a variety of analyses. A ranking may be calculated on a semantic analysis of the user's correspondence with the contact, which may take into account characteristics such as the frequency of an activity like the frequency of letter sending, and a user's response time or response rate. The ranking may also be calculated based on background information about the contact. Such background information may include factors like closeness/relation/degree, company, position, and so on, from sources like social networks (e.g., number of friends, number of mutual friends, number of followers, etc.), content of the messages themselves (e.g., email signature), telephone communications (e.g., frequency of telephone calls and message exchange). The ranking may also be calculated based on other specific information that indicates that the contact is important. For example, the user herself may create her own tags in the system. Tags may be created to correspond with varying entities, such as organizations, persons, projects, places, and dates. If the user creates a specific tag (such as for an organization), the systems and methods of the illustrated embodiments may implement varying hypotheses. For example, the systems and methods disclosed herein may assume that this tag is more important to the user since the user created it herself. Moreover, the systems and methods disclosed herein may make certain associations automatically (i.e., the organization in the email address from whence the email originated is associated with the sender of the email, such as an employee) and query the user to verify these associations.

**[0090]** The system can search and analyze a user's emails, a contact's rating or ranking, telephone communications, and the like, and clears up the disorganization in the user's tasks. For this purpose, the system may implement a series of alerts, follow-ups, reminders (as well as by means of SMS). In one embodiment, the search results may be displayed differently in the tree structure depending on the information being searched. If, for example, the search request is interpreted by the mechanisms as a request to find a particular photo or set of photos, then the tree structure interface presented to the user may be reserved in large measure (e.g., 70%) for various photos which were located, while other possible files take up the remainder of the viewable area of the hierarchical tree structure area (e.g., 30%). These other possible files that may correlate in some way to the search may include music files, spreadsheet, document, presentation, and other files, for example. The 30% remaining viewable tree structure area may, accordingly, be taken by these possible files and displayed to the user.

**[0091]** In one embodiment, if a 100% match for a particular search query is not identified, the system may use various approaches such as morphology, semantics, fuzzy, and other approaches to perform a broader query, so as to eliminate a "no results found" result as many e-mail and file search systems would currently yield.

**[0092]** In one example, a user may query for a particular document by searching using the keyword "schedule." In a

typical search return string, perhaps **100** results of files containing the word “schedule” may be linearly listed and presented to the user. However, such a scenario may not be the most beneficial to the user. For example, this scenario may look similar to the following:

---

Search query: “Schedule”
Search results:
Doc 1 containing “Schedule”
Doc 2 containing “Schedule”
...
Doc 100 containing “Schedule”

---

**[0093]** One embodiment disclosed herein may add further results of terminology that the engine considers related, such as the following search for “schedule” that returned further results with the related terminology “program”:

---

Search query: “Schedule”
Search results:
Doc 1 containing “Schedule”
Doc 2 containing “Schedule”
...
Doc 100 containing “Schedule”
Doc 101 containing “Program”
Doc 102 containing “Program”
...
Doc 250 containing “Program”

---

**[0094]** In contrast to the above, in a further embodiment, the system may cluster the results of the searching by different common vectors, such as the following:

---

Search query: “Schedule”
Search results:
Cluster 1: “Your upcoming NY trip - 15 results”
Doc 1 containing “Schedule”
Doc 102 containing “Program”
See other 13 results ...
Cluster 2: “Your daughter’s school - 50 results”
Doc 24 containing “Schedule”
Doc 28 containing “Schedule”
Doc 209 containing “Program”
See other 47 results ...
Cluster 3: “Your last week Palo Alto event - 30 results”
Doc 15 containing “Schedule”
Doc 57 containing “Schedule”
Doc 234 containing “Program”
See other 27 results ...
Cluster 4: “Your this month emails - 5 results”
Doc 13 containing “Schedule”
Doc 47 containing “Schedule”
Doc 214 containing “Program”
See other 2 results ...
Cluster 5: “Your photos - 25 results”
Doc 63 containing “Schedule”
Doc 77 containing “Schedule”
Doc 211 containing “Program”
See other 22 results ...

---

**[0095]** In another embodiment, the system may conduct a search dialogue with a smart search assistant. Currently, search queries are beset with issues such as a return of zero results, or a return of too many results and/or coupled with poor ranking of those results (i.e., the target document is not ranked first in the list of results). The system and methods

disclosed herein introduce a live assistant. The live assistant takes verbal or textual queries from the user, who describes the data to be located. Based on this description, the live assistant may then query the user with suggestions. In one example, the live assistant may query the user for more information to broaden the search. In another example, the live assistant may suggest detail so as to narrow the search. In an illustrative embodiment, the live assistant is computer software code that can respond to inputs from the user and provide prompts to the user as disclosed herein. In an alternative embodiment, the live assistant may be a human operator using another computing device to communicate with the user and provide prompts to the user as disclosed herein.

**[0096]** The user may have the option to conduct a dialog (verbal, text, or otherwise) with the live assistant in a natural language format. For example, the user may type the search query “find all the emails which include information about my meeting with investors,” or order, “move to folder 1 all of the photos taken yesterday.” The live assistant may then respond with a confirmation, such as “would you like to delete all storyboard photos?”

**[0097]** As previously mentioned, the file search functionality provided by the systems and methods disclosed herein may be based on natural language and meaningful information extracted from the file and metadata. In one embodiment, image and pdf text files may be pre-Optical Character Read (OCR’d) (at least in some extent) so that if a user is searching the word “passport,” they can find images that include the word “passport.” Photos and videos may be searched based on GPS location data and metadata information about the time of shooting (e.g., a user can specify a range of time the photo was taken “from/to,” a user can specify the geographical units in their own language, “photo made in the United States”). Part of the metadata may be generated by the cloud service client system itself, such as, from where the file was received (e.g., saved from such a man of attachment to the letter, copied to the HDD #12, etc.).

**[0098]** Object recognition technology may also be applied to create metadata about the content of photos and video files, for searching. In one example, a user is looking for a photo where their family is at sea from 2005 to 2008. The system may to understand in advance that the image contains an object “body of water” so, when the search receives the search term “sea,” the mechanisms of the illustrated embodiments will find images containing a “body of water.” The system may also create an ontology of contacts, including relatives, their photos etc.

**[0099]** In addition, color, shape, size, and position tagging on business graphics may be employed. For example, a user may enter the search input: “Seeking Excel file, which has the round red diagram at the bottom of the last spreadsheet.”

**[0100]** In one embodiment, the systems and methods disclosed herein may implement photo tagging per the following example. The user may have 500 photos of Pete which he posts to a social media website. Subsequently, another individual tags one of the photos in which Pete appears. The mechanisms identify this tag of the photo, use this tag to create a “Pete” tag in the system, and query the user to confirm by displaying similar faces of Pete to the user.

**[0101]** In another embodiment, the user may have comments of his photos from other individuals in a particular social media. The comments are used by the mechanisms of

the illustrated embodiment for full-text searching, and extracted to create tags, which later are used to characterize various photos.

**[0102]** The methods and systems disclosed herein offer advantages over a simple keyword search, particularly since the methods and systems herein relate to a user environment across multiple devices, platforms, file formats, and user file sharing and social networking systems. A user may have to remember precise wording to find a desired file in traditional keyword searching. Advantageously, with the system and methods disclosed herein, a user may be able to find files, mails, and other objects by the general sense. To emulate search by the sense the system may parse the query, build a semantic structure and translate the semantic structure into an extended query using query interpretation and synonyms expansion based on dictionaries and ontologies. The extended query may produce many irrelevant results so the ranking algorithms, systems, and methods disclosed herein based on semantic similarities, word positions, word frequencies, usage levels, importance levels, etc. are valuable in ordering and ranking search results. If relevant results are not found, it is beneficial not to show irrelevant results unless a user allows or desires such irrelevant results to be displayed.

**[0103]** Additionally, searching can include parsing through all content related to the contacts of the user to seamlessly organize the search results for display, wherein the content includes at least the email data and the files, a multiplicity of social networks, data structures, email addresses, signatures of the email data, messages, SMS data, phone calls, text messages, video files, audio files, and images.

**[0104]** The displaying includes displaying label information, representative of each of the email data and files, to disclose each of the email data and files in a single graphical user interface (GUI) window by means of a client application. The displaying may also include displaying the hierarchical tree structure of all of the email data and files of one of the multiplicity of devices.

**[0105]** In a further embodiment, the result of the search operation is displayed in the single GUI window, and the email data and files on separate tabs are displayed in the single GUI window. The user may toggle between the email data and files with a single click. More specifically, the search results are displayed in the hierarchical tree structure information on a number of the email data and files that are relevant to a search query and associated files for the number of the email data and files.

**[0106]** FIG. 1 is a flow chart illustrating a method **100** of displaying a tree representation of search results in accordance with an illustrative embodiment. In alternative embodiments, fewer, additional, and/or different operations may be performed. Also, the use of a flow diagram is not meant to be limiting with respect to the order of operations performed. The method **100** begins at an operation **102**. In an operation **104**, a search operation of email data and files is performed. The search operation is common to each one of the multiplicity of devices for seamless display of the search results. The search operation is performed whether a particular one of the multiplicity of devices is online or offline, passive or active, connected or unconnected, currently operational or turned off, or set in a lower functional mode. The searching includes parsing through all content related to the contacts of the user to seamlessly organize the search results for display, where the content includes at least the email data and the files, a multiplicity of social networks, data structures, email

addresses, signatures of the email data, messages, SMS data, phone calls, text messages, video files, audio files, and images.

**[0107]** Pursuant to the search operation in an operation **106**, the system groups clusters of search results by a data type. In an operation **108**, the system displays the clusters with at least one search result corresponding to each of the clusters. In an operation **110**, the system accepts an input from the user to display a remaining set of results corresponding to a particular cluster not previously displayed. If the result of the search returns data corresponding to a data type in a number of results exceeding a predetermined percentage: those of the search results of the data type are displayed more prominently in the single GUI window, and those of the search result of remaining data types are aggregated and displayed less prominently in the single GUI window.

**[0108]** In an operation **112**, the system displays the search results (e.g., displays a most relevant portion of the search results) in a hierarchical tree structure, each portion of the hierarchical tree structure remaining visible to, and able to be manipulated by, a user regardless of whether a local device associated with the portion of the hierarchical tree structure is online or offline at a current time. A most relevant portion of the search results may be displayed while maintaining organizational relationships between the search results and related objects in the hierarchical tree structure, with the displayed portion of the results remaining visible to, and able to be manipulated by, a user and/or application regardless of whether a local device associated with the portion of the hierarchical tree structure is online or offline at a current time. The method also allows manipulating (e.g., via an application) the displayed portion, including allowing a traversal of the hierarchical tree structure to display at least one of the search results and the related objects. The method also allows manipulating (e.g., via an application) the displayed portion, including allowing a traversal of the hierarchical tree structure for determining a relationship between the most relevant portion of the search results that are displayed and the related objects. The method determines the most relevant portion of the search results based on one of predetermined criteria, access patterns of the multiplicity of devices, frequency of use of the multiplicity of devices, a ranking of the email data and files, a calculation operation performed on the email data and files. The displaying of the search results within the hierarchical tree structure, can include displaying label information, representative of each of the email data and files, to disclose each of the email data and files in a single graphical user interface (GUI) window by means of a client application. The displaying may also include displaying the result of the search operation in the single GUI window and/or displaying the email data and files on separate tabs in the single GUI window. The user may toggle between the email data and files with a single click. The displaying may also include information on a number of the email data and files that are relevant to a search query and associated files for the number of the email data and files. The method **100** ends at an operation **114**.

**[0109]** FIG. 2 is a flow chart illustrating a method **200** of displaying a tree representation of search results using ranking of search results in accordance with an illustrative embodiment. In alternative embodiments, fewer, additional, and/or different operations may be performed. Also, the use of a flow diagram is not meant to be limiting with respect to the order of operations performed. The method **200** begins at an operation **202**. In an operation **204**, the system performs a

search operation of email data and files. The search includes searching and analyzing the content in social networking channels, email address, data structures, signatures in messages, audio and/or video files, images, and/or other telephonic communication, text messages, SMS messages, and the like across multiple devices whether the devices are offline or online at a particular time. In an operation **206**, the system dynamically assigns a rank of importance to the search results. The ranking of importance may be based upon a numerical numbering system from 1 to 100 with 1 being the highest ranking (e.g., most important) and 100 being the lowest ranking (e.g., least most important). The user and/or application may employ other ranking methods. The rankings may be altered and/or reassigned as needed. In an operation **208**, the system links information returned from a search operation to a watchdog tag, which is used to monitor search results by performing subsequent searches using the watchdog tag. In an operation **210**, the system uses statistical information on directories and/or the tags in the search results. In an operation **212**, the system displays the search results in a hierarchical tree structure. More specifically, a most relevant portion (and/or, a desired portion, a selected portion, a predetermined portion, a calculated portion, and/or the least most relevant portions) of the search results may be displayed while maintaining organizational relationships between the search results and related objects in a hierarchical tree structure, with the displayed portion of the results remaining visible to, and able to be manipulated by, a user and/or application regardless of whether a local device associated with the portion of the hierarchical tree structure is online or offline at a current time. The hierarchical tree structure may be displayed in a single graphical user interface (GUI) window by means of a client application. Each portion of the hierarchical tree structure remains visible to, and able to be manipulated by, a user and/or the client application regardless of whether a local device associated with the portion of the hierarchical tree structure is online or offline at a current time. The method **200** ends at an operation **214**. When switching to directories (e.g., switching directories in the GUI display page), the selection in the search results may be narrowed (or expanded).

**[0110]** FIG. 3 is a block diagram illustrating a computer system **300** that may be used in accordance with an illustrative embodiment. In alternative embodiments, fewer, additional, and/or different components may be included in the system. The system **300** includes one or more processors **302** connected to a memory **304**. The processor(s) **302** may contain one or more computer cores or may be a chip or other device capable of performing computations (for example, a Laplace operator may be produced optically). The memory **304** may be random access memory (RAM) and may also contain any other types or kinds of memory, particularly non-volatile memory devices (such as flash drives) or permanent storage devices such as hard drives, etc. In addition, an arrangement can be considered in which the memory **304** includes remotely located information-storage media, as well as local memory such as cache memory in the processor(s) **302**, used as virtual memory and stored on an external or internal permanent memory device **310**.

**[0111]** The computer system **300** also usually includes input and output ports to transfer information out and receive information. For interaction with a user, the computer system **300** may contain one or more input devices **306** (such as a keyboard, a mouse, a scanner, or other) and output devices

**308** (such as displays or special indicators). The computer system **300** may also have one or more permanent storage devices **310** such as an optical disk drive (CD, DVD, or other), a hard disk, or a tape drive. In addition, the computer system **300** may have an interface with one or more networks **312** that provide connection with other networks and computer equipment. In particular, this may be a local area network (LAN) or a Wi-Fi network, and may or may not be connected to the World Wide Web (Internet). It is understood that the computer system **300** may include analog and/or digital interfaces between the processor **302** and each of the components **304**, **306**, **308**, **310** and **312**.

**[0112]** The computer system **300** is managed by an operating system **314** and includes various applications, components, programs, objects, modules and other, designated by the consolidated number **316**. The programs used to accomplish the methods corresponding to the embodiments disclosed herein may be a part of an operating system or may be a specialized peripheral, component, program, dynamic library, module, script, or a combination thereof.

**[0113]** FIG. 4 is a representation of a graphical user interface (GUI) **400** demonstrating a tree representation of search results in accordance with an illustrative embodiment. In alternative embodiments, fewer, additional, and/or different components may be included in the GUI. The GUI **400** display shows, among other aspects, relationships between various functionality brought to bear by mechanisms of the illustrated embodiments, as described herein. In addition, the sample GUI **400** screens also depict various additional functionality as disclosed herein.

**[0114]** In one embodiment, software incorporating the embodiments disclosed herein may be installed via sharing (link shared via email or other social networking contact), file sharing requiring installation of the software itself, or download from the appropriate website. Upon first use, the software may prompt for login information, such as the following:

**[0115]** Already using (Software name)?

**[0116]** Your email . . .

**[0117]** Password . . .

**[0118]** New user?

**[0119]** Your email . . .

**[0120]** (Once email address is entered, a confirmation message containing a temporary password may be sent to the user's email.)

**[0121]** After confirming the user's email, the user will see a software window **400**, configured and/or customized by and/or for the user. In one embodiment, on the right **402**—the working part, it is empty initially. On the left side **404** the user views available icons for sources such as my hard drive, places, documents, photos, attachments, external drives, social media, if currently connected, cloud drives, Exchange, mail servers (Gmail™ etc.), other locations: camera, media centers etc., and other computers. A visible arrow **414** on the GUI **400** screen offers to drag any icon in the left (workspace) it will mean that the user assigns that service/device to their personal cloud.

**[0122]** At the top of the GUI **400** window, the user views one or more large tabs **406**, such as Logo, Files, E-Mail, Contacts, work, all, important, friends, social, promo, and/or more, and the like (only the logo **406A**, email tab **406B**, and the files **406C** is labeled as **406** for brevity sake). Additional tabs will appear later: such as Calendar, and Tasks, and others. There is also a large search field **412**. The software feeds the

user with information among the various tabs **406**. As analyzing and/or indexing begins, the software displaying progress and results of analyzing/indexing. In the analyzing and indexing, the system detects duplicate files in the user's personal cloud. For example, such duplicates may be categorized as follows: 100% duplicates: the content and metadata includes the date and time. 99.9% of duplicate: files, including content, metadata are the same, but a different date and time. Optional: 99% similar: where there are small differences in content.

**[0123]** The smart searching and analysis of the user's contacts offers to consider these files as different versions and ranks the date and author of last modification, looks at which of the files come in the mail, by whom, when etc. The system may offer to merge duplicates in one virtual file. In such a case, the system may treat these physical files as a single logical file and synchronize the changes of this file across all its physical copies. If the duplicate is located within the same physical medium, the software will remove the physical duplicate (keeping logical copy).

**[0124]** The GUI **400** screen also shows the ranked content (shown in FIG. **4** as **420A** being the ranked the highest as Rank **1** and the remaining ranked contents "rank **2-100**" shown as **420B**). The search operation may also display the search results in a summary **422**. There may also be a smart analysis result **424** as part of the search results. The terminology described herein as "smart analysis" is a search and analysis of a user's contacts, including, for example, analyzing the user's emails (e.g., emails relating to one or more contacts) and clearing up the user's disorganization in the multiple devices (e.g., the disorganization of contacts in these multiple devices). The data summary **422** may include information like how many results were found, how many results were found from each electronic device, the total size of all of the results, the totals in size and quantity by type or format of results, length of time taken to complete search, and other data relating to a search. The smart analysis results **424** may include information about how the results were ranked and prioritized according to the methods and systems as disclosed herein.

**[0125]** The GUI **400** screen also displays the search results in a hierarchical tree structure **408**. The hierarchical tree structure **408** may be displayed in a single graphical user interface (GUI) **400** window by means of a client application. Each portion of the hierarchical tree structure remaining visible to, and able to be manipulated by, a user and/or the client application regardless of whether a local device associated with the portion of the hierarchical tree structure is online or offline at a current time.

**[0126]** FIG. **5** is a flow chart illustrating a method of displaying a tree representation of search results across a multiplicity of devices in accordance with an illustrative embodiment. In alternative embodiments, fewer, additional, and/or different operations may be performed. Also, the use of a flow diagram is not meant to be limiting with respect to the order of operations performed. The method **500** begins at an operation **502**. In an operation **504**, the system performs a search operation of email data and files across a multiplicity of devices. The search operation is performed regardless of whether the multiplicity of devices are either online and/or offline. The search operation common to each of the multiplicity of devices for seamless display of results. In an operation **506**, the system displays a portion of the search results (e.g., a most relevant portion) while maintaining organiza-

tional relationships between the search results and related objects in a hierarchical tree structure, with the displayed portion of the results remaining visible to, and able to be manipulated by, a user and/or client application regardless of whether a local device associated with the portion of the hierarchical tree structure is online or offline at a current time.

**[0127]** In an operation **508**, the displayed portion of the hierarchical tree structure is manipulated. The manipulating includes allowing a traversal of the hierarchical tree structure to display at least one of the search results and the related objects. The manipulating the displayed portion, includes allowing a traversal of the hierarchical tree structure for determining a relationship between the most relevant portion of the search results that are displayed and the related objects.

**[0128]** In an operation **510**, the system determines the portion of the search results to be displayed (e.g., the most relevant portion of the search results). In one embodiment, the portion of the search results to be displayed is determined by one or more of the following: predetermined criteria set by the user and/or an application, historical usage and/or access patterns of the multiplicity of devices, a frequency of use of the multiplicity of devices, a ranking of the email data and files, a calculation operation performed on the email data and files, a filtering and/or a queuing operation for creating a most relevant ranking/queue and a least most relevant ranking/queue of the search results based on the search queue, search results obtained from the multiplicity of devices for those of the multiplicity of devices that are offline and those of the multiplicity of devices that are online, and/or most related search results. The method **500** ends at an operation **512**.

**[0129]** In one embodiment, the methods and systems disclosed herein provide a solution for an efficient tree representation of search results across a multiplicity of devices by a processor device. A search operation is performed of email data and files. The search operation common to each of the multiplicity of devices for seamless display of results. A most relevant portion of the search results is displayed while maintaining organizational relationships between the search results and related objects in a hierarchical tree structure, with the displayed portion of the results remaining visible to, and able to be manipulated by, a user regardless of whether a local device associated with the portion of the hierarchical tree structure is online or offline at a current time.

**[0130]** In one embodiment, the methods and systems disclosed herein allow manipulating the displayed portion, including allowing a traversal of the hierarchical tree structure to display at least one of the search results and the related objects. The methods and systems disclosed herein allows manipulating the displayed portion, including allowing a traversal of the hierarchical tree structure for determining a relationship between the most relevant portion of the search results that are displayed and the related objects.

**[0131]** In one embodiment, the methods and systems disclosed herein determines the most relevant portion of the search results based on one of predetermined criteria, access patterns of the multiplicity of devices, frequency of use of the multiplicity of devices, a ranking of the email data and files, a calculation operation performed on the email data and files.

**[0132]** FIG. **6** is a representation of a GUI **600** demonstrating a hierarchical tree structure search results in accordance with an illustrative embodiment. In alternative embodiments, fewer, additional, and/or different components may be displayed on the GUI. The GUI **600** includes an e-mails tab **605** and a files tab **610**. Here the files tab **610** is selected. Further,

the Macintosh HD is selected from the workspace. Accordingly, in this embodiment, the search only applies to files stored on the Macintosh HD electronic device. In the workspace, a pointer has been directed by a user to the attachments/dropbox location **615**. As a result, the system has displayed a highlighted box around the attachments/dropbox location **615**. Further, in the workspace, certain locations also indicate how many files relevant to the search that are physically stored at each location. For example, the Macintosh HD has 249 files stored on its hard drive relevant to the search being performed here. In an alternative embodiment, such a number may indicate something different, such as files indexed from this location within the last week or month.

[0133] The GUI **600** also includes a search analysis section **620**. This section indicates trends, data, and/or statistics regarding the search results. For example, 44 of the search results are associated with the tag “GREEN” and eleven of the search results are from files from the year 2012. In alternative embodiments, other statistics may also be displayed, such as size of files, author of files, sender of files, number of versions or changes from original, or importance. Such data may be displayed if the user opens the smart analysis results **424** of FIG. 4 discussed above.

[0134] A search term “REDCAT” has been entered into a search dialog **625**. Accordingly, each one of the files or file representations in the GUI **600** are related to the search term. Additionally, each time the search input REDCAT appears on the GUI **600**, it is highlighted with a box around it because that is the current search term. Highlights could also involve shading, different shapes, different colors, or any other method that makes the search input stand out from the other text on the page. A hierarchical tree structure **630** is shown in the GUI **600**. The hierarchical tree structure **630** shows a navigable hierarchy that shows folders and sub-folders where relevant search results are located. Each level of the hierarchy also includes an indicator of how many relevant search results are in each folder. The folders displayed here can represent physical or virtual folders. A search results section **640** may show the contents of a folder selected from the file tree hierarchy **630**. In some embodiments, if no folder is currently selected (other than the fact that the search results represent only result from the Macintosh HD, which is selected on the left of the GUI **600**), the search results section **640** shows results from the total results of the search. The system may automatically populate the search results **640** with various relevant and or high ranking search results **660**, such as photo **650** and document **665**. The search results **660** may also be modified or sorted using a drop down menu **645**. Here it is indicated that the results have been modified.

[0135] The hierarchical tree structure **630** in the GUI shows search results grouped or clustered together in a way that is convenient for the user. The file tree hierarchy starts out with a broad folder, Macintosh HD folder **632**. The folder **632** is displayed as darkened black. In this embodiment, this indicates that search results are present in the folder **632**. Additionally, alongside the folder name of the folder **632** is the number of search results that are in the folder **632**, in this case **249**. If the user selects the folder **632**, all of the results of the search can be viewed. However, the folder **632** can be expanded to reveal further sorted and/or clustered search results. For example, in FIG. 6, a work sub-folder project is selected and expanded out. In some embodiments, the system may automatically display some folders already expanded out based on ranking, total number of search results, etc.

Advantageously, each folder also shows how many search results are part of that cluster or group. The work folder shows 201 results. Any of the sub-folders, in addition to being expanded, can also be selected. Once selected, the search results are displayed in the search results section **640**. Advantageously, the hierarchical tree structure **630** gives the user of the GUI **600** a convenient and simple way to manageably view and deal with large numbers of search results. The project reecat folder **635** also appears (based on the appearance of the icon and the name of the folder) to have search results. However, a number indicating how many search results were found there is not present. Here, that is because the folder **635** merely includes the search input term in the name of the folder but does not contain relevant search results. This is in contrast to the project reecat folder **634**, which shows 156 search results. The look folder **637** shows just one example of how a folder with no relevant search results can be visually differentiated from the other folders in the hierarchical tree structure **630**. The folder **637** is shown with a white in color folder symbol and the text of the folder **637** has been grayed out.

[0136] There are also subsections of the search results **660**. Here, the subsections include media and document type results, though other types may be included in other embodiments such as contacts and messages. The media results also include a show all button **647**. By interacting with the show all button **647**, all of the media results may be displayed. In the alternative, if the user would like to view more results, an arrow **655** may be interacted with to scroll through the different media results that are not currently displayed.

[0137] Here, document **665** has been selected from among the search results. Accordingly in preview section **675**, a preview **680** of the document **665** is displayed. Further information **690** about the previewed file is also displayed. If a user has previously made a reminder regarding the file, the reminder is also displayed. Here, reminder **685** is displayed along with the preview **680**. At collaborators **695**, the system may display who has collaborated on the file with the user. However, in this embodiment, nobody has collaborated on the file.

[0138] FIG. 7 is a block diagram illustrating various computing and electronic storage devices that may be used in accordance with an illustrative embodiment. In alternative embodiments, fewer, additional, and/or different components may be included in the system. FIG. 7 includes a personal computing device **700**, a portable storage device **797**, a network **725**, a cloud storage system **730**, a tablet device **740**, and a mobile electronic device **765**. The personal computing device **700** includes a processor **715** that is coupled to a memory **705**. The personal computing device **700** can store and recall data and applications in the memory **705**. The processor **715** may also display objects, applications, data, etc. on a display/interface **710**. The display/interface **710** may be a touchscreen, a game system controller, a remote control, a keyboard, a mouse, a trackpad, a microphone, a camera, a set of buttons, a standard electronic display screen, a television, a computer monitor, or any combination of those or similar components. The processor **715** may also receive inputs from a user through the display/interface **710**. The processor **715** is also coupled to a transceiver **720**. With this configuration, the processor **715**, and subsequently the personal computing device **700**, can communicate with other devices, such as the cloud storage system **730** through a connection **787** and the network **725**. Although FIG. 7 shows

one personal computing device **700**, an alternative embodiment may include multiple personal computing devices.

[0139] The tablet device **740** includes a processor **755** that is coupled to a memory **745**. The processor **755** can store and recall data and applications in the memory **745**. The processor **755** may also display objects, applications, data, etc. on a display/interface **750**. The display/interface **750** may be a touchscreen as most tablets have, but may also include or incorporate a keyboard, a game system controller, a remote control, a mouse, a trackpad, a microphone, a camera, a set of buttons, a standard electronic display screen, a television, a computer monitor, or any combination of those or similar components. The processor **755** may also receive inputs from a user through the display/interface **750**. The processor **755** is also coupled to a transceiver **760**. With this configuration, the processor **755**, and subsequently the tablet device **740**, can communicate with other devices, such as the personal computing device **700** through a connection **795** and the network **725**.

[0140] The mobile electronic device **765** includes a processor **775** that is coupled to a memory **785**. The processor **775** can store and recall data and applications in the memory **785**. The processor **775** may also display objects, applications, data, etc. on a display/interface **780**. The display/interface **780** may be a touchscreen, a game system controller, a keyboard, a remote control, a mouse, a trackpad, a microphone, a camera, a set of buttons, a standard electronic display screen, a television, a computer monitor, or any combination of those or similar components. The processor **775** may also receive inputs from a user through the display/interface **780**. The processor **775** is also coupled to a transceiver **770**. With this configuration, the processor **775**, and subsequently the viewer electronic device **765**, can communicate with other devices, such as the tablet device **740** through a connection **790** and the network **725**. Although FIG. 7 shows only one mobile electronic device **765**, an alternative embodiment may include multiple mobile electronic devices.

[0141] FIG. 7 also includes the cloud storage system **730**. The cloud storage system **730** may include a number of servers that may have memory and processors. The cloud storage system **730** is connected to the network through a connection **735** and may communicate with other devices such as the mobile electronic device **765**.

[0142] FIG. 7 also includes the portable storage device **797**. This portable storage device **797** may be a removable USB drive for example. Since such a portable memory does not have a processor, the portable storage device **797** must be connected to a computing device for the information and files stored on the portable storage device **797** can be read by such devices. Accordingly, in this embodiment, the portable storage device **797** is plugged into the personal computing device **700**, which can read the data from the portable storage device **797** and communicate its contents or data related to its contents to any of the other devices that have access to the network **725**.

[0143] The devices shown in the illustrative embodiment may be utilized in various ways. For example, any of the connections **787**, **790**, **795**, and **735** may be varied. Any of the connections **787**, **790**, **795**, and **735** may be a hard wired connection. A hard wired connection may involve connecting the devices through a USB (universal serial bus) port (like connection **799**), serial port, parallel port, or other type of wired connection that can facilitate the transfer of data and information between a processor of a device and a second

processor of a second device. In another embodiment, any of the connections **787**, **790**, **795**, and **735** may be a dock where one device may plug into another device. While plugged into a dock, the client-device may also have its batteries charged or otherwise be serviced. In other embodiments, any of the connections **787**, **790**, **795**, and **735** may be a wireless connection. These connections may take the form of any sort of wireless connection, including but not limited to Bluetooth connectivity, Wi-Fi connectivity, or another wireless protocol. Other possible modes of wireless communication may include near-field communications, such as passive radio-frequency identification (RFID) and active (RFID) technologies. RFID and similar near-field communications may allow the various devices to communicate in short range when they are placed proximate to one another. In an embodiment using near field communication, two devices may have to physically (or very nearly) come into contact, and one or both of the devices may sense various data such as acceleration, position, orientation, velocity, change in velocity, IP address, and other sensor data. The system can then use the various sensor data to confirm a transmission of data over the internet between the two devices. In yet another embodiment, the devices may connect through an internet (or other network) connection. That is, any of the connections **787**, **790**, **795**, and **735** may represent several different computing devices and network components that allow the various devices to communicate through the internet, either through a hard-wired or wireless connection. Any of the connections **787**, **790**, **795**, and **735** may also be a combination of several modes of connection. The network **725** may also include similar components described above with respect to the connections **787**, **790**, **795**, and **735**. In addition, the network **725** may include intermediate servers, routing devices, processors, data traffic management services, and wired or un-wired connections.

[0144] To operate different embodiments of the system or programs disclosed herein, the various devices may communicate using the software systems and methods disclosed herein. Software applications may be manually installed on the devices or downloaded from the internet. Such software applications may allow the various devices in FIG. 7 to perform some or all of the processes and functions described herein. Additionally, the embodiments disclosed herein are not limited to being performed only on the disclosed devices in FIG. 7. It will be appreciated that many various combinations of computing devices may execute the methods and systems disclosed herein. Examples of such computing devices may include smart phones, personal computers, servers, laptop computers, tablets, blackberries, RFID enabled devices, video game console systems, smart TV devices, or any combinations of these or similar devices.

[0145] In one embodiment, a download of a program to the mobile electronic device **765** involves the processor **775** receiving data through the transceiver **770** through connection **790** and the network **725**. The network **725** may be connected to the internet. The processor **775** may store the data (like the program) in the memory **785**. The processor **775** can execute the program at any time. In another embodiment, some aspects of a program may not be downloaded to the viewer electronic device **765**. For example, the program may be an application that accesses additional data or resources located in a server, or even the cloud storage system **730** which may include one or more servers. In another example, the program may be an internet-based application, where the program is executed by a web browser and stored in a server

that is part of the network **725** or the cloud storage system **730**. In the latter example, temporary files and/or a web browser may be used on the mobile electronic device **765** in order to execute the program, system, application, etc. In additional embodiments, the tablet device **740** and the personal computing device **700** may use, store, or download software applications and web based programs in a similar way.

**[0146]** The configuration of the personal computing device **700**, the portable storage device **797**, the tablet device **740**, the mobile electronic device **765**, the network **725**, and the cloud storage system **730** is merely one physical system on which the disclosed embodiments may be executed. Other configurations of the devices shown exist to practice the disclosed embodiments. Further, configurations of additional or fewer devices than the ones shown in FIG. 7 may exist to practice the disclosed embodiments. Additionally, the devices shown in FIG. 7 may be combined to allow for fewer devices or separated where more than the five devices shown exist in a system.

**[0147]** FIG. 8 is a flow diagram illustrating a method **800** of performing a cross-platform search in accordance with an illustrative embodiment. In alternative embodiments, fewer, additional, and/or different operations may be performed. Also, the use of a flow diagram is not meant to be limiting with respect to the order of operations performed. In an operation **805**, the system receives search terms input. The search terms may be received from a keyboard, an on-screen keyboard, speech-to-text technology, or other input from a user. In another embodiment, the user may select a search term from a graphical user interface without having to enter particular characters and text with an input device. For example, the user may select a tag to initiate a search as discussed above with respect to FIG. 6.

**[0148]** In an operation **810**, the system searches various electronic devices and their electronic memory in order to find electronic files related to the search input. In an alternative embodiment, and as discussed above, the system may also search an index of electronic files rather than the actual electronic memories of various devices. In a third embodiment, the system may actually search a combination of an index and the physical memory of various user devices.

**[0149]** In an operation **815**, the system determines the presence of various electronic files relating to the input search terms. In this step, the system actually finds hits relating to the search terms.

**[0150]** In an operation **820**, the system displays the various electronic files relating to the input search terms on a graphical user interface (GUI). That is, the system displays the search results. The search results may be displayed in a variety of ways as disclosed herein, including a tree representation or hierarchical tree structure.

**[0151]** FIG. 9 is a flow diagram illustrating a method **900** of grouping search results into clusters in accordance with an illustrative embodiment. In alternative embodiments, fewer, additional, and/or different operations may be performed. Also, the use of a flow diagram is not meant to be limiting with respect to the order of operations performed. In an operation **905**, the system determines electronic files from multiple electronic devices that are related to search terms input from a user.

**[0152]** In an operation **910**, the system groups the electronic files that were determined based on the search terms into groups (or clusters or amalgamations) with common

characteristics. For example and as disclosed herein, the groupings may involve electronic files of a similar type, date, relation to a contact or particular conversation, etc. In an operation **915**, the system displays on the GUI the various clusters and groups relating to the search results. The system may display the groups in many different ways as disclosed herein.

**[0153]** In an operation **920**, the system receives an input from the user regarding a particular group or cluster. The input may be a click from a mouse pointer, a touch through a touch screen, a voice command, or any other type of user input. In an operation **925**, the system displays additional information on the GUI regarding the group or cluster. For example, the system may display information about the cluster as a whole: total size, number of electronic files, etc. In another example, the system may display information about the electronic files that make up the cluster: names of electronic files in the group, sizes of the electronic files in the group, previews of the electronic files in the group, etc. Furthermore, the system may only display information regarding electronic files that are regarded as higher relevance to the search than other electronic files in the grouping. Accordingly, when the system displays additional information about the files, the more relevant files are displayed more prominently on the GUI.

**[0154]** FIG. 10 is a flow diagram illustrating a method **1000** of displaying search results in a hierarchical tree structure in accordance with an illustrative embodiment. In alternative embodiments, fewer, additional, and/or different operations may be performed. Also, the use of a flow diagram is not meant to be limiting with respect to the order of operations performed. In an operation **1005**, the system determines search results according to a search input.

**[0155]** In an operation **1010**, the system sorts the search results into a first level of the tree structure based on a first criteria. In this embodiment, the first criteria is a relatively broad criteria meant to capture many if not all of the search results. For example, the criteria may consist of all search results. In another example, the criteria may consist of all documents in the search results. In this way, the hierarchical tree structure can be organized from broader tree structure levels to narrower to facilitate a useful interface for the user.

**[0156]** In an operation **1015**, the system sorts the search results into a second level of the tree structure based on a second criteria. The second criteria is relatively narrower as compared to the first criteria. For example, where the first criteria consists of all documents in the search results, the second criteria may be all documents received as e-mail attachments in the search results. As explained above, in this way, the hierarchical tree structure can offer an organization of search results that continually narrows the results for a user in a coherent manner that makes it easy to locate particular search results. In another embodiment where a folder structure is utilized, the narrowing criteria is inherent since subfolders are already stored within other folders (and are therefore subservient).

**[0157]** In an operation **1020**, the system displays the hierarchical tree structure on the GUI. Here the user can actually interact with the hierarchical tree structure and view the search results. This can be done in many ways according to the embodiments disclosed herein.

**[0158]** FIG. 11 is a flow diagram illustrating a method **1100** of customizing search result groupings according to user defined groups in accordance with an illustrative embodi-

ment. In alternative embodiments, fewer, additional, and/or different operations may be performed. Also, the use of a flow diagram is not meant to be limiting with respect to the order of operations performed. In an operation 1105, the system receives custom definitions from the user on how to cluster search results. In other words, the user has determined here the criteria for how to determine clusters of search results.

[0159] In an operation 1110, the system receives search terms and input from the user. In an operation 1115, the system performs a search based on the search terms and input from the user.

[0160] In an operation 1120, the system organizes the search results in the user defined clusters. In an operation 1125, the system receives an interaction from the user. The interaction received is a selection one of the clusters generated in the operation 1120. As a result of the interaction, the system displays more information regarding the cluster, including a preview of at least one electronic file represented in the cluster that they system deems to be more relevant to the search terms than other electronic files in the cluster.

[0161] In an operation 1130, the system receives a second interaction from the user, and the system subsequently opens the electronic file. Here, the user presumably likes what he or she was seeing from the preview displayed in the operation 1125, so the user further interacted with the preview and the cluster to open the electronic file.

[0162] As will be appreciated by one skilled in the art, aspects of the embodiments disclosed herein may be embodied as a system, method or computer program product. Accordingly, aspects of the embodiments disclosed herein may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a “circuit,” “module” or “system.” Furthermore, aspects of the embodiments disclosed herein may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

[0163] Any combination of one or more computer readable medium(s) may be utilized. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that may contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

[0164] Program code embodied on a computer readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wired, optical fiber cable, RF, etc., or any suitable combination of the foregoing. Computer program code for carrying out operations for aspects of the embodiments disclosed herein may be written

in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the “C” programming language or similar programming languages. The program code may execute entirely on the user’s computer, partly on the user’s computer, as a stand-alone software package, partly on the user’s computer and partly on a remote computer, or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user’s computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

[0165] Aspects of the embodiments disclosed herein are described with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments disclosed herein. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, may be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, implement the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0166] These computer program instructions may also be stored in a computer readable medium that may direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the flowchart and/or block diagram block or blocks. The computer program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatus or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0167] The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments disclosed herein. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, may be implemented by special purpose hardware-based systems that perform the specified

functions or acts, or combinations of special purpose hardware and computer instructions.

**[0168]** In an illustrative embodiment, any of the operations described herein can be implemented at least in part as computer-readable instructions stored on a computer-readable medium or memory. Upon execution of the computer-readable instructions by a processor, the computer-readable instructions can cause a computing device to perform the operations.

**[0169]** The foregoing description of illustrative embodiments has been presented for purposes of illustration and of description. It is not intended to be exhaustive or limiting with respect to the precise form disclosed, and modifications and variations are possible in light of the above teachings or may be acquired from practice of the disclosed embodiments. It is intended that the scope of the invention be defined by the claims appended hereto and their equivalents.

What is claimed is:

1. A method according to a set of instructions stored on a memory of a computing device, the method comprising:

receiving, by a processor of the computing device, a search input;

determining, by the processor, the presence of a plurality of electronic files related to the search input; and

displaying, by the processor, on a graphical user interface (GUI), representative information of the plurality of electronic files related to the search input, wherein the representative information comprises a hierarchical tree structure.

2. The method of claim 1, wherein the determining the presence of the plurality of electronic files related to the search input comprises searching a plurality of different electronic storage locations.

3. The method of claim 1, further comprising receiving, by the processor, a manipulation command.

4. The method of claim 3, wherein the manipulation command comprises a request for expanding a first level of the hierarchical tree structure.

5. The method of claim 4, further comprising displaying, by the processor and in response to the manipulation command, preview data of at least one electronic file associated with the first level of the hierarchical tree structure.

6. The method of claim 4, further comprising displaying, by the processor and in response to the manipulation command, a second level of the hierarchical tree structure, wherein the second level is related to the first level, and further wherein the second level was not displayed on the GUI prior to reception of the manipulation command.

7. The method of claim 1, wherein the hierarchical tree structure comprises a plurality of folders, and further wherein some of the plurality of folders are subordinate to other of the plurality of folders.

8. The method of claim 7, wherein the plurality of folders indicate different physical locations of the plurality of electronic files, different virtual locations of the plurality of electronic files, or different data types of the plurality of electronic files.

9. The method of claim 7, wherein displaying the representative information comprising a hierarchical tree structure comprises:

determining, by the processor, a number of the plurality of electronic files in each of the plurality of folders that are related to the search input; and

displaying, by the processor, the number of the plurality of electronic files in each of the plurality of folders that are related to the search input alongside a representation of each of the plurality of folders.

10. The method of claim 9, further comprising:

determining, by the processor, that one of the plurality of folders contains no electronic files related to the search input;

displaying, by the processor, a representation of the folder with no electronic files related to the search input that is different than the representation of folders that contain electronic files related to the search input.

11. The method of claim 10, wherein the representation of the folder with no electronic files related to the search input is displayed as a different color or shade of color than the representation of folders that contain electronic files related to the search input.

12. The method of claim 1, further comprising determining, by the processor, a more relevant portion of the hierarchical tree structure based on a predetermined criteria for organizing search results.

13. The method of claim 12, wherein the predetermined criteria comprises a device access pattern of one or more electronic storage locations where the plurality of electronic files are stored, a frequency of use of the one or more electronic storage locations, file access patterns of each of the plurality of electronic files, file frequency of use of each of the plurality of electronic files, a device ranking of the one or more electronic storage locations, or a file ranking of each of the plurality of electronic files.

14. The method of claim 12, wherein the more relevant portion of the hierarchical tree structure is displayed on the GUI.

15. The method of claim 14, wherein a less relevant portion of the hierarchical tree structure is not displayed on the GUI unless a manipulation command to display the less relevant portion is received.

16. The method of claim 1, wherein the plurality of electronic files comprise both e-mails and other data files, and further wherein the method comprises displaying, by the processor, the e-mails and the other data files on separate tabs on the GUI, wherein display of the separate tables can be toggled back and forth between the e-mails and the other data and wherein the GUI comprises a single window.

17. The method of claim 1, further comprising:

grouping, by the processor, clusters of the plurality of electronic files by data type; and

displaying, by the processor and on the GUI, cluster representative information regarding each of the clusters of the plurality of electronic files.

18. The method of claim 17, wherein sub-folders of the cluster representative information indicate virtual location or physical storage location.

19. An apparatus comprising:

a memory;

a processor coupled to the memory; and

a first set of instructions stored on the memory and configured to be executed by the processor, wherein the processor is configured to:

receive a search input;

determine the presence of a plurality of electronic files related to the search input; and

display, on a graphical user interface (GUI), representative information of the plurality of electronic files related to

the search input, wherein the representative information comprises a hierarchical tree structure.

**20.** A non-transitory computer readable medium having instructions stored thereon that, upon execution by a computing device, cause the computing device to perform operations, wherein the instructions comprise:

- instructions to receive a search input;
- instructions to determine the presence of a plurality of electronic files related to the search input; and
- instructions to display, on a graphical user interface (GUI), representative information of the plurality of electronic files related to the search input, wherein the representative information comprises a hierarchical tree structure.

\* \* \* \* \*