



(51) International Patent Classification:

H04W 36/36 (2009.01) H04W 84/12 (2009.01)  
H04W 12/06 (2009.01) H04L 12/28 (2006.01)

(21) International Application Number:

PCT/US2014/065582

(22) International Filing Date:

14 November 2014 (14.11.2014)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

14/081,202 15 November 2013 (15.11.2013) US

(71) Applicant: MICROSOFT TECHNOLOGY LICENSING, LLC [US/US]; One Microsoft Way, Redmond, WA 98052 (US).

(72) Inventors: ALAM, Mohammad Shabbir; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). ASSAD, Javier Flores; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). BANKS, Nicholas A.; c/o Microsoft Corporation, LCA - International Patents

(8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). GOYAL, Piyush; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). GRAY, James Christopher; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). GUDAY, Shai; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). KUEHNEL, Thomas W.; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). LAMBA, Triptpal Singh; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). MACDONALD, David Neil; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). MAZANDARANY, Darya; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). NABAR, Sidharth Uday; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). RICE, Christopher R.; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way,

[Continued on next page]

(54) Title: APPLYING CROWDSOURCED SEQUENCED INSTRUCTIONS TO AUTOMATICALLY INTERACT WITH CAPTIVE PORTALS OF WI-FI NETWORKS

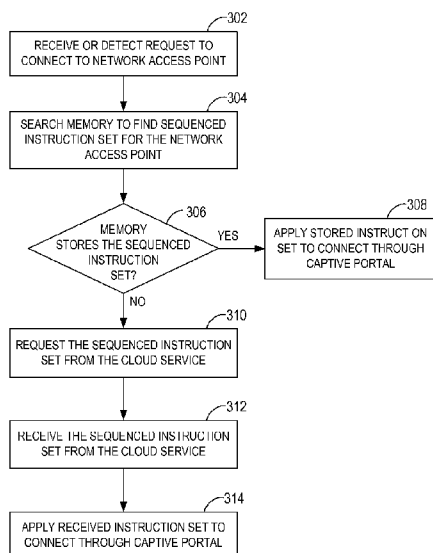


FIG. 3

(57) Abstract: A computing device detects a network access point's SSID (e.g. Wi-Fi) and obtains, through a cloud service from crowdsourced data describing interactions between mobile computing devices and the network access point, a set of sequenced instructions enabling the device to automatically interact with the captive portal in order to connect to the network. Applying the instruction set includes performing actions such as navigating web pages to accept terms and conditions, provide user or device information, and more.

WO 2015/073754 A1



Redmond, Washington 98052-6399 (US). **SHARMA, Saumaya**; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). **STAMPER, Douglas E.**; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US).

GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

**(81) Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

**Declarations under Rule 4.17:**

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

**Published:**

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

**(84) Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH,

## APPLYING CROWDSOURCED SEQUENCED INSTRUCTIONS TO AUTOMATICALLY INTERACT WITH CAPTIVE PORTALS OF WI-FI NETWORKS

## BACKGROUND

5 [0001] The availability of network access points, such as wireless fidelity (Wi-Fi) hotspots, has increased significantly. Many of these access points are gated and require user intervention for connection to the Internet or other networks. For example, the user may be required to accept terms and conditions, provide an email address or hotel room number, etc. Some existing solutions attempt to help the users connect to networks, such as by  
10 managing multiple passwords with a single authentication personal identification number (PIN). Some other existing solutions attempt to locate Wi-Fi hotspots near the user. However, these solutions lack a seamless auto-connect experience and fail to leverage understanding gained from connection attempts by other users.

## SUMMARY

15 [0002] Embodiments of the disclosure automatically connect to a network through a captive portal based on crowdsourced instructions. A computing device detects a network access point. The network access point acts as a captive portal to a network. A sequenced instruction set associated with the detected network access point is requested and received from a cloud service. The sequenced instruction set is derived from crowdsourced data  
20 describing interactions between mobile computing devices and the detected network access point. The computing device applies the received sequenced instruction set to the captive portal to obtain access to the network via the network access point.

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended  
25 to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIG. 1 is an exemplary block diagram illustrating mobile computing devices connecting to networks through network access points.

[0005] FIG. 2 is an exemplary block diagram illustrating a mobile computing device  
30 storing computer-executable components for replaying instructions to connect to networks through captive portals.

[0006] FIG. 3 is an exemplary flow chart illustrating operation of a computing device to obtain and apply an ordered set of instructions connect through a captive portal.

[0007] FIG. 4 is an exemplary block diagram illustrating a cloud service storing sets of sequenced instructions corresponding to captive portals.

[0008] FIG. 5 is an exemplary block diagram illustrating distribution of a sequenced instruction set via a geospatial tile data structure and application of the sequenced instruction set by a mobile computing device.

[0009] Corresponding reference characters indicate corresponding parts throughout the drawings.

#### DETAILED DESCRIPTION

[0010] Referring to the figures, embodiments of the disclosure replay sequenced instructions 212 to connect computing devices 202 to networks 108 through network access points (NAPs) 110. In some embodiments, a cloud service 104 crowdsources data sessions (e.g., logs) to learn how users 106 typically navigate through each of the NAPs 110. For example, the cloud service 104 reconstructs a series of events that occurred to connect mobile computing devices 102 through the gated NAPs 110. The mobile computing devices 102 receive the sequenced instructions 212 from the cloud service 104 and replay the received, sequenced instructions 212 on behalf of the users 106 to connect through the NAPs 110. Aspects of the disclosure thus eliminate the challenges, friction, and tediousness encountered when attempting to connect to the networks 108 by enabling connections in an automated and seamless manner.

[0011] The gated NAPs 110 may be described as captive portals with click-through user interfaces. For example, when the user 106 of the mobile computing device 102 accesses the Internet at a hotel, the mobile computing device 102 is redirected to a captive portal web page. The captive portal web page may present advertisements, receive acceptance of terms and/or conditions of use, present confirmation screens, accept text input (e.g., shared keywords for gaining access to the networks 108), accept payment (e.g., credit card number, expiration date, etc.), and the like. The crowdsourced data sessions include, as an example, user actions, over-the-air (OTA) data such as hypertext transfer protocol (HTTP) requests and responses, device hardware context, device software context, any enterprise context, and data describing the NAPs 110.

[0012] Aspects of the disclosure adapt to changing NAPs 110 by receiving updated sets of sequenced instruction 212 from the cloud service 104 without explicit user intervention. Aspects of the disclosure further enable automatic connection to NAPs 110 without user intervention by automatically replaying the appropriate set of sequenced instructions 212 when connecting to one of the NAPs 110. While some embodiments may be described as

implemented by a wireless fidelity (Wi-Fi) assist module executing on the mobile computing devices 102, aspects of the disclosure are operable with any communications protocol. For example, the mobile computing device 102 may automatically connect to a free network through a Wi-Fi hotspot by accepting default terms of service, by accepting  
5 terms of service with values inserted by scripts executing on the mobile computing device 102, by accepting terms of service with text input values (e.g., electronic mail address, telephone number, name, etc.), and/or by providing a shared key (e.g., password). In some embodiments, the mobile computing device 102 guarantees that terms of service associated with the Wi-Fi hotspot are not breached.

10 **[0013]** Referring next to FIG. 1, an exemplary block diagram illustrates the mobile computing devices 102 connecting to the networks 108 through the NAPs 110. In the example of FIG. 1, the user 106 interacts with the mobile computing device 102, or any other computing device 202 of the user 106, to access content 112 via the network 108. The mobile computing device 102 connects to one or more of the networks 108 through one or  
15 more NAPs 110, such as a Wi-Fi beacon and a cellular tower. For each NAP 110, the mobile computing device 102 may perform a series of actions to gain access, via the NAP 110, to the network 108 associated with the NAP 110. After gaining access to the network 108 through the NAP 110, the mobile computing device 102 is able to transmit and receive the content 112. In this manner, the NAP 110 acts as a captive portal.

20 **[0014]** Communication between the mobile computing device 102 and other devices may occur using any protocol or mechanism over any wired or wireless connection. For example, the mobile computing device 102 may attempt to transmit and/or receive data over a Wi-Fi network, a cellular network, a satellite network, a whitespace network, or any network 108. In some embodiments, a NAP 110 is defined to include all NAPs 110 that  
25 share the same SSID within a particular geographic area (e.g., 100-meter radius). For example, a plurality of NAPs 110 within an airport may share the same SSID.

**[0015]** As described further herein, the mobile computing device 102 receives sets of sequenced instructions 212 from the cloud service 104. The sets of sequenced instructions 212 are derived from previous connection attempts to connect to the networks 108 through  
30 the NAPs 110, and generally describe the series of actions to perform to establish a connection through the NAPs 110.

**[0016]** Referring next to FIG. 2, an exemplary block diagram illustrates the computing device 202 storing computer-executable components for replaying the instructions 212 to connect to networks 108 through captive portals. The computing device 202 represents any

device executing instructions (e.g., as application programs, operating system functionality, or both) to implement the operations and functionality described herein. The computing device 202 may include the mobile computing device 102 or any other portable device. In some embodiments, the mobile computing device 102 includes a mobile telephone, laptop, tablet, computing pad, netbook, gaming device, and/or portable media player. The computing device 202 may also include less portable devices such as desktop personal computers, kiosks, tabletop devices, industrial control devices, wireless charging stations, and electric automobile charging stations. Additionally, the computing device 202 may represent a group of processing units or other computing devices.

10 [0017] In some embodiments, the computing device 202 has at least one processor 204 and a memory area 206. The processor 204 includes any quantity of processing units, and is programmed to execute computer-executable instructions for implementing aspects of the disclosure. The instructions may be performed by the processor 204 or by multiple processors executing within the computing device 202, or performed by a processor external  
15 to the computing device 202. In some embodiments, the processor 204 is programmed to execute instructions such as those illustrated in the figures (e.g., FIG. 3).

[0018] In some embodiments, the processor 204 represents an implementation of analog techniques to perform the operations described herein. For example, the operations may be performed by an analog computing device and/or a digital computing device.

20 [0019] The computing device 202 further has one or more computer readable media such as the memory area 206. The memory area 206 includes any quantity of media associated with or accessible by the computing device 202. The memory area 206 may be internal to the computing device 202 (as shown in FIG. 2), external to the computing device 202 (not shown), or both (not shown). In some embodiments, the memory area 206 includes read-  
25 only memory and/or memory wired into an analog computing device.

[0020] The memory area 206 stores, among other data, one or more applications 208. The applications 208, when executed by the processor 204, operate to perform functionality on the computing device 202. Exemplary applications 208 include mail application programs, web browsers, calendar application programs, address book application programs, messaging programs, media applications, location-based services, search programs, and the like. The applications 208 may communicate with counterpart applications or services such as web services accessible via the network 108. For example, the applications 208 may represent downloaded client-side applications that correspond to server-side services executing in a cloud.

[0021] The memory area 206 further stores interaction data 210. The interaction data 210 represents the interactions captured by the computing device 202 when the computing device 202 attempts to connect to the networks 108. For example, the interaction data 210 represents any set of actions, operations, procedures, or other steps undertaken when attempting to establish the connection. In some embodiments, the interaction data 210 is described as manual portal assist data.

[0022] The interaction data 210 may also represent interactions captured by other computing devices 202 and received from those computing devices 202 and/or received from the cloud service 104. In some embodiments, the interaction data 210 includes request and response data from a web browser, such as HTTP request and HTTP response data. However, the interaction data 210 may represent data in any form or protocol. The interaction data 210 may be grouped, for example, by NAP 110, location, and/or protocol.

[0023] One or more sets of sequenced instructions 212 for connecting to the networks 108 are derived from the interaction data 210. For example, the computing device 202 transmits the captured interaction data 210 to the cloud service 104. The cloud service 104 generates the sets of sequenced instructions 212 based on the captured interaction data 210 from the computing device 202 (and from other computing devices 202), and distributes the sets of sequenced instructions 212 to one or more of the computing devices 202.

[0024] The memory area 206 further stores one or more of the sets of sequenced instructions 212 for connecting to captive portals. Each of the instruction sets is associated with at least one of the captive portals, and describes interactions between the computing device 202 and the captive portal to occur to connect through the captive portal. Each instruction set includes a chain of actions from a first request to a last request to identify the user interactions to perform to connect through the captive portal. The chain of actions include, for example, clicking on a hyperlink, executing scripts, completing a form, etc. Each instruction set is capable of being executed, applied, replayed, or otherwise processed by the computing device 202 to connect through the captive portal associated with that instruction set. In some embodiments, the instruction set includes machine-readable instructions, such as extensible markup language (XML) commands. The XML commands include a sequence of verbs for execution to connect through the NAP 110. An example structure of a set of sequenced instructions 212 is shown below:

```
<Script>  
  <Operation xsi:type=Verbname>  
    <Param1> Value1</Param1>
```

```

        <Param2> Value2</Param2>
    </Operation>
    <Operation xsi:type=Verbname>
        ...
5    </Operation>
        ...
    </Script>

```

**[0025]** The instruction sets may be received from the cloud service 104 or another source such as a peer computing device and/or at least one of the NAPs 110. In an example, the NAP 110 provides network access to a geographic area (e.g., to the computing devices 202 within range of the NAP 110). In this example, the cloud service 104 distributes the instruction set for the NAP 110 only to one or more of the computing devices 202 located within the geographic area. The cloud service 104 is able to identify the computing devices 202 located within a geographic area by using location services available on the computing devices 202. In another example, the cloud service 104 receives location information from a plurality of the computing devices 202. Exemplary location information includes a street address, zip code, global positioning system (GPS) coordinates, landmark information, and/or other location or position information. The cloud service 104 selects, based on the received location information, the computing devices 202 to receive the instruction set. For example, the cloud service 104 selects the computing devices 202 that have provided the location information corresponding to a location associated with the NAP 110 corresponding to the instruction set.

**[0026]** In some embodiments, the computing device 202 receives the sets of sequenced instructions 212 via one or more geospatial tile data structures (e.g., “tiles”). Each tile stores the sequenced instruction sets for the NAPs 110 within a particular geographic area (e.g., within one square mile). As an example, each tile stores information such as a service set identifier (SSID) and basic service set identifier (BSSID) of the NAPs 110, the locations of the NAPs 110, and the sequenced instruction sets for connecting to the networks 108 through the NAPs 110. As a further example, the tiles may store connection quality data for each of the networks 108 describing the anticipated, expected, and/or predicted experience of the applications 208 executing on the computing devices 202 when transmitting and/or receiving data over that network 108. One or more of the tiles may be downloaded (e.g., once a day) by the computing devices 202 for caching.



[0027] The memory area 206 further stores one or more computer-executable components. Exemplary components include a tile component 214, a communications interface component 216, an auto-connect component 218, and an enforcement component 220. In some embodiments, the communications interface component 216 includes a  
5 network interface card and/or computer-executable instructions (e.g., a driver) for operating the network interface card. While the communications interface component 216 is operable with short range communication technologies such as near-field communication (NFC), communication between the computing device 202 and other devices may occur using any protocol or mechanism over any wired or wireless connection.

10 [0028] The computing device 202 may further include at least one user interface. In some embodiments, the user interface may include a display (e.g., a touch screen display or natural user interface) and/or computer-executable instructions (e.g., a driver) for operating the display (e.g., displaying data to the user 106 and receiving data from the user 106). For example, the user interface may have a graphics card associated therewith executing the  
15 computer-executable instructions. The user interface may also include one or more of the following to provide data to the user 106 or receive data from the user 106: speakers, a sound card, a camera, a microphone, a vibration motor, one or more accelerometers, a BLUETOOTH brand communication module, GPS hardware, and a photoreceptive light sensor. For example, the user 106 may input commands or manipulate data by moving the  
20 computing device 202 in a particular way.

[0029] Operation of the components, when executed, is described below with reference to FIG. 3.

[0030] Referring next to FIG. 3, an exemplary flow chart illustrates operation of the computing device 202 to obtain and apply an ordered set of instructions 212 to connect  
25 through a captive portal. The operations illustrated in FIG. 3 are described with reference to execution by the computing device 202, such as the mobile computing device 102. However, aspects of the disclosure contemplate execution of the one or more of the operations by other devices or processors, such as by the cloud service 104. For example, the cloud service 104 may identify, to the computing device 202, which instruction set to  
30 apply based which NAP 110 is detected by the computing device 202. Further, while some operations are described with reference to execution of the computer-executable components embodied or otherwise stored in the memory area 206 of the computing device 202, one or more of the operations may be performed by other components or modules of the computing device 202.

[0031] At 302, the computing device 202 receives or detects a request to connect to one of the NAPs 110 acting as a captive portal. For example, one of the applications 208 executing on the computing device 202 may request to send and/or receive data over one of the networks 108. An operating system executing on the computing device 202 receives the request, such as via an application programming interface (API). As another example, the computing device 202 may detect the NAPs 110 within range of (e.g., proximate to) the computing device 202.

[0032] In some embodiments, the computing device 202 detects a plurality of the NAPs 110 and further selects one or more of the detected NAPs 110 for connection. Selecting which NAP 110 to connect through may occur, for example, based on user preference (e.g., explicit selection by the user 106 before or after NAP 110 detection), user history (e.g., whether the user 106 has previously selected or connected to the NAPs 110), input from the NAPs 110, and/or input from the computing device 202 (e.g., device manufacturer, operating system manufacturer, etc.). For example, the computing device 202 may designate one of the NAPs 110 as a preferred NAP 110 based on a configuration of the computing device 202 and/or an account of the user 106 of the computing device 202. If the computing device 202 ultimately connects to the preferred NAP 110, the user 106 may receive priority treatment such as premium content, improved bandwidth speeds, etc.

[0033] At 304, the computing device 202 searches the memory area 206 to find the set of sequenced instructions 212 corresponding to the NAP 110. For example, the computing device 202 searches the memory area 206 based on an SSID of the detected NAP 110. In some embodiments, there may be two instruction sets associated with the detected NAP 110. Each instruction set represents a different path or method for connecting through the same NAP 110. As an example, there may be a free path and a paid path. The free path represents a set of actions for obtaining access to the network 108 without paying a fee. The paid path represents a different set of action for obtaining access to the network 108 through the same NAP 110 as the free path, but by paying a fee. For example, the paid path may be taken by the users 106 who pay a monthly fee to a service provider, or who have purchased a particular amount of time on the network 108 (e.g., hourly pass, daily pass, etc.).

[0034] If the memory area 206 contains the sequenced instruction set for the NAP 110 at 306, the computing device 202 retrieves and applies the stored instruction set to connect through the captive portal at 308. If the memory area 206 does not contain the set of sequenced instructions 212 for the NAP 110 at 306, the computing device 202 requests the set of sequenced instructions 212 for the NAP 110 from the cloud service 104 at 310. The

computing device 202 receives the requested instruction set from the cloud service 104 at 312. The instruction set is derived, by the cloud service 104, from data crowdsourced from a plurality of the computing devices 202. The crowdsourced data describes interactions between those computing devices 202 and the NAP 110. In some embodiments, the computing device 202 requests and receives the sequenced instruction set from a front-end web service that has access to the instruction sets generated by the cloud service 104.

[0035] At 314, the computing device 202 applies the received set of sequenced instructions 212 to the captive portal to establish a connection to the network 108 through the captive portal, or otherwise obtain access to the network 108 via the NAP 110. Applying the instruction set includes, for example, executing each of the machine-readable instructions in the instruction set. In some embodiments, the computing device 202 validates the instructions against a predefined schema. An example validation schema is shown in Appendix A.

[0036] In some embodiments, the set of instructions 212 includes one or more expressions. When applying the instruction set, the computing device 202 replaces the expressions with data particular to the computing device 202 and/or particular to the user 106. For example, the computing device 202 may insert an electronic mail address of the user 106, a telephone number of the user 106, a name of the user 106, an Internet Protocol (IP) address of the user 106, and/or a media access control (MAC) address of the computing device 202.

[0037] Aspects of the disclosure support per-instance authorization of the replacement of personal information of the user 106. For example, in some embodiments, the user 106 selectively decides what data types (e.g., email address, name, telephone number, etc.) are permitted or blocked from being replaced in the expressions. In this example, the user 106 is able to prevent a telephone number from being provided to the captive portal. In another example, the user 106 manually inputs the data (e.g., a work telephone number instead of a home telephone number) for insertion into the expressions. In these examples, the computing device 202 prompts the user 106 to confirm or otherwise authorize the replacement of the expressions with the personal data of the user 106. The user 106 selectively and/or manually provides the authorization, or inputs other data to be inserted, in response of the prompt. In this manner, the user 106 is able to prevent the release of personal information. The computing device 202 further provides the user 106 with a setting enabling the user 106 to configure authorization and/or the data to be inserted into the expressions, to be applied to all sets of instructions 212 for the user 106.

[0038] In some embodiments, the computing device 202 automatically secures the connection to the network 108 when applying the instruction set. For example, securing the connection (e.g., HTTPS data) may occur in response to one of the instructions indicating that the connection should be a secure connection. The instruction set may also indicate  
5 that authentication is needed to connect through the captive portal, and the computing device 202 may then authenticate with the NAP 110 when applying the instruction set.

[0039] After gaining access to the network 108 behind the captive portal, the computing device 202 may further collect report data, such as a telemetry event, describing the connection process. The report data details, for example, whether or not the connection was  
10 successful and whether any problems were encountered during application of the instruction set. If the connection was unsuccessful, the connection report may further indicate at which instruction 212 the connection attempt failed. An example of the collected report data is described in Appendix B.

[0040] The report data may further include performance data describing the quality of the  
15 NAP 110. The performance data may be obtained by the computing device 202 passively and/or actively. Passive operation contemplates observing communications to describe the connection, such as an amount of time the user 106 dwells on the NAP 110 (e.g., dwell time), a quantity of bytes sent and received or other transmission control protocol (TCP) statistics (e.g., upward bandwidth, downward bandwidth, retransmit rate, latency, etc.), and  
20 a disconnect reason (e.g., out of range, manual disconnect, connection failure, etc.). Active operation includes, for example, performing an active test on the network 108 (e.g., a test Voice over Internet Protocol telephone call), and then collecting statistics, such as those associated with passive operation.

[0041] The computing device 202 sends the collected report data to the cloud service 104.  
25 The cloud service 104 analyzes the report data, relating to the NAP 110, collected from each of the computing devices 202 to determine whether any changes should be made to the instruction set for the NAP 110. For example, the cloud service 104 may remove the set of sequenced instructions 212 for the NAP 110 if a success rate from the computing devices 202 falls below a pre-defined threshold. As another example, the cloud service 104 may  
30 throttle data collection (e.g., crowdsource more or less of the interaction data 210) from the computing devices 202. For example, if a success rate for one or more of the instructions falls below a pre-defined threshold (e.g., 95%), the cloud service 104 may send a command to one or more of the computing devices 202 to increase data collection for one or more of the NAPs 110 to accommodate for changes at those NAPs 110. The cloud service 104 may

also send a command to decrease data collection if the cloud service 104 has enough data for a particular NAP 110. In some embodiments, the cloud service 104 sends the command by updating a flag in a geospatial tile to implement the throttling.

[0042] In some embodiments, the cloud service 104 modifies the instruction set for the NAP 110 based on the collected report data. For example, the cloud service 104 may modify a particular instruction in the set of sequenced instructions 212 to reduce the likelihood of failure at that particular instruction for subsequent attempts. The cloud service 104 makes the modified instruction set available to the computing devices 202. For example, the cloud service 104 may provide the modified instruction set to the computing device 202 in response to a request from the computing device 202, or may push or otherwise distribute the modified instruction set to the computing device 202 without first receiving a request from the computing device 202 (e.g., via the user 106 of the computing device 202).

[0043] In some embodiments, the computing device 202 identifies the NAP 110, obtains the instruction set for the NAP 110, and applies the NAP 110 automatically in response to detecting the NAP 110. For example, identifying, obtaining, and/or applying occur without explicit user input.

[0044] In some embodiments, the computer-executable component stored in the memory area 206 of the computing device 202 execute to perform one or more of the operations illustrated in FIG. 3. For example, the tile component 214, when executed by the processor 204, causes the processor 204 to receive (e.g., intermittently) sets of the sequenced instructions 212 from the cloud service 104. The communications interface component 216, when executed by the processor 204, causes the processor 204 to detect at least one of the NAPs 110 acting as a captive portal to the network 108. The auto-connect component 218, when executed by the processor 204, causes the processor 204 to automatically replay the set of sequenced instructions 212 received by the tile component 214 and corresponding to the NAP 110 detected by the communications interface component 216 to obtain access to the network 108 through the captive portal.

[0045] After establishing a connection through the captive portal, the auto-connect component 218 may further receive a single sign-on token, or other data, from the NAP 110. The auto-connect component 218 shares the single sign-on token with one or more other computing devices 202 of the user 106 (e.g., tablet, mobile telephone, gaming console, etc.). In embodiments in which the computing device 202 is a mobile telephone supporting multiple subscriber identity modules (SIMs), the auto-connect component 218 shares the single sign-on token with each of the SIMs. Alternatively or in addition, the auto-connect

component 218 may implement a captive portal on the computing device 202 to gate access to the established NAP 110 connection by the other computing devices 202 of the user 106. In this manner, the other computing devices 202 of the user 106 connect to the network 108 through the computing device 202 that applied the instruction set to establish the connection.

5 [0046] The auto-connect component 218 may also identify and provide, to the user 106, terms of service associated with the NAP 110. The terms of service may be described by data included in the instruction set for the NAP 110, or may be obtained through other means (e.g., published by a web service). The terms of service may specify that video streaming is not allowed through the NAP 110, or that use of particular communication port numbers is prohibited. The terms of service may also specify bandwidth limitations, such as a maximum cumulative bandwidth and/or a peak bandwidth. In some embodiments, the auto-connect component 218 generates a summary of the terms of service or otherwise simplifies the terms of service. For example, the auto-connect component 218 generates an easy-to-view listing of the connected NAPs 110, popular services or applications, and whether each item in the list is allowed or not allowed. The auto-connect component 218 may also detect the applications 208 executing on the computing device 202 that are allowed to transmit through the NAP 110, and identify those applications 208 to the user 106. The summary may be derived from data associated with the set of sequenced instructions 212 received by the tile component 214 and corresponding to the NAP 110 detected by the communications interface component 216. The summary may be displayed in a user interface element (e.g., a toast or popup), via audio, and/or via a message sent through a channel such as a short message service (e.g., a text message).

15 [0047] The enforcement component 220, when executed by the processor 204, causes the processor 204 to selectively allow one or more of the applications 208 to transmit and/or receive data on the network 108 after a connection has been established to the network 108 (e.g., after the auto-connect component 218 obtains access). For example, the enforcement component 220 executes to enforce the terms of service associated with the NAP 110.

20 [0048] Referring next to FIG. 4, an exemplary block diagram illustrates the cloud service 104 storing sets of sequenced instructions 212 corresponding to captive portals. In the example of FIG. 4, the cloud service 104 has storage 402 for the interaction data 404 for a plurality of the captive portals, such as for captive portal #1 through captive portal #N. The cloud service 104 generates the sequenced instructions 212 from the interaction data 404 for the captive portals. As a result, the storage 402 has correlations, associations, and/or correspondences between the interaction data 404 for each of the captive portals and the

sequenced instructions 212 for that captive portal. The sequenced instructions 212 are made available to the computing devices 202 on-demand, via a push service, or any other distribution means or configuration.

5 [0049] Referring next to FIG. 5, an exemplary block diagram illustrates distribution of the sequenced instruction set via a geospatial tile data structure and application of the sequenced instruction set by the mobile computing device 102. While a particular sequence of operations is illustrated in FIG. 5, other sequences are within the scope of the disclosure.

10 [0050] In the example of FIG. 5, the interaction data 404 crowdsourced by the cloud service 104 is processed by an instruction set generator 502 to produce at least one of the sequenced instruction sets. The sequenced instruction set is distributed to the mobile computing devices 102 by, for example, a tile module 504. The mobile computing device 102 replays the sequenced instruction set from the tile when connecting to a Wi-Fi portal gated hotspot 506 to connect to the Internet, or other network 108.

#### Additional Examples

15 [0051] In some scenarios, the computing device 202 obtains and presents localized content associated with the NAP 110. Localized content includes, for example, advertisements, public service announcements, and the like.

20 [0052] An example language set for the sequenced instructions 212 is next described. Aspects of the disclosure, however, are operable with any set of verbs, operands, parameters, etc. Exemplary verbs include Probe for Connectivity (PC), Follow Redirect (FR), Follow Anchor (FA), Submit Form (SF), Navigate To (NT), and Submit Post String (SPS).

25 [0053] The PC verb may be the first verb in the instruction set, and probes for network connectivity by attempting to connect to a network location (e.g., a HTTP address). The FR verb follows the redirect from a previous browser page and retrieves a redirected browser page. Exemplary parameters to this verb include delay (e.g., a time delay before executing this verb), a redirect type (e.g., HttpRedirect, HtmlMetaRefresh, JavaScriptRefresh, etc.), and a timeout (e.g., a timeout for receiving a response from the web server). In XML, the FR verb may appear as shown below:

```
30 <Operation xsi:type="FollowRedirect">
    <Delay>0</Delay>
    <RedirectType>1</RedirectType>
    <Timeout>5000</Timeout>
</Operation>
```

[0054] The FA verb follows an anchor from a previous page and retrieves a target page identified by the anchor. Exemplary parameters to this verb include delay (e.g., a time delay before executing this verb), an anchor identifier (e.g., a zero-based index to the anchor on the previous page), and a timeout (e.g., a timeout for receiving a response from the web server). In XML, the FA verb may appear as shown below:

```

5   <Operation xsi:type="FollowAnchor">
      <Delay>0</Delay>
      <AnchorId>1</AnchorId>
      <Timeout>5000</Timeout>
10  </Operation>

```

[0055] The SF verb submits a form on a previous page and retrieves the response. The SF verb looks at the action attribute in the form to identify the post uniform resource location (URL). Exemplary parameters to this verb include delay (e.g., a time delay before executing this verb), a form identifier (e.g., a zero-based index to the forms on the previous page), a submit option (e.g., a flag to indicate how to obtain contents for the post request), one or more key-value pairs (e.g., a set of key-value pairs to include in the post request), and a timeout (e.g., a timeout for receiving a response from the web server). The submit option parameter may appear as shown below:

```

20   public enum FormFieldSubmitOption
      {
          None,
          UseDefaultValues,
          EmptyValues,
25   OverrideValues,
          SpecifiedPairsOnly,
      }

```

[0056] Exemplary key-value pairs include one or more of the following predefined expressions which are replaced by the mobile computing device 102: %WPEmail% replaced with an email address of the user 106, %WPPhone% replaced with a telephone number of the user 106, %WPFIRSTNAME% replaced with a first name of the user 106, %WPLASTNAME% replaced with a last name of the user 106, %WPNAME% replaced with a complete name of the user 106, %WPIPADRESS% replaced with an IP address of the mobile



computing device 102, and %WPMacAddress% replaced with a media access control (MAC) address of the mobile computing device 102.

[0057] In XML, the SF verb may appear as shown below:

```

5      <Operation xsi:type="SubmitForm">
      <Delay>0</Delay>
      <FormId>0</FormId>
      <SubmitOption>1</SubmitOption>
      <KeyValuePairs>
      <KeyValuePair key="name" value="Anna"/>
10     <KeyValuePair key="accessCode"
value="A973BEEBD3BE78DD"/>
      </KeyValuePairs>
      <Timeout>5000</Timeout>
      </Operation>

```

15

[0058] The NT verb retrieves the contents of the target page. Exemplary parameters to this verb include delay (e.g., a time delay before executing this verb), address (e.g., the destination URL), and timeout (e.g., a timeout for receiving a response from the web server). The address parameter may contain one or more predefined expressions in the query string, which are replaced by the mobile computing device 102 with the respective values. In XML, the NT verb may appear as shown below:

```

      <Operation xsi:type="NavigateTo">
      <Delay>0</Delay>
      <Address>http://www...com</Address>
25     <Timeout>5000</Timeout>
      </Operation>

```

[0059] The SPS verb is used to submit the specified post string to the specified address. Exemplary parameters to this verb include delay (e.g., a time delay before executing this verb), address (e.g., the destination URL), one or more key-value pairs (e.g., a set of key-value pairs to include in the post request), and a timeout (e.g., a timeout for receiving a response from the web server). The address parameter may contain one or more predefined expressions in the query string, which are replaced by the mobile computing device 102 with the respective values. In XML, the SPS verb may appear as shown below:

```

      <Operation xsi:type="SubmitPostString">

```

```

    <Delay>0</Delay>
    <Address>http://www...com</Address>
    <KeyValuePairs>
      <KeyValuePair key="name" value="Anna"/>
5      <KeyValuePair key="accessCode"
value="A973BEABD3BE45ED" encrypted="true"/>
    </KeyValuePairs>
    <Timeout>5000</Timeout>
  </Operation>

```

10

**[0060]** At least a portion of the functionality of the various elements in the figures may be performed by other elements in the figures, or an entity (e.g., processor, web service, server, application program, computing device, etc.) not shown in the figures.

15

**[0061]** In some embodiments, the operations illustrated in the figures may be implemented as software instructions encoded on a computer readable medium, in hardware programmed or designed to perform the operations, or both. For example, aspects of the disclosure may be implemented as a system on a chip or other circuitry including a plurality of interconnected, electrically conductive elements.

20

**[0062]** The term “Wi-Fi” as used herein refers, in some embodiments, to a wireless local area network using high frequency radio signals for the transmission of data. The term “BLUETOOTH” as used herein refers, in some embodiments, to a wireless technology standard for exchanging data over short distances using short wavelength radio transmission. The term “NFC” as used herein refers, in some embodiments, to a short-range high frequency wireless communication technology for the exchange of data over short

25

**[0063]** Embodiments have been described with reference to data monitored and/or collected from the users 106. In some embodiments, notice may be provided to the users 106 of the collection of the data (e.g., via a dialog box or preference setting) and users 106 are given the opportunity to give or deny consent for the monitoring and/or collection. The

30

**[0064]** Further, the user 106 may opt-out of replaying the sequenced instructions 212 for particular NAPs 110. For example, the user 106 may designate the NAPs 110 by name (e.g., SSID), location (e.g., metropolitan service area, address, etc.), venue (e.g., retail store or restaurant), network, etc.

Exemplary Operating Environment

[0065] Exemplary computer readable media include flash memory drives, digital versatile discs (DVDs), compact discs (CDs), floppy disks, and tape cassettes. By way of example and not limitation, computer readable media comprise computer storage media and communication media. Computer storage media include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media are tangible and mutually exclusive to communication media. In some embodiments, computer storage media are implemented in hardware. Exemplary computer storage media include hard disks, flash drives, and other solid-state memory. In contrast, communication media embody computer readable instructions, data structures, program modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism and include any information delivery media.

[0066] Although described in connection with an exemplary computing system environment, embodiments of the disclosure are capable of implementation with numerous other general purpose or special purpose computing system environments, configurations, or devices.

[0067] Examples of well-known computing systems, environments, and/or configurations that may be suitable for use with aspects of the invention include, but are not limited to, mobile computing devices, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, gaming consoles, microprocessor-based systems, set top boxes, programmable consumer electronics, mobile telephones, mobile computing and/or communication devices in wearable or accessory form factors (e.g., watches, glasses, headsets, or earphones), network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like. Such systems or devices may accept input from the user in any way, including from input devices such as a keyboard or pointing device, via gesture input, proximity input (such as by hovering), and/or via voice input.

[0068] Embodiments of the invention may be described in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices in software, firmware, hardware, or a combination thereof. The computer-executable instructions may be organized into one or more computer-executable components or modules. Generally, program modules include, but are not limited to, routines, programs, objects, components, and data structures that perform particular tasks or

implement particular abstract data types. Aspects of the invention may be implemented with any number and organization of such components or modules. For example, aspects of the invention are not limited to the specific computer-executable instructions or the specific components or modules illustrated in the figures and described herein. Other embodiments of the invention may include different computer-executable instructions or components having more or less functionality than illustrated and described herein.

5 [0069] Aspects of the invention transform a general-purpose computer into a special-purpose computing device when configured to execute the instructions described herein.

[0070] The embodiments illustrated and described herein as well as embodiments not specifically described herein but within the scope of aspects of the invention constitute exemplary means for automatically connecting to one of the networks 108 through a captive portal based on crowdsourced instructions. For example, one or more of the embodiments contemplate means for detecting one of the captive portals within range of the mobile computing device 102, identifying one of the instruction sets associated with the detected captive portal, and applying the identified instruction set to the detected captive portal to obtain access to the network 108.

10 [0071] The order of execution or performance of the operations in embodiments of the invention illustrated and described herein is not essential, unless otherwise specified. That is, the operations may be performed in any order, unless otherwise specified, and embodiments of the invention may include additional or fewer operations than those disclosed herein. For example, it is contemplated that executing or performing a particular operation before, contemporaneously with, or after another operation is within the scope of aspects of the invention.

20 [0072] When introducing elements of aspects of the invention or the embodiments thereof, the articles "a," "an," "the," and "said" are intended to mean that there are one or more of the elements. The terms "comprising," "including," and "having" are intended to be inclusive and mean that there may be additional elements other than the listed elements. The term "exemplary" is intended to mean "an example of." The phrase "one or more of the following: A, B, and C" means "at least one of A and/or at least one of B and/or at least one of C."

30 [0073] Having described aspects of the invention in detail, it will be apparent that modifications and variations are possible without departing from the scope of aspects of the invention as defined in the appended claims. As various changes could be made in the above constructions, products, and methods without departing from the scope of aspects of the

invention, it is intended that all matter contained in the above description and shown in the accompanying drawings shall be interpreted as illustrative and not in a limiting sense.

#### Appendix A

5 [0074] An example of a validation schema for validating the sequenced instructions 212 to be applied by the computing device 202 is shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
10   <xs:complexType name="OperationType" abstract="true"/>

   <xs:complexType name="ProbeForConnectivity">
     <xs:complexContent>
       <xs:extension base="OperationType"/>
15   </xs:complexContent>
</xs:complexType>

   <xs:complexType name="FollowLink">
     <xs:complexContent>
20   <xs:extension base="OperationType">
       <xs:sequence>
         <xs:element type="xs:int" name="Delay"/>
         <xs:element type="xs:int" name="LinkId"/>
         <xs:element type="xs:int" name="Timeout"/>
25   </xs:sequence>
       </xs:extension>
     </xs:complexContent>
   </xs:complexType>

   <xs:complexType name="SubmitForm">
30   <xs:complexContent>
     <xs:extension base="OperationType">
       <xs:sequence>
         <xs:element type="xs:int" name="Delay"/>
35   <xs:element type="xs:int" name="FormId"/>
         <xs:element type="xs:int" name="SubmitOption"/>

```

```

        <xs:element name="KeyValuePairs" minOccurs="0"
maxOccurs="1">
            <xs:complexType>
                <xs:sequence>
5                 <xs:element name="KeyValuePair" minOccurs="1"
maxOccurs="unbounded" >
                    <xs:complexType>
                        <xs:attribute name="key" type="xs:string"
use="required" />
10                     <xs:attribute name="value" type="xs:string"
use="required" />
                    </xs:complexType>
                </xs:element>
                </xs:sequence>
15            </xs:complexType>
            </xs:element>
            <xs:element type="xs:int" name="Timeout"/>
        </xs:sequence>
    </xs:extension>
20 </xs:complexContent>
</xs:complexType>

<xs:complexType name="FollowRedirect">
    <xs:complexContent>
25     <xs:extension base="OperationType">
        <xs:sequence>
            <xs:element type="xs:int" name="Delay"/>
            <xs:element type="xs:int" name="RedirectType"/>
            <xs:element type="xs:int" name="Timeout"/>
30         </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

35 <xs:complexType name="NavigateTo">
    <xs:complexContent>
        <xs:extension base="OperationType">

```

```

    <xs:sequence>
      <xs:element type="xs:int" name="Delay"/>
      <xs:element type="xs:string" name="Address"/>
      <xs:element type="xs:int" name="Timeout"/>
5    </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

10 <xs:complexType name="SubmitPostString">
  <xs:complexContent>
    <xs:extension base="OperationType">
      <xs:sequence>
        <xs:element type="xs:int" name="Delay"/>
15        <xs:element type="xs:string" name="Address"/>
        <xs:element name="KeyValuePairs" minOccurs="1"
maxOccurs="1">
          <xs:complexType>
            <xs:sequence>
20              <xs:element name="KeyValuePair" minOccurs="1"
maxOccurs="unbounded" >
                <xs:complexType>
                  <xs:attribute name="key" type="xs:string"
use="required" />
25                  <xs:attribute name="value" type="xs:string"
use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:element type="xs:int" name="Timeout"/>
    </xs:sequence>
  </xs:extension>
35 </xs:complexContent>
</xs:complexType>
```

```

    <xs:element name="Script">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Operation" type="OperationType"
5 maxOccurs="unbounded" minOccurs="1"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>

```

10

### Appendix B

[0075] An example of a telemetry event collected by the cloud service 104 from the mobile computing devices 102 is next described. The telemetry event indicates whether or not the connection attempt was successful. As shown in the example code below, after the connection attempt, the mobile computing device 102 sends the telemetry event containing, for example, the BSSID, SSID, device identifier, instruction identifiers, and a flag indicating success (e.g., "True") or failure (e.g., "False") of the connection attempt. If the connection was successful, the flag indicates success. If the connection failed yet the instructions 212 were replayed or executed properly, the flag indicates failure. If the mobile computing device 102 was unable to replay each of the instructions 212 properly, the flag indicates failure and the mobile computing device 102 updates the telemetry event to further identify which operation failed, the reason for the failure, and a session identifier.

```

/// <summary>
  /// WiFi Portal Assist Telemetry Event
25 /// </summary>
  public sealed class WFPATelemetry
  {
    /// <summary>
    /// Bssid of the access point
30 /// </summary>
    [DataMember(Name = "bssid")]
    public string Bssid { get; set; }

    /// <summary>
35 /// Ssid of the access point

```



```
    /// </summary>
    [DataMember(Name = "ssid")]
    public string Ssid { get; set; }

5    /// <summary>
    /// Device Id
    /// </summary>
    [DataMember(Name = "dvc")]
    public string DeviceId { get; set; }

10    /// <summary>
    /// Instruction Id
    /// </summary>
    [DataMember(Name = "inid")]
15    public int InstructionId { get; set; }

    /// <summary>
    /// Flag to indicate success
    /// </summary>
20    [DataMember(Name = "succ")]
    public bool Success { get; set; }

    /// <summary>
    /// Zero based index to indicate the failed
25    operation.
    /// </summary>
    [OptionalField]
    [DataMember(Name = "opid")]
    public int OperationIndex { get; set; }

30    /// <summary>
    /// Failure reason code
    /// </summary>
    [OptionalField]
```

```
[DataMember(Name = "reas")]
public FailureReason reason { get; set; }

    /// <summary>
5    /// WebSession
    /// </summary>
    [OptionalField]
    [DataMember(Name = "sess")]
    public IEnumerable<RequestResponsePairs> session {
10 get; set; }
    }

    /// <summary>
15    /// The Response structure
    /// </summary>
    public class Response
    {
        /// <summary>
20    /// The response code i.e. 200, 302, 404 etc.
        /// </summary>
        [DataMember(Name = "stat")]
        public int Status { get; set; }

25    /// <summary>
        /// The collection containing the header keyvalue
        pairs
        /// </summary>
        [DataMember(Name = "head")]
30    public IEnumerable<Header> Headers { get; set; }

        /// <summary>
        /// The Content part of the response
        /// </summary>
```

```
        [DataMember(Name = "cont")]
        public string Content { get; set; }
    }

5    /// <summary>
    /// The Request structure
    /// </summary>
    public class Request
    {
10        /// <summary>
        /// The http Method being used i.e. GET, POST.
        /// </summary>
        [DataMember(Name = "meth")]
        public string Method { get; set; }

15        /// <summary>
        /// The target URL
        /// </summary>
        [DataMember(Name = "url")]
20        public string Url { get; set; }

        /// <summary>
        /// The Content part of the request
        /// </summary>
25        [DataMember(Name = "cont")]
        public string Content { get; set; }
    }

    /// <summary>
30    /// Failure Reasons
    /// </summary>
    public enum FailureReason
    {
        Unknown,
```

```
Timeout,  
UnexpectedResponse,  
Disconnected  
}
```

## CLAIMS

1. A system for connecting to a network through a captive portal based on crowdsourced instructions, said system comprising:

a memory area associated with a mobile computing device, said memory area storing a plurality of instruction sets for connecting to captive portals, each of the plurality of instruction sets being associated with one of the captive portals, the plurality of instruction sets being received from a cloud service; and

a processor of the mobile computing device programmed to:

detect one of the captive portals within range of the mobile computing device;

identify, from the memory area, one of the plurality of instruction sets associated with the detected one of the captive portals;

apply the identified one of the plurality of instruction sets to the detected one of the captive portals to obtain access to the network; and

establish a connection to the network through the detected one of the captive portals after applying the identified one of the plurality of instruction sets.

2. The system of claim 1, wherein the processor is programmed to identify the one of the plurality of instruction sets by searching the memory area based on a service set identifier (SSID) of the detected one of the captive portals.

3. The system of claim 1, wherein the identified one of the plurality of instruction sets comprises machine-readable instructions, and wherein the processor is programmed to apply the identified one of the plurality of instruction sets by executing each of the machine-readable instructions.

4. The system of claim 1, wherein the processor is programmed to apply the identified one of the plurality of instruction sets without input from a user of the mobile computing device.

5. The system of claim 1, wherein the processor is programmed to identify the one of the plurality of instruction sets and said apply the identified one of the plurality of instruction sets automatically in response to detecting the one of the captive portals, and wherein each of the plurality of instruction sets describes interactions between the mobile computing device and one of the captive portals.

6. A method comprising:

detecting, by a computing device, a network access point, the network access point acting as a captive portal to a network;

requesting, from a cloud service, a sequenced instruction set associated with the detected network access point;

receiving the requested sequenced instruction set, the received sequenced instruction set being derived from crowdsourced data describing interactions between mobile computing devices and the detected network access point;

applying the received sequenced instruction set to the captive portal to obtain access to the network via the network access point; and

establishing a connection to the network via the network access point after applying the received sequenced instruction set.

7. The method of claim 6, further comprising:

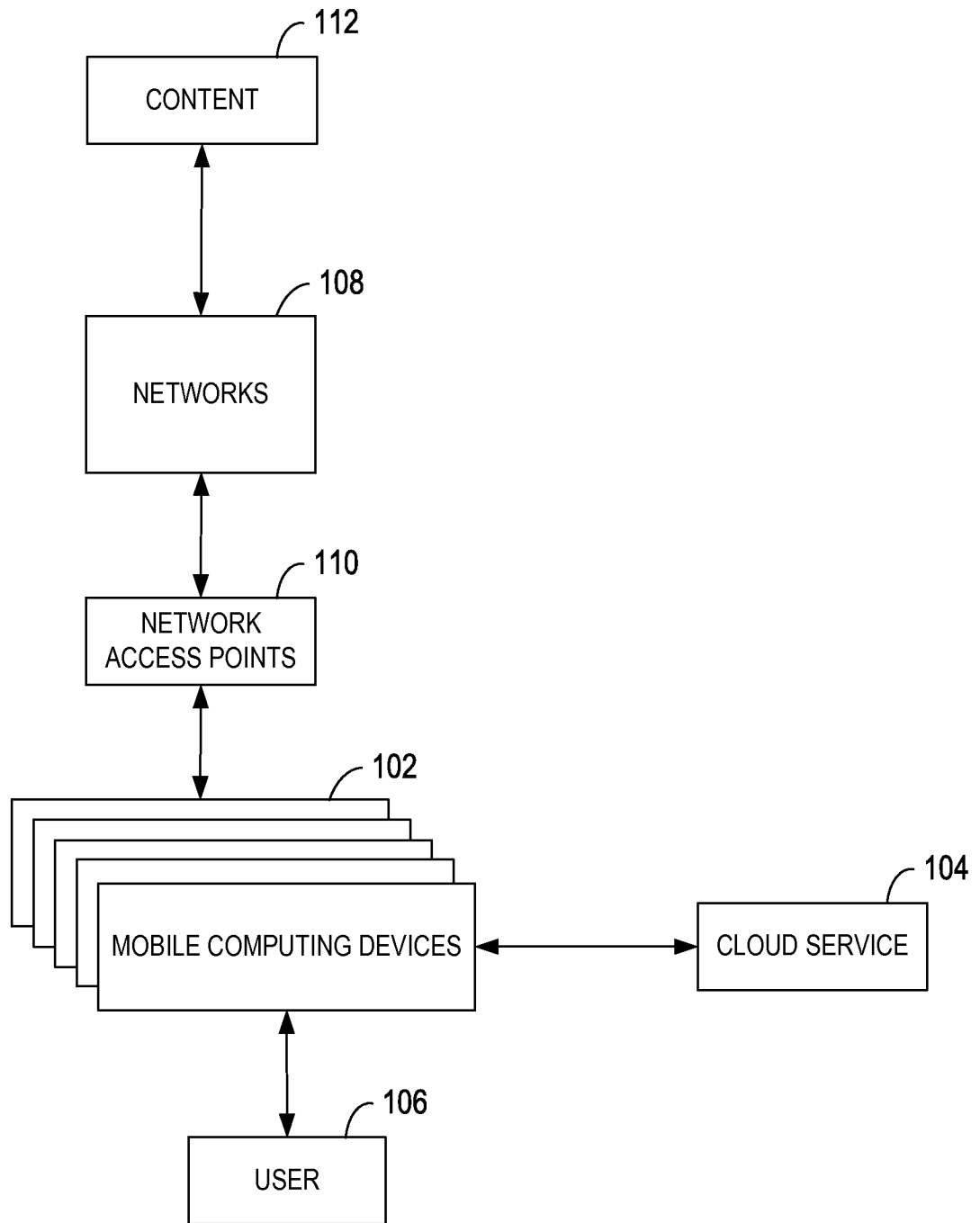
collecting report data describing application of the received sequenced instruction set to the captive portal; and

sending the collected report data to the cloud service, the cloud service modifying the sequenced instruction set based on the collected report data.

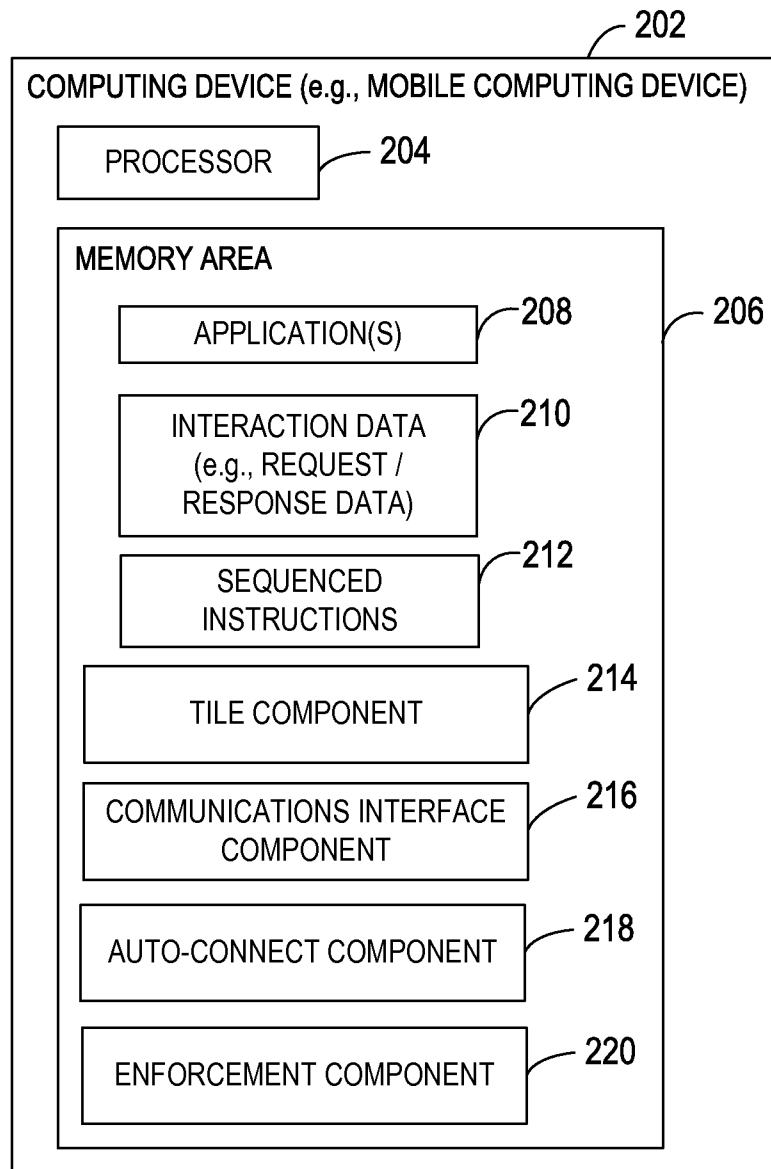
8. The method of claim 6, further comprising receiving an updated version of the sequenced instruction set from the cloud service without input from a user of the computing device, wherein applying the received sequenced instruction set comprises establishing a connection to the network, and further comprising automatically securing the connection based on an instructions in the received sequenced instruction set.

9. The method of claim 6, wherein the received sequenced instruction set has one or more expressions therein, wherein applying the received sequenced instruction set comprises replacing the one or more expressions with data particular to the computing device and/or particular to a user of the computing device, and wherein the data comprises at least one of the following: an electronic mail address of the user, a telephone number of the user, a name of the user, an Internet Protocol (IP) address of the user, or a media access control (MAC) address of the computing device.

10. The method of claim 6, wherein detecting the network access point comprises detecting a plurality of the network access points, and further comprising designating one of the plurality of network access points as a preferred network access point based on a configuration of the computing device and/or an account of a user of the computing device.



*FIG. 1*



**FIG. 2**



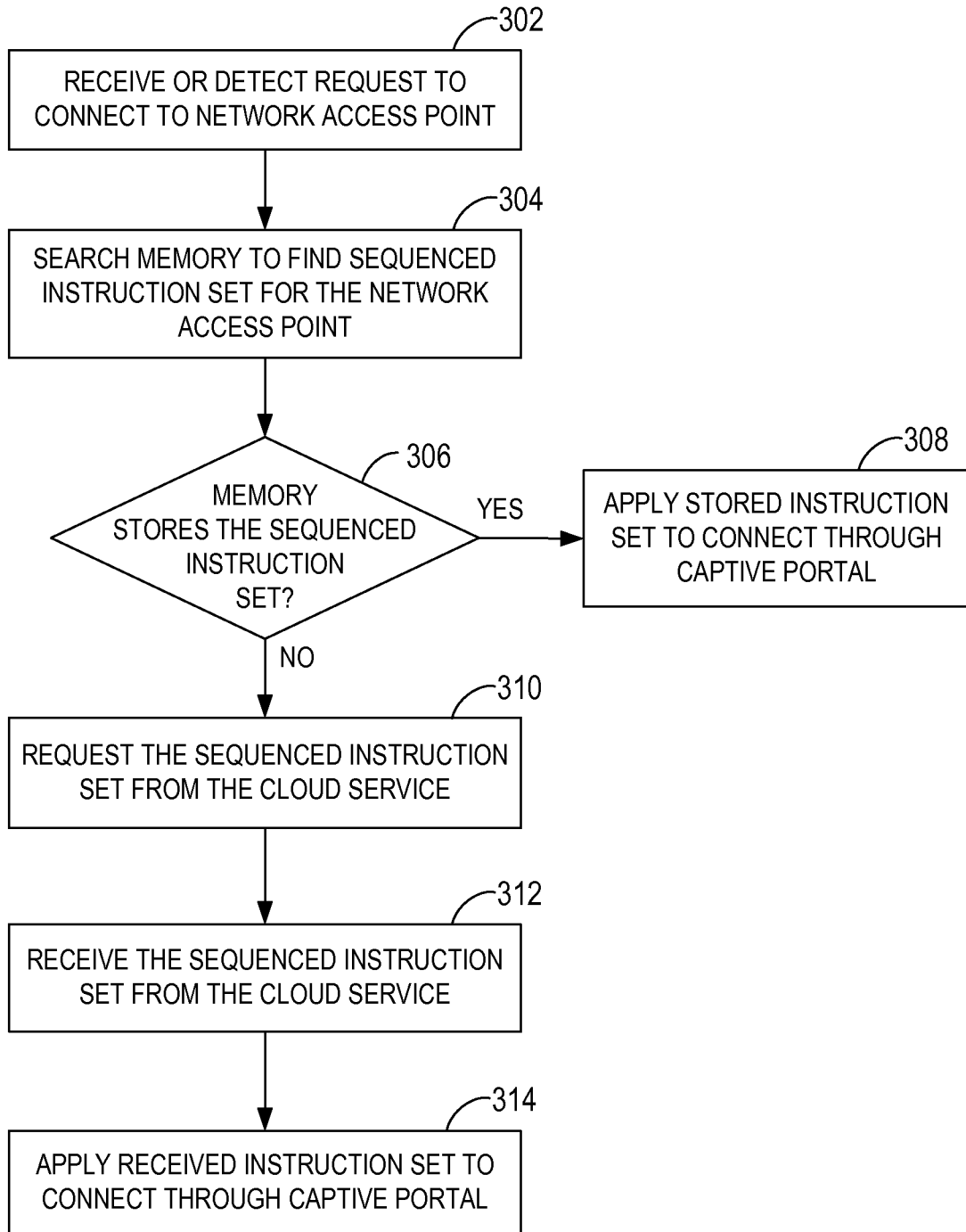


FIG. 3

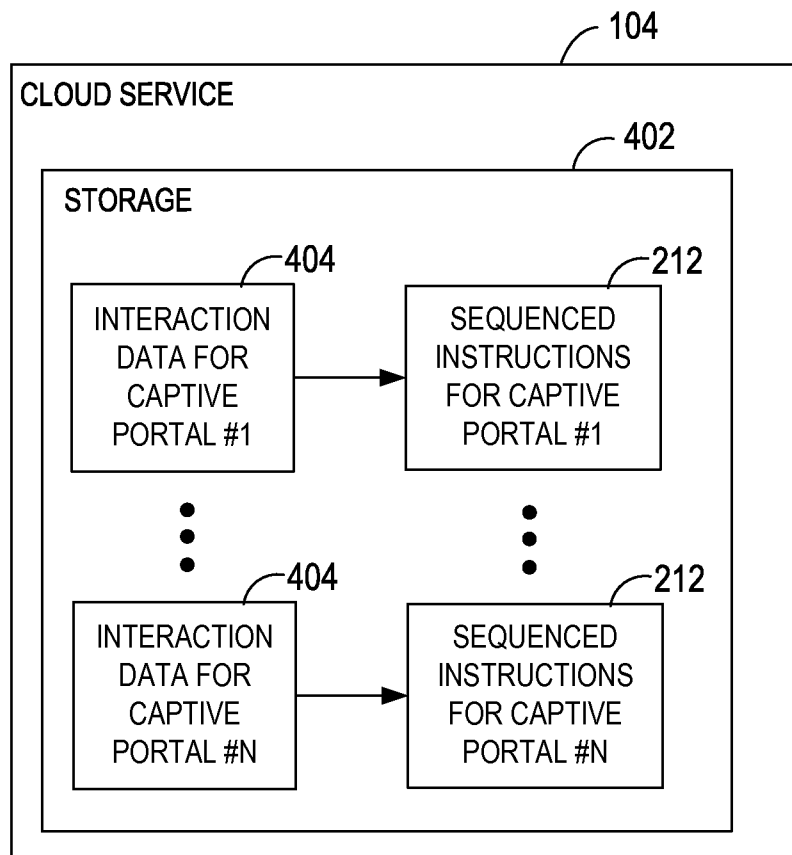
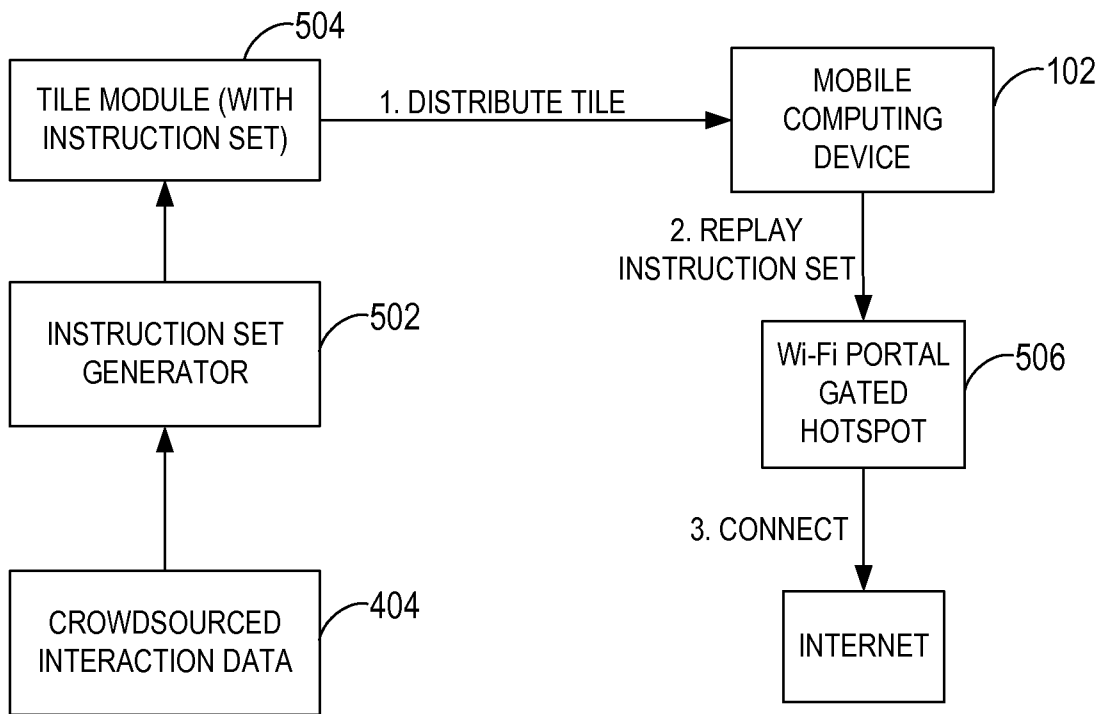


FIG. 4



*FIG. 5*

**INTERNATIONAL SEARCH REPORT**

International application No PCT/US2014/065582
---

**A. CLASSIFICATION OF SUBJECT MATTER**  
 INV. H04W36/36 H04W12/06  
 ADD. H04W84/12 H04L12/28

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**  
 Minimum documentation searched (classification system followed by classification symbols)  
 H04W H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)  
 EPO-Internal, WPI Data

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2013/095451 A1 (INTEL CORP [US]; ZHU JING [US]; AGERSTAM MATS [US]) 27 June 2013 (2013-06-27) paragraphs [0015], [0020] - [0022], [0036], [0040], [0054]; figures 4,5 -----	1-10
X	US 2012/196644 A1 (SCHERZER SHIMON [IL] ET AL) 2 August 2012 (2012-08-02) paragraphs [0059], [0076] ----- -/--	1-10

Further documents are listed in the continuation of Box C.       See patent family annex.

\* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E" earlier application or patent but published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search  6 March 2015	Date of mailing of the international search report  17/03/2015
---	--

Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer  Domingos, Luis
--	--

**INTERNATIONAL SEARCH REPORT**

International application No PCT/US2014/065582
---

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>GERMAN CASTIGNANI ET AL: "Wi2Me: A Mobile Sensing Platform for Wireless Heterogeneous Networks", 32ND IEEE INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS WORKSHOPS, ICDCSW 2012, IEEE, 18 June 2012 (2012-06-18), pages 108-113, XP032217936, DOI: 10.1109/ICDCSW.2012.36 ISBN: 978-1-4673-1423-7 section I. INTRODUCTION in section III. WI2ME TRACES EXPLORER: A. traces C. Automatic Authentication -----</p>	1-10
A	<p>US 2013/155876 A1 (POTRA ADRIAN [US] ET AL) 20 June 2013 (2013-06-20) paragraphs [0024], [0040] - [0050]; figures 4,5 -----</p>	1-10
A	<p>US 2013/115945 A1 (HOLOSTOV VLADIMIR [US] ET AL) 9 May 2013 (2013-05-09) paragraphs [0016] - [0018], [0042] -----</p>	1-10

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No PCT/US2014/065582
---

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2013095451 A1	27-06-2013	US 2013326587 A1 WO 2013095451 A1	05-12-2013 27-06-2013
-----			
US 2012196644 A1	02-08-2012	US 2012196644 A1 US 2014080539 A1	02-08-2012 20-03-2014
-----			
US 2013155876 A1	20-06-2013	AU 2012355532 A1 CA 2859735 A1 CN 104094640 A EP 2795969 A1 JP 2015502125 A KR 20140109444 A US 2013155876 A1 WO 2013096146 A1	24-07-2014 27-06-2013 08-10-2014 29-10-2014 19-01-2015 15-09-2014 20-06-2013 27-06-2013
-----			
US 2013115945 A1	09-05-2013	US 2013115945 A1 US 2014162638 A1	09-05-2013 12-06-2014
-----			