

# Technical Disclosure Commons

---

Defensive Publications Series

---

September 17, 2019

## Providing online information via a virtual assistant

Viraj Mahesh

Follow this and additional works at: [https://www.tdcommons.org/dpubs\\_series](https://www.tdcommons.org/dpubs_series)

---

### Recommended Citation

Mahesh, Viraj, "Providing online information via a virtual assistant", Technical Disclosure Commons, (September 17, 2019)  
[https://www.tdcommons.org/dpubs\\_series/2489](https://www.tdcommons.org/dpubs_series/2489)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

## **Providing online information via a virtual assistant**

### **ABSTRACT**

This disclosure describes techniques for a virtual assistant to provide a response to user queries when the requested information is not available directly via an application programming interface (API) but is available via a website. When a user issues a query for such information, a cloud platform accesses the website, e.g., using an automated web browser, with parameters determined based on the query. The cloud platform parses the resultant webpage to determine the answer to the query and provides the answer to the virtual assistant, e.g., which delivers the response via a voice interface.

### **KEYWORDS**

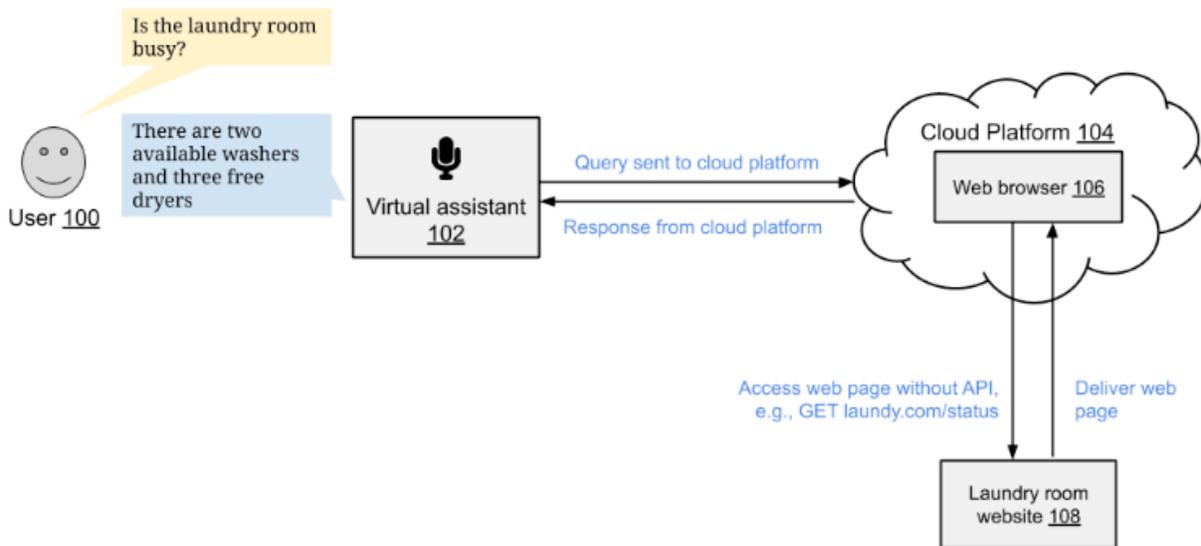
- Virtual assistant
- Voice assistant
- Browser automation
- Third party integration
- Data extraction
- Smart speaker
- Smart appliance

### **BACKGROUND**

Users routinely use virtual assistant applications, e.g., via a smart speaker or other device, to query for information, e.g., by issuing a voice query. Many virtual assistant platforms provide integration and development platforms that enable third-party developers to extend the virtual assistant application by building custom integrations, e.g., by calling application programming interfaces (APIs). However, many sources of information exist that do not provide APIs that can

be directly queried. Thus, obtaining and providing information from these sources in response to a user query is difficult. For example, such sources can include websites that dynamically render information, e.g., using JavaScript or other programming language. This further complicates extracting data from the website.

## DESCRIPTION



**Fig. 1: Use of automated browser to extract and provide information via a virtual assistant**

Fig. 1 illustrates an example scenario in which a user provides a query to a virtual assistant (102). For example, the query “is the laundry room busy?” is regarding the state of laundry equipment in a laundry room of an apartment building that the user resides in. The status of laundry equipment, e.g., washers and dryers, may be available via a website.

The query is sent to a cloud platform, e.g., a server that implements the backend for the virtual assistant application. The cloud platform (104) uses an automated browser (106) to access the laundry room website (108). With user permission, user credentials of the user for the website are to access information on the website. For example, a web page that includes state information for equipment in the laundry room is obtained by the automated web browser and

parsed by the cloud platform to extract information. The cloud platform provides the extracted information data to the virtual assistant for a response to a user query.

In this illustrated example, a resident can use the virtual assistant to determine whether a washer or dryer is available even when the information source, e.g., the laundry website does not provide an API. Similar techniques can be used for websites or other information sources that do not offer API access.

In this manner, the described techniques enable a virtual assistant application to access and provide users answers to queries for which the data is not accessible via an API. The virtual assistant application can also provide the automated browser to third-party developers to enable the third-party developers to build integrations to answer user queries.

The described techniques are implemented with specific user permission, e.g., to identify the website based on user's address or other information, and to access information from the website (e.g., by the use of user credentials. If the user denies or restrict permission, only such information that the user permits is utilized to access websites. Users are provided with options to select the information that the virtual assistant can utilize to automatically access websites, or to turn off the feature entirely.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user's social network, social actions or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can

be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

## CONCLUSION

This disclosure describes techniques for a virtual assistant to provide a response to user queries when the requested information is not available directly via an application programming interface (API) but is available via a website. When a user issues a query for such information, a cloud platform accesses the website, e.g., using an automated web browser, with parameters determined based on the query. The cloud platform parses the resultant webpage to determine the answer to the query and provides the answer to the virtual assistant, e.g., which delivers the response via a voice interface.