November 30, 2017

Smart Kitchen Application Programming Interface Suite

Sean Blaes

Smart Kitchen Application Programming Interface Suite

Abstract:

Home kitchens are rapidly becoming more and more automated. Many appliances, such as ranges, ovens, and slow cookers, are wi-fi capable and can be individually controlled, as part of the Internet of Things (IoT), using a dedicated application on a remote device such as a smart phone. A suite of one or more application programming interfaces (API) is described that serves to automate an entire smart kitchen based on a voice-recognizing personal assistant, including use of cooking information and recipes available from applications or cloud-based services.

Keywords:

Internet-of-Things (IoT), home automation, smart appliances, smart kitchen, recipes

Background:

Today, with advancements in communication technologies and with computing/sensing electronics embedded in a myriad of devices, the ability for devices to collect and exchange data with one another is escalating. Devices such as smart phones, voice-recognizing personal assistants, computers, automobiles, home entertainment systems/appliances, and the like, are able to communicate with one another either directly, in a machine-to-machine environment, or indirectly over a network. Such communications and exchange of data across the myriad of devices is commonly referred to as the Internet-of-Things (IoT). The communications and exchange of data can have purposes that include, for example, collecting usage data for vendor analytics, remote initiation/shut-down of an operating system, automating a home environment, monitoring a person's health, and so forth.

A view of an example IoT environment is represented in Fig. 1 below:

Internet of Things (IoT)

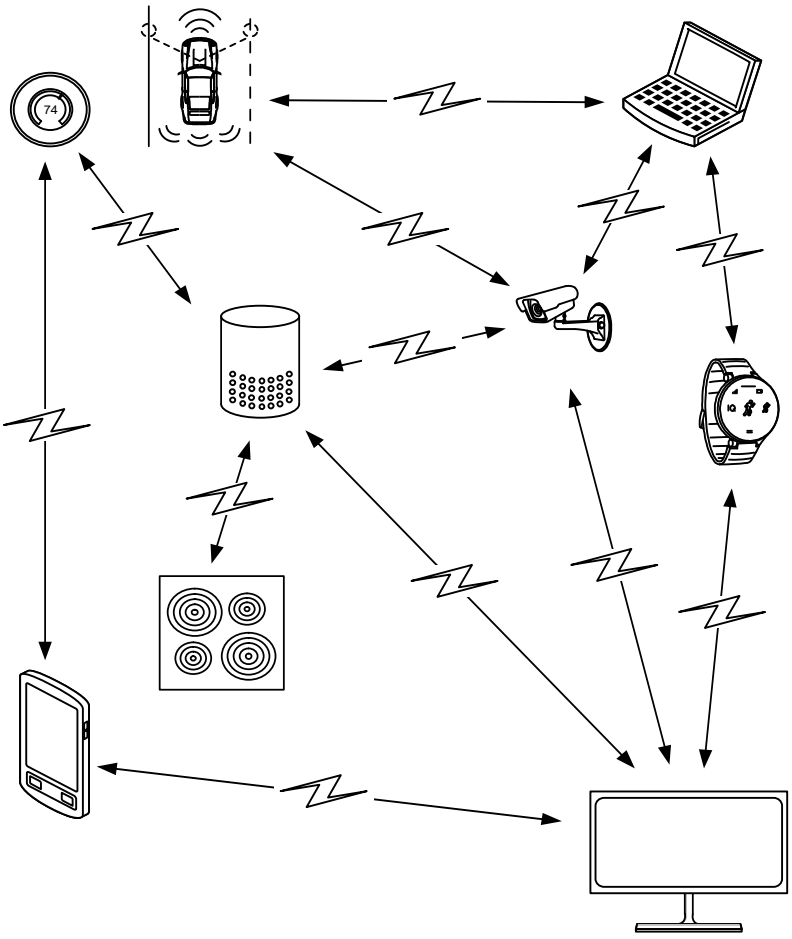

Fig. 1

In the IoT environment of Fig. 1, data may be collected by sensors of a device and shared with another device. Processing of data may be performed local to the device collecting the data or remote from the device collecting the data. Combinations of hardware (*e.g.*, sensors, microprocessors, memory), software (*e.g.,* algorithms, GUI's), and services (*e.g.*, communication networks) may be used to sense, collect, and exchange data. Large amounts of data are expected to be exchanged, as part of the IoT, across a horizon that is developing and changing frequently.

A particular aspect of the IoT includes smart kitchens, which are capable of being automated via devices such as a smart phone, tablet, or voice-recognizing personal assistant. Combining the capability of a smart kitchen to be automated, via devices such as the aforementioned devices, with data available via recipe applications or cooking websites, can be accomplished with a suite of application programmable interfaces (APIs) to offer advantages not currently realized today.

Description:

Figure 2, below, illustrates an example smart kitchen environment, including example appliances of (1) an environmental control system, (2) a voice-recognizing personal assistant, (3) a slow cooker, (4) an cooktop stove and range, (5) a microwave oven, and (6) a refrigerator.

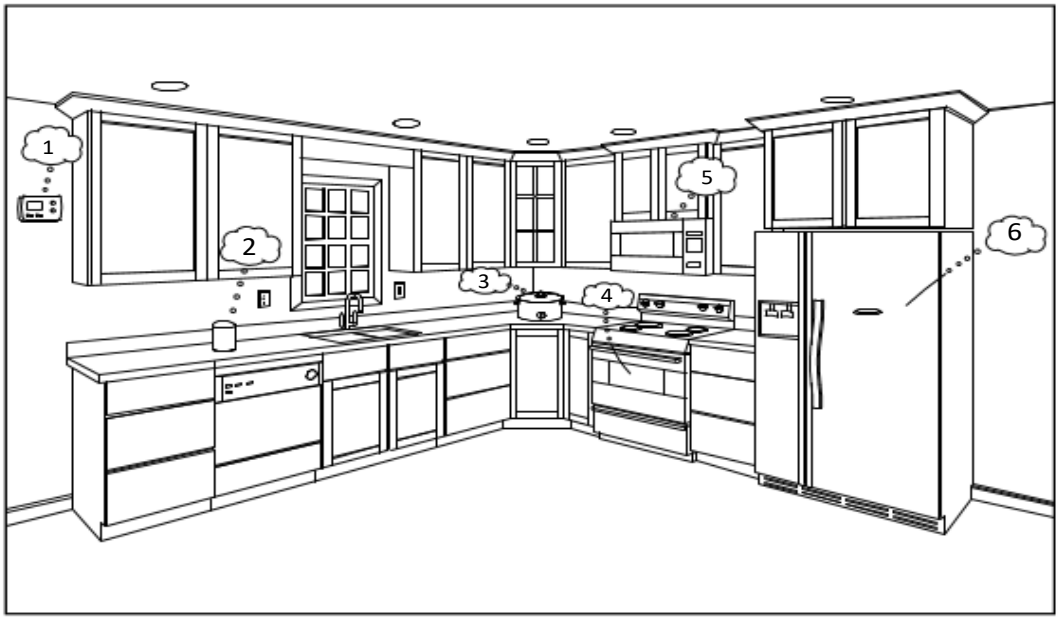

Fig. 2

Each example appliance, as illustrated in Fig. 2, possesses a combination of components, which renders it to be a "smart appliance". The combination may include, for example, a logic

component, a memory component storing instructions, and a component enabling communication between the appliance and another device in either a machine-to-machine environment or over a network (*e.g.*, a wi-fi/Bluetooth® transceiver or an ethernet interface board).

In particular, the voice-recognizing personal assistant may have connectivity to cloud-based services as well as any appliance in the example smart kitchen. With this connectivity, it is possible for the voice-recognizing personal assistant to directly or indirectly control any of the appliances using an application programming interface (API) suite that may reside across the cloud-based services, the appliances, and the voice-recognizing personal assistant itself. The control of the appliances, using such an API suite, offers the ability to optimize performance of the smart kitchen and also allow a chef within the smart kitchen to operate "hands-free".

Consider the basic scenario illustrated in Fig. 3 below:

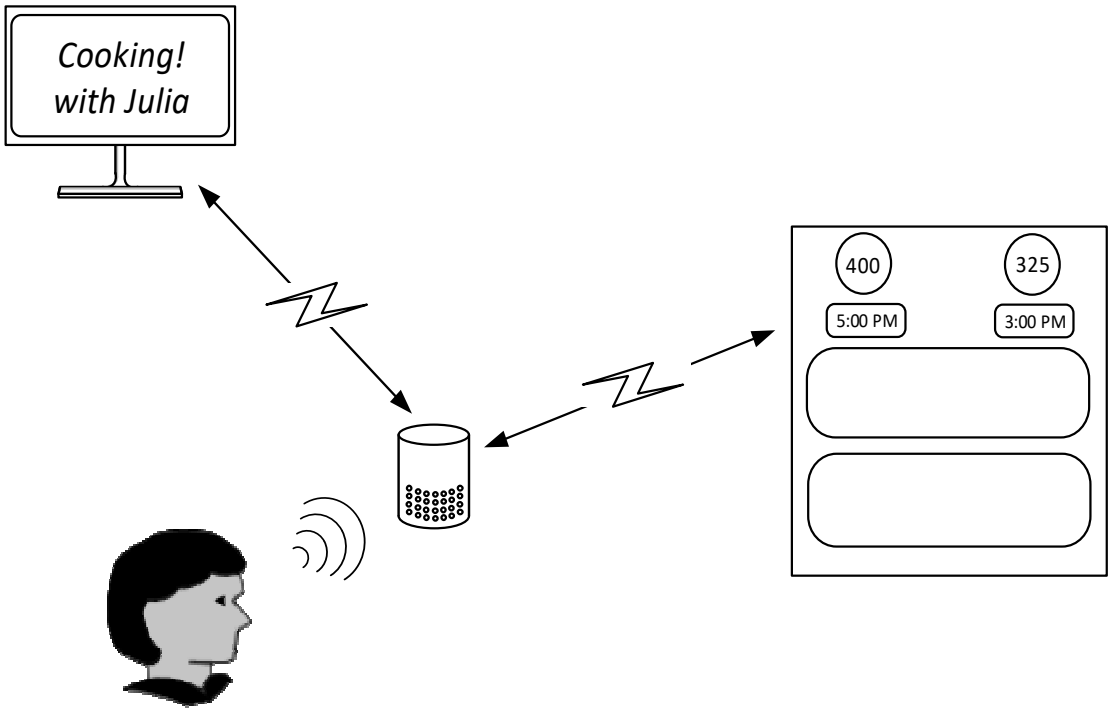

Fig. 3

As depicted in Fig. 3, a chef is vocalizing commands to a voice-recognizing personal assistant (hereinafter referred to as "Chef's Assistant"). The Chef's Assistant could be a dedicated device such as a smart speaker, or integrated into a mobile device such as a smart phone or a smart watch.

As part of a first example, the chef may be following a cooking show (*Cooking! with Julia*) being presented on a media player and be pre-occupied preparing ingredients. As his hands are dirty, he may ask the Chef's Assistant to turn on a smart double oven at two different temperatures, or he may ask the Chef's Assistant to rewind and replay a portion of the cooking show. As the smart kitchen is being automated, the chef is free to perform other duties (wash/prep food, multi-task, etc.). In effect, the chef is freed up to perform other functions without needing to directly interface with the smart double oven or the media player. Overall efficiencies of both the chef and the smart kitchen, as a result, improve.

More developed commands and responses using the API suite to automate the smart kitchen are possible. Consider an example command and automation sequence that begins with the command "Chef's Assistant, I would like to cook the roast beef and baked potato dinner that was broadcast last night on *Cooking! with Julia*. Please prepare the kitchen for a 6:00 PM serving". Upon receiving the command, the Chef's Assistant would then initiate a search, possibly using assistance of a cloud-based service, to retrieve details (from a website, recipe application, or other similar data source) associated with the roast beef and baked potato dinner broadcast on *Cooking! with Julia* the previous evening. After retrieving details associated with the broadcast, a determination would be made that one oven (of an available smart double oven) needs to be at 325ºF at 3:00 PM to begin cooking the roast while the other oven needs to be at 400ºF at 5:00 PM to bake the potatoes. Sensors contained in the smart double oven may monitor not only

temperature of the oven environment, but actual temperature profiles of the roast to ensure it is cooked to a desired state (rare, medium rare, or well-done).

Using such an API suite may also enhance cooking quality within the smart kitchen. Continuing with the example of the roast beef and baked potato dinner, temperature profiles may be optimized and controlled automatically. Based on data available from any number of websites or applications, it may be determined that the best roast would be produced not by cooking at the single temperature indicated by the immediate recipe (*e.g.*, the temperature obtained via the *Cooking! with Julia* recipe), but rather an optimized temperature profile which begins at a lower temperature of 275°F at 2:00 PM and ramps to 425°F at 5:40 PM to sear the roast during final stages of cook. Another example of enhanced cooking quality may include determining, based on data available through the smart kitchen's environmental control system, that the atmospheric pressure of the smart kitchen is equivalent to that of Denver, Colorado, and that due to the equivalence of a change in elevation, temperature profiles need to be optimized even further.

Depending on construct of the API suite, determinations and communications of commands that automate the smart kitchen may be performed by the Chef's Assistant, the cloud-based service, the smart appliance, or combinations thereof. For example, a particular configuration and combination of APIs may first rely on the Chef's Assistant communicating a request to a cloud-based recipe service via a wi-fi router and modem. After making a determination, the cloud-based recipe service would communicate commands to the Chef's Assistant back through the modem and wi-fi router. The Chef's Assistant could then communicate a command to the smart appliance via the wi-fi router or via a Bluetooth® connection. A multitude of communication protocols and combinations exists depending on the construct of the API suite and the capabilities of smart appliances within the smart kitchen.

The API suite can also evaluate the status of a smart kitchen to support a particular recipe. In the example of the roast-beef and baked potato dinner, an inventorying of contents of a smart refrigerator could be performed and conclude that there is no roast available for cooking or no potatoes available for baking. In this instance, the inventorying might be performed by an API that is directed to the refrigerator, and may rely on barcoding, visual recognition, or other techniques performed by the smart refrigerator.

Similarly, it may be possible to ask the Chef's Assistant to inventory the entire smart kitchen, and based on what ingredients are available, generate a dinner menu and prepare, accordingly, appliances in the smart kitchen that support courses of the menu for cooking. Expiration dates may be considered in generating a dinner menu, prioritizing use of ingredients with shorter remaining shelf-lives over those with longer remaining shelf-lives in order to minimize waste of food.

The API suite may gather data in order to provide a closed-loop feedback mechanism for machine-learning (ML) or artificial intelligence (AI). Recipes that have been aborted or received negative feedback, via the chef speaking the Chef's Assistant, may be recorded in a database supported by the API suite for future reference, modification, and the like.

It is also noteworthy that although the above environment is described in the context of a voice-recognizing personal assistant, other input devices may be used with similar effect. For example, a smartphone, a tablet, or a computer, may utilize a similar API suite to automate a smart kitchen, using key inputs (as opposed to voice commands).