Demo: Browsing the Web of Things with *Summon*

Thomas Zachariah, Joshua Adkins, and Prabal Dutta
Electrical Engineering and Computer Science Department
University of Michigan
Ann Arbor, MI 48109
{tzachari,adkinsjd,prabal}@umich.edu

Abstract

We are becoming increasingly surrounded by smart and connected devices, popularly known as the Internet of Things. The emerging user interface paradigm for many such things eschews physical buttons, knobs, and displays in favor of virtual interfaces that are downloaded from the web and rendered on remote platforms—like smartphones. However, such smartphone app-based interfaces often require tedious discovery and installation, as well as device discovery, pairing, and configuration before a user can interact with a nearby device. Requiring an explicit app install for each new device type scales poorly with device growth, and particularly hinders casual interactions with ambient devices. Instead of the high-friction, walled-garden approach now taking root, we propose *Summon*, a physical web browser that provides a seamless, scalable approach to browsing and interacting with nearby things. *Summon* leverages multiple network patterns and modern web technologies to provide users with rich device interfaces, even for devices under network or power constraints. We argue that this approach scales better and that it provides more intuitive and natural functionality for both users and developers. This demo presents the basic concept, allows others to experience our preliminary implementation, and raises several open research questions.

1. INTRODUCTION

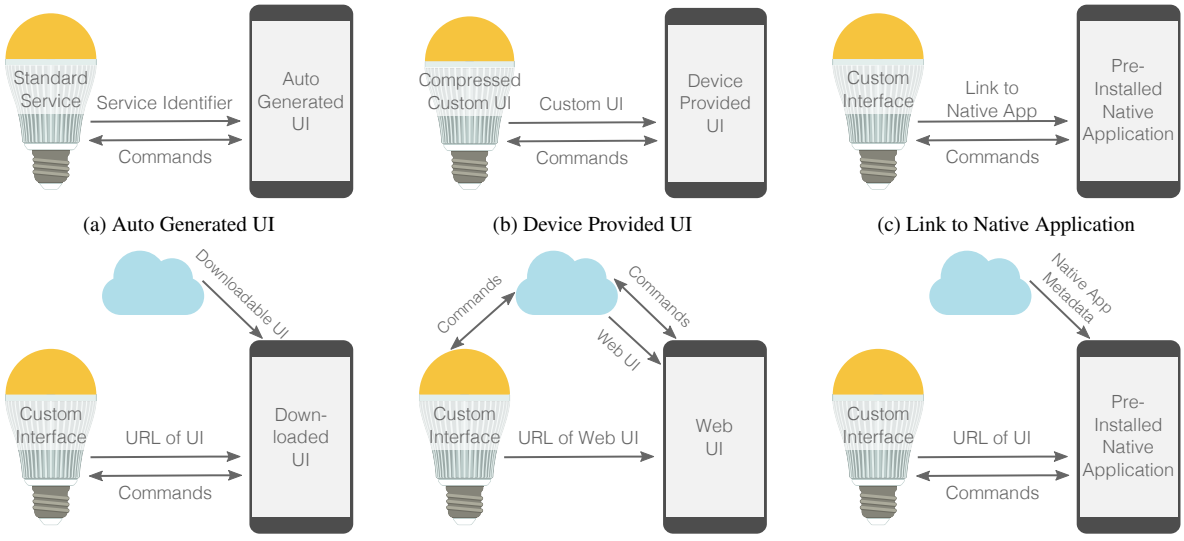
The number of smart devices is increasing dramatically. For these devices, the paradigm for user-interaction has shifted away from on-device interfaces to richer, downloadable soft user interfaces (UIs) on remote devices like smartphones. However, requiring the involved process of app installation for every new device, as is currently prescribed in industry, is unrealistic and does not scale well. The process often requires users to have some prior knowledge of and about the devices around them, and introduces the latency, digital clutter, and privacy concerns associated with installing a native application. We need a simpler system that can enable interaction with any nearby device at any time.

Fortunately, systems are being developed to solve this problem. Google has even launched a project—the Physical Web—to support browsing the web of things without requiring users to install native applications [1]. Their system defines a protocol that devices can use to advertise information about themselves using Bluetooth Low Energy (BLE) or zero-configuration network services. Mobile phones using the system then scan for these special broadcasts, display the results to the user in a Physical Web browser, and navigate the user to either the native application or a web page referenced by the broadcast when the corresponding device result is selected. However, this system requires that both the broadcasting device and the phone maintain constant Internet connection to create an interactive user interface. This is not practical for all classes of smart devices and usage scenarios. Users need to be able to browse nearby devices and load dynamic, interactive user interfaces for devices that are not otherwise Internet connected. Moreover, devices should be able to take advantage of the reduced power and complexity of interacting using only local networking protocols.

To solve this problem, we introduce *Summon*, our physical device browser. The *Summon* architecture, like that of the Physical Web, allows smart devices to point mobile phones to rich user interfaces, as is depicted in Figure 1. Unlike the Physical Web, *Summon* defines and supports a new class of interactive user interfaces that account for the possibility of the device or phone having limited network connectivity. This new class of UIs allows *Summon* to provide users with the best user interaction experience possible for any device under whatever constraints may be at play.

Figure 1: The *Summon* physical device browsing architecture. Peripheral devices like smart things (light bulb, thermostat) and beacons make their presence known by broadcasting information about themselves. The mobile phone *Summon* service scans the local vicinity, parses broadcast data, and displays to the user a selection of available user interfaces corresponding to the nearby devices. When a user selects an item from the list, the best possible interface for a given device is opened within the app, presenting a user interface that may interact *directly* with the device.

SenSys'15, November 1–4, 2015, Seoul, South Korea.
ACM 978-1-4503-3631-4/15/11.
DOI: http://dx.doi.org/10.1145/2809695.2817864 .



(a) Auto Generated UI (b) Device Provided UI (c) Link to Native Application

(d) Downloadable UI (e) Web UI (f) Native Application instead of Web UI

Figure 2: The different types of user interfaces that can be instantiated by *Summon*. While Physical Web only supports (c) and (e), all the user interfaces are important to a complete device browser, especially those interfaces that do not require the smart device to be connected to the Internet or the installation of a native application: (a), (b), and (d). It should also be noted that, as in (f), the device browser should give the user the richest interface possible for a given set of constraints. In (f), *Summon* starts to retrieve the downloadable UI, but checks for an installed native application as specified in the meta-data, and opens the native application instead of the downloadable UI.

2. LOADING AN INTERFACE

To achieve the goal of loading the best user interface for devices under any constraints, *Summon* defines and supports five interface types: 1) web based UIs with which devices interact entirely through the cloud, 2) native applications, 3) downloaded web-like UIs which interact directly with the device, 4) device provided web-like UIs which interact directly with the device, and 5) auto-generated UIs. The architectures for each of these UIs is shown in Figure 2. For *Summon* to load one of these UIs for a device, the device must be compatible with the *Summon* broadcasting protocol.

2.1 Broadcasting Parameters

Like Physical Web, *Summon* can receive broadcasts as Bluetooth Low Energy advertisements, or over local network service discovery protocols, mDNS and SSDP, over WiFi. As part of the broadcast parameters, the peripheral must indicate the target location from which to receive the UI. This location can be expressed as a unique identifier for a user interface hosted on the device, a package name link for a native phone application, or a URL. If the destination is a URL, for instance, it is treated as the source address for either a downloadable or web-based user interface.

2.2 Destination Resolution

An explicit goal of *Summon* is not only to provide a user with a user interface, but to provide a user with the best user interface possible given the broadcasted information. Therefore *Summon* first attempts to provide the user with a native application if one is already installed, it then attempts to load a web or web-like UI from the cloud, then attempts to load a web-like UI from the device, and on last resort, *Summon* will attempt to auto-generate a UI. *Summon* utilizes a combination of metadata at broadcasted URLs, self-identifying information held on the device, and its knowledge of the smartphone's network status to step through the possible interfaces and select the best one.

2.3 Other Design Details

Many other design decisions are necessary to fulfill the *Summon*'s goals. To achieve downloadable web-UIs that can interact directly with a device, *Summon* exposes native APIs in web-scripting languages. To allow for devices to provide UIs directly to *Summon* over BLE, *Summon* defines a Bluetooth service that specifies the retrieval of these UIs. Caching is implemented to further reduce latency and increase network independence of downloadable user interfaces, and phone services such as location and device information are exposed to allow for richer downloadable applications.

3. DEMONSTRATION

Summon will be available as a mobile application on the Android play store. Interested participants can download the application, or use our provided smartphones to interact with *Summon*-compatible peripheral devices that we have created or modified. We will configure the *Summon*-compatible peripherals such that every UI type supported by *Summon* will be on display.

We will also demonstrate the ease of using modern web tools to create user interfaces for direct interaction with devices. We will walk through the quick setup for each UI type, including steps for hosting UIs online or storing them on the device for limited connectivity scenarios. This will exhibit how *Summon* allows users to casually interact nearby devices, while providing developers a simple method for creating rich UIs for their devices.

4. ACKNOWLEDGMENTS

This material is based upon work supported by NSF under grants CNS-1239031, CNS-1350967 and CPS-1505684, and by the NSF/Intel Partnership on CPS Security and Privacy.

5. REFERENCES

[1] Google Inc. The physical web. http://www.github.com/google/physical-web.