iConfig: What I See is What I Configure

Michael Haus
Technical University of Munich

Aaron Yi Ding
Technical University of Munich

Pan Hui
Hong Kong University of Science and Technology

Jörg Ott
Technical University of Munich

ABSTRACT

Managing IoT devices in urban areas is becoming crucial because the majority of people living in cities and the number of deployed IoT devices are steadily increasing. In this demo, we present iConfig, an edge-driven platform dedicated to manage densely deployed IoT devices in smart cities. Our goal is to address three challenging issues in current IoT management: registration, configuration, and maintenance. The core of iConfig is its programmable edge module, which can be deployed across smartphones, wearables, and smart boards to configure and interact with physically proximate IoT devices. Our system evaluation shows that iConfig can effectively address the aforementioned IoT management challenges by harnessing mobile and edge cooperation. The live demonstration further showcases how iConfig can utilize speech recognition on smart wearables (e.g., smart glass) to achieve more natural and frustration-free IoT management.

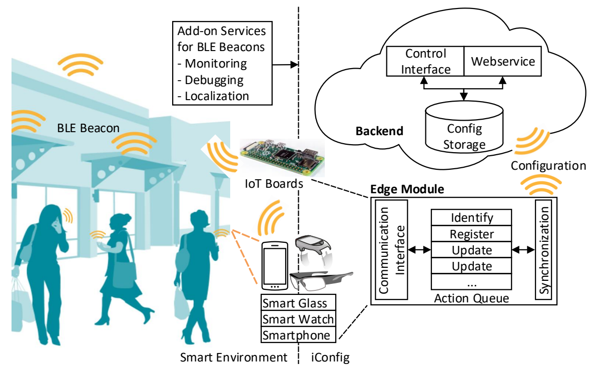

Figure 1: iConfig in the context of smart environments

1 INTRODUCTION

The majority of people lives in cities and the number of deployed Internet of Things (IoT) devices are steadily increasing. Managing urban areas and its applications is hence becoming important. These urban IoTs support the smart city concept [1] which integrates traditional and modern information and communications technology (ICT) for a unified and simple access to services for the city administration (e.g., managing urban areas) and the residents [2, 3]. The aim is an enhanced use of public resources, improving quality of services for citizens while reducing operational costs of public administration [2]. In spite of a growing demand for IoT management to keep an up-to-date overview of densely distributed devices during different phases of their life cycle, including installation, registration, user customization, and device control. There is a lack of instrument to seamlessly manage large IoT deployments in which ad hoc management is becoming untenable, especially for IoT devices without Internet connectivity. Therefore, we propose iConfig, an edge-driven platform to bridge such gap. Fig. 1 shows iConfig in the context of smart environments taking advantage of wearable and edge computing in the form of programmable devices to run edge modules on smart glasses or IoT boards. This allows iConfig to consolidate various connected devices to its management backend, which enforces a unified configuration procedure. As an example, we successfully tested iConfig on Android smartphone and smart glass [4]. In particular, iConfig edge module is dedicated for users interested in managing and interacting with IoT environments. The design principles of iConfig include: 1) automatic configuration of IoT devices to avoid misconfigurations which become one of the dominant causes of system failures [5], 2) easy to use frontend, 3) orchestrate devices via global view, and 4) serve as platform for developer to enable add on services.

In this demo, we use iConfig to configure Bluetooth Low Energy (BLE) beacons, which represent one of the most challenging classes of IoT devices due to missing internet connectivity. These battery powered BLE beacons are small-size wireless devices that transmit a short-range BLE signal to mobile computing devices (e.g., smartphones) [6]. Via BLE, users' devices are notified of the beacon proximity by receiving signals which contain contextual information, typically about indoor surroundings and its contents. Thus, the receiving end is able to perform location aware actions, such accessing specific URLs for marketing and assistance purposes [7]. The ability of iConfig to programmatically adjust device parameters facilitates different use cases: 1) set up an automated testbed for research projects, 2) debug and monitor IoT devices, and 3) energy aware management of IoT devices deployed in smart buildings.

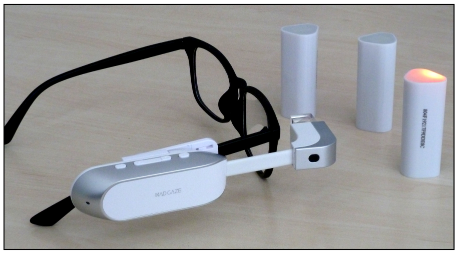

Figure 2: Smart glass with BLE beacons, currently configured beacon blinks

2 ICONFIG FOR IOT MANAGEMENT

The iConfig system architecture, as illustrated in Fig. 1, consists of two major modules: mobile edge module and backend module. The framework is able to identify, register, and update IoT devices (as in our case BLE beacons). For instance, supported by our speech recognition functionality, a user wearing an iConfig-enabled smart glass can discover, register, and configure BLE beacons while walking around. Fig. 2 shows the demo hardware including smart glass and BLE beacons. In the discovery phase, the edge module shows only unregistered beacons. After discovering unregistered beacons, the user is able to register one beacon at a time. The beacon shows a red light as feedback for device identification. When the beacon identification was successful, the user can register the beacon to the iConfig backend with additional information for device localization, such as nearest room number, picture of device place. Afterwards, the edge module will automatically configure the BLE beacon with a default configuration including password, URL, transmission power, and packet advertisement rate. Finally, edge module will synchronize all configuration data to the backend. The registration has to be done only once per beacon. For advanced management, iConfig backend provides a control interface for the administrator, including a global view about installed BLE beacons. The device update is independent of user actions and automatically triggered when two conditions are satisfied: 1) adapted BLE configuration from the administrator via iConfig backend available, and 2) currently discovered beacon by the user via iConfig edge module.

3 EVALUATION

We analyzed the system performance of iConfig regarding memory usage and scalability over multiple beacons. Additionally, we conducted a user study to highlight the drawback of manual configuration, which is untenable in dense deployment.

Regarding the memory usage of iConfig edge module, we measured a deployment on Android smartphone (OS 7.1.1) in offline and online mode during configuration of ten BLE beacons. In online mode, the edge module used 6.50 ± 0.98 MB similar to offline mode with a memory usage of 6.47 ± 0.96 MB. These results show that the memory footprint of the iConfig edge module is small enough to run on programmable IoT devices.

Table 1: Beacon configuration time (s) for scalability test from one to ten beacons

1	2	3	4	5	6	7	8	9	10
2.7	5.3	7.5	10	12.4	15.1	16.5	16.4	17.8	22

For scalability we evaluated ten cases (from one to ten beacons) each over 20 rounds in a dense testbed deployment. Our evaluation yielded 18 unauthenticated connect errors (meaning BLE configuration is not possible) out of 1100 connect attempts, i.e., rate of 1.64 %. Table 1 shows a linear increase of configuration time over all beacons, on average an increase of 2.2 s per beacon. In addition, we evaluated the success rate which describes whether configuration parameters were correctly set. The lowest rate was 75 % during configuration of four beacons. In most cases, iConfig achieved 100 % success rate.

Our user study further revealed the gap between a manual and automatic configuration system. Ten participants manually configured three beacons with a predefined configuration using the vendor application. In the next step, the same beacons were configured via iConfig, in which the default configuration was automatically written to each BLE beacon. The manual configuration took in average six times longer than iConfig automatic configuration, which reflects a time saving of 83 %. Only 1/3 of all manual configurations were entirely correct. On the other hand, iConfig achieved for all BLE configurations the success rate of 100 %.

4 HANDS-FREE IOT DEVICE CONFIGURATION

We take advantage of speech recognition to enable hands free device configuration and implemented two prototypes for different device types: smartphone and smart glass. On the smartphone the speech recognition can be optionally activated, on the smart glass it is automatically activated to allow a convenient usage of the iConfig edge module. The speech recognition is based on two implementations: Google speech recognition and Pocketsphinx [8], and works completely offline on the smart glass. iConfig can be activated via the key phrase *geronimo* for device discovery and registration. Afterwards, the user is able to control iConfig via the following speech commands: *select* nearest beacon automatically detected via strongest received signal strength, *target* specific beacon via beacon ID, *identify* as visual feedback of the beacon, and *register* beacon at iConfig backend.

To capture an image of beacon placement for easier localization, we implemented a customized camera dialog via speech control. The keyword *shot* takes an image and the zoom range of the camera is divided into five levels. Users can control the zoom range via the speech commands: *more*, *less*, *min*, and *max*.

To enhance the robustness of our speech recognition (e.g., number three sometimes recognized as "free"), we used several metrics for string similarity of the speech input: Euclidean, Levenshtein, and Jaro-Winkler distance. In our setting the Jaro-Winkler distance, which is designed for short strings and can detect typos, provided the best results to calculate the string similarity of iConfig speech input.

5 DISCUSSION AND OUTLOOK

A key observation from our user study and experiments is that the interaction among users, their smart gadgets and surrounding IoT devices via the conventional screen-keyboard setup is far from optimal. Especially for smart cities with a multitude of services empowered by IoT devices, the system interaction should be more natural and fluent. This is the main reason for our prototype dedicated for smart wearables (e.g., smart glass). The combination of speech recognition and hands-free devices can enable a more integrated interaction during user movement and limit the distraction of user attention. iConfig is hence our endeavor to enhance user experience and to streamline the management of large scale IoT deployments, especially for low-budget devices such as BLE beacons without backend connectivity.

To tackle more challenging scenarios of IoT management where IoT devices are deployed in harsh environments (e.g., high ceilings, steel-making factories), we plan to explore using iConfig-enabled drones for achieving autonomous configuration, which is highly desired in dense IoT deployments.

6 DEMONSTRATION PROCEDURE

In this demonstration, we present a typical use case of our platform to configure BLE beacons via smart glass over speech control. For comparison, we provide a smartphone to configure the beacons. The testbed layout is depicted in Fig. 3. The laptop runs the iConfig backend server together with a MongoDB database which is accessible via a local Wi-Fi hotspot. The iConfig backend server receives the beacon configurations and enables add-on services. Besides that, the laptop projects the screen content of the smart glass to a larger monitor. This monitor shows the GUI of the iConfig edge module running on the smart glass and the control panel of the iConfig backend. For device configuration, our demo includes a smart glass and a smartphone running the iConfig edge module to identify, configure, and update BLE beacons. We provide ten BLE beacons for configuration.

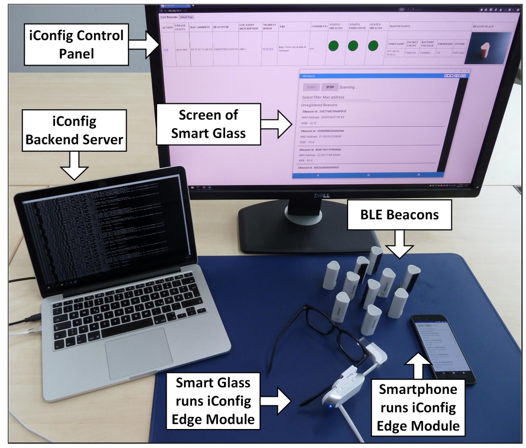

Figure 3: Demo setting

Our demo includes the following devices: 1) smart glass running iConfig edge module with speech control for hands-free device configuration, 2) smartphone running iConfig edge module for conventional manual input or optional speech control, 3) ten BLE beacons for device configuration, 4) a laptop to run the iConfig backend server and to project the screen content of the smart glass to an external monitor, and 5) a headphone for improved speech control in noisy environments.

We would need space for a poster board and a table, big enough for one laptop and a 24-inch monitor. Our prototype system requires approximately 30 minutes to setup.

In addition, we need a table and power for the demo. We also prefer to have a large monitor with HDMI or VGA cable to show the screen content of the smart glass running the iConfig edge module. We will cast the control panel of the iConfig backend module on the monitor as well.

REFERENCES

[1] Schaffers et al. Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation. In *Future Internet. Lecture Notes in Computer Science, Vol. 6656. Springer.* 2011. 431–446.
[2] Zanella et al. Internet of Things for Smart Cities. *IEEE Internet of Things Journal* 1, 1 (2014), 22–32.
[3] Zdraveski et al. ISO-Standardized Smart City Platform Architecture and Dashboard. *IEEE Pervasive Computing* 16, 2 (2017), 35—43.
[4] http://madgaze.com/x5/ (visited on 05/20/2017)
[5] Xu et al. Systems Approaches to Tackling Configuration Errors. *ACM Computing Surveys* 47, 4 (2015), 1–41.
[6] Wang et al. Managing Large Scale, Ultra-Dense Beacon Deployments in Smart Campuses. In *Proceedings of IEEE INFOCOM WKSHPS* 2015.
[7] http://www.airport-world.com/news/general-news/5976-lax-uses-bluetooth-beacon-technology-to-improve-wheelchair-operations.html (visited on 05/20/2017)
[8] D. Huggins-Daines et al. Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices. In *Proceedings of the IEEE ICASSP*. 2006. 185–188.