IoT Linking BLE

Technology Reports

“Linking” Device Functions to Expand the IoT

With the expansion of the IoT device market of recent years, many companies have been developing products in this field, although not many consumers have taken up using these products in their daily lives. Therefore, by establishing “Project Linking” between various companies nationwide, NTT DOCOMO has set up a platform called “Linking” to make it easier to develop IoT devices and services linking with these IoT devices. This article describes a summary of Linking, examples of its practical application and how they are deployed.

Communication Device Development Department *Katsuki Kiminami*
Yuji Tamiya
Kouen Koh
Sadaatsu Kato †

1. Introduction

With the rapid spread of smartphones of recent years, many wearable devices that can link with smartphones such as smart watches, smart glasses and wristband-style healthcare devices have appeared. In addition, with the advocacy of the IoT concept [1], in which various devices used in daily life is connected to the Internet such as home appliances, automobiles and smart meters, and various devices with sensors, many companies are proactively developing products for the wearable and IoT fields as growth and expansion of markets for these devices (hereinafter referred to as “devices”) are predicted [2].

However, many consumers are not using devices as often as they use smartphones. This is likely due to the following causes:

1) Devices are too Expensive

Many of the devices currently on the market cost tens to hundreds of dollars, so consumers are reluctant to buy them. Hence, the price of the devices is an impediment to their popularization and the growth of the IoT.

2) There are Few Killer Applications Available

Various IoT services are already on offer by many companies. The ‘smart home’ in which various consumer appliances are controlled with a smartphone is one example of such an IoT service. However, consumers in general do not feel the need for such devices and services in their daily lives, and there are not many killer IoT services that a majority of people would find attractive enough to want to use.

3) Developers Face Hurdles to IoT Service and Device Development

Most of the devices currently available for use with smartphone applications (hereinafter referred to as “service apps”) connect using the unique specifications of individual manufacturers, which means these service apps must be developed to match each device, and service app developers have to also understand the specifications of each device.

To counter these problems, Application Programming Interfaces (API)[*1] for

† Currently Cellular Phone Division
DOCOMO Technology, Inc.

[*1] **API:** An interface that makes the functions provided by the OS, middleware and other such software available to upper-level software.

connecting devices with smartphones using Bluetooth®*2 etc. have been published, such as Android Wear™*3 [3] and Google Fit™*4 [4] provided by Google, and Apple Watch™*5 [5] and HomeKit®*6 [6] provided by Apple. However, these specifications require devices to have operating systems installed or high-spec RAM*7, which means compatible devices are limited to those with high functionality. Thus, developing devices compatible with those specifications requires a greater amount of development, making it more difficult to provide devices.

Thus, to expand the IoT market, NTT DOCOMO developed the Linking platform to make it easier to develop IoT devices and service apps, by prescribing simplified, general-purpose specifications for connecting devices with service apps. This article describes an overview of Linking, examples of its practical application and how they are deployed.

2. Overview

Linking is a platform that makes it easy to connect devices and smartphone service apps. Developing with the interface prescribed by Linking enables device and service app development in which device developers need not be concerned with service app specifications or operations, and similarly service app developers need not be concerned with device specifications. Furthermore, so long as there is compatibility with the Linking interface, connection is possible regardless of the type of device or service app. Linking also enables multiple devices to connect to multiple service apps.

2.1 Configuration of Linking on Android

Figure 1 shows the configuration of Linking on Android™*8. Note that the configuration of Linking on Android differs from iOS*9. The configuration on iOS is described later.

As shown in Fig. 1, connections between devices and service apps are achieved by the Linking app. The Linking app supports the interface for devices (hereinafter referred to as "device IF") and the interface for service apps (hereinafter referred to as "service app API"). To connect devices with smartphones, we adopted the Bluetooth Low Energy (BLE)*10 standard for its particularly good energy-saving characteristics, and because Bluetooth has become a standard inclusion in smartphones and a range of other devices such as PCs.

The Linking app also centrally manages settings such as service app or device connectivity, device connection status, and provides detailed user settings for how to combine devices and service apps. By using the Linking app, users do not need to operate individual service apps or devices to make service app-device connections, which raises the level of user convenience, especially when using multiple devices and service apps.

Furthermore, since the Linking app runs on the Android application layer, it can be used by simply downloading and installing it in a smartphone, and can also be used on smartphones provided by NTT DOCOMO and other companies.

2.2 Device IF

1) Linking Supported Devices

As of April 2016, devices compatible with Linking are those equipped with LEDs, vibrators and buttons, and acceleration, gyro, orientation, humidity and

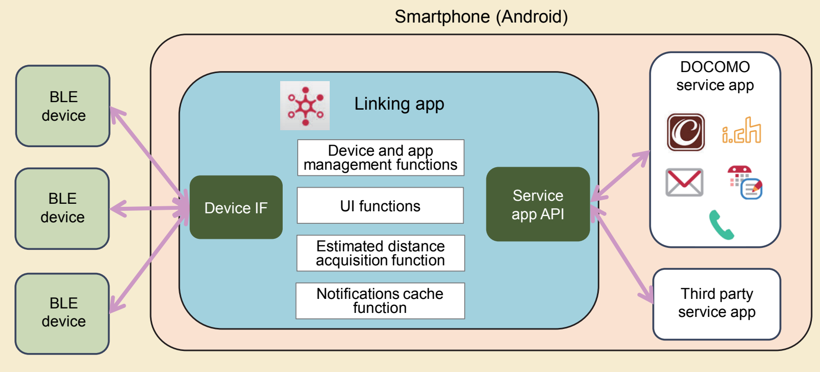


Figure 1 Configuration of Linking on Android

*2 **Bluetooth®:** A short-range wireless communication standard for interconnecting mobile terminals such as cell phones, notebook computers and PDAs. A registered trademark of Bluetooth SIG Inc. in the United States.

*3 **Android Wear™:** An Android OS-based operating system for smart watches. A trademark of Google Inc.

*4 **Google Fit™:** The name of Google's health management and health care applications, and their platform. A trademark or registered trademark of Google Inc.

*5 **Apple Watch™:** The smart watch provided by Apple Inc. A trademark or registered trademark of Apple Inc.

*6 **HomeKit®:** The platform provided by Apple Inc. for connecting iOS devices with home appliances etc. A registered trademark of Apple Inc.

*7 **RAM:** High-speed read/write access memory.

air pressure sensors. We selected these functions so that simple devices with no operating systems or small RAMs of several hundred kilobytes like those discussed above can be connected. This approach enables use of inexpensive devices in daily life, and hence should provide impetus for the expansion of the IoT.

As the format of the BLE advertisement packet*11 is also defined in Linking, devices that support the format can be used as beacons*12.

2) Device IF Overview

Table 1 describes the device IF specified with Linking for connecting with service apps. The device IF is specified as a BLE communications profile*13, and sends and receives service information on the Generic Attribute profile (GATT)*14.

Although standard profiles such as Alert Notification Profile (ANP)*15 and Health Device Profile (HDP)*16 are prescribed in Bluetooth SIG*17 for different purposes, there are no generic profiles that enable communications between a range of devices with different characteristics. Thus, as shown in Table 1, the minimum BLE profiles are specified in Linking for various communications with a range of devices. Also, because the maximum service data size that can be transmitted in one communication is 20 bytes in BLE, a number of communications are required to transmit large amounts of data. Hence, devices notify the Linking app of their functional capabilities. Then, by specifying that only information that the device can handle be used in exchanges with the Linking app, the number of communications can be minimized, and unnecessary communications between the device and the Linking app can be avoided. These approaches mean that devices with a variety of characteristics can connect generically and with low power.

Also, devices compatible with the Linking can be used regardless of whether the operating system is Android or iOS.

2.3 Service App API

Table 2 describes the service app API required to connect service apps and devices. The service app API on Android are composed of standard and basic functions such as Android intent*18 and Content Provider*19, which make it easy for service app developers to use the API if they have basic app development skills. For example, in the Notification API as shown in Table 2, various notifications can be made to devices without concern about device specifications, by only sending intent for those parameters set according to API specifications from the service app to the Linking app.

2.4 Linking App

The main Linking app functions are described below.

1) Device and App Management Functions/UI Functions

The Linking app centrally manages settings such as app/device connectivity and device connection status, and provides detailed user settings for combining devices and apps. **Figure 2** shows an image of the Linking App Management screen.

(1) As shown in Fig. 2 (a), With "Device Management screen," the Linking app enables management and connectivity with a

Table 1 Device IF

Service name	Details
PeripheralDevicePropertyInformationService	Obtains device information (device name, device ID, device capabilities etc.)
PeripheralDeviceNotificationService	Provides device with information from various service apps (notification ID, service app name, notification patterns etc.)
PeripheralDeviceSensorInformationService	Obtains sensor information from device (sensor type, status, data values etc.)
PeripheralDeviceSettingOperationService	Exchanges operational information between devices and service apps (information about device buttons being pushed etc.)

*8 **Android™:** A software platform for smartphones and tablets consisting of an operating system, middleware and major applications. A trademark or registered trademark of Google Inc., United States.

*9 **iOS:** A trademark or registered trademark of Apple Inc. in the U.S.A. and other countries.

*10 **BLE:** Also called Bluetooth version 4.0, a Bluetooth standard that enables communications with low power consumption.

*11 **Advertisement packets:** Packets used in BLE by devices to transmit their device information.

*12 **Beacon:** A device that periodically transmits information.

*13 **Profile:** Inter-device protocols formulated on a service-by-service basis for use in communications by Bluetooth and BLE.

*14 **GATT:** A profile used for transmitting the data in BLE.

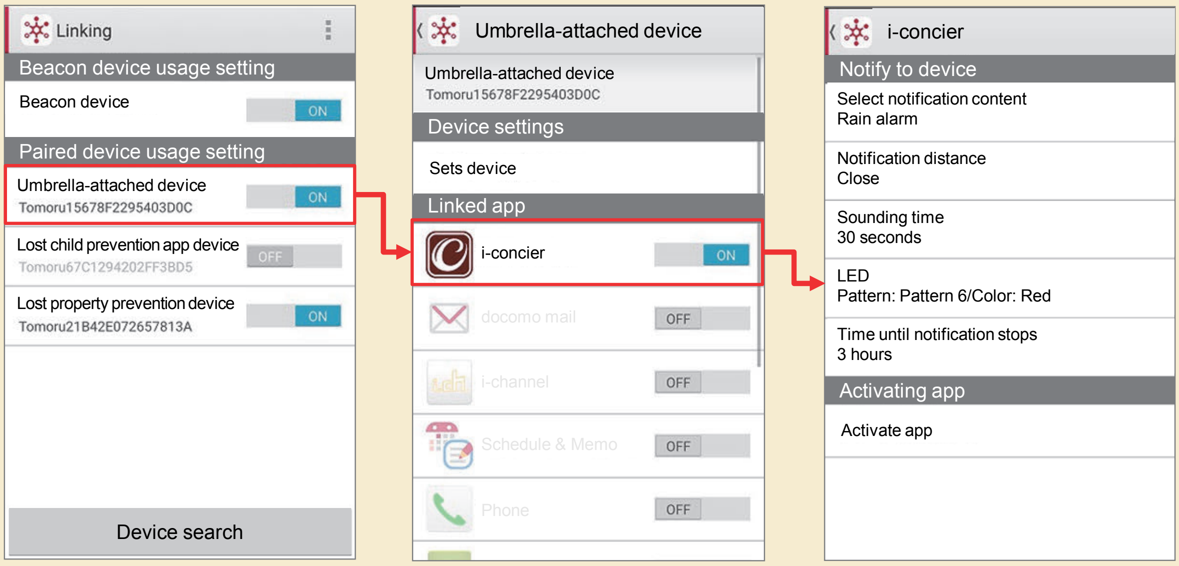
list of paired devices. Users can also activate or deactivate beacon devices.

(2) Users can select the service apps to be used with devices on the "Service App Management screen" at Fig. 2 (b), by selecting a device on the Device Management screen. For example, "a device attached to an umbrella" is connected to the "i-concier app" in Fig. 2 (b).

(3) Selecting a service app from the Service App Management screen transitions to "Preferences screen" at Fig. 2 (c), where users can select information to be notified to devices or make detailed service app settings such as LED patterns and illumination time. In the Preferences screen, the available settings items depend on device or service app specifications, for example, "Notification de-

Table 2 Service app API

API name	Details
Notification	Sends notifications (title, content, icon etc.) to devices
General notification	Sends more detailed information other than "Notification" to devices
Get device information	Acquires device information (device-specific information, connection status, device capabilities etc.)
Receive device information	Receives a range of device information (button info etc.)
Distance notification	Notifies distances between devices and smartphone
Sensor information	Starts/Stops acquisition of sensor information
Beacon information	Starts/Stops search for beacon devices
BLE connectivity information	Notifies BLE connection/disconnection between smartphones and devices



Linking

Beacon device usage setting

Beacon device ON

Paired device usage setting

Umbrella-attached device ON
Tomoru15678F2295403D0C

Lost child prevention app device OFF
Tomoru67C1294202FF3BD5

Lost property prevention device ON
Tomoru21B42E072657813A

Device search

(a) Device management screen

Umbrella-attached device

Umbrella-attached device
Tomoru15678F2295403D0C

Device settings

Sets device

Linked app

i-concier ON

docomo mail OFF

i-channel OFF

Schedule & Memo OFF

Phone OFF

(b) Service app management screen

i-concier

Notify to device

Select notification content
Rain alarm

Notification distance
Close

Sounding time
30 seconds

LED
Pattern: Pattern 6/Color: Red

Time until notification stops
3 hours

Activating app

Activate app

(c) Preferences screen

Figure 2 Linking app management screen

*15 **ANP:** A profile standardized by Bluetooth SIG (see *17). for notification of an incoming phone call or SMS.

*16 **HDP:** A profile standardized by Bluetooth SIG (see *17). for connecting medical and healthcare management devices.

*17 **Bluetooth SIG:** Bluetooth Special Interest Group. A non-profit industry group that represents the Bluetooth standard and oversees the development and licensing of Bluetooth technology.

*18 **Intent:** A function provided by Android OS to exchange parameters. Used between components within an application, or between applications.

*19 **Content Provider:** A function provided by Android OS which handles storage and searching of data related to general image, audio or video files, personal or other information. Content Provider enables applications to easily access various kinds of data.

tails" and "LED" are only displayed if the device or service app is equipped those functions. Therefore, users can easily understand settings because unavailable items are not displayed.

2) Distance Acquisition Function

(1) Function overview

The Linking app periodically estimates physical distances between each device and smartphone, and sends the estimated data to connected service apps. This function enables service app operations that are dependent on the distance between the smartphone and the device, such as acquisition of device sensor information only when devices are close, or alerts when the device and smartphone become to too far apart.

(2) Calculation method

In the Linking app, the estimated distance d [m] between the device and the smartphone is calculated from the Bluetooth radio intensity from the device (Received Signal Strength Indicator (RSSI)[*20]), using the equation below.

$$d = 10^{\frac{A-RSSI}{10\times n}} \qquad (1)$$

Where A is the RSSI value one meter from the device, n is the attenuation ratio, and $RSSI$ is the measured value of the Bluetooth radio intensity [dBm]. However, the measured RSSI value will change due to conditions in the environments in which the device and smartphone are being used, or the way the user is holding the device or smartphone etc. Therefore, by calibrating the variable A in the equation (1) for the actual usage environment, Linking mitigates the effects of variations in RSSI values on the calculated distance.

(3) Notifications cache function

If the Bluetooth is disconnected between the device and smartphone, or the device is too far away from the smartphone, the Linking app can temporarily save (cache) notifications from the service app, and automatically send them to the device when the Bluetooth connection is reestablished. These specifications include settings for individual service apps about whether to cache notifications and how long they should be cached. By using these functions, service apps can notify devices of various information when the user becomes close to the device, which means users can in turn get available information about the places they are in, in step with their own movements, without having to operate the smartphone.

2.5 Linking Configuration on iOS

Figure 3 describes the configuration of Linking on iOS.

Because iOS does not have the same kind of functions as the Android intent and Contents Provider, and because iOS only allows limited resident applications to operate, connections between devices and service apps cannot be achieved in the same way as Android.

Nevertheless, since it is important to be able to use devices regardless of the type of smartphone, we provide a library[*21] of Android Linking app functions for iOS (hereinafter referred to as "Linking library"), which makes it easy for iOS service app developers to develop apps to connect to devices by incorporating the Linking library into each service app (as shown in Fig. 3). Device developers can also use the device IF mentioned earlier, which is compati-

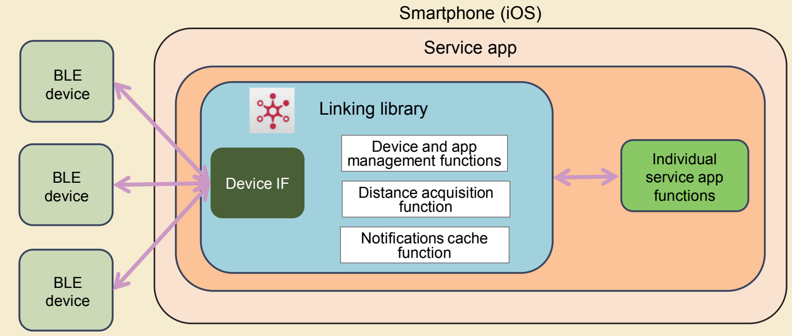

Smartphone (iOS)
Service app
Linking library
BLE device
BLE device
BLE device
Device IF
Device and app management functions
Distance acquisition function
Notifications cache function
Individual service app functions

Figure 3 Linking configuration on iOS

[*20] **RSSI:** The intensity of the signal received at the receiving device.

[*21] **Library:** A collection of high-versatility programs in a reusable form.

ble with both Android and iOS.

3. Examples of Application

Linking enables devices and apps to be combined according to user preferences. The following describes examples of device usage with Linking in two different scenes (a carry rain gear alert and lost property prevention), and describes the processing sequences in the devices, Linking app and service apps.

3.1 Alert to Carry Rain Gear

It is possible to connect a device to “Rain alarm,” one type of information available with NTT DOCOMO’s i-concier app. Rain alarm warns the user’s smartphone when rain clouds are approaching user’s current location.

For example, a device could be attached to the umbrella at entrance of the user’s house. Then, when the smartphone receives a rain alarm, and the user is about to go out, and the smartphone comes close enough to the device, the smartphone automatically sends the rain alarm to the device, which then alerts the user. In this way, the user need not check the smartphone to know that it is going to rain when they are about to leave the house, and can be intuitively alerted to the need to take an umbrella (**Figure 4**).

3.2 Lost Property Prevention

This describes a service app that alerts users to prevent lost property. To prevent users from leaving their wallets or bags somewhere, user can attach devices to things they don’t want to lose. When the device and smartphone become separated by a certain distance, the service app can sound an alarm on the user’s smartphone as well as the device to alert the user (**Figure 5**).

4. Linking Deployment

To increase the use of Linking-compatible devices and service apps, and likewise expand the IoT, NTT DOCOMO has established “Project Linking” in partnership with several other Japanese companies, and is taking initiatives to spread and promote Linking [7] [8]. Also, device IF and service app API specifications for developers are published on the company web site [9], which device and service app developers can use free of charge. Also for service app developers, NTT DOCOMO provides a Soft-

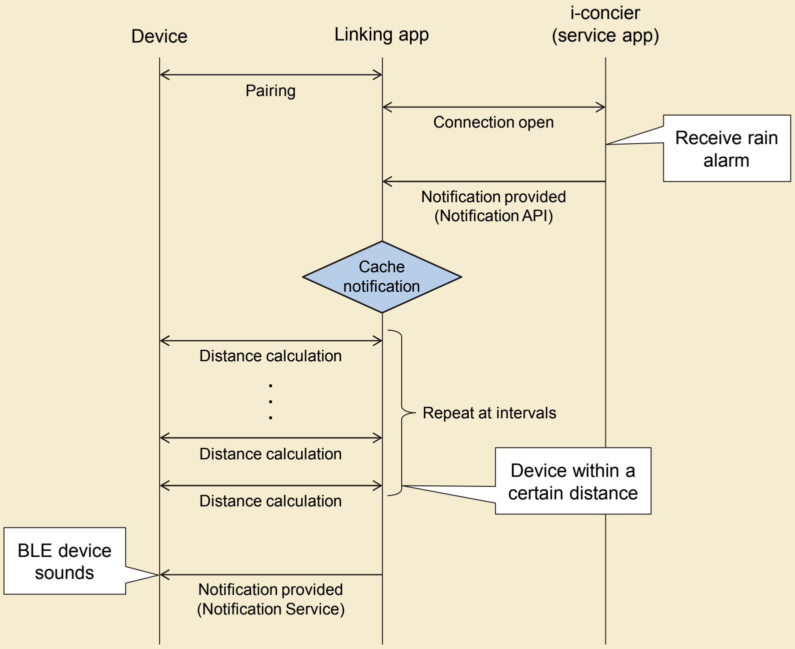


Figure 4 Sequence in case of alert to carry rain gear

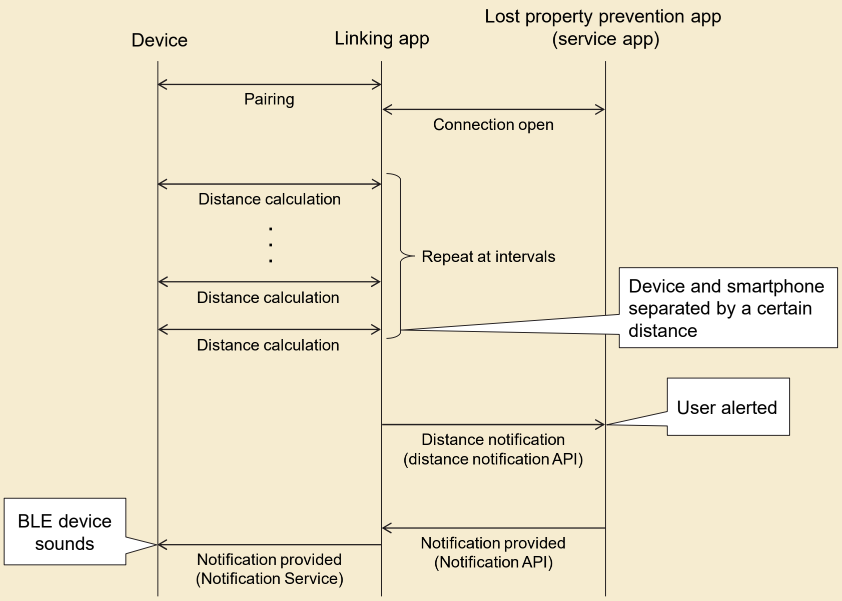

Device
Linking app
Lost property prevention app (service app)
Pairing
Connection open
Distance calculation
Repeat at intervals
Distance calculation
Distance calculation
Device and smartphone separated by a certain distance
User alerted
Distance notification (distance notification API)
BLE device sounds
Notification provided (Notification API)
Notification provided (Notification Service)

Figure 5 Lost property prevention sequence

ware Development Kit (SDK)*22 to simplify Linking-compatible service app development.

5. Conclusion

In this article, we have described an overview of Linking, which we developed with the aim of expanding the IoT market, as well as some practical examples of its deployment methods and usage.

In the future, we intend to continue to upgrade the development environment with such measures as expanding sample code for developers, and adding to and upgrading the device IF and service app API to provide and maintain a development environment for businesses and individuals to utilize. Also, Linking will support device WebAPI [10] to enable both native service apps and web apps to access devices via the Linking app. We aim to expand and promote Linking compatible devices and service apps, and hence improve the convenience of user's daily lives with smartphones linked to various devices.

References

[1] K. Ashton: "That 'Internet of Things' Thing," RFID Journal, Jul. 2009.

[2] IDC Japan Ltd.: "Domestic IoT (Internet of Things) Market Predictions," Feb. 2015 (In Japanese).
http://www.idcjapan.co.jp/Press/Current/20150903Apr.html

[3] Android Developers: "Android wear."
http://developer.android.com/intl/ja/wear/index.html

[4] Google Developers: "Google Fit."
https://developers.google.com/fit/?hl=ja

[5] Apple: "Apple watch."
http://www.apple.com/jp/watch/

[6] Apple: "iOS9-HomeKit."
http://www.apple.com/jp/ios/homekit/

[7] NTT DOCOMO Press Release: "New "Linking" platform developed for connecting smartphones with external devices," Nov. 2015 (In Japanese).
https://www.nttdocomo.co.jp/info/news_release/2015/11/25_00.html

[8] Linking portal site.

[9] Linking developer site.

[10] T. Yamazoe, et al.: "Device Connect WebAPI - Web Interface for Variety of Smartphone-linked Devices -," NTT DOCOMO Technical Journal, Vol.17, No.1, pp.4-9, Jul. 2015.

*22 **SDK:** A tool or set of tools used for software development.