Eddystone beacons (and the Google Nearby API) on Android

DevFest Hamburg 2015

Codelab

Daniel Hartwich

Agenda

1. Introduction (5 minutes)
2. Eddystone (5 minutes)
3. (Nearby API + Proximity API (5 minutes))
4. Codelab (~60 minutes)
5. Discussion round (Ideas, Problems, UseCases etc.) (10-20 minutes)

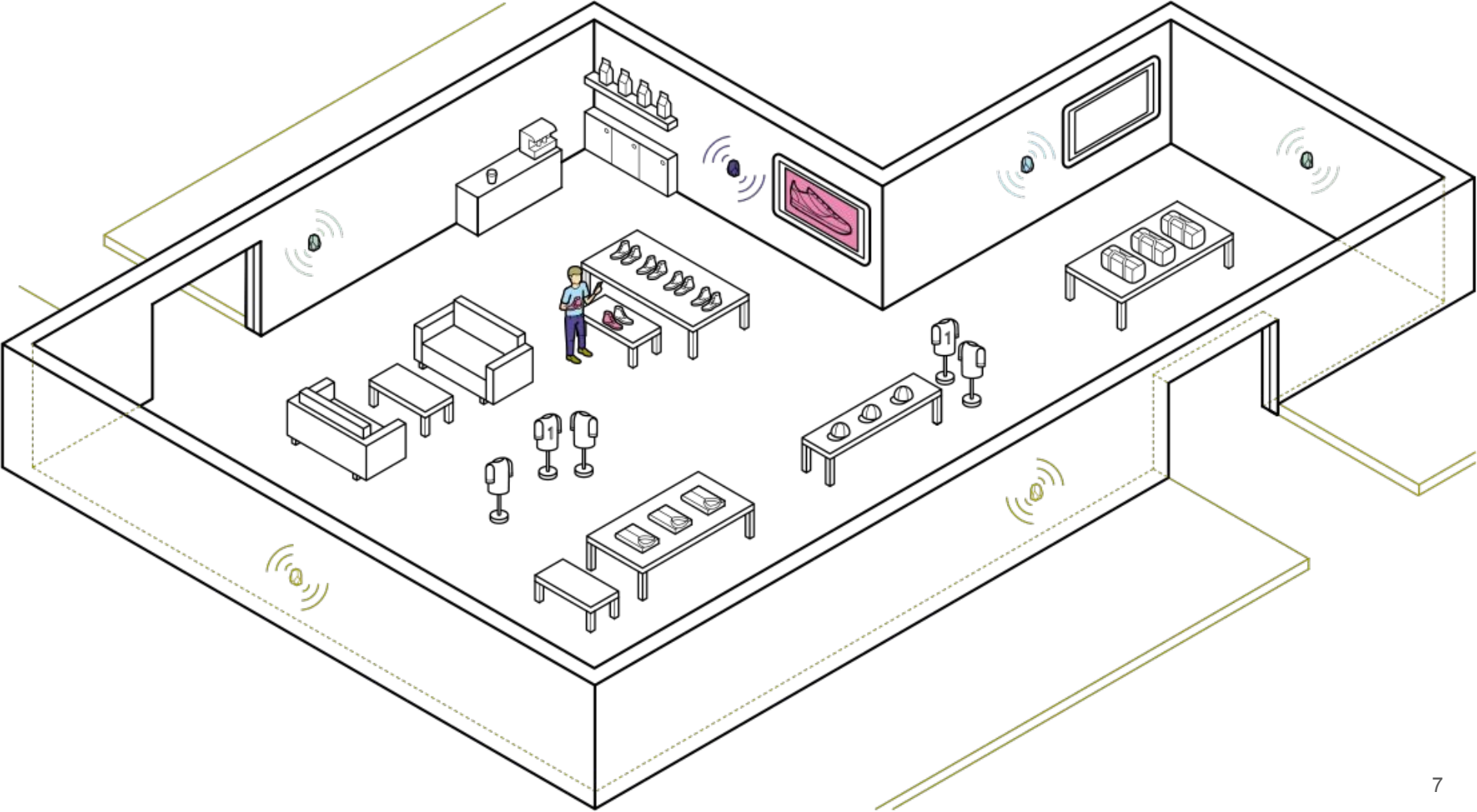
About Me

- Daniel Hartwich
- Android Developer in the Framework Team @ XING
- XING: https://www.xing.com/profile/Daniel_Hartwich
- Google+: https://plus.google.com/+DanielHartwich1
- Twitter: https://twitter.com/KiLLyA_

Introduction

Introduction

- Bluetooth Beacons are sending out information continously to their environment
- Beacons should be interpreted more as a **context** (not only as a position)
- If you know the location as well you have a correlation between context and position
  - different ways of getting a users location (QR-Code, Bluetooth, WIFI, Cell ID etc.)
- Your app can now react on this beacons and display appropriate information according to the data received by the beacon



Opportunities

- Bus Stop: Show arrival dates
- Detect Indoor Location (Navigation)
- Contactless Payment
- Proximity Marketing
- Check-in coupons
- Presence detection (check if user is nearby an object to show additional info about this object e.g. A plant that reminds you to water it)
- Tracking (beacon in each room to detect where you are)
- Security (why not put a beacon in your bag?)

Eddystone

Eddystone

Plenty of different beacons:

- AltBeacon
- Apple’s property: iBeacon
- Estimote
- Gimbal
- PayPal Beacon
- yoints...

- Bluetooth Smart Beacon
  - **Eddystone (based on AltBeacon)**
    - https://github.com/google/eddystone
    - Announced by Google in June 2015

Eddystone, whats so special?

- Openness
  - It is an open Bluetooth 4.0 protocol
  - While iBeacon is officially supported by iOS devices only, Eddystone has official **support for both iOS and Android**

Packet types / frames

- Eddystone UID (identifier)
  - Namespace as UUID
  - Instance (6 bytes) much like major and minor
- Eddystone-URL
- Eddystone-TLM (telemetry)
  - battery voltage
  - temperature
  - number of packets since last reboot
  - beacon uptime since last reboot
- ...

Hardware

- Phones can also be Smart Beacons themselves

- TxEddystone UID (https://github.com/google/eddystone/tree/master/eddystone-uid/tools/txeddystone-uid)

- Almost all devices with BLE can become smart beacons


accent
systems
inspiring electronic
solutions


BKON


blesh
blending experiences


BLUECATS


bluvision


®


estimote


GIMBAL™


kontakt.io


KST


orange™


PassKit

Some Bluetooth Info

- Bluetooth / Bluetooth LE is a wireless protocol
- Signal Strength is an indicator for proximity (RSSI - Received signal strength indicator)
- BLE has reduced power consumption
- Bluetooth SIG predicts more than 90% of Bluetooth-enabled smartphones will support the low energy standard by 2018
- Signal (RSSI Strength) is effected by reflection and attentuation (e.g. fluids)

Codelab

Eddystone

Codelab - Prerequisites

- Real device with BLE (emulator has no Bluetooth support)
- Android 4.3 (Jelly Bean, API Level 18)
- Android 4.4.4 (KitKat fixes some issues)
- Android 5.0 (Lollipop, API Level 21) recommended due to some API changes (e.g. Advertisement and LE Scanner)
- Android 6.0 (Marshmallow requires the permission for Location)

Let’s get started

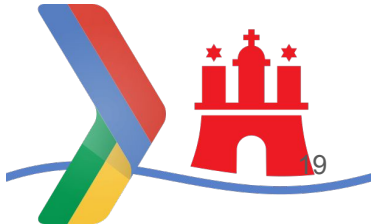
- Download and install: Android Studio + JDK 7
- Create an account and register your beacon at https://cloud.estimote.com/ || https://cloud.estimote.com/#/activateBeacon
- Install the Estimote app for Android, Log in with your account and find your beacon
- In this CodeLab we’ll build an example app without using the Nearby API since it required a lot of overhead and time is money

Codelab

Eddystone

Codelab

1. Create a basic android app with an empty screen
2. Constantly scan for all beacons nearby and Log their address
3. Filter to only receive information about your beacon
   a. Use the namespace + instance information for that
   b. Beacons are sending their UID by default (Frametype: 0x00)
4. Refer to the documentation found on https://github.com/google/eddystone/tree/master/eddystone-uid to find out more about the data received
5. Calculate the distance between your phone and your beacon using the following formulae: 10^ ((txPower at 0m - rssi) - 41 / 20.0)
6. Show the distance in your app and update it whenever you receive new information from your beacon

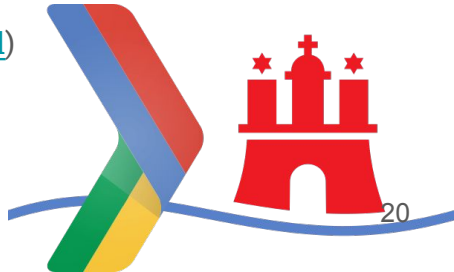


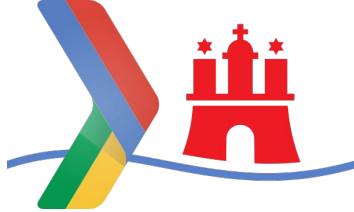
Codelab Part 2 - Okay, that was too easy for me!

Beacons are also sending a primary packet each 12? seconds with some telemetry data

1. Extend your app to also showing the temperature of your beacon
   a. Make use of the telemetry data (https://github.com/google/eddystone/tree/master/eddystone-tlm)
2. Define some temperature ranges and switch the display color according to these ranges.
3. Extend your app in any way you want (maybe change your beacon to broadcast an url (https://github.com/google/eddystone/tree/master/eddystone-url)

Eddystone





Useful links

Sample app repo: https://github.com/dhartwich1991/BeaconWorkshop

Eddystone Repo: https://github.com/google/eddystone

Eddystone UID Info: https://github.com/google/eddystone/tree/master/eddystone-uid

Eddystone TLM Info: https://github.com/google/eddystone/tree/master/eddystone-tlm

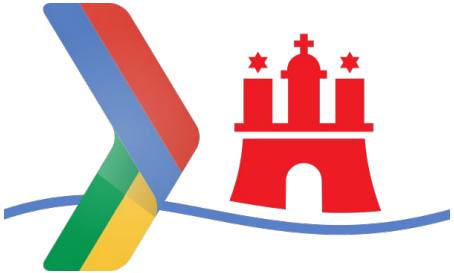
Eddystone URL Info: https://github.com/google/eddystone/tree/master/eddystone-url

Nearby API get started: https://developers.google.com/nearby/messages/android/get-started

Proximity API get started: https://developers.google.com/beacons/proximity/guides

Discussion Round

Ideas, problems, use cases



Thank you very much!