Detailed instructions for using the Proxity mobile phone application (Android, iOS)

The Proxity app is a demo application for the Proxity framework. It is adequate to attach the physical world to cloud services in a basic way by defining simple webhooks. Any functionality beyond that requires to build custom apps with the framework.

In the app, there is a slight distinction between privileged users who define regions (content creators) and ordinary users who can run the app with less permissions. Of course all users can assume both roles anytime, but content creators need GPS enabled and need to be logged in whereas ordinary users do not need to grant these privileges.

In the Proxity app, privileged users define tags of interest (e.g. #cloud or #food) on their mobile devices. Once they or other ordinary users are in a region that has been defined to carry that tag, a notification is triggered, that can be an in-app message and/or a simple webhook to an external service.

** Note: The app does a binary match (Bluetooth device found or not found). It does not inform about the signal strength (RSSI) and is therefore not suitable to build services where the distance to any Bluetooth device (in particular beacon) is essential.

Proxity app links:

https://www.proxity.app/
App downloads (stable version) for both Android and iOS – June 2021
→ https://apps.apple.com/ca/app/proxity-discover-digital-dimension/id1174965266
→ https://play.google.com/store/apps/details?id=eu.proxity.and

** Note: The Android stable version has a bug where you cannot define regions because photos cannot be selected in the region definition on certain devices (e.g. Lenovo tablet). On Android you **should** use the development version in this case. In fact for any student work using the development version is **highly recommended** to benefit from new features including those that are brought in from student feedback.

https://testflight.apple.com/join/dzADor2R
App download (development branch) for iOS.

https://play.google.com/apps/testing/eu.proxity.and
App download (development branch) for Android.

** Note: This works like a signup to updates; you will need to **wait some hours** until you receive the update. An APK file for direct updates is not available.

https://proxity.slack.com/
Developer chat where Proxity developers can be reached directly in case of specific questions.

Proxity app instructions:

Please refer to the Proxity User Guide v1.1 for detailed step-by-step instructions. Here just a few additional technical explanations are given. Place the beacon. Open the beacon box by twisting the cap and press the small button for more

than two seconds until a blue light turns on for a short duration. (If it is blinking instead you inadvertently turned it off!)

Start the app. From a user perspective, this requires only WiFi + Bluetooth enabled. However, to define a webhook, location (i.e. GPS) also needs to be enabled. The app will behave strangely if this is not the case.

Proceed to create regions of interest («My» tab). Log into Proxity upon first use in that tab. Once logged in you should see an add button («+» sign).

Create a new region by drawing with the finger. Assign it a name. Set up a webhook pointing to an HTTP endpoint provided by your service. You can also set up messages with tags for testing. Tags work like notification filters – notifications will only be triggered if the user has defined matching tags of interest. However, for the webhook you do not need tags and instead would do any filtering within your service that exposes the endpoint.

** Note: Do **not expect** the Bluetooth beacons or any other devices to show up on the map immediately while defining the region. Although counter-intuitive from a developer perspective, the app only captures knowledge about Bluetooth devices internally whereas the shown map comes from the Proxity server where this information is not yet registered and will show up around one hour later. This map can be verified on the Proxity website www.proxity.eu.

** Note: The way the app works is as follows: Upon region definition, it fingerprints **all Bluetooth devices** that are presumably within the defined region (using signal strength estimation) and transmits these fingerprints to the Proxity server. When a user encounters such a device with the Proxity app running, even when the device has moved position (because users do not need to have GPS enabled), it will trigger the notification. The app focuses on privacy and therefore does not differentiate devices by MAC address from the user perspectve.

Webhook format:

There is a test service available, hosted on Google Cloud Run, that ingests web hooks and summarises them. Your service should accept the same message format via HTTP POST.

```
curl -X POST -d '{"id": "test1", "data": "test2", "device_data": "test3"}' \
       https://webhook-test-e5re7ipn5a-oa.a.run.app/webhook

curl https://webhook-test-e5re7ipn5a-oa.a.run.app/stats
```

Proxity framework links:

** Note: Use these only if you want to do mobile application development to achieve functionality beyond the Proxity app. This is experimental and most likely requires getting in touch with the Proxity developers on their Slack channel due to absence of documentation.

https://github.com/proxity-eu/proxity_flutter
Recommended framework for Android (Kotlin) and iOS (Swift)
→ https://pub.dev/packages/proxity_flutter/example (Dart)
Getting started: currently undocumented

https://github.com/proxity-eu/ProxityKit
Only for iOS, only binary format for now. Will be updated during December 2021.
Getting started: currently undocumented