

THE HUMAN CONTEXT

Exploring Google's Nearby APIs



Stuart Kent, @skentphd

Mobile developer @ Detroit Labs



2½ years



9 months



~2 months

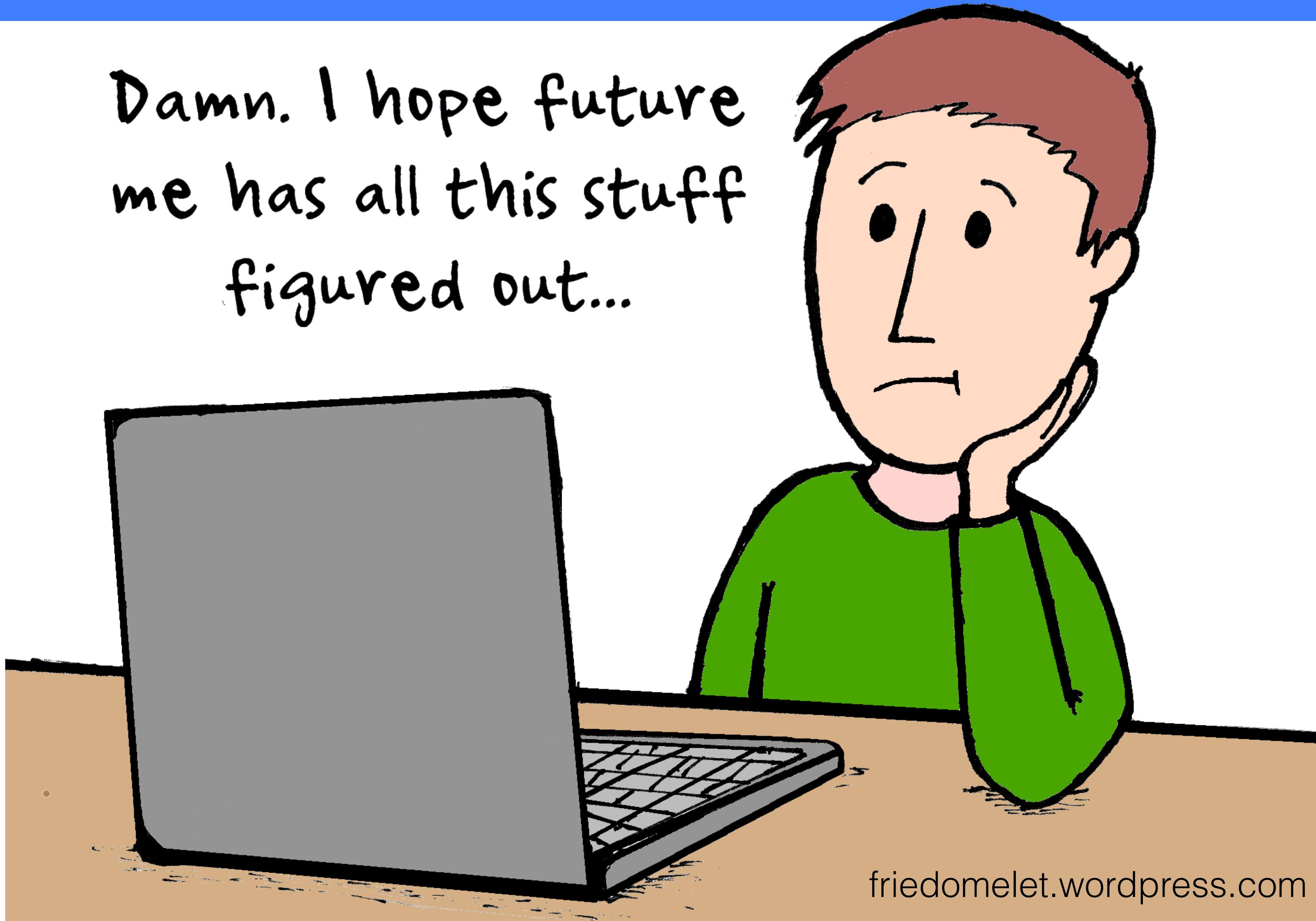
The Nearby APIs

"Build simple interactions between
nearby devices and people"

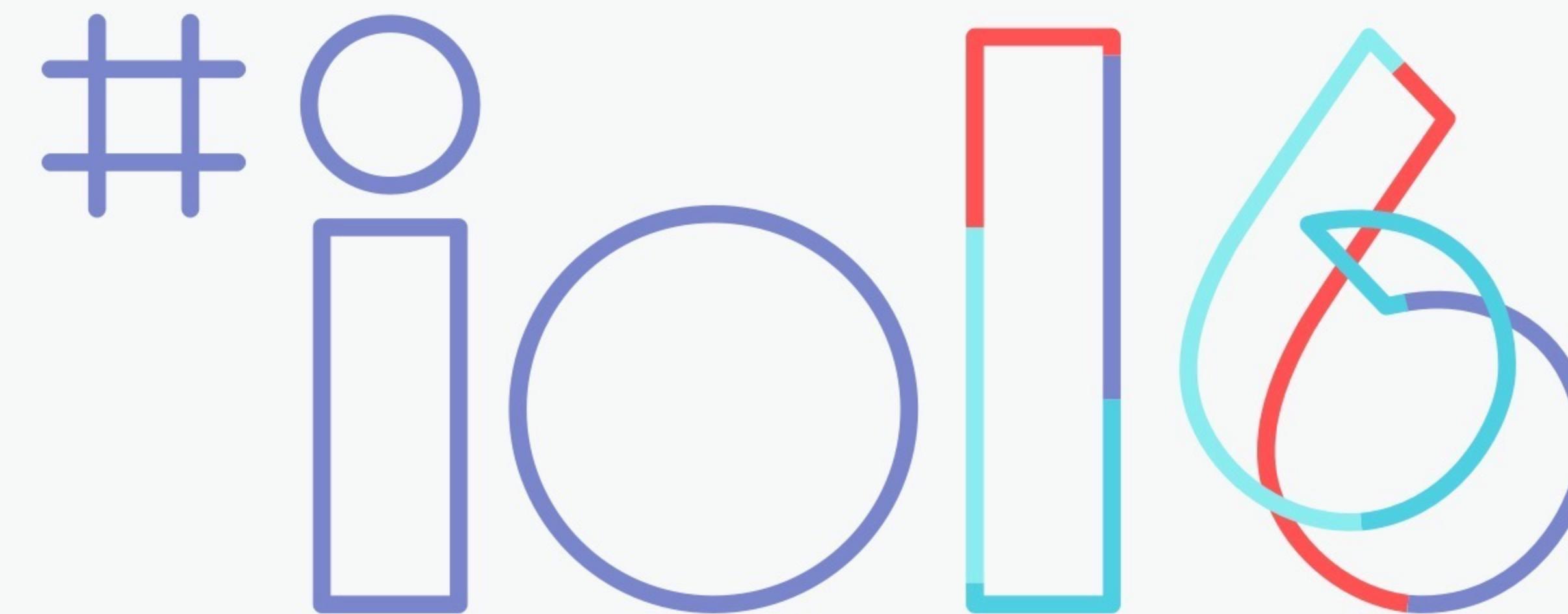


The Process

Damn. I hope future
me has all this stuff
figured out...



Google I/O



Roadmap

Background & Motivation

Roadmap

Background & Motivation



Capabilities

Roadmap

Background & Motivation



Capabilities



Practicalities

Roadmap

Background & Motivation



Capabilities



Practicalities



Responsibilities

Roadmap

Background & Motivation



Capabilities



Practicalities



Responsibilities



Opportunities

BACKGROUND & MOTIVATION

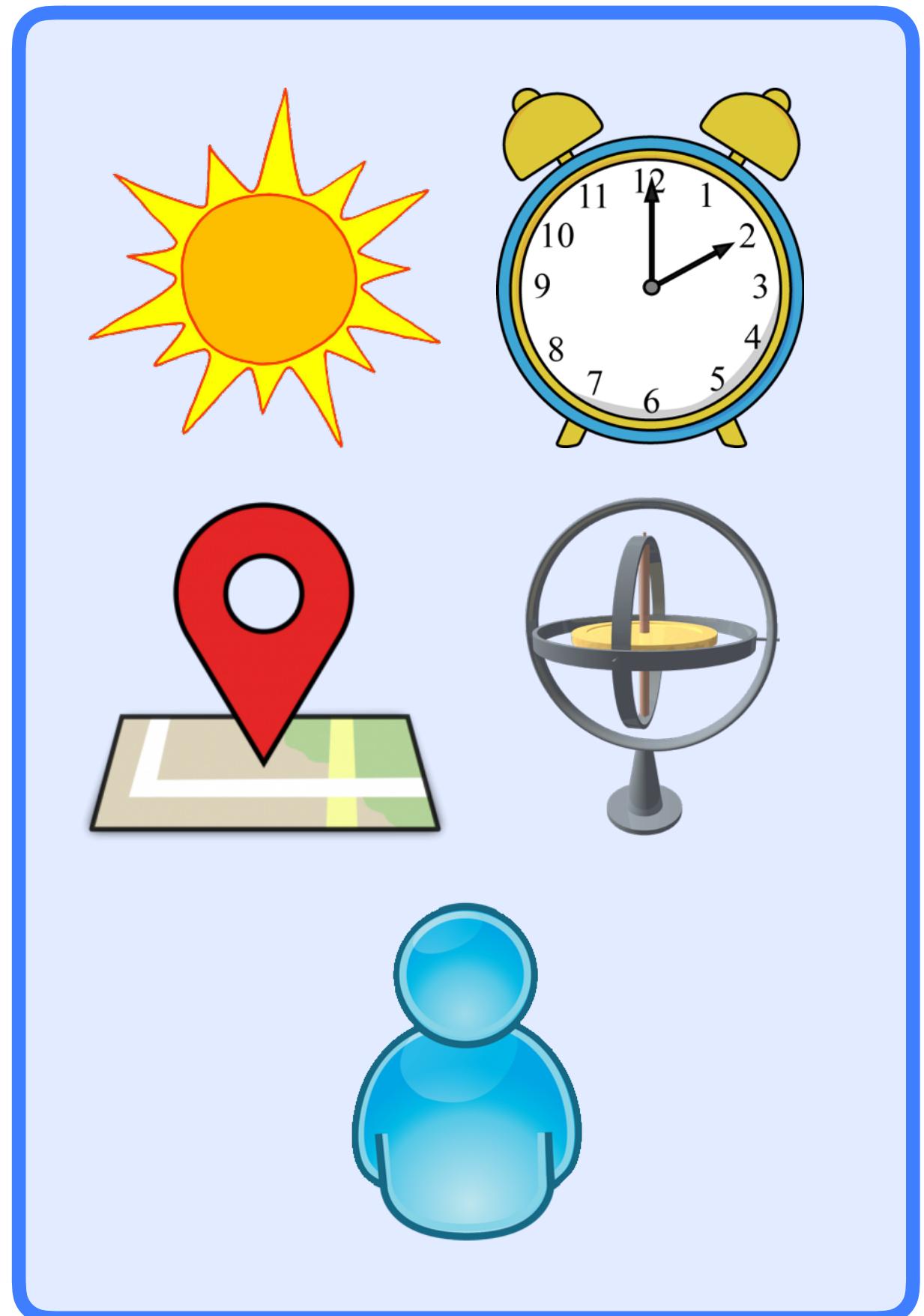
Context Aware Apps

Context Aware Apps

**Observe, interpret, and act based
on their environment.**

Context Aware Apps

Environment

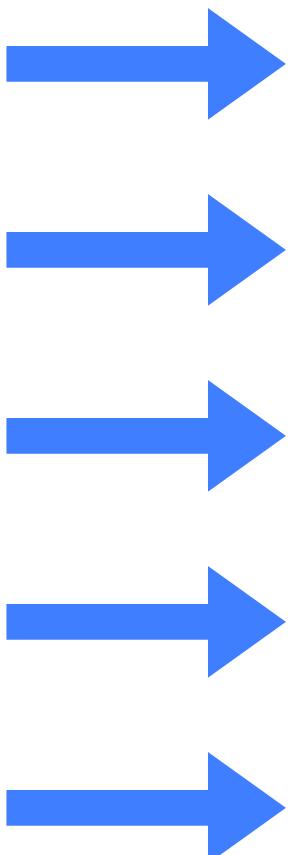
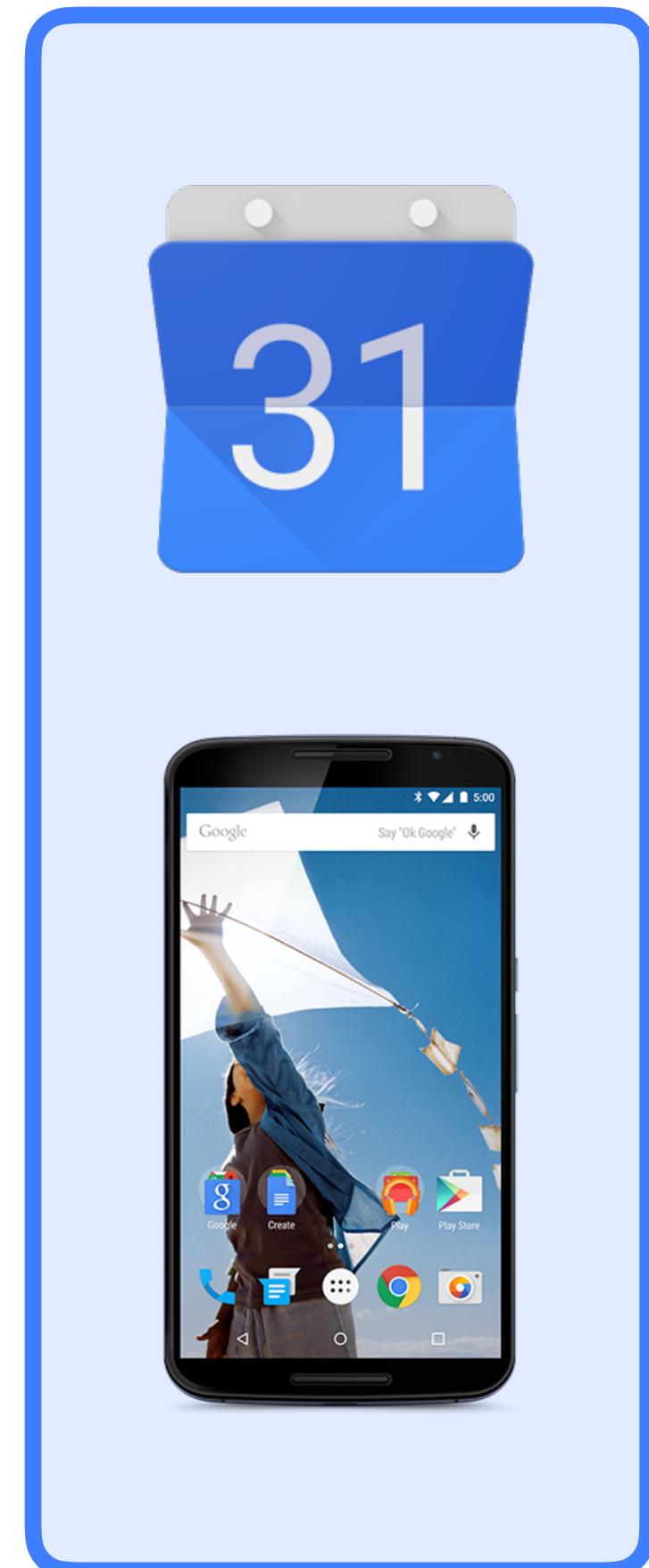


Context Aware Apps

Environment

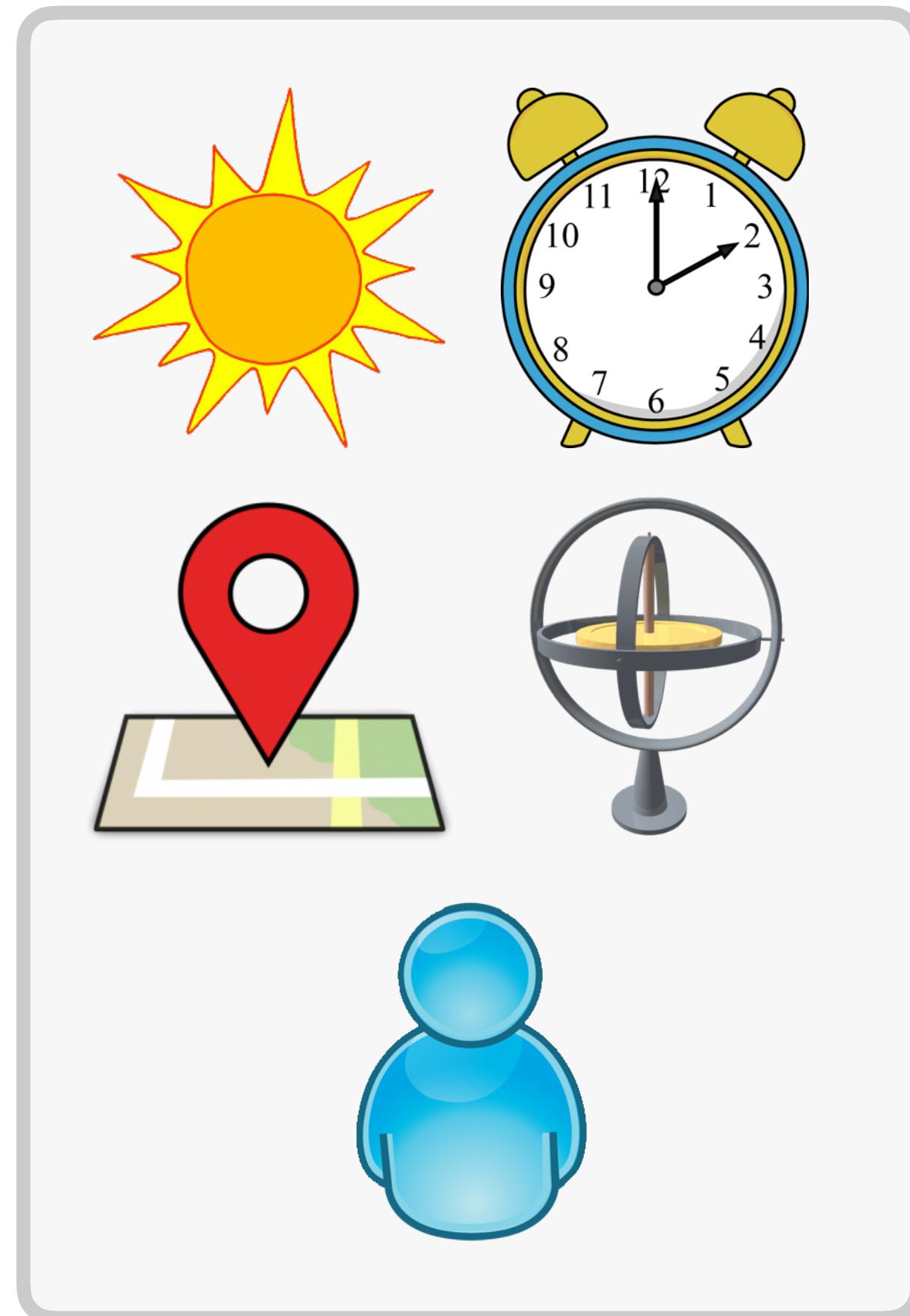


Observers

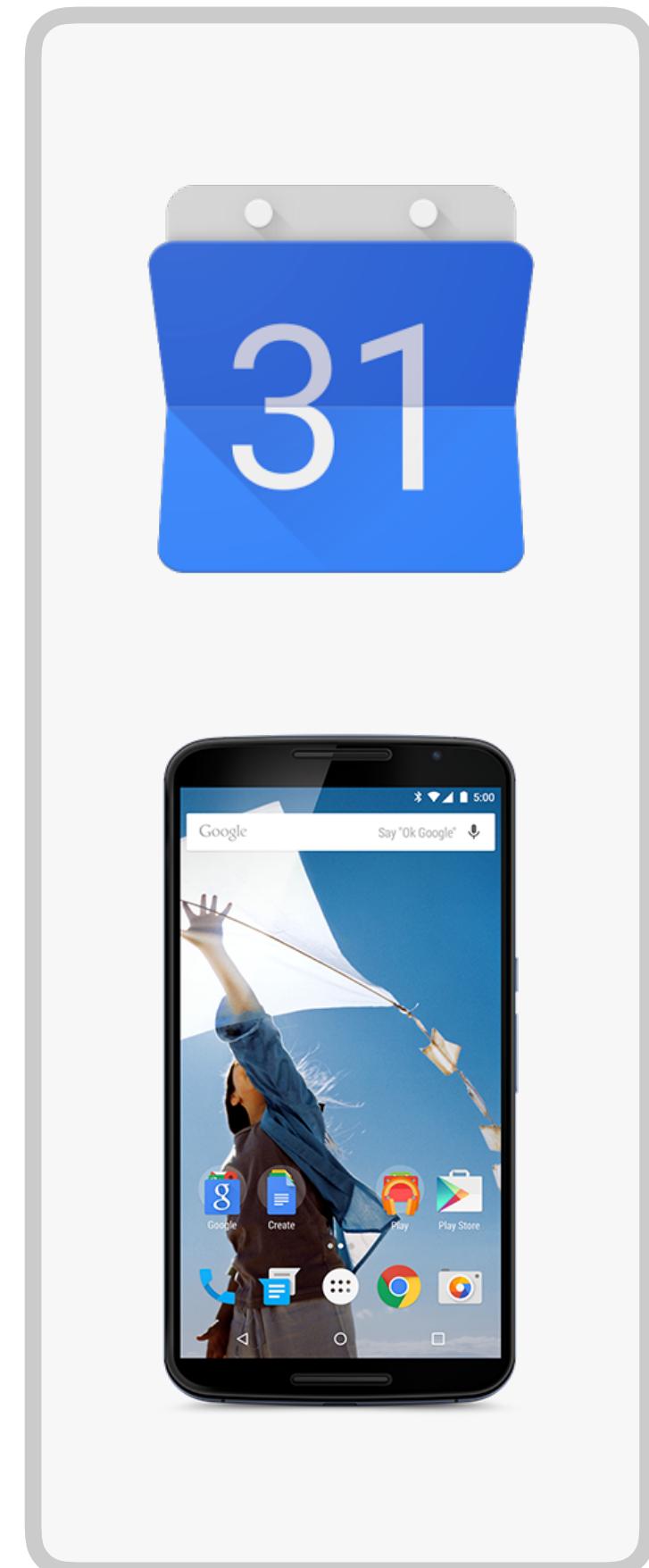


Context Aware Apps

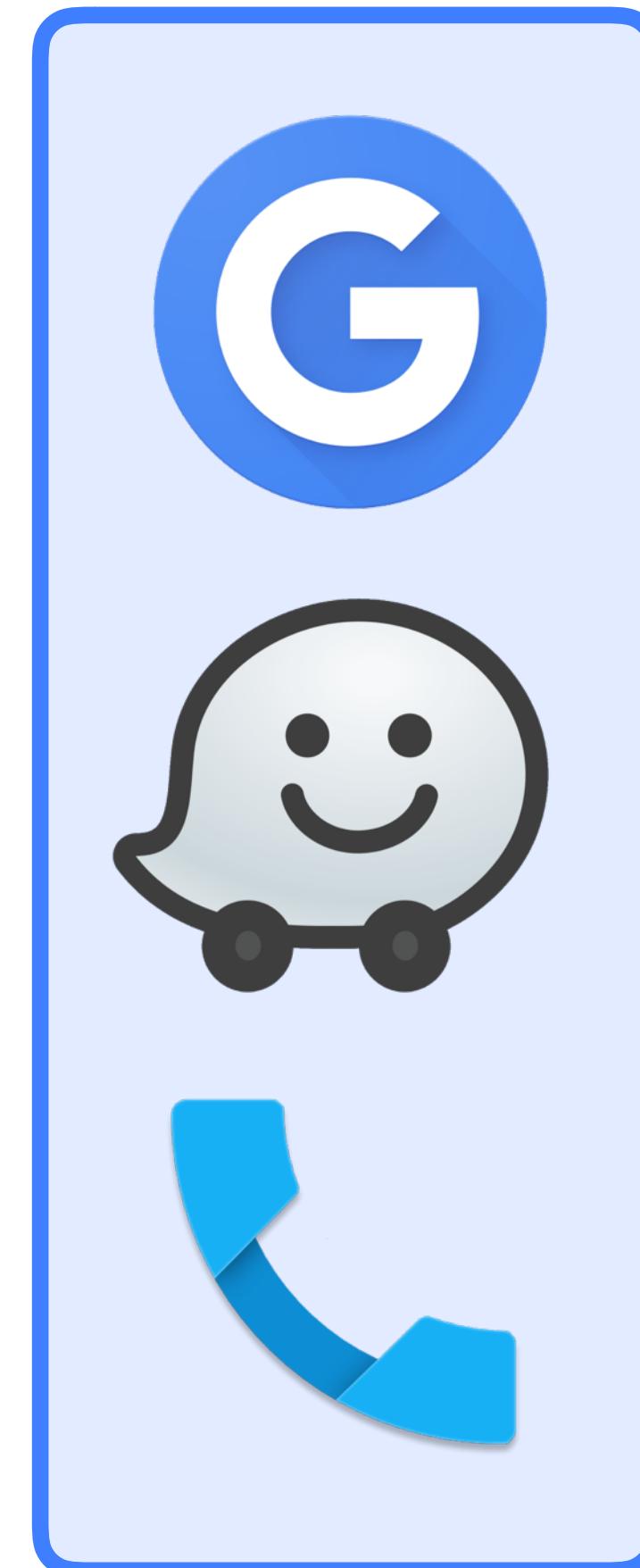
Environment



Observers



Interpreters/Actors



2001



"My internet browser heard us saying the word Fry and it found a movie about Philip J. Fry for us. It also opened my calendar to Friday and ordered me some french fries."



2016

Ticket type: World Perk Rewards Premier Access

 View email

Car rental

Economy 2 door sedan

Hertz rental car reservation

Name
Mr. John Smith

Booking Number
E12345678

Thu, 18 Apr, 2013
11:40

Fri, 26 Apr, 2013
21:50

Hertz San Diego
987 Harbor Dr, San Diego, CA 92101

 Get directions

 Manage reservation

 View email

Next Appointment

Agency Meeting
11:30 AM

Ninth Ave, New York, NY 10011

 Email guests

San Francisco



63°

SCATTERED CLOUDS	TUE	WED	THU	FRI
 5mph				
10%	68°	67°	65°	57°

68°

48°

67°

44°

65°

48°

57°

46°

Hotels

The Connaught Hotel

Carlos Place, Mayfair, London W1K 2AL, United Kingdom

Check in from 12:00pm today



 Call

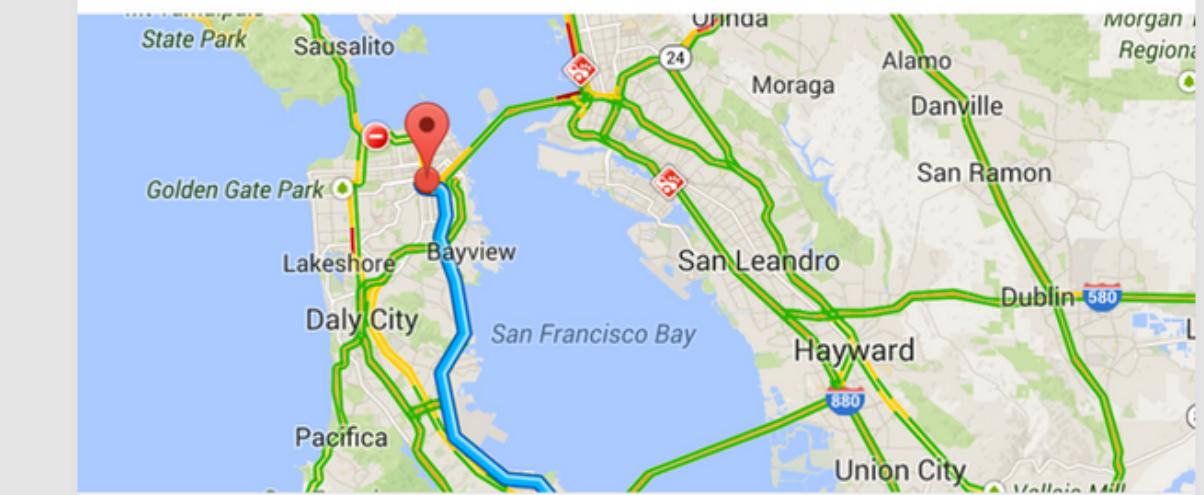
 Hotel information



Traffic & Transit

57 mins to work

Normal traffic on US - 101



 Navigate / 57 mins via US - 101

Restaurant Reservations

Broder

2508 SE Clinton St, Portland, OR 97202

Reservation in 1 hour

Travel time walking 45 minutes



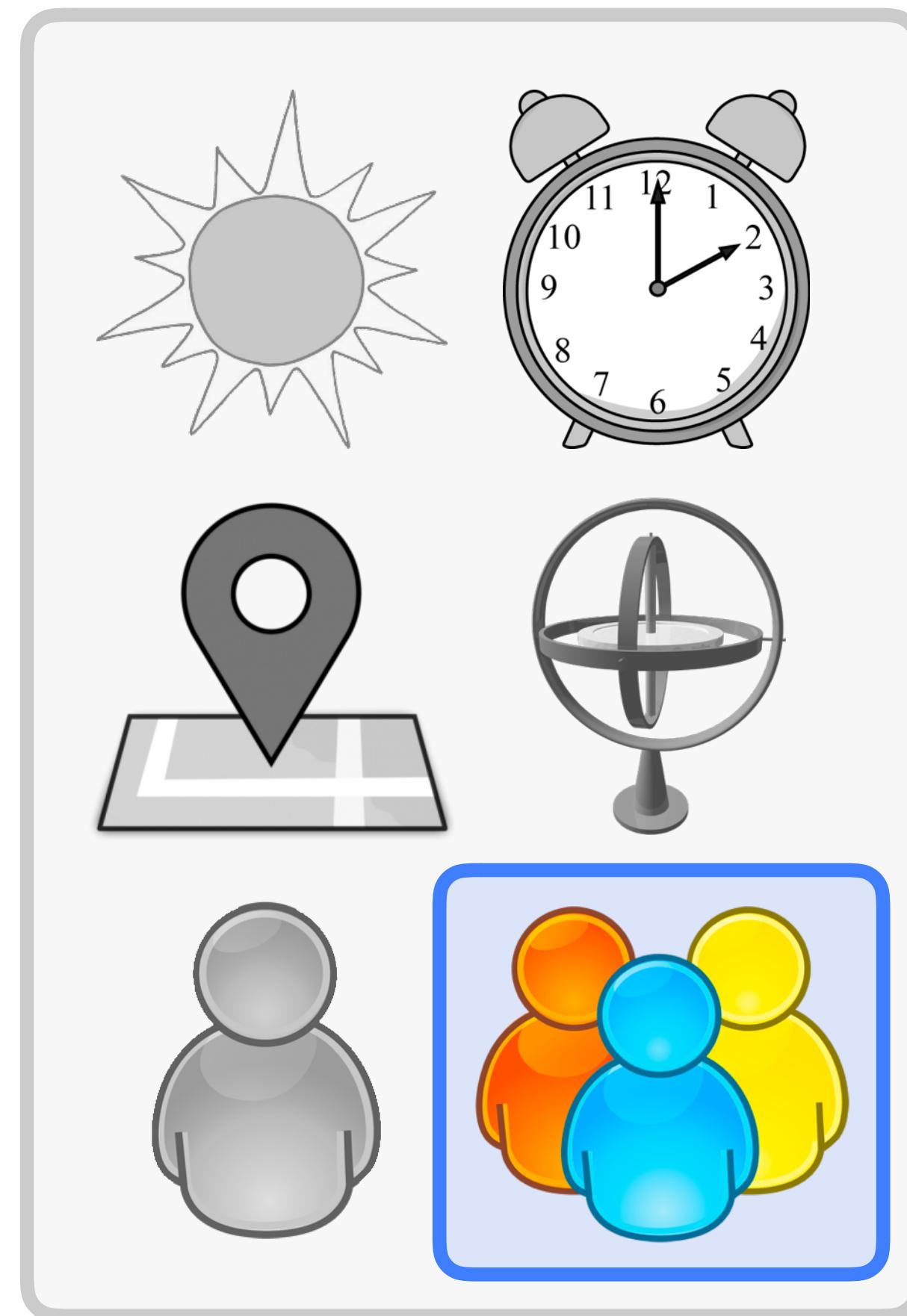
The Human Context



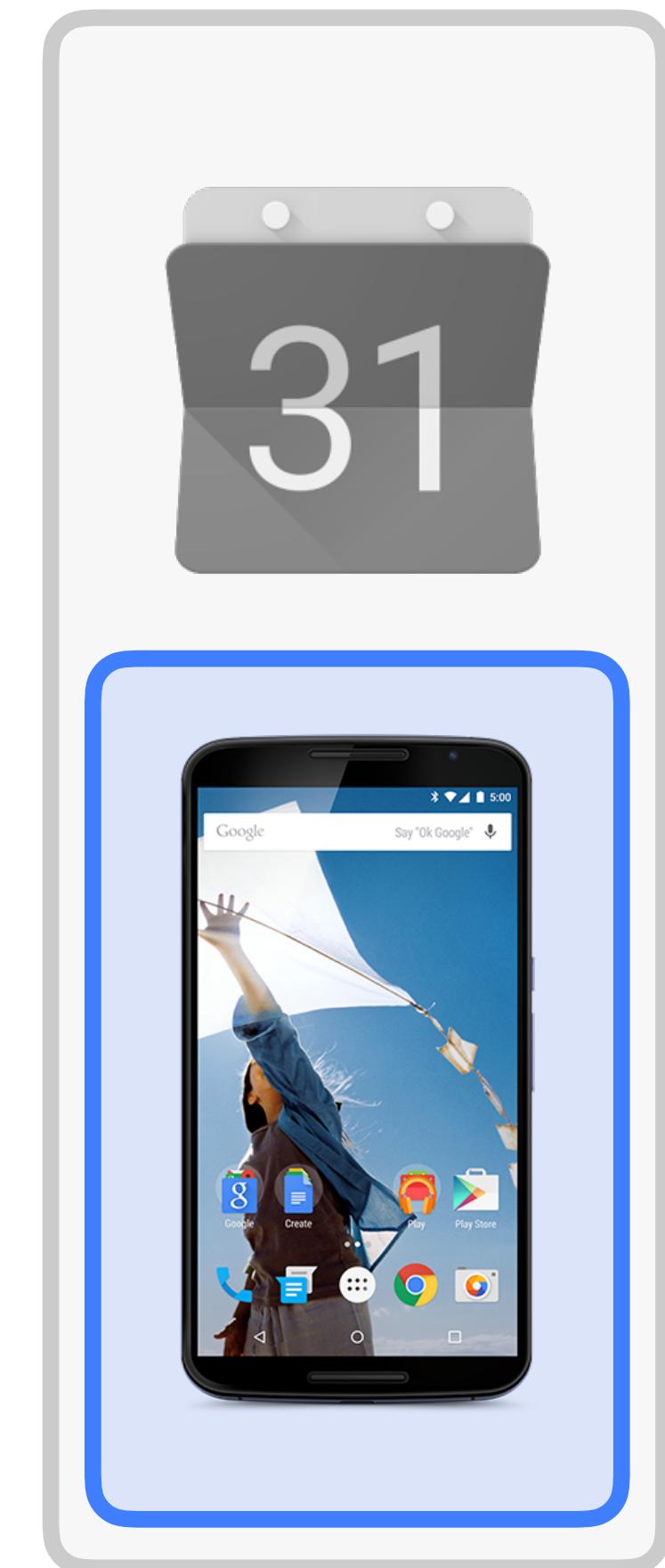
**"Man is by nature
a social animal"**

The Nearby APIs

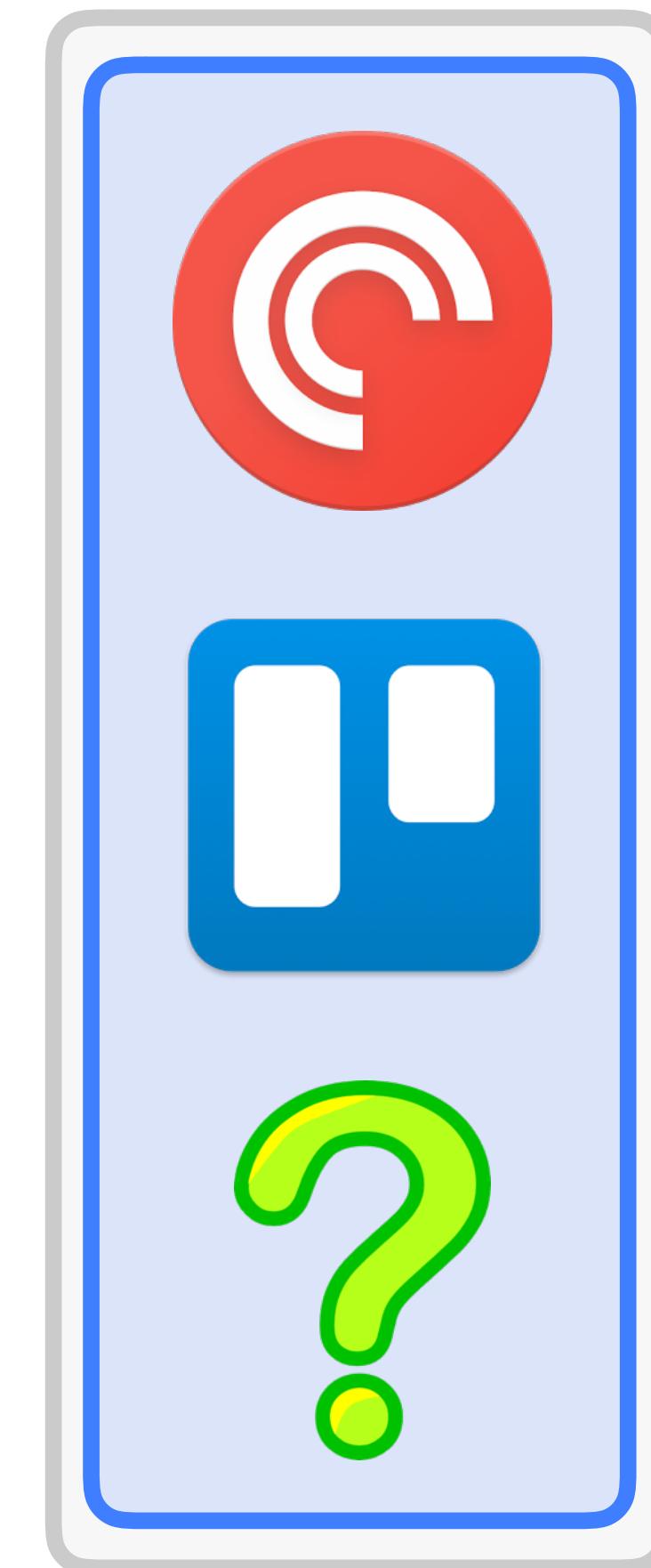
Environment



Observers



Interpreters/Actors



Refined

Easier

CAPABILITIES

The Nearby APIs

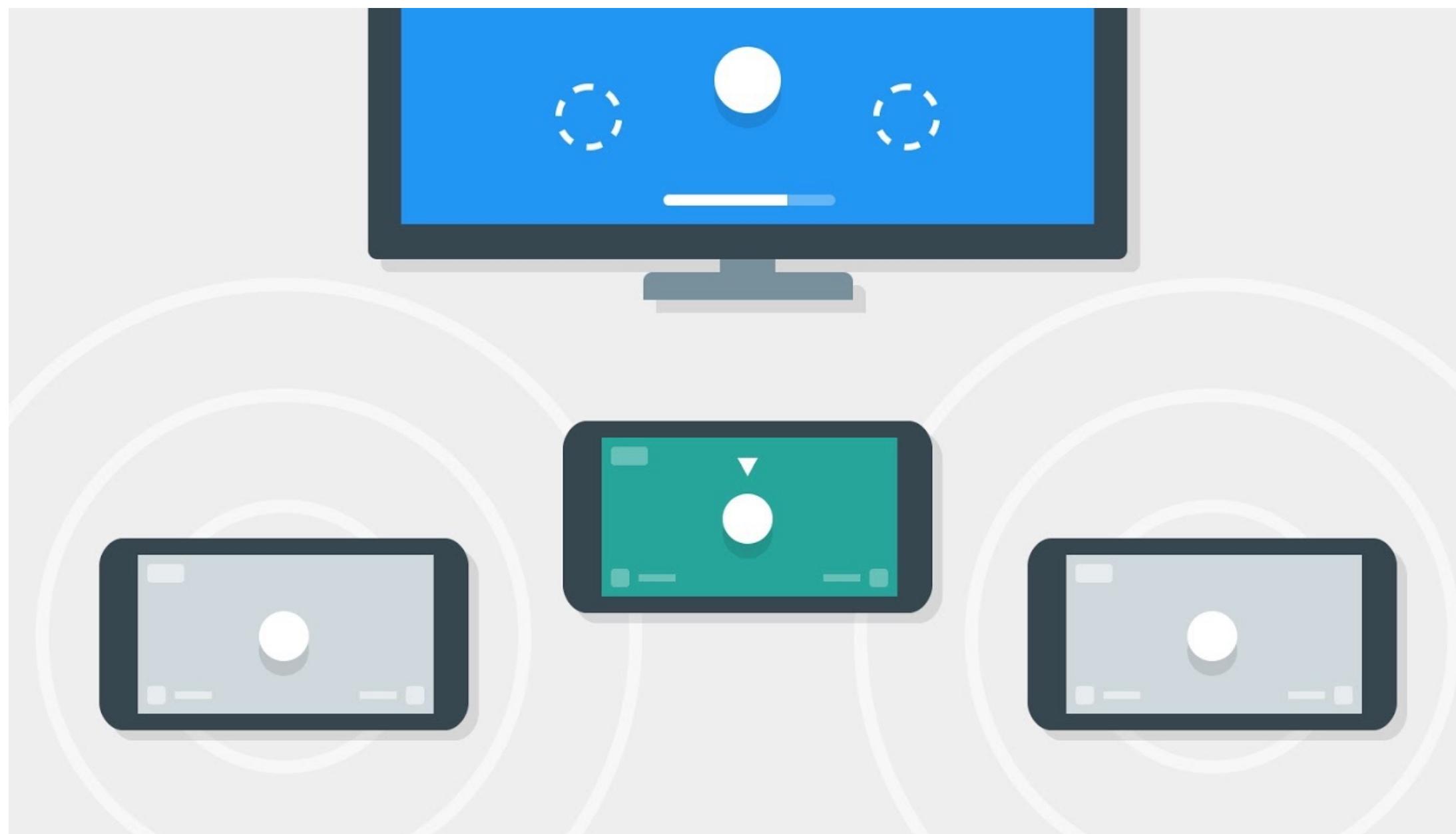
"Build simple interactions between
nearby devices and people"



Current Categories

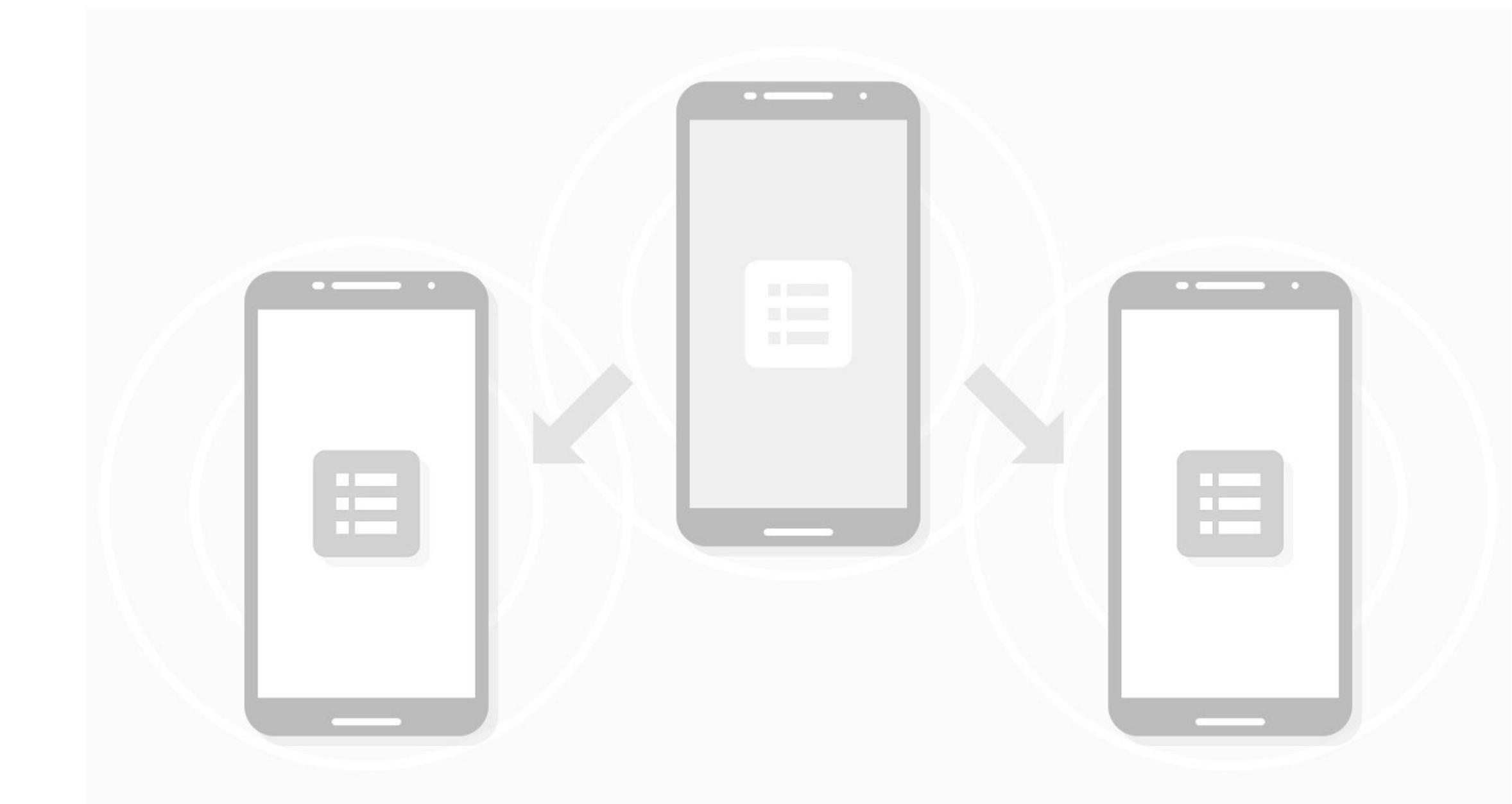
Nearby Connections

Local Multiplayer Games



Nearby Messages

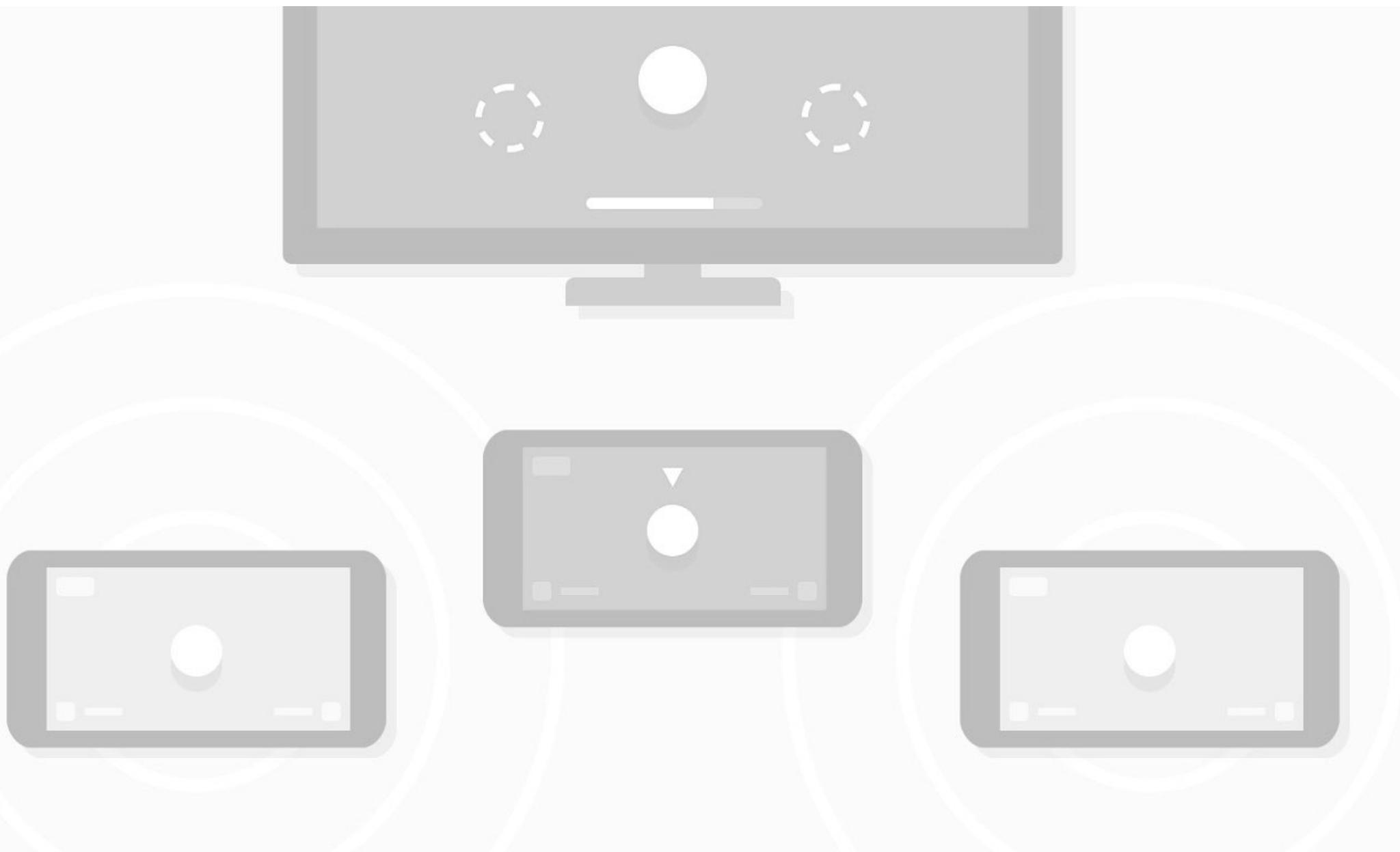
General purpose



Current Categories

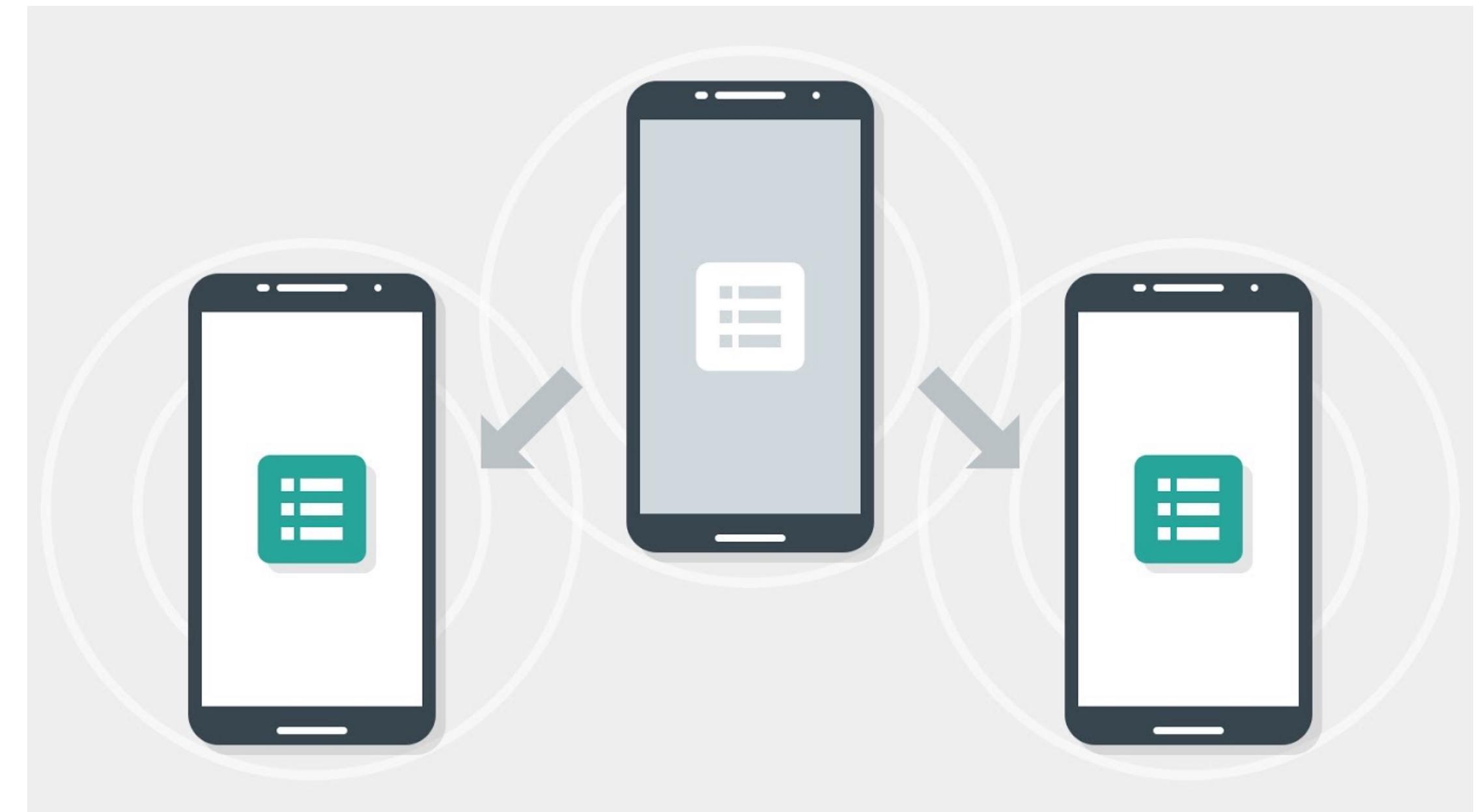
Nearby Connections

Local Multiplayer Games



Nearby Messages

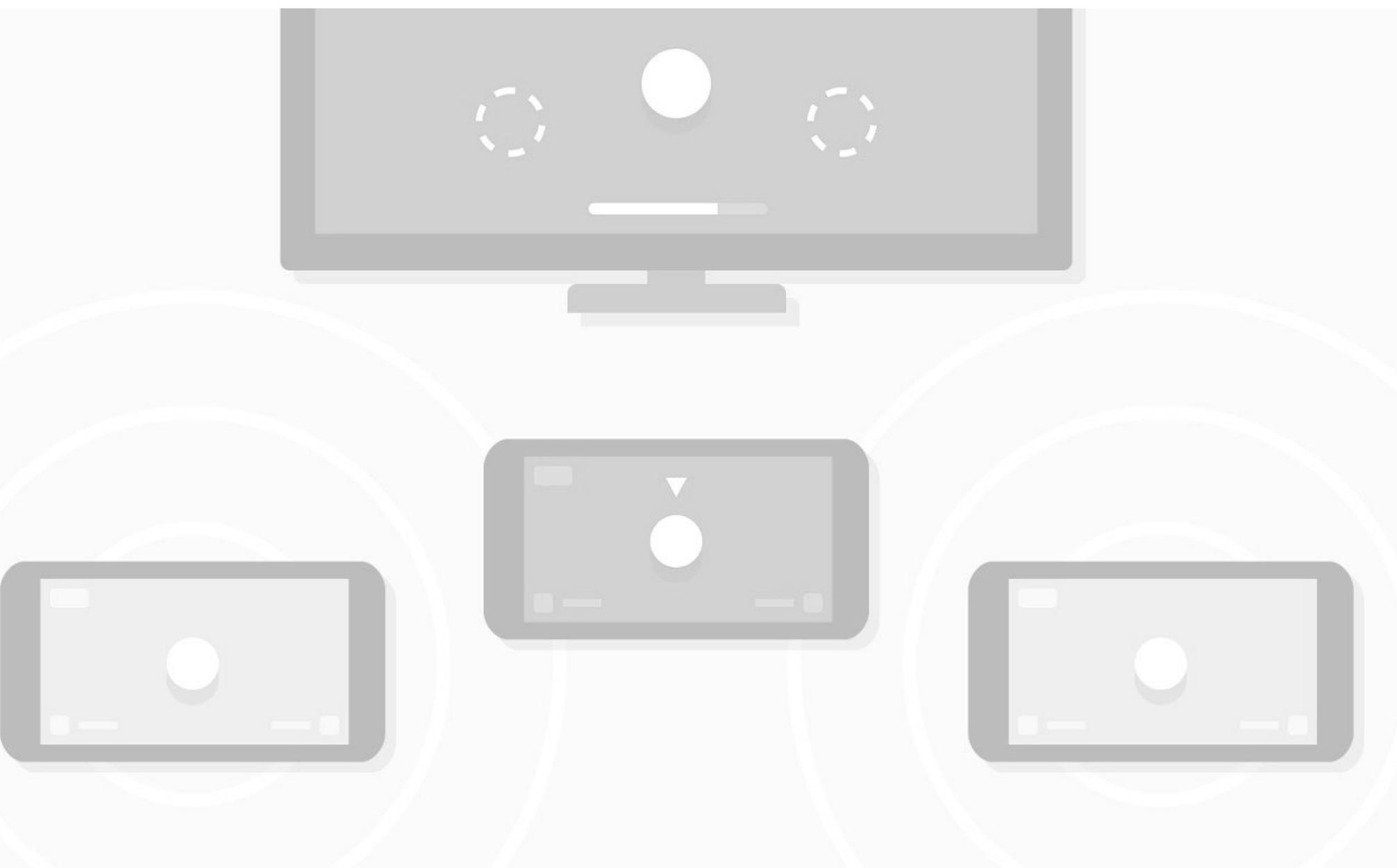
General purpose



Current Categories

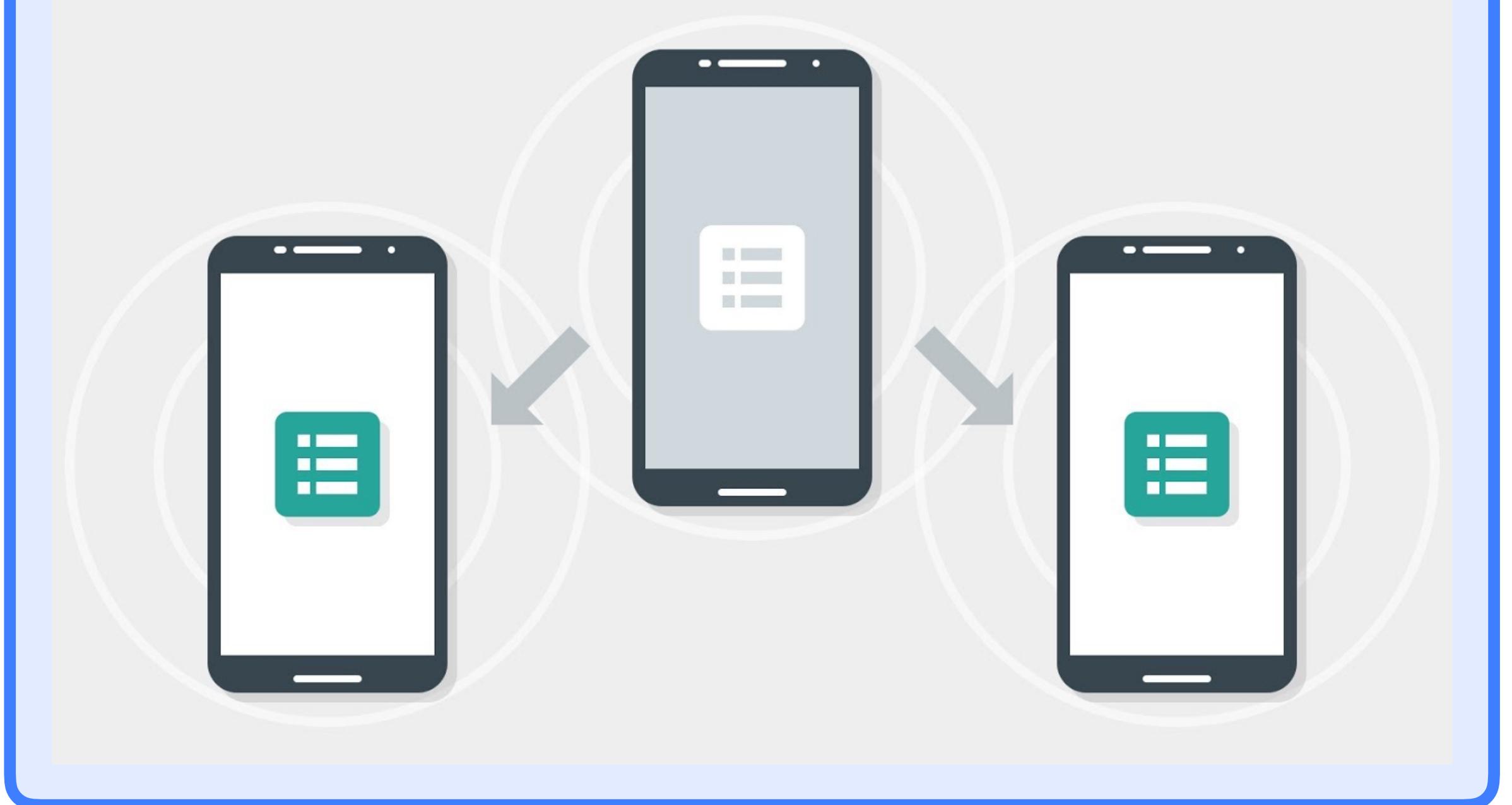
Nearby Connections

Local Multiplayer Games



Nearby Messages

General purpose



The Nearby Messages API

- detect and communicate with nearby devices

The Nearby Messages API

- detect and communicate with nearby devices
- simple pub/sub API

The Nearby Messages API

- detect and communicate with nearby devices
- simple pub/sub API
- iOS & Android

The Nearby Messages API

- detect and communicate with nearby devices
- simple pub/sub API
- iOS & Android
- unauthenticated*

The Nearby Messages API

- detect and communicate with nearby devices
- simple pub/sub API
- iOS & Android
- unauthenticated*
- default (~100ft) and earshot 🧑 (~ 5ft) ranges

The Nearby Messages API

- detect and communicate with nearby devices
- simple pub/sub API
- iOS & Android
- unauthenticated*
- default (~100ft) and earshot  (~ 5ft) ranges
- [BLE beacon support]

Roles

Publisher

0 or 1 messages

Subscriber

Multiple messages

Roles

Publisher

0 or 1 messages

Handheld (dynamic)

Subscriber

Multiple messages

Handheld

Roles

Publisher

0 or 1 messages

Handheld (dynamic)

+ Beacon (static)

Subscriber

Multiple messages

Handheld

Roles

Publisher

0 or 1 messages

Handheld (dynamic)

+ Beacon (static)

Subscriber

Multiple messages

Handheld

Messages

- Byte array (JSON, protocol buffers)

Messages

- Byte array (JSON, protocol buffers)
- Recommended: **< 3kb (Android); "fairly small" (iOS)**

Messages

- Byte array (JSON, protocol buffers)
- Recommended: < 3kb (Android); "fairly small" (iOS)
- Max: **100kb (Android & iOS)**

Messages

- Byte array (JSON, protocol buffers)
- Recommended: < 3kb (Android); "fairly small" (iOS)
- Max: **100kb (Android & iOS)**
 - ↳ Send references, not resources

PRACTICALITIES

Communication Methods

Discovery & Messaging

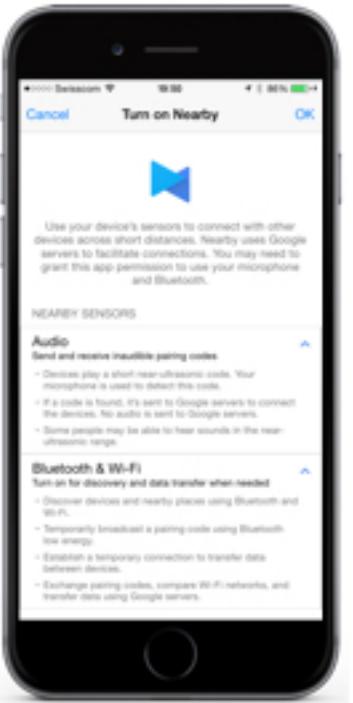


Image credit: p2pkit.io

Discovery & Messaging

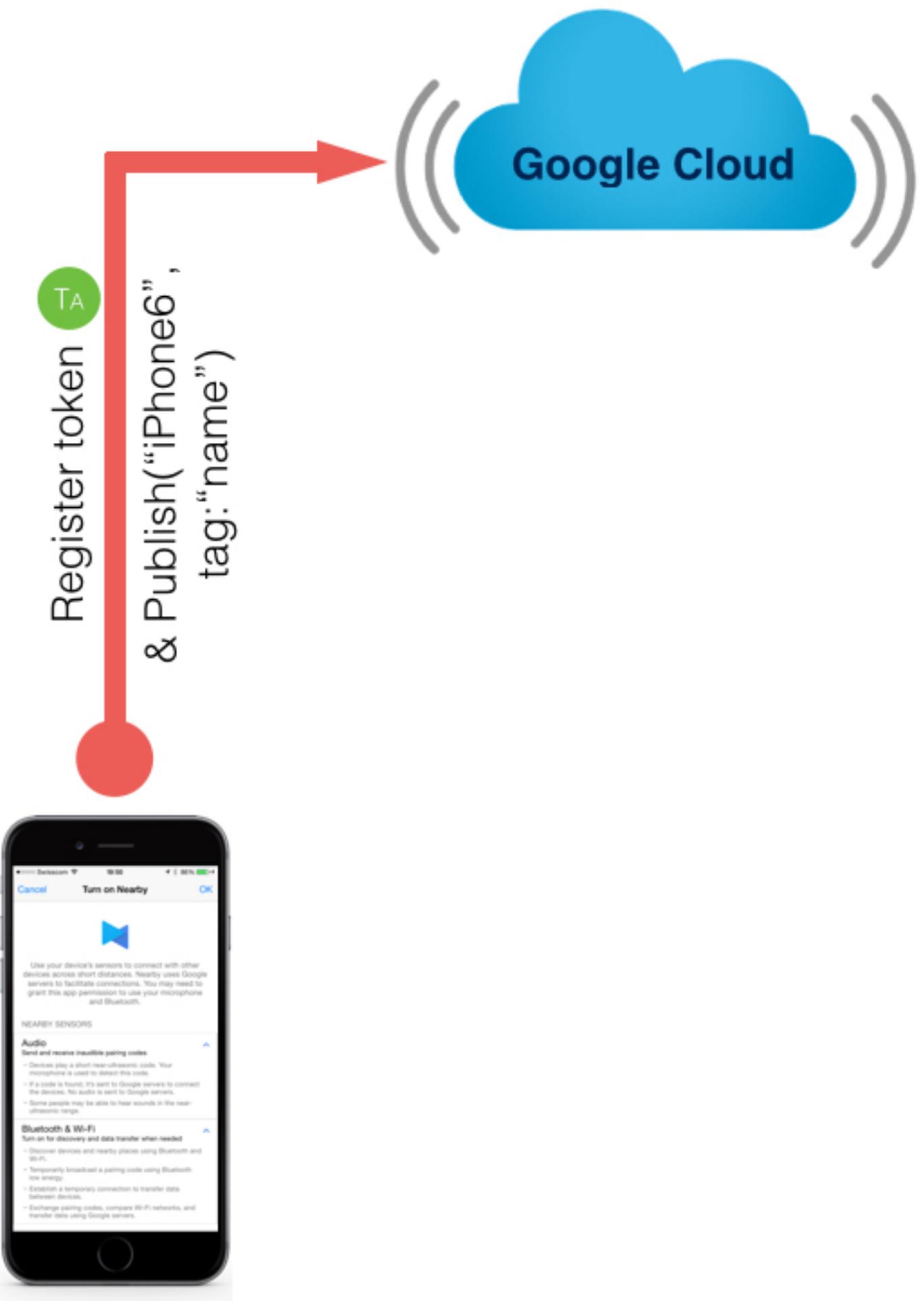


Image credit: p2pkit.io

Discovery & Messaging

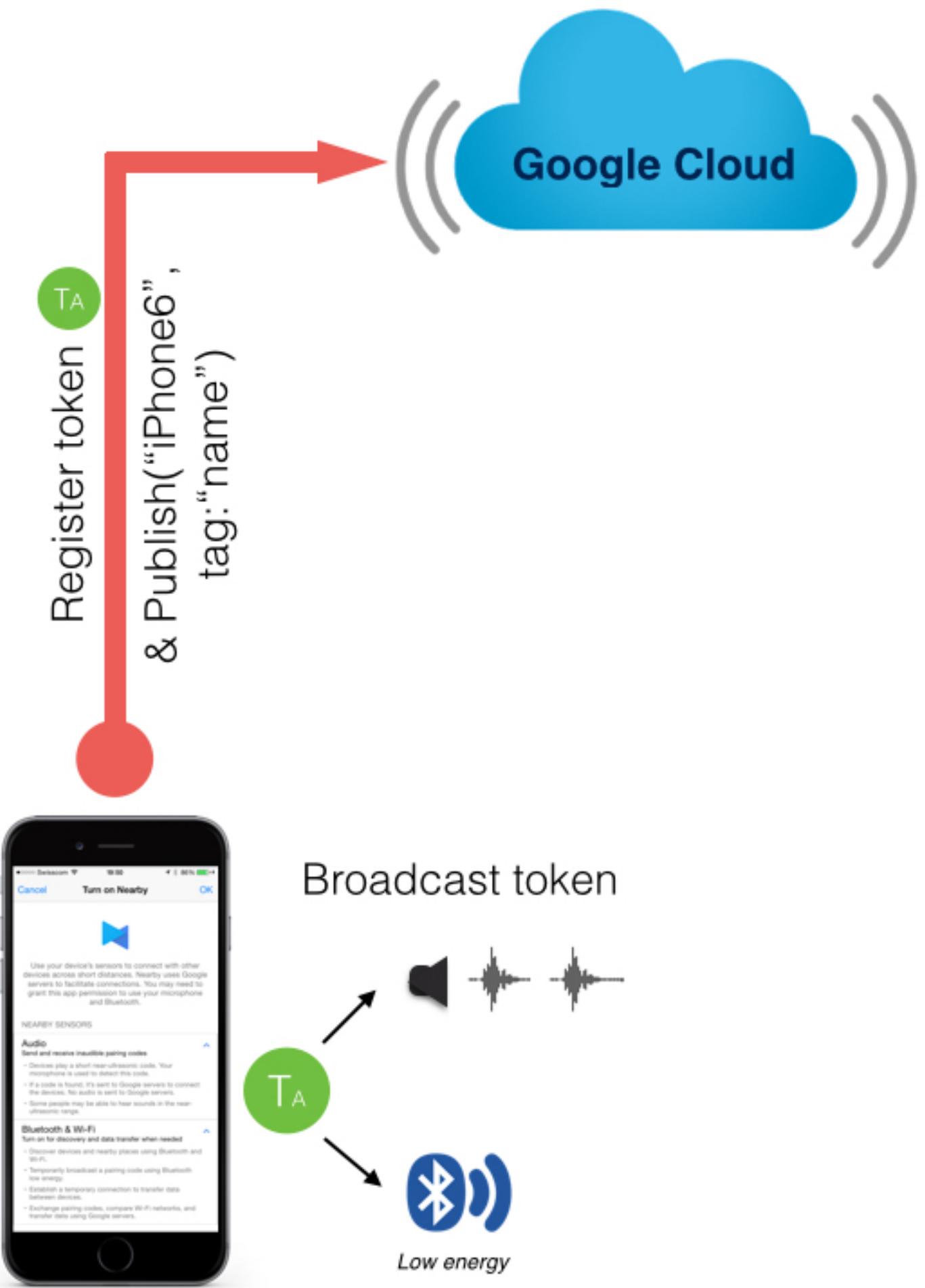


Image credit: p2pkit.io

Discovery & Messaging

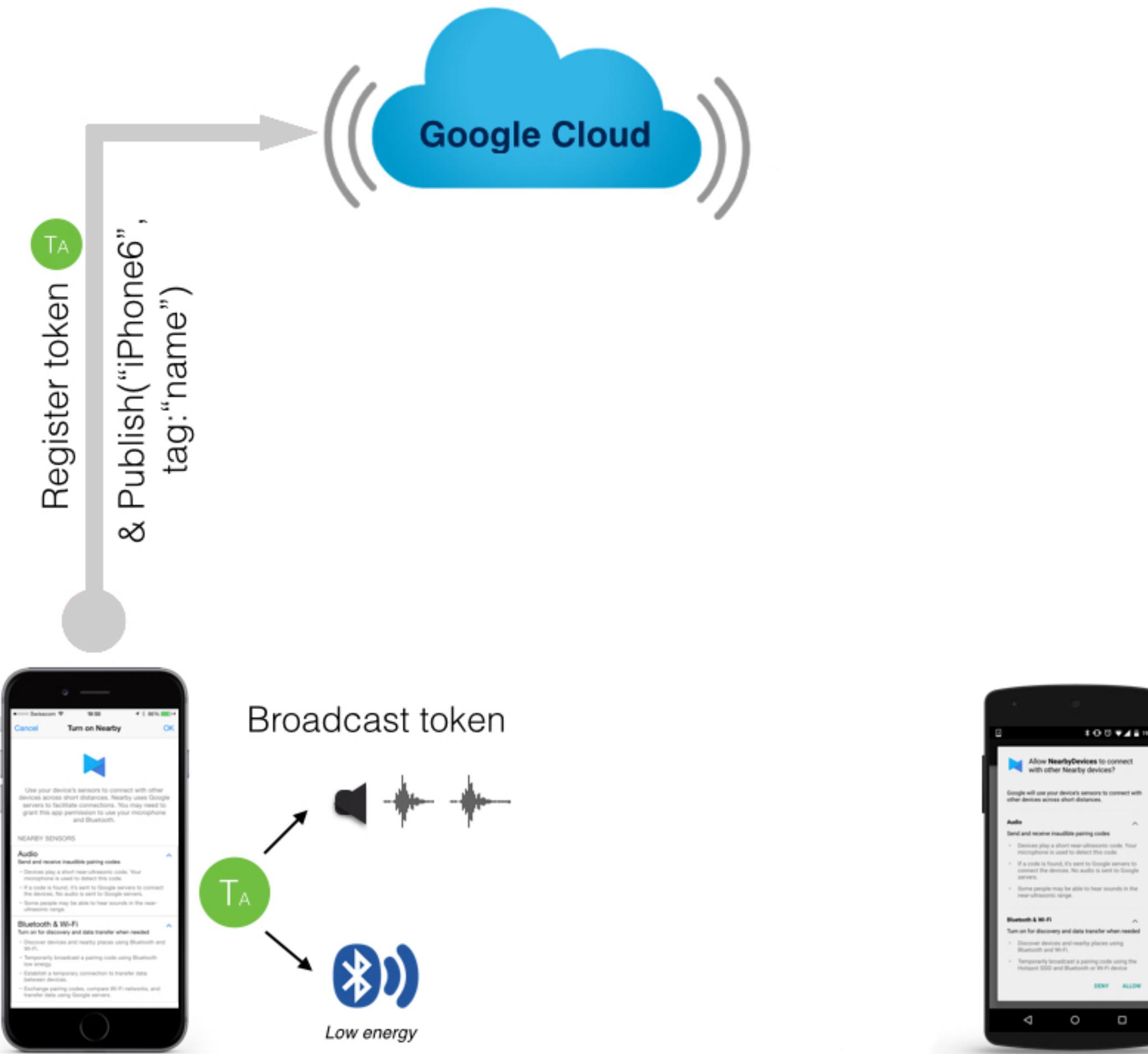


Image credit: p2pkit.io

Discovery & Messaging

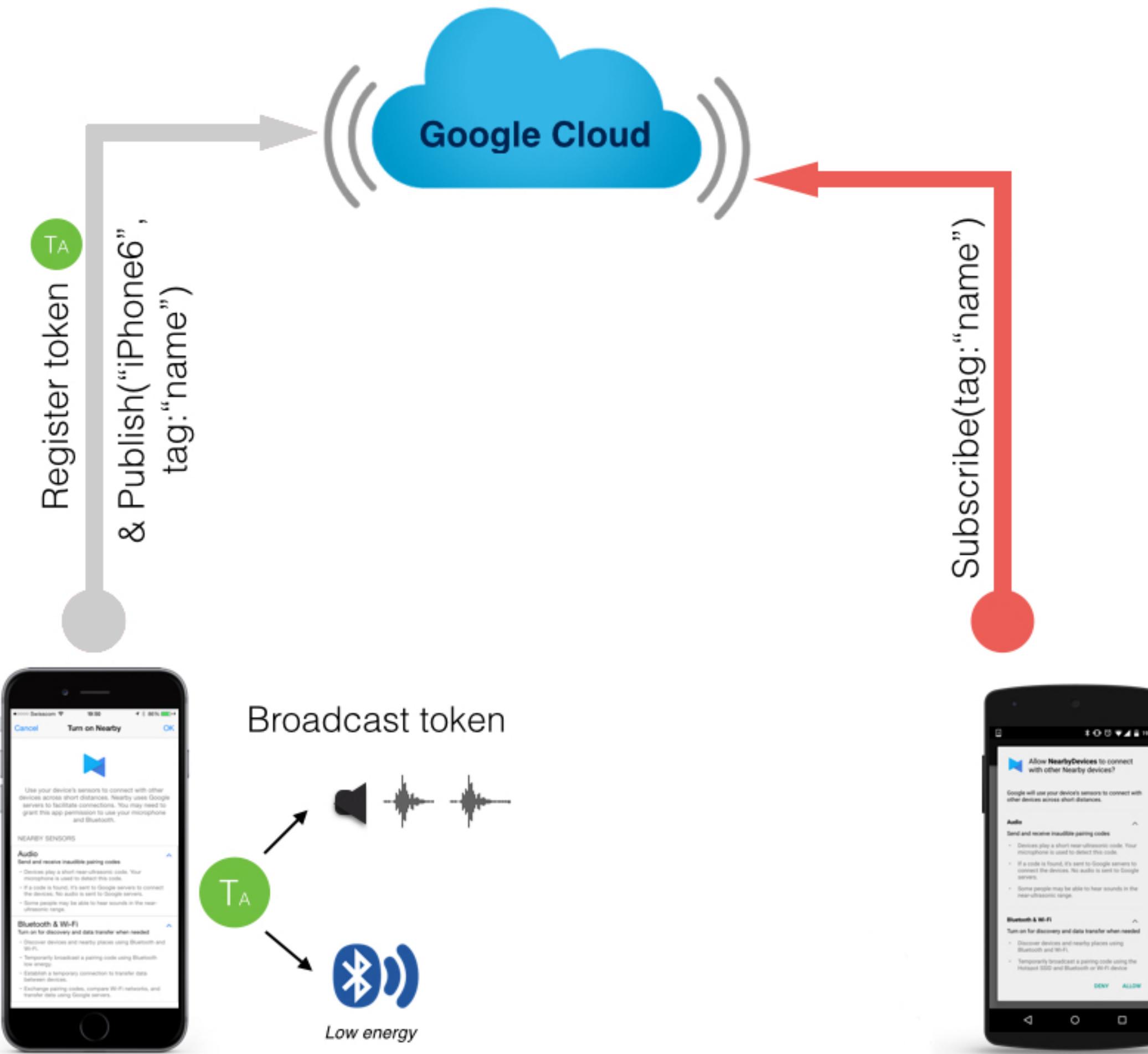


Image credit: p2pkit.io

Discovery & Messaging

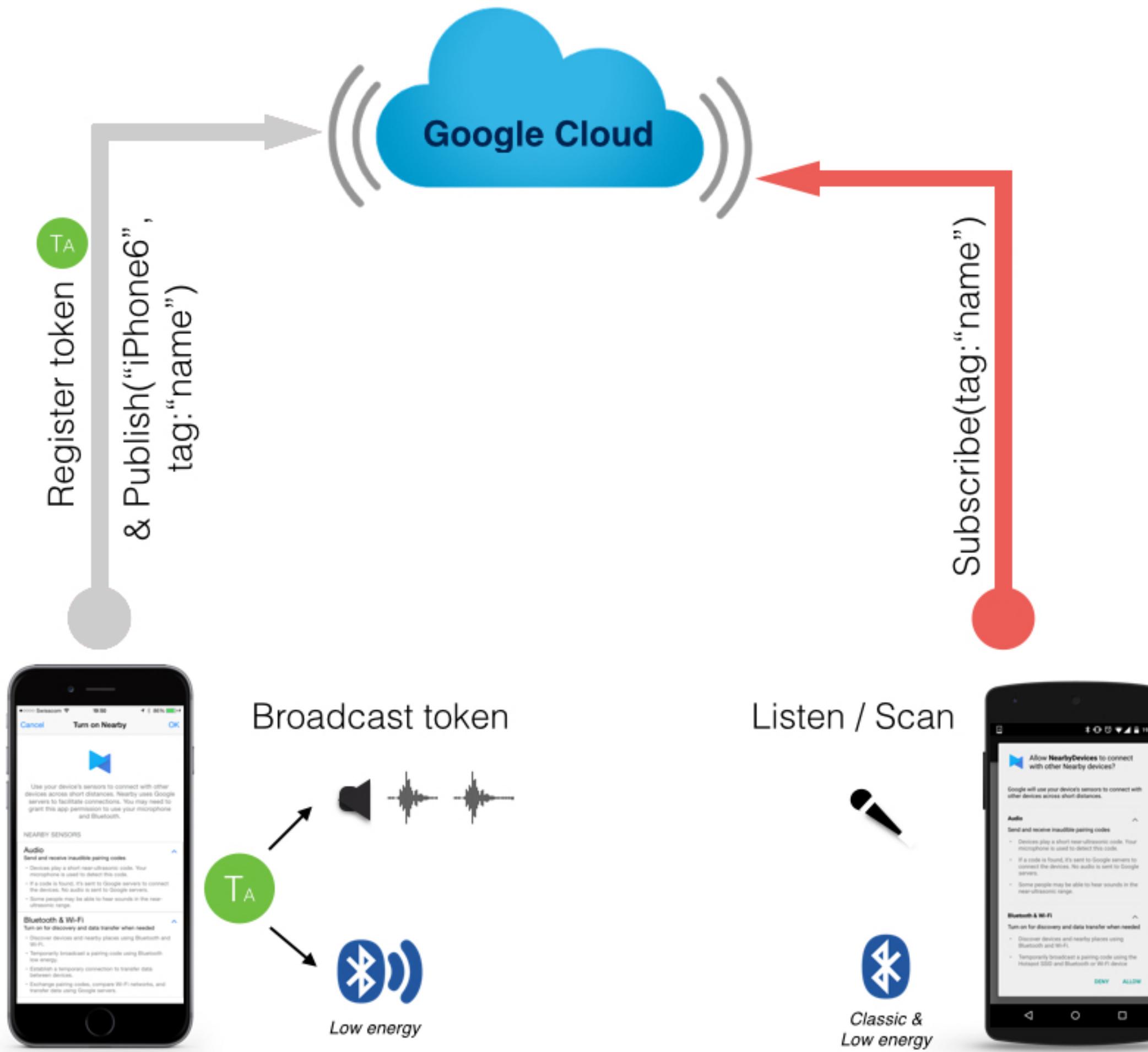


Image credit: p2pkit.io

Discovery & Messaging

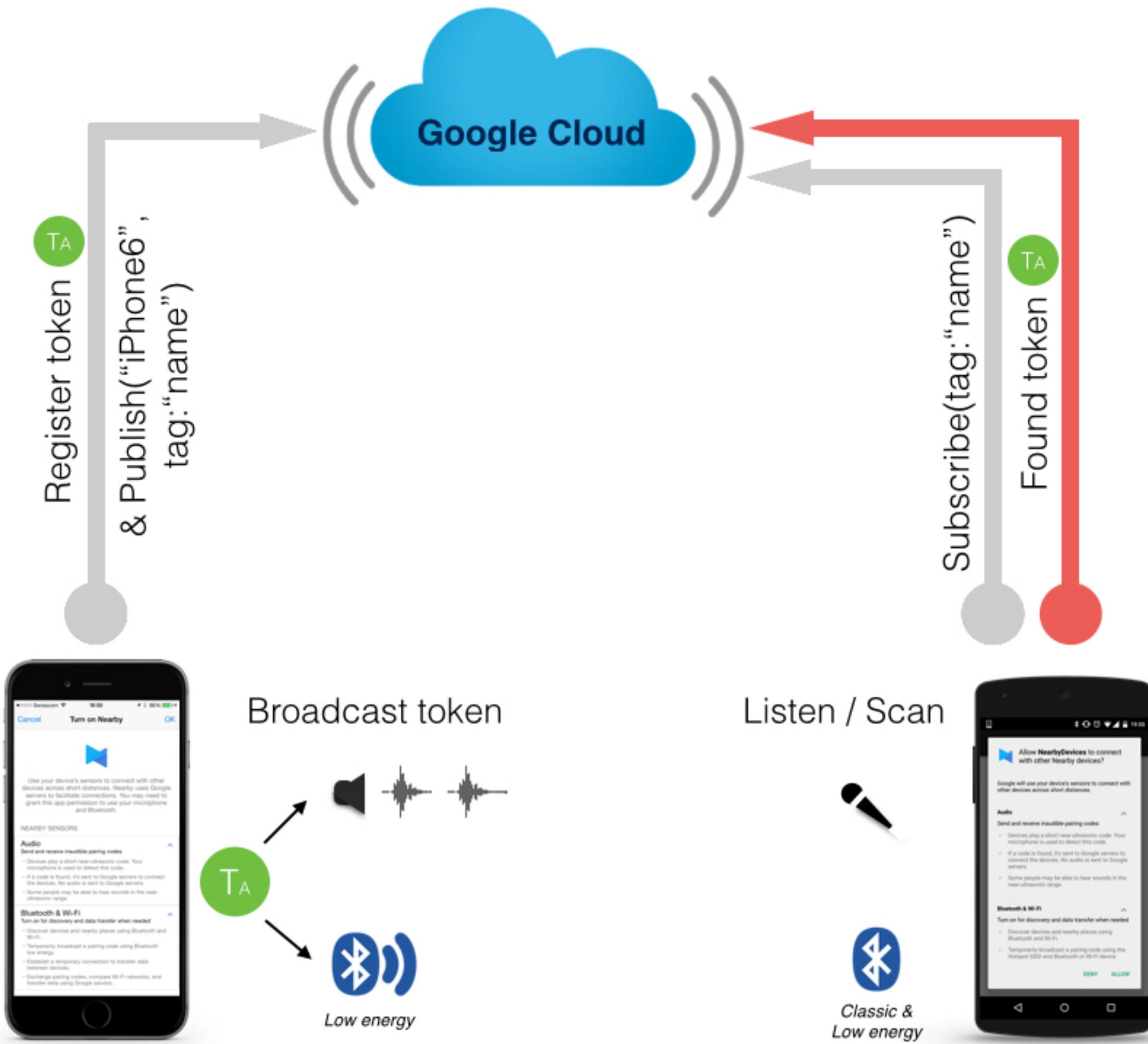


Image credit: p2pkit.io

Discovery & Messaging

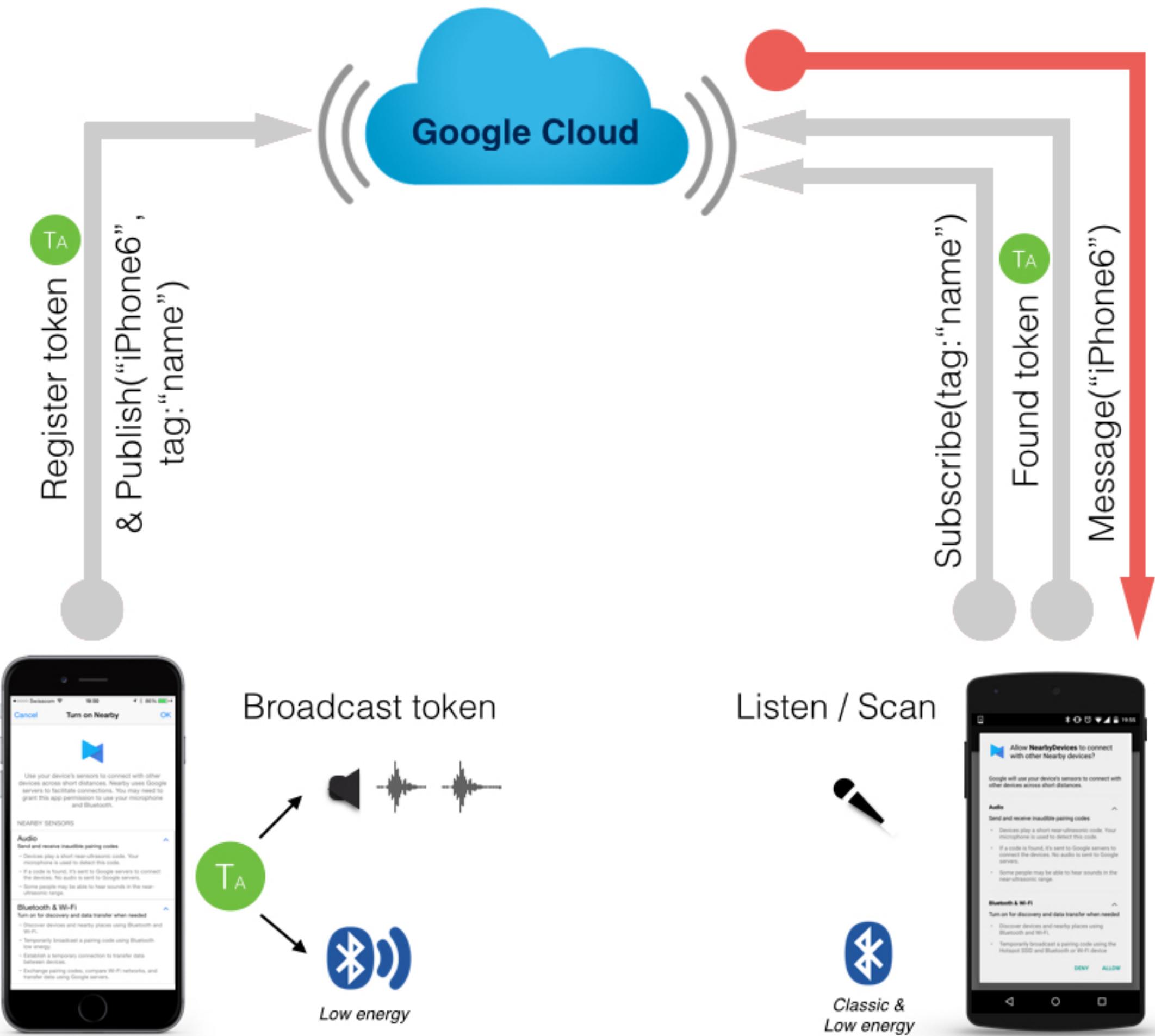


Image credit: p2pkit.io

Ultrasound

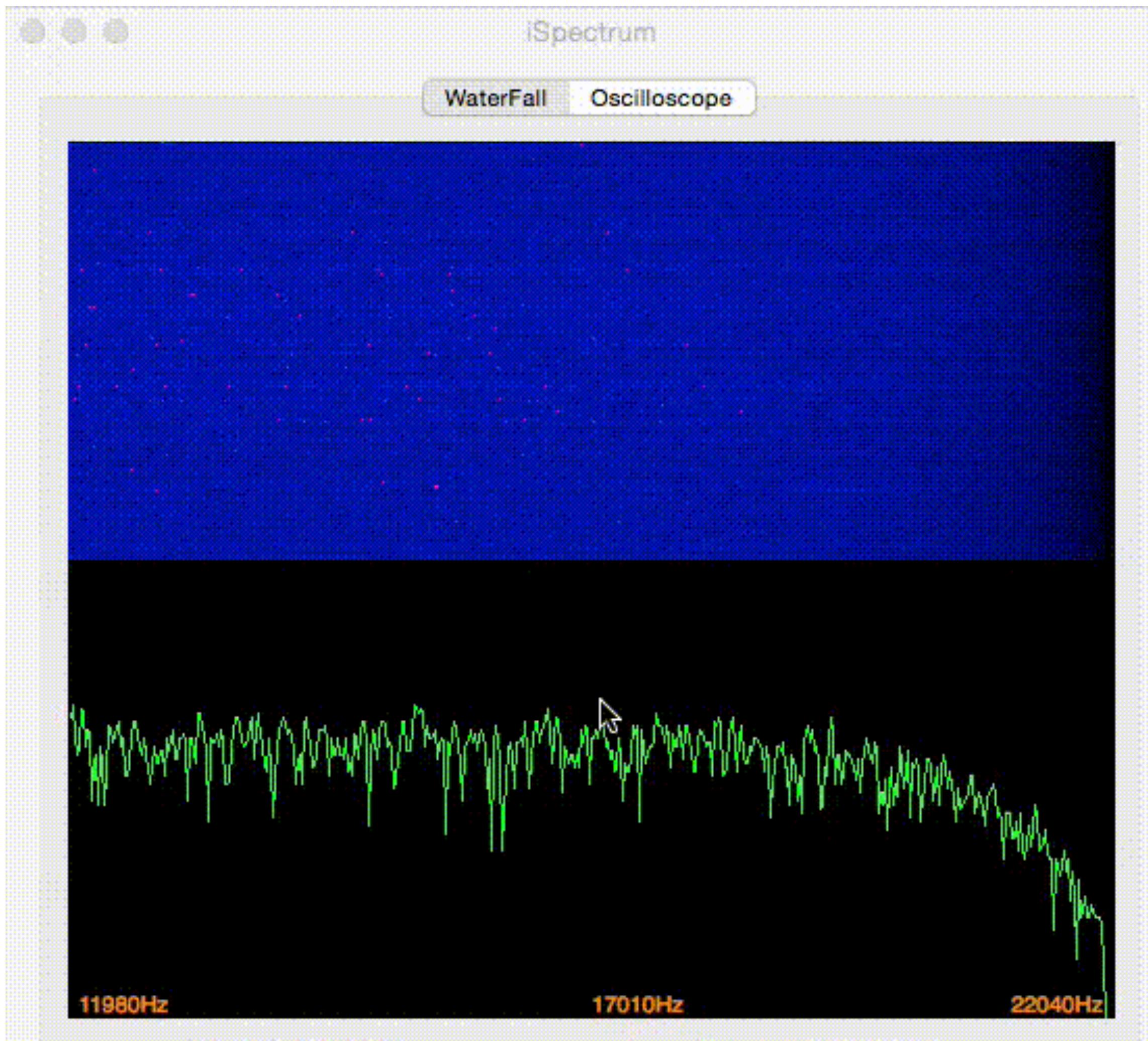


Image credit: p2pkit.io

Ultrasound

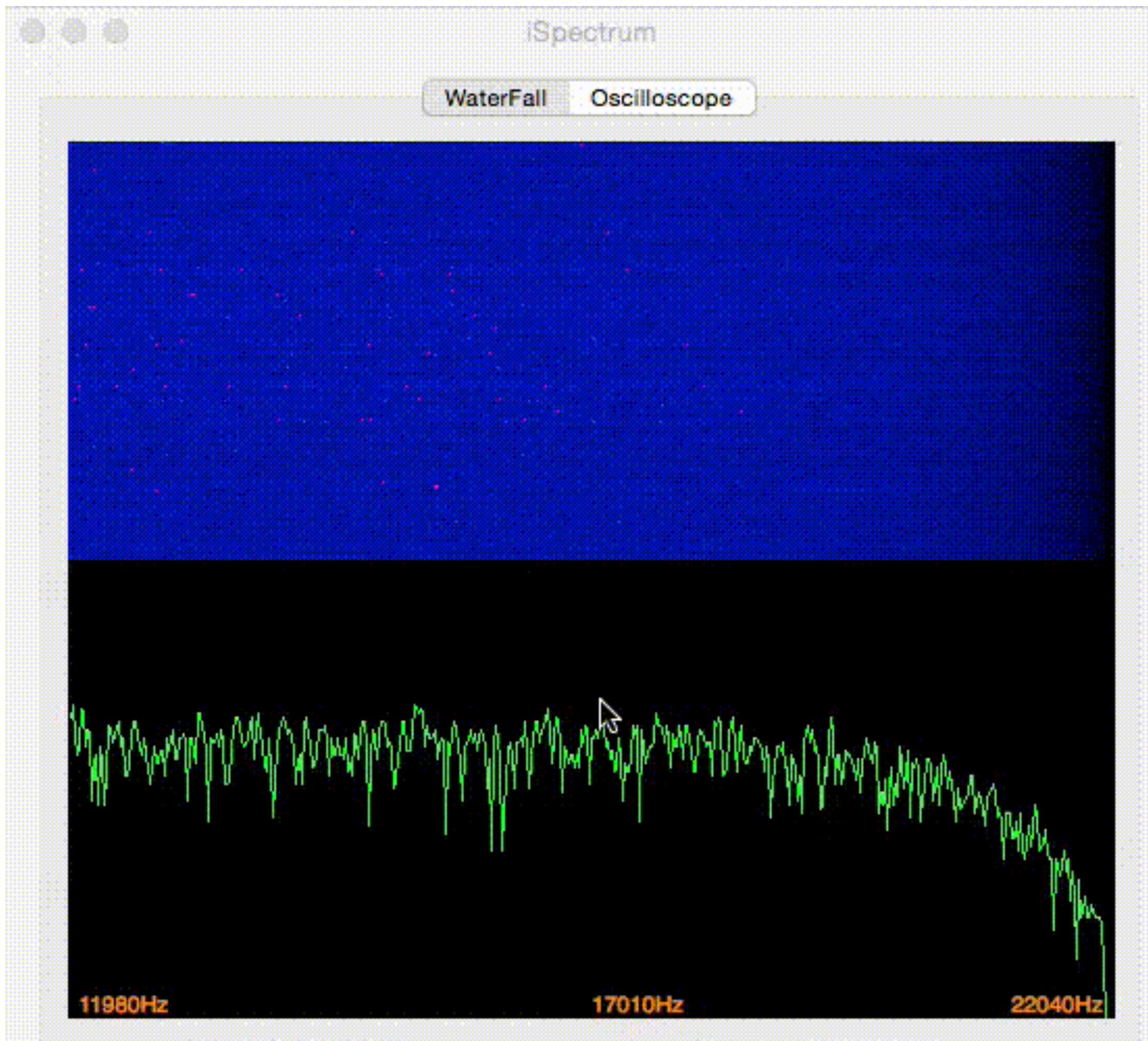


Image credit: p2pkit.io

Summary

- Tokens communicated using Bluetooth/speaker+mic

Summary

- Tokens communicated using Bluetooth/speaker+mic
 - ↳ range control

Summary

- Tokens communicated using Bluetooth/speaker+mic
 - ↳ range control
 - ↳ heavy battery usage

Summary

- Tokens communicated using Bluetooth/speaker+mic
 - ↳ range control
 - ↳ heavy battery usage
- Message content delivered via Google servers

Summary

- Tokens communicated using Bluetooth/speaker+mic
 - ↳ range control
 - ↳ heavy battery usage
- Message content delivered via Google servers
 - ↳ security/insecurity?

Summary

- Tokens communicated using Bluetooth/speaker+mic
 - ↳ range control
 - ↳ heavy battery usage
- Message content delivered via Google servers
 - ↳ security/insecurity?
 - ↳ connectivity requirement

Summary

- Tokens communicated using Bluetooth/speaker+mic
 - ↳ range control
 - ↳ heavy battery usage
- Message content delivered via Google servers
 - ↳ security/insecurity?
 - ↳ connectivity requirement
- Apps must share Google Developers Console project

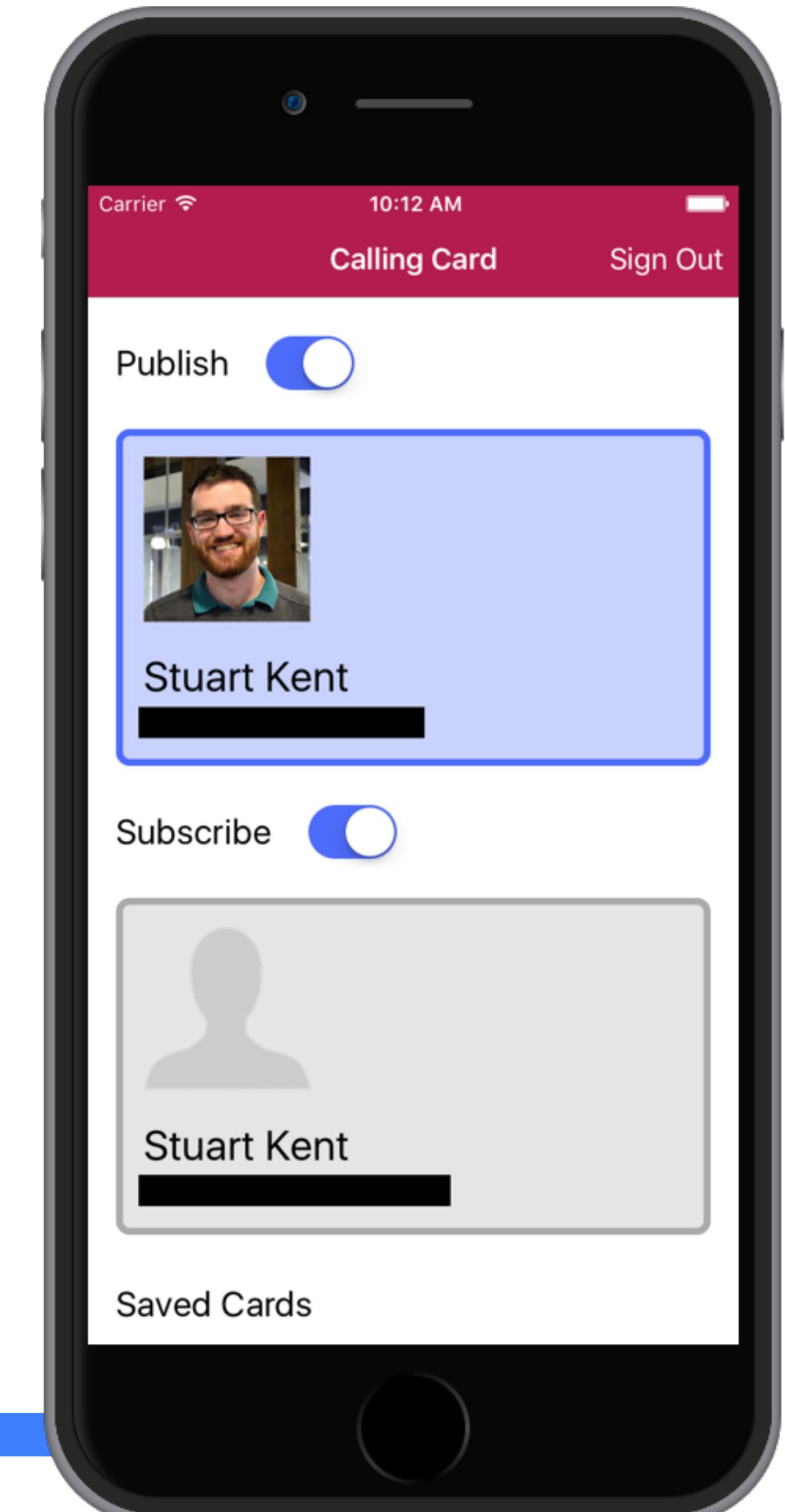
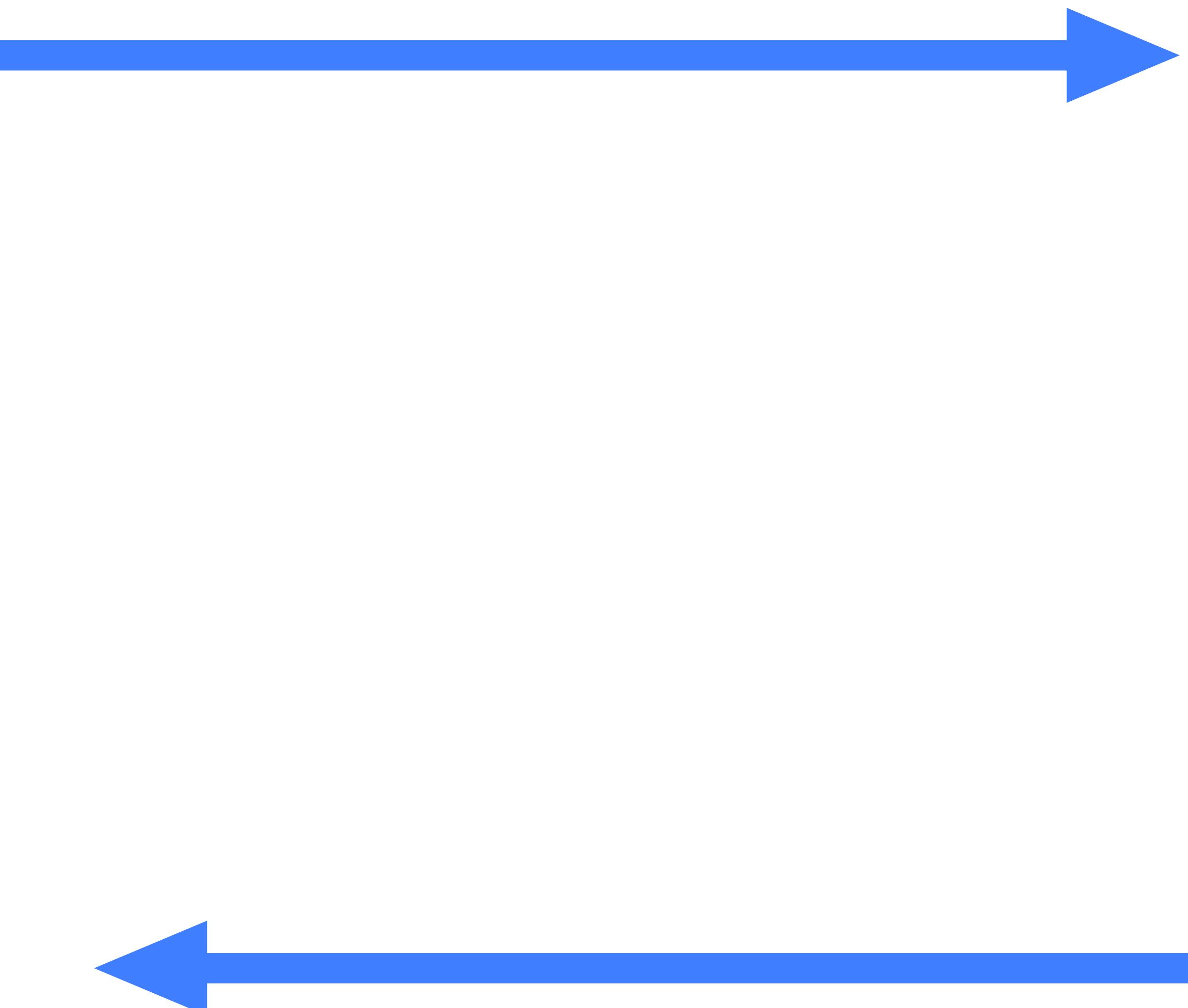
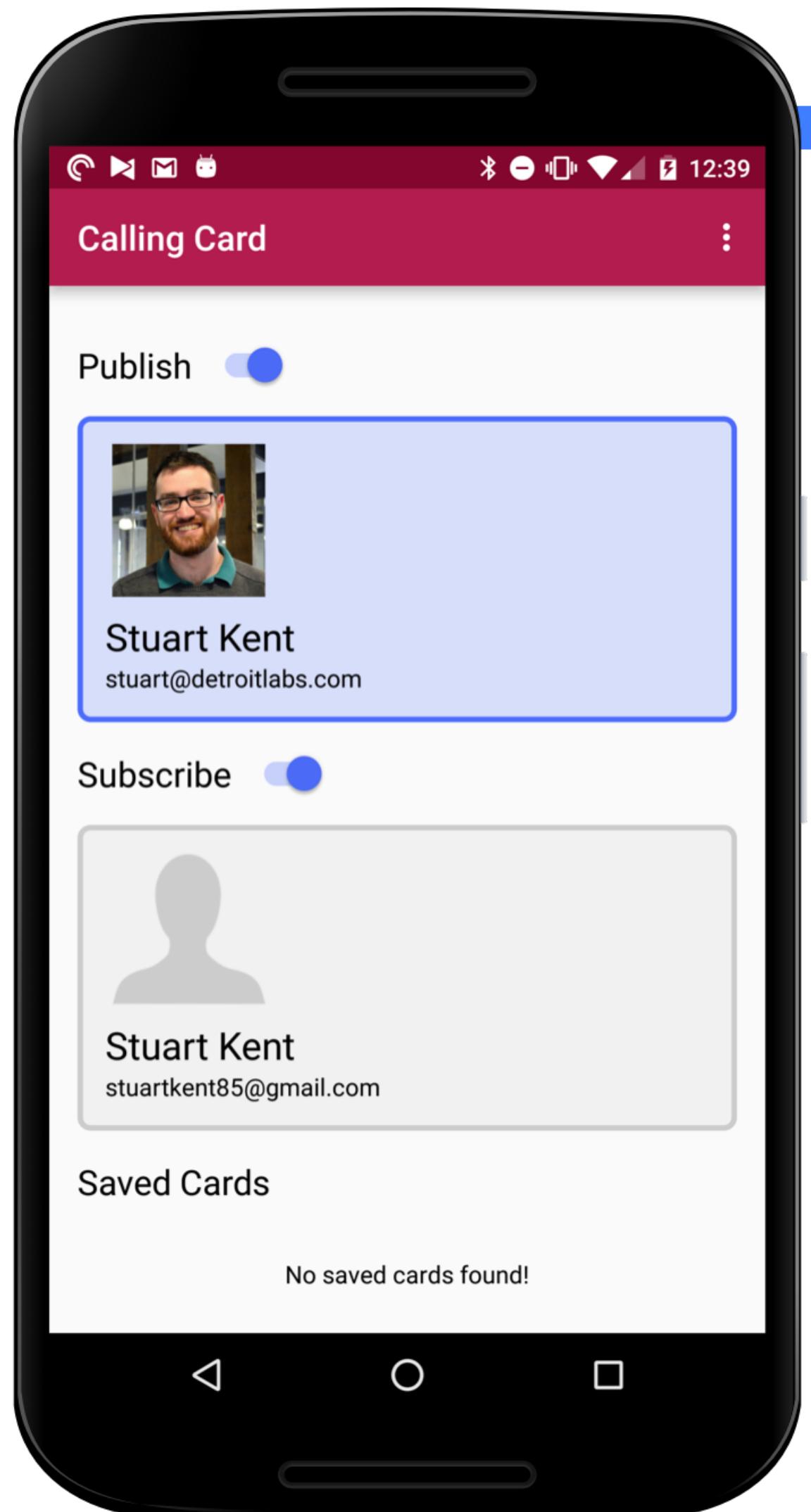
Summary

- Tokens communicated using Bluetooth/speaker+mic
 - ↳ range control
 - ↳ heavy battery usage
- Message content delivered via Google servers
 - ↳ security/insecurity?
 - ↳ connectivity requirement
- Apps must share Google Developers Console project
 - ↳ more security

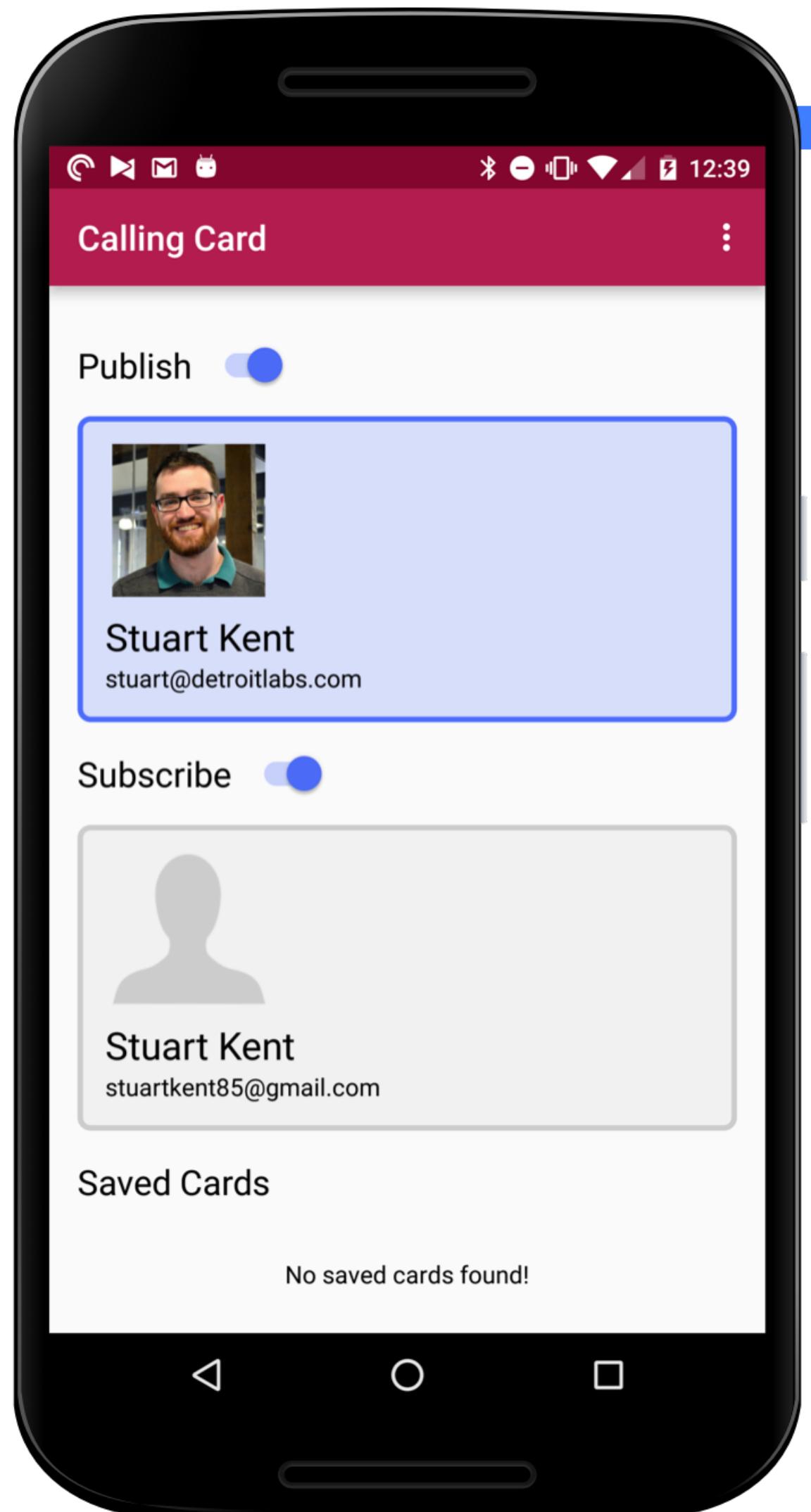
PRACTICALITIES

Demo & Code!

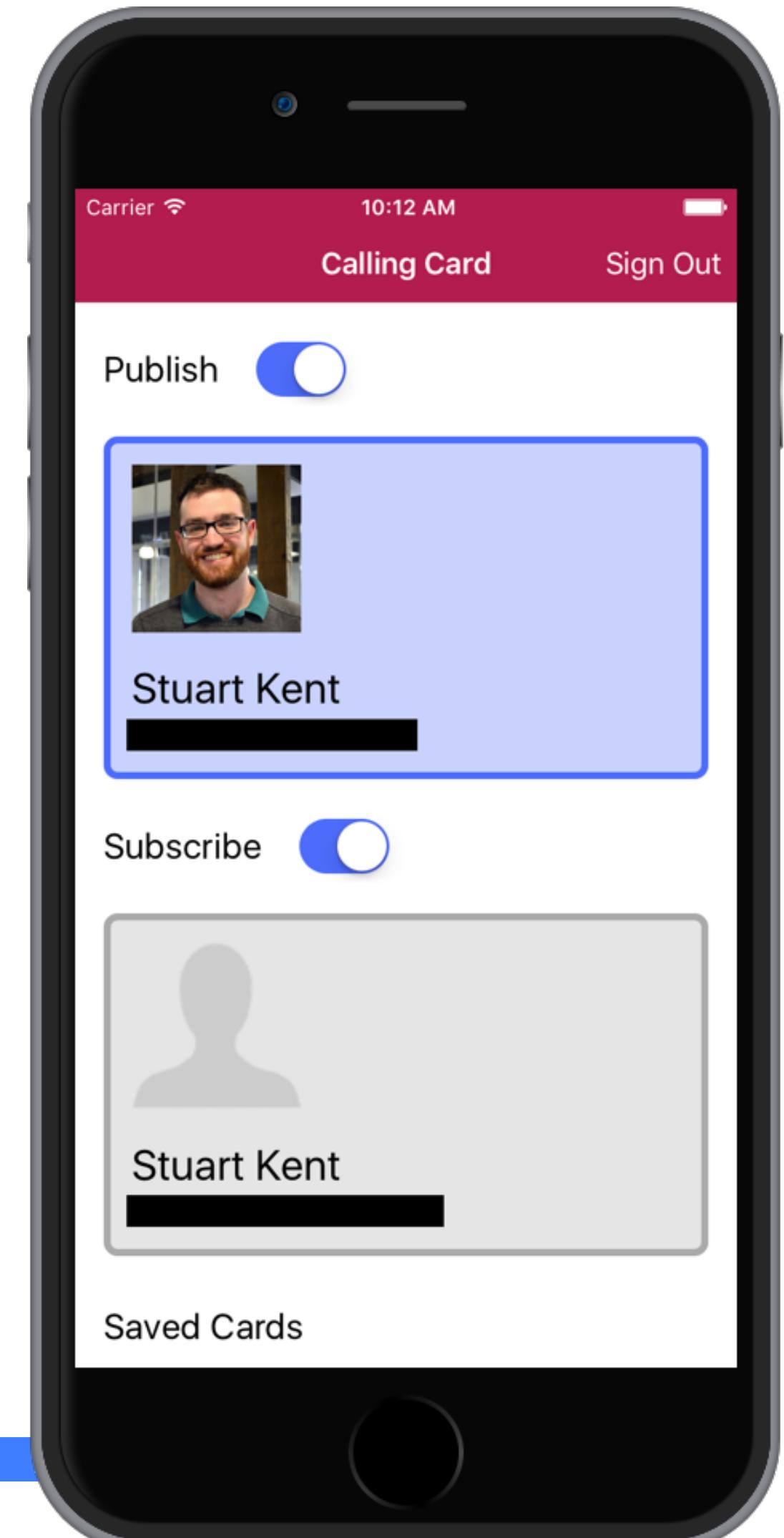
Calling Card



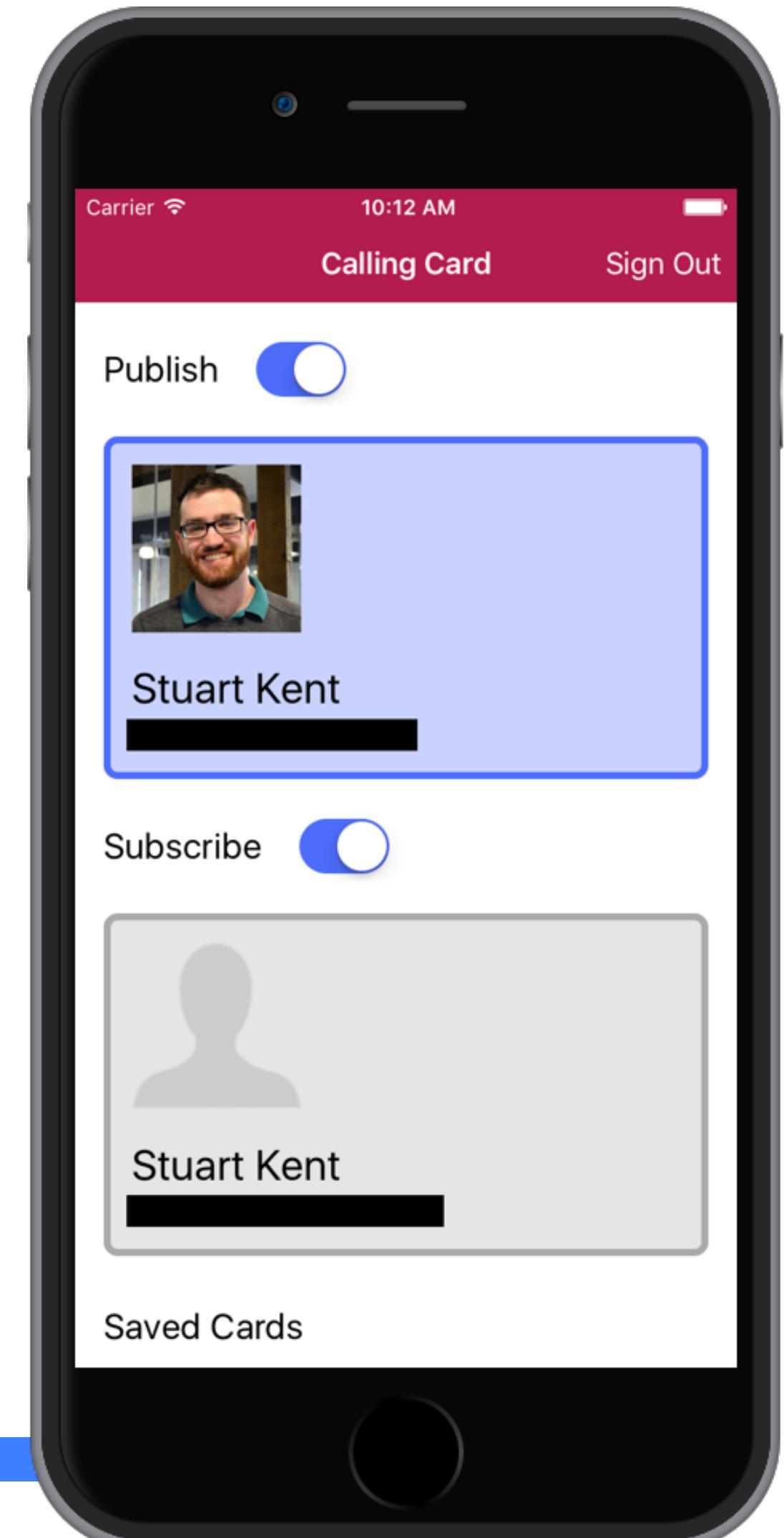
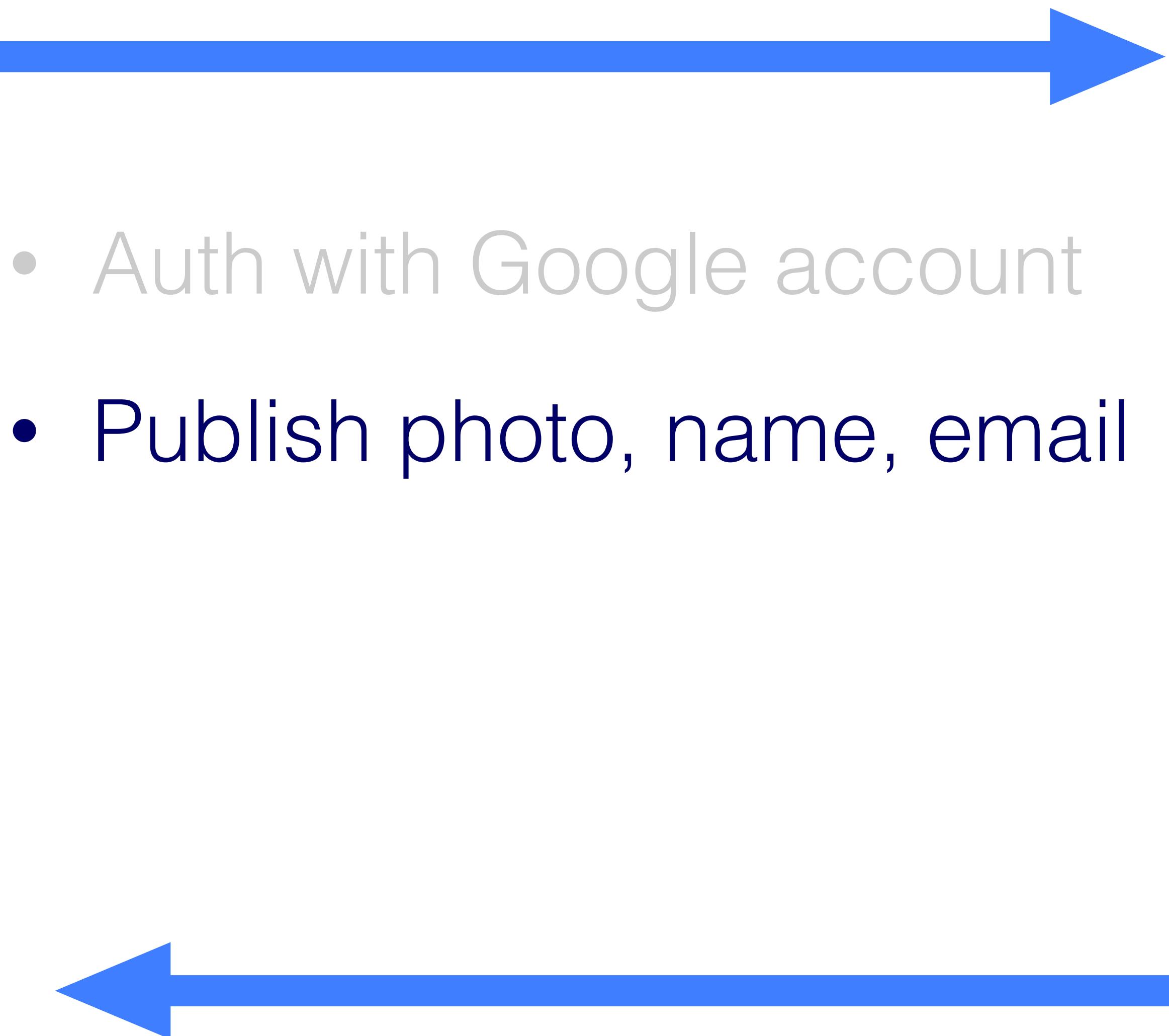
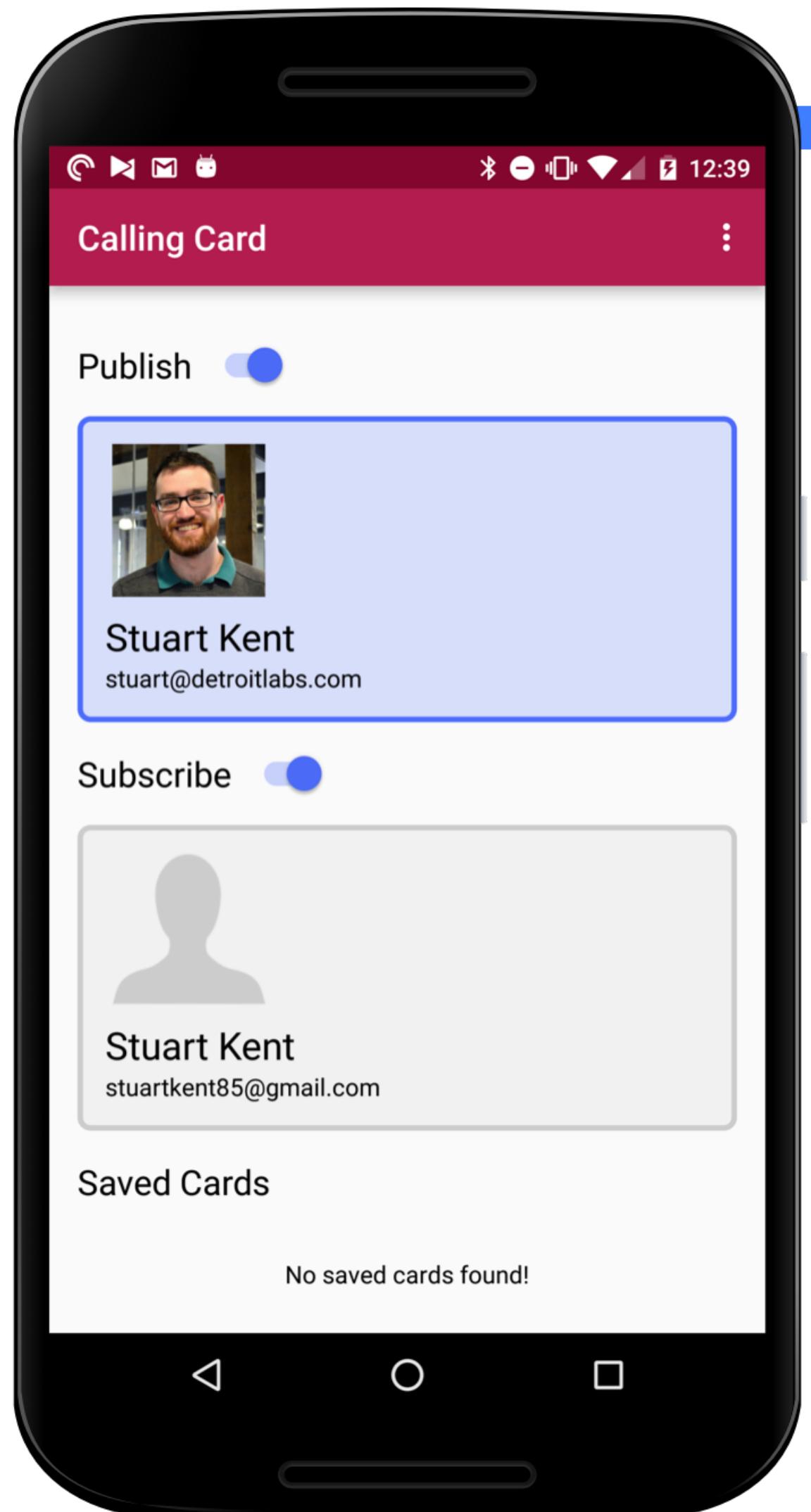
Calling Card



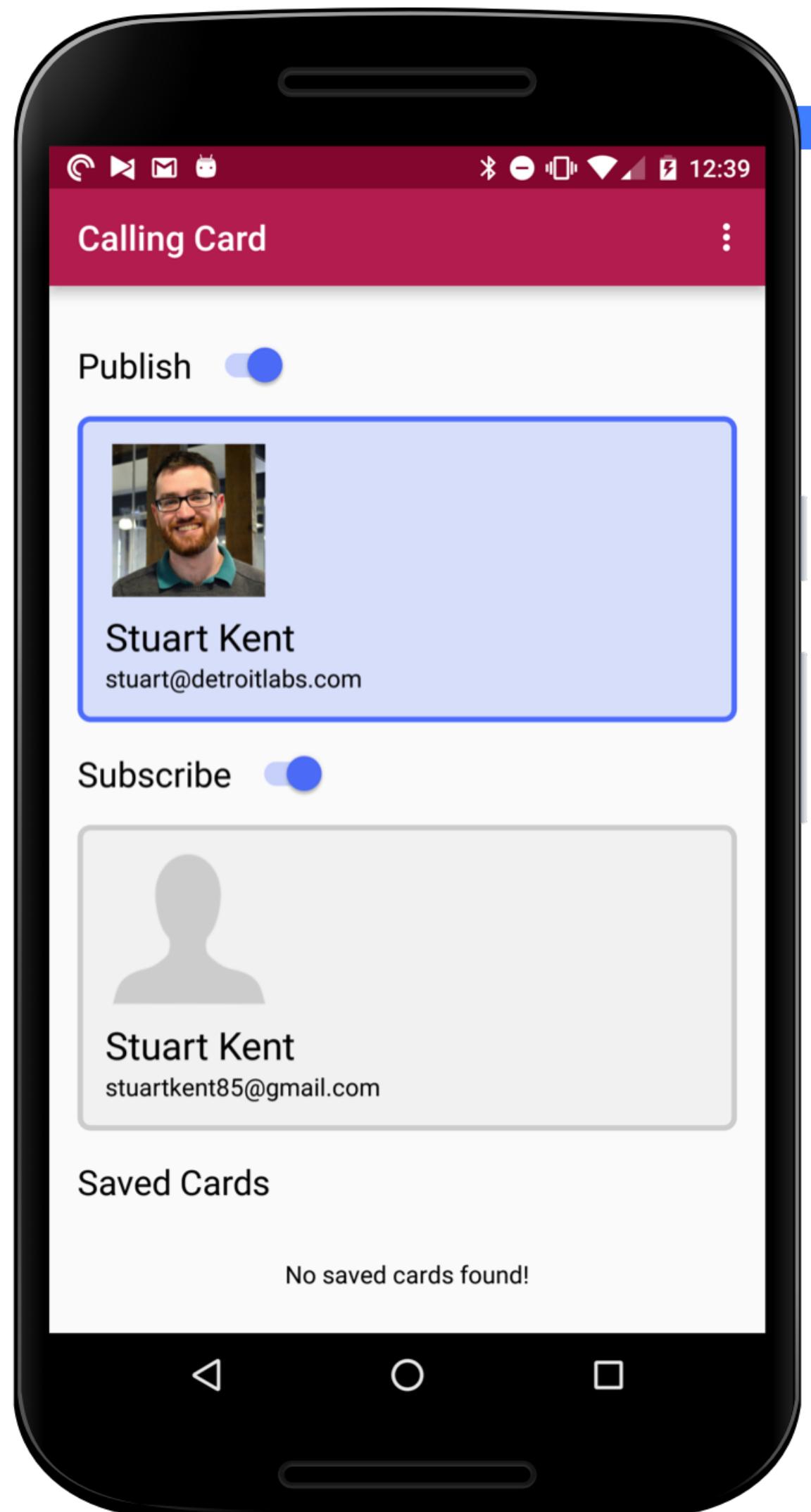
- Auth with Google account



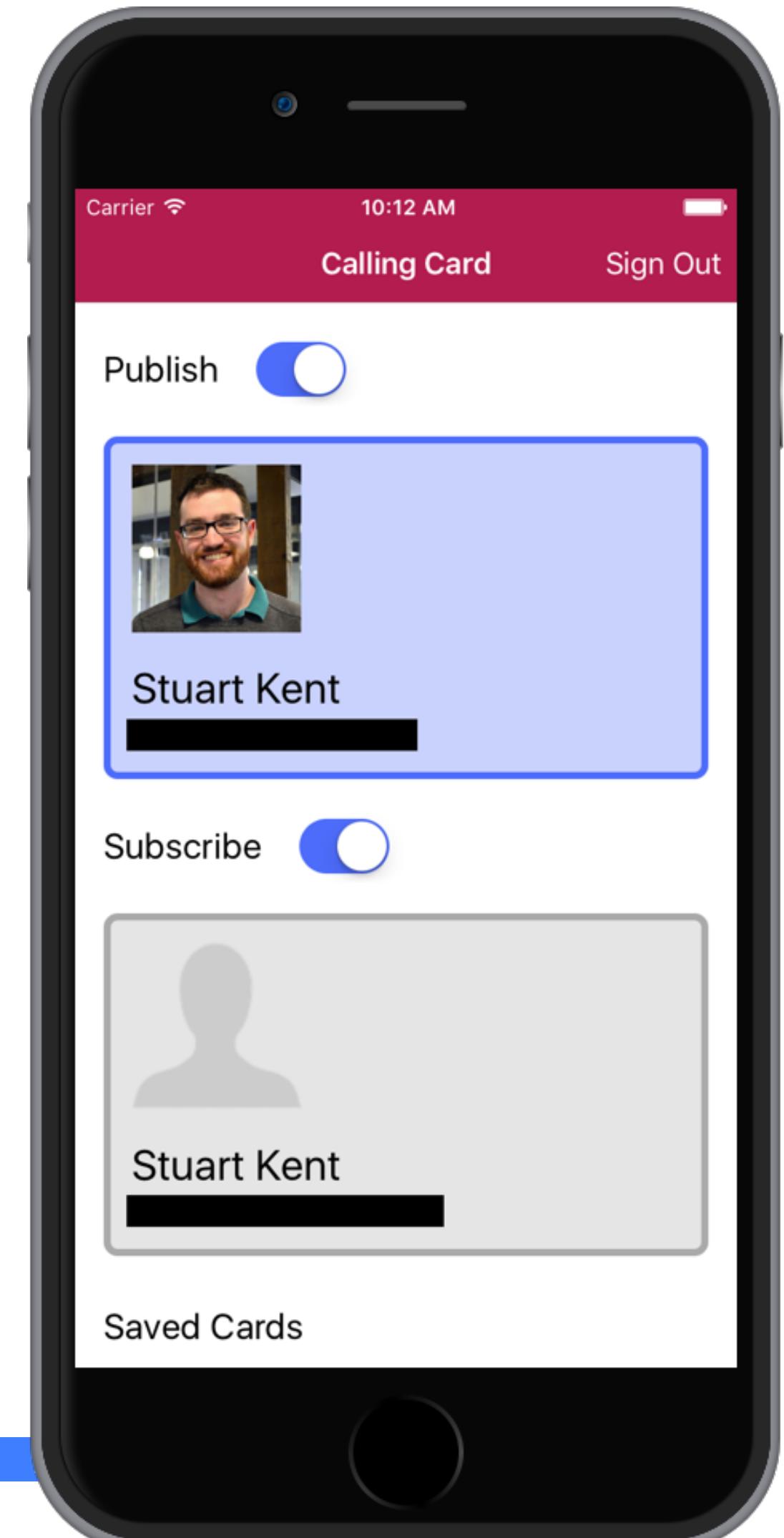
Calling Card



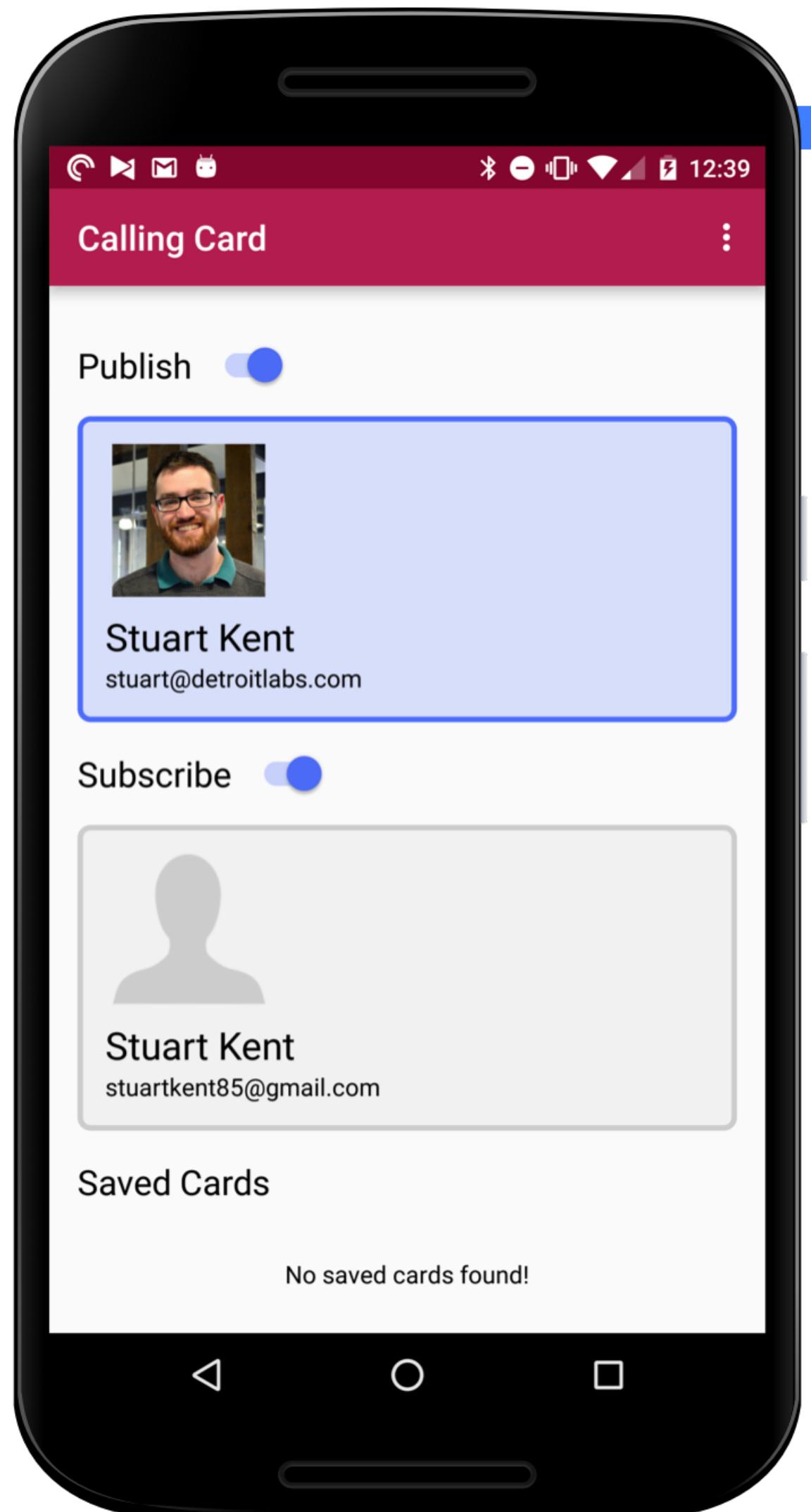
Calling Card



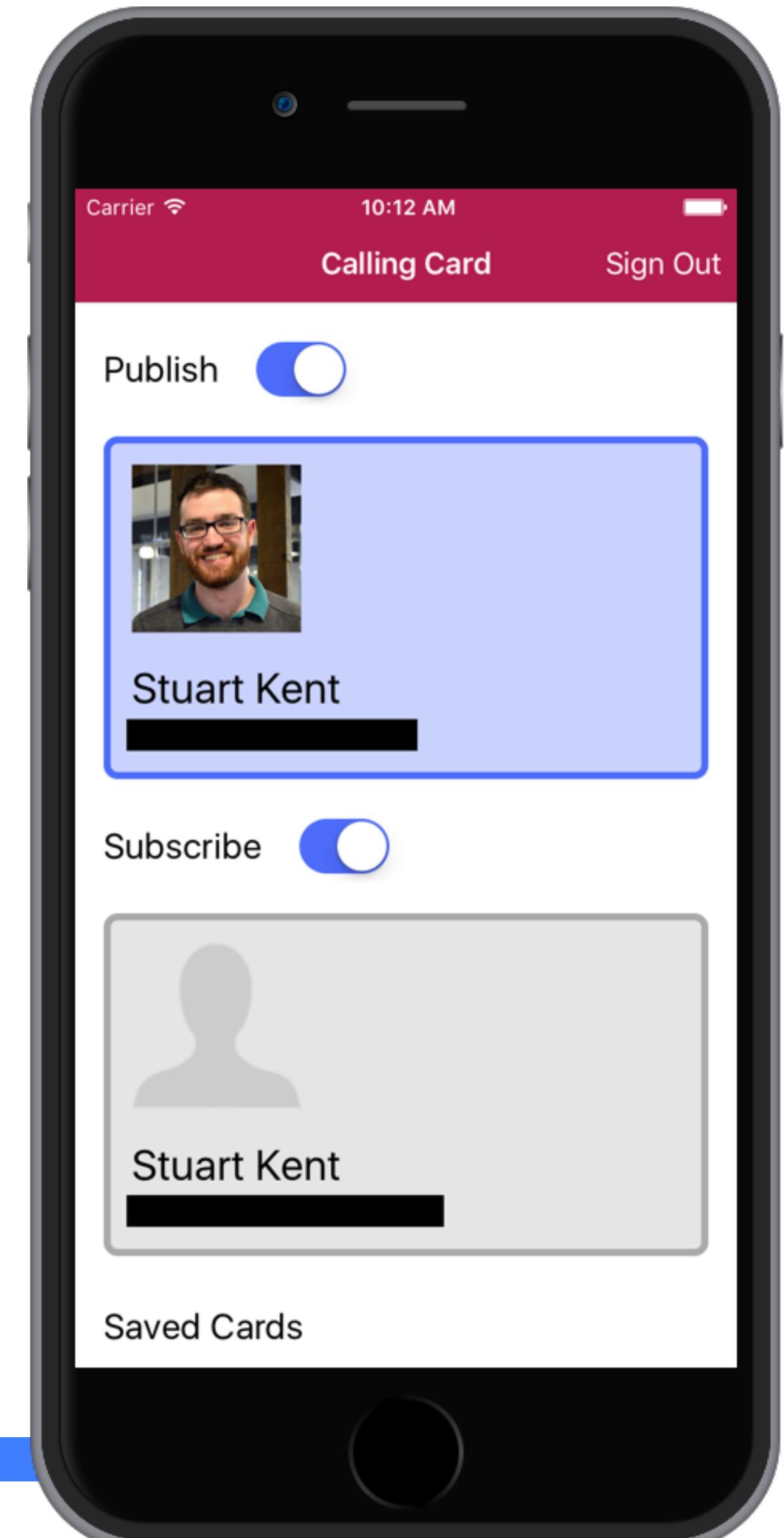
-
- Auth with Google account
 - Publish photo, name, email
 - See nearby users (100ft)
- ←



Calling Card



- Auth with Google account
- Publish photo, name, email
- See nearby users (100ft)
- Save contact info



Calling Card

Android

Play Store
goo.gl/dkmGxa

GitHub
[stkent/calling-card](https://github.com/stkent/calling-card)

iOS



GitHub
[stkent/calling-card-ios](https://github.com/stkent/calling-card-ios)

Implementation Notes

Apps must handle:

- runtime permission (Bluetooth/Audio)

Implementation Notes

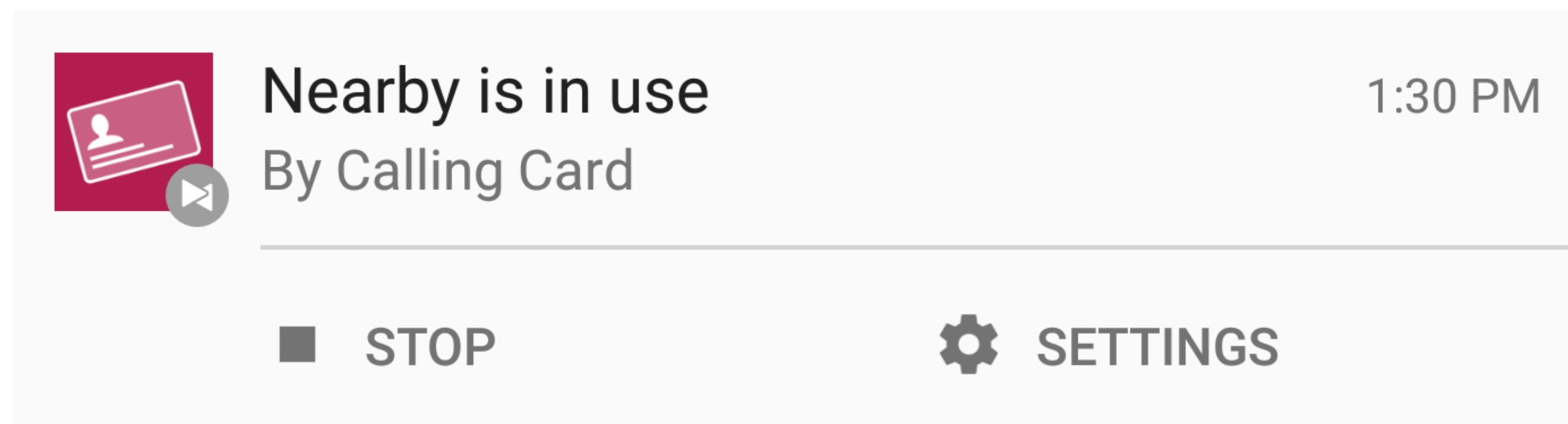
Apps must handle:

- runtime permission (Bluetooth/Audio)
- message serialization & deserialization

Implementation Notes

Apps must handle:

- runtime permission (Bluetooth/Audio)
- message serialization & deserialization
- notification kill (Android)



Implementation Notes

Apps must handle:

- runtime permission (Bluetooth/Audio)
- message serialization & deserialization
- notification kill (Android)
- teardown

Android



build.gradle

```
dependencies {  
    ...  
    compile 'com.google.android.gms:play-services-nearby:9.0.0'  
    ...  
}
```



AndroidManifest.xml

```
<application>  
    ...  
    <!-- API key from Google API Console -->  
    <meta-data  
        android:name="com.google.android.nearby.messages.API_KEY"  
        android:value="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" />  
    ...  
</application>
```



NearbyActivity.java

```
Nearby.Messages.publish(googleApiClient, message, pubOptions)
```



NearbyActivity.java

```
Nearby.Messages.publish(googleApiClient, message, pubOptions)  
//////////  
googleApiClient = new GoogleApiClient.Builder(this)  
    .addApi(NearbyMESSAGES_API)  
    .addConnectionCallbacks(this)  
    .addOnConnectionFailedListener(this)  
    .build();
```



NearbyActivity.java

```
Nearby.Messages.publish(googleApiClient, message, pubOptions)  
//////////  
  
@Override  
protected void onStart() {  
    super.onStart();  
    googleApiClient.connect();  
}
```



NearbyActivity.java

```
Nearby.Messages.publish(googleApiClient, message, pubOptions)

//////////////////////////////
```

```
@Override
public void onConnectionFailed(ConnectionString result) {
    if (result.hasResolution()) {
        try {
            result.startResolutionForResult(this, NEARBY_CODE);
        } catch (final IntentSender.SendIntentException e) {
            // cancel all Nearby operations
        }
    } else { // cancel all Nearby operations }
}
```



NearbyActivity.java

```
Nearby.Messages.publish(googleApiClient, message, pubOptions)

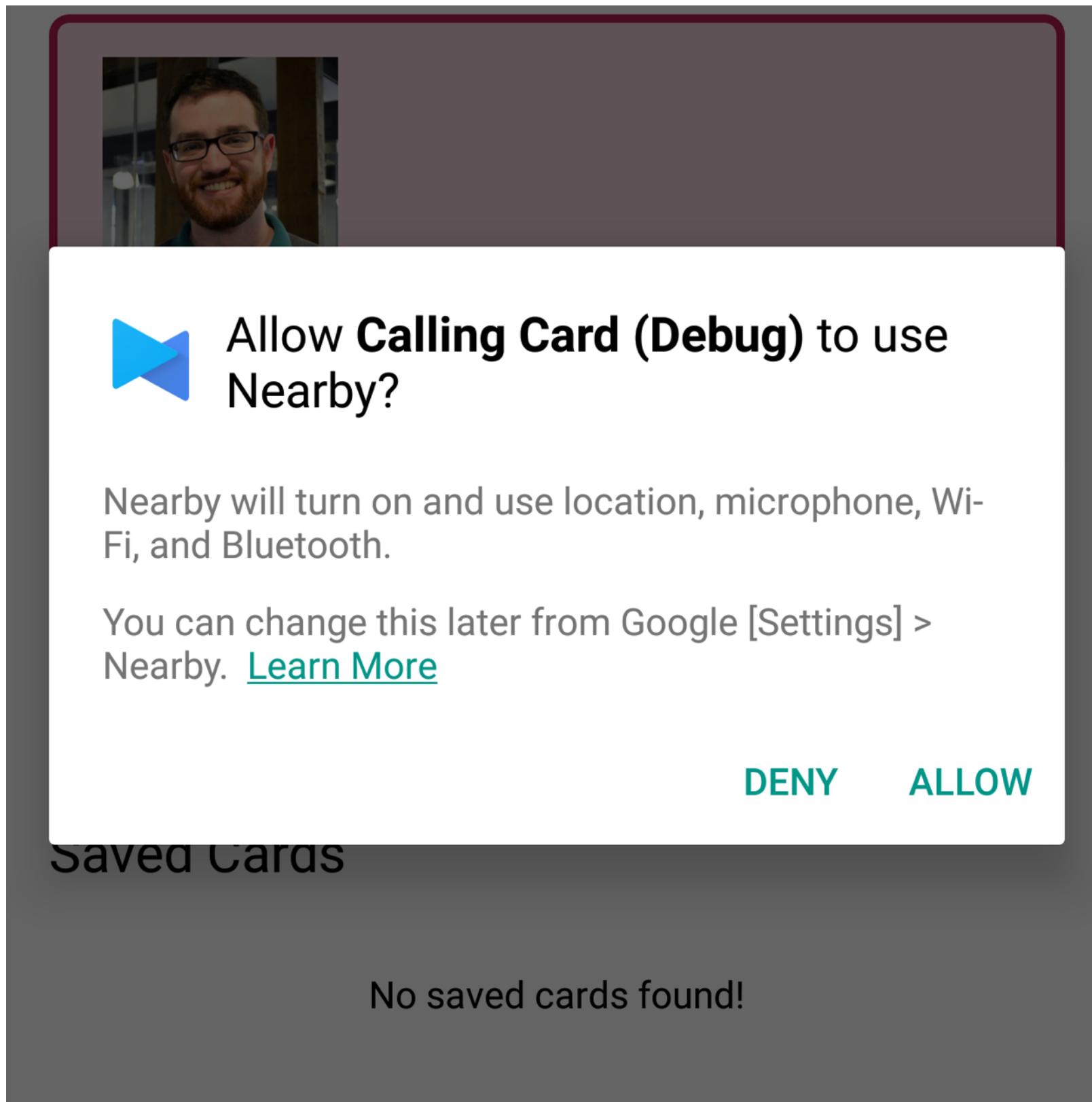
//////////////////////////////
```

```
@Override
public void onConnectionFailed(ConnectionString result) {
    if (result.hasResolution()) {
        try {
            result.startResolutionForResult(this, NEARBY_CODE);
        } catch (final IntentSender.SendIntentException e) {
            // cancel all Nearby operations
        }
    } else { // cancel all Nearby operations }
}
```



NearbyActivity.java

```
Nearby.Messages.publish(googleApiClient, message, pubOptions)
```



NearbyActivity.java

```
Nearby.Messages.publish(googleApiClient, message, pubOptions)

//////////////////////////////
```

```
@Override
protected void onActivityResult(
    int requestCode, int resultCode, Intent data) {

    if (requestCode == NEARBY_CODE) {
        if (resultCode == RESULT_OK) { googleApiClient.connect(); }
        else { // cancel all Nearby operations }
    } else {
        super.onActivityResult(requestCode, resultCode, data);
    }
}
```



NearbyActivity.java

```
Nearby.Messages.publish(googleApiClient, message, pubOptions)

//////////////////////////////
```

```
@Override
protected void onActivityResult(
    int requestCode, int resultCode, Intent data) {

    if (requestCode == NEARBY_CODE) {
        if (resultCode == RESULT_OK) { googleApiClient.connect(); }
        else { // cancel all Nearby operations }
    } else {
        super.onActivityResult(requestCode, resultCode, data);
    }
}
```



NearbyActivity.java

```
Nearby.Messages.publish(googleApiClient, message, pubOptions)  
//////////  
private Message message;  
...  
User user = getIntent().getParcelableExtra(USER_EXTRA_KEY);  
message = new Message(GSON.toJson(user).getBytes());
```



NearbyActivity.java



```
Nearby.Messages.publish(googleApiClient, message, pubOptions)
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
// can also be used to set role and range
private PublishOptions pubOptions = new PublishOptions.Builder()
    .setCallback(publishCallback)
    .build();
```



NearbyActivity.java



```
Nearby.Messages.publish(googleApiClient, message, pubOptions)  
//////////  
  
// can also be used to set role and range  
private PublishOptions pubOptions = new PublishOptions.Builder()  
.setCallback(publishCallback)  
.build();  
  
...  
  
private PublishCallback publishCallback = new PublishCallback() {  
    @Override  
    public void onExpired() { // cancel all Nearby operations }  
};
```



NearbyActivity.java

```
private Message message;  
...  
Nearby.Messages.unpublish(googleApiClient, message);  
// show inactive UI
```



NearbyActivity.java

```
private Message message;
```

```
...
```

```
Nearby.Messages.unpublish(googleApiClient, message);  
// show inactive UI
```



NearbyActivity.java

```
Nearby.Messages.subscribe(googleApiClient, listener, subOptions)
```



NearbyActivity.java

```
Nearby.Messages.subscribe(googleApiClient, listener, subOptions)
```

```
//////////
```



NearbyActivity.java

```
Nearby.Messages.subscribe(googleApiClient, listener, subOptions)
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
private MessageListener listener = new MessageListener() {  
    @Override  
    public void onFound(Message message) {  
        // received once; deserialize data and cache it  
    }  
  
    @Override  
    public void onLost(Message message) {  
        // received once; deserialize data and remove from cache  
    }  
};
```



NearbyActivity.java



Nearby.Messages.subscribe(googleApiClient, listener, subOptions)

Same API as publish options.



NearbyActivity.java

```
private MessageListener listener;  
...  
Nearby.Messages.unsubscribe(googleApiClient, listener);  
// show inactive UI
```



NearbyActivity.java

```
private MessageListener listener;  
...  
Nearby.Messages.unsubscribe(googleApiClient, listener);  
// show inactive UI
```



NearbyActivity.java

```
@Override  
protected void onStop() {  
    // cancel all Nearby operations  
    // disconnect GoogleApiClient  
    super.onStop();  
}
```



iOS



Podfile (0.39)

```
source 'https://github.com/CocoaPods/Specs.git'  
platform :ios, '9.0'  
pod 'NearbyMessages', '0.10.0'
```



Nearby-Bridging-Header.h

```
#import <GNSMessages.h>
```



NearbyViewController.swift

```
private lazy var messageManager: GNSMessageManager = {  
    GNSMessageManager(APIKey: "xxxxxxxxxxxxxxxxxxxxxxxxxxxx")  
}()
```



NearbyViewController.swift

```
private var message: GNSMessage? {
    guard let userData = user?.toNSData() else {
        return nil
    }

    return GNSMessage(content: userData)
}
```



NearbyViewController.swift

```
private var message: GNSMessage? {  
    guard let userData = user?.toNSData() else {  
        return nil  
    }  
  
    return GNSMessage(content: userData)  
}
```



NearbyViewController.swift

```
private var permissionProxy: GNSPermission?  
...  
override func viewDidAppear(animated: Bool) {  
    super.viewDidAppear(animated)  
  
    permissionProxy = GNSPermission { [weak self] granted in  
        if granted {  
            // show active UI  
        } else {  
            // show inactive UI  
        }  
    }  
}
```



NearbyViewController.swift

```
private var permissionProxy: GNSPermission?  
...  
override func viewDidAppear(animated: Bool) {  
    super.viewDidAppear(animated)  
  
    permissionProxy = GNSPermission { [weak self] granted in  
        if granted {  
            // show active UI  
        } else {  
            // show inactive UI  
        }  
    }  
}
```



NearbyViewController.swift

```
private var currentPub: GNSPublication?  
  
...  
  
if let message = message {  
    // shows permissions request dialog  
    // can also pass params to set role and range  
    currentPub = messageManager.publicationWithMessage(message)  
}
```



NearbyViewController.swift

```
private var currentPub: GNSPublication?  
...  
private func stopPublishing() {  
    currentPub = nil  
    // show inactive UI  
}
```



NearbyViewController.swift

```
private var currentSub: GNSSubscription?  
  
...  
  
// can also pass params to set role and range  
currentSub = messageManager.subscriptionWithMessageFoundHandler(  
    { [weak self] foundMessage in  
        // received once; deserialize data and cache it  
    },  
    messageLostHandler: { [weak self] lostMessage in  
        // received once; deserialize data and remove from cache  
    }  
)
```



NearbyViewController.swift

```
private var currentSub: GNSSubscription?  
...  
private func stopSubscribing() {  
    currentSub = nil  
    // show inactive UI  
}
```



NearbyViewController.swift

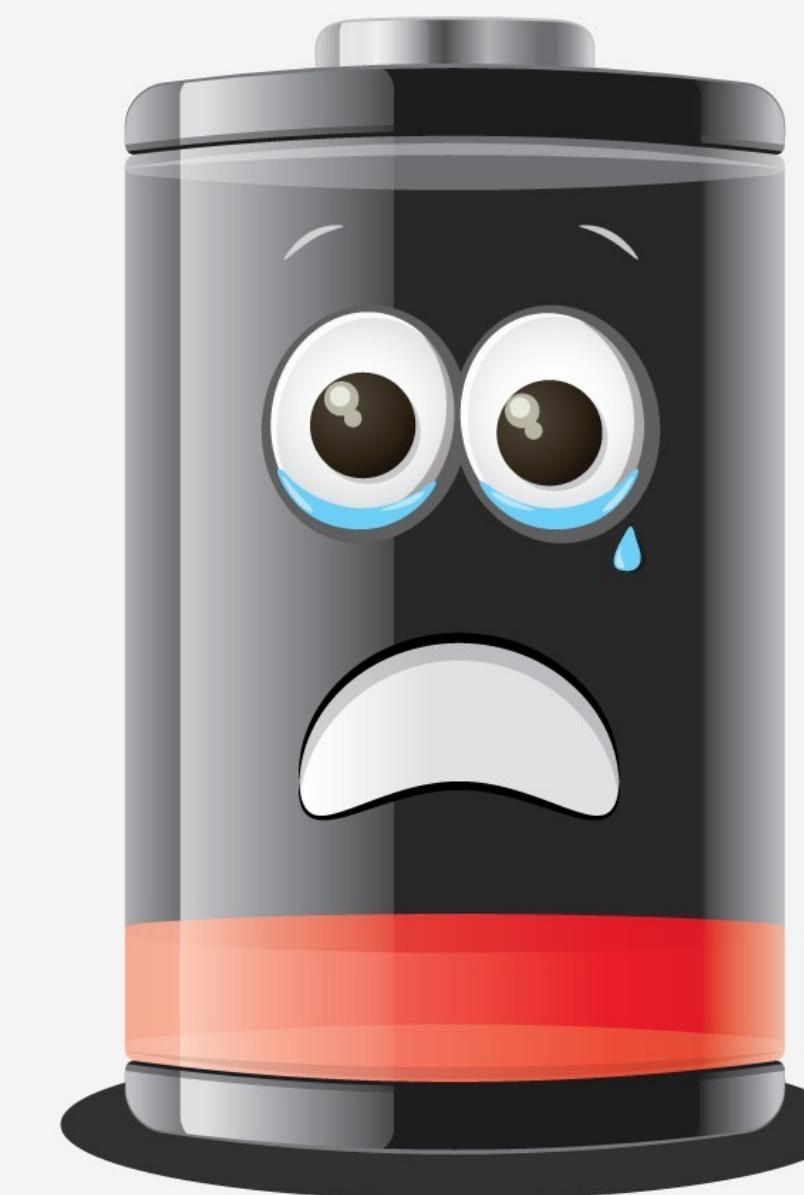
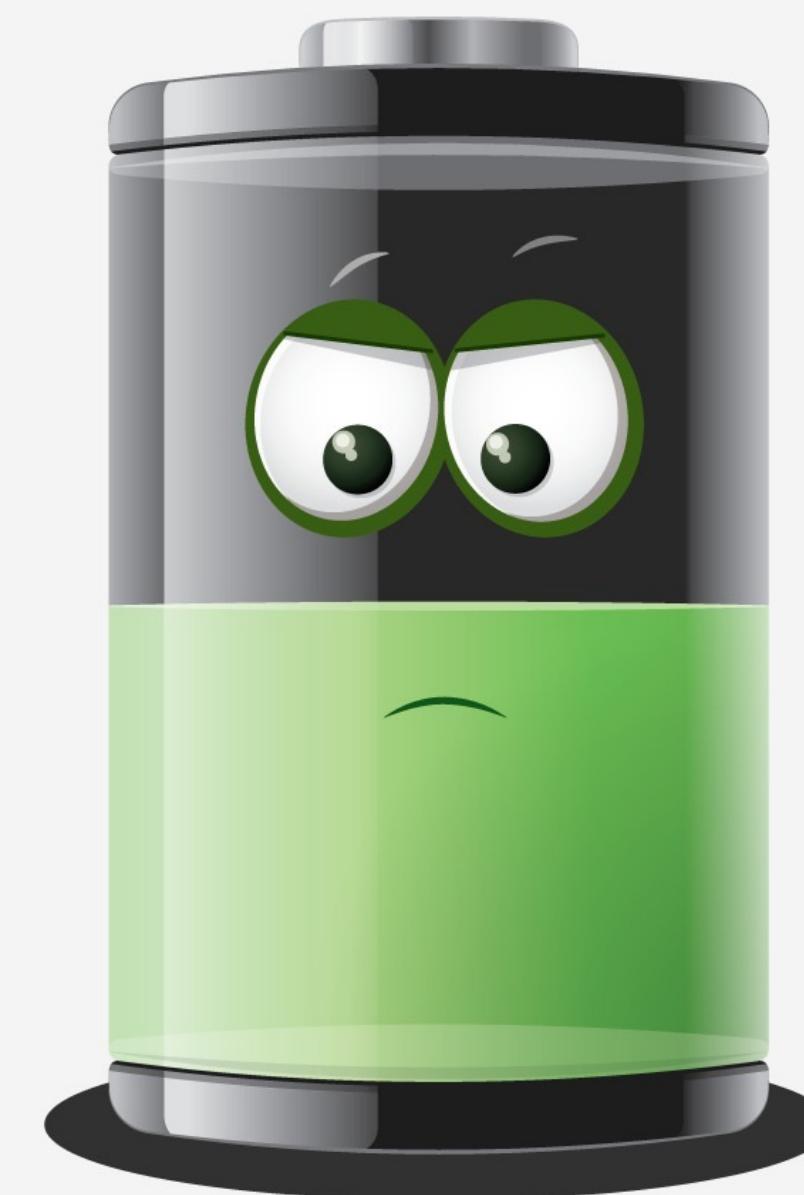
```
override func viewWillDisappear(animated: Bool) {  
    // cancel all Nearby operations  
    super.viewWillDisappear(animated)  
}
```



RESPONSIBILITIES

Battery

Be a good citizen

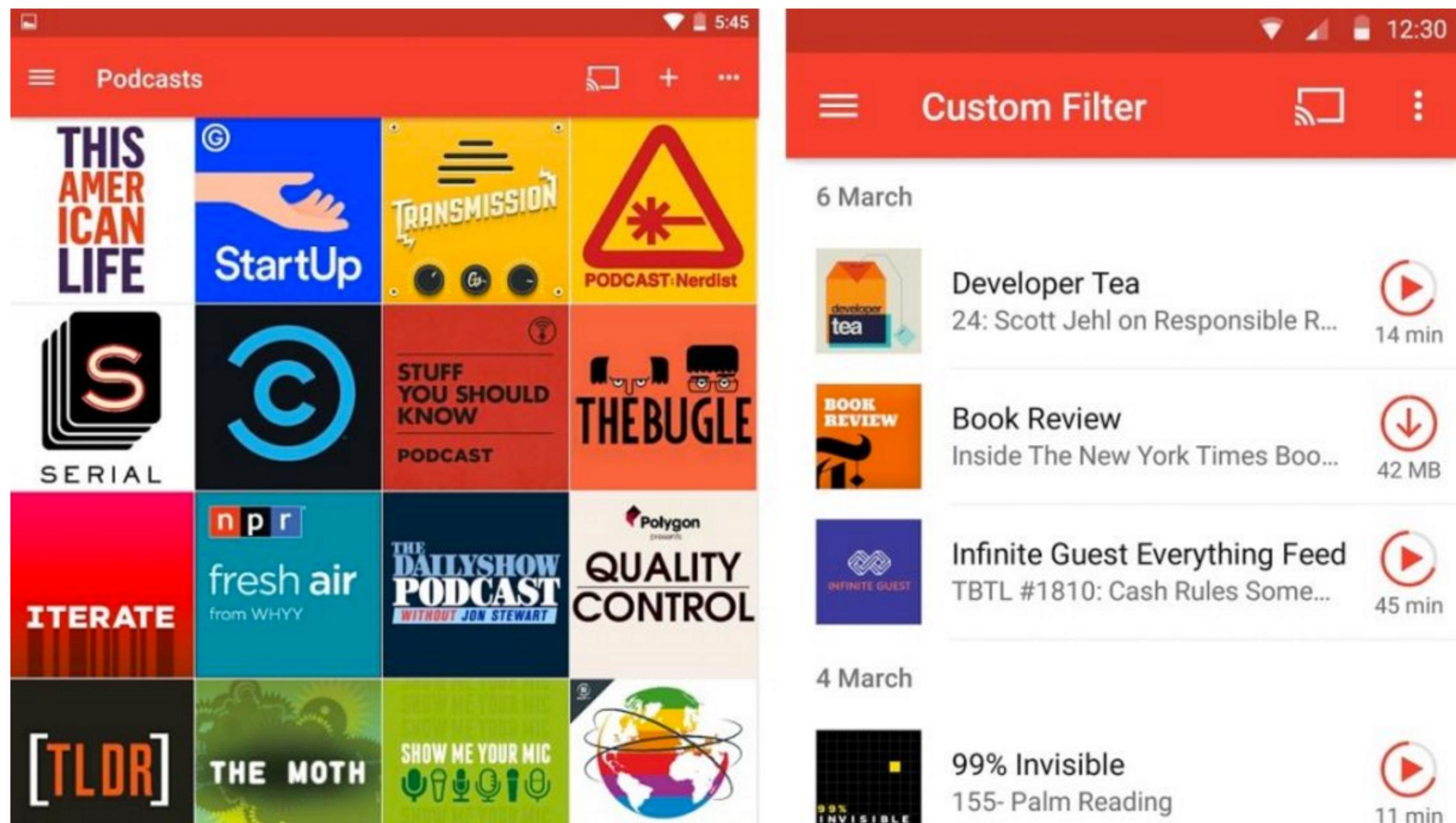


Privacy

Pocket Casts now lets you spy on what people nearby are listening to

By Matt Hanson August 20, 2015 Mobile phones 

Update shows you what's popular in your area



Anonymity

VS

Authenticity

Privacy



"Yik Yak is a **location-based social network** that helps people discover their local community, letting them share news, crack jokes, offer support, ask questions, and interact freely."

Anonymity

vs

Authenticity

Privacy

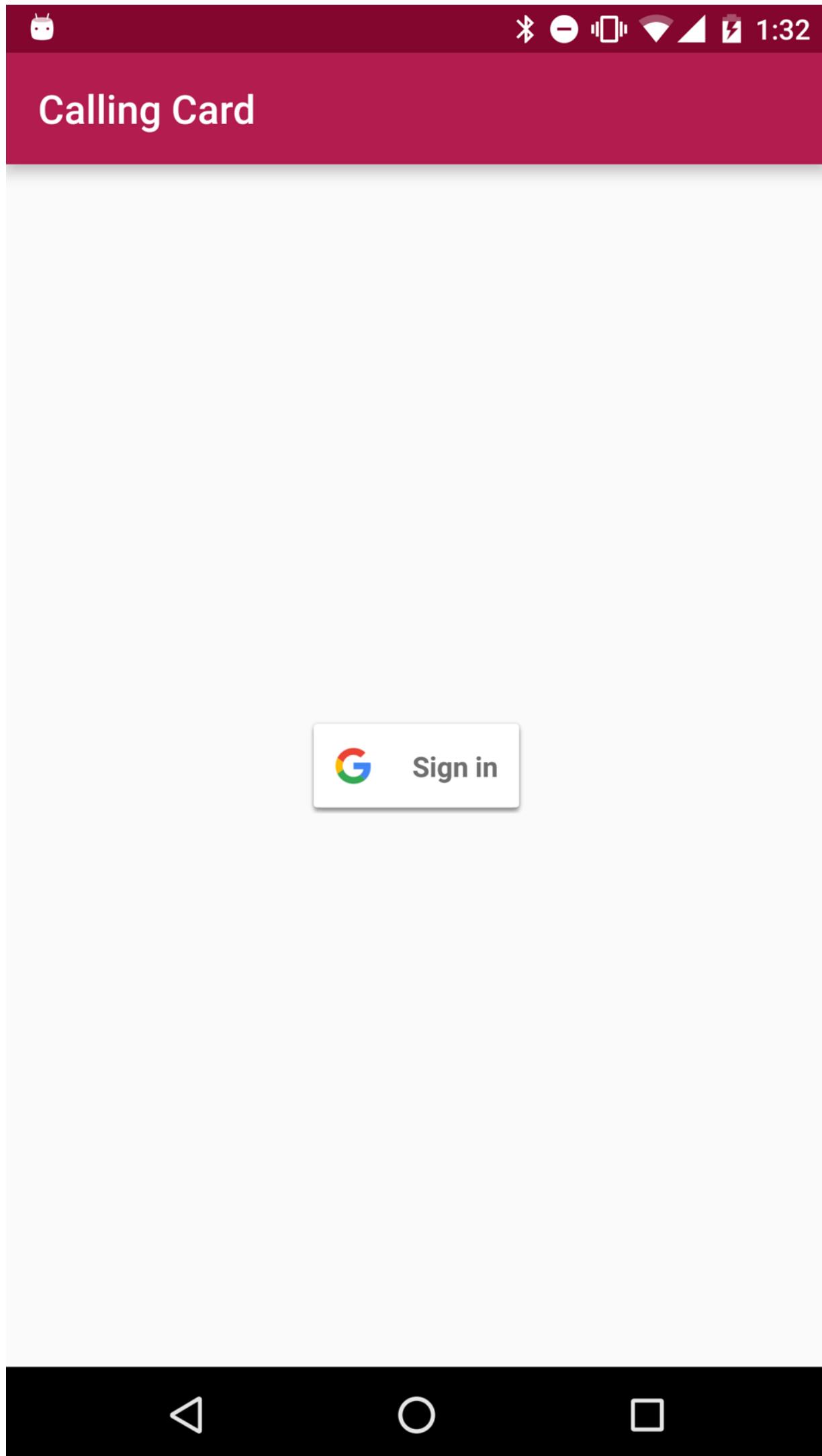
Brogrammers' Anonymous App Yik Yak Takes Off
Amid Controversy



STEVE JENNINGS/GETTY IMAGES

Anonymity
vs
Authenticity

Privacy



Anonymity

vs

Authenticity

OPPORTUNITIES

Trajectories

- Augmentation

Trajectories

- Augmentation
- Foundation

Trajectories

- Augmentation
- Foundation
- Competition

Trajectories

- Augmentation
- Foundation
- Competition
- Evolution

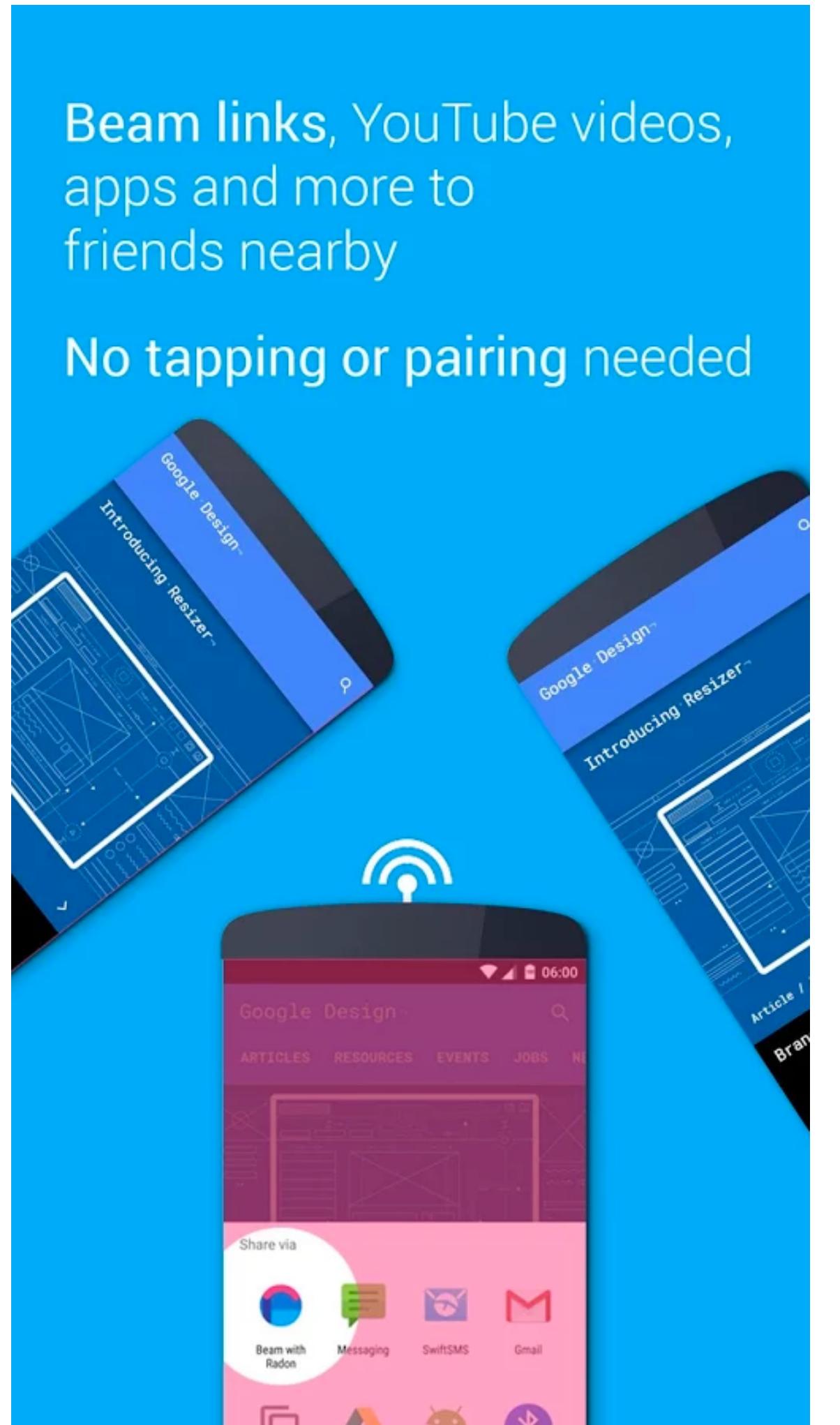
Augmentation

The screenshot shows the 'Discover' screen of the Pocket Casts mobile application. At the top, there is a red header bar with the word 'Discover' in white. Below the header are five tabs: 'FEATURED', 'TRENDING', 'TOP', 'NEARBY' (which is currently selected), and 'NETWORKS'. To the right of these tabs is a 'CATEGORIES' section. The main content area displays four podcast cards:

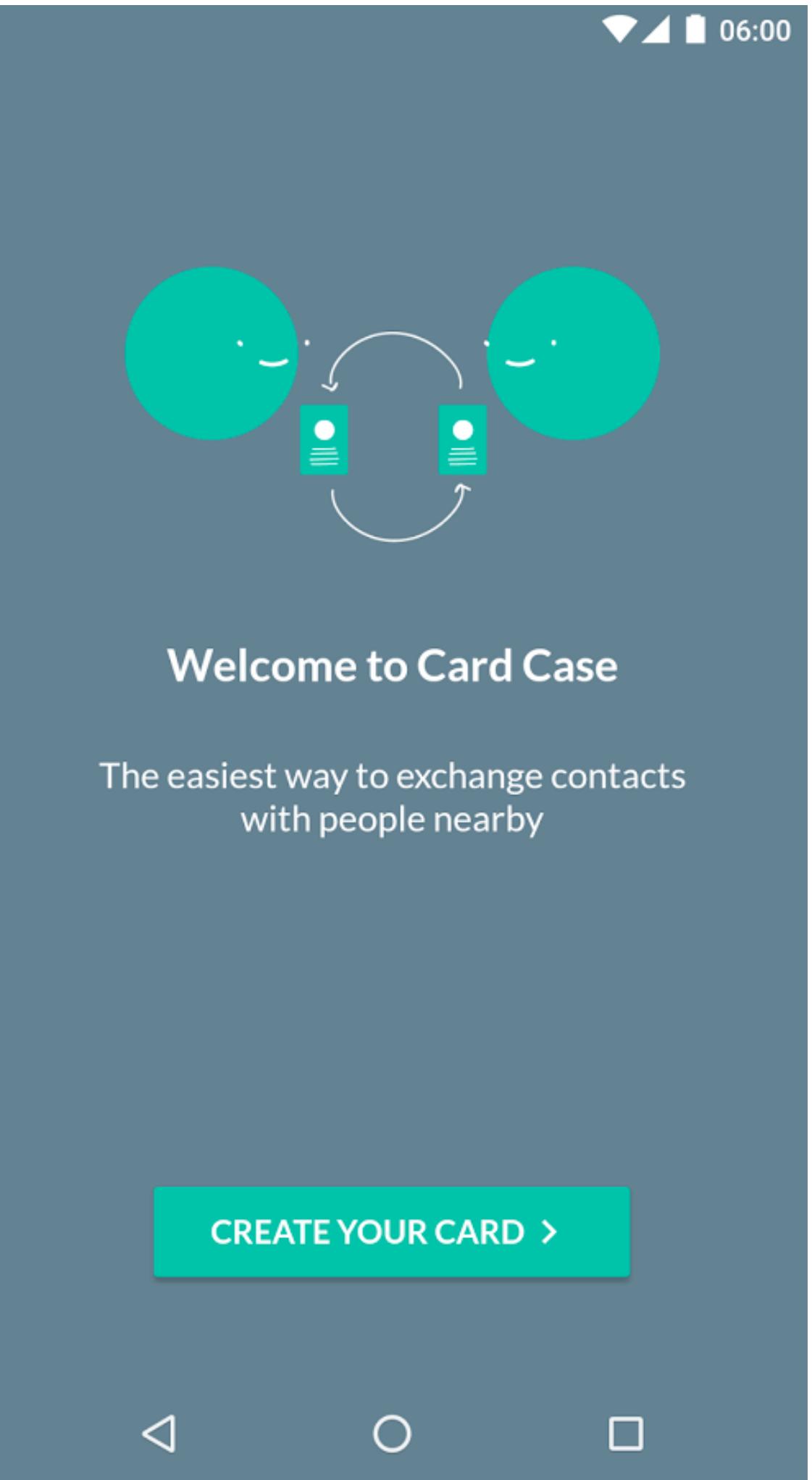
- Freakonomics Radio** by WNYC Studios and Dubner Productions. The card includes the show logo, the title, and the producer information.
- Material** by Relay FM. The card includes the show logo, the title, and the producer information.
- /dev/loop - Audio** by Lincoln Loop. The card includes the show logo, the title, and the producer information.
- 99% Invisible** by Roman Mars. The card includes the show logo, the title, and the producer information.

On the right side of the screen, there is a vertical black sidebar with several icons: a square, a circle, a triangle, and a plus sign. There are also three green circular icons with the numbers 2, 2, and 1, each accompanied by a plus sign, likely indicating new items or notifications. The bottom right corner of the sidebar has a large white plus sign.

Foundation



Radon



Card Case



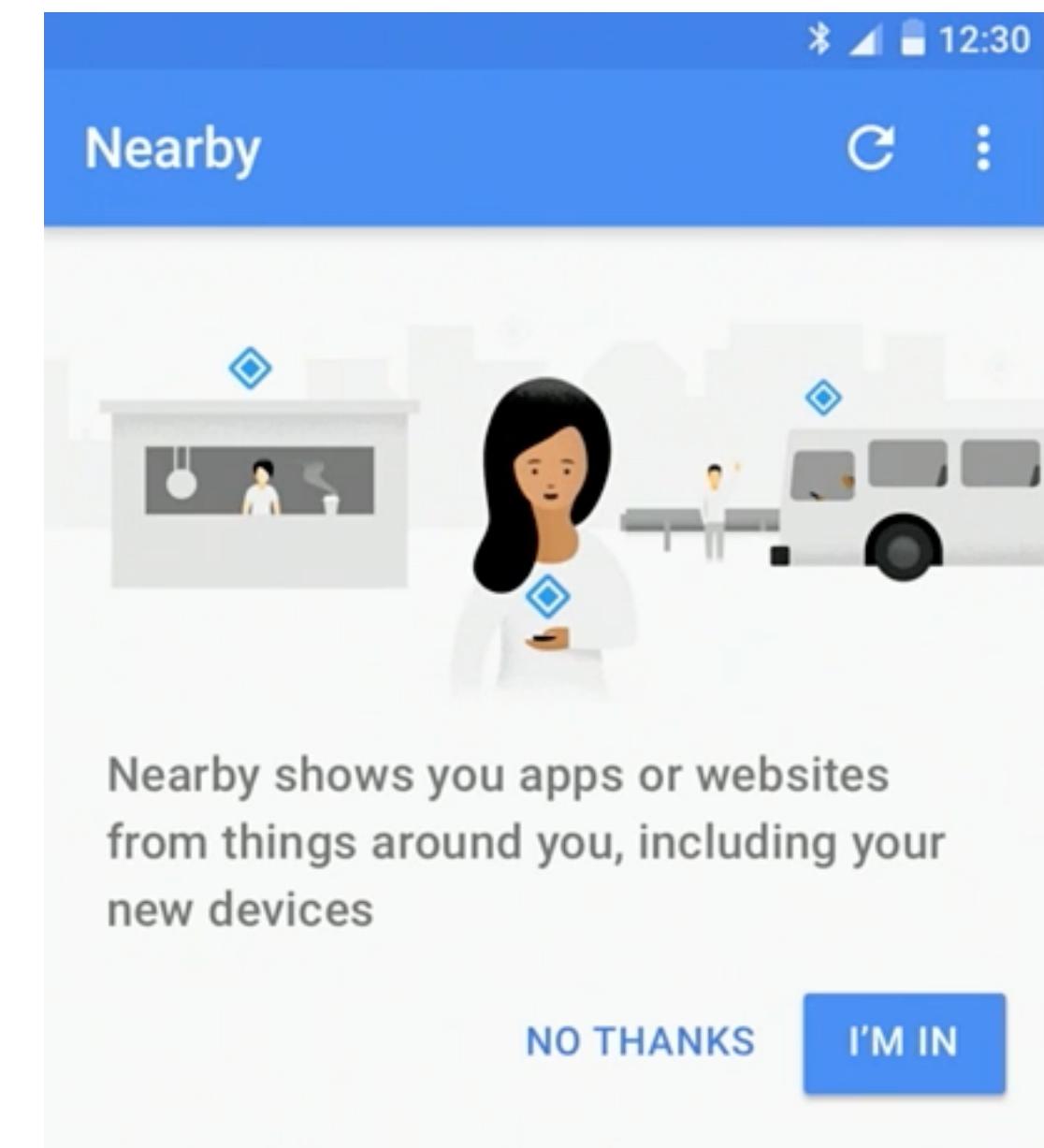
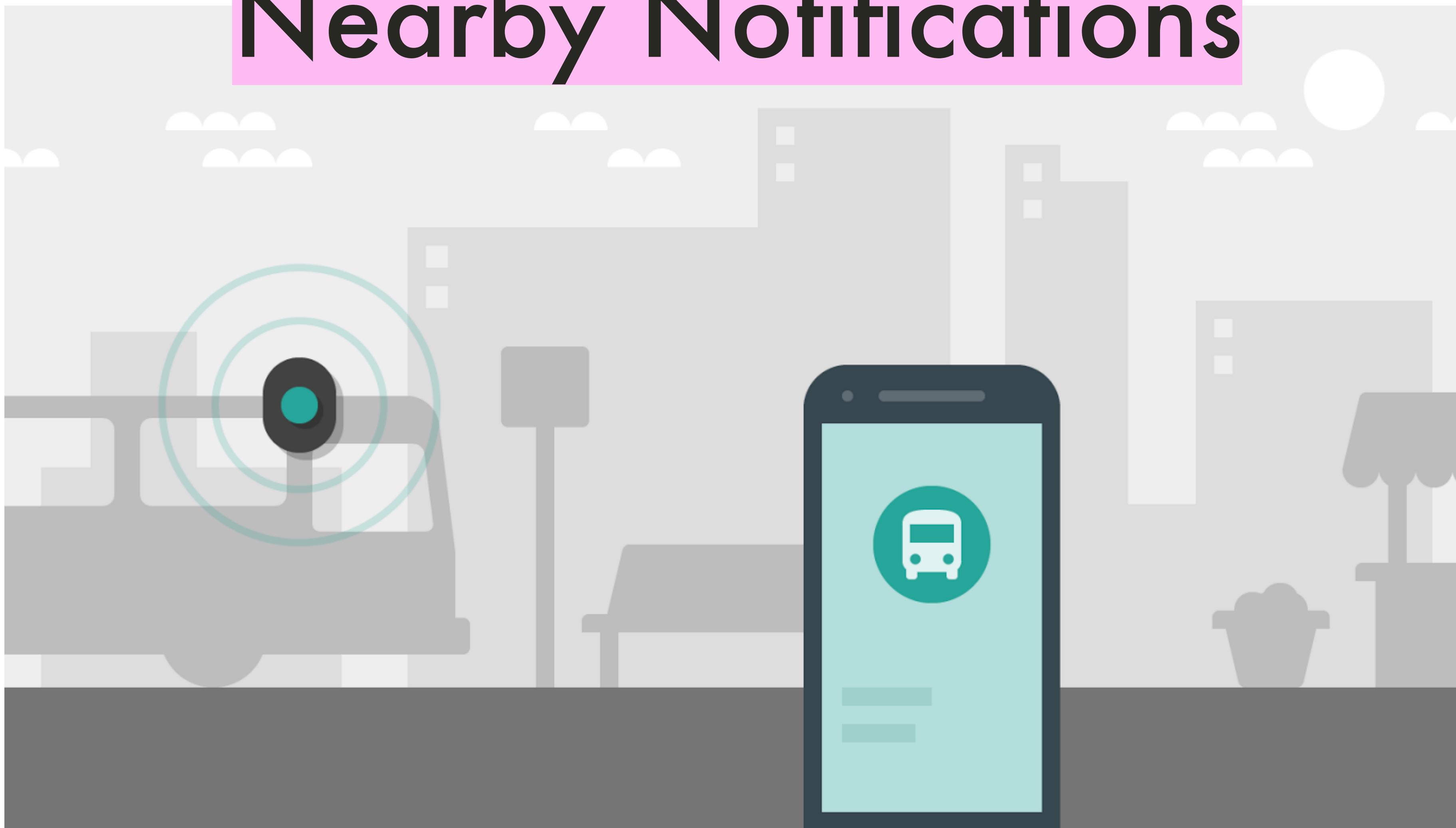
Competition



p2kit
PROXIMITY IS HERE

Evolution

Nearby Notifications



THANKS!

Questions?

Stuart Kent • stuart@detroitlabs.com • @skentphd