



TRIFORK.
...think software

iBeacons for Android
GOTO Night

Classic vs. BLE

Technical Specification	Classic Bluetooth technology	Bluetooth low energy technology
Distance/Range (theoretical max.)	100 m (330 ft)	>100 m (>330 ft)
Over the air data rate	1–3 Mbit/s	1 Mbit/s
Application throughput	0.7–2.1 Mbit/s	0.27 Mbit/s
Active slaves	7	Not defined; implementation dependent
Security	56/128-bit and application layer user-defined	128-bit AES with Counter Mode CBC-MAC and application layer user-defined
Robustness	Adaptive fast frequency hopping, FEC , fast ACK	Adaptive frequency hopping, Lazy Acknowledgement, 24-bit CRC , 32-bit Message Integrity Check
Latency (from a non-connected state)	Typically 100 ms	6 ms
Total time to send data (def. battery life)	100 ms	3 ms, [citation needed] <3 ms ^[21]
Voice capable	Yes	No
Network topology	Scatternet	Scatternet
Power consumption	1 as the reference	0.01 to 0.5 (depending on use case)
Peak current consumption	<30 mA	<15 mA
Service discovery	Yes	Yes
Profile concept	Yes	Yes
Primary use cases	Mobile phones, gaming, headsets, stereo audio streaming, automotive, IoT, security, proximity, healthcare, sports & fitness, etc.	Mobile phones, gaming, PCs, watches, sports and fitness, healthcare, security & proximity, automotive, home electronics, automation, industrial, etc.

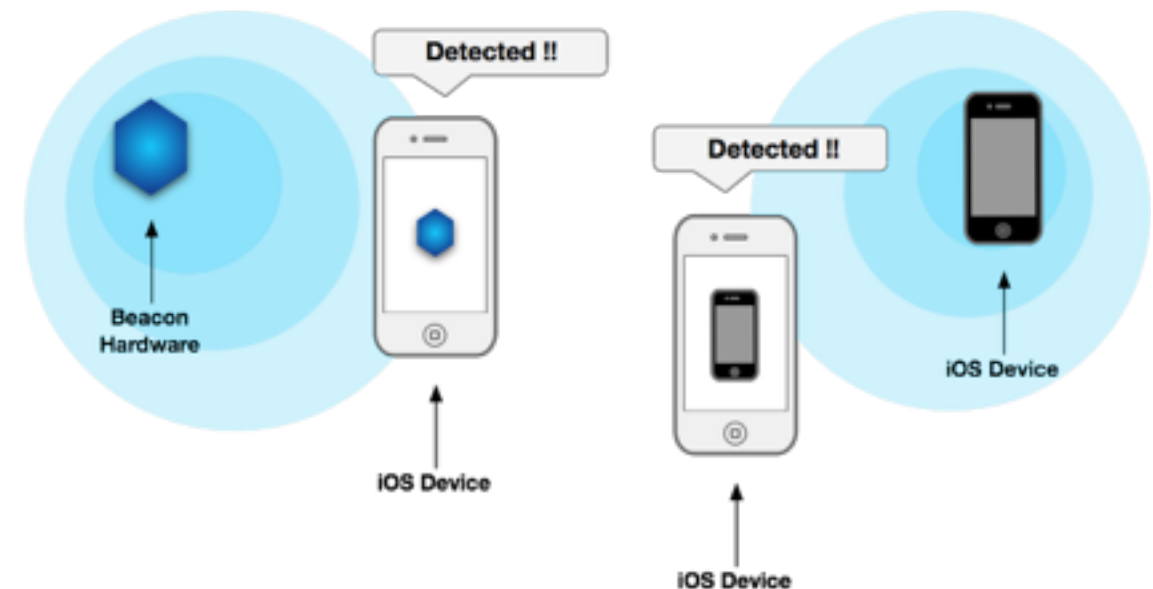
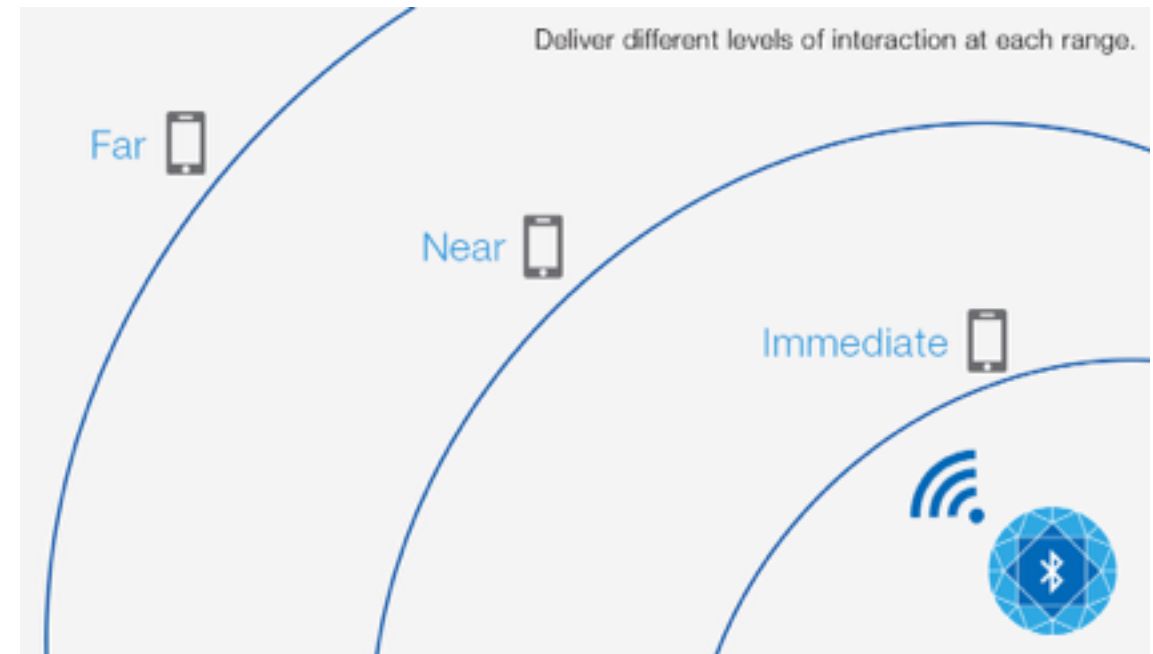
BLE support

- Requires a Bluetooth LE chip. Only newer phones/tablets has it.
- Android: 4.3 - SDK 18+ (29%), Samsung Galaxy S3+, Nexus 4+, Nexus 7 2013
- iOS: 7.0+ (91%), iPhone 4s+, iPad 3rd generation+



What is iBeacons

- iBeacon is a BLE profile specification formalised by Apple. It is not a standard.
- It can work on all devices that support BLE, not just iOS.
- 2 primary use cases: Proximity and Ranging.
- Works indoors.

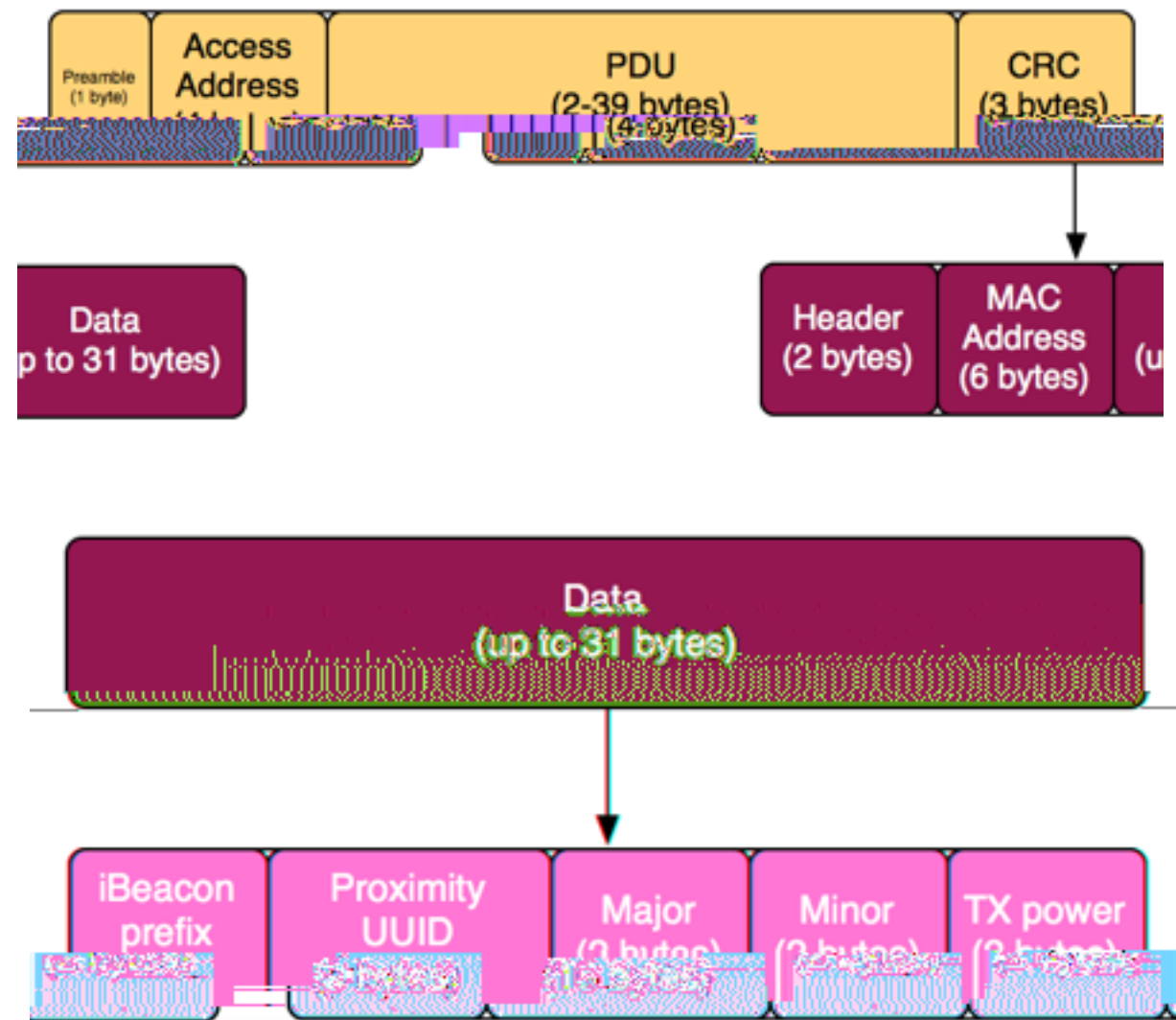


What is iBeacons

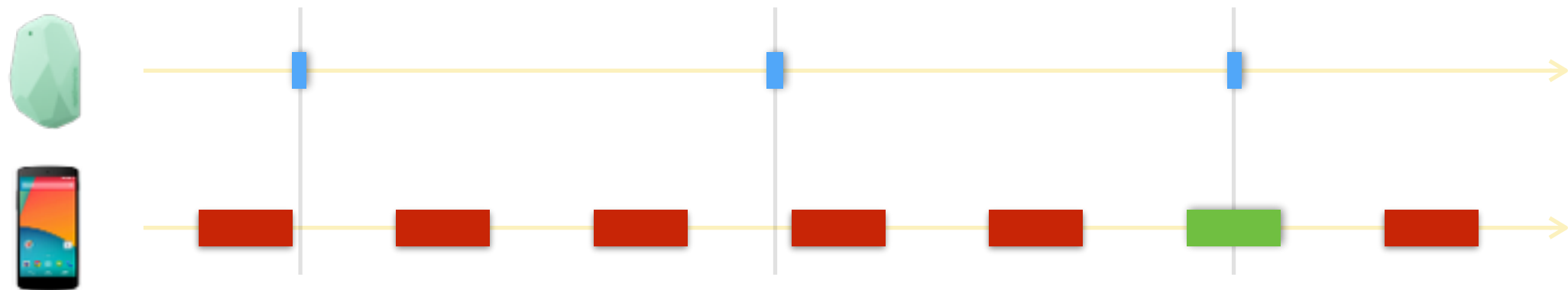


iBeacon protocol

- Standard BLE advertisement.
- Mac-address is spoofed by some vendors.
- Proximity UUID: 16 byte UUID
- Major: 0 - 65.535
- Minor: 0 - 65.535
- TxPower: Signal Strength at 1m from iBeacon.



Detecting iBeacons



- Drive-by-detection is hard. Battery matters on both ends.
- Devices do not talk to each other. Phone simply listens to advertisements from iBeacon.
- **Tip:** Scan times should be non-regular numbers, eg. scan period of 1.1s. instead of 1s.

Range

- Hard problem. Many scientific papers on the subject.
- Water blocks bluetooth signals. Guess what humans are mostly made of?
- Recommendation is to use relative range instead of absolute.
- You know if you are close to an iBeacon, otherwise you don't know anything.

```
protected static double calculateAccuracy(int txPower, double rssi) {  
    if (rssi == 0) {  
        return -1.0; // if we cannot determine accuracy, return -1.  
    }  
  
    double ratio = rssi*1.0/txPower;  
    if (ratio < 1.0) {  
        return Math.pow(ratio,10);  
    }  
    else {  
        double accuracy = (0.89976)*Math.pow(ratio,7.7095) + 0.111;  
        return accuracy;  
    }  
}
```

```
protected static double calculateDistance(int txPower, double rssi) {  
    if (rssi == 0) {  
        return -1.0; // if we cannot determine accuracy, return -1.  
    }  
  
    double ratio = rssi*1.0/txPower;  
    double distance;  
    if (ratio < 1.0) {  
        distance = Math.pow(ratio,10);  
    }  
    else {  
        distance = (0.42093)*Math.pow(ratio,6.9476) + 0.54992;  
    }  
    return distance;  
}
```


iBeacons on Android

- Android doesn't provide a common API. Each app must implement iBeacon support.
- Android GeoFencing API does not support iBeacons. Will probably not happen due to Apple trademark rights.
- Vendor SDK's exist that makes this very easy: Radius Network, Estimote, Gimbal ...



-



```
public static boolean isEstimateBeacon(Beacon beacon)
{
    return (isMacBeacon(beacon)) || (isIOSBeacon(beacon)) || (isAdvertisingEstimateId(beacon)) || (isAdvertisingName(beacon, adName));
}
```

Radius Network



- Open Source
- Previous version of Android SDK was shut down by Apple.
- Current implementation is “Beacon agnostic”
- <https://github.com/AltBeacon/android-beacon-library>

AltBeacon

- Competing spec from Radius Network.
- Beacons can emit two types of advertisements to support this.
- Highly unlikely this will gain traction.
- <http://altbeacon.org/>
- <https://news.ycombinator.com/item?id=8098681>



Getting started

- Clone Github repo or download JAR.
 - <https://github.com/AltBeacon/android-beacon-library>
 - <http://altbeacon.github.io/android-beacon-library/download.html>
- Not in Maven Central yet.

<https://github.com/RadiusNetworks/android-ibeacon-service>

Rewind history for better API,
but no bug fixes and support.



Vocabulary

- ID1 = Proximity UUID
- ID2 = Major
- ID3 = Minor
- TxPower = Measured Power

iBeacon setup

- AndroidManifest.xml

```
<uses-feature android:name="android.hardware.bluetooth_le" android:required="true" />  
<uses-permission android:name="android.permission.BLUETOOTH"/>  
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
```

iBeacon setup

```
public class BeaconController implements BeaconConsumer {

    private Context context;

    public BeaconController(Context context) {
        this.context = context;
        // TODO Initialize BeaconManager
    }

    @Override
    public void onBeaconServiceConnect() {
        // BeaconService ready.
    }

    @Override
    public Context getApplicationContext() {
        return context(getApplicationContext());
    }

    @Override
    public void unbindService(ServiceConnection connection) {
        context.unbindService(connection);
    }

    @Override
    public boolean bindService(Intent intent, ServiceConnection connection, int mode) {
        return context.bindService(intent, connection, mode);
    }
}
```

iBeacon setup

```
BeaconManager beaconManager = BeaconManager.getInstanceForApplication(context);
beaconManager.getBeaconParsers().set(0,
    new BeaconParser().setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,i:22-23,p:24-24")
);

beaconManager.setForegroundScanPeriod(1100);
beaconManager.setForegroundBetweenScanPeriod(0);
beaconManager.setBackgroundScanPeriod(10000);
beaconManager.setBackgroundBetweenScanPeriod(5*60*1000);
beaconManager.setBackgroundMode(false);

beaconManager.bind(this); // onBeaconServiceConnect() called when ready
```

Monitoring

```
beaconManager.setMonitorNotifier(new MonitorNotifier() {
    @Override
    public void didEnterRegion(final Region region) {
        // Callback after scan cycle if region entered.
    }

    @Override
    public void didExitRegion(Region region) {
        // Callback after scan cycle if region was exited.
    }

    @Override
    public void didDetermineStateForRegion(int i, Region region) {
        // Callback after scan cycle for every region that switched state
        // MonitorNotifier.INSIDE or MonitorNotifier.OUTSIDE
    }
});

try {
    Region region = new Region("foo", Identifier.parse("UUID"), null, null);
    beaconManager.startMonitoringBeaconsInRegion(region);
} catch (RemoteException e) {
    // Callback after scan cycle for every region that switched state
}
```

Ranging

```
beaconManager.setRangeNotifier(new RangeNotifier() {
    @Override
    public void didRangeBeaconsInRegion(Collection<Beacon> iBeacons, Region region) {
        // Callback after every scan cycle
    }
});

try {
    Region region = new Region("foo", Identifier.parse("UUID"), null, null);
    beaconManager.startRangingBeaconsInRegion(region);
} catch (RemoteException e) {
    // Ranging failed
}
```

Advertisement - L

```
BluetoothManager btManager = (BluetoothManager)
getApplicationContext().getSystemService(Context.BLUETOOTH_SERVICE);
BluetoothLeAdvertiser bleAdvertiser = btManager.getAdapter().getBluetoothLeAdvertiser();

AdvertisementData data = new AdvertisementData.Builder()
    .setManufacturerData(0x0118, constructAdvertisementPackage(transmitBeaconData.getBeacon(), -56))
    .build();

AdvertiseSettings settings = new AdvertiseSettings.Builder()
    .setAdvertiseMode(AdvertiseSettings.ADVERTISE_MODE_BALANCED)
    .setTxPowerLevel(AdvertiseSettings.ADVERTISE_TX_POWER_HIGH)
    .setType(AdvertiseSettings.ADVERTISE_TYPE_NON_CONNECTABLE)
    .build();

AdvertiseCallback callback = new AdvertiseCallback() {
    @Override
    public void onSuccess(AdvertiseSettings advertiseSettings) {
        // Advertisement started
    }

    @Override
    public void onFailure(int i) {
        // Advertisement failed
    }
};

bleAdvertiser.startAdvertising(settings, data, callback);
```

Setting	Transmit Frequency
ADVERTISE_MODE_LOW_LATENCY	approx 1 Hz
ADVERTISE_MODE_BALANCED	approx 3 Hz
ADVERTISE_MODE_LOW_POWER	approx 10 Hz

Setting	RSSI @1m (iOS)	RSSI @1m (Mac)
ADVERTISE_TX_POWER_HIGH	-56 dBm	-55 dBm
ADVERTISE_TX_POWER_MEDIUM	-66 dBm	-66 dBm
ADVERTISE_TX_POWER_LOW	-75 dBm	-70 dBm
ADVERTISE_TX_POWER_ULTRA_LOW	(not detected)	-79 dBm

Advertisement - L

```
public byte[] constructAdvertisementPackage(Beacon beacon, int measuredPower) {
    byte[] advertisementPackage = new byte[25];
    advertisementPackage[0] = (byte) 0x4C; // Apple ID, Big Endian
    advertisementPackage[1] = (byte) 0x00;
    advertisementPackage[2] = (byte) 0x02; // Identifies advertisement as iBeacon
    advertisementPackage[3] = (byte) 0x15;
    System.arraycopy(uuidToBytes(beacon.getId1().toString()), 0, advertisementPackage, 4, 16);
    System.arraycopy(uint16ToBytes(beacon.getId2().toInt()), 0, advertisementPackage, 20, 2);
    System.arraycopy(uint16ToBytes(beacon.getId3().toInt()), 0, advertisementPackage, 22, 2);
    advertisementPackage[24] = int8ToByte(measuredPower);
    return advertisementPackage;
}
```

DEMO

<https://github.com/cmelchior/trifork-ibeacon-demo>

Software beacons

- Macs with Maverick and Bluetooth 4.0 support:
 - MacBeacon (10\$) : <http://www.radiusnetworks.com/ibeacon/macbeacon/>
 - BeaconOSX (free) : <https://github.com/deadfalkon/BeaconOSX/releases>
- iPhone 4s, iOS 7
- iPad 3, iOS 7
- Nexus 5 with Android L

Hardware beacons

- Gimbal (15\$ for 3)
- Estimote (99\$ for 3)
- Radius Network RadBeacon (20\$ for 1)
- **Tip:** Try before buying many.
- <http://www.fosbury.co/beacon-comparison>

Hardware beacons

- Still a very young field.
- All vendors are in “beta”.
- Most of them try to keep you in their own eco-system.
- **Tip:** Consider how you want to configure and maintain beacons.

Beacons and battery life

		Broadcasting power		
		-30 dBm [low]	-4 dBm	+4 dBm [high]
Advertising interval	2000 ms [long]	3.3 years	3 years	2.3 years
	1000 ms	1.9 years	1.7 years	1.3 years
	600 ms	1.2 years	1 year	300 days
	200 ms	160 days	140 days	104 days
	50 ms [short]	40 days	35 days	26 days

Source: <http://blog.estimote.com/post/83618039493/how-to-extend-estimote-beacon-battery-life>

Gotya's

- Android Bluetooth stack is flawed. Crashes when it has seen ~2000 MAC's: <https://code.google.com/p/android/issues/detail?id=67272>
- Nexus 4: Wifi and Bluetooth cannot run at the same time. Combined Wifi+Bluetooth chips causes problems.
- Beware of ranging, you will get fooled by the distance.
- Do not trust the iBeacon. Only use them as hints.

QUESTIONS?

cme@trifork.com
@chrnelchior

Additional links

- <http://en.wikipedia.org/wiki/IBeacon>
- <http://www.warski.org/blog/2014/01/how-ibeacons-work/>
- <http://stackoverflow.com/questions/18906988/what-is-the-ibeacon-bluetooth-profile>
- <http://stackoverflow.com/questions/20416218/understanding-ibeacon-distancing>
- <https://github.com/AltBeacon/altbeacon-transmitter-android>
- <https://www.youtube.com/watch?v=vUbFB1Qypg8>
- <https://www.youtube.com/watch?v=2fZThdNbHcQ#t=187>
- <https://www.youtube.com/watch?v=4POOiVrdnX8>
- <http://localz.co/blog/ibeacon-ble-hardware-commercial-comparison/>