



Bluetooth[®]

Dawid Gdański



Hi there!

1. Device's working modes
2. GATT, Services, Characteristics
3. LOLLIPOP improvements
4. Versioning
5. Performance...
6. Testing
7. Building your SDK
8. Why should you care?



Device's working modes

- Advertising by default
- Pairing and data exchange



GATT, Services, Characteristics

Consider map:

```
Map<GattService, GattCharacteristic> GATT;
```

Reading device's parameters:

```
GattCharacteristic transmissionPower  
    =  
GATT.get(GattServices.TRANSMISSION_POWER);
```

Writing to the device:

```
transmissionPower.setValue(Integer.MAX_VALUE);
```



LOLLIPOP improvements

- Scan modes
- Filtering
- Advertising mode
- Render thread



Versioning

- Support for filtering
- @Deprecated
`BluetoothAdapter.startLeScan(LeScanCallback)`
- Scan modes - unfortunately...



Performance

Optimization helpers:

- ADM's goods
- Jake Wharton's Hugo
- Anyone?



Testing:

- Robolectric:
testCompile 'org.robolectric:robolectric:2.4' or
testCompile 'org.robolectric:robolectric:3.0-SNAPSHOT'
- To mock final classes:
testCompile 'org.powermock:powermock-api-mockito:1.6.+'
testCompile 'org.powermock:powermock-module-junit4:1.6.+'- No emulators.



Building your SDK

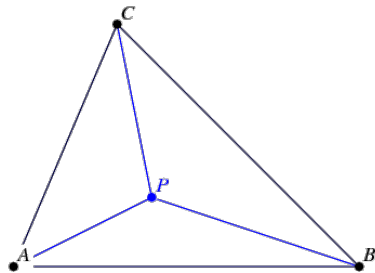
1. Service-backed Manager "pattern".
2. Scan types:
 - Ranging
 - Monitoring
 - Up to you
3. Thread-based Consumer-Producer pattern.
4. Caching:
 - "Limited" LinkedHashMap
 - LruCache
5. Immutability - not always
6. Device recognition - prefer hash code rather than e.g. device's MAC address



Why should you care?

1. BLE - simple, easy applicable, lots of use cases.
2. Facebook tests Bluetooth 'Beacons' to feed users local content.

3. Triangulation.



4. Putting context into physical places.
5. Why shouldn't you actually?



Questions?



Sample to start, enjoy!

<https://github.com/dawidgdanski/BLE-matters>

<https://github.com/kontaktio/kontakt-beacon-admin-sample-app>



Thank you.

