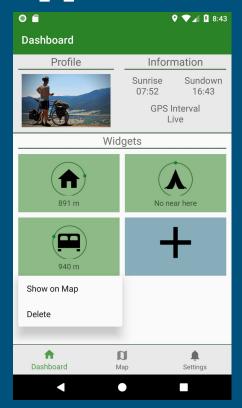
# The Adventurer App

**Final Product** 



# The Application





Live-Demo

# Technologies



### Adventurer App





Sensors

Magnetic Field for compass

Google Location API

dynamic GPS

OSMDroid Map Data and POI

# Challenges

#### Connectivity

Usage of Offline Maps and cached Map-Data, when possible

**Battery Optimization** 

Optimize energy consumption depending on the type of adventure

Connection Info

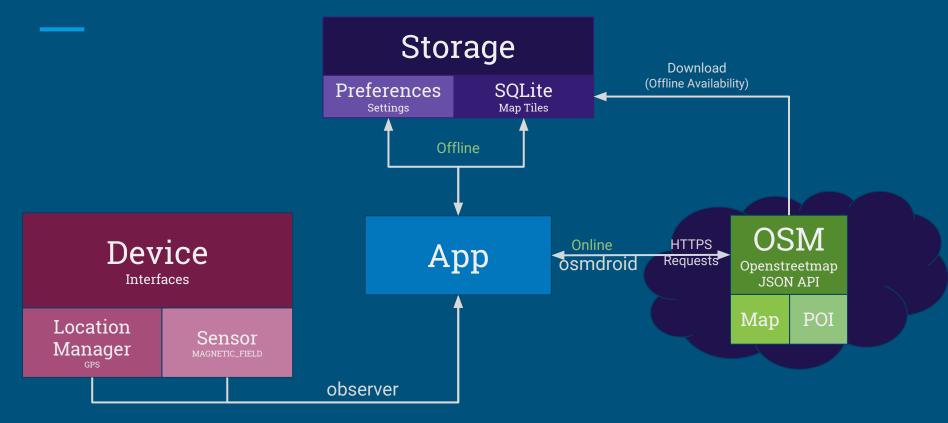
Adventurer profile

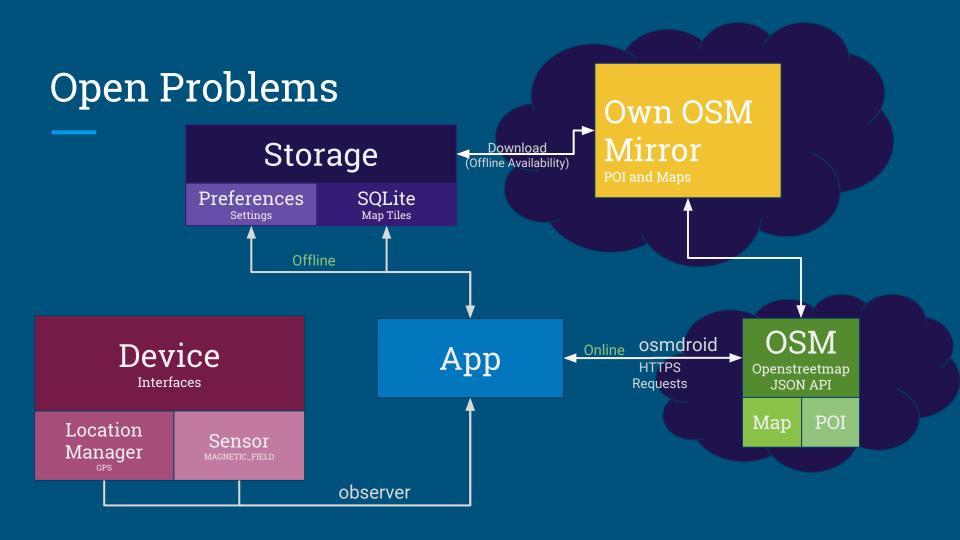
Age of cached data

context information

adaptation mechanisms

## Final Architecture





## Open Problems

- Issues with public OSM Servers
- They have limited bandwidth and requests
  - You can't load > 500 POIs at the same time
  - Map download is very slow
  - Fix: Setup own server with OSM data and provide the data that way

#### What have we learned?



#### Code Comparison Java

```
TextView text = (TextView) findViewById(R.id.textView);
text.setText("Hello World");
```

#### Kotlin

textView.setText("Hello World")

What have we learned?

Kotlin >> Java

Creating

performance issues

is easy

Correcting them is hard

Implementing compass and GPS is harder than expected

