



Final Presentation: KarmaSurf

Bink, Raphael
3678968

Füsslin, Maximilian
3689756

Dresden, 29.01.2016



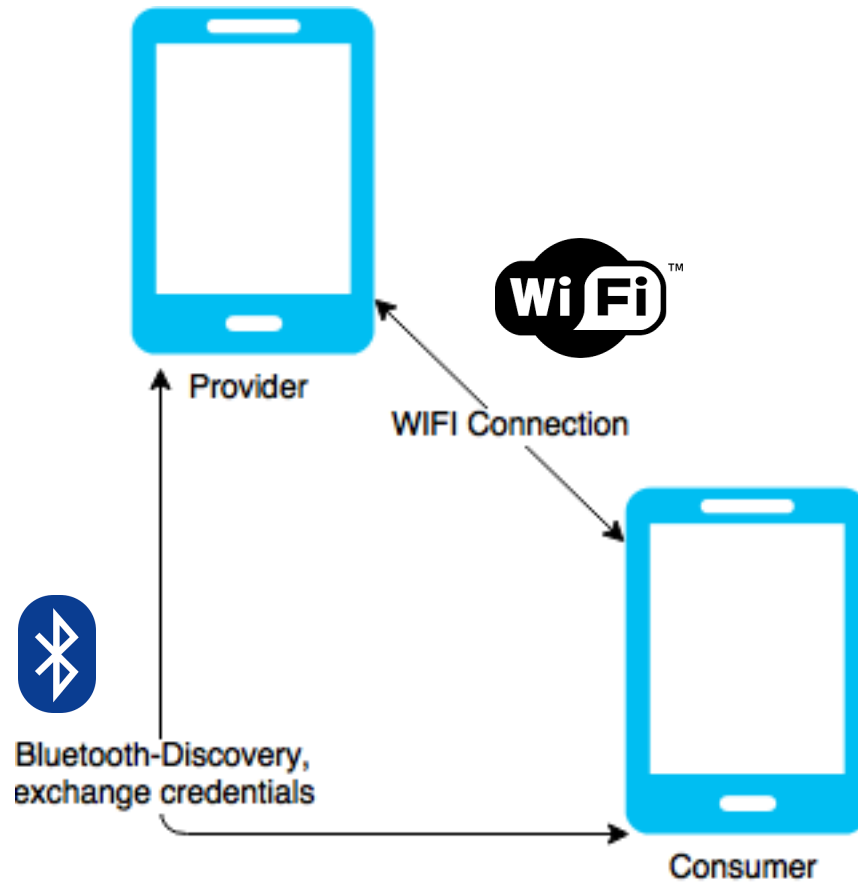
- Application Scenario
- Technologies and Architecture
- Challenges
- Adaption and Context
- Experiences and Pitfalls

Provide Network
Connectivity for
foreigners

Avoidance of Roaming



Do something nice with your „overcapacity“



- Background Service for calc. Karmapoints
- SQLite for data persistence
- Insecure Bluetooth Connection

Challenges that we provide a solution for:



- Connectivity Challenge
 - With KarmaSurf you have access to mobile data f. e. in foreign countries, on the road or if you already exceeded your monthly amount of traffic.
- Offline Challenge
 - With broader connectivity, you are less dependent on offline apps/functionality.

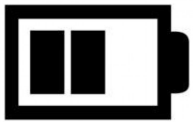
- Challenges that we provide a solution for:



- Usability Challenge

- Providing data or connecting to an access point should be possible with minimal user interaction.
- **Our Solution:** Provider and User-Apps communicate without user input: One-tap-App
 - The only user input required is to change the perspective with a switch, which has the states *Consumer*, *StandBy* and *Provider*.

- Challenges that we provide a solution for:



Energy Challenge

- To provide or search for an access point is energy consuming.
- **Our Solution:** Search is done by Bluetooth, whereas the interval of the search is adapted to current available battery power:
 - The lower the battery power is, the longer is the interval between discovery-processes.
 - Apart from that, the interval will increase if the user is inactive.


```
protected void startThreadLoop() {
    IntentFilter ifilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
    Intent batteryStatus = activity.registerReceiver(null, ifilter);

    int level = batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
    int scale = batteryStatus.getIntExtra(BatteryManager.EXTRA_SCALE, -1);
    float batteryPercentage = level / (float)scale;

    int seconds;

    if (batteryPercentage > 0.8) {
        seconds = 22;
    } else if (batteryPercentage > 0.5) {
        seconds = 35;
    } else if (batteryPercentage > 0.2) {
        seconds = 48;
    } else {
        seconds = 60;
    }
}
```

```
// This schedule a runnable task every xx seconds
this.scheduleTaskExecutor.scheduleAtFixedRate(() -> {
    bluetoothManager.cancelDiscovery();
    bluetoothManager.tryConnection();
    bluetoothManager.doDiscovery();

    if (scheduleTaskCloserExecutor == null) {
        scheduleTaskCloserExecutor = Executors.newScheduledThreadPool(5);

        scheduleTaskCloserExecutor.scheduleAtFixedRate(() -> {
            Log.d(TAG, "start ThreadLoop again...");
            startThreadLoop();
        }, 300, 300, TimeUnit.SECONDS);
    }
}, 0, seconds, TimeUnit.SECONDS);
```

Gameification of sharing/consuming traffic is realized by KarmaPoint-reward system, which is context aware:

Context Capturing

- Battery power
- Amount of traffic already shared/consumed
- Network type (faster is better, f.e. LTE > 3G)

Context Abstraction

- Measured values are interpreted, mathematical functions used to calculate karmapoints which scale depending on input values

Context Usage

- Context changes change the way, how KarmaPoints are calculated and discovery is done:
 - F.e. low battery: bigger discovery intervals and if spending traffic more points are gained

Adaption for mobile computing challenges:

- Adaption of **application structure**:
 - User and provider are *bounded dynamically*:
 - After credentials are transmitted via Bluetooth, the binding is done via Wifi
- Adaption of **communication**:
 - Wifi is only activated if provider/user of mobile traffic were discovered via bluetooth: Therefore communication between phones depends on situation.
- Adaption of **battery consumption**:
 - Bluetooth discovery intervall scheduled according to available battery power.

- Experiences
 - Android documentation is sometimes quite bad
 - Devices are different and needed to be covered different sometimes (f.e. Bluetooth management)
- Pitfalls
 - „insecure“ Bluetooth connections are not working properly on every device (many oddities)
 - Programmatically enabling access point is not natively supported by many devices (Reflection needed)