Second Presentation: KarmaSurf

Bink, Raphael
3678968

Füsslin, Maximilian
3689756

Dresden, 11.12.2015

- Application Scenario
- Architecture / Technologies
- Challenges
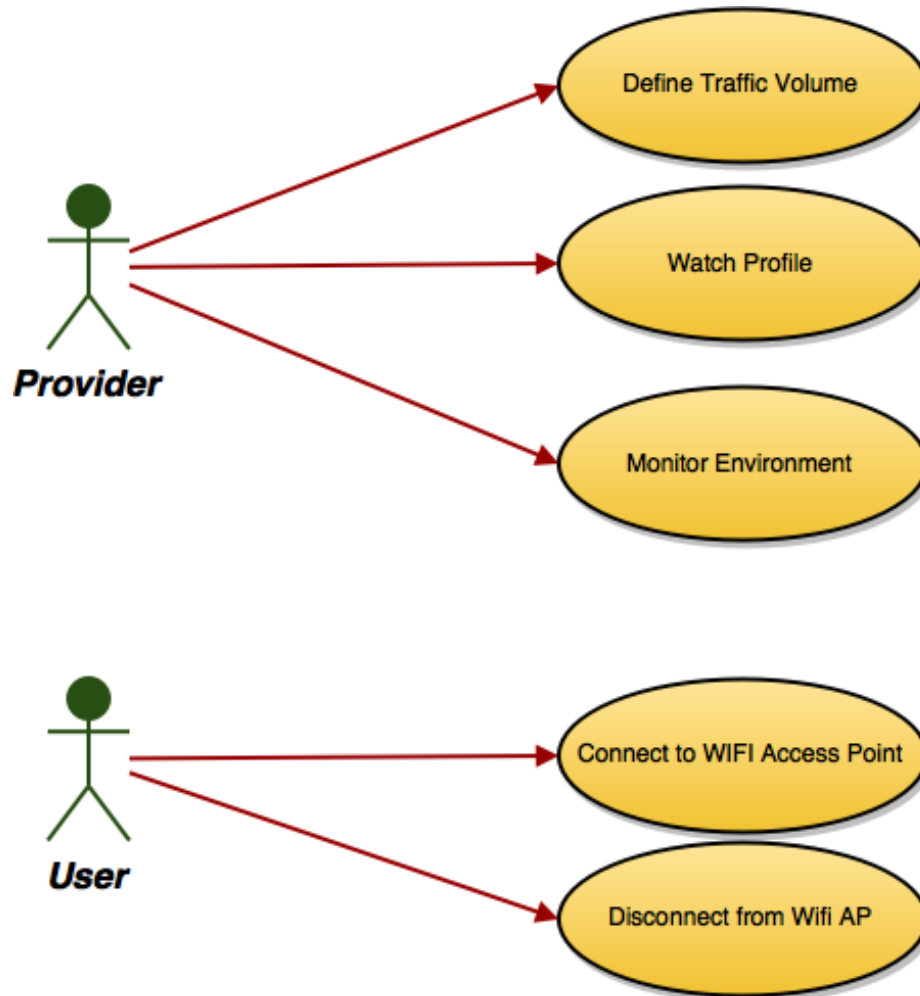- Adaption and Context
- Current state / outlook

TECHNISCHE UNIVERSITÄT DRESDEN

Provide Network Connectivity for foreigners

Avoidance of Roaming

KarmaSurf
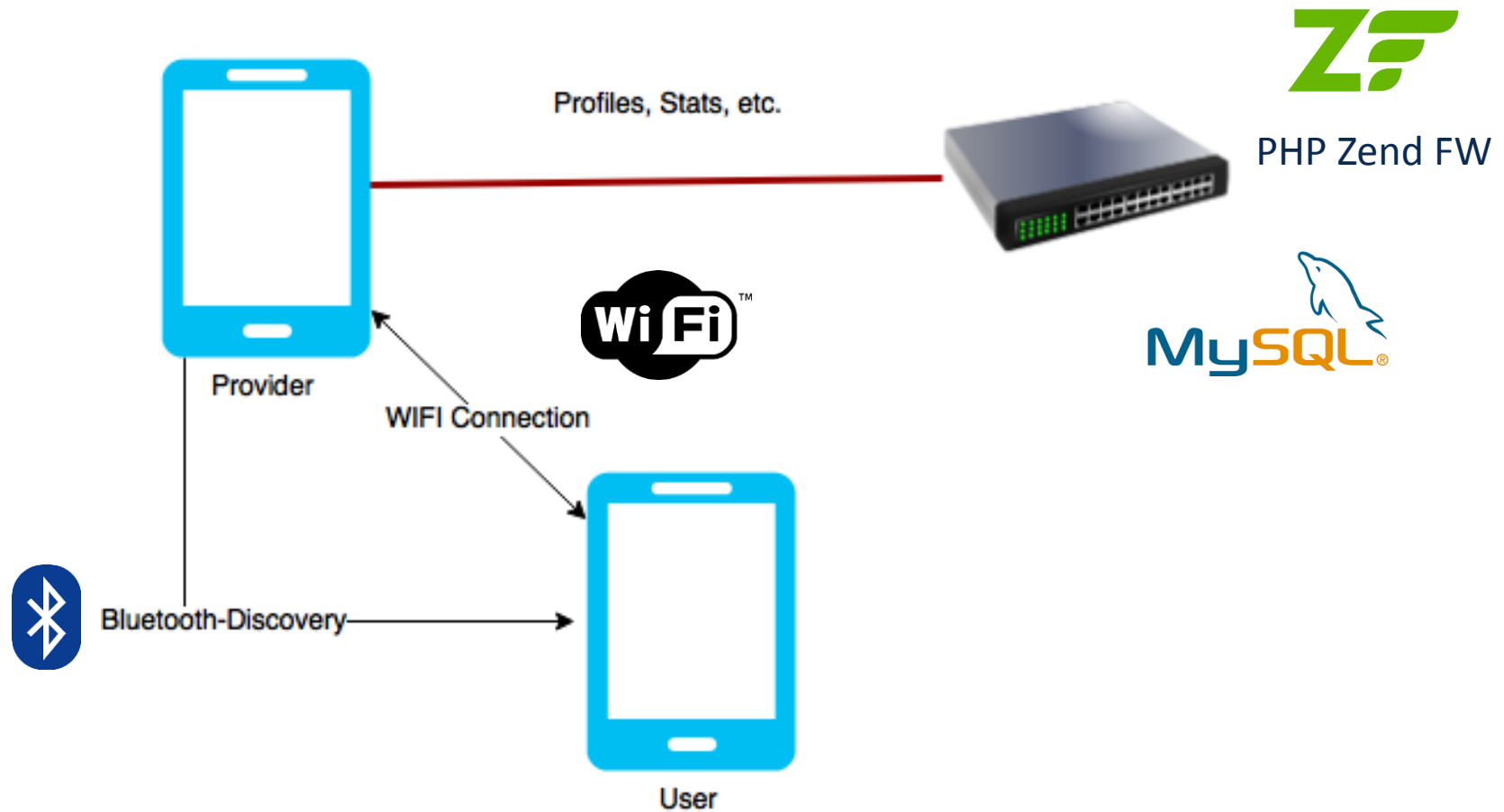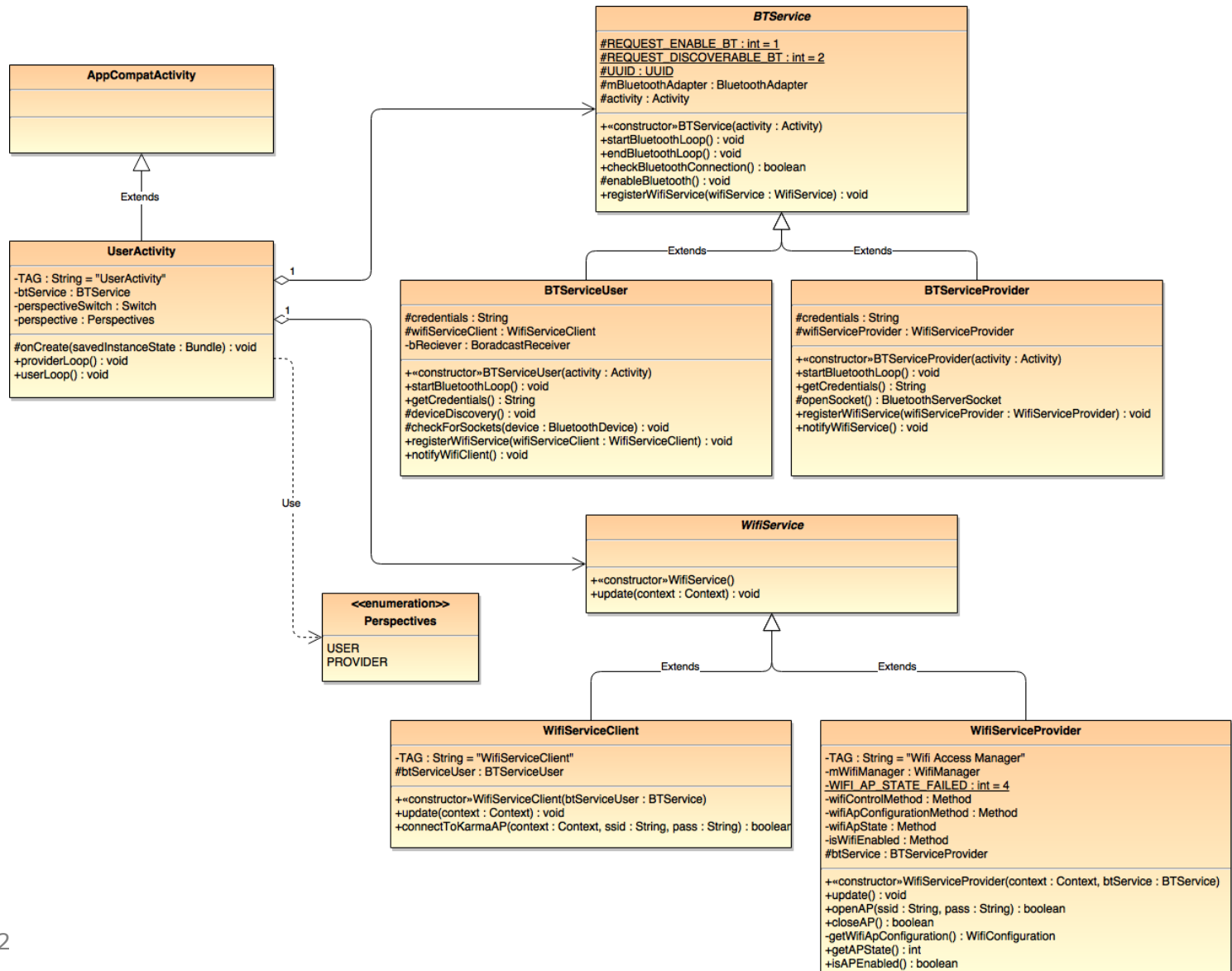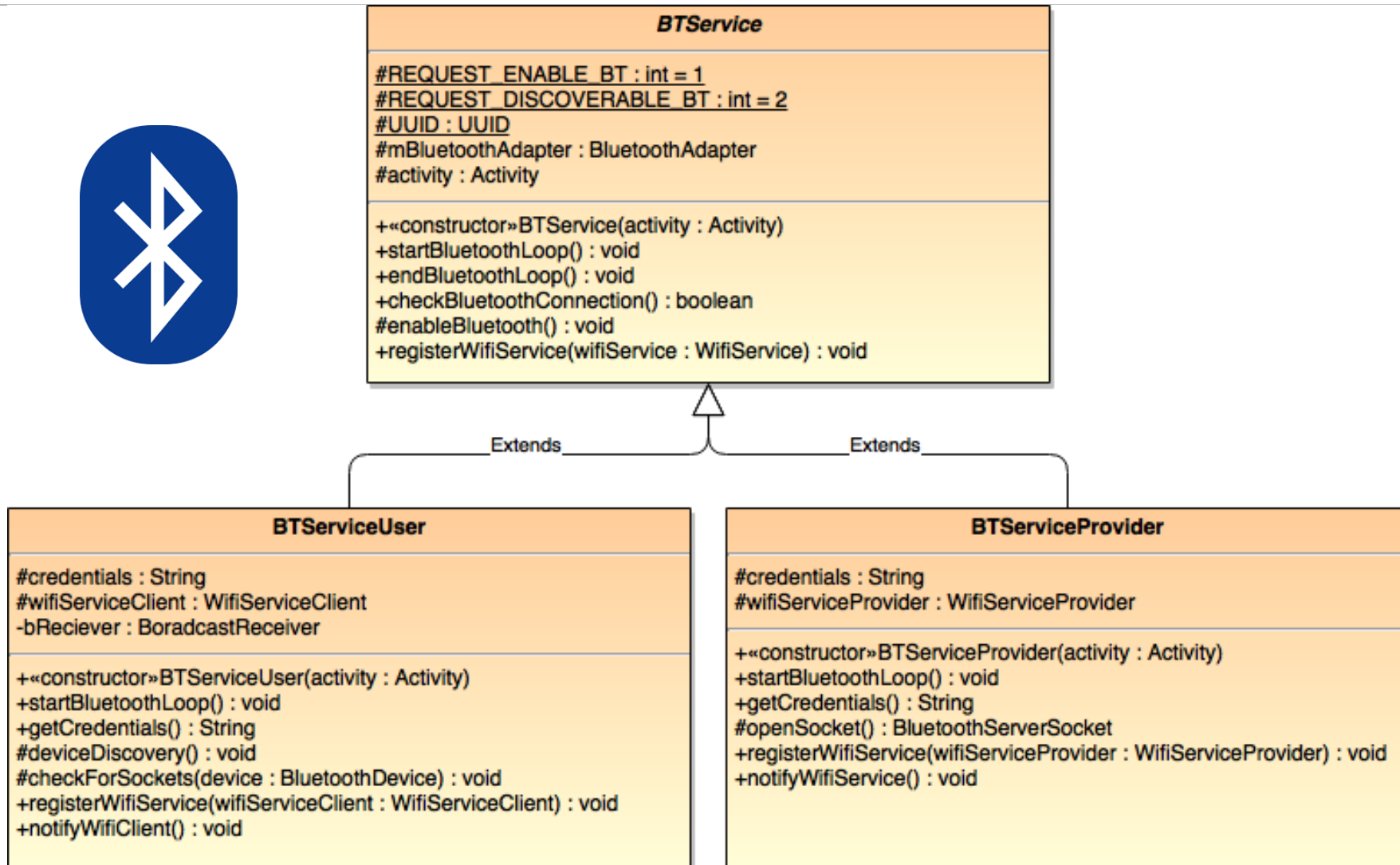
Do something nice with your „overcapacity"

- Gameification: Karma system
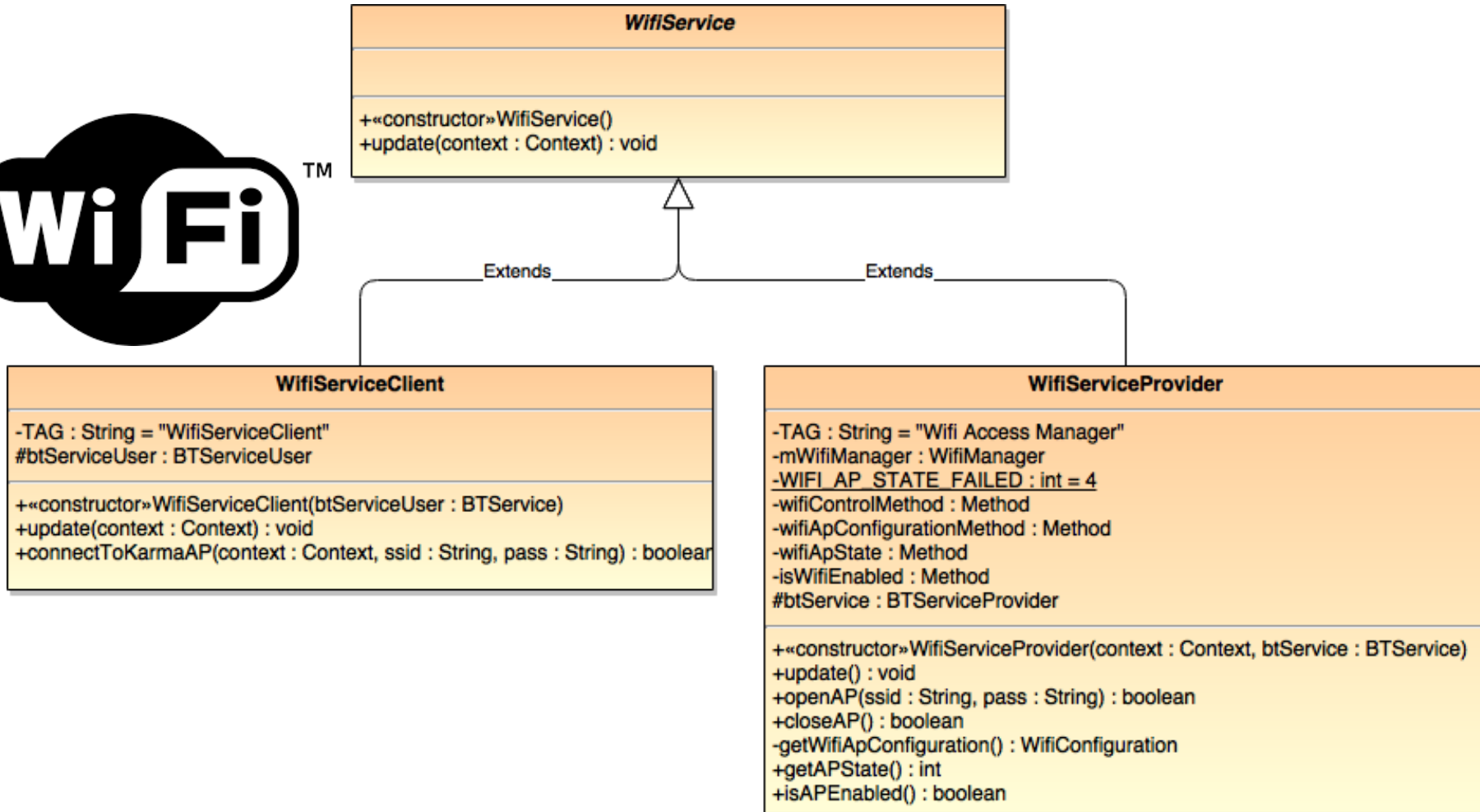- User earn points in dependency of gracious or commited traffic sharing and consumption.

- f.e. avoiding Roaming (using mobile data of natives in foreign countries)
- Using mobile data in trains, festivals or even at Elbwiesen

BTService

#REQUEST_ENABLE_BT : int = 1
#REQUEST_DISCOVERABLE_BT : int = 2
#UUID : UUID
#mBluetoothAdapter : BluetoothAdapter
#activity : Activity

+«constructor»BTService(activity : Activity)
+startBluetoothLoop() : void
+endBluetoothLoop() : void
+checkBluetoothConnection() : boolean
#enableBluetooth() : void
+registerWifiService(wifiService : WifiService) : void

Extends   Extends

BTServiceUser

#credentials : String
#wifiServiceClient : WifiServiceClient
-bReciever : BoradcastReceiver

+«constructor»BTServiceUser(activity : Activity)
+startBluetoothLoop() : void
+getCredentials() : String
#deviceDiscovery() : void
#checkForSockets(device : BluetoothDevice) : void
+registerWifiService(wifiServiceClient : WifiServiceClient) : void
+notifyWifiClient() : void

BTServiceProvider

#credentials : String
#wifiServiceProvider : WifiServiceProvider

+«constructor»BTServiceProvider(activity : Activity)
+startBluetoothLoop() : void
+getCredentials() : String
#openSocket() : BluetoothServerSocket
+registerWifiService(wifiServiceProvider : WifiServiceProvider) : void
+notifyWifiService() : void

WifiService

+«constructor»WifiService()
+update(context : Context) : void

Extends   Extends

WifiServiceClient

-TAG : String = "WifiServiceClient"
#btServiceUser : BTServiceUser

+«constructor»WifiServiceClient(btServiceUser : BTService)
+update(context : Context) : void
+connectToKarmaAP(context : Context, ssid : String, pass : String) : boolean

WifiServiceProvider

-TAG : String = "Wifi Access Manager"
-mWifiManager : WifiManager
-WIFI_AP_STATE_FAILED : int = 4
-wifiControlMethod : Method
-wifiApConfigurationMethod : Method
-wifiApState : Method
-isWifiEnabled : Method
#btService : BTServiceProvider

+«constructor»WifiServiceProvider(context : Context, btService : BTService)
+update() : void
+openAP(ssid : String, pass : String) : boolean
+closeAP() : boolean
-getWifiApConfiguration() : WifiConfiguration
+getAPState() : int
+isAPEnabled() : boolean

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 have to face:

Energy Challenge

- To provide or search for an access point is energy consuming.

- **Our Solution:** Search is done by Bluetooth, whereas the intervall 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.

Challenges that we have to face:

Usability Challenge

- Providing data or connecting to an access point should be possilbe 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.*

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

Context Capturing

- Battery power
- Number of connected Consumers
- Amount of traffic already shared/consumed

Context Abstraction

- Context Information is interpretated and classified in roles:
    - Big Spender, Greedy Consumer, Risky Spender etc.

Context Usage

- The different Roles change the way, how KarmaPoints are calculated:
    - Roles considered as generous/risky/positive are adding more KarmaPoints

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**:
  - KarmaPoints *are transmitted to server only if there is a wifi connection (not a mobile accesspoint) available*: The *data access* of the KarmaPoints-Stats is rescheduled to not spoil shared traffic for that.
  - 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 availiable battery power.

- Core operations working
    - Control role via Switch (Provider vs. Consumer)
    - Discover new devices via Bluetooth
    - Connect without pairing and exchange credentials
    - Enable access point and connect to it

- TBD:
    - Communication with server (Gameification)
        - REST API
    - Transfer discovery mechanisms to a background service
    - Bug fixing / testing