

#### DCAITI-Project: Implementation of a Traffic Light Service on an (Android) Smartphone

Oday Kabha <u>oday.kabha@ipk.fraunhofer.de</u>
Yiyang Song <u>yiyang.song@campus.tu-berlin.de</u>
Yuanheng Mu <u>johanmu1994@mailbox.tu-berlin.de</u>

Supervisor: Birgit Kwella

[ dɪsiː aɪtiː ] Overview

- Background
- Develop Framework
- III. System Architecture
- IV. Tasks
- V. Milestone

### 1. Background

GLOSA (Green Light Optimized Speed Advisory)

suggests speeds to vehicles to pass through an intersection

RSU (Roadside unit)

SPaT (Signal Phase and Time)

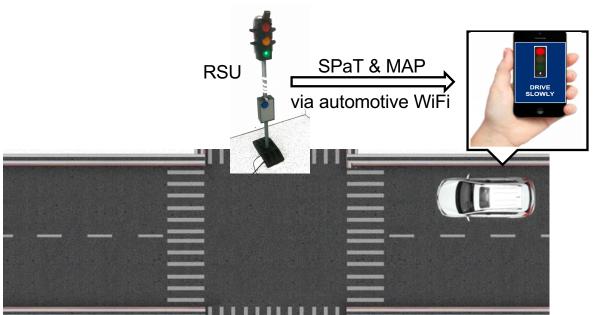
FOKUS Traffic Light Service

MAP (Map data)

Topology of the intersection

#### Road Users

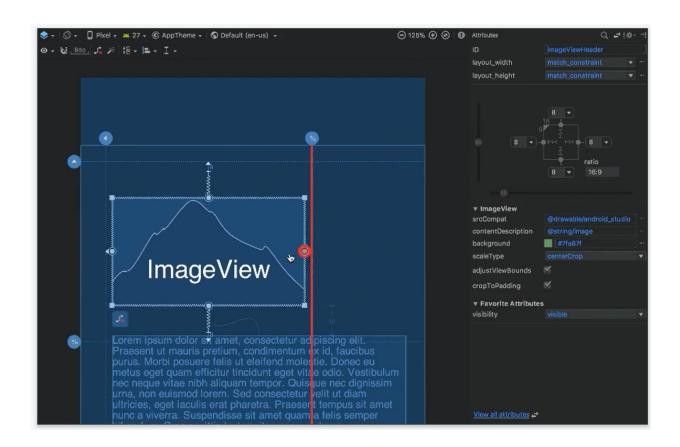
- car/ bicycle/ pedestrian
- receive recommendation for action



## [disi: aiti:]

#### **Android Studio**

- 1. Visual layout editor
- build layouts by dragging UI elements instead of writing layout XML by hand



## [disi: aiti:]

#### **Android Studio**

- 2. Fast emulator
- run apps on the Android Emulator

```
🖊 An... 🗘 😤 🗘 — 🥷 GardenActivityTest.kt 🗵 🥷 GardenPlantingDaoTest.kt 🗵
▼ lim app
                               1 / Copyright 2018 Google LLC .../
                                                                                                                                                               U • • ◊ ◊ ◊ ○ □ ◎ ■
   ▶ ■ manifests
                                          package com.google.samples.apps.sunflower.data
                                                                                                                                                                               1:16 🗊 🕝
    ► Imassets
                                        class GardenPlantingDaoTest {
  ► A Gradle Scripts
                                            private lateinit var <u>database</u>: AppDatabase
private lateinit var <u>gardenPlantingDao</u>: GardenPlantingDao
private var <u>testGardenPlantingId</u>: Long = 0
                                              var instantTaskExecutorRule = InstantTaskExecutorRule()
                                              @Before fun createDb() = runBlocking { this: CoroutineScope
                                                   val context = InstrumentationRegistry.getInstrumentation().targetContext
                                                  database = Room.inMemoryDatabaseBuilder(context, AppDatabase::class.java).build()
                                                   gardenPlantingDao = database.gardenPlantingDao()
                                                   \underline{\texttt{testGardenPlantingId}} = \underline{\texttt{gardenPlantingDag.insertGardenPlanting}}(\underline{\texttt{testGardenPlanting}})
                                                                                                                                                                                          Your garden is empty
                                              @After fun closeDb() {
                                                                                                                                                                                                  ADD PLANT
                                              @Test fun testGetGardenPlantings() = runBlocking { this: CoroutineScope
                                                  val gardenPlanting2 = GardenPlanting(
                                                 ).also { it.gardenPlantingId = 2 }
gardenPlantingDag.insertGardenPlanting(gardenPlanting2)
                                                   assertThat(getValue(gardenPlantingDag.getGardenPlantings()).size, equalTo( operand: 2))
                                             @Test fun testDeleteGardenPlanting() = runBlocking { this: CoroutineScope
                                         Install successfully finished in 212 ms.

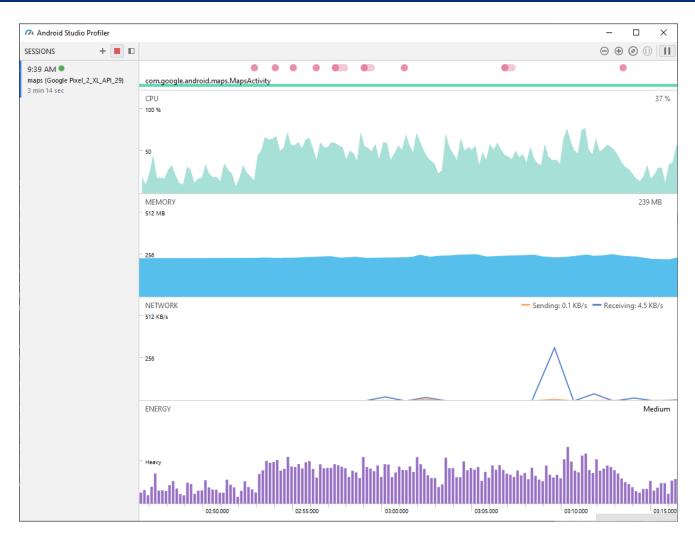
App restart successful without requiring a re-install.
                                                                                                                                                                                                   1:1 LF UTF-8 4 spaces 🔓 😀 😩 👼
```

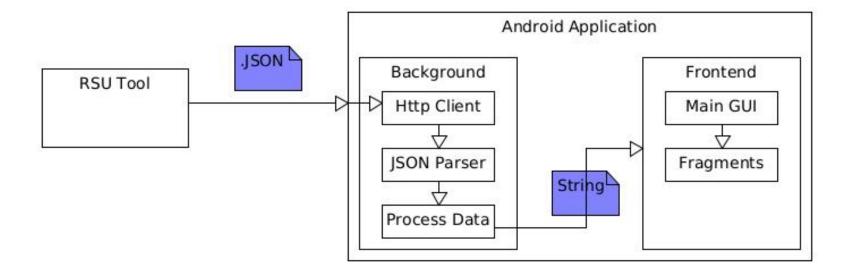
### 2. Develop Framework

# [ dɪsiː aɪtiː ]

### **Android Studio**

- 3. Intelligent code editor
- based on IntelliJ IDEA
- 4. Realtime profilers
- measure app performance





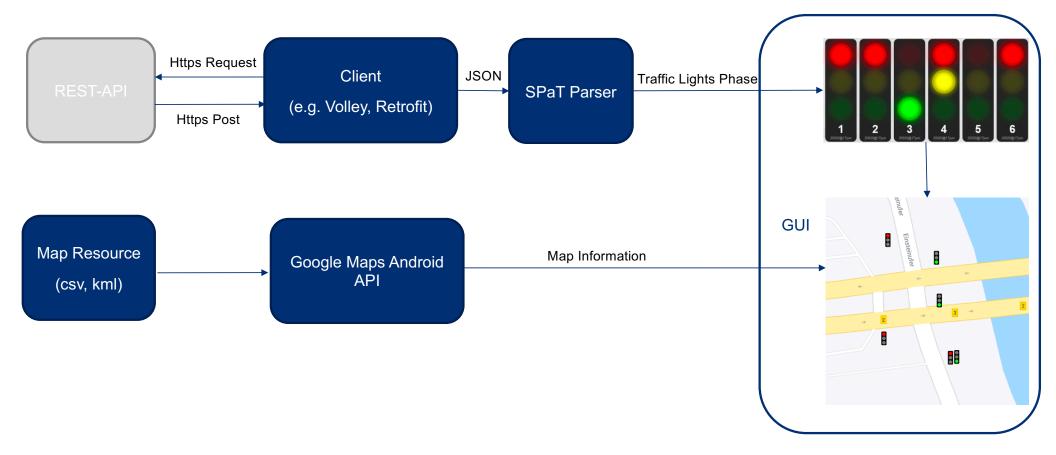
• Request, analysis and visualization of traffic lights information

• Decision of driving strategy

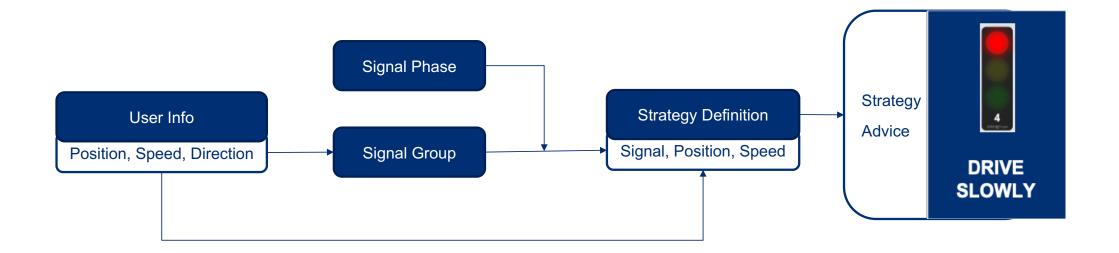
Simulation and evaluation

### 4.1. Request, analysis and visualization of traffic lights information

# [ disi: aiti: ]

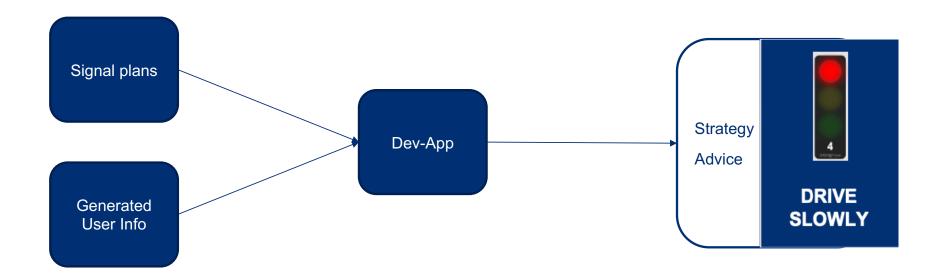


10



18.11.2020 DCAITI -Projekt

Testcases with different **signal plans** of traffic lights, **random users** (different position, direction and speed) and assertion of **driving strategies** 



5. Milestone [dɪsiː aɪtiː]

CW\Tasks	Literature Research	Framework & Libs Test	SPaT Analysis & Visual	Determination of signal group	2nd Pre.	Work about driving strategies	Simulation and Evaluation	3rd Pre.
47								
48	2nd Meeting							
49								
50								
51								
52								
53								
1					6-Jan			
2								
3								
4								
5								
6								10-Feb



### Thanks for your attention!

Oday Kabha Yiyang Song Yuanheng Mu oday.kabha@ipk.fraunhofer.de yiyang.song@campus.tu-berlin.de johanmu1994@mailbox.tu-berlin.de