Architectural Styles for the Development of WoT Applications

Jose Garcia-Alonso
@jmgaralo
Javier Berrocal
@jberolm
Before we start

• https://forms.gle/YvTJscVQGXuRsoU76
Context

- Mobile apps
Index

- Server-Centric vs Mobile-Centric
- Consumption estimations
- Generating Mobile-Centric APPs
SC vs MC

- Server-Centric
SC vs MC

- Mobile-Centric
SC vs MC

- Mobile-Centric
SC vs MC
SC vs MC

HeatMap
HeatMap– Funcionality

• Two main funcionalities
  – Gather users’ positioning
  – Generate heatmap
HeatMap – Implementation

• Three architectures
  – Server-Centric
  – Mobile-Centric
  – Hybrid
HeatMap– SC
HeatMap– MC
HeatMap – Hybrid
HeatMap – Implementation

https://goo.gl/Bnvhno
Consumption estimations
Consumption estimation

• If we analyze mobile apps, we find lots of functionalities created by composing the same primitive operations
Consumption estimation

- If we analyze mobile apps, we find lots of functionalities created by composing the same primitive operations

<table>
<thead>
<tr>
<th>Primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>store(content_size)</td>
</tr>
<tr>
<td>post(content_size)</td>
</tr>
<tr>
<td>get(content_size)</td>
</tr>
<tr>
<td>receivePush()</td>
</tr>
<tr>
<td>getGPS()</td>
</tr>
</tbody>
</table>
Consumption estimation

- If we analyze mobile apps, we find lots of functionalities created by composing the same primitive operations

<table>
<thead>
<tr>
<th>Primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>store(content_size)</td>
</tr>
<tr>
<td>post(content_size)</td>
</tr>
<tr>
<td>get(content_size)</td>
</tr>
<tr>
<td>receivePush()</td>
</tr>
<tr>
<td>getGPS()</td>
</tr>
</tbody>
</table>

- The used operations and its order depend on the specific functionality and the architecture
Consumption estimation

- Conceptual Framework
  - Primitive operations
    \[ op_{i}^{rj} : X_1 \times \cdots \times X_{ki} \to \mathbb{R} \]
  - Different architectures
    \[ A = \{ \text{server - centric, mobile - centric} \} \]
  - Several use cases
    \[ uc_{i}^{rj} : Y_1 \times \cdots \times Y_{ki} \to \mathbb{R} \]
  - Use cases are composed by primitive operations
    \[ uc_{i}^{rj} = \sum_{k=1}^{nop} op_{k}^{rj} \times n_k \]
  - Architectures are composed of use cases
    \[ arch_{i}^{rj} = \sum_{k=1}^{nuc} (uc_{k}^{rj} \times f_k) \]
Consumption estimation

• Emergency Alerts
Consumption estimation

• Emergency Alerts

UC – 1 Send GPS position
UC – 2 Receive alert message
Consumption estimation

- Emergency Alerts

UC - 1 Send GPS position
UC - 2 Receive alert message
Consumption estimation

- Emergency Alerts

getGPS getGPS getGPS getGPS

Server-Centric | Mobile-Centric
Consumption estimation

• Emergency Alerts
Consumption estimation

• Emergency Alerts

Server-Centric  Mobile-Centric
Estimación Consumo

- Emergency Alerts

Server-Centric

\[ UC1 = (\text{getGPS()} + \text{post (16b)}) \times \text{GPSFreq} \]

\[ UC2 = \text{receivePush()} \times \text{PUSHFreq} \]

\[ SC = UC1 + UC2 \]
Consumption estimation

- Emergency Alerts

Server-Centric

Mobile-Centric
Consumption estimation

• Emergency Alerts

Server-Centric

Mobile-Centric
Consumption estimation

- Emergency Alerts

Mobile-Centric

\[ UC1 = \emptyset \]

\[ UC2 = (\text{receivePush()} + \text{getGPS()} + \text{post (16b)}) \times \text{PUSHFreq} \]

\[ MC = UC1 + UC2 \]
Consumption estimation

- Knowing the consumption of each primitive, we could estimate the consumption of mobile applications. Even for different architectures
Consumption estimation

• Knowing the consumption of each primitive, we could estimate the consumption of mobile applications. Even for different architectures

On early stages and without implementing the apps
Consumption estimation

- Knowing the consumption of each primitive, we could estimate the consumption of mobile applications. Even for different architectures

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Size (Bytes)</th>
<th>Battery (μAh)</th>
<th>Data (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>store</td>
<td>16</td>
<td>0,44</td>
<td>0</td>
</tr>
<tr>
<td>post</td>
<td>16</td>
<td>16,83</td>
<td>1067</td>
</tr>
<tr>
<td>get</td>
<td>16</td>
<td>16,29</td>
<td>657</td>
</tr>
<tr>
<td>receivePush</td>
<td>140</td>
<td>18,36</td>
<td>407</td>
</tr>
<tr>
<td>getGPS</td>
<td>n/a</td>
<td>7,20</td>
<td>0</td>
</tr>
</tbody>
</table>

On early stages and without implementing the apps
Consumption estimation

• Simple apps → simple estimation

• Complex apps → not so direct estimation

• If we want to follow the app evolution under different circumstances, estimation gets even more complex

https://api-consumptions.herokuapp.com/
Consumption estimation

• The API takes a JSON describing the app and calculate its consumption
Consumption estimation

• We can see an example

Repository:
https://github.com/jberolm/ICWE19

API:
https://api-consumptions.herokuapp.com/
Consumption estimation

• The API generates a CSV result following this format:

<table>
<thead>
<tr>
<th>Architecture_“arName1”_Battery(μAh)</th>
<th>Var1</th>
<th>Var2</th>
<th>...</th>
<th>VarN</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Architecture_“arName1”_Data(Byte)</th>
<th>Var1</th>
<th>Var2</th>
<th>...</th>
<th>VarN</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CU_“cuName”_Battery(μAh)</th>
<th>Var1</th>
<th>Var2</th>
<th>...</th>
<th>VarN</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
Coffee Break!
Generating Mobile–Centric APPs

Javier Berrocal
@jberolm
OpenAPI

- Almost any application make use of APIs to connect with other applications or with other parts of that application.
OpenAPI

Server-Centric APPs
OpenAPI

Server-Centric APPs
OpenAPI

Server-Centric APPs
Generating MC APPs

Mobile-Centric APPs
Generating MC APPs

Mobile–Centric APPs
Generating MC APPs

Steps:

- Create the APP’s specification with OpenAPI.
- An example can be seen in the following URL
  https://github.com/jberolm/ICWE19

```json
openapi: 3.0.1
info:
  title: Emergency Alerts
  description: This application monitors the users' location in order to send them alerts about possible emergencies in their areas.
  version: '1.0'
  termsOfService: 'https://emergencies.spilab.es/terms'
  contact:
    name: Emergency Alerts
    url: 'https://spilab.es'
    email: info@spilab.es
paths:
  /emergency:
    post:
      tags:
        - Emergencies
      summary: Sends an emergency to devices
```
Generating MC APPs

- Steps:
  - Generate the mobile-centric app following the spec.
    https://openapi-generator-spilab.herokuapp.com/
Generating MC APPs

- Steps:
  - Download the generated APP.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>successful operation</td>
</tr>
<tr>
<td>201</td>
<td>Created</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
</tr>
<tr>
<td>403</td>
<td>Forbidden</td>
</tr>
<tr>
<td>404</td>
<td>Not Found</td>
</tr>
</tbody>
</table>

Example Value:
```
{
  "code": "d408e2be-eda6-4d62-b1ef-d05e2e91a72a",
  "link": "http://localhost:8080/api/gen/download/d408e2be-eda6-4d62-b1ef-d05e2e91a72a"
}
```
Generating MC APPs

• Steps:
  – Create a Firebase project

https://console.firebase.google.com
Generating MC APPs

• Steps:
  – Register app and download the file *google-services.json*

1. Package name → This info is in the file *build.gradle* of the downloaded app.
2. Label *applicationId*

Google-services.json should be stored in the root directory.
Generating MC APPs

• Steps:
  – The file `local.properties` should be created in the APP’s root directory
  – This file should include information about the directory of the Android SDK

```bash
## This file must *NOT* be checked into Version Control Systems,
# as it contains information specific to your local configuration.
#
# Location of the SDK. This is only used by Gradle.
# For customization when using a Version Control System, please read the
# header note.
#Thu Jan 24 11:18:00 CET 2019
sdk.dir=C:\Users\usuario1\AppData\Local\Android\Sdk
```
Generating MC APPs

• Steps:
  – Build and generate the APK.

  \texttt{gradlew assembleDebug}

  • The APK should be located in the folder “build\outputs\apk\debug\”
Calling a MC endpoint
Calling a MC endpoint

- **Steps:**
  - Before calling the endpoint, we need two important data.
    - **Token:** provided by the installed app
    - **Authorization:** obtained from the Firebase Console
Calling a MC endpoint

- **Steps:**
  - (Option 1) using Postman:
    - **Url:** [https://fcm.googleapis.com/fcm/send](https://fcm.googleapis.com/fcm/send)
    - **Headers:**
      - Content-Type: application/json
      - Authorization: key=<obtained in the previous slide>
    - **Body:**

```json
{
  "to": "dEzT8zQ:APA91bHay1pYzcFw0sKB34GZG44D47MYzojGV4PQWQaCaOu3kXB8z1x8K_xRpwCTSn3LeLepUAIIFUe1wQfmoXgQ5mCeJ8NyAgthS3Q2ATYZdtTjo9BIkY05CZRe0Epo0m",
  "data": {
    "resource": "Emergencies",
    "method": "postEmergency",
    "params": {
      "emergency": {
        "id": 1,
        "title": "Earthquake",
        "description": "Earthquake in Japan"
      }
    }
  }
}
```
Calling a MC endpoint

- Steps:
  - (Option 2) Generate a HTML client to call the MC API’s endpoints:
    
Calling a MC endpoint

- **Steps:**
  - (Option 2) Use the generated client:
    - Provide the token, Authorization key and the endpoint parameters.
A favor

https://forms.gle/Ko8DH8grZNhL9eU49
THANK YOU!