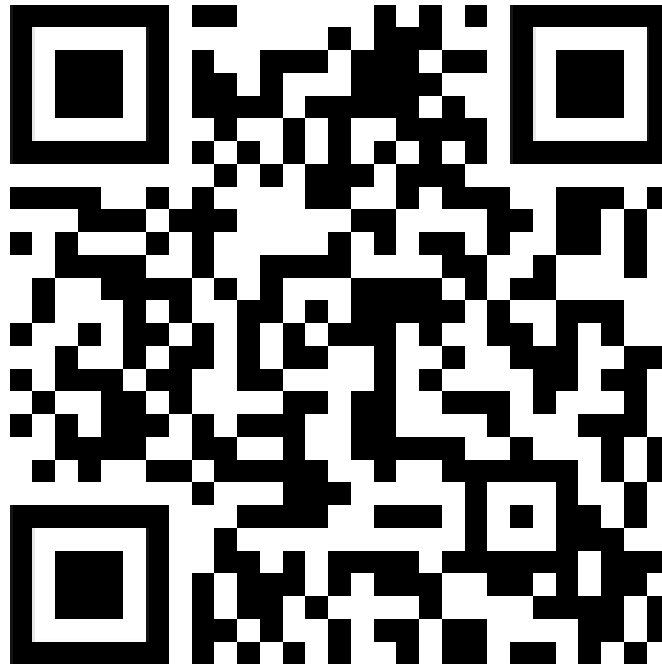


Architectural Styles for the Development of WoT Applications

Jose Garcia-Alonso
@jmgaralo
Javier Berrocal
@jberolm

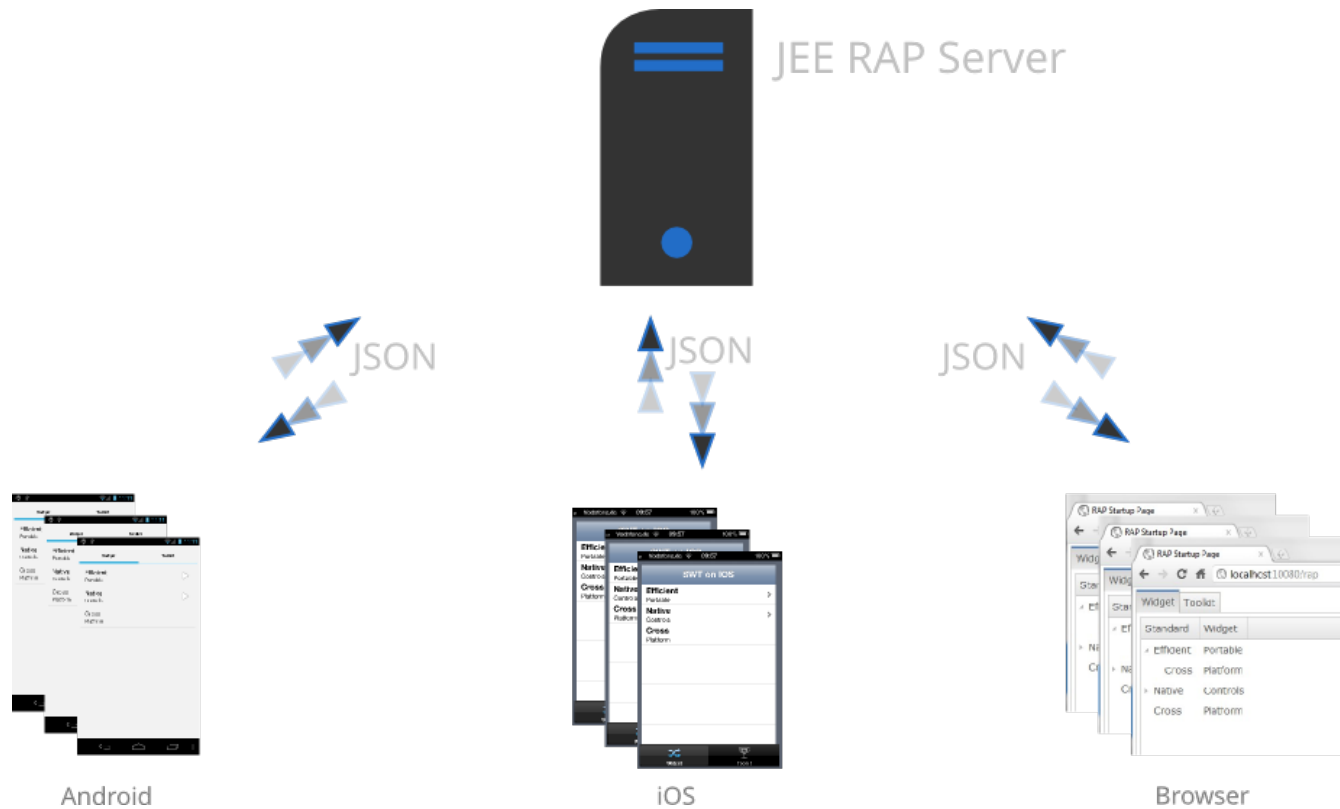
Before we start

- <https://forms.gle/YvTJscVQGXuRsoU76>



Context

- Mobile apps

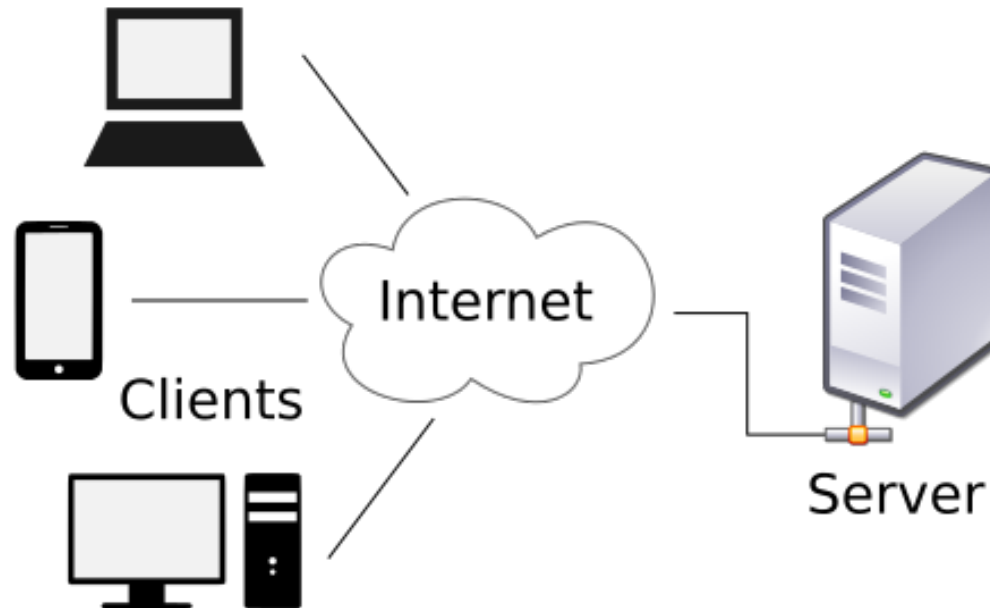


Index

- Server–Centric vs Mobile–Centric
- Consumption estimations
- Generating Mobile–Centric APPs

SC vs MC

- Server-Centric



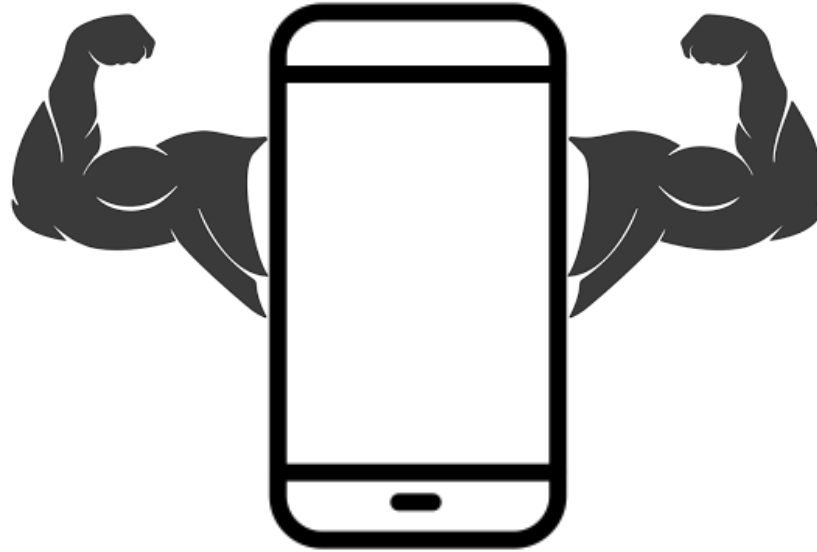
SC vs MC

- Mobile-Centric

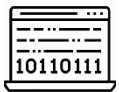
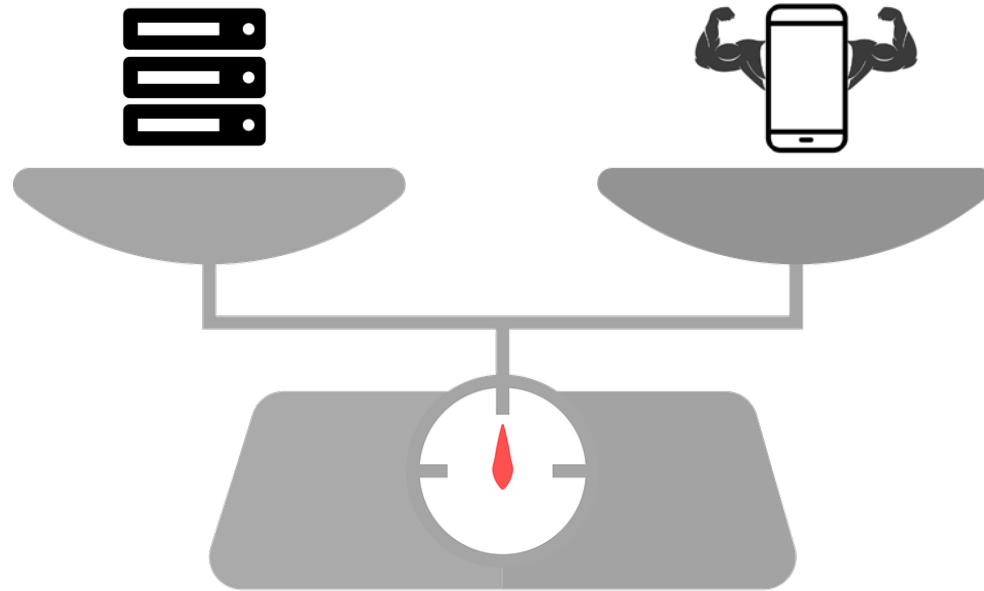


SC vs MC

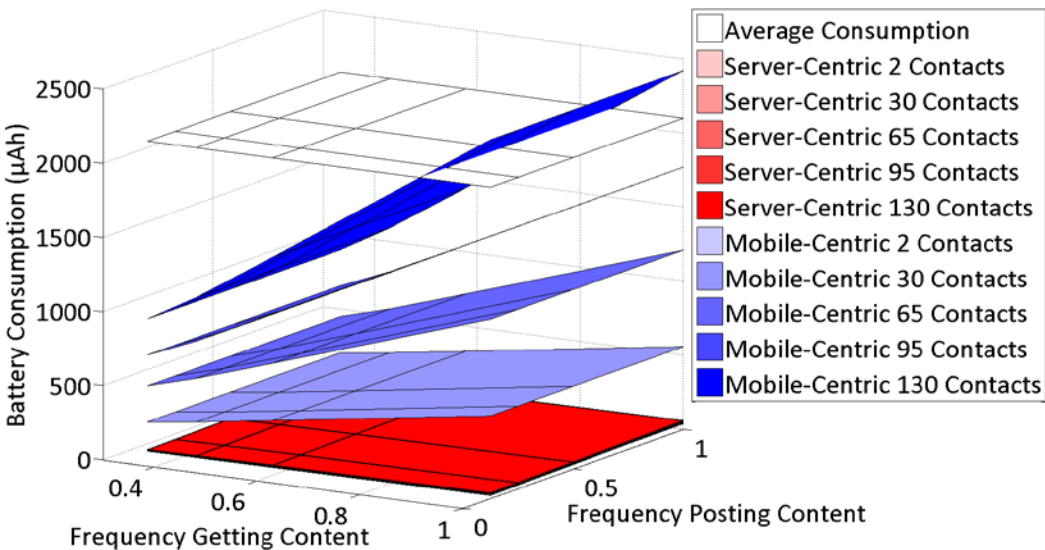
- Mobile-Centric



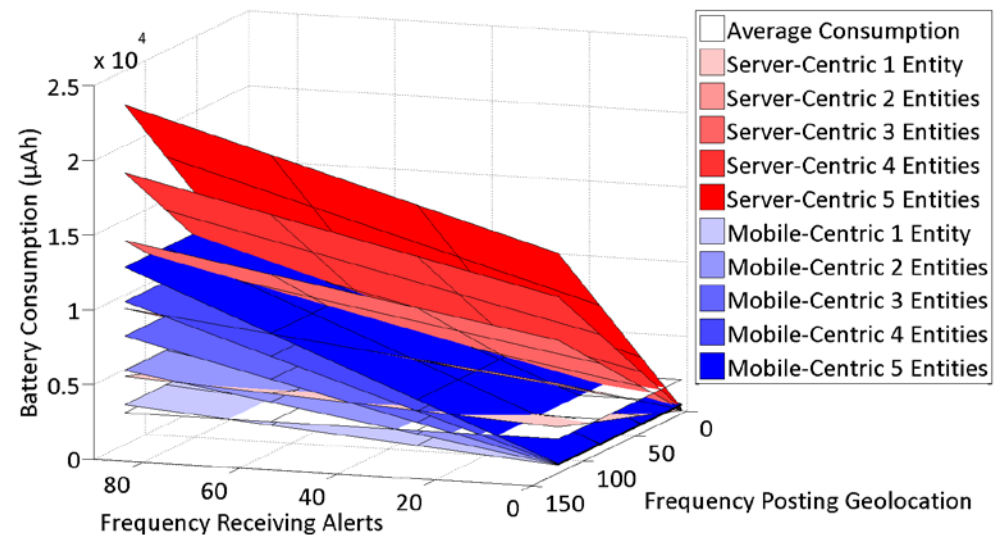
SC vs MC



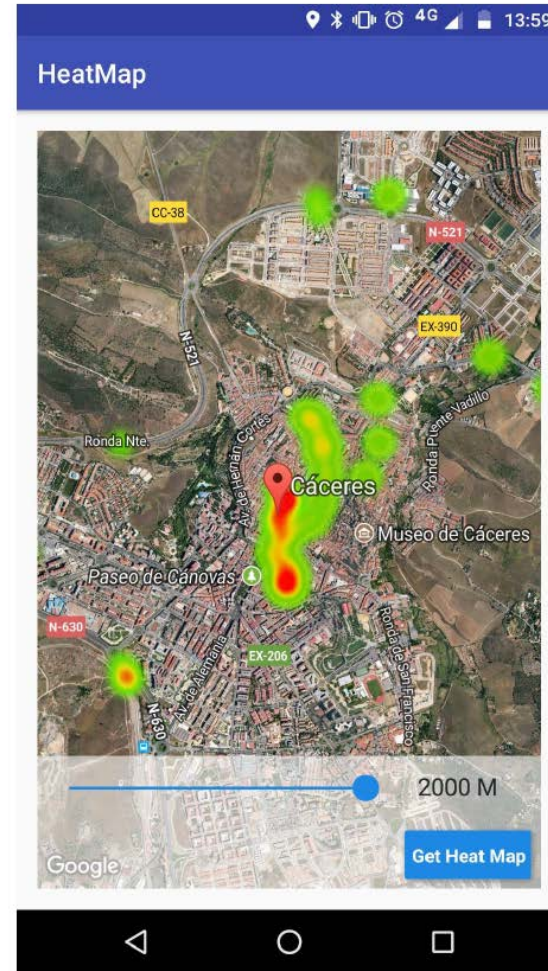
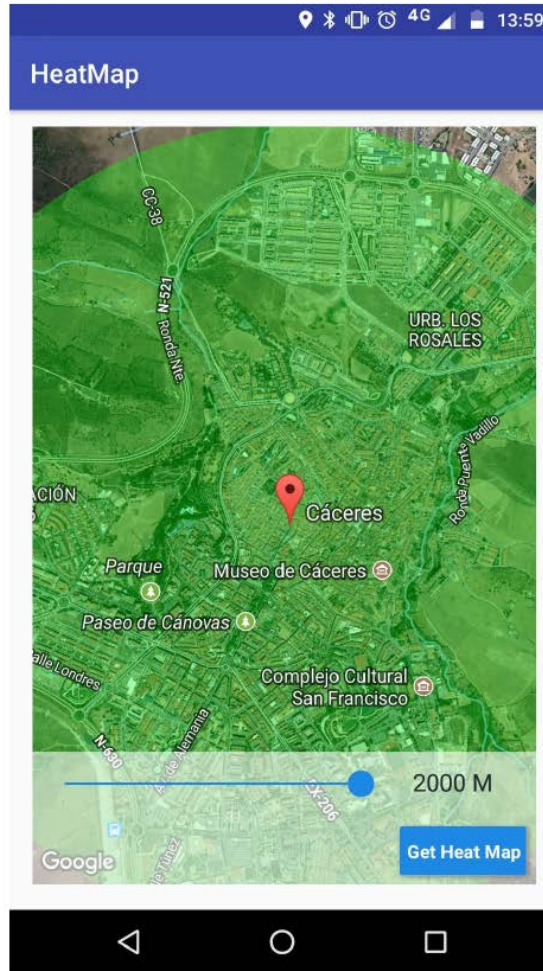
SC vs MC



Javier Berrocal, José García-Alonso, Cristina Vicente-Chicote, Juan Hernández Núñez, Tommi Mikkonen, Carlos Canal, Juan Manuel Murillo: **Early analysis of resource consumption patterns in mobile applications.** Pervasive and Mobile Computing 35: 32-50 (2017)



HeatMap



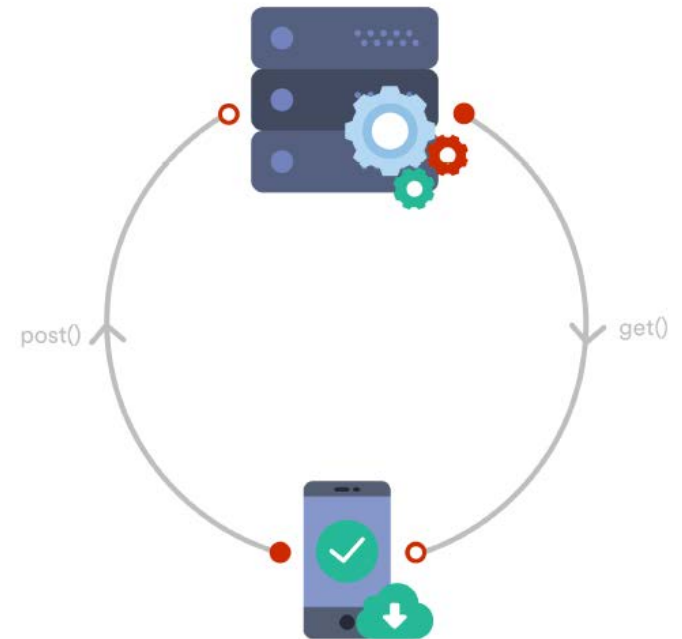
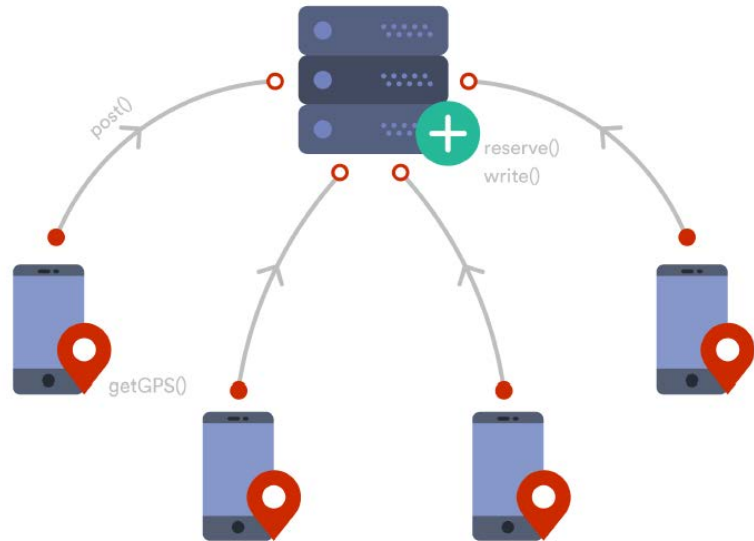
HeatMap– Functionality

- Two main functionalities
 - Gather users' positioning
 - Generate heatmap

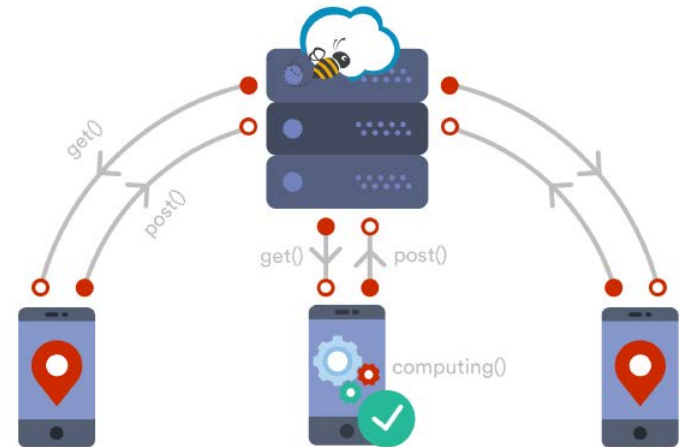
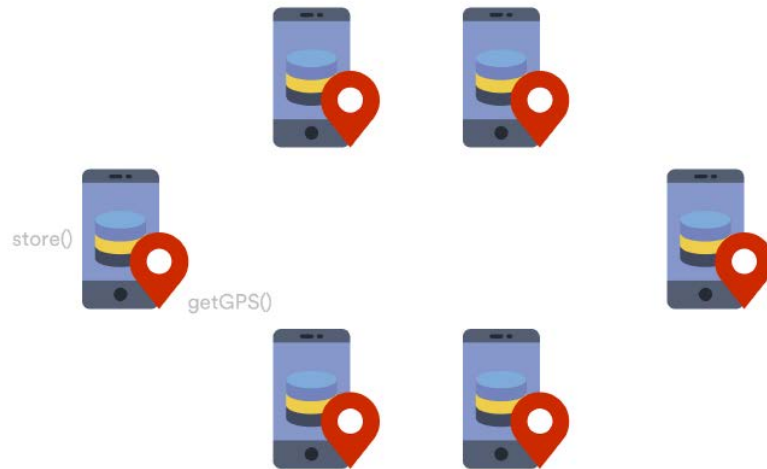
HeatMap– Implementation

- Three architectures
 - Server–Centric
 - Mobile–Centric
 - Hybrid

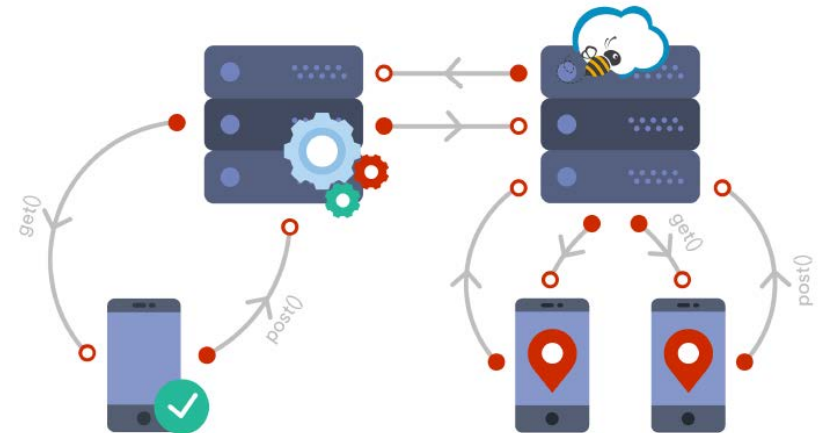
HeatMap- SC



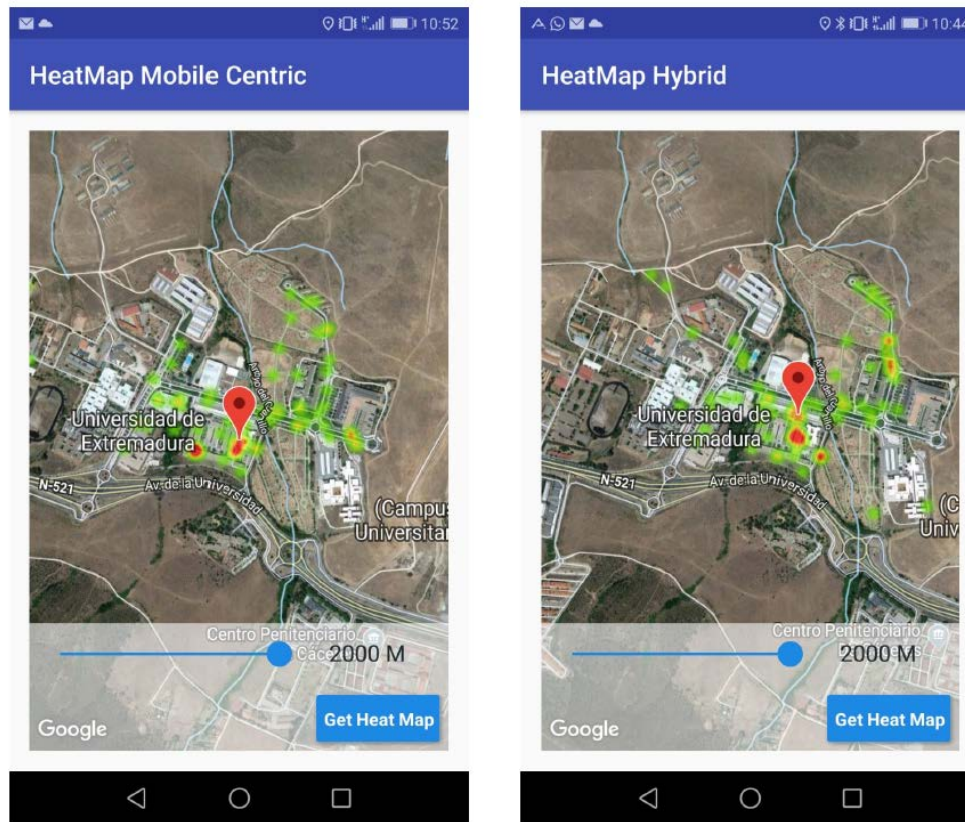
HeatMap- MC



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

Primitive
store(content_size)
post(content_size)
get(content_size)
receivePush()
getGPS()

Consumption estimation

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

Primitive
store(content_size)
post(content_size)
get(content_size)
receivePush()
getGPS()

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

Consumption estimation

- Conceptual Framework

- Primitive operations

$$op_i^{rj} : X_1 \times \cdots \times X_{ki} \rightarrow \mathbb{R}$$

- Different architectures

$$\mathcal{A} = \{server - centric, mobile - centric\}$$

- Several use cases

$$uc_i^{rj} : Y_1 \times \cdots \times Y_{ki} \rightarrow \mathbb{R}$$

- Use cases are composed by primitive operations

$$uc_i^{rj} = \sum_{k=1}^{nop} op_k^{rj} * n_k$$

- Architectures are composed of use cases

$$arch_i^{rj} = \sum_{k=1}^{nuc} (uc_k^{rj} * f_k)$$

Consumption estimation

- Emergency Alerts

Consumption estimation

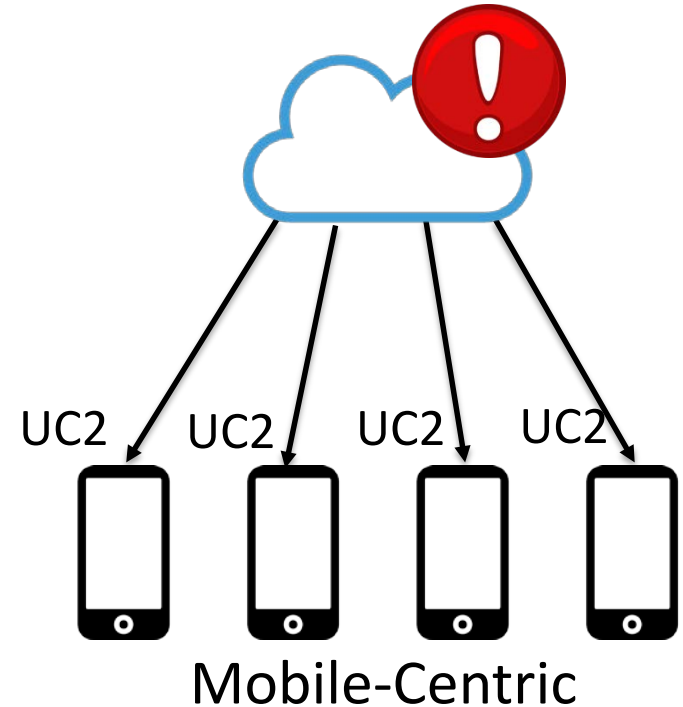
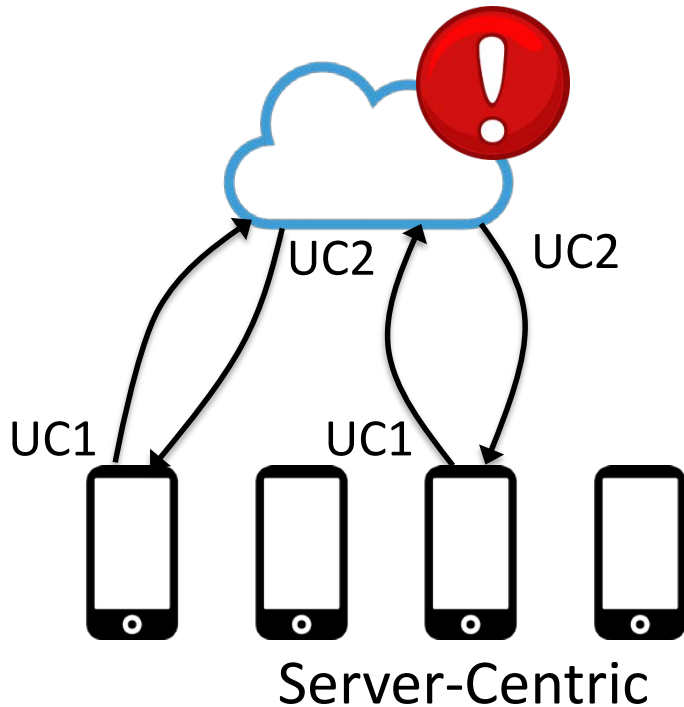
- Emergency Alerts

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



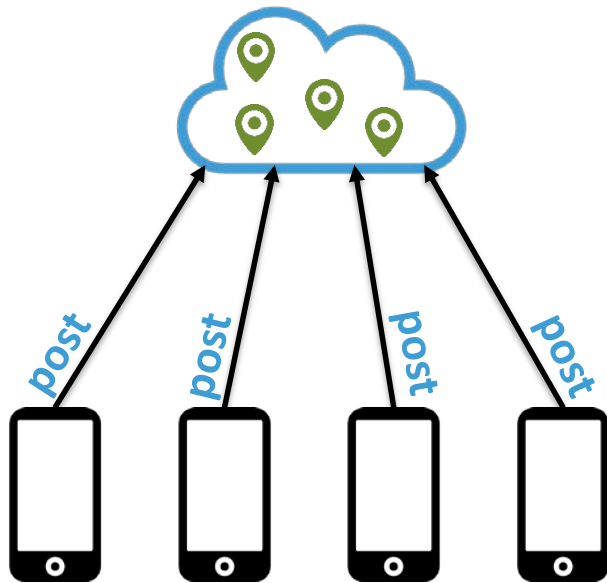
getGPS

Server-Centric

Mobile-Centric

Consumption estimation

- Emergency Alerts

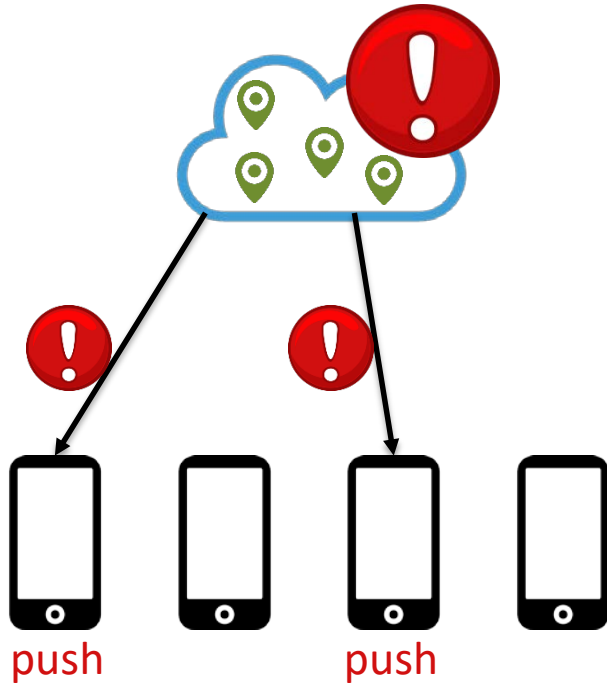


Server-Centric

Mobile-Centric

Consumption estimation

- Emergency Alerts

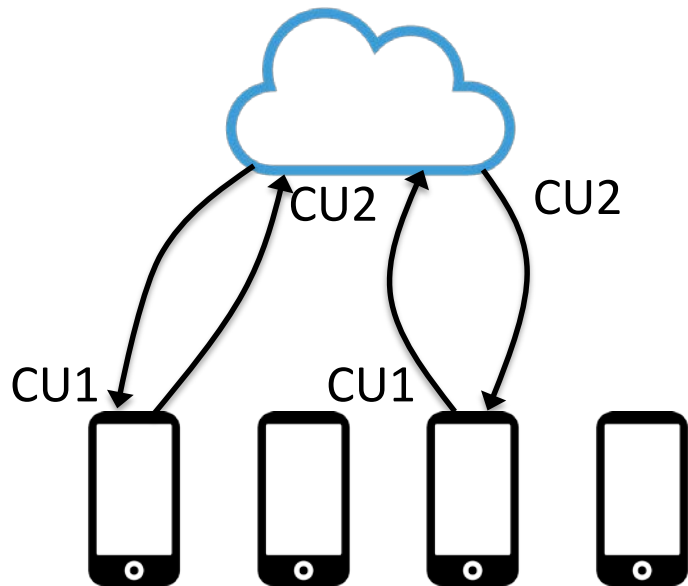


Server-Centric

Mobile-Centric

Estimación Consumo

- Emergency Alerts



Server-Centric

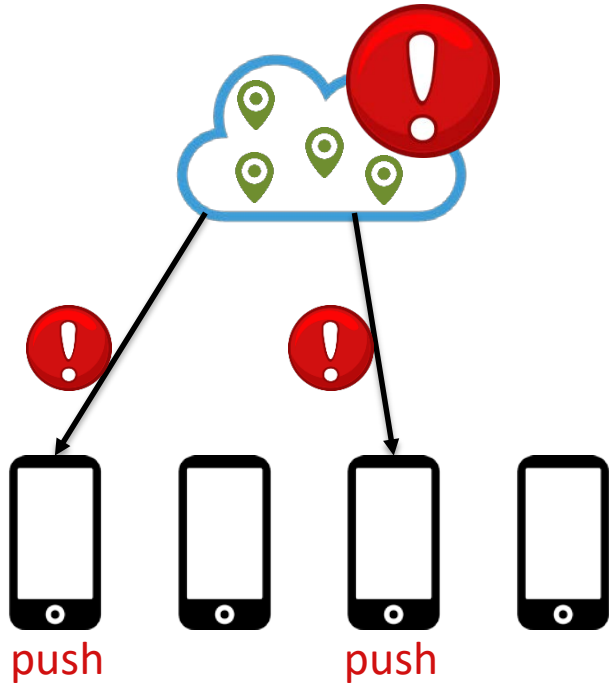
$$UC1 = (\text{getGPS}() + \text{post} (16b)) \times \text{GPSFreq}$$

$$UC2 = \text{receivePush}() \times \text{PUSHFreq}$$

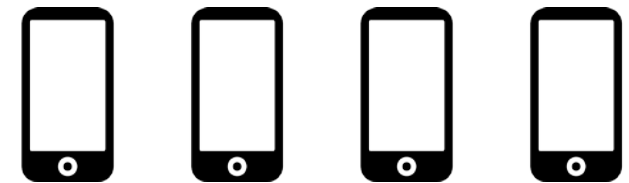
$$SC = UC1 + UC2$$

Consumption estimation

- Emergency Alerts



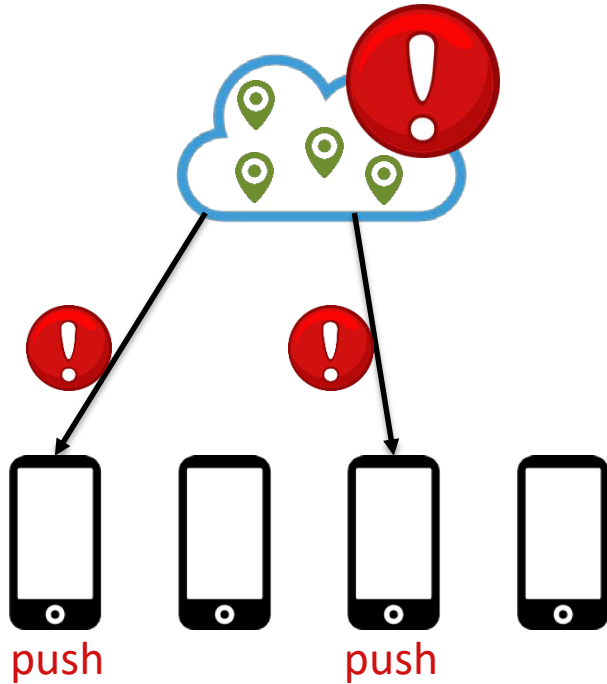
Server-Centric



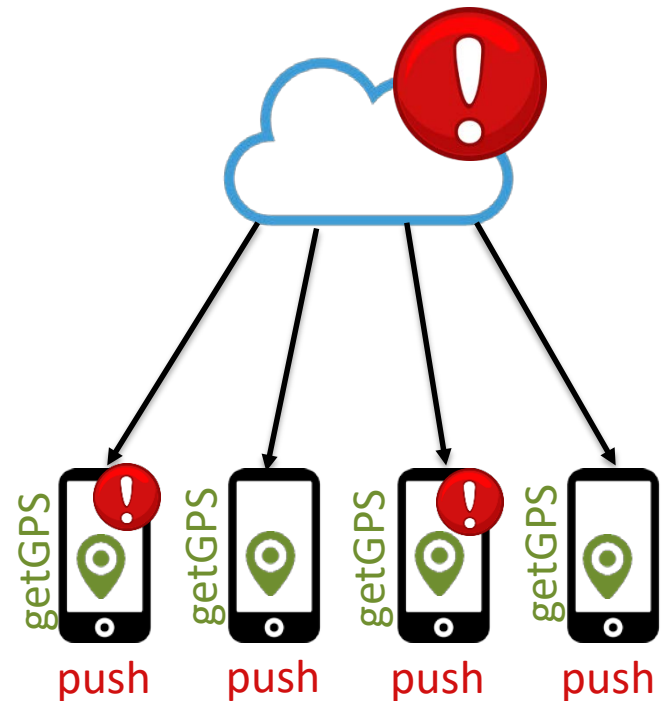
Mobile-Centric

Consumption estimation

- Emergency Alerts



Server-Centric



Mobile-Centric

Consumption estimation

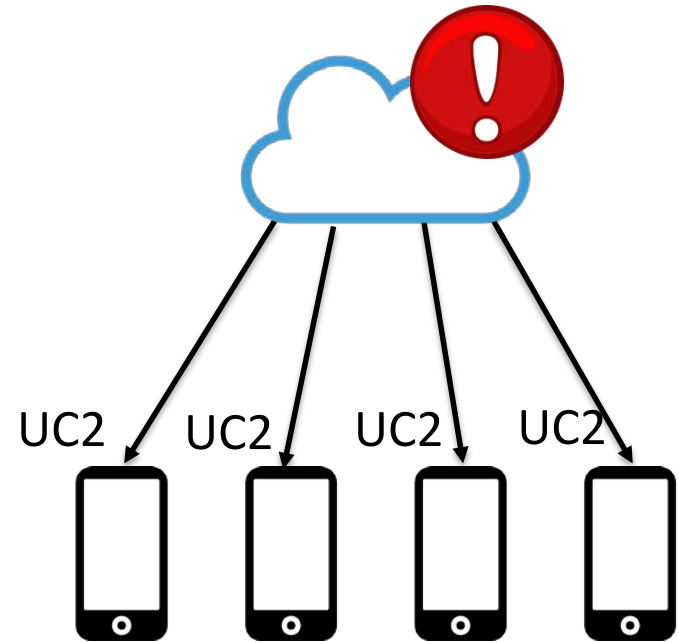
- Emergency Alerts

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

Primitive	Size (Bytes)	Battery (μ Ah)	Data (Bytes)
store	16	0,44	0
post	16	16,83	1067
get	16	16,29	657
receivePush	140	18,36	407
getGPS	n/a	7,20	0

On early stages and without implementing the apps

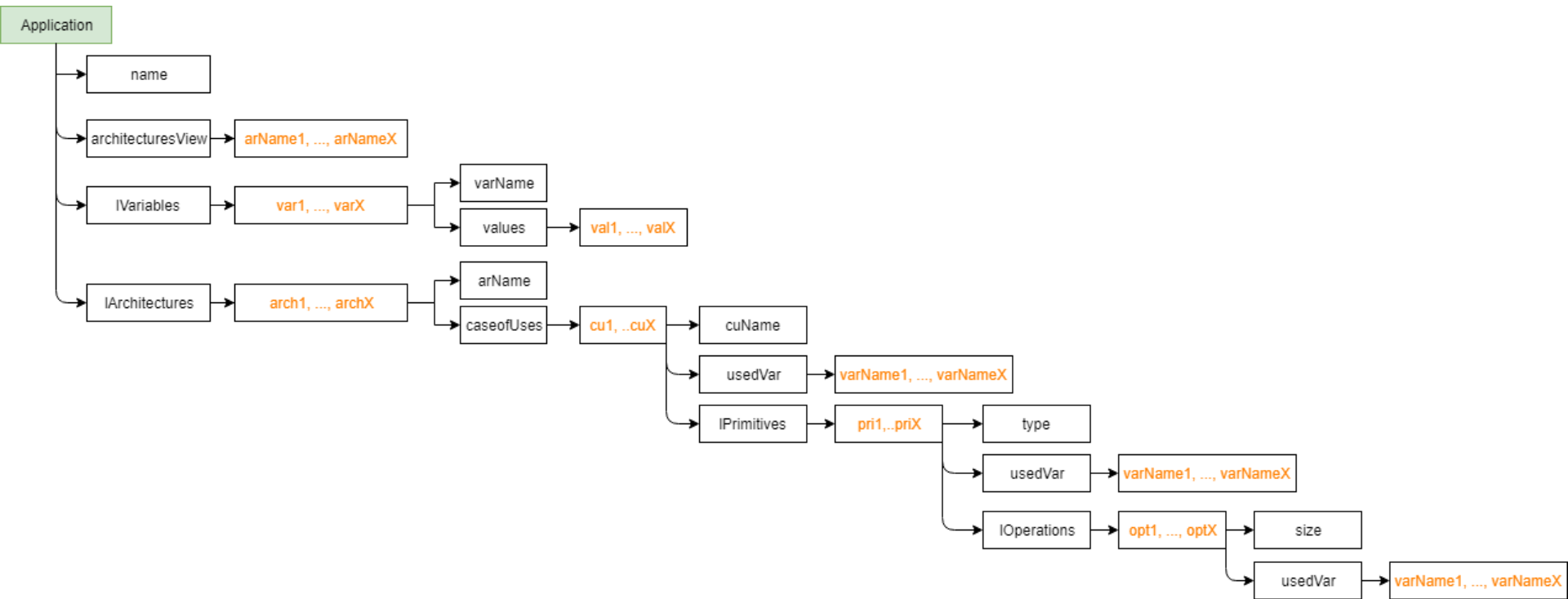
Consumption estimation

- 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:

Architecture_“arName1”_Battery(μAh)				
Var1	Var2	...	VarN	Result
--	--	--	--	--
--	--	--	--	--
Architecture_“arName1”_Data(Byte)				
Var1	Var2	...	VarN	Result
--	--	--	--	--
--	--	--	--	--
CU_“cuName”_Battery(μAh)				
Var1	Var2	...	VarN	Result
--	--	--	--	--
--	--	--	--	--

Coffee Break!

Generating Mobile-Centric APPs

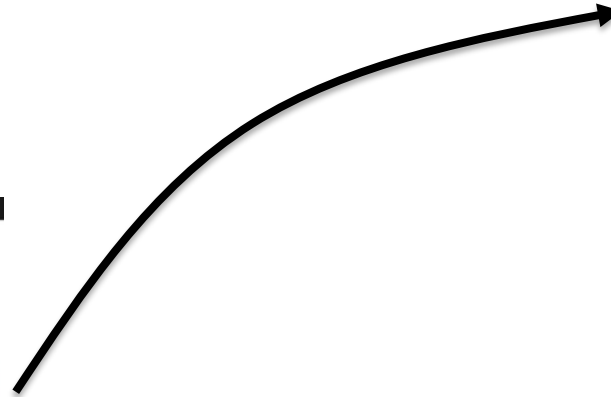
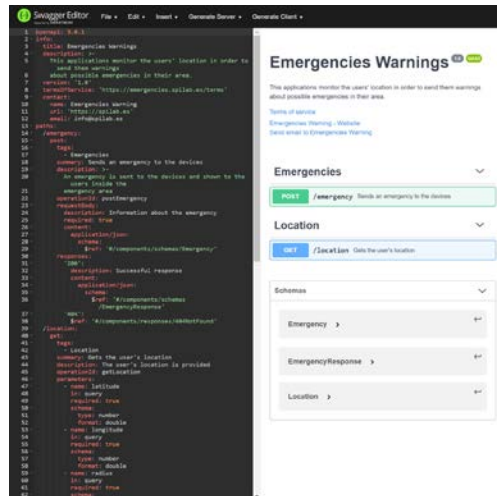
Javier Berrocal
@jberolm

OpenAPI

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

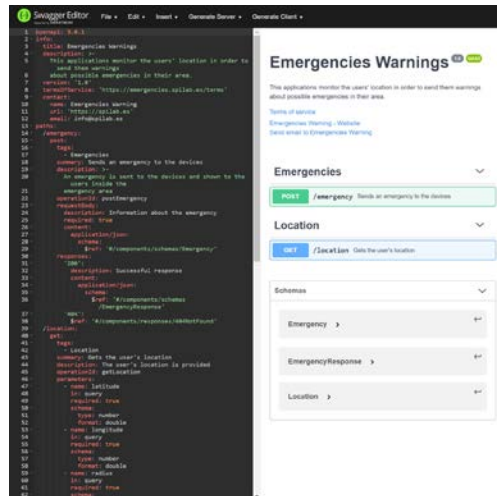


OpenAPI



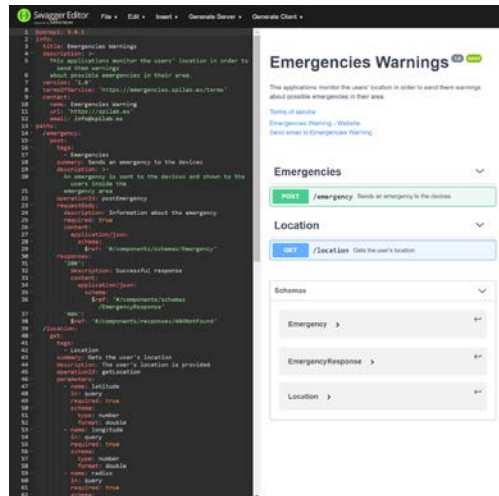
Server-Centric APPs

OpenAPI



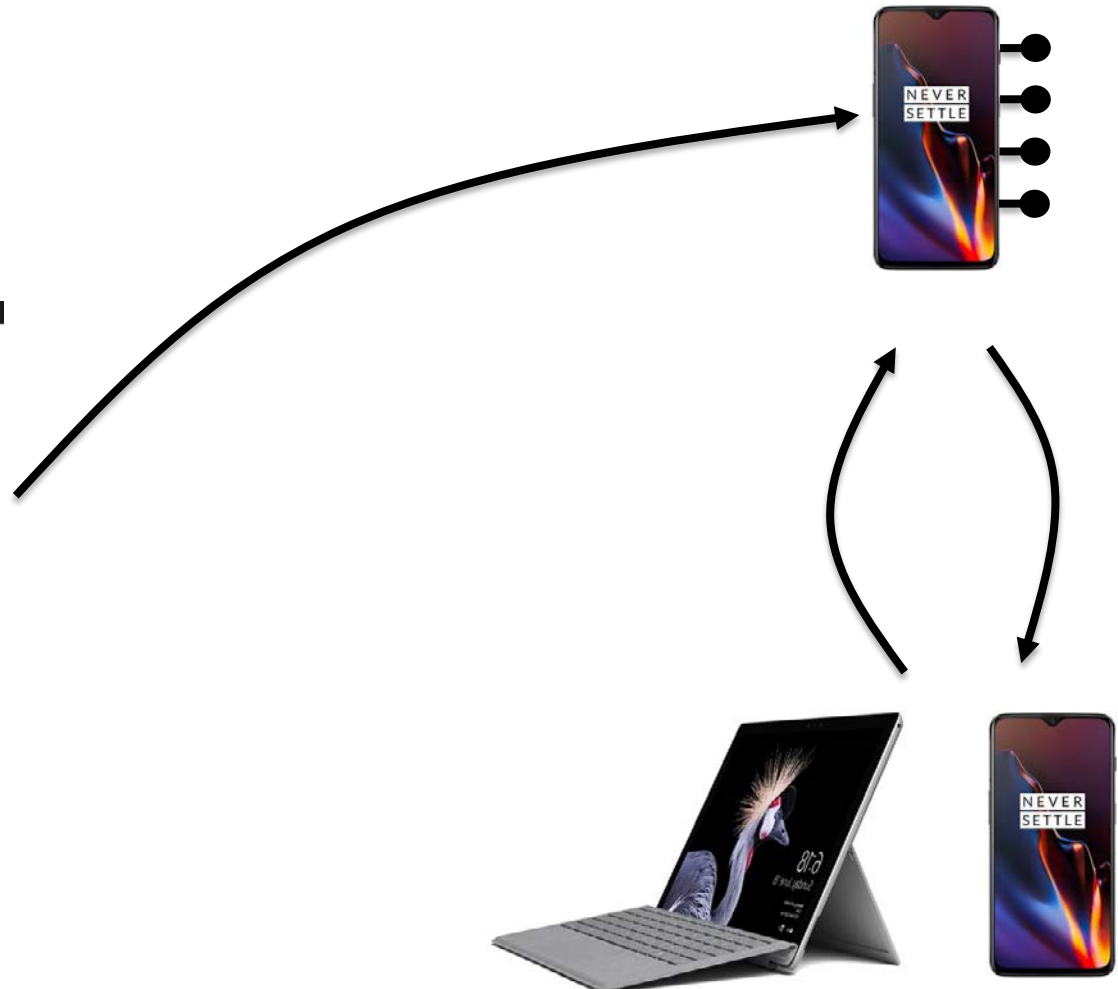
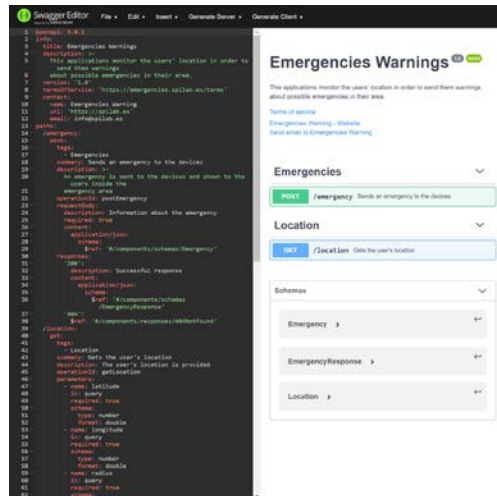
Server-Centric APPs

OpenAPI



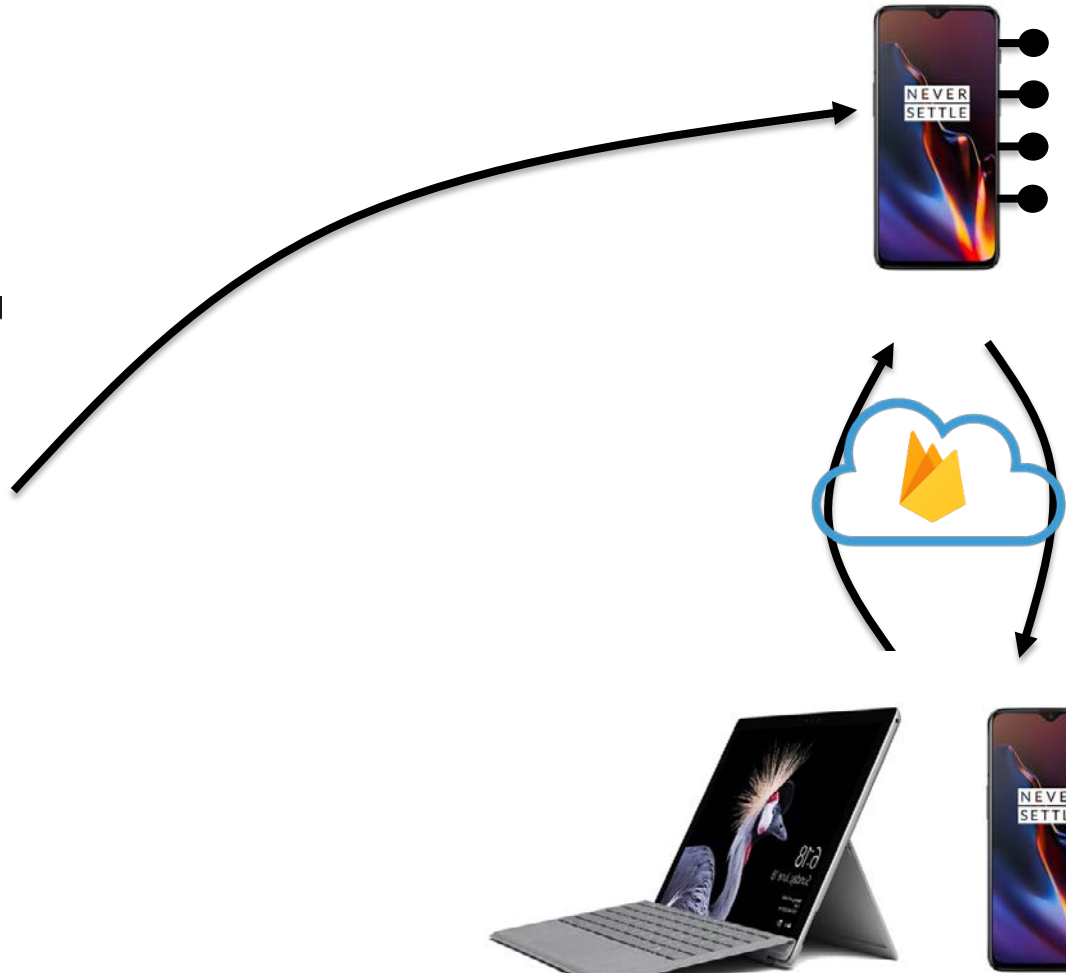
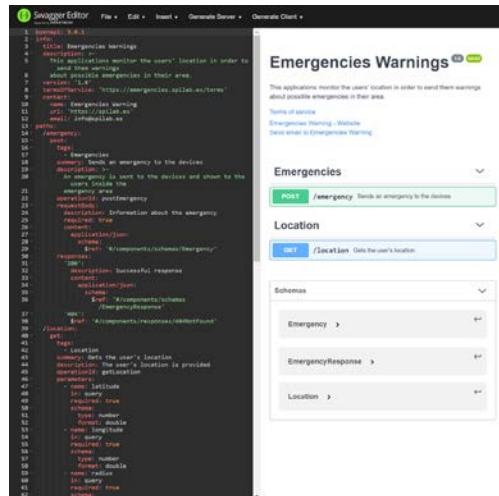
Server-Centric APPs

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>

125 lines (124 sloc) 3.03 KB

Raw

Blame

History



```
1  openapi: 3.0.1
2  info:
3    title: Emergency Alerts
4    description: This application monitors the users' location in order to send them alerts about possible emergencies in their areas.
5    version: '1.0'
6    termsOfService: 'https://emergencies.spilab.es/terms'
7    contact:
8      name: Emergency Alerts
9      url: 'https://spilab.es'
10     email: info@spilab.es
11  paths:
12    /emergency:
13      post:
14        tags:
15          - Emergencies
16        summary: Sends an emergency to devices
```

Generating MC APPs

- Steps:
 - Generate the mobile-centric app following the spec.

<https://openapi-generator-spilab.herokuapp.com/>

POST

/api/gen/servers/{framework} Generates a server library

Accepts a GeneratorInput options map for spec location and generation options.

Parameters

Cancel

Name	Description
framework * required string (path)	framework <div>android-server</div>
generatorInput * required (body)	parameters Example Value Model <div><pre>{ "openAPIUrl": "https://raw.githubusercontent.com/jberolm/ICWE19/master/EmergencyAlerts.yaml", }</pre></div> <div>Cancel</div> <div>Parameter content type application/json</div>

Execute

Generating MC APPs

- Steps:
 - Download the generated APP.

Responses		Response content type
		/
Code	Description	
200	<i>successful operation</i>	
	Example Value Model	
	<pre>{ "code": "d40029be-eda6-4d62-b1ef-d05e2e91a72a", "link": "http://localhost:8080/api/gen/download/d40029be-eda6-4d62-b1ef-d05e2e91a72a" }</pre>	
201	<i>Created</i>	
401	<i>Unauthorized</i>	
403	<i>Forbidden</i>	
404	<i>Not Found</i>	

Generating MC APPs

- Steps:
 - Create a Firebase project

<https://console.firebase.google.com>

1

Welcome to Firebase!

Tools from Google for developing great apps, engaging with your users and earning more through mobile ads.

[Learn more](#) [Documentation](#) [Support](#)

Recent projects



Add project



Explore a demo project

EmergencyAlerts
emergencyalerts-c6d84

2

Add a project

Project name

My awesome project

+ iOS + </>

Tip: Projects span apps across platforms

Project ID

my-awesome-project-id

Locations

United States (Analytics)

nam5 (us-central) (Cloud Firestore)

☒ Use the default settings for sharing Google Analytics for Firebase data

- ✓ Share your Analytics data with all Firebase features
- ✓ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable technical support
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google Account Specialists

☐ I accept the [controller-controller terms](#). This is required when sharing Analytics data to improve Google Products and Services. [Learn more](#)

Cancel

Create project


Generating MC APPs


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

1

Project

Project name

Emergency 

Project ID 

emergency-9521e

Cloud Firestore location


nam5 (us-central)


Web API Key


AlzaSyBogg6PE0xJpv44EZ4hM2Xb_nq1M_Jc4Eo


Public settings

These settings control instances of your project shown to the public:

Public-facing name 

project-223484223610 





Support email 

Not configured 

Your apps

There are no apps in your project

Select a platform to get started



2

Add Firebase to your Android app

[illegible]

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

- 2 Download config file
- 3 Add Firebase SDK
- 4 Run your app to verify installation

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

```
## 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.
.\gradlew assembleDebug
- The APK should be located in the folder “build\outputs\apk\debug\”

```
PS C:\Users\usuariol\Desktop\android-server-server> ./gradlew assembleDebug

> Configure project :
Could not find google-services.json while looking in [src/nuullnull/debug, src/debug/nuullnull, src/nuullnull, src/debug, src/nuullnullDeb
ug]
registerResGeneratingTask is deprecated, use registerGeneratedResFolders(FileCollection)
Could not find google-services.json while looking in [src/nuullnull/release, src/release/nuullnull, src/nuullnull, src/release, src/nuulln
ullRelease]
registerResGeneratingTask is deprecated, use registerGeneratedResFolders(FileCollection)

> Task :processDebugGoogleServices
Parsing json file: C:\Users\usuariol\Desktop\android-server-server\google-services.json

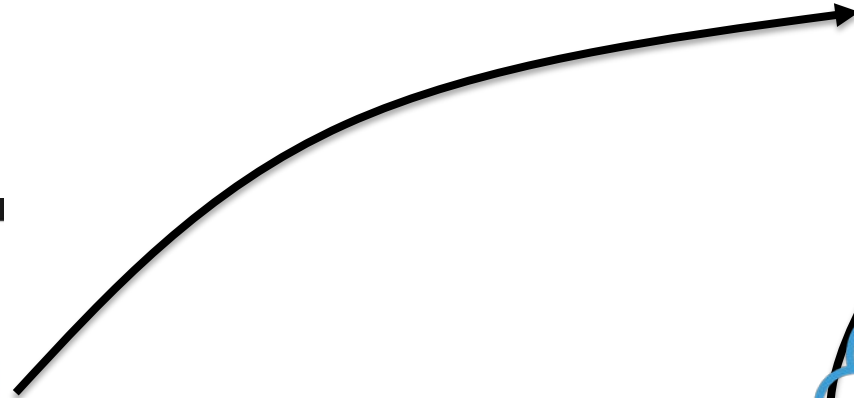
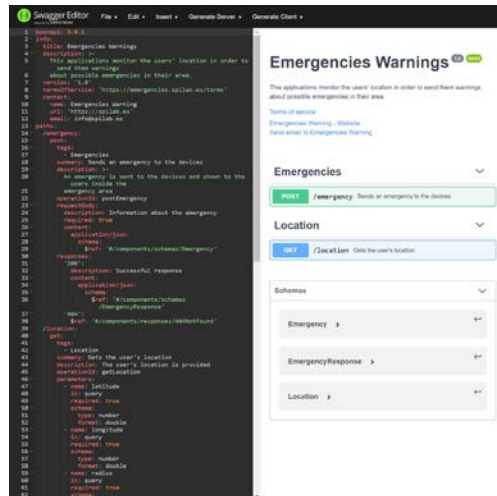
> Task :compileDebugJavaWithJavac
Gradle may disable incremental compilation as the following annotation processors are not incremental: compiler-1.0.0.jar (android.arc
h.persistence,room:compiler:1.0.0).
Consider setting the experimental feature flag android.enableSeparateAnnotationProcessing=true in the gradle.properties file to run an
notation processing in a separate task and make compilation incremental.
C:\Users\usuariol\Desktop\android-server-server\src\main\java\org\openapitools\server\database\NotificationDatabase.java:10: warning:
Schema export directory is not provided to the annotation processor so we cannot export the schema. You can either provide 'room.schem
aLocation' annotation processor argument OR set exportsSchema to false.
public abstract class NotificationDatabase extends RoomDatabase {
    ^
Note: C:\Users\usuariol\Desktop\android-server-server\build\generated\source\apt\debug\org\openapitools\server\database\NotificationDA
O_Impl.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
1 warning

BUILD SUCCESSFUL in 13s
27 actionable tasks: 15 executed, 12 up-to-date
PS C:\Users\usuariol\Desktop\android-server-server>
```



<https://github.com/jberolm/ICWE19>

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

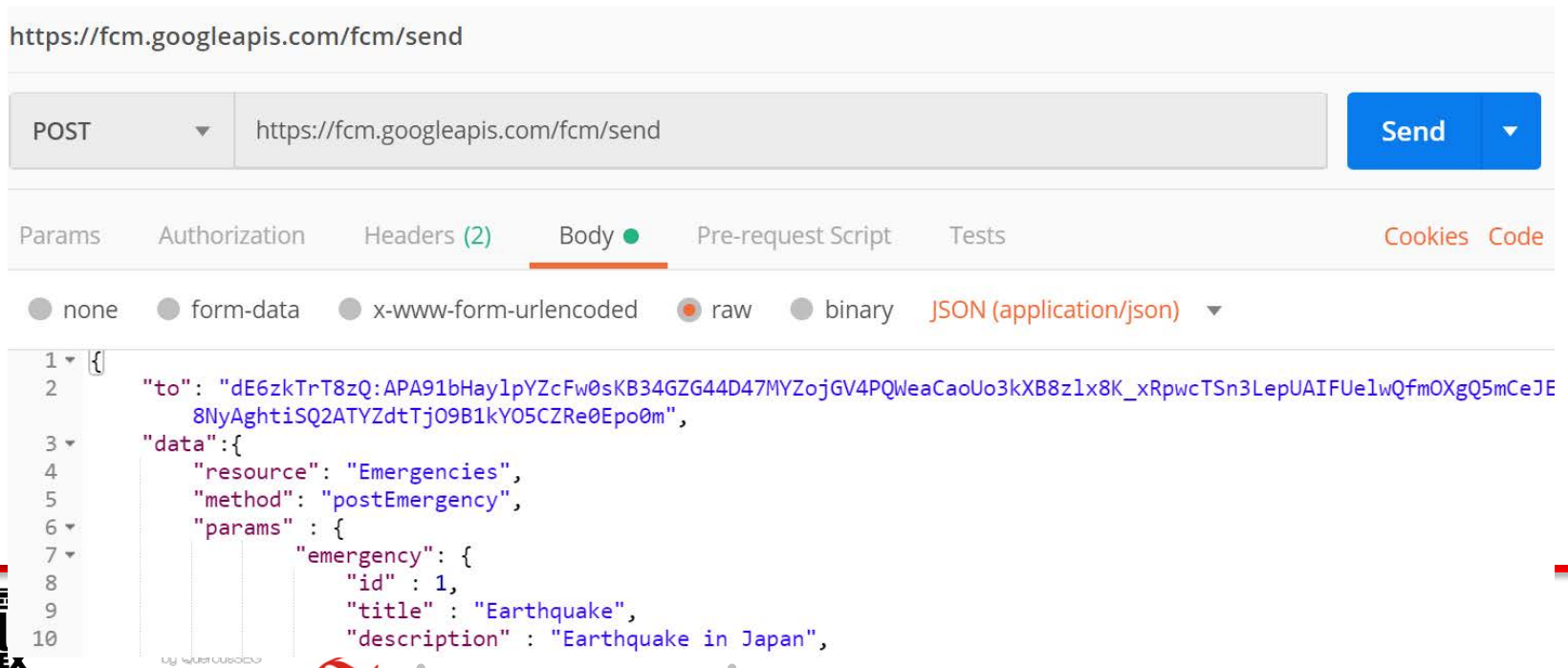
The screenshot shows the 'Settings' page in the Firebase Console, with the 'Cloud Messaging' tab selected. Under 'Project credentials', there is a table with the following data:

Key	Token
Server key	AAAAyWq6InY:APA91bEWJatN7fwooSaOjUyjkvuJ6VGwvU3lHaDUvSVAID9CsJl4cQpZWbdBW2GGyogVP42LiJ_deDJOAzSiDRYBtU2gMgv8h9l62eLwO9wweXRZuFpsMDHJmpVClA9Yeoml8cE1Yie_
Legacy server key ?	AlzaSyCvNOTkTqSFQFZs9g31bT6j2EMumXGCVQw
Sender ID ?	865079009910

An 'Add server key' button is located in the top right corner of the table.

Calling a MC endpoint

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



Calling a MC endpoint

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

<https://openapi-generator-spilab.herokuapp.com/swagger-ui.html>

clients ▾

GET /api/gen/clients Gets languages supported by the client generator

GET /api/gen/clients/{language} Returns options for a client library

POST /api/gen/clients/{language} Generates a client library

Accepts a GeneratorInput options map for spec location and generation options

Parameters Cancel

Name	Description
language * required string (path)	The target language for the client library <div>android html ▾</div>
generatorInput * required (body)	Configuration for building the client library <div>Example Value Model</div> <pre>{ "openAPIUrl": "https://raw.githubusercontent.com/iberolm/RCIS19/master/EmergencyAlerts.yaml"}</pre>

Calling a MC endpoint

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

The screenshot displays the documentation for the 'Emergency Alerts' API. On the left, a sidebar lists navigation options: API SUMMARY, API METHODS - EMERGENCIES (with sub-links for postEmergency and getLocation), API METHODS - LOCATION (with sub-link for getLocation), API METHODS - EMERGENCIES (with sub-link for postEmergency), API METHODS - LOCATION (with sub-link for getLocation), and MODELS (with sub-links for Emergency, EmergencyResponse, and Location). The main content area is titled 'Emergency Alerts' and 'API and SDK Documentation', with a version of 1.0. It describes the application's purpose: monitoring user location to send emergency alerts. A 'Connection Parameters' section contains input fields for 'Authorization Header' (with 'key=' as a placeholder) and 'Token Server'. Below this, the 'Emergencies' section details the 'postEmergency' endpoint, which sends an emergency to devices within a specific area. A 'Parameters' table lists a body parameter named 'emergency' with a description of 'Information about the emergency'. The page concludes with a 'Try it out' section.

API SUMMARY

API METHODS - EMERGENCIES

postEmergency

API METHODS - LOCATION

getLocation

API METHODS - EMERGENCIES

postEmergency

API METHODS - LOCATION

getLocation

MODELS

Emergency

EmergencyResponse

Location

Emergency Alerts

API and SDK Documentation

Version: 1.0

This applications monitor the users' location in order to send them alerts about possible emergencies in their area.

Connection Parameters

Authorization Header

key=

Token Server

Emergencies

postEmergency

Sends an emergency to the devices

An emergency is sent to the devices and shown to the users inside the emergency area

Parameters

Body parameters

Name	Description
emergency *	Information about the emergency

Try it out

A favor



<https://forms.gle/Ko8DH8grZNhL9eU49>

THANK YOU!

