

FH_FFC PROJECT SMART APP BASED ON SAPELLI TOOLBOX

LUCA FRIGERIO

A CASE OF STUDY



istituto per il rilevamento
elettromagnetico dell'ambiente

THE FHFFC PROJECT



- FUTURE HOME FOR FUTURE COMMUNITIES

- The Irea tasks within this project are related to use Geo **Big Data** created by smart app, open data and social media in order to :
 - facilitate the connection between people and their community
 - create a scalable system based on a semantic platform.

THE FIRST STEP

- The first step is to determine the target stakeholders of our project: people in need of help because they are forced at home

Handicapped



The elderly



Children



STAKEHOLDERS NEEDS

- Finding some that provides:
 - Caregiver
 - Elderly assistance
 - Baby sitters
 - Home-shopping service
 - Generic help



Need to find the resources within their neighbourhood

WHERE TO LOOK FOR THE RESOURCES

- Lombardy Region Open DATA



- Tweets



- VGI created by a smart APP



THE APP

Created to enable the offer of resources by individuals.

Everyone now has a smartphone

So, we were searching for a tool to easily create an app.

An App, **easy to use**, containing a form, **easy to fill** :

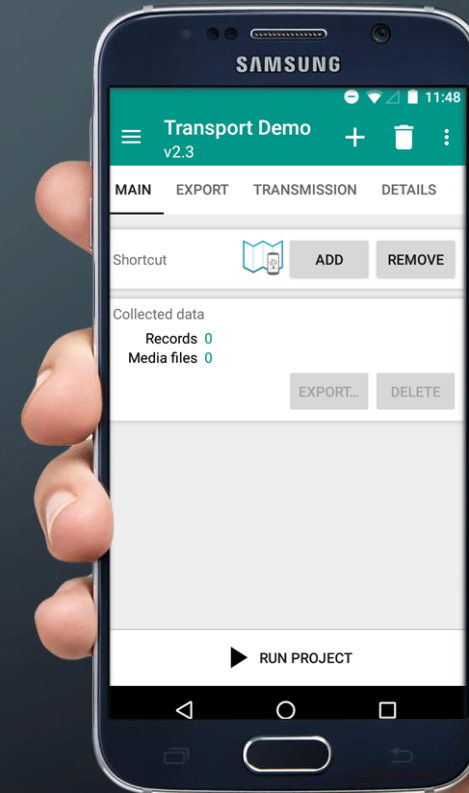
- **who** they are
- **what** kind of service they want to offer
- **where** they want to offer the service.



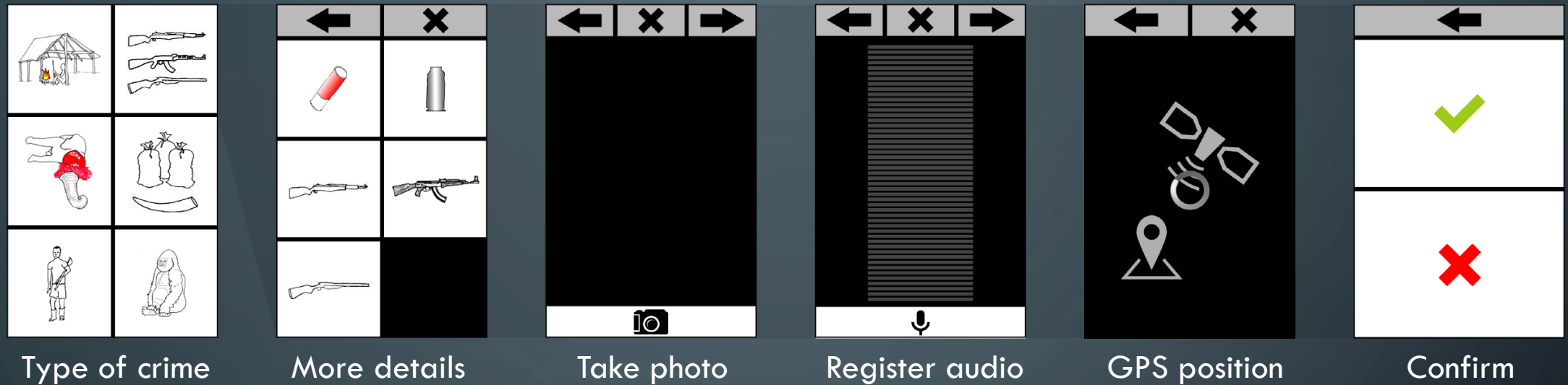
WHAT IS SAPELLI ?

Sapelli is an open-source project that facilitates data collection across language or literacy barriers through **highly configurable icon-driven** user interfaces.

Basically Sapelli is an app, available on google play on git-hub, capable of reading a **custom XML** file that describes a decision tree and automatically creates a set of forms to represent it.



AN EXAMPLE : REPORTING WILDLIFE CRIME



This project has been created to provide a way for forest people – indigenous Baka hunter-gatherers and Bantu farmers in Cameroon – to report **illegal wildlife crime**.

Both indigenous communities are largely non-literate or illiterate but the Sapelli project is entirely icon-based.

HOW DOES IT WORKS ?

- First step : Creation of the decision tree.
- Sapelli uses a **custom set of instructions based** on XML.
- It is easy to learn for someone that knows Html
- You can find a complete documentation here :
<http://www.sapelli.org/full-xml-documentation/>
- To write the «code», you don't need any specific software
you can use notepad, and paint for the images.



CODE EXAMPLE 1A

- All the XML instructions are among the tag SapelliCollectorProject :

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<SapelliCollectorProject id="000" version="1.01" defaultLanguage="it-IT"  
name="FHfFC2" >
```

```
...
```

```
</SapelliCollectorProject>
```

- **id** : for recognizing this project.
- **defaultLanguage** : the language to use when displaying the project to the user (e.g. for text-to-speech synthesis)
- **name** : human-readable name of this project.

CODE EXAMPLE 1 B : THE FORM TAG

A Form element consists of one or more Fields and allows for the collection of data.

A project can have more than one form.

```
<Form id="Main" defaultLanguage="it-IT" clickAnimation="true" screenTransition="horizontal"
saveVibrate="true" >
```

...[HERE THE ELEMENTS OF YOUR FORM]...

```
</Form>
```

- **id** : used to recognise and refer to this Form.
- **clickAnimation** : animate the widget that the user has clicked for providing visual feedback.
- **screenTransition** : make the device vibrate when the user saves their input for this form.

Other important attributes :

- **storeEndTime** : store the time at which the user saved and exited this form
- **Next** : what to do once the user completes this form: start this form again ("loopform"), start this project again ("loopproj"), exit the application ("exitapp"), return to the previous form ("prevform") or go to the next form ("nextform").
Using this command you can create complex decision trees.

CODE EXAMPLE 1C : SIMPLE FORM

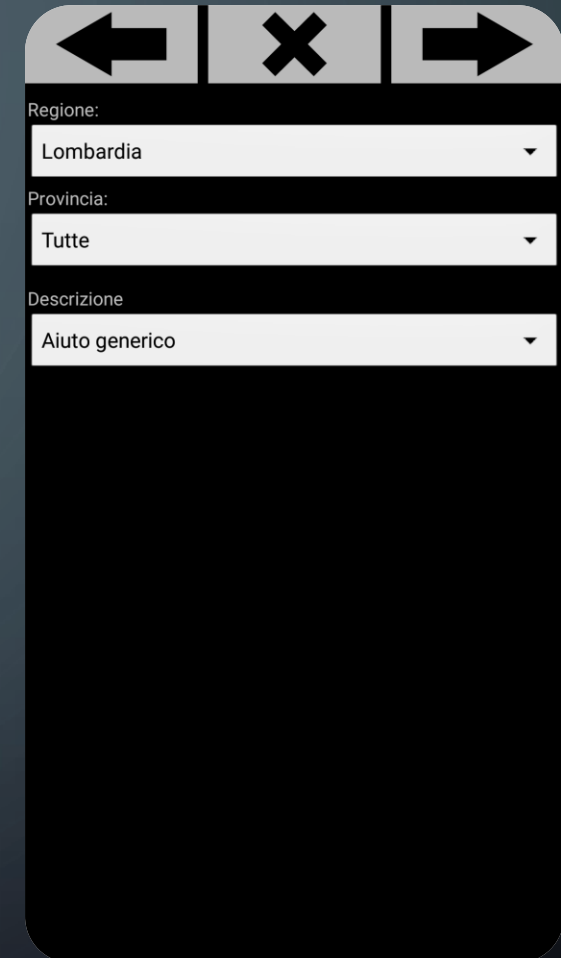
- `<Page id="Anagrafica">`
 - `<Label caption="FHfFC PROJECT v1.0" centered="true" />`
 - `<Label caption=" " centered="true" />`
 - `<Label caption="Iserire i propri dati" centered="true" />`
 - `<Text id="email" caption="Email" content="email" optional="true" />`
 - `<Text id="telefono" caption="Telefono" content="phonenumber" optional="false" />`
 - `<Text id="firstname" caption="Nome" optional="true" content="text" />`
 - `<Text id="secondname" caption="Cognome" optional="true" content="text" />`
 - `<Text id="organization" caption="Organizzazione" optional="true" content="text" />`
- `</Page>`
- **caption** : the field caption is used as a label for the text box.
- **content** : the content type that this field expects (text, password, email, phonenumber...)
- **optional** : this field is mandatory or not

Other important attributes are the constraints for data i.e. `minLength` : the minimum number of characters that should be deemed an acceptable submission for a field.

The image shows a mobile application interface for a form titled "FHfFC PROJECT v1.0". At the top, there is a header bar with a close button (X) on the left and a next button (arrow) on the right. Below the header, the title "FHfFC PROJECT v1.0" is displayed. Underneath the title, the instruction "Iserire i propri dati" (Enter your data) is shown. The form consists of five text input fields, each with a label to its left: "Email" (containing "info@lucafrigerio.com"), "Telefono", "Nome" (containing "Luca"), "Cognome" (containing "Frigerio"), and "Organizzazione" (containing "Cnr Irea"). The "Organizzazione" field is highlighted with an orange border. The form is set against a dark background.

CODE EXAMPLE 1D : LIST AND MULTILIST

```
• <MultiList id="whereDoYouLive" captions="Regione;Provincia:" preSelectDefault="true">
  <Item value="Lombardia" default="true">
    <Item value="Tutte" default="true" />
    <Item value="Bergamo" />
    <Item value="Brescia" /> <Item value="Como" />
    <Item value="Cremona" /> <Item value="Lecco" />
    <Item value="Lodi" /> <Item value="Mantova" />
    <Item value="Milano" />
    <Item value="Monza e Brianza" />
    <Item value="Pavia" /> <Item value="Sondrio" />
    <Item value="Varese" />
  </Item>
  <Item value="Piemonte">
    <Item value="Tutte" />
    <Item value="Alessandria" />
    <Item value="Asti" />
    <Item value="Biella" />
    <Item value="Cuneo" />
    <Item value="Novara" />
    <Item value="Torino" />
    <Item value="Verbano Cusio Ossola" />
    <Item value="Vercelli" />
  </Item>
</MultiList>
```



The image shows a mobile application interface with a dark theme. At the top, there is a navigation bar with three icons: a back arrow, a close 'X' button, and a forward arrow. Below the navigation bar, there are three sections, each with a label and a dropdown menu:

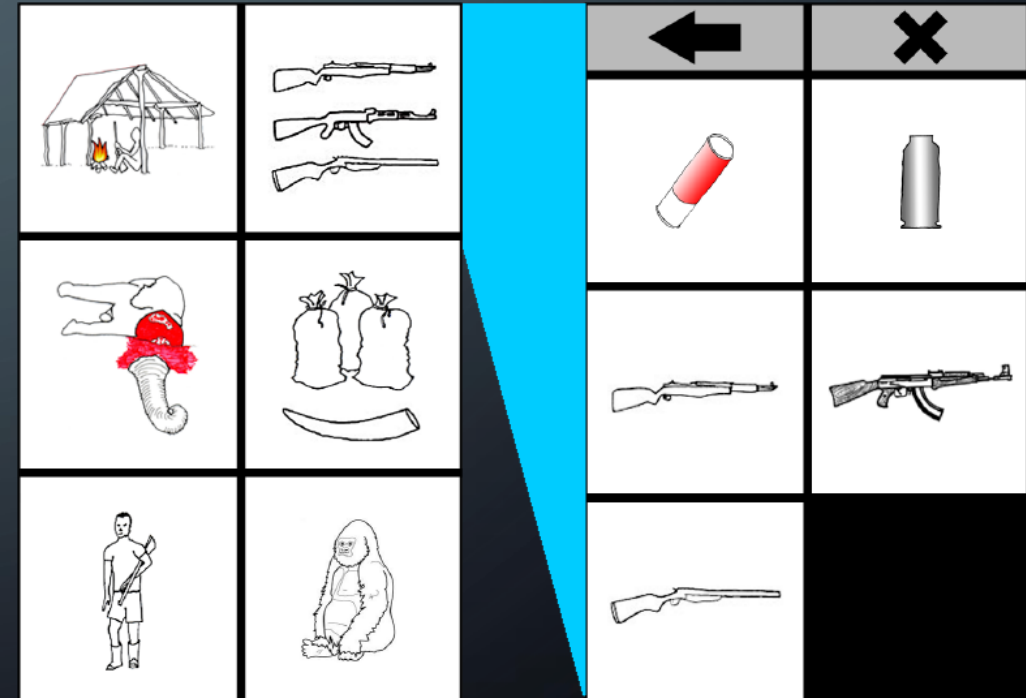
- Regione:** The dropdown menu is open, showing 'Lombardia' as the selected option.
- Provincia:** The dropdown menu is open, showing 'Tutte' as the selected option.
- Descrizione:** The dropdown menu is open, showing 'Aiuto generico' as the selected option.

Below these sections, there is a large, empty white rectangular area, likely a placeholder for content or a list of items.

CODE EXAMPLE 1E : ICON FORM

```
<Choice img="Weapons.png" rows="3" cols="2">  
  <Choice value="Cartouche de calibre-douze" img="ShotgunCartridge.png" />  
  <Choice value="Cartouche de fusil" img="RifleBulletHousing.png" />  
  <Choice value="Fusil" img="WWIIRifle.png" />  
  <Choice value="AK47" img="AK47.png" />  
  <Choice value="Calibre-douze" img="ShotGun.png" />  
</Choice>
```

Sapelli supports both png and jpeg format.



CODE EXAMPLE 1E : MEDIA TYPE AND GPS

- The Photo element is given an ID name, this Field has been declared as optional for the user.

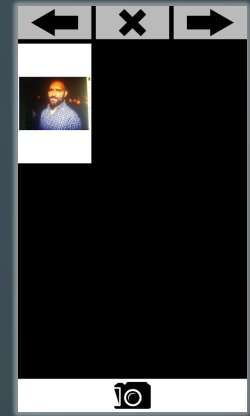
```
<Photo id="Photo" optional="true"/>
```

- The Audio element is given an ID name, this Field has been declared as optional for the user. A custom image is set to represent an audio recording on the review screen.

```
<Audio id="Audio" optional="true" recordingImg="AudioClip.png" />
```

- The Location element is given an ID name. The user's location will be recorded using GPS if a location better than 50m accuracy is captured. Altitude will be discarded. The location field will be skipped when the user goes backwards through fields they have already encountered.

```
<Location id="Location" maxAccuracyRadius="50.0"  
  storeAltitude="false" skipOnBack="true"/>
```



CODE EXAMPLE 1F : END OF THE FORM

```
<Choice id="Confirmation" optional="never" rows="2" showCancel="false"
noColumn="true">
```

If the user were to click the “Salva” button, a special jump is executed to save the collected data and start the form again.

```
<Choice caption="Salva" jump="_SAVE+EXITAPP" />
```

When pressing the “Rifai” button, a Special Jump “_END” will be executed, meaning that this option will not save the user's data, will clear the data from the Form and start the form again.

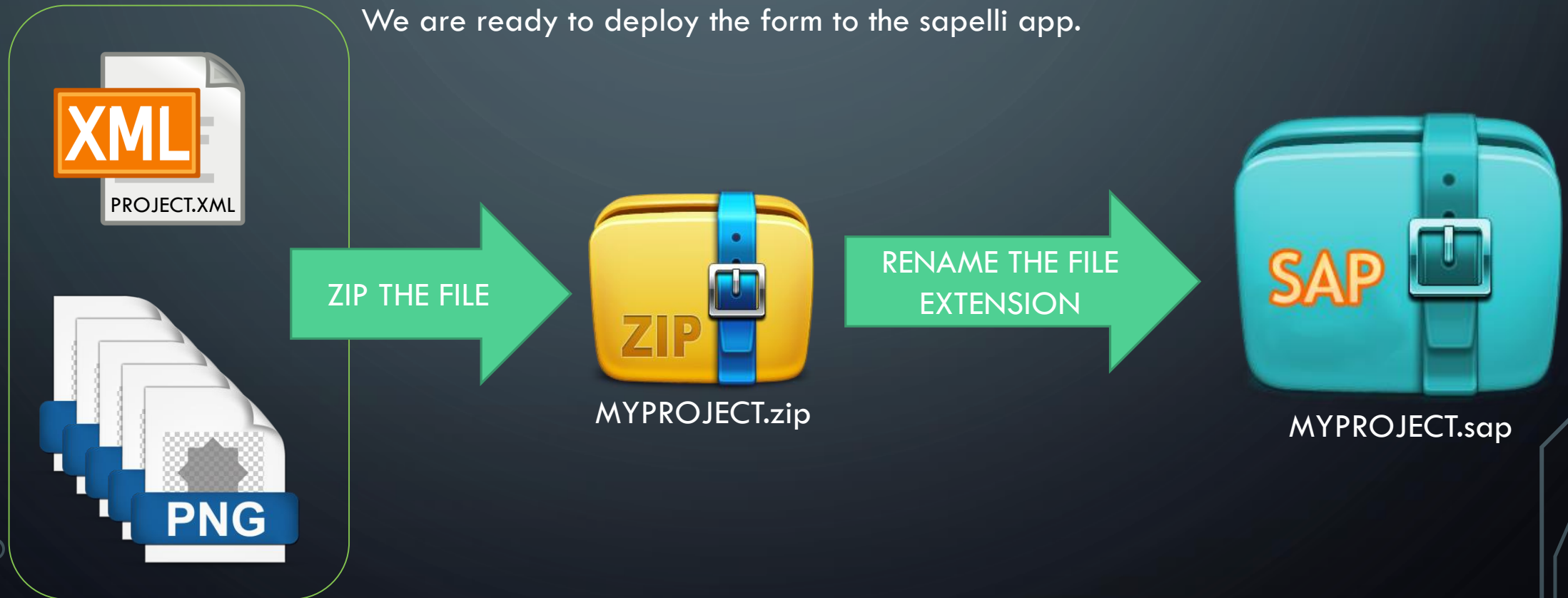
```
<Choice caption="Rifai" jump="_LOOPFORM" />
</Choice>
```



The image shows a mobile application form with a dark blue background. At the top, there is a grey bar with a black back arrow. Below this, the form is divided into two main sections. The first section has a white background and contains the text "Salva" in a large, black, sans-serif font. The second section also has a white background and contains the text "Rifai" in a large, black, sans-serif font. The form is reflected on the surface below it.

CODE EXAMPLE 1 G : CREATE THE FILE

- Now we can save our file as PROJECT.XML, and put all the images in the same folder.
- Zip all the file (not the folder) with the name you want.
- Rename the extension from .zip to .sap



HOW TO INSTALL THE PROJECT APP ON A MOBILE DEVICE

First, download the app from the store (or from Git-Hub) :

<https://play.google.com/store/apps/details?id=uk.ac.ucl.excites.sapelli.collector>

You have to 3 modalities.

1. USB cable

Save the file directly to the filesystem of the device



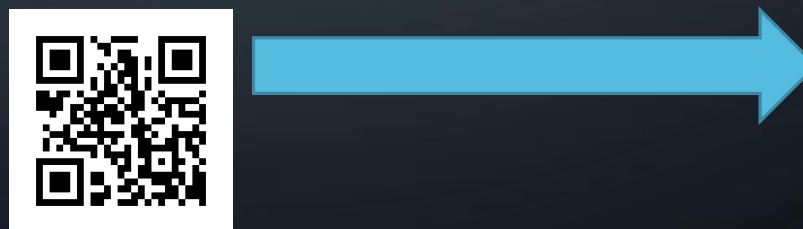
2. Online LINK / Shared file

Store the .sap file online

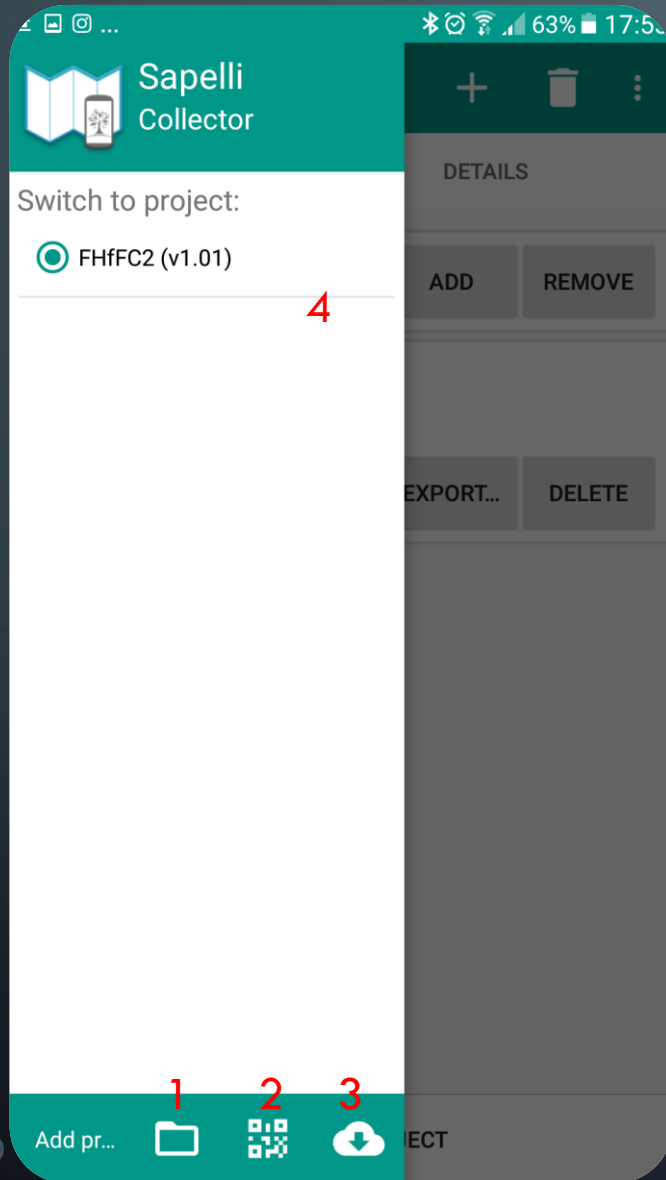


3. QR Code (containing the link)

Sapelli Collector can read QRCode

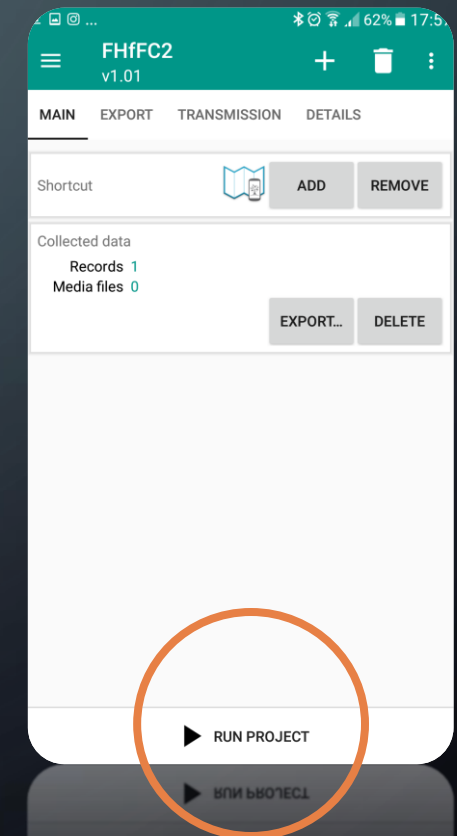


HOW TO INSTALL THE PROJECT APP ON A MOBILE DEVICE



1. Select a .sap from the file system
2. Scan a QR Code
3. Select an URL
4. Select a Project already downloaded

Then Run the selected project :



HOW TO CREATE VGI BY THE FHFFC SAPELLI APP

FHFFC PROJECT v1.0

Iserire i propri dati

Email
info@lucafrigerio.com

Telefono

Nome
Luca

Cognome
Frigerio

Organizzazione
Cnr Irea

Regione:
Lombardia

Provincia:
Tutte

Descrizione:
Aiuto generico

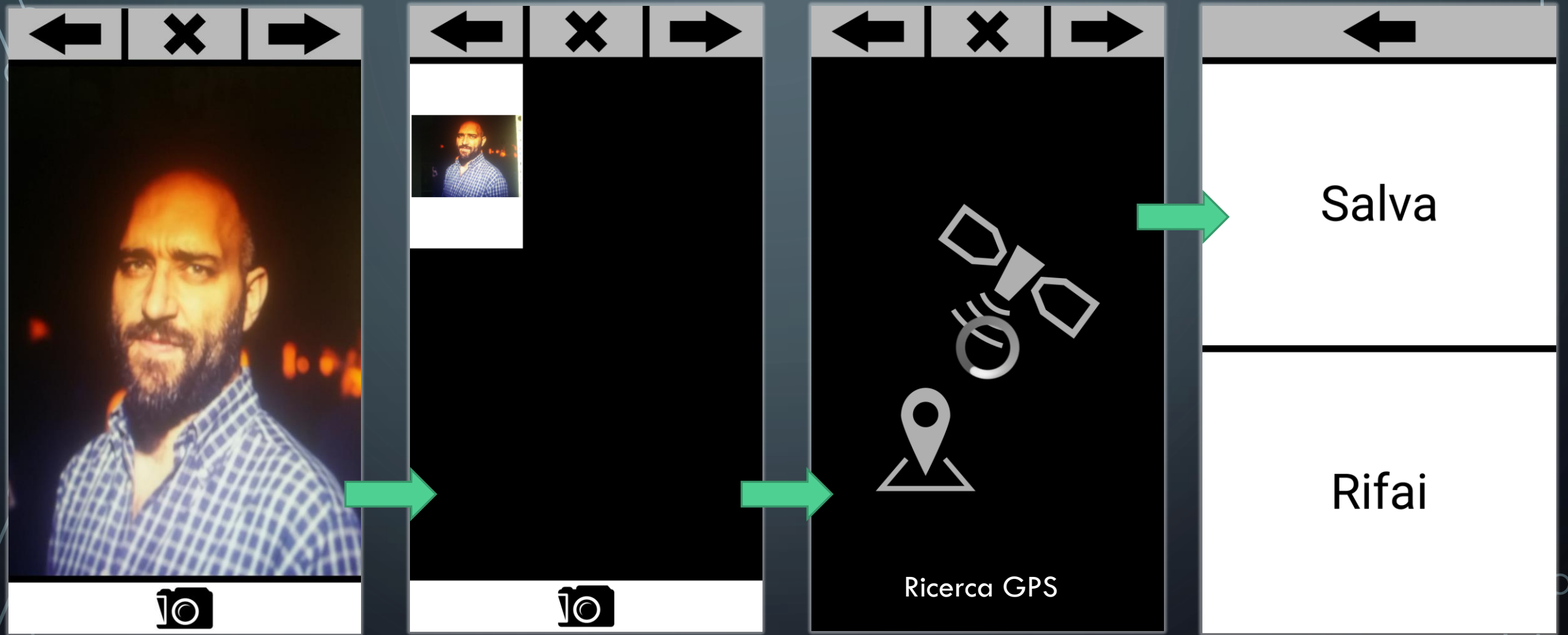
Descrizione

- ☒ Aiuto generico
- ☐ Assistenza Anziani
- ☐ Care Giver
- ☐ Baby Sitter
- ☐ Spesa a Domicilio

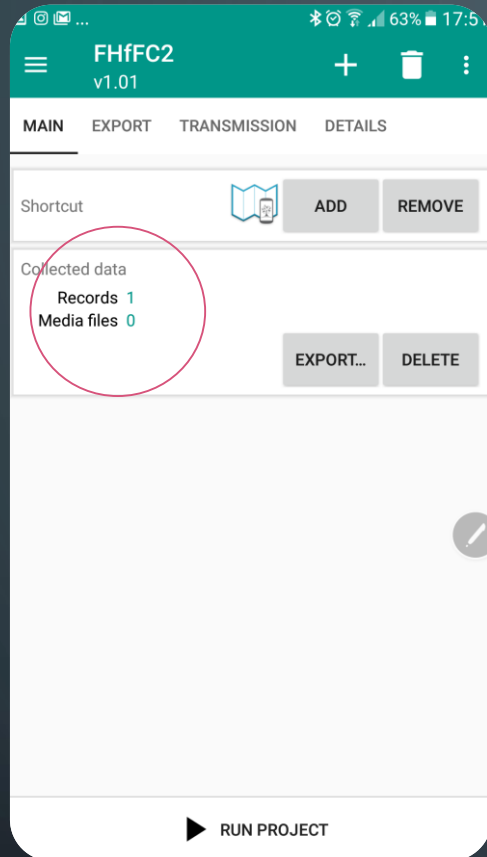
Provincia:

- ☒ Tutte
- ☐ Bergamo
- ☐ Brescia
- ☐ Como
- ☐ Cremona
- ☐ Lecco
- ☐ Lodi
- ☐ Mantova
- ☐ Milano
- ☐ Monza e Brianza

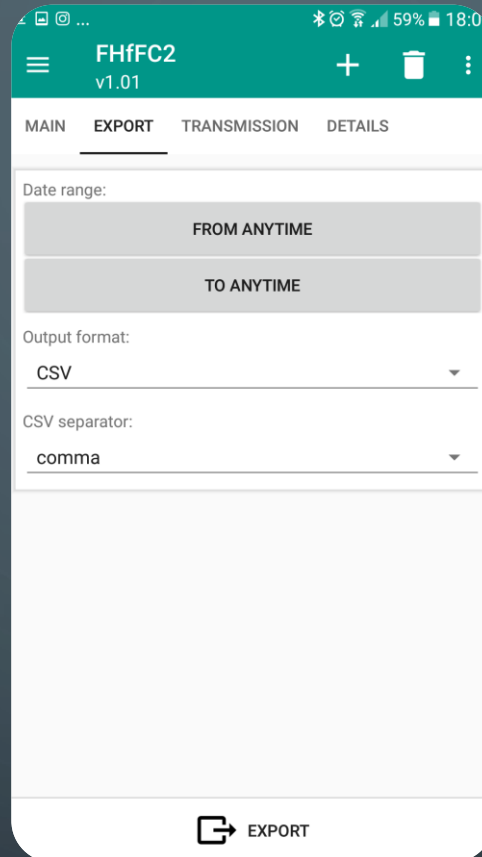
HOW TO CREATE VGI BY THE FHFFC SAPELLI APP



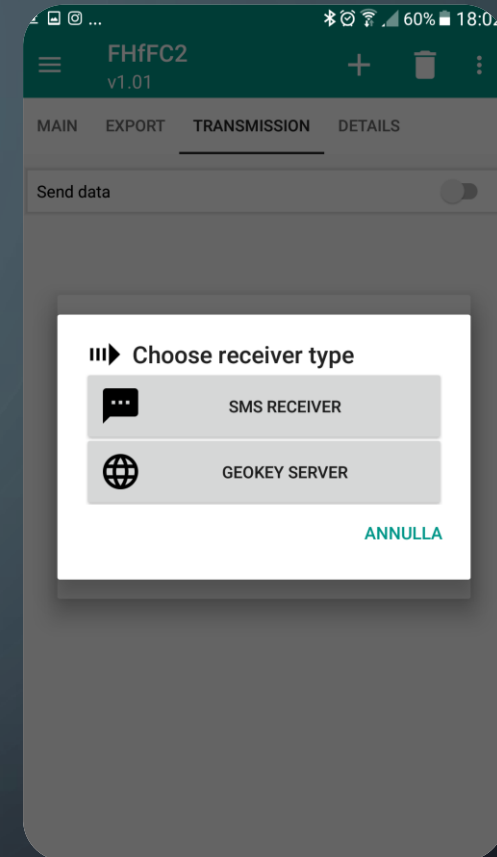
HOW TO SAVE VGI LOCALLY OR REMOTELY



Now we have a VGI record created to our project



LOCAL STORAGE:
We can simply export it to mobile file Sytem in XML o CSV format



REMOTE STORAGE:

- ✓ we can send it to an SMS receiver by using SMS
- ✓ We can upload to «GeoKey»

WHAT IS GEOKEY ?

- GeoKey IS a generic data platform, written in JAVA that uses PostGres/Postgis as database.
Thanks to an external plugin, GeoKey can interact with Sapelli app and stores the data received by the project apps.
It also offers some API to query and retrieve the data.
- GeoKey is NOT part of the Sapelli collector.
<http://geokey.org.uk/>

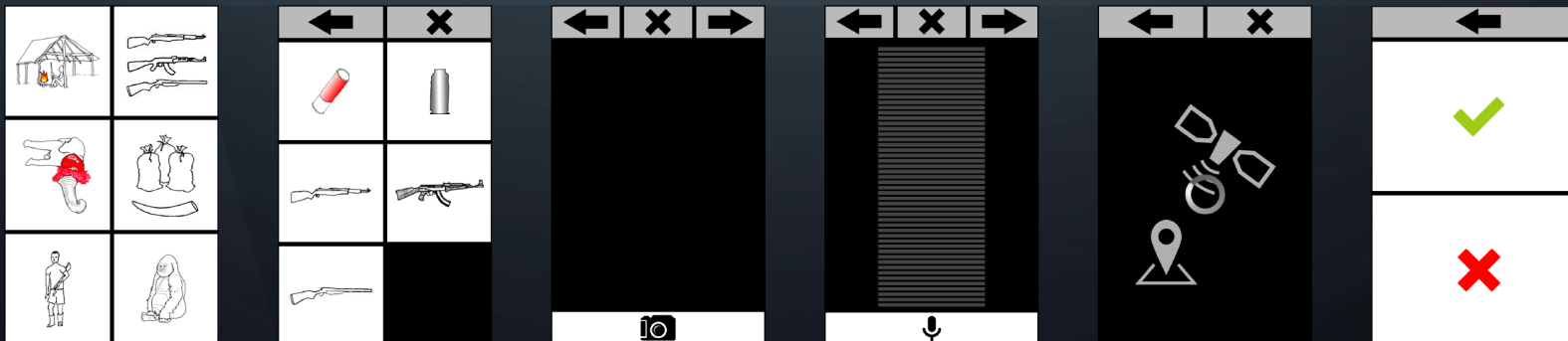


SAPELLI STRONG POINTS

FAST DEPLOY of BOTTOM-UP VGI creation:

In case of emergency, in **few hours** and **without any internet** connection you can deploy also on **very low-end devices**:

- a set of forms **icon based**, easy to understand to everyone
- You can also send information using mail (using export) or share it on cloud... it is a simple file!
- **You don't need know how to code in a programming language but just need to know Sapelli XML**



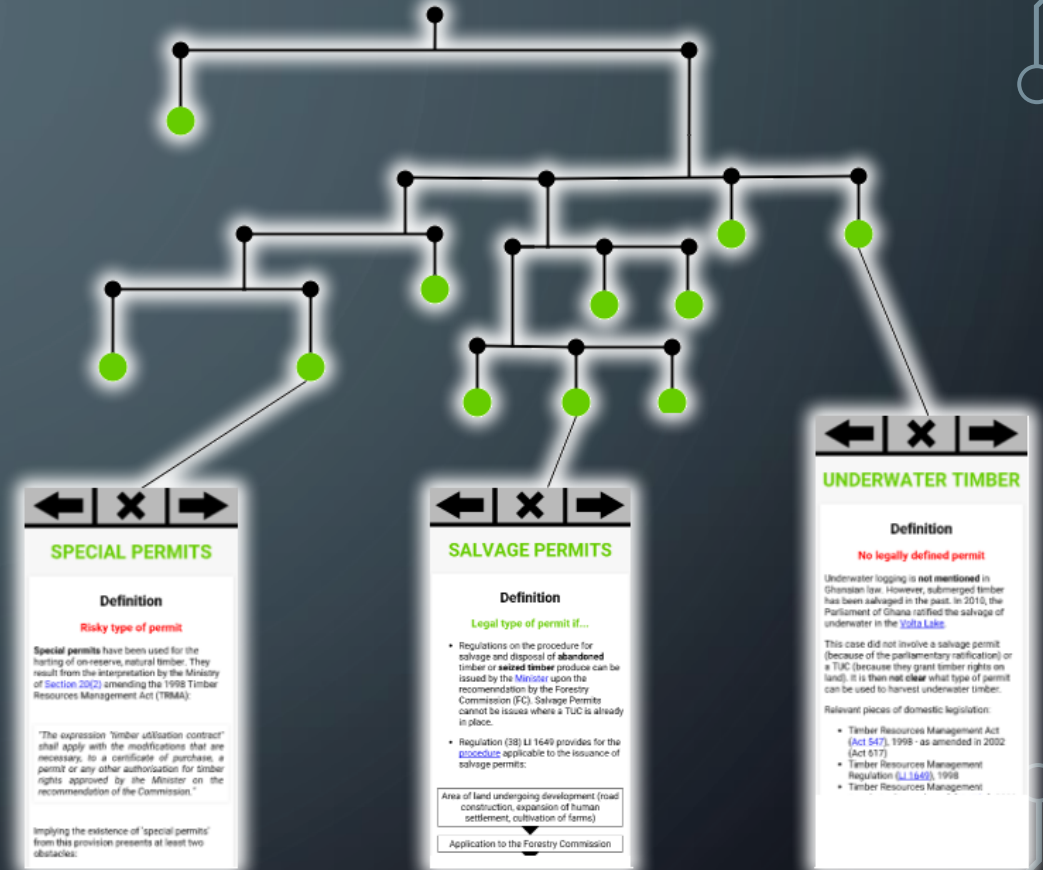
SAPELLI STRONG POINTS

EASY TOP-DOWN information deploy:

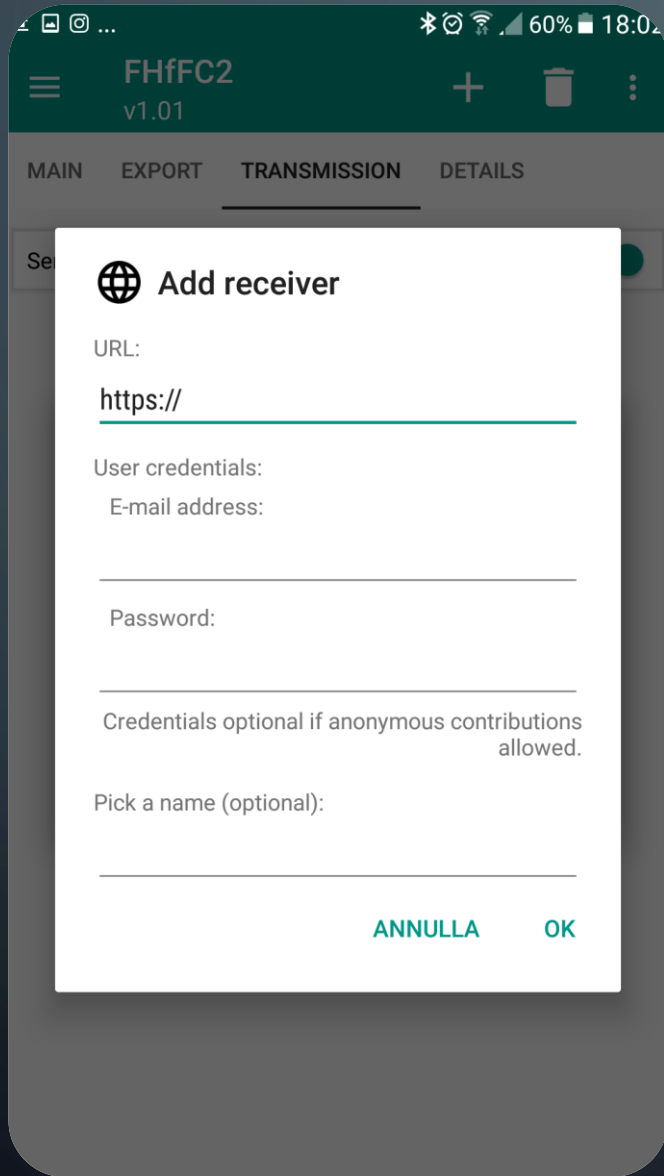
Sapelli serves other purposes than just creating VGI...

It allows creating forms representing **decision trees** to provide guidance instructions to stakeholders: for example how to behave in case of emergency.

Case study: was used to provide legal guidance to protect natural resources and local community rights in Ghana.



THE SAPELLI DRAWBACKS



The screenshot shows a mobile application interface for 'FHfFC2 v1.01'. The top navigation bar includes 'MAIN', 'EXPORT', 'TRANSMISSION', and 'DETAILS'. A dialog box titled 'Add receiver' is open, featuring a globe icon. It contains the following fields and text: 'URL:' with a text input containing 'https://'; 'User credentials:' with a sub-label 'E-mail address:' and a text input; a 'Password:' text input; a note 'Credentials optional if anonymous contributions allowed.'; and a 'Pick a name (optional):' text input. At the bottom of the dialog are two buttons: 'ANNULLA' and 'OK'.

It is not possible to set the connection properties in the SAP file.

So every user must set the connection manually.

It is not difficult, for me...

But ... do you remember the target of our FHfFC projet ?



Even considering a generic caregiver or a person who offers assistance to elderly, it is a little bit too complicated...

THE SAPELLI DRAWBACKS

It is not possible to visualize my data on the mobile device

Then, how to visualize the created data by Sapelli?

using an external tool that query Geokey repository,
retrieved the json data and maps them.

But this must be developed ad hoc

Moreover:

You can't:

- Modify data stored in Geokey
- Delete data stored in Geokey



THE SAPELLI DRAWBACKS: IMHO

1. GeoKey it is **not easy to install** and it lacks of documentation to help you.

I've spent 2 hour to create a form, but 4 days and half to make GeoKey work, configure the PostGress database, the Sapelli plugin, configure the project, ask help on Stack Overflow, and understand how to retrieve data.

2. You can not decide the data structure in GeoKey.

3. Sapelli collector is in English only.

4. Sapelli collector is an **Android Native app**, and it is open source.

You can't use it on web or on iPhone.

The structure is not easy to manipulate, but I'll show you next why.

5. Sapelli collector uses a custom XML set of tag.

The only documentation is on their web site and it is difficult to find help elsewhere.

THE SAPELLI DRAWBACKS TO MODIFY/EXTEND IT

- To modify/extend Sapelli one needs to write Java Native code, which is verbose, further Sapelli code not so easy to understand and manage.
- Using Java you have to handle asynchronous tasks with care in uploading or downloading data. That means it takes more attention and time to implement web request (ie sending VGI) with respect to other languages such as Javascript.
- Json is not native in JAVA.

```
1 // Import the classes we need
2 import java.io.*;
3 import java.net.*;
4 import java.util.*;
5
6 // Main class
7 public class Main {
8     // Main method
9     public static void main(String[] args) {
10         // Create a URL object
11         URL url = new URL("http://www.example.com");
12         // Create a HttpURLConnection object
13         HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
14         // Set the request method to GET
15         httpURLConnection.setRequestMethod("GET");
16         // Set the request headers
17         httpURLConnection.setRequestProperty("User-Agent", "Mozilla/5.0");
18         // Get the response code
19         int responseCode = httpURLConnection.getResponseCode();
20         // Check if the response is successful
21         if (responseCode == HttpURLConnection.HTTP_OK) {
22             // Get the response data
23             InputStream inputStream = httpURLConnection.getInputStream();
24             // Read the response data
25             BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));
26             // Read the response data line by line
27             String line;
28             while ((line = bufferedReader.readLine()) != null) {
29                 // Print the response data
30                 System.out.println(line);
31             }
32         } else {
33             // Print the response code
34             System.out.println("Response code: " + responseCode);
35         }
36     }
37 }
```

Subscribe Request JAVA
(118 lines)

```
1 // Import the classes we need
2 import java.io.*;
3 import java.net.*;
4
5 // Main class
6 public class Main {
7     // Main method
8     public static void main(String[] args) {
9         // Create a URL object
10         URL url = new URL("http://www.example.com");
11         // Create a HttpURLConnection object
12         HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
13         // Set the request method to GET
14         httpURLConnection.setRequestMethod("GET");
15         // Set the request headers
16         httpURLConnection.setRequestProperty("User-Agent", "Mozilla/5.0");
17         // Get the response code
18         int responseCode = httpURLConnection.getResponseCode();
19         // Check if the response is successful
20         if (responseCode == HttpURLConnection.HTTP_OK) {
21             // Get the response data
22             InputStream inputStream = httpURLConnection.getInputStream();
23             // Read the response data
24             BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));
25             // Read the response data line by line
26             String line;
27             while ((line = bufferedReader.readLine()) != null) {
28                 // Print the response data
29                 System.out.println(line);
30             }
31         } else {
32             // Print the response code
33             System.out.println("Response code: " + responseCode);
34         }
35     }
36 }
```

Subscribe Request Javascript
(5 lines)



Using android native Java code, it is more difficult to embed open source elements for managing maps.

THAT'S THE REASONS WHY I CHOSE A RECENT TECHNOLOGY



I've decided to develop the FHfFC app using NODE environment and IONIC framework.

Everything open source.

The server has been created using nodejs.

- Full access on the data structure and use noSQL database to ease integration with open data and collected tweets .
- Easy to write (this is all the code needed 200 lines)
- Low resource consumption (Geokey is in Java)

```
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const app = express();
const PORT = 3000;

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

mongoose.connect('mongodb://localhost:27020/fhf');

const tweetSchema = new mongoose.Schema({
  text: String,
  user: String,
  retweets: Number,
  likes: Number,
  createdAt: Date
});

const Tweet = mongoose.model('Tweet', tweetSchema);

app.post('/tweets', async (req, res) => {
  const { text, user, retweets, likes } = req.body;
  const tweet = new Tweet({
    text,
    user,
    retweets,
    likes,
    createdAt: Date.now()
  });
  await tweet.save();
  res.status(201).json(tweet);
});

app.get('/tweets', async (req, res) => {
  const tweets = await Tweet.find({});
  res.status(200).json(tweets);
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```



- User Interface is written in HTML-5 (HTML + JavaScript)
- Multiplatform environment: You write the code one time, and you can produce the WebSite, the Android App and the IOS app simultaneously;
- With 230 lines of code and 60 lines for the UI you can define a form, detect GPS location, send VGI item to the server, view the VGI created by all users.
- Basically, for a programmer:
- it takes the same amount of time for configuring Sapelli and Geokey.
 - User CRUD
 - Service CRUD
 - More control over structure and code
 - More performance
 - Full Multilanguage text



Source code
JavaScript



UI Ionic



Sapelli Xml

SCREEN SHOT APP 1

← FHfFC App

Inserisci i tuoi dati

Tutti i campi sono obbligatori

Email *
info@lucafrigerio.com

Telefono
+55 555 5555555

Nome *
Luca

Cognome *
Frigerio

Password *

Conferma Password *

Desideri privilegiare le ricerche
che riguardano prevalentemente:

ANZIANI ☒

BAMBINI ☐

DISABILI ☐

INVIA

≡ FHfFC Offerte

Inserisci la tua offerta!

Solo i campi marchiati con il simbolo
asterisco sono obbligatori

Che servizio si desidera offrire ? *

Organizzazione
Luca Frigerio Assistenza

Descrizione
Assistenza Anziani

Indicare l'indirizzo e l'area
in cui si desidera offrire servizio

Indirizzo *
via Ugo Foscolo,1

Città o paese *
Monticello

Cap
Inserire il C.A.P.

Area in cui si desidera offrire il servizio *

INVIA

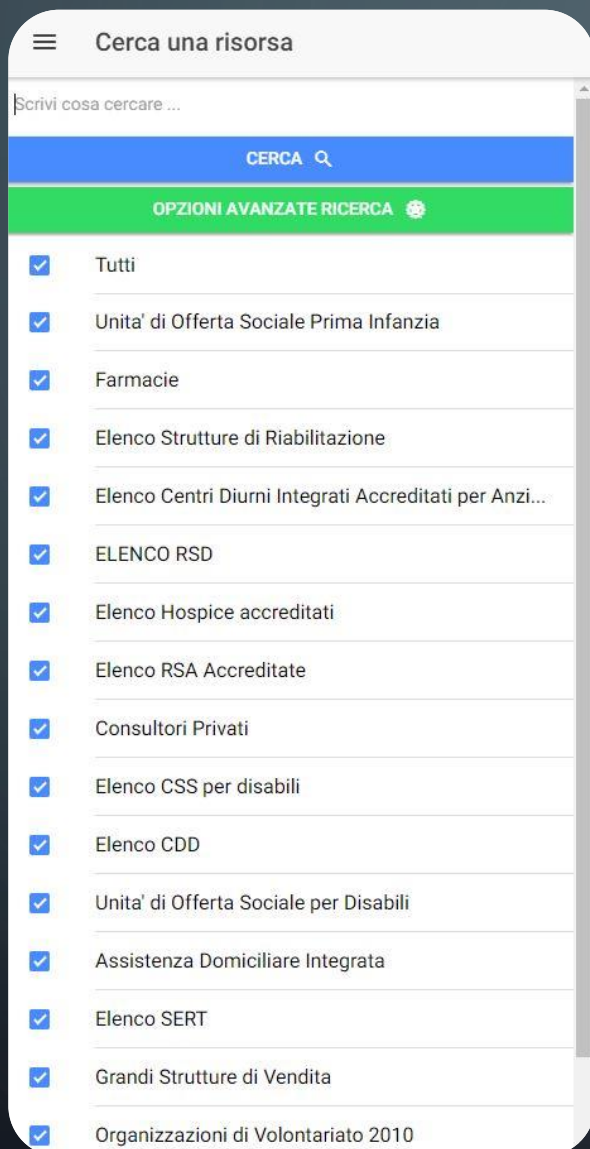
← FHfFC Cerca una risorsa

Scegli file Nessun file selezionato

INVIA ->



SCREEN SHOT APP 2



SCREEN SHOT APP 3

