Introduction to OpenNoiseMap.org

Based on slides by:
Marius Appel
Holger Hopmann
André Roß
Arne de Wall
Agenda

• Introduction
• Mobile application - noiseDroid
• Server - opennoisemap.org
Smartphones

• Multi-sensor platform
  – In your pocket (anywhere, anytime)
  – GPS, camera, microphone, accelerometer, pressure, magnetic field, light
  – 3G and WiFi connection

• Enables crowd-sourcing /crowd-sensing

• Example
  – OpenFloorMap.org
  – OpenNoiseMap.org
Noise as environmental pollution

#2
Noise as environmental pollution

• Effects
  – Hearing loss
  – Cardiological effects
  – Stress
  – Annoyance

• Emitters
  – Airports, cars, industrial sites, etc.

• Subjective
  – Cannot be modeled or calculated
Features

- Collecting noise observations
- Accessing noise observations
  - Mobile and browser-based
  - Own data & community data
- Central storage of noise observations and anonymous access
- Map-based visualization
- Multi-lingual app NoiseDroid
- www.opennoisemap.org
- NoiseDroid in the Android market!
Mobile App - NoiseDroid

• Noise mapping through automatic and manual measurements

• Upload to opennoisemap.org

• Visualizing noise observations

• Android-based
Mobile App

- Capturing noise measures
  - Manual capture
  - Automatic capture
  - Event-based capturing
  - Measure series

- Access measures
  - List
  - Map

- Community
  - Register & log in
  - Upload of observations
  - Download of observations
- manual measurement
- automatic measurement
- event-based measurement
- series

MOBILE APP - NOISEDROID
Manual Measurement

• Measurement in dB
• Duration: 5 Sekunden
• Calculates
  – minimum dB Wert
  – maximum dB Wert
  – Average dB Wert
  – Quality of measurement
• Automatic positioning (GPS, WiFi, cell-ID)
Manual Measurement

• Measurement in dB
• Duration: 5 Sekunden
• Calculates
  – minimum dB Wert
  – maximum dB Wert
  – Average dB Wert
  – Quality of measurement
• Automatic positioning (GPS, WiFi, cell-ID)
Manual Measurement

• Measurement in dB
• Duration: 5 Sekunden
• Calculates
  – minimum dB Wert
  – maximum dB Wert
  – Average dB Wert
  – Quality of measurement
• Automatic positioning (GPS, WiFi, cell-ID)
Additional information

- Subjective rating
- Tagging
- Comment
Additional information

• Subjective rating
• Tagging
• Comment
Additional information

• Noise emitter (source)
  – Close to current position
  – Further away
    • Distance (Measure by distance tool)
    • Exact position
  – Cannot be determined
Additional information

• Distance tool
  – Helps to measure the distance to the source
  – Calculated based on angle function
  – Height of device to be configured in settings
  – How do I use it? Simple!
    • Point & shoot 😊
Automatic measurement

• Measurements based on time intervals

• Configure under settings

• Automatic measurements are indicated in the status bar
Automatic measurement

- Measurements based on time intervals
- Configure under settings
- Automatic measurements are indicated in the status bar
Automatic measurement

• Measurements based on time intervals

• Configure under settings

• Automatic measurements are indicated in the status bar
Event-based Measurement

• Extension of the automatic measurement

• Measurement taken upon specific system event
Event-based Measurement

• Extension of the automatic measurement

• Measurement taken upon specific system event
Event-based Measurement

- Extension of the automatic measurement
- Measurement taken upon specific system event
Measurement quality

- Quality important aspect of automatic measurements
- Quality provides context to the measurement
- Rule-based evaluation of sensors for calculating context
Series

- Logic collection of measurements
- Status
  - active
  - Paused / on hold
  - terminated
- Break conditions
  - Elapsed time
  - Spatial extent
Temporal constraint

- Automatic termination when time/date is reached
- Predefined and user-defined duration
Temporal constraint

- Automatic termination when time/date is reached
- Predefined and user-defined duration
Temporal constraint

- Automatic termination when time/date is reached
- Predefined and user-defined duration
Temporal constraint

- Automatic termination when time/date is reached
- Predefined and user-defined duration
Temporal constraint

- Automatic termination when time/date is reached
- Predefined and user-defined duration
Spatial constraint

- Terminate series whenever smartphone is outside of the spatial area
Spatial constraint

• Terminate series whenever smartphone is outside of the spatial area
Start/Stop of the series

- Series requires a title
- Series can be put on hold
List

- List of all (local & community) measures
- Grouped and sorted by time
List

• Symbols
  - Automatic measures
  - Manual measures
  - Measure uploaded
  - Measure downloaded
  - Error while uploading
Functions

- Grouping
- Sorting
- Uploading
- Filtering
- Community
Measurement details

- Detailed information per measurement
Map

- Map button
- Each measurement is a circle
  - Green: Low noise (35 dB)
  - Yellow: Medium noise (50 dB)
  - red: high noise (70 dB)
Map functions

- Layers
- Search
- My location
- Selection
- (Filtern)
- (Community)
BROWSER-BASED MAPS
Map

• Technology
  – OpenLayers & Ext Js
• Measurements served by WFS, GeoServer
• Basemaps
  – OpenStreetMap
  – Google Maps
• Client-side interpolation of measurements
Map

Selecting measurements

Basemap

Temporal selection
Map

Activate/hide tab
Map
• Architecture
• Software
• XML-messaging

INSIDE THE PLATFORM
Architecture

Userdata Measurement data

Server

Java EE 6 Infrastructure (GlassFish 3.1)

Servlets 3.0

Enterprise Java Beans 3.1

JAAS JPA 2.1

GeoServer 2.1.1

HTTP-Post (XML) Authentifi. complex queries, adding measurement.

HTTP-Post (JSON) Simple queries

NoiseTube Data

Internet

App

Measurement Management

Map View

List View

Website

Map View

List View

WFS

XML

WMS

JSON
Software used

Client Tier
- App
- Browser

Web Tier
- Java Server Pages/Faces

Business Tier
- Enterprise Beans

EIS Tier
- Datenbank

Android-basiertes Smartphone
Clientseitiger Browser

Java EE 6 Technologie auf einem Glassfish Application Server

PostgreSQL mit PostGIS extension
Architecture overview
Endpoint

Server: http://giv-noismappin1.uni-muenster.de:8080/NoiseServerServlets/AppServlet

Geoserver: http://giv-noismappin1.uni-muenster.de:8080/geoserver

SVN (read-only): URL: https://subversion.ifgi.de/noise-game/
  Login: initNoiseGame
  Password: initNoiseGame
## Message structure

<table>
<thead>
<tr>
<th>Client (Request)</th>
<th>Server (Response)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;Requests&gt;</code></td>
<td><code>&lt;Responses&gt;</code></td>
</tr>
<tr>
<td><code>&lt;Selfdescription&gt;</code></td>
<td><code>&lt;Response type=&quot;Typ1&quot;&gt;</code></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>&lt;/Selfdescription&gt;</td>
<td>&lt;/Response&gt;</td>
</tr>
<tr>
<td><code>&lt;Request type=&quot;Typ1&quot;&gt;</code></td>
<td><code>&lt;Response type=&quot;Typ2&quot;&gt;</code></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>&lt;/Request&gt;</td>
<td>&lt;/Response&gt;</td>
</tr>
<tr>
<td><code>&lt;Request type=&quot;Typ2&quot;&gt;</code></td>
<td>&lt;/Responses&gt;</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Server operations

• Server supports
  – Register
  – Login
  – GetInfo
  – AddMeasures
  – UpdateSeries
  – GetMeasures
## Example: Register

<table>
<thead>
<tr>
<th>Request „Register“</th>
<th>Response: Registering successful</th>
<th>Response: Registering failed</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;Request type=&quot;Register&quot;&gt;</code>&lt;br&gt;  <code>&lt;Username value=&quot;test&quot; /&gt;</code>&lt;br&gt;  <code>&lt;Email value=&quot;test&quot; /&gt;</code>&lt;br&gt;  <code>&lt;Password value=&quot;f868315c1b...a223&quot; /&gt;</code>&lt;br&gt;  <code>&lt;/Request&gt;</code></td>
<td><code>&lt;Response type=&quot;Register&quot;&gt;</code>&lt;br&gt;  <code>&lt;Successful /&gt;</code>&lt;br&gt;  <code>&lt;/Response&gt;</code></td>
<td><code>&lt;Response type=&quot;Register&quot;&gt;</code>&lt;br&gt;  <code>&lt;Error type=&quot;AlreadyRegistered&quot; /&gt;</code>&lt;br&gt;  <code>&lt;/Response&gt;</code></td>
</tr>
</tbody>
</table>
AddMeasures: Request

**Attribute: Automatic measurement**

```xml
<Measure id="0" type="automatic">
  <Time value="2011-02-11 16:53:42" />
  <LocationMeasure longitude="7.000000" latitude="51.000000"
         accuracy="131.0" provider="network" />
  <Noise value="42.451363" />
  <Moral value="100.0" />
</Measure>
```
AddMeasures: Request

### Attribute: Manual Measurement

```xml
<Measure id="0" type="manual">
  <Time value="2011-02-11 16:53:42"/>
  <LocationMeasure longitude="7.000000" latitude="51.000000"
      accuracy="131.0" provider="network"/>
  <Noise value="42.451363"/>
  <Moral value="100.0"/>
  <Description>Test</Description>
  <Distance value="0.0"/>
  <Rating value="0.0"/>
  <Tags>
      <Tag value="Verkehr"/>
      <Tag value="Baustelle"/>
  </Tags>
  <LocationSource longitude="7.000000" latitude="51.000000"/>
</Measure>
```
AddMeasures: Request

Attribute: Series

```xml
<MeasureSeries id="134">
  <Description>Test</Description>
  <Title value="" />
  <Time value="2011-02-16 18:03:25" />
  <Tags>
    <Tag value="Verkehr"/>
    <Tag value="Baustelle"/>
  </Tags>
  <Measure id="6" type="manual">
    ...
  </Measure>
  <Measure id="7" type="automatic">
    ...
  </Measure>
</MeasureSeries>
```
AddMeasures: Response

Attribute: AddMeasures Response

```xml
<Response type="AddMeasures">
  <Successful>
    <Measure localid="0" serverid="123"/>
    ...
  </Successful>
  <Error type="">
    <Measure localid="1"/>
    ...
  </Error>
</Response>
```
GetMeasures: Request

---

**Attribute: GetMeasures Request**

```xml
<Request type="GetMeasures">
  <Filter>
    <Time from="" to="" />
    <BBox top="" bottom="" left="" right="" />
    <Tags mode="all/some">
      <Tag value="" />
      ...
    </Tags>
    <Description contains="" />
    <MaxResults value="" />
    <Geocode value="" />
  </Filter>
</Request>
```
GetMeasures: Response

<table>
<thead>
<tr>
<th>Attribute: GetMeasures Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;Response type=&quot;GetMeasures&quot;&gt;</code></td>
</tr>
<tr>
<td><code>  &lt;Measures&gt;</code></td>
</tr>
<tr>
<td><code>   ...</code></td>
</tr>
<tr>
<td><code>  &lt;/Measures&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/Response&gt;</code></td>
</tr>
</tbody>
</table>