SeismoCloud: Earthquake Early Warning with a Network of Low Cost Seismometers

Salvatore Barba
Istituto Nazionale di
Geofisica e Vulcanologia
Rome, Italy
salvatore.barba@ingv.it

Valerio De Rubeis
Istituto Nazionale di Geofisica e
Vulcanologia
Rome, Italy
valerio.derubeis@ingv.it

Emanuele Panizzi
Sapienza Università di Roma,
Computer Science Department
Rome, Italy
panizzi@uniroma1.it

CHITALY'15, September 28–30, 2015, Rome, Italy

Abstract
We describe SeismoCloud, an Earthquake Early Warning (EEW) system based on a network of low cost seismometers. The aim of this system is to collect crowdsourced localized vibration data from the seismometer network and detect a possible earthquake when vibrations are related in both time and space. If the estimated intensity of the detected earthquake is higher than a given threshold, an alarm is sent to users.

Author Keywords
Mobile app; Earthquake; Early Warning; EEW; Seismometer; Usability; Crowdsourcing; Android; iOS;

ACM Classification Keywords
Categories and subject descriptors: H.5.2 [Information interfaces and presentation]: User interfaces --- Interaction styles; J.2 Physical sciences and engineering --- Earth and atmospheric sciences.

Related work
Other systems have been developed or are currently under design like [1],[2]. Nevertheless our system has gone through a User Centered Design process, exploits mobile as well as fixed seismometers,

allows direct connection of seismometers to the network without any extra hardware, exposes APIs and source code for seismometers, and offers a wide range of information to the users to reward their collaboration.

The SeismoCloud System

The system is composed of seismometers, mobile apps, a server, and a web site. We have developed a set of APIs that seismometers use to connect to the server over a standard http connection. The APIs allow each seismometer to inform the server when they are active, and send their location. When a seismometer detects a vibration, it reports it to the server via the APIs.

The server continually monitors all the received vibration information and relates it to the active seismometers in the area. We developed an algorithm that is able to detect an earthquake when a number of seismometers report a vibration in the same area. The algorithm estimates the intensity of the earthquake, in the MCS scale, and sends an alarm to users that are located in places where the earthquake can be dangerous, i.e. where the estimated intensity is greater or equal than the fourth grade in the MCS scale. These users will in fact receive a push notification on their smartphones. Phone notifications can be delivered several seconds before the seismic wave reaches the user, so allowing him/her to take any decisions (e.g. exit from a building) earlier.

We have developed two type of seismometers: a fixed seismometer, and a mobile one. The fixed seismometer is based on an Arduino-like microcontroller, namely an Intel Galileo, and on a low cost 3-axes accelerometer. The microcontroller polls the accelerometer data and, if higher than an empirically estimated threshold, connects to the server and reports the vibration using the API. Using the app, it is possible to setup the seismometer and to monitor it both locally and remotely.

The mobile seismometer is integrated in the app, as described below.

The SeismoCloud website delivers information about the data collected in crowdsourcing, in a visual form. Real time data including active seismometers and vibrations are reported on maps; past data are showed in Time Lapse; correlation of seismometer vibration with (low intensity) earthquakes are shown dynamically on maps as well; many graph visualizations are used to display device connections and activity; a simple gamification approach is used to incentivize seismometer installation and usage; last but not least, on the website it is possible to run simulations of the main past Italian earthquakes.

The SeismoCloud Mobile Application

We have developed two apps, for iOS and Android. The Android app includes the integrated seismometer, while the iOS one is currently only used for the early warning and to provide other crowdsourced, manually input, information: users can report their experience after an earthquake by answering a questionnaire.

The apps ask the user for their province of residence. This information will be used in case of EEW: according to the estimated intensity at the main town in the province, the system can send a notification to all the users that have selected that province. This coarse grained information is enough for the early warning and avoids any privacy issues. We decided not to require any user registration/login for privacy reasons as well.

The app shows in a list all the seismometers that the user may have: the internal one plus any fixed seismometer that the user owns, plus any one that is available in the current wifi local network, or that the user saved when being in their local network.

The apps were designed and developed in a user centered approach with the help of students in the HCI course at Sapienza University. In fact, for each of the planned functions and services, we ran many questionnaires and interviews, designed storyboards, designed and tested paper prototypes, and developed the code incrementally while testing with users in an agile-UCD fashion.

App-embedded Seismometer

The Android app turns the smartphone into a mobile seismometer, which works while the smartphone is in stand-by. The app uses the internal accelerometer of the smartphone to detect vibrations and connects to the server to report them. Localization services of the smartphone are used to inform the server about the location of the seismometer.

We worked hard on the interface, not only to make it as usable as possible, but also to discriminate between movements of the smartphone due to normal usage from vibrations due to local causes or possibly to earthquakes. From our point of view, when the smartphone is in use in the hands of its owner, or when it is in stand by but it is moving, e.g. in a vehicle, detecting vibrations is useless. So, we activate the seismometer only after a few seconds since the smartphone is stable on a flat surface, for example a table. This approach not only reduces the noise from the seismometer that could result in false positives, i.e. wrong earthquake detections, at the server level, but it also reduces battery consumption on the smartphone, as the accelerometer is a quite energy-consuming device.

References

[1] Matthew Faulkner, Robert Clayton, Thomas Heaton, and K. Mani Chandy et al. Community sense and response systems: Your phone as quake detector, Communication of the ACM, 2014.

[2] Francesco Finazzi, and Alessandro Fassò. Real-time detection of earthquakes through a smartphone-based sensor network. GRASPA 2015, Bari, Italy June 2015 **http://www.graspa.org/wp-content/uploads/2015/06/Graspa2015_Proceedings.pdf**