



# Détecter et monitorer les séismes grâce aux capteurs embarqués dans les smartphones

Anne-Marie Lesas

## ► To cite this version:

Anne-Marie Lesas. Détecter et monitorer les séismes grâce aux capteurs embarqués dans les smartphones. INFORSID 2016, May 2016, Grenoble, France. 2016, <<http://inforsid.fr/Grenoble2016>>. <hal-01362224>

**HAL Id: hal-01362224**

**<https://hal.archives-ouvertes.fr/hal-01362224>**

Submitted on 8 Sep 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

---

# Détecter et monitorer les séismes grâce aux capteurs embarqués dans les smartphones

Anne-Marie Lesas<sup>1,2</sup>

1. Université de Nice – Sophia-Antipolis, MBDS,  
1645, route des Lucioles, Sophia Antipolis, 06410 Biot, France
2. Aix Marseille Université, LSIS UMR 7296,  
13397 Marseille, France  
amlesas@yahoo.fr

---

*RESUME.* Dans cet article, nous présentons un algorithme original de détection des séismes utilisant le capteur accéléromètre embarqué dans le smartphone. Nos travaux s'inscrivent dans le cadre du projet SISMAPP d'études et de recherche proposé aux étudiants du MBDS ([www.mbds-fr.org](http://www.mbds-fr.org)) de l'Université de Nice – Sophia-Antipolis (UNS) en partenariat avec le Centre Sismologique Euro-Méditerranéen (CSEM, [www.emsc-csem.org](http://www.emsc-csem.org)). Notre objectif est de montrer que la connectivité du smartphone et les capteurs embarqués pourraient devenir de facto une station sismique mobile pouvant être aisément déployée à grande échelle et à faible coût.

*ABSTRACT.* In this paper, we present an original algorithm for detecting earthquakes using the accelerometer sensor embedded into the smartphone. Our work fits within SISMAPP research project proposed to MBDS ([www.mbds-fr.org](http://www.mbds-fr.org)) students at University of Nice – Sophia-Antipolis (UNS) in partnership with the European-Mediterranean Seismological Centre (EMSC, [www.emsc-csem.org](http://www.emsc-csem.org)). We aim to show that the smartphone connectivity and embedded sensors could turn into a mobile seismic station that can be easily and widely deployed at low cost.

*MOTS-CLES :* Surveillance des tremblements de terre, Application mobile, Capteurs embarqués, Big Data.

*KEYWORDS:* Earthquake monitoring, Mobile application, Embedded sensors, Big Data

---

## 1. Introduction

En moins d'une décennie, les téléphones mobiles ont évolué jusqu'à rivaliser avec les ordinateurs de bureau. De par sa popularité (près de 2 milliards de

smartphones vendus dans le monde fin 2015 <sup>1</sup>) et sa connectivité (réseau cellulaire, Wi-Fi, Bluetooth), le smartphone qui embarque toutes sortes de capteurs dont l'accéléromètre et le GPS, permet d'analyser et de collecter des informations géolocalisées et horodatées. Au mois d'août 2014, l'enregistrement d'un séisme de magnitude 6 survenu dans le Sud de Napa (Californie, US) a été capturé par hasard à 38 km de l'épicentre avec l'application « MyShake » développée dans le laboratoire de sismologie de l'université de Berkeley ([www.seismo.berkeley.edu](http://www.seismo.berkeley.edu)). Les enregistrements du séisme de Napa avec une application mobile ont démontré que les données en provenance des capteurs du smartphone sont d'une grande qualité lorsque l'appareil est positionné sur un support stable, et qu'elles pourraient être exploitées par les sismologues.

Depuis 2013 les étudiants du Master Mobilité Bases de Données et intégration de Systèmes (MBDS) de l'Université de Nice – Sophia-Antipolis (UNS) contribuent au projet SISMAPP de prototypage d'outils numériques dédiés à une plateforme de gestion avant / pendant / après un séisme en partenariat avec le Centre Sismologique Euro-Méditerranéen (CSEM, [www.emsc-csem.org](http://www.emsc-csem.org)) et le suivi des membres du laboratoire de recherche en Géophysique GéoAzur ([www.geoazur.oca.eu](http://www.geoazur.oca.eu)) du Centre National de la Recherche Scientifique (CNRS) Unité Mixte de Recherche (UMR) de l'UNS.

Dans cet article, nous montrons comment une application mobile peut détecter et déclencher des alertes d'origine potentiellement sismiques et collecter des mesures en utilisant le capteur accéléromètre embarqué dans le smartphone. Les données collectées sont destinées à alimenter un entrepôt de données massives destiné à l'étude de modèles de prédictibilité : les travaux apparentés ainsi que les motivations pour le travail présenté dans cet article sont abordés dans la section 2. Dans la section 3, le lecteur est familiarisé avec les capteurs embarqués dans les smartphones. Nous décrivons notre algorithme basé sur le dépassement de seuils dans la section 4. À la fin de cet article, nous concluons en donnant quelques perspectives d'évolution de nos travaux.

## **2. Etat de l'art et motivations**

### ***2.1. Travaux apparentés***

Depuis 2008, le projet Quake Catcher Network (QCN, [www.qcn.stanford.edu](http://www.qcn.stanford.edu)), initié par les universités de Stanford et de Californie à Riverside, utilise le logiciel libre de calcul réparti « Berkeley Open Infrastructure for Network Computing » (BOINC) développé par l'université de Berkeley en Californie (USA). BOINC est une plateforme open source hébergeant des projets scientifiques dont le principe est d'utiliser une partie des ressources des systèmes informatiques des utilisateurs bénévoles connectés au réseau. La fiabilité de l'interprétation des données

---

1. Source Cbnews : <http://www.cbnews.fr/digital/pres-de-2-milliards-de-smartphone-dans-le-monde-fin-2015-a1016742>

remontées par le réseau QCN a été démontrée avec des données collectées par des capteurs inertiels fixés dans les parties inférieures des bâtiments à Christchurch en Nouvelle Zélande (Cochran et al., 2011) : ces données comparées aux données des stations sismiques GeoNet ([www.geonet.org.nz](http://www.geonet.org.nz)) de la région lors du tremblement de terre survenu le 18 octobre 2011, ont montré que les mouvements du sol observés par GeoNet et QCN avaient des amplitudes comparables à une distance donnée de l'épicentre du séisme.

Le CSEM propose deux programmes dans le cadre de QCN : le programme « QCN Sensor Monitoring Program » qui fonctionne en mode « déclenchement », c.-à-d. que l'information n'est transmise que lorsque l'accélération est significativement plus élevée que celle des secondes précédentes en se basant sur l'algorithme « Short-Term Average/Long-Term Average » (STA/LTA) (Cuenot, 2003), et le programme « QCN Continual Monitoring Program » fonctionnant en mode « continu » pour collecter des échantillonnages de données.

La plupart des applications mobiles dédiées aux tremblements de terre s'appuient sur les données en provenance des organismes de surveillance sismique tels que l'US Geological Survey (USGS), le CSEM, ou l'Earthquake Early Warning (Japan Meteorological Agency) dont les informations qui proviennent majoritairement de stations sismiques fixes implantées sur toute la surface de la planète sont accédées à la demande en temps décalé ou semi-réel (requêtes continues avec ou sans notification d'alerte). Les applications mobiles utilisant les capteurs embarqués des smartphones ont émergé récemment avec le prototype japonais i-Jishin (Naito et al., 2012) (Naito et al., 2013) et le projet « iShake » de Berkeley (Reilly et al., 2013). En janvier 2014, QCN a publié son application mobile utilisant l'accéléromètre du smartphone et fonctionnant avec la version Android de BOINC toujours avec un déclenchement au repos (en charge). Cette application ne consomme que 1 à 5 % des ressources (CPU) de l'appareil et seuls quelques kilo-octets de données sont transmis quotidiennement sur le réseau.

Des applications mobiles telles que l'application Japonaise « Yurekuru Call » proposée par le centre Earthquake Early Warning sont spécialisées dans la propagation d'alertes précoces. Des applications ou des réseaux sociaux (par exemple Facebook) proposent également un bouton « je suis en vie » qui permet aux familles d'être rassurées en cas de sinistre. En outre, il existe des applications mobiles (e.g. « LastQuake » proposée par le CSEM) et des portails web permettant une contribution participative (« crowdsourcing ») du grand public pour partager des témoignages, des photos et des vidéos...

### **2.1. Motivations pour nos travaux**

La détection basée sur les capteurs embarqués dans les smartphone est un nouveau champ d'étude, par exemple, une étude scientifique menée par plusieurs chercheurs (Minson et al., 2015), mentionne qu'un séisme pourrait être détecté en 5 secondes avec seulement une participation communautaire de 0,2 % des utilisateurs de mobiles (~5000 appareils) dans un rayon de 5 km autour de l'épicentre d'une

zone fortement peuplée comme San Francisco ou San Jose (aux USA). L'article précise que la détection n'est possible que lors des tremblements de terre de forte magnitude : la précision des capteurs mobiles limite la détection à des séismes de magnitude supérieure à 2. Cependant, les technologies évoluent plus vite que les usages et les limitations d'aujourd'hui ne doivent pas restreindre la recherche.

« *La région Provence-Alpes-Côte d'Azur est la région de France métropolitaine la plus soumise au risque sismique* » (Virieux, 2014). Déjà sensibilisés par le tremblement de terre dévastateur survenu à Haïti en 2010 et ayant entraîné la destruction des locaux du Master MBDS délocalisé, la contribution du MBDS au développement d'outils numériques autour du projet SISMAPP de gestion des séismes coulait de source. Le travail présenté dans cet article sur la détection de mouvements d'origine potentiellement sismique à partir des mesures collectées sur le capteur accéléromètre embarqué dans le smartphone s'inscrit dans le cadre d'un ensemble de travaux réalisés autour du projet SISMAPP présenté dans un rapport antérieur (Lesas et al., 2014). Le projet SISMAPP a pour double finalité de permettre aux étudiants d'exploiter leurs compétences dans la mise en pratique des enseignements théoriques avec à terme, l'implémentation et le déploiement d'une plateforme régionale dédiée à la recherche et à l'étude des modèles de prédictibilité en partenariat avec les chercheurs en sismologie et les industriels spécialisés dans le traitement des données en provenance des capteurs.

Dans cet article nous nous focalisons sur la faisabilité de l'utilisation des capteurs du smartphone dans la détection des séismes à travers la présentation de notre implémentation d'un prototype preuve de concept. La valeur ajoutée de notre contribution partiellement présentée et centrée ici sur l'analyse des données en provenance du capteur accéléromètre embarqué tient essentiellement dans la flexibilité ainsi qu'à la possibilité de collecter non seulement localement mais aussi de transmettre les données à un serveur distant ce qui en fait un appareil de mesure portable particulièrement adapté pour la recherche. En effet, contrairement aux autres applications, la solution proposée permet de tester plusieurs configurations et d'enregistrer des notifications d'alerte potentiellement sismiques ainsi que des mesures collectées dans un laps de temps prédéfini avant et après les alertes. Les données sont enregistrées dans une (micro) base de données (BDD) et/ou envoyées vers un serveur (dont l'URL est également paramétrable) au format texte de type clé-valeur (JSON). Le serveur met simplement à disposition un web service de type « Representational State Transfer » (REST) acceptant du texte sans contrainte de format de façon à permettre les évolutions ultérieures sans aucun impact sur l'interface : cette architecture a été conçue dans le but de pouvoir recueillir des données hétérogènes de différents formats et de différentes origines pouvant faire l'objet d'un traitement map reduce (Hadoop) toujours dans le cadre du projet SISMAPP.

### 3. Les capteurs

Les capteurs sont des instruments de mesure qui transforment une information d'ordre physique, chimique ou organique en donnée manipulable et interprétable.

Les capteurs sismiques sont des capteurs qui mesurent des inerties par l'accélération perçue sur un axe de l'espace à une fréquence d'échantillonnage donnée.

### 3.1. Les capteurs MEMS <sup>2</sup>

Les capteurs MEMS sont des systèmes micro-électro-mécaniques économiques de très petite taille (tendant vers les nanotechnologies) qui utilisent des propriétés électromagnétiques, thermiques, optiques, chimiques ou biologiques pour effectuer et collecter une mesure.

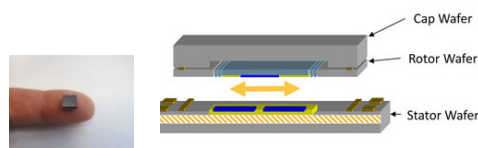


Figure 1. Capteur sismique MEMS Hewlett Packard (Source HP <sup>3</sup>)

Les capteurs sismiques MEMS (e.g. fig. 1) sont généralement constitués d'accéléromètres qui mesurent les variations du champ de force exercé sur une petite masse témoin (seconde loi du mouvement de Newton :  $F = m.a$ ), appelée aussi « masse sismique », se mouvant entre des lamelles fixes à la surface d'un circuit intégré en silicium. Il est ainsi possible de mesurer des accélérations horizontales et verticales en fonction de la position. Les capteurs triaxiaux retournent une valeur scalaire ( $x, y, z$ ) qui correspond à la magnitude du vecteur d'accélération dans l'espace tridimensionnel.

### 3.2. Les capteurs de mouvements embarqués dans les smartphones

La plupart des smartphones embarquent en standard des capteurs inertiels mesurant les mouvements sur les 3 coordonnées spatiales relatives à l'appareil mobile (axe vertical, frontal et latéral). La restitution des mouvements dans le système géodésique se fait par translation des coordonnées (à l'aide d'un quaternion ou d'une matrice de rotation). L'inertie est mesurée par des capteurs physiques tels que l'accéléromètre qui renvoi un vecteur d'accélération en  $m.s^{-2}$  (gravité incluse), le magnétomètre duquel est issu un vecteur de direction en micro-tesla <sup>4</sup>, le gyroscope mesurant l'effet de Coriolis en rad/s restitué sous la forme d'un vecteur de vitesse angulaire par rapport aux axes, et le capteur de pression atmosphérique mesurée en hectopascals (millibars).

2. Micro-Electro-Mechanical Systems

3. <http://www8.hp.com/us/en/hp-news/press-kit.html?id=1096990>

4. Pour avoir une mesure correcte dans le temps, il faut normaliser le signal en prenant un échantillonnage de la vitesse angulaire, c.-à-d. la racine de la somme des carrés

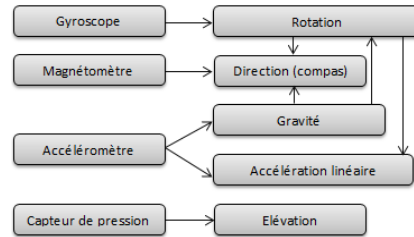


Figure 2. Capteurs fusion (Android)

La fusion des informations en provenance des capteurs physiques fournit une couche d'abstraction logicielle du matériel dont les relations illustrées par la fig. 2 montrent les dépendances de la source vers son utilisation entre les capteurs physiques (à gauche) et les capteurs « fusion » (à droite) issus de calculs combinant des mesures relevées sur les capteurs sources. Par exemple, le vecteur de rotation indique la direction avec l'azimut (autour de l'axe vertical), le pitch (autour de l'axe frontal) et le roll (autour de l'axe latéral) en degrés (fig. 3), la répartition de la gravité ( $g = 9,81 \text{ m.s}^{-2}$ ) sur les axes est mesurée  $\text{m.s}^{-2}$  par combinaison des données du vecteur d'accélération et celui de la direction, l'accélération linéaire est calculée en soustrayant la gravité à l'accélération et l'élévation en mètres est déduite du capteur de pression. Les capteurs de mouvement sont présents sur tous les smartphones modernes, avec cependant des APIs qui diffèrent au niveau de la couche logicielle/middleware et du prétraitement des données (i.e. filtrage et fusion des données), par exemple, l'API Windows Phone n'est pas utilisable dans un service de tâche de fond<sup>5</sup>.

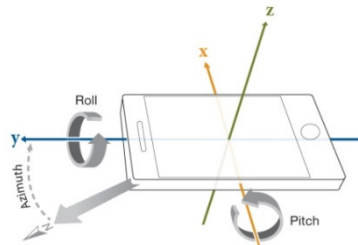


Figure 3. Axes des coordonnées géospatiales du smartphone<sup>6</sup>

5. Source de l'image : [http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh202962%28v%3Dvs.105%29.aspx#BKMK\\_UnsupportedAPIs](http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh202962%28v%3Dvs.105%29.aspx#BKMK_UnsupportedAPIs)

6. Source : <http://www.mathworks.com/matlabcentral/fileexchange/screenshots/9373/original.jpg>

### 3.3. Sources d'erreurs et correction des mesures

L'une des préoccupations principales dans la manipulation des capteurs concerne l'élimination du bruit :

- l'accéléromètre présente un décalage constant entre la valeur produite et la valeur réelle qui se cumule au fil du temps et croît de façon quadratique lors des doubles intégrations,
- le décalage dans les données du gyroscope provoque une erreur angulaire qui croît linéairement dans le temps, mais une erreur du facteur d'échelles dans le calibrage, peut ajouter une dérive aux calculs basés sur les mesures faussées se répercutant dans les capteurs fusion,
- le magnétomètre est soumis aux interférences provoquées par la contamination magnétique environnante (proximité de matériaux ferreux)...

Mais la dérive engendrée par ces erreurs peut être atténuée significativement après détermination du coefficient de décalage (Shala et al., 2011) en fonction du modèle de capteur. La fusion des données multi-capteurs implémente également des filtres issus du traitement du signal (e.g. passe-bas, Kalman, Butterworth) qui permettent d'éliminer le bruit et de compenser des dérives.

## 4. Le service mobile SISMAPP

Le service mobile SISMAPP (Android) analyse en temps réel les données en provenance du capteur accéléromètre triaxial embarqué dans le smartphone afin de détecter des mouvements susceptibles de provenir d'un tremblement de terre. Pour ne pas impacter la consommation de la batterie (cf. tab. 1) et limiter la consommation des données, le service n'est déclenché que lorsque l'appareil est en charge et qu'il est connecté à un réseau Wi-Fi. L'analyse n'est déclenchée que lorsque l'état stationnaire du mobile est détecté.

*Tableau 1. Consommation de batterie supplémentaire due à l'utilisation des capteurs et du GP*

Utilisation en continu (Google Nexus S)	Durée pour 1% de conso. batterie	Durée pour 100% de conso. batterie	Conso. sup. (%)
Normale (78% de batterie)	00:03:28	05:46:40	-
Accéléromètre	00:03:01	05:01:40	+13%
Accéléromètre + GPS	00:01:38	02:43:20	+53%

Le service SISMAPP s'inspire du principe de détection d'événements par franchissement de seuils couramment utilisé dans les systèmes d'alerte. Cette technique permet de déterminer efficacement une anomalie sur un réseau distribué suffisamment dense (Cuenot, 2003). L'exécution en tâche de fond se base sur différents paramètres que l'utilisateur peut modifier pour adapter la sensibilité en



fonction du résultat attendu : fréquence du capteur, accélération minimum pour un état stationnaire, accélération maximum pour un état stationnaire, variance minimum pour un état stationnaire, durée du temps de glissement des données analysées et collectée. Le paramétrage permet en outre de modifier les options de collecte et de notification : adresse du serveur et des web services, activation/désactivation du service ou de l'envoi des données, durée de la collecte, délais entre deux notifications, intervalle entre les localisations.

La méthode du franchissement de seuil sur une période est illustrée par l'algorithme (5) où les  $a_t$  sont les accélérations mesurées à l'instant  $t$  et  $\delta(M)$  est le différentiel d'accélération qui déclenche le traitement :

$$\forall t > t_0, a \in [a_{t_0} \dots, a_{t_n}], \text{ si } |a_t - a_{t-1}| > \delta(M) \rightarrow \text{Traitement} \quad (5)$$

#### 4.1. Analyse et traitement des données en provenance de l'accéléromètre

Notre adaptation de la méthode du franchissement de seuil s'exécute sur une période glissante paramétrable très courte (~1s) dont le déclenchement est proposé dans l'algorithme 1 : lors du lancement de l'application, le service s'abonne aux événements de connexion et déconnexion d'alimentation sur secteur, de changement de connectivité réseau (Wi-Fi), au service de localisation<sup>7</sup>, et à l'événement d'activation et de désactivation du service accessible dans le setup de l'application. Lors de la réception des événements, la mise à jour d'indicateurs booléens conditionne le démarrage ou l'arrêt du traitement d'analyse et de collecte des données en provenance de l'accéléromètre.

##### Algorithme 1. Abonnement aux événements

---

```

1: /* Abonnement aux événements qui conditionnent le traitement */
2: AbonnerEvénements (alimentation, connectivitéRéseau, localisation, activationService)
3:
4: /* Gestion du traitement en fonction des événements */
5: RéceptionEvénement(Evénement)
6: {
7:     Enregistrer l'état lié à l'événement
8:     si localisation alors
9:         /* Mise à jour de la localisation avec l'événement « localisation » */
10:        Mettre à jour la latitude et la longitude
11:    sinon
12:        si (enCharge  $\wedge$  connectéAuWi-Fi  $\wedge$  serviceActivé) alors
13:            /* Si les conditions sont réunies, l'analyse des données est démarrée */
14:            début glissement  $\leftarrow$  horodatage
15:            durée collecte  $\leftarrow$   $\emptyset$ 
16:            maxX  $\leftarrow$  0
17:            maxY  $\leftarrow$  0

```

---

<sup>7</sup> GPS et GPS fusion disponibles en fonction du réseau et de la couverture

```

18:         maxX ← 0
19:         stationnaire ← faux
20:         collection ← faux
21:         DémarrerLectureAccéléromètre(fréquence)
22:     sinon
23:         /* Sinon, l'analyse est stoppée */
24:         StopperLectureAnalyseAccéléromètre()
25:     finsi
26: finsi
27: }

```

---

La lecture des données de l'accéléromètre se présente aussi sous la forme d'un abonnement aux événements de l'API capteurs à la fréquence choisie (cf. tab. 2) :

Tableau 2. Fréquence de lecture du capteur

Option de fréquence de lecture	Délais entre chaque lecture
SENSOR_DELAY_NORMAL	0,2 s
SENSOR_DELAY_UI	0,06 s (convient au rafraîchissement de l'affichage)
SENSOR_DELAY_GAME	0,02 s (convient aux jeux)
SENSOR_DELAY_FASTEST	Pas de délais : fréquence maximum

Les mesures des 3 axes x, y, et z du capteur accéléromètre (linéaire<sup>8</sup>) horodatées sont réceptionnées dans la méthode de rappel dans laquelle elles sont analysées et traitées selon l'algorithme 2 :

Algorithme 2. Analyse et traitement des données en provenance de l'accéléromètre

```

1: /* Traitement d'analyse des données de l'accéléromètre */
2: AnalyseValeursAccéléromètre(valeurs courantes de l'accéléromètre, horodatage)
3: {
4:     x, y, z ← valeurs courantes de l'accéléromètre
5:     collecte ← {x, y, z, horodatage} ∪ collecte
6:
7:     /* Glissement de l'analyse sur les valeurs lues pendant la durée paramétrée */
8:     si (durée collecte > paramètre glissement) alors
9:
10:         /* Vérification de l'état stationnaire : */
11:         si ¬stationnaire ∧ ¬collection alors
12:             stationnaire ← VérifierStationnaire(collecte)
13:         finsi
14:
15:         /* Lorsque l'état stationnaire a été détecté : */

```

---

8. Auquel on a soustrait la gravité.

```

16:      si stationnaire alors
17:          /* Vérification du dépassement de seuil sur données collectées */
18:          collection ← VérifierAlerte(collecte)
19:      finsi
20:
21:      /* Lorsque le dépassement de seuil a été détecté : */
22:      si collection alors
23:          id_collection ← id_collection + 1
24:          /* Envoyer notification d'alerte à l'utilisateur */
25:          TraiterAlerteUtilisateur(horodatage)
26:
27:          /* Si le serveur est configuré, envoi de la notification au serveur */
28:          TraiterAlerteServeur(horodatage, id_collection, dernière localisation)
29:
30:          /* Vérifier si la durée de collecte paramétrée est atteinte */
31:          si (durée collecte – paramètre glissement) > paramètre collection alors
32:              /* Traiter la collection et faire une pause */
33:              collection ← faux
34:              TraiterCollection(id_collection, collecte)
35:              TraiterPause(horodatage)
36:          finsi
37:      sinon
38:          /* Supprimer les 1ères valeurs collectées pour ne garder */
39:          /* que le temps de glissement paramétré et mettre à jour */
40:          /* l'horodatage du début de la collecte... */
41:          TraiterGlissement(collecte)
42:      finsi
43:  finsi
44:
45:      durée glissement ← (horodatage – début glissement)
46:  }

```

---

L'algorithme 2 montre que les données horodatées des valeurs mesurées (x, y, et z) sont collectées dans une liste sur une période glissante ; le dépassement de seuil déclenchant une alerte et la sauvegarde de la collecte n'est vérifié qu'après détection de l'état stationnaire (cf. algo. 3) : s'il y a un dépassement de seuil (algo. 4), une notification d'alerte est déclenchée (dans la barre des notifications) et selon le paramétrage, envoyée au serveur. La collection est poursuivie sur la durée paramétrée avant d'être traitée selon l'algorithme 5. Les variables (état stationnaire et maximums des moyennes collectées à l'état stationnaire) sont réinitialisées, et le service est mis en pause pendant la durée paramétrée (ce qui évite les notifications intempestives et limite les dérives du capteur dans le temps). Si aucun franchissement du seuil d'alerte n'est détecté, les premières mesures sont supprimées de façon à ne conserver dans la collecte que la période de glissement.

### Algorithme 3. Vérification de l'état stationnaire

---

```

1:  /* Vérification de l'état stationnaire */
2:  VérifierStationnaire(collecte)

```

```

3: {
4:    $\forall x, y, z \in \text{collecte} :$ 
5:   stationnaire  $\leftarrow$  faux
6:    $\text{avg}X \leftarrow \sum x > 0 / \text{nb}(x > 0)$ 
7:    $\text{avg}Y \leftarrow \sum y > 0 / \text{nb}(y > 0)$ 
8:    $\text{avg}Z \leftarrow \sum z > 0 / \text{nb}(z > 0)$ 
9:
10:  /* comparaison des valeurs au seuil minimum déclaré dans le setup */
11:  si ( $\text{avg}X < \text{minAccel}$ )  $\vee$  ( $\text{avg}Y < \text{minAccel}$ )  $\vee$  ( $\text{avg}Z < \text{minAccel}$ ) alors
12:
13:    /* si au moins une des valeurs moyennes mesurées sur un axe est */
14:    /* inférieure à la valeur minimum, alors on considère l'état stationnaire */
15:    /* et on mémorise les valeurs moyennes maximum de l'état stationnaire */
16:    stationnaire  $\leftarrow$  vrai
17:     $\text{max}X \leftarrow \text{avg}X > \text{max}X ? \text{avg}X : \text{max}X$ 
18:     $\text{max}Y \leftarrow \text{avg}Y > \text{max}Y ? \text{avg}Y : \text{max}Y$ 
19:     $\text{max}Z \leftarrow \text{avg}Z > \text{max}Z ? \text{avg}Z : \text{max}Z$ 
20:  fin
21: }

```

---

Afin d'exclure d'éventuelles valeurs parasites, l'état stationnaire est vérifié sur la moyenne de toutes les données collectées (période glissante) : si la moyenne mesurée sur un des axes est inférieure au seuil minimum pour un état stationnaire, c'est une condition suffisante pour considérer l'état stationnaire car cela permet de s'absoudre d'un éventuel décalage du zéro d'un des axes. Les moyennes maxima sont mémorisées pour calculer le delta d'accélération seuil d'alerte par rapport au delta configuré dans le setup (cf. algo. 4).

Remarque : Parce que la détection se fait en partant d'un état stationnaire et qu'une accélération positive est normalement suivie d'une accélération négative, seules les accélérations positives sont prises en compte.

#### Algorithme 4. Vérification du dépassement de seuil pour une « alerte »

---

```

1: /* Vérification d'une alerte par une dérivée de la méthode du dépassement de seuil */
2: VérifierAlerte(collecte)
3: {
4:    $\forall x, y, z \in \text{collecte} :$ 
5:    $\text{delta}X \leftarrow (\sum x > 0 / \text{nb}(x > 0)) - \text{max}X$ 
6:    $\text{delta}Y \leftarrow (\sum y > 0 / \text{nb}(y > 0)) - \text{max}Y$ 
7:    $\text{delta}Z \leftarrow (\sum z > 0 / \text{nb}(z > 0)) - \text{max}Z$ 
8:
9:   /* Si le delta moyen est supérieur à une valeur maximum paramétrée */
10:  /* c'est que le mouvement qui l'a provoqué est trop brusque être */
11:  /* d'origine sismique... */
12:  si ( $\text{delta}X > \text{maxAccel}$ )  $\vee$  ( $\text{delta}Y > \text{maxAccel}$ )  $\vee$  ( $\text{delta}Z > \text{maxAccel}$ ) alors
13:
14:    stationnaire  $\leftarrow$  faux
15:

```

```

16:      /* Il faut donc stopper une éventuelle collection */
17:      si collection alors
18:          /* Il faut donc stopper une éventuelle collection */
19:          collection ← faux
20:
21:          /* Mais la collection est tout de même traitée avant la pause */
22:          TraiterCollection(id_collection, collecte)
23:      finsi
24:      TraiterPause(horodatage)
25:  sinon
26:      /* Sinon, si le delta moyen est supérieur au delta d'accélération paramétré */
26:      /* pour une alerte, la collection pour sauvegarde démarre */
27:      si (deltaX > deltaAccel) ∨ (deltaY > deltaAccel) ∨ (deltaZ > deltaAccel) alors
28:          collection ← vrai
29:          début collection ← horodatage
30:      finsi
31:  finsi
32: }

```

---

Le déclenchement d'une alerte vérifie deux conditions (cf. algo. 4) : (i) que les deltas d'accélération moyens par rapport aux maxima moyens calculés à l'état stationnaire dans l'algorithme 3 ne dépassent pas un seuil trop élevé (e.g. dû à une reprise en main de l'appareil par l'utilisateur), et (ii) qu'au moins l'un des deltas mesurés sur les axes x, y, ou z dépasse un delta d'accélération minimum de déclenchement. Lorsque le delta d'accélération calculé dépasse un seuil trop élevé et qu'une collection était en cours, la collecte est stoppée mais les données sont quand même traitées pour une sauvegarde locale et/ou l'envoi au serveur (cf. algo. 5) car le dépassement pourrait avoir été engendré par la chute de l'appareil lors d'un séisme.

*Algorithme 5. Traitement d'une collection de données consécutive à une alerte*

```

1:  /* Vérification d'une alerte par une dérivée de la méthode du dépassement de seuil */
2:  TraiterCollection(id_collection, collecte)
3:  {
4:      si envoiServeur ∧ serveurRenseigné() alors
5:          envoyerCollecte(id_collection, collecte, début glissement,
6:                          début collection, horodatage, dernière localisation,
7:                          type de capteur, fréquence, modèle de mobile)
8:      finsi
9:      enregistrerLocalement(id_collection, collecte)
10: }

```

---

#### 4.2. Traitement des alertes et des mesures collectées

Nous avons mentionné précédemment que lors d'un séisme, il se peut que les réseaux soient saturés ou endommagés. Dans la perspective d'une alternative à

l'incapacité de contacter le serveur et dans un contexte de recherche, la sauvegarde locale permet d'analyser les données collectées à postériori.

Les notifications d'alertes qui déclenchent la collecte et les mesures sont stockées localement dans une BDD embarquée (SQLite) accessible dans un dossier créé par l'application SISMAPP dans la mémoire de stockage du mobile (cf. fig. 4).

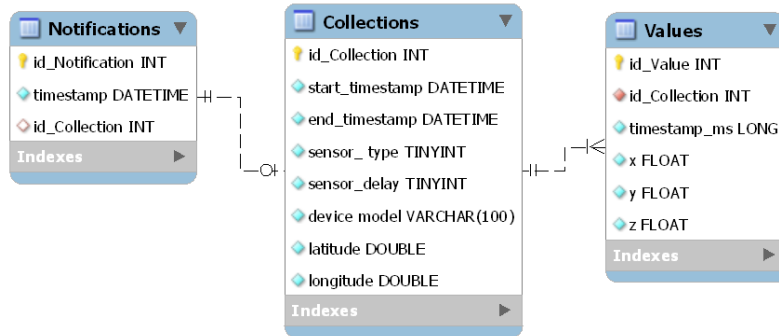


Figure 4. Modèle de la BDD locale du service SISMAPP

Les données collectées peuvent être supprimées à partir de l'application mobile ou bien récupérées pour être analysées (par exemple dans un tableur). Le temps  $t$  écoulé à l'occurrence  $n$  des valeurs collectées est donné en millisecondes ; il suffit alors de calculer en secondes les  $t_n$  d'un axe temporel horizontal démarrant à  $t_0$  tel que :  $t_0 = 0$  et  $t_n = (t(n) - t(0)) * 10^{-9}$  et de sélectionner les séries de valeurs  $x$ ,  $y$ , et  $z$  représentées en  $m.s^{-2}$  sur l'axe vertical : la fig. 5 montre la visualisation graphique des données où le pic d'accélération correspond à une vibration du mobile (obtenue par un test) ayant déclenché une alerte et la sauvegarde de la collection.

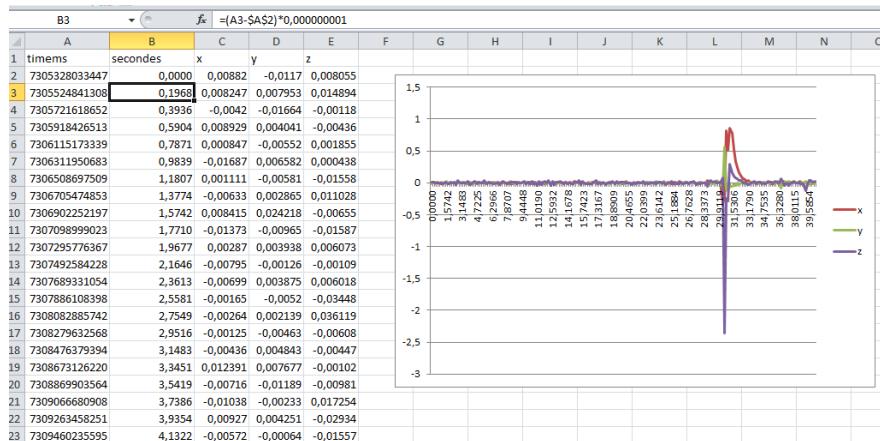


Figure 5. Visualisation graphique des données collectées

Si l'envoi au serveur a été paramétré et que le serveur est joignable, les notifications (en temps réel) et les collectes associées (à la fin de la collecte) peuvent être transmises dans le format clé-valeur (JSON) donné en exemple ci-après :

```
- Alertes : {"notifications":{"id_device":1,"id_notification":4,
"timestamp":"2015-03-15_10:01:59","latitude":43.7055339,"longitude":
7.2820472, "id_collection":4}}
```

```
- Mesures : {"collections":{"id_device":1, "id_collection":4,
"sensor_type":"Linear acceleration", "sensor_delay":3,
"start_timestamp":"2015-03-15_10:01:49", "end_timestamp":"2015-03-
15_10:02:19", "latitude":43.7055339, "longitude":7.2820472,
"device_model":"Android 21 - LGE NEXUS 5", "values":
[{"timestamp_ms":63382020019531, "x":0.0019194632768630981,
"y":1.209452748298645E-4, "z":0.0025548934936523438},
{"timestamp_ms":63382218017578, "x":0.0013000965118408203,
"y":0.006275080144405365, "z":0.02202129364013672},
{"timestamp_ms":63382416015625, "x":8.293241262435913E-4, "y":-
0.012210480868816376, "z":0.010548591613769531},
{"timestamp_ms":63382614013671, "x":-0.014498397707939148,
"y":0.00841280072927475, "z":0.02068042755126953},...]}
```

### 4.3. Expérimentation du prototype et limitations

#### 4.3.1. Détermination des seuils

Dans un premier temps, la détermination des seuils de déclenchement des alertes s'est faite empiriquement par l'observation sur plusieurs modèles de smartphones (Google/Samsung Nexus S, Samsung Galaxy S3, Microsoft/Nokia Lumia 720, et Google/LG Nexus 5) : ne disposant pas de table vibrante, l'expérience consistait à détecter une très légère vibration (seuil minimum) du support (provoquée manuellement) sur un laps de temps arbitraire suffisant (1s) après détermination de l'état stationnaire. Le seuil maximum a quant à lui été déterminé en mesurant plusieurs reprises naturelles du mobile. À ce stade, une approche plus rationnelle de la détermination des seuils est envisagée en collaboration avec des partenaires industriels (par exemple dans le cadre d'un déploiement expérimental).

#### 4.3.2. Perte de réseau en cas de destruction des infrastructures

Des applications comme « WhatsApp » ([www.whatsapp.com](http://www.whatsapp.com)), par exemple, ont démontré que la connectivité des smartphones avec le Bluetooth ou en Wi-Fi direct permet en agglomération de communiquer en pair-à-pair par réseau maillé lorsque les réseaux de télécommunication ne sont plus disponibles. Nous avons d'ailleurs prototypé les deux connectivités avec notre protocole qui permet de corroborer la présomption d'un séisme détecté par les capteurs du smartphone et d'alerter les utilisateurs lorsqu'un des utilisateurs a confirmé l'origine sismique. Cependant le système n'est fonctionnel que quand les appareils sont couverts par la connectivité utilisée.

#### 4.3.3. Identification des faux positifs et corrélation des données

Nous partons du principe qu'il n'est pas possible d'éliminer les faux positifs individuels, c'est la raison pour laquelle nous portons toute notre attention sur la

corrélation des événements et le traitement massif des données. Plusieurs pistes sont à envisager ; l'étude des données collectées (en temps réel ou pas), à partir du moment où elles sont horodatées et géolocalisées, permet une analyse à posteriori de laquelle pourraient être extraits des modèles de prédictibilité. Mais nous pensons également que la corrélation des sources d'informations et l'implantation d'un réseau de capteurs fixes communiquant avec le smartphone à des emplacements choisis pour leur faible exposition aux mouvements susceptibles d'engendrer des faux positifs est une solution adaptée pour les alertes précoces.

## 5. Conclusion

Dans cet article, nous avons montré comment grâce à un algorithme basé sur le franchissement de seuils, le smartphone avec ses capteurs inertiels embarqués et sa connectivité, peut contribuer avantageusement à la surveillance et la détection des séismes pour alerter les utilisateurs et collecter des données géolocalisées et horodatées pouvant être corrélées avec des événements sismiques dans l'objectif d'en déduire des modèles de prédictibilité.

Le service mobile SISMAPP est idéalement conçu pour dans un premier temps permettre la détermination d'un paramétrage efficace des seuils de sensibilité en laboratoire. Mais une phase de pilotage est nécessaire pour évaluer la fréquence des notifications et le volume des données pertinentes. L'exploitation du prototype SISMAPP dans le contexte de la recherche sismologique n'en est qu'à ses débuts, mais sa flexibilité en fait un outil de mesure portable qui pourrait trouver son utilité dans d'autres domaines, par exemple pour mesurer la résonance des ondes d'origine explosive (e.g. dans des carrières de minerais).

### Remerciements

*Nous tenons à adresser ici tous nos remerciements à Rémy Bossu à la Direction du CSEM et à M. Frédéric Roussel du CSEM pour leurs conseils avisés, ainsi que l'ouverture d'un web service sur un serveur du CSEM nous permettant de tester l'envoi des données. Nous remercions également Anne Deschamps, Damien Provitolo, et Emmanuel Tric du laboratoire GéoAzur pour leur intérêt et le suivi accordé au projet SISMAPP, ainsi que tous les étudiants qui ont participé au projet, le Pr. Serge Miranda pour avoir initié ce projet et le Pr. Omar Boucelma pour son encadrement et je remercie le comité de relecture et plus particulièrement le Pr. Florence Sedes pour ses critiques constructives et ses commentaires pertinents quant à la rédaction de cet article.*

### Bibliographie et références

- Bossu R. (2015). *Earthquake Risk Mitigation Using Social Media and Sensor-Based Citizens' Participation*, internal report, CSEM, 2015.
- Cochran E. S., Lawrence J. F., Kaiser A., Fry B., Chung A., Christensen C. (2011). Comparison between low-cost and traditional MEMS accelerometers: a case study from the M7.1 Darfield, New Zealand, aftershock deployment, *Annals Of Geophysics, Citizen*



- Empowered Seismology*, Special Section edited by R. Bossu and P.S. Earle, 2011, p.728-737.
- Cuenot O. (2003). *Les algorithmes de détection automatique d'ondes sismiques*, probatoire CNAM, 2003.
- Dolui K., Mukherjee S., Kanti Datta S. (2013). Smart Device Sensing Architectures and Applications, *International IEEE Computer Science and Engineering Conference (ICSEC), 2013*, Bangkok, Thailand.
- Faulkner M., Olson M., Chandy R., Krause J. (2011). The Next Big One: Detecting Earthquakes and other Rare Events from Community-based Sensors, *International Conference on Information Processing in Sensor Networks (IPSN), 2011*, Chicago, Illinois, USA.
- Kong Q. (2012). *Using Smartphones to Detect Earthquakes*, Berkeley Seismological Laboratory annual research report, University of California, 2012.
- Lesas A.M., Ardoin A., Cano S. (2014). SISMAPP : *Une station sismique mobile dans le smartphone*, rapport technique, MBDS, Université de Nice – Sophia-Antipolis, 2014.
- Minson S. E., Brooks B. A., Glennie C. L., Murray J. R., Langbein J. O., Owen S. E., Heaton T. H., Iannucci R. A., Hauser D. L. (2015). Crowdsourced earthquake early warning, *Science Advances*, vol. 1, n° 3, e1500036.
- Naito S., Azuma H., Senna S., Yoshizawa M., Nakamura H., Fujiwara H., Yoshida M. (2013). *On-site experiment of seismic monitoring network by utilization inside sensors of mobile terminal*, National research Institute for Earth science and Disaster Prevention, Conference paper, Japan Geoscience Union Meeting, 2013.
- Naito S., Azuma H., Senna S., Yoshizawa M., Nakamura H., Hao K. X., Fujiwara H., Hirayama Y., Yuki N., Yoshida M. (2013). Development and Testing of a Mobile Application for Recording and Analyzing Seismic Data, *Journal of Disaster Research*, vol. 8, n° 5, p. 990-1000.
- Olson M., Liu A., Faulkner M. (2011). Rapid Detection of Rare Geospatial Events: Earthquake Warning Applications, *Distributed Events-Based Systems' conference, 2011*, New York, USA.
- Reilly J., Dashti S., Ervasti M., Bray J. D., Glaser S. D., Bayen A. M. (2013). Mobile Phones as Seismologic Sensors: Automating Data Extraction for the iShake System, *ONS On Automation Science And Engineering*, vol. 10, n° 2, p. 242-251.
- Shala U., Rodriguez A. (2011). *Indoor Positioning using Sensor-fusion in Android Devices*, Graduation work for a Degree of Master in Embedded Systems, School of Health and Society, Department Computer Science, Kristianstad University, Sweden, 2011.
- Virieux J., Le Figaro, <http://www.lefigaro.fr/sciences/2014/04/07/01008-20140407ARTFIG00398-le-sud-est-est-la-region-francaise-la-plus-exposee-au-risque-sismique.php>.