



POLYTECH[®]
MARSEILLE



Département Informatique, Réseau et Multimédia
Promotion 2015

BLACK SHEEP

MIRROR

Projet de Fin d'Etudes

Rapport du projet

Equipe :

Yohann DUFOUR

Petru-Maria MAGDELEINE

Supervision :

Stéphane AYACHE

Association Réso-Nance

Association VideoSpread

Année 2014 - 2015

Table des matières

| | |
|--|-----------|
| 1 INTRODUCTION | 3 |
| 2.1 ARCHITECTURE GLOBALE DU SYSTEME..... | 4 |
| 2.1 ENVIRONNEMENT MOBILE | 4 |
| 2.2 ENVIRONNEMENT WEB..... | 4 |
| 2.3 INSTALLATION..... | 5 |
| 3 FONCTIONNALITES GENERALES | 5 |
| 3.1 APPLICATION MOBILE | 5 |
| 3.2 MIDDLEWARE | 5 |
| 3.3 APPLICATION WEB..... | 6 |
| 4 PRESENTATION DES FONCTIONNALITES | 7 |
| 4.1 DONNEES ACCESSIBLES DANS L'APPLICATION MOBILE | 7 |
| 4.2 DONNEES ENVOYEEES SUR LE SERVEUR WEB | 7 |
| 4.3 VISUALISATION DES CONTACTS PAR NOMBRE DE SMS ECHANGE | 7 |
| 4.4 VISUALISATION DES SMS ECHANGES PAR HEURE..... | 8 |
| 4.5 VISUALISATION DES MOTS LES PLUS UTILISES DANS LES SMS..... | 8 |
| 4.6 PRESENTATION DE L'ŒUVRE NUMERIQUE | 8 |
| 5 DOCUMENTATION..... | 11 |
| 5.1 APPLICATION MOBILE ANDROID..... | 11 |
| 5.2 EXPLOITATION D'UNE API REST..... | 11 |
| 5.3 VISUALISATION WEB | 11 |

1 Introduction

Le projet de fin d'étude s'inscrit dans la 5ème année du cursus Informatique Réseaux et Multimédia de l'école Polytech Marseille.

Notre but est de concevoir et de développer une application multimédia Web et Mobile sur le thème de l'intrusion dans la vie privée et de l'exploitation des données personnelles. Notre mission étant de sensibiliser les utilisateurs lorsqu'ils installent des applications qui nécessitent des autorisations intrusives (jusqu'au vol de données) qui ne sont pas nécessaires au fonctionnement propre de l'application.

Ce document a pour but de présenter les caractéristiques principales des applications que nous avons réalisées. Notre objectif est d'apporter les éléments principaux décrivant le fonctionnement global et leur mise en œuvre dans le cadre du projet.

La suite de ce document vise à détailler les différentes composantes du projet, avec une approche de spécification et de conception, tout en fournissant l'ensemble des détails techniques nécessaires à la compréhension du projet.

2 Environnement technique

2.1 Architecture globale du système

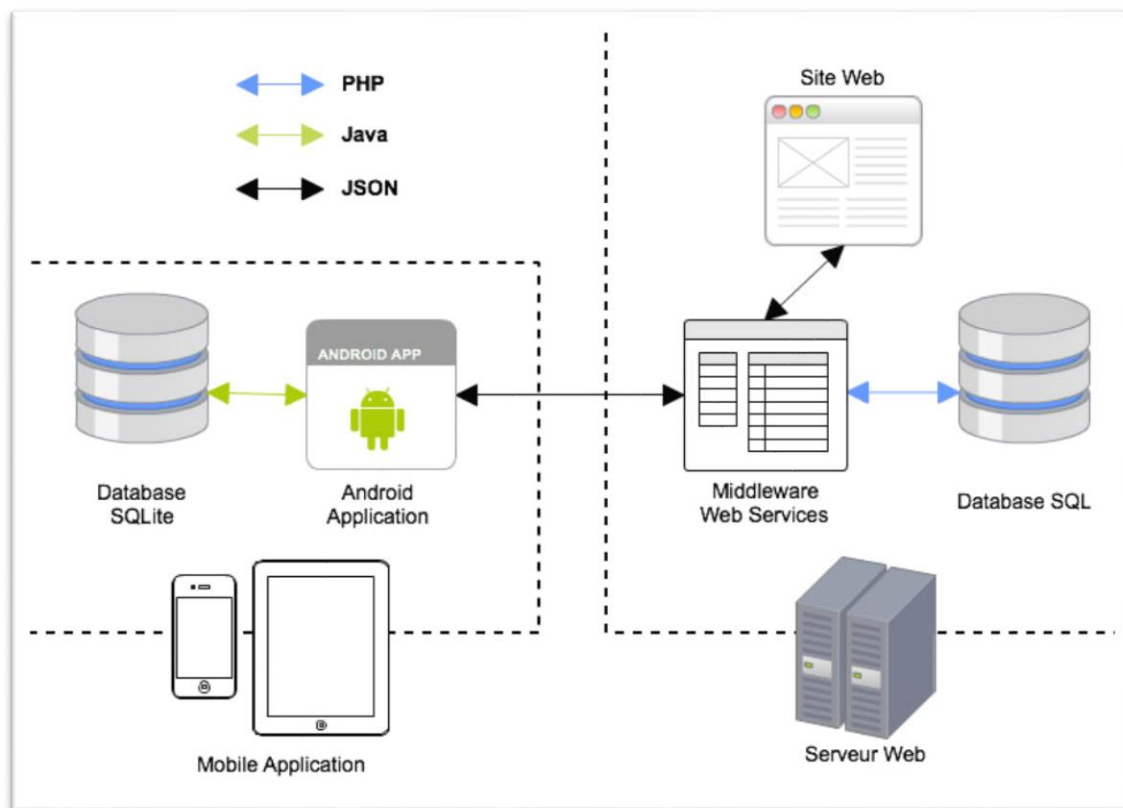


Figure 1 : Architecture globale du système

La figure 1 présente la solution retenue pour l'architecture globale du projet. Celle-ci comprend deux parties, une partie mobile et une partie web.

2.1 Environnement mobile

La partie mobile est composée d'une application mobile sous Android. Elle est associée à une base de données SQLite intégrée au smartphone. Elle est nécessaire au fonctionnement du système Android et des différentes applications. Les applications communiquent avec cette base en utilisant des requêtes en Java.

2.2 Environnement Web

La partie web regroupe un middleware, un site web et une base de données :

Le middleware une API REST écrite en PHP. C'est cette application qui permet d'interagir avec la base de données SQL en traitant les différentes requêtes.

Les échanges se font par requêtes HTTP POST en utilisant des données sous le format JSON. L'API utilise des requêtes SQL en PHP pour interagir avec la base de données.

Le site web qui permet de visualiser les données utilise le Framework Bootstrap ainsi que plusieurs librairies en JavaScript comme D3 ou HighCharts.

2.3 Installation

L'application mobile est fournie sous forme de fichier .apk qui peut être directement installé sur des mobiles sous Android.

L'application web nécessite un serveur Web sous Apache. Dans le cadre de notre projet, nous avons utilisé le service fourni par OVH. Nous avons donc un espace dédié pour accueillir le middleware et le site web, ainsi qu'une base de données MySQL.

3 Fonctionnalités générales

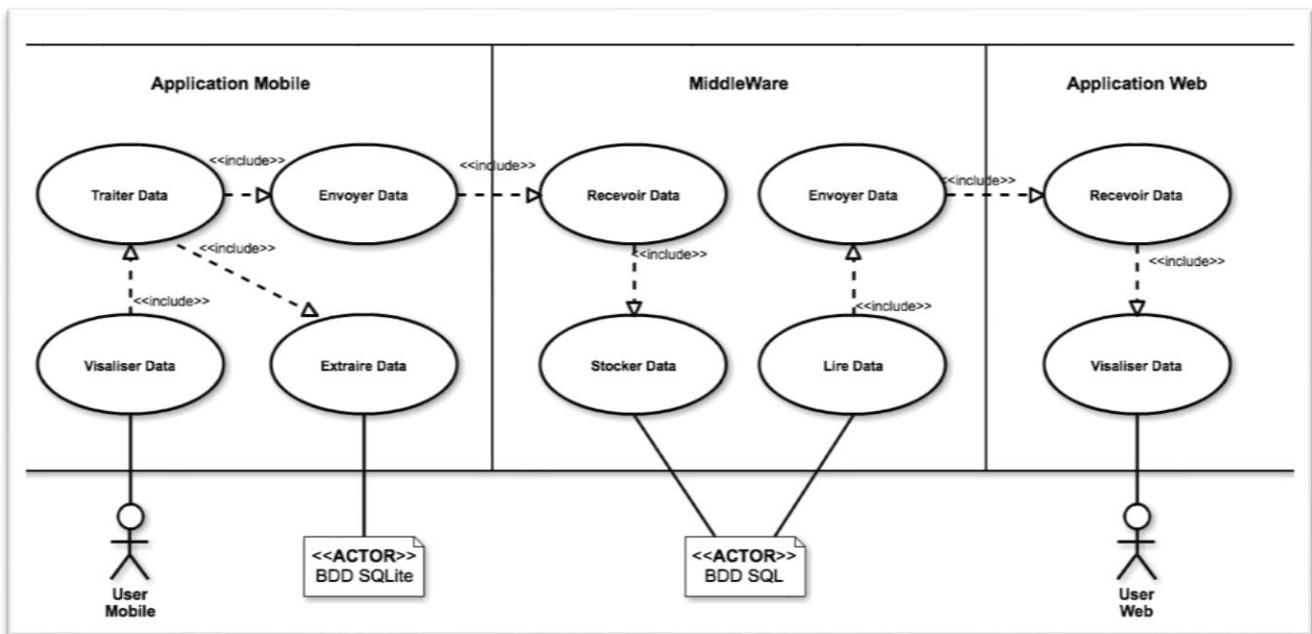


Figure 2 : Diagramme d'utilisation du système

La figure 2 illustre l'ensemble des fonctionnalités du diagramme d'utilisation. Elles sont réparties en trois parties, selon l'architecture : Application Mobile, Web et MiddleWare.

Les acteurs sont :

- un utilisateur mobile (acteur principal),
- un utilisateur web (acteur principal),
- la base de données SQLite (matériel externe)
- la base de données SQL (matériel interne)

3.1 Application Mobile

Les principales fonctionnalités de l'application mobile sont :

- l'extraction des données personnelles (SMS, appels, contacts et localisation GPS),
- le traitement de ces données,
- la transmission d'une partie des données vers un service web,
- la visualisation des données par l'utilisateur.

3.2 MiddleWare

Le rôle du Middleware en tant que service web est de faire l'interface entre l'application mobile, l'application web et la base données :

- Recevoir les données utilisateurs depuis une application mobile,
- Charger des données dans la base de données,
- Récupérer des données depuis la base de données,
- Transmettre des données à l'application web.

3.3 Application Web

L'application web permet une visualisation externalisée des données personnelles extraites d'un smartphone par l'application mobile. Elle est liée à la base de données au travers du middleware.

Le but est de :

- proposer une visualisation propre à chaque utilisateur selon ces propres données personnelles,
- présenter les résultats d'un traitement global de l'ensemble des données des utilisateurs.

4 Présentation des fonctionnalités

4.1 Données accessibles dans l'application mobile

Au lancement de l'application mobile, plusieurs types de données sont chargés à partir de la base de données et des périphériques du téléphone :

- accès aux SMS
- accès à la position GPS
- recherche d'une borne Wifi à proximité
- accès à la liste des comptes connectés (Google, Facebook, Skype, etc.)
- accès à l'historique des appels
- accès à la liste des connexions internet
- accès aux données sur les contacts

Par ailleurs, cette liste est une liste non exhaustive des données accessibles et exploitables via l'API d'Android.

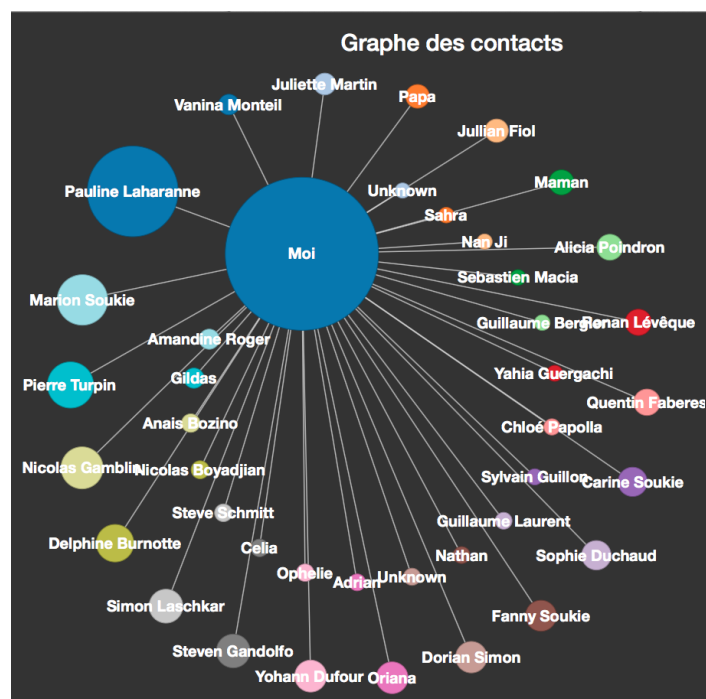
4.2 Données envoyées sur le serveur web

Nous avons décidé de récupérer qu'une partie réduite des données que nous avons chargées au sein de l'application mobile. Nous avons privilégié les données permettant une visualisation pertinente et riche sur l'application web.

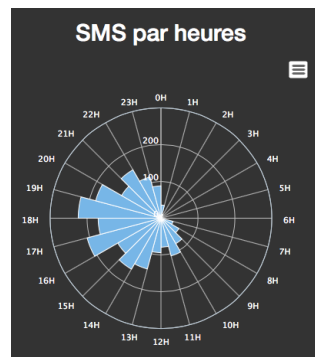
Les données envoyées au serveur web et stockées dans la base de données web sont :

- le nombre de SMS échangés (envoyés et reçus) par heure de la journée
- les mots les plus utilisés dans les conversations SMS
- le nombre de SMS envoyés par contact (nom et numéro du contact)
- la localisation GPS de l'utilisateur
- la détection d'un utilisateur proche de la borne Wifi « BlackSheepMirror »

4.3 Visualisation des contacts par nombre de SMS échangé



4.4 Visualisation des SMS échangés par heure



4.5 Visualisation des mots les plus utilisés dans les SMS

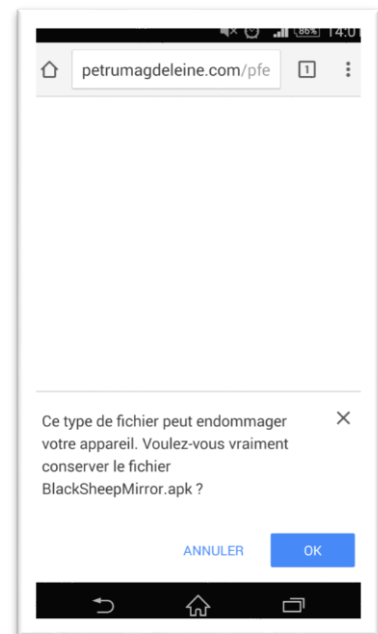


4.6 Présentation de l'œuvre numérique

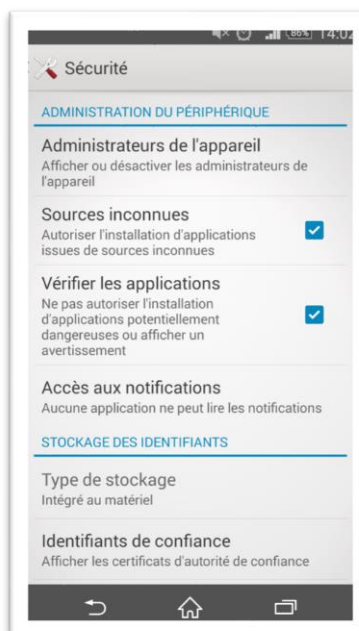
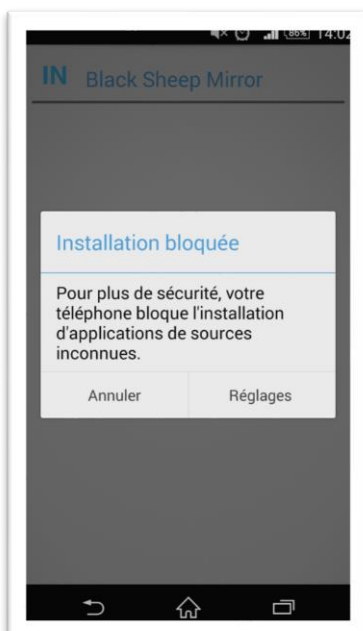
Dans le but d'atteindre la finalité du projet qui consiste à présenter cette application mobile comme une œuvre numérique, nous avons mis en place un système permettant de présenter de manière ludique et automatique les différentes visualisations.

Dans une pièce, on émet un réseau wifi du nom de « BlackSheepMirror ». Dès lors qu'un utilisateur rentre dans la pièce, on lui propose d'installer l'application :

- le téléchargement de l'application se fait via le serveur web : <http://petrumagdeleine.com/pfe/app/BlackSheepMirror.apk> ou en flashant le QRCode suivant :



- l'installation de l'application se fait en cliquant sur l'application installée et en acceptant l'installation d'applications tierces de sources inconnues (Paramètres > Sécurité > Sources inconnues)



- le lancement de l'application est ensuite possible via le menu d'Android



Par ailleurs, sur un écran est projetée la page web suivante :

<http://petrumagdeleine.com/pfe/wifi.html>

Dès que l'application est lancée, l'application mobile envoie les données nécessaires à la visualisation. De plus, elle détecte qu'elle est à proximité du réseau « BlackSheepMirror » et le notifie au serveur. Ce dernier va alors afficher sur la page web projetée les données des personnes étant à proximité de la borne.

Une visualisation « personnelle » est aussi disponible à l'adresse suivante :

<http://petrumagdeleine.com/pfe/id.html>

5 Documentation

5.1 Application mobile Android

La logique métier de l'application mobile est découpé en différentes tâches assurées par des classes spécifiques :

- ContactProvider : extrait les informations sur les contacts
- Conversation : représente une conversation (nom et numéro du contact, nombre de SMS échangés)
- GlobalClass : représente l'application en générale et permet de stocker des variables globales partagées par toutes les classes
- JSONFactory : construit les éléments JSON à envoyer au serveur web
- LocationService : envoie de manière continue (que l'application soit en cours d'exécution ou non) la localisation GPS
- Main : coordonne l'extraction et l'envoi des données
- ProfileView : gère l'affichage de l'application
- Sender : permet l'envoi de requêtes POST au serveur
- SMS : représente un SMS (nom et numéro du contact, texte du message, date, envoyé ou reçu)
- SMSProvider : extrait les données des boîtes de réception et d'émission des SMS
- SMSWordsCount : trie et renvoie les mots les plus utilisés parmi les conversations SMS
- WifiDeviceService : cherche de manière continue (que l'application soit en cours d'exécution ou non) la borne wifi « BlackSheepMirror »

5.2 Exploitation d'une API REST

L'ensemble des échanges de données avec l'application mobile ou l'application web se fait via une API REST implémenté dans le fichier api.php. Toutes les requêtes sont de la forme :

<http://petrumagdeleine.com/rest/requete>

Pour qu'une requête soit traitée, il suffit d'implémenter la fonction requête dans la classe API du fichier api.php.

5.3 Visualisation web

Chaque visualisation web est implémentée dans un fichier JavaScript. Chaque fichier contient au moins les deux méthodes :

- getJSONData : envoie une requête GET à l'API REST pour récupérer les données de type Data au format JSON
- drawData : dessine la visualisation dans la page HTML