

LU1

Titre du projet : Récupération de flux de données personnelles

Guobao LI Kévin Le Heurt

01/11/2015

Catalogue

1. Présentation de l'entreprise	2
2. Contexte du projet	2
3. Modèle du domaine avec explications	3
4. Analyse des exigences particulière par rapport à la qualité du logiciel	4
5. Objectifs globaux du projet.....	6
6. Définition du premier sprint et avancement des premières semaines.	6
7. Les rapports de stand-up meetings et les burn-down diagrammes	7
8. Planning du projet global contenant:	7
9. Une analyse de risques	10

1. Présentation de l'entreprise

Keeme est une startup basée sur un concept innovant, l'internet des objets, créée par Eric Grall. Elle est située 18 rue du calvaire 29000 Quimper, France.

2. Contexte du projet

Ces dernières années, le secteur des objets connectés a littéralement explosé. Ce sont notamment les bracelets fitness qui ont envahi le marché. Le succès est tel que de nombreux fabricants – le géant Apple... dernièrement – se sont lancés sur celui des montres connectées.

Keeme est une solution pour particulier de gestion de ses données personnelles à fin de stockage et de vente. Keeme s'appuie sur les objets du quotidien (pc, mac, smartphone,...), et sur les objets connectés (bracelet fitness, montre connecté,...). La solution récupère l'ensemble des données de chaque utilisateur afin de les centraliser dans un cloud. A partir de cette plateforme il peut gérer ses données à sa guise, et peut ainsi les revendre.

Dans ce cadre la start-up a besoin de développer des outils sur lesquels l'application s'appuiera. Il s'agit pour nous de traiter la récupération des données depuis les objets, et la mise en place d'éventuels traitements de ces données. Des grands axes ont déjà été tracés quant à l'architecture de cette partie et les technologies à employer.

Le système à mettre en place consiste en un broker sécurisé et scalable. Ce broker fera le lien entre les objets et le cloud tout en permettant d'implémenter des opérations de traitement sur les données. Le broker sera basé sur Apache Kafka : il fonctionnera donc sur le paradigme publisher-subscriber. Du côté objets, la communication passera par le protocole MQTT. Ainsi il faudra réaliser un connecteur MQTT pour Kafka, qui n'en possède pas pour le moment. Côté cloud la communication se fera via un module Spark Stream. Pour ces technologies nous seront amenés à programmer en Scala.

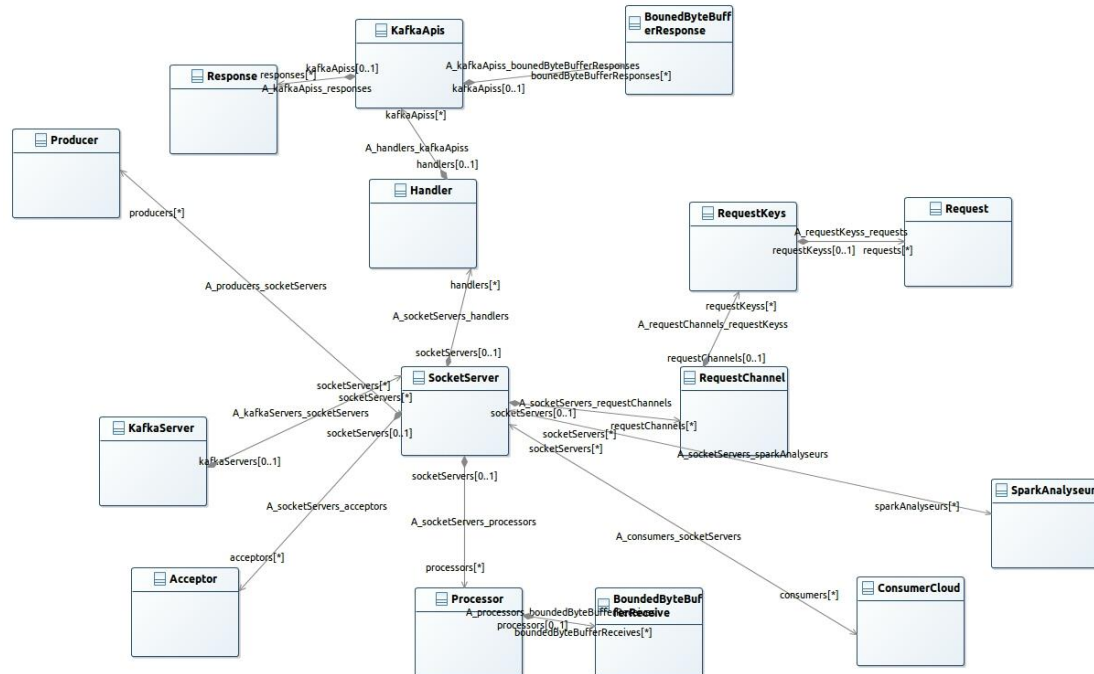
Concernant les technologies utilisées dans notre projet:

MQTT(Message Queuing Telemetry Transport) est un protocole de messagerie publisher-subscriber basé sur le protocole TCP/IP. Il est léger, ouvert, simple et dédié à la communication de IOT(Internet of things) où un petit code empreinte est demandé et une petite bande passante suffit.

Apache Kafka est un système de message haut débit et distribué. Il est rapide et un seul broker permet de manipuler les lectures et les écritures des centaines de données par seconde. Il est scalable. Il est dédié à permettre à un seul cluster de servir comme un centre de données pour une grande organisation. Il peut être étendu transparent sans le temps d'arrêt. Les flux de données sont partitionnés et copiés à une groupe de broker assurant la sécurité de données et agrandissant la capacité de service.

Apache Spark est un cadre open source de calcul distribué, initialement développé à Berkeley par AMPLab et maintenant supporté par la fondation Apache. Contrairement à Hadoop qui utilise le pattern Map-Reduce sur des disques, Spark travaille en In-Memory ce qui potentiellement est 100 fois plus rapide.

3. Modèle du domaine avec explications



Connecteur MQTT de Kafka

KafkaServer: C'est une classe de serveur à la base de SocketServer, qui gère le scheduler, le zookeeper, le système de log, le gérant de décalage et le KafkaApis pour l'analyse des requests.

SocketServer: C'est un socket serveur de NIO, qui constitue un accepteur manipulant les nouvelles connections, plusieurs processors lisant les données vients du socket et plusieurs handlers manipulant les requests et produisant les réponses à les processors pour l'écriture.

Acceptor: C'est un thread qui accepte et configure les nouvelles connections.

Process: C'est un thread qui manipule tous les requests vients d'une seule connections.

BoundedByteBufferReceive: C'est une structure de données qui représente une communication entre le client et le serveur.

RequestChannel: C'est une passante qui transfère les requests.

RequestKeys: C'est une marque qui représente le but d'une request.

Request: C'est une structure de données qui représente une type de request.

Handler: C'est un thread qui répond à le request de Kafka.

KafkaApis: C'est une classe qui manipule les requests variés de façon logique.

BoundedByteBufferReponse: C'est une structure de données qui représente une communication entre le client et le serveur.

Réponse: C'est une structure de données qui représente une type de réponse.

Un subscriber de Cloud Keeme sur Kafka

Cloud Keeme: C'est un serveur de cloud qui garde tous les données personnelles envoyées par les clients. Il s'abonnera à un topic sur le Kafka qui envoyera les messages dans ce topic au Cloud.

Un analyseur des données par Spark Streaming

Spark Streaming: C'est un outil qui analyse les flux de données dans la passante entre le Kafka et Cloud Keeme.

4. Analyse des exigences particulière par rapport à la qualité du logiciel

			très faible	faible	moyenne	important	très important	Commentaire
P	Performan ce (Efficicenc y)	Temps de réponse t comportem ent dynamique					√	Notre composan tes sont très sensible à le temps réel, car dans le reseau, ils sont souhaités à transmettr e une grosse somme de donées dans un temps limit.
		Utilisation			√			Nous

		des ressources (mémoire, débit en transactionnel, etc)						allons servir de le Kafka qui est scalable sous une forme de groupe, donc les ressources pour la fonction ne sont pas une limite.
S	Serviceability, Maintainability (Garantie de service, MCO)	Capacité et facilité de						
		- Analyse des défaillances			√			
		- Modification		√				
		- Stabilité (confinement des défaillances)					√	Nous nous concentrons à assurer la stabilité de transmission.
		- Test (automatisation, non régression etc)		√				
E	Evolution, Portability, Adaptability (Evolutivité)	Capacité et facilité de						
		- Adaptation et évolution				√		ça permet de développer notre projet à long terme.
		- Installation des modifications			√			

7. Les rapports de stand-up meetings

Nous avons rédigé des fiches de suivi chaque semaine pour enregistrer le travail que nous avons fait. Et au-dessous c'est le rapport de stand-up réunion.

29/09/2015 Au 06/10/2015

TRAVAIL EFFECTUE

Nous avons travaillé en binôme sur le sujet afin de préparer la première réunion. Nous avons tenté de retracer le travail demandé dans ce projet, et de découvrir les technologies impliquées. Nous avons ce mardi rencontré Monsieur Parrein. Monsieur Grall nous a joint par téléphone. Nous avons abordé de nombreux points lors de cet échange : généralités du projet, contexte, objectifs, et notamment les technologies. Nous comprenons que notre travail consistera à mettre en place un broker. Celui-ci fonctionnera avec le protocole MQTT. Il permettra de faire le lien entre des flux de données « publish » et « suscribe », entre des objets connectés et le cloud Keeme. Les données seront stockées par le broker. Nous nous appuierons sur Apache Kafka (ou éventuellement RabbitMQ) et Spark Streaming pour réaliser ce broker. Dans un deuxième temps nous pourrons développer des modules de machine learning associés à un outil de visualisation. Dans la semaine à venir nous allons nous documenter sur ces technologies. Edgar se chargera de Kafka tandis que Kévin se chargera de Spark. Nous travaillerons ensemble sur MQTT. Pour communiquer avec la start-up nous envisageons d'utiliser Slack.

8. Planning du projet global contenant:

Nous avons divisé notre projet en deux phases. Dont la première c'est la partie du connecteur MQTT, et la deuxième c'est la partie du Spark Streaming.

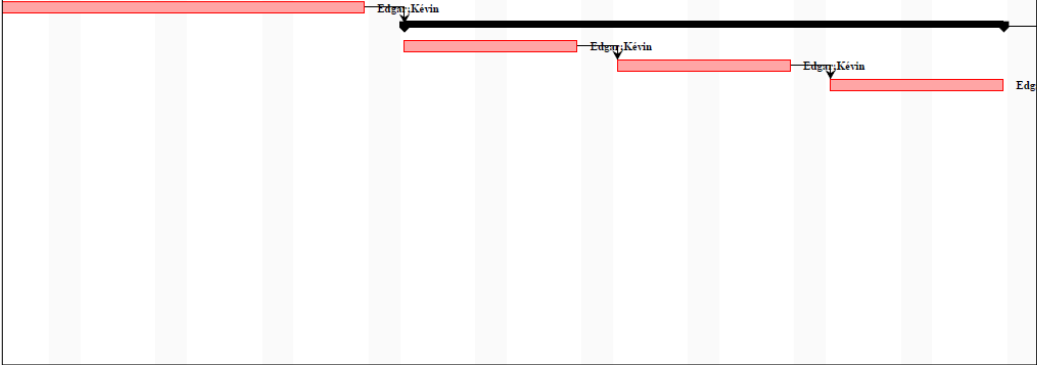
Dans la première phase, nous avons le divisé en trois sprints, dont le première c'est la réalisation du connecteur MQTT qui consititue la spécification, l'implémentation et le test. Dont le deuxième c'est une liaison avec Kafka qui consititue la documentation, la mise en oeuvre et le test. Dont le troisième c'est la intégration qui consititue l'implémentation et le test.

Dans la deuxième phase, nous avons le divisé en deux sprints, dont le première c'est la réalisation du analyseur à la base de Spark qui consititue la spécification, l'implémentation et le test. Dont le deuxième c'est la intégration qui consititue l'implémentation et le test.

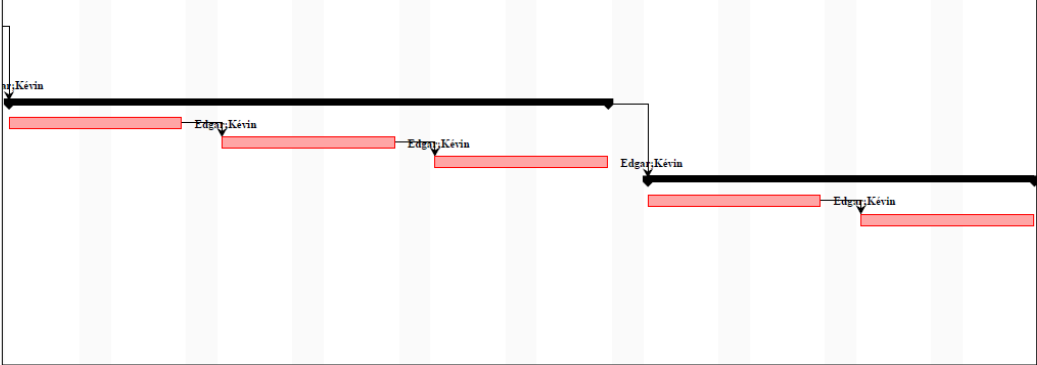
C'est la diagramme de Gantt au dessous :

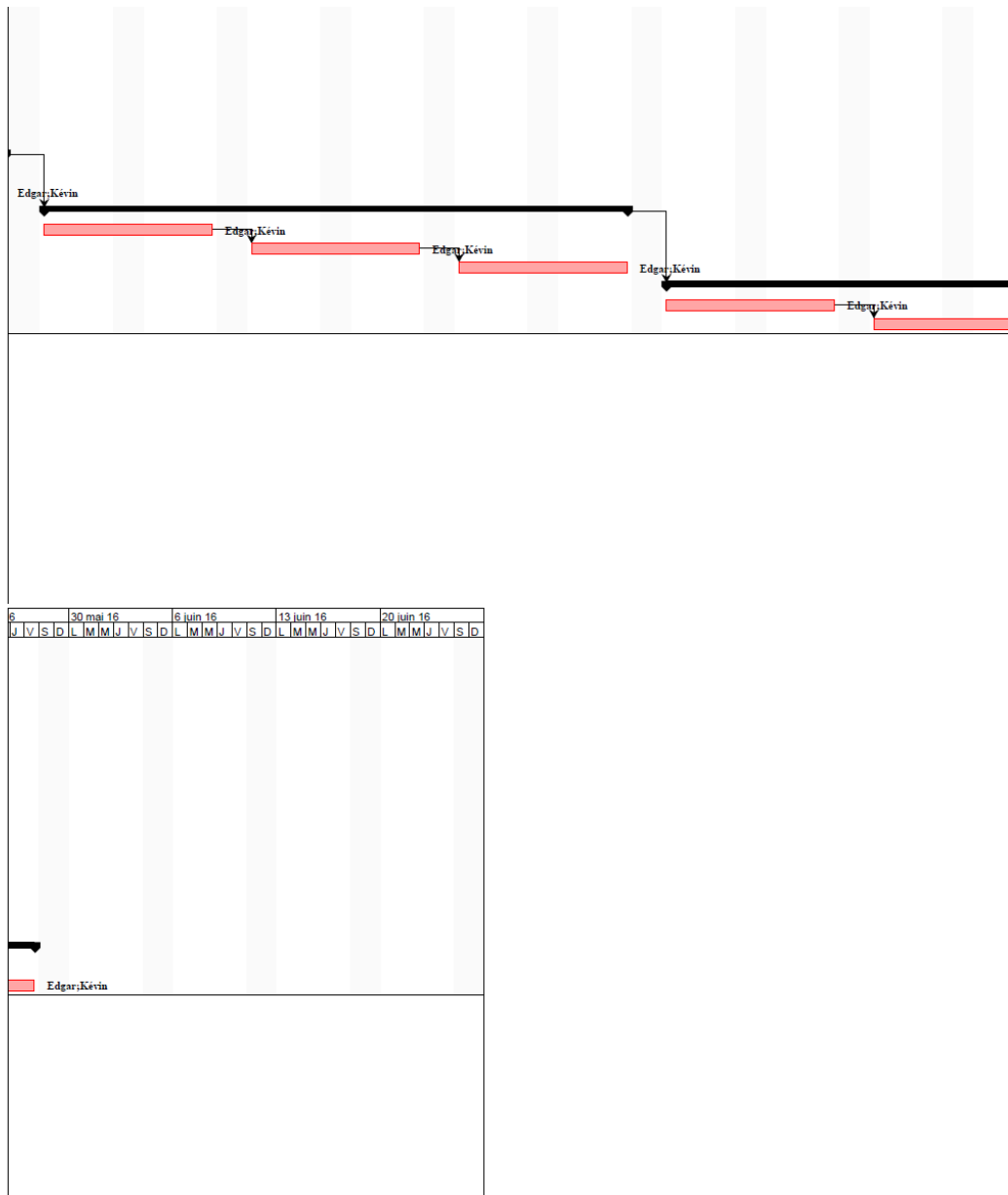
	⑩	Nom	Durée	Début	Fin	Prédécesseurs	Noms des ressources			2 nov
								V	S	D
1		Analyse des besoins	20 jours	02/11/15 08:00	27/11/15 17:00		Edgar;Kévin			
2		Connecteur MQTT	30 jours	30/11/15 08:00	08/01/16 17:00	1				
3		Spécification	10 jours	30/11/15 08:00	11/12/15 17:00		Edgar;Kévin			
4		Implémentation	10 jours	14/12/15 08:00	25/12/15 17:00	3	Edgar;Kévin			
5		tests unitaires	10 jours	28/12/15 08:00	08/01/16 17:00	4	Edgar;Kévin			
6		Kafka	30 jours	11/01/16 08:00	19/02/16 17:00	2				
7		Documentation	10 jours	11/01/16 08:00	22/01/16 17:00		Edgar;Kévin			
8		Mise en oeuvre	10 jours	25/01/16 08:00	05/02/16 17:00	7	Edgar;Kévin			
9		Tests unitaires	10 jours	08/02/16 08:00	19/02/16 17:00	8	Edgar;Kévin			
10		Intégration	20 jours	22/02/16 08:00	18/03/16 17:00	6				
11		Implémentation	10 jours	22/02/16 08:00	04/03/16 17:00		Edgar;Kévin			
12		Tests	10 jours	07/03/16 08:00	18/03/16 17:00	11	Edgar;Kévin			
13		Spark	30 jours	21/03/16 08:00	29/04/16 17:00	10				
14		Spécifications	10 jours	21/03/16 08:00	01/04/16 17:00		Edgar;Kévin			
15		Implémentation	10 jours	04/04/16 08:00	15/04/16 17:00	14	Edgar;Kévin			
16		Tests unitaires	10 jours	18/04/16 08:00	29/04/16 17:00	15	Edgar;Kévin			
17		Intégration	20 jours	02/05/16 08:00	27/05/16 17:00	13				
18		Implémentation	10 jours	02/05/16 08:00	13/05/16 17:00		Edgar;Kévin			
19		Tests	10 jours	16/05/16 08:00	27/05/16 17:00	18	Edgar;Kévin			

15	9 nov. 15					16 nov. 15					23 nov. 15					30 nov. 15					7 déc. 15					14 déc. 15					21 déc. 15					28 déc. 15					4 janv. 16					
M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D



11 janv. 16			18 janv. 16			25 janv. 16			1 févr. 16			8 févr. 16			15 févr. 16			22 févr. 16			29 févr. 16			7 mars 16			14 mars 16					
L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V





STOR : 1.05 (notion : High, raison : Le Kafka enrégistre les données de façon du fichier.)
PVOL : n/a (notion : Very Low, raison : Nous prévoyons aucune changement de l'environnement d'exécution du programme.)
ACAP : 1.00 (notion : Nominal)
PCAP : 0.88 (notion : High, raison : Nous avons des très bons programmeurs.)
PCON : 0.90 (notion : High)
APEX : 1.00 (notion : Nominal, raison : Nous avons assez des expériences pour faire un tel produit.)
PLEX : 1.00 (notion : Nominal)
LTEX : 1.00 (notion : Nominal)
TOOL : 1.00 (notion : Nominal)
SITE : 0.93 (notion : High)
SCED : 1.00 (notion : Nominal, raison : Nous avons assez du temps.)
Produit : 1.05

Nous allons travailler 6 mois à deux personnes dans le coeur du projet.

A=2.9, B=0.91

Donc

Taille en KSLOC : 6.7

9. Une analyse de risques

Risques sur les hommes et les compétences

Nous avons les compétences pour développer un logiciel, pourtant c'est notre première fois à toucher les technologies de Kafka et Spark. Donc sans la maîtrise de ces technologies, nous risquons passer plus longue du temps pour réaliser le projet. En effet, un de nous est un chinois qui est en train d'améliorer sa français. Pour l'instant, il a un peu du mal à lire et parler de façon français.

Risques sur le planning

Nous risquons de la décision sur la façon pour réaliser le connecteur MQTT de Kafka, car nous avons plusieurs chemins pour le faire. Quant à le délais, nous n'avons pas encore le préciser.

Risques sur les technologies

Nous risquons de la possibilité de ne pas pouvoir réaliser le connecteur MQTT de Kafka sans l'aide de l'autre projet de open source, car ce sera un peu difficile pour nous à le développer dans l'intérieur de Kafka. Dans un l'autre cote, il existe une différence entre le format de request et réponse dans MQTT et dont le Kafka, donc nous devons les accorder dans quelque façon.