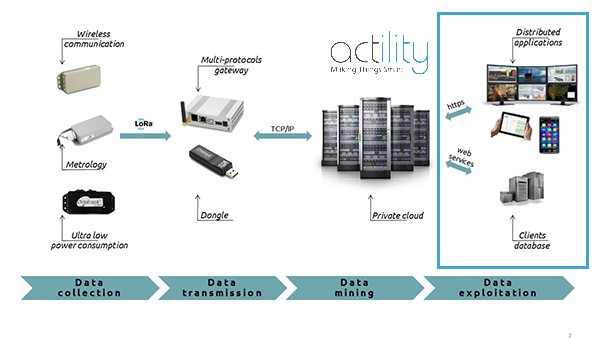
Recherche d’une Interface Utilisateur Web pour afficher les données.

Objectif : Faire une page web qui affiche les données des capteurs sous forme de graphe après leur récolte.



On s’intéresse ici à la partie encadrée en bleu qui est l’exploitation des données.

# 1 / Quelles données ? Pourquoi ?

## 1.1 Pour traiter quelles données ?

Il s’agit des données récoltées par les différents capteurs du client. Ces données sont stockées sur les serveurs du client après être passés par les serveurs d’Actility. (Image)  
Il peut s’agir de températures, de taux d’humidité, de luminosité, etc…

## 1.2 Pour quelles raisons ?

Pour récolter les données et pouvoir les afficher à travers un modèle rapidement compréhensible rapidement pour l’utilisateur.  
En effet, un histogramme ou une courbe est plus facilement compréhensible pour l’utilisateur qu’une suite de valeurs brutes.

# 2 / Les différentes solutions techniques

## 2.1 FLASH

Flash est un logiciel permettant de créer des contenus multimédias (animations, vidéos, jeux, applications,...) destinés à être publiés sur Internet. Il est nécessaire d’avoir un Flash player pour pouvoir visualiser du contenu Flash (Ex : Adobe Flash Player).

### 2.1.1 Avantage(s)

Permets d’afficher du contenu sous de multiples formes graphiques. Cette solution permettrait d’obtenir des graphes interactifs, que l’utilisateur pourrait modifier à souhait (changement de l’échelle, zoom sur une partie du graphe, ajout de conditions, …).

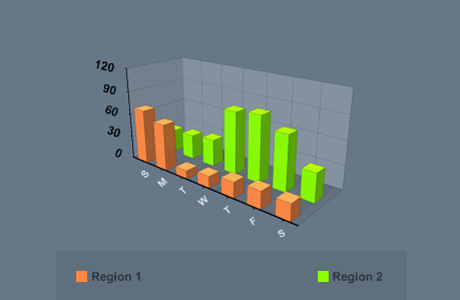
### 2.1.2 Inconvénient(s) & contrainte(s)

Nécessite l’installation du Flash Player sur son navigateur. Il est préférable de l’avoir tout le temps à jour, et tous les OS n’ont pas la dernière version.

De plus, le plugin est lourd et consomme des ressources système (Processeur, RAM, …).  
En ce qui concerne la réalisation, de bonne bases en ActionScript sont nécessaires ainsi qu’en animation vidéo.

### 2.1.3 Solution(s)

Bibliothèque(s) retenue(s): Maani.



Exemple(s):

Maani : <http://www.maani.us/charts/index.php>

## 2.2 Génération d’image statique

Il s’agit d’une solution simple qui consiste à l’aide d’outils (bibliothèques) de créer des images de graphes automatiquement.

### 2.2.1 Avantage(s)

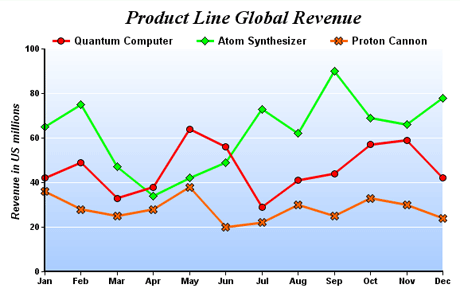
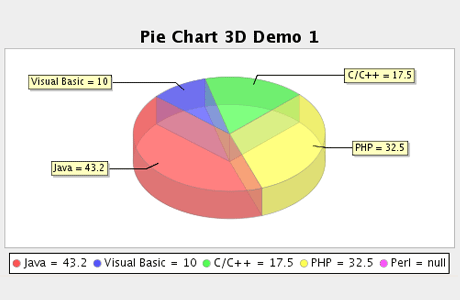
Les images sont réutilisables, une fois générée en fonction des données on peut les mettre sur n’importe quel site ou blog afin de les partager. On peut aussi garder les images pour les transporter à l’aide de n’importe quel support de stockage (Clé USB, Disque dur, Serveur, ...).

### 2.2.2 Inconvénient(s) & contrainte(s)

Les images étant fixes, on a une solution qui n’est pas interactive, l’utilisateur n’a aucun pouvoir sur les graphiques.   
Si l’utilisateur veut afficher une donnée en plus, ou ajouter une précision sur l’affichage de ses données, il sera nécessaire de générer une nouvelle image qui correspond à la demande de l’utilisateur.

### 2.2.3 Solution(s)

Bibliothèque(s) retenue(s): JFreeChart, Chart Director.

Exemple(s):  
(à gauche JFreeChart, à droite Chart Director).

## 2.3 JAVASCRIPT

### 2.3.1 Avantage(s)

Similaire à [FLASH](#h.ydpfy3fy3rpo) avec l’avantage de ne pas avoir à d’installer un plugin.

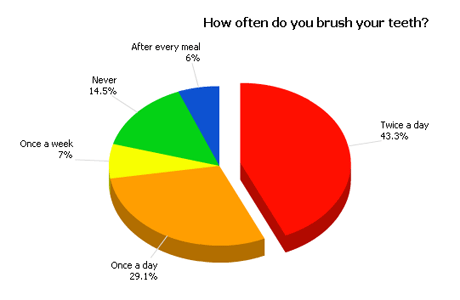
### 2.3.2 Inconvénient(s) & contrainte(s)

Il est nécessaire de connaître le langage JavaScript (similaire à ActionScript) en plus des différents langages dédiés au Web (HTML, CSS, …).

### 2.3.3 Solution(s)

Bibliothèque(s) retenue(s): EJSCharts, amCharts.

Exemple(s):



amCharts (Image) :

<https://www.amcharts.com/demos/>

EJSCharts : <http://www.ejschart.com/examples.php>

# 3 / Conclusion

Après avoir effectué une étude sur les trois solutions présentées précédemment, notre choix se porte sur [JavaScript](#h.s671eq6hzu6h).  
En effet, il apporte de nombreux avantages, et corrige les principaux inconvénients des images et de Flash.