

# TRIBS



Retour d'expérience

Meetup Phonegap



# --> Agenda

- Présentation
- Choix techniques
- Focus
  - Mode Online/Offline
  - Modularisation
  - Difficultés rencontrées

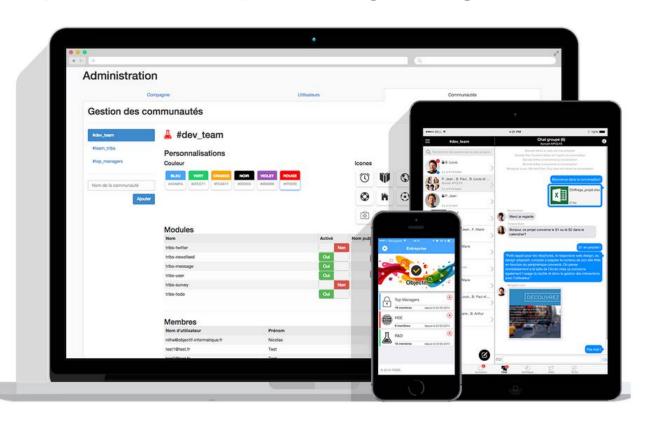






## ---> Présentation

Un outil pour Communiquer, Partager et Agir en communauté





## Présentation



**Annuaire** 



Messagerie groupée



Veille communautaire







**Sondages** 



# Choix techniques

## Besoins

Application mobile, avec comme cible IOS, Android et WinPhone

## Contraintes

Chez Objectif, nous avons:

- beaucoup de développeurs Java (JEE), mais peu connaissent Android
- quasiment aucun (1 ou 2) développeurs qui connaissent IOS

En outre, Objectif milite depuis longtemps pour les applications web HTML5 et pour le développement d'applications mobiles hybrides avec Phonegap.

En 2011, nous avons gagné le concours d'application pour le projet de Mobilier Urbain Interactif de JCDecaux, avec l'application P@rcours, développée en GWT/Phonegap.

Nous utilisons maintenant Angular.js pour développer les applications web.

Donc tout naturellement, on a choisi le couple Angular.js/Phonegap pour développer Tribs.





# --> Choix techniques

#### **Front**

Une application web (CSS responsive, Online/Offline, ....), développée en Angular.js avec lonic Framework et Phonegap.

Fonctionnalités mobiles spécifiques nécessitant l'utilisation de plugins Phonegap :

- Notifications Push sur iOS et Android :
  - com.phonegap.plugins.pushplugin
- Manipulation de la barre de status sur iOS pour une meilleur intégration :
  - com.phonegap.plugin.statusbar
- Ouverture de liens directement dans l'application :
  - org.apache.cordova.inappbrowser
- Gestion du copier/coller :
  - com.verso.cordova.clipboard

L'appel à tous ces plugins a été encapsulé dans des services Angular.js dédié.

A l'exécution, en fonction de l'environnement (Phonegap ou Navigateur standard), nous activons le service Phonegap ou non.





# → Choix techniques

## Front (suite)

Fonctionnalités HTML5 mal ou non supportés par les navigateurs mobiles natif et nécessitant donc l'utilisation de plugins Phonegap :

- Gestion du Online/Offline
  - org.apache.cordova.network-information
- Gestion de la capture de média (Image, Vidéo et Son) :
  - org.apache.cordova.camera
  - org.apache.cordova.media-capture
  - org.apache.cordova.device
  - org.apache.cordova.file
  - org.apache.cordova.file-transfer

L'appel à tous ces plugins a été encapsulé dans des services Angular is dédié.

A l'exécution, en fonction de l'environnement (Phonegap ou Navigateur standard), nous basculons automatiquement entre le service Phonegap et un autre service utilisant les fonctionnalités natives du navigateur, si nécessaire.

**N.B.**: Le plugin FileTransfert pour WinPhone plantait. Il a fallut le corriger.





# --> Choix techniques

### Back

Back end développé en Node.js

Création d'un framework maison qui s'appuie sur :

- Nconf: pour la gestion de la configuration
- Express : pour la gestion des conversations HTTP
- Socket.io : pour la communication en temps réel (Websockets ou fallback)
- Passport : pour l'authentification
- Connect-Roles: pour les habilitations
- Sequelize : pour la persistence dans une base de données MySQL
- Broadway : pour l'injection de dépendances.

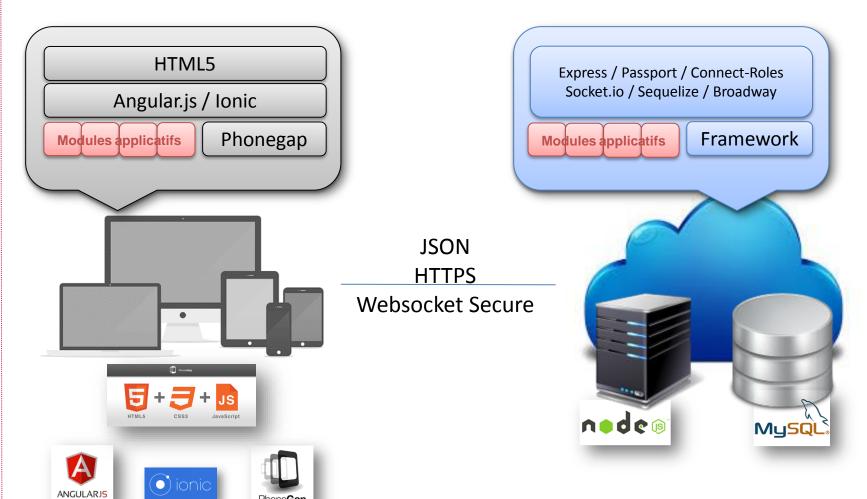
Ce framework permet de créer le routage HTTP par configuration (via un ou plusieurs fichiers JSON).





# → Choix techniques

## Architecture



Phone **Gap** 





## --> Focus

## Mode Online/Offline

Les navigateurs mobiles implémentent tous la détection de perte de connexion, mais le plugin Cordova propose d'autre fonctionnalités comme la détection du type de connexion.

#### Plugin utilisé:

org.apache.cordova.network-information

L'appel à ce plugins a été encapsulé dans un service Angular.js qui peut fonctionner en mode « dégradé » si l'application ne s'exécute pas dans Phonegap.

Pour fluidifier l'utilisation de l'application, même sans connexion internet, nous avons développé un service Angular.js de cache local, basé sur l'utilisation de Lawnchair.js, une librairie d'abstraction du stockage local, qui prend en charge :

- Localstorage
- WebSQLDB
- IndexedDB
- Cookies
- ...





# → Focus

## Modularisation

En utilisant la capacité de Node.js à charger dynamiquement des modules et la possibilité de créer des routes par configuration de notre framework, nous avons développé un gestionnaire de modules, qui permet de charger dynamiquement un module applicatif.

Un module applicatif est un module Node.js contenant :

- Un descripteur JSON pour le déploiement à chaud
- Un ou plusieurs services back
- Une ou plusieurs entités (Modèle de données en base)
- Un ou plusieurs fichier de configuration des routes HTTP
- Les ressources Angular.js de l'interface utilisateur du module.

Dans l'application web, en s'appuyant sur la librairie ocLazyLoad, qui permet de charger dynamiquement des modules Angular.js, nous avons développé notre service de gestion des modules applicatifs qui s'interface avec le gestionnaire de module coté serveur. Au rechargement de l'application, pour optimiser les performances, le gestionnaire de module coté serveur, génèrera dynamiquement (en fonction de l'utilisateur connecté) un manifeste pour inclure les modules dans l'Application Cache HTML5 du navigateur.

**N.B.**: l'utilisation de l'Application Cache HTML5 avec Phonegap est parfois problématique (problème de configuration du stockage)





# --> Focus

## Difficultés rencontrées

#### IOS

#### **Production & Distribution**

- Nécessite la mise en place de certificats de distribution pour l'application Nécessite un certificat spécifique aux notifications

#### **UI Kit**

- Non présence du kit iOS7 Quelques problèmes de positionnement iOS7 / iOS8

#### **Android**

#### **Production & Distribution**

Nécessite une Clé pour l'utilisation des notifications

## Prise en charge des versions OS

Difficultés d'adaptations (Android 2, Android 4)





# ---> Focus

## Difficultés rencontrées (suite)

## Windows Phone Routing

url formatée avec x-wmapp0

#### **CSS** et interface

Windows Phone 8 utilise IE10 avec le moteur de rendu « Trident »

#### WP8 natif et PhoneGap

- Upload Image & Camera: Erreur côté PhoneGap dans le code.
- Retard dans le support PhoneGap

#### **Notifications**

Pas de plugin existant

#### **Production & Distribution**

Certificat pour signer l'application



## **Contact**



## **Cédric BEURTHERET**

Responsable d'Objectif Labs cbeurtheret@objectif-informatique.fr

## **Nicolas ILHE**

Chef de produit Tribs - <a href="http://www.tribs.fr">http://www.tribs.fr</a>
nicolas.ilhe@tribs.fr

#### **Objectif**

6 Rue Laferrière 75009 PARIS

Tel 01 56 43 48 48 - Fax 01 56 43 48 49

www.objectif-informatique.fr