

*2016 /  
2017*

*Documentation Technique -  
Partie Développement*

Auteurs :  
Romain GARCIA  
Dorian GAILLETON  
Mathieu CALEMARD-DU  
GARDIN  
Kassem DAOUSSI  
Nicolas MAITREJEAN

[Nom de la société]

2016 / 2017

## Présentation du socle technique :

Pour répondre à l'attente du projet, nous avons décidé de partir sur des technologies Web pour la faciliter l'accessibilité aux utilisateurs.

Le framework Meteor a été pour nous le meilleur choix pour mettre en place l'architecture de l'application. D'un autre côté MongoDB nous permet de manipuler la structure de la base de données avec facilité et rapidité. Ce choix de système de gestion donnée à renforce notre choix pour Meteor. Pour coupler l'aspect back end avec le front end nous somme parti sur le framework Angular.

Angular embarque la technologie TypeScript qui permet de maintenir un code JavaScript plus propre, facile à lire et aisément maintenable.

Couplé à cela, on retrouve les outils de design Material et Bootstrap qui sont regroupés dans le framework Materialize CSS.

## Installation de l'application

Comme précisé plus haut, l'application se base sur Meteor. Pour installer Meteor, rendez-vous sur <https://www.meteor.com/>

Il faut exécuter l'installateur. Ouvrez votre favorite, installez les dépendances avec la commande (prenez garde à vous positionner dans le dossier du script) :

```
meteor npm install
```

Exécutez l'application :

```
meteor run
```

## Compte rendu technique de l'application :

### Mise en page, UI et UX

La mise en page est gérée par Material CSS et Bootstrap. Nous avons utilisé le système de colonnes de Bootstrap pour permettre une gestion du responsive design. Côté icônes, nous avons empruntés celles qui sont disponibles avec Material CSS.

Concernant l'UI/UX, nous avons opté pour la mise en place d'une barre latérale qui ne bougera jamais, ainsi qu'un container placé sur le reste de la page.



La barre latérale servira principalement à transiter à travers les différentes pages de l'application, qui seront affichées sur le container. C'est une application mono-page.

## Login

La gestion des logins est gérée par le module Account de Meteor. L'avantage de ce module est qu'il nous met à disposition une base de données déjà bien constituée et exhaustive pour les utilisateurs (meteor account). Il fournit aussi toute une panoplie de fonctionnalités comme le chiffrement / déchiffrement des mots de passe par exemple.

## Messages

L'envoi de messages est géré par un formulaire géré par un composant Angular qui sauvegardera le message côté serveur. Concernant l'affichage des messages en temps réel, nous faisons appel à Meteor qui de son côté, se charge d'actualiser le « mini mongo » et de synchroniser la page.

## Gestion des appels audio et vidéo

On utilise la librairie peer.js. Elle permet de gérer les flux de pair à pair en créant des liens entre les identifiants utilisateurs via un serveur Peer.js. Au démarrage, le navigateur demande à l'utilisateur l'accès à sa caméra et son microphone.

L'émetteur envoie une demande d'appel au destinataire. S'il l'accepte, une session est créée de son côté et le correspondant la rejoint. Sinon, dans le cas où il refuse, la session est supprimée. Si aucune réponse la demande d'appel persiste tant qu'il n'y a pas de réponse.

Dans le cas d'un appel de groupe, chaque utilisateur se connecte sur toutes les autres sessions d'utilisateur ouvertes.

## Les menus contextes :

Dans le menu des groupes et des contacts nous avons souhaité mettre en place un menu secondaire avec le clic-droit. Pour répondre à cette demande, le module angular-context. Il s'agit d'un module qui peut nous créer une balise HTML qui sert à gérer le menu contextuel via des divs template. Cette div est capable d'authentifier des types d'objets et de les utiliser.

