

BACHELIER EN INFORMATIQUE ET SYSTEMES

Finalité : réseaux et télécommunications

Développement Web : Prod-screen & Design

Rapport de stage effectué dans la firme
Powerdale

Najihi Fatima Zahra

Janvier 2019



Table des matières

Remerciements	4
Abstract	5
1 Introduction	6
2 Présentation générale	7
2.1 Présentation de l'entreprise	7
2.2 Présentation du travail	10
2.2.1 Moniteur de production en temps réel	10
2.2.2 Amélioration du design	12
3 Travail réalisé	13
3.1 Prod-screen	13
3.1.1 Analyse	13
3.1.2 Serveur	15
3.1.3 Client	16
3.1.4 Outils et protocoles	16
3.1.5 Résultat final	20
3.2 Nexxtmove service management	21
4 Conclusion	25
Annexes	26
Documentation prod-screen	26
Autorisation d'impression	29
Evaluation continue du stage	30
Carnets de bord de stage	31
Bibliographie	46



Remerciements

Tout d'abord, j'adresse mes remerciements à mes responsables de stage, Mr. Poelman Lieven ainsi que Mr. Jaffredo Alban pour le partage de leur expertise, leurs apports de connaissances au quotidien et surtout leur confiance.

Je tiens à remercier vivement mon professeur, Mme. Bastregghi Monica, pour son encadrement et ses conseils pour la rédaction de ce rapport, ainsi que mon coordinateur de stage, Mr. Nabet David, qui m'a permis de postuler dans cette entreprise et de trouver ce stage qui était en totale adéquation avec mes attentes.

Je remercie également toute l'équipe Powerdale pour son accueil et sa collaboration.

Enfin, je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de mon stage et notamment A. Smael pour sa précieuse aide à la relecture et à la correction de mon rapport de stage.



Abstract

During my internship at Powerdale I took part in two projects, both for internal use. The company has two main working fields: electric mobility and energy management. Its objective is to develop and sell charging stations for electric vehicles and offer solutions to help customers understand their energy usages.

The first project is a display screen, named “prod-screen”, that is used for testing purposes for the production team. It scans a range of IP addresses using ssh¹ (Secure Shell) and displays the data fetched from each Karo². A Npm³ package was created to make scanner types and functions available for any project that will need them. It is a library that can be imported by any project.

The display screen makes it possible for the production team to directly visualize the information concerning the karos that are being put to test in real time without having to reboot the charging points every time and is an easy way to verify that the data injected was saved. It is updated every three minutes and is installed on a Raspberry Pi, is automatically launched at start-up and doesn't require any account. It consists of two parts: the server side that is written in Node.js⁴ and the client side that uses Angular6⁵ and Typescript⁶.

The second project consists of front-end tasks like the styling of a billing project named “Nexxtmove service management”. Less⁷ was used in this project in combination with Bootstrap⁸.

In this report I will be presenting the company and I will then talk in detail about the project developed, the tasks I accomplished during this period of four months, the problems encountered and the results.

¹ SSH : https://en.wikipedia.org/wiki/Secure_Shell

² A karo references the control card that is contained in each charging point and that is linked to the SD card that contains all the configuration. Ka-Ro Tx28 <https://www.karo-electronics.com/906.html?&L=1>

³ Npm is a package manager for Javascript.

⁴ Node.js is a software that executes Javascript on different platforms.

⁵ Angular is a client-oriented framework based on Typescript.

⁶ Typescript is a programming language that contains Javascript and more and is transpiled to Javascript since browsers can't interpret it.

⁷ Less is an advanced version of CSS.

⁸ Bootstrap is a front-end framework for styling with Html, Css, etc. Bootstrap4 was used in this project.



Introduction

Dans le cadre de mon bachelier en informatique et systèmes, j'ai réalisé mon stage chez Powerdale, entreprise qui existe depuis 2003 et qui est fortement présente dans le marché des véhicules électriques.

Ce stage a été effectué en développement web, domaine auquel je m'intéresse beaucoup et qui ne cesse d'évoluer.

J'ai travaillé sur deux projets durant cette période de stage, un qui est fini mais qui n'est pas à l'abri de modifications éventuelles et l'autre qui est toujours en cours.

Le premier projet, « prod-screen », a consisté à développer une application web pour un affichage en temps réel afin que l'équipe de production l'utilise pour vérifier les données et faire les tests sur les bornes de chargement.

Le deuxième projet, pur front-end, a consisté en l'amélioration du design d'un projet de facturation, appelé « Nexxtmove service management ».

Dans ce rapport, je présenterai dans un premier temps l'entreprise et ses domaines ainsi que mon travail.

Ensuite je parlerai plus en détail des missions de stage réalisées, des outils utilisés, des problèmes rencontrés et des résultats.

Présentation générale

2.1 Présentation de l'entreprise



Powerdale SA est une société anonyme qui a été créée en 2003 et qui ne comptait pas plus d'une dizaine de personnes lors de ses débuts. Aujourd'hui, l'équipe s'élargit de plus en plus, elle compte près d'une quarantaine d'employés et est principalement présente dans deux secteurs : l'électromobilité et la gestion énergétique. Le concept sur lequel se base l'entreprise est « **EM²** » pour à la fois Energy Management & Electrical Mobility.

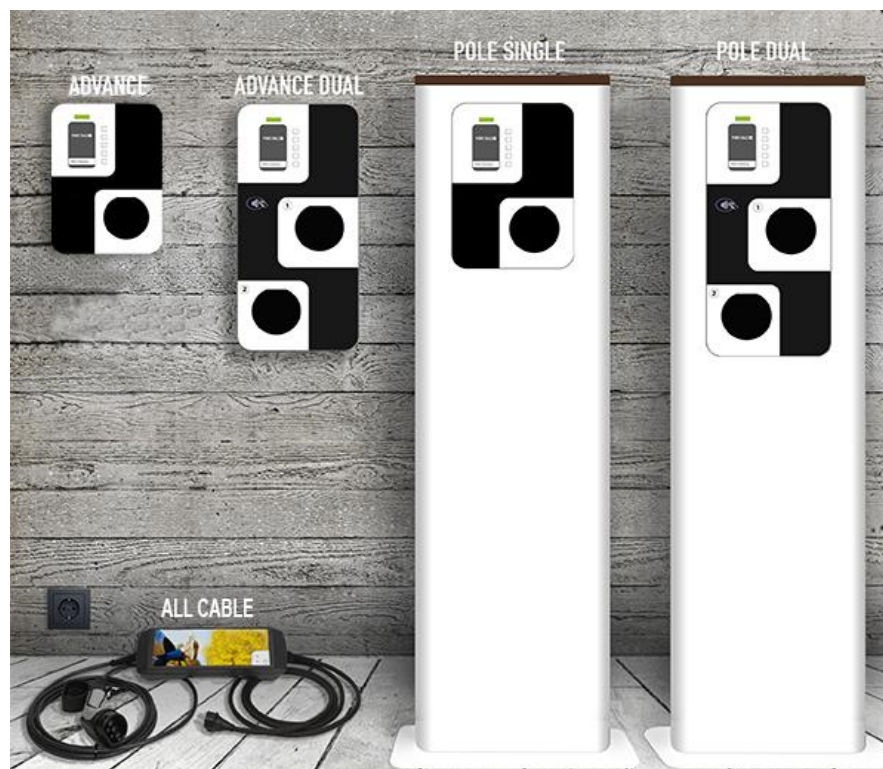
D'un côté, cette entreprise offre des solutions complètes de chargement pour véhicules électriques ainsi qu'une plateforme de gestion en ligne pour permettre aux utilisateurs de facilement surveiller leurs activités de chargement. Les solutions proposées sont basées sur les besoins finaux du client et la vente se fait indirectement, via des partenaires.

En plus du domaine des véhicules électriques, l'entreprise s'occupe aussi du déploiement de solutions de monitoring énergétique adaptées pour les entreprises industrielles afin de leur donner une vision détaillée des consommations concernant, entre autres, l'eau, le gaz et l'électricité.

Powerdale détient également une expertise dans la mise en œuvre de services de gestion dynamique de la demande, ce qui fait d'elle un partenaire de choix pour Electrabel et Engie, avec qui elle a déjà assuré le suivi énergétique de clients industriels et introduit la solution CarPlug, qui fait partie d'une ligne complète de produits et services dédiés au chargement des véhicules électriques.

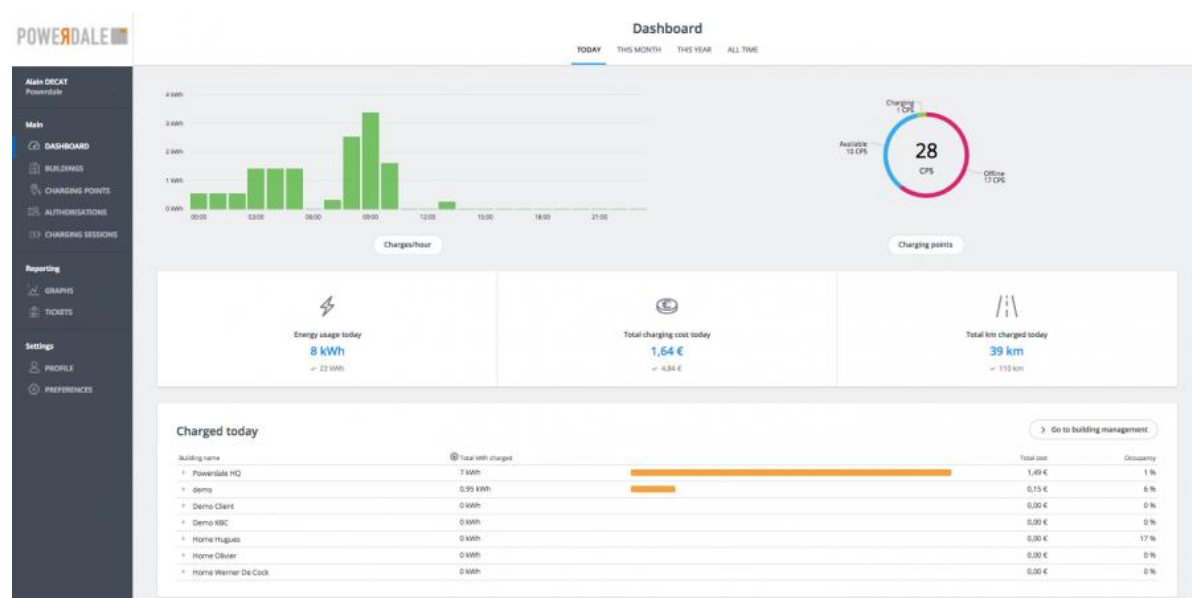
Powerdale poursuit le développement de solutions innovantes, que ce soit au niveau hardware ou software, autour de la gestion énergétique et l'électromobilité en incluant les processus de facturation et la gestion intelligente du processus de chargement. Leurs produits sont marketés sous les noms de Nexxtender pour les bornes de chargement, et Nexxtmove pour la plateforme en ligne qui soutient les processus d'électromobilité.

Il y a plusieurs bornes de chargement allant de la plus basique à la plus interactive et intelligente. Par exemple, Nexxtender Advance qui intègre l'électromobilité et la gestion énergétique et qui est équipée d'un écran qui facilite son usage.



Les différents types de bornes existants

Pour ce qui est plateforme software en ligne Il existe trois versions : Nexxtmove Driver pour tout particulier souhaitant suivre de près les données de chargement de son véhicule, Nexxtmove Business utilisée dans un contexte d'entreprise et Nexxtmove Partner développée pour l'administration d'un environnement de services dédié à l'électromobilité.



Plateforme Nexxtmove

Les départements existants dans cette entreprise sont les suivants :

- La production, l'IT, les opérations, les sales, l'administration et comptabilité et le département RH.

Au sein de cette société, il m'a été aisé de percevoir l'interaction constante entre les différents services décrits plus haut, compte tenu du fait que la première partie de mon stage a été réalisée au sein du département IT, en collaboration avec mon maître de stage, et que le test et déploiement ont été faits dans l'atelier de la production.

Au niveau des outils utilisés, j'ai travaillé avec de nouveaux frameworks, langages de programmation et bibliothèques, je devais donc être autonome et apprendre au plus vite.

La partie serveur de l'application web développée durant la première partie du stage est écrite en :

- **Typescript** : sur-ensemble de Javascript et qui est compilé et traduit en Javascript pour qu'il soit possible pour un browser de l'interpréter puisque Typescript ne peut être interprété par un navigateur.
- **Node.js** : plateforme logicielle orientée Javascript.

Pour la partie graphique, elle a été faite avec **Angular6** qui est un framework de développement d'applications web basé sur Typescript.

Le code tourne sur un Raspberry Pi branché à un moniteur de l'équipe de production.

Pour le deuxième projet, les outils suivants ont été utilisés pour tout ce qui est mise en forme des pages web :

- **Less** : langage dynamique de génération de Css.
- **Bootstrap** : framework front-end Css.

L'entreprise utilise des outils de communication et versioning tels que Office ou gitlab.

En ce qui concerne l'analyse et la gestion de projet, l'entreprise s'appuie sur la méthode agile d'analyse Scrum. Dans Scrum il existe des rôles pour chaque membre de l'équipe : le scrum master qui guide l'équipe, l'équipe qui peut être constituée de développeurs, analystes ou autres et le product owner.

La vie d'un projet Scrum est rythmée par un ensemble de réunions clairement définies et strictement limitées dans le temps. Le cycle de vie Scrum est basé sur des itérations de quelques semaines appelées « sprints » qui découpent le projet en plusieurs sous-projets. Ces sprints permettent de fixer des objectifs concernant une partie du projet à réaliser dans un délai précis.



Généralement en fin de chaque semaine, une réunion de toute l'équipe est organisée au cours de laquelle on présente l'avancement des fonctionnalités sur lesquelles on travaille et on anticipe ce qui peut être fait durant la période de temps restante par rapport au sprint, aux spécifications du projet et aux exigences du client qui peut à tout moment proposer de nouvelles modifications ou des ajouts.

Avec cette méthode, on voit concrètement l'évolution du projet car après chaque itération, il est possible de visualiser un bout de projet qui fonctionne, ce qui n'est pas forcément possible lorsqu'on adopte d'autres méthodes telles que Merise ou le modèle du cycle en V. Cette évolution permet de prioriser les besoins réels du client et l'application s'enrichit au fur et à mesure.

2.2 Présentation du travail

2.2.1 Moniteur de production en temps réel

Les membres de l'équipe de production font des tests sur les bornes avant de les vendre aux partenaires. Ils ont constamment besoin d'accéder aux données de configuration des bornes afin de les modifier ou les réinitialiser. Cela requiert parfois le redémarrage de la borne. Travailler en parallèle devient difficile et les données de calibration peuvent facilement se perdre.



Bornes de chargement

Le travail consiste donc en la réalisation d'une application web qui permet un affichage instantané des informations des bornes de chargement. Cela permettra de gagner plus de temps en travaillant ainsi en parallèle et d'avoir un outil de vérification de la sauvegarde des données injectées.

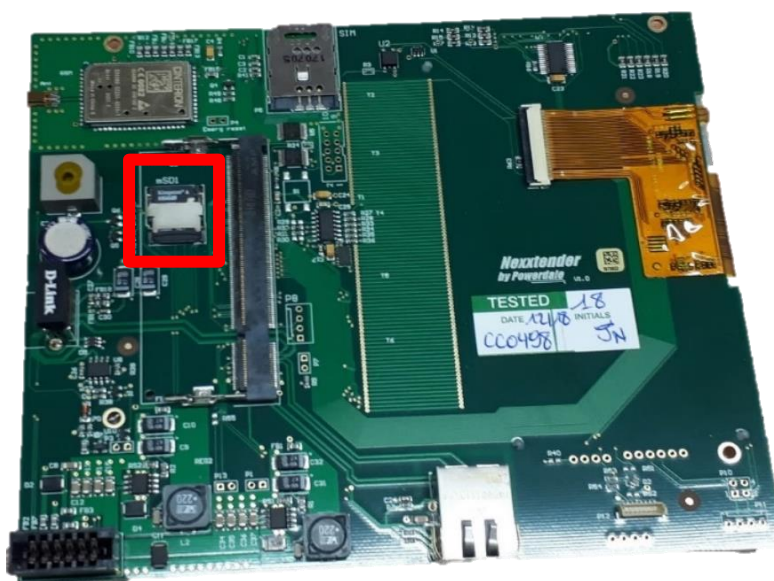
Les bornes de chargement ont une carte contrôle qui est reliée à un Karo⁹, une carte SD et une carte SIM.

Le Karo est un ordinateur complet implémenté sur une petite carte et prêt pour être utilisé dans les systèmes embarqués. Il est doté d'un système d'exploitation Linux et contient quelques données de configuration.

Il est relié à la carte SD qui, elle, contient une base de données locale dans laquelle on trouve la configuration nécessaire à l'affichage comme les données de calibrations, l'adresse IP, les versions etc.

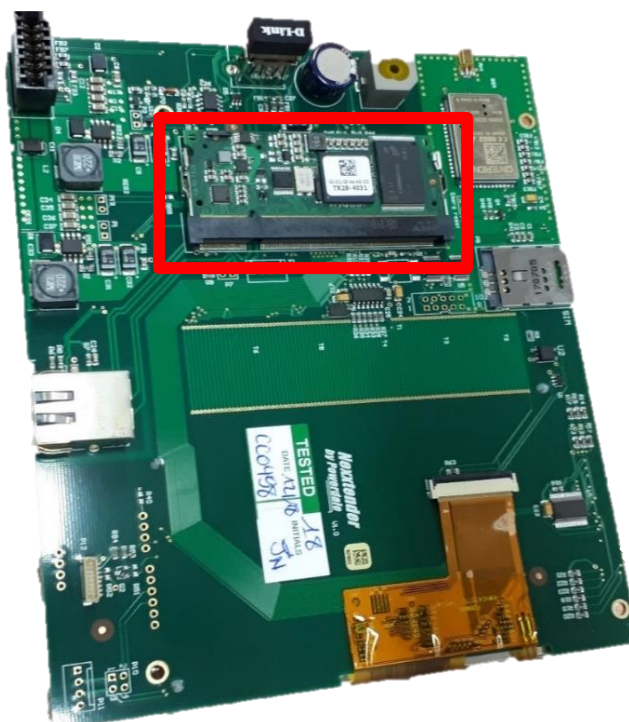
La carte SIM, quant à elle, permet d'avoir une adresse Gprs pour pouvoir accéder à la borne à distance.

Cette configuration donne une certaine flexibilité en cas de perte de données ou d'endommagement de la carte SD par exemple, et son coût est acceptable.



Carte contrôle avec carte SD

⁹ Karo appelé ainsi suivant sa marque de fabrication : Ka-Ro TX28 : <https://www.karo-electronics.com/906.html?&L=1>



Carte contrôle avec Karo inséré

Le projet existant, appelé « karo-scanner », est une application web qui permet de scanner en ssh¹⁰ (Secure Shell) une adresse IP donnée d'une borne. Elle donne aussi la possibilité de détecter les erreurs liées à celle-ci et les signaler, d'avoir l'historique de recherche, et pour y accéder un login et mot de passe sont indispensables.

Le projet sur lequel j'ai travaillé se nomme « prod-screen ». C'est une application web dont le but est de scanner une plage d'adresses IP donnée toutes les 3 minutes à la recherche de karos existants sur ce range. Si elle en trouve elle en affiche les informations, telles que l'id, la date de mise sous tension, date du dernier scan, les adresses IP etc.

Ce projet utilisera les classes et fonctions de scan du projet « karo-scanner ».

2.2.2 Amélioration du design

En ce qui concerne le deuxième projet, « Nexxtmove service management », celui-ci consiste en un projet de facturation destiné au service de comptabilité, il avait déjà débuté, il y avait donc une base sur laquelle j'ai travaillé. Comme indiqué plus haut le travail qui m'a été confié est d'améliorer le style des pages web qui constituent le projet.

¹⁰ SSH : https://en.wikipedia.org/wiki/Secure_Shell

Travail réalisé

3.1 Prod-screen

L'application reçoit un range d'adresses IP et essaie de se connecter en ssh sur chacune des adresses. La connexion est établie quand un Karo se trouve à l'adresse IP scannée et que la clé fournie est la bonne.

3.1.1 Analyse

Avant toute conception technique de projet il y a eu une phase d'apprentissage pendant laquelle j'ai fait connaissance avec tous les frameworks, librairies et langages de programmation qui m'étaient imposés tels que Typescript, Node.js, Rxjs¹¹ ou Angular6. Les 2 premières semaines du stage ont été donc consacrées à l'apprentissage et la compréhension des outils, de l'environnement et des tâches à réaliser.

Ensuite, des code reviews ont été faites pour comprendre le projet existant et son fonctionnement et de le comparer au travail demandé.

Après avoir eu une idée assez claire de ce qui m'était demandé, j'ai mis en pratique ce que j'ai appris tout en suivant les conseils de mon responsable de stage avec qui des réunions ont eu lieu afin de vérifier les fonctionnalités implémentées et l'avancement du projet.

Quand le projet prenait forme et qu'il y avait quelque chose à montrer à l'équipe de production, je me suis installée chez eux et ai réalisé toutes les modifications nécessaires et adéquates à leurs demandes. J'ai eu la chance de travailler avec différentes équipes et dans différents environnements de travail.

La dernière partie de ce projet consistait en l'installation du Raspberry Pi et la documentation ainsi que l'adaptation de certaines fonctionnalités au fur et à mesure qu'on remarquait des améliorations ou changements à faire.

¹¹ Angular Docs. Rxjs, <https://angular.io/guide/rx-library>

Pour avoir une vue globale d'avant/après le travail fait durant mon stage voici des diagrammes du « karo-scanner » ainsi que du « prod-screen » pour mieux comparer les deux et comprendre les ajouts qui ont été faits :

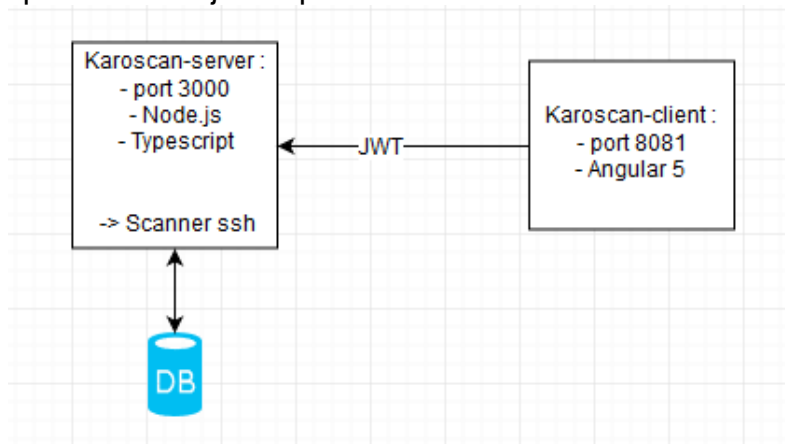


Diagramme représentant le « karo-scanner » avant.

- L'authentification du client auprès du serveur se fait via des JSON Web Token (JWT)¹².

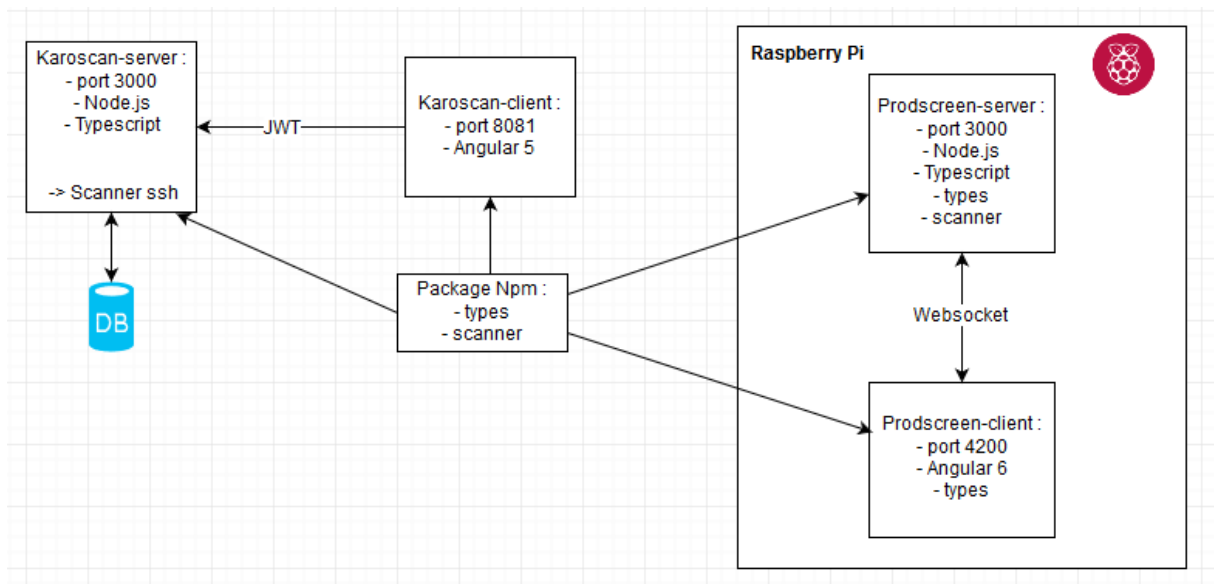


Diagramme représentant le lien entre le « prod-screen » et le « karo-scanner » après.

- Le « prod-screen » utilise les Websocket pour la communication.
- Ajout du package npm.
- Le client utilise Angular6.
- Le serveur quant à lui il est aussi en Typescript et Node.js.

¹² JWT : <https://jwt.io/>

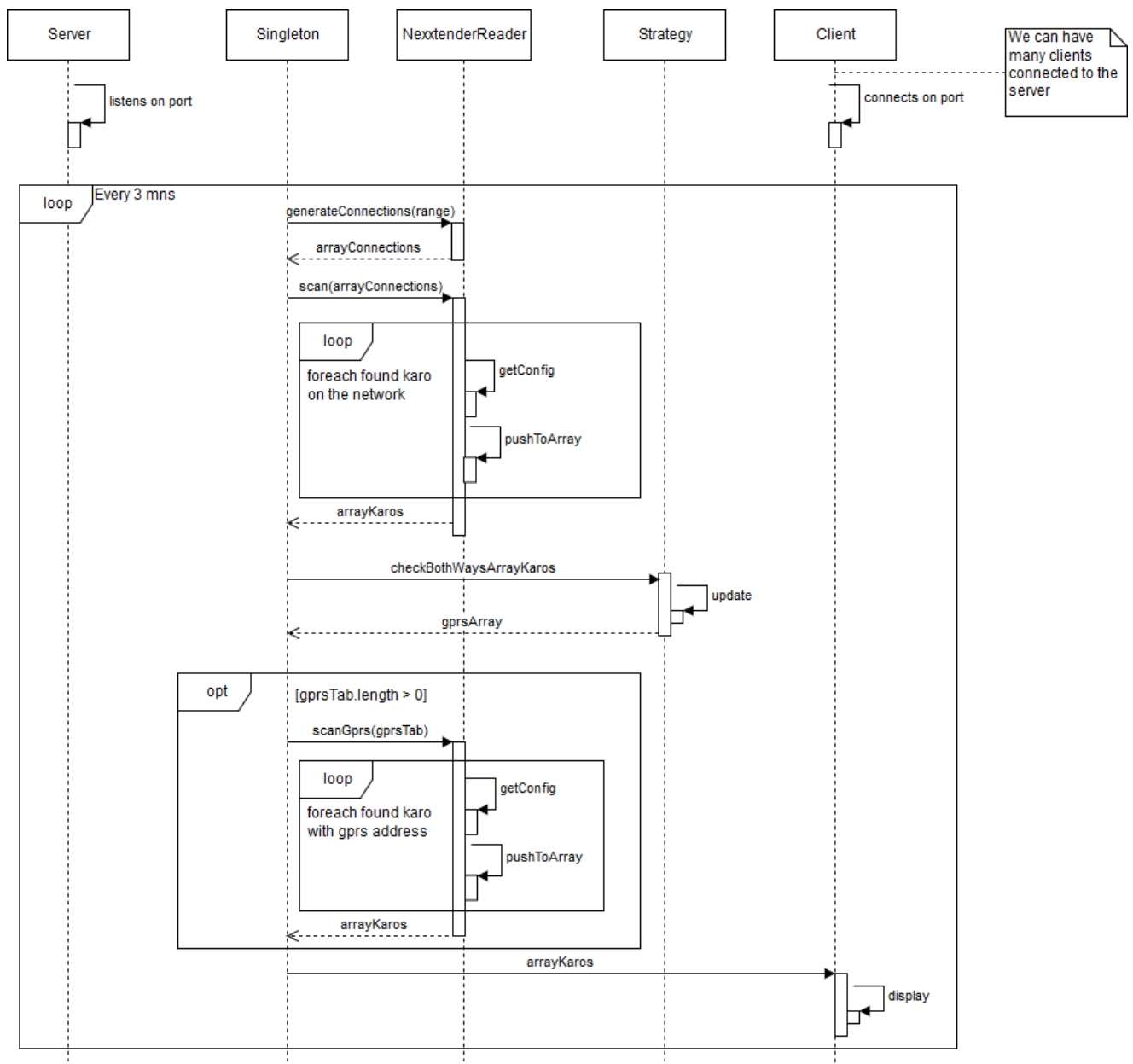


Diagramme de séquence du « prod-screen »

3.1.2 Serveur

Lorsque le serveur est lancé, il écoute sur un port donné et débute le scan. Le gathering d'informations se fait au niveau du serveur. Pour chaque Karo trouvé, les données de configuration sont copiées et ajoutées au tableau qui sera envoyé par la suite au client, ou ce qu'on appelle le stack, qui est vide au début mais qui contient ensuite tous les karo scannés en Lan ou en Gprs.

Par défaut le scan se fait sur des adresses IP locales. Il se peut que les bornes soient déconnectées physiquement du Lan (Local area network), donc du réseau local et soient connectées en Gprs (General Packet Radio Service), ce qui veut dire qu'elles ne seront plus détectées via le scan Lan.

Pour ce faire, un deuxième scan a été prévu, qui scanne les adresses gprs non détectées en Lan après un certain temps et la procédure de mise à jour et de filtrage restent les mêmes que pour le scan Lan.
Les données sont ainsi mises à jour après chaque scan et envoyées via Websocket.

3.1.3 Client

Le client est le navigateur web dans notre cas. Lorsque celui-ci se connecte sur le port, il attend de recevoir les données. Le traitement de l'affichage et l'affichage se font à ce niveau-ci.

Si un karo se déconnecte, on lui donne un certain temps avant de commencer à réduire petit à petit l'opacité de son affichage sur l'écran jusqu'à ce qu'il soit supprimé, s'il réapparaît son affichage redevient normal.

Différentes informations sont affichées : l'id du karo, son logo, la date et l'heure du dernier scan, la date et l'heure de démarrage, si la connexion et la synchronisation avec la plateforme ont été faites avec succès, si la calibration et les données injectées sont bonnes, le type et nombre des connecteurs, l'ampérage, les modes d'activation et de connexion, les langues et les adresses IP.

Le client Websocket est créé dans un service qui peut être injecté dans d'autres composants si besoin. La séparation des fonctionnalités de la vue des autres fonctionnalités permet d'éviter qu'un composant reçoive ou manipule des données directement et permet d'augmenter la réutilisabilité et la modularité du service.

3.1.4 Outils et protocoles

- Typescript

TypeScript TypeScript est un langage de programmation qui constitue un sur-ensemble de Javascript, c'est-à-dire que tout code Javascript peut être utilisé dans Typescript.

Javascript étant le plus souvent utilisé pour gérer des événements, il est plus connu pour son utilisation au niveau client, mais il peut aussi être utilisé au niveau du serveur.

Une des différences qui est un avantage de Typescript est que contrairement à Javascript dans lequel la notion de classe n'existe pas, Typescript améliore la programmation orientée objet. Il existe des classes, des interfaces des fonctions, des variables, mais surtout du typage de variables.



Il est aussi possible de faire des imports de modules et des déclarations de librairies externes.

<pre>1 class voiture { 2 passagers = 4; 3 constructor(personnes : number) { 4 this.passagers = personnes; 5 } 6 mamethode(message : string) { 7 console.log(message); 8 } 9 } 10 } 11 12 var mavoiture = new voiture(2);</pre>	<pre>1 var voiture = (function () { 2 function voiture(personnes) { 3 this.passagers = 4; 4 this.passagers = personnes; 5 } 6 voiture.prototype.mamethode = function (message) { 7 console.log(message); 8 }; 9 return voiture; 10 })(); 11 12 var mavoiture = new voiture(2); 13</pre>
--	---

Code Typescript (à gauche) et sa transpilation en Javascript (à droite)

Typescript supporte aussi la généricité et le scoping de fonctions c'est-à-dire le raccourcissement de la syntaxe des fonctions. À travers cela, on remarque le rapprochement avec Java.

```
var add = function(a:number,b:number):number {
  return a + b;
}
```

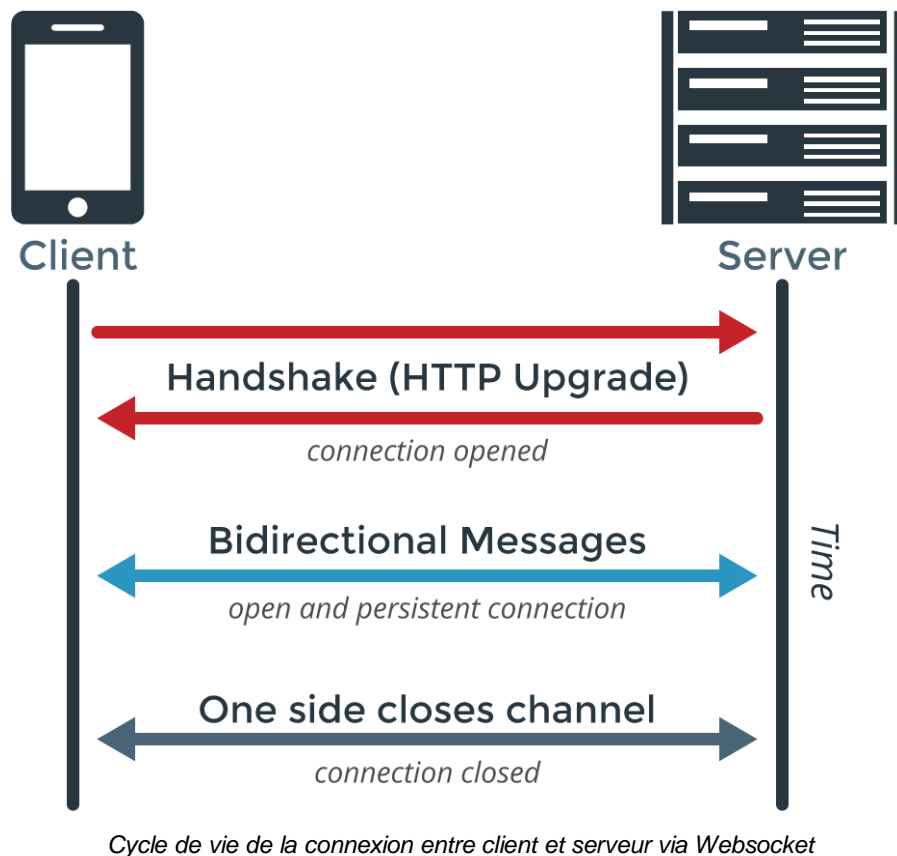
Scoping de fonction

Les génériques sont des composant réutilisables, qui peuvent fonctionner sur plusieurs types plutôt qu'un seul type défini. Cela permet à l'utilisateur de consommer ces composants et d'utiliser ses propres types pour une plus grande vitesse d'exécution.

```
function identity<T>(arg: T): T {
  return arg;
}
```

Fonction générique

- Websocket



Websocket est un protocole de communication, tout comme Http¹³ (HyperText Transfer Protocol), mais qui permet d'avoir un canal full-duplex entre un client et un serveur, c'est-à-dire qu'il permet aux deux communicants d'échanger indépendamment l'un de l'autre.

En Http, le client ne peut envoyer une nouvelle requête que si le serveur a terminé de répondre à la requête précédente.

Il n'existe pas de mécanisme de message prédéfini ou de request/response. Il faut donc qu'un serveur écoute sur un port et qu'un client se connecte sur le même port.

La connexion est établie via un Http handshake (établissement d'une liaison), les deux communicants envoient leur handshakes et si l'établissement de la liaison est réussi, le transfert de messages peut débuter.

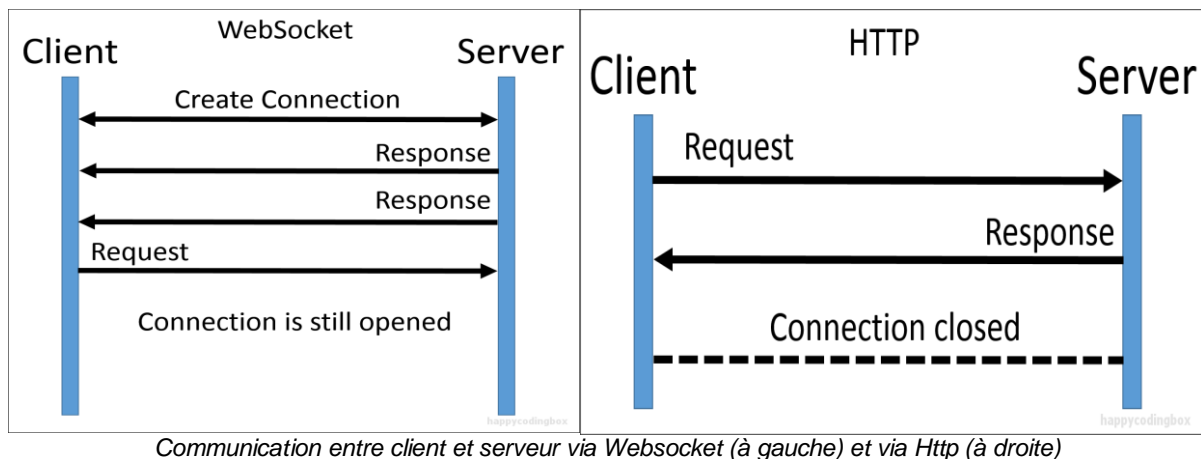
Http est un protocole unidirectionnel ou le client initialise la connexion, envoie la demande, le serveur la reçoit et répond au client (request/response). Ils ne peuvent donc communiquer que s'il y a eu cette demande.

¹³ Http : https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Websocket, quant à lui, est un protocole bidirectionnel : aussi bien le client que le serveur peut envoyer des données.

Le besoin de création d'applications web qui requièrent une communication bidirectionnelle entre le client et le serveur demande une utilisation assez lourde du protocole Http du fait que le serveur utilise plusieurs connexions TCP par client aussi bien pour envoyer des données au client que pour recevoir chaque message envoyé par le client.

De ce fait, le but de cette technologie est d'offrir un mécanisme pour ces mêmes applications, qui ne demande pas l'ouverture de plusieurs connexions TCP.



En outre, ils ont aussi un mécanisme de détection des clients qui se déconnectent et contrairement au protocole Http qui ferme la connexion TCP établie et ouvre une nouvelle à chaque nouvelle requête, les Websocket font un « upgrade¹⁴ » de la connexion établie au début ainsi le serveur et client communiquent via la même connexion TCP tout au long du cycle de vie de la communication Websocket. Ainsi, lorsque le client se déconnecte, ou que le serveur se déconnecte, cette connexion TCP est fermée.

- Npm & Node.js



Node.js est une plateforme logicielle événementielle orientée Javascript. Elle permet d'utiliser Javascript côté serveur en offrant un environnement serveur. Il existe différentes plateformes pour Node.js telles qu'Express, utilisé dans le projet « prod-screen ». Node.js est considéré comme concurrent de PHP, qui lui est un langage de programmation orienté serveur, plus connu et plus utilisé.

¹⁴ Upgrade Header : <https://tools.ietf.org/html/rfc2616#section-14.42>



Npm est un gestionnaire des packages Javascript et le gestionnaire officiel pour Node.js. Les packages contiennent des librairies ou modules qui peuvent être inclus dans les projets.

Le package npm créé, appelé « nexxtender scan lib », est une librairie constituée de classes, interfaces et méthodes relatives au scan et aux différentes fonctions d'accès et de manipulation des bornes en ssh, qui existaient dans le « karo-scanner ».

Il permet aux projets qui y sont liés de les utiliser et est utilisé par le « karo-scanner » ainsi que le « prod-screen ».

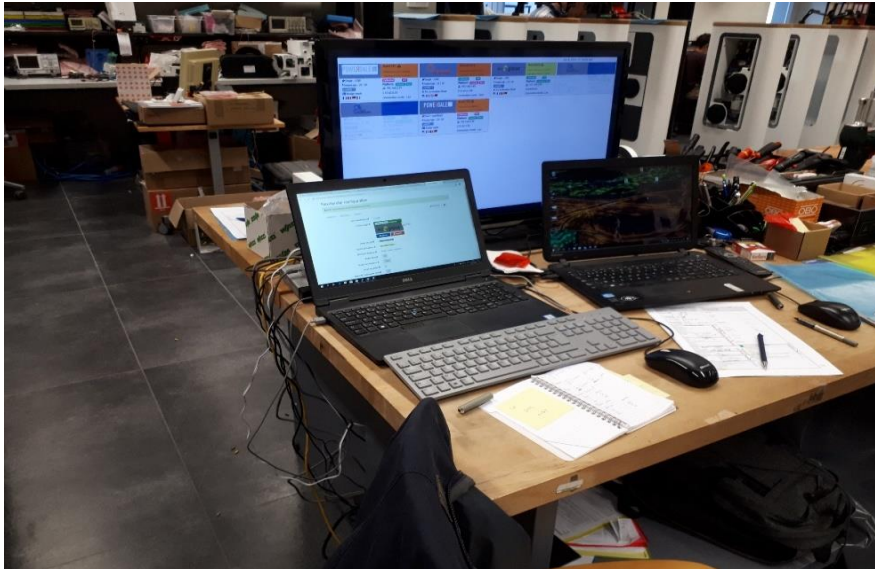
3.1.5 Résultat final

L'installation est faite sur un Raspberry Pi qui lance automatiquement le programme au démarrage et fait un affichage en plein écran.

L'automatisation est faite via la modification de quelques fichiers qui sont exécutés au démarrage du Raspberry Pi.

POWERDALE <small>EM2</small>		POWERDALE <small>EM2</small>	
Karo161		Karo795	
Online since Jan 10 04:00:08 Last scan 1/10/19, 12:02 PM		Online since undefined Last scan 1/10/19, 12:02 PM	
Single : ADE	Calibration CFD	Dual : undefined	Calibration CFD
Amperage : 26 10	Platform : Connect. Sync	Amperage : 32 10	Platform : Connect. Sync
LoadCtrl : 1	192.168.6.67	LoadCtrl : 1	192.168.6.54
Badge mode	10.42.0.72	Badge mode	10.42.2.52
	Connection mode : Lan		Connection mode : Lan

Affichage des données de 2 bornes



Moniteur dans l'atelier de la production

3.2 Nexxtmove service management

Une réunion a été organisée avec l'équipe de développement et une revue de code a été faite, suivis de l'installation des différents serveurs et outils nécessaires pour mieux cerner le travail attendu concernant le « Nexxtmove service management ».

- Less



Less est aussi un langage de mise en forme des pages web tout comme CSS¹⁵ (Cascading Style Sheets). Il fonctionne aussi bien côté client que côté serveur, avec Node.js. L'utilisation de CSS dans du code Less est possible puisqu'il est compatible avec Less. Less étend le CSS avec un comportement dynamique, utilisant des variables ou constantes comme suit :

```
1  /* Déclaration */
2  @blue: #17B6FF;
3  @magenta: #FF17B6;
4  @lime: #B6FF17;
5  @green: #1BFF17;
6
7  /* Utilisation */
8  #header { background: @blue; }
9  #footer { background: @magenta; }
```

Déclaration et utilisation de variables en Less

¹⁵ Ccss est un langage qui met en forme les pages web.

, des imbrications :

```
1 /* LESS */
2 #foo {
3     background: lightgrey;
4     &:hover { color: orange; }
5 }
6
7 /* CSS généré */
8 #foo { background: lightgrey; }
9 #foo:hover { color: orange; }
```

Utilisation des imbrications en Less et équivalent en Css

, des opérations :

```
1 @size_image: 100px;
2
3 .box {
4     width: @size_image + 10px;
5     height: @size_image + 10px;
6
7     img {
8         width: @size_image;
9         height: @size_image;
10    }
11 }
```

Utilisation des opérations

Ou encore des fonctions qu'il propose, des inclusions ou imports, etc..

- Bootstrap



Bootstrap est une librairie qui offre différents outils utiles à la création du design de sites et d'applications web. C'est un framework Css qui embarque également des composants Html et Javascript. Différentes versions existent, la plus récente a été utilisée : Bootstrap4.

Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web.

Il apporte du style pour les boutons, les formulaires, la navigation... Il permet ainsi de concevoir un site web rapidement et avec peu de lignes de code ajoutées.

Ce framework front-end est constitué lui-même de feuilles de style Less qui implémentent différents composants.

Voici un exemple de ce que peut-être du code Bootstrap :

.col-xs-12 .col-md-8		.col-xs-6 .col-md-4	
.col-xs-6 .col-md-4		.col-xs-6 .col-md-4	
.col-xs-6		.col-xs-6	

```

<!-- Stack the columns on mobile by making one full-width and the other half-width -->
<div class="row">
  <div class="col-xs-12 col-md-8">.col-xs-12 .col-md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
<div class="row">
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- Columns are always 50% wide, on mobile and desktop -->
<div class="row">
  <div class="col-xs-6">.col-xs-6</div>
  <div class="col-xs-6">.col-xs-6</div>
</div>

```

Copy

Système de colonnes de Bootstrap

Bootstrap propose des composants pratiques intégrés à mettre en œuvre : barres de navigation, effets typographiques, panneaux, etc. mais aussi des éléments de base tels que les formulaires, les tableaux, les listes...

```

1 <form class="col-lg-6">
2   <legend>Légende</legend>
3   <label for="text">Text : </label>
4   <input id="text" type="text" class="form-control">
5   <label for="textarea">Textarea : </label>
6   <textarea id="textarea" type="textarea" class="form-control"></textarea>
7   <label for="select">Select : </label>
8   <select id="select" class="form-control">
9     <option>Option 1</option>
10    <option>Option 2</option>
11    <option>Option 3</option>
12  </select>
13  <br>
14  <button>Envoyer</button>
15 </form>

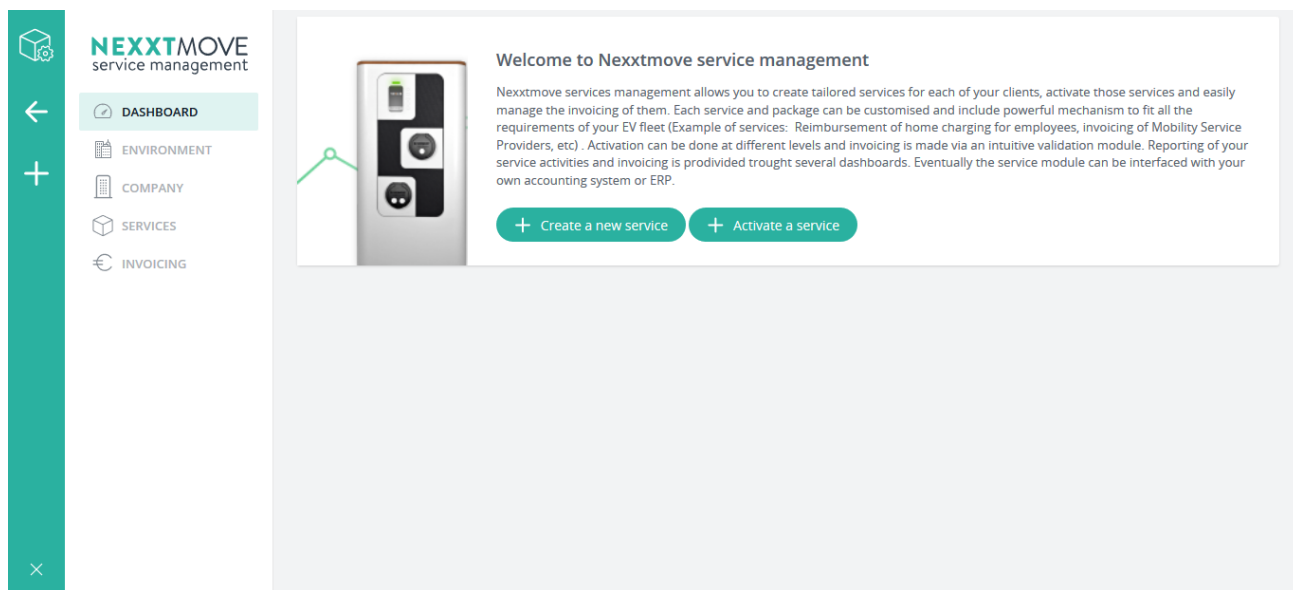
```

Un formulaire en Bootstrap

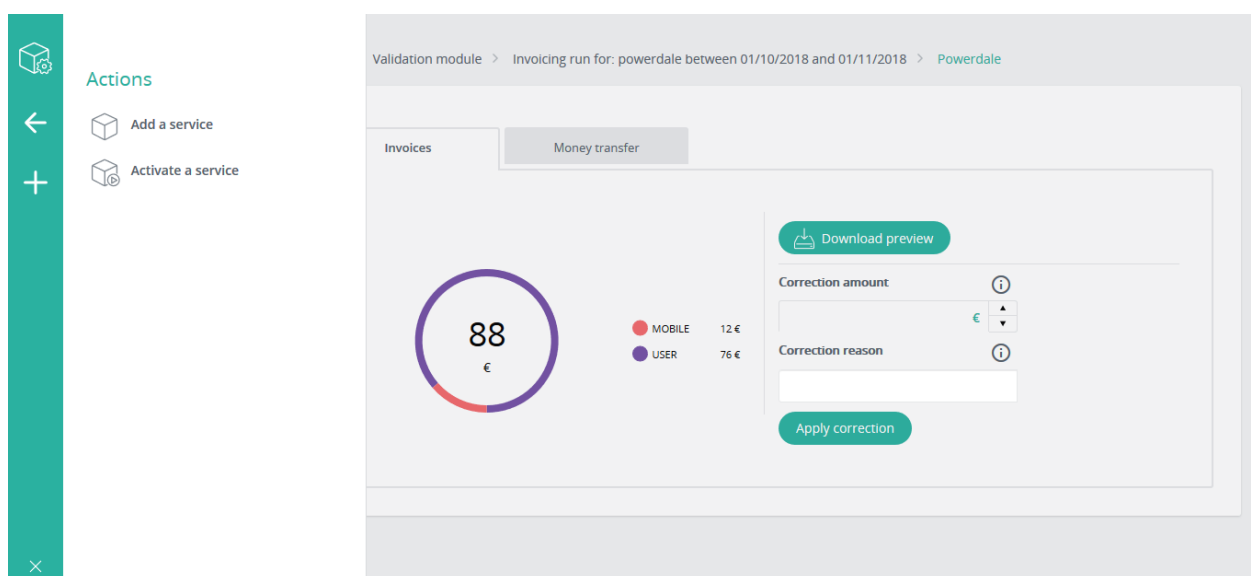


- Résultats

Voici quelques-unes des pages web qui constituent le « Nexxtmove service management ».



Dashboard



Module de validation et actions

Conclusion

Durant ces quatre mois de stage effectué chez Powerdale, ma mission première était de mettre en place un écran d'affichage en temps réel qui montre les informations des bornes mises en test à la production.

Ensuite, j'ai fait partie d'un deuxième projet et ma tâche était d'améliorer le style des pages d'une application web qui est en cours de développement, je me suis donc intégrée durant toute cette période en tant que développeuse web.

J'ai fait partie de différents projets et j'ai pu voir les différences, les niveaux de difficulté entre les deux projets, et j'ai appris beaucoup de nouveaux concepts en une courte période. Ceci m'a permis de mettre en avant mes compétences dans ce domaine en contribuant à l'évolution de projets qui diffèrent de ceux vus à l'école et de gagner en compréhension.

J'ai compté sur mes compétences acquises au cours de ma formation, mais j'ai dû faire preuve d'autonomie et d'auto-apprentissage pour travailler avec tous les nouveaux outils mis à disposition et réaliser tout ce qui était attendu de moi.

À travers ceci, j'ai expérimenté la réalité du monde du travail et j'ai pu mettre en œuvre mes compétences professionnelles et humaines.

Ce stage a été une première vraie opportunité pour moi de mettre en valeur mes acquis et de les exploiter à travers différents projets concrets.

Annexes

Documentation « prod-screen »

Prod-screen

Doc

Prod-screen is a software tool that scans an ips range on a network and searches for Karos every 3 minutes. If at least one is found, it starts fetching data from its sd card.

- Prod-screen-server

It listens on port 8999 and starts scanning the ip addresses using ssh on the network given which is '192.168.6.0' from ip '41' untill ip '120' both included(hardcoded for now). The database configuration on the sd card is assigned to each karo found on the network. By default, the tool scans ethernet ip addresses. The Karo class[1] contains a failCount number that is set to 10 and a connectionMode that is set to 0, 1 or 2 depending on the connection mode of the karo which can either be lan, gprs or wifi respectively. The Strategy class[2] compares both scanned-karos array(or mapped-karos array) and the server's stack(the already displayed karos). It adds or updates karos found on the scanned-karos array depending on if they already exist on the stack or not, and checks the other way round, meaning it checks karos that are in the stack and that weren't found in the scan, which could either mean that they are turned off(if failCount is 0 the karo is deleted from the stack) or that they are switched to connection mode gprs. In the first case we just decrement the failCount. Otherwise we do the gprs scan on every one that wasn't found meaning every time a karo has failCount set to 7. If found with the second scan, the karos are updated with the actual connection mode and parameters. After that they are sorted and sent to the client via websockets. If the gprs karos are back in lan or wifi mode, when scanned again everything is updated.

- Prod-screen-client

Once the client receives the karos, they are groupped by their logo and displayed. Depending on the failCount,the opacity of the displayed info may decrease. The data displayed contains : logo,if successfully synced, amperage, connectors values, activation mode(s), languages, id, uptime date, last scan date, if calibration parameters are ok, if the the connection to nexxtmove was successful, eth and ppp addresses if any and the connection mode. Note that the connection mode and the icon next to the karo id may be different if the lan cable is still plugged in but the connection mode is set to gprs, this will set the icon to lan mode.

- Websockets

Websockets are used here to transfer data from server to client and vice versa.



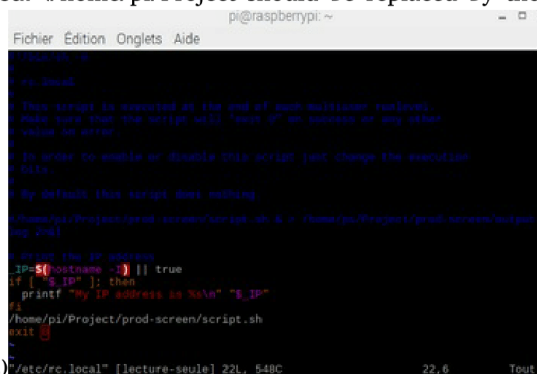
- Npm package

The project uses a library that contains necessary types for both the server and the client. It also contains the scanner class that makes the scan possible through ssh connections.

Installation on a Raspberry Pi

Assuming we have an existing Raspbian on the Raspberry Pi's sd card and it's connected to the network, and that Node.js, typescript, angular/cli, npm are all installed on the Raspberry Pi, run the script file **update.sh** that is in the project's directory **prod-screen** so that the files are compiled and everything is up to date. To run the software at start-up, 3 files should be modified as follows :

- **/etc/rc.local** (/home/pi/Project should be replaced by the path where the



```

pi@raspberrypi ~
Fichier Édition Onglets Aide
#!/bin/sh

# This script is executed at the end of each boot session.
# Make here the script will "test if" on success or any other
# value on error.

# In order to enable or disable this script just change the execution
# bit.

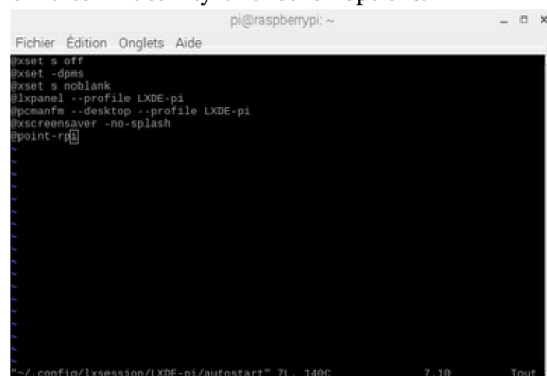
# By default this script does nothing.

# Uncomment the IP address
IP=$(hostname -i) || true
if [ "$IP" ]; then
  printf "My IP address is %s\n" "$IP"
fi
/home/pi/Project/prod-screen/script.sh & > /home/pi/Project/prod-screen/output.log 2>&1
exit 0

```

project is) **/etc/rc.local** [lecture-seule] 22L, 548C 22,6 Tout

- **~/config/lxsession/LXDE-pi/autostart** (screen saver off, screen doesn't turn off after inactivity and other options)



```

pi@raspberrypi ~
Fichier Édition Onglets Aide
Exec s off
Exec -dpms
Exec s noblank
Execpanel --profile LXDE-pi
pcmanfm --desktop --profile LXDE-pi
lxscreensaver -no-splash
@point-rf

```

- **~/config/autostart/auto.desktop** (autostart directory should probably be created. The Exec line starts Chromium browser on kiosk mode and goes to the url localhost:8080)

Autorisation d'impression



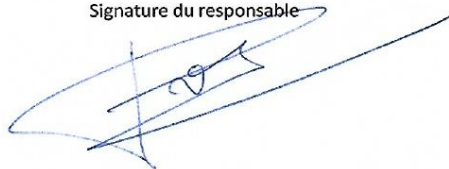
Autorisation d'impression

Je sous signé DELHAN LIEVEN responsable du stage de

M./Melle Najih Fatima Zahra confirme la lecture du rapport de stage et autorise

son impression et sa remise à l'HE2B – ESI

Signature du responsable

A handwritten signature in blue ink, appearing to be 'DELHAN LIEVEN', written over a large, stylized blue 'X' mark.

signature de l'étudiant

A handwritten signature in blue ink, appearing to be 'Najih Fatima Zahra', written in a cursive style.

Ce document doit être remis - complété et signé - le jour de la remise du rapport de stage

Evaluation continue du stage



STAGE EN INFORMATIQUE 17-18 : EVALUATION CONTINUE DU
STAGE PAR LE RESPONSABLE AU SEIN DE L'ORGANISME.

Document du stagiaire

Nom de l'Organisme Powerdale
Personne qui évalue le stage
Nom, Prénom Jaffaride Alban Tél/GSM 04 74354974
Concerné l'étudiant: Nom Najibi Prénom Fatima Zahra
Echelle de valeurs proposées
Insuffisant --- Faible -- Très Moyen - Moyen M Bon + Très bon ++ Excellent +++

Au terme du stage l'étudiant a atteint les seuils de compétences suivants :

	mi septembre	mi octobre	mi décembre
FORMATION			
Connaissances théoriques	--- -- - M <u>+</u> ++ +++	--- -- - M + <u>++</u> +++	--- -- - M + <u>++</u> +++
Analyse	--- -- - M <u>+</u> ++ +++	--- -- - M <u>+</u> ++ +++	--- -- - M + <u>++</u> +++
Pratique de la programmation	--- -- - M <u>+</u> ++ +++	--- -- - M + <u>++</u> +++	--- -- - M + <u>++</u> +++
TRAVAIL FOURNI			
Quantité de travail fourni	--- -- - M + <u>++</u> +++	--- -- - M + <u>++</u> +++	--- -- - M + <u>++</u> +++
Qualité du travail fourni	--- -- - M <u>+</u> ++ +++	--- -- - M <u>+</u> ++ +++	--- -- - M + <u>++</u> +++
Adaptabilité au système et à l'environnement	--- -- - M + ++ <u>+++</u>	--- -- - M + ++ <u>+++</u>	--- -- - M + ++ <u>+++</u>
Capacité d'auto-apprentissage	--- -- - M + ++ <u>+++</u>	--- -- - M + ++ <u>+++</u>	--- -- - M + ++ <u>+++</u>
Capacité d'organisation	--- -- - M + <u>++</u> +++	--- -- - M + <u>++</u> +++	--- -- - M + <u>++</u> +++
COMPORTEMENT			
Aptitudes à la communication	--- -- - M + ++ <u>+++</u>	--- -- - M + ++ <u>+++</u>	--- -- - M + ++ <u>+++</u>
Sociabilité	--- -- - M + <u>++</u> +++	--- -- - M + <u>++</u> +++	--- -- - M + <u>++</u> +++
Réceptivité aux recommandations	--- -- - M <u>+</u> ++ +++	--- -- - M <u>+</u> ++ +++	--- -- - M + <u>++</u> +++

Comment remplir la grille

Après une période de +/- 4 semaines, un entretien est réalisé entre le superviseur du stage et le/la stagiaire. Chacun des items de cette grille sont évalués et explicités par le superviseur.
Le/la stagiaire est tenu de prendre note des remarques émises en les transcrivant dans son "Carnet de bord de Stage".

Ce document ainsi que le "Carnet de bord de Stage" DOIVENT être présenté spontanément au professeur de référence, qui pourra en tenir compte pour sa part d'évaluation du stage. Le "Carnet de bord de Stage" fait partie du rapport final



Carnets de bord de stage

Carnet de bord de Stage

A remplir en fin de semaine

Semaine ① 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Angular /TypeScript/ Node js/ Express
- Code review du scanner
- produ-screen et package npm

Compte rendu explicatif

- Apprentissage et utilisation de TypeScript, Angular + différents framework, api et librairies...
- La review du code du scanner a permis de mieux comprendre comment ce dernier fonctionne et quelle sera la partie à développer et/ou adapter.
- Produ-scan est le projet sur lequel je vais travailler, ce sera une adaptation du karo-scan existant.
- Produ-screen (serveur & client) est la partie affichage par laquelle je commencerai, viendra ensuite le package npm.

Notes et remarques personnelles



Semaine 1 (2) 3 4 5 6 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Prod-screen server
- Prod-screen client

Compte rendu explicatif

- Le prod-screen server permettra pour l'instant de faire tourner un Singleton qui aura à exécuter 1 des 3 fonctions(crud) selon un intervalle de temps et aléatoirement.
- Le prod-screen client permettra l'affichage des infos des Karo.

Notes et remarques personnelles

Carnet de bord de Stage

A remplir en fin de semaine

Semaine 1 2 ③ 4 5 6 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Prod-screen server
- Websocket
- Prod-screen client

Compte rendu explicatif

- La partie serveur permet pour l'instant d'afficher en console les informations des Karo scannés selon un intervalle de temps, avec une partie affichage à tester sur le client.
- Les Websockets seront implémentés pour communiquer les infos des Karos entre serveur et client.
- La partie client sera par la suite implémentée sur Angular

Notes et remarques personnelles



Semaine 1 2 3 ④ 5 6 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Prod-screen client en Angular 6

Compte rendu explicatif

- Le serveur tourne et les données sont envoyées, après scan, via des Websockets selon un intervalle de temps. Encore quelques modifications à faire tel que la gestion des erreurs etc.
- Prod-screen client est en cours d'implémentation.

Notes et remarques personnelles

Semaine 1 2 3 4 (5) 6 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Affichage client
- Gestion d'erreurs
- Stratégie

Compte rendu explicatif

- Gestion des erreurs liées aux websocket (connexion, déconnexion, interruption ..)
- Les données liées aux karos sont affichées sous forme de liste après scan toutes les 2 minutes.
- A ajouter : Une stratégie qui permettra d'ajouter ou pas les karos scannés selon s'ils existent déjà dans le stack du client. Un compteur sera ajouté pour supprimer un karo du stack après un certain nombre de check. Si le karo existe celui-ci sera updaté. (Côté serveur)
- Amélioration de l'affichage avec Bootstrap4

Notes et remarques personnelles

Semaine 1 2 3 4 5 ⑥ 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Tests en production

Compte rendu explicatif

- Le prod screen permet maintenant de scanner toutes les 2minutes et afficher les informations tout en tenant compte du fait qu'un ou plusieurs soient déconnectés ou reconnectés.
- Ajout du scan sur adresses gprs si les karos ont été déconnectés ou mis en gprs. -
Le serveur tourne sur le Raspberry qui est installé dans le même réseau que les Karos.

Notes et remarques personnelles

Semaine 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Tests en production & adaptation aux besoins

Compte rendu explicatif

- Après avoir été en prod, l'équipe a proposé de modifier et ajouter les informations qui leur seront utiles lors des tests.
- Modifications & ajouts des données de la base de données :
 - o Logo
 - o Valeurs de calibration
 - o Ajout de la date du boot du nexxtender
 - o Connexion au serveur / time sync

Notes et remarques personnelles

Carnet de bord de Stage

A remplir en fin de semaine

Semaine 1 2 3 4 5 6 7 ⑧ 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Automatisation & finalisation

Compte rendu explicatif

- Configuration du Raspberry Pi pour le lancement des scripts au démarrage
- Affichage sur écran et vérification et tests des données affichées

Notes et remarques personnelles



Carnet de bord de Stage

A remplir en fin de semaine

Semaine 1 2 3 4 5 6 7 8 ⑨ 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Installation physique
- Documentation

Compte rendu explicatif

- Réalisation physique du projet sur écran pour la prod et dernières modifications
- Documentation du projet et de l'installation.

Notes et remarques personnelles



Carnet de bord de Stage

A remplir en fin de semaine

Semaine 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Projet Web App

Compte rendu explicatif

- Compréhension et découverte du code existant du projet nexxtender web app sur lequel je commencerai à travailler qui consiste en l'interface graphique de la page de configuration des bornes

Notes et remarques personnelles



Carnet de bord de Stage

A remplir en fin de semaine

Semaine 1 2 3 4 5 6 7 8 9 10 ⑪ 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Projet Web App

Compte rendu explicatif

- Compréhension et découverte du code existant partie serveur du projet nexxtender web app sur lequel je commencerai à travailler et qui est la page de configuration des bornes

Notes et remarques personnelles



Carnet de bord de Stage

A remplir en fin de semaine

Semaine 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Projet Web App

Compte rendu explicatif

- Compréhension et découverte du code existant partie serveur du projet nexxtender web app sur lequel je commencerai à travailler et qui est la page de configuration des bornes

Notes et remarques personnelles

Absence ces 2 derniers jours pour cause de maladie.



Carnet de bord de Stage

A remplir en fin de semaine

Semaine 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Projet de facturation

Compte rendu explicatif

- Compréhension du travail demandé, des composants du projet ainsi que du code.

Notes et remarques personnelles



Carnet de bord de Stage

A remplir en fin de semaine

Semaine 1 2 3 4 5 6 7 8 9 10 11 12 13 ⑭ 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Projet de facturation

Compte rendu explicatif

- Avancement du projet et amélioration du Less, Bootstrap..

Notes et remarques personnelles



Carnet de bord de Stage

A remplir en fin de semaine

Semaine 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (entourez le numéro qui correspond à la semaine)

Objectif(s)

- Projet facturation

Compte rendu explicatif

- Avancement du projet et amélioration du design avec Less, Bootstrap, Html

Notes et remarques personnelles



Bibliographie

- [1] CarPlug fait passer Powerdale à la vitesse supérieure, <http://corporate.engie-electrabel.be/fr/actualite/carplug-fait-passer-powerdale-a-la-vitesse-superieure/>
- [2] Dmitry Pashkevich, D. P. (2019, 6 janvier). 7 bad excuses for not using TypeScript. <https://blog.logrocket.com/7-bad-excuses-for-not-using-typescript-dbf5e603a9a8?gi=617b65dc6f33>
- [3] Iso, I. Simplifiez-vous la vie avec LESS. <https://openclassrooms.com/fr/courses/1281906-simplifiez-vous-la-vie-avec-less>
- [4] Isode Ltd., I., & A. Melnikov, A. M. The WebSocket Protocol. <https://tools.ietf.org/html/rfc6455>
- [5] Mark Otto, Jacob Thornton, and Bootstrap contributors, M. O., J. T. Bootstrap. <https://getbootstrap.com/>
- [6] N. Sakimura, N. S., & Microsoft. JSON Web Token (JWT). <https://tools.ietf.org/html/rfc7519>
- [7] Powerdale. <https://www.powerdale.info/>
- [8] Powerdale. <https://nl.linkedin.com/company/powerdale-scrl>
- [9] Tarh, T. Pourquoi utiliser TypeScript ? <https://tarh.developpez.com/articles/typescript/pourquoi-utiliser-typescript/>