

---

# CPE Lyon - 3IRC - 2023-2024

## Techniques et Langages du Web

### Sujet du projet “SmartEV”

---



## Contexte du projet

A partir de la fin des années 2000, sous la pression de l’augmentation du prix du pétrole, des préoccupations environnementales grandissantes, et grâce aux progrès techniques, les constructeurs automobiles ont commencé<sup>1</sup> à produire des véhicules électriques (*Tesla Roadster* en 2008, *Renault Zoe* en 2012...). En France, après le Grenelle de l’Environnement en 2007, le Gouvernement français a fait du développement des véhicules « décarbonés » (véhicules rechargeables 100% électriques (VE) ou hybrides rechargeables (VHR)) une priorité importante de sa politique environnementale et énergétique.

Dans les années 2010, on assiste ainsi à une croissance du nombre d’immatriculations de véhicules électriques; mais cette croissance est freinée par la “peur de la panne sèche” et le manque d’informations concernant les infrastructures de recharge publiques<sup>2</sup>. Depuis, diverses solutions ont été apportées, soit de la part des constructeurs (Tesla Trip Planner, application Renault My ZE) soit indépendantes et collaboratives ([chargemap.com](http://chargemap.com)). Le problème est qu’aucune de ces solutions n’est vraiment complète : les solutions constructeurs sont limitées aux véhicules et / ou bornes de recharge de la marque, et les référentiels de bornes de recharge se limitent au rôle d’annuaire et ne proposent pas d’outil plus complexe comme un calculateur d’itinéraire “intelligent”, indiquant les étapes de recharge nécessaires.

**L’objectif de ce projet est donc de mettre en application vos connaissances et compétences en technologies et langages du Web pour concevoir une solution qui répond à ce manque, en permettant à n’importe quel possesseur de véhicule électrique de localiser des stations de recharge compatibles, et d’obtenir des itinéraires “intelligents”**

💡 Rem. : dans les faits, par manque de temps, nous nous limiterons ici à deux véhicules : la *Tesla Model 3* et la *Renault Zoe avec moteur R75*.

## Cahier des charges / Spécifications techniques et fonctionnelles

La solution à développer doit comporter les éléments suivants :

1. une carte affichant l’emplacement des différentes infrastructures de charge, et permettant d’obtenir des informations sur celles-ci (adresse, types de prises, puissance...) (figures ??, ??) ;
2. une interface permettant de réaliser un calcul d’itinéraire “intelligent” adapté aux véhicules électriques (figure ??).

### Partie Carte des infrastructures de recharge

La carte doit permettre de visualiser l’emplacement des points de charge, et obtenir des informations sur ceux-ci (adresse, types de prises, puissance...). Pour le fond cartographique, vous utiliserez les données disponibles sur [openstreetmap.org](http://openstreetmap.org), que vous afficherez sur votre page web à l’aide de la bibliothèque JavaScript **Leaflet** ([leafletjs.com](http://leafletjs.com)). Lorsque l’utilisateur clique sur un point de charge sur la carte, il doit obtenir des informations sur celui-ci (notamment l’adresse, les types de connecteurs proposés ainsi que la puissance fournie). Les données sur les infrastructures de recharge sont disponibles en accès libre sur le site **OpenChargeMap** ([openchargemap.org](http://openchargemap.org)).

---

1. Ou plutôt *recommencé* : en 1900, près du tiers des véhicules circulant en France étaient électriques (source : [fr.wikipedia.org/wiki/Voiture\\_électrique\\_en\\_France](http://fr.wikipedia.org/wiki/Voiture_électrique_en_France))

2. Voir par exemple le rapport 13-561 du Sénat : [www.senat.fr/rap/13-561/13-561.html](http://www.senat.fr/rap/13-561/13-561.html)

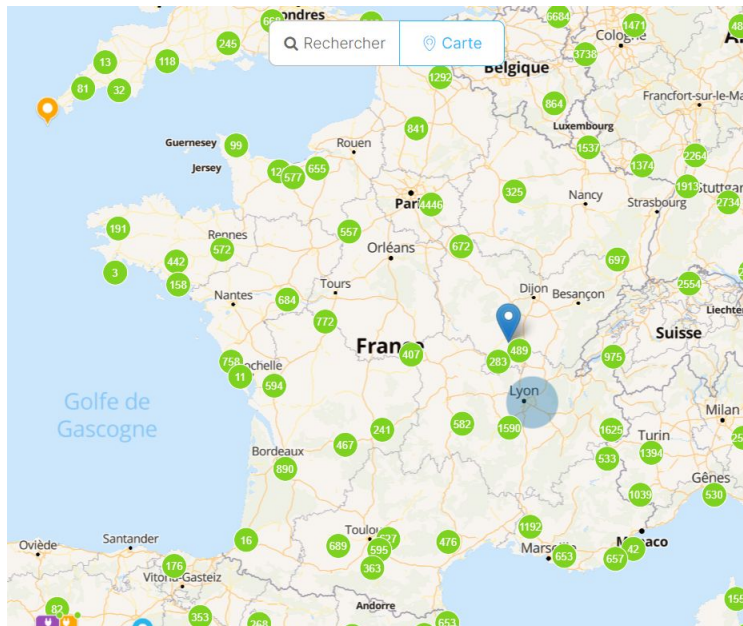


FIGURE 1 – Carte des infrastructures de recharge

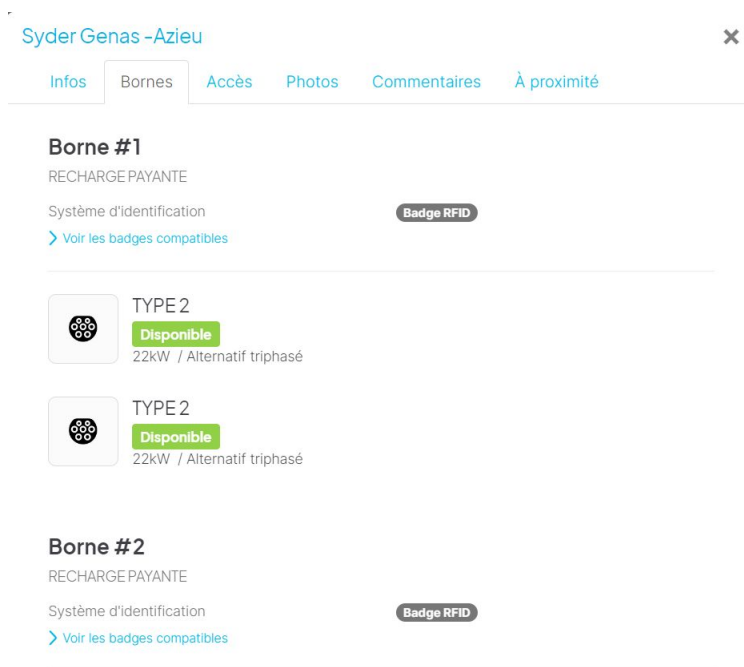


FIGURE 2 – Détails d’une infrastructure de recharge

- 💡 Vous pourrez vous limiter à une zone correspondant à la France.
- 💡 Les données fournies par OpenChargeMap sont assez nombreuses, et donc volumineuses. Il peut être judicieux de conserver et d'utiliser une copie *locale* de ces données sur votre site.

## Partie Calculateur d'itinéraire

Outre la partie “Carte” présentée dans la section précédente, votre solution doit comporter un calculateur d'itinéraires adapté aux véhicules électriques.

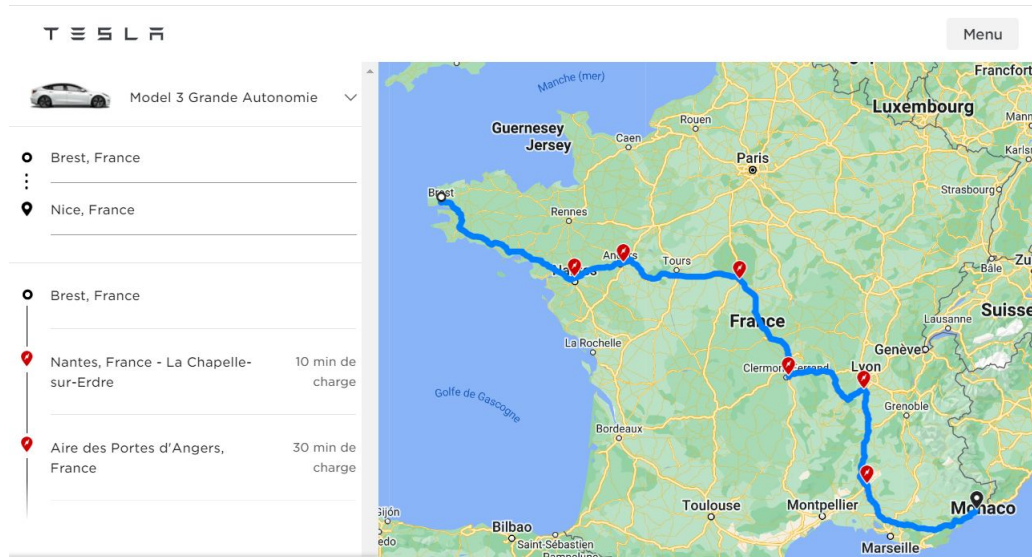


FIGURE 3 – Calculateur d'itinéraire Tesla

### A Géocodage

L'utilisateur doit disposer d'un moyen lui permettant de saisir une adresse de départ et une adresse d'arrivée, ou d'obtenir ces adresses par un clic sur la carte (à la manière de ce qu'on peut faire dans Google Maps). La conversion d'une adresse en coordonnées géographiques s'appelle le *géocodage* ou *geocoding* en anglais, l'opération inverse s'appelant... *reverse geocoding*. Il existe de nombreuses API de géocodage gratuites (ou limitées en nombre d'appels) sur Internet, comme Mapbox ([www.mapbox.com](http://www.mapbox.com)) (mais qui demande depuis 2021 un numéro de carte de crédit lors de l'inscription).

### B Calcul d'itinéraire “intelligent”

Bien qu'il soit techniquement possible de récupérer depuis OpenStreetMap le graphe du réseau routier, on ne vous demande ici pas d'implémenter un algorithme de calcul d'itinéraire (ou de plus court chemin dans un graphe). Vous utiliserez un service existant, par exemple celui de Mapbox ([docs.mapbox.com/api/navigation/](https://docs.mapbox.com/api/navigation/)), pour calculer un itinéraire entre deux points *A* et *B*. Pour permettre l'affichage de l'itinéraire sur une carte, les API fournissent généralement une *polyligne*, encodée au format *Polyline* ([developers.google.com/maps/documentation/utilities/polylinealgorithm](https://developers.google.com/maps/documentation/utilities/polylinealgorithm)) ou *GeoJSON*. L'utilisateur doit également pouvoir visualiser sa *feuille de route*, c'est à dire la liste de ses étapes, du point de départ au point d'arrivée.

⚠ Attention à bien utiliser le même référentiel cartographique pour l'affichage et le calcul d'itinéraire, faute de quoi un itinéraire pourrait sembler “décalé” par rapport aux routes.

Cependant, le calculateur d'itinéraire proposé sur votre site doit être plus “intelligent”, en ce sens qu'il doit permettre :

- d'estimer l'autonomie restante du véhicule en tout point du trajet (cf. partie C)
- de proposer si nécessaire des étapes de recharge (cf. partie D)

## C Calcul de l'autonomie

La consommation énergétique instantanée d'un véhicule électrique est régie par les lois de la mécanique, et peut être modélisée par des équations assez simples (cf. Annexe), mais qui font appel à un certain nombre de données parfois difficiles à trouver (par exemple les caractéristiques des véhicules, qui ne sont pas toujours rendues publiques par les constructeurs).

💡 Par conséquent, dans ce projet nous allons grandement simplifier l'estimation de l'autonomie d'un véhicule, en partant du principe qu'il s'agit simplement d'une constante en km propre à chaque modèle de véhicule (par exemple, vous direz qu'une *Tesla model 3* a une autonomie de 400 km, et qu'une *Renault Zoe* a une autonomie de 250 km, et que l'autonomie est proportionnelle au nombre de kilomètres parcourus. Ainsi, une Renault Zoe qui démarre un trajet avec une batterie chargée à 100% a une autonomie restante de 150 km si elle a déjà parcouru 100 km. Vous ne tiendrez donc compte ni de la vitesse, ni de l'accélération, ni des pentes.

## D Détermination des étapes de recharge

Une fois que vous savez estimer l'autonomie d'un véhicule, et connaissant la capacité et le niveau de charge de la batterie du véhicule, il est facile de savoir si une (ou plusieurs) recharge(s) est (sont) nécessaire(s).

💡 L'idée consiste à rechercher sur l'itinéraire un point à partir duquel il serait préférable de commencer à rechercher une borne de recharge car le niveau de la batterie est en-dessous d'un seuil (un peu de maths seront sans doute utiles!), puis de rechercher une borne compatible autour de ce point. La bibliothèque JavaScript *Turf* ([turfjs.org](https://turfjs.org)) offre de très nombreux outils d'analyse géospatiale et géométriques pour résoudre ce type de problèmes. Une fois qu'on a trouvé une borne en un point  $C$ , il faut donc modifier l'itinéraire  $A \rightarrow B$  en itinéraire  $A \rightarrow C \rightarrow B$ .

💡 On ne vous demande pas de déterminer l'itinéraire *optimal* (quel que soit le sens qu'on donne à ce mot ici : le moins de recharge possible, ou le plus petit détour, ou encore l'itinéraire qui conduit à un temps de charge minimal...). De même, dans un premier temps, concentrez-vous sur des itinéraires nécessitant *une* recharge, puis généralisez votre solution à plusieurs recharges.

### ⚠ Situation particulière : itinéraires divergents

Il est possible de construire des itinéraires divergents, c'est-à-dire des itinéraires qui ne conduiront jamais à destination :



Fig. 1 : le trajet de A à B est trop long et ne peut pas se faire en une seule fois  
=> on recherche une borne de recharge R1



Fig. 2 : Nouveau trajet, via R1  
=> le trajet de R1 au point B est encore trop long  
=> on doit chercher une deuxième borne R2

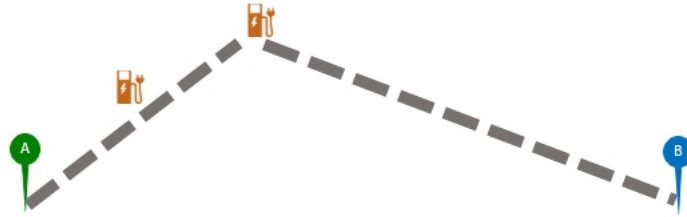


Fig. 3 : Nouveau trajet, en passant par R1 et R2  
 => le trajet de la 2<sup>ème</sup> borne au point B est encore trop long  
 => on doit chercher une troisième borne

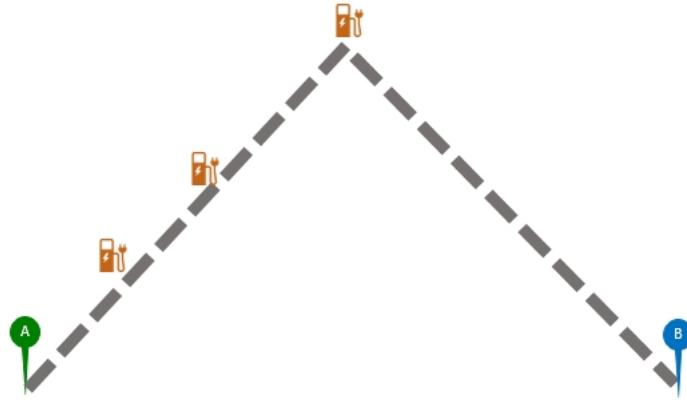


Fig. 4 : En continuant ainsi, on s'éloigne potentiellement  
 de plus en plus du point d'arrivée B

Pour éviter ces situations, il suffit de s'assurer, quand on est en un point  $P$  sur un itinéraire de  $A$  à  $B$  et qu'on considère une borne de recharge  $R$ , que l'équation suivante est vérifiée :

$$Aut_{100} - D_{R \rightarrow B} > Aut_{cour} - D_{P \rightarrow B}$$

où :

- $Aut_{100}$  désigne l'autonomie du véhicule chargé à 100 %,
- $Aut_{cour}$  son autonomie courante au point  $P$ ,
- $D_{x \rightarrow y}$  est la distance de  $x$  à  $y$ .

💡 Cette équation signifie simplement qu'il est avantageux de passer par la borne  $R$  pour atteindre la destination  $B$ .

#### E Calcul du temps de recharge :

La recharge d'une batterie n'est pas linéaire ; elle fonctionne plutôt un peu comme le remplissage d'un verre d'eau à ras bord : le remplissage commence d'abord à pleine puissance, puis la puissance de charge diminue au fur et à mesure qu'on se rapproche de la pleine charge. Dans notre cas, on modélise la courbe de charge en faisant une approximation par paliers : à 100% de la puissance de la borne pendant un temps  $T1$ , puis à 75% pendant un temps  $T2$ , 50% pendant un temps  $T3$  et enfin à 25% jusqu'à la charge pleine :

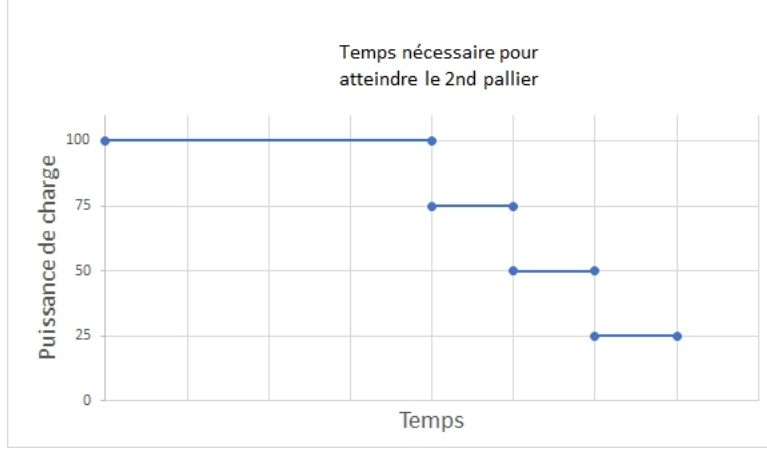


FIGURE 5 – Courbe de charge

Comme pour l'image du remplissage du verre d'eau, le temps sur les différents paliers dépend de la puissance avec laquelle on charge la batterie. Ainsi, on charge à pleine puissance jusqu'à atteindre un niveau de charge (ou *SOC*, de l'anglais *State of Charge*)  $SOC_{\text{palier}}$  (exprimé en %) qui dépend de la puissance de la borne. Cette correspondance est donnée dans le tableau suivant :

| Puissance de la borne (kW) | $SOC_{\text{palier}}$ |
|----------------------------|-----------------------|
| [40,...[                   | 75                    |
| [35,40[                    | 81                    |
| [30,35[                    | 83                    |
| [25,30[                    | 85                    |
| [20,25[                    | 88                    |
| [15,20[                    | 92                    |
| [10,15[                    | 95                    |
| [5,10[                     | 98                    |
| [0,5[                      | 98                    |

TABLE 1 –  $SOC_{\text{palier}}$  en fonction de la puissance de la borne

En notant  $Cap_{\text{Bat}}$  la capacité de la batterie du véhicule,  $SOC_{\text{initial}}$  son niveau de charge à son arrivée à la borne, et  $P_{\text{max}}$  la puissance de la borne, le temps sur chaque palier est donné par :

- 1<sup>er</sup> palier : puissance max jusqu'à atteindre  $SOC_{\text{palier}}$ , soit  $T1$  minutes, avec :
  - si  $SOC_{\text{initial}} < SOC_{\text{palier}}$   
 $\rightarrow T1 = 60 * ((SOC_{\text{palier}} - SOC_{\text{initial}}) / 100) * Cap_{\text{Bat}} / P_{\text{max}}$
  - si  $SOC_{\text{initial}} \geq SOC_{\text{palier}}$   
 $\rightarrow T1 = 0$
- 2<sup>ème</sup> palier : on charge à 75% de  $P_{\text{max}}$  pendant  $T2$  minutes :
  - si  $SOC_{\text{initial}} < SOC_{\text{palier}}$   
 $\rightarrow T2 = 60 * (((SOC_{\text{palier}} + 100) / 2) - SOC_{\text{palier}}) / 100 * Cap_{\text{Bat}} / (0,75 * P_{\text{max}})$
  - sinon, si  $SOC_{\text{initial}} < (SOC_{\text{palier}} + 100) / 2$   
 $\rightarrow T2 = 60 * (((SOC_{\text{palier}} + 100) / 2) - SOC_{\text{initial}}) / 100 * Cap_{\text{Bat}} / (0,75 * P_{\text{max}})$
  - sinon  
 $\rightarrow T2 = 0$
- 3<sup>ème</sup> palier : on charge à 50% de  $P_{\text{max}}$  pendant  $T3$  minutes :
  - si  $SOC_{\text{initial}} < (SOC_{\text{palier}} + 100) / 2$   
 $\rightarrow T3 = 60 * (((SOC_{\text{palier}} + 500) / 6) - ((SOC_{\text{palier}} + 100) / 2)) / 100 * Cap_{\text{Bat}} / (0,5 * P_{\text{max}})$
  - sinon, si  $SOC_{\text{initial}} < (SOC_{\text{palier}} + 500) / 6$   
 $\rightarrow T3 = 60 * (((SOC_{\text{palier}} + 500) / 6) - SOC_{\text{initial}}) / 100 * Cap_{\text{Bat}} / (0,5 * P_{\text{max}})$

- sinon  
→  $T3 = 0$
- 4<sup>ème</sup> palier : on charge à 25% de  $P_{max}$  pendant  $T4$  minutes :
  - si  $SOC_{initial} < (SOC_{palier} + 500)/6$   
→  $T4 = 60 * (100 - ((SOC_{palier} + 500)/6))/100 * Cap_{Bat}/(0,25 * P_{max})$
  - sinon, si  $SOC_{initial} < 100$   
→  $T4 = 60 * (100 - SOC_{palier})/100 * Cap_{Bat}/(0,25 * P_{max})$
  - sinon  
→  $T4 = 0$

💡 Dans les équations ci-dessus, le facteur 60 sert simplement à convertir des heures en minutes.

**Exemple :** un véhicule ayant une batterie d'une capacité  $Cap_{Bat} = 41$  kWh arrive à une borne de recharge d'une puissance  $P_{max} = 22$  kW avec un  $SOC_{initial} = 15\%$ .

Pour une borne d'une puissance 22 kW, le tableau 1 ci-dessus donne un  $SOC_{palier}$  de 88 % ;

- $SOC_{initial} < SOC_{palier}$  donc on recharge à 100%  $P_{max}$  jusqu'à atteindre un SOC de 88%, ce qui prend  $T1 = 60 * (88 - 15)/100 * 41/22 = 81$  min
- $SOC_{initial} < SOC_{palier}$  donc on recharge ensuite à 75%  $P_{max}$ , jusqu'à atteindre un SOC de 94% ( $= (88 + 100)/2$ ), ce qui prend  $T2 = 60 * ((88 + 100)/2 - 88)/100 * 41/(0,75 * 22) = 9$  min
- $SOC_{initial} < (SOC_{palier} + 100)/2$  donc on recharge ensuite à 50%  $P_{max}$ , jusqu'à atteindre un SOC de 98% ( $= (88 + 500)/6$ ), ce qui prend  $T3 = 60 * (((88 + 500)/6 - ((88 + 100)/2))/100) * 41/(0,5 * 22) = 9$  min
- $SOC_{initial} < (SOC_{palier} + 500)/6$  donc on recharge enfin à 25% pendant un temps  $T4 = 60 * (100 - ((88 + 500)/6))/100 * 41/(0,25 * 22) = 9$  min

## F Evolutions possibles :

- gérer le clustering des points de charge
- ajouter un widget permettant d'afficher soit tous les points de charge, soit uniquement les points de charge compatibles avec le véhicule sélectionné
- ajouter des points de passage (= arrêts) intermédiaires sur l'itinéraire
- gérer d'autres modèles de voitures
- prendre en compte les pentes de la route
- gérer les utilisateurs (connexion, profil, etc.)
- gérer les horaires d'accessibilité des points de charge
- gérer les conditions d'accès (certaines bornes sont réservées aux porteurs d'un badge ou aux clients d'un hôtel, et ne devraient pas être proposées dans les solutions)
- estimer les vitesses sur les tronçons (compliqué)
- prendre en compte les pentes des routes
- ...



## Organisation du projet

Vous travaillerez en **trinôme**. Le projet démarre le **6 février** et est à rendre pour le **dimanche 2 avril à 23h59** (date et heure de fermeture automatique du dépôt). **Aucun rendu ne sera possible après cette date**, et le code qui sera évalué sera celui présent sur le dépôt à ce moment-là.

Chaque trinôme dispose d'un dépôt sur Github. Vous devez commencer par créer une équipe *via* le lien disponible sur la page e-campus du cours; **pour des raisons pratiques, cette équipe doit impérativement être désignée par les deux noms de famille des membres du trinôme, sous la forme NOM1-NOM2-NOM3**. L'idée est donc de mettre l'accent sur le travail d'équipe, la collaboration, et le partage de code à l'aide de **Git** : chaque membre du trinôme travaille sur une fonctionnalité précise **en accord avec l'autre membre**, puis **pousse** son code sur le dépôt partagé pour la mise en commun (*fusion*).

Attention à bien gérer votre temps et votre collaboration, ce sont deux composantes fondamentales de la gestion et de la réussite d'un projet !

## Annexes

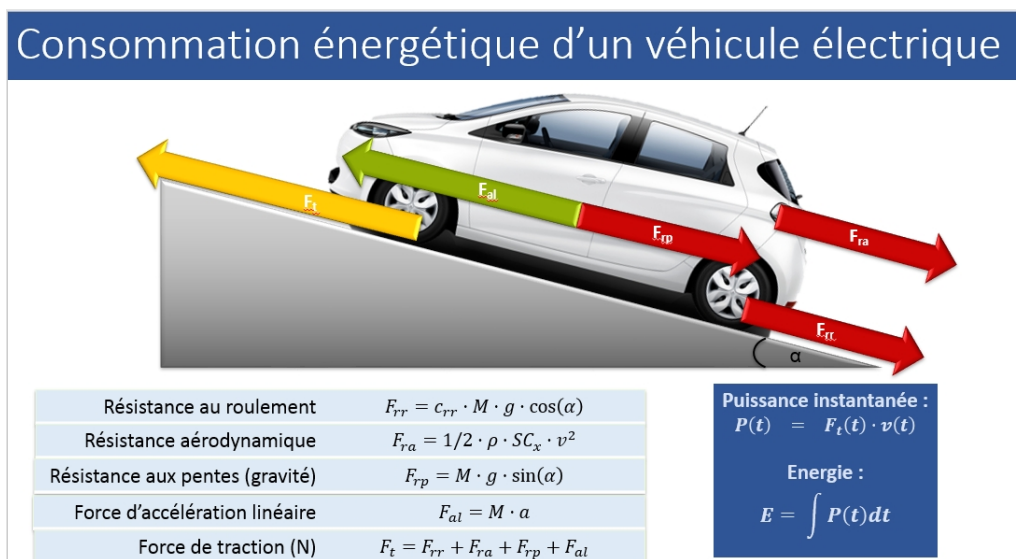
### Annexe 1 : données sur les véhicules

| Modèle          | Capacité Batterie | Autonomie | Puissance de charge max | Connecteurs            |
|-----------------|-------------------|-----------|-------------------------|------------------------|
| Renault Zoe R75 | 41 kWh            | 290 km    | 22 kW                   | Type 2 (id 25 et 1036) |
| Tesla Model 3   | 57 kWh            | 485 km    | 170 kW                  | Supercharger (id=27)   |

💡 Les *id* des connecteurs ci-dessus correspondent à ceux de l'API *OpenChargeMap.org*.

### Annexe 2 : modèle physique de consommation énergétique

En toute rigueur, un véhicule en déplacement est soumis à diverses forces qui peuvent être modélisées par les équations de la figure ci-dessous. Si l'on connaît la vitesse instantanée du véhicule, on peut alors très facilement calculer la puissance dégagée pour le déplacement. Une intégration par rapport au temps permet alors de déterminer l'énergie consommée durant la période considérée.



Dans cette figure, les valeurs  $M$ ,  $c_{rr}$  et  $SC_x$  sont des caractéristiques du véhicule (respectivement, la *masse*, le *coefficient de résistance au roulement* et le *coefficient d'aérodynamisme*), qui sont souvent jalousement gardées par les constructeurs. De plus, il est assez difficile de modéliser les vitesses et les accélérations instantanées d'un véhicule. C'est pourquoi ce modèle a été largement simplifié dans le projet.