



Présentation PFA

Application mobile pour rediriger les usagers lors des pannes de transport en commun sur le réseau de Bordeaux-Métropole

Équipe de développement

Victor	SAINT GUILHEM
Mehdi	BOUNAKHLA
Nathan	REAVAILLE
Romain	RAMBAUD
Mohammed	RIHANI
Pierre	ROUX

Client

Hassene BEN SALEM

Responsable Pédagogique

Toufik AHMED

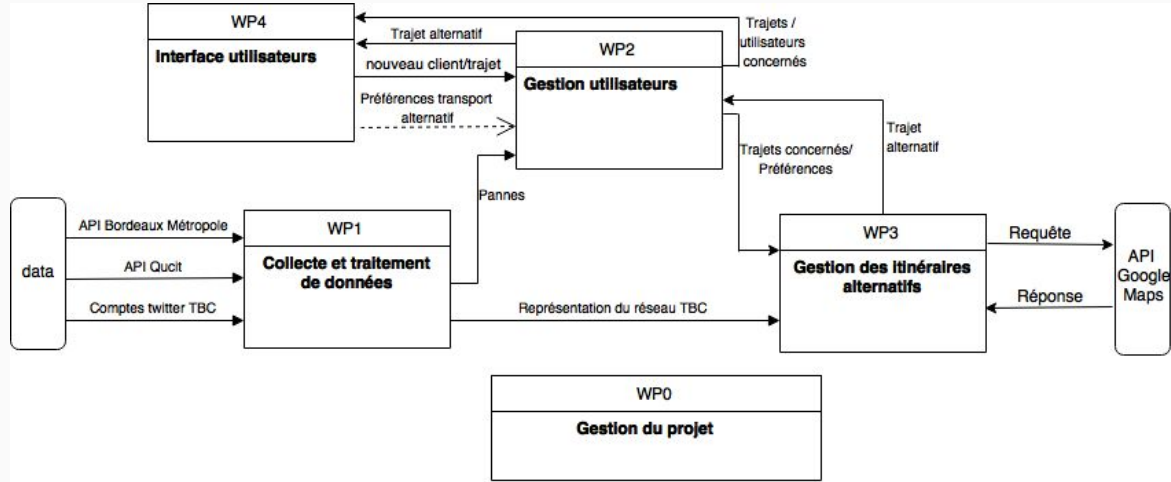
- Analyse du Besoin du Client
 - Démonstration de l'Application
 - Conception de l'Application : Polymer & Cordova
 - Serveur et Interface
 - Représentation du Réseau
 - Détection des Pannes
 - Bilan

Le projet de Qucit

- Créer une application dans le thème des transports en commun...
- ... qui collecte les données de *data.bordeaux-metropole*...
- ... tout en testant le framework Polymer de Google

Les Spécifications

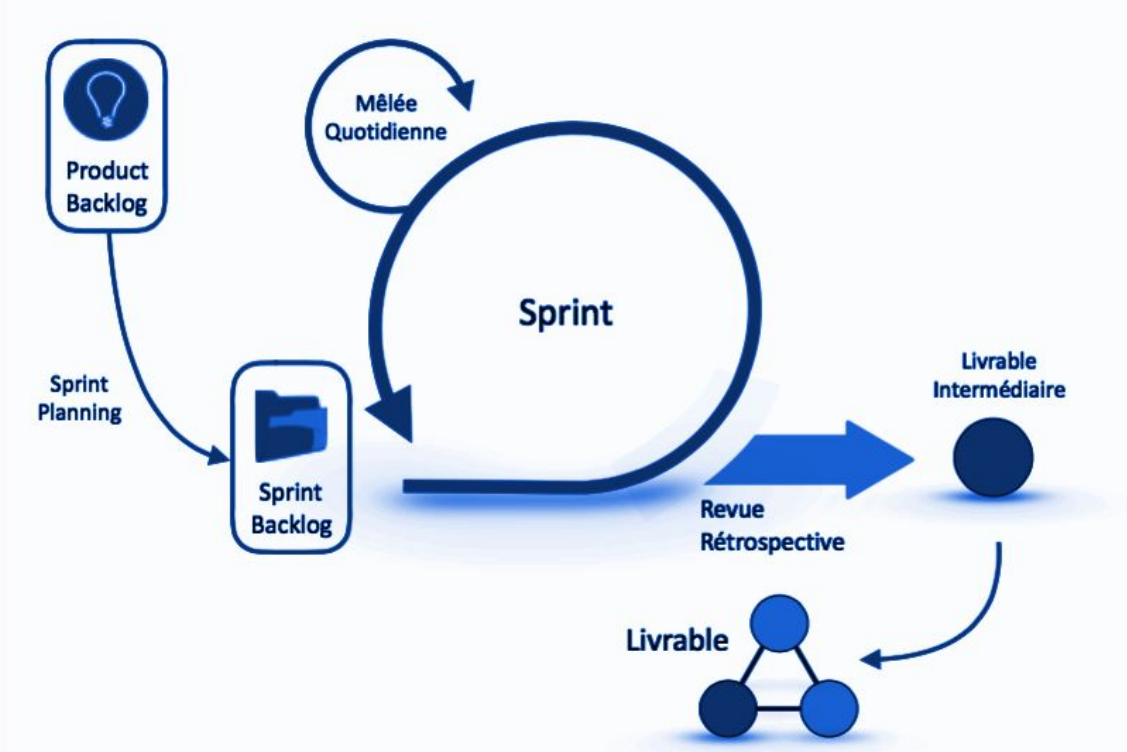
- Complètes pour les besoins pédagogiques
- Vouées à évoluer de part l'agilité du projet



Découpage du projet en Working Packages

L'adaptation de la méthode SCRUM

- Des rôles difficiles à définir
- Une durée de Sprint dilatée
- Des outils nouveaux



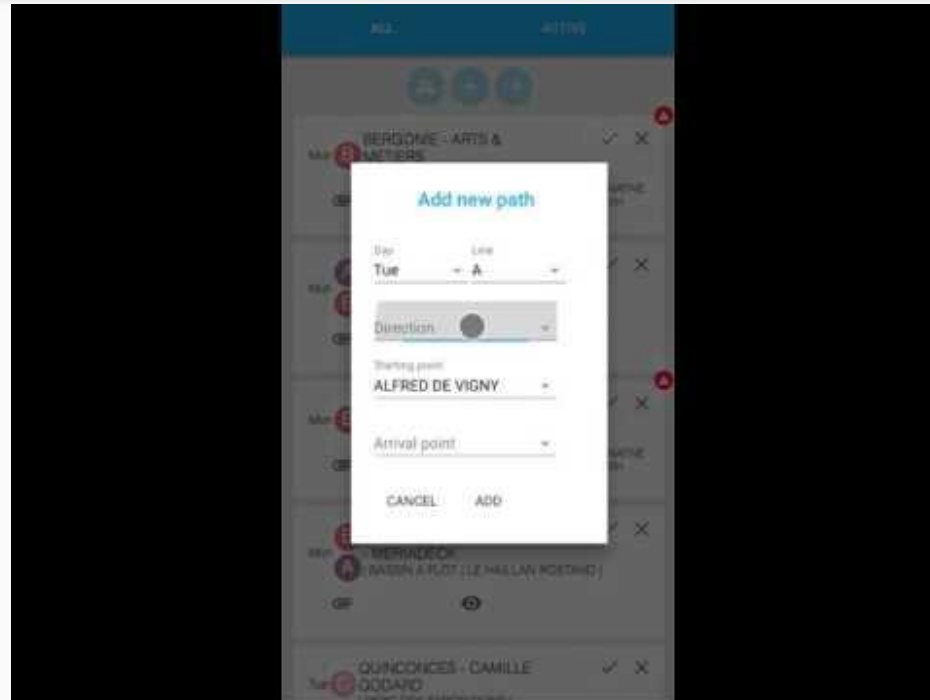
Les objectifs à atteindre ...

- Représenter le réseau de la Cub à partir des données collectées
- Faire émerger les pannes du réseau
- Générer des itinéraires alternatifs
- Tester le framework Polymer

- « L'API » de *data.bordeaux-metropole*
- L'analyse des tweets pour faire émerger les pannes
- Générer des itinéraires alternatifs sur une représentation partielle
- La prise en main de Cordova

- Analyse du Besoin du Client
- **Démonstration de l'Application**
- Conception de l'Application : Polymer & Cordova
- Serveur et Interface
- Représentation du Réseau
- Détection des Pannes
- Bilan

Présentation de l'application



- Analyse du Besoin du Client
- Démonstration de l'Application
- Conception de l'Application : Polymer & Cordova
- Serveur et Interface
- Représentation du Réseau
- Détection des Pannes
- Bilan

Code Client : Besoins

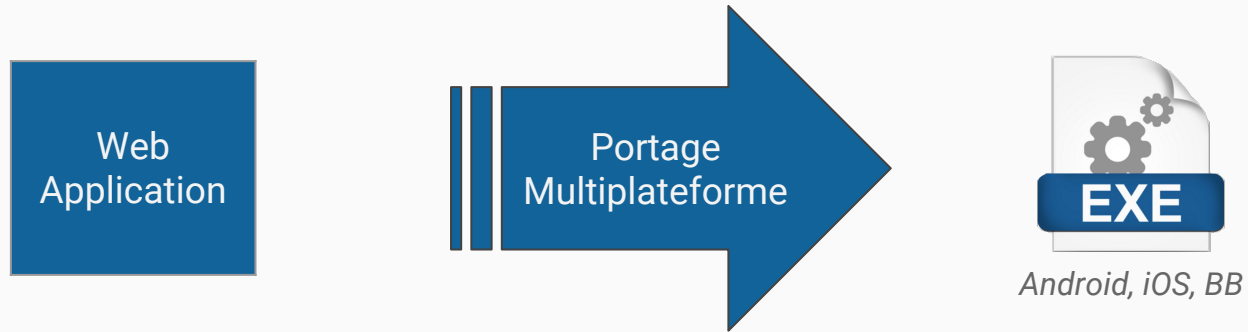
- Fournir une application **mobile** pour *Android* et *iOS*
- Se baser sur des **technologies web** dans la conception de l'application
- Utiliser le nouveau **framework web** de Google basé sur les **Web Components** : *Polymer*

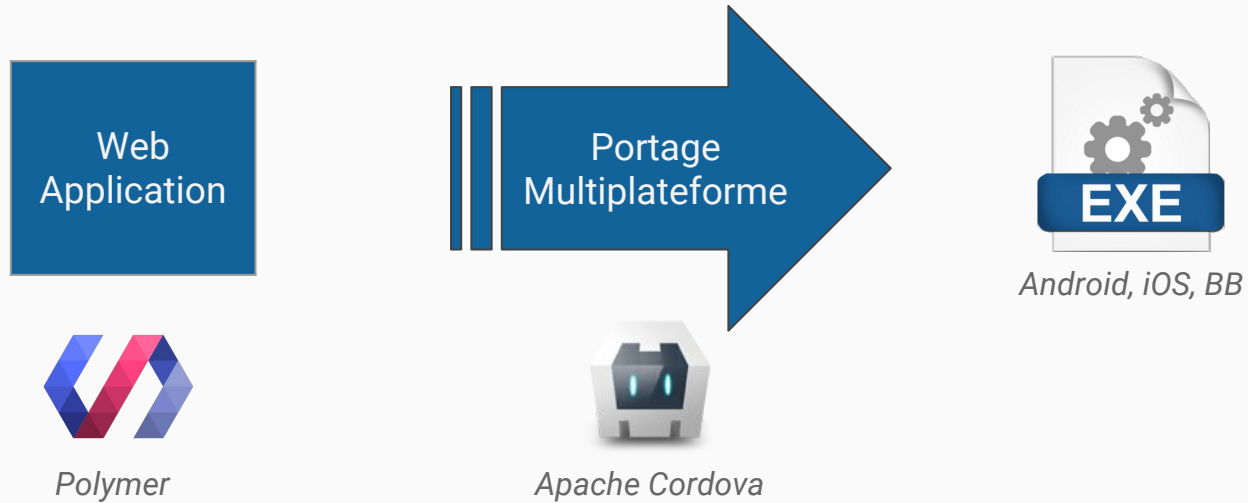
C'est quoi cette application ?

L'objectif du code client est de fournir une application mobile pour *Android* et *iOS* en se basant sur des technologies web.

C'est le principe même d'une **application hybride**, et donc qui est le produit conçu dans ce projet avec la technologie web souhaitée : **Polymer**.

C'est une application « Hybride »







Polymer ?

Nouveau *framework web (open source)* de Google (encapsule *HTML*, *CSS* et *JavaScript*), basé sur les **Web Components** (composantes web) pour la construction d'*éléments* et d'applications.

Nouvelles spécifications du W3C (World Wide Web).
Permettent d'étendre le vocabulaire *HTML*.



- Création d'éléments personnalisés (*Custom elements*) à partir d'éléments existants.
- Encapsulation de ces éléments (personnalisés et/ou les éléments fournis par le framework utilisé et/ou par le web) dans une web application.

(Développement plus rapide, profit de l'expansion des *Web Components* et des technologies autour...)

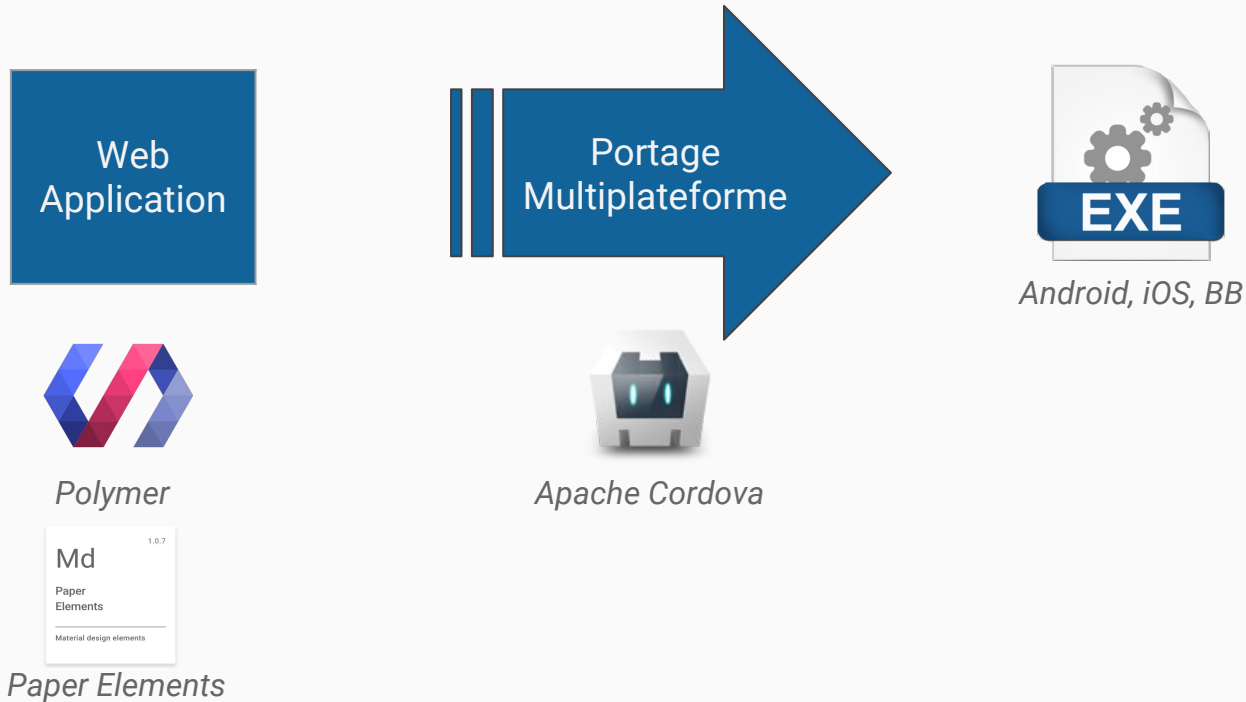

```
<link rel="import" href="my-element.html">
```

```
<my-element></my-element>
```

Paper Elements

Catégorie d'éléments mis à disposition dans le *framework Polymer*.
Elle implémente des éléments du nouveau langage visuel d'Android :
Material Design.

Donc plus précisément avec les techno





Apache Cordova ?

Framework de développement mobile (open source) pour le portage multiplateforme. Il permet de générer une application native contenant une **Web View** et permet de spécifier la page *HTML* à afficher.

(Autrement dit : génère une application native se comportant comme un **navigateur chromeless** (i.e *navigateur personnalisable*) en plein écran contenant la **Web View** souhaitée.)

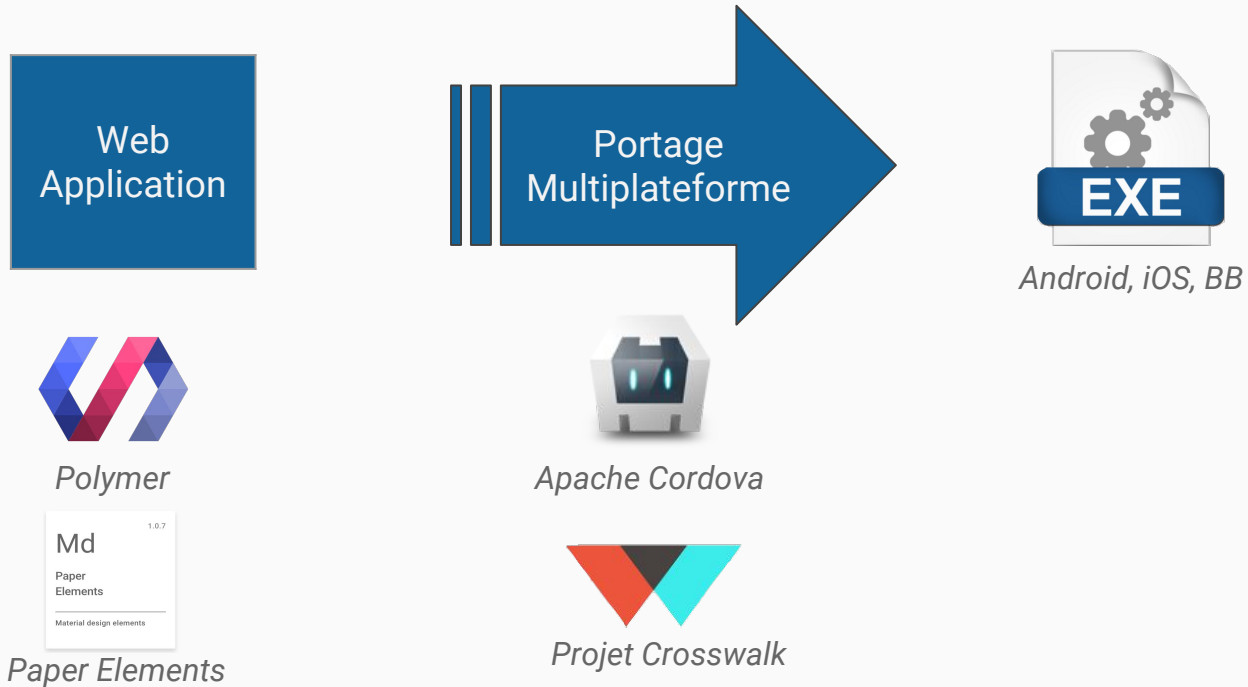
Problème



Permet d'intégrer la *Web View* de *chrome* (*chromium Web View*) au mobile et l'utilise au lieu de la *Web View* par défaut, si celle-ci n'est pas supportée.



Exécutable pour les versions d'Android qui précèdent la 4.4



L'élément personnalisé : post-card



```
{
  "pid": 5,
  "subPaths": [{"line": "C",
                  "direction": "PARC DES EXPOSITIONS",
                  "startStop": "QUINCONCES",
                  "finishStop": "CAMILLE GODARD"}],
  "day": "Tue",
  "active": false,
  "breakdown": false,
  "connection": false,
  "beginHour": "13h30",
  "endHour": "14h",
  "deleted": false
}
```

Représentation du réseau de la TBC au niveau du client

Add new path

Day Line
Mon B

Direction
BASSINS A FLOT

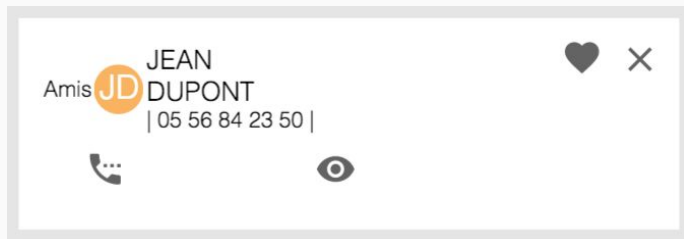
Starting point
ARTS & METIERS

Arrival point
BERGONIE

CANCEL ADD

```
{  
  "id": 3,  
  "lid": "1",  
  "line": "1",  
  "route_color": "7a78b4",  
  "route_text_color": "FFFFFF",  
  "stops": [],  
  "directions": [],  
}
```


Dans un autre contexte (fiche de contacts à l'iOS)



```
{
  "pid": 14,
  "subPaths": [{"line": "JD",
                  "direction": "05 56 84 23 50",
                  "startStop": "JEAN",
                  "finishStop": "DUPONT"}],
  "day": "Amis",
  "active": false,
  "breakdown": false,
  "connection": false,
  "beginHour": "13h30",
  "endHour": "14h",
  "deleted": false
}
```

Ce qu'on fournit au niveau du code client finalement

- Le code client de l'application.
- Un *framework d'éléments personnalisés* (**Custom elements**) créés pour l'application, qui sont réutilisables
 - dans un même contexte (de trajets en transport en commun)
 - dans un autre contexte avec quelques modifications (fiche de contacts à l'iOS, alarme, pense bête...).

- Analyse du Besoin du Client
- Démonstration de l'Application
- Conception de l'Application : Polymer & Cordova

• Serveur et Interface

- Représentation du Réseau
- Détection des Pannes
- Bilan



Serveur : Interface

GET /paths/{pathId}

path

Summary

Find path by ID

Description

Returns a path with his ID

Parameters

Name	Located in	Description	Required	Schema
pathId	path	ID of path	Yes	⇒ integer (int64)

Responses

Code	Description	Schema
200	successful operation	⇒ ▼Path { id: integer * active: boolean * day: integer * beginHour: integer * endHour: integer * subpaths: ▶[] }
400	Invalid ID supplied	
404	Path not found	

Try this operation

DELETE /paths/{pathId}

path

Summary

Delete path by ID

Parameters

Name	Located in	Description	Required	Schema
pathId	path	ID of the path that needs to be deleted	Yes	⇒ integer (int64)

Responses

Code	Description
400	Invalid ID supplied
404	Path not found

Try this operation

- Analyse du Besoin du Client
- Démonstration de l'Application
- Conception de l'Application : Polymer & Cordova
- Serveur et Interface
- Représentation du Réseau
- Détection des Pannes
- Bilan

Données GTFS (General Transit Feed Specification)

- Présentes sur data.bordeaux-metropole.fr
- Format standardisé.
- Fichiers format `csv`.
- Pandas (bibliothèque Python)

Format csv \Rightarrow Tableau Dataframe

```
[MBP-de-Mehdi:gtfs mbounakhla$ head -n 6 stops.txt
```

```
stop_id,stop_name,stop_lat,stop_lon,stop_desc,zone_id,stop_url,stop_code,location_type,parent_station
3684,Lauriers,44.879107,-0.517867,,33249,,LAURI,0,LAURI
3685,Bois Fleuri,44.875987,-0.519344,,33249,,BFLEA,0,BFLEU
3686,Bois Fleuri,44.876052,-0.519272,,33249,,BFLER,0,BFLEU
3687,Gravières,44.872741,-0.520231,,33249,,GRAVA,0,GRAVI
3688,Gravières,44.872806,-0.520171,,33249,,GRAVR,0,GRAVI
```

```
In [4]: df_stops[:5]
```

```
Out[4]:
```

	stop_id	stop_name	stop_lat	stop_lon	stop_desc	zone_id	stop_url	stop_code	location_type	parent_station
0	3684	Lauriers	44.879107	-0.517867	NaN	33249	NaN	LAURI	0	LAURI
1	3685	Bois Fleuri	44.875987	-0.519344	NaN	33249	NaN	BFLEA	0	BFLEU
2	3686	Bois Fleuri	44.876052	-0.519272	NaN	33249	NaN	BFLER	0	BFLEU
3	3687	Gravières	44.872741	-0.520231	NaN	33249	NaN	GRAVA	0	GRAVI
4	3688	Gravières	44.872806	-0.520171	NaN	33249	NaN	GRAVR	0	GRAVI

Tableaux Dataframe \Rightarrow Classes Stop et Trip

- Deux dictionnaires Python.

```
class Stop:
    def __init__(self):
        self.iden = "UNDEFINED"
        self.name = "UNDEFINED"
        self.lon = 1000.0
        self.lat = 1000.0
        self.trans_type = -1
        self.trip_ids = list()
```

```
class Trip:
    def __init__(self):
        self.iden = "UNDEFINED"
        self.route_id = -2
        self.direction_id = -1
        self.trip_headsign = "UNDEFINED"
        self.route_name = "UNDEFINED"
        self.path = list()
        self.tran = -1
```

Dictionnaires de Stops et de Trips \Rightarrow Graphe intermédiaire

- Network : bibliothèque Python (création / utilisation de graphes)
- Graphe = Union de graphes (graphe/liste).

Graphe intermédiaire \Rightarrow Graphe final

- Ajout d'arêtes entre deux noms de même nom.
- Est- ce suffisant pour prétendre à une représentation complète?
- Arrêts de bus Victoire.
- Ajout d'arêtes pour deux Stops de positions géographiques proches.
- Formule Haversine
- Choix de la distance maximale

Etapes de la construction du graphe :

- Format csv
- Tableaux Dataframe
- Dictionnaires Python des classes Stop et Trip
- Graphe intermédiaire sans correspondances
- Graphe final avec correspondances

Impact d'une panne sur le graphe :

- Définir ce qu'est une panne (liste d'arrêts, une ligne + 2 arrêts...)
- Traitement sur une copie du graphe initial (complexité trop importante autrement).
- Suppression des arêtes concernées.
- Plusieurs pannes : construction récursive du graphe.

Chemin alternatif

- `shortest_path(G, Source, Target)`
 - `G` : Le graphe en question
 - `Source` : Le point de départ
 - `Target` : Le point d'arrivée

- Analyse du Besoin du Client
- Démonstration de l'Application
- Conception de l'Application : Polymer & Cordova
- Serveur et Interface
- Représentation du Réseau
- Détection des Pannes
- Bilan

Détection des Pannes

- **Collecte** des données sur twitter
 - **6 comptes twitter** de la TBC : ligne tram A, B et C et lianes bus 1, 3 et 8
 - Au total environ **10 000 tweets**
- **Analyse**
 - Sont-ils formatés ?
 - Quelles informations contiennent-ils ?
 - Qu'est-il possible d'en extraire ?
- **Traitement**
 - Comment détecter une panne ?
 - Comment extraire les informations ?

Analyse :

- Des éléments inutile aux traitements :
 - Hashtag et liens par exemple
- Parfois des fautes :
 - Orthographes / frappes
 - Abréviations (Plusieurs orthographes d'un même mots...)



tbc Tram C
@tbc_TrnC

 Suivre

Malaise voyageur station "Saint Michel",
ralentissement direction Lycée V.Havel. Merci
de votre compréhension #tbc #InfoTrafic



tbc Tram C
@tbc_TrnC

 Suivre

Incident de circulation : trafic fortement ralenti
sur secteur Grand Parc<->Parc des Expo en
direction du L.V.Havel. #tbc #InfoTrafic



tbc Tram C
@tbc_TrnC

 Suivre

Ligne Tram C interrompue entre Terres Neuves
et Lycée Václav Havel. [infotbc.com/alertes](https://infotbc.com/alertes/ligne-)
/ligne- ... #tbc #InfoTrafic

Classification : l'Algorithme

- Utilisation d'algorithmes d'apprentissage machine
 - "Machine Learning"
- Algorithmes capable d'adapter leur réponses en sortie en fonction de données d'apprentissages
 - Processus de classification en 2 phases :
 - Apprentissage (sur des exemples)
 - Classification
- Algorithme de classification Naïve Bayésienne

Analyse :

- Les tweets peuvent annoncer plusieurs évènements :
 - Panne et Reprise
 - Reprise et Ralentissement



tbc Tram B

@tbc_TrAmB


 Suivre

Reprise du trafic sur la ligne. Retards à prévoir sur #tbc #InfoTrafic



tbc Tram A

@tbc_TrAmA

 Suivre

Reprise du trafic entre Buttinière et La Morlette. Ligne A interrompue entre La Morlette et Dravemont. [infotbc.com/alertes/ligne- ...](https://infotbc.com/alertes/ligne-...) #tbc #tbcTramA

Analyse :

- Dates dans les tweets
- Certains tweets annoncent des perturbations à l'avance
- Les éléments temporels ont des formats très varié



tbc Tram C

@tbc_TrainC

 Suivre

Préavis de grève le "Jeudi 31 Mars" :
perturbations à prévoir sur le réseau
infotbc.com/alertes/preavi ... #tbc #InfoTrafic



tbc Tram C

@tbc_TrainC

 Suivre

Manifestation secteur centre-ville le 12/03 à
14h : Déviations bus et interruptions tram à
prévoir infotbc.com/alertes/manife ... #tbc

Traitements :

1. Pré-traitements
2. Classification
3. Extractions



- Pré-traitements :
 - Nettoyage des tweets
 - Découpage en phrases
 - Corrections
- Classification
 - Donner une étiquette à chaque tweet
 - Panne ? Reprise ?
- Extractions
 - Noms des arrêts
 - Éléments temporels

Classification : Le modèle

- 4 Classes :
 - **Panne** : Interruption du trafic sur une ligne (ou une partie) ;
 - **Ralentissement** : Diminution des fréquences de passages sur une ligne (ou une partie) ;
 - **Reprise** : Reprise du trafic après un incident ;
 - **Non Pertinent** : Tweet qui ne concerne pas le trafic.
- On suppose que les classes sont équiréparties

Données d'apprentissages et de tests:



Distribution des tweets de D'apprentissages



Distribution des tweets de tests

Résultats :

- De bon résultats pour la classification 7,6 % d'erreurs
- La moitié des erreurs entre Panne et Ralentissement



Résultats Classification



Matrice confusion

Extraction d'information :

- Extraction des noms d'arrêts
 - A partir de dictionnaires (taxonomie)
 - Naïves Bayes
- Extraction des éléments temporels
 - Expressions régulières

- Analyse du Besoin du Client
- Démonstration de l'Application
- Conception de l'Application : Polymer & Cordova
- Serveur et Interface
- Représentation du Réseau
- Détection des Pannes
- Bilan

Bilan : Réalisation des objectifs

- Détection des pannes
- Serveur : Intégration de la détection
- Application mobile : Présentation d'itinéraire alternatif
- Intégration Client-Server
- Notifications des pannes vers application mobile

Bilan : Retours

Polymer :

- Projet innovant et d'avant-garde
- Pas forcément pour application hybride

Détection de pannes :

- Collecte facile
- Prédiction sans faute difficile

Bilan : Conclusion

- Un projet complexe mais intéressant
- Découverte de nouveaux domaines
- Initiation aux Data-Sciences