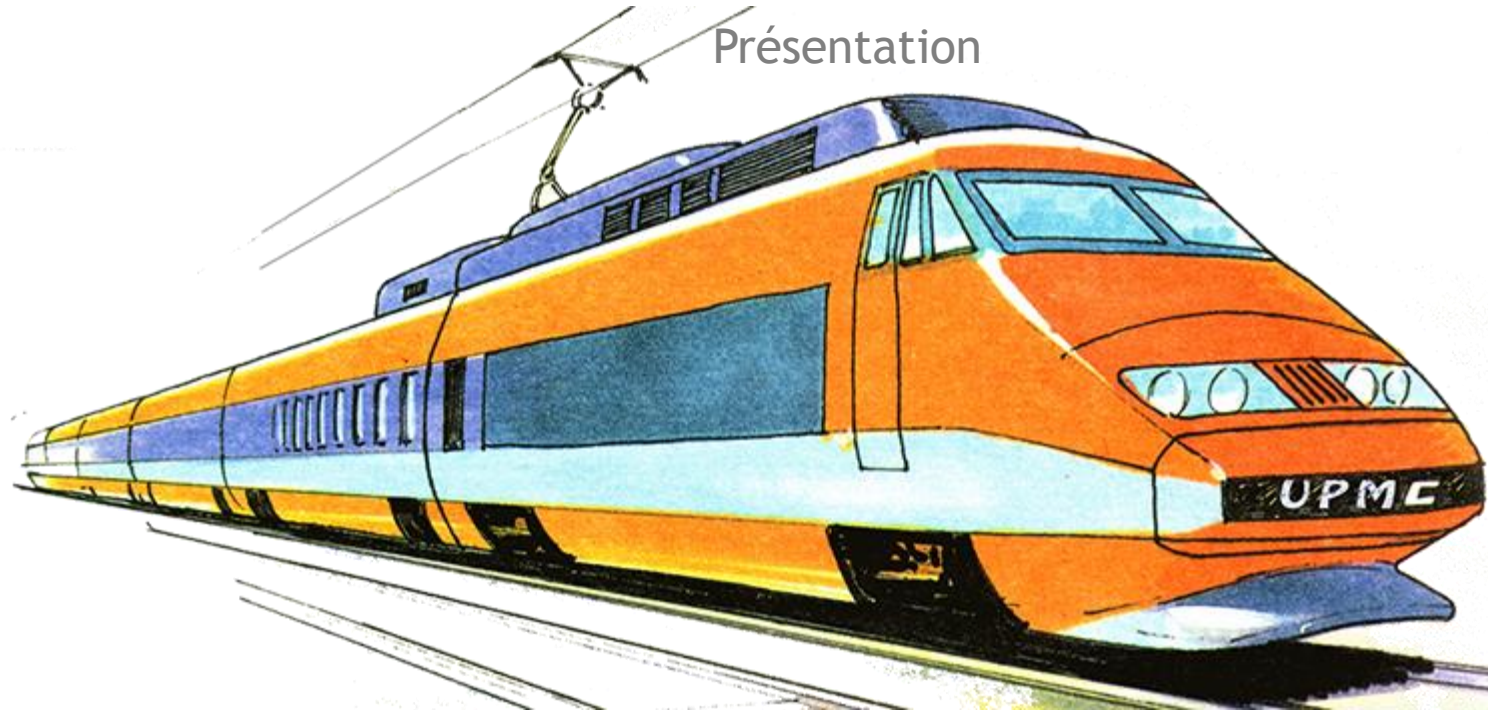


Transilien-UPMC

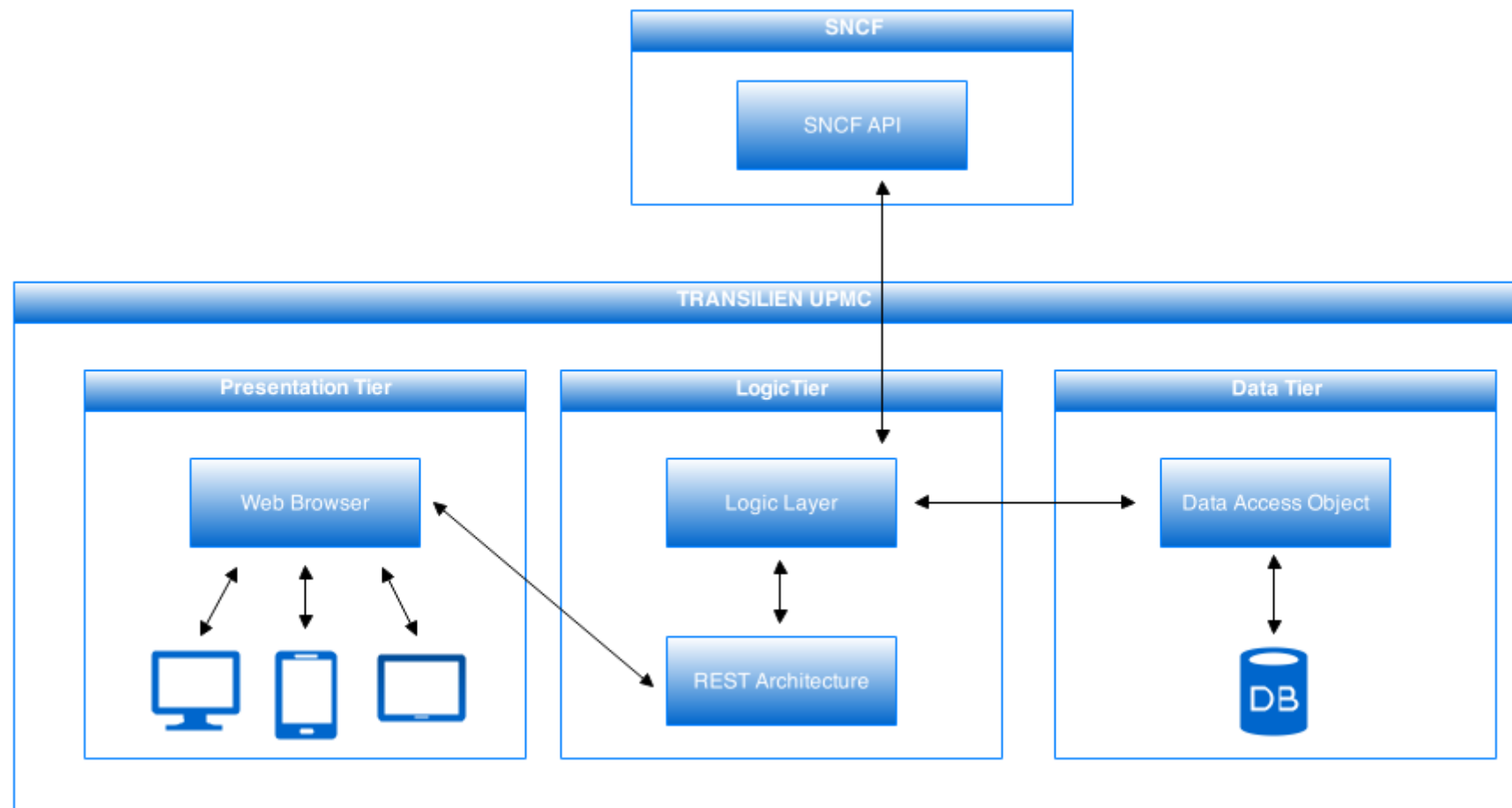
Présentation



Plan

- ▶ Introduction
- ▶ Diagramme d'Architecture
- ▶ Initialisation de l'application
- ▶ Interface graphique
- ▶ Prochain Départ
- ▶ Itinéraire
- ▶ Statistique
- ▶ Conclusion

Diagramme d'architecture



Initialisation

- ▶ Chargement de fichier statique
- ▶ Parsing
- ▶ Filtrage
- ▶ Orientation des lignes
 - ▶ On ajoute toutes les gares à une liste
 - ▶ Tant que la liste n'est pas vide
 - ▶ On calcul la distance entre les gares
 - ▶ Tant qu'il y a des gares pour lesquelles nous avons une distance
 - ▶ Si la gare a aucun voisin, on ajoute la distance la plus courte
 - ▶ Si la gare a déjà un voisin, on recherche la gare la plus proche qui ne dépasse un éloignement de 20 km et qui admet un angle avec la 1^{ère} gare voisine supérieur à 45°
 - ▶ On calcul via les voisins l'ordre d'apparition des gares dans la ligne

Interface Graphique

- ▶ Sobre
- ▶ Élégante
- ▶ Immersive
- ▶ Tracé approximatif des lignes

Prochain Départ

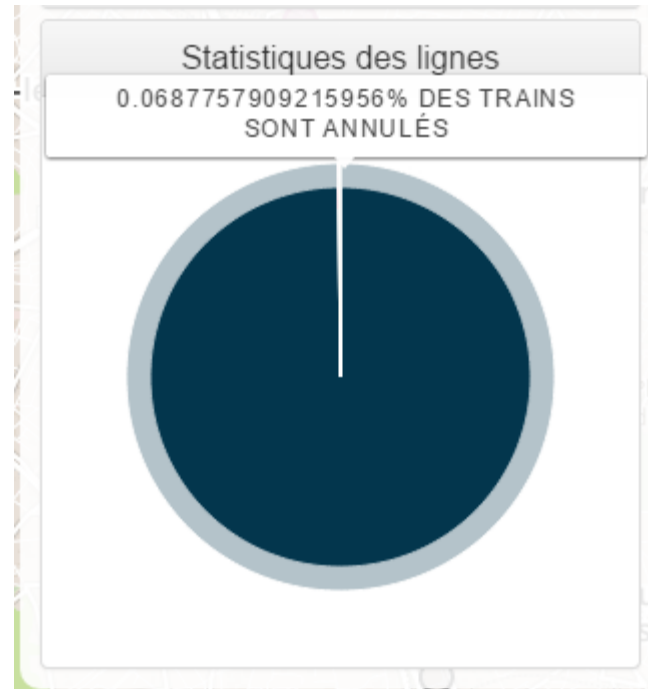
- ▶ Auto-complétion
- ▶ Rendu graphique travaillé

Itinéraire (~Dijkstra)

- ▶ Un direct existe ?
 - ▶ On renvoie le direct
- ▶ On ajoute la gare de départ à une file de priorité
- ▶ Tant que la file n'est pas vide
 - ▶ Pour chaque ligne de la gare
 - ▶ Pour chaque gare de la ligne
 - ▶ On calcul la distance vers la gare en cours de traitement
- ▶ Une fois la gare d'arrivée traitée et son score inférieur à celui des gares en attente ou bien une fois la file vide
 - ▶ On part de la destination et itérons vers le départ via les prédécesseurs

Statistique

- ▶ Peu de train dans la base
- ▶ Graphique indiquant le taux de service



Conclusion

- ▶ Conception et développement à la GAE
- ▶ Gestion du projet en Scrum
- ▶ Découverte de nouvelles technologies