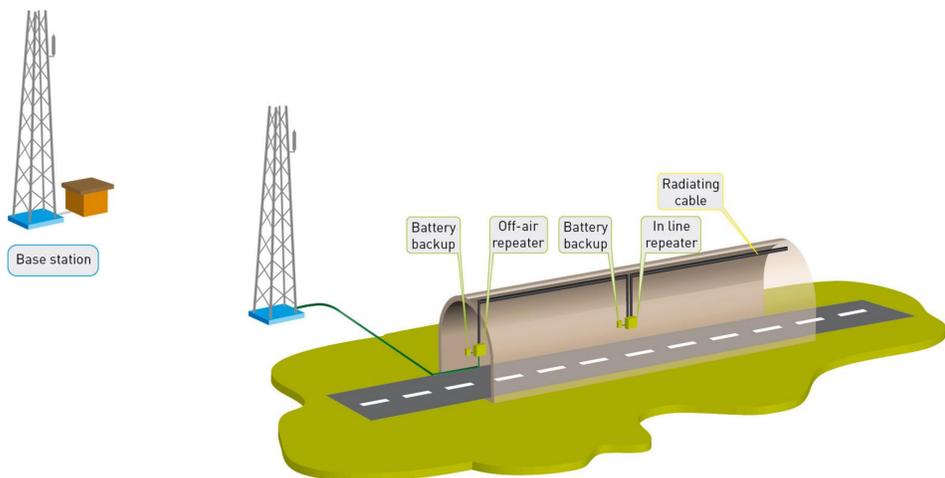


1) Contexte

Les ondes radio ne passent pas naturellement dans les **milieux confinés**, comme les tunnels, les métros ou les bâtiments. La société partenaire du projet, SEE Telecom, développe des répéteurs pour pouvoir créer une **continuité de service radio** au sein de ces milieux afin d'assurer la communication de services de secours tels que les pompiers, ambulances, police ou encore des services commerciaux tels que le GSM, la FM ou l'AM. Ces services ont une importance capitale pour la **sécurité** lors d'un incident au sein du milieu confiné.



Incendie du tunnel du Saint Gothard



Exemple d'architecture matérielle en milieu confiné,
Copyright SEE Telecom

L'entreprise partenaire dispose déjà à ce jour de répéteurs pour de nombreux services radio. Toutefois, ces répéteurs sont basés sur un **matériel figé**. Par conséquent, si l'on souhaite modifier la configuration pour assurer la transmission d'un autre service, il est nécessaire de se rendre sur place pour opérer les changements, ce qui n'est pas aisé.

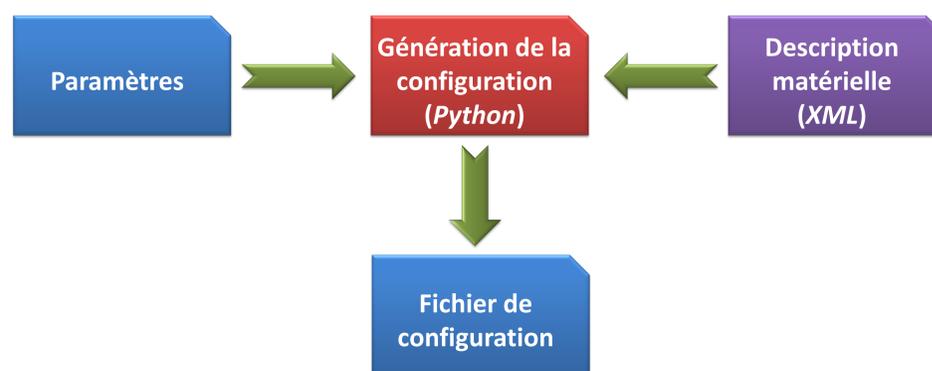
2) Objectifs

Les objectifs relatifs à ce projet sont multiples :

- **adapter** la configuration du répéteur en fonction du matériel;
- rendre **reconfigurable** le répéteur sans devoir agir physiquement sur celui-ci;
- permettre une **gestion intelligente** de la reconfiguration automatique des modules en cas de panne.

3) Génération des configurations

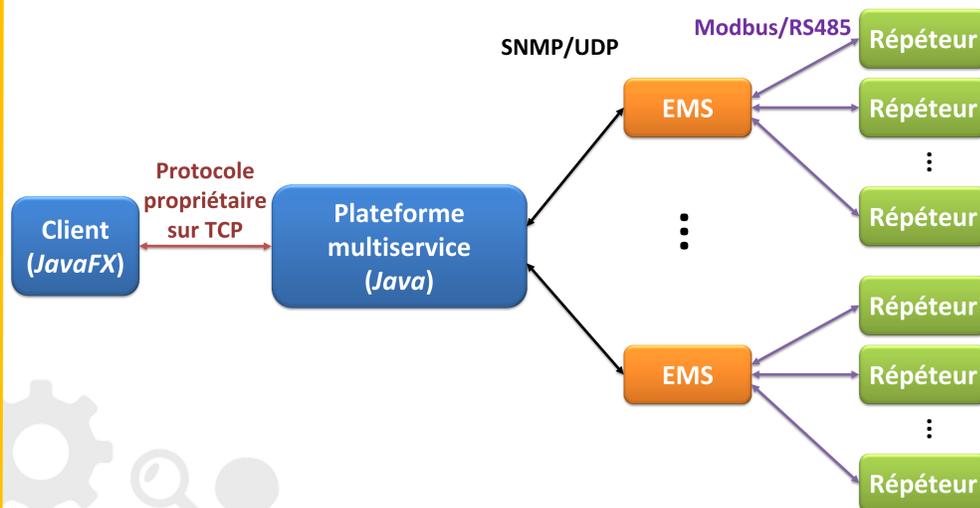
Il s'agit ici de **générer les configurations** dites « statiques » d'un répéteur, par exemple, les coefficients d'un **filtre** donné. Chaque répéteur peut contenir plusieurs configurations statiques et ainsi, l'utilisateur peut passer d'une configuration à une autre. La génération de la configuration statique se fait à l'aide d'un script Python. Les caractéristiques du matériel sont reprises dans un fichier XML. Ainsi, il est facile de générer de nouvelles configurations si le matériel change. À cela peuvent être ajoutés d'autres paramètres spécifiques au service répété.



Architecture de la génération de configuration

4) Plateforme

La **plateforme logicielle** n'en est qu'à ses débuts mais elle aura pour objectif de « commander » la reconfiguration des cartes en cas de nécessité. Cette plateforme sera codée en Java. Pour permettre une meilleure gestion graphique de l'interface homme machine, la partie client sera codée en JavaFX. La plateforme supervise les cartes EMS (Element Management System) qui elles-mêmes sont reliées aux divers répéteurs.



Architecture de la plateforme