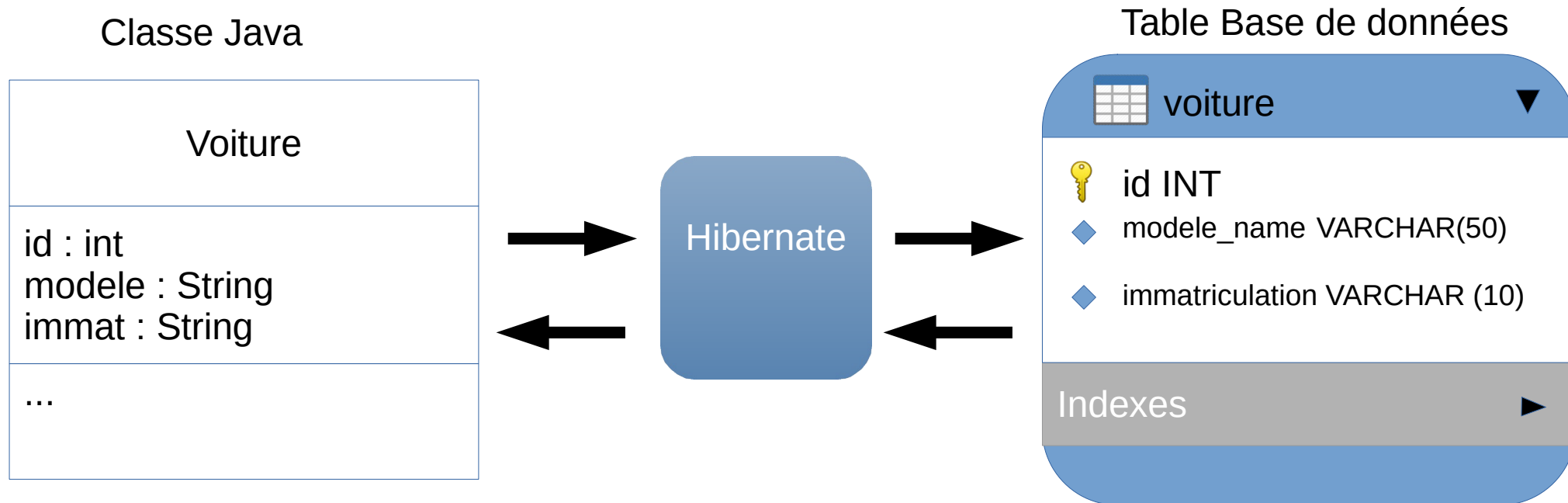




Hibernate – les relations entre objets

Mapping d'une classe isolée



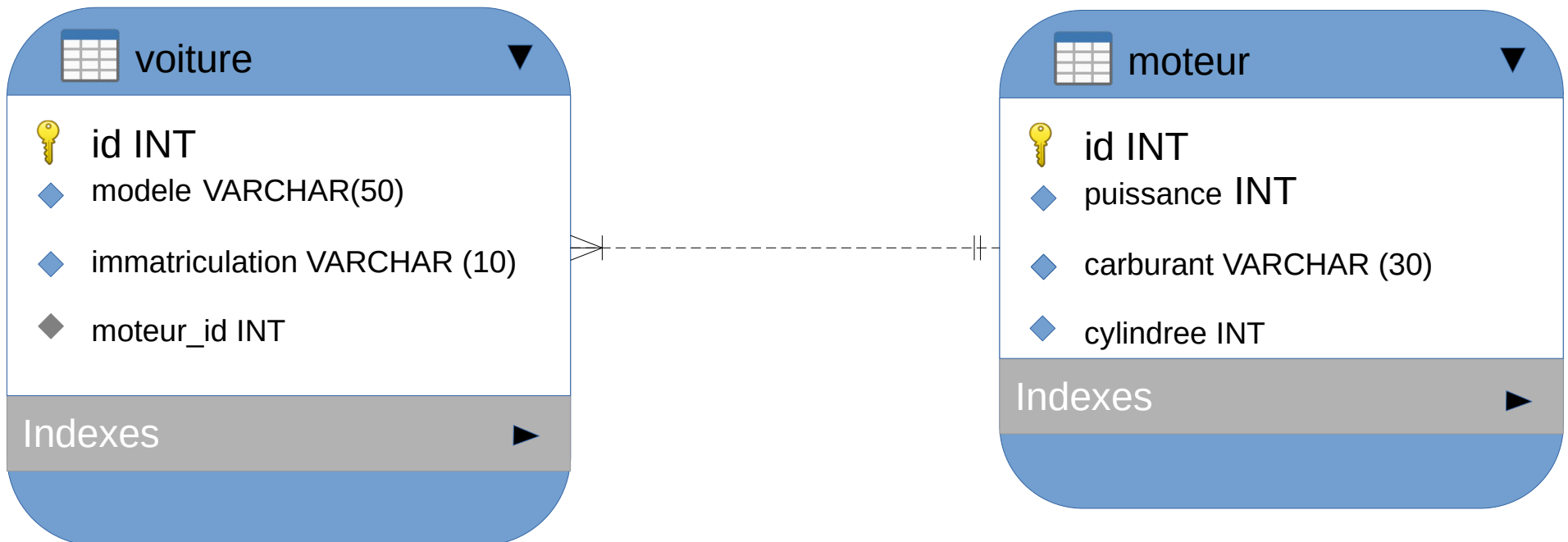
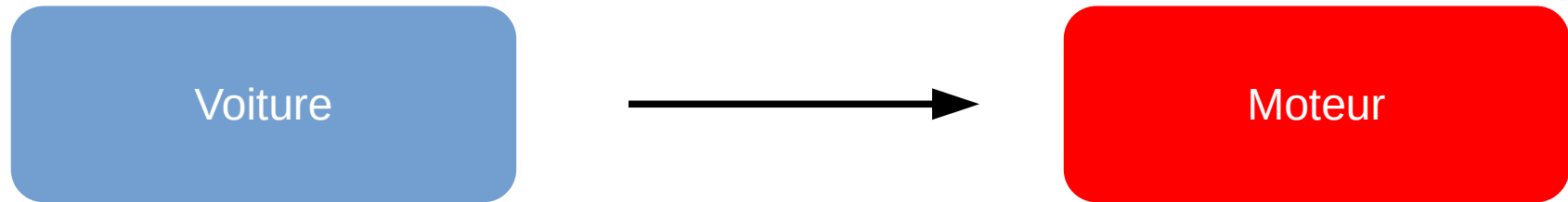
Mapping challenges

- Comment mapper plusieurs tables
- Comment organiser les relations entre les tables
- Comment Hibernate répond-il à ces besoins ?
 - @One-to-One , @Embedded, @Embedable
 - @One-to-Many , @Many-to-One
 - @Many-to-Many
- Imaginons les exemples de relations suivantes :

One-To-One

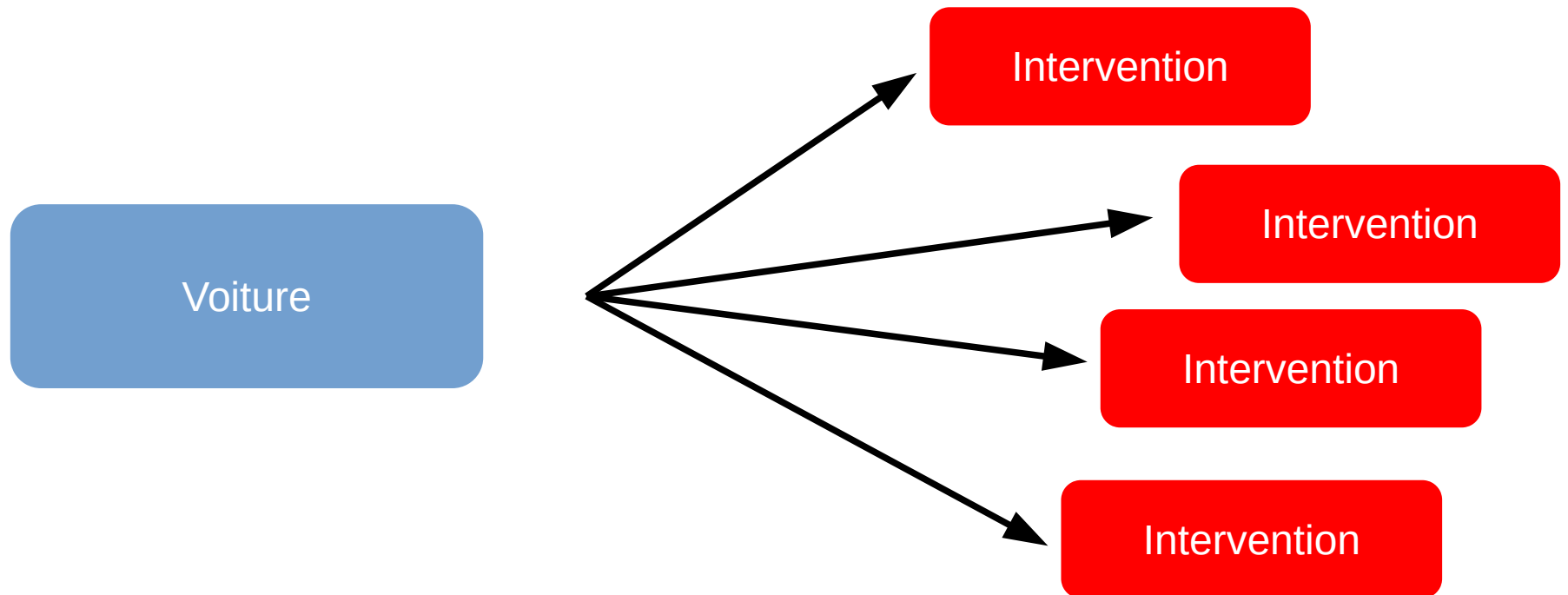
Une voiture possède un seul moteur.

Et ce moteur fait partie d'une seule voiture



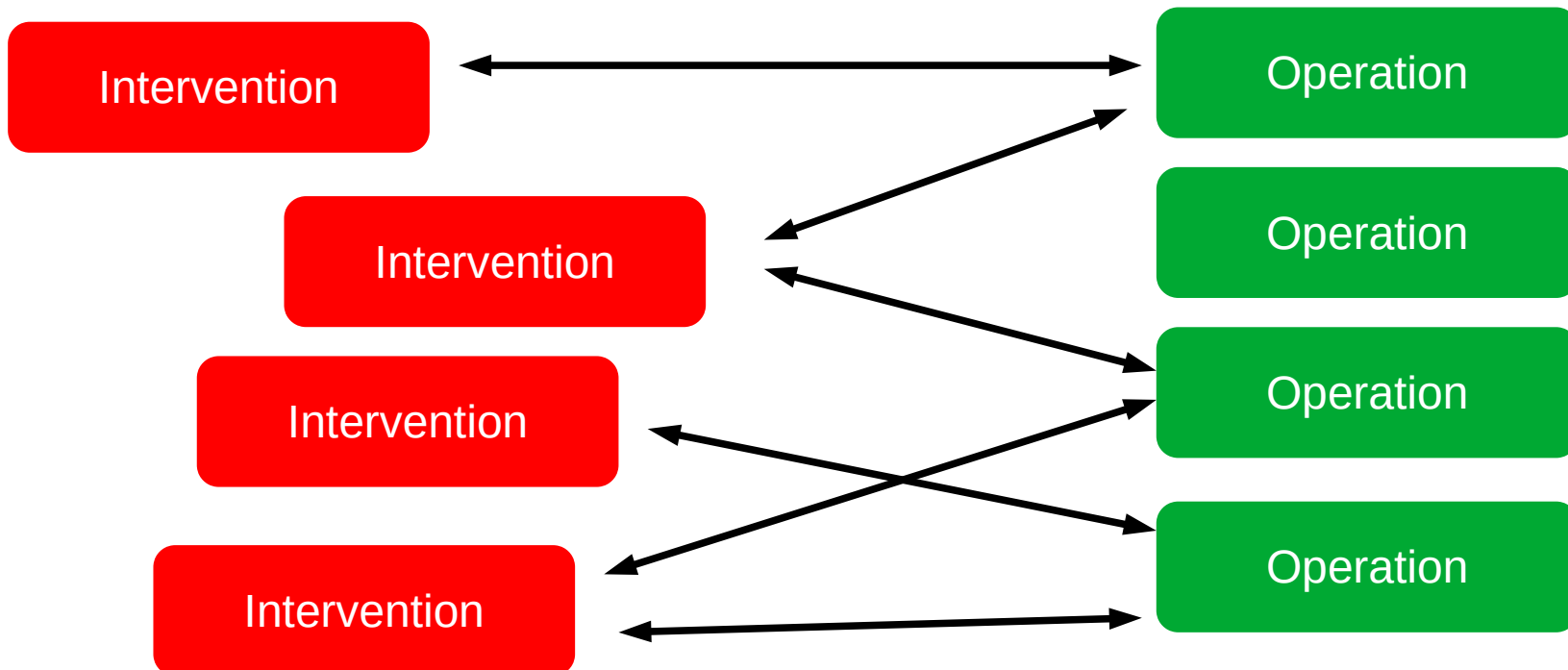
One-To-Many

- Une Voiture peut avoir fait l'objet de plusieurs visites de maintenance.
- Une visite de maintenance ne concerne, elle , qu'une seule voiture.



Many-To-Many

- Ay cours d'une Visite sont effectuées plusieurs opérations
- Une opération peut être effectuée au cours ~~cours~~ de plusieurs visites.



Concepts important Base de données

- Clés primaires

...

- Clés étrangères

un champs dans une table qui "pointe", qui se réfère, à une clé primaire d'une (autre) table.

Cela permet de relier les tables entre elles.

- Cascade

Exemple de clé étrangère

Table : Voiture

id	modele	immatriculation	moteur_id
1	picasso	CDF-548-13	100
2	kangoo	Zg-047-84	200

Colonne
Clé étrangère

Table : Moteur

id	puissance	cylindree	carburant
100	125	1800	essence
200	90	1600	diesel

Concepts important Base de données

- Clés primaires

...

- Clés étrangères

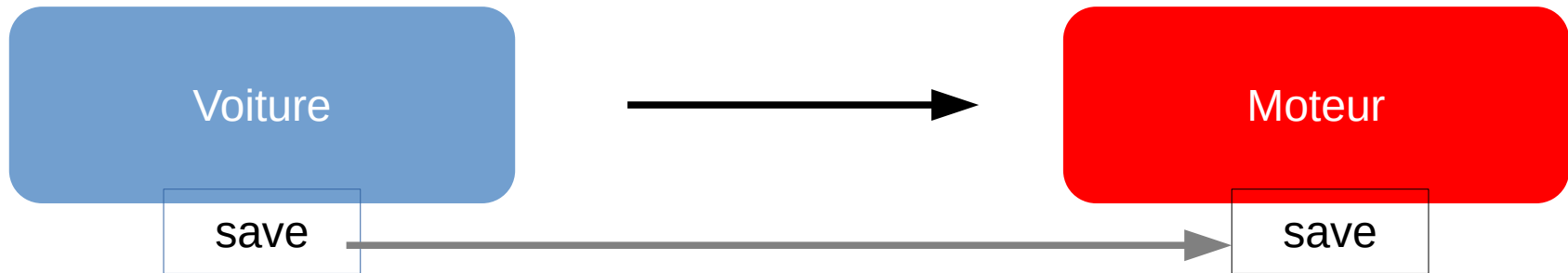
un champs dans une table qui "pointe", qui se réfère, à une clé primaire d'une (autre) table.

Cela permet de relier les tables entre elles.

- Cascade

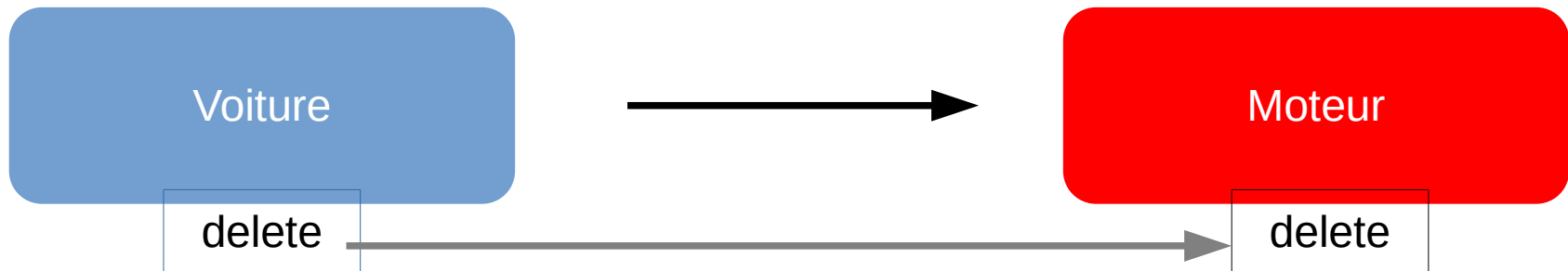
"Cascader" une opération , c'est appliquer la même opération à l'entité liée

Cascade



Si je créer une Voiture, c'est une opération , si elle est "cascadée" , je vais également créer un Moteur.

De même si je supprime une Voiture , la "suppression en cascade" supprimera le Moteur qui est lié à cette Voiture.



Suppression en cascade.

Table : Voiture

id	modele	immatriculation	moteur_id
1	piaggio	CDE-548-13	100
2	kangoo	Zg-047-84	200

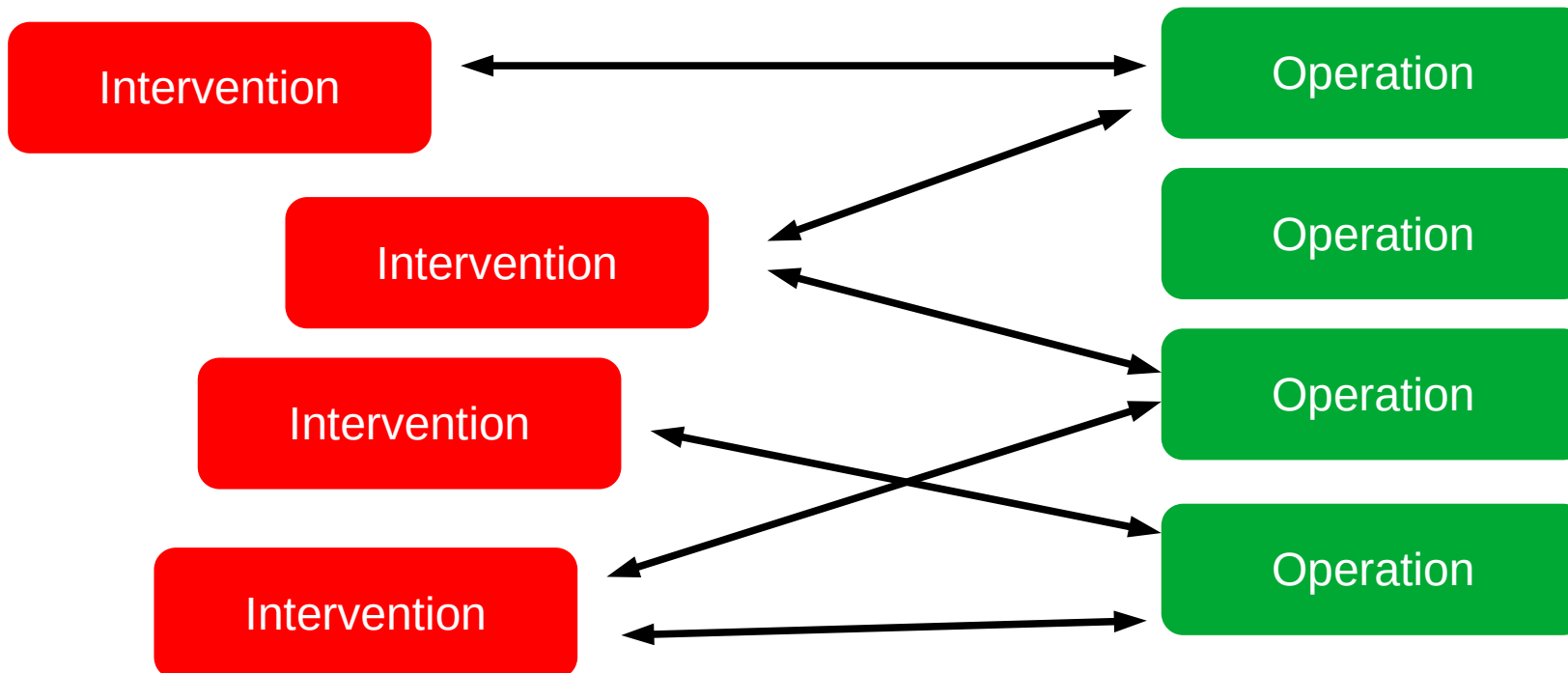
Table : Moteur

id	puissance	cylindree	carburant
100	125	1800	essence
200	90	1600	diesel

Attention avec le Cascade

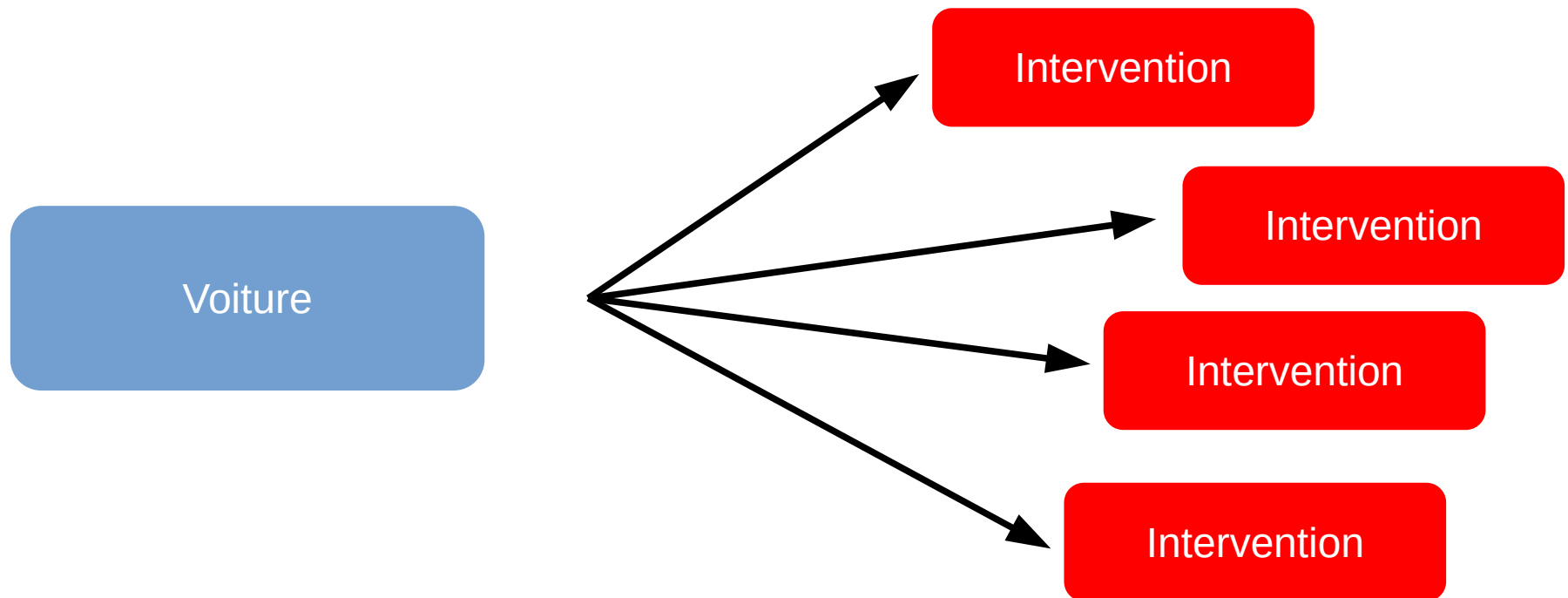
Dans certains cas il n'est pas judicieux de supprimer en cascade.

D'après vous quelle est la question qu'il doit se poser un développeur dans ce cas ?

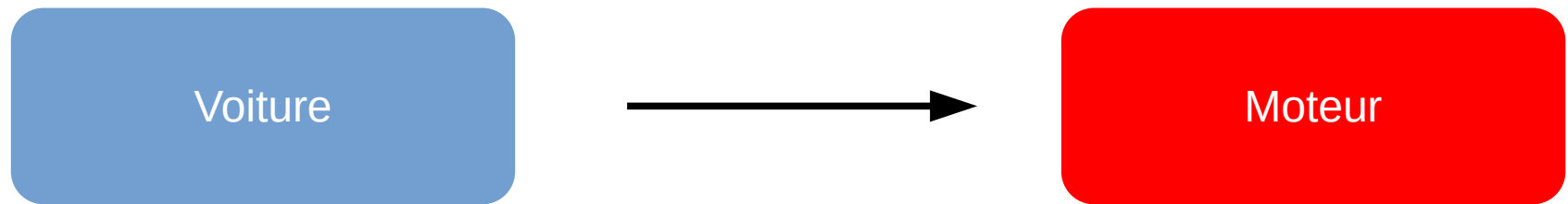


Fetch Types : Eager vs Lazy loading

- La question qui se pose ici c'est : "quand je récupère une donnée , dois-je récupérer tout ? "
- Eager signifie récupérer toute la donnée
- Lazy signifie que la donnée sera récupérable à la demande



Une relation Uni directionnelle



Une relation bi-directionnelle

