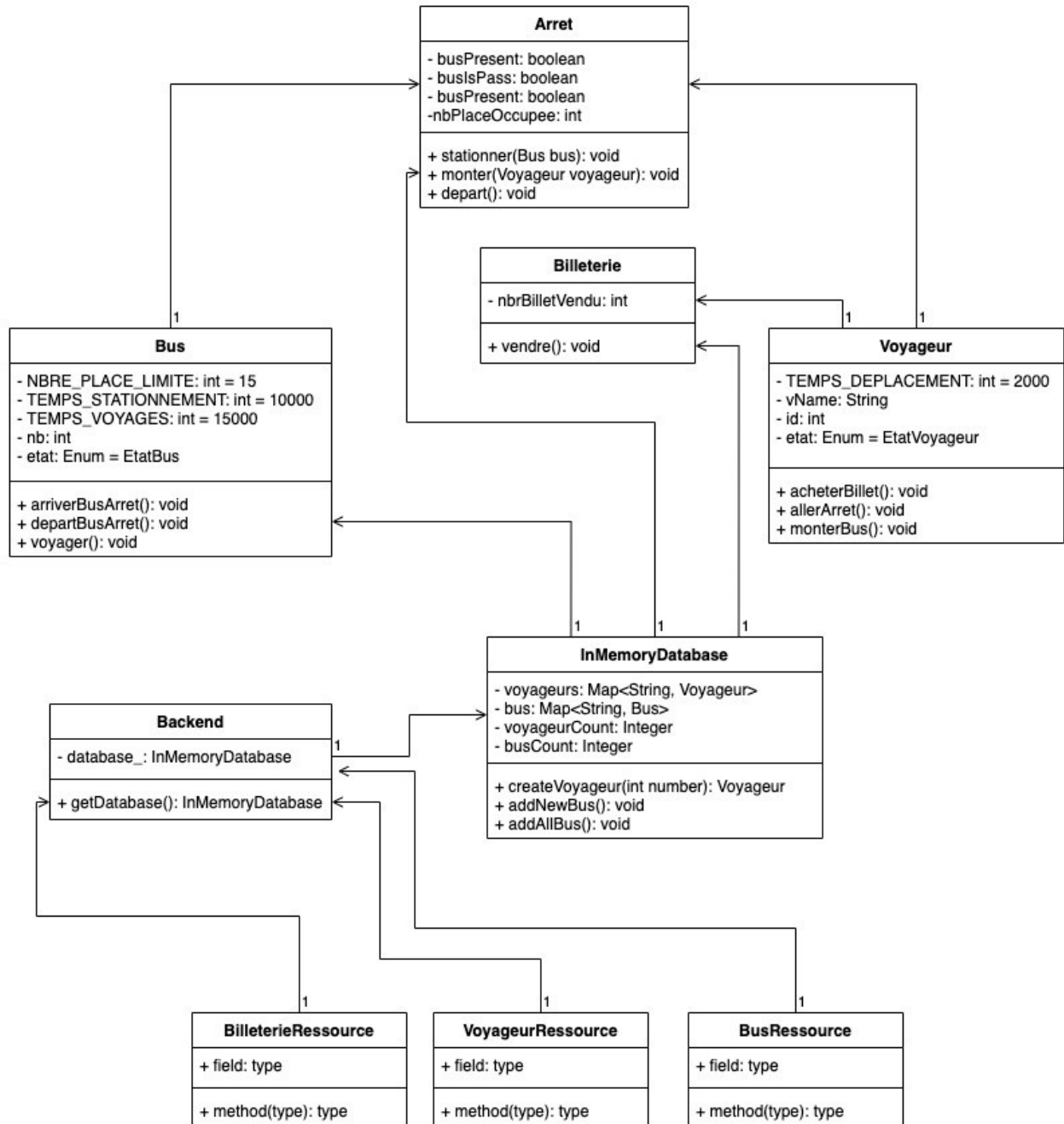


RAPPORT CSR

(a) modélisation

- ◆ Threads
 - Voyageur
 - Bus
- ◆ Objets partagés
 - Arret



(b) problèmes

- ◆ Quand il y'a un grand nombre de voyageurs à l'arrêt les voyageurs n'ayant pas pu monter à bord du bus ne le font pas au prochain bus qui stationne.
- ◆ La montée des voyageurs dans le bus et le stationnement des bus à l'arrêt rentre en concurrence du fait que les deux méthodes présentes dans l'objet partagé sont **synchronized**. Ainsi, il se produit parfois une inactivité des voyageurs.

(c) API

- ◆ Bus
 - ✓ récupérer l'état d'un bus
 - GET
 - */bus/state/{busId}*
 - *Content-type: "application/json"*
 - *Accept: "application/json"*
 - ✓ lister l'état des bus
 - GET
 - */bus/all/state*
 - *Content-type: "application/json"*
 - *Accept: "application/json"*
- ◆ Voyageur
 - ✓ ajouter un voyageur et le démarrer
 - POST
 - */voyageur/add*
 - *{"name":"2"}*
 - *Content-type: "application/json"*
 - *Accept: "application/json"*
 - ✓ ajouter plusieurs voyageurs et les démarrer
 - POST
 - */voyageur/addAll*
 - *{ "voyageurs":[{"name":"2"}, {"name":"2"}] }*
 - *Content-type: "application/json"*
 - *Accept: "application/json"*
 - ✓ récupérer l'état actuel d'un voyageur
 - GET
 - */voyageur/state/{voyageurId}*
 - *Content-type: "application/json"*
 - *Accept: "application/json"*

- ✓ lister les voyageurs
 - GET
 - */billeterie/tickets*
 - *Content-type: "application/json"*
 - *Accept: "application/json"*

◆ Billeterie

- ✓ récupérer le nombre ticket vendu
 - GET
 - */billeterie/tickets*
 - *Content-type: "application/json"*
 - *Accept: "application/json"*

(d) infos

(e) trace