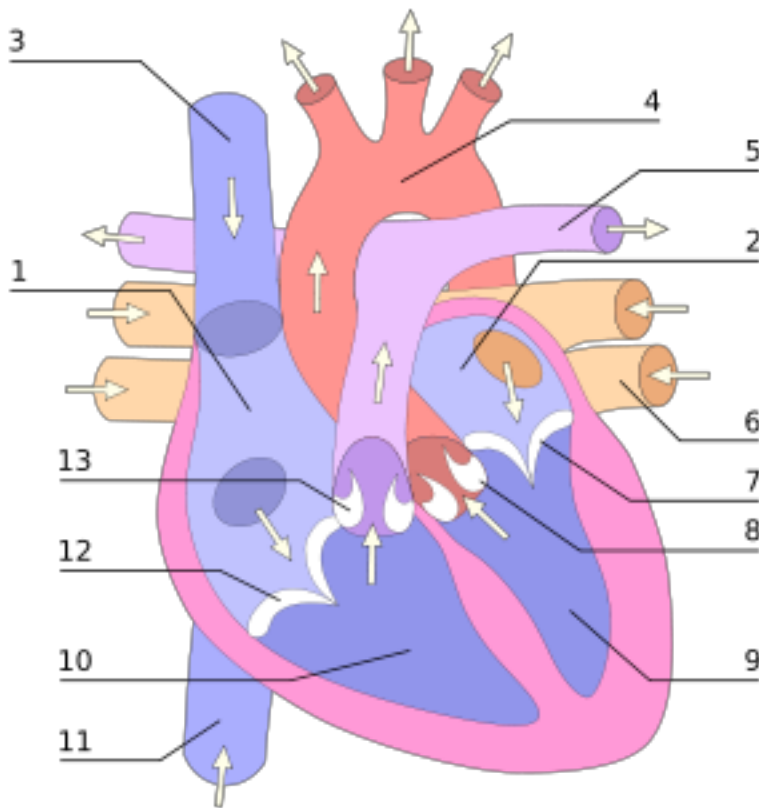




# Démarche de modélisation

- **Décomposition système/sous-système**
  
- **Identification des composants logiciels et de leurs interfaces**
  
- **Identification des composants matériels et de leurs interfaces**
  
- **Description du comportement des composants**
  - Synchronisation du générateur de pulsation avec le fonctionnement naturel du cœur
  - Modes de fonctionnement et transitions de mode

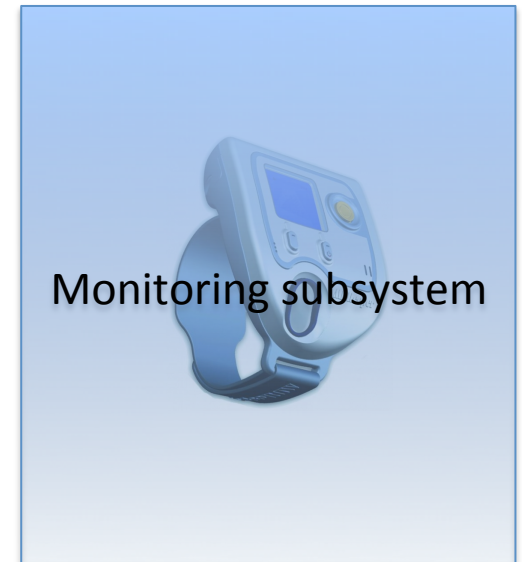
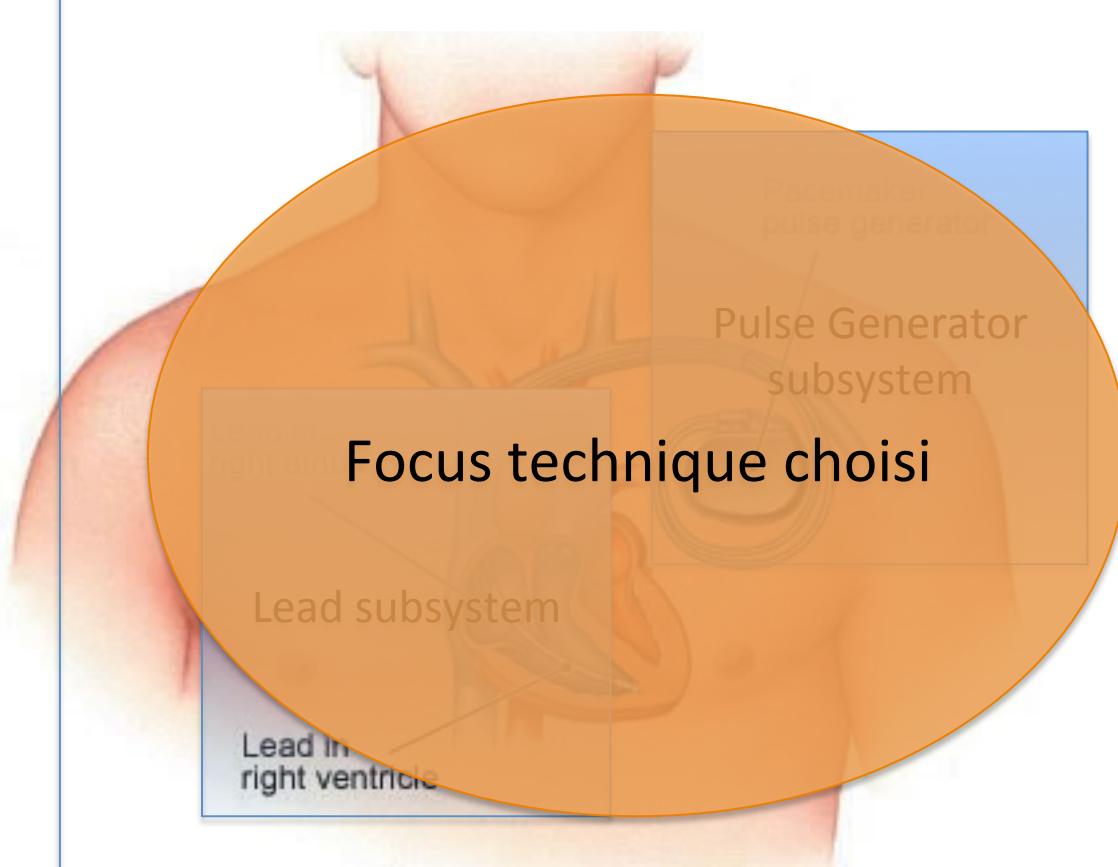
# Bref rappel de biologie



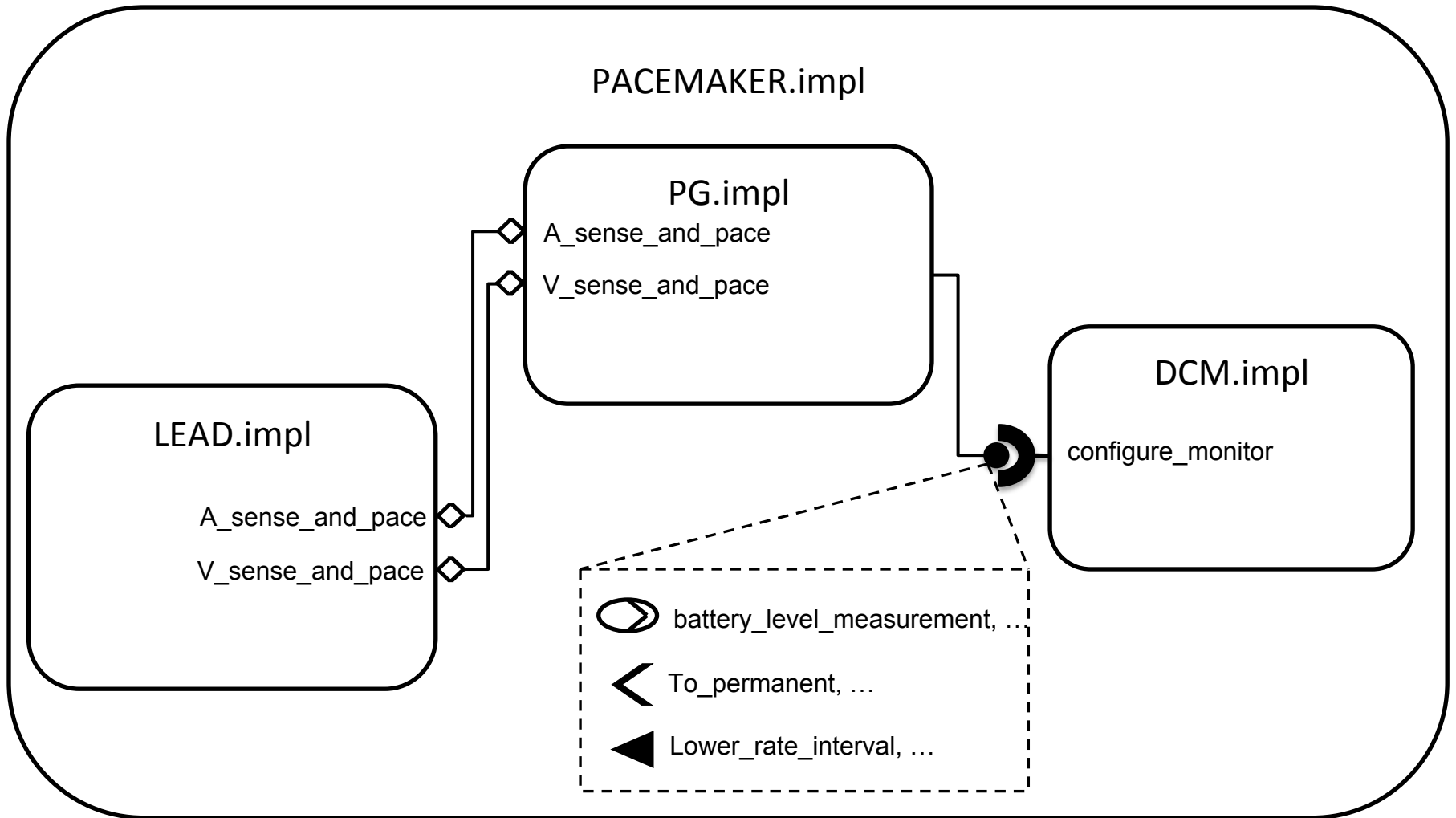
1. Atrium droit
2. Atrium gauche
3. Veine cave supérieure
4. Aorte
5. Artère pulmonaire
6. Veine pulmonaire
7. Valve mitrale (atrio-ventriculaire)
8. Valve aortique
9. Ventricule gauche
10. Ventricule droit
11. Veine cave inférieure
12. Valve tricuspide (atrio-ventriculaire)
13. Valve sigmoïde (pulmonaire)

# Décomposition systèmes/sous-systèmes

## PACEMAKER system



# Modèle AADL: système global (Question 1)



# Identification des composants logiciels et de leurs interfaces

## ■ Pulse Generator

- Un processus de contrôle
  - Entrées: détection de battements (ventriculaire/atriale)
  - Sorties: stimulation (ventriculaire/atriale)
- Une de tâches de contrôle
  - Mêmes entrées/sorties
- Une ou plusieurs tâches de configuration/supervision pour s'interfacer avec le DCM

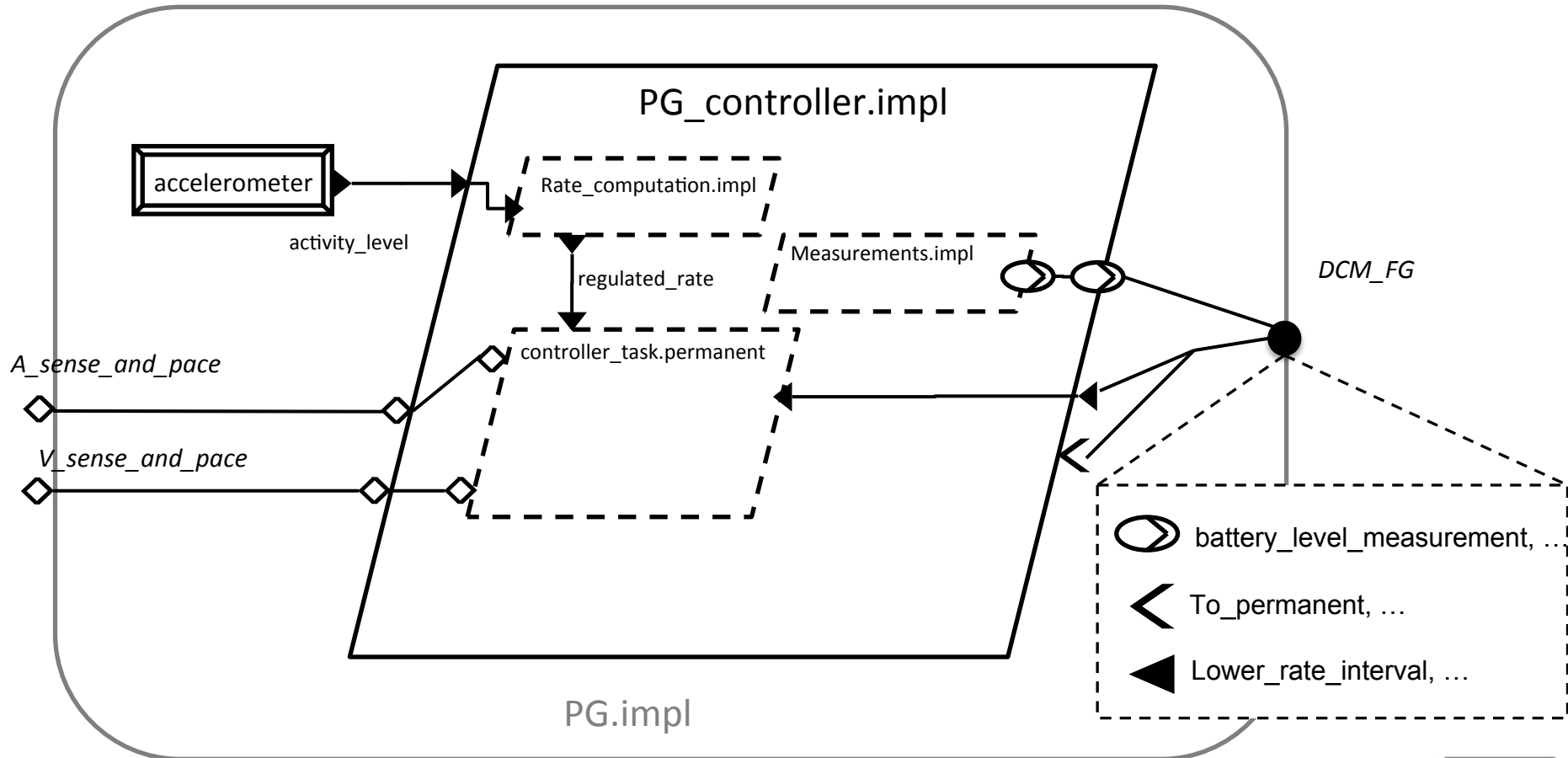
## ■ Ensembles de connections triviales:

- Périphériques de capture → entrées du processus de contrôle
- Sorties du processus de contrôle → périphériques de stimulation

# Composants logiciels

## (Question 2 solution étendue)

### ■ PG: processus de controle

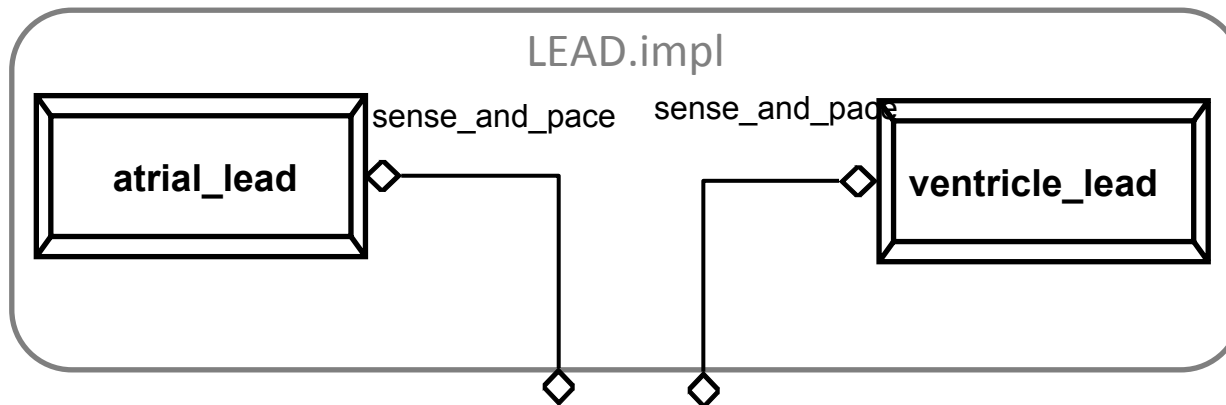


# Composants du sous-système LEAD

## (Question 3)

### ■ Leads

- Capteurs (ventriculaire/atriale) des battements naturels
- Stimulation (ventriculaire/atriale) du coeur



# Modèle AADL: Lead system

## (Question 3)

```
device compartment_lead
  features
    sense_and_pace: in out event port;
end compartment_lead;

system implementation LEAD.impl
  subcomponents
    atrial_lead: device compartment_lead;
    ventricle_lead: device compartment_lead;
  connections
    atrial_lead. sense_and_pace -> A_sense_and_pace;
    ventricle_lead. sense_and_pace -> V_sense_and_pace;
end LEAD.impl;
```



# Plate-forme d'exécution DCM et PG

## (Question 3)

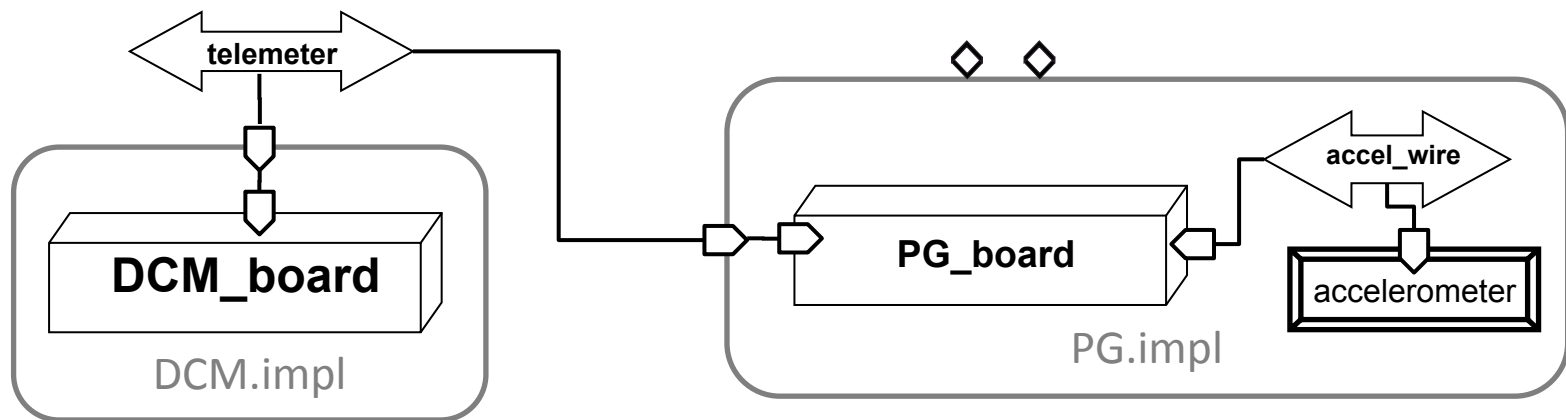
### ■ Pulse Generator

- Ressource d'exécution des fonctionnalités de régulation cardiaque
- Accéléromètre

### ■ DCM

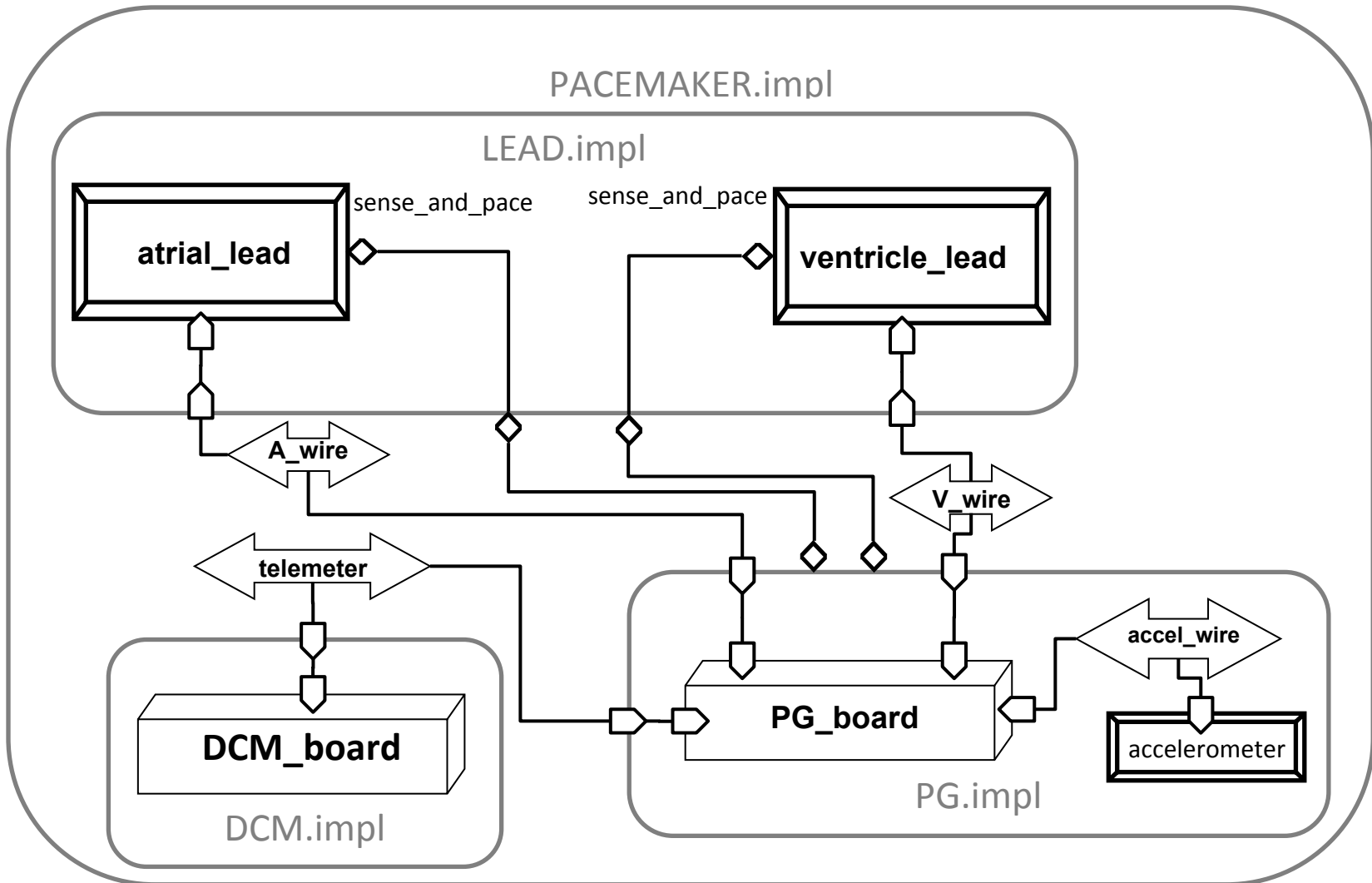
- Ressource d'exécution des fonctionnalités de régulation cardiaque

### ■ Medium de communication entre DCM et PG



# Composants matériel

## (Question 3)





# Description du comportement

- **Prédominance des modes de fonctionnement dans la définition du comportement**
- **Dont la portée peut-être décomposé en**
  - Impact des changements de mode sur la structure de l'architecture (composants/interfaces/connections...)
  - Impact des changements de mode sur le comportement (périodes/dispatch\_protocole/fonctionnalités...)
- **Enfin, à mode constant, certains éléments de l'architecture sont “programmable” ou “variable”**
  - Exp. le rythme de battement (période d'une fonctionnalité) dépend des mouvements de l'utilisateur (données fournis par un accéléromètre)

# Ordonnancement des tâches

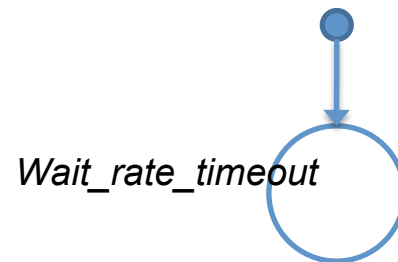
## (Question 4)

- ***Scheduling\_Protocol, Dispatch\_Protocol, et éventuellement Period et Priority***
  - PG\_Controller
  - Rate\_Computation
  - Measurements
  
- **Attention à maintenir la cohérence entre les propriétés**
  - Ex: pas de période pour une tâche apériodique
  - Ex: pas de tâche apériodique si la politique d'ordonnancement est RMS
  - Ex: Period ne peut prendre qu'une valeur entière constante

# Comportement de la tâche de contrôle dans le mode DOOR (Question 5)

## ■ Contraintes temporelles:

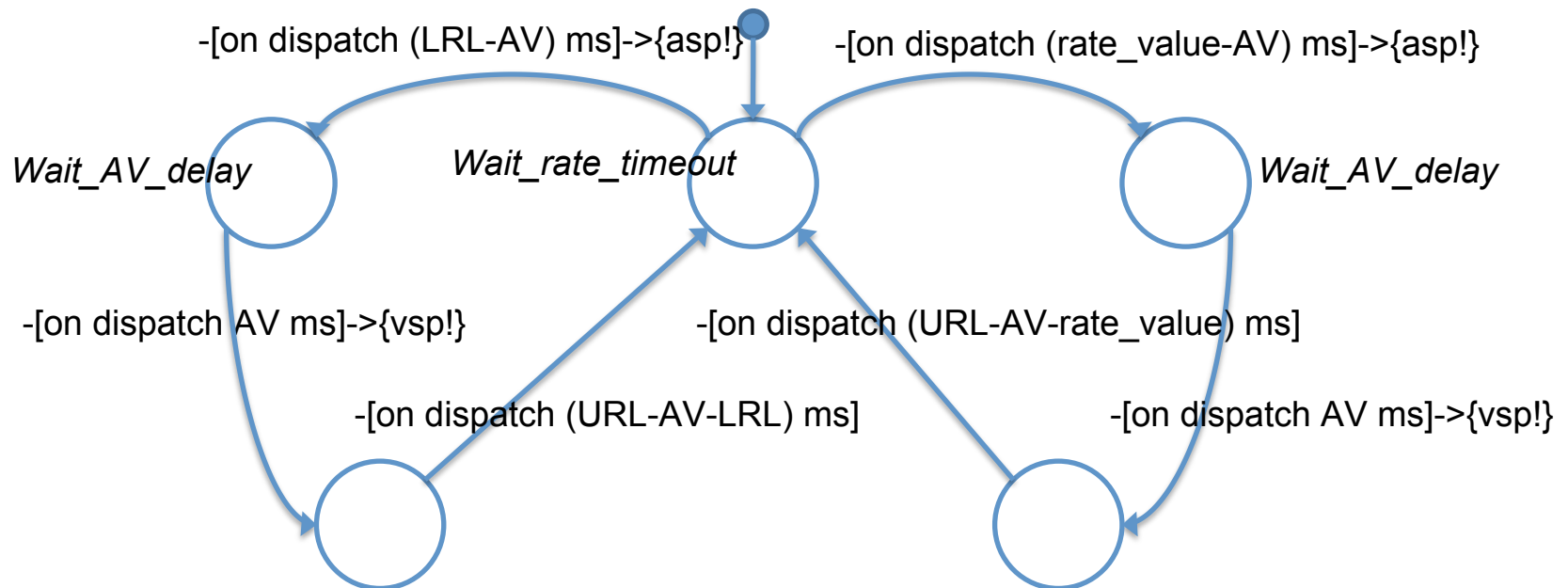
- LRL, AV, URL, et rate\_value



# Comportement de la tâche de contrôle dans le mode DOOR (Question 5)

## ■ Contraintes temporelles:

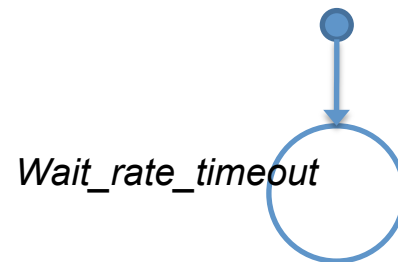
- LRL, AV, URL, et rate\_value



# Comportement de la tâche de contrôle dans le mode AAI (Question 6)

## ■ Contraintes temporelles:

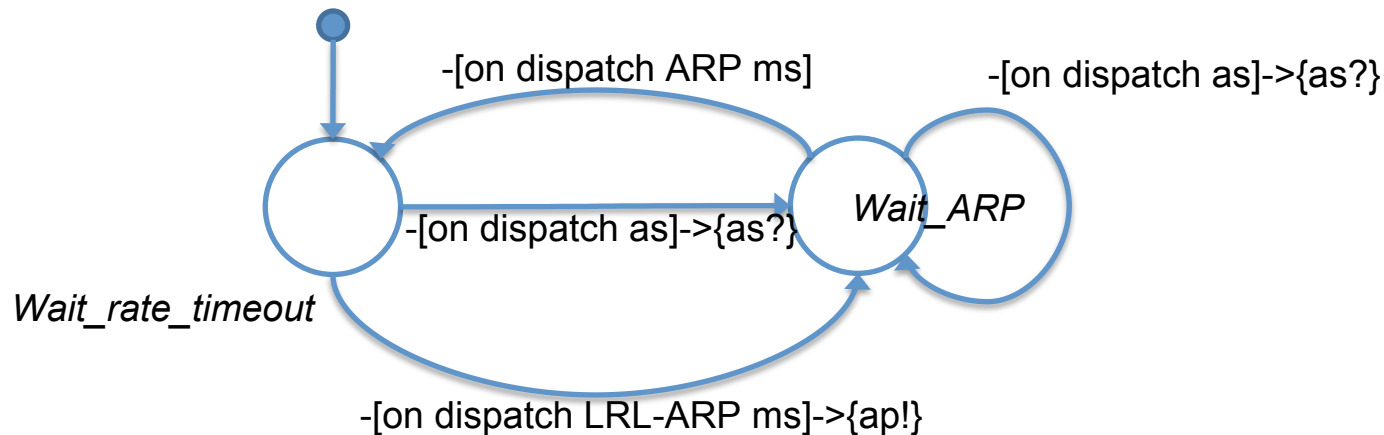
- LRL, AV, et URL



# Comportement de la tâche de contrôle dans le mode AAI (Question 6)

## ■ Contraintes temporelles:

- LRL, AV, et URL





# Défauts de la conception: partie comportementale (Question 7)

- La solution proposée permet-elle d'assurer le respect des exigences temporelles?
- Quelque soient les conditions d'utilisation?

# Défauts de la conception: partie comportementale (Question 7)

- La solution proposée permet-elle d'assurer le respect des exigences temporelles?
- Quelque soient les conditions d'utilisation?
  - En cas de modification des valeurs LRL, etc... sur les data ports d'un thread ???
- Quelle solution à ce problème?



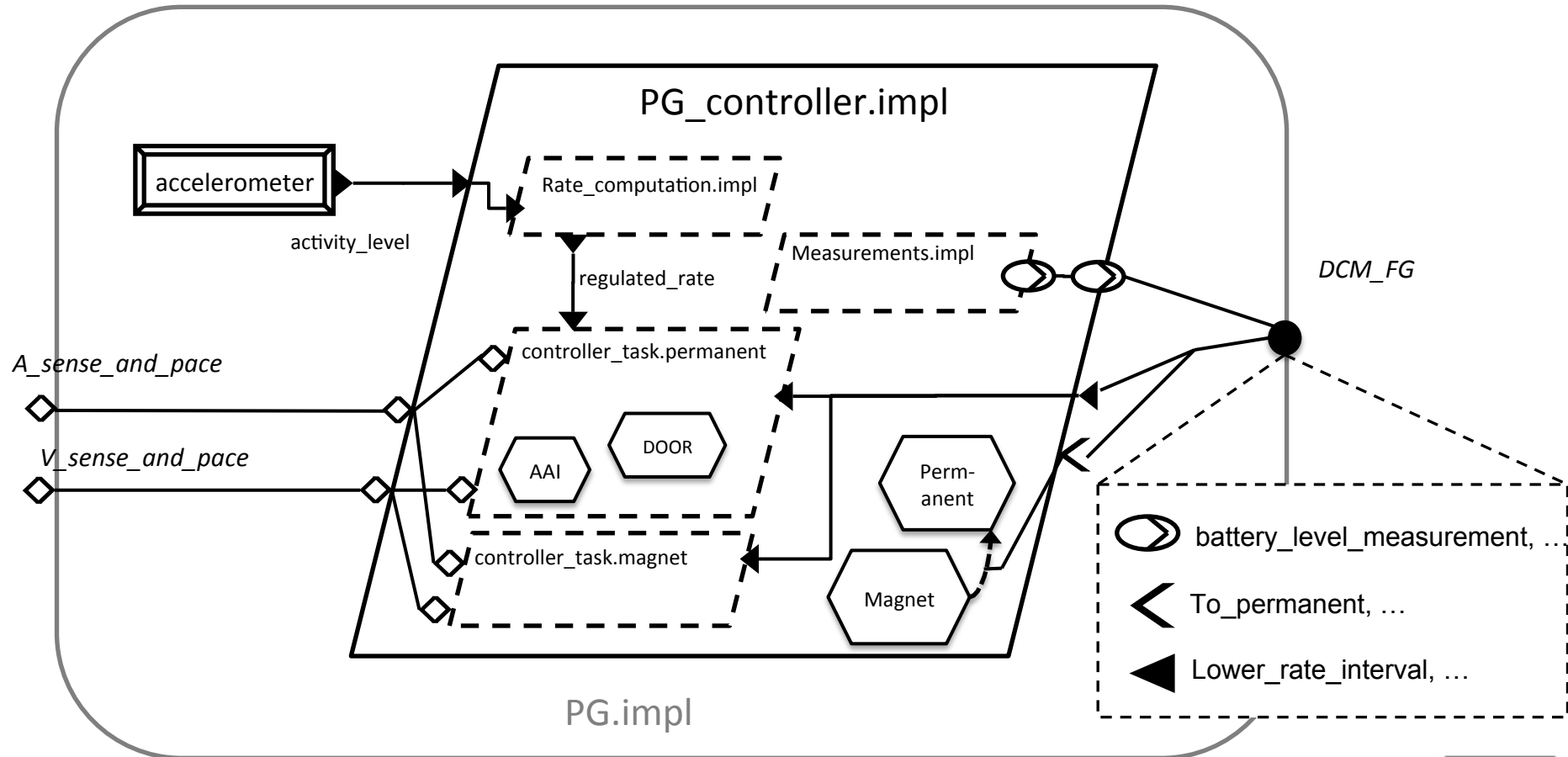
# Modes de fonctionnement

## Présence importante des modes de fonctionnement

- 5 modes principaux: Permanent/Temporary/Pace-Now/Magnet/Power-On Reset
- + de 30 Sous-modes:
  - Permanent: Off, DDDR, VDDR, DDIR, DOOR, VOOR, AOOD, VVIR, AAIR, DDD, VDD, DDI, DOO, VOO, AOO, VVI, AAI, VVT and AAT.
  - Temporary: OVO, OAO, ODO, and OOO.
  - Pace-Now: VVI.
  - Magnet: AXXX, VXXX, DXXX en fonction du mode source; et sous-modes BOL, ERT, ERN.
  - Power-On reset: VVI.

# Modélisation des modes de fonctionnement (Question 8)

## ■ PG: processus de controle



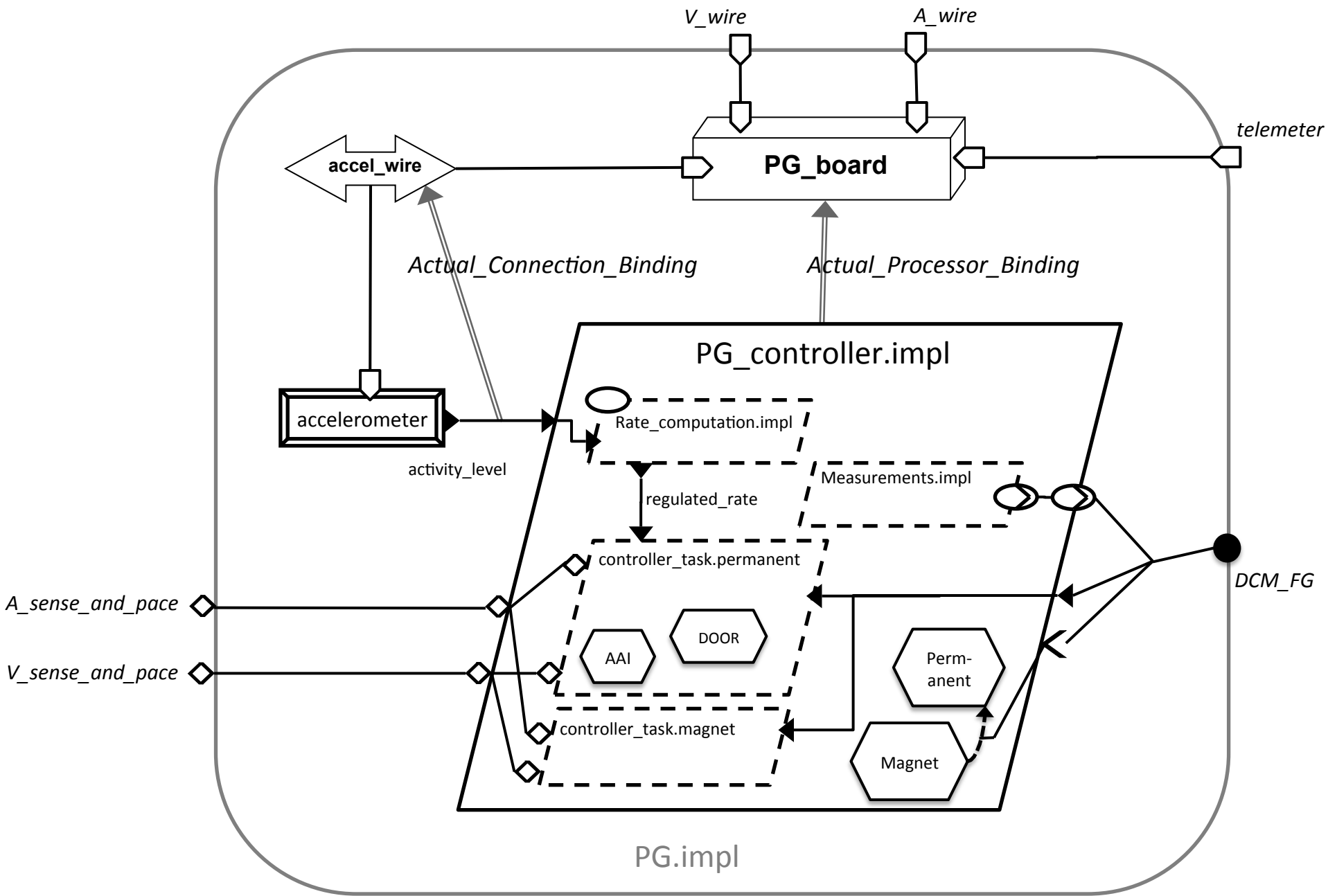
# Défauts de la solution

## (Question 8)

- **La solution proposée permet-elle d'assurer le respect des exigences temporelles?**
  
- **Quelque soient les conditions d'utilisation?**
  - En cas de changement de mode ?
  
- **Quelle solution à ce problème?**

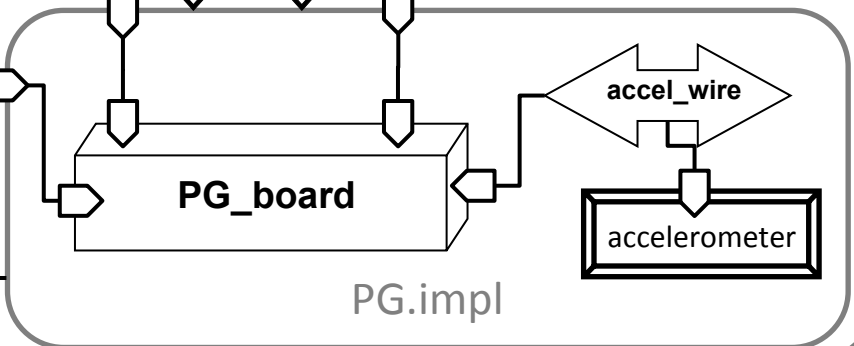
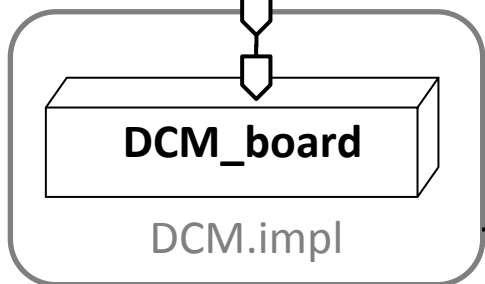
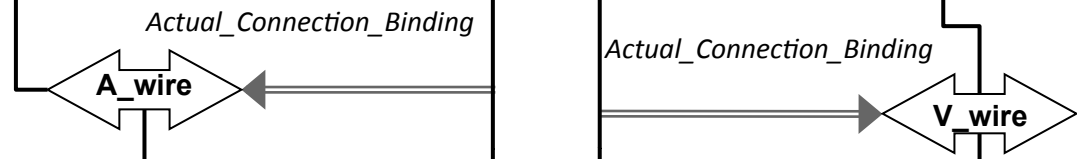
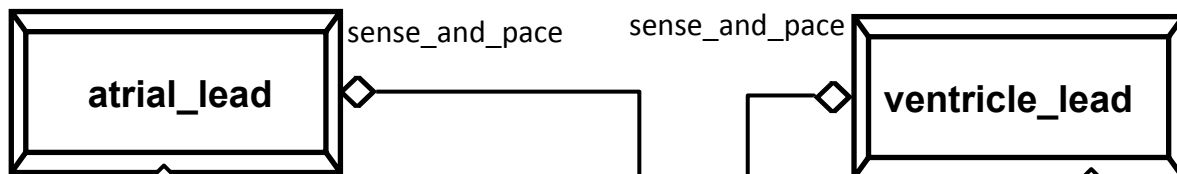
# Déploiement complet (Question non répertoriée)

- Représenter le *binding* entre composants et composants matériels
- Représenter les binding entre flux logiciel et transport matériel

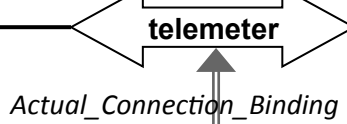


# PACEMAKER.impl

## LEAD.impl



*DCM\_FG*







# Conclusion

- **Le PACEMAKER est un système beaucoup plus complexe qu'il paraît être de prime abord**
- **Les informations contenues dans la spécification système sont très éloignées de la conception finale**
- **La modélisation permet de lever des ambiguïtés de fonctionnement à condition d'avoir**
  - une expertise forte dans le langage de modélisation choisi
  - une expertise suffisante dans le système à concevoir (donc son domaine d'application)
- **Quelle exploitation pour ces modèles?**
  - Communication
  - Analyse et vérification
  - Génération de code