

## Design correction du prof

Task/Thread → multithread

Broker

accept  
connect

RDV

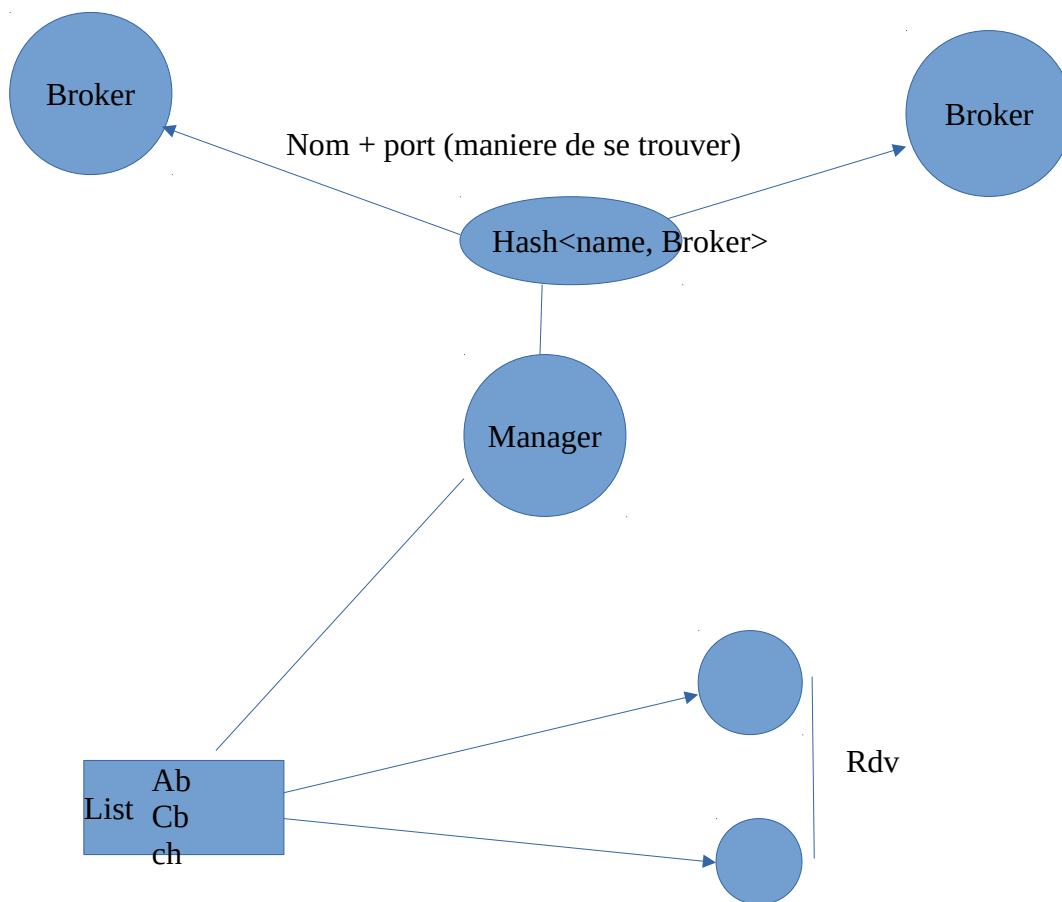
Channel (créé par accept/connect)

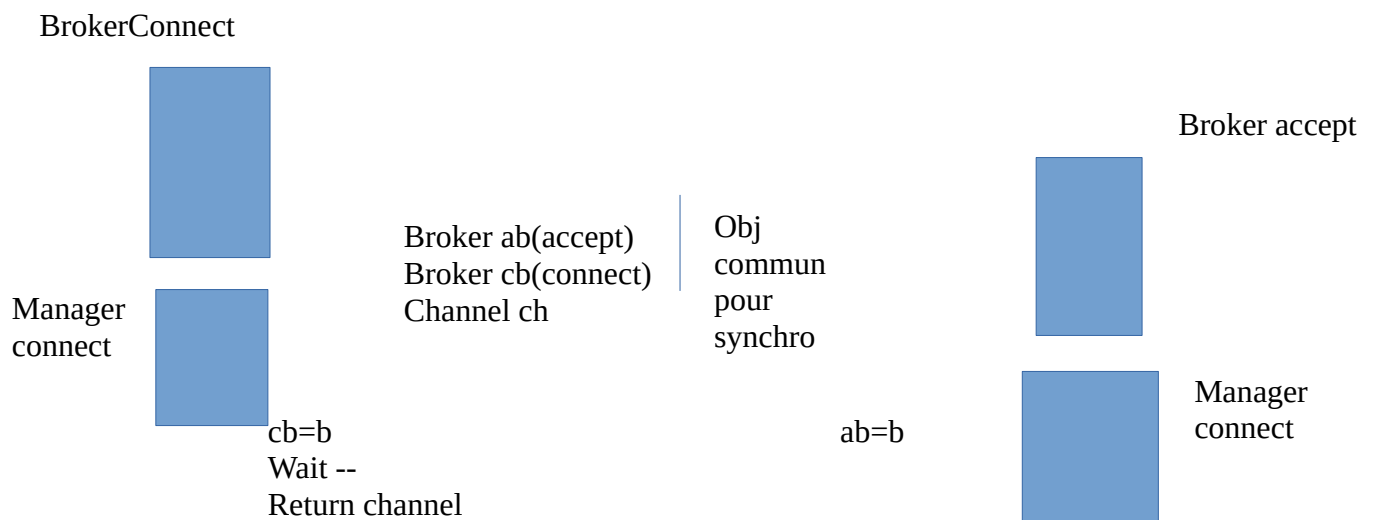
read/write

Java → obj / monitor / wait/notify()

But rdv = créer Channel

CircularBuffer





RDV pour 1 accept 1 conenct (pas suffisant car multithread)

```

synch
connect()
    if(ab!=null){
        ch
    }else{
        wait() ;
    }return ch

```

Besoin de N version de ab,cb, ch pour chaque rdv

Broker ab(accept)  
Broker cb(connect) ———— Objet RDV  
Channel ch  
String nom  
Int port

Manager traque Broker et crée rdv  
Broker viennent sur rdv pour wait/notify

**Manager**

```
connect(Broker, Nom, Port)
list(ou hash)<rdv> rdv
```

accept(Broker, Nom, Port)

**Hypothèse :** 1 seul accept nom-port car accept bloquant, plusieurs connect

si plusieurs connect avant accept → 1accept-connect, la reste s'endorment , un connect peut se bloquer sur manager si déjà connect sur nom-port (peut pas ajouter un rdv a la liste si déjà même nom-port)

- si connect alors que déjà accept → connexion établie
- si connect alors que déjà connect → on se met en attente
- si seul connect et pas accept → on se met dans rdv
- si accept et y a déjà connect → connexion

si accept et pas déjà rdv → on se met dans rdv

broker viennent sur manager où il y a la liste rdv et délègue le fait que les 2 broker se trouvent à rdv

## Design correction

