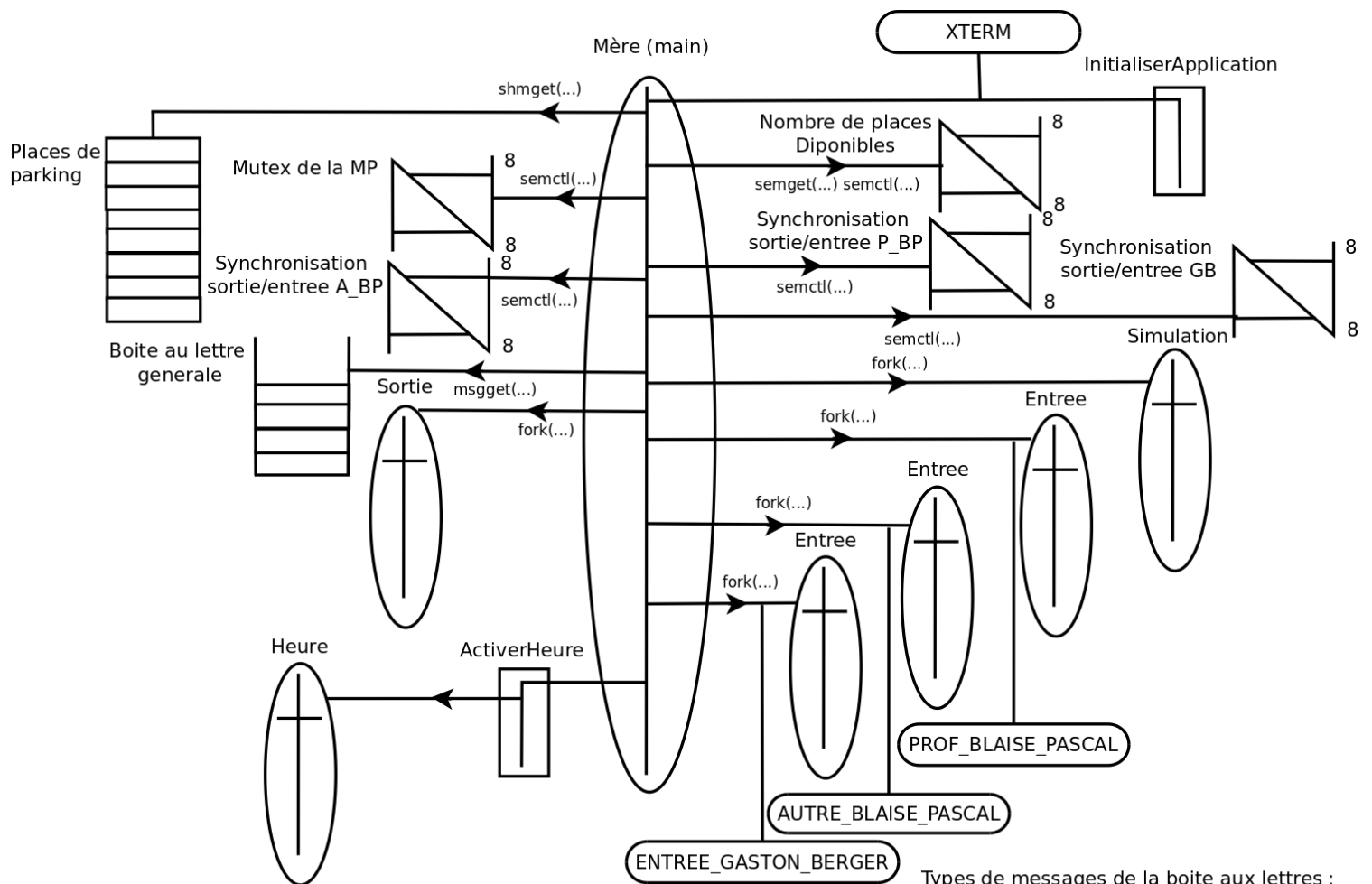


# Compte Rendu TP Processus

## I. Conception :

### 1. Tâche Mère (main) :

#### A. Initialisation :



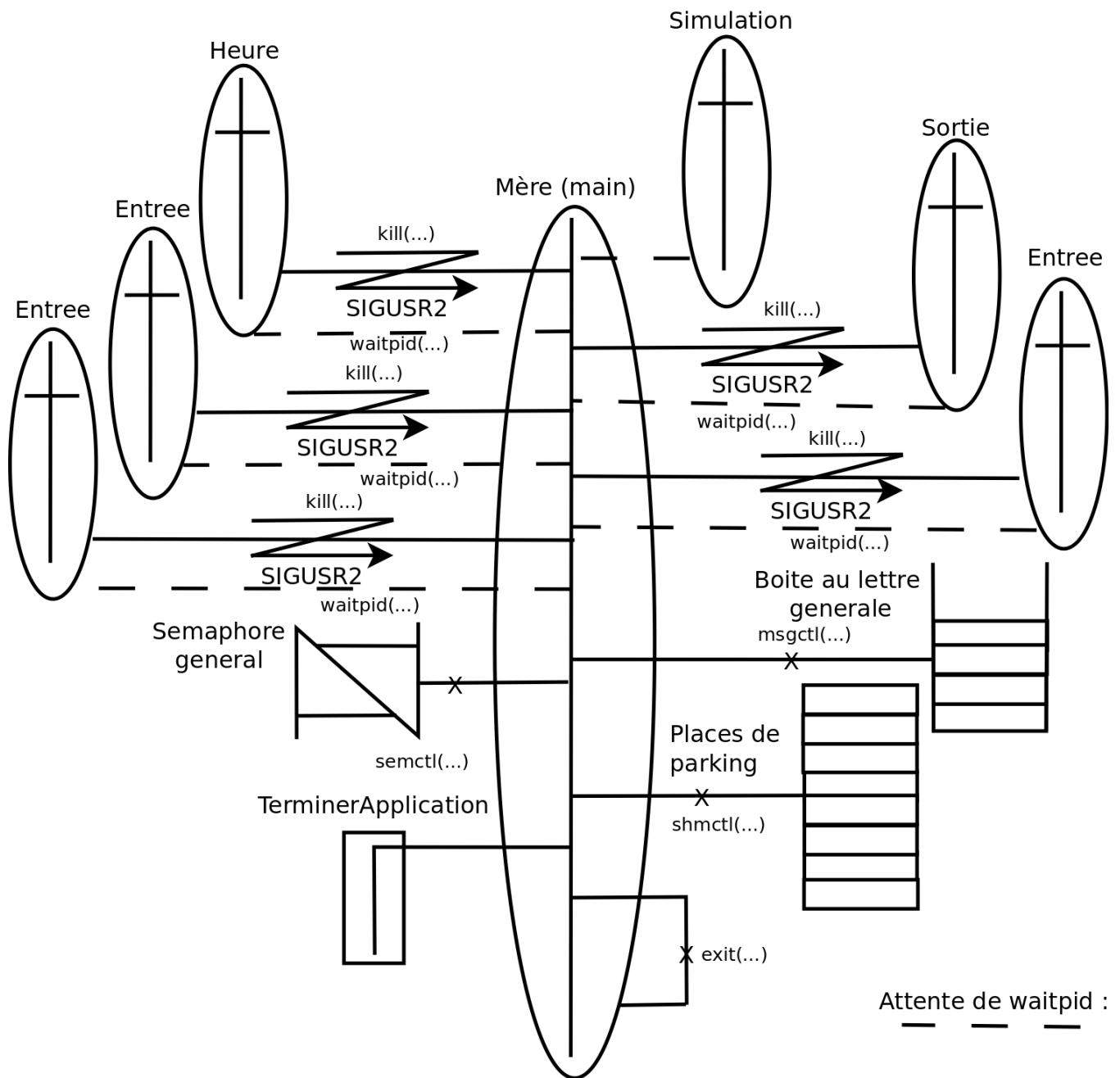
Types de messages de la boite aux lettres :

```
ComandeSorsrtie {
    unsigned numParking }
```

```
Requete {
    TypeUsager typeUsager
    unsigned immatriculation
    time_t heureArrivee }
```

```
Voiture {
    TypeUsager typeUsager
    unsigned immatriculation
    time_t heureArrivee
    time_t heureDepart }
```

## B. Destruction

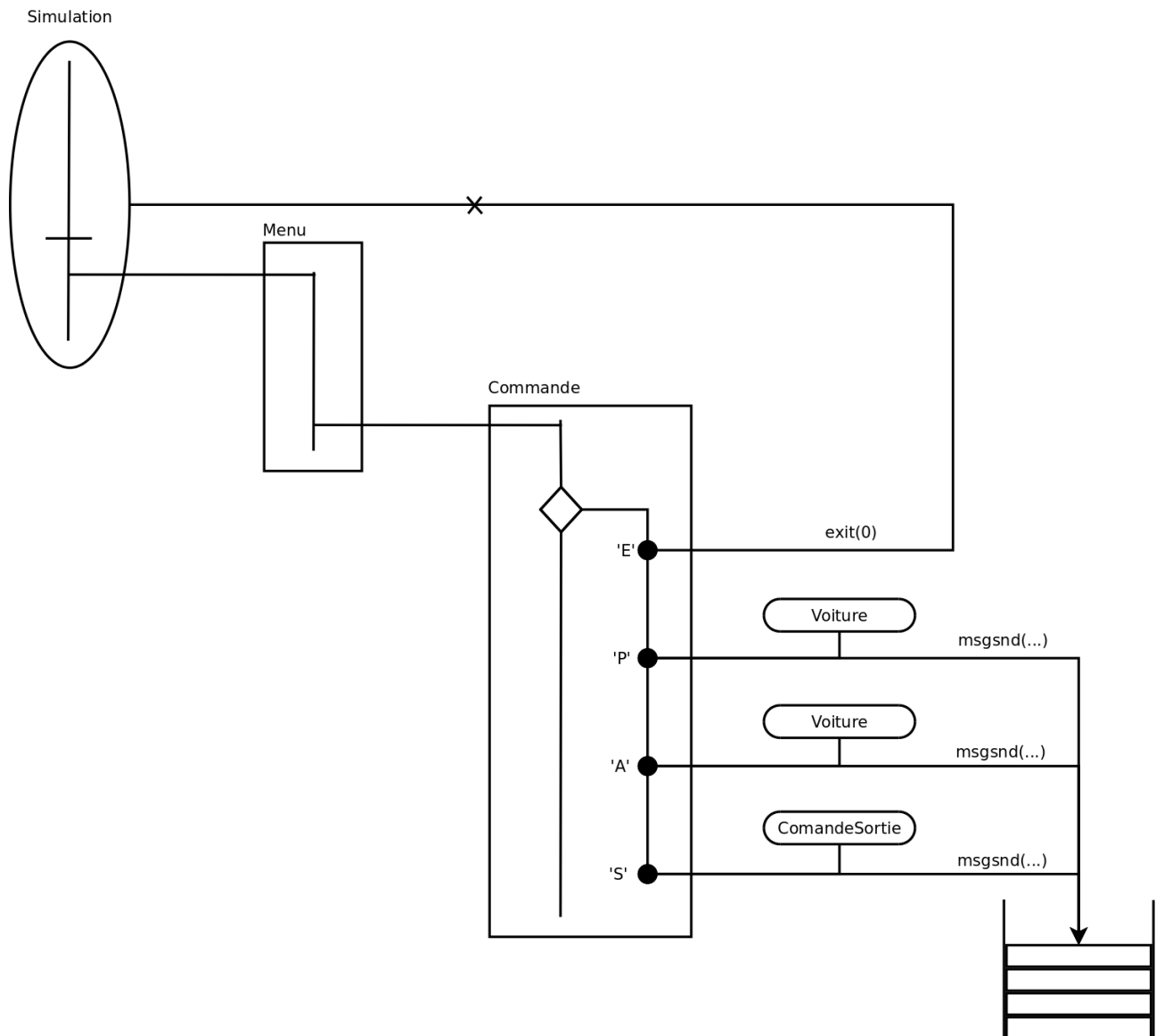


## 2. Tâche Simulation

### A. Initialisation et destruction

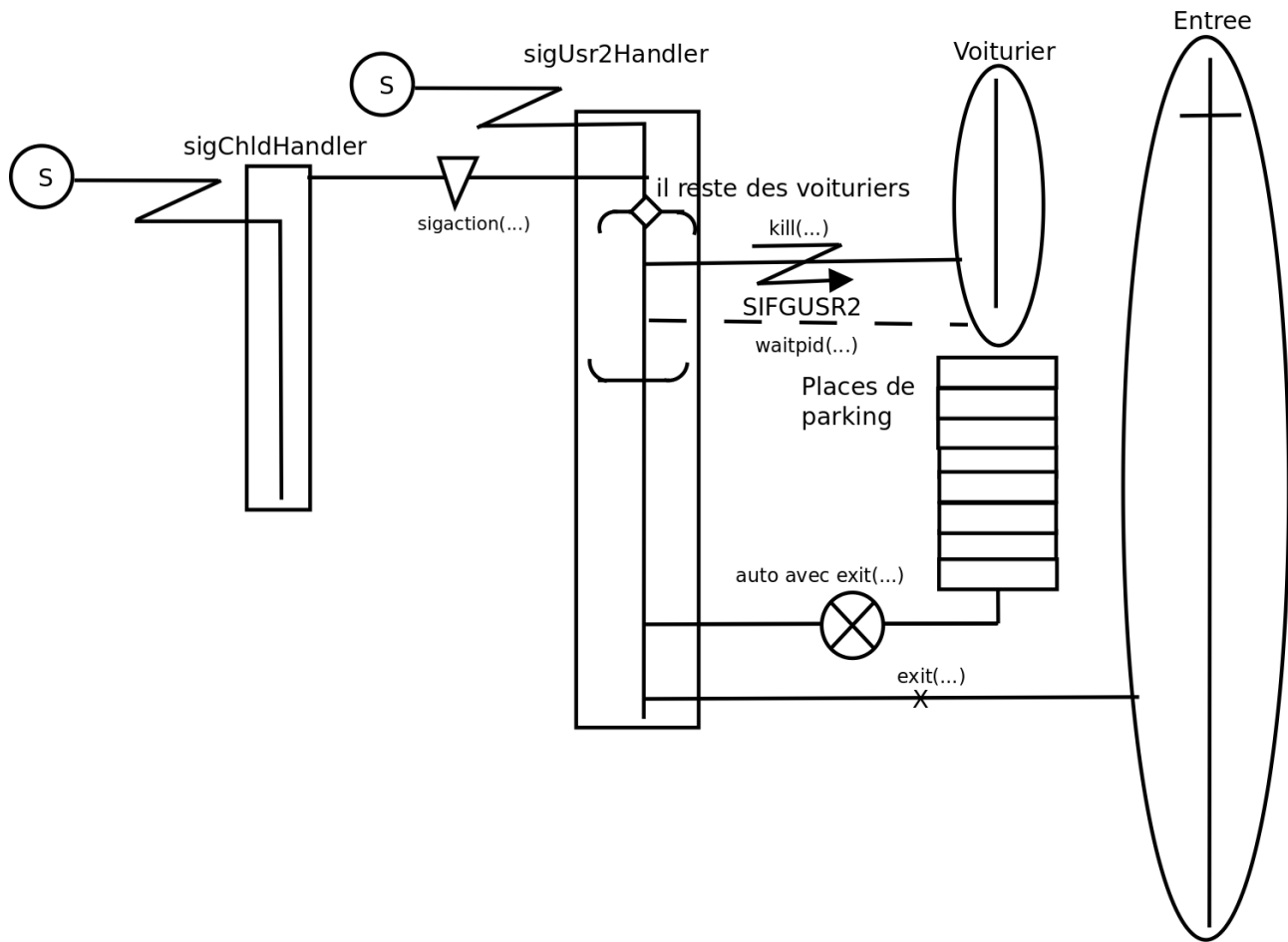
Ces phases ne présentent aucune instruction

### B. Moteur



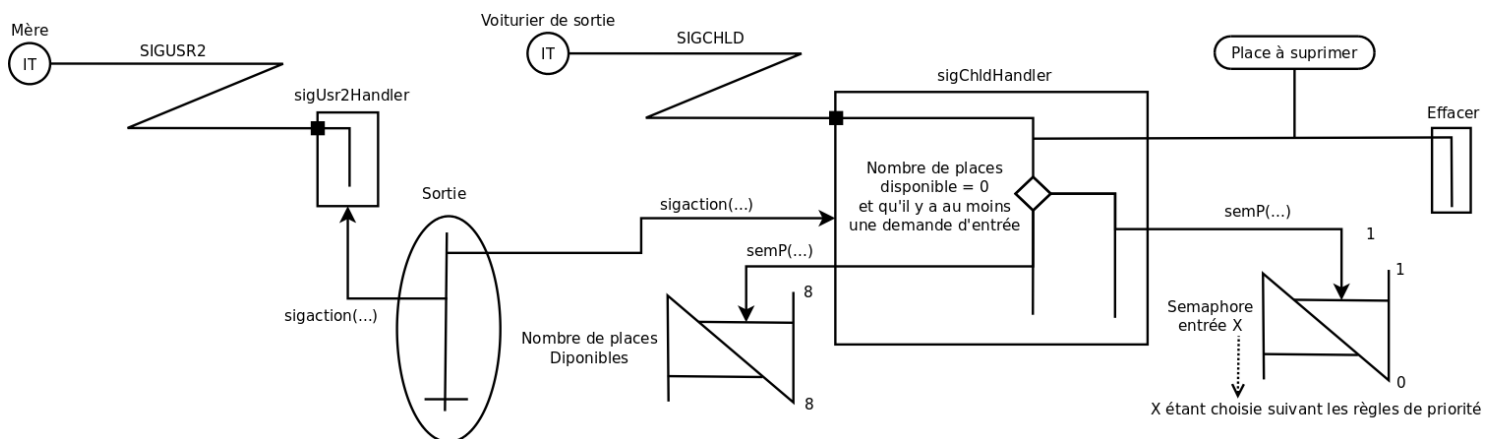
The diagram illustrates a process (represented by a large oval) with two signal handlers and a shared resource. The process has a vertical line representing its execution flow, with a '+' sign at the bottom indicating the start of execution. On the left, a vertical rectangle represents the **sigChldHandler**. A signal source **S** (circle) sends a signal to this handler. The handler sends a **sigaction(...)** signal to the process, which then sends a **SIGCHLD** signal back to the handler. On the right, a vertical rectangle represents the **sigUsr2Handler**. A signal source **S** (circle) sends a signal to this handler. The handler sends a **sigaction(...)** signal to the process, which then sends a **SIGUSR2** signal back to the handler. Below the process, a circle labeled **RW** (Read-Write lock) is connected to a stack of horizontal rectangles labeled **Places de parking** (Parking spaces). The process also has a **shmat(...)** connection to the **RW** lock.

## C . Destruction

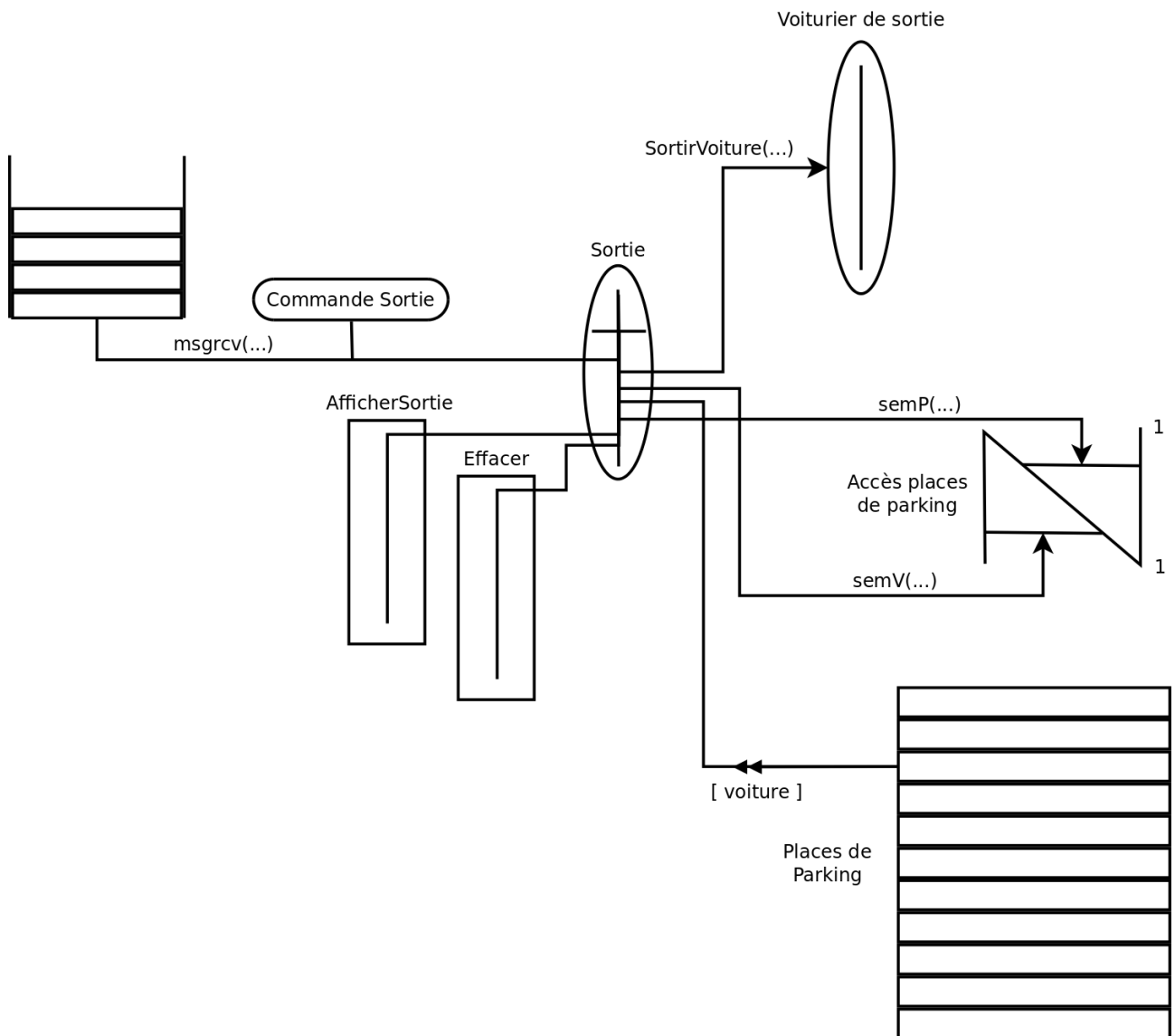


## 4. Tâche Sortie

### A. Initialisation



## B. Moteur



## C. Destruction

