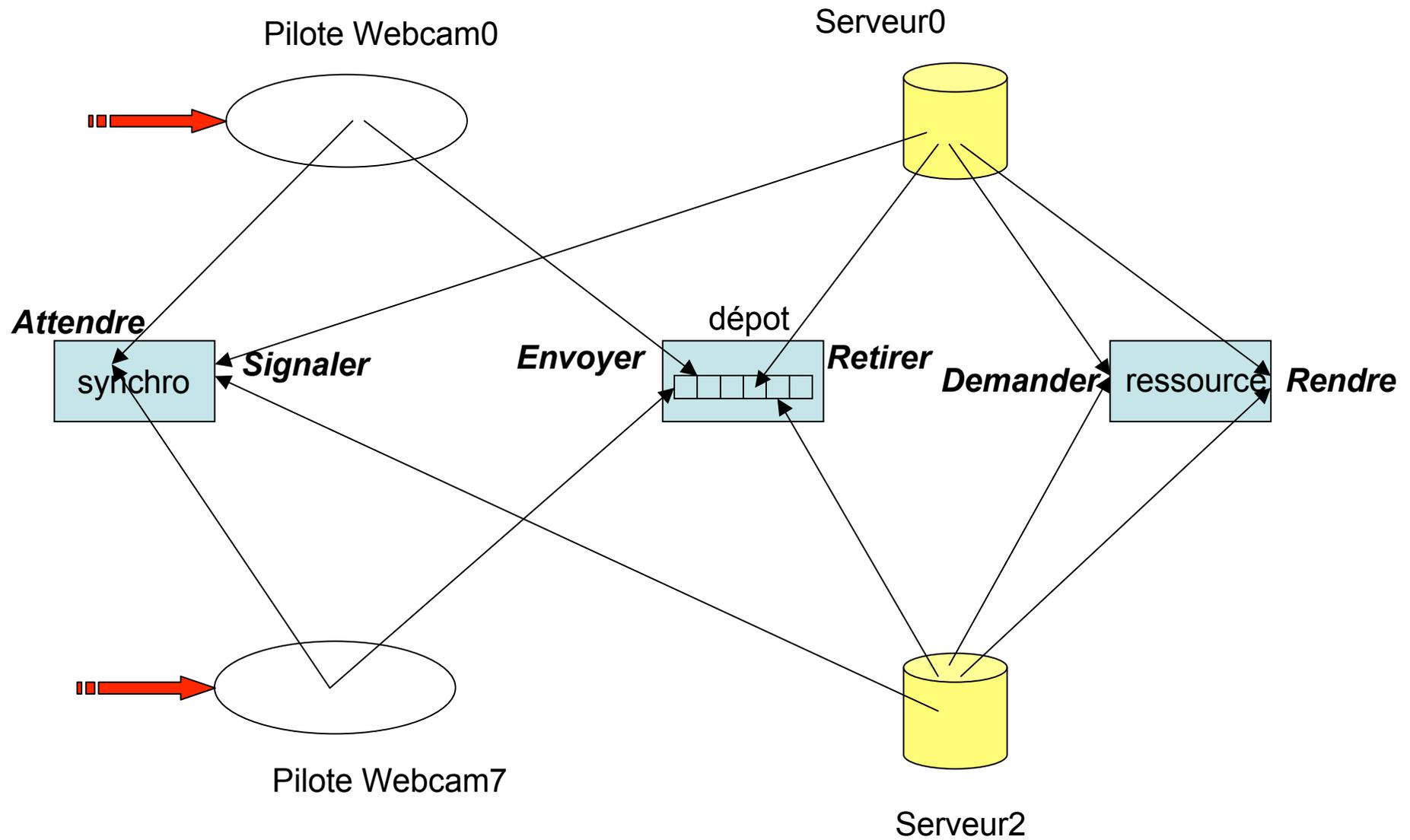


Révisions : étude de cas

Etude de cas 1 : surveillance par Webcam

- Les activités :
 - 8 Pilotes qui gèrent 8 webcam
 - 3 Serveurs qui récupèrent les images : chaque serveur gère 3 webcam.
- La communication :
 - Chaque pilote dépose une image dans un tampon commun
 - Chaque serveur demande à être autorisé à retirer les 3 images du dépôt
 - Le serveur autorisé retire les images.



- La synchronisation :

- Le pilote attend qu'un serveur demande des images
- Un serveur demande d'accéder à 3 webcam
- Lorsque l'accès lui est permis, il signale à chaque pilote qu'il est prêt à retirer l'image associée
- Chaque pilote acquiert une image et envoie l'image dans le dépôt à destination du serveur
- Le serveur retire les 3 images
- Le serveur libère l'accès à la ressource

Passage de témoin

Producteurs-
Consommateurs

Exclusion mutuelle

Exclusion mutuelle

Algorithme

Un pilote

```
X : nom unique de la webcam; //entier 0..7
Y : image; Z : Bitmap (image reçue par un pilote)
tant que vrai faire
// attente d'un signal envoyé à X par un serveur
  synchro.Attendre (X) ;
// acquisition de l'image bitmap
  Lecture_de_la_Camera (Z) ;
  Préparation de l'image Y avant dépôt;
// envoi vers les serveurs de l'image Y prise par la
Webcam X
  depot.Envoyer (X,Y)
fait
```

Un Serveur

Algorithme

```
I : nom unique du serveur;//entier 0..2
A,B,C : nom webcam;//entier 0..7
Y,Z,T : image;
Tant que vrai faire
  // le client donne 3 noms de Webcam
  negociation_avec_le_client (A,B,C) ;
  // réservation des Webcams
  ressource.Demander(A) ; ressource.Demander(B) ;
  ressource.Demander(C) ;
  // réveil des pilotes
  synchro.Signaler(A) ; synchro.Signaler(B) ; synchro.Signaler(C) ,
  depot.Retirer(A,Y) ; // retrait de l'image de A et copie dans Y
  depot.Retirer(B,Z) ; // retrait de l'image de B et copie dans Z
  depot.Retirer(C,T) ; // retrait de l'image de C et copie dans T
  envoi_images_au_client(Y,Z,T);
// libération des Webcams
  ressource.Rendre(A) ; ressource.Rendre(B) ; ressource.Rendre(C) ,
fait
```

Les éléments de coopération

Synchro //synchronisation pilotes-serveurs

```
procedure Attendre(X : nom webcam) ;
```

```
// attente par le pilote X du signal d'un serveur
```

```
procedure Signaler(X : nom webcam) ;
```

```
// réveil d'un pilote X par un serveur
```

Ressource //gère l'allocation des webcams

```
procedure Demander(X : nom webcam) ;
```

```
// Acquisition de la webcam X
```

```
procedure Rendre(X : nom webcam) ;
```

```
// Restitution de la webcam X
```

Depot //coopération pilotes-serveurs

```
procedure Envoyer(X:nom webcam; Y:image) ;
```

```
// Depot dans un tampon de l'image Y par un pilote X
```

```
procedure Retirer(X:nom webcam; Y:image) ;
```

```
// Retrait du tampon de l'image Y déposée par X
```