

# AppsGate

## Scénario de dispositif UPnP

---

Ce scénario permet d'illustrer le suivi du déplacement de l'utilisateur au moyen de capteurs de présence. Ici l'utilisateur est en train de regarder un film sur une TV et le système va tenter de faire suivre la lecture du film en fonction de son changement de pièce.

L'utilisateur est seul dans la maison au moment du scénario. Les dispositifs utilisés sont la TV du salon, la TV de la cuisine, chacune présentée comme un media-renderer UPnP, un serveur de fichiers, présenté comme un media-server UPnP, et deux capteurs de présence EnOcean situés dans la cuisine et dans le salon. On fait l'hypothèse que ces dispositifs ont été préalablement installés et fonctionnent correctement.

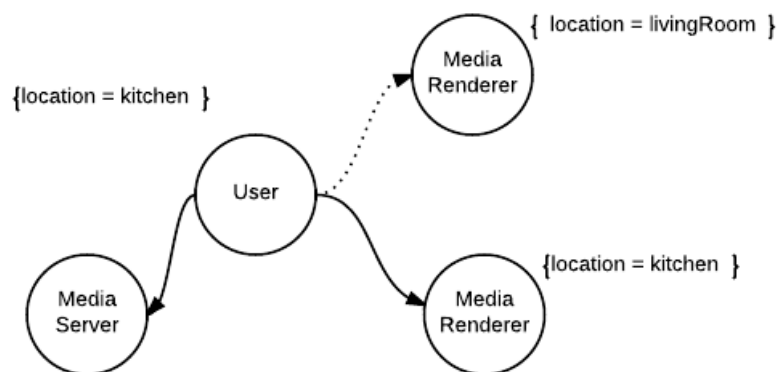
### 1. Scénario nominal

1. L'utilisateur lance l'IHM pour sélectionner le film, sur le serveur de fichiers, qui est à lire sur la TV du salon.
  - (a) Le système récupère l'URI du film sur le media-server UPnP (serveur de fichiers)
  - (b) Le système envoie l'URI du fichier au media-renderer choisi (TV du salon)
  - (c) Le système envoie la commande Play au media-renderer choisi, ici la TV du salon
  - (d) La TV du salon notifie la Gateway de l'habitat de son changement d'état (stop → Play).
  - (e) La Gateway notifie l'utilisateur que la lecture est en cours sur la TV du salon.
2. L'utilisateur quitte le salon.
  - (a) Le capteur du salon notifie la Gateway qu'il n'y a plus personne dans le salon
  - (b) La Gateway met en pause le film sur le media-renderer en lecture (TV du salon)
  - (c) La TV du salon notifie la Gateway de son changement d'état (Play → pause).
  - (d) La Gateway notifie l'utilisateur que la TV du salon est en pause
3. L'utilisateur entre dans la cuisine.
  - (a) Le capteur de la cuisine notifie la Gateway qu'il y a quelqu'un dans la cuisine
  - (b) La Gateway cherche un dispositif de rendu dans la cuisine
  - (c) La Gateway vérifie l'état du dispositif de rendu. Celui-ci apparaît prêt
  - (d) La Gateway récupère l'état de la lecture du film (temps, vitesse, etc.)
  - (e) La Gateway envoie l'URI du fichier au nouveau media-renderer (TV de la cuisine)
  - (f) La Gateway met à jour les paramètres de lecture (temps, vitesse, etc.)
  - (g) La Gateway envoie la commande Play au nouveau media-renderer (TV de la cuisine)

- (h) La TV de la cuisine notifie la Gateway de son changement d'état (Stop → Play)
- (i) La Gateway notifie l'utilisateur que la TV de la cuisine est en mode lecture
- (j) La Gateway envoie la commande stop à l'ancien media-renderer (TV du salon)
- (k) La Gateway notifie l'utilisateur que la lecture sur la TV du salon est stoppée

### Solution APAM :

De ce qui suit on représente l'utilisateur, le renderer et serveur par des instances ApAM. On a alors deux moyens pour implanter le scénario :



**Hypothèse :** L'instance User, va contenir l'information du choix du film, elle jouera le rôle de « control point ». Cette instance décrit un attribut location qui est mis à jour dynamiquement selon la position de l'utilisateur. Les « media-renderer » disposent eux aussi d'un attribut location.

Dans cette architecture, l'instance utilisateur devra récupérer la position de la lecture encours au moment du changement de pièce, pour la transmettre au nouveau media-renderer.

### Avec Manager : (reconfiguration manuel, ad hoc)

On peut utiliser un ApAM Manager qui sera à la fois un **PropertyManager** et un **DependencyManager**. L'interface des **PropertyManager**, va permettre de réagir à la modification des propriétés d'une instance (ici location) :

```

public interface PropertyManager {
    /**
     * The attribute "attr" has been modified.
     * @param component. The component (Spec, Implem, Instance) holding that attribute.
     * @param attr. The attribute name.
     * @param newValue. The new value of that attribute.
     * @param oldValue. The previous value of that attribute.
     */
    public void attributeChanged (Component component, String attr, String newValue,
    String oldValue) ;

    /**
     * The attribute "attr" has been removed.
     * @param component. The component (Spec, Implem, Instance) holding that attribute.
     * @param attr. The attribute name.
     * @param oldValue. The previous value of that attribute.
     */
    public void attributeRemoved (Component component, String attr, String oldValue) ;
  
```

```

/**
 * The attribute "attr" has been added (instantiated for the first time).
 * @param component. The component (Spec, Implem, Instance) holding that attribute.
 * @param attr. The attribute name.
 * @param newValue. The new value of that attribute.
 */
public void attributeAdded (Component component, String attr, String newValue) ;
}

```

On capturera alors les modifications de l'attribut de l'instance « user », et on pourra supprimer le wire vers l'instance media-renderer grâce à l'API :

```

public interface Instance extends Component {
...
/**
 * returns all the wire from the current instance
 *
 * @return
 */
public Set<Wire> getWires();
/**
 * remove that wire.
 *
 * @param wire
 */
public void removeWire(Wire wire);
...
}

```

Le manager pourra garder la localisation de l'utilisateur dans un attribut à lui, par exemple :

```

...
private String userLocation;
...

userLocation = instance.getProperty("location");
...

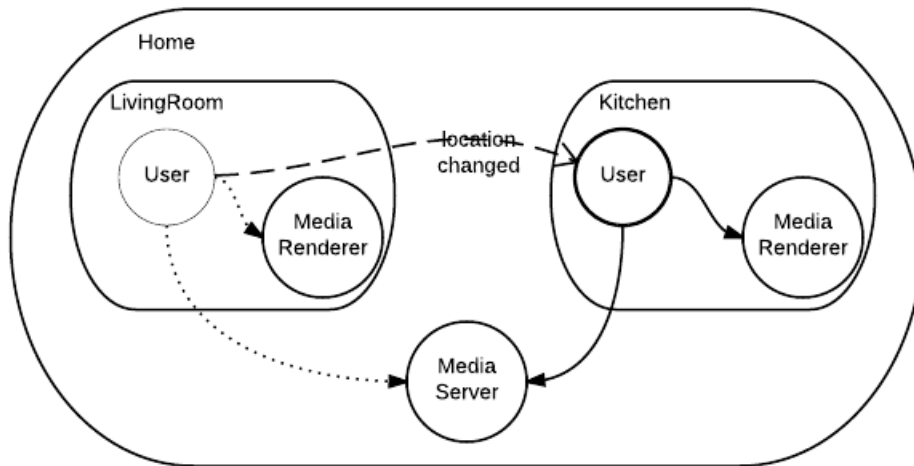
```

La deuxième interface est de type **DependencyManager**, qui va permettre au manager de préciser la sélection de la nouvelle instance du « media-renderer ». Il pourra alors ajouter une contrainte de sélection sur **location=userLocation** qui correspond au nouvel emplacement de l'utilisateur. ApAM sélectionnera alors la bonne instance media-renderer.

### Avec Composite : (reconfiguration automatique)

En utilisant les composites on peut avoir une reconfiguration automatique de l'architecture. Pour cela, on déclare deux composites ApAM qui représenteront les pièces LivingRoom et Kitchen et qui contiendront les media renderer, et l'utilisateur quand il est présent dans cette pièce.

La figure suivant décrit cette architecture d'une manière informelle :



Dans ce cas, il suffit de déclarer les deux composites comme suit :

1) on indique que chaque pièce « possède » les media renderers de cette pièce, et l'utilisateur quand il est présent dans cette pièce.

```

<composite name="Kitchen-Composite">
  <contentMngt>
    <local instance="(&(specification=UpnpMedia-Render)(Location=kitchen))"/>
    <own specification="UpnpMedia-Render" property="Location" value="kitchen" />
    <own specification="User" property="Location" value="kitchen" />
  </contentMngt>
</composite>

```

```

<composite name="Living-Composite">
  <contentMngt>
    <local instance="(&(specification=UpnpMedia-Render)(Location=Living))"/>
    <own specification="UpnpMedia-Render" property="Location" value="kitchen" />
    <own specification="User" property="Location" value="Living" />
  </contentMngt>
</composite>

```

Dans ce cas les wires entre l'instance « User » et « Media Renderer » vont se faire et se défaire en fonction de la propriété location. Selon comment L'instance « User » est programmé, il pourra transmettre l'information nécessaire au media-renderer.

**Dans cette solution, Le seul code qui doit être écrit et le code métier ! Ce code fera le transfert des états de lecture entre les deux pièces.**

**Le transfert de l'état peut être réalisé par des callback sur le changement de la propriété location, ou par des callback d'apparition et disparition de renderer.**

Il y a d'autre façon d'implémenter le scenario, tout dépend de comment le « control point » est programmé et comment il interagit avec l'instance utilisateur. On peut aussi envisager des scénarios multi-utilisateurs.

Dans cette solution on a supposé que l'instance du « control point » et « User » sont les mêmes. Mais on peut également les découpler et/ou les mettre dans un composite « User Composite » par exemple.

## 2. Variante 1 : la TV de la cuisine est éteinte ou en panne

Ce scénario présente le cas d'exception où la TV de la cuisine est indisponible pour cause de problèmes techniques (pannes, problèmes réseaux, etc.).

1. Idem jusqu'en 3b
2. La Gateway ne trouve aucun media-renderer dans la cuisine
3. La Gateway notifie l'utilisateur que l'accès à la TV de la cuisine est impossible et lui précise que s'il l'allume alors la lecture de son film se poursuivra dans la cuisine.

Dans le cas où l'utilisateur allume la TV de la cuisine, laquelle fonctionne dorénavant normalement, le scénario reprend de façon normale à partir du point 3c.

### Solution APAM :

APAM : rien à ajouter à part la notification ; dès que la TV est allumée, elle est prise (comportement par défaut).

## 3. Variante 2 : la TV de la cuisine est déjà occupée

Ce scénario présente le cas d'exception où la TV de la cuisine est déjà utilisée.

1. Idem jusqu'en 3b
2. La Gateway vérifie l'état du dispositif de rendu qui indique busy
3. La Gateway vérifie s'il existe une politique de priorité à appliquer
4. La Gateway applique la politique trouvée ou interroge l'utilisateur pour prendre une décision

Une fois que la politique à appliquer est déterminée, le scénario reprend son exécution normale à partir du point 3c.

### Solution APAM :

APAM : rien à ajouter. Les TV sont déclarées « shared="false" instantiable="false" ». On peut ici faire des scénarios plus sophistiqués pour déterminer qui peut utiliser la TV, et dans quelle circonstance (grant). **Là aussi, pas de programmation** (peut-être)

```
<own specification="User" property="Location" value="Living" >
  <grant when="dad" implementation="dadImpl" dependency="renderer" />
</own>
```

Indique que c'est le père (macho) qui a priorité sur la TV dans cette pièce (le salon). Ce peut être différent dans la chambre de sa fille par exemple !