Déroulement d’une attaque sur un système

# Introduction

Ce document présente le déroulement d’une attaque sur un système. L’attaque se déroule en deux étapes :

* **Reconnaissance du système,** lors de cette étape l’attaquant tente d’obtenir le plus d’information possible sur le système sans interagir avec ce dernier (observation, piratage de boîtes mail, …)
* **Découverte du système,** lors de cette phase l’attaquant interagit avec le système dans le but d’en comprendre l’architecture interne.

Table des matières

[Introduction 1](#_Toc434840262)

[Cas d’étude 2](#_Toc434840263)

[Reconnaissance du système 3](#_Toc434840264)

[Regroupement des informations 3](#_Toc434840265)

[Simulation du modèle du système supposé 5](#_Toc434840266)

[Conclusion 7](#_Toc434840267)

[Découverte du système 8](#_Toc434840268)

# Cas d’étude

Nous baserons notre exemple sur un système simple de détection de forme. Le système est composé d’une caméra relié à un FPGA lui-même relié à un processeur ARM et un disque dure SSD. Le système dispose d’une connexion internet via le processeur ARM.

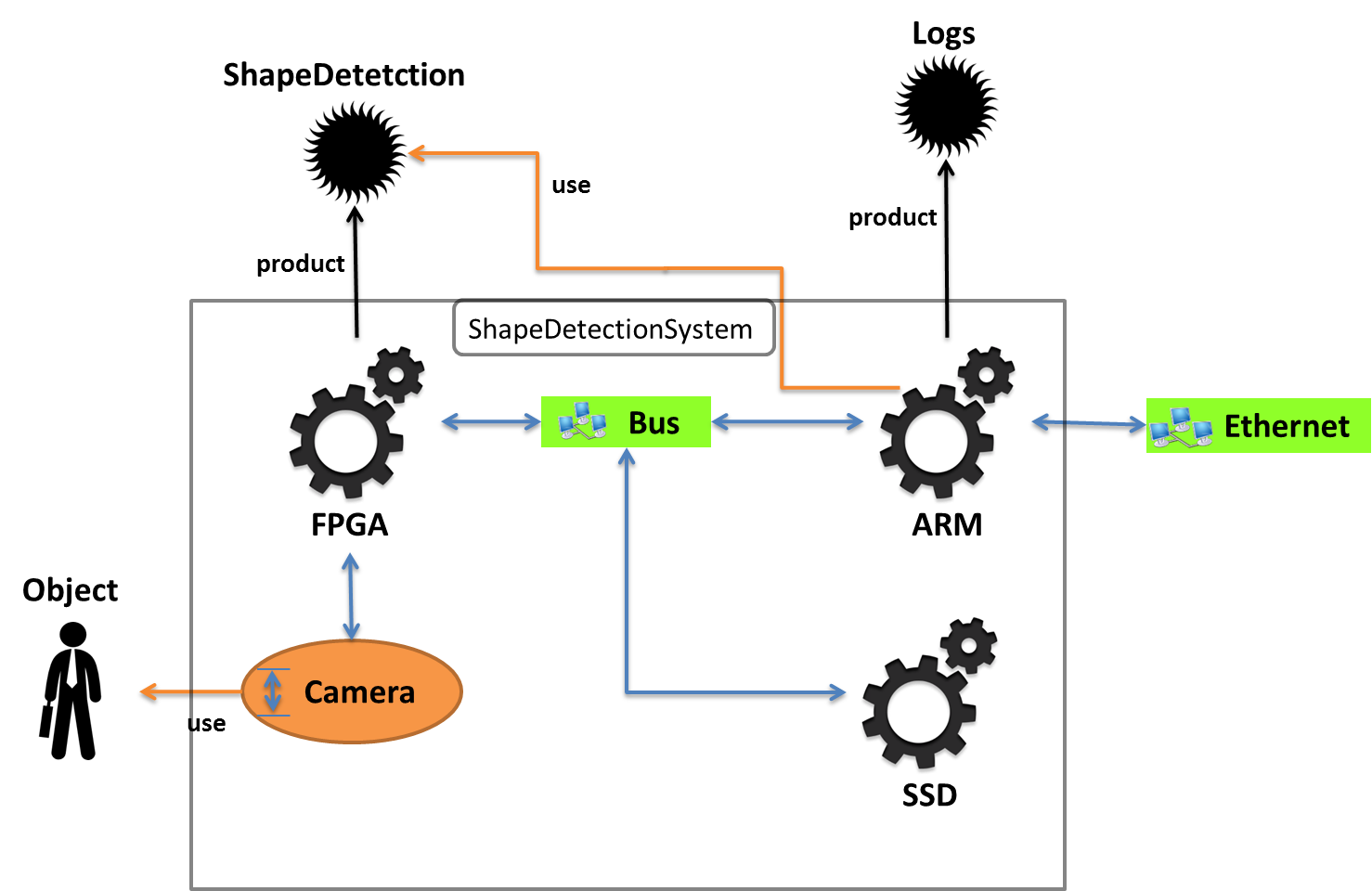


Figure 1: Schémas du système réel dans son ensemble

Le figure ci-dessus représente le système réel dans son ensemble, c’est cette information que l’attaquant cherche à obtenir lors de la 1er phase de l’attaque (Reconnaissance du système).

# Reconnaissance du système

## Regroupement des informations

Nous supposerons pour notre exemple que l’attaquant possède :

* **Une photo du système** : Cette photo a pu être prise par un drone ou grâce à un téléphone. Elle est de qualité moyenne mais permet tout de même d’obtenir quelques informations sur le système.
* **Des informations concernant la consommation énergétique du système :** Ces informations peuvent être déduites d’une image du système sur laquelle les sources énergétiques sont visibles ou via un autre moyen tel que le piratage d’une boîte mail.



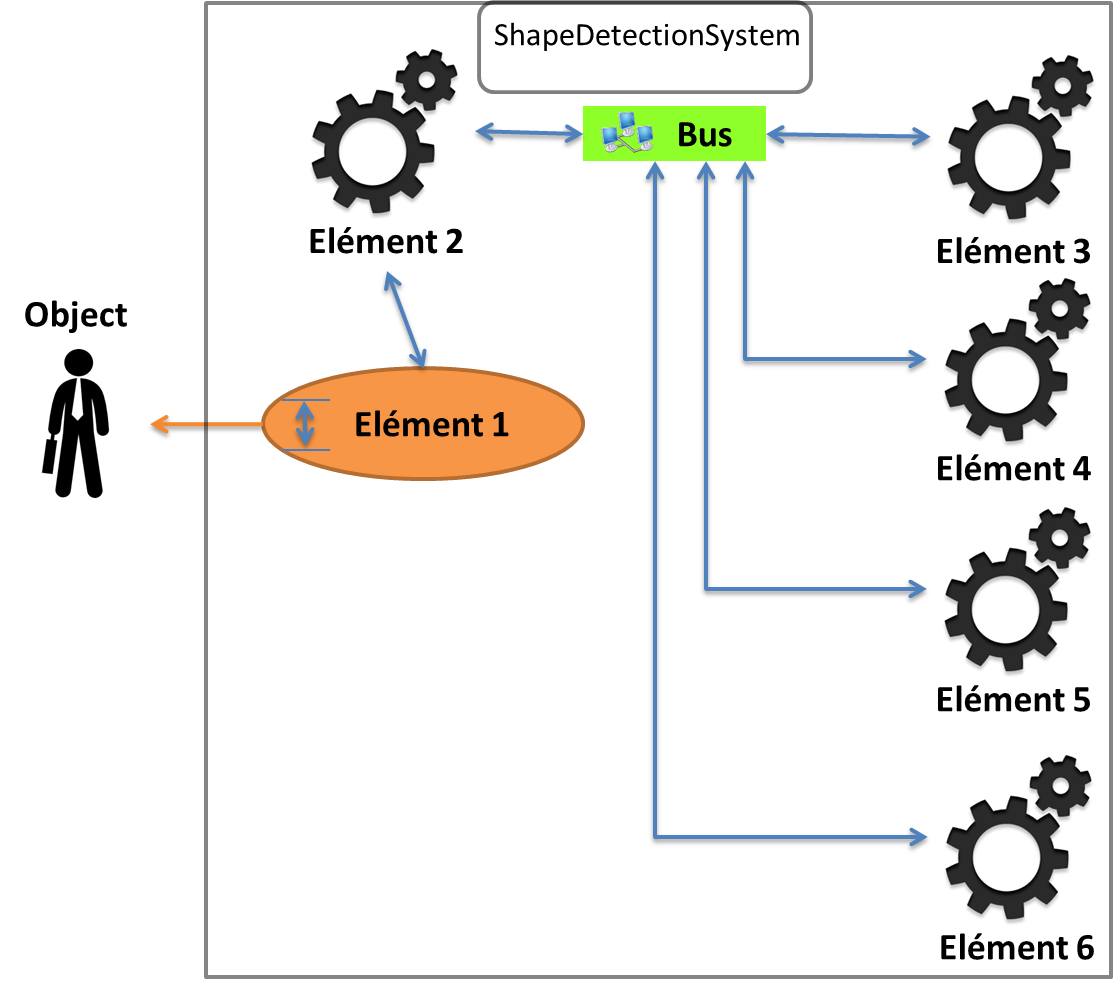
Figure 2: Photo du système

Sur la photo si dessus, prise par l’attaquant, il est possible de reconnaitre une caméra branché à un FPGA lui-même relié à plusieurs autres composants. Le système global semble relié au réseau via un port Ethernet. Les observations faites grâce à la photo permettent à l’attaquant de créer le modèle supposé du système via Pimca. Pour créer ce modèle l’attaquant doit faire des hypothèses sur le système. Dans un premier temps il suppose que le système possède 6 entités distinctes : une caméra, un FPGA, un processeur I7, un disque dure HDD, de la mémoire RAM et un GPU.

**Hypothèse 1 :**

* Camera
* FPGA
* I7
* HDD
* RAM
* GPU

Dans un premier temps l’attaquant va donc créer un modèle de l’architecture supposé via Pimca. Son modèle (suivant les hypothèses 1) sera donc le suivant :

Figure 3: Modèle du système supposé créait grâce aux hypothèses 1

Le modèle ainsi défini représente l’architecture supposé du système et défini des liens entre les différents éléments. L’attaquant peut alors spécifier ces éléments en y associant des comportements. La notion de comportement regroupe les informations relatives à la taille de l’élément, sa consommation mais aussi son temps de réponse, etc. La description de tel comportement est rendu possible grâce à Morphose. Le lien entre un élément dessiné sur Pimca et des caractéristique définies sous Morphose ce fait grâce à la notion rôle et donc grâce à Role4All.

Le schéma ci-dessous montre un exemple d’association entre un élément Pimca et un comportement défini sous Morphose. Il est a noté qu’un élément ou un comportement peut posséder plusieurs rôles.

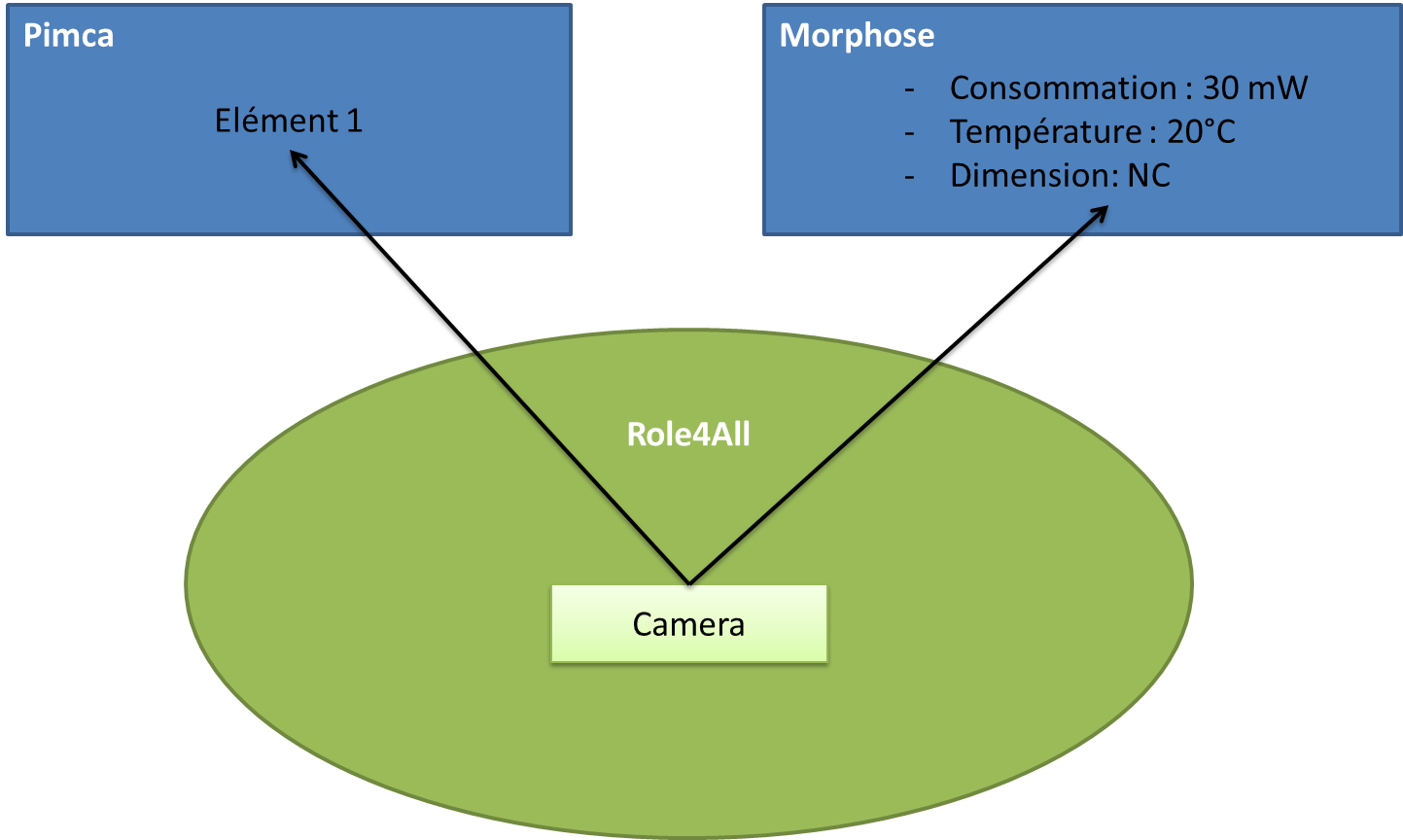


Figure 4:Mise en place d’une notion de rôle sur deux éléments de deux outils différents

On peut alors donner un rôle à tous les éléments de notre architecture et associé un comportement (ou plusieurs) à chacun de ces rôles. Ainsi l’élément 3 prend le rôle de processeur I7, ce rôle est relié à deux comportement celui de processeur et celui de composant Intel, cela permet de spécifié au mieux son comportement. Une fois toutes les spécifications comportementales effectuer le système supposé est :

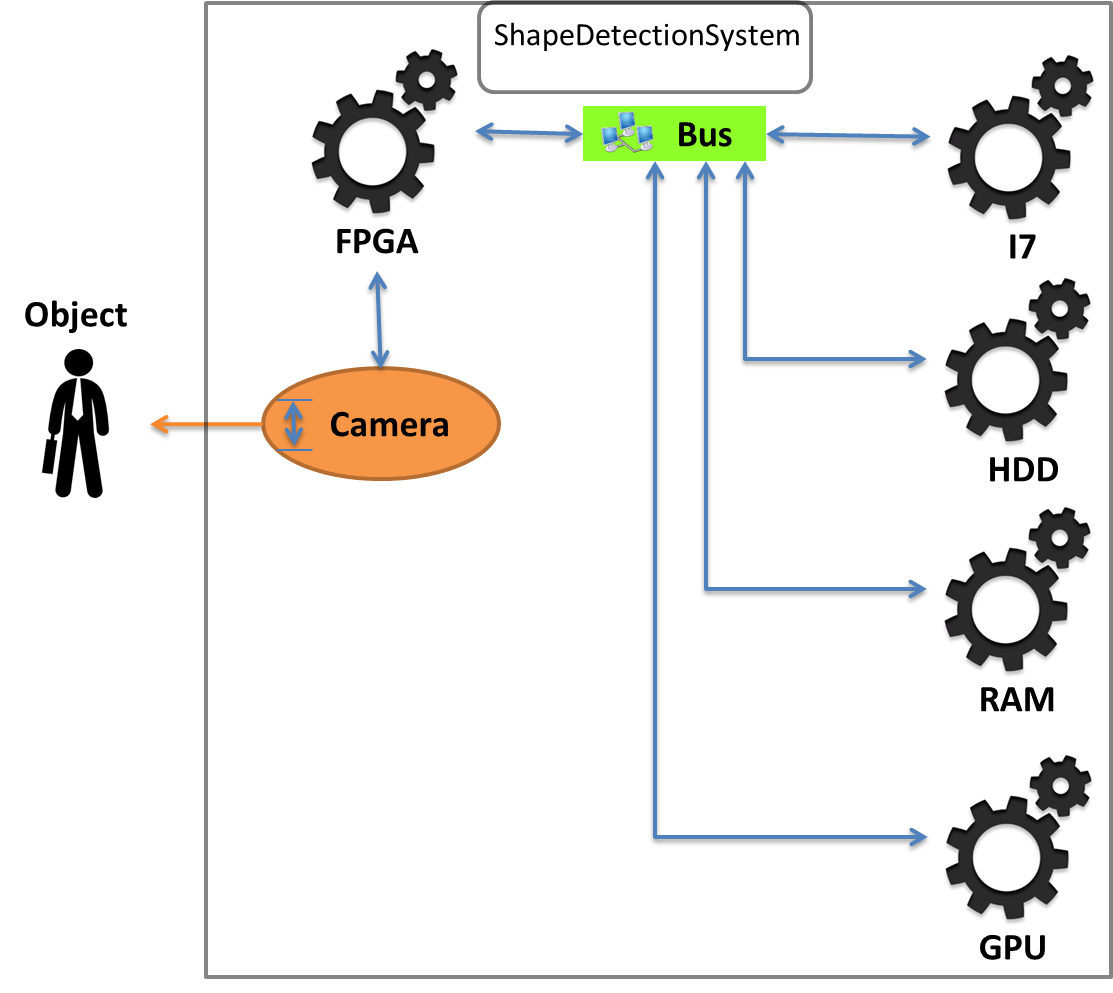


Figure 5: Modèle du système supposé créait grâce à l’architecture défini sur Pimca et aux spécifications comportementales définies sous Morphose

En plus de la photo du système l’attaquant a réussi à intercepter un échange d’email entre plusieurs acteurs du projet :

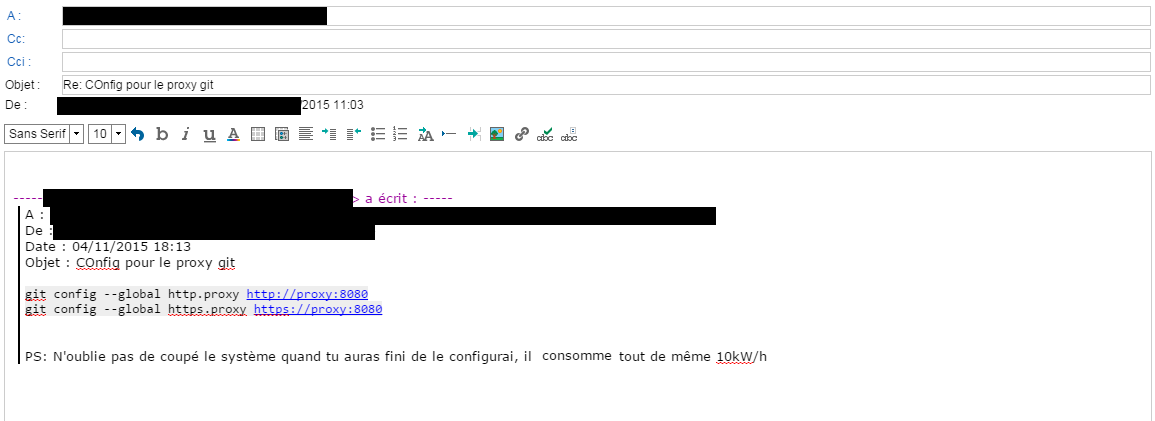


Figure 6: Email provenant d'une boite mail piraté

L’Email présenté dans la figure ci-dessus permet à l’attaquant d’obtenir une connaissance approximative de la consommation du système (10kW/h).  L’attaquant peut alors ajouter cette information à un fichier Excel que l’on nommera *informationsComplementaires.xlsx.*

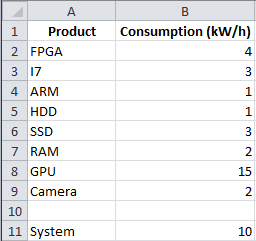


Figure 7 : Consommations de divers produits (chiffre non réaliste)

## Simulation du modèle du système supposé

Une fois que l’attaquant a regroupé des informations sur le système et qu’il les a misent en forme grâce à plusieurs outils (Pimca et Excel), il peut tester les hypothèses qu’il a formulé. Pour ce faire l’attaquant utilise Role4All ce qui lui permet de synchronosé le modèle du système supposé, fait sur Pimca, et les informations complémentaires contenus dans le fichier Excel. Une fois ces données reliées et synchrone le système supposé est virtualisé et testé grâce à Morphose. Morphose effectue alors plusieurs mesure (consommation, température, …) sur le système virtuel et les comparées aux mesures effectuées sur le système réel.

On a alors deux phases distinctes mais complémentaire une de synchronisation des outils et une de simulation du système.

### Synchronisation

Cette étape a pour but d’harmoniser les informations redondantes contenues dans les divers outils. Pour l’exemple on peut prendre ici les noms des éléments du système (FPGA, I7, …) qui est une information contenu à la fois dans le modèle Pimca et dans le fichier Excel. Conformément aux règles d’utilisations de Role4All chaque élément, qui soit défini sous Pimca ou Excel, joue un rôle. Ce rôles est défini dans le model de rôles crée via Role4All. Pour notre exemple nous prendrons en compte seulement un rôle, celui de FPGA. Le model de rôle serais donc le suivant :

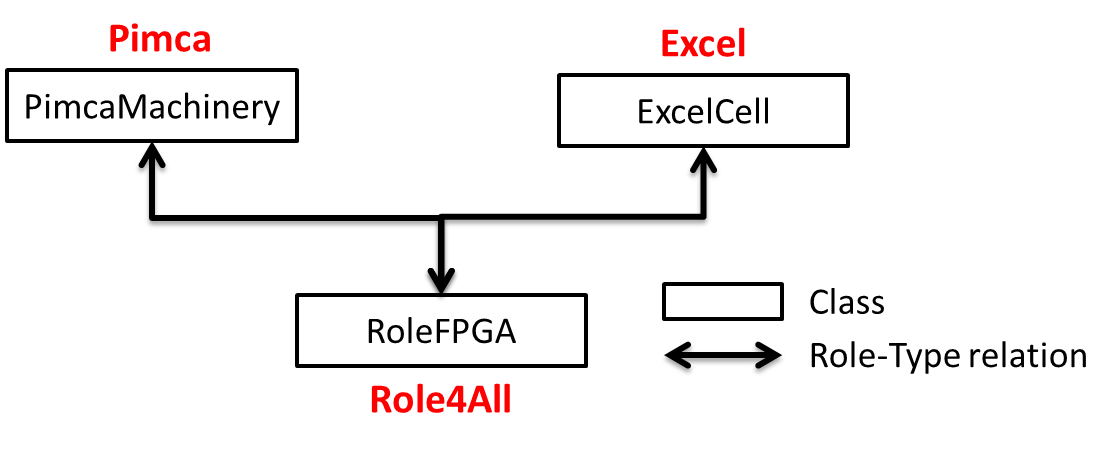


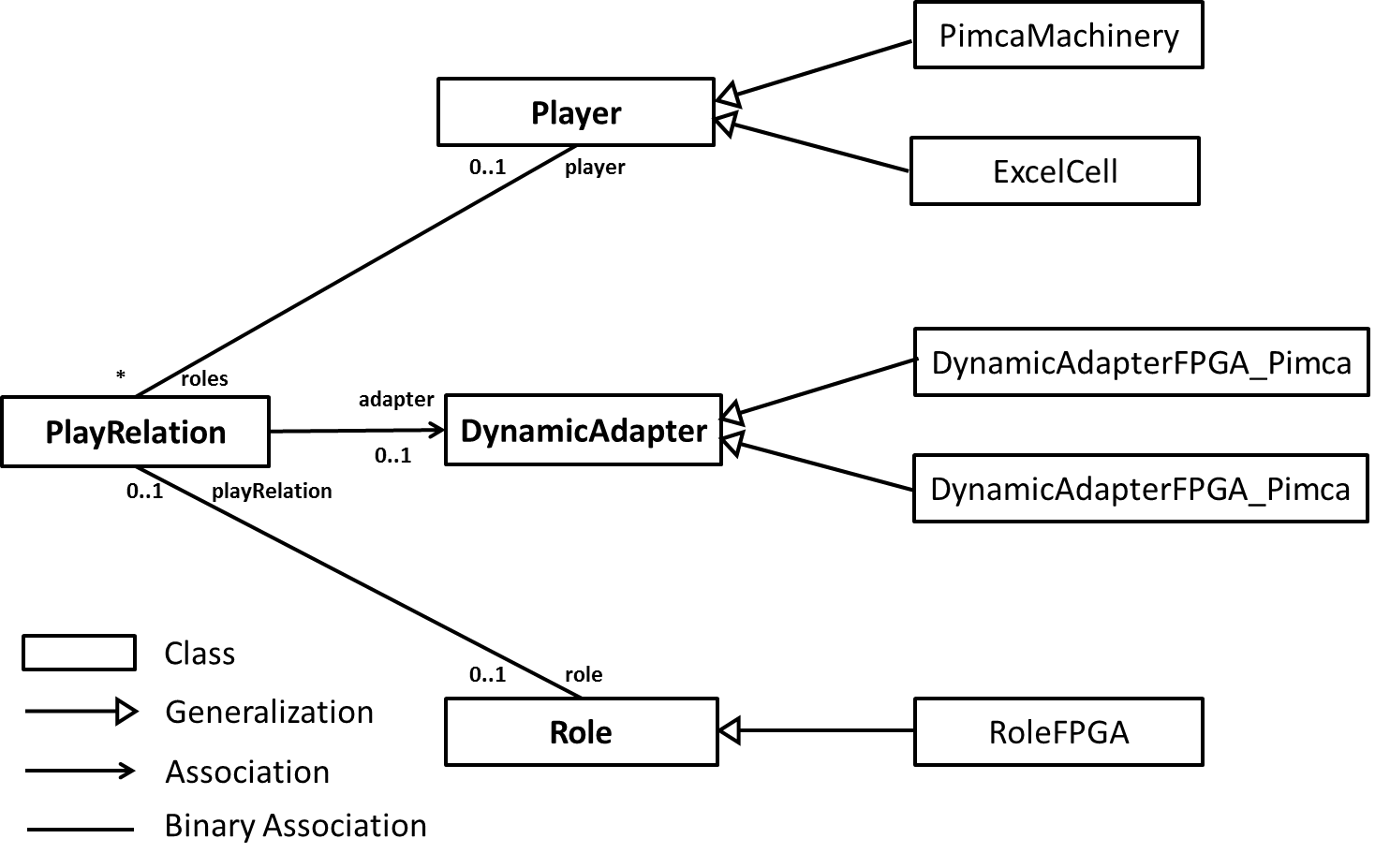
Figure 8 : Role model of FPGA role.

La figure 8 présente le model de rôle associé au rôle de FPGA. On peut y voir que les éléments de Pimca de type PimcaMachinery et les éléments de Excel de type ExcelCell sont capables de jouer le même rôle, celui de FPGA. Il faut bien noter que le rôle de FPGA est indépendant de l’outil.

La relation entre les éléments de model et les types de rôles (Role-Player relation) se fait via un connecteur. Ce connecteur est un élément de Role4All de type PlayRelation.

La présence de cet élément permet de dissocier les aspects de connexions entre rôles et éléments de modèles, de la définition du rôle. Un élément de type PlayRelation est un simple connecteur entre trois classes : Player, Role et DynamicAdapter, cette dernière permet de spécifier la relation entre un player et un rôle. L’adaptateur est donc dissocié du connecteur ce qui permet de modifié les spécificités d’une relation Role-Player sans avoir à redéfinir la relation elle-même. On a 3 élément indépendant : le rôle, le connecteur et l’adaptateur ce qui permet de spécifié une relation Role-Player sans interagir sur le rôle ou le player. La relation entre Player et Role est alors dynamique, car adaptable indépendamment du rôle et du player.

La figure 9 détaile la relation Role-Player :



La relation entre les éléments de model et les types de rôles a, dans Role4All, une incidence au niveau des instances.

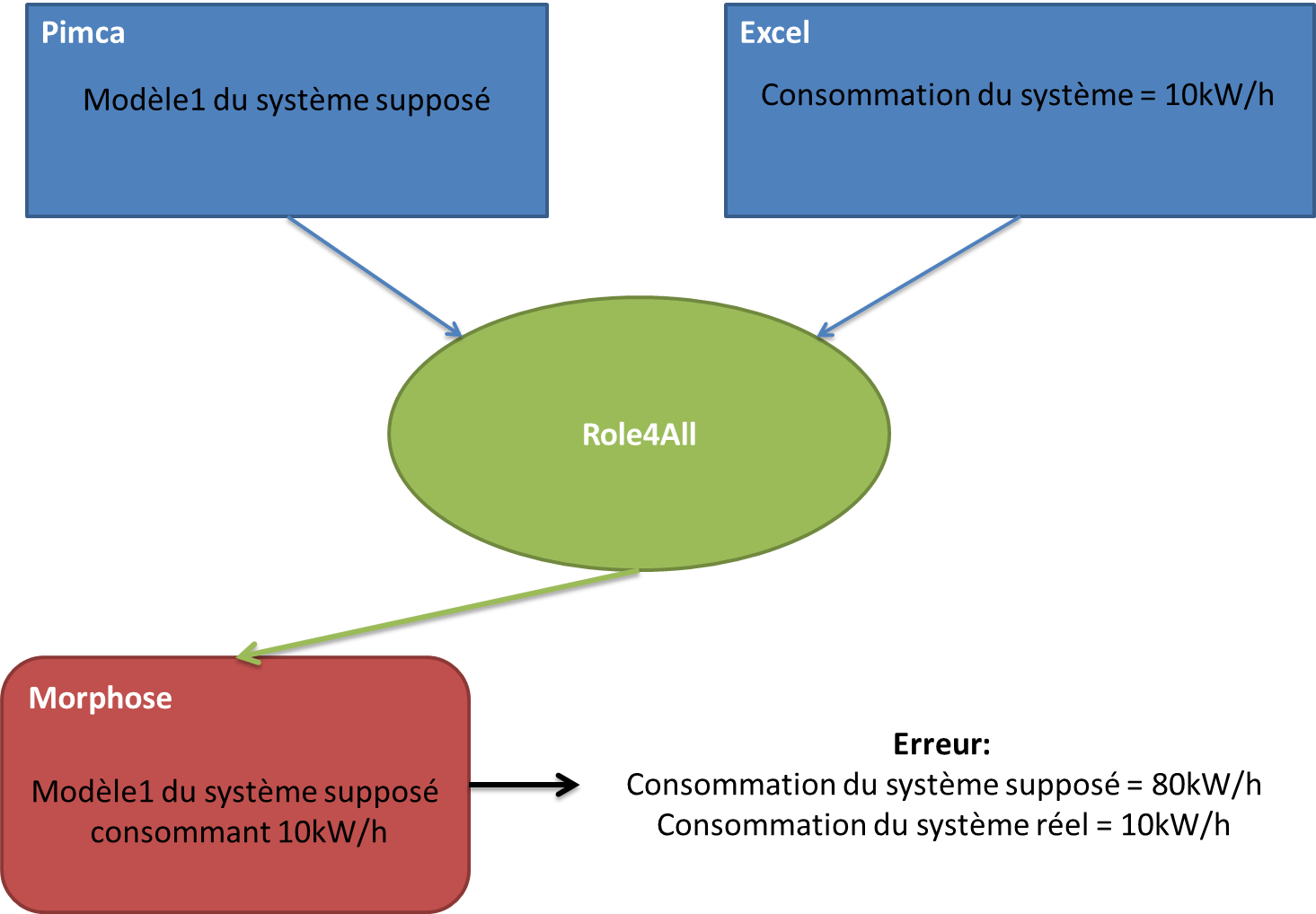


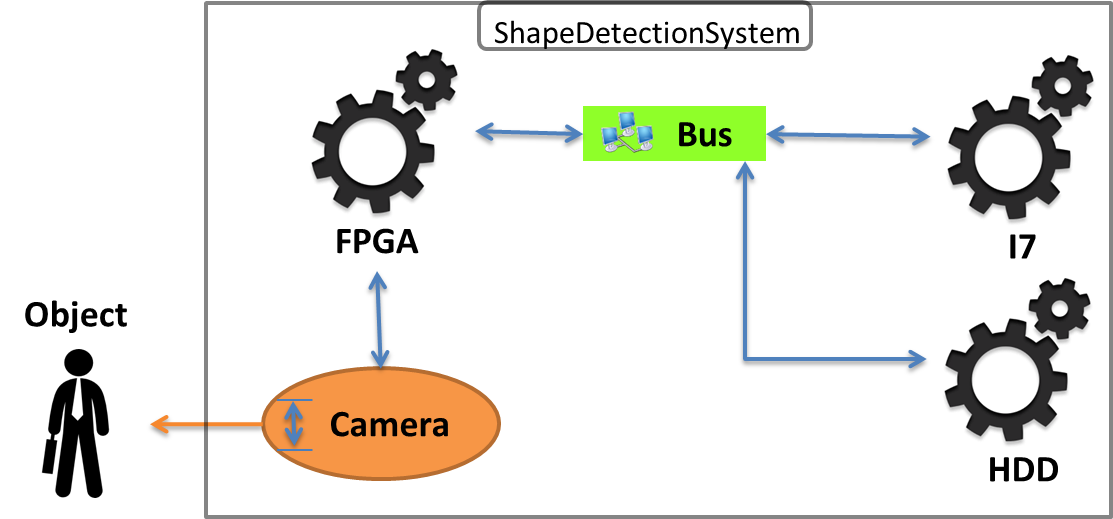
Figure 9: Teste des hypothèses 1

La figure ci-dessus schématise une phase de validation d’hypothèse. Le résultat des tests effectués nous permet de conclure que le modèle du système supposé ne représente pas le système réel. En comparant les consommations réelles et simulées on peut déduire que le modèle du système supposé est trop gourmand en énergie. L’attaquant peut alors modifier ses hypothèses en conséquence. Dans ce cas précis il décide de suppriment le GPU et la RAM.

**Hypothèse 2 :**

* I7
* HDD

Pour effectuer ses modifications l’attaquant doit simplement supprimé les rôles qui ne lui sont pas utile.

Figure 10: Modèle du système supposé créait grâce aux hypothèses 2

L’attaquant peut alors simuler son nouveau modèle en suivant le même cheminement que pour le modèle précédent.

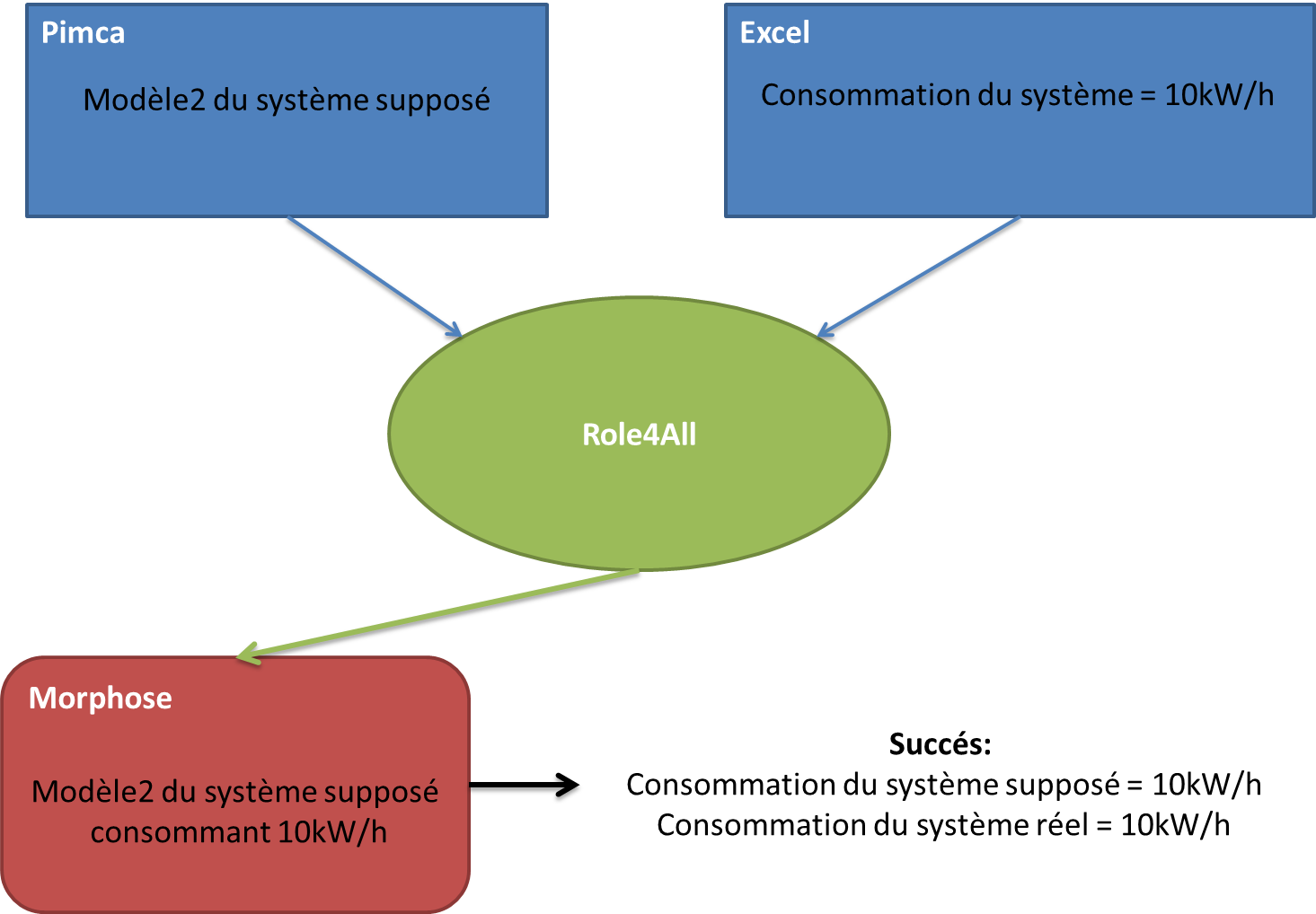


Figure 11: Teste des hypothèses 2

Le 2ème modèle proposé par l’attaquant semble cohérant. Avec les informations que possède l’attaquant (photos et consommations) il ne lui est pas possible de plus raffiné son modèle.

L’attaquant doit alors rechercher de nouvelles informations sur le système (temps de réponse, temps d’allumage, dégagement thermique, …). Ces informations seront alors formatées (Excel, Pimca ou autre) et injectées dans la simulation du système supposé. Ces nouvelles données permettront de raffiner la connaissance que l’attaquant a du système.

Nous supposerons que l’attaquant, grâce à de nouvelles sources d’informations, a réussis à raffiner ça connaissance du système jusqu’à arriver au modèle suivant :

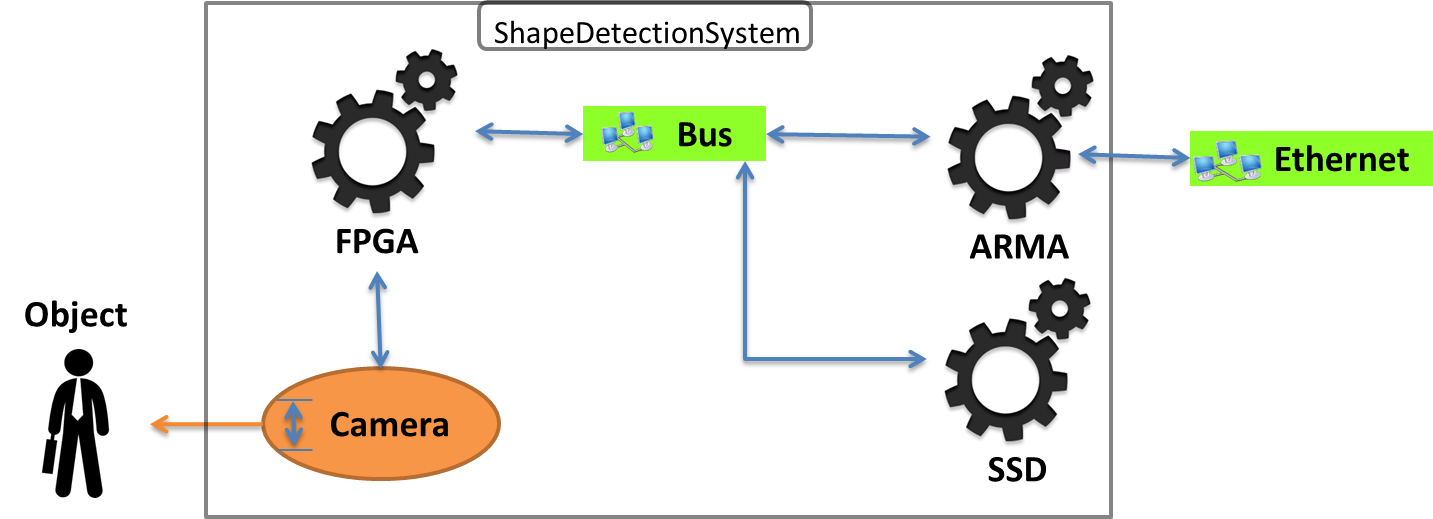


Figure 12: Modèle du système supposé

La figure ci-dessus est semblable à la figure 8, seuls les rôles associés aux éléments ont été modifiés. La possibilité de modifier dynamiquement les rôles permet une plus grande modularité dans la définition d’architecturé système.

## Conclusion

L’attaquant, grâce à un regroupement d’informations périphériques au système, a pu créer un schéma représentant l’architecture externe du système. Ce schéma sera utile lors de la prochaine étape qui consiste en la découverte de l’architecture interne du système.

# Découverte du système

TODO