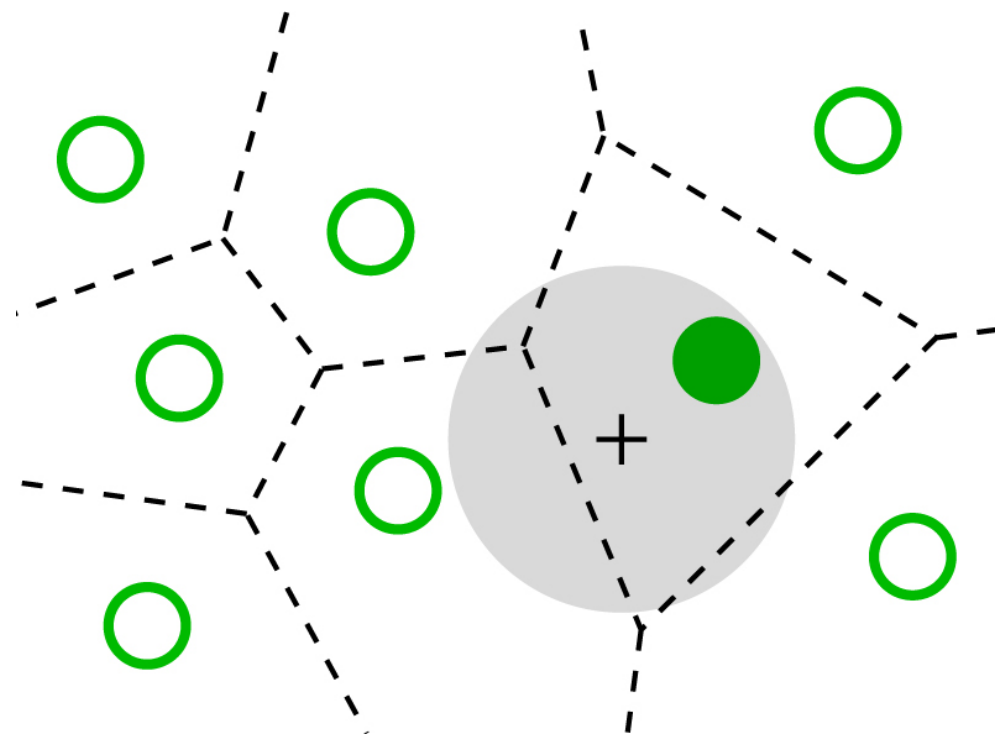


# Application du modèle Entité-Composant-Système à la programmation d'interactions

*Thibault Raffailac, Stéphane Huot*



# Programmation de nouvelles techniques d'interaction



Bubble Cursor (Grossman et Balakrishnan)



ExposeHK (Malacria et al.)

Utilisation de frameworks d'interaction **monolithiques**

The screenshot displays a graphical user interface with a toolbar on the left and a network diagram on the right.

**Toolbar:**

- A button labeled "Label" with a rounded rectangular icon.
- A button with a rectangular icon containing a right-pointing arrow.
- A button labeled "Delete" with a rectangular icon containing an 'X'.
- A button labeled "Quit" with a rectangular icon containing a small square.

**Network Diagram:**

The diagram shows several nodes (rounded rectangles) and their connections (lines):

- Brad A. Myers** (top left)
- Brad Vander Zanden** (top right)
- Ed Berwin** (middle right)
- Roger B. Dannenberg** (middle left)
- Dario Giuse** (bottom right)
- David Kosbar** (bottom left)
- Philippe Marchal** (bottom right, with a selection border)
- John A. Kolosiejchick** (bottom left)

Connections are as follows:

- Brad A. Myers is connected to Ed Berwin and Dario Giuse.
- Brad Vander Zanden is connected to Ed Berwin and Dario Giuse.
- Ed Berwin is connected to Roger B. Dannenberg and David Kosbar.
- Roger B. Dannenberg is connected to David Kosbar.
- Dario Giuse is connected to David Kosbar.

Process  
active | inactive

Property  
value

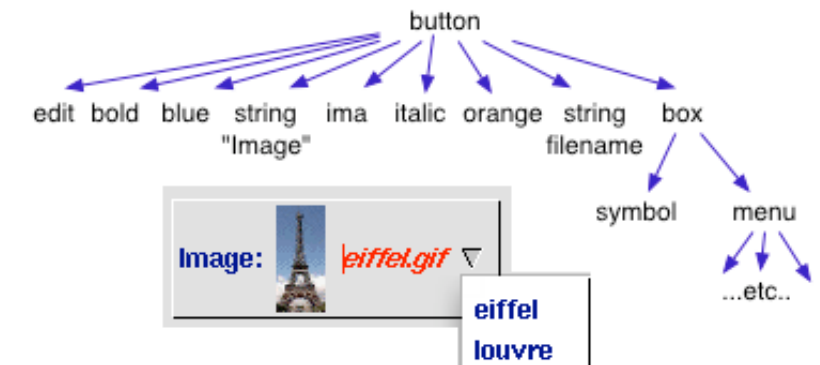
P<sub>1</sub> P<sub>2</sub>

Activates P<sub>2</sub> when P<sub>1</sub> is activated

P<sub>1</sub> P<sub>2</sub>

P<sub>2</sub> = P<sub>1</sub> each time P<sub>1</sub> is set

Djinn (Chatty et. al)



ECS et programmation d'IHM – Raffailac &amp; Huot – 3

# Notre solution

Adaptation du modèle **Entité-Composant-Système** à la programmation d'IHM

Issu du domaine du Jeu Vidéo

Défini par Adam Martin (2007)



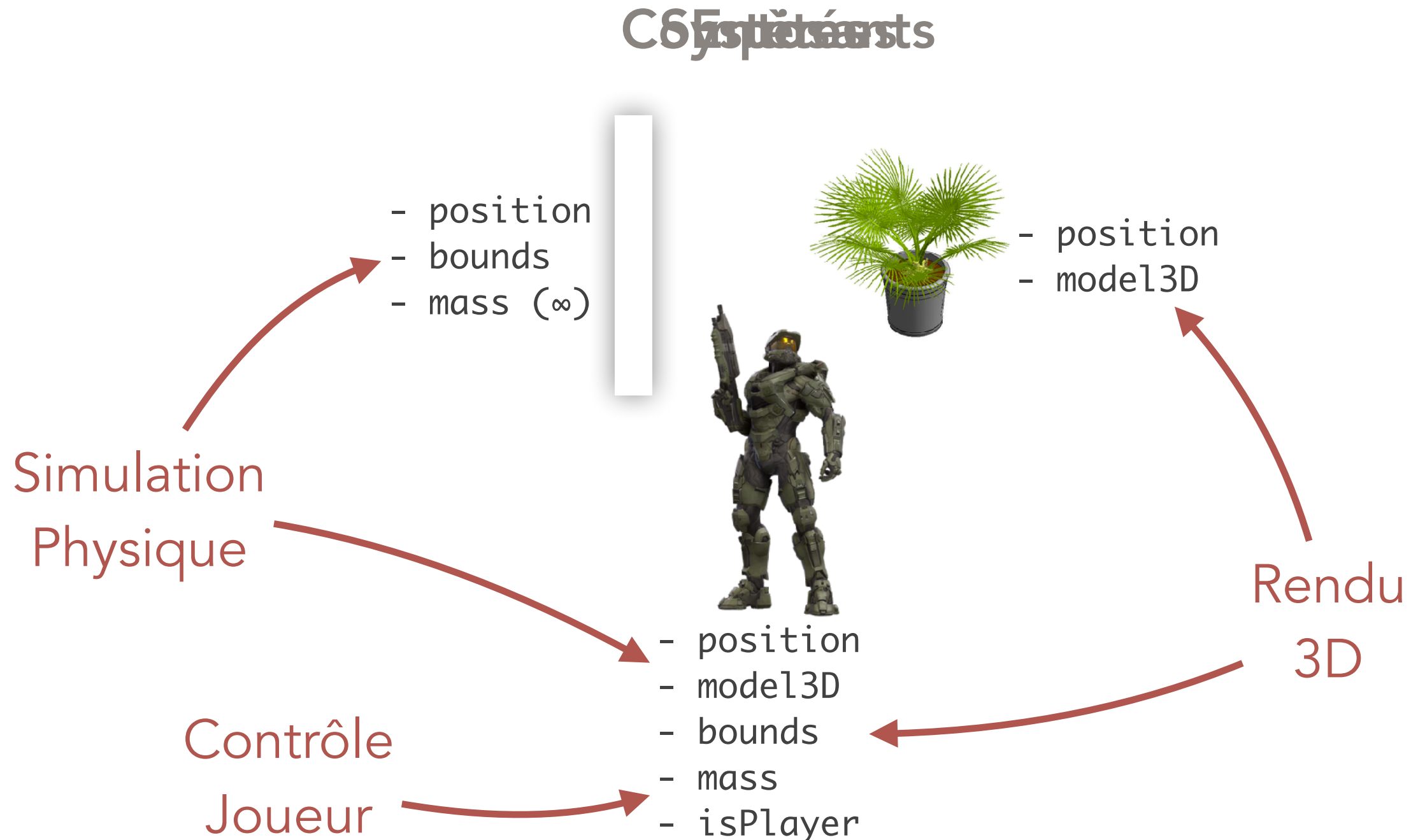
Thief: The Dark Project (1998)



Dungeon Siege (2002)



# Entité-Composant-Système



---

# Du Jeu Vidéo à l'IHM

---

Jeu Vidéo	IHM
MÀJ à tickrate fixe (ex. 60Hz)	MÀJ réactive à des événements
Relations faibles entre éléments	Relations de contenance et proximité
Un couple clavier/souris	Multiples périphériques
Pipeline d'exécution prédéterminé	Orchestration dynamique de comportements

---

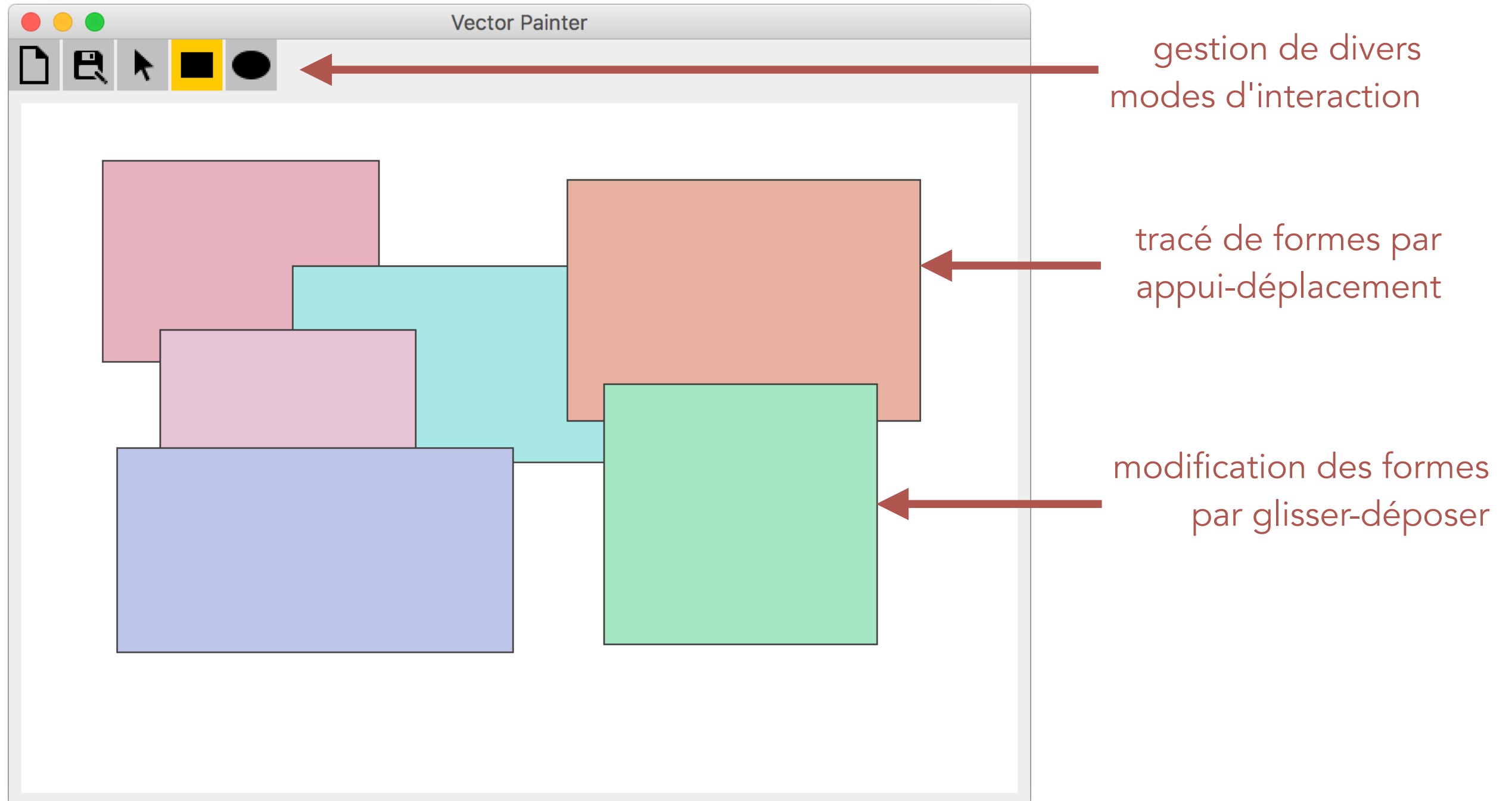
# Adaptation à l'IHM

---

## Tout est Entité !

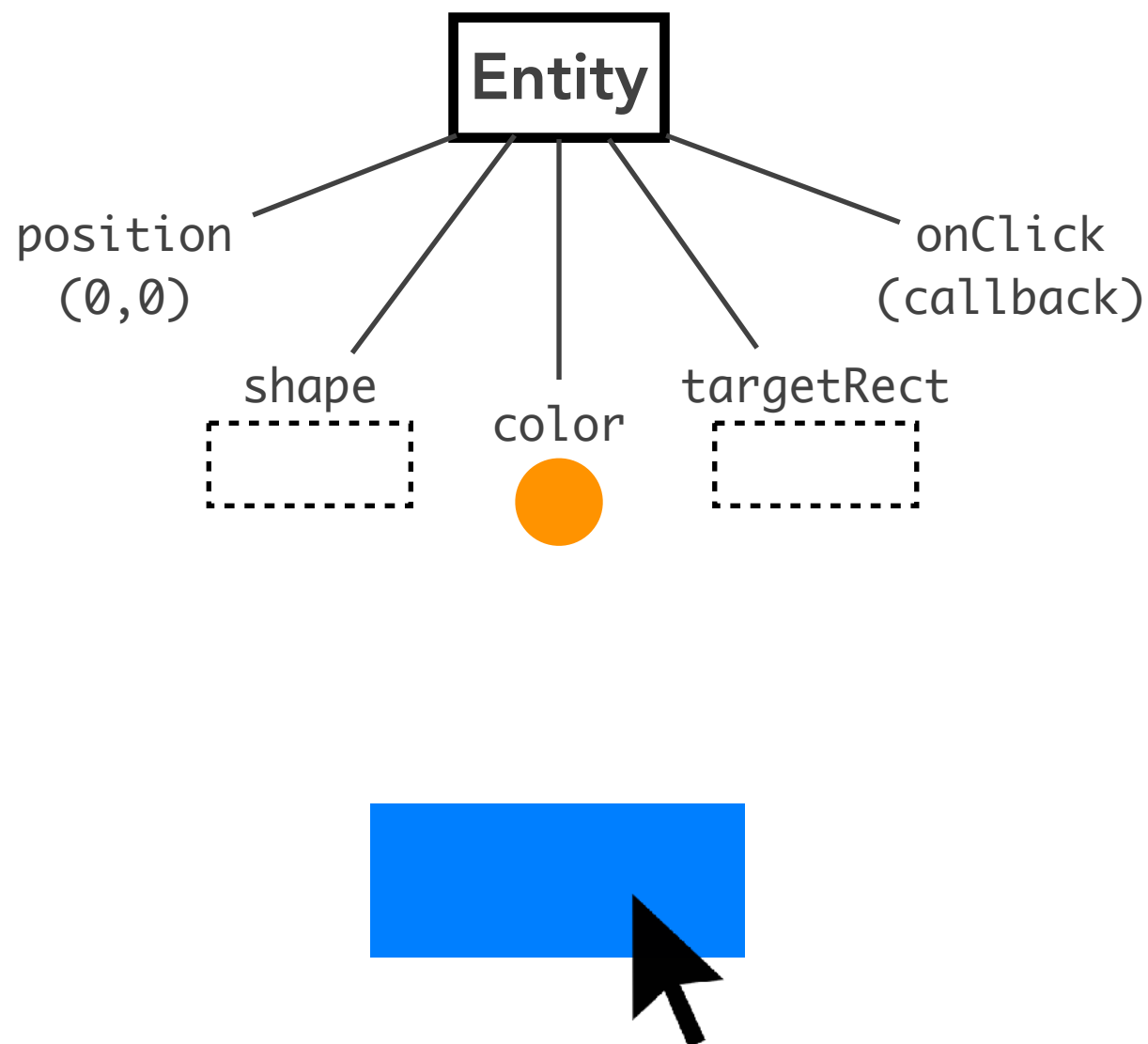
- ⇒ widgets (boutons, menus, ...)
- ⇒ périphériques (souris, clavier, ...)
- ⇒ Systèmes

# Exemple d'application



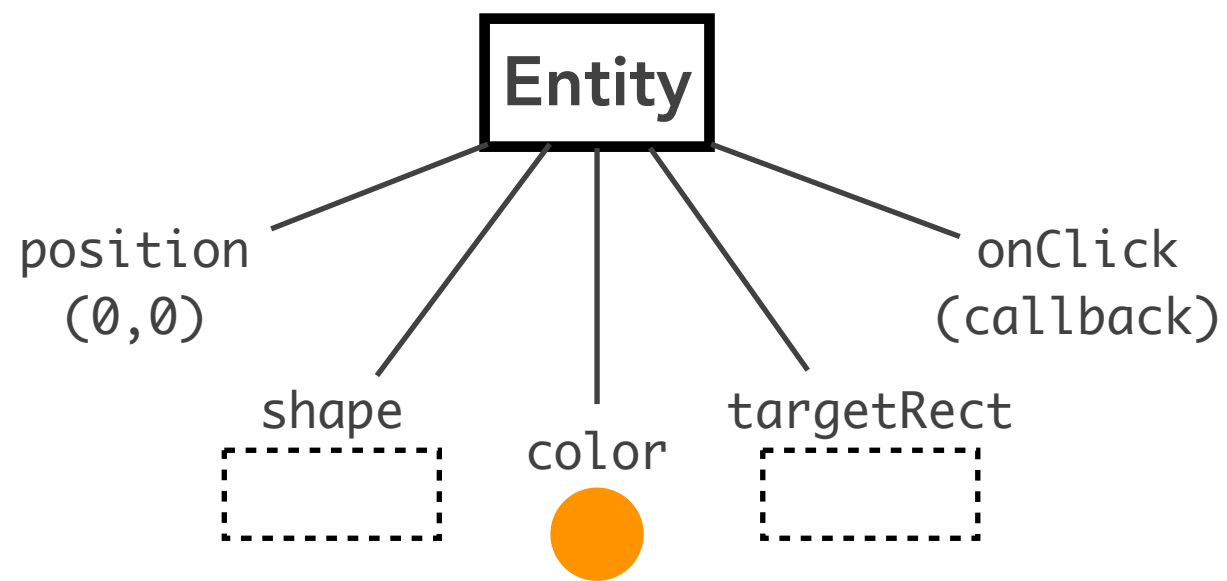


# Création d'Entité



```
e = new Entity()
e.add("position",
      new Point(0,0))
e.add("shape",
      new Rectangle(0,0,50,20))
e.add("color",
      new Color(255,127,0))
e.add("targetRect",
      e.get("shape"))
e.add("onClick",
      () => {e.color.set(0,127,255)})
```

# Fabrique d'Entité



```
e = new Button(...)
```

---

# Caractéristiques d'une Entité

---

Ajout/retrait de champs (*Composants*) dynamique

↳ **dictionnaire**

Suppression manuelle (pas de ramasse-miettes)

Visible de tous (à partir des Composants qu'elle possède)

↳ **existe** dès sa création, sans arbre de scène

Aucun héritage (classe ou prototype)

# Liste des Entités actives

## Interface

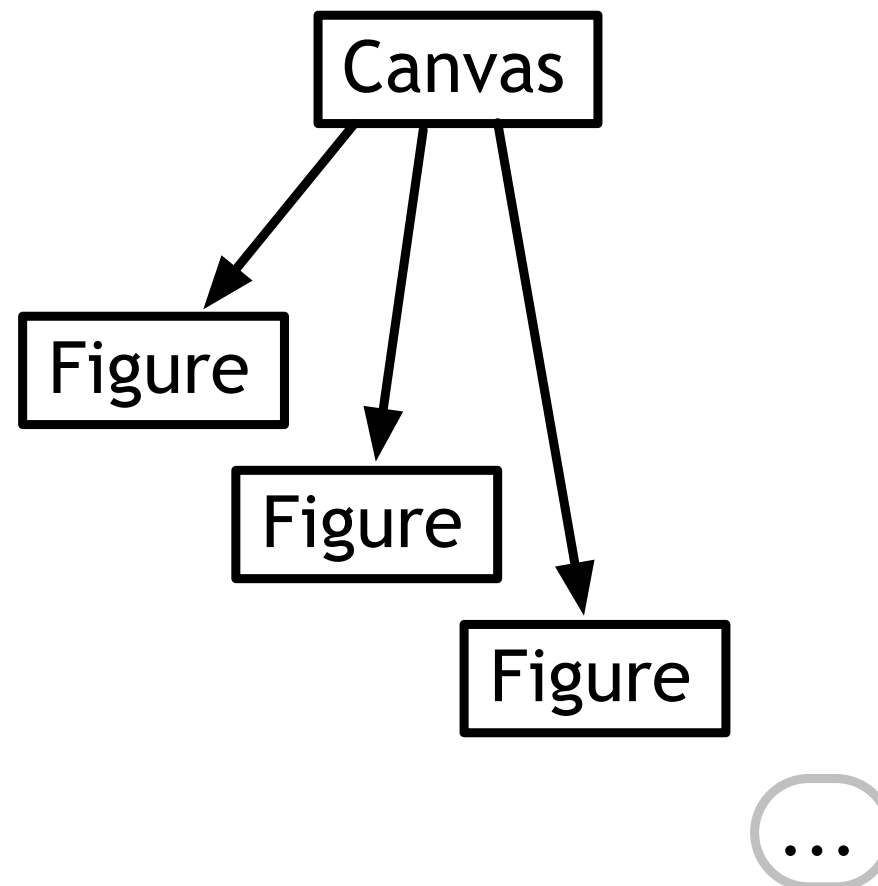
Button (📄)

Button (📁)

Button (🖱️)

Button (■)

Button (●)



---

# Sélecteurs

---

Obtenir une liste des Entités possédant certains Composants

↳ `selectAll(composantA, composantB, ...)`

Analogue aux sélecteurs CSS et tags de SwingStates

Utilisation systématique de caches pour optimiser

# Implémentation d'un Système

```
s = new Entity()
s.add("execute", () => {
    targets = selectAll("position",
                        "targetShape")
    for (p : selectAll("isPointer")) {
        t = findClosest(p, targets)
        p.add("target", t)
        t.add("pointedBy", p)
    }
})
s.add("readsMouseMotion", true)
```

Un Système est caractérisé  
par le Composant execute

Liste des cibles

Liste des pointeurs

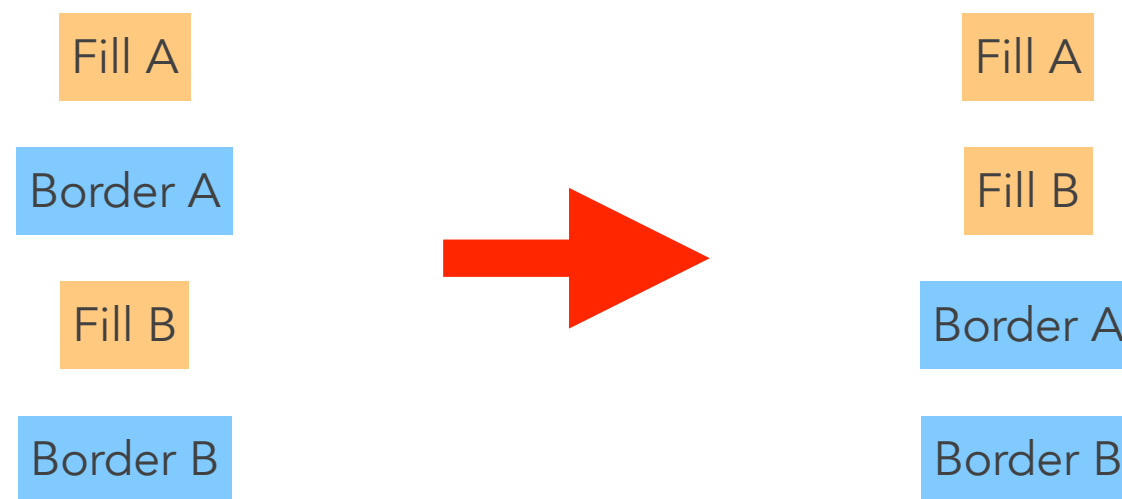
Bubble Cursor

Quand s'exécuter

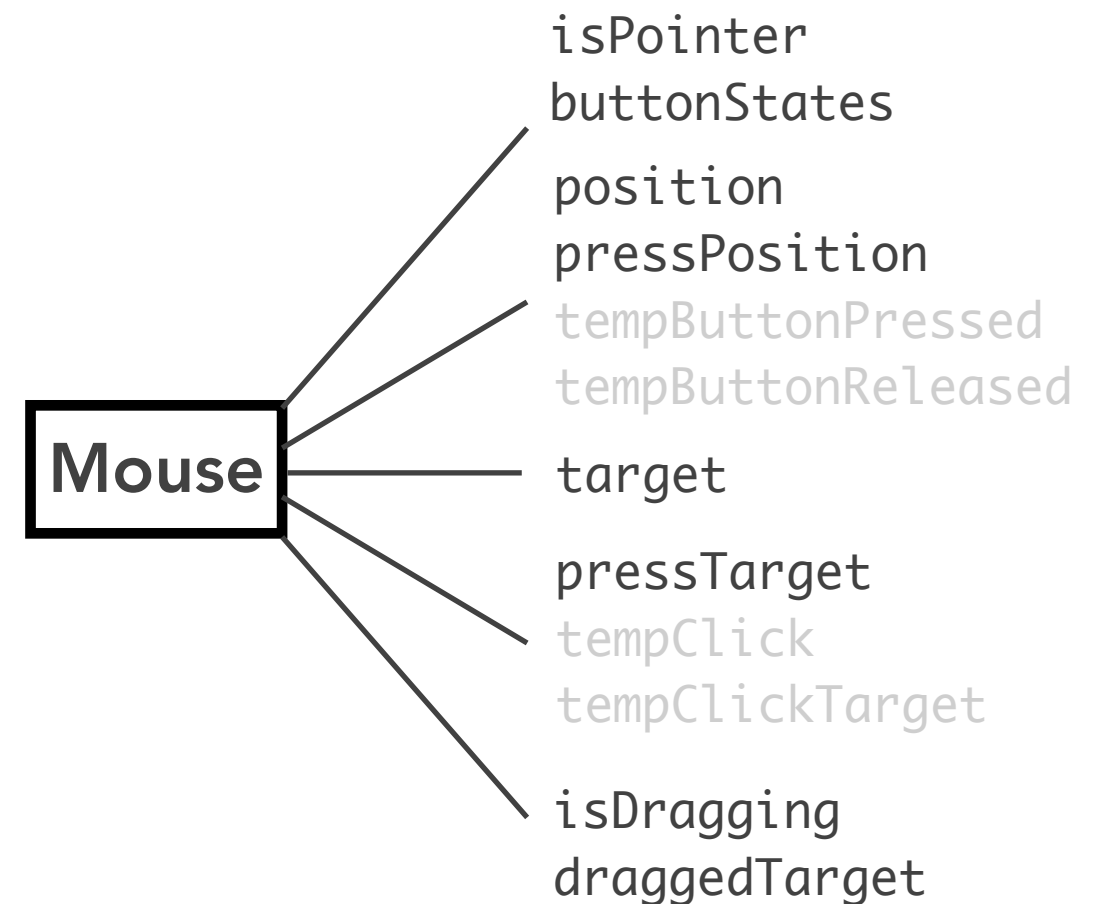
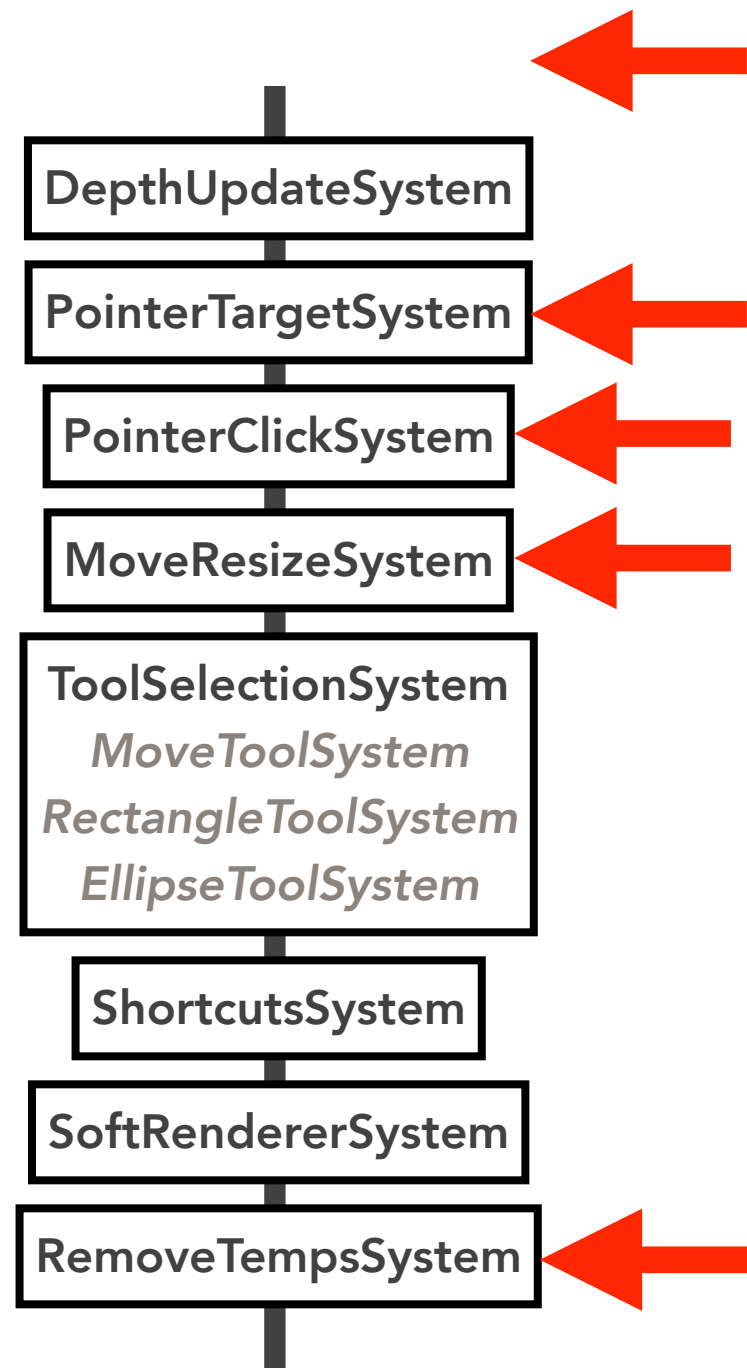
# Caractéristiques d'un Système

Exécute **un** type de comportement sur **plusieurs** Entités  
(ex. tracé de bordures, ciblage de la souris,  
déclenchement de raccourci clavier)

Ordonne les comportements par *type* plutôt qu'*Entité*

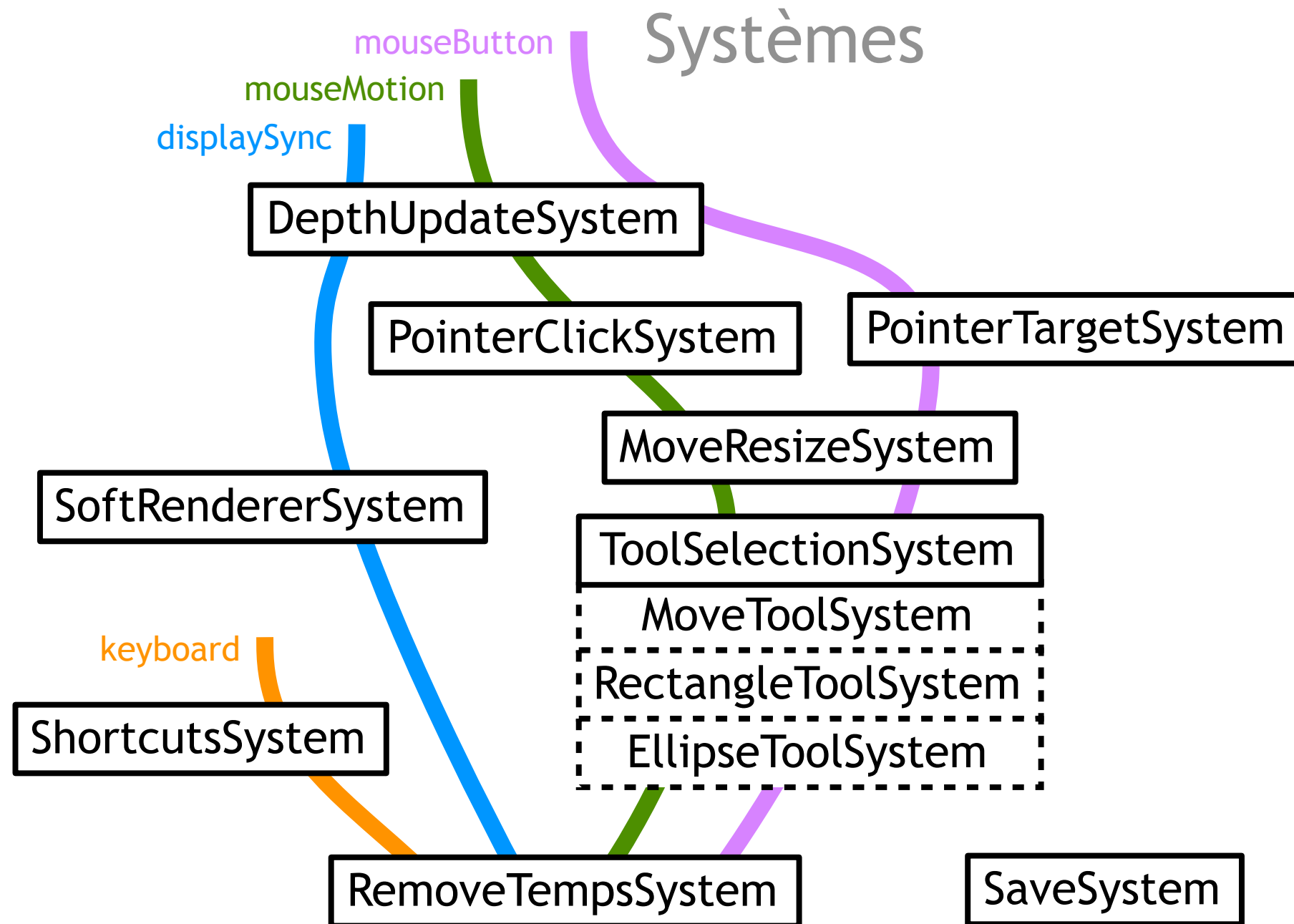


# Chaîne des Systèmes





# Chaînes des Systèmes



---

# Liste des périphériques actifs

---

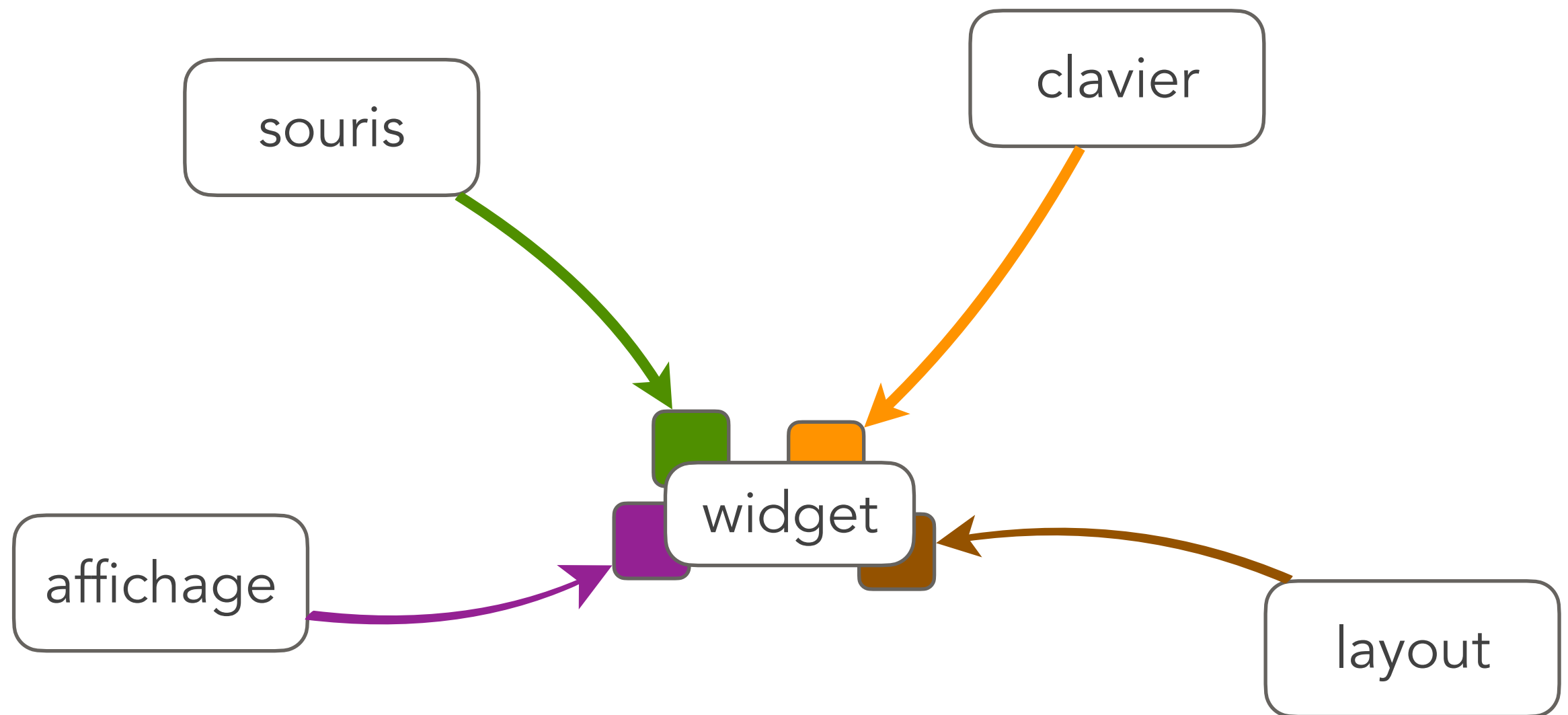
## Périphériques

Mouse

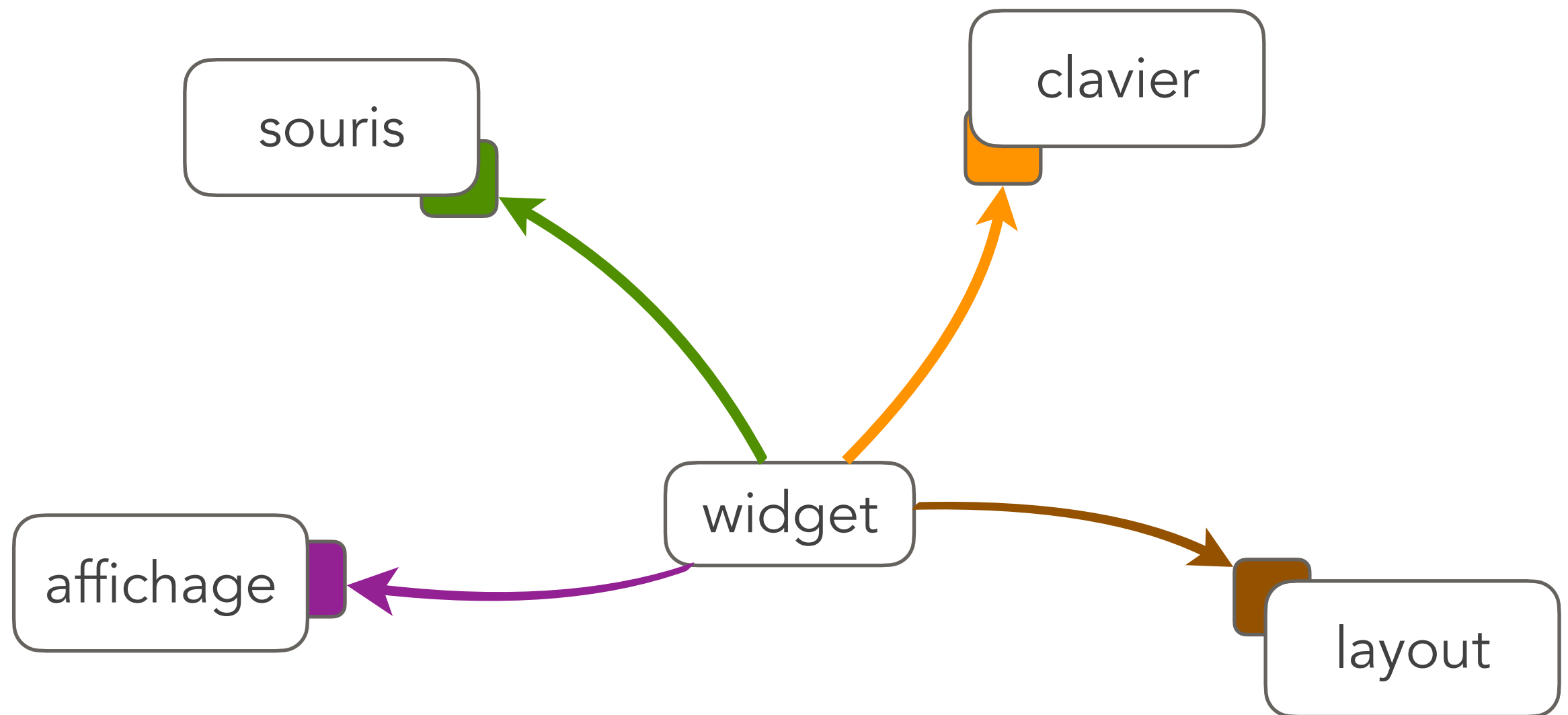
Keyboard

View

# OOP



# ECS



---

# Contribution

---

Clarification et extension d'ECS pour la programmation d'IHM

Présentation d'une boîte à outils (Polyphony) et implémentation d'une application de dessin

Étude de 3 implémentations majeures et choix de conception

---

# Perspectives

---

- ⇒ Utilisation d'un langage de programmation plus adapté
- ⇒ Développement d'interactions plus complexes
- ⇒ Mise à disposition d'autres utilisateurs

Merci de votre attention