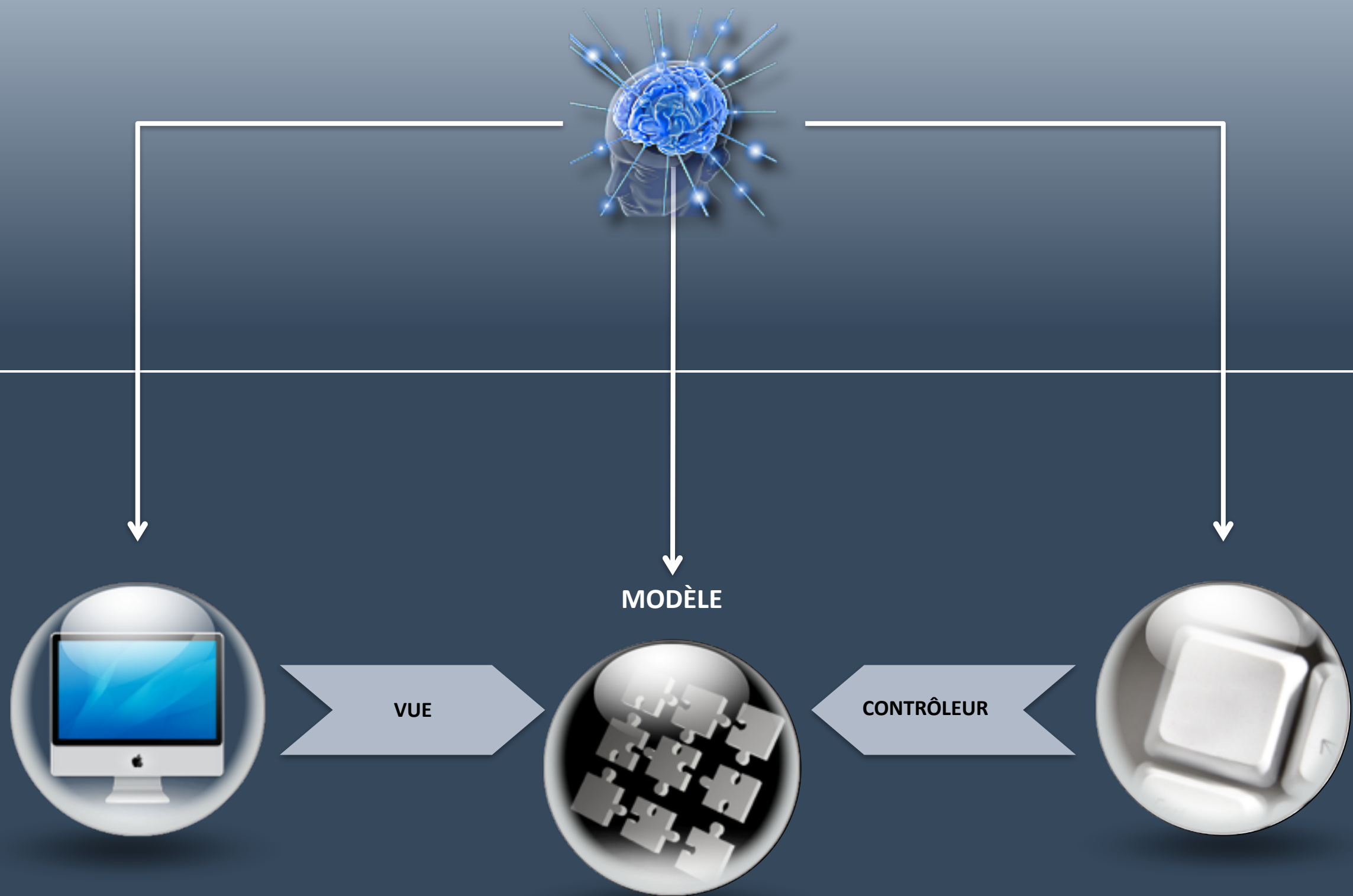


MVC : MODÈLE - VUE - CONTRÔLEUR





INTRODUCTION



CONTENU DE LA PRÉSENTATION

PRÉSENTATION ET HISTORIQUE

ARCHITECTURE

MVC II

SYNTHÈSE





INTRODUCTION



CONTENU DE LA PRÉSENTATION

PRÉSENTATION ET HISTORIQUE

ARCHITECTURE

MVC II

SYNTHÈSE



La naissance « MVC »

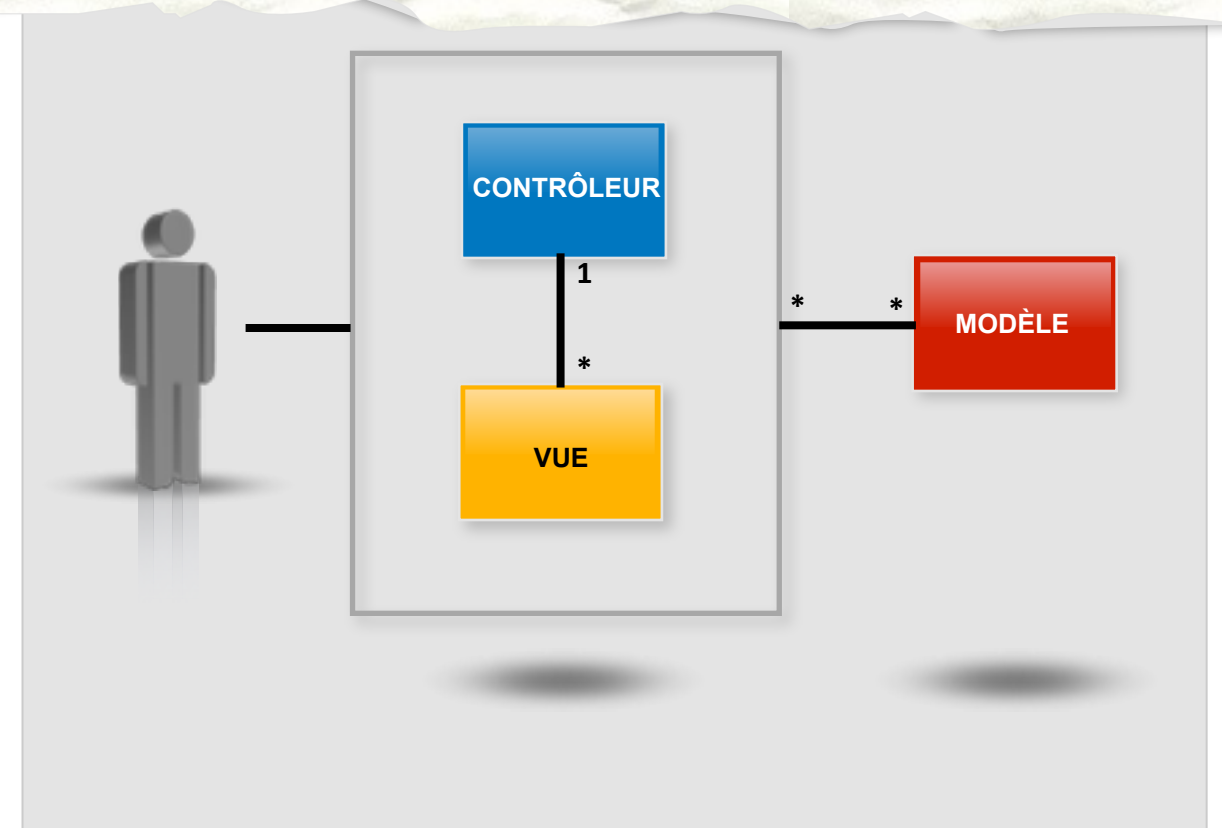
COMPOSANTS

- 👤 **Le modèle MVC (Modèle Vue Contrôleur) a fait son apparition en 1978.**
- 👤 **Un an plus tard, Alan Kay le présente dans un projet Smalltalk en les termes suivants : « Le but principal est de combler l'écart entre la représentation humaine du modèle et le modèle digital dans l'ordinateur ».**
- 👤 **Il établit le diagramme à venir.**

“Dans le paradigme MVC l'entrée utilisateur, la modélisation du monde extérieur, l'aspect visuel présenté l'utilisateur sont explicitement séparés et gérés par trois types d'objet, chacun spécialisé dans sa tâche.” [Burbeck 92]

INTERÊTS

- 👤 **Le contrôleur peut être utilisé par plusieurs vues.**
- 👤 **L'utilisateur ne voit qu'une vue, qui fait appel à un contrôleur.**
- 👤 **Le contrôleur fait appel au modèle pour réaliser des traitements de données.**
- 👤 **Le résultat de ces traitements sera récupéré par la vue qui les affichera.**



Le modèle MVC permet de créer une application permet de séparer les différentes parties constituant une application

Modèle

- ✓ Implémente les fonctions de l'application.
- ✓ Représente la structure des données dans l'application et les opérations spécifiques sur ces données.

Vue

- ✓ Présente l'information à l'utilisateur.
- ✓ Présente les données sous une certaine forme à l'utilisateur, suivant un contexte d'exploitation.

Contrôleur

- ✓ Gère les interactions de l'utilisateur.
- ✓ Traduit les interactions utilisateur par des appels de méthodes (comportement) sur le modèle et sélectionne la vue appropriée basée sur l'état du modèle.

Le modèle MVC permet de créer une application à partir de parties distinctes

différentes



L'objet « Modèle »

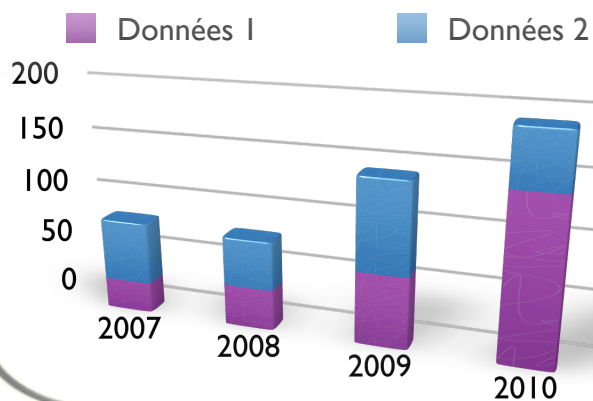
Variables d'instance :

- ☒ données1 : [2007:27, 2008:36, 2009:63, 2010:143]
- ☒ données2 : [2007:55, 2008:43, 2009:80, 2010:48]

Modèle

- ✓ Implémente les fonctions de l'application.
- ✓ Représente la structure des données dans l'application et les opérations spécifiques sur ces données.

V2



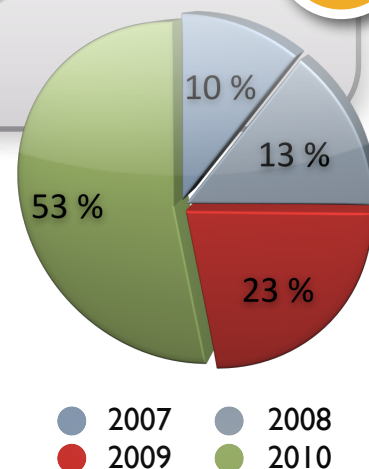
Vue

- ✓ Présente l'information à l'utilisateur.
- ✓ Présente les données sous une certaine forme à l'utilisateur, suivant un contexte d'exploitation.

V1

	2007	2008	2009	2010
Données 1	27	36	63	143
Données 2	55	43	80	48

V3



C1

CONTRÔLEUR

C2

CONTRÔLEUR

Contrôleur

- ✓ Gère les interactions de l'utilisateur.
- ✓ Traduit les interactions utilisateur par des appels de méthodes (comportement) sur le modèle et sélectionne la vue appropriée basée sur l'état du modèle.



L'objet « Modèle »

Le modèle MVC permet de créer un objet à partir de différentes parties constitutives

différentes

Vue

Contrôleur

Modèle

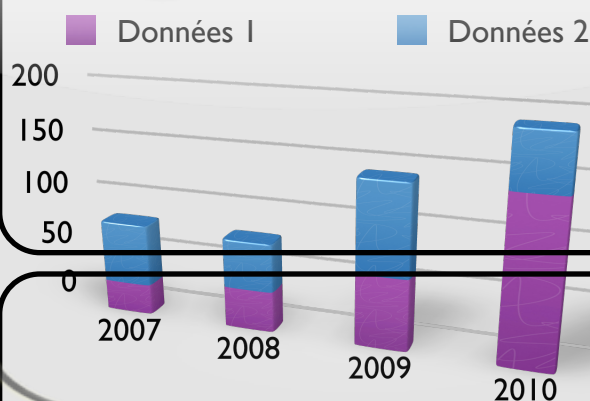
✓ Implémente les fonctions de l'application.

✓ Représente la structure des données dans l'application et les opérations spécifiques sur ces données.

Variables d'instance :

- ✓ données1 : [2007:27, 2008:36, 2009:63, 2010:143]
- ✓ données2 : [2007:55, 2008:43, 2009:80, 2010:48]

V2



Vue

✓ Présente l'information à l'utilisateur.

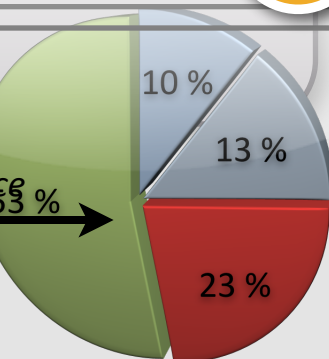
V1

✓ Présente les données sous une certaine forme à l'utilisateur, suivant un contexte d'exploitation.

	2007	2008	2009	2010
Données 1	27	36	63	143
Données 2	55	43	80	48

V3

référence



2007 2008
2009 2010

C1

CONTRÔLEUR

C2

CONTRÔLEUR

Contrôleur

✓ Gère les interactions de l'utilisateur.

✓ Traduit les interactions utilisateur par des appels de méthodes (comportement) sur le modèle et sélectionne la vue appropriée basée sur l'état du modèle.

référence

Modélisation objet



Les principales dates

Phase 1

Créé dans la fin des années 1970 par Trygve Reenskaug au Xerox PARC.
Applications aux GUI (Graphical User Interfaces).

Phase 3

En 1983, les vues spécifient les messages à envoyer au modèle avec des symboles.

Phase 5

In 1992 VisualWorks a remplacé ObjectWorks et a utilisé des composants de fine granularité GUI pour spécifier l'interface MVC.



Phase 2

Première version en 1980 qui utilise une sous-classe pour chaque vue à adapter au modèle.

Phase 4

ObjectWorks en 1989 a utilisé les « ValueHolder ».

Phase 6

Visualworks a ajouté la gestion des événements par des contrôleurs en 1998.



INTRODUCTION



CONTENU DE LA PRÉSENTATION

PRÉSENTATION ET HISTORIQUE

ARCHITECTURE

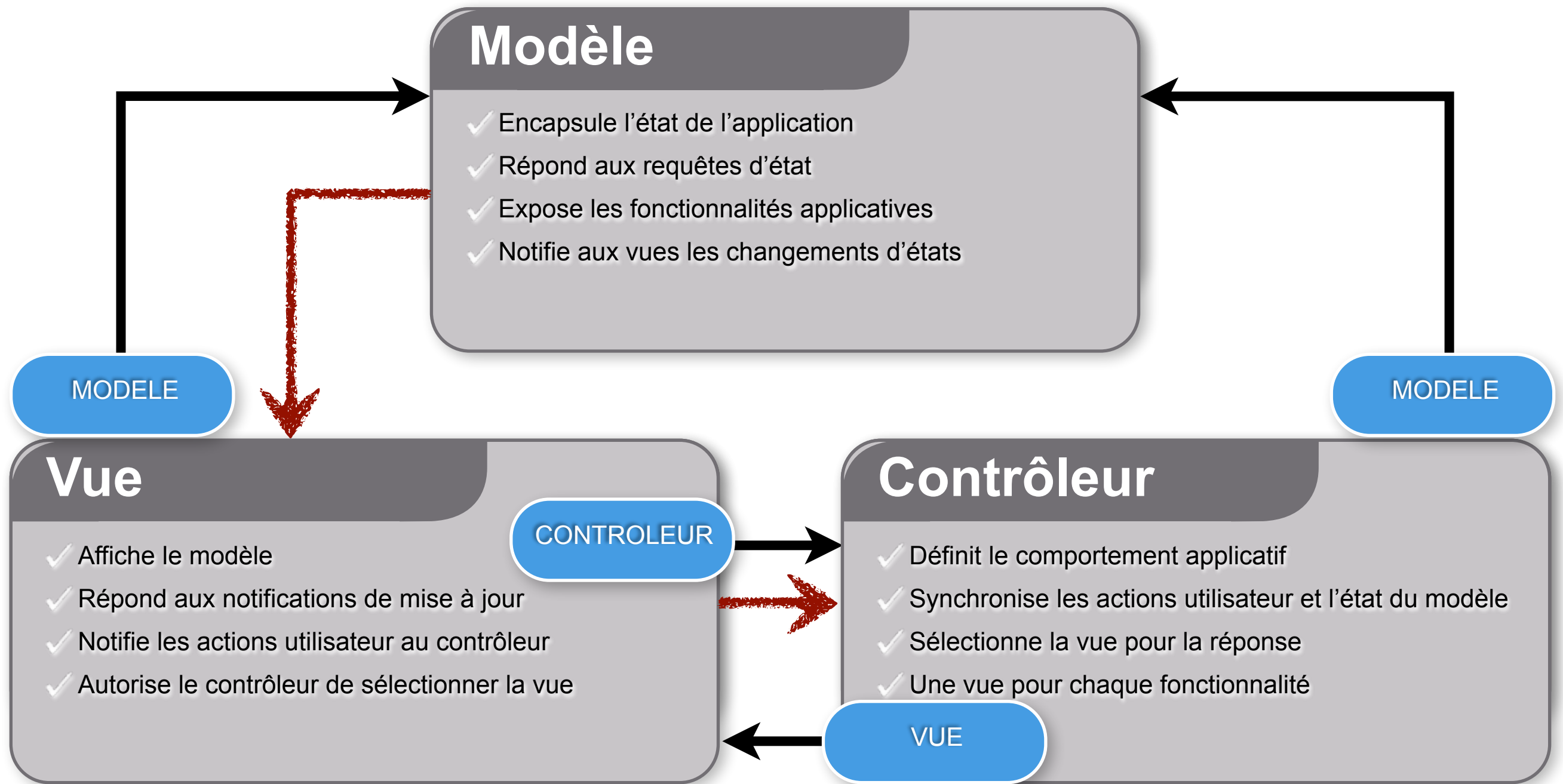
MVC II

SYNTHÈSE



ARCHITECTURE

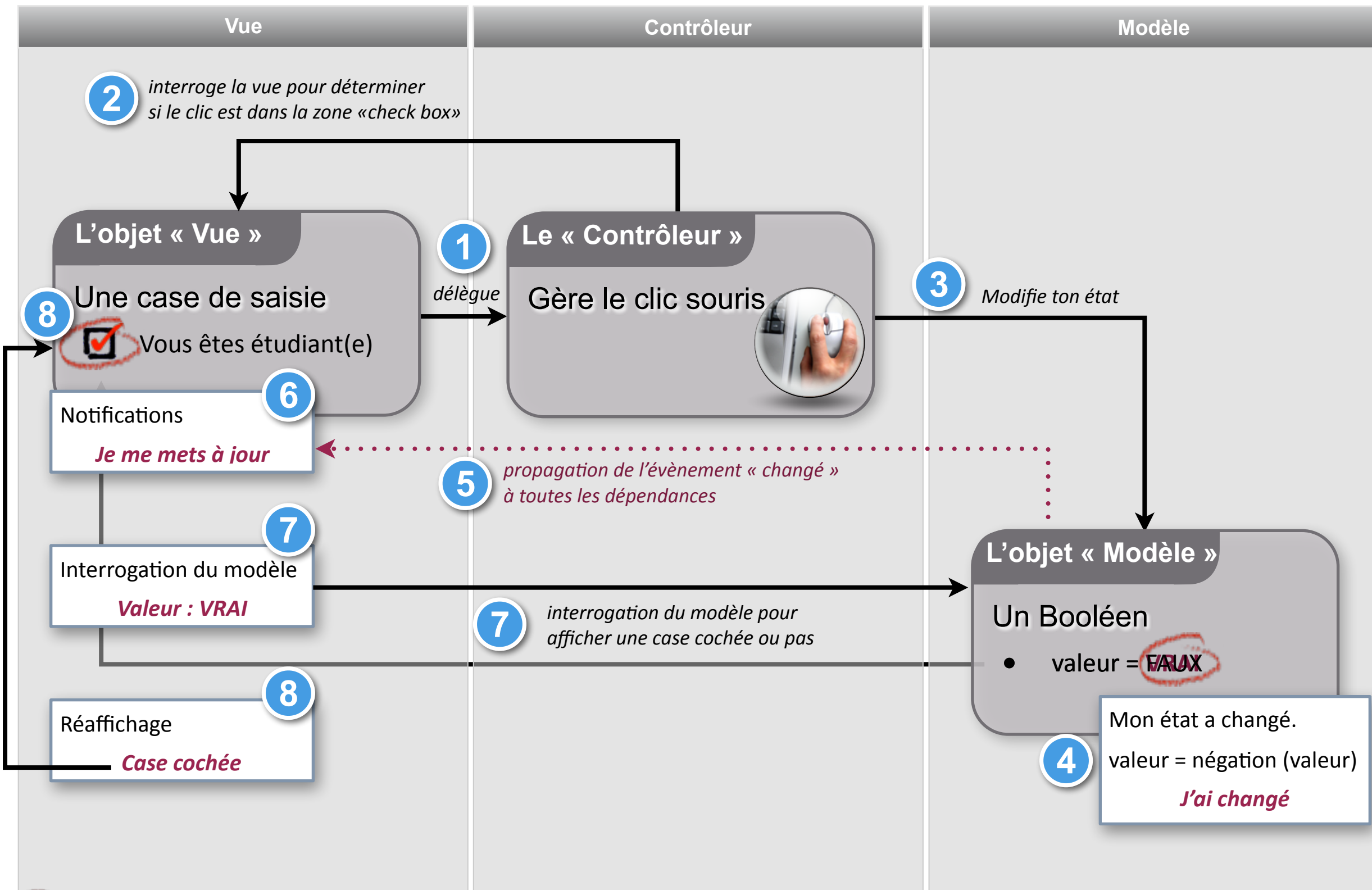
Le modèle MVC permet de créer une application permet de séparer les différentes parties constituant une application



→ Invocation explicite de méthode

→ Invocation par évènement

Le modèle MVC appliqué aux composants graphiques de base L'exemple de la «check box»





INTRODUCTION



CONTENU DE LA PRÉSENTATION

PRÉSENTATION ET HISTORIQUE

ARCHITECTURE

MVC II

SYNTHÈSE



MODÈLE MVC II

Evolution du modèle MVC pour l'appliquer à la programmation web

Modèle

- ✓ Il représente la couche métier d'une application, présentant des classes permettant de créer et de manipuler les objets contenant des données métier manipulées par l'application au travers de traitements, constituant les services métiers.

JavaBeans, EJB, Web Services, ...

Vue

- ✓ Elle constitue les éléments d'interface utilisateurs : pages web, contrôles Web...

JSP, JSF, HTML, XML, ...

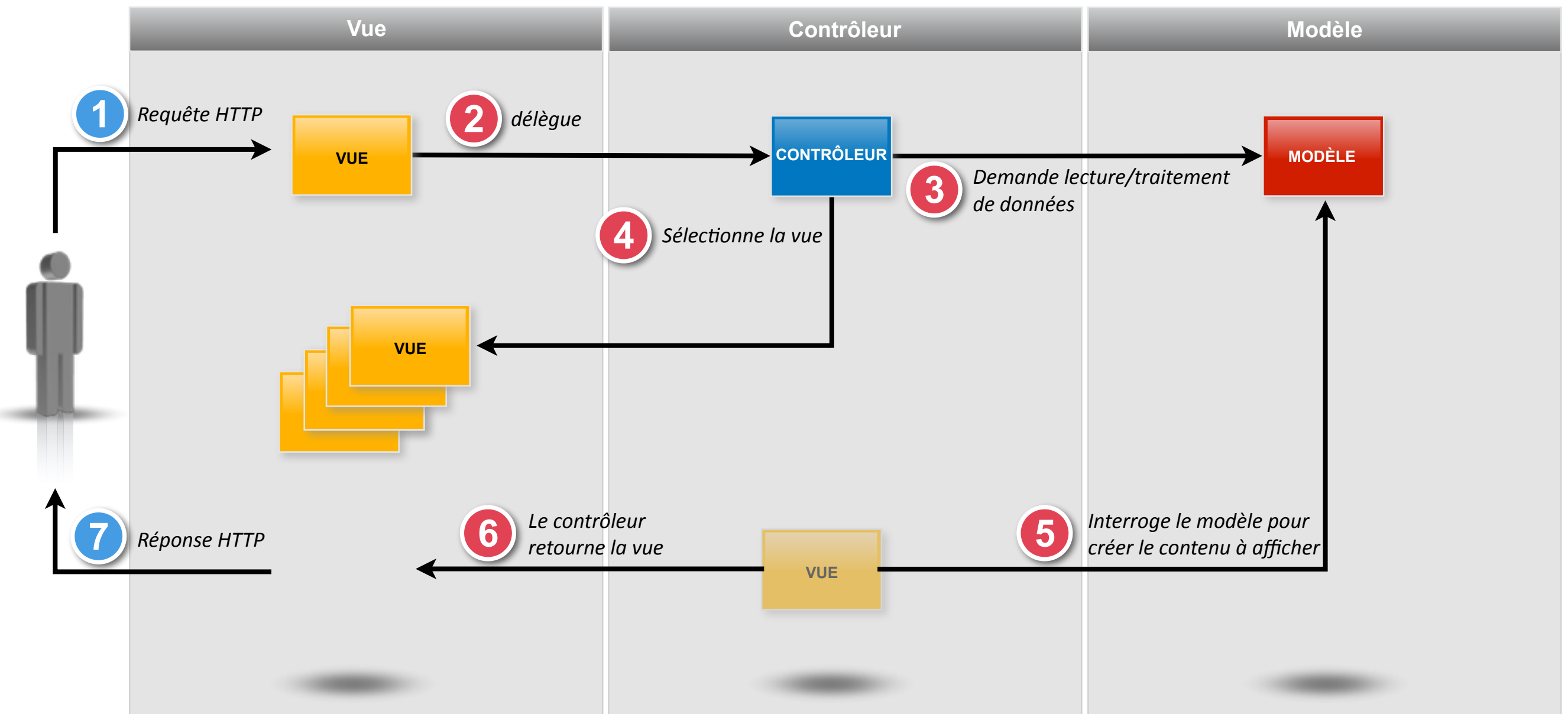
Contrôleur

- ✓ Il permet de piloter l'application, il interprète les actions à réaliser et ordonne leur exécution (lecture, traitement de données et mises à jour).

Action Servlets, qui dirigent les actions à effectuer en fonction de la requête émise par la JSP.

MODÈLE MVC II

Evolution du modèle MVC pour l'appliquer à la programmation web





INTRODUCTION



CONTENU DE LA PRÉSENTATION

PRÉSENTATION ET HISTORIQUE

ARCHITECTURE

MVC II

SYNTHÈSE



Bonnes pratiques

1

Le modèle doit être orienté métier pour supporter le couplage des vues et des contrôleurs.

2

Le modèle ne doit jamais contenir des informations concernant l'état de la GUI et ne doit pas proposer de services spécifiques à des vues particulières et à des contrôleurs.

3

Souvent, la vue et le contrôleur sont groupés -> réduit la complexité

Les principaux

1

Fiabilité

Les couches présentation et métier sont complètement séparées.

2

Adaptabilité

Toute représentation visuelle du modèle peut être facilement intégrée.
Par exemple, une Application Web initialement conçue pour des clients HTML et qui doit être modifiée afin d'accepter également des clients Smartphones.

3

Productivité

La durée de développement est réduite de manière significative, en permettant la parallélisation du travail des équipes.

4

Coûts réduits

Coûts réduits de développement grâce à la séparation fonctionnelle qui fait que les compétences de haut niveau ne sont plus nécessaires que pour la réalisation de certains composants de l'application.

INCONVÉNIENTS

Les principaux

Fiabilité

1

La complexité de l'application est accrue.

Adaptabilité

2

La mise à jour peut se révéler potentiellement excessive.

Productivité

3

Bien qu'interchangeables, la vue et le contrôleur sont fortement liés au modèle.

Ce qu'il faut retenir...

- 1 Le MVC favorise le développement et la maintenance du code.
- 2 Sur de gros projets et/ou avec de grandes équipes de développements, l'application d'un tel modèle de conception se révèle très performant. Il existe aujourd'hui des frameworks très avancés qui se basent sur le MVC ou le MVC2.
- 3 L'utilisation de ces frameworks facilite sa mise en place et cadre sa réalisation.

MERCI DE VOTRE ATTENTION

QSA

