

# Architectures et frameworks Flex / AIR

Matthieu Segret

<http://matthieusegret.com>

27 mai 2008



# Intellicore Tech Talks

- Java WebObjects : de Toyota à l'iPhone
- Comment utiliser l'humanité pour faire du calcul ?
- Planification et gestion de projet informatique : SCRUM
- Comment apprendre à lire à une machine ?
- WOX : Gestion alternative de la persistance en Flex
- Search Engine Optimization - L'art du référencement web
- Modèle, modèle ... vous avez dit modèle ?
- Architectures et frameworks Flex / AIR
- Les techniques avancées de référencement
- Apache Web Server Internals

<http://techtalks.intellicore.net>

# Architectures et frameworks Flex / AIR

Matthieu Segret

<http://matthieusegret.com>

27 mai 2008



Flex / AIR

FLEX / AIR

# Flex / AIR

- Conçu pour réaliser des clients riches

# Flex / AIR

- Conçu pour réaliser des clients riches
- Un framework

# Flex / AIR

- Conçu pour réaliser des clients riches
- Un framework
- 2 langages : ActionScript et MXML

# Flex / AIR

- Conçu pour réaliser des clients riches
- Un framework
- 2 langages : ActionScript et MXML
- Philosophie basée sur le pattern Composite

# Flex / AIR

- Conçu pour réaliser des clients riches
- Un framework
- 2 langages : ActionScript et MXML
- Philosophie basée sur le pattern Composite
- AIR est basé sur Flex

Pourquoi utiliser un  
framework d'architecture ?

# Pourquoi utiliser un framework d'architecture ?

- Normaliser le développement

# Pourquoi utiliser un framework d'architecture ?

- Normaliser le développement
- Accélérer le développement

# Pourquoi utiliser un framework d'architecture ?

- Normaliser le développement
- Accélérer le développement
- Augmenter la robustesse

# Pourquoi utiliser un framework d'architecture ?

- Normaliser le développement
- Accélérer le développement
- Augmenter la robustesse
- Facilite l'évolutivité

# Pourquoi utiliser un framework d'architecture ?

- Normaliser le développement
- Accélérer le développement
- Augmenter la robustesse
- Facilite l'évolutivité
- Améliorer la maintenance

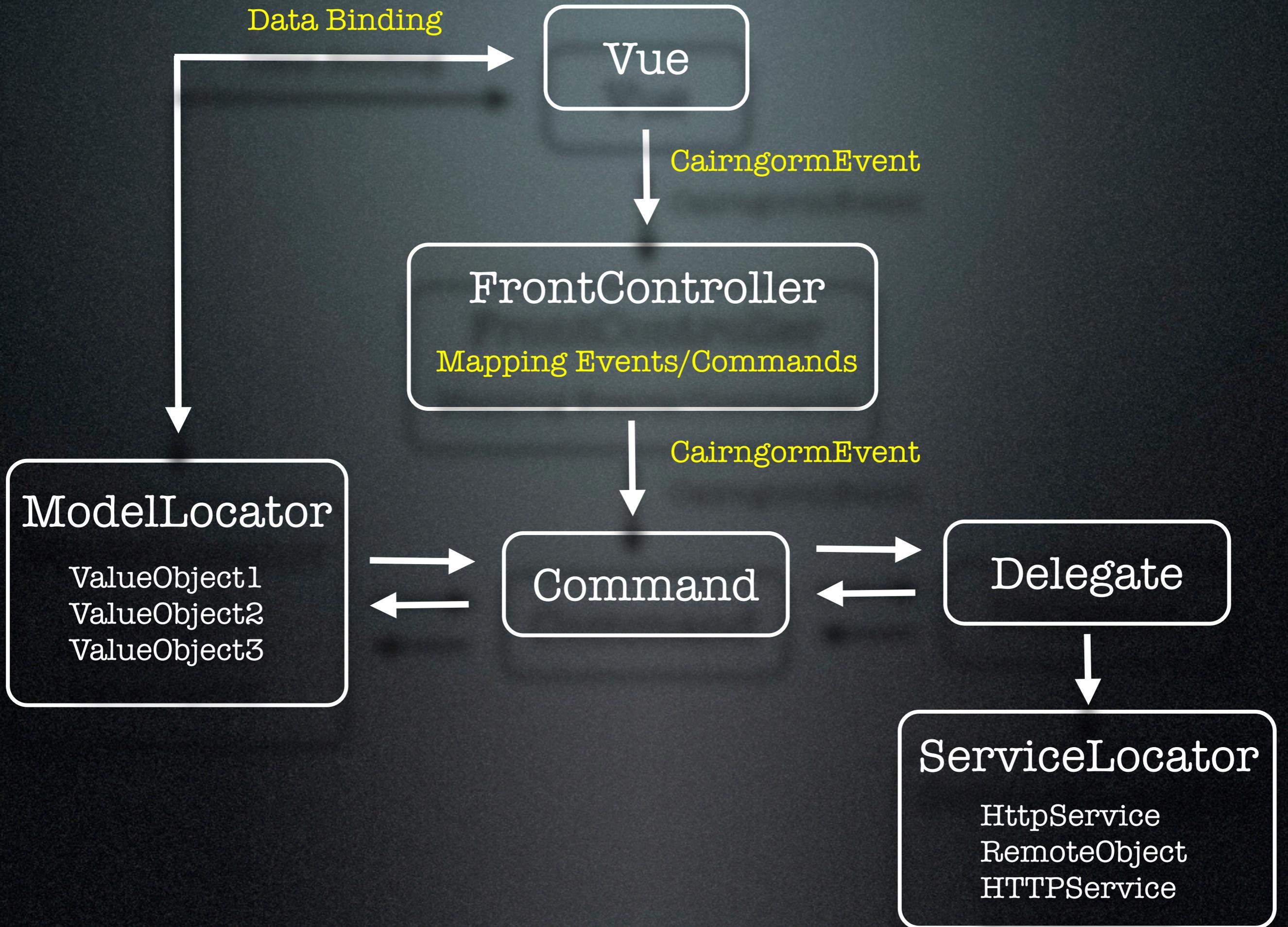
# Framework d'architecture

- Cairngorm
- PureMVC
- Guasax
- Mate
- Indigo
- Pandora
- Adapter un framework existant
- ... Votre framework

Cairngorm

# Cairngorm

- Le singleton **FrontController** gère des **Commands**
- Le singleton **ModelLocator** gère des **ValueObjects**
- Le singleton **ServiceLocator** gère des **Services**
- Les **Delegates** utilisent les **Services** du **ServiceLocator** pour communiquer le serveur



# Binding

[Bindable]

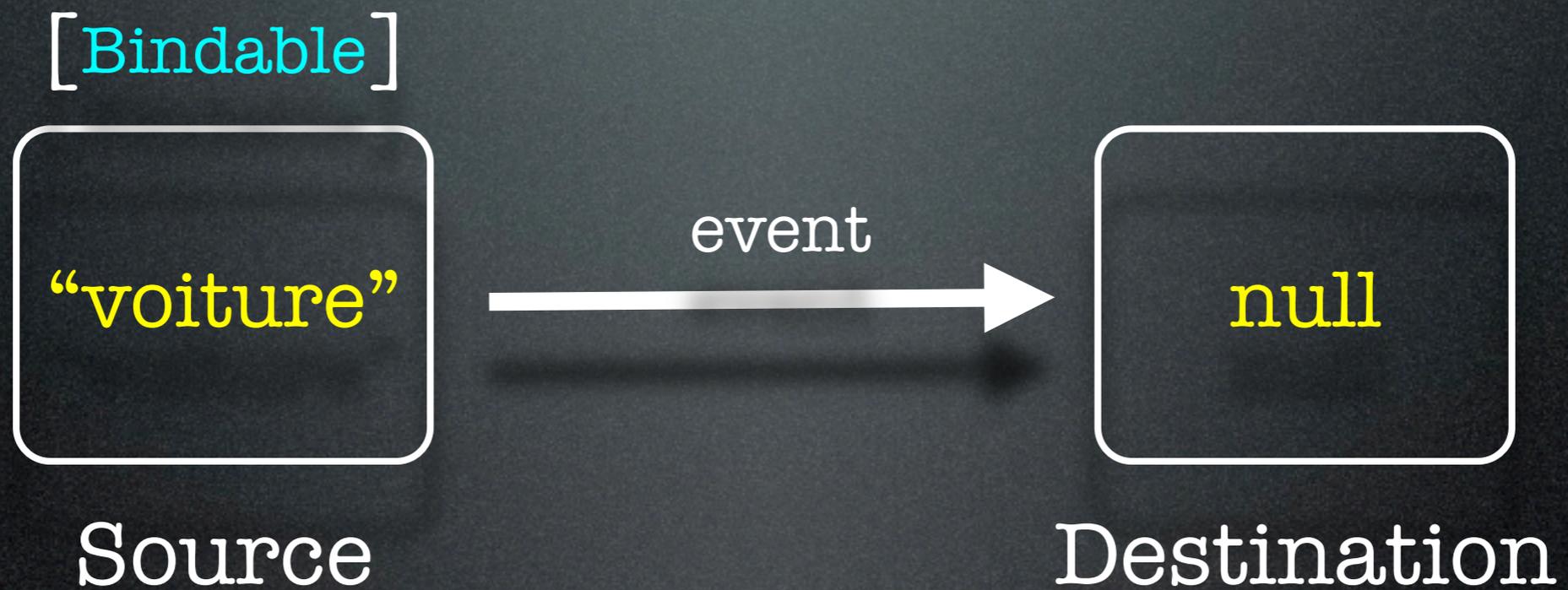
“voiture”

Source

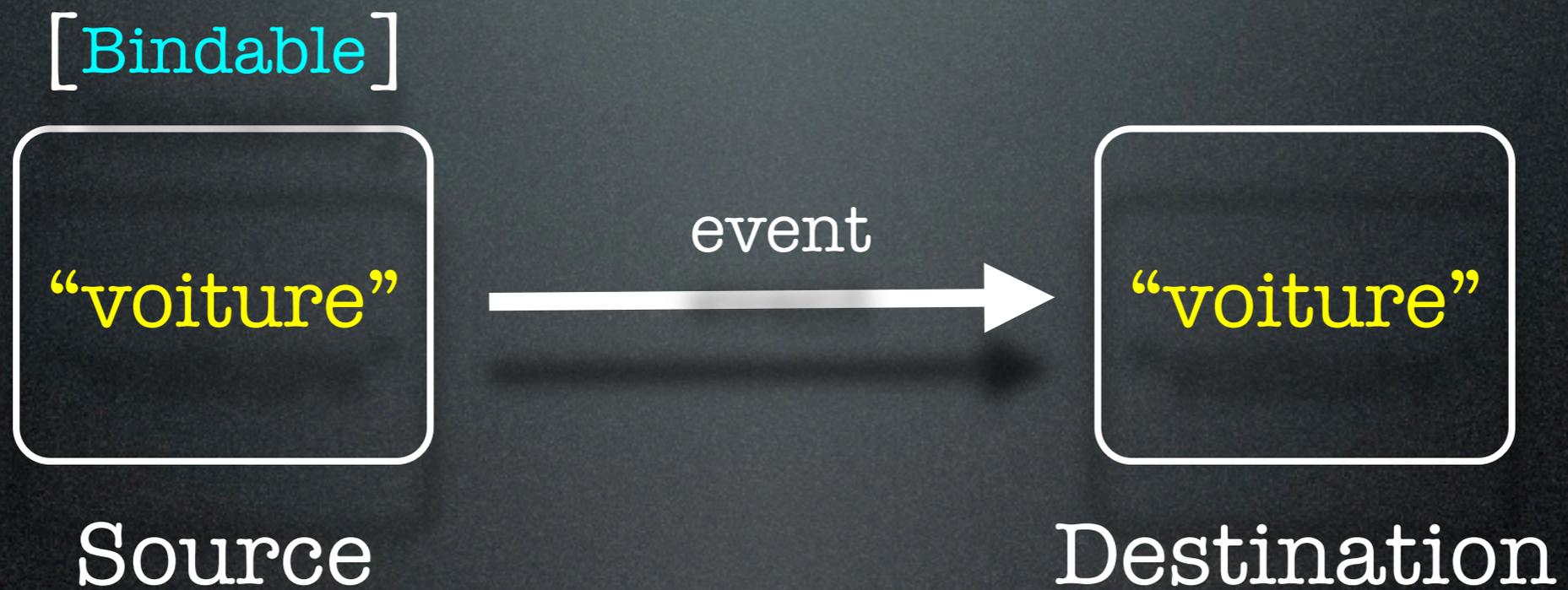
null

Destination

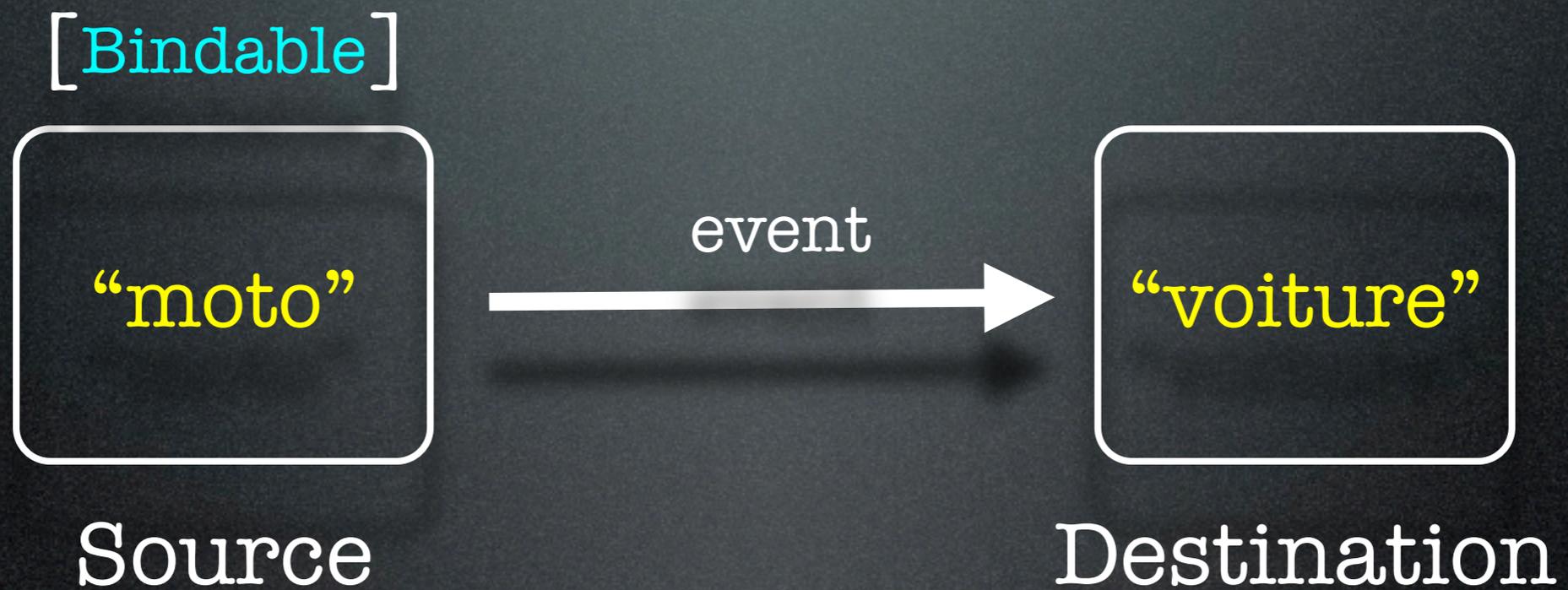
# Binding



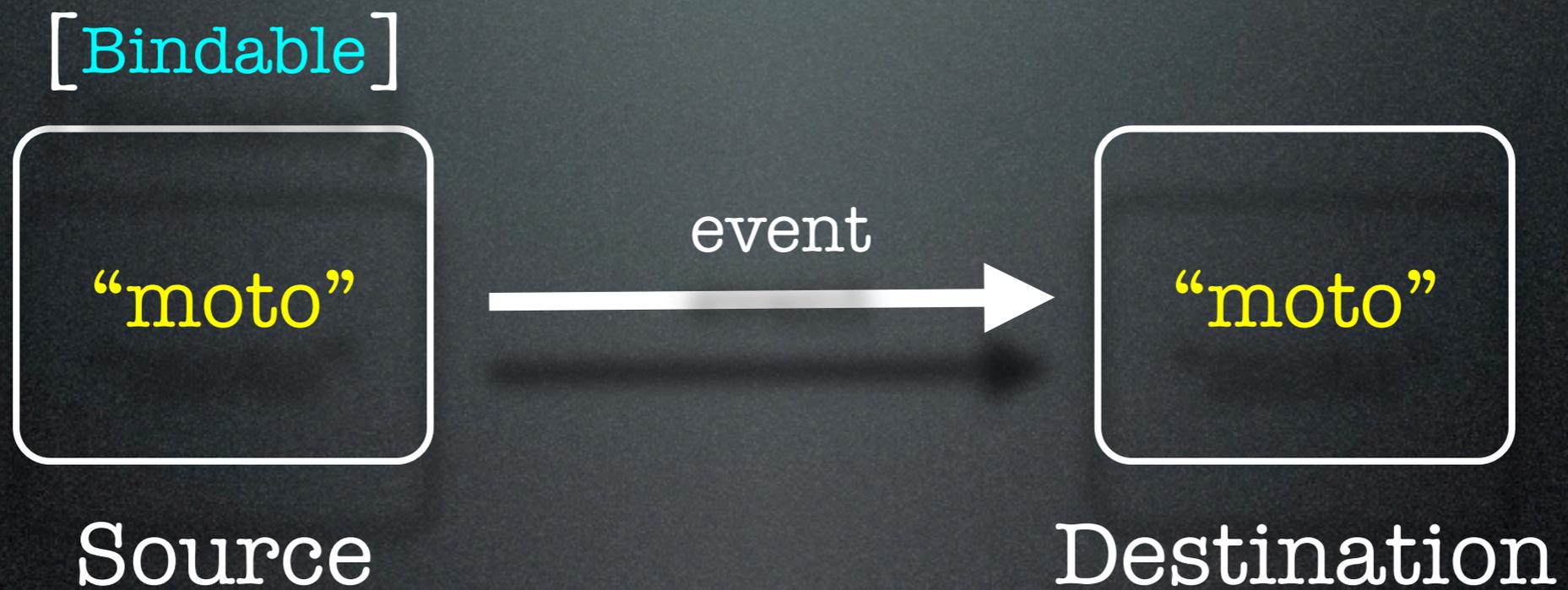
# Binding



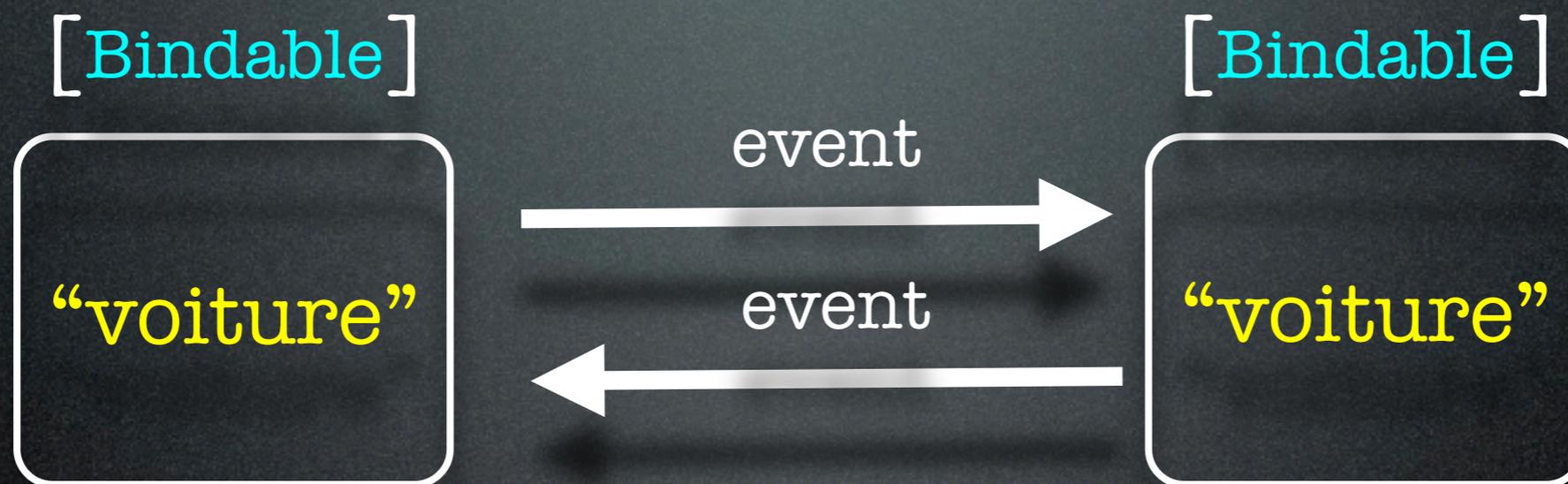
# Binding



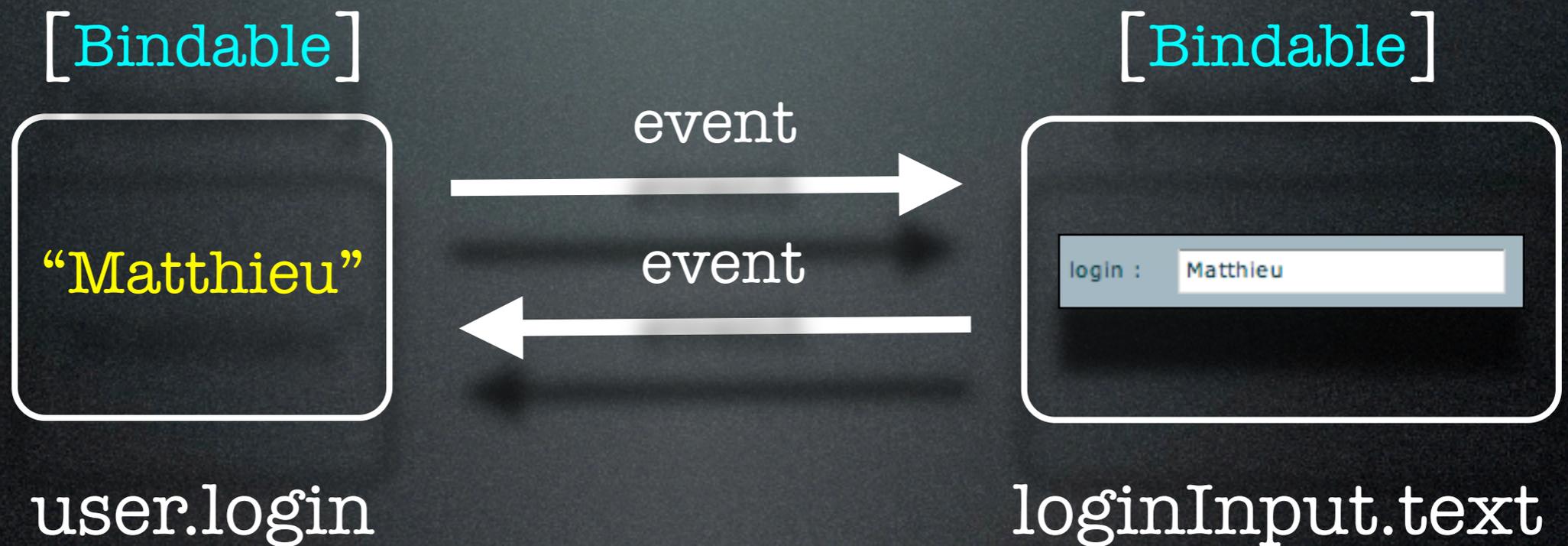
# Binding



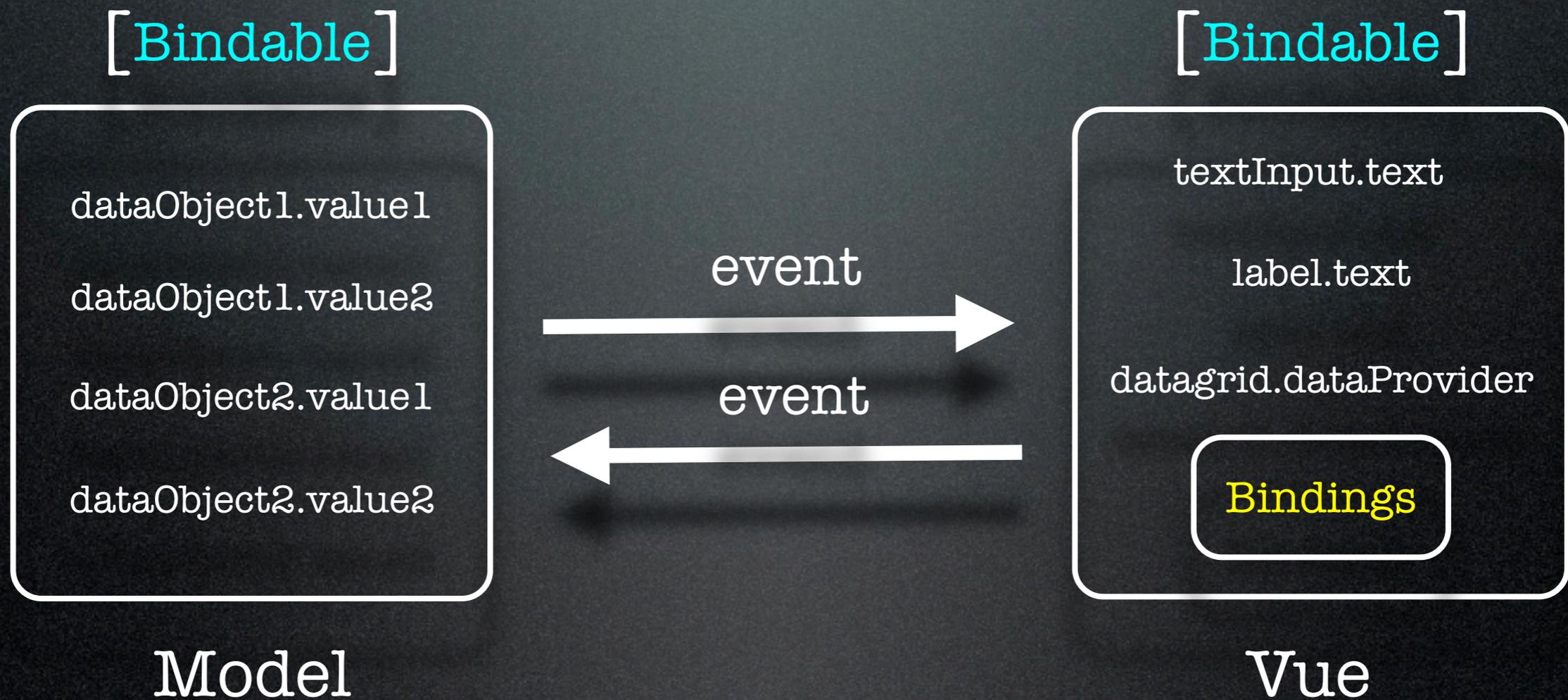
# Binding



# Binding View/Model



# Binding View/Model



# Inconvenients Cairngorm

# Inconvenients Cairngorm

- Une maintenance incertaine

# Inconvenients Cairngorm

- Une maintenance incertaine
- Gestion des événements Cairngorm

# Inconvenients Cairngorm

- Une maintenance incertaine
- Gestion des événements Cairngorm
- Peu de mises à jour

# Inconvenients Cairngorm

- Une maintenance incertaine
- Gestion des événements Cairngorm
- Peu de mises à jour
- Limité pour des projets de grande taille

# Avantages Cairngorm

# Avantages Cairngorm

- Data Binding

# Avantages Cairngorm

- Data Binding
- Assez facile à apprendre

# Avantages Cairngorm

- Data Binding
- Assez facile à apprendre
- Adapté pour des projets de taille moyenne

# Avantages Cairngorm

- Data Binding
- Assez facile à apprendre
- Adapté pour des projets de taille moyenne
- Supporté par Adobe

PureMVC

# PureMVC

- ActionScript 2
- ActionScript 3
- C#
- ColdFusion
- Java
- Perl
- PHP
- Python
- Ruby

# PureMVC

- Le singleton **Model** gère des **Proxys**
- Le singleton **View** gère des **Mediators**
- Le singleton **Controller** gère des **Commands**

# PureMVC

- Le singleton **Facade** est une interface unique pour communiquer avec la vue, le modèle et le contrôleur.

# Facade

- LE point d'accès aux **Commands**, **Mediators**, **Proxys**
- Notifie les **Commands**, **Mediators**, **Proxys**
- Contrôle le processus d'initialisation

# La Facade contient :

Facade

# La Facade contient :

Facade

- Instance Controller

Controller

# La Facade contient :

Facade

- Instance Controller
- Instance Model

Model

Controller

# La Facade contient :



- Instance Controller
- Instance Model
- Instance View

# La Facade contient :

Facade

Application

View

Model

Controller

- Instance Controller
- Instance Model
- Instance View
- Référence vers l'application

# La Facade contient :

## Facade

```
public static const STARTUP:String = "startup";  
public static const LOGIN:String = "login";  
public static const ADD_USER:String = "addUser";  
public static const REMOVE_USER:String = "removeUser";
```

Application

View

Model

Controller

- Instance Controller
- Instance Model
- Instance View
- Référence vers l'application
- Définition des constantes pour les noms des events

# La Facade contient :

## Facade

```
public static const STARTUP:String = "startup";  
public static const LOGIN:String = "login";  
public static const ADD_USER:String = "addUser";  
public static const REMOVE_USER:String = "removeUser";
```

Logger

Helpers

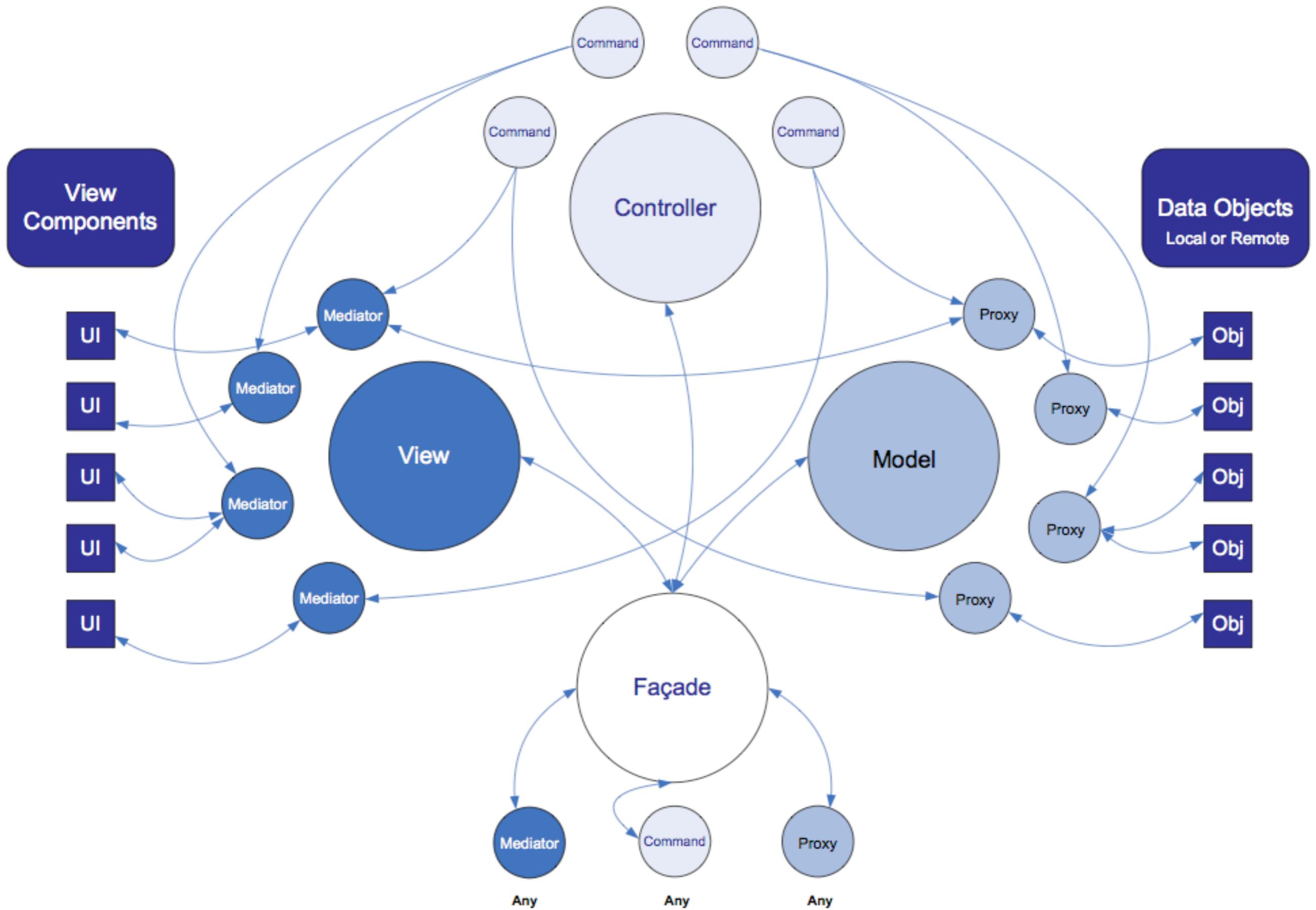
Application

View

Model

Controller

- Instance Controller
- Instance Model
- Instance View
- Référence vers l'application
- Définition des constantes pour les noms des events
- Instances outils



# Inconvenients PureMVC

# Inconvenients PureMVC

- Une maintenance incertaine

# Inconvenients PureMVC

- Une maintenance incertaine
- Plus lourd que Cairngorm

# Inconvenients PureMVC

- Une maintenance incertaine
- Plus lourd que Cairngorm
- Trop générique, pas assez Flex

# Inconvenients PureMVC

- Une maintenance incertaine
- Plus lourd que Cairngorm
- Trop générique, pas assez Flex
- Réception des notifications par les médiateurs difficile

# Avantages PureMVC

# Avantages PureMVC

- Architecture évolutive

# Avantages PureMVC

- Architecture évolutive
- Rôles et responsabilités bien répartis

# Avantages PureMVC

- Architecture évolutive
- Rôles et responsabilités bien répartis
- Adapté à la réalisation de projets de grande taille

# Avantages PureMVC

- Architecture évolutive
- Rôles et responsabilités bien répartis
- Adapté à la réalisation de projets de grande taille
- Différentes technologies supportées

# Avantages PureMVC

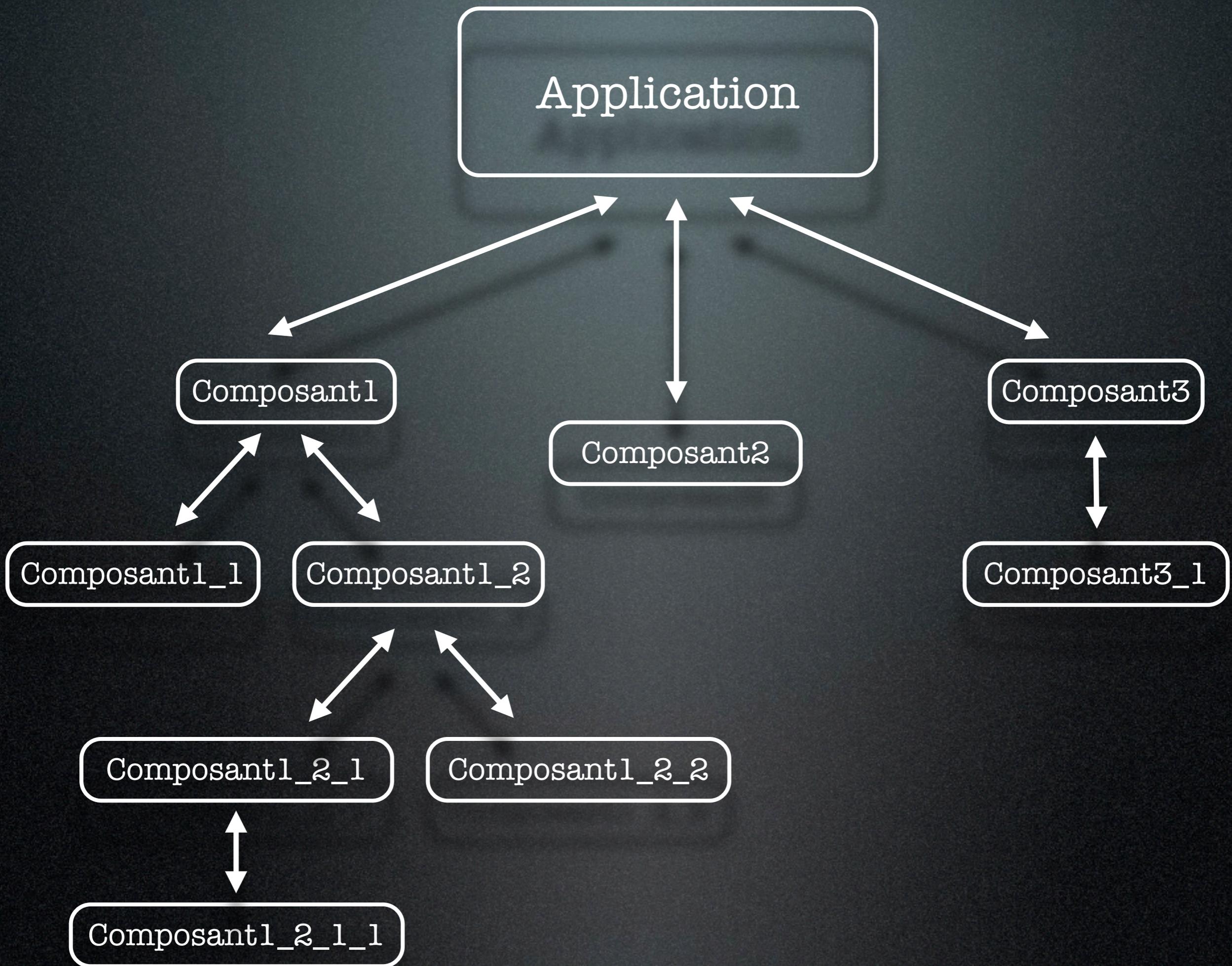
- Architecture évolutive
- Rôles et responsabilités bien répartis
- Adapté à la réalisation de projets de grande taille
- Différentes technologies supportées
- Documentation solide

# Avantages PureMVC

- Architecture évolutive
- Rôles et responsabilités bien répartis
- Adapté à la réalisation de projets de grande taille
- Différentes technologies supportées
- Documentation solide
- Communauté dynamique

monolithique

# Composants



Module

# Module

Application.swf  
ou  
module.swf

# Module

Application.swf  
ou  
module.swf



# Module

Application.swf  
ou  
module.swf

load

module1.swf



# Module

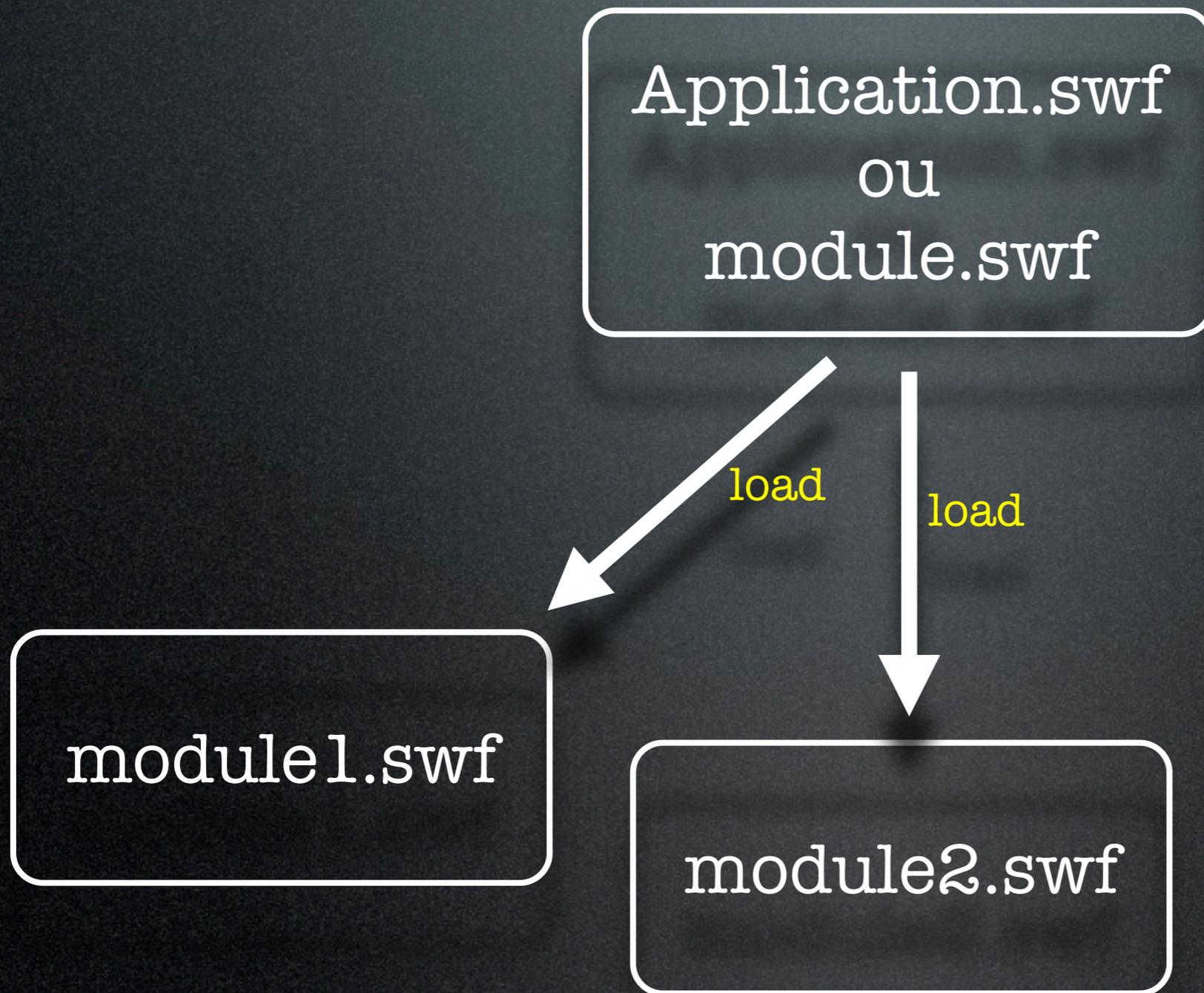
Application.swf  
ou  
module.swf

load

load

module1.swf

module2.swf



# Module

Application.swf  
ou  
module.swf

load

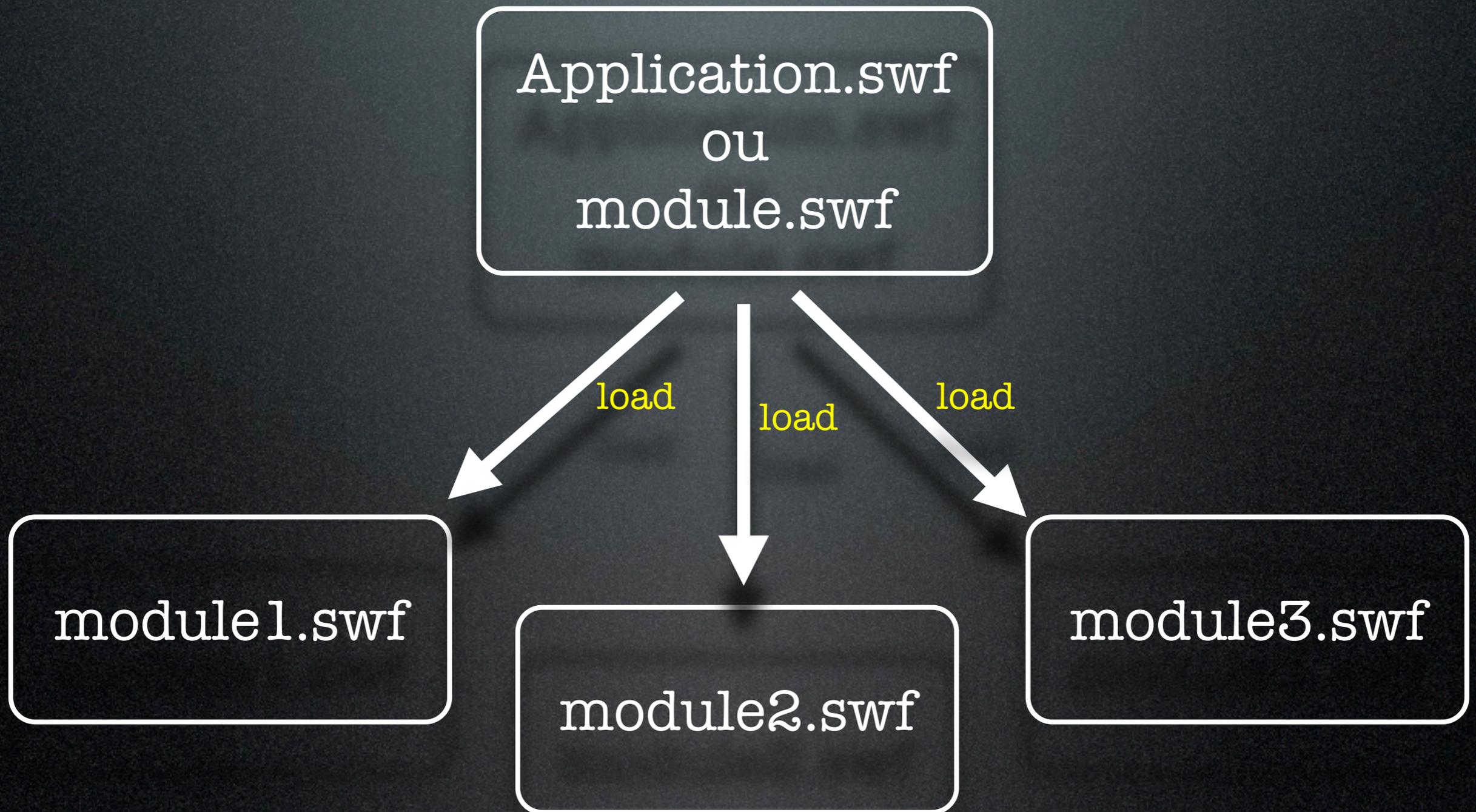
load

load

module1.swf

module2.swf

module3.swf



# Module

Application.swf  
ou  
module.swf

load

load

~~unload~~

module1.swf

module2.swf

module3.swf

# Module

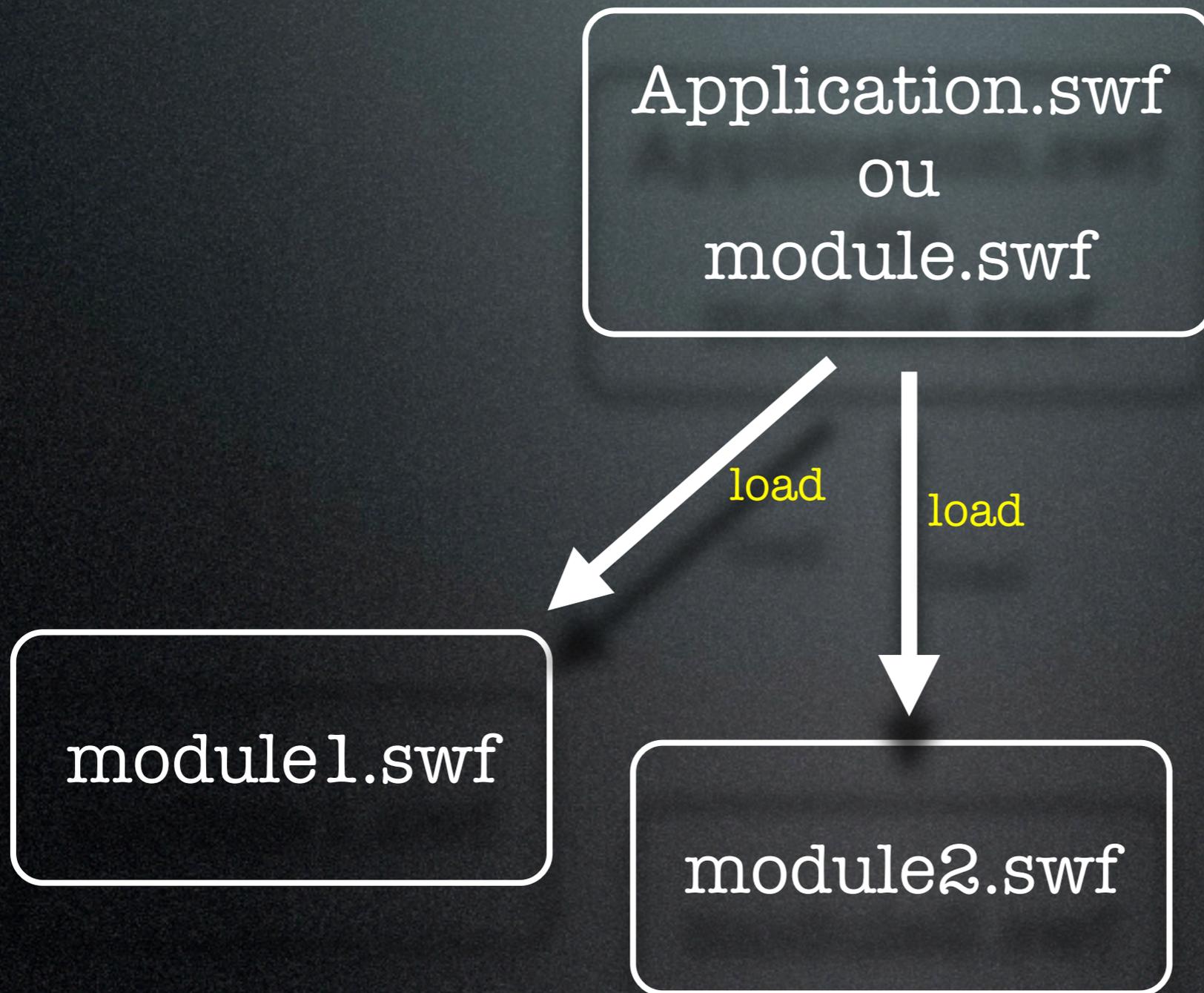
Application.swf  
ou  
module.swf

load

load

module1.swf

module2.swf



# Module

Application.swf  
ou  
module.swf

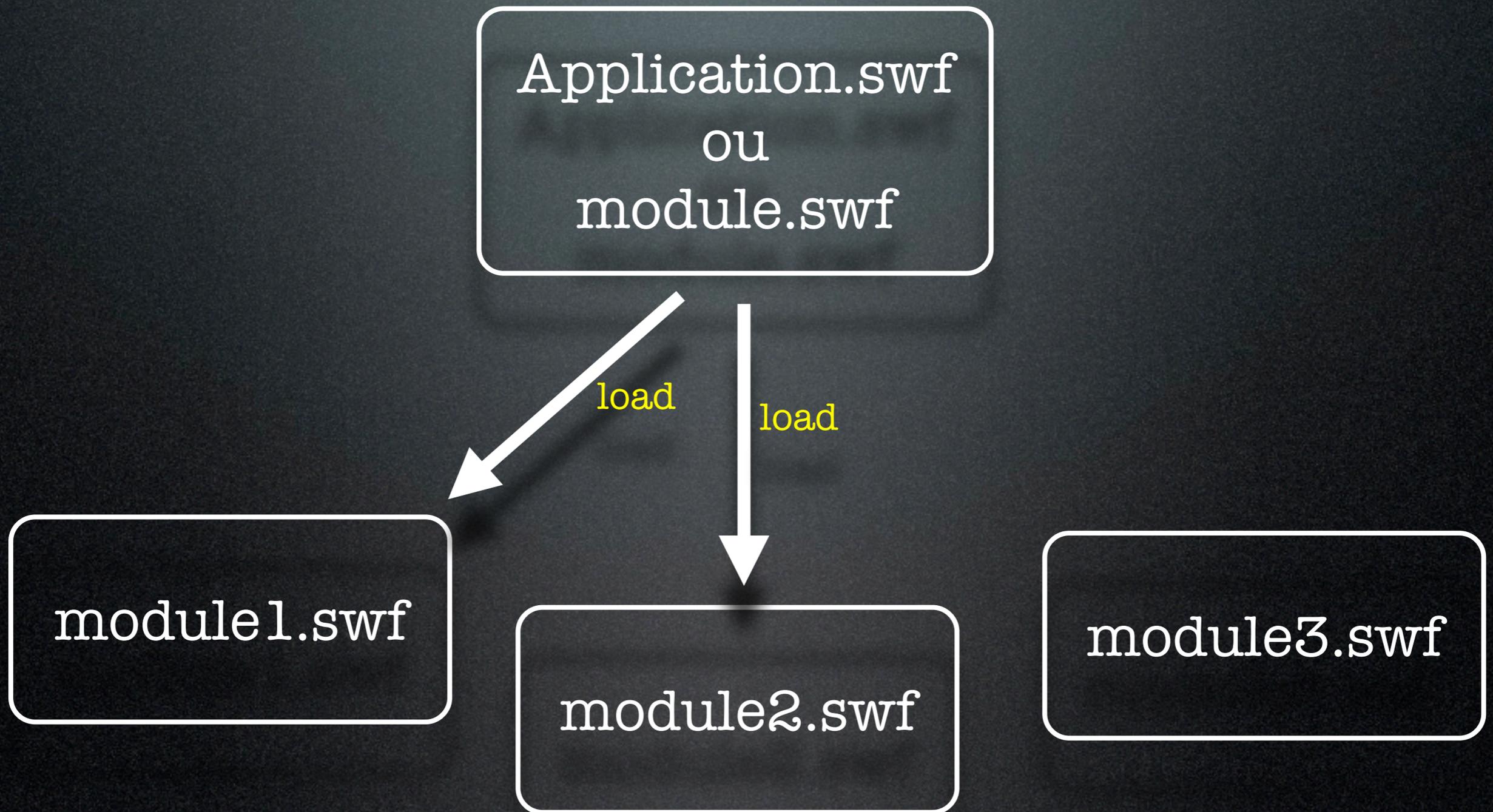
load

load

module1.swf

module2.swf

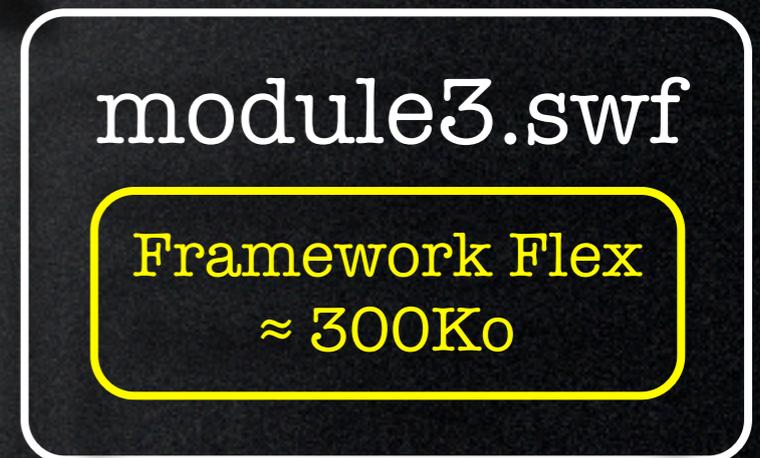
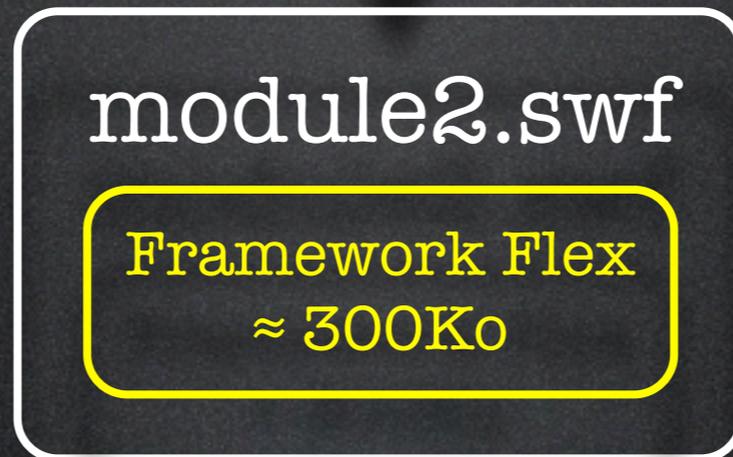
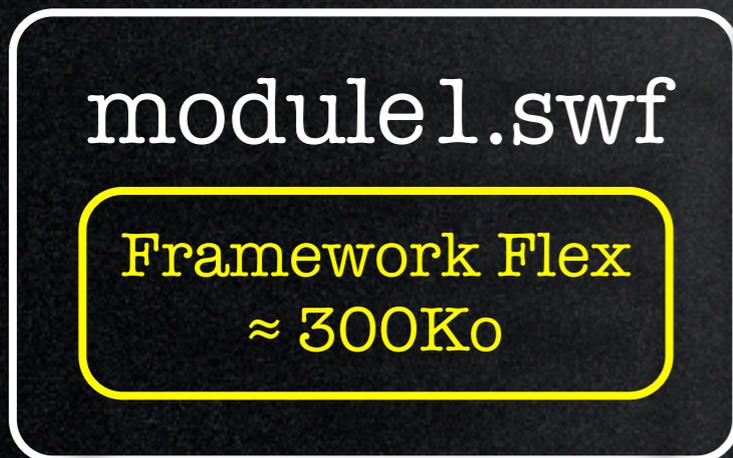
module3.swf



# Module

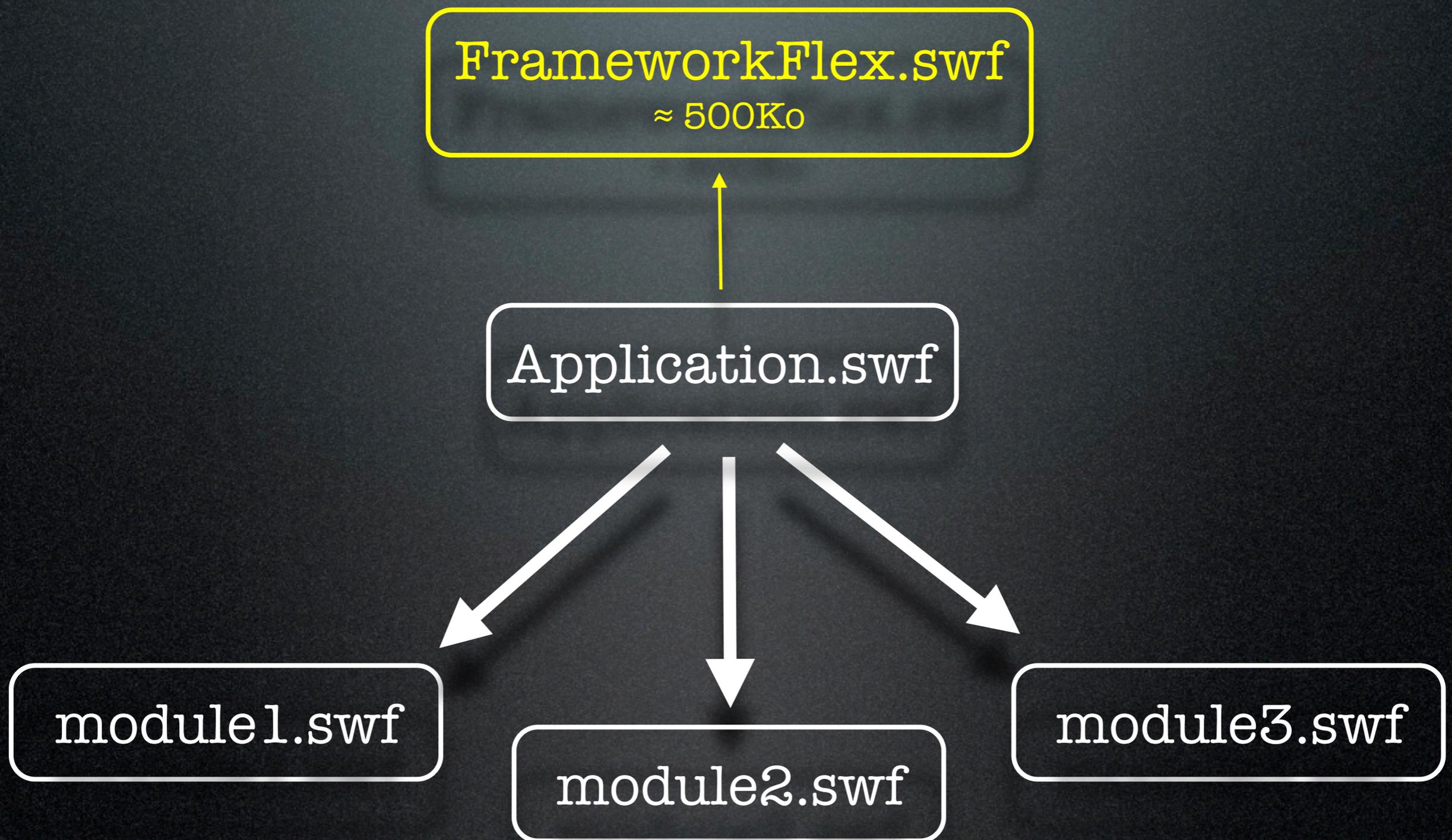
- Cairngorm : possible
- PureMVC : bonne gestion des modules

# Module



# Runtime Shared Libraries

# Runtime Shared Libraries



FrameworkFlex.swf

Application

Module1

Facade

Model

View

Controller

Module2

Facade

Model

View

Controller

Module3

FrameworkArchitecture.swf

# Multicore

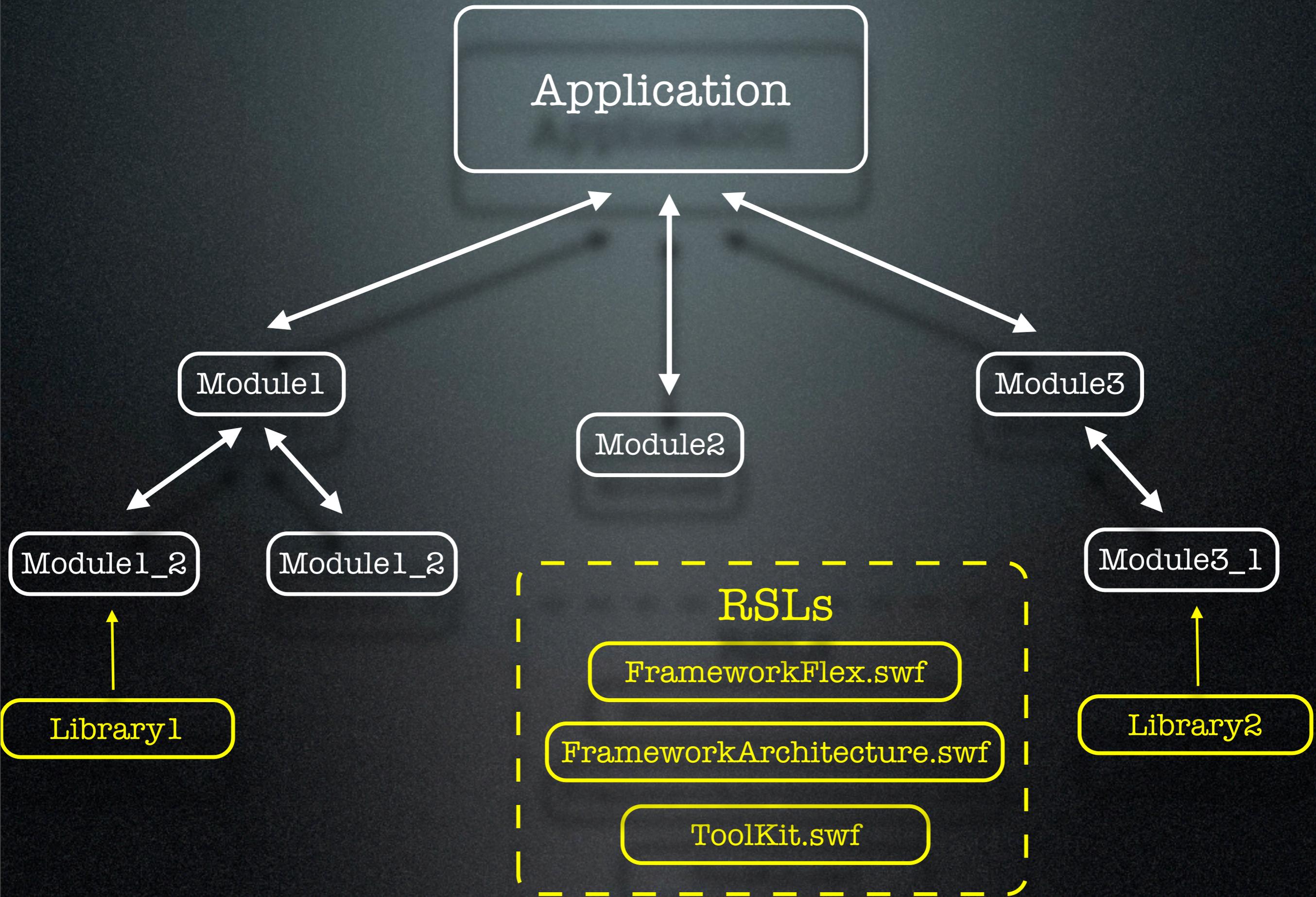
- Cairngorm : très difficile
- PureMVC : Multiton

# Multiton

```
public class Multiton
{
    private static var instances:Array = new Array();

    public static function getInstance(key:String):
    Multiton
    {
        if (instances[key] == undefined)
        {
            // Create instance and add it to map
            instances[key] = new Multiton();
        }

        return instances[key];
    }
}
```





- Utiliser un framework d'architecture

- Utiliser un framework d'architecture
- Choisir un framework adapté à ses besoins

- Utiliser un framework d'architecture
- Choisir un framework adapté à ses besoins
- Adapter un framework existant si besoin est

- Utiliser un framework d'architecture
- Choisir un framework adapté à ses besoins
- Adapter un framework existant si besoin est
- Séparer convenablement son application en composants indépendants

- Utiliser un framework d'architecture
- Choisir un framework adapté à ses besoins
- Adapter un framework existant si besoin est
- Séparer convenablement son application en composants indépendants
- Pour des applications de grande taille, choisir un framework gérant les modules et le multicore

- Utiliser un framework d'architecture
- Choisir un framework adapté à ses besoins
- Adapter un framework existant si besoin est
- Séparer convenablement son application en composants indépendants
- Pour des applications de grande taille, choisir un framework gérant les modules et le multicore
- Utiliser convenablement les RSLs

# Sources

- [labs.adobe.com](http://labs.adobe.com)
- [cairngormdocs.org](http://cairngormdocs.org)
- [lafabrick.com](http://lafabrick.com)
- [puremvc.org](http://puremvc.org)
- [kapit.fr](http://kapit.fr)
- [guasax.com](http://guasax.com)
- [mate.asfusion.com](http://mate.asfusion.com)
- [onflex.org](http://onflex.org)
- [as3dp.com](http://as3dp.com)



[www.Intellicore.net](http://www.Intellicore.net)

tous droits réservés Intellicore Services © 2008

<http://matthieusegret.com>