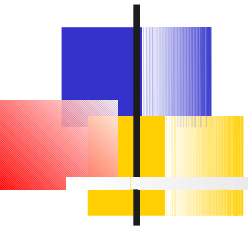
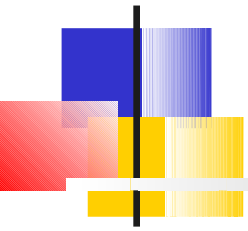


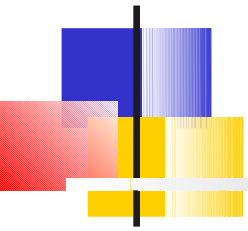
Le Multimédia dans les SID

Introduction



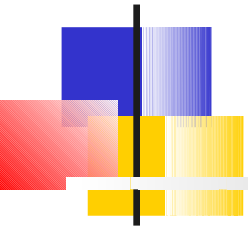
Le cours

- Fonctionnement
  - 21h TD réparties en 7 séances de 3h chacune
  - 1h30 cours, 1h30 pratique
  - 1 projet
    - Utiliser un Framework MVC Flex pour :
      - ❖ Visite virtuelle de musée
      - ❖ Avec une application sur téléphone mobile
      - ➔ soutenance le 12 décembre 2011



Le cours

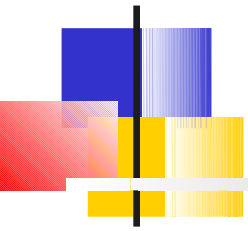
- Contenu
  - Création d'une A.I.R. (Application Internet Riche)
- Plan
  - Introduction
    - Exigences et problèmes du MM distribué
  - L'environnement de développement Flash Builder 4
    - Créer des applications Flex 4.5
      - MXML pour la vue : conteneurs, positionnement, style
      - AS3 pour le contrôle, transition entre vues, effets
    - Gérer des données
      - Modèle, dataBinding, validation des données
    - Alimenter Flex en données
      - Accéder à la couche métier : webservices
    - Créer des applications pour les mobiles
      - Ergonomie, navigation, déploiement



Introduction

Définition du mot multimédia

- A l'origine
  - Buzzword, apparu fin 80, lorsque les CD-ROM se sont développés
  - désignait les applications capables de gérer ou piloter différents médias simultanément : musique, son, image, vidéo
    - ➔ On parlait de produit multimédia
- Fin des années 90
  - Arrivée des méthodes de compression du son et de la vidéo
  - Croissance des ordinateurs personnels
    - ➔ systèmes de bonne qualité
    - ➔ On parlait de station ou poste multimédia : console de jeux…



Définition

- Aujourd'hui
  - Le multimédia s'est externalisé avec l'Internet
  - Le MM est vu comme une IHM qui associe
    - des informations d'origines diverses
      - avec réelle intégration et non une juxtaposition !!!
      - sur des serveurs dédiés
    - offrant à l'utilisateur la possibilité de les consulter de façon **interactive**
    - disponible sous forme numérique, sur Internet

➔ On parle d'Application Internet Riche (R.I.A.)

Autre classification

Off-line vs On-line

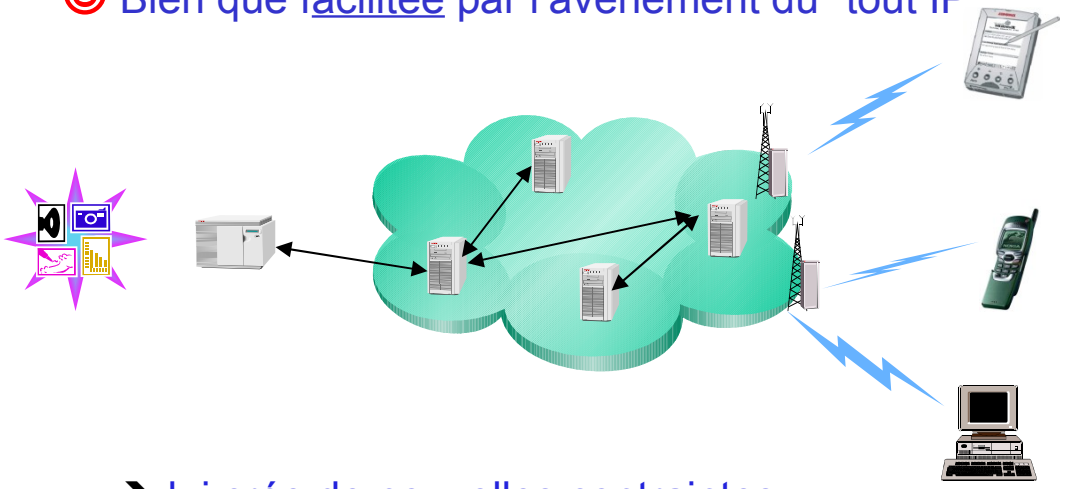
- Le Off-Line
  - C'est l'univers du DVD
    - ➔ On consulte un produit multimédia sur une station de travail individuelle
    - C'est à 65% des jeux
      - utilisation des techniques d'audiovisuel
      - prix de conception très élevé ~ long métrage
    - En plus des jeux, on trouve des encyclopédies, des CD de formation, des DVD d'entreprise

Autre classification
Off-line vs On-line

- Le on-line
  - C'est l'univers d'Internet
    ➔ On consulte des données distantes via le réseau
  - Connaît une évolution technologique constante, assurée par :
    - réseaux rapides (ADSL)
    - standards pour la représentation numérique des médias
  - Se veut être une application conviviale et efficace, avec :
    - édition et présentation de documents multimédia
    - adaptation du contenu à différents terminaux

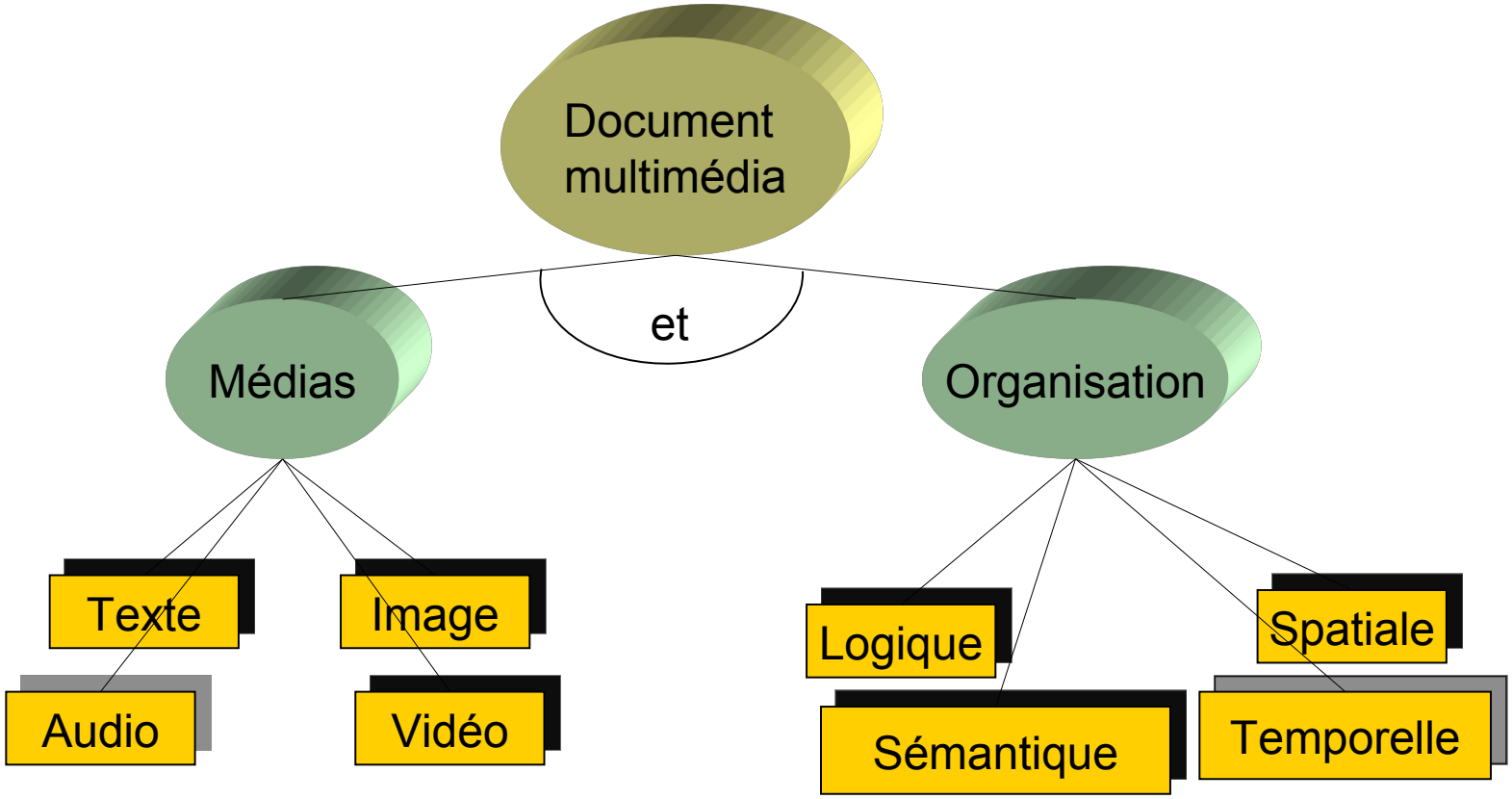
Le on-line

- De nouveaux défis se posent à lui :
  - ➔ Mise en place de systèmes multimédia adaptables
    - ☹ Rendue difficile par l'hétérogénéité des machines et des réseaux
    - ☺ Bien que facilitée par l'avènement du "tout IP"



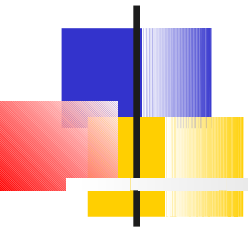
  - ➔ lui crée de nouvelles contraintes
    - ❖ Offrir la meilleure qualité de service pour le plus grand nombre de profils d'utilisateurs et de machines
    - ❖ Gérer en ***temps réel*** la restitution du contenu à l'utilisateur

C’est quoi un document multimédia ?



Caractéristiques spatiales et temporelles

Scénario d’implémentation



Document multimédia

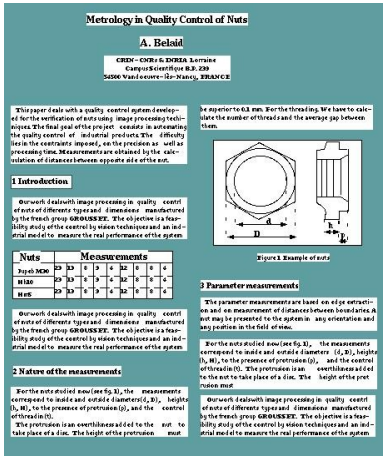
- Les médias
  - Briques de base
  - Leur caractéristique est de contenir l'information
    - Exemples : image fixe, son, etc.
    - Objet multimédia composite : vidéo, page web html
  - Ont des propriétés intrinsèques
    - Le texte est linéaire
    - L'image a une occupation surfacique
    - L'audio a une étendue temporelle
    - La vidéo a une étendue surfacique (par frame) et une étendue temporelle

Les médias

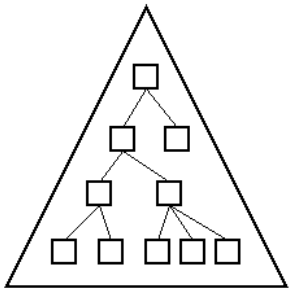
Les problèmes qu'ils posent dans un S.I.D.

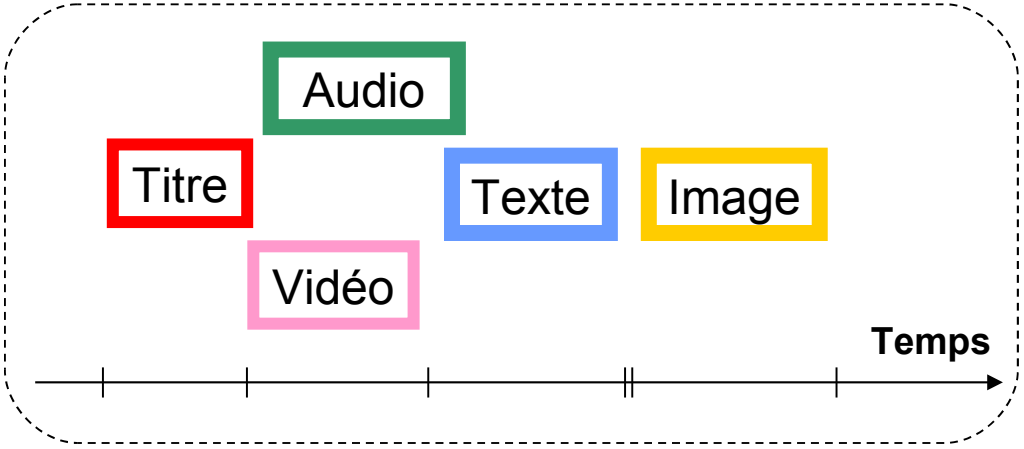
- *Transmission* :
  - ➔ ou comment assurer la restitution avec une qualité adaptée aux besoins et à l'environnement d'utilisation ?
- *Stockage* :
  - ➔ ou comment manipuler des masses d'information de plus en plus élevées ?
    - ❖ on est passé du méga-octets ou giga-octets, au terra-octets …!
- *Distribution* :
  - ➔ comment assurer la confidentialité et la qualité de service durant la distribution sur des réseaux avec contrôles ?
- *Hétérogénéité* :
  - ➔ quels standards (formats) utiliser pour la limiter ?

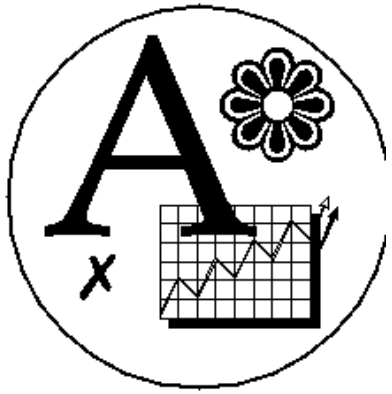
- L'organisation
  - Plusieurs types
    - ***Logique*** : organisation hiérarchique (contenu)
    - ***Spatial*** : style graphique et positionnement géométrique (présentation)
    - ***Sémantique*** : lien de navigation intra et inter-documents
    - ***Temporel*** : synchronisation entre objets multimédia

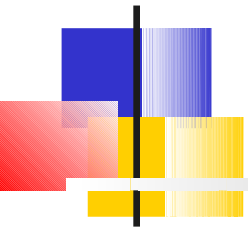

Metrology in Quality Control of Nuts

A. Belaid


STRUCTURE

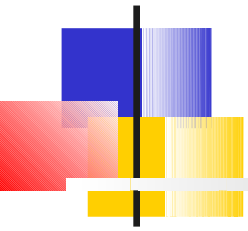



PRESENTATION



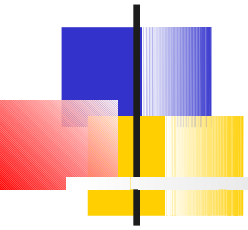
L'organisation

- Les problèmes qu'elle pose dans une A.I.R.
  - Modélisation :
    - L'information est hétérogène et complexe
    - ➔ Comment améliorer l'identification de contenus et permettre davantage de déclarativité dans la création de documents ?
  - Recherche :
    - Les données sont distantes, dans des bases de données
    - ➔ Comment faire des recherches de contenu à la volée ?
  - Expressivité :
    - Une application MM nécessite d'être scénarisée
    - ➔ Quel langage utiliser permettant de déclarer des contenus multimédia, et mettre en scène un scénario ?



La solution

- Une IHM
  - Si elle est bien pensée, elle peut apporter des solutions pour :
    - l'affichage visuel de données différentes
    - la création d'effets
    - l'interaction avec des données qui évoluent dans le temps
  - sans altération ou ralentissement dû à
    - la volumétrie
    - l'éloignement des ressources, avec les problèmes de retard et de perte d'information pendant le transport
  - Sur n'importe quelle architecture existante, la plus adaptée possible
    - Client léger, lourd…



Comment faire une bonne IHM ?

Tendances technologiques qui ont émergé

- Point sur la situation actuelle
  1. Apparition de nouveaux acteurs innovants, comme Adobe et Google dans le monde des IHM Web
     - venus concurrencer les acteurs historiques :
       - Sun (Java/Swing, les API Java EE, les framework MVC type Struts …)
       - Microsoft (HTML, XHTML, CSS, …)
     - proposent des solutions innovantes et standards
       - Google Web toolkit (framework pour le développement de pages Web dynamiques en utilisant la techno AJAX : JS + XML + CSS)
       - Flex pour Adobe

Tendances technologiques qui ont émergé
Retour du client lourd

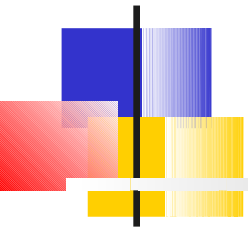
2. L'alourdissement du client léger, avec l'émergence du « client serveur web »
   - du traditionnel modèle web « server-side »
     - (c'est-à-dire page générée côté serveur et échange de flux HTML entre navigateur et le serveur)
   - on revient au modèle du client lourd où :
     - seules les données sont échangées entre le navigateur et le serveur

     ➔ la gestion de l'IHM est intégralement déléguée au navigateur

   ➔ Ceci impacte sur deux autres enjeux liés à la sécurité et la performance

Tendances technologiques qui ont émergé
Retour du client lourd

3. La sécurité
   - déplacement des responsabilités d'authentification et d'habilitation au niveau du navigateur
     - ➔ Solution : limitation des points d'entrée permettant l'accès aux données, appels asynchrones
4. Les performances
   - Gestion minutieuse de l'échange de données avec le serveur
     - ➔ Demander moins de données au serveur
       - ➔ Demander uniquement les enregistrements et les champs dont vous avez besoin
     - ➔ Créer et gérer vous-mêmes des tables d'index locales
     - ➔ Réduire le nombre de contrôles dépendants, comme dans les formulaires
       - ➔ Lorsqu'un formulaire est ouvert, chacun de ses contrôles envoie une requête distincte au serveur



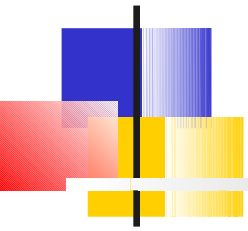
Tendances technologiques qui ont émergé
Retour du client lourd

5. Les IHMs « vectorielles » ont débarqué en force :
   - Flex, SilverLight, JavaFX et… HTML5
     - L'avenir de l'IHM se joue ici
   - proposent des modes de développement plus efficaces pour les développeurs
     - déclaratifs, frameworks,...

       ➔meilleure intégration design / code…

Tendances technologiques qui ont émergé
Retour du client lourd

6. HTML5 pointe le bout de son nez
   - Après une longue période de sommeil et sous l'impulsion d'acteurs pour lesquels HTML est un critère de survie (Google notamment), la norme HTML recommence à bouger avec
     - la normalisation des APIs permettant de gérer du déconnecté
     - la gestion de la vidéo
     - le support d'API de dessin vectoriel
     - la gestion de tâches de fonds dans le navigateur
       - (ce qui permettra par exemple d'améliorer les performances des applications HTML / Javascript)
     - l'apparition d'APIs orientées terminaux mobiles (gestion du GPS par exemple)



Les innovations

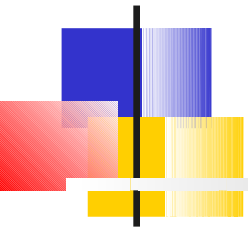
1. Le « push Web »
   - qui consiste à pouvoir diffuser (pousser) des données/événements à l'ensemble des clients connectés
     - banale en client/serveur,
     - beaucoup plus complexe sur un protocole déconnecté comme HTTP et dans des architectures Web où les connexions sont à l'initiative des clients...

Les innovations

2. L'utilisation nomade
   - de plus en plus fréquente dans nos usages du web
   - cela passe bien entendu par
     - l'évolution des « devices »
       - Ex : l'iPhone
     - par le support du déconnecté par les navigateurs
       - Ex : HTML5 !
     - par l'expansion du "connect everywhere"
       - capter des bornes Wifi à peu près partout

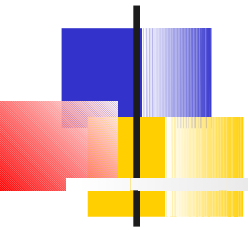
Les innovations

3. les IHM mobiles connaissent une nouvelle jeunesse,
   - Sous l'impulsion de l'iPhone,
     - premier mobile à proposer une usabilité mobile intuitive
     - un catalogue d'applications dédiées aux petits écrans et des interactions multi-touch voient le jour



Panorama des technologies d'IHM Web

…favorables au multimédia



Technologies d'IHM

- Vision classique de la techno !
  - L'évolution technologique actuelle a fait explosé la vision classique qui opposait
    - « applications web »
      - RIA – Rich Internet Applications
    - aux « applications client lourd »
      - RDA – Rich Desktop Applications

  ➔ Il est désormais tout à fait possible d'utiliser des technologies initialement dédiées aux RIA pour développer des applications RDA
- La Classification peut donc se faire suivant ces deux axes
  - Environnement d'exécution (navigateur ou pas)
  - Modèle de programmation

Classification des IHM

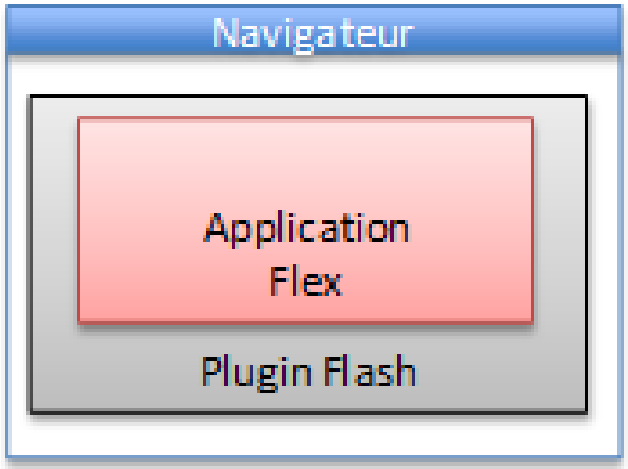
Par l'environnement d'exécution

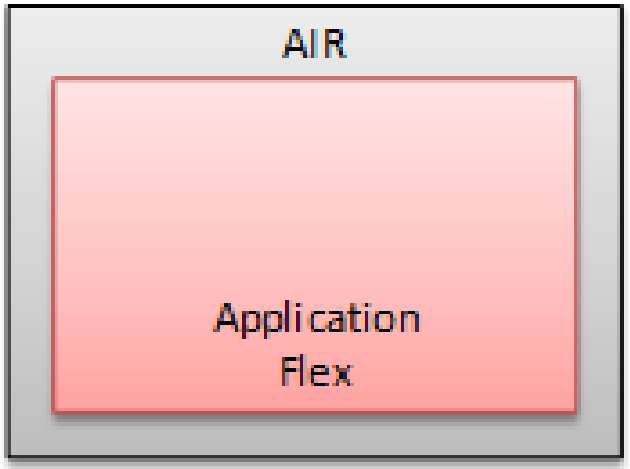
- 3 types
  1. Un simple navigateur Internet :
     - l'IHM, dite intégrée : « **Browser Rendered** », est intégralement gérée et affichée par les moteurs HTML et JavaScript du navigateur
  2. Une extension ou un plugin du navigateur :
     - l'IHM, dite embarquée : « **Browser Plugin Rendered** », gérée par cette extension elle-même hébergée au sein du navigateur
  3. Un runtime autre que le navigateur (par ex. une machine virtuelle) :
     - l'IHM, dite personnalisée « **Custom Rendered** »
       - On communique avec le serveur (via par ex. http) mais le navigateur web n'intervient pas

Classification des IHM

Par l'environnement d'exécution

- 2 exemples
  1. La plateforme AIR + plugin navigateur, proposée par Adobe, permet d'exécuter le même code en mode embarqué (browser plugin rendered) et en mode machine virtuelle (custom rendered)



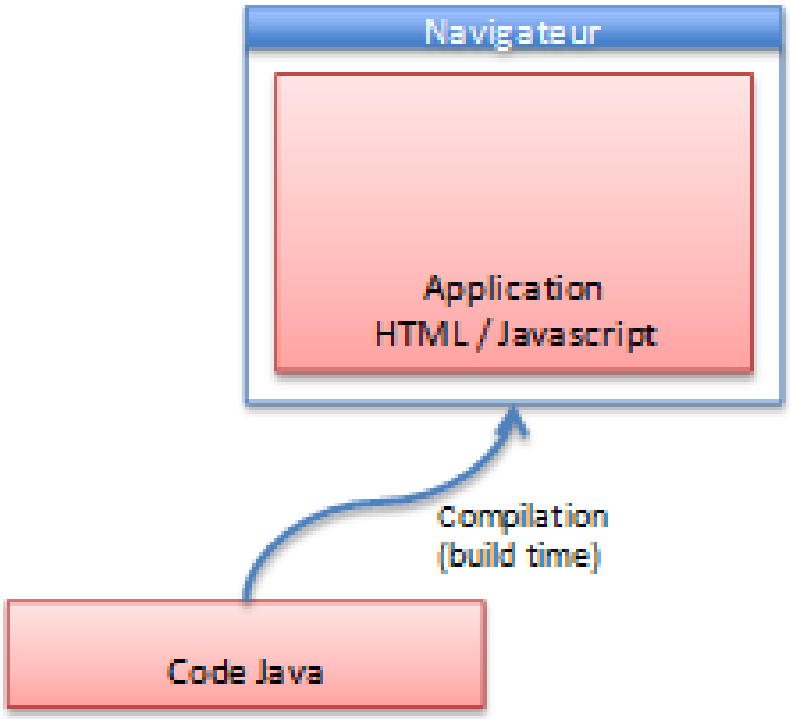


- Attention : les APIs disponibles dans AIR sont plus riches et donc une application qui tourne sur AIR ne tournera pas forcément dans le plugin… (la réciproque étant vraie)

Classification des IHM

Par l'environnement d'exécution

2. Google Web Toolkit
   - propose un modèle de développement en Java qui est compilé (transformé en fait) en code Javascript
   - Au runtime, seul du Javascript est exécuté par le navigateur

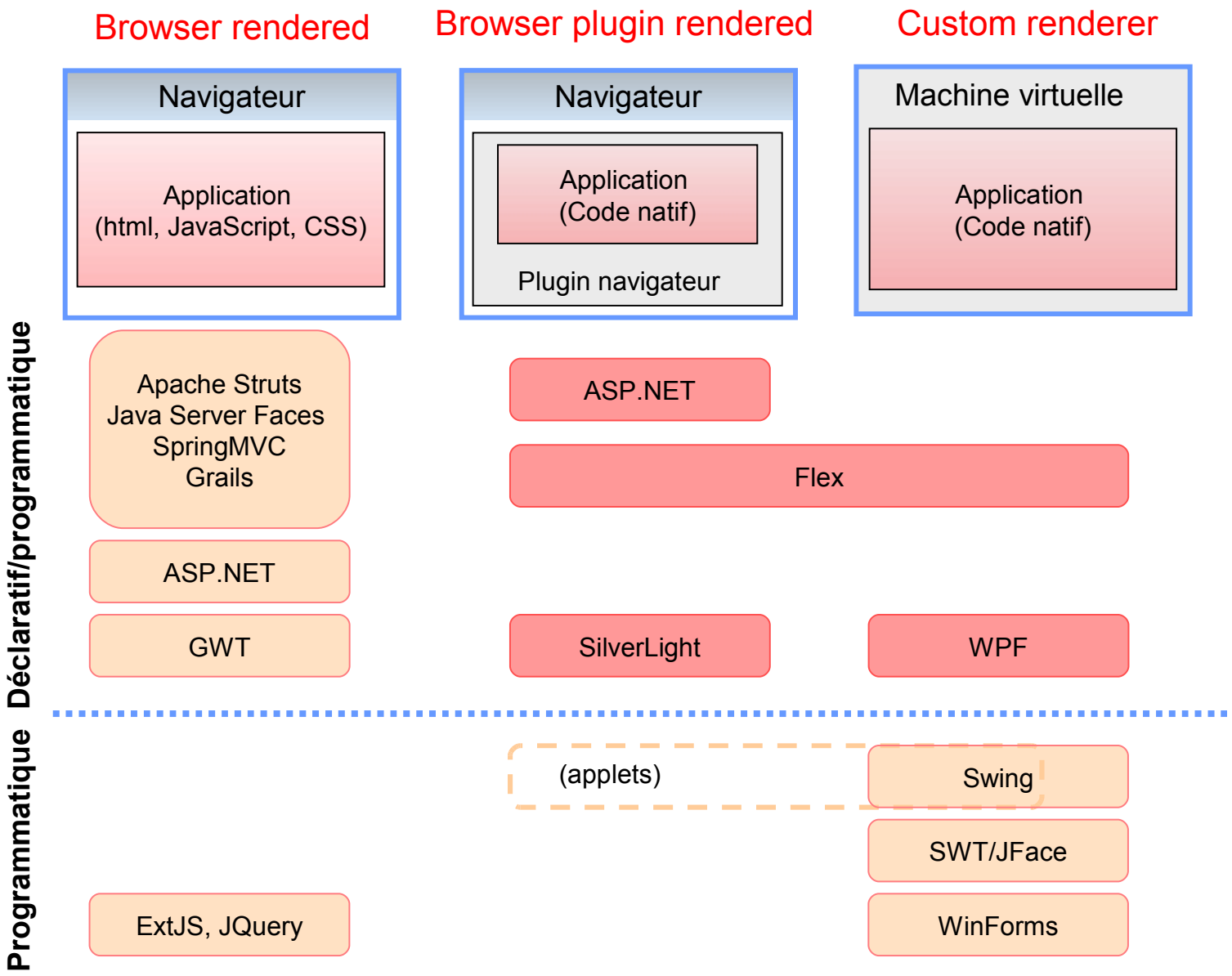


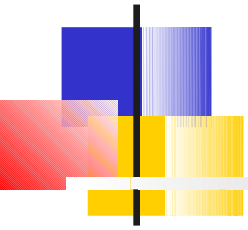
Classification des IHM

Par le modèle de programmation

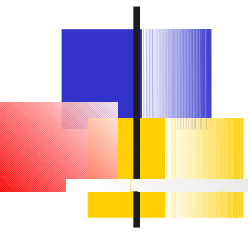
- Deux types
  - Modèle mixte (descriptif et programmatique)
    - La description de la page (ce que l'utilisateur voit) et le codage des comportements (ce que l'utilisateur peut faire) se font via des langages distincts
    - Par exemple : HTML + JavaScript
  - Modèle totalement programmatique
    - Pages et comportements sont codés via un langage unique
    - Par exemple : Java

Offres de frameworks d'IHM disponibles (en rouge les technologies vectorielles)



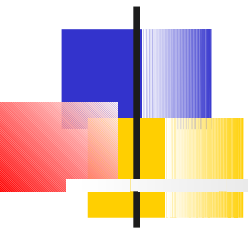


Les frameworks d'IHM



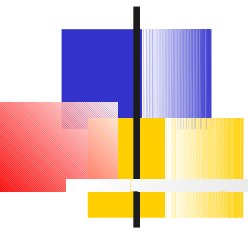
Les frameworks du monde
Javascript

- On peut distinguer 3 approches et 3 frameworks leaders en JS
  - L'approche décoration du HTML
    - avec des attributs permettant une interprétation du DOM par du code Javascript (Ex : Dojo)
  - La manipulation du DOM HTML en Javascript
    - HTML pour la présentation et
    - JS pour la manipulation des données (Ex : JQuery)
  - L'approche totalement Javascript :
    - plus de HTML, tout est fait en Javascript, avec des objets qui correspondent aux éléments du DOM (Ex : ExtJS)



Les frameworks du monde
Java

- Vus sous l'axe « Browser Rendered »
  - Axe historique du développement web en Java, avec
    - Apache Struts comme premier standard de fait,
    - Java Server Faces, désormais la solution estampillée « standard JEE »
    - et SpringMVC comme challenger
  - Ces technologies se caractérisent par
    - un modèle de programmation «descriptif & programmatique» :
      - un langage évolué enrichissant le HTML (Java Server Pages ou Facelet)
      - et du code Java implémentant les comportements



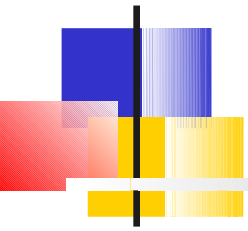
Les frameworks du monde

Java (suite)

- Google Web Toolkit
  - propose un compilateur Java vers JavaScript ainsi qu'un environnement de développement complet
- Grails
  - *framework open source* de développement d'applications web basé sur le langage Groovy
  - Groovy
    - utilise une syntaxe très proche de Java, avec des accolades,
    - a une gestion native des langages de balisage comme XML et HTML

Les frameworks du monde
Flex

- Vus sous l'axe « Browser Plugin Rendered »
  - Deux produits en compétition, positionné dans le monde Java
    - Flex et son plugin Flash
      - MXML + ActionScript
    - JavaFX :
      - C'est du Java pour faire des RIA. Il comprend :
        - JavaFX Runtime permettant aux applications JavaFX de s'exécuter sur toutes les plateformes disposant de Java SE ou Java ME
        - Il offre la possibilité d'avoir un client riche sur de multiples plateformes, écrans et terminaux comme par exemple : JavaFX Desktop, JavaFX Mobile et JavaFX TV
        - JavaFX Tools Suite : un ensemble d'utilitaires utilisables par les concepteurs, designers et développeurs de Web

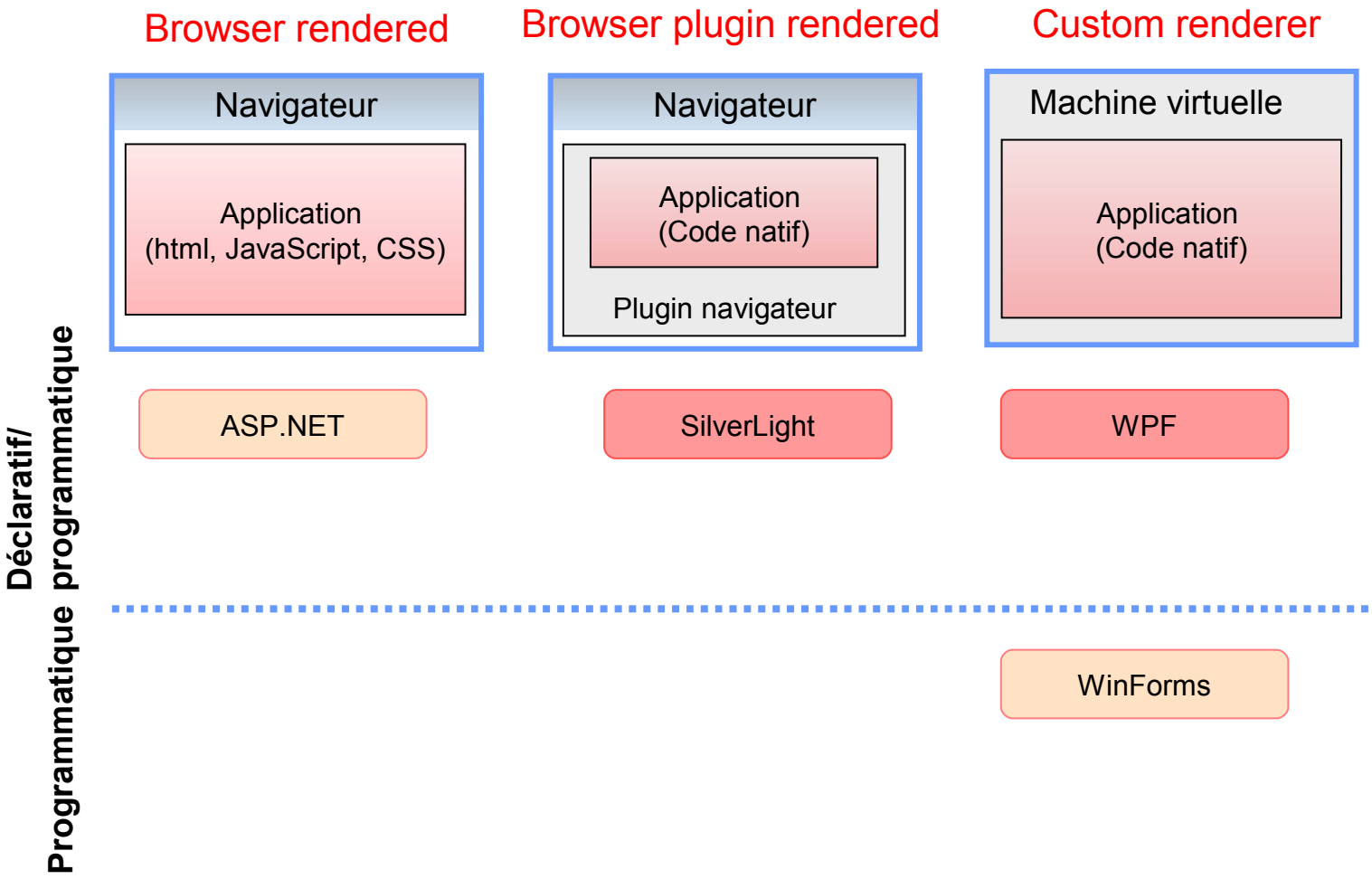


Les framework d'IHM

- Flex et Java
  - Les deux technologies proposent une offre comparable :
    - un SDK Java, SWT ou Swing pour le graphique
    - une couche de framework de plus haut niveau facilitant les développements, notamment par l'ajout de composants graphiques plus évolués ou bien systématiquement re-développés (menu…)
      - JFace pour SWT
      - des librairies plus éparses pour Swing (SwingX, JGoodies par exemple)
    - un « Rich Client Platform » = un socle de développement et de déploiement d'applications
    - des IDE tels qu'Eclipse ou NetBeans

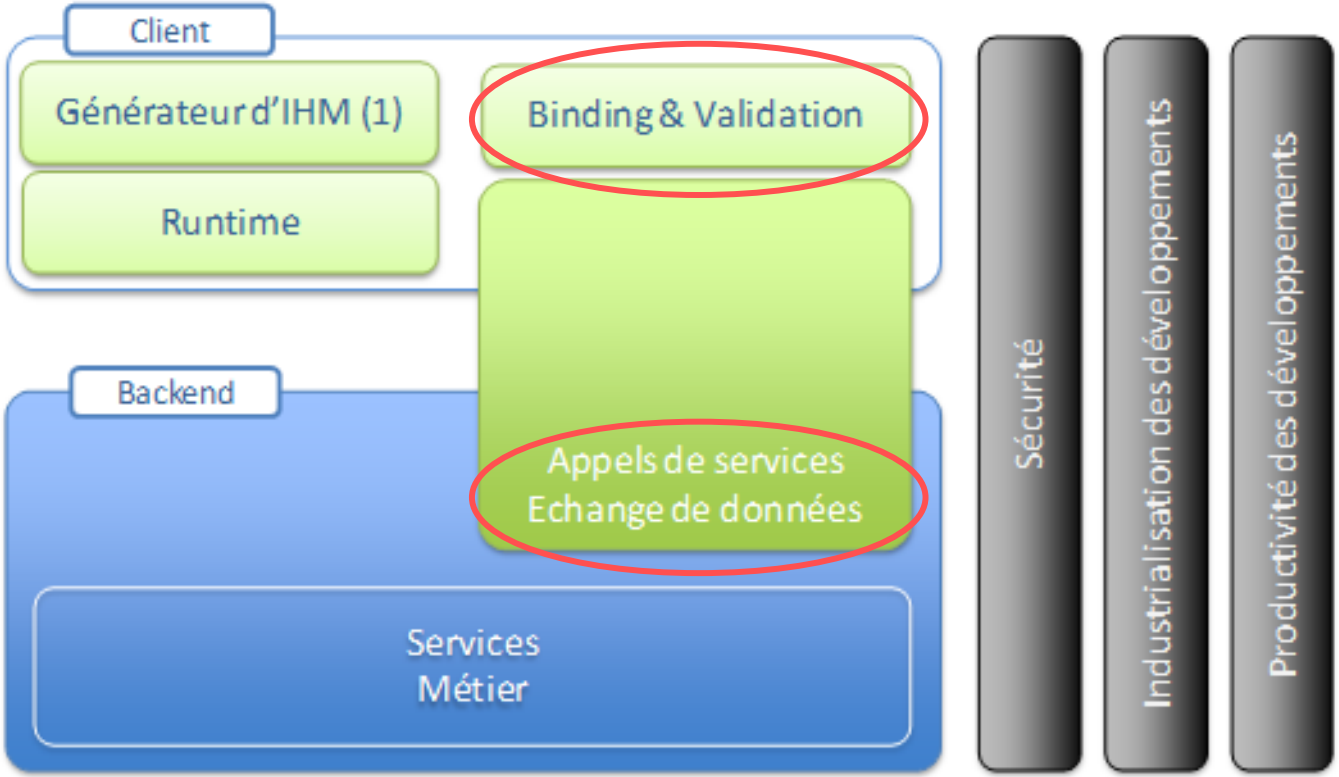
Les framework d'IHM

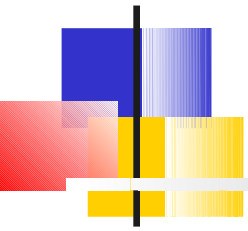
- A titre d'information : le monde .NET (non développé ici)



Les framework d'IHM

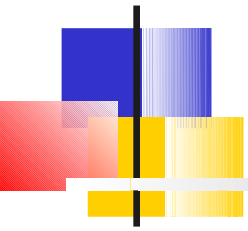
- Un petit mot sur les composants principaux d'une IHM
  - utiles au multimédia





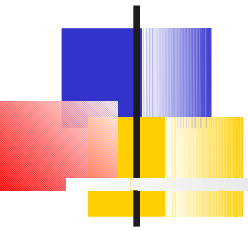
Les framework d'IHM

1. Binding & Validation
   - assure un mapping bidirectionnel entre un objet (au sens du codage) et un écran
     - ➔ meilleure séparation des problématiques graphiques (codage de la page), comportementales (codage des comportements), de données (stockage des données)
       - ➔ en bref, le pattern MVC
     - ➔ meilleure automaticité de la mise à jour et la récupération des saisies des utilisateurs depuis les écrans et les composants graphiques
       - ❖ d'où meilleure productivité des développements



Les framework d'IHM

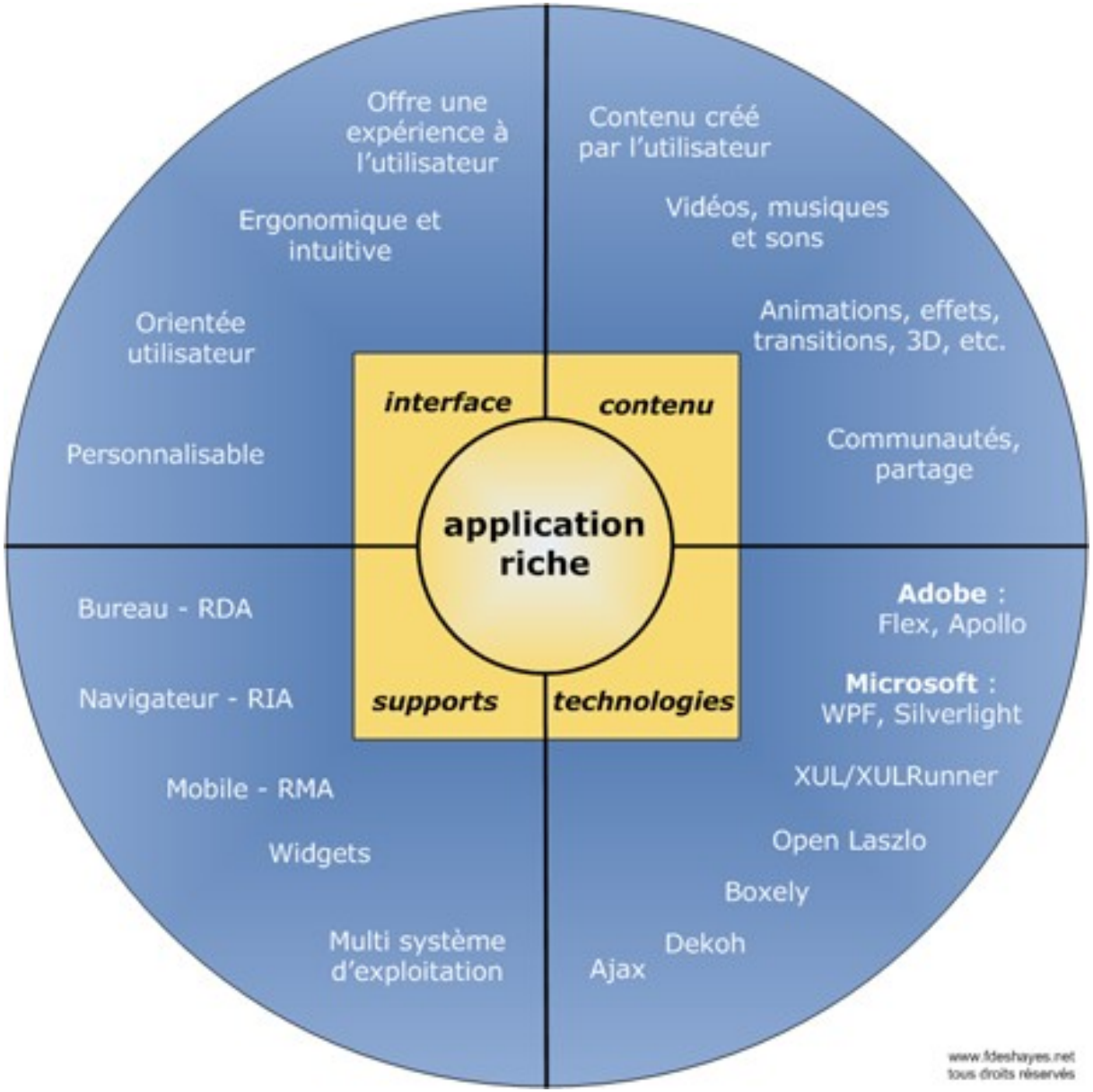
- ➔ meilleure gestion des erreurs ou validation des saisies
  - ❖ Exemple : en utilisant un masque ou format de saisie, lorsqu'à l'exécution, une saisie ne comporte pas le format voulu, le framework de binding sait remonter une erreur utilisateur intelligible
- ➔ Plus grande efficacité et richesse des mécanismes d'association entre les données et le widget graphique
  - ❖ On peut faire une association riche entre structures complexes

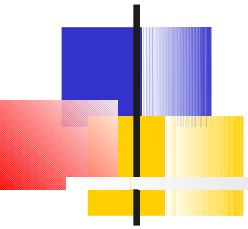


Les framework d'IHM

2. Appel de services et échange de données

- Les divers enjeux concernent :
  - Appels navigateur/serveur
    - La capacité à réaliser des appels asynchrones
    - La capacité à faire du « push » depuis le serveur, c'est-à-dire à notifier un ou plusieurs clients d'un événement serveur
  - Mécanismes de « Stubing »
    - La facilité de sérialisation et de désérialisation des structures échangées





L'architecture MVC des framework

© http://eric.groise.com/?/Usability/Le-retour-du-client-lourd

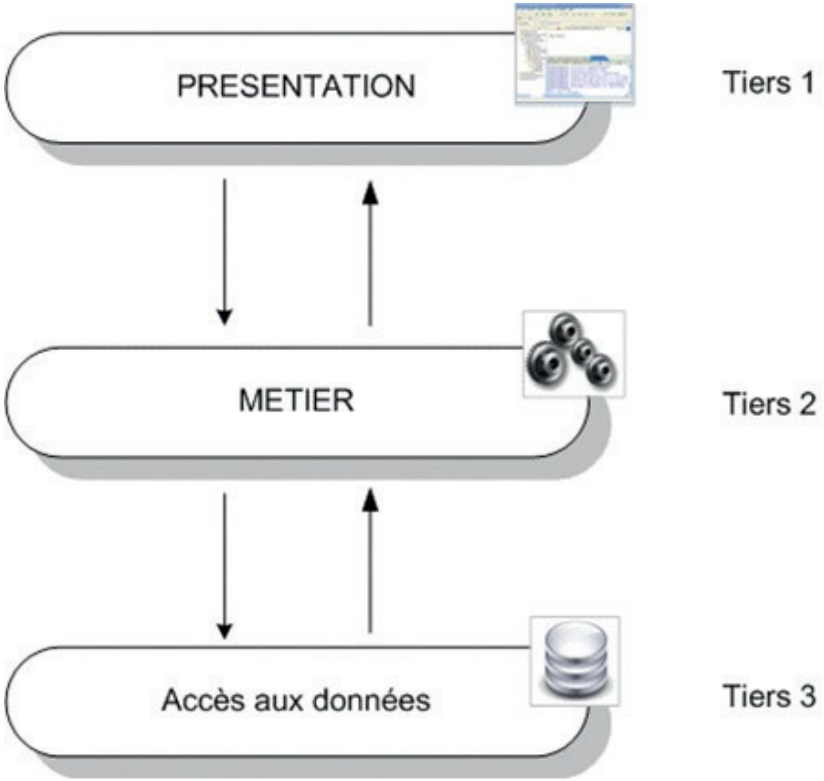
Architecture 3-tiers

- Flex suit une architecture 3-tiers : 3 couches

➔ chargée de présenter les données (IHM)

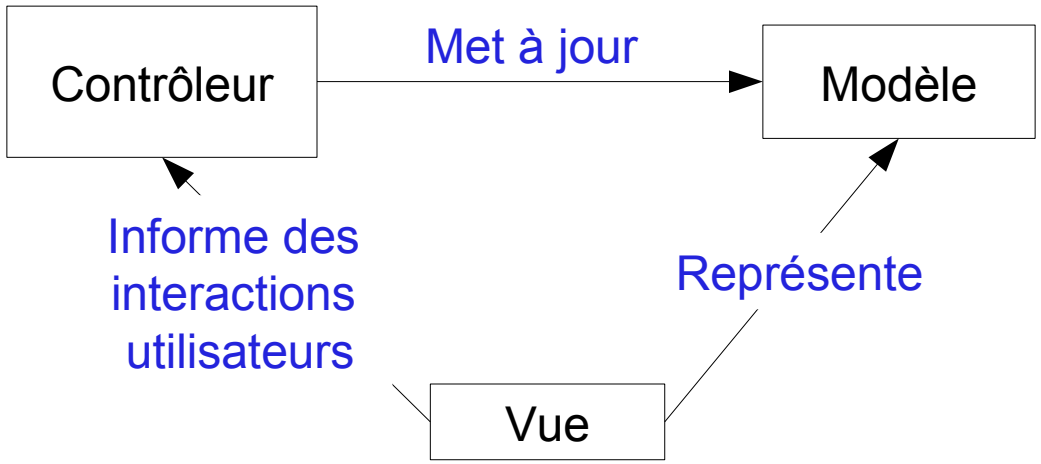
➔ implémente la logique métier (règles de gestion, etc.)

➔ réalise la tâche de persistance des données (stockage) via une BD, par ex.



Architecture 3-tiers

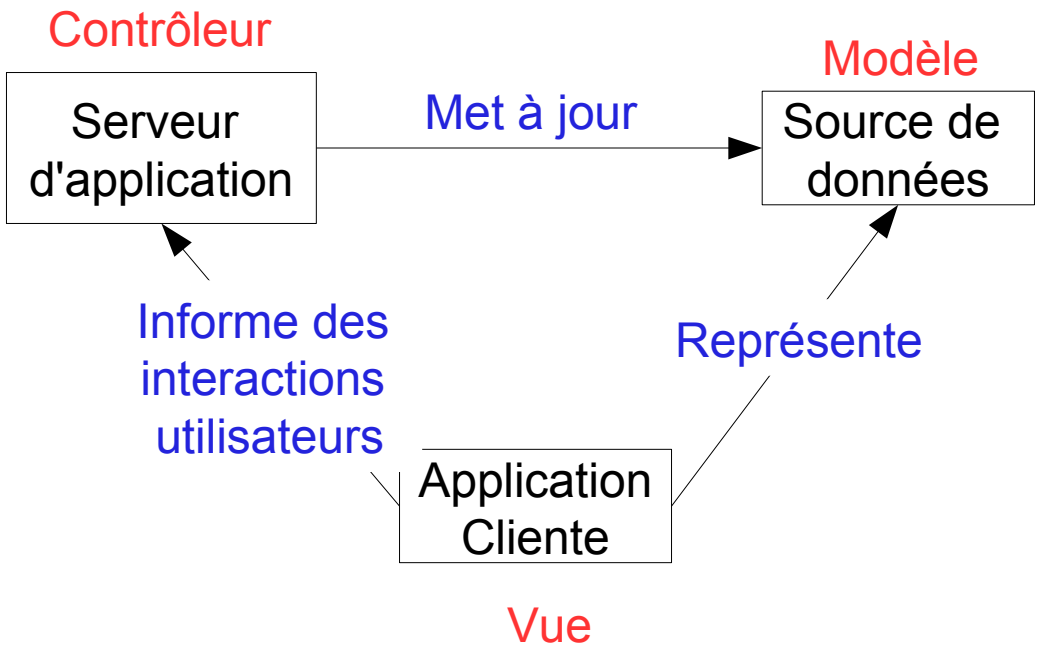
- L'architecture 3-tiers classique
  - Est composée de 3 éléments essentiels assimilés au modèle de conception MVC (pattern MVC)



http://www.flashxpress.net/ressources-flex/les-fondamentaux-de-l%E2%80%99architecture-flex-episode-1/

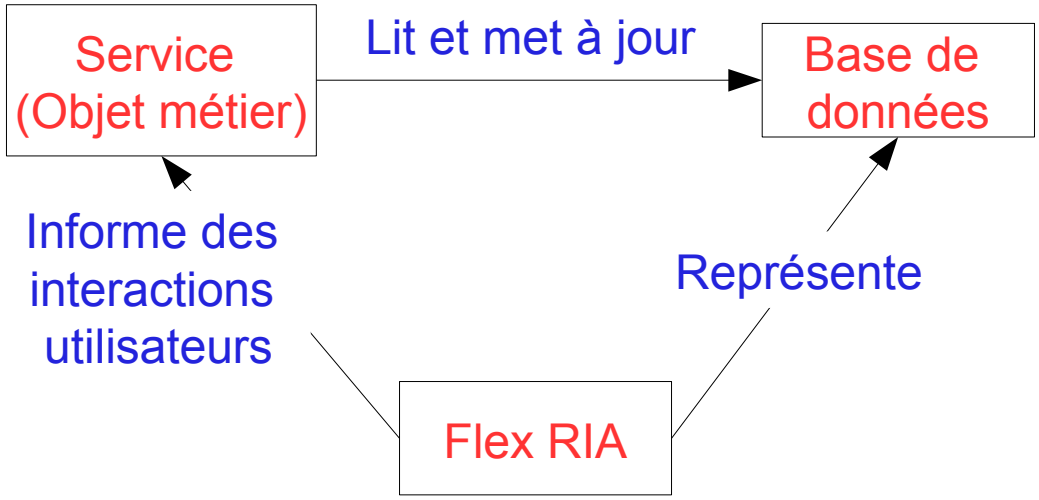
Architecture 3-tiers

- Dans un contexte d'application Web
  - Les composants ont un rôle plus adapté



Architecture 3-tiers

- Dans le cas d'une RIA Flex
  - Le contrôleur peut prendre la forme d'un service (ou d'un objet métier) sur le serveur d'application
  - Celui-ci lit et met à jour la base de données (le Modèle) quand il en reçoit la demande de la part de l'application cliente Flex (la Vue) qui, elle-même, affiche la Base de données à l'utilisateur



Architecture 3-tiers

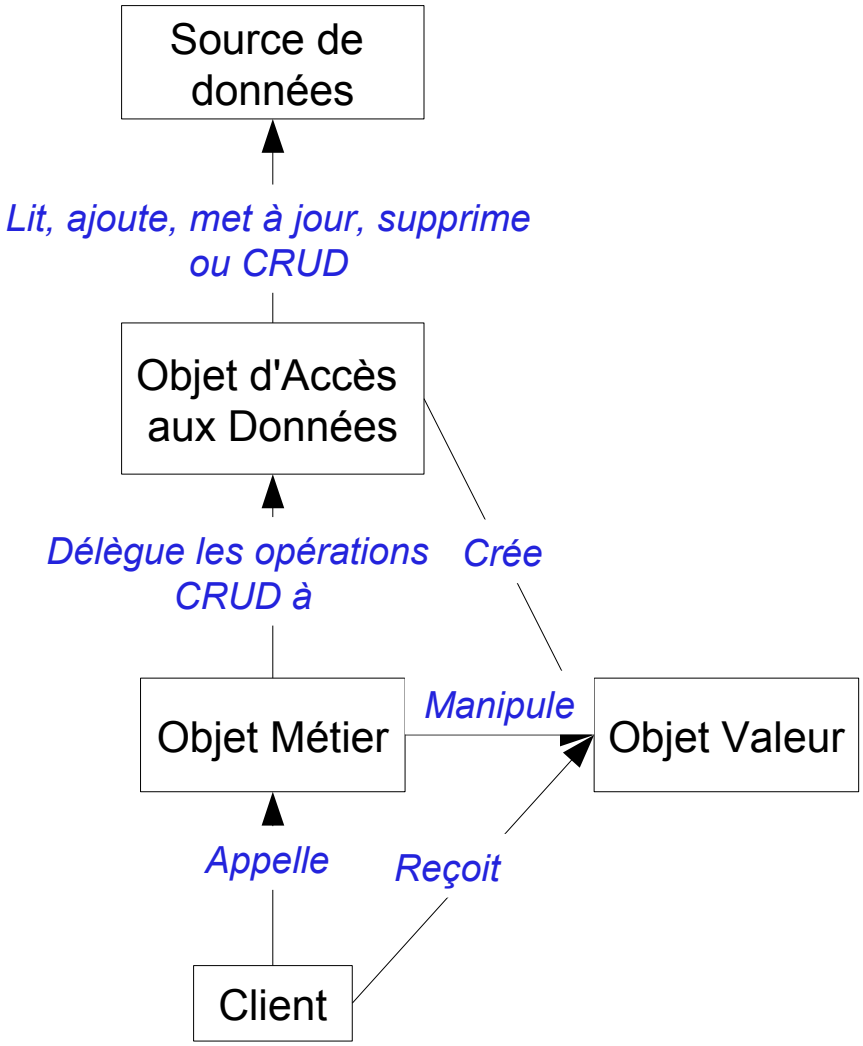
- Le serveur d'application
  - Son architecture peut varier d'une simple page ou script à une architecture complexe
  - On utilise souvent un modèle de conception (pattern) composé d'objets de gestion de données
    - Data Access Object – DAO

  ➔i.e. le service délègue à des objets autonomes la gestion de l'accès aux données pour réaliser les opérations, type CRUD (Create, Read, Update, Delete)

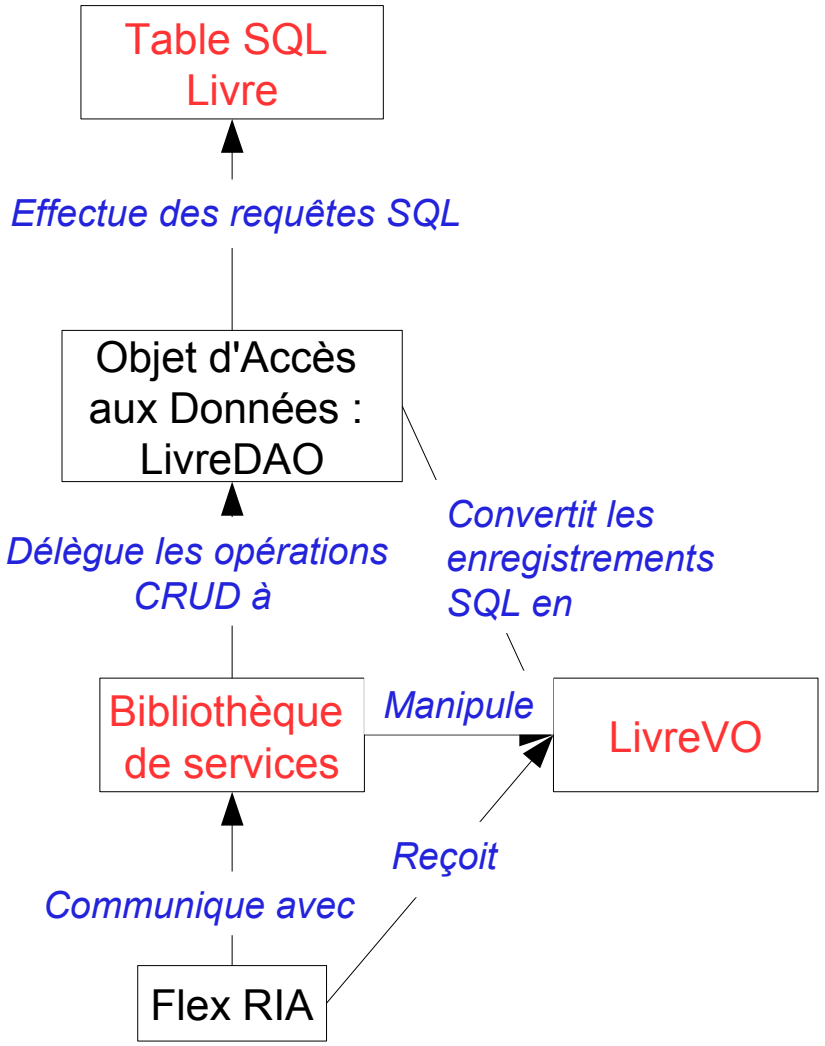
  ➔Ces objets retournent les résultats sous forme d'objets représentant les enregistrements de la BD, appelés VO (Value Objects)

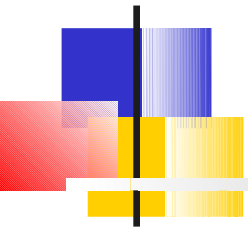
Le modèle de conception : Objet d'accès aux données

Représentation abstraite



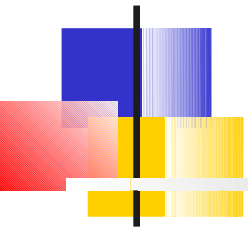
Représentation dans le monde réel





Le modèle de conception

- Objet d'accès aux données
  - La communication Flex est plus simple que dans d'autres framework
  - le client Flex devra communiquer avec un service distant qui attendra et retournera notamment des Value Objects
  - Si vous êtes développeur Flex, vous aurez seulement à connaître l’adresse du service ainsi que l’API qui contient les méthodes et parfois des Value Objects

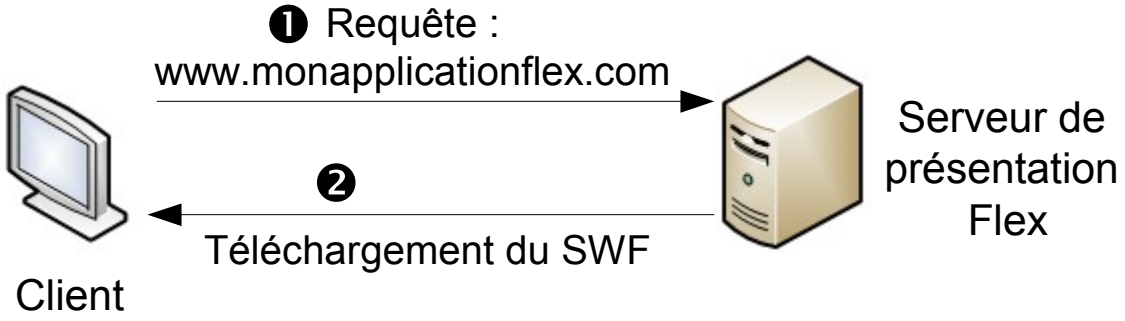


Flex

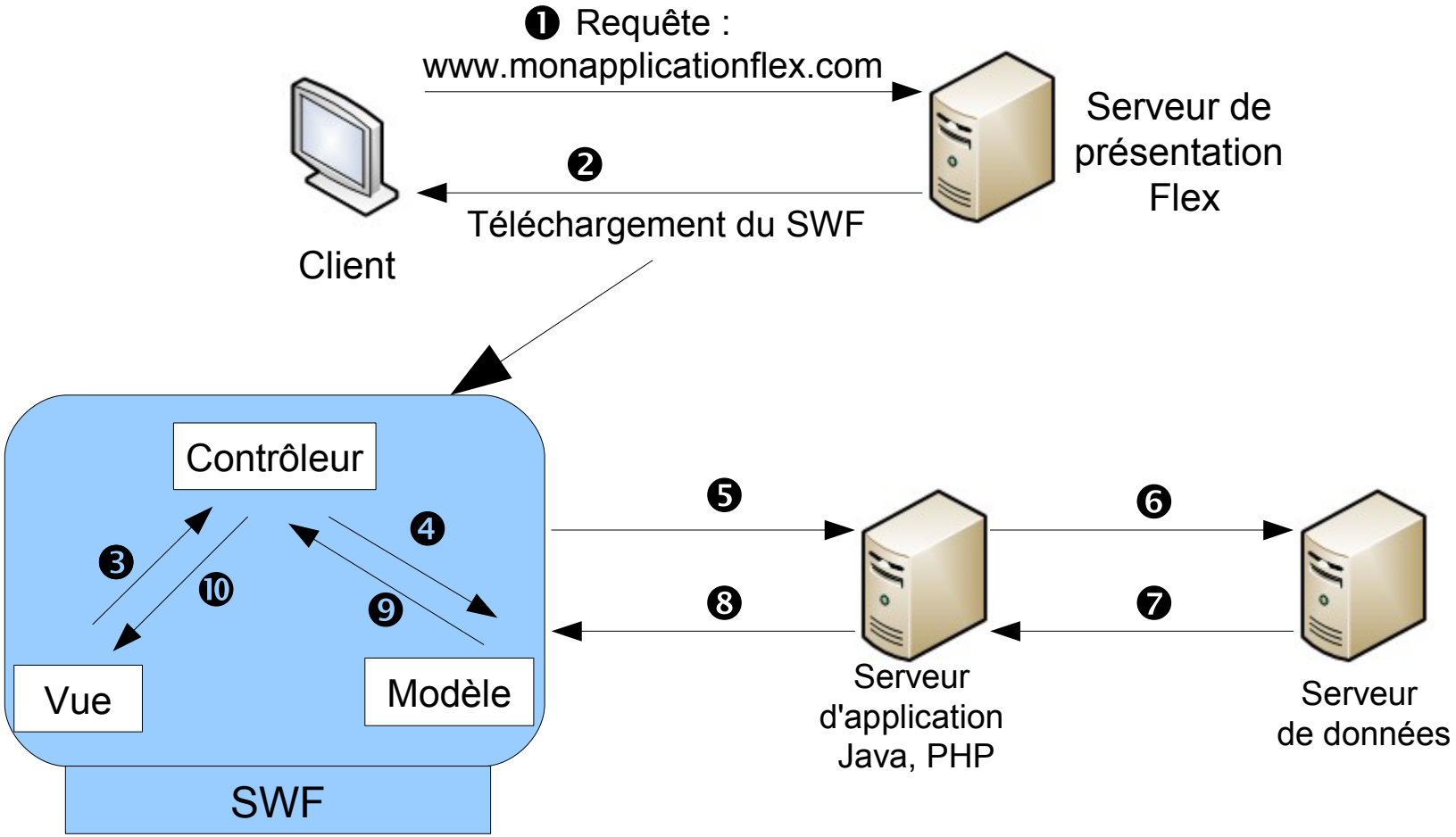
- Le mécanisme client/serveur
  - il est possible d'architecturer une application en trois couches distinctes, chacune d'entre elles pouvant être hébergée (sous sa forme extrême) sur trois serveurs différents :
    - le serveur de base de données (couche Accès aux données) ;
    - le serveur d'application (couche Métier) ;
    - le serveur Flex (couche présentation contenant le fichier SWF et la page web associée)

Le mécanisme Client/Serveur

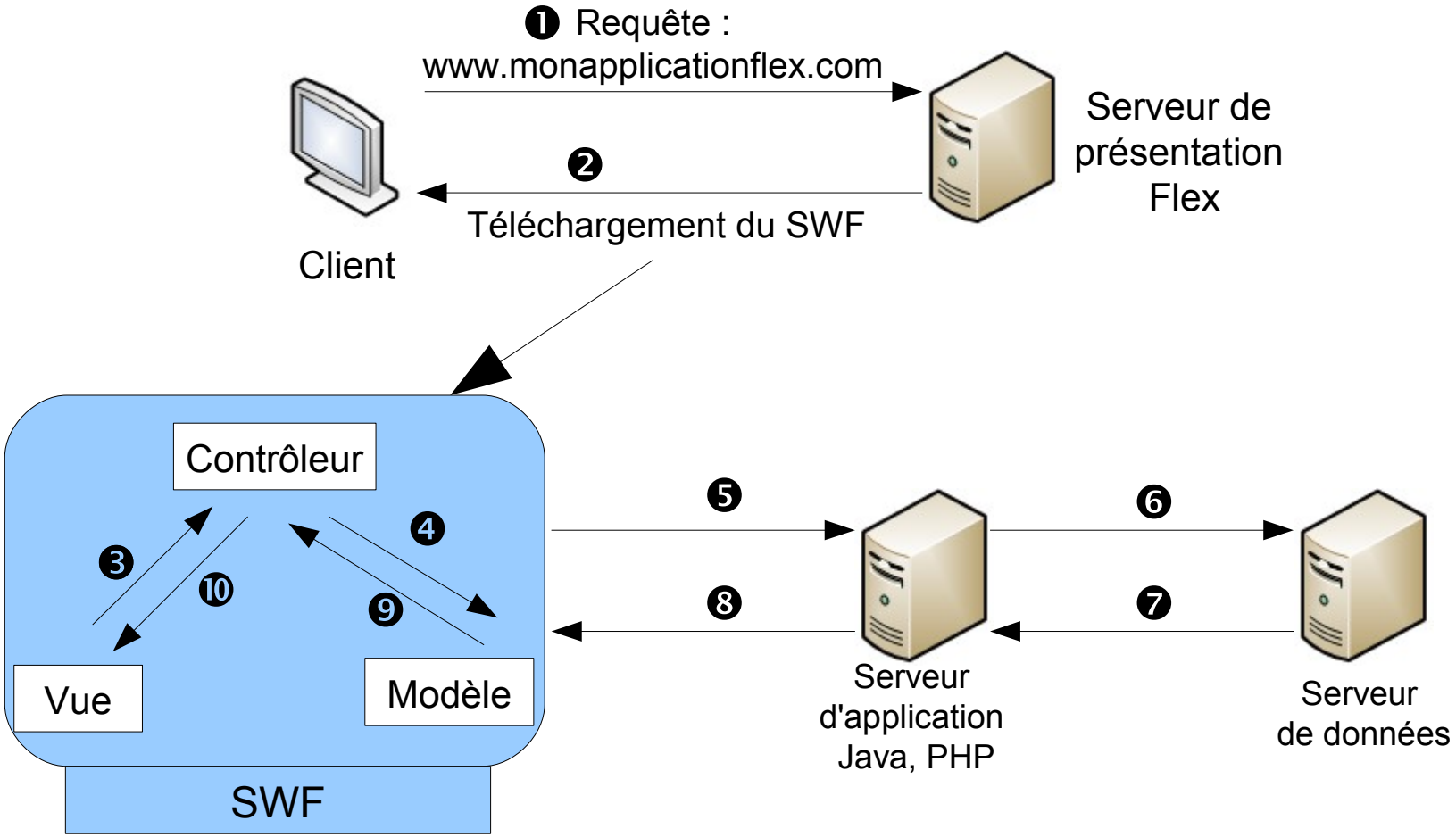
- Exemple
  - Un utilisateur souhaite utiliser l'application Flex située à l'adresse http://www.monapplicationflex.com, pointant sur le serveur de présentation qui sera le point d'entrée du processus
  - Les mouvements réalisés sont les suivants :
    - L'application est téléchargée et exécutée sur le poste client, ce qui nécessite par conséquent que Flash Player soit installé sur celui-ci

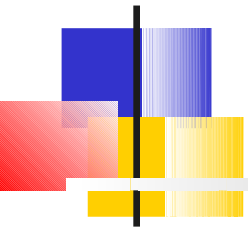


- L'application présente sur le serveur, permettant de lister les produits d'un magasin, la deuxième étape va consister à faire appel à une suite de services permettant d'obtenir la liste des produits
- Lorsque l'utilisateur cliquera sur le bouton Mise à jour situé sur l'interface graphique, cela déclenchera un événement qui sera transmis au contrôleur (3), lequel ira chercher le modèle concerné par cet événement (4)



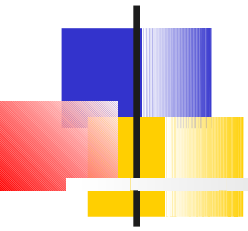
- Le modèle va ensuite faire appel au service approprié situé sur le serveur d'application (5). Ce service va alors exécuter une requête sur le serveur de base de données (6), s'en suivra une succession de réponses de la part des différents serveurs concernés (7 et 8)
- La dernière étape consistera ensuite à mettre à jour la couche de présentation via l'interaction du modèle avec le contrôleur (9 et 10)





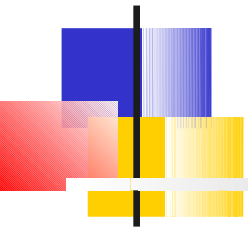
MVC en Flex

- Flex utilise deux langages pour faire son IHM
  - MXML
    - Le langage MXML qui permet de réaliser la vue (à l'aide d'une série de composants)
  - ActionScript
    - Permet le développement de la partie dynamique de l'application par gestion d'événements, c.a.d. Le contrôle



MVC en Flex

- Implémentation des composants
  - Le modèle
    - à l'aide d'ActionScript dont l'orientation objet
      - permet de créer facilement des classes et des méthodes associées
  - Le contrôleur
    - lui aussi développé à l'aide d'AS
      - met en application le modèle grâce à la notion de DataBinding
  - La vue
    - implémentée à l'aide du langage MXML
      - permet de décrire l'interface graphique avec ses notions de conteneurs



MVC en Flex

- Le mécanisme client/serveur
  - Flex est composée d'un fichier MXML principal faisant appel à des fichiers ActionScript permettant d'intégrer la notion d'architecture MVC
  - L'ensemble de ces fichiers est ensuite compilé, donnant naissance à un fichier SWF pouvant être intégré dans une page HTML

Socle d'exécution Client