

FW4SPL

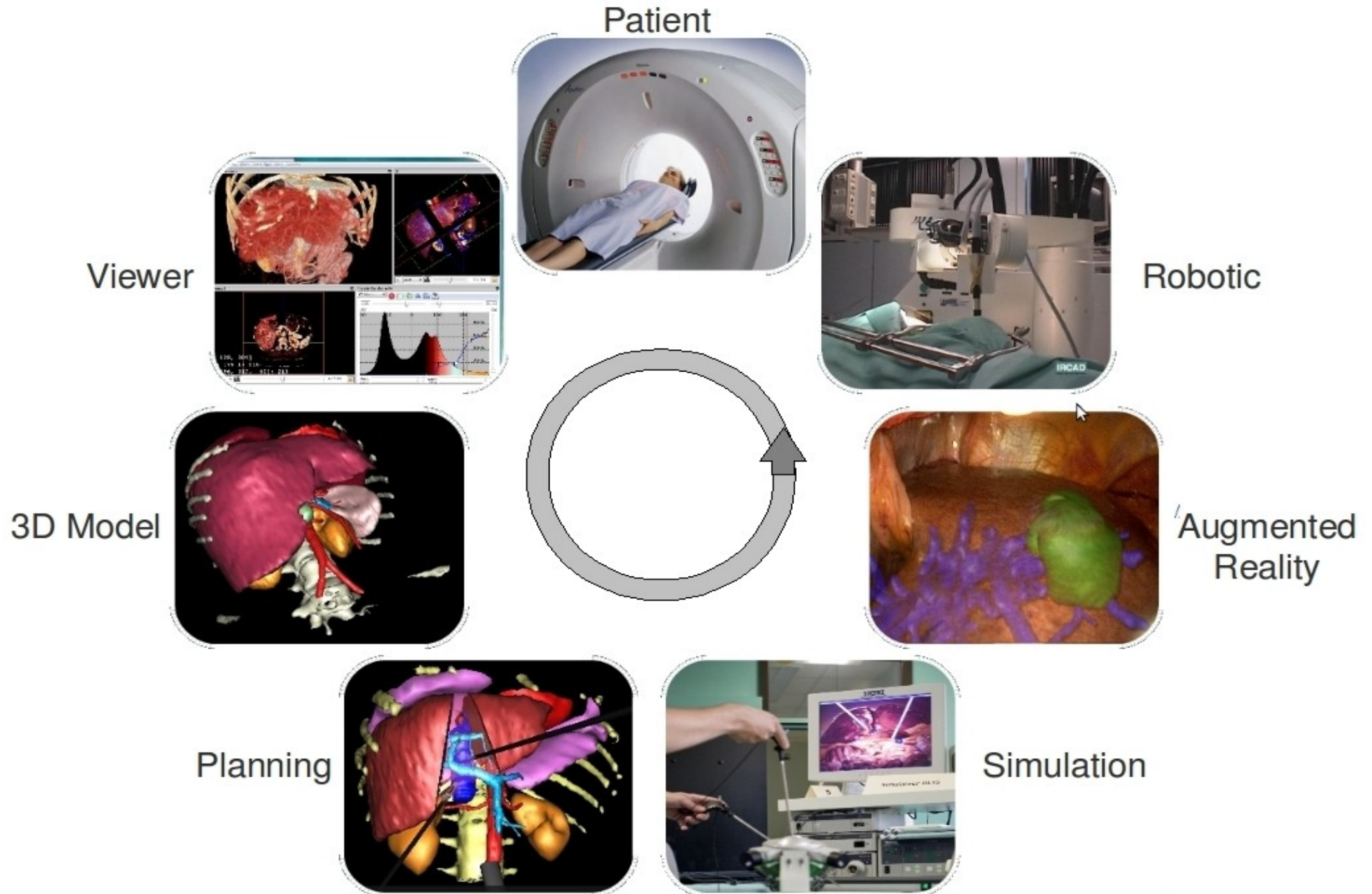
FrameWork for Software Production Line

Arnaud Charnoz
Ircad R&D

arnaud.charnoz@ircad.fr

European Developer Forum 5, Octobre 2009

Contexte



FW4SPL : Historique

- 2004-2009 : Développement de la V1 du FW4SPL
- 2007-2009 : Développement de la V2 du FW4SPL
- 2008 : Développement de VR-Render 0.7
- 2009 : Développement d'AR-Surg
- 09/2009 : Ouverture en Open Source du noyau
- 01/2010 : *Développement de VR-Planning*
- 06/2010 : *Développement de VR-Anat*

- **Caractéristiques :**
 - Architectures à base de composants
 - Architectures à base de objets/services (fonctionnalités)
 - Développé en C++
 - Architecture Multi-OS
- **Le framework est Open Source :**
 - <http://code.google.com/p/fw4spl/>

Plan

- Architecture à base de composants
- Architecture orientée objets/services
- Création d'un service
- Création d'applications
- Conclusion

Plan

- **Architecture à base de composants**
- Architecture orientée objets/services
- Création d'un service
- Création d'applications
- Conclusion

Un composant doit être...

- Encapsulé par une interface connue
- Réutilisable et recombinaison
- Déployable atomiquement (attention aux dépendances)

Les composants favorisent

- La maintenance des logiciels
- La réutilisation du code
- La collaboration entre développeurs

Exemples de systèmes de composants

- Tubes & filtres (unix)
- CORBA
- DCOM
- D-BUS
- Java Beans
- Eclipse
- Mozilla
- .Net/Mono
- Web Services
- XML-RPC
- Apache

Prérequis techniques

- OOP (interfaces, héritage, surcharge)
- C++ (méthodes virtuelles)
- STL (conteneurs, itérateurs)
- libboost (pointeurs malins)
- XML (notions de base)
- Notions sur les bibliothèques dynamiques (dlopen, ...)

Les composants en action

Composants disponibles

Specific Components



Base Components

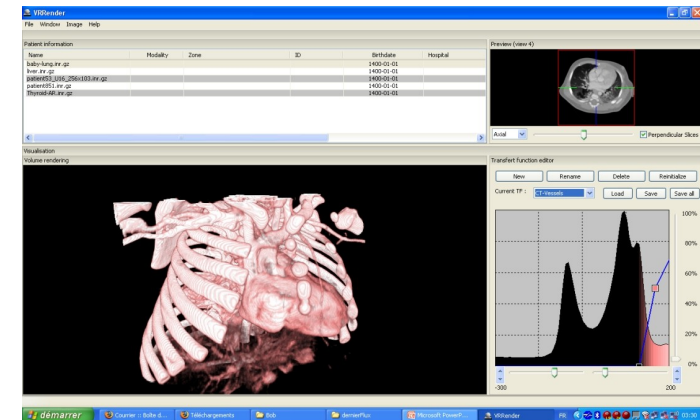
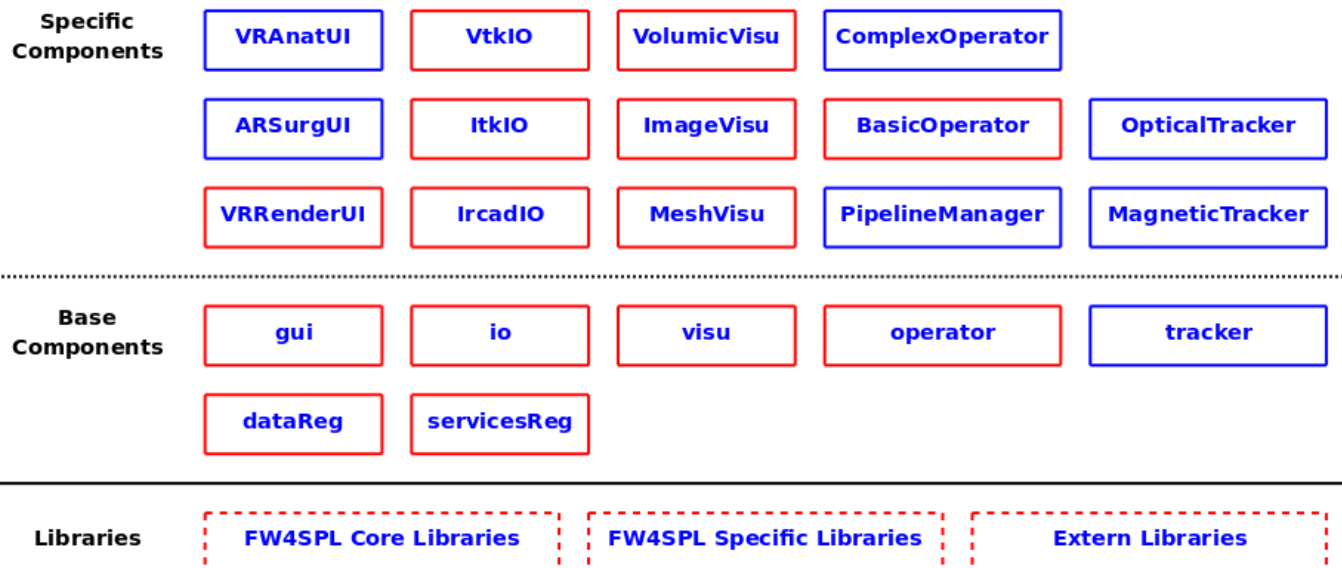


Libraries



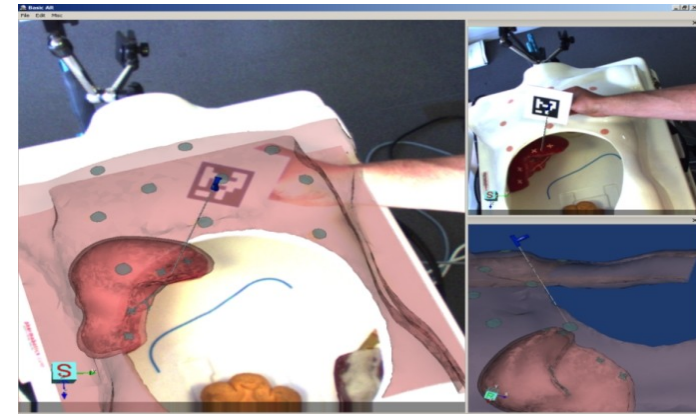
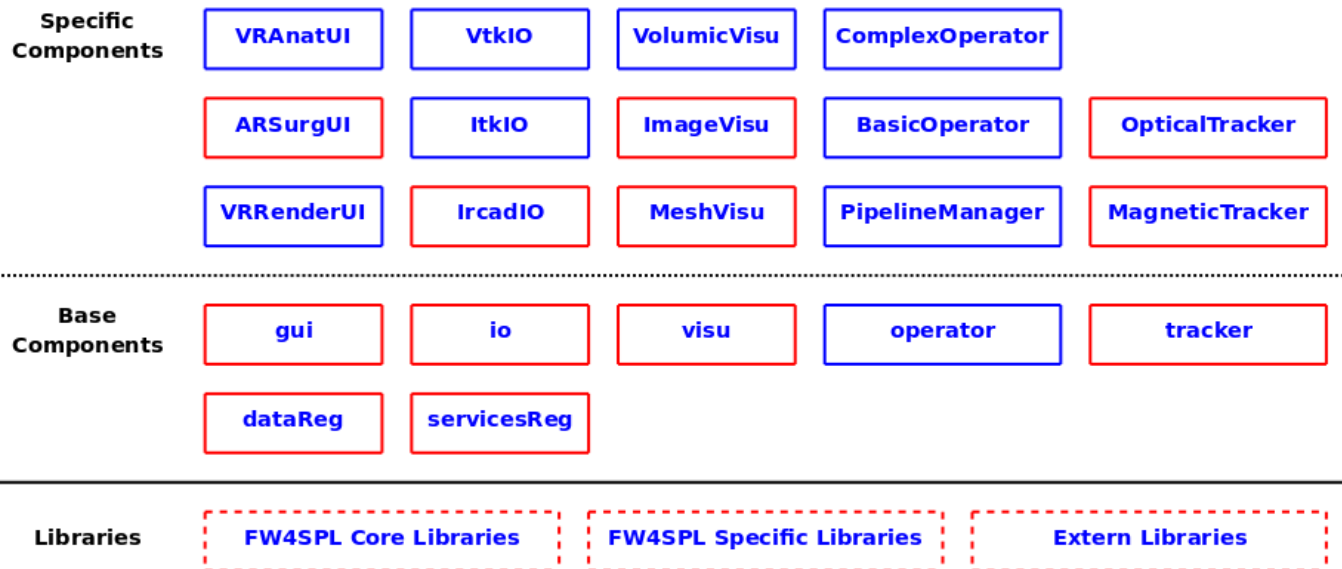
Les composants en action

Selection des composants pour VR-Render



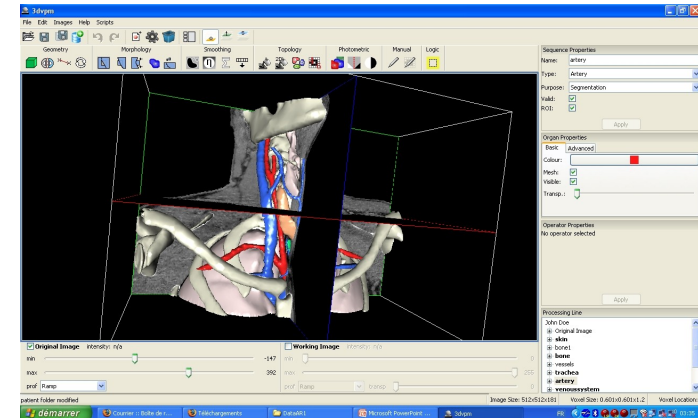
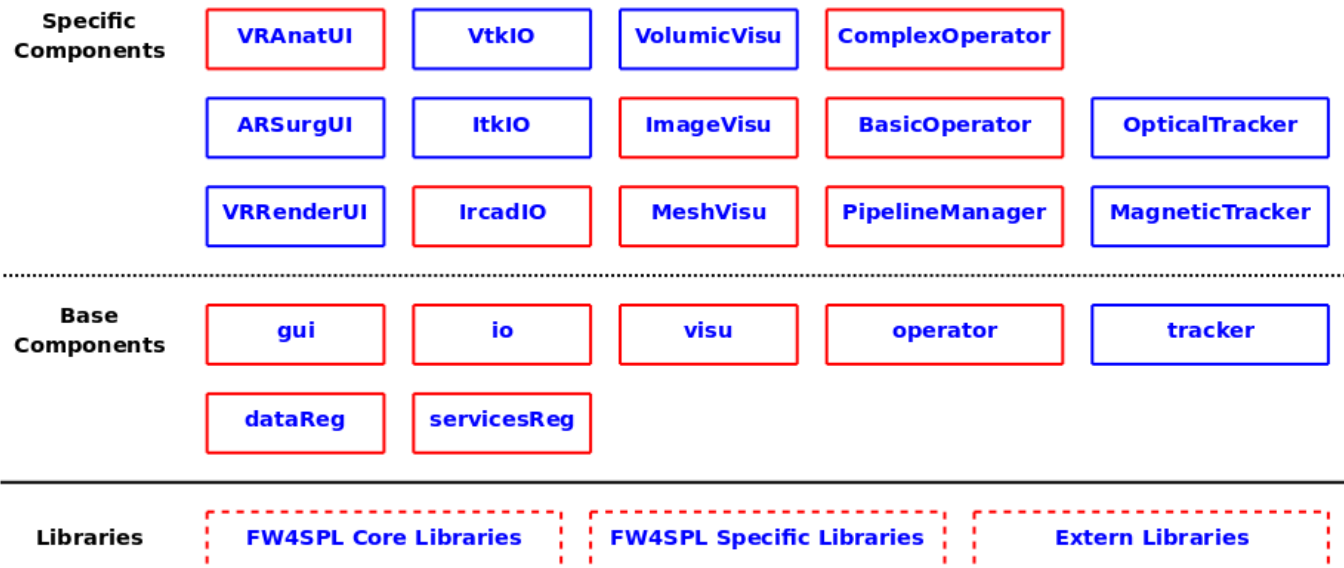
Les composants en action

Selection des composants pour AR-Surg



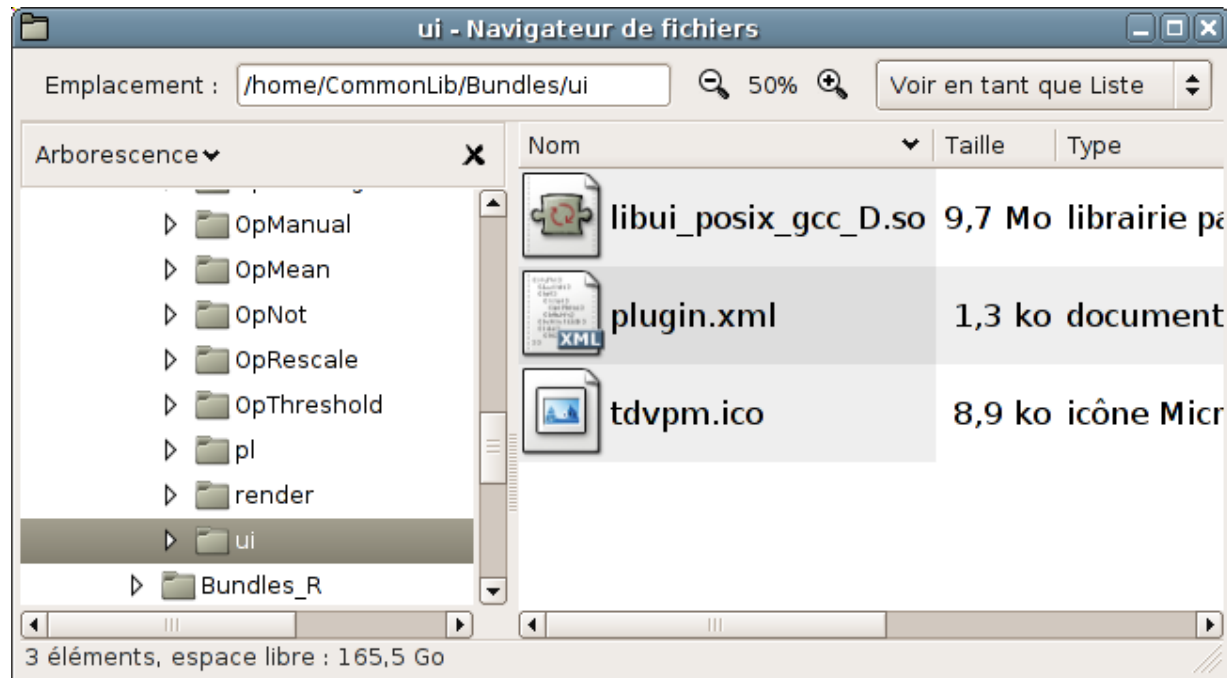
Les composants en action

Selection des composants pour VR-Anat



Structure générale d'un composant

- Un fichier de description (XML)
- Des bibliothèques dynamiques optionnelles (.so, .dll, ...)
- Des ressources optionnelles (fichiers quelconques)
- Empaqueté dans un répertoire



Description d'un composant

```
<plugin id="ui" class="ui::Plugin">
```

```
  <library name="ui" partial-name="yes"/>
```

```
  <extension-point id="ui::views"/>
```

```
  <extension-point id="ui::actions"/>
```

```
  <extension-point id="ui::aspects"/>
```

```
  <extension-point id="ui::closeVetos"/>
```

```
  <extension implements="ui::actions">
```

```
    <menu identifier="file" text="File">
```

```
      <group identifier="begin"/>
```

```
      <separator/>
```

```
      <action identifier = "quit"
```

```
        text = "Quit"
```

```
        menuBarPath = "/File/End"
```

```
        class = "ui::QuitAction"/>
```

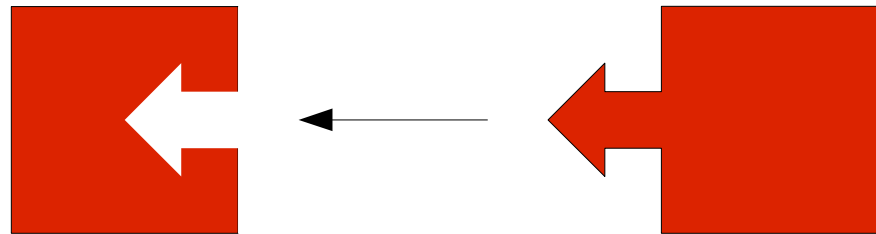
```
    </menu>
```

```
  </extension>
```

Contributions entre composants

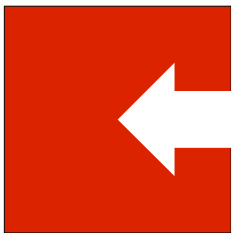
Points d'extension

Extensions



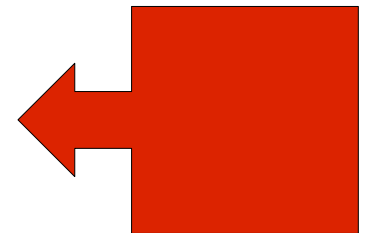
Buts d'un point d'extension

- Création d'un rendez-vous pour des contributions
- Définition d'une interface
- Découverte des contributions disponibles
- Utilisation des contributions :
 - Utilisation des données associées en paramètres
 - Création dynamique d'instances de classes associées

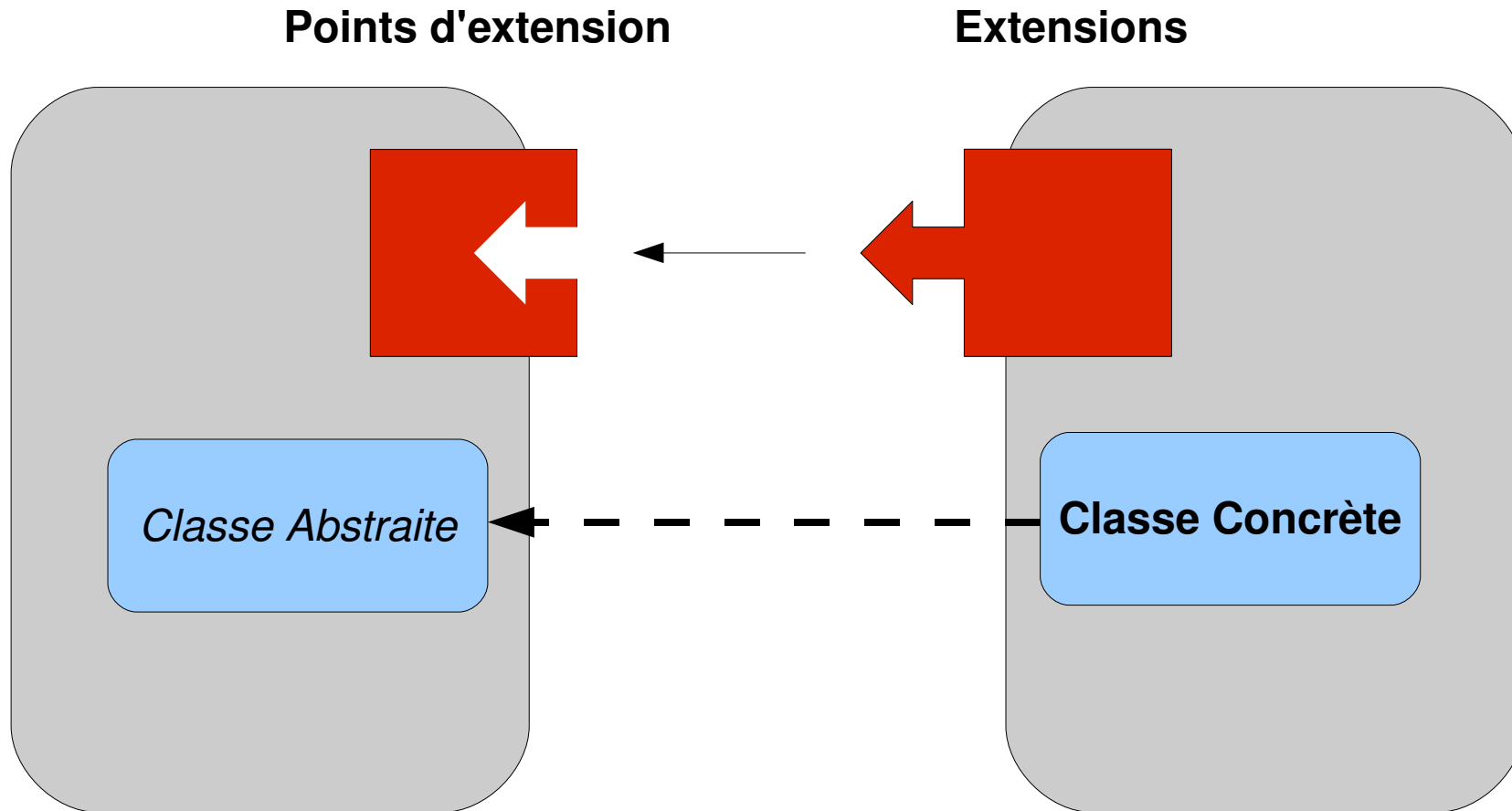


Buts d'une extension

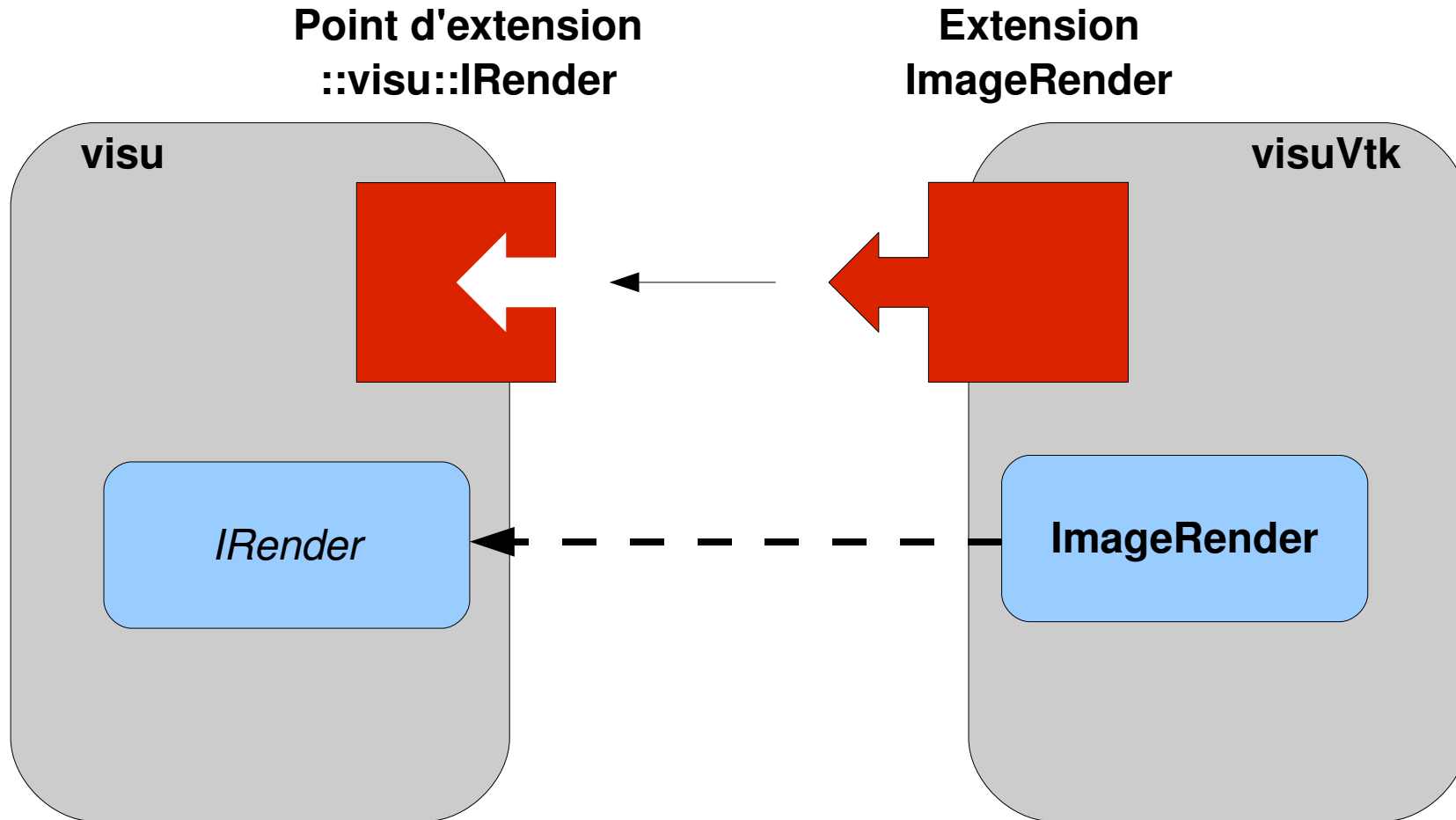
- Connexion à un point d'extension
- Implantation de l'interface associée
- Fourniture de données en paramètres



Contributions entre composants



Contributions entre composants



La bibliothèque fwRuntime gère...

- Les composants
- Les points d'extensions et des extensions
- Le chargement des bibliothèques dynamiques
- La création dynamique d'instances des classes associées aux extensions
- L'accès aux données en paramètres
- L'accès aux ressources

Plan

- Architecture à base de composants
- **Architecture orientée objets/services**
- Création d'un service
- Création d'applications
- Conclusion

Un service doit être...

- Manipulé par une interface connue
- Utilisable dans une architecture à base de composants
- Capable de communiquer avec les autres services
- Capable de démarrer/s'arrêter à chaud dans le programme

L'approche objet/service favorise

- La séparation données / fonctionnalités
- La configuration dynamique d'application
- La maintenance des logiciels

Architecture orientée objets/services



My Software



API d'un service : IService

- **SetConfiguration(cfg)** - initialise les paramètres du service
- **Configure()** - analyse les paramètres du service
- **Start()** - démarre le service
- **Update()** - mets à jour le service
- **Update(msg)** - notification d'un message
- **Stop()** - arrête le service

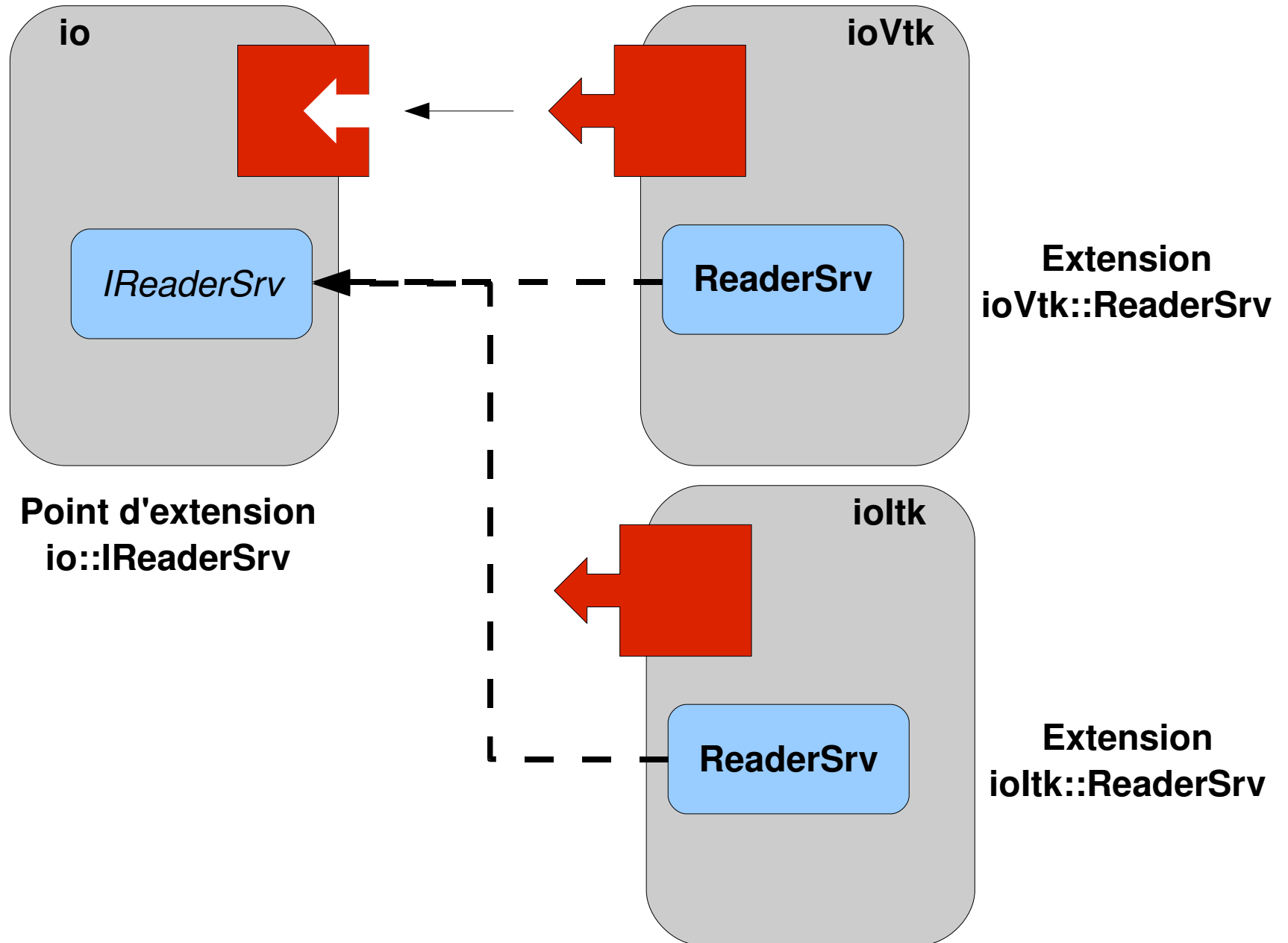
Ex : service de lecture

- **SetConfiguration(cfg)** - initialise les paramètres du service
- **Configure()** - récupère et valide le chemin du fichier
- **Start()** - ne fait rien
- **Update()** - effectue la lecture du fichier et notifie les autres services
- **Update(msg)** - ne fait rien
- **Stop()** - ne fait rien

Ex : service de visu

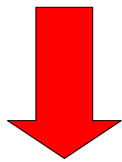
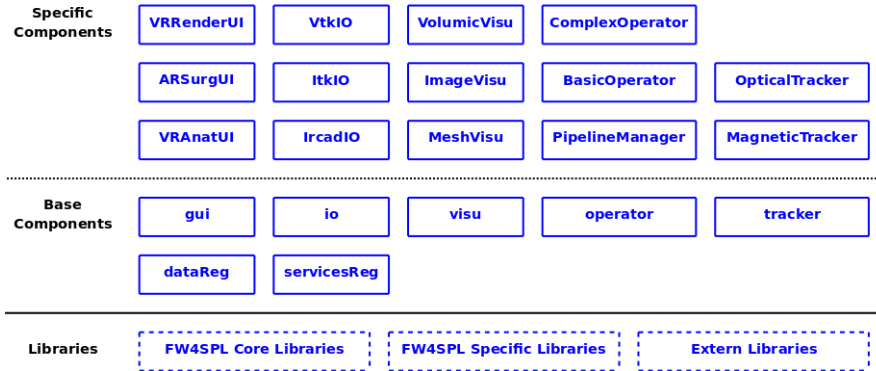
- **SetConfiguration(cfg)** - initialise les paramètres du service
- **Configure()** - récupère le container IHM
- **Start()** - créer une fenêtre de rendu dans le container IHM
- **Update()** - force le refresh de la scène
- **Update(msg)** - si l'objet a été modifié, force le refresh de la scène
- **Stop()** - détruit la fenêtre de rendu

Services & composants



Services, composants & application

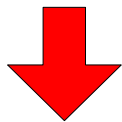
Mes Composants



profile.xml

Factory services/objets

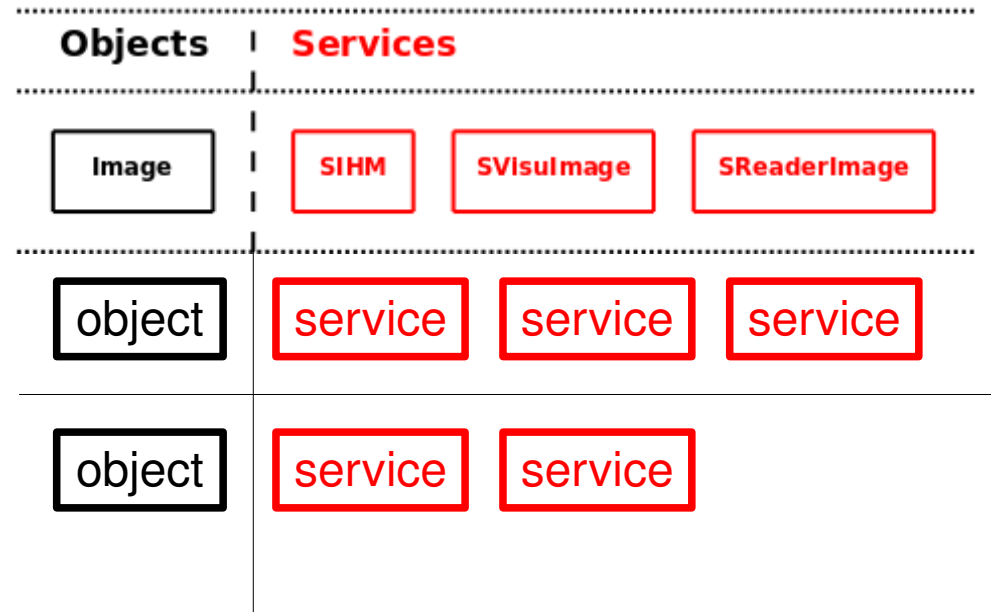
Srv class	Srv type	Object type
ioVtk::ImgReader	io::Reader	data::Image
ioItk::ImgReader	io::Reader	data::Image
ioVtk::MeshReader	io::Reader	data::Mesh



```
::layer:add(myImage, "::io::Reader", "::ioVTK::ImageReaderService", "reader")
```

OSR : Object Service Registry

My Software



Config.xml
ou
Modification dynamique

Plan

- Architecture à base de composants
- Architecture orientée objets/services
- **Création d'un service**
- Création d'applications
- Conclusion

Création d'un service

Service de visualisation d'image

Srv class	Srv type	Object type
edf::ImageViewerService	visu::IRender	data::Image

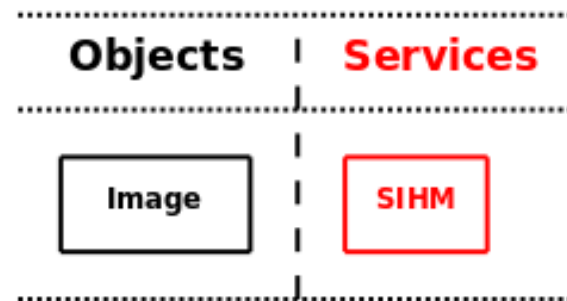
Plan

- Architecture à base de composants
- Architecture orientée objets/services
- Création d'un service
- **Création d'applications**
- Conclusion

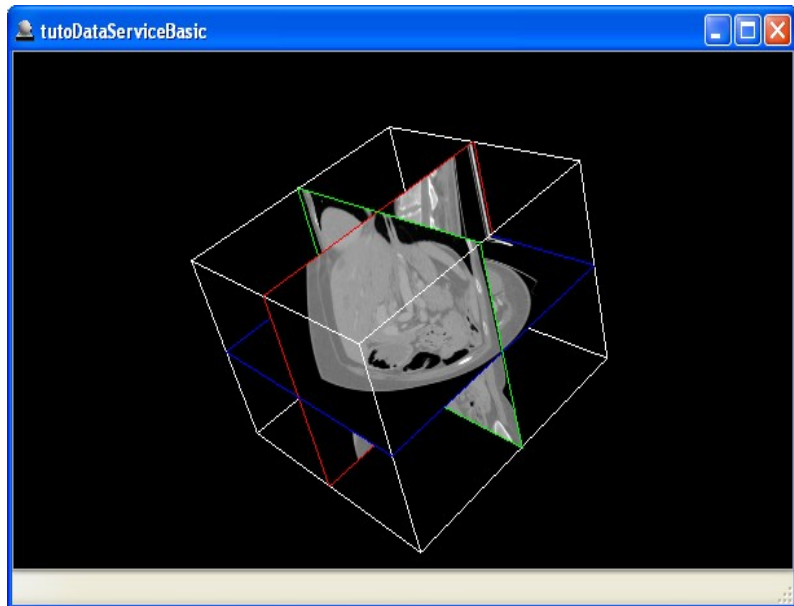
Tutorial 1



My Software



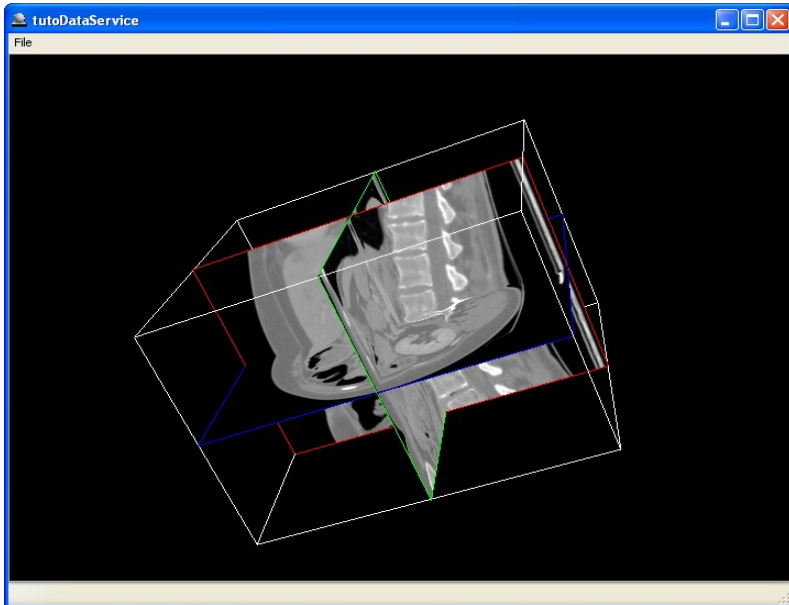
Tutorial 2



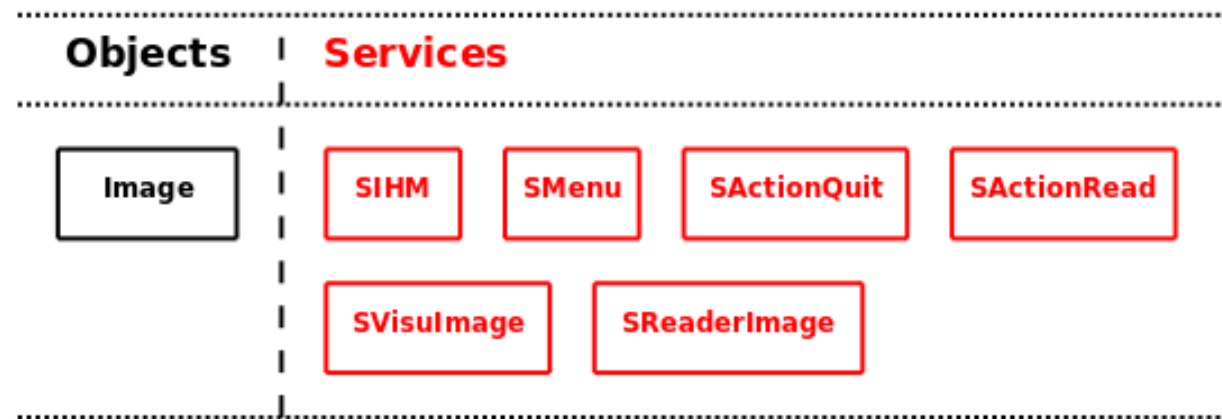
My Software



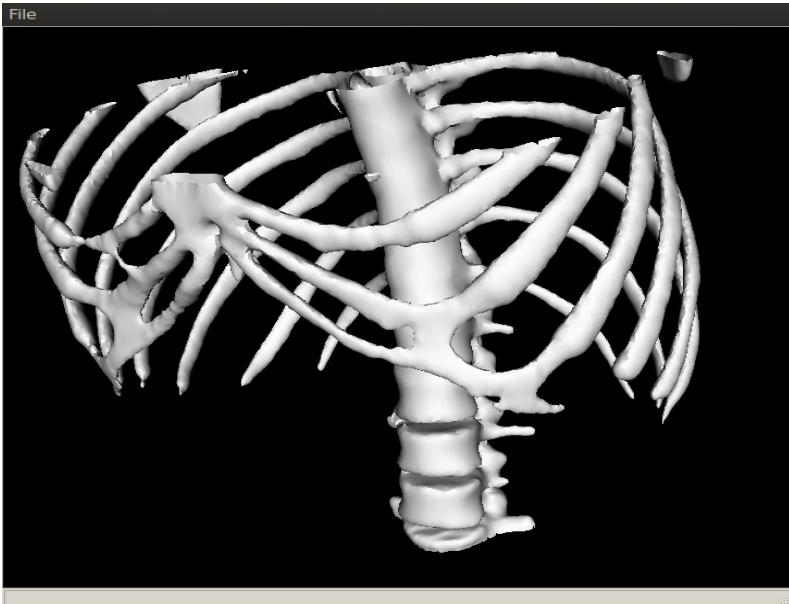
Tutorial 3



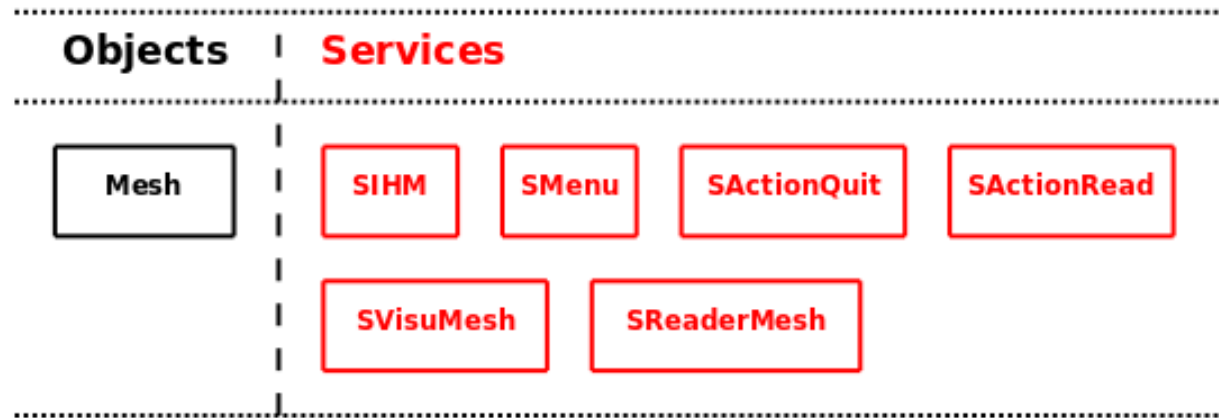
My Software



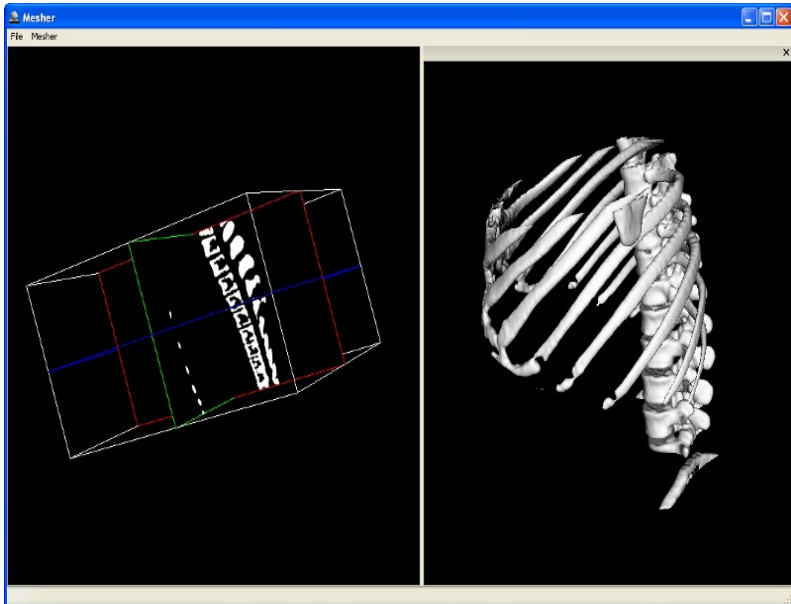
Tutorial 4



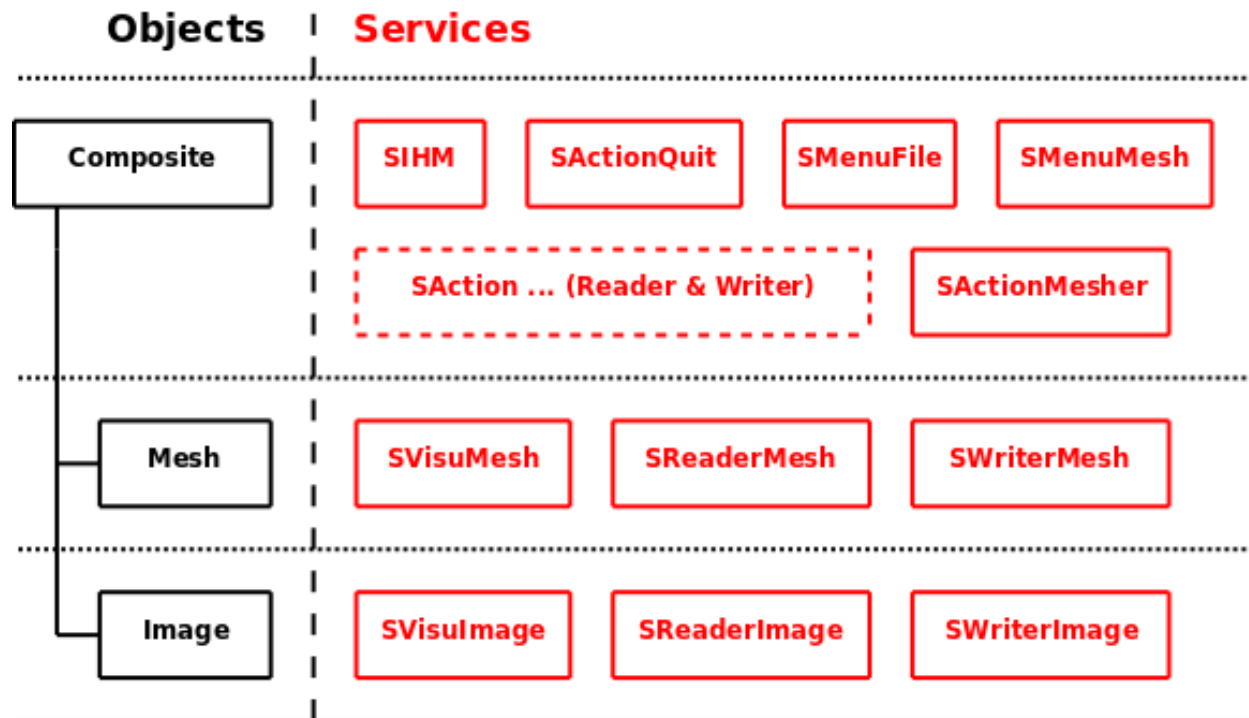
My Software



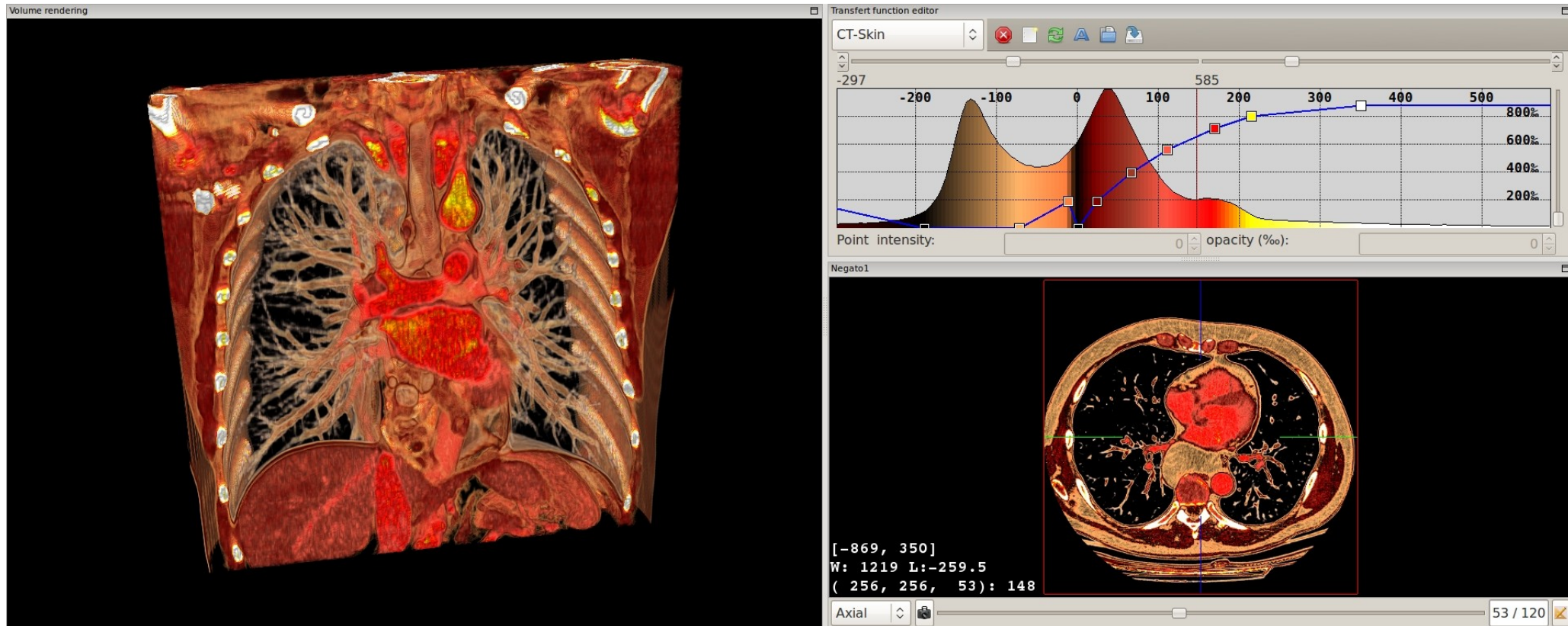
Tutorial 5



My Software



VR-Render 0.8



VR-Render 0.8 : <http://www.ircad.fr/software/vr-render>

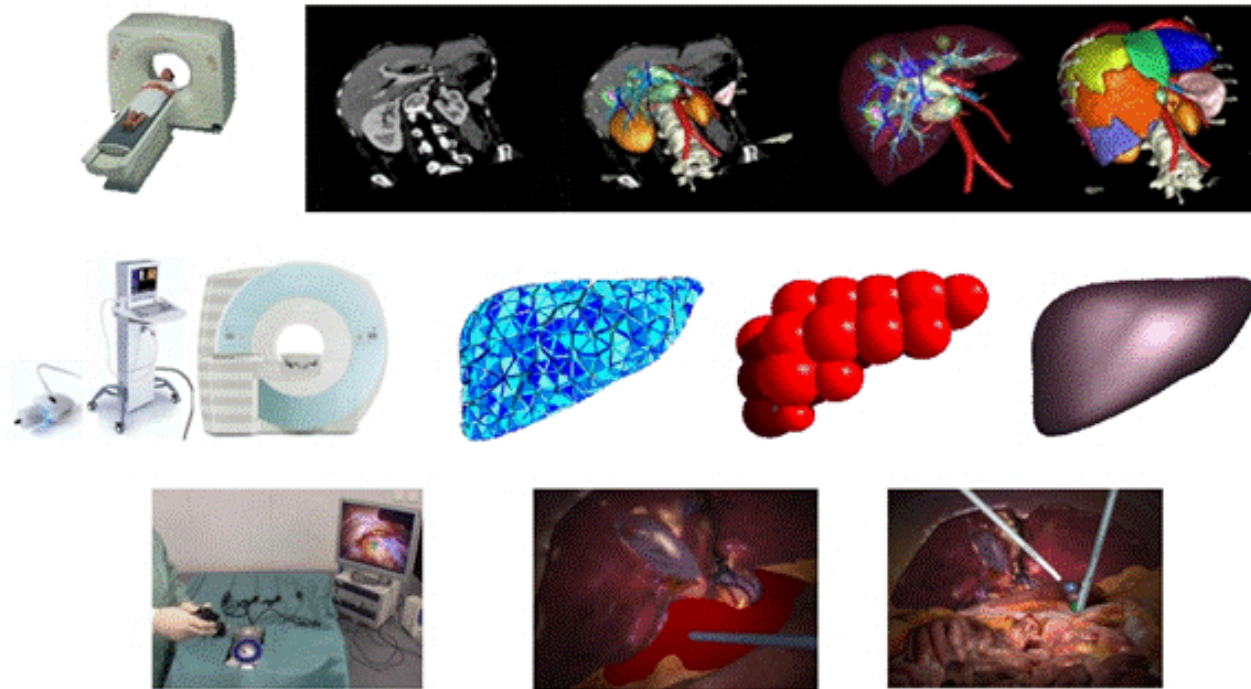
Plan

- Architecture à base de composants
- Architecture orientée objets/services
- Création d'un service
- Création d'applications
- **Conclusion**

Conclusion

- **Fw4spl :**
 - Architecture modulaire & évolutive
 - Multi-OS : Windows, macOS, Linux
 - Open source
 - Concept validé et utilisé dans plusieurs applications
- **Sconspiracy :** Outils de compilation open source
- **Améliorations futures :**
 - Améliorer le système de communication
 - XML – Validation plus complète de l'ensemble des fichiers
 - Améliorer les performances des librairies centrales (applications temps réels).
 - Meilleur support du multi-threading

Patient-Specific Simulation for Pre-Operative Realistic Training of Liver Surgery



FP7, ICT-2007.5.3

Stages 2009-2010

<http://www.ircad.fr/applied/stages>

- Maquette à base de technologies VoIP, VideoOnLan, visioconférence IP
- Intégration de fonctions de fusion d'images dans des logiciels médicaux
- Evaluation des frameworks C++ : Mozilla XPFE/XPCOM, Qt et FW4SPL
- Evolution du stockage d'images médicales
- Evaluation des solutions de virtualisation du marché
- Rendu Volumique : manipulation des fonctions de transfert couleur

Ouf !

Merci de votre attention !

- FW4SPL Open Source
 - <http://code.google.com/p/fw4spl>
- SConspiracy Open Source
 - <http://code.google.com/p/sconspiracy>
- VR-Render 0.8
 - <http://www.ircad.fr/software/vr-render>
- Stage 2009-2010
 - <http://www.ircad.fr/applied/stages>