

# IMGT/Choreography



# Sommaire

- L'architecture Orientée Services du W3C :
  - Les protocoles fondateurs (SOAP, WSDL, UDDI).
- Approches pour la chorégraphie de services :
  - OASIS vs W3c vs DARPA.
- Approches dans le domaine de la bioinformatique :
  - MyGrid et BioMoby.
- IMGT :

Feuille de route pour l'implémentation d'IMGT-Choreography

# SOA

- L'architecture orientée services

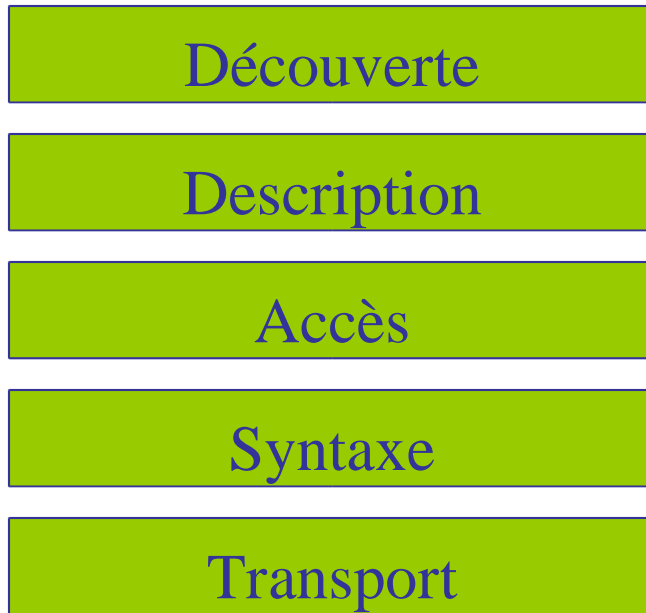
# Services Web : Concepts

- Définition :

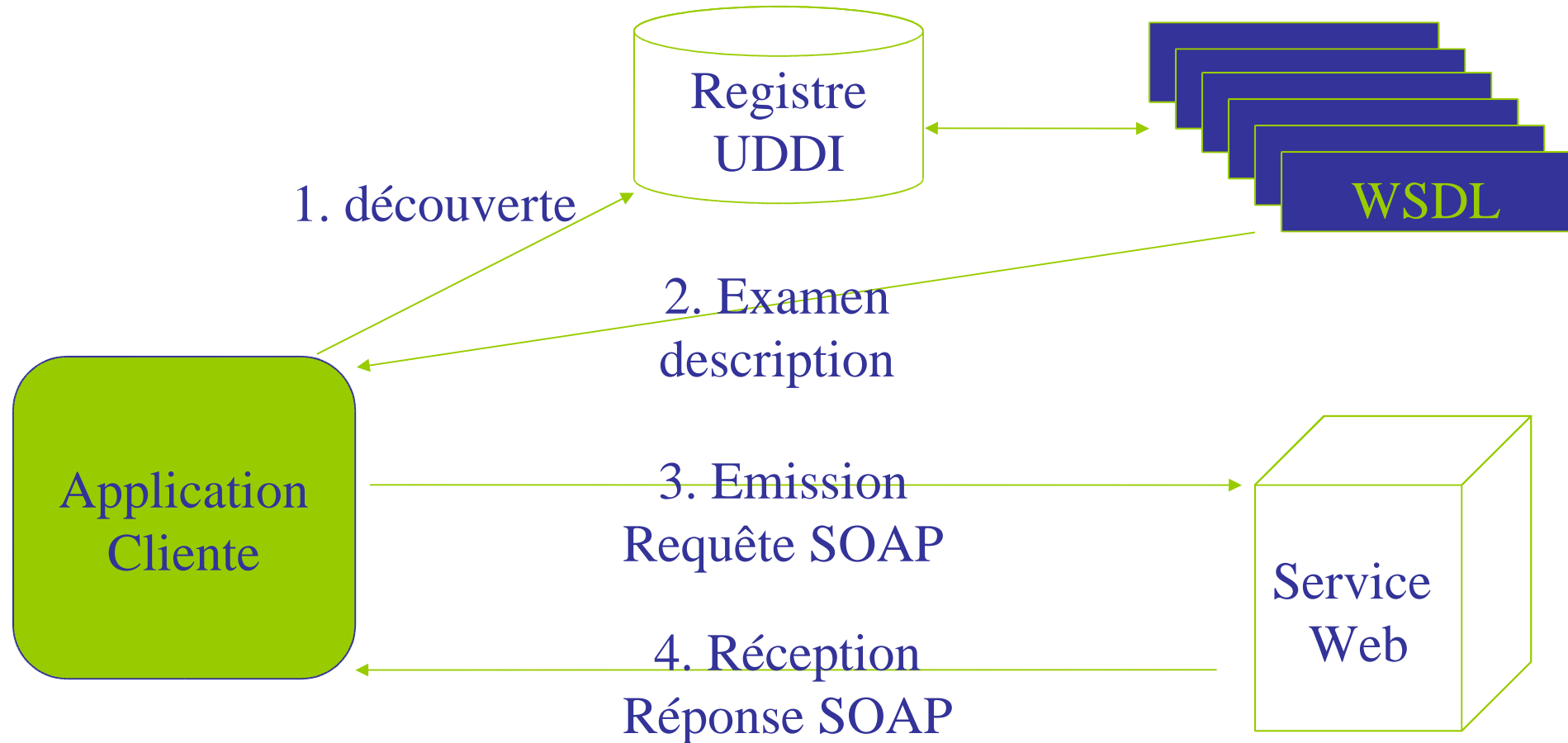
"Applications logicielles qui peuvent être découvertes, décrites et auxquelles on peut accéder sur le Web."

- Intégrer des applications existantes .
- S'absoudre de mises en oeuvre hétérogènes.
- Exposer (publier) ces applications sur le WWW.
- Composer des services pour créer des activités.
- Orchestrer, chorégraphier ces activités.

# Service Oriented Architecture : Concepts



# Service Oriented Architecture : Concepts



# SOAP : Médiateur de l'interopérabilité

```
<soap:Envelope
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:imgt="http://imgt.org/ontology/"
  xmlns:imgtWS="http://imgt.org/ontology/WebServices">
  <soap:Header>
    <imgtWS:parameterList SOAP:mustUnderstand="true">
      <param name="stringency" value="3.0" type="xsd:double"/>
    </imgtWS:parameterList>
  </soap:Header>
  <soap:Body>
    <imgt:querySeqData>
      <seqData>
        <simpleCatalogEntry numacc="M26678" />
      </seqData>
    </imgt:querySeqData>
  </soap:Body>
</soap:Envelope>
```

Le tout est ensuite véhiculé en s'appuyant sur un protocole réseau:  
Le plus fréquemment HTTP (SMTP, JMS également supportés)

# Description des services : WSDL

```
<description>
```

```
  <types>
    "Définition des types de données (XSD)"
  </types>

  <message>
    <part>
      "Définition des messages (selon types)"
    </part>
    <part>...</part>
  </message>

  <portType>
    "un portType est comparable à une Interface Java"
    <operation>
      "une opération est composée de messages"
    </operation>
    <operation>...</operation>
  </portType>
```

Description abstraite

```
  <binding>
    "liaison des portType à un protocole, p.e. SOAP"
  </binding>

  <service>
    Description concrète
    <port>
      "localisation du service"
    </port>
  </service>
```

Description concrète

```
</description>
```



# Découverte des services : UDDI

## Pages blanches

- Nom business
- Contact information
- Description

## Pages jaunes

- Services et produits
- Codes industriels
- Index géographique

## Pages vertes

- Règles métier
- Description du service
- WSDL

- Puis-je avoir un logiciel d'alignement pour mes séquences d'immunoglobulines ?
- La sortie de tel logiciel est elle compatible avec l'entrée d'un autre ?

# Ce qui nous manque

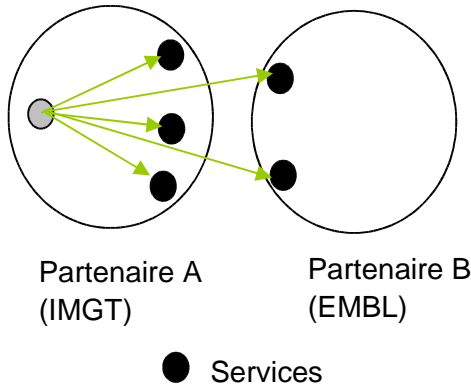
- Définition des données :
  - XML Schema est un langage de définition pour les messages entrées/sorties
  - Pas de définition explicite des données spécifiques à un domaine
- Comportement des invocations :
  - Absence de définition des séquences opérationnelles
- Prise en charge de la médiation des données ou des opérations
- Qualité du service (QoS)
- Sécurité

# Chorégraphie

- Approches pour la chorégraphie de services :

# La Composition de services : topologie

- Orchestration :



- Processus métier *exécutable*.

- Interactions "intra" et "extra" organisationnelles.

- L'interaction est décrite au niveau des *messages*.

- Prise en compte de la *séquentialité*.

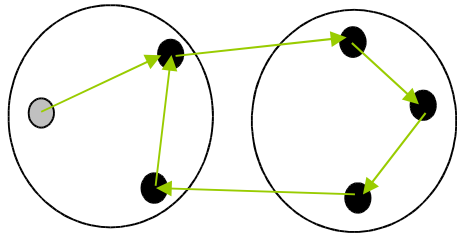
- Processus *transactionnels* de *longue durée*.

- *Transaction compensatoires* (<>ACID)

- ~BPEL4WS (OASIS)

# La Composition de services : topologie

- Chorégraphie :



Partenaire A  
(IMGT)

Partenaire B  
(EMBL)

● Services

- Les mêmes concepts restent vrais.

- Modèle collaboratif.

- Chaque participant décrit son rôle dans le flux.

- Décrit les messages publiques opérant entre les différents participants.

- WSCI, WSCL (W3C)

Convergence des concepts mais pas des standards

# La Chorégraphie de SW :

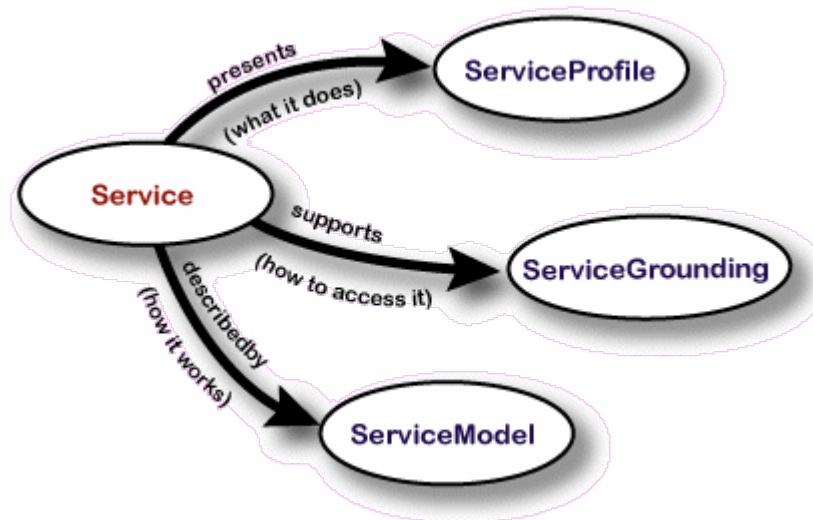
- En Bref : 2 approches (complémentaires)
  - Une approche syntaxique focalisant sur la structure des messages échangées.
    - Les "solutions" actuelles du W3C et d'OASIS
  - Description sémantique et syntaxique des SW
    - OWL-S

# OWL-S : Concepts

- Améliorer la découverte et l'intégration de services nécessite d'accroître le degré de description de leurs interfaces.
- Associer l'interface à une ontologie.
- Passer d'une approche "*machine-readable*" à une approche "*machine-understandable*".

# OWL-S

- Définit des ontologies\* pour la construction de modèles de services.
  - Ce que fait le service : *Profile*
  - Comment il le fait : *Model*
  - Comment accéder concrètement au service : *Grounding*



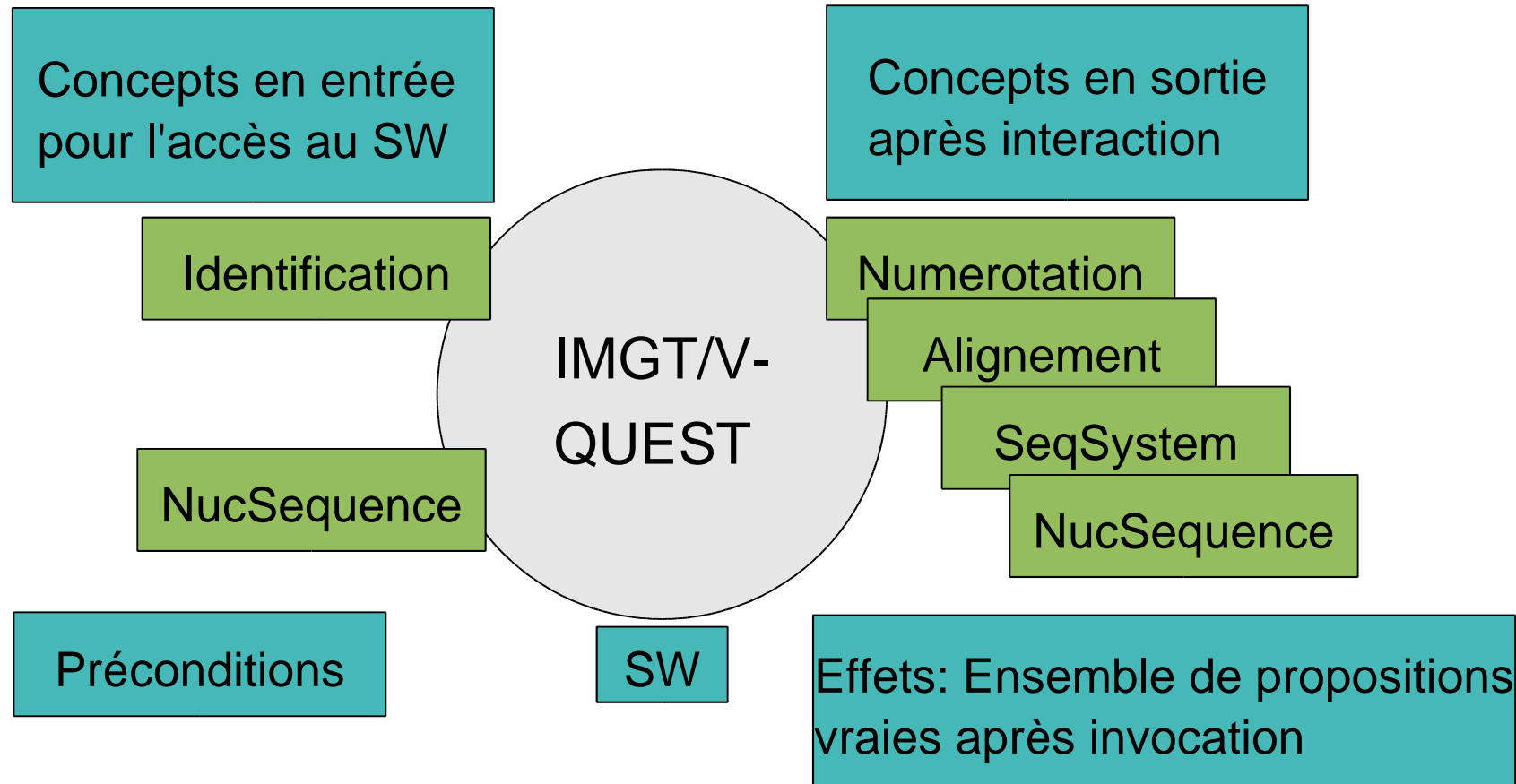
\*Modèle spécifié formellement

[The OWL services coalition]



# OWL-S : Service Profile

- Détails sur le service :



[The OWL services coalition]

# Bioinformatique

- Approches dans le domaine de la bioinformatique :

# myGrid

- "Projet visant à fournir un middleware Grid, orienté services, open-source, permettant des analyses bio-informatiques basées sur l'échange de données."

Les principaux acteurs de myGrid sont les auteurs de TAMBIS

"Reconduire *in-silico* les bonnes pratiques utilisées à la paille"

- Workflow et services : repositaires
  - Traçabilité des résultats intermédiaires
  - Provenance des résultats
  - Notification des mises à jour
- ~ Un cahier de laboratoire pour services web ...

Une boîte noire n'a pas de sens en recherche

Extrait d'un audit :

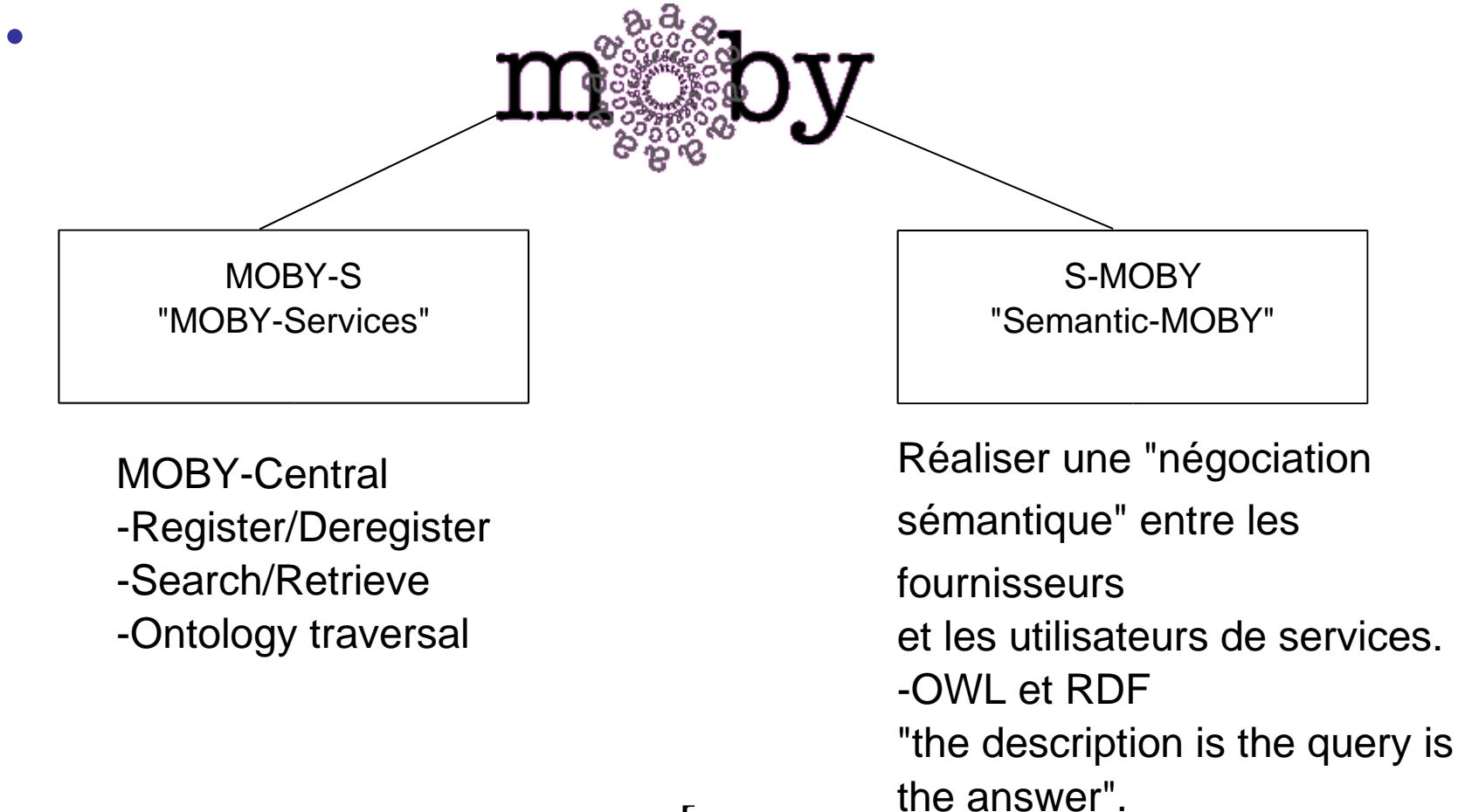
"J'ai cliqué là, puis là, puis j'ai découvert ça"

# myGrid: Orchestration de services

- FreeFluo : Un moteur de chorégraphie
  - Réalisation de workflows WSFL et Xscufl (Simple Conceptual Unified Flow Language).
  - Support de UDDI.
  - IHM Taverna sert de support graphique à la composition de services.
  - Fonctionne indépendamment de myGrid.
  - Les workflows peuvent être exposés en tant que services web (Axis).

# bioMOBY

"Projet open-source pour la découverte, l'intégration et l'interopérabilité des bases de données biologiques".



[AS Intégration des sources de données en génomique Jan 2004]

# MOBY-S

- Buts :

Définir et pouvoir manipuler (API) :

1. Système de représentation des données basé sur une "ontologie".
2. Système de description des services web basé sur une "ontologie".
3. Annuaire de services web "*ontology-aware*".

# MOBY-S

- MOBY-Central :
  - 250 service connectés
  - Connaît les types de données, de services et les relations entre eux.
- Travaux en commun avec <sup>my</sup>Grid sur les ontologies



## Conclusions à mi-parcours

- Assister plutôt qu'automatiser :

Publier un article sur la maladie de Behcet requiert bien plus de transparence que de réserver un vol et un hôtel.

- Besoin de formats flexibles :

Demander aux nombreux organismes de recherche de restructurer leurs données car le format X<sup>©</sup> "lave plus blanc" que le format Y<sup>©</sup> est dans de nombreux cas une cause perdue. Surtout que le format Z<sup>©</sup> est bien mieux.

- Permettre aux utilisateurs d'annoter selon leur vision.

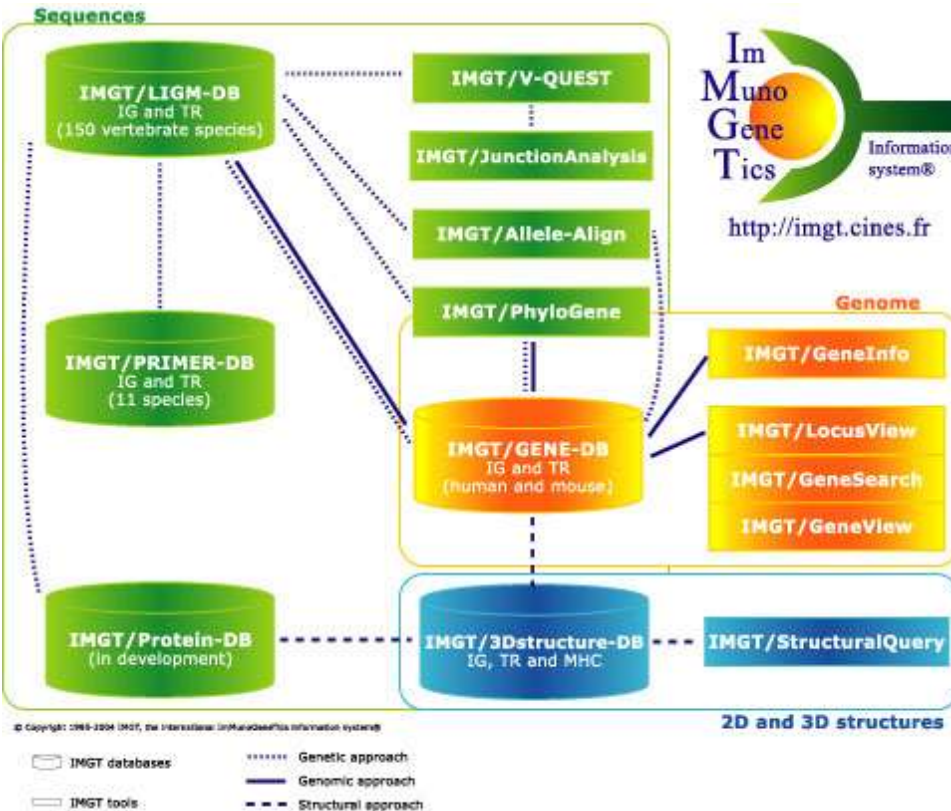
# IMGT

- Feuille de route pour l'implémentation d'IMGT-Choreography

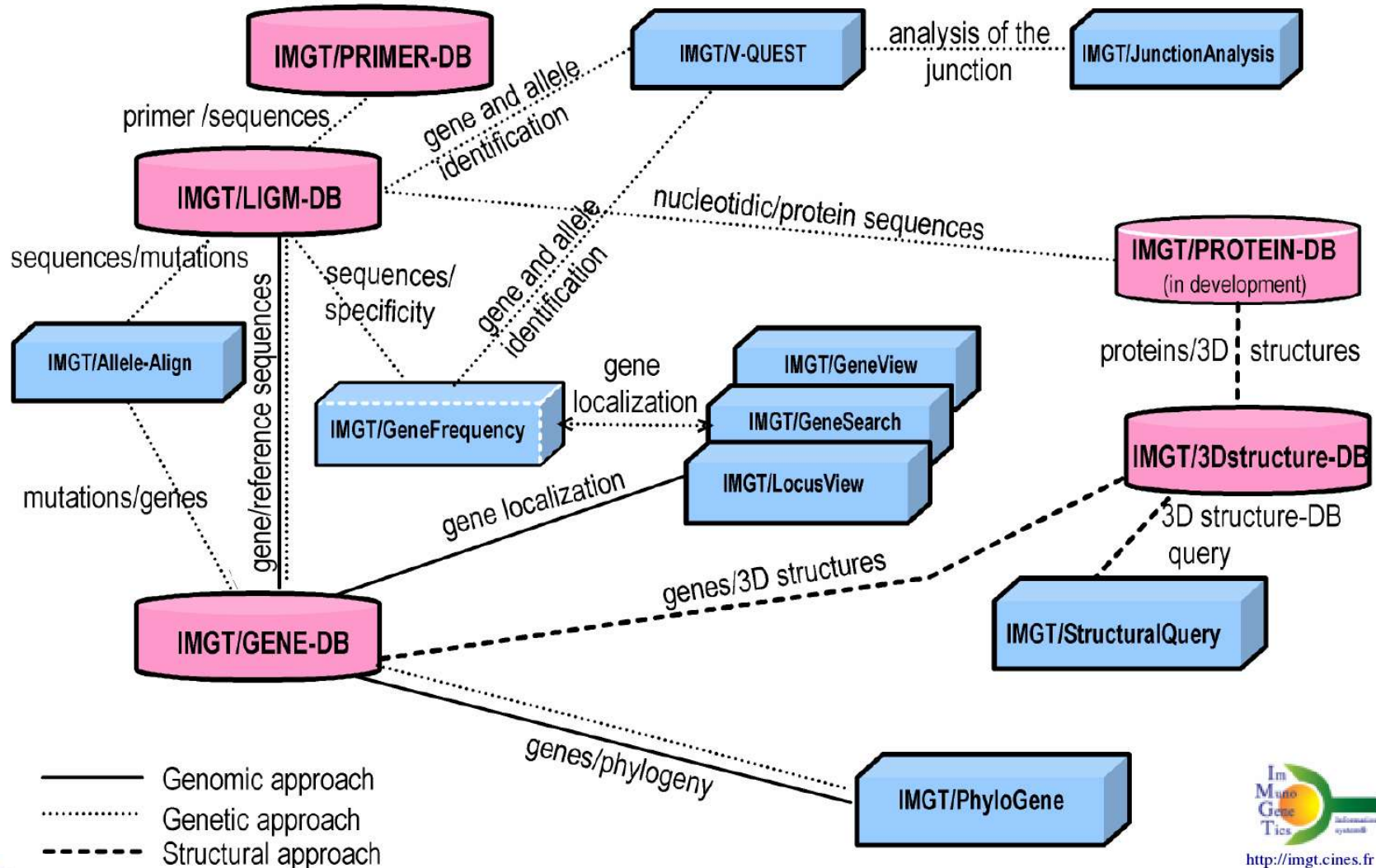
# IMGT : Un Système d'Information complexe



- 5 bases de données
  - 88 000 séquences
- 9 outils dédiés en ligne
- 1 corpus de 8000 pages HTML



# IMGT : Diagramme des composants



# IMGT : Hétérogénéité des mises en oeuvre

<b>Base de donnée</b>	<b>Langage</b>	<b>Persistence</b>
IMGT/LIGM-DB	Java	Sybase/Transact SQL
IMGT/GENE-DB	Java	Sybase/Transact SQL
IMGT/PROTEIN-DB	Java	Sybase/Transact SQL
IMGT/PRIMER-DB	Perl	MySQL/SQL
IMGT/3DStructure-DB	Perl	MySQL/SQL

# IMGT : Hétérogénéité des mises en oeuvre

Outil	Langage
IMGT/V-QUEST	Java + natif
IMGT/JCTA	Java
IMGT/Allele-align	Java
IMGT/PhyloGene	Perl + natif + MySQL
IMGT/LocusView	Perl
IMGT/GeneView	Perl
IMGT/GeneSearch	Perl
IMGT/CloneSearch	Perl
IMGT/GeneInfo	Java
IMGT/StructuralQuery	Perl
IMGT/GeneFrequency	Java

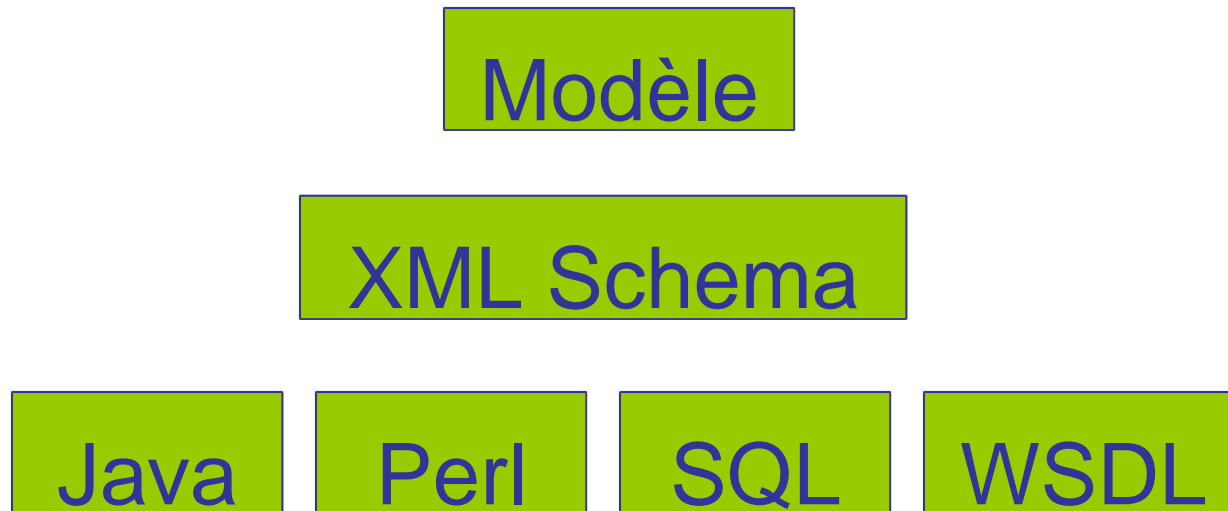
# IMGT-ONTOLOGY : Une approche cohérente

- Spécification formelle des concepts en ImmunoGénétique
- Représentation culturelle : IMGT Scientific chart
  - pages HTML
- Représentation littérale: IMGT-ML
  - schémas XML W3C
- Stratégie Bottom-Up :
  - circulation des concepts du biologiste vers la machine

Une sémantique distribuée à tous les niveaux  
du Système

# IMGT-ONTOLOGY : Une approche cohérente

- Architecture pilotée par le modèle (MDA)
  - version 4 de l'API en cours de développement





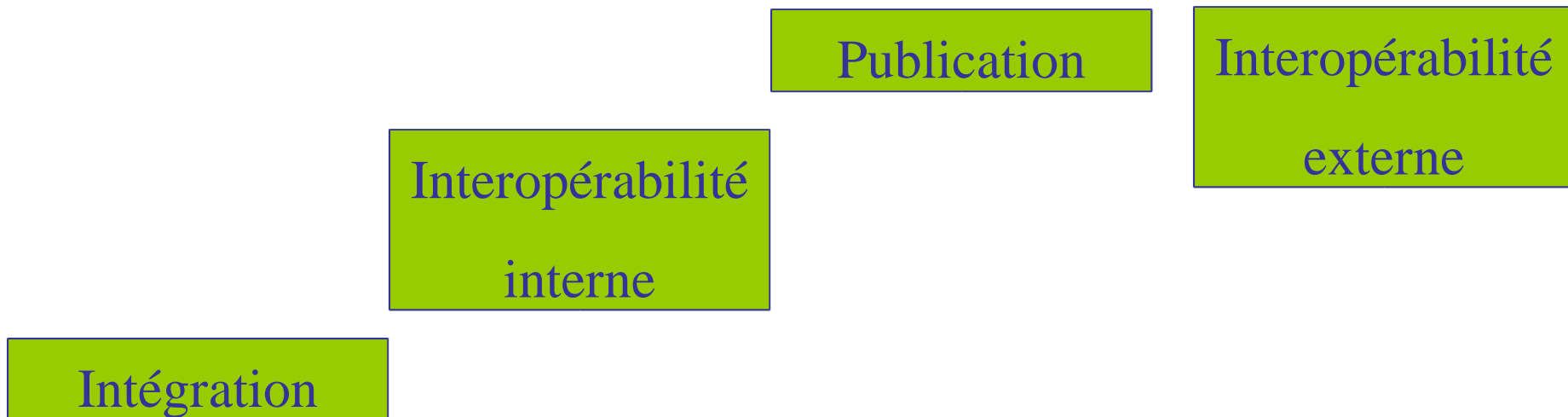
# IMGT : Prototypes de Services Web

- IMGT/LIGM-DB :
  - querySeqData
  - queryKnowledge
- IMGT/JCTA
- IMGT/V-QUEST
- IMGT/GeneFrequency
- IMGT/AlleleAlign
- BLAST

...Travaux en cours...

# Services Web à l'IMGT

- Interopérabilité entre les composantes du SI
  - Flexibilité des développements
  - Composants découplés
- Mise à disposition de nos compétences métier



# Services Web à l'IMGT

- Un Dialogue unifié :
  - Requête : Instance de document IMGT-ML
  - Réponse : Instance de document IMGT-ML

# Services Web à l'IMG

```
<querySeqData>
  <seqData>
    <simpleCatalogEntry
      numacc="M26678" />
    </seqData>
  </querySeqData>
```

```
<seqDataList>
  <seqData>
    <catalogEntry id="M26678" >
      <simpleCatalogEntry
        numacc="M26678"
        name="MMIGKZZZ"/>
    </catalogEntry>
    <identification>
      <partIdent
        moleculeType="DNA"
        configuration="rearranged">
        <taxon taxonName="Mus musculus"/>
      </partIdent>
    </identification>
    <classification>
      <group name="IGKV">
        <subgroup name="IGKV8"/>
      </group>
    </classification>
    ....
  </seqData>
</seqDataList>
```

# Services Web à l'IMGT : IMGT-ML

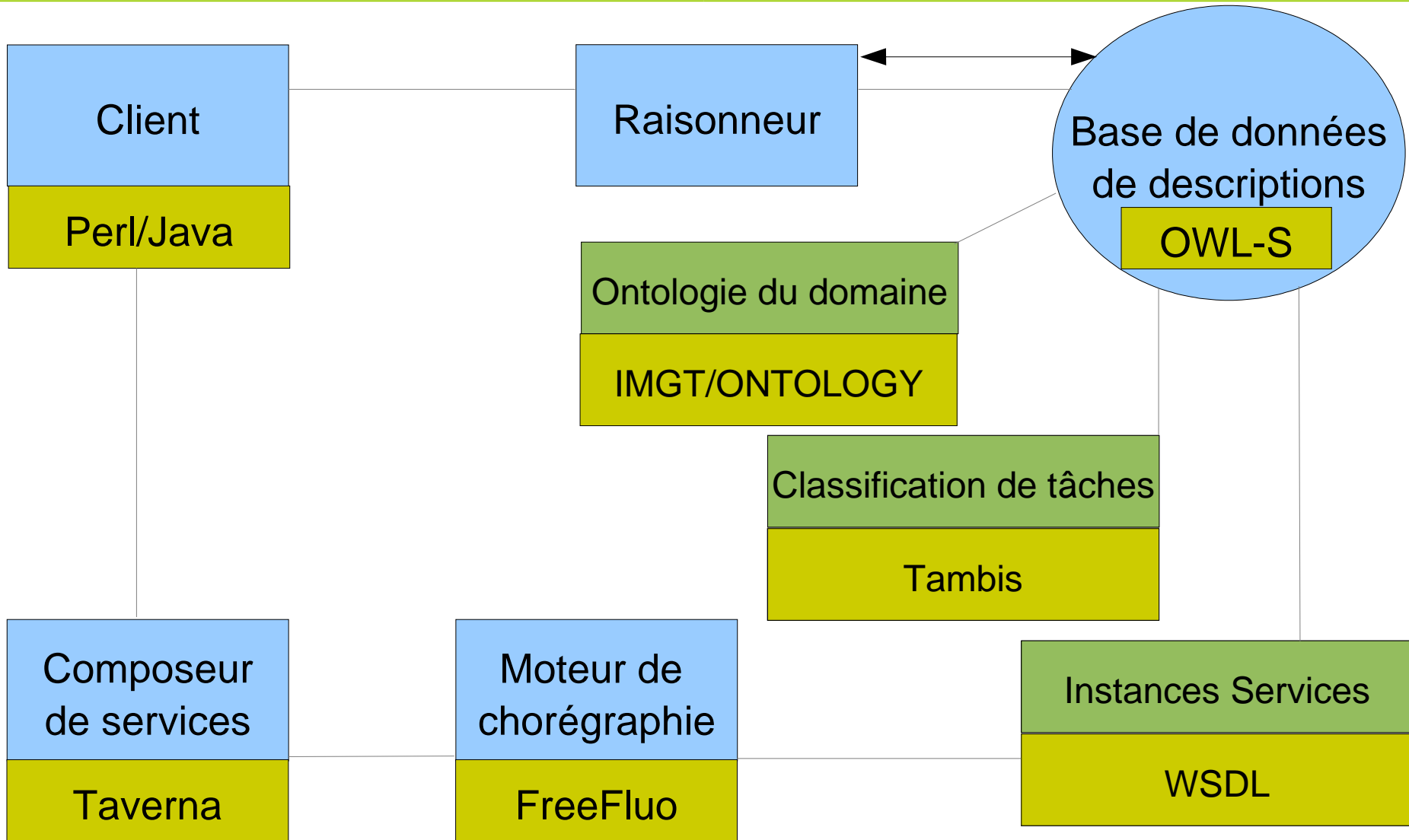
- Motivations :
  - 1 unique formalisme
  - extensibilité
  - ne pas lier le client à une interface
  - le service extrait les paramètres pertinents lors de l'invocation
  - moins coûteux que des invocations RPC

## Services orientés documents

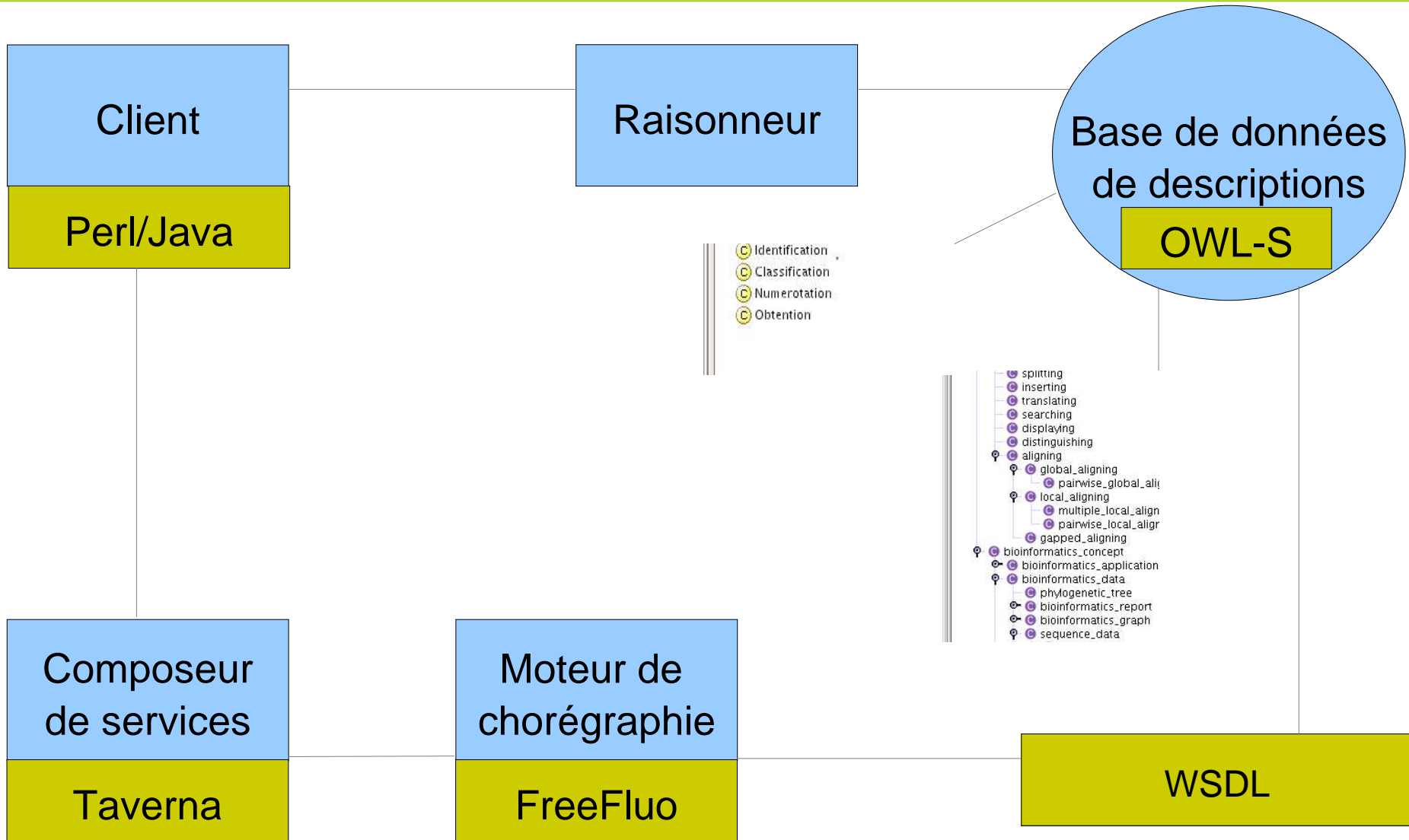
# Services Web à l'IMGT : Description

- Corrolaire : Une description peu verbeuse
  - RPC (Remote Procedure Call):  
public methode (param:String):String
  - Document :  
<xsd:any>
- Bottom-up (automatisation) ne fonctionne pas :
  - Java vers WSDL
- Approche Top Down :
  - Commencer par la description

# Systeme Prototype



# Systeme Prototype





# Merki !

?