Web Services

Concepts, composants et mise en pratique

Reynald Borer

IL2007 ////////
PPE ///////////
HEIG-VD /////

24 janvier 2007

Programme

- Introduction
- Concepts des services web
- Composants
  - XML, SOAP, WSDL, UDDI, ...
- Mise en pratique
  - Java, Python
- Autres technologies
  - CORBA, RMI
- Conclusion
- Questions ?

Introduction

- Contexte de cette technologie:
  - Lancé vers 1999 - 2000
  - Fait suite à l'émergence de XML (1ère norme: 1998)
  - Internet devient de plus en plus utilisé

- Idée de base: partager de l'information pure
  - HTML ? Contient de la mise en page => XML
  - Utiliser Internet, réseau des réseau, pour le partage
- Mise à disposition de services divers indépendants de toute plateforme et langage de programmation
- Emergence d'une nouvelle terminologie: **Services Web**

Programme

- Introduction
- Concepts des services web
- Composants
  - XML, SOAP, WSDL, UDDI, ...
- Mise en pratique
  - Java, Python
- Autres technologies
  - CORBA, RMI
- Conclusion
- Questions ?

Concepts

- Les **services web** englobent plusieurs concepts fondamentaux:
  - Interopérabilité
  - Indépendance de la plateforme
    - Interactions indépendantes du langage de programmation
  - Echange d'informations sans mise en forme
    - Echange de machine à machine
  - Réutilisabilité (exporter fonctions d'un logiciel en services web)
  - Flexibilité (couplage faible entre applications)

Programme

- Introduction
- Concepts des services web
- Composants
  - XML, SOAP, WSDL, UDDI, ...
- Mise en pratique
  - Java, Python
- Autres technologies
  - CORBA, RMI
- Conclusion
- Questions ?

Composants

- Technologie utilisant plusieurs composants différents
- Utilise des technologies standards
  - Technologies maîtrisées
  - Temps d'adaptation d'une application très court
- Technologie mise en place avant standardisation
  - Problèmes de compatibilité entre les implémentations
  - Actuellement résolus

Composants

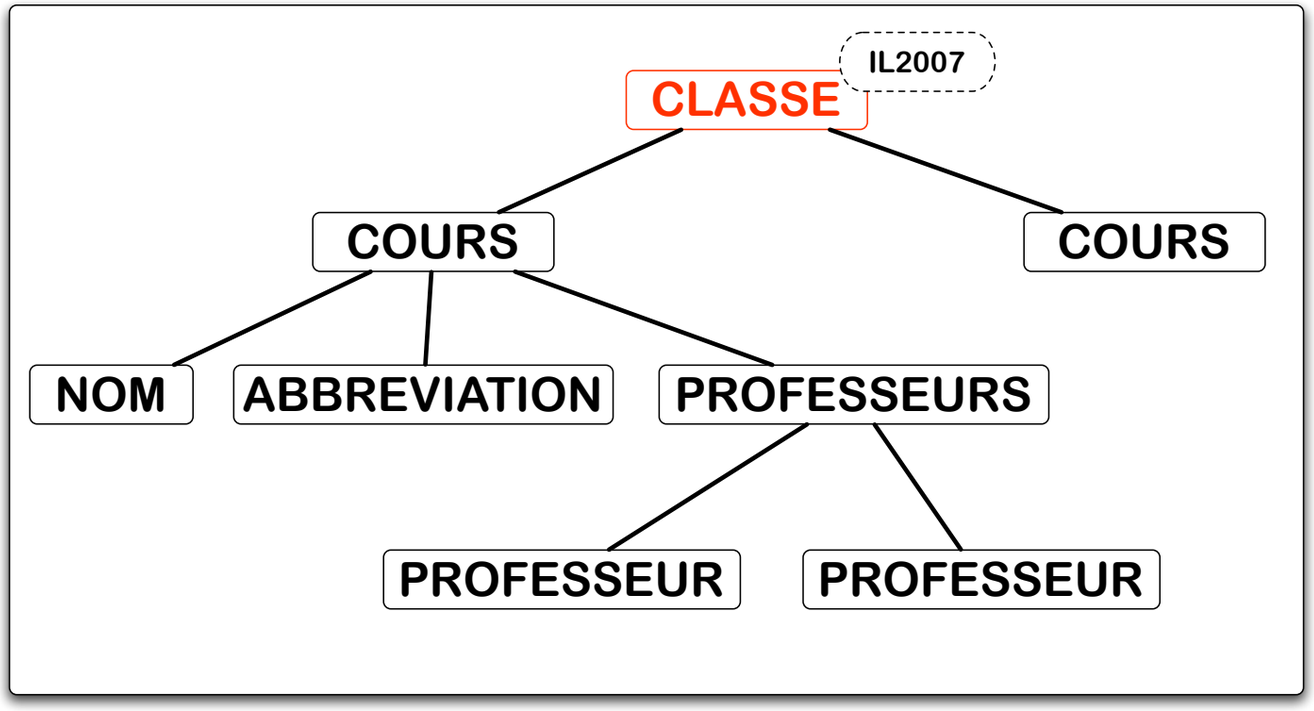
XML

- XML fournit la brique de base
  - XML: langage de balisage générique sous forme textuelle
  - Standard du W3C (10 février 1998)
  - Structure de balises en arbre
  - Orienté pour contenir des données
  - Facilite l'échange automatisé de contenu
- Un document XML est définissable et validable par un Schéma XML
  - Ce schéma peut être contenu dans le document
  - Et peut définir des types *complexes*

Composants: XML

exemple de représentation

```xml
<?xml version="1.0" encoding="UTF8"?>
<classe nom="IL2007">
    <cours>
        <nom>Présentations Personnelles</nom>
        <abreviation>PPE</abreviation>
        <professeurs>
            <professeur>Pierre Breguet</professeur>
            <professeur>Abdelali Guerid</professeur>
        </professeurs>
    </cours>
    <cours>
        <nom>Bases de Données</nom>
    </cours>
</classe>
```

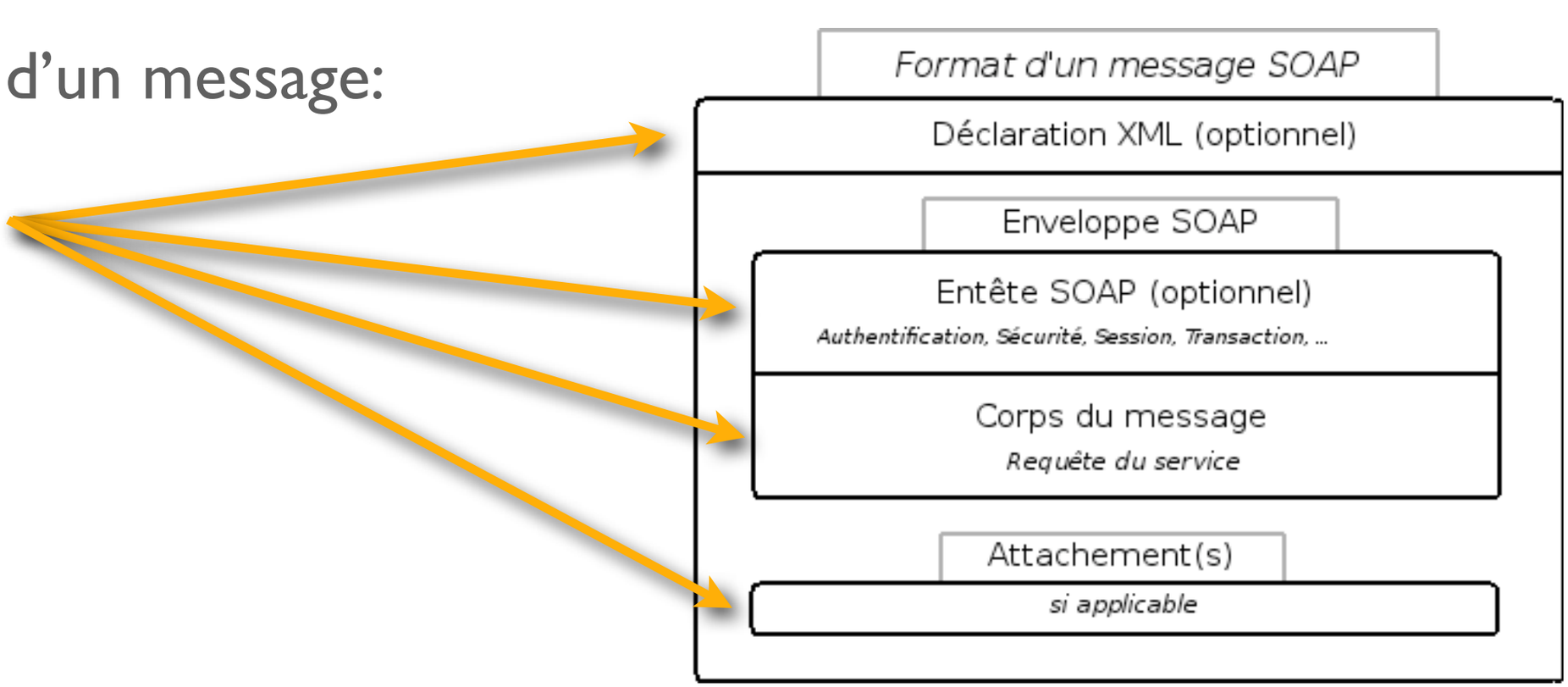


Composants

SOAP

- Echange de messages au format XML
  - basé sur d'autres protocoles de communication (HTTP, SMTP)
- Utilisé généralement pour l'invocation de méthodes distantes

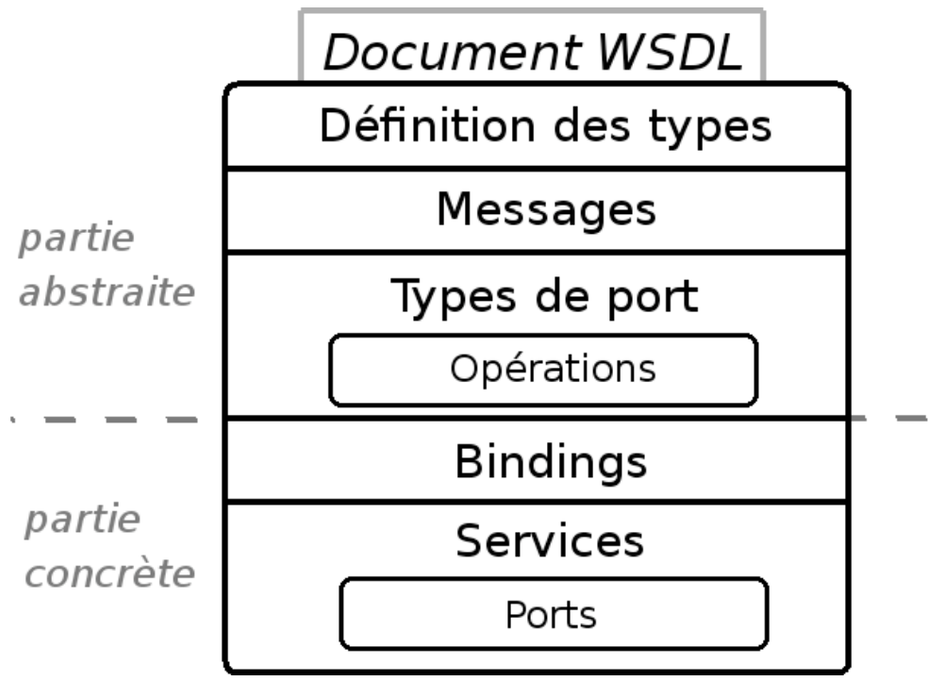
- Format d'un message:



Composants

WSDL

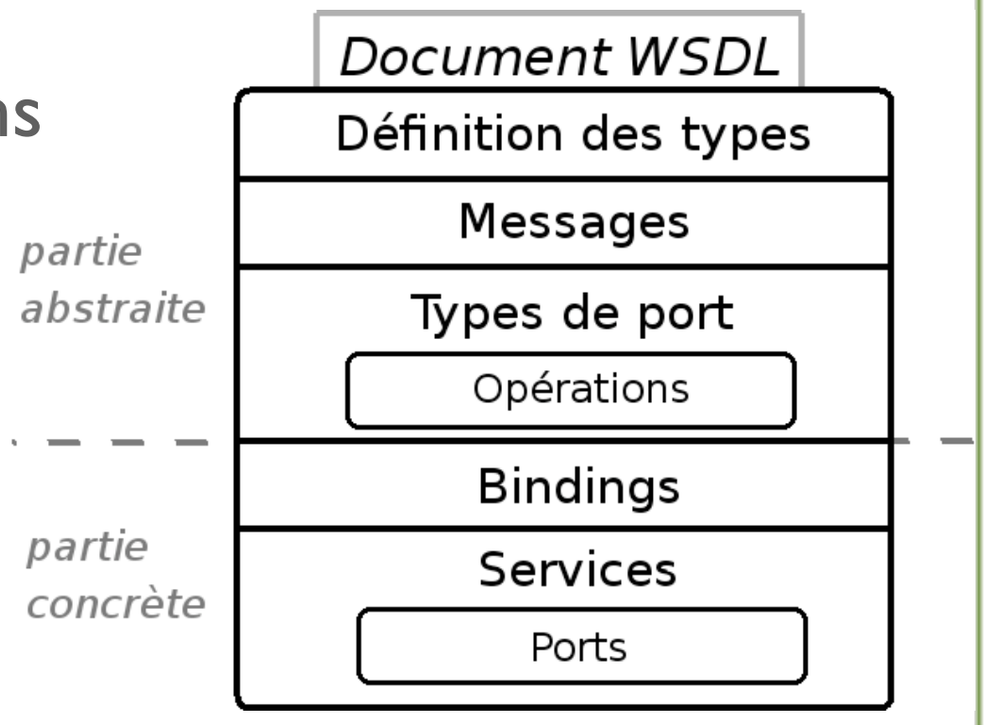
- Description des services web (basé sur XML)
  - Méthodes
  - Messages
  - Types des paramètres
  - Transport des messages
  - ...
- Décomposé en deux parties:
  - Partie abstraite
  - Partie concrète
- Facilite la réutilisation



Composants

WSDL

- Partie abstraite
  - Types
  - Messages (paramètres)
  - Types de ports => liste d'opérations
  - Opération définit les messages échangés
- Partie concrète:
  - Binding => format et transport
  - Services => ensemble de ports + adresse de communication



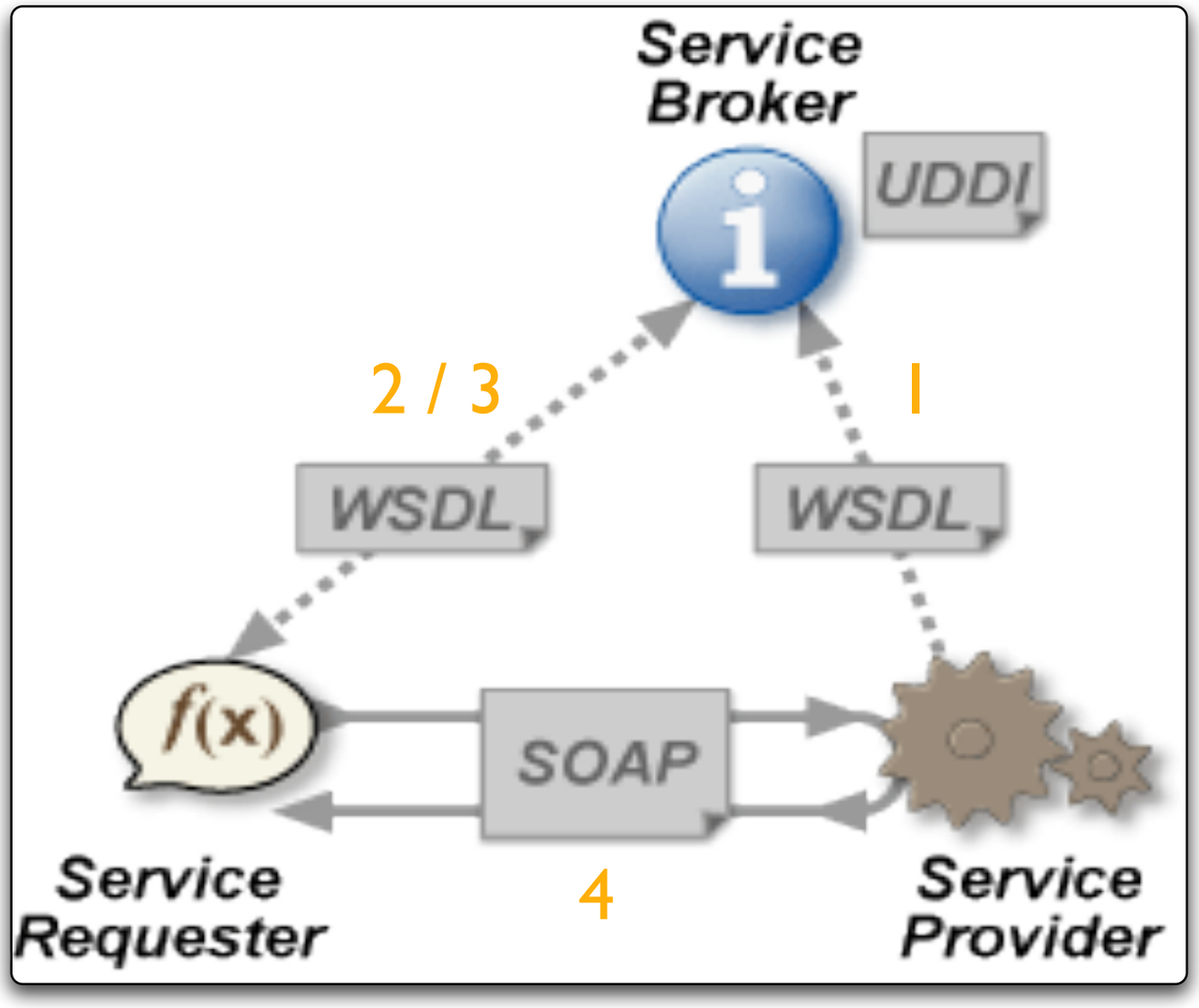
Composants

UDDI

- Annuaire des services web pour les entreprises (lancé en 1999)
  - Publication
  - Recherche
- Messages SOAP pour la communication
- Stocke les documents WSDL des services web

- 3 schémas d'accès:
  - Pages blanches: nom et informations d'une entreprise
  - Page jaunes: liste des services web (WSDL)
  - Pages vertes: informations techniques sur les services fournis

1. publication
2. recherche
3. réponse
4. invocation



Vue globale

Autres composants

- Composants de base des services web
- Modulaire pour faciliter autres normes et technologies
- Exemples d'autres normes :
  - WS-Addressing: messages SOAP bi-directionnelle
  - WS-Security: authentification, intégrité, cryptage
  - XRI: localisation d'un service web
  - WS-Reliability: garantie de réception des messages
  - Et plein d'autres

Programme

- Introduction
- Concepts des services web
- Composants
  - XML, SOAP, WSDL, UDDI, ...
- Mise en pratique
  - Java, Python
- Autres technologies
  - CORBA, RMI
- Conclusion
- Questions ?

Mise en pratique

JAVA

- Librairie utilisée: **Axis2** (version 1.1.1)
  - librairie libre la plus aboutie
- Points clés:
  - Modules pour ajouter des fonctionnalités
  - Correspondance XML ↔ Java selon plusieurs types de données
  - Plusieurs schémas d'échange possible
  - Interactions synchrones et asynchrones
  - Transformation d'une classe Java en service web
  - Multiples outils fournis pour faciliter la création de services web

Mise en pratique

JAVA

- Exemple pratique: service web avec POJO (Plain Old Java Objects)

1. Créer une classe Java quelconque
2. La compiler et la placer dans la structure de répertoire ci-contre
3. Modifier le fichier XML *services.xml*
4. Créer une archive *jar* (extension .aar)
5. Déployer l'archive avec l'interface d'administration de Axis2


```
calculatrice en notation polonaise inversée
rmée, grâce à Axis2, en service web */



{
= new Stack(); /** pile de nombres */

dans la pile */
```

Mise en pratique

PYTHON

- Librairie utilisée: **ZSI** (version 2.0 rc3)
  - encore en développement
- Points clés:
  - Implémente la dernière norme SOAP (1.1)
  - Conversion types des messages ↔ types Python
  - Supporte les messages avec attachements
  - Fournit plusieurs utilitaires de conversion WSDL ↔ Python
  - Transport laissé à d'autres modules spécialisés

Mise en pratique

PYTHON

- Client pour un service web de chiffrement avec le carré de Vigenère

Utilise l'utilitaire *wsdl2py* afin de convertir le document WSDL en classes Python

```python
#!/usr/bin/python
# -*- coding: utf-8 -*-                                                    #
#                                                                          #
# vigenere.py                                                              #
# Un simple exemple d'un appel à un web services en pyhon (le service web #
# chiffre un texte avec le carrè de vigenère)                              #

# importation des fichiers générés par WSDL2Py
from Vigenere1_services import *
import sys

def main(args):
   # instance d'un proxy pour les requêtes
   loc = Vigenere1Locator()
   port = loc.getVigenere1Soap()

   print '-- Chiffrement avec le carré de Vigenère par web services --'
   print 'Chiffrement de', sys.argv[1], 'avec', sys.argv[2], 'comme cle'

   # nouvelle requête de cryptage
   cipher = CipherSoapIn()
   cipher.set_element_text(args[1])
   cipher.set_element_key(args[2])

   # appel de la procédure distante
   resp = port.Cipher(cipher)

   # résultat
   print 'Texte chiffré:', resp.CipherResult

if __name__ == '__main__':
   if len(sys.argv) < 3:
      sys.exit('Usage : vigenere.py texte cle')
   main(sys.argv)
```

Programme

- Introduction
- Concepts des services web
- Composants
  - XML, SOAP, WSDL, UDDI, ...
- Mise en pratique
  - Java, Python
- Autres technologies
  - CORBA, RMI
- Conclusion
- Questions ?

Autres technologies

CORBA

- Norme permettant à des composants logiciels de communiquer
  - communication entre plusieurs machines
  - indépendance du langage
  - composants assemblés afin de fournir une application complète

- Interface essentiellement objet (interface en langage IDL)
- Utilise ses propres protocoles réseau (IIOP et HTIOP)
- Multitude de composants pour les fonctionnalités
- Principes incluent transparence, héritage, ...

Autres technologies

CORBA

- Comparatif avec les services web:

Avantages:	Inconvénients:
• Fonctionnalités (orienté objets) • Performance (pas de XML) • Sécurité	• Mise en place • Couplage clients - serveurs

Autres technologies

RMI

- Interface de programmation afin d'invoquer des objets distants
  - spécifique à Java
- Objets distants perçus comme objets locaux
- Nécessite l'emploi d'un registraire sur chaque serveur
- Utilise IIOP pour la communication
- Communication en couches pour interopérabilité entre programmes
- Possibilité de transférer directement des objets
  - et de récupérer dynamiquement l'interface de ces objets

Autres technologies

RMI

- Comparatif avec les services web:

Avantages:	Inconvénients:
• Fonctionnalités (transport d'objets) • Performance • Sécurité (natif à Java) •Communication transparente	• Mise en place • Couplage clients - serveurs

Programme

- Introduction
- Concepts des services web
- Composants
  - XML, SOAP, WSDL, UDDI, ...
- Mise en pratique
  - Java, Python
- Autres technologies
  - CORBA, RMI
- Conclusion
- Questions ?

Pour conclure

- Services web fournissent une solution élégante au besoin de partage de l'information
- Des librairies complètes pour une multitude de langages existent
- Mais difficile de juger de l'utilisation de cette technologie
  - Google est passé des services web à Ajax
- Emergence de l' "Architecture Orientée Services"
  - Prévisions: utilisé par 75% des projets en 2008
- Avantage de taille: couplage très faible clients et serveurs
  - grâce à XML entre autre

Programme

- Introduction
- Concepts des services web
- Composants
  - XML, SOAP, WSDL, UDDI, ...
- Mise en pratique
  - Java, Python
- Autres technologies
  - CORBA, RMI
- Conclusion
- Questions ?

Merci de votre attention

Des questions ?