

Cours Middleware

Introduction au middleware ou intergiciel

Cours Middleware

- Introduction
- Corba
- J2EE/RMI
- Servlet/JSP
- Web Services
- Grid, temps -réel, surprise

Equipe pédagogique

- Bart Lamiroy
- Laurent Ciarletta
- Rémi Badonnel
- Vincent Cridlig
- ...

Prénom.Nom@loria.fr

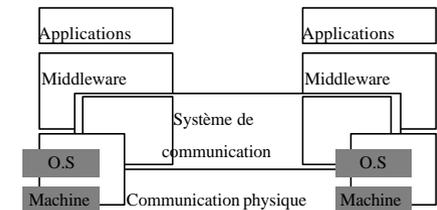
Middleware : définitions (1)

- L'intergiciel (middleware en anglais) est un ensemble de logiciels ou de technologies informatiques qui servent d'intermédiaire entre les applications et le transport des données via le réseau. Ils offrent des services de haut niveau liés aux besoins de communication des applications (temps réel, sécurisation, sérialisation, transaction informatique, etc.).
fr.wikipedia.org/wiki/Middleware
- Couche logicielle intermédiaire entre les applications et le réseau, permettant le dialogue entre des applications hétérogènes.
Synonyme : Intergiciel.
lexicom.free.fr/lexicommo.htm

Middleware : définitions (2)

- Logiciel standard qui permet à deux autres logiciels de communiquer en utilisant un protocole applicatif. Dans un Centre d'appels, prend en charge les communications entre les différents constituants : ACD, SVI, applicatif...
www.digirway.fr/html/glossairejm.htm
- Un middleware permet la communication entre des clients et des serveurs ayant des structures et une implémentation différentes. Il permet l'échange d'informations dans tous les cas et pour toutes les architectures. Enfin, le middleware doit fournir un moyen aux clients de trouver leurs serveurs, aux serveurs de trouver leurs clients et en général de trouver n'importe quel objet atteignable.
cynode.projet-enoch.net/resource/xml/doc/memoire/generated_html/memoire_g1107.html

Le Middleware



Pour quoi faire?

- Interface de haut-niveau (API) pour le développement d'applications
- Cacher hétérogénéité du système sous-jacent (matériel et logiciel)
- Cacher les problèmes de la répartition au maximum
- Fournir des services répartis d'usage courant

Faciliter la programmation répartie

Faciliter la programmation répartie

- Développement, évolution, réutilisation des applications
- Portabilité des applications
- Interopérabilité d'applications hétérogènes

Aspects importants

- Applications en constante évolution (besoins, architectures, ressources etc...)
 - QoS
 - Capacité d'évolution
- Mobilité, dynamisme des environnements/utilisateurs

- Adaptation
 - Découverte de service
 - Reconfiguration
 - Comportement
 - Réutilisation, simplicité etc ..
 - Composants
- Dynamique/adaptable

Applications réparties : Système et Application

- Système :
 - gestion
 - des ressource communes
 - De l'infrastructure
 - Lié au matériel et logiciels sous-jacents
 - Système
 - D'exploitation
 - De communication
 - Cacher la complexité du matériel, des communications :
fourniture de services d'un haut niveau

Applications réparties : Système et Application

- Application :
 - Réponse à un problème spécifique
 - Fournir des services à des utilisateurs (personnes, autres applications)
 - Utilisant les services généraux offerts par le système
- Certaines applications travaillent directement sur le matériel (systèmes embarqués, réseaux de capteurs etc...

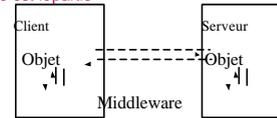
Panorama des architectures et technologies

Plan

- Contexte
- Vision OMG
- Vision .NET
- Vision Sun
- Les Services Web/Web Services

Contexte

- L'informatique d'entreprise est répartie
 - Applications distribuées
 - Objets répartis
- Divers moyens d'accès en mode « client-serveur »
 - Mobile, Internet, Minitel, Téléphone
- Besoin d'architecture ouverte et adaptative
- Vers « l'entreprise étendue »



Contexte

- Passage du Web des utilisateurs/clients à un Web inter-système
 - Client, B2C, B2B, EAI etc...
- Modèles qui évoluent :
 - Client/serveur -> Grid, P2P

Architecture orientée service

- SOA (Service-Oriented Architecture)
 - Pas nouveau
 - Permet l'interopérabilité de systèmes hétérogènes
 - Couplage « lâche » ou mou entre les composants logiciels
 - Service = action exécutée par un « fournisseur » pour un « client »
 - Différence avec l'orienté objet :
 - Découplage données – mode de traitement
 - Ce n'est plus le « technique » qui dicte l'architecture, mais le « métier »

3 principales approches

- Autour de Corba : informatique d'entreprise, intégration d'applications 'legacy'
- Autour de Java : Internet et Web
 - Application et Web Serveurs
- COM+ , .NET : position forte de MS sur les postes de travail

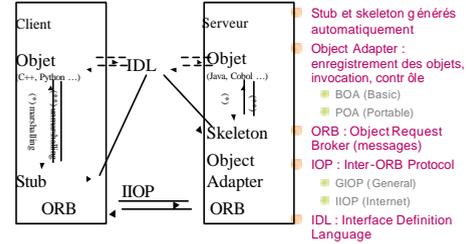
CORBA

L'OMG Object Management Group

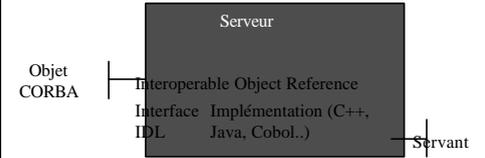
- Créé en 1989
- But non lucratif
- Plus de 850 membres
- Création et maintenance de spécifications:
 - CORBA
 - UML
- www.omg.org

Common Object Request Broker Architecture

Une architecture d'interopérabilité distribuée et orientée objet selon un modèle client-serveur ouvert



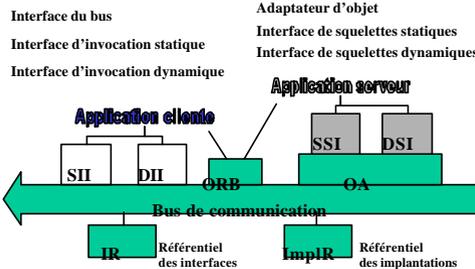
Objet Corba



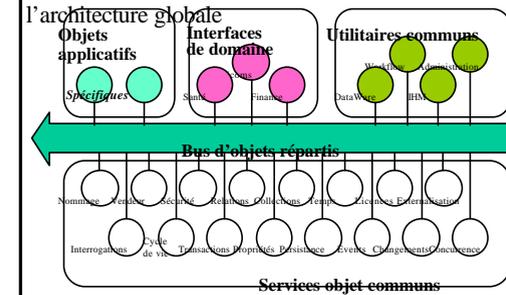
CORBA : le mode dynamique

- Un « référentiel des interfaces » stocke sous forme d'objets les descriptions d'interfaces OMG-IDL.
- Une API (DII: Dynamic Invocation Interface) permet de construire des requêtes à l'exécution.

CORBA : les composants du bus



OMA (Object Management Architecture)



CCM (Corba Component Model)

- Extension logique des EJB (interopérabilité avec les EJB)
- Composant « emballés »
 - Description XML du contenu
 - Binaires (différentes plateformes)
- 4 catégories de composants
 - Service
 - Session
 - Entity
 - Process
- CIF (Component Implementation Framework)
 - Utilise CIDL (Component Implementation Description Language) pour générer le code complémentaire au code du composant
 - Chaque composant est placé dans un « container »

Vers CORBA 3

- CORBA 2.2
 - client/serveur orienté objet, les interfaces de base et le POA, les mécanismes dynamiques, GIOP, OMG-IDL et ses projections vers C, C++, SmallTalk, COBOL, ADA et Java.
- CORBA 3.0
 - interfaces multiples, passage par valeur,
 - modèle de composant, langage de script,
 - minimumCORBA, realtimeCORBA, CORBA/COM/DCE
 - messaging, printing, fault tolerance, firewall

Implémentations

- BEA WebLogic
- IBM WebSphere
- IONA Orbix
- Borland Enterprise Edition

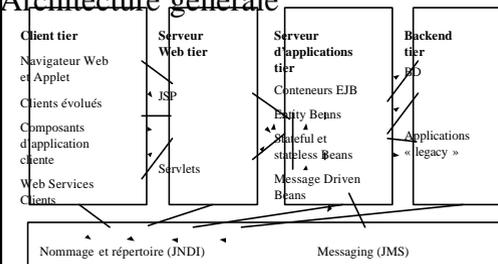
- MICO
- OmniORB
- ORBit

CORBA, UML, XML et MDA

- UML = standard OMG
- W3C DOM (Document Object Model) :
 - Accéder à XML via des interfaces
 - Se fonde sur l'IDL OMG
- Types Corba passés par valeurs mappés sur du XML
- CCM : encodage, configuration et déploiement décrits en XML
- Etc...
- Evolution vers le MDA (Model Driven Architecture)
 - Spécifications écrites à 2 niveaux
 - PIM (Platform-Independent Model)
 - PSM (Platform-Specific Model)
 - Mapping PIM vers PSM
 - Processus business et entités modélisés au niveau PIM
 - Génération automatique avec les mappings
 - « englober CORBA, J2EE, XML, .NET et les autres technologies »

J2EE

Sun's Java 2 Enterprise Edition Architecture générale



Composants dans l'univers Java

- Applets (J2SE)
- JavaBeans (J2SE)
- Enterprise Java Beans (J2EE)
- Servlets/JSP (J2EE)
- Composants d'application cliente (J2EE)

Applets et JavaBeans

- Applets :**
 - Composants légers
 - Téléchargeables pour les clients (navigateurs Web)
 - Modèle de sécurité fort
- JavaBeans :**
 - Programmation « connection-oriented » (« wiring »)
 - Flux d'événements entre des sources et des « listeners »
 - Clients (le plus souvent) et serveurs
 - Support pour la conception visuelle d'applications

Servlets/JSP

- Servlet :**
 - Esprit des applets, côté serveur
 - Composants légers instanciés par les serveurs Web
 - Exécuté
- JSP :**
 - « code Java et HTML entrelacés »
 - Pour la présentation
 - Définition déclarative : génération automatique de pages Web (et des servlets correspondantes)
 - Interprété

EJB

- Services intégrés dans des conteneurs d'EJB Beans**
 - Nécessité de déclarer leurs attributs
 - Besoin de « descripteurs de déploiement »
 - Différent des Java Beans ! :
 - Composition « orientée objet »
 - Création d'instances, appels de méthodes
 - Pas très adapté au « wiring »
 - Point fort : composition contextuelle

EJB

- 4 types de E-Beans**
 - Stateless Session
 - Traitement, workflow
 - Sans état
 - Stateful Session
 - Traitement, workflow
 - Avec état
 - Entity
 - Mémoriser un composant métier
 - Partage entre plusieurs utilisateurs
 - Message Driven (depuis EJB 2.0)
 - (MOM) Middleware orientés message : point à point ou publication/souscription

Application client components

- Applications non contraintes côté client
- Utilisant JNDI pour accéder aux propriétés de l'environnement, aux EJB, et ressources sur les serveurs

Directions explorées par Sun

Jini

- Fédérations de clients et de services Java (protocole de découverte de services)
 - Service de noms, transactions réparties, événements

JXTA

- « juxtaposition »
- P2P

Les composants selon Microsoft

- Chemin choisi : construire ses propres applications et plateformes, et les modifier continuellement (pas de standard global)**

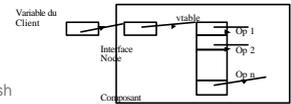
Composants s'améliorent au fur et à mesure

- VBX (non orienté-objet!), OLE (Object Linking and Embedding), ...
- ActiveX, ASP (Active Server Page).

Récemment :

- Focus sur l'Internet et le Web (IETF, W3C)
- Standards ECMA pour les spécifications .NET (CLI et C#)

COM



Modèle initial : COM

- Microsoft et Macintosh
- Tierces parties (HP)

A influencé d'autres modèles (XPCOM de Mozilla, CCM de l'OMG)

Standard « binaire »

- Pas de description de la liaison avec des langages de programmation particuliers
- Objets : ni favorisés ni déniés
- Seule définition : l'interface

DCOM, OLE/ActiveX, COM+

DCOM

- Distributed COM
- Client-Side Proxy et Server-Side Stub :
 - interprocessus dans COM, inter-machine
 - Mapping des types simples de données vers et depuis des flux d'octets
 - Standard binaire

OLE

- Object Linking and Embedding
- Combiner les applications « legacy »
- Paradigme « document-centric »
- Collection prédéfinie d'interfaces COM

DCOM, OLE/ActiveX, COM+

OLE Control

- VBX, OCX puis ActiveX
- Composants COM implantant une des interfaces prédéfinies (serveurs de documents et événements essentiellement)
 - OCX : très lourds
- ActiveX : certaines interfaces sont optionnelles, interfaces « sortantes » et propri étés (modifiables par l'application ou l'utilisateur : Look and Feel)

COM+

- transactions, messages asynchrones, clustering, équilibrage de charge

.NET

CLR (Common Language Runtime)

- Implantation de :

CLI (Common Language Infrastructure) : spécification ECMA

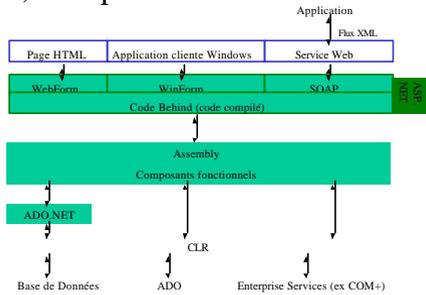
- Continuation/interopérabilité de/avec COM
 - Chargement/déchargement dynamique, ramasse-miette, reflection, persistance etc... indépendamment du langage
- C#, Jscript, Managed C++, VB .NET

Modules en CIL (Common Intermediate Language)

- Sorte de bytecode

Assemblies : unités de déploiement (sorte de fichiers jar)

.NET, Composants



.NET

☑ Cible à 3 niveaux

- Services Web
- Plateformes de déploiement (côtés clients et serveurs)
- Plateforme de développement

Evolution

- ☑ Interopérabilité 2 à 2 des différentes visions
- ☑ Composants « tout XML »

Quelques chiffres

☑ Gartner (USA Août 2004)

- 40% Java/J2EE, 41% .NET
- Outils : 24/25% Microsoft/IBM ... Open Source 5%
- Travail sur les Web Services : 50%
- Travail sur le SOA : 30%
- .NET perçu comme « solution pour Web Services » par 70%
 - 49% : solution complète et intégrée

☑ IDEO (France sept 2004)

- >60% Java/J2EE / <40% .NET
- Frameworks J2EE
 - Utilisés par 56%
 - 68% Open Source, 1/3 interne, 17% commerciaux
- Web Services :
 - 50% intéressés, 18% projet pilote
 - 40% promoteur, 38% manque de recul

XML

L'émergence d'XML

- ☑ eXtensible Markup Language, W3C (1998)
- ☑ Représentation de données (partiellement) structurées
- ☑ BD :
 - organisation plutôt autour de tables
- ☑ Comparé à XML :
 - plutôt arbres, mais autres structures aussi
- ☑ Si on a des données non-structurées : y faire référence depuis le document XML

XML

Représentation de données

- Messages
- Pages Web
- Documents traditionnels
- Données de configuration

Langage commun (syntaxe)

Éléments XML

- éléments, attributs XML, texte non structuré

Racine unique du document

Principes

```
<<course>
  <titre> XML Web Services </titre>
  <chapitre>
    <numero> 1 </numero>
    <titre> States de XML </titre>
    <contenu> quelques informations </contenu>
  </chapitre>
</course>
```

Ensemble non fini de balises

- L'utilisateur peut créer de nouvelles balises

Définition de grammaires : XML est un Meta-Langage

- MathML, NewsML, XMI, Doc, Slides, ...

Séparation de la forme et du fond

- Un document XML peut être constitué de deux entités (le fond et la forme)

Grammaire

Deux façons de définir une grammaire XML :

DTD (Document Type Description)

- Non extensible
- Langage de définition de grammaire XML
 - Contraintes sur la validité d'un document
- Largement utilisé
- Expression faible (type, structure)

XML Schéma

- Langage XML de définition de grammaire XML
 - Extensible
- Interaction avec autres standards XML (XML Namespace et Xpath)
- De + en + utilisé
- Expression puissante (type, structure, héritage)

Un document XML est dit valide lorsqu'il est conforme à une grammaire

XML Namespace : Espaces de noms

- Mécanismes permettant de partitionner les balises XML (permet d'avoir deux fois le même nom de balise)
- Un espace de nom est défini dans n'importe quelle balise par l'attribut xmlns et par une URI.
- Dans un document XML, un espace de noms est identifié par un nom logique, les balises appartenant à cet espace doivent alors être préfixées par ce nom logique.
- Ex :

```
<meta:body xmlns:meta="http://meta.lip6.fr/meta/"
```

Pourquoi XML?

- Standard W3C
- La syntaxe XML ne contient que peu de mots clefs : Simplicité
- XML est indépendant des plates-formes : Portabilité
- XML est un méta-langage, il est possible de créer ses propres balises : Extensibilité
- Outils disponibles (et gratuits)

Largement utilisé pour les échanges inter-applications

XML-Web Services : un nouveau Middleware

« les Web Services »

Limitations des middlewares

Passage à large échelle : Web

⊕ Protocoles hétérogènes

- IIOP, RMI, DCOM
- Firewall

⊕ Pas d'ouverture des services

- Notion de moteur de recherche inexistante

⊕ Trop de contraintes sur le client !

- Doit posséder les souches
- Difficulté de construire dynamiquement

Limitations des middlewares

Inconvénients intrinsèques

⊕ Complexité

- CORBA : IDL, Mapping, ...
- EJB : Container, JNDI, ...

⊕ Pérennité : remise en question

- CORBA, EJB, .Net, ...

⊕ Prix

- Plates-formes
- Compétences

Solutions existantes

⊕ Modification du Protocole

- RMI / IIOP

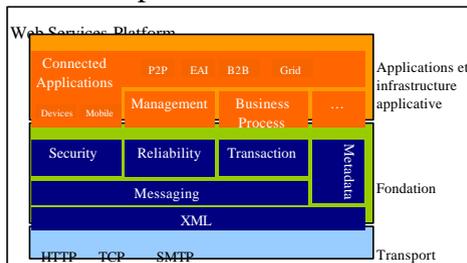
⊕ Passerelles

- CORBA vers DCOM

⊕ Portage d'applications existantes difficile

⊕ Solutions non standards

Plateforme pour les WS

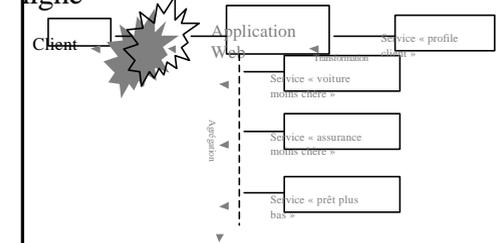


XML

⊕ Besoin de connectivité, d'interaction, d'interopérabilité, intra et inter-organisation dans le cadre du Web

⊕ Donc ce sont les standards du Web et donc le XML et le modèle de Web Service qui fournissent une couche d'intégration prometteuse, pour toutes les technologies y compris celles des composants

Exemple : achat de véhicule en ligne



les Web-Services

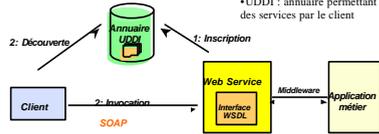
- Applications modulaires, autodescriptives, publiables, invoquées par des protocoles du Web.

Acteurs

- Client (invoque)
- Fournisseur (service)
- Annuaire (informations)

3 spécifications associées basées sur XML

- SOAP : invocation des services (transport des données)
- WSDL : description des services offerts (méthodes invocables)
- UDDI : annuaire permettant la découverte des services par le client



Écoles Mines de Nancy – SI151- 2005-2006

61

Approche Envisagée

Un nouveau protocole : SOAP

- Basé sur XML
 - Portabilité, Hétérogénéité
- Porté par des protocoles large échelle existants
 - HTTP, SMTP, ...

Paradigme orienté service : WSDL

- Définition de services offerts (en XML)

Découverte automatique des services (dynamicité) : UDDI

- Référentiel de Web Service (Pages Jaunes, Vertes, Blanches)

Écoles Mines de Nancy – SI151- 2005-2006

62

Vers SOAP

Remote Object Invocation

XML-RPC

SOAP (Simple Object Access Protocol)

- L'adresse de l'invocation
- Embarque une large gamme de types de données dans les messages d'invocation
- Décrit les parties obligatoires et optionnelles

Écoles Mines de Nancy – SI151- 2005-2006

63

XML Web services

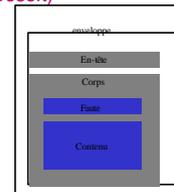
- WSDL, UDDI, WSFL, XLANG
- Reposent sur SOAP
- Services offerts par des serveurs Web
 - Pas pour les individus
 - Mais pour d'autres systèmes
- WS-I Web Service Interoperability (WS-I Organisation)

Écoles Mines de Nancy – SI151- 2005-2006

64

SOAP

- Protocole d'échange de messages (client / serveur)
- Basé entièrement sur XML
- Standard W3C (Initiative IBM et Microsoft)
 - Actuellement SOAP 1.1/1.2
- Concepts
 - Message = Enveloppe
 - (Header + Body)
- Extensibilité
 - Porté sur HTTP, SMTP, FTP...



Écoles Mines de Nancy – SI151- 2005-2006

65

SOAP (exemple)

Requête

```
POST /trStock HTTP/1.1
Host: www.stock.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 341
<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2001/11/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/11/soap-encoding">
  <soap:Body
    xmlns:m="http://www.stock.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
      </m:GetStockPrice>
    </soap:Body>
  </soap:Envelope>
```

Écoles Mines de Nancy – SI151- 2005-2006

66

SOAP (exemple)

Réponse

```

HTTP/1.1 200 OK
Content-Type: application/soap charset=utf-8
Content-Length: nnn
<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Body
    xmlns:m="http://www.stock.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>
</soap:Envelope>
  
```

Description des Services Web

- WSDL : Web Service Description Language
- Langage de définition de Web Services
- Basé entièrement sur XML
- Standard W3C (Initiative IBM et Microsoft)
 - Actuellement WSDL 1.1
- Définition de l'interface, de l'URL et du port du Web Service.
- Utilise le système de typage de XML Schéma

Découverte de Web Service

- UDDI : Universal Description, Discovery and Integration
- Référentiel de définitions Web Service
- Permet de construire dynamiquement des clients
- Recommandation OASIS
- Référentiel défini lui-même en WSDL
- Référentiel Public / Privé

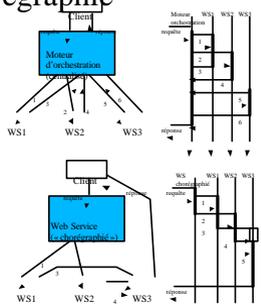
Orchestration/Chorégraphie

A partir du

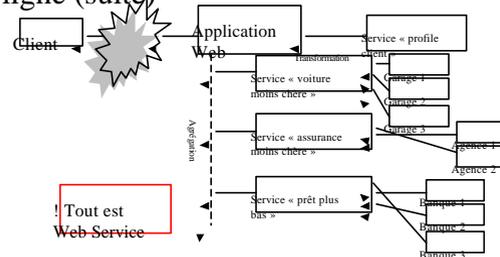
- Business Process Modeling (BPM)

2 visions :

- Orchestration : vision centralisée
 - BPEL (Business Process Execution Language) d' OASIS
 - Produits
- Chorégraphie : vision globale des interactions
 - W3C WS-CDL
 - Pas de produits



Exemple : achat de véhicule en ligne (suite)



Sécurité dans/pour les Web Services

Services de sécurité

- Authentification
- Autorisation
- Confidentialité
- Intégrité

Modèles proposés

- Pour XML
 - XML Signature
 - XML Encryption
 - SAML-XML (Security Assertions Markup Language)
 - XACML (Extensible Access Control Markup Language)
 - XKMS (XML Key Management Specification)
- Pour WS
 - WS-Security : messages SOAP (et WSDL) sécurisés

Niveaux de service

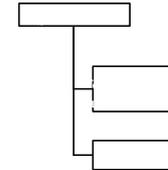
- SLA : Service Level Agreement
- SLS : Service Level Specification

Service Level Agreement



- Un SLA est un contrat entre un fournisseur et un client garantissant des niveaux de performance et de fiabilité à un certain coût.
- Les SLA peuvent être utilisés pour établir les besoins en disponibilité, fiabilité, temps de réponse, etc.

SLA / SLS : vue réseau



- Un SLA est établi entre un client et un fournisseur de service.
 - SLA intra-domaine
 - SLA inter-domaine
- Un SLS est la partie technique du SLA.
- Un SLS est utilisé par un fournisseur de réseau pour configurer les éléments de son réseau afin de ne pas violer le SLA.

Structure d'un SLA : application aux Web Services

- Désignation des parties
- Période de validité
- Services fournis
- Service Level Specifications
- Pénalités
- Reporting

WSLA

- Participants
- Définitions de service
- Obligations



Indicateurs de niveau de service pour les Services Web

- Temps de réponse
- Disponibilité
- Bande passante
- Niveau de sécurité
- Fréquence de pannes
- Temps de résolution d'une panne
- Temps d'enregistrement

Problèmes

- Négociation des paramètres
- Vérifications de la validité/ du respect des contrats
- Pénalités / compensation
- Sécurisation

Produits sur le marché

- BEA WebLogic8 et 9, framework Beehive
 - IBM WebSphere (6.0) et Atlantic
 - Microsoft .NET , WSE (Web Service Enhancement)
 - HP Netaction
 - Borland
 - Compuware
 - WebMethod ...
 - Sun One
- Concepts :
- « On Demand » d'IBM
 - « Agile » de Microsoft

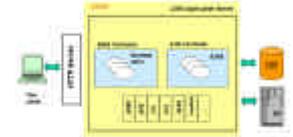
Et l'Open Source

- Apache Software Foundation
 - Axis
 - JaxMe (Java XML binding)
 - XML-RPC
 - jUDDI (Java UDDI)
 - SOAP
 - WSFX (Web Service Functionality Extension)
 - WSS4J (WS Security)
 - Etc ...
 - WSIF (Web Service Invocation Framework), WSIL (Web Service Inspection Language), WSRP4J (Web Service 4 Remote Portlet)
- BEA a offert Beehive à Apache

Et l'Open Source

- Pas d'administration ou de monitoring
- Jonas (Java Open Application Server - ObjectWeb) Serveur d'application « middleware opensource »

- Mono .NET



Business Process

- Commercial
 - Amberpoint, Actional, Blue Titan
- Open Source
 - JBPM (définitions de processus basés sur UML/J2EE)
 - Open Business Engine (norme WiMC workflow management coalition), J2EE
 - Werflow (BPML et BPELWS)
- Etc...

Directions et conclusion

- Intégration des architectures existantes parallèlement aux Web Services
- Mise en œuvre de la logique métier
- Clustering :
 - Équilibrage de charge
 - Réplication de session
- Prometteur mais encore immature
 - Standard
 - Interopérabilité
 - Réutilisation
 - Assemblage souple de composants
- Mais la différenciation se fait par des « services » non standards

Références

- Clemens Szypersky
 - « Component Software, Beyond Object Oriented programming »
- Sacha Krakowiak
- Deepak Alur et al.
 - « J2EE et les Design Patterns »
- Xavier Blanc (transparents de cours)
- Christophe Bouthier (transparents de cours)
- Francine Krief (transparents)