Segurança Computacional no Desenvolvimento de Web Services

Júlio Cézar Estrella, Douglas Rodrigues, Kalinka R. L. J. C. Branco, Regina H. C. Santana, Marcos José Santana

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

{jcezar,douglasr,kalinka,rcs,mjs}@icmc.usp.br

OWASP

28/10/2009

APOIO: ***CNPq, OWASP Foundation***

The OWASP Foundation
http://www.owasp.org

Roteiro

1. Introdução
   - Mudança de foco
   - Evolução dos negócios
   - Problemas
   - SOA
   - Web Services
1. Segurança
   - Visão Geral
   - Vulnerabilidades
   - Pilha TCP/IP
   - Web Services
   - Políticas para Web Services

Roteiro

1. Axis2
   - Visão Geral
   - Componentes
   - Modelo de Processamento
   - Modelo de Implantação
   - Ciclo de vida de um serviço
   - Instalação e configuração do ambiente de programação com Axis2
     - Donwload dos fontes
     - Descompactação dos fontes
     - Configuração do classpath
     - Teste das configurações

Roteiro

1. Desenvolvimento de aplicação segura com Axis2
   - Criação do serviço
   - Criação de políticas de segurança
   - Geração da WSDL
   - Geração do cliente a partir da WSDL
   - Compilação e geração do código
   - Discussões sobre a aplicação

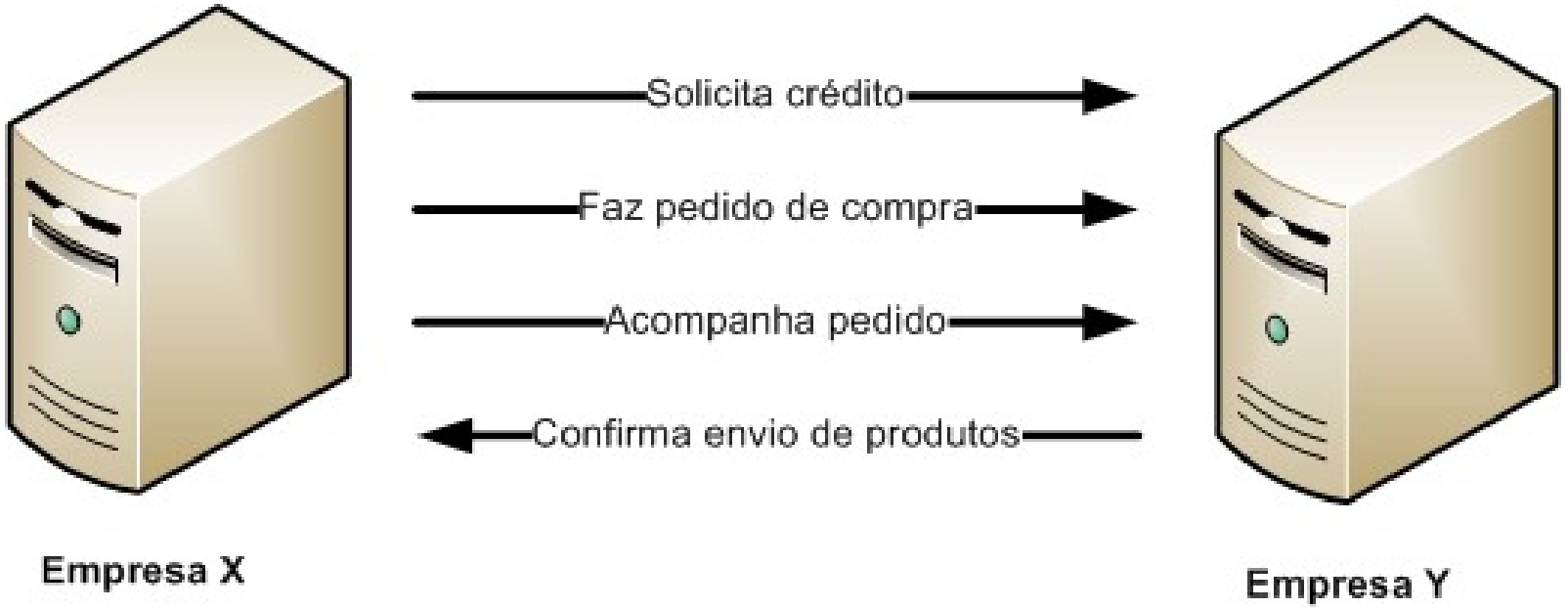
Mudança de Foco

- A Internet mudou a maneira de se fazer negócios
- Facilidade para acesso às aplicações
- HTML e HTTP -> Padrão de troca de informção
- Navegadores Web tornaram a comunicação simples e possível

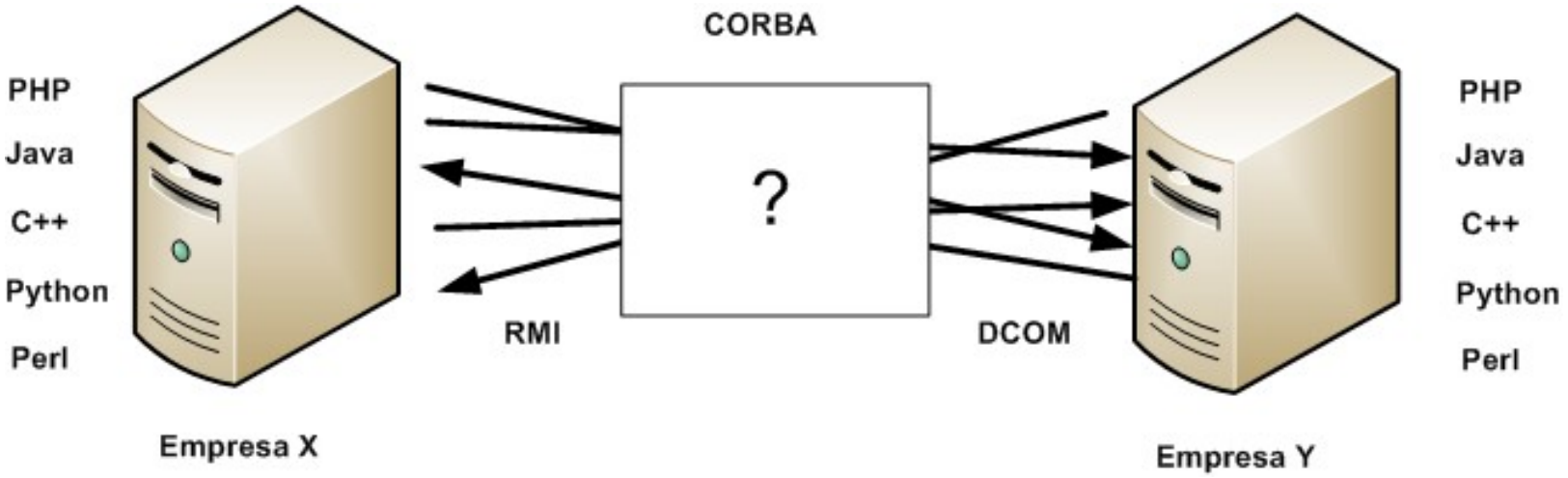
Evolução dos Negócios

- Evolução dos negócios
- Necessidade de novas soluções
- Transações e trocas de informações tornaram-se essenciais para os negócios
- Automatização nos dias atuais é mais que fundamental
- A segurança é um quesito fundamental nas transações Web

Evolução dos Negócios



Evolução dos Negócios



Problemas

- Há alguns problemas que são emergentes [Endrei et al., 2004]:
  1. Dificuldades para integrar sistemas heterogêneos;
     - Diferentes empresas utilizam diferentes tecnologias e soluções;
     - Variados domínios;
     - Muitos fabricantes;
     - Como integrar todas essas soluções de forma automatizada?
     - Uma solução para cada tecnologia é inviável!
     - Mudar de provedor de serviços pede nova implementação;
  2. Crescente demanda por manutenção (evolutiva, adaptativa, corretiva)
     - Novos requisitos => mudanças constantes;
     - Novas tecnologias => reimplementação;
     - Problemas como => desempenho, composição de serviços;

SOA - Arquitetura Orientada a Serviços

- Service-Oriented Architecture

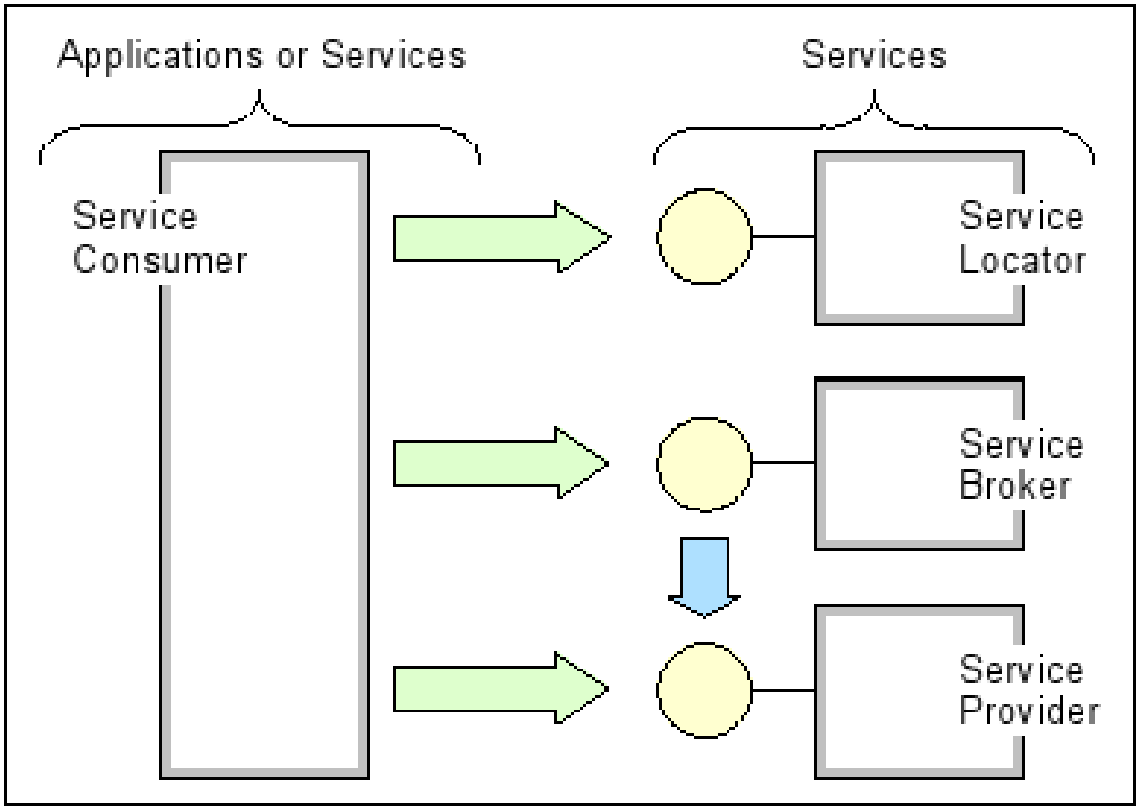
"SOA e um estilo arquitetural para construção de aplicações de software que utilizam serviços disponíveis em uma rede como a Web" [Endrei et al., 2004].

- Mas o que é um serviço?

"É a implementação de uma funcionalidade qualquer, bem definida, que pode ser utilizada por clientes em diferentes aplicações e processos de negócios" [Mahmoud, 2005].

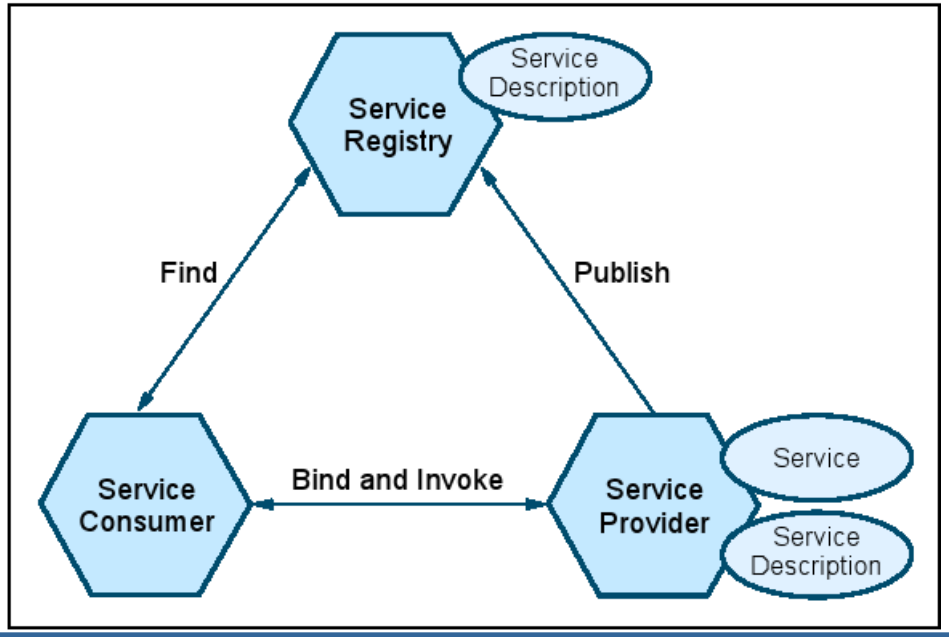
SOA - Estrutura e Terminologia

- Estrutura e terminologia básica [Endrei et al., 2004]:



SOA - Estrutura e Terminologia

- Papéis e colaborações [Endrei et al., 2004]:
  - Cliente do serviço: aplicação, componente ou outro serviço que requisita o serviço;
  - Provedor do serviço: entidade de rede endereçável que provê o serviço;
  - Registro de serviços: repositório que habilita a “descoberta” dos serviços.



Web Services

- ***Web Service – WS***
  - É um sistema de software projetado para suportar interoperabilidade entre máquinas sobre uma rede. Web Services caracterizam-se por ser uma tecnologia para computação distribuída baseada na Web.

Web Services

- Tecnologia para comunicação entre aplicações via rede
- Independente de linguagem ou plataforma
  - Troca de mensagens XML
    - Descreve uma operação a ser executada ou dados a serem trocados

Web Services - Vantagens [Endrei et al., 2004]

- Permite diferentes serviços distribuídos executarem em uma variedade de plataformas e arquiteturas
- Integração com sistemas existentes
- Integrar processos de negócios com clientes e fornecedores com um custo menor. Web services permitem compartilhamento de processos sem compartilhar tecnologias

Web Services - Vantagens [Endrei et al., 2004]

- Oferecer serviços de negócio através da Web
- Liberdade de escolha de ferramentas, produtos e tecnologias
- Usam padrões e protocolos abertos
- Não fica preso a um middleware específico
  - Não importa se está utilizando Java, Microsoft ou CORBA
- Permite o reuso de serviços e componentes

Web Services - Vantagens [Endrei et al., 2004]

- Como oferecem essas vantagens?
  - Utilização de XML
  - Utilização de protocolos padrões amplamente utilizados pela indústria
    - HTTP, HTTPS, SOAP
  - A descrição dos serviços é disponibilizada através de um arquivo WSDL (formato XML)

Web Services - Desvantagens [Endrei et al., 2004]

- Integrar Web services dinamicamente requer que o conteúdo do registro UDDI seja confiável. Somente redes privadas UDDI podem promover controle sobre seu conteúdo.
- Problemas de desempenho
  - Custo maior para processar documentos XML
- Padrões para integração de processos de negócio e gerenciamento de transações ainda estão em desenvolvimento
- A segurança em Web Services ainda carece de melhorais, pois não há padrões de fato

Segurança

- Mecanismos de segurança estão sendo propostos para Web Services
- No entanto, tais mecanismos ainda não contemplam todas as necessidades exigidas para segurança em servicos Web e alguns são propostas iniciais que ainda não se consolidaram como padrão de fato.

“A segurança é vista como um atributo de qualidade de serviço (QoS)”.

Segurança

- As propriedades de segurança computacional precisam ser aplicadas ao contexto dos Web Services:
  - Autenticidade (A) - O parceiro da comunicação deve ser verdadeiro
  - Confidencialidade (C) - Os dados transmitidos não devem ser espiados
  - Integridade (I) - Os dados transmitidos não devem ser alterados
  - Disponibilidade (D) - Os dados precisam estar disponíveis para o acesso.

Segurança

- Vulnerabilidade em sistemas computacionais:
  - Erro de programação
  - Erro de configuração
  - Erro de operação

Segurança

- A seguranca pode ser abordada em três níveis: física, lógica ou gerencial
- Ameaça
  - Uma possível ação que caso seja concretizada poderá produzir efeitos indesejados ao sistema
- Ataque
  - Concretização de uma ameaça através da exploração de alguma vulnerabilidade
- Ataques podem transpor a segurança em cada um dos níveis

Segurança

- Os ataques podem ser de:
  - Interrupção
  - Interceptação
  - Modificação
  - Personicação

Segurança

- Rede: IPSec
- Tranporte: TLS (Transport Layer Security). Segurança de sessões HTTP
- Aplicação:
  - Kerberos: Sistema de autenticação global, baseado em bilhetes. Chave privada (DES)
  - PGP (Pretty Good Privacy): Utilizado com email para (de)cifrar mensagens. Assinaturas digitais.
  - S/MIME: Cifragem de mensagens + assinaturas eletrônicas
  - SSH: Secure Shell. Substituto seguro do rsh/rlogin

Segurança em Web Services

- Não há soluções completas e concretas para a segurança de Web services e processos de negócios
- Os Web services estão suscetíveis à:
  - Negação de serviços
  - Estouro de pilha
  - Outros

Segurança em Web Services

Ataque	Impacto no Desempenho
DoS - Denial of Service	Sobrecarga cpu e memória
Coercive Parsing	Sobrecarga de cpu
SOAP Action Spoofing	Sobrecarga de cpu
XML Injection	Acesso a conteúdo proibido
WSDL Scanning	Acesso a conteúdo proibido
Metadata Spoofing	Escuta ou modificação de dados
Attack Obfuscation	Sobrecarga de cpu
Oversized Cryptography	Sobrecarga cpu e memória
BPEL State Deviation	Sobrecarga de cpu
Instantiation Flooding	Sobrecarga de cpu
Indirect Flooding	Sobrecarga cpu e memória
Workflow Engine Hijacking DoS	Sobrecarga cpu e memória

Especificação de Segurança em Web Services

- Atualmente estão disponíveis algumas especificações de segurança que podem ser aplicadas/utilizadas no contexto dos Web services:
  - XML Signature
  - XML Encryption
  - XACML
  - SAML
  - XMKS
  - WS-Security

Especificação de Segurança em Web Services

- XML Signature
  - Gera e valida assinaturas digitais expressas em XML
  - XML Canonical: Define meios para representar documentos XML na forma canônica (documentos que sejam sintaticamente diferentes, porém logicamente equivalentes, serão representados por uma mesma forma canônica)
- Não define novos algoritmos, mas faz uso dos algoritmos existentes
- Formas de assinatura:
  - Enveloped
  - Enveloping
  - Detached Assignature

Especificação de Segurança em Web Services

- XML Encryption
  - Visa prover segurança fim-a-fim no nível de mensagem
  - Apresenta de forma estruturada dados cifrados e permite cifrar documentos XML ou não
  - Oferece confidencialidade persistente, garantindo assim a confidencialidade dos dados mesmo após o término da sessão

Especificação de Segurança em Web Services

- XACML
  - Descreve uma linguagem para políticas de controle de acesso e também um formato de mensagens de solicitação e resposta

Especificação de Segurança em Web Services

- SAML
  - Conjunto de especificações e esquema XML que juntos definem uma forma padrão para criar, trocar e interpretar asserções de segurança entre entidades de uma mesma aplicação distribuída
  - Primeira versão: Permitia a transferência de autenticação e autorização entre aplicações Web
  - Segunda versão: Melhora a interoperabilidade e garante uma melhor interação com o XMLDSign
  - Um emissor SAML pode conceder asserções de segurança. Asserções são um conjunto de afirmações concedidas por um emissor. Três tipos de asserções possíveis:
    - Autenticação
    - Atributo de um sujeito
    - Autorização

Especificação de Segurança em Web Services

- XMKS
  - Especificação que define interfaces baseadas em Web services, retirando dos desenvolvedores a complexidade em se trabalhar com infra-estrutura de chave-pública (X509, PGP, SPKI)

Especificação de Segurança em Web Services

- WS-Security
  - Propõe extensões ao SOAP para permitir a construção de Web services seguros. A especificação tem como objetivo garantir a segurança fim-a-fim no nível de mensagem e não somente no nível de transporte.
  - Principais pontos da WS-Security:
    - Credenciais de segurança
    - Integridade da mensagem
    - Confidencialidade da mensagem
  - A WS-Security padroniza as informações relacionadas a assinatura e à cifragem dos dados da mensagem SOAP e suporta credenciais de segurança (UsernameToken e BinarySecureToken)

WS-Policy

- A WS-Policy define uma gramática para especificar políticas, mas não especifica como associar tais políticas aos Web services, ou mesmo como divulgá-los. Separa a definição das políticas com a associação aos recursos.

WS-Policy

- **Política**: Expressa um conjunto de alternativas de políticas válidas
- **Asserção de Política**: Expressa a habilidade do recurso, específica a um domínio
- **Alternativas de Políticas**: Descreve as combinações aceitáveis de obrigações e requisitos
  - **Operadores de Políticas**
    - **ExactlyOne:** Somente uma das asserções contidas na política poderá fazer parte de uma alternativa de política
    - **All**: Permite a combinação de todas as asserções apresentadas como uma alternativa de política

Axis2 – Visão Geral

- Axis2
  - Uma API (dentre muitas) utilizada para o desenvolvimento de Web services. E desenvolvida e mantida pela Apache Software Foundation.

Axis2 – Visão Geral

- O Axis2 é a evolução natural da mais conhecida API - Application Programming Interface para Web Services nos dias atuais.
- Diferentemente do Axis 1.0, o Axis2 apresenta diversas melhorias, principalmente em relação ao desempenho e também em relação à modularidade. A arquitetura do Axis2 é separada em componentes (módulos), e subdivide-se em componentes do núcleo e componentes não pertencentes ao núcleo.

Axis2 – Visão Geral

- Dentre as principais funcionalidades do Axis2 destacam-se:
  - ***Mecanismo de implantação (deployment) com base na plataforma Java 2 Enterprise Edition (J2EE) - (baseado em arquivo)***:
    - Recursos, arquivos de configuração e binários todos em um único arquivo que descreve o serviço.
  - ***Hot deployment e hot update***: Possibilidade de implantar atualizar serviços sem reiniciar o servidor de aplicação. No caso da atualização é recomendável somente o uso em ambientes de teste.
  - ***Presença de um repositório (onde se localizam os serviços e módulos)***: Onde ficam armazenados os serviços que devem ser acessados pelos clientes.
  - ***Mudanças na implantação de manipuladores (módulos)***: Como a arquitetura do Axis2 é modular, tudo que não é necessário estar no núcleo é disponibilizado como módulo. Esses módulos implementam as especificações de Web Services (WS-Policy, WS-Reliable, WS-Security).

Axis2 – Visão Geral

- ***Novos descritores de implantação***:
  - Descritor Global (axis2.xml)
  - Descritor do Serviço (services.xml)
  - Descritor do Módulo (module.xml)

Axis2 – Visão Geral

- O arquivo ***axis2.xml*** deve apresentar:
  - Parâmetros do serviço
  - Tipo de Transporte do Rementente
  - Tipo de Transporte do Emissor
  - Fases
  - Módulo Global

Axis2 – Visão Geral

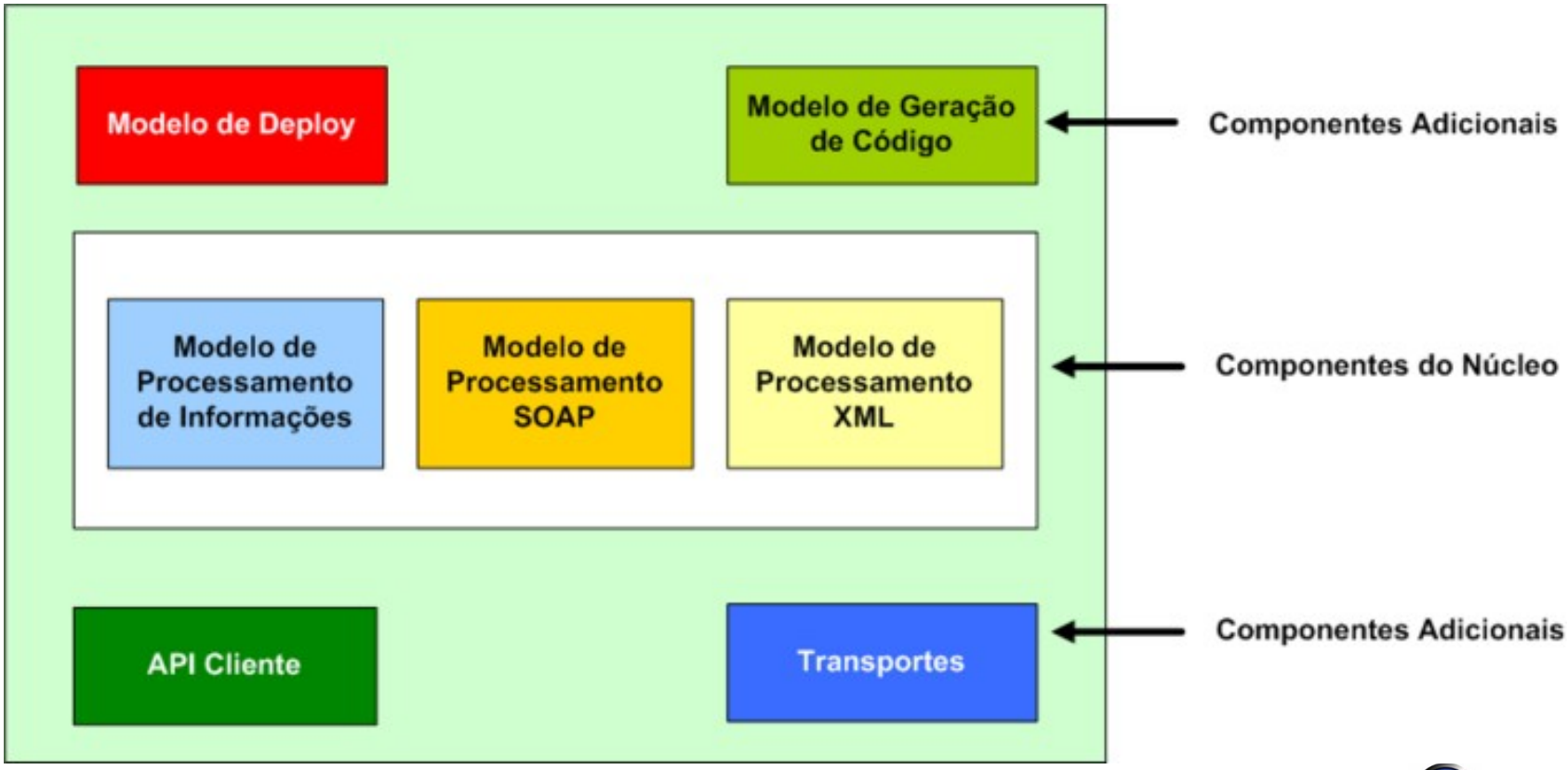
- O arquivo ***services.xml*** deve ser caracterizado com as opções abaixo:
  - Parâmetros de Nível de Serviço
  - Descrição do Serviço
  - Receptores de Mensagem
  - Operação necessária para expor o serviço como uma operação web
  - Módulos de Nível de Serviços

Axis2 – Visão Geral

- O arquivo ***module.xml*** deve apresentar os seguintes parâmetros:
  - Manipuladores e suas regras de fases
  - Parâmetros do módulo
  - Descrição sobre o módulo
  - Pontos finais (Endpoints) - No caso de utilização de mensagens confiáveis, este parâmetro é fundamental

Axis2 – Componentes

- O Axis2 é um motor de processamento de mensagens SOAP bastante modular. Sua divisão é baseada em componentes, como pode ser descrito na figura a seguir:



Axis2 - Componentes

- **Componentes do núcleo:**
  - AXIOM - Axis Object Model: Modelo de Objetos em XML
  - Módulo de Processamento SOAP: Framework Manipulador
  - Modelo de Processamento de Informações: Contextos e Descrições

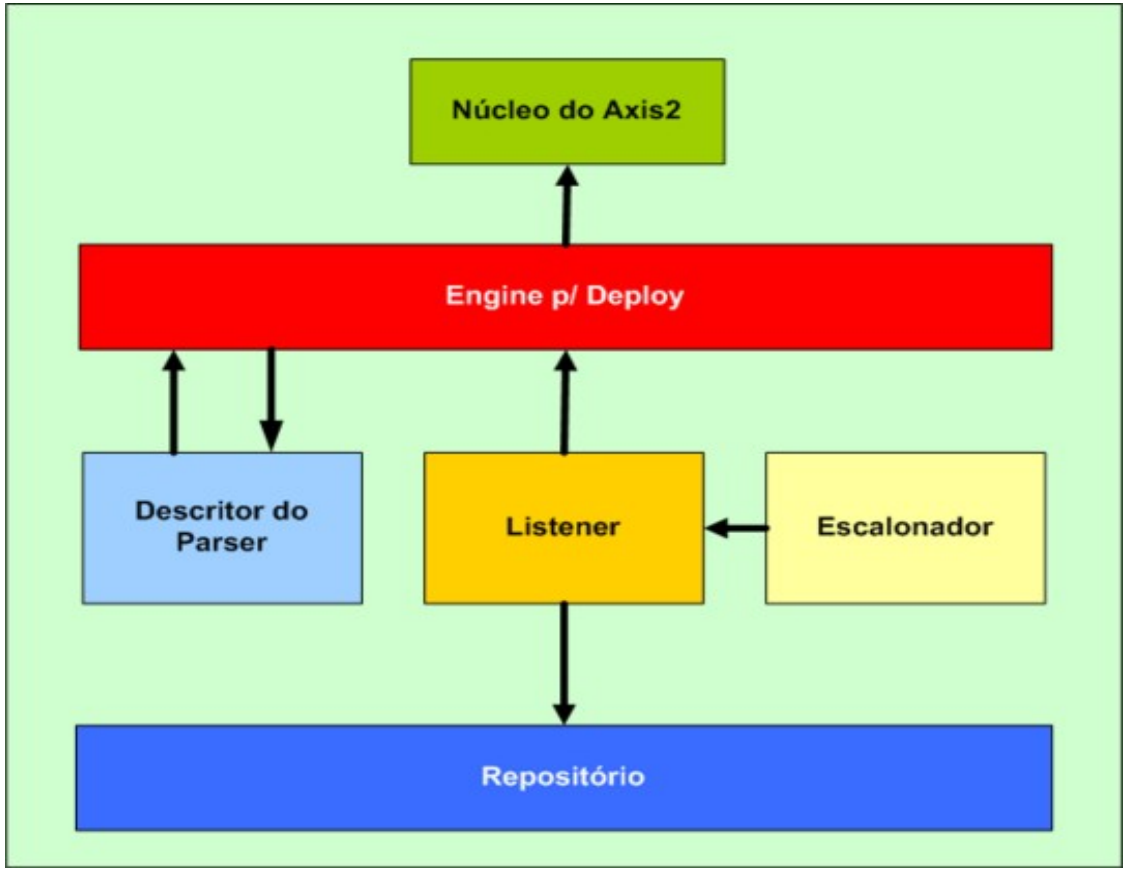
- **Outros componentes:**
  - Modelo de Deploy
  - Transportes
  - API Cliente
  - Modelo de Geração de Código

Axis2 – Modelo de Processamento

- Modelo mais adequado que a versão 1.0 do Axis
- Inicialmente baseada em DOM – Document Object Model
- Atualmente baseado em modelo de objetos próprio denominmado AXIOM (Axis2 Object Model)
- **DOM x AXIOM**
  - **DOM**: Problema de manter a hierarquia completa dos objetos em memória
  - **AXIOM**: Representa qualquer mensagem do framework (entrada e saída)
- Técnicas **Pull x Push**
  - **Pull**: O invocador tem o controle completo sobre o parser e pode perguntar sobre o próximo evento. O objeto é construído sob demanda (SAX)
  - **Push**: O parser procede até atingir o final do documento (StAX).

Axis2 – Modelo de Implantação

- Um mecanismo de implantação de aplicações (deployment) baseada em repositórios. As principais partes desta arquitetura são evidenciadas na figura a seguir



Axis2 – Modelo de Implantação

- **Escalonador**: Componente que informa ao listener para procurar por serviços no repositório Listener: Busca por atualizações no repostório de serviços

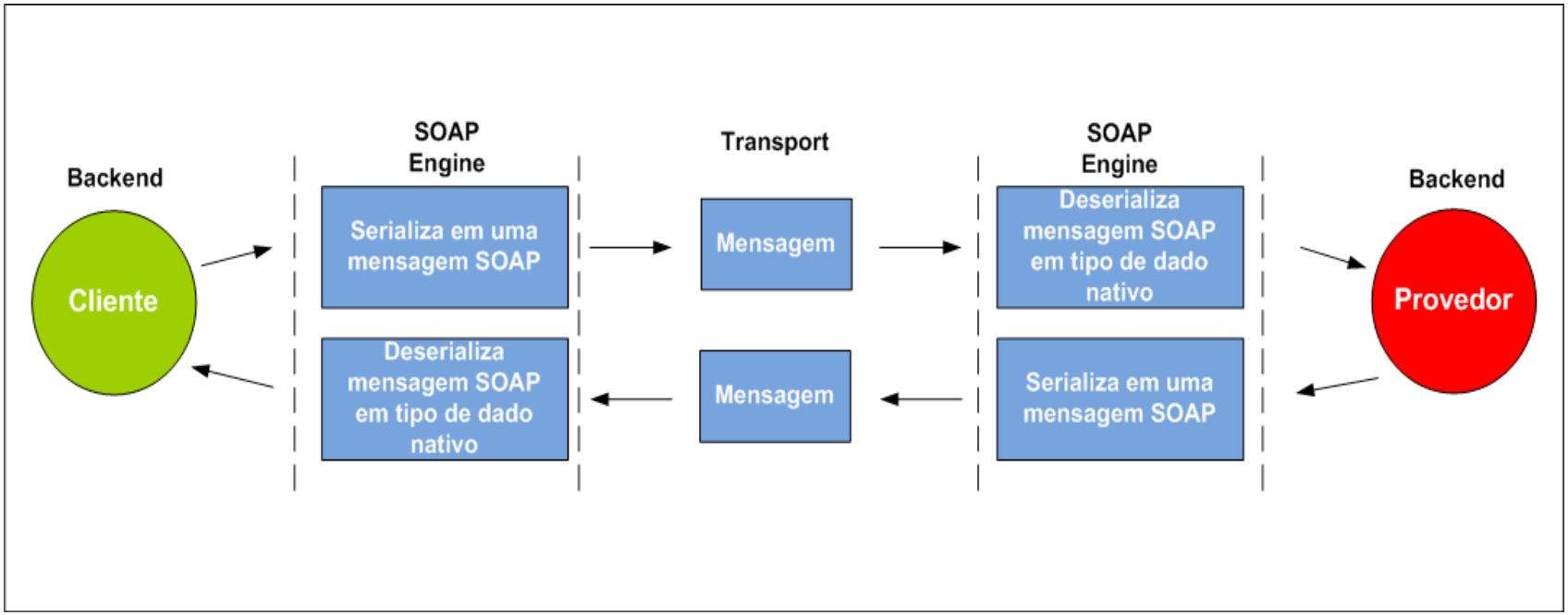
- **Descritor do Parser**: Um componente para processar os serviços e descritores de módulos para deploy: O componente central da implantação (deployment}) que faz todo o processamento lógico

- **Núcleo do Axis2**: O Axis2 é independente da implantação e vice-versa

- **Repositório**: Um diretório no sistema de arquivos que armazena os serviços implantados (deployed)

Axis2 – Ciclo de vida de um serviço

- Com a utilização do Axis2 é possível construir serviços que se comunicam através de trocas de mensagens no formato XML, utilizando como protocolo de transporte o HTTP.



Axis2 – Ciclo de vida de um serviço

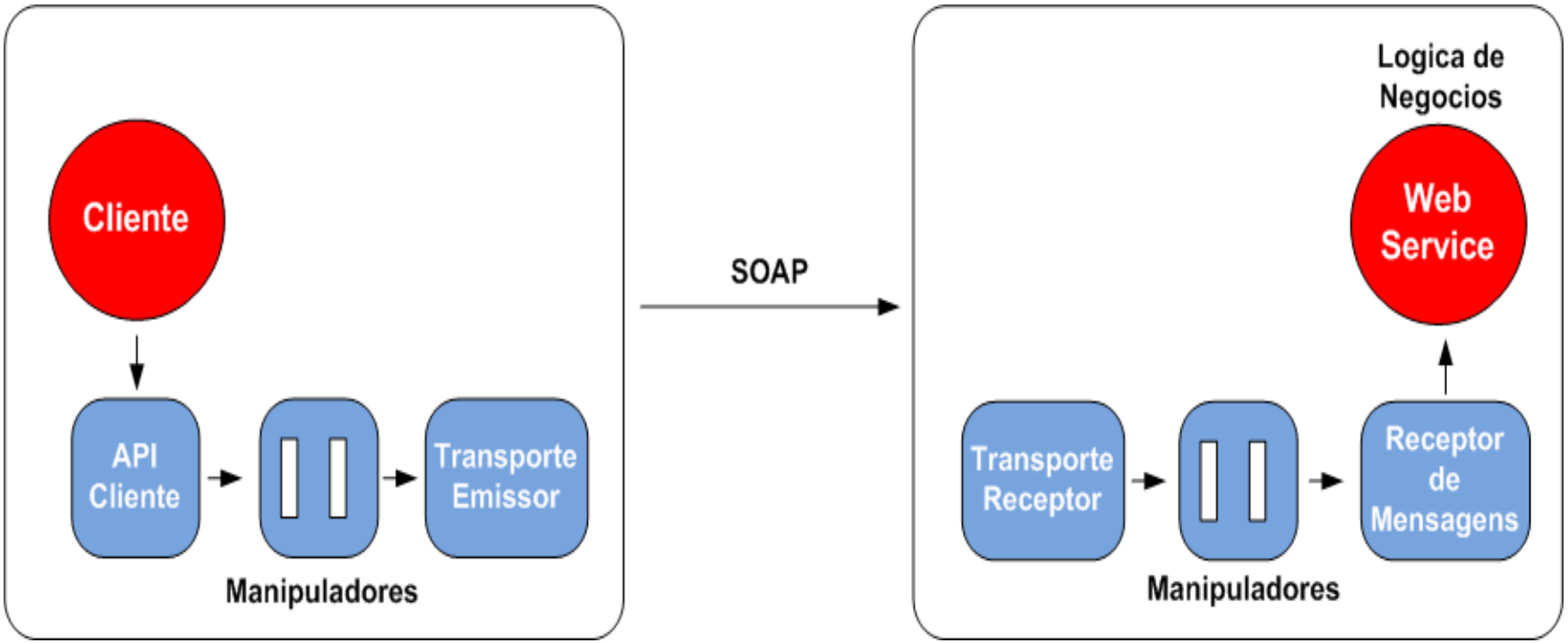
- O Axis2 executa as seguintes tarefas:
  - Cria mensagens SOAP
  - Recebe e processa mensagens SOAP
  - Cria um Web Service a partir de uma classe Java
  - Cria classes de implementação para o servidor e o cliente, usando a WSDL
  - Recupera facilmente a WSDL para um serviço
  - Envia e recebe mensagens SOAP com anexos
  - Cria ou utiliza serviços que tiram a vantagem dos padrões WS-Security, WS-ReliableMessaging, WS-Addressing, WS-Coordination e WS-AtomicTransaction

Axis2 – Ciclo de vida de um serviço

- Assumindo-se que o Axis2 execute tanto no cliente quanto no provedor de serviços, a interação entre cliente e provedor deve seguir algumas fases:
  - O cliente (emissor) envia uma mensagem SOAP
  - O manipulador do Axis2 realiza ações necessárias de acordo com o que foi definido pelo usuário
  - O módulo responsável pelo transporte envia a mensagem
  - Do lado do receptor (provedor) a detecção da mensagem é feita pelo módulo responsável por este finalidade
  - O responsável pelo trasporte (transport listener) passa a mensagem para um dos manipuladores do receptor (provedor)
  - Uma vez processada, a mensagem é entregue à aplicação

Axis2 – Ciclo de vida de um serviço

- Manipulação de mensagens



Axis2 – Ciclo de vida de um serviço

- O Axis2 permite quebrar as ações na transmissão de mensagens em várias fases. Essas fases são definidas como:
  - **Fases Pré-Definidas**
    - Pré-Despacho
    - Despacho
    - Processamento de Mensagens
  - **Fases definidas pelo usuário**: As fases pré-definidas são invocadas independente do serviço especificado. As fases definidas pelo usuário são invocadas quando o despachante encontrar uma operação. Cada fase possui uma coleção de manipuladores. O Axis2 permite controlar quais manipuladores estarão em quais fases e a ordem em que tais manipuladores são executados dentro de cada fase. O mais interessante é que pode ser adicionada uma nova fase juntamente com seus respectivos manipuladores. Entenda-se por manipuladores os módulos componentes do Axis2. Exemplos:
    - **WS-Security**
    - **WS-Reliability**

Desenvolvimento de Aplicação Segura com Axis2

- Este tópico deve ser discutido no dia do evento
  - A construção da aplicação deve ser feita passo-a-passo

Referências

- Introduction to Web services and the WSDK v5.1.

Disponível em www.ibm.com/developerWoks - último acesso em 30/09/2009.

- Web services and CORBA.

Disponível em www.xs4all.nl/~irmen/comp/CORBA_vs_SOAP.htmI - último acesso em 30/09/2009.

- Web services and SOAP.

Disponível em www.w3c.org/2002/ws - último acesso em 30/09/2009.

- Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2003).

Web services: Concepts, architectures, and applications.

V. Springer Verlag.

Referências

- Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., Luo, M., and Newling, T. (2004).

Patterns: Service-Oriented Architecture and Web Services.

IBM Redbooks Series. IBM Corporation.

Disponível em www.redbooks.ibm.com/abstracts/sg246303.html - último acesso em 30/09/2009.

- Mahmoud, Q. H. (2005).

Service-oriented architecture (SOA) and Web services: The road to enterprise application integration (EAI).

Sun Technical Articles.

Disponível em http://java.sun.com/developer/technicalArticles/WebServices/soa/index.html - último acesso em 30/09/2009.