Avec le soutien de Wallonie-Bruxelles International

# SÉCURITÉ ET WEB SERVICES

Prof. Jean-Noël Colin
jean-noel.colin@fundp.ac.be

# Agenda



- Introduction
- Technologies des web services
- Sécurité et XML
- WS-Security
- Au-delà du modèle RPC

# Introduction



- Web Service: application offrant une API accessible via le Web
  - Modèle client/serveur
  - Echange requête/réponse
  - Accessible via un 'endpoint', identifié par une URI

# Introduction



- Objectif: Intégration!
  - Entre applications hétérogènes
    - Langage
    - Environnement d'exploitation (architecture, OS…)
    - Réseau
    - Organisation
  - Motivations
    - Intégration d'applications
    - Services B2C ou B2B
    - Automatisation de business processes
    - Intégration d'information
  - SOA?

# Introduction

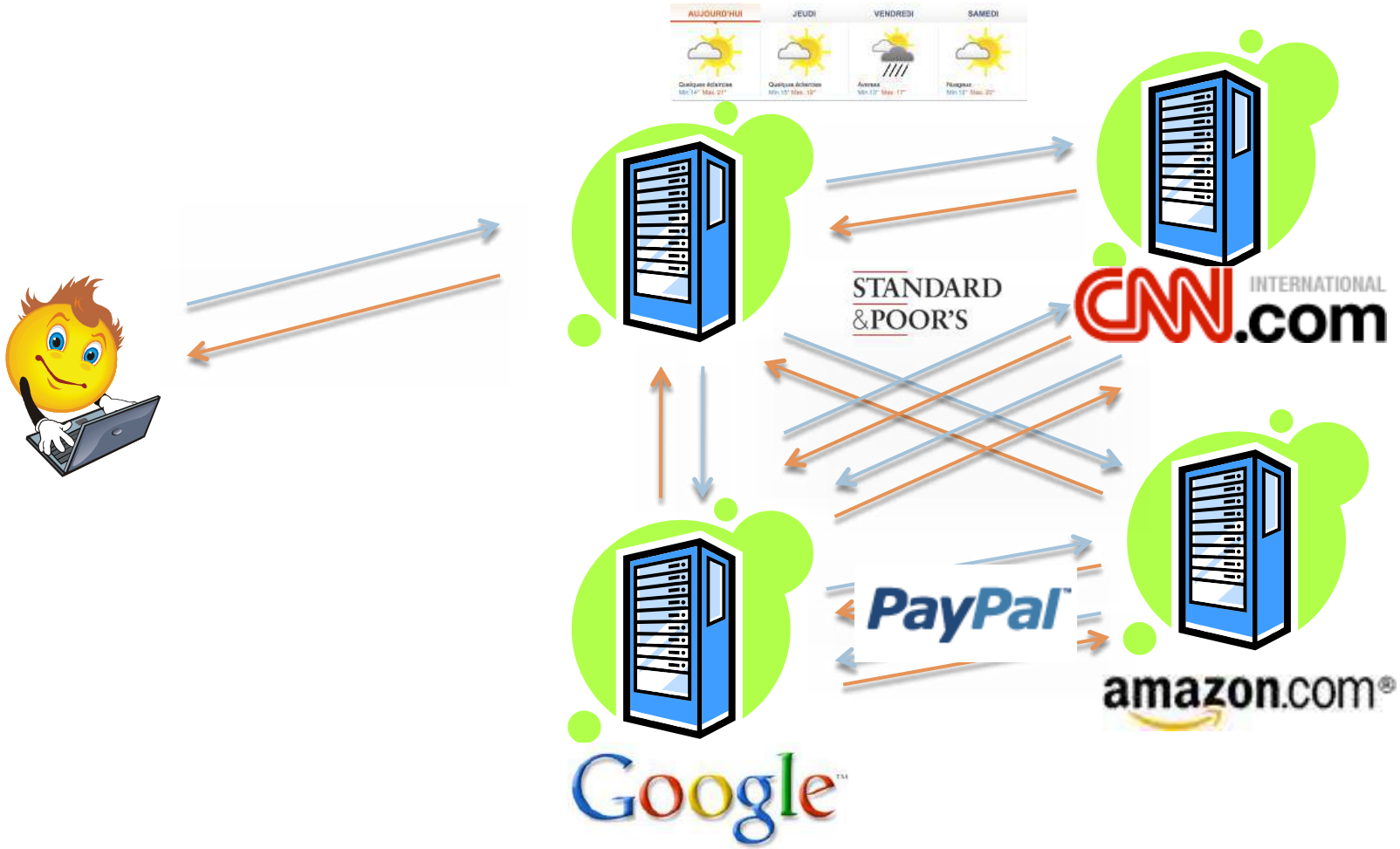


- Un peu d'histoire…
  - DCE – Distributed Computing Environment
  - CORBA – Common Object Request Broker Architecture
  - Microsoft's DCOM -- Distributed Component Object Model
  - Pour arriver à une standardisation (toujours en cours) des protocoles, outils, langages et interfaces

⇒ Web Services

# Introduction





# Introduction



- Défis de sécurité
  - Identité
    - Des utilisateurs, des applications, anonyme…
  - Contrôle d'accès, traçabilité, non-répudiation…
  - Sécurité des messages
    - Confidentialité, intégrité, non-répudiation
- Impact business

FUNDP
NAMUR

8

# Technologies des Web Services

# Technologies des Web Services



- Ensemble de standards et de technologies
  - XML: expression des messages
  - SOAP: échange des messages
  - WSDL: description des services
  - UDDI: répertoire de services

# Technologies des Web Services

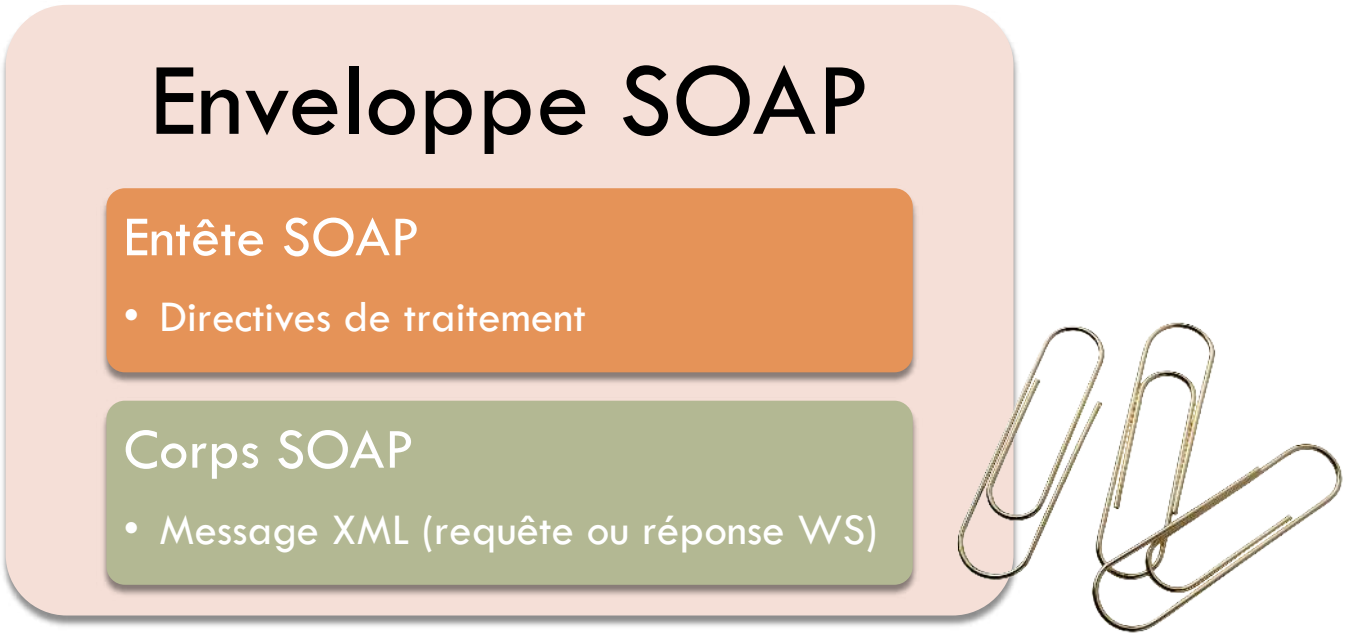


- XML – eXtensible Markup Language
  - Format de données permettant de structurer et échanger l'information
  - Indépendant de la plateforme d'exécution
  - Extensible
  - Schéma XML: définit la structure d'un document
    - Type de données (éléments, attributs, types…)
    - Composition d'éléments
    - Permet la création et validation
  - Xpath: mécanisme d'accès à un composant du document XML
  - XSLT: transformation d'un document XML vers une autre forme
  - Xquery: recherche dans document XML (ou collection de documents)

# Technologies des Web Services



- SOAP – ~~Simple Object Access Protocol~~
  - Protocole d'envoi de messages XML
  - Textuel (vs binaire)
  - Structure d'un message SOAP



# Technologies des Web Services

FUNDP NAMUR



- Types de communication
  - RPC
    - Le document XML transmis dans la requête SOAP est calqué sur la syntaxe de la méthode invoquée
    - Tâches à fin grain
    - Traitement synchrone
  - Document
    - Le document XML transmis dans la requête SOAP est traité par le serveur, qui renvoie un document XML en retour
    - Client ne sait pas comment le service est implémenté, ni comment le message est traité
    - Tâches à gros grain, typiquement B2B
    - Traitement asynchrone

# Technologies des Web Services



- Exemple:

```
package be.ac.fundp.info.wstest;

public class MathTeacher {
public int add(int a, int b) {
return a + b;
}

public int sub(int a, int b) {
return a - b;
}

public long mult(int a, int b) {
return a * b;
}
}
```

# Technologies des Web Services



- Requête SOAP

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <mult xmlns="http://wstest.info.fundp.ac.be">
            <a>6</a>
            <b>9</b>
        </mult>
    </soapenv:Body>
</soapenv:Envelope>
```

# Technologies des Web Services



- Réponse SOAP

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <multResponse xmlns="http://wstest.info.fundp.ac.be">
            <multReturn>54</multReturn>
        </multResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

# Technologies des Web Services



- WSDL – Web Services Definition Language
  - Fichier XML décrivant l'interface d'un Web Service
    - Services, types de données, messages, encodage…
    - Répond aux questions: Quoi? Comment? Où?
  - Au niveau serveur
    - Permet la génération d'un squelette ou est généré à partir du service
  - Au niveau client
    - Sert de 'manuel' pour le service
    - Permet la génération d'un squelette

# Technologies des Web Services



## WSDL – définition des types

```
<wsdl:types>
   <schema elementFormDefault="qualified" targetNamespace="http://wstest.info.fundp.ac.be"
     xmlns=http://www.w3.org/2001/XMLSchema>
     <element name="mult">
       <complexType>
         <sequence>
           <element name="a" type="xsd:int"/>
           <element name="b" type="xsd:int"/>
         </sequence>
       </complexType>
     </element>
     <element name="multResponse">
       <complexType>
         <sequence>
           <element name="multReturn" type="xsd:long"/>
         </sequence>
       </complexType>
     </element>
   </schema>
 </wsdl:types>
```

# Technologies des Web Services



## WSDL – définition des messages et ports

```
<wsdl:message name="multRequest">
    <wsdl:part element="impl:mult" name="parameters"/>
</wsdl:message>
<wsdl:message name="multResponse">
    <wsdl:part element="impl:multResponse" name="parameters"/>
</wsdl:message>
```

```
 <wsdl:portType name="MathTeacher">
...
    <wsdl:operation name="mult">
      <wsdl:input message="impl:multRequest" name="multRequest"/>
      <wsdl:output message="impl:multResponse" name="multResponse"/>
    </wsdl:operation>
...
  </wsdl:portType>
```

# Technologies des Web Services



## WSDL – définition des bindings

```
<wsdl:binding name="MathTeacherSoapBinding" type="impl:MathTeacher">
    <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
…
    <wsdl:operation name="mult">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="multRequest">
        <wsdlsoap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="multResponse">
        <wsdlsoap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
…
  </wsdl:binding>
```

# Technologies des Web Services



- WSDL – définition du service

```
<wsdl:service name="MathTeacherService">
    <wsdl:port binding="impl:MathTeacherSoapBinding" name="MathTeacher">
      <wsdlsoap:address location="http://localhost:8080/WSTest/services/MathTeacher"/>
    </wsdl:port>
  </wsdl:service>
```

# Technologies des Web Services



- UDDI – Universal Description, Discovery and Integration
  - Annuaire de web services
    - Publier, mettre à jour, consulter
    - Infos. de contact, catégories, bindings
  - Deux modes d'utilisation
    - Interne à l'organisation: annuaire central des services
    - Public: difficile à mettre en place

# Technologies des Web Services

FUNDP NAMUR



- Web Services RESTful
  - REST = REpresentational State Transfer
    - Ensemble de principes d'architecture
  - Principes
    - Chaque ressource est identifiable
      - Sur HTTP, via une URL
    - Interface uniforme
      - Réutilisation des méthodes HTTP: GET (read), PUT (insert), POST (update), DELETE (delete)
    - Protocole stateless

FUNDP
NAMUR

23

# Sécuriser les Web Services

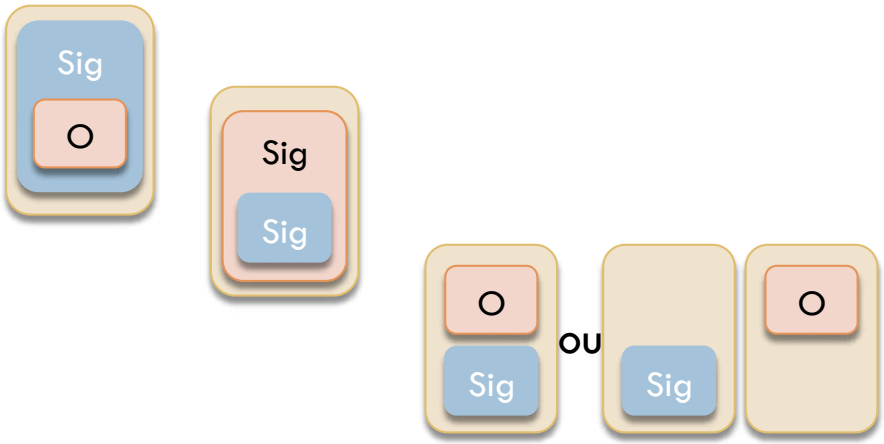
## Sécurité et XML

## WS-Security

# Sécurité et XML



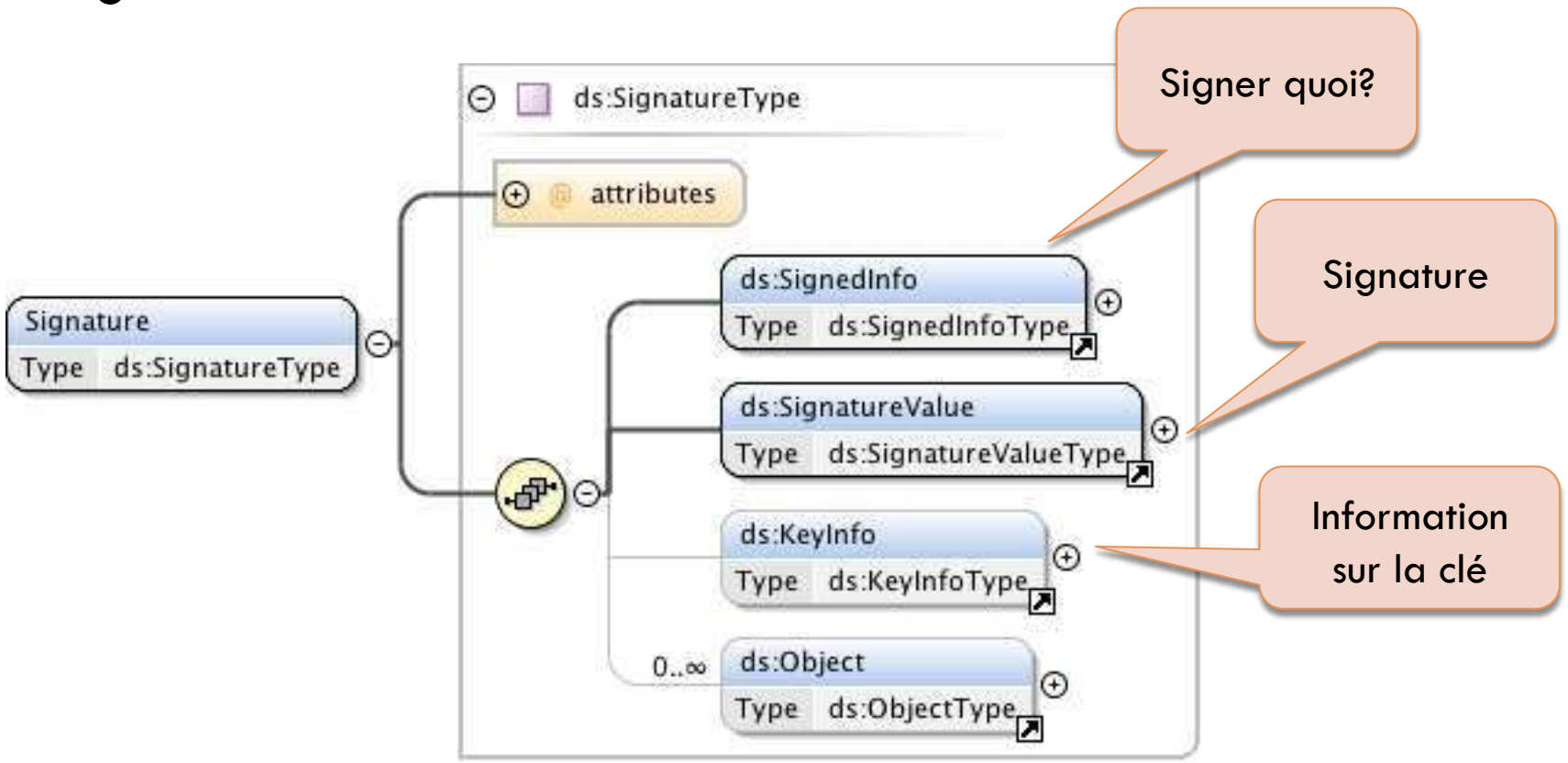
- Signature XML
  - Objectif: signature numérique d'un document XML
    - Garantir l'authenticité et l'intégrité du document
  - Recommandation W3C: XML Signature Syntax and Processing
  - http://www.w3.org/TR/xmldsig-core/
  - Types de signature
    - Enveloppante ('enveloping')
    - Enveloppée ('enveloped')
    - Détachée ('detached')



# Sécurité et XML



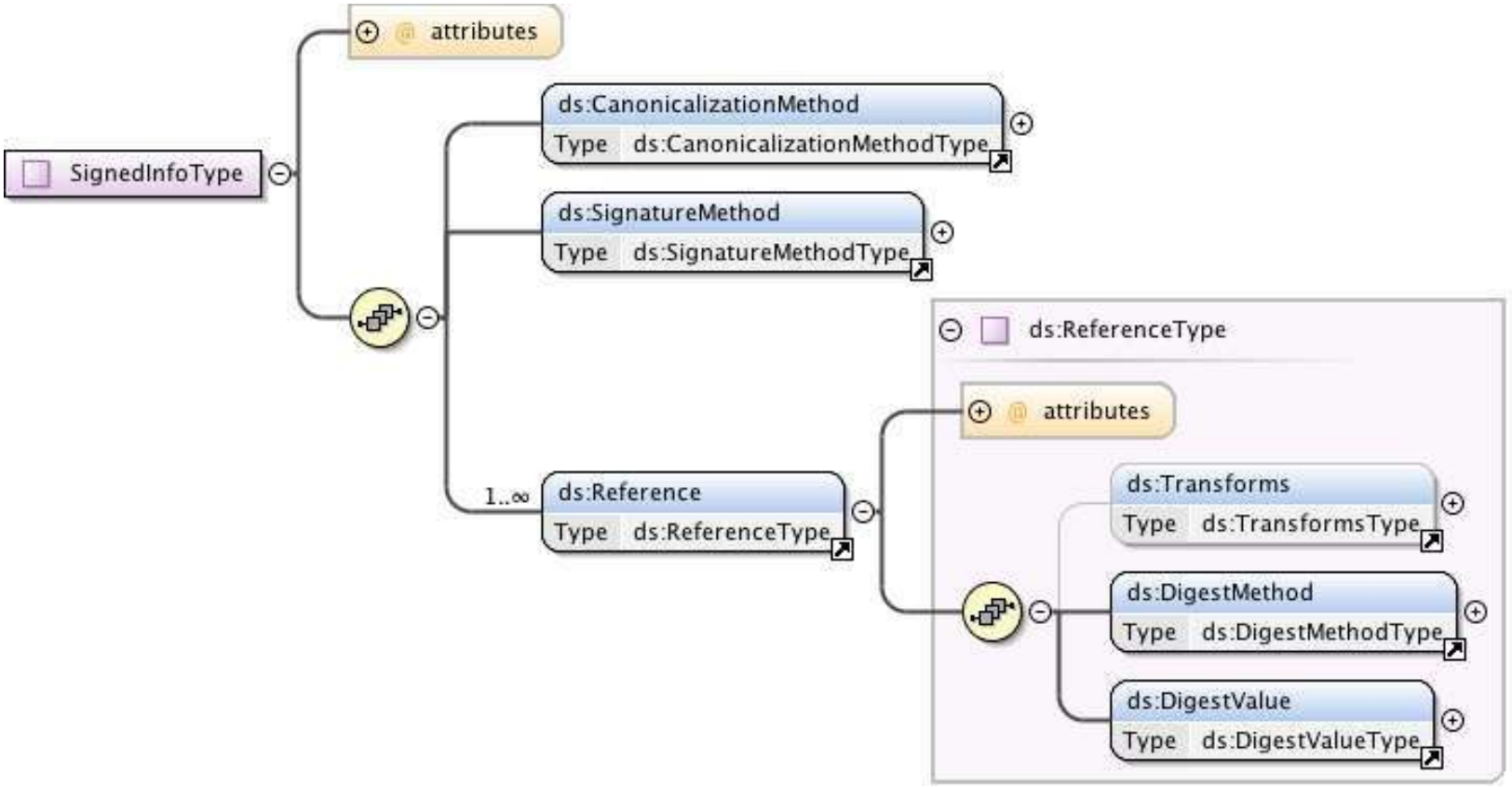
- Signature XML – Schéma



# Sécurité et XML



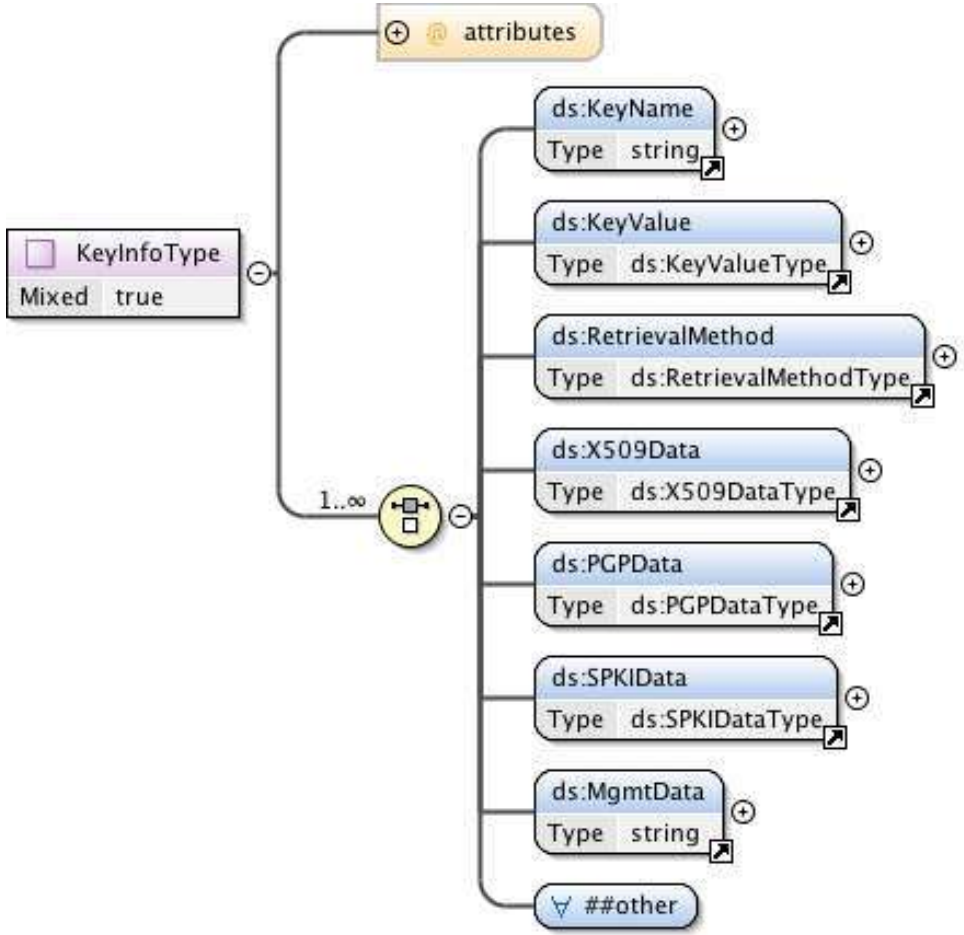
- Signature XML – Objet de la signature



# Sécurité et XML



- Signature XML – Information sur la clé



# Sécurité et XML



- Signature XML – Mécanisme
  - Génération des références
    - Transformation éventuelle
    - Calcul de l'empreinte
  - Génération de la signature
    - Construire l'élément SignedInfo
    - Appliquer la CanonicalizationMethod
    - Calculer l'empreinte à partir du résultat de l'étape précédente
    - Calculer la signature suivant SignatureMethod sur l'empreinte
  - ☝ Ce qui est signé, c'est le contenu de SignedInfo, pas les ressources pointées par les References.

# Sécurité et XML

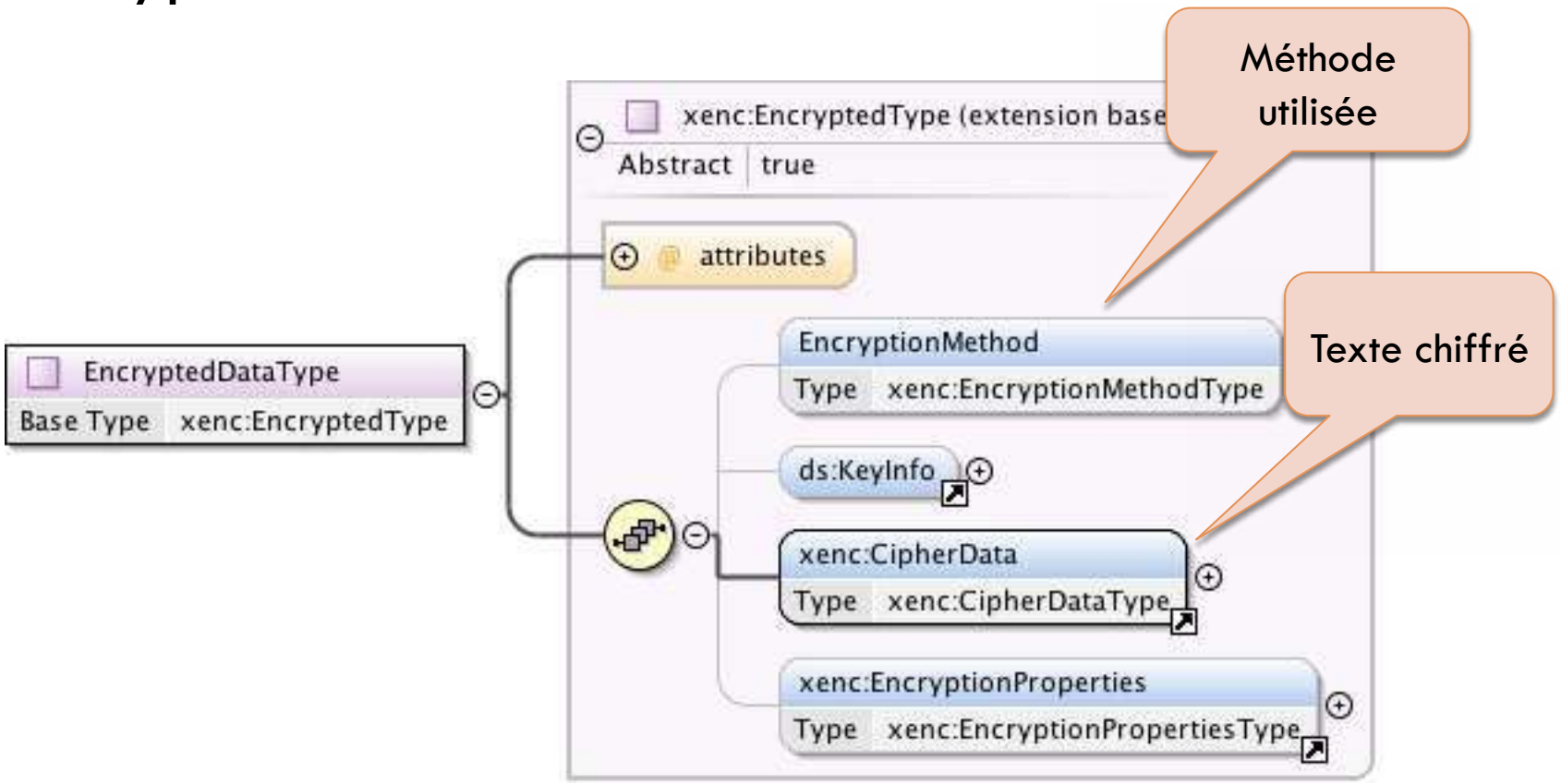


- Encryption XML
  - Objectif: chiffrement d'un document XML
    - Garantir la confidentialité de bout en bout du document
  - Recommandation W3C: XML Encryption Syntax and Processing
  - http://www.w3.org/TR/xmlenc-core/
  - Flexible
    - Possibilité d'encrypter tout ou partie du document, avec 1 ou différentes clés
  - ☝ Chiffrement symétrique!

# Sécurité et XML



- Encryption XML – Schéma



# Sécurité et XML



- Encryption XML
  - Encryption symétrique
  - Quid de la clé? Différentes possibilités:
    - Clé connue des deux parties
    - Plusieurs clés communes et identifiant de la clé utilisée transmis
    - Transmission de la clé partagée encryptée avec la clé publique du correspondant

# Sécurité et XML



- Encryption XML – Mécanisme
  - Choix d'un algorithme (3DES ou AES)
  - Obtention ou génération de la clé
  - Sérialisation des données à encrypter
  - Encryption
- Décryption
  - Identifier l'algorithme et la clé utilisés
  - Obtenir la clé
  - Déchiffrer les données
  - Intégrer les données déchiffrées dans le document

# Sécurité et XML



- Outils
  - Package javax.xml.crypto.dsig
  - XWSS – XML and Web Services Security
    - https://xwss.dev.java.net/
  - Apache XML Security
    - http://santuario.apache.org/download.html
  - …

# WS-Security

FUNDP NAMUR



- Standard OASIS
- V1.0 – 2004, V1.1 – 2006
- Objectifs
  - Authentification
  - Confidentialité des messages
  - Intégrité des messages

# WS-Security

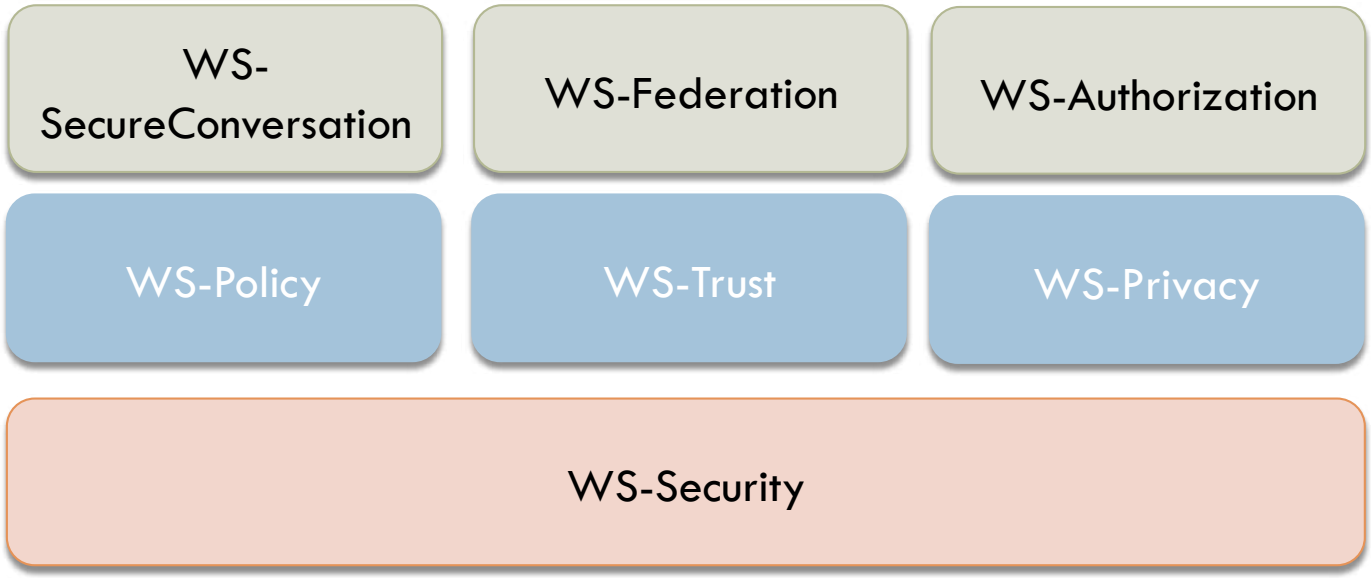


- Intégration de différentes technologies
  - Certificats, SAML, Sécurité XML…
- pour protéger les messages WS de bout en bout

# WS-Security



- Fondation d'autres standards



# WS-Security



- Notion de jeton de sécurité (security token)
  - Servant pour l'authentification ou l'autorisation
    - Ex: username/password, certificat X509, assertion SAML
- Extension de SOAP
  - Définition d'un header SOAP contenant l'information de sécurité
    - Jetons de sécurité
    - Signatures numériques
    - Élements encryptés

# WS-Security

FUNDP NAMUR



- Security Tokens
  - UsernameToken
    - Username/password
    - Username/digest
  - Jetons binaires
    - Certificat X.509
      - + signature
    - Ticket Kerberos
      - TGT ou ST

# WS-Security



- Security Tokens
  - Jetons XML
    - SAML
      - Jeton = Assertion SAML
      - Problème: comment garantir au fournisseur de service que l'émetteur de la requête est bien le sujet de l'assertion?
        - SubjectConfirmation/ConfirmationMethod
          - holder-of-key
          - sender-vouches
    - Autres possibilités: XrML, XCBF (XML Common Biometric Format)

# WS-Security



- Confidentialité des messages SOAP
  - Utilisation de XML-Encryption
    - Encryption d'un ou plusieurs éléments du message SOAP
    - Référence vers les éléments encryptés dans le header
  - Clé partagée
  - Key wrapping
    - Élément 'EncryptedKey' dans le header WS-Security
  - Possibilité d'encrypter différents éléments avec des clés différentes

# WS-Security



- Considérations supplémentaires
  - Taille des messages: WS-Security augmente parfois de manière importante la taille des messages
    - Impact sur les communications, mémoire et CPU
  - Traitement des messages
    - pas de possibilité de 'streaming'
    - Nécessité d'acquérir l'entièreté du message avant de débuter le traitement
  - Performances: fonctions cryptographiques gourmandes en ressources
    - Solution: utilisation de hardware spécialisé

# WS-Security



- Outils
  - WSS4J – Web Services Security For Java
    - http://ws.apache.org/wss4j/
  - Apache Rampart
    - http://ws.apache.org/rampart/
  - .Net Web Services Enhancements
  - XWSS – XML and Web Services Security
    - https://xwss.dev.java.net/

# WS-Policy



- Recommandation W3C (v1.5 – Nov. 2007)
- Objectif: spécifier des informations et des exigences pour un WS
  - S'applique aussi bien au serveur qu'au client
  - Exemples:
    - utilisation d'une version spécifique de SOAP
    - Exigence de signature
    - Information sur le format de la réponse (encrypté, signée…)

# WS-Policy



- WS-Policy: formalisme pour la définition de politiques s'appliquant à un WS
  - Assertion = caractéristique ou exigence d'un sujet
    - Sujet : endpoint, message, operation…
    - Contient des 'policy expression'
  - Policy alternative: collection d'assertions
  - Policy: ensemble de *policy alternatives*
  - Opérateurs: ExactlyOne ou All
- WS-PolicyAttachment: lie les politiques et les ressources auxquelles elles s'appliquent
  - 2 stratégies
    - Policy incluse dans le WSDL
    - Document indépendant liant le WS et la policy applicable
- WS-PolicyAssertions: ensemble de politiques pré-définies

# WS-Policy

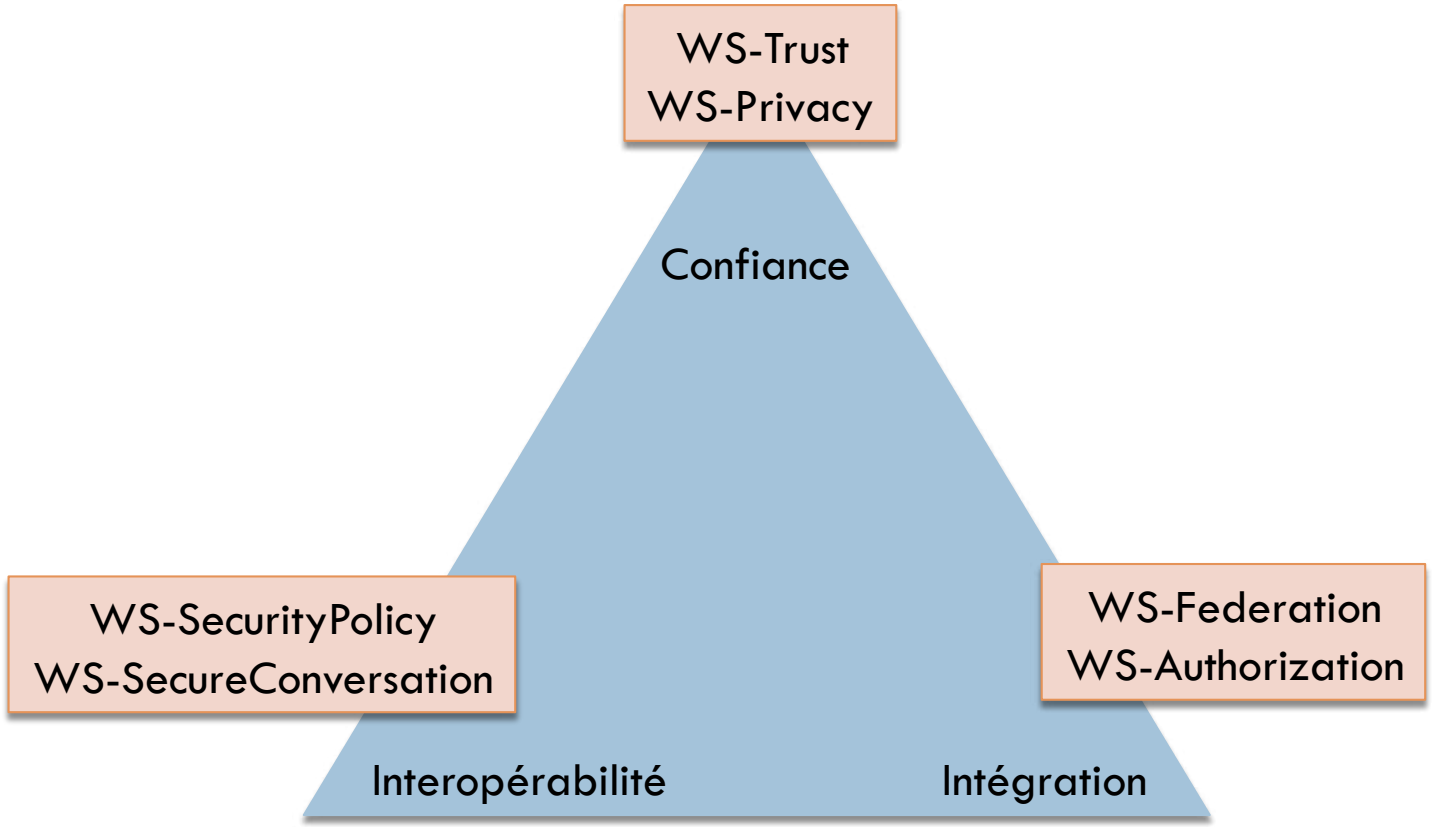


- WS-SecurityPolicy
  - Standard OASIS (v1.2 – 2007) basé sur WS-Policy
  - Sujets:
  - Définit 6 types d'assertions WS-Policy
    - Assertion de protection: définit ce qui doit être protégé et comment
      - Intégrité, confidentialité, éléments (en-têtes) obligatoires,
    - Conditionnelle
    - Types de jetons
    - Mécanismes de sécurité: algorithmes, TransportBinding
- Possibilité d'utiliser XACML poru l'expression de policy

# Au-delà du modèle RPC

# Au-delà du modèle RPC





# Au-delà du modèle RPC

FUNDP NAMUR



- WS-Trust
  - Standard OASIS (1.4 – 2009)
  - Modèles de confiance nombreux et variés
    - Et transorganisationnels
  - Problèmes
    - Émettre et obtenir des jetons de sécurité
    - Etablir et valider des relations de confiance
  - Définition d'un Security Token Service
    - Émet, valide ou échange un jeton de sécurité

# Au-delà du modèle RPC

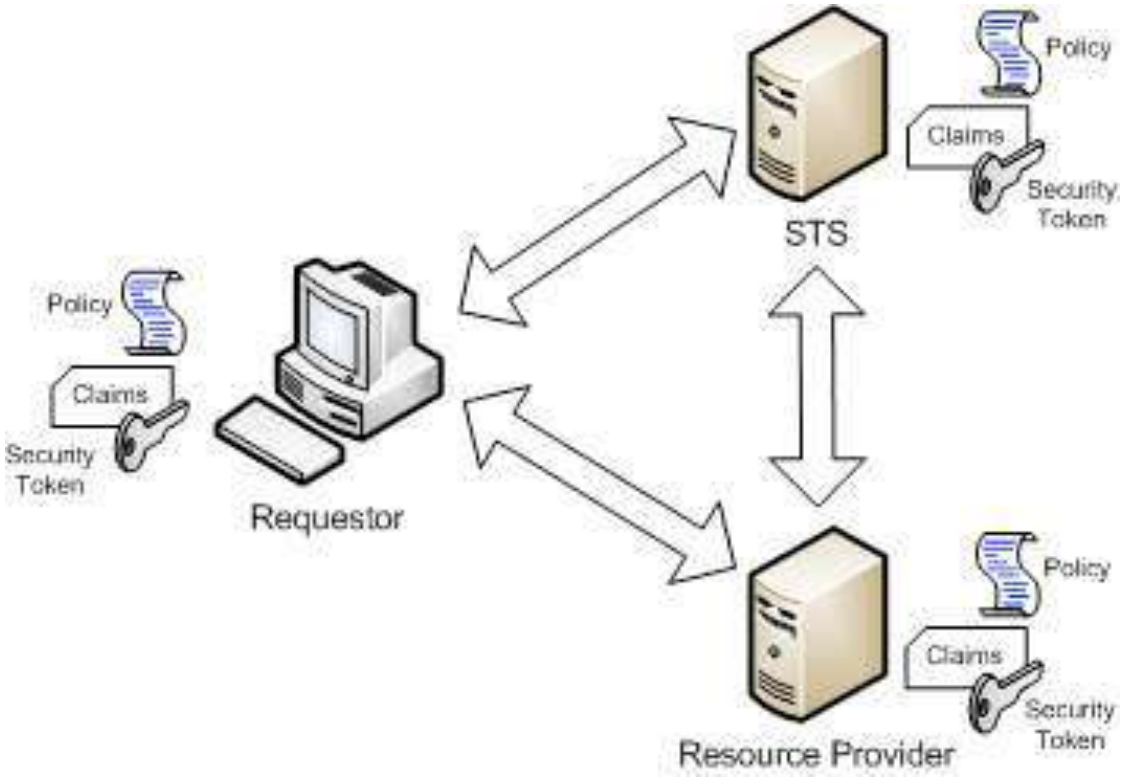


- WS-Trust
  - Modèle de fonctionnement
    - WSP définit sa politique d'utilisation (WS-Policy)
    - WSC envoie sa requête à WSP
      - Peut demander au STS un jeton de sécurité
    - WSP valide la requête
      - Vérifie qu'elle rencontre les exigences exprimées dans la politique
      - Vérifie les signatures
      - Valide les jetons de sécurité
        - Eventuellement en interrogeant le STS

# Au-delà du modèle RPC



- WS-Trust



# Au-delà du modèle RPC

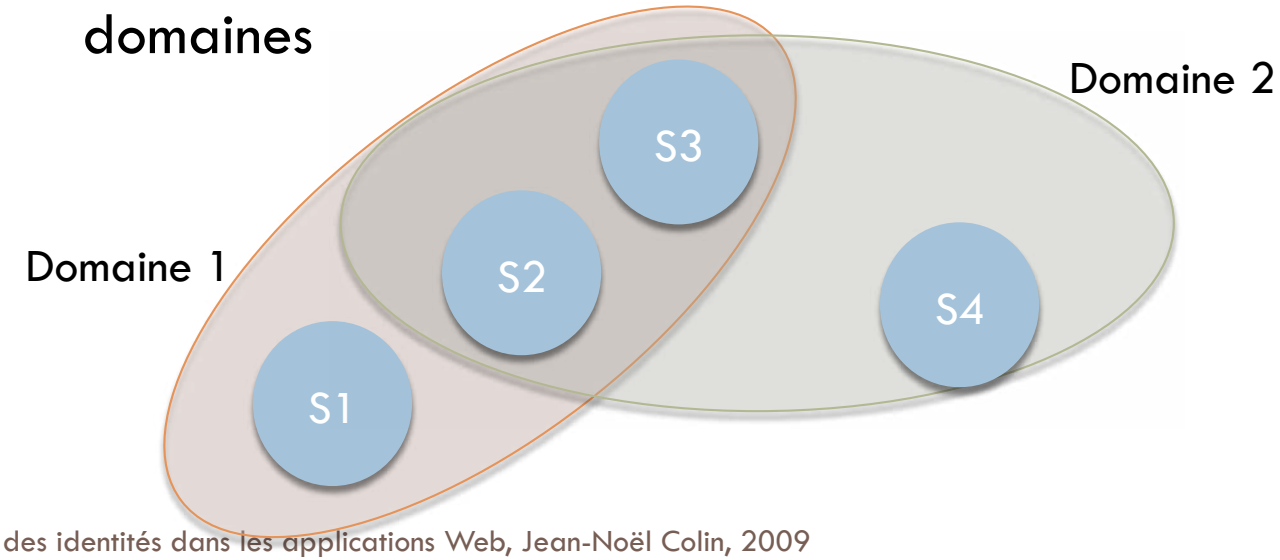


- WS-SecureConversation (OASIS v1.3 – 2007)
  - Conversation = séquence d'échanges requête/réponse
  - Etablissement d'un Contexte de Sécurité entre WSC et WSP
    - Établi par le STS
    - Défini par WSC ou WSP et communiqué au partenaire
    - Négocié entre WSC et WSP
  - Utilisation d'un SecurityContextToken comme jeton de sécurité dans WS-Security

# Au-delà du modèle RPC



- WS-Federation (v1.1 – 2006)
  - Scénarios complexes faisant intervenir des domaines de confiance multiples et imbriqués
    - Domaine de confiance: ensemble de fournisseurs de service ayant établi des relations de collaboration
    - Un fournisseur de service peut appartenir à différents domaines



# Au-delà du modèle RPC



- WS-Federation
  - Etend WS-Trust
  - Mécanismes définis
    - Meta-données: format et modèle d'échange
    - Service d'autorisation: protocole de décision
    - Service d'attributs: protocole d'interrogation
    - Pseudonyme
    - Sign-out global