



# Web Services and SOA

Laurent PETROQUE

System Engineer,  
F5 Networks

**OWASP-Day II**  
Università "La Sapienza", Roma  
31st, March 2008

Copyright © 2008 - The OWASP Foundation  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License.

**The OWASP Foundation**  
<http://www.owasp.org>

---

# Agenda

## What are

Web Services

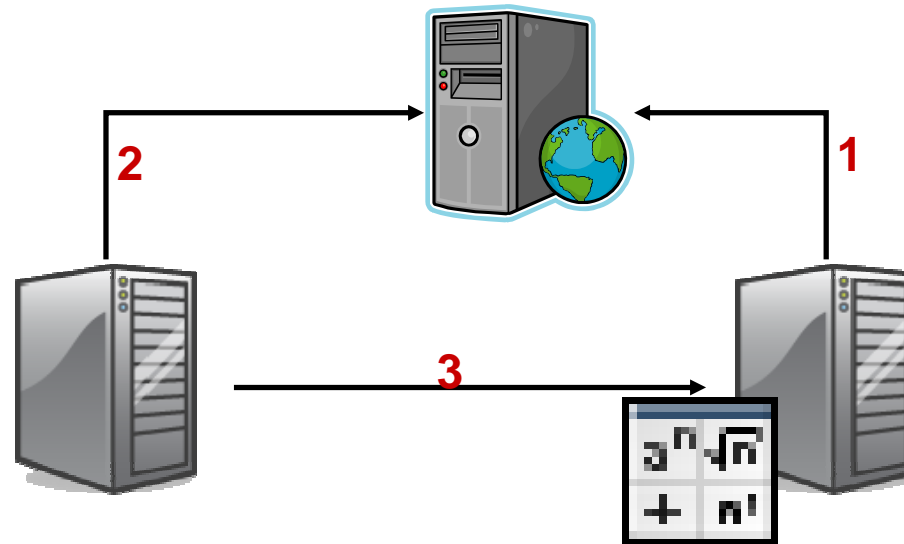
SOA

What are the threats

What is done in customers environments



# Web Service Architecture



- Service provider implements the service, publishes the service and provide the service
- Server requestor finds the service, and consumes the service
- Server registry centralized the services published by the service provider, and



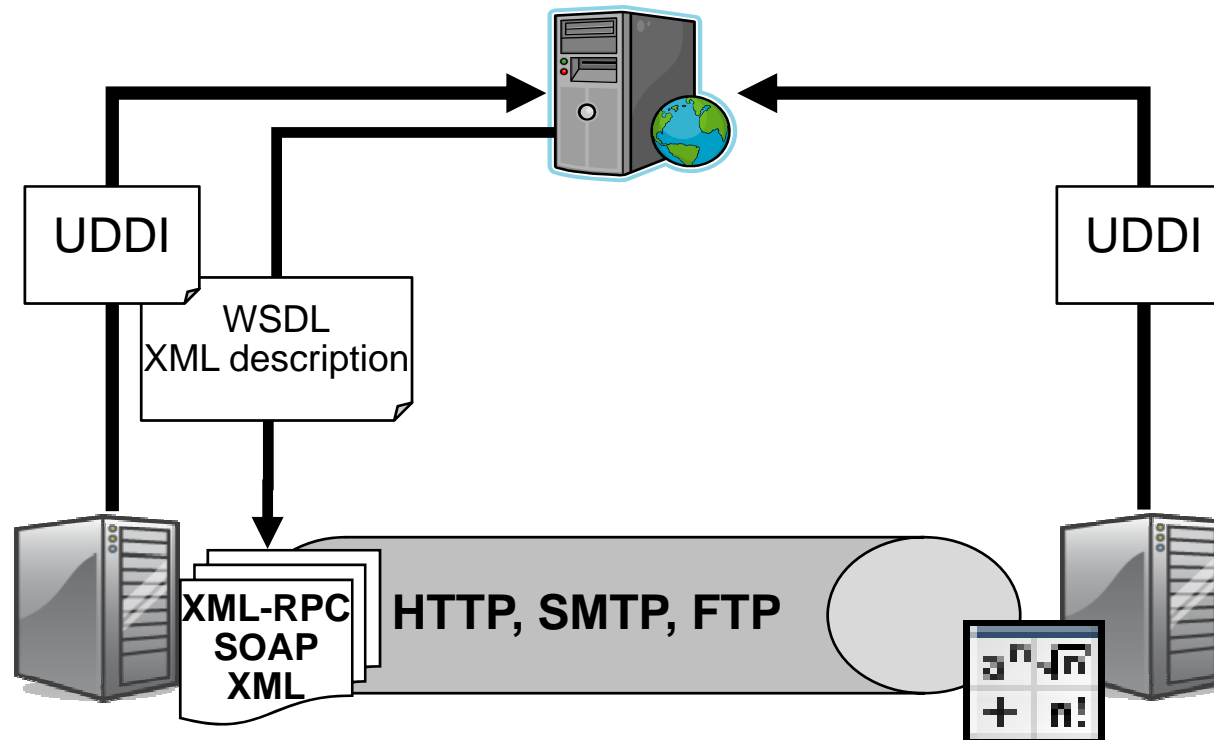
---

# Agenda

- What are Web Services ?
- What is SOA ?
- What are the threats ?
- What are the solutions ?
- What customers are doing with this and how do they protect it ?



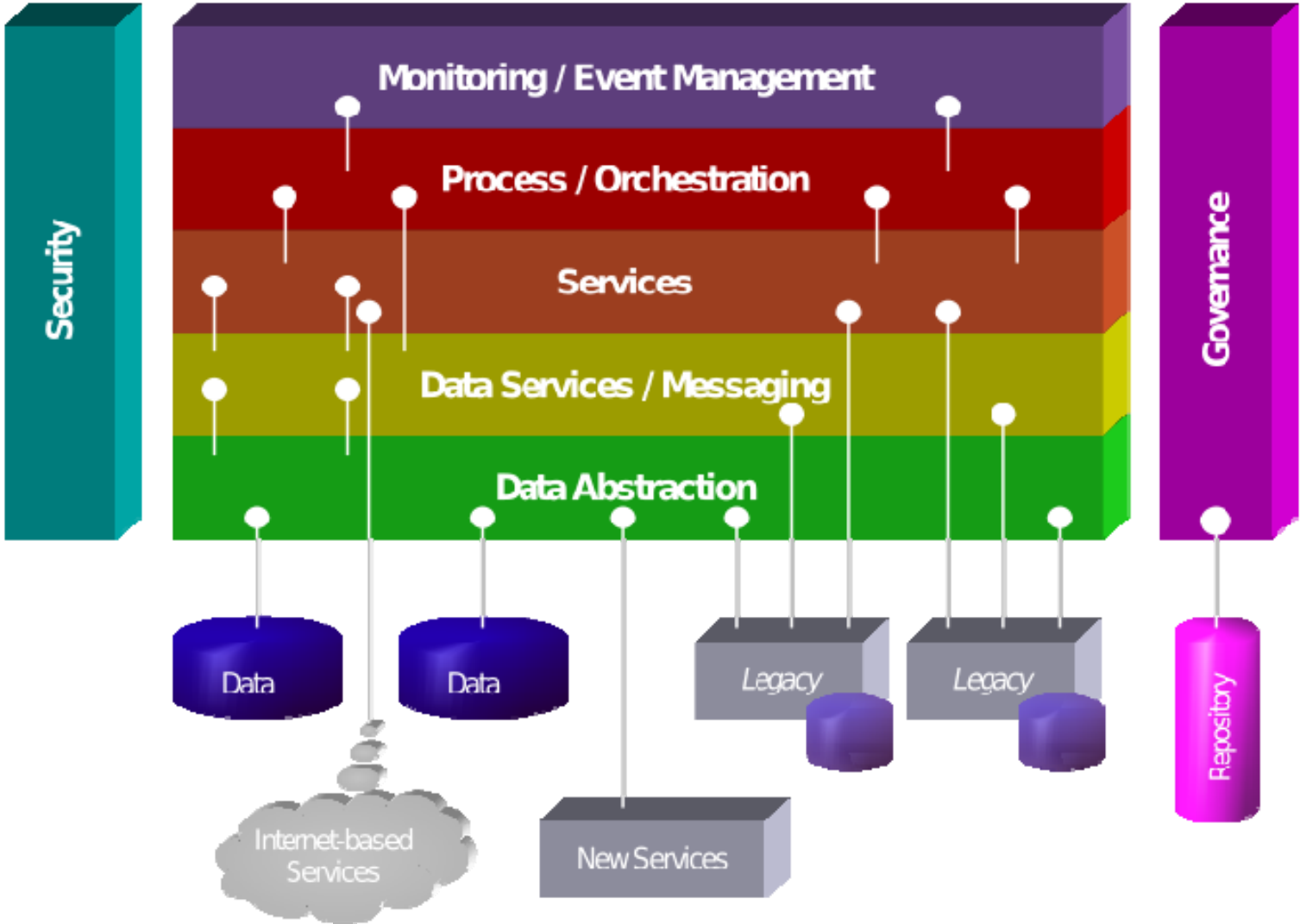
# Web Service Architecture



- Service provider implements the service, publishes the service using UDDI(Universal Description, Discovery, and Integration) to the Server registry
- Server registry centralized the services published by the service provider.
- Server requestor finds the service using UDDI, retrieve the WSDL(Web Service Description Language) and consumes the service.
- Server requestor send the message using a service transport.



# SOA : Service Oriented Architecture



# What are the base elements ?

## HTTP

- ▶ Transports information between systems

## XML

- ▶ Precise how information are exchanged

## WDSL

- ▶ Enforce the compliance of the data part of the communications

## SOAP

- ▶ Enables and regulates communications between systems



## Security and XML

- Attacks against an XML parser (mainly DOS eg forcing the parser to crash, to consume too much memory, etc)
- (re)definition  
When a DTD is included or is referred elsewhere,
  - ▶ replace the DTD/XML schema
  - ▶ attack the parser –DOS again- via a problematic URI
- XXE (Xml eXternal Entity) attack  
XXE attack is an attack on an application that parses XML input from untrusted sources using incorrectly configured XML parser. The application may be coerced to open arbitrary files and/or TCP connections.



# Security and XML

## XML BOMBS

- ▶ XML document contains too many bytes
- ▶ XML document contains too many characters (one character doesn't necessarily translate to one byte...)
- ▶ Nesting depth too deep
- ▶ Too many elements
- ▶ Too many siblings to an element
- ▶ Too many attributes
- ▶ Too many namespaces
- ▶ Element/attribute/namespace-prefix/value too long (bytes? characters?)
- ▶ recursive nesting of elements (this is not well formed XML!)
- ▶ too many times opening and closing a tag (too many push/pop stack operations)
- ▶ entity resolution depth



# Security and Web Services

- **OWASP top 10 still apply to web services**
- **WSDL Enumeration/Scanning**  
Gives useful source of information about web services.  
Sol:  
Determine the degree of exposure provided by the WSDL document.
- **Parsing Exploits**
  - SAX/DOM known common exploits on Vendor Framework
  - Custom parsers that are poorly writtenSol:  
Do not implement custom parsers.  
Use SAX-based parsing whenever possible  
Validate the XML stream size before the XML parsing
- **XML injection**
  - XML can be injected through application
  - The user-input includes XML tags which are parsedSol:  
Validation of the XML message

# Security and Web Services

## ● XPath Injection Attacks

- XPath is used to query XML documents, so like SQL, XPath is susceptible to injection.

Sol:

Validation of the XML message.

## ● XML Manipulation (i.e. CDATA Manipulation, etc)

- DTD is dangerous as it can be defined internally, externally, or both.
- CDATA can include non-legal characters in data.

Sol:

Use XSD to validate XML messages

If a DTD is used, don't allow the DTD before the root element.

Validation of the XML message



# Specific patterns in an XML application

- Possibility to check for XML patterns.  
The patterns are specific to the customers' architecture.

The screenshot displays the 'Defense Configuration' window, which is currently set to 'Advanced' mode. It is organized into three main sections:

- Defense Level:** A dropdown menu is set to 'High'.
- Applications:** This section contains two lists. The 'Selected Applications' list on the left includes 'EAServer' and 'Oracle'. The 'Available Applications' list on the right includes 'RSS20Feed', 'SampleRSS20Server', 'OWA', and 'XMLParser'. Navigation buttons '<<' and '>>' are positioned between the two lists.
- Patterns:** This section also contains two lists. The 'Enabled Patterns' list on the left includes 'SQLCommandInjection', 'SQLCommandInjection.MEDIUM.1', 'SQLCommandInjection.MEDIUM.2' (which is highlighted), 'SQLCommandInjection.MEDIUM.3', and 'SQLCommandInjection.MEDIUM.4'. The 'Disabled Patterns' list on the right is currently empty. Navigation buttons '<<' and '>>' are positioned between the two lists.



# XML Format Enforcement

```

<?xml version="1.0" encoding="UTF8"?>
<CreateQueueResponse
  xmlns="http://webservices.amazon.com/AWSSimpleQueueService/2004-10-14">
  <OperationRequest>
    <HTTPHeaders>
      <Header Name="UserAgent" value="Mozilla/4.0 (compatible; MSIE 6.0;
        windows NT 5.1)"/>
    </HTTPHeaders>
    <RequestId>
      054FT186J7SQ6CKZDJC8
    </RequestId>
    <Arguments>
      <Argument Name="service" value="awssimplequeueservice"/>
      <Argument Name="QueueName" value="Jason"/>
      <Argument Name="SubscriptionId" value="[Your subscription id]"/>
      <Argument Name="version" value="2004-10-14"/>
      <Argument Name="operation" value="CreateQueue"/>
    </Arguments>
  </OperationRequest>
  <CreateQueueResult>
    <Request>
      <IsValid>
        True
      </IsValid>
      <CreateQueueRequest>
        <QueueName>
          Jason
        </QueueName>
      </CreateQueueRequest>
    </Request>
    <QueueId>
      0KJ2NTM8MW1QNJQY2YZB
    </QueueId>
  </CreateQueueResult>
</CreateQueueResponse>
  
```

document

children

Namespace (NS)  
element

attribute  
attribute's value

depth

name

Maximum Document Size	<input type="text" value="10240000"/>
Maximum Elements	<input type="text" value="512000"/>
Maximum Name Length	<input type="text" value="1024"/>
Maximum Attribute Value Length	<input type="text" value="4096"/>
Maximum Document Depth	<input type="text" value="128"/>
Maximum Children Per Element	<input type="text" value="4096"/>
Maximum Attributes Per Element	<input type="text" value="64"/>
Maximum NS Declarations	<input type="text" value="256"/>



# XML Format Enforcement

- ❖ A DTD defines the legal building blocks of an XML document
- ❖ A DTD can be embedded in an XML document
- ❖ A DTD can be referenced in an XML document
- ❖ Possibility to work with embedded and referenced DTD.

Allow DTDs	<input checked="" type="checkbox"/>
Validate DTDs	<input checked="" type="checkbox"/>

```
<?xml version='1.0'?>
<!DOCTYPE test [
<!ELEMENT test (#PCDATA) >
<!ENTITY % xx '&#37;zz;' >
<!ENTITY % zz '&#60;!ENTITY tricky "error-prone" >' >
%xx;
]>
<test>this sample shows a &tricky; method.</test>
```

} **DTD**

Allow External References	<input type="checkbox"/>
---------------------------	--------------------------

```
<?xml version='1.0'?>
<!DOCTYPE SCHEMA PUBLIC "-//FOOBAR//DTD ARTICLES XML V1.0//EN"
"http://www.foobar.com/dtd/schema.dtd">
<test>this sample shows a &tricky; method.</test>

<?xml version='1.0'?>
<!DOCTYPE SCHEMA SYSTEM "../dtd/schema.dtd">
<test>this sample shows a &tricky; method.</test>
```




# XML Format Enforcement

- ❖ Prevent DOS using Entities in a DTD schema.

Maximum Entity Expansion

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE document [
  <!ELEMENT document (quote)+>
  <!ELEMENT quote (#PCDATA)>
  <!ENTITY r 'Rumple'>
  <!ENTITY s 'stilskin'>
  <!ENTITY % y '&#60;!ENTITY x "&#38;r;&#38;s;"&#62;'>
%y;
]>
<document>
  <quote>
    &#34;My name is &x;, but you can call me r12n.&#34;
  </quote>
</document>
```

**MEMORY**  **expansion**


```
<document>
  <quote>
    &quot;My name is Rumplestilskin, but you can call me r12n.&quot;
  </quote>
</document>
```

**ENTITY** **XML Message**

Maximum Entity Recursion

```
<?xml version='1.0'?>
<!DOCTYPE root [
  <!ENTITY ha "Ha !">
  <!ENTITY ha2 "&ha; &ha;">
  <!ENTITY ha3 "&ha2; &ha2;">
  <!ENTITY ha4 "&ha3; &ha3;">
  <!ENTITY ha5 "&ha4; &ha4;">
]>
<root>&ha5;</root>
```

**ENTITY Recursion** **XML Message**

OWASP-Italy 

# Protect XML Data

## 🌐 Encrypt XML Data

```
<?xml version='1.0' ?>
<PurchaseOrder>
  <Order>
    <Item>book</Item>
    <Id>123-958-74598</Id>
  </Order>
  <Payment>
    <CardId>
      <EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Content'
        xmlns='http://www.w3.org/2001/04/xmlenc#'>
        <CipherData>
          <CipherValue>A23B45C564587</CipherValue>
        </CipherData>
      </EncryptedData>
    </CardId>
    <CardName>visa</CardName>
    <ValidDate>12-10-2004</ValidDate>
  </Payment>
</PurchaseOrder>
```



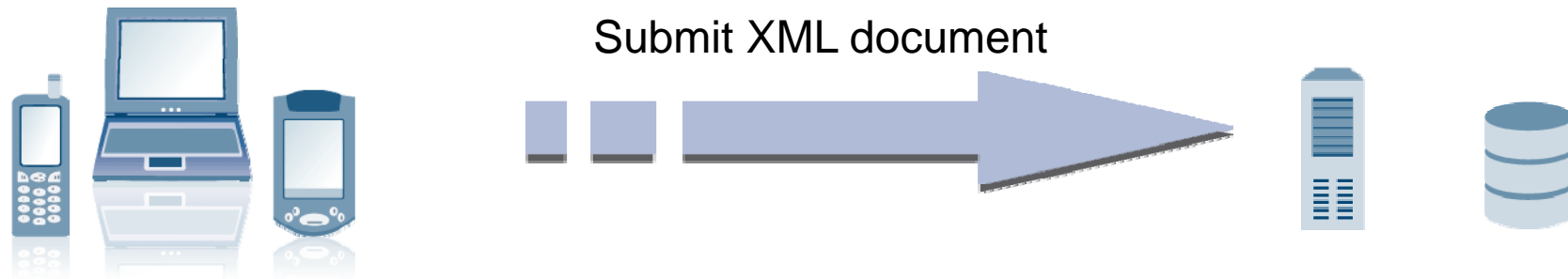
# Restrict SOAP implementation

- Provide specific WSDL
  - ▶ Publish only necessary SOAP method for each specific usage
- Rely on well known XML parser
  - ▶ Microsoft, Oracle, WebLogic, ...
- Disable DTD parsing



# What customer's are mostly doing

## Supporting XML Document



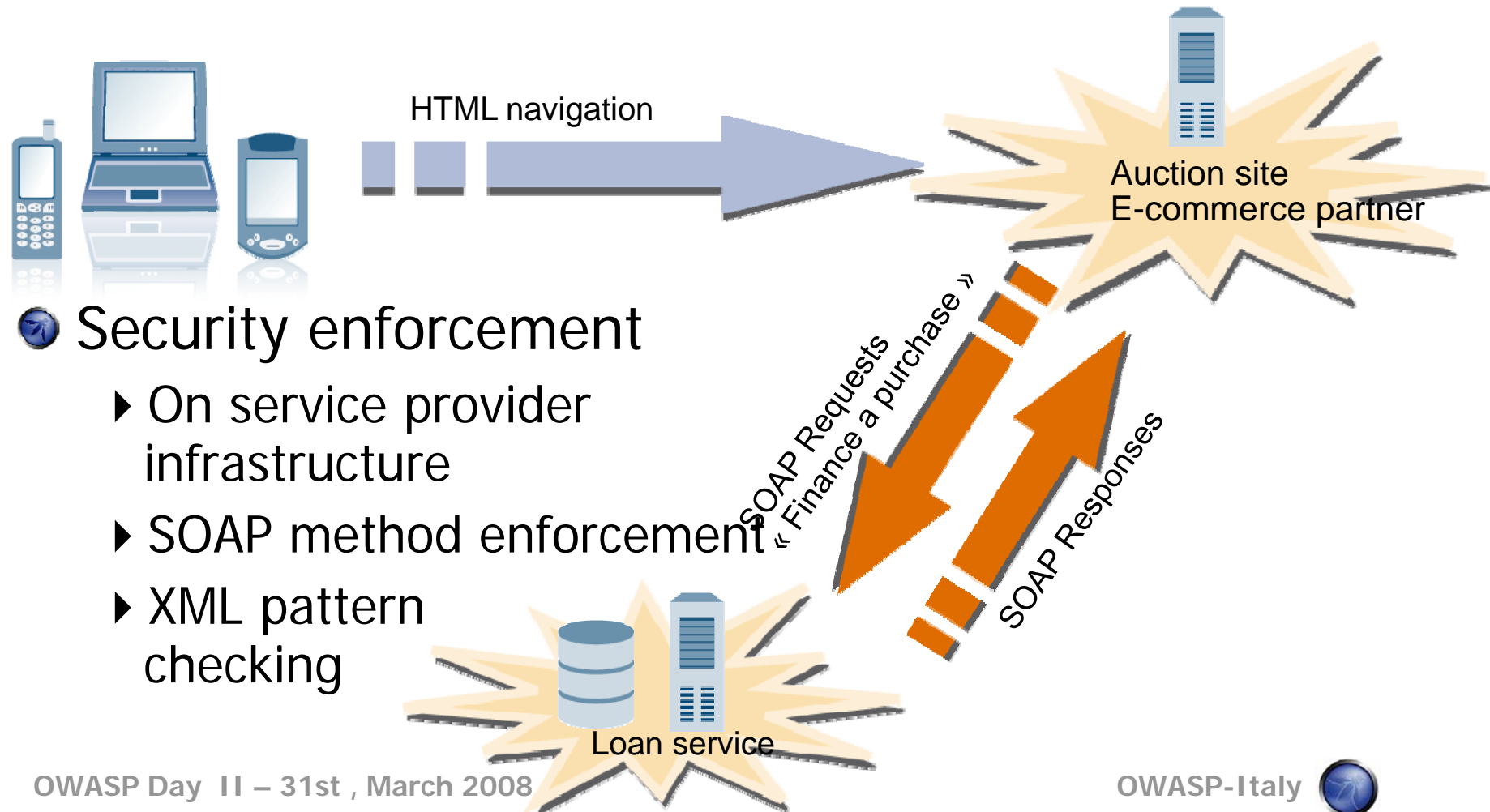
## Security enforcement :

- ▶ Check Schema
- ▶ Check SOAP methods and signatures



# What customer's are mostly doing

- Open few WebServices to partners



---

## References

-  Wikipedia

- ▶ [http://en.wikipedia.org/wiki/Service-oriented\\_architecture](http://en.wikipedia.org/wiki/Service-oriented_architecture)



---

# Thank you

