

Security Challenges in Architecting Webservices Solutions



Peter Herzum
CTO & Software Ecologist
Herzum Software

Presentation Material

Legal Copyright Notice

Please note that the session materials have been prepared and provided by Herzum Software LLC. They are not to be copied or used without written permission from Herzum Software LLC and are protected by the following legal notices.

Copyright © Herzum Software LLC 2002 (Unpublished). All Rights Reserved.

CONFIDENTIAL AND PROPRIETARY. This document contains copyrighted and confidential information proprietary to Herzum Software LLC and is provided strictly under written license with Herzum Software LLC. No part of this document may be disclosed, used, reproduced, reformatted, stored in a retrieval system, or transmitted in any form by any means, electronic, mechanical, photocopying, recording, or otherwise (whether known now or in the future) except pursuant to the terms of such license or other written agreement with Herzum Software LLC.

This document is protected by copyright, trade secret and other proprietary right laws and international treaties. Unauthorized disclosure, reproduction or distribution of any part of this document will be prosecuted to the full extent of the law and may include forfeiture, civil damages, injunction and criminal penalties.



Company website: www.HerzumSoftware.com

COSM™ website: www.ComponentFactory.org

Email: herzum@herzumsoftware.com

Phone: (847) 733 – 7539

Fax: (847) 733 – 7547

Web Service Conceptual Framework & Security Layers

◆ Webservice

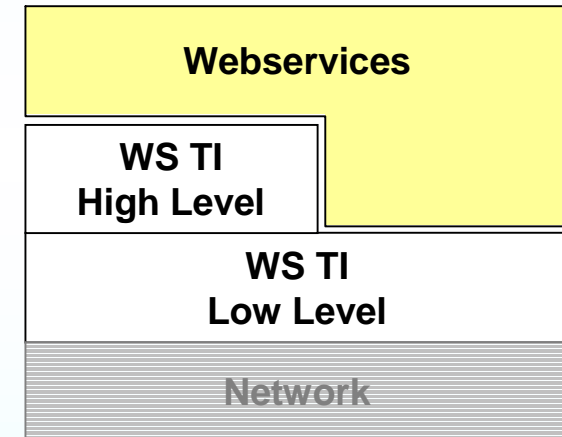
- Standalone webservice
- Value Added Service (VAS)
- Virtual Enterprise (VE)
- Webservice Community, Island of Webservices

◆ Webservice Technical Infrastructure High level

- Distributed Registry Service, Security Infrastructure, Ontology , Business Process & Workflow Modeler ...
- Webservices

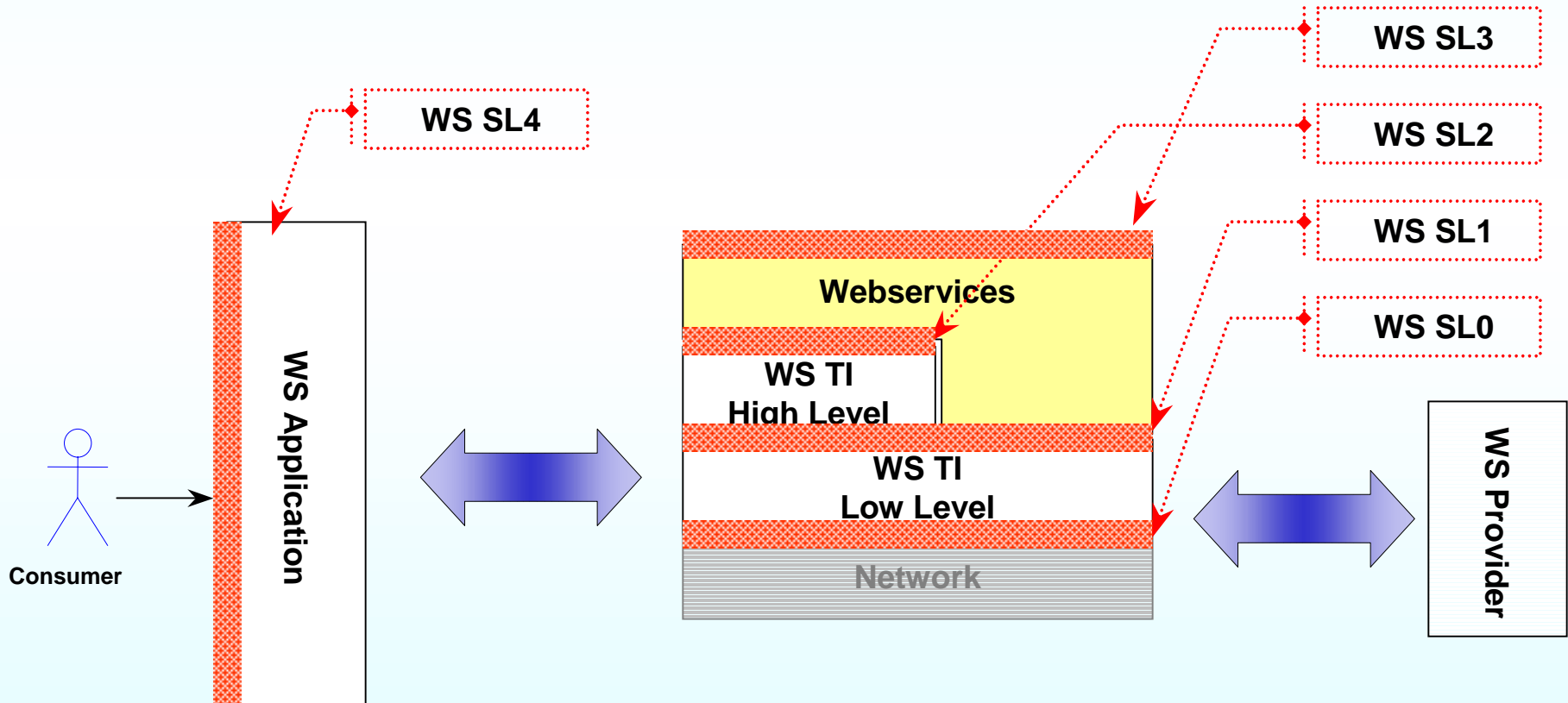
◆ Webservice Technical Infrastructure Low level

- Bare Lookup & Register
 - WS Specific Network layer
- Webservice
Technical
Infrastructure



Conceptual Framework & Security Layers

- ◆ Security applies to each different layer in different ways
- ◆ Each WS Security layer has a tradeoff between data confidence and performance drawbacks



- ◆ **Security Level 0**
 - Network based, VPN, IPSec
 - Address Authentication at the IP level for all the upper layers
- ◆ **Security Level 1**
 - Technical Infrastructure Low Level
 - Like the DNS operations, it is generally Public, but it is possible to consider an alternate secure Technical Infrastructure
- ◆ **Security Level 2**
 - Technical Infrastructure High Level
 - Login, authentication to gain access to the Community support system
- ◆ **Security Level 3**
 - Webservice
 - Most important in this presentation
- ◆ **Security Level 4**
 - Not specific to webservices

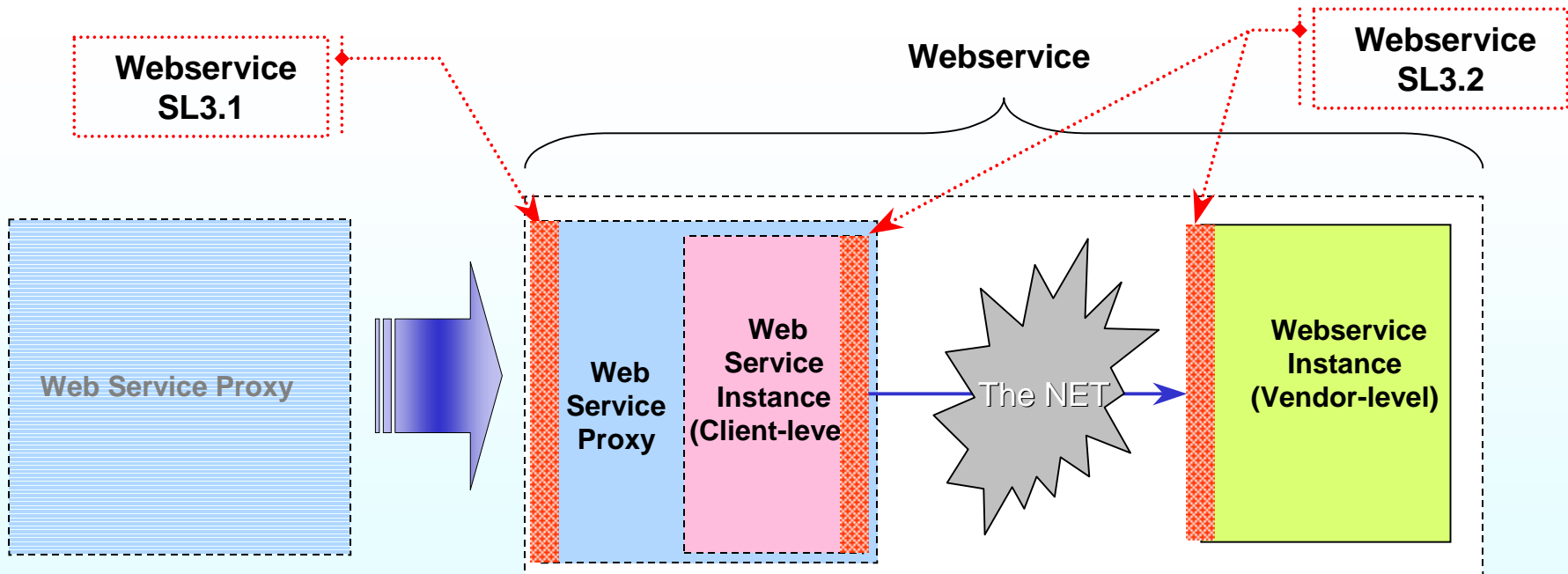
- ◆ **Inherently public, again the similarity with DNS**
 - **Lookup is indeed a public kind of service**
 - **Deploy is a public Service (services can be not)**
 - **Retrieve can be public**
- ◆ **“Deny of Service” (DoS) can not be avoided, but can be detected**
- ◆ **Example: Jini™ Technological Infrastructure provides a distributed Registry, more concerns are required when applying security models in this case:**
 - **Each registry node have to replicate security processes**
 - **Security need to be extremely lightweight**
 - **Be careful with performance drawback in case security is enabled at this level**

Security Level 1 Negotiation

- ◆ **It is a massive distribution system of information, high frequency of use:**
 - **It must be quick, lite, flexible, security should be avoided is performance is an issue**
- ◆ **Deploy is a public Service:**
 - **Vendors should be free to Deploy services without restriction, but...**
 - **... we could guarantee Users about what they are going to execute**
- ◆ **Deploy as a restricted Service:**
 - **Requires authentication of Registrars (like DNS Registration Authority)**
- ◆ **Secure and un-secure Deploy mechanism can leave side by side, it's then up to the client to decide if to trust the Service or not**
- ◆ **Example: Jini™ sends Proxies (executable code) to clients and Security is an issue (the Proxy could reveals security leaks)**
 - **Client identification shall be required prior to Retrieving**

- ◆ It aims at giving the Technical Infrastructure High Level basic login and authentication mechanisms
- ◆ It resemble Security Level 3 but it has less interoperability requirements because it applies to a quasi static set of Services
 - Authentication and Certification of the User
- ◆ Ontology System
 - Distinguish protected from public ontologies, administrator from generic user,...
- ◆ Distributed Registry
 - Browsing is public
 - Updating Requires authentication
- ◆ Business Process & Workflow Modeler
 - Requires authentication
 - Difference between modeler, executor, registrar,...

- ◆ **Webservice Instance tier needs to trust parties:**
 - The Client needs to be sure Service/Vendor is identified
 - The business logic needs to be sure the Client is identified
- ◆ **The communication channel have to be secured and remote WS Instance be identified**
- ◆ **These two layers are referred to as SL3.1 and SL3.2**



- ◆ **Webservice SL3.1**
 - Login procedure
 - Identification
 - Policy rules
- ◆ **Webservice SL3.2**
 - **Attackers could mimic the Webservice Instance at both the client and the server-level, Webservice Instance tiers needs to perform some two-way certification mechanism**
 - **Secured Communication channel**
- ◆ **Such capability is not mandatory, a Webservice Client can decide to take advantage of it, even if no Certification has been provided about the Webservice Proxy**
- ◆ **Webservice SL 3.1 deals with Security at business level**
- ◆ **Webservice SL 3.2 apply plain Technical Infrastructures**
 - **the IT marked is plenty of ready to use solutions (https, SSL, TLS, SSH,...)**

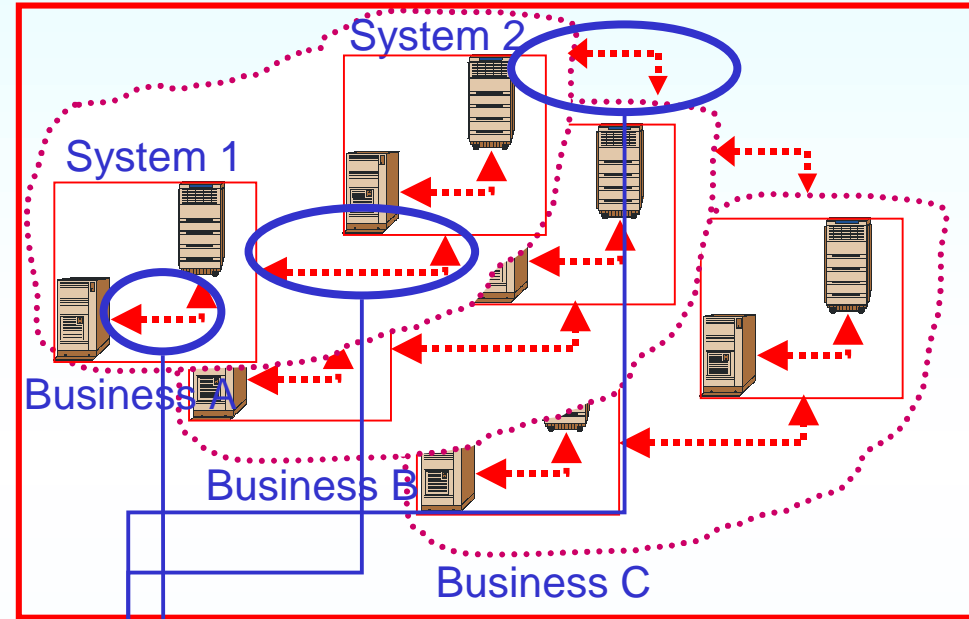
- ◆ **Webservice SL 3.1 is the most strategic for this presentation, it deals with security at the business level**
 - From now on, we will refer to this security level
- ◆ **If there were no other Webservice Community, this would only be a private matter**
- ◆ **We need to address the criteria for Federating Communities: we are not alone out there**
 - WSDL, SOAP, UDDI, PKI are not the silver bullets
- ◆ **The primary issues are:**
 - Handles intra Community information exchange
 - Handles extra Community information exchange
 - *Standardization and Interoperability*

***WS Security Level 3.1 and
Levels of Information Exchange***

Levels of Information Exchange (LIE)

- The different levels of information exchanges typically require:
- different architectural approaches
 - different technologies
 - different standards

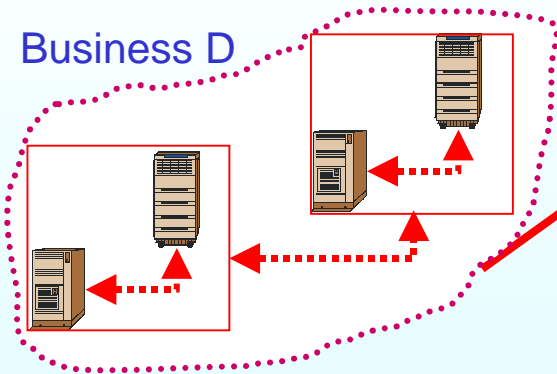
e-Business Communities, Virtual Enterprises



Information exchanges

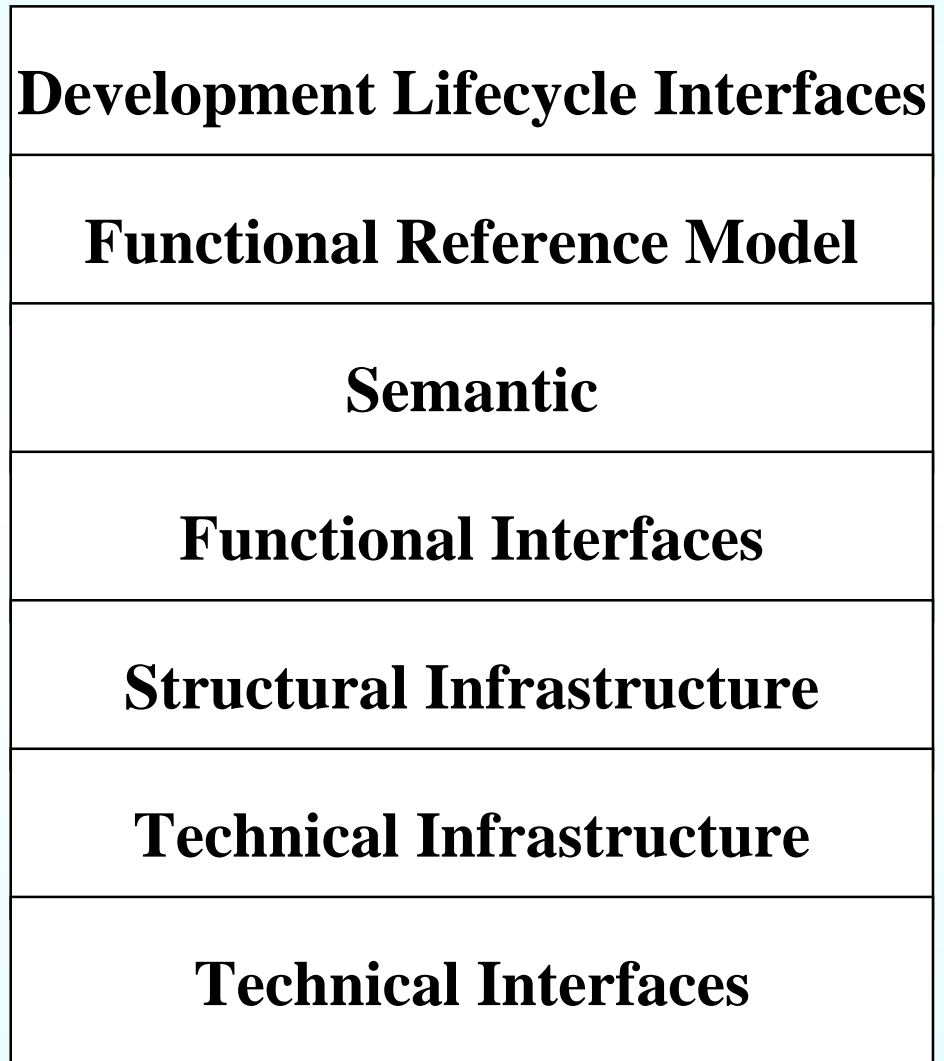
- Intra-system (L1)
- Inter-systems, intra-enterprise (L2)
- Inter-enterprises (between partners) (L3)
- Inter-communities (L4)

Business D

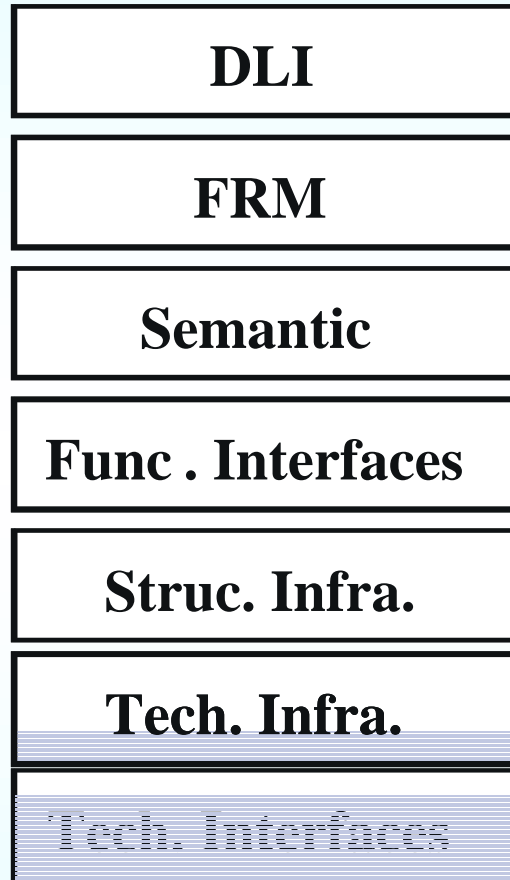


Also known as Seven
Layer Protocol Model
or "7LP"

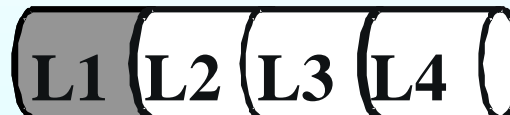
See www.cutter.com/consortium/architecture/omg.html or
[Herzum 2000] book for more
info



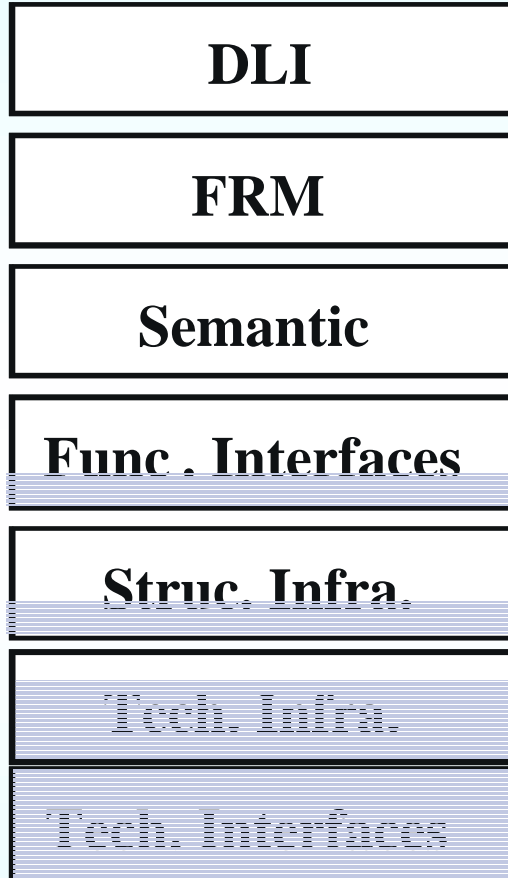
- ◆ No security effort down to the Structural Infrastructure level
- ◆ Minimum Technical Interfaces
- ◆ Some technical Infrastructure



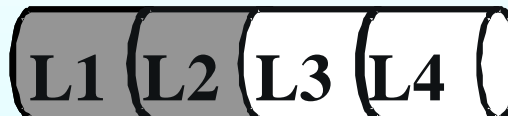
4LIE



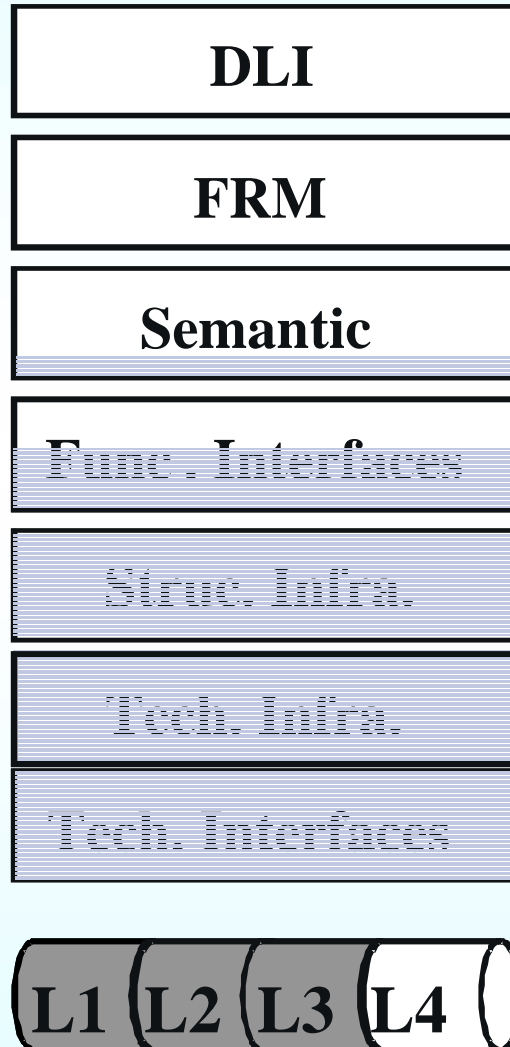
- ◆ **Some Structural Infrastructure and Functional Interfaces are needed**
- ◆ **Yet the Technical Infrastructure need not be entirely applied**



4LIE



- ◆ It starts to be a tough challenge
- ◆ Some Semantic effort needs to be made, the rest is left to the common background



4LIE

Security Level 3.1 & LIE4

- ◆ At the L4, there is the need to push at the Semantic Level
- ◆ Not only define a common standard in Policy Rules but Security needs a standard semantic model

DLI

PRM

Semantic

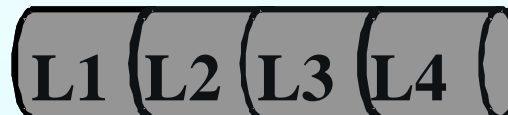
Func. Interfaces

Struc. Infra.

Tech. Infra.

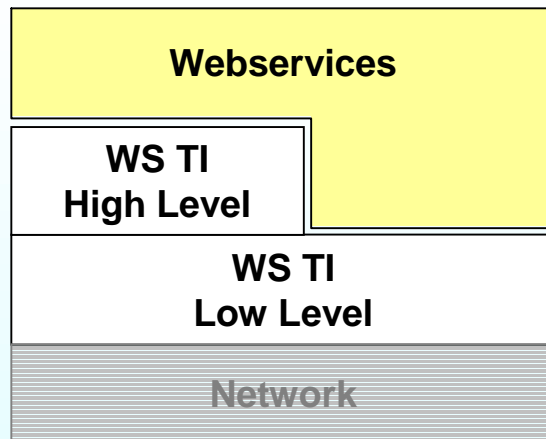
Tech. Interfaces

4LIE

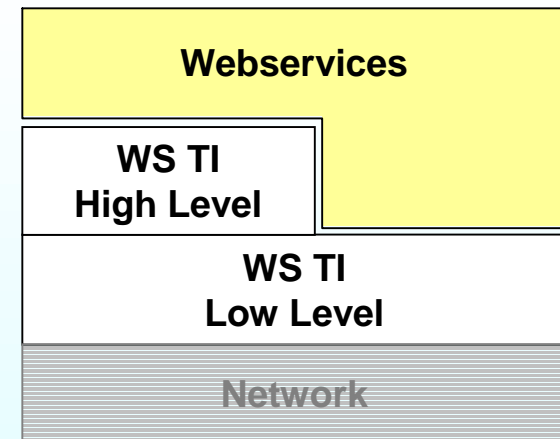


Security for Federation of WS Communities

- ◆ We realize that establishing an Architecture for Security is not just a matter of Technical Interfaces and Infrastructures
 - WSDL, SOAP, UDDI, PKI are not enough
- ◆ Needs to define interoperability solutions to the different approaches used in external Communities:
 - Different CA engines
 - Non XML based PKI vs. XML PKI
 - Incompatible signature algorithms
 - Consistent Semantic in Policy Rules
 - Identification name conflict of parties



WS Community A



WS Community B

*Security:
Brief Example in Tourism*

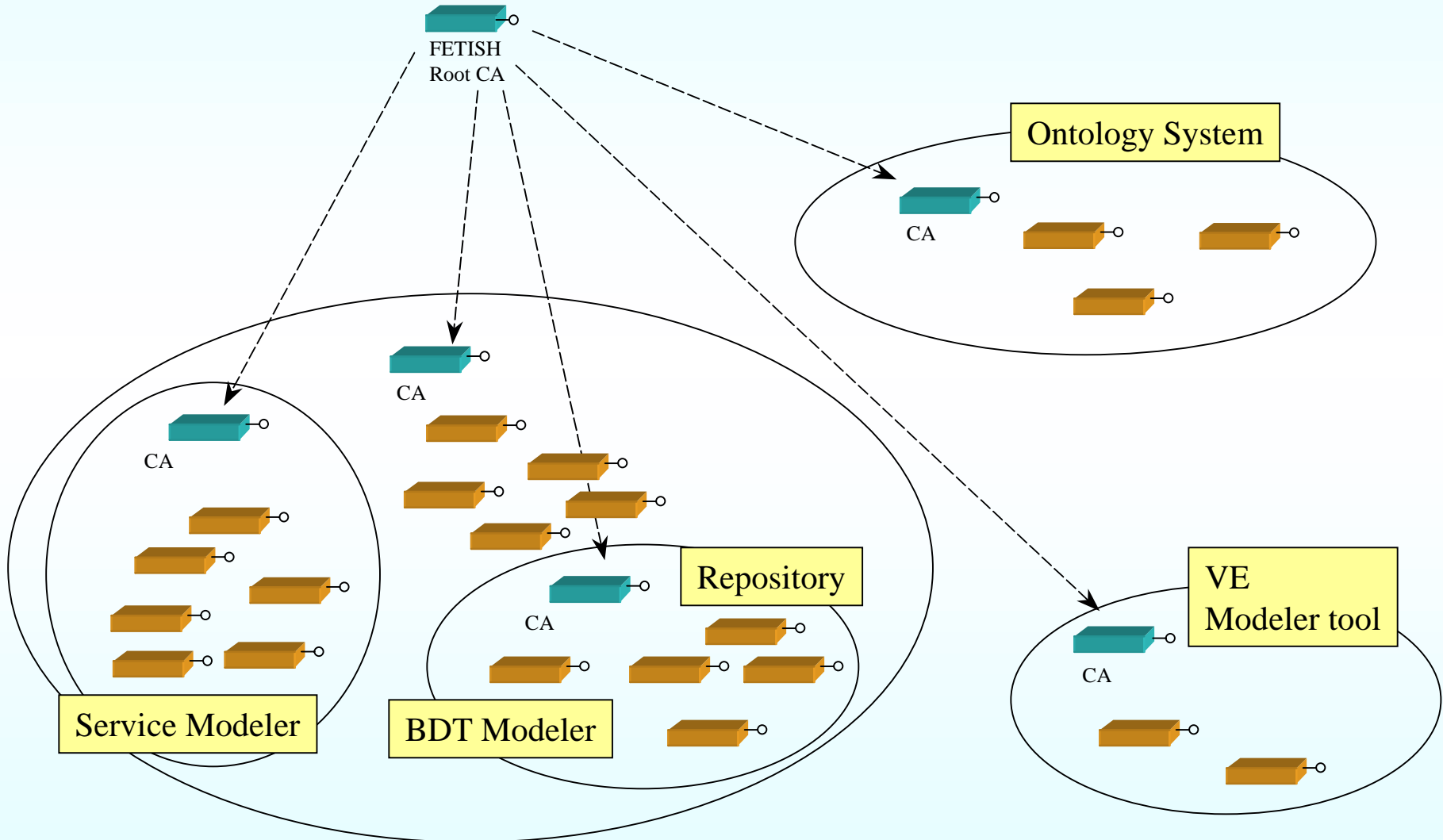
*Federated European Tourism Infrastructure System Harmonization
an intelligent environment
for interoperable Value Added Services and Systems*

- ◆ **The value proposition of FETISH**
 - **Connect information systems (an Internet-based service bus) and value-added services (VAS) into a community where a critical mass of European resources and data can be shared in a network across a wide geography**
 - **A Reference Service Network that will create a competitive advantage in the travel service provider market**
- ◆ **The vision:**
 - **to be a leading supplier of distributed service solutions, that will provide tourism information and service integration.**
 - **By using a Service Oriented Architecture, FETISH gives to V.A.S.providers an high-level, open architecture model that supports the production of reliable, platform-independent services**

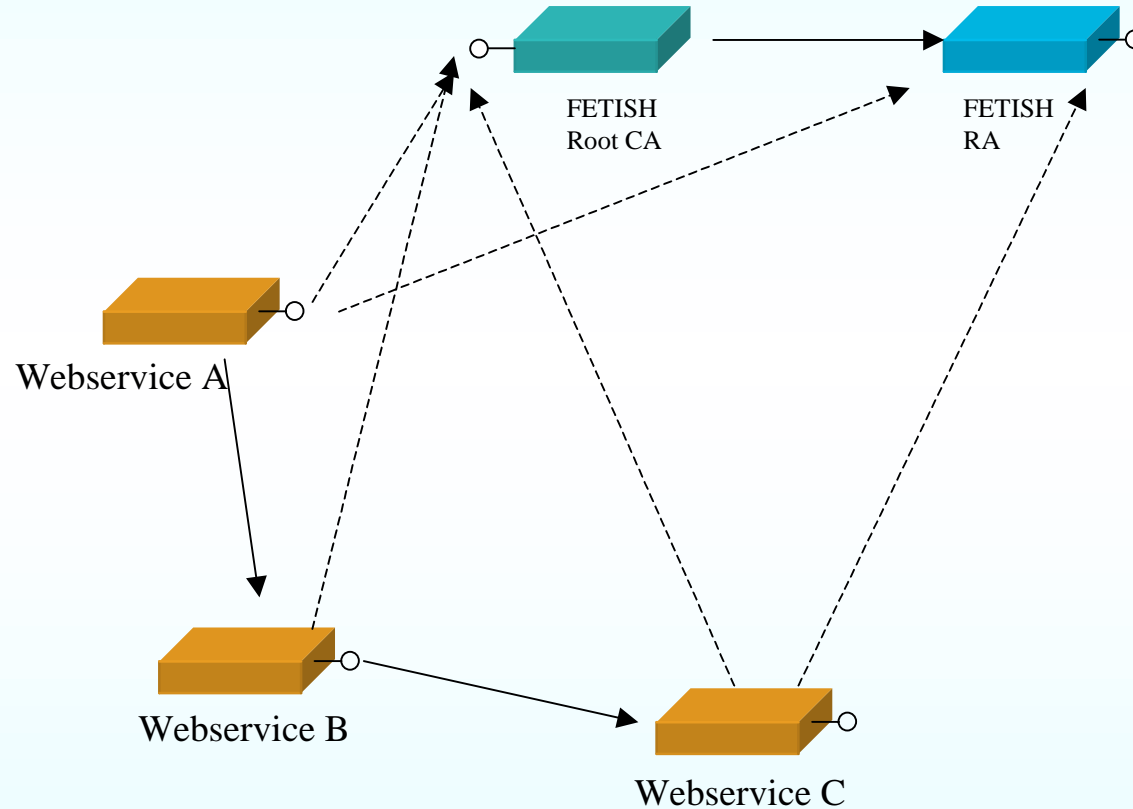
- ◆ **Security Level 0:**
 - **Public:** no security has been implemented so far
- ◆ **Security Level 1:**
 - **Public:** no security
 - **Reserved:** a separate access point allows to make Secure Registration via IPSec
- ◆ **Security Level 2:**
 - **One level hierarchy of CA, one root CA and a certification path for each CAs of the Infrastructure Services (Ontology System, Repository, ...)**
 - **Each request is attached with the Signed Public Key**
- ◆ **Security Level 3.1:**
 - **Proxies can be optionally signed and certified by the root CA**
 - **FETISH gives Services the same Security mechanism used for the Structural Infrastructure (SL2) but the certification path is always resolved in the root CA**
- ◆ **Security Level 3.2: is not mandatory because it adds no benefit to the Federation**

- ◆ **An hierarchy of CA is a scalable and feasible approach for L3**
- ◆ **PKI Certificate already defines a slot where to put Policy rules**
- ◆ **Policy rules**
 - **Policy rules are harmonized in the FETISH Community**
 - **Still need to define a feasible solution when federating other WS Communities**
- ◆ **Java™ Security Package has been adopted**
- ◆ **This approach allows cross certification and authentication of both users and Services**
- ◆ **Services who join our framework do not need to adopt our Security Framework**
 - **They will loose the advantage of the interoperability features, it will not be possible to build a VAS from it**

WS Security Level 2

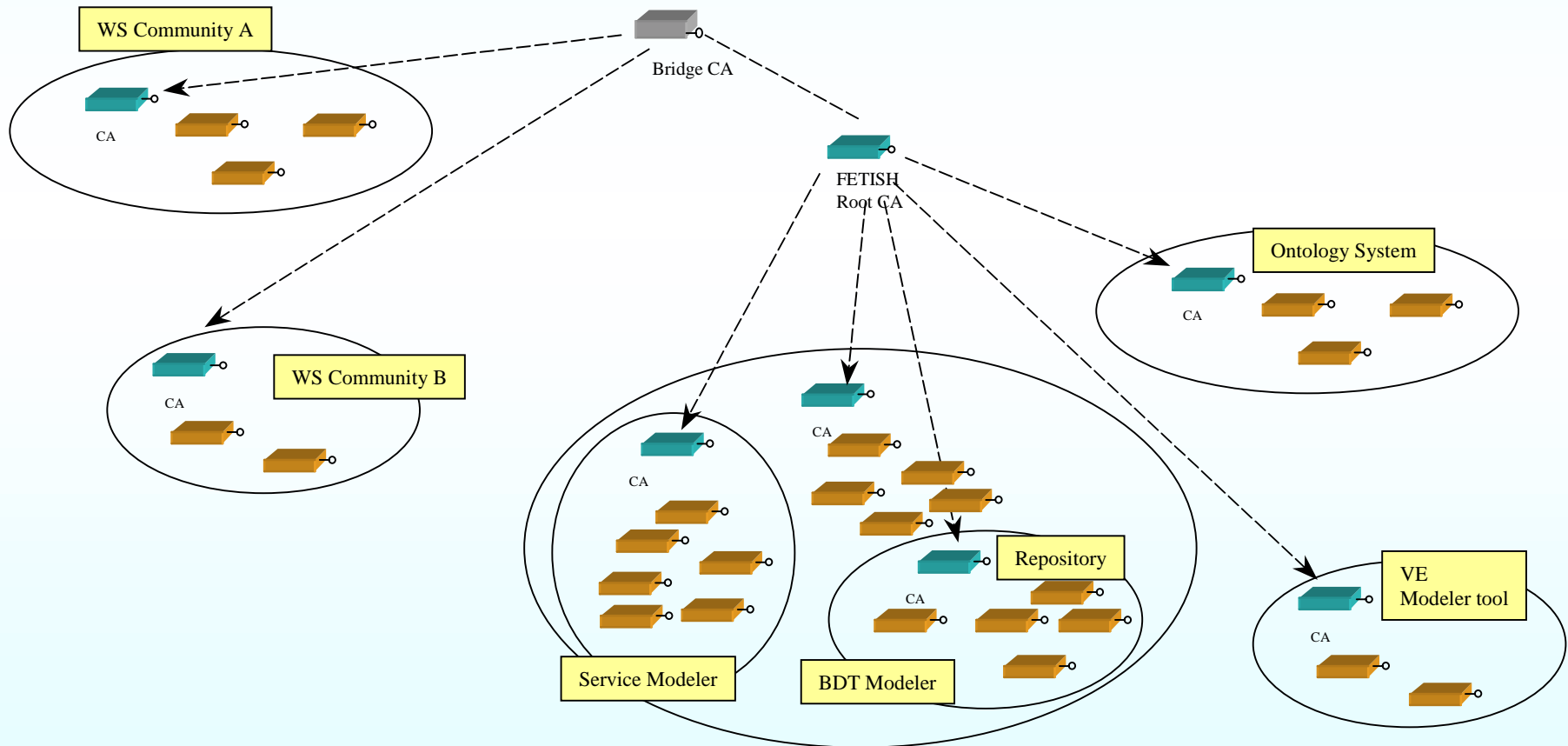


WS Security Level 3



- ◆ **Several WS Communities (Crumpet, Estia, Etour, Palio,...) will soon join the FETISH Community, ad-hoc strategy need to be taken into account**
 - **Need to take into account different standards and semantics**
 - **One projects is using Kerberos as the authentication protocol that is not compatible with PKI**
 - **Another has not enabled any security mechanism, a secure -cross Community- VAS can not be realized**
- ◆ **We will adopt a Bridge CA**
 - **We need to agree on Certificate standard**
 - **All the WS Communities need to trust the bridge CA**
 - **Mapping Policies Descriptors is still an issue**
- ◆ **Projects using PKI are not adopting XML**
 - **Need convergence of PKI and XML**

Planning for LIE4



PKI and XML

- ◆ **Certificates, Digital Signatures and Policy Rules shall be exchanged in an XML compliant protocol and not as binary coding**
- ◆ **XML is a must to drive PKI solutions in a webservice environment and address all the Interoperability issues**
- ◆ **PKI has to become webservice-enabled**
 - **Need convergence of PKI and XML**

- ◆ For all these solution to happen, Certificates and Digital Signatures shall be exchanged in an XML compliant
- ◆ **XML Trust Center** is a vendor-neutral source of information
 - It has been set up to aggregate information about advances in XML and public key infrastructure technologies
 - *Sponsored by Verisign*
- ◆ Specifications available so far in the field are:
 - **XML Key Management Specification (XKMS)**
 - **XML Signature**
 - **Security Assertion Markup Language (SAML)**
 - Some others ...
- ◆ All these specifications are in the working stage

- ◆ **A XML Key Management Specification-compliant service supports the following operations:**
 - **Register:** XKMS services can be used to register key pairs for escrow services. Once keys are registered, the XKMS-compliant service manages the revocation and recovery of registered keys, whether client- or server-generated
 - **Locate:** The Locate service is used to retrieve a public key registered with an XKMS-compliant service
 - **Validate:** The Validate service is used to ensure that a public key registered with an XKMS-compliant service is valid, and has not expired or been revoked
- ◆ **Instead of coding complex PKI functions into applications, webservice vendors simply use XML to "hook in" to a webservice trust utility, which performs most of the complex tasks related to key management.**
- ◆ **By delegating the complexity of PKI to trust utilities, webservice developers are able to focus on core applications**
- ◆ **Submitted to W3C in March 2001 (www.w3.org/TR/xkms)**
- ◆ **Currently only Verisign and Entrust have enabled XKMS services**

- ◆ **XMLSignature** defines the schema that enables data associated with digital signatures to be modeled in XML
- ◆ XML Signature is managed by joint Working Group of the Internet Engineering Task Force (IETF) and W3C
 - Specification released 20th of August 2001
- ◆ This Working Group does not address broader XML security issues including XML encryption and authorization

- ◆ **Security Assertion Markup Language (SAML), pronounced "sam-I"**
- ◆ **It's effort by Oasis to define a way to communicate security-based assertions**
 - **An Industry standard for XML-based security standard for exchanging authentication and authorization information, based upon:**

“SAML allows companies to securely exchange authentication, authorization, and profile information between their customers, partners, or suppliers regardless of the security systems or e-commerce platforms that they have in place today.” (from OASIS Web Site)
- ◆ **It's a meta standard, it aims at resolving interoperability issues between security systems**
- ◆ **Submission to a Committee Specification to the OASIS membership for its approval is planned by 1 March 2002**
- ◆ **Promoted by OASIS**

Final Considerations

- ◆ **Webservice can be, and should be, the supporting infrastructure for PKI services**
- ◆ **Security needs a semantic model**
- ◆ **Need XML standard for Policy Descriptors**
- ◆ **Following the webservice philosophy, a single access point solution for webservice, even if charming, have to be avoided**
- ◆ **Webservice Security Models have to take advantage of the Webservice Infrastructure**
 - **I.e. Certification Authority (CA) and Certificate Repository have to be to webservices themselves**
- ◆ **CA, RA part of the WS Technological Infrastructure High Level:**
 - **WebService Security Service**
 - **WebService Ontology Service**
 - **WebService PKI Service**
 - **Key Pair generations**
 - **Signature validations**
 - **Policy Update Managers**

- ◆ **Need standards for Policy Descriptors**
 - **Must be interchangeable**
 - **Must be XML based**
- ◆ **VAS and VE need to define coherent policies within Webservices**
- ◆ **Policy Rules have to be modeled upon an Ontology**
- ◆ **Policies shall NOT be stored in a centralized repository but distributed**
 - **CA and Certificate offers a valuable location where to store Policy Descriptors**
 - **Decisions shall be based upon policy size and interoperability strategies**

- ◆ **There will never be a unique Security Model**
- ◆ **There will never be “silver bullet” for security issues, either technological, tools or architectural**
- ◆ **There will hopefully be some XML specifications for PKI (Services, Signatures, Protocols, Policy Descriptors and Entities)**
- ◆ **Bridge and mapping patterns will be applied to connect Webservice Communities**
- ◆ **Each Webservice Community will have it’s own Security model and Interoperability will be an issue especially at the semantic level**
- ◆ **Security for Webservice is more then having just a set a secure and trusted services, this does not mean they are “Good” members of the federation**



- ◆ www.componentfactory.org
- ◆ www.webservices.org
- ◆ www.fetishproject.com
- ◆ www.w3c.org/Signature
- ◆ www.xmltrustcenter.org/index.htm



- ◆ **Gamma, Helm, Johnson and Vlissides, “Design Patterns”. Addison Wesley, 1995**
- ◆ **Herzum, Peter and Oliver Sims. “*Business Component Factory*”. John Wiley & Sons, 2000, New York, NY**
- ◆ **R. Housley and T.Polk, “Planning for PKI”, Wiley, 2001**
- ◆ **Scott Oaks, “Java Security”, O’Reilly, 1998**