Security in SOA
WS/XML-Security

Software and Solutions Group

Marcelo da Cruz Pinto

October, 2007

Agenda

- Context of this talk
- Securing service oriented architectures
- Introducing the technology
- Cryptography (quick overview)
- WS-Security in action
- Performance Considerations
- WS-Security using Open Source
- References
- Next talk (in this series)
- Summary – Q&A

Context of this talk

Tecnologías Emergentes para Aplicaciones Empresariales

13/09. **Introducción al Mercado SOA, Evolución y Tendencia** - *Mariano Cilia*
SCA y Tuscany - *Mario Antollini.*

20/09. **BPEL, un Enfoque a SOA Basado en Procesos** - *Alejandro Houspanossian*

27/09. **Web 2.0: Elementos y Tendencias** - *Sebastián Salvucci*

04/10. **Seguridad en SOA (WS-Security)** - *Marcelo Da Cruz Pinto*

11/10. **Sistemas Peer-to-Peer (P2P)** - *Cristian Fiorentino*

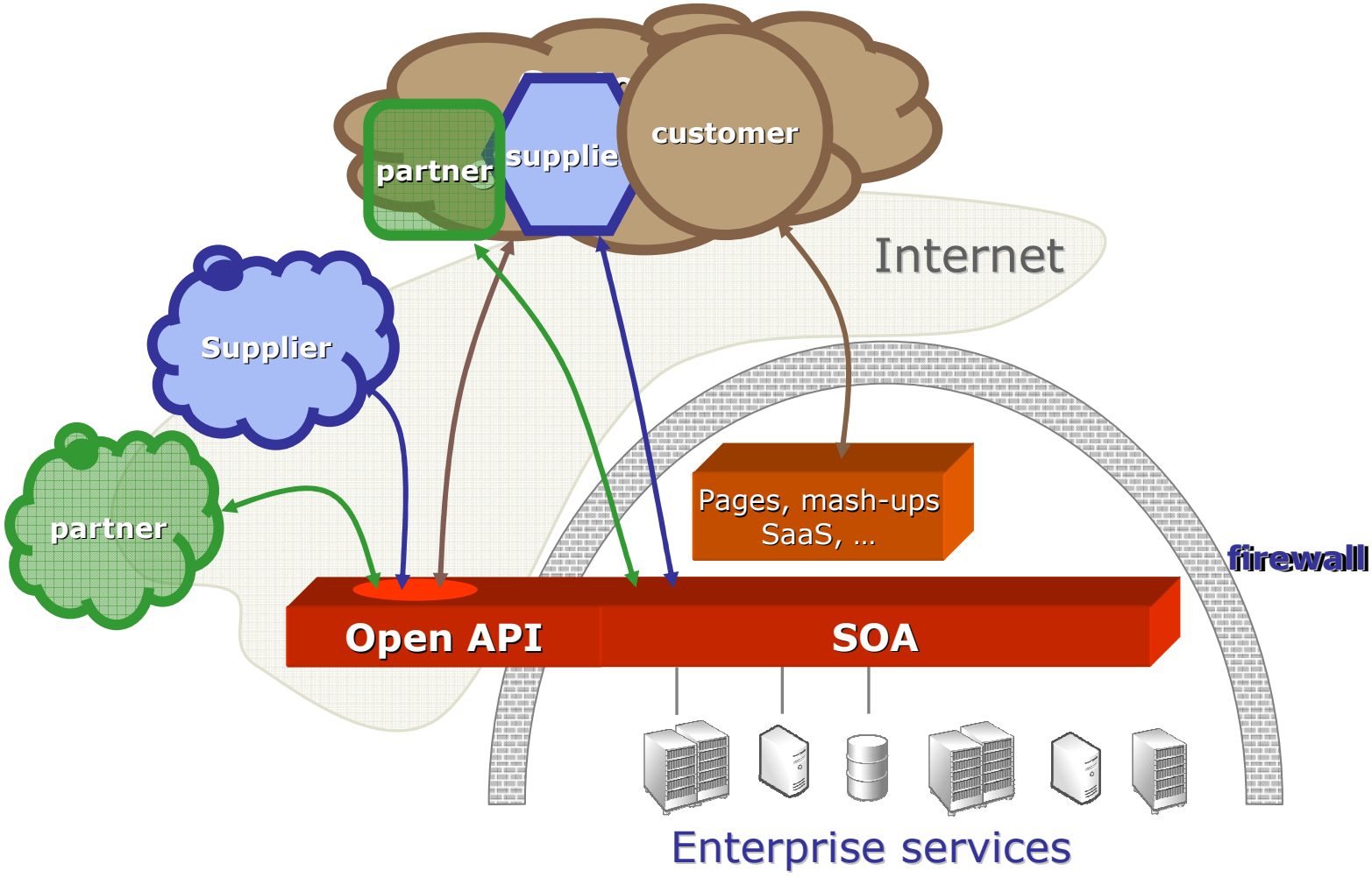
18/10. **High-Performance Computing (HPC)** - *Silvana D' Cristofaro y Andrés More*

25/10. **Virtualización** - *Diego Palmisano*

01/11. **Administración y monitoreo de recursos de SW y HW** - *Sebastián Ganame*
Cierre del ciclo de charlas -*Mariano Cilia*

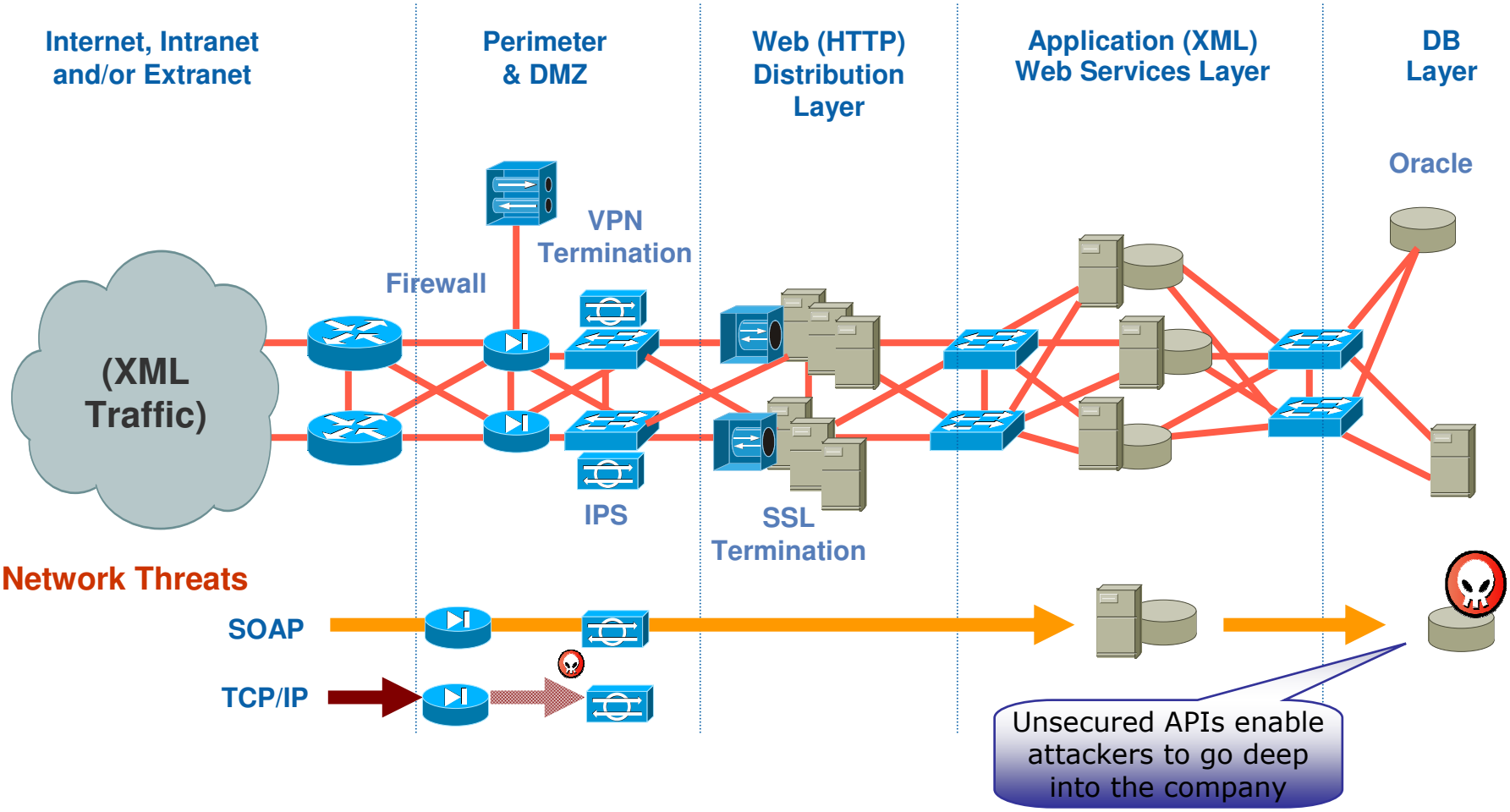
Securing service oriented architectures

The evolution of Internet services



Anatomy of the SOA Security challenge

The need for content based security



Perimeter defense is not enough, WS-Security can help with data integrity and authentication

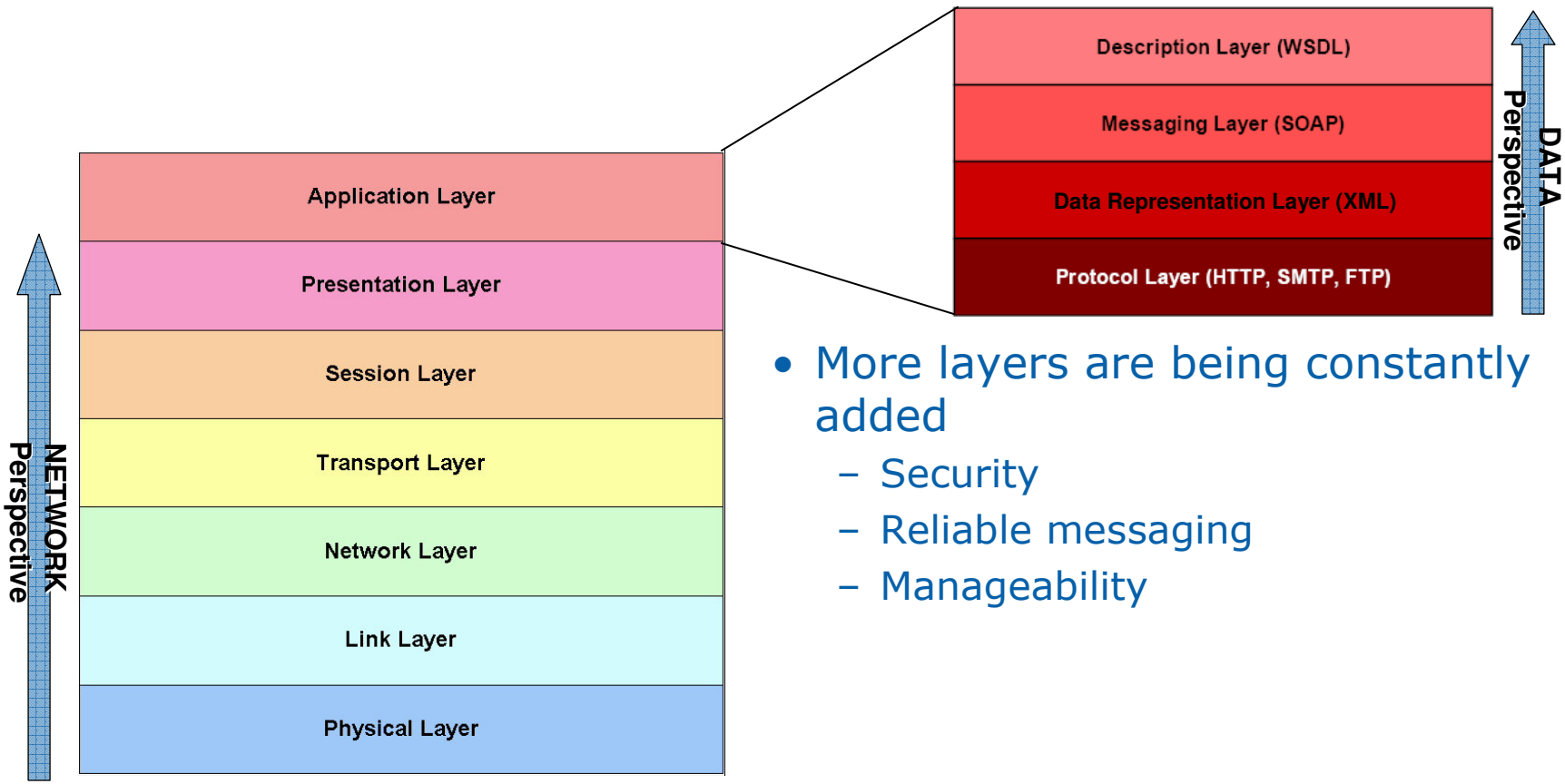
Why looking at SOA, Why Security?

- **SOA is a huge success…**
  - **71% of the companies** have already invested in SOA, 85% is predicted for 2008 (Source: IDC 2006)
  - **Microsoft** just launched their SOA stack called Windows Communication Foundation (March 2007)
  - **Sun** released Java 6, the first version to include a SOA stack (Dec 2006)
- **B2B integration is becoming a reality**
  - $7000B will be spent on B2B transactions in 2007 (i.e. 45% of the total) (Source IDC)
- **SOA simplifies B2B but also exposes a bigger attack front**
  - Shared APIs allow partners and attackers to access the core business apps.
  - Automated attacks are easier than ever
    - Web Services Description Language (WSDL)
    - Universal description, discovery, and integration (UDDI)
    - Off the shelve software stacks are everywhere
  - 75% of hacks occur at the Application/Service level (Source: Gartner)
- **WS-Security is to SOA what SSL is to HTTP (albeit at different rates ☺)…**
  - WS-Security is the **only standard way to secure SOA**
    - backed by the big guys (IBM, MS, Verisign) at OASIS (2006)
    - WS-Security adoption doubled in 2005

WS-Security is needed to support SOA growth…

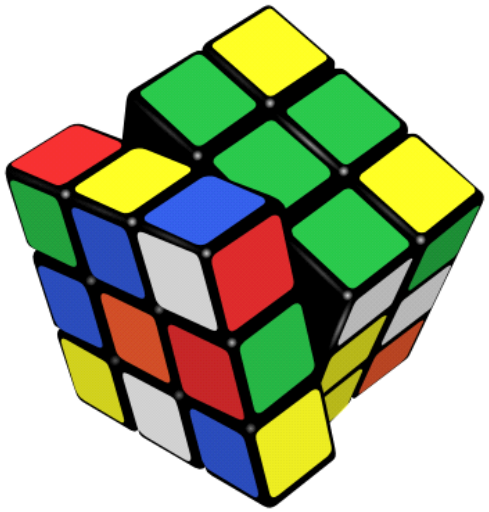
Introducing the technology

The OSI model meets the SOA Stack



- More layers are being constantly added
  - Security
  - Reliable messaging
  - Manageability

Cryptography
Quick overview



Digest Algorithms
Quick Overview

- **Objective**: creating a fixed-length digital thumbprint of a message.
- **Examples**:
  - MD5 (Message Digest Algorithm 5):
    - Produces a 128 bits hash value
    - In 2004, flaws were discovered and its usage was discouraged.
  - SHA (Secure Hash Algorithm):
    - Ranges from 160 bits to 512 bits, depending on the chosen flavor.
    - One of the most secure algorithms.
  - CRC-32 (Cyclic Redundancy Check):
    - Used for data integrity in networks
    - Not suitable for security (very poor randomness).
  - Parity bit:
    - One bit long digest of a message.
    - Believe it or not, this is also a digest! (very poor one though...)

More computing power → more insecure/obsolete a digest becomes

Symmetric Encryption
Quick Overview

- **Objective**: Encrypting large chunks of data using a secret key.
- **Examples**:
  - AES (Advanced Encryption Standard):
    - 128, 192 and 256 bits key size block cipher.
    - Adopted by the U.S. government as an standard.
    - Also known as Rijndael.
  - Triple DES (Triple Data Encryption Standard):
    - 168 bits key length (192 if we count parity bits).
    - Created as an improvement to DES (that hat only 56 bits keys)
  - RC4 (ARCFOUR):
    - Widely used stream cipher.
    - Used in SSL and WEP
    - Created by RSA Security.

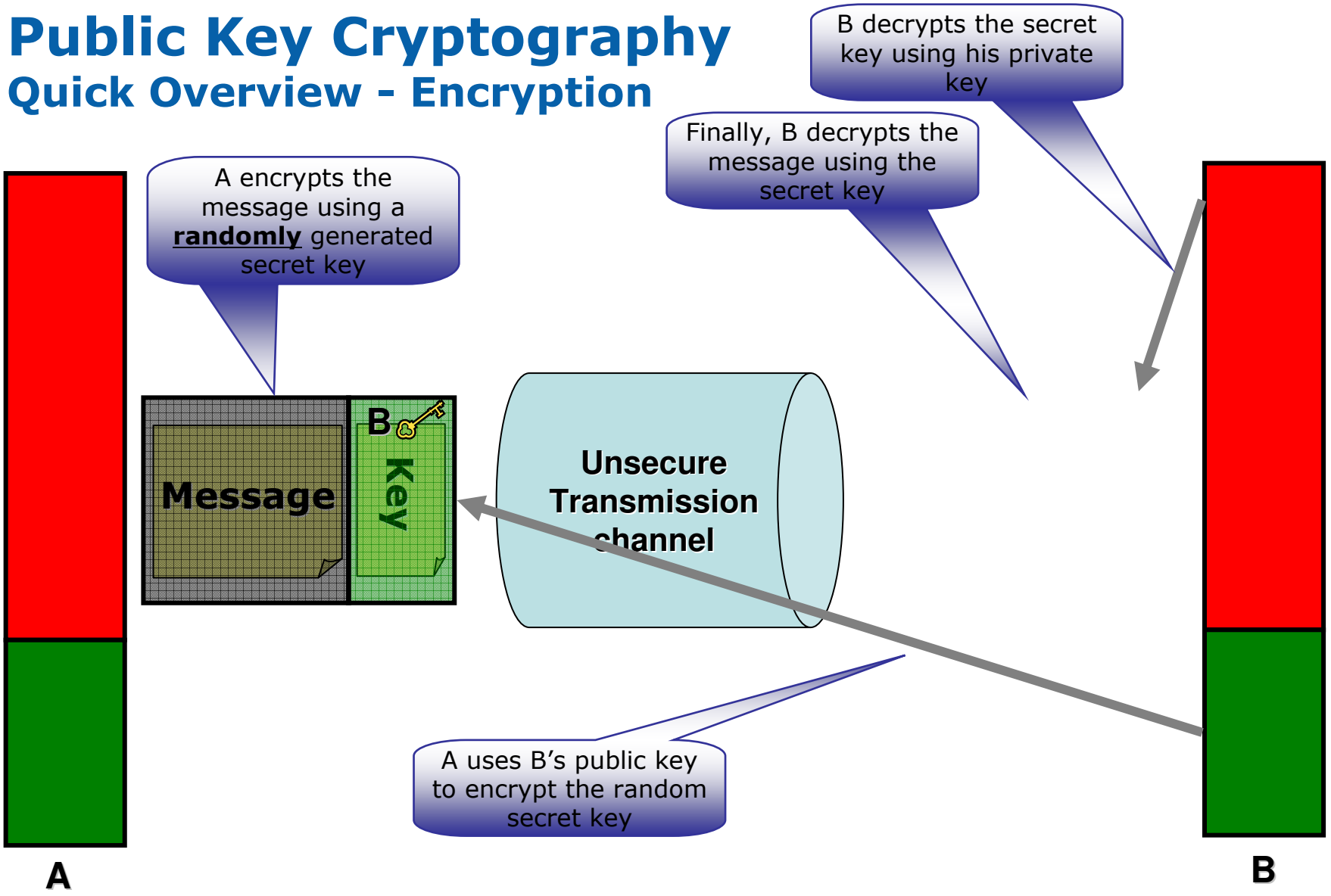
How do we share the secret key?

Public Key Cryptography
Quick Overview

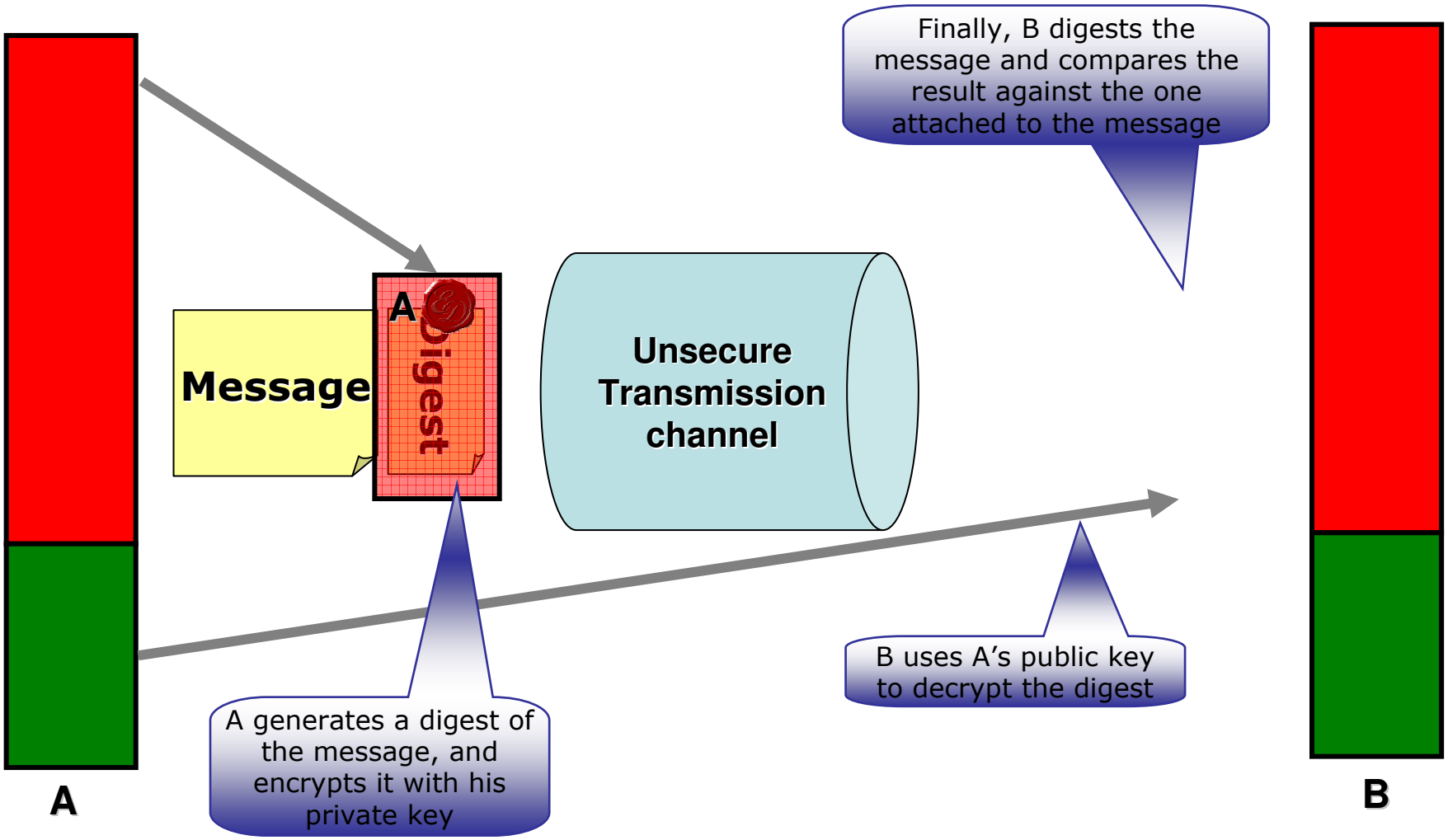
- Each party has it's own Public/Private key pair.
- RSA is one of the most known Pub Key cryptography algorithms
  - With keys ranging from 1024 to 4096 bits.
- Key properties:
  - What gets encrypted with the Public key can only be decrypted with the Private key of the same pair, and vice versa.
  - Can be used both for data encryption and for non-repudiation (signature)
- Limitations:
  - Slower than symmetric encryption algorithms such as AES or 3DES.
  - Needs longer keys to achieve same cryptographic strength as symmetric algorithms.
  - Can only encrypt a small amount of data.

Combining Public Key cryptography with Symmetric encryption solves the limitations.

Public Key Cryptography
Quick Overview - Encryption



Public Key Cryptography
Quick Overview - Signature



Backup

Canonicalization (c14n) Algorithms
Quick Overview

- **Objective**: creating a normalized representation of an XML document.
- **Examples**: Exclusive and Inclusive c14n.
- **Problem**:
  - Two XML documents might look different but have the same meaning:

```
<A      >Hello world</A>
```

Has the same "meaning" than:

```
<A>Hello world</A>
```

- **Solution**:
  - Before applying a digest algorithm, we need to transform this documents into a normalized representation.
  - Canonicalization solves this issue.

One of the most expensive XML-related operations

Simple Object Access Protocol (SOAP)

- SOAP is a protocol for exchanging XML-based messages over computer networks
  - Normally using HTTP
- SOAP forms the foundation layer of the Web Services (WS) stack
- Inside a SOAP message
  - Envelope, Header & Body

```
1  <soap:Envelope xmlns:soap="http://..." xmlns:wsu="http://...">
2      <soap:Body>
3          <!-- The goal is to sign only the candidates that will actually be promoted -->
4          <promotion_candidates>
5              <manager wsu:Id="WWID46464">John</manager>
6              <engineer wsu:Id="WWID34396">Mike</engineer>
7              <manager wsu:Id="WWID34364">Joe</manager>
8              <engineer wsu:Id="WWID32896">Tom</engineer>
9          </promotion_candidates>
10     </soap:Body>
11 </soap:Envelope>
```

WS-Security Anatomy

Secured SOAP Message

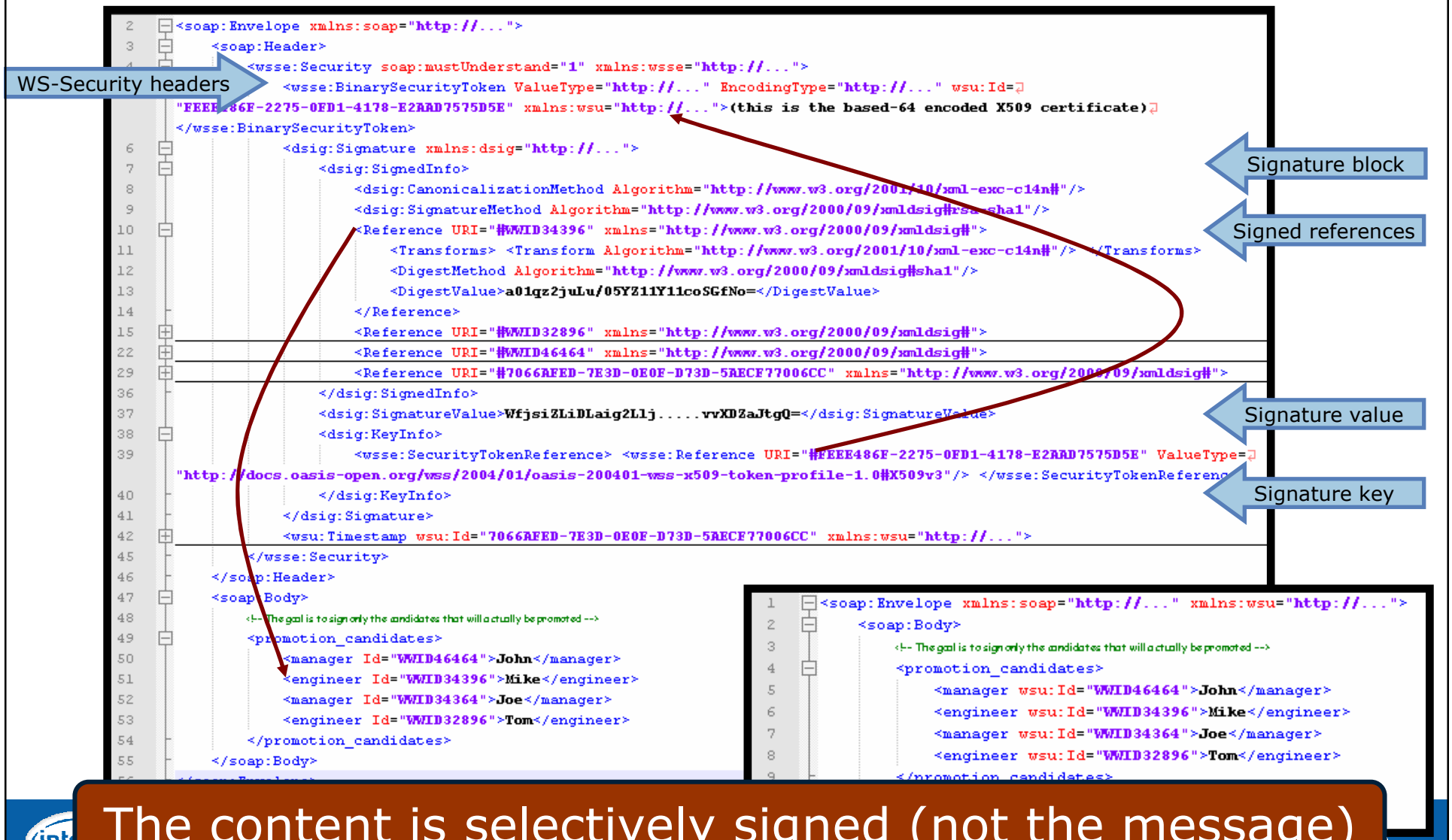
```
<soap:Envelope>
    <soap:Header>
        <wsse:Security>
            <Signature>
            </Signature>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <A>
        </A>
        <B>
        </B>
    </soap:Body>
</soap:Envelope>
```





Security Feature	Function
SOAP Header	
WS-Security	•Attaches signature, encryption, security tokens to SOAP messages
SAML Token	•Authenticates initiator of SOAP request. •Enables role based authorization. •Time-limited. •Interoperable.
X.509 Certificate	•Encryption and signature verification.
XML Signature, DSIG	•Multiple signed areas of header and body. •Integrity protection via PKI based cryptography. •Prevents tampering.
SOAP Body	
XML Encryption	•Multiple encrypted areas of body. •Prevents disclosure.

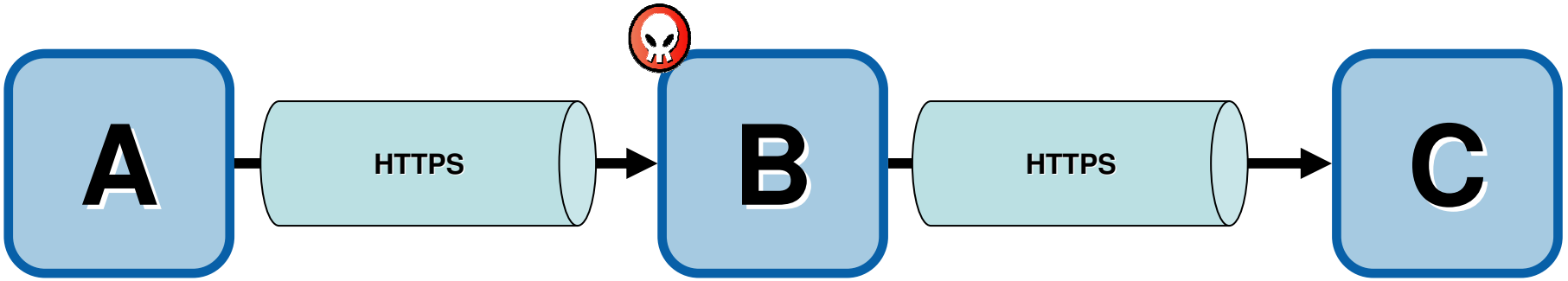
A signed SOAP message



The content is selectively signed (not the message)

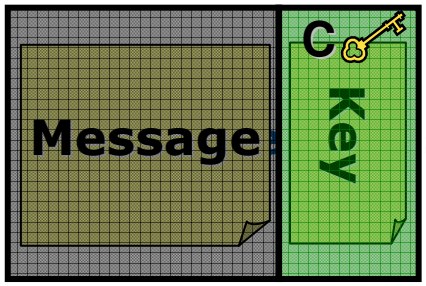
Transport vs. Message level security

- What if A wants to send a message to C, which needs to be routed through B?
- Current approach: HTTPS / SSL

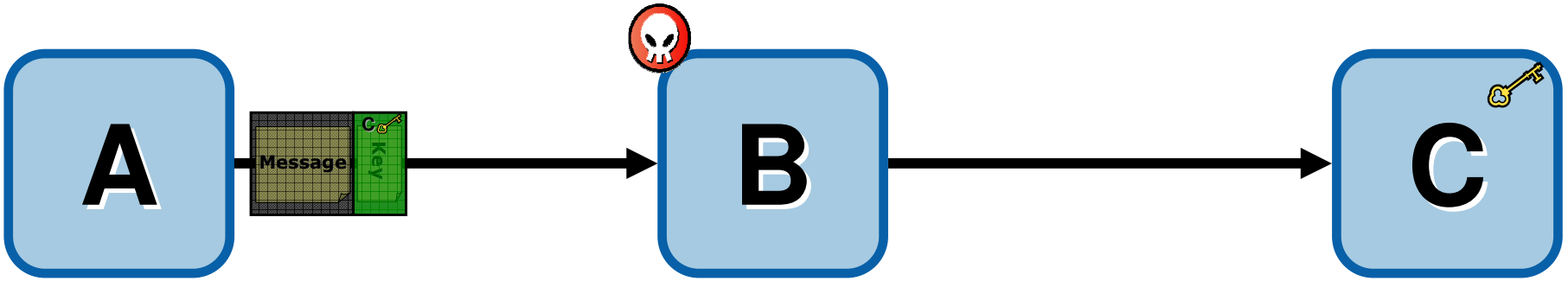


- If B is not trusted (or gets hacked), then HTTPS is not enough.

Transport vs. Message level security



security solves the problem.

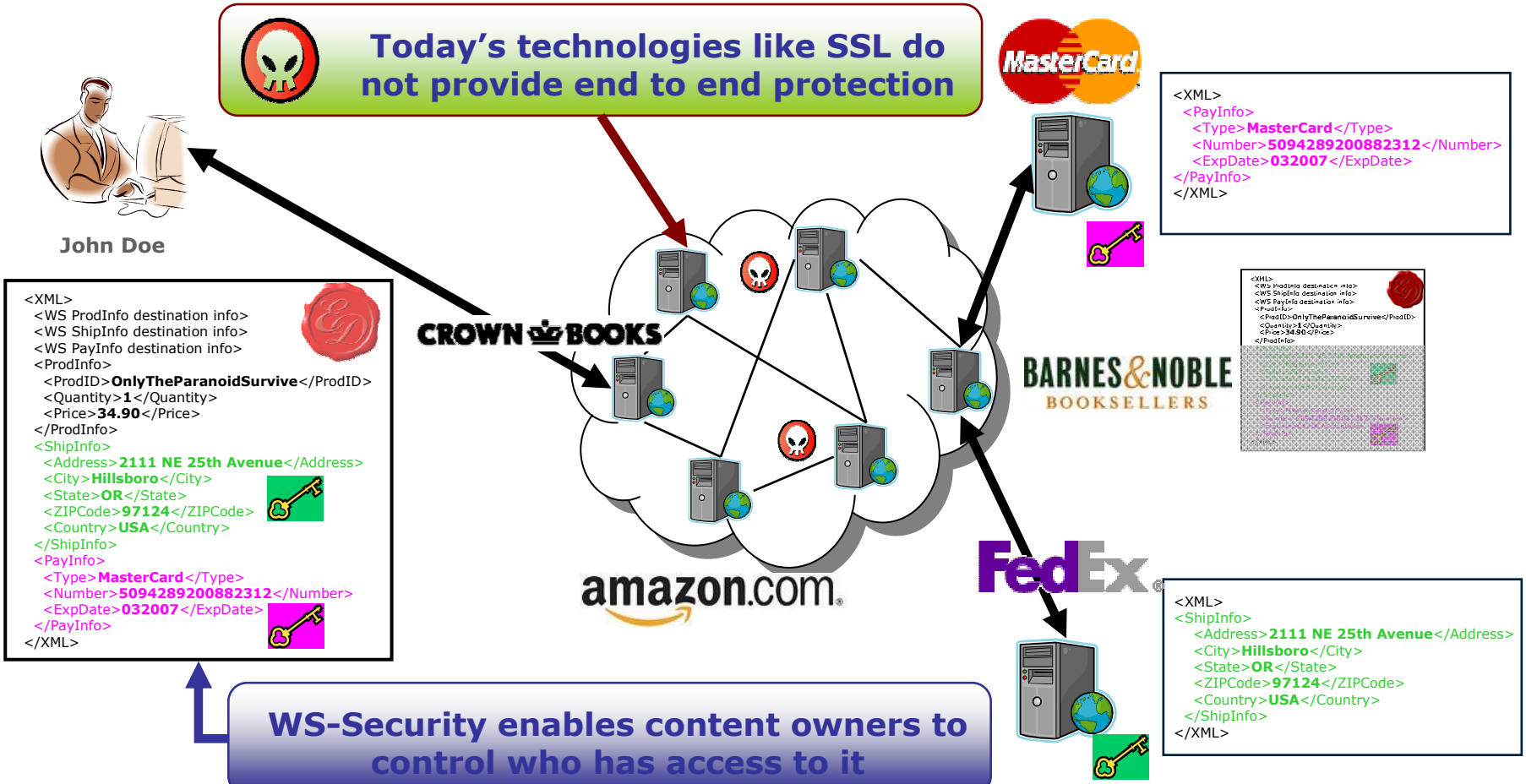


- Although B is not trusted or has been compromised, message level security ensures that only C will be able to decrypt the message.
- Signature is also needed to prevent tampering.

WS-Security in action

Anatomy of the SOA Security challenge

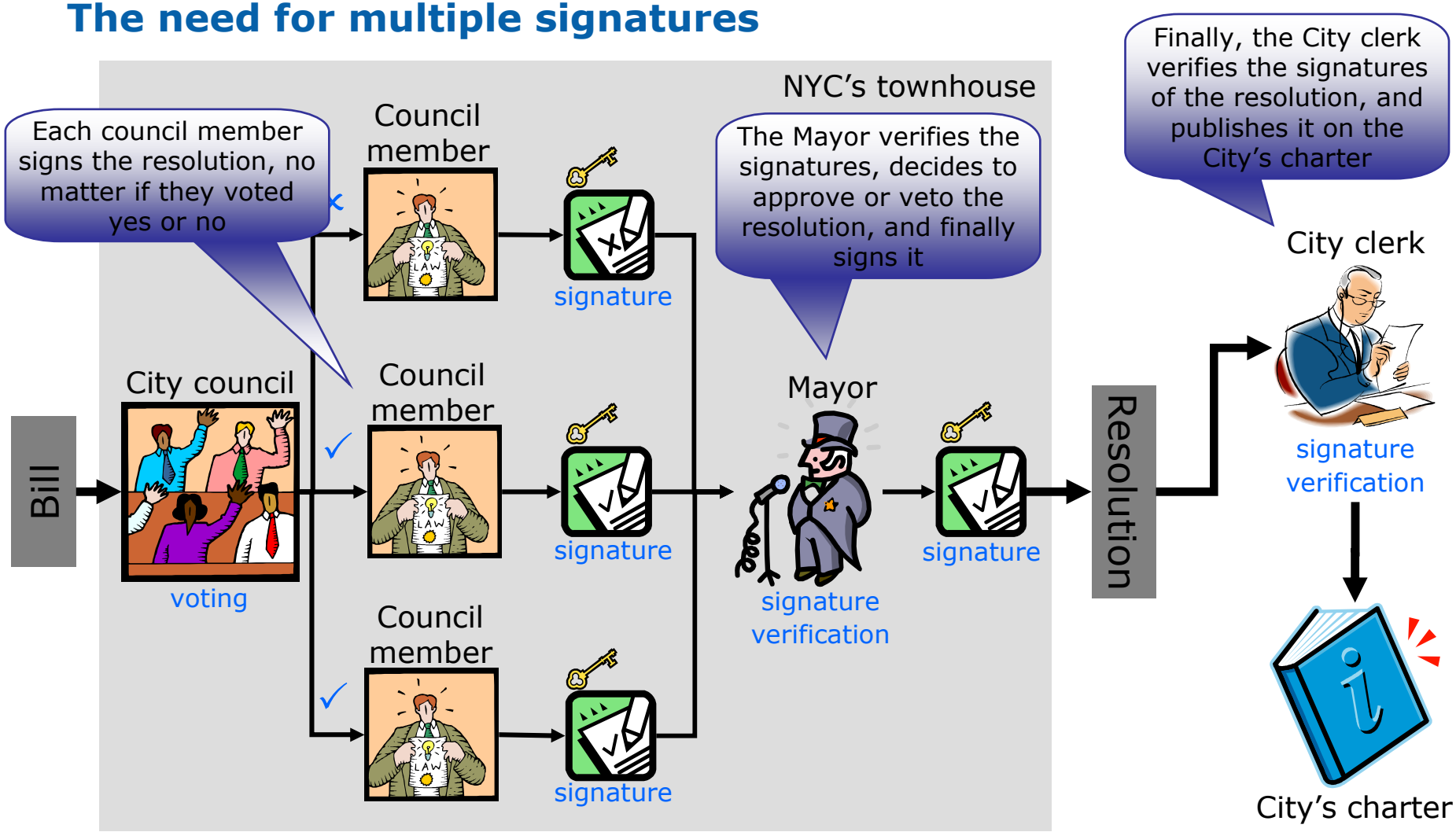
The need for end to end security



Content based security is the only solution for securing enterprise integration

Anatomy of the SOA Security challenge

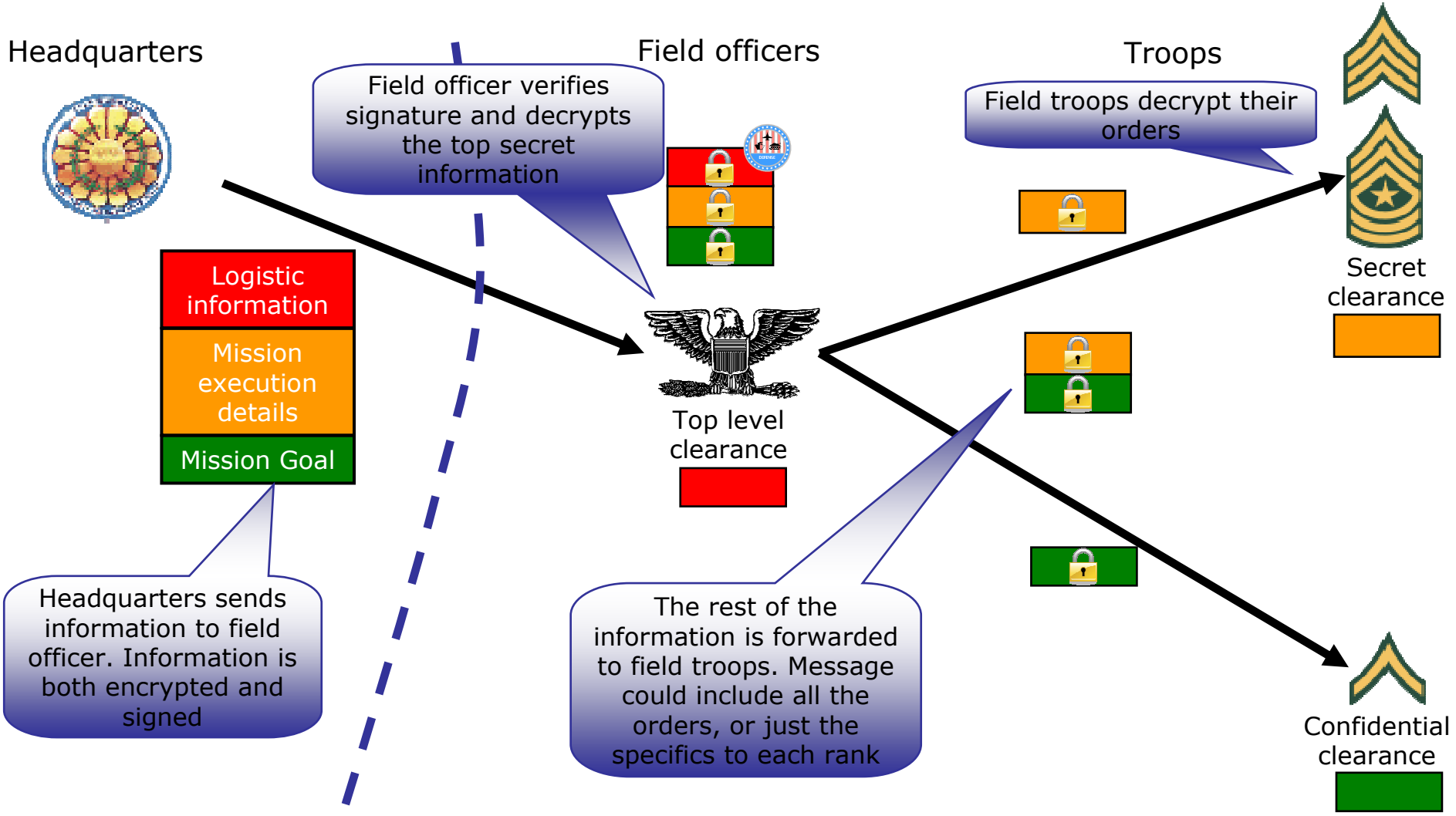
The need for multiple signatures



Content based security allows distributed transactions to be executed across vendors solutions

Anatomy of the SOA Security challenge

The need for multiple levels of clearance



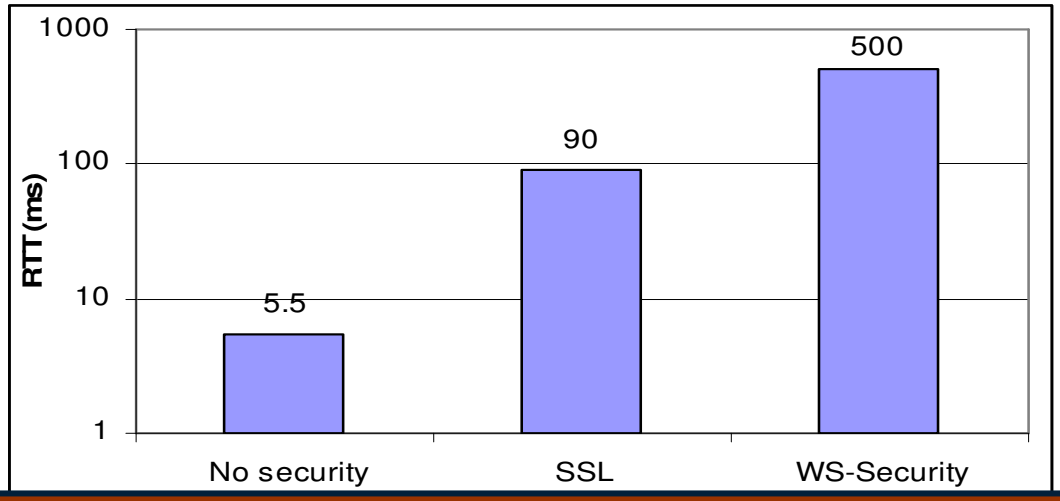
So what is new in all of this?

- The solution for all theses problems are well known
  - All this can be done with standard cryptography
- But….
  - Security is tricky: one mistake and it's over
  - Custom solutions rarely help systems integration
- WS-Security is
  - An OPEN STANDARD
    - It is the work of lots of smart individuals
  - It is implemented by several vendors (IBM, MS, Oracle, BEA, etc.)
    - It is easy to provide security across systems
  - There are open source implementations

Performance Considerations (i.e. there is no free lunch)

How expensive is all this?
SSL vs. WS-Security in Grid Computing

- The experiment (by Shirasuna et.al., 2004)
  - Goal: compare SSL & WS-Security for message integrity
    - 8 clients saturate a server with small messages (5 bytes payload)
  - Environment
    - XSUL using Apache XML Security library (XSUL is faster than GT3.2)
    - Tomcat 4.1.30. Sun J2SE 1.4.2_04, Linux 2.4.21
    - Dual Xeon 2.8GHz with 2GB of RAM



SSL adds a 10X slowdown, WS-Security adds 100X!
(most of this cost is XML processing)

What is the culprit?

- Let's do some back of the envelope calculations

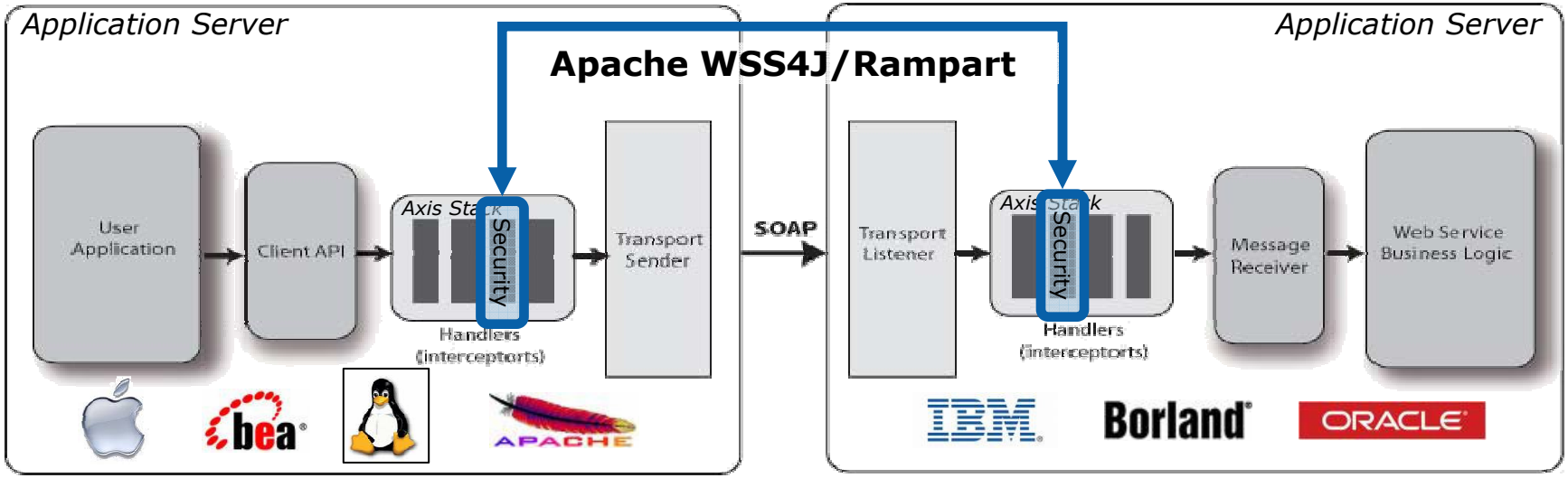
	WS-Security (enc.only)	HTTPS
RSA (No. operations)	6	6
DES (% of content processed)	150%	300%
XML overhead (% of content processed)	150%	0
No. SSL Negotiations	0	6

  - SSL requires more crypto than WS-Security!!
- About the XML overhead (Liu et.al., 2005)
  - It takes 10 ms to sign or encrypt 100KB
  - Using WS-Security takes 100-200ms to do the same
  - Environment
    - Sun's J2SE 1.4.2 with Bouncy Castle (JCE) & Apache's WSS4J
    - Linux 2.4.10
    - Pentium 4 CPU 2.79 GHz with 768MB of RAM

WS-Security using Open Source Solutions

WS-Security using Open Source

Apache Axis2 architecture



- Axis is a communication & content processing stack developed by Apache
- Axis is used internally by many commercial products to create Web Services
  - Apple, BEA, Borland, IBM, Adobe, Oracle, Globus Toolkit, JBoss, JOnAS
- Axis follows a policy based architecture
  - Developers specify WHAT they want to call and its parameters
  - Sysadmins specify the HOW
- Security can be added as module inside the Axis stack
  - E.g. Apache Rampart & WSS4J

Communication, Persistence, Management, Security, Transactions, Clustering/Scalability comes for FREE!!

References

- XML-Signature Syntax and Processing
- W3C Recommendation 12 February 2002
- http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/
- XML Encryption Syntax and Processing
- W3C Recommendation 10 December 2002
- http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/
- Exclusive XML Canonicalization Version 1.0
- W3C Recommendation 18 July 2002
- http://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718/
- Other References
- AES
- NIST FIPS 197: Advanced Encryption Standard (AES). November 2001.
- http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
- AES-WRAP
- RFC3394: Advanced Encryption Standard (AES) Key Wrap Algorithm. J. Schaad and R. Housley. Informational, September 2002.
- DES
- NIST FIPS 46-3: Data Encryption Standard (DES). October 1999.
- http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
- HMAC
- RFC 2104. *HMAC: Keyed-Hashing for Message Authentication.* H. Krawczyk, M. Bellare, R. Canetti. February 1997. http://www.ietf.org/rfc/rfc2104.txt
- HTTP
- RFC 2616. *Hypertext Transfer Protocol -- HTTP/1.1*. J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. June 1999. http://www.ietf.org/rfc/rfc2616.txt
- KEYWORDS
- RFC 2119. *Key words for use in RFCs to Indicate Requirement Levels.* S. Bradner. March 1997. http://www.ietf.org/rfc/rfc2119.txt
- MD5
- RFC 1321. *The MD5 Message-Digest Algorithm.* R. Rivest. April 1992. http://www.ietf.org/rfc/rfc1321.txt
- MIME
- RFC 2045. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. N. Freed & N. Borenstein. November 1996. http://www.ietf.org/rfc/rfc2045.txt
- PKCS1
- RFC 2437. *PKCS #1: RSA Cryptography Specifications Version 2.0.* B. Kaliski, J. Staddon. October 1998. http://www.ietf.org/rfc/rfc2437.txt

Open Standards

References

- SHA-1
- FIPS PUB 180-1. *Secure Hash Standard.* U.S. Department of Commerce/National Institute of Standards and Technology. http://csrc.nist.gov/publications/fips/fips180-1/fip180-1.txt
- SOAP
- Simple Object Access Protocol (SOAP) Version 1.1. W3C Note. D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Frystyk Nielsen, S. Thatte, D. Winer. May 2001.
- http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
- SOAP Schema
- http://schemas.xmlsoap.org/ws/2002/07/secext
- Unicode
- The Unicode Consortium. *The Unicode Standard.* http://www.unicode.org/unicode/standard/standard.html
- UTF-8
- RFC 2279. *UTF-8, a transformation format of ISO 10646*. F. Yergeau. January 1998. http://www.ietf.org/rfc/rfc2279.txt
- URI
- RFC 2396. *Uniform Resource Identifiers (URI): Generic Syntax.* T. Berners-Lee, R. Fielding, L. Masinter. August 1998. http://www.ietf.org/rfc/rfc2396.txt
- URL
- RFC 1738. *Uniform Resource Locators (URL).* T. Berners-Lee, L. Masinter, and M. McCahill. December 1994.
- http://www.ietf.org/rfc/rfc1738.txt
- XML
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation. T. Bray, E. Maler, J. Paoli, C. M. Sperberg-McQueen. October 2000.
- http://www.w3.org/TR/2000/REC-xml-20001006
- XML-C14N

Open Standards

Next talk

Tecnologías Emergentes para Aplicaciones Empresariales

- 13/9. *Introducción al Mercado SOA, Evolución y Tendencia*, con Mariano Cilia.
- *SCA y Tuscany*, con Mario Antollini.
- 20/9. *BPEL, un Enfoque a SOA Basado en Procesos*, con Alejandro Houspanossian.
- 27/9. *Web 2.0: Elementos y Tendencias*, con Sebastián Salvucci.
- 04/10. *Seguridad en SOA (WS-Security)*, con Marcelo Da Cruz Pinto.
- **11/10. *Sistemas Peer-to-Peer (P2P)*, con Cristian Fiorentino.**
- 18/10. *High-Performance Computing (HPC)*, con Silvana D' Cristofaro y Andrés More.
- 25/10. *Virtualización*, con Diego Palmisano.
- 1/11. *Administración y monitoreo de recursos de SW y HW*, con Sebastián Ganame.
- *Cierre del ciclo de charlas*, con Mariano Cilia, de 20.30 a 21.

Summary of the talk

- SOA has changed the way we think about software
- Business integration is now possible
- We need to address security in order to keep the momentum
  - Eventually we will have large & agile B2B systems
- WS-Security is open standard which is ready for the challenge
- There are plenty of open software stacks to build SOA

Q & A

asdc

Argentina Software
Development Center

Why is c14n so demanding?
Rules for canonicalization

1. The document is encoded in UTF-8
2. Line breaks normalized to #xA on input, before parsing
3. **Attribute values are normalized, as if by a validating processor**
   - This means that special characters inside an attribute value are replaces by they corresponding character reference
4. **Character and parsed entity references are replaced**
   - This means that character references such as #xD are replaces by their real value.
5. CDATA sections are replaced with their character content
6. The XML declaration and document type declaration (DTD) are removed
7. **Empty elements are converted to start-end tag pairs**
   - An empty element is one that does not contain any text or sub-elements, and is generally denoted by **<element/>**
8. Whitespace outside of the document element and within start and end tags is normalized
9. All whitespace in character content is retained (excluding characters removed during line feed normalization)
10. Attribute value delimiters are set to quotation marks (double quotes)
11. Special characters in attribute values and character content are replaced by character references
12. **Superfluous namespace declarations are removed from each element**
    - If a node contains a namespace that is already present in an ancestor element, then the namespace node will be removed.
13. **Default attributes are added to each element**
    - This means that default attributes will be added to the c14n output (the ones that are defined on the DTD, if it is present)
14. Lexicographic order is imposed on the namespace declarations and attributes of each element

Canonicalization example

```
<!DOCTYPE doc [<!ATTLIST e9 attr CDATA "default">]>
<doc>
   <e1   />
   <e2   ></e2>
   <e3   name = "elem3"   id="elem3"   />
   <e4   name="elem4"   id="elem4"   ></e4>
   <e5 a:attr="out" b:attr="sorted" attr2="all" attr="I'm"
       xmlns:b="http://www.ietf.org"
       xmlns:a="http://www.w3.org"
       xmlns="http://example.org"/>
   <e6 xmlns="" xmlns:a="http://www.w3.org">
      <e7 xmlns="http://www.ietf.org">
         <e8 xmlns="" xmlns:a="http://www.w3.org">
            <e9 xmlns="" xmlns:a="http://www.ietf.org"/>
         </e8>
      </e7>
   </e6>
</doc>
```

- **Demonstrates:**

Retention of namespace prefixes from original document
Empty element conversion to start-end tag pair
Normalization of whitespace in start and end tags
Relative order of namespace and attribute axes
Lexicographic ordering of namespace and attribute axes
Elimination of superfluous namespace declarations
Addition of default attribute

Original XML

Canonicalized XML

```
<doc>
   <e1></e1>
   <e2></e2>
   <e3 id="elem3" name="elem3"></e3>
   <e4 id="elem4" name="elem4"></e4>
   <e5 xmlns="http://example.org" xmlns:a="http://www.w3.org" xmlns:b="http://www.ietf.org" attr="I'm" attr2="all" b:attr="sorted" a:attr="out"></e5>
   <e6 xmlns:a="http://www.w3.org">
      <e7 xmlns="http://www.ietf.org">
         <e8 xmlns="">
            <e9 xmlns:a="http://www.ietf.org" attr="default"></e9>
         </e8>
      </e7>
   </e6>
</doc>
```

Step One: Open Standards

WS-Security

SAML 2.0

WS-Trust

WS-SecureConversation

XML Encryption

XML DSig

SOAP 1.2

WS-ReliableMessaging

XPATH 2.0

WS-SecurityPolicy

WSDL

XML Schema

XML

UDDI

Open standards help Open Source compete on a level plain field