# A Web Services Security Framework

Jorgen Thelin
Chief Scientist
Cape Clear Software Inc.



CAPE CLEAR
www.capeclear.com

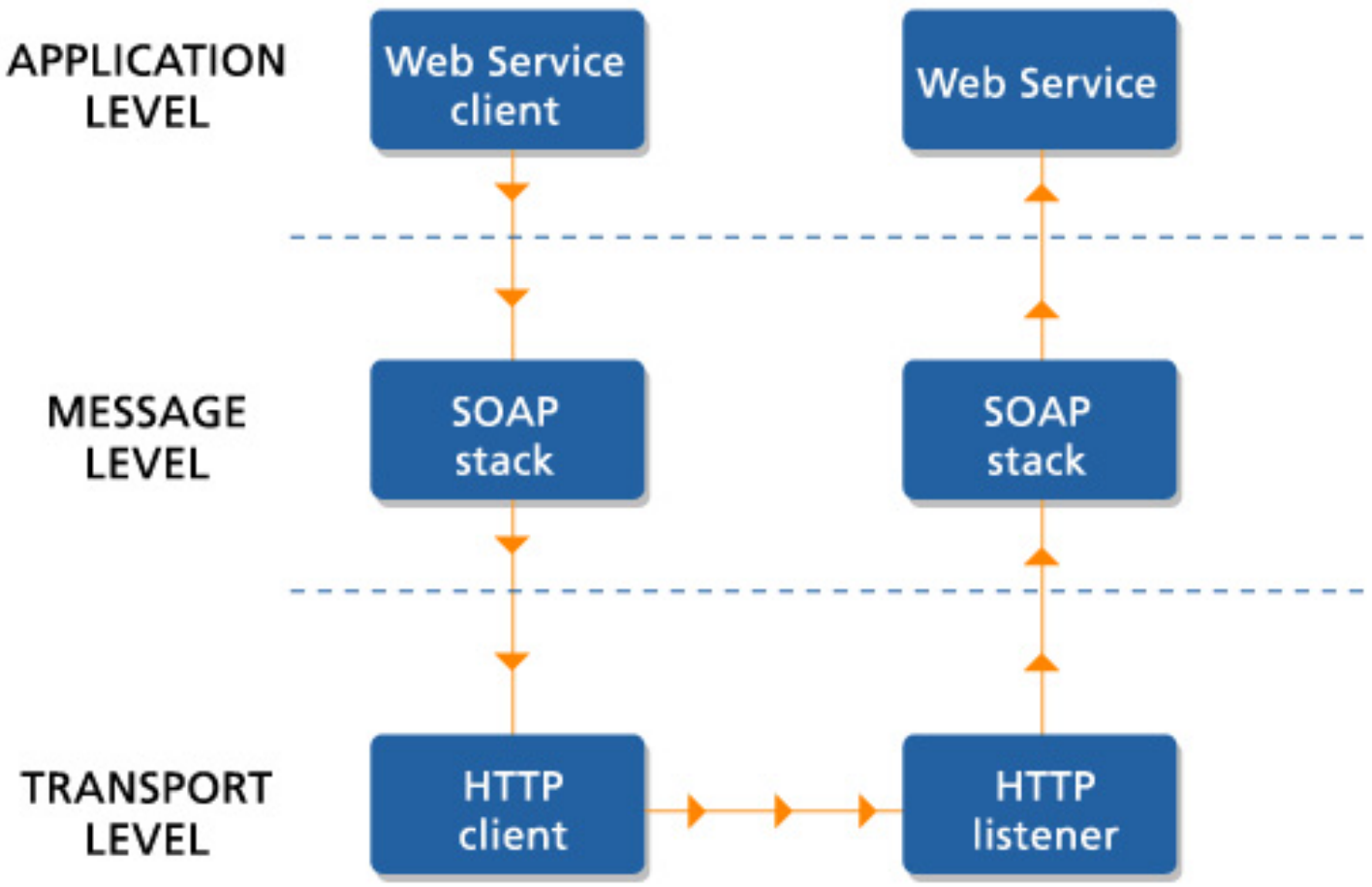
# 4 Main Concerns of a Security Framework

- Authentication – identity
  - Who is the caller?
  - How do we prove they are who they say they are?
- Authorization – access control
  - What is the caller authorized to do?
  - Is the caller permitted by perform the operation it is requesting?
- Confidentiality – encryption
  - How do we prevent snoopers viewing our messages and data?
- Integrity – tamper-proofing
  - How do we prevent messages being tampered with between sender and receiver?

# Non-Repudiation

- This is ultimately the major business requirement for a security framework
  - Can a trading partner possibly claim that:
    - They didn't send a message
    - They sent a different message from the one you received
  - Requires framework support for:
    - **Authentication** – we know who sent the message
    - **Integrity** – the message did not change in transit
    - **Audit record storage** – we can prove what happened

# Web Service Interaction Levels



# Transport Level Security

- Uses existing Web tier technology such as HTTP and SSL

- **Authentication**
  - HTTP authentication schemes – Basic or Digest
  - SSL client side certificates

- **Authorization**
  - URL access control policies in the web tier
  - J2EE Servlet declarative security constraints

- **Confidentiality**
  - SSL encrypted connections

- **Integrity**
  - Point-to-point SSL encryption to avoid data interception

# Message level security

- Security data built in to the XML message text – usually as additional SOAP header fields

- **Authentication**
  - SSO (single sign-on) header tokens
  - SAML authentication assertions

- **Authorization**
  - SSO session details
  - SAML attribute assertions

- **Confidentiality**
  - XML Encryption specification

- **Integrity**
  - XML Digital Signatures specification

# Application level security

- A Web Service application handles its own security scheme – for example, UDDI

- **Authentication**
  - App specific authentication messages
  - App specific credential headers in other messages
  - App maintains its own security domain

- **Authorization**
  - App performs its own access control checks

- **Confidentially**
  - App can apply an encryption scheme to some or all data fields

- **Integrity**
  - XML Digital Signature can be used for tamper detection
  - App specific integrity data such as MD5 hash can be sent for some or all data fields

# Conclusions – Key Issues

- A Web Services security framework must support existing security products
- Must be an end-to-end framework to avoid any security gaps
- New XML security specifications are not yet stabilized or proven
- Use existing proven Web tier security infrastructure until XML security specifications and infrastructure is validated
- WS-I Basic Security Profile will deliver a standardized XML security infrastructure over time