A Public Web Services Security Framework Based on Current and Future Usage Scenarios

Internet Computing 2002 Conference, Las Vegas, June 2002

J.Thelin, Chief Architect
PJ.Murray, Product Manager

Cape Clear Software Inc.

Web Services Usage Scenarios

- Point-to-point system integration
- Enterprise application integration
- Technology integration
- Business partner collaboration
- Composite business processes
- Reducing I.T. lifecycle costs
- I.T. investment protection

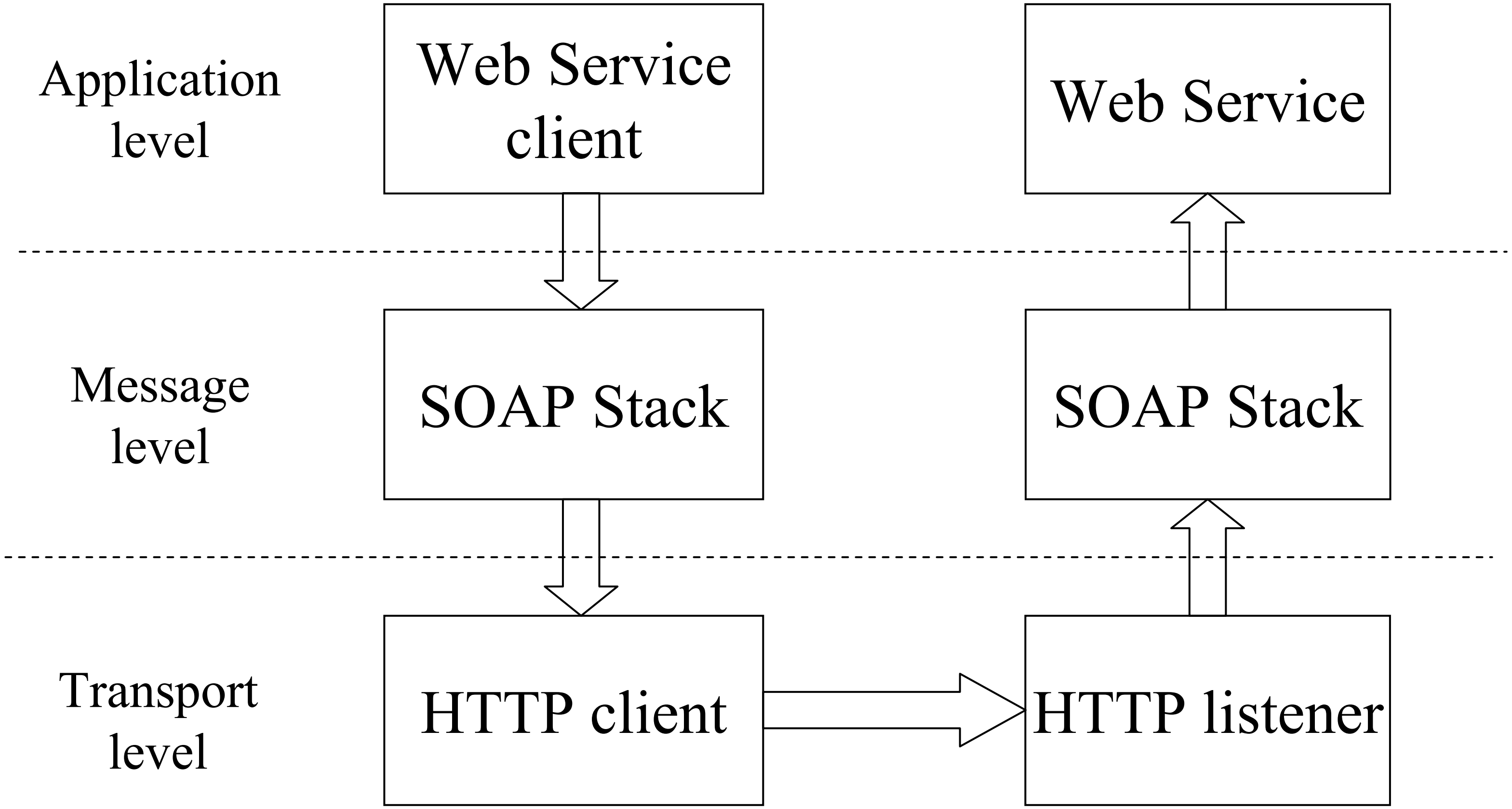
3 Main Concerns of a Security Framework

- 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 and tamper-proofing
  - How do we prevent snoopers viewing our messages and data?
  - How do we prevent messages being tampered with between sender and receiver?

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
  - J2EE Servlet declarative security constraints
- Confidentiality
  - SSL encrypted connections

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
  - 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
  - XML Digital Signature specification for tamper detection

Lessons from the First Wave

- Existing Web tier security infrastructure usually sufficient for internal projects
- Necessary to accommodate third-party security products already in use in the organization
- End-to-end framework is necessary to avoid security gaps
- Operational security procedure best practices for Web services have yet to be developed
- XML security standards have not yet been widely adopted
- Rival XML security standards are still emerging
- Lack of experience and training on XML security standards is holding back adoption

Recommendations for the future

- Track usage scenarios in an organization to determine security levels
- Start with “proof-of-concept” projects to gain experience
- Integration with Microsoft .NET security schemes will be vital
- Track emerging XML security specifications
- Don’t throw away the organization’s existing security infrastructure
- Plan to implement end-to-end security

Conclusions – Key Issues

- A Web Services security framework must support existing security products
- Must be an end-to-end framework (not just a “firewall” layer) to avoid any security gaps
- New XML security standards are not yet proven (so probably contain “holes”)
- Use existing proven Web tier security infrastructure until XML security standards and infrastructure is validated

Resources

- CapeScience
  - Papers, articles, tutorials, and webcasts for Web Services developers
  - http://www.capescience.com
- Cape Clear Academic Licenses
  - Free licenses for Cape Clear products to academic users
  - http://www.capescience.com/academic/