LIYANA ARACHCHIGE RANIL

SHORT NOTES / INTEGRATION AND MESSAGING

1. INTEGRATION and MESSAGING is related to HOW to SEND data to and receive from ANOTHER SYSTEM or APPLICATION
2. A WEB SERVICE is a piece of software designed to allow SYSTEM-to-SYSTEM communication over a network
3. Vast majority of WEB SERVICES today COMMUNICATE using SOAP or XML over HTTP or HTTPS
4. More general concept of WEB SERVICE according to W3C allow for MULTIPLE TRANSPORTS (something like HTTP/HTTPS) and DATA PROTOCOLES (something like SOAP etc)
5. Web services address SECURITY , RELIABILITY and TRANSACTION through WS-Security , WS-ReliableMessaging , WS-Coordination and WX-AtomicTransactions specifications
6. WS-Security – Web service security is needed due to various reasons
    a. Over HTTP one can AUTHENTICATE the caller , SIGN the MESSAGE , and ENCRYPT the CONTENT of a MESSAGE
    b. For those who need to solve BIGGER problems HTTP based security is not enough
    c. Many of the BIGGER PROBLEMS involve sending the message ALONG a PATH more COMPLICATED that REQUEST/RESPONSE
    d. Or over a transport which DOES NOT INVOLVE HTTPS
    e. More than ONE encryption key MAY BE used ALONG the route
    f. HTTP and its SECURITY mechanism ONLY ADDRESS POINT-TO-POINT security. More complex solutions need END-To-END security

g. WS-Security addresses how to MAINTAIN a SECURE CONTEXT over a MULTI POINT message path

h. WS-Security addresses SECURITY by LEAVERAGING existing standards and specifications

i. Kerberos and X.509 addresses AUTHENTICATION, XML Signature and XML Encryption describes ways of encrypting and signing the content of XML messages, XML canonicalization describes ways of making XML ready to be SIGNED and ENCRYPTED. What WS-Security adds to existing specifications is a FRAMEWORK to embed these MECHANISMS into SOAP messages

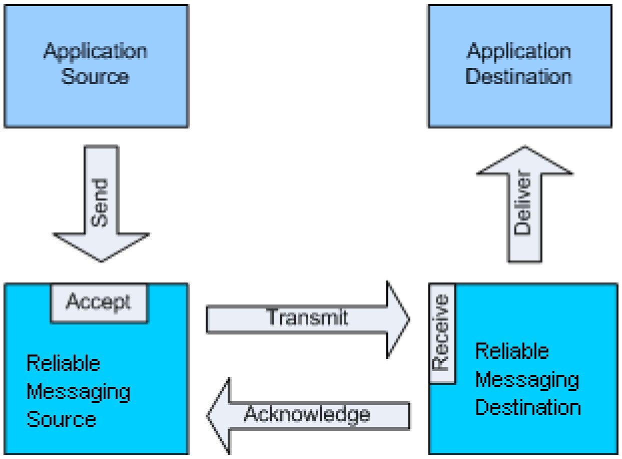
j. XML SIGNATURE is fairly complicated than normal digital signatures due to the nature of XML encoding. There can be whitespaces , different line ending characters , but the XML message may be the same. But if there is a single whitespace difference then the XML Signature validation would fail

k. To avoid such situations , XML CANONICALIZATION is introduced , this would format XML document to make it ready for SIGNING

l. WX-Security defines a SOAP header element to CARRY SECURITY related DATA

7. WS-ReliableMessaging – This describes a protocol that allows SOAP messages to be RELIABLY DELIVERED between DISTRIBUTED applications in the presence of SOFTWARE COMPONENT , SYSTEM or NETWORK FAILURES

   a. An Application Source (AS) wishes to RELIABLY send messages to an Application Destination (AD) over an UNRELIABLE infrastructure. To accomplish this they make use if RELIABLE MESSAGING SOURCE (RMS) and a RELIABLE MESSAGING DESTINATION(RMD)



b.

c. RMS uses the WS-ReliableMessagin protocol to TRANSMIT the message to the RMD. The RMD delivers the message to the AD. If the RMS can not transmit the message to the RMD for some reason it MUST raise an EXCEPTION or otherwise indicate to the AS tat the MESSAGE was NOT TRANSMITTED

d. The AS and the RMS may be implemented within the same process space or they may me separate components.

e. Similarly AD and EMD may exist within the same process space or they may be separate components

f. WS-ReliableMessaging ONLY deals with the CONTENTS and BEHAVIOUR of MESSAGES as they appear “ON THE WIRE”. How messages are SENT from AS to RMS , how they are DELIVERED from the RMS to AD, whether messages are PERSISTED on-disk or held in memory are non of the parts of WS-RM spec

g. WS-ReliableMessaging – DELIVERY ASSURANCE

        i. **AtLeastOne** [each message must be delivered to AD at least once, a message may be delivered more than once-duplicates]
        ii. **AtMostOne** [Each message will be delivered to the AD at most once. Message may not be delivered to the AD, but AD NEVER gets DUPLICATES]
        iii. **ExactlyOnce** [Each message will be delivered to AD exactly once. AD never gets DUPLICATES]
        iv. **InOrder**[Message will be delivered from RMD to AD in the ORDER that they are SENT from the AS to the RMS, this assurance can be COMINED with any of the ABOVE assurances] are the types of delivery assurance
    h. RESTFul web services DOES NOT HAVE reliable MESSAGING infrastructure
8. SOAP stands for SIMPLE OBJECT ACCESS PROTOCOLE
9. SOAP is an XML based extensible method of REPRESENTING DATA
10. WEB SERVICES can be written to CONSUME and EMIT SOAP based messages or XML messages
11. SOAP has benefit of adding additional information using ENVELOPES , ENCODING RULES and DATA REPRESENTATION CONVENTIONS over RAW XML
12. But SOAP has the drawback of being MORE COMPLEX and Introduces MORE OVERHEAD
    a. In terms of WIRE TRANSITION
    b. MESSAGE PARSING
13. WSDL stands for WEB SERVICE DESCIPTION LANGUAGE
14. WSDL defined the web service and provides information needed to access and consume the web service

15. JAX-RPC stands for Java API fro XML RPC
16. JAX-RPC was created quickly to add web service support to the J2EE platform
17. JAX-RPC is replaced by JAX-WS
18. JAX-WS is the primary API for the XML based web services both SOAP based and RESTFul
19. JAX-WS users lower level API(S) such as JAXB , SAAJ (SOAP with Attachments API for JAVA),JAXP , StAX
20. SOAP is needed when you design you WEB SERVICE not as a SERIES of MESSAE EXCHANGES , but as a WEB ACCESSIBLE API
21. JAXB stands for JAVA API FOR XML BINDING
22. JAXB helps in MARSHALLING and UNMARSHALLING between Java Objects and XML
23. JAXR stands for JAVA API for XML REGISTRIES
24. JMS is a NON web service integration element of JAVA EE platform
25. JMS is the CORE MESSAGING structure used in JEE to allow ASYNCHRONOUS JAVA to JAVA integration via QUEUE(s) and TOPIC(s)
26. If the architect can control both the MESSAGE produces and CONSUMERS , then the preferred mechanism would be JMS
27. JMS concept is simple , any JAVA class can create a message , LOOK UP via JNDI or have a REFERENCE to a JMS queue or topic INJECTED by the CONTAINER
28. In a JEE application MDB is the most LOGICAL way to CONSUME messages from a JMS provider
29. JCA stands for JAVA CONNECTOR ARCHITECTURE
30. JCA provides STANDARDIZED access mechanism to ENTERPRISE INFORMATION SOURCES (EIS) from JAVA EE platform

31. JCA is useful In Wrapping LEGACY application which still have a BUSINESS VALUE and EXPOSING their functionality as a WEL FORMED API for CONSUMPTION by JAVA EE application
32. Need for JCA varies widely by industry. WEB 2.0 companies building social network applications no need for JCA. But FINANCIAL companies with a SUBSTANTIAL investment in LEGACY SYSTEMS that still MEET BUSINESS REQUIREMENTS use JCA widely to EXPOSE these EIS(s) to JAVA EE application SERVER
33. INTEGRATION is the process by which INFORMATION is passed BETWEEN TWO or MORE DISTINCT software ENTITIES
34. Below question should be asked when selecting an integration mechanism
    a. Is the transfer ASYNCHRONOUS or SYNCHRONOUS
    b. Is the transfer of information ACKNOWLEDGED
    c. Is the transfer of information TRANSACTIONAL
    d. Does the transfer of information occur in BATCHES composed of multiple messages or ONE message at a time
    e. Does the transfer of messages occur systems BUILD using the SAME technology or DIFFERENT technology
    f. Does the transfer of information use a TECHNOLOGY specific MESSAGING/TRANSPORT protocol or a technology INDEPENDENT protocol
35. In general all the integration scenarios can be boiled down to few major things such as JAVA to JAVA integration , JAVA-to-NON-JAVA Integration
36. JAVA –to –JAVA integration
    a. Adopting a solution that INTRODUCES overhead in the form of a MORE VERBOSE data representation or that is JAVA-INDEPENDENT is only warranted if a NON-JAVA SYSTEM can

be EXPECTED to also INTEGRATE with the system currently under consideration

b. JAVA to JAVA systems integration is dealt with JMS precisely

37. JMS is intrinsically ASYNCHRONOUS.

38. SYNCHRONOUS semantics can also be emulated

39. But SYNCHRONOUS usage goes against the DESIGN INTENT of JMS

40. JMS supports

    a. Publish and subscriber , Point to Point MESSAGING MODELS

    b. Message DELIVERY ACKNOWLEDGEMENT

    c. Message level ENCRYPTION

    d. Distributed TRANSACTIONS via JTA using XAConectionFactory , XAConnection,XASession

41. JAVA – to – Non – JAVA integration

    a. When integrating JAVA systems to NON-JAVA systems which are not owned or controlled b y the architect, WEB SERVICES are needed

    b. JMS is not appropriate for this scenario

    c. There are two options available , WEB SERVICES with SOAP or RESTFul and JCA

42. WEB SERVICES – Java to Non Java Integration

    a. Web services were intrinsically designed to FACILITATE the INTEGRATION of HETEROGENEOUS systems, JMS was NOT

    b. JMS is OPTIMIZED for JAVA producers and consumers of messages

    c. As long as WSDL stay unchanged changes to one system DOES not EFFECT other SYSTEMS

    d. WEB SERVICES are TRULY TECHNOLOGY INDEPENDENT

43. Within WEB SERVICE one can user SOAP or SIMPLE XML as messaging protocol

44. Adhering SOAP allows the message to be SELF DESCRIBING
45. SOAP has HIGHER RUNTIME OVERHEAD to construct and parse SOAP messages , STEEPER LEARNING CURVE , MORE VERBOSE REPRESENTATION
46. JCA stands for JAVA CONNECTOR ARCITECTURE
47. JCA is used to provide STANDARDISED ACCESS to an EXISTING ENTERPRISE INFORMATION SYSTEM (EIS) from the JEE platform
48. These EIS are usually MAINFRAME systems
49. A COMPLETE JCA system would have
    a. EIS itself
    b. Resource ADAPTOR (RA) that implements CONNECTOR SPECIFICATION
    c. JAVA EE Application Server which users the RS to access EIS
50. JCA focuses on the NON-FUNCTIONAL aspects such as
    a. Full support fro TRANSACTION CONTEXT PROPAGATION to preserve ATOMICITY , CONSISTANCY , ISOLATION and DURABILITY
    b. SCALABILITY
    c. CORRECT RECOVERY FROM FAILED / INTERUPTED OPERATIONS