SOAP Web Services 2

Monday, February 8, 2021 1:54 PM

SOAP Web Services

1. Discovery and self-describing
2. Security (granular)
3. Protocol

Classification:

1. RPC
   - Specifying method name and parameter (since there are multiple methods)
   - Synchronous invocation (client invokes, gets results, then client resumes)
2. Document-centric
   - Customer submits purchase order without method name or parameter

Request for document-centric service

```
<?xml>
    <envelope>
        <body>
            <PurchaseOrder>
                <item></item>
                <payment></payment>
                <shipping></shipping>
            </PurchaseOrder>
        </body>
    </envelope>
</xml>
```

No method/parameter name.
Provider may keep services hidden.

Assume there is a purchase order service from Amazon. It will look into the registry and invoke an inventory service by specifying parameters and method name. It does this for every service in the registry. Amazon has knowledge of the parameters and method name. These services may be RPC or document-centric services. But Amazon is a document-centric web service for the client. Amazon asynchronously gives the invoice.

Both client -> Amazon & Amazon -> vendor are SOAP responses.

Message queue in asynchronous communication:
Client sends SOAP request for purchase order without waiting for result and continues to code. When the results come, Amazon will pass the result *asynchronously*. Amazon sends the invoice as a SOAP response.

How does the code run without response? Client has a message queue that is given a SOAP request and the rest of the code is executed.

Assume Amazon server is down, client passes request. Message queue keeps pinging the server for a response until it gets it.

Once Amazon is up, message queue dumps request, Amazon processes request through a sequence of web services. Amazon also has a message queue where it passes the SOAP response. Invoice received by the client.

Mobile communication is synchronous
WhatsApp messages are asynchronous
MSMQ: Microsoft message queue
JMS: Java message queue
MQ: IBM message queue

Cybersecurity in web services:

Cryptography:

1. **Confidentiality**:
   When information is transferred over the internet, it must be protected from intruders (only visible to authorized to limited users).
2. **Integrity**: unauthorized changes to data must be detected (using Java based APIs)
3. **Authenticity**: some services require authentication from the client. For instance, access to a subscribers-only service. *What are the multiple ways of authentication?*
4. **Non-repudiation**: digital signing to prevent clients from denying receiving a service and refusing payment.

Confidentiality through encryption:

1. **Symmetric-key cryptography**
   - Sender and receiver share a binary key. Bigger the key, longer it takes to break the code. Faster compared to other encryption methods.
2. **Asymmetric-key cryptography**
3. **Hybrid cryptography**

Ciphertext: encrypted data