《面向服务的体系结构》第八章作业

1. 比较数据库中事务与web服务事务概念的异同。

答：　数据库中的事务和web服务事务都具有以下特性：

- 原子性（Atomicity）：无论是数据库事务还是Web服务事务，都强调了操作的原子性。这意味着事务中的操作要么全部完成，要么全部不执行。
- 一致性（Consistency）：两者都确保在事务开始和结束时，系统保持一致的状态。
- 事务管理：它们都需要事务管理机制来保证事务的正确执行和在出现故障时的恢复。

它们的不同之点在于：

- 原子事务被广泛应用于数据库系统中，但原子事务在执行时要进行资源的锁定，因此不适用于某些时间跨度比较大的电子商务协作场景。因此，出现了基于补偿的事务模型。基于补偿的事务模型放宽了对原子事务的原子性和隔离性要求，允许事务中的操作改变系统的状态，而在事务失败时采取补偿机制，以便将系统恢复到有效状态。尽管基于补偿的事务和原子事务采用的策略有所不同，但同样可以保障系统的一致性。
- 同时，由于Web服务具有分布式、松耦合的特点，因此Web服务应保证事务的两个重要条件：
  - 需要同时支持原子事务和基于补偿的事务：因为在Web服务的应用场景中，既有资源可控的短期事务，也有时间跨度较长的长事务。
  - 需要同时支持集中式管理和分布式事务管理：分布式事务管理是指事务管理器本身是分布的，这是Web服务自主性所要求的。

2. Web服务的安全标准包括哪些内容？

答：W3C、OASIS、IBM和微软等一起制订了Web服务的安全标准。

1） 认证和权限

- 通过在SOAP消息头中包含认证和权限信息实现。
- SAML(Security Assertions Markup Language)和XACML(XML Access Control Markup Language)规范可以一起实现单点登录SSO(Single Sign-on)。

2） 机密性

- 通过XML-Encryption规范实现。

3） 完整性

- 通过XML-Signature规范实现----http://www.w3.org/Signature/2001/
- Web服务安全性不仅要求在传输层实现点到点(point-to-point)的安全，而且需要在消息层实现端到端(end-to-end)的安全。

3. WS-Security是Web服务安全的标准语言，请介绍在SOAP消息中添加用户名/密码信息，Kerberos安全证书，X.509数字签名，以及为SOAP消息体加密的方法。

答：

1） **添加用户名/密码信息：**

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
    xmlns:s="http://schemas.xmlsoap.org/soap/envelop"
    xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext">
    <s:Header>
    <wsse:Security>
        <!-- 用户名和密码 -->
        <wsse:UsernameToken>
            <wsse:Username>students</wsse:Username>
            <wsse:Password>software</wsse:Password>
        </wsse:UsernameToken>
     </wsse:Security>
     </s:Header>
   ...
 </s:Envelope>
```

可以看出，在Username元素中包含了用户名和密码，虽然在SOAP消息中没有对这些信息进行加密，但在传输层会通过SSL等安全协议对整个SOAP消息进行加密。

2） Kerberos安全证书：

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
    xmlns:s="http://schemas.xmlsoap.org/soap/envelop"
    xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext">
    <s:Header>
    <wsse:Security>
        <!-- Kerberos安全证书声明-->
        <wsse:BinarySecurityToken>
              ValueType=“wsse:Kerberosv5ST”
               EncodingType=“wsse:Base64Binary” >
               XSETt…
        </wsse:BinarySecurityToken>
     </wsse:Security>
     </s:Header>
   ...
 </s:Envelope>
```

BinarySecurityToken元素包含了二进制的Kerberos安全令牌，其中ValueType属性指明遵循的是Kerberos V5标准。

3） X.509数字签名：

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
    xmlns:s=http://schemas.xmlsoap.org/soap/envelop
```

```
    xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext">
    <s:Header>
    <wsse:Security>
        <!-- X.509安全证书凭证声明-->
        <wsse:BinarySecurityToken>
              ValueType=“wsse:X509v3” wsu:Id=“X509Cert”
               EncodingType=“wsse:Base64Binary”>
               XSETt…
        </wsse:BinarySecurityToken>
        <!--数字签名声明-->
         <ds:Signature  xmlns=“http://www.w3.org/2000/09/xmldsig#”>
         <ds:SignedInfo>
        <ds:CanonicalizationMethod
            Algorithm=“http://www.w3.org/2000/10/xml-3xc-c14N/”>
         <ds:SignatureMethod
            Algorithm=“http://www.w3.org/2000/09/xmlsig#rsa-sha1”>
         <ds:Reference  URI=“MessageBody”>
                    <ds:DigestValue>…</ds:DigestValue>
         </ds:Reference>
         <ds:SignedValue>
                …
         </ds:SignedValue>
         <ds:keyInfo> … </ds:keyInfo>
     </ds:SignedInfo>
     </wsse:Security>
    </s:Header>
    <s:Body  wsu:Id=“MessageBody”>
          ...
     </s:Body>
 </s:Envelope>
```

BinarySecurityToken元素指定了消息发送者的安全证书符合X.509标准，且在XML-Signature为消息体进行了数字签名。

4) 为SOAP信息体加密：

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
    xmlns:s="http://schemas.xmlsoap.org/soap/envelop"
    xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext"
    xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
    xmlns:ds="http://www.w3.org/2002/09/xmldsig#"
    xmlns:xenc="http://www.w3.org/2002/09/xmlenc#">
    <s:Header>
    <wsse:Security>
        <!-- 加密方法声明-->
```

```
        <xenc:EncryptionKey>
            <xenc:EncryptionMethod
             Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
        <ds:KeyInfo>
            <ds:KeyName>
                CN=Key13, C=US
            </ds:KeyName>
        </ds:KeyInfo>
          <xenc:CipherData>
           <xenc:CipherValue>
              fds7#rt...
           </xenc:CipherValue>
         </xenc:CipherData>
         <xenc:ReferenceList>
            <xenc:DataReference  URI="#EncryptionMessageBody"/>
         </xenc:ReferenceList>
       </xenc:EncryptionKey>
     </wsse:Security>
   </s:Header>
   <s:Body>
      <!-- 加密声明 -->
      <xenc:EncryptedData wsu="EncryptedMessageBody">
         <xenc:EncryptedMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#triplededs-
cbc"/>
         <xenc:CipherData>
            <xenc:CipherValues>
               GDSW#df...
            </xenc:CipherValues>
         </xenc:CipherData>
     </xenc:EncryptedData>
   </s:Body>
 </s:Envelope>
```

EncryptedData元素包含了实际加密的数据，加密方法和密钥在消息头中给出。