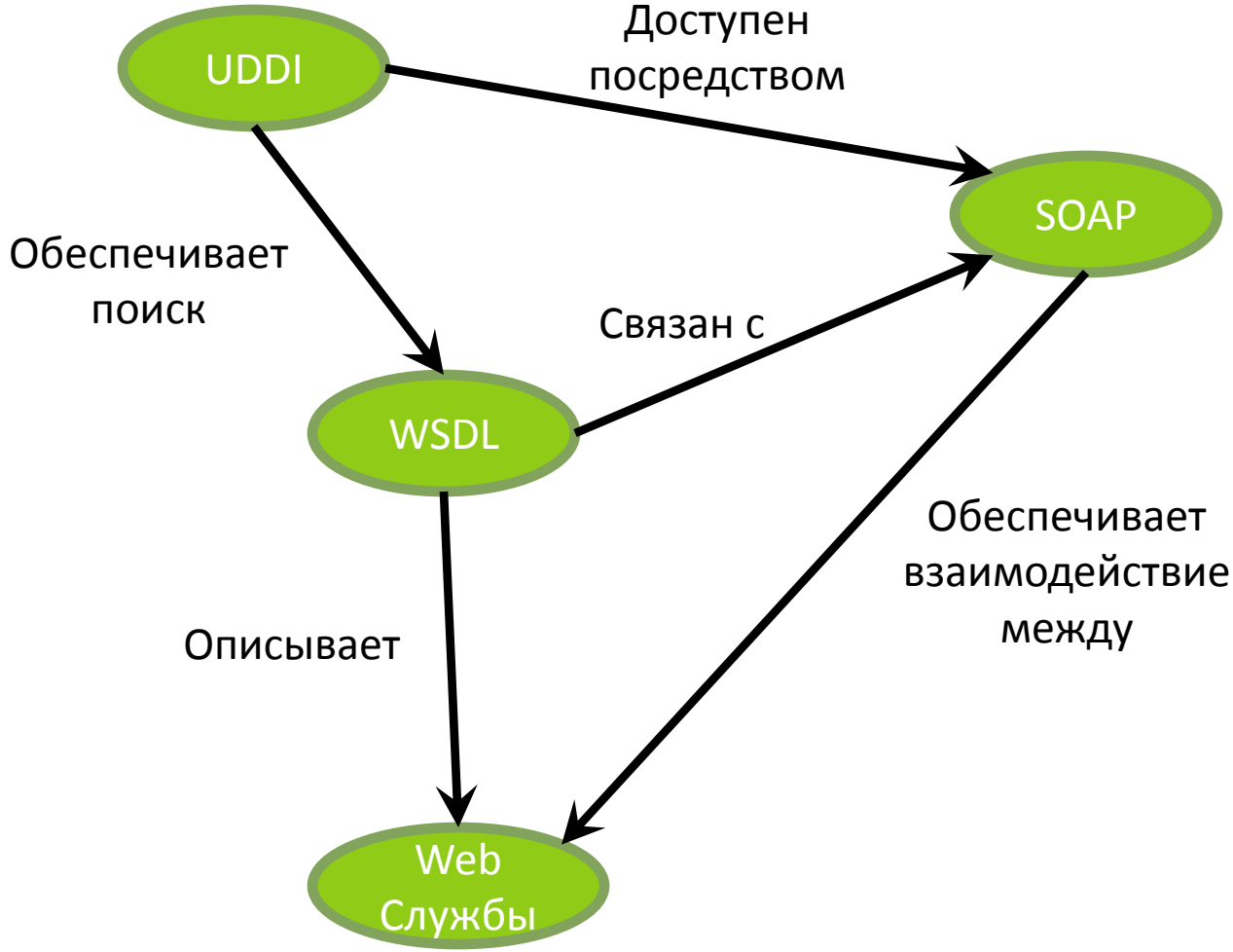
Распределенные вычислительные системы

Веб-сервисы - продолжение

Введение. Второе поколение стандартов WS

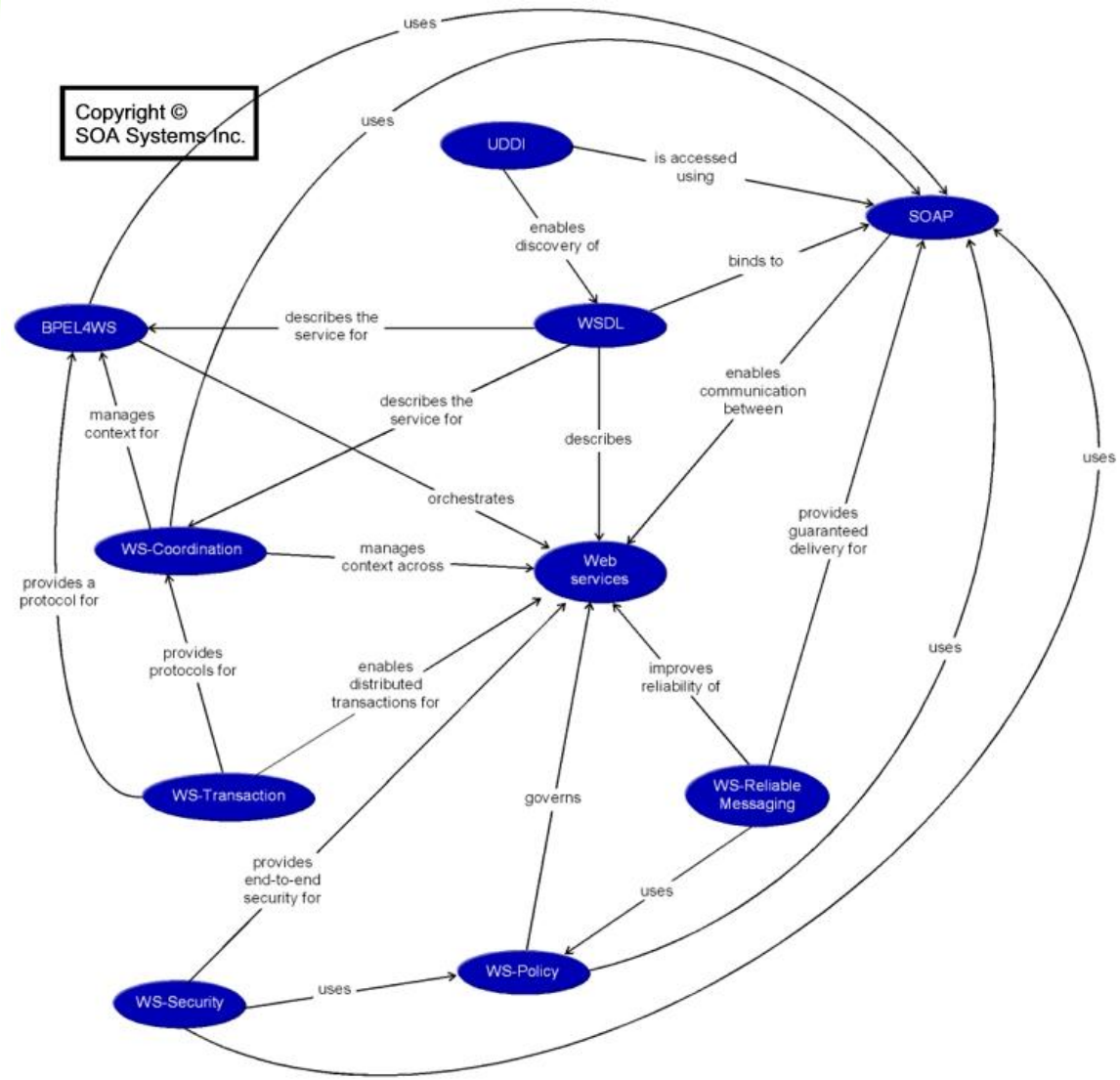
Стандарты WS первого поколения



Некоторые стандарты WS-* (стандарты второго поколения)

1. WS-Coordination
2. WS-Transaction (and the WS-TX TC)
3. WS-AtomicTransaction
4. WS-BusinessActivity
5. WS-BPEL
6. BPEL4WS
7. WS-ReliableMessaging (and the WS-RX TC)
8. WS-Addressing
9. WS-Attachments
10. SwA
11. DIME
12. Plain Old XML (POX)
13. Representational State Transfer (REST)
14. WS-CDL (Choreography Description Language)
15. WS-Security (and the WS-SX TC)
16. WS-Federation
17. WS-SecureConversation
18. WS-Trust
19. XML Encryption
20. XML Signature
21. XKMS
22. XACML
23. SAML
24. WS-I Basic Security Profile
25. WS-Policy
26. WS-PolicyAssertions
27. WS-PolicyAttachments
28. WS-MetadataExchange
29. WS-Eventing
30. WS-Notification
31. WS-RF (Resource Framework)

Некоторые стандарты WS-* (стандарты второго поколения)



http://www.soaspecs.com

Организации-разработчики стандартов WS-*

Коммерческие организации

- Microsoft
- IBM
- Sun
- Oracle
- Globus
- …

Консорциумы по стандартизации

- W3C (World Wide Consortium)
- OASIS (Organization for the Advancement of Structured Information Standards)
- GGF (Global grid Forum)
- DMTF (Distributed Management Task Force)
- WS-I (Web Services Interoperability Organization)
- …

Сферы WS-* стандартов

Безопасность

- WS-Security
  - Microsoft, IBM, OASIS
- XML Encryption, XML Signature
  - W3C
- ...

Маршрутизация и адресация

- WS-Addressing
  - W3C, Microsoft
- WS-Attachments
  - IBM
- WS-RX TC
  - OASIS
- ...

Ориентированные на грид и другие области

- WSRF
  - OASIS
- WS-Notification
  - IBM
- WS-Eventing
  - Microsoft
- ...

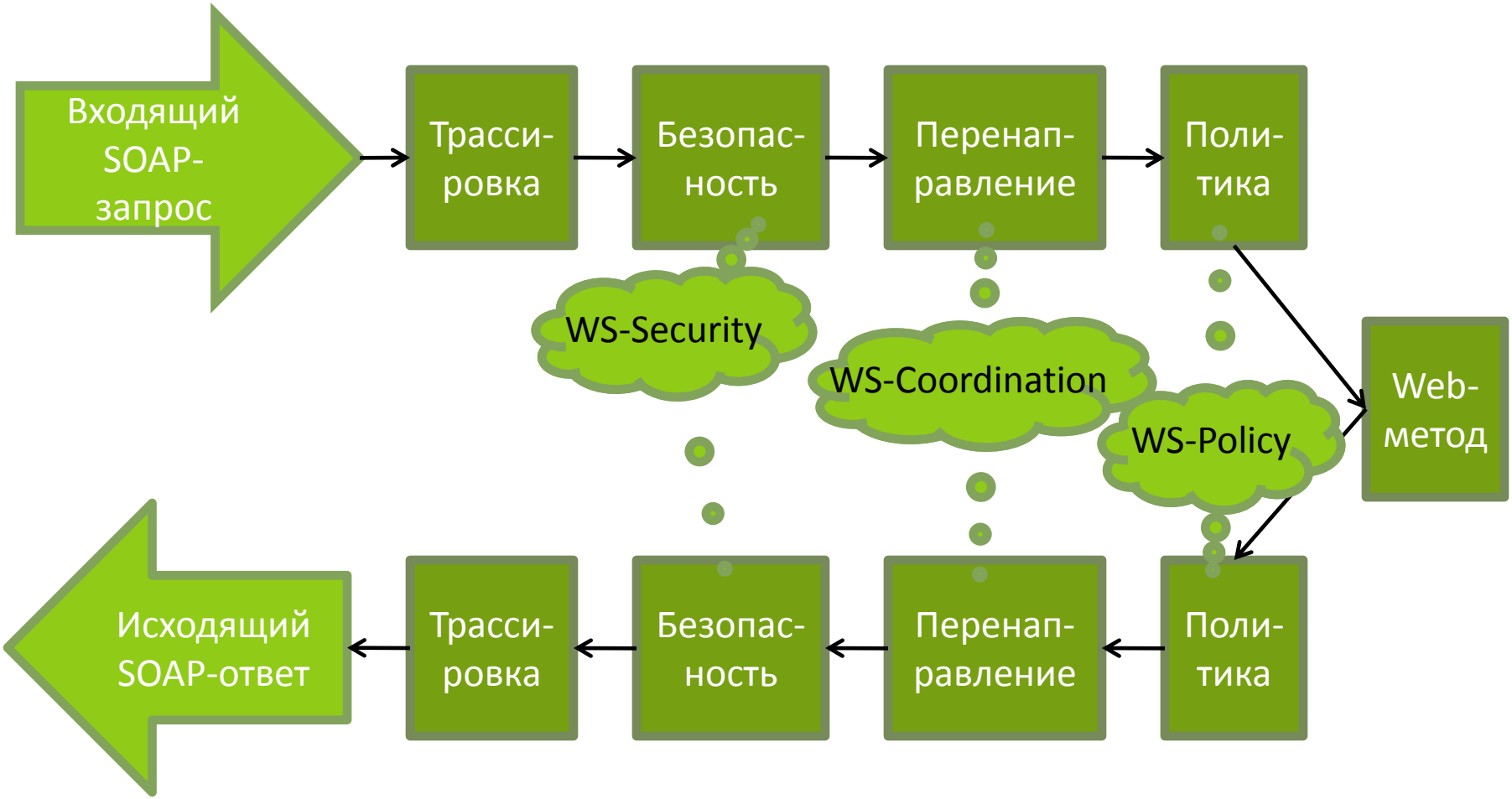
Бизнес-процессы, Workflow

- WS-BPEL
  - OASIS
- BPEL4WS
  - IBM, Microsoft

Управление транзакциями и контекстом

- WS-Coordination
  - IBM, Microsoft
- WS-Transaction
  - OASIS, IBM, Microsoft
- ...

Цепочки SOAP-фильтров



Девид С. Платт Платформа Microsoft 2005. М.: Русская Редакция. -2004

Безопасность WS и WS-Security

Проблемы с безопасностью у WS первого поколения

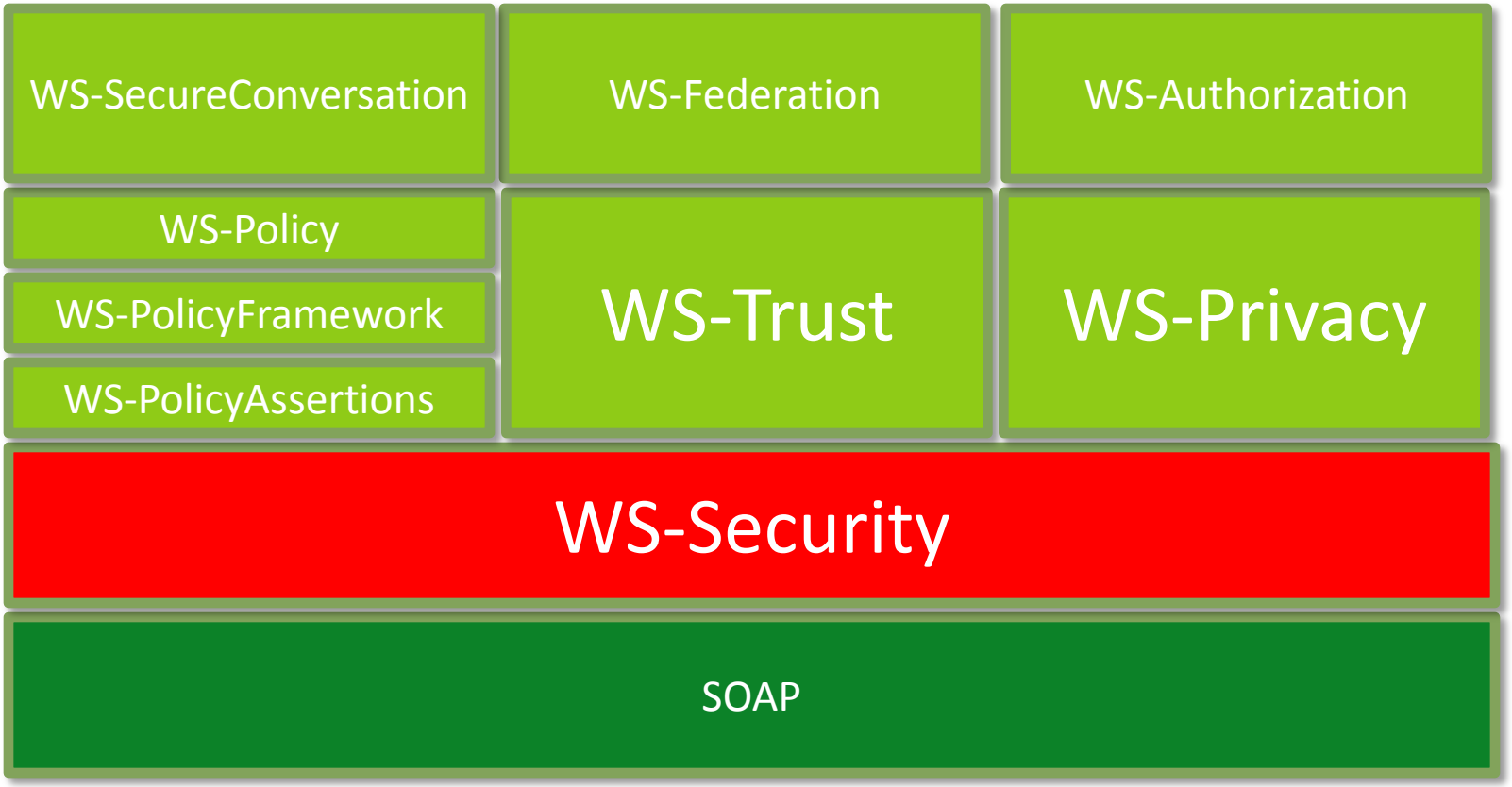
- Стандарты Web служб первого поколения не подразумевали обеспечения какой-либо безопасности.
- Таким образом, практическое применение WS в бизнес сфере было значительно ограничено.
- Отсутствовали стандартные решения аутентификации, разграничения прав доступа, шифрования и т.п. Таким образом, каждый решал задачу безопасности самостоятельно.

Стек протоколов обеспечения безопасности в СХА

- СХА - Global XML Web Services Architecture: семейство стандартов, представленных когламератом IBM+Microsoft+… Проходят стандартизацию OASIS и W3C

- Конкурирующий набор стандартов: SAML представлен корпорациями близкими Sun Microsystems и Oracle.

Стек протоколов обеспечения безопасности в СХА



Влад Боркус. Безопасность Web-сервисного взаимодействия. – PCWEEK, №33(447). -2004

WS-Security – комплексное решение задач безопасности WS

Стандарт WS-Security ориентирован на комплексное решение задач безопасности при взаимодействии Web-служб:

- Идентификация
- Цифровые подписи
- Шифрование

Это позволяет ответить на такие вопросы безопасности, как:

- Кого я авторизую?
- Было ли изменено сообщение при пересылке?
- Пришло ли это сообщение именно от того, от кого я думаю?
- Как я спрячу конфиденциальную информацию?

WS-Security – это только архитектура. Реальные операции по обеспечению безопасности она полагается на технологии PKI, Kerberos, SSL и т.п.

Scott Seely. Понимание WS-Security. Microsoft corp. [http://msdn.microsoft.com/webservices/understanding/advancedwebservices/default.aspx]

WS-Security и SOAP сообщения

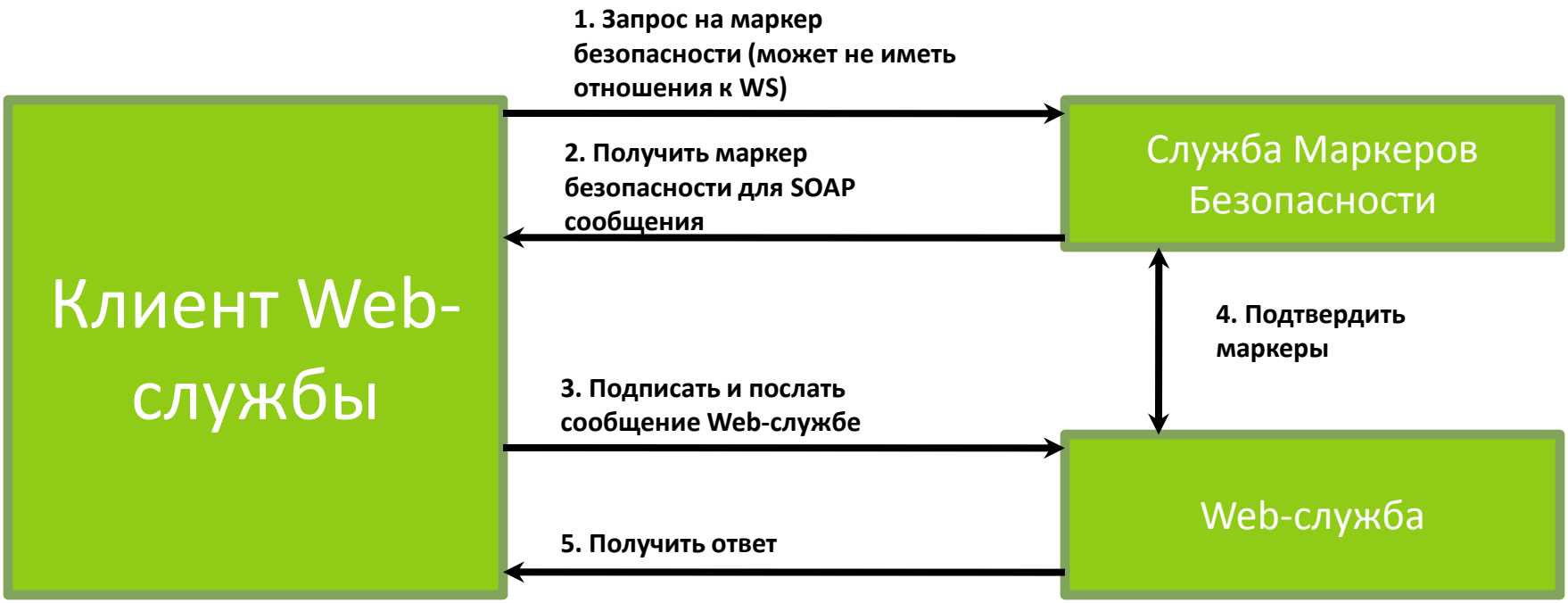
WS-Security переносит процедуру идентификации и авторизации в пространство SOAP-сообщений.

Используя *маркеры безопасности (Security Tokens)* SOAP сообщение может переправить следующую информацию:

- Идентификацию вызывающего: *Я User Vasya Pupkin*.
- Принадлежность к группе: *Я разработчик PupkinSite.com*.
- Подтверждение прав: *Поскольку я разработчик PupkinSite.com, я могу создавать базы данных и добавлять Web приложения в машины PupkinSite.com.*

Scott Seely. Понимание WS-Security. Microsoft corp. [http://msdn.microsoft.com/webservices/understanding/advancedwebservices/default.aspx]

Процедура обеспечения безопасности при передаче сообщения



Scott Seely. Понимание WS-Security. Microsoft corp. [http://msdn.microsoft.com/webservices/understanding/advancedwebservices/default.aspx]

Аутентификация пользователя

Внедряется в заголовок SOAP-сообщения:

```
<soap:Envelope>
   <soap:Header>
      <wsse:Security soap:mustUnderstand="1">
         <wsse:UsernameToken>
            <wsse:Username>scott</wsse:Username>
            <wsse:Password Type="wsse:PasswordText">password
            </wsse:Password>
         </wsse:UsernameToken>
     </wsse:Security>
   </soap:Header>
   ...
</soap:Envelope>
```

Виды аутентификации WS-Security

1. <wsse:UsernameToken> - аутентификация пользователя посредством пары Имя пользователя-пароль.

2. <wsse:X509v3> - аутентификация посредством сертификата X.509v3

3. Kerberos – аутентификация посредством протокола Kerberos (Kerberos Domain Controller) (используется в Windows2000, Red Hat Linux и т.п.)

UsernameToken - имя пользователя и пароль

Пароль в виде простого текста (например, при использовании SSL):

```
<wsse:UsernameToken>
    <wsse:Username>scott</wsse:Username>
    <wsse:Password Type="wsse:PasswordText">password</wsse:Password>
</wsse:UsernameToken>
```

Пароль в виде цифрового хэша:

```
<wsse:UsernameToken>
    <wsse:Username>scott</wsse:Username>
    <wsse:Password Type="wsse:PasswordDigest">
        KE6QugOpkPyT3Eo0SEgT30W4Keg=</wsse:Password>
    <wsse:Nonce>5uW4ABku/m6/S5rnE+L7vg==</wsse:Nonce>
    <wsu:Created xmlns:wsu=
        "http://schemas.xmlsoap.org/ws/2002/07/utility">
            2002-08-19T00:44:02Z
    </wsu:Created>
</wsse:UsernameToken>
```

X509v3 – сертификат безопасности

```
<wsse:BinarySecurityToken
    ValueType="wsse:X509v3"
    EncodingType="wsse:Base64Binary"
    Id="SecurityToken-f49bd662-59a0-401a-ab23-1aa12764184f">
  MIIHdjCCBCCAWqgAwIBAgIBGzANBgkqhkiG9w0BAQQFADBHMQ0wCwYDVQQKEwR
  MRwwGgYDVQQLExNDYWViZWFucyBkZXZlbG9wZXJzMRgwFgYDVQQLEw9jYWViZW
  Fucy5uZXQucnUwHhcNMDcxMjAxMDAwMDAwWhcNMDgwMTMxMjM1OTU5Wj...
  </wsse:BinarySecurityToken>
```

Для обеспечения безопасности при использовании сертификата надо прибегнуть к дополнительным средствам обеспечения безопасности:

- подпись сообщения секретным ключом сертификата;
- добавление wsu:Timestamp для определения времени жизни сообщения.

Подпись сообщения

- Подпись сообщения лежит в рамках спецификации XML Signature.
- При подписи сообщения используется секретная аутентификационная информация:
  - UsernameToken – пароль пользователя;
  - X.509 – секретный ключ;
  - Kerberos – сеансовый ключ.

Подпись сообщения

```
<soap:Envelope>
    <soap:Header>
       <wsse:Security soap:mustUnderstand="1">
       ...
      <ds:Signature>
          <ds:Signedlnfo>
                ...
          </ds:Signedlnfo>
          <ds:SignatureValue>
                Hp1ZkmFZ/2kQLXDJbchm5gK...
          </ds:SignatureValue>
      <ds:Keylnfo>
          <wsse:SecurityTokenReference>
             <wsse:Reference URI=" #X509Token"/>
          </wsse:SecurityTokenReference>
      </ds:Keylnfo>
       </ds:Signature>
    </wsse:Security>
    </soap:Header>
    ...
</soap:Envelope>
```

*Пространство имен ds принадлежит спецификации XML Signature

Влад Боркус. Безопасность Web-сервисного взаимодействия. – PCWEEK, №33(447). -2004

Шифрование

- Аутентификация и подпись сообщения – это не всегда достаточная мера обеспечения безопасности, особенно при передаче конфиденциальной информации.
- За шифрование отвечает стандарт XML Encryption.

Шифрование

```
<soap:Envelope>
    <soap:Header>
       <wsse:Security soap:mustUnderstand="1">
          ...
       </wsse:Security>
    </soap:Header>
    <soap:Body>
      <xenc:EncryptedData
      Id="EncryptedContent-f6f50b24-3458-41d3-aac4-390f476f2e51"
      Type="http://www.w3.org/2001/04/xmlenc#Content">
         <xenc:EncryptionMethod Algorithm=
         "http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
         <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
            <KeyName>Symmetric Key</KeyName>
         </KeyInfo>
         <xenc:CipherData>
            <xenc:CipherValue>
             InmSSXQcBV5UiT...  Y7RVZQqnPpZYMg==
             </xenc:CipherValue>
         </xenc:CipherData>
      </xenc:EncryptedData>
    ...
</soap:Envelope>
```

*Пространство имен xenc принадлежит спецификации XML Encryption

Влад Боркус. Безопасность Web-сервисного взаимодействия. – PCWEEK, №33(447). -2004

Адресация и WS-Addressing

Адресация в стандартах первого поколения (WSDL)

- В стандартах первого поколения полный адрес WS содержался в WSDL-описании WS в блоке <port>. Это может доставить значительные неудобства, т.к. при изменении адреса службы приходится редактировать WSDL-файл целиком.
- При обмене сообщениями SOAP по стандарту первого поколения, адресация возложена на транспортный протокол (при связывании с HTTP) и не может быть изменена непосредственно в SOAP-сообщении.

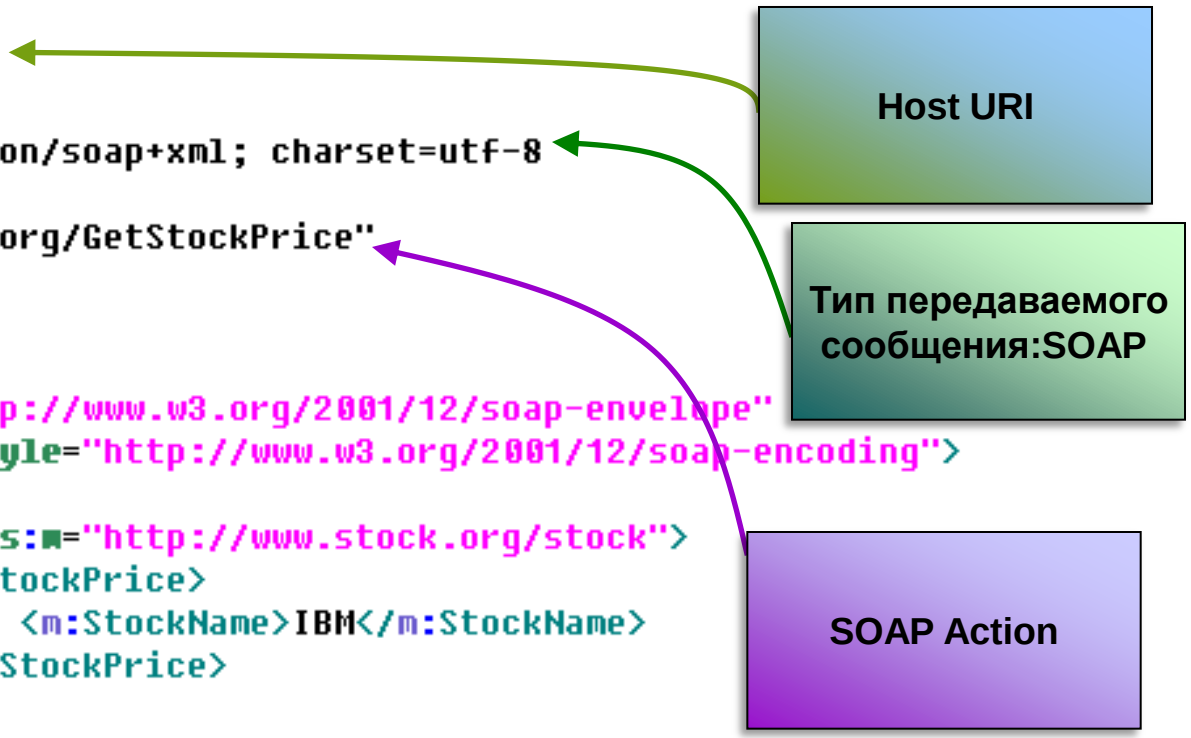
Пример адресации SOAP по стандарту первого поколения

```
POST /InStock HTTP/1.1
Host: www.stock.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
SOAPAction: "www.stock.org/GetStockPrice"

<?xml version="1.0"?>
<soap:Envelope
        xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
        soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

        <soap:Body xmlns:m="http://www.stock.org/stock">
                <m:GetStockPrice>
                        <m:StockName>IBM</m:StockName>
                </m:GetStockPrice>
        </soap:Body>

</soap:Envelope>
```



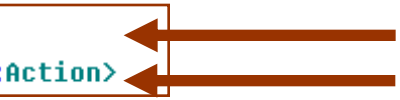
Описание WS-Addressing ведется по презентации:Siyamed Seyhmus. WS-Addressing. SINIR SRDC

Использование WS-Addressing для обеспечения независимости от транспорта

В стандарте WS-Addressing предусматривается введение полей <wsa:To> и <wsa:Action> определяющих URI приемника сообщения и соответствующее действие:

```
<?xml version="1.0"?>
<soap:Envelope
        xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
        soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
        xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing">
        <soap:Header>
                <wsa:To>http://www.stock.org/InStock</wsa:To>
                <wsa:Action>http://www.stock.org/GetStockPrice</wsa:Action>
        </soap:Header>
        <soap:Body xmlns:m="http://www.stock.org/stock">
                <m:GetStockPrice>
                        <m:StockName>IBM</m:StockName>
                </m:GetStockPrice>
        </soap:Body>

</soap:Envelope>
```



Управление ответом

- В стандарте первого поколения подразумевается, что ответное сообщение передается по уже открытому HTTP каналу.
- При этом, нет стандартной поддержки асинхронной коммуникации между Web службами.
- Стандарт WS-Addressing вводит следующие поля: <MessageID>, <From>, <ReplyTo>, <FaultTo>, <RelatedTo>

Управление ответом

```
POST /InStock HTTP/1.1
Host: www.stock.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
SOAPAction: "www.stock.org/GetStockPrice"

<?xml version="1.0"?>
<soap:Envelope
   xmlns:soap="http://www.w3.org/2001/12/soap-env
   soap:encodingStyle="http://www.w3.org/2001/12/
   xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing">
   <soap:Header>
      <wsa:To>http://www.stock.org/InStock</wsa:To>
      <wsa:Action>http://www.stock.org/GetStockPrice</wsa:Action>
      <wsa:From>
         <wsa:Address>http://www.yalova.srdc.metu.edu.tr/WSCaller</wsa:Address>
      </wsa:From>
      <wsa:ReplyTo>
         <wsa:Address>http://www.yalova.srdc.metu.edu.tr/WSReplyListener</wsa:Address>
      </wsa:ReplyTo>
      <wsa:FaultTo>
         <wsa:Address>http://www.yalova.srdc.metu.edu.tr/WSFaultListener</wsa:Address>
      </wsa:FaultTo>
      <wsa:MessageID>uuid:someid</wsa:MessageID>
      <wsa:RelatesTo RelationshipType="Reply">uuid:someotherid</wsa:MessageID>
   </soap:Header>
   <soap:Body xmlns:m="http://www.stock.org/stock">
      <m:GetStockPrice>
         <m:StockName>IBM</m:StockName>
      </m:GetStockPrice>
   </soap:Body>

</soap:Envelope>
```

Передается адрес отправителя
адрес получения ответа и
адрес для извещения об ошибках

У данного сообщения есть идентификатор “uuid:someid”, и оно **относится (related)** к сообщению “uuid:someotherid” как **Ответ (“Reply”)**

Адресация конечных точек

- WS-Addressing обеспечивает расширенную адресацию конечных точек.
- Так как WSDL не поддерживает расширение элемента «Service», в WS-Addressing определен элемент <EndpointReference>, который может быть использован в WSDL.

<EndpointReference> VS < Service >

```
<wsa:EndpointReference>
        <wsa:Address/>
        <wsa:ReferenceProperties/>
        <wsa:ServiceName PortName=""/>
        <wsa:PortType/>
        <wsp:Policy/>
</wsa:EndpointReference>
```

*EndpointReference element

```
<service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
       <soap:address location="http://example.com/stockquote"/>
    </port>
 </service>
```

*WSDL Service element

<EndpointReference> расширяет элемент <Service> добавляя поля ReferenceProperties и Policy. Address, ServiceName и PortType уже включены в элемент <Service>.

Формирование SOAP-сообщения на основе <EndpointReference>

```
<wsa:EndpointReference xmlns:wsa="..." xmlns:fabrikam="...">
   <wsa:Address>http://www.fabrikam123.example/acct</wsa:Address>
   <wsa:ReferenceProperties>
       <fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
       <fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
   </wsa:ReferenceProperties>
</wsa:EndpointReference>
```

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
   xmlns:wsa="="http://schemas.xmlsoap.org/ws/2004/08/addressing"
   xmlns:fabrikam="... ">
   <S:Header>
        ...
        <wsa:To>http://www.fabrikam123.example/acct</wsa:To>
        <fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
        <fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
        ...
   </S:Header>
   <S:Body>
        ...
   </S:Body>
</S:Envelope>
```

Конструкции WS-Addressing

- Таким образом, WS-Addressing определяет два вида конструкций, позволяющих унифицировать адресацию WS независимо от базового транспортного протокола:

- <EndpointReference> (описание адресации конечных точек)

```
<wsa:EndpointReference>
        <wsa:Address/>
        <wsa:ReferenceProperties/>
        <wsa:ServiceName PortName=""/>
        <wsa:PortType/>
        <wsp:Policy/>
</wsa:EndpointReference>
```

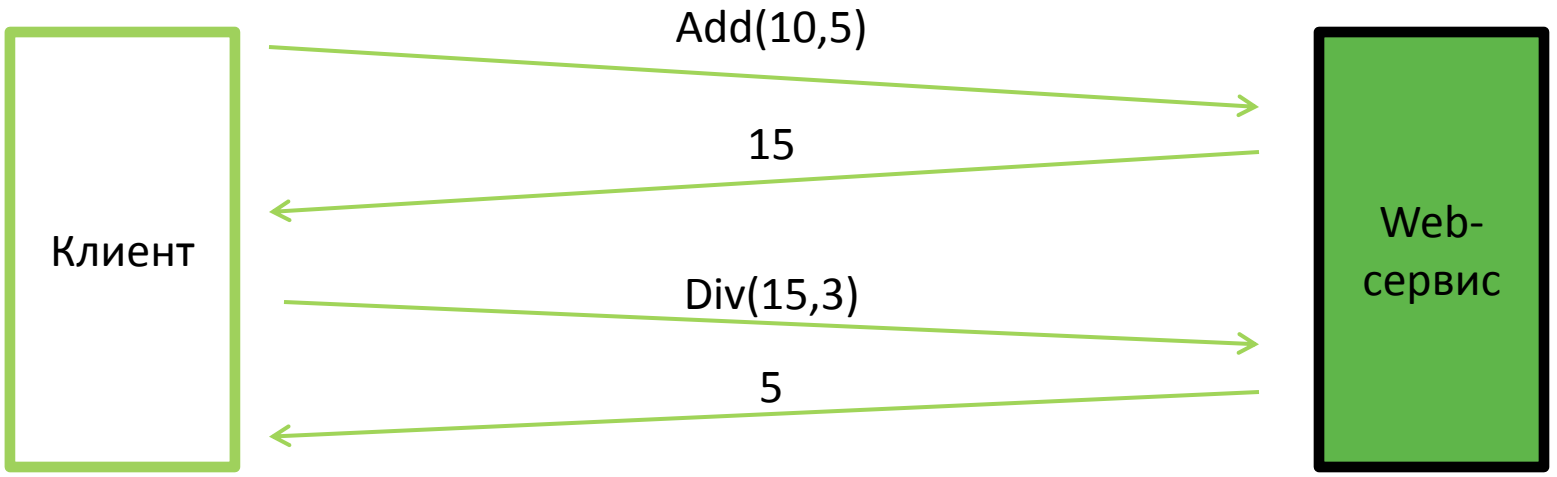
- Message Information Header (обеспечение асинхронного, транспортно-независимого взаимодействия между WS)

```
<wsa:MessageID> xs:anyURI </wsa:MessageID>
<wsa:RelatesTo RelationshipType="..."?>xs:anyURI</wsa:RelatesTo>
<wsa:To>xs:anyURI</wsa:To>
<wsa:Action>xs:anyURI</wsa:Action>
<wsa:From>endpoint-reference</wsa:From>
<wsa:ReplyTo>endpoint-reference</wsa:ReplyTo>
<wsa:FaultTo>endpoint-reference</wsa:FaultTo>
<wsa:Recipient>endpoint-reference</wsa:Recipient>
```

Состояние WS и WSRF

«Состояния» и стандарты первого поколения

- Изначально, использование Web-служб не подразумевало существования «Состояния».
- Типовой сценарий использования Web-службы: запрос – ответ – отключение. Каждый следующий запрос не зависит от предыдущего запроса.



WSRF

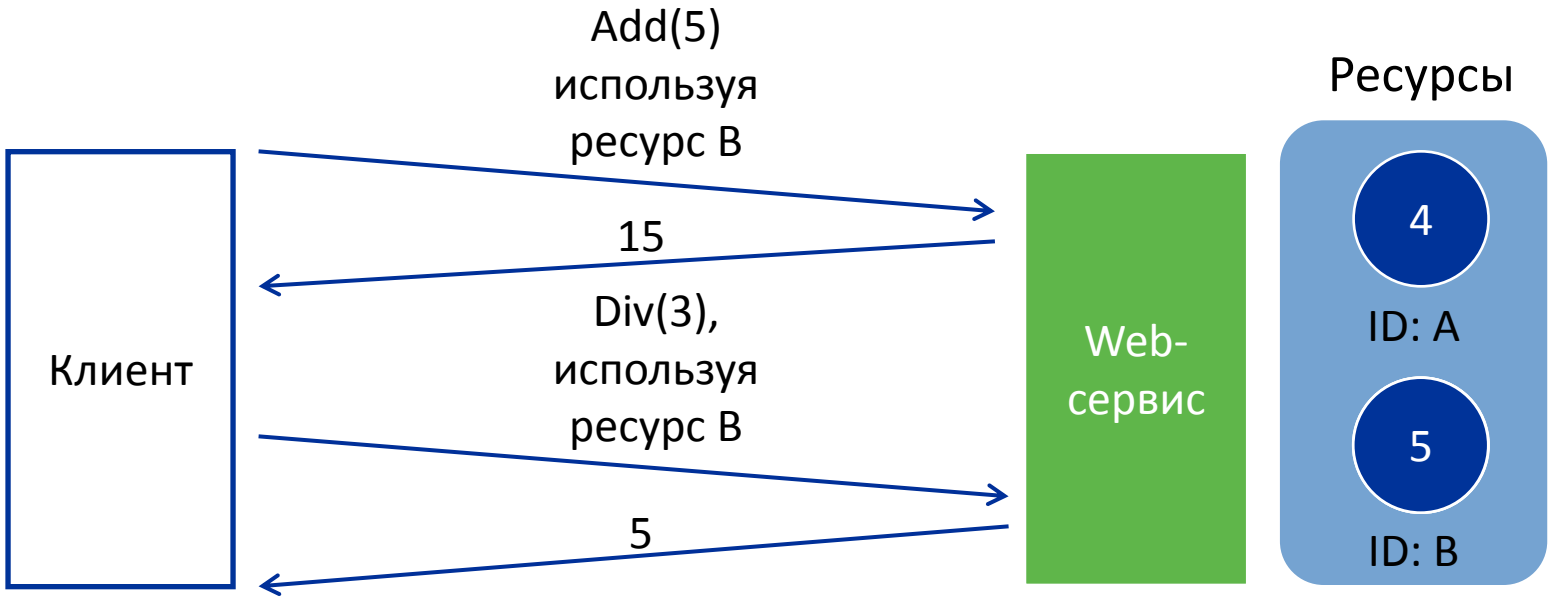
Предложена в 2004-м году, утверждена в качестве стандарта OASIS в 2006-м году. Включает следующие стандарты:

- WS-Resource specification
- WS-ResourceProperties (WSRF-RP)
- WS-ResourceLifetime (WSRF-RL)
- WS-ServiceGroup (WSRF-SG)
- WS-BaseFaults (WSRF-BF)

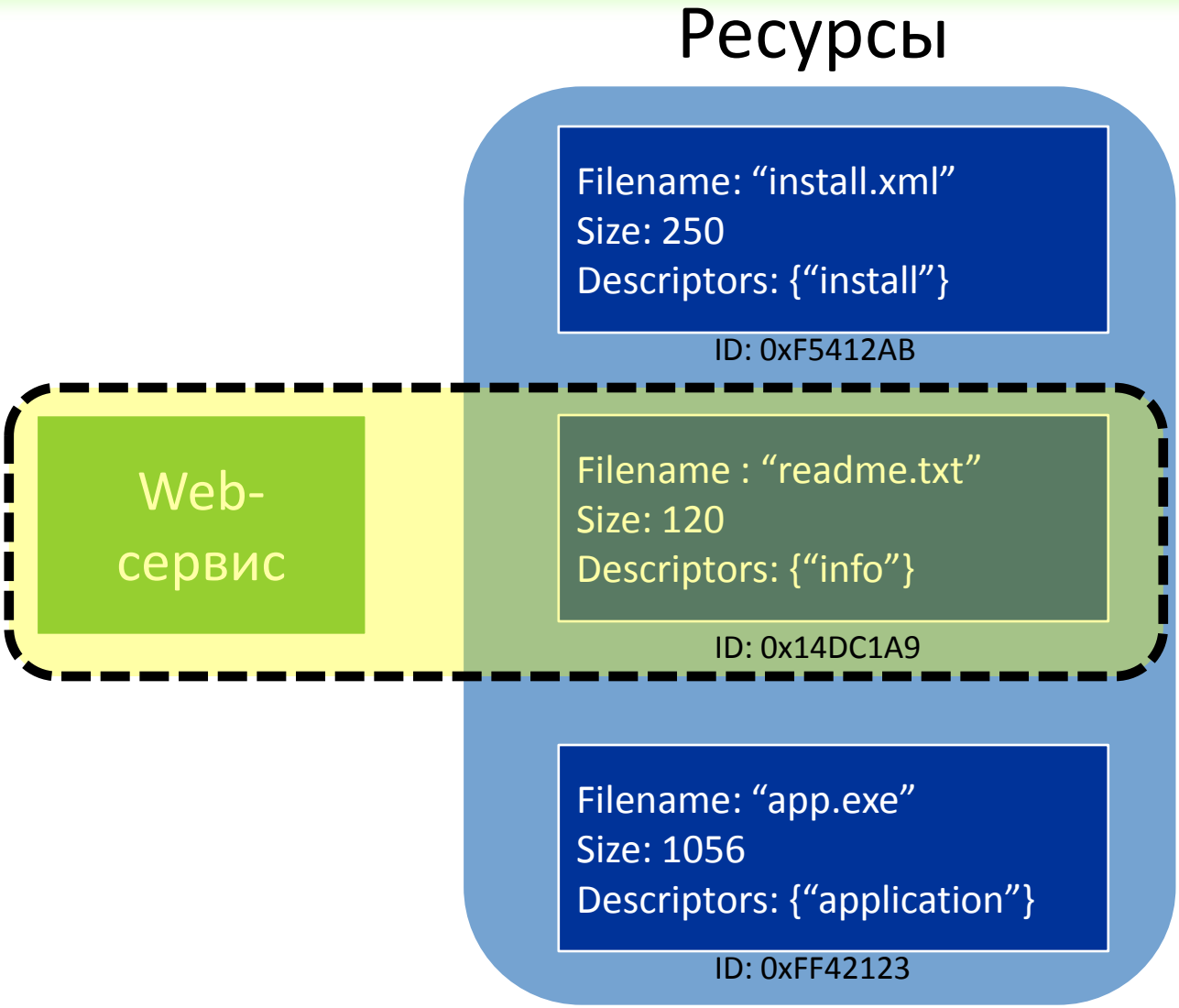
Концепция WSRF - Web Service Resource Framework

Для разработки Grid не получилось применить “чистые” Web-сервисы, т.к. они не обладали “состоянием”.

WSRF является попыткой решить указанную архитектурную проблему с помощью введения понятия «состояние» в Web-сервисы, превратив их в Web-ресурсы, и указав механизмы использования этого понятия.



Web-сервис + Ресурс = WS-ресурс



1. Определение свойств ресурса

```
<satProp:GenericSatelliteProperties
xmlns:satProp="http://example.com/satellite">
    <satProp:latitude>30.3</satProp:latitude>
    <satProp:longitude>223.2</satProp:latitude>
    <satProp:altitude>47700</satProp:altitude>
    <satProp:pitch>49</satProp:pitch>
    <satProp:yaw>0</satProp:yaw>
    <satProp:roll>32</satProp:roll>
    <satProp:focalLength>21999992</satProp:focalLength>
    <satProp:currentView>
        http://example.com/satellite/2239992333.zip
    </satProp:currentView>
</satProp:GenericSatelliteProperties>
```

2. Базовый WSDL-файл

```
<?xml version="1.0" encoding="UTF-8"?>

<definitions name="Satellite"
    targetNamespace="http://example.com/satellite"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:tns="http://example.com/satellite"
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsrp=
    "http://docs.oasis-open.org/wsrf/2004/06/wsrf-
    WS-ResourceProperties-1.2-draft-01.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <wsdl:import namespace=
    "http://docs.oasis-open.org/wsrf/2004/06/wsrf-
    WS-ResourceProperties-1.2-draft-01.wsdl"
    location="WS-ResourceProperties.wsdl" />
    <types>
        <xsd:schema targetNamespace="http://example.com/satellite"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
                <xsd:import namespace=
                "http://schemas.xmlsoap.org/ws/2004/03/addressing"
                schemaLocation="WS-Addressing.xsd" />
        </xsd:schema>
    </types>
</definitions>
```

3. Добавление ресурса к WSDL

```
<definitions name="Satellite" ...>
    ...
    <types>
        <xsd:schema targetNamespace="http://example.com/satellite"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
                ...
                <xsd:element name="latitude" type="xsd:float" />
                <xsd:element name="longitude" type="xsd:float" />
                <xsd:element name="altitude" type="xsd:float" />
                ...
                <xsd:element name="GenericSatelliteProperties">
                <xsd:complexType>
                        <xsd:sequence>
                        <xsd:element ref="latitude" minOccurs="1"
                        maxOccurs="1"/>
                        <xsd:element ref="longitude" minOccurs="1"
                        maxOccurs="1"/>
                        ...
                        </xsd:sequence>
                </xsd:complexType>
                </xsd:element>
        </xsd:schema>
    </types>
        <portType name="SatellitePortType"
                wsrp:ResourceProperties=
                "tns:GenericSatelliteProperties">
        </portType>
</definitions>
```

4. Операция на запрос ресурса

```
...
    <types>
        ...
        <xsd:element name="createSatellite">
                <xsd:complexType/>
        </xsd:element>
        <xsd:element name="createSatelliteResponse"> <xsd:complexType>
                <xsd:sequence>
                <xsd:element ref="wsa:EndpointReference"/>
                </xsd:sequence>
        </xsd:complexType> </xsd:element>
        ...
</types>
<message name="CreateSatelliteRequest">
    <part name="request" element="tns:createSatellite">
</message>
<message name="CreateSatelliteResponse">
    <part name="response" element="tns:createSatelliteResponse"/>
</message>
<portType name="SatellitePortType" wsrp:ResourceProperties=
                "tns:GenericSatelliteProperties">
    <operation name="createSatellite">
        <input message="tns:CreateSatelliteRequest"
                wsa:Action="http://example.com/CreateSatellite" />
        <output message="tns:CreateSatelliteResponse"
                wsa:Action="http://example.com/CreateSatelliteResponse"/>
    </operation>
</portType>
```

5. Запрос ресурса

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <createSatellite xmlns="http://example.com/satellite"/>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Ответ:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.
org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
        <wsa:EndpointReference
        xmlns:wsa="http://www.w3.org/2005/02/addressing"
        xmlns:sat="http://example.org/satelliteSystem">
                <wsa:Address>http://example.com/satellite</wsa:Address>
                <wsa:ReferenceProperties>
                        <sat:SatelliteId>SAT9928</sat:SatelliteId>
                </wsa:ReferenceProperties>
        </wsa:EndpointReference>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5. Запрос состояния ресурса

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/
soap/envelope/"
     xmlns:sat="http://example.org/satelliteSystem"
     xmlns:wsa="http://www.w3.org/2005/02/addressing"
     xmlns:wsrp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-
     WS-ResourceProperties-1.2-draft-01.xsd">
<SOAP-ENV:Header>
     <wsa:Action>
          http://docs.oasis-open.org/wsrf/2004/06/WS-ResourceProperties/
          GetResourceProperty
     </wsa:Action>
     <wsa:To SOAP-ENV:mustUnderstand="1">
          http://example.com/satellite
     </wsa:To>
     <sat:SatelliteId>SAT9928</sat:SatelliteId>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
     <wsrp:GetResourceProperty
          xmlns:satProp="http://example.com/satellite">
          satProp:altitude
     </wsrp:GetResourceProperty>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Реализация WSRF

- **The Globus Toolkit 4:** реализация WSRF на Java и C (WS-Core).
- **WebSphere Application Server 6.1:** предоставляет среду WSRF для реализации конечных точек.
- **Muse 2.0 (Apache Foundation):** реализация на Java WSRF, WS-Notification и WSDM (Web Services Distributed Management).
- **WSRF::Lite:** реализация WSRF на perl.
- **WSRF.NET:** реализация стандарта WSRF на .NET.
- **UNICORE 6.0:** реализация стандарта WSRF 1.2 и WS-Notification на Java.

Ссылки и литература

- http://www.soaspecs.com – перечень стандартов и спецификаций Web-служб с ссылками на организации-разработчики.
- http://www.devarticles.com/c/b/Web-Services - сборник статей по разработке и практическому применению Web-служб, включая стандарты WSRF, WS-Addressing и WS-Notification