WS-Resource Framework
とは何か？

稚内北星学園大学
丸山不二夫

OGSIからWSRFへ

Refactoring & Evolution

Grid = WSRF = Web Service

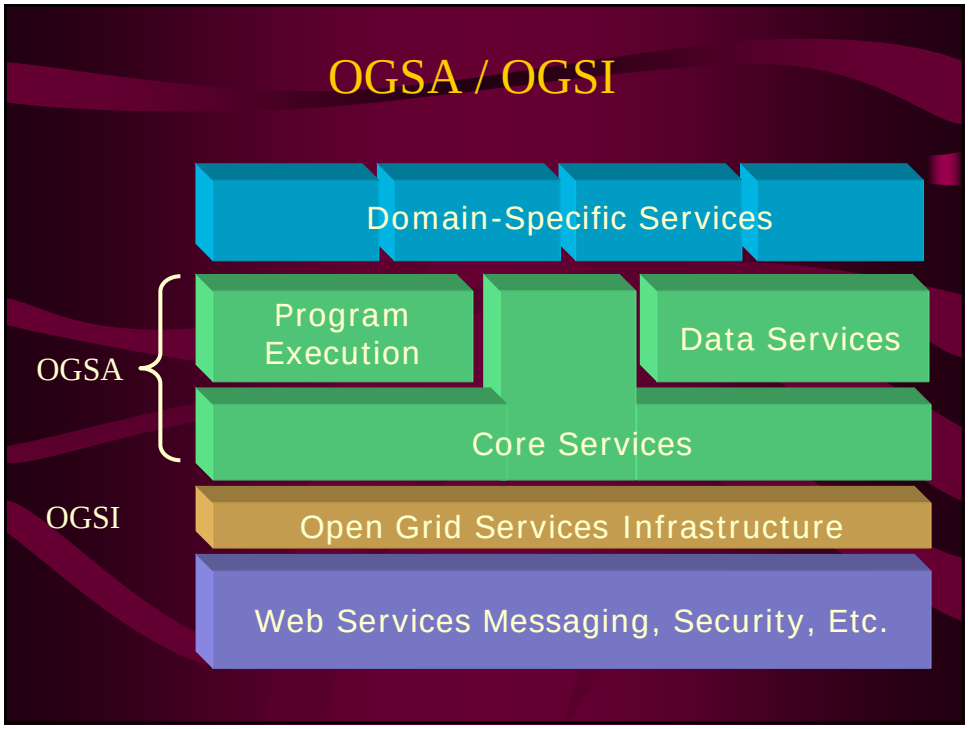
Grid と Web Servicesは接近してきた。

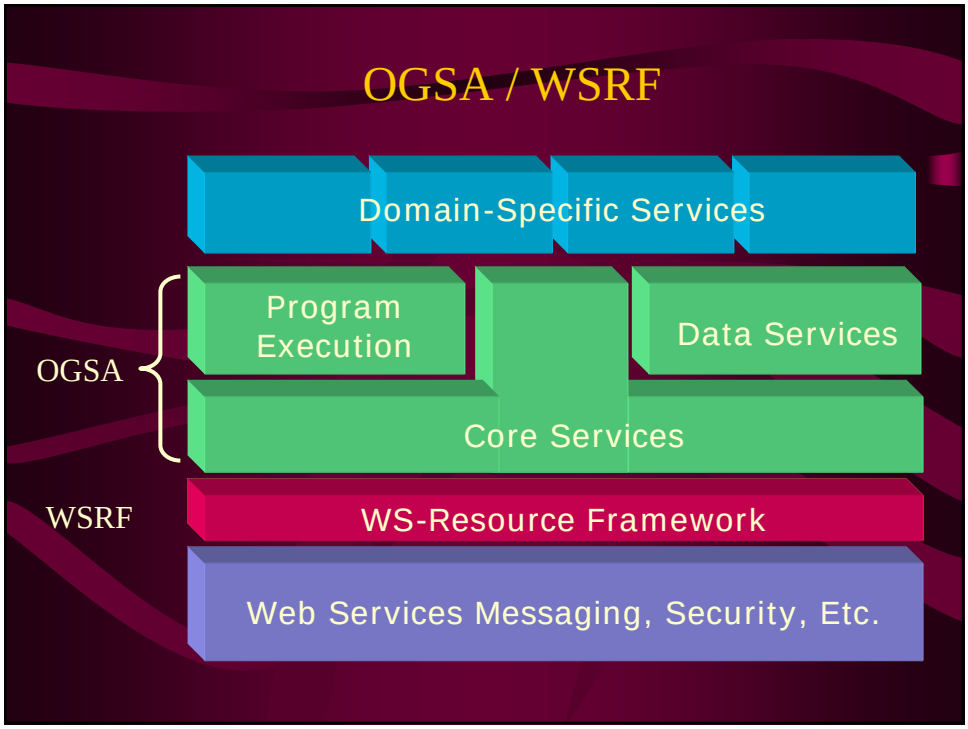
しかし、すれ違って別の道を
進む可能性が出てきた。

OGSIに対する熱狂にもかかわらず、Web サービスコミュニティからのOGSIの受容には 問題が多いことが明らかになった

Webサービスのコミュニティからの OGSIに対する四つの批判

1. ひとつの仕様にあまりに沢山の材料を詰め込みすぎてる。
2. Webサービスの一般的なツールが使えない
3. あまりにオブジェクト志向が強すぎた。
4. WSDL2.0で標準的に提供されるべき能力を、WSDL1.1の非標準的な拡大として実装した。


OGSA / OGSI
Domain-Specific Services
Program Execution
Data Services
OGSA
Core Services
OGSI
Open Grid Services Infrastructure
Web Services Messaging, Security, Etc.


OGSA / WSRF
Domain-Specific Services
Program Execution
Data Services
OGSA
Core Services
WSRF
WS-Resource Framework
Web Services Messaging, Security, Etc.


これまでのOGSA/OGSIと
Webサービスの関係
OGSA
OGSI
Web Service


これからのOGSA/WSRFと
Webサービスの関係
OGSA
Web Service = WSRF

GridとWeb Servicesは WSRFで、一体化する。

WSRFの定義は、GridとWebサービスのコミュニティが、
共通の土台の上に前進できるということを意味している。

WSRF関連のドキュメント群

Whitepapers & Specifications

http://www.globus.org/wsrf/#relevant

WSRFの基本的な whitepaper

- Modeling Stateful Resources with Web Services
- The WS-Resource Framework
- From Open Grid Services Infrastructure to WS-Resource

- Publish-Subscribe Notification for Web services

WSRFの5つの仕様

仕様名	内容
WS-ResourceLifetime	WSリソースの生存時間の管理、WSリソースの破棄
WS-ResourceProperties	WSリソースの定義。プロパティ値の獲得、設定、削除等
WS-RenewableReferences	使えなくなったエンドポイントリファレンスの更新機能
WS-ServiceGroup	複数のヘテロなWebサービスのグループを定義
WS-BaseFaults	基礎となるFault型のXMLでの定義

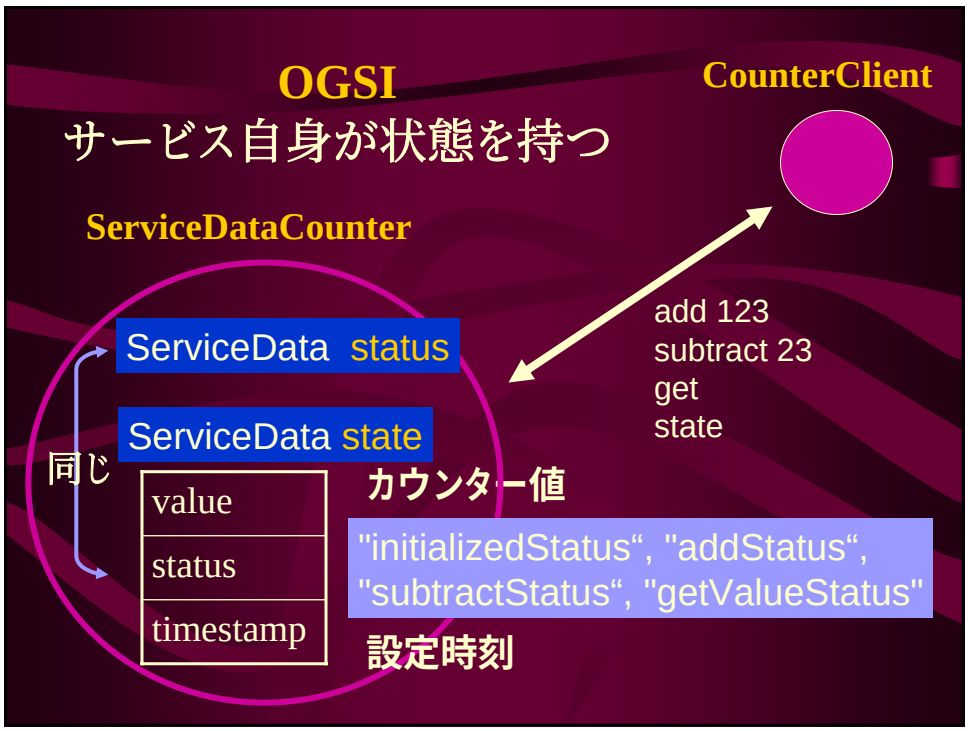
WS-Notificatonの3つの仕様

仕様名	内容
WS-BaseNotification	**通知の生産者と消費者間のWebサービスのインターフェース定義**
WS-Topics	**通知の購読に際して利用される「トピック」の系統的な構成法。あらかじめ、三つの方法を定義**
WS-BrokeredNotification	**通知の生産者と消費者の中間の「通知ブローカ」のWebサービスのインターフェース定義**

OGSIとWSRFの仕様の対比

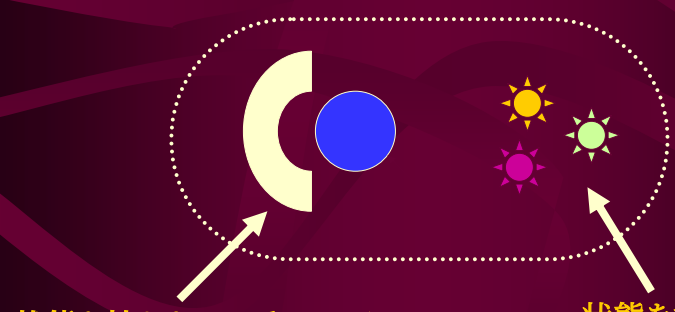
OGSI	WSRF
Grid Service Reference	*WS-Addressing* Endpoint Reference
Grid Service Handle	*WS-Addressing* Endpoint Reference
HandleResolver portType	**WS-RenewableReferences**
Service data defn & access	**WS-ResourceProperties**
GridService lifetime mgmt	**WS-ResourceLifetime**
Notification portTypes	**WS-Notification**
Factory portType	Treated as a pattern
ServiceGroup portTypes	**WS-ServiceGroup**
Base fault type	**WS-BaseFaults**

「状態を持つリソース」をどのように
Webサービスでモデル化するか？



WS-Resourseとは何か?

「状態を持たないWebサービス」と
「状態を持つリソース」を
分離したうえで、組み合わせたもの



状態を持たないWebサービス

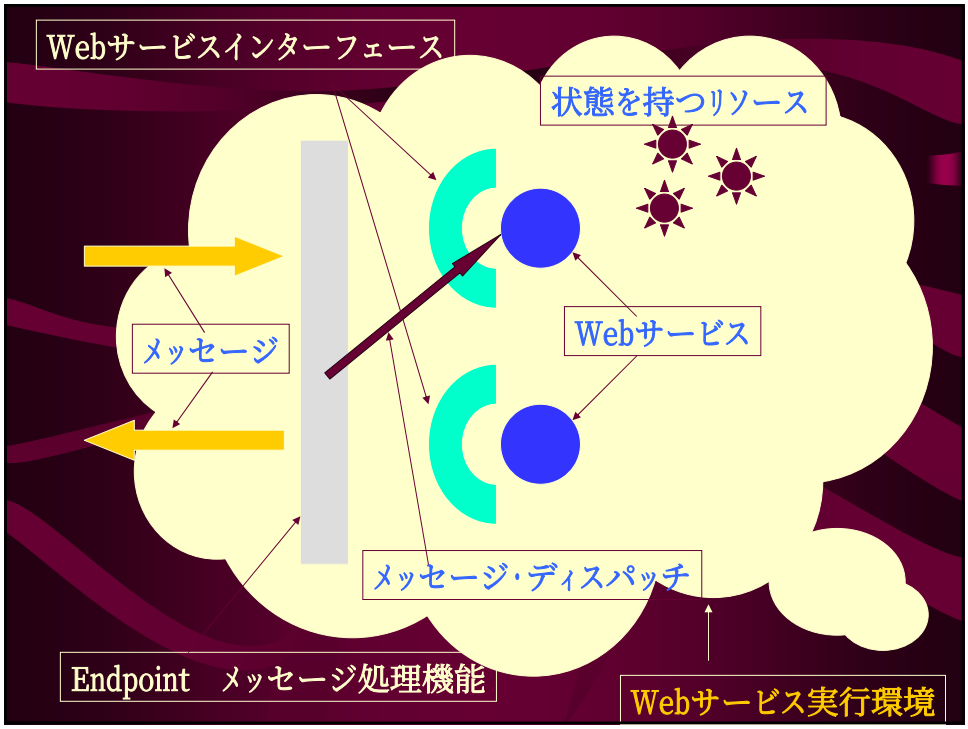
状態を持つリソース

WS-Resourseの「状態」としての WS-Resource Properties Document

WSDLのportTypeでResource Properties Documentoを定義

```
<GenericDiskDriveProperties
  xmlns:tns="http://example.com/….” >
  <tns:NumberOfBlocks>
    22
  </tns:NumberOfBlocks>
  <tns:BlockSize>
    1024
  </tns:BlockSize>
  <tns:Manufacturer>
    DrivesRUs
  </tns:Manufacturer>
</GenericDiskDriveProperties>
```

```
<!-- Association of resource properties document to a portType -->
<wsdl:portType name="GenericDiskDrive"
wsrp:ResourceProperties="tns:GenericDiskDriveProperties" >
```



WS-Addressing

Message Information Headers

```
POST /interop HTTP/1.0
Host: www.whitemesa.net
User-Agent: White Mesa SOAP Interop Client/1.0
Content-Type: text/xml; charset="utf-8"
Content-Length: 502
SOAPAction: "http://soapinterop.org/"

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <SOAP-ENV:Body>
  <m:echoString xmlns:m="http://soapinterop.org/">
   <inputString>hello world</inputString>
  </m:echoString>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAPメッセージにはアドレス情報が含まれていない

通常のWebサービスのリクエスト・メッセージ

```
HTTP/1.0 200 OK
Date: Wed, 20 Jun 2001 02:44:16 GMT
Server: WhiteMesa SOAP Server/2.3
Content-Type: text/xml; charset="utf-8"
Content-Length: 508

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <SOAP-ENV:Body>
  <m:echoStringResponse xmlns:m="http://soapinterop.org/">
   <return>hello world</return>
  </m:echoStringResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAPメッセージにはアドレス情報が含まれていない

通常のWebサービスのレスポンス・メッセージ

WS-Addressingサンプル

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/ soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/ addressing">
  <S:Header>
    <wsa:ReplyTo>
      <wsa:Address> http://business456.com/client1 </wsa:Address>
    </wsa:ReplyTo>
    <wsa:To>http://fabrikam123.com/Purchasing</wsa:To>
    <wsa:Action>http://fabrikam123.com/SubmitPO</wsa:Action>
  </S:Header>
  <S:Body>
    ...
  </S:Body>
</S:Envelope>
```

SOAP-Headerにアドレス情報を埋め込む

Message Information Headers

[宛先](必須)
<wsa:To> xs:anyURI </wsa:To>
[送信元エンドポイント]
<wsa:From> エンドポイントリファレンス </wsa:From>
[応答先エンドポイント]
<wsa:ReplyTo> エンドポイントリファレンス </wsa:ReplyTo>
[障害報告エンドポイント]
<wsa:FaultTo> エンドポイントリファレンス </wsa:FaultTo>
[アクション] (必須)
<wsa:Action> xs:anyURI </wsa:Action>
[メッセージID]
<wsa:MessageID> xs:anyURI </wsa:MessageID>
[関係]
<wsa:RelatesTo RelationshipType="..."> xs:anyURI
</wsa:RelatesTo>

Webサービスの拡大シナリオ
SMTPのサポート

HTTPに深く依存してきたWebサービスだが、
WS-Addressingによって、トランスポート層で、
他のプロトコルの利用も可能となる。

**WS-Addressingを利用すれば、アドレスもactionも
SOAP-Envelopeの中に入るので、ネットワーク上の
ファイアウォールやルータを通りやすく出来る。**

Webサービスの拡大シナリオ

- リクエストとレスポンスの間隔が、非常に長い場合への対応

  沢山のビジネスプロセスが協調しあって、一つのサービスを提供しているような場合、長く処理時間がかかることがありえる。通常のWebサービスの場合には、コネクションがタイムアウトで切れてしまって、レスポンスが返れなくなる

- 一方向(One Way) の通知のメッセージングへWebサービスの利用

  Webサービスで印刷Jobを実装したら、レスポンスに「印刷開始」をいれるか「印刷終了」を入れるか? どちらかをNotificationで実現するのがスマート。

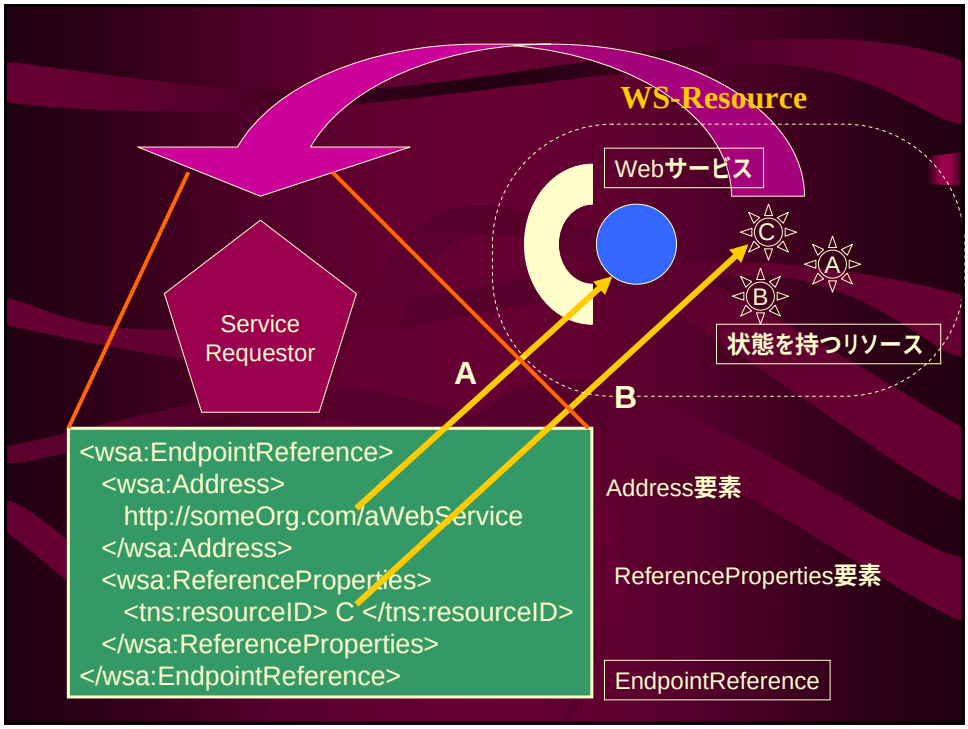
WS-Addressing

Endpoint Reference

EndpointReferenceのサンプル

```
<wsa:EndpointReference>
  <wsa:Address>
    http://someOrg.com/aWebService
  </wsa:Address>
  <wsa:ReferenceProperties>
    <tns:resourceID> C </tns:resourceID>
  </wsa:ReferenceProperties>
</wsa:EndpointReference>
```

EndpointReferenceは、例えば、WS-ResourceのFactoryでcreateServiceが呼び出された時に返されるWS-Resourceへのポインタである。ここでは、WebサービスのアドレスをさすAddress要素と、状態を持つリソースをポイントする、ReferenceProperties要素が見える。



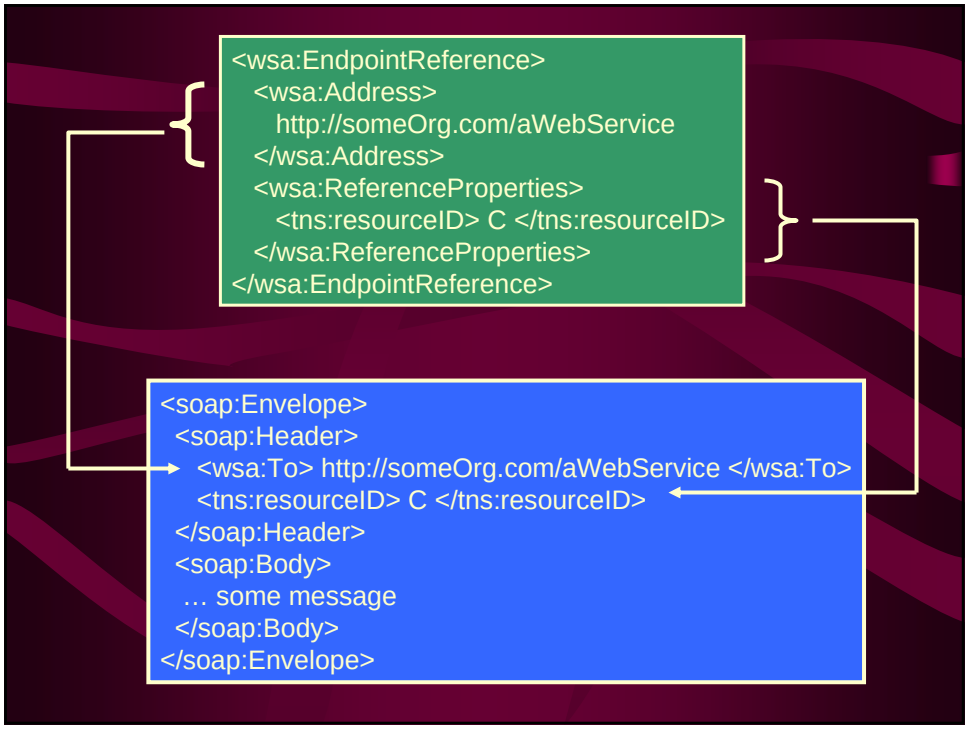
EndpointReferenceの定義

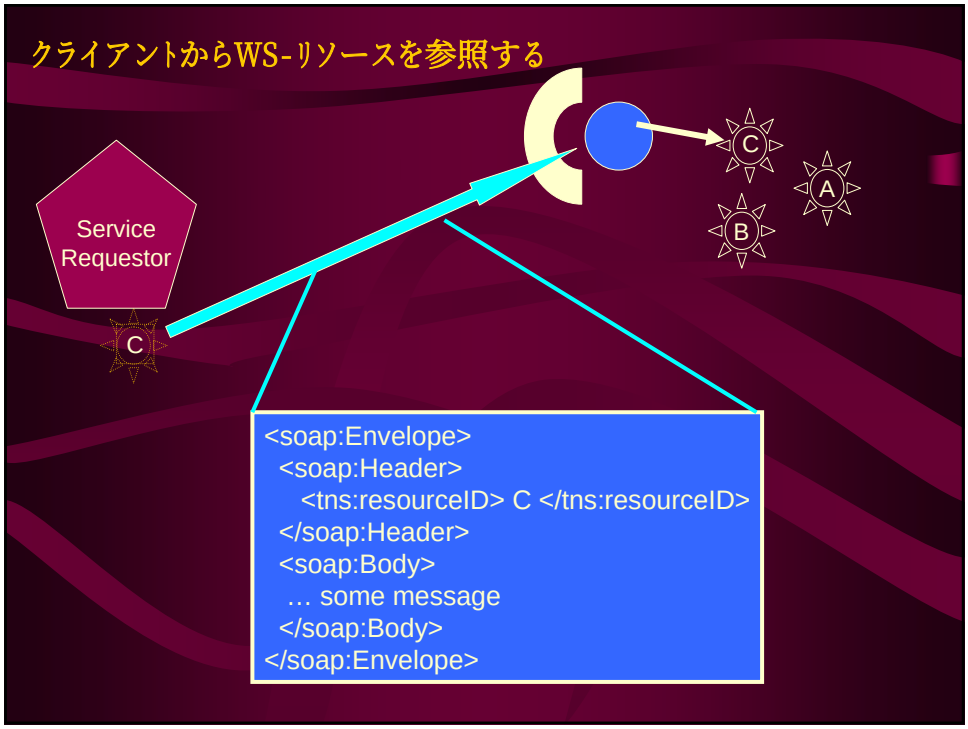
```
<wsa:EndpointReference>
   <wsa:Address>xs:anyURI</wsa:Address>
   <wsa:ReferenceProperties> ...
   </wsa:ReferenceProperties> ?
   <wsa:PortType>xs:QName</wsa:PortType> ?
   <wsa:ServiceName PortName="xs:NCName"?>
      xs:Qname
   </wsa:ServiceName> ?
   <wsp:Policy/> *
</wsa:EndpointReference>
```

EndpointReferenceの SOAPメッセージへのマッピング

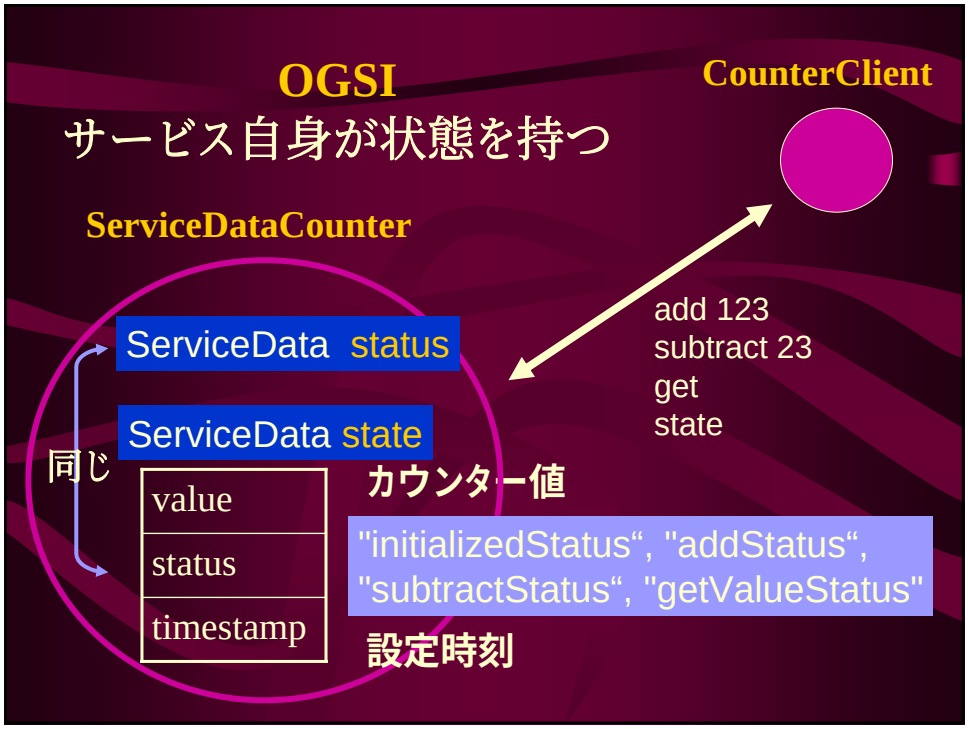
1. エンドポイントリファレンスのwsa:Address情報は、SOAPメッセージの宛先wsa:Toにコピーされる。

2. エンドポイントリファレンスのwsa:ReferenceProperties情報は、SOAPメッセージのヘッダブロックになる。wsa:ReferenceProperties内のそれぞれの要素が、直接、ヘッダーブロックに追加される。




クライアントからWS-リソースを参照する
Service Requestor
C
C
A
B
<soap:Envelope>
<soap:Header>
<tns:resourceID> C </tns:resourceID>
</soap:Header>
<soap:Body>
… some message
</soap:Body>
</soap:Envelope>


WSRF ResourceProperties




```
<gwsdl:portType name="GridService">
    <operation name="findServiceData" .... />
    <operation name="setServiceData" .... />
    <operation name="requestTerminationAfter" .... />
    <operation name="requestTerminationBefore" .... />
    <operation name="destroy" .... />
    <sd:serviceData name="interface"..... />
    <sd:serviceData name="serviceDataName" .... />
    <sd:serviceData name="factoryLocator" .... />
    <sd:serviceData name="gridServiceHandle" .... />
    <sd:serviceData name="gridServiceReference" .... />
    <sd:serviceData name="findServiceDataExtensibility" .... />
    <sd:serviceData name="setServiceDataExtensibility" .... />
    <sd:serviceData name="terminationTime" .... />
    <sd:staticServiceDataValues> .....
    </sd:staticServiceDataValues>
</gwsdl:portType>
```

ServiceDataの定義

```
<xsd:element name="GenericDiskDriveProperties">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="tns:NumberOfBlocks"/>
      <xsd:element ref="tns:BlockSize" />
      <xsd:element ref="tns:Manufacturer" />
      <xsd:any minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
…
<!-- ResourcePropertyDocumentをポートタイプと関連つける -->
<wsdl:portType name="GenericDiskDrive"
   wsrp:ResourceProperties="tns:GenericDiskDriveProperties" >
   <operation name="start" …/>
   <operation name="stop" …/>
    …
</wsdl:portType>
```

ResourcePropertyの定義

ResourcePropertyDocumentサンプル

```
<GenericDiskDriveProperties
    xmlns:tns="http://example.com/diskDrive" >
   <tns:NumberOfBlocks>22</tns:NumberOfBlocks>
   <tns:BlockSize>1024</tns:BlockSize>
   <tns:Manufacturer>DrivesRUs</tns:Manufacturer>
</GenericDiskDriveProperties>
```

WSRF GetResourceProperty

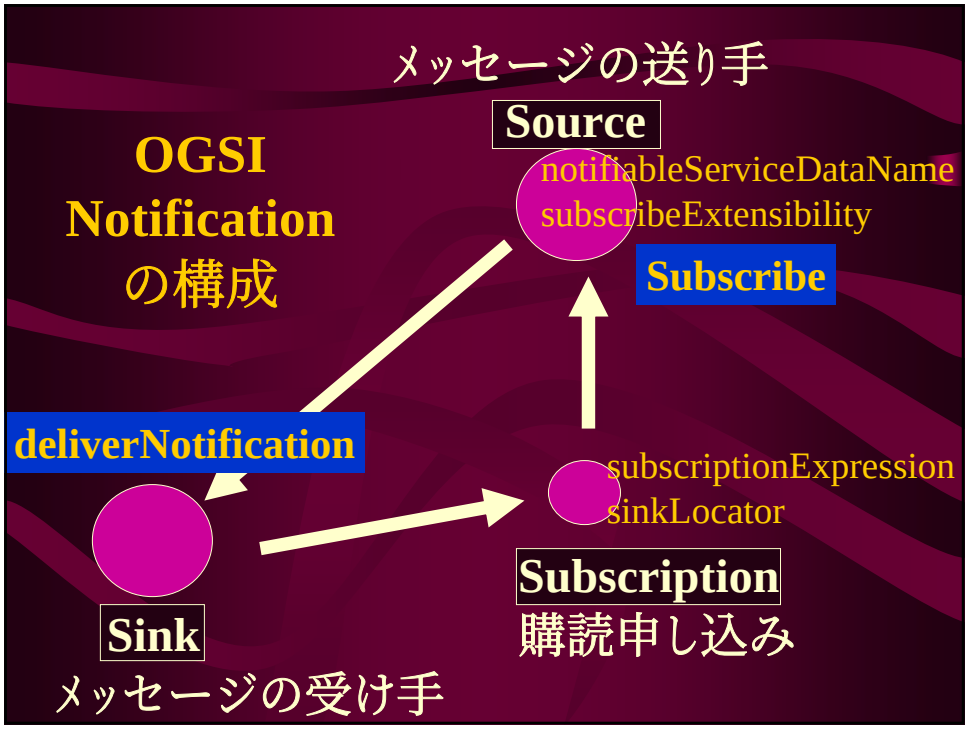
```
<wsrp:GetResourcePropertyRequest
    xmlns:tns="http://example.com/diskdrive">
   tns:NumberOfBlocks
</wsrp:GetResourcePropertyRequest>
```

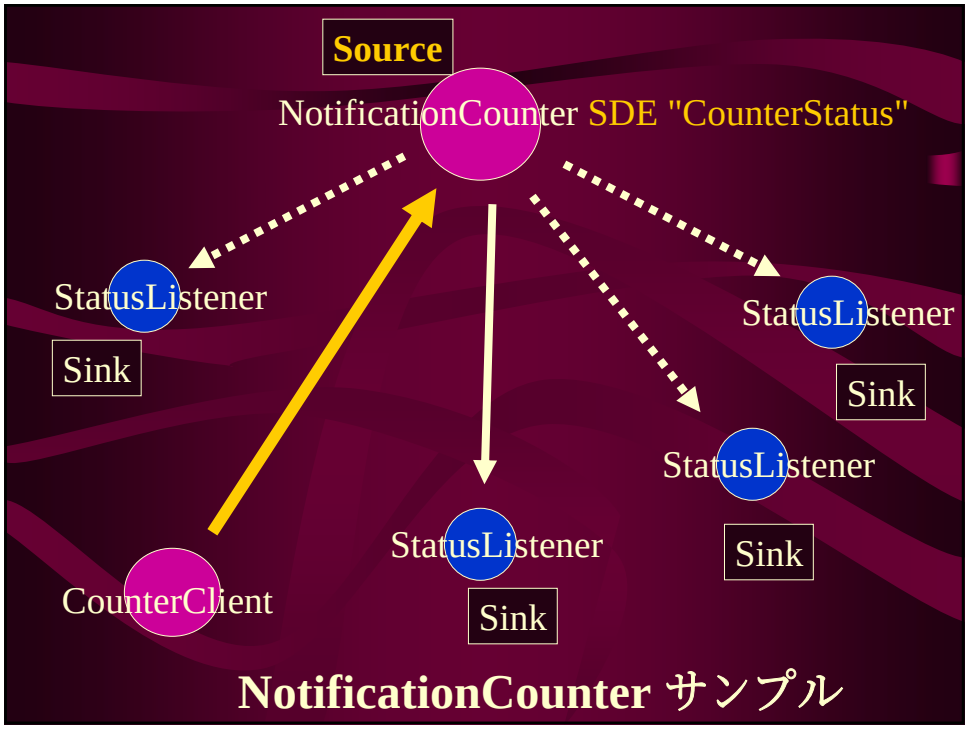
```
<wsrp:GetResourcePropertyResponse
    xmlns:ns1="http://example.com/diskdrive" …>
   <ns1:NumberOfBlocks>22</ns1:NumberOfBlocks>
</wsrp:GetResourcePropertyResponse>
```

WSRF SetResourceProperties

```
<wsrp:SetResourcePropertiesRequest
    xmlns:tns="http://example.com/diskdrive">
  <wsrp:Update resourceProperty="tns:NumberOfBlocks">
    <tns:NumberOfBlocks>143</tns:NumberOfBlocks>
  </wsrp:Update>
  <wsrp:Delete resourceProperty="tns:Manufacturer" />
  <wsrp:Insert>
    <tns:someElement>42</tns:someElement>
  </wsrp:Insert>
</wsrp:SetResourcePropertiesRequest>
```

WS-Notification





WS-Notification PortType Definition

```
<wsdl:portType name="NotificationProducer"
    wsrp:ResourceProperties
  ="wsnt:NotificationProducerRP">

<wsdl:portType name="SubscriptionManager"
    wsrp:ResourceProperties
  ="wsnt:SubscriptionManagerRP">
```

Resource Properties for NotificationProducer

```
<xsd:element name="NotificationProducerRP" >
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="wsnt:Topic"
                    minOccurs="1" maxOccurs="unbounded" />
            <xsd:element ref="wsnt:FixedTopicSet"
                    minOccurs="1" maxOccurs="1" />
            <xsd:element ref="wsnt:TopicExpressionDialects"
                    minOccurs="1" maxOccurs="unbounded" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
```

Resource Properties for SubscriptionManager

```
<!-- SubscriptionManager specific  -->
   <xsd:element ref="wsnt:ConsumerReference"   minOccurs="1"
   <xsd:element ref="wsnt:TopicExpression"     minOccurs="1"
   <xsd:element ref="wsnt:UseNotify"           minOccurs="1"
   <xsd:element ref="wsnt:Precondition"        minOccurs="0"
   <xsd:element ref=“wsnt:Selector             minOccurs="0"
   <xsd:element ref="wsnt:SubscriptionPolicy"  minOccurs="0"
   <xsd:element ref="wsnt:CreationTime"        minOccurs="0"
```

NotificationProducer

Subscribe(
(ConsumerEndpointReference, TopicExpression, [UseNotify],
[Precondition], [Selector], [SubscriptionPolicy],
[InitialTerminationTime])
returns: WS-Resource qualified EPR to a Subscription

GetCurrentMessage(topicExpression)
returns: a NotificationMessage (xsd:any)

NotificationConsumer

Notify(
NotificationMessage
(TopicExpression, ProducerReference, Message)*
returns: n/a (one way)