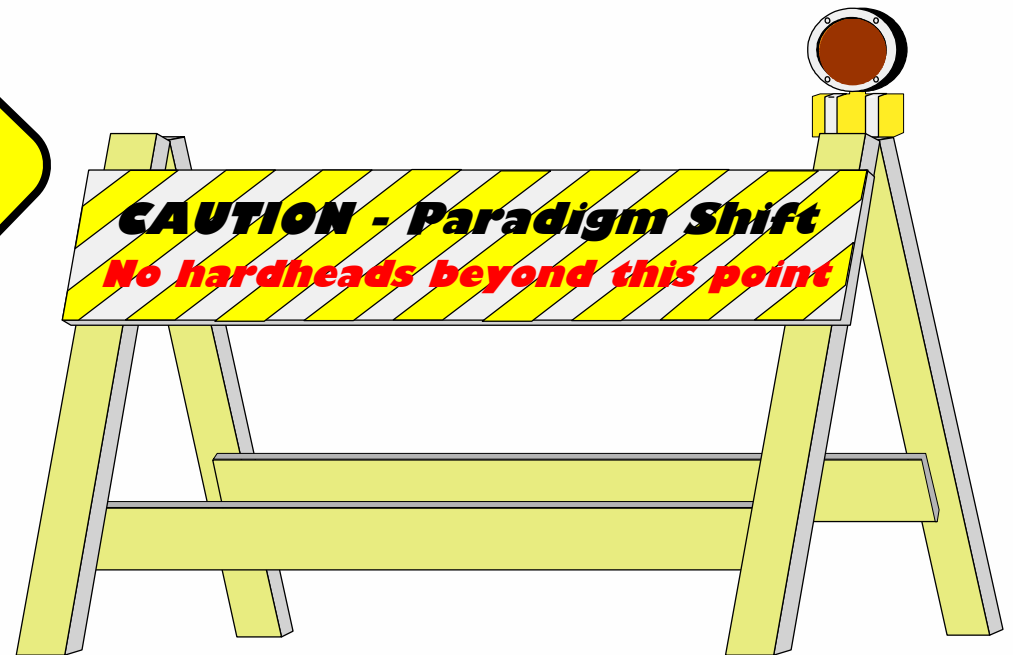
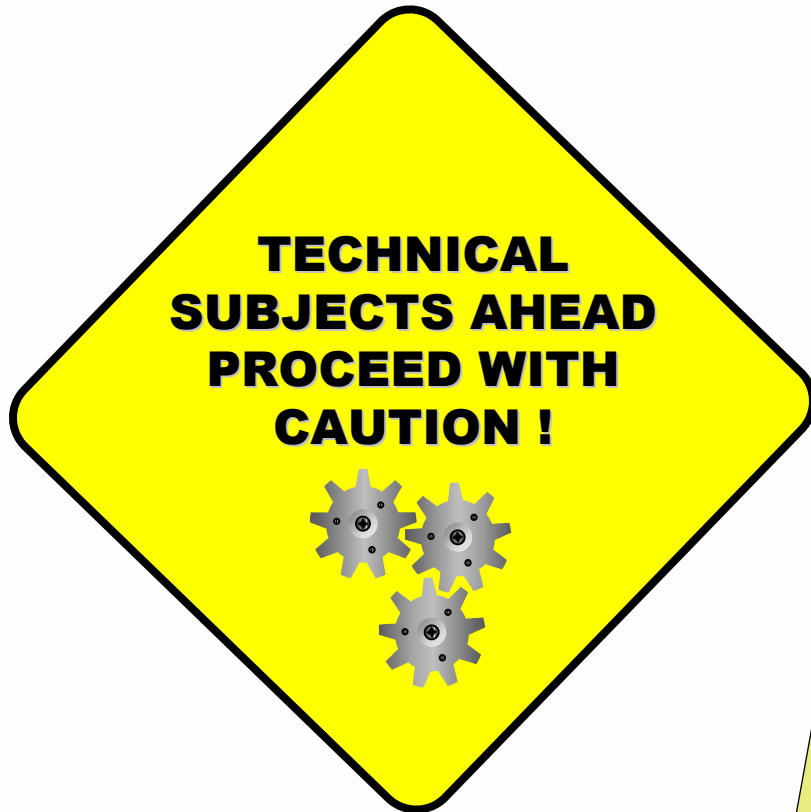


# ***WS-ResourceFramework and WS-Notification Technical Overview***

San Francisco, CA  
January , 2004

Steve Graham (IBM)



# **Introduction and Background**

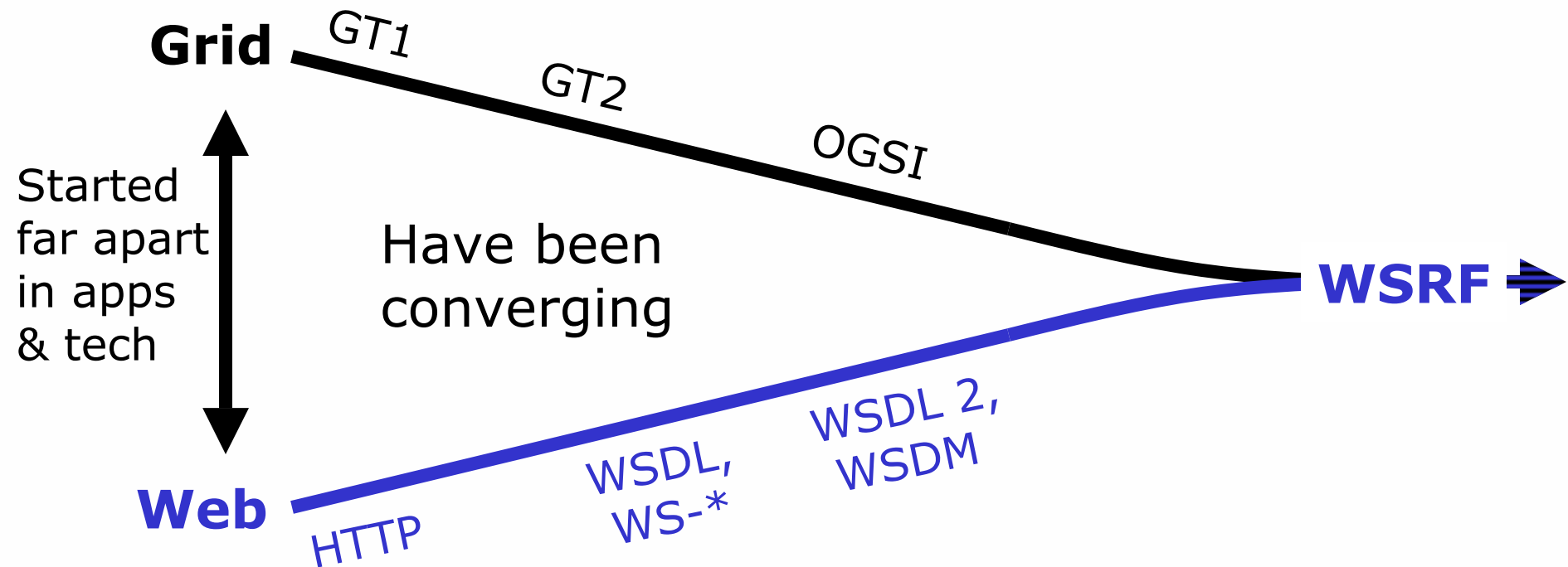
---

## **Context: Open Grid Services Architecture**

- **Define a service-oriented architecture ...**
  - **the key to effective virtualization**
- **... to address vital “Grid” requirements**
  - **AKA utility, on-demand, system management, collaborative computing**
- **... building on Web services standards**
  - **extending those standards where needed**

# Introduction and Background

## Grid and Web Services: Convergence



The definition of WSRF means that Grid and Web communities can move forward on a common base

# Introduction and Background

## Concerns Addressed

- Too much stuff in one specification
  - ➔ WSRF partitions OGSI v1.0 functionality into a family of composable specifications
- Does not work well with existing Web services tooling
  - ➔ WSRF tones down the usage of XML Schema
- Too object oriented
  - ➔ WSRF makes an explicit distinction between the “service” and the stateful “resources” acted upon by that service

# What is a Web Service ?

---

- An operation execution component made available at an endpoint address
  - A service is defined in terms of the operations it implements
  - An operation is defined in terms of a message exchange
  - The supported set of messages exchanges (operations) implemented by a service may be described as a WSDL portType (The Web service interface definition)
  - The Web service itself is typically stateless
- Accessible through use of a WS-Addressing Endpoint Reference
- Lifecycle of a Web service typically described in terms of “deployment”
- Service interface definitions often imply the existence of stateful resources that are used and manipulated in the processing of a Web service request message

# What do we mean by Stateful Resource ?

---

- A specific set of state data expressible as an XML document that defines the type of the resource;
- Having a well-defined identity and lifecycle; and
- Known to, and acted upon, by one or more Web services.
- Many possible implementations
  - Files, Database tables, EJB Entities, XML documents, Composed from multiple data sources, etc.
- Lifecycle expressed in terms of resource creation and destruction
  - Identity is assigned at creation time

# WS-Addressing

---

- Standardizes the representation of the address of a Web service deployed at a given network endpoint
- A WS-Addressing endpoint reference is an XML serialization of a network-wide pointer to a Web service
- EPRs can be used to pass services to other services by reference
- An EPR contains:
  - Service address (wsa:Address)
  - Metadata associated with the Web service such as service description information
  - Policy information related to the use of the service
  - Reference properties, which can be used to define a contextual use of the endpoint reference (wsa: ReferenceProperties)



# The WS-Resource framework model

## *Web Service*



Run-time environment

# The WS-Resource framework model

## *Invoking a Web Service*



# Implied Resource Pattern

---

- A specific kind of relationship between a Web service and a stateful resource
- Used to associate a stateful resource with the execution of message exchanges implemented by a Web service
- The stateful resource associated with a given message exchange is treated as implicit execution context for the message request
- By implicit, we mean to say that the requestor does not provide the identity of the resource as an explicit parameter in the body of the request message
- The context used to designate the implied stateful resource is encapsulated in the WS-Addressing endpoint reference used to address the target Web service at its endpoint (Use of EPR Reference Properties).

# WS-Resource

---

- When a stateful resource participates in the implied resource pattern, we refer to it as a WS-Resource
- The wsa:Address component refers to the network transport-specific address of the Web service (often a URL in the case of HTTP-based transports)
- The wsa:ReferenceProperties component contains an XML serialization of the WS-Resource identity, as understood by the Web service addressed by the endpoint reference
- The WS-Resource identity represents the WS-Resource to be used in the execution of the request message
- The set of reference properties used to hold the WS-Resource identity within the endpoint reference is referred to as the WS-Resource context.
- An endpoint reference containing a WS-Resource context is a WS-Resource-qualified endpoint reference.

## WS-Resource (Continued)

- The content of the WS-Resource context is opaque to the service requestor
- The service requestor's applications should not examine or attempt to interpret the contents of the WS-Resource context
- The WS-Resource context is meaningful only to the Web service, and is used by the Web service in an implementation specific way to identify the WS-Resource needed for the execution of the request message
- From the point of view of the service requestor:
  - the WS-Resource qualified endpoint reference represents the pointer to the Web service that has been further constrained to execute its message exchanges within the context of a specific WS-Resource
  - Or, the WS-Resource qualified endpoint reference represents the pointer to a WS-Resource accessible through the message exchanges implemented by the associated Web service

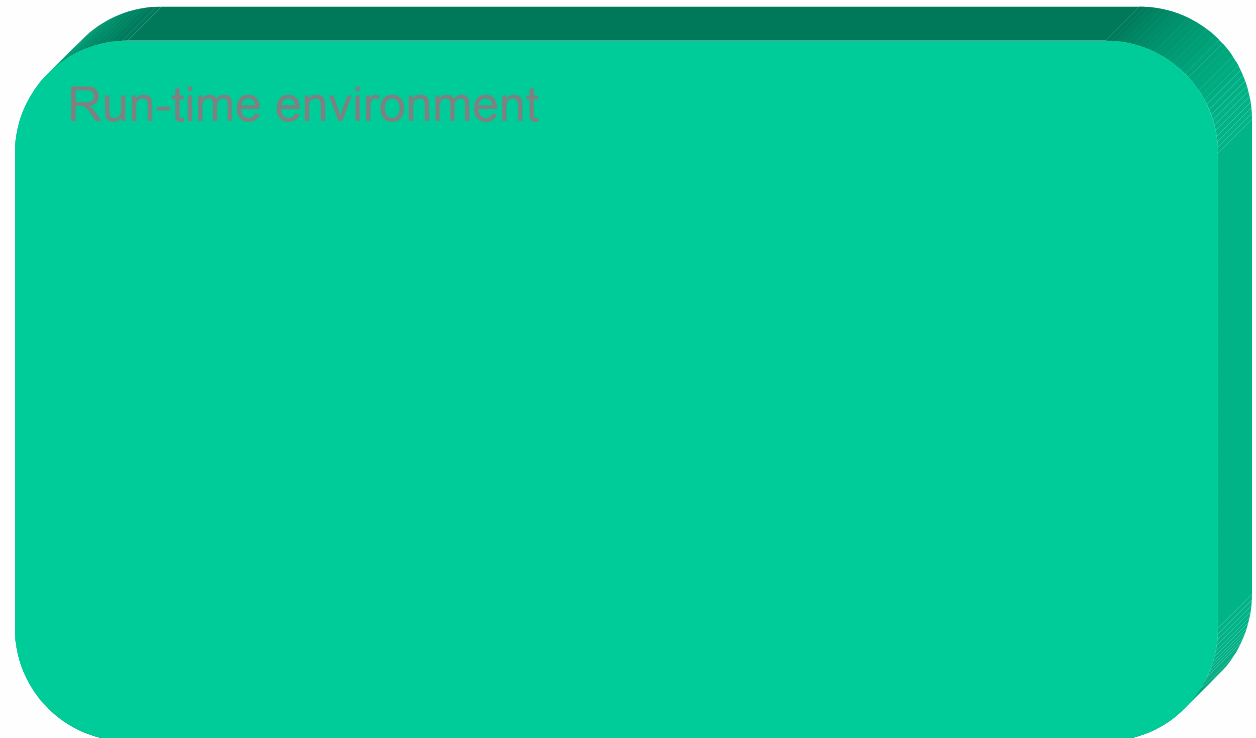
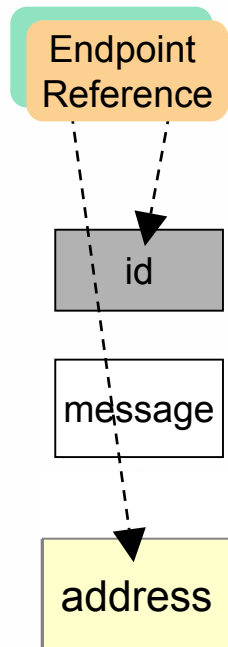
# The WS-Resource framework model

*Using a Web service to access a WS-Resource*



# The WS-Resource framework model

*Using a Web service to access another WS-Resource*



# WS-Resource Factory

---

- Any Web service capable of bringing a WS-Resource into existence and assigning the new WS-Resource an identity
- The response message of a WS-Resource factory operation must contain a WS-Resource-qualified endpoint reference containing a WS-Resource context that refers to the new WS-Resource
- Note also that what we refer to here as a WS-Resource factory is a use pattern for Web services, not a single standard operation. This use pattern may be encoded in a variety of different Web service operations that may, for example, create one or many WS-Resources



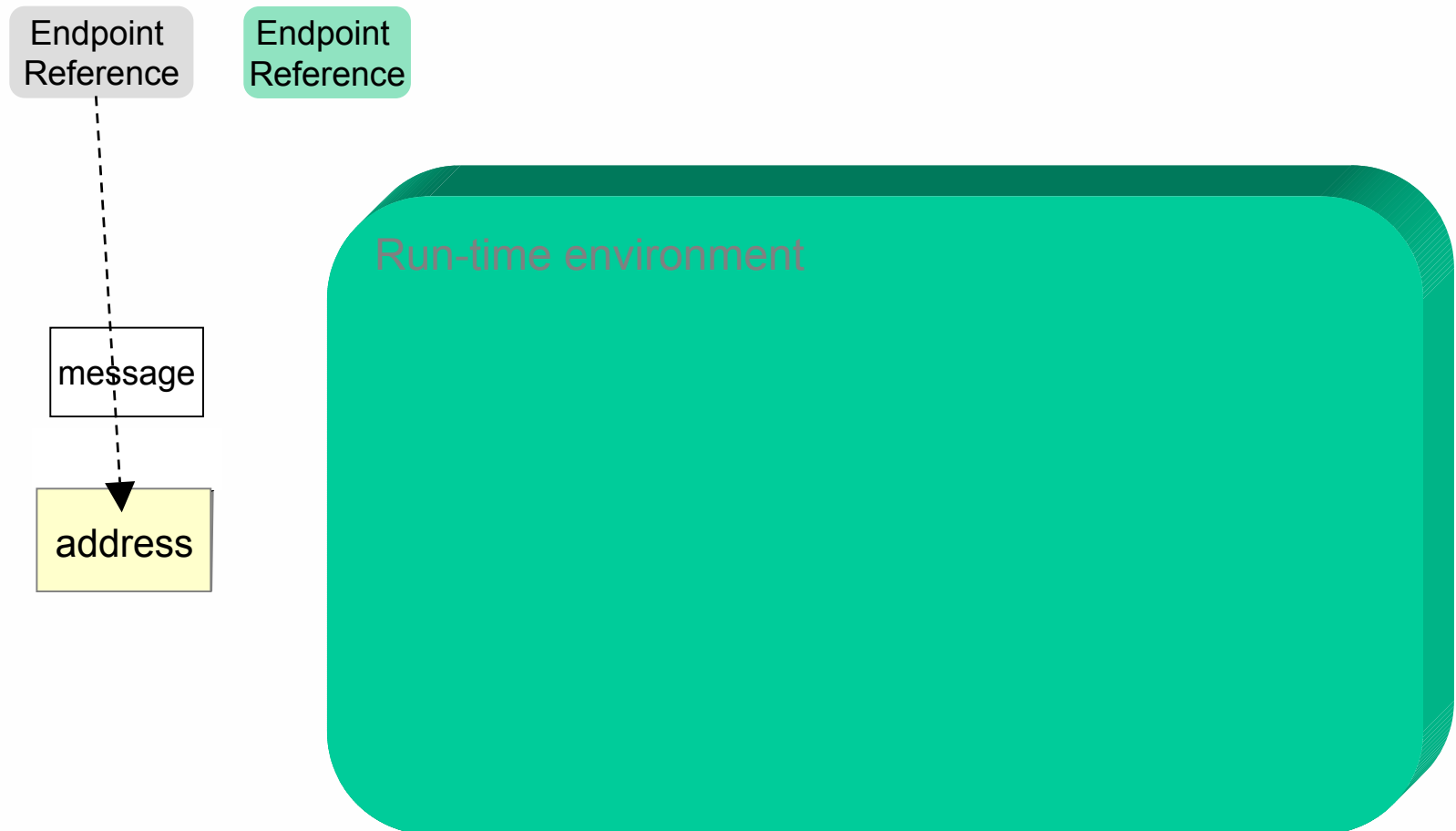
# The WS-Resource framework model

## *Creating / Locating a WS-Resource*



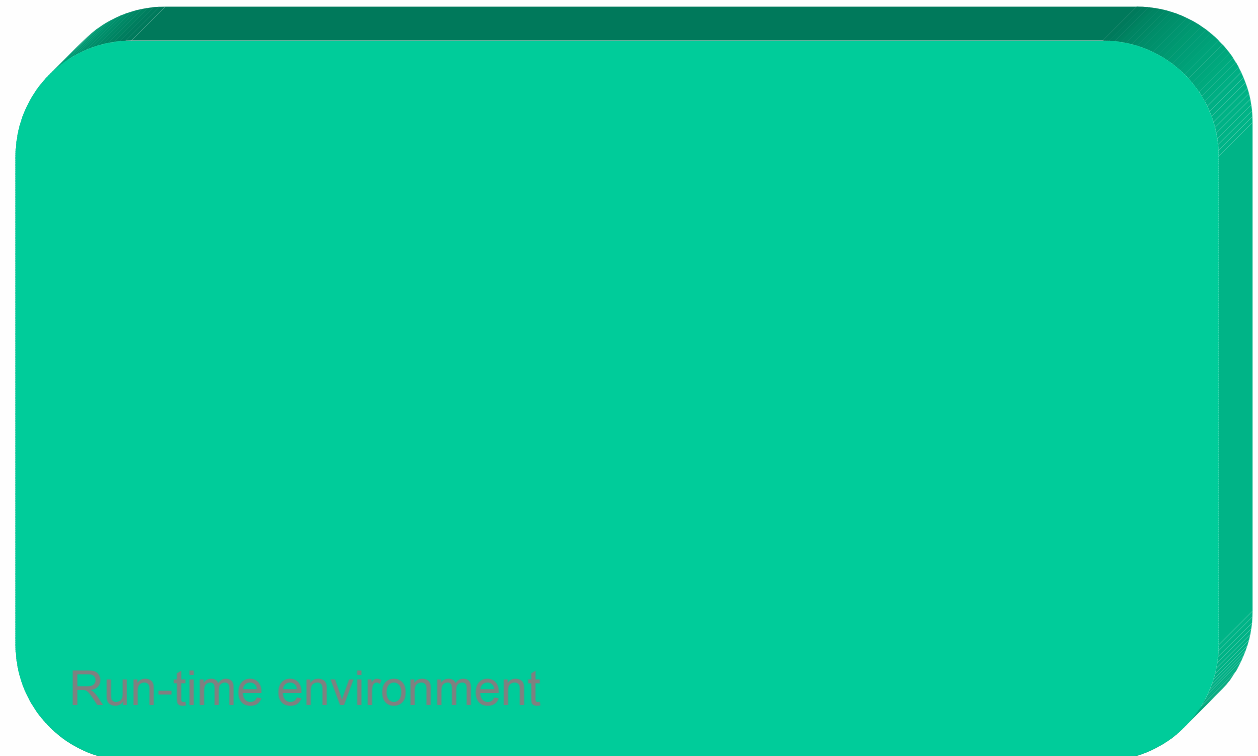
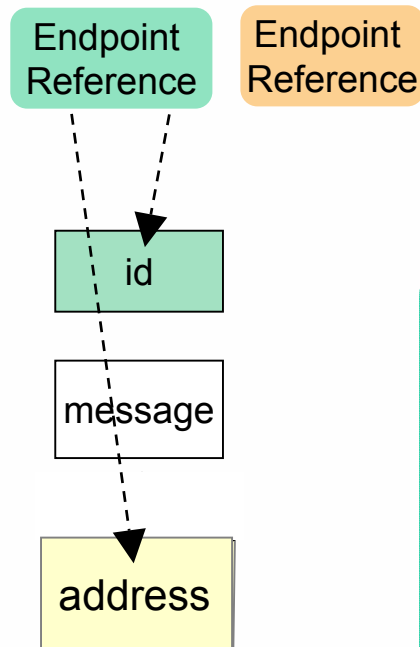
# The WS-Resource framework model

## *Creating / Locating another WS-Resource*



# The WS-Resource framework model

## *Passing a WS-Resource as additional context*



# WS-Resource Relationship Cardinality

---

- A Web service can execute message exchanges against zero or more WS-Resources of a given type
- At the type level, a WSDL 1.1 portType, defining the interface to a Web service, can be associated with at most one type of WS-Resource
- One type of WS-Resource can be associated with many WSDL 1.1 portTypes

# WS-Resource and ACID Properties

- In the presence of a transactional unit of work, a Web service capable of participating in the transactional protocol must abide by the rules of two-phase-commit transaction management. However, in the absence of a transaction management policy, the Web service is under no obligation to recover the state of the WS-Resource in the event of a failure
- The WS-Resource definition is not prescriptive with respect to policy that governs concurrent read or write access to a WS-Resource through a Web service. The definition of specific policy governing concurrent updates, whether or not separate message executions targeting the same WS-Resource may be interleaved, and whether partially completed WS-Resource updates within a given message execution may be observed by other concurrent requests is beyond the scope of the WS-Resource definition
- If WS-Resource isolation is needed, we suggest the use of a transaction to provide a context within which isolation of WS-Resource updates can be provided
- In the absence of a transactional unit of work, the level of WS-Resource update atomicity, recovery, isolation, and durability provided by a Web service is implementation dependent

# WS-Resource Security

- In the presence of a valid security context associated with a message exchange, a Web service capable of participating in the expressed security protocols must implement and enforce the security policies implied by the security context
- In the absence of such security policy, the Web service is under no obligation to secure the execution of the message exchange nor the state of the WS-Resource designated by the WS-Resource context associated with the message request
- The WS-Resource framework is not prescriptive with respect to policy that governs access permission to a WS-Resource through a Web service. The definition of specific security policy governing access to the WS-Resource is beyond the scope of the WS-Resource definition
- If WS-Resource access control is required, we suggest the use of the functions such as those defined in the WS-Security specifications to provide a security context for the WS-Resource
- In the absence of a valid security context and associated access control policies, the extent to which the Web service provides security of the WS-Resource is implementation dependent.

# WS-ResourceProperties

- Operations and meta data associated with elements of a resource's state
- Resource Properties document
  - Presented via Web service as an XML document

```
<GenericDiskDriveProperties xmlns:tns="http://example.com/diskDrive" >  
  <tns:NumberOfBlocks>22</tns:NumberOfBlocks>  
  <tns:BlockSize>1024</tns:BlockSize>  
  <tns:Manufacturer>DrivesRUs</tns:Manufacturer>  
</GenericDiskDriveProperties>
```
  - Modelled using standard XML Schema
- PortType declares association between Web service and resource properties document
  - Information is available at design time, as part of the interface
  - Use `xsd:ref` to mix in resource properties from multiple interfaces

# WS-ResourceProperties

- Resource Properties operations

- Get

```
<wsrp:GetResourcePropertyRequest>  
  QName  
</wsrp:GetResourcePropertyRequest>
```

- Get Multiple

```
<wsrp:GetMultipleResourcePropertiesRequest>  
  QName *  
</wsrp:GetMultipleResourcePropertiesRequest>
```

- Query

```
<wsrp:QueryResourcePropertiesRequest>  
  <wsrp:QueryExpression dialect="URI">  
    xsd:any  
  </wsrp:QueryExpression>  
</wsrp:QueryResourcePropertiesRequest>
```



# WS-ResourceProperties

- Resource Properties operations (con't)
- Set

```
<wsrp:SetResourcePropertiesRequest>
{
  <wsrp:Insert >
    xsd:any
  </wsrp:Insert> |

  <wsrp:Update ResourceProperty="QName">
    xsd:any
  </wsrp:Update> |

  <wsrp>Delete ResourceProperty="QName" />
}+
</wsrp:SetResourcePropertiesRequest>
```

# WS-ResourceLifetime

- **Immediate Destruction**

```
<wsrl:DestroyRequest />
```

- **Scheduled Destruction**

```
<wsrl:SetTerminationTimeRequest>  
  <wsrl:RequestedTerminationTime>  
    xsd:dateTime  
  </wsrl:RequestedTerminationTime>  
</wsrl:SetTerminationTimeRequest>
```

- **Resource Properties**

- **Current Time**
- **Termination Time**

- **Initial Termination Time**

# WS-ResourceLifetime

## ■ Resource Destruction Notification

```
<wsnt:topicSpace name="ResourceLifetime"
  targetNamespace=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceLifetime"
... >
<wsnt:topic name="ResourceTermination">
```

## ■ Suggested Contents

```
<wsrl:TerminationNotification>
  <wsrl:TerminationTime>xsd:dateTime</wsrl:TerminationTime>
  <wsrl:TerminationReason>xsd:any</wsrl:TerminationReason>?
</wsrl:TerminationNotification>
```

# WS-Notification

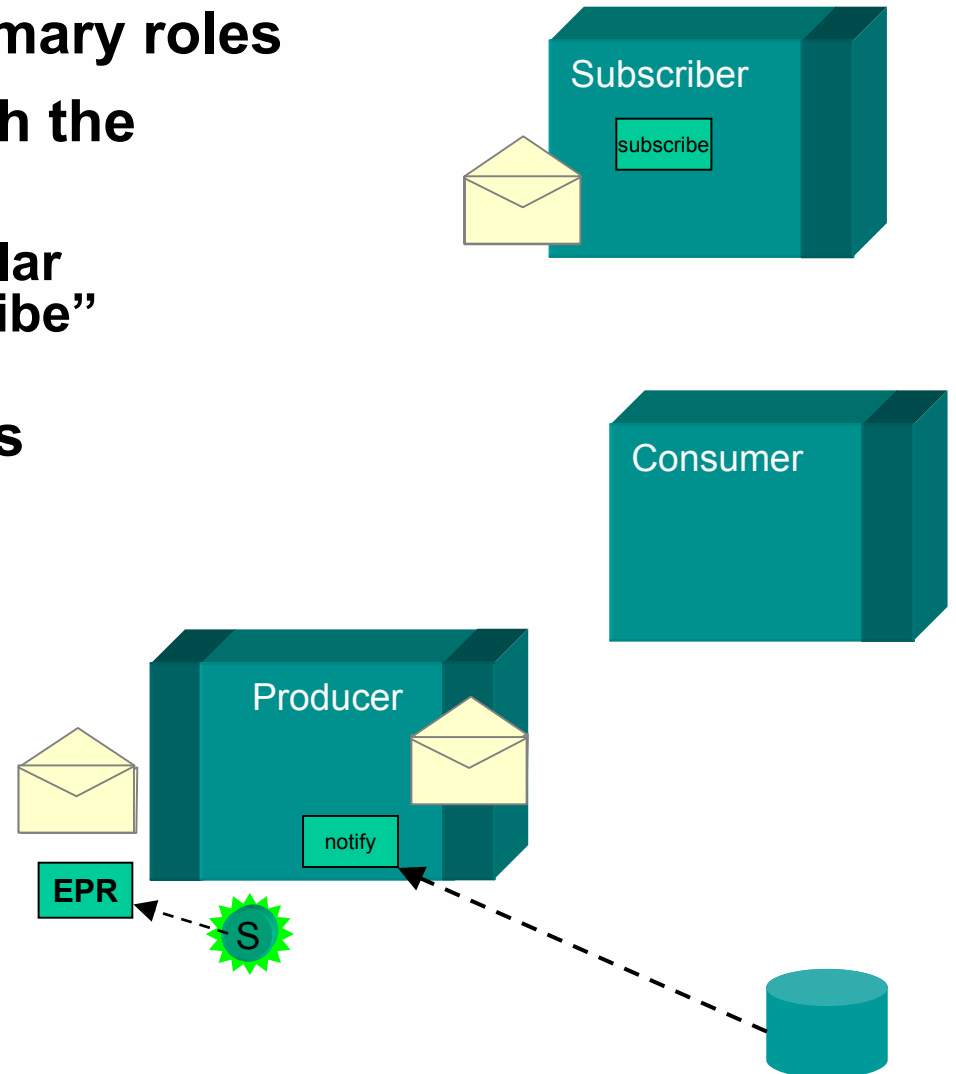
---

## ■ WS-Notification

- Brings enterprise quality publish and subscribe messaging to Web services
  - Loosely coupled, asynchronous messaging in a Web services context
- WS Notification exploit WS Resource framework and Web services technologies
- Direct and Brokered notification
- Topics and Topic Spaces
- More on subscribe
- Other WS-Notification concepts

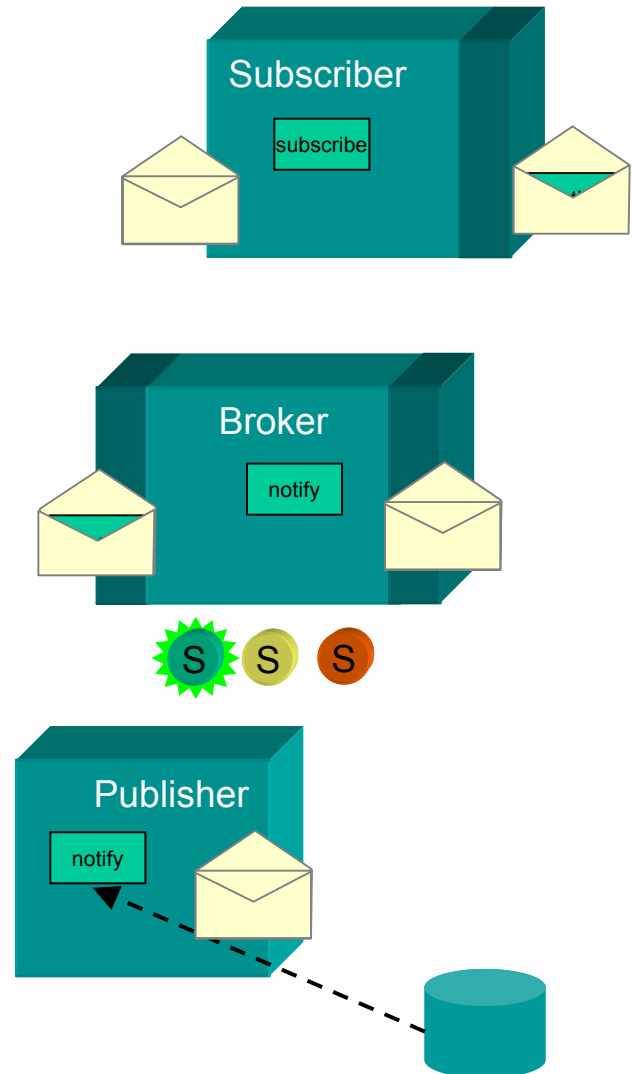
# WS-Notification

- **Direct notification: Three primary roles**
- **Subscriber deals directly with the producer of the Notifications**
  - indicates interest in a particular “Topic” by issuing a “subscribe” request
- **An EPR to the subscription is returned**
- **Producer is responsible for detecting situation and creating the notification**
- **Subscriptions that match receive the notification**



# WS-Notification

- Subscriber indicates interest in a particular “Topic” by issuing a “subscribe” request
- Broker (intermediary) permits decoupling Publisher and Subscriber
- “Subscriptions” are WS-Resources
  - Various subscriptions are possible
- Publisher need NOT be a Web Service
- Notification may be “triggered” by:
  - WS Resource Property value changes
  - Other “situations”
- Broker examines current subscriptions
- Brokers may
  - “Transform” or “interpret” topics
  - Federate to provide scalability



# WS-Notification

- **More on Subscribe Request**

```
<wsnt:SubscribeRequest>  
  <wsnt:ConsumerReference>EPR </wsnt: ConsumerReference>  
  <wsnt:TopicPathExpression />  
  <wsnt:UseNotify> xsd:boolean </wsnt:UseNotify>?  
  <wsnt:Precondition> wsrp:QueryExpression </Precondition>?  
  <wsnt:Selector> wsrp:QueryExpression </wsnt:Selector>?  
  <wsnt:SubscriptionPolicy> wsp:Policy </wsnt:SubscriptionPolicy>?  
  <wsrl:InitialTerminationTime> xsd:dateTime</wsrl:InitialTerminationTime>?  
</wsnt: SubscribeRequest>
```

- **Returns EPR to a Subscription WS-Resource**

# WS-Notification

## ■ Topics and Topic Spaces

- Meta-data to help
  - Organize Notifications
  - Tell the Subscriber what to subscribe to

```
<?xml version="1.0" encoding="UTF-8"?>
<wsnt:topicSpace name="TopicSpaceExample1"
  targetNamespace="http://example.org/topicSpace/example1"
  ... >
  <wsnt:topic name="t1">
    <wsnt:topic name="t2" messageTypes="tns:m1 tns:m2"/>
    <wsnt:topic name="t3" messageTypes="tns:m3"/>
  </wsnt:topic>
  <wsnt:topic name="t4">
    <wsnt:topic name="t5" messageTypes="tns:m3"/>
    <wsnt:topic name="t6" aliasRef="tns:t1/t3"/>
  </wsnt:topic>
</wsnt:topicSpace>
```



# WS-ResourceProperties

- Subscription for Value Change
- Rules for mapping resource properties to Topics
  - QName of resource property corresponds QName of Topic
- To subscribe to changes in `tns:NumberOfBlocks`:

```
<wsnt:SubscribeRequest>  
  <wsnt:ConsumerReference>...  
  
  <wsnt:TopicPathExpression >  
    tns:NumberOfBlocks  
  </wsnt:TopicPathExpression>  
  ...  
</wsnt: SubscribeRequest>
```

- Suggested format of value change message

# Completing the WS-ResourceFramework

---

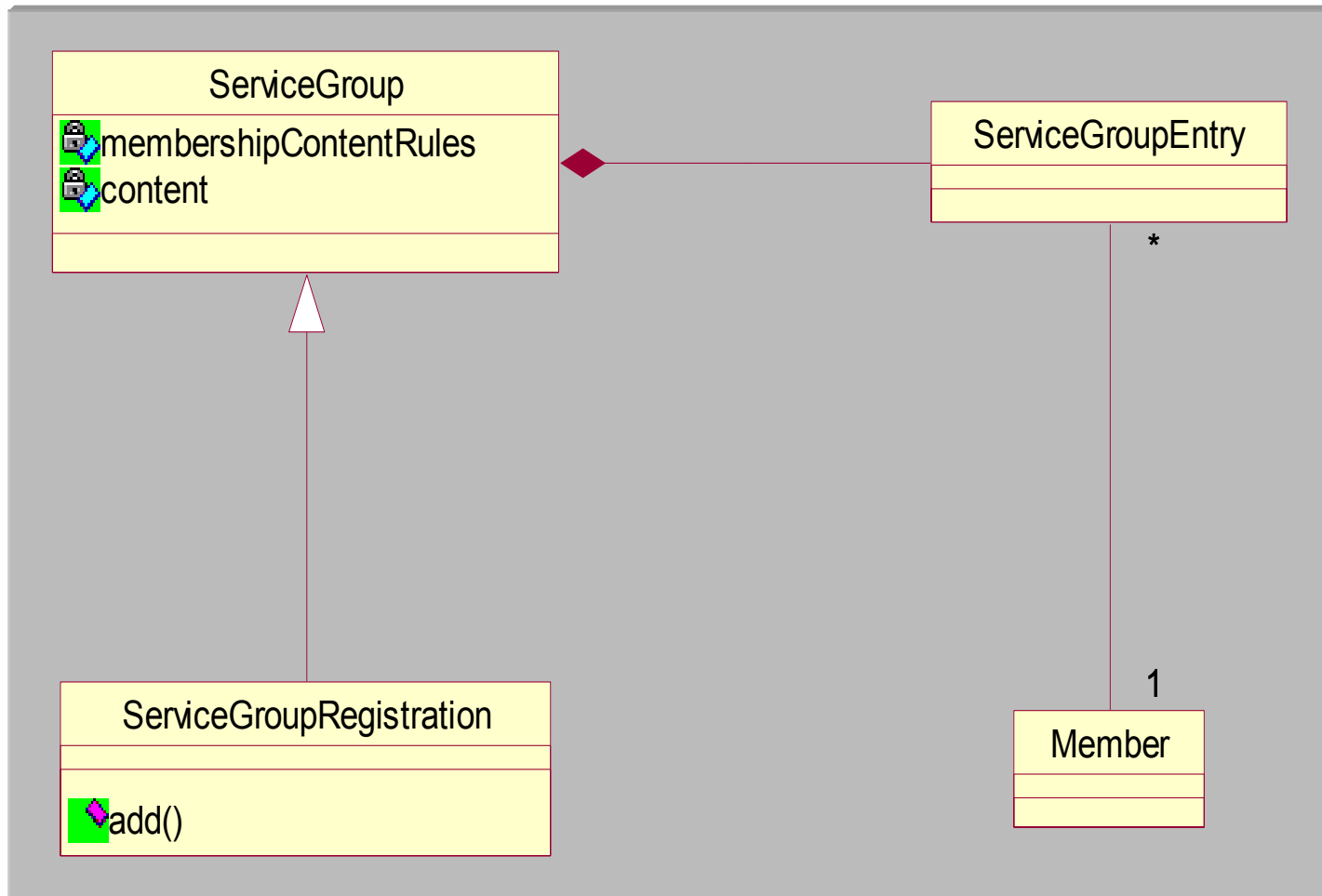
## ■ WS-ServiceGroup

- A Web service that maintains information about a group of other Web services or WS-Resources.
- Services may be members of a group for a specific reason, such as being part of a federated service

or

- they may have no specific relationship, such as the services contained in an index or registry operated for discovery purposes.

# Completing the WS-ResourceFramework



# Completing the WS-ResourceFramework

## ■ WS-RenewableReferences

- Adjunct to the WS-Addressing specification
- Brings enterprise quality to Endpoint References
- Not a general purpose service naming capability
- Multiple, optional ReferenceResolver EPRs
  - Uses WS-Policy element of the EPR to hold ReferenceResolver EPRs
  - Allows transparency to the client programming model by hiding resolution function in the reference proxy
- “Handle” is encoded as reference properties of the ReferenceResolver EPR
- Renewal request includes original EPR as parameter

# Completing the WS-ResourceFramework

```
<wsa:EndpointReference>
  <wsa:Address>xs:anyURI</wsa:Address>
  <wsa:ReferenceProperties/>
  <wsa:PortType>xs:QName</wsa:PortType> ?
  <wsp:Policy>
    <wsrr:Renewable>
      <wsrr:ReferenceResolver>
        wsa:EndpointReference
      </wsrr:ReferenceResolver>
    </wsrr:Renewable>
  </wsp:Policy>
</wsa:EndpointReference>
```

# Completing the WS-ResourceFramework

---

- **Endpoint References provide**
  - **A reference to the WS-Resource**
  - **A mechanism for renewing that reference**
  - **A collection of alternative addresses for a WS-Resource**

**In other words an endpoint reference logically becomes the Locator**

# Completing the WS-ResourceFramework

---

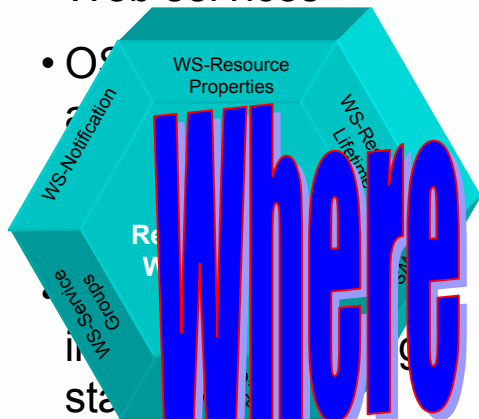
## ■ WS-BaseFaults

- Similar to OGSi v1.0 common fault definition
- Add structure to WSDL error messages
- Define mapping to SOAP 1.2 faults

# How these proposals relate to OGSA

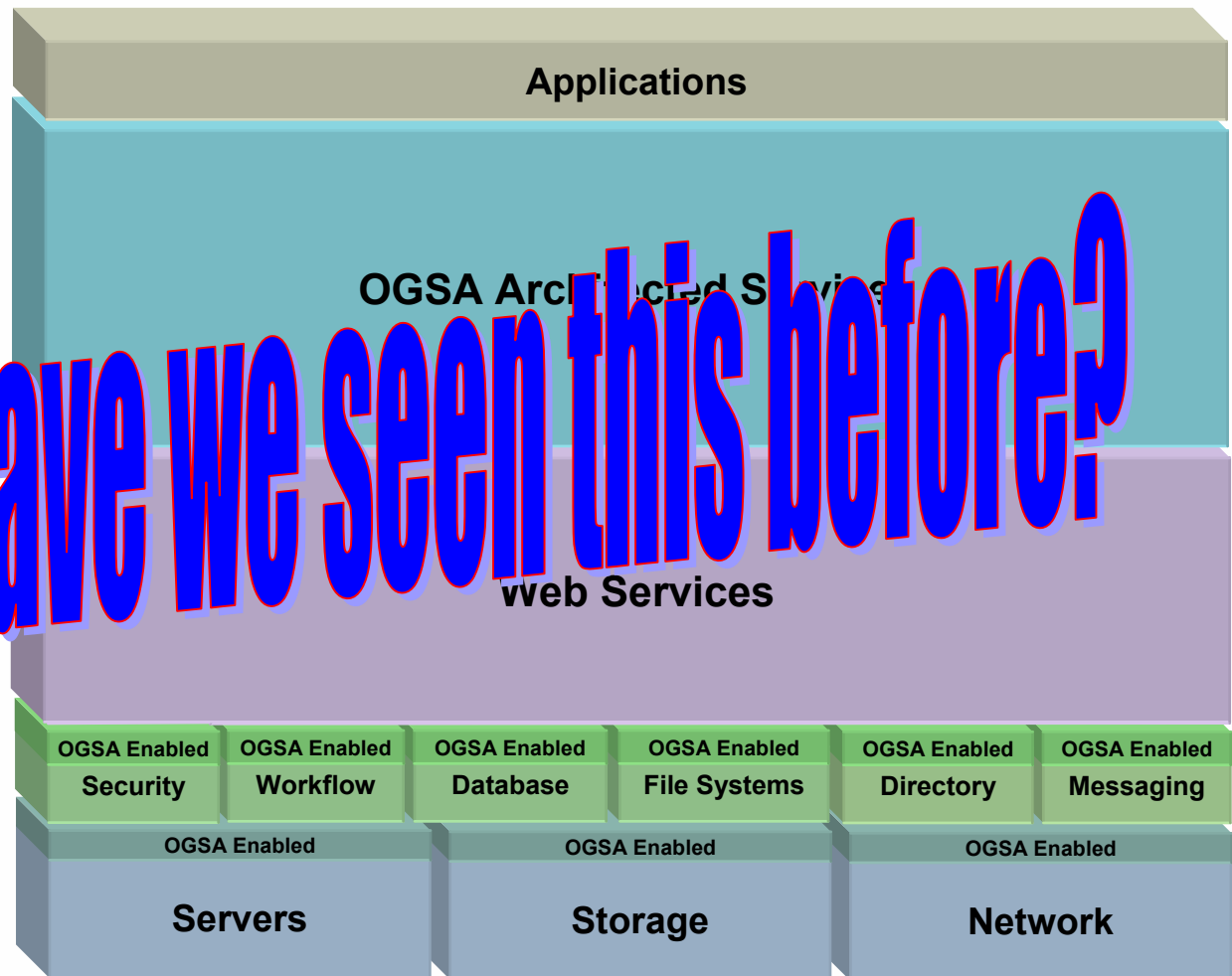
*WS-Resource Framework & WS-Notification are an evolution of OGSI*

- OGSA Services can be defined and implemented as Web services



- OGSA services development tools

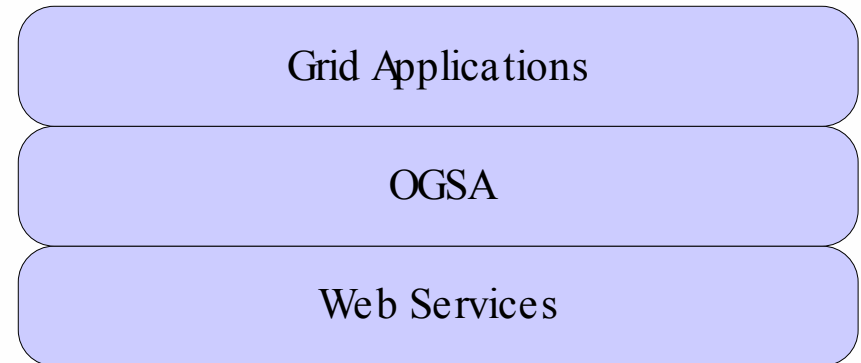
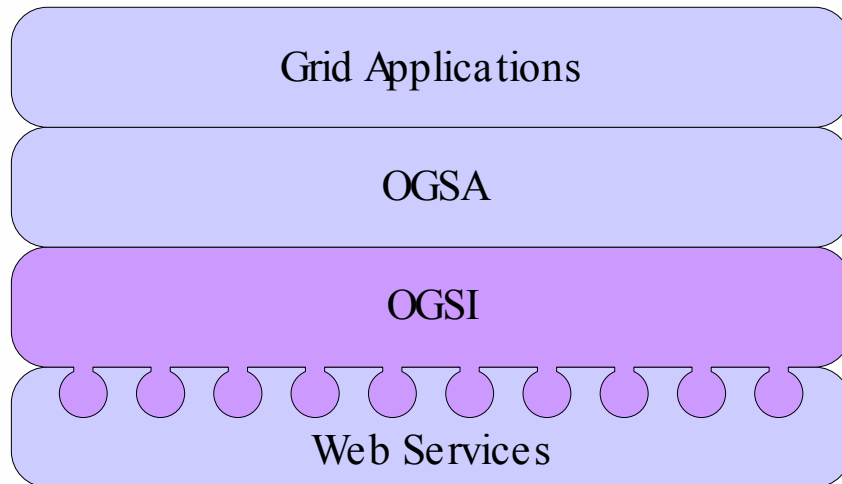
- Grid applications will NOT require special Web services infrastructure





# How these proposals relate to OGSA

## From GGF9: Savas Parastatidis



# How these proposals relate to OGSA

## Head to Head Lineup

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

# How these proposals relate to OGSA

---

## WS-Addressing Endpoint References

### ■ Two Important Parts

- Address - same as conventional endpoint
  - Effectively the GSR
- ReferenceProperties - go into SOAP header

### ■ Reference Properties

- ResourceId in examples, but name not specified.
- Could serve the GSH naming function
- Could be a
  - UUID
  - WSDM: Managed Object Id (MOID)

# How these proposals relate to OGSA

## WS Renewable References

- Not Published Yet
- In OGSi:
  - Each GSH scheme defines how to find a resolver
  - Returns one or more references
  - Knows to ignore ones you already know about
    - (if you tell it about them)
- WS Renewable References
  - All we have in WSRF is Endpoint References.
  - Therefore, get a new Endpoint Reference from an old one.
  - Some imagination needed for a while.

# **How these proposals relate to OGSA**

---

## **Resource Properties & Notification**

- **Publication**
  - Schema in WSDL
- **Query Support**
  - By name and by XPath
- **Notification**
  - Some minor differences in implementation strategy
- **Attributes**
  - Not specified in WSRF
  - Could be added as an OGSA level specification

# How these proposals relate to OGSA

## Lifetime, Faults, Factories, & Service Groups

- **Lifetime**
  - Terminate Before removed
- **Faults**
  - May be simpler, but watch this space
  - Are required by whole community
- **Factory**
  - Pattern rather than portType,
  - Consistent with OGSI experience
- **Service Groups**
  - As it's name implies, this should be very familiar

# Questions

---