Web Services Resource Framework– WSRF

Richard Hopkins
National e-Science Centre, Edinburgh
February 23 / 24 2005

Information Society

OUTLINE

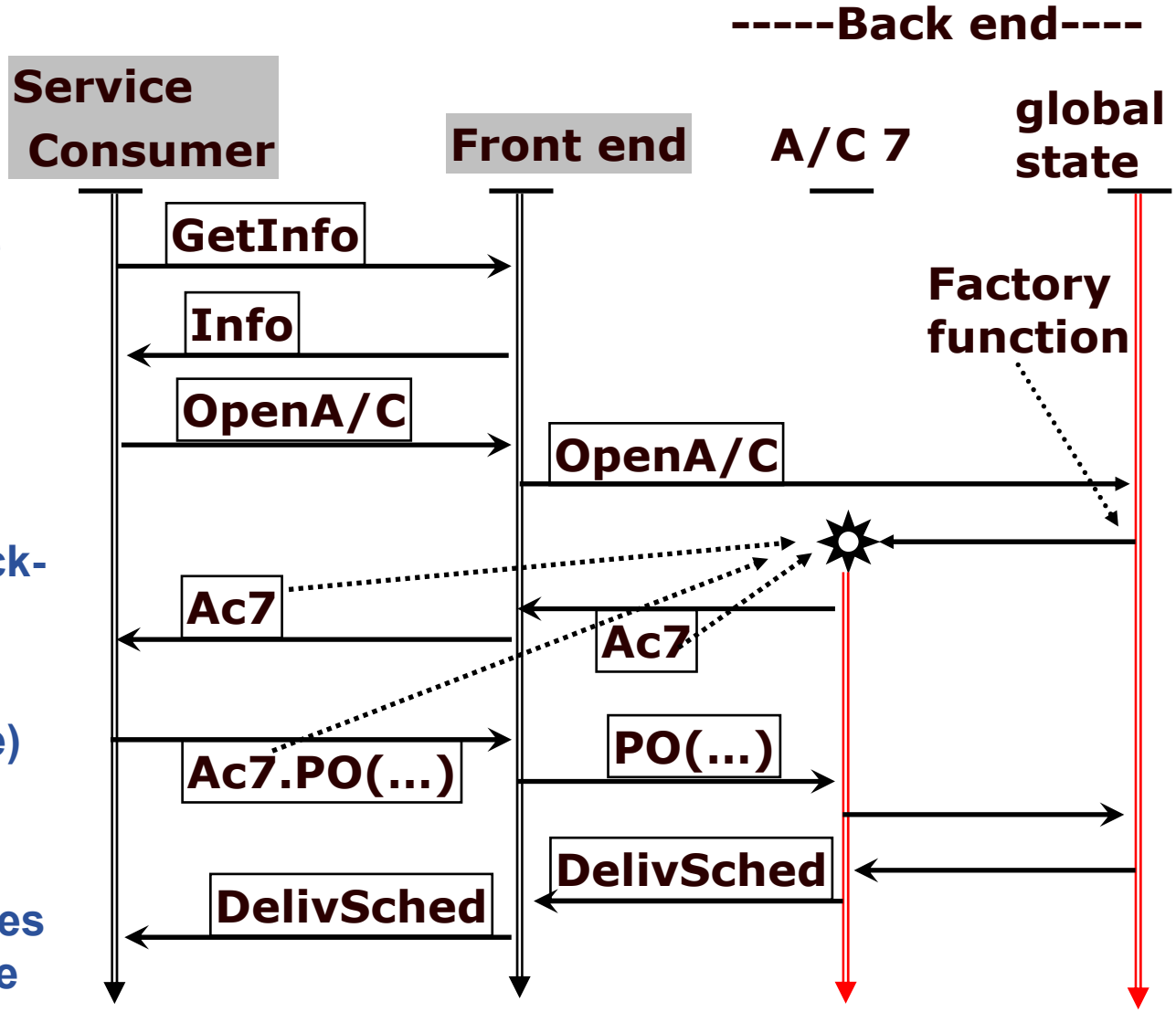
- **Goals**
  - To be gain an understand of
    the (proposed) Web Services Resource Framework
- **Outline**
  - General
  - Resource Properties Document
  - Lifetime
  - Notification and WSRF

Context

- Basic Standards
  - XML
  - Schema
  - SOAP
  - WSDL
- Provide the ground framework
- Supplementary Standards
  - Build on basic standards to meet particular requirements, e.g.
  - WS-Notification
  - WS-Security
  - WS-RF
  - WS-TransactionFramework
  - ….
- WSRF is important
  - Particularly for grids
  - Addresses Fundamental architectural issue in
    - doing O-O-like approach on web services

Stateful Resources

- Web service itself
  - (Front end)
  - is stateless
- Freely have multiple instances that come and go –
  - Scaleability
  - Reliability
- Maintains state in a back-end
- That is somewhat O-O
  - a resource (instance)
  - corresponds to an object (instance)
- Service request identifies the specific resource



Taxonomy of States and Services

- **Stateless** – implements message exchanges for which there is no access or use of information not contained in the input message. E.g. document compression / de-compression
- **Out-of-band persistent** state – response is affected by information that changes by some no-WS means. E.g weather forecast service
- **Transient** State (**conversational**) – to co-ordinate a collection of related message exchanges E.g : shopping-basket;
  - Booking holiday - book hotel, flights and car-hire via different services with two-phase comit – confirm a reservation when all are held.
  - Proppsed standards for this – WS-TransactionFramework
- **Persistent** state (**stateful resource**) – one message exchange produces a long-lived change in state which affects other message exchanges

  if shopping basket were carried forward from session, this would be persistent state
- **Combination** – Booking holiday is conversational involving several persistent state services
- **WSRF** is for Persistent State, not Conversational

WSRF Architecture

- **A stateful (WS) resource**
  - Is a repository for persistent state
    - Like an object in an object-oriented architecture
  - Has state that Comprises a set of state data
    - Each item of state data is a resource property
    - A resource property is expressible as an XML document,
    - which can in principle be retrieved and updated
  - E.g Account has properties
    - Balance owed
    - Credit limit
    - Outstanding deliveries - complex
    - Latest statement - complex
  - …

WSRF Architecture

- **A stateful (WS) resource**

  ….

  - Has a well-defined life-cycle – creation and destruction
    - Destruction can be explicit or scheduled
  - Can be known and acted upon by one or more Web Services
    - Has a globally unique identifier –
      - http://www.company/CreditCard#Ac7
      - Can be passed between services to identify the resource
  - Is associated with one or more web services, providing interface for manipulating it
  - **A WS-resource comprises: its service; the resource itself**

Not O-O

Differences from Object-Oriented Architecture

- **Explicit State**
  - State can be expressed as an XML document which in principle is retrievable and updatable (resource properties doc)
  - In O-O an object is defined by the operations on it and the affect of those operations on future operations
    - Could have an object state which is not representable as a document - An irrational number with operations: get/set n-th digit in decimal notation; set to known rational, e.g “pi”
  - The type of a WS-resource is
    - the type of its resource properties document
    - not the signatures of its operations

Not O-O

- **Scheduled Destruction**
  - Can say this resource may self-destruct after 3 days
  - In O-O destruction is by either
    - Explicit or implicit in some action
      - (but in web, service to do the action may be gone)
    - Garbage collection
      - (but un-realistic for web)

- **Looser Encapsulation**
  - In O-O an object has one interface
  - In WS a resource can be operated on by several services
    - With several different interfaces –
    - the type of the resource is that of its properties doc, not the signatures of its operation

Not O-O

- **Out-of-Band operations**
  - In O-O everything happens within one coherent framework –

    Other than for the initial object
    - Every object is created by the single initial object or an already created object
    - Every change to an object's state is a result of an operation performed on it by some other object deriving from the intital object
  - In WS
    - Creation and Modification
    - Can happen by non-WS mechanism
      - Human intervention
    - Services seem to spontaneously appear and disappear

- **Fault tolerant**
  - In WS partial (permanent) failure is expected and accommodated
  - Partial failure is a permanent condition

"Implied Resource" Pattern

The Implied-Resource Pattern

- **Implied Resource –**

  An association between message exchange and the particular resource – an implied input to the operation

  - Implied – the resource identifier is NOT an explicit parameter in the request
  - Implicit association is either
    - Static – association is made when the web service is deployed – 1:1
    - dynamic – association at time of message exchange – which can be as a property of the address. Could be as a header.
  - But will use explicit parameter in tutorial – not yet any WSRF tool support so otherwise would need to explictly gernerate SOAP messages
- **Pattern**
  - A set of usage conventions on existing technologies

Component Standards

- **WSRF builds on**
  - WS-Addressing – W3C submission Aug 2004
  - WS-Notification
- **WSRF comprises standards**
  - **WS-ResourceLifetime** 1.2 – working draft, June 2004
  - **WS-ResourceProperties** 1.2 – working draft, June 2004
  - WS-RenewableReferences – who knows?
  - WS-ServiceGroup 1.2 – working draft, June 2004
  - WS-BaseFaults 1.1 – initial draft, March 2004
- **WSRF supports –**
  - **WS-Notification**
    - WS-BaseNotification 1.0 – OASIS initial draft 1.0 May 2004
    - WS-BrockeredNotification 1.0 – OASIS initial draft 1.0 May 2004
    - WS-Topics 1.2 – OASIS working draft July 2004

Running Example

Seat Booking System for a Specific Event

- **Resource – Event6**
  - Properties –
    - Places – number of seats in total
    - Held – number of seats with provisional bookings
    - Booked – number of seats with confirmed bookings
  - Service
    - End-point – www.events.org/E6
    - Operations

      Get – returns the resource properties

      Reserve – creates a reservation resource

      ….

Running Example

- **Resource – A Booking**
  - Properties –
    - RNo – reservation number Identifier, not an actual property
    - Firm - number of seats with confirmed booking
    - Hold – number of seats with provisional booking
    - Seats – list of seat numbers allocated
    - Arrival – expected time of arrival
  - Service
    - End-point - www.events.org/E6res
    - Operations

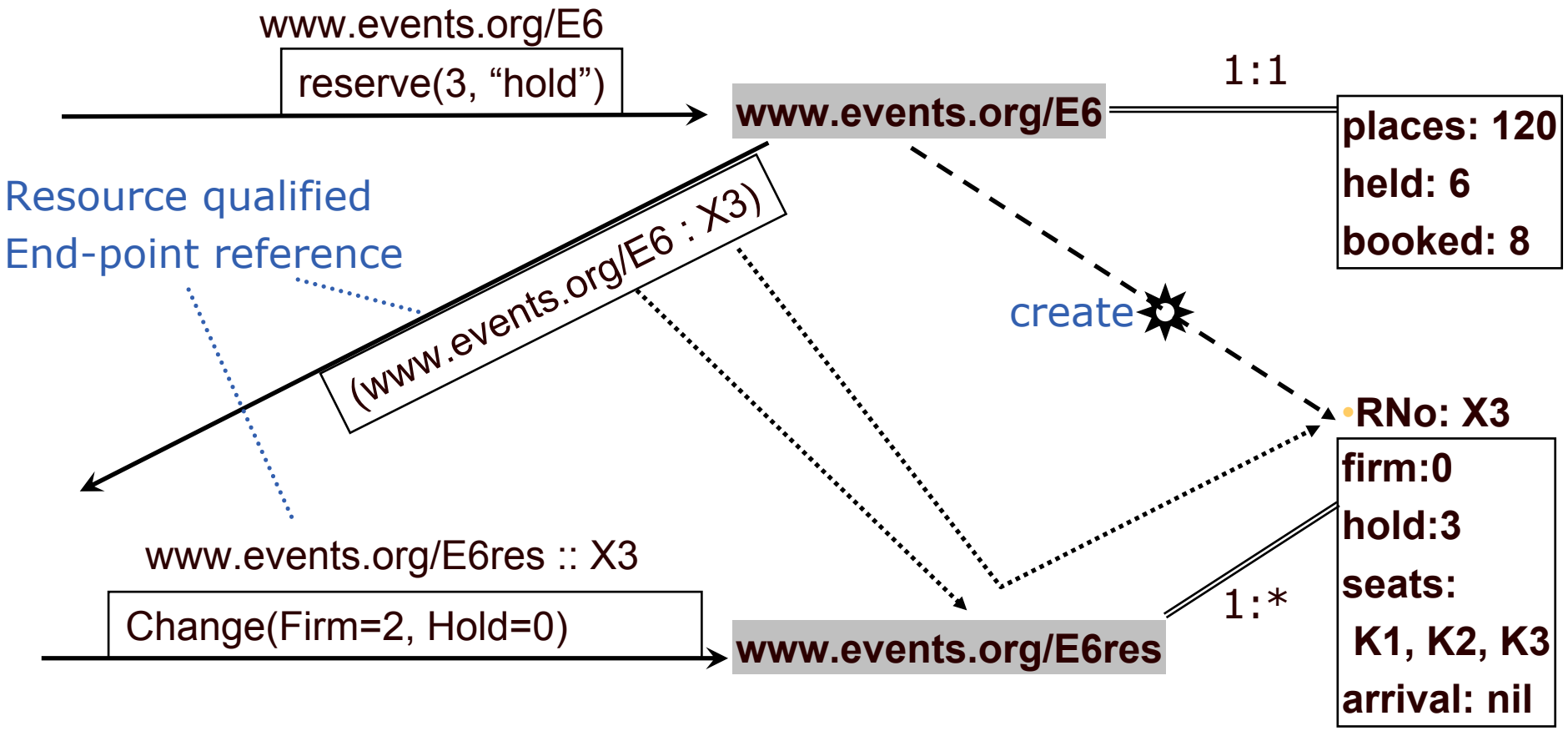
      Get – retrieve the properties

      SetArrival – change/set the Arrival property

      Change – reset reservation properties (firm=n; hold=m)

      ….

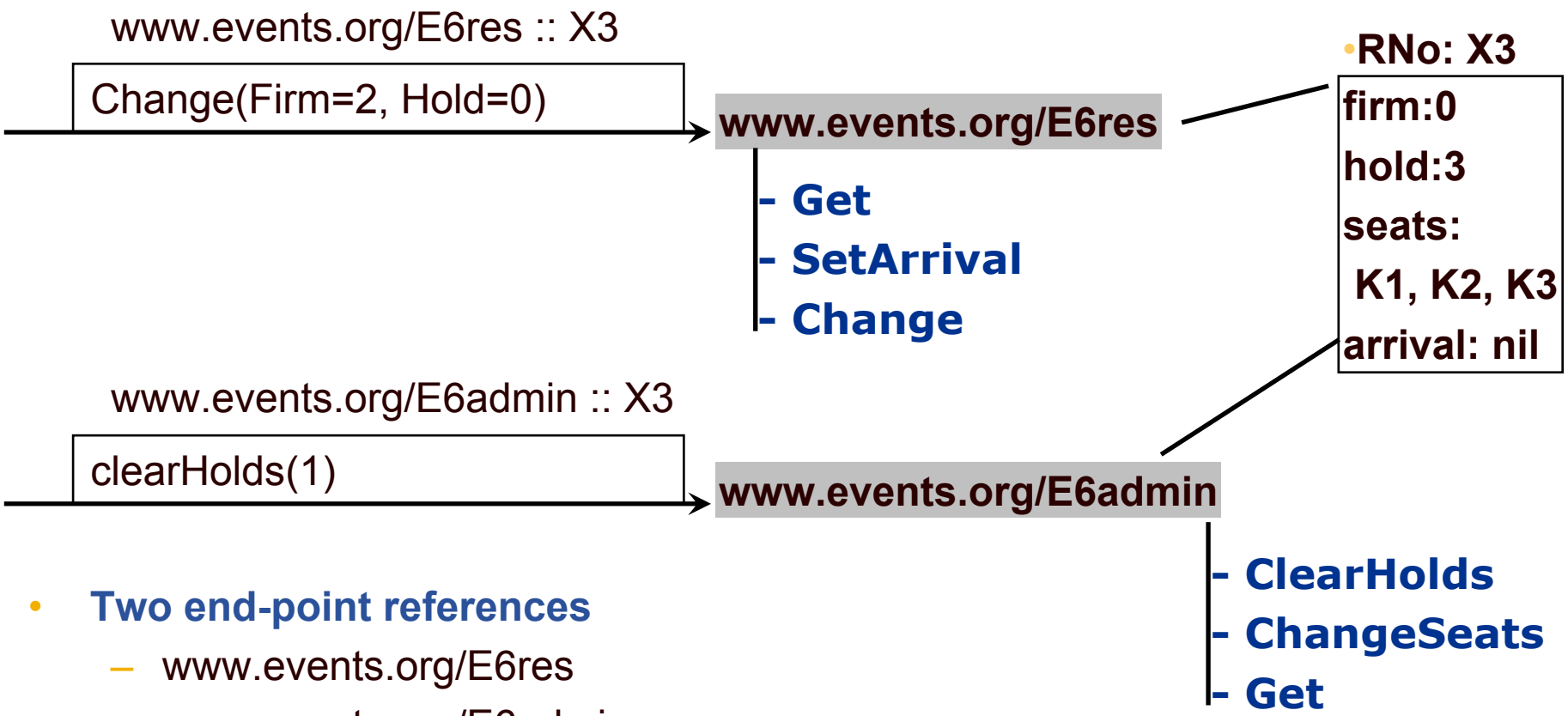
Referencing



Resource-qualified endpoint reference – the service address and specific resource identifier – part of WS_Adressing Standard

If service has only one resource instance (1:1) don’t need to include resource identifier in address

Multiply-serviced Resource



- Two end-point references
  - www.events.org/E6res
  - www.events.org/E6admin
- Each providing a different interface to the same set of resource instances

Resource Identifiers

- **A resource identifier is**
  - Managed by the associated service(s)
  - Unique within each associated service
    - Service-name :: resource-id is universally unique
    - Service-name :: resource-id can be passed around and is guaranteed to identify that resource whoever uses it
  - Opaque
    - No-one other than an associated service should attempt to interpret it or de-compose it – no semantics outside an associated service
    - Can't even compare two to see if they identify the same resource
- **Identity of a resource is some non-opaque identifier**
  - E.g: a person's e-mail address; an ISBN
  - Allows resources in independent services to be X-referred
  - Not covered by the WSRF standards
  - Would be a property of the resource
  - Would be namespace scoped
    - ISBN:123-64-27694 ISBN = www.ISBNs.org

OUTLINE

- **Goals**
  - To be gain an understand of
    the (proposed) Web Services Resource Framework
- **Outline**
  - General
  - Resource Properties Document
  - Lifetime
  - Notification and WSRF

Resource Properties

- **A resource has a resource properties document**
  - gives values for those aspects of the resource's state which
  - Can be retreived and possibly modified by service consumer
  - Through a Web Services interface
- **That document has a type**
  - fixed for all instances of the resource type
  - defined by a Schema
  - each **resource property** is a global element within that schema
  - The properties document as a whole is a global element
    - with all the resource properties as children (not attributes)
    - identified using "ref"
    - using a sequence or all constructor (not choice)
    - The actual order is immaterial

A Resource Properties Schema

```
<xsd:schema targetNamespace=www.events.org/E6res/ResProps
      xmlns:tns =www.events.org/E6res/ResProps
      xmlns=http://www.w3.org/2001/XMLSchema … >
  <element name = “firm” type=“xsi:integer”/>
  <element name = “hold” type=“xsi:integer”/>
  <element name = “seat” type=“xsi:integer” />
  <element name = “arrival” type=“xsi:time” />
  <element name=“resProps”>
    <complexType>   <sequence>
       <element ref=“tns:firm”>    <element ref=“tns:hold”>
       <element ref=“tns:arrival”>
       <element ref=“tns:seat” minocurs=“1” maxoccurs=“10”> </></></></>
```

```
<resProps>   <firm>0</>  <hold>3</>   <arrival>17:30:00</>
                  <seat>K1<> <seat>K2</> <seat>K3</> </>
```

- **Could have put the seat elements within a seats grouping –**
- **generally better?? (avoids achange notification issue)**

WSDL for a resourced Service

- **The interface to the resource is a WSDL portType**
  - The WSDL definition has an attribute for portType which identifies the resource properties schema

    <wsdl:portType name="eventsRervationPortT"

    wsrp:ResourceProperties="tns:ResevationPropertiesT">

    <operation ..> …. </>
  - The fact that the PortType has a properties type document is what says it is a resourced port –
    - Must obey the WSRF standards -
      - Particular operations for resource access/manipulation

Get and Set Operations

- **GetResourceProperty**
  - To retrieve value of a single resource property
  - Mandatory if there is ResourceProperties attribute for the PortType – the port support a WS-Resource
- **Optional -**
- **GetMultipleResourceProperties**
  - To retrieve values of several properties – important for granuality considerations
- **SetResourceProperties**
  - Provide a set of changes, e.g existing properties document

    <resProps> <firm>0</> <hold>3</> <arrival>17:30:00</>
    <seat>K1<> <seat>K2</> <seat>K3</> </>
  - **Insert** – e.g. add a new seat element
  - **Update** – e.g. remove all seat elements and put in a new set

    e.g. remove firm element and put in a new one
  - **Delete** – e.g remove all seat elements

GetResourceProperty

```
<env:envelope ….namespace definitions ..>
<env:Header>     <wsa:Action> http://...wsrf.../GetResourceProperty </>
                 <wsa:To env:mustUnderstand="1"> www.events.org/E6res.</>
                 <m:ResId>X7</>  </>
<env:Body>       <wsrp:GetResourceProperty>tns:seat </> </></>
```

```
<env:envelope ….namespace definitions ..>
<env:Header>     <wsa:Action> http://...wsrf.../ GetResourcePropertyResponse </>
                 <wsa:To env:mustUnderstand="1"> www. … requestor … </>
                 <m:ResId>X7</>  </>
<env:Body>       <wsrpl:GetResourcePropertyResponse>
                       <seat>K1</> <seat>K2</> <seat>K3</>  </></>
```

- **Returns all elements with the specified element name**
- **Faults –**
  - ResourceUnknownFault – X7 does not exist
  - InvalidResourcePropertyQName – tns:seat is not a property
  - … (allowed read access is an authorisation issue, not a WSRF issue ???)

Get Multiple Resource Properties

Enabling Grids for E-sciencE

```
<env:envelope ….namespace definitions ..>
<env:Header>     <wsa:Action> http://...wsrf.../GetMultipleResourceProperties </>
                 <wsa:To env:mustUnderstand="1"> www.events.org/E6res.</>
                 <m:ResId>X7</>  </>
<env:Body>       <wsrp:GetMultipleResourceProperties>
                        <wsrp:ResourceProperty>tns:firm </>
                        <wsrp:ResourceProperty>tns:seat </></></>
```

```
<env:envelope ….namespace definitions ..>
<env:Header>     <wsa:Action> http://...wsrf.../ GetResourcePropertyResponse </>
                 <wsa:To env:mustUnderstand="1"> www. … requestor … </>
                 <m:ResId>X7</>  </>
<env:Body>       <wsrpl:GetResourcePropertyResponse>
                        <firm>3</> <seat>K1</> <seat>K2</> <seat>K3</>  </></>
```

- **Must specify at least one**
- **Order in response should follow order in request**
- **Same faults as for single one**

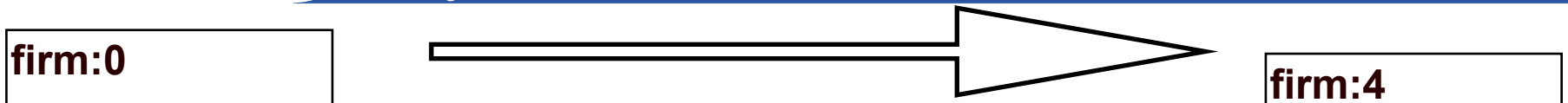
SetResourceProperties

```
<env:envelope ….namespace definitions ..>
<env:Header>      <wsa:Action> http://...wsrf.../SetResourceProperties </>
                  <wsa:To env:mustUnderstand="1"> www.events.org/E6res.</>
                  <m:ResId>X7</>  </>
<env:Body>
   <wsrp:SetResourceProperties>
        <wsrp:Update><tns:hold>0</>  <tns:firm>4</></>
        <wsrp:Delete resourceProperty="tns:arrival">
        <wsrp:Insert> ><tns:seat>J9</> </></></></>
```

```
<env:envelope> ……..<env:Body>   <wsrpl:GetResourcePropertyResponse></></></></>
```

- **A number of SetRequestComponents, each – insert, update, delete**
- **Must be done in given order – could have several for same element name**
- **If failure on one,**
  - must not do any subsequent ones
  - Final result may reflect the partial processing
  - Final result may be the original

SetResourceProperties

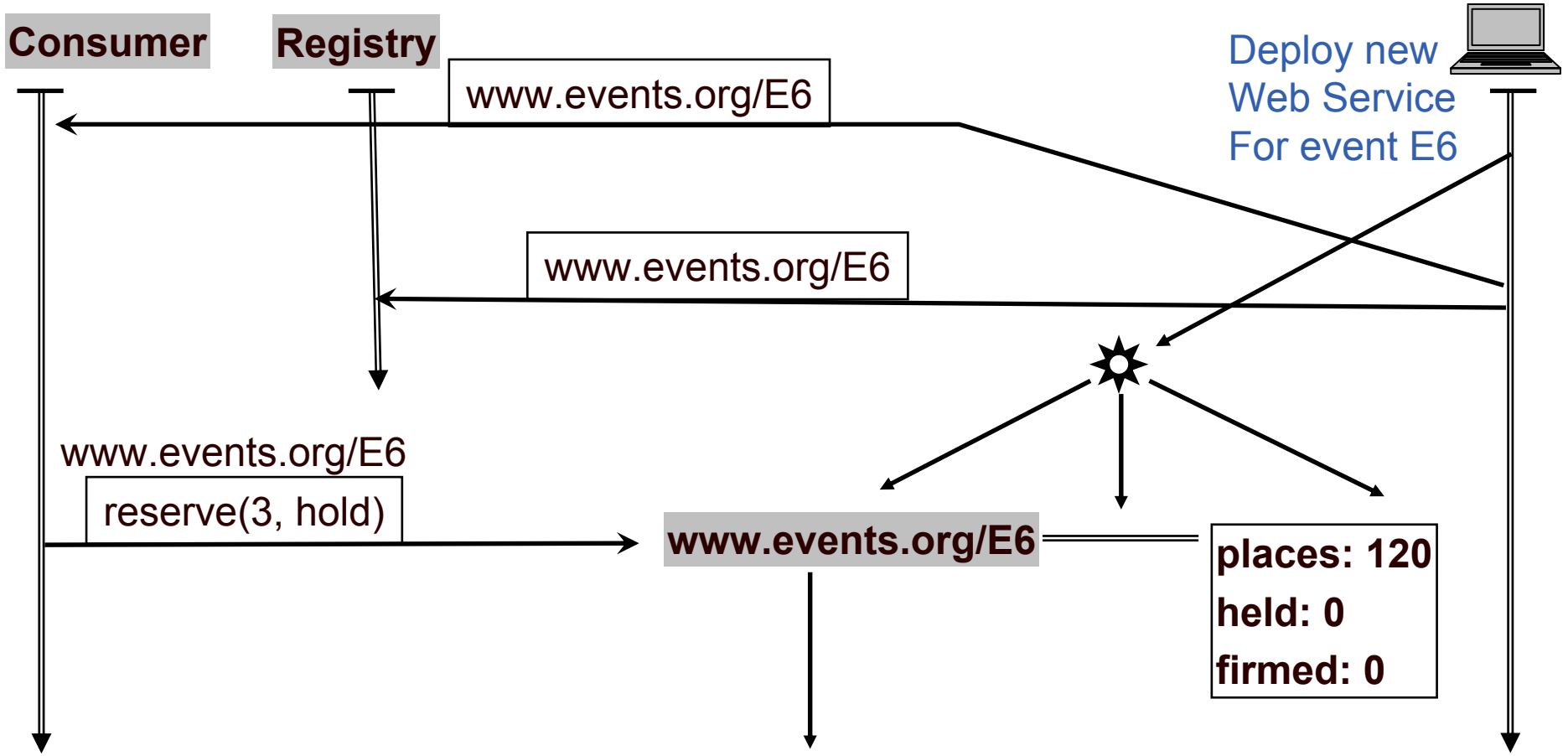


- **Faults**
  - ResourceUnknownFault
  - InvalidResourcePropertiesRequestContent
    - The result would be a properties document which is invalid, e.g. too any seats if maxoccurrs=3
  - UnableToModifyResourceProperty – a read-only resource
  - InvalidResourcePropertyQName
  - SetResourcePropertyRequestFailed – one or more components failed
  - … to be defined
  - Fault message must indicate whether effects of processing non-failed components were restored

OUTLINE

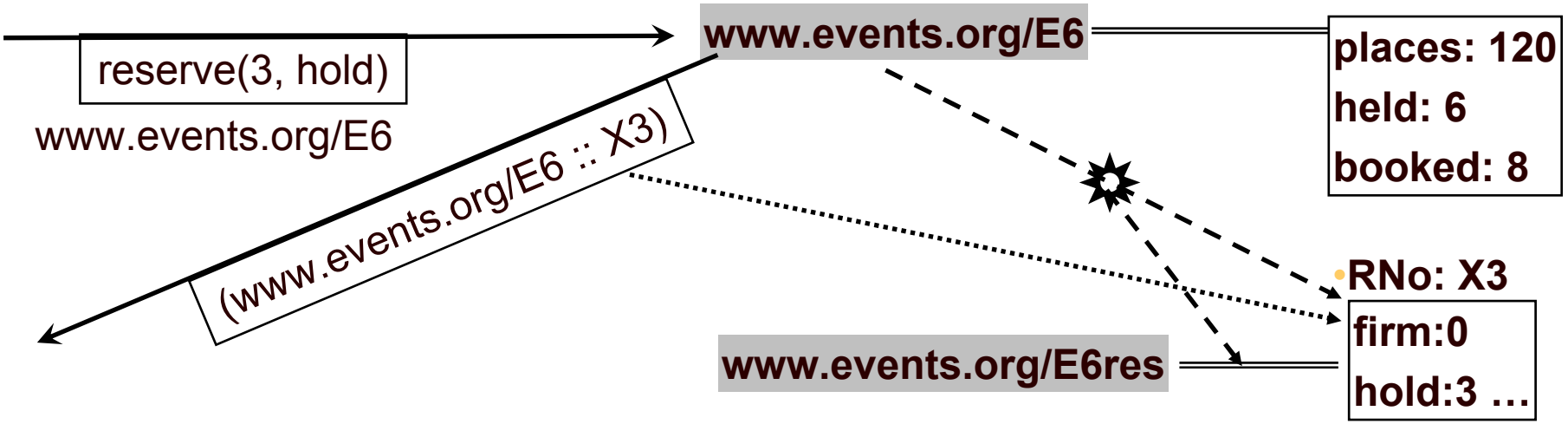
- **Goals**
  - To be gain an understand of
    the (proposed) Web Services Resource Framework
- **Outline**
  - General
  - Resource Properties Document
  - Lifetime
  - Notification and WSRF

Out-of-Band Creation



- **Create the resource, its service and their connection**
- **Inform potential users directly and/or via registry**

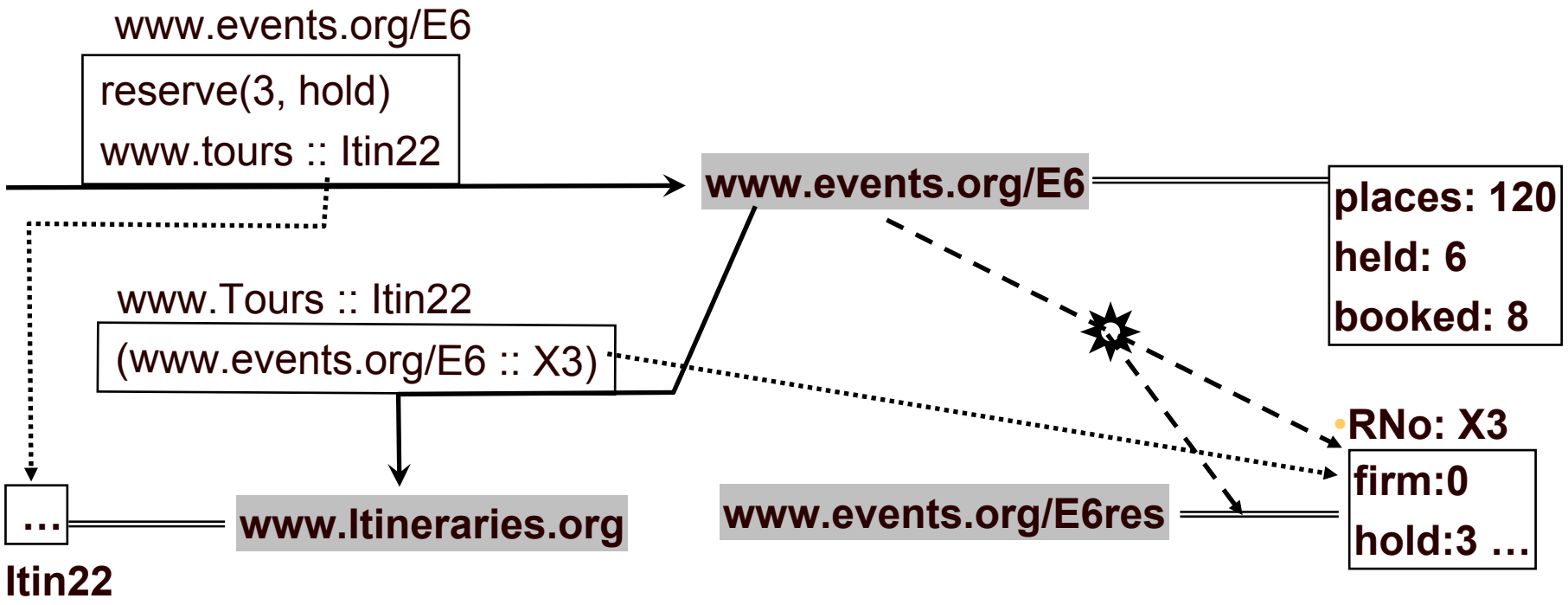
Resource Factory



Resource Factory Pattern

- **Pattern, not defined message formats**
- **A WS resource factory is a Web Service capable of bringing WS-resources into existence**
  - Create the resource
  - Assign its identity
  - Create the association between the new resource and its service
  - Provide the consumer with the reference to the resource

Resource Factory II



- **New Resource identifier may be passed back indirectly**
  - Via a registry
  - Here by adding to a specified Itinerary resource

Resource Destruction

- **Immediate Destruction**
  - Send a destroy message to the resource-qualified endpoint
  - Thereafter any attempt to access it must result in a *Unknown Resource* fault message – this is a synchronisation point – the reply to the destroy
  - The consumer could decide to destroy the reservation resource – cancelling the reservation

- **A resourced service should have a destruction policy which does not depend on action by the consumer service –**
  - Consumer may disappear at any time
  - Consumer may be impolite

- **Risk of having the physical resources never recovered, and performance consequences of large number of useless resource instance**

Scheduled Destruction

- **Scheduled Destruction**
  - Can establish a scheduled termination time for the resource
    - Possibly by negotiation at create time
    - A reservation resource has termination time at latest the event date
    - One with non-firm parts has termination time of 2 days from creation
  - Can request a modification in the termination time
    - Extend the provisional booking for another 2 days
    - If termination time is in the past this may be interpreted as an immediate asynchronous destroy
  - Termination time may change non-monotonically –
    - New termination time may be earlier or later than the old one
- **A resourced service should have a destruction policy which does not require action by the consumer service –**
  - Consumer may disappear at any time
  - Consumer may be impolite

Immediate Destruction Exchange

Enabling Grids for E-sciencE

```
<env:envelope ….namespace definitions ..>
<env:Header>     <wsa:Action> http://...wsrf.../Destroy </>
                 <wsa:To env:mustUnderstand="1"> www.events.org/E6res</>
                 <m:ResId>X7</>  </>
<env:Body>       <wsrl:Destroy/> </></>
```

```
<env:envelope ….namespace definitions ..>
<env:Header>     <wsa:Action> http://...wsrf.../DestroyResponse </>
                 <wsa:To env:mustUnderstand="1"> www. … requestor …</
                 <m:ResId>X7</>  </>
<env:Body>       <wsrl:DestroyResponse/> </></>
```

- **If there is a destroy capability, then this is it**
- **Possible faults are –**
  - ResourceUnknownFault –identifier didn't identify a known resource
  - ResourceNotDestroyedFault – for some other reason
  - … to be defined

Scheduled Destruction - Properties

If supporting scheduled then follow this standard

- **resource has a current time property**
  - a get-able, non-set-able, single occurrence, element
  - <xsd:element name="CurrentTime type="xsd:dateTime"/>
  - This is to help consumer determine clock difference – which is relevant to setting termination time
- **resource has a current termination time property**
  - a get-able, non-set-able, single occurrence, element
  - if no time zone, then UTC (universal time)
  - <xsd:element name="TerminationTime nillable="true" type="xsd:dateTime"/>
  - If value is nil termination time is indefinite
- **get-able / set-able –** can be accessed/modified by the GetResourceProperties / SetResourceProperties operations
- **Throughout -** Use xsd:dateTime (YYY-MM-DDThh:mm: ss) – clock of resource; if no time zone then UTC (universal time)

Set Termination Time

```
<env:envelope ….namespace definitions ..>
<env:Header>      <wsa:Action> http://...wsrf.../SetTerminationTime </>
                  <wsa:To env:mustUnderstand="1"> www.events.org/E6res.</>
                  <m:ResId>X7</>  </>
<env:Body>        <wsrl:SetTerminationTime>
                    <wsrl:RequestedTerminationTime>2005-04-31T12:00:00</>
</> </></>
```

- **Operation to set the termination time**
  - If value is nil, then no termination time; if this is allowed resource should support immediate Destroy
  - May be rejected for any reason (policy limits lifetime)

Set Termination Time Response


```
<env:envelope ….namespace definitions ..>
  <env:Header>  <wsa:Action> http://...wsrf.../SetTerminationTimeResponse </>
                <wsa:To env:mustUnderstand="1"> www…requestor…</>
                <m:ResId>X7</>  </>
 <env:Body>     <wsrl:SetTerminationTimeResponse>
                    <wsrl:NewTerminationTime>2005-04-31T12:30:00</>
                    <wsrl:CurrentTime>2005-04-15T12:30:00</></> </></>
```

- **Can't depend on it being destroyed at that time – arbitrary delay allowed**
- **Termination Time property must match NewTerminationTime**
- **May be in the future relative to the requested termination time**
- **Fault messages –**
  - ResourceUnknownFault – as always
  - UnableToSetTerminationTimeFault – for some reason
  - TerminationTimeChangeRejectedFault
    - Could reply with a hint as to acceptable new value for termination time

Notification

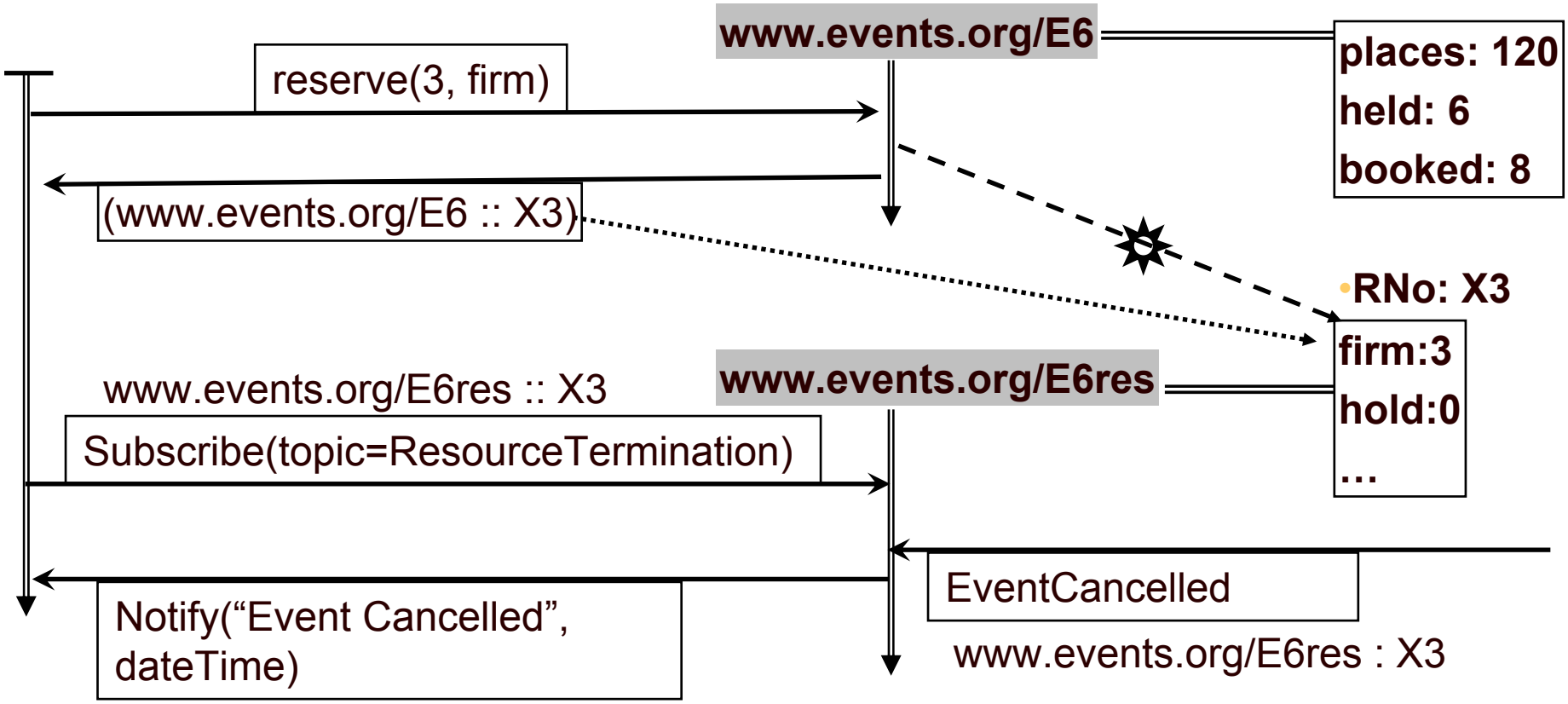
- **WS-Notificatoin is draft standards dealing with the**
  - The Notification-based Interaction pattern – Event Driven
- **Model - Subscribing to a Notification service on some topics**
  - E.g. My boss (**Subscriber**) informs a press-cutting service (**Publisher**) that it is to notify me (**Consumer**) of articles on WebServices (**Topic**) appearing in the popular press (**Producer**)
  - **Topic Space** - a forest of **topic Trees**

  - **Publisher** – distributes **notification** messages according to **subscriptions**
  - **Producer** – generates notification messages for **Consumers**
  - Can combine Producer and publisher - same service generates the event and sends it to the subscribers; otherwise **Publisher** is a **Broker**
  - Can separate them – producer generates the notification and sends it to a **broker** who distributes it according to subscriptions
  - **Subscriber** creates a subscription for a **consumer** in a Publisher
  - Consumer receives notification messages (may combine with subscriber)

Notification and WSRF

- **Relation to WSRF**
  - A subscription is a resource
  - A resourced service can be producer/publisher –
    - to notify consumers of changes in state of the resource
      - Value change
      - Destruction

Destruction Notification Pattern



- **WS-Notification standard deals with this**
- **Subscribe to the resource**
- **Resource notifies subscriber**

Destruction Notification

- If Resource chooses to support the pattern of notifying interested parties when it is destroyed
- And to use the WS-Notification standard,
- Then must follow this standard
- The TopicSpace = “ResourceLifetime”
- The Topic name=“ResourceTermination”
- The notificaiton message must include the following element

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

Value Change Notification

- **Can similarly subscribe to being notified of value changes for the resource.**
- **If the resource supports the property value-change notification pattern, and it uses WS-Notification then it must follow these standards**
  - Subscription can be to a sub-set of the the resource properties
    - E.g. wanting notification of changes in seat numbers
  - The notification message must contain an element of the form

```
<wsrp:ResourcePropertyValueChangeNotification>
        <wsrp:OldValue> <seat>K1</> <seat>K2</> <seat>K3</></>
        <wsrp:NewValue> <seat>K1</> <seat>K2</> <seat>J4</></></></>
```

- **One such notification for every value change**
- **OldValue – if nil, there was no value; if absent the old value was not recorded**
- **NewValue – can be nil**
- **!!!Standard does not actually allo multiple components!!!**

The End

THE END