



Enabling Grids for E-science

PPARC Summer School, May 2005

Web Services Resource Framework– WSRF

Richard Hopkins

National e-Science Centre, Edinburgh

www.eu-egee.org



- **Goals**
 - To be gain an understanding of the (proposed) Web Services Resource Framework

- **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 = WSRF
 - WS-TransactionFramework
 -
- **WSRF is important**
 - Addresses Fundamental architectural issue in
 - doing O-O-like approach on web services
 - Particularly relevant to grids

Web service itself

(Front end)

is stateless

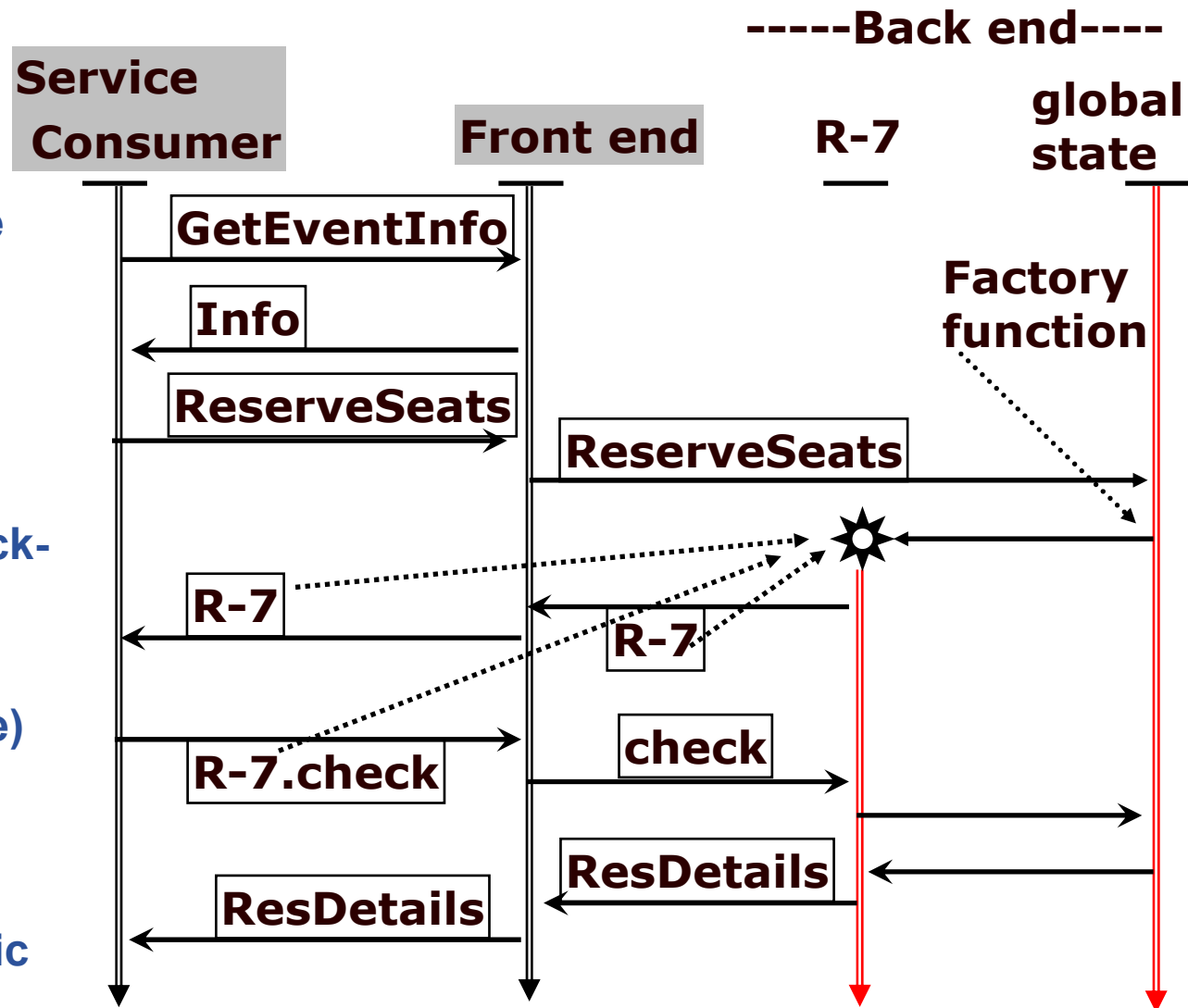
Freely have multiple instances that come and go –
Scalability
Reliability

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

Example – reserve seats for a specific event



- **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.
 - Proposed 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

- A stateful (WS) resource is a repository for persistent state
 - Has state that Comprises a set of state data 
 - Has a well-defined life-cycle – creation and destruction
 - Two kinds of resources (not formally distinguished)
 - Reservation – dynamically created by Web Service interactions
 - Event – creation is “out of band” – a separate mechanism
 - Is associated with one or more web services, providing interface for manipulating it
 - A WS-resource comprises: its service; the resource itself
 - Has a globally unique identifier –
 - www.events.org/E6#R-7
 - Can be passed between services to identify the resource
 - Might offer notification of state changes
 - (Is similar to an object in O-O architecture)

Each item of state data is a resource property

- which can in principle be retrieved and updated
- E.g Reservation has properties
 - Number of seats on hold
 - Number of seats confirmed
 - Seat numbers allocated
 - Outstanding Payment due
- E.g. Event has properties
 - Number of seats - in total
 - on hold
 - firmly booked
- **This is typically an abstraction of the actual state, the “exposed state”**
 - Might not be the entire state
 - Might not correspond to any physically stored data

E.g – a Queue of “jobs”

- **Full actual state –**
 - a linked list of job entries, each giving
 - Identifier, executable, resource requirements



- **Not appropriate to expose all of this**
 - Might be a very large amount of data
 - Users do not have a “need to know” it all
- **Exposed state might be**
 - Details of first job
 - Statistics on the complete job queue
- **Not the entire state**
- **Some properties not actually stored**
- **Therefore, some operations change the actual state but not the exposed state**

```

ResourceProperties
FirstJob:
  Name: J23;
  Exec: ...;
  CPUload:10MFLOPS;
  ...
Stats
  JobCount: 100
  CPUload: 265 MFLOP;
  ...
    
```


- The set of resource properties can be represented as a XML document –
- the resource properties document
 - Gives values for those aspects of the resource's state which
 - can be retrieved 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

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


```
<wsdl:portType name="eventsReservationPortT"
                wsrp:ResourceProperties="tns:ReservationPropertiesT">
  <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

- **Mandatory - GetResourceProperty**
 - To retrieve value of a single resource property – specify property name
 - Mandatory if there is ResourceProperties attribute for the PortType – the port supports a WS-Resource
- **Optional - GetMultipleResourceProperties**
 - To retrieve values of several properties – specify property names
 - important for granularity considerations

- **Optional - SetResourceProperties**

- Provide a set of changes
- e.g existing properties document

<resProps>

<firm> 0 </>

<hold> 3 </>

<owes> 169.36 </>


<seat> K1 </>

<seat>K2 </>

<seat>K3 </> </>

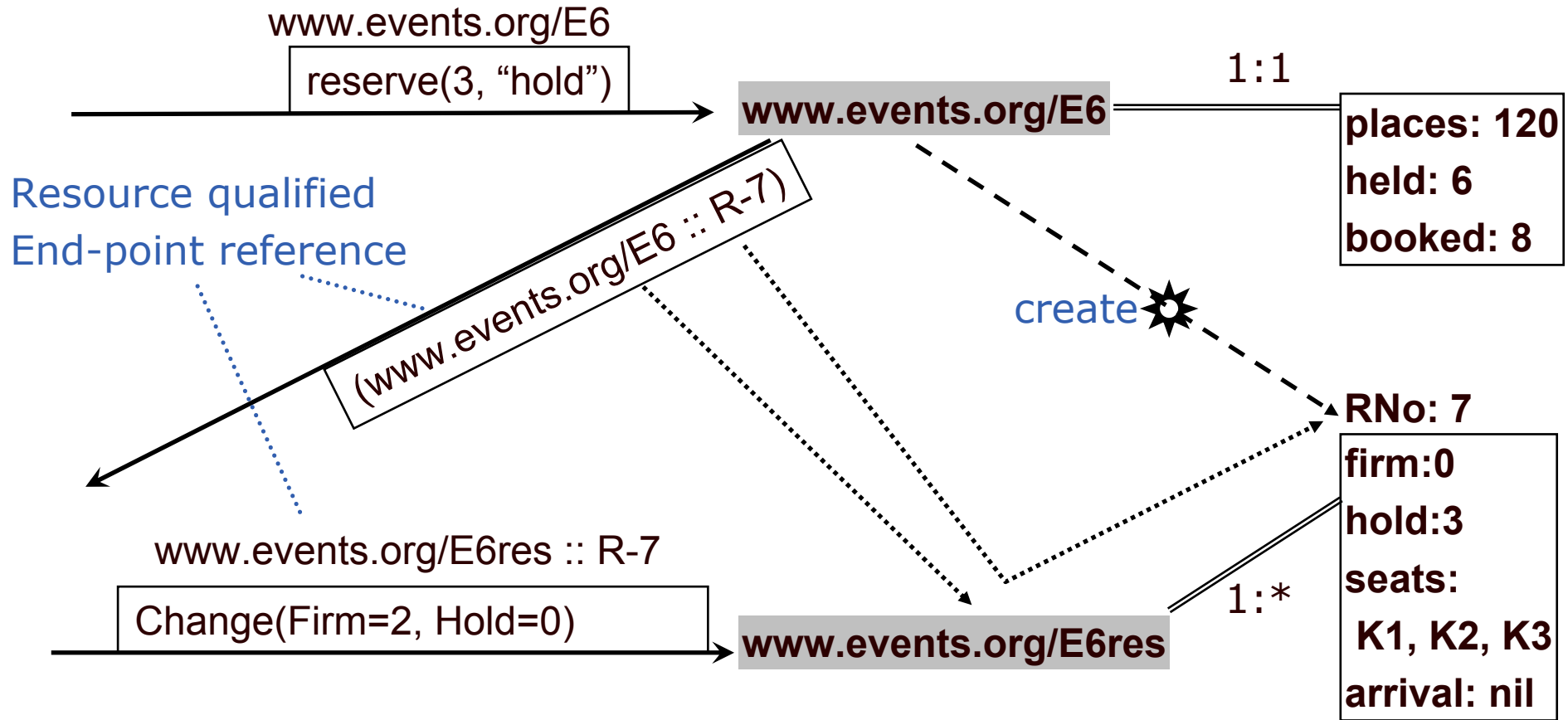
- **Insert** – e.g. add a new **seat** element
 - **Update** – e.g. remove all **seat** elements and put in a new set
e.g. remove **firm** element and put in a new one
 - **Delete** – e.g remove all **seat** elements
- **Might be partially applied**

A stateful (WS) resource

- Has state that Comprises a set of state data
- Has a well-defined life-cycle – creation and destruction
- Is associated with one or more web services
- **Has a globally unique identifier** 
- Might offer notification of state changes
- (Is similar to an object in O-O architecture)

The resource identifier gives an association between message exchange and the particular resource

- It is an implied input to the operation
 - 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.
- **Implied-Resource Pattern**
 - A set of usage conventions on existing technologies
 - Not necessarily a mandatory standard
- **In dynamic case - uses resource-qualified references**




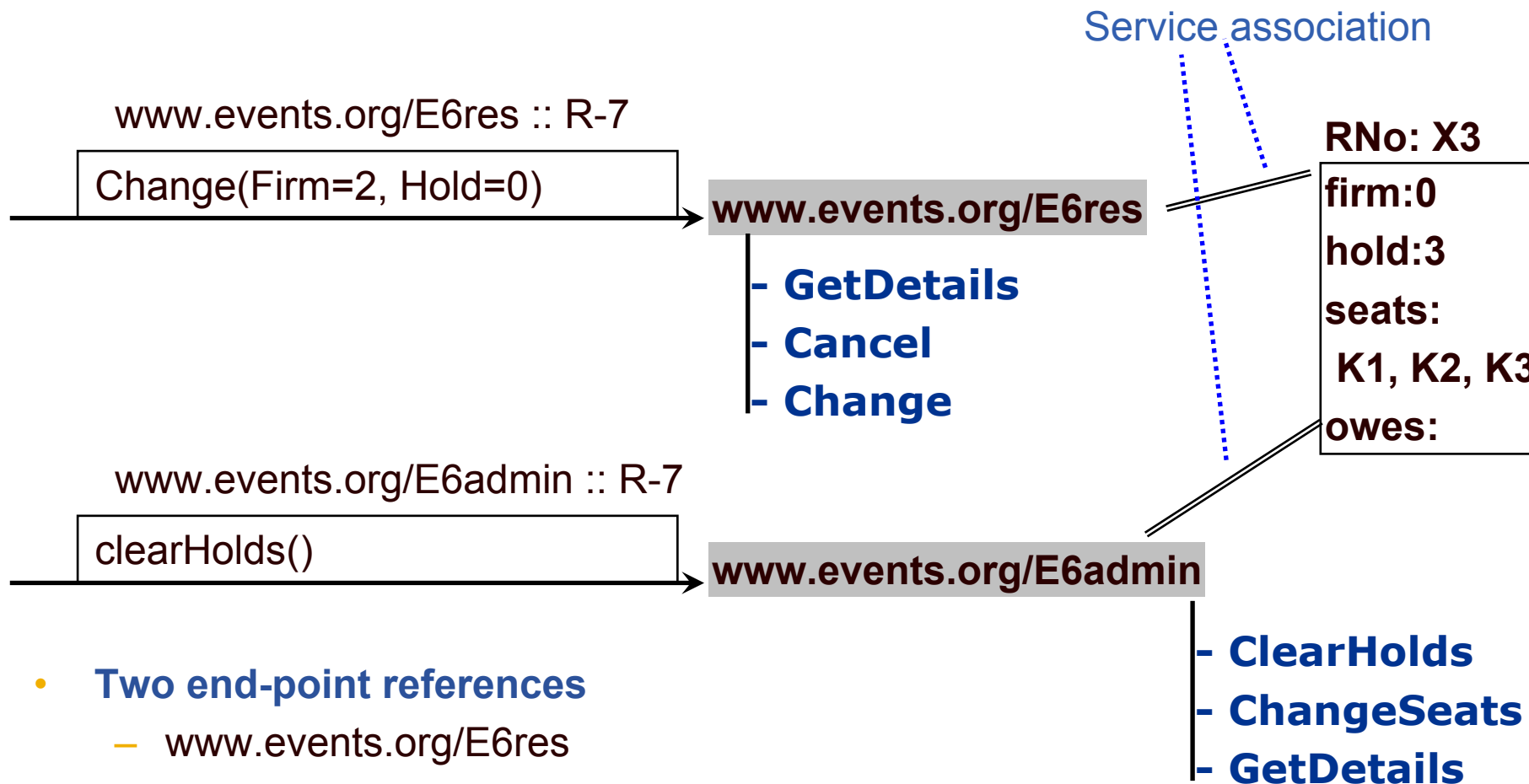
Resource-qualified endpoint reference – the service address and specific resource identifier – part of WS_Adressing Standard

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

- **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

A stateful (WS) resource

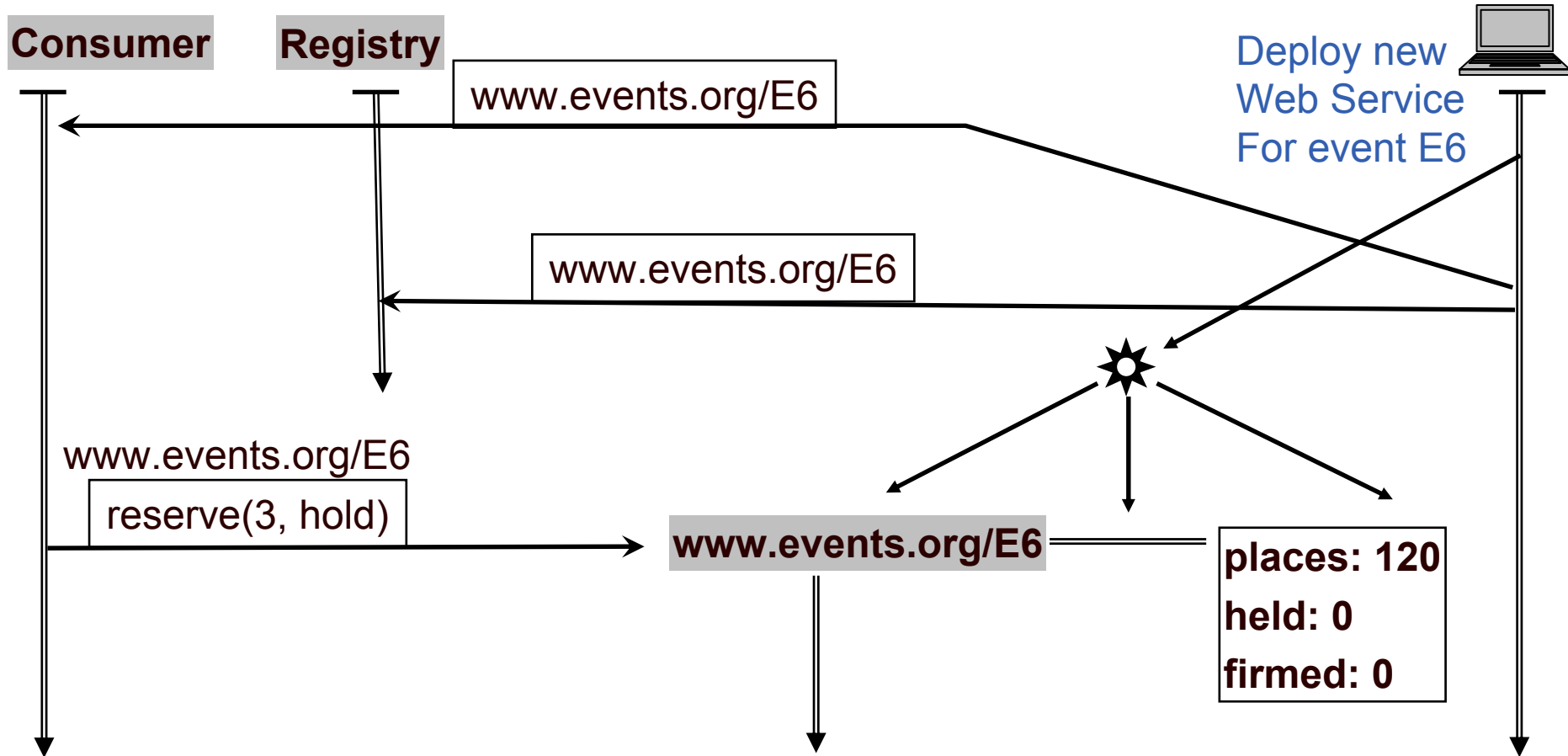
- Has state that Comprises a set of state data
- Has a well-defined life-cycle – creation and destruction
- **Is associated with one or more web services** 
- Has a globally unique identifier
- Might offer notification of state changes
- (Is similar to an object in O-O architecture)



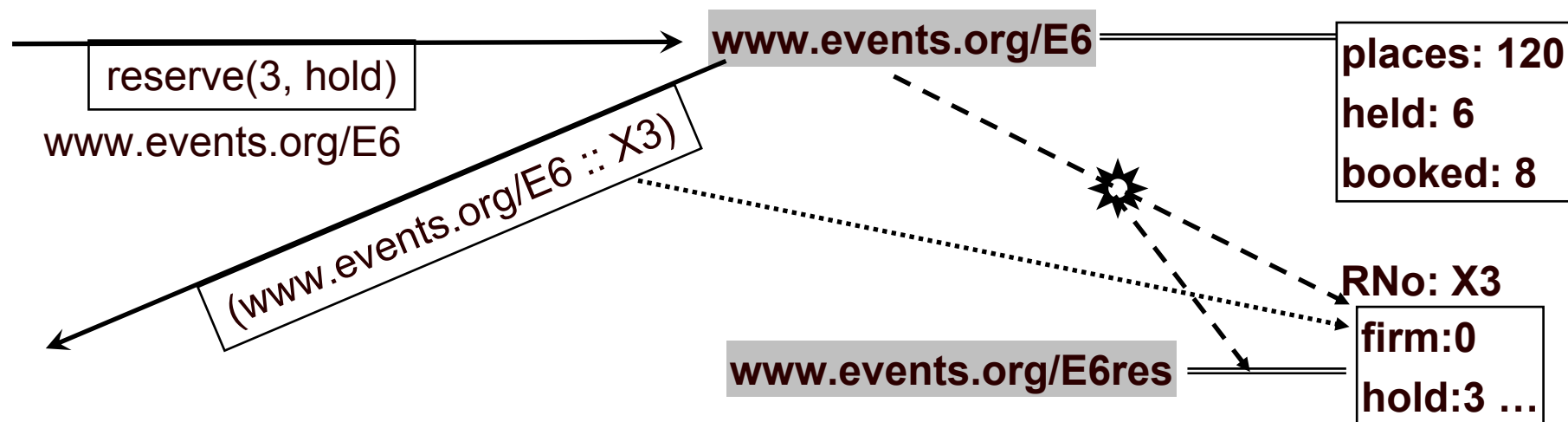
- **Two end-point references**
 - `www.events.org/E6res`
 - `www.events.org/E6admin`
- **Each providing a different interface to the same set of resource instances**

A stateful (WS) resource

- Has state that Comprises a set of state data
- **Has a well-defined life-cycle** 
 - **Creation**
 - **Out-of-band**
 - **Resource Factory**
 - **Destruction**
- Is associated with one or more web services
- Has a globally unique identifier
- Might offer notification of state changes
- (Is similar to an object in O-O architecture)



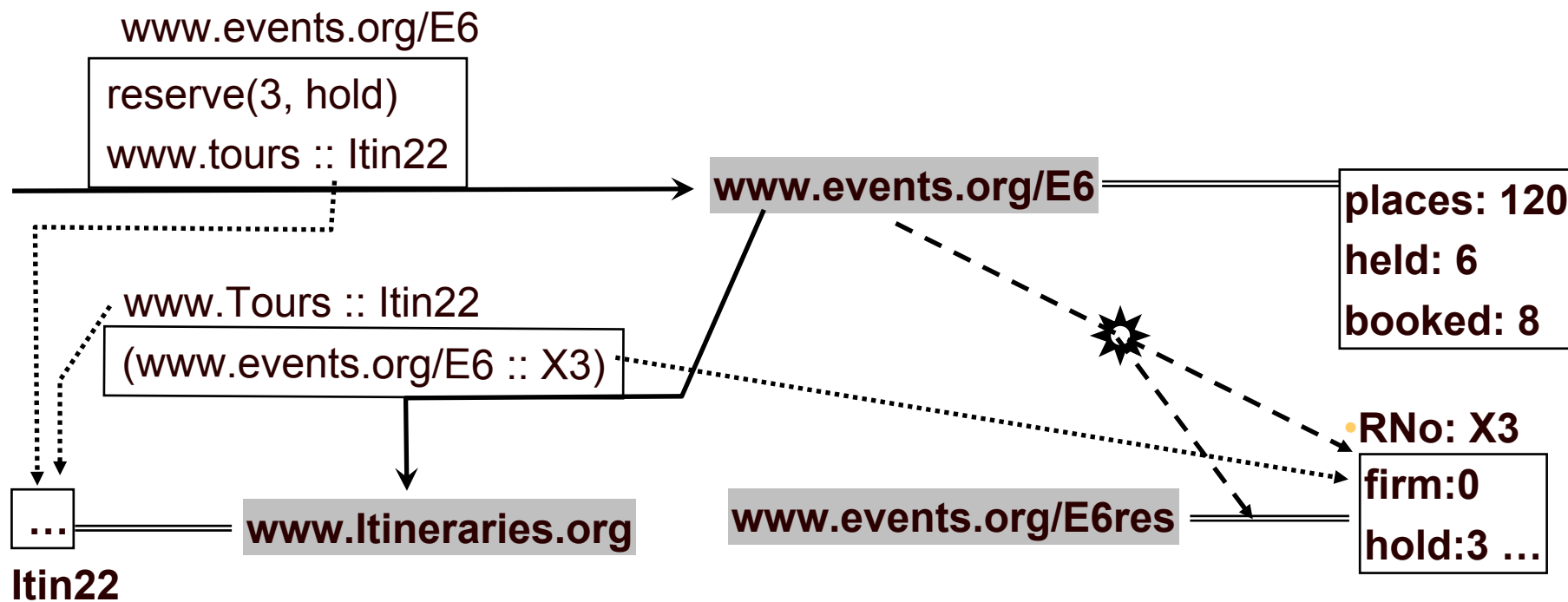
- Create the resource, its service and their connection
- Inform potential users directly and/or via registry
- User can access the spontaneously created resource



Resource Factory Pattern

- **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
- **Pattern - no defined operations nor message formats**

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



- **Immediate Destruction**
 - Send a destroy message to the resource-qualified endpoint
 - E.g. The consumer service could decide to destroy the reservation resource – cancelling the reservation
- **Specific standards for destroy operation**
 - Request / response / faults

- **A resourced service should have a destruction policy which does not depend on action by the consumer service –**
 - Consumer may be impolite or disappear at any time
 - Giving Risk of the physical resources never being recovered,
 - With performance consequences of large number of useless resource instance
- **Therefore Scheduled Destruction**
 - E.g.
 - 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 establish a scheduled termination time for the resource
 - Possibly by negotiation at create time
 - Can request a modification in the termination time
 - E.g. 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

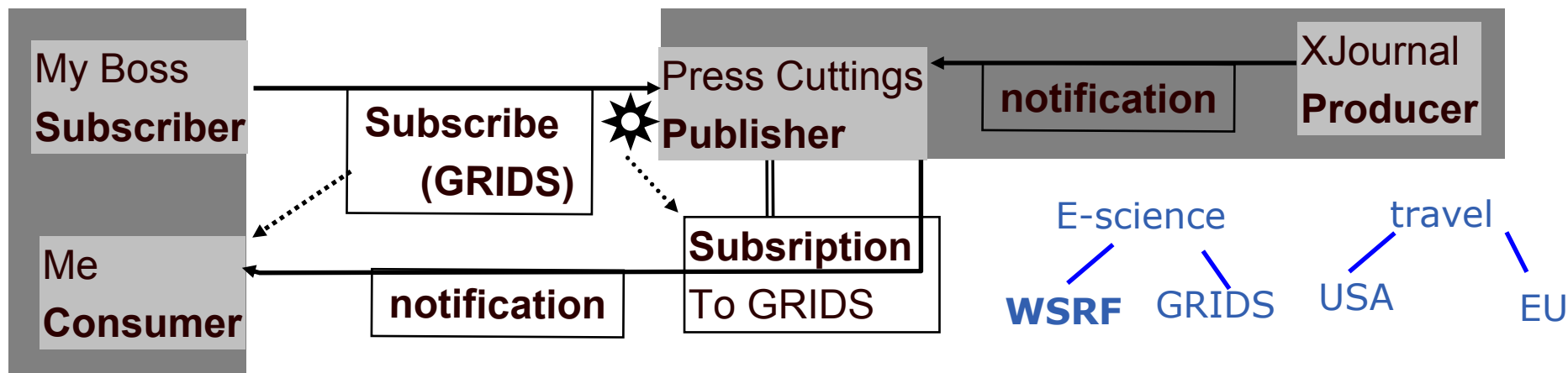
If supporting scheduled destruction, then follow this standard

- resource has a current time property
 - This is to help consumer determine clock difference – which is relevant to setting termination time
- resource has a current termination time property
- Set termination time operation – request and response

A stateful (WS) resource

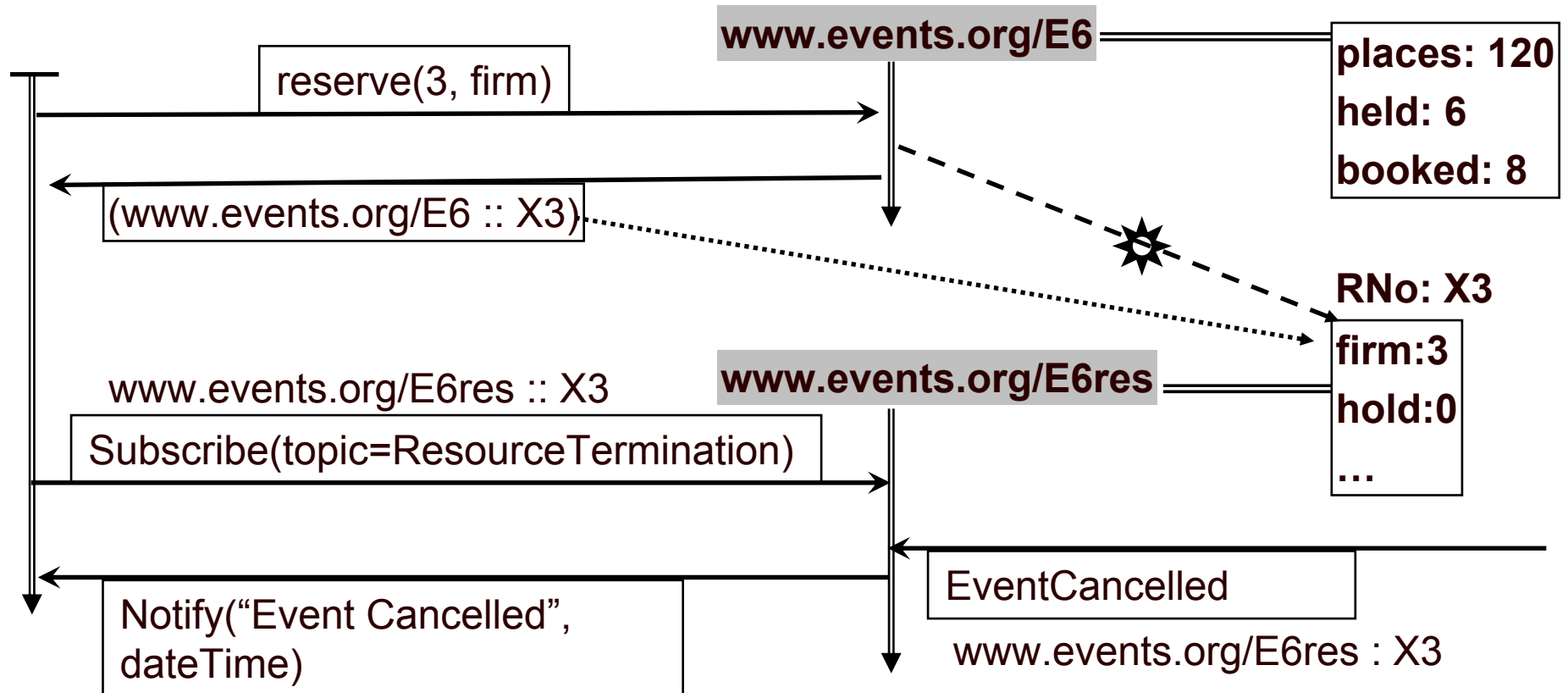
- Has state that Comprises a set of state data
- Has a well-defined life-cycle
- Is associated with one or more web services
- Has a globally unique identifier
- **Might offer notification of state changes**
- (Is similar to an object in O-O architecture)





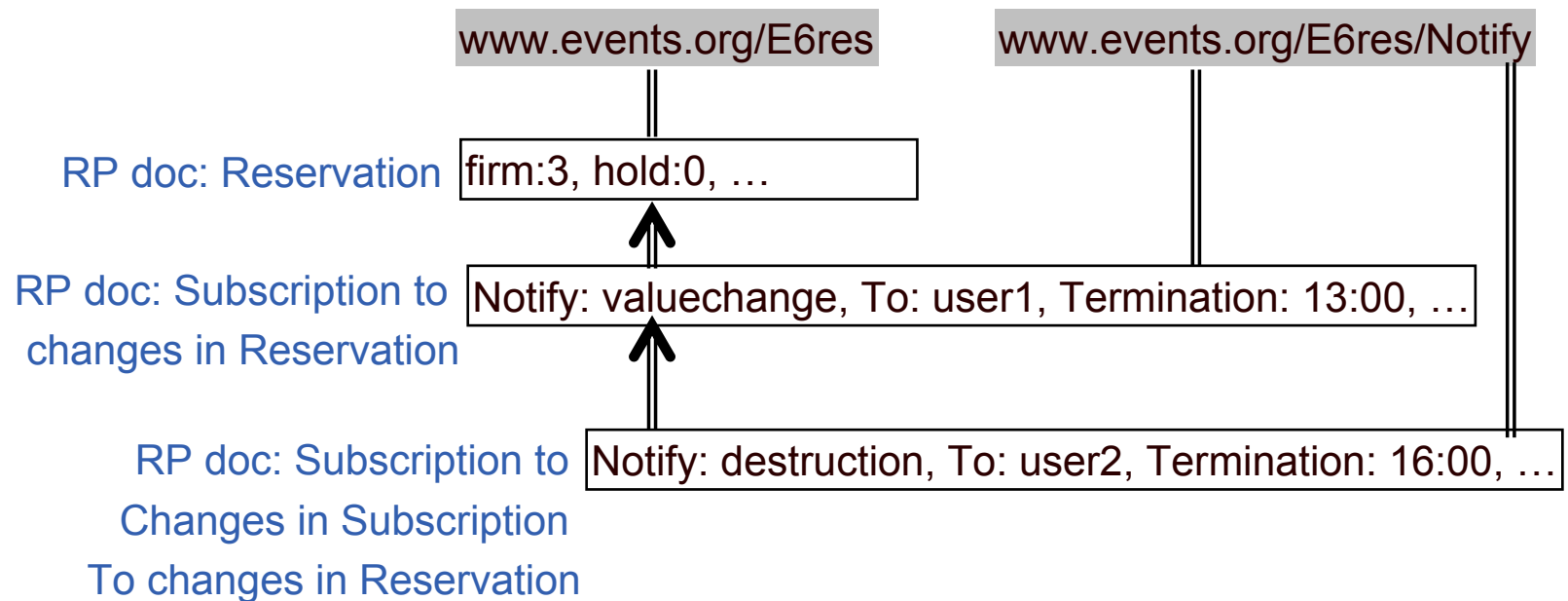
My boss (**Subscriber**) asks a press-cutting service (**Publisher**) to notify me (**Consumer**) of articles on GRIDS (**Topic**) in the popular press (**Producer**)

- Topic Space - a forest of topic Trees
- Producer – generates notification messages for Consumers
- Publisher – distributes notification messages according to subscriptions
 - Can combine Producer and publisher - same service generates the event and sends it to the subscribers
 - If separated - Publisher is a “Notification Broker”
- Subscriber - creates a subscription for a consumer in a Publisher
- Consumer - receives notification messages
 - Can combine subscriber and consumer - subscribe myself



- **Original creation of reservation resource**
- **Subscribe to the resource – for ResourceTermination Topic**
- **Resource notifies subscriber**
- **Can similarly subscribe to being notified of value changes for the resource.**

- **Following the WS-Notification standard requires –**
 - The subscription for Notification of a change to the (reservation) resource is itself a WS-resource, so
 - Must have a resource properties (R.P) document
 - Subscription R.P. document is different form reservation R.P document
 - So must be a different portType – therefore a different service endpoint?
 - Could itself support change notification



A stateful (WS) resource

- Has state that Comprises a set of state data
- Has a well-defined life-cycle
- Is associated with one or more web services
- Has a globally unique identifier
- Might offer notification of state changes
- **(Is similar to an object in O-O architecture)**



Differences from Object-Oriented Architecture

- **Explicit State**
 - **Exposed State is expressible as an XML document which in principle is retrievable and updatable**
 - **In O-O an object is defined by the operations on it and the affect of those operations on future operations**
 - **The type of a WS-resource is**
 - **More the type of its resource properties document**
 - **Rather than the signatures of its operations**

- **Scheduled Destruction**
 - 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)
 - So in WS need an additional mechanism – scheduled destruction
 - Gives the resource provider the right to re-claim physical resources used

- **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 operations

- **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 initial 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

- **WSRF builds on**
 - WS-Addressing – W3C submission Aug 2004 (rumoured changes)
- **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 (and uses) –**
 - **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

THE END