WSRP

Interfaces and Protocols Update

Agenda

- Summarize current status
- Discuss F2F goals/process
- Define terminology
- Present draft
- Describe unresolved issues
- Get your feedback

Current Status

- Requirements Document Coming Together
- Draft API Under Development
- Issues Are Being Discussed

F2F goals/process

- Goals:
  - Brief you on where we stand
  - Listen to your feedback
  - Resolve (many) issues during breakouts

Decision Criteria

- Functionality-Does the protocol support the function we desire?
- Generality – Is the protocol general enough to satisfy WSIA?
- Performance - Is the protocol expressed efficiently? Will it scale?
- Security- Does the protocol allow necessary data to be protected, authorized and secured?
- Usability - Is the protocol simple and self-describing?

Decision Criteria

- Flexibility – Does the protocol support differing usage styles?
- Vendor Extensibility - Is the protocol extensible?
- Protocol Extensibility – Is it easy/possible to revise the protocol.
- Modularity-Is the usage of the protocol subsettable?

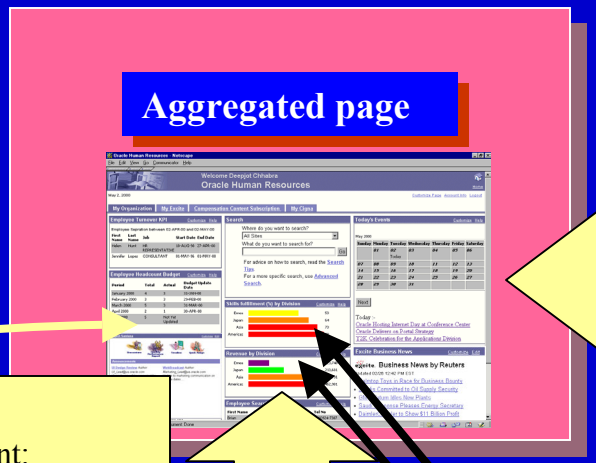
Today's Focus

- Big picture: the basic model
- What are the characteristics of the consumer environment that a producer runs in?
- What producer abstractions result?
- What are their lifecycles and how managed?
- Does the Draft reflect this fully and simply?

Let's Start With Terms

- Part 1: Define Elements in the System
- Part 2: Define Request Scopes between Elements.

Portal



Portal:

Application that presents an aggregated view of portlet content

Client:

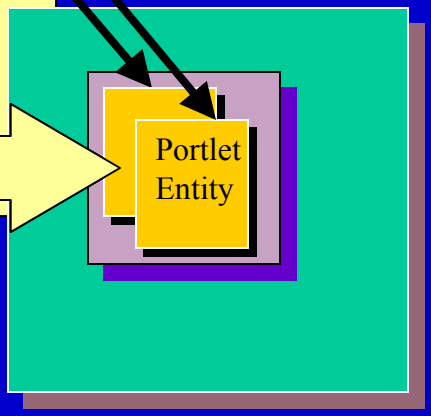
End user/browser talking to the portal

Client

Portlet Entity:

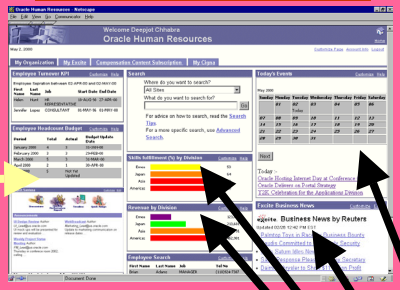
Abstraction that corresponds to a portlet instance.

ance:
rence:



Portal

Aggregated page



Client

Portlet Producer:

The WSRP (web) service that a… communicat… interact with… entity. The… producer may… represent 1 or m… portlet entity types.

Portlet Entity Type: The kind of portlet entity. E.g. stock portlet, weather portlet

Portlet Producer

Portlet Entity

Part 1: Terminology summary

- **Client**: end user/browser talking to the portal.
- **Portal**: application that presents an aggregated view of portlet content.
- **Portlet Instance or Reference**: specific instance/reference of a portlet within the portal.
- **Portlet Implementation**: implementation that generates specific content meant to be aggregated by a portal.

Part 1: Terminology summary

- **Portlet Entity**: The portlet producer abstraction that corresponds to a portlet instance.
- **Portlet Producer**: The WSRP (web) service that a portal communicates with to command a portlet. The portlet producer may represent 1 or more portlet entity types.
- **Portlet Entity Type**: A specific variety of portlet entity.

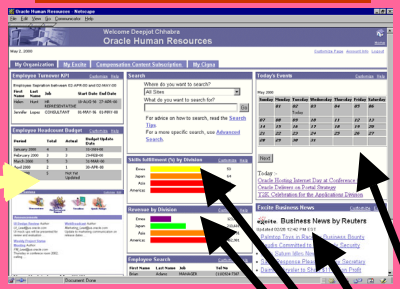
Part 1: Terminology Q & A

Part 2: Request Scopes

- Request Scopes define a meaningful group of requests between a portal and a portlet producer.
- Requests scopes are interesting because:
  - Define scope of (transient) state
  - Define lifecycle of (transient) state

Portal

Aggregated page



Lifecycle:
Init: first request from portal

Destroy: portal application "closed" or timeout.

Consumer Scope:

All requests from a Portal to a Portlet Producer for all client requests.

Client

Data:
Producer wide. Commonly used for shared resources such as database connections.

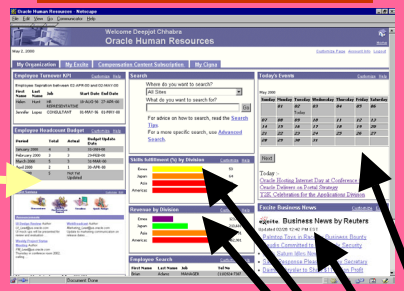
tlet Producer

Portlet Entity

Many Clients

Portal

Aggregated page



Lifecycle:
Init: first request from portal from a given user.

Destroy: portal user session ends or timeout.

Session Scope:

All requests from a Portal to a Portlet Producer stemming from requests from a single client.

Data:
Producer wide. Commonly used for sharing state with other entities of your own or other types.

Client

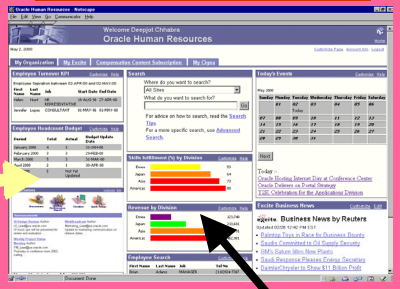
Many Clients

tlet Producer

Portlet Entity

Portal

Aggregated page



Lifecycle:
Init: first request from portal to this entity

Destroy: portal "closes" entity, portal user session ends or timeout.

Entity Scope:

All requests from a Portal to a Portlet Producer concerning a particular portlet entity.

Data:
Entity specific. Derivative of Consumer Scope. For entity private resources such as a newsfeed cache.

Client

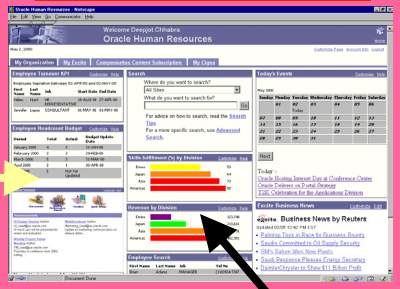
Many Clients

tlet Producer

Portlet Entity

Portal

Aggregated page



Lifecycle:
Init: first request from portal to this entity within a session

Destroy: portal "closes" entity or timeout.

Entity Session Scope:

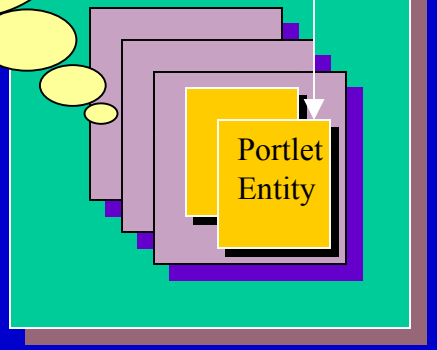
All requests from a Portal to a Portlet Producer concerning a particular portlet entity stemming from requests from a single client.

Client

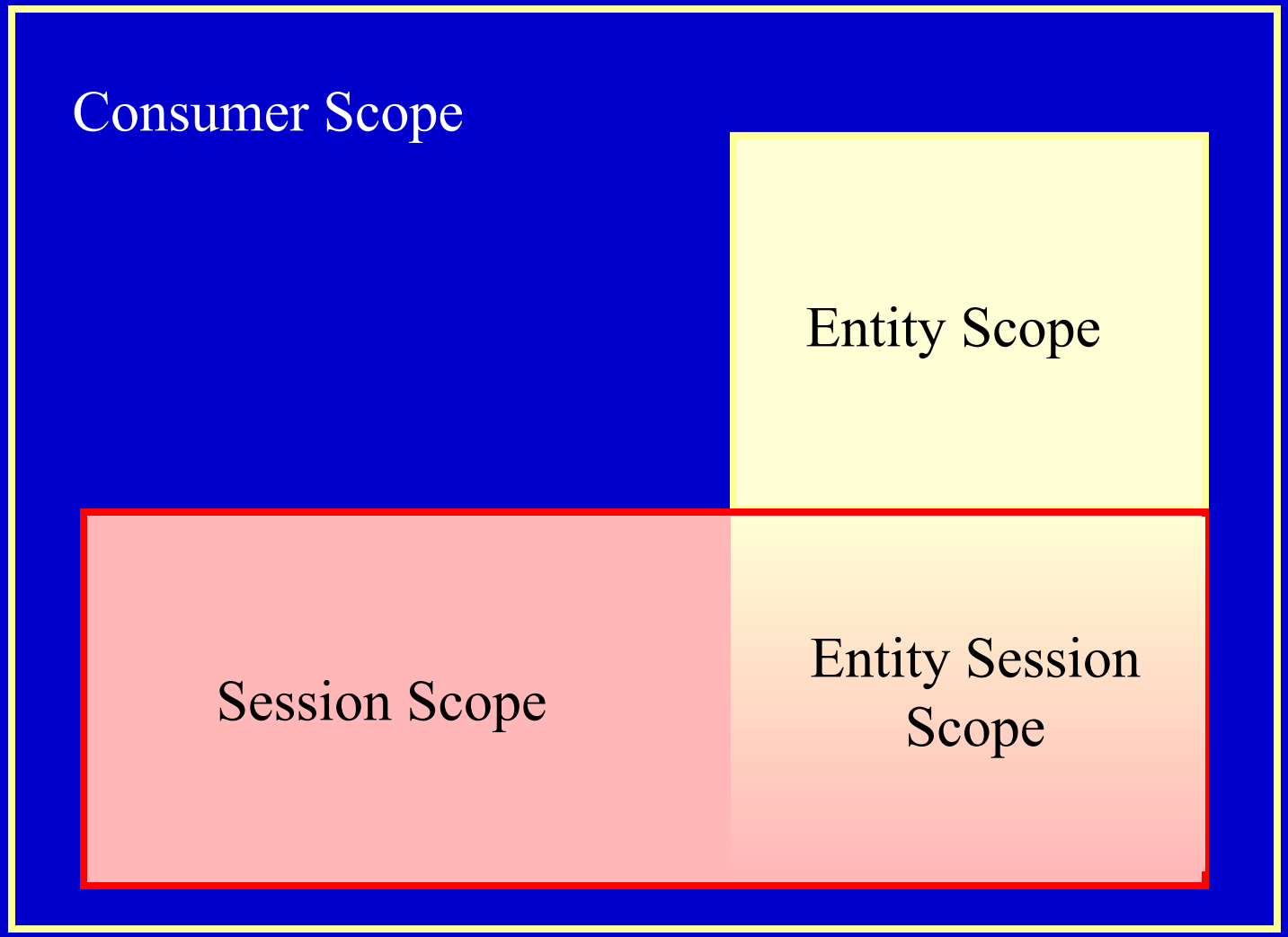
Data:
Entity specific. Derivative of Session Scope. For per user entity private data such as current state.

Many Clients

tlet Producer



Producer Data Scopes



Part 2: Terminology summary

- **Consumer scope**: All requests from a Portal to a Portlet Producer for all client requests.
- **Session scope**: All requests from a Portal to a Portlet Producer stemming from requests from a single client.

Part 2: Terminology summary

- **Entity scope**: All requests from a Portal to a Portlet Producer concerning a particular portlet entity.
- **Entity Session scope**: All requests from a Portal to a Portlet Producer concerning a particular portlet entity stemming from requests from a single client.

Part 2: Terminology Q & A

Strawman API

WSRP

Publish, Find, Bind, MetaData Update

Agenda

- Summarize current status
- Present existing product metadata

Current Status

- Publish and Find
  - Default is portal knows the location
  - Requires portlets be self-describing
  - UDDI will be preferred registry
- Bind
  - registerConsumer() for explicit registration
  - getServiceDocument() to describe
- MetaData
  - Details pending more detailed protocol definition
  - Start by reconciling existing vendor data

Oracle's MetaData

- Portlet Producer level:
  - usesSession: [boolean] Indicates producer maintains session state.
- Portlet level:
  - id: [number] our "portlet entity type"
  - title: [string] Portlet display name
  - shorttitle: [string] Short form of portlet's name

Oracle's MetaData

- Portlet level:
  - description: [string] Displayable description
  - timeout: [seconds]
  - timeoutMessage: [number]
  - acceptContentType: [mime type] multivalued
  - hasAbout, hasHelp, showDetails, showEdit, showEditDefault, showLink, showPreview
  - minEditAuthLevel [*PUBLIC* | *WEAK* | *STRONG*]

Oracle's MetaData

- Portlet level:
  - defaultLocale: [locale]
  - mobileFlags: [enumerated]

IBM's MetaData

- T-model ID: indicates that this is a WSRP service
- PortletClassID - portlet id to create an instance
- Service Name: e.g. "Stock Quote Service"
- Supported locales: The list of supported locales
- DefaultLocale
- Service Titles: Titles [in all locales] of the service
- Service Short Titles: Short Titles [in all locales]
- Service Descriptions: Descriptions [in all locales]

IBM's MetaData

- Supported Modes: e.g. view, edit, config, help
- Supported Markups: e.g. HTML, XHTML, WML·
- Cachability:
- Instance awareness:
- Keywords ? Searchable keywords
- Supported View States - [Minimized, Normal, Maximized]
- Allowed mods - [Transcoding, Translation, Adaptation]

SilverStream's MetaData

- class-name
- display-name
- description
- preview-image
- lifetime
- enable-title-bar
- component-styles
- default-style

SilverStream's MetaData

- data-definition
- defaults
- categories
- supported-options
- run-role-map
- list-role-map
- component-style

Netegrity's MetaData

- name
- description
- default height & width
- -isResizable(boolean)
- icon(for use page-building interfaces)
- portlet version
- minimum portal version required(compatibility)
- Parameters

Netegrity's MetaData

- Parameters
  - parameter group
  - parameter
    - name
    - friendly name
    - default value
    - config mechanism(text entry, dropdown selection, custom(portlet draws UI))

Netegrity's Meta Data

- layouts
  - layout
  - name
  - isDefault(boolean)
- methods/actions
  - method
    - name
    - layout
    - errorlayout

WSRP Security Subgroup

F2F Update

June 25, 2002

Mark Cassidy

Security Considerations

- Authentication
  - Consumer -> Producer
  - End user security tokens(SSO)
- Integrity
  - Message content is not compromised
- Confidentiality
  - Unauthorized parties can’t see confidential information
- Replay & Denial of Service attacks
- Access Control
  - Business policy based on attributes/credentials of the requester

Security Subgroup Priorities

- Leverage established and emerging standards as much as possible
- Plug n Play

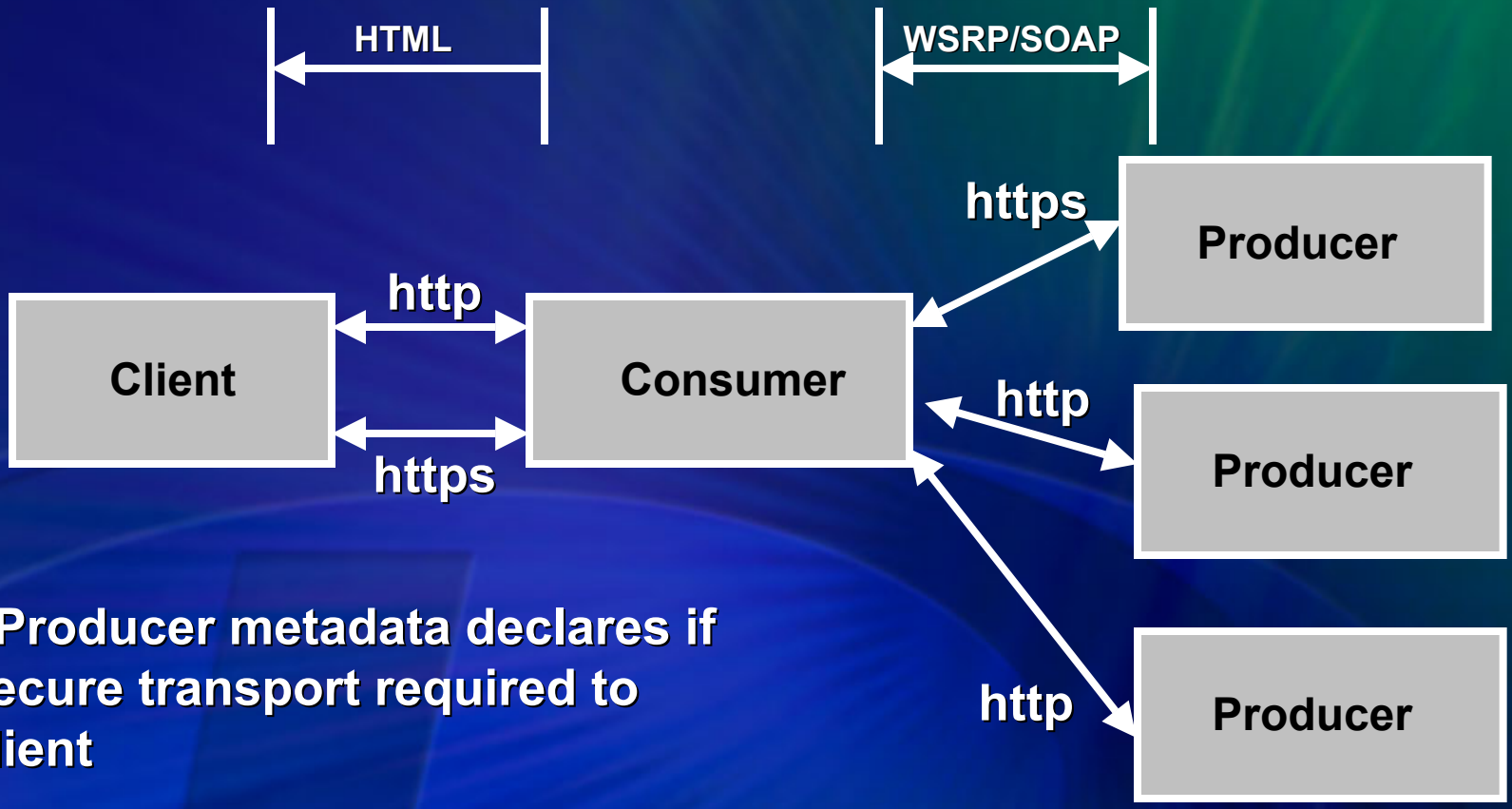
Security Mechanisms and Relevant Standards

- ***Transport-level***
  - **SSL/TLS(RFC2246)**
    - **authentication, integrity, confidentiality**
  - **Client certificates(x.509v3)**
    - **authentication**
  - **HTTP-Basic**
    - **authentication**
- ***Document-level***
  - **WS-Security**
    - **Authentication, integrity, confidentiality**
  - **XML-Signature**
    - **Integrity, authentication**
  - **XML Encryption**
    - **Confidentiality**
  - **SAML**
    - **Authentication, integrity, access control**

WSRP using Secure Transport

- SSL/TLS over HTTP
  - Client certificate for Consumer auth by Producer
- Benefits:
  - No impact on WSRP protocol
  - Widely used infrastructure
  - Addresses key integrity, confidentiality and authentication requirements
- Limitations:
  - Only secures point-point connections

Secure Transport Scenario



- Producer metadata declares if secure transport required to client
- Consumer page uses https when any portlet on page requires https
- No metadata required for https between Producer and Consumer

Challenges with Document Level Mechanisms

- Typically, several mechanisms used together to achieve overall goals
- Many different deployment scenarios possible
- No current standards for:
  - Describing encryption and signature capabilities
  - Describing policy for signing and encryption

Subgroup Status: Document Level Security

- Important to have some form of support for signatures and encryption in v1.0 spec
- WS-Security provides support for use of signatures and encryption in SOAP messages(via security header blocks)
- WSRP policy for use of sig & enc
  - Tightly constrained to maximize plug & play
  - Replace with standard policy description when available
- Defer SAML scenarios for now due to added complexity

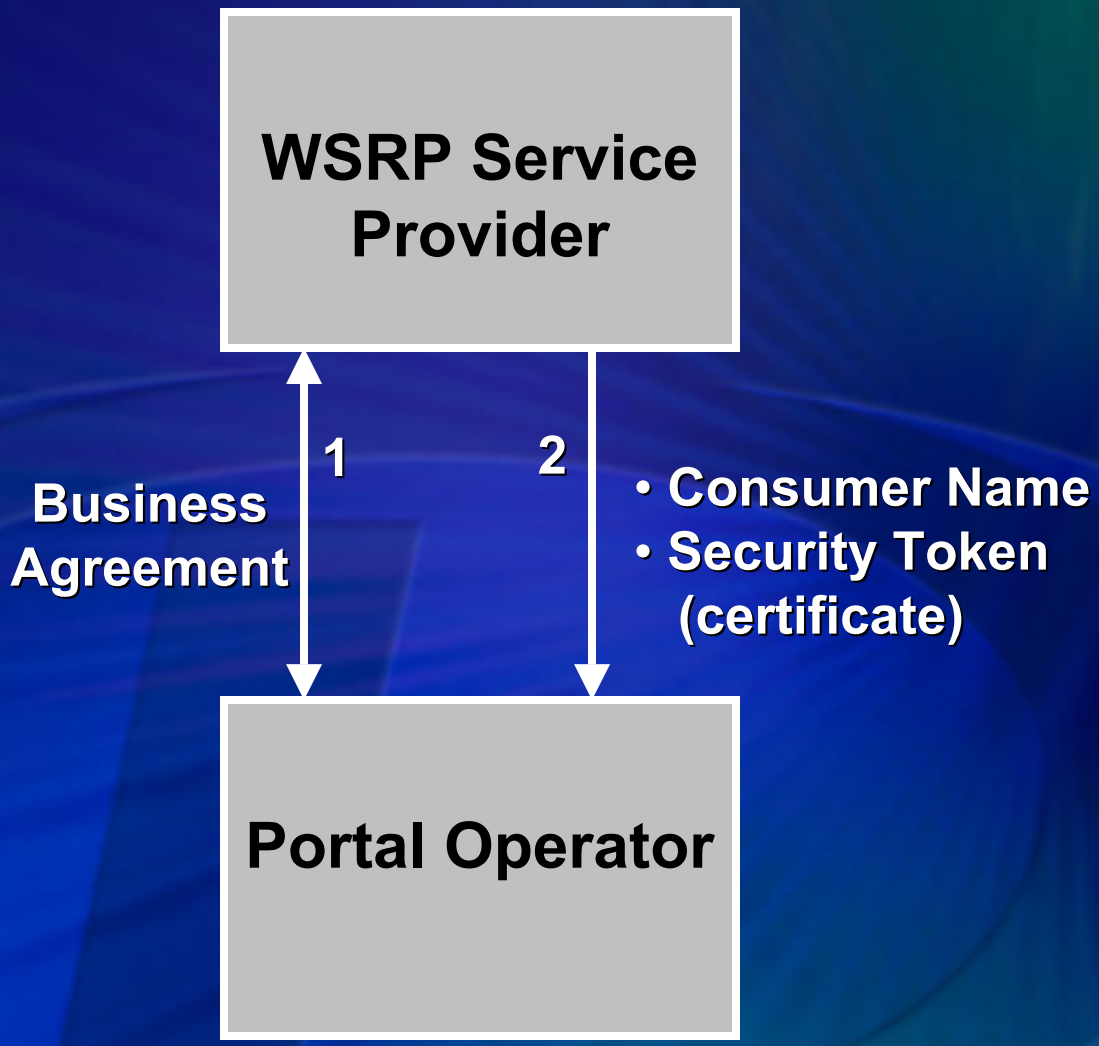
Impacts on WSRP Metadata

- Producer policy for signature
  - When specified, all consumer requests must be signed
  - Scope of signature is entire message
  - Defer requiring client signatures
- Producer capabilities for signature algorithms
- Producer policy for property data encryption
  - Only element data is encrypted, not elements themselves
- Producer capabilities for encryption algorithms

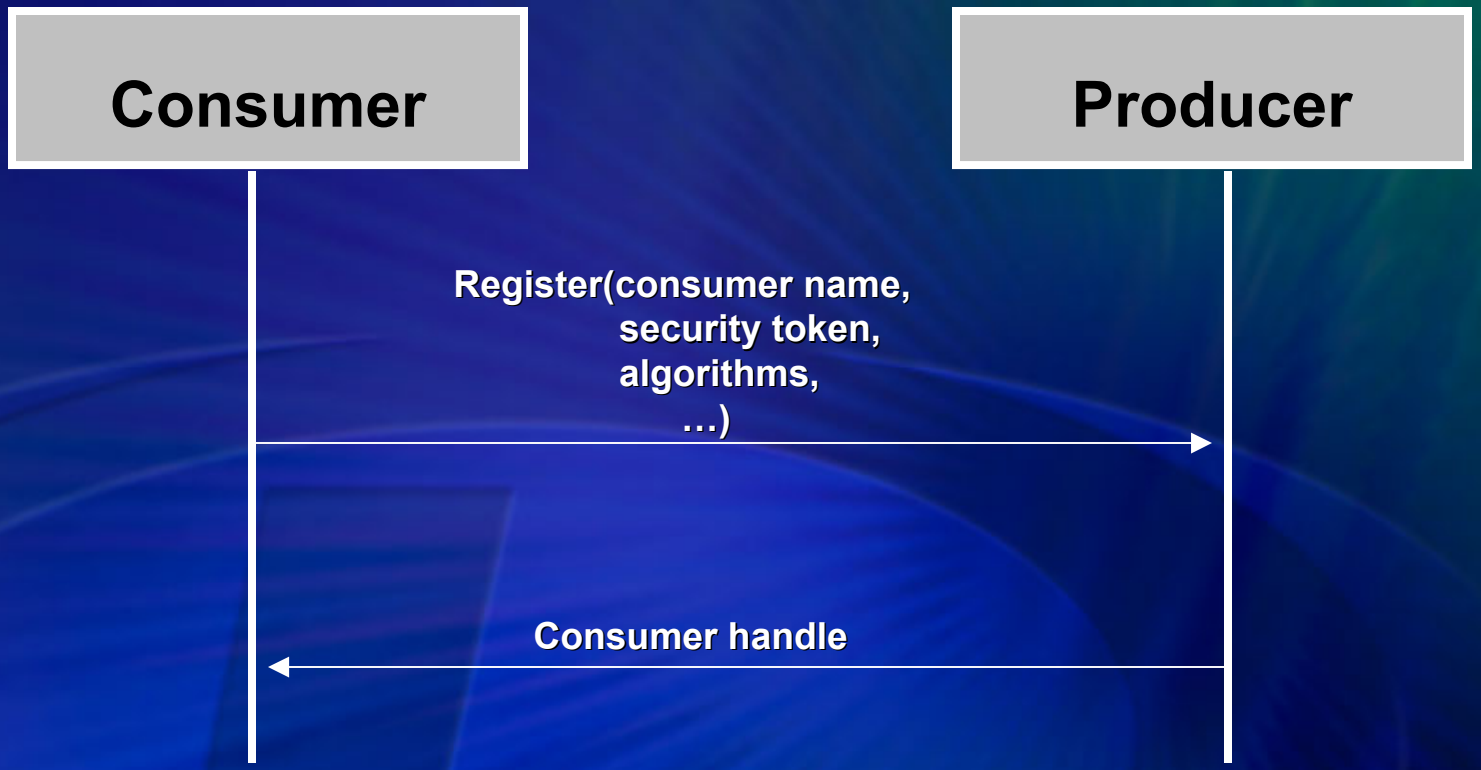
Impacts on WSRP Protocol

- Consumer must specify its signature and encryption algorithm capabilities in the registration operation
- Data objects carried in WSRP messages may be encrypted

Business Relationship

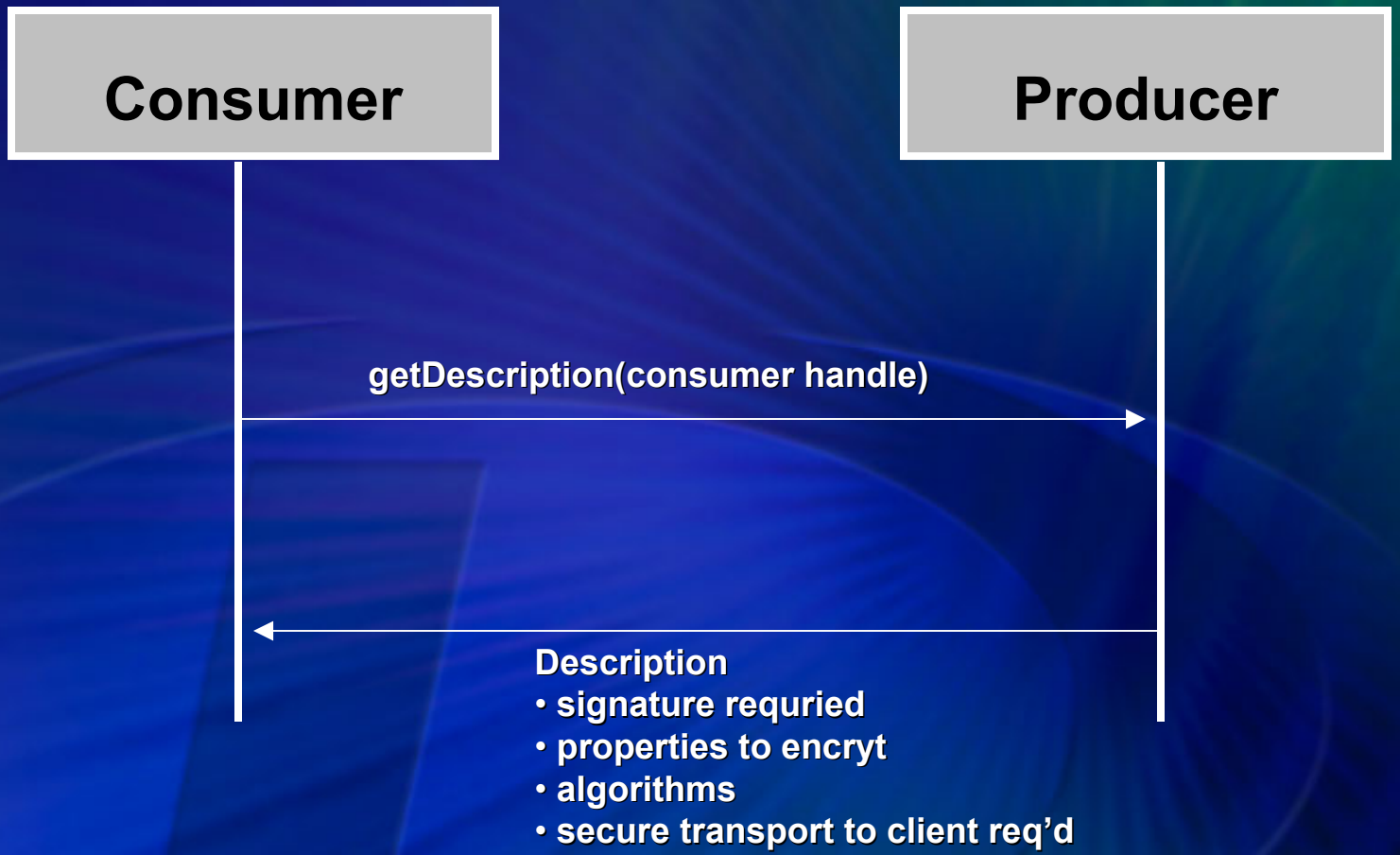


Consumer Registration

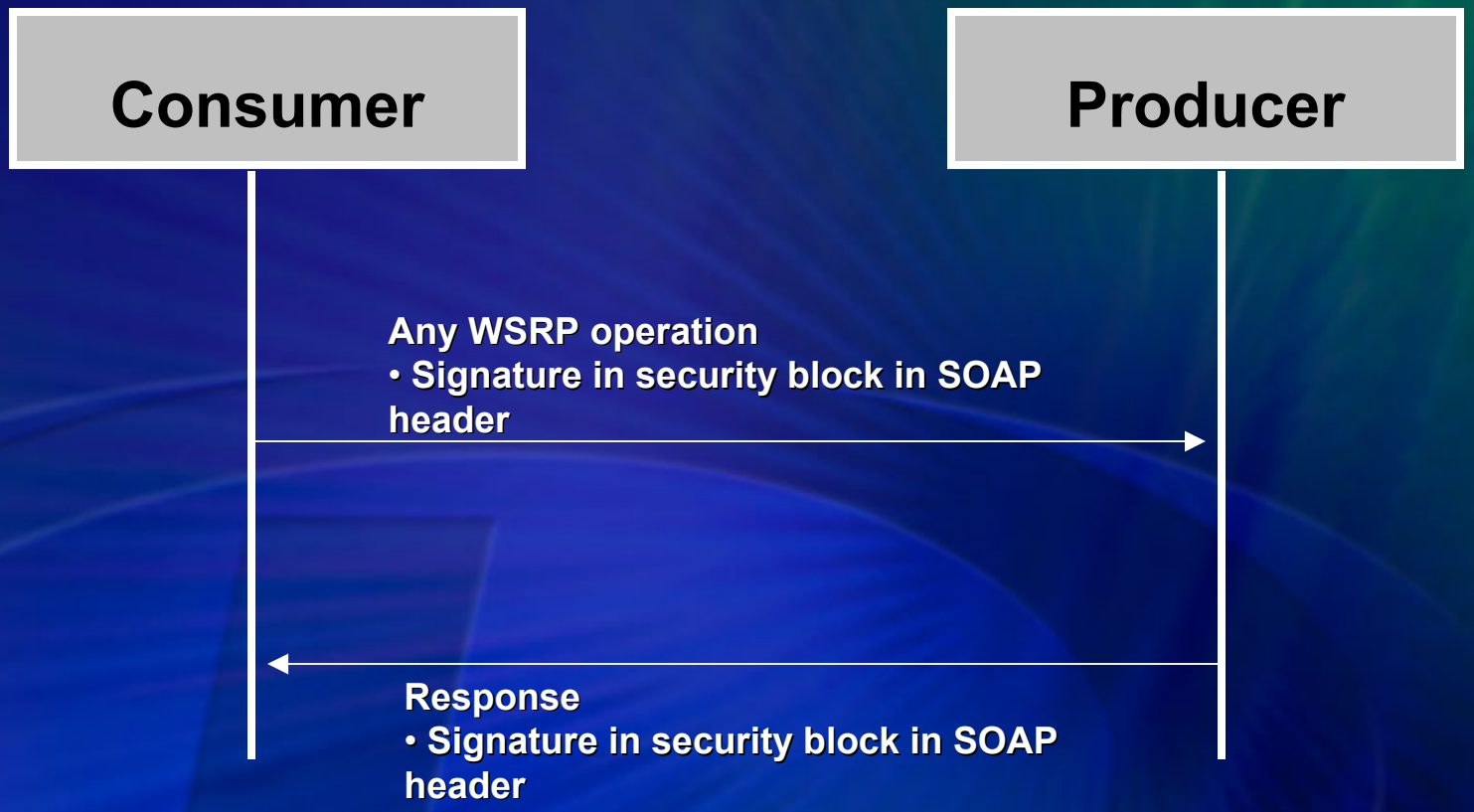


WS-Security describes how to include a security token in a SOAP header.

Get Service Description

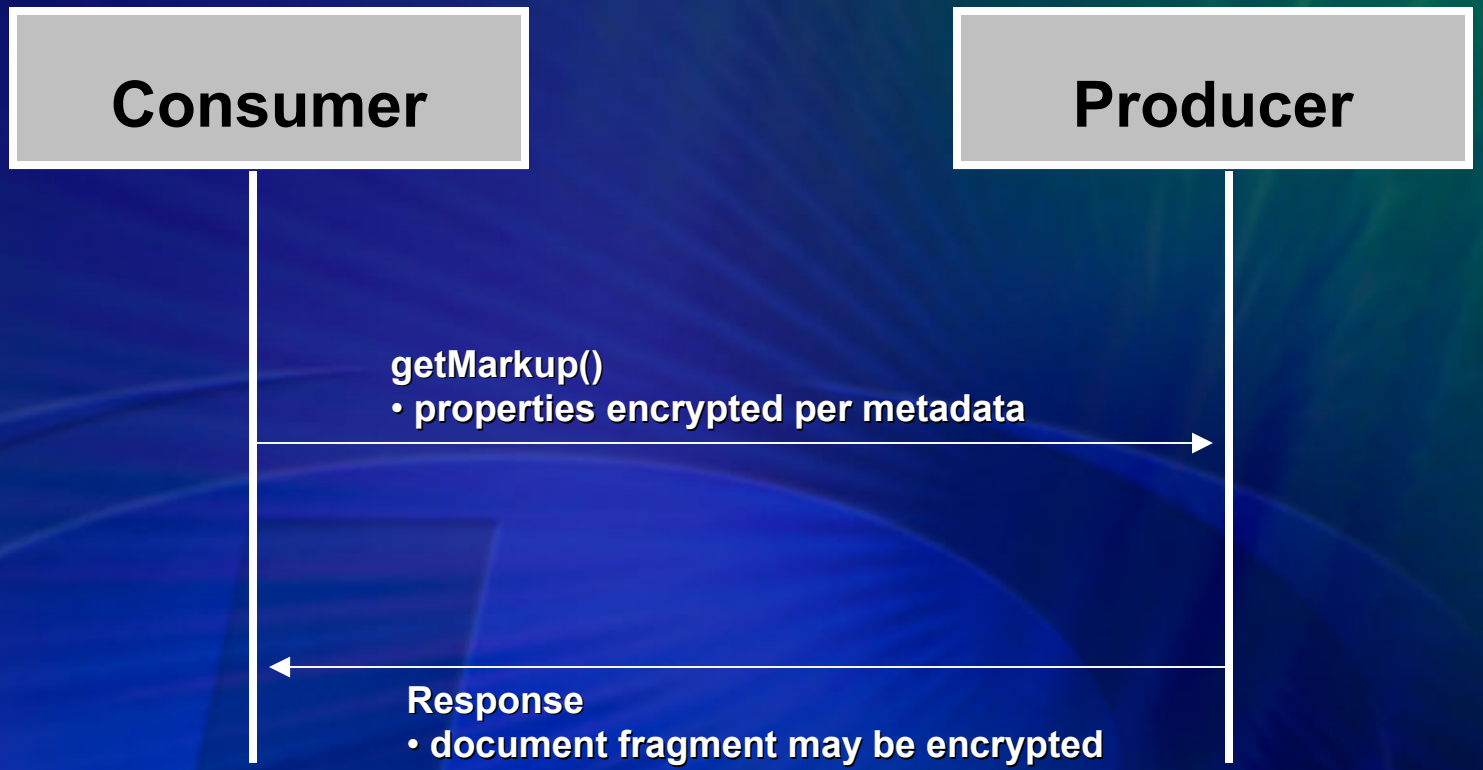


Signatures



- Signatures are self-describing
  - Algorithm & key info within the signature
- Producer can't enforce that Consumer validate signatures on response messages

Encrypted Data



- Encrypted data elements are self-describing
  - Algorithm referenced within the element

Encrytion & Digest Keys

- KeyInfo is(reference or inclusion) is carried in sub-elements of WS-Security header.

Discussion Points

- **WSRP-specific policy descriptions for signature and encryption**
  - **Eventually this will be replaced with a standard policy description mechanism**
- **Support for signature & encryption not broadly available in SOAP stacks**
  - **WSRP producers and consumers will need to supplement SOAP stack in short term**
- **WS-Security not yet an open standard**
  - **Working group currently IBM, Verisign, Microsoft**

Access Control & Roles

- Consumer may implement access control to WSRP operations in a declarative fashion with no impact to protocol or metadata
- Producer may implement programmatic access control via role definitions
  - Consumer is responsible for mapping portlet roles to portal users
  - Role data is passed to producer with service request
- Issue: are standard roles required?
  - No agreed on semantics for standard WSRP roles

User Profile

- It has been suggested to define a 'user profile' data object for WSRP, and this object would be passed from Consumer to Producer
- Profile elements could draw from existing directory or other suitable standards
- Issue: not yet defined how this data object would be used by a producer, or how its data elements relate to entity properties

Privacy Preferences

- Legal requirements and end-user-specified constraints for sharing personal information needs to be considered
- Initial opinion of the subgroup is that adhering to these constraints is a responsibility of the Consumer; possible impact to protocol is limited to consumer informing the producer of client's privacy profile

Intermediary Scenarios

- While WS-Security is specifically focused on supporting message routing through intermediaries, we need to do more investigation on WSRP scenarios involving intermediaries.

Discussion

WSRP – Markup

Chris Braun
SilverStream
cbraun@silverstream.com

Charter

- Define standard mechanisms to allow common look and feel across aggregated portlets
- Specify markup rules that define valid portlet markup fragments for:
  - HTML
  - XHTML
  - Others (WML, cHTML, VoiceXML, …)
- Define standard mechanisms for URL rewriting and namespace encoding.

Visual Themes

- Mechanism by which portlets can follow common look and feel.
- Use CSS for HTML and XHTML
- Predefined well known classes prefixed “wsia”
- Defined in an appendix of the WSRP specification.
- Other Markups?

WSIA CSS Classes

- Fonts, Messages, Sections, Forms, Menus, Portlet, Colors
- Merged from list provided by various vendors (IBM, WSUI, SilverStream, Plumtree, and Oracle)
- Examples:
  - .wsia-font
  - .wsia-msg-error
  - .wsia-section-header
  - .wsia-form-button

CSS Outstanding Issues

- Some proposed classes with no home. Find a home or eliminate them.
- Should background and text styles be separate classes? Example:
  - wsia-section-background is not needed because wsia-section can be used to define background styles.

Discussion

Markup Fragment Rules

- Concentrate on HTML/XHTML
- Some tags can not be part of the portlet markup. Such as <title>. Disallowed.
- Some tags may be supported by browsers but are denied by the HTML spec. Such as <link>. Discouraged
- We will not require that the container validate portlet markup.

Problem Tags

Disallowed	Discouraged
base	link
body	meta
frame	style
frameset	
head	
html	
title	

Discussion

Markup Fragment Decomposition

- We will not specify a mechanism for which portlets can modify the containing pages <head> tag.
- So how do portlets use…
  - <link>, <meta>,<style>
  - <xforms:model>
- Consider added extensibility mechanism so that container vendors can allow the portlet to modify other areas of the containing page.

URL Rewriting

- Need a way for container to intercept portlet actions and proxy resource requests.
- A proxy resource is a resource that is served by the consumer, but has originated from another location.

Rewriting Scenarios

1. **Using a prefix sent by the consumer**

   The consumer sends a prefix with the request that the producer will use to do the URL boundary demarcation. The consumer then parses the markup looking for the prefix it provided.

2. **Using a predefined prefix**

   All portlets use a predefined prefix, which is part of the specification, to do the URL boundary demarcation. The consumer then parses the markup looking for the well known prefix.

Rewriting Scenarios (cont)

3. **At the consumer side**

   The consumer automatically parses markup and heuristically determines URL boundaries and does the necessary rewriting automatically.

4. **At the producer side**

   The consumer sends the URL prefix to use to the remote portlet, allowing it to do the rewriting itself on all the necessary URLs. The markup sent back to the consumer is then ready for immediate inclusion in the page, with no parsing necessary.

URL Types

- Fully Qualified (no rewriting)
- Relative
- Portlet Action
- Proxy
- Action to Other Portlet

Action URLs (Scenario 4)

A. Consumer sets a entity property:
ActionURL = `http://Consumer.com?WSIA_entity=7,WSIA_actionName={actionName}{params}`

B. Entity's
URL: `http://Consumer.com?WSIA_entity=7,WSIA_actionName=DoTransaction,parm1=foo`

C. Consumer passes URL as is:

Action URLs (Scenario 4)

D. End-User browser sees:
`http://Consumer.com?WSIA_entity=7,WSIA_actionName=DoTransaction,parm1=foo`

E. Post to Consumer: Consumer does a lookup of the entity and calls Producer

F. SOAP invocation to Producer:
`performAction(entityHandle,..., DoTransaction, ...)`

Action URLs (Scenario 2)

A. Entity's URL: `{StartToken}{urlType = action}{actionName}{EndToken}`
B. Consumer rewrites URL: Stores Entity's URL, and generates the final URL to reference the action
C. End-User browser sees: `http://Consumer.com?WSIA_urlref=5`

Action URLs (Scenario 2)

D. Post to Consumer: Consumer does a lookup and calls Producer
E. SOAP invocation to Producer: `performAction(entityHandle, ..., actionName, ...)`

URL Rewriting Outstanding Issues

- Scenarios 2 vs. 4
- Well defined semantics
- Local vs. WSRP Rewriting
- Actions across Portlets

Discussion

Name-spacing

- Aggregation of multiple portlet markup from different Portlet entities can potentially result in naming conflicts
  - Named tags (forms, fields, buttons, etc)
  - JavaScript functions and variables
- This problem is similar to URL Rewriting. We should consider using similar mechanisms to solve both situations.

That’s It