

---

# Provenance Situations: Use Cases for Provenance on Web Architecture

# Provenance and Web Architecture: Consider Five Diverse Provenance Situations

---

1. SIMPLE OH YEAH SITUATION
  - User retrieves a document, then clicks on “oh yeah” button, then site returns a provenance record
2. LICENSING SITUATION
  - User retrieves a document (eg an image), then wants to check permission to use
3. REFERRAL SITUATION
  - Site refers queries about provenance in terms of pointers to other site’s provenance facilities
4. REPEATED QUERIES SITUATION
  - Service repeatedly queries a site, wants provenance for all the answers
5. VERSIONING SITUATION
  - User retrieves a document, then wants to see its provenance, but the document has been updated in the original site (its provenance as well)

## Five Diverse Provenance Situations --

### 1) SIMPLE OH YEAH SITUATION:

User accesses a document, then clicks on “oh yeah” button

---

#### ■ OPTIONS:

- *Embedded provenance*: Documents could have provenance included when available and returned when they are accessed
  - By value: provenance records included in the document
  - By reference: a URL to retrieve provenance records
- *On-demand provenance*: Site could return provenance upon request
  - Convention: a mechanism to access provenance directly given object handle
- ??

#### ■ ISSUES:

- Provenance records could be quite large
- Provenance records often refer to entities: people, institutions, web objects, etc
  - Need unique identifier for them?
- ??

## Five Diverse Provenance Situations --

### 2) LICENSING SITUATION

User retrieves a document, then wants to check permission to use

---

#### ■ OPTIONS:

- By-value provenance
- By-reference provenance
- ??

#### ■ ISSUES:

- Provenance record needs to be accessed selectively
  - License and Copyright may be a tiny aspect of it
- Need to verify that what is stated the provenance record is truthful, at least by verifying that there is a (legally binding) entity that vouches for it
  - Digital signature
- ??

## Five Diverse Provenance Situations --

### 3) REFERRAL SITUATION

User may want to further research provenance, by following links in the provenance record provided

---

#### ■ OPTIONS:

- Self-contained provenance: Site offers a complete provenance record (can contain URIs but not to other provenance records)
- Delegated provenance: Site refers queries about provenance in terms of pointers to other site's provenance facilities
- ??

#### ■ ISSUES:

- Requires provenance to be accessible on its own (have unique identifier)
- ??

## Five Diverse Provenance Situations --

### 4) REPEATED QUERIES SITUATION

Service repeatedly queries a site, wants provenance for all the answers

---

#### ■ OPTIONS:

- *Individualized provenance*: A provenance record is sent for each query (embedded or on-demand)
- *Shared provenance*: A provenance record is sent once with the first query and given a unique ID, the site can refer to that record for subsequent queries
- *Bulk provenance*: Site may associate provenance records to types of queries (so the record applies to all query instances of that type)
- ??

#### ■ ISSUES:

- ??

## Five Diverse Provenance Situations --

### 5) VERSIONING SITUATION:

User retrieves a document, then wants to see its provenance, but the document has been updated in the original site (its provenance as well)

---

#### ■ OPTIONS:

- By value
- By reference:
  - Provenance is retrievable per document per timestamp (can access old provenance)
  - Provenance queries return provenance plus latest version of document
- ??

#### ■ ISSUES:

- Identifiers for different versions of the document, deltas
- Managing deltas (ie small updates) in provenance records
- ??

---

# Provenance and the Web Architecture

# Introduction

---

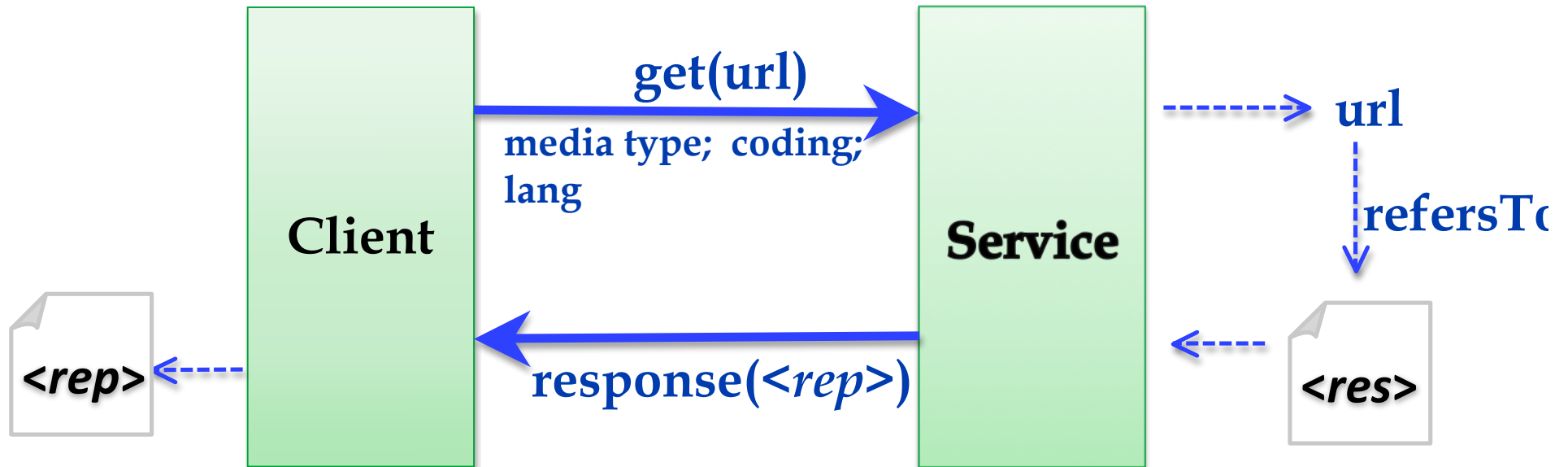
- Provenance situations = use cases for provenance
- We:
  - consider several communication patterns in the context of the Web Architecture
  - outline possible ways of integrating provenance
- **Our goal is to seek feedback!**
  - Here, we assume the existence of an ontology for provenance

# Considered Patterns

---

- HTTP Request Response
- Obtain provenance:
  - Provenance service
  - SPARQL Query
- Web Service Request/Response (additional material)

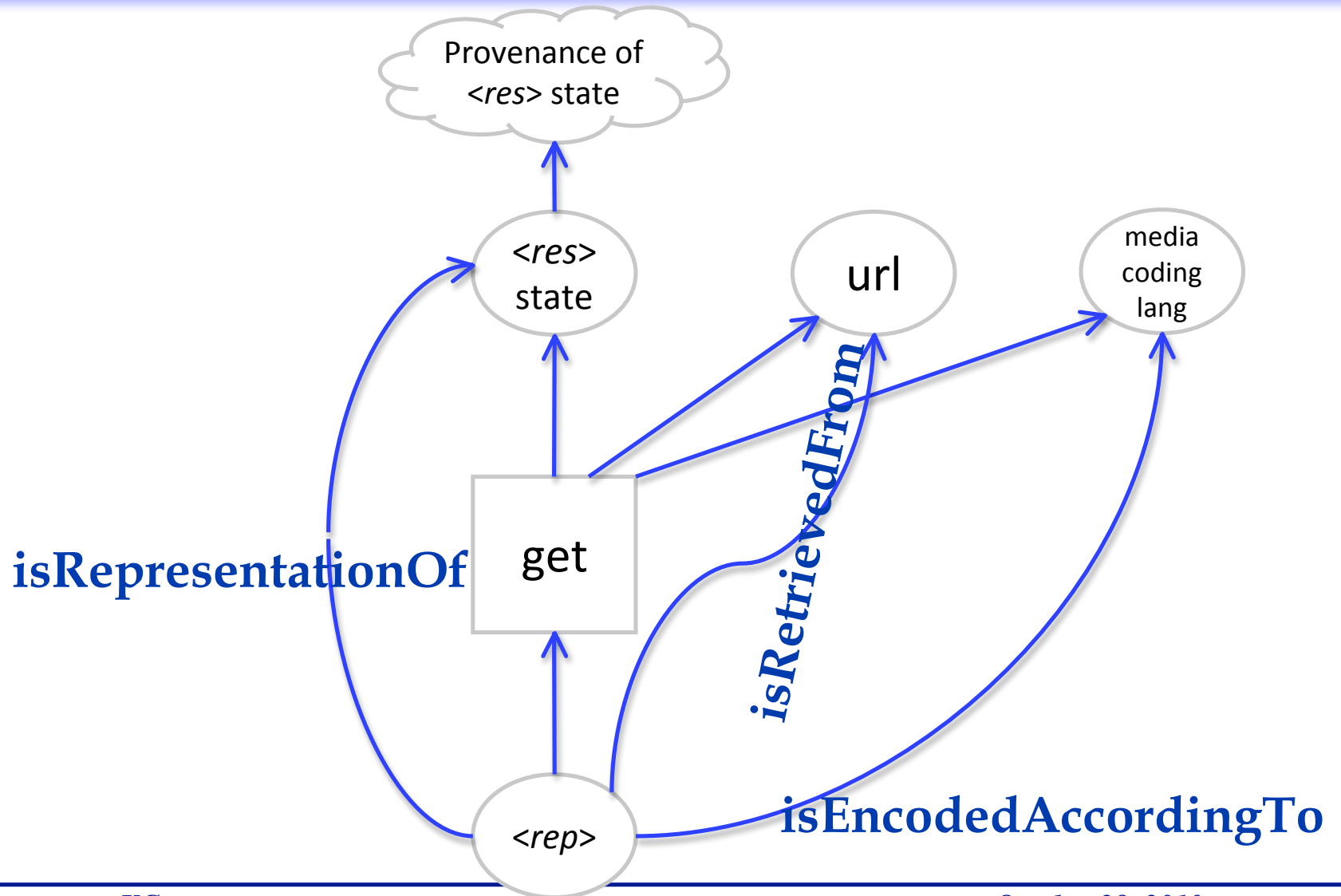
# HTTP Request/Response



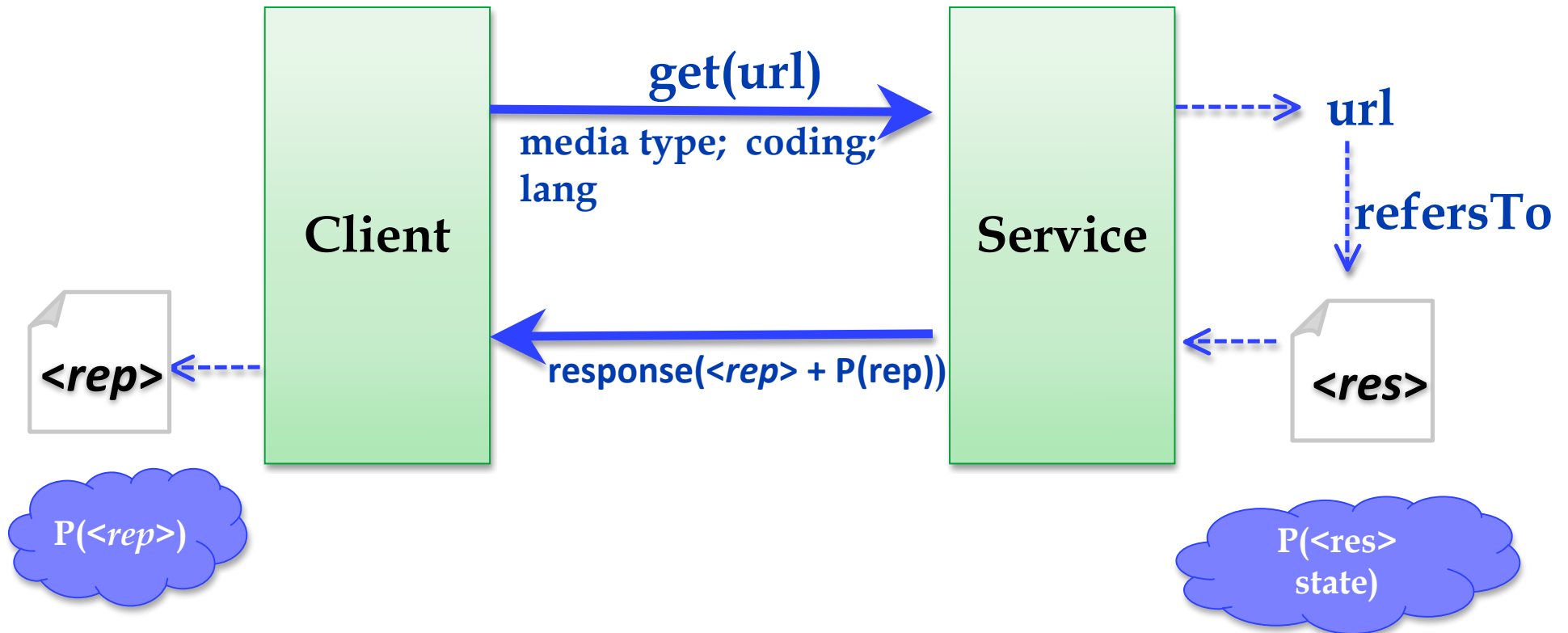
In response to a `get(url)` request, the client obtains `<rep>`, a representation of the state of the resource `<res>`, which existed at the time the request was processed. The representation is a negotiable serialization of the resource state, according to media type, coding, and language

**The client may wonder what the provenance of `<rep>` is?**

# Provenance of a Representation



# HTTP Request/Response



In response to a `get(url)` request, the client obtains a representation of a resource, and the provenance of the representation.

# Provenance Passing

---

## ■ By value

- + Provenance is always in sync with exchanged representation
- Provenance may be much bigger than representation
- All representations of a static resource share a common history  
 $P(\langle res \rangle \text{ state})$

## ■ By reference

- + Client receives a url for retrieving provenance (small size)
- Burden on server to maintain and keep provenance for all delivered representations
- Particularly problematic for dynamically generated contents

# Where to insert provenance?

---

## ■ HTTP Level

- HTTP header
  - Provenance:  
`http://example.com/doc?prov_v20056`
  - Provenance: `<<provenance by value>>`
    - (implementation limit on header size!)
- In body
  - Multipart MIME message (is this feasible?)

## ■ Document level

- RDFa embedded in html document
  - Can embed provenance by value or by reference (see next two slides)
- Any media type with metadata capabilities, e.g. pdf, jpeg, exif, etc

## HTML with RDFa Metadata

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
      xmlns:cc="http://creativecommons.org/ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:opmv="http://purl.org/net/opmv/ns#">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Surf's Up!</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<div id="wrapper">
  <div id="content">
    <div id="header">
      <h1><a href="#">SURF BLOG</a></h1>
    </div>
    <div id="main">
      <h2> Kelly Slater on the New Age </h2>
      <div typeof="opmv:Artifact" about="#quote">
        "That's the future of surfing," said Kelly Slater, 38,
        a nine-time world champion from Cocoa Beach, Fla.
        "It's really in the air. The deepest barrels that are
        ever going to be ridden have already happened.
        Probably the best carving that's ever going to be done
        is being done now or it's been done."<br>
        <span rel="opmv:wasGeneratedBy">
          <span about="#aggregation" typeof="opmv:Process">
            from: <a rel="opmv:used" href="http://www.nytimes.com/2010/03/14/sports14surf.html">
              Surfing's Next Generation Takes to the Air</a>
            <br/><br/>
            <i>Post by <span property="opmv:wasPerformedBy">John Smith</span></i>
          </span>
        </div>
      </div><!--main-->
    </div><!-- content -->
  </div><!-- wrapper -->
</body>
</html>
```

# Provenance by reference

---

```
<html xmlns=http://www.w3.org/1999/xhtml
      xmlns:pr="http://example.org/provenance#">
<head>
<link rel="pr:provenanceAt"
      href="http://example.com/doc?prov_v20056" />
</head>
...
</html>
```

## +/- Analysis

---

	HTTP Level	In Message Contents
By Value		
By Reference		
Mixed (some by value, rest by ref)		

**Is there a possibility of provenance negotiation?**

# How to obtain provenance?

---

- Given a news article URI:

- <http://www.nytimes.com/2010/06/12/arts/12iht-melik12.html>

How can we find its provenance?

- Obtain all the provenance from a third-party “provenance service”, e.g. `provenance.com`

- Use HTTP Get
- Provenance service provides an *account* of the provenance of article
- Multiple such services can provide multiple accounts
- Provenance may be large

- SPARQL endpoint for selecting relevant provenance

```
SELECT ?r, ?l WHERE {  
    ?l a cc:License.  
    ?r opmv:wasDerivedFrom ?l }
```

# Conclusion

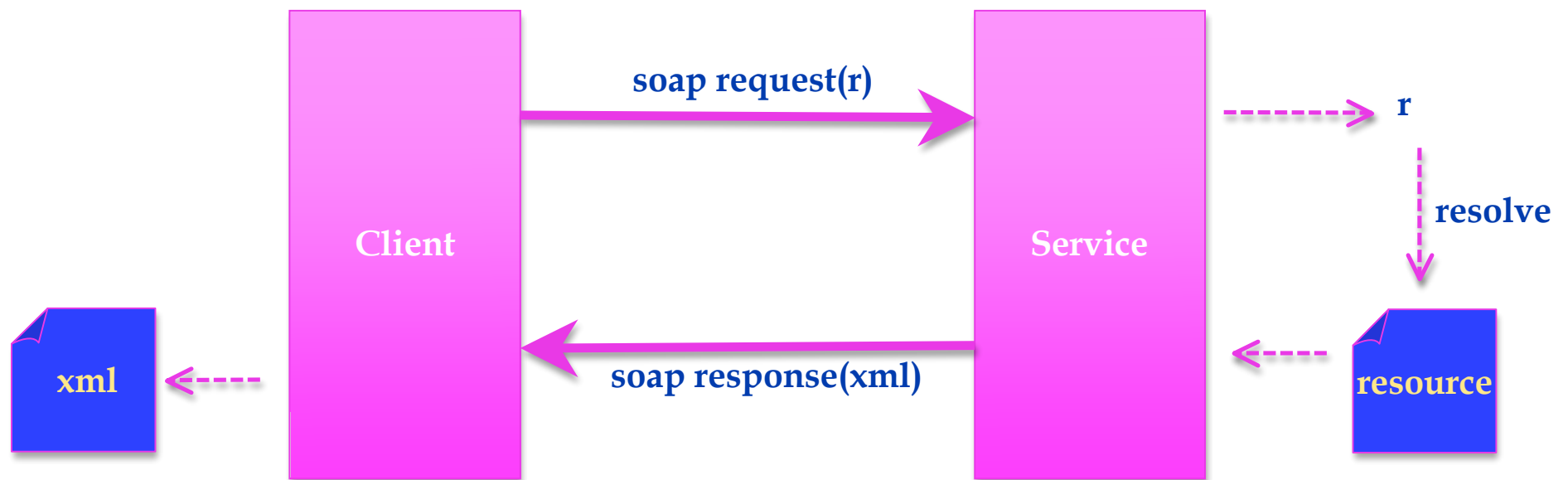
---

- Web Architecture offers many opportunity to insert or to obtain provenance information
- A matrix of possibilities has been identified
  - Pros/Cons to be discussed
  - Are there other options to consider
- What can realistically be achieved in the context of W3C?

---

# **BACKUP SLIDES: Web services**

# Web Service Request/Response



In response to a soap request( $r$ ), the client obtains xml, a representation of resource. The client may wonder what the provenance of 'xml' is?

# Embedding Provenance in SOAP Messages

---

- SOAP allows “message metadata” to be embedded in the header
  - E.g. ws-security signatures of message parts
- Same technique can be applied to provenance
- By value/by reference/mixed embedding of provenance in header is possible

# WS-Security Signing of SOAP Content

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      soap:mustUnderstand="1">
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="Signature-1">
        <ds:SignedInfo>
          <ds:Reference URI="#id-2">
            <ds:DigestValue>FZEKXmwDH+3vPvTQMyz1xO4+Agc=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>
          n2zNZiEvVrFZhG1/YjRXk6jSqzWGgysbZPwPyp5xQSV7+29ye8k6E+58idb9iPWmIWA//Crk2utB
          H6scFkw0ek3g9Gk89TJ+WFvNGUdOgPRNZaqBA6kQAvZhQOD2Ved7riEzvmaHRK/PRWE5dBfTZezS
          WaBlgsnwYIDqa8n4pcc=
        </ds:SignatureValue>
      </ds:Signature>
    </wsse:Security>
  </soap:Header>
  <soap:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    wsu:Id="id-2">
    <ns2:trade xmlns:ns2="http://tdata.comp6017.ecs.soton.ac.uk/">
      <security>ab</security>
      <quantity>100</quantity>
    </ns2:trade>
  </soap:Body>
</soap:Envelope>
```

# Provenance of SOAP Contents

---

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <prov:Provenance xmlns:prov="http://prov/prov.xsd">
      <prov:Reference URI="#id-2"/>
      <prov:Location>http://example.com/#id-2?prov</prov:Location>
    </prov:Provenance>
  </soap:Header>
  <soap:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-provcurity-utility-1.0.xsd"
    wsu:Id="id-2">
    <ns2:trade xmlns:ns2="http://tdata.comp6017.ecs.soton.ac.uk/">
      <security>ab</security>
      <quantity>100</quantity>
    </ns2:trade>
  </soap:Body>
</soap:Envelope>
```