

Web Services and the Grid – WSRF and WSRF::Lite

Mark Mc Keown
WSRF::Lite developer
Formerly UK Grid Support Center
Mark.McKeown@manchester.ac.uk



open middleware
infrastructure institute



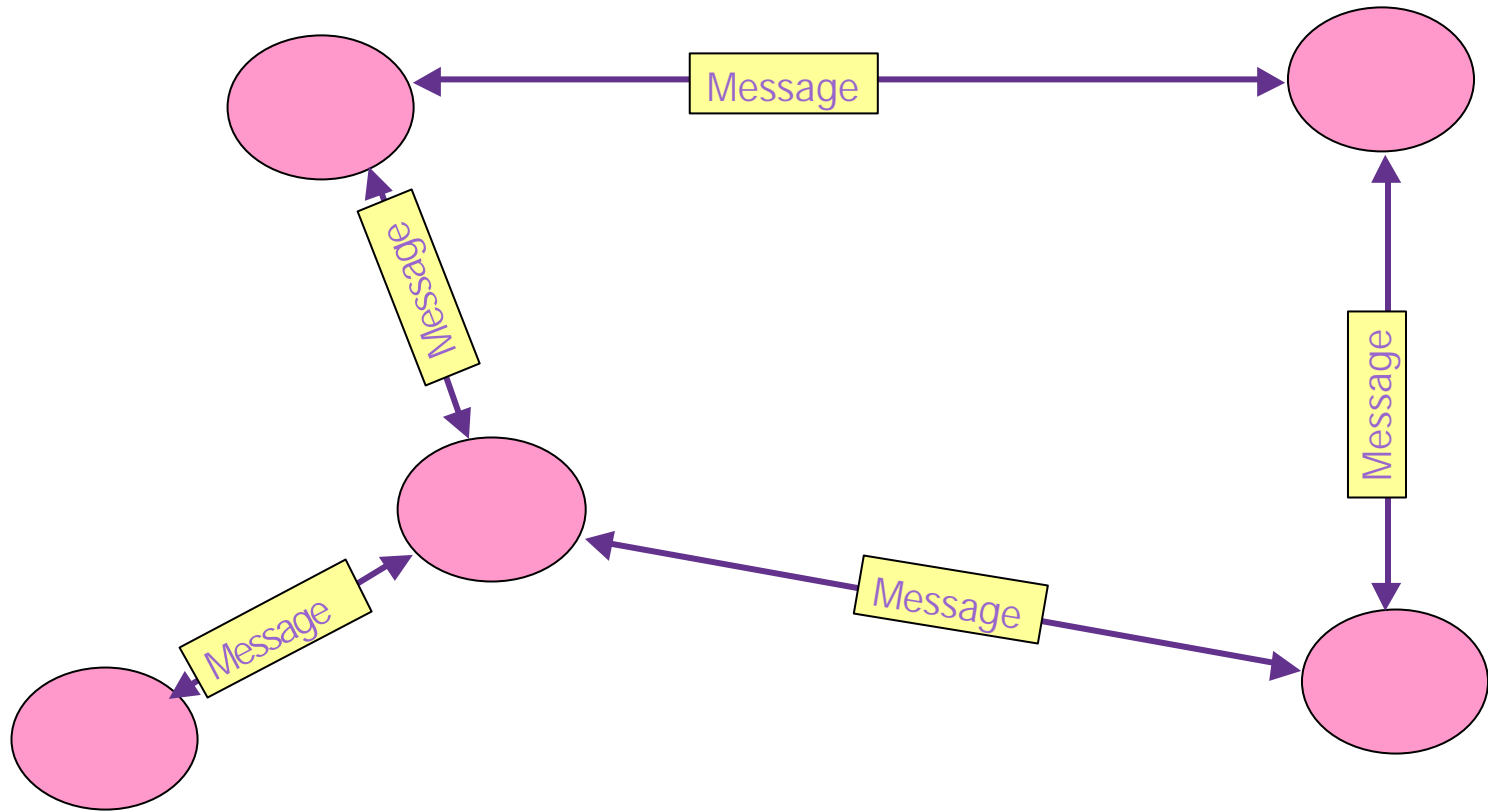
THE UNIVERSITY
of MANCHESTER

- Grid
- Web Services
- WSRF
- WSRF::Lite

What is a Grid...

- [...provides] "Flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resource"
 - "The Anatomy of the Grid"
- "...is distributed computing across multiple administrative domains"
 - Dave Snelling.
- "...internet scale distributed computing..."
- *Etc...*

Distributed Systems...



Designing a Distributed System...

- Design the messages.
- Design the message exchange patterns.
- What are the pink blobs?
 - Computer systems, processes on a system, institutes, objects, resources, State Machines?
- Deal with latency, concurrency, partial failure etc...

Web Services

- **“A Web service is a software system designed to support interoperable machine-to-machine interaction over a network.”**
- **“It has an interface described in a machine-processable format (specifically WSDL).”**
- **“Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.”**

W3C – Web Services Architecture

HTTP HEADERS

```
<Envelope>  
  <Header>  
    Some XML...  
  </Header>  
  <Body>  
    Some XML...  
  </Body>  
</Envelope>
```

OGSI – The Grid Service Instance

- “A Grid service instance is a (potentially transient) service that conforms to a set of conventions, expressed as WSDL interfaces, extensions and behaviours, for such purposes as lifetime management, discovery of characteristics, and notification.”
- “Grid services provide for the controlled management of distributed and often long-lived state that is commonly required in sophisticated distributed applications”.

“Anatomy of the Grid”

Why OGISI didn't succeed...

- Too much stuff in one specification.
- Does not work well with existing Web service and XML tooling.
- **Too object orientated.**

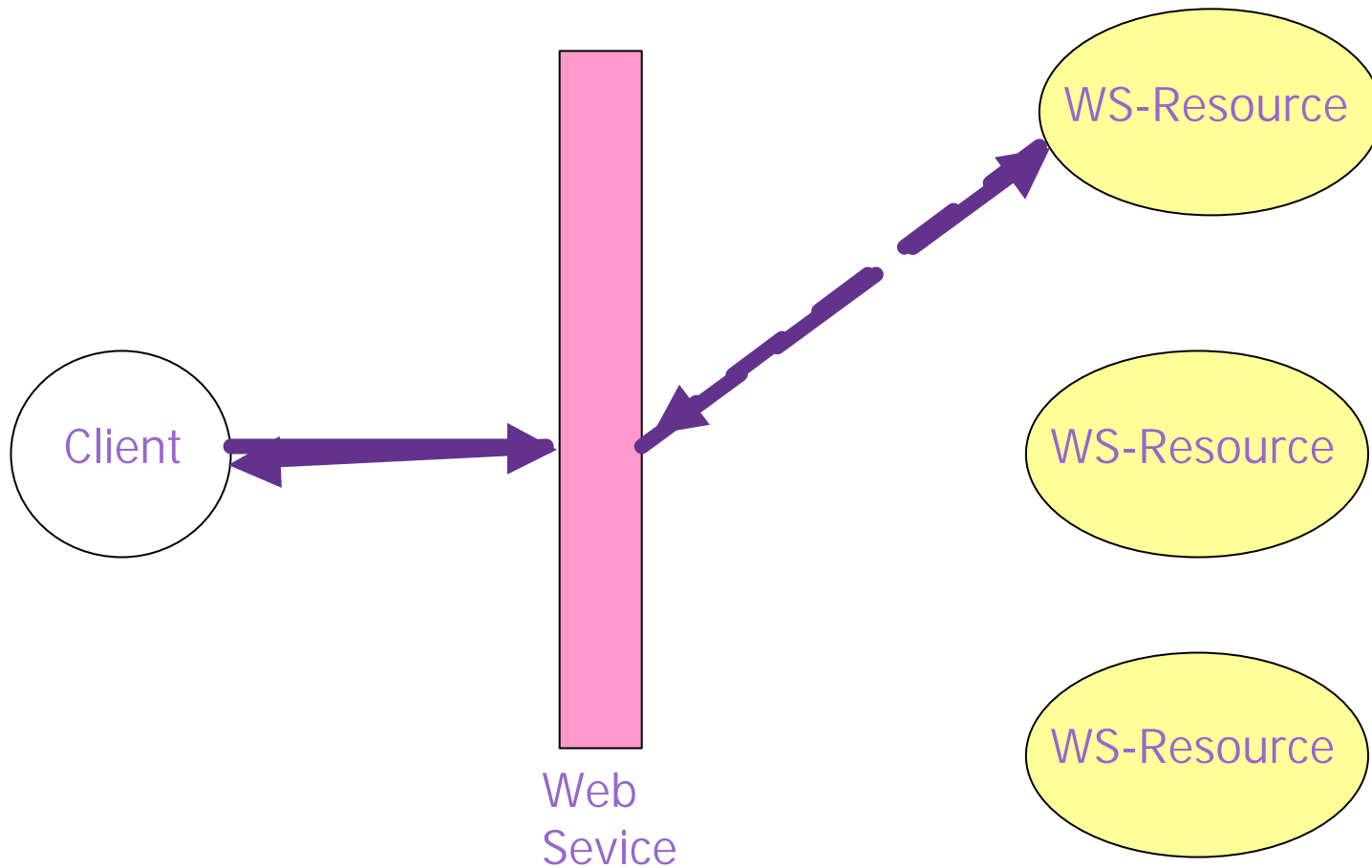
(see "A Note on Distributed Computing" by Waldo et al. for a critique of distributed object systems)

WSRF

Web Service Resource Framework

- WSRF effectively has replaced OGSI since January 2004.
- Addresses the issues with OGSI..
- Simply a re-factoring of OGSI – I wish ☹
- Instead of Grid Service Instances we have WS-Resources.

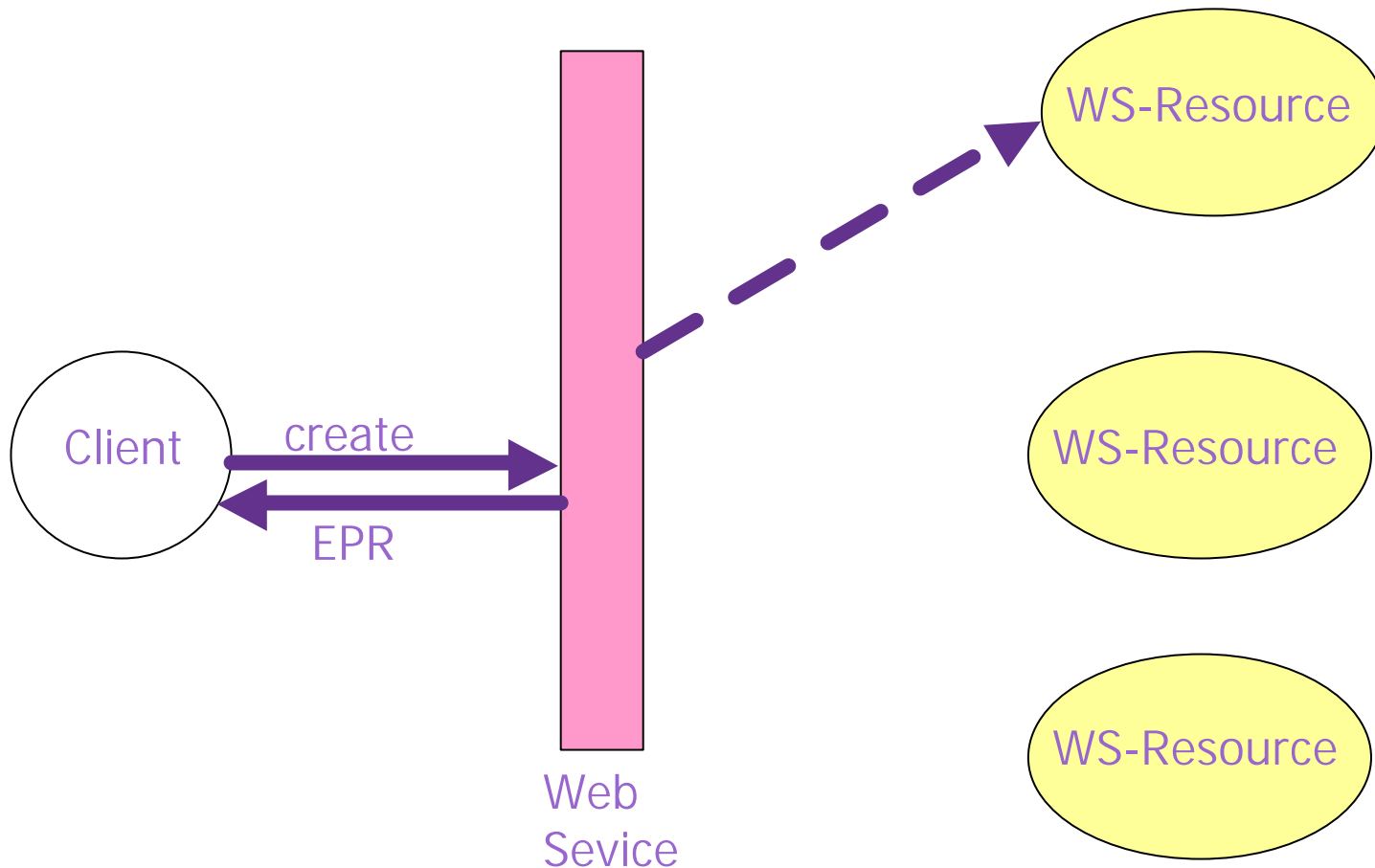
Implied Resource Pattern



POST http://vermont.man.ac.uk/Counter

```
<Envelope>  
  <Header>  
    <counterID>12342-dsfgsdfg</counterID>...  
  </Header>  
  <Body>  
    <add><value>10</value></add>  
  </Body>  
</Envelope>
```

Creating a new WS-Resource



```
<EndpointReference>  
  <Address>http://vermont.man.ac.uk/Counter</Address>  
  <ReferenceProperties>  
    <counterID>12342-dsfgsdfg</counterID>  
  </ReferenceProperties>  
</EndpointReference>
```

WSRF Family of Specifications

- WS-ResourceProperty
- WS-ResourceLifetime
- WS-BaseFaults
- WS-ServiceGroup
- WS-RenewableReferences (not yet released)

WS-ResourceProperty

- Provides a “projection” of the WS-Resource’s state.
- The ResourceProperties are described in an XML document – the WSDL for the service should have a pointer to this document.
- GetResourceProperty and GetMultipleResourceProperties operations allows client to query the state of the WS-Resource
- SetResourceProperty operation allows client to modify the state of the WS-Resource – supports Insert, Delete and Update.

ResourceProperty Document

```
<xs:schema .....>  
  <xs:element name="foo"  xsd:type="int" />*  
  <xs:element name="bar"   xsd:type="string" />?  
  <xs:element name="count" xsd:type="int" />?  
</xs:schema>
```

WS-ResourceLifetime

- Unlike OGSi, lifetime management is optional in WS-RF.
- Destroy and SetTerminationTime operations allows the client to control the lifetime of the WS-Resource.
- The lifetime of the WS-Resource is just another WS-ResourceProperty – can use GetResourceProperty to find termination time.
- However you **CANNOT** set the termination time through the SetResourceProperty operation – must use SetTerminationTime!!

- A standard way to report errors:

```
<BaseFault>
```

```
  <Timestamp>...</Timestamp>
```

```
  <OriginatorReference>...</OriginatorReference>?
```

```
  <ErrorCode>...</ErrorCode>?
```

```
  <Description>...</Description>*
```

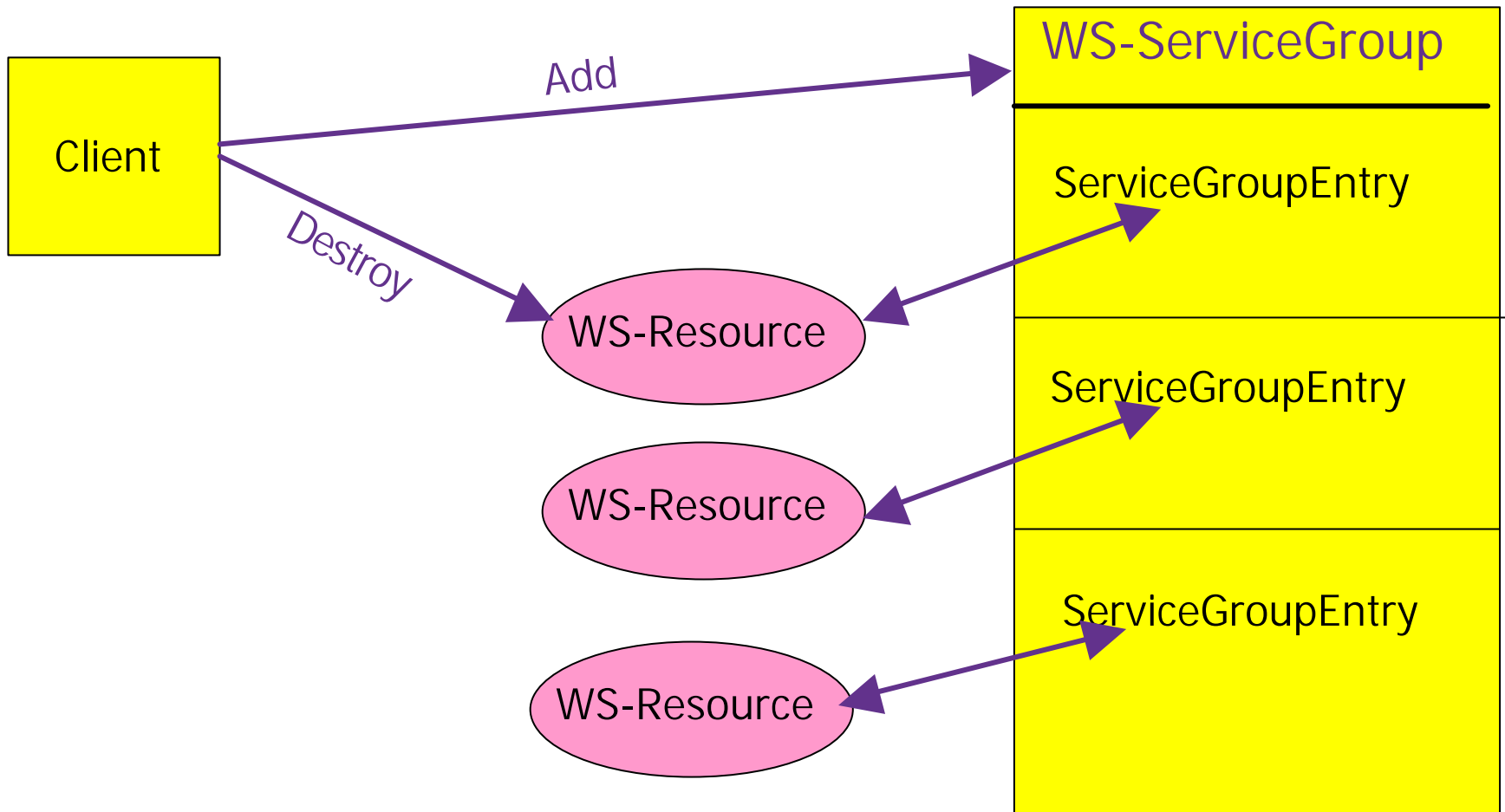
```
  <FaultCause>...</FaultCause>*
```

```
</BaseFault>
```

WS-ServiceGroup

- Mechanism to group a set of WS-Resources together – basic building block for registries.
- WS-Resources come and go, need to garbage collect stale entries in the ServiceGroup – but how?
- When we register a WS-Resource in a ServiceGroup a new WS-Resource is created by the ServiceGroup.
- The sole purpose of this new WS-Resource is to control the lifetime of the entry in the ServiceGroup – destroy this WS-Resource and the entry disappears.

WS-ServiceGroup



Other WS Options for building Grids

- **WS-I.** (<http://www.ws-i.org/>)
- **REST**, Representational State Transfer.
(<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>)
- **WS-Transfer.**
(<http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-transfer.pdf>)
- **MEST.**

- OGSI::Lite, the pre-cursor to WSRF::Lite, started out as an exercise to understand OGSI and the concepts behind Grid Services.
- The name is derived from SOAP::Lite the excellent Web Service module written by Pavel Kulchenko on which WSRF::Lite is built.
- “Don’t be mislead by the Lite suffix – this refers to the effort it takes to use the module, not its capabilities”

Pavel Kulchenko

Perl – Are you mad?

- “If Perl is the solution, you’re solving the wrong problem.”
Erik Naggum.
- “Perl as a language has less a design than a thousand special features flying in close formation.”
- “Perl: the first post-modern computer language.”
Larry Wall.

Apache CGI Script

```
#!/usr/bin/perl -w
```

```
use SOAP::Transport::HTTP;
```

```
SOAP::Transport::HTTP::CGI
```

```
- >dispatch_to( '/home/zzcgumk/modules' )
```

```
- >handle;
```

Simple Service

```
package HelloWorld;

sub Hello {
    my ($self, $name) = @_;
    return "Hello ".$name;
}

1;
```

Simple Client

```
#!/usr/bin/perl
use SOAP::Lite;

my $service = SOAP::Lite->service( "URLtoWSDL" );

my $ans = $service->Hello( "Mark" );

if ($ans->fault) { die $ans->faultstring }

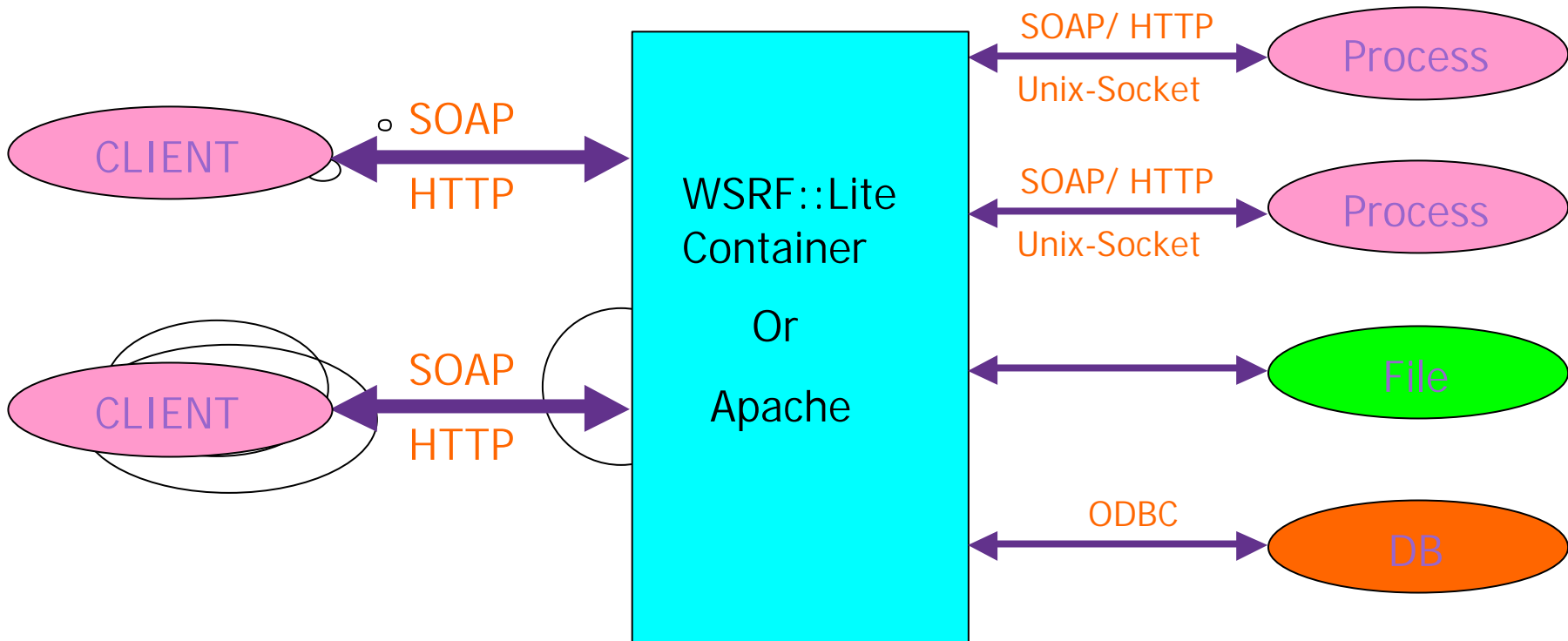
print $ans->result."\n";
```

WSRF::Lite Supports...

- WS-Addressing
- WS-ResourceProperties
- WS-ResourceLifetimes
- WS-BaseFaults
- WS-ServiceGroups

- WS-Resources can be hosted in the WSRF::Lite Container, Apache or using a simple standalone script.
- A WS-Resource simply inherits the core WSRF functionality from a base module.
- The WS-Resource state is stored and managed using either:
 - A Process.
 - A Database.
 - A File.

WSRF::Lite



- Currently the Container scripts support Transport Layer Security through HTTPS with x509 certificates.
- If the WS-Resources are hosted using GridSite then authentication using GSI proxy certificates is possible.
- Plan to implement WS-Security – unclear how this will work out.

Questions?

“...I still think the WS-* stack is bloated, opaque, and insanely complex. I think it's going to be hard to understand, hard to implement, hard to interoperate, and hard to secure.”

Tim Bray

<http://www.tbray.org/ongoing/When/200x/2004/09/18/WS-Oppo>