

---

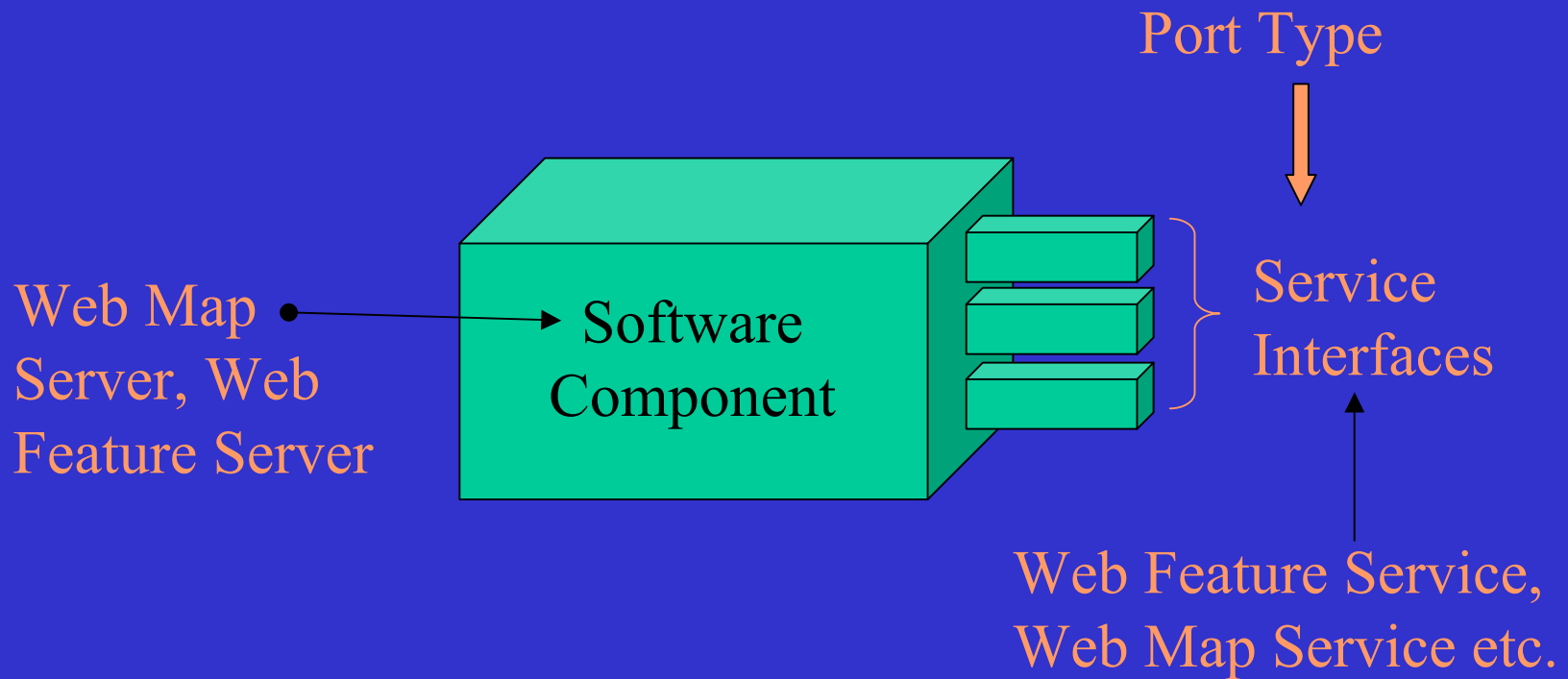
# WSDL Overview

June 4, 2001

# Some general issues

- Everything is a service !!
- Service = set of operations or interfaces!
- Component implements services.
- Need to build service type hierarchy.
- Service specification should separate abstract service description from binding to implementation protocols/formats.

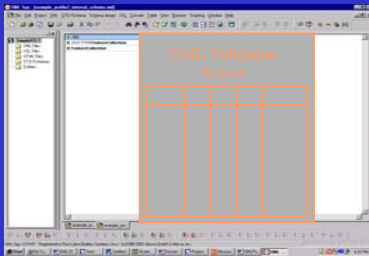
# Service & Server



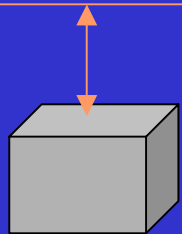
# OGC Services Architecture

Few distinguished things – services types !

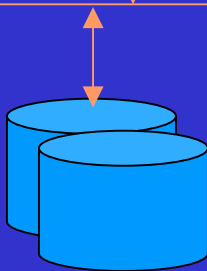
Browser  
Client



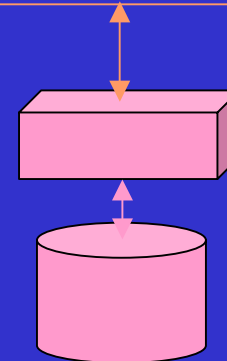
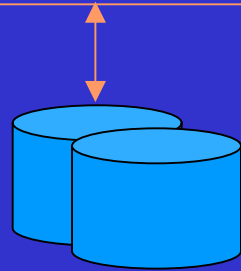
Service  
Directory



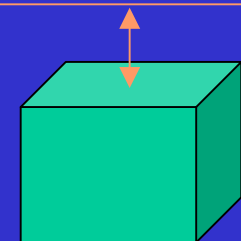
Registry  
Services



Data Services



Data Dependent Services

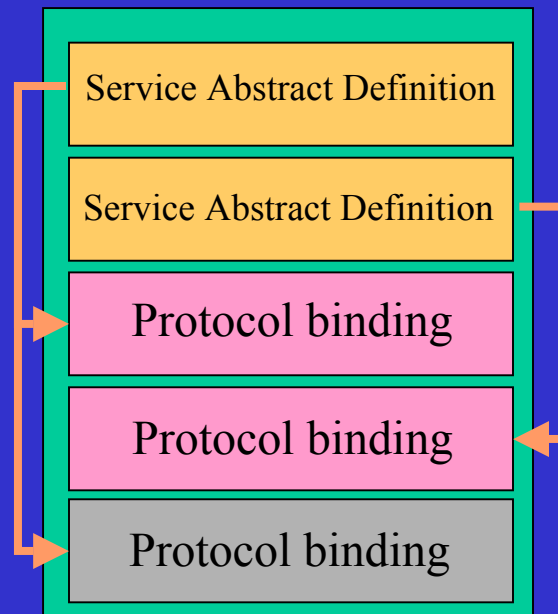


Transformation/  
Analysis  
Services

# Service Directory Concepts

A service directory contains or can provide WSDL document descriptions of a service including....

## WSDL Document



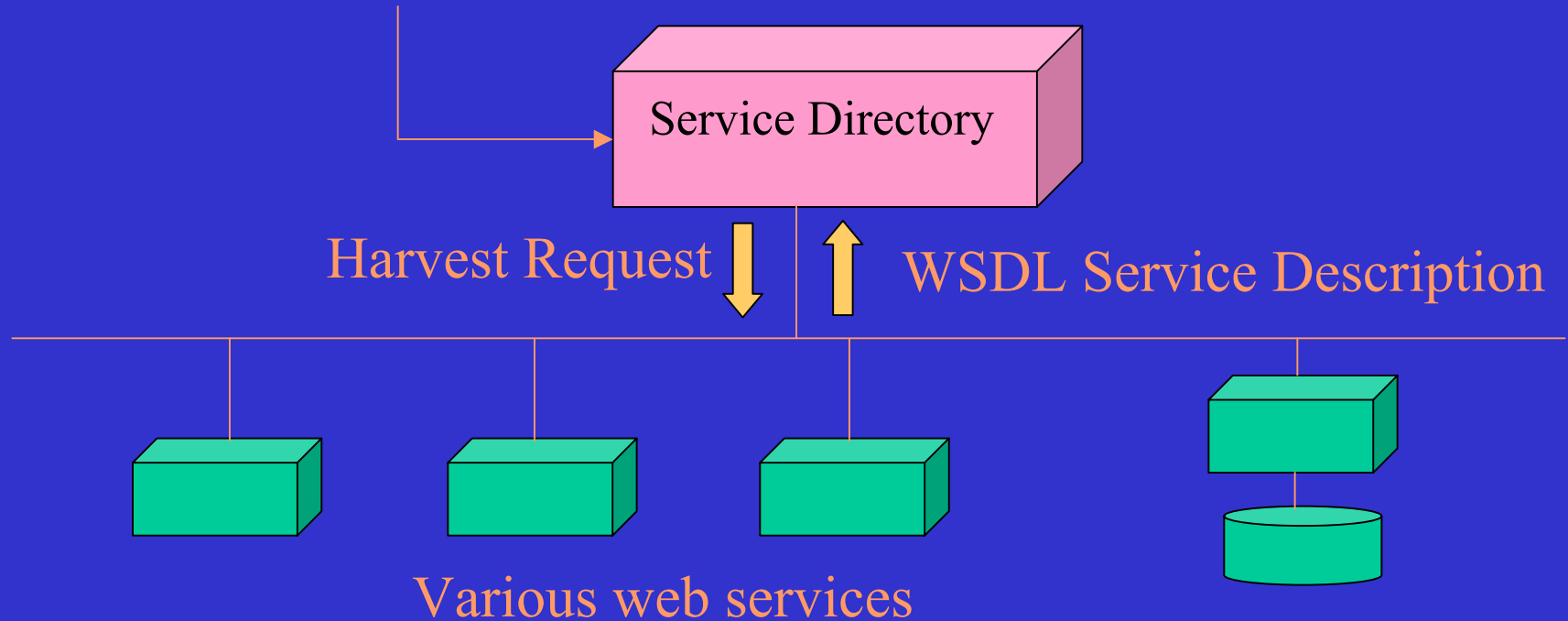
- Service interfaces
- Interface bindings
- Service metadata
- Message formats



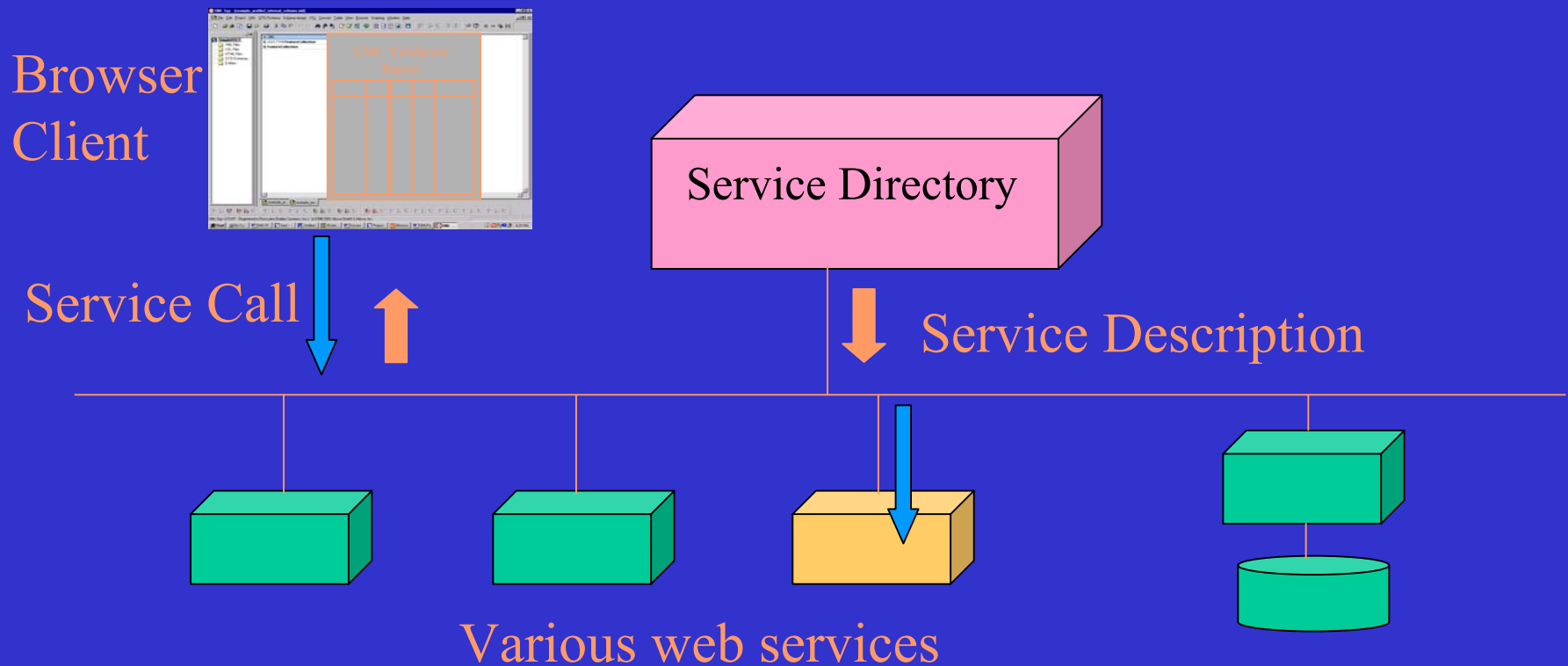
Service Invocation!

# Harvesting Service Descriptions

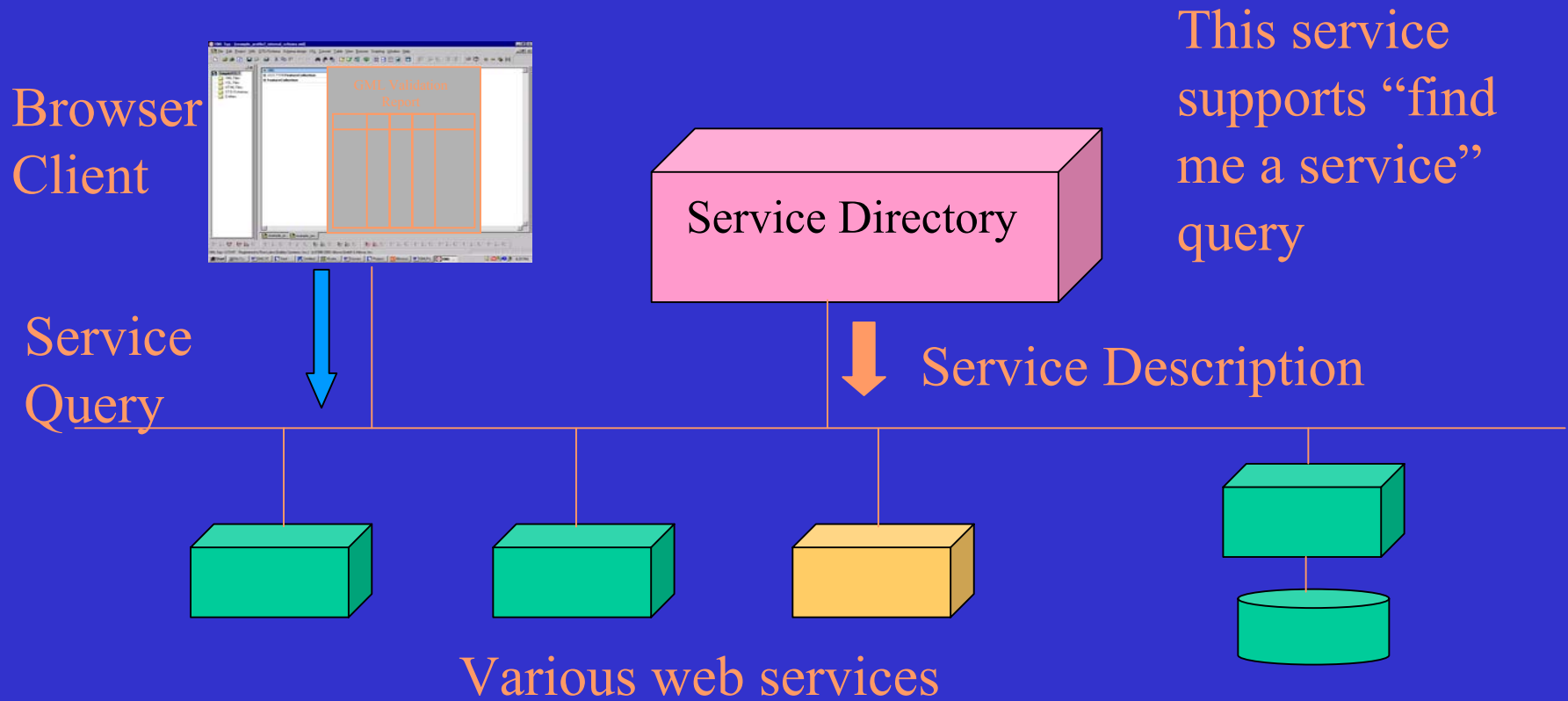
## The Service Directory Service



# Invoking Service Descriptions



# Service Discovery



# Where does WSDL fit?

- WSDL is a language to describe services.  
Specification and/or implementation vehicle !!
- This includes the Service Directory Service!
- This means WSDL does not itself provide anything to support service discovery itself!
- WSDL does provide the ability to describe the services for registration, harvesting, search.
- WSDL does provide the ability to attach metadata to a service to ENABLE service discovery.

# WSDL Summary

Port Type = Collection of abstract interfaces

```
<portType name = "WebFeatureServerType">  
  <documentation> ... </documentation>  
  <operation name = "GetCapabilities">  
    <input message = "exp:GetCapbilitiesInput" />  
    <output message = "exp:CapabilitiesOutput" />  
  </operation>  
  <operation name = "GetFeature">  
    <input message = "exp:GetFeatureInput" />  
    <output message = "exp:FeatureOutput" />  
  </operation>  
</portType>
```

Operation  
metadata goes  
here !!

Input/output  
messages

# WSDL Summary

Port Type = Collection of abstract interfaces

- One-way transmission e.g. SetStatus
- Request-message e.g. Response/Send
- Solicit-Response e.g. Send/Receive
- Notification e.g. Event like

# WSDL Summary

Message = abstract definition of data being transmitted

```
<message name = "GetFeatureInput" >  
  <documentation>... </documentation>  
  <part name = "body" element = "exp:FeatureRequest" />  
</message>
```

Message  
metadata goes  
here !!

Message can be expressed as a sequence of parts or a single part which is an XML element with specified schema.

# WSDL Summary

Don't use a special encoding !

```
<binding name="WebFeatureServerSoapBinding" type="exp:WebFeatureServerType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetFeature">
    <soap:operation soapAction="http://example.com/GetFeature"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

Binding associates the port types with a selected protocol.

# WSDL Summary

```
<schema targetNamespace="http://example.com/stockquote/schemas"
  xmlns="http://www.w3.org/2000/10/XMLSchema">
  <element name="FeatureRequest">
    <complexType>
      <element name="Region" type="gml:Polygon"/>
    </complexType>
  </element>
  <element name="FeatureList">
    <complexType>
      <sequence>
        <element name="featureType" type="string"/>
      </sequence>
    </complexType>
  </element>
</schema>
```

XML Schema describes the message structure.

```
<FeatureRequest>
  <Region>
    <gml:Polygon ...>
      .....
    </gml:Polygon>
  </Region>
  <FeatureList>
    <featureType>Road</featureType>
    <featureType>Park</featureType>
    <featureType>Lake</featureType>
  </FeatureList>
</FeatureRequest>
```

# WSDL Summary

```
<service name="BobsWebMapServer">  
  <port name="WebMappingPort" binding="exp:b1">  
    <soap:address location="http://bobco.com/webmap"/>  
  </port>  
</service>
```

The binding

The address

Service connects a binding and a web address !

# Next Steps

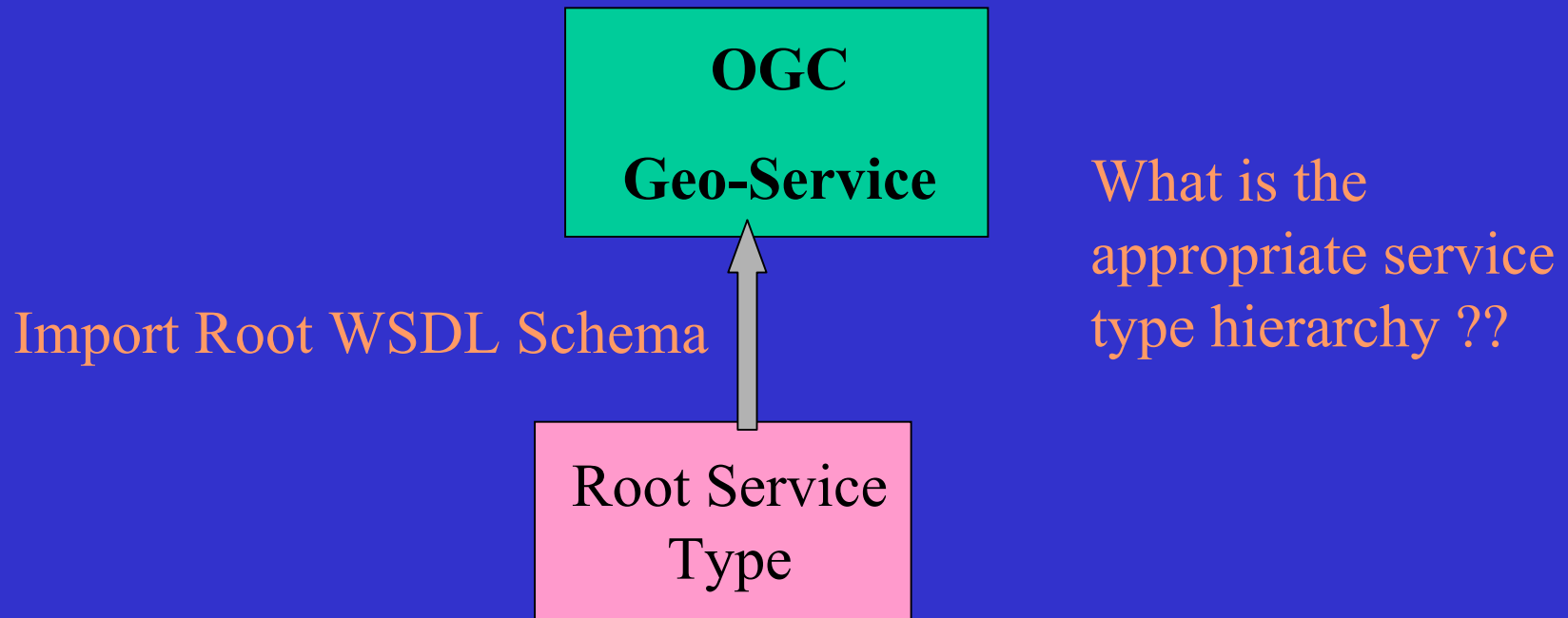
- Investigate integration of GML with WSDL
  - Service metadata
  - Messages containing GML data
- Look at implications for service architecture.
- Write up WSDL Descriptions for WFS, WMS, WCS etc.
- Investigate what is needed for service chaining.
- Develop mechanisms for service type hierarchy.
- Develop root operations (e.g. DescribeSelf )
- Develop Service Directory Description

# WSDL Summary

```
<types>
  <schema ... >
    <element name = "DescriptionStructure" type = "wsdl:definitionsType"/>
  </schema>
</types>
<message name="SelfDescriptionRequest">
  <part name = "request" element = "requestStructure" />
</message>
<message name="SelfDescription">
  <part name = "description" element = "DescriptionStructure" />
</message>
<portType name="DescribeSelfType">
  <operation name = "GetSelfDescription" >
    <input message = "SelfDescriptionRequest" />
    <output message = "SelfDescription" />
  </operation>
</portType>
```

A root  
PortType ??

# WSDL Summary



# Summary

- Start with WSDL as OGC Service Specification Language (successor to NASSL) as soon as possible!
- Work the current services (WFS, WCS .. ) into WSDL.

WSDL Specification:

<http://www.w3.org/TR/wsd1>