

Reference Resolution

Early Binding

With the workspace in Tuscany all the information is available to resolve all the references with targets in the domain. An input set of composites such as:

```
<composite xmlns="http://www.osea.org/xmlns/sca/1.0"
  targetNamespace="http://primary"
  xmlns:primary="http://primary"
  xmlns:dependent="http://dependent"
  name="CalculatorA">

  <include name="dependent:CalculatorB" />

  <component name="CalculatorServiceComponentA">
    <implementation.java class="calculator.CalculatorServiceImpl" />
    <service name="CalculatorService">
      <binding.sca />
    </service>
    <reference name="addService" target="AddServiceComponentA AddServiceComponentB">
      <binding.sca />
      <binding.ws />
    </reference>
    <reference name="subtractService" target="SubtractServiceComponentB" />
    <reference name="multiplyService" target="MultiplyServiceComponentA" />
    <reference name="divideService" target="DivideServiceComponentA" />
  </component>

  <component name="AddServiceComponentA">
    <implementation.java class="calculator.AddServiceImpl" />
    <service name="AddService">
      <binding.sca />
    </service>
  </component>

  <component name="MultiplyServiceComponentA">
    <implementation.java class="calculator.MultiplyServiceImpl" />
  </component>

  <component name="DivideServiceComponentA">
    <implementation.java class="calculator.DivideServiceImpl" />
  </component>

</composite>
```

```
<composite xmlns="http://www.osea.org/xmlns/sca/1.0"
  targetNamespace="http://dependent"
  xmlns:dependent="http://dependent"
  name="CalculatorB">

  <component name="AddServiceComponentB">
    <implementation.java class="calculator.AddServiceImpl" />
    <service name="AddService">
      <binding.ws uri="http://L3AW203:8085/AddServiceComponentB" />
    </service>
  </component>

  <component name="SubtractServiceComponentB">
    <implementation.java class="calculator.SubtractServiceImpl" />
  </component>

</composite>
```

Will result in the following resolved composite (which is actually invalid as it defines both reference targets and binding URIs on the same reference - They just happen to be consistent in this case)

```

<?xml version='1.0' encoding='UTF-8'?>
<composite targetNamespace="http://primary" name="CalculatorA"
  xmlns="http://www.osoa.org/xmlns/sca/1.0">
  <component name="CalculatorServiceComponentA"
    uri="CalculatorServiceComponentA">
    <implementation.java class="calculator.CalculatorServiceImpl" />
    <service name="CalculatorService">
      <interface.java interface="calculator.CalculatorService" />
      <binding.sca name="CalculatorService"
        uri="/CalculatorServiceComponentA" />
      <callback />
    </service>
    <reference name="addService"
      target="AddServiceComponentA AddServiceComponentB">
      <interface.java interface="calculator.AddService" />
      <binding.sca
        name="addService#AddServiceComponentA/AddService"
        uri="/AddServiceComponentA" />
      <binding.ws
        name="addService#AddServiceComponentB/AddService"
        uri="http://L3AW203:8085/AddServiceComponentB" />
      <callback />
    </reference>
    <reference name="subtractService"
      target="SubtractServiceComponentB">
      <interface.java interface="calculator.SubtractService" />
      <binding.sca
        name="subtractService#SubtractServiceComponentB/SubtractService"
        uri="/SubtractServiceComponentB" />
      <callback />
    </reference>
    <reference name="multiplyService"
      target="MultiplyServiceComponentA">
      <interface.java interface="calculator.MultiplyService" />
      <binding.sca
        name="multiplyService#MultiplyServiceComponentA/MultiplyServiceImpl"
        uri="/MultiplyServiceComponentA" />
      <callback />
    </reference>
    <reference name="divideService"
      target="DivideServiceComponentA">
      <interface.java interface="calculator.DivideService" />
      <binding.sca
        name="divideService#DivideServiceComponentA/DivideServiceImpl"
        uri="/DivideServiceComponentA" />
      <callback />
    </reference>
  </component>
  <component name="AddServiceComponentA" uri="AddServiceComponentA">
    <implementation.java class="calculator.AddServiceImpl" />
    <service name="AddService">
      <interface.java interface="calculator.AddService" />
      <binding.sca name="AddService" uri="/AddServiceComponentA" />
      <callback />
    </service>
  </component>
  <component name="MultiplyServiceComponentA"
    uri="MultiplyServiceComponentA">
    <implementation.java class="calculator.MultiplyServiceImpl" />
    <service name="MultiplyServiceImpl">
      <interface.java interface="calculator.MultiplyServiceImpl" />
      <binding.sca name="MultiplyServiceImpl"
        uri="/MultiplyServiceComponentA" />
      <callback />
    </service>
  </component>
  <component name="DivideServiceComponentA"
    uri="DivideServiceComponentA">
    <implementation.java class="calculator.DivideServiceImpl" />
    <service name="DivideServiceImpl">
      <interface.java interface="calculator.DivideServiceImpl" />

```

```
        <binding.sca name="DivideServiceImpl"
            uri="/DivideServiceComponentA" />
        <callback />
    </service>
</component>
<component name="AddServiceComponentB" uri="AddServiceComponentB">
    <implementation.java class="calculator.AddServiceImpl" />
    <service name="AddService">
        <interface.java interface="calculator.AddService" />
        <binding.ws name="AddService"
            uri="http://L3AW203:8085/AddServiceComponentB" />
        <callback />
    </service>
</component>
<component name="SubtractServiceComponentB"
    uri="SubtractServiceComponentB">
    <implementation.java class="calculator.SubtractServiceImpl" />
    <service name="SubtractService">
        <interface.java interface="calculator.SubtractService" />
        <binding.sca name="SubtractService"
            uri="/SubtractServiceComponentB" />
        <callback />
    </service>
</component>
</composite>
```

Late Binding

In the late binding case reference endpoints are not know until runt time. Hence a reference/target can remain unresolved until this point.