BAE SYSTEMS
INSPIRED WORK

Simple Example – Cross Platform View

Introduction

This document describes an ECOA® client-server example named "*Simple Example – Cross Platform View*".

The example is an extension to the "Simple Example" (ref. [2]) client-server example, with the addition of multiple instances of Client component deployed across multiple ECOA Platforms. A Cross Platform View has also been developed to demonstrate how it is possible to manage large-scale systems.

It is assumed that the reader is conversant with the ECOA Architecture Specification (ref. [1]) and the process of defining and declaring ECOA Assemblies, ASCs (components), Modules, and deployments in XML, and then using code generation to produce Module framework (stub) code units and ECOA Container and Platform code.

Aims

This ECOA "*Simple Example Cross Platform View*" example is intended to demonstrate the usage of a Cross Platform View to enable parts of an ECOA system to be developed in isolation. The Cross Platform View is an enabler for scalable systems to be developed using ECOA.

ECOA Features Exhibited

- Composition of an ECOA Cross Platform View
- Deployment of multiple composites across multiple Platforms
- Isolation of a Platforms view from within the full ECOA system

Design and Definition

Client-Server Functional Design

The "Simple Example Cross Platform View" client-server example will demonstrate how a Cross Platform View can be used to segregate a system and develop Platforms in isolation. The example will deploy 4 instances of client components, each of which will periodically perform a request, to a single server and will receive a data item in return (Figure 1).

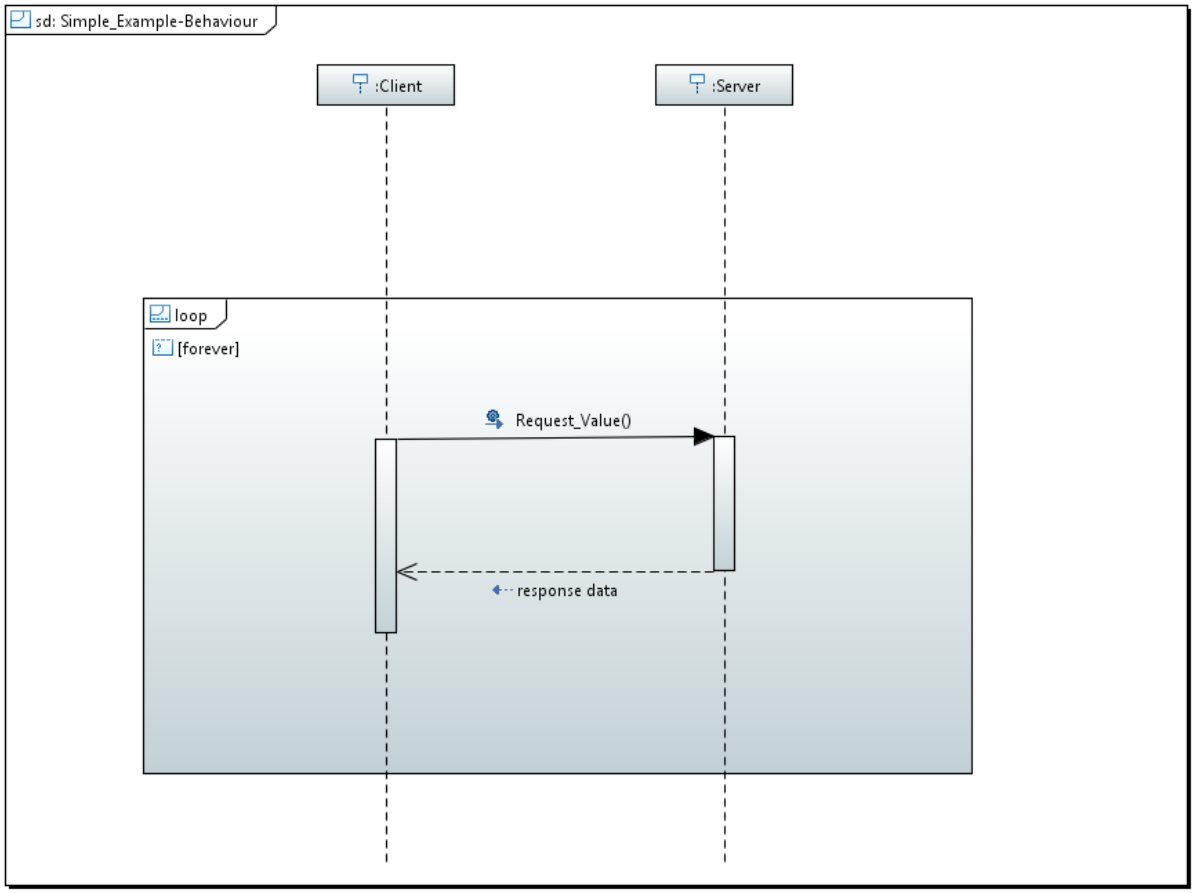

Figure 1 - ECOA "Simple Example – Cross Platform View" Client-Server Behaviour

The data content of the request will be the current absolute time and the response will be of a user defined type.

The Client will set a local variable to zero and output this to the log prior to performing the request. The result will be returned into this variable and logged.

The Client will be periodically activated at a rate of 0.5Hz (once every 2 seconds).

For detail on the specification and implementation of the Components used in this example, refer to the ECOA Simple Example (ref [3]) documentation on which this is based.

Logical System Architecture (Gods Eye View)

Figure 2 depicts the logical system architecture for this example. There is a single instance of a “Server” component and four instances of a “Client” component.

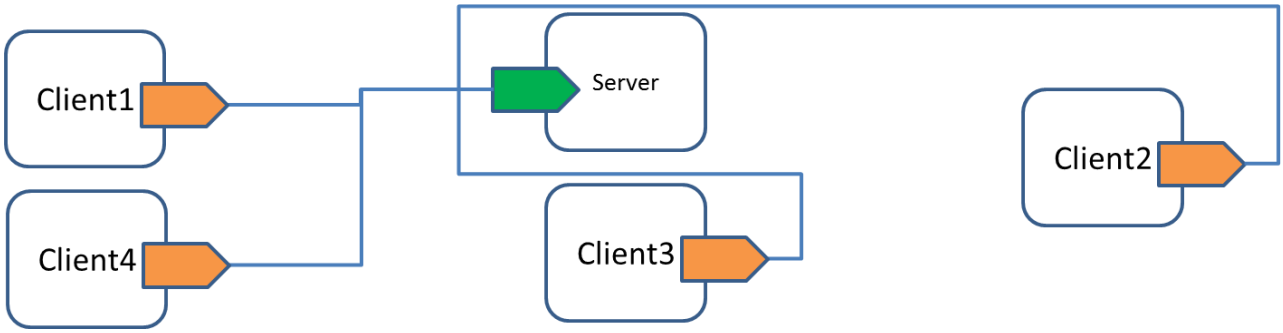

Figure 2 - Logical System Architecture

This system can be deployed in many different ways. Figure 3 shows an example deployment whereby the components are deployed across 3 ECOA Platforms.

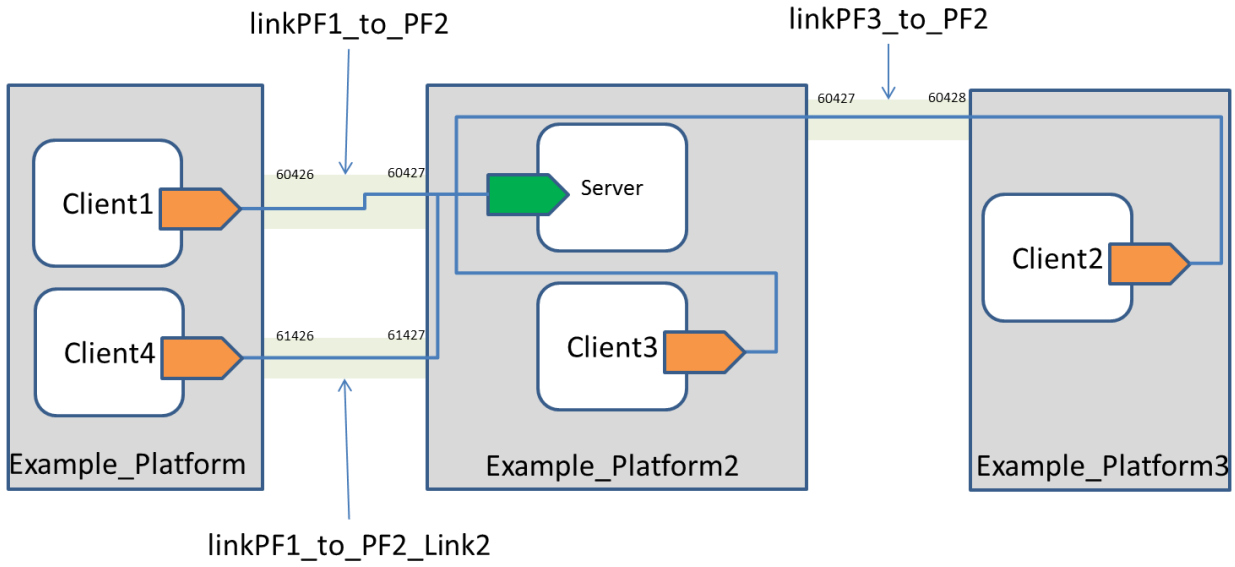

Figure 3 - Example Deployment

Logical System Architecture (Cross Platform View)

It is also possible to model the same ECOA system from a more abstract viewpoint. The Cross Platform View in ECOA allows you to "hide" the actual implementation of the Components deployed on a Platform. This is done by representing the Components deployed on a Platform as a composite.

Figure 4 shows the same logical system architecture but abstracted to a Composite level.

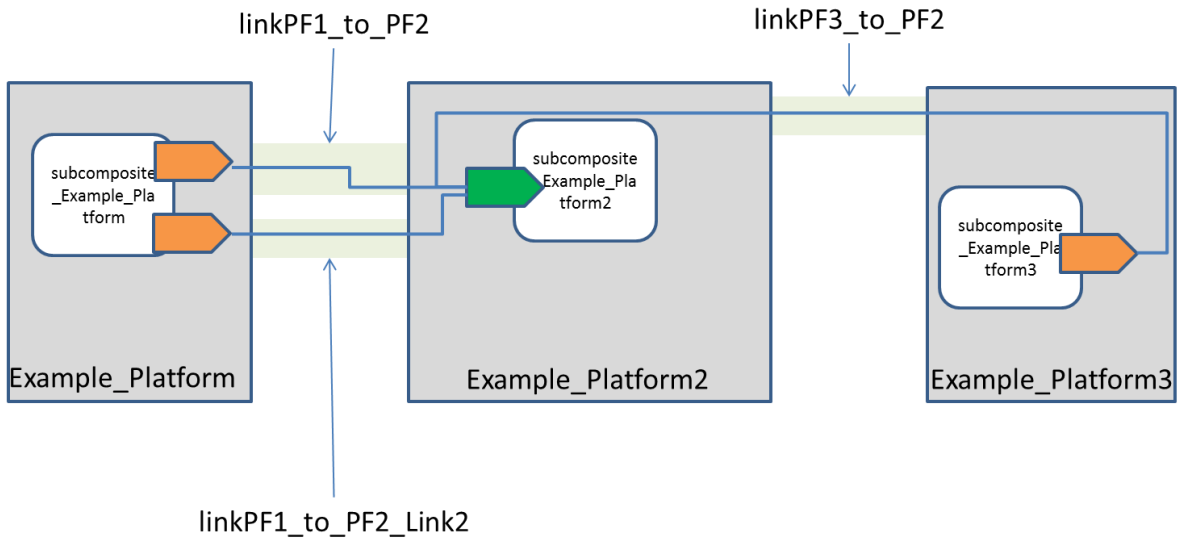

Figure 4 - Example Deployment - Cross Platform View

The following XML represents this Cross Platform View (defined in cross_platform_view.xml):

Cross Platform View

```
<view name="cross_platform_view" assembly="upper"
logicalSystem="cross_platform_system" xmlns="http://www.ecoa.technology/cross-
platforms-view-2.0">
  <composite name="sub_Example_Platform"
deployedOnComputingPlatform="Example_Platform"/>
  <composite name="sub_Example_Platform2"
deployedOnComputingPlatform="Example_Platform2"/>
  <composite name="sub_Example_Platform3"
deployedOnComputingPlatform="Example_Platform3"/>

  <wireMapping source="sub_Example_Platform/Request_Value_Service"
target="sub_Example_Platform2/Provide_Value_Service"
mappedOnLinkId="linkPF1_to_PF2" />
  <euidsBinding EUIDs="CPF" boundToLinkId="linkPF1_to_PF2" />

  <wireMapping source="sub_Example_Platform/Request_Value_Service2"
target="sub_Example_Platform2/Provide_Value_Service"
mappedOnLinkId="linkPF1_to_PF2_Link2" />
  <euidsBinding EUIDs="CPF" boundToLinkId="linkPF1_to_PF2_Link2" />

  <wireMapping source="sub_Example_Platform3/Request_Value_Service"
target="sub_Example_Platform2/Provide_Value_Service"
mappedOnLinkId="linkPF3_to_PF2" />
  <euidsBinding EUIDs="CPF" boundToLinkId="linkPF3_to_PF2" />

</view>
```

This XML references 3 sub composites which are to be deployed on each of the 3 platforms. The sub composites are abstractly defined in the assembly schema "upper"; represented by the following XML:

upper Assembly

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<csa:composite xmlns:csa="http://docs.oasis-open.org/ns/opencsa/sca/200912"
xmlns:ecoa-sca="http://www.ecoa.technology/sca-extension-2.0"
  name="upper" targetNamespace="AssemblySchema">

  <csa:component name="sub_Example_Platform">
    <csa:implementation.composite name="TBD"/>
    <csa:reference name="Request_Value_Service">
          <ecoa-sca:interface syntax="svc_Value"/>
    </csa:reference>
    <csa:reference name="Request_Value_Service2">
          <ecoa-sca:interface syntax="svc_Value"/>
    </csa:reference>
  </csa:component>

  <csa:component name="sub_Example_Platform2">
    <csa:implementation.composite name="TBD"/>
    <csa:service name="Provide_Value_Service">
          <ecoa-sca:interface syntax="svc_Value"/>
      </csa:service>
  </csa:component>

  <csa:component name="sub_Example_Platform3">
    <csa:implementation.composite name="TBD"/>
    <csa:reference name="Request_Value_Service">
          <ecoa-sca:interface syntax="svc_Value"/>
    </csa:reference>
  </csa:component>

  <csa:wire source="sub_Example_Platform/Request_Value_Service"
target="sub_Example_Platform2/Provide_Value_Service" />
  <csa:wire source="sub_Example_Platform/Request_Value_Service2"
target="sub_Example_Platform2/Provide_Value_Service" />
  <csa:wire source="sub_Example_Platform3/Request_Value_Service"
target="sub_Example_Platform2/Provide_Value_Service" />

</csa:composite>
```

The upper assembly details the composites which make up the system. At this abstract view no implementation of the composites has yet been defined (as this decomposition is at the discretion of each platform integrator). The implementation is therefore identified as to be defined (TBD).

Each of the sub composites can then be decomposed as required by each platform integrator. The following sections show examples of the sub composite implementation for each platform:

subcomposite_Example_Platform

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<csa:composite xmlns:csa="http://docs.oasis-open.org/ns/opencsa/sca/200912"
xmlns:ecoa-sca="http://www.ecoa.technology/sca-extension-2.0"
  name="subcomposite_Example_Platform" targetNamespace="AssemblySchema">

  <csa:reference name="Request_Value_Service" multiplicity="0..n"
promote="Client_Inst/Request_Value_Service"/>
  <csa:reference name="Request_Value_Service2" multiplicity="0..n"
promote="Client_Inst4/Request_Value_Service"/>

  <csa:component name="Client_Inst">
    <ecoa-sca:instance componentType="Client"/>
    <csa:reference name="Request_Value_Service"/>
  </csa:component>

  <csa:component name="Client_Inst4">
    <ecoa-sca:instance componentType="Client"/>
      <csa:reference name="Request_Value_Service"/>
  </csa:component>

</csa:composite>
```

Here the two required services, Client_Inst/Request_Value_Service and Client_Inst4/Request_Value_Service, are visible at Composite level and are promoted.

subcomposite_Example_Platform2

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<csa:composite xmlns:csa="http://docs.oasis-open.org/ns/opencsa/sca/200912"
xmlns:ecoa-sca="http://www.ecoa.technology/sca-extension-2.0"
  name="subcomposite_Example_Platform2" targetNamespace="AssemblySchema">

  <csa:service name="Provide_Value_Service"
promote="Server_Inst/Provide_Value_Service"/>

  <csa:component name="Client_Inst3">
    <ecoa-sca:instance componentType="Client"/>
    <csa:reference name="Request_Value_Service"/>
  </csa:component>

  <csa:component name="Server_Inst">
    <ecoa-sca:instance componentType="Server"/>
      <csa:service name="Provide_Value_Service"/>
  </csa:component>

  <csa:wire source="Client_Inst3/Request_Value_Service"
target="Server_Inst/Provide_Value_Service" />

</csa:composite>
```

Here the provided service Server_Inst/Provide_Value_Service is visible at Composite level and is promoted. Note that the Client_Inst3/Request_Value_Service is not promoted as it is not visible at the Composite level.

subcomposite_Example_Platform3

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<csa:composite xmlns:csa="http://docs.oasis-open.org/ns/opencsa/sca/200912"
xmlns:ecoa-sca="http://www.ecoa.technology/sca-extension-2.0"
  name="subcomposite_Example_Platform3" targetNamespace="AssemblySchema">

  <csa:reference name="Request_Value_Service" multiplicity="0..n"
promote="Client_Inst2/Request_Value_Service"/>

  <csa:component name="Client_Inst2">
    <ecoa-sca:instance componentType="Client"/>
    <csa:reference name="Request_Value_Service"/>
  </csa:component>

</csa:composite>
```

Here the required service Client_Inst2/Request_Value_Service is visible at Composite level and is promoted.

Development of Platforms in Isolation

The cross platform view means that Platforms may be developed in isolation. In this example, it is possible each platform to be developed in complete isolation from each other.

Development of Example_Platform

Figure 5 shows the view of the system from the perspective of Example_Platform. Note that the final composite which is deployed on an ECOA is flat (therefore no composites may exist). This means that the composite “subcomposite_Example_Platform2” must be represented as a Component (but without any implementation defined).

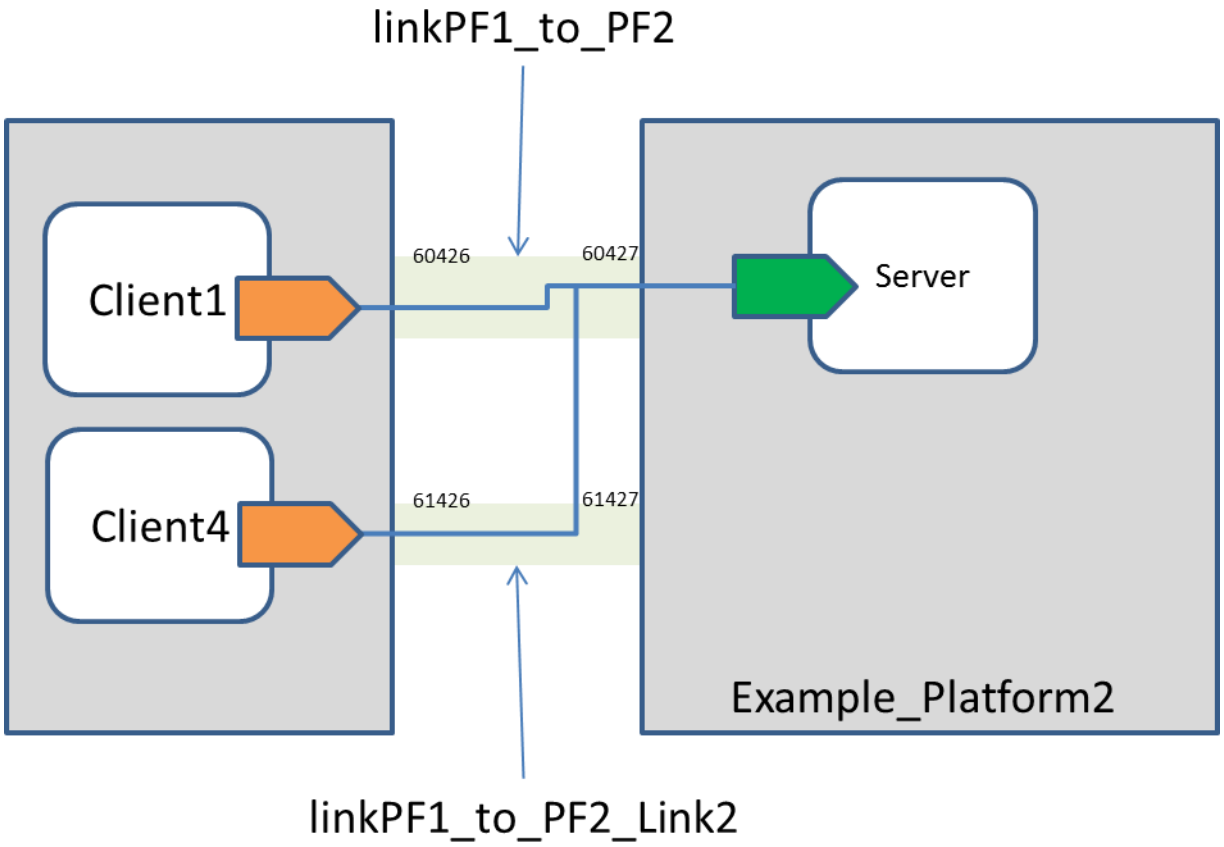

Figure 5 - View from Example_Platform

The XML which represents this composite is shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<csa:composite
	xmlns:csa="http://docs.oasis-open.org/ns/opencsa/sca/200912"
	xmlns:ecoa-sca="http://www.ecoa.technology/sca-extension-2.0"
	name="example"
	targetNamespace="http://www.ecoa.technology">

	<csa:component name="Client_Inst">

		<ecoa-sca:instance componentType="Client">
			<ecoa-sca:implementation name="Client_Im"/>
		</ecoa-sca:instance>

	</csa:component>

	<csa:component name="Client_Inst4">

		<ecoa-sca:instance componentType="Client">
```

```
                <ecoa-sca:implementation name="Client_Im"/>
            </ecoa-sca:instance>

        </csa:component>

        <csa:component name="Server_Inst" >

            <ecoa-sca:instance componentType="Server">
            </ecoa-sca:instance>

        </csa:component>

        <csa:wire source="Client_Inst/Request_Value_Service"
target="Server_Inst/Provide_Value_Service"/>
        <csa:wire source="Client_Inst4/Request_Value_Service"
target="Server_Inst/Provide_Value_Service"/>

</csa:composite>
```

This shows that Example_Platform has no knowledge of the existence of Example_Platform3 and its interactions with Example_Platform2.

Development of Example_Platform2

Figure 6 shows the view of the system from the perspective of Example_Platform2. Note that the final composite which is deployed on an ECOA is flat (therefore no composites may exist). This means that the composites "subcomposite_Example_Platform" and "subcomposite_Example_Platform3" must be represented as Components (but without any implementation defined).

The Component that represents the part of the system on Example_Platform is called Client_Dual and is defined by the following XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<componentType xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200912"
        xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ecoa-
sca="http://www.ecoa.technology/sca-extension-2.0">

        <reference name="Request_Value_Service1">
            <ecoa-sca:interface syntax="svc_Value" />
        </reference>

        <reference name="Request_Value_Service2">
            <ecoa-sca:interface syntax="svc_Value" />
        </reference>

</componentType>
```

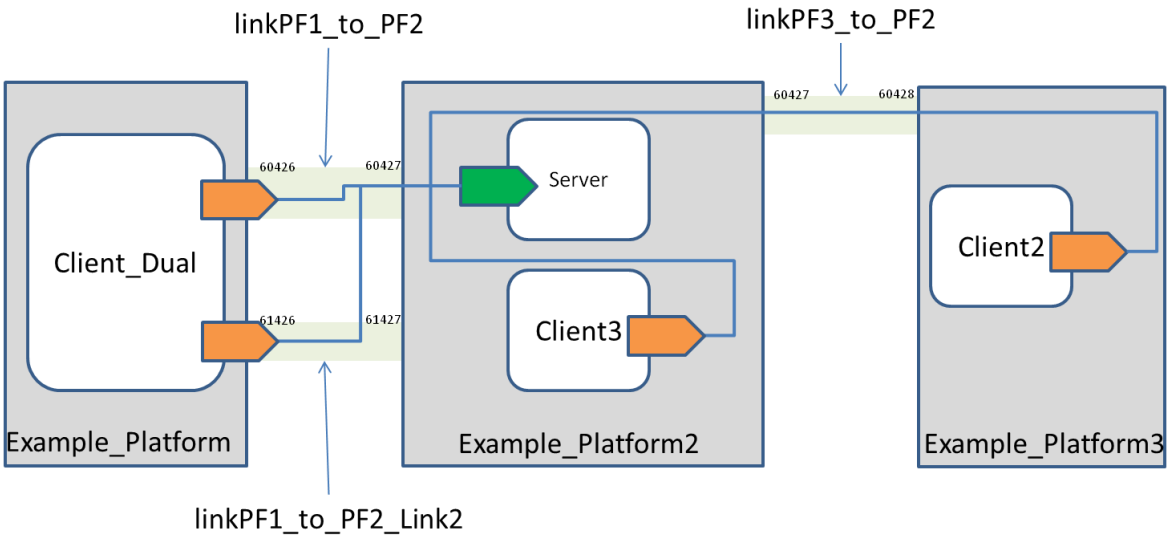

Figure 6 - View from Example_Platform2

The XML which represents this composite is shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<csa:composite
	xmlns:csa="http://docs.oasis-open.org/ns/opencsa/sca/200912"
	xmlns:ecoa-sca="http://www.ecoa.technology/sca-extension-2.0"
	name="example"
	targetNamespace="http://www.ecoa.technology">

	<csa:component name="Client_Dual">

		<ecoa-sca:instance componentType="Client_Dual">
		</ecoa-sca:instance>

	</csa:component>

	<csa:component name="Client_Inst2">

		<ecoa-sca:instance componentType="Client">
		</ecoa-sca:instance>

	</csa:component>

	<csa:component name="Client_Inst3">

		<ecoa-sca:instance componentType="Client">
			<ecoa-sca:implementation name="Client_Im"/>
		</ecoa-sca:instance>

	</csa:component>

	<csa:component name="Server_Inst" >
```

```
            <ecoa-sca:instance componentType="Server">
                  <ecoa-sca:implementation name="Server_Im"/>
            </ecoa-sca:instance>

      </csa:component>

      <csa:wire source="Client_Dual/Request_Value_Service1"
target="Server_Inst/Provide_Value_Service"/>
      <csa:wire source="Client_Dual/Request_Value_Service2"
target="Server_Inst/Provide_Value_Service"/>
      <csa:wire source="Client_Inst2/Request_Value_Service"
target="Server_Inst/Provide_Value_Service"/>
      <csa:wire source="Client_Inst3/Request_Value_Service"
target="Server_Inst/Provide_Value_Service"/>

</csa:composite>
```

This shows that Example_Platform2 has visibility of both Example_Platform and Example_Platform3; however it has no visibility of the client components within Example_Platform.

Development of Example_Platform3

Figure 7 shows the view of the system from the perspective of Example_Platform3. Note that the final composite which is deployed on an ECOA is flat (therefore no composites may exist). This means that the composite "subcomposite_Example_Platform2" must be represented as Component (but without any implementation defined).

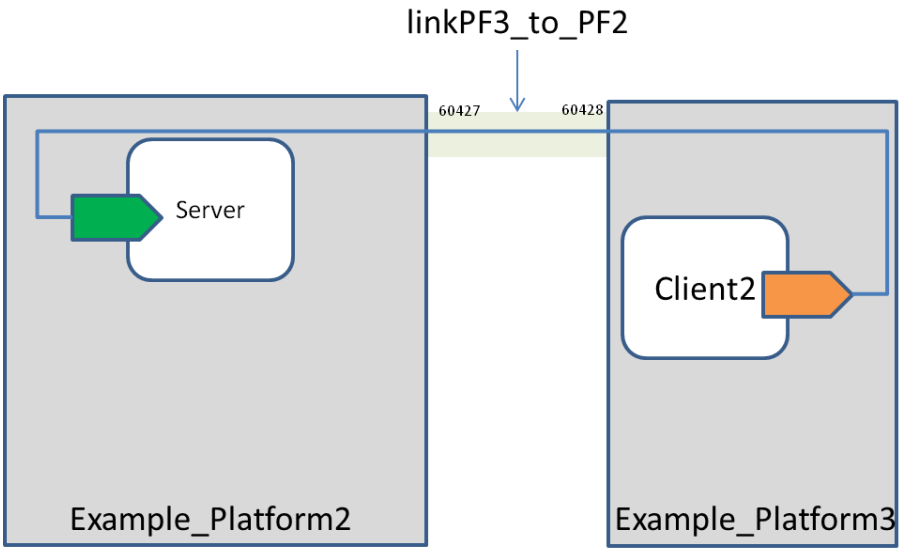

Figure 7 - View from Example_Platform3

The XML which represents this composite is shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<csa:composite
```

```
        xmlns:csa="http://docs.oasis-open.org/ns/opencsa/sca/200912"
        xmlns:ecoa-sca="http://www.ecoa.technology/sca-extension-2.0"
        name="example"
        targetNamespace="http://www.ecoa.technology">

        <csa:component name="Client_Inst2">

              <ecoa-sca:instance componentType="Client">
                    <ecoa-sca:implementation name="Client_Im"/>
              </ecoa-sca:instance>

        </csa:component>

        <csa:component name="Server_Inst" >

              <ecoa-sca:instance componentType="Server">
              </ecoa-sca:instance>

        </csa:component>

        <csa:wire source="Client_Inst2/Request_Value_Service"
target="Server_Inst/Provide_Value_Service"/>

</csa:composite>
```

This shows that Example_Platform3 has no knowledge of the existence of Example_Platform and its interactions with Example_Platform2.

Conclusion

This example demonstrates how platforms, and the components that are deployed on them, may be developed independently of other platforms. This provides the system integrator with a way to abstract parts of the system at the most appropriate level and allows scalable systems to be developed using ECOA.

References

1	European Component Oriented Architecture (ECOA) Collaboration Programme: Architecture Specification (Parts 1 to 11) "ECOA" is a registered trade mark.
2	Simple Example. http://www.ecoa.technology/tutorials.html