

GLUE2 XML Renderings

David Meredith

Warren Smith

XSD Style: Flat, Nested or Combined

- Flat chosen as preferred style at OGF35, but....a consensus? (are still requirements for nested and for backward compatibility).
- Style choice applies to how an entity's <Associations> element models relationships to other entities.
- **Solely Flat:**
 - <Associations> nests Element ID reference elements
 - All entity elements are siblings, no nesting.
- **Solely Nested:**
 - <Associations> uses 2 methods:
 - Nesting used to define (most) child associations +
 - Some associations also modelled using Element ID references. Is necessary when child has many parents and for *-to-* associations (and we MUST NOT duplicate elements).
 - Nested is thus really a 'Partial Hybrid' approach
- **Combined:**
 - XSD provides a choice between Flat, Nested or a style-mix depending on rendering requirements.
 - Capable of solely flat, solely nested, or hybrid docs using the same XSD.

Nested

```
<glue:Entities
```

```
  <AdminDomain BaseType="Domain">
```

```
    <ID>99876</ID>
```

```
    <WWW>http://ngs.ac.uk</WWW>
```

```
    ...
```

```
    <ComputingService BaseType="Service">
```

```
      <ID>2341</ID>
```

```
      <Type>org.some.compute</Type>
```

```
      <QualityLevel>production</QualityLevel>
```

```
      <TotalJobs>434</TotalJobs>
```

```
      ...
```

```
      <ComputingEndpoint>
```

```
      <ComputingEndpoint>
```

```
      <ComputingEndpoint>
```

```
    </ComputingService>
```

```
    <StorageService BaseType="Service">
```

```
      <ID>2342</ID>
```

```
      <Type>org.some.compute</Type>
```

```
      <QualityLevel>production</QualityLevel>
```

```
      <StorageEndpoint>
```

```
      <StorageShare>
```

```
      <StorageManager>
```

```
    </StorageService>
```

```
    <Service BaseType="Service">
```

```
      <ID>2342</ID>
```

```
      <Type>org.srb.SRB3</Type>
```

```
      <QualityLevel>production</QualityLevel>
```

```
      <Endpoint>
```

```
    </Service>
```

```
  </AdminDomain>
```

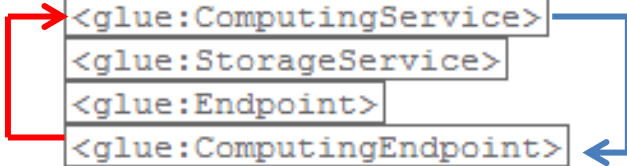
```
</glue:Entities>
```

(... detail elided)

1. <Entities> is single Doc Root Element ✓
 - Using <Entities> is same as flat style which means we do not have to support different Doc-Root elements.
2. Nesting enforces parent-child relationships. ✓
3. Easy doc traversal for most associations (easy XPath to select nested children rather than cross referencing IDs, nesting is the 'natural' approach in XML) ✓
4. Associations: ID references are still required for some associations (consider *-to-* and multiple parents). ✗
5. Cascading Children: For any given entity, you MUST always render child associations in full, inc. all children/grandchildren and so on... ✗
 - Thus can't always 'Project' just the required data using a Projection Query ('select * Services where...' without rendering all associations, see supporting slides). This can lead to XML bloat.
 - Consider rendering 1000s of <*Service> records and we don't require child associations such as Endpoints, Contacts, Shares etc to be rendered.
 - Note, most child associations are optional, *so its possible to not render those child elements*, but this is incomplete/misleading - Entities MUST be rendered in full.
 - Same applies if we just need to render immediate <*Domain> data (i.e. excluding child Services³ etc).

Flat

```
<glue:Entities
  xmlns:xsi='http://www.w3.org
  xmlns:glue='http://schemas.
  xsi:schemaLocation='http://
    <glue:Location>
    <glue:Contact>
    <glue:Contact>
    <glue:UserDomain>
    <glue:AdminDomain>
    <glue:AdminDomain>
    <glue:Service>
    <glue:ComputingService>
    <glue:StorageService>
    <glue:Endpoint>
    <glue:ComputingEndpoint>
    <glue:StorageEndpoint>
    <glue:ComputingShare>
    <glue:StorageShare>
    <glue:ComputingManager>
    <glue:StorageManager>
    <glue:DataStore>
    <glue:ExecutionEnvironment>
    <glue:Activity>
    <glue:ComputingActivity>
    <glue:AccessPolicy>
    <glue:MappingPolicy>
</glue:Entities>
```



(elements are collapsed)

1. <Entities> is single Doc Root element ✓
2. Relationships modelled using ID element references only (caters for *-to-* and Bi/Uni directional associations). ✓
3. Efficient. Do not have to render children/grandchildren/etc.. when rendering results from Projection Queries (only need to render immediate child ID <Associations>). ✓
4. Weaker association; relationship is not enforced by XSD.
 - Pro: a grid can be represented as multiple XML docs. ✓
 - Con: extra coding effort to validate that a reference points to the correct element. ✗
5. Traversing associations requires sub-queries to cross reference element IDs ✗

```
<glue:ComputingService BaseType="Service">
  <glue:ID>computingService1</glue:ID>
  <glue:Type></glue:Type>
  <glue:QualityLevel>production</glue:QualityLevel>
  <glue:Associations>
    <glue:ComputingEndpointID>computingEndpoint1</
  </glue:Associations>
</glue:ComputingService>

<glue:ComputingEndpoint BaseType="Endpoint">
  <glue:ID>computingEndpoint1</glue:ID>
  <glue:URL>uri://some.url.ac.uk/service</glue:URL>
  <glue:InterfaceName></glue:InterfaceName>
  <glue:QualityLevel>development</glue:QualityLevel>
  <glue:HealthState>ok</glue:HealthState>
  <glue:ServingState>production</glue:ServingState>
  <glue:Associations>
    <glue:ComputingServiceID>computingService1</gl
  </glue:Associations>
</glue:ComputingEndpoint>
```

```

<complexType name="ComputingService_t">
  <complexContent>
    <extension base="glue:ServiceBase_t">
      <sequence>

```

XSD

```

        <element name="TotalJobs" type="unsignedInt" minOccurs="0" maxOccurs="1" />
        <element name="RunningJobs" type="unsignedInt" minOccurs="0" maxOccurs="1" />
        <element name="WaitingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1" />
        <element name="StagingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1" />
        <element name="SuspendedJobs" type="unsignedInt" minOccurs="0" maxOccurs="1" />
        <element name="PreLRMSWaitingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1" />
        <element name="Associations" minOccurs="1" maxOccurs="1">

```

All
associations
are element
ID references

```

          <complexType>
            <sequence>
              <element name="ComputingEndpointID" type="glue:ID_t" minOccurs="1" maxOccurs="1" />
              <element name="ComputingShareID" type="glue:LocalID_t" minOccurs="1" maxOccurs="1" />
              <element name="ComputingManagerID" type="glue:ID_t" minOccurs="1" maxOccurs="1" />
              <element name="StorageServiceID" type="glue:ID_t" minOccurs="1" maxOccurs="1" />
              <element name="ContactID" type="glue:ID_t" minOccurs="1" maxOccurs="1" />
              <element name="LocationID" type="glue:ID_t" minOccurs="1" maxOccurs="1" />
              <element name="ServiceID" type="glue:ID_t" minOccurs="1" maxOccurs="1" />
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

</complexType>
</complexContent>
</extension>
</sequence>
</element>
</complexType>

```



ID References
to endpoints

```

<glue:ComputingService BaseType="Service">
  <ID>computingServiceReferencedEndpoints</ID>
  <Type></Type>
  <QualityLevel>production</QualityLevel>
  <Associations>
    <ComputingEndpointID>computingEndpoint1</ComputingEndpointID>
    <ComputingEndpointID>computingEndpoint3</ComputingEndpointID>
  </Associations>
</glue:ComputingService>

```

Flat XSD
Sample XML

XML


```

<!-- Entities is the DOCUMENT ROOT ELEMENT -->
<element name="Entities" type="glue:ExtensibleEntit:
  <annotation>
</element>
<complexType name="ExtensibleEntities_t">
  <sequence>
    <element ref="glue:Location" minOccurs="0" />
    <element ref="glue:Contact" minOccurs="0" />
    <!-- Abstract element references -->
    <element ref="glue:Domain" minOccurs="0" />
    <element ref="glue:AbstractService" minOccurs="0" />
    <element ref="glue:AbstractEndpoint" minOccurs="0" />
    <element ref="glue:Share" minOccurs="0" />
    <element ref="glue:Manager" minOccurs="0" />
    <element ref="glue:Resource" minOccurs="0" />
    <element ref="glue:AbstractActivity" minOccurs="0" />
    <element ref="glue:Policy" minOccurs="0" />
    <!-- Concrete element references -->
    <element ref="glue:Benchmark" minOccurs="0" />
    <element ref="glue:ApplicationEnvironment" minOccurs="0" />
    <element ref="glue:ToComputingService" minOccurs="0" />
    <element ref="glue:ToStorageService" minOccurs="0" />
    <element ref="glue:StorageAccessProtocol" minOccurs="0" />
    <element ref="glue:StorageServiceCapacity" minOccurs="0" />
    <element ref="glue:StorageShareCapacity" minOccurs="0" />
    <element ref="glue:ApplicationHandle" minOccurs="0" />
  </sequence>
</complexType>

<element name="Location" type="glue:Location_t" />
<element name="Contact" type="glue:Contact_t" />
<element name="Domain" type="glue:Domain_t" abstract="true" />
<element name="AdminDomain" type="glue:AdminDomain_t" />
<element name="UserDomain" type="glue:UserDomain_t" />

```

<Entities> is Common Doc Root Element for All Styles ✓

- A global element bag that lists GLUE entities as siblings.
- Applies to all styles (flat, nested, and combined).
- Enables Projection style queries regardless of style, i.e. select/render required entities only:
 - 'select * Services where...'
 - Note, projection for the nested style is less efficient as you still need to render all children/grandchildren/so on (Cascading children).

Q. Should we consider a Combined approach that enables a choice of nesting and/or element ID refs?

- For any single entity, use `<xsd:choice>` to allow a choice between Element ID refs OR nesting to model `<Associations>`.
- Variations on the `<xsd:choice>` pattern possible.

Q. Alternatively, define **two separate XSDs**: one flat and one nested? (also valid, but requires 2 XSDs)

Combined Approach Sample XSD

For an element, choose
between Nested or ID Refs
for EACH <Association>

A single element can mix
both refs and nesting in
<Associations/>.

Reference a <Contact>
that is declared globally as
a child of <Entities>, OR
directly nest a <Contact> ?

Rule: <ElementID> reference
elements MUST only reference
global elements that are
immediate children of <Entities>.

```
<complexType name="ComputingService_t">
  <complexContent>
    <extension base="glue:ServiceBase_t">
      <sequence>
        <element name="TotalJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="RunningJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="WaitingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="StagingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="SuspendedJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="PreLRMSWaitingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="Associations" minOccurs="1" maxOccurs="1">
          <complexType>
            <sequence>
              <choice>
                <element name="ComputingEndpointID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
                <element ref="glue:ComputingEndpoint" minOccurs="0" maxOccurs="1"/>
              </choice>
              <choice>
                <element name="ComputingShareID" type="glue:LocalID_t" minOccurs="0" maxOccurs="1"/>
                <element ref="glue:ComputingShare" minOccurs="0" maxOccurs="1"/>
              </choice>
              <choice>
                <element name="ComputingManagerID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
                <element ref="glue:ComputingManager" minOccurs="0" maxOccurs="1"/>
              </choice>
              <choice>
                <element name="ToStorageServiceID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
                <element ref="glue:ToStorageService" minOccurs="0" maxOccurs="1"/>
              </choice>
              <choice>
                <element name="ContactID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
                <element ref="glue:Contact" minOccurs="0" maxOccurs="1"/>
              </choice>
              <choice>
                <element name="LocationID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
                <element ref="glue:Location" minOccurs="0" maxOccurs="1"/>
              </choice>
              <element name="ServiceID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Combined Approach Sample XML

Both styles can be rendered in
same Doc.
Nested and Flat <AdminDomain>

```
<Entities
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns='http://schemas.opengis.org/glue/2009/03/spec_2.0_r1'
  xsi:schemaLocation='http://schemas.opengis.org/glue/2009/03/spec_2.0_r1'
  <AdminDomain BaseType="Domain">
    <ID>ad1</ID>
    <Name>GRID-LCG2</Name>
    <Associations>
      <ComputingService BaseType="Service">
        <ID>service1</ID>
        <Capability>executionmanagement.jobexecution</Capability>
        <Type>CREAM-CE</Type>
        <QualityLevel>production</QualityLevel>
        <StatusInfo>http://creamcel23.statusinfo.ac.uk/page</StatusInfo>
        <Complexity>some extra info</Complexity>
        <Associations>
          <ComputingEndpoint BaseType="Endpoint">
            <ID>computingEndpoint1</ID>
            <URL>uri://some.url.ac.uk/service</URL>
            <InterfaceName></InterfaceName>
            <QualityLevel>development</QualityLevel>
            <HealthState>ok</HealthState>
            <ServingState>production</ServingState>
            <Associations>
              <ComputingEndpoint ParticipationID="ad1" ServiceID="service1" LocationID="location1" ContactID="contact1" />
            </Associations>
          </ComputingEndpoint>
        </Associations>
      </ComputingService>
      <Contact>
        <ID>contact1</ID>
        <Detail>http://some.uri/embedding/contact/info</Detail>
        <Type>general</Type>
        <Associations>
          <Contact ParticipationID="ad1" />
        </Associations>
      </Contact>
      <Location>
        <ID>location1</ID>
        <Associations>
          <Location ParticipationID="ad1" />
        </Associations>
      </Location>
    </Associations>
  </AdminDomain>
  ....
```

```
....
  <AdminDomain BaseType="Domain">
    <ID>adminDomain2</ID>
    <Name>GRID-LCG2</Name>
    <Description>Site Description</Description>
    <WWW>http://www.some.service</WWW>
    <Distributed>false</Distributed>
    <Associations>
      <ServiceID>computingService1</ServiceID>
      <ServiceID>storageService1</ServiceID>
      <AdminDomainParticipationID>adminDomain2</AdminDomainParticipationID>
      <ContactID>contact1</ContactID>
      <ContactID>contact2</ContactID>
      <LocationID>location1</LocationID>
    </Associations>
  </AdminDomain>
</Entities>
```

Rule: <ElementID> reference
elements MUST only reference
global elements that are immediate
children of <Entities>.

Combined Approach

Sample XML

Both styles can be rendered in same_doc.

Nested and Flat <ComputingService>

```
<Entities
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns='http://schemas.opengis.net/2009/03/spec/2.0'
  xsi:schemaLocation='http://schemas.opengis.net/2009/03/spec/2.0'

  <ComputingService BaseType="Service">
    <ID>computingServiceIDrefAssociations</ID>
    <Type></Type>
    <QualityLevel>production</QualityLevel>
    <Associations>
      <ComputingEndpointID>computingEndpoint1</ComputingEndpointID>
      <ComputingEndpointID>computingEndpoint3</ComputingEndpointID>
      <ContactID>someContactID1</ContactID>
      <ContactID>someContactID2</ContactID>
      <LocationID>location1</LocationID>
      <ServiceID>otherService</ServiceID>
    </Associations>
  </ComputingService>

  <ComputingEndpoint BaseType="Endpoint">
    <ID>computingEndpoint1</ID>
    <URL>uri://some.url.ac.uk/service</URL>
    <InterfaceName></InterfaceName>
    <QualityLevel>development</QualityLevel>
    <HealthState>ok</HealthState>
    <ServingState>production</ServingState>
    <Associations>
    </ComputingEndpoint>
```

```
....
<ComputingService BaseType="Service">
  <ID>computingServiceNestedEndpoint</ID>
  <Type></Type>
  <QualityLevel>production</QualityLevel>
  <Associations>
    <ComputingEndpoint BaseType="Endpoint">
      <ID>computingEndpoint1</ID>
      <URL>uri://some.url.ac.uk/service</URL>
      <InterfaceName></InterfaceName>
      <QualityLevel>development</QualityLevel>
      <HealthState>ok</HealthState>
      <ServingState>production</ServingState>
      <Associations>
      </ComputingEndpoint>
    <ComputingEndpoint BaseType="Endpoint">
      <ID>computingEndpoint3</ID>
      <URL>uri://some.url.ac.uk/service</URL>
      <InterfaceName></InterfaceName>
      <QualityLevel>development</QualityLevel>
      <HealthState>ok</HealthState>
      <ServingState>production</ServingState>
      <Associations>
      </ComputingEndpoint>
    <Contact>
      <ID>someContactID1</ID>
      <Detail>http://some.uri/embedding/contact/info</Detail>
      <Type>general</Type>
      <Associations/>
    </Contact>
    <Contact>
      <ID>someContactID2</ID>
      <Detail>http://some.uri/embedding/contact/info</Detail>
      <Type>general</Type>
      <Associations/>
    </Contact>
    <Location>
      <ID>location1</ID>
      <Associations/>
    </Location>
    <ServiceID>otherService</ServiceID>
  </Associations>
</ComputingService>
```

```

<complexType name="ComputingService_t">
  <complexContent>
    <extension base="glue:ServiceBase_t">
      <sequence>
        <element name="TotalJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="RunningJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="WaitingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="StagingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="SuspendedJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="PreLRMSWaitingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1"/>
        <element name="Associations" minOccurs="1" maxOccurs="1">
          <complexType>
            <sequence>
              <choice>
                <!-- Style choice: ElementID sequence for flat rendering or nested -->
                <sequence>
                  <element name="ComputingEndpointID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
                  <element name="ComputingShareID" type="glue:LocalID_t" minOccurs="0" maxOccurs="1"/>
                  <element name="ComputingManagerID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
                  <element name="ToStorageServiceID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
                  <element name="ContactID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
                  <element name="LocationID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
                </sequence>
                <sequence>
                  <element ref="glue:ComputingEndpoint" minOccurs="0" maxOccurs="1"/>
                  <element ref="glue:ComputingShare" minOccurs="0" maxOccurs="1"/>
                  <element ref="glue:ComputingManager" minOccurs="0" maxOccurs="1"/>
                  <element ref="glue:ToStorageService" minOccurs="0" maxOccurs="1"/>
                  <element ref="glue:Contact" minOccurs="0" maxOccurs="unbounded"/>
                  <element ref="glue:Location" minOccurs="0" maxOccurs="1"/>
                </sequence>
              </choice>
              <!-- ServiceID is always ID association -->
              <element name="ServiceID" type="glue:ID_t" minOccurs="0" maxOccurs="1"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

ID refs

or

Nested

Note, a nested 'Service' association is not suitable here (thus only provide <ServiceID> option)

Variation on Combined Approach Sample XSD

For any single element, choose between ALL Nested or ALL Refs for all <Associations>.

Any single element can't mix both refs and nesting in single <Associations/> element.

Combined Approach

- **Pros:** Single XSD facilitates both styles

- **XML Producers** can choose style to suit rendering requirements:



1. Use ElementID refs for projection queries when you just need to render selected entities; “select * Services, Endpoints where...”
2. Use Nesting for other (eg XPath friendly) renderings and for backward compatibility (ARC)

- (Using both styles is not new; e.g. Spring framework caters for both Inner/Nested Beans + Bean references in ‘spring-beans.xml’ in very similar way)

- **Cons:** Is this too flexible/complex?

- **XML Consumers** need logic to deal with both approaches for full interoperability.



- Docs can appear ‘jumbled’ (although XSD validation doesn’t care about that).

Summary

- All variations have now been covered and discussed, now lets choose:

✗ 1. Nested Only

- Does not seem to be an option, too many requirements for the flat style.

? 2. Flat Only

- Drop any support for element nesting. Not backward compatible.

? 3. Combined

- XML producers can choose between Flat + Nesting renderings
- Too flexible for XML consumers? (full interop requires support for both styles)

? 4. Two related but separate XSDs (one flat, one nested, prefer single XSD)

- Interop still requires support for both XSDs, unless this can be profiled;
 - ‘Interoperability requires support for at least the flat style...’ ?
 - ‘Implementations MAY optionally choose to implement the nested style’ ?

- Regardless of style choice, these common validation rules still apply:

1. <Entities> is the only supported Doc Root element.
2. MUST not duplicate elements: Fail validation if a document contains two or more elements with the same <ID> value.
3. Element ID refs MUST only reference globally declared elements that are immediate children of <Entities>.

Additional (minor) style choices for Flat rendering

Flat XSD: Grouping Elements and/or BaseType Attribute

Flat XSD: Grouping Elements and/or BaseType Attribute

- At least one (or both) of these approaches is required to simplify XPath querying of a Flat XML document.
- For the fully flat XSD, then the BaseType attribute is required to simplify XPath queries.
- (note, BaseType attribute was carried over from original nested XSD).
- Grouping elements adds slight complexity as not all entities are siblings, but does allow easy collapsing/expanding of elements belonging to the same substitution group (but is this really a Pro?).
- Grouping elements are just a 'nice to have'

Flat with Grouping Elements

<glue:Entities

```
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:glue='http://schemas.ogf.org/oglue'
xsi:schemaLocation='http://schemas.ogf.org/oglue http://schemas.ogf.org/oglue'
```

```
<glue:Location>
<glue:Contact>
<glue:Contact>
```

<glue:Domains>

```
<glue:UserDomain>
<glue:AdminDomain>
<glue:AdminDomain>
```

</glue:Domains>

<glue:Services>

```
<glue:Service>
<glue:ComputingService>
<glue:StorageService>
```

</glue:Services>

<glue:Endpoints>

```
<glue:Endpoint>
<glue:ComputingEndpoint>
<glue:StorageEndpoint>
```

</glue:Endpoints>

<glue:Shares>

```
<glue:ComputingShare>
<glue:StorageShare>
```

</glue:Shares>

<glue:Managers>

```
<glue:ComputingManager>
<glue:StorageManager>
```

</glue:Managers>

<glue:Activities>

```
<glue:Activity>
<glue:ComputingActivity>
```

</glue:Activities>

<glue:Policies>

```
<glue:AccessPolicy>
<glue:MappingPolicy>
```

</glue:Policies>

</glue:Entities>

Grey boxes =
Collapsed
elements

No grouping elements needed for
concrete elements (they have no
substitutable alternatives)

<Services>
groups different
Service impls

<Endpoints>
groups different
Endpoint impls

<Managers> groups
different Manager
impls

Sample XPath to select all services,
endpoints with Grouping elements:

```
/Entities/Services/*
/Entities/Endpoints/*
```

Flat with no Grouping Elements

<glue:Entities

```
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:glue='http://schemas.ogf.org/oglue'
xsi:schemaLocation='http://schemas.ogf.org/oglue http://schemas.ogf.org/oglue'
```

```
<glue:Location>
```

```
<glue:Contact>
```

```
<glue:Contact>
```

```
<glue:UserDomain>
```

```
<glue:AdminDomain>
```

```
<glue:AdminDomain>
```

```
<glue:Service>
```

```
<glue:ComputingService>
```

```
<glue:StorageService>
```

```
<glue:Endpoint>
```

```
<glue:ComputingEndpoint>
```

```
<glue:StorageEndpoint>
```

```
<glue:ComputingShare>
```

```
<glue:StorageShare>
```

```
<glue:ComputingManager>
```

```
<glue:StorageManager>
```

```
<glue:DataStore>
```

```
<glue:ExecutionEnvironment>
```

```
<glue:Activity>
```

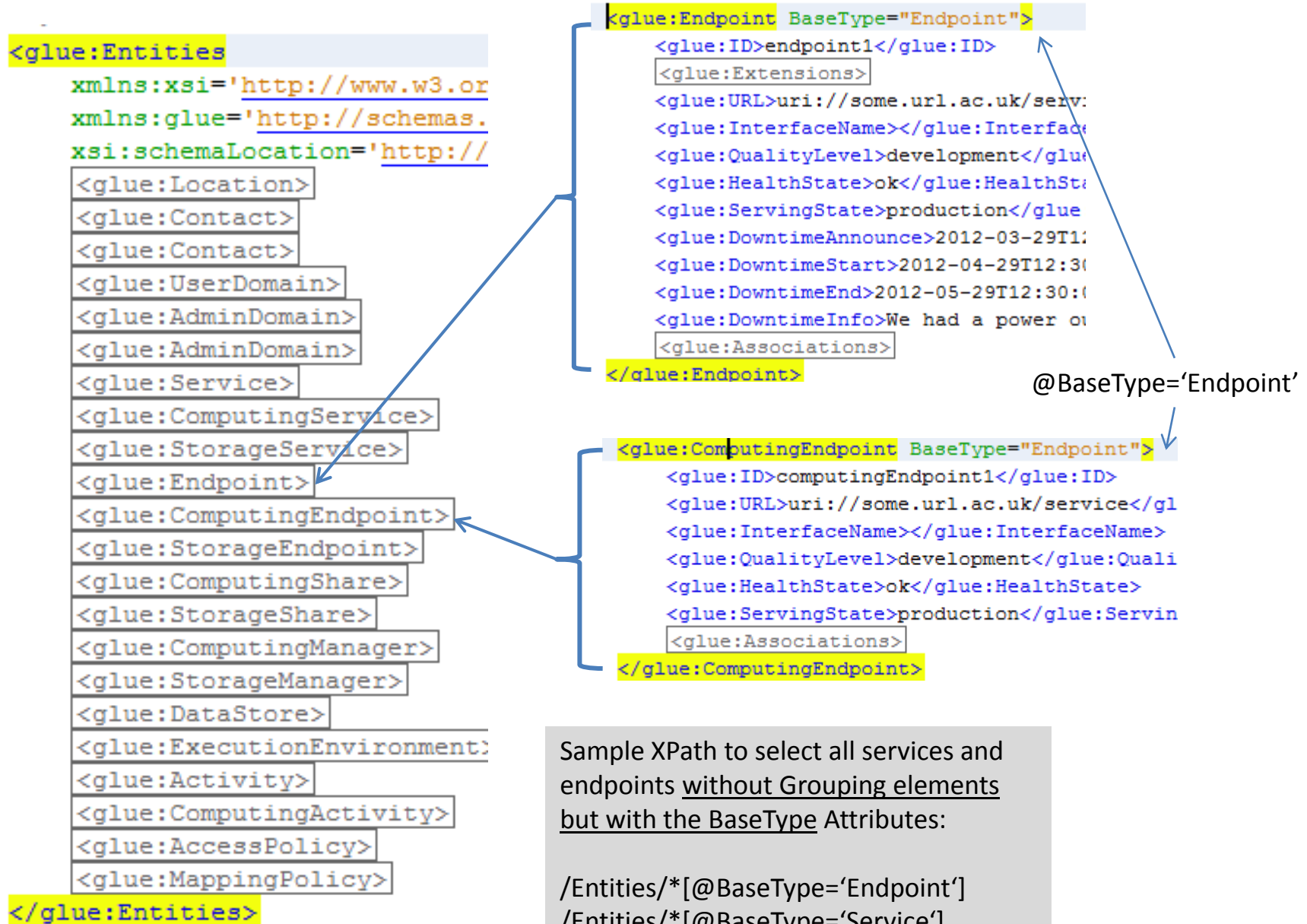
```
<glue:ComputingActivity>
```

```
<glue:AccessPolicy>
```

```
<glue:MappingPolicy>
```

</glue:Entities>

Flat with no Grouping Elements Requires the BaseType Attribute to simplify Xpath



Additional/backup slides

GLUE2: Projection Query Support

- Render results from Projection queries
 - Projection queries simply specify the entities you need to render when building a SELECT query (for SQL, you would normally specify fields/cols).
- E.g. GOCDB provides 18 projection style methods:
 - get_service_endpoint
 - get_ngi
 - get_site
 - get_contact
 - get_downtime
 - get_site_contacts ...

e.g. 'Select * service_endpoints where'
(don't need to return all parent and child associations)

GLUE2: Sample Projection Query Results from GOCDB

```
<results>
  <ROC ROC_NAME="NGI_UK">
    <ROCNAME>NGI_UK</ROCNAME>
    <MAIL_CONTACT>UKNGI-OPERATIONS@jiscma
    <CONTACT>
    <CONTACT>
    <CONTACT>
    <CONTACT>
    <CONTACT>
    <CONTACT>
    <CONTACT USER_ID="5955046" PRIMARY_KEY="44479G0">
      <FORENAME>John</FORENAME>
      <SURNAME>Kewley</SURNAME>
      <TITLE>Mr</TITLE>
      <DESCRIPTION/>
      <EMAIL>john.kewley@stfc.ac.uk</EMAIL>
      <TEL>+44 1925 603513</TEL>
      <WORKING_HOURS_START/>
      <WORKING_HOURS_END/>
      <CERTDN>/C=UK/O=eScience/OU=CLRC/
      <ROLE_NAME>NGI Operations Deputy
    </CONTACT>
    <CONTACT>
    <CONTACT>
    <CONTACT>
    <CONTACT>
    <CONTACT>
    <CONTACT>
    <CONTACT>
    <CONTACT>
```

1.

```
<results>
  <DOWNTIME>
  <DOWNTIME>
  <DOWNTIME>
  <DOWNTIME>
  <DOWNTIME ID="25994" PRIMARY_KEY="44479G0" C1>
    <PRIMARY_KEY>44479G0</PRIMARY_KEY>
    <HOSTNAME>goc.egi.eu</HOSTNAME>
    <SERVICE_TYPE>egi.GOCDB</SERVICE_TYPE>
    <ENDPOINT>goc.egi.eu/egi.GOCDB</ENDPOINT>
    <HOSTED_BY>GRIDOPS-GOCDB</HOSTED_BY>
    <GOCDB_PORTAL_URL>https://goc.egi.eu/portal</GOCDB_PORTAL_URL>
    <SEVERITY>WARNING</SEVERITY>
    <DESCRIPTION>Network disruptions to allow for maintenance</DESCRIPTION>
    <INSERT_DATE>1299751740</INSERT_DATE>
    <START_DATE>1300176000</START_DATE>
    <END_DATE>1300190400</END_DATE>
    <FORMATED_START_DATE>2011-03-15 08:00</FORMATED_START_DATE>
    <FORMATED_END_DATE>2011-03-15 12:00</FORMATED_END_DATE>
  </DOWNTIME>
  <DOWNTIME>
  <DOWNTIME>
  <DOWNTIME>
  <DOWNTIME>
  <DOWNTIME>
  <DOWNTIME>
  <DOWNTIME>
  <DOWNTIME>
  <DOWNTIME>
```

2.

~ Consider 1000's of records = can produce large XML documents.

- 1) https://goc.egi.eu/gocdbpi/private/?method=get_roc_contacts&roc=NGI_UK
- 2) https://goc.egi.eu/gocdbpi/public/?method=get_downtime&topentity=GOCDDB

Sample Flat Rendering (projecting services and endpoints)

```
<?xml version="1.0" encoding="UTF-8"?>
<glue:Entities
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:glue='http://schemas.ogf.org/glue'
  xsi:schemaLocation='http://schemas.ogf.org/glue'
  <glue:ComputingService>
  <glue:ComputingService>
  <glue:ComputingService>
  <glue:ComputingService>
  <glue:ComputingService>
  <glue:ComputingService>
  <glue:StorageService>
  <glue:Endpoint BaseType="Endpoint">
    <ID>endpoint1</ID>
    <Extensions>
    <URL>uri://some.url.ac.uk/service</URL>
    <InterfaceName></InterfaceName>
    <QualityLevel>development</QualityLevel>
    <HealthState>ok</HealthState>
    <ServingState>production</ServingState>
    <DowntimeAnnounce>2012-03-29T12:30:00Z</DowntimeAnnounce>
    <DowntimeStart>2012-04-29T12:30:00Z</DowntimeStart>
    <DowntimeEnd>2012-05-29T12:30:00Z</DowntimeEnd>
    <DowntimeInfo>We had a power outage!</DowntimeInfo>
    <Associations>
  </glue:Endpoint>
  <glue:ComputingEndpoint>
  <glue:StorageEndpoint>
  <glue:StorageEndpoint>
  <glue:StorageEndpoint>
</glue:Entities>
```

- Can select/render (project) just the required entities under the same Doc root.
- Efficient: No redundant data (consider 1000s of records).
- When selecting multiple entities (e.g. 'select * services, endpoints, Contacts for NGI_X') its harder to traverse the associations in the results (lots of ID lookups).

```

<complexType name="ComputingService_t">
  <complexContent>
    <extension base="glue:ServiceBase_t">
      <sequence>
        <element name="TotalJobs" type="unsignedInt" minOccurs="0" maxOccurs="1",
        <element name="RunningJobs" type="unsignedInt" minOccurs="0" maxOccurs="1",
        <element name="WaitingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1",
        <element name="StagingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1",
        <element name="SuspendedJobs" type="unsignedInt" minOccurs="0" maxOccurs="1",
        <element name="PreLRMSWaitingJobs" type="unsignedInt" minOccurs="0" maxOccurs="1",
        <element name="Associations" minOccurs="1" maxOccurs="1">

```

Another variation on the Combined Approach


Unrestricted Mixing of Nested + ID Refs in <Associations>

Associations can be directly nested and/or referenced

```

    <sequence>
      <element name="ComputingEndpointID" type="glue:ID_t" minOccurs="0" maxOccurs="1",
      <element ref="glue:ComputingEndpoint" minOccurs="0" maxOccurs="1">
        <element name="ComputingShareID" type="glue:LocalID_t" minOccurs="0" maxOccurs="1",
        <element ref="glue:ComputingShare" minOccurs="0" maxOccurs="1">
          <element name="ComputingManagerID" type="glue:ID_t" minOccurs="0" maxOccurs="1",
          <element ref="glue:ComputingManager" minOccurs="0" maxOccurs="1">
            <element name="StorageServiceID" type="glue:ID_t" minOccurs="0" maxOccurs="1",
            <element ref="glue:StorageService" minOccurs="0" maxOccurs="1">
              <element name="ContactID" type="glue:ID_t" minOccurs="0" maxOccurs="1",
              <element ref="glue:Contact" minOccurs="0" maxOccurs="unbounded">
                <element name="LocationID" type="glue:ID_t" minOccurs="0" maxOccurs="1",
                <element ref="glue:Location" minOccurs="0" maxOccurs="unbounded">
                  <element name="ServiceID" type="glue:ID_t" minOccurs="0" maxOccurs="1",

```

This variation is  probably too flexible.

```

    </sequence>
  </complexType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>

```



2 Referenced
+
1 Nested
endpoints

```

<glue:ComputingService BaseType="Service">
  <ID>computingServiceReferencedEndpoints</ID>
  <Type></Type>
  <QualityLevel>production</QualityLevel>
  <Associations>
    <ComputingEndpointID>computingEndpoint1</ComputingEndpointID>
    <ComputingEndpointID>computingEndpoint3</ComputingEndpointID>
    <glue:ComputingEndpoint BaseType="Endpoint">
      <ID>computingEndpoint1</ID>
      <URL>uri://some.url.ac.uk/service</URL>
      <InterfaceName></InterfaceName>
      <QualityLevel>development</QualityLevel>
      <HealthState>ok</HealthState>
      <ServingState>production</ServingState>
      <Associations>
        <ComputingServiceID>service2</ComputingServiceID>
      </Associations>
    </glue:ComputingEndpoint>
  </Associations>
</glue:ComputingService>

```