To enable track and trace for aggregation, disaggregation, and relabling events a new type of event `<tAggregationEvent>` is proposed to the draft proposal draft-thompson-esds-schema-03. The proposed schema changes are outlined below:

```
<xs:complexType name="tAbstractEvent" abstract="true">
  <xs:sequence>
    <xs:element
      name="extension"
      type="esds:tExtension"
      minOccurs="0"/>
    <xs:element
      name="lifeCycleStepID"
      type="esds:tLifeCycleStepID"
      minOccurs="0"/>
    <xs:element
      name="sourceTS"
      type="xs:dateTime"/>
    <xs:element
      name="serviceIDList"
      type="esds:tServiceIDList"
      minOccurs="0"/>
    <xs:element
      name="action"
      type="esds:tAction"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="tEventTypeChoice">
  <xs:choice>
    <xs:element
      name="objectEvent"
      type="esds:tObjectEvent"/>
    <xs:element
      name="aggregationEvent"
      type="esds:tAggregationEvent"/>
  </xs:choice>
</xs:complexType>

<xs:complexType name="tObjectEvent">
  <xs:complexContent>
    <xs:extension base="esds:tAbstractEvent">
      <xs:sequence>
        <xs:element
          name="objectID"
          type="esds:tObjectID"/>
        <xs:element
          name="messageDigest"
          type="esds:tMessageDigest"
          minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="tAggregationEvent">
  <xs:complexContent>
    <xs:extension base="esds:tAbstractEvent">
      <xs:sequence>
        <xs:element
          name="parentID"
          type="esds:tObjectID"
          minOccurs="0"/>
        <xs:element
          name="childIDList"
          type="esds:tObjectIDList"
          minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tAction">
  <xs:restriction base="xs:token">
    <xs:enumeration value="ADD">
```

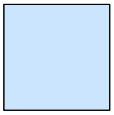
```
        <!-- if parent element is ObjectEvents, ADD means birth of identifier.
             if parent element is AggregationEvents, ADD means children added to parentID
(Aggregation).
        -->
      </xs:enumeration>
      <xs:enumeration value="DELETE">
        <!-- if parent element is ObjectEvents, DELETE means death of identifier.
             if parent element is AggregationEvents, DELETE means children disaggregated from parent
(if child list empty then object is totally disaggregated) (Disaggregation).
        -->
      </xs:enumeration>
      <xs:enumeration value="OBSERVE">
        <!-- if parent element is ObjectEvents, OBSERVE is used for all events which are neither birth
or death of an identifier.
             if parent element is AggregationEvents, OBSERVE means parent and child observed together
but no aggregation and no disaggregation events took place.
        -->
      </xs:enumeration>
      <xs:enumeration value="RELABEL">
        <!-- if parent element is ObjectEvents, RELABEL action cannot be applied.
             if parent element is AggregationEvents, RELABEL means that an objects identifier has
changed. The relabelling can be thought of as 'encapsulating', where the original ID will be the child
and the new ID will be the parent.
        -->
      </xs:enumeration>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="EventLookupIn">
    <xs:complexContent>
      <xs:extension base="esds:tAbstractIn">
        <xs:sequence>
          <xs:choice>
            <xs:element
              name="sid"
              type="esds:tSessionID" minOccurs="0"/>
            <xs:element
              name="login"
              type="esds:tLoginCreds" minOccurs="0"/>
          </xs:choice>
          <xs:element
            name="supplyChainID"
            type="esds:tSupplyChainID"/>
          <xs:choice>
            <xs:element
              name="MATCH_parentIDList"
              type="esds:tObjectIDList" minOccurs="0"/>
            <xs:element
              name="MATCH_childIDList"
              type="esds:tObjectIDList" minOccurs="0"/>
            <xs:element
              name="MATCH_anyIDList"
              type="esds:tObjectIDList" minOccurs="0"/>
          </xs:choice>
          <xs:element
            name="lifeCycleStepID"
            type="esds:tLifeCycleStepID"
            minOccurs="0"/>
          <xs:element
            name="startingAt"
            type="xs:dateTime"
            minOccurs="0"/>
          <xs:element
            name="endingAt"
            type="xs:dateTime"
            minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="EventInfoOut">
    <xs:complexContent>
      <xs:extension base="esds:tAbstractOut">
        <xs:sequence>
          <xs:element
            name="event"
            type="esds:tEventTypeChoice"/>
          <xs:element
            name="voided"
            type="esds:tVoided"
```

```
            minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

In addition to the underlying data model discussed above, there will also be a framework for expressing access control policies in a machine-readable manner. Within this framework, a supply chain fragment may be expressed as a particular cluster of companies. There may be some access rights that are common to companies within that supply chain fragment. There may be other factors, such as time windows that are used to control visibility of events. When a DS enforces these access control policies on behalf of the publisher, the effect is to restrict the amount of event information that is returned to a particular client, usually based on their trust relationship and/or contractual relationship with the company that posted each particular link or event. The restriction may take the form of denying access to some events completely - but could also result in hiding some data fields from particular events - so for example, an aggregation event might be returned but with only the parent or only certain child EPCs being present - or an event might be returned with the life cycle step (bizStep) field missing if the publisher does not want to reveal that.

Legend:

An event being published to Discovery Service for object with identifier 1. Two arrows following each other implies the sequence of the events, the time lapse between the two events does not have any restrictions.



Coloured background refers to a Discovery Service, so if the background colour changes it means that the object has moved between two Discovery Service nodes.

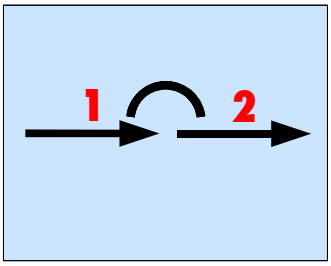
The final goal is that all scenarios will be fully traceable by an authorized users. No Scenario should result in a broken chain in following the sequence of events on the object of interest.

The XML ESDS commands given in the examples are incomplete and only contain relative information (not all required attributes and elements are present).

Relabelling: e.g. object 1 is relabelled to object 2

Relabelling recorded via aggregationEvent (sorted by event source timestamp):

```
<EventCreateIn>
   <objectEvent>
      <objectID>1</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>2</parentID>
      <childID><objectID>1</objectID></childID>
      <action>RELABEL</action>
   </aggregationEvent>
   <objectEvent>
      <objectID>2</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <objectEvent>
      <objectID>1</objectID>
      <action>DELETE</action>
   </objectEvent>
</EventCreateIn>
```



MATCH_parentID lookup on Object ID 1 will return:

```
<EventLookupIn>
   <MATCH_parentID>
<objectID>1</objectID>
   </MATCH_parentID>
      </EventLookupIn>
```

```
<EventLookupOut>
   <objectEvent>
      <objectID>1</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <objectEvent>
      <objectID>1</objectID>
      <action>DELETE</action>
   </objectEvent>
</EventLookupOut>
```

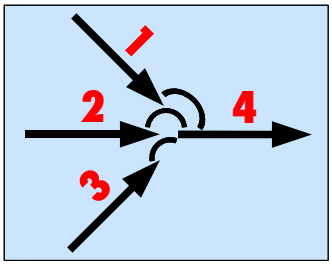
MATCH_parentID lookup on Object ID 2 will return:

```
                                                                    <EventLookupIn>
                                                                  <MATCH_parentID>
                                                               <objectID>2</objectID>
                                                                 </MATCH_parentID>
                                                                    </EventLookupIn>
<EventLookupOut>
   <aggregationEvent>
      <parentID>2</parentID>
      <childID><objectID>1</objectID></childID>
      <action>RELABEL</action>
   </aggregationEvent>
   <objectEvent>
      <objectID>2</objectID>
      <action>OBSERVE</action>
   </objectEvent>
</EventLookupOut>
```

MATCH_anyEPC lookup on Object ID 1 will return:

```
                                                                    <EventLookupIn>
                                                                   <MATCH_anyEPC>
                                                               <objectID>1</objectID>
                                                                  </MATCH_anyEPC>
                                                                    </EventLookupIn>
<EventLookupOut>
   <objectEvent>
       <objectID>1</objectID>
       <action>OBSERVE</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>2</parentID>
      <childID><objectID>1</objectID></childID>
      <action>RELABEL</action>
   </aggregationEvent>
   <objectEvent>
      <objectID>1</objectID>
      <action>DELETE</action>
   </objectEvent>
</EventLookupOut>
```

Aggregation: e.g. objects 1,2,3 are put together into a new package labelled 4



Event sequence (sorted by event source timestamp):

```
<EventCreateIn>
   <objectEvent>
      <objectID>1</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <objectEvent>
      <objectID>2</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <objectEvent>
      <objectID>3</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <objectEvent>
      <objectID>4</objectID>
      <action>ADD</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>4</parentID>
      <childID>
        <objectID>1</objectID>
        <objectID>2</objectID>
        <objectID>3</objectID>
      </childID>
      <action>ADD</action>
   </aggregationEvent>
</EventCreateIn>
```

MATCH_parentID lookup on Object ID 1 will return:

```
<EventLookupOut>
   <objectEvent>
       <objectID>1</objectID>
       <action>OBSERVE</action>
   </objectEvent>
</EventLookupOut>
```

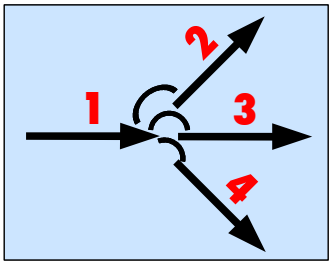
MATCH_parentID lookup on Object ID 4 will return:

```
<EventLookupOut>
   <objectEvent>
      <objectID>4</objectID>
      <action>ADD</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>4</parentID>
      <childID>
        <objectID>1</objectID>
        <objectID>2</objectID>
        <objectID>3</objectID>
      </childID>
      <action>ADD</action>
   </aggregationEvent>
</EventLookupOut>
```

MATCH_anyEPC lookup on Object ID 2 will return:

```
<EventLookupOut>
   <objectEvent>
      <objectID>2</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>4</parentID>
      <childID>
        <objectID>1</objectID>
        <objectID>2</objectID>
        <objectID>3</objectID>
      </childID>
      <action>ADD</action>
   </aggregationEvent>
</EventLookupOut>
```

Disaggregation: e.g. package 1 is broken down into individual objects 1,2,3



Event sequence (sorted by event source timestamp):

```
<EventCreateIn>
   <objectEvent>
      <objectID>1</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>1</parentID>
      <childID>
        <objectID>2</objectID>
        <objectID>3</objectID>
        <objectID>4</objectID>
      </childID>
      <action>DELETE</action>
   </aggregationEvent>
   <objectEvent>
      <objectID>1</objectID>
      <action>DELETE</action>
   </objectEvent>
</EventCreateIn>
```

MATCH_parentID lookup on Object ID 1 will return:

```
<EventLookupOut>
   <objectEvent>
       <objectID>1</objectID>
       <action>OBSERVE</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>1</parentID>
      <childID>
        <objectID>2</objectID>
        <objectID>3</objectID>
        <objectID>4</objectID>
      </childID>
      <action>DELETE</action>
   </aggregationEvent>
   <objectEvent>
      <objectID>1</objectID>
      <action>DELETE</action>
   </objectEvent>
</EventLookupOut>
```

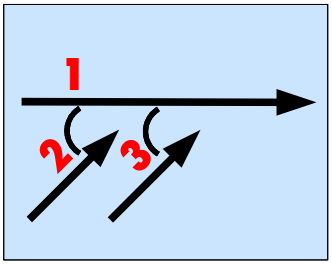
MATCH_parentID lookup on Object ID 4 will return:

```
<EventLookupOut>
</EventLookupOut>
```

MATCH_anyEPC lookup on Object ID 2 will return:

```
<EventLookupOut>
   <aggregationEvent>
      <parentID>1</parentID>
      <childID>
        <objectID>2</objectID>
        <objectID>3</objectID>
        <objectID>4</objectID>
      </childID>
      <action>DELETE</action>
   </aggregationEvent>
</EventLookupOut>
```

Association: e.g. objects 2,3 are placed on container 1



Event sequence (sorted by event source timestamp):

```
<EventCreateIn>
   <objectEvent>
      <objectID>1</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <objectEvent>
      <objectID>2</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <objectEvent>
      <objectID>3</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>1</parentID>
      <childID>
        <objectID>2</objectID>
        <objectID>3</objectID>
      </childID>
      <action>ADD</action>
   </aggregationEvent>
</EventCreateIn>
```

MATCH_parentID lookup on Object ID 1 will return:

```
<EventLookupOut>
   <objectEvent>
      <objectID>1</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>1</parentID>
      <childID>
        <objectID>2</objectID>
        <objectID>3</objectID>
      </childID>
      <action>ADD</action>
   </aggregationEvent>
</EventLookupOut>
```

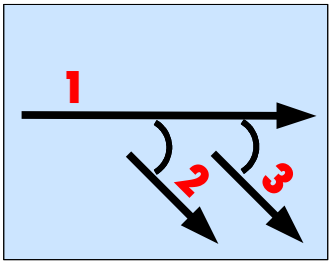
MATCH_parentID lookup on Object ID 2 will return:

```
<EventLookupOut>
   <objectEvent>
      <objectID>2</objectID>
      <action>OBSERVE</action>
   </objectEvent>
</EventLookupOut>
```

MATCH_anyEPC lookup on Object ID 2 will return:

```
<EventLookupOut>
   <objectEvent>
      <objectID>2</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>1</parentID>
      <childID>
        <objectID>2</objectID>
        <objectID>3</objectID>
      </childID>
      <action>ADD</action>
   </aggregationEvent>
</EventLookupOut>
```

Disassociation: e.g. objects 2,3 are taken out of the container 1



Event sequence (sorted by event source timestamp):

```
<EventCreateIn>
   <objectEvent>
      <objectID>1</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>1</parentID>
      <childID>
        <objectID>2</objectID>
        <objectID>3</objectID>
      </childID>
      <action>DELETE</action>
   </aggregationEvent>
</EventCreateIn>
```

MATCH_parentID lookup on Object ID 1 will return:

```
<EventLookupOut>
   <objectEvent>
      <objectID>1</objectID>
      <action>OBSERVE</action>
   </objectEvent>
   <aggregationEvent>
      <parentID>1</parentID>
      <childID>
        <objectID>2</objectID>
        <objectID>3</objectID>
      </childID>
      <action>DELETE</action>
   </aggregationEvent>
</EventLookupOut>
```

MATCH_parentID lookup on Object ID 2 will return:

```
<EventLookupOut>
</EventLookupOut>
```

MATCH_anyEPC lookup on Object ID 2 will return:

```
<EventLookupOut>
   <aggregationEvent>
      <parentID>1</parentID>
      <childID>
        <objectID>2</objectID>
        <objectID>3</objectID>
      </childID>
      <action>DELETE</action>
   </aggregationEvent>
</EventLookupOut>
```