# Requirements from the NAV group – DRAFT

1. **Introduction**

The rationale for the following requirements is to define the needs of the CCSDS NAV-WG regarding the implementation of requirements. Whereas the definitions provided by the CSSM provides a suitable framework that NAV-WG can use there are some details that may be taken into considerations. Some of these are:

* Need to support more than one time scale. The unique use of UTC does not suffice the navigation needs. Application whose reference system is GPS or events generated in a context where no access to UTC time is possible (e.g. on-board generated events based on on-board time).
* Need to support relative time for the event time stamping.
* Need to allow that the events are collected in a container that provides context to the events. Whereas this is not a requirement to the events themselves it is necessary that the definition of the events do not impose limitation in its grouping.
* For compatibility with already existing definition within the NAV-WG, it is necessary (advisable?) that the definition of time is not dependent on the definition of events.

These needs with additional details are compiled in the following set of requirements.

In addition an example events file is provided. Snippets have been extracted from the example to illustrate the requirements in some cases.

**Note to the draft provided by CSSM:**

Because the event is also the class itself, calling this field "event" may be confusing. Actually, encapsulation rules would remove the reference to the container, hence *eventName* should be just *name* as it is already contained in *event* and can only be the name of the event. Analogously *eventType* 🡪 *type*, *eventIdentifier* 🡪 *identifier*, …

This has been taken into account in the requirements below and in the accompanying XML example.

**1.1 Event Requirements**

|  |  |
| --- | --- |
| 1.1.10 | Each type of event shall be uniquely identified by an event identifier.Note: event identifiers are in general application domain specific and therefore the event structure shall mandate the appearance of the event type but not the detailed enumeration of event types in the application domain. |

 <event event="second" type="ordinal" UID="second\_event\_id">

 <epoch>

 <absoluteTime>2017-10-02T23:00:00</absoluteTime>

 </epoch>

 <parameter name="iAmDouble">3.14159265358979323846264</parameter>

 <parameter name="iAmNot">3.1416</parameter>

 <parameter name="iAmUnknown">0xFF4523C0</parameter>

 </event>

|  |  |
| --- | --- |
| 1.1.20 | Each individual event shall allow for a unique identifier of the event.Note: This is a unique identifier within the event that can be used to uniquely identify the event. |

 <event event="second" type="ordinal" UID="second\_event\_id">

 <epoch>

 <absoluteTime>2017-10-02T23:00:00</absoluteTime>

 </epoch>

 <parameter name="iAmDouble">3.14159265358979323846264</parameter>

 <parameter name="iAmNot">3.1416</parameter>

 <parameter name="iAmUnknown">0xFF4523C0</parameter>

 </event>

|  |  |
| --- | --- |
| 1.1.30 | The event structure shall allow children parameters to qualify the event. |

 <event event="second" type="ordinal" UID="second\_event\_id">

 <epoch>

 <absoluteTime>2017-10-02T23:00:00</absoluteTime>

 </epoch>

 <parameter name="iAmDouble">3.14159265358979323846264</parameter>

 <parameter name="iAmNot">3.1416</parameter>

 <parameter name="iAmUnknown">0xFF4523C0</parameter>

 </event>

|  |  |
| --- | --- |
| 1.1.40 | Each event type shall have a predefined set of fully qualified parameters (that may be none).Note: this would be, e.g. an enumerated type in the XML schema. |

**1.2 Event Timing Requirements**

|  |  |
| --- | --- |
| 1.2.10 | Events shall contain timing information |

 <event event="second" type="ordinal" UID="second\_event\_id">

 <epoch>

 <absoluteTime>2017-10-02T23:00:00</absoluteTime>

 </epoch>

 <parameter name="iAmDouble">3.14159265358979323846264</parameter>

 <parameter name="iAmNot">3.1416</parameter>

 <parameter name="iAmUnknown">0xFF4523C0</parameter>

 </event>

|  |  |
| --- | --- |
| 1.2.20 | It shall be possible to express the timing information as either an absolute or a relative time. |

|  |  |
| --- | --- |
| 1.2.30 | Absolute time shall consist of: * an absolute time stamp: date + time.
* an optional time scale.
 |

 <event event="second" type="ordinal" UID="second\_event\_id">

 <epoch time\_system="TAI">

 <absoluteTime>2017-10-02T23:00:00</absoluteTime>

 </epoch>

 <parameter name="iAmDouble">3.14159265358979323846264</parameter>

 <parameter name="iAmNot">3.1416</parameter>

 <parameter name="iAmUnknown">0xFF4523C0</parameter>

 </event>

|  |  |
| --- | --- |
| 1.2.35 | The representation of absolute time shall adhere to CCSDS Type A and Type B. |

|  |  |
| --- | --- |
| 1.2.40 | Relative time shall consist of: * an elapsed time expressed as a real number.
* an optional time scale.
* an optional reference epoch (from which the elapsed time is measured).

Note: respecting the specification to use SI units, the elapsed time should be given in seconds. |

 <event event="fourth" type="ordinal">

 <epoch time\_system="TAI">

 <relativeTime>-5.0</relativeTime>

 <epochTime>2017-10-02T23:00:00</epochTime>

 </epoch>

 </event>

|  |  |
| --- | --- |
| 1.2.50 | The reference epoch shall be an absolute time (as defined in 1.2.30) or a reference to another event (by its unique identifier as in 1.1.20). |

 <event event="sixth" type="ordinal">

 <epoch>

 <relativeTime>-5.0</relativeTime>

 <eventTime ref="second\_event\_id" />

 </epoch>

 </event>

|  |  |
| --- | --- |
| 1.2.60 | The optional time scale shall default to UTC or implicitly to the scale defined in the events container (see 1.3 below).Note: The events container refers to the element in the data hierarchy that embraces a set of events to be managed as a collection (e.g. an events list, a class EventsCollection, etc.) |

 <metadata>

 <OBJECT\_NAME>HOME</OBJECT\_NAME>

 <OBJECT\_ID>19651002</OBJECT\_ID>

 <CENTER\_NAME>EARTH</CENTER\_NAME>

 <TIME\_SYSTEM>UTC</TIME\_SYSTEM>

 <REFERENCE\_EPOCH>2016-10-18T17:50</REFERENCE\_EPOCH>

 </metadata>

|  |  |
| --- | --- |
| 1.2.70 | The optional time scale (in each context) shall override the time scale of the containing context.Note: this requirement is meant to provide flexibility in the management of events with more than one time scale simultaneously. |

 <metadata>

 <OBJECT\_NAME>HOME</OBJECT\_NAME>

 <OBJECT\_ID>19651002</OBJECT\_ID>

 <CENTER\_NAME>EARTH</CENTER\_NAME>

 <TIME\_SYSTEM>UTC</TIME\_SYSTEM>

 <REFERENCE\_EPOCH>2016-10-18T17:50</REFERENCE\_EPOCH>

 </metadata>

 <data>

 <event event="first" type="ordinal">

 <epoch time\_system="TAI">

 <relativeTime>1.3</relativeTime>

 </epoch>

 <parameter name="iAmBoolean">true</parameter>

 <parameter name="iAmString">Lorem ipsum dolor sit amet</parameter>

 </event>

|  |  |
| --- | --- |
| 1.2.80 | For absolute timing, the order of priority in the assignment of time scales shall be:* Own time scale given.
* Time scale of the events container.
* UTC
 |

|  |  |
| --- | --- |
| 1.2.90 | For relative time, the order of priority in the assignment of time scales shall be:* Own time scale given.
* Time scale of the reference epoch (if given).
* Time scale of the events container.
* UTC
 |

|  |  |
| --- | --- |
| 1.2.100 | The definition of the time (absolute or relative) shall be independent of the definition of the events.Note: the naming of the time types shall then be independent of the containing element, whether an event or any other data type. Hence AbsoluteEventTime 🡪 AbsolutTime, RelativeEventTime 🡪 RelativeTime, etc. |

|  |  |
| --- | --- |
| 1.2.110 | The definition of the time scales shall be independent of the definition of the events. |

|  |  |
| --- | --- |
| 1.2.120 | Event timing shall allow for optional timing offsets around the event epoch (i.e. timeLatestOffset and timeLatestOffset) given as real numbers.Note: respecting the specification to use SI units, the time offsets should be given in seconds. This requirements arises from the need to support events at multi-Hertz rate. |

**1.3 Event Container Requirements**

|  |  |
| --- | --- |
| 1.3.10 | It shall be possible to collect events in a container (e.g. events list) |

|  |  |
| --- | --- |
| 1.3.20 | The events container shall allow attributes to provide general information for events and default values.  |

|  |  |
| --- | --- |
| 1.2.30 | The events container shall allow the definition of a default time scale for the contained events. |