

# ***Memorandum***

---

**To** :  
**From** : GSI/C/SZ  
**Copy** :  
**Ref.** : EUM/OPS/DOC/13/689200, v1  
**Date** : 31 January 2013

---

**Subject** : **Comments\_on\_PDGS-SAFE-GMV-TN-13-0230\_SAFE\_Design\_pdf**

## 1. Section 2

Is a mapping foreseen between the schema packages and their contents? I.e., how would an application, which is to create a new EO Product or Auxiliary Package, know which package (and thus containing file) to reference for its representation information, i.e. how does this application identify which logical name to refer to?

## 2. Section 3

- a. I cannot see that a schema file stored in a package can become easily obsolete. It does this only when ultimately all references to it infer that this is the new applicable version. How can this be ensured?

Example: EO Product Av1 refers to e.g. DFLD schema Dv1. As described, if the version of an EO Product changes, a new EO Product needs to be archived, let's call it Av2. This one now wants to reference DFLD schema Dv2. Both DFLD schemas are in the same package, without any version information. By adding Dv2 one invalidates Dv1 and automatically Av1 – and this is mostly, IMO, not what one wishes. Instead, both should still be usable. The proposed versioning schema prohibits this. I.e., as described in the document “Obsolete schema version...”, I don't really think those become obsolete, they simply become not the most recent but might be still applicable for “older” versions of products, unless one really plans to replace something, e.g. because of errors...

I propose to rethink this approach. From the top of my head one could include version information in filenames, or via directories inside the SAFE packages. e.g. .../3.2/schema.xsd. I seem to remember as well that SAFE asked for identifying applicable versions in references to internal or externally kept files, and this is common practice.

- b. ISO 8601 requires the “T” as a separator between date and time! The proposed naming extension for renamed files is not ISO 8601 compliant.

## 3. Section 4

- a. The logical identifier for files inside packages is not understood. The proposal reads that a file is identified only by a number. I fail to understand how this number maps to a physical file. What is the mapping applied here? Sort by date/time, sort by filename, other? So I don't really understand how one would be able to refer from one, say, EO package to an externalised file. Obviously one would need the package-id as well as the file-id. If this is solely managed by the mapper as one can see in the example as of section 5.1, IMO, a certain risk is introduced. I propose to identify both packages and names by logical identifiers which are somehow meaningful. The given example in section 5.1 as well places a big burden on the mapper as it requires to map not only physical positions of packages but as well of single files. I would expect that this is over the top and that it can be solved differently, as each single file can be identified inside a package, e.g. by its (file)name. An alternative approach could be, e.g., to transfer the role of mapping the filenames to the manifest of the package itself, i.e. in the manifest of a referenced package it is defined how a logical identifier of a file used elsewhere is related to a physical file inside a package. This would be enough as it is not planned to move single files of packages around. The atomic entity wrt. archiving is "package" and not "file".
- b. In section 4.3.2 it is said that the mapper should be able to associate the logical identifiers of several packages, i.e. implement a 1..\* relationship. I fail to understand why this is necessary. In my understanding the references from e.g. EO package to AUX data is done via the manifest of the EO package only, by using the logical identifiers as described.
- c. The examples given in section 4.3.3 are wrong if one would take the comments made above into consideration and would need to be reworked.

#### 4. Section 5

The examples need to be revisited if any of the above made comments will be considered for changing the presented specification/design.

I am happy to get into discussions about the above, when necessary.