

STORAGE SPACE REPORTING

... an iterative process trying to make the space reporting json more GLUE2 compliant

- step 0) Ale's original proposal (
https://indico.cern.ch/event/394833/contributions/2271993/attachments/1335423/2008468/2016-StorageAccounting-Evolution_v0.8.pdf)

```
{ "capacity_id": "ATLASDATADISK",
  "status": "online/offline",
  "status_message": "The report can not be created because ...",
  "list_of_paths": [ "/castor/ads.rl.ac.uk/prod/atlas/stripInput/atlasdatadisk/" ],
  "total_space": 5000000000,
  "used_space": 2000000000,
  "num_files": 123456,
  "time_stamp": 1447936989 },
{ "capacity_id": "ATLASSCRATCHDISK",
  ...
},
{ "capacity_id": "ATLASLOCALGROUPDISK",
  ...
}
```

- step 1) Paul Millar proposed "One can describe most of what you want already with GLUE and it has a JSON rendering. Here's a rough idea of how the output would look like:"

```
{ "StorageShare" : [
  {
    "CreationTime": "...",
    "Name": "ATLASDATADISK",
    "ServiceState": "production",
  },
  {
    "CreationTime": "...",
    "Name": "ATLASSCRATCHDISK",
    "ServiceState": "production",
  },
  {
    "CreationTime": "...",
    "Name": "ATLASGROUPDISK",
    "ServiceState": "production",
  },
],
"StorageShareCapacity" : [
  {
    "Name": "ATLASDATADISK disk usage",
    "TotalSize": "5000000000",
    "UsedSize": "2000000000",
  },
  {
    "Name": "ATLASSCRATCHDISK disk usage",
    "TotalSize": "...",
    "UsedSize": "..",
  },
  {
    "Name": "ATLASGROUPDISK disk usage",
    "TotalSize": "...",
    "UsedSize": "...",
  }
]
```

```

}
]
}

```

- step 2: Paul's more elaborated GLUE2 structure:

```

{
  "StorageShare": [
    {
      "Associations": {
        "StorageShareCapacityID": [
          "atlas:big-site.example.org:ATLASDATADISK-usage"
        ]
      }
      "CreationTime": "2016-11-02T12:16:03Z",
      "ID": "atlas:big-site.example.org:ATLASDATADISK",
      "SharingID": "ATLASDATADISK",
      "ServingState": "production", # or closed|draining|queueing
      "Path": "/atlas/data",
      "AccessLatency": "ONLINE",
      "Tag": "ATLASDATADISK",
      "OtherInfo": [
        "NumberOfFiles=123456",
        "Path=/vo/atlas/data",
        "Path=/users/atlas-vo/data",
        "Problem=The report cannot be created because ...",
      ]
    },
    {
      "Associations": {
        "StorageShareCapacityID": [
          "atlas:big-site.example.org:ATLASSCRATCHDISK-usage"
        ]
      }
      "CreationTime": "2016-11-02T12:16:03Z",
      "ID": "atlas:atlas-site.example.org:ATLASSCRATCHDISK"
      "SharingID": "ATLASSCRATCHDISK",
      "ServingState": "production",
      "Path": "/atlas/scratch",
      "AccessLatency": "ONLINE",
      "Tag": "ATLASSCRATCHDISK"
    },
    {
      "Associations": {
        "StorageShareCapacityID": [
          "atlas:big-site.example.org:ATLASLOCALGROUPDISK:usage"
        ]
      }
      "CreationTime": "2016-11-02T12:16:03Z",
      "ID": "atlas:atlas-site.example.org:ATLASLOCALGROUPDISK"
      "SharingID": "ATLASLOCALGROUPDISK",
      "ServingState": "production",
      "Path": "/atlas/local",
      "AccessLatency": "ONLINE",
      "Tag": "ATLASLOCALGROUPDISK"
    }
  ],
  "StorageShareCapacity": [
    {
      "Associations": {
        "StorageShareID": "atlas:big-site.example.org:ATLASSCRATCHDISK"
      }
      "CreationTime": "2016-11-02T12:16:03Z",
      "ID": "atlas:big-site.example.org:ATLASSCRATCHDISK:usage",

```

```

    "TotalSize": 5000, # NB. values are in GB (not GiB)
    "UsedSize": 2000
  },
  {
    "Associations": {
      "StorageShareID": "atlas:big-site.example.org:ATLASDATADISK"
    }
    "CreationTime": "2016-11-02T12:16:03Z",
    "ID": "atlas:big-site.example.org:ATLASDATADISK:usage",
    "TotalSize": 3000,
    "UsedSize": 1298
  },
  {
    "Associations": {
      "StorageShareID": "atlas:big-site.example.org:ATLASLOCALGROUPDISK"
    }
    "CreationTime": "2016-11-02T12:16:03Z",
    "ID": "atlas:atlas-site.example.org:ATLASLOCALGROUPDISK:usage",
    "TotalSize": 4000,
    "UsedSize": 3170
  }
]
}

```

- step 3) I commented that no need for such a complex structure and we could merge some of the GLUE2 objects and posted a quick idea of a simplified StorageShare object.

```

"StorageShare": [
  "CreationTime": "2016-11-02T12:16:03Z",
  "ID": "atlas:big-site.example.org:ATLASDATADISK",
  "SharingID": "ATLASDATADISK",
  "ServingState": "production", # or closed|draining|queueing
  "Path": "/atlas/data",
  "AccessLatency": "ONLINE",
  "Tag": "ATLASDATADISK",
  "OtherInfo": [
    "NumberOfFiles=123456",
    "Path=/vo/atlas/data",
    "Path=/users/atlas-vo/data",
    "Problem=The report cannot be created because ...",
  ]
  "TotalSize": 3000, # these are the two capacity attributes
  "UsedSize": 1298 # these are the two capacity attributes
],

```

I got to this fragment by merging the "useful" attributes of the StorageShare.Capacity into the StorageShare because i assumed a one-to-one relation between a Share and its capacity numbers (a Share always has one Totalsize and one UsedSize value).

Paul rightfully commented that the above is not a valid GLUE2 Json any longer. Yes, that's true but my idea was to re-use the GLUE2 attributes and if necessary deviate from the GLUE2 structures

- step 4) Couple of iterations later (on a very long email thread) we arrived to this structure:

"here comes the cleaned up version of the final iteration example for storage space reporting based on GLUE2 terminology. The storage space is modelled by the (modified) GLUE2 StorageShare entity."

```

{ "StorageShare": [
  "PolicyRule": "vo:ATLAS",

```

```
"CreationTime": "2016-11-02T12:16:03Z",
"ID": "atlas:big-site.example.org:ATLASDATADISK",
"SharingID": "HUMAN_READABLE_NAME_OF_THE_SHARE",
"ServicingState": "production",
"Path": ["/atlas/data", "/users/atlas-vo/data", "/another/path"],
"AccessLatency": "ONLINE",
"Tag": ["ATLASDATADISK", "SCRATCHDISK"],
"TotalSize": 3000,
"UsedSize": 1298,
"NumberOfFiles": 123456,
"OtherInfo": "The report was created successfully, all green."
]
},
```

please note that i kept the GLUE2 JSON rendering idea where entities are represented as arrays.

This will be useful if we want to describe other storage objects as well (like storage endpoints, so on), and not just the storage space.

The object is composed of GLUE2 attributes but violates the GLUE2 JSON rendering structure.

Also some extra attributes were necessary to be added.

CONCLUSION:

It is not feasible to use the GLUE2 JSON rendering for the ATLAS storage space reporting.

Nevertheless quite some stuff can be re-used from GLUE2 model. That is what you see in the above step 4 iteration.

If you compare it to Ale's draft (step 0), you'll see it is not much different. Nevertheless what i like in the GLUE2-inspired object that the attributes (the json keys and their values) are well-defined, their meaning is given in a standard document.