EMI proposal for a Storage Accounting Record standard

Jon Kerr Nilsen

Dept. of Physics, Univ. of Oslo

On behalf of the EMI StAR team

Outline

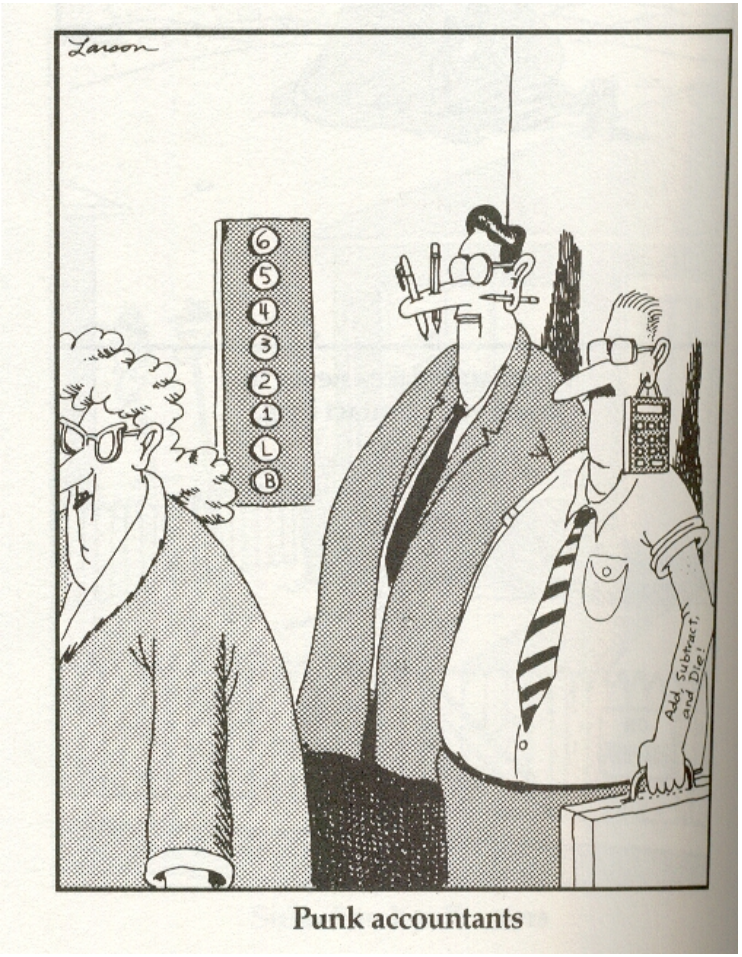
- Why storage accounting?
- Storage accounting vs. job accounting
- Processing model
- Storage accounting work in EMI and OGF
- StAR proposal
- Way forward

Why storage accounting?

- Storage resources are shared between organizations – more than one owner
- Developing a storage infrastructure
  - We need to know how much storage space is used, by which group/user, on which storage media
  - To know where to put the money when increasing the storage space
  - To know who to ask for the money to increase the storage space
- Basis for billing
  - Storage is expensive
  - Some non-academic resource owners may not like to give it away for free

Accounting

- Accounting is the recording and summarizing of the consumption of a resource by an individual user or a group of users
- Typically used to find out who uses how much of a set of resources
- Typically not used to find out how individual resource components are used



Punk accountants

Storage vs. job accounting

Job accounting

- Job accounting is the recording and summarizing of the consumption of computing resources
- Recording wall-time, cpu-time,

  values.
- When a job finishes it stays finished
- Typically measured per job – information readily available from local batch system
- Standardized through UR1.0

Storage accounting

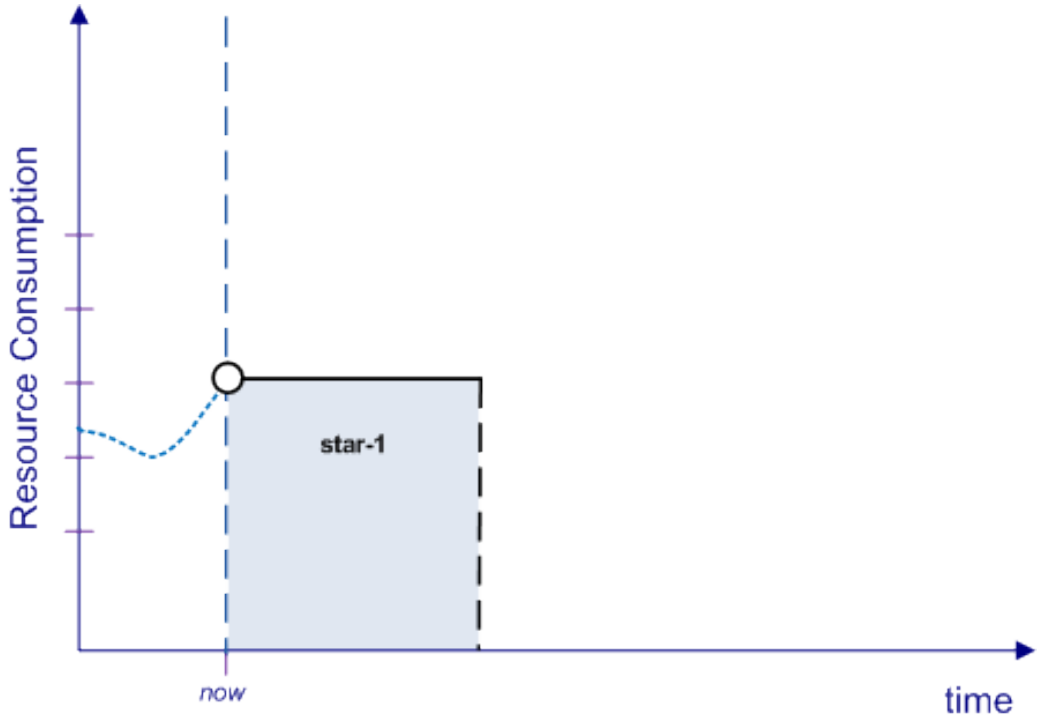
- Storage accounting is the recording and summarizing of the consumption of a storage resource
- Storage usage varies in time – recorded usage is only relevant in a given time frame
- Data is rarely in finished state and can live for a long time
- Typically recorded through (in) frequent usage snapshots – high fluctuation rate

  No standardized way to record storage usage in a distributed environment

Storage vs. job summary

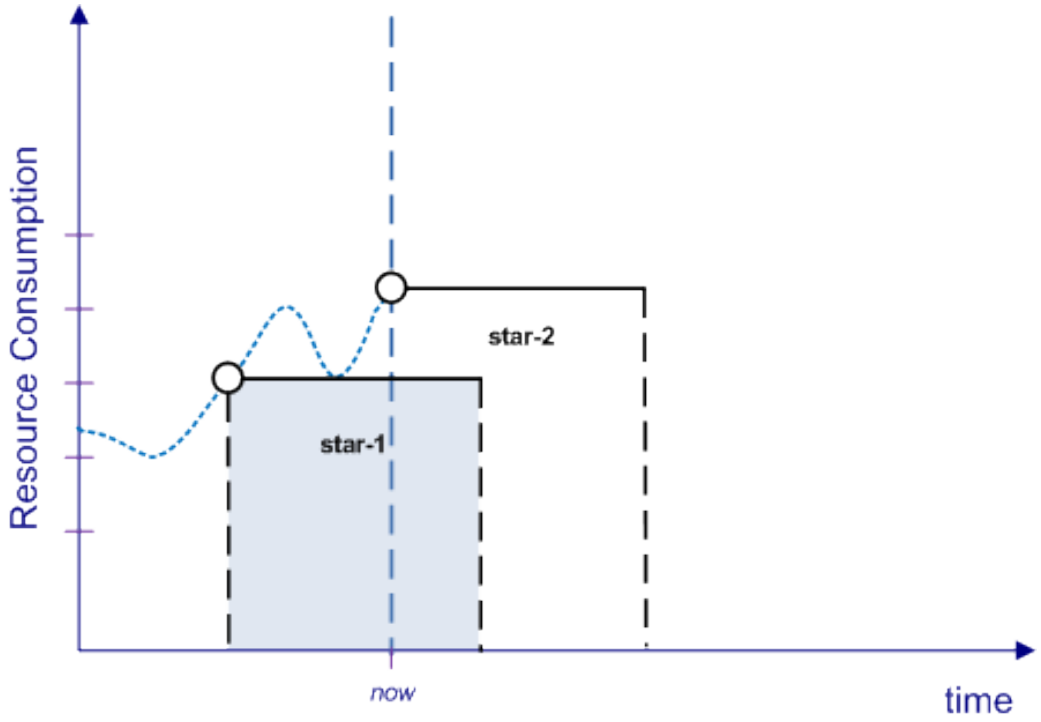
- Storage records must be treated different than compute records wrt. aggregation
- Storage records need special treatment
  - Only valid between measurement time and expiration time
  - Can be invalidated if newer record is registered for same resource
  - Nontrivial processing model

Processing model



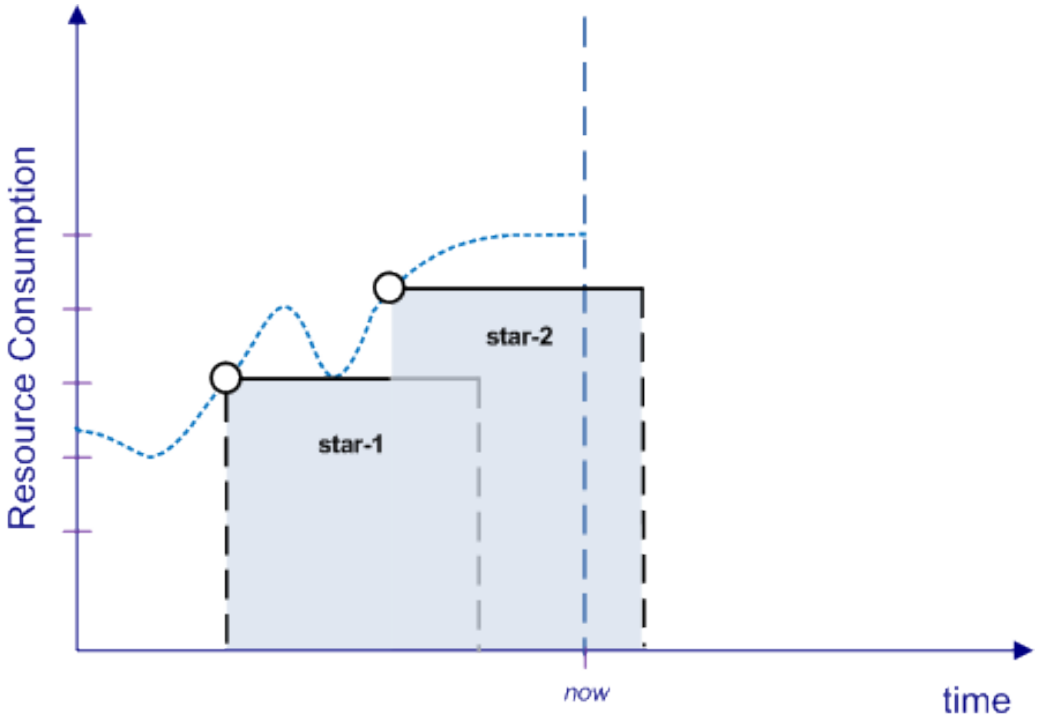
- Records can overlap in time – cannot simply sum up to identify total consumption
- Must ask for a certain point in time and sum up the most recent, still-valid records

Processing model



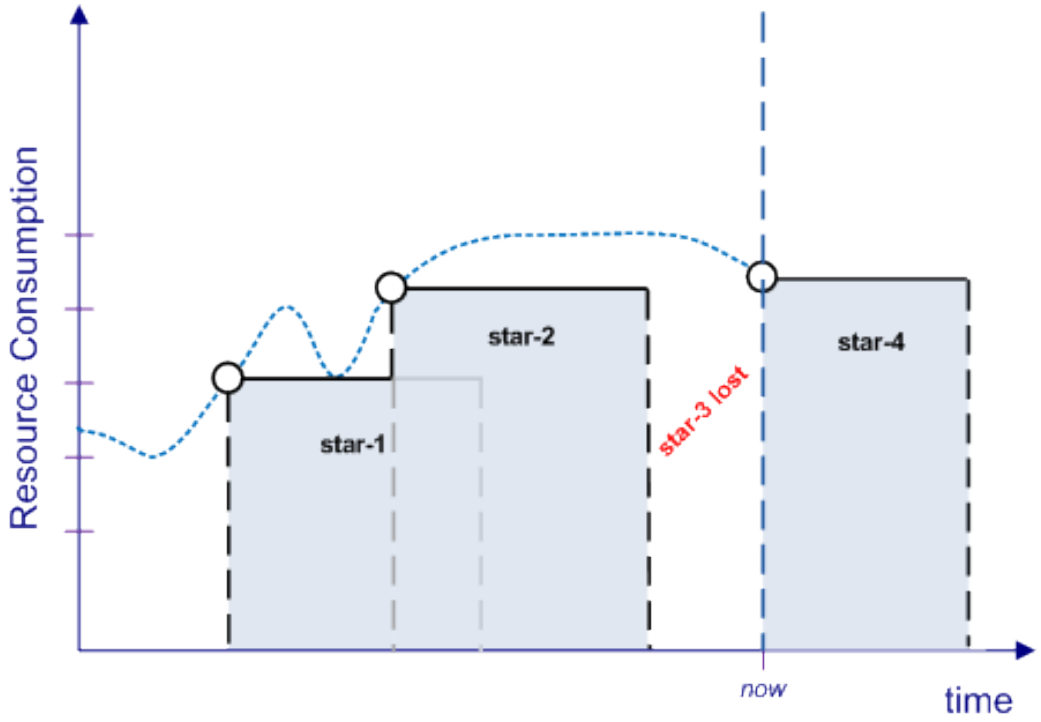
- Records can overlap in time – cannot simply sum up to identify total consumption
- Must ask for a certain point in time and sum up the most recent, still-valid records

Processing model



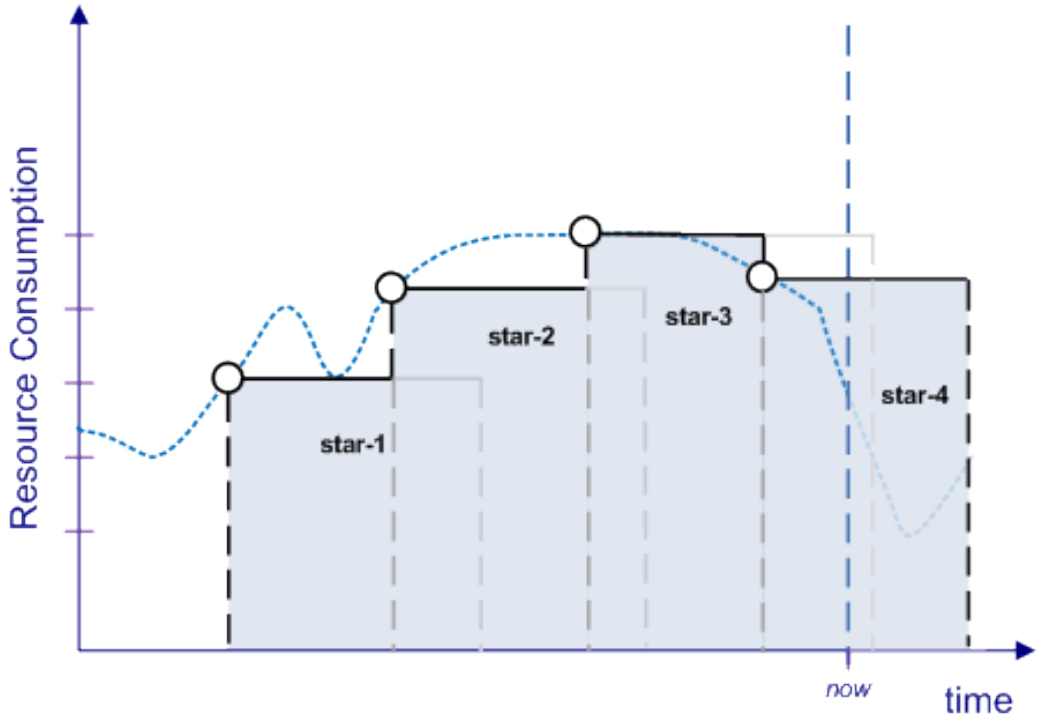
- Records can overlap in time – cannot simply sum up to identify total consumption
- Must ask for a certain point in time and sum up the most recent, still-valid records

Processing model



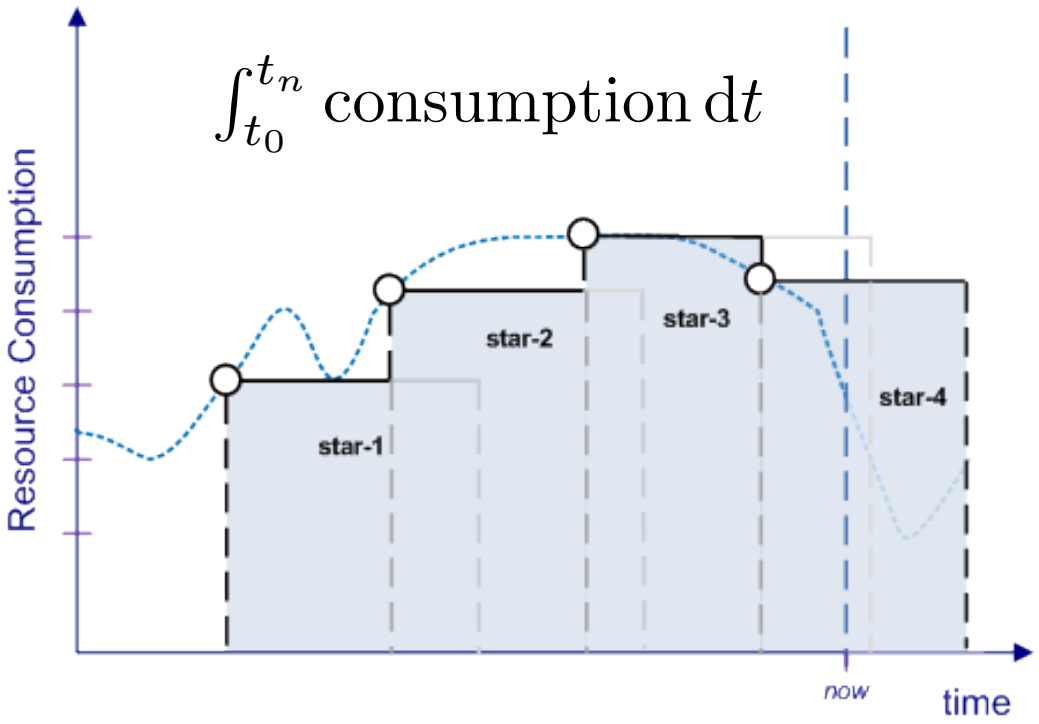
- Records can overlap in time – cannot simply sum up to identify total consumption
- Must ask for a certain point in time and sum up the most recent, still-valid records

Processing model



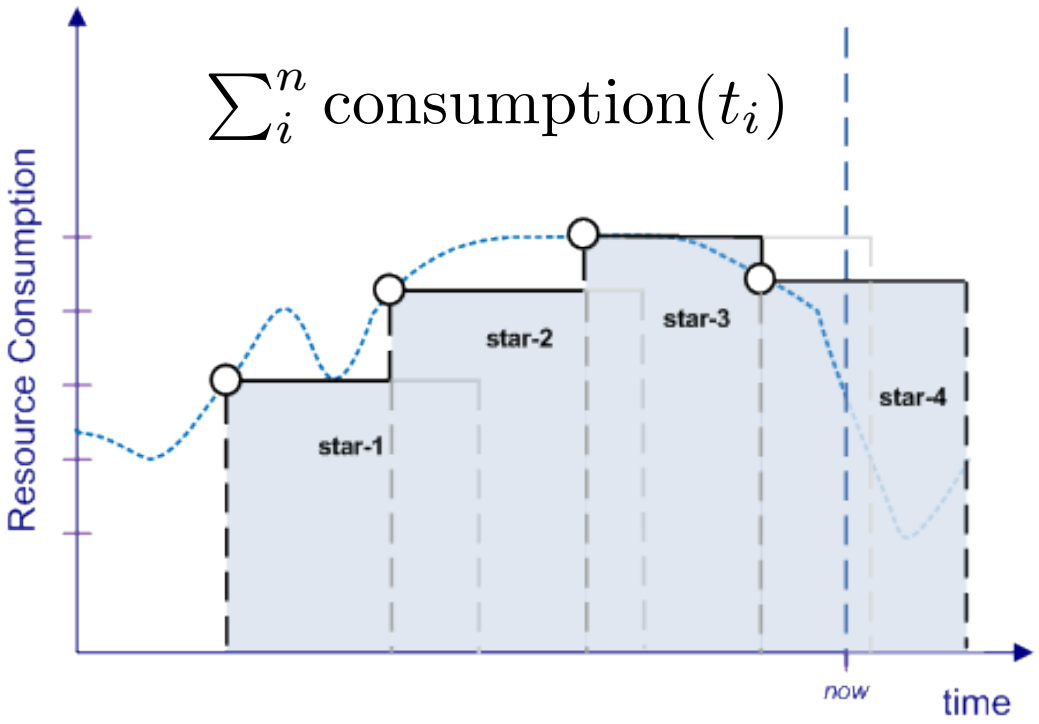
- Records can overlap in time – cannot simply sum up to identify total consumption
- Must ask for a certain point in time and sum up the most recent, still-valid records

Processing model



- Records can overlap in time – cannot simply sum up to identify total consumption
- Must ask for a certain point in time and sum up the most recent, still-valid records
- To get usage per time, you must choose a set of timestamps and repeat process per time stamp – numerical integration

Processing model



- Records can overlap in time – cannot simply sum up to identify total consumption
- Must ask for a certain point in time and sum up the most recent, still-valid records
- To get usage per time, you must choose a set of timestamps and repeat process per time stamp – numerical integration

Storage accounting in EMI

- From EMI Description of Work:
  - EMI will address known UR issues and extend accounting records to include VO-aware storage usage accounting
  - The refined and extended standard UR format, service interface and data transport protocol will be implemented in gLite, ARC, dCache and UNICORE
  - EMI will thus contribute to the definition of UR 2.0 of OGF

Storage accounting in OGF

- 41 mails to UR-WG since september – almost all about storage accounting
- "Next Generation Usage Records" OGF session on Tuesday
  - Discussed the way forward for UR
  - A properly defined storage record and an improved compute record are good steps towards UR2.0
  - Lessons learned in storage records can be ported back into compute records (e.g. long-lived services)
  - EMI will work actively to have standard ready before end of EMI

Why a separate storage record?

- There is already a OGF UR 1.0 (GFD.98)
- Only a limited number of properties in common with what is needed in StAR

  information
- No draft for service record in UR 2.0 yet
- EMI needs record format now to implement before end of EMI project
- Decided to keep the StAR format close to OGF UR 1.0, but still separate
- Merge with UR 2.0 when available

StAR

- EMI is finalizing a proposal for a storage accounting record (StAR)
- Contacted and received input from organizations and potential user groups

- Current draft: StAR-EMI-tech-doc-v7.pdf
- Final version will be proposed to OGF as a standard
- Will be implemented by EMI storage providers

StAR structure

- Resource
  - Fields describing the system the resource was consumed on
  - `StorageSystem`, `StorageShare`
- Resource consumption
  - Resource
  - Fields describing the system the resource was consumed on
  - Resource consumption
  - `ResourceCapacityUsed`,
  - `StorageMedia`
  - Identity - person/group accountable for the consumption
    - `UserIdentity`

Local example

```
<sr:StorageUsageRecord
    xmlns:sr="http://eu-emi.eu/namespaces/2011/02/storagerecord">
  <sr:RecordIdentity sr:createTime="2010-11-09T09:06:52Z"
                     sr:recordId="host.example.org/sr/
87912469269276"
 sr:StorageUsageRecord xmlns:sr="http://
   sr:                  eu-emi.eu/namespaces/2011/02/storagerecord">
  <sr:RecordIdentity sr:createTime="2010-11-09T09:06:52Z"
87912469269276"         sr:recordId
  <                       sr:StorageMedia "host.example.org/sr/
  <sr:SubjectIdentity> <sr:StorageSystem>host.example.org</
    <sr:LocalUser>johndoe</sr:LocalUser
  </sr:SubjectIdentity>ssd</sr:StorageMedia>
  <sr:FileCount>42</                      >
  <                            >
```

Grid example

```
<sr:StorageUsageRecord
    xmlns:sr= http://eu-emi.eu/namespaces/2011/02/storagerecord">
  <sr:RecordIdentity sr:createTime="2010-11-09T09:06:52Z"
                     sr:recordId="host.example.org/sr/
879124692692 76"/>
  <sr:              >host.example.org</sr:StorageSystem>
  <sr:SubjectIdentity>
    xmlns:sr
  <sr:RecordIdentity xmlns:sr="http://eu-emi.eu/namespaces/2011/02/storagerecord">
                     sr              "2010-11-09T09:06:52Z"
                       :recordId=
879124692692 76"/>host.example.org
  <sr:SubjectIdentity>              </sr:StorageSystem>
            >binarydataproject.example.org</sr:Group>
                     sr:              ="subgroup">
                                              ukusers
</sr:StorageUsageRecord>
```

Way forward

- Definition of StAR is a first step to establish common storage accounting record
- A step towards a common usage record (UR2.0)
- Next jump (two steps in one go)
  - Propose StAR to OGF to start standardization process
    - Will take time
    - Will include further discussions
    - May lead to changes in StAR
    - EMI will take active part in this process
    - Draft has been made available through UR-WG mailing list
    - Will be made available in SourceForge
  - Implement StAR record into EMI storage middleware
    - EMI ends in two years
    - Want to gain user experience before end of EMI
- Comments and input more than welcome throughout the process!

THANK YOU!

Contact: emi-jra1-data-sar@eu-emi.eu
Web page: http://twiki.cern.ch/twiki/bin/view/EMI/StorageAccounting
Current StAR draft:
http://twiki.cern.ch/twiki/pub/EMI/StorageAccounting/StAR-EMI-tech-doc-v7.pdf
http://twiki.cern.ch/twiki/pub/EMI/StorageAccounting/StAR-EMI-tech-doc-v7.doc