

Publishing Accounting Data from Outside gLite

Péter Dóbé
<dobe@iit.bme.hu>

Henrik Thstrup Jensen
<htj@ndgf.org>

Balázs Kónya
<balazs.konya@hep.lu.se>



Accounting in EGEE

Sites with gLite deployed use the **APEL** (Accounting Processor for Event Logs) component to submit resource usage data to the GOC, where it is stored in a central accounting database.

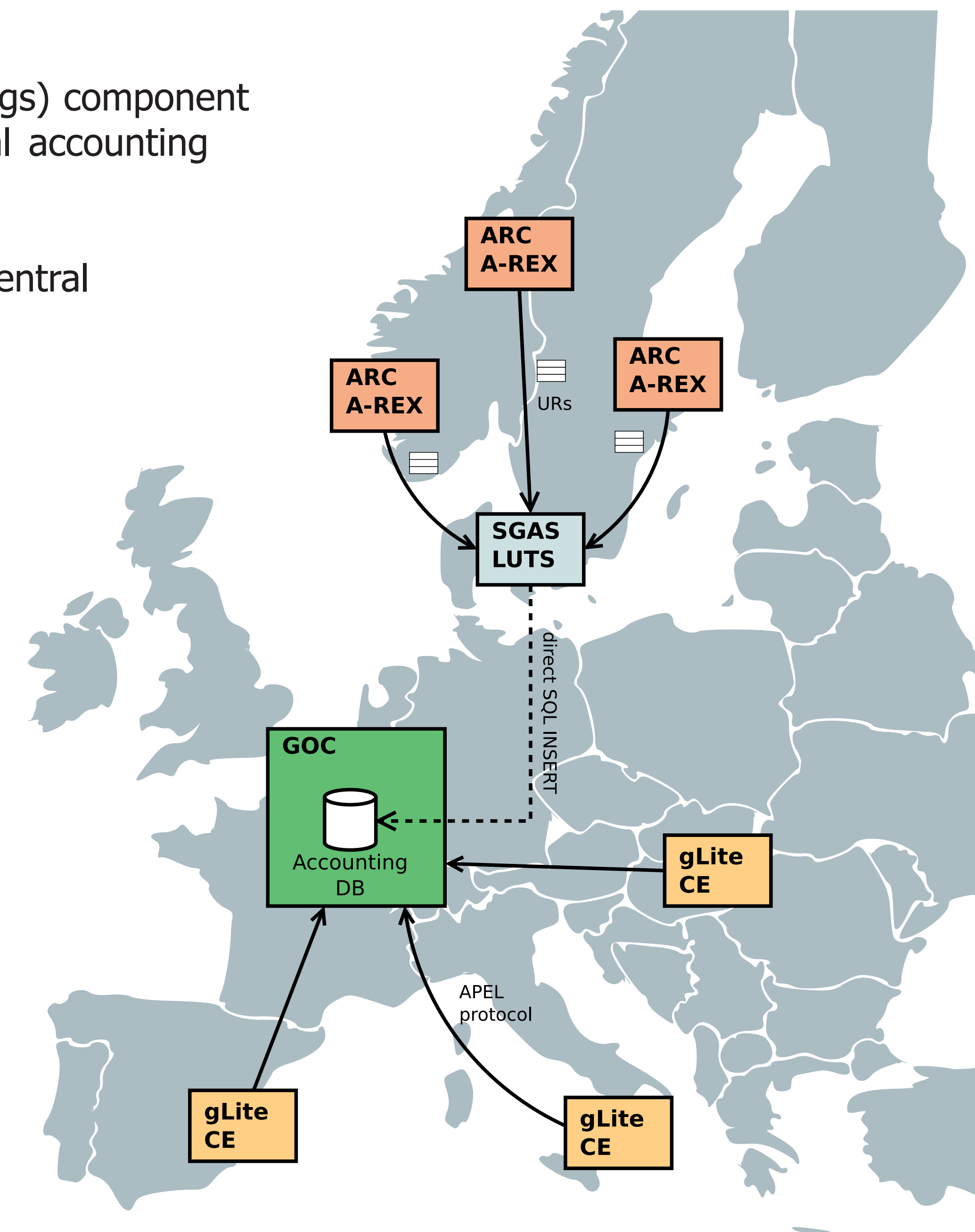
Non-gLite computing elements have difficulty submitting usage data into this central database:

1. APEL is only available for Scientific Linux as part of the gLite distribution.
2. APEL's protocol is complex and lacks detailed documentation, thus it is hard to reimplement.

Current status at NDGF

Computing elements of **NDGF** (Nordic DataGrid Facility) run the **ARC** (Advanced Resource Connector) middleware. **SGAS** (SweGrid Accounting System) has been adopted for regional accounting purpose.

For EGEE jobs (e.g. jobs of ALICE, ATLAS etc.) running on NDGF sites, accounting data is queried from SGAS, transformed from OGF Usage Record to the appropriate format and inserted into the GOC Database. The insertion is done via direct SQL insert, bypassing the protocol of APEL. This crude approach leads to several problems regarding security, robustness etc.



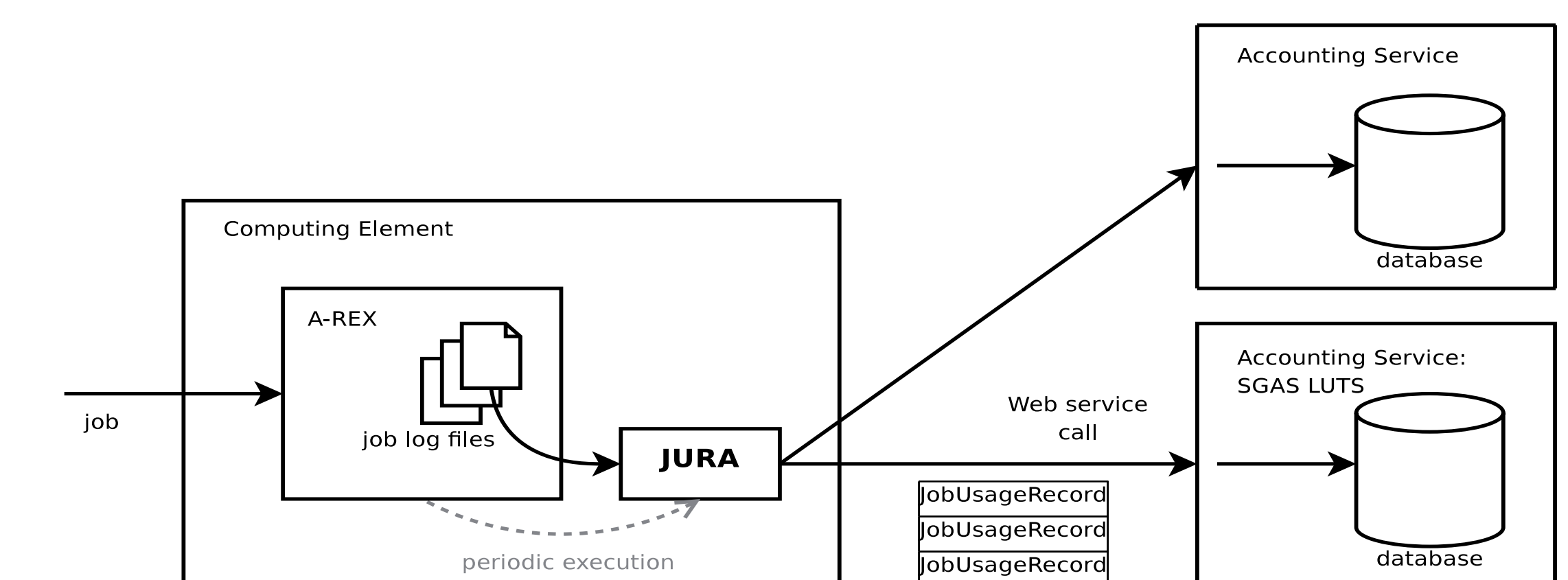
OGF Usage Record Format

The usage data interchanged with LUTS complies with Open Grid Forum's **UR** (Usage Record) specification. UR is a proposed standard to represent different metrics for various types of resources e.g. CPU and wall clock time, disk usage, average and maximum memory usage. OGF has defined an **XML schema** for the UR. This XML representation is used in communication with the LUTS. Example UR XML:

```
<JobUsageRecord xmlns="http://schema.ogf.org/urf/2003/09/urf">
  <RecordIdentity>
    <createTime>2009-06-03T10:30:09Z</createTime>
    <recordId>"ur-ui.example.com-953812440233661461086358"</recordId>
  </RecordIdentity>
  <JobIdentity>
    <GlobalJobId>
      <url>"https://ce.example.com:60000/arc/953812440233661461086358"</url>
    </GlobalJobId>
    <LocalJobId>9763</LocalJobId>
  </JobIdentity>
  <UserIdentity>
    <GlobalUserName>DC=eu/DC=KnowARC/O=TEST/CN=demo1</GlobalUserName>
    <LocalUserId>griduser001</LocalUserId>
  </UserIdentity>
  <Status>completed</Status>
  <Memory>
    <Memory storageUnit="KB" metric="average" type="virtual">5</Memory>
    <Memory storageUnit="KB" metric="max" type="physical">5</Memory>
    <Memory storageUnit="KB" metric="average" type="physical">5</Memory>
  </Memory>
  <WallDuration>PT32S</WallDuration>
  <CpuDuration usageType="user">PT22S</CpuDuration>
  <CpuDuration usageType="kernel">PT1S</CpuDuration>
  <StartTime>2009-06-03T10:02:46Z</StartTime>
  <EndTime>2009-06-03T10:03:18Z</EndTime>
  <MachineName>wn</MachineName>
  <Host primary="true">wn2</Host>
  <SubmitHost>ui.example.com</SubmitHost>
  <Queue>job1</Queue>
  <NodeCount>1</NodeCount>
</JobUsageRecord>
```

JURA: Job Usage Reporter of ARC

JURA is a C++ reimplement of SGAS JARM functionalities, distributed as part of the newest version of the ARC middleware. It creates OGF-compliant URs, and submits them to SGAS LUTS.



Compared to the original JARM, JURA has a number of benefits:

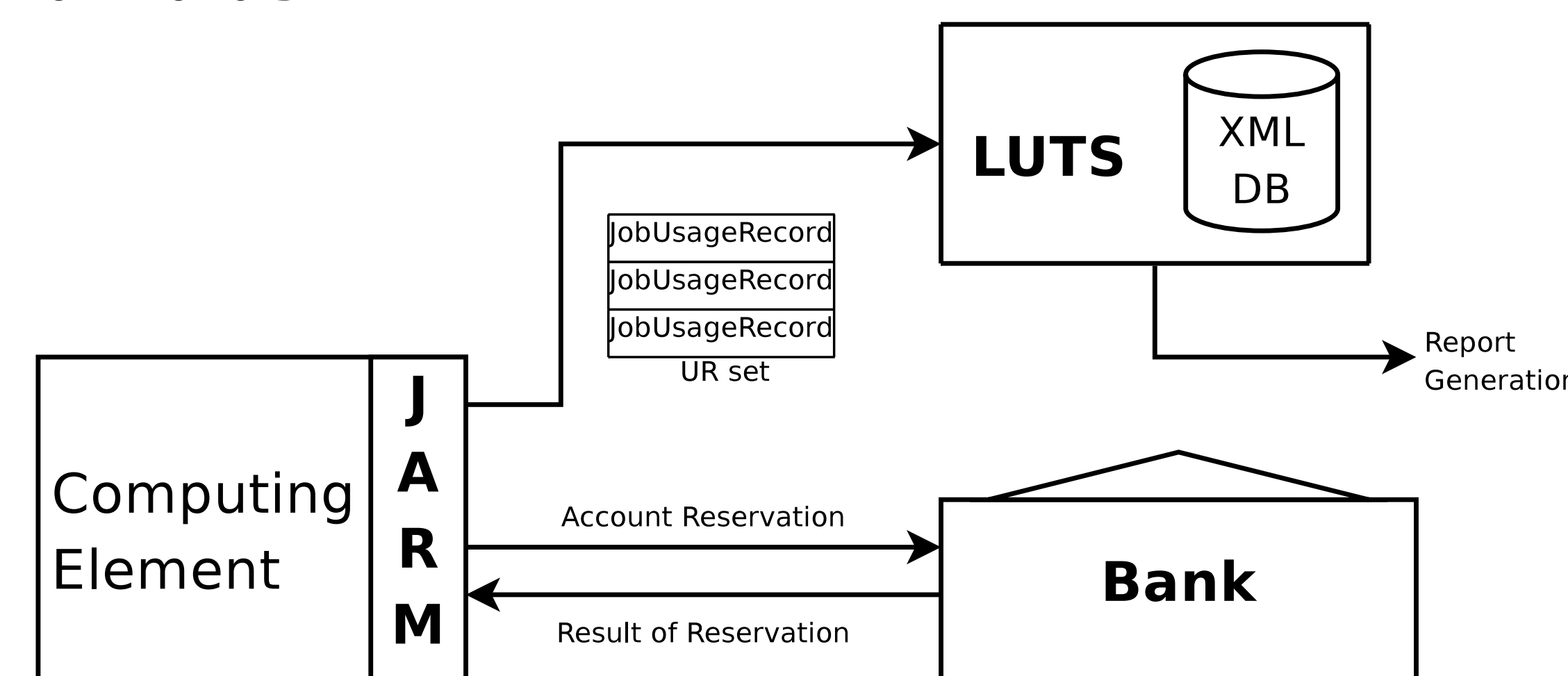
- Provides better integration with **A-REX** (ARC Resource-coupled Execution service), the job manager of ARC.
- No extra dependencies in addition to those of the core components of the middleware.
- Improved performance and robustness.
- Generated URs are more complete and standard-conformant; an ARC-specific extension to the record for the requested **runtime environment** is also added.

In the future, JURA is planned to be enhanced to support other accounting services besides LUTS, including **APEL compatibility** in order to supply accounting data directly to the GOC.

SGAS: SweGrid Accounting System

SGAS is a pure Java grid accounting software originally developed at Umeå University.

The original concept of SGAS contains three main components: **LUTS**, **Bank** and **JARM**.



- **LUTS** (Logging and Usage Tracking Service): Stores resource usage data in a native XML database, and makes it available for report generation (invoicing).
- **Bank**: manages resource usage credits for pre-execution access control
- **JARM** (Job Account Reservation Manager): Client for both the Bank and the LUTS. Resides on the computing element, enforces access control for jobs based on success of credit reservation. After job completion, logs usage information in the LUTS. Due to lack of demand for the Bank service, it was dropped from recent versions of the SGAS distribution.

Future plans for SGAS:

- Create more flexible XML DB bindings.
- Develop a sophisticated report generation tool.
- Longer term plan: change LUTS interface to a standard one.

Contact Information

- Home page of the NorduGrid community: <http://www.nordugrid.org/>
- KnowARC project pages: <http://www.knowarc.eu/>
- SGAS home page: <http://www.sgas.se/>