

## ARC ACCOUNTING COMPONENT – JURA

*Technical document*

Péter Dóbe \*

# 1 Purpose

The *Job Usage Reporter for ARC* (JURA) is a component implementing accounting functionality in the ARC middleware. Its objective is to gather metered resource usage data for each job and submit it to accounting services along with the job submitter's identity and miscellaneous job-related metadata.

The accounting service stores the received usage data in a database, and provides an interface for querying it. Queries can be made by the consumers of the accounting data, such as a billing component. The service itself is a third party application, separate from the middleware distribution. JURA is currently capable of using the logging service of the SweGrid Accounting System (SGAS)[?], but maintaining the possibility to enable utilizing other services has been kept in mind during design.

Before the usage data collected from the resource manager is submitted, it is transformed into records of job-level granularity. To every job corresponds exactly one Grid user, therefore reports over a time period (e.g. an invoice) can be generated per-user, or alternatively on a larger scale such as job project or VO level.

# 2 Architecture

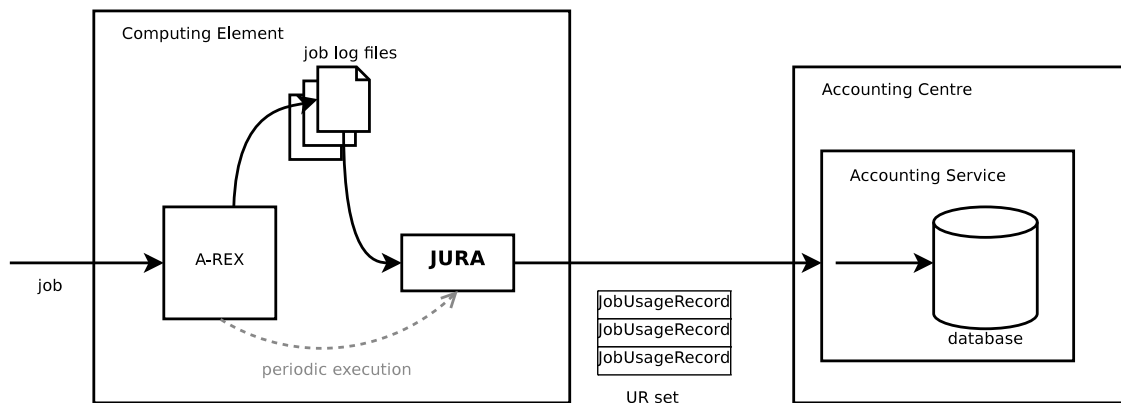


Figure 1: Components participating in accounting.

JURA offers a complete replacement in functionality for the old *logger* utility[?], a job metadata logging tool with a purpose very closely related to that of this new tool. However, backwards compatibility is maintained, so the old logger can still be deployed.

The ARC execution manager, A-REX[?] initiates JURA in a similar way to invoking logger. JURA reads job log files provided by A-REX. These files have the same format as those meant for logger, with additional lines which are necessary for accounting scenarios.

It also acts as a client for one or more accounting services, specifically SGAS Logging and Usage Tracking Services (LUTS's), inserting the generated records in batches. (See Figure 1.)

# 3 Operation

## 3.1 Invocation

JURA is a stand-alone executable application, executed hourly by A-REX. It has no separate configuration file; it gets all necessary configuration options from A-REX, in part through command-line arguments, but mostly via lines in the job log files (see Appendix A for details). The source of the latter are lines in the grid-manager configuration file.

The command line format is a subset of that of logger:

```
jura [-E <expiration_time>] <control_dir>
```

where *expiration\_time* is the validity length of job log files in days, after which time they are considered invalid; *control\_dir* is the A-REX control directory for a mapped local UNIX user.

## 3.2 Parsing job log files

The job logs generated by A-REX reside under the directory `<control_dir>/logs`. They have file name format `<ngjobid>.<random>`, where *ngjobid* is the identifier created for the job by A-REX, *random* is a randomly generated sequence of alphanumeric characters to avoid collision of different files pertaining to the same job.

2 for each logging destination: when submitted, when finished

UR generated from them: only from finished

## 3.3 Accessing LUTS

insert method

certificates from:...

## 4 Security

runs as: same user as a-rex, typically root

accesses sensitive data in job logs

sgas security:

standard ssl cert, possibly proxy

no other attributes but DN considered

access control: flat-file based + XACML-like(?) configured thru SAM porttype

in this case: service credentials, no proxy

## 5 Implementation

c++

sw dependencies

full arc

ssl

install location

## A Configuration

jobreport

jobreport\_options

jobreport\_credentials

## B Usage Record properties

input file format

filled properties

missing properties