



## EGI-Engage

# Accounting Repository Release

### D3.3

---

<b>Date</b>	07 March 2016
<b>Activity</b>	WP3
<b>Lead Partner</b>	STFC
<b>Document Status</b>	FINAL
<b>Document Link</b>	<a href="https://documents.egi.eu/document/2659">https://documents.egi.eu/document/2659</a>

---

### Abstract

This deliverable describes the first release of the EGI Accounting Repository during EGI-Engage including the developments made during the first year of the EGI-Engage project. The EGI Accounting Repository runs using software from the APEL project, which collects accounting data from sites participating in the EGI and WLCG infrastructures as well as from sites belonging to other Grid organisations that are collaborating with EGI. It generates statistical summaries that are available through the EGI/WLCG Accounting Portal to Users, VO Managers and Site Administrators.



This material by Parties of the EGI-Engage Consortium is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

The EGI-Engage project is co-funded by the European Union (EU) Horizon 2020 program under Grant number 654142 <http://go.egi.eu/eng>

**COPYRIGHT NOTICE**

This work by Parties of the EGI-Engage Consortium is licensed under a Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). The EGI-Engage project is co-funded by the European Union Horizon 2020 programme under grant number 654142.

**DELIVERY SLIP**

	<i>Name</i>	<i>Partner/Activity</i>	<i>Date</i>
<b>From:</b>	A. Coveney, S. Pullinger	STFC / JRA1	2016-02-25
<b>Moderated by:</b>	Małgorzata Krakowian	EGI.eu/NA1	
<b>Reviewed by</b>	Luciano Gaido	INFN/PMB	2016-02-09
	Yannick Legré	EGI.eu/NA1	2016-02-13
	Vincenzo Spinoso	INFN/SA1	2016-02-18
<b>Approved by:</b>	AMB and PMB		2016-03-06

**DOCUMENT LOG**

<i>Issue</i>	<i>Date</i>	<i>Comment</i>	<i>Author/Partner</i>
<b>v0.1</b>	2016-01-14	Document creation	A. Coveney / STFC
<b>v0.2</b>	2016-02-01	Revised following internal review	A. Coveney / STFC
<b>v0.3</b>	2016-02-02	Moderated	D. Scardaci / EGI.eu
<b>FINAL</b>	2016-02-25	Revised following external review	A. Coveney / STFC

**TERMINOLOGY**

A complete project glossary is provided at the following page: <http://www.egi.eu/about/glossary/>

## Contents

1	Introduction.....	5
2	Service architecture .....	6
2.1.1	High-Level Service architecture .....	6
2.1.2	Integration and dependencies .....	7
3	Release notes.....	7
4	Result of testing.....	8
5	Future plans.....	8

## Executive summary

This deliverable describes the first EGI-Engage release of the EGI Accounting Repository including the developments made during the first year of the project. The EGI Accounting Repository runs using software from the APEL project, which collects accounting data from sites participating in the EGI and WLCG infrastructures as well as from sites belonging to other Grid organisations that are collaborating with EGI. It generates statistical summaries that are available through the EGI/WLCG Accounting Portal to Users, VO Managers and Site Administrators.

The software has the following high-level architecture: APEL clients can run an APEL parser or use a third-party tool to extract records from a batch system. Clients use the Secure Stomp Messenger (SSM) to send these records via the EGI Message Brokers to the central APEL server. The central server runs a receiving SSM and a “loader” processes the records and loads them into the central repository. A “summariser” process runs once a day to create summaries, which are sent using SSM to the EGI Accounting Portal. There are no changes to this architecture in this release.

The software has the following dependencies: All communication between clients and servers is via the EGI Message Broker network using the APEL SSM package. The central APEL server uses the EGI service registry<sup>1</sup> to get a list of APEL endpoints for authorization purposes. SSM can be configured to get a list of message brokers from the EGI information system (querying a BDII). There are no changes to the dependencies in this release.

This release includes a number of fixes and usage improvements, including changes to support new versions of three batch systems: HTCondor, Univa Grid Engine, and Torque.

The APEL project uses a development process based around GitHub, which includes a semi-automatic testing procedure used to assess the quality of software releases. This release passed all the relevant tests.

The next few features that are planned to be added to the software include a parser for ARC accounting records, a new schema for cloud accounting, and support for long running cloud VMs.

---

<sup>1</sup> <http://goc.egi.eu>

# 1 Introduction

This report documents the first EGI-Engage release of the EGI Accounting Repository including the developments made during the first year of the project. The Accounting Repository runs using software from the APEL project.

APEL is an accounting tool that collects accounting data from sites participating in the EGI and WLCG infrastructures as well as from sites belonging to other Grid organisations that are collaborating with EGI, including OSG, NorduGrid and INFN.

The accounting information is gathered from different sensors into a central accounting repository where it is processed to generate statistical summaries that are available through the EGI/WLCG Accounting Portal.

Statistics are available for view in different detail by Users, VO Managers, Site Administrators and anonymous users according to well-defined access rights.

Table 1 provides a summary of the tool covered in this release.

Table 1 – APEL tool summary

<b>Tool name</b>	APEL
<b>Tool url</b>	<a href="http://apel.github.io/">http://apel.github.io/</a>
<b>Tool wiki page</b>	<a href="https://wiki.egi.eu/wiki/Accounting_Repository">https://wiki.egi.eu/wiki/Accounting_Repository</a>
<b>Description</b>	EGI Core Service – The Accounting Repository collects and stores user accounting records from various services offered by EGI.
<b>Customer of the tool</b>	EGI
<b>User of the service</b>	Site admins
<b>User Documentation</b>	<a href="https://twiki.cern.ch/twiki/bin/view/EMI/EMI3APELClient">https://twiki.cern.ch/twiki/bin/view/EMI/EMI3APELClient</a>
<b>Technical Documentation</b>	<a href="https://twiki.cern.ch/twiki/bin/view/EMI/EMI3APELClient">https://twiki.cern.ch/twiki/bin/view/EMI/EMI3APELClient</a>
<b>Product team</b>	STFC
<b>License</b>	Apache License, Version 2.0
<b>Source code</b>	<a href="https://github.com/apel/apel">https://github.com/apel/apel</a>

The outline of this deliverable is as follows: first we provide a short introduction to the tools provided by the APEL project as part of the accounting repository. Then the high-level architecture of the tool and its components are described, along with the integrations and dependencies it has. Release notes and the results of testing for this release are provided. Finally, a selection of future developments is shown.

## 2 Service architecture

### 2.1.1 High-Level Service architecture

Figure 1 shows how the APEL client, central APEL server and EGI Accounting Portals interact.

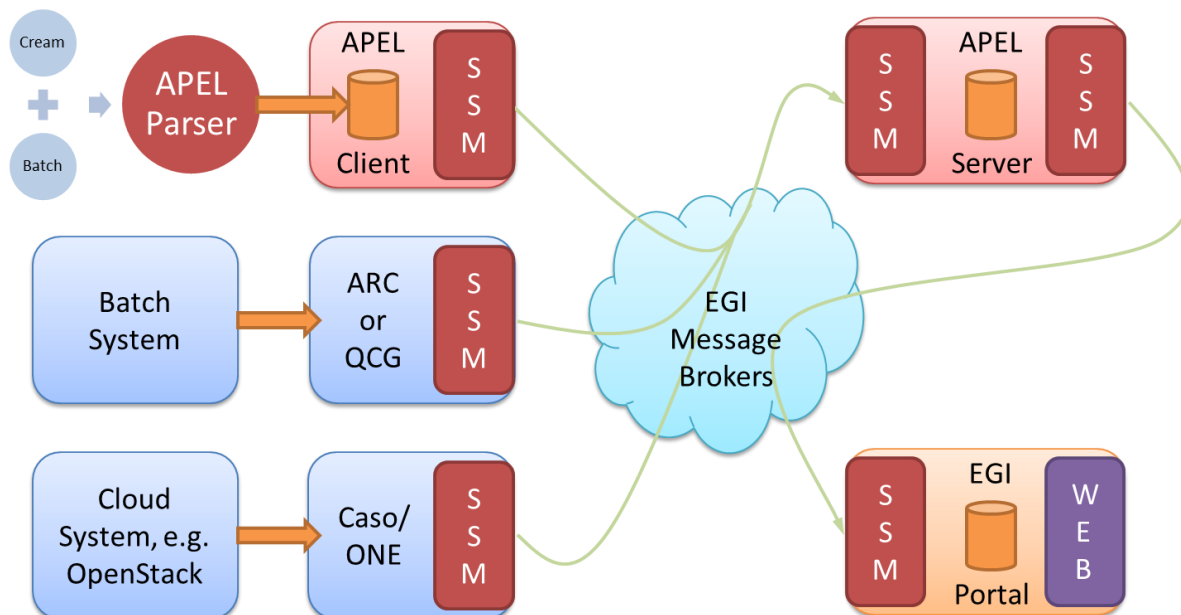


Figure 1 - APEL components and their interactions. Components in red are provided by the APEL project.

1. APEL clients can run an APEL parser to extract data from a batch system and place it in their client database, or they can use third-party tools to extract batch or cloud data. This data is then unloaded into a message format suitable for transmission.
2. APEL clients run a sending Secure Stomp Messenger<sup>2</sup> (SSM) to send these messages containing records via the EGI Message Brokers the central APEL server. The messages can contain either Job Records or Summary records. This is configurable in the APEL client.
3. The central APEL server runs an instance of the SSM, which receives these messages and a “loader” processes the records in the messages and loads them into a MySQL database.
4. A “summariser” process runs to create summaries of any Job Records received and load them in a “SuperSummaries” table along with any Summary records. This summariser runs as a cron job approximately once a day.
5. A database “unloader” process unloads the summary records into the message format to be sent on by the sending SSM via the EGI Message Brokers to the EGI Accounting Portal.

There are no changes to the service architecture in this release.

<sup>2</sup> <https://github.com/apel/ssm>

### 2.1.2 Integration and dependencies

All communication between clients and servers is via the EGI Message Broker network using the APEL SSM package. The SSM can be configured to send or receive messages. Where the messages are destined for is controlled by the queue, which is set in the SSM configuration.

The central APEL server uses the EGI service registry (GOCD<sup>3</sup>) to get a list of APEL endpoints so that only data from endpoints correctly defined in GOCD is processed.

SSM can be configured to get a list of message brokers from the EGI information system (querying a BDII) or it can be pointed directly at a message broker.

There are no changes to the dependencies in this release.

## 3 Release notes

These are the changes included in this release of the APEL software, version 1.5.1, since the start of EGI-Engage.

- Added support for Torque 5.1.2 time duration format.
- Changed `dirq`<sup>4</sup> call to use absolute path to support versions of `dirq`  $\geq$  1.7.
- Fixed a crash when the storage accounting record loader encounters a valid XML file with no records in it.
- Added sorting of accounting logs before parsing which makes reading through the parser log easier, especially if files use the YYYYMMDD date format.
- Added the first version of a basic HTCondor<sup>5</sup> parser that can be customised by sites if they require additional functionality.
- Fixed the server schema to correctly aggregate super summaries when viewed as normalised super summaries.
- Changed defaults so that parallel jobs are reported for new installations.
- Corrected year calculation in the `migrate_apel.py` script.
- Added an option to the batch configuration section to allow for the millisecond timestamps introduced in Univa Grid Engine<sup>6</sup> version 8.2.0.
- Improved logging output to make it more useful to APEL servers when tracing messages.

---

<sup>3</sup> <http://goc.egi.eu/>

<sup>4</sup> <https://pypi.python.org/pypi/dirq>

<sup>5</sup> <https://research.cs.wisc.edu/htcondor/>

<sup>6</sup> <http://www.univa.com/products/>

## 4 Result of testing

The APEL project uses a development process based around GitHub, which includes a semi-automatic testing procedure used to assess the quality of software releases.

For details of the testing procedure used, see the APEL Development Process document<sup>7</sup>. Table 2 summarises the results of testing this release.

Table 2 - APEL 1.5.1 testing results

	<i>Result</i>	<i>Link</i>
<b>Unit tests</b>	All unit tests passed	<a href="https://travis-ci.org/apel/apel/builds/101811397">https://travis-ci.org/apel/apel/builds/101811397</a>
<b>Coverage</b>	Coverage increased by 9.1%	<a href="https://coveralls.io/builds/4697410">https://coveralls.io/builds/4697410</a>
<b>Code quality</b>	Score increased by 1%	<a href="https://landscape.io/github/apel/apel/109">https://landscape.io/github/apel/apel/109</a>

## 5 Future plans

The features that are planned to be added to the software for the second release of the Accounting Repository (D3.12) are described below. They map to tasks in the EGI Engage Accounting Repository roadmap<sup>8</sup>.

- Task 3.1.1: A new schema for cloud accounting based on the EGI Federated Cloud Accounting Usage Record Version 0.4<sup>9</sup>. This work has been completed for the accounting repository, but won't be rolled out until external cloud developers have tested sufficiently against the test instance of the service.
- Task 3.1.3: Support for accounting for long running cloud VMs and assigning their usage to intermediate months.
- Task 3.1.4: A parser for ARC<sup>10</sup> accounting records.
- Task 3.1.6: Support for using the OGF Usage Record version 2 for storage accounting records.

<sup>7</sup> <https://documents.egi.eu/document/2739>

<sup>8</sup> [https://wiki.egi.eu/wiki/TASK\\_JRA1.3\\_Accounting#Accounting\\_Repository](https://wiki.egi.eu/wiki/TASK_JRA1.3_Accounting#Accounting_Repository)

<sup>9</sup> [https://wiki.egi.eu/wiki/Federated\\_Cloud\\_Accounting#Cloud\\_Accounting\\_Usage\\_Record](https://wiki.egi.eu/wiki/Federated_Cloud_Accounting#Cloud_Accounting_Usage_Record)

<sup>10</sup> <http://www.nordugrid.org/arc/>