

The Execution Environment Service

Mischa Sallé

Oscar Koeroo & Aram Verstegen

Nikhef, Amsterdam

- LCAS - meant to do authorisation: OK or FAIL.

- LCAS - meant to do authorisation: OK or FAIL.
- LCMAPS - can do a user-mapping, authorisation decision etc.

- LCAS - meant to do authorisation: OK or FAIL.
- LCMAPS - can do a user-mapping, authorisation decision etc.
- SCAS - does the decisions centrally, uses LCMAPS as backbone.

- LCAS - meant to do authorisation: OK or FAIL.
- LCMAPS - can do a user-mapping, authorisation decision etc.
- SCAS - does the decisions centrally, uses LCMAPS as backbone.

Widely used, also in OSG, well tested, robust...

- LCAS - meant to do authorisation: OK or FAIL.
- LCMAPS - can do a user-mapping, authorisation decision etc.
- SCAS - does the decisions centrally, uses LCMAPS as backbone.

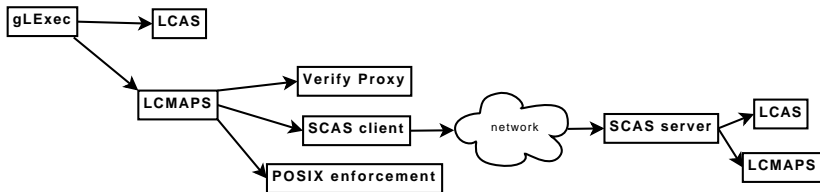
Widely used, also in OSG, well tested, robust...

...but a bit confusing and not so extendible

- LCAS - meant to do authorisation: OK or FAIL.
- LCMAPS - can do a user-mapping, authorisation decision etc.
- SCAS - does the decisions centrally, uses LCMAPS as backbone.

Widely used, also in OSG, well tested, robust...

...but a bit confusing and not so extendible

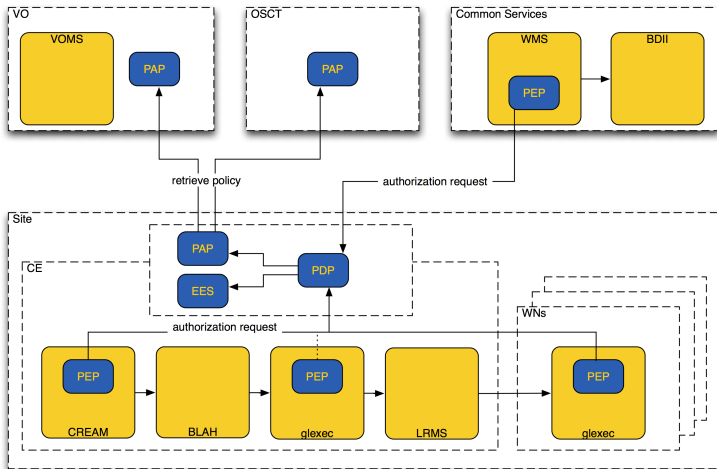


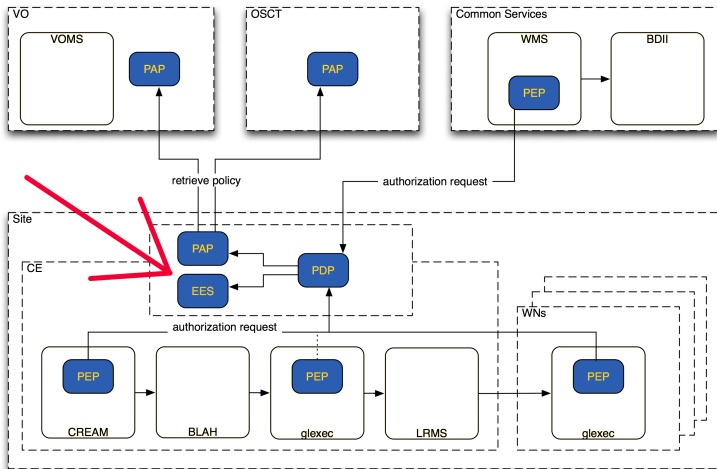
- Based on SAML2-XACML2 profile...

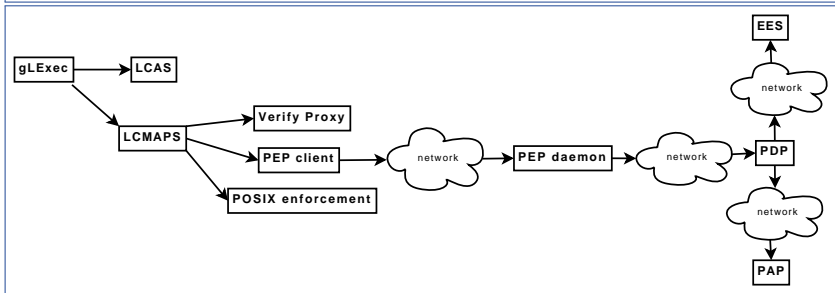
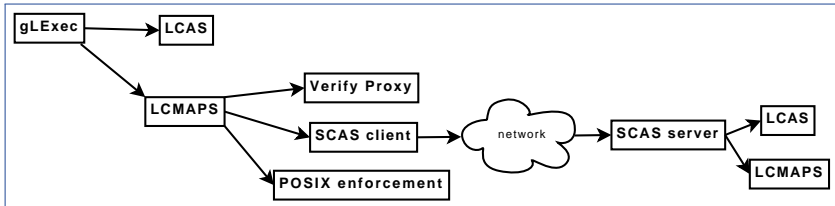
- Based on SAML2-XACML2 profile...
- ... aims at interoperability with OSG

- Based on SAML2-XACML2 profile...
- ... aims at interoperability with OSG
- Step-by-step transition/introduction

- Based on SAML2-XACML2 profile...
 - ... aims at interoperability with OSG
 - Step-by-step transition/introduction
- *needs to be compatible with existing tools!*







The EES versus the rest of Argus

- Argus P?P components are high-level, mostly oblivious of OS
- Someone has to do the dirty work
- ...

The EES versus the rest of Argus

- Argus P?P components are high-level, mostly oblivious of OS
- Someone has to do the dirty work
- EES!

The EES versus the rest of Argus

- Argus P?P components are high-level, mostly oblivious of OS
- Someone has to do the dirty work
- EES!
- EES ensures that
the boundary conditions of the PDP decision can be fulfilled

The EES versus the rest of Argus

- Argus P?P components are high-level, mostly oblivious of OS
- Someone has to do the dirty work
- EES!
- EES ensures that
the boundary conditions of the PDP decision can be fulfilled
- Examples:
 - Creation of new user accounts via LDAP
 - Finding a proper poolaccount
 - Setting up a Torque/Maui jobqueue
 - Installing needed libraries, setting environment
 - Instructing installation, configuration, staging etc. of a VM by OpenNebula/Haizea

EES requirements:

- should talk SAML2-XACML2

EES requirements:

- should talk SAML2-XACML2
- easy access to OS API

EES requirements:

- should talk SAML2-XACML2
- easy access to OS API
- extendible / pluggable

EES requirements:

- should talk SAML2-XACML2
- easy access to OS API
- extendible / pluggable
- fast

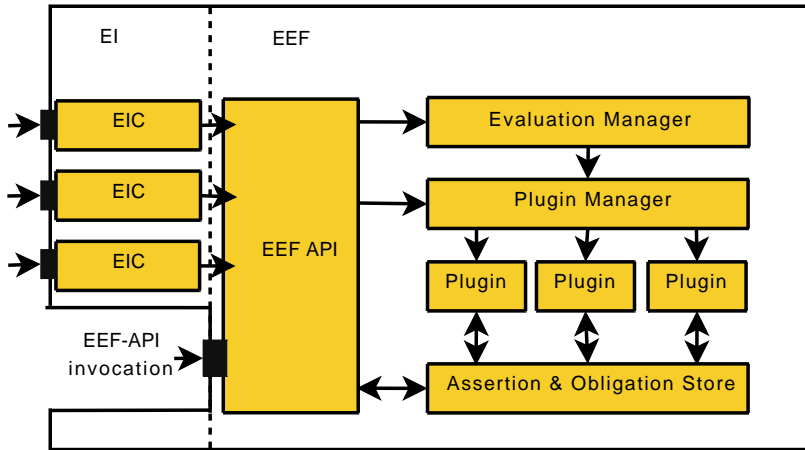
EES requirements:

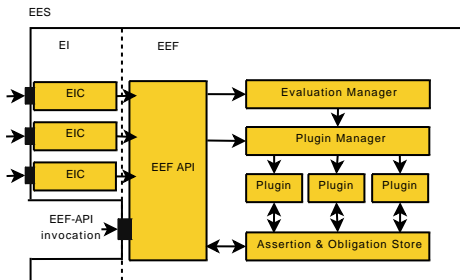
- should talk SAML2-XACML2
- easy access to OS API
- extendible / pluggable
- fast
- portability

EES requirements:

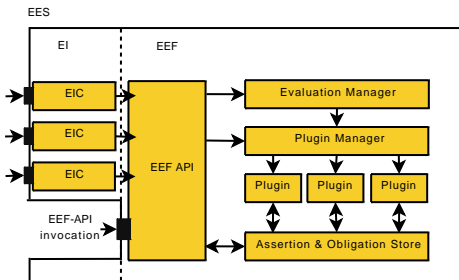
- should talk SAML2-XACML2
- easy access to OS API
- extendible / pluggable
- fast
- portability
- support for scripting languages

EES

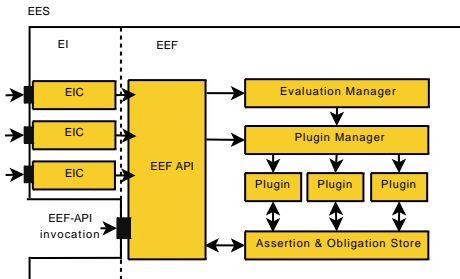




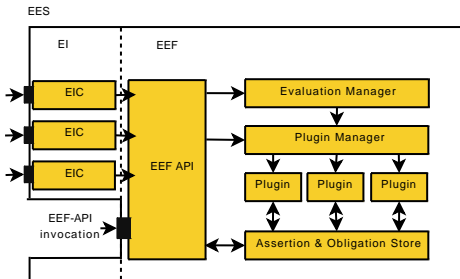
- EES Interface Components: *drivers, talk to EEF API*



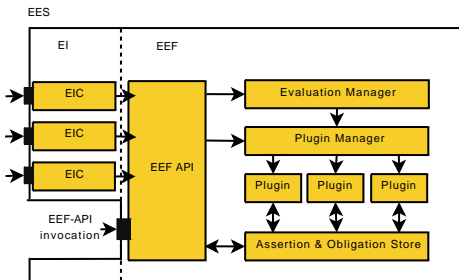
- EES Interface Components: *drivers, talk to EEF API*
- EES Framework API



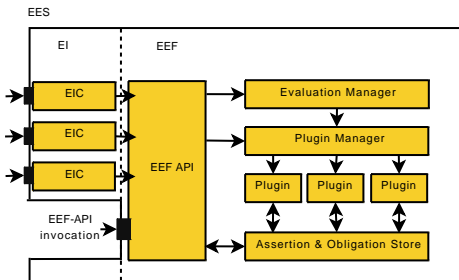
- EES Interface Components: *drivers, talk to EEF API*
- EES Framework API
- Evaluation Manager: *drives the 'business logic'*



- EES Interface Components: *drivers, talk to EEF API*
- EES Framework API
- Evaluation Manager: *drives the 'business logic'*
- Plugin Manager: *executes the 'business logic'*



- EES Interface Components: *drivers, talk to EEF API*
- EES Framework API
- Evaluation Manager: *drives the 'business logic'*
- Plugin Manager: *executes the 'business logic'*
- Plugins: *do the really dirty work*



- EES Interface Components: *drivers, talk to EEF API*
- EES Framework API
- Evaluation Manager: *drives the 'business logic'*
- Plugin Manager: *executes the 'business logic'*
- Plugins: *do the really dirty work*
- AOS: *Common memory store for the plugins and the EEF*

- Argus for the time being:
user-mapping in a PEPd plugin (obligation handler)
could have been done using SCAS

- Argus for the time being:
user-mapping in a PEPd plugin (obligation handler)
could have been done using SCAS
- Draft design document:
<https://edms.cern.ch/document/1018216/1>
Feedback → final version

- Argus for the time being:
user-mapping in a PEPd plugin (obligation handler)
could have been done using SCAS
- Draft design document:
<https://edms.cern.ch/document/1018216/1>
Feedback → final version
- C-code is being written

- Argus for the time being:
user-mapping in a PEPd plugin (obligation handler)
could have been done using SCAS
- Draft design document:
<https://edms.cern.ch/document/1018216/1>
Feedback → final version
- C-code is being written
- Using experience with LCAS, LCMAPS and SCAS (not code)

- Argus for the time being:
user-mapping in a PEPd plugin (obligation handler)
could have been done using SCAS
- Draft design document:
<https://edms.cern.ch/document/1018216/1>
Feedback → final version
- C-code is being written
- Using experience with LCAS, LCMAPS and SCAS (not code)
- Investigating use cases: which (type of) plugins: e.g. Perl
and/or python interpreter

- EES Design Document
<https://edms.cern.ch/document/1018216/1>
- Authorization Service Design (29/08/2008)
<https://edms.cern.ch/document/944192/1>
- MJRA1.4: gLite Security Architecture (30/01/2009)
<https://edms.cern.ch/document/935451/2>
- Argus Twiki pages
<https://twiki.cern.ch/twiki/bin/view/EGEE/AuthorizationFramework>
- XACML Attribute and Obligation Profile for Authorization Interoperability in Grids (29/08/2008)
<https://edms.cern.ch/document/929867/2>

?

- C:
 - Excellent OS API, included in *standard* library
 - Available on every UNIX (and Windows) platform
 - Well portable, especially if POSIX
 - Fast
 - Standalone exec., difficulties mainly at compile time, *not* at runtime

- C:
 - Excellent OS API, included in *standard* library
 - Available on every UNIX (and Windows) platform
 - Well portable, especially if POSIX
 - Fast
 - Standalone exec., difficulties mainly at compile time, *not* at runtime
- Java:
 - Rest of Argus in Java, but is that an argument?
 - Relies on external libraries/tools for many OS interactions (e.g. JBoss)
 - Portable, but problems with JDK versions: especially runtime problems.