



Enabling Grids for E-scienceE

AuthZ Interoperability: INFN view

Speaker *Alberto Forti*
Location *Budapest*

www.eu-egee.org



- **Authorization in gLite**
- **A unified approach for AuthZ**
- **Authorization Interoperability and G-PBox standardization**
- **Conclusions**

- **Non homogeneous:**

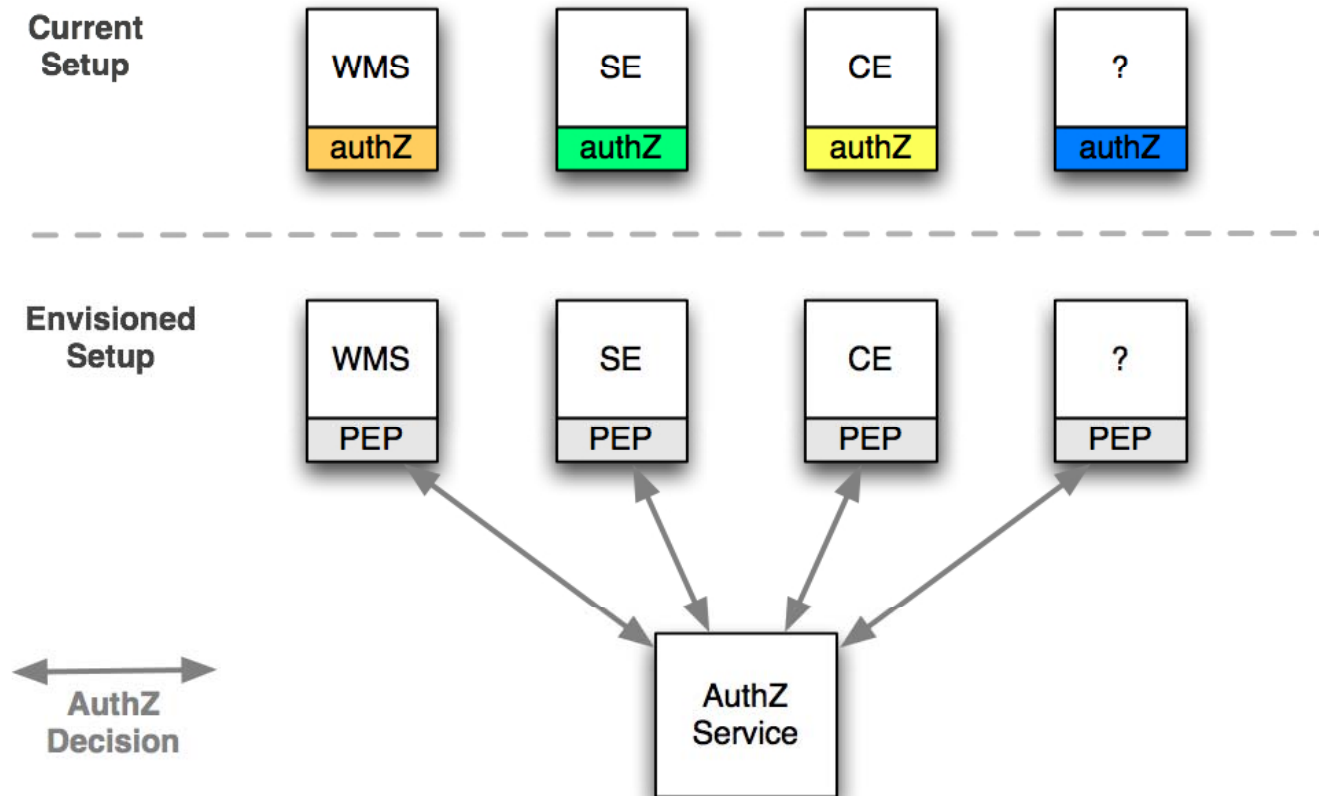
- Practically every component has its own dedicated mechanism to deal with AuthZ issues:
 - DM components -> ACLs
 - CEs -> Pluggable authZ (gridmapfile, etc.)
 - RGMA -> None
 - WMS -> Whitelists
- The only common thing is that (most) of them leverage VOMS groups and roles (FQANs).
 - However there is no common agreement on how FQANs should be used.
- Means that for every component a new set of rules and a new set of configurations must be learnt.

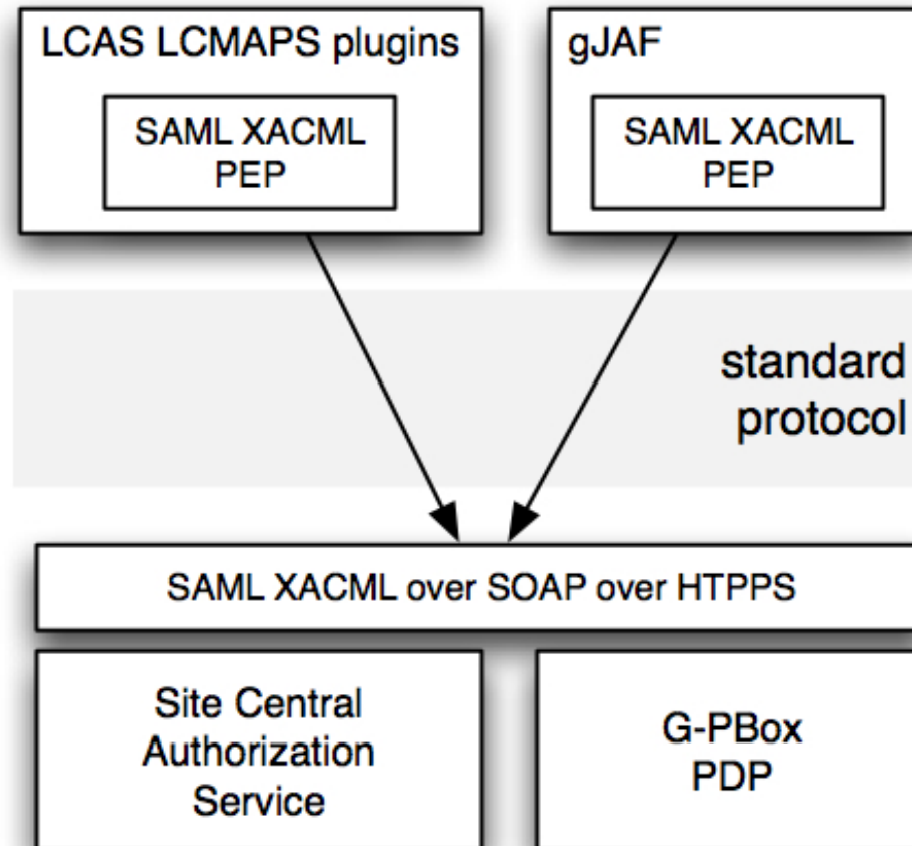
- **Untraceable:**
 - It is difficult to trace the set of AuthZ decisions that regulates resource access
 - Who authorized (or *not* authorized) this job?
 - Where are the related policy configurations?
 - What that configuration means?

- **Uncoordinated:**
 - Different sites may only coordinate “by hand”:
 - Explicitly modifying their own policies to match grid mandated requirements
 - Time consuming and inherently fragile

- **A unified approach for authorization would allow:**
 - Homogeneity
 - Traceability
 - Manegability
- **A unified approach needs a policy language expressive enough to cover possible authorization scenarios:**
 - Computing element
 - Storage element
 - High-level services (e.g. WMS)
 - ...
- **---> XACML!**

- A unified approach allows the use of a single tool to take authorization decisions for the different scenarios

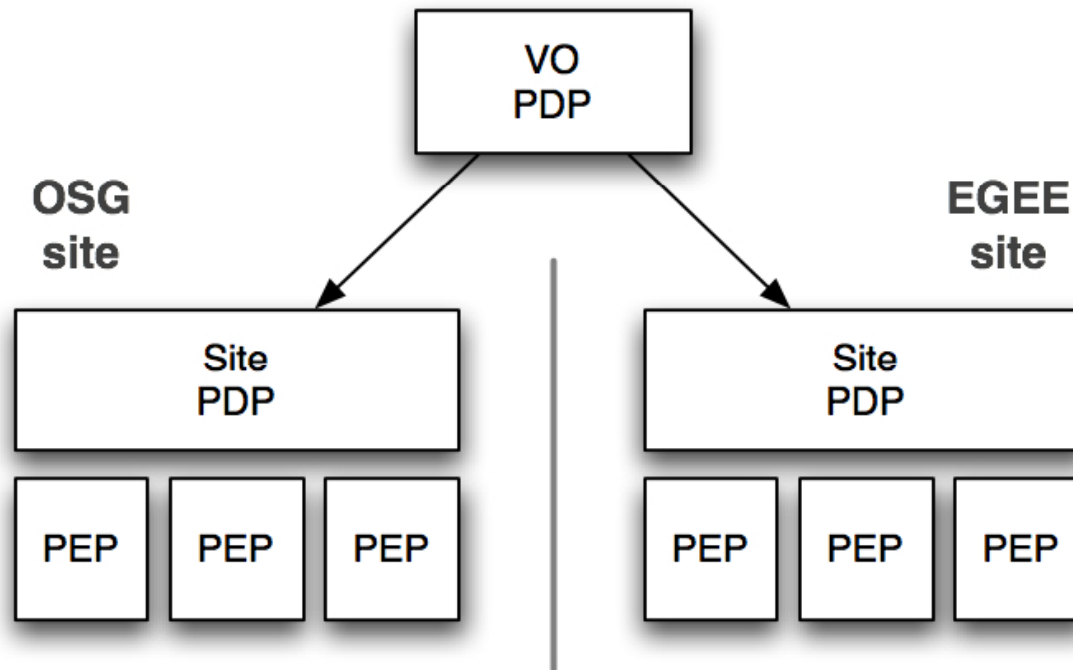




- **There is an ongoing effort for having interoperable authorization services among EGEE, OSG and GT**
- **For what concern the G-PBox team, the scope is wider**
 - Implementing an OGF specification to have interoperable authorization services
 - Reach agreements on what else is needed for OSG and EGEE to interoperate
 - Namespaces (Obligations, Attributes)
- **G-Pbox is going to implement the SAML V2.0 Profile for XACML as it will be agreed**
 - But we are analyzing the better way to blunt the WS overhead
 - There are requirements for calls to a PDP to be very fast

- **Having a common implementation is not the main point, the interface is**
 - Developers should be free to choose whatever tool they like to implement a service
 - But we understand the urge under the common implementation, and we are helping on that
- **What's on the table right now?**
 - GT code (standard compliance and fragility issues)
 - OpenSAML 2 on its way out
 - We are using it in VOMS
 - It's XML we are talking about, there are plenty of tools one can use
 - G-Pbox already implements the XACML layer, we don't need the amount of work that other services need
 - We just need to add the SAML and WS layer

- **New issue:**
 - Coordination of policies among different Grids



- **Are there other PDPs willing to use the XACML language to take authorization decisions?**
- **Are there other PDPs willing to devise new complex languages to define general purpose policies?**
- **Stand up now, because it would be duplicated work!**
- **Interoperability**
 - Policy coordination among different Grids
 - Performance analysis of the communication layer

- We have implemented a solution that tries to overcome the above issues
 - We have a service that unwraps requests and forward them
 - Deployed on Tomcat
 - Using XFire
 - Replaceable if another choice will be made
 - Current XACML interface still available
 - PEPs may choose what to put forward, interoperability or performance
 - Small difference on the client side, the core of the message is the same

