



SVOPME



A Scalable Virtual Organization Privileges Management Environment

Nanbor Wang <nanbor@txcorp.com>

Gabriele Garzoglio <garzoglio@fnal.gov>

Balamurali Ananthan <bala@txcorp.com>

Steven Timm <timm@fnal.gov>

Tanya Levshina <levshin@fnal.gov>

Tech-X Corporation
Fermi National Accelerator Laboratory

ISGC 2010, Taipei, Taiwan

March 11, 2010

**Funded by US DOE OASCR
Grant #DE-FG02-07ER84733**





- **Project overview**
 - What SVOPME tries to address
- **Architecture and implementations**
- **Outlook and planning**

What are VO Privileges?



Virtual Organizations:

- VOs use resources
- VOs wish to define usage policies for various resources for different users within the VOs
 - Example 1: Production team members submit jobs with higher priority
 - Example 2: Software team members can write to disk area for software installations
- VOs define user privileges at different resources to comply with the expressed usage policies
- However, VOs do not manage/configure all Grid sites

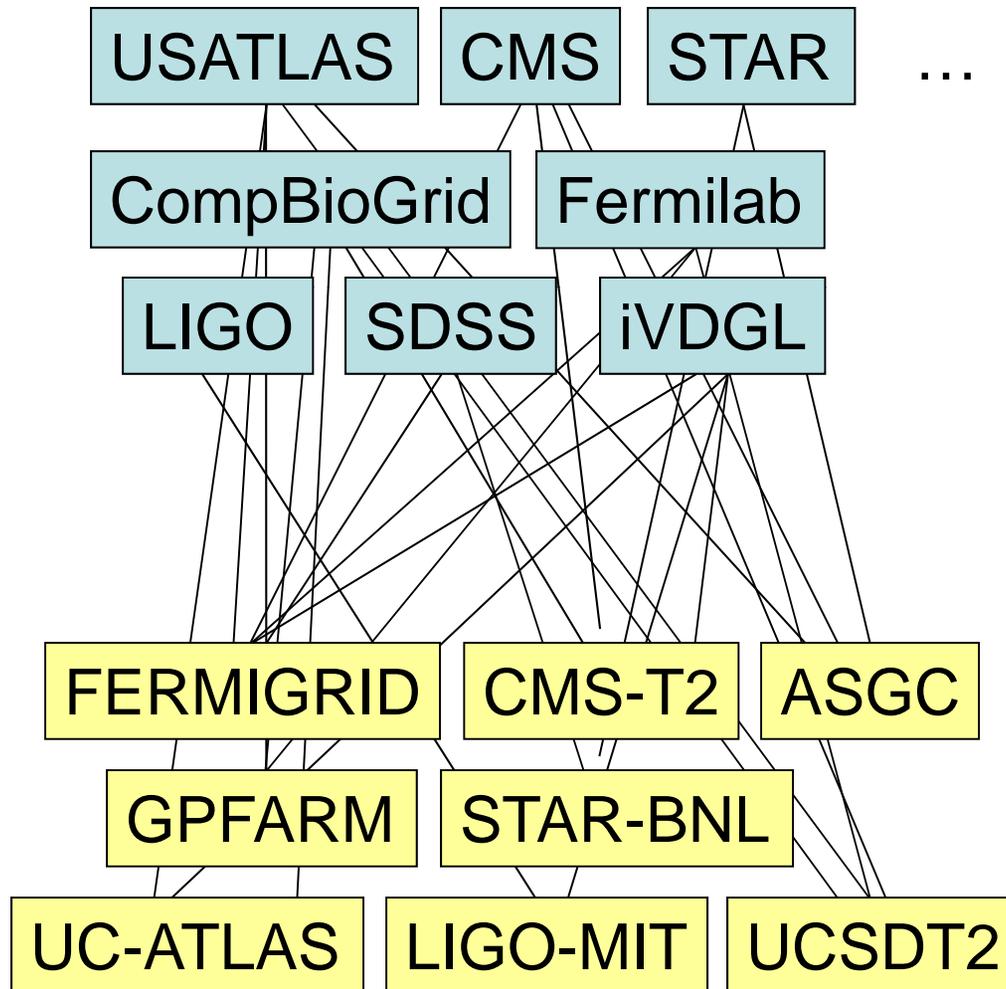
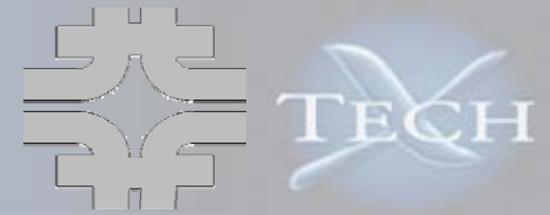
Grid Sites:

- Grid sites provide resources
- Grid sites may want to provide different services to different VOs
 - Example 3: site X has a special agreement with VO Y; therefore, jobs from VO Y might have higher priority than others
- Grid sites help VOs to enforce their usage policies by managing user privileges
- Grid sites don't define VOs' usage policies

**Site and VO Challenge: Enforcing heterogeneous VO privileges on multiple Grid sites to provide uniform VO Policies across the Grid
(ad hoc solution: verbal communication)**

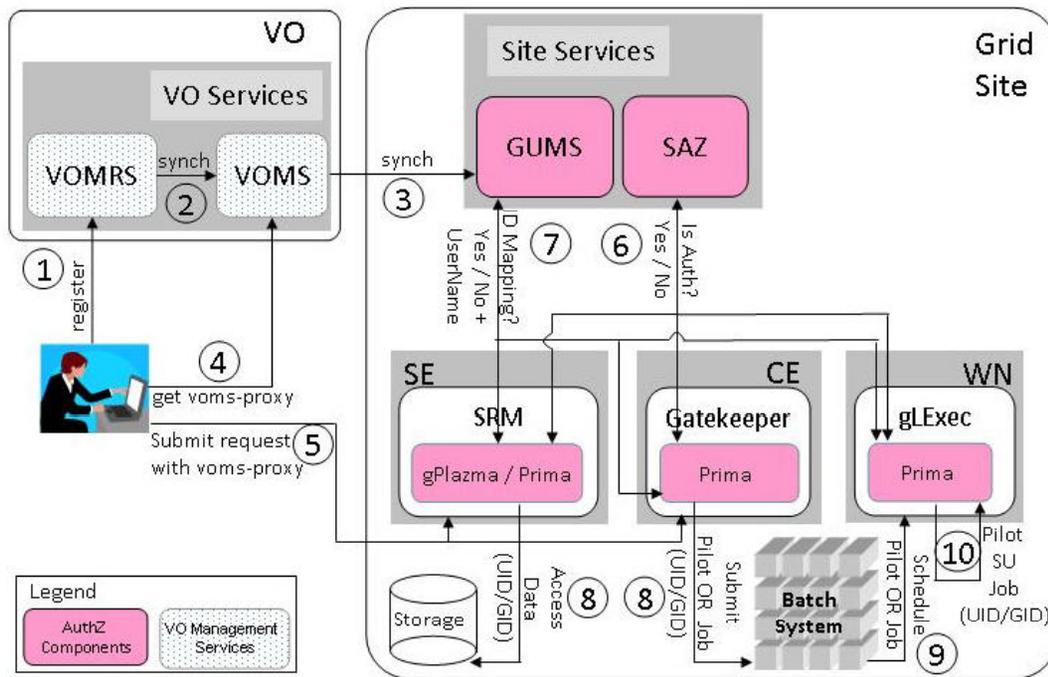
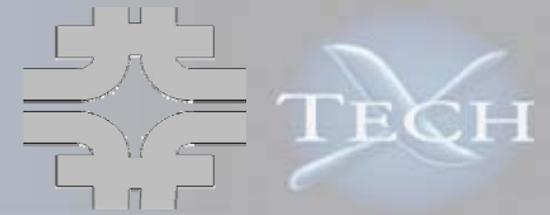
Motivations of SVOPME

Address scalability



- With the growth in Grid usage, both the numbers of **VOs** and **Grid-sites** increase
- Serious scalability problems in propagating VO privilege policies
- **SVOPME:**
 - Provide the tools and infrastructure to help
 - VOs express their policies
 - Sites support a VO
 - Reuse proven administrative solutions – we adopt common system configuration patterns currently in use in major grid sites

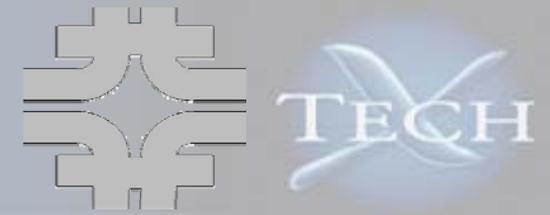
Modern User Privilege Management



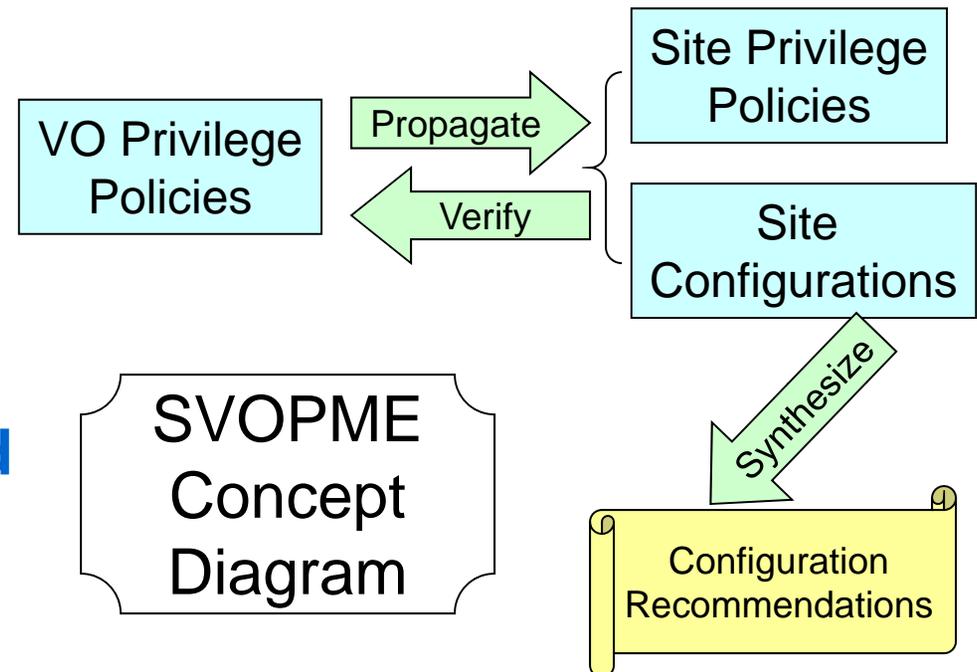
The OSG Authorization Infrastructure

- Moving away from the use of gridmap files to VOMS/GUMS role-based privilege management
 - Eliminate the need for multiple user certificates
 - Similar trend can be observed in EGEE (LCAS/LCMAPS + SCAS and VOMS)
- Managing requests priority for both SE and CE

SVOPME Helps VO's Propagate Privilege Policies to Grid Sites

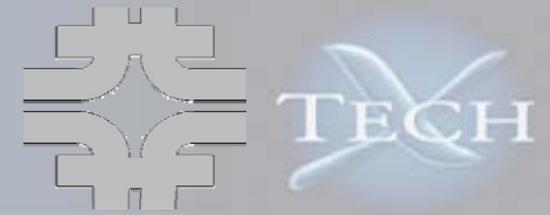


- **SVOPME aims to replace the verbal interaction between VO and site admin's with automated workflows**
- **VO's intended privilege policies are clearly defined**
 - Using eXtensible Access Control Markup Language (XACML)
 - No ambiguity
 - Allow programmatic verification of policies
 - XACML is also used by AuthZ Interoperability project



- **Site's actual policies can be verified**
- **SVOPME provides recommendations to site configurations for better VO supports**

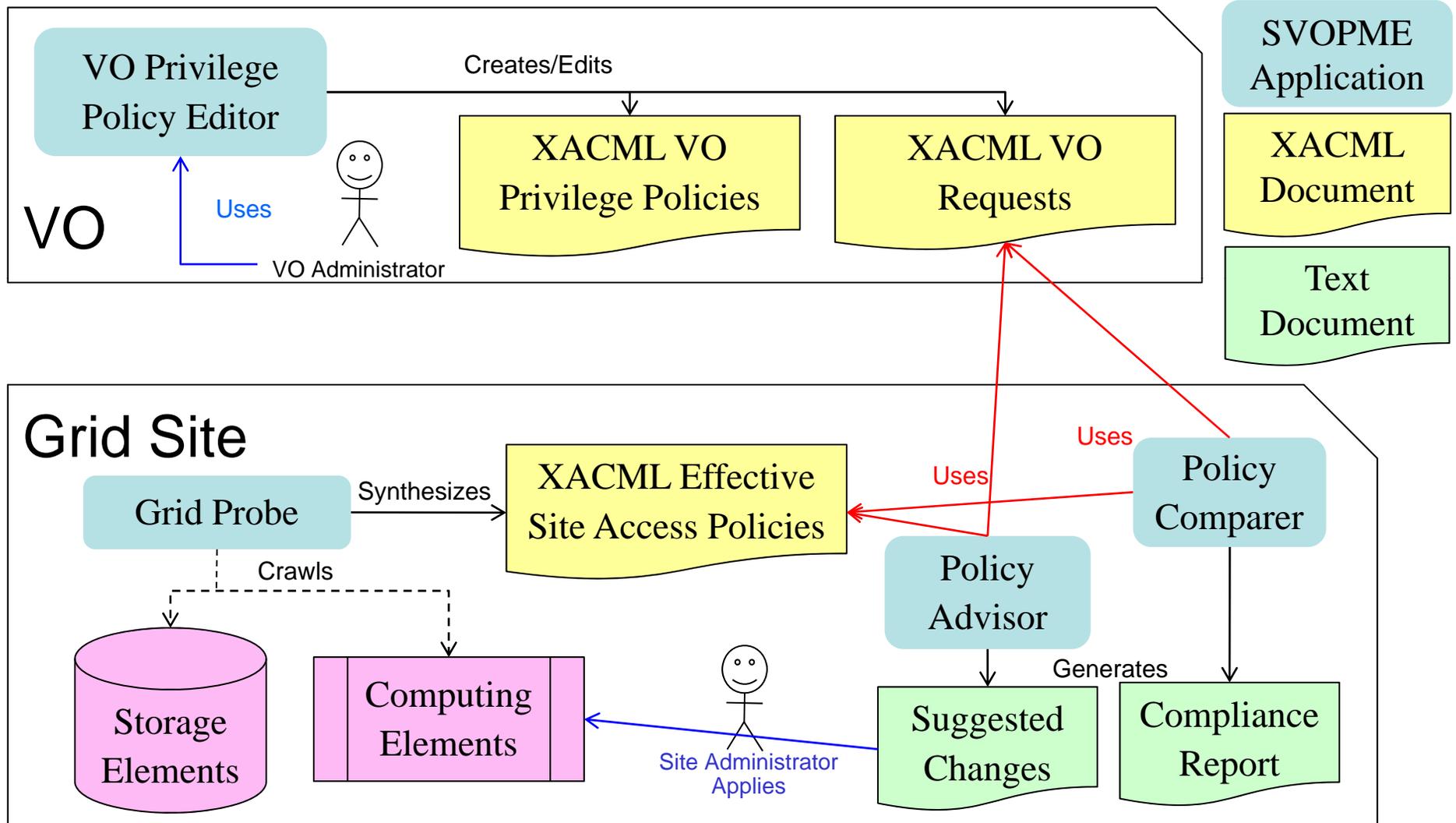
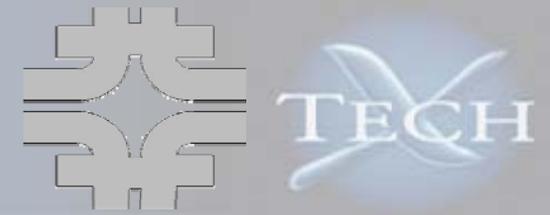
Survey of Resources and Policies Managed on the Grid



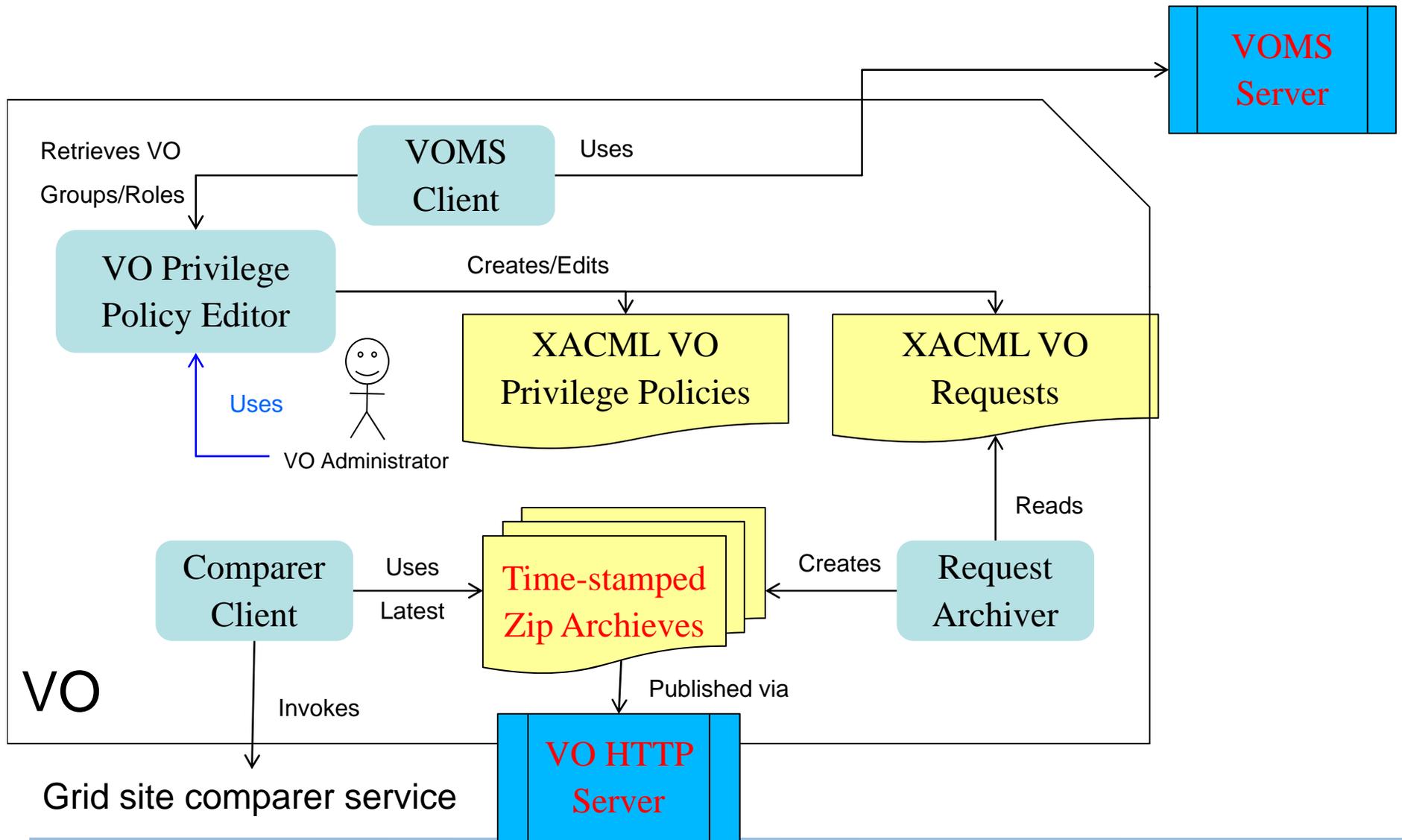
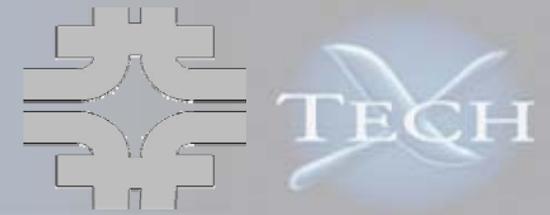
- **Resources**
 - OS protection (account types: group or pool)
 - Batch system
 - File system
 - External storage (SRM/dCache)
 - Network access (inbound/outbound)
 - Edge services
- **Policies expressed by the Site**
 - Timed availability (execution time slots for certain VO users)
- **Policies expressed by both**
 - **Disk quota**
 - File retention period
 - Network (inbound/outbound) access control
- **Policies expressed by the VO**
 - **Account type**
 - **Intra-VO relative priority in batch system**
 - **Directory access (group privacy) permissions**
 - **Job pre-emption (Consecutive execution period)**
 - **Suspension/resumption of jobs**
 - **User file privacy**
 - **Two roles to share the same GID**
 - Repeat execution (Allowing restart or not in batch system) ?

Highlighted policies are supported

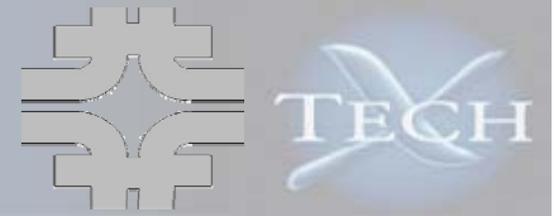
SVOPME Architecture



SVOPME VO Tools



XACML VO Policy Editor (Domain Specific)



- **XACML is**
 - An XML-based language for specifying access control policies
 - Suitable for machine processing (deciding permissions on actions)
 - Way too generic to reason an arbitrary policy
- **SVOPME**
 - Takes a domain specific approach
 - Defines a set of “profiles” of meta-policies
 - Each meta-policy defines a type of policy VO can define
 - For example: Account Mapping Policy - Group X should run with pool account
- **The VOMS client obtains information about all the Group/Role and the number of users from the VOMS server on VO editor’s behave.**
- **Support for new policy types can be added as “Policy Template” plug-in’s**
- **VO Administrator can create and edit a set of policies**
- **Reject contradicting policies – (will leverage Model checking Grid Policies by JeeHyun)**

VO Policy Editor Screenshot



The screenshot shows the SVOPME VO Policies Editor window. The title bar reads "SVOPME VO Policies Editor". The menu bar includes "File", "Edit", and "Help".

Existing VO Policies

- VO Policies
 - Account Mapping (4)
 - Priority (3)
 - AB_0.xacml
 - AB_2.xacml (selected)
 - AB_1.xacml
 - Job Runtime (2)
 - Disk Quota (0)
 - Job Suspension (0)
 - Files Privacy (0)
 - Shared Unix Group Account (0)
 - Package Installation (0)

Step 2 of 3 - Priority Policy

Policy Description
FQAN A should have higher priority than FQAN B in the batch system

User Input

Policy Id: AB_2 (Ex: PriorityPolicy_1)

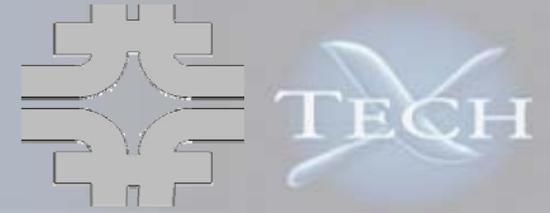
FQAN A: /dzero/users/Role=D0Production (20 Users) Highest Priority

FQAN B: /dzero/services (40 Users)

Buttons: Save, Cancel, Delete

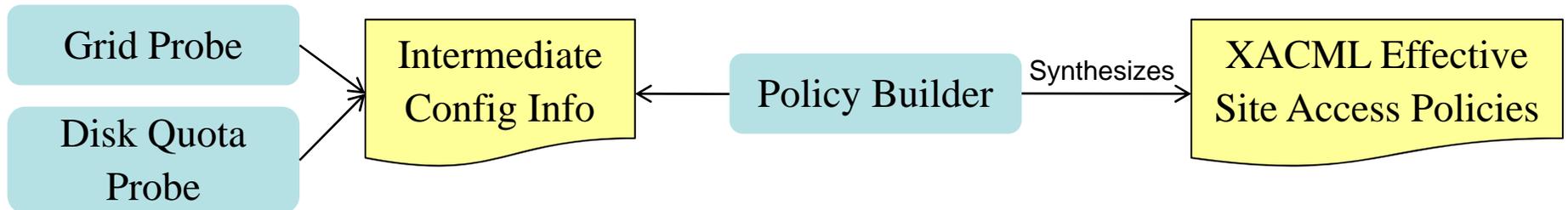
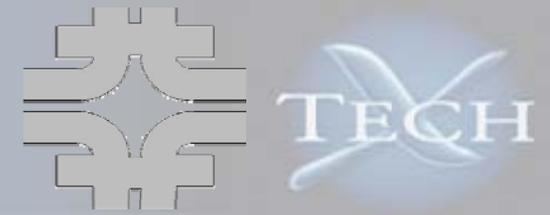
Console Messages
Compile a new Policy from File->New VO Policy
Choose a policy template

VO Policy Data Management



- **The Editor stores the policies and verification requests under predefined directories**
- **Request Archiver collects and zips up verification requests into time-stamped zip files**
 - Can be used by sites to examine their compliance
 - Time-stamped request zip archives are made available to site via a simple web page
 - Sites can scan the page and determine the latest version
- **VO admins and users can use Comparer Client to contact and check a site's support to VO policies**

Mechanism for Synthesizing Grid Site Privilege Policies



- **“Grid Probe” in a nutshell**

- Policy building and configuration crawling functions are separated
- Depending on the target privilege, different info is necessary: there are multiple crawling executables
- Invoked by different cron tasks with diff privileges
- Dump the info as simple text files at a specific directory
- Allow site-specific probes

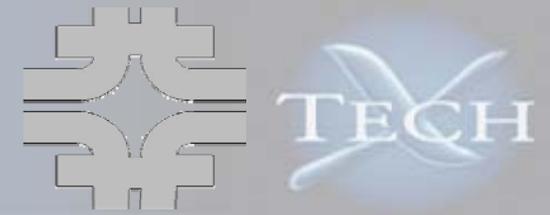
- **Configuration checked**

- Condor/GUMS config
- Disk quota/directory permissions

- **Policy Builder**

- Parses the intermediate configuration info
- Synthesizes the effective privilege policies of a site into XACML policies

Analyzing Site Configurations

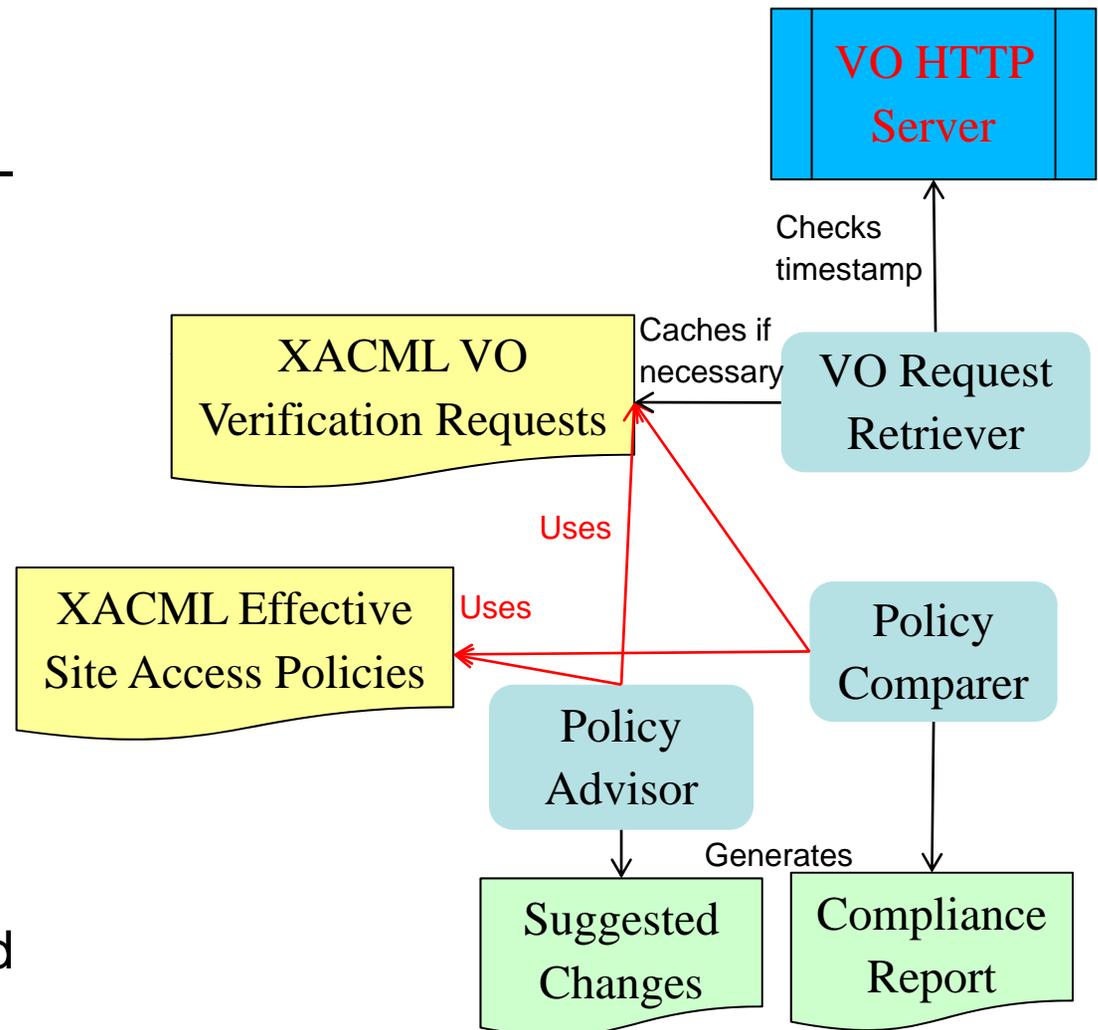


• VO Request Retriever

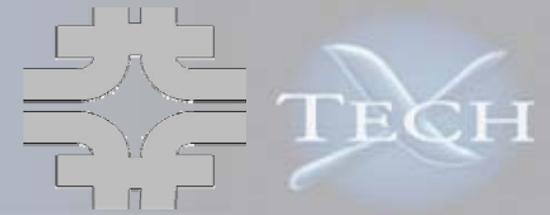
- Checks if the local VO verification requests is up-to-date
- Cache the new verification requests if needed

• Policy Comparer and Advisor

- Test compliance by testing the verification requests one-by-one
- Since all requests and policies are based on our XACML profiles, reports and advises can be derived



VO/Grid Policies Comparer



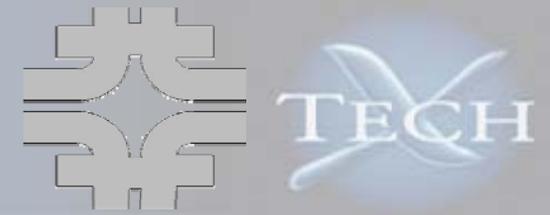
■ Example output:

```
[java] VO/Grid Grid Accounts Policy Comparison
[java] -----
[java] /TECHX/Role=User is mapped to 1 account(s) on the Grid site. Passed!
[java] No Account Mapping Policies for /TECHX/VISITORS were found on the
Grid site.
```

■ Policy Comparer Grid Service

- Allow VO users to check privilege policy compliance at a site
- Instead of cached verification requests, users supply a list of verification requests related to policies of interests
- SVOPME provides a policy comparer client as part of the VO tools
- Currently only provide text reports – should provide a mechanism for further automate the information gathering

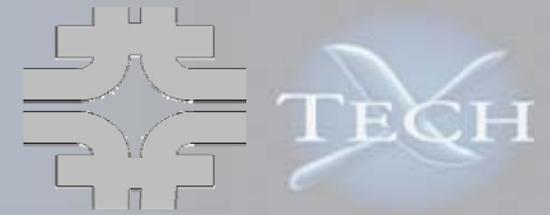
VO/Grid Policies Advisor



- Provide advice for the **Grid site administrator** on what amendments need to be done on the Site; such that the Grid site complies with the VO policies
- Example output:
 - VO requested 3 accounts for VISITORS role via VO policies
 - Site-policies derived from GUMS do not match

```
[java] VO/Grid Grid Accounts Policy Advices
[java] -----
[java] No matching Grid Accounts Policy was found for /TECHX/VISITORS
on the Grid site. Create a mapping in GUMS config such that
/TECHX/VISITORS be mapped to at least 3 account(s)
[java] TECHX/Role=VO-Admin mapped to 1 account(s) (techxVOadmin) on
the Grid site, Needs to be mapped to atleast 3 accounts.
```

Advantages for VOs and Sites



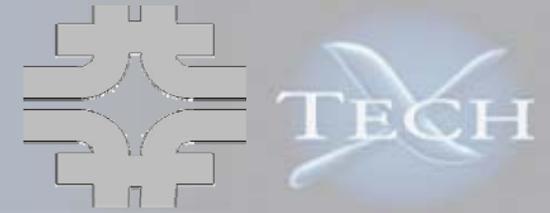
• VO's

- No need to run ad-hoc jobs to figure out what policies are enforced and what not
- Provides templates to define commonly used policies
- Automates most of the communication with Sites that support the VO
- Provides the basis for the negotiation of privileges at sites that provide opportunistic access

• Sites

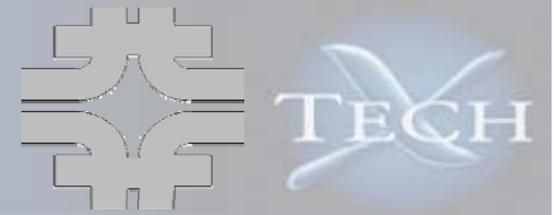
- Sites can advertise and prove that a VO is supported
- Sites that want to support a VO have a semi-automated mechanism to enforce the VO policies
- Privilege enforcement remains responsibility of the Site, informed by formal VO policy assertions

Experiments on FermiGrid's Integrated TestBed



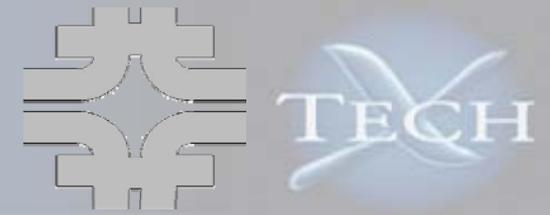
- Using “Dzero” and “Engage” VO’s privileges as a real-world examples
- Validation requests are copied over to the site (FGITB) using the “Retriever” tool
- Two different probes run with different privileges
- “Engage” VO will continue to expand and incorporate other smaller sub-VO’s
- Was able to detect several anomalies
 - Enhanced disk quota probes – multiple filesystems
 - Re-wrote quota/filesystem probe to use python – easier for admins to examine
 - Detected one missing account mapping
 - Legacy pool account configurations
- Separating probes allows easy adaption to site with unconventional configurations

Extending Meta-Policies



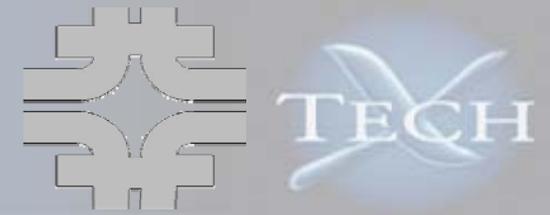
- **Steps to extend SVOPME to support new privilege policy profiles**
 - Define what access right the policy type would control (subject, action, etc.)
 - Define how the XACML policy would look like
 - Extend the VO Editor to support the policy type
 - Extend Grid Probe to crawl relevant resource configs
 - Extend Policy Comparer/Advisor to interpret the test results
- **Currently, it's not so trivial to extend the supported meta-policies (profiles)**
- **Need to refactor design to guide developers**
 - Using interfaces
 - Using generic classes

Future Directions



- **We are recruiting VO's and sites to deploy SVOPME to production Grids**
- **Ongoing enhancements**
 - VO-side needs to be able to deal with multiple grid sites for policy compare
 - Grid-side needs to be able to organize multiple VO info
 - Overall site status chart for VO's
 - Code refactoring
- **Prioritize further improvements to the tools based on feedbacks**
 - Correctness
 - Comparer may need to be changed to only return a list of allow/deny decisions
 - Currently, only examine compliance, not redundant policies
 - Additional meta-policies
 - Better defined dataflow/tools
 - Adopting OSG AuthZ profiles

Conclusions



- **SVOPME ensure uniform access to resources by providing an infrastructure to propagate, verify, and enforce VO policies at Grid sites**
- **SVOPME integrates with the OSG Authorization Infrastructure**
- **We continue to enhance SVOPME design and implementations**
- **We are soliciting interested VO's and sites to deploy SVOPME in a production environment**
- **We love to hear your comments and suggestions**
<https://ice.txcorp.com/support/wiki/MidSys/SVOPME>