Introducing SVOPME, A Scalable Virtual Organization Privileges Management Environment (VO’s Perspective)

Tech-X Corporation
Fermi National Accelerator Laboratory

Contacts:
Nanbor Wang nanbor@txcorp.com
Gabriele Garzoglio garzogli@fnal.gov

What are VO Privileges?

Virtual Organizations:

- VOs use shared resources
- VOs need to define resource usage policies 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 but others can't
- However, VOs do not manage/configure Grid sites

Grid Sites:

- Grid sites provide resources
- Grid sites don't define VOs' usage policies
- Grid sites enforce and manage user privileges
- Grid sites do not allow others (such as VO admins) to change the site configurations

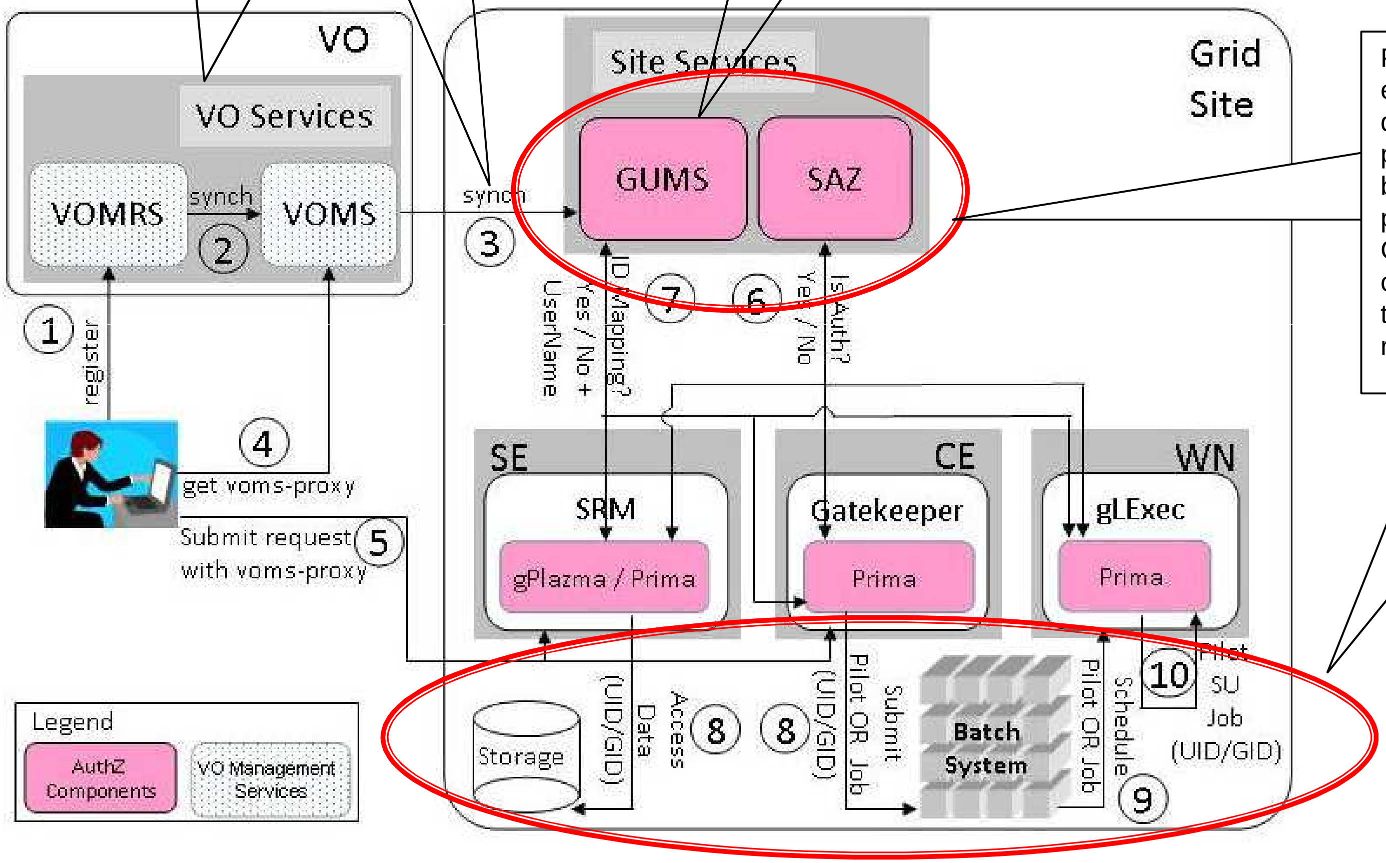
Site and VO Challenge: Enforcing heterogeneous VO privileges on multiple Grid sites to provide uniform VO Policies across the Grid (ad hoc solution: verbal communication)

State-of-the-Art
User Privilege Management

VO Services manage only VO groups and roles, and user memberships.

A VO passes only information about groups and roles to sites.

GUMS maps a user (with group/role membership) to a local ID.

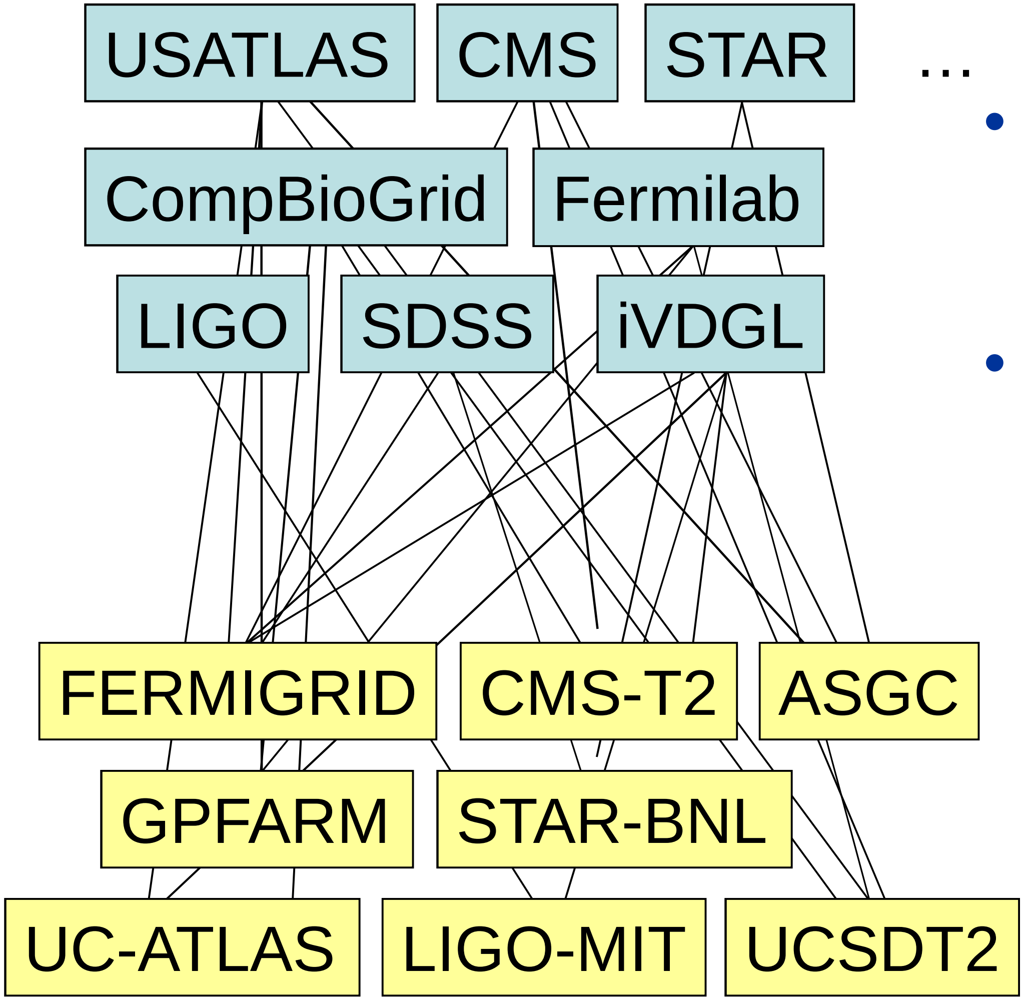


Privileges management and enforcement points are scattered at different places, using mechanisms provided by the resources. This can be as straightforward as setting the permissions to certain directories. Grid sites manage and control these configurations. They involve tweaking GUMS and all the local resources configurations.

The OSG Authorization Infrastructure

Motivations of SVOPME

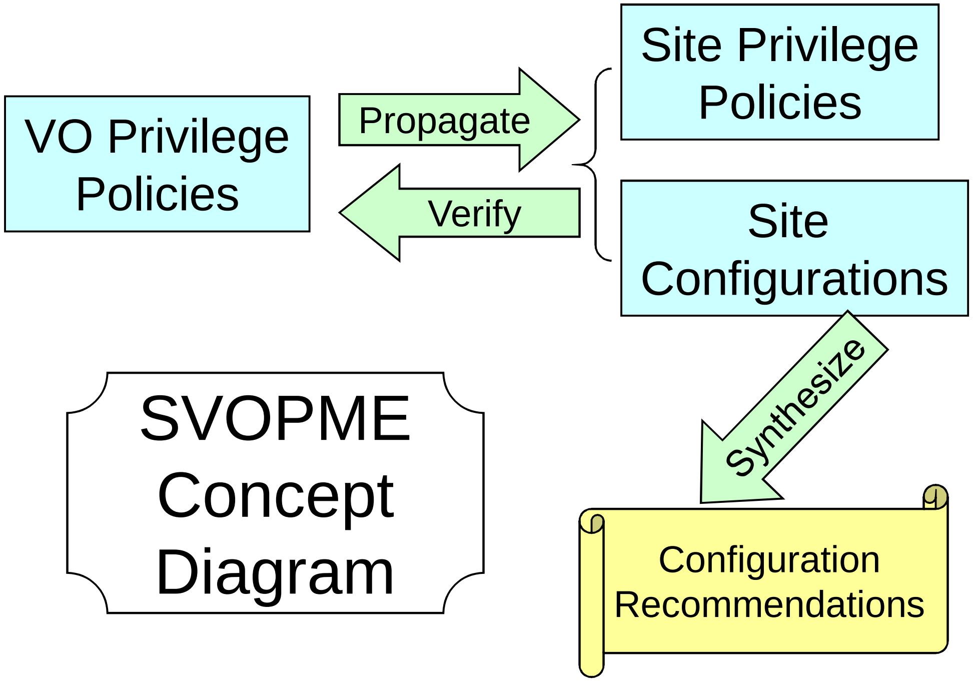
Address scalability



- **With the growth in Grid usage, both the numbers of VOs and Grid-sites increase**
- **Propagating privilege policies by verbal communication between VO and Grid site admins no longer scales**
- **SVOPME fills the gap by**
  - Providing the tools and infrastructure to help
    - VOs express their policies
    - Sites support VOs
  - Reuse proven administrative solutions – we adopt common system configuration patterns currently in use in major grid sites

SVOPME Helps VO's Propagate Privilege Policies to Grid Sites

- **SVOPME aims to replace the for verbal interaction with automated workflows**
- **SVOPME provides a policy editor to make this easy**
- **We predefine a set of policy types VOs can use to build their intended privilege policies**
- **Editor checks for conflicting policies**
- **Policies are documented in XACML format, no ambiguity**
- **Allow programmatic verification of policies**
- **Grid sites' policies can be verified against those of VOs'**



VO Privilege Policies

Propagate

Verify

Site Privilege Policies

Site Configurations

Synthesize

SVOPME Concept Diagram

Configuration Recommendations

- **SVOPME can provide recommendations to site configurations for better VO supports**

Advantages of SVOPME

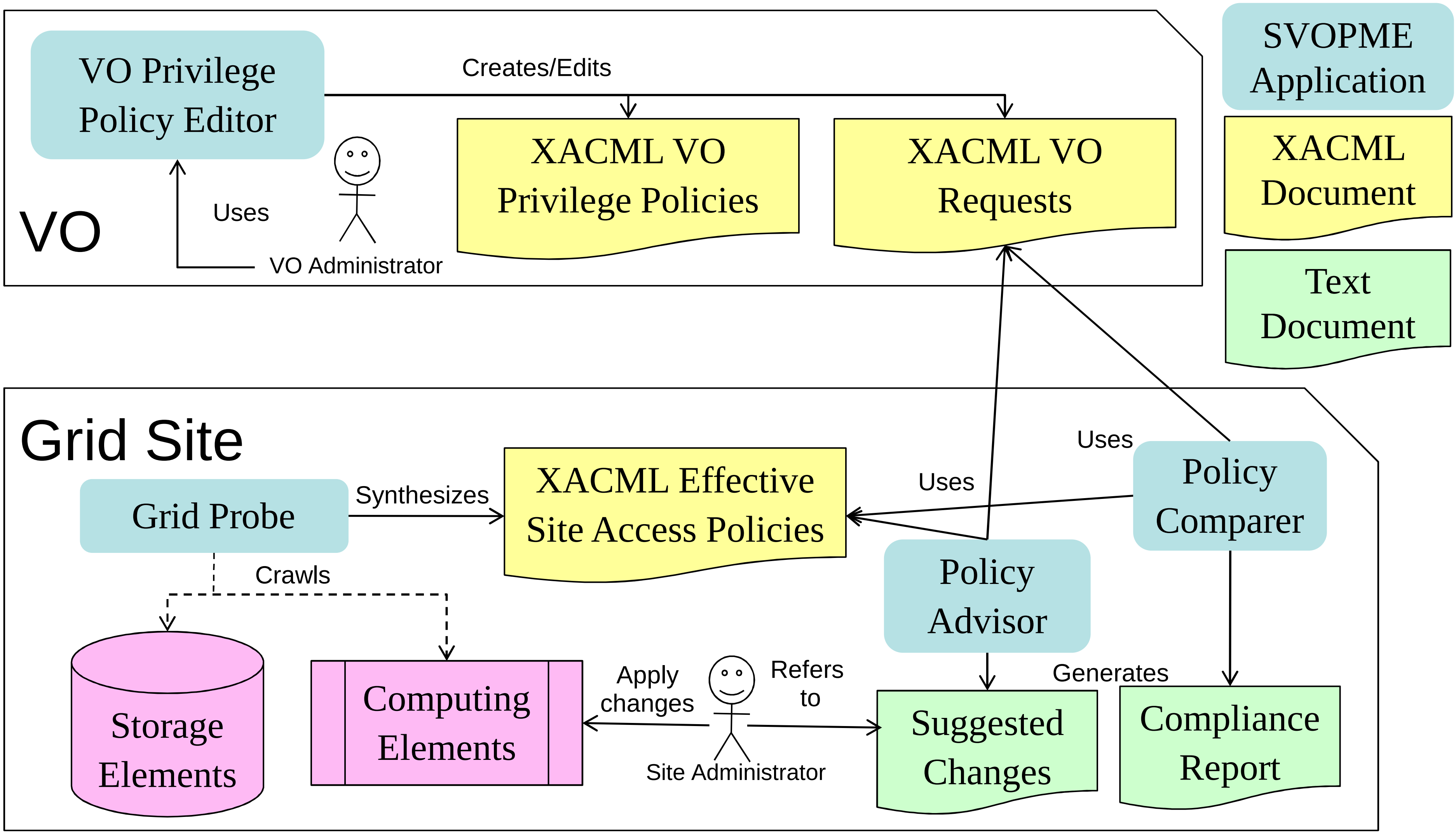
- **VO’s**
- No need to run ad-hoc jobs to figure out what policies are enforced and what not
- Provides templates to define commonly used policies
- Automates most of the communication with Sites that support the VO
- Provides the basis for the negotiation of privileges at sites that provide opportunistic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

SVOPME Currently Support These Types of Policies (VOs can define)

- **Account Type Policy:** Run job from Group(G) and Role(R) using Pool (unique)/ Group (shared) accounts.
- **Account Mapping Policy:** Must have accounts for all users in Group (G) and Role(R) (may be pool accounts or Group accounts).
- **Relative Priority Policy:** Jobs from Group (G1) and Role (R1) should have higher priority than those from user of Group (G2) and Role (R2).
- **Preemption Policy (Batch system):** Jobs from Group (G) and Role (R) should be allowed to execute for n consecutive hours without preemption.
- **Package Installation Policy (Storage):** Allow Group (G) and Role (R) to install software in $OSG_APP (assuming there is NO space reserved for any VO)
- **Unix Group Sharing Policy (Batch system):** Accounts belonging to /Group/Role=A and /Group/Role=B must share the same unix Group ID
- **File Privacy Policy (Storage):** Files Privacy Policy: Users belonging to /Group/Role=A expect privacy for their files
- **Job Suspension Policy (Batch system):** Do not suspend / resume jobs submitted from /Group/Role=A
- **Disk Quota Policy (Storage):** Assign disk quota of X GB and Y MB to accounts mapped to /Group/Role=A

SVOPME Architecture Overview



VO

VO Privilege Policy Editor

Creates/Edits

XACML VO Privilege Policies

XACML VO Requests

Uses

VO Administrator

Grid Site

Grid Probe

Synthesizes

XACML Effective Site Access Policies

Crawls

Storage Elements

Computing Elements

Apply changes

Site Administrator

Refers to

Uses

Uses

Policy Comparer

Policy Advisor

Generates

Suggested Changes

Compliance Report

SVOPME Application

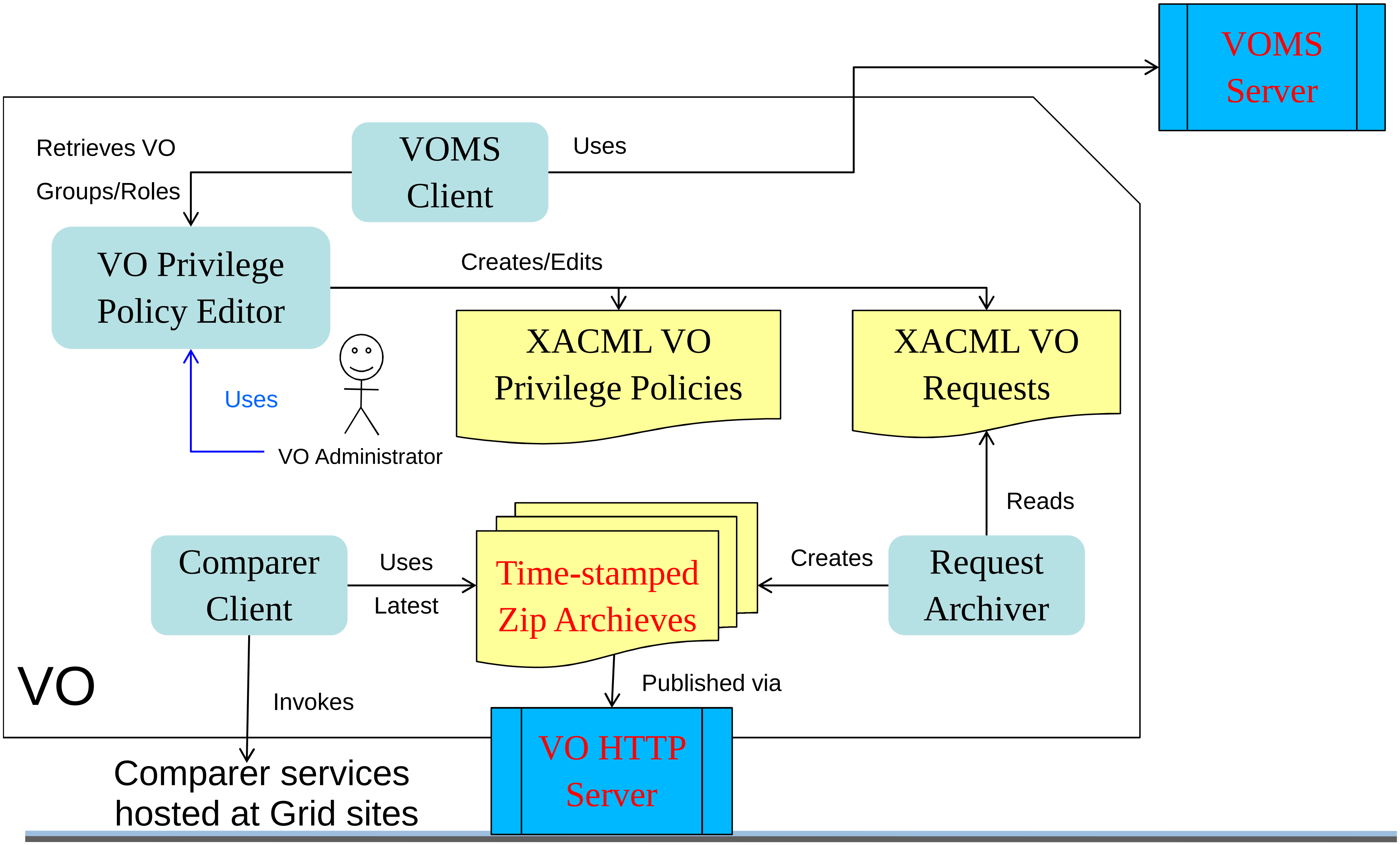
XACML Document

Text Document

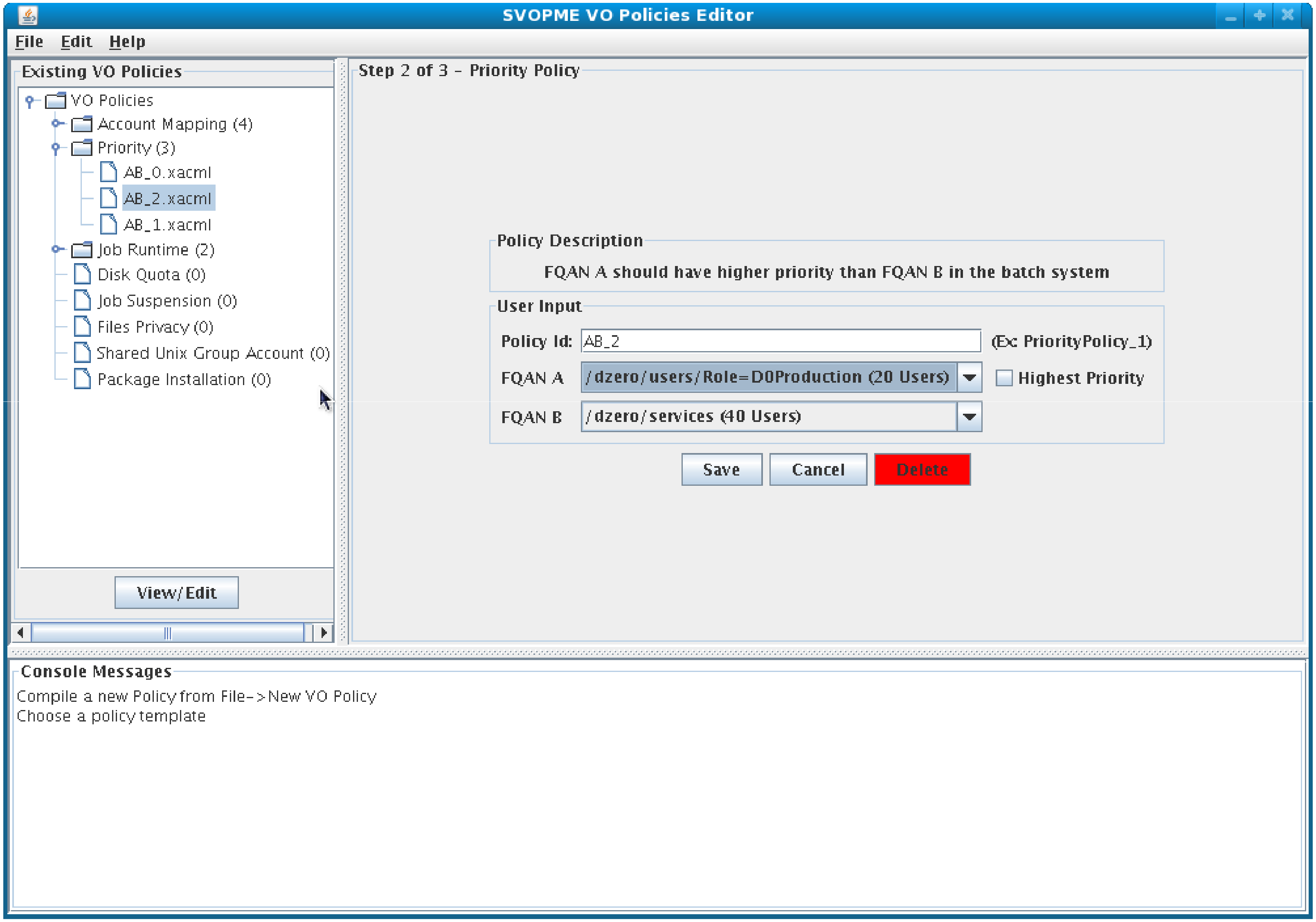
SVOPME VO Release

- VO package can be obtained from: https://ice.txcorp.com/trac/svopme/attachment/wiki/Download/svopme_vo.tar.gz
- Installation should be pretty straightforward
- We are ready to help in any areas
  - Installation
  - Configuration
  - Defining privileges
- Detailed Instructions for VO: https://ice.txcorp.com/trac/svopme/wiki/VoInsts

SVOPME VO Tools



VO Policy Editor Screenshot



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)

Console Messages

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

XACML VO Policy Editor (Create, Edit and Manage Policies)

- **Currently provide a GUI editor**
- **Environment for manage all the policies of a VO**
- **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 relatively easily.**
- **Reject redundant and contradicting policies**

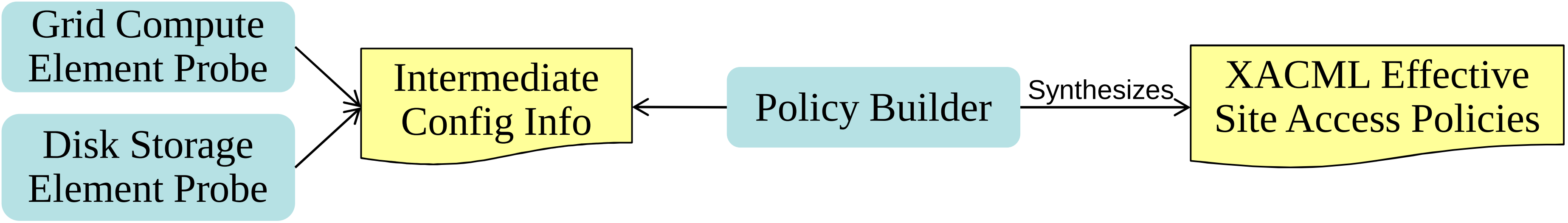
VO Policy Data Management Tools

- **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**

SVOPME Grid-Site Release

- **Grid-site package are downloaded and installed separately**
- **We will explain how SVOPME works on a site**
  - But not installation and operation details
- **Currently, SVOPME is available on FermiGrid’s Integrated Testbed (ITB)**
- **We can make it available at more sites**

Mechanism for Synthesizing Grid Site Privilege Policies



Grid Compute Element Probe

Disk Storage Element Probe

Intermediate Config Info

Policy Builder

Synthesizes

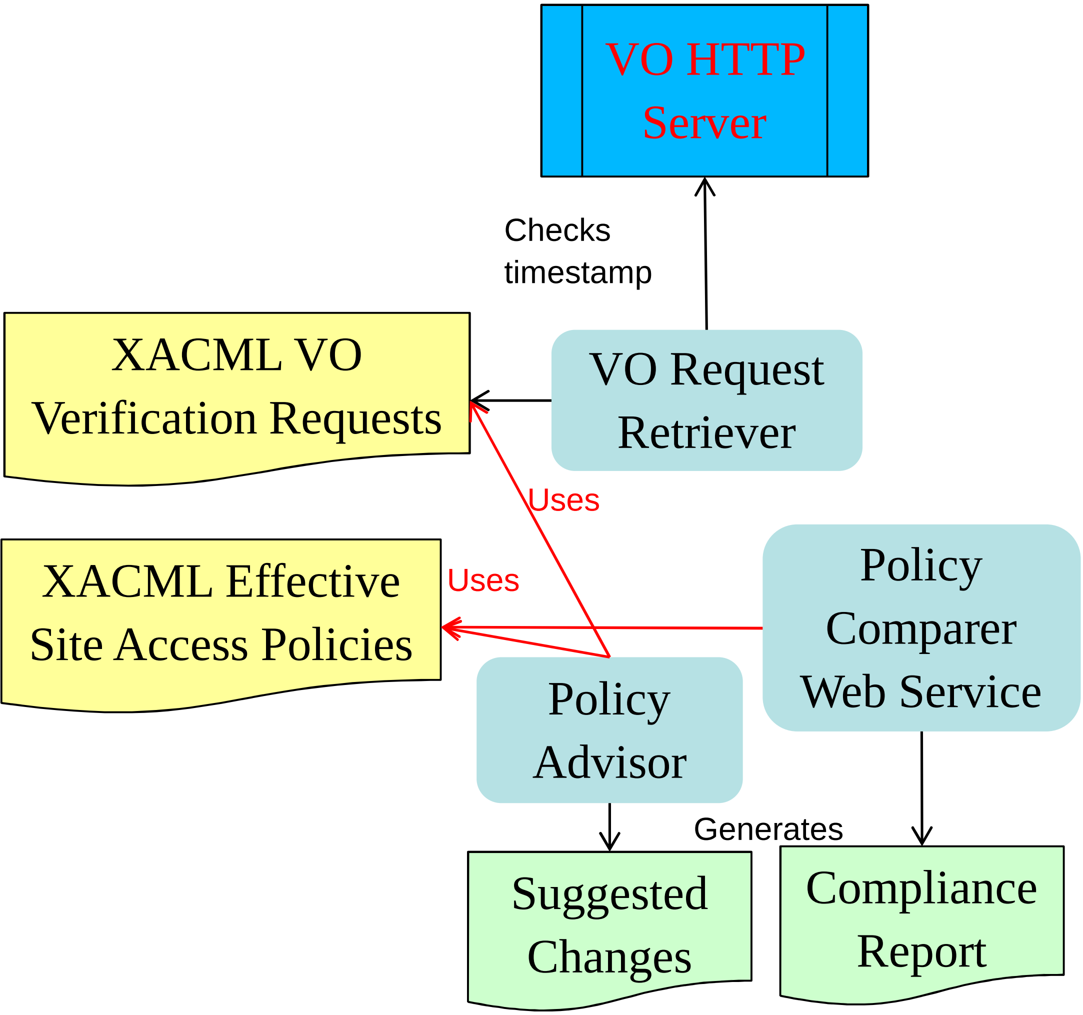
XACML Effective Site Access Policies

- **"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, is not suffient enough. Needs to be mapped to atleast 3 accounts.
```

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 confiurations

Conclusions

- SVOPME ensure uniform access to resources by providing an infrastructure to propagate, verify, and enforce VO policies at Grid sites
- 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`