Virtual Organisations

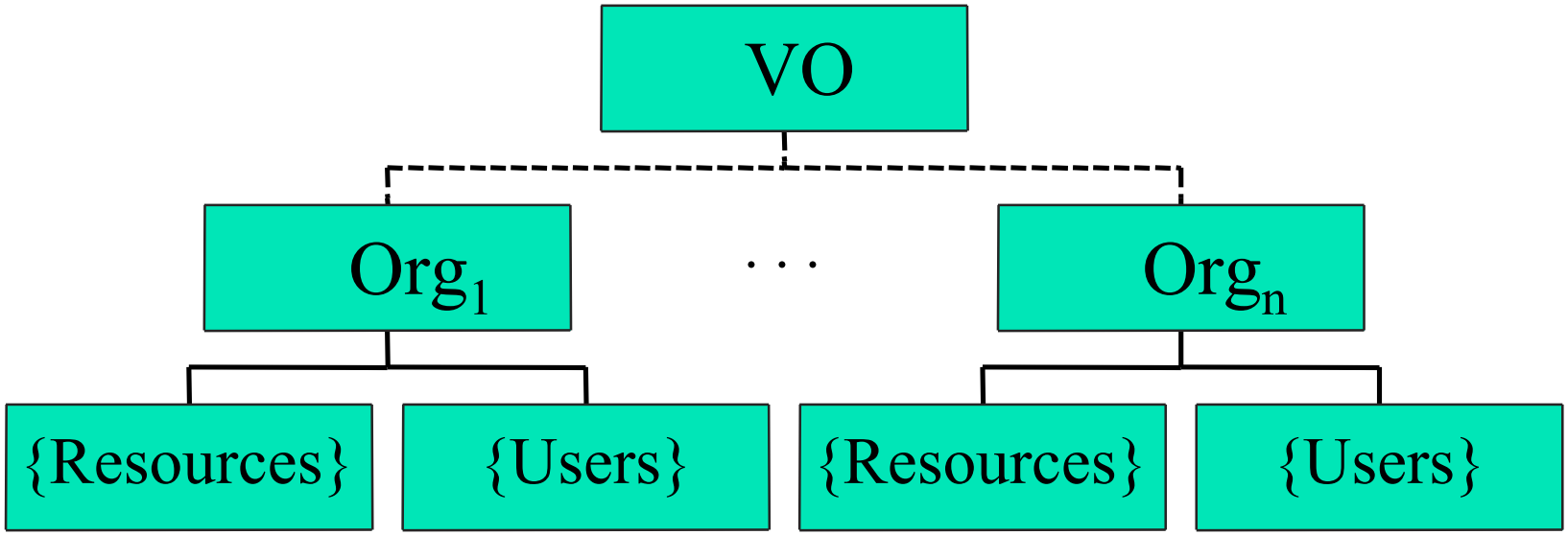
Richard Sinnott

http://csperkins.org/teaching/2004-2005/gc5/



What is a Virtual Organization?

- Definition of VO:
  - *dynamic* collection of distributed resources shared by *dynamic* collection of users from one or more organizations

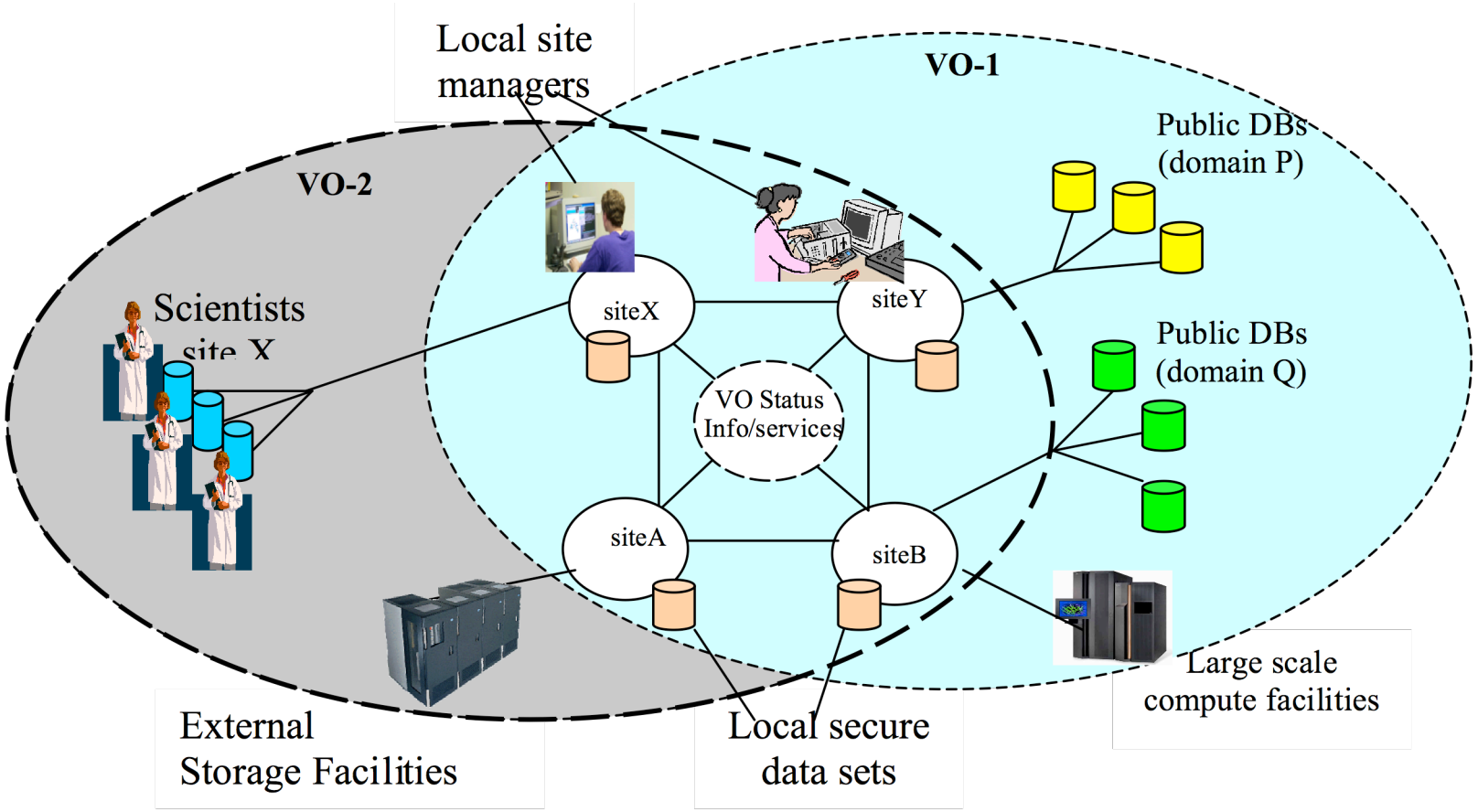


- VO technologies must scale
  - Dealing with potentially huge number of users, resources
  - Broad array of requirements from applications
    - Security, data management, high throughput computing…

Copyright © 2004 University of Glasgow

Why are VOs important?

- VOs fundamental to Grids and e-Science
  - Arguably main benefit of Grid technology
- Ability to **securely offer and access** dynamically changing distributed resources **in controlled manner** to dynamically changing groups of users



Copyright © 2004 University of Glasgow

VO Relation to Peer-Peer Network

- Sharing distributed resources
- Large user groups
- Dynamic membership
- Cooperation/Trust

- But differ in:
  - Structure, Hierarchy
  - Users, Applications
  - Open vs Closed communities
  - Objectives, Goals
    - File sharing vs resource sharing (data, compute, …)
  - Security
  - Status Information
  - …

Copyright © 2004 University of Glasgow

VO Relation to VPNs

- Sharing distributed resources
- Connecting potentially large user groups/resources

- But
  - Flexibility
  - Extensibility
  - Open vs Closed communities
  - Security
    - Network level, application level, outsourcing…
  - Status Information
  - …

Copyright © 2004 University of Glasgow

VO Practicalities

- VOs need rules/contracts (policies)
  - Who can do what, on what, in what context, …
- Policies can be direct assertions/obligations/prohibitions on specific resources/users
  - Policies can be local to VO members/resources
    - e.g. user X from site A can have access to P% resource B on site C
      - (site C responsible for local policy – autonomy!!!)
  - Policies can be on remote resources
    - users from site A can access / download data Y from site B provided they do not make it available outside of site A
      - …site B trusts site A to ensure this is the case
        - » and possibly to ensure that the security is comparable with site B
        - » … trust!!!

Copyright © 2004 University of Glasgow

VO Global Policy Options

- Policies can be global across the VO
  - Compute load across VO should be balanced between all resources
    - Implies
      - scheduling
      - job management
      - accounting
      - … agreed by all VO members
    - Policy aims to try to keep steady state of resource usage
      - May include actions to be taken to maintain desired state
        - » e.g. if any site is performing less than 25% of the work of other sites, new jobs will be scheduled on that site until the workload is balanced
  - Any user using more than 25% of total VO resources have their future jobs not accepted until below this limit
    - Difficult - distributed job management
    - What if nobody else using resources and user has large job?
    - What if policies not explicitly defined, implicit, not implementable, …?
    - Promise you won't make this data public?

Copyright © 2004 University of Glasgow

VO Global Policy Options

- VO members agree to share resources
  - "Give what you can when you can…" type policy
    - Good will and trust!
      - Easiest to achieve
    - Are we happy that others use our large resource and we get access to their smaller resource?
      - What if we are always busy? They are always free?
  - "Resource usage divided equally among VO members/organisations"
    - How do we measure resource use across VO?
      - Centralised interface (broker) through which all requests flow?
        - » Performance?
      - Job monitoring?
        - » Number of jobs completed? Time processing? Disks used?
        - » Monitoring all jobs, some jobs, jobs per user/per project/per site/per VO…
  - "Get what you give …" type policy
      - Each VO member/organisation receives credit equivalent to the resource utilisation they provide to other users
        - » What is unit of accounting?

Copyright © 2004 University of Glasgow

VO Policy Issues

- Type and quality of resources vary
  - How do we compare different processors?
    - A 2 day job on a PC with PIII processor and 2GB RAM might complete in 5 minutes on a IBM P690 Regatta Server with 2TB RAM
  - How do we compare processors to disks to IO characteristics to available network at that resource site to …?
    - A 1 day job mining data in flat text files could be done in seconds if the data was indexed and in a DB
  - Often cannot be decided until know exact nature of jobs themselves
    - Some jobs lot more IO intensive
    - Some jobs require inter-process communication
    - Some jobs designed for specific hardware infrastructure, others more generic
    - Some jobs need to move lots of data to/from resource

Copyright © 2004 University of Glasgow

Policy Considerations

- Do we always want to make such detailed agreements
  - Do we know before setting up VO exactly what policies will be/should be?
- Can we adapt to changing conditions?
- When should the VO take action to enforce it's policy?
  - Always for everything
    - Performance?
  - First violation (trust broken)
  - Sometimes based on statistical averaging of resource usage
- What action should the VO take?
  - Warn/cut-off
    - Demand more access to resources?
    - Restrict access to resources?
    - Remove user/resource from VO
      - Trust broken
  - Redirection
- What if policy violation beyond control of VO partner?
    - network failure, snooper accessing data in transit between sites

Copyright © 2004 University of Glasgow

VO Consequences

- Members/organizations need to know what will be expected of them before they join VO and what it means to allow someone/some site to join their VO
  - …and consequences of what happens if they don't meet the agreements
- Individual sites trusted to implement the agreed policy
  - If some sites do not conform to policy (or violate) policy?
    - Security ramifications…?
      - Weakest link can affect all others!
        - » Totally secure supercomputing facility allowing access to scientist with own PC in remote location
        - » How do we know they are taking adequate security precautions?
      - Legal impact,
        - » e.g. Data protection act
      - Loss of trust
      - …
    - Increased load on other resources

Copyright © 2004 University of Glasgow

Technologies for VO

- How does Grid technology meet these challenges?
  - Key that we need way to describe, implement and check/enforce policies

- Should be done at many levels
  - Abstract level to capture overall agreements
    - How best to describe resources, actions, people, …?
  - Design level to ensure that specific points where decisions needed are identified
    - Is there a generic way to achieve this…?
  - Implementation level to ensure that agreements/policies enforced in right places
    - Need to implement collections of rules that can be easily enforced across a variety of end systems

Copyright © 2004 University of Glasgow

Technologies for VO

- Historically expression of policies not fine grained with Grid toolkits, e.g. Globus
  - For example policies on security based on PKI (previous lecture) and GSI (next lecture)
    - Globus uses gridmapfile
      - "/C=UK/O=eScience/OU=Glasgow/L=Compserv/CN=john watt" jwatt
      - "/C=UK/O=eScience/OU=Glasgow/L=Compserv/CN=richard sinnott" ros
      - …
    - Users have X.509 certificates which are used to support PKI (single sign on)
    - Applications can check that invoker has appropriate credentials to invoke service (more on GSI in next lecture)
      - i.e. I know that the person with this certificate is registered in my gridmapfile
        - » provides for authentication but need finer grain security (rules/policies)
        - » i.e. authorisation

Copyright © 2004 University of Glasgow

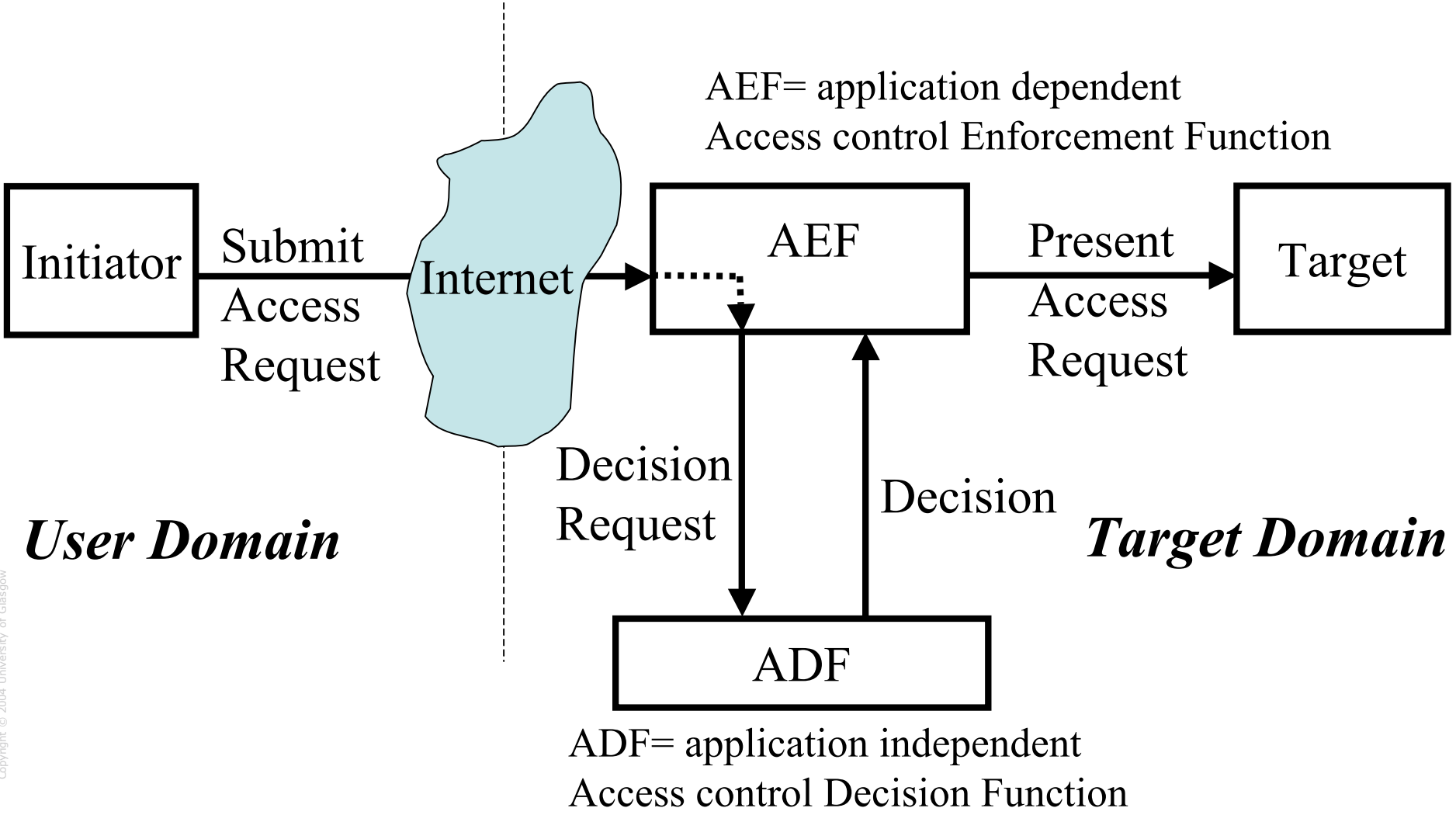
Authorization Technologies for VO

- Various technologies for authorization including
  - PERMIS
    - PrivilEge and Role Management Infrastructure Standards Validation
      - http://www.permis.org
  - Community Authorisation Service
    - http://www.globus.org/security/CAS/
  - AKENTI
    - http://www-itg.lbl.giv/security/akenti
  - CARDEA
    - http://www.nas.nasa.gov/Research/Reports/Techreports/2003/nas-03-020-abstract.html
  - VOMS
    - http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html
  - All of them predominantly work at the local policy level

Copyright © 2004 University of Glasgow

Standards for Generic Authorisation

Generic way to achieve authorisation defined in X.812|
ISO 10181-3 Access Control Framework



Copyright © 2004 University of Glasgow

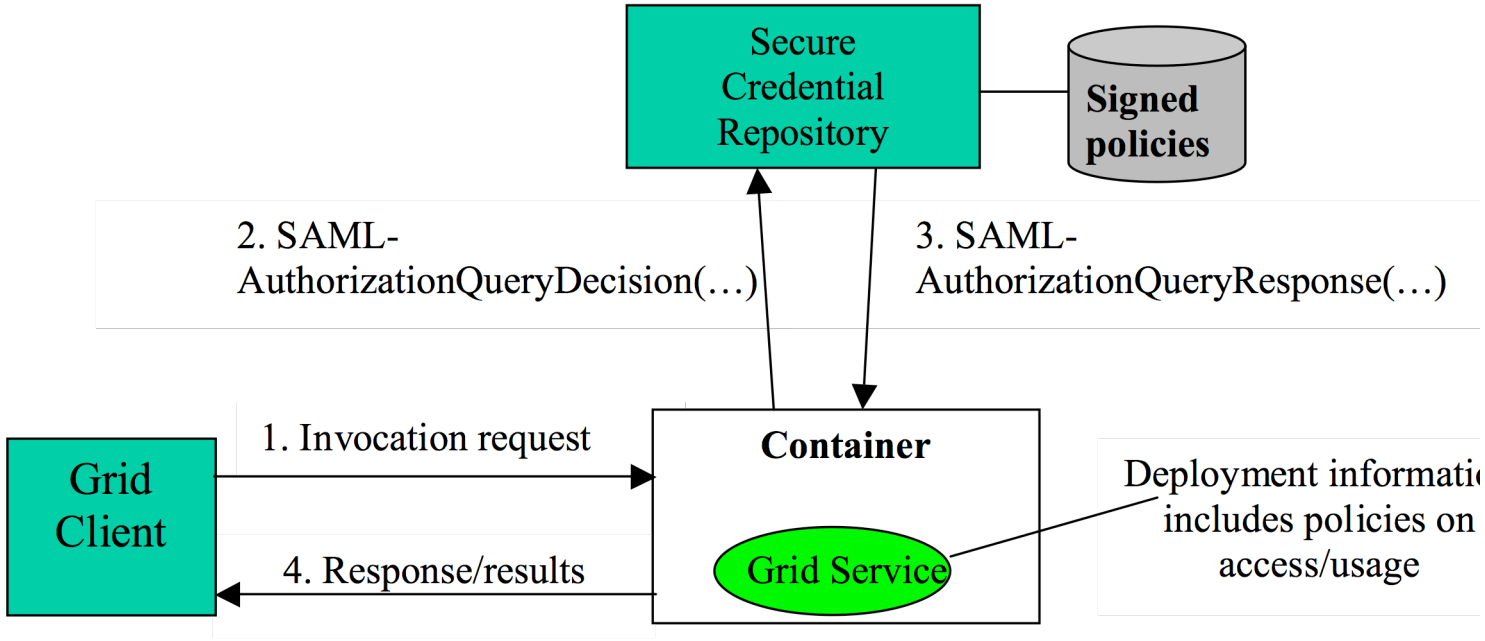
Grid APIs for Generic Authorisation

- GGF have defined generic API for Grid service authorization
  - SAML AuthZ specification defines a number of elements for making assertions and queries regarding authentication, authorization decisions
  - Includes message exchange between a policy enforcement point (PEP) and a policy decision point (PDP)
    - consisting of *AuthorizationDecisionQuery* flowing from the PEP to the PDP, with an assertion returned containing some number of *AuthorizationDecisionStatements*
    - *AuthorizationDecisionQuery* itself consists of
      - A *Subject* element containing a *NameIdentifier* specifying the initiator identity
      - A *Resource* element specifying the resource to which the request to be authorized is being made.
      - One or more *Action* elements specifying the actions being requested on the resources
    - Result is a *SimpleAuthorizationDecisionStatement* (granted/denied Boolean) and an *ExtendedAuthorizationDecisionQuery* that allows the PEP to specify whether the simple or full authorization decision is to be returned

Copyright © 2004 University of Glasgow

Grid APIs for Generic Authorisation …ctd

- SAML AuthZ specification provides generic PEP approach for ALL Grid services
  - … or at least all GT3.3 based services



  - PDP application specific
    - Will look at PERMIS
      - Default behaviour is if not explicitly granted by policy, then rejected

Copyright © 2004 University of Glasgow

Role Based Access Controls

- Need to be able to express and enforce policies
  - Common approach is role based authorisation infrastructures
    - PERMIS, CAS, …
- Basic idea is to define:
  - *roles* applicable to specific VO
    - roles often hierarchical
      - Role X ≥ Role Y ≥ Role Z
      - Manager can do everything (and more) than an employee can do who can do everything (and more) than a trainee can do
  - *actions* allowed/not allowed for VO members
  - *resources* comprising VO infrastructure (computers, data resources etc)
- A policy then consists of sets of these rules
    - *{ Role x Action x Target }*
      - Can user with VO role X invoke service Y on resource Z?
    - Policy itself can be represented in many ways,
      - e.g. XML document, SAML, XACML, …

Copyright © 2004 University of Glasgow

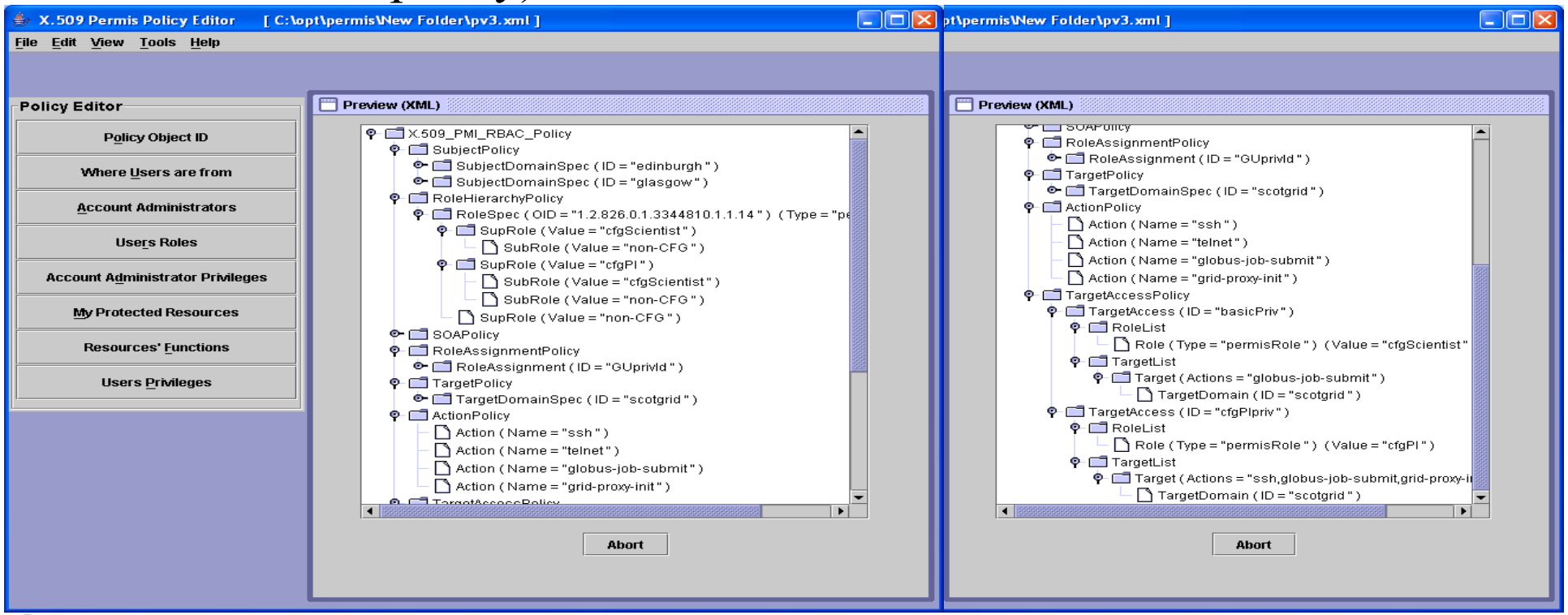
RBAC Policy Components

- Subject Policy
  - Specifies subject domains, e.g. dcs.gla.ac.uk
- Role Hierarchy Policy
  - Specifies hierarchy of role values, e.g. VO scientist, sys-admin
- SOA Policy
  - Specifies who is trusted to issue ACs (typically local sys-admin)
- Role Assignment Policy
  - Says which roles can be given to which subjects by which SOAs, with which validity times and whether delegation is allowed (depends on VO)
- Target Policy
  - Specifies the target domains covered by this policy
- Action Policy
  - Specifies the actions (operations) supported by the targets
- Target Access Policy
  - Specifies which roles are needed to access which targets for which actions, and under what conditions

Copyright © 2004 University of Glasgow

PERMIS Based Authorisation

- PERMIS Policies created with PERMIS PolicyEditor (output is XML based policy)



- PERMIS Privilege Allocator then used to sign policies
  - Associates roles with specific users
    - Policies stored as attribute certificates in LDAP server

Copyright © 2004 University of Glasg

Status of Technologies

- PERMIS is arguably most mature authorization technology
  - … but open issues with tools still
- BRIDGES/DyVOSE projects only ones exploring the SAML AuthZ API right now
  - GT3.3 stability issues
    - GT4 support, WSRF support…
  - CAS, others
    - …various levels of "flakiness"
- Don't address system wide policies?

Copyright © 2004 University of Glasgow

Conclusions

- VOs crucial to Grids
  - Must overcome limitations of PKI scalability, security
- Need way to express rules/policies
  - How detailed?
  - How dynamic?
  - What about performance…?
- Standards and specifications/implementations being put together
  - GGF AuthZ looks promising
    - Will be explored in assignment
- Clear need for more experiences applying technologies

Copyright © 2004 University of Glasgow