Grid Technologies for AAI*

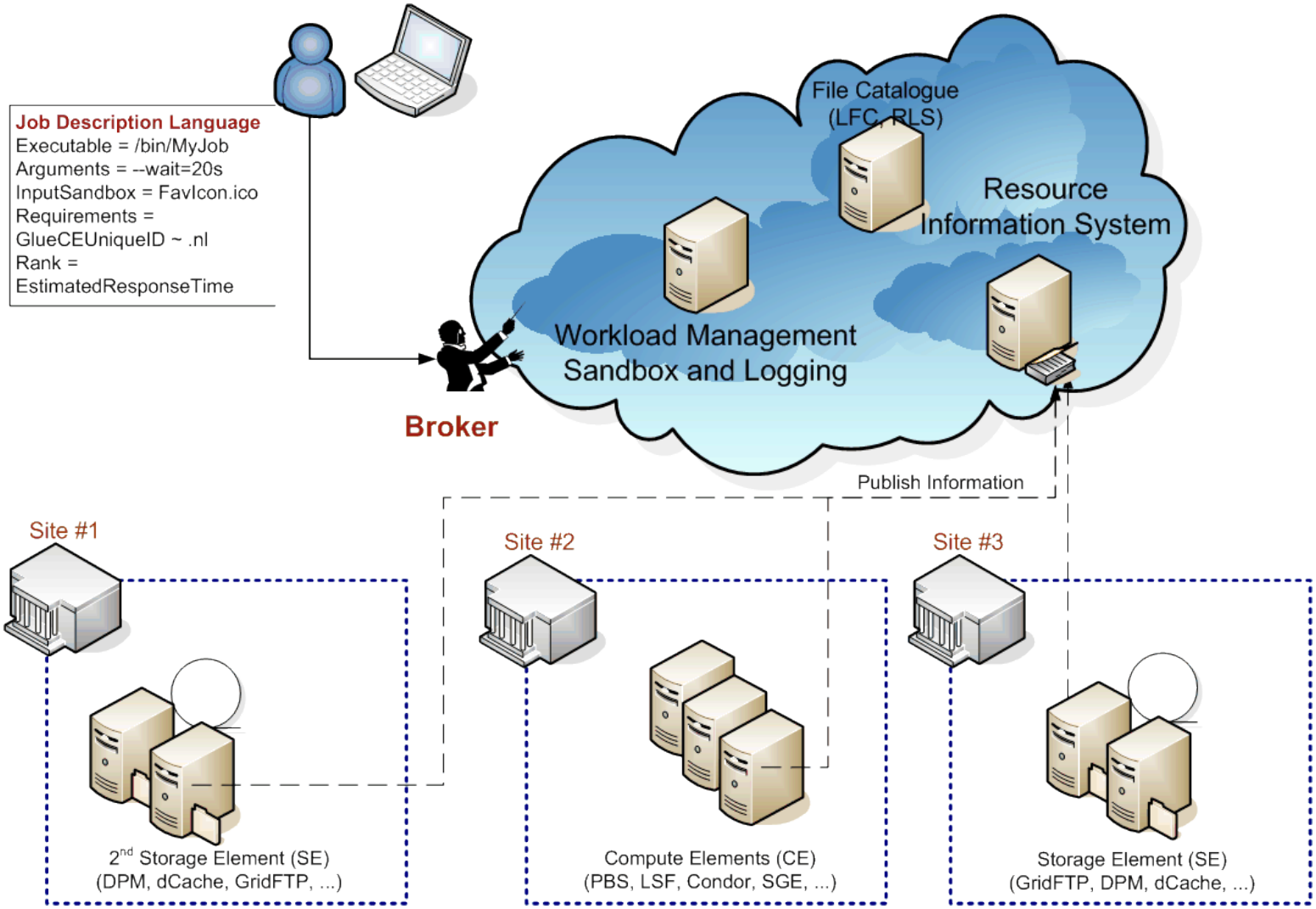
> in Selected Grid Infrastructures and using a Subset of the available technologies (2010)

David Groep, Nikhef

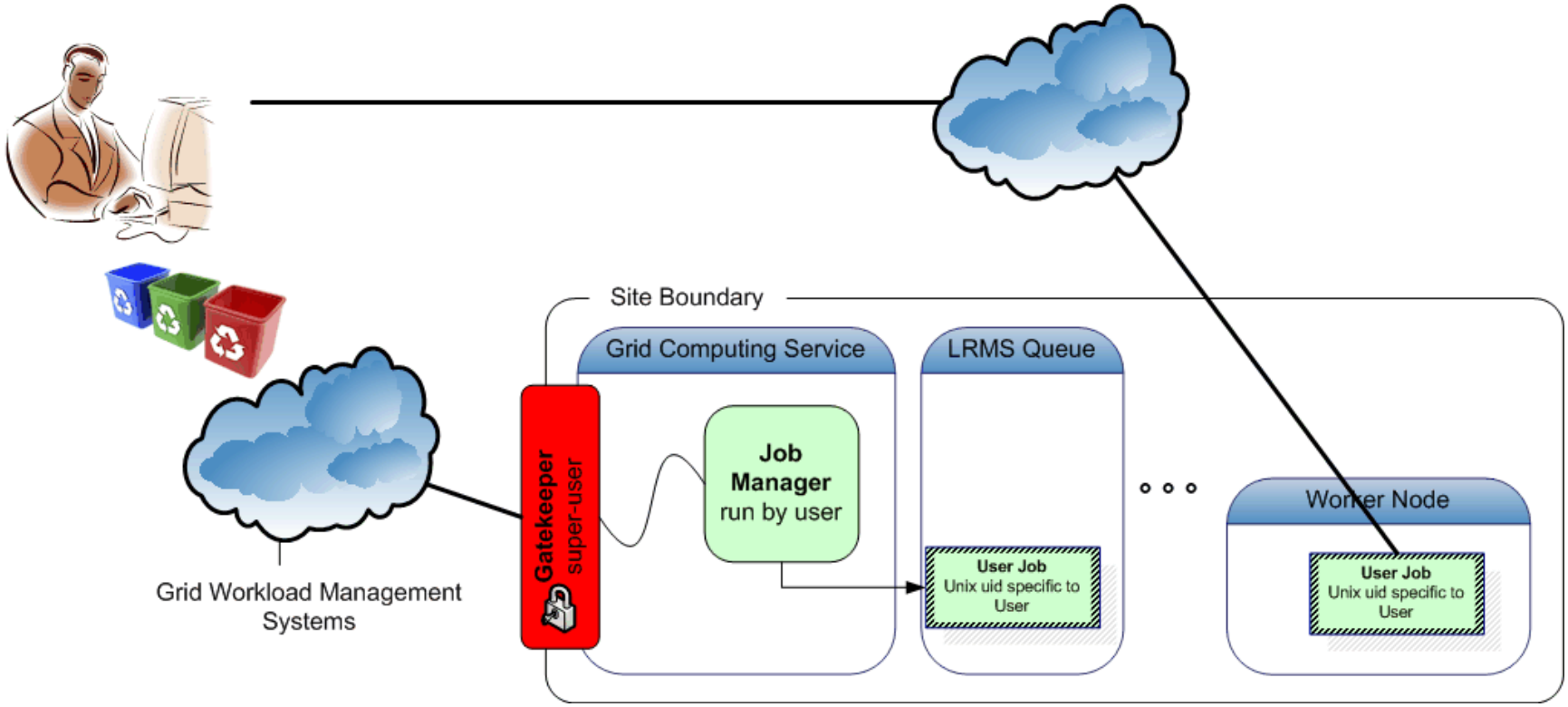
with graphics by many others from publicly available sources …
based on the ISGC2010 Security Middleware presentation

- Grid is *global*
- based around (dynamic)
  user communities
  not around their home organisations
- that may live long or be over quickly
- deal with compute, data, visualisation, services, and more
- and can consist of staff, students, technicians, …

A Typical Grid Scenario

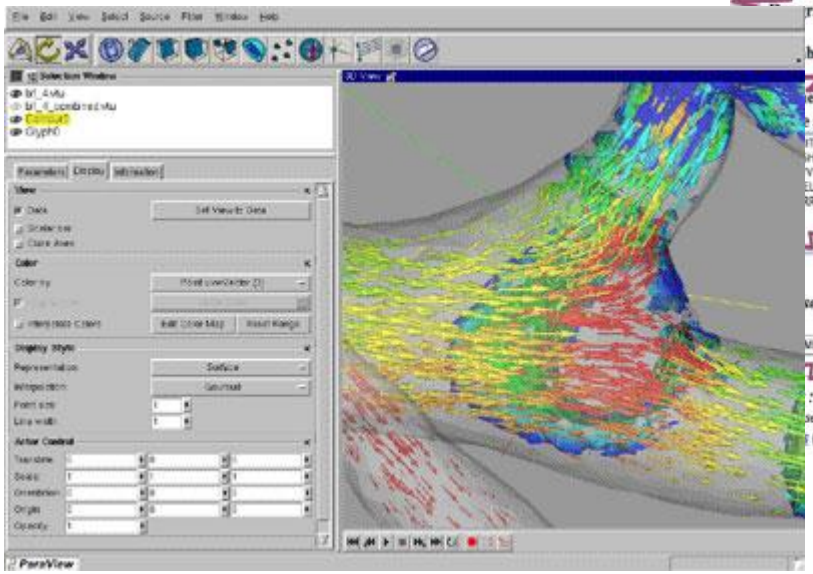


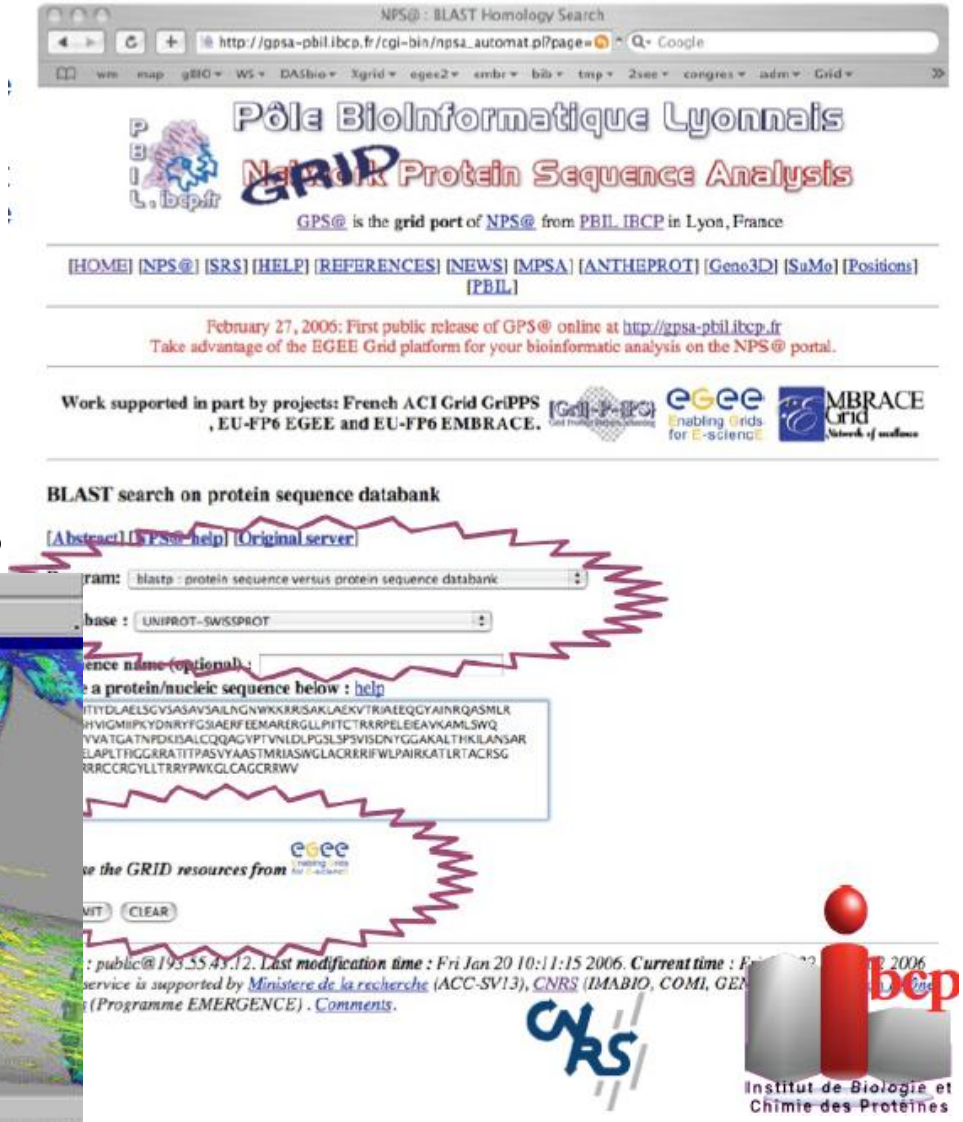
Non-interactive, autonomous work



Or via portals

- Flexible portals acting on behalf of the user,
- work-flow portals with canned applications
- turn-around: min-hours





What drove the Grid AAI model

- Accommodate multiple sources for assertions
  - AuthN vs. AuthZ is a logical implementable separation
- Accommodate delegation (disconnected operation)
  - Entities act on behalf of a user
  - Service providers do not know (or cannot fully trust) each other
  - Commensurate impact of resource compromise
    - compromise of small resource should have limited impact
- Accommodate individual, independent researchers
  - collaboration without necessity to involve bureaucracy
- Inspire enough trust for resource providers to relinquish per-user local registration and allow direct access to their systems
- Has to work *now* (and has had to work since 2002!)

Authentication (vs. Authorization)

Obtaining trustworthy unique, persistent ID

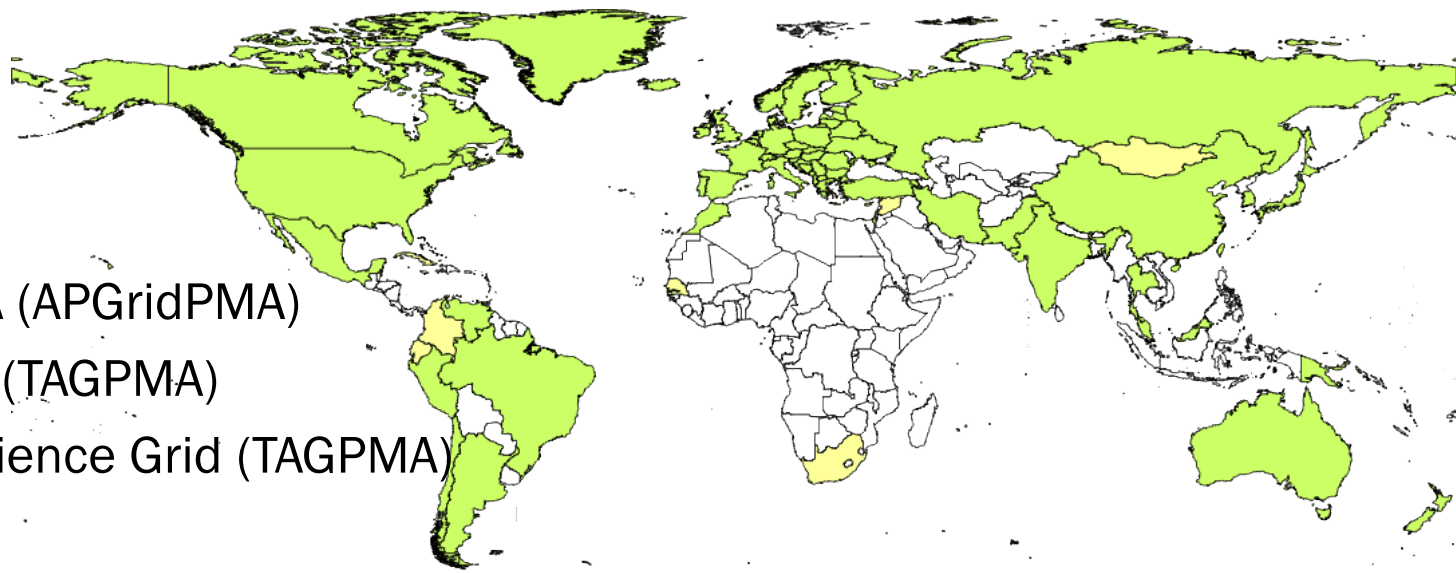
Delegation and proxies

‘GRID’ SECURITY MECHANISM FOUNDATIONS AND SCOPE

A coordinated trust fabric: IGTF

A ‘policy bridge’ infrastructure for authentication

- Today there are 86 accredited authorities
- From 54 countries or economic regions
- Direct relying party (customer!) representation & influence
- from countries … and major cross-national organisations
  - EGI
  - DEISA
  - wLCG
  - TERENA
  - PRAGMA (APGridPMA)
  - Teragrid (TAGPMA)
  - Open Science Grid (TAGPMA)



Authentication Policy Guidelines

IGTF established a single trust fabric, incorporating authorities using different techniques

Common Elements

- Unique Subject Naming
- Identifier Association
- Publication & IPR
- Contact and incident response
- Auditability

Profiles

- Classic PKI
  - Real-time vetting (F2F or TTP)
  - 13 months life time
- SLCS
  - Existing IdM databases
  - 100k – 1Ms life time
- MICS
  - IdM Federation with F2F
  - managed, revocable, identity
  - 13 months max

https://www.eugridpma.org/guidelines/

Hiding PKI internals from the User

- PKI is a great transport technology …
  … but a no-go for most users

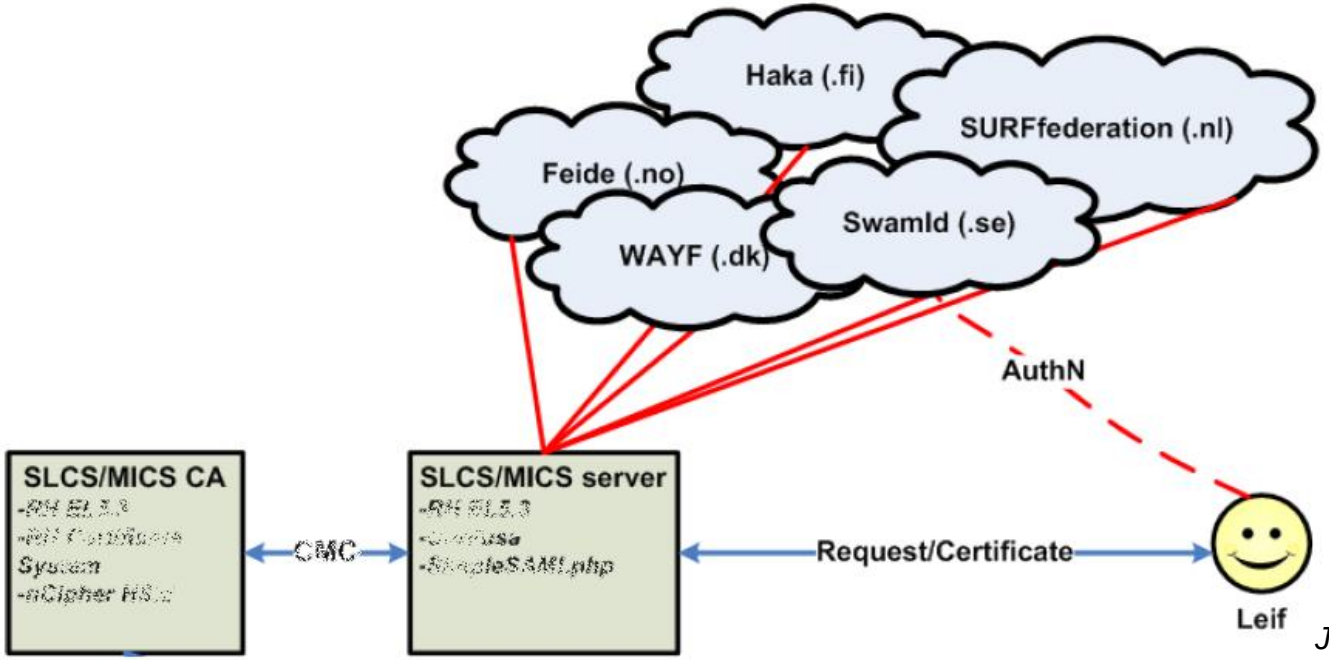
- How to hide the PKI internals?
  - do away with multiple ID checks by leveraging federations (*TERENA TCS, SWITCHaai, DFNaai*)
  - hide credential management in client tools (*jGridstart*)
  - use offer credential management as a service (*MyProxy*)

- *user does not see PKI that drives the infrastructure*

A Federated PKI

Implementations:
- *DFN Grid CA*
- *SWITCHaai SLCS*
- *TERENA eScience Personal CA*
- *CI Logon (Q4 2010)*
- *ARCS CA (end 2010)*

> Use your federation ID
> ... to authenticate to a service
> ... that issues a certificate
> ... recognised by the Grid today



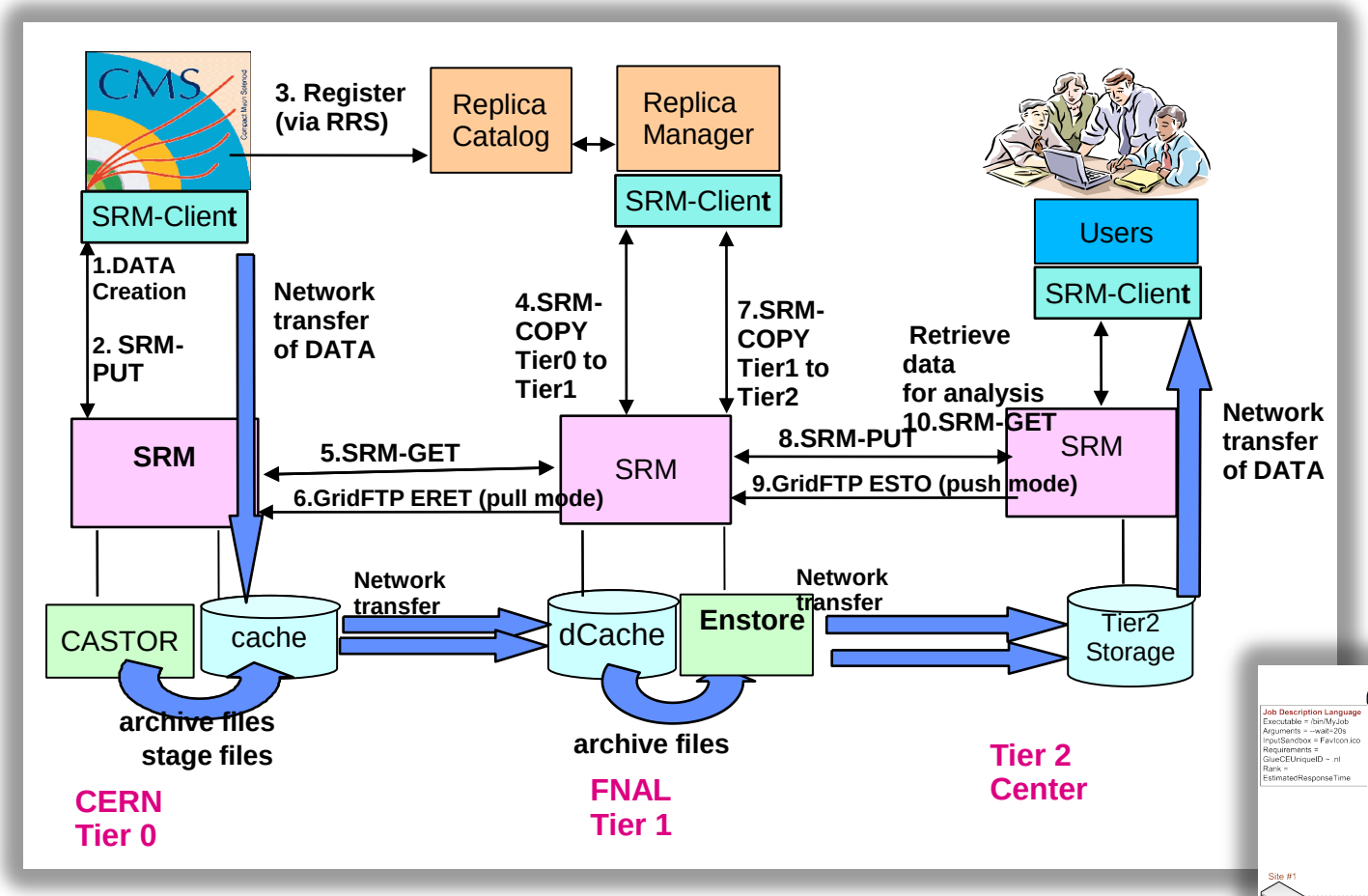
Outdated Graphic from: Jan Meijer, UNINETT

Delegation

RFC3820

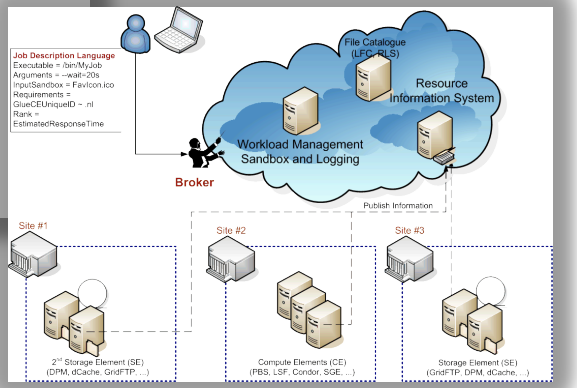
AUTOMATED TASKS, SERVICES, AND BROKERING

Distributed Services in Grid



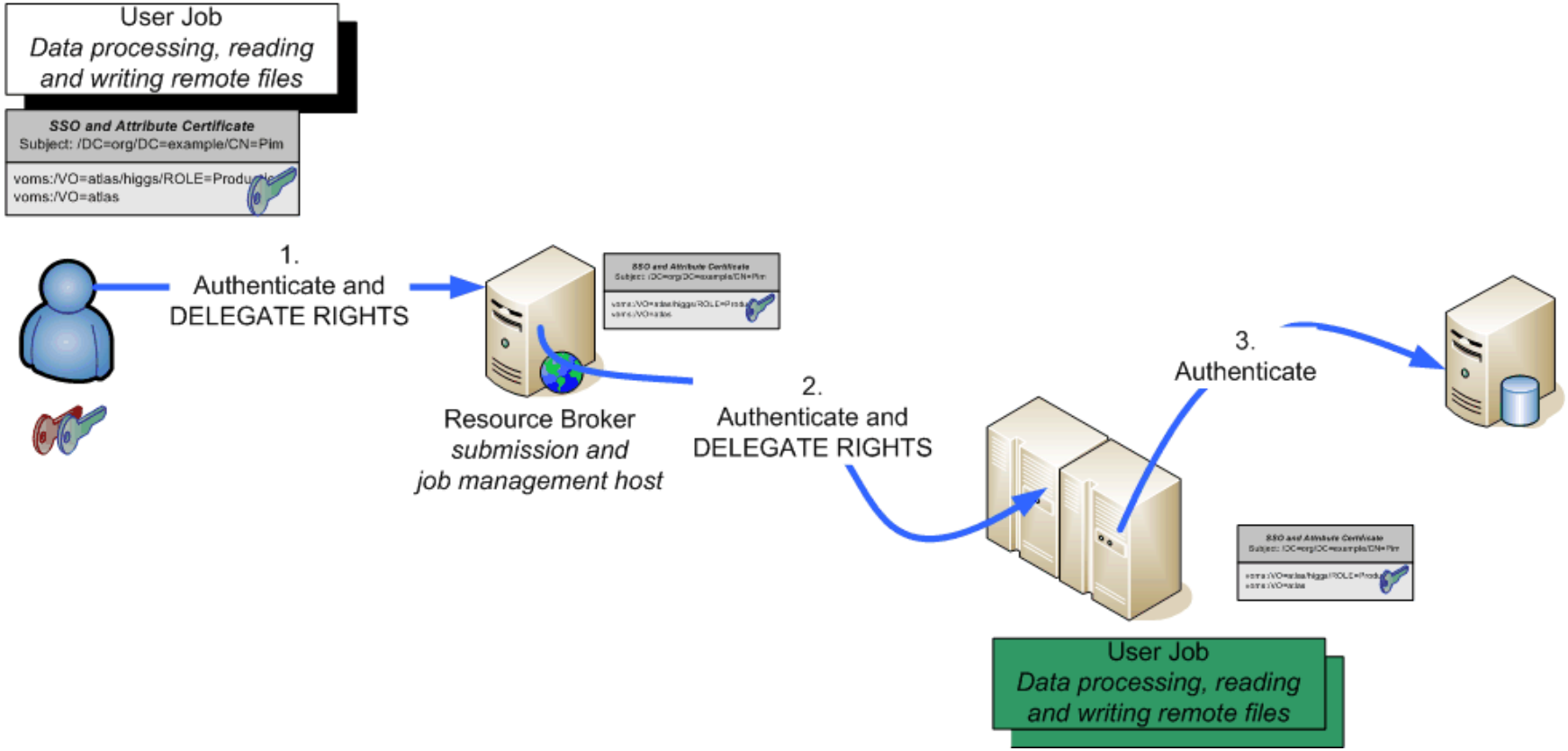
SRM graphic: Timur Perelmutov and Don Petravick, Fermilab, US

Example
file transfer services using managed third-party copy via the SRM protocol



Example
automatic workload distribution across many sites in a Grid

Delegating rights and privileges



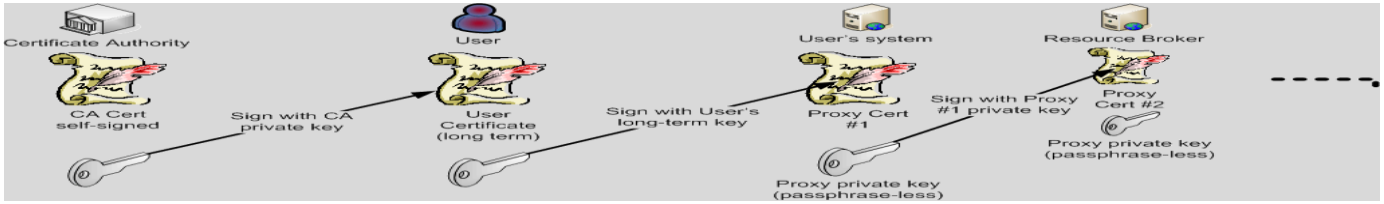
Delegation – why break the recursion?

- Mechanism to have someone, or some-thing – a program – act on your behalf
  - as yourself
  - with a (sub)set of your rights
- Essential for the grid model to work
  - since the grid is highly dynamic and resources do not necessarily know about each other
  - only the user (and VO) can 'grasp' the current view of their grid
- GSI-PKI (and now finally some recent SAML) define
  - GSI (PKI) through 'proxy' certificates (see RFC3820)
  - SAML through *Subject Confirmation*, linking to at least one key or name

Delegation, but to whom?

- RFC3820 – dynamic delegation via ‘proxy certs’
  - Subject name of the proxy derived from issuer

    “/DC=org/DC=example/CN=John Doe/CN=24623/CN=535431”
    is a proxy for user
    “/DC=org/DC=example/CN=John Doe”
  - Contains **policy constraints on delegation**



- AuthZ based on end-entity + embedded attributes&policies
- with SAML, delegation can be to any NameID
- in RFC3820, these are called ‘independent proxies’

Verifying authentication and X.509

- ‘Conventional’ PKI engines in *nix domain
  - OpenSSL, Apache mod_ssl, nss
  - Java JCE providers, such as BouncyCastle
  - Perl, Python usually wrappers around OpenSSL
- With proxy support
  - OpenSSL (0.9.8+)
  - Globus Toolkit (C, Java)
  - gLite proxyVerify library (LCMAPS)
  - gLite TrustManager on Java’s BouncyCastle
  - GridSite
- and always ensure *proxy policies* are implemented & enforced

Community organisation

Proxies and delegation with attributes: VOMS

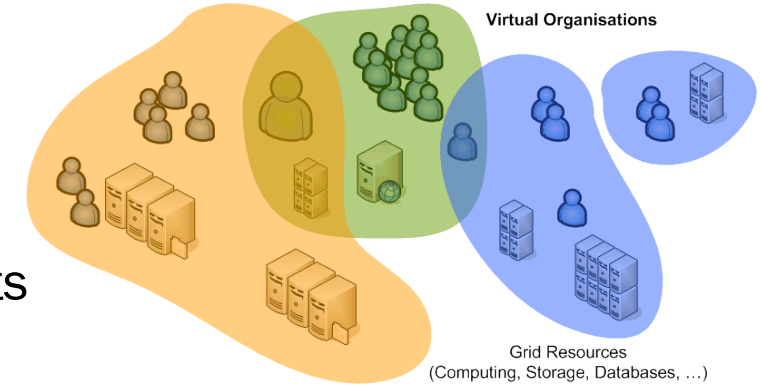
Authorization with VOMS: autonomous, GUMS

Towards a multi-authority world

USER COMMUNITY MODELS

Authorization: VO representations

- VO*: directory (database) of members, groups, roles, attributes
- based on identifiers issues at the AuthN stage
- Membership information is to be conveyed to the resource providers



  - configured statically, out of band
  - in advance, by periodically pulling lists VO (LDAP) directories
  - in VO-signed assertions pushed with the request: VOMS, Community AuthZ Service
  - Push or pull assertions via SAML

* this is the 'EGI' or e-Infrastructure sense of VO, representing users.
Other definitions at times include resources providers, in a more vertically oriented 'silo' model

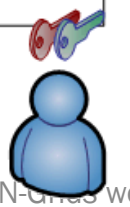
VOMS: the ‘proxy’ as a container

Virtual Organisation Management System (VOMS)

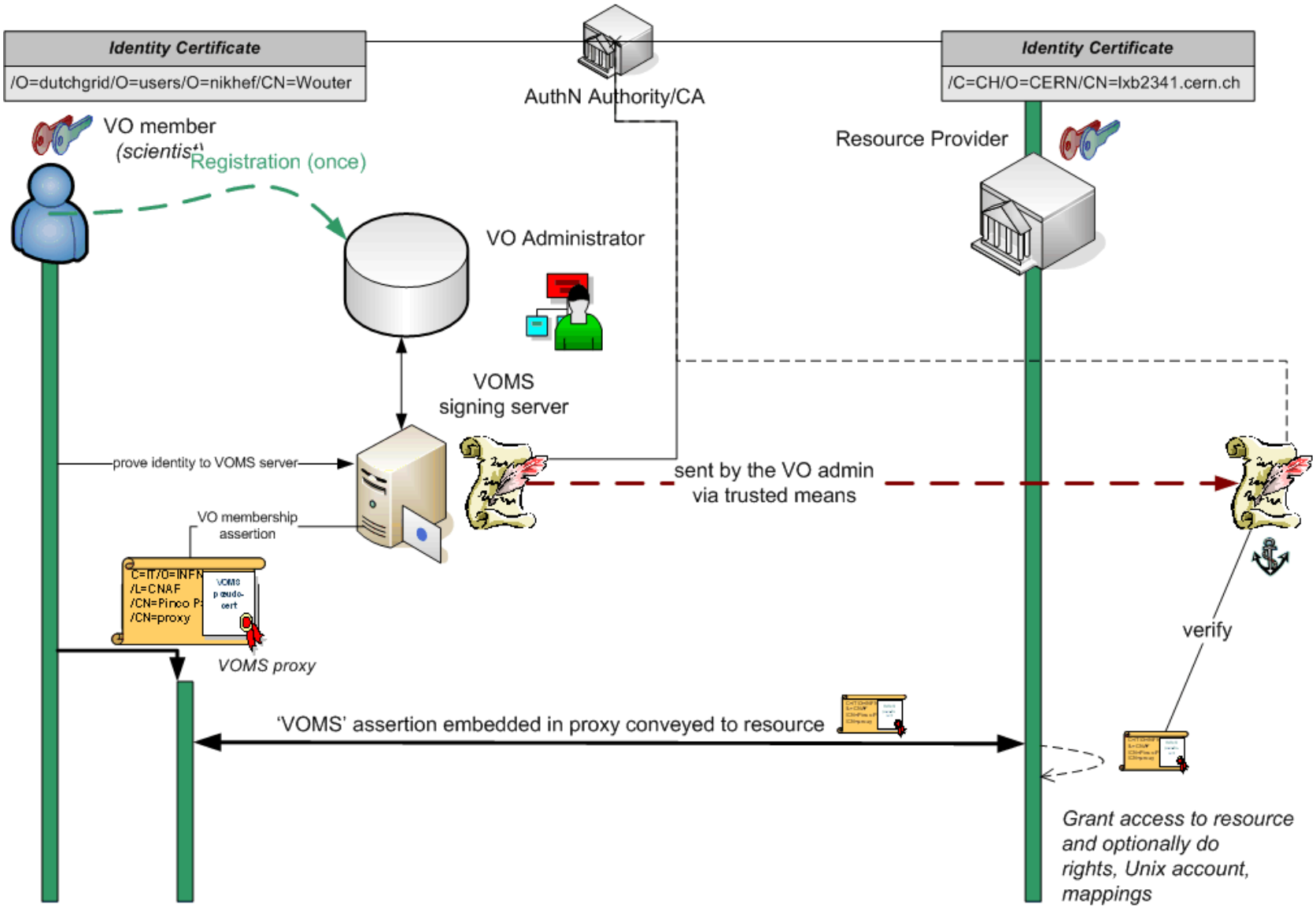
- developed by INFN for EU DataTAG and EGEE
- used by VOs in EGI, Open Science Grid, NAREGI, …
- push-model signed VO membership tokens
  - using the traditional X.509 ‘proxy’ certificate for trans-shipment
  - fully backward-compatible with only-identity-based mechanisms

VOMS proxy with embedded VO assertion
Serial Number: 26423 (0x6737)
Issuer: O=dutchgrid, O=users, O=nikhef, CN=David Groep
Not Before: Oct 16 12:46:28 2006 GMT
Not After : Oct 17 00:51:28 2006 GMT
Subject: O=dutchgrid, O=users, O=nikhef, CN=David Groep, CN=proxy
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (512 bit)
X509v3 extensions:
1.3.6.1.4.1.8005.100.100.5:
0...0...0...0.....0W.U0O.M0K1.0...U./dteam/ne/ROLE=null/0...0...0...0
X509v3 Key Usage:
Digital Signature, Key Encipherment, Data Encipherment
Signature Algorithm: md5WithRSAEncryption

Attribute Certificate	
INTEGER	1
SUBJECT	/O=dutchgrid/O=users/O=nikhef/CN=David Groep
SERIAL	0396
ISSUER	/C=CH/O=CERN/CN=lcg-voms.cern.ch
OCTET STRING	/dteam/Role=NULL/Capability=NULL
OCTET STRING	/dteam/ne/Role=NULL/Capability=NULL
OBJECT	No revocation available
AuthorityKeyIdentifier	0....H....0.....<3...#..
SignatureAlgorithm	md5WithRSAEncryption

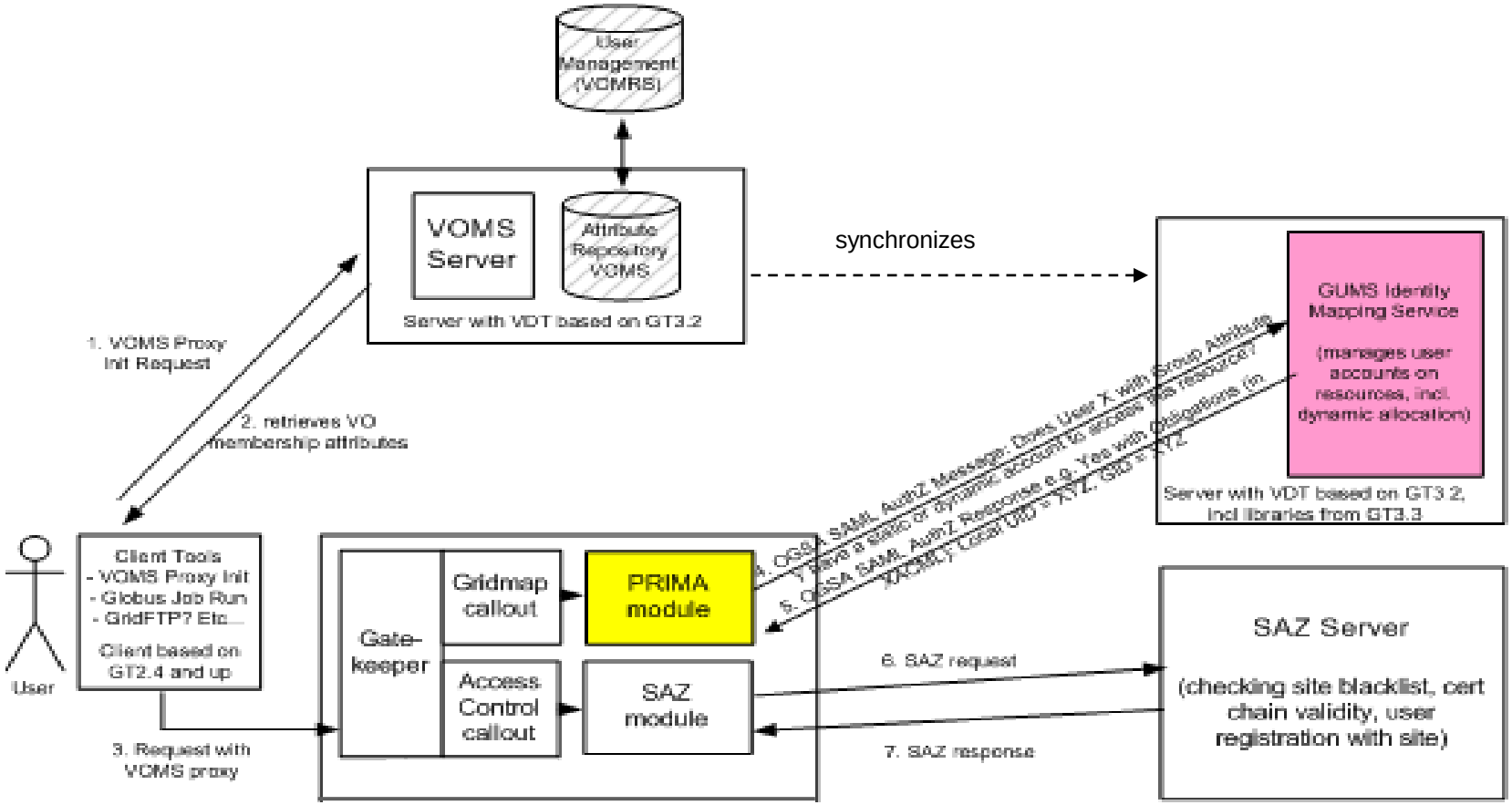


VOMS model


Identity Certificate
/O=dutchgrid/O=users/O=nikhef/CN=Wouter
AuthN Authority/CA
Identity Certificate
/C=CH/O=CERN/CN=lxb2341.cern.ch
VO member
(scientist)
Registration (once)
Resource Provider
VO Administrator
VOMS
signing server
prove identity to VOMS server
sent by the VO admin
via trusted means
VO membership
assertion
VOMS proxy
verify
'VOMS' assertion embedded in proxy conveyed to resource
Grant access to resource
and optionally do
rights, Unix account,
mappings

GUMS model

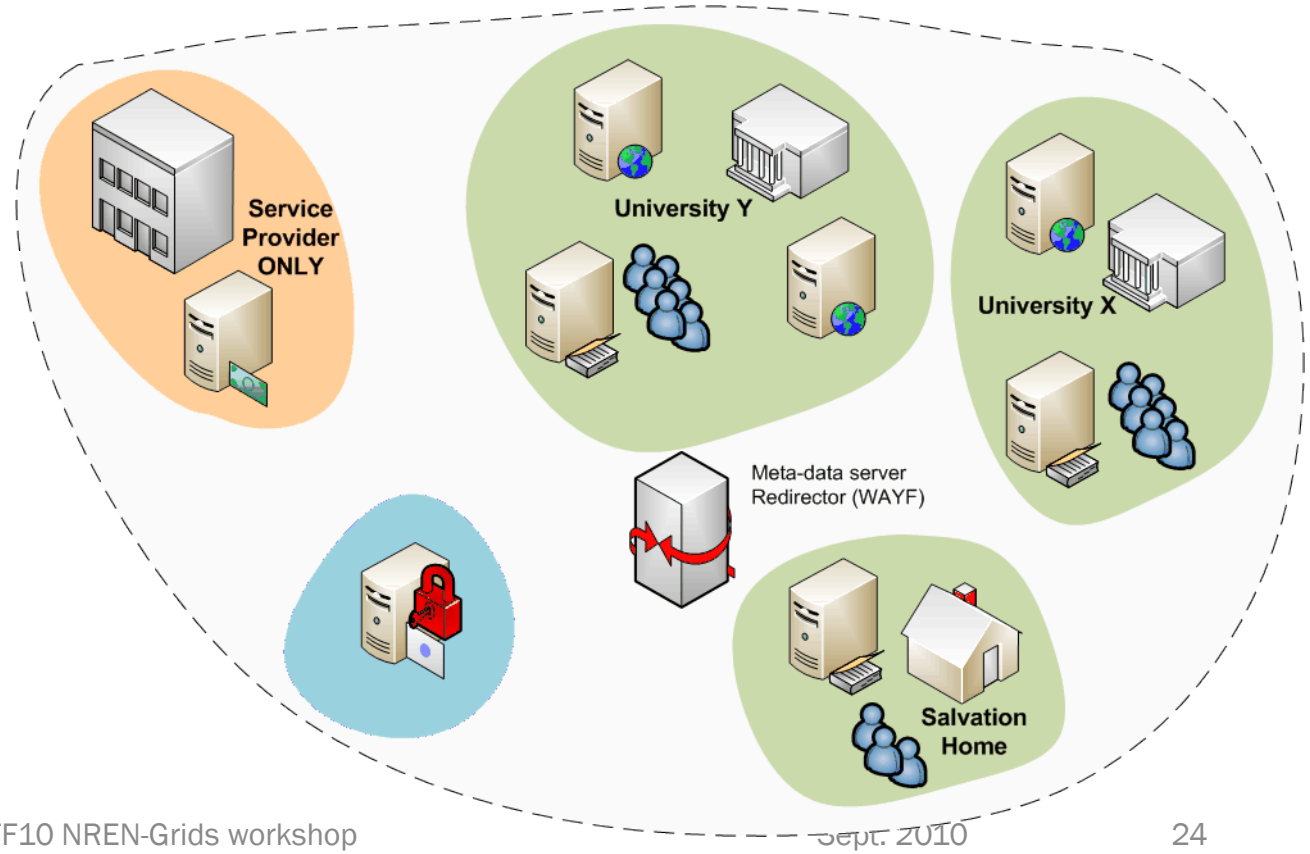
- VO configuration replicated locally at the site
- Here, pushed VOMS attributes are advisory only



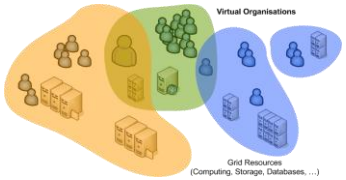
Graphic: Gabriele Garzoglio, FNAL

Attributes from many sources

- In ‘conventional’ grids, all attributes assigned by VO
- but there are many more attributes, and some of these may be very useful for grid



grid structure was not too much different!



Towards a multi-authority world (AAI)

Interlinking of technologies can be done at various points

1. Authentication: linking (federations of) identity providers to the existing grid AuthN systems
   - ‘Short-Lived Credential Services’ translation bridges
2. Populate VO databases with UHO Attributes
3. Equip resource providers to also inspect UHO attributes
4. Expressing VO attributes as function of UHO attributes

> *and most probably many other options as well …*

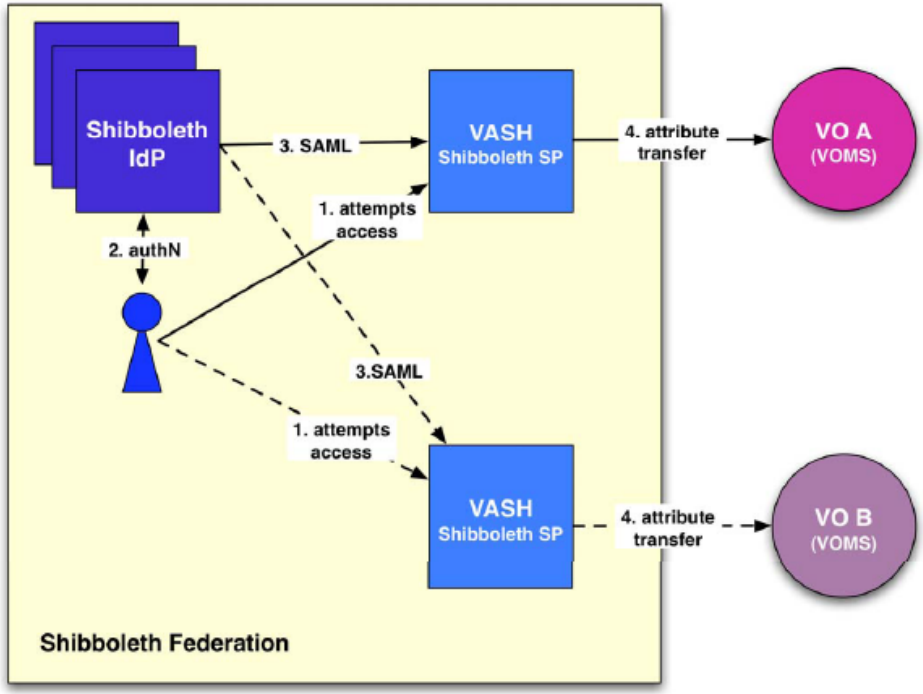
Leads to assertions with multiple LoAs in the same decision

- thus all assertions should carry their LoA
- expressed in a way that’s recognisable
- and the LoA attested to by ‘third parties’ (e.g. the federation)

Attributes from multi-authority world

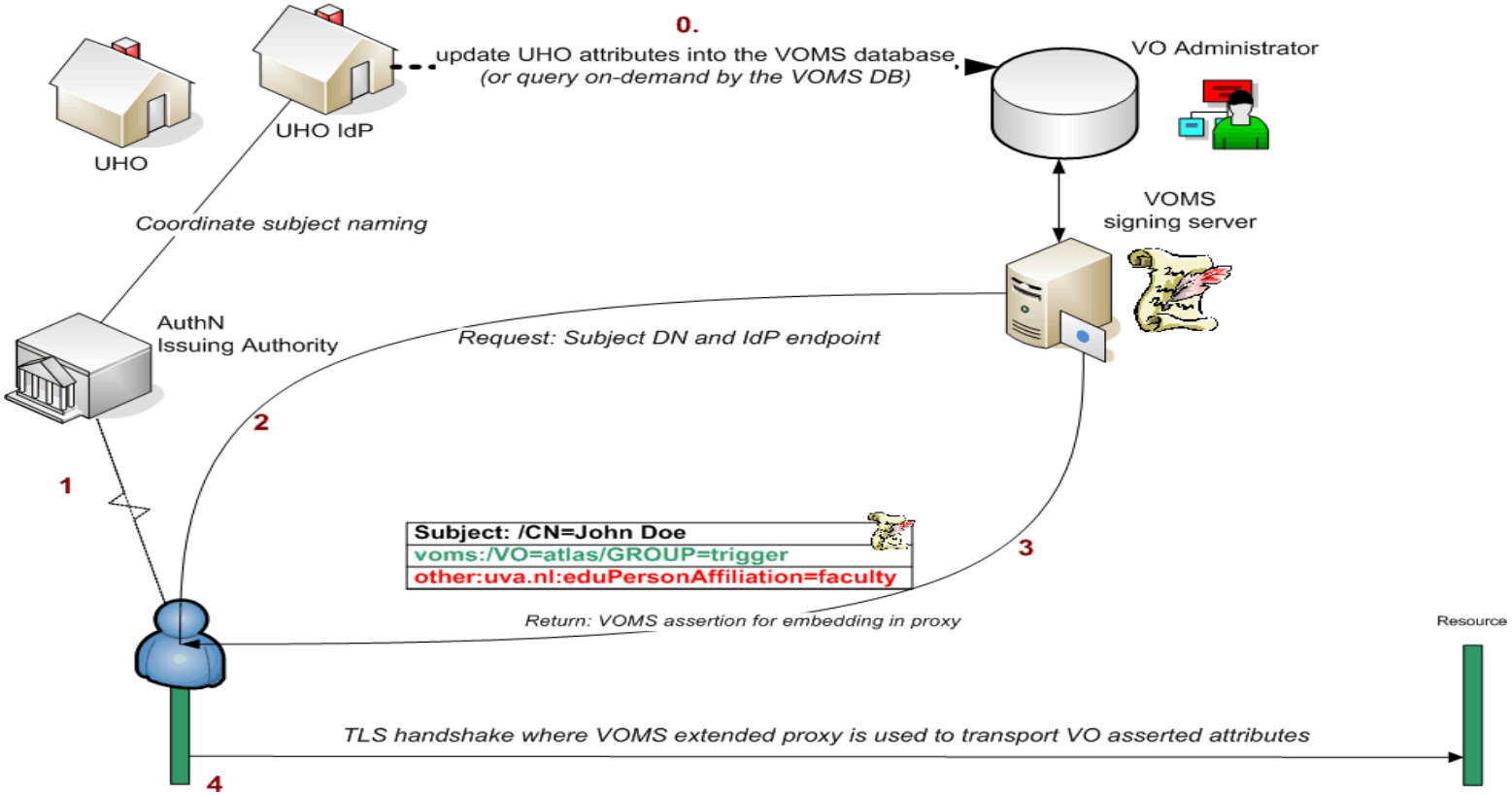
- Linking two worlds example –
- VASH: 'VOMS Attributes from Shibboleth'
  - Populate VOMS with generic attributes
  - Part of gLite (SWITCH)

http://www.switch.ch/grid/vash/



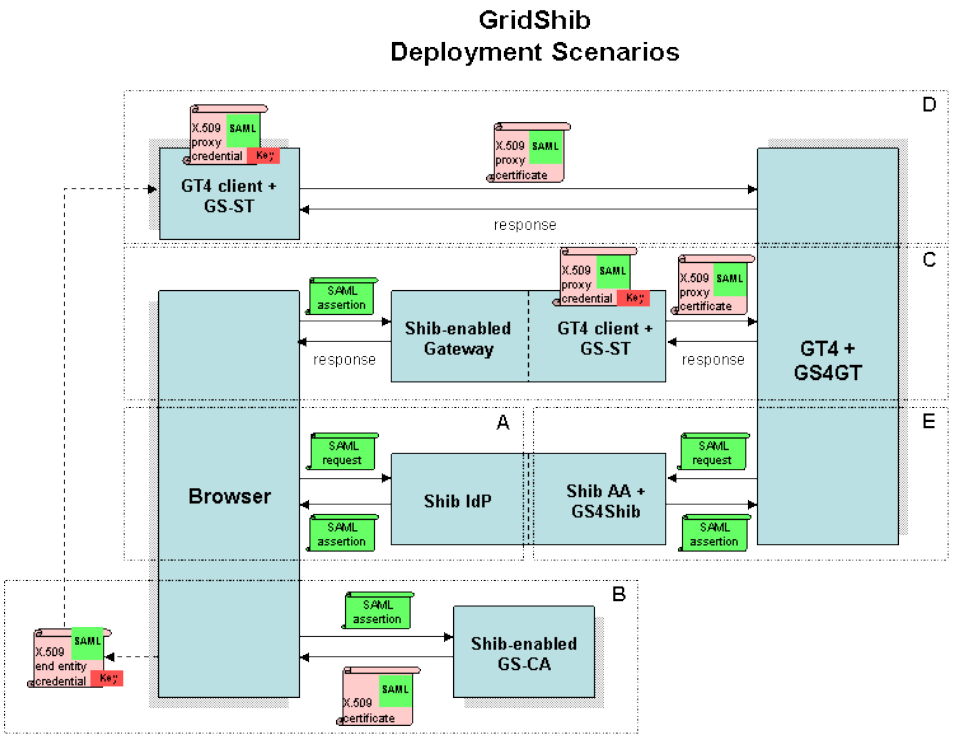
Graphic: Christoph Witzig, SWITCH

Putting home attributes in the VO

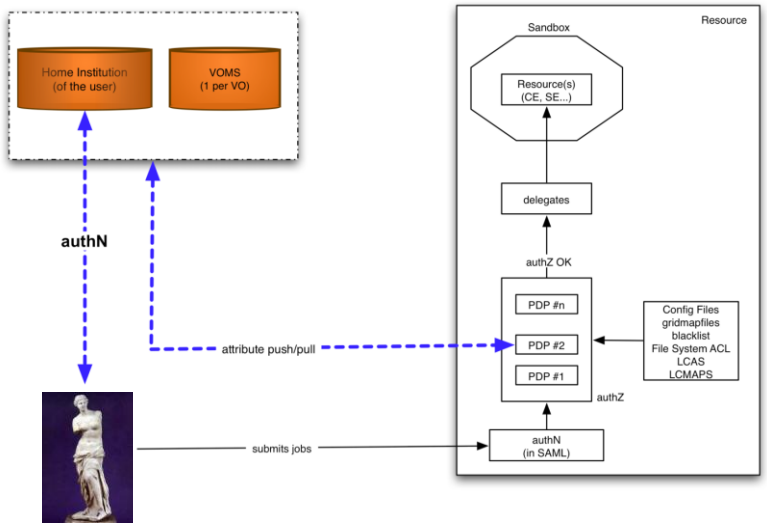


- Characteristics
  - The VO will know the source of the attributes
  - Resource can make a decision on combined VO and UHO attributes
  - but for the outside world, the VO now has asserted to the validity of the UHO attributes – over which the VO has hardly any control

Attribute collection ‘at the resource’



Graphic: the GridShib project (NCSA)
http://gridshib.globus.org/docs/gridshib/deploy-scenarios.html



graphic from:
Chistoph Witzig, SWITCH, GGF16, February 2006

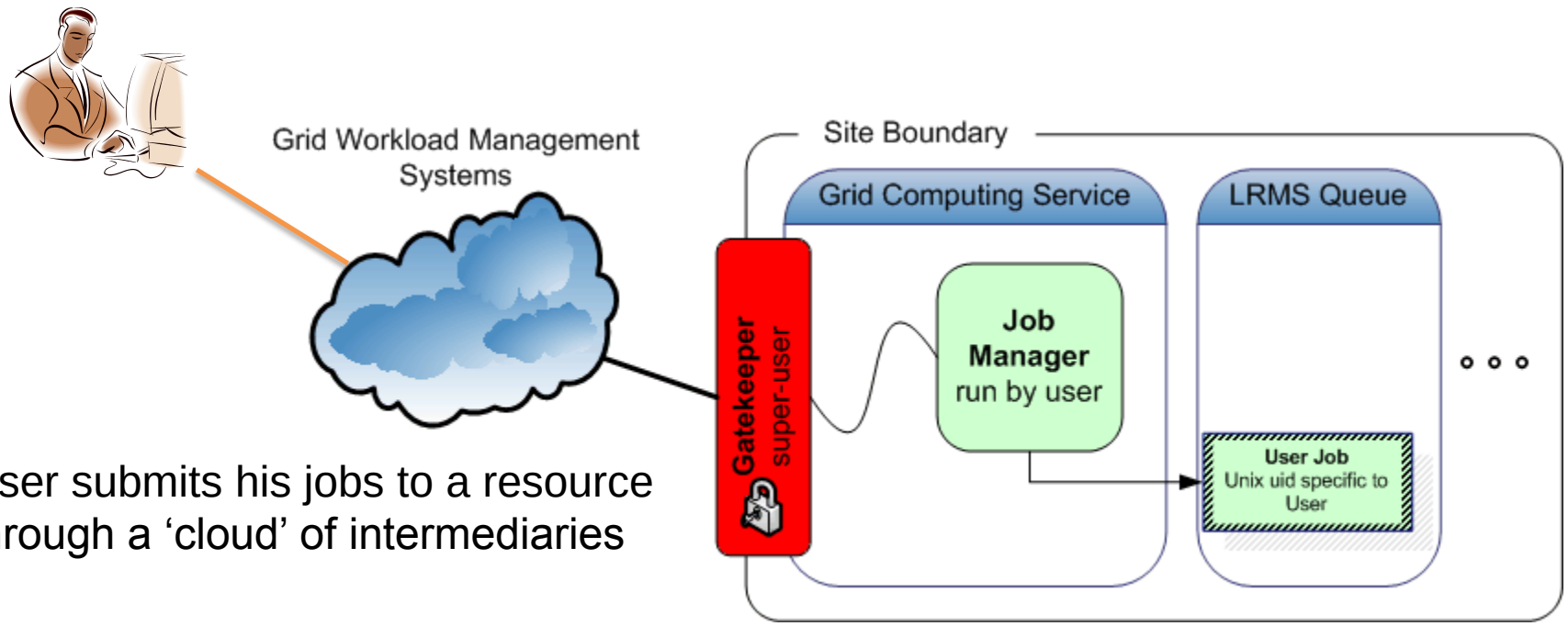
- Characteristics
  - The RP (at the decision point) knows the source of all attributes
  - but has to combine these and make the ‘informed decision’
  - is suddenly faced with a decision on quality from different assertions
  - needs to push a kind of ‘session identifier’ to select a role at the target resource

Example: running compute jobs

The Meaning of Attributes: Unix domain mapping

ACCESS CONTROL FOR COMPUTE

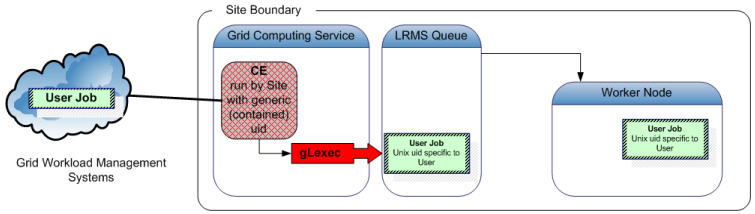
Job Submission Today



User submits his jobs to a resource through a ‘cloud’ of intermediaries

Direct binding of payload and submitted grid job

- job contains all the user’s business
- access control is done at the site’s edge
- inside the site, the user job should get a specific, site-local, system identity



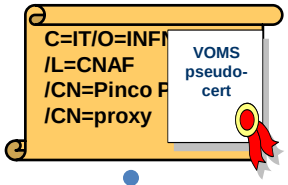
But basic yes-no does not get you far

- *If yes, what are you allowed to do?*
  - Credential mapping via obligations, e.g. unix account, to limit what a user can do and disambiguate users
  - Intended side effects: allocating or creating accounts … or virtual machines, or …
  - Limit access to specific (batch) queues, or specific systems

- Additional software needed
  - Interpreting policy and constraints
  - Handling 'obligations' conveyed with a decision
  - e.g.
    LCMAPS: account mappings, AFS tokens, Argus call-out
    Argus: pluggable obligation handlers per application
    - and interpret (pre-provisioned) policies applicable to a transaction/credential

To the Unix world: Problem

Identity

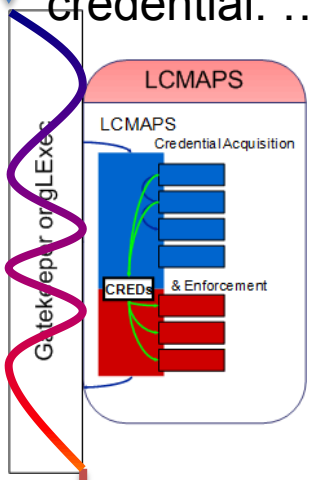


VOMS + other attributes

translate

```
pvier001:x:43401:2029:PoolAccount VL-e P4 no.1:/home/pvier001:/bin/sh
```

run as **root**
credential: …/CN=Pietje Puk



run as target user
uid: ppuk001
uidNumber: 96201

- Unix does not talk Grid, so translation is needed between grid and local identity

1. translation has to happen somewhere
2. something needs to do that

What does this all mean?

- Attribute interpretation is much more than mere mapping
  - what do the attributes mean, and do all VOs mean similar things with the same kinds of attributes?
  - Is the order in which the attributes are presented important?
  - Can the same bag of attributes (or same priority) be used for both compute and data access?
  - How do changing attributes reflect access rights on persistent storage, if the VO evolves its attribute set?

- Is there a driving use case by RPs (VO, sites) for an attribute?
  - only then makes harmonization any sense…
- Let RPs (co-)define requirements, not only IdPs, CAs, or VOs!
  - attributes and policies needed, and the meaning of attributes
  - levels of assurance

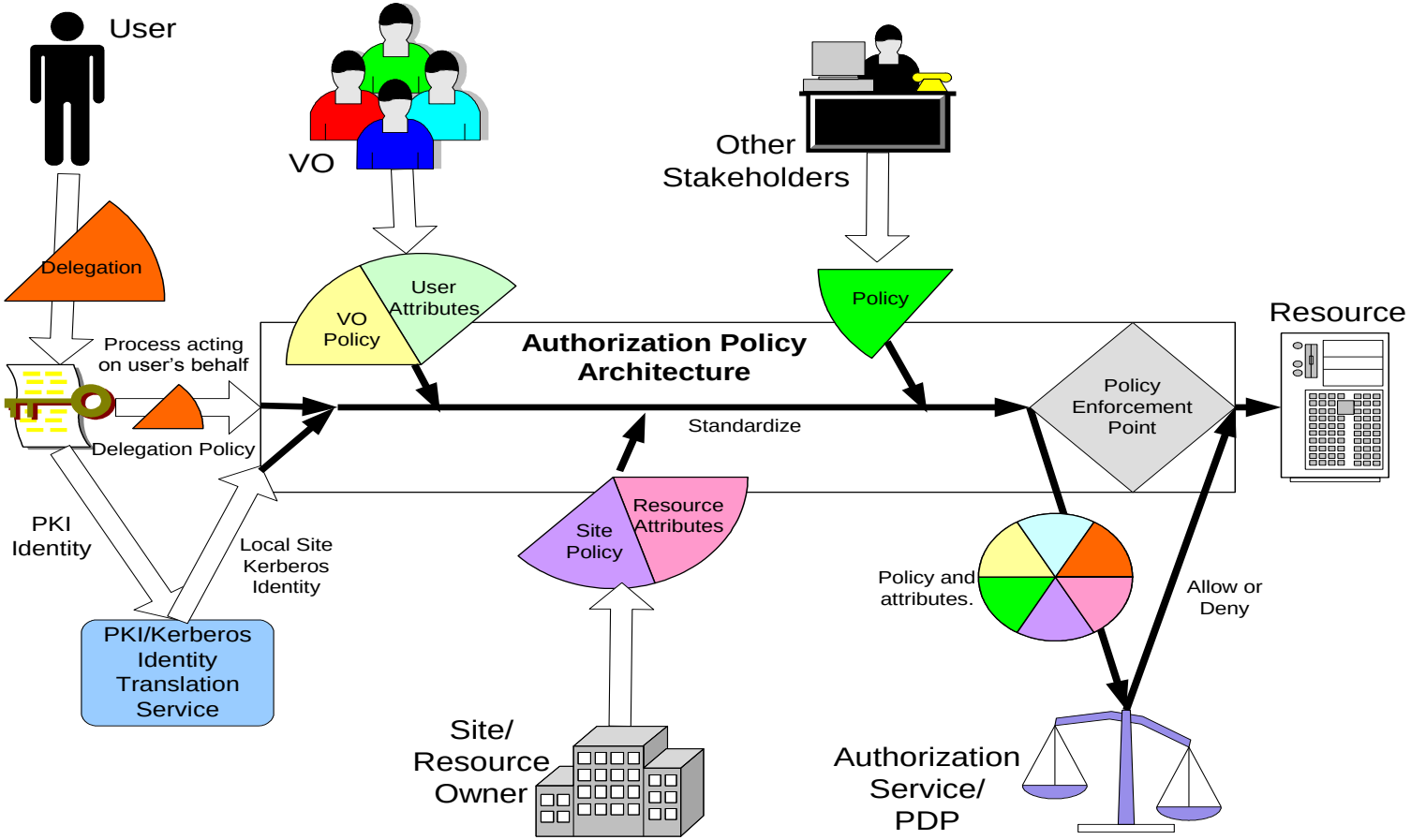
Policy from multiple sources

Frameworks

AUTHORIZATION FRAMEWORKS

A multi-authority world

- Authorization elements (from OGSA 1.0)



Graphic: OGSA Working Group

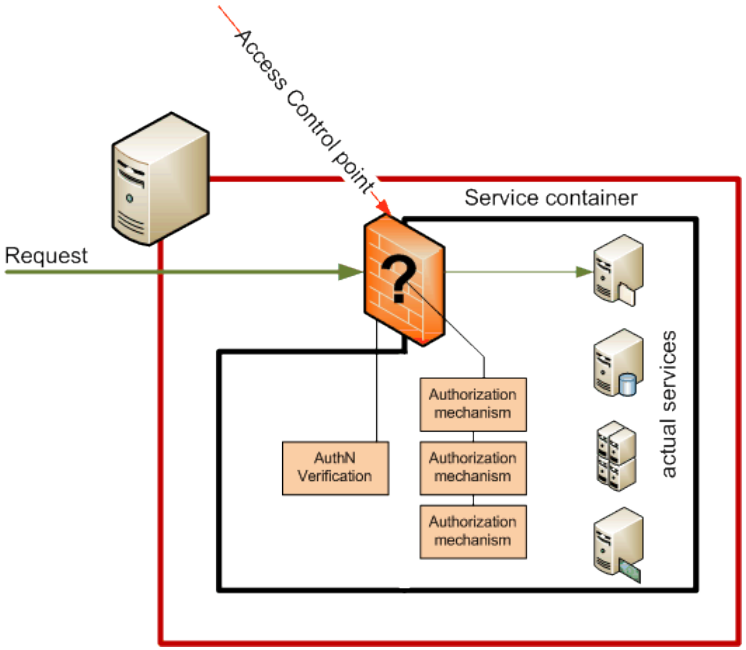
Control points

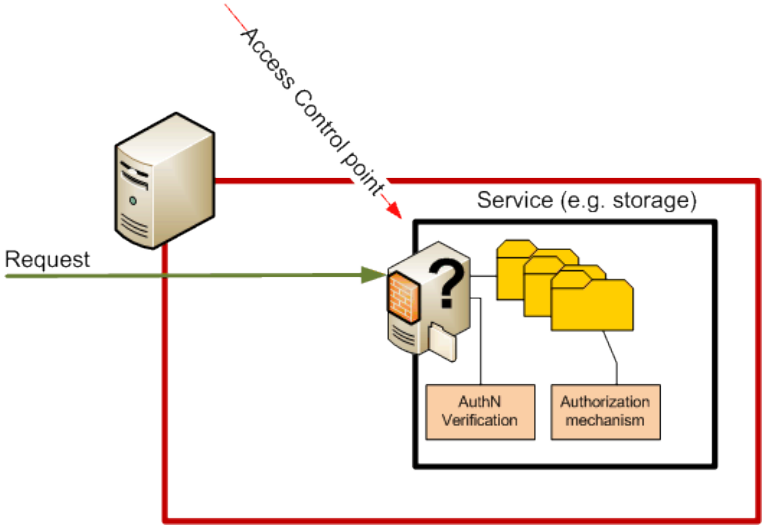
Container based

- Single control point
- Agnostic to service semantics

In-service based

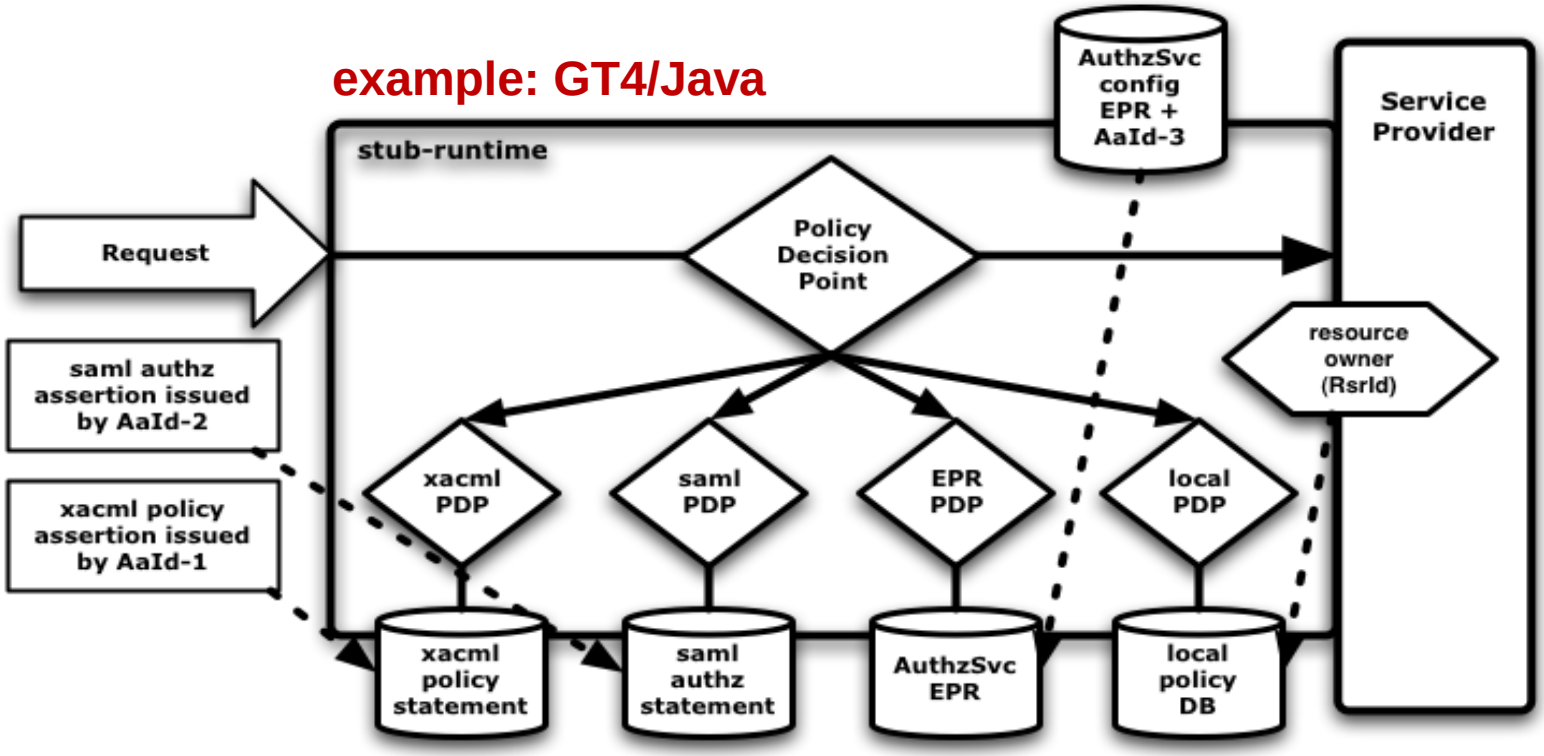
- Many control points
- Authorization can depend on requested action and resource





Frameworks

- (chain of) decision making modules controlling access
  - Loosely or tightly coupled to a service or container
  - Generic ‘library’, or tied into the service business logic



Graphic: Frank Siebenlist, Globus and ANL

Example framework implementations

- PRIMA-SAZ-GUMS-gPlazma suite
- Globus Toolkit Authorization Framework
- Site Access Control 'LCAS-LCMAPS' suite
- gLite Argus
- GridSite & GACL
- …

interop

interop

… and don't forget 'native' service implementations

Different frameworks

- Each framework has
  - own calling semantics (but may/will interoperate at the back)
  - its own form of logging and auditing

- Most provide
  - Validity checking of credentials
  - Access control based on Subject DN and VOMS FQANs
  - Subject DN banning capability
- And some have specific features, e.g.,
  - Capability to process arbitrary 'XACML' (composite) policies
  - Calling out to obtain new user attributes
  - Limiting the user executables, or proxy life time, …
  - allow embedding inside the application business logic

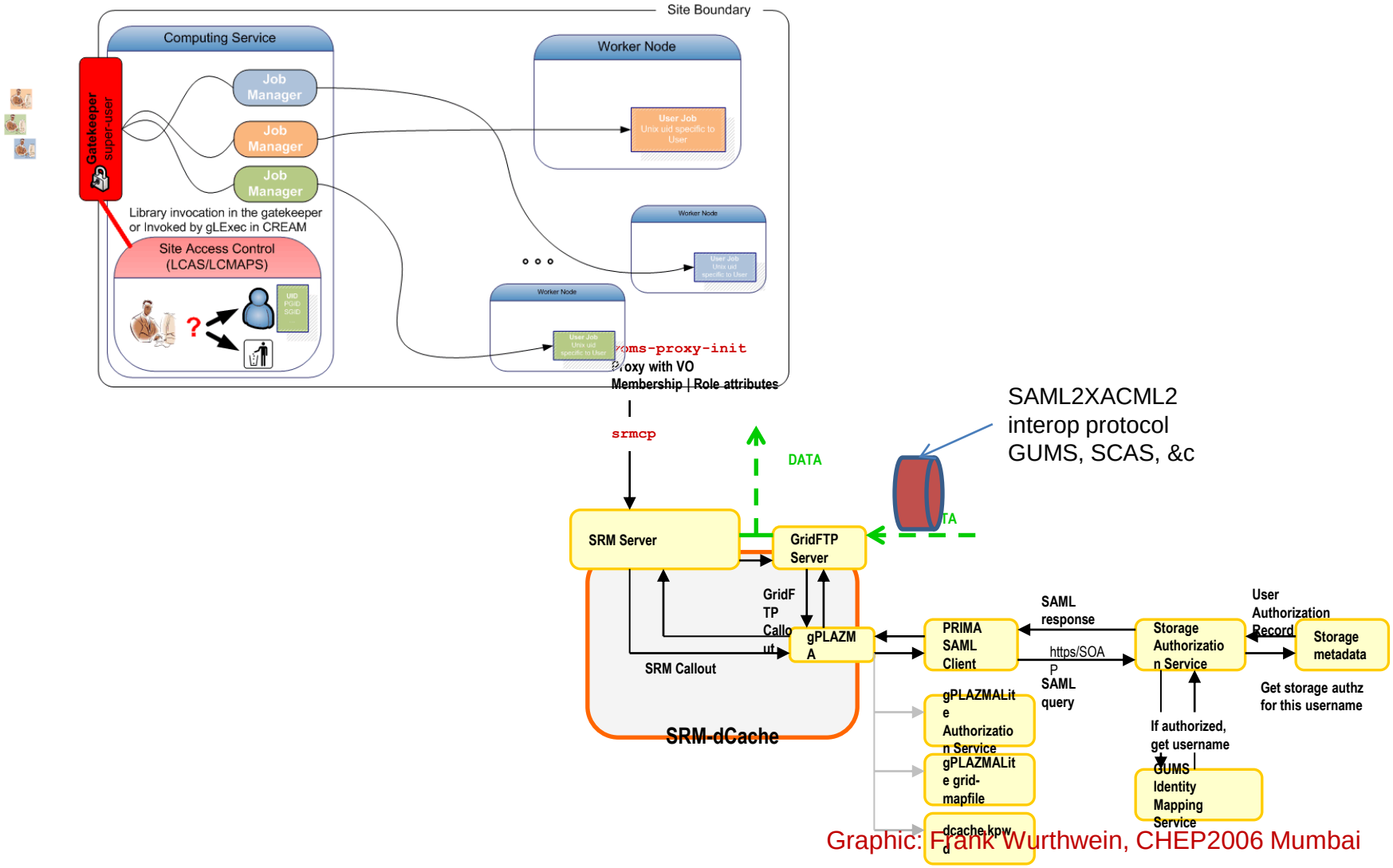
Centralizing Authorization in the site

Available middleware: GUMS and SAZ, Argus, …

Interoperability through common protocols

ACCESS CONTROL MANAGEMENT SYSTEMS

Embedded controls: CE, dCache, …



Graphic: Frank Wurthwein, CHEP2006 Mumbai

Access Control at the Service

Most prevalent solution today …

Pros:

- services independent and have no common failure mode
- quick and easy to develop and deploy

Con:

- no single ‘Big Red Button’
- difficult auditing…
- risk of inconsistency

Centralizing decentralized Access Control

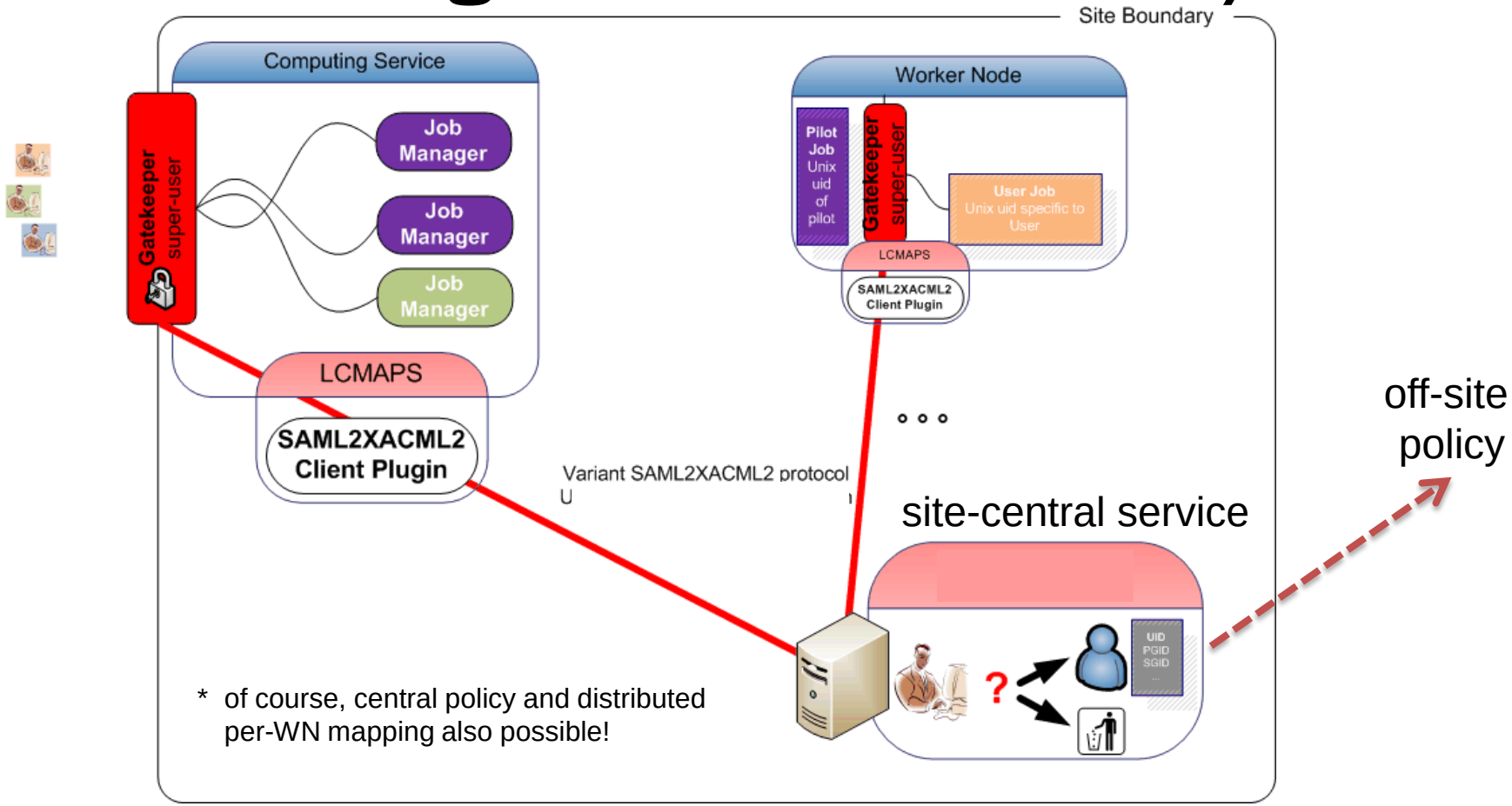
Aim: support consistently

- policy management across services
- quick banning of bad users
- coordinated common user mappings (if not WN-local)

Different options to implement it …

- Regular site management tools (CFengine, Quattor, etc)
  - Addresses site-wide banning in a trivial and quick way
  - but appears 'out of band' and works only for managed installations
- One of the 'central authorization services'
  - these can be department-central, site-central, but even grid-wide or global!
  - some to choose from in Grid: Argus, GUMS, …
  - like 'inverse' IdP, centrally processing assertions for AuthZ instead of making …

Centralizing access control in M/W

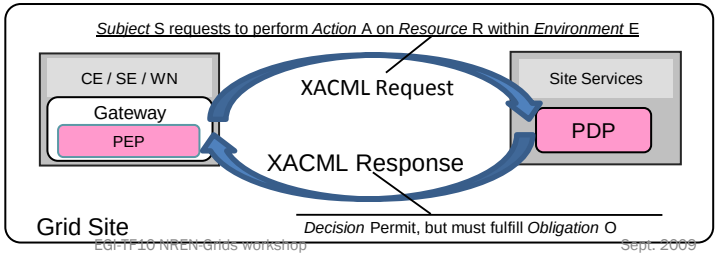


* of course, central policy and distributed per-WN mapping also possible!

Key Elements for interop

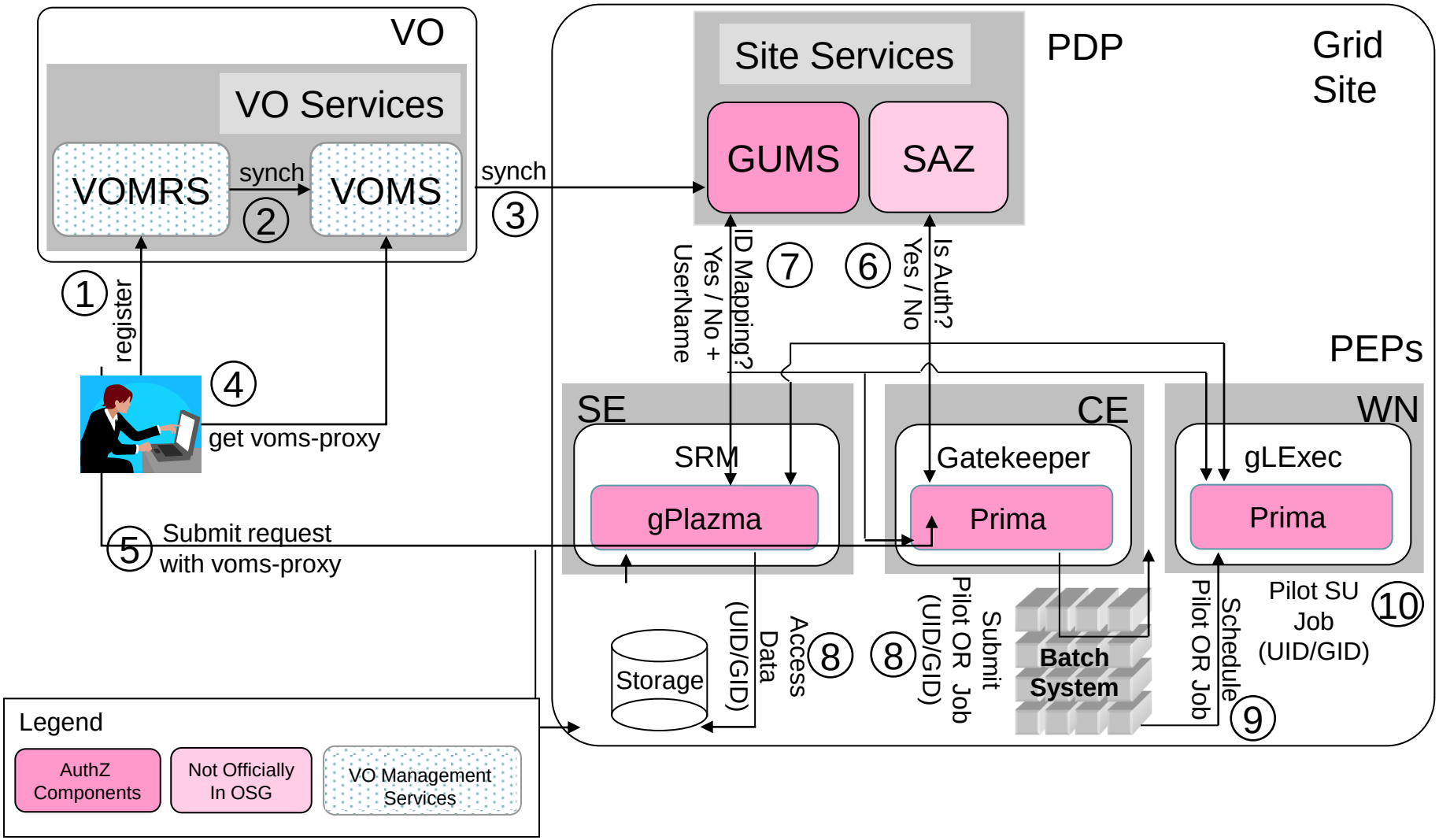
- Common *communications* profile
  - Agreed on use of SAML2-XACML2
  - http://www.switch.ch/grid/support/documents/xacmlsaml.pdf

Graphic: Gabriele Garzoglio, FNAL



- Common *attributes and obligations* profile
  - List and semantics of attributes sent and obligations received between a 'PEP' and 'PDP'
  - Now at version 1.1
  - http://cd-docdb.fnal.gov/cgi-bin/ShowDocument?docid=2952
  - http://edms.cern.ch/document/929867

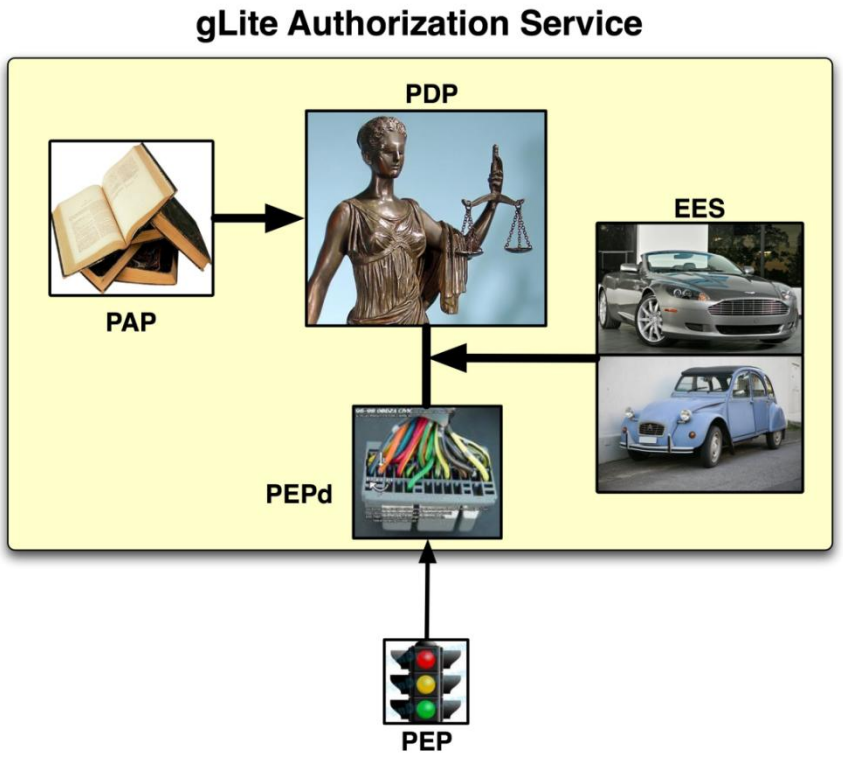
GUMS and SAZ



graphic: Dave Dykstra, Fermi National Accelerator Laboratory, CHEP, March 2009

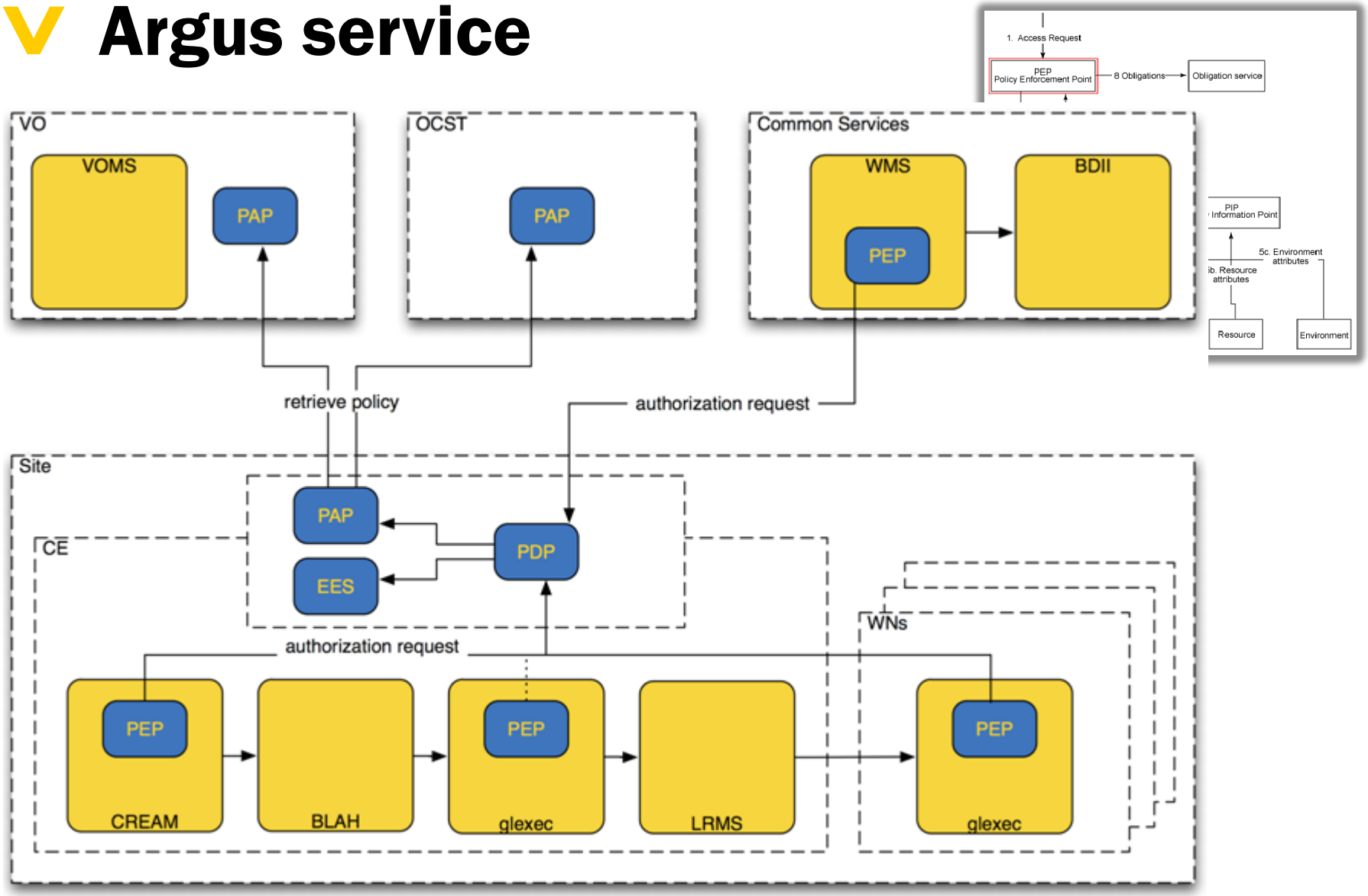
Argus services and daemons

- Administration Point
  Formulating rules through CLI and/or file-based input
- Decision Point
  Evaluating a request from a client based on the rules
- Enforcement Point
  Thin client part and server part: all complexity in server part
- Runtime Execution Environment
  Under which env. must I run? (Unix UID, GID, ...)



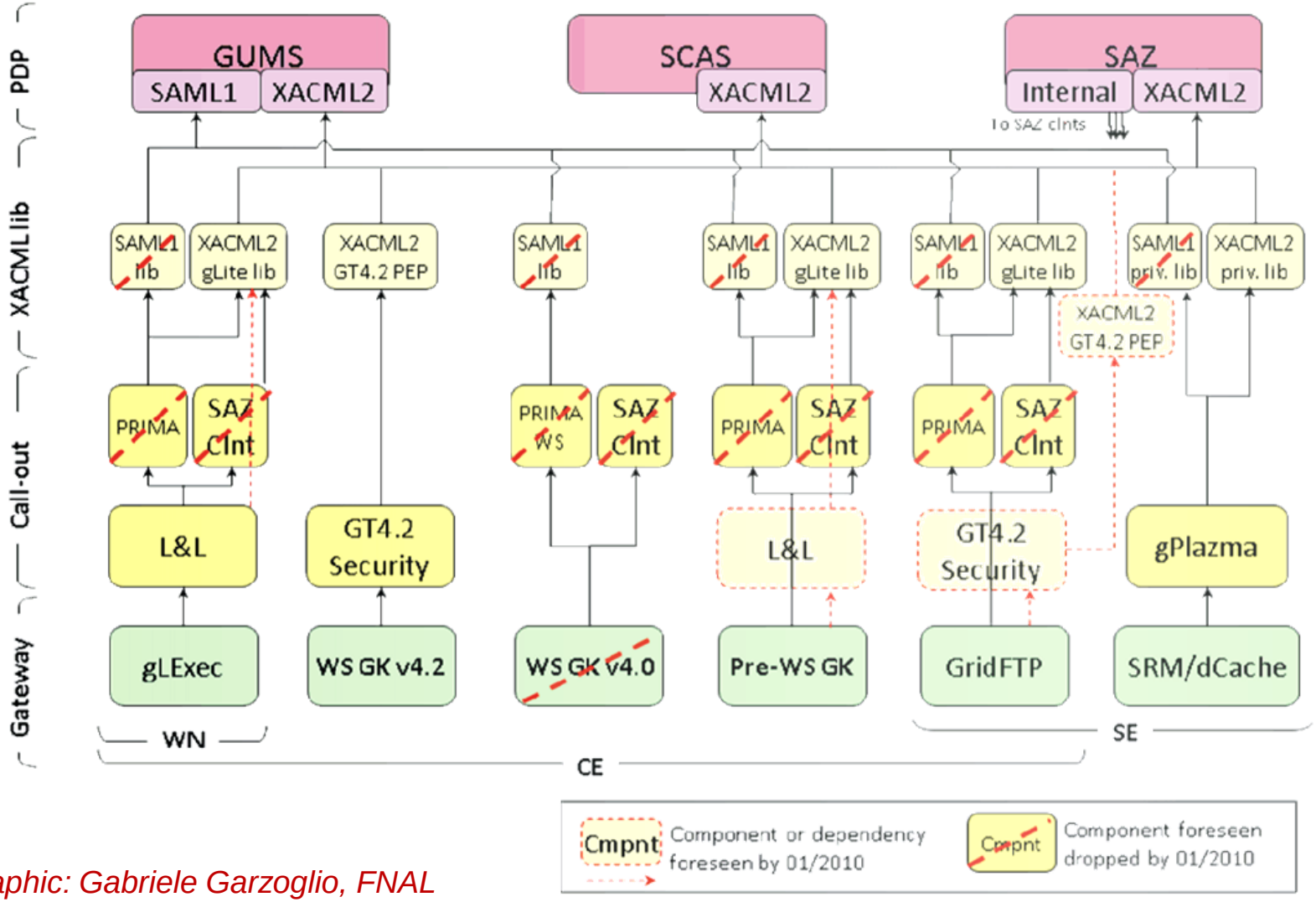
Graphic: Christoph Witzig, SWITCH and EGEE

Argus service



graphic: MJRA1.4 (EGEE-II) gLite security architecture, Oct 2008, Christoph Witzig

Interoperability achievements



graphic: Gabriele Garzoglio, FNAL

Capabilities (Argus as an example)

- Enables/eases various authorization tasks:
  - Banning of users (VO, WMS, site, or grid wide)
- Composition of policies – e.g.
  CERN policy + experiment policy + CE policy
  + OCST policy + NGI policy=> Effective policy
- Support for authorization based on more detailed information about the job, action, and execution environment
  - Support for authorization based on attributes other than FQAN
  - Support for multiple credential formats (not just X.509)
  - Support for multiple types of execution environments
  - Virtual machines, workspaces, …

https://twiki.cern.ch/twiki/bin/view/EGEE/AuthorizationFramework

Summary and last words

FROM HERE?

What Grid AAI does for you today

- Accommodates multiple sources for assertions
  - AuthN vs. AuthZ separated, with multiple VO membership off same ID
  - With the 'PKI bits' being cleverly hidden from the user
- Accommodate delegation (disconnected operation)
  - Entities act on behalf of a user
  - services like MyProxy and SLCS make it transparent even for portals and long-running jobs
- Accommodate individual, independent researchers
  - even though federations will aid 99% percent, full coverage will be rare
- EGI demonstrates that the mechanisms **and associated policies and standards** convinced 300+ resource providers grid is trustworthy enough
- Users **actually see a single interface** (VO), and no longer need to register at 100s different sites and fill in 100+ AUP statements … since 2002!

Having left out a lot of things … are there any

QUESTIONS?