Enabling Grids for E-sciencE

SCAS technical

Site Central Authorization Service

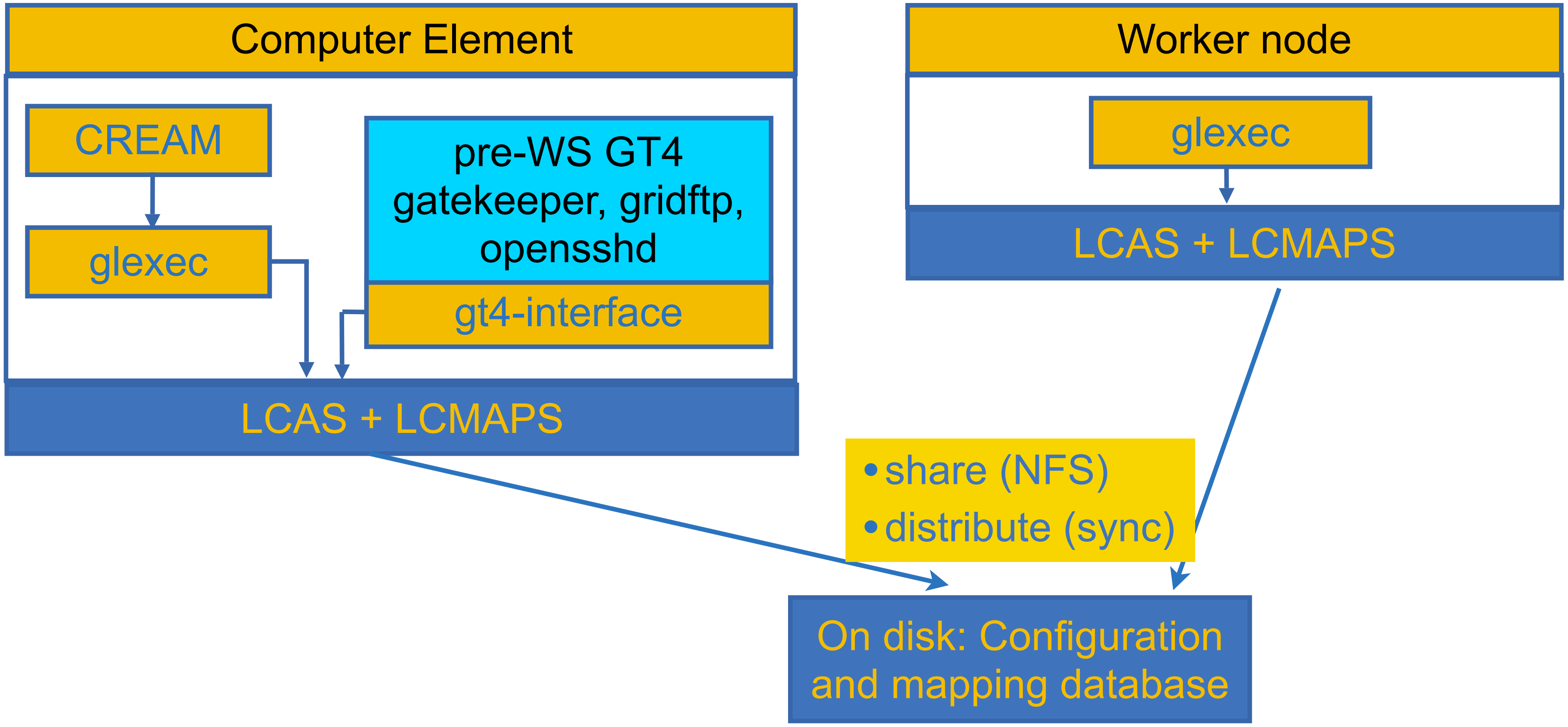
Oscar Koeroo

JRA1

www.eu-egee.org

index

Current setup issues



Issues with this setup:

- share/distribute the **gridmapdir** for mapping consistency
- share/distribute the **configurations** for the nodes
- share/distribute **authorization** files, like **grid/groupmapfiles** and a **blacklisting** file
- **Scaling** issues; lots of active nodes will probably **overload** an NFS server

What is SCAS?

The Site Central Authorization Service

- **It implements a client/server architecture to query:**
  - Authorization decisions (LCAS), allow/ban:
    - From a trusted resource
    - From an authorized pilot job executor
    - For an authorized user
  - Centralized grid identity to Unix ID translations (LCMAPS)
    - Full LCMAPS support
      - *VOMS pool and local accounts mappings*
      - *Non VOMS pool and local account mappings*

- **Uses mutual authenticated SSL/TLS**

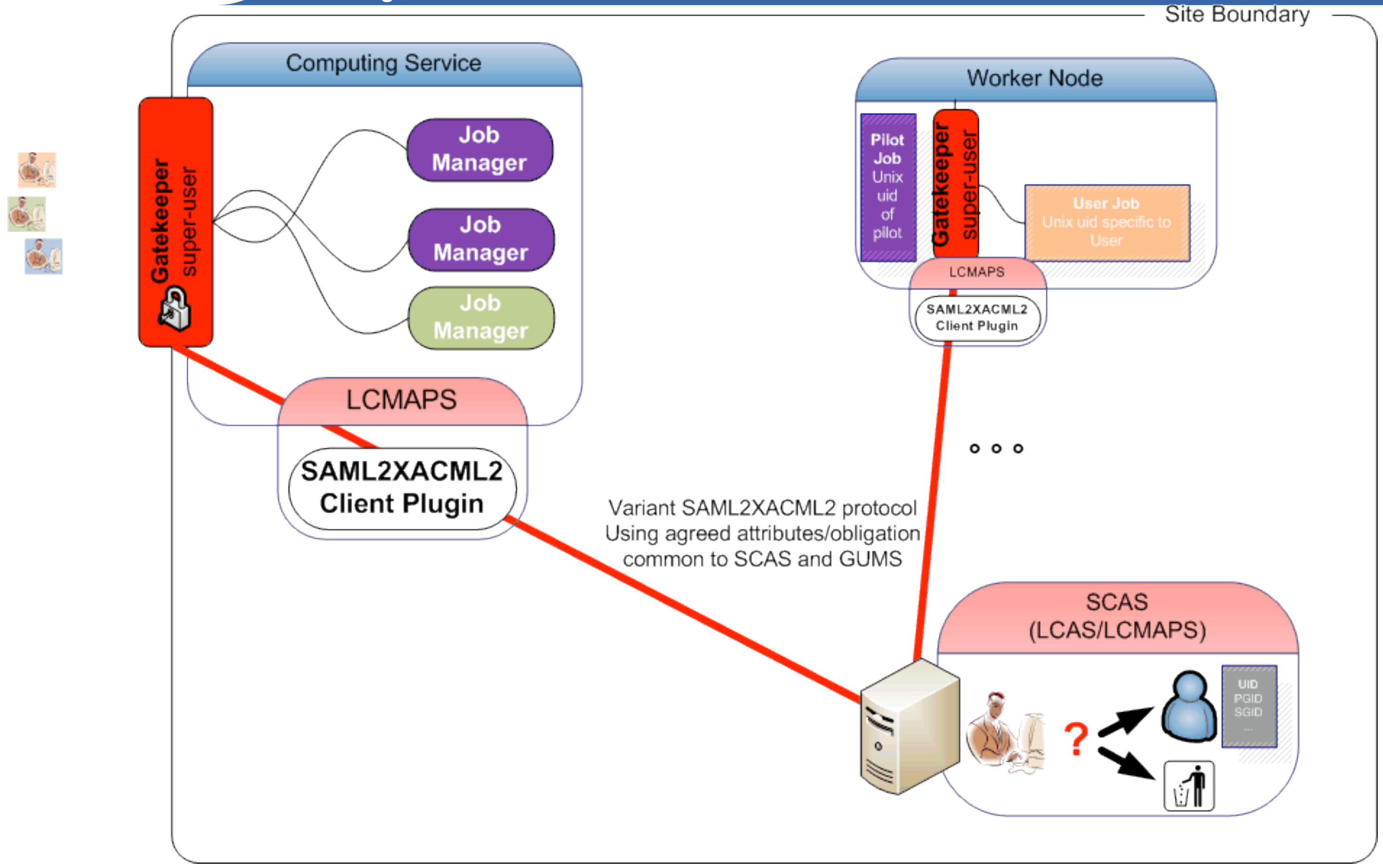
What isn’t SCAS?

It’s not a centralized authentication service (....yet)

... although the option is left open for future investigation

– Clients must authenticate credentials before it goes on the wire:
  - Requirement on clients:
    - *CA certificates and VOMS authorization files (.lsc)*

Site-central access control



PRO single unique account mapping per user across whole farm, CE, and SE
can do instant banning and access control in a single place
protocol profile allows interop between SCAS and GUMS (but no others!)

CON replicated setup for redundancy needed for H/A sites
still cannot do credential validation (formalistic issues with the protocol)

Why SAML2-XACML2?

- **The protocol is flexible**
  - by adding Attributes and/or by adding Obligations
- **Obligation handling semantics:**
  - “Returned Obligations must be handled” or fail...
- **OSG / Privilege**
  - Use a ‘patched’ SAML based protocol for GUMS. In the race for something more standards compliant (already in contact with Globus dev team)
  - Heavy gLExec users with the wish to connect natively to GUMS
- **We shared lots of commonalities in our use cases for our site central solution**
  - Must be separate from the existing Globus Toolkit (done)

SAML-XACML lib requirements

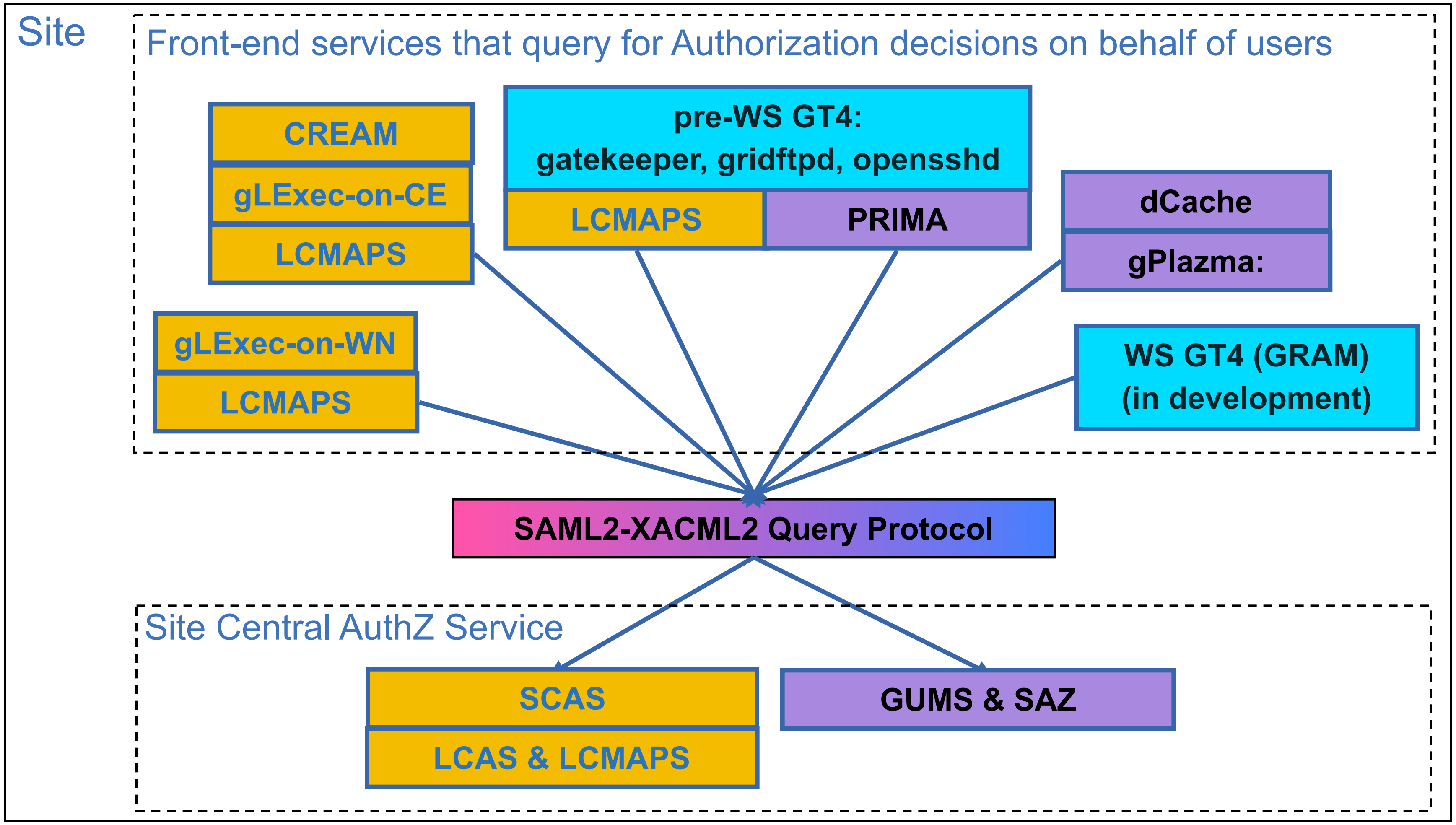
- **Requirements to SCAS dev:**
  - Easy interoperation
    - Understand a common set of obligations and its attributes
  - Scalability
    - Low network traffic
    - Low overhead at the end points
  - Keeping compatibility with existing LCAS and LCMAPS plug-ins and their functionalities

- **Requirements to Globus:**
  - Must be separate from the existing Globus Toolkit (low dependency overhead)

Interoperable components

Legenda: Color code indicates component developers:

Globus	EGEE	OSG / Privilege Project	Globus, EGEE, OSG / Privilege Project


Site
Front-end services that query for Authorization decisions on behalf of users
CREAM
gLExec-on-CE
LCMAPS
pre-WS GT4:
gatekeeper, gridftpd, opensshd
LCMAPS
PRIMA
dCache
gPlazma:
gLExec-on-WN
LCMAPS
WS GT4 (GRAM)
(in development)
SAML2-XACML2 Query Protocol
Site Central AuthZ Service
SCAS
LCAS & LCMAPS
GUMS & SAZ

The C implementation

- **Globus provided the SAML2-XACML2 library:**
  - Implements the parser and the message handling (and more)
  - Also hooks to trigger obligation handlers
- **Work done on my end:**
  - Implemented the SSL/TLS layer by the exposed socket hooks
  - Helper functions
    - Registration of the supported obligations with obligation handlers
    - Adding the registered obligations into the request message declared as supported obligations
  - The code went in
    - SCAS service
    - LCMAPS plugin SCAS client
    - PRIMA component

The Query Protocol: Request (0)

- **Setups up SSL/TLS connection to authz service:**
  - SCAS:
    - From CE/SE: Host credential
    - From WN (pilot job): Pilot job executor credential
  - GUMS
    - Using host credential in all cases (CE/SE/WN)

The Query Protocol: Request (1)

- **Request message payload to authz service:**
  - Subject
    - User info for who is the authorization request
  - Action
    - Send a queue-able job, execute now (fork/glexec) or accessfile
  - Resource
    - From which (type of) node (WN, CE, SE, RB) and host id (if avail.)
  - Environment
    - Advertise PEP supported obligation handlers
    - Job invoker (replicate subject) and type (could be unprivileged Condor daemon or pilot)

Difference SCAS & GUMS (0)

- **SCAS:**
  - Pilot job request is authorized at the SCAS service
    - at the SSL handshake by LCAS
    - using the regular set of LCAS plugins (VOMS enabled)
- **GUMS:**
  - Pilot job request is authorized in GUMS
    - In the database by fetching the pilot job ID out of the Environment section

The Query Protocol: Response

- **Basic: Yes/No**
- **EGEE Obligations:**
  - UID + GID
  - Optional multiple secondary GIDs
  - Optional AFS token (type string)
- **VO Services Obligations:**
  - Username (for CE)
  - RootPath + HomeDir (gPlazma)
  - Priorities (gPlazma)
  - File creation mask + directory creation mask

Difference SCAS & GUMS (1)

- **SCAS:**
  - Returns the obligations UID+GID and Secondary GIDs
    - SCAS specifies the mapped account based on the **numerical** representation of the Unix account and the Unix groups
- **GUMS:**
  - Returns the obligation Username by default
    - GUMS specifies the mapped account based on the **string value** of the Unix account. The PEP will need to do a lookup of the primary GID and secondary GIDs from the password file.
  - For gPlazma use cases it can return Storage system obligations

Performance (1)

- **7 VMs to one service, hardware:**
  - dual-quad xeons for clients
  - dual Opteron for the SCAS service
- **Goals for the service:**
  - Stability and 6Hz nominal rate authz decisions and mappings
- **Results:**
  - Nominal rate reached: ~11Hz
  - Load:
    - Server side: average ~3.5%, peak ~10%
    - Client side (each): average ~3%
- **The bottleneck is in the network**
  - Bottleneck in the IO is caused by the VM host
    - need more clients...
  - Session caching might lower IO requirement a bit...

Performance (2)

- **15 VMs to one service, hardware:**
  - dual-quad xeons for clients
  - dual Opteron for the SCAS service
- **Goals for the service:**
  - Stability and 6Hz nominal rate authz decisions and mappings
- **Results:**
  - Nominal rate reached: ~24Hz
  - Load:
    - Server side: average ~10%, peak ~13%
    - Client side (each): average ~3%
- **The bottleneck is in the network**
  - Bottleneck in the IO is caused by the VM host
    - need more clients...
  - Session caching might lower IO requirement a bit...

Final words

- **The site central solution allows for improved emergency response**
  - Central blacklist
  - Consistent mappings across a cluster or a site for all the supported services

- **Profiled document on the used attributes:**
  - "An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids"
    - https://edms.cern.ch/document/929867/1

Thanks!

- **Thanks to my SA3 colleagues at Nikhef**
  - Exposed stupid mistakes
  - Helped with performance testing the SCAS service

Tag! Your it.

“Where’s *the* tag?”

Planning

- **Next week:**
  - SCAS service code (tidbits mostly)
    - Tagging code, plus redo test at Nikhef and HIP
  - Fix install notes
- **In two weeks:**
  - *The* tag
  - *The* patch
- **After the patch:**
  - Awaiting comments from SA3

?