SCAS Progress

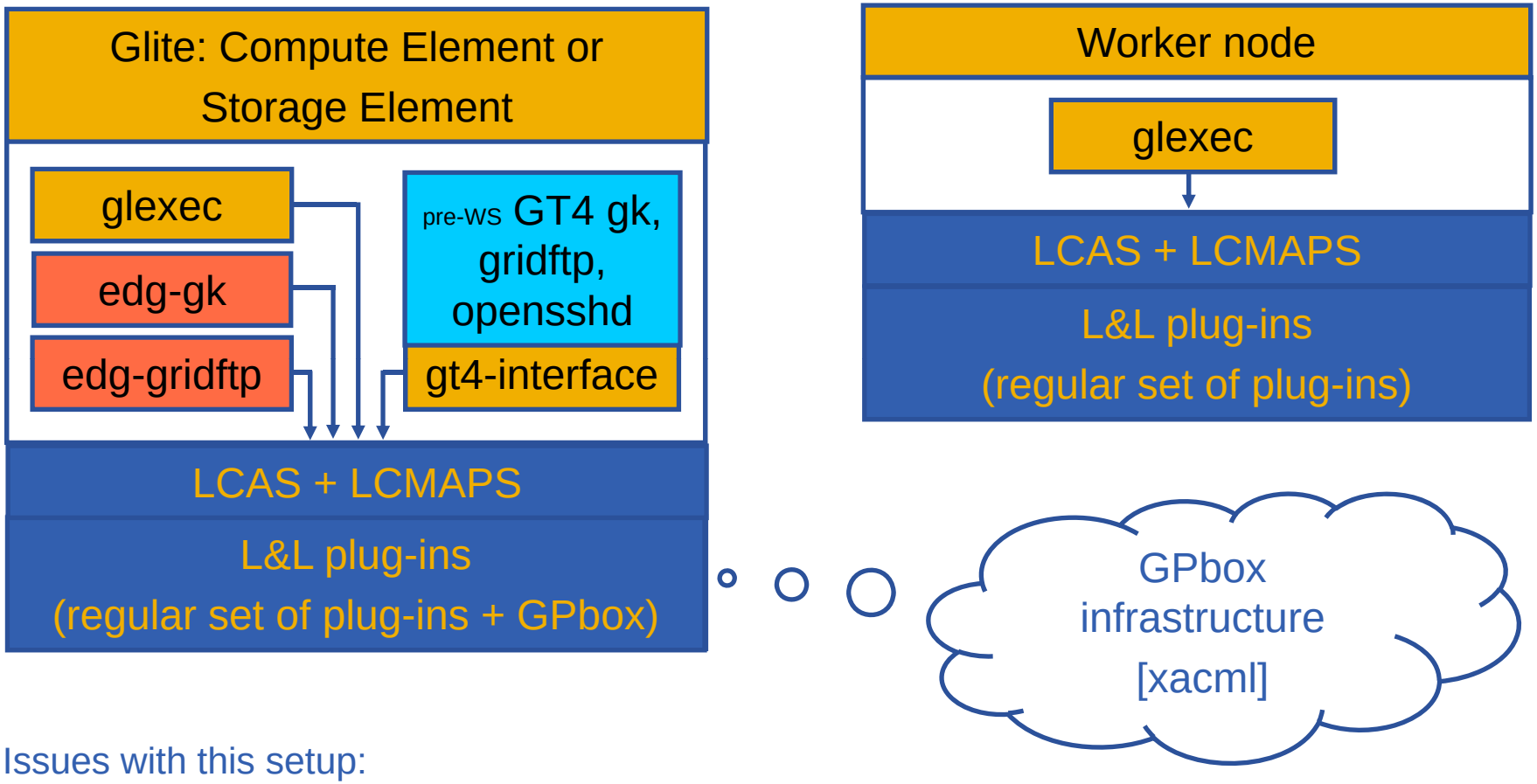
Oscar Koeroo

index

- The architecture
- The paper work
- The implementation

The architecture

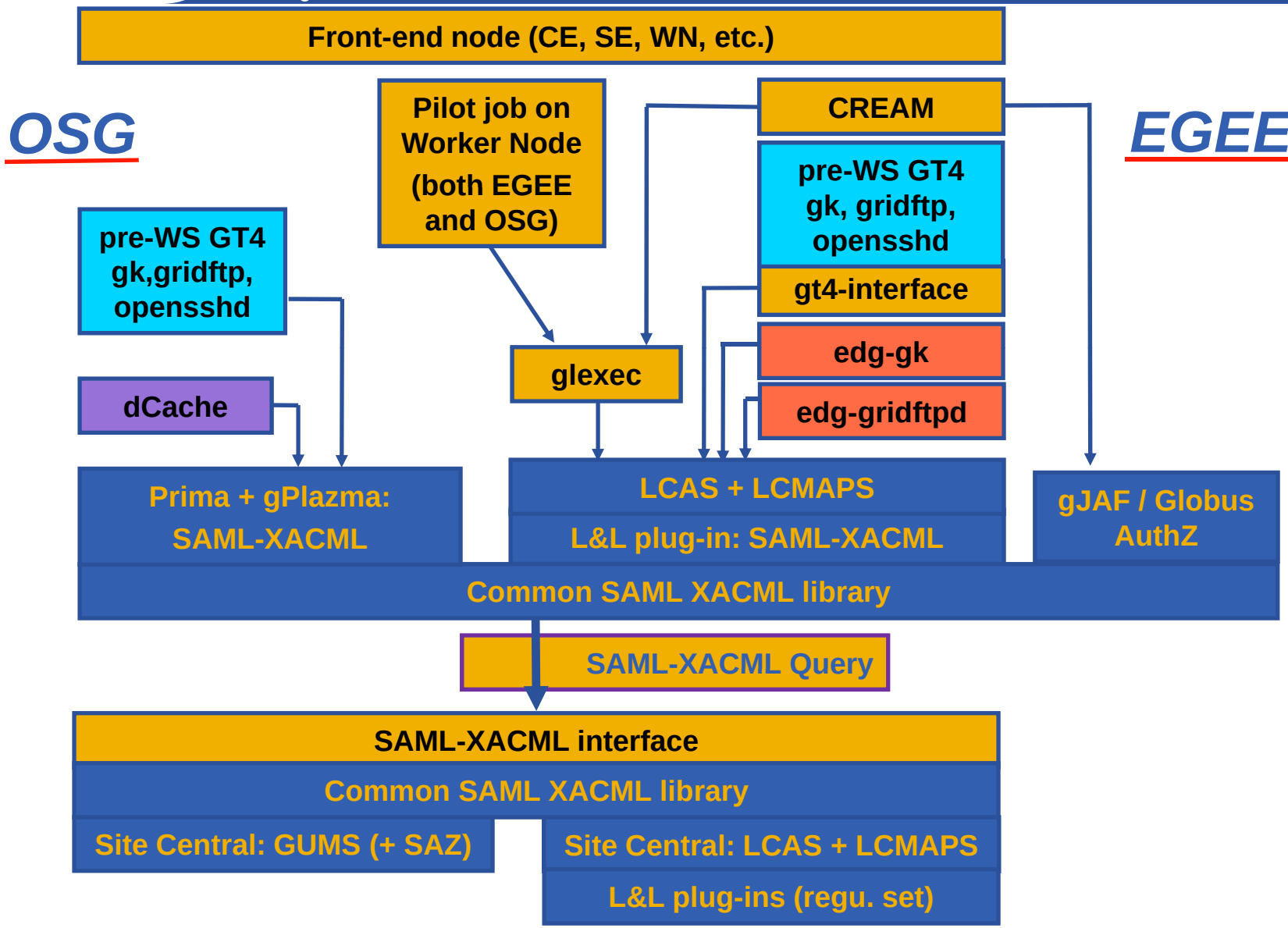
Our current architecture



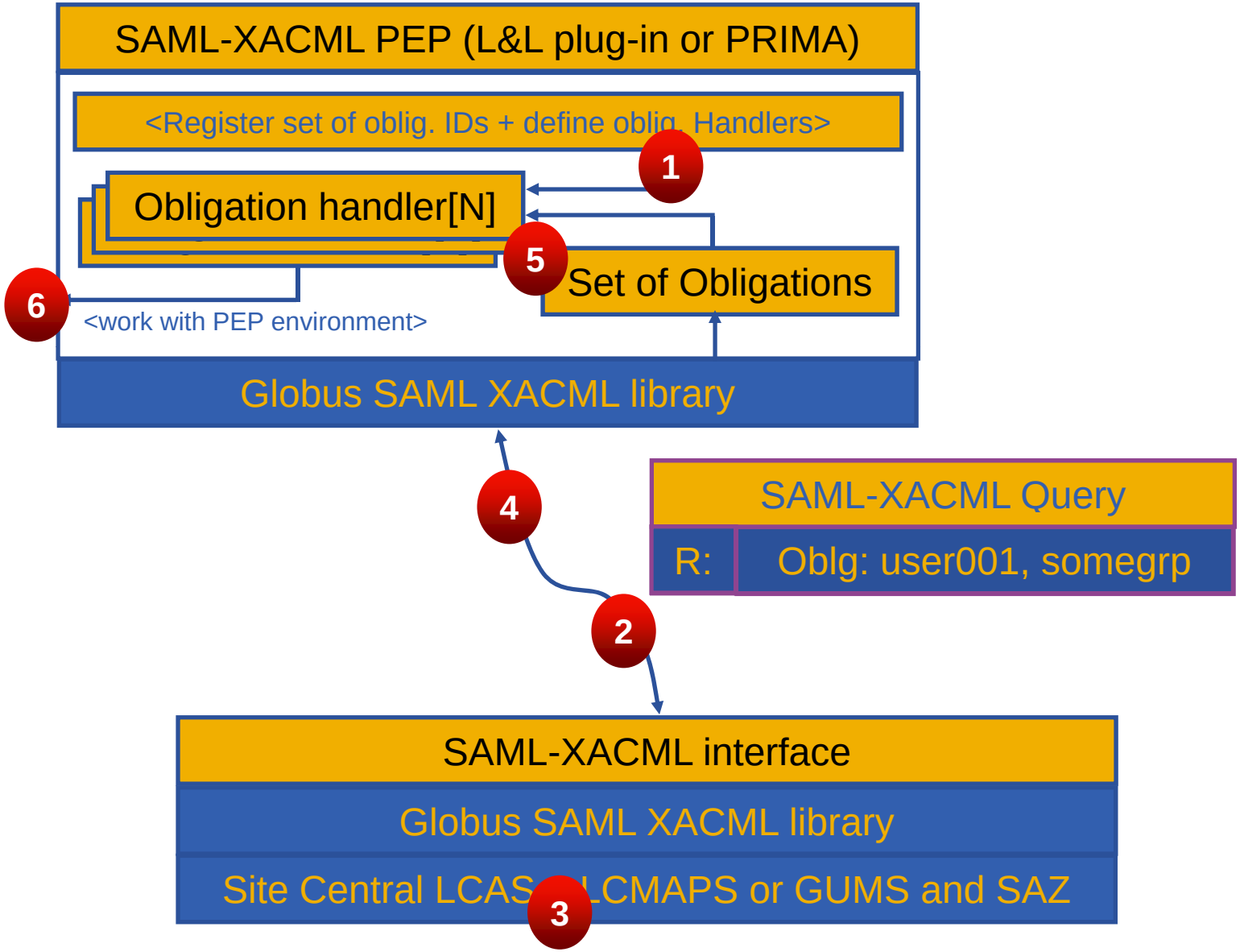
Issues with this setup:

- 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 node will probably **overload** an NFS server

The (old) big picture



How it should work (conceptual)



The paper work

The AuthZ workgroup

- **The group members:**
  - OSG: Igor Sfiligoi, Gabriele Garzoglio, Ted Hesselroth, Jay Packard, John Hover, Mine Altunay, Valery Sergeev, John Weigand, Keith Chadwick, Tanya Levshina
  - EGEE: Oscar Koeroo, Yuri Demchenko, Håkon Sagehaug
  - EGEE / INFN: Alberto Forti, Andrea Ferraro, Vincenzo Ciaschini, Valerio Venturi
  - Globus: Rachana Ananthakrishnan, Frank Siebenlist, Joe Bester

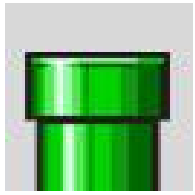
- **Will include the Condor team in the near future**
  - Discussions underway to support their requirements
  - Condor contacts:
    - Ian Alderman, Zackery Miller

Main discussion points

- **PEP & PDP interaction**
  - Different types of PEPs will need to interact with the PDP (SCAS)
    - Gatekeeper: PRIMA or LCMAPS backend
    - GridFTPd: PRIMA or LCMAPS backend
    - Glexec-on-{CE|WN}
    - dCache
    - …
  - The information that is contained in the request and response
  - Regardless of where the application that implements the PEP is created
  - How to get the required authorization information from the PEP
- **Upgradeability**
  - Changes in the attribute (datatype, value form., name(space))
  - Changes in the obligations, regarded as a set of attributes

The docs

- **The documents in work**
  - "XACML-SAML profile" (done)
    - Profiles the use of XACML and SAML



  - "An XACML Attribute and Obligation Profile for AuthZ Interoperability in Grids" (reaching v1.0)
    - Profiles the use of the attributes and obligations in the XACML request & response protocol

Request: Subject

- Subject-id: X.509 DN (OpenSSL oneline notation)
- Subject-issuer: X.509 Issuer DN (OpenSSL oneline notation)
- Subject-Certificate-Serial-Number
- Subject-vo
- VOMS-signing-subject: X.509 DN (OpenSSL oneline notation)
- VOMS-signing-issuer: X.509 DN (OpenSSL oneline notation)
- VOMS-dns-port
- VOMS-FQAN
- VOMS-Primary-FQAN
- Subject End-entity X509v3 Certificate Policies OIDs
- CA serial number
- Certificate chain (experimental)

Request: Action

- Run Job Queued: “queue”
  - Particularly via a CE to a Batch system
- Run Job Now: “execute-now”
  - On a CE; that’s the fork invokation
  - On a WN; direct execution
- Access file: “access”
  - No granularity in (specific) file permission (like read/write)

Request: Resource

- CE: Computing Element resource type
- WN: Worker Node resource type
- SE: Storage Element resource type
- Host DNS name

Request: Environment

- Supported obligations
  1. Handling of returned obligations is mandatory at the PEP
  2. The supported obligations are send to the PDP as advisory information to avoid returning useless obligations
     - *see previous statement*
- Pilot job invoker identity
  - This means ***all*** Subject attributes of the pilot job identity
  - Policy statement example:
    - *"The VO of the pilot job invoker and real user job MUST be the same"*

Response: Obligations (0)

- **UIDGID**
  - UID (integer): Unix User ID local to the PEP
  - GID (integer): Unix Group ID local to the PEP
  - Must be consistent with: Username (if receiving both)
- **Username**
  - Username (string): Unix username or account name local to the PEP.
  - Must be consistent with: Username (if receiving both)
- **SecondaryGIDs**
  - Multi recurrence
    - GID (integer): Unix Group ID local to the PEP
- **AFSToken**
  - AFSToken (string) in base64: AFS Token passed as a string

Response: Obligations (1)

- **RootAndHomePaths**
  - RootPath (string): this parameter defines a sub-tree of the whole file system available at the PEP. The PEP should mount this sub-tree as the “root” mount point (‘/’) of the execution environment. This is an absolute path.
  - HomePath (string): this parameter defines the path to home areas of the user accessing the PEP. This is a path relative to RootPath.
  - Needs obligation(s): UIDGID or Username
- **StorageAccessPriority**
  - Priority (integer): an integer number that defines the priority to access storage resources.
  - Needs obligation(s): UIDGID or Username

Open to discussion

- **Explicit declaration of an multi-user pilot job scenario?**
- **Were do we send the RSL string?**
  - Action?
  - Environment?
- **Requirements from the Condor team?**
  - Condor's canonical name: <useraccount>@some.site
  - Problem with the subject-id being used for the X.509 subject DN

The implementation

It works!

```
okoeroo@kvasir:~ — bash — 118x10
oscar-koeroos-computer:~/dev/globus/xacml-alpha-04/dist/xacml-1.0 okoeroo$ ./xacml-client -e http://`hostname`:8080/
Got obligation urn:gt-egee-osg:pool:uidgid
 urn:oasis:names:tc:xacml:1.0:subject:subject-id [http://www.w3.org/2001/XMLSchema#string] = pool001
 urn:oasis:names:tc:xacml:1.0:subject:subject-id [http://www.w3.org/2001/XMLSchema#string] = grppool
Got obligation urn:gt-egee-osg:pool:sgids
 urn:oasis:names:tc:xacml:1.0:subject:subject-id [http://www.w3.org/2001/XMLSchema#string] = sgidppool0
 urn:oasis:names:tc:xacml:1.0:subject:subject-id [http://www.w3.org/2001/XMLSchema#string] = sgidppool1
 urn:oasis:names:tc:xacml:1.0:subject:subject-id [http://www.w3.org/2001/XMLSchema#string] = sgidppool2
Server said: urn:oasis:names:tc:SAML:2.0:status:Success:0
oscar-koeroos-computer:~/dev/globus/xacml-alpha-04/dist/xacml-1.0 okoeroo$
```

- **Localhost (low latency, but having the laptop hardware as a bottleneck)**
  - Optimum rate (with SSL) was:
    - Nominal: 7Hz
    - Burst: 20Hz
    - Interval between bursts: 12 seconds

New components in CVS & Etics

- **org.glite.security.saml2-xacml2-c-lib-R_0_0_2_1**
  - This is version alpha-0.0.7 from Globus
  - Contains:
    - the gSOAP stuff
    - SAML2-XACML2 schema
    - Helper functions
    - Optional overriding of network layer
    - Pushes registered obligations in the Environment of the Request
- **org.glite.security.lcmaps-plugins-scas-client-HEAD**
  - Depends on saml2-xacml2-c-lib
  - Implements the client code for the protocol
    - Uses the network layer overriding to implement SSL/TLS
  - Implements the handlers for the supported obligations

Todo

- **Tying the loose ends together**
  - The LCMAPS plugin is kinda ready
    - Integrated test: gLExec will be used to stress test the framework
  - The prototype SCAS service should be ready any day
    - Expecting first CVS checking of it next week, if all works as promised
    - Expecting pretty nice performance
      - *Simple tests showed to exceed the CERN requirement*
  - Name spaces for the attributes and identifiers in all sections
    - Having a discussion now about this topic to include OGF in the process
    - We'll use 'something' in the meanwhile

?

The implementation