



The GSI plug-in for gSOAP: building cross-grid interoperable secure grid services

Massimo Cafaro^{1,2}, **Daniele Lezzi**^{1,2}, Sandro Fiore^{1,2}, Giovanni Aloisio^{1,2}
and Robert van Engelen³

¹ University of Salento, Lecce & SPACI Consortium, Italy

² Euro-Mediterranean Centre for Climate Change, Lecce, Italy

³ Computer Science Department Florida State University, USA



Outline

- Motivations
- The gSOAP Toolkit
- The GSI plug-in for gSOAP
- The GSI plugin-in and GRelC project
- Conclusions and future work



Motivations



Motivations

- Work started in 2001 in the context of the GridLab project
- Need for GSI enabled Grid Services (OGSA)
- GSI plugin + gSOAP = open source solution to the problem of securing Web Services in grid environments providing full interoperability between grid environments based on the Globus Toolkit and gLITE middleware.



The gSOAP Toolkit



The gSOAP Toolkit

- A toolkit for Web Services development in C/C++
- <http://www.cs.fsu.edu/~engelen/soap>
- Automatic generation of stubs & skeletons
- Automatic generation of WSDL documents
- WSDL importer for client development
- SSL support for secure Web Services & clients
- Cookies support
- DIME & MIME attachment support
- Plug-in extensibility
- Globus GSI support with our plug-in



The GSI plug-in



The GSI plug-in v3.2

- based on the GSS API for improved performances;
- extensive error reporting related to GSS functions;
- powerful debugging framework;
- support for both IPv4 & IPv6 networking;
- support for development of both client and server;
- support for mutual authentication
- support for authorization;
- support for delegation of credentials;
- **support for delegation of delegated credential;**
- support for connection caching;
- **support for VOMS extensions**



The GSI plug-in v3.2

- Our plug-in exploits the modular architecture of the gSOAP toolkit that enables a simple extension mechanism of gSOAP capabilities
- Developers must register plug-ins with gSOAP, so that full access to run-time environment & function callbacks is granted
- The registration associates the plug-in's local data with gSOAP run-time and is done using the gSOAP *soap_register_plugin* function, supplying as one of the arguments the plug-in initialization function



Plug-in local data

- Every plug-in has local data associated with the run-time environment upon registry with gSOAP
- Local plug-in data can be accessed through the lookup function *soap_lookup_plugin*



Plug-in local data

```
struct gsi_plugin_data
{
/* this is needed to save a gSOAP callback,
 * to be used in gsi_connect()
 */
int (*fopen)(struct soap *soap, const char *endpoint, const char *host, int
port);

/* this is needed to save a gSOAP callback,
 * to be used in gsi_accept()
 */
int (*faccept)(struct soap *soap, int s, struct sockaddr *a, int *n);

/* Authorization callback prototype */
int (*gsi_authorization_callback) (struct soap *soap);
/* distinguished name of the client */
char *client_identity;

/* distinguished name of the server */
char *server_identity;
```



Plug-in local data

```
/* filename to be used to save delegated credential;
 * needed only for concurrent servers using fork()
 */
char *proxy_filename;

/* GSS context */
gss_ctx_id_t    context;

/* GSS credential */
gss_cred_id_t   credential;

/* the distinguished name of the GSS credential */
char *identity;

/* delegated credential */
gss_cred_id_t   delegated_credential */

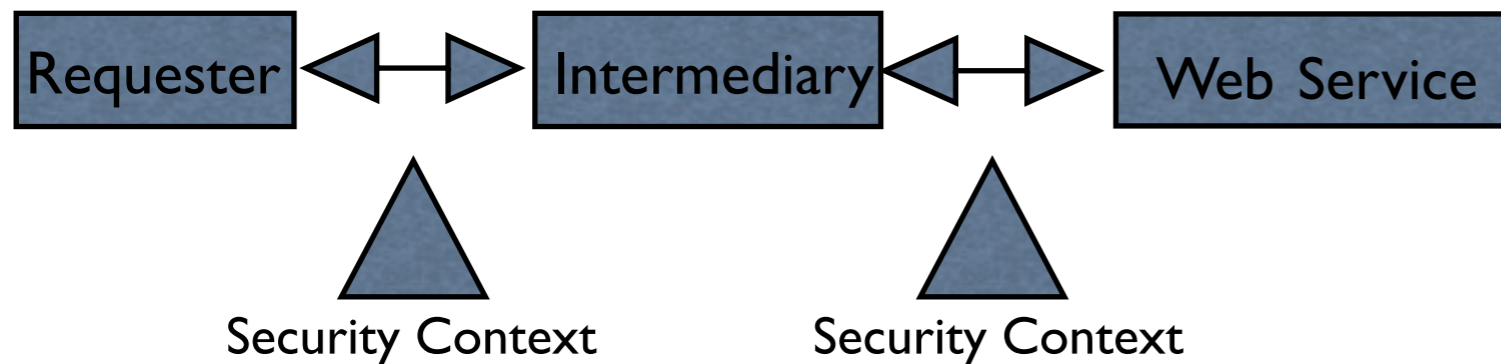
/* GSI connection request flags */
OM_uint32       req_flags;

/* DN of the remote web service
 * this is needed when requesting delegation of credentials
 */
char *target_name;

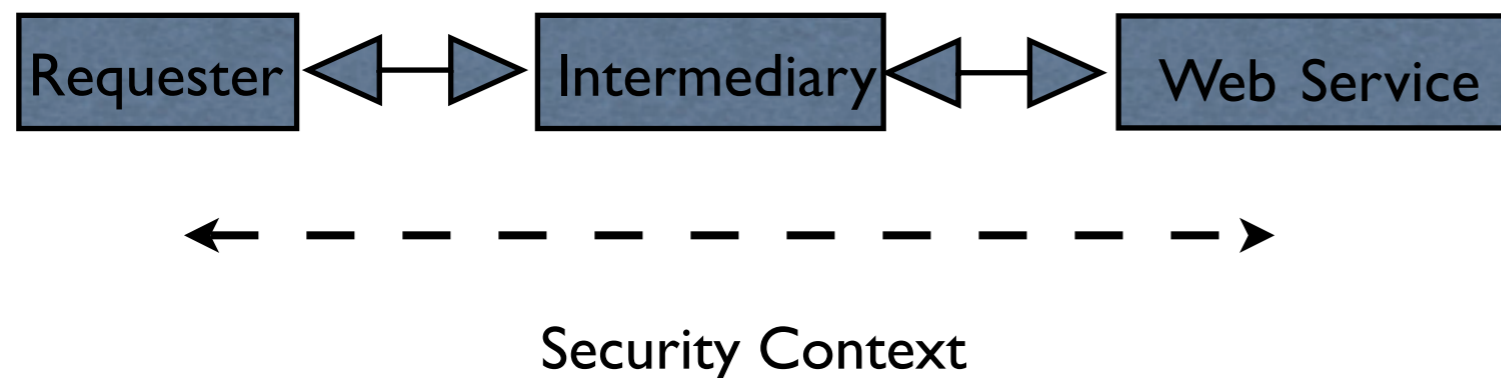
char **fqans; /* VOMS attributes */
};
```



TLS vs Message Level Security



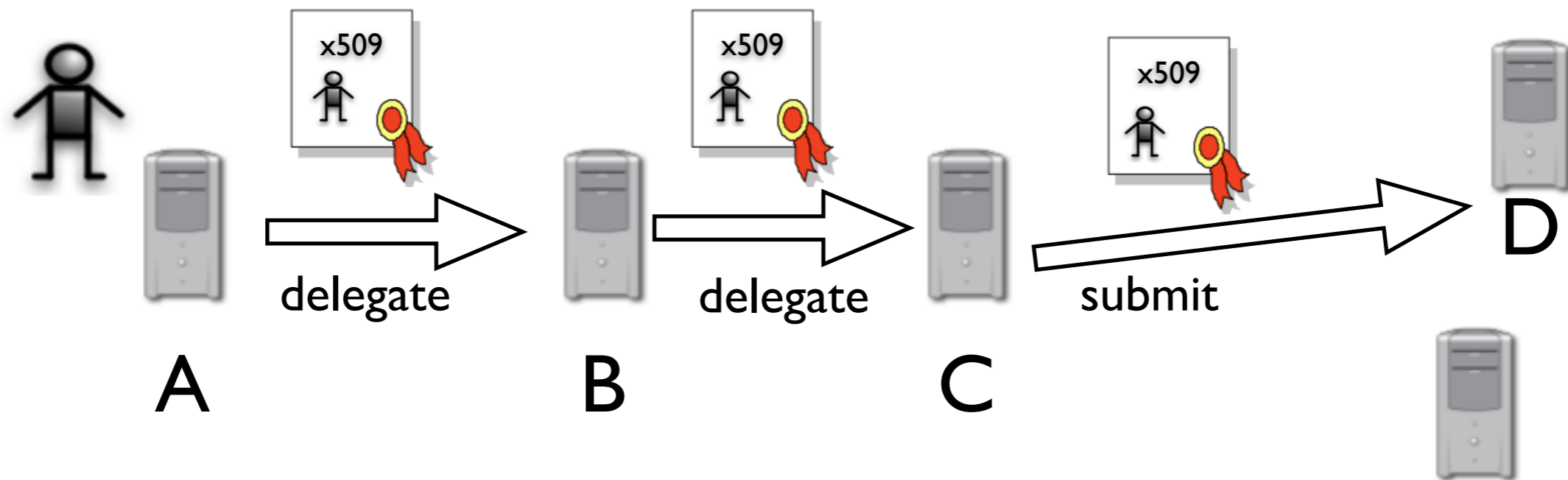
- Point-to-point (host-to-host) security
- SSL, IPSEC
- Authentication, data integrity, confidentiality



- End-to end security
- Multihop topology with intermediaries
- Firewall/DMZ traversal



Delegation of delegated credentials



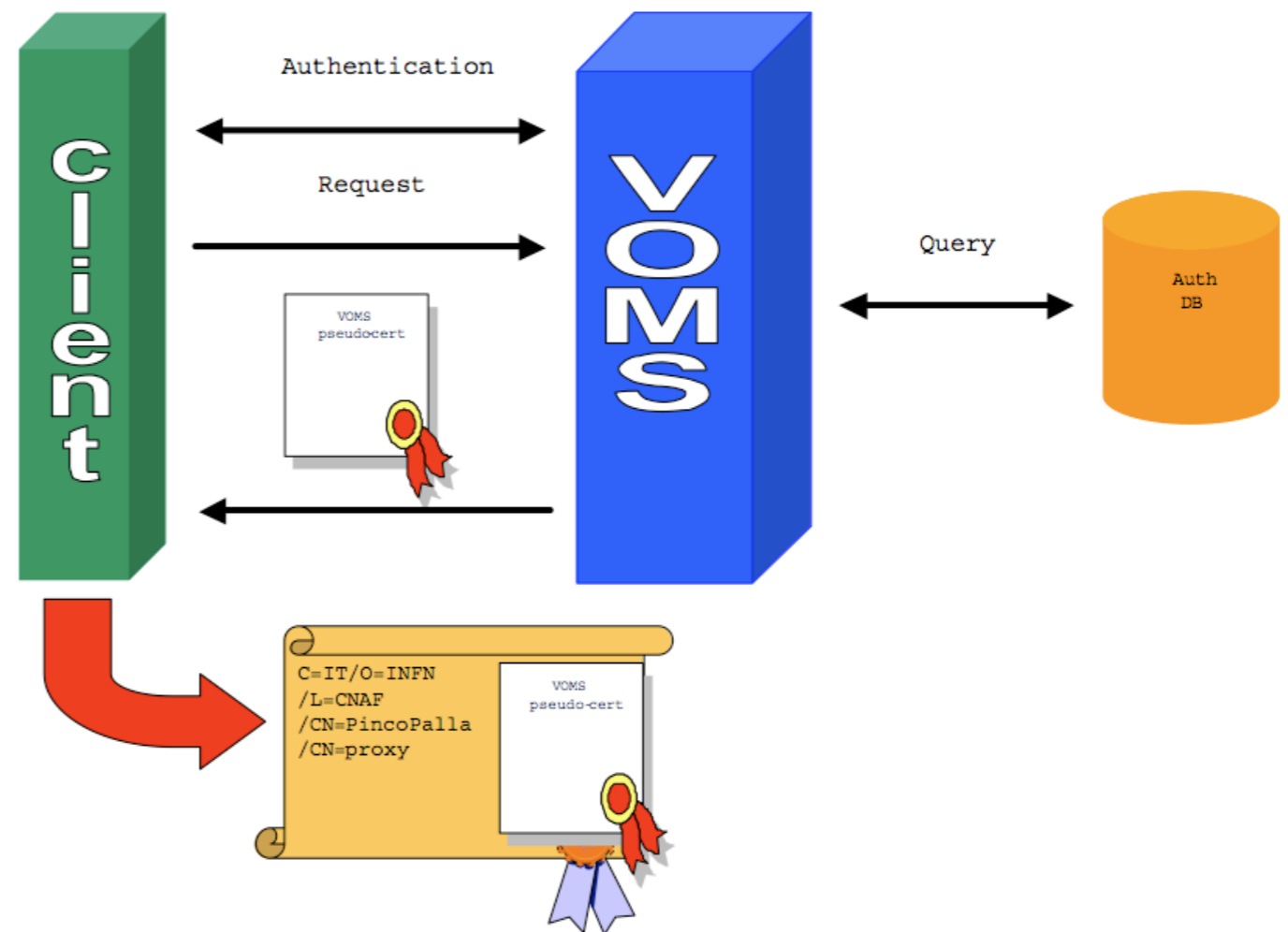
Client A delegates its credentials to an intermediate service B which, in turn, delegates A credentials to a service C (i.e. a grid broker) that finally uses them to contact a service D on behalf of A.



VOMS support

Provides information on the user's relationship with her Virtual Organization: her groups, roles and capabilities.

- single login using voms-proxy-init only at the beginning of the session
- expiration time: the authorization information is only valid for a limited period of time as the proxy certificate itself
- backward compatibility: the extra VO related information is in the user's proxy certificate, which can be still used with non VOMS-aware services
- multiple VOs: the user may "log-in" into multiple VOs and create an aggregate proxy certificate, which enables her to access resources in any of them



FQAN

/VO[/group[/subgroup(s)]][/Role=role][/~~Capability=cap~~]

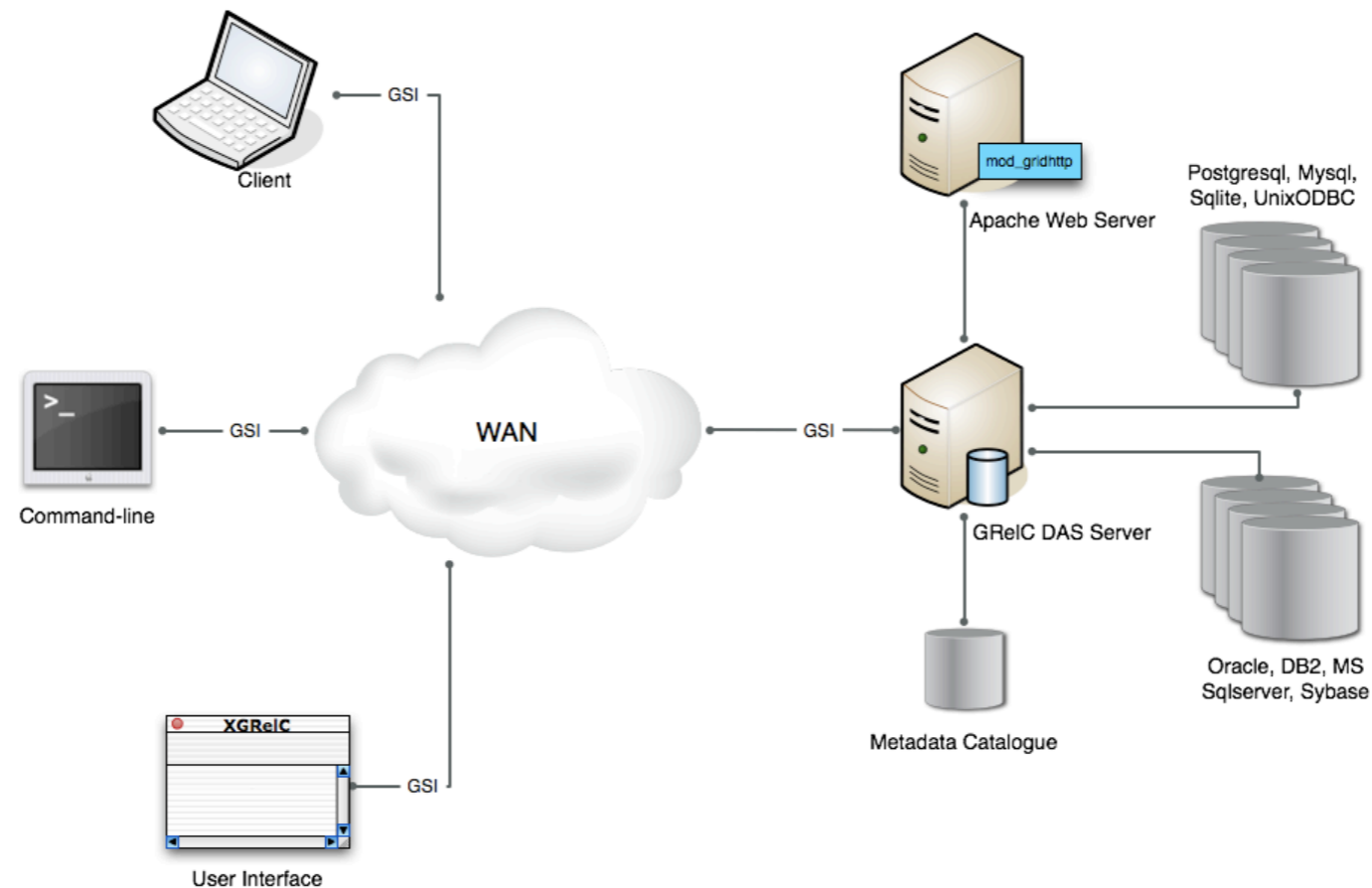


VOMS support

```
subject : /C=IT/O=INFN/OU=Personal Certificate/L=HPCC University of Lecce/  
CN=Alessandro Negro/CN=proxy  
issuer : /C=IT/O=INFN/OU=Personal Certificate/L=HPCC University of Lecce/  
CN=Alessandro Negro  
identity : /C=IT/O=INFN/OU=Personal Certificate/L=HPCC University of Lecce/  
CN=Alessandro Negro  
type : proxy  
strength : 512 bits  
path : /tmp/x509up_u503  
timeleft : 11:59:19  
=== V0 libi extension information ===  
V0 : libi  
subject : /C=IT/O=INFN/OU=Personal Certificate/L=HPCC University of Lecce/  
CN=Alessandro Negro  
issuer : /C=IT/O=INFN/OU=Host/L=CNAF/CN=voms.cnaf.infn.it  
attribute : /gilda/grelc/das/grelc02.unile.it/sakila/Role=grelc-db-select/  
Capability=NULL  
attribute : /gilda/Role=NULL/Capability=NULL  
attribute : /gilda/grelc/Role=NULL/Capability=NULL  
timeleft : 11:59:35
```



VOMS based authorization: GRelC DAS example



- Developed at University of Salento
- Grid-database management and Access Service
- Transparent, uniform and secure (GSI) access to etherogeneous DBMS
- Integrated in the EGEE gLITE middleware



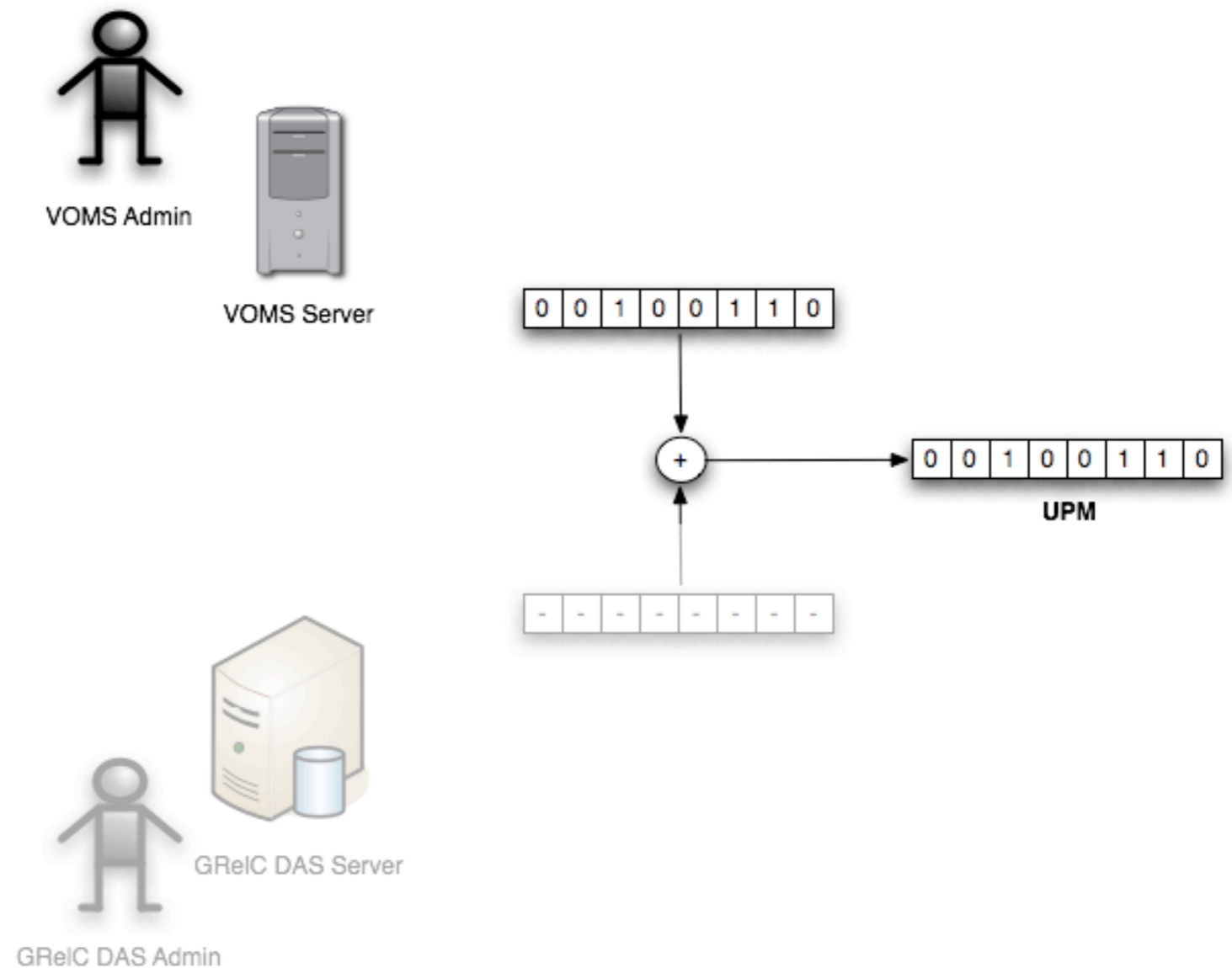
Two-level authorization (I)

- Global authorization (through VOMS extensions)
- Local authorization (by means of the local GRelC DAS authorization framework)
- The two masks obtained from global and local authorization are combined to infer the final User Privileges Mask (UPM)
- 3 scenarios
 - ▶ global mode, coarse grained approach
 - ▶ local mode, fine grained approach
 - ▶ combined mode



Global Mode

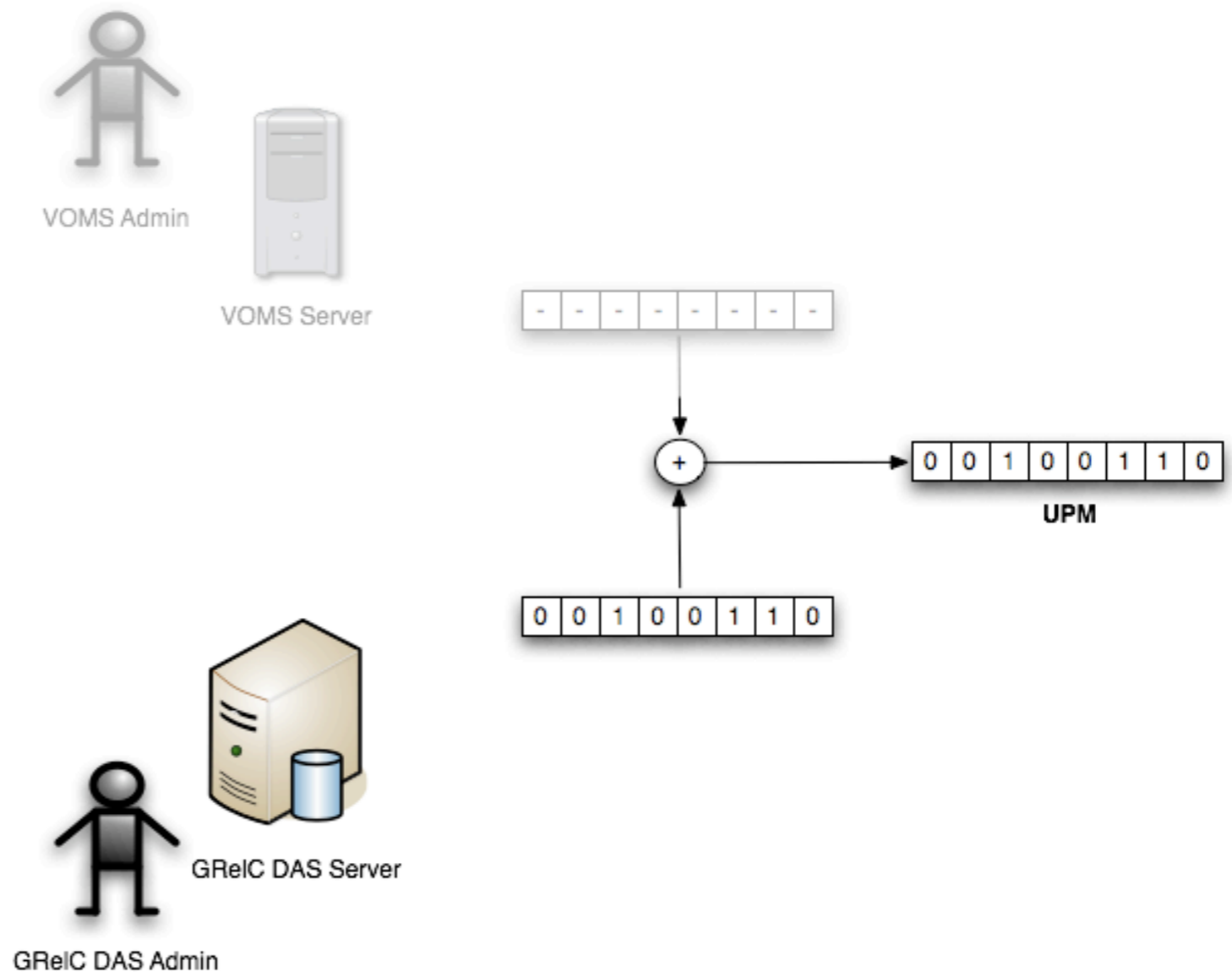
- User credentials must be obtained through *voms-proxy-init*
- The UPM is inferred from the available VOMS extensions
- No additional authorization setting is required on the GRelC DAS
- Easy and fast setup procedure
- It scales well
- Feasible for a real production grid environment





Local Mode

- User credentials must be obtained through *grid-proxy-init*
- The UPM is drawn out of the GRelC DAS metadata catalogue
- No VOMS extensions are added to the user proxy
- The setup procedure must be carried out on each GRelC DAS
- Scalability is worse



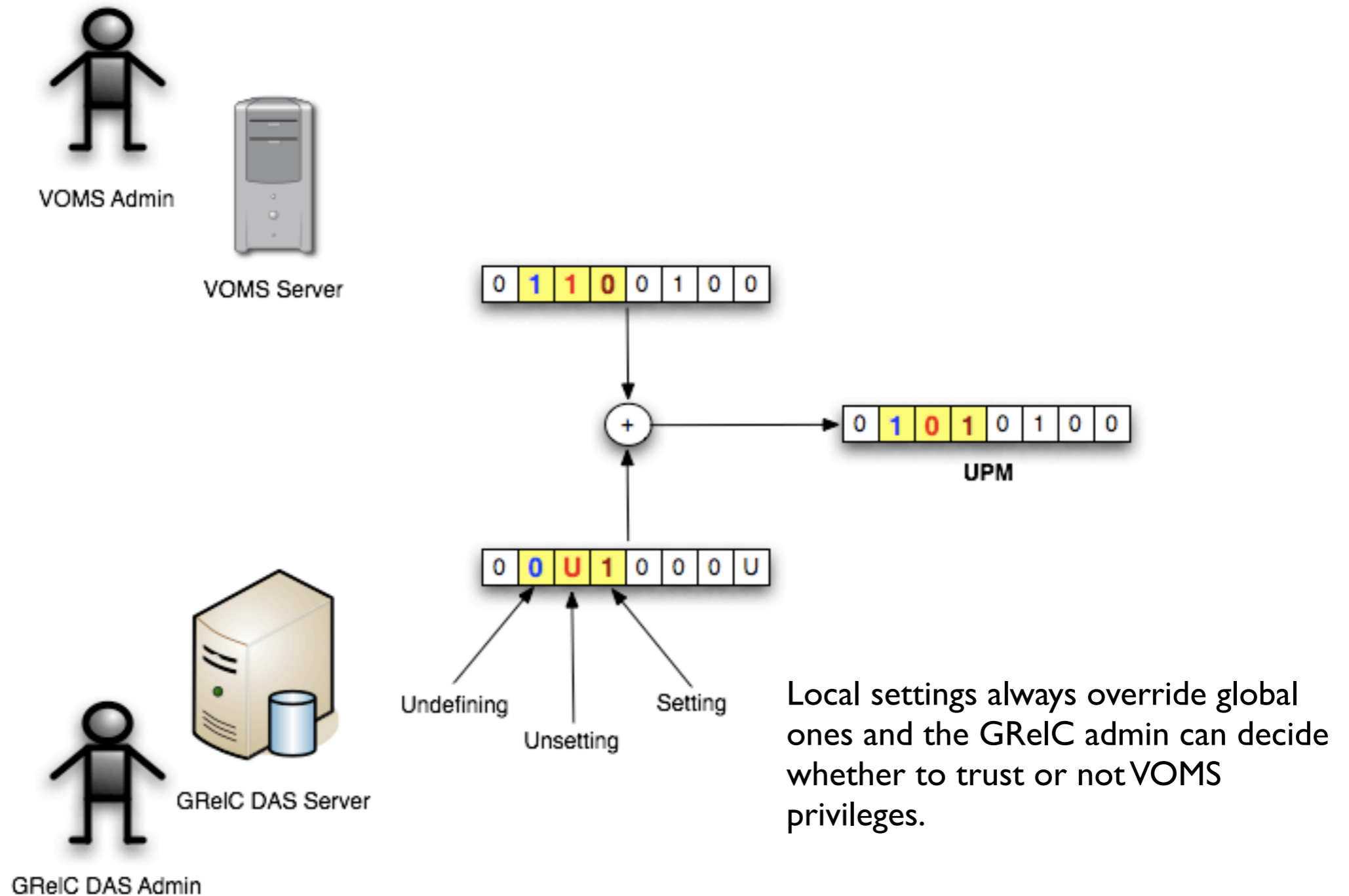


Combined Mode

- User credentials must be obtained through *voms-proxy-init*
- The UPM is inferred joining information on access policies coming from VOMS extensions and the GRelC DAS metadata catalogue
- VOMS level (grant or revoke)
- GRelC DAS level (setting, undefining, unsetting)

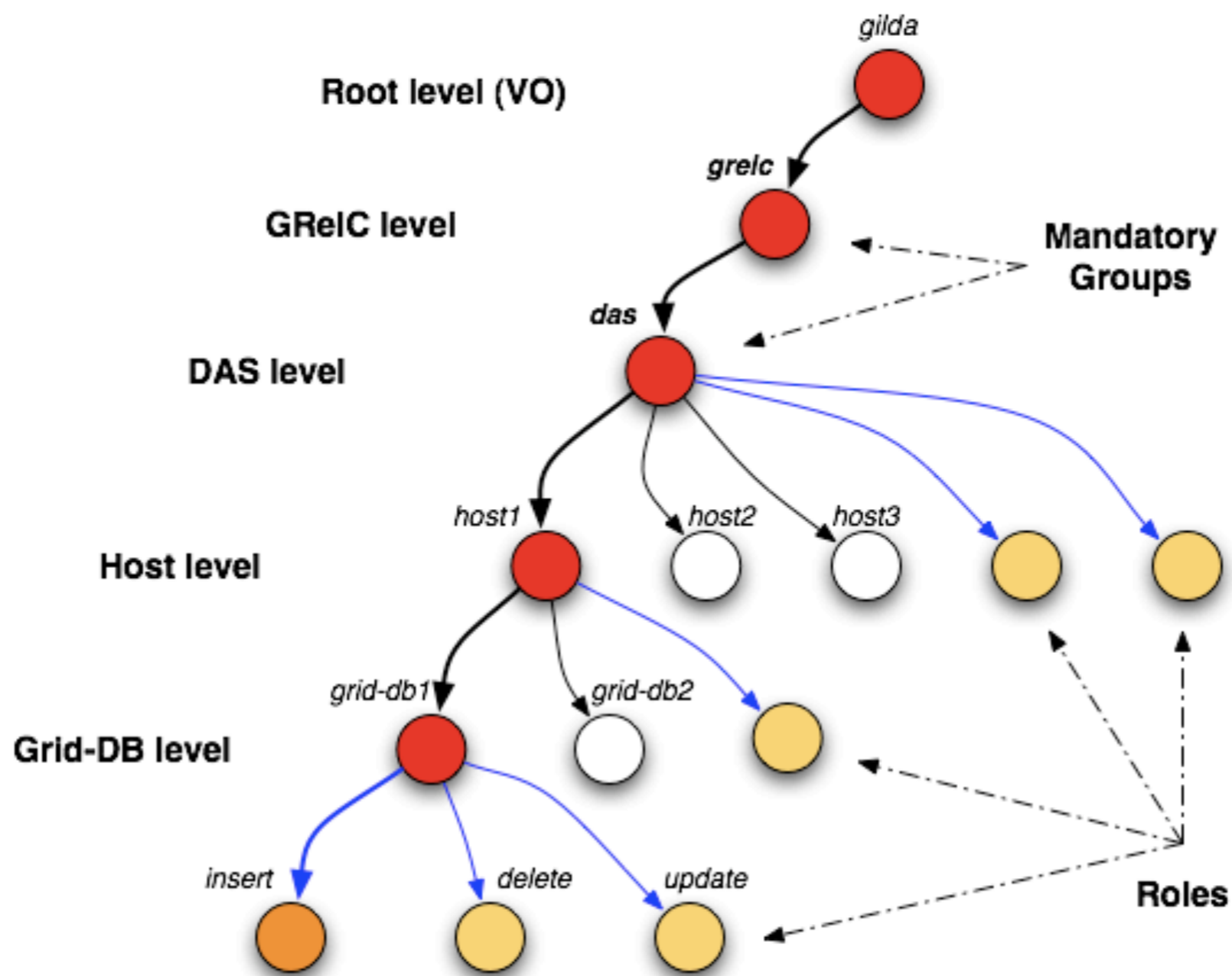


Combined Mode - An Example





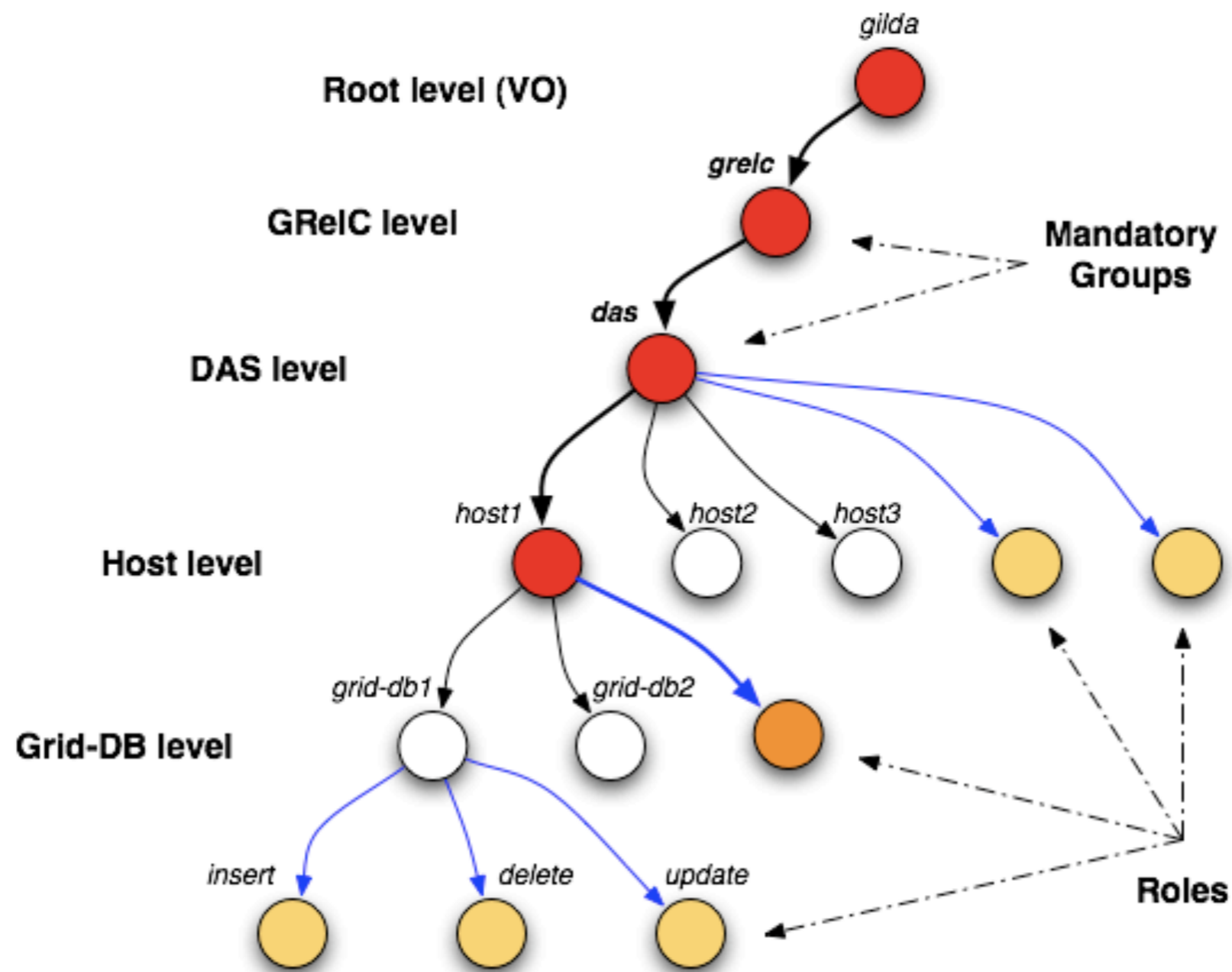
Roles and Groups on VOMS (II)



`/gilda/greic/das/host1/grid-db1/Role=greic-db-insert`



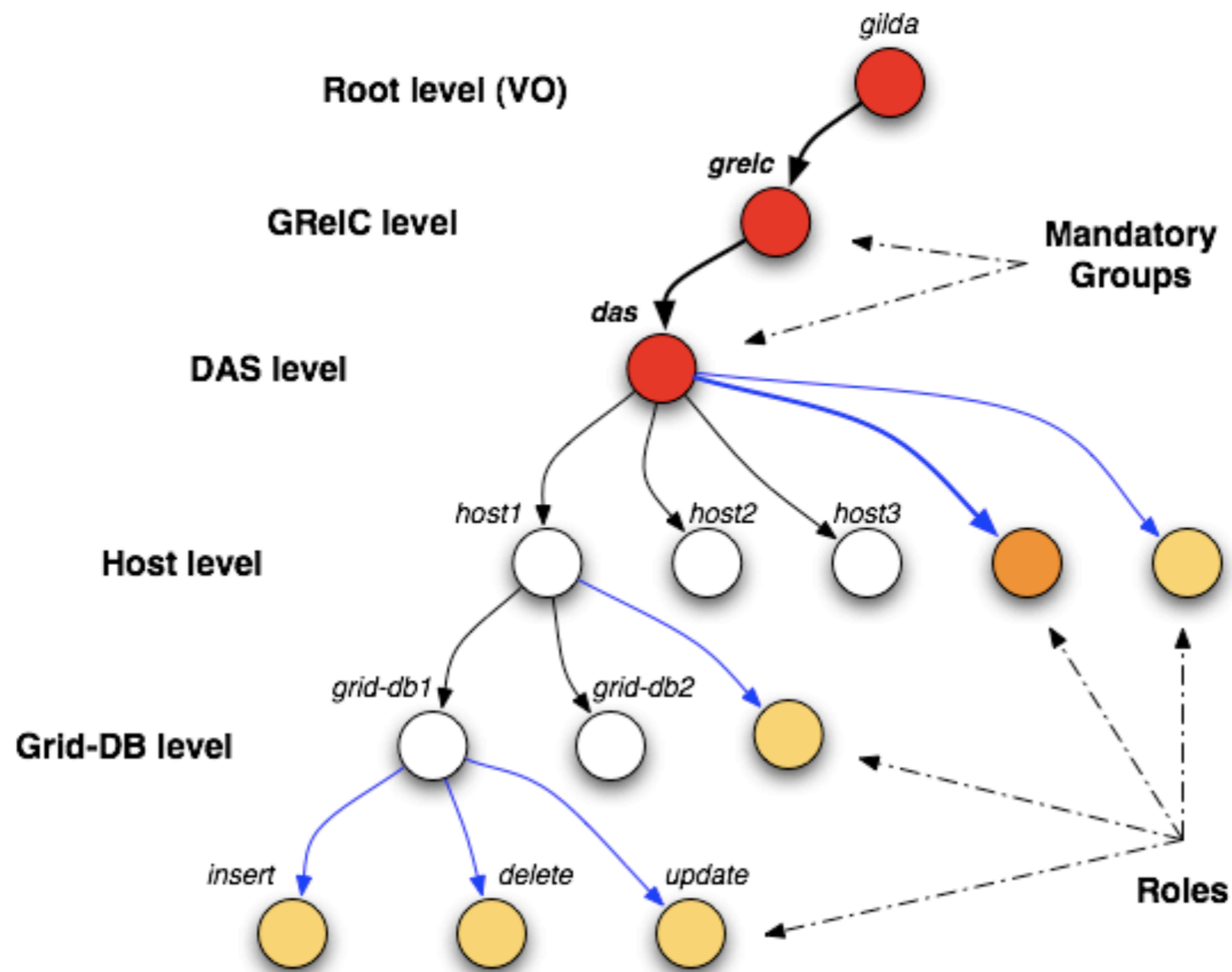
Roles and Groups on VOMS (III)



`/gilda/greic/das/host1/Role=greic-db-insert`



Roles and Groups on VOMS (IV)



`/gilda/greic/das/Role=greic-db-insert`



Roles and Groups on VOMS (V)

- VOMS usage of groups/roles does not add an extra level of dependence on the VO admin
- Once the VO admin has created a new group or role, she has the faculty to delegate the management of the groups/roles to other authorized users (e.g. the GRelC admins)
- GRelC admins can
 - ▶ add users already belonging to the VO to VOMS groups
 - ▶ grant roles to users
 - ▶ create new groups
- VOMS admin has to
 - ▶ create mandatory VO subgroups (`/<VO>/grelic/das`)
 - ▶ setup the GRelC roles
 - ▶ define VOMS authorized GRelC admins



GSI plugin on SEPAC Grid



- **Southern European Partnership for Advanced Computing**
- All services in the SEPAC grid are GSI enabled through the gSOAP-GSI plugin
- Deployment

<http://www.sepac-grid.org>



Related Work

- **Globus WS Core**
 - ▶ Support for GSI or WS-Security enabled grid services
 - ▶ No multi-threading
 - ▶ C WS Core container can only use the default SELF authorization scheme

- **CGSI plugin (Castor manager)**
 - ▶ Voms extensions
 - ▶ No client/serve mixed mode



GRB WebSite (<http://grb.spaci.it>)

- Downloads of GRB components
- News
- Events
- Publications
- Staff
- Deployment
- ...



Conclusions and Future Work

- Support for Shibboleth
 - ▶ Retrieve Shibboleth/VOMS mixed attributes.
- Support for WS-Security



For Any Information

- **Supervisor:** **prof. Giovanni Aloisio** (giovanni.aloisio@unile.it)
- **Project PI:** **Massimo Cafaro, Ph. D.** (massimo.cafaro@unile.it)
- **Team Members:**
 - ▶ **Daniele Lezzi, Ph. D.** (daniele.lezzi@unile.it)
 - ▶ **Sandro Fiore, Ph. D.** (sandro.fiore@unile.it)
 - ▶ **Robert van Engelen** (engelen@cs.fsu.edu)
- **Website:** <http://grb.spaci.it/grb>



Q&A

