WP2 security status

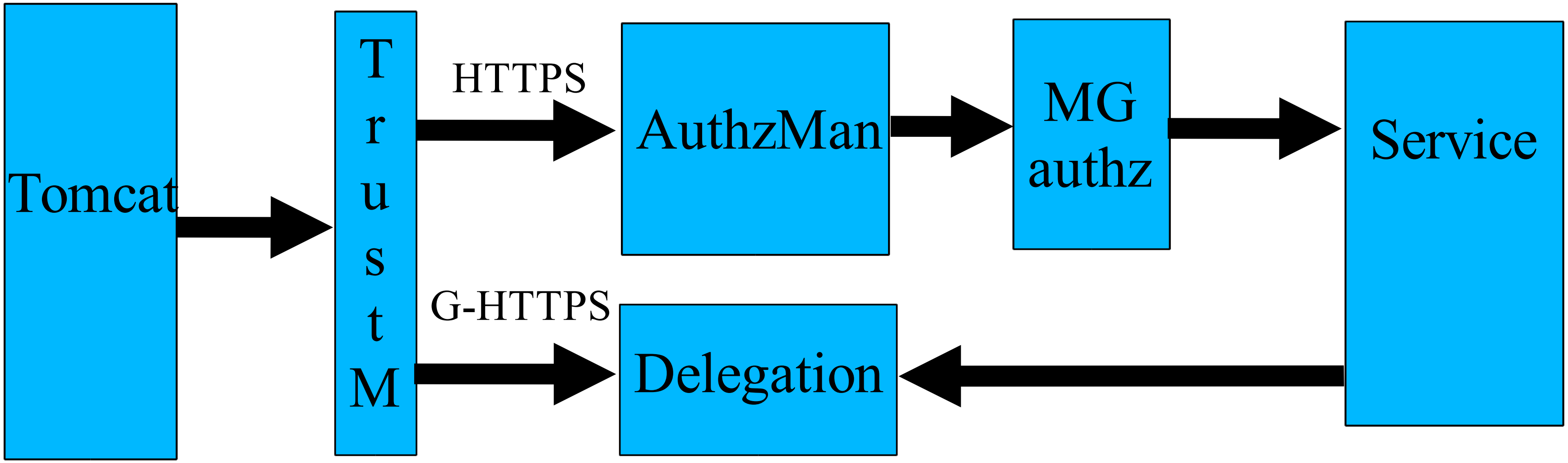
- RPMS for secure tomcat, authentication and authz out
- LCFG for secure tomcat out
- Prototype or delegation ready, but further development prostponed until authz integrated.
- Currently integrating authz to WP2 software
- Medium grained authorization (or enforcement)

Delegation

- Uses G-HTTPS-ish protocol
- Command line client and client api
- Only java, plans for c/c++
- Delegations currently only in memory in tomcat
- Delegations per service
- Service api to get the delegation from the delegation storage in the server
- Client can delegate and get info about delegations in server
- Delegation ID allows several delegations in single service

Medium grained authz

- Could also be called enforcement
- Stopgap for WP2 until time for finegrained
- Simple method based access control
- Authzmanager provides service specific roles (LRC)
- Medium grained authz filters using roles and service specific method access lists
- Ex: LRCAdmin=delete, replicate, list
- Ex: LRCUser=list, replicate



Tomcat

TrustM

HTTPS

G-HTTPS

AuthzMan

MG authz

Service

Delegation

gSOAP

- **Edg-gsoap-base, basic gSOAP package with:**
  - User proxy certifiate
  - Server proxy certificate
  - Authorization policy and role support (when not using VOMS)
  - Additional useful packaging to ease development
- **Needed in the WSDD creation, header files during compilation and library while running**

```
#include "WP2Security.h"
#include <iostream.h>
#include "SpitfireInfoService.h"
#include "stdsoap2.h"
/*
  Example to contact the information service and get a list of tables
  as well as their 'create table' commands
*/

int main(int argc, char * const argv[])
{
  char *credentials = NULL;
  char *cacertdir = NULL;
  char *passphrase = NULL;
  char *rseedfile = NULL;
  char *accessurl = "https://localhost:8443/Spitfire/services/SpitfireInfo";
  char *role = "default-readrole"; char *policy = "test";
  struct soap *s = soap_new();

  if (edg_data_soap_sec_ctxt_init_m(s,passphrase,credentials,cacertdir,rseedfile)){
      cout << "wp2 soap security init failed " << endl;
      exit(1);
  }
  edg_data_soap_sec_add_role_and_policy_m(s,role,policy,role,policy);
  //edg_data_soap_sec_print_sec_parameters(s, stdout);
  SpitfireInfoService query(accessurl,s);
```