Globus Toolkit Security

Plamen Alexandrov, ISI Masters' Student

Softwarepark Hagenberg, January 24, 2009

Table of Contents

- Introduction (3-5)
- Grid Security Infrastructure (6-15)
  - Transport & Message-level security
  - Authentication
  - Authorization
  - Single sign-on and Delegation
- Configuration with Security Descriptors (16-21)
- Demo: Simple Client/Service Security (22-22)
- Security services (23-26)
  - Community Authorization Service
  - Passport Online CA & MyProxy
  - Delegation Service

WHAT IS A SECURE COMMUNICATION?

- Privacy:
  - only the sender and the receiver should be able to understand the conversation.
- Integrity:
  - the receiving end must be able to know *for sure* that the message he is receiving is exactly the one that the transmitting end sent him.
- Authentication (and Non-repudiation):
  - ensure that the parties involved in the communication are who they claim to be.
  - protected from malicious users who try to *impersonate* one of the parties in the secure conversation.
- Authorization vs. Authentication (Access control):
  - Accounting/auditing for policy compliance.
- Managing user credentials
- Administering access rights

GRID SECURITY REQUIREMENTS

- Authentication for User/Processes/Resources
- Appliance of local access control mechanisms
- Constraints:
  - *Single sign-on & Delegation*
  - *Protection of credentials*
  - *Interoperability with local security solutions*
  - *Exportability (standard X.509v3)*
  - *Support for secure group communication*
  - *Support for multiple implementations*

JAVA WS AUTH & AUTHZ

- Web Services use SOAP over HTTP for communicating messages.
- Implements the WS-Security standard and the WS-SecureConversation specification.
- Provided features are:
  - authentication of the sender.
  - encryption of the message.
  - integrity protection of the message.
  - replay attack protection.

Grid Security Infrastructure

- Provides:
  - Transport-level and message-level security.
  - Authentication through X.509 digital certificates.
  - Several authorization schemes.
  - Credential delegation and single sign-on.
  - Different levels of security:
    - container, service, resource and client.

Grid Security Infrastructure: Transport & Message-level security

- Transport-level security
  - Encrypts the **complete** communication.
- Message-level security
  - Encrypts only the **content** of the SOAP message.
- Both are based on public-key cryptography.
- Can guarantee privacy, integrity, authentication.
- Security schemes (not mutually exclusive):
  - *GSI Secure Message*: message-level, WS-Security.
  - *GSI Secure Conversation*: message-level, WS-SecureConversation, secure context, credential delegation.
  - *GSI Transport*: transport-level, TLS(SSL).

Grid Security Infrastructure: Transport & Message-level security

	GSI Secure Message	GSI Secure Conversation	GSI Transport
Technology	WS-Security	WS-SecureConversation	TLS (SSL)
Privacy (Encrypted)	YES	YES	YES
Integrity (Signed)	YES	YES	YES
Anonymous authentication	NO	YES	YES
Delegation	NO	YES	NO
Performance	Good if sending few messages	Good if sending many messages	Best

GRID SECURITY INFRASTRUCTURE: AUTHENTICATION

- GSI supports three authentication methods:
  - X.509 Certificates:
    - provides strong authentication.
    - all three protection schemes support X.509 certificates.
  - Username and password:
    - more limited form of authentication.
    - privacy, integrity and delegation features are not supported.
  - Anonymous authentication:
    - unauthenticated communication.
    - when using more than one security scheme.
    - ex.: GSI Secure Conversation (with X.509 certificates) and anonymous GSI Transport – escape redundant authentication.

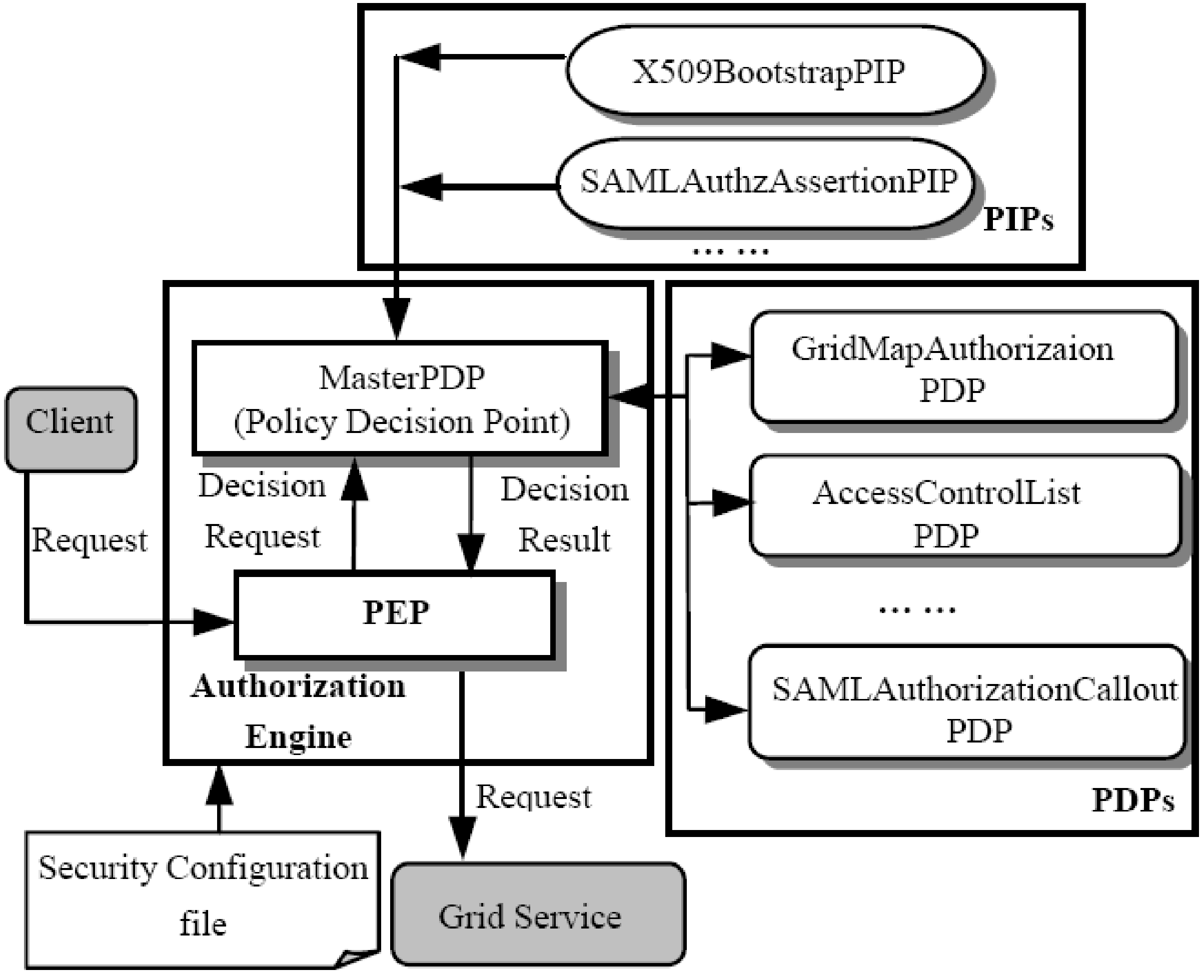
Grid Security Infrastructure: Basic authentication (How to)

- Obtaining host certificates
  - Request a certificate from well-known CA.
  - Use SimpleCA:
    - $GLOBUS_LOCATION/setup/globus/setup-simple-ca.
    - $GLOBUS_LOCATION/setup/globus_simple_ca_*CA_Hash*_setup/setup-gsi –defaul.
    - grid-cert-request -host 'hostname‘.
    - grid-ca-sign -in hostcert_request.pem -out hostsigned.pem.
  - Use Globus’s low-trust cert.: http://gcs.globus.org:8080/gcs.
- Install host credentials in container
  - Should be located in /etc/grid-security/.
  - Container key should be only readable by ‘globus’ user.
- Verify basic security (creating local proxy certificate)
  - grid-proxy-init -verify –debug.

Grid Security Infrastructure: Authorization

- Based on XACML authorization model
  - Policy Enforcement Point (PEP)
    - intercepts the access requests from users and sends the requests to the PDP.
  - Policy Decision Point (PDP)
    - makes **access decisions** according to the security policy or policy set written by PAP.
    - **queries** the PIP for attributes of the subjects, the resource, and the environment.
    - the access decision is sent to the PEP.
  - Policy Information Point (PIP)
  - Policy Administration Point (PAP)
- A policy language
  - Policies organized hierarchically into PolicySets, Policies and Rules.
  - A rule is composed of a target, an effect and a condition.
  - A Policy consists of a target, one or more rules, and an optional set of obligations.

Grid Security Infrastructure: GT4 Authorization Framework

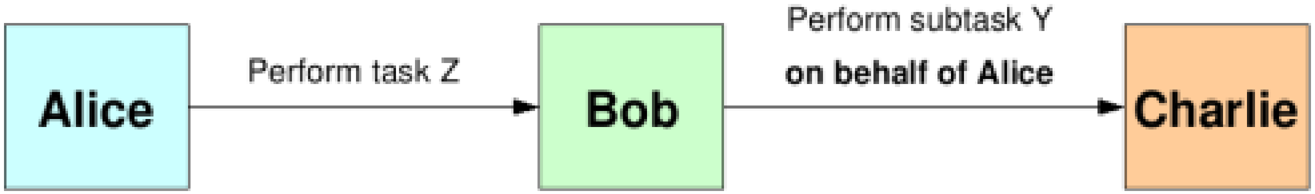


GRID SECURITY INFRASTRUCTURE: BUILT-IN AUTHORIZATION OPTIONS

- Server-side authorization PDPs at Container, Service or Resource level
  - *None*: no authorization is performed.
  - *Self*: client's identity should match server's identity.
  - *Gridmap*: a list of 'authorized' users (ACL).
  - *Identity authz.*: one user gridmap conf. programmatically.
  - *Host authz.*: client should provide a host credential.
  - *SAML callout authz.*: OGSA Authorization Service.
- Client-side authorization options
  - *None*: no authorization is performed.
  - *Self*: server's identity should mach client's identity.
  - *Identity authz.*: service should have specified identity.
  - *Host*: service should have a host credential and client should resolve address of the host.
- Custom authorization

Grid Security Infrastructure: Single sign-on and Delegation

- A user should be able to initiate computations by authenticating only once.
- A computation may acquire resources, use resources, release resources, and communicate internally without further authentication of the user.
- Users should be able to delegate (restricted) rights to computational units (credential delegation).



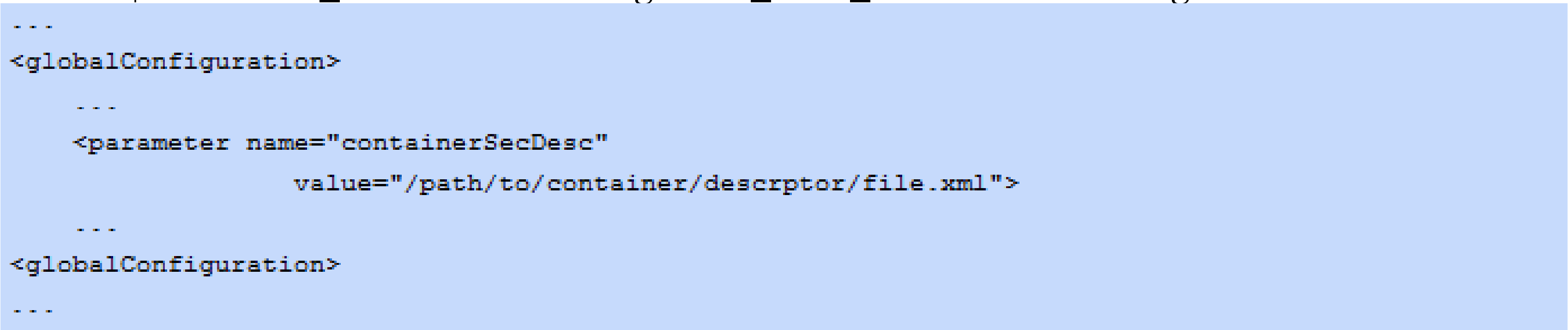
- Public key based authentication – do not expose your private key.

Grid Security Infrastructure: X.509 Proxy Certificates

- A “proxy certificate” is a special type of X.509 certificate that is signed by the normal end entity cert (or by another proxy).
- Gives the owner of the proxy the right of *temporarily* acting on the original entity’s behalf.
- The private key of a proxy may not be secured by a password (exposure is limited).
- Contains embedded restriction policies
  - Policy is evaluated by resource (upon proxy use).
  - Reduces rights available to the proxy to a subset of those held by the original entity.
- May be used in local environment or created remotely and signed by the original entity to support delegation.

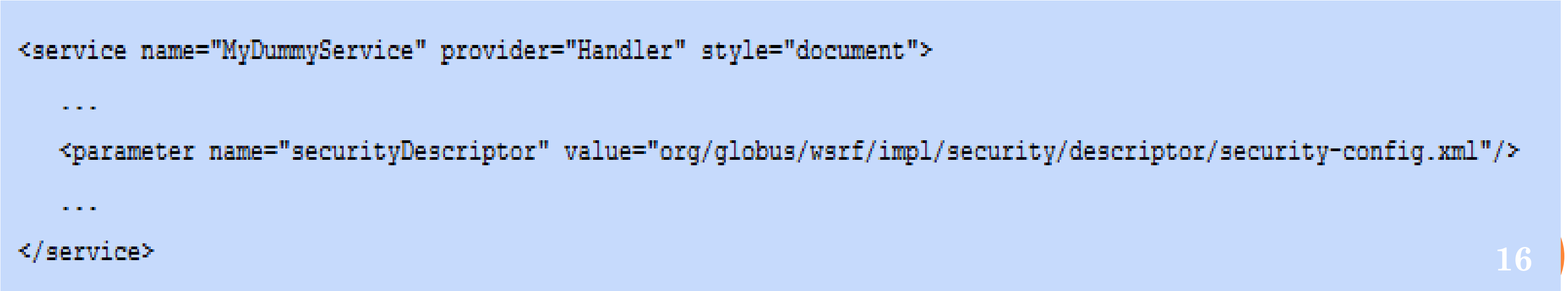
Configuration with Security Descriptors: Setup

- Configuring Container Security Descriptor
  - $GLOBUS_LOCATION/etc/globus_wsrf_core/server-config.wsdd.



```
...
<globalConfiguration>
    ...
    <parameter name="containerSecDesc"
                  value="/path/to/container/descrptor/file.xml">
    ...
<globalConfiguration>
...
```

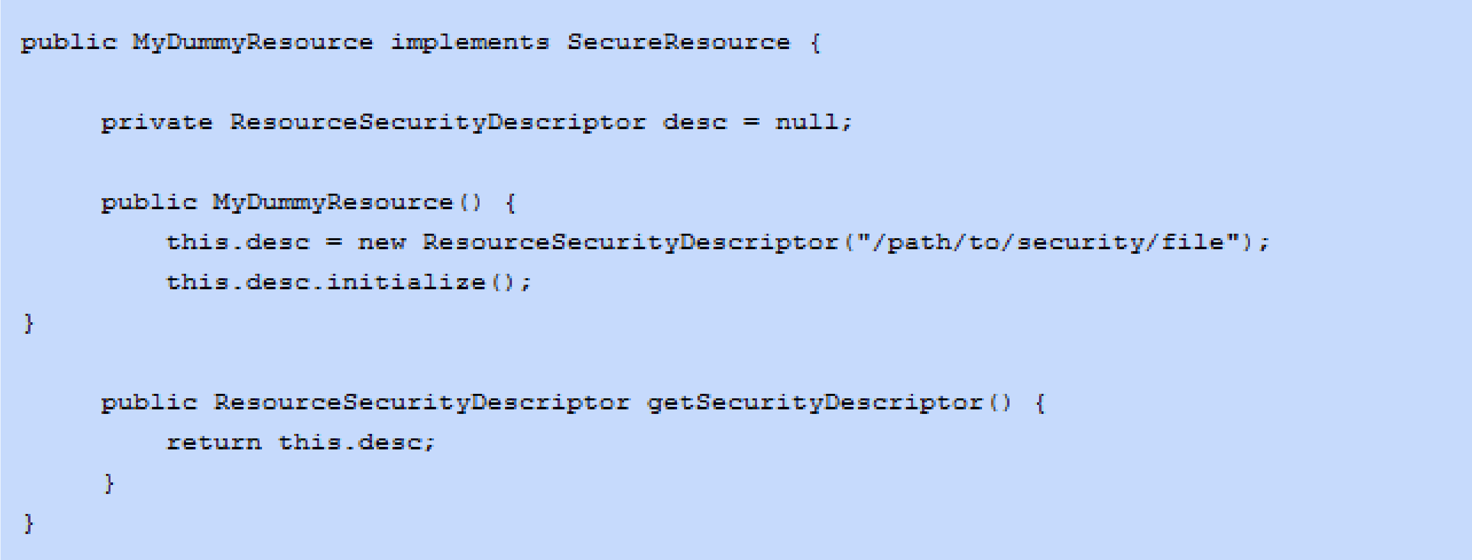
- Configuring Service Security Descriptor
  - in the service's deployment descriptor section as a parameter.



```
<service name="MyDummyService" provider="Handler" style="document">
   ...
   <parameter name="securityDescriptor" value="org/globus/wsrf/impl/security/descriptor/security-config.xml"/>
   ...
</service>
```

Configuration with Security Descriptors: Setup

- Configuring Resource Security Descriptor
  - the object should be returned by getSecurityDescriptor method .



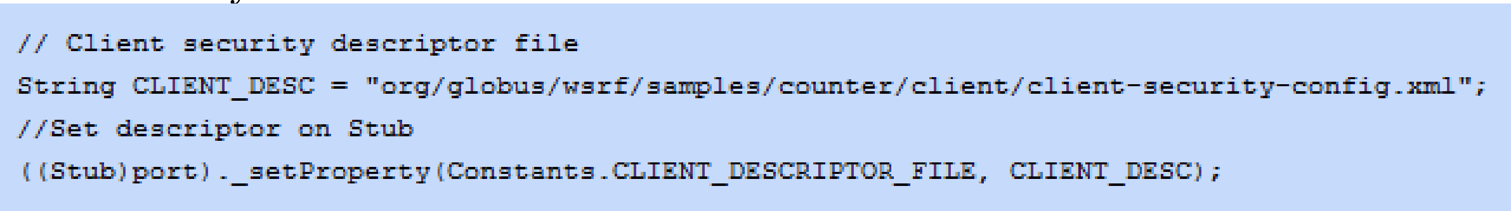
```
public MyDummyResource implements SecureResource {

     private ResourceSecurityDescriptor desc = null;

     public MyDummyResource() {
         this.desc = new ResourceSecurityDescriptor("/path/to/security/file");
         this.desc.initialize();
}

     public ResourceSecurityDescriptor getSecurityDescriptor() {
         return this.desc;
     }
}
```

- Configuring Client Security Descriptor
  - directly on the stub.



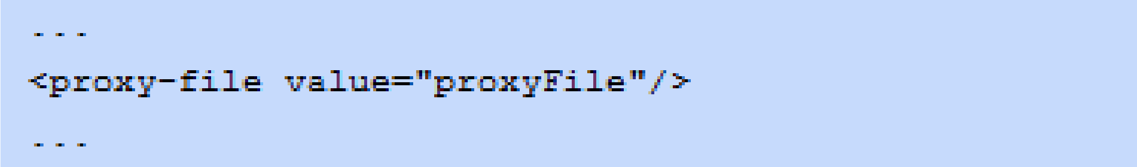
```
// Client security descriptor file
String CLIENT_DESC = "org/globus/wsrf/samples/counter/client/client-security-config.xml";
//Set descriptor on Stub
((Stub)port)._setProperty(Constants.CLIENT_DESCRIPTOR_FILE, CLIENT_DESC);
```

Configuration with Security Descriptors: Communication

- Valid only for Service and Client Security Descriptors.
- <auth-method>
  - <none/>: (**server only**) No authentication is performed.
  - <GSISecureMessage>
    - <protection-level>
      - <integrity/>: the message must be integrity-protected (signed).
      - <privacy/>: the message must be privacy-protected (encrypted and signed).
      - <peer-credential value=" *path to file with credentials to encrypt with*">: (**client only**).
  - <GSISecureConversation>
    - <protection-level>
      - <integrity/>: the message must be integrity-protected (signed).
      - <privacy/>: the message must be privacy-protected (encrypted and signed).
      - <anonymous/>: (**client only**) Server is accessed as anonymous.
      - <delegation value="full/limited"/>: (**client only**) Type of delegation to be done.
      - <context-lifetime/>: (**client only**) the lifetime of the context established.
  - <GSITransport>
    - <protection-level>
      - <integrity/>: the message must be integrity-protected (signed).
      - <privacy/>: the message must be privacy-protected (encrypted and signed).
      - <anonymous/>: (**client only**) Server is accessed as anonymous.

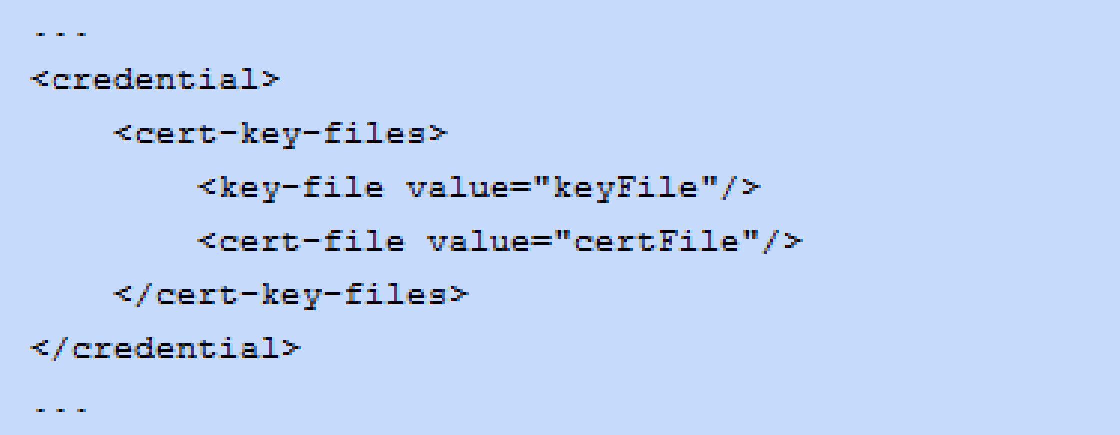
Configuration with Security Descriptors: Credentials

- Valid for Container, Service, Resource and Client Security Descriptors
- The credentials can be set using either:
  - the path to a proxy file.



```
...
<proxy-file value="proxyFile"/>
...
```

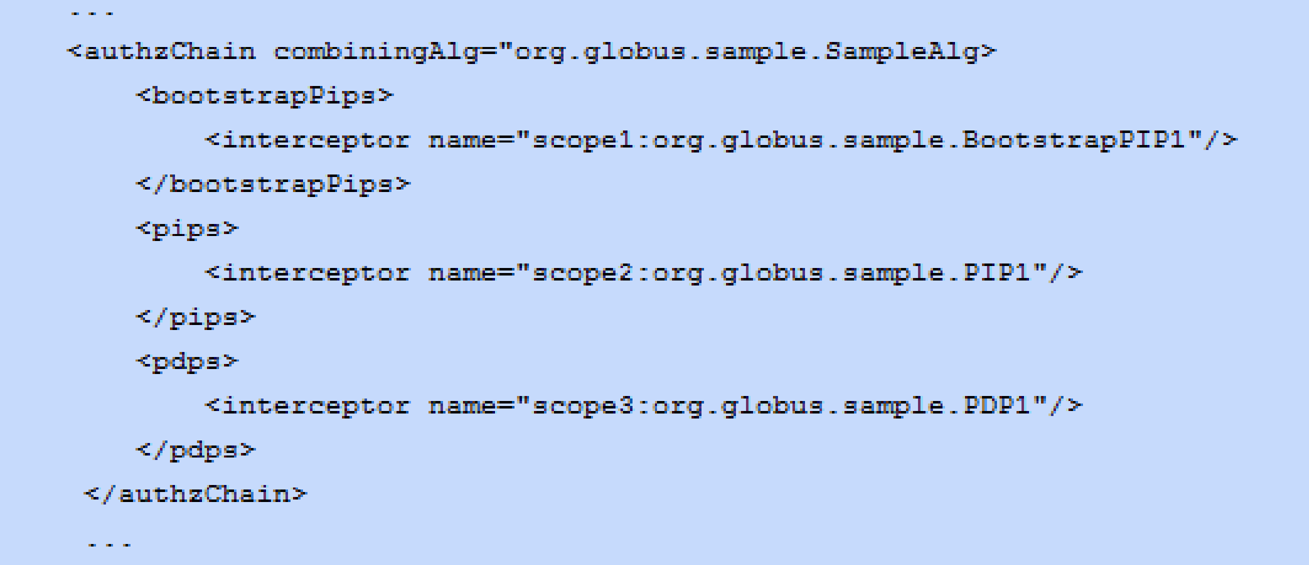
  - the path to a certificate and key file.



```
...
<credential>
    <cert-key-files>
        <key-file value="keyFile"/>
        <cert-file value="certFile"/>
    </cert-key-files>
</credential>
...
```

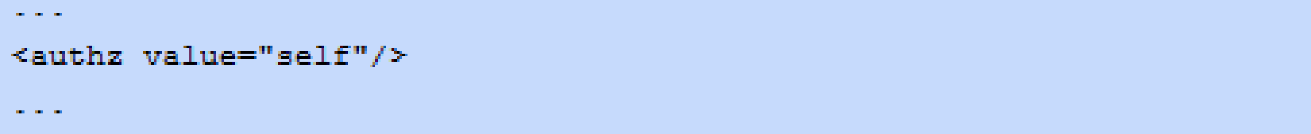
Configuration with Security Descriptors: Authorization

- Server-side configuration:



```
    ...
    <authzChain combiningAlg="org.globus.sample.SampleAlg>
        <bootstrapPips>
            <interceptor name="scope1:org.globus.sample.BootstrapPIP1"/>
        </bootstrapPips>
        <pips>
            <interceptor name="scope2:org.globus.sample.PIP1"/>
        </pips>
        <pdps>
            <interceptor name="scope3:org.globus.sample.PDP1"/>
        </pdps>
     </authzChain>
     ...
```

- Client-side configuration:
  - Identity authorization is done using the value as the identity.



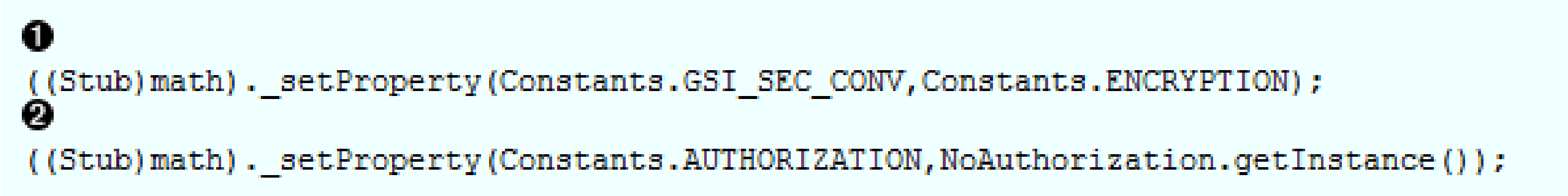
```
...
<authz value="self"/>
...
```

Configuration with Security Descriptors: Other

- Server-side configuration:
  - Reject Limited Proxy - if clients that present limited proxies can be allowed to authenticate successfully.
  - Replay attack prevention - messages outside of this time window will be rejected automatically, inside – checked by UUID.
  - Context lifetime – lifetime of security context (GSI Secure Conv.).
  - Default gridmap (resource level).
  - Run-as mode (service level): credentials your service should use for the operation being invoked:
    - <run-as value="caller"/>.
    - <run-as value="service"/>.
    - <run-as value="resource"/>.
    - <run-as value="system"/>.
  - Authentication and run-as per-method (service level).
  - Context Timer Interval (container level) (GSI Secure Conversation).
  - Replay Timer Interval (container level) (GSI Secure Message).
- Client-side configuration:
  - Username/Password - <usernameType> element.

DEMO: SIMPLE CLIENT/SERVICE SECURITY

- Changes in MathService's source code
  - Only logging security information.
  - Logging the invoked method, identity of the caller, invocation subject, service subject and system subject.
- Using simple service security configuration
  - Requiring private GSI Secure Conversation in service SD.
  - No authorization (with "none" PDP interceptor).
- Shorthand client-side source code configuration (no SD)



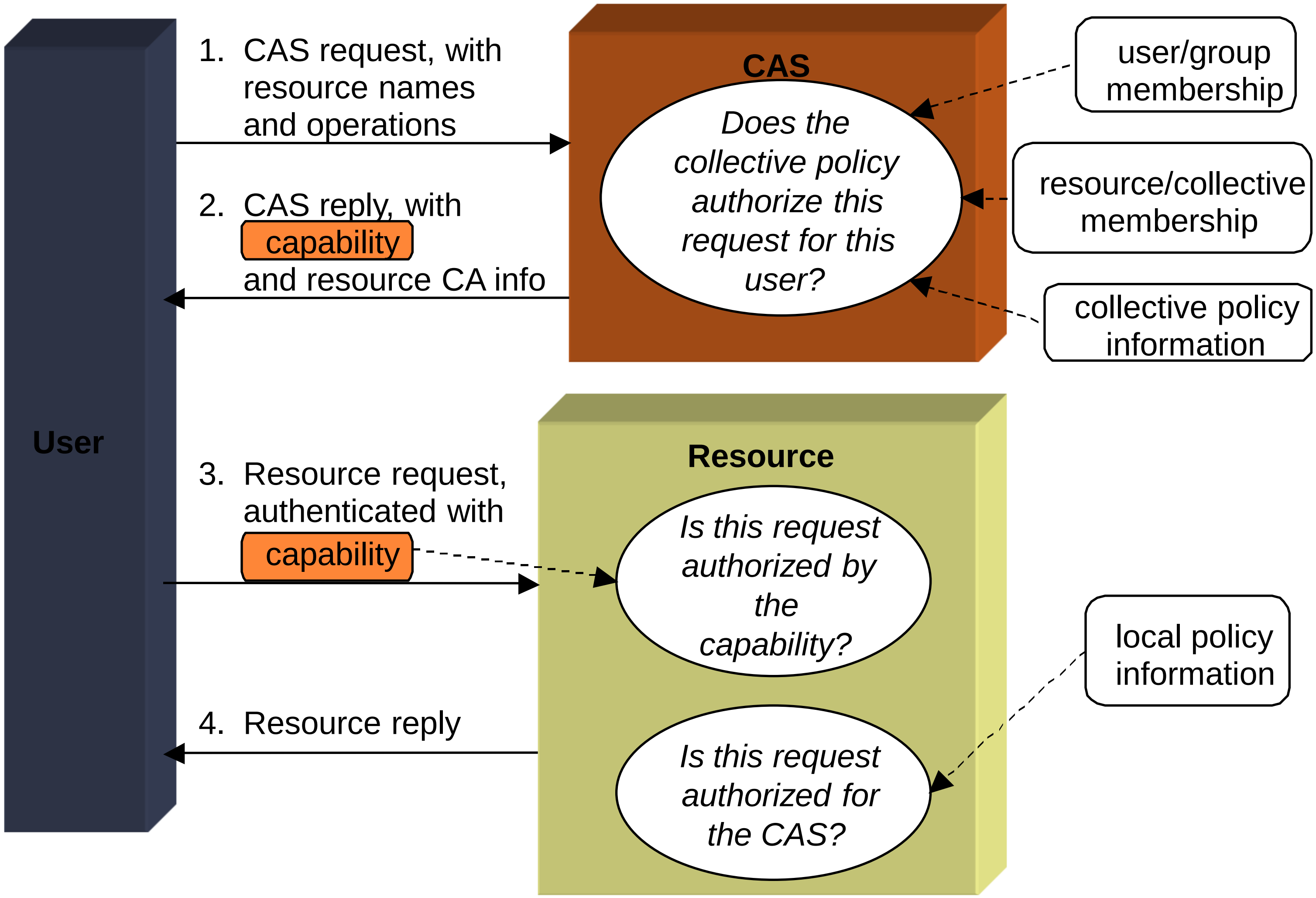
```
❶
((Stub)math)._setProperty(Constants.GSI_SEC_CONV,Constants.ENCRYPTION);
❷
((Stub)math)._setProperty(Constants.AUTHORIZATION,NoAuthorization.getInstance());
```

- 1. Using GSI Secure Conversation with Encryption.
- 2. No client side authorization.

SECURITY SERVICES: COMMUNITY AUTHORIZATION SERVICE

- Q: How does a large community grant its users access to a large set of resources?
  - Should minimize burden on both the users and resource providers.
- Community Authorization Service (CAS)
  - Community negotiates access to resources.
  - Resource outsources fine-grain authorization to CAS.
  - Resource only knows about "CAS user" credential.
    - CAS handles user registration, group membership…
  - User who wants access to resource asks CAS for a capability credential.
    - Restricted proxy of the "CAS user" cred., checked by resource.

Security services: Community Authorization Prototype


User
1. CAS request, with resource names and operations
2. CAS reply, with capability and resource CA info
3. Resource request, authenticated with capability
4. Resource reply
CAS
Does the collective policy authorize this request for this user?
user/group membership
resource/collective membership
collective policy information
Resource
Is this request authorized by the capability?
Is this request authorized for the CAS?
local policy information

Security services: Passport Online CA & MyProxy

- Requiring users to manage their own certs and keys is annoying and error prone.
- A solution: Leverage Passport global authentication to obtain a proxy credential.
  - Online credential repository.
  - Creates and issues new (restricted) proxy cert. to the user on demand (**myproxy-init, myproxy-logon)**.
  - Store X.509 proxy credentials in the MyProxy repository, protected by a passphrase, for later retrieval over the network.
  - Users can store and retrieve multiple X.509 end-entity credentials (**myproxy-store**, **myproxy-retrieve**.
  - Eliminates the need for copying private keys and certificate files.

Security services: Delegation service

- Provides an interface for the delegation and renewal of credentials to a hosting environment.
- Allows for a single delegated credential to be reused across multiple service invocations.
- Improved online credential repositories.
- globus-credential-delegate - delegation client.
- globus-credential-refresh - delegation refresh client.
- globus-delegation-client - C delegation client.
- wsrf-destroy - destroys a resource.
- wsrf-query - performs query on a resource property document.