

Grid Security

Haresh Patel
hareshe@caip

Introduction to Grid Security

- Kerberos
- Secure Shell
- Grid Security Infrastructure (GSI)
 - Based on Public Key Infrastructure (PKI)
 - Single sign-on
 - Cross domain authentication
 - Delegation mechanisms for creating temp credentials for users processes

Security Requirements

- Single sign-on
- Protection of credentials
- Interoperability with local security solutions
- Exportability
- Uniform Credentials/certification infrastructure
- Support for secure group communication
- Support for multiple implementations

Security Policy

- Grid environment consists of multiple trust domains
- Operations on single trust domain subject to local security policy only
- Partial mapping from global to local subjects
- Mutual authentication for entities in different trust domain
- Global subject mapped to local subject is assumed to be equivalent to being locally authenticated
- Access control decisions are made locally
- Process is allowed to act on behalf of a user
- Processes running on behalf on the same subject within the same trust domain may share one set of credentials

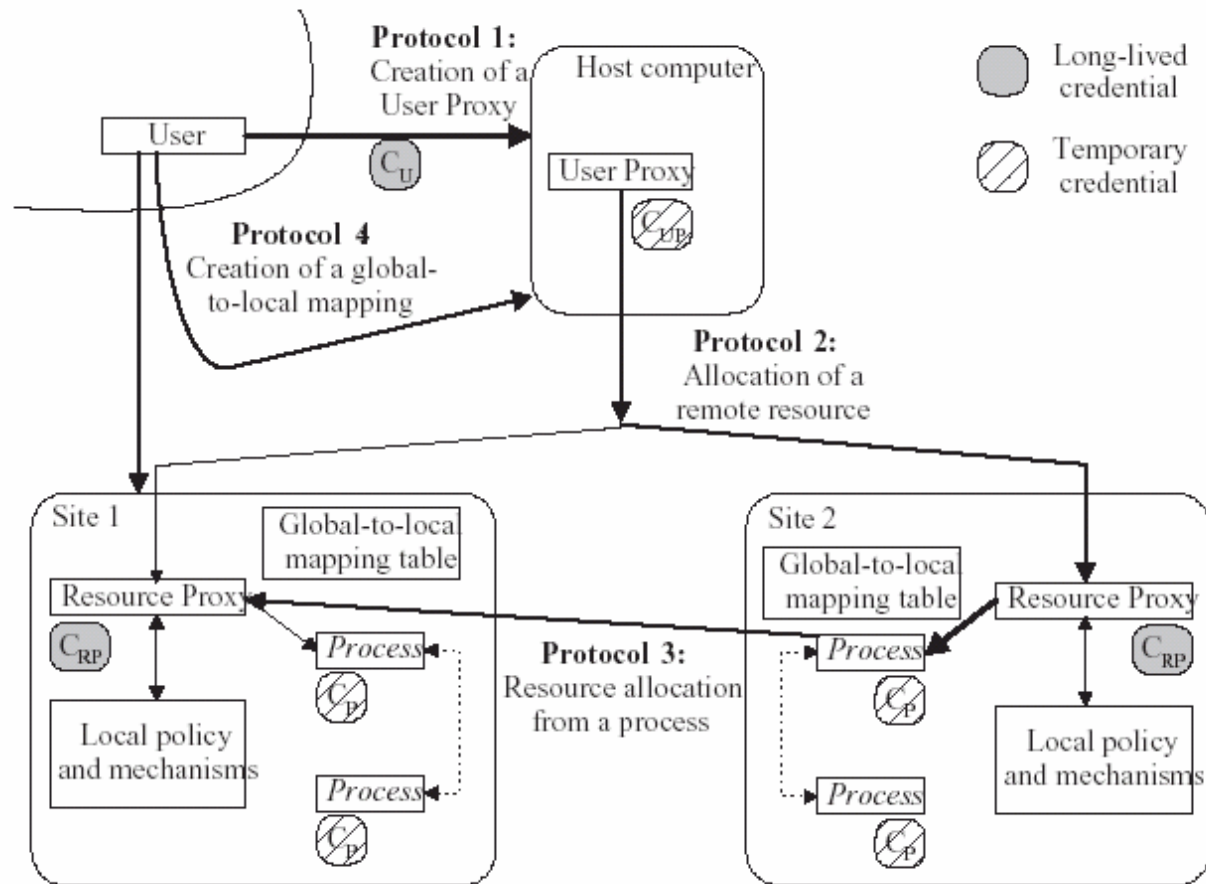
GSI Proxy Components

- **User Proxy**
 - Session manager process with permission to act on behalf of user for limited time
- **Resource Proxy**
 - Agent used to translate between interdomain security operations and local intradomain mechanisms

GSI Protocols

- User Proxy Creation Protocol
- Resource Allocation Protocol
- Resource Allocation from a Process Protocol
- Mapping Registration Protocol

Computational Grid Security Architecture

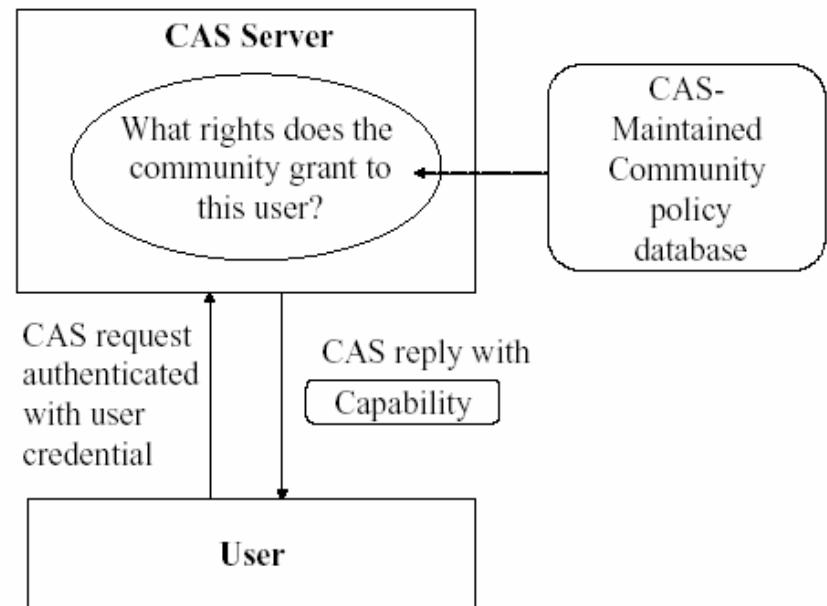


Community Authorization Service (CAS)

- Allows resource owners to grant access to blocks of resources to a community and lets the community manage fine-grained access control
- Community runs a CAS Server to track of membership & access policies
- CAS architecture builds on public key authentication provided by GSI
- Can be replicated to avoid single point of failure and bottleneck issues

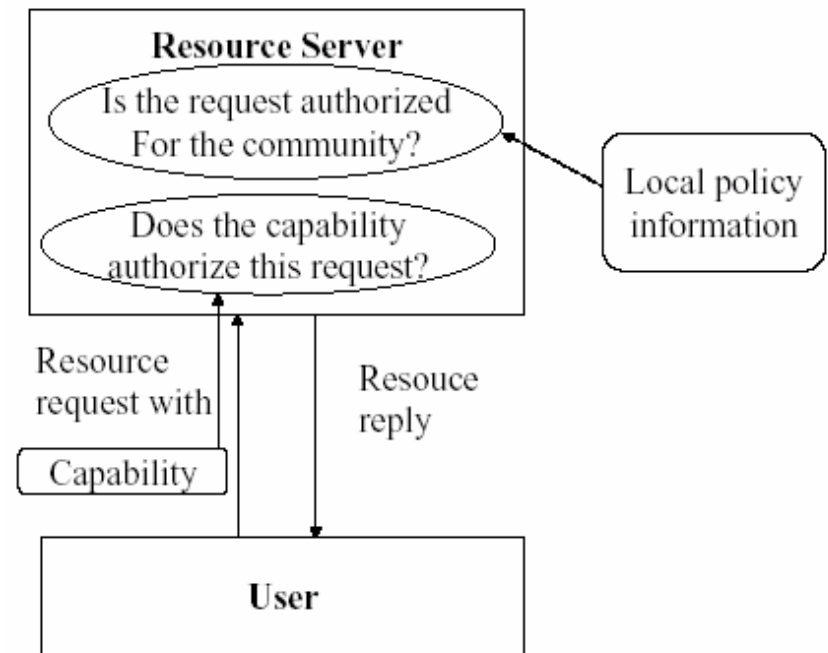
CAS Authorization

- CAS Server contains policy statements that specify who has permission, which resource or resource group the permission is granted on and what permission is granted
- User must first acquire a capability from CAS server.



CAS Resource Authentication

- The resource grants the user access to the local community resource based on local policy for the community and the community policy for the user
- Trust relationships reduced from $(C \times P)$ to $(C + P)$



Restricted Proxies

- GSI supports simple form of delegation
- CAS supports rich restriction policies to place specific limits on rights

Restricted Proxy Considerations

- Restricted proxy cannot delegate more authority than it has
- Effective validity time for a proxy certificate is the intersection of the validity times of all the certificates in the chain
- Restricted proxies do not provide a mechanism for revocation of proxy certificates.

Compromised Servers

- Compromised Resource Server
 - A very serious problem, but does not create cascaded security problems.
- Compromised CAS Server
 - Can issue credentials that do that reflect the policies of the community.
 - Can issue credentials that attempt to grant access to resources that don't belong to the community.

MyProxy

- Designed and developed to bridge the incompatibility between Web and Grid security
- Allows Grid Portals to use GSI-protected resources

Grid Portal Requirements

- Users must be able to use any standard web browser to access the Grid portals
- Users must be able to use a web browser from locations where their Grid credentials would not be available to them
- Users must be able to do anything through a Grid portal that their credentials allow them to do.

Software Constraints

- Grid Credentials are normally stored on a file system
- Not all applications are GSI-enabled
 - Web browsers are not GSI-enabled
- These restrictions forced Grid portals to give the portal permanent privileges.

Goals for MyProxy

- Allow users to access their credentials from anywhere on the Grid.
- Allow them to delegate credentials to resources to which they would normally would not be able to
- Remove any credentials from the portal except for when they are needed
- Be scalable. Multiple portals should be able to use a single system and a portal should be able to use multiple systems
- Give the user as much control of their credentials and proxy credentials as possible. Portals should not be able to get user credentials unless authorized by the user.

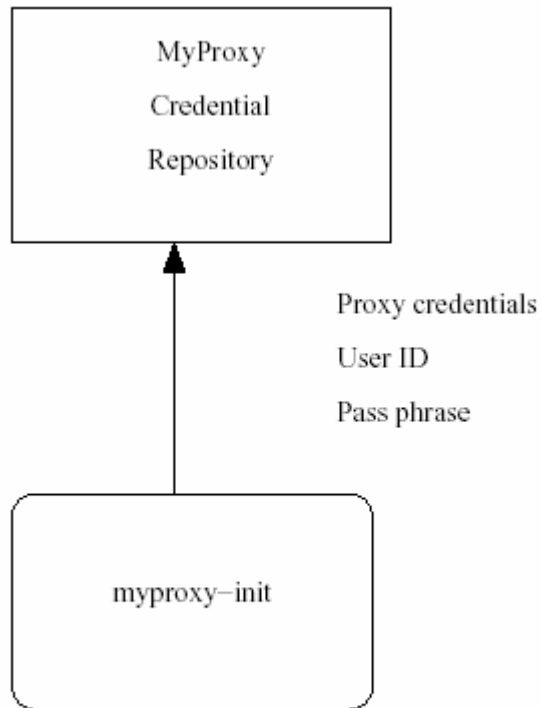


Figure 1. MyProxy-init process

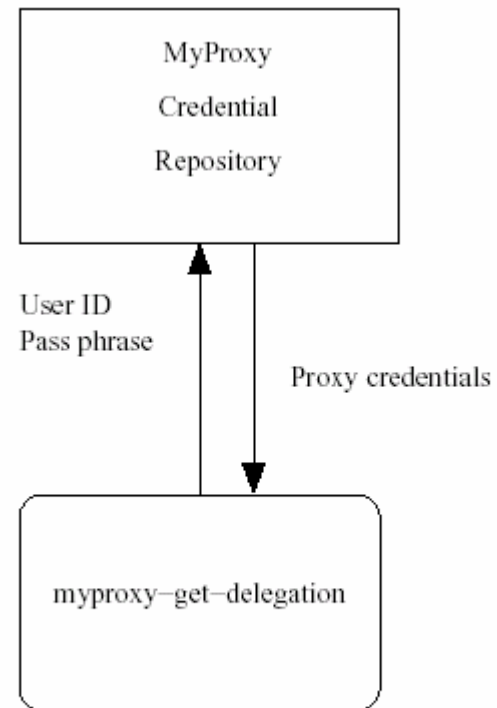


Figure 2. MyProxy retrieval process

MyProxy Authentication

