Globus Toolkit User Tutorial Part 2

Security and Remote Process Creation

The Globus Project Team

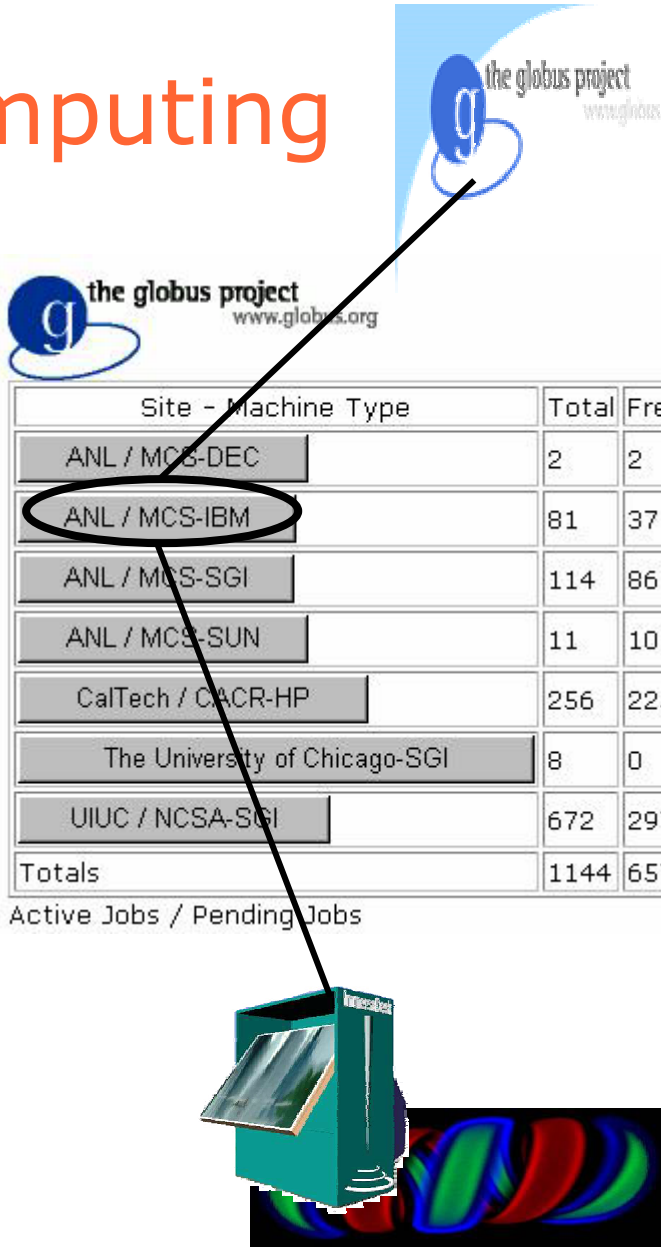
http://www.globus.org\

Desktop Supercomputing

Seamlessly, from the desktop

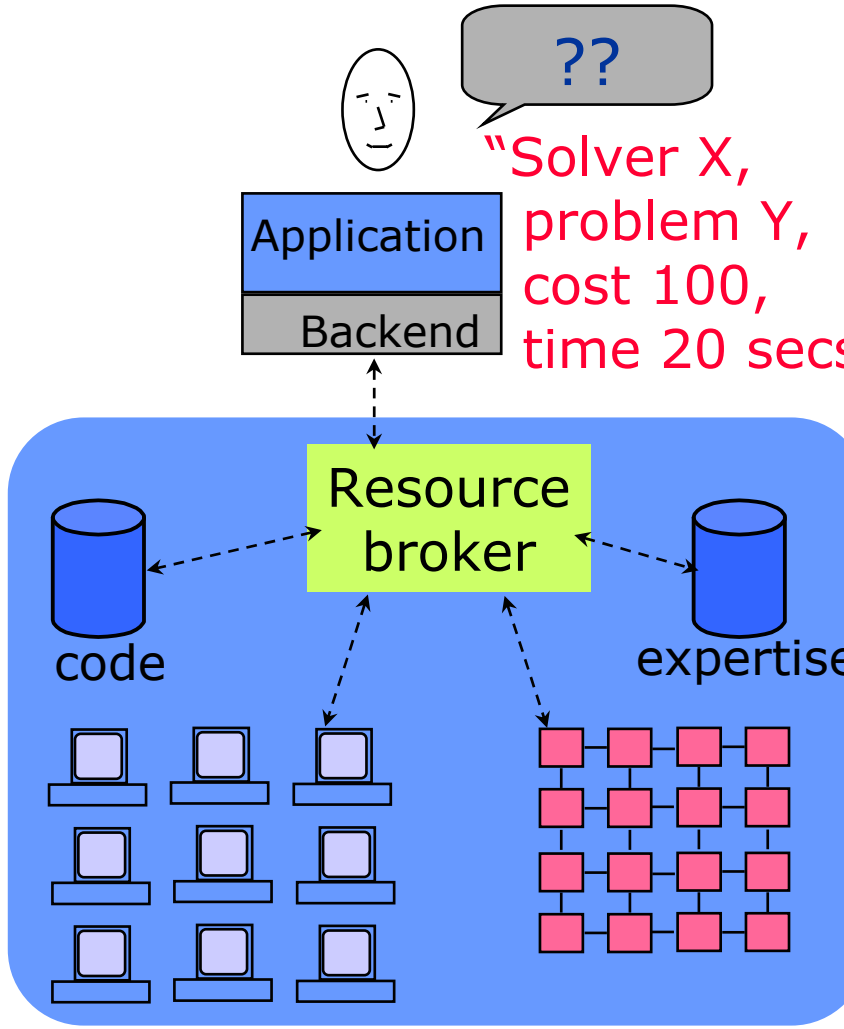
- Sign-on once
- Locate available computers
- Start computation on an appropriate system
- Monitor progress
- Get output files
- Manipulate locally

E.g. ECCE', Cactus, Hotpage, Chemical Eng. Workbench, WebFlow, LSA



Network-Enabled Servers

- Seamless access of remote resources
  - Examples: NEOS, NetSolve, Nimrod
- Issues
  - Scheduling for real-time & high-throughput
  - Code management & security
  - Algorithm design



Problems

- Security
  - How do we authenticate ourselves at the remote site?
- Resource specification
  - How do we locate and request a resource?
- Staging of code and data
  - How do we stage a user's executables and data to the remote resource?
- Computation
  - How do we start & manage computation?

The Globus Advantage

- Single sign-on for all resources
  - No need for user to keep track of accounts and passwords at multiple sites
  - No plaintext passwords
- Uniform interface to various local scheduling mechanisms
  - PBS, Condor, LSF, NQE, LoadLeveler, fork, etc.
  - No need to learn and remember obscure command sequences at different sites

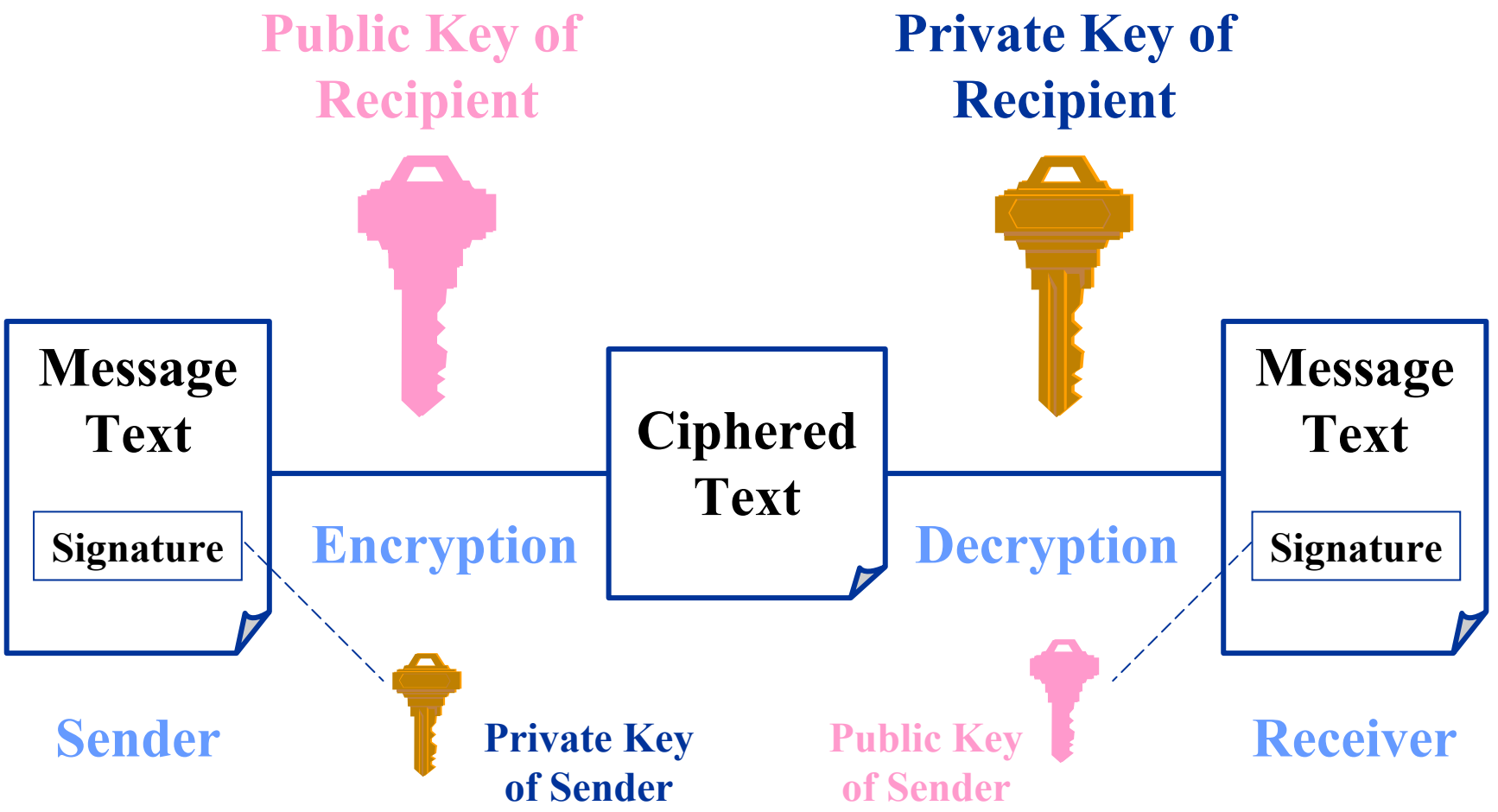
Authentication Model

- Authentication is done on a "user" basis
  - Single authentication step allows access to all grid resources
- No communication of plaintext passwords
- Most sites will use conventional account mechanisms
  - You must have an account on a resource to use that resource
- Sites may use "generic" Grid accounts
  - Not common, but Globus can deal with it

Grid Security Infrastructure

- Based on public key technology
  - Standard X.509 certificate, same as certificates used for the Web
- Each user has:
  - a Grid user id (called a Subject Name)
  - a private key (like a password)
  - a certificate signed by a Certificate Authority (CA)
- A "gridmap" file at each site specifies grid-id to local-id mapping

Encryption


Public Key of Recipient
Private Key of Recipient
Message Text
Signature
Encryption
Ciphered Text
Decryption
Message Text
Signature
Sender
Private Key of Sender
Public Key of Sender
Receiver

Certificate Based Authentication

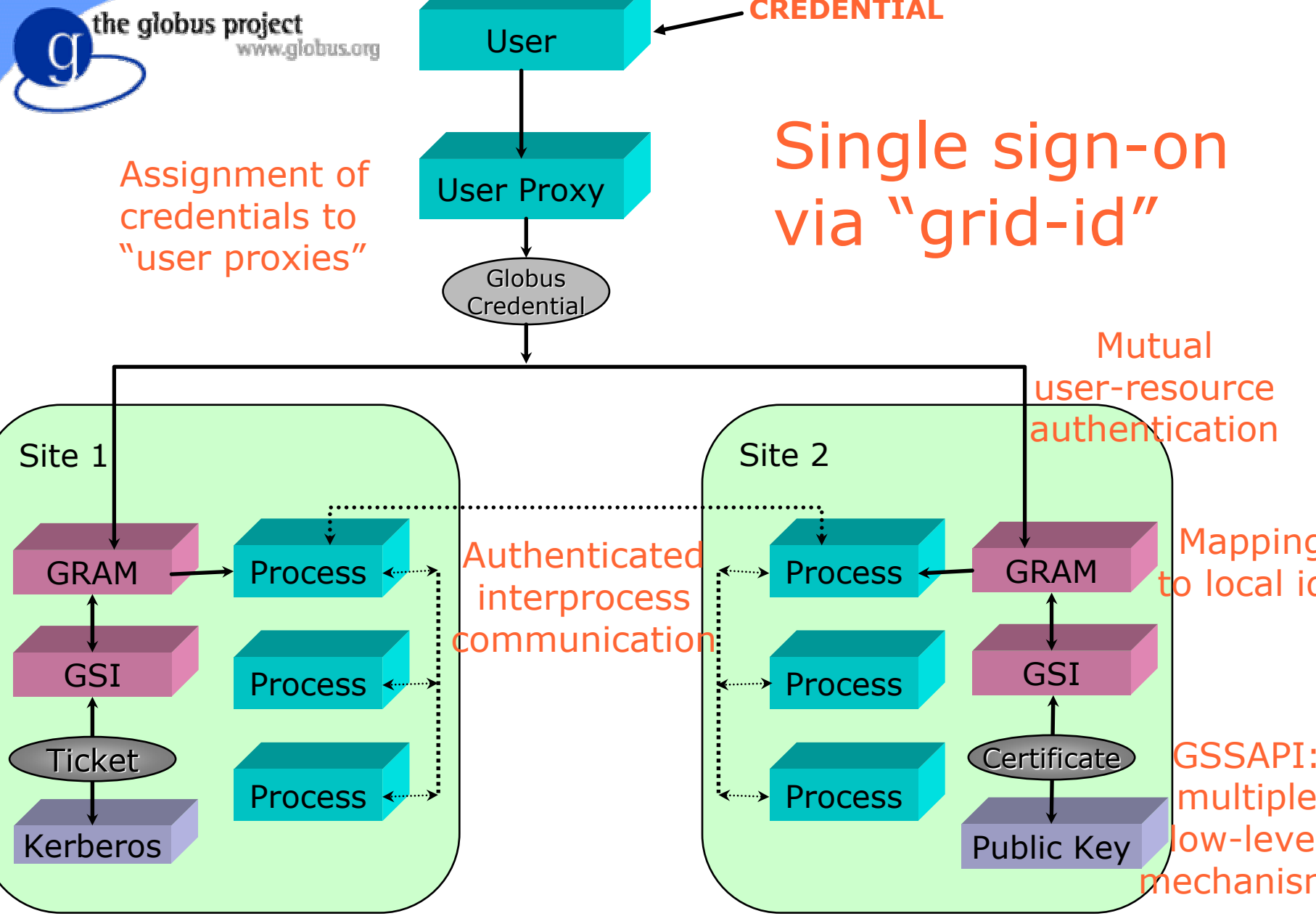
- User has a certificate, signed by a trusted "certificate authority" (CA)
  - Certificate contains users name and public key
  - Globus project operates a CA
- User's private key is used to encode a challenge string
- Public key is used to decode the challenge
  - If you can decode it, you know the user
- Treat your private key carefully!!
  - Private key is stored in encrypted form

User Proxies

- A temporary credential for use by our computations
  - We call this a user proxy certificate
  - Allows process to act on behalf of user
  - User-signed user proxy certificate stored in local file
- Minimize exposure of user's private key
- Proxy's private key is not encrypted
  - Rely on file system security, proxy certificate file must be readable only by the owner

Delegation

- Remote creation of a user proxy
- Allows remote process to act on behalf of the user
- Avoids sending passwords or private keys across the network


User
CREDENTIAL
User Proxy
Globus Credential
Assignment of credentials to "user proxies"
Single sign-on via "grid-id"
Mutual user-resource authentication
Site 1
Site 2
GRAM
GSI
Ticket
Kerberos
Process
Process
Process
Authenticated interprocess communication
Process
Process
Process
GRAM
GSI
Certificate
Public Key
Mapping to local id
GSSAPI: multiple low-level mechanisms

Globus Authentication Setup

- Before you can run Globus applications:
  - Install Globus
  - Obtain a Grid certificate and key
  - Set up your environment so Globus knows where to find certificates and keys
  - Contact sites to set up local accounts and globusmap entries
  - Create proxy certificate for each application run
- Documentation
  - http://www.globus.org/security