

THE US NATIONAL VIRTUAL OBSERVATORY

Interoperable Authentication And Authorization for the VO

Ray Plante
NVO@NCSA

Von Welch & Jim Basney
NCSA Grid Security Team

Mike Freemon & Randy Butler
**National Middleware Initiative (NMI)
Grid Center**

Background:

- Security 101 *Matthew Graham*
(see **teamwg mail archive**)
- VO-friendly, Community-based Authorization Framework, Pt. 1 *Ray Plante*
<http://chart.stsci.edu/twiki/pub/Main/TechWG/CommunityAuthorizationP1.pdf>
- A Trust Model for Security in the VO *Guy Rixon*
<http://wiki.astrogrid.org/bin/view/Astrogrid/TrustModelForVO>



What we'd like to see from interoperable security

- Trustworthy access to restricted resources
 - Proprietary data
 - Remote storage
 - CPU cycles
 - Sufficient control and auditing by resource providers
- Single sign-on
 - User enters a username/password once per session
 - Can access many restricted resources in an operation from different providers
 - Interoperates with public (non-restricted) data and services seamlessly
 - Interoperability with Grid security
- A framework that can be leveraged by observatories
 - A common way to get at proprietary data
 - A common way to support security in portals

What we'd *not* like to see from interoperable security

- A framework so cumbersome that no one uses it
 - Users & service providers
 - *"Why do we even need this?"*
- A hacking incident that erodes trust in the system
 - We're used to relying on goodwill
 - The VO will raise our profile to hackers

Certificate-based Security

- Beyond the browser: Good for “grid” applications
 - Web services talking to each other
 - Can handle users across administrative boundaries
- X.509 Certificates in wide use today
 - To support SSL connections
 - Libraries available
- Certificates presented at socket connection time assure identity of holder
 - Identities of user and service
 - Each end must already have copy of Certificate Authority's certificate
 - Each end holds own private key; cert contains public key
 - “handshake” tests that each side has private key corresponding to public key in cert

Certificate Authorities: the root of Trust

- A service/user will trust a CA by acquiring the CA's public key
 - Public key acquired "out-of-band" of a user request
 - Benefit from a small number of CAs
- **Recommendation:** Each VO project runs its own CA
 - IVOA level: trust chain too hard to track
 - Trust model to be discussed later...*
 - Organization level: too many CAs
- **Recommendation:** Accept other established CAs from scientific community as a matter of practice
 - DOE, NCSA/TeraGrid
 - Leverage their trust model

Certificate Management

- User-managed certificates
 - User stores certificate & private key on local machine
 - User must “load” certificate into client applications
 - e.g. web browser, Globus
 - User must also “load” certificates of trusted CAs
 - Often considered part of the “pain” of certificates
- Portal-managed certificates
 - Portal manages certificate & private key on users behalf
 - users don’t have to know certificates are being used
 - Possible to pass cert to client apps when needed

Portal-Managed Certificates



Portal

*Registration
Service*

CA

*Certificate
Repository
(MyProxy)*

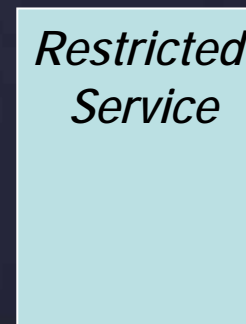
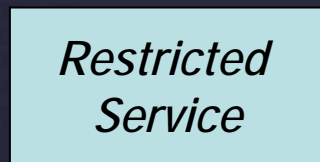
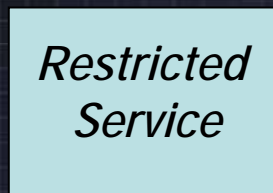
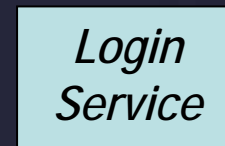
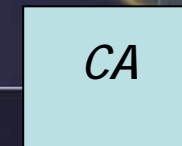
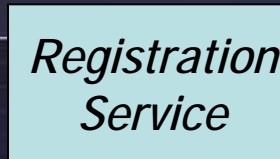
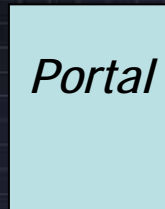
*Login
Service*

*Restricted
Service*

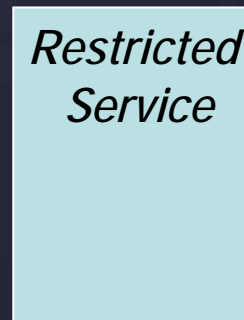
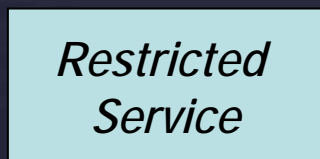
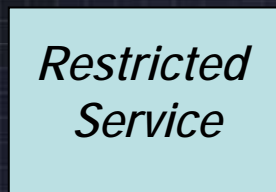
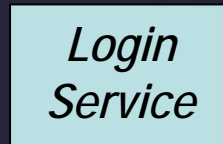
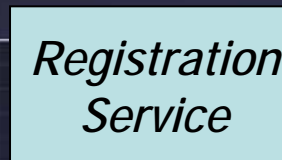
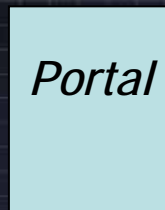
*Restricted
Service*

*Restricted
Service*

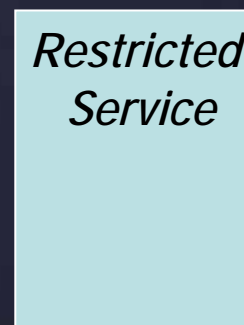
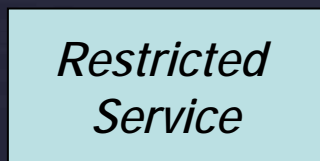
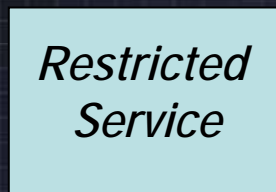
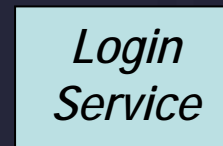
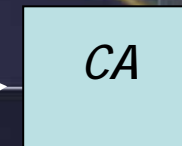
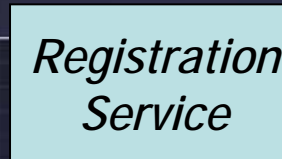
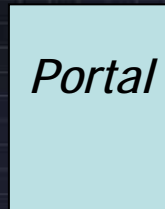
Portal-Managed Certificates



Portal-Managed Certificates



Portal-Managed Certificates



Portal-Managed Certificates



Portal

*Registration
Service*

CA

*Certificate
Repository
(MyProxy)*



*Login
Service*

*Restricted
Service*

*Restricted
Service*

*Restricted
Service*

Portal-Managed Certificates



Portal

*Registration
Service*

CA

*Certificate
Repository
(MyProxy)*



*Login
Service*

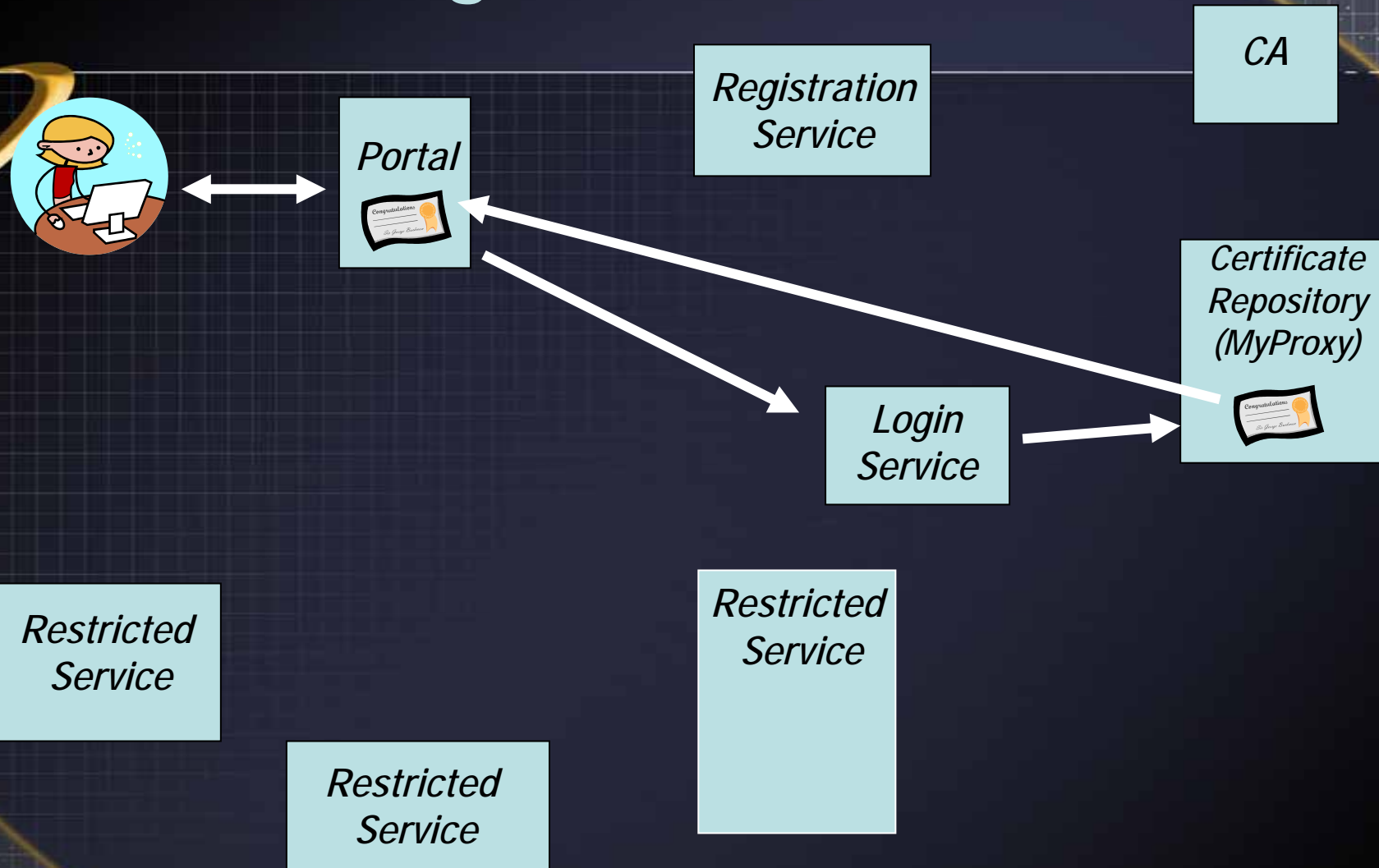


*Restricted
Service*

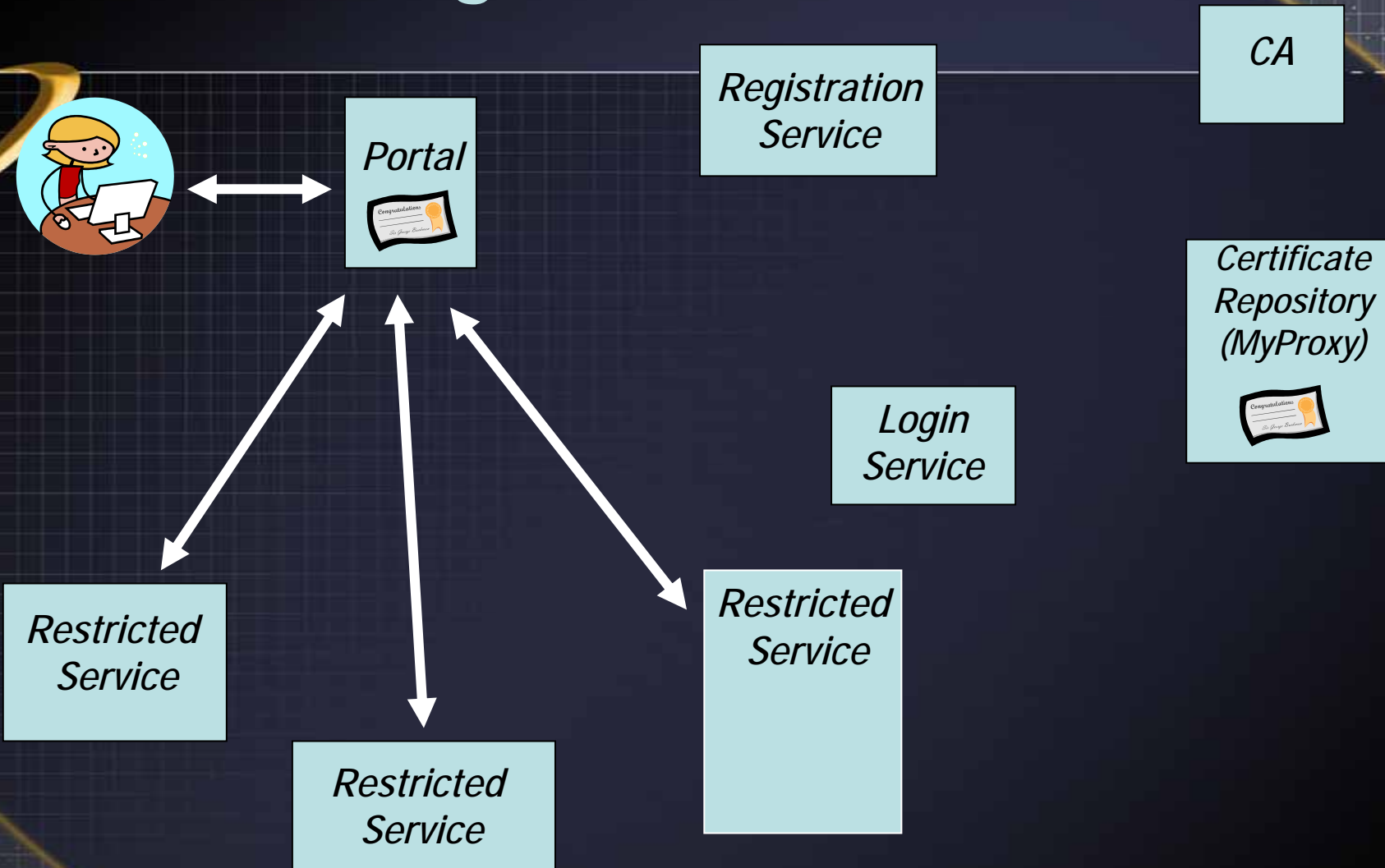
*Restricted
Service*

*Restricted
Service*

Portal-Managed Certificates



Portal-Managed Certificates



Certificate Management

- Both certificate models will be important
 - Important to support portal-based management for ease of use
 - The savvier users will manage own certs when useful
- **Recommendation:** NVO runs the following services:
 - Registration service (to create certs)
 - Certificate Repository (MyProxy) to store certs for use with portals
 - A login service for use with portals
 - Cert Download service: for retrieving certs for local client apps

Existing tools: GAMA (SDSC), PURSE (NMI)



Portal-Managed Certificates



Portal

*Local
Registration
Service*

*NVO
Registration
Service*

CA



*Certificate
Repository
(MyProxy)*

*Login
Service*

*Restricted
Service*

*Restricted
Service*

*Restricted
Service*



Portal-Managed Certificates



Portal



*Local
Registration
Service*

*NVO
Registration
Service*

CA



*Certificate
Repository
(MyProxy)*

*Login
Service*

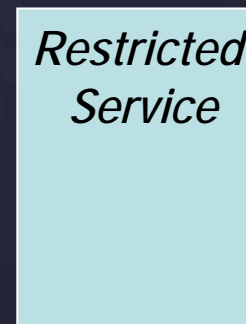
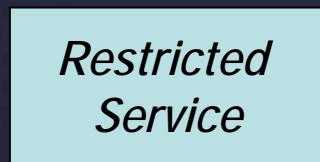
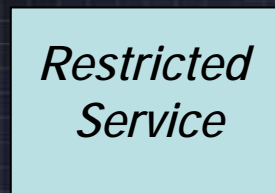
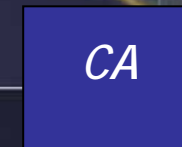
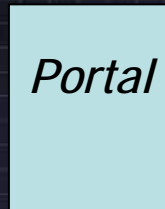
*Restricted
Service*

*Restricted
Service*

*Restricted
Service*



Portal-Managed Certificates



Portal-Managed Certificates



Portal



*Local
Registration
Service*



*NVO
Registration
Service*



CA



*Certificate
Repository
(MyProxy)*



*Login
Service*

*Restricted
Service*

*Restricted
Service*

*Restricted
Service*



Portal-Managed Certificates



Portal

*Local
Registration
Service*

*NVO
Registration
Service*

CA



*Certificate
Repository
(MyProxy)*



*Login
Service*

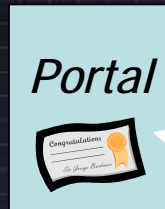
*Restricted
Service*

*Restricted
Service*

*Restricted
Service*



Portal-Managed Certificates



*Local
Registration
Service*

*NVO
Registration
Service*

CA



*Certificate
Repository
(MyProxy)*

*Login
Service*



*Restricted
Service*

*Restricted
Service*

*Restricted
Service*



Trust Model

- The certificate “handshake”
 - Means that...
 - User has private key issued to “Joe Astronomer”
 - Service has private key issued to “Fab Storage”
 - Useful enough in some cases
 - Service can save per user state across sessions
 - Will control what user can do
 - Did “Joe” give a phony name?
- The CA signature means that the CA has made an effort to verify the identity
 - How does a CA determine that users are who they say they are?

Commercial Trust Model: Thawte

- Weak certificates
 - Generic identity: name not included
 - Can do low trust activities (encrypt email)
- Strong certificates (has identity info in it):
 - Identity notarized by multiple somewhat-trusted people—"Web of Trust"
 - Point system
 - a notary can issue up to 35 points
 - Need 50 points to get strong cert

An Astronomical Observatory

To gain access to telescope and resulting proprietary data

- User writes a proposal
 - Includes identity information: affiliation, email
 - Includes references to published work
 - Includes collaborators with references
- Observatory contacts PIs, Co-Is by email
- Proposal is evidence that they are legit.

Can we leverage this process?



Building a Trust Model

- How do we verify identities?
 - someone NVO trusts tells them (out-of-band) that the user is who she says she is.
 - Registration Authorities: enlisting the help of existing institutions
 - Academic departments? (akin to Shibboleth model)
 - Observatories?
 - How much is enough?
 - Faxed driver's license?
- Verification takes time and human involvement!
 - Can weak certificate enable some activity requiring less trust?
Allows user to get started right away!
- Need trusted system to issues certs to services
 - So that users can trust services to behave with their ID
 - Can build on trust model for users

Authorization

- Authorization policies tend to be service specific
 - Provider will want to control who's allowed to do what
 - Expect observatories to centralize authorization management
 - Authorization policies assigned to groups
 - Observatory: groups defined on the proposal level
 - Groups defined around archive-based research
- **Recommendation:** Leave authorization management to service providers. No interoperable standards are needed at this time

Three models for handling authorization

- Map external identities to internal accounts
 - One account per group, groups define set of authorization policy (coming up next from Neha!)
- Call out:
 - User accesses service with personal cert
 - Service calls out to authorization database to retrieve privileges associated with user
 - Service tests user request against policies
- Certificate wrapping (Globus CAS):
 - User visits authorization database with personal cert, gets back proxy cert with policies encoded inside (using SAML)
 - User presents proxy cert to service
 - Service test user request against policies
 - Good for portal-based access but inconvenient for outside clients