Grid Security: General Considerations and Globus Specifics

Thomas Sandholm <sandholm@pdc.kth.se>

KTH VETENSKAP OCH KONST
CENTER FOR PARALLEL COMPUTERS

Outline

- **Security Basics**
- What is Grid Security? What makes it different?
- Current Grid Security?
- OGSA Security and Web Services Security
- Globus Security

Security Terminology

- Authentication: Establishing identity
- Authorization: Establishing rights
- Message protection
  - Message integrity
  - Message confidentiality
- Digital signature
- Auditing

Public Key Security Terminology

- Certificate
- Certificate Authority (CA)
- Private and Public Key
- Public Key Infrastructure (PKI)

Outline

- Security Basics
- **What is Grid Security? What makes it different?**
- Current Grid Security?
- OGSA Security and Web Services Security
- Globus Security

What is Grid Security?

The Grid problem is to enable

"**coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations**."

From The Anatomy of the Grid

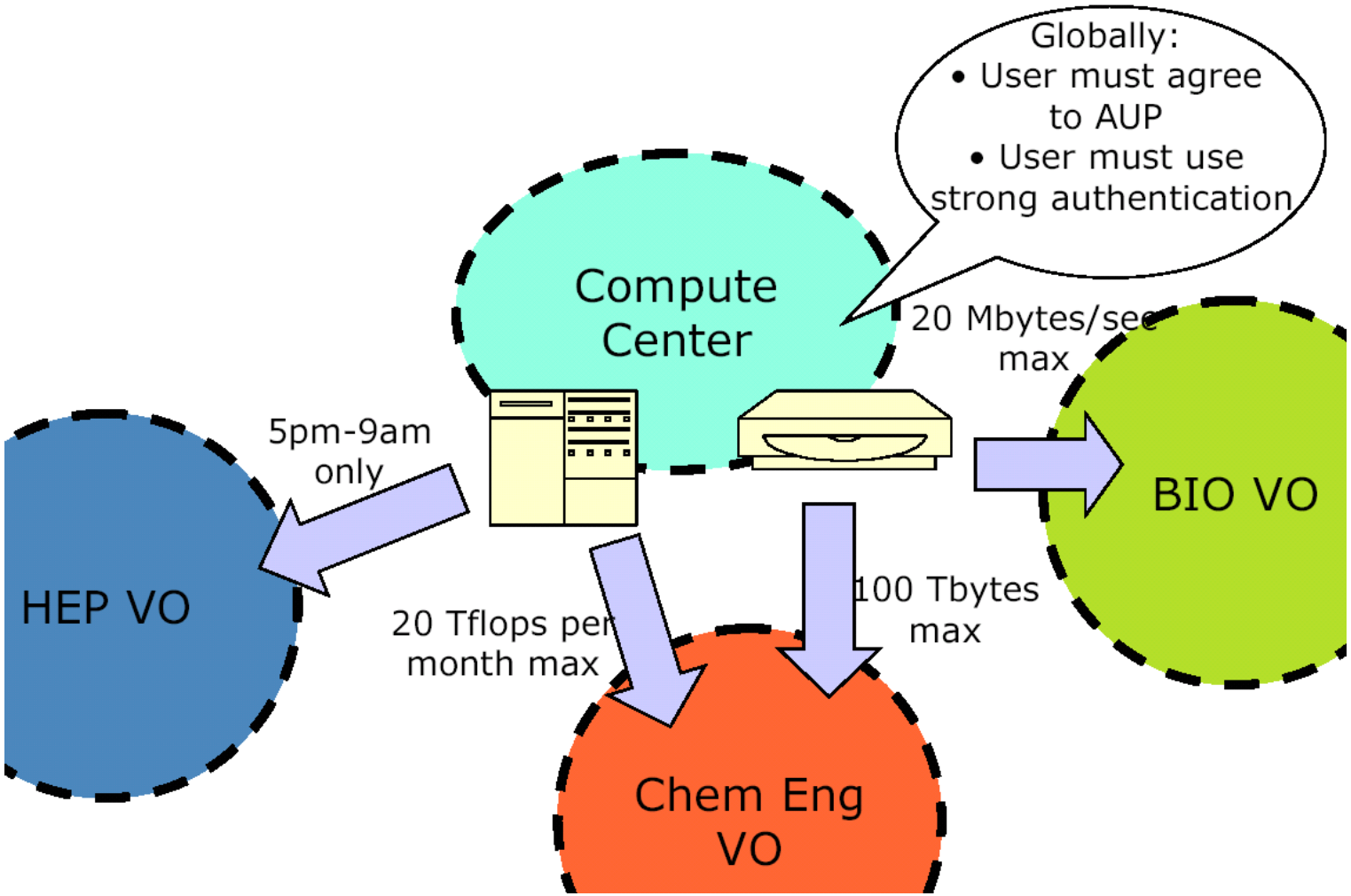
- So Grid Security is security to enable VOs
- What is needed in terms of security for a VO?

Resource Sharing

"…coordinated **resource sharing** and problem solving in dynamic, multi-institutional virtual organizations."

- Resources being used are still owned by their respective organization and subject to its policies
  - Sharing may be controlled amongst a number of VOs
  - Non-trivial policy in regards to QoS, QoP, etc.

Controlled Resource Sharing



Requires Coordination by VO

"…**coordinated** resource sharing and problem solving in dynamic, multi-institutional virtual organizations."

- Resources contributed to VO need to be coordinated by the VO in order to work together effectively.
  - All need to have a coherent policy in order to interoperate
  - Requires policy from VO back to resources

Dynamic Users, Resources, Policies

"…coordinated resource sharing and problem solving in **dynamic**, multi-institutional virtual organizations."

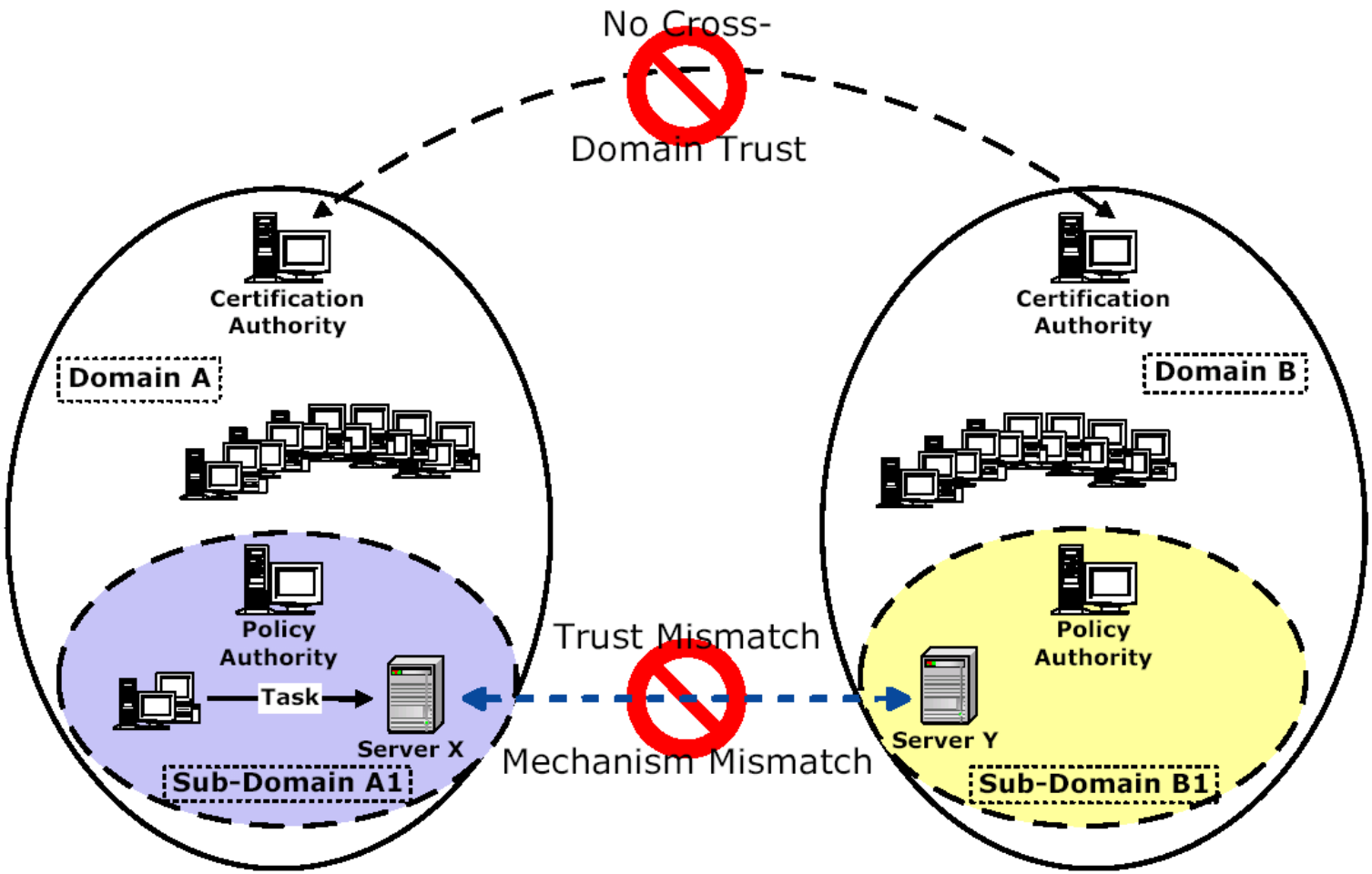
- Users, resources may be large, unpredictable, and changing at any point
- Roles of both may also be distinct and dynamic (not all users are equal).
- Doesn't allow for static configuration

Multiple Organizations, Mechanisms, Policies

"…coordinated resource sharing and problem solving in dynamic, **multi-institutional** virtual organizations."

- Each resource and user will have local policies and technologies that cannot be replaced by the VO
- Cannot assume cross-organizational trust relationships

Multi-Institution Issues



Why Grid Security is so Hard

- Resources being used may be valuable & the problems being solved sensitive
  - Both users and resources need to be careful
- Dynamic formation and management of virtual organizations (VOs)
  - Large, dynamic, unpredictable…
- VO Resources and users are often located in distinct administrative domains
  - Can't assume cross-organizational trust agreements
  - Different mechanisms & credentials
    - X.509 vs Kerberos, SSL vs GSSAPI, X.509 vs. X.509 (different domains), X.509 attribute certs vs. SAML assertions

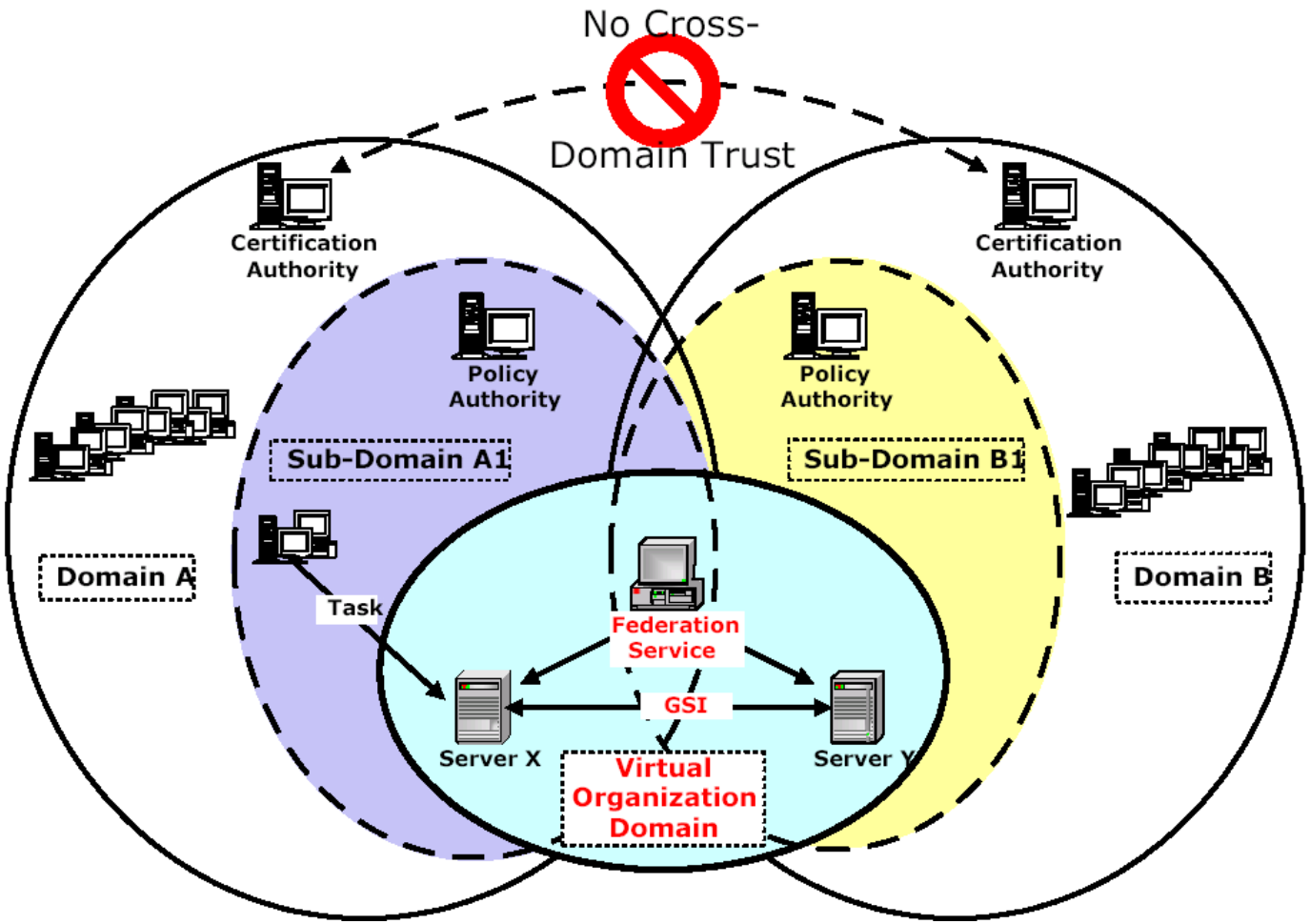
Why Grid Security is Hard…

- Interactions are not just client/server, but service-to-service on behalf of the user
  - Requires delegation of rights by user to service
  - Services may be dynamically instantiated
- Standardization of interfaces to allow for discovery, negotiation and use
- Implementation must be broadly available & applicable
    - Standard, well-tested, well-understood protocols; integrated with wide variety of tools
- Policy from sites, VO, users need to be combined
  - Varying formats
- Want to hide as much as possible from applications!

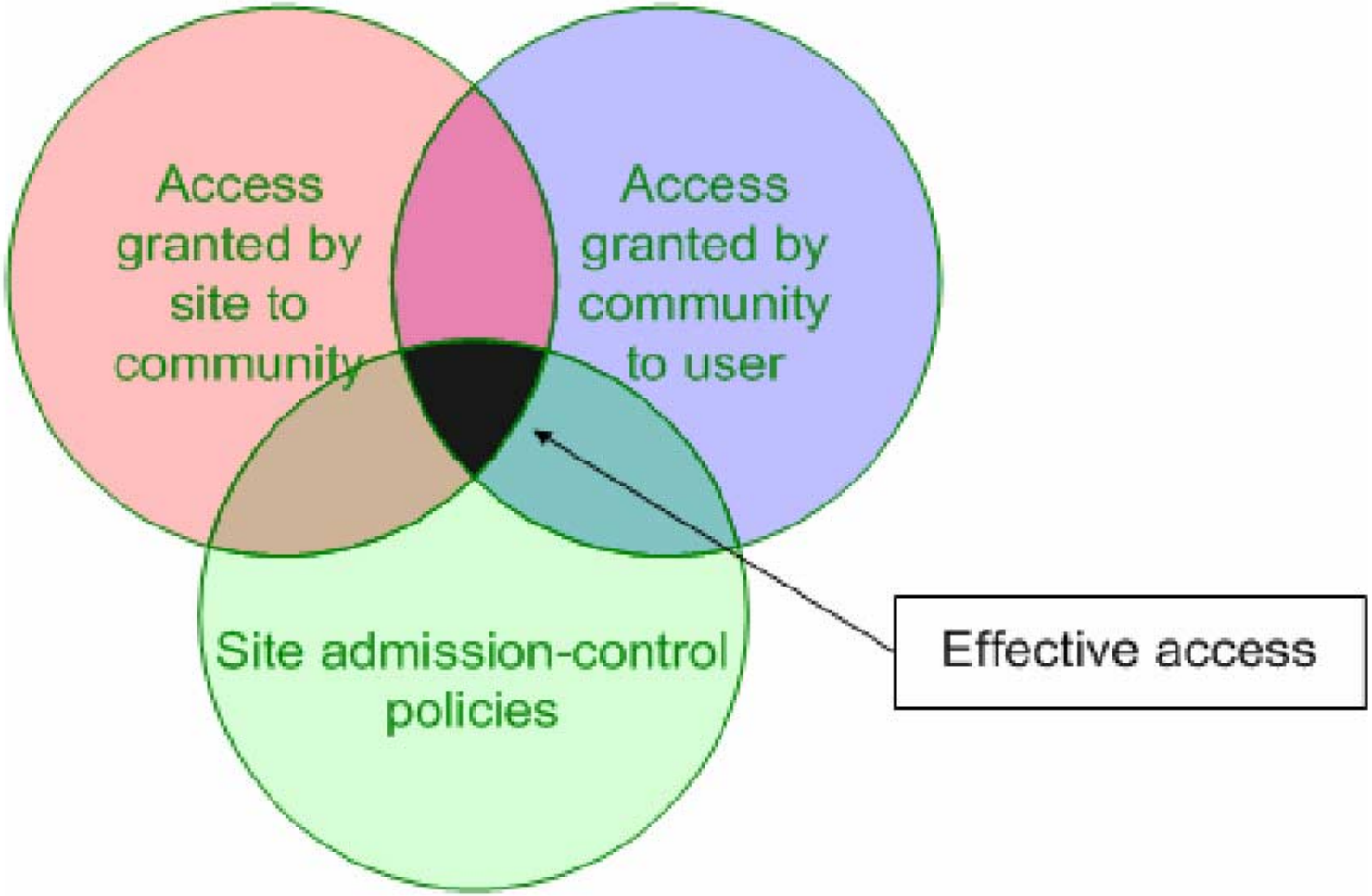
The Grid Trust Solution

- Instead of setting up trust relationships at the organizational level (lots of overhead, possible legalities -expensive!) set up trust at the user/resource level
- Virtual Organizations (VOs) for multi-user collaborations
  - Federate through mutually trusted services
  - Local policy authorities rule
- Users able to set up dynamic trust domains
  - Personal collection of resources working together based on trust of user

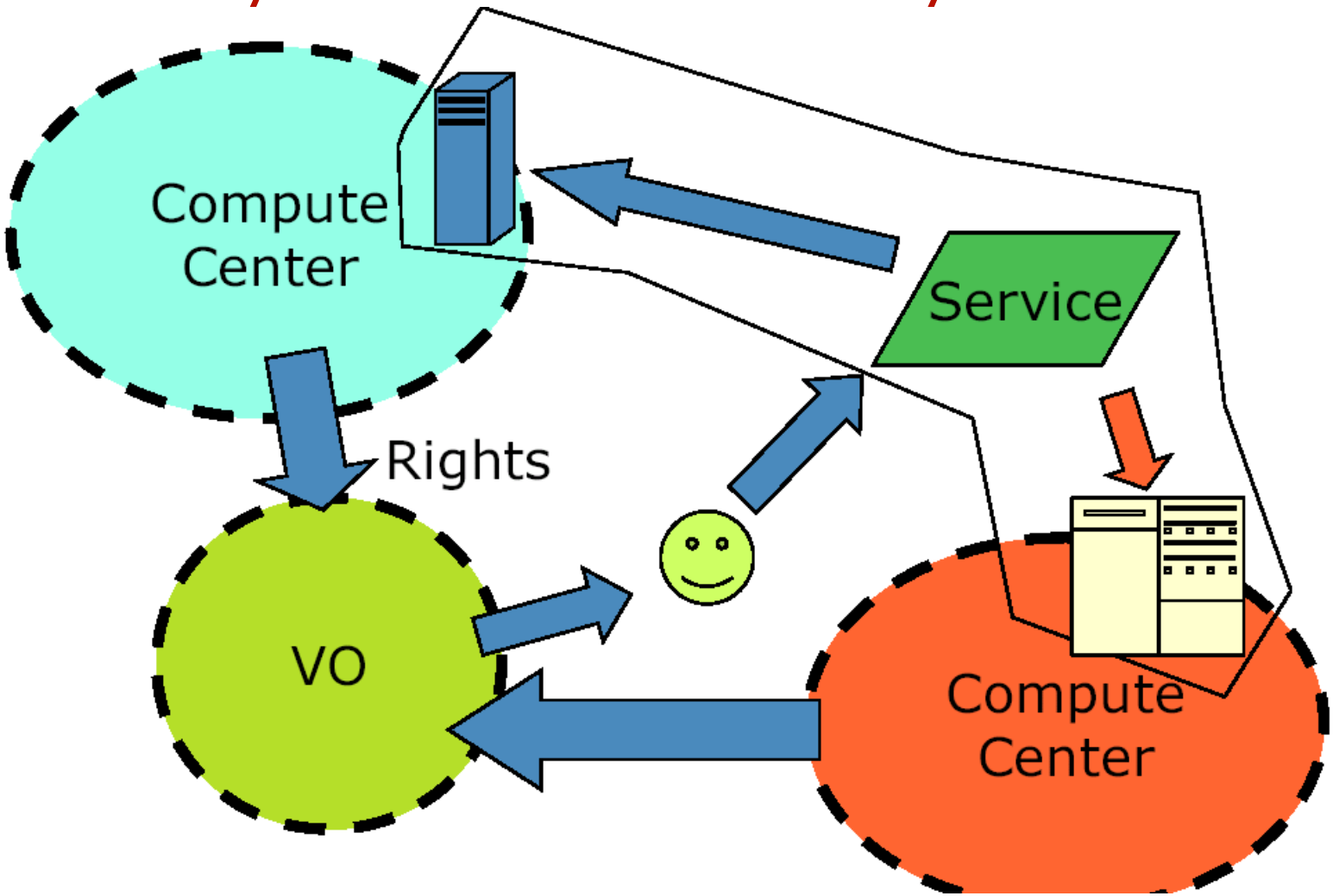
Grid Solution:
Use Virtual Organization as Bridge



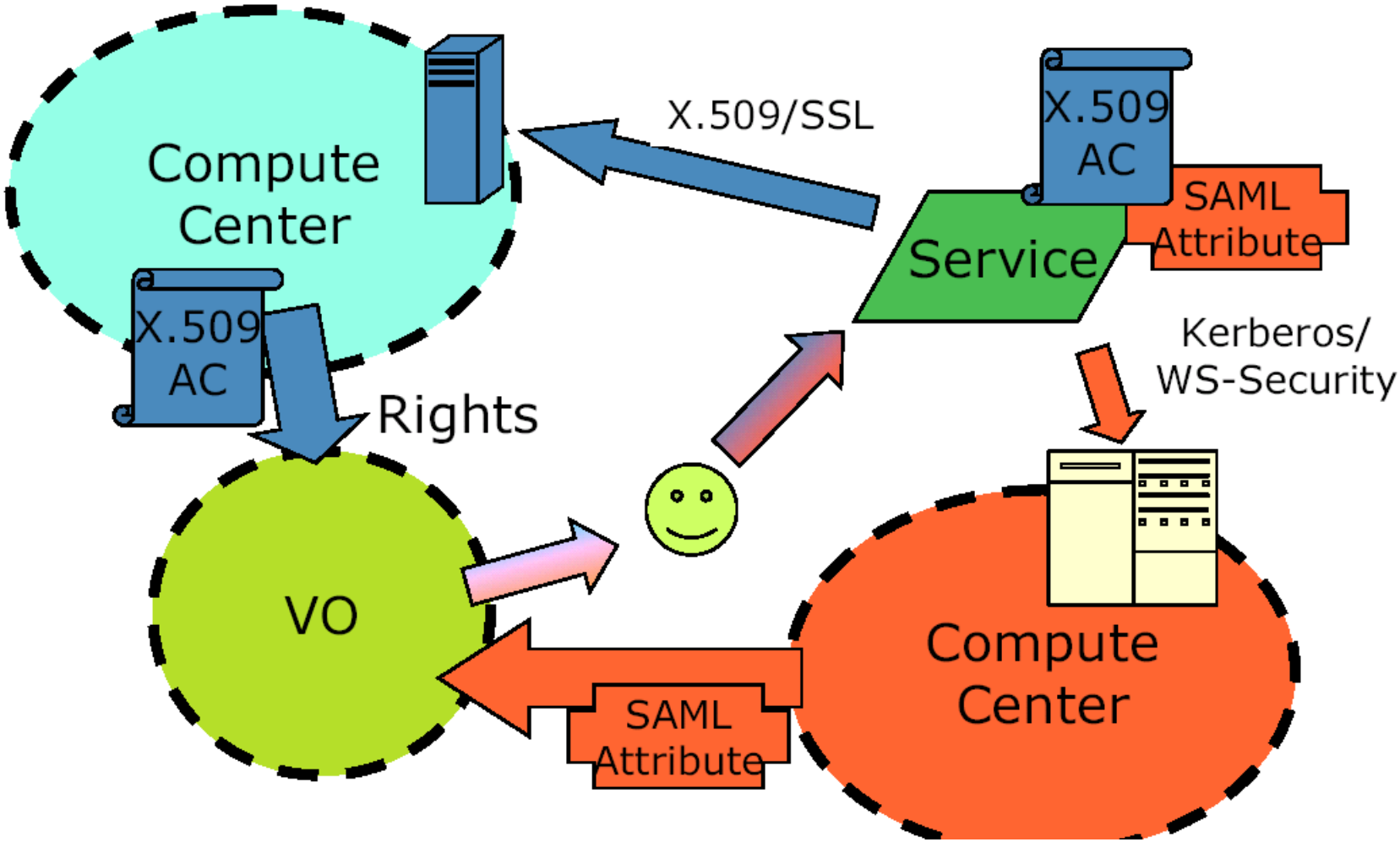
Effective Policy Governing Access Within a Collaboration



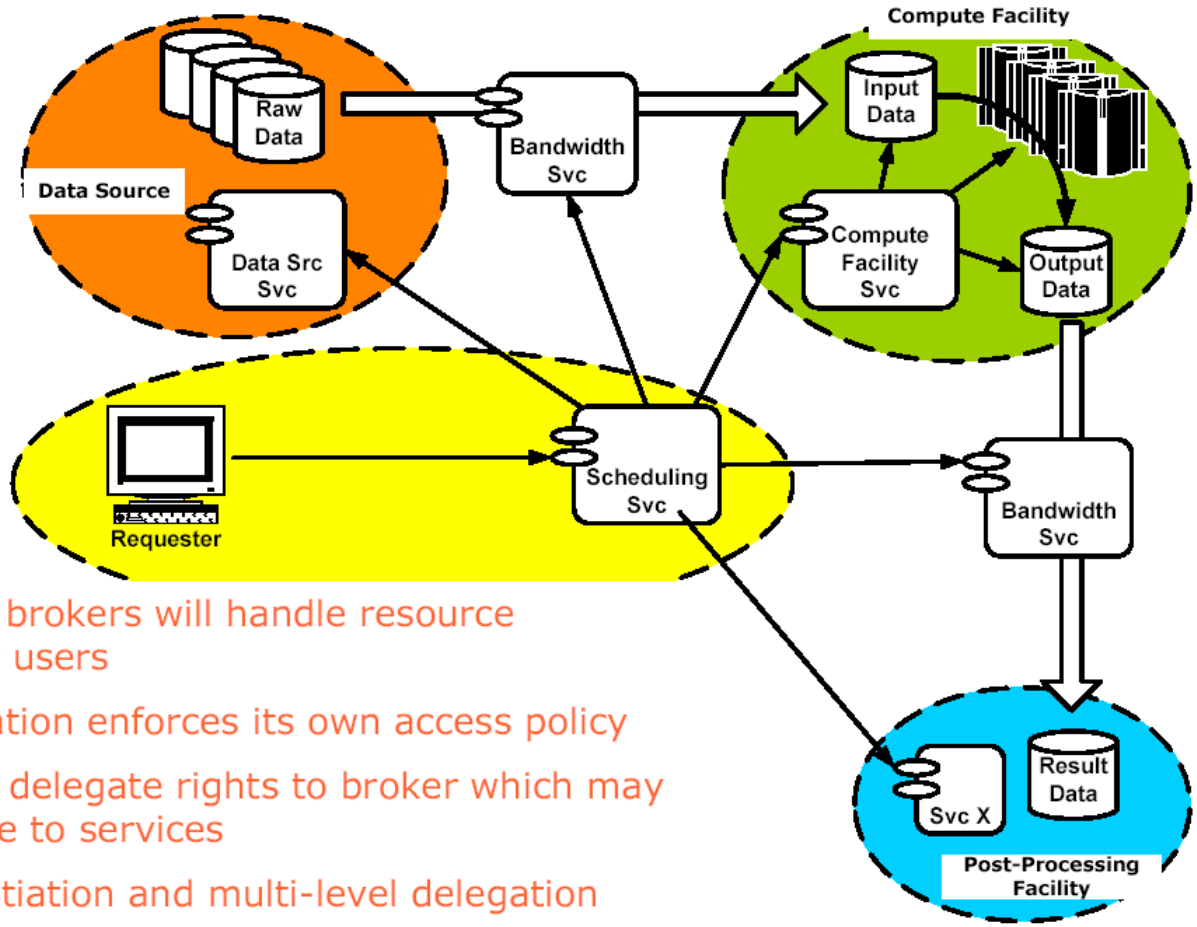
Use Delegation to Establish Dynamic Distributed System



Goal is to do this with arbitrary mechanisms

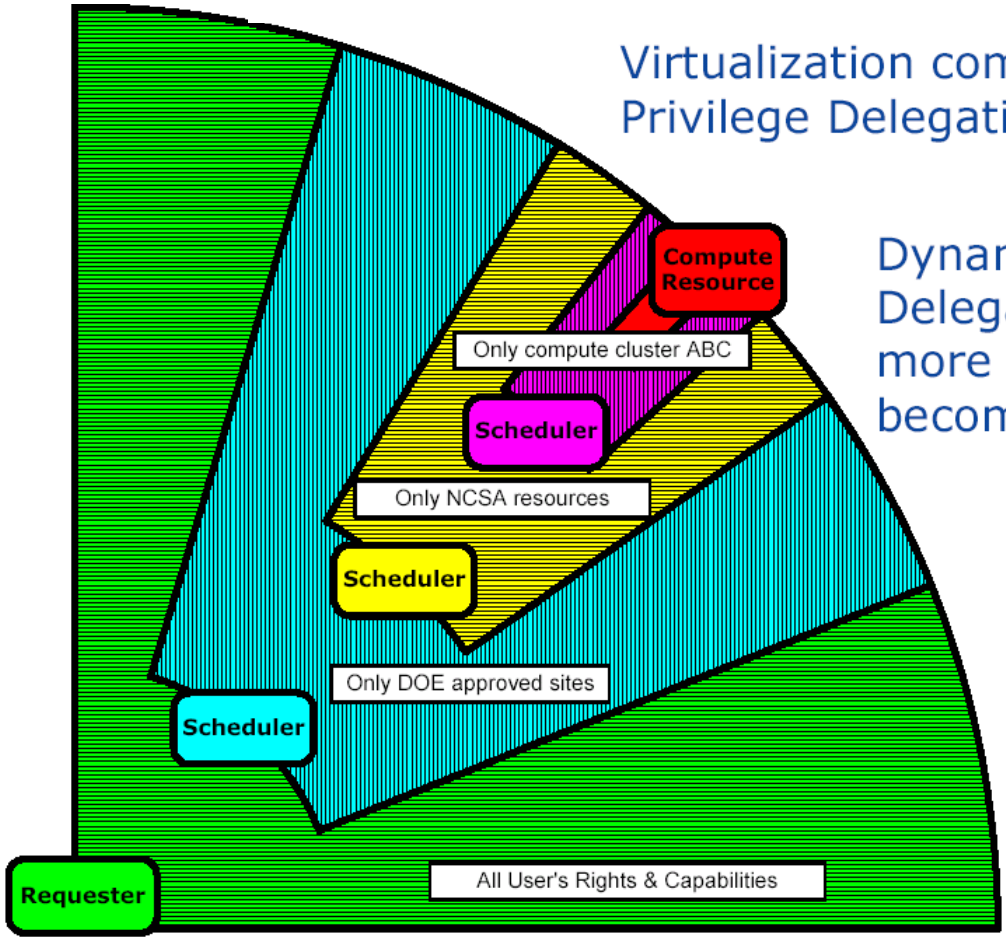


Security of Grid Brokering Services



- It is expected brokers will handle resource coordination for users
- Each Organization enforces its own access policy
- User needs to delegate rights to broker which may need to delegate to services
- QoS/QoP Negotiation and multi-level delegation

Propagation of Requester's Rights through Job Scheduling and Submission Process



Virtualization complicates Least Privilege Delegation of Rights

Dynamically limit the Delegated Rights more as Job specifics become clear

Trust parties downstream to limit rights for you… or let them come back with job specifics such that you can limit them

Grid Security must address…

- Trust between resources without organization support
- Bridging differences between mechanisms
  - Authentication, assertions, policy…
- Allow for controlled sharing of resources
  - Delegation from site to VO
- Allow for coordination of shared resources
  - Delegation from VO to users, users to resources
- ...all with dynamic, distributed user communities and least privilege.

Outline

- Security Basics
- What is Grid Security? What makes it different?
- **Current Grid Security?**
- OGSA Security and Web Services Security
- Globus Security

Grid Security Infrastructure (GSI)

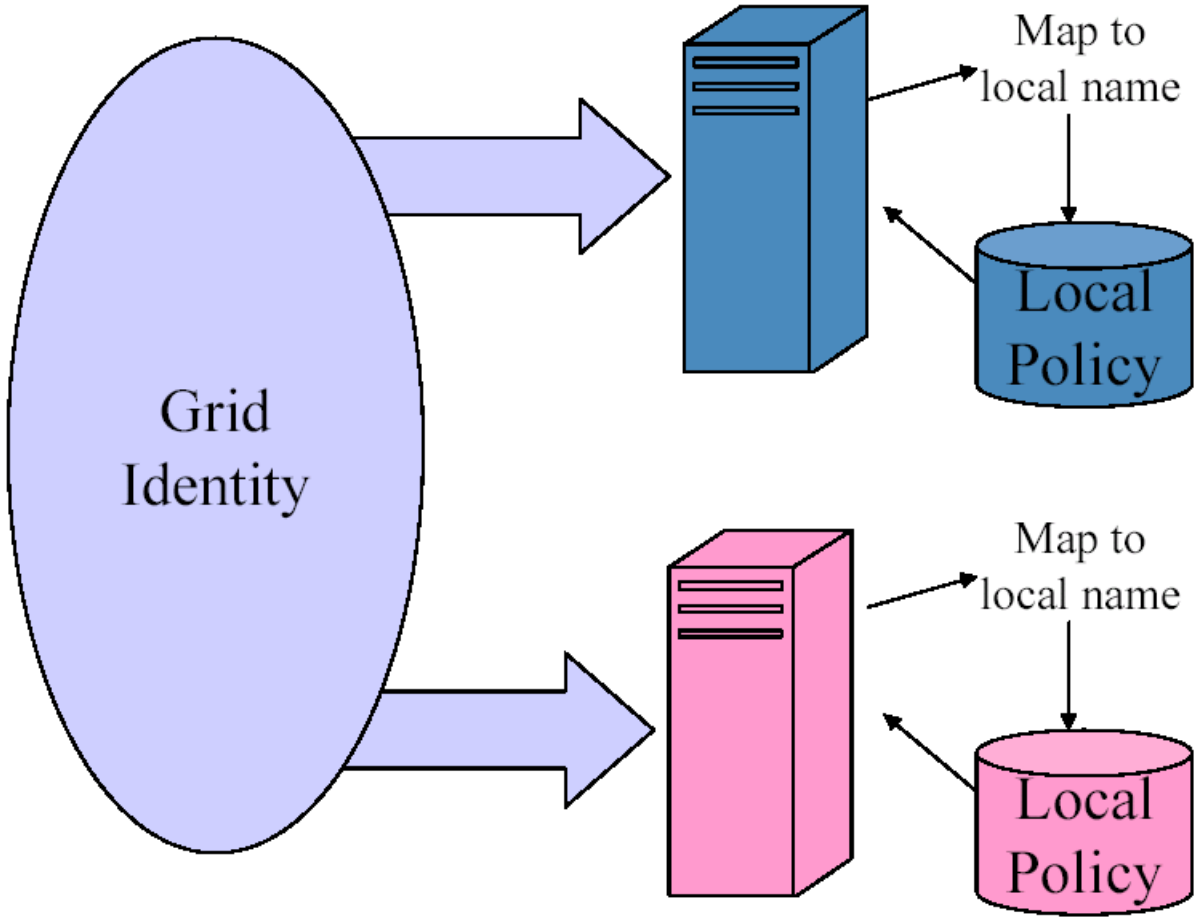
- Use GSI as a standard mechanism for bridging disparate security mechanisms
  - Doesn't solve trust problem, but now things talk same protocol and understand each other's identity credentials
  - Basic support for delegation, policy distribution
- Translate from other mechanisms to/from GSI as needed
- Convert from GSI identity to local identity for authorization

Grid Security Infrastructure (GSI)

- Based on standard PKI technologies
  - SSL protocol for authentication, message protection
  - CAs allow one-way, light-weight trust relationships (not just site-to-site)
- X.509 Certificates for asserting identity
  - for users, services, hosts, etc.
- Proxy Certificates
  - GSI extension to X.509 certificates for delegation, single sign-on

Grid Identity, Local Policy

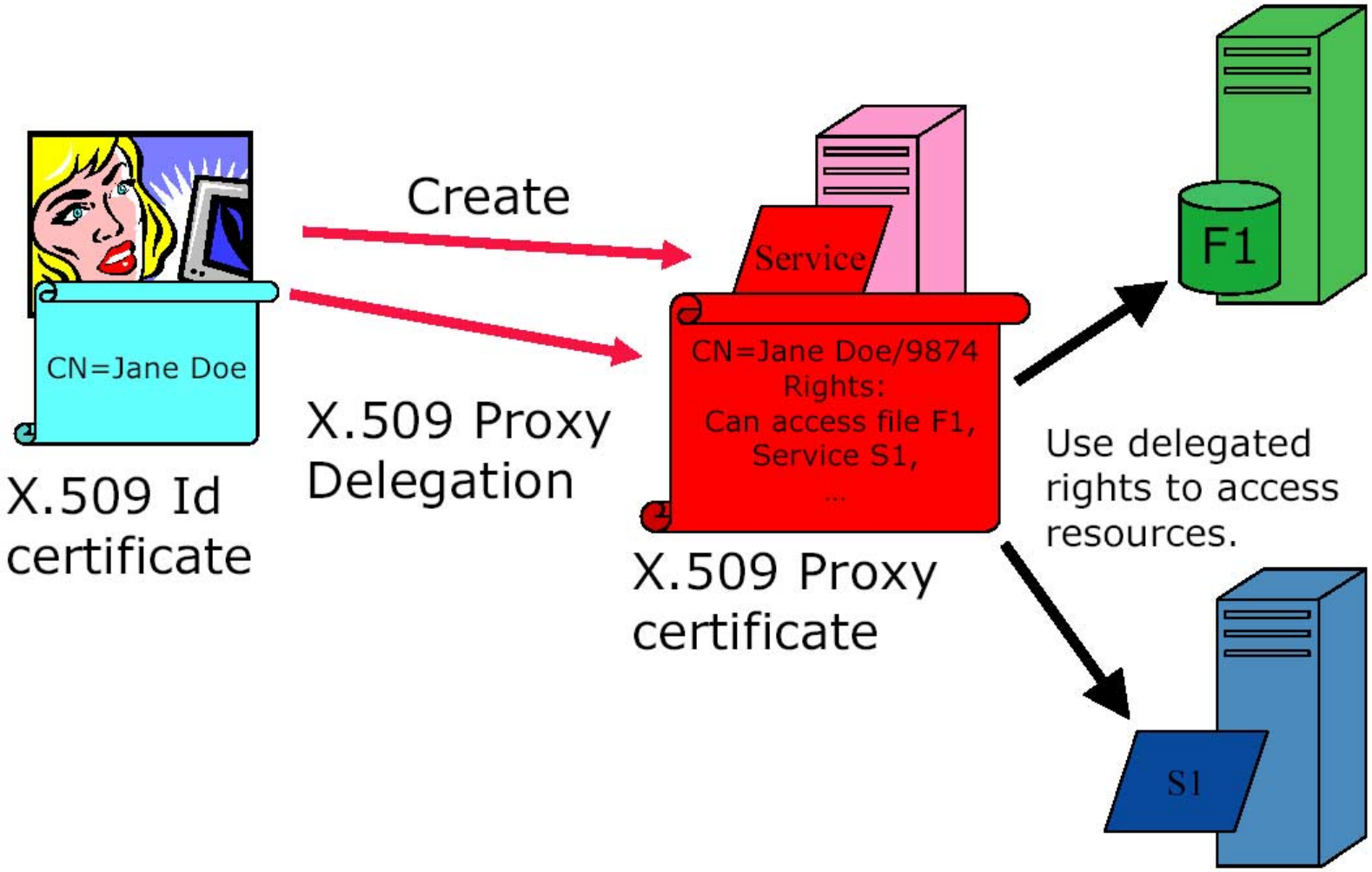
- In current model, all Grid entities assigned a PKI identity.
- User is mapped to local identities to determine local policy.



X.509 Proxy Certificates

- GSI Extension to X.509 Identity Certificates
  - Now public RFC and being implemented in OpenSSL
- Enables single sign-on
- Allow user to dynamically assign identity and rights to service
  - Can name services created on the fly and give them rights (i.e. set policy)
- What is effectively happening is the user is creating their own trust domain of services
  - Services trust each other with user acting as the trust root

Proxy Certificates

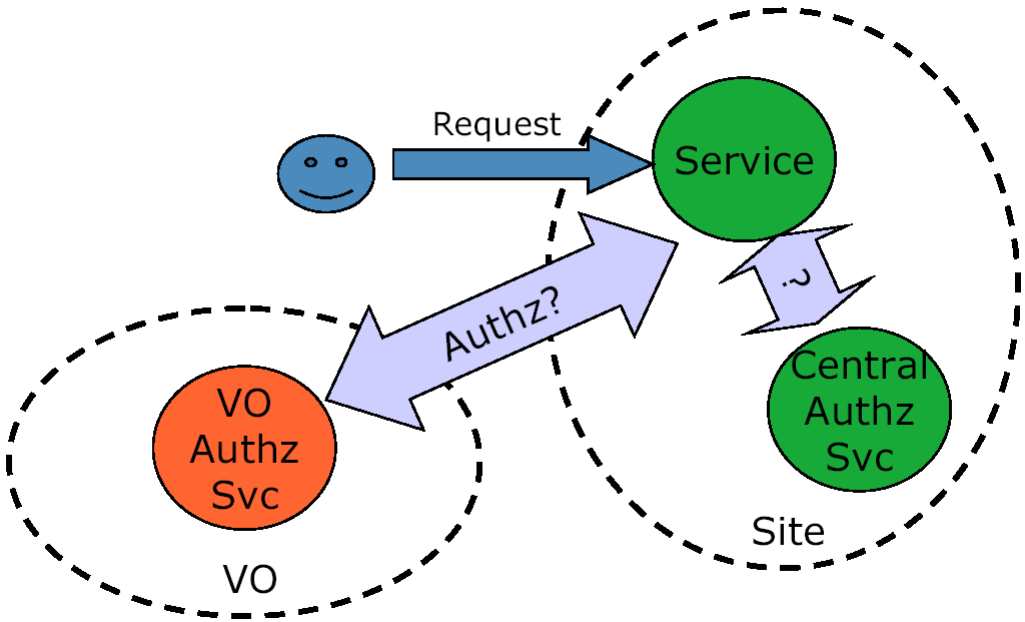


MyProxy: Credential Wallet/Converter

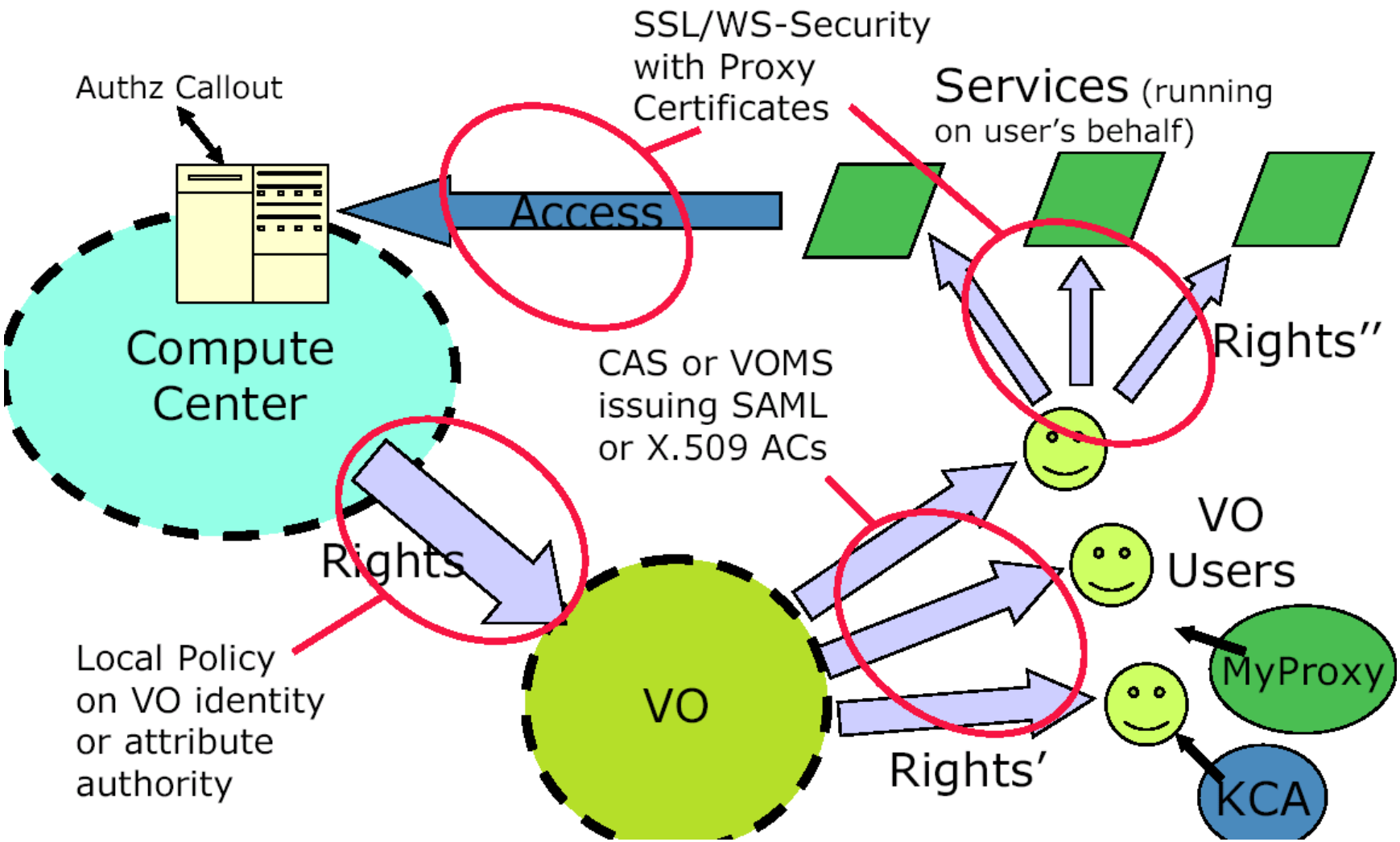
- MyProxy allows users to store GSI credentials and retrieve them
  - With username/pass phrase or other credential
  - Can act as a credential translator from username/pass phrase to GSI
- Used by services that can only handle username and pass phrases to authenticate to Grid
  - Services limited by client implementations
    - E.g. web portals
- Also handle credential renewal for long-running tasks

Beyond Local Identity for Authorization

- Mapping to local identity works ok, but has limitations
  - Scalability, granularity, consistency…
- Requirement for greater flexible



GSI Implementation



Outline

- Security Basics
- What is Grid Security? What makes it different?
- Current Grid Security?
- **OGSA Security and Web Services Security**
- Globus Security

Web Services and OGSA preview

- Web Services
  - Standard XML based messaging over the Internet (W3C SOAP)
  - Standard Implementation-model-agnostic remote service interfaces (W3C WSDL)
  - Architecture where clients find service providers in discovery service (SOA)
- OGSA
  - Open Grid Services Architecture defined by GGF
  - "… defining, within a service-oriented architecture, a set of core capabilities and behaviors that address key concerns in Grid systems."
  - Resource Management Services (GRAM), Information Services (MDS), Security Services (CAS), Self Management Services, Data Services (GridFTP)

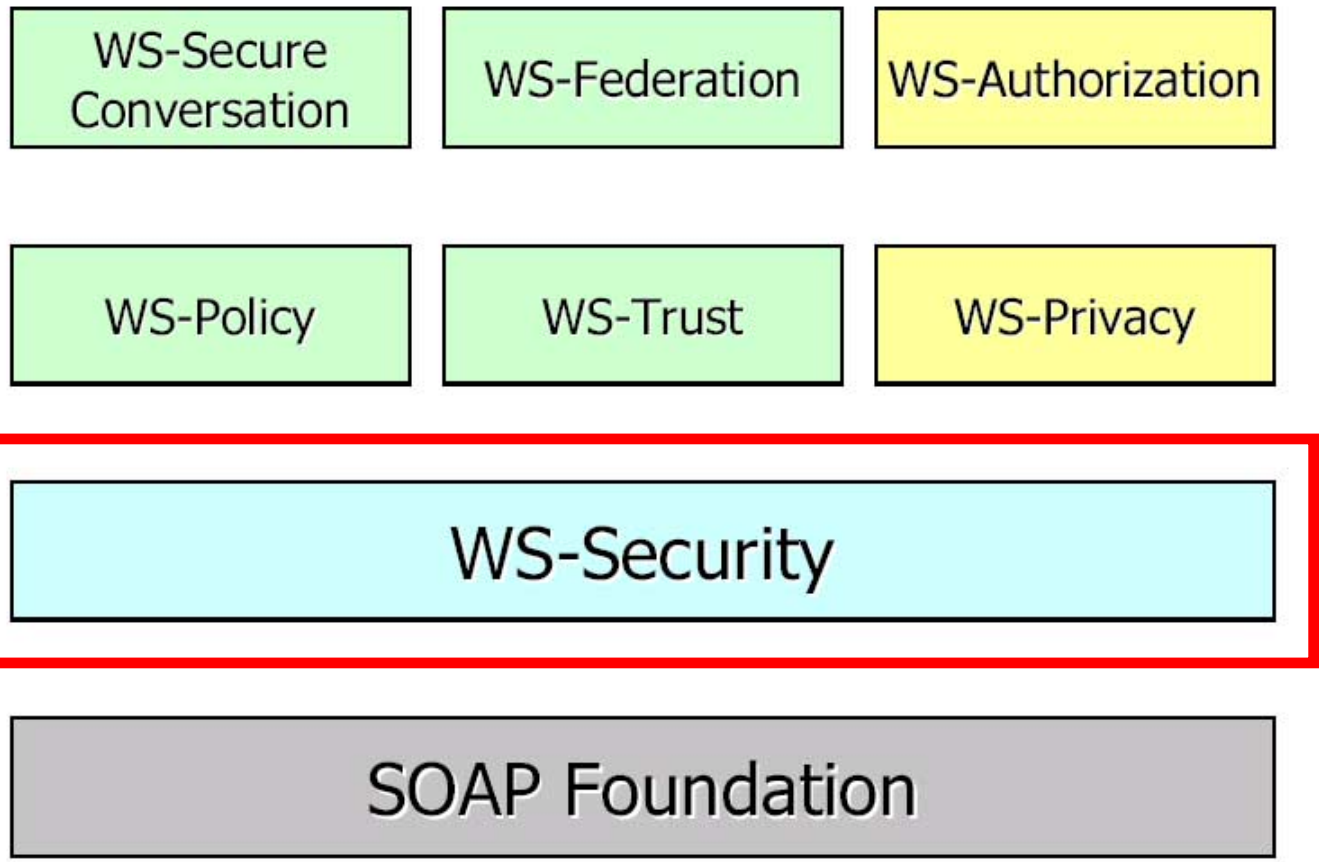
Leveraging Existing/Emerging Security Standards

- WS-Security/Policy/Trust/Federation/Authorization/ SecureConversation/Privacy…
- XKMS (Key Management), XML-Signature/Encryption, SAML (Security Assertions), XACML (Access Control Markup), XrML (Digital Rights Language)

But…

- Need to OGSA'fy
- Need to define Profile/Mechanisms
- Need to address late/missing specs
- Support for delegation, transient services

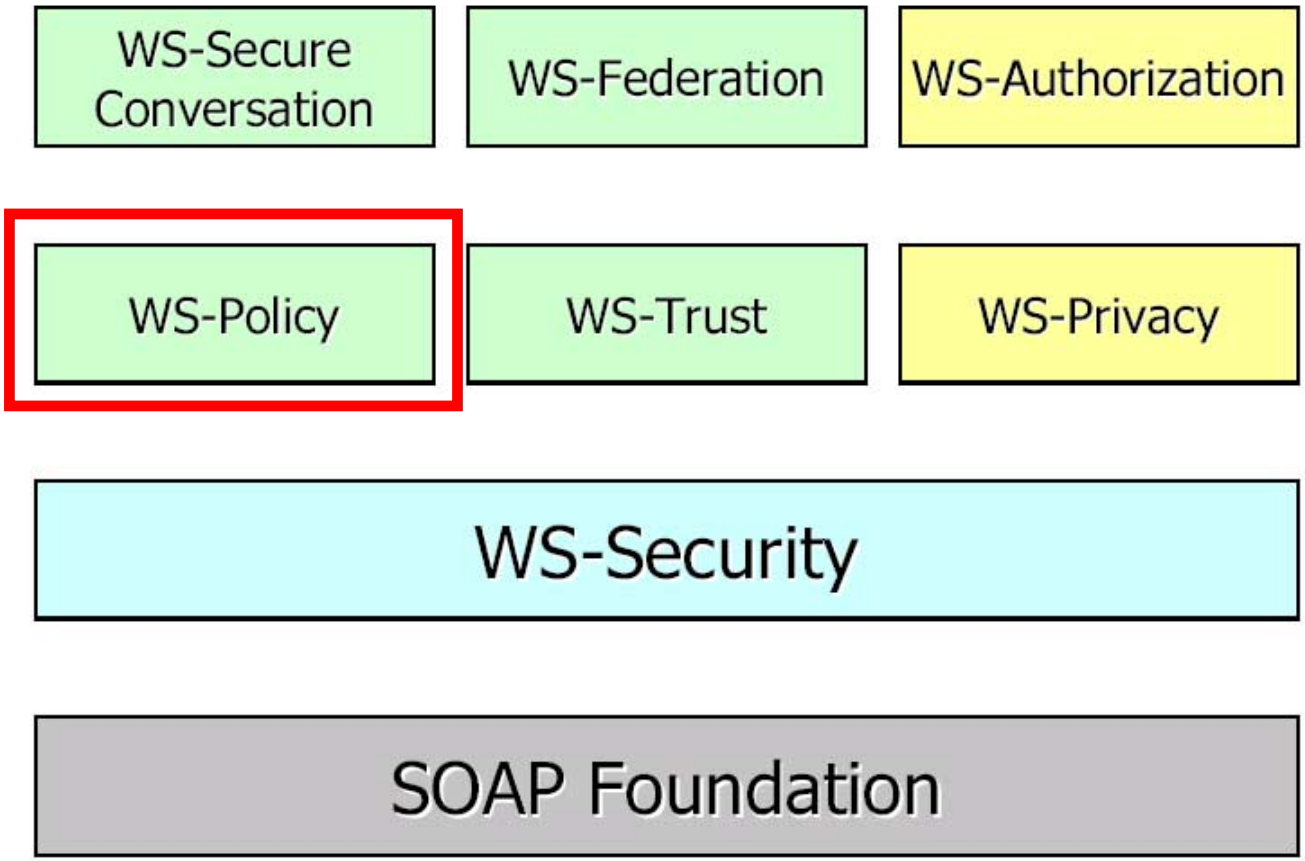
WS-Security Suite: WS-Security



Describes SOAP Extensions for secure messaging, provides foundation for other building blocks

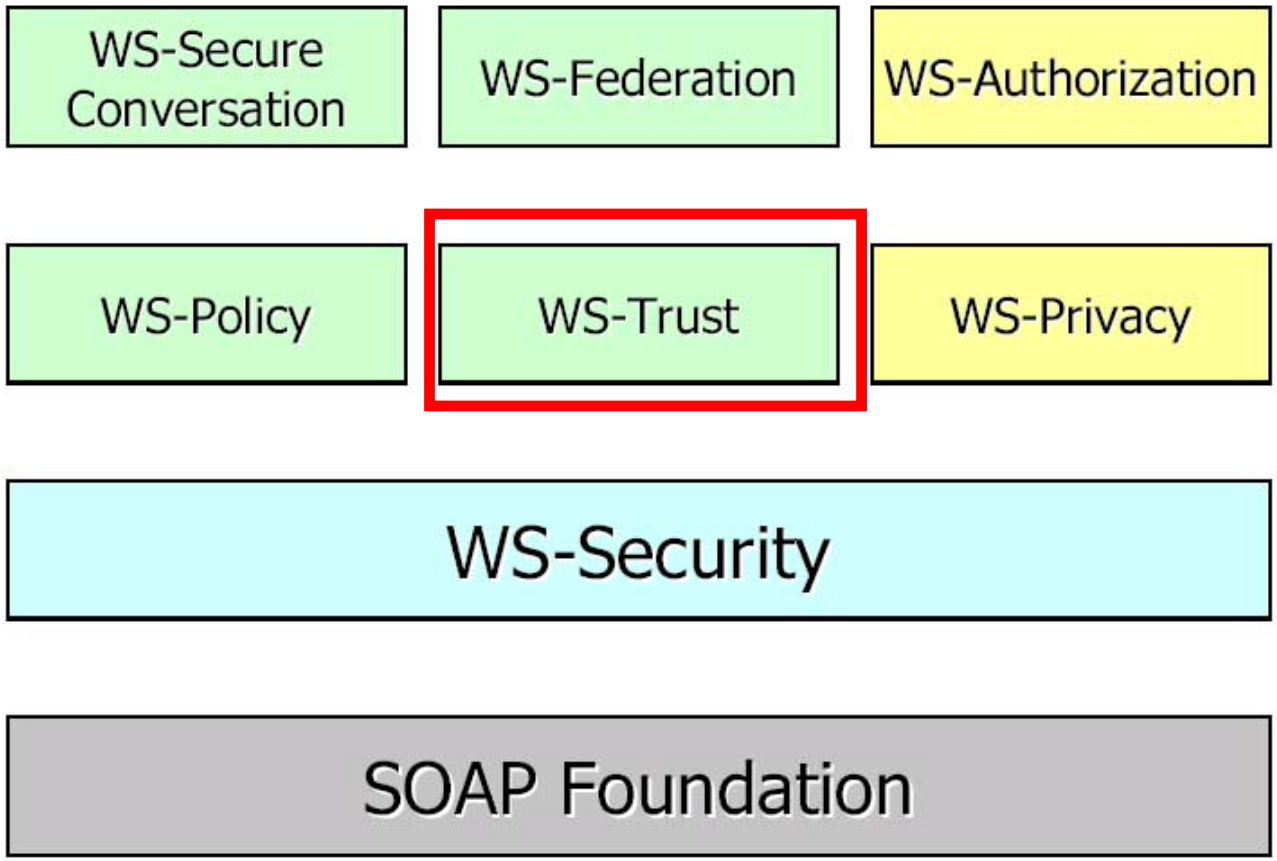
WS-Security Suite: WS-Policy

How to express capabilities and constraints of security policies. Along with WS-SecurityPolicy, WS-PolicyAttachments



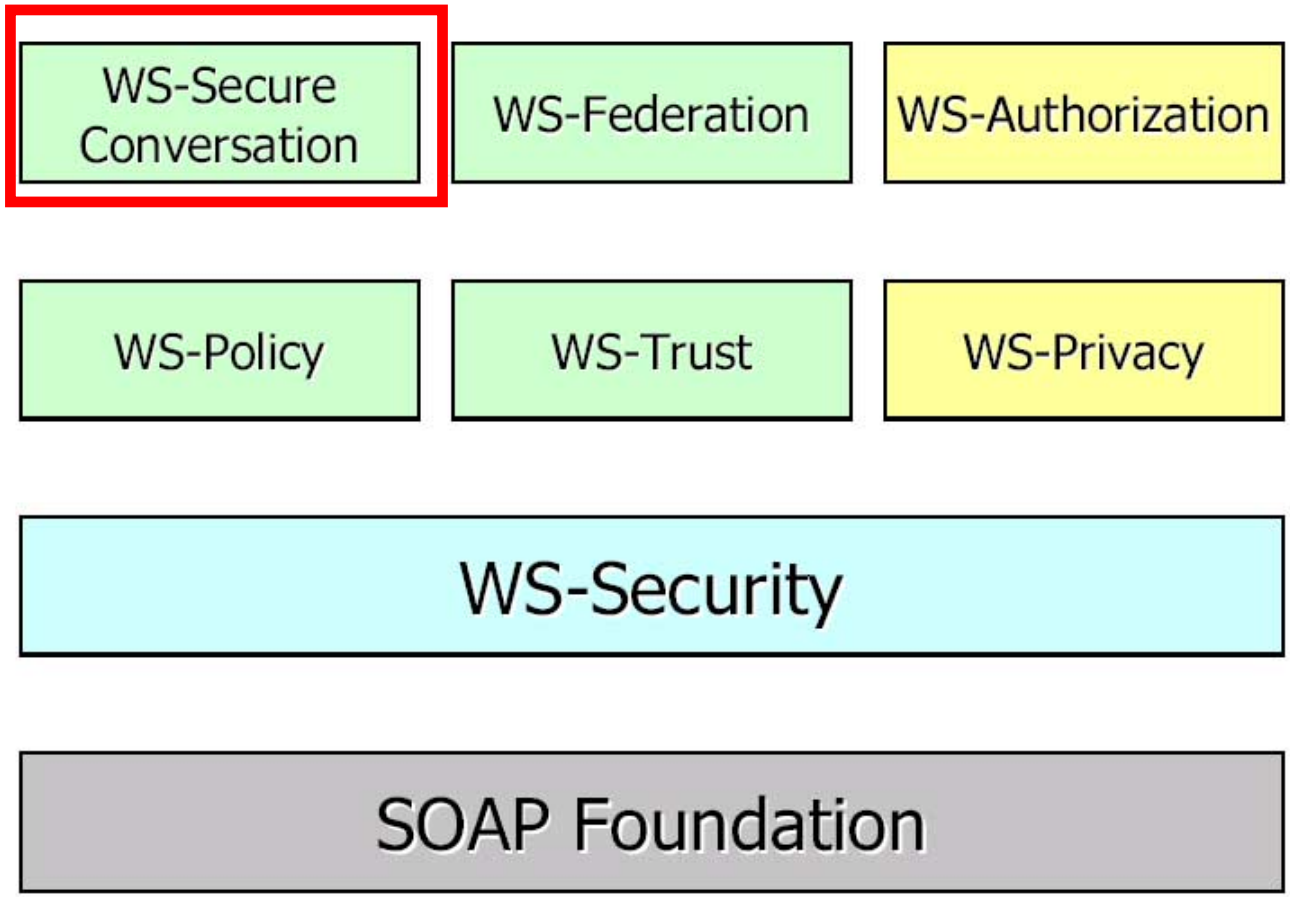
WS-Security Suite: WS-Trust

Describes the model for establishing both direct and brokered trust relationships (including 3rd parties and intermediaries)



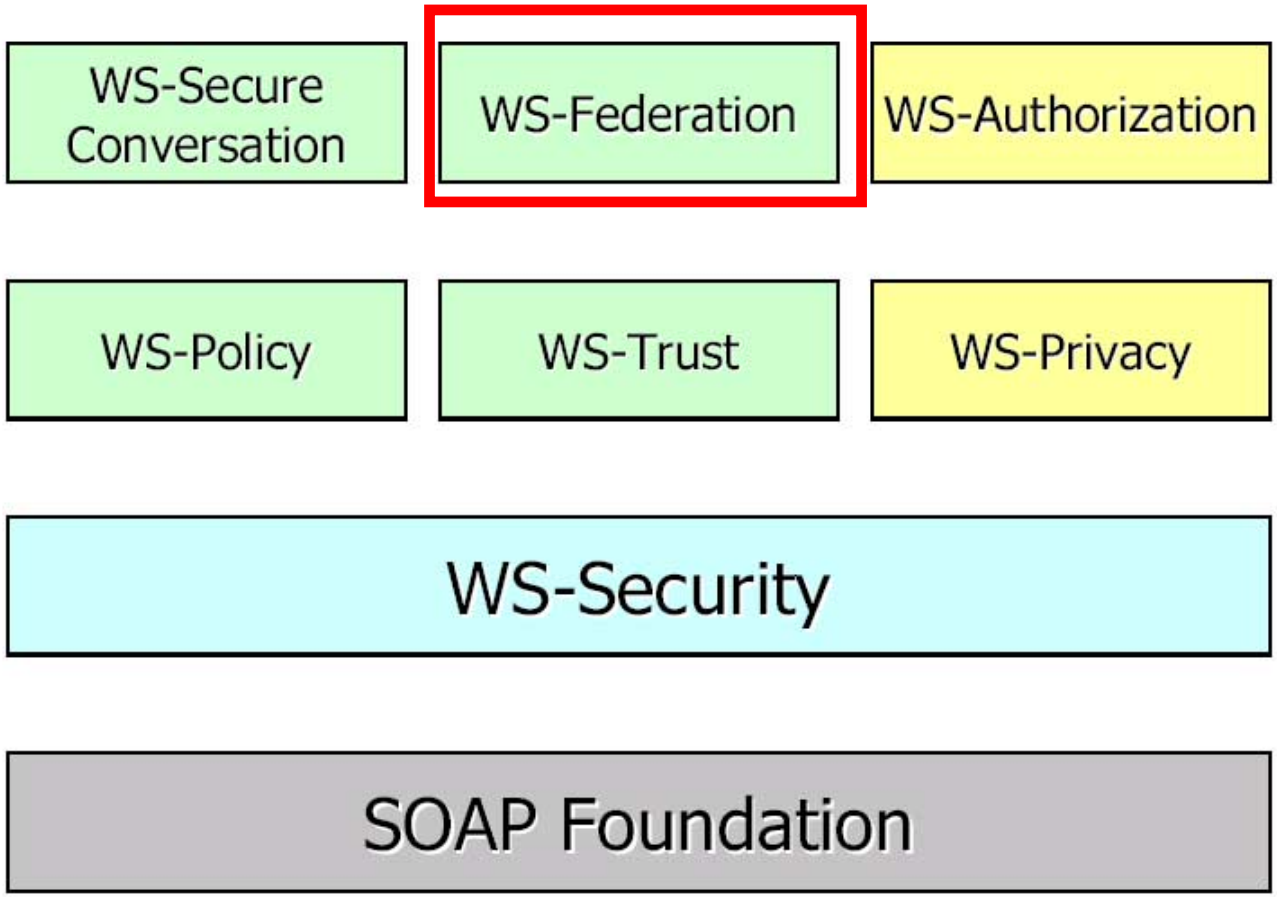
WS-Security Suite: WS-SecureConversation

How to manage and authenticate message exchanges between parties including security context exchange and deriving session keys



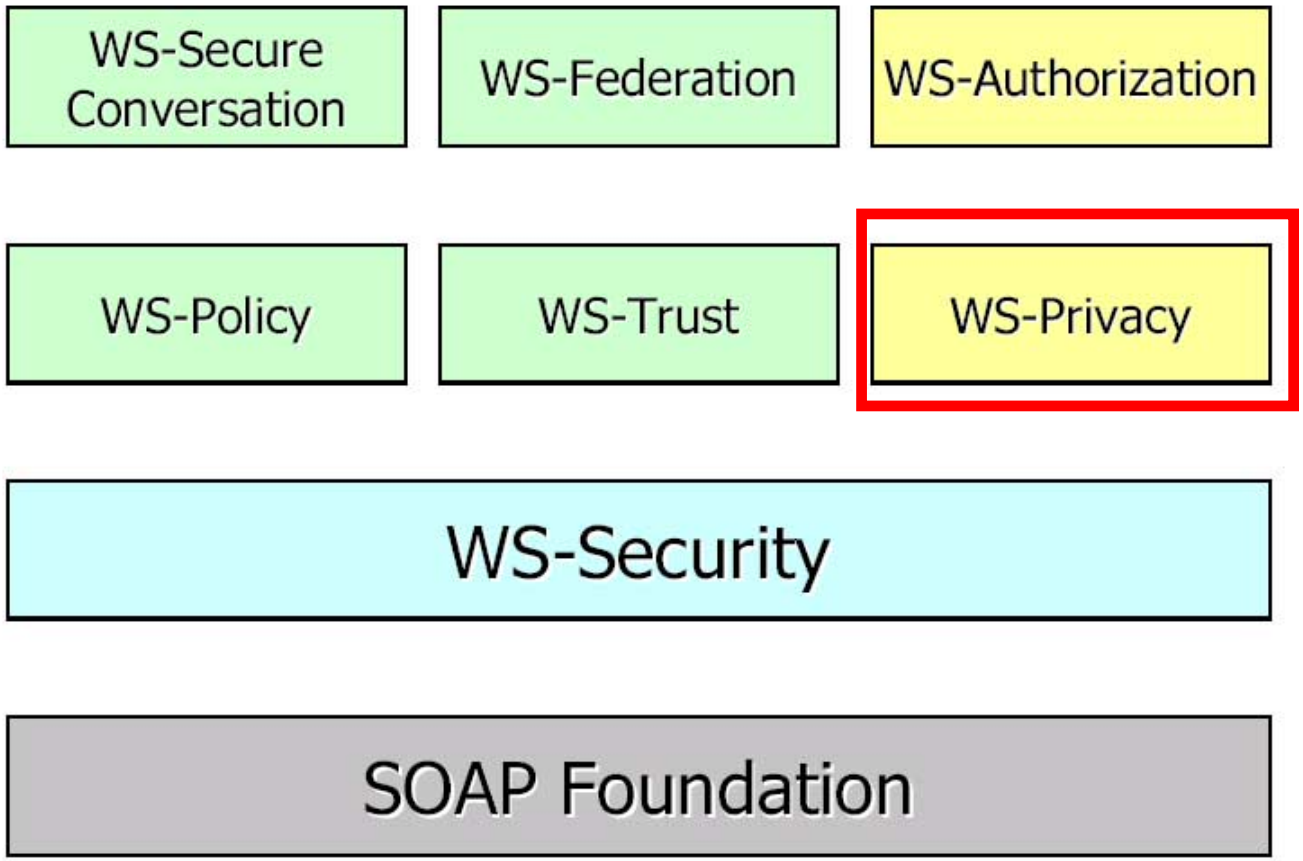
WS-Security Suite: WS-Federation

Describes how to manage and broker the trust relationships in a heterogeneous federated environment including support for federated identities



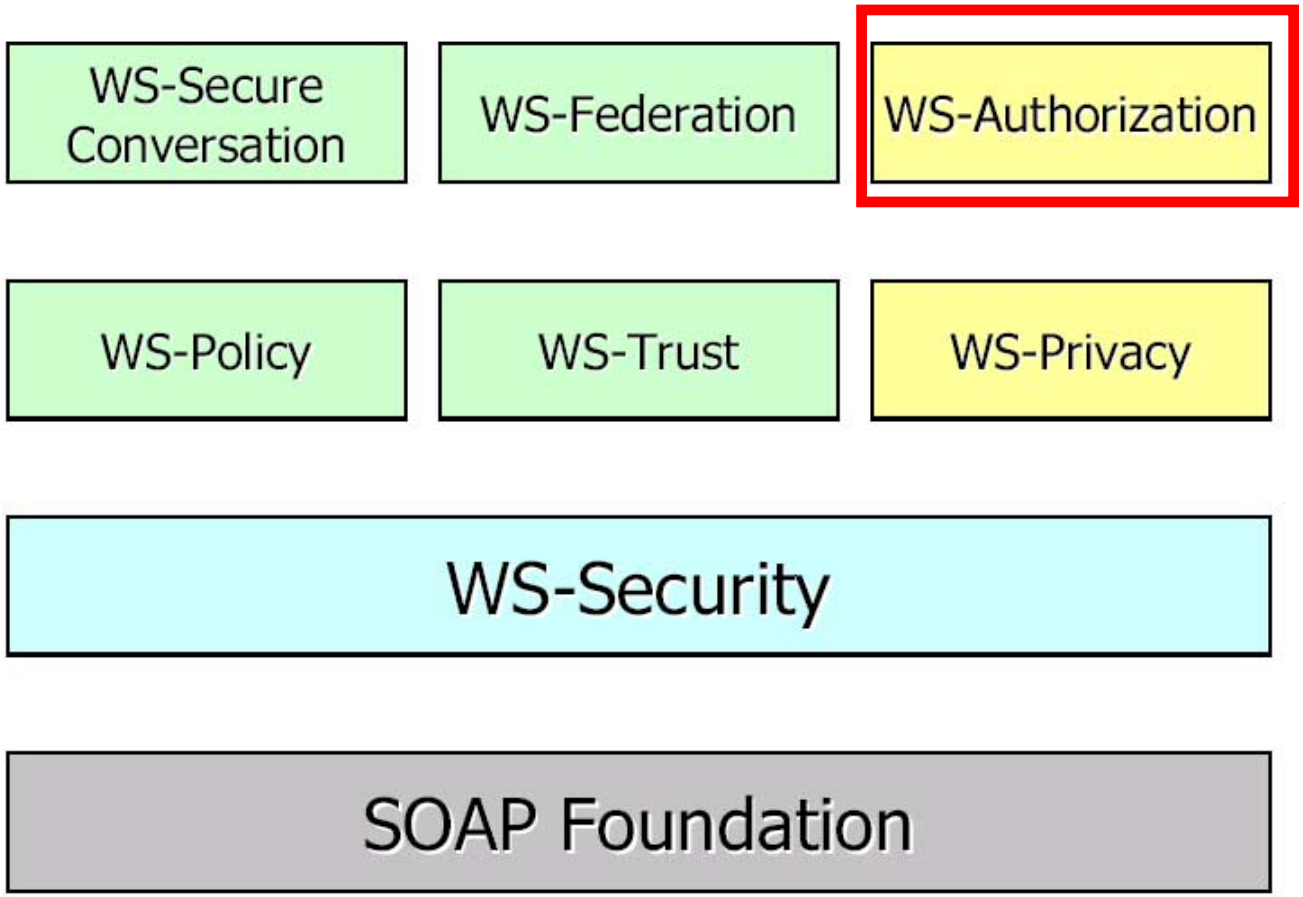
WS-Security Suite: WS-Privacy

Models for how users state privacy preferences, and for how Web services state and implement privacy practices

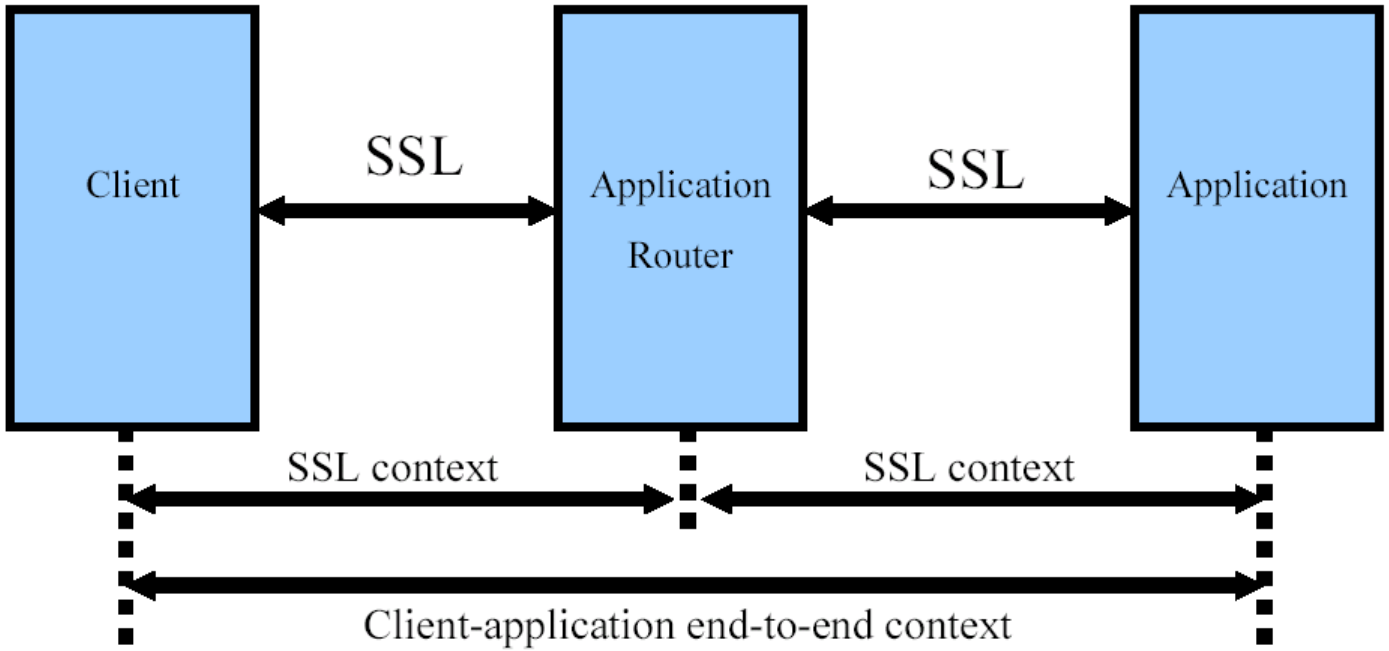


WS-Security Suite: WS-Authorization

Defines how Web services manage authorization data and policies



Transport vs. Message Protection



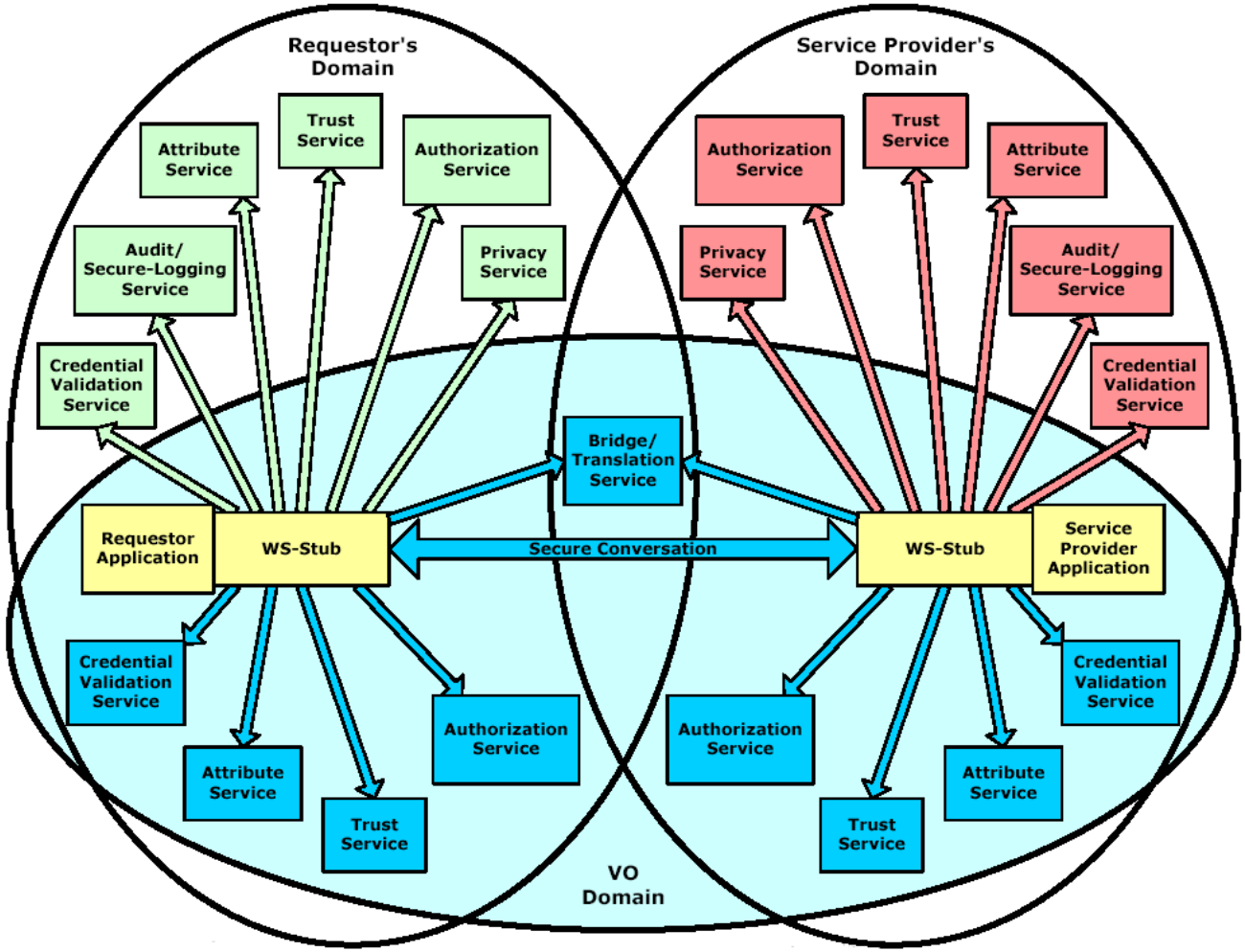
- **SSL Security Context determined by endpoints of socket connection**

  => Application Router becomes part of Trust Chain

- **Message level protection => end-to-end client-app security context**

  ("tunneled" through the routing elements)

OGSA Security Services



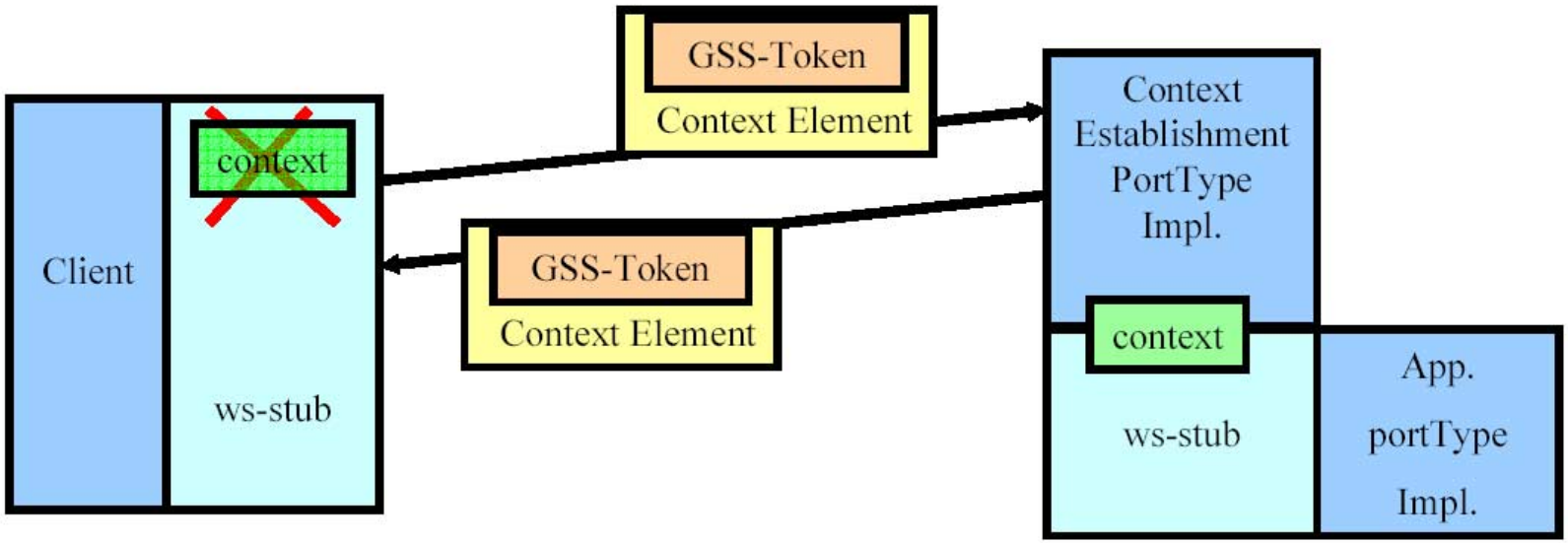
Outline

- Security Basics
- What is Grid Security? What makes it different?
- Current Grid Security?
- OGSA Security and Web Services Security
- **Globus Security**

Interacting with GSI: Globus CLI

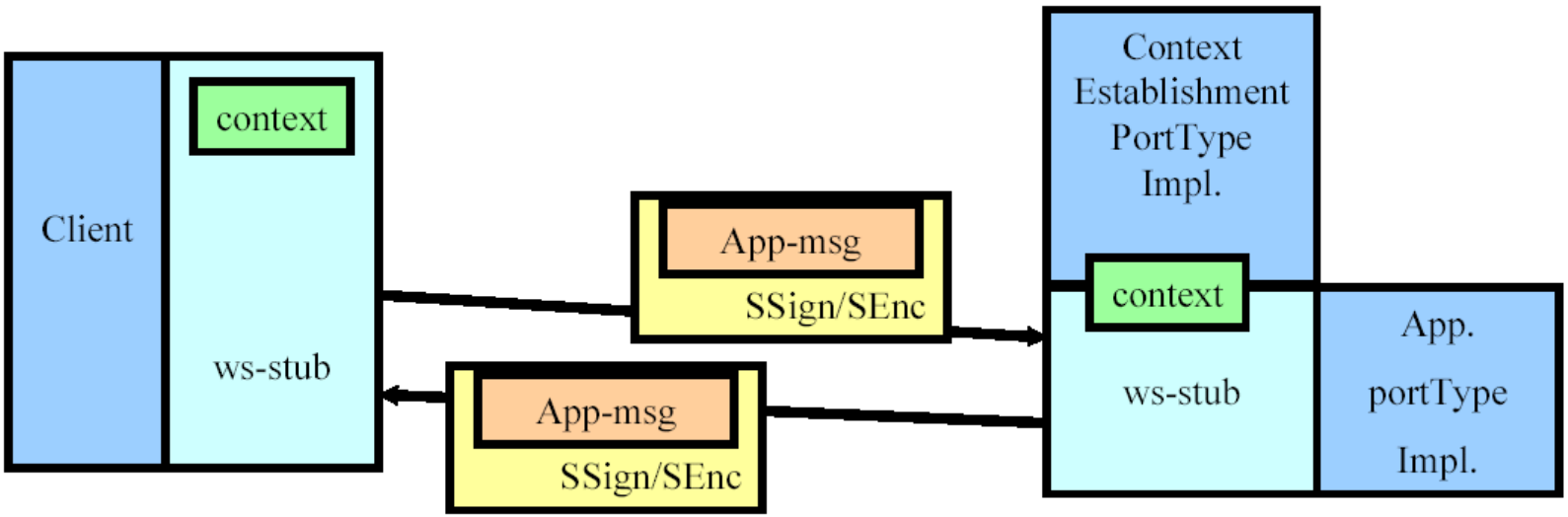
- grid-cert-request –h
  - Issue a certificate request to a CA to get a signed public key certificate to use with the private key generated
- grid-cert-info –h
  - View certificate details such as Subject DN and CA Issuer DN
- grid-proxy-init –h
  - Create a local proxy from a public/private key pair on the local machine as a single sign-on step
  - Allows proxy based delegation (impersonation)
  - Should be treated as a decrypted private key (limited time and should only be available to local user)
  - Default 24h lifetime (customizable)
- grid-proxy-info –h
  - Similar to grid-cert-info, but with proxy specific info

Globus Secure Conversation: Context Establishment



- New security context is established if none exists
- Dedicated context establishment WSDL portType (interface)
- Transparent to client and service application code

Globus Secure Conversation: Message Protection



- Application messages protection through established context
- Integrity and confidentiality protection through shared session key
- Transparent to client and service application code

Authorization Goals

- Build on existing WS standards
  - SAML, XAMCL, WS Security Suite, XrML, etc.
- Support multiple mechanisms
  - But specify set for interoperability
- Remove Authz from application
  - Allow deployer to select
- Enable VO-driven policies
  - Limited delegation

Remove Authorization from Applications

- Allow deployment-time selection of supported mechanisms and policies
- OGSA resource virtualization allows for policy management using application-independent operation invocations
- Place as much security functionality as possible into hosting environments

Community Authorization Service

- Question: How does a large community grant its users access to a large set of resources?
- Community Authorization Service (CAS)
  - Outsource policy admin to VO sub-domain
  - Enables fine-grained policy
- Resource owner sets course-grained policy rules for foreign domain on "CAS-identity"
- CAS sets policy rules for its local users
- Requestors obtain capabilities from their local CAS that get enforced at the resource

Community Authorization Service