Re-thinking

Grid Security Architecture

Yuri Demchenko
SNE Group, University of Amsterdam

Presenting joint work by
Y. Demchenko, C. de Laat, O. Koeroo, D. Groep

E-Science 2008 Conference
10-12 December 2008, Indianapolis, USA

Outline

- Background and motivation
- OGF: Grid definition and Grid types
  - Use case: Grid enabled multidomain Network Resource Provisioning
- OGF Standardisation in Grid Security
- Use case: Pilot Job submission and execution in Computer/Cluster Grids
- Two basic security models (TCB and OSI/Internet)
  - Retrospective overview
- Suggested research problems and ideas in Grid Security

Background and motivation

- Experience and result of more than 5 years working on Grid security
  - Developing, researching, studying, plumbing, fixing
  - Trying to understand all diverse technologies in Grid and Grid security
  - Three major projects DataGrid, EGEE, Phosphorus + cooperation with OSG and Globus
- Open Grid Forum (OGF) standardisation contribution (as time and travel budget allows)
  - Standardisation is the key for Grid interoperability
  - Clear gap between (more advanced) practice and (lagging) standardisation
- There are not many research on Grid security except some particular issues of general security
  - Is this area not attractive or too complex for academia?

Gap between standardisation and practice

- Standardisation is the key for Grid interoperability
- Grid standardisation is coordinated and supported by Open Grid Forum
  - However, some kind of volatility in Grid standardisation
- Current standardisation in Grid is narrowly focused on minimum necessary standards for interoperability
  - OGSA standard defines only Grid Security Services model
  - Standardisation is slow and time consuming
  - GIN (Grid Interoperability Now) WG at OGF made a good contribution to resolving urgent issues in 3 years :-)
- Large Grid project and infrastructures such as EGEE and OSG are solving interoperability based on mutual agreement basis
  - Practical needs and deadlines produce ad-hoc and stop-gap solutions
  - Very limited possibilities to do standardisation and research work

Evolution of the Grid definition

"Anatomy of the Grid" actually described the goal of this new technology at that time:

- "Grid systems and applications aim to integrate, virtualise, and manage resources and services within distributed, heterogeneous, dynamic "virtual organizations"

Open Grid Services Architecture v1.5 (OGSA) (GFD.80, 2006)

- "A system that is concerned with the integration, virtualization, and management of services and resources in a distributed, heterogeneous environment that supports collections of users and resources (virtual organizations) across traditional administrative and organizational domains (real organizations)"

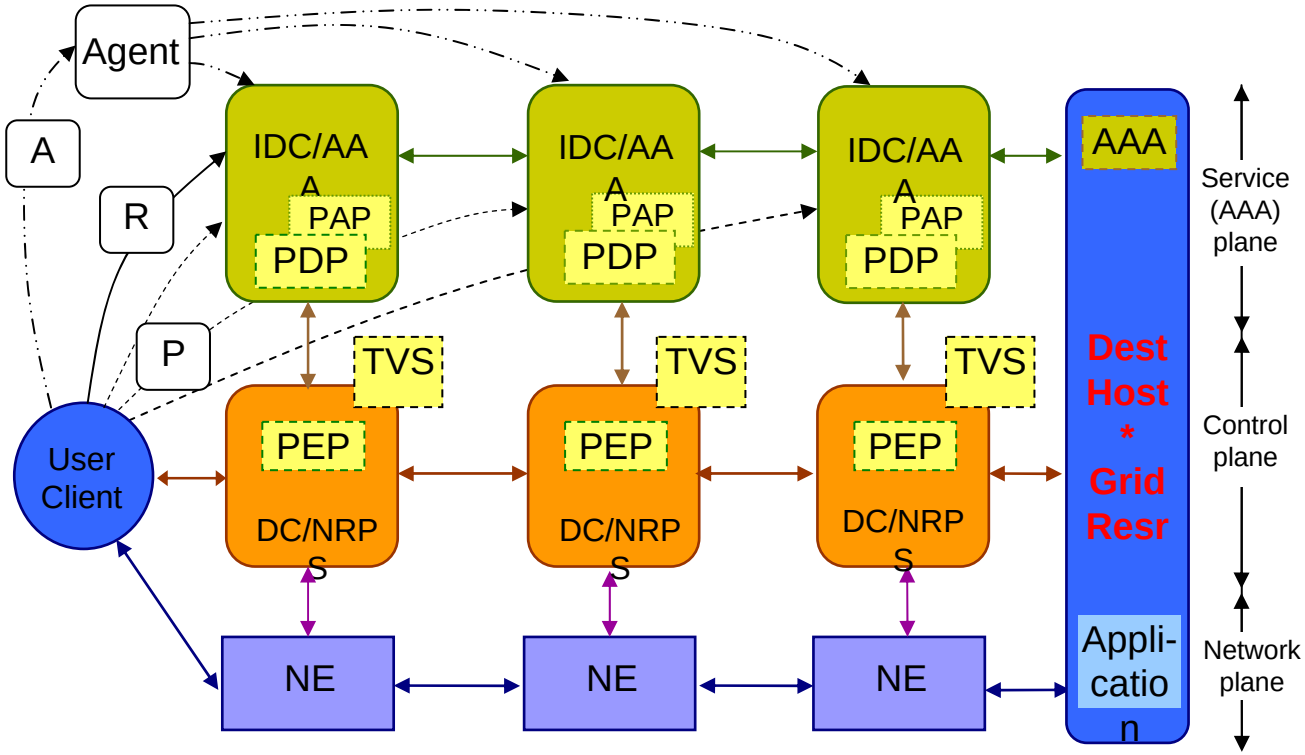
GFD-I.113 – Technical strategy for OGF 2007-2010, the Grid definition is extended

- "Scalable, distributed computing across multiple heterogeneous platforms, locations, organisations"
- Characteristics and goals of Grids in general defined as
  - Dynamic resource provisioning
  - Resource pooling and sharing
  - Management of Virtualised Infrastructure
  - Self-monitoring and improvement
  - Highest quality of service

Grid types (GFD-I.113) and Security issues

- **Cluster Grids** – that have predominantly homogeneous structure and focused on shared use of high performance computing resources.
  - Problem: Bridging open WS/Grid environment and protected UNIX executing environment
- **Collaboration Grids** – that are targeted at supporting collaborative distributed group of people over multiple domains and involving heterogeneous resource
  - Problem: Dynamic task/project oriented inter-domain/multi-domain users and resources association and security context management
- **Data Center Grids** – are actually adding provider specific aspects in managing resources, users, their associations and supporting whole provisioning life-cycle
  - Customer centric security for dynamically provisioned resources
  - Securing remote (to user) virtualised workspace service environment

Multidomain Network Resource Provisioning in/for Grid



- Common Grid/Netw provisioning model
- Interdomain AuthN/AuthZ
- Dynamic security association for provisioned resources

Provisioning sequences

- Agent (A)
- Polling (P)
- Relay (R)

NRPS – Network Resource Provisioning System

DC – Domain Controller

IDC – Interdomain Controller

AAA – AuthN, AuthZ, Accounting Server
PDP – Policy Decision Point
PEP – Policy Enforcement Point
TVS – Token Validation Service
KGS – Key Generation Service

Practical Grid Security – Authentication, Delegation and Trust Management

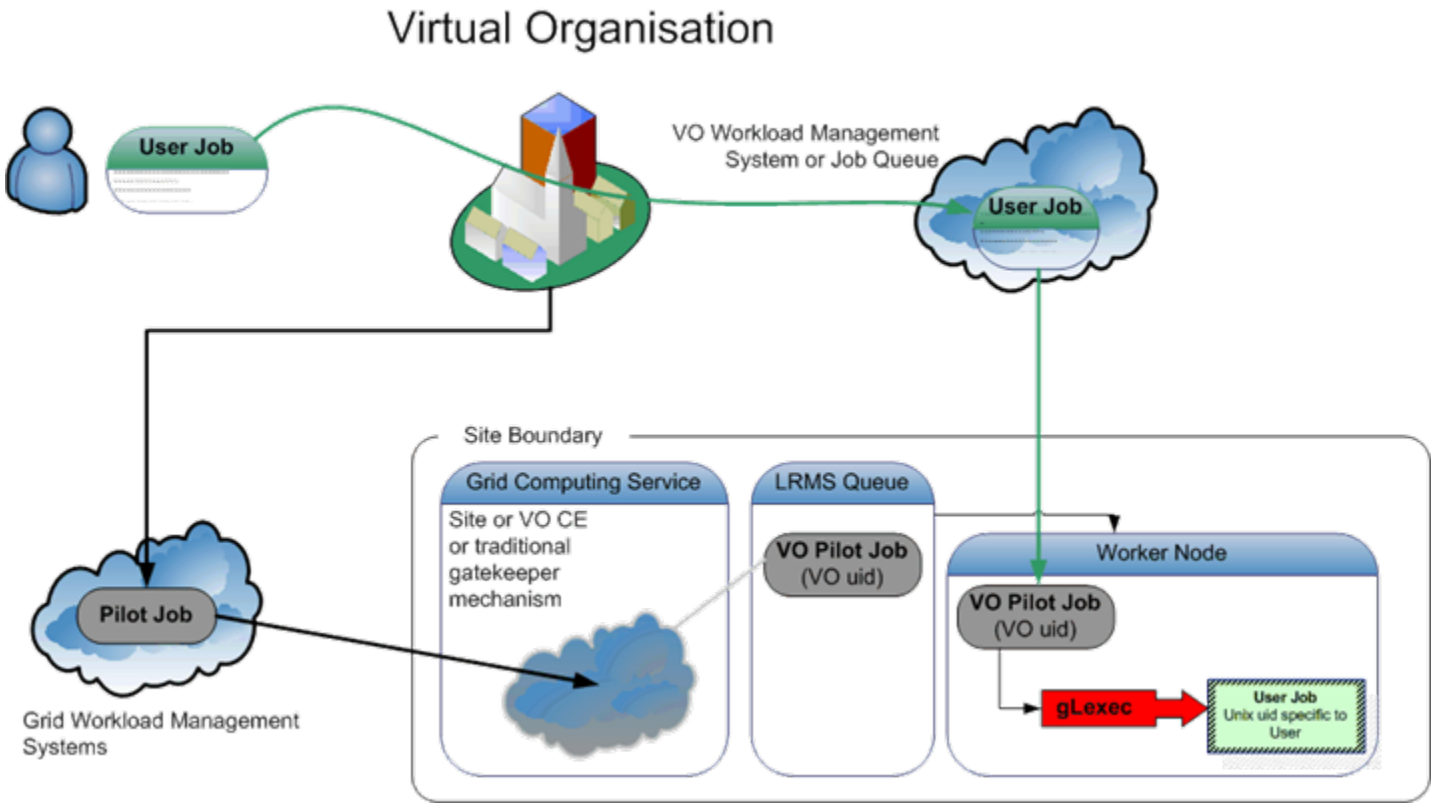
- Grid is for sharing computing resources and other resources in the distributed heterogeneous environment by means of resource and user virtualisation
  - Grid Security is built around Virtual Organisations (VO)
  - Using Web Services Security as a base
- Authentication in the Grid is based on PKI and can use different (user) credentials (PKI, SAML, Kerberos tickets, password, etc.)
- Authorisation is based on VO attributes
  - Simple AuthZ session management by using Proxy or Short Lived Creds (CLC) together with CRL
- Delegation (restricted and full)
  - Job submission in Grid environment requires (credentials) delegation
  - Implemented using X.509 Proxy Certificate (Proxy or PC)
  - Proxy is generated by the user client based on user master PKC or Proxy
  - Limited delegation chain (typically not more than 10)
- Trust is an important component of PKI based AuthN and Delegation
  - Trust relations are represented by a certificate chain
  - Typical Proxy Certs chain

    PKC (DN1, CA) => PC (DN2, (ACa) , PKC) => PPC (DN2, (ACb) , PC) => …
  - International Grid Trust Federation GridPMA – http://www.gridpma.org/

Standardisation in Grid Security by OGF (AuthN/AuthZ)

- OGSA Security Profile 2.0 (GFD.138), also known as “OGSA Express Authentication Profile”
  - Enable discovery of common security mechanisms and define extendibility points to accommodate new mechanisms, credentials, etc.
- Secure Communication Profile 1.0 (GFD.132)
  - Extends WS-I Basic Security Profile and WS-SecurityPolicy 1.2
  - Provides a framework for writing instant policies for WS/SOAP Message Level Security (MLS)
- Secure Addressing Profile 1.0 (GFD.131)
  - Format for secure End-Point Reference (EPR)
- Group of standards produced by OGSA Authz WG
  - AuthZ service components and basic operational models
  - Protocols and messaging WS-Trust, SAML-XACML
  - VOMS Attribute Certificate and SAML profiles
- Ongoing/started work
  - Define requirements to AuthN service, including Levels of Assurance (LOA) - Levels of Authentication Assurance Research Group (LOA-RG)

Pilot Jobs and gLExec on Worker Node (WN)

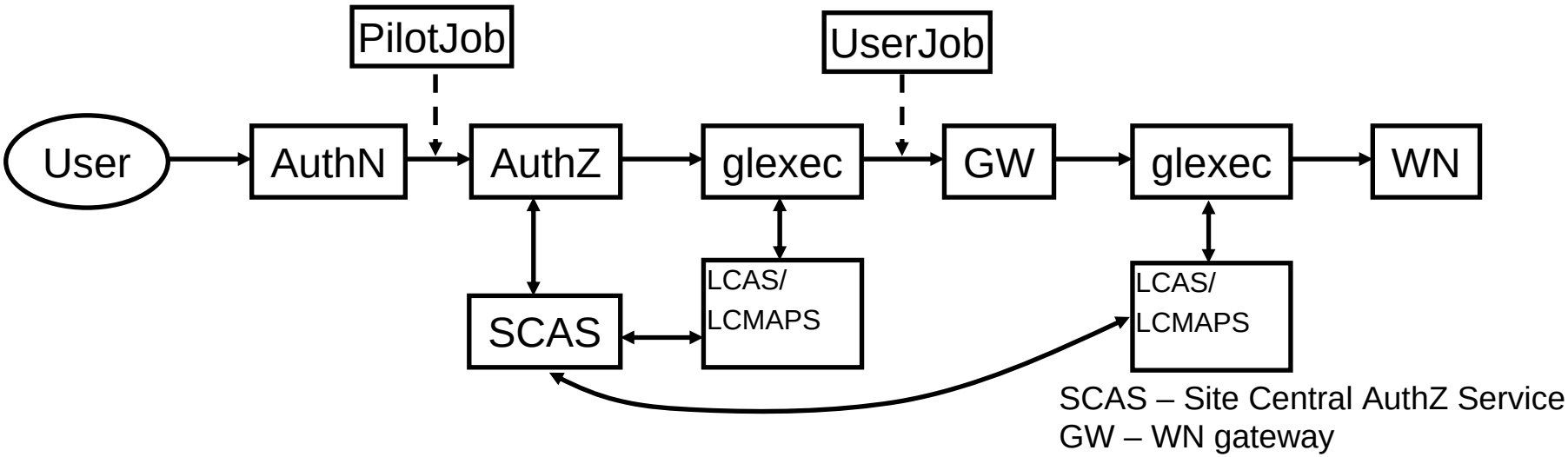


On success: the site will set the uid/gid to the new user's job

On failure gLExec will return with an error, and pilot job can terminate or obtain other user's job

Source: presentation by David Group at EGEE AH meeting – 20-22 Feb 2008

Pilot Jobs and gLExec on WN - Abstraction



Make pilot job subject to normal site policies for jobs

VO submits a pilot job to the batch system

- the VO 'pilot job' submitter is responsible for the pilot behavior
  - ***this might be a specific role in the VO, or a locally registered 'special' user at each site***
- Pilot job obtains the true user job, and presents the user credentials and the job (executable name) to the site (gLExec) to request a decision on a cooperative basis

Preventing 'back-manipulation' of the pilot job

- make sure user workload cannot manipulate the pilot
- project sensitive data in the pilot environment (e.g. not revealing job and user ID)
- Fare resource sharing if multiple user jobs

Pilot job scenario implemented using policy obligations

OSI-Security vs TCB Security – Multi-layer vs Multi-level security

Open Systems and Internet

Open Systems Interconnection (OSI) Security Architecture

- **ISO7498-2/X.800 – Multilayer Security**

Independently managed interconnected system

Trust established mutually or via 3rd party

PKI and PKI based AuthN and key exchange

Concept of the Security Context

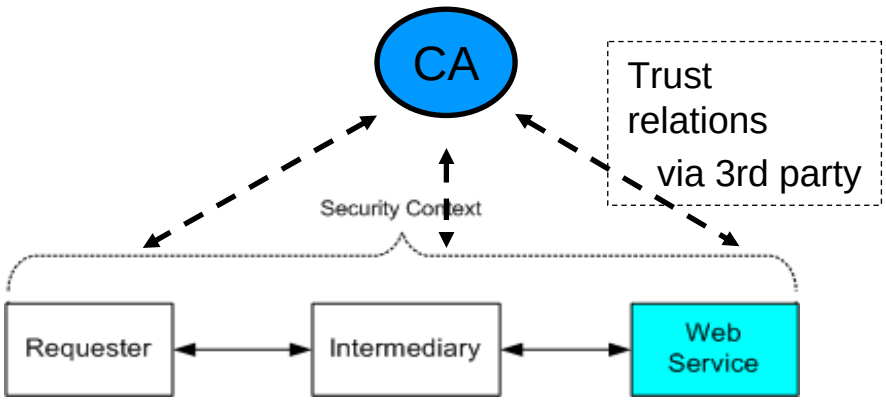
Trusted Computing Base (TCB)

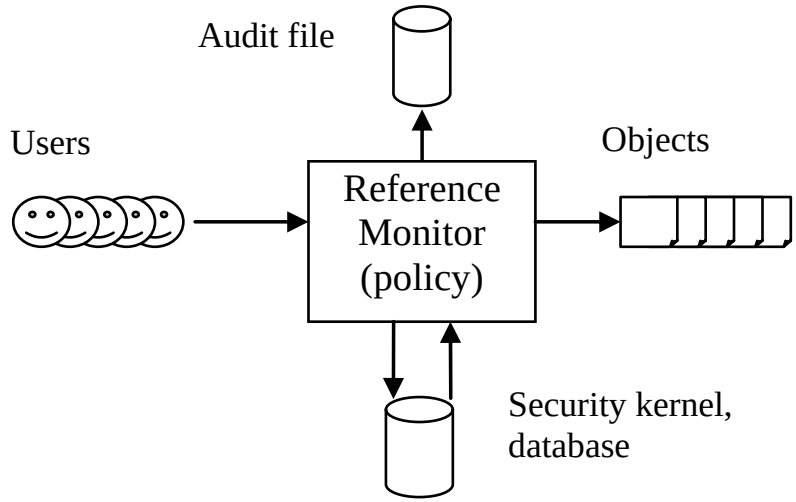
Uses Reference Monitor (RM) concept proposed by J.P.Anderson in 1972

RM property provides a basis for **Multi-Level Security (MLS)**

- Complete mediation
- Isolation
- Verifiability

MLS models – Bell-LaPadula, Biba, Clark-Wilson

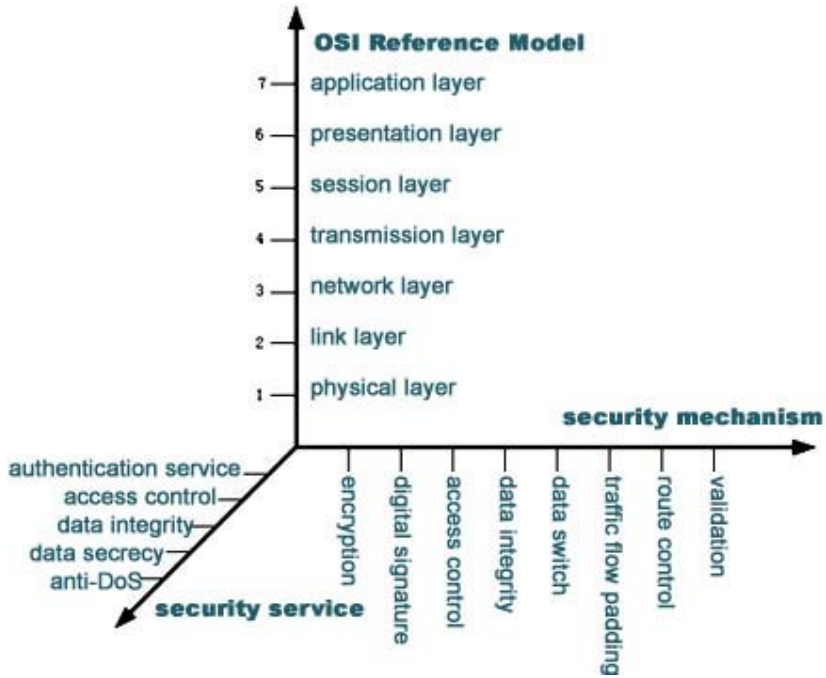




X.800/OSI Security – Layers vs Services vs Mechanisms

Service \ Mechanism ->	Encipherment	Digital signature	Access control	Data integrity	Authentication exchange	Traffic padding	Routing control	Notarization
Authentication, Peer entity	Y	Y			Y			
Authentication, Data origin	Y	Y						
Access control service	Y		Y					
Connection confidentiality	Y						Y	
Connectionless confidentiality	Y						Y	
Selective field confidentiality	Y							
Traffic flow confidentiality	Y					Y	Y	
Connection Integrity with recovery	Y			Y				
Connection integrity without recovery	Y			Y				
Selective field connection integrity	Y			Y				
Connectionless integrity	Y	Y		Y				
Selective field connectionless integrity	Y	Y		Y				
Non-repudiation. Origin		Y		Y				Y
Non-repudiation. Delivery		Y		Y				Y

Service	Layer						
	1	2	3	4	5	6	7*
Peer entity authentication			Y	Y			Y
Data origin authentication			Y	Y			Y
Access control service			Y	Y			Y
Connection confidentiality	Y	Y	Y	Y		Y	Y
Connectionless confidentiality		Y	Y	Y		Y	Y
Selective field confidentiality						Y	Y
Traffic flow confidentiality	Y		Y				Y
Connection Integrity with recovery				Y			Y
Connection integrity without recovery			Y	Y			Y
Selective field connection integrity							Y
Connectionless integrity			Y	Y			Y
Selective field connectionless integrity							Y
Non-repudiation Origin							Y
Non-repudiation. Delivery							Y



Similar model should be probably proposed for the WS SOAP based security services and mechanisms

Upper layers (above application) can be defined for WS/SOA messaging

From OSI/Internet to SOA/WSA Security Model

X.800 Security Architecture for Open Systems Interconnection for CCITT applications. ITU-T (CCITT) Recommendation, 1991

- ISO 7498-2:1989 Information processing systems -- Open Systems Interconnection -- Basic Reference Model -- Part 2: Security Architecture

Web Services Security Roadmap (2002)

- http://www.ibm.com/developerworks/library/specification/ws-secmap/

OGSA Security Model Components (2002-2006)

- GFD.80 - OGSA version 1.5, Section 3.7 Security Services
- Re-states Web Services Security roadmap

WS-Security stds specify using SOAP header for security related issues

- Considered as orthogonal to major service

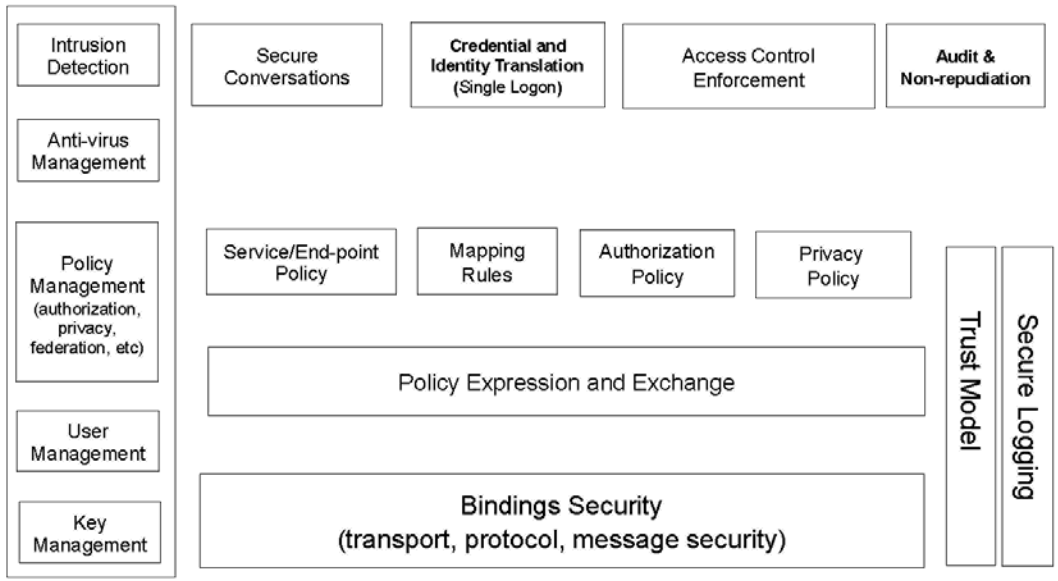


Figure 2: Components of Grid Security Model

Developing Consistent Grid Security Architecture – Suggested Issues for Research

- Complex Resource Provisioning (CRP) model
- AuthZ and Security session and inter-domain security context management
- Policy Obligations – bridging two fundamental security models
  - Re-factoring policy-based access control to policy-based object management
- Define/extend security zones model
  - Including AuthN, (Delegation,) AuthZ, (AuthZ Session,) gLExec/Unix
- Extending User Controlled Security Domain in Virtualised Workspace Service (VWSS) with TCPA
  - Leverage Trusted Computing Platform Architecture (TCPA) for basic Grid use cases, in particular for securing remote virtual execution environment
- Identity Based Cryptography for interdomain trust management and “trusted introduction”

Complex Resource Provisioning (CRP) Model

Two use case of the general Complex Resource Provisioning (CRP)

- On-demand Network Resource Provisioning
- Grid Computing Resource – Distributed and heterogeneous

3 major stages/phases in CRP operation/workflow

- Reservation consisting of 3 basic steps
  - Resource Lookup
  - Resource composition (including options)
  - Component resources reservation (in advance), including combined AuthZ/policy decision, and assigning a global reservation ID (GRI)
- Deployment – reservation confirmation and distributing components/domain configuration (including trusted keys)
- Access (to the reserved resource) or consumption (of the consumable resource)

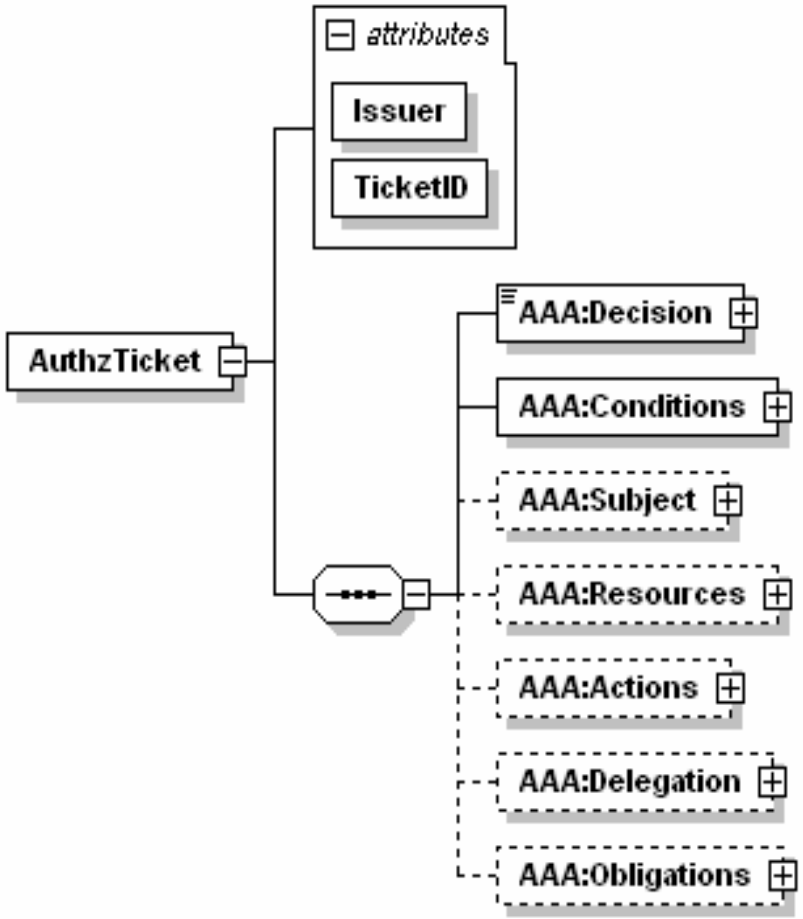
Now considering two other stages: “decommissioning” and “relocation”

- Topic for future research and discussions
- Will allows integrating resource provisioning into the upper layer scientific workflow in more consistent way

AuthZ session and Inter-domain security context management

- Type of context and required functionality
  - Obligations
  - Identity mapping
  - Delegation
  - Evidence/context/provenance
  - Support inter-domain advance reservation
    - With the new proposed pilot token concept (primary use for NRP)
- Current Grid security model/middleware uses Proxy certificate for implicit session management
  - Contains embedded VOMS AC and has limited validity time
- Can be implemented with such mechanisms as
  - Proprietary AuthZ ticket and/or token
  - SAML credentials

Proprietary AuthZ ticket format for extended security context management – Main elements

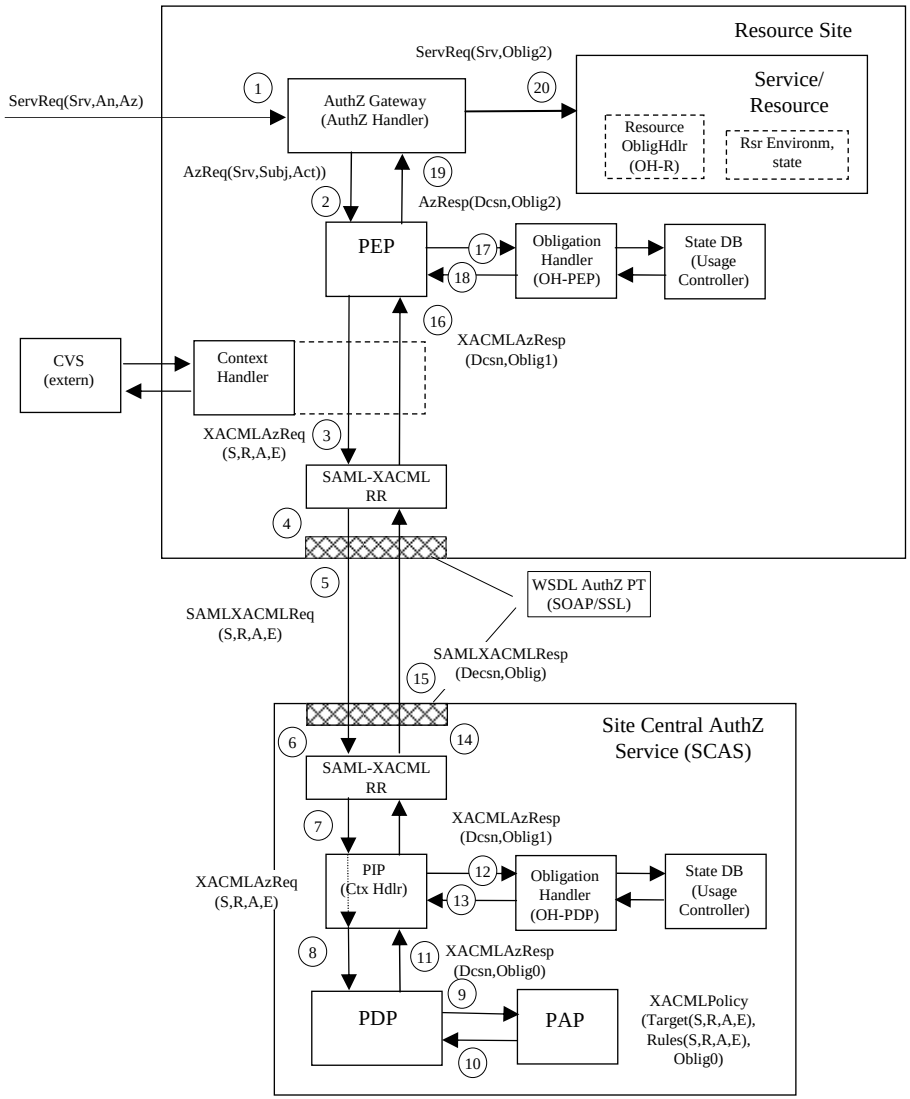


- Required functionality to support multidomain provisioning scenarios
  - Allows easy mapping to SAML and XACML related elements
- Allows multiple Attributes format (semantics, namespaces)
- Establish and maintain Trust relations between domains
  - Including Delegation
- Ensure Integrity of the AuthZ decision
  - Keeps AuthN/AuthZ context
  - Allow Obligated Decisions (e.g. XACML)
- Confidentiality
  - Creates a basis for user-controlled Secure session

Policy Obligations in Grid – Bridging two security concepts

- Policy obligations and access control in Grid
  - Account mapping, quota assignment, usable resource
  - Environment setup/configuration
- Policy Obligation is one of the policy enforcement mechanisms
- Obligations enforcement scenarios
  - Obligations are enforced by PEP at the time of receiving obligated AuthZ decision from PDP
  - Obligations are enforced at later time when the requestor accesses the resource or service
    - Require use of AuthZ assertions/tickets/(restricted proxy?)
  - Obligations are enforced before or after the resource or service accessed/delivered/consumed
- XACML policy obligations
  - ***Obligations*** are a set of operations that must be performed by the ***PEP*** in conjunction with an ***authorization decision*** [XACML2.0]
  - PEPs that conform with XACMLv2.0 are required to deny access unless they understand and can discharge all obligations returned by PDP
- Obligations in policy based management – applied to managed object
  - Obligated policy decision & Provisional policy decision
  - Ponder policy language and framework

Proposed Obligations Handling Reference Model



Generic AuthZ service model

Obligation0 = tObligation

=> Obligation1 (“OK?”, (Attributes1 v Environments1))

=> Obligation2 (“OK?”, (Attributes2 v Environments2))

=> Obligation3 (Attributes3 v Environments3)

Note. Obligation1 handling at the SCAS or PDP side allows stateful PDP/SCAS.

PEP – Policy Enforcement Point

PDP – Policy Decision Point

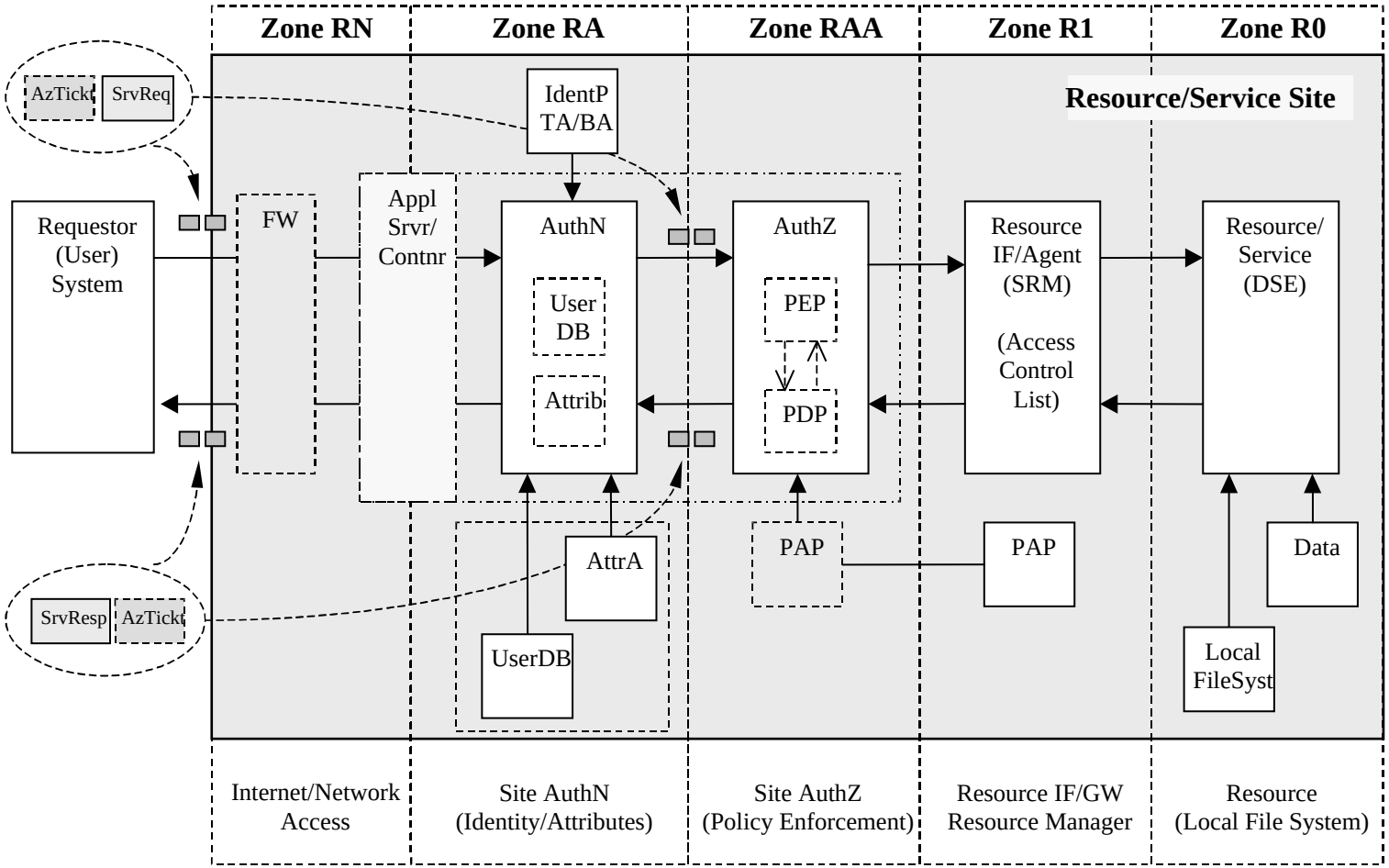
PAP – Policy Authority Point

OH – Obligation Handler

CtxHandler – Context Handler

(S, R, A, E) – components of the AuthZ request (Subject, Resource, Action, Environment)

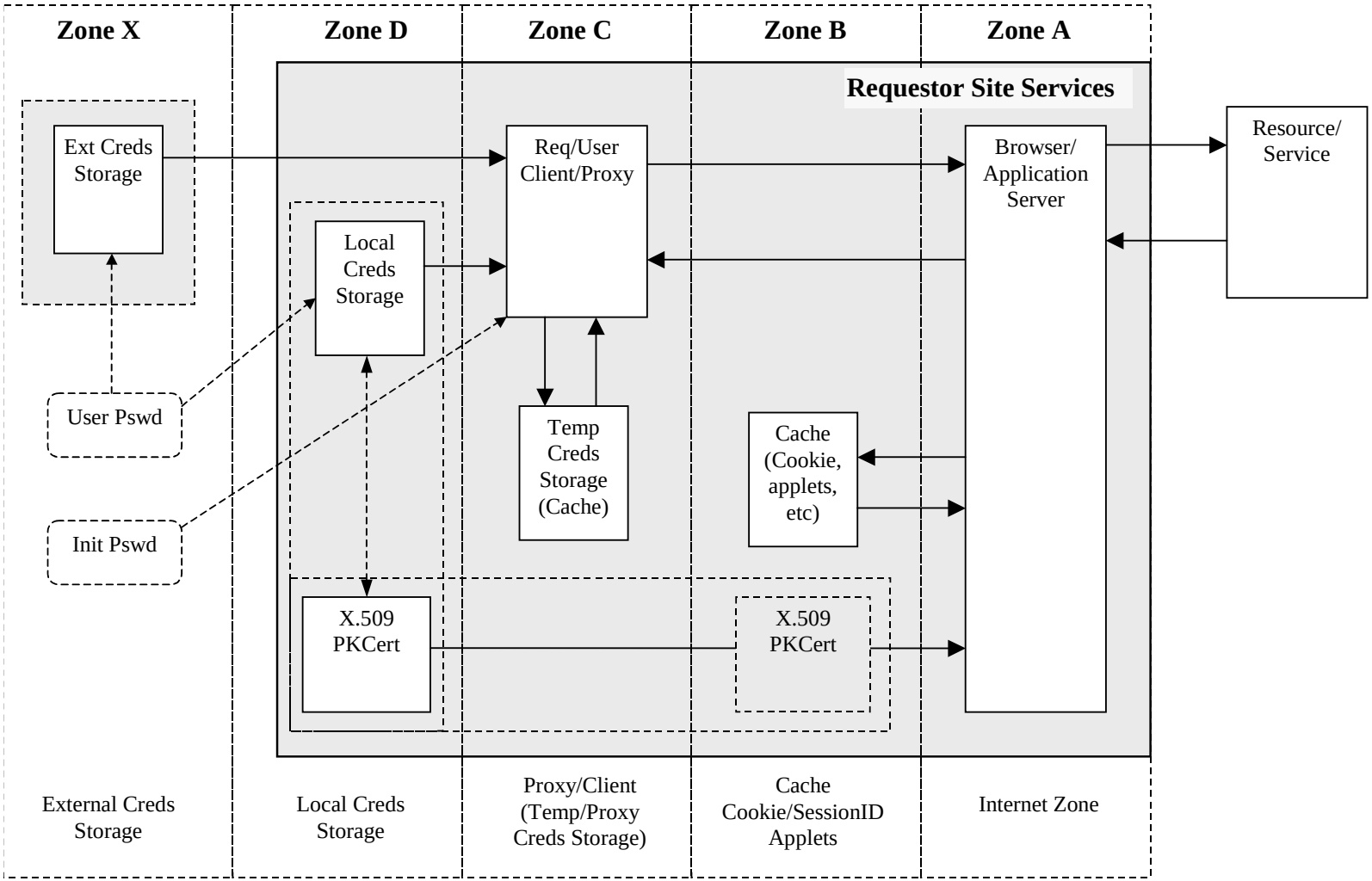
Resource Zone Security model for Grid/Web Services



Goal: Consistent Grid systems design and analysis (and e.g. threats analysis)

- Defines security zones for the resource/service and user/client

Client/Requestor Security Zones model for Grid/Web Services



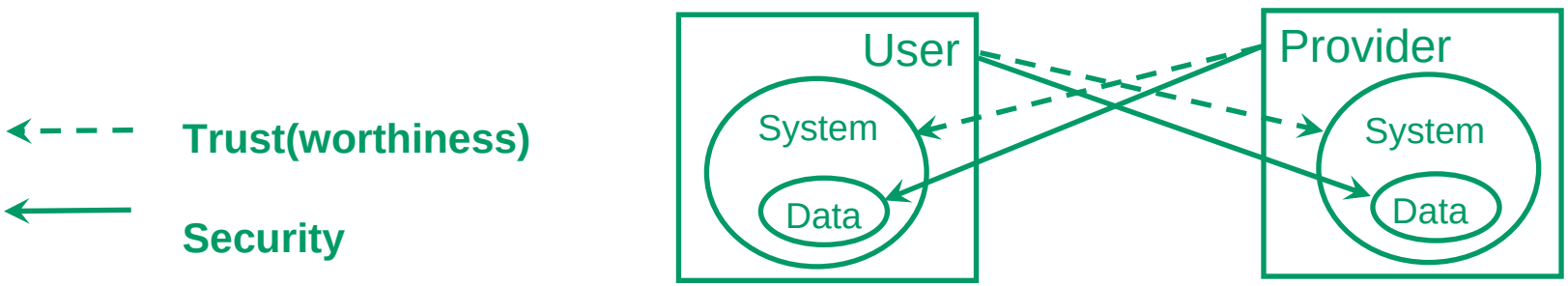
Extending User Controlled Security Domain in Virtualised Workspace Service (VWSS)

Different sides of the Security and Trust

- Modern paradigm of remote distributed services and digital content providing makes security and trust relations between User and Provider more complex
- User and Service Provider – two actors concerned with own Data/Content security and each other System/Platform trustworthiness
- Two other aspects of security/trust
  - Data stored vs Data accessed/processed
  - System Idle vs Active (running User session)

High-tech industry use case – Can we trust running analysis on the competitor's facility?

- *Sharing and/or offering unique/complex analytical instruments is limited by not sufficient protection of the instrument's hosting environment*
- *Analysis may need to be done on highly confidential specimens*
- *How to make remote environment trusted by even competitors*



Trusted Computing Platform Architecture (TCPA)

Promoted by the Trusted Computing Group (TCG)

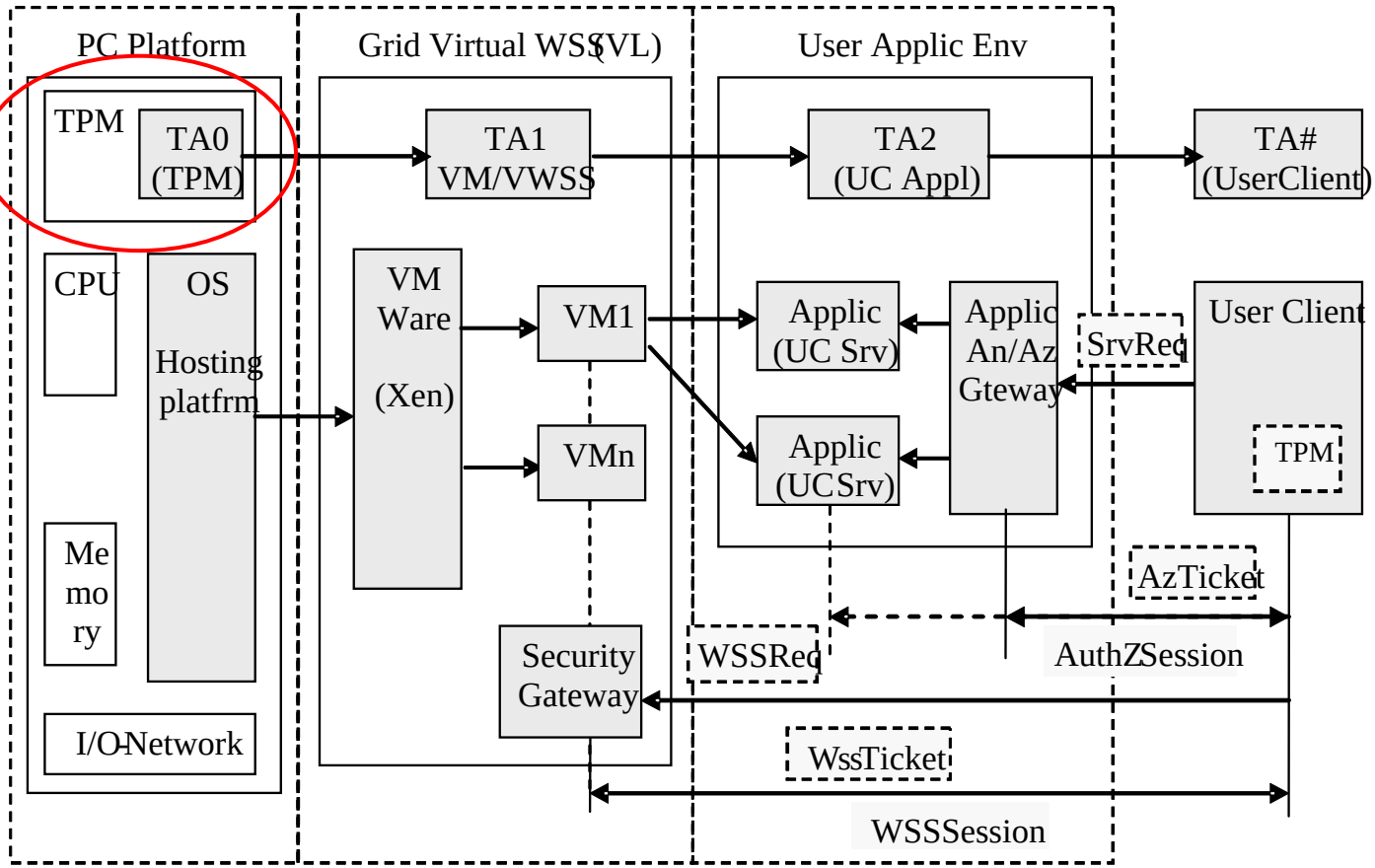
- Basis for building and managing controlled secure environment for running applications and processing (protected) content
  - https://www.trustedcomputinggroup.org/home
- Standards for trusted network, client, server and mobile agent
- TMP software stack (TSS) defines API's for remote access, Identity Mngnt, PKI, Secure e-mail, file/folder encryption, etc.
- Can be considered as a TCB implementation for distributed computing environment

TCPA components

- **Trusted Platform Module (TPM)**
- "Curtained memory" in the CPU
- Security kernel in the OS and security kernel in each application
- Back-end infrastructure of online security servers maintained by hardware and software vendors

Trusted Network Connect (TNC) – to enforce security policies before and after endpoints or clients connect to multi-vendor environment

User-controlled Virtual Workspace Service (VWSS-UC) – Proposed 3 layer model



- Trust Anchors: T0 (TPM) – TA1 (VM/VWSS) – TA2 (Application) – TA# (User)
- WVSS session and Application AuthZ sessions

PKI vs Identity Based Cryptography (IBC)

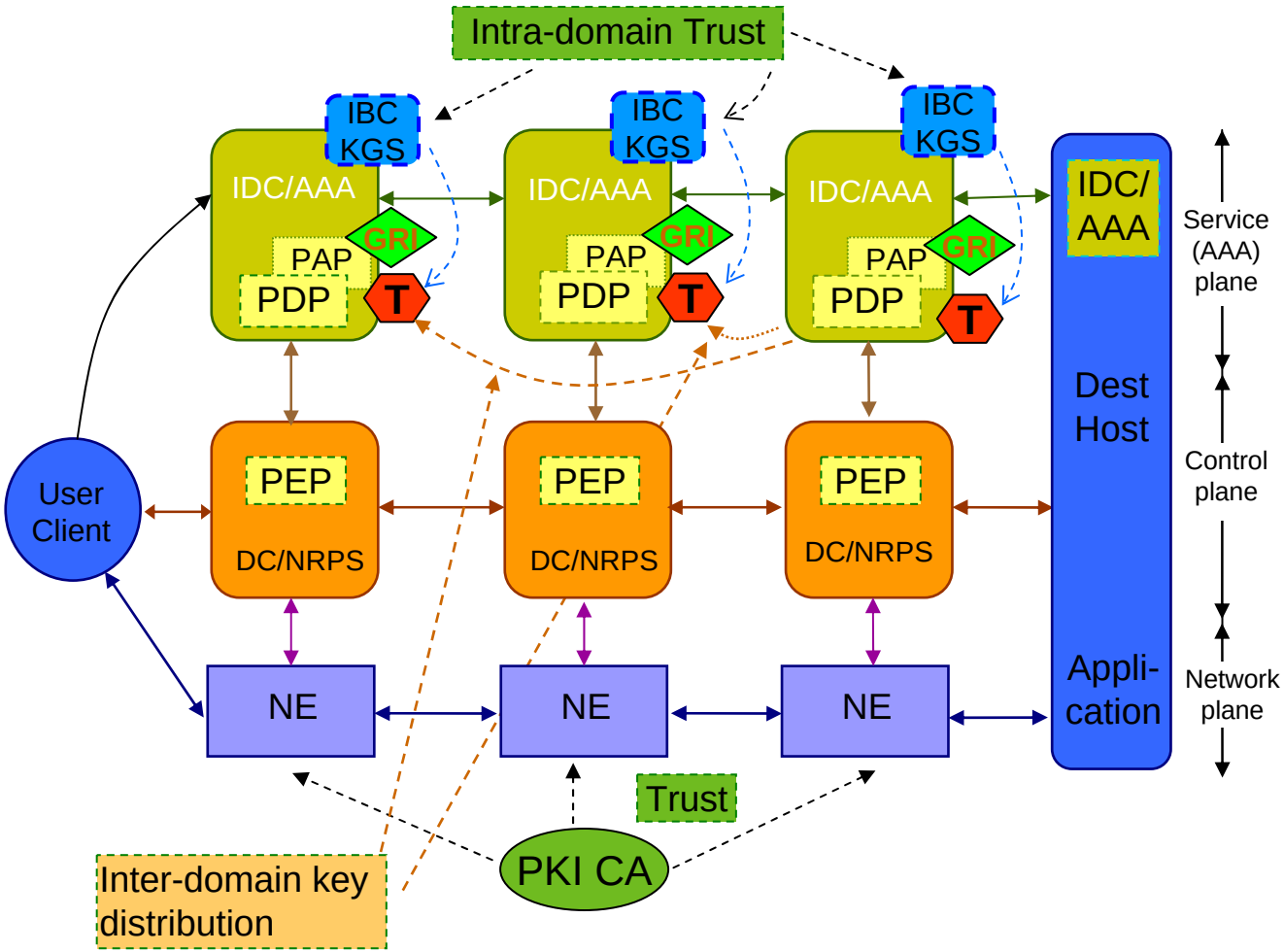
Uses publicly known remote entity's identity as a public key to send encrypted message or initiate security session

- Idea was proposed by Shamir in 1984 as an alternative to PKI and implementation by Dan Boneh and Matthew K. Franklin in 2001
- Identity can be email, domain name, IP address
- Allows conditional private key generation

Requires infrastructure different from PKI but domain based (doesn't require trusted 3rd party outside of domain)

- Parties may encrypt messages (or verify signatures) with no prior distribution of keys between individual participants
- Private key generation service (KGS)
  - Generates private key to registered/authenticated users/entities
  - To operate, the PKG first publishes a master public key, and retains the corresponding **master private key** (referred to as *master key*).
  - Given the master public key, any party can compute a public key corresponding to the identity *ID* by combining the master public key with the identity value.
- Exchange inter-domain trust management problem to intra-domain trust

Identity Based Cryptography (IBC) infrastructure operation when distributing token keys in multidomain NRP



- Uses intra-domain trust relation in exchange to simpler inter-domain trust management
- Simplifies key management problem
- Allows flexibility in deploying/configuring intra-domain network path/infrastructure
- Used at deployment stage
- IBC KGS are setup independently but publish their public parameters

Summary and Future Research

- Taxonomy of the Grid security issues
- Authorisation session and security context management framework and mechanisms
- Security zones model
- Contributing to OGF standardisation
- Adding new features to Grid middleware and AuthZ frameworks
  - GAAA-TK as a basis (developed by University of Amsterdam, currently in the framework of the EU Phosphorus project)
- Security models in Grid and Cloud Computing
  - Just as a matter of fashion :-)

Grids and/vs Clouds – Unconference discussion

The following major issues identified

1. Who is concerned?
   - Developers, and not managers
2. Clouds are suitable for individuals tasks and Grids are natively developed to support collaboration and resources and users federation
3. Grid uses security/trust model with “zero” risk, and Clouds accept “non-zero” risk model because of inherited payment cards security

- Data are part of the Grid infrastructure but not seen as a part of Clouds

Additional materials

- Obligations in other AuthZ and policy based management frameworks
- Multi-layer vs Multi-level security models
- Identity Based Cryptography

What's beyond AuthN/Z services - Application vs Security service view

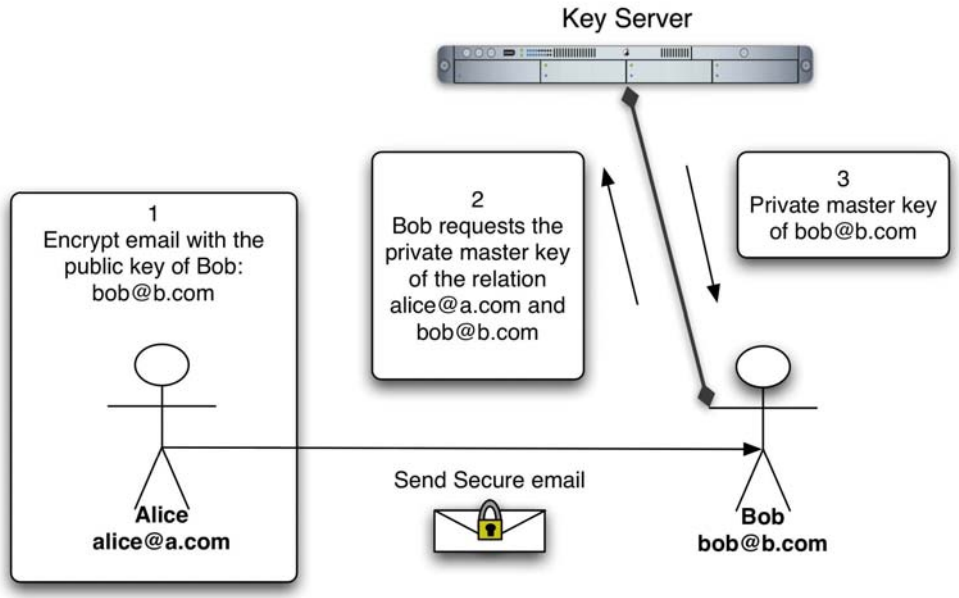
- Authentication – first/initial step in accessing a system or handling service request
  - Creating process, invoking service or object
  - Retrieving user attributes
  - In general, creating security context for further command/service execution
- Authorisation
  - Applied to user commands/actions, or managed objects
  - Starting/executing process/job/request
  - Creating AuthZ session and AuthZ context
    - Attribute mapping and policy Obligations
- Managing security and AuthZ context
  - User AuthZ session – e.g. web browser cookie
  - Process environment – e.g. Unix processes environment
  - Managed Object property – e.g. job, running code permissions, agents

Obligations in other AuthZ and policy based management frameworks

XACML policy obligations definition is originated from the two other concepts

- Provisional Authorisation model by Kudo (implemented in the IBM's XACL)
  - Includes Provisional AuthZ Module (PAM) and Request Execution Module (REM)
  - PAM can authorise a request provided the requestor or system (actually REM) will take some security actions, defined as "provisional actions" prior to the request execution, e.g. presenting additional credentials, signing privacy statements, logging events, etc.
- Obligation policies (by Sloman) are defined together with Authorisation policies as part of the policy based management in distributed systems
  - Obligation policies provide simpler way of enforcing state-based policies over managed objects
    - Stateful part of the management policies can be implemented as obligation policies
  - Requires trusted manager (that can be treated similar to the Reference Monitor concept in the Trusted Computing Base (TCB))
  - Provisions and obligations concepts have been further developed by Bettini et al

Identity Based Cryptography (IBC) - Operation



Four algorithms form a complete IBE system (as proposed by Dan Boneh and Matthew K. Franklin):

Setup: This algorithm is run by the PKG one time for creating the whole IBE environment.

- The master key is kept secret and used to derive users' private keys, while the system parameters are made public. It accepts a security parameter k (i.e. binary length of key material) and outputs:
- A set P of system parameters, including the message space and ciphertext space M and C, a master key Km (master) .

Extract: This algorithm is run by the PKG when a user requests his private key.

- It takes as input P, Km and an identifier ID={0,1} and returns the private key D for user ID.
- Requires strong authentication and out of IBE model scope

Encrypt: Takes P, a message m={M} and ID={0,1} and outputs the encryption c={C}.

Decrypt: Accepts d, P and c={C} and returns m={M}

Multi-Level Security Models

Bell–LaPadula (BLP) data confidentiality model

- No write down - No read up

Biba model data and control system Integrity

- No write up - No read down

Clark – Wilson data and process/operations integrity policy (for reliable business operation)

- Data and processes integrity criteria
  - Authentication of all user accessing system
  - Audit – all modifications should be logged
  - Well-formed transactions
  - Separation of duties
- Defines enforcement rules E1 – E4 and certification rules C1-C5 for procedures and entities
  - TP – transformational procedure and IVP – integrity verification procedure
  - CDI – constrained data item and UDI - unconstrained data Item
- Use as a basis for OS security/integrity policy (e.g., Windows, Linux)

Multi-layer vs Multi-level security models

Multi-layer security means the following:

1) layers are defined according to the OSI reference model, i.e. data layer, network, transport, application, what can be mapped into e.g. network element/node, router/network, application
2) security services and security mechanisms are defined in such a way that they can be applied to network/security layers independently/orthogonally. This means e.g. that many (the same) security services can be used at the different networking layers

Internet/network services, client/server, Web services, service–oriented applications use multi-layer security model.

Multi-level security means the following:

1) Security levels are defined as:
   - object/document/resource security classification level, e.g. public, secret, top secret, and
   - subject/user/requestor clearance level that allows access to this resources.
2) the system corresponds to the Trusted Computing Base (TCB) model and uses centralised security management model (aka Reference Monitor (RM) in TCB). This can be explained as similar to OS security.
   - RM regulates the access of subjects to objects on the basis of their security parameters: the access privileges (security clearance) of subjects, and the protection attributes (classification level) of objects.
3) In application to networked/distributed applications this means that all traffic is completely encrypted and labeled/tokenised, there is very strict and well defined procedure for managing and establishing keys.

Typical examples of MLS are operating system security, military applications' security.

Obligations Handling Stages

Obligation0 = tObligation => Obligation1 (“OK?”, (Attributes1 v Environments1))
=> Obligation2 (“OK?”, (Attributes2 v Environments2))
=> Obligation3 (Attributes3 v Environments3)

Obligation0 – (stateless or template)
Obligations are returned by the PDP in a form as they are written in the policy. These obligations can be also considered as a kind of templates or instructions, tObligation.

Obligation1 and Obligation 2
Obligations have been handled by Obligation handler at the SCAS/PDP side or at the PEP side, depending on implementation. Templates or instructions of the Obligation0 are replaced with the real attributes in Obligation1/2, e.g. in a form of “name-value” pair.

- The result of Obligations processing/enforcement is returned in a form of modified AuthzResponce (Obligation1) or global Resource environment changes
- Obligation handler should return notification about fulfilled obligated actions, e.g. in a form of Boolean value “False” or “True”, which will be taken into account by PEP or other processing module to finally permit or deny service request by PEP.
- *Note. Obligation1 handling at the SCAS or PDP side allows stateful PDP/SCAS*.

Obligation3
Final stage when an Obligation actually takes effect (Obligations “termination”). This is done by the Resource itself or by services managed/controlled by the Resource.