GAAA Authorisation Framework and Security for Grids

Advanced Internet Research Group Update

MWSG7 – 14-15 December 2005, Amsterdam

Yuri Demchenko <demch@science.uva.nl>
AIRG, University of Amsterdam

Outline

- AIRG Projects and generic AAA Architecture development
  - Policy based access control in Collaboratory.nl (CNL/CNL3)
  - Multidomain AAA services in Optical Light Path Provisioning (OLPP) – GigaPort-NG Research on Network
- JRA3/Security in EGEE
  - Security middleware development
  - Grid and Web Services Vulnerabilities and Threats analysis and model

AIRG projects and GAAA development

- **GigaPort-NG Research on Network (2004-2008+)**
  - GAAA architecture for policy/token based networking (TBN)
  - GAAA-P Authorisation profile for OLPP and complex resource provisioning
- **Collaboratory.nl (CNL3 – 2005-2006)**
  - Distributed Security Architecture for Open Collaborative Environment (OCE)
  - Job-centric security model and Role/Policy based Access Control
- **Grid related EU projects**
  - EGEE
    - Security middleware development (site local security – LCAS/LCMAPS, glexec, etc.)
    - Authorisation framework and SAML/XACML (suggested for EGEE-II)
    - Operational security and vulnerabilities and threats analysis
  - NextGrid
    - Dynamic security in next generation Grids

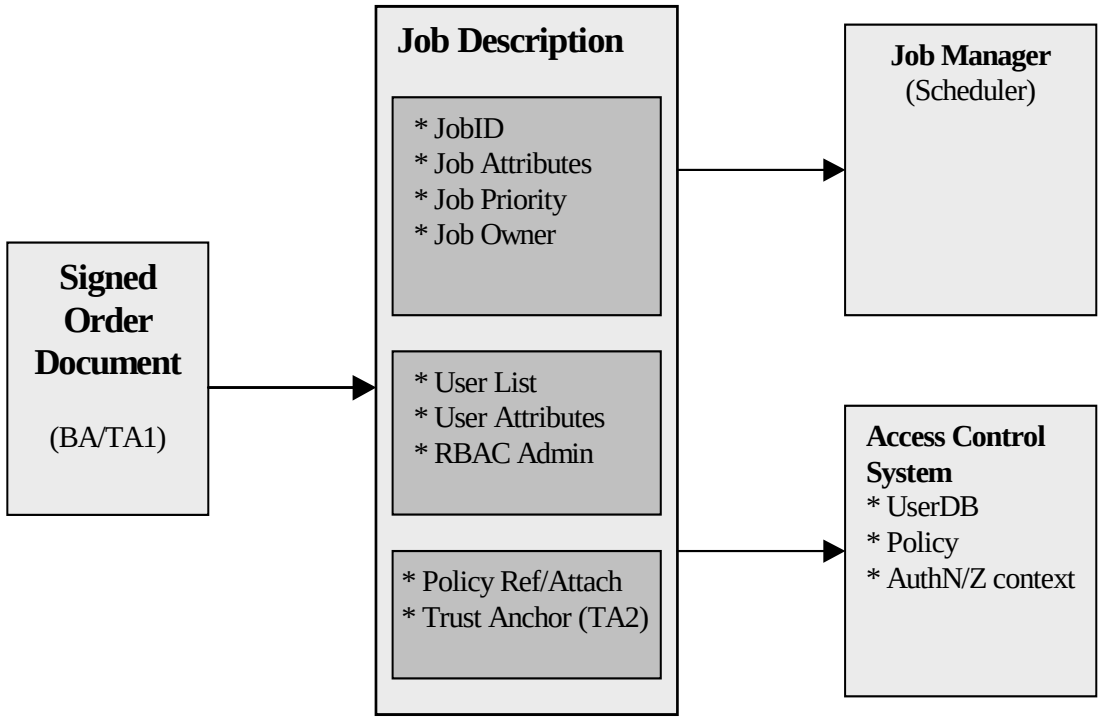
GAAA AuthZ f/w development (cont.)

- **General research and development**
  - Multidomain complex resource provisioning (OLPP as a use case)
    - Provisioning workflow and local policy enforcement
  - Trust relations and dynamic trust management for Collaborative Grid environment
    - Using VO concept for dynamic security associations
  - Authorisation service performance
  - (Customer driven) Access Control for SOA and Grid

OCE security requirements and common problems

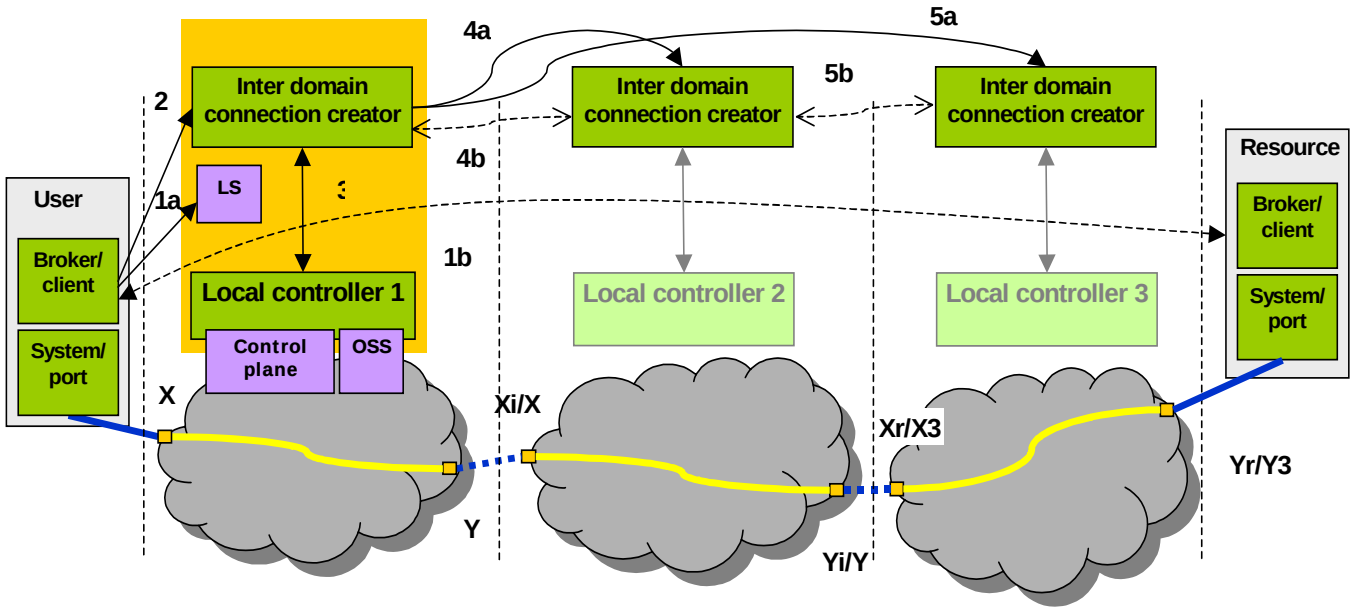
- **OCE specific security requirements**
  - Dynamic and multidomain
  - Customer driven
  - Human controlled and interactive
  - Data protection: personal, experimental data and metadata
- **Common problems - Access Control**
  - Authorisation service performance
    - Using XML based ticket/token – integrity and secure context management
  - Session management in RBAC Authorisation
  - Key management and trust relations in distributed access control infrastructure
  - Compatibility and integration with existing access control tools
    - Different policy formats mapping for flexible policy exchange and combination

OCE/ CNL Security built around Job description



- Job Description as a semantic object defining Job attributes and User attributes
  - Requires document based or semantic oriented Security paradigm
- Trust domain based on Business Agreement (BA) or Trust Agreement (TA) through PKI

OLP provisioning operation – Simple multidomain model



- Step 1. Path lookup to the target system or resource
- Step 2. Building interdomain connection
- Step 3. Reservation of calculated path
- Step 4. Provision reserved OLP

Required AAA/Security services functionality for OLPP

- Authentication and Identity management
- Authorisation
- Attribute management
- Federation
- Trust management

- **Conceptual issues and models**
  - OLP provisioning model and process must be defined in details
    - It is used as a basis for defining AAA/Security functionality and operation
  - GAAA Authorisation framework for complex resource provisioning
    - Multiple resources and multiple domains
    - Multiple policies combination and evaluation
  - Driving policy re-factoring/implementation by separating flow management and policy enforcement
  - Dynamic security context and trust management model
  - VO infrastructure and management for dynamic user controlled service provisioning

Extending GAAA Authorisation Framework for OLPP

- **GAAA AuthZ framework – two basic profiles are defined**
  - GAAA-RBAC for Collaborative Environment
  - GAAA-P for interdomain network/resource provisioning
- **Major GAAA-P components/extensions**
  - Workflow control in the GAAA based provisioning model
    - WSFL and WSBPEL as upper layer to (stateless) WS/WS-Security
  - Dynamic trust management using federated trust model
    - Based on dynamic VO federation model
    - Compatibility with GridShib-SAAS
  - Attributes and metadata resolution and mapping
    - Support of common naming scheme and resolution
  - Policy combination and aggregation
    - For complex multi-component and multidomain resources
    - For combined policy audit/evaluation

Workflow management

- Separate policy evaluation and flow control and make flow control interpretable at runtime
  - Policy is a static set of rules that in general can be defined by the agreement between user and provider
  - Workflow is an instant dynamic process that orchestrates interaction of multiple services and processes to deliver final service to the requestor
- Workflow management for two basic provisioning scenarios
  - ***Centralised:*** Reservation (and provisioning) is controlled by one of domain Interdomain Connection Controller (ICC), e.g. from user domain, and the workflow is managed by a single ICC
    - *individual policies are evaluated centrally and published into central repository*
  - ***Distributed:*** Reservation (and provisioning) is chained and the workflow object may need to be transferred between participating domains
    - *individual policies are evaluated locally in each domain, without populating policy between all participating domains*
- Available technologies and tools
  - OASIS BPEL and IBM's WSFL
  - Oracle and Apache plugins for Eclipse
  - ActiveBPEL, FreeFluo and Taverna (developed by myGrid, UK)

Dynamic trust management

- **Foundation for secure user controlled service provisioning**
  - Security context should be present explicitly or implicitly in any session on the protected resource
    - Such security context is established during session start based on the positive AuthZ decision
  - During dynamic trust negotiation, in general, or security context establishing, in particular, negotiating parties must present initial credentials
  - The framework for (dynamic or session based) trust and credentials negotiation is defined in two complimentary specifications WS-Trust (WST) and WS-SecureConversation (WSSC)
    - WST defines SOAP based mechanisms for brokering trust relationships, requesting and returning security tokens.
    - Requests for security tokens are made by sending a Request Security Token (RST) to the Security Token Service (STS)
  - NOTE: Initial trust relations (or security context) establishment is considered outside of the WS-* scope and must be presented in all WS-* interactions in a form of trust (TA) or business anchor (BA)
    - VO is suggested for the initial trust introduction

Dynamic Security Associations

- **Session** – establishes security context in the form of session key that can be a security token or simple UID bound to secure credential/context
  - Session may associate/federate users, resources and actions/processes
- **Job/workflow** – more long-lived association and may include few sessions
  - May need to associate more distributed collection of users and resources for longer time required to deliver a final product or service
  - Job and workflow may contain decision points that switch alternative flows/processes
  - Security context may change during workflow execution or Job lifetime
  - Job description may contain both user and resource lists and also provide security policy and trust anchor(s) (TA)
- **Project or mission oriented cooperation** – established for longer time cooperation (involving people and resources) to do conduct some activity
  - This is actually the area of currently existing VO associations
- **Inter-organisational association or federation** – established for long-term cooperation, may have a wide scope of cooperative areas
  - This is the area of inter-university (Shibboleth-based) associations

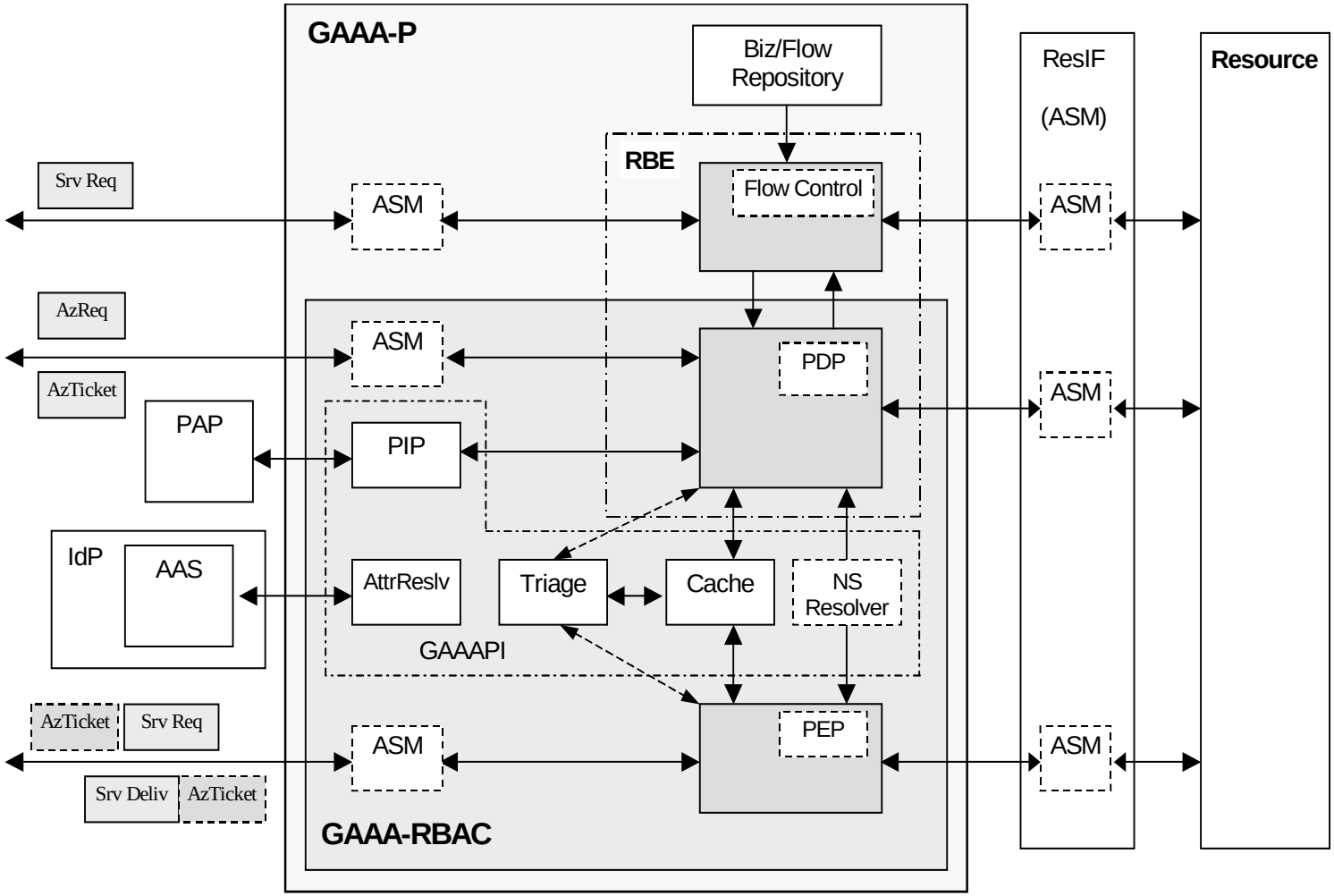
Using VO concept for Dynamic Security Associations

- **Dynamic VO infrastructure can provide a solution for dynamic distributed trust management and attribute authority**
  - VOMS provides basic functionality for creating ad-hoc dynamic VO associations
- **(Conceptual) VO operational models**
  - **User-centric VO (VO-U)** - manages user federation and provide attribute assertions on user (client) request
  - **Resource/Provider centric VO (VO-R)** - supports provider federation and allows SSO/access control decision sharing between resource providers
  - **Agent centric VO (VO-A)** - provides a context for inter-domain agents operation, that process a request on behalf of the user and provide required trust context to interaction with the resource or service
  - **Project centric VO (VO-G)** - combines User centric and Provider centric features what actually corresponds to current VO use in Grid projects

Extending GAAA Toolkit to support new functionality

- SAML 2.0 assertion and protocol support, including SAML XACML profile that will simplify AuthZ tickets management
- XACML policy support as a policy meta-format and exchange format
- Simple policy management tools supporting multiple policy formats, first of all, AAA-format and XACML
- Support for different types of secure credentials, in particular, X.509 PKI Certificate and Attribute Certificate, SAML assertions, and related callouts to issuing authorities, in particular VOMS and Shibboleth
- WS-Trust Secure token support and Secure Token Service (STS) functionality for credentials mapping and dynamic trust management
- Integration with GT4 and gLite Authorisation Framework
  - Using GT4 WS/messaging firmware to provide WS-based access to GAAA_tk authorisation service, to allow easy GAAA_tk integration into different applications
  - Adding GAAA AuthZ callouts to GT4/gLite AuthZ framework; this will allow using GAAA RBE as one of regular services for GT4 and gLite
  - Integrating GAAA AuthZ/RBE into GT4 AuthZ framework as one of PDP's

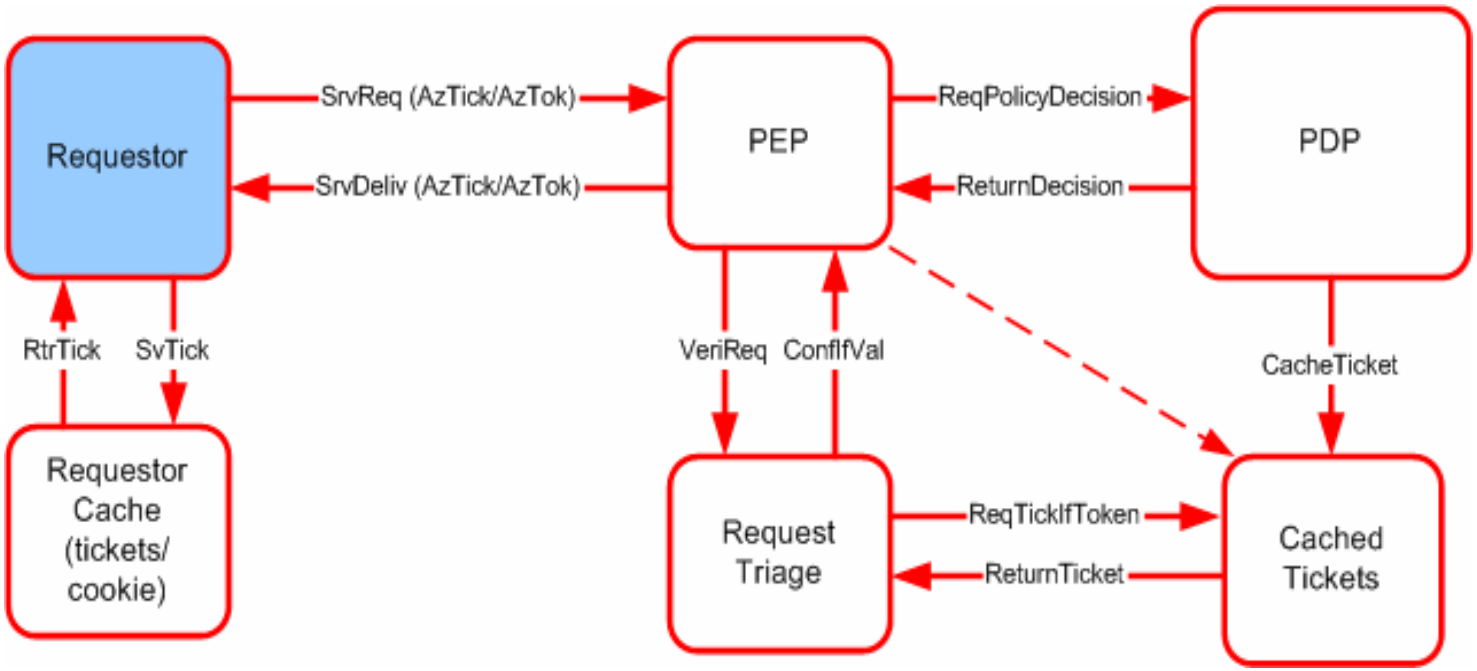
Extended GAAA Toolkit structure



Session management in CNL2/CNL3 Access Control system

- Maintaining AuthZ session is a part of the generic RBAC functionality
- Session can be started only by authorised Subject/Role
  - Session can be joined by other less privileged users
- SessionID is included into AuthzTicket together with other decision attributes
  - Signed AuthzTicket is cached by PEP or PDP
- If session is terminated, cached AuthzTicket is deleted
  - Note: AuthzTicket revocation should be done globally for the AuthZ trust domain ***– often missed functionality***
- Triage functionality and module proposed for initial AuthZ request investigation and evaluation
  - The idea was picked up by the GEANT AA activity and being developed

Tickets/Tokens handling in AuthZ system



- AuthzTicket is issued by PDP and may be issued by PEP
- AuthzTicket must be signed
- AuthzTicket contains all necessary information to make local PEP-Triage Request verification
- When using AuthzTokens, AuthzTickets must be cached; Resolution mechanism from token to ticket must be provided

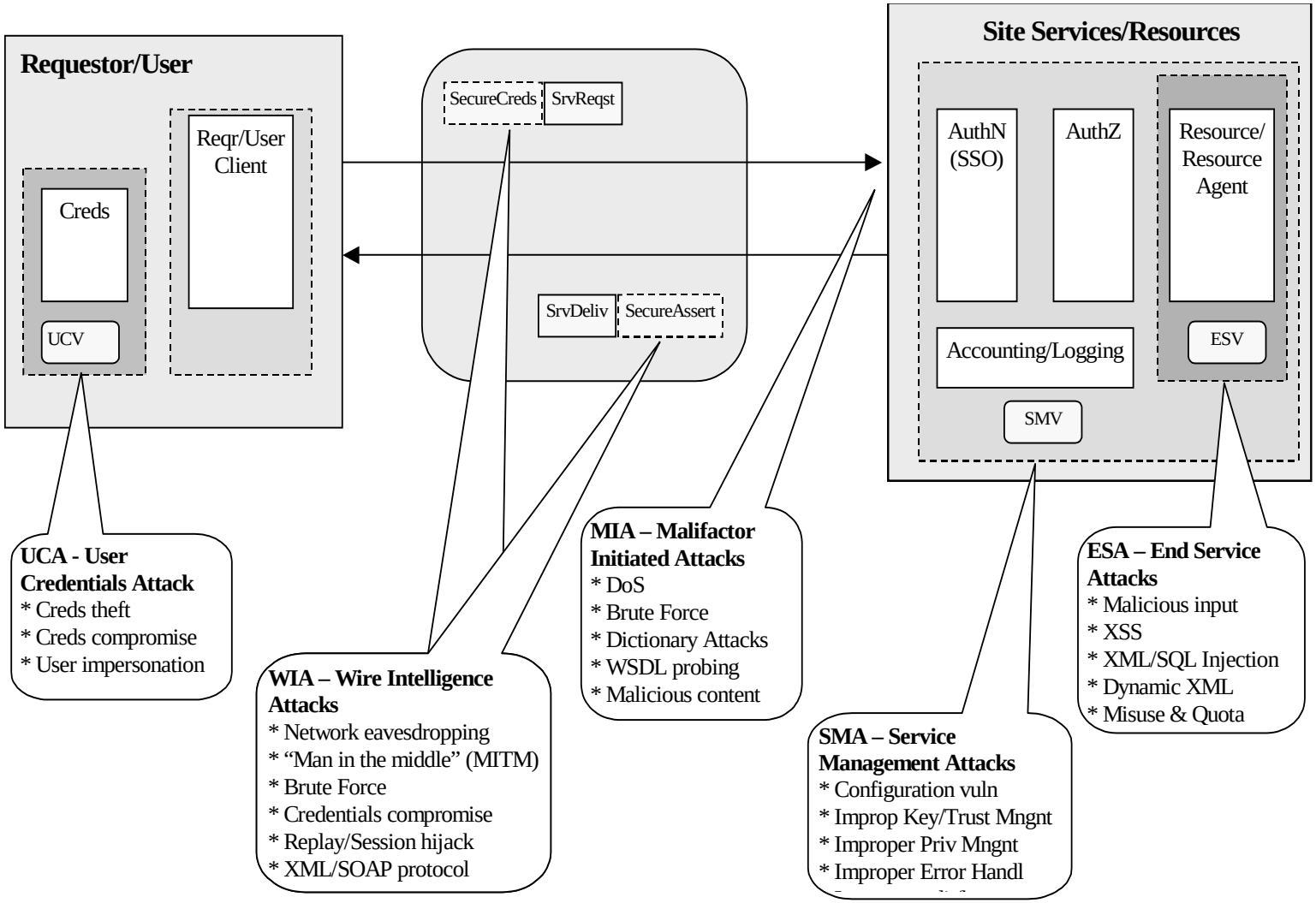
Grid Security Vulnerabilities and Threats Analysis

- **Developers and Grid Operational Centers (GOC) know major security vulnerabilities**
  - Those that are *actually* obvious
  - We can expect more will be discovered when we apply regular security vulnerability analysis and risk assessment
- **(Already perceived) Problems**
  - There is no common approach/model for analysing security vulnerabilities in Web Services and Grids
  - All security models and methodologies are complex and multifaceted
    - Grid is new but not unique – can benefit from already existing experience in other areas

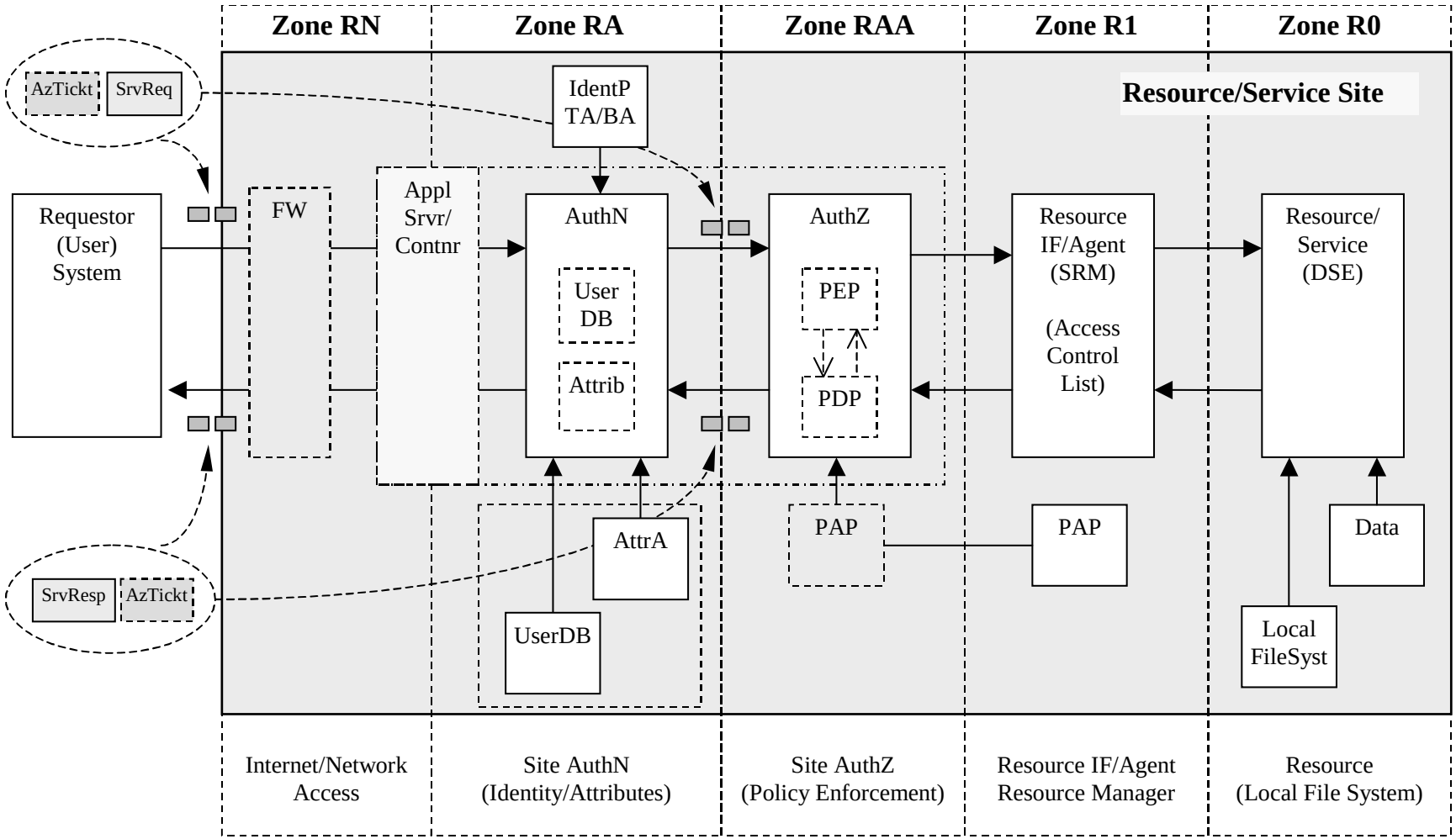
Web Services and Grid security model

- **End-to-end (or application-to-application) and data/job centric security model**
  - In contrary to point-to-point (host-to-host) and host-based security models in networking
    - With new attacking tools and spyware host based and p2p security model is proven to be vulnerable to credentials compromise
    - “Year 2004 is marked as the year when we lost our desktops” [somebody]
- **Security services re-use (in SOA) requires explicit security context management**
- **WS and Grids potentially exposed to the new kind of attacks and will attract another category of attackers**
  - ***“white collar” attacks***, in contrast to ordinary “blue collar” attacks, target vulnerabilities in applications to gain access to most valuable resources

Threats/Attacks grouping in interacting services



Service/Resource site security zones



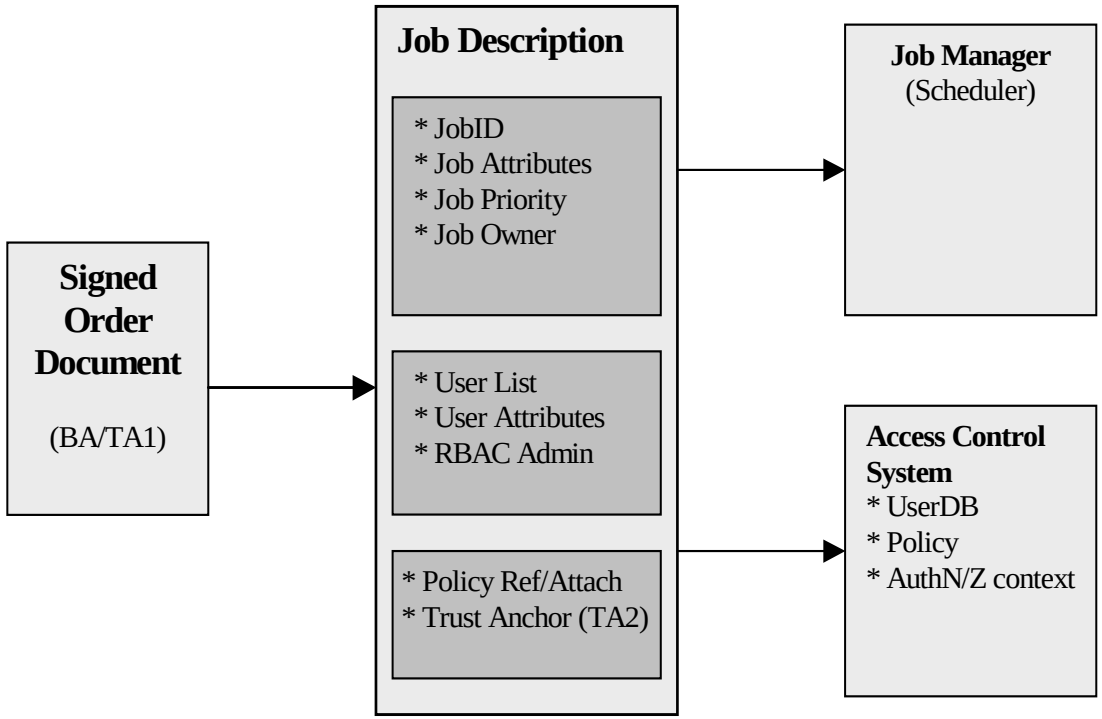
Grid Security Incident datamodel and XWS profile for IODEF

- **Special profile for the general Security Incident format**
  - Mostly security credentials compromise (e.g., private key, proxy credentials, etc.)
    - patterns of credential usage
    - broken chain of PKC/keys/credentials
  - Similar to Incident in financial industry
- **Provides suggestions for logging and auditing services**
  - What data to collect and how to present them
  - Potentially to be compatible with existing models and tools

Additional information

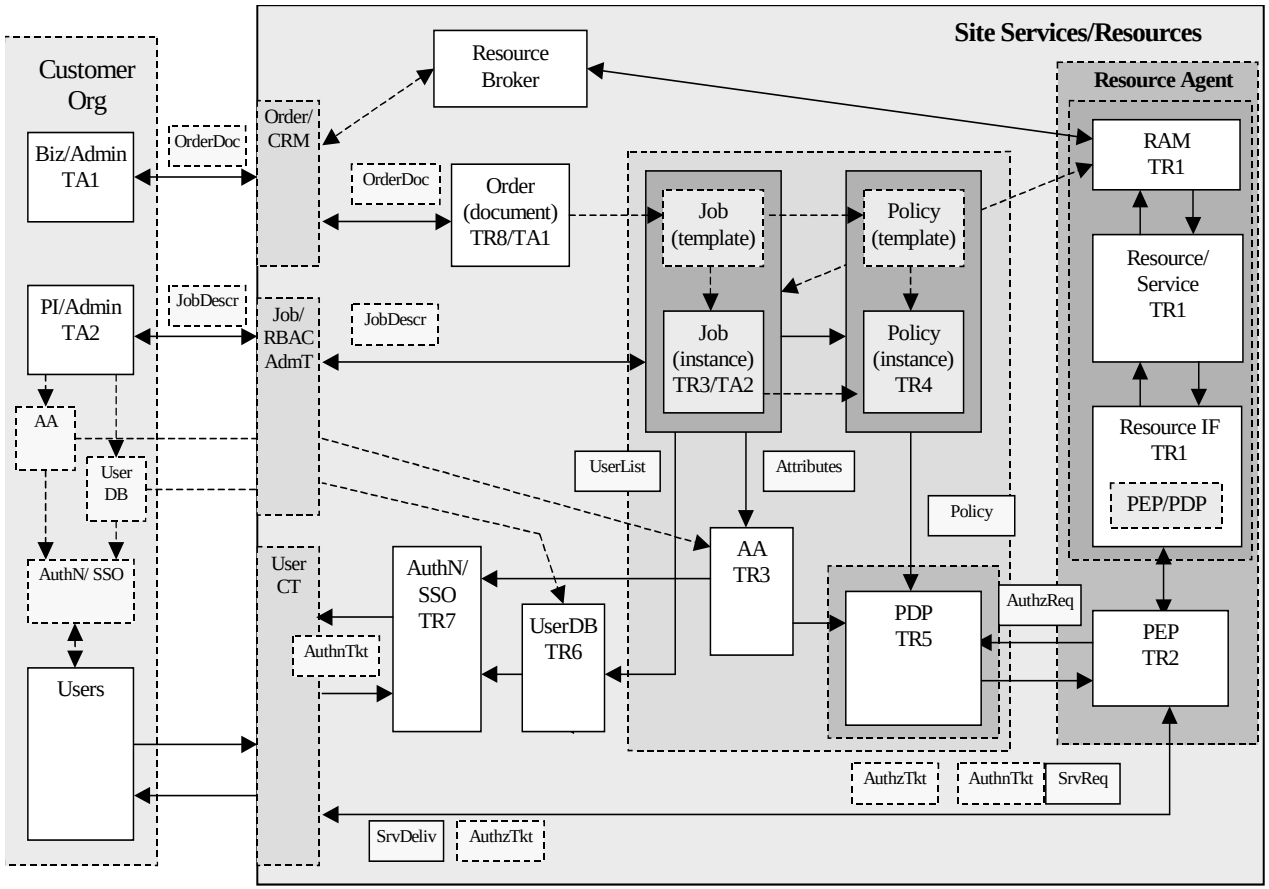
- Authorisation in complex resource provisioning
- Multiple/multidomain policy combination
- Authorisation in CNL2 Demo system
- CNL2 XACML policy format

OCE/ CNL Security built around Job description



- Job Description as a semantic object defining Job attributes and User attributes
  - Requires document based or semantic oriented Security paradigm
- Trust domain based on Business Agreement (BA) or Trust Agreement (TA) through PKI

Major interacting components and entities in the Job-centric security model

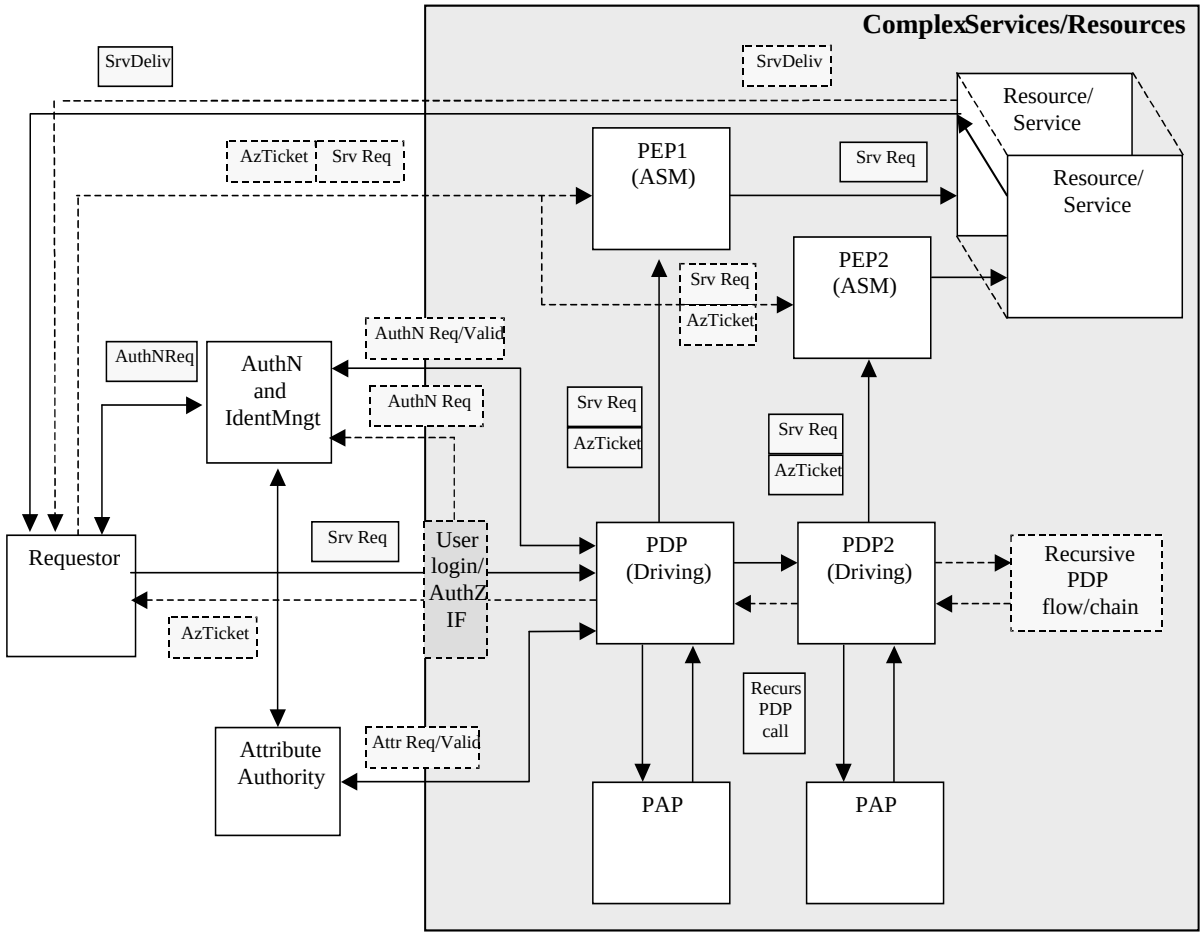


- **TA – Trust Anchor; TR# - trust path from root (resource); RAM – Resource Allocation and Management; UserCT – User Collaborative Tools**

General implementation suggestions for complex resource provisioning

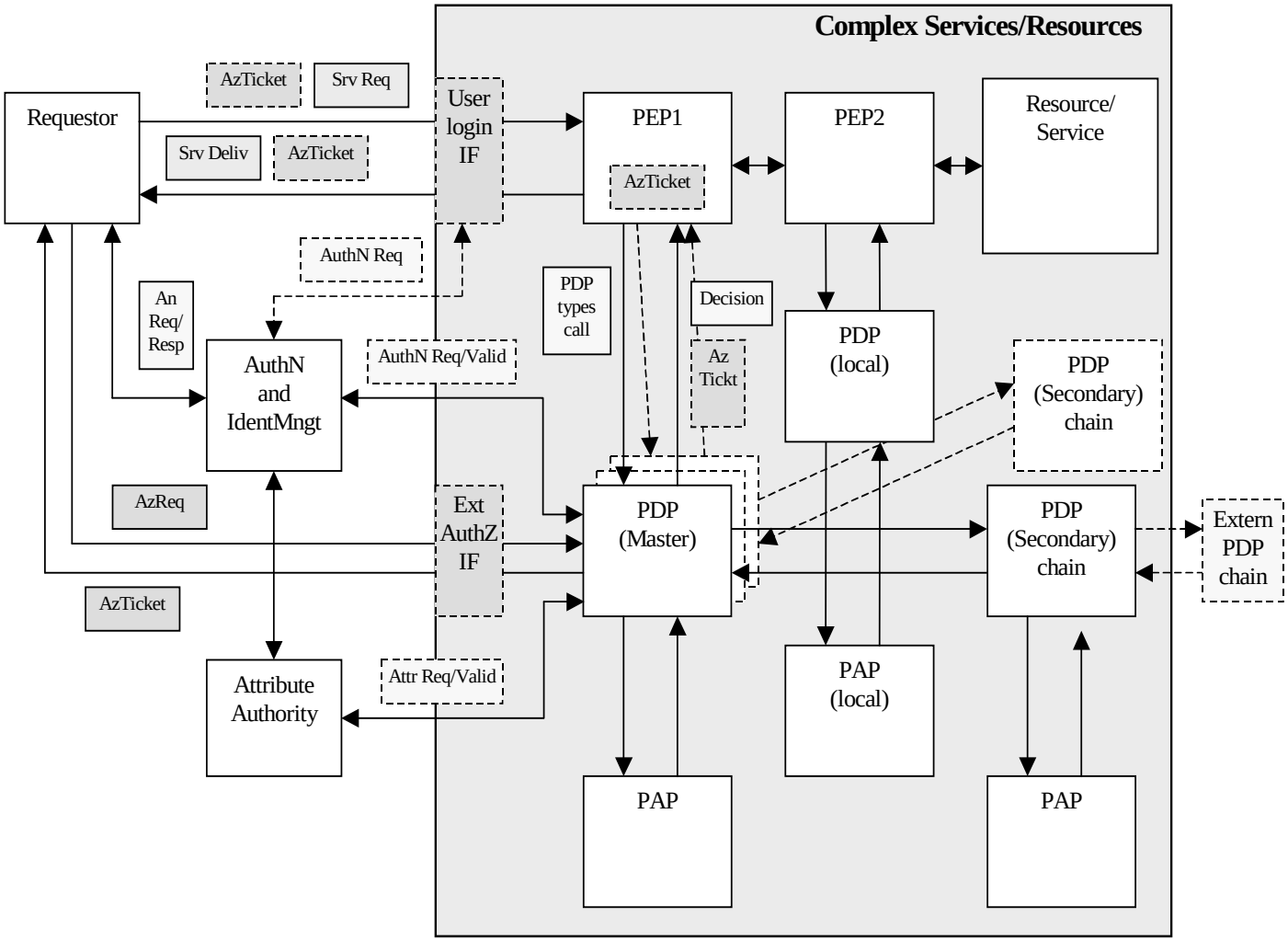
- PDP and PAP must share common namespace
- Policy and respectively PAP should be referenced in the request message explicitly or known to PEP and PDP a priory
- Every PEP in the chain of policy enforcement should take care of the whole request evaluation/enforcement by calling to a single (master) PDP.
  - PEP should not do multiple decision combination.
- Only one PDP should provide a final decision on the whole request
  - However, PEP may have a possibility to request different PDP types based on request semantics/namespace and referred policy
- When using ticket/token based access control model, the PEP should understand and have a possibility to validate the AuthZ ticket issued by trusted PDP
  - The AuthZ ticket should have validity and usage restriction and contain information about the decision and the resource.
- For the further validation of the AuthZ tickets/token, the PEP may cache the ticket locally to speed-up the validation procedure.

Authorisation in complex Resource/Service



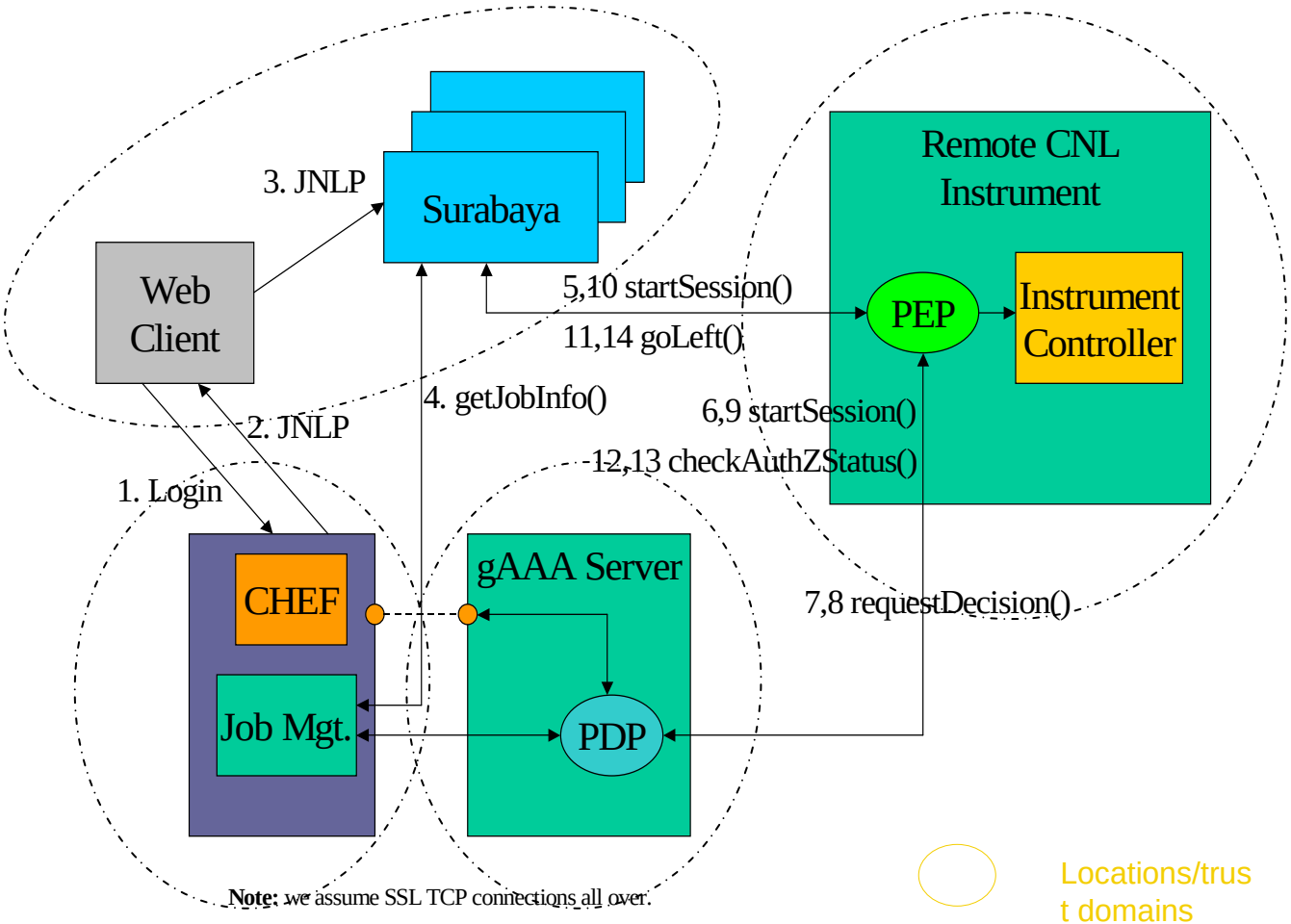
- Complex/multi-component resource
- Combined push and agent model

Multiple/multi-domain policies combination



- **Multiple policies and/or multiple PDP's**

Implementation: Authorisation Service operation in a CNL2 Demo system



- JNLP – Java Network Launch Protocol
- CHEF – Collaborative tool
- Surabaya – Collaborative Workspace environment
- GAAAPI Trust Domains Configuration

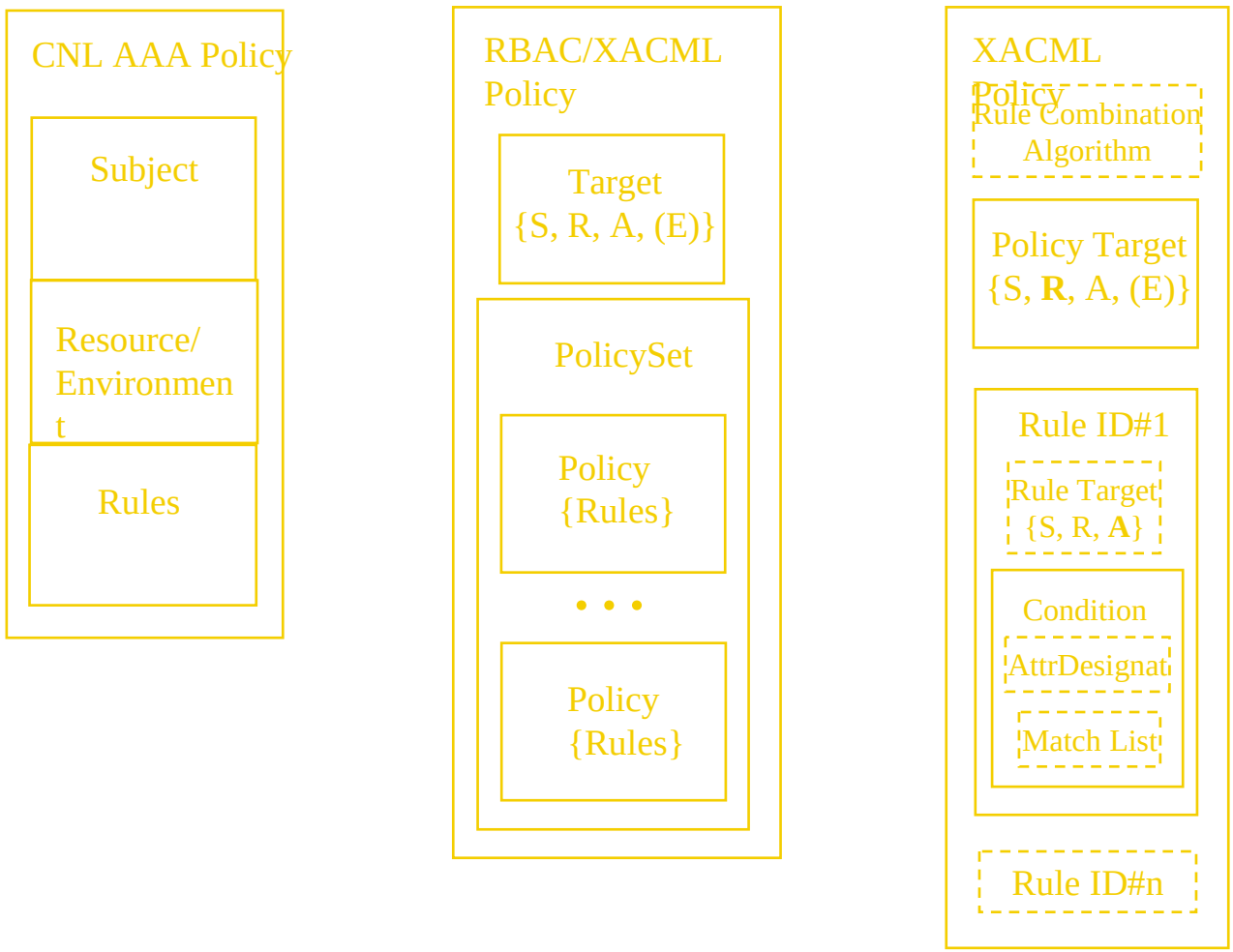
Traditional Access Control model – setting up trust and authority relations

- Policy, attributes semantics and namespaces are known a priory to all participating parties
  - A requestor knows what information to present to adhere to a specific policy and in what format
- PEP and PDP locations are known and interacting parties are known
- Trust relations between PDP, AA and resource are established
  - Resource trusts PDP's decision that can be delivered to a Resource in a form of AuthzTicket or based on default trust between PEP and Resource
  - Root of policy enforcement hierarchy, like in real life, belongs to the resource owner
- This approach is not sufficient for emerging Service Oriented Architecture (SOA)
  - In search of adequate trust description model

CNL2 AuthZ policy: RBAC using XACML format

- **Policy generation conventions**
  - Policy Target is defined for the Resource and may include Environment checking
  - Policy combination algorithm is “ordered-deny-override” or “deny-override”
  - Rule Target is defined for the Action
    - Rule’s Condition provides matching of roles which are allowed to perform the Action
  - Access rules evaluation
    - Rules are expressed as permissions to perform an action against Subject role
    - Rules effect is “Permit”
  - Subject validation – is not supported by current XACML functionality
    - TODO: add Function or do validation at/by PEP or Context Handler

AAA Policy and XACML Policy formats



Requestor/User site security zones