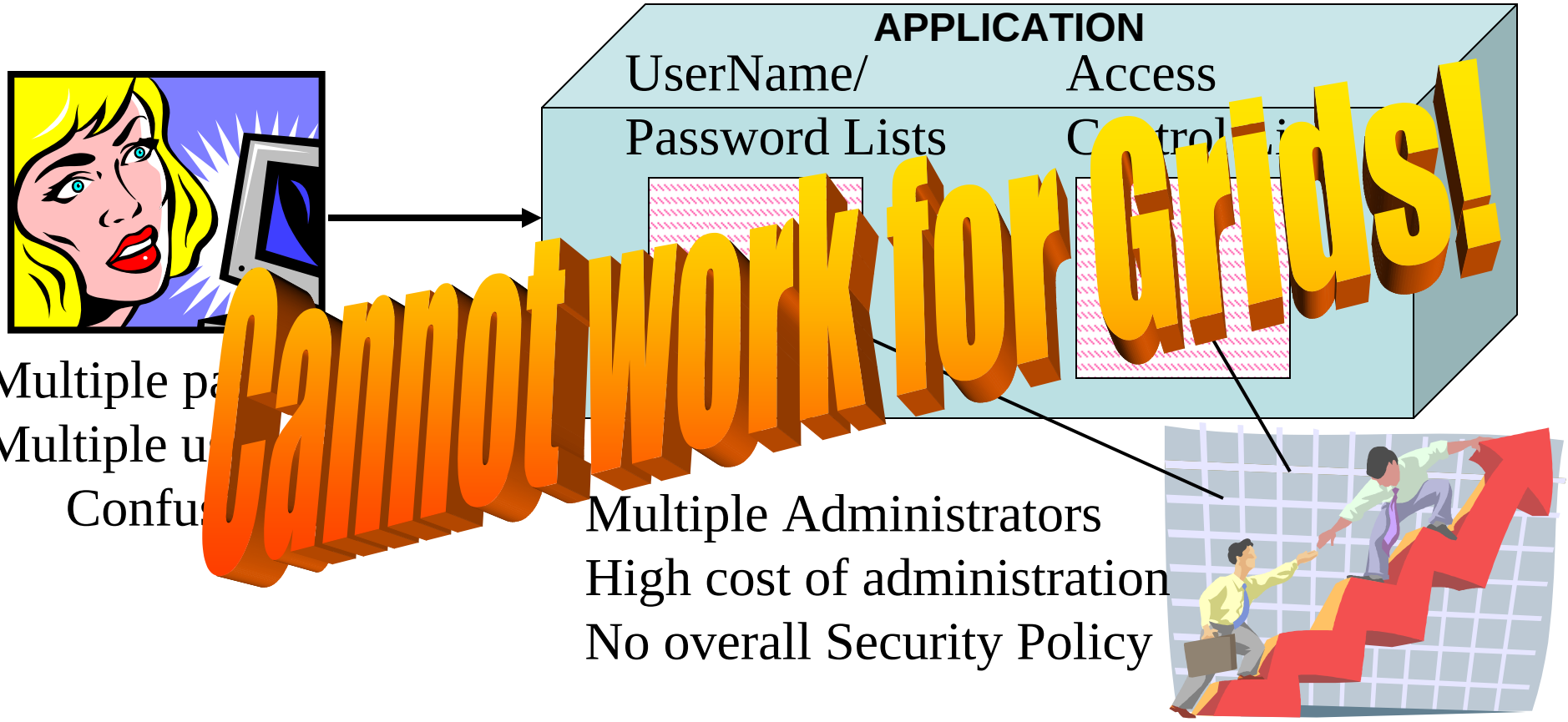
Modular Authorisation for Grids

David Chadwick

University of Kent

Traditional Applications

- Authentication and Authorisation are Internal to the Application

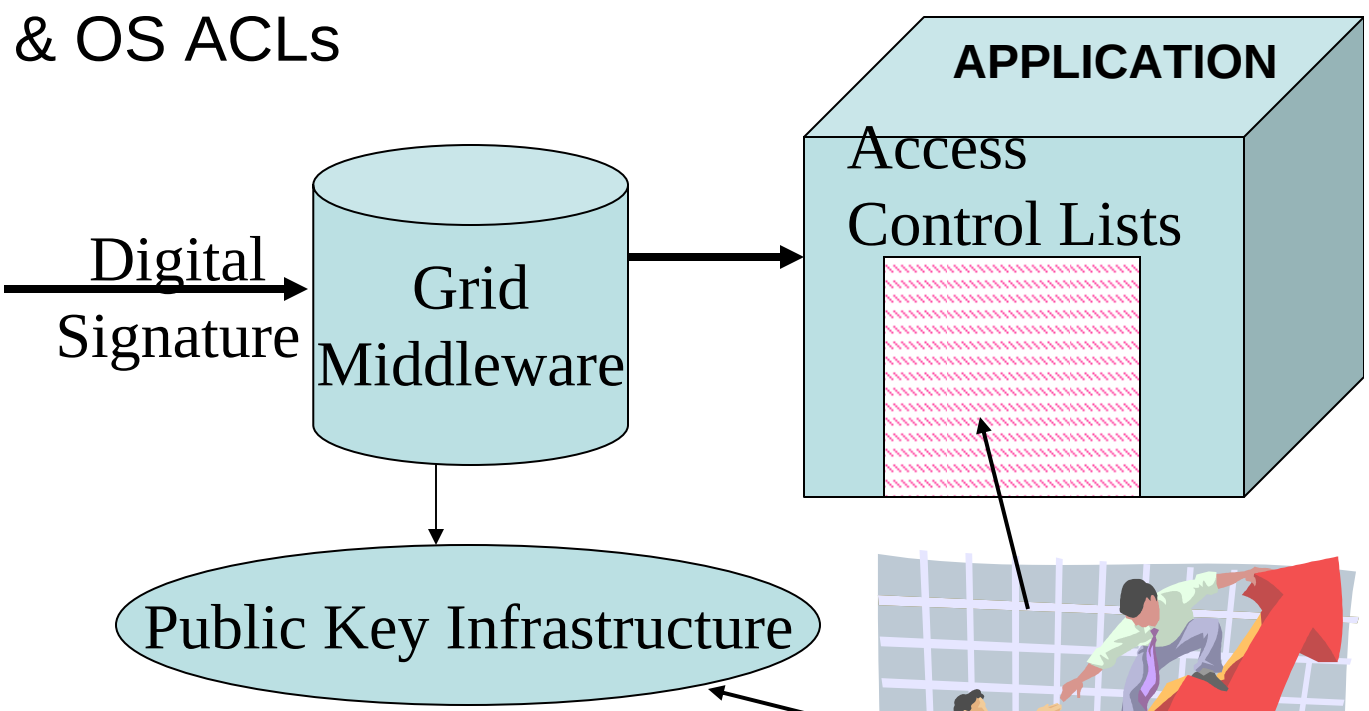


Enter PKI

- Authentication is External to the Application
- But Authorisation is still mainly internal
  - Grid Mapfiles & OS ACLs

One password or pin to access private key

Happy Users!



Multiple Administrators

High cost of administration

No overall Security Policy

Enter PMI

- Authentication and Authorisation are External to the Application

Digital Signature

Grid Middleware

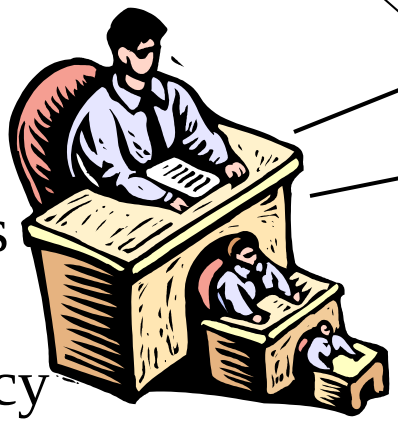
Application

Public Key Infrastructure

Privilege Management Infrastructure

One password or pin to access private key

Happy Users!



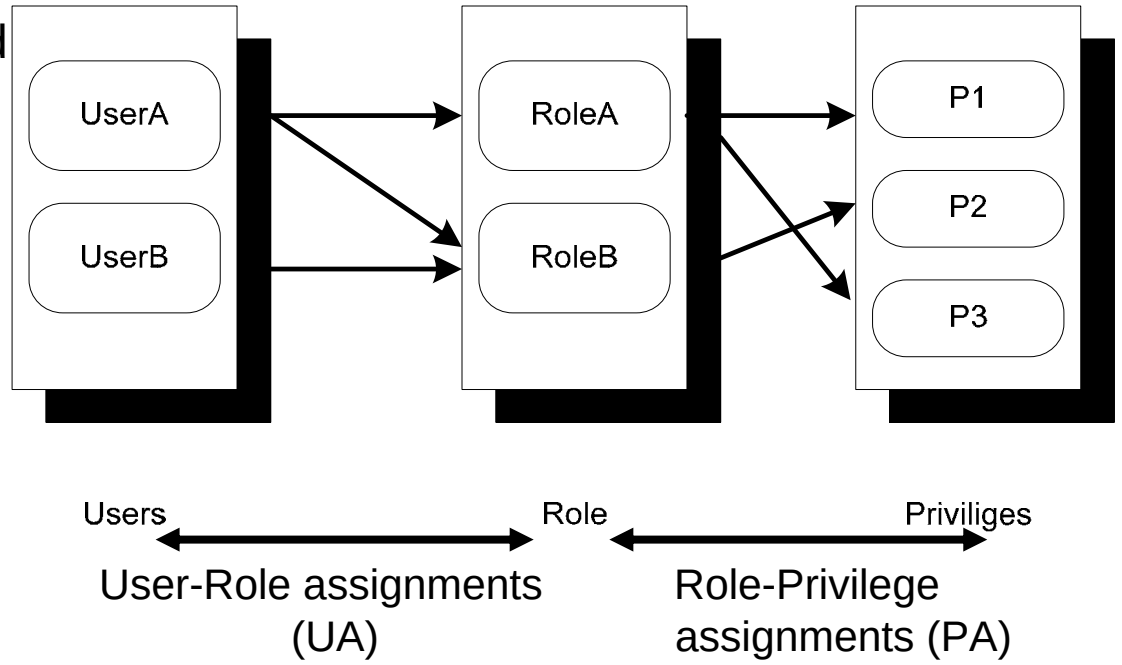
Fewer Administrators

Lower cost of admin

Overall Security Policy

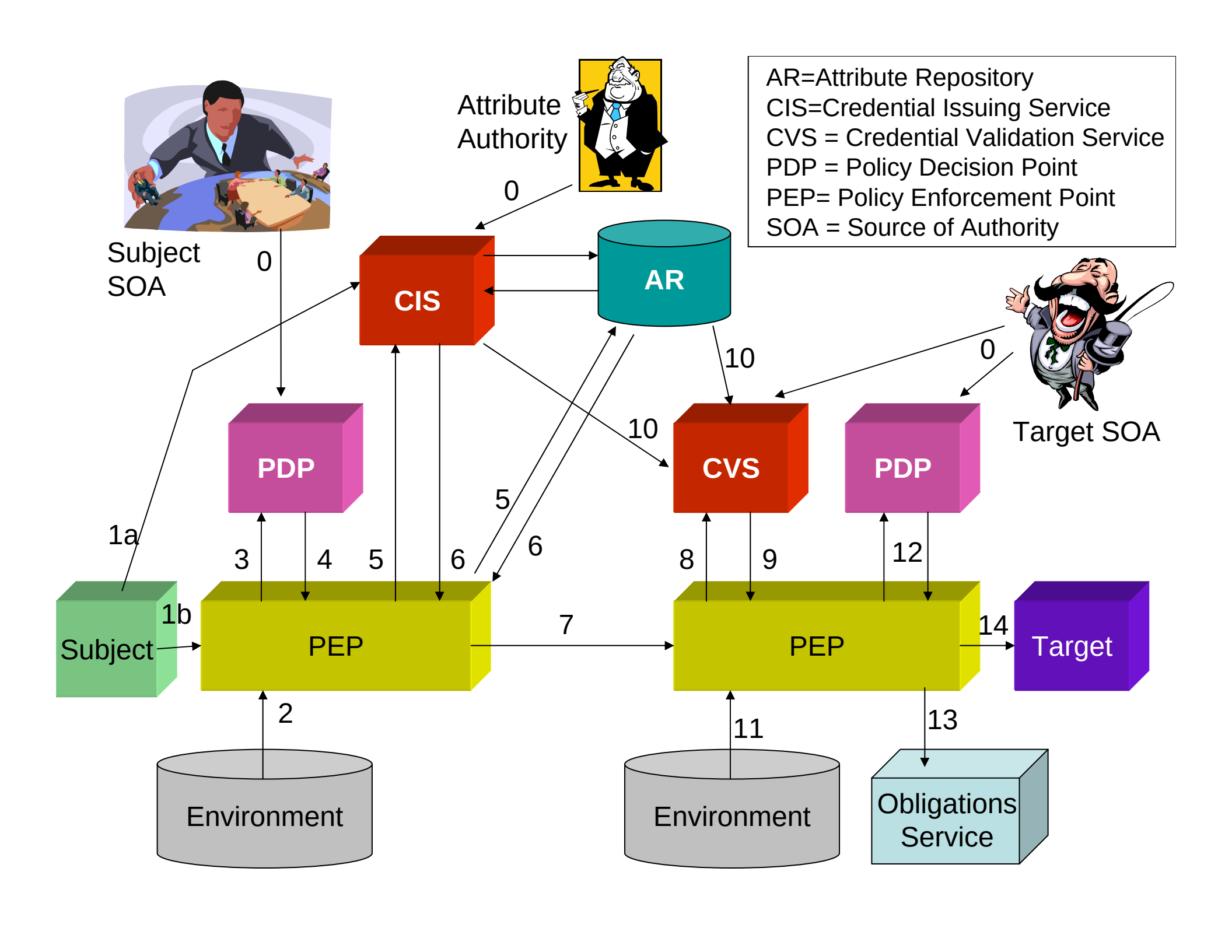
Role/Attribute Based Access Control Model

- Hierarchical Role based Access Control (RBAC)
  - Permissions are allocated to roles/attributes
  - Superior roles/attributes inherit privileges of subordinate roles/attributes
  - Users are assigned role memberships
  - Role members acquire roles' permissions
- Benefits
  - Security
    Remove a user's roles and all privileges are gone
  - Manageability
    Users change more frequently than roles
  - Scalability
    No of roles usually much less than no of users



In Grids we separate UA from PA

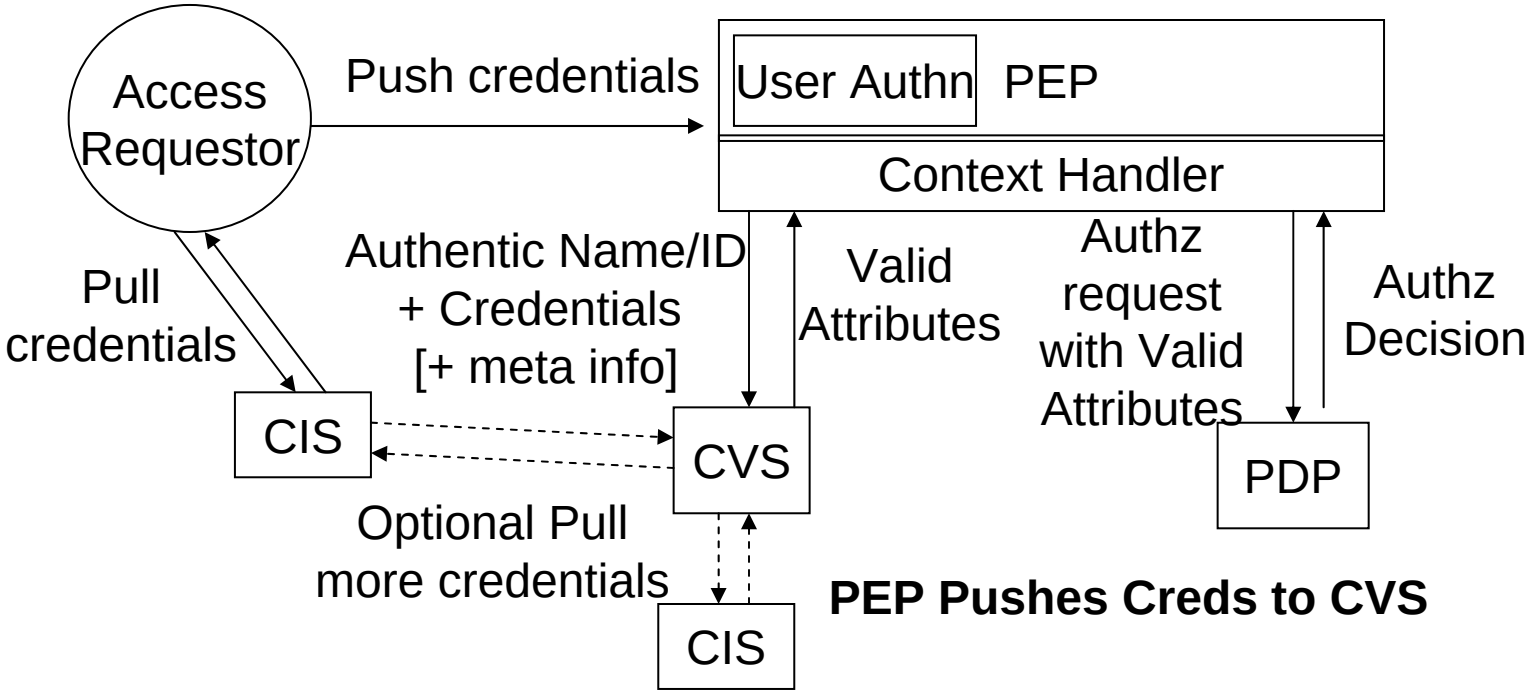
- In traditional RBAC systems both UA and PA are under the control of a single central authority
- In Grids we can no longer assume this is the case
- UA is performed by VO managers and other Attribute Authorities e.g. Government, Health Authorities etc.
- PA is performed by the resource owner (always) but some permissions may be delegated to VO managers
- Leads to the following model


AR=Attribute Repository
CIS=Credential Issuing Service
CVS = Credential Validation Service
PDP = Policy Decision Point
PEP= Policy Enforcement Point
SOA = Source of Authority
Attribute Authority
Subject SOA
Target SOA
CIS
AR
CVS
PDP
PDP
PEP
PEP
Subject
Target
Environment
Environment
Obligations Service
0
0
0
1a
1b
2
3
4
5
5
6
6
7
8
9
10
10
11
12
13
14

OGF OGSA Authz WG

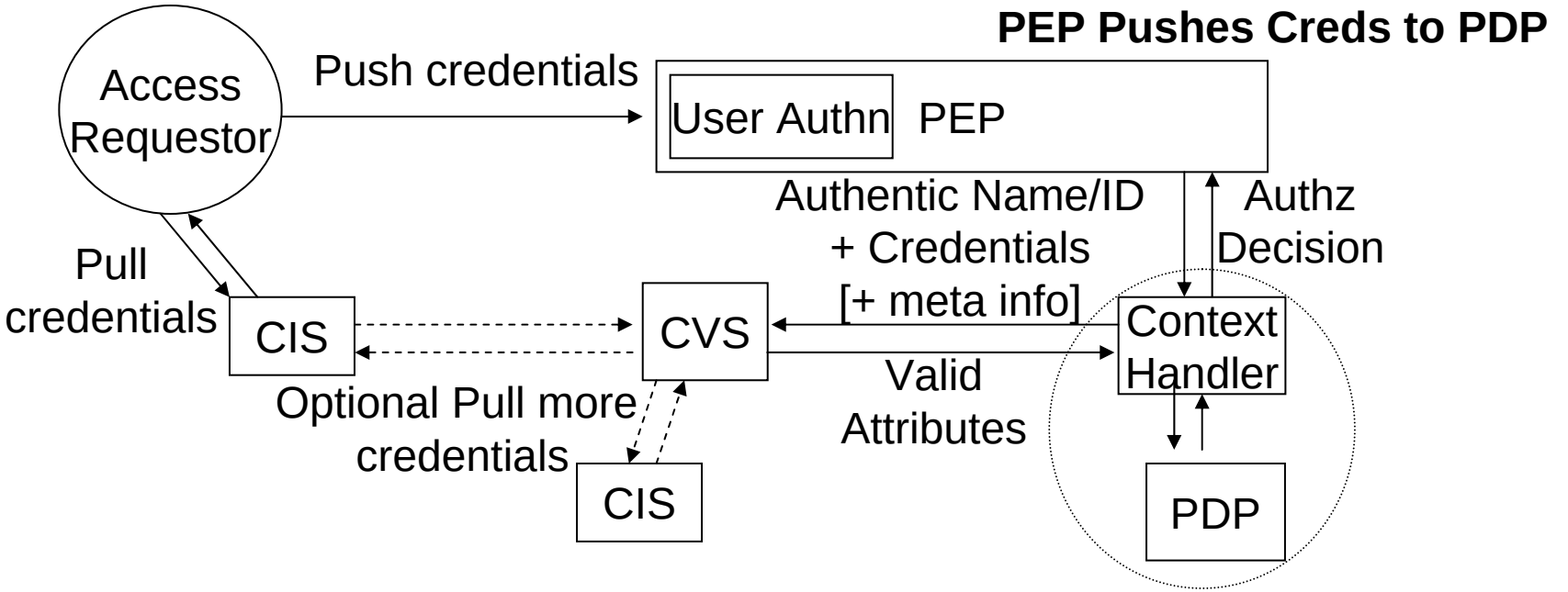
- Has worked at standardising the protocols for the various entities to communicate with each other
- Currently have protocols for
  - PEP to PDP
  - CVS or PEP or User to CIS
  - PEP or PDP to CVS
- Currently don’t have protocols for
  - PEP to Obligations Service
  - CVS or PEP to Attribute Repositories because short term freshly minted authz credentials are preferred

User Pushes Authz Credentials to Grid

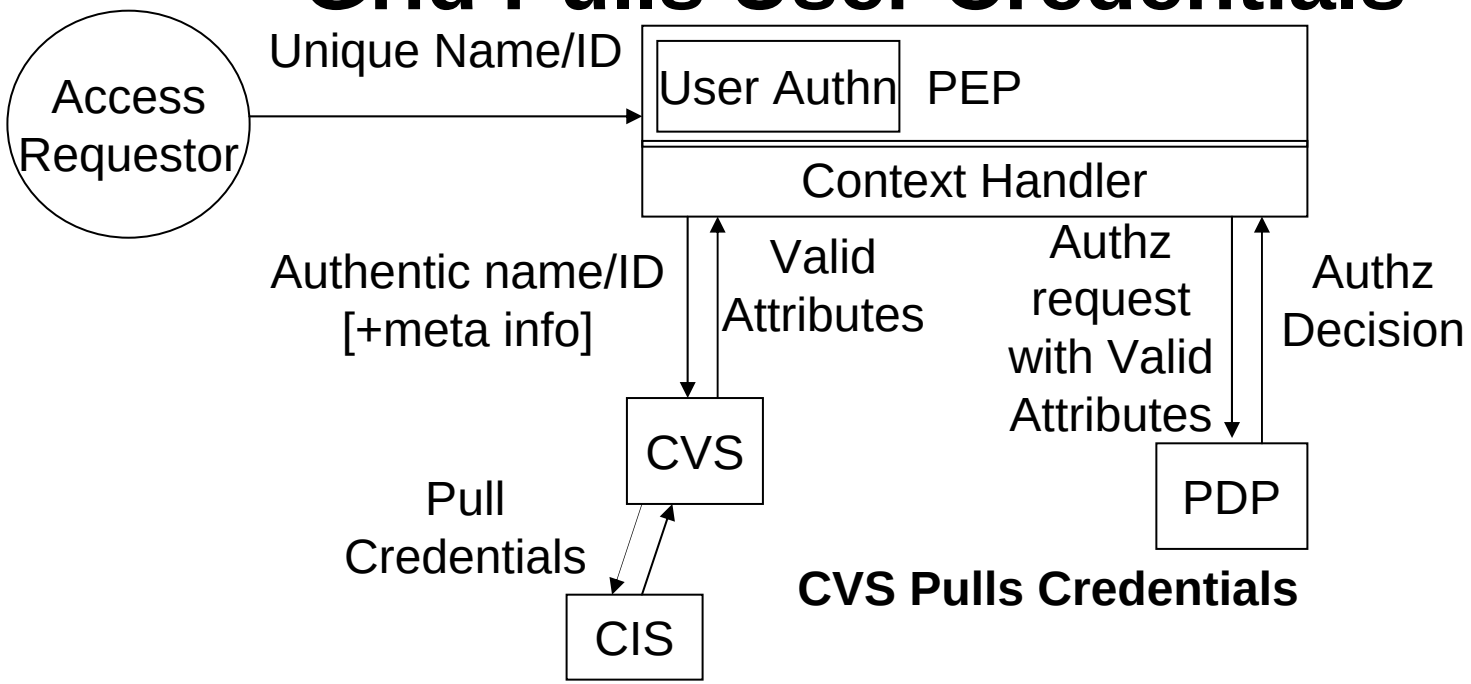


PEP Pushes Creds to CVS

PEP Pushes Creds to PDP

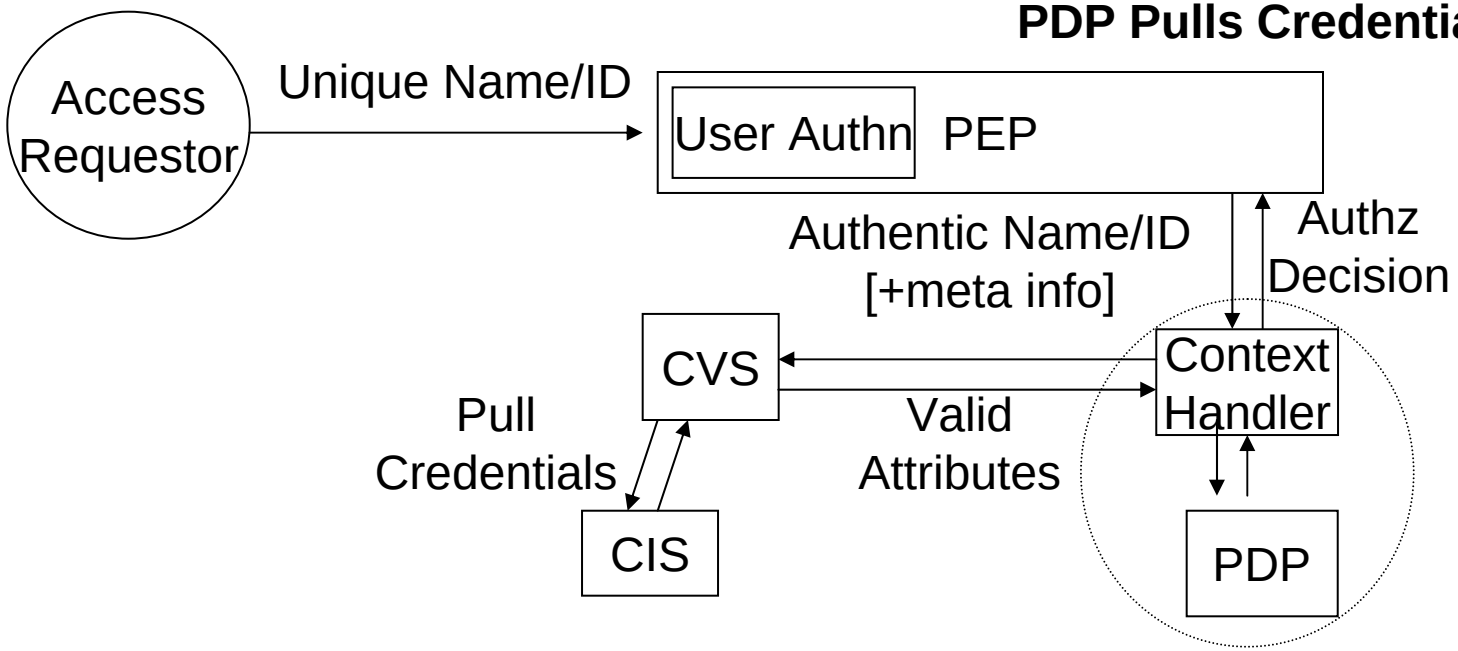


Grid Pulls User Credentials



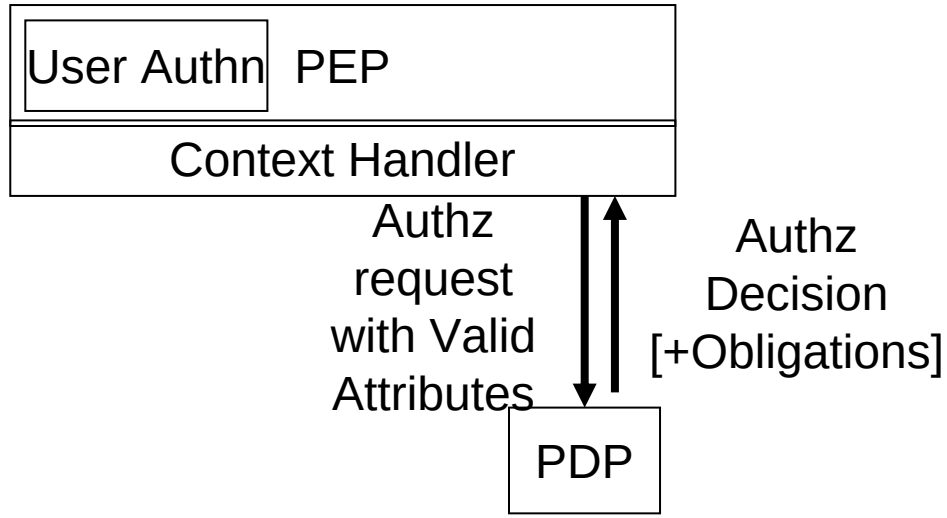
CVS Pulls Credentials

PDP Pulls Credentials

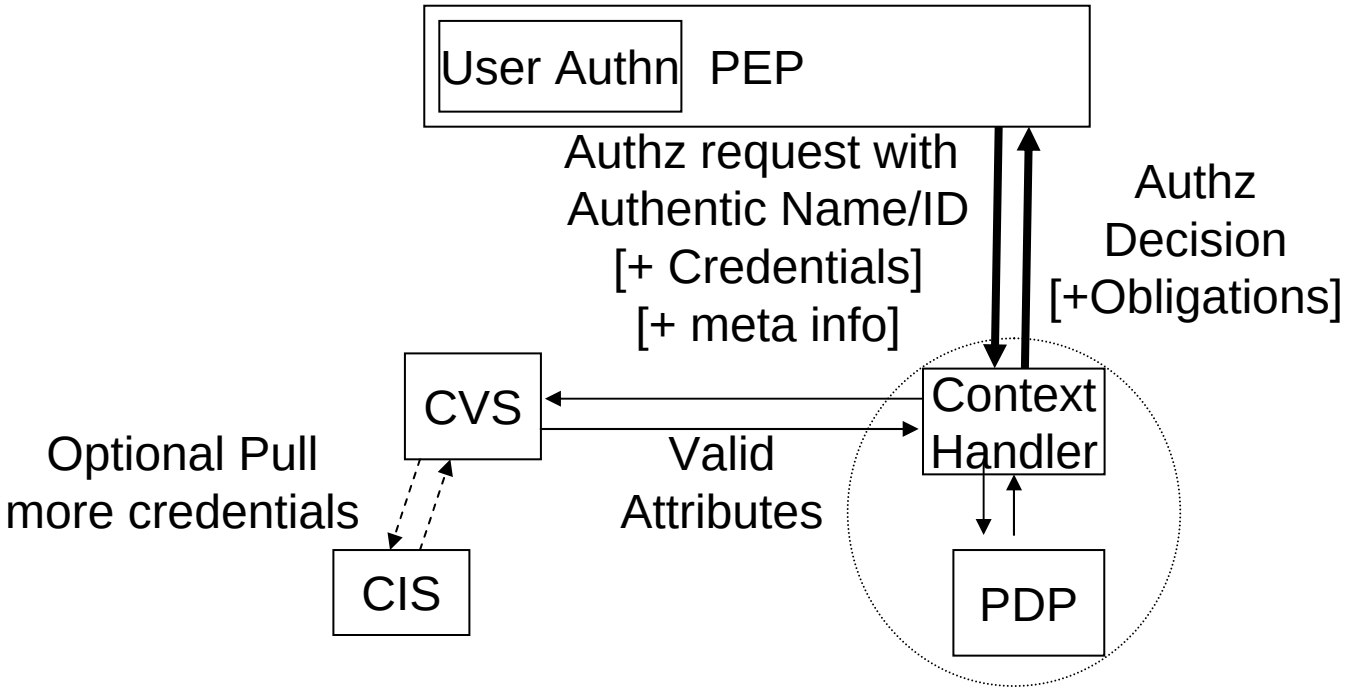


PEP-PDP Protocol

- The XACML Request or Response Context carried in SAML Request or Response message
- Request comprises attributes of the subject, action, resource and environment, or optionally credentials of the subject
- Response comprises the Decision
  - Permit, Deny, Indeterminate or NotApplicable
- plus optional Obligations



PEP Pushes Valid Attributes to PDP

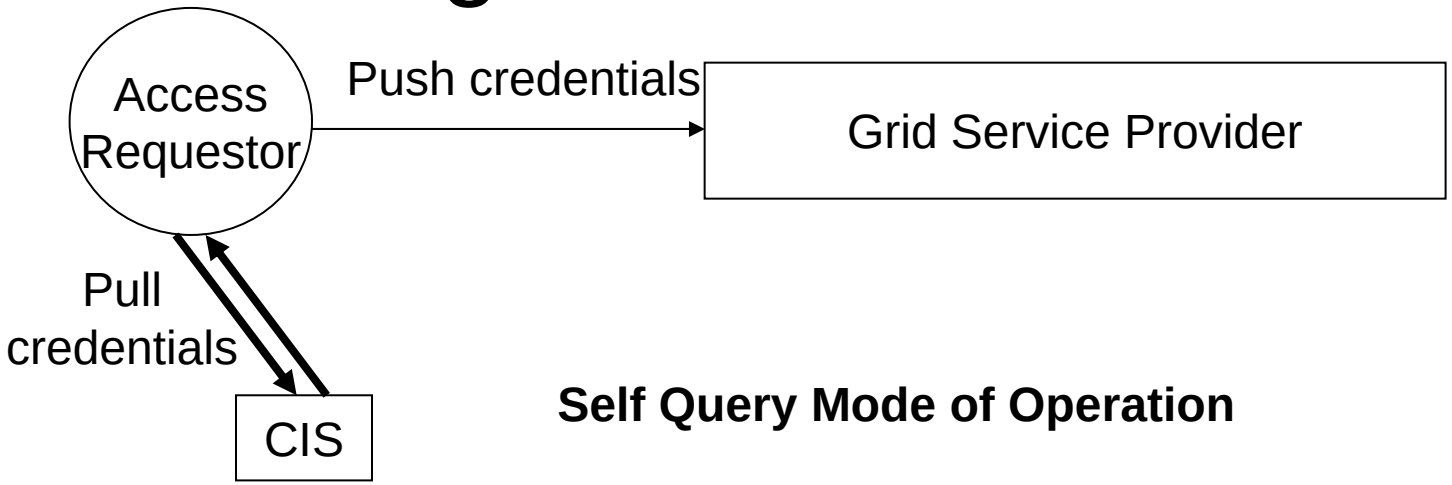


PEP Pushes Credentials/Meta information to PDP

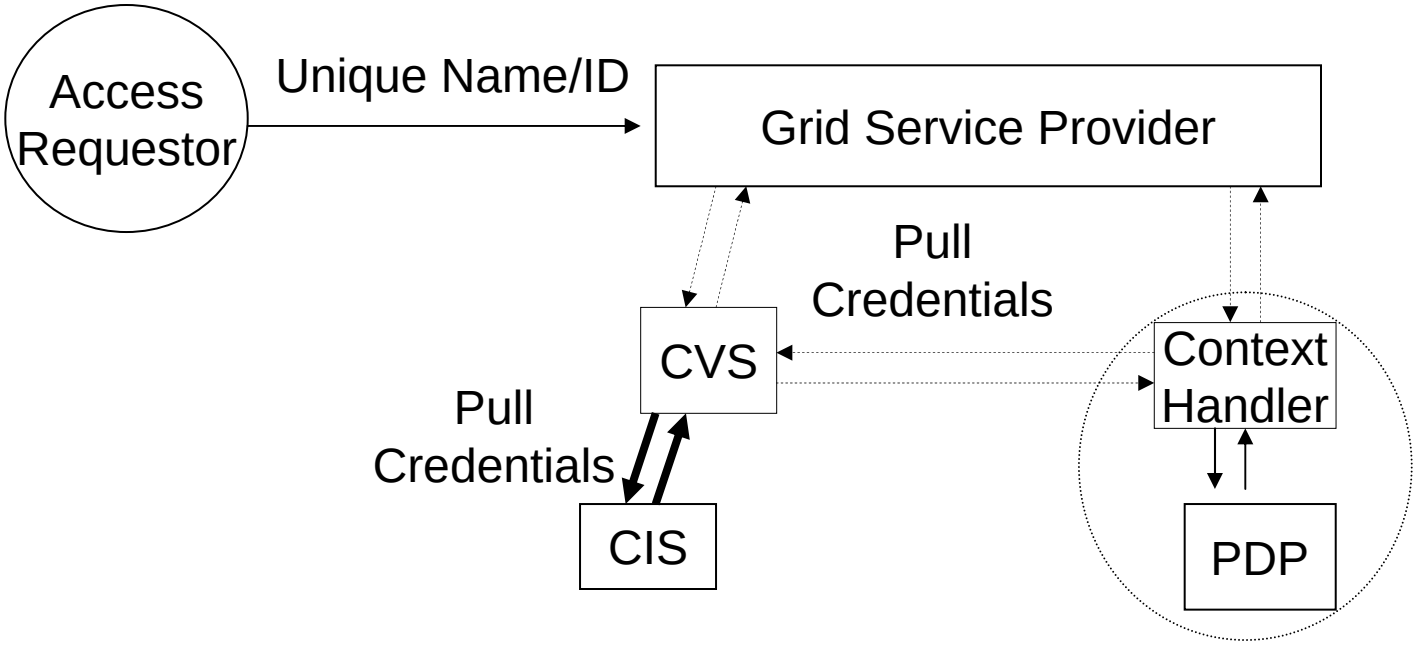
PEP or CVS or User to CIS Protocol

- This is a profile of the SAML Attribute Assertion Query and Request protocol
- Supplemented with an OASIS draft profile for use with X.509 subjects
- Two modes specified: Self Query Mode and Third Party Mode
- Currently a privacy issue with Third Party Mode
  - how does the CIS know that the user is currently accessing the Grid when the user's credentials are being asked for?
  - rely on trust between Grid operator and CIS operator

Pulling Credentials from CIS



Self Query Mode of Operation



Third Party Query Mode of Operation

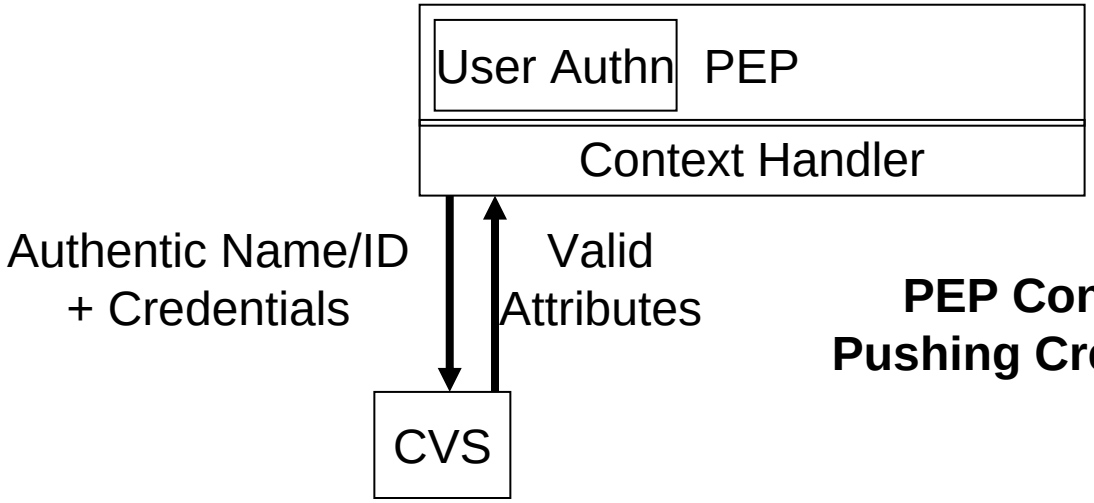
Privacy Issues with 3rd Party Mode

- How does CIS know user has given consent for his/her attributes to be retrieved?
  - Ans. Must have Attribute Release Policy at CIS i.e. a PDP controlling access to the CIS
- How does CIS know that user is currently accessing the Grid when the Grid pulls the user's credentials?
  - Current Answer is TRUST. The CIS must trust the Grid middleware/resource owner to act properly. But some say this is not sufficient. We need delegation of authority from the user to the Grid middleware for it to act on user's behalf at time of access. UNRESOLVED
- N.B. Neither are issues when user pulls credentials him/her self

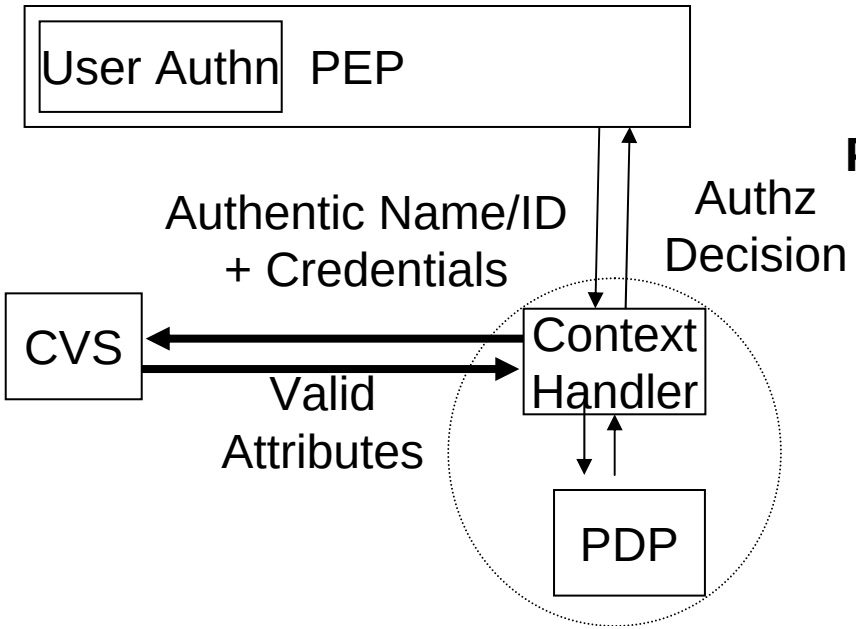
PEP or PDP Context Handler to CVS Protocol

- Uses SAML Attribute Assertions carried in WS-TRUST protocol
- Can work in either push or pull modes
- Push Mode. CH asks CVS to validate credentials (optionally signed SAML assertions carrying LDAP attributes) and return SAML assertions carrying validated XACML formatted attributes, so that latter can be given to PDP
- Pull Mode. CH asks CVS to pull credentials (SAML assertion contains subject DN, optional Authn statement and SubjectAttributeReferenceAdvice) and return SAML assertions carrying validated XACML formatted attributes so that latter can be given to PDP

Push Mode

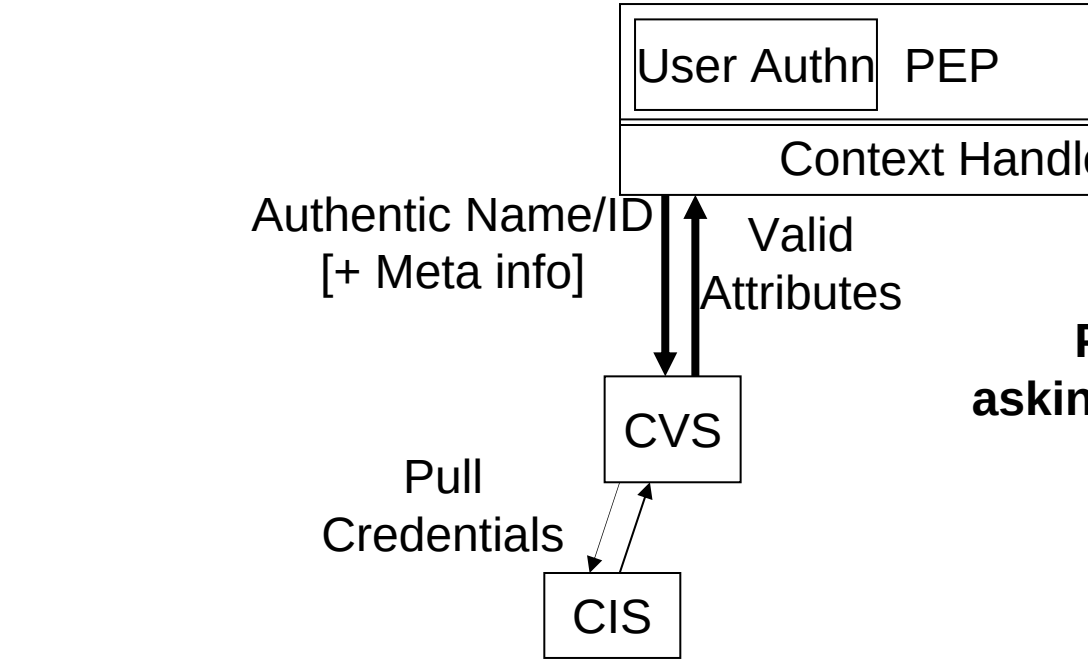


PEP Context Handler
Pushing Credentials to CVS

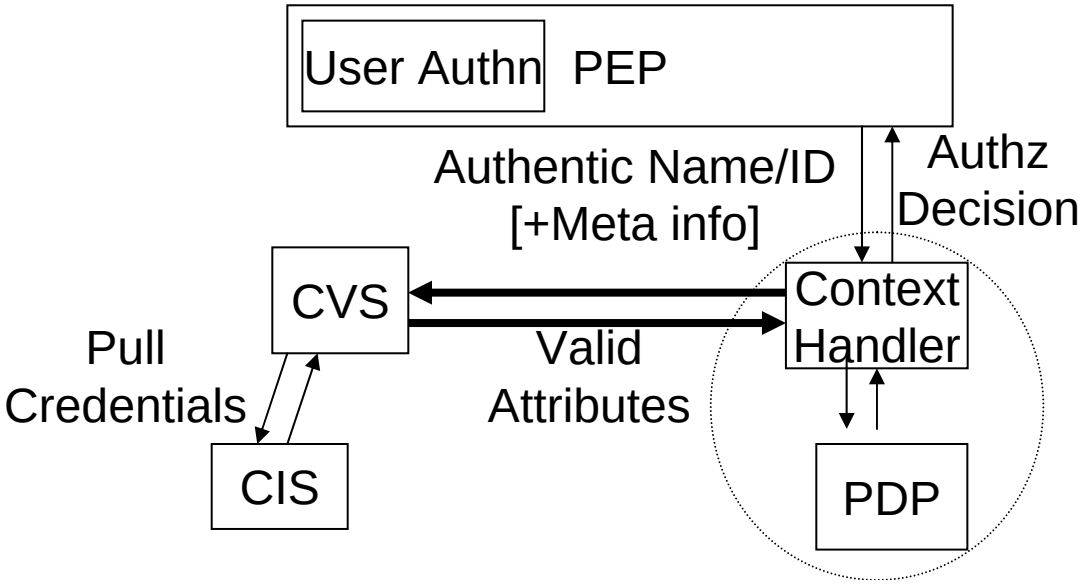


PDP Context Handler
Pushing Credentials to CVS

Pull Mode



PEP Context Handler asking CVS to Pull Credentials

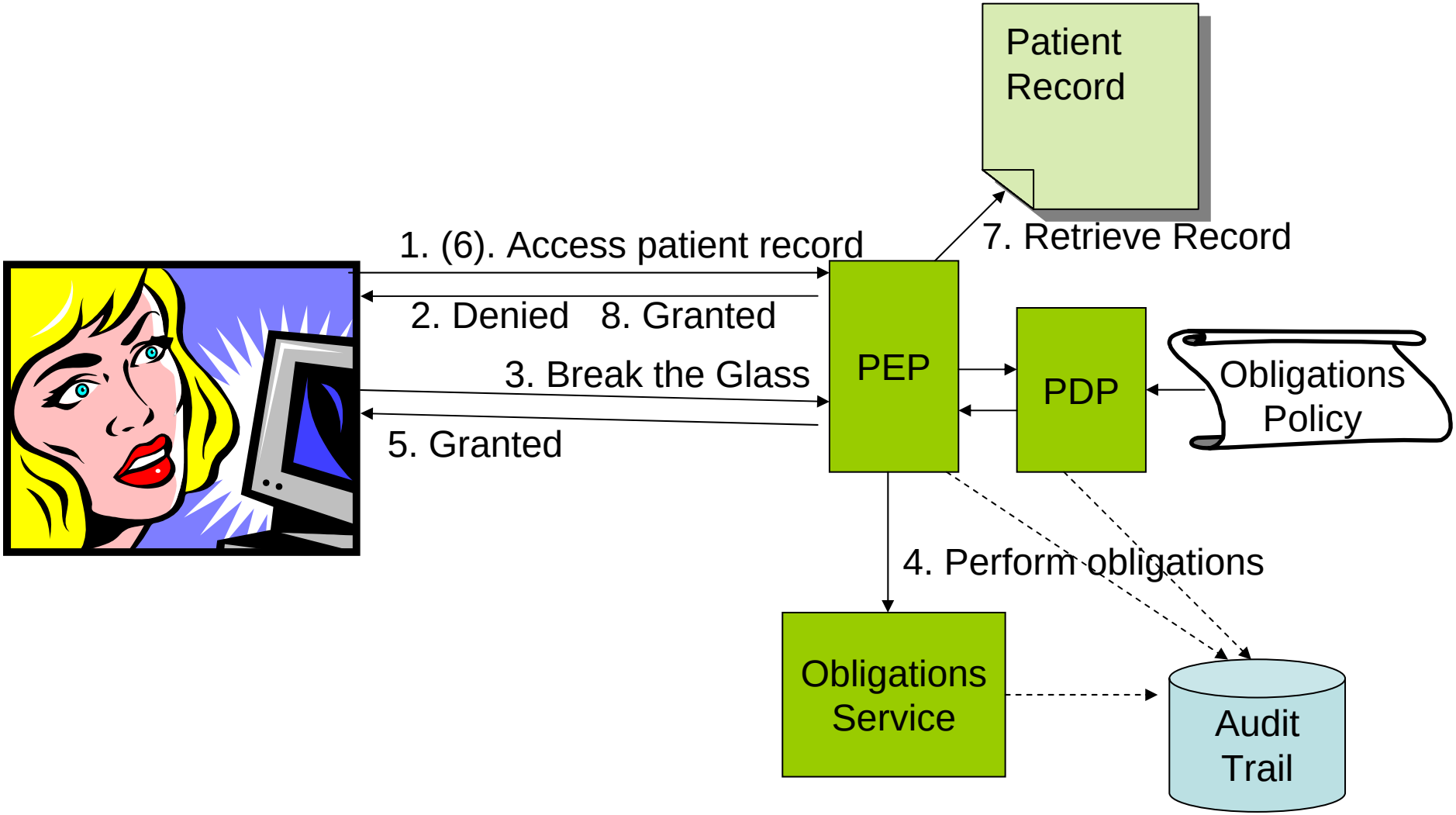


PDP Context Handler asking CVS to Pull Credentials

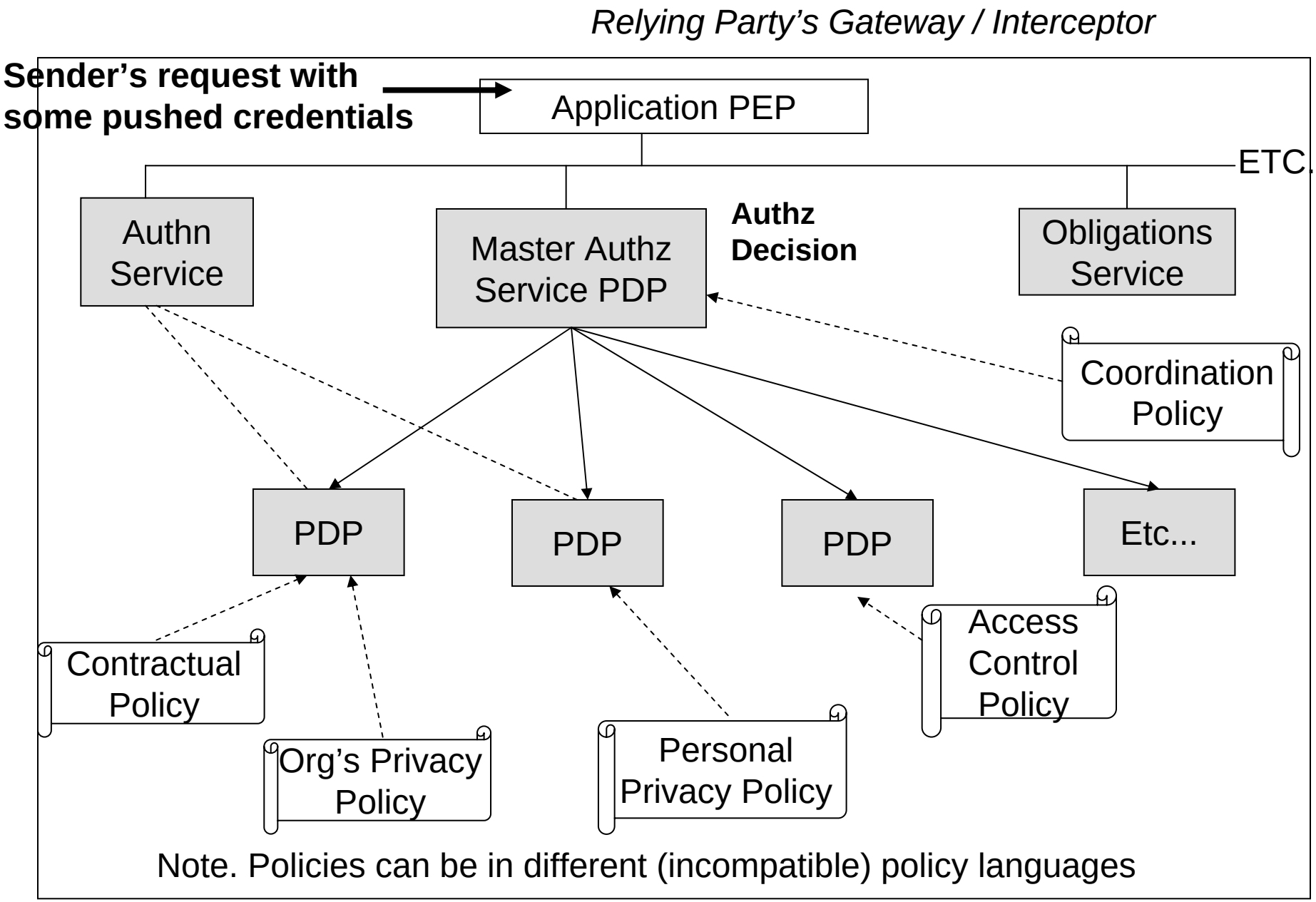
What Standardisation Work is Still To Do?

- Standardising the attributes that are passed
- Standardising the obligations that are returned
- Standardising a protocol for talking to an Obligations Service
- Break the Glass Policies
- Support for Multiple PDPs executing different policies from different authorities (e.g. user policy, resource owner policy, VO policy, government policy etc.) and the policy combining rules
- Coordinated Decision Making
- Retrieving attributes from multiple sources

Break the Glass Policies



Support for Multiple PDPs



Coordinated Decision Making

- Motivation/Problem Statement
- Sometimes one access control decision depends upon prior decisions
  - E.g. You can only draw £250 from ATM machines in a day
  - E.g. You are only entitled to use 5GB memory per grid job
- Decision may depend upon previous decisions at the same or different resources in the distributed system
- Relatively easy to solve if only one PDP is involved
  - Have a stateful PDP
  - Use existing PEP – PDP protocol from all nodes in Grid

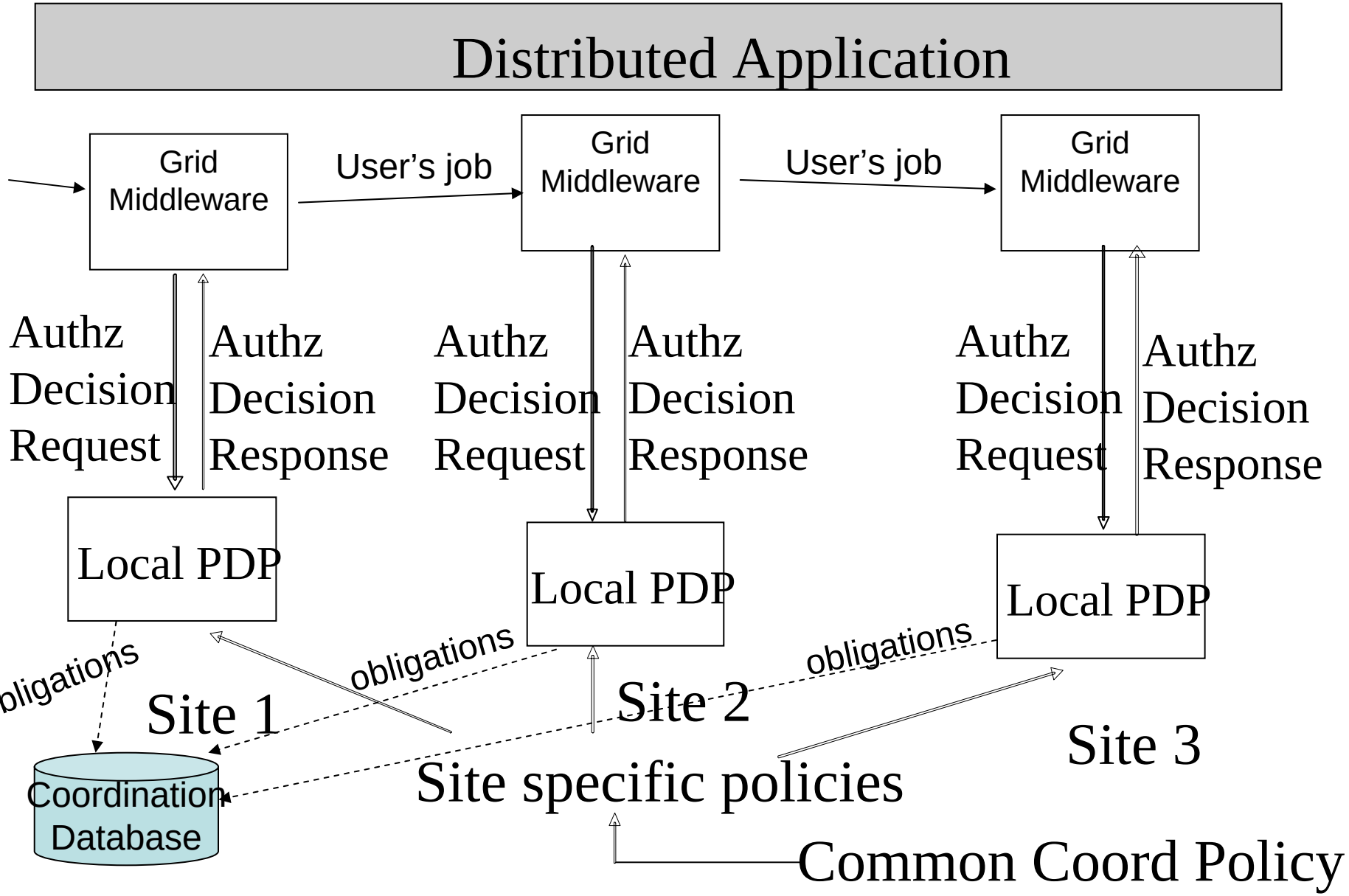
Disadvantages of one PDP Solution

- Requires a stateful PDP, but most PDPs (e.g. XACML) today are stateless
- In many VOs and grids there are multiple PDPs each with own policies
- Who would define policy for the entire Grid?
- Conclusion. We need to share state information between all the PDPs in the Grid, whilst using stateless PDPs! A dilemma

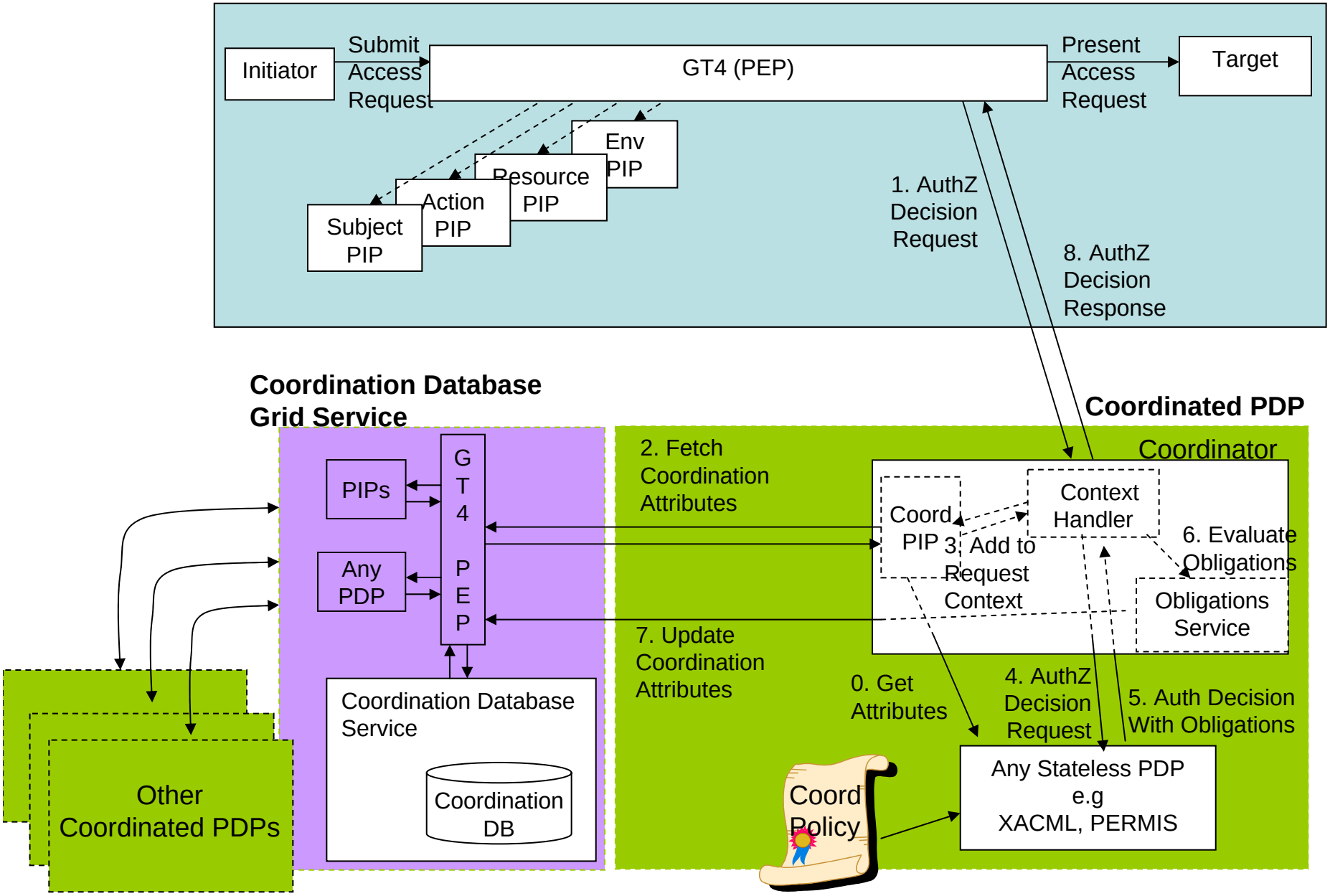
Conceptual Solution in Brief

- Store state information in *coordination* attributes of a *coordination* object
- Introduce a coordination policy for the distributed application (which each site can include as part of its access control policy)
  - Access control decisions will then depend upon values of these coordination attributes [as well as subject, resource, action and environmental attributes]
  - Obligations are used to update these coordination attributes
- Implement coordination object and attributes in a database grid service (DB provides stable storage, fast lookup, distribution, replication etc.)

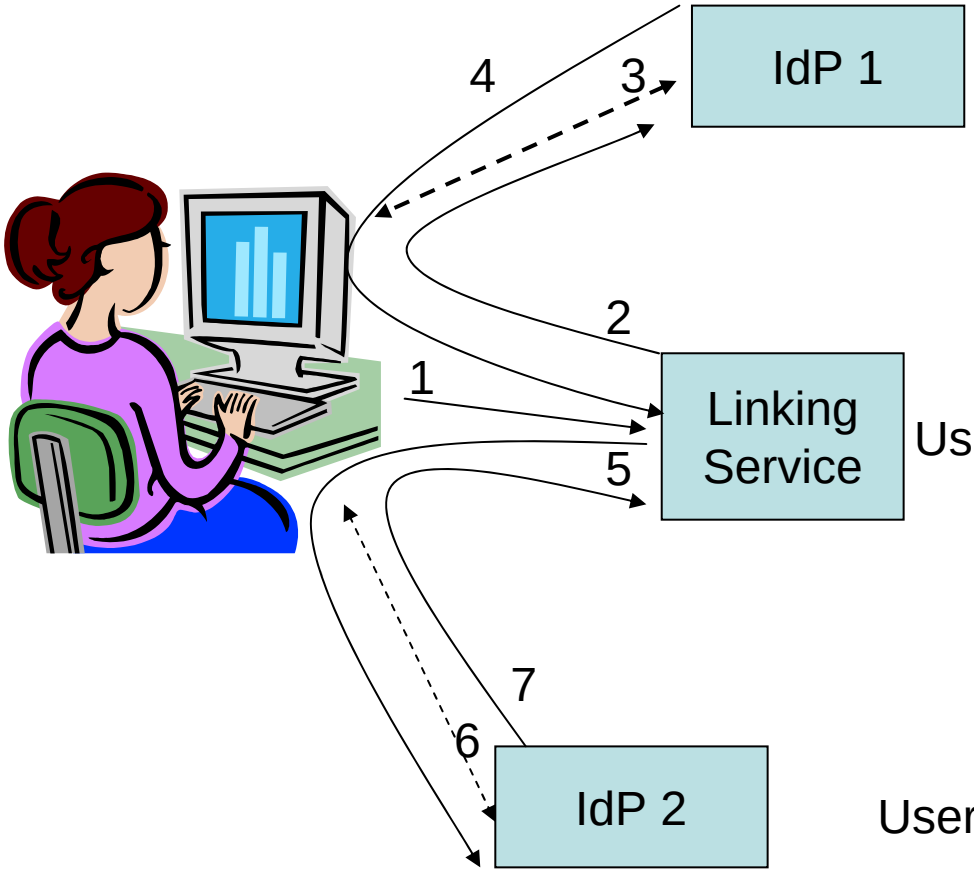
Conceptual Solution



Implementation in GT4



Linking Service

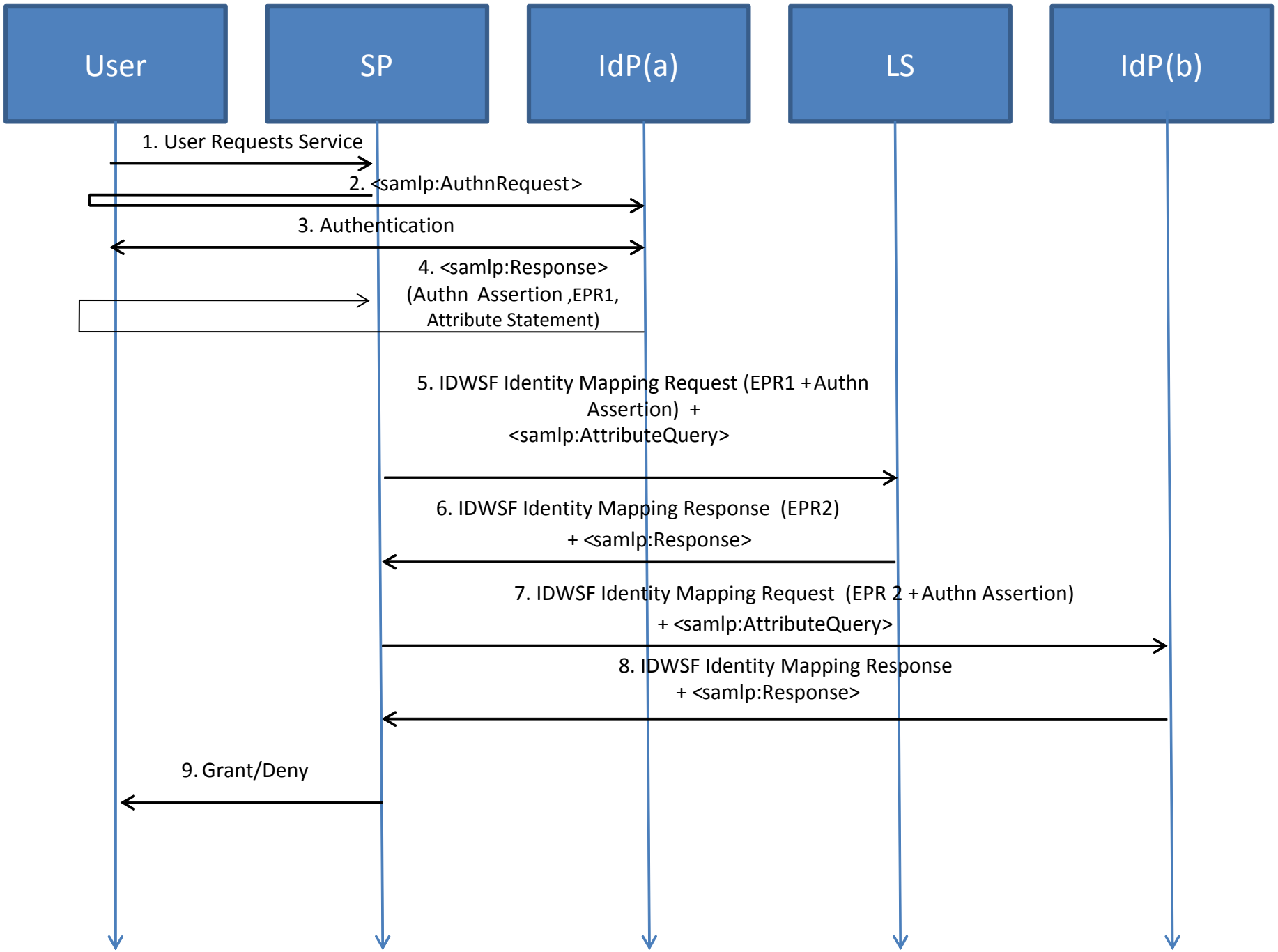


Storage Requirements

UserX, Attr1, RegLoA, PID 1:LS

UserZ, IdP1:PID 1:LLoA1,
IdP2:PID2:LLoA2

UserA, Attr2, RegLoA, PID 2:LS



IdP Direct SP aggregation with IDWSF Id Mapping

Conclusions

- Research and standardisation is pressing ahead in Grids, often far in advance of what users are actually using today.
- Have not yet properly addressed the problems of ease of use – even current “simple” PKI based systems fail here.

- Any Questions?