

Adding Distributed Trust Management to Shibboleth

David Chadwick
University of Kent

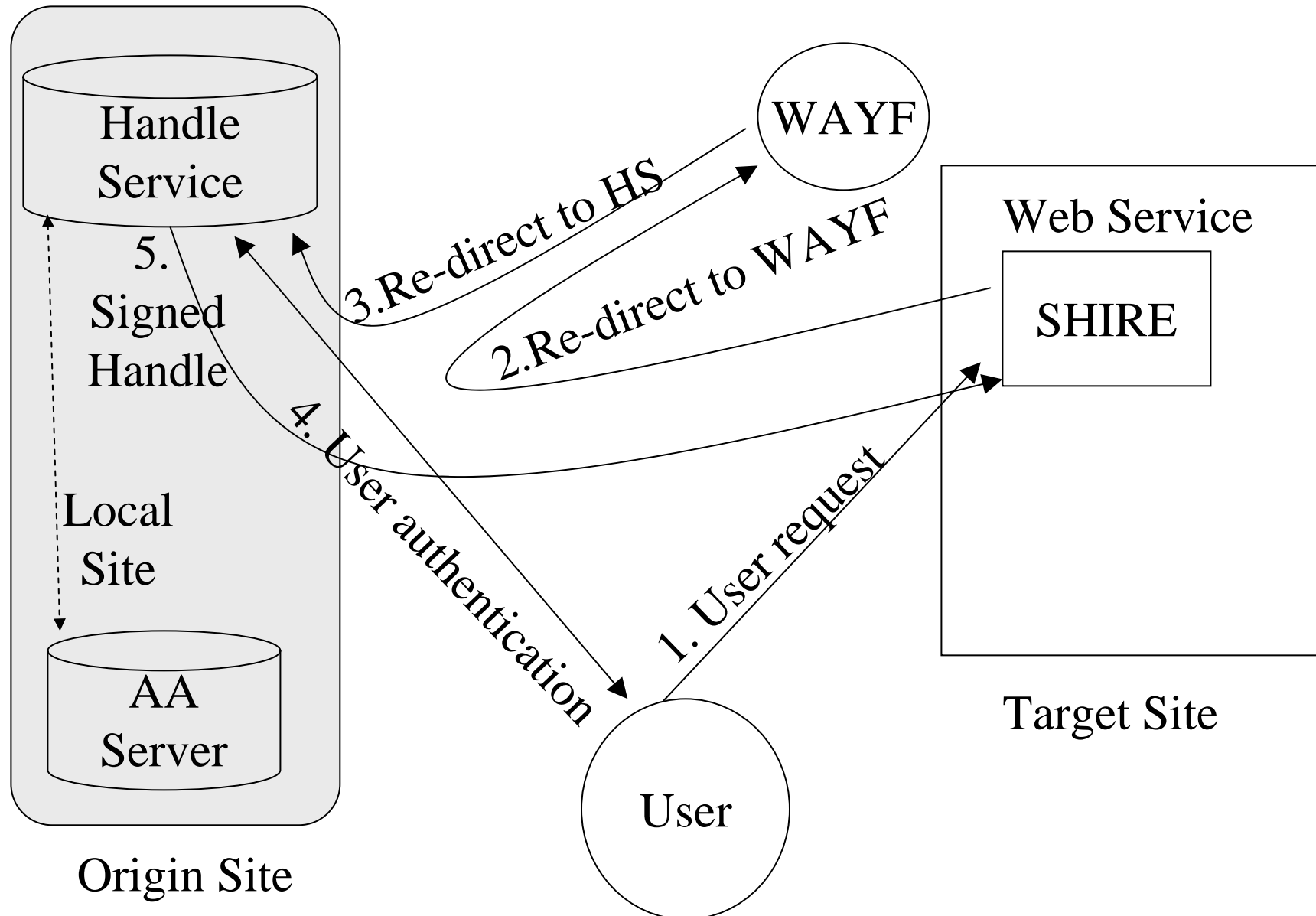
Contents

- Introduction to Shibboleth
- Limitation of Shibboleth
- Features of an Enhanced Trust Model
- Introduction to PERMIS X.509 PMI
- Combining Shibboleth and PERMIS together
- Privacy protection in X.509 PMIs/PERMIS
- Revocation and Performance Issues

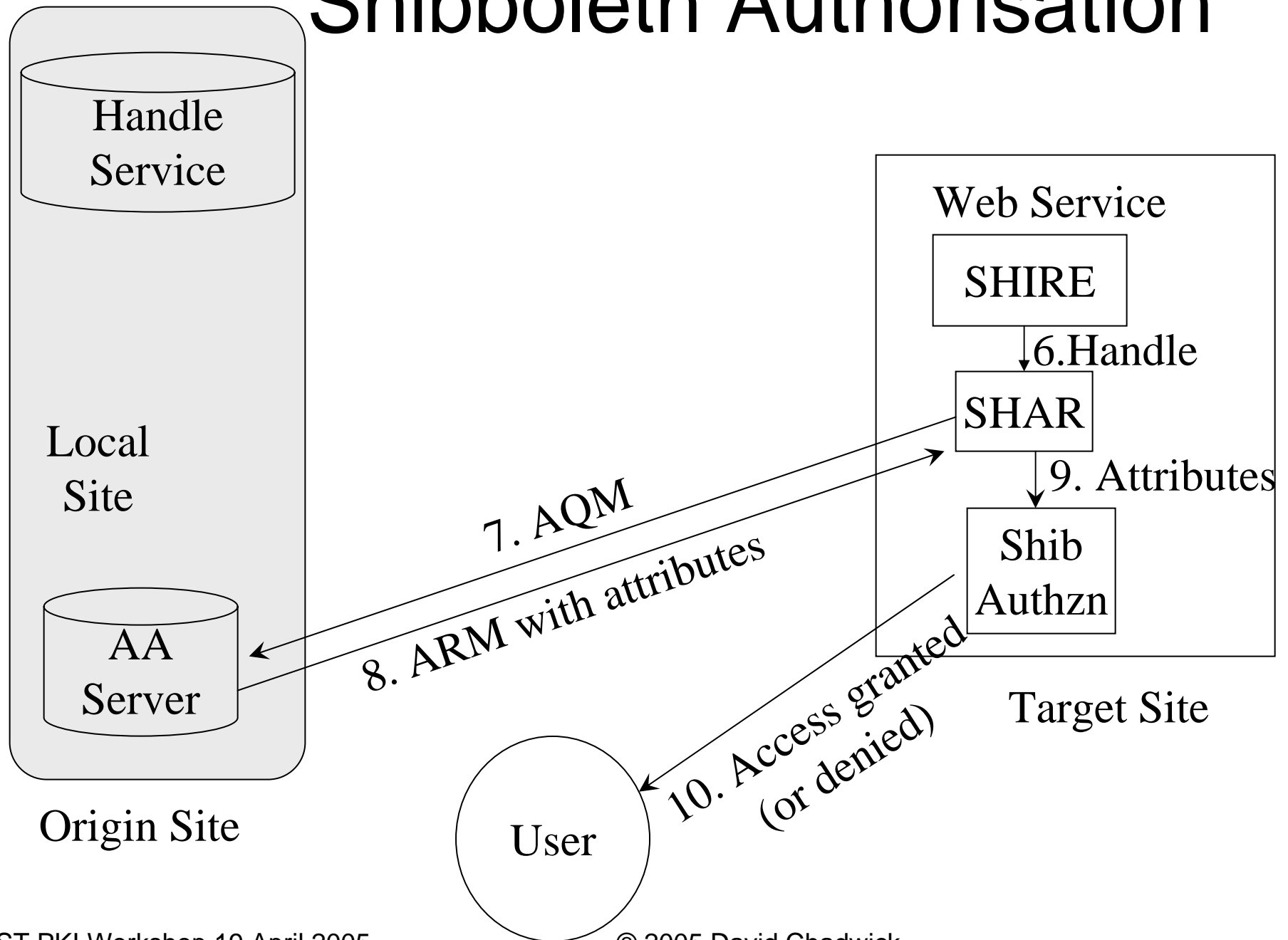
What is Shibboleth?

- Web based authentication and authorisation system that enables single sign on
- User always authenticates to his home (origin) site
- User's attributes are transferred to target site to provide authorisation
- User's name does not have to be revealed, so privacy protection
- Uses PKI certificates and digital signatures to authenticate message exchanges between sites

Shibboleth Authentication



Shibboleth Authorisation



Deficiencies in Shibboleth

- Limited trust model. Target site trusts origin site and vice versa. Period
 - Single key pair held by each site
 - No differentiation between authorisation and attribute authorities
 - No support for dynamic delegation of authority
 - Limited authzn decision making capability
 - Implementations often use a single centrally administered LDAP server or other repository as source of both authentication and authorisation data
 - Target site does not know if the presented attributes are the correct ones for a given user, who allocated them, or if they are still valid
- Using an X.509 Privilege Management Infrastructure solves all these problems

An Enhanced Trust Model

- Multiple attribute authorities should be able to issue attributes and the target should be able to choose which ones it trusts
- All the attribute authorities should not need to be resident at the origin site
- The trust infrastructure should support dynamic delegation of authority
- The target site should not have to rely on the security of the origin site's repository

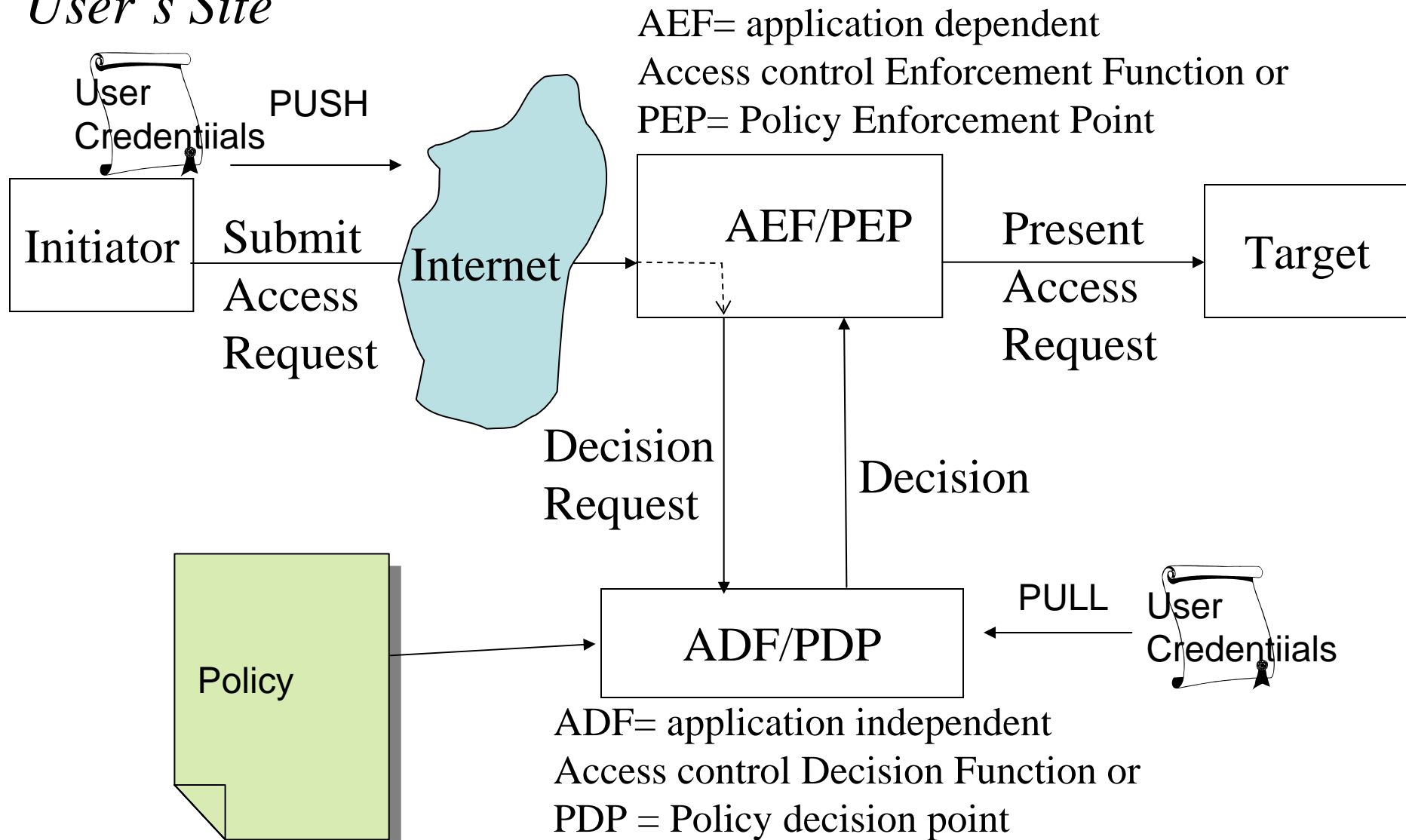
What is PERMIS?

- It is a policy controlled role based authorisation system that uses digitally signed X.509 attribute certificates to hold users roles/attributes – its an X.509 PMI
- It can work with any and every authentication system (Shibboleth, Liberty, Kerberos, PKI, username/PW, etc.)
- Given a username, a target and an action, PERMIS says whether the user is granted or denied access based on the policy for the target domain
- The policy says which roles/attributes users must have to access which targets and under what conditions
- It can work in push or pull mode (user attribute certificates are sent to PERMIS, or PERMIS fetches them itself). Conforms to ISO 10181-3 model

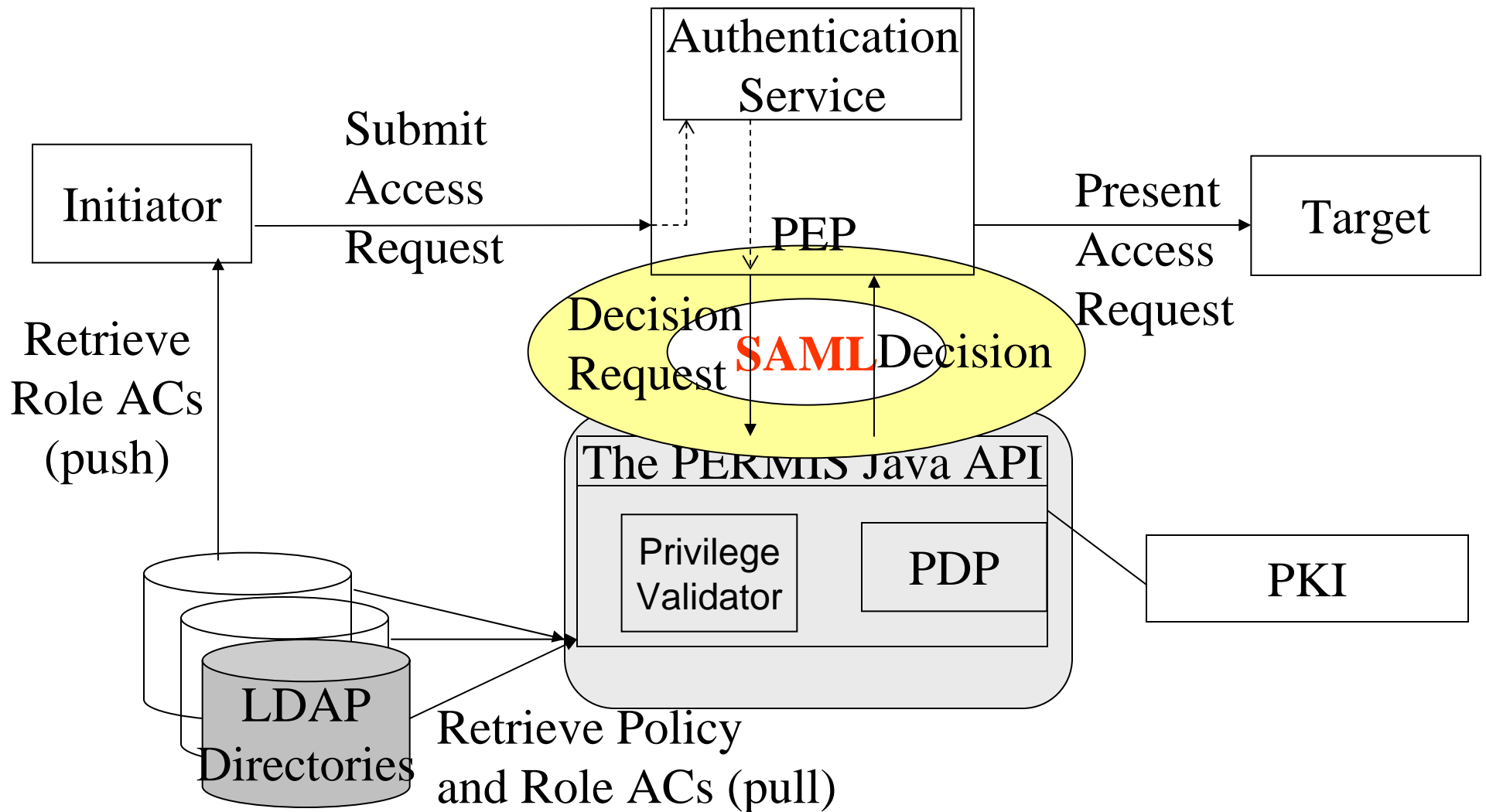
Authorisation Framework (from X.812|ISO 10181-3 and RFC 2753)

Target Site

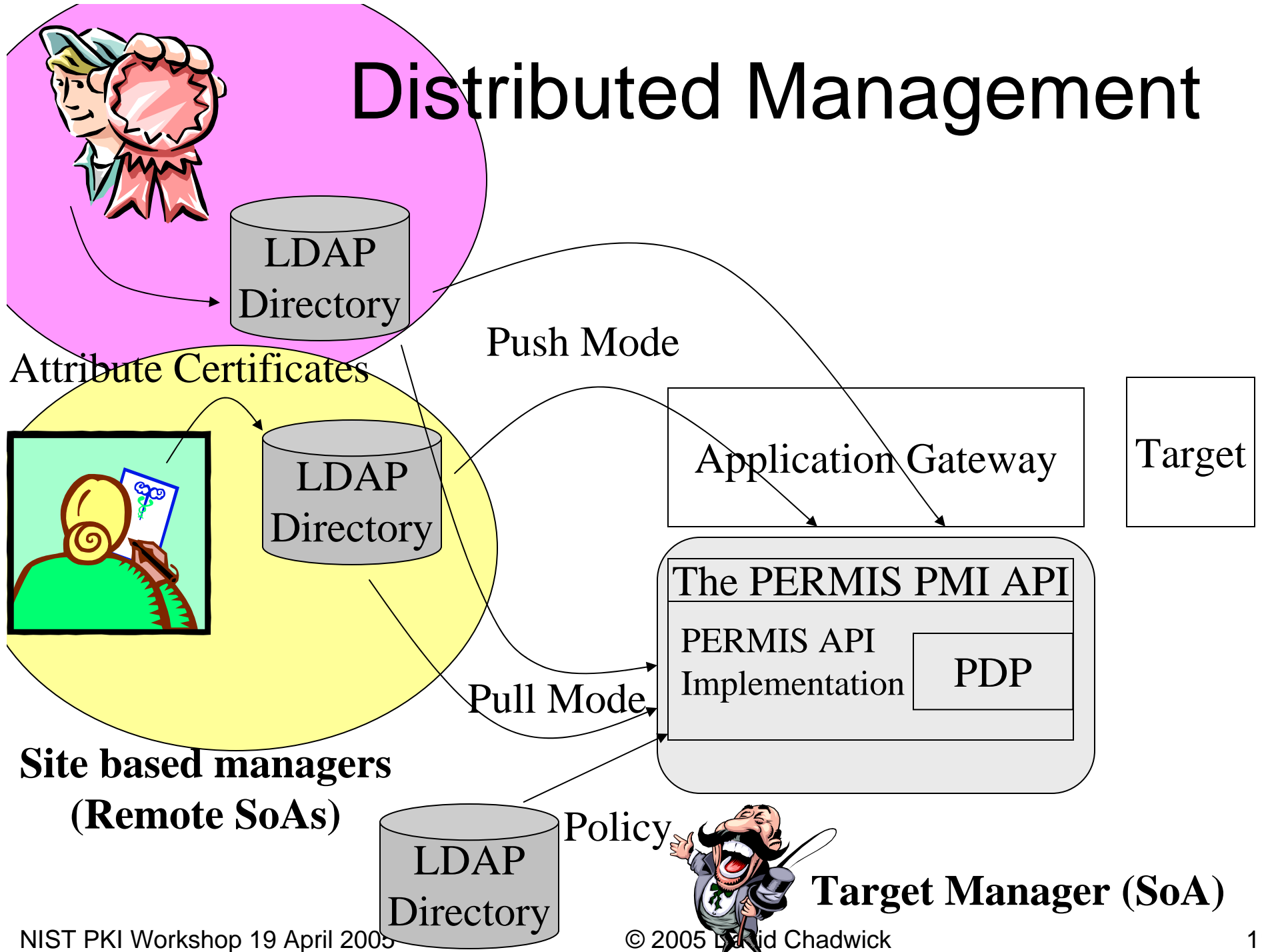
User's Site



PERMIS System



Distributed Management



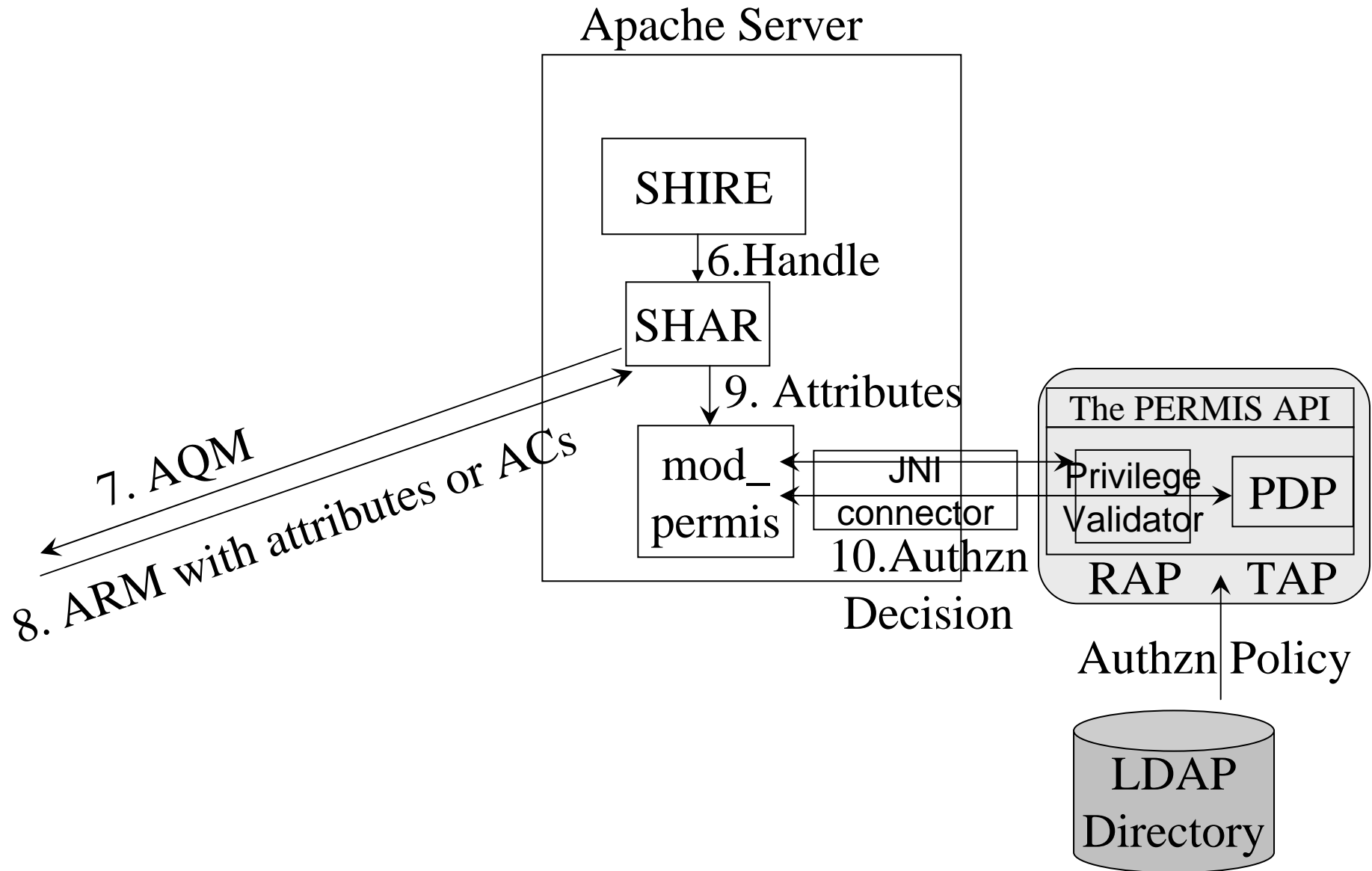
PERMIS Trust Model

- The Target/Resource is the root of trust (Source Of Authority SoA) for access to itself
- The Target is configured with its SoA name at start up
- The Policy is signed by the SoA (Permis checks this)
- The SoA says in the policy which remote SoAs it trusts to allocate roles
- The SoA says what roles they can allocate
- The SoA says what access rights are given to each role
- The remote SoAs authenticate the users and allocate roles to them

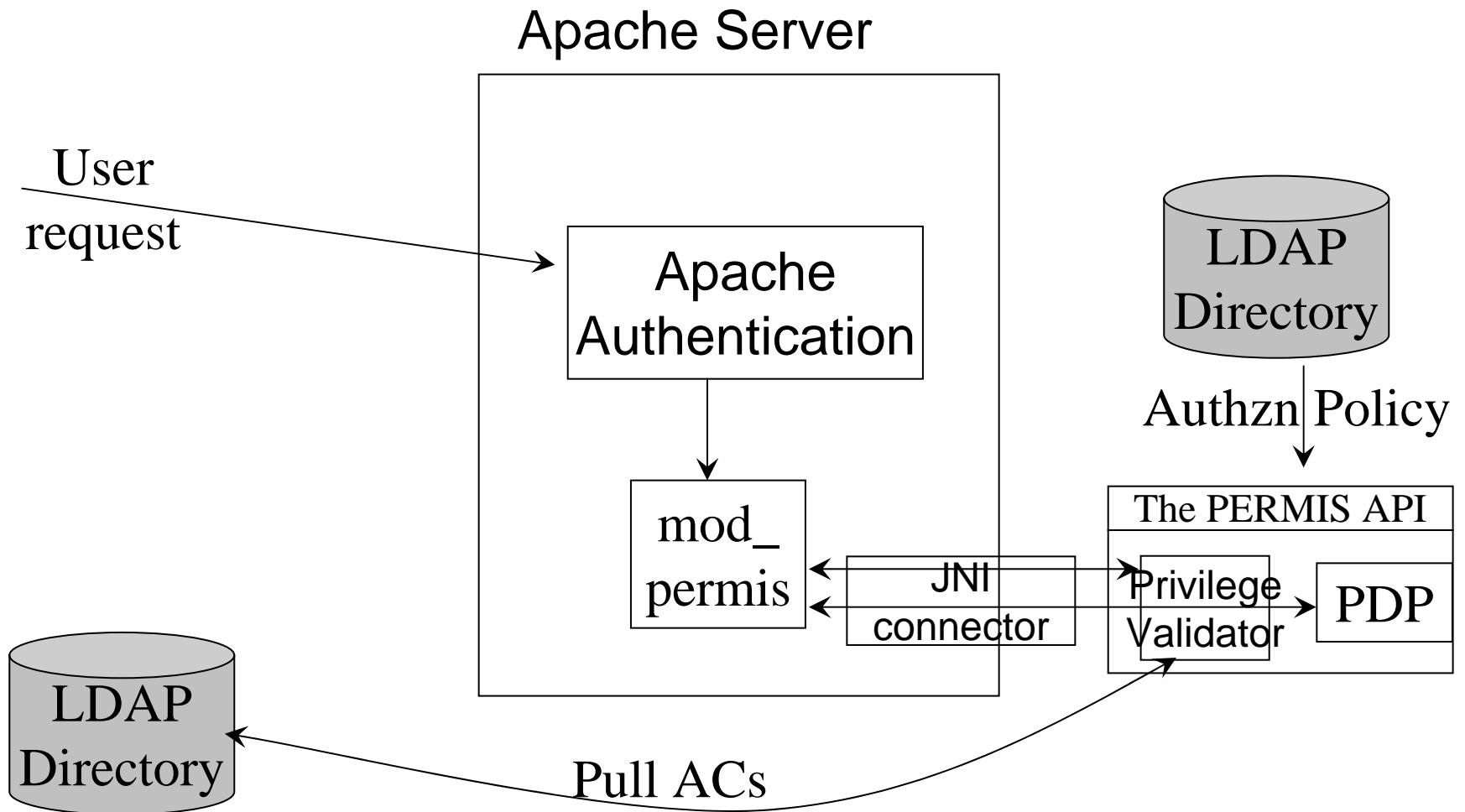
PERMIS Policy

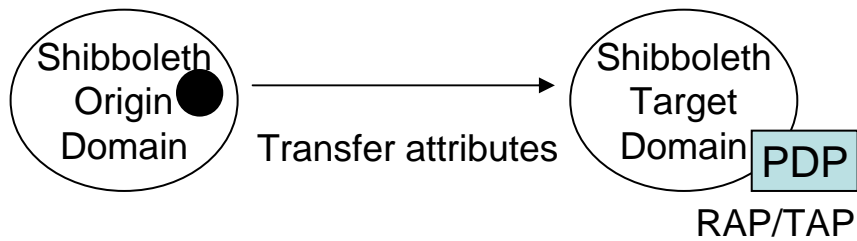
- Each policy has a unique OID, is put in an X.509 AC and digitally signed by the target SoA
- Role Assignment Policy (RAP)
 - Defines the role hierarchy and
 - Says which roles remote SoAs are trusted to assign to which sets of users, and whether dynamic delegation of authority is allowed or not
- Target Access Policy (TAP)
 - Says which roles can access which resources under which conditions
- Coming soon
 - Static and Dynamic Separation of Duties/Mutually Exclusive Roles and support for Dynamic Delegation of Authority in the PDP

Shibboleth with PERMIS Authorisation

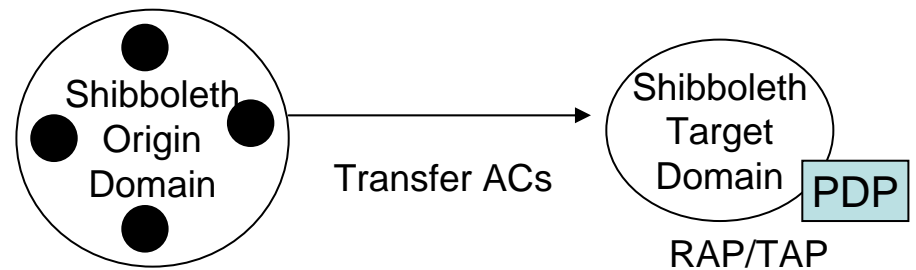


Apache with PERMIS Authorisation

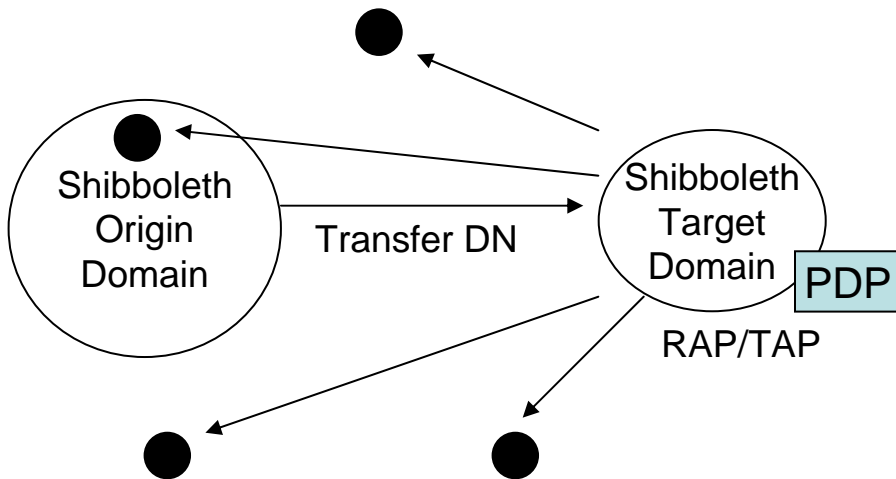




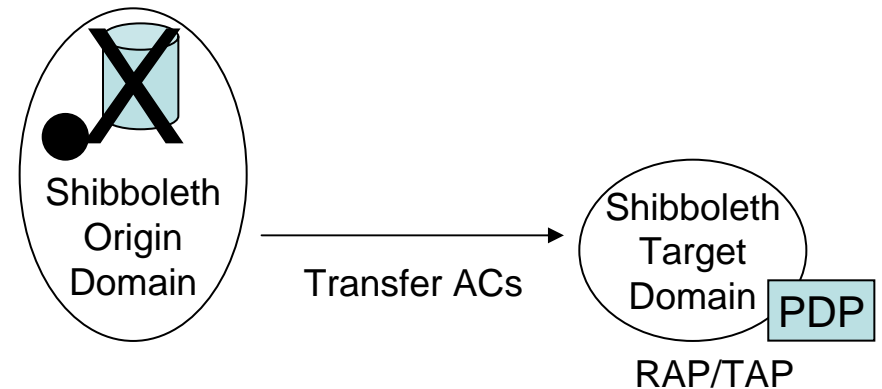
1 Standard Shibboleth



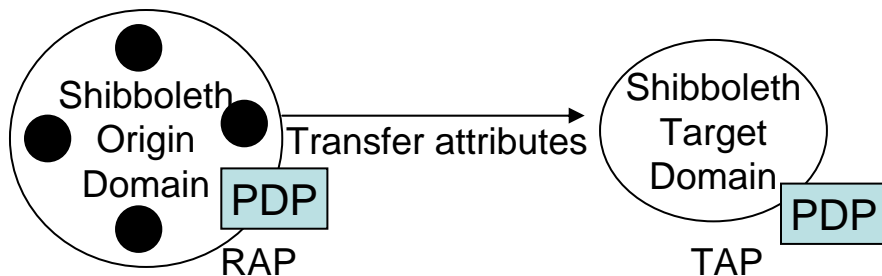
2 Multiple AAs at Origin



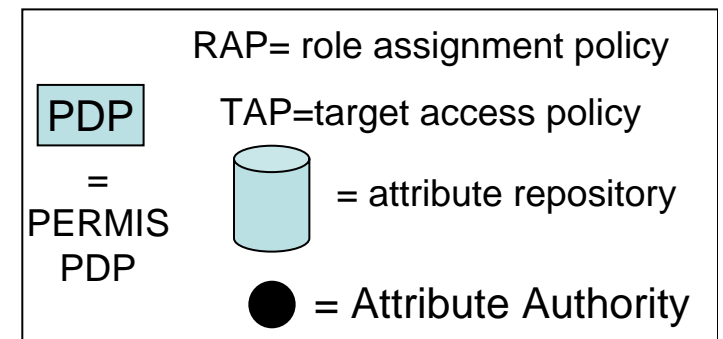
3 Pull ACs from multiple AAs



4 Untrustworthy attribute repository



5 Multiple AAs at Origin not trusted by Target



Privacy Issues

- X.509 ACs bind attributes to a holder, usually their DN
 - E.g. {CN=David Chadwick, OU=Computing Laboratory, O=University of Kent, C=GB}
- But does not have to be the real name of the user
 - Pseudonym {CN=123456789, O=University of Kent, C=GB}
 - Group name {CN=Programmer, OU=Computing Laboratory, O=University of Kent, C=GB}
- Can also be a pointer to user's PKC
 - {x509serialNumber=123456 + x509issuer = {OU=Some CA, O=Some Org, C=US}}
- Or a hash of a public key
 - {rfvbnj756tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6}
- Trade-off between the “degree of anonymity” and the “quality of issuance”.

Revocation and Performance Issues

- Signed SAML assertions in Shibboleth are short lived so don't need to be revoked but have overhead of signing per message
- X.509 ACs are signed so don't need secure channel for communication. But typically long lived so either need ACRLs or repositories under control of issuer so ACs can be deleted
- Could use short lived ACs but then wont be generated by a human and are same as short lived SAML assertions
- ACs typically faster to create and validate than signed XML assertions