

The Role of Directories in PKIs (and PMIs)

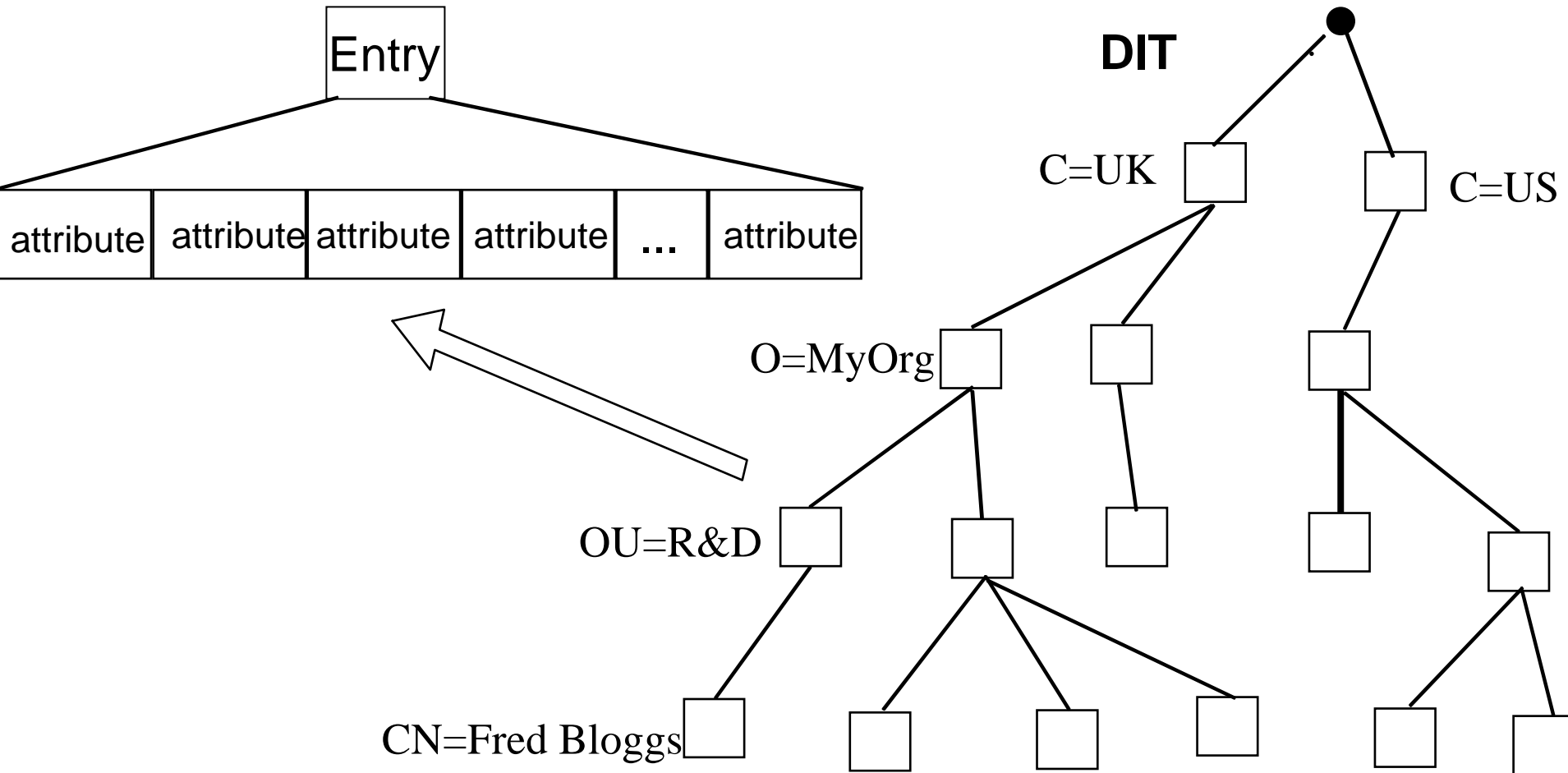
David Chadwick

d.w.chadwick@kent.ac.uk

Which directories are we talking about?

- X.500 based directories, including LDAP directories
 - Both use the X.500 information model for naming entries and describing attributes
 - Probably not MS Active Directory, which is very proprietary!!

X.500 based Directories



What is the difference between an X.500 directory and an LDAP directory?

- The access protocol (DAP vs. LDAP)
 - LDAP has won this battle (alas! See later)
- The way X.509 attributes are understood and stored (see next slide)
- The distribution model (distributed servers with DSP protocol and chaining vs. standalone servers that refer to each other (sometimes))
 - X.500 is still superior here, especially for cross certified PKIs, as various PKI trials have shown

Some more differences

- X.500 is specified using ASN.1. ASN.1 defines the abstract syntax for protocol elements, as well as a series of encoding rules for producing protocol messages
- LDAP simply defines a way of mapping SELECTED X.500 protocol elements into (mainly ASCII) protocol messages
- X.509 is native to X.500, so the protocol for storage, retrieval and searching for X.509 attributes is built in
- LDAPv2 (and v3) omitted to define most of the protocol encodings and all the matching rules (schemas) for X.509 attributes. So they are simply stored as binary blobs
- Hence LDAP does not easily support PKIs or PMIs

Consequences (of no matching rules)

- LDAP PKI clients cannot search for specific X.509 attributes stored in LDAP directories e.g.
 - Find the encryption PKC for the person whose email address is fred.bloggs@myorg.com
 - Find the CRLs issued by OU=MyCA, O=MyOrg, C=US after 9am, 20March 2004
 - Find the roles of David Chadwick embedded in an AC
- PKI clients currently can only store and retrieve X.509 attributes by knowing the Distinguished Name of the entry they are held in – but often the users do NOT know the DNs of entries. At most they know their email address
- PKI clients cannot add X.509 attributes to an entry in an LDAP server (can only read and replace the entire set)

One Current Workaround

- PKI vendors such as Entrust, suggest that the PKI/LDAP administrator extracts the email address of the PKC subject from the SubjectAltName field, and stores this in an email attribute in the same entry as the PKC, and then the PKI client searches the LDAP server for the email address and asks for the PKC to be returned from the same entry

Solution Requirements

- Define LDAP schema for
 - Carrying X.509 attributes in the LDAP protocol
 - Specifying LDAP encodings for particular or all fields of the X.509 attributes
 - Specifying the LDAP matching rules for the X.509 attributes
 - Implement it at least in OpenLDAP (and preferably in all other LDAP products, clients and servers)

Three Alternative Solutions Have Been Proposed

- String Matching
 - Original proposal by David Chadwick in July 2000
- Component Matching
 - Subsequent proposal by Stephen Legg in July 2001
- Attribute Extraction
 - Final proposal by Norbert Klasen and Peter Geitz in Feb 2002

String Matching

- Use existing LDAP encoding mechanisms
- Take existing X.500 matching rules
- Specify unique LDAP string encodings for selected fields of the X.509 attribute
- Encode these fields in the X.509 attribute on receipt, and store these in its indexes
- LDAP client can use new encodings to search for fields
- Solution deficiencies
 - Does not specify encodings for all current X.509 fields
 - Does not specify encodings for future fields e.g. new certificate extensions

Component Matching (1)

- Specify text encodings for every ASN.1 built in data type: integer, boolean, SET OF, SEQUENCE etc. so that any attribute syntax will automatically have a string encoding
 - Specified in RFC 3641 “Generic String Encoding Rules (GSER) for ASN.1 Types”. S. Legg. October 2003.
- These rules are based on standard LDAP encoding rules e.g. Boolean -> “TRUE” or “FALSE”, Sequence and Set -> comma separate list etc.

Component Matching (2)

- Specify how string assertions can be specified about components of ASN.1 values, through a ComponentAssertion construct

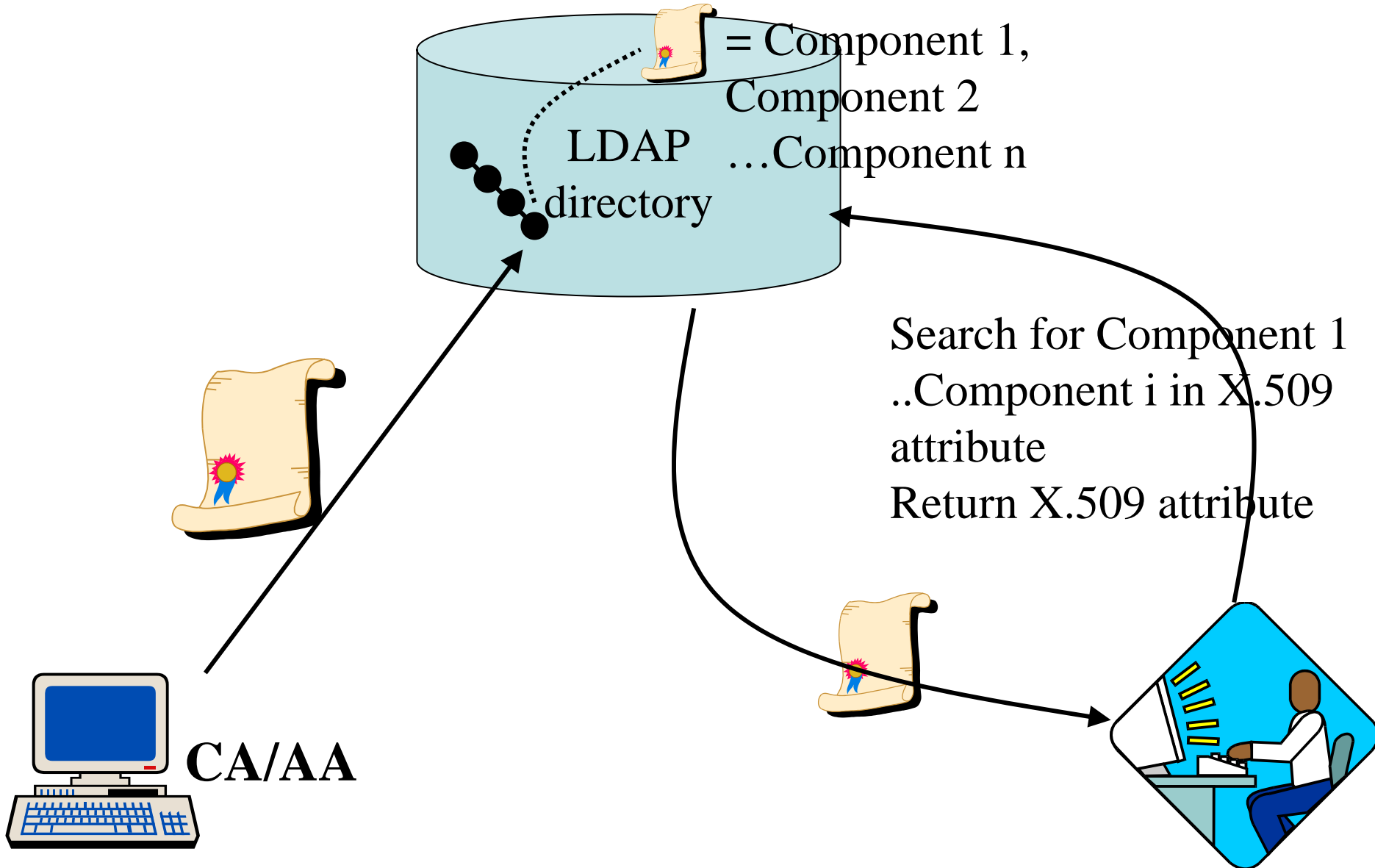
```
ComponentAssertion ::= SEQUENCE {  
    component      ComponentReference (SIZE(1..MAX)) OPTIONAL,  
    useDefaultValues  BOOLEAN DEFAULT TRUE,  
    rule           MATCHING-RULE.&id,  
    value          MATCHING-RULE.&AssertionType }
```

```
– E.g. ExampleType ::= SEQUENCE {  
    part1    [0] INTEGER,  
    part2    [1] ExampleSet,  
    part3    [2] SET OF OBJECT IDENTIFIER }
```

Component reference “part3.2” refers to the second OID occurrence in the ASN.1 value

- and define a new LDAP component filter matching rule which allows a user to match on multiple different components within a complex attribute value
 - Specified in RFC 3687 Lightweight Directory Access Protocol (LDAP) and X.500 Component Matching Rules. S. Legg. February 2004.

Component Matching



Component Matching Example

- Suppose you want to search for person with an email address of "support@foobar.com" within the rfc822Name within a subjectAltName extension in a PKC.

The search filter is:

```
(userCertificate:componentFilterMatch:=item:{  
component  
"toBeSigned.extensions.*.extnValue.content.(2.5.29.1  
7).*rfc822Name", rule caseIgnoreIA5Match, value  
"support@foobar.com"})
```

How it might work

- Andrew Sciberras a colleague of Steven Legg, configured Mozilla Address Book via Advanced Settings with the following search filters to search View 500 directory
- Title "Cert v3 People"
Filter: (userCertificate:componentFilterMatch:=item:{ component "toBeSigned.version", rule integerMatch, value 2})
- Title "Certificates of people from COMPANY"
Filter: (userCertificate:componentFilterMatch:=item:{ component "toBeSigned.subject.rdnSequence.*", rule rdnMatch, value "O=COMPANY"})

Component Matching

Pros

- Elegant solution
- Extensible. Once implemented it should easily cater for new X.509 attribute syntaxes and new extensions to certificates and CRLs

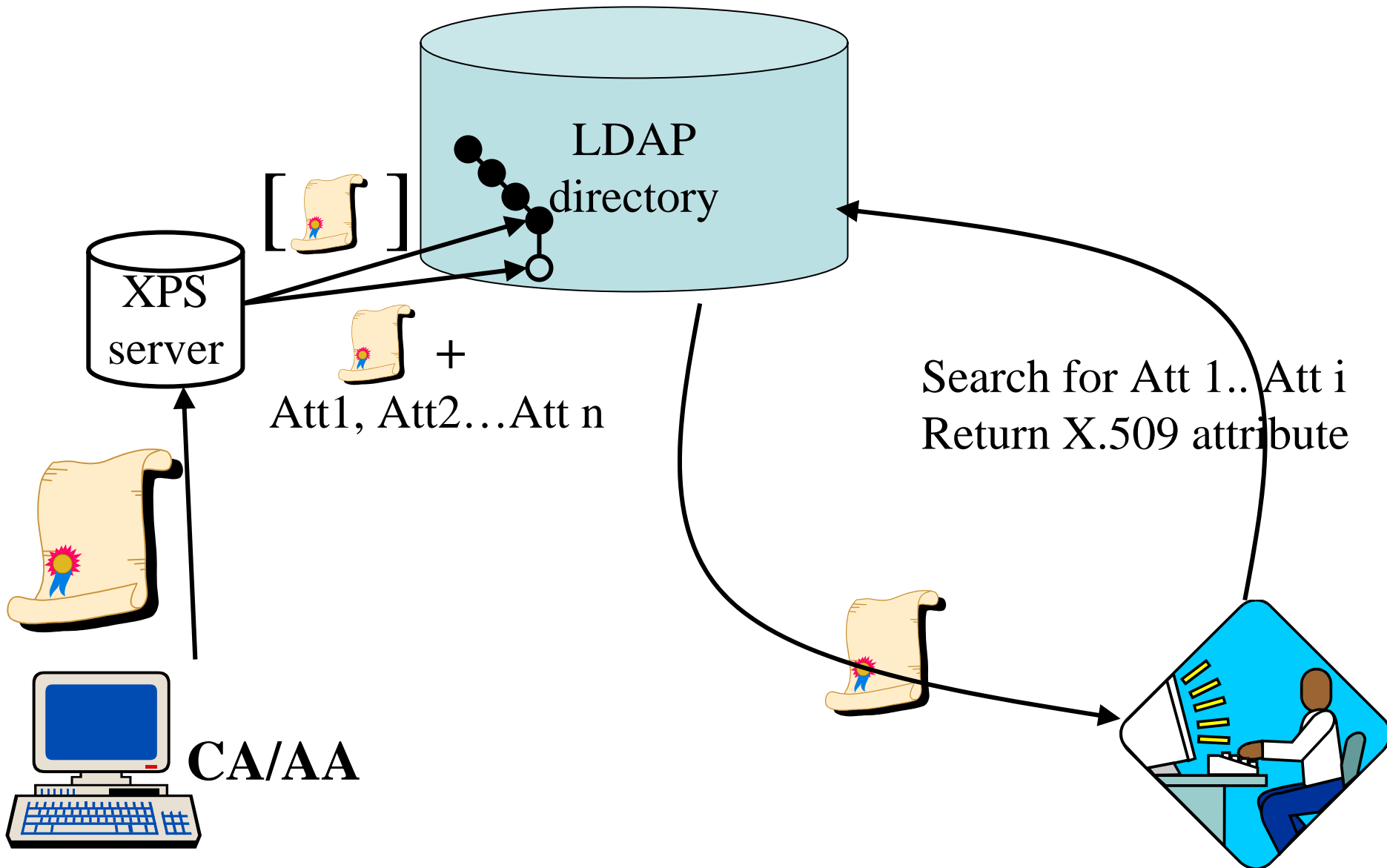
Cons

- Complex to implement in server.
- All LDAP servers will need modifying. Few LDAP server suppliers have indicated any willingness to implement it
- Most PKI clients will need modifying to implement GSER and new matching rule, and a new API will need building for them

Attribute Extraction

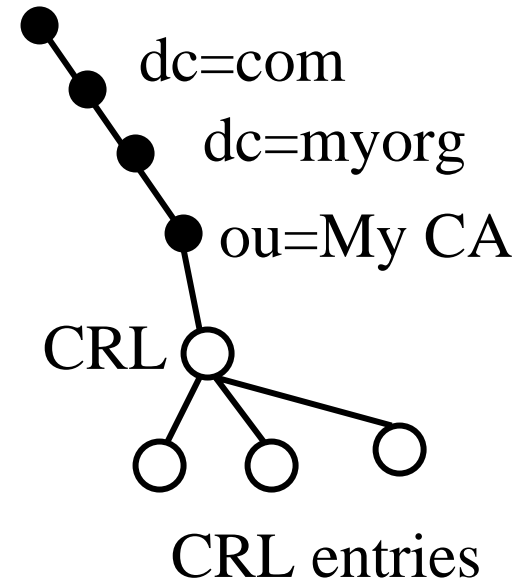
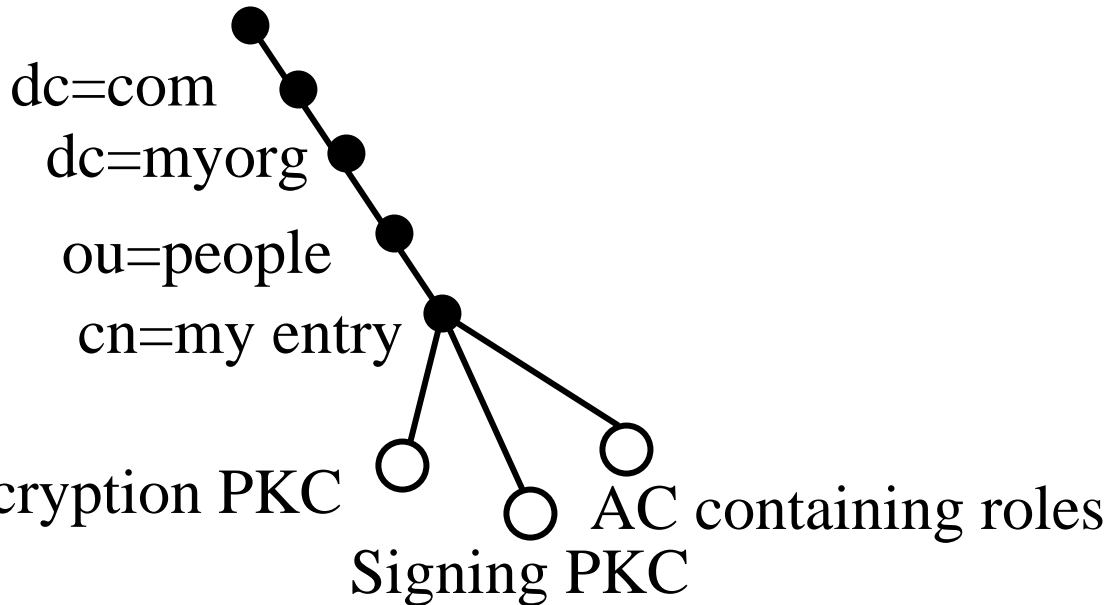
- Specify new LDAP attributes for (most) fields of the X.509 attributes
- Use existing LDAP schema for these attributes (storing and matching)
- Use a front end to parse the X.509 attributes to be stored, and break them up into the new LDAP attributes
 - Front end creates a new LDAP entry for each X.509 attribute comprising the original X.509 attribute and the set of extracted attributes
 - The LDAP server creates the new entry and adds the extracted attributes to its indexes, using existing mechanisms
 - CA talks to the new front end instead of the LDAP server
 - PKI clients search the LDAP server for the extracted attributes and ask for the X.509 attribute to be returned

Attribute Extraction



The DIT Structure

- PKCs and ACs are held in child entries
- CRLs are held in child subtrees (so that individual revoked certificates can be searched for)



Naming the X.509 attribute entries

- CRL entries are named with the x509crlThisUpdate attribute
- CRL revoked certificate entries may be named in one of 3 ways
 - x509serialNumber+x509issuer
 - x509issuerSerial
 - x509serialNumber
- Certificate Entries may be named in one of 3 ways
 - x509serialNumber+x509issuer
 - x509issuerSerial
 - x509serialNumber

New Schemas Specified In PKIX IDs

- Internet X.509 Public Key Infrastructure Lightweight Directory Access Protocol Schema for X.509 Certificates
<draft-ietf-pkix-ldap-pkc-schema-01Internet X.509>
- Public Key Infrastructure LDAP Schema for X.509 CRLs
<draft-ietf-pkix-ldap-crl-schema-03.txt>
- Internet X.509 Public Key Infrastructure LDAP Schema for X.509 Attribute Certificates
<draft-ietf-pkix-ldap-ac-schema-02.txt>

Attribute Extraction

Pros

- Existing LDAP servers do not need to be modified
- Supports enhanced matching (searching on multiple fields within a single PKI attribute)
- PKI clients need less modification, and might just need re-configuring with the new attribute types

Cons

- Storage requirements in the LDAP server are doubled [or tripled]
- Adding new X.509 attribute syntaxes or new certificate or CRL extensions means new attributes have to be defined and the front end has to be recompiled and rebuilt

State of Implementations Today

- Component matching is implemented in OpenLDAP 2.3 (presently in alpha) and in View 500 from Australia
- Attribute extraction (XPS server) is implemented in OpenLDAP 2.2.8 but not integrated into the development branch therefore it is available from University of Salford web site at [http://sec.cs.kent.ac.uk/download/openldap-2.2.8\(final\).zip](http://sec.cs.kent.ac.uk/download/openldap-2.2.8(final).zip)

The Salford XPS Server

- X.509 attribute Parsing Server (XPS)
- Built by modifying OpenLDAP server
- Modified OpenLDAP can run as a combined XPS/LDAP server, or can front end existing LDAP servers
- Creates a Write Ahead Log to store the transactions to be made to the LDAP server, enabling rollback and recovery
- Config parameters determine
 - how the new entries are to be named
 - Which X.509 attributes are to be parsed
 - The original X.509 attribute can optionally be stored in the parent entry
- A config table determines which new attributes are to be extracted and stored

