IBM Research / IBM Linux Technology Center

Design and Implementation of LDAP Component Matching for Flexible and Secure Certificate Access in PKI

Sang Seok Lim
IBM Research
slim@us.ibm.com

Jong Hyuk Choi
IBM Research
jongchoi@us.ibm.com

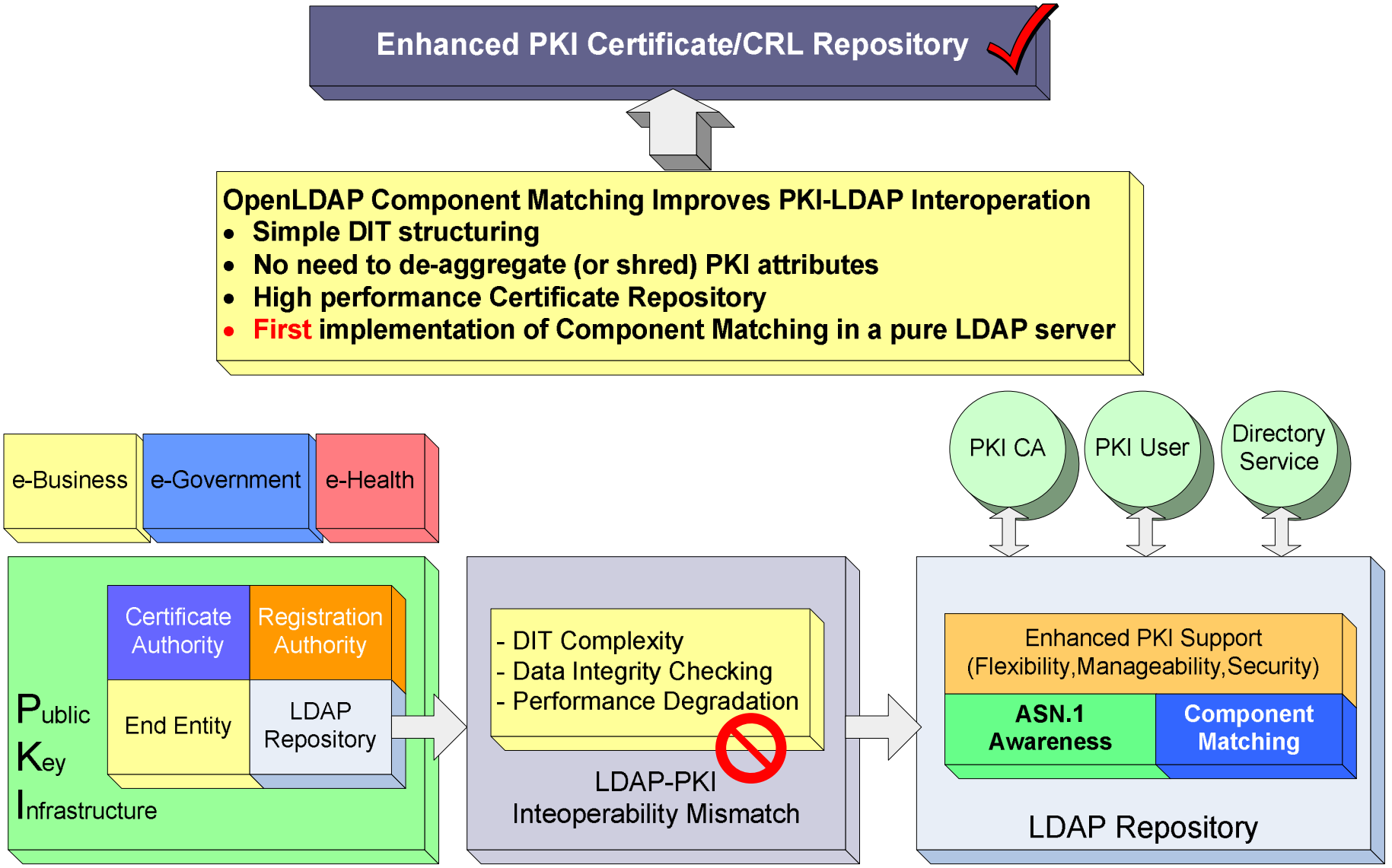
Kurt Zeilenga
IBM Linux Technology Center
zeilenga@us.ibm.com

4th PKI R&D Workshop, NIST, Gaithersburg MD 2005

Outline

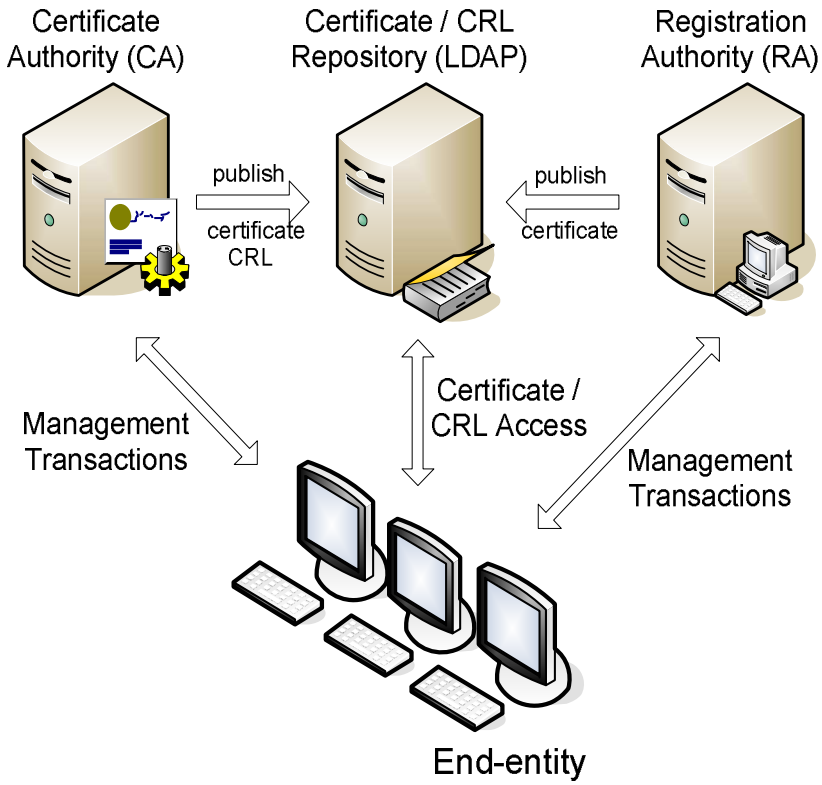
- **Introduction**
- **LDAP-PKI Interoperability Limitation**
- **Interoperability Enabling Technologies**
  - Component Matching
  - ASN.1 Awareness
  - Generic String Encoding Rule (GSER)
- **OpenLDAP Implementation**
  - ASN.1 Generation of ASN.1 Decoders / Matching Rules
  - Component Matching Architecture and Data Structures
  - Component Matching Optimizations
- **Performance Evaluation**
- **Summary**

OpenLDAP Component Matching: Overview

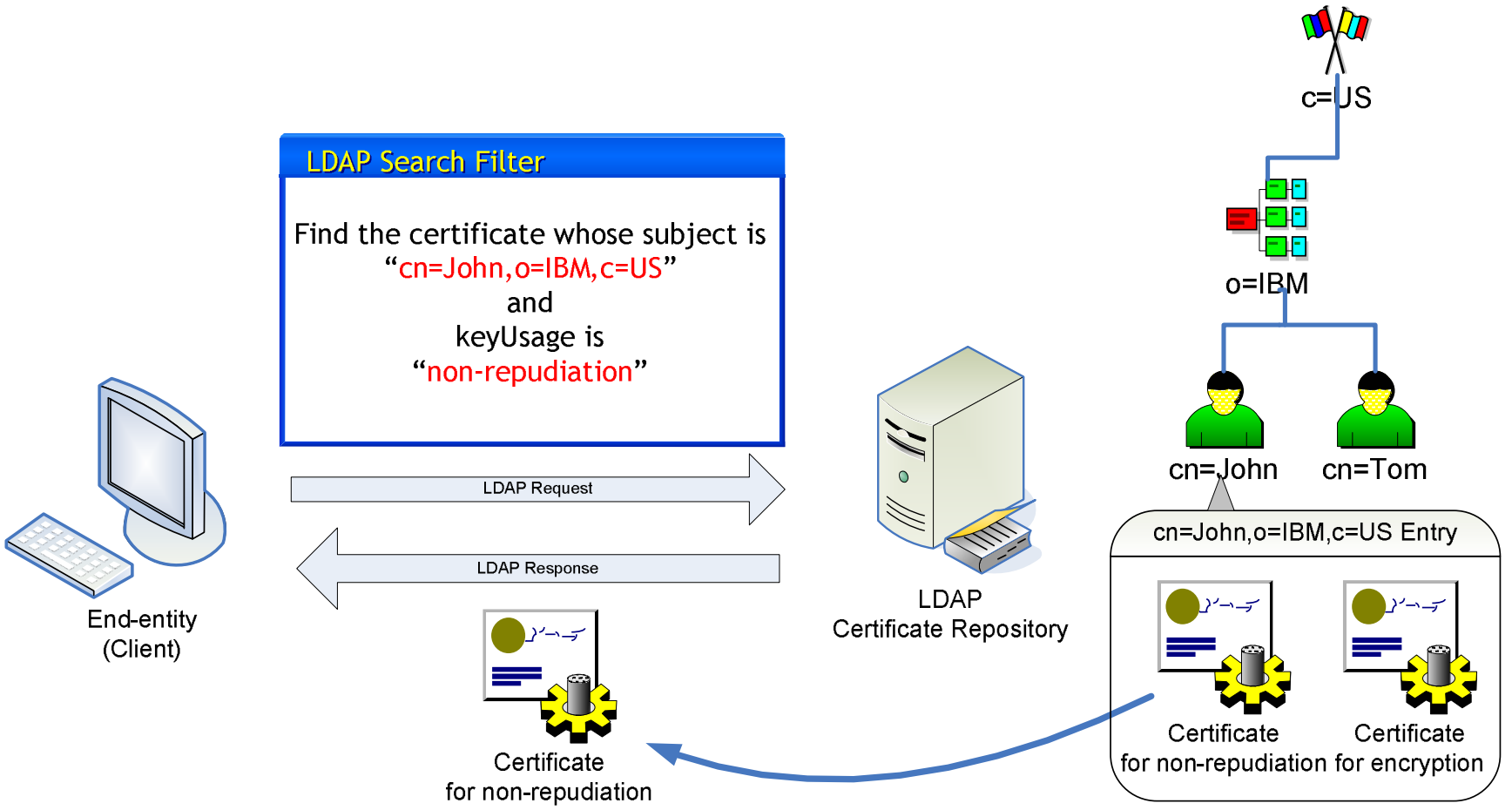


Usage of LDAP in PKI

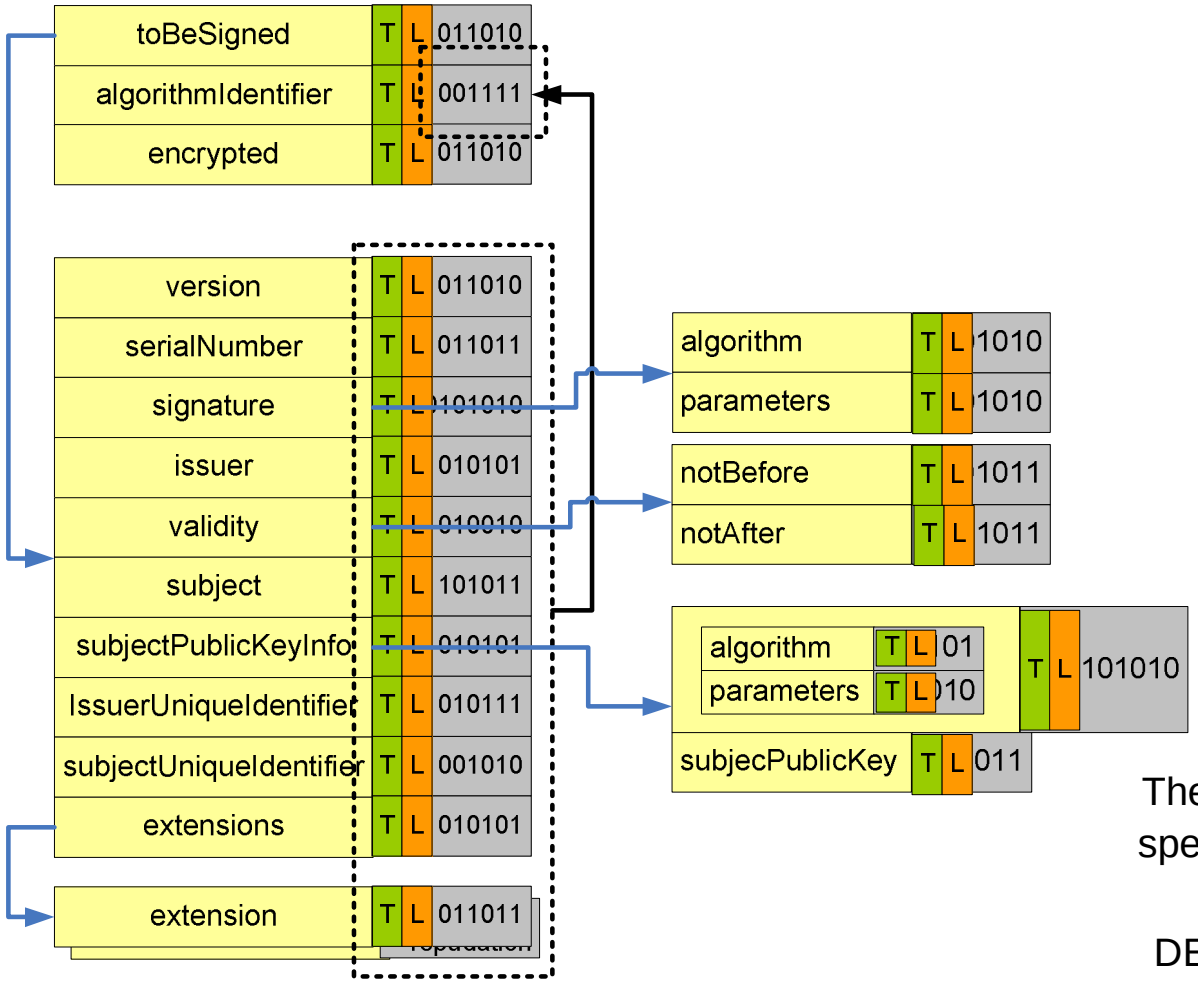
- **Certificate access protocols in PKI**
  - LDAP, HTTP, FTP, etc.
- **LDAP: a predominant way of implementing a PKI repository**
  - X.509 was originally specified in the X.500 context
  - LDAP is predominantly used directory protocol for the Internet
    - Various protocol operations: search, modify, update, etc
    - Authentication: Simple Authentication Security Layer (SASL)
    - LDAP Control
    - Access control
- **LDAP has X.500 DAP simplified by mainly**
  - String-based encoding
    - Incapable of preserving structural information
    - DAP uses ASN.1 encodings (BER/DER)
  - Direct TCP/IP mapping
  - Simple-protocol encoding



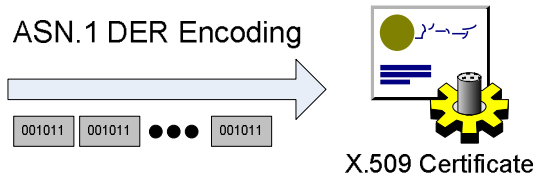
Certificate Access using LDAP



Certificate Structure and Encodings (bits-on-the-line)



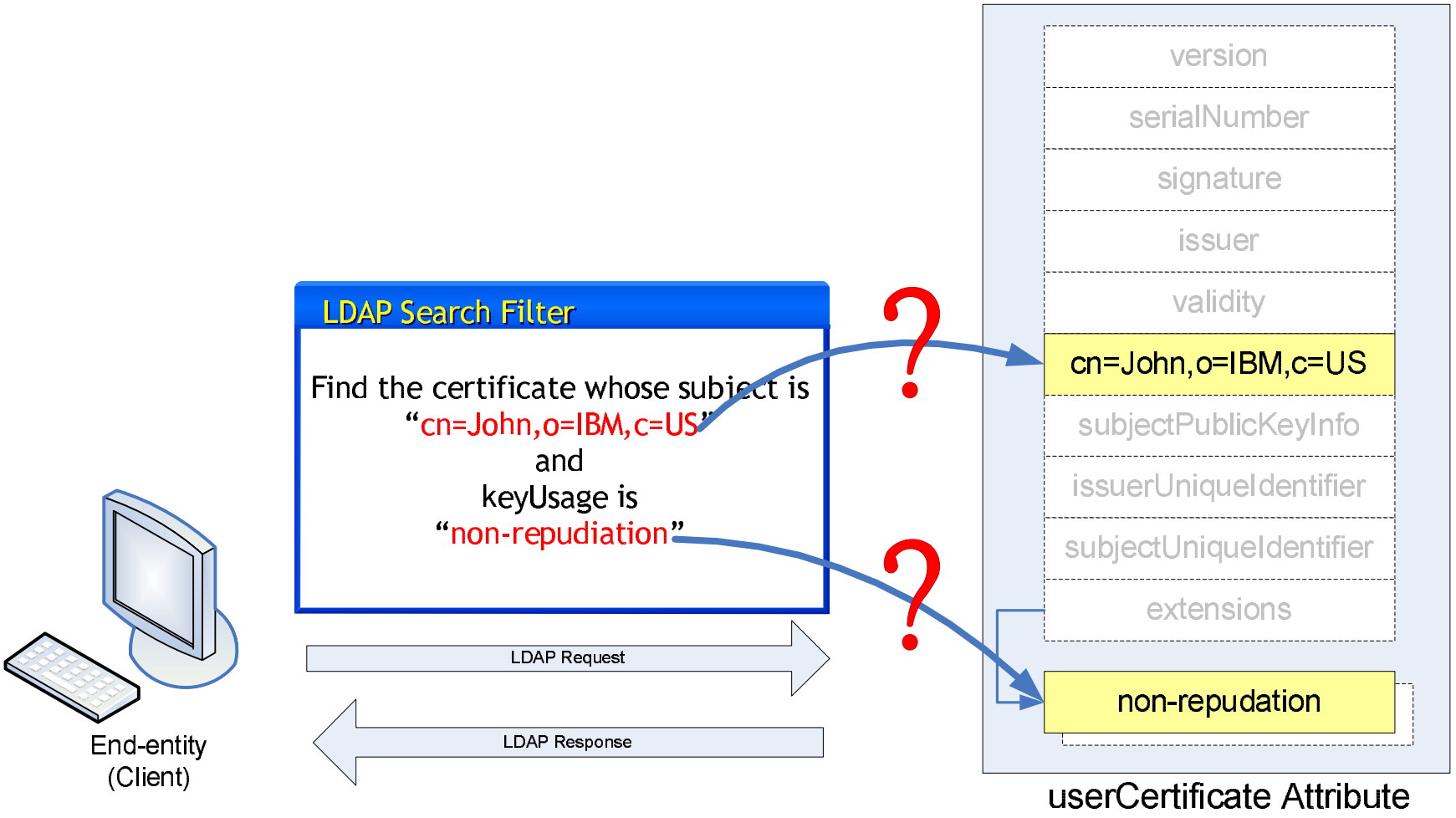
Pictorial Representation of
X.509 Certificate ASN.1 Specification



The structural information of ASN.1 types specification is preserved in the encodings

DER: Distinguished Encoding Rules

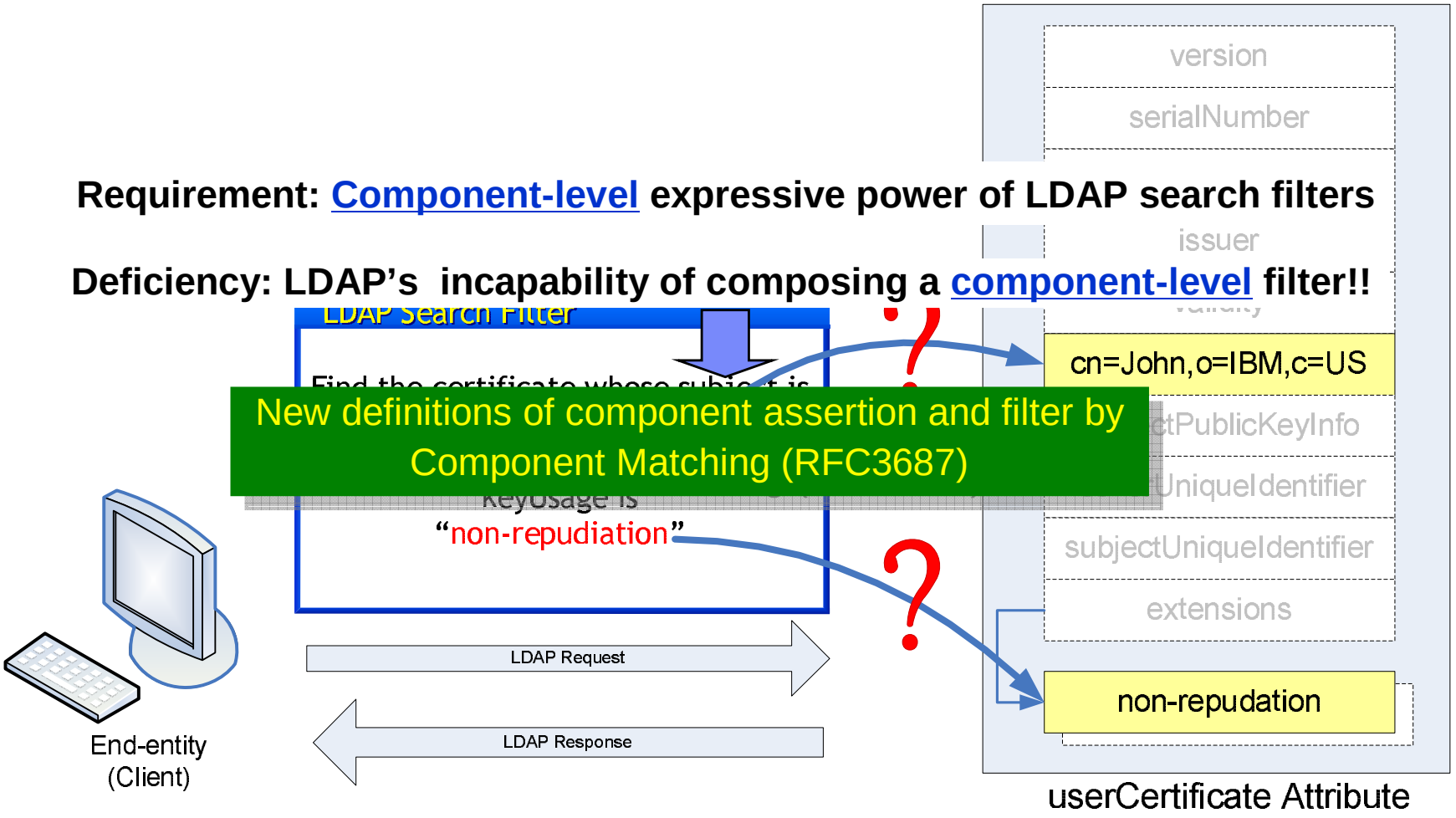
LDAP Client Side Requirement



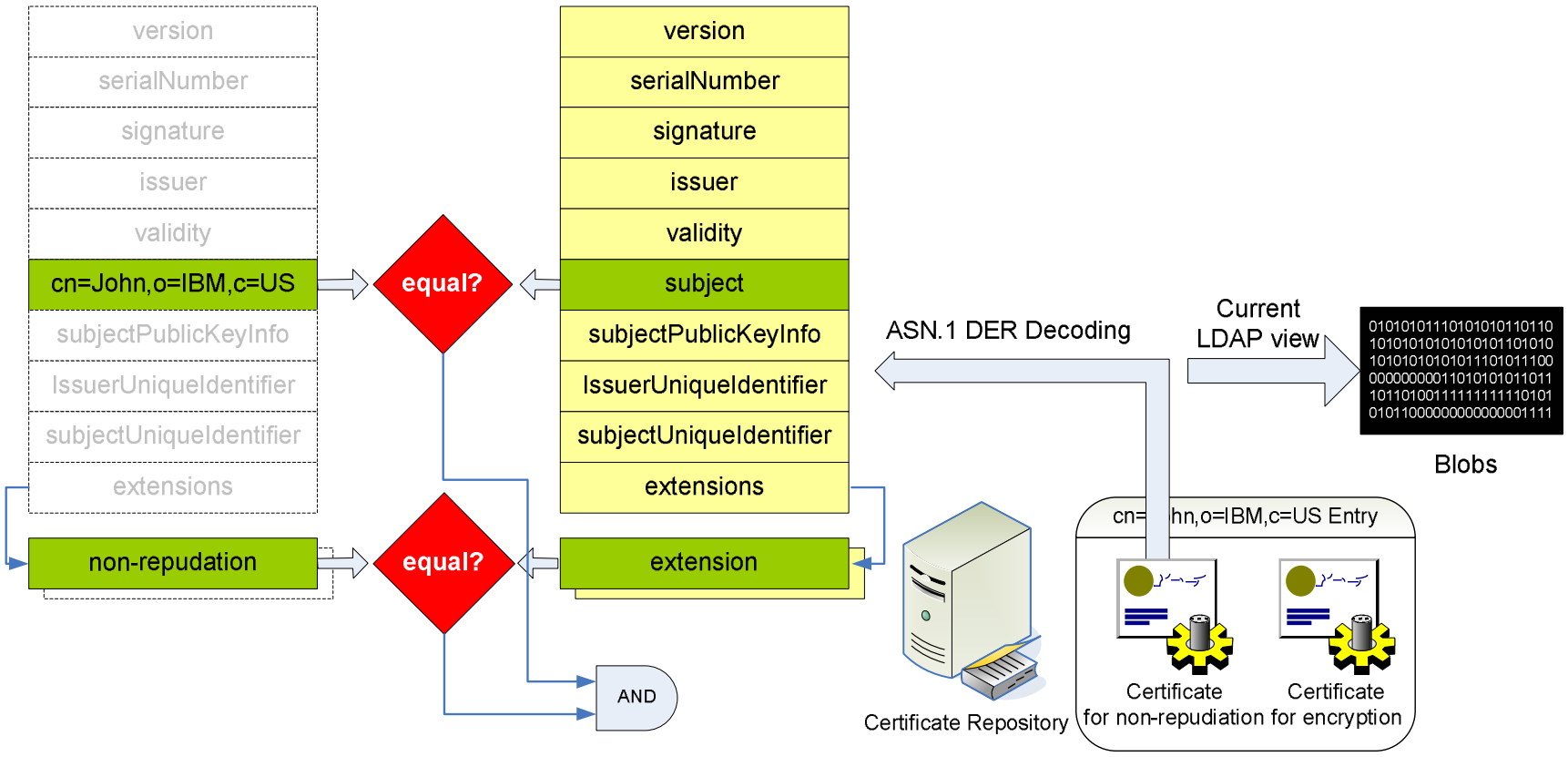
LDAP Client Side Requirement

Requirement: Component-level expressive power of LDAP search filters

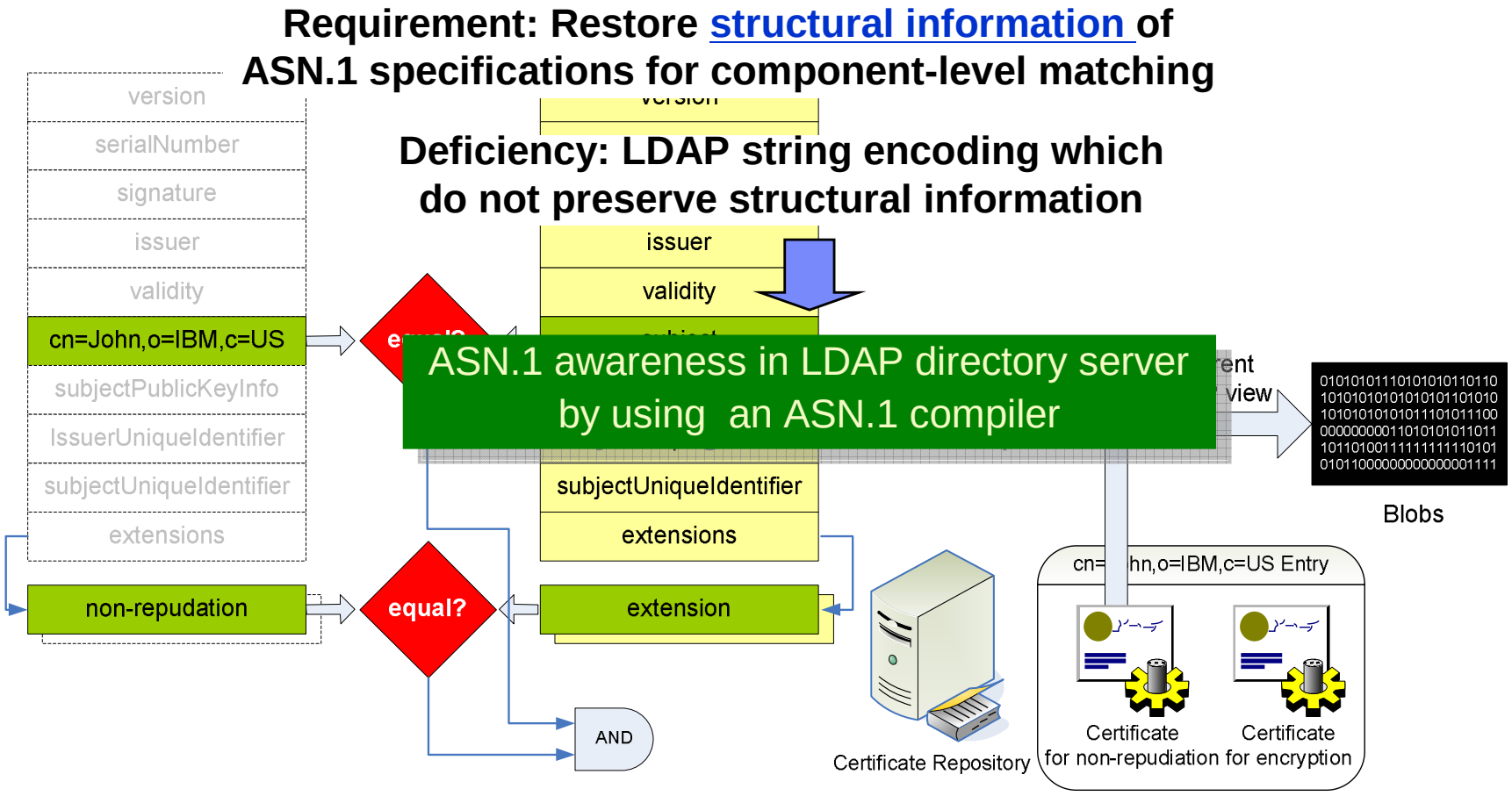
Deficiency: LDAP's incapability of composing a component-level filter!!



LDAP Server Side Requirement

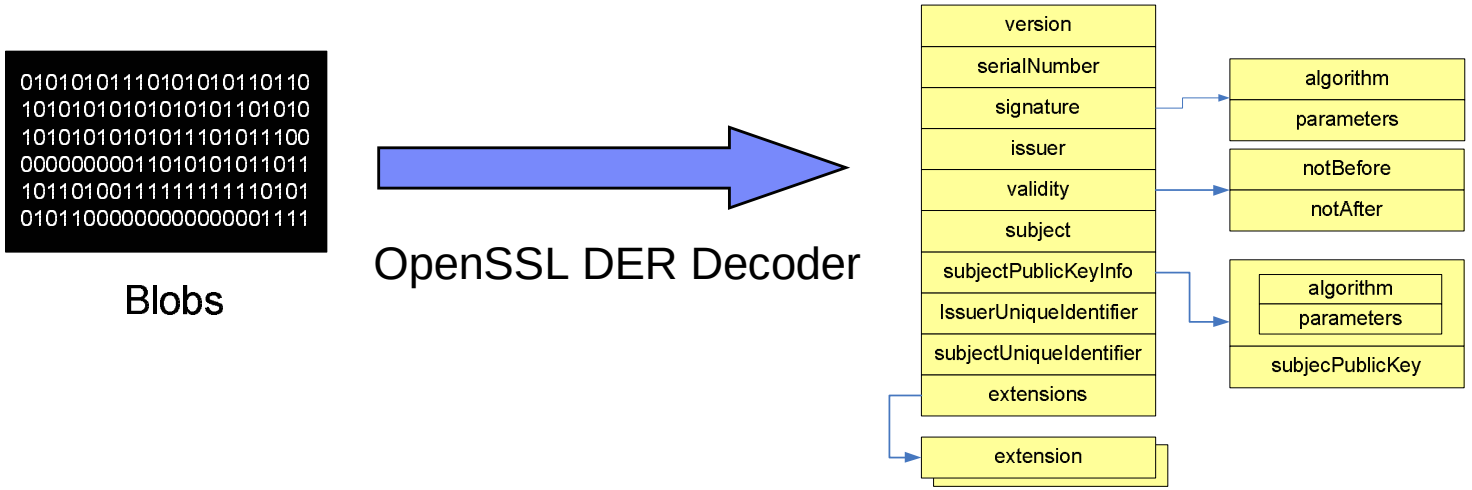


LDAP Server Side Requirement



Certificate Syntax Specific Parsing

- **Parsing the blobs (DER) by certificate-syntax specific decoders**



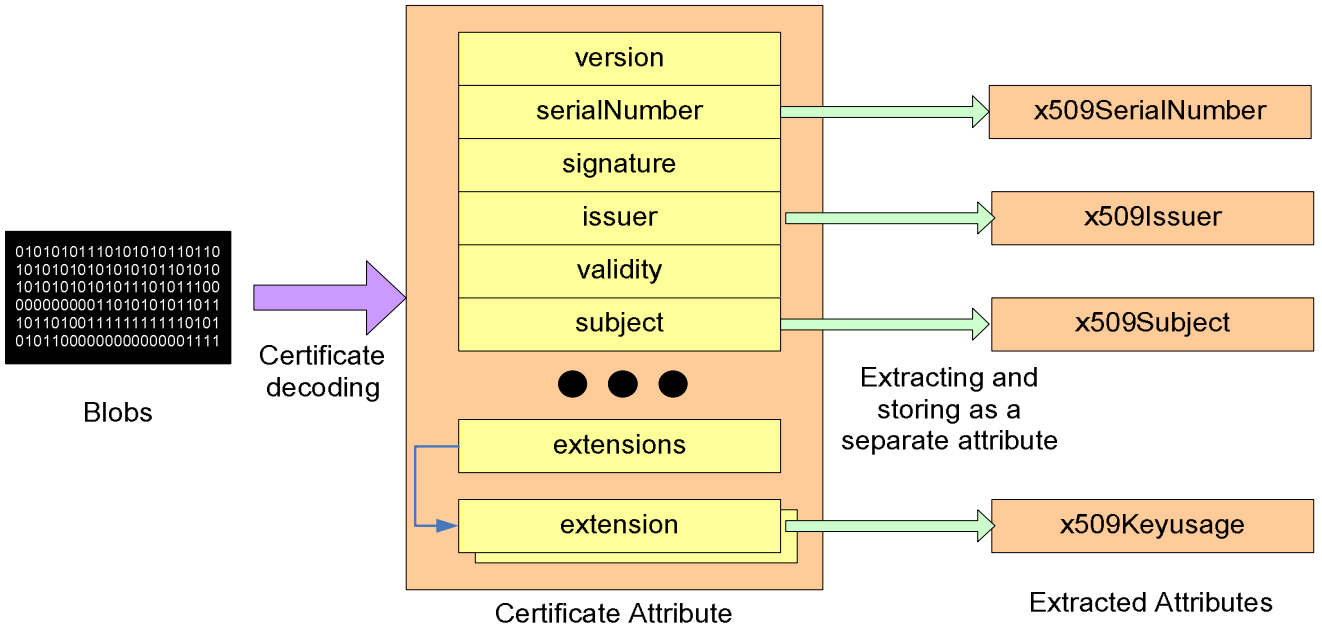
- **Limited and inflexible component-level filter composing**
  - Example search filter

userCertificate:**certificateExactMatch**=cn=CA,o=IBM,c=US**$**12345

- **Disadvantages**
  - Inflexibility

issuer **separator** **serial**

Attribute Extraction



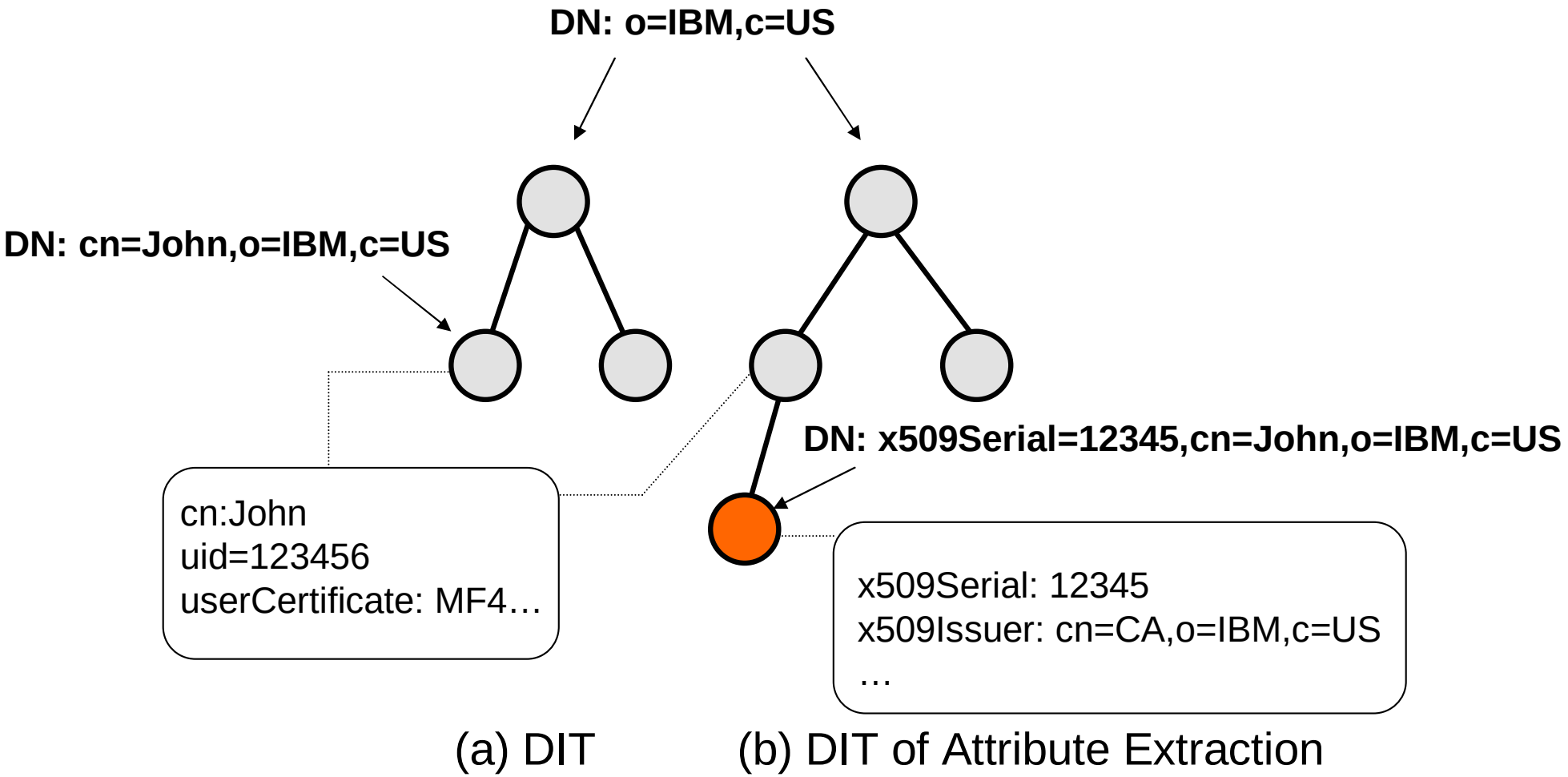
- **Example search filter (use an attribute-level filter)**

  (&((x509KeyUsage=010000000)(x509Subject=cn=John,o=IBM,c=US)))

- **Certificate Parsing Server (XPS): automates the extraction process**
- **Disadvantages**
  - High storage requirements
  - Security Issue
    - Data integrity checking: matching is performed unsigned extracted attributes
  - Poor manageability
    - DIT Restructuring to support multiple certificate for an user

Directory Information Tree of Attribute Extraction

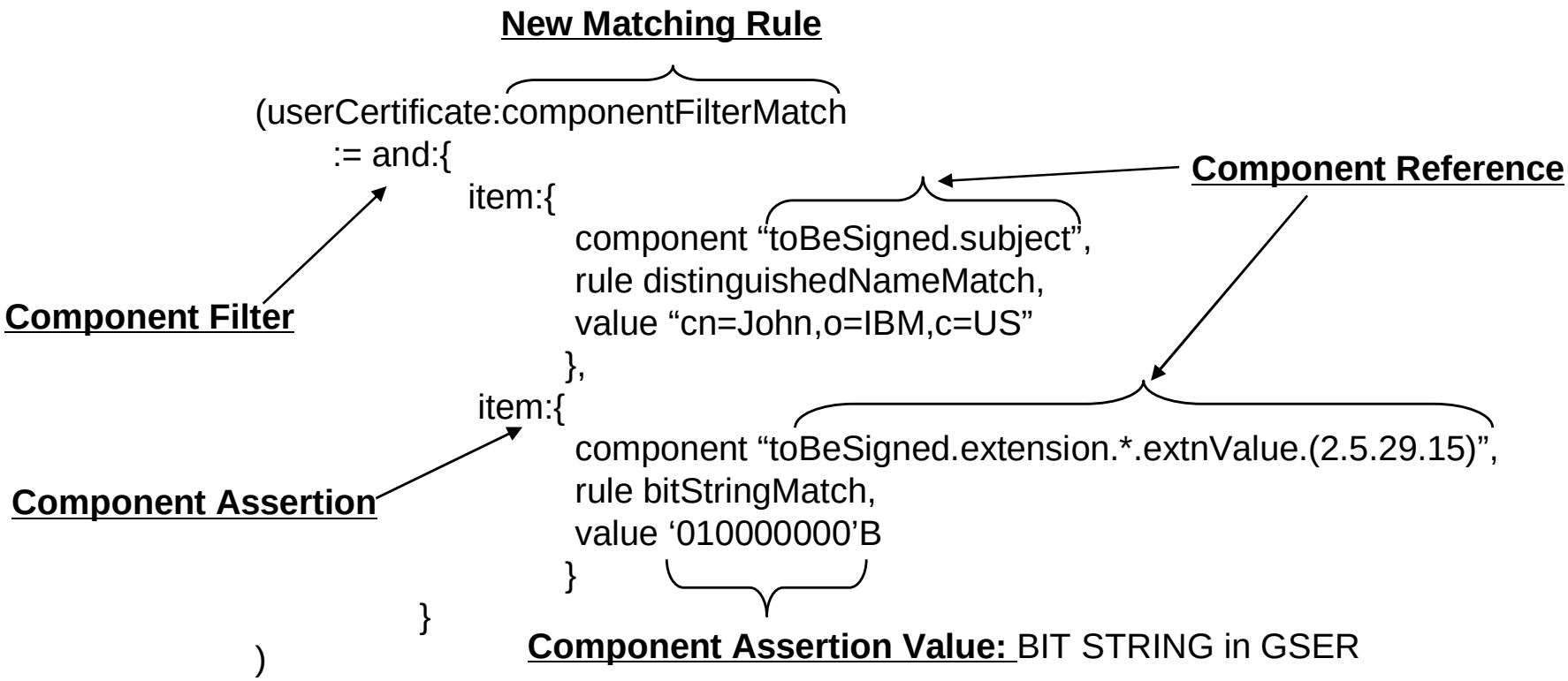
- **DIT Restructuring for storing multiple certificates for a user**



Example Component Matching Filter

User request statement:

"Find a certificate whose subject is cn=John,o=IBM,c=US and keyUsage is *non-repudiation*"



ASN.1 Awareness and Component Matching

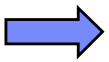
- **ASN.1 awareness of LDAP directory servers**
  - Ability to utilize the structural information of an ASN.1 type.
  - Understand how to construct complex types by combining basic types and composite types of ASN.1
- **Component matching for ASN.1 awareness**
  - Define how to describe component filter and assertion for components
  - Define how to reference a component : *component reference*
  - Define a generic way of matching among components at an ASN.1 level
  - Generic String Encoding Rule (GSER)
    - Introduce structure information into UTF-8 based string encodings
    - The value of component assertion is encoded in GSER

Generic String Encoding Rule

- **Define UTF8 string encodings rule for basic ASN.1 types and composite ASN.1 types**
  - INTEGER, BOOLEAN, OCTET STRING etc
  - SEQUENCE, SET, CHOICE, etc

```
toBeSigned  :: = SEQUENCE {
    version       [0] EXPLICIT Version DEFAULT v1,
    serialNumber  CertificateSerialNumber,
    signature         AlgorithmIdentifier,
    issuer              Name,
    validity            Validity,
    subject             Name,
    subjectPublickKeyInfo subjectPublicKeyInfo,
      …
    extensions    [3] EXPLICIT Extensions OPTIONAL
}
```

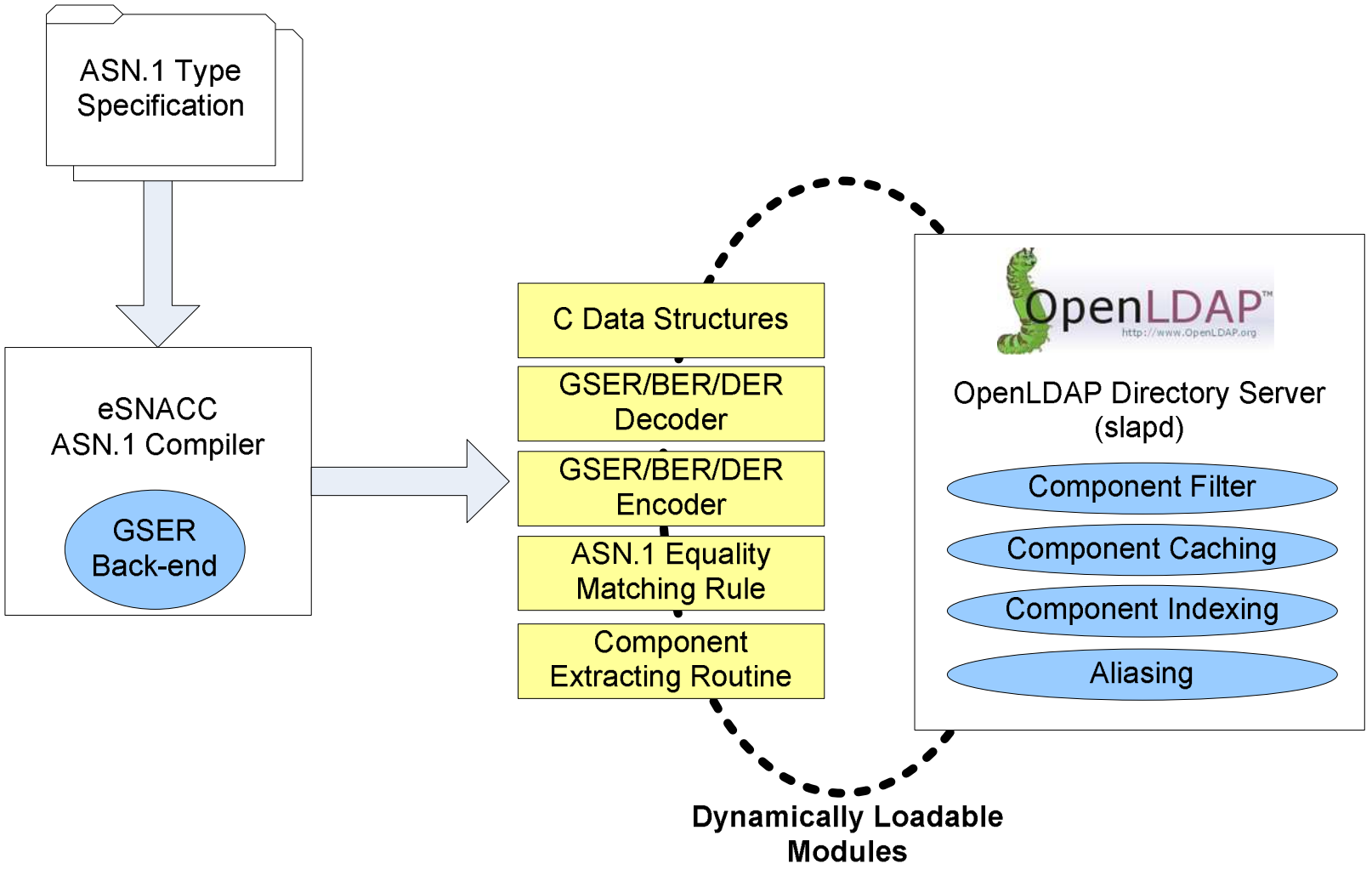
Certificate ASN.1 type



```
{ version 2,
  serialNumber 12345 ,
  signature { algorithm 1.2.840.113549.1.14, parameters NULL},
  issuer {{type cn, value IBM trust} , {type o, value IBM},{type c, value US}},
  validity {notBefore {2004 01 13 18 59}, notAfter {2005 01 13 18 59} },
  …
}
```

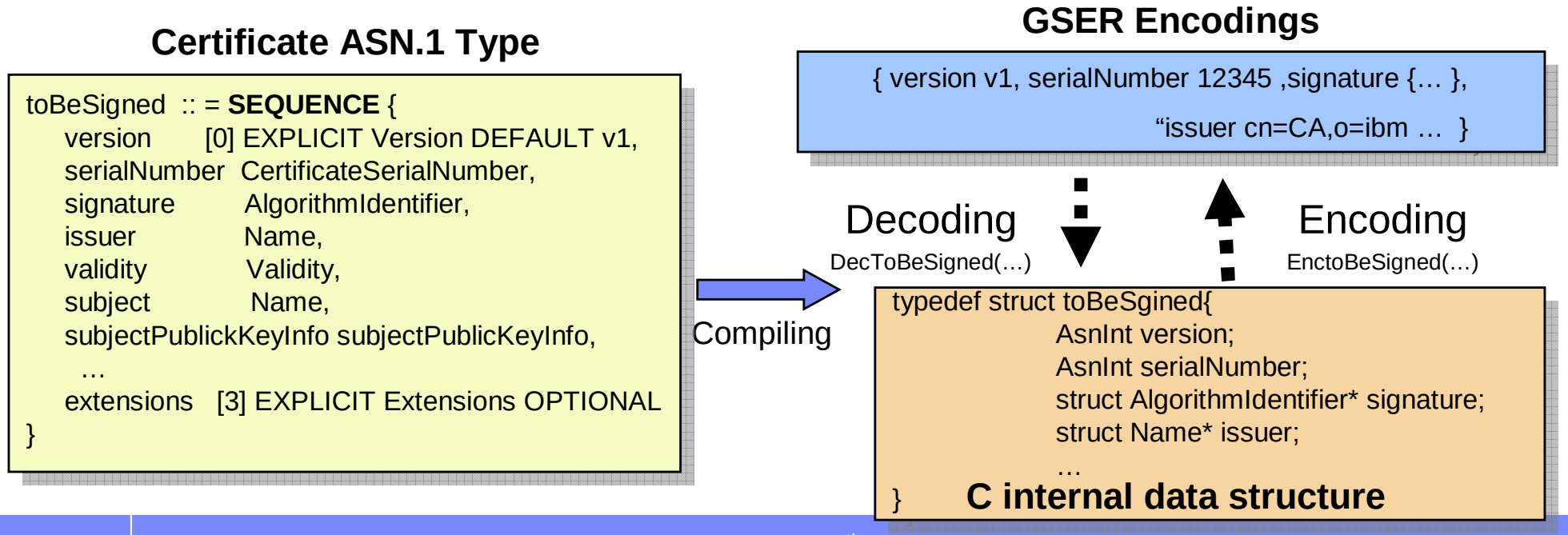
Example GSER Encodings of Certificate

Our Framework of Component Matching Enabled *OpenLDAP*



An *eSNACC* Compiler and a GSER Back-end

- **What does the compiler do?**
  - Compile ASN.1(Abstract Syntax Notation One) modules into
    - ASN.1 equivalent C data structures
    - Routines to convert to/from the internal (C or C++) representation from/to the corresponding BER/DER formats
- **GSER back-end**
  - Generate GSER encoding/decoding functions of given ASN.1 types



Component Representation

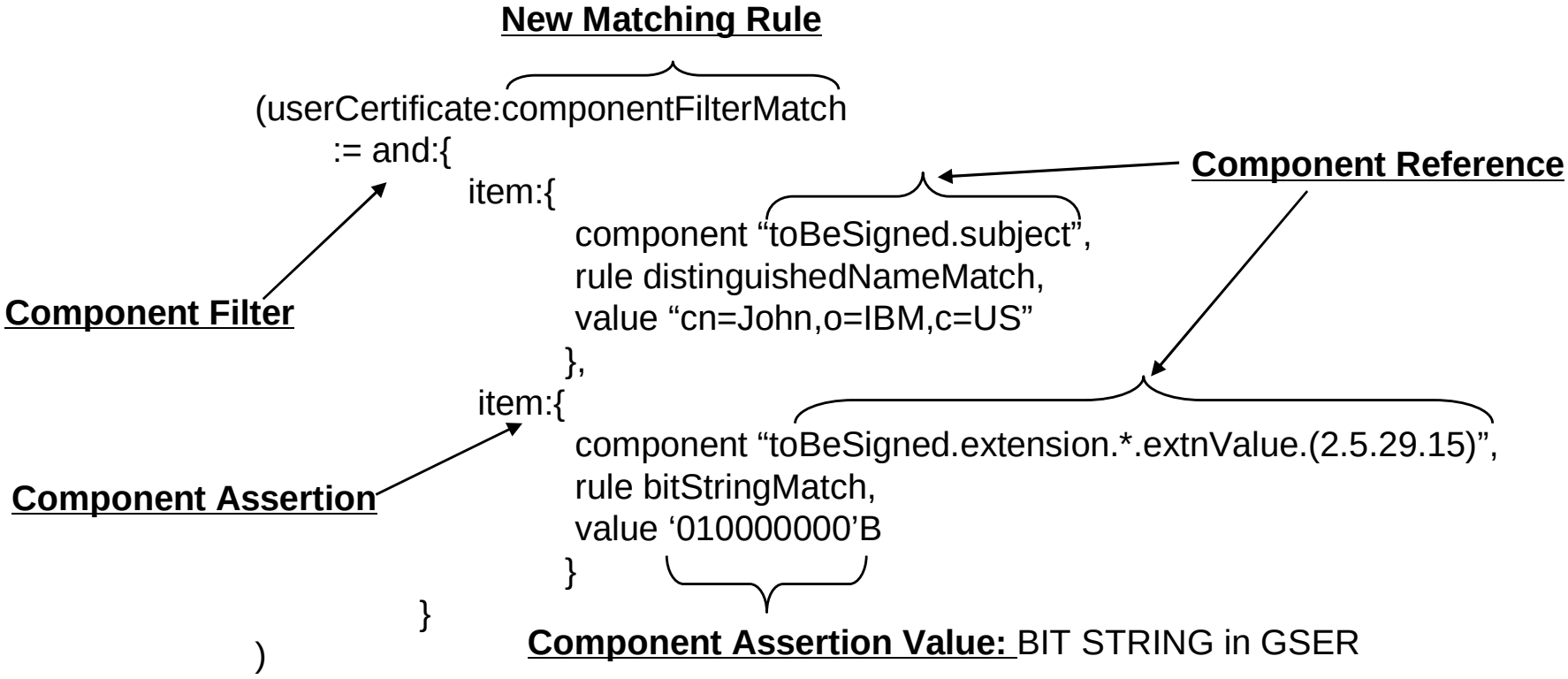
- **Component**
  - The arbitrary part of attribute value that can be referenced or identified by Component Reference
- **Component representation**
  - Preserve ASN.1 structure information in the internal representation
  - Two parts
    - Data value
      - Value in a C internal data structure
    - Component descriptor for its value processing
      - Value-specific encoder/decoder/matching rule/extract
  - Component tree
    - All components are comprised of one tree
- **Component extraction**
  - Extract referenced components from a component tree
  - Generated by the eSNACC ASN.1 compiler automatically

Component Matching Implementation

- **Component assertion (GSER encoded)**
  - An assertion about the presence, or values of, components within an ASN.1 value
  - Component reference
    - Identifying the component part of a ASN.1 value.
  - Matching rule
  - Component assertion value in GSER
- **Component filter (GSER encoded)**
  - An “and”, “or”, “not” expression of ComponentAssertion, evaluates to either TRUE, FALSE or Undefined
- **Component equality matching rule**
  - allComponentsMatch and refined matching rules
  - Generated by the eSNACC ASN.1 compiler automatically

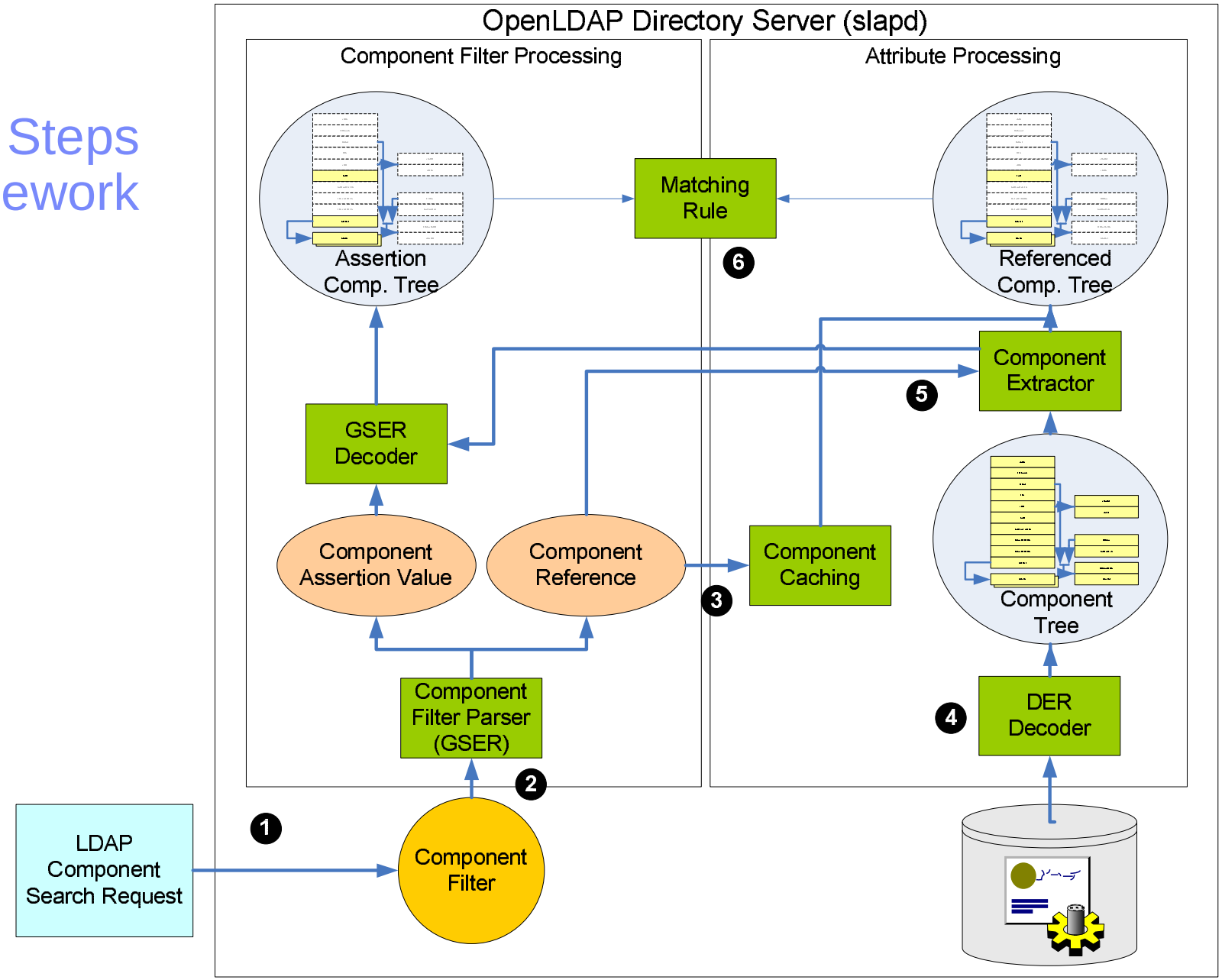
User request statement:

"Find a certificate whose subject is cn=John,o=IBM,c=US and keyUsage is *non-repudiation*"



Example Component Filter

Overall Operational Steps in Our Framework



Component Matching Optimization

- **Attribute/matching rule aliasing**
  - Backward compatibility for legacy clients
  - Avoid expensive extensible filter processing

x509certificateSerial=12345678

attribute aliasing

userCertificate:componentFilterMatch =:
item:{
component "toBeSigned.serialNumber",
rule IntegerMatch,
value 12345678
}

Alias Attribute	Aliased Attribute	Component Reference	Matching Rule
x509certificateSerial	userCertificate	toBeSigned.serial	integerMatch
x509certificateIssuer	userCertificate	toBeSigned.issuer	distinguishedNameMatch

Example Attribute Aliasing Table

Component Matching Optimization Contd.

- **Component indexing**
  - Boost search performance by supporting component-level indexing
    - Indices on serial number, issuer name, version, etc
- **Component caching**
  - Eliminate certificate (DER) decoding overheads
    - 72usec/certificate : Intel Xeon 2.8GHz
  - Cache decoded internal representations of a certificate

Performance Evaluation

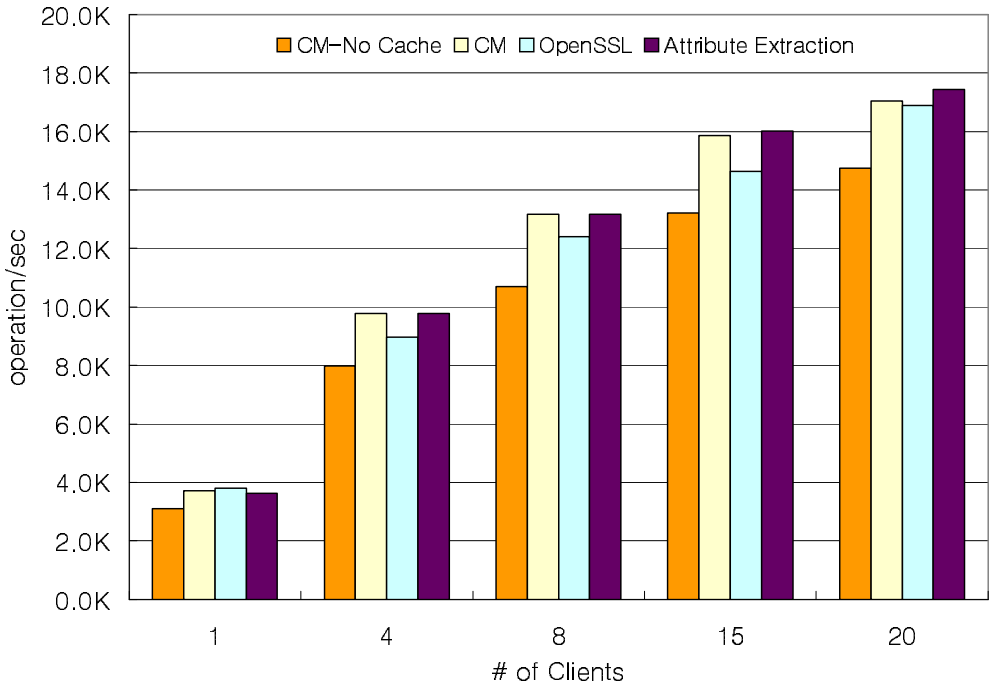
- **Directory population**
  - OpenSSL and Component Matching
    - DirectoryMark generated entries: 100k and 500k
  - Attribute extraction
    - Used XPS (OpenLDAP patch) to extract attributes
- **System under test (SUT)**
  - IBM xSeries 445 servers : 4-way Intel Xeon 2.8GHz with 12GB memory
  - Network connection: 1 Gbps Ethernet
  - Server: OpenLDAP 2.2.2 and Berkeley DB 4.3
  - Clients: DirectoryMark scripts (8-way IBM xSeries 445 servers)
- **Performance measurement**
  - Search throughput (operations/sec)
    - Matching against *serialNumber*
  - 100K entry add performance

Performance (Population / Search)

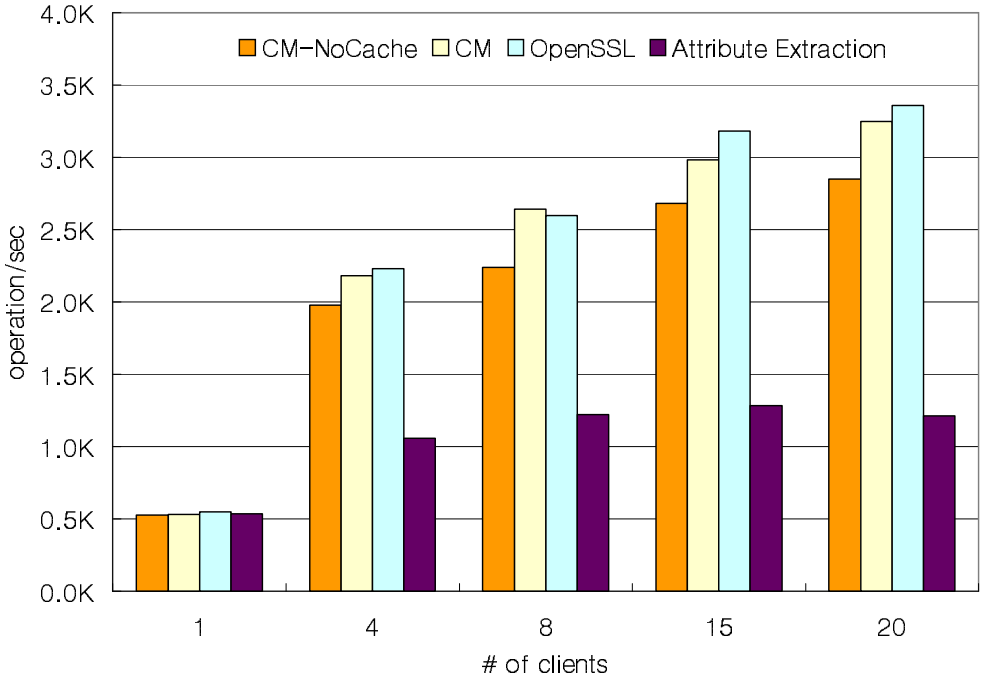
Directory Population Performance

100K entries	Component Matching	OpenSSL	Attribute Extraction
Time (sec)	178	167	815
DB Size(Mbytes)	234	234	410

Directory Search Performance



100k entries, DB cache = 1GBytes



500k entries, DB cache = 200MBtyes

Summary

- **Enhance LDAP-PKI interoperation for flexibility and manageability**
- **Component matching enabled OpenLDAP server**
  - ASN.1 awareness
    - Supports any ASN.1 type from its specification
    - Supports GSER in attribute / assertion values
  - Component Matching
    - Supports searching of any Certificate attributes
  - First implementation of component matching in a pure LDAP server
- **Component optimization techniques**
  - Component aliasing, component indexing, and component caching
  - Enable high-performance, scalable LDAP certificate repository
- **Component matching enabled OpenLDAP will help to use PKC more flexibly in many PKI applications of Internet2, OASIS, GRID…**