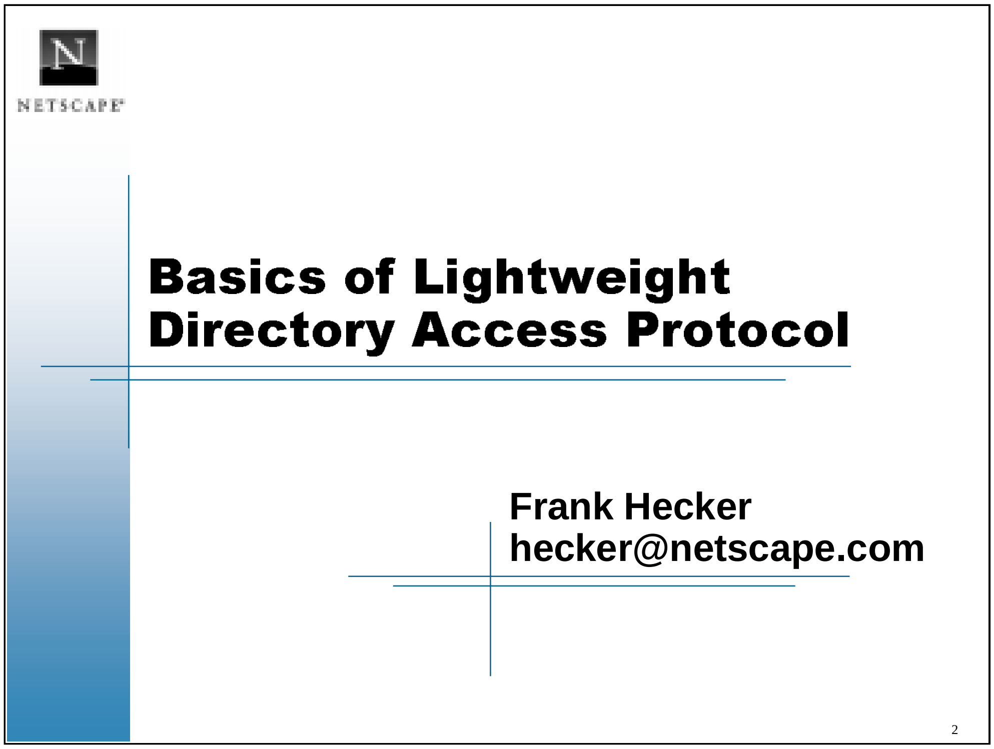
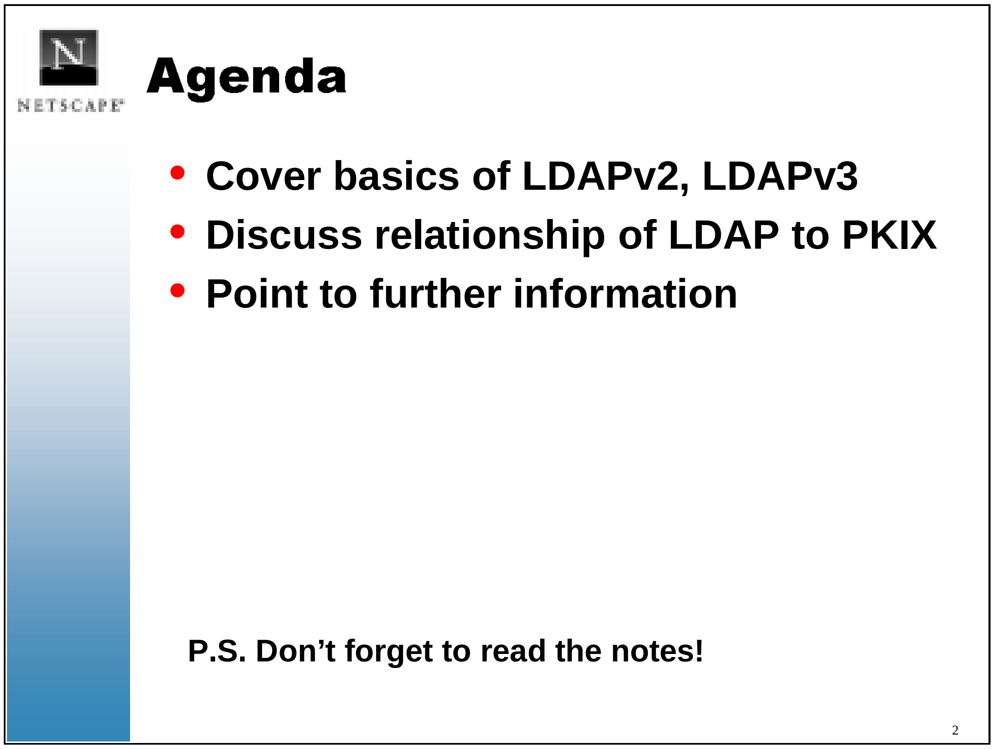


NETSCAPE

# Basics of Lightweight Directory Access Protocol

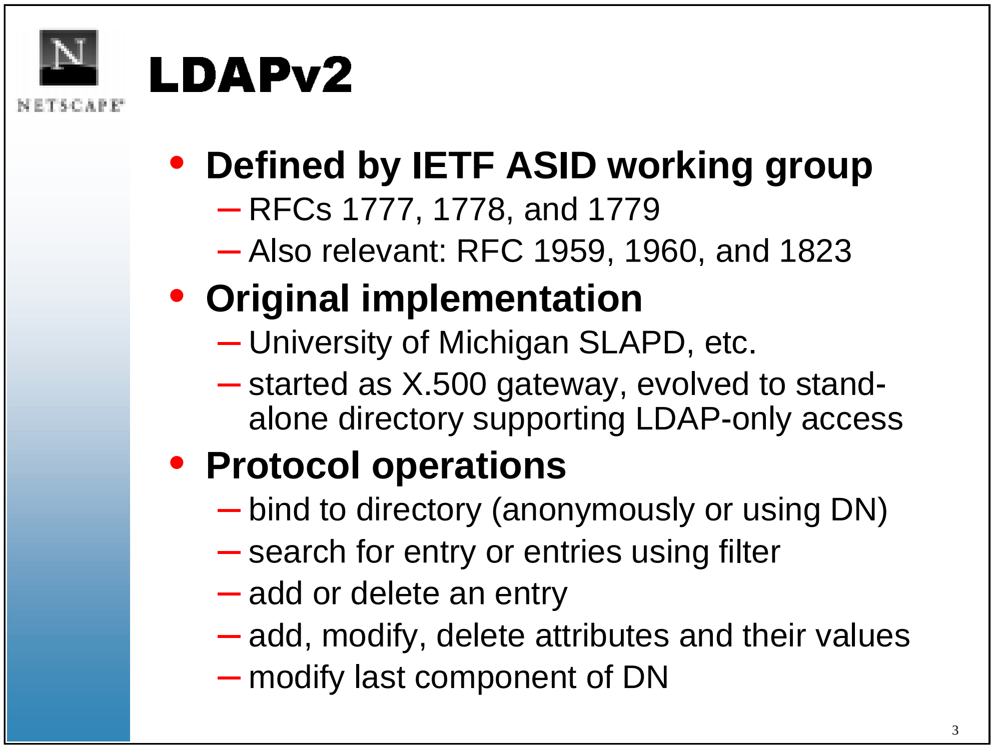
Frank Hecker
hecker@netscape.com



# Agenda

- Cover basics of LDAPv2, LDAPv3
- Discuss relationship of LDAP to PKIX
- Point to further information

P.S. Don’t forget to read the notes!



## LDAPv2

- **Defined by IETF ASID working group**
  - RFCs 1777, 1778, and 1779
  - Also relevant: RFC 1959, 1960, and 1823
- **Original implementation**
  - University of Michigan SLAPD, etc.
  - started as X.500 gateway, evolved to stand-alone directory supporting LDAP-only access
- **Protocol operations**
  - bind to directory (anonymously or using DN)
  - search for entry or entries using filter
  - add or delete an entry
  - add, modify, delete attributes and their values
  - modify last component of DN

RFC 1777 defines the LDAPv2 protocol itself.

RFC 1778 defines the attribute syntaxes, i.e., which attributes are expected to be retrievable from an LDAPv2-compliant directory, and how the values of those attributes are to be represented (encoded) when retrieved from the directory via LDAPv2.

RFC 1779 defines a string (i.e., printable) representation of DNs

RFC 1959 defines “ldap:” URLs, i.e., how an LDAPv2 search operation against a particular directory with particular filter, etc., can be represented as a URL.

RFC 1960 defines a string representation for LDAPv2 search filters.

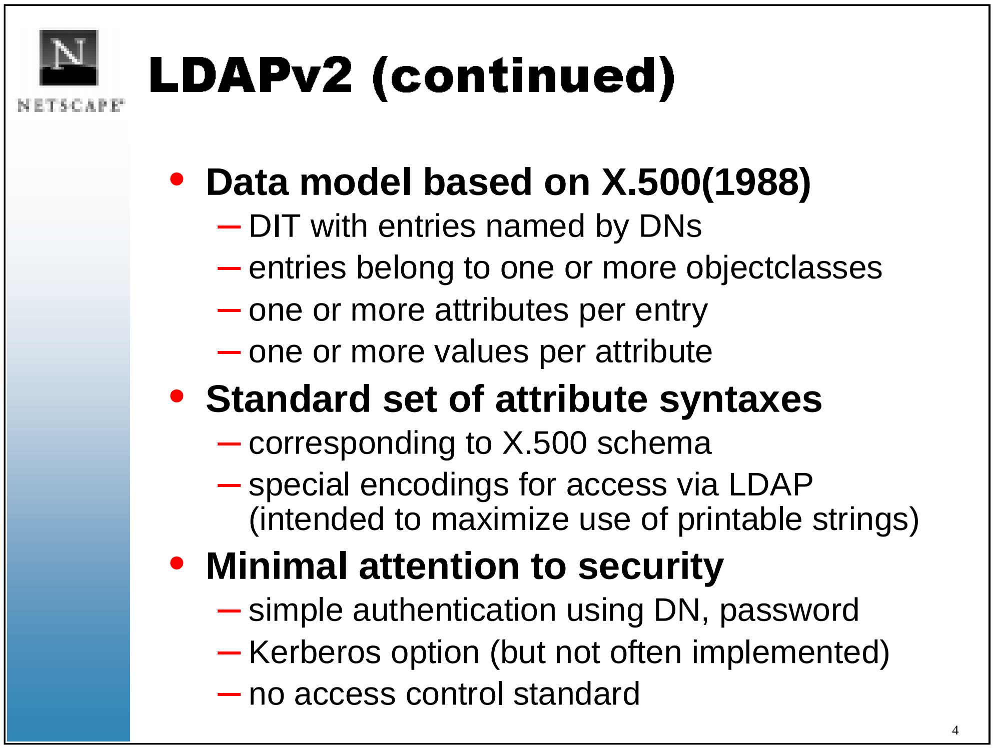
RFC 1823 defines a C API for LDAPv2. This corresponds to the client API in the original University of Michigan implementation.

RFCs 1777, 1778, and 1779 are currently listed as IETF draft standards, although for various reasons they will not be advanced to full standards.

Protocol operations:

Anonymous bind is using a null DN

A filter specifies the search parameters, e.g., look for entries with a value of CN containing “Doe”.



## LDAPv2 (continued)

- **Data model based on X.500(1988)**
  - DIT with entries named by DNs
  - entries belong to one or more objectclasses
  - one or more attributes per entry
  - one or more values per attribute
- **Standard set of attribute syntaxes**
  - corresponding to X.500 schema
  - special encodings for access via LDAP (intended to maximize use of printable strings)
- **Minimal attention to security**
  - simple authentication using DN, password
  - Kerberos option (but not often implemented)
  - no access control standard

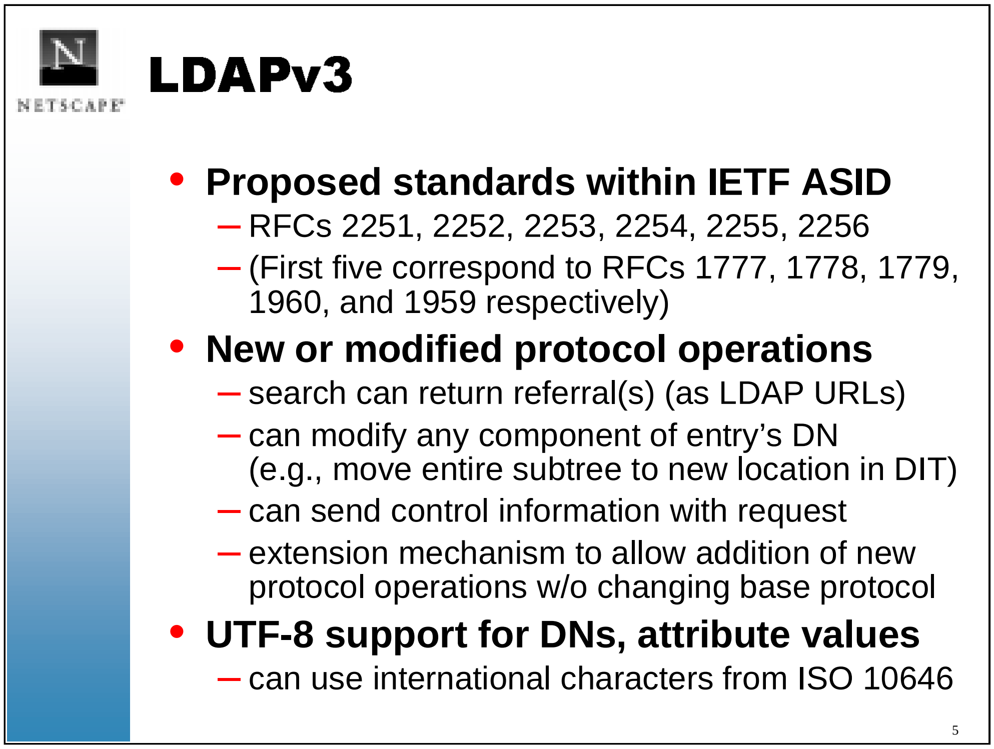
4

Attribute syntaxes:

Note that LDAPv2 places no restrictions on how a directory actually stores values internally; rather it specifies how those values are to be encoded when returned to the user as a result of an LDAPv2 protocol operation.

Security:

Even though there is no LDAPv2 access control standard, existing LDAPv2 products in fact implement access control. Note that access control is transparent to the LDAP client because it is handled in the server and does not change the LDAPv2 protocol itself.



## LDAPv3

- **Proposed standards within IETF ASID**
  - RFCs 2251, 2252, 2253, 2254, 2255, 2256
  - (First five correspond to RFCs 1777, 1778, 1779, 1960, and 1959 respectively)
- **New or modified protocol operations**
  - search can return referral(s) (as LDAP URLs)
  - can modify any component of entry’s DN (e.g., move entire subtree to new location in DIT)
  - can send control information with request
  - extension mechanism to allow addition of new protocol operations w/o changing base protocol
- **UTF-8 support for DNs, attribute values**
  - can use international characters from ISO 10646

5

RFC 2251 defines the LDAPv3 protocol itself.

RFC 2252 defines the attribute syntaxes, i.e., which attributes are expected to be retrievable from an LDAPv3-compliant directory, and how the values of those attributes are to be represented (encoded) when retrieved from the directory via LDAPv3.

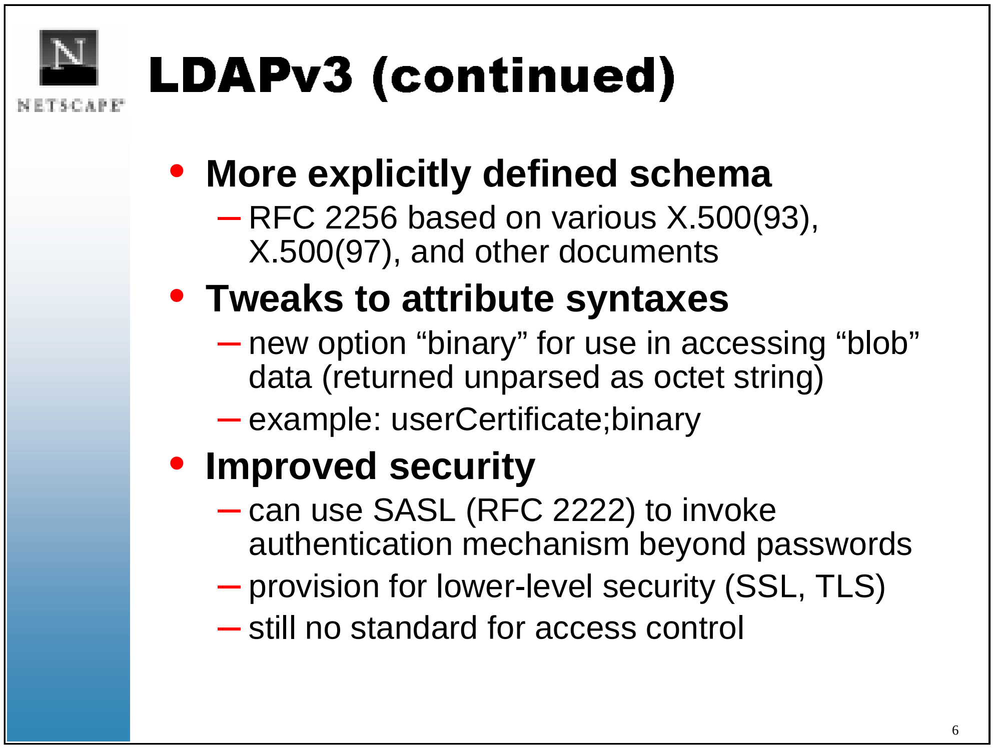
RFC 2253 defines a string (i.e., printable) representation of DNs

RFC 2254 defines a string representation of LDAPv3 search filters.

RFC 2255 defines “ldap:” URLs, i.e., how an LDAPv3 search operation against a particular directory with particular filter, etc., can be represented as a URL.

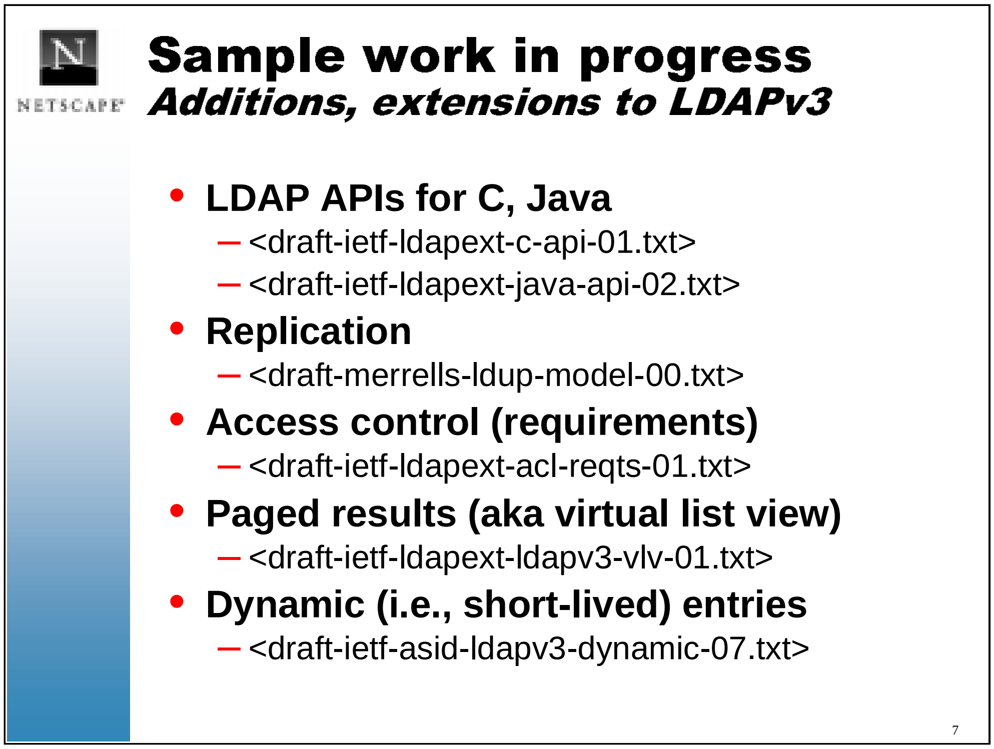
For RFC 2256 see next slide.

RFCs 2251-2256 are currently listed as IETF proposed standards.



# LDAPv3 (continued)

- **More explicitly defined schema**
  - RFC 2256 based on various X.500(93), X.500(97), and other documents
- **Tweaks to attribute syntaxes**
  - new option “binary” for use in accessing “blob” data (returned unparsed as octet string)
  - example: userCertificate;binary
- **Improved security**
  - can use SASL (RFC 2222) to invoke authentication mechanism beyond passwords
  - provision for lower-level security (SSL, TLS)
  - still no standard for access control



# Sample work in progress

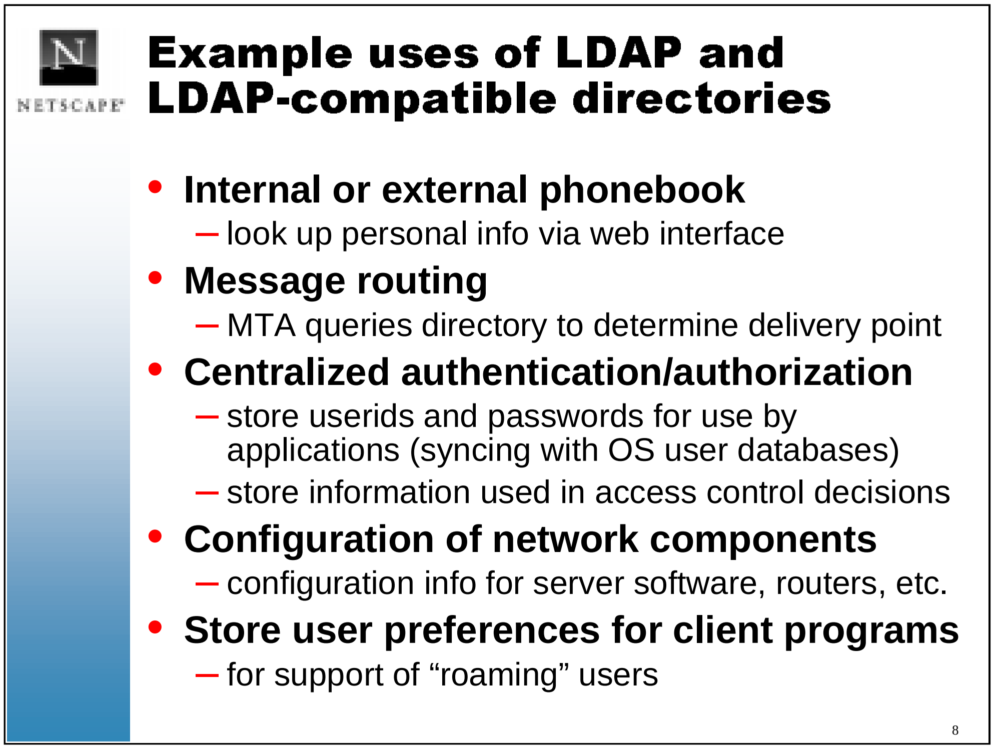
## *Additions, extensions to LDAPv3*

- **LDAP APIs for C, Java**
  - <draft-ietf-ldapext-c-api-01.txt>
  - <draft-ietf-ldapext-java-api-02.txt>
- **Replication**
  - <draft-merrells-ldup-model-00.txt>
- **Access control (requirements)**
  - <draft-ietf-ldapext-acl-reqts-01.txt>
- **Paged results (aka virtual list view)**
  - <draft-ietf-ldapext-ldapv3-vlv-01.txt>
- **Dynamic (i.e., short-lived) entries**
  - <draft-ietf-asid-ldapv3-dynamic-07.txt>

The C API is the LDAPv3 counterpart to that defined in RFC 1823 for LDAPv2.

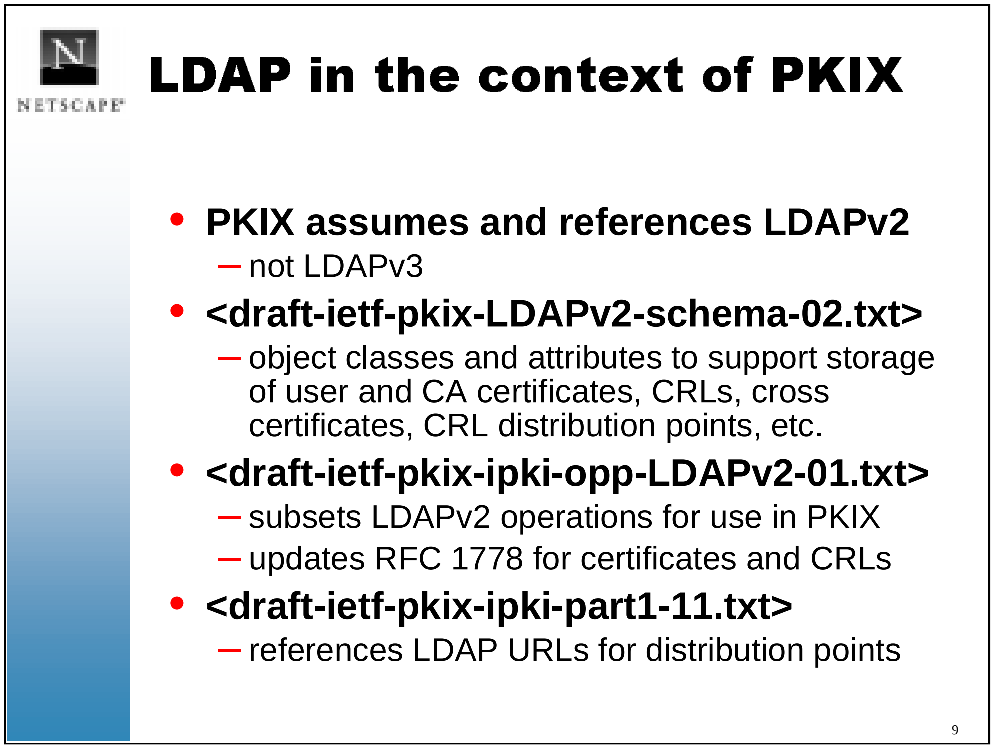
Paged results (often referred to as virtual list view or VLV) is for cases where a client does a search resulting in many entries being returned, and wishes to process them in manageable batches (e.g., displaying them to the user a “page” at a time).

Dynamic entries are intended for such applications as online chat or online conferences where users wish to look up information about other users currently online, meetings currently being held, etc., in a directory, where that information is relatively short-lived and expires at some point.



# Example uses of LDAP and LDAP-compatible directories

- **Internal or external phonebook**
  - look up personal info via web interface
- **Message routing**
  - MTA queries directory to determine delivery point
- **Centralized authentication/authorization**
  - store userids and passwords for use by applications (syncing with OS user databases)
  - store information used in access control decisions
- **Configuration of network components**
  - configuration info for server software, routers, etc.
- **Store user preferences for client programs**
  - for support of “roaming” users



## LDAP in the context of PKIX

- PKIX assumes and references LDAPv2
  - not LDAPv3
- <draft-ietf-pkix-LDAPv2-schema-02.txt>
  - object classes and attributes to support storage of user and CA certificates, CRLs, cross certificates, CRL distribution points, etc.
- <draft-ietf-pkix-ipki-opp-LDAPv2-01.txt>
  - subsets LDAPv2 operations for use in PKIX
  - updates RFC 1778 for certificates and CRLs
- <draft-ietf-pkix-ipki-part1-11.txt>
  - references LDAP URLs for distribution points

9

Note that <draft-ietf-pkix-ipki-opp-LDAPv2-08.txt> is actually stored online as <draft-ietf-pkix-ipki2opp-08.txt>.

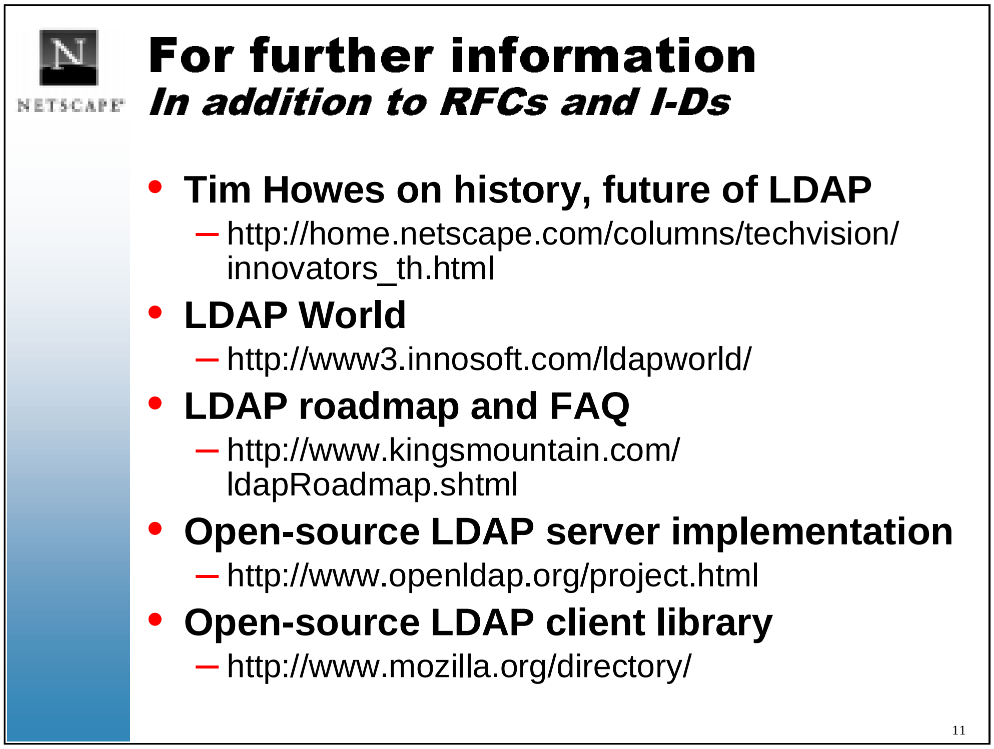
For more details on the updating of RFC 1778 see the next slide.

Note that <draft-ietf-pkix-ipki-part1-11.txt> is incorrect in referencing RFC 1778 in relation to LDAP URLs; LDAP URLs (for LDAPv2) are actually defined in RFC 1959.

NETSCAPE

# The certificate format issue

- **LDAPv2 requires certificates and CRLs to be returned in parsed format (RFC 1778)**
- **PKIX overrides RFC 1778**
  - mandates that values of userCertificate and related attributes be returned encoded as if their values were unparsed octet strings
- **LDAPv3 introduces general “binary” option to handle this case (RFC 2252)**
  - foo;binary returned encoded as if its value were an unparsed octet string, regardless of treatment of foo itself
  - mandates use of userCertificate;binary, etc.
  - no string representation defined for transfer of plain userCertificate, etc., without binary option



# For further information

## *In addition to RFCs and I-Ds*

- **Tim Howes on history, future of LDAP**
  - http://home.netscape.com/columns/techvision/innovators_th.html
- **LDAP World**
  - http://www3.innosoft.com/ldapworld/
- **LDAP roadmap and FAQ**
  - http://www.kingsmountain.com/ldapRoadmap.shtml
- **Open-source LDAP server implementation**
  - http://www.openldap.org/project.html
- **Open-source LDAP client library**
  - http://www.mozilla.org/directory/

Tim Howes was one of the primary inventors of LDAP while at the University of Michigan.

The OpenLDAP project is using the University of Michigan SLAPD code as a base.  This is the same code base used by Netscape for the initial version of Netscape Directory Server.

The Netscape directory client SDK is available for C, Java, and Perl (as PerLDAP).

# Q & A

**Send any remaining questions to hecker@netscape.com**