

**Federal PKI Directory Profile**  
**version 1.0**  
**7/25/2000**

## **1. Introduction**

This profile defines the requirements for the initial operational Federal Public Key Infrastructure (FPKI) directory system. The FPKI builds upon FBCA prototype that was successfully demonstrated during the Electronic Messaging Association (EMA) Challenge in April 2000. This prototype supported S/MIME messaging among several disparate PKI domains using several different CA products, X.500 directory products, and S/MIME e-mail clients. This demonstration illustrated interoperability on several levels – between CAs, between directories, and between e-mail clients. Each client created, then processed a certificate trust path between the domain of the recipient and the domain of the sender in order to validate the signer's digital signature on the e-mail. Trust paths up to seven certificates were constructed and validated. Directories were chained using the X.500 Directory System Protocol (DSP), while the Lightweight Directory Access Protocol (LDAP) was employed by the e-mail client to access its local directory [1].

The FPKI will use a Federal Bridge CA that cross-certifies with agency Principal CAs to provide trust paths between the agencies. A Federal Bridge CA directory server will be chained to agency border directories to make certificates available for PKI users. The Border CA concept is described in [2].

In the following sections, this profile will address the minimum required schema, the naming conventions, the directory protocols to support, alternatives to consider, and issues to bear in mind in order to adapt to this evolving technology. As we attempt to define the directory profile, we also strive to raise issues that are relevant for the working group's discussions and resolutions. Familiarity with the PKI technology, concepts and general terms of the directory service is assumed.

The draft is based on several sections of the following documents:

- *The Evolving Federal Public Key Infrastructure* [1],
- *Governmentwide Directory Support 2 Technical Series, the Updated US Gold Schema document* [3],
- *The Bridge CA Demonstration Repository Requirements Draft 4/8/1999* [4], and
- *NSA Bridge Certification Authority Demonstration Phase II - Directory Requirements and Architecture, 7/3/2000* [5].

## **2. Schema Requirements**

This section addresses the minimum schema requirements for agency directories to interoperate with the FPKI directory. The schema is limited to just the objects needed to support the PKI.

In order to define the minimum schema requirement, it may be desirable to consider the types of applications that will use the PKI and the information these applications may require. At a minimum, the directories are required to store and disseminate the following PKI related attributes:

- Certification Authority Certificates
- Certificate Revocation Lists
- Authority Revocation Lists
- Cross Certificates
- End-entity certificates
- RFC822MailUser
- User password?
- Certificate Policy definitions?
- Certification Practice Statements?

In the Internet X.509 Public Key Infrastructure LDAPv2 Schema [6], these attributes are:

- cACertificate
- certificateRevocationList
- authorityRevocationList
- crossCertificatePair
- userCertificate
- rfc822Mailbox
- userPassword?
- Need attribute for Certificate Policy definitions?
- Need attribute for Certification Practice Statements?

This schema is used in some commercial CA products.

Some agencies may wish to make other information available externally to support their PKI applications. However, this profile does not address or impose requirements on application-specific data in agency directories.

The cACertificate and crossCertificatePair attributes require special attention when accessing the directory to build the certificate path. Neither the PKIX specification nor the X.509 standards explicitly provide an algorithm to construct a certificate path. The PKIX LDAP-V2-schema provides guidance on what can be stored in the specific attributes. The draft states the following about the cACertificate attribute and the crossCertificatePair attribute:

The cACertificate attribute of a CA's directory entry shall be used to store self-issued certificates (if any) and certificates issued to this CA by CAs in the same realm as this CA.

The forward elements of the crossCertificatePair attribute of a CA's directory entry shall be used to store all, except self-issued certificates issued to this CA. Optionally, the reverse elements of the crossCertificatePair attribute, of a CA's directory entry may contain a subset of certificates issued by this CA to other CAs. When both the forward and the reverse elements are present in a single attribute value, issuer name in one certificate shall match the subject name in the other and vice versa, and the subject public key in one certificate shall be capable of verifying the digital signature on the other certificate and vice versa.

When a reverse element is present, the forward element value and the reverse element value need not be stored in the same attribute value; in other words, they can be stored in either a single attribute value or two attribute values.

In the case of V3 certificates, none of the above CA certificates shall include a `basicConstraints` extension with the `cA` value set to `FALSE`.

A path development algorithm must consider that the CA's certificate must be stored in the `crossCertificatePair` attribute, but the algorithm may consult the `cACertificate` attribute first, for performance reasons.

The following sections define the attributes and object classes that are required for end entities and CAs.

## 2.1 End Entities

### Attributes

End entity (EE) directory entries shall contain, as a minimum, the following attributes:

1. *userCertificate* as defined in 1997 X.509 (OID: 2.5.4.36)
2. *attributeCertificate* as defined in 1997 X.509 (OID: 2.5.4.58)
3. *commonName* as defined in 1997 X.521 (OID: 2.5.4.3)
4. *surname* as defined in 1997 X.521 (OID: 2.5.4.4)

**NOTE:** The EE relative distinguished name (RDN) shall consist of the *commonName* attribute type and value. For example: `cn=John Smith`

### Object Classes

EE entries shall be made up of the following object classes:

1. *person* as defined in 1997 X.521 (OID: 2.5.6.6)
2. *pkiUser* as defined in RFC 2587: LDAPv2 Schema (OID: 2.5.6.21) for non-Entrust EEs -- OR -- *entrustUser* as defined in "Entrust Directory Schema Requirements" version 1.0, dated August, 1998 (OID: 1.2.840.113533.7.67.0) for Entrust EEs.
3. *securePkiUser* as defined in ACP 120 dated April 1999 (OID: 2.16.840.1.101.2.2.3.66). This auxiliary object class includes *attributeCertificate* and *supportedAlgorithms* as optional attribute types.

Optionally, EEs may include the following object classes:

1. *organizationalPerson* as defined in 1997 X.521 (OID: 2.5.6.7)
2. *inetOrgPerson* as defined in IETF draft: draft-smith-ldap-inetorgperson-04.txt, dated 31 January 2000 (OID: 2.16.840.1.113730.3.2.2)

## 2.2 Certification Authorities

### Attributes

CA (including PCAs and PAAs) entries in the directory shall contain at a minimum the following attributes:

1. ***commonName*** OR ***organizationalUnit*** as defined in 1997 X.509 (OIDs: 2.5.4.3 and 2.5.4.11 respectively).
2. ***cACertificate*** as defined in 1997 X.509 (OID: 2.5.4.37). As per the LDAPv2 Schema (RFC 2587), the ***cACertificate*** attribute shall be populated as follows:
 

“The ***cACertificate*** attribute of a CA's directory entry shall be used to store self-issued certificates (if any) and certificates issued to this CA by CAs in the same realm as this CA.”
3. ***certificateRevocationList*** as defined in 1997 X.509 (OID: 2.5.4.39)
4. ***crossCertificatePair*** as defined in 1997 X.509 (OID: 2.5.4.40). As per the LDAPv2 Schema (RFC 2587), the ***crossCertificatePair*** shall be populated as follows:
 

“The forward elements of the ***crossCertificatePair*** attribute of a CA's directory entry shall be used to store all, except self-issued certificates issued to this CA. Optionally, the reverse elements of the ***crossCertificatePair*** attribute, of a CA's directory entry may contain a subset of certificates issued by this CA to other CAs. When both the forward and the reverse elements are present in a single attribute value, issuer name in one certificate shall match the subject name in the other and vice versa, and the subject public key in one certificate shall be capable of verifying the digital signature on the other certificate and vice versa.

“When a reverse element is present, the forward element value and the reverse element value need not be stored in the same attribute value; in other words, they can be stored in either a single attribute value or two attribute values.”

CAs entries in the directory may optionally contain the ***authorityRevocationList*** attribute as defined in 1997 X.509 (OID: 2.5.4.38).

**NOTE:** The CA relative distinguished name (RDN) shall consist of either the ***commonName*** attribute type and value or the ***organizationalUnit*** attribute type and value. For example: cn=NSA CA -- OR -- ou=ECA1

### Object Classes

CA entries shall be made up of the following object classes:

1. ***pkiCA*** as defined in RFC 2587: LDAPv2 Schema (OID: 2.5.6.22) for non-Entrust CAs - - OR -- ***entrustCA*** as defined in “Entrust Directory Schema Requirements” version 1.0, dated August, 1998 (OID: 1.2.840.113533.7.67.1) for Entrust CAs.

The base object class of CAs shall be one (or more) of the following:

2. ***person*** as defined in 1997 X.521 (OID: 2.5.6.6)
3. ***organizationalPerson*** as defined in 1997 X.521 (OID: 2.5.6.7)
4. ***inetOrgPerson*** as defined in IETF draft: draft-smith-ldap-inetorgperson-04.txt, dated 31 January 2000 (OID: 2.16.840.1.113730.3.2.2)
5. ***organizationalUnit*** as defined in 1997 X.521 (OID: 2.5.6.5)

### 3. Namespace Control and DIT Structure

Directory Distinguished Names (DN) are assigned to each directory entry to provide a mapping between user-friendly names and entry values. Entries in an X.500 directory are related to each other through a hierarchical Directory Information Tree (DIT) structure. In order to support a government-wide PKI, it is essential to have a government-wide DIT. A Federal Root Directory is needed in order to facilitate chaining and shadowing among agency directories. There is currently no overall operational Federal Root Directory; therefore, the FPKI directory will act as the US government root directory.

This section addresses namespace control and the DIT structure to ensure distinguished naming for the government-wide DIT. It describes the X.500-based USGold DIT, an alternative that forms the DN based on Internet Domain Name, and domain component naming that includes Internet Domain Names with the USGold style names.

### 3.1 USGold X.500 Naming Convention

The USGold document [3] provides a detailed design for an electronic directory that is based on the implementation of International Telecommunications Union (ITU-T) 1993 X.500 Series of Recommendations. The document describes the infrastructure and conventions for supporting the naming and registration of entities within the Government Electronic Directory.

In this design, the Federal Government X.500 name space is defined as those entries in the DIT that are directly subordinate to the U.S. Government in the global DIT. Specific policy is recommended for the naming and registration of first level organizationalUnits, and general guidance is offered for naming and the DIT structure below this first level (L3).

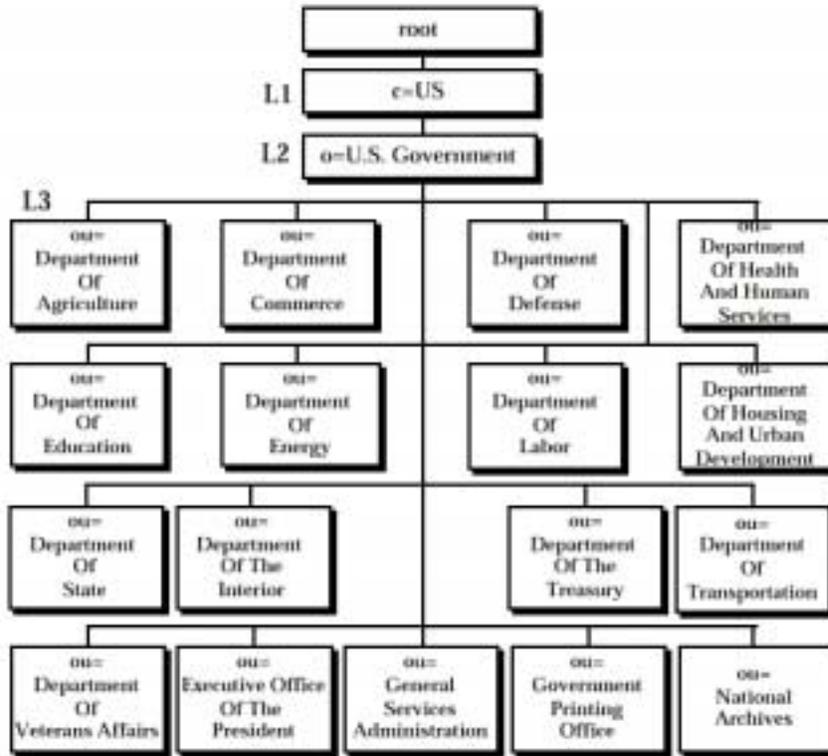
The U.S. Government is registered as an *organization* (O=) object class in the Global DIT. Agencies and departments are registered as *organizationalUnit* (OU=) object classes immediately beneath the O=U.S. Government entry in the Global DIT. US Gold recommended that the source for the registration of agencies and departments be the Federal Government Manual. This publication cites first level agencies and departments (*organizationalUnits*) in each branch of the Federal Government. First level *organizationalUnits* are those that are not subsumed hierarchically under any other *organizationalUnit*.

While ANSI is responsible for the c=US namespace, GSA is responsible for registration of names directly subordinate to c=US, o=U.S. Government. *Question: Is GSA actually doing this????*

US-Gold recommended that the name for the registered entity be the name recorded in the FIPS PUB 95-2 "Codes for the identification of Federal and Federally Assisted Organizations" [7]. The USGold DIT structure, however, does not distinguish between the various categories of Federal agencies defined in FIPS 95-2:

- Legislative Branch;
- Judicial Branch;
- Executive Departments;
- Other Independent Federal Agencies, Boards, Commissions, Councils, Foundations, offices, quasi-federal organizations and federal-state organizations;
- International Organizations.

This is in agreement with the working group's recommendation, for reasons that will become clear in later discussions. An example DIT for the top level of the Government Directory is shown in Figure 3-1.



**Figure 3-1  
Federal Government Top Level DIT**

### 3.1.1 First Level *OrganizationUnits* - Naming Policy

The syntax for the agency name can be:

Department of (name of agency)

Department of the (name of agency)

At the time of registration, one or more alternative names may be designated. For example, the Department of Transportation could designate the following alternative names:

organizationalUnitName = Transportation

organizationalUnitName = DOT.

Any number of organizationalUnitNames may be entered for the registered entity to aid in directory searches; however, only one of the specified names can appear as the relative distinguished name (RDN). The RDN is the name that will be presented to identify the OU to the user as a result of browsing or searching the directory. For example, the Department of Transportation directory entry may have the following names entered into the directory entry:

organizationalUnitName = Department of Transportation (specified RDN)  
 organizationalUnitName = Transportation  
 organizationalUnitName = DOT.

Any search conducted using part or all of the above names will bring up the directory entry listed as the “Department of Transportation”.

To find out the responsibilities of the first level *organizationalUnits*, refer to [3].

### 3.1.2 Second Level and Below *organizationalUnits*

Authority for naming and registration of second level and below *organizationalUnits* will be delegated to all first level OUs. Each first level OU will be responsible for the formation and maintenance of a structure that best defines their organization, operating under the guidelines and recommendations discussed briefly below and more detailed in [3].

#### Directory Distinguished Names

Each entry in the DIT is uniquely identified by its DN. The DN of a given object is defined as being the sequence of the RDNs of the entry that represent the object and those of all of its superior entries in descending order. Each entry has a unique RDN, which is an attribute value that forms the DN of the entry. Agency/department Directory Registration Officials (DROs) are responsible for the registration of *organizationalUnitNames* below the OU level (second level OUs). Individuals (*organizationalPersons*) are listed under the OU subtree for the agency/department, as are *devices* and *applicationEntities* shown in the following example.

c=US, o=U.S.Government, ou=First level organizationalUnits, ou=..., ou=..., cn=Jones, James R (organizationalPerson)  
 c=US, o=U.S.Government, ou=First level organizationalUnits, ou=..., ou=..., cn=DSAname (applicationEntity)  
 c=US, o=U.S.Government, ou=First level organizationalUnits, ou=..., ou=..., cn=security officer (organizationalRole)

The recommended DIT for the Federal Government is shown in Figure 3-2.

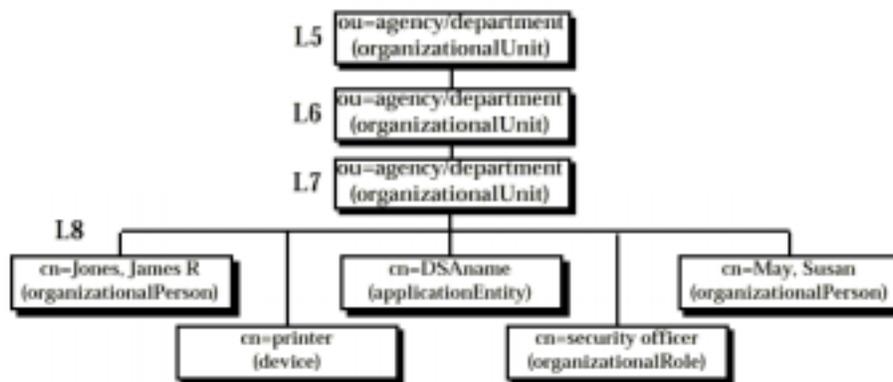


Figure 3-2  
Organizational Subtree

## Naming Convention - *organizationalPerson*

Federal Government person entries should be listed in the agency/departments organizationalUnit name space, allowing for a locality to be designated in the organizational subtree in the Agency DIT for the purposes of uniqueness and unambiguity. The following guidelines are recommended for establishing commonNames for organizational persons (CN=organizational person's name). If possible, the name should comprise "LASTNAME, FIRSTNAME". The use of a middle initial (I) will occur only when required to form a unique name. As a general rule, commonNames will be formed from entries found in the organization's Personnel Records. Each Agency should generate their names and resolve duplicates using the simple rules described in [3].

### 3.1.3 Advantages and Disadvantages of X.500 Naming

The X.500 naming scheme is well understood. It has been supported in current PKI products, which have been successfully demonstrated in the PKI BCA and the EMA challenge demonstrations. Moreover, the USGold schema was used on an international level during the WEMA challenge (*SHC: Can we provide any reference here?*) and it held up pretty well. It did not introduce a lot of new objects but it did recognize the major schemata. The DIT structure and naming conventions described in the USGold document are straightforward and palatable to most agencies. However, the drawback of this naming scheme is that it is not used by anyone other than for PKI. Generally speaking, users do not necessarily understand or care about the finer distinctions of the Federal structure, therefore, distinguished names with organizational structure embedded in them are difficult for users to comprehend or remember. Besides, the more structure that is embedded in names, the more certificates that would need to be revoked when structures change. And the more structure that is built into the names, the more the name space needs to be administered. For this reason, it is a good idea that the USGold DIT does not distinguish between the various categories of Federal agencies as defined in FIPS 95-2.

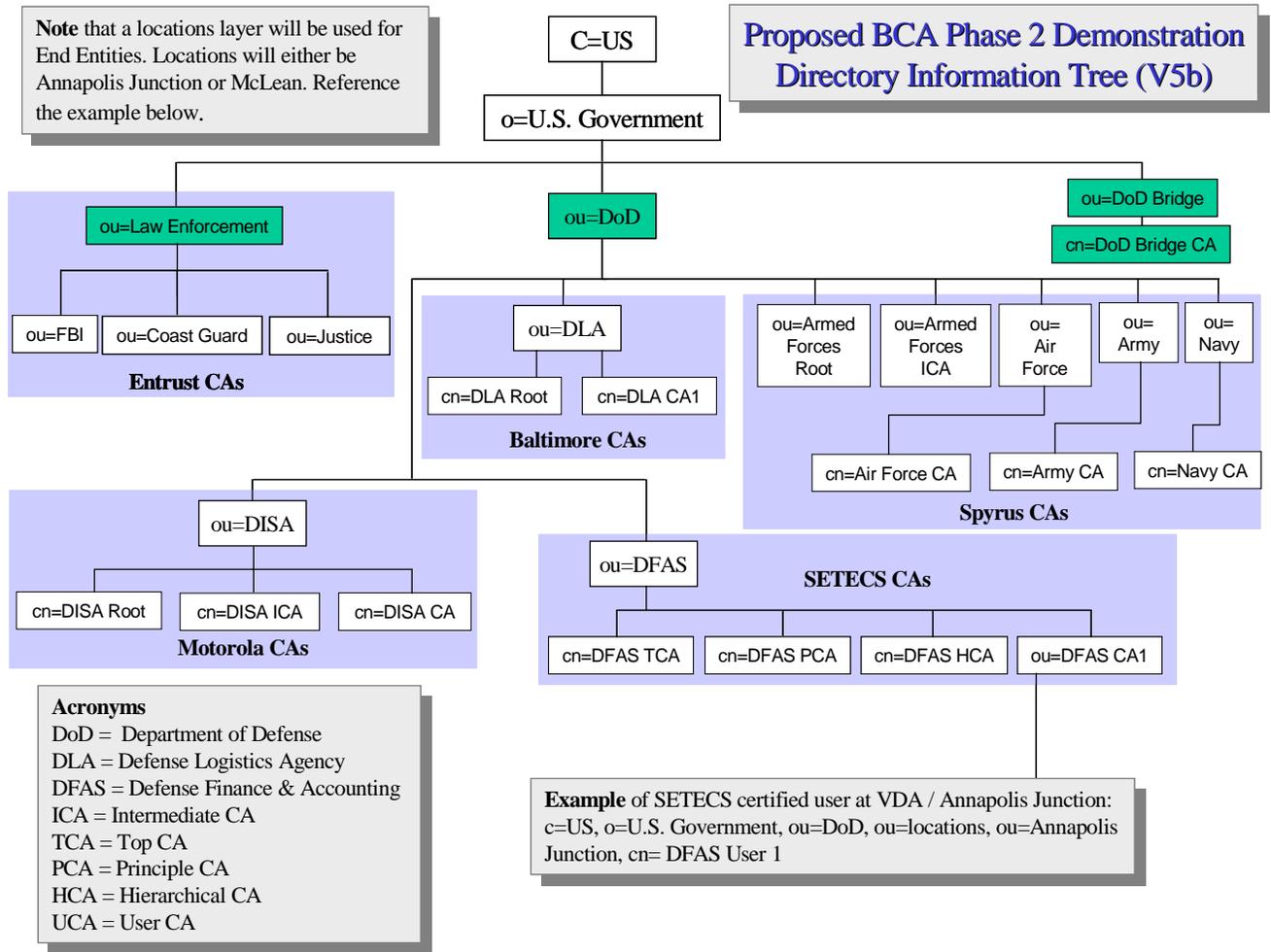
### 3.1.4 Examples of X.500 Naming

Using this naming scheme, the following DNs have been (or planned to be) used for the DOD CAs in its BCA Phase II Demonstration, and the DIT layout is shown in Figure 3-3.

- **Bridge CA:**
  - c=US; o=U.S. Government; ou=DoD Bridge; cn=DoD Bridge CA
- **Law enforcement community Cas:** (*Per Entrust SOP, not identified as a CA in the cn=element, but stored in a CA attribute at the relevant DN:*)
  - CA 1: c=US; o=U.S. Government; ou=Law Enforcement; ou=Justice
  - CA 2: c=US; o=U.S. Government; ou=Law Enforcement; ou=FBI
  - CA 3: c=US; o=U.S. Government; ou=Law Enforcement; ou=Coast Guard
- **Military Services CAs:**
  - Root: c=US; o=U.S. Government; ou=DoD; cn=Armed Forces RootICA: c=US; o=U.S. Government; ou=DoD; cn=Armed Forces ICACA: c=US; o=U.S. Government; ou=DoD; ou=Army; cn=Army CA
  - CA: c=US; o=U.S. Government; ou=DoD; ou=Navy; cn=Navy CA
  - CA (Revoked): c=US; o=U.S. Government; ou=DoD; ou=Air Force; cn=Air Force CADISA
- **CAs:**
  - Root: c=US; o=U.S. Government; ou=DoD; ou=DISA; cn=DISA Root

- ICA: c=US; o=U.S. Government; ou=DoD; ou=DISA; cn=DISA ICACA:  
c=US; o=U.S. Government; ou=DoD; ou=DISA; cn=DISA CADLA **CAs:**
- Root: c=US; o=U.S. Government; ou=DoD; ou=DLA; cn=DLA Root
- CA: c=US; o=U.S. Government; ou=DoD; ou=DLA; cn=DLA CA1
- (Future) CA: c=US; o=U.S. Government; ou=DoD; ou=DLA; cn=DLA CA2
- **DFAS CAs:**
  - TCA: c=US; o=U.S. Government; ou=DoD; ou=DFAS; cn=DFAS TCA
  - PCA: c=US; o=U.S. Government; ou=DoD; ou=DFAS; cn=DFAS PCA
  - HCA: c=US; o=U.S. Government; ou=DoD; ou=DFAS; cn=DFAS HCA
  - CA: c=US; o=U.S. Government; ou=DoD; ou=DFAS; cn=DFAS CA1
  - (Future) CA c=US; o=U.S. Government; ou=DoD; ou=DFAS; cn=DFAS CA2

**Figure 3-3 DoD DIT Layout and Naming Conventions**



### 3.2 Internet Domain Name Based Naming

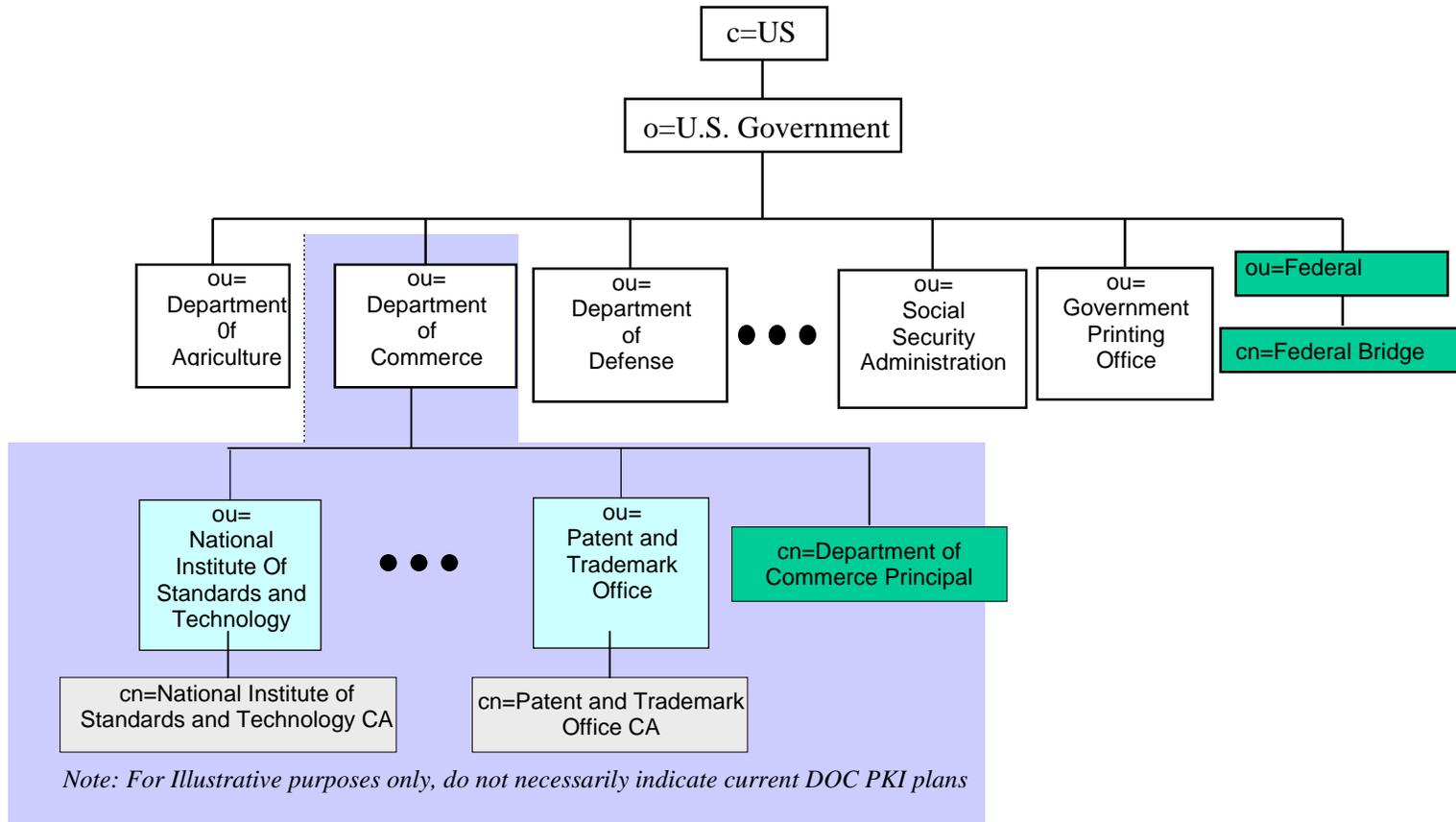
To make the distinguished names more palatable for general users, an alternative scheme was proposed by W. Burr, the WG chair. The idea proposed is that we form a DN based on the combination of the domain name and the email address, such as making *c=US*, *o=.gov*, *.mil*, *.edu* ...etc., *ou=* the higher order parts of the domain name, and *cn=* email address. For example, for a user named John Doe working at NIST, his DN may be *c=US*, *o=.gov*, *ou=.nist.gov*, [\*cn=john.doe@nist.gov\*](mailto:john.doe@nist.gov). (Will provide more examples in later versions.)

Compared with the X.500 based naming shown in Figure 3-4, this alternative, shown in Figure 3-5, has the advantage that it is easier for users to remember the DNs formed using this mechanism. Users are familiar with email addresses, and, in general, have some ideas about what a domain means. This scheme is more manageable, because it can tag along with the existing DNS infrastructure for namespace control without a separate administrative entity. It is also consistent with the basic belief that names should be unique, not changed often, and any information they convey should be simple, clear, widely understood, and well aligned with the names users understand and use.

Among the disadvantages of such a scheme are:

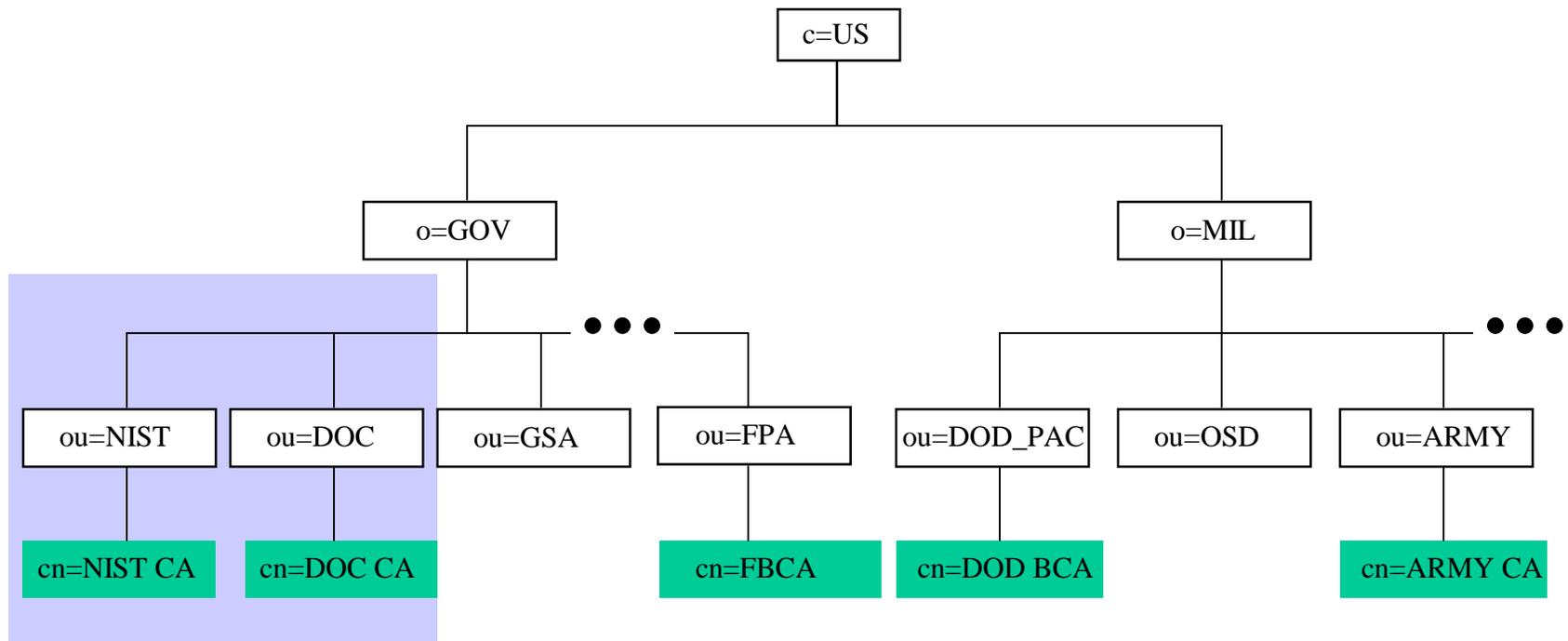
- It does not reflect organizational structure well, that is many agencies domain names do not include their parent organization or department.
- The domain names are usually terse and not necessarily descriptive.
- There is an investment in some agencies in directories with more traditional domain names.

Figure 3-4. X.500-based Federal PKI Directory Information Tree



W. E. Burr  
6 July 2000

Figure 3-5. Internet Domain Name Based DIT



### 3.3 Domain component naming [8]

The Lightweight Directory Access Protocol (LDAP) uses X.500-compatible distinguished names for providing unique identification of entries. The domain component naming defines an algorithm by which a name registered with the Internet Domain Name Service can be represented as an LDAP distinguished name.

The Domain (Nameserver) System (DNS) provides a hierarchical resource labeling system. A name is made up of an ordered set of components, each of which is a short string. An example domain name with two components would be "critical-angle.com".

#### 3.3.1 Mapping Domain Names into Distinguished Names

The mechanism described below provides an enterprise a distinguished name for each domain name it has obtained for use in the Internet. These distinguished names may be used to identify objects in an LDAP directory. However, the mechanism does not define how to represent objects that do not have domain names.

The algorithm for transforming a domain name is to begin with an empty distinguished name and then attach Relative Distinguished Names (RDNs) for each component of the domain, with the most significant component, i.e. closest to the root of the namespace, written last. Each of these RDNs is a single AttributeTypeAndValue, where the type is the attribute "DC" and the value is an IA5 string containing the domain name component. Thus the domain name "CS.UCL.AC.UK" can be transformed into *DC=CS, DC=UCL, DC=AC, DC=UK*

Distinguished names in which there are one or more RDNs, all containing only the attribute type DC, can be mapped back into domain names.

Refer to [8] for the definitions of the additional attributes and object classes, such as *domainComponent* attribute type and the *dcObject*, *domain* object classes that are required to support domain component naming.

An example entry using domain component naming would be:

```
dn: dc=critical-angle,dc=com
objectClass: top
objectClass: organization
objectClass: dcObject
dc: critical-angle
o: Critical Angle Inc.
```

<more discussions to follow>

## 4. Directory Protocols

This section addresses protocol-related issues regarding which directory protocols to support, which to ignore, and which to keep an eye on as the directory technology evolves. For the Directory User Agent (DUA) and the Directory System Agent (DSA) to communicate with each other, there are five primary protocols defined in X.500:

- Directory Access Protocol (DAP) -- client protocol (DUA) which communicate directly with the DSA.
- Lightweight Directory Access Protocol (LDAP) -- A scaled down version of DAP. It provides a subset of DAP functionality.
- Directory System Protocol (DSP) -- DSP is used between DSAs to service user queries that require information that might be distributed over multiple DSAs. The DSP consists of service ports supporting chained operations. Chaining enables a directory information request to be forwarded (through multiple DSAs) to the appropriate DSA containing the requested information. Once the appropriate DSA is identified, the specified operation can then be performed on the target DSA through a DAP operation with attached chaining arguments and chaining results. Chaining is transparent to the user and is performed by progressively forwarding a query through a number of DSAs, each collecting and evaluating results until the data is retrieved and passed back through the “chained” DSAs.
- Directory Operational Binding Protocol (DOP) -- DOP is used between any two DSAs that are entering into an association agreement, either to shadow information or to keep knowledge reference pointers up to date. The DOP enables DSAs to negotiate the nature of the binding agreement and define the parameters that will be used to govern their association.
- Directory Information Shadowing Protocol (DISP) -- Protocol used to transfer replicated data between two or more DSAs.

Among these, our initial focus will be on LDAP (version 2 or 3) for the directory client protocol and DSP for the server-to-server protocol. As directory technology evolves, other alternative protocols will be considered and evaluated. The following section discusses the client and server protocols in more detail.

#### **4.1 Directory Client Protocols**

Recently, LDAP has become the predominant directory client access protocol. Clients will access directories via Lightweight Directory Access Protocol (LDAP) V2 and LDAP V3 ((Request For Comment) RFC1777 and RFC 2251 respectively). The LDAP standard is currently being extended beyond Version II to support new features, some of which are required for the successful operation of a public key infrastructure (PKI). Where applicable, these new features will be incorporated into the FPKI directory. For example, a standard way to post and retrieve binary objects, such as certificates, is now available in LDAP V3. Support for binary transmission of attributes is required for the Federal PKI. Adherence to RFC 2559 (“Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2”) is also required.

<Need to do more research to address the following issues>

- V2 or v3?
- How near is v3 to general commercial use?
- Will there be a PKI LDAPv3 schema?
- LDAP referrals - Is it ready now? How about client support?

## **4.2 Directory Server Protocols**

The directory servers shall support LDAP V2 and LDAP V3 client to server access protocol. Optionally, the servers may support the X.500 Directory Access Protocol (DAP) for client to server access.

Support for ITU X.518 (1993) is required for server-to-server communication (chaining). This standard defines procedures for distributed operations through the Directory System Protocol (DSP). DSP will be used for all server-to-server communications until other viable, commercially available, multi-vendor interoperable technology is developed.

## **4.3 Authentication Requirements**

Directories are required to support simple authentication for LDAP and DSP communications.

### **4.3.1 Client Authentication**

FPKI directory clients that read the FPKI directory (read, list, search directory operations) require no authentication (i.e. anonymous bind to the directory is acceptable). This profile does not address directory access control requirements to update FPKI directory servers. Agencies must ensure that only authorized parties can update FPKI directories.

### **4.3.2 Server Authentication**

FPKI directories are required to support simple authentication for server to server chaining (X.518 DSP) communications.

## **5. References**

- [1] The Evolving Federal Public Key Infrastructure, Federal Public Key Infrastructure Steering Committee, Federal Chief Information Officers Council, [gits-sec.treas.gov](http://gits-sec.treas.gov).
- [2] Burr, W., "Public Key Infrastructure (PKI) Technical Specifications: Part A-Technical Concept of Operations", September 1998
- [3] Governmentwide Directory Support 2 Technical Series, the Updated US Gold Schema document, 7/14/1997, by Booz Allen & Hamilton.
- [4] The Bridge CA Demonstration Repository Requirements Draft 4/8/1999 by Chromatix, Inc.
- [5] NSA Bridge Certification Authority Demonstration Phase II - Directory Requirements and architecture, 7/3/2000, by Entegrity Solutions.
- [6] Boeyen S., Howes T., and P. Richard, "Internet X.509 Public Key Infrastructure LDAPv2 Schema", RFC2587, June 1999.
- [7] FIPS 95-2 "CODES FOR THE IDENTIFICATION OF FEDERAL AND FEDERALLY ASSISTED ORGANIZATIONS", NIST, available from NTIS, Springfield, VA; FIPS-PUB-95-2, date???

- [8] Kille, S., Wahl, M., Grimstad, A., Huber, R., and S. Sataluri, "Using Domains in LDAP Distinguished Names", RFC 2247, January 1998.