```
<entrydatablock> ::= <type> <CRLF> <version> <CRLF> <data>
<type>           ::= "MASTER" | "SLAVE" | "CACHE"
<version>        ::= <printablestring>
<data>           ::= <entry> | <entry> <CRLF> <data>
<entry>          ::= <rdn> <CRLF> <attributelist> <CRLF>
<attributelist>  ::= <attribute> | <attributelist> <CRLF> <attribute>
```

Figure B.3: BNF Used in EDB Files

Appendix C

ASN.1 Summary

This appendix summarises the ASN.1 used (or planned for use) in QUIPU, and incorporates the system naming aspects.

```
QuipuDirectoryDefinitions
DEFINITIONS ::=

BEGIN

IMPORTS
EVERYTHING
   FROM QuipuNameDefinitions

mhs-or-addresses                                                    10
   FROM MHSDirectoryObjectsAndAttributes
         {joint-iso-ccitt mhs-motis(6) arch(5) modules(0) directory(1)}

ABSTRACT-OPERATION, ABSTRACT-ERROR
   FROM AbstractServiceNotation {joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1)}

NameError, ServiceError, SecurityError
   FROM DirctoryAbstractService {joint-iso-ccitt ds(5) modules(1) directoryAbstractService(2)}

DistinguishedName, RelativeDistinguishedName, Attribute, AttributeType,    20
ATTRIBUTE, ATTRIBUTE-SYNTAX, OBJECT-CLASS
   FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1)}
```

```
distinguishedNameSyntax, commonName, description, presentationAddress
   FROM SelectedAttributeTypes {joint-iso-ccitt ds(5) modules(1) selectedAttributeTypes(5)}

top, dsa
   FROM SelectedObjectClasses {joint-iso-ccitt ds(5) modules(1) selectedObjectClasses(6)}

AccessPoint                                                              30
   FROM DistributedOperations {joint-iso-ccitt ds(5) modules(1) distributedOperations(3)} ;

QuipuObject ::= OBJECT-CLASS
        SUBCLASS OF top
        MUST CONTAIN {aCL}
        MAY CONTAIN {lastModifiedBy, lastModifiedTime, entrySecurityPolicy}

QuipuNonLeafObject ::= OBJECTCLASS
        SUBCLASS OF quipuObject
        MUST CONTAIN {masterDSA}                                         40
        MAY CONTAIN {slaveDSA, treeStructure, inheritedAttribute}

ExternalNonLeafObject ::= OBJECTCLASS
        SUBCLASS OF quipuObject
        MAY CONTAIN {subordinateReference, crossReference,
              nonSpecificSubordinateReference}

QuipuDSA ::= OBJECT-CLASS
   SUBCLASS OF dsa
   MUST CONTAIN { aCL, edbInfo, userPassword, manager, quipuVersion}     50
   MAY CONTAIN { description, lastModifiedBy, lastModifiedTime,
        dsaDefaultSecurityPolicy, dsaPermittedSecurityPolicy, relayDSA,
        listenAddress, info }

EDBInfoSyntax ::= SEQUENCE {
        edb         DistinguishedName,
        getFromDSA  DistinguishedName OPTIONAL,
                                  -- If omitted DSA is master
                                  -- Determine mode of update from this DSA
        sendToDSAs  NameList,                                            60
                                  -- Send these DSAs incremental updates
```

```
                                -- Namelist is defined with the ACLs
        getEDBAllowed NameList
                                -- List of DSAs allowed to pull EDB
        }

EdbInfo ::= ATTRIBUTE
   WITH ATTRIBUTE-SYNTAX EDBInfo
   MULTI VALUE

MasterDSA ::= ATTRIBUTE
   WITH ATTRIBUTE-SYNTAX distinguishedNameSyntax
                        -- Master QSR
                        -- Usually, but not necessarily single valued

SlaveDSA ::= ATTRIBUTE
   WITH ATTRIBUTE-SYNTAX distinguishedNameSyntax
                        -- Slave QSR

SubordinateReference ::= ATTRIBUTE
   WITH ATTRIBUTE-SYNTAX AccessPoint
   SINGLE VALUE

CrossReference ::= ATTRIBUTE
   WITH ATTRIBUTE-SYNTAX AccessPoint
   SINGLE VALUE

NonSpecificSubordinateReference ::= ATTRIBUTE
   WITH ATTRIBUTE-SYNTAX AccessPoint

QuipuVersion ::= ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX caseIgnoreStringSyntax

DSAControl ::= ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX caseIgnoreStringSyntax

TreeStructureSyntax ::= SET {
   mandatoryObjectClasses [1] SET OF OBJECT IDENTIFIER,
   optionalObjectClasses [2] SET OF OBJECT IDENTIFIER OPTIONAL,
   permittedRDNs [3] SET OF SET OF AttributeType }
```

```
TreeStructure ::= ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX TreeStructureSyntax
    MULTI VALUE

InheritedListSyntax ::= SET OF CHOICE {
    AttributeType,          -- Take value from the entry
    Attribute }

InheritedAttributeSyntax ::= SET {                                  110
    default [0] InheritedListSyntax OPTIONAL,
        -- default which can be overriden in lower entry.
    always  [1] InheritedListSyntax OPTIONAL,
        -- always present in lower entry.
    objectclass OBJECT IDENTIFIER OPTIONAL
        -- object class to inherit to.
        -- The null case means the objectclass attribute
        --    itself is inherited
}
                                                                    120
InheritedAttribute ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX InheritedAttributeSyntax
    MULTI VALUE

GetEntryDataBlock ABSTRACT-OPERATION
        ARGUMENT GetEntryDataBlockArgument
        RESULT GetEntryDataBlockResult
        ERRORS {NameError,ServiceError,SecurityError}

getEntryDataBlock GetEntryDataBlock ::= 10                          130
                                -- will make this an OBJECT IDENTIFER
                                -- when ISODE can support this form
                                -- of operation code

GetEntryDataBlockArgument ::= SET {
        entry
                [0] DistinguishedName [[p ga_entry]] ,
        CHOICE {
            -- Can only use CHOICE [1] if remote DSA is
```

```
                -- version is QUIPU 6.8 or less.                    140
              sendIfMoreRecentThan [1] EDBVersion,
              getVersionNumber [2] NULL,
              getEDB [3] NULL,
              continuation [4] SEQUENCE {
                  EDBVersion,
                  nextEntryPosition INTEGER }
            },
          maxEntries [5] INTEGER OPTIONAL
                -- Must omit if remote DSA is QUIPU 6.8 or less
}                                                                   150

GetEntryDataBlockResult ::= SEQUENCE {
                  versionHeld [0] EDBVersion,
                  [1] EntryDataBlock OPTIONAL,
                  nextEntryPostion INTEGER OPTIONAL
                -- Must omit if remote DSA is QUIPU 6.8 or less
          }
}

EDBVersion ::= UTCTime                                              160

ACLInfo ::= SET OF SEQUENCE {
        AccessSelector,
        AccessCategories }

AccessCategories ::= ENUMERATED {
        none (0),
        detect (1),
        compare (2),
        read (3),                                                   170
        add (4),
        write (5) }

AccessSelector ::= CHOICE {
        entry [0] NULL,
                -- DUA identified by the entry
        other [2] NULL,
                -- This indicates "public" rights
```

```
        prefix [3] NameList,
                -- This identifies a prefix name for specified DUAs
                -- e.g., anyone in the UK
        group [4] NameList
                -- For specifying group rights
        }

NameList ::= SET OF DistinguishedName

ACLSyntax ::= SEQUENCE {
    childACL            [0] ACLInfo DEFAULT {{other, read}},
    entryACL            [1] ACLInfo DEFAULT {{other, read}},
    defaultAttributeACL [2] ACLInfo DEFAULT {{other, read}},
    [3] SET OF AttributeACL }
                        -- Defaults to a publicly readable
                        -- read only directory

AttributeACL ::= SEQUENCE {
    SET OF AttributeType,
    ACLInfo }

ACL ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ACLSyntax
    SINGLE VALUE

ProtectedPassword ::=
        SEQUENCE {
        algorithm [0] AlgorithmIdentifier OPTIONAL,
        salt [1] SET {
                time1 [0] UTCTime OPTIONAL,
                time2 [1] UTCTime OPTIONAL,
                random1 [2] BIT STRING OPTIONAL,
                random2 [3] BIT STRING OPTIONAL}
                OPTIONAL,
        password [2] OCTET STRING}

END
```

Figure C.1: Summary of ASN.1

Appendix D

The QUIPU Naming Architecture

This appendix defines the QUIPU Naming Architecture. These definitions are needed in order to support the QUIPU mechanism for distributed operations.

This are used in conjunction with the COSINE/Internet schema document [BK91].

```
QuipuNameDefinitions DEFINITIONS ::=

BEGIN

IMPORTS
ABSTRACT-OPERATION, ABSTRACT-ERROR
    FROM AbstractServiceNotation
        {joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1)}

NameError, ServiceError, SecurityError                                  10
    FROM DirctoryAbstractService
        {joint-iso-ccitt ds(5) modules(1) directoryAbstractService(2)}

DistinguishedName, RelativeDistinguishedName, Attribute,
ATTRIBUTE, ATTRIBUTE-SYNTAX, OBJECT-CLASS
    FROM InformationFramework
        {joint-iso-ccitt ds(5) modules(1) informationFramework(1)}
```

```
caseIgnoreStringSyntax
    FROM SelectedAttributeTypes                                          20
        {joint-iso-ccitt ds(5) modules(1) selectedAttributeTypes(5)}

TreeStructure, ACL, EDBInfo, MasterDSA, SlaveDSA, DSA, QuipuVersion,
InheritedAttribute, SubordinateReference, Crossreference,
NonSpecificSubordinateReference,
QuipuDSA, QuipuObject, QuipuNonLeafObject, ExternalNonLeafObject,
    FROM QuipuDirectoryDefinitions;

quipu OBJECT IDENTIFIER ::= {ccitt data(9) pss(2342) ucl(19200300) quipu(99)}
                                -- interim QUIPU OID                     30

attributeType OBJECT IDENTIFIER ::= {quipu attributeType(1)}

attributeSyntax OBJECT IDENTIFIER ::= {quipu attributeSyntax(2)}

objectClass OBJECT IDENTIFIER ::= {quipu objectClass(3)}

treeStructure TreeStructure ::= {attributeType 1}

acl ACL ::= {attributeType 2}                                            40

eDBInfo EDBInfo ::= {attributeType 3}

masterDSA MasterDSA ::= {attributeType 4}

slaveDSA SlaveDSA ::= {attributeType 5}

control DSAControl ::= {attributeType 15}

quipuVersion QuipuVersion ::= {attributeType 16}                         50

protectedPassword ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX    ProtectedPassword
        ::= {attributeType 17}

SecurityPolicy ::= ANY
```