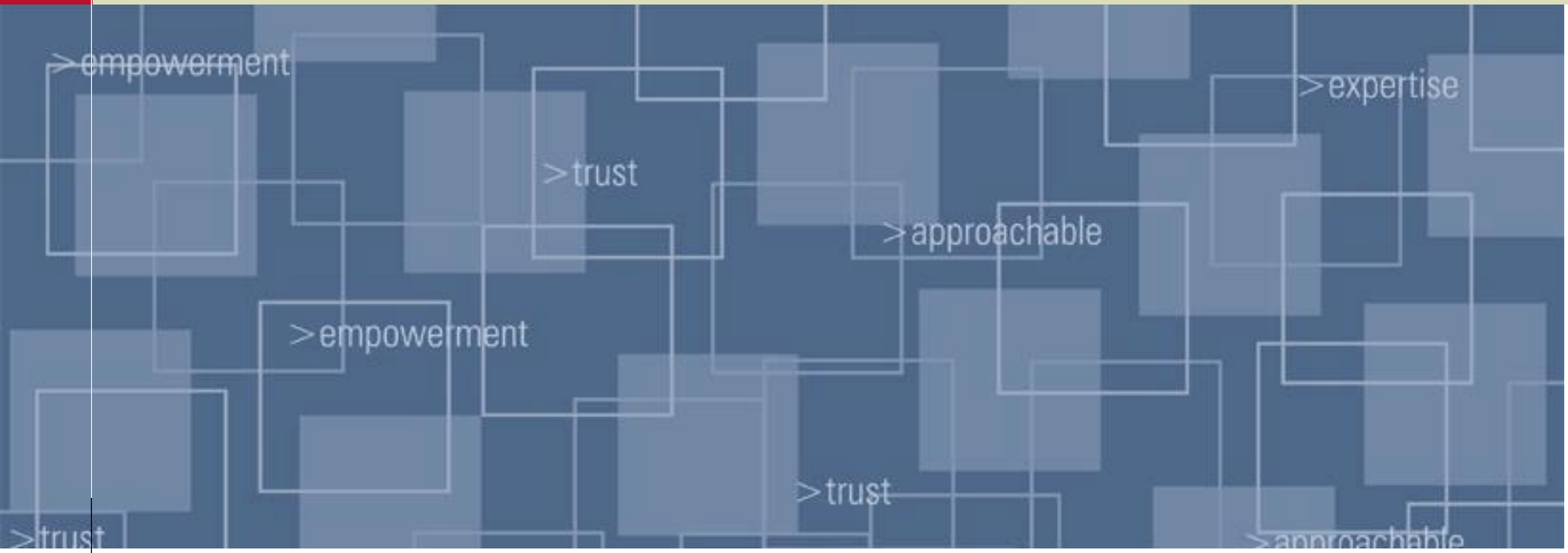
Digital Certificates – GSE 2007

Nigel Pentland



Clydesdale Bank Yorkshire Bank National

Let’s consider a Server certificate

- What are we trying to achieve?
- How will we generate the certificates?
- What will they look like when produced?
- Finally, how will they be used?

Generate a certificate

What do we really mean by generate a certificate?

It is essentially a 2 step process.

First, generate an RSA **key pair**, i.e. a public key and a private key

Second, take the public key, plus additional info, package and **sign** to produce public certificate

Notation

I tend to be pedantic when talking about certificates, keys, etc

I always prefer the following terminology for clarity

- public certificate
- private key

Certificate generation

Certificate generation is sometimes seen as a single step, or as multiple steps

The RACDCERT command can accommodate both methodologies, e.g.

either

RACDCERT GENCERT

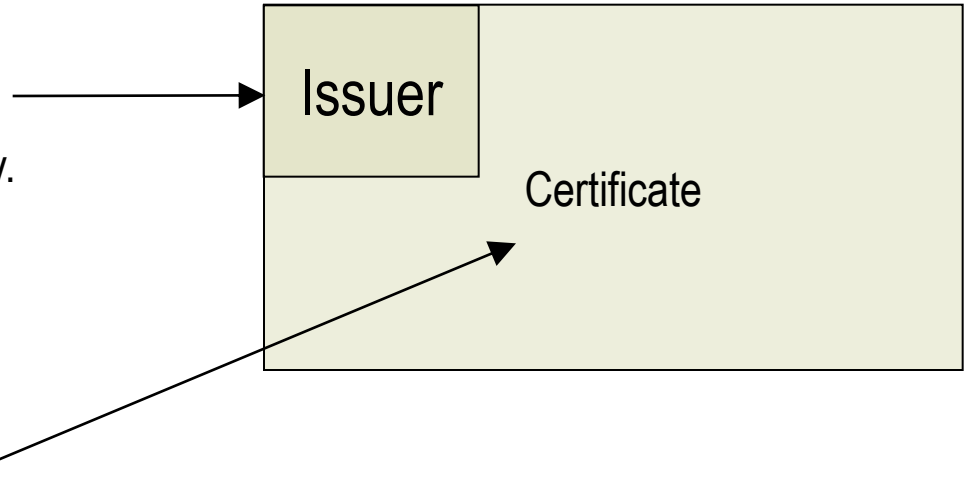
or

RACDCERT GENREQ

RACDCERT GENCERT (‘DATA.SET.NAME’)

Certificate basics

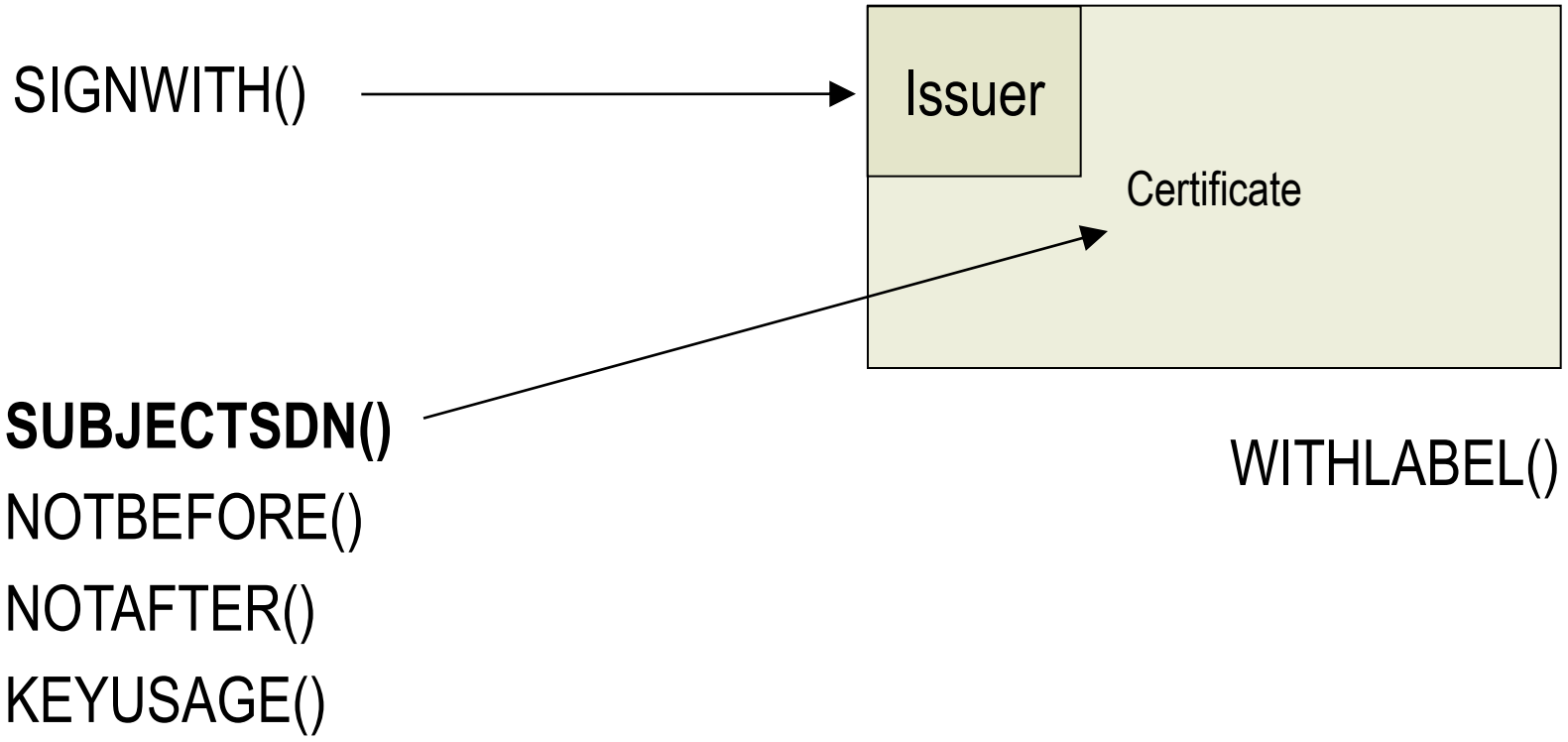
The certificate details are signed by
the issuer, normally referred to as CA,
or as IBM would say, Certificate Authority.



Generally, the main fields are

- Distinguished Name (DN)
- Expiry date
- Keyusage

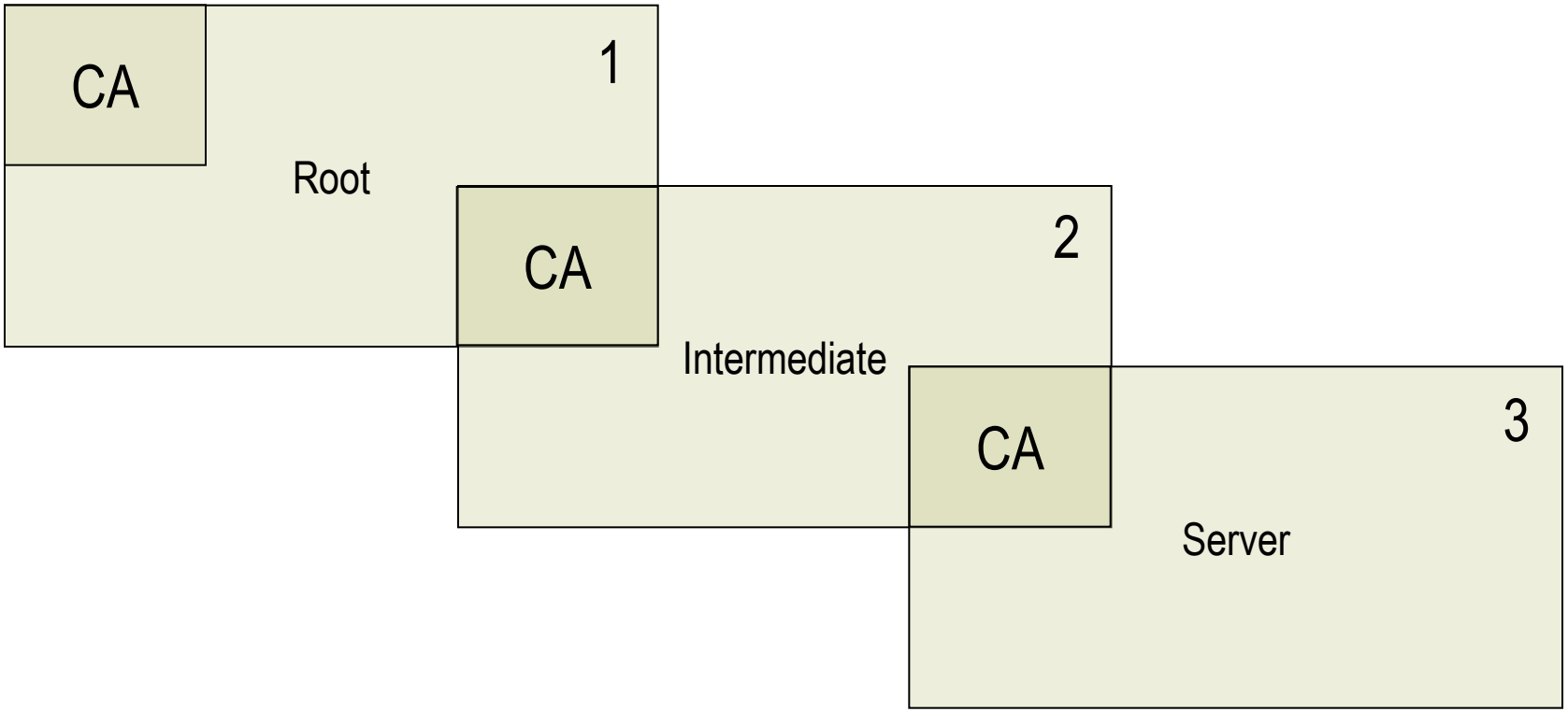
RACDCERT basics



SUBJECTSDN()

```
GENCERT [(request-data-set-name)]
    [ SUBJECTSDN( [ CN('common-name')]
            [ T('title')]
            [ OU('organizational-unit-name1'
                        [ , 'organizational-unit-name2', ... ] )]
            [ O ('organization-name')]
            [ L ('locality')]
            [ SP ('state-or-province')]
            [ C ('country') ]) ]
```

Pictorial Representation



Generate certificate 1

```
RACDCERT CERTAUTH GENCERT +
 SUBJECTSDN(CN('GSE demo root') +
            OU('RACF Group') +
             O('Guide Share Europe') +
             C('GB')) +
 SIZE(1024) +
 NOTBEFORE(DATE(2007-05-30)) +
 NOTAFTER(DATE(2027-05-30)) +
 WITHLABEL('GSE-ROOT') +
 KEYUSAGE(CERTSIGN)
```

RACF limitations

RACDCERT command has following limitations :

```
WITHLABEL('GSE-ROOT')
```

maximum length of label is **32** characters

```
SUBJECTSDN(CN('GSE demo root') +
              OU('RACF Group') +
               O('Guide Share Europe') +
               C('GB')) +
```

maximum length of any single element of DN is **64** characters

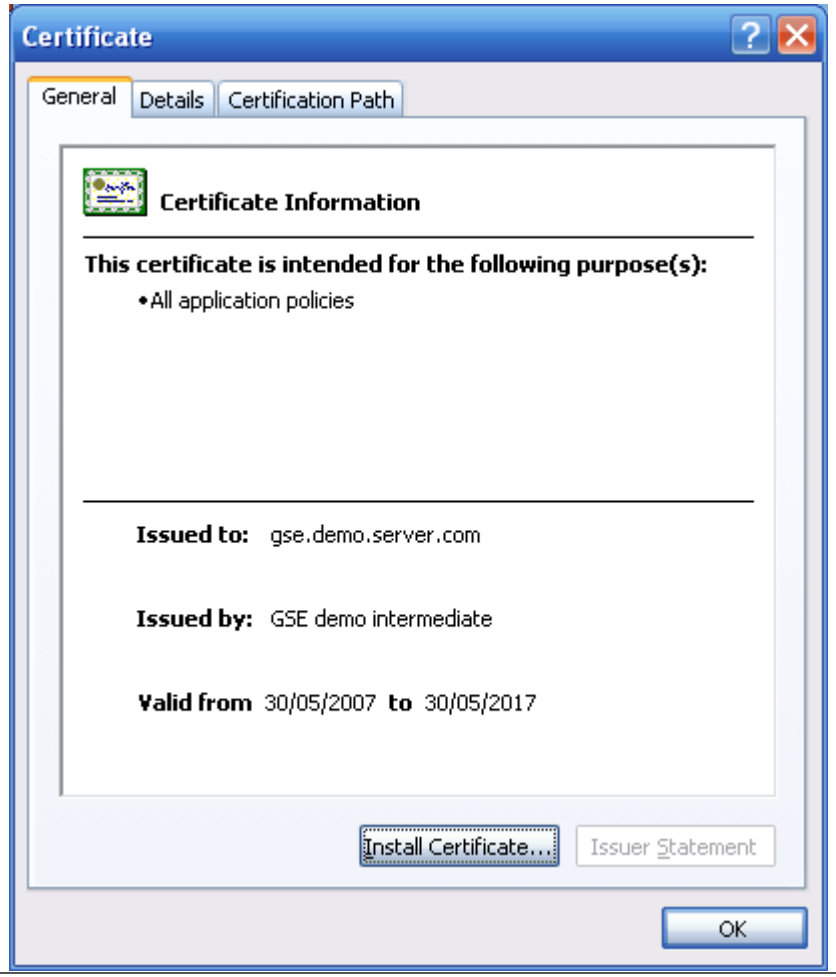
Generate certificate 2

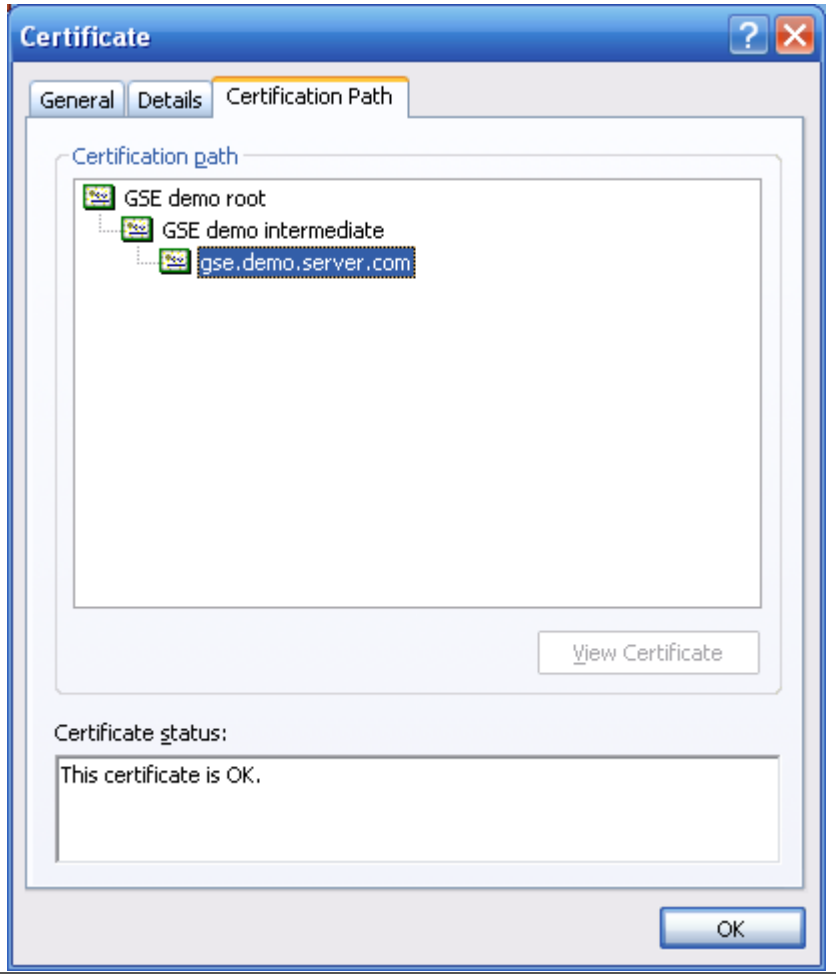
```
RACDCERT CERTAUTH GENCERT +
 SUBJECTSDN(CN('GSE demo intermediate') +
            OU('RACF Group') +
             O('Guide Share Europe') +
             C('GB')) +
 SIZE(1024) +
 NOTBEFORE(DATE(2007-05-30)) +
 NOTAFTER(DATE(2027-05-30)) +
 WITHLABEL('GSE-INTERMEDIATE') +
 SIGNWITH(CERTAUTH LABEL('GSE-ROOT')) +
 KEYUSAGE(CERTSIGN)
```

Generate certificate 3

```
RACDCERT ID(GSECERT) GENCERT +
 SUBJECTSDN(CN('gse.demo.server.com') +
            OU('RACF Group') +
             O('Guide Share Europe') +
             C('GB')) +
 SIZE(1024) +
 NOTBEFORE(DATE(2007-05-30)) +
 NOTAFTER(DATE(2017-05-30)) +
 WITHLABEL('GSE-SERVER') +
 SIGNWITH(CERTAUTH LABEL('GSE-INTERMEDIATE')) +
 KEYUSAGE(HANDSHAKE,DATAENCRYPT)
```

So, let's see what we've generated





Wait a minute, how did we just view that?

First export the certificate chain (caveat – ICSF)

```
RACDCERT ID(GSECERT) +
         EXPORT(LABEL('GSE-SERVER')) +
                DSN('HLQ.DER.GSE3') +
                FORMAT(PKCS12DER) +
                PASSWORD('GSE')
```

RACF PKCS12

RACDCERT always exports the full certificate chain, so in our example the PKCS12 will contain the following components

- Root public certificate
- Intermediate public certificate
- Server public certificate
- Server private key

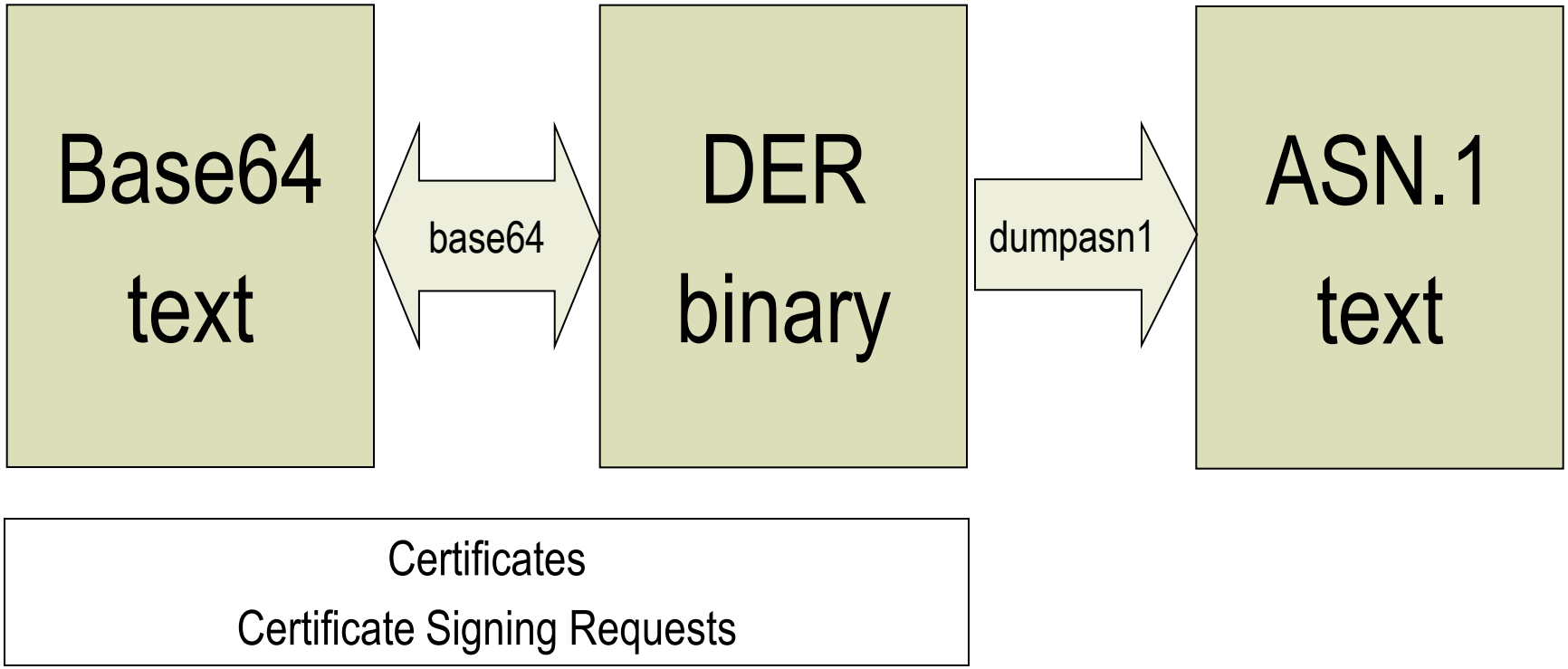
Available RACF export formats

CERTB64	Specifies a DER encoded X.509 certificate that has been encoded using Base64.
CERTDER	Specifies a DER encoded X.509 certificate.
PKCS7B64	Specifies a DER encoded PKCS #7 package that has been encoded using Base64.
PKCS7DER	Specifies a DER encoded PKCS #7 package.
PKCS12B64	Specifies a DER encoded PKCS #12 package that has been encoded using Base64.
PKCS12DER	Specifies a DER encoded PKCS #12 package.

PKCS12 (aka pfx)

- Binary DER (Distinguished Encoding Rules) format
  - ASN.1 (Abstract Syntax Notation) is human readable equivalent
- Ways of viewing
  - **certmgr.msc** **Microsoft (Windows)**
  - certutil Microsoft (SDK)
  - **base64** **John Walker**
  - **dumpasn1** **Peter Gutmann**
  - **OpenSSL** **www.openssl.org**

Pictorial


Base64
text
base64
DER
binary
dumpasn1
ASN.1
text
Certificates
Certificate Signing Requests

Graphic

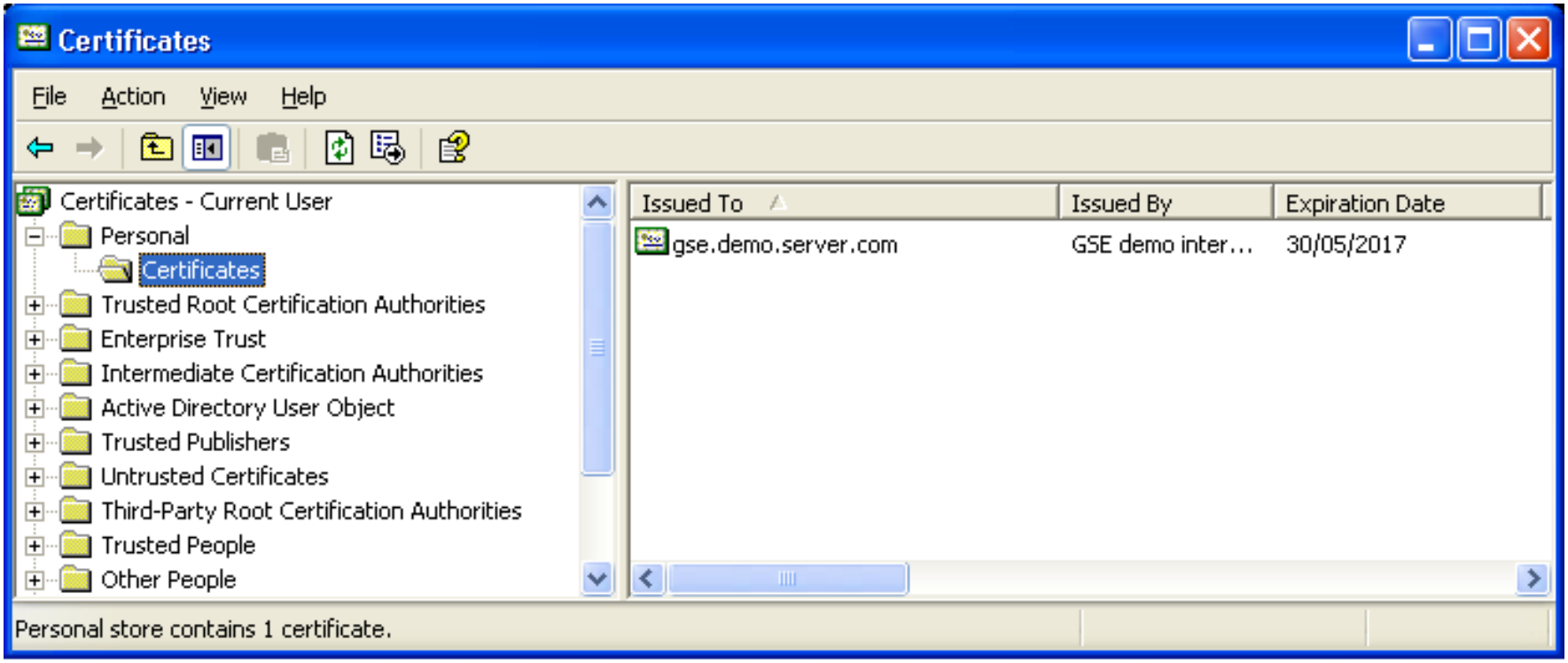
Base64

```
-----BEGIN CERTIFICATE-----
MIICxzCCAjCgAwIBAgIBATANBgkqhkiG9w0BAQUFADB
fMQswCQYDVQQGEwJHQjEbMBkGA1UEChMSR3VpZGUgU2
hhcmUgRXVyb3BlMRMwEQYDVQQLEwpSQUNGIEdyb3VwM
R4wHAYDVQQDExVHU0UgZGVtbyBpbnRlcm1lZGlhdGUw
HhcNMDcwNTI5MjMwMDAwWhcNMTcwNTMwMjI1OTU5WjB
dMQswCQYDVQQGEwJHQjEbMBkGA1UEChMSR3VpZGUgU2
hhcmUgRXVyb3BlMRMwEQYDVQQLEwpSQUNGIEdyb3VwM
RwwGgYDVQQDExNnc2UuZGVtby5zZXJ2ZXIuY29tMIGf
MA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCZiSeAuUy
F15hQ52DjLlL35z+o0LXlVyhnv0/eUsp8tN+Jnac87W
mWXYAwnJOEkzNaR7k0SIOmpnjebxMqFyjyCAkkKr/9m
ap9PPAvcCwADo15QfvFX3LRDFlOCnLD9F0NQPjqok8J
/hHIUhShqtw+YgfDL8htKdNYfT0ld2RsnwIDAQABo4G
UMIGRMD8GCWCGSAGG+EIBDQQyEzBHZW5lcmF0ZWQgYn
kgdGhlIFNlY3VyaXR5IFNlcnZlciBmb3Igei9PUyAoU
kFDRikwDgYDVR0PAQH/BAQDAgSwMB0GA1UdDgQWBBTV
LGhuoSPEv7Aq4QkrbSAShIIIfTAfBgNVHSMEGDAWgBQ
FIQVUDKGczdSKE4ExWVYArtbbBzANBgkqhkiG9w0BAQ
UFAAOBgQAxE+i9YGvf/jviTLZ0ZZ7zZWPbGG31N2SZL
HEbCFjJZ1IIZNspGgFj4Rac1kVbGLK3ZfrP4/8vaPjp
MvKgWo/Q/YnIls9WjmzKd4kunH28Gw6n8kkDgbG4uU0
FqZb+VeDafhqZqjsnYMO6cRcVq2syciTkJa3amdDLYs
RsOuhcDg==
-----END CERTIFICATE-----
```

ASN.1

```
<snip>
  46   27:       SET {
  48   25:         SEQUENCE {
  50    3:           OBJECT IDENTIFIER organizationName (2 5 4 10)
  55   18:           PrintableString 'Guide Share Europe'
         :           }
         :         }
  75   19:       SET {
  77   17:         SEQUENCE {
  79    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
  84   10:           PrintableString 'RACF Group'
         :           }
         :         }
  96   30:       SET {
  98   28:         SEQUENCE {
 100    3:           OBJECT IDENTIFIER commonName (2 5 4 3)
 105   21:           PrintableString 'GSE demo intermediate'
         :           }
         :         }
         :       }
 128   30:     SEQUENCE {
 130   13:       UTCTime 29/05/2007 23:00:00 GMT
 145   13:       UTCTime 30/05/2017 22:59:59 GMT
         :       }
<snip>
```

certmgr.msc


Certificates
File
Action
View
Help
Certificates - Current User
Personal
Certificates
Trusted Root Certification Authorities
Enterprise Trust
Intermediate Certification Authorities
Active Directory User Object
Trusted Publishers
Untrusted Certificates
Third-Party Root Certification Authorities
Trusted People
Other People
Issued To
Issued By
Expiration Date
gse.demo.server.com
GSE demo inter...
30/05/2017
Personal store contains 1 certificate.

certutil

```
402.203.0: 0x80070057 (WIN32: 87): ..CertCli Version
X509 Certificate:
Version: 3
Serial Number: 01
Signature Algorithm:
    Algorithm ObjectId: 1.2.840.113549.1.1.5 sha1RSA
    Algorithm Parameters:
    05 00
Issuer:
    CN=GSE demo intermediate
    OU=RACF Group
    O=Guide Share Europe
    C=GB

NotBefore: 30/05/2007 00:00
NotAfter: 30/05/2017 23:59

Subject:
    CN=gse.demo.server.com
    OU=RACF Group
    O=Guide Share Europe
    C=GB
```

```
Public Key Algorithm:
    Algorithm ObjectId: 1.2.840.113549.1.1.1 RSA
    Algorithm Parameters:
    05 00
Public Key Length: 1024 bits
Public Key: UnusedBits = 0
    0000  30 81 89 02 81 81 00 99  89 27 80 b9 4c 85 d7 98
    0010  50 e7 60 e3 2e 52 f7 e7  3f a8 d0 b5 e5 57 28 67
    0020  bf 4f de 52 ca 7c b4 df  89 9d a7 3c ed 69 96 5d
    0030  80 30 9c 93 84 93 33 5a  47 b9 34 48 83 a6 a6 78
    0040  de 6f 13 2a 17 28 f2 08  09 24 2a bf fd 99 aa 7d
    0050  3c f0 2f 70 2c 00 0e 8d  79 41 fb c5 5f 72 d1 0c
    0060  59 4e 0a 72 c3 f4 5d 0d  40 f8 ea a2 4f 09 fe 11
    0070  c8 52 14 a1 aa dc 3e 62  07 c3 2f c8 6d 29 d3 58
    0080  7d 3d 25 77 64 6c 9f 02  03 01 00 01
Certificate Extensions: 4
    2.16.840.1.113730.1.13: Flags = 0, Length = 32
    Netscape Comment
        Generated by the Security Server for z/OS (RACF)

    2.5.29.15: Flags = 1(Critical), Length = 4
    Key Usage
        Digital Signature, Key Encipherment, Data Encipherment (b0)
```

dumpasn1

```
175   27:       SET {
177   25:         SEQUENCE {
179    3:           OBJECT IDENTIFIER organizationName (2 5 4 10)
184   18:           PrintableString 'Guide Share Europe'
       :           }
       :         }
204   19:       SET {
206   17:         SEQUENCE {
208    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
213   10:           PrintableString 'RACF Group'
       :           }
       :         }
225   28:       SET {
227   26:         SEQUENCE {
229    3:           OBJECT IDENTIFIER commonName (2 5 4 3)
234   19:           PrintableString 'gse.demo.server.com'
       :           }
       :         }
       :       }
255  159:     SEQUENCE {
258   13:       SEQUENCE {
260    9:         OBJECT IDENTIFIER rsaEncryption (1 2 840 113549 1 1 1)
271    0:         NULL
       :         }
273  141:       BIT STRING, encapsulates {
277  137:         SEQUENCE {
280  129:           INTEGER
       :             00 99 89 27 80 B9 4C 85 D7 98 50 E7 60 E3 2E 52
```

OpenSSL

```
Bag Attributes: <No Attributes>
subject=/C=GB/O=Guide Share Europe/OU=RACF Group/CN=GSE demo root
issuer=/C=GB/O=Guide Share Europe/OU=RACF Group/CN=GSE demo root
-----BEGIN CERTIFICATE-----
MIICqTCCAhKgAwIBAgIBADANBgkqhkiG9w0BAQUFADBXMQswCQYDVQQGEwJHQjEb
-----END CERTIFICATE-----
Bag Attributes: <No Attributes>
subject=/C=GB/O=Guide Share Europe/OU=RACF Group/CN=GSE demo intermediate
issuer=/C=GB/O=Guide Share Europe/OU=RACF Group/CN=GSE demo root
-----BEGIN CERTIFICATE-----
MIIC0jCCAjugAwIBAgIBAjANBgkqhkiG9w0BAQUFADBXMQswCQYDVQQGEwJHQjEb
-----END CERTIFICATE-----
Bag Attributes
    friendlyName: GSE-SERVER
    localKeyID: 00 00 00 01
subject=/C=GB/O=Guide Share Europe/OU=RACF Group/CN=gse.demo.server.com
issuer=/C=GB/O=Guide Share Europe/OU=RACF Group/CN=GSE demo intermediate
-----BEGIN CERTIFICATE-----
MIICxzCCAjCgAwIBAgIBATANBgkqhkiG9w0BAQUFADBfMQswCQYDVQQGEwJHQjEb
-----END CERTIFICATE-----
Bag Attributes
    friendlyName: GSE-SERVER
    localKeyID: 00 00 00 01
Key Attributes: <No Attributes>
-----BEGIN RSA PRIVATE KEY-----
MIICWwIBAAKBgQCZiSeAuUyF15hQ52DjLlL35z+o0LXlVyhnv0/eUsp8tN+Jnac8
-----END RSA PRIVATE KEY-----
```

OpenSSL command

```
openssl
 pkcs12
 -in gse3.pfx
 -out gse3.pem
 -nodes
```

OpenSSL as diagnostic tool

Server.bat

openssl s_server -accept 443 -cert server.cer -key server.key -CAfile ca.cer -state -WWW

Browser URL

https:/127.0.0.1/index.htm

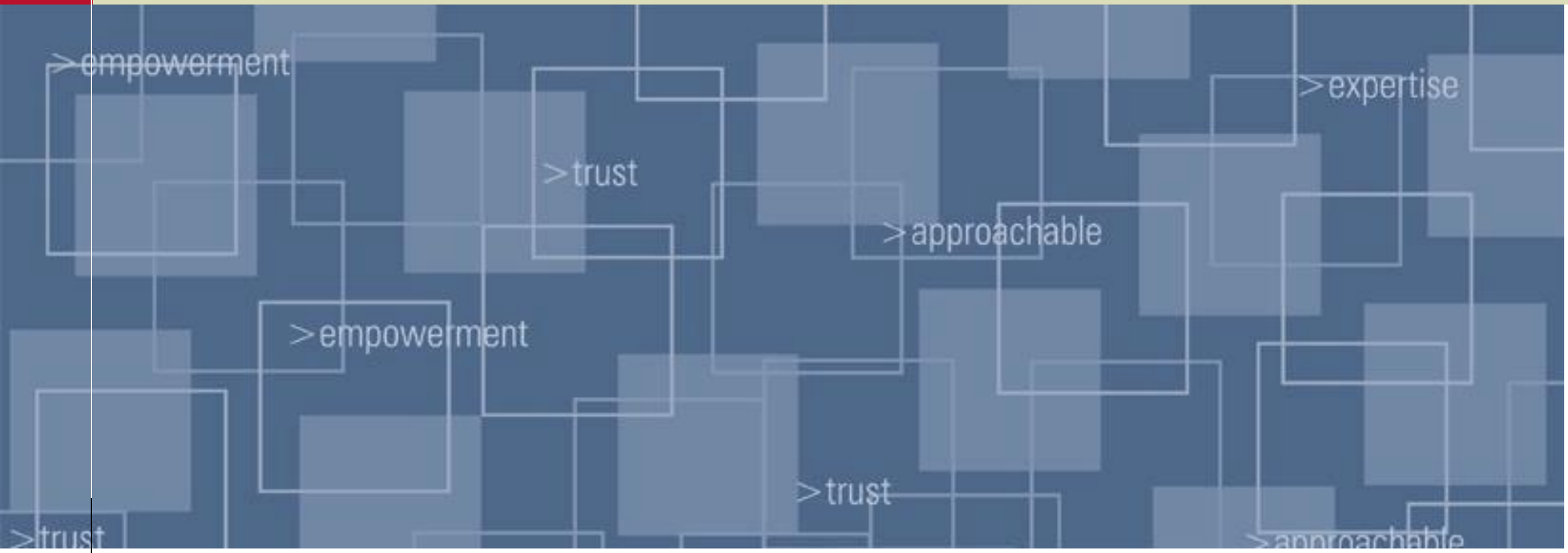
Client.bat

openssl s_client -connect 127.0.0.1:443 -CAfile ca.cer -state -verify ca.cer

GET /index.htm HTTP/1.1

Digital Certificates – GSE 2007

Nigel Pentland



Clydesdale Bank Yorkshire Bank National