



Mike AS Jones

- *Authentication*
- *Authorisation*
- *Accounting*

Date 5 July 2004
Event eSocial Science All Hands Meeting
Venue Hulme Hall



- **Authentication**
 - Who are you?
- **Authorisation**
 - Are you allowed to do that?
- **Accounting**
 - I'm recording who you are and what you are doing.
- All parts important – all parts *really* should be present

AAA in a basic web page context

- point browser at google... to search for NCeSS
- **Authentication**
 - Who are you? – Anyone and everyone.
- **Authorisation**
 - Does “Anyone and everyone” have permission to search my database for what they've asked for? ...yes
- **Accounting**
 - At 15:05 5 July 2004 – Connection from machine “wallace.mvc.mcc.ac.uk” with the identity “Anyone and everyone” asked to search for NCeSS ... success.

AAA in a complex web page context

- point browser at MIMAS; search for data about GDPs
- **Authentication**
 - Who are you? – Mike Jones from The University of Manchester
 - So you say, let me check... What's your password? *****.
 - Dear E.G. Athens,
A person here says they're "Mike Jones from The University of Manchester" with password "*****". Can you authenticate them?
- **Authorisation**
 - Can any entity from "wallace.mvc.mcc.ac.uk" access this data? ...no
 - Can any entity whose favourite colour is blue access the data? ...no
 - What about Athens ID "Mike Jones from The University of Manchester" ... yes
- **Accounting**
 - At 15:10 5 July 2004 – Connection from machine "wallace.mvc.mcc.ac.uk" identifying themselves as "Mike Jones from The University of Manchester" asked for Population Density Data ... success.

Why change what we have

- **Single Sign On**

- Easiest way into a system is still Social Engineering
 - Find passwords on post-it notes on desks, in log books etc.
 - Lots of passwords, lots of usernames makes people write them down and store them badly.
 - Easier to remember one complex, difficult-to-crack password

- **Delegation**

- We can create systems, programs and agents to take a proxy of our identity and act on our behalf.

- **Mutual Authentication**

- To do the delegation step Mutual Authentication is essential
- Also necessary if you're using a old-fashioned password challenges

*An online verification of identity
...passport?*

Methods of Authentication

Simple Client – Server world

- Can be as simple as username/password challenge
- Server need only look to itself to check a client's identity

Something more sophisticated required for online and distributed/grid:

- **Shared Secret Method (e.g. 1st half of Kerberos)**
 - *Pro.* Must contact server each attempt to authenticate: Revocation can be immediate and usage can be monitored.
 - *Pro.* Server can impose strict conditions on passwords and renewal times.
 - *Con.* Must contact server each attempt to authenticate: open to denial of service and slow network conditions and hacking.
 - *Con.* Secret is shared: Identity can be forged.
- **Public Key Method**
 - *Pro.* Only one copy of Key: ID cannot be forged: Non-Repudiation
 - *Pro.* Certificate Revocation, (not subject to denial of service unless sustained)
 - *Con.* User owns key and can be as careless as they wish
 - *Con.* Computationally intensive

The UK eScience Certificate Authority the root of trust

The screenshot shows a Microsoft Internet Explorer browser window displaying the UK eScience Certificate Authority Home Page. The address bar shows the URL <http://ca.grid-support.ac.uk>. The page title is "Certificate Authority - Home Page". The main content area contains several links and instructions:

- Get the CA Root Certificate** [Import it into your browser]
- Request a Certificate** [Apply for certification]
- Renew a Certificate** [Renew a certificate that is about to expire]
- Download Certificate** [Import into your browser]
- Import Certificate Revocation List in browser** [Import CRL into your browser]
- Test Certificate** [Test your certificate with Netscape]
- View Certificate Lists** [View pending, valid and revoked certificate lists]
- Download certificate revocation list file** [Download CRL as a PEM file]
- Revocation Request** [Revoke your certificate with Netscape]

At the bottom of the page, it states: "This product includes software developed by the OpenCA Group for use in the OpenCA project (<http://www.OpenCA.org/>)".

- Offline (in a large safe!)
 - CA private key
 - Certificate Signing
- Online
 - CA Root Certificate
 - Certificate Policy and Practises Statement.
 - Certificate Repository
 - Revocation List
 - Certificate Applications
- Distributed
 - Registration Authorities

Getting Certificates

- Create a private and public key pair
- Send public key to CA
- Identify yourself to the CA (as specified in CPS)
- CA signs your public key.
- CA sends you a digital certificate which contains your public key and the CA's digital signature

- in UK eScience case this can be done two ways:
 - in your browser Netscape/IE certificate request
 - on the command line: `grid-cert-request (pkcs10)`

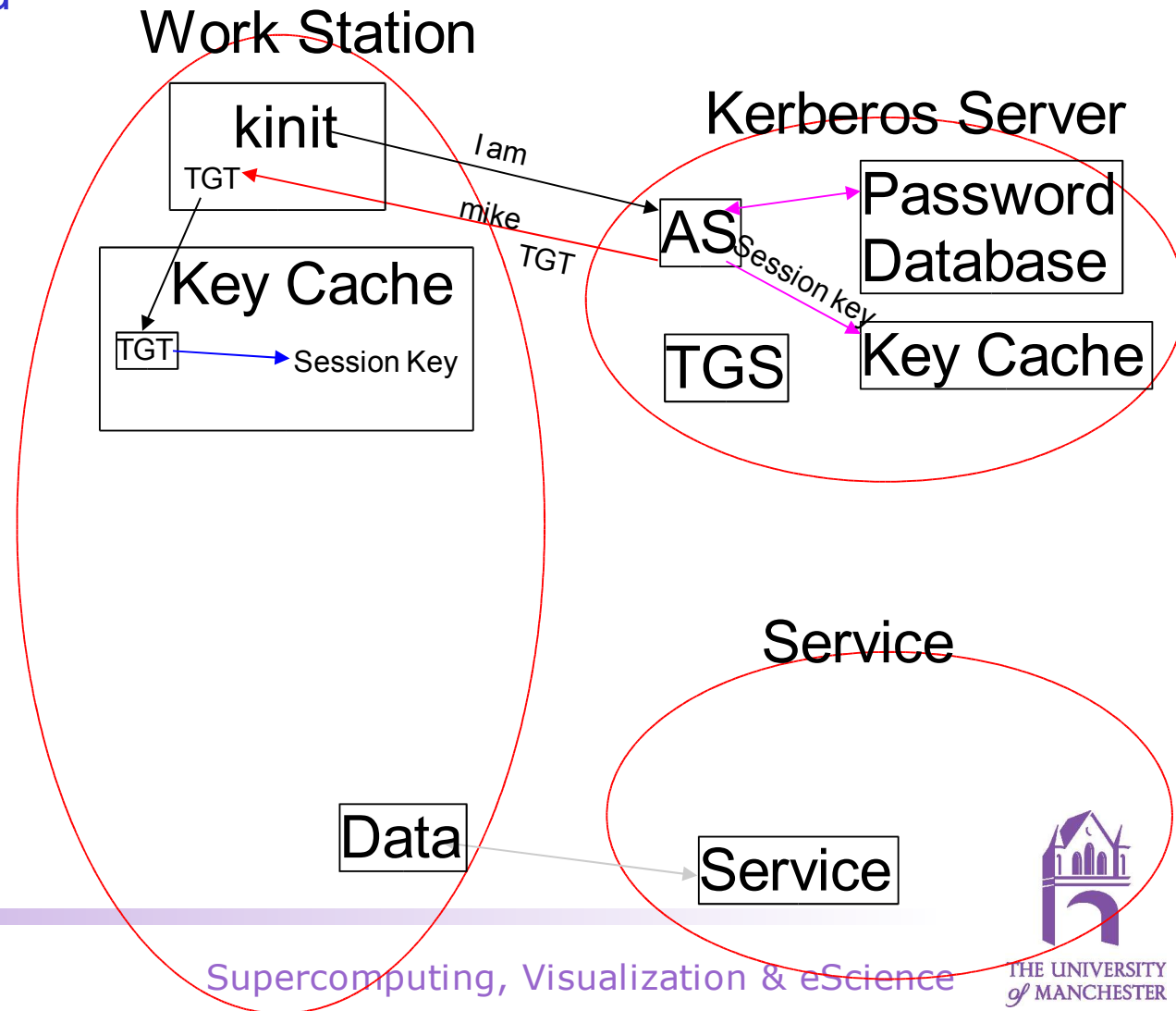
Limitations of UK eScience CA

- Designed to be as scalable as possible in the eScience Context
 - OK for the issuing and maintenance of order 1000 certificates
- Requires a lot from the end user
 - specific Web Browsers, specific Java Versions, PKI knowledge...
- RA's responsibility to do the initial identification
 - Passport. Driving Licence or Institution's Photo ID.
- RA generally ends up walking users through the procedure
- RA's need to brief users on how to look after private keys
 - Possession of the private key = Ownership of that Identity,
any fine grain restrictions are meaningless when the identity is impersonated
ie Gaining someone's private key opens all doors to all the services they subscribe to.
- CA still signs all requests; federated not distributed.
 - Better to have Hierarchy of CAs; but that's hard to achieve.

Kerberos (first bit)

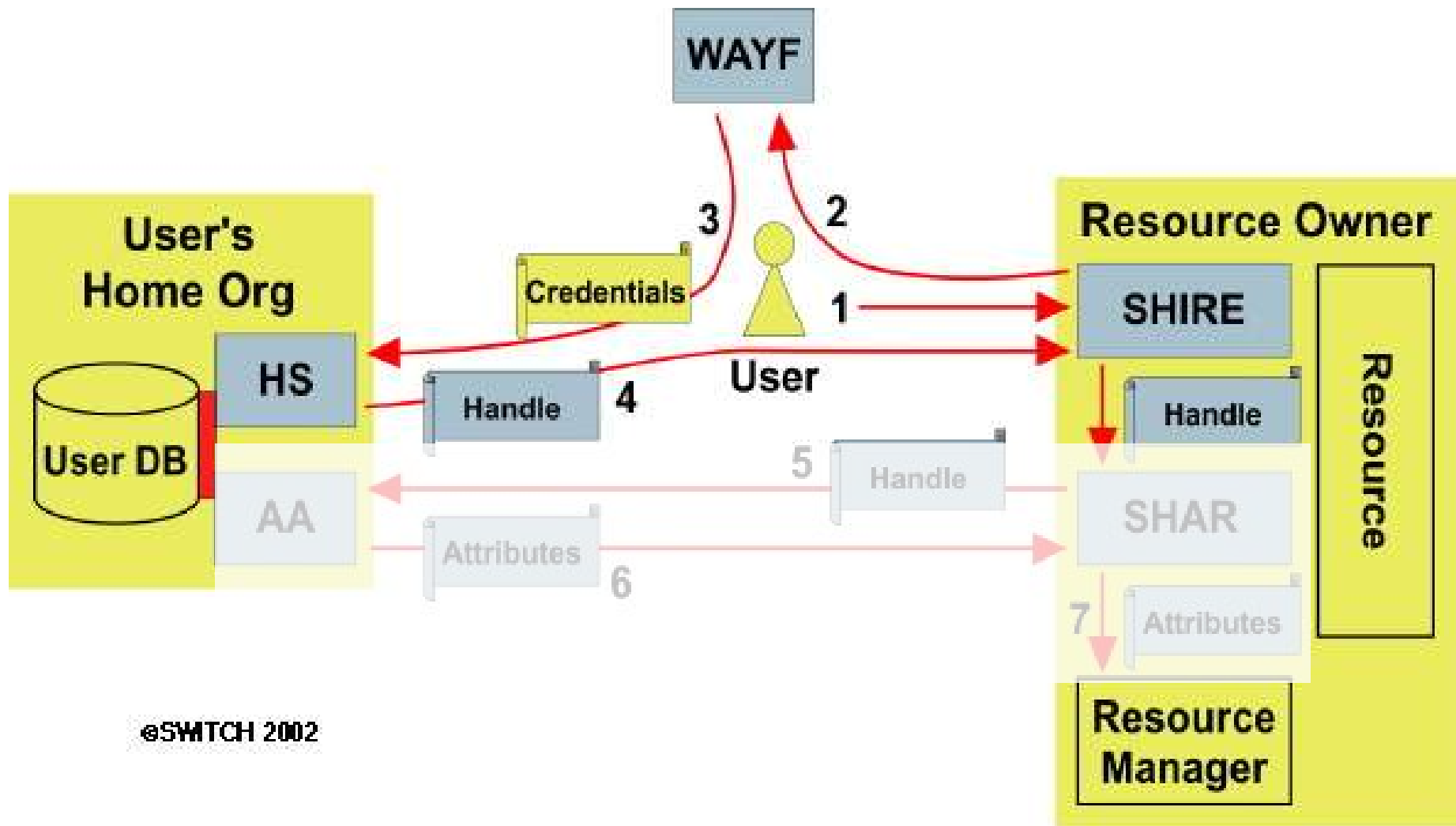
- Ask Kerberos server for TGT
- TGT is encrypted with shared key
- If you can decrypt it you can get an authorisation token

- Ticket Granting Ticket
 - [Session key, Time Stamp, Lifetime & ID] encrypted with password
- Session key for comms.



Shibboleth Architecture

http://www.eapartnership.org/docs/Apr2004/Shibboleth_Gettes.ppt



eSWITCH 2002

- **Hardware**

- Smart Cards – solution to user stored
 - Private key is generated on the card.
 - Private key never leaves the card
 - Card is used to sign Authentication requests only
 - Card can only be activated with a PIN
 - Time based Safety net
 - Number of attempts limitation

- **Biometric**

- Finger Prints,
- Photo Recognition,
- Retinal Scans
- Voice Recognition
- and other Sci-Fi
- storage of comparison data becomes the weakest link
- Still need cryptography if authentication server is elsewhere

*An online verification of permission
...visa?*

Authorisation Steps

Once authenticity has been established authority needs to be ascertained.

- **Push mode.**
 - The Client may be able to provide evidence of their authority
- **Pull mode.**
 - The Server may be able to retrieve evidence from another source.
- In principle both can be done and it's a policy choice of the service provider.

Once evidence has been provided/obtained rights need to be calculated.

Evidence of Authority

- **Components of an Identity certificate**
 - /C=uk/O=eScience/OU=Manchester/L=MC/CN=michael jones
- **Attribute Certificates**
 - Akenti Style:
 - Signed XML file containing key value pairs
 - Permis Style:
 - Signed X509 PMI certificates similar to X509 Authentication Certificates
- **System**
 - Anything that can be retrieved from the local environment
 - e.g. local Time
 - IP address of incoming connection (not a good idea though)
- **Attribute Standards**
 - e.g. eduPerson

Authority Engines

- A number of competing engines exist for measuring and granting access rights:
 - Akenti – facility to manage access control by distributed entities (+ remote calls)
 - CAS – credentials in proxy allowing access to groups
 - Grid-mapfile – maps IDs to UIDs
 - Kerberos (part 2) – Allows short term access to specific service from specific location
 - Permis – Handles is this person allowed to do this (+ remote calls)
 - Shibboleth
 - VOM (see grid-mapfile)
 - VOMS (see mainly grid-mapfile at the moment)
 - ...
- Some methods take an action and ID and return yes/no/maybe
- Some methods take ID, and return a username
- Some methods populate lists of ID mappings to UIDs
- Some methods do a mixture.

Record of activity

...Immigration's records?

Accounting Motivations

- The record of usage for charging purposes
 - Most grids at the moment are free to those authorised
 - Mobile phones have a model that works
- The record of attempts to use a service
 - Hacker?
 - Grid fabric failure
- Tracking usage for security purposes
 - Audit Trails for legal usage

Keeping Records

- **What to record**
 - How much usage
 - What time
 - Which components of my service
 - by whom...
- **How to record**
 - Take from log files
 - Service logs eg Web server access
 - System logs
 - Take from computing environment
 - Properties from Job containers
- **Location of storage**
 - Local copy
 - Central copy
 - Monitor Grid Fabric
 - Back up of Local Copy

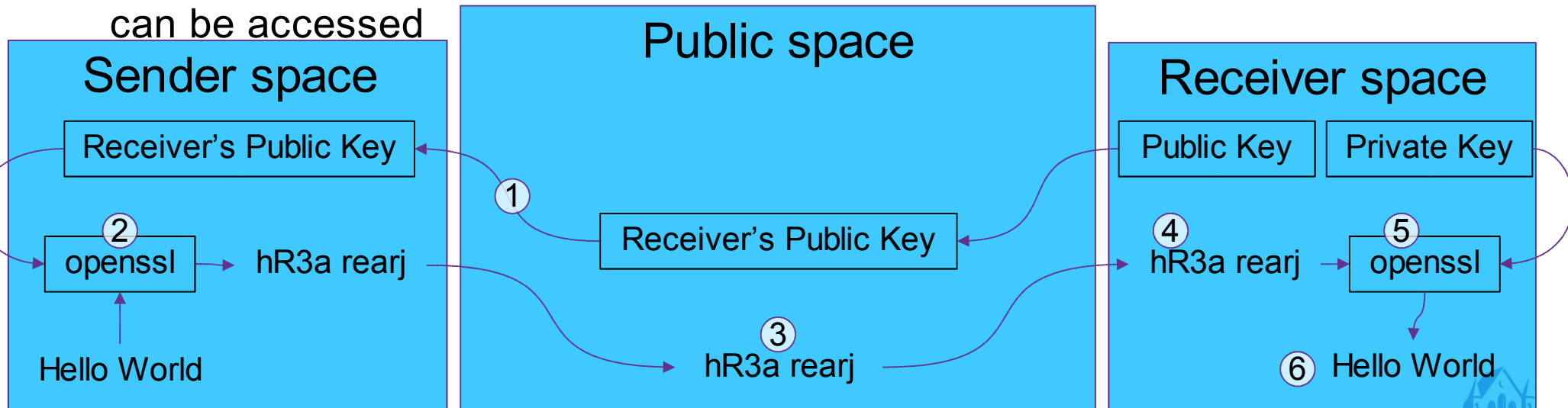
The End

Appendix

Encryption/Decryption

- Using PKI to send and receive encrypted data

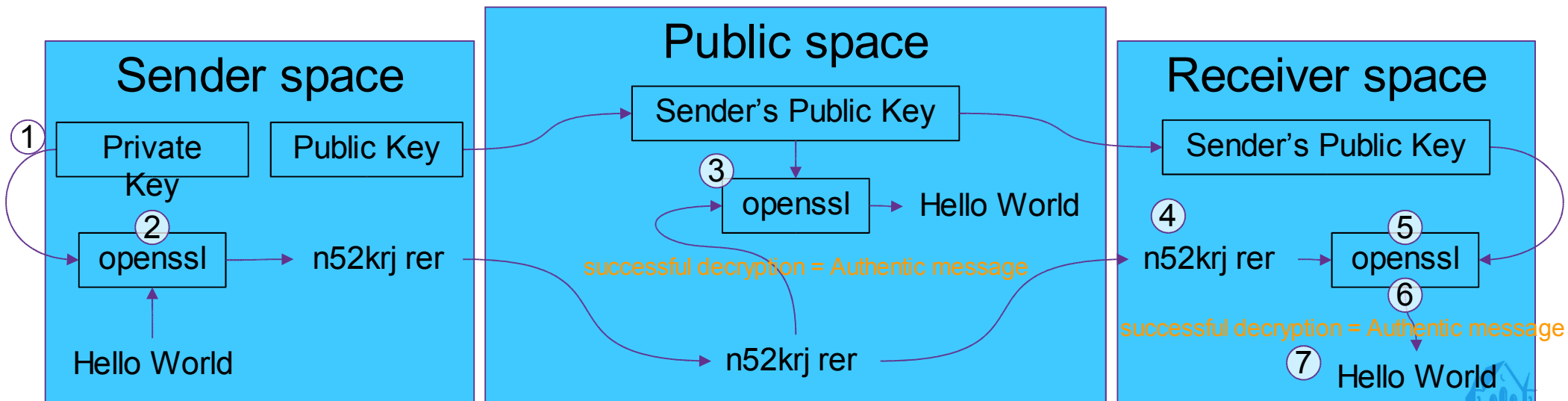
1. Find recipient's public key
2. Use eg openssl to encrypt message
3. Encrypted message can not be read unless recipient's private key can be accessed
4. Receive encrypted message
2. Use eg openssl to decrypt message
 - requires private key
6. Read message



Basic Signing

- Using PKI to send and receive authenticated data

1. Use sender's private key
2. Encrypt message with private key
3. Encrypted message can be read by anyone who has sender's public key
1. Receive encrypted message
5. decrypt message with sender's public key
6. Success guarantees authenticity
7. Read message



Signing and Encrypting

- We've discussed encryption/decryption
- and basic signing (or proof of origin)
 - encrypting message with own private key requires recipient to decrypt before reading
- instead, create a *Hash* and encrypt that.
- Hash is a one way digest of the message by a specific algorithm (eg SHA1 or MD5)
- Encrypt the hash and include it in the message.
- Verify message by
 - making the hash
 - decrypting the signature
 - matching hash and decrypted signature

We rely on ourselves to get true public keys:

- We can get public key directly from the owner
- Or we can have someone we trust sign the public key as authentic
 - Web of trust rules
 - A public key may be digitally signed by many people
 - some of whom you may trust.
 - you may set up some rules based on your trust of other people
 - CA method (Certificate Authority)
 - CA has a “root certificate” and a document called CP/CPS (Policy & Practice) <http://www.grid-support.ac.uk/ca/cps>
 - You choose to trust on the basis of CP/CPS.
 - CA signs your public key (issues your X.509 certificate).
 - Large scale CAs are difficult and costly

Trust Chain

- Servers (and clients) trust a set of CAs
- Incoming message is signed with a personal key
 - It is accompanied by the public part of that personal key pair (the certificate).
- Recipient verifies the authenticity of the certificate
 - check certificate contains signature of a trusted CA
- Recipient verifies the message
 - check message contains signature of verified certificate
- Recipient Trusts the origin of message by trusting the CA

What is in a certificate

- Next few slides show the output from the following openssl commands:
 - `openssl x509 -in filename -text`
- You may see certificates can be stored in the following formats.
 - pkcs12/pcx
 - easy to handle, import and export from browsers etc.
 - May contain a number of certificates is a chain including private keys
 - PEM
 - the format Globus and most service configurations require
 - each key/certificate is a separate PEM entity (one file may contain many PEM)
 - DER – used in Unicore installations

x509 Certificates

Certificate:

Data:

```
Version: 3 (0x2)
Serial Number: 127 (0x7f)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=UK, O=eScience, OU=Authority, CN=CA/Email=ca-operator@grid-support.ac.uk
Validity
  Not Before: Oct 31 15:50:59 2002 GMT
  Not After : Oct 31 15:50:59 2003 GMT
Subject: C=UK, O=eScience, OU=Manchester, L=MC, CN=michael jones
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (1024 bit)
  Modulus (1024 bit):
    00:c6:96:fd:7a:e0:fa:f1:e6:43:9d:c1:cb:72:38:
    e1:4e:44:86:da:a7:8a:ed:8a:fc:f3:64:d8:9e:bd:
    af:ce:7c:55:39:cd:61:74:a8:1d:6d:60:6e:65:91:
    dc:2c:c2:64:80:f6:f9:1a:3c:fe:d4:d2:1c:52:fa:
    c6:47:ea:a6:4e:92:b5:c9:1d:93:dd:48:61:54:40:
    b5:17:84:3f:5c:47:48:29:2b:83:82:c7:d6:ad:d3:
    60:5d:6d:5d:f7:08:25:17:d2:14:e2:8e:af:37:3b:
    e4:3b:63:f7:31:24:b4:66:78:8e:06:93:c6:8d:b6:
    fe:50:79:3a:4a:f8:59:58:3d
  Exponent: 65537 (0x10001)
X509v3 extensions:
  X509v3 Basic Constraints:
    CA:FALSE
  Netscape Cert Type:
    SSL Client, S/MIME
  X509v3 Key Usage:
    Digital Signature, Non Repudiation, Key Encipherment, Key Agreement
  Netscape Comment:
    UK e-Science User Certificate
  X509v3 Subject Key Identifier:
    BF:00:02:4B:3A:45:A6:B8:EB:66:E4:F2:EE:CA:60:9D:B8:D1:B2:0D
  X509v3 Authority Key Identifier:
    keyid:02:38:AB:11:A3:96:80:8B:0D:D3:15:2B:08:A5:8E:30:DA:B2:DA:A8
    DirName:/C=UK/O=eScience/OU=Authority/CN=CA/Email=ca-operator@grid-support.ac.uk
    serial:00
```

- Version
- Serial Number
- Issuer
- Times of Validity
- Subject
- Public Key
- Extensions

- Constraints
- Type and Use
- Thumbprint



x509 Certificates

```
X509v3 extensions:  
  X509v3 Basic Constraints:  
    CA:FALSE  
  Netscape Cert Type:  
    SSL Client, S/MIME  
  X509v3 Key Usage:  
    Digital Signature, Non Repudiation, Key Encipherment, Key Agreement  
  Netscape Comment:  
    UK e-Science User Certificate  
X509v3 Subject Key Identifier:  
  BF:00:02:4B:3A:45:A6:B8:EB:66:E4:F2:EE:CA:60:9D:B8:D1:B2:0D  
X509v3 Authority Key Identifier:  
  keyid:02:38:AB:11:A3:96:80:8B:0D:D3:15:2B:08:A5:8E:30:DA:B2:DA:A8  
  DirName:/C=UK/O=eScience/OU=Authority/CN=CA/Email=ca-operator@grid-support.ac.uk  
  serial:00
```

```
X509v3 Issuer Alternative Name:  
  email:ca-operator@grid-support.ac.uk  
Netscape CA Revocation Url:  
  http://ca.grid-support.ac.uk/cgi-bin/importCRL  
Netscape Revocation Url:  
  http://ca.grid-support.ac.uk/cgi-bin/importCRL  
Netscape Renewal Url:  
  http://ca.grid-support.ac.uk/cgi-bin/renewURL  
X509v3 CRL Distribution Points:  
  URI:http://ca.grid-support.ac.uk/cgi-bin/importCRL
```

Extensions

- CA Information
- CA Alternative Info
- CRL Location
- CA Signature

Signature Algorithm: md5WithRSAEncryption

```
3a:1f:81:a8:1a:83:ff:2c:0f:7b:b6:1e:2a:87:31:13:d9:ca:  
9e:c1:9e:e4:42:b5:22:56:7b:01:98:11:13:29:a3:d8:d2:37:  
80:58:ac:7f:44:f7:1e:ba:00:f4:8b:c8:34:00:ff:44:27:c2:  
2a:54:8b:95:e9:a0:00:f8:3d:60:92:c4:99:2b:72:d4:b7:dd:  
78:bd:c9:4a:01:d7:14:1d:3c:d9:6f:60:7b:23:90:8e:d6:3a:  
2d:45:39:5e:bc:fd:6d:77:7b:1e:cf:43:8c:e4:05:4c:1b:91:  
e5:bb:da:3d:cd:9d:05:6b:be:21:b0:e8:43:b2:4b:4e:c4:4f:  
6b:4e:23:9e:03:d2:03:86:1b:44:68:60:41:5d:64:ae:2d:52:  
e2:7d:9b:99:60:71:7f:4a:00:1e:5d:9d:14:59:4f:4b:d7:9a:  
ee:e0:01:3d:87:36:16:bf:24:b3:84:fd:62:d1:d6:21:ae:3b:  
f7:e1:e5:52:ec:ef:68:f4:73:4f:1b:62:a6:f4:47:0b:6c:1e:  
28:23:6b:25:d3:a1:f7:37:f6:55:d6:82:7c:49:a9:1d:71:57:  
e6:bc:74:71:94:0d:df:fc:21:63:16:54:c9:0f:51:1c:7a:bf:  
5c:ef:7d:28:23:73:64:84:eb:f2:b6:52:89:ca:48:78:31:e8:  
dd:b9:91:3f
```

PEM Encoding

-----BEGIN CERTIFICATE-----

```
MIIFBDCCA+ycAwIBAgIBfzANBgkqhkiG9w0BAQoFADBwMQswCQYDVQQGEwJVSzER  
MA8GA1UEChMIZVNZjaWVuY2UxEjAQBgNVBAsTCUFlldGhvcml0eTELMakGA1UEAxMC  
Q0ExLTAhBgkqhkiG9w0BCQEWHmNhLW9wZXJhdG9yQGdyaWQtc3VwcG9ydC5hYy51  
azAeFw0wMjEwMzExNTUwNTlaFw0wMzEwMzExNTUwNTlaMFoxCzAJBgNVBAYTAlVL
```

GSI modifications

Proxy Certificate

- Issuer (Me! pretending to be a CA)
- Short Lived

Special Subject

- Requires Subject-Issuer constraint

Smaller key size

- Signed by my X.509 certificate
- Breaks x509 Standard

Encoded Certificate

Contains Unencrypted RSA Key

- Includes my certificate (the Issuer)



Certificate:

Data:

```
Version: 3 (0x2)
Serial Number: 127 (0x7f)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=UK, O=eScience, OU=Manchester, L=MC, CN=michael jones
Validity
  Not Before: Jan  5 16:43:48 2003 GMT
  Not After : Jan  6 04:48:48 2003 GMT
Subject: C=UK, O=eScience, OU=Manchester, L=MC, CN=michael jones, CN=proxy
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (512 bit)
  Modulus (512 bit):
    00:99:16:b5:ff:4b:f4:90:48:7b:8e:95:8c:e0:8a:
    b8:ad:51:c3:74:9f:e2:e7:ba:61:ee:1c:d8:f7:bc:
    96:66:57:3a:01:36:1a:e1:e1:55:7e:f8:64:2e:c7:
    f5:d4:23:b1:42:3e:0b:61:1c:fb:fd:5f:06:f6:2f:
    57:b7:81:1c:ff
  Exponent: 65537 (0x10001)
Signature Algorithm: md5WithRSAEncryption
a9:9a:e0:33:70:29:0a:9c:57:02:1a:80:c1:f1:c2:6e:6c:34:
3d:f3:3e:32:49:83:c8:b1:c6:21:d9:3c:84:d3:5d:17:ca:d6:
fa:96:b0:37:e2:4d:95:08:b7:3e:1f:6c:4a:79:7d:83:5e:21:
43:5d:42:60:2f:2c:5d:61:f9:e8:82:97:82:9b:89:cb:a4:ae:
97:0c:26:df:39:76:15:a6:38:53:8f:7a:f5:6f:ed:d6:76:ae:
a9:28:db:52:69:1c:e8:25:cf:7b:31:10:a1:49:2d:bb:91:eb:
af:d3:e7:d0:6d:28:21:3c:d8:16:3b:7c:4e:c9:94:d2:ff:23:
4e:2a
```

-----BEGIN CERTIFICATE-----

```
MIIB9DCCA2gAwIBAgIBfzANBgkqhkiG9w0BAQQFADBaMQswCQYDVQQGEwJVSzER
MA8GA1UEChMIZVNjaWVWYyU2UXEzARBgNVBAsTCk1hbmNoZXRhZXRhZXRhZXRh
Ak1DMRYwFAYDVQQDEw1taWNoeWVWYyIGpvcmbVzMB4XDTAzMDEwMTE2MTUw
MDEwNjA0NDg0OFowajELMAkGA1UEBhMCVUswETAPBgNVBAoTCGVTY21lbmNl
MQswCQYDVQQLLEwvYyU2UXEzARBgNVBAsTCk1hbmNoZXRhZXRhZXRhZXRhZXRh
bCBqY25lc2EOMAwGA1UEAxMFChJveHkwXDANBgkqhkiG9w0BAQEFANLADBIaKEA
mRa1/0v0kEh7jPwM4Iq4rVHddJ/i57ph7hzY97yWZlC6ATYa4eFVfVhkLsf11COx
Qj4LYRz7/V8G9i9Xt4Ec/wIDAQABMA0GCSqGSIb3DQEBAUAA4GBAKma4DNwKQqc
VwIagMHxwm5sND3zPjJJg8ixxiHZPITTXRfK1vqWsDfiTZUItz4fbEp5fYNeIUND
```

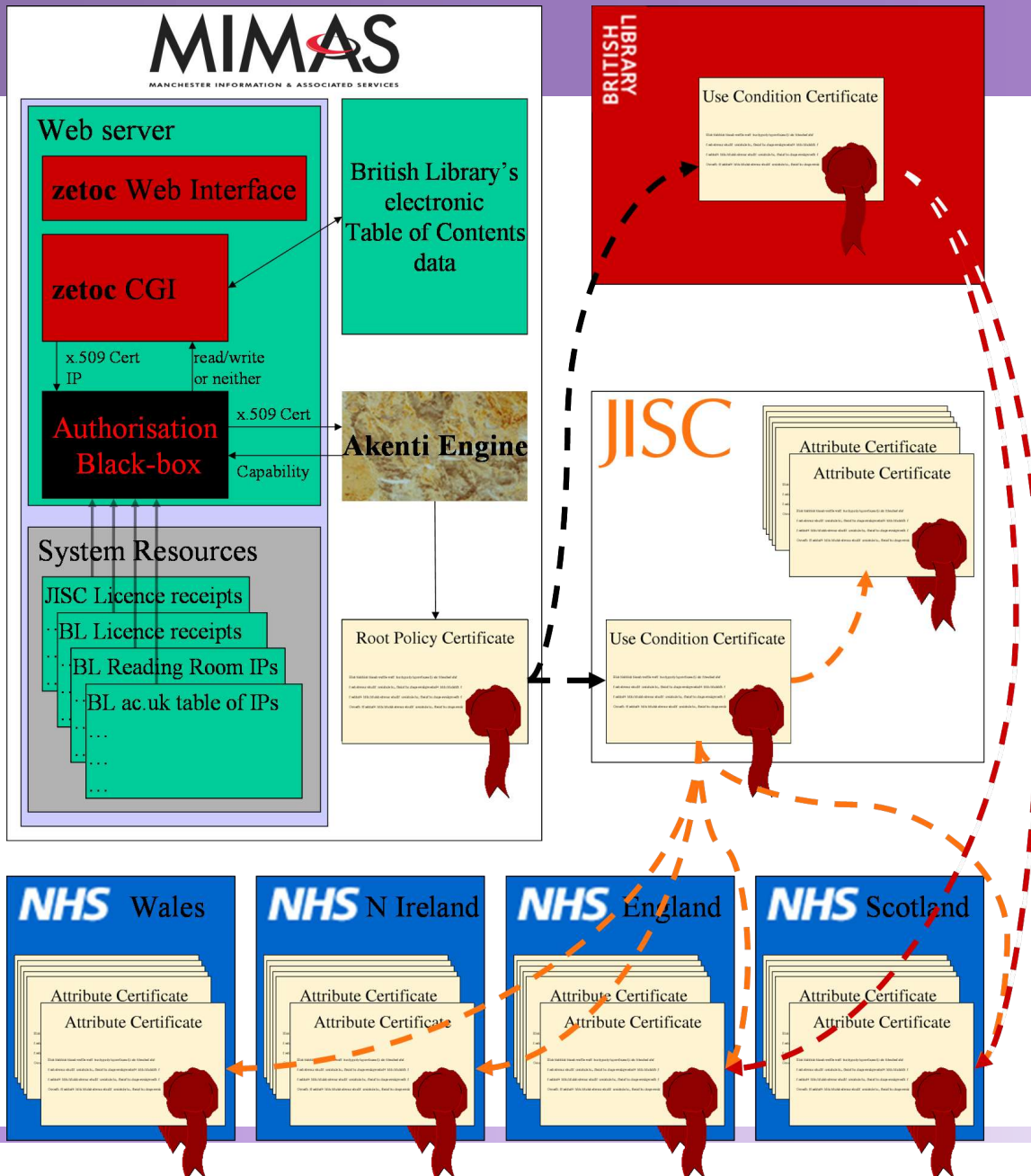

Proxy Certificates

- `grid-proxy-init [-cert cert.pem -key key.pem]`
- `X509_USER_PROXY = /tmp/x509up_u`id -u``
 - Contains certificate chain from but excluding CA
 - Contains unencrypted key
 - Has a short lifetime
 - is read only to owner
- Limited Proxies delegated to remote resources
 - eg: `X509_USER_PROXY = $HOME/.globus/.gass_cache/local/md5/cb/ab/8e/5031401cebf3a4b1da92857230/md5/95/5c/21/a3713a03e529757cc677fdb079/data`
- `grid-proxy-destroy`

- A good CA will produce some way of revoking certificates
 - If a certificate needs to be reissued (good CAs issue only one certificate per entity)
 - If a private key is compromised
- A CRL is a Certificate Revocation List
 - It is a Signed document that contains
 - a list of no longer valid certificates
 - a valid from date
 - an expires by date
 - If available it should always be installed
 - If installed with a globus installation revoked certificates will not be trusted
 - If the installed CRL is out of date all certificates from that CA will not be trusted
 - If the CRL is not installed all otherwise valid certificates issued by that CA will be trusted.

Akenti Architecture example

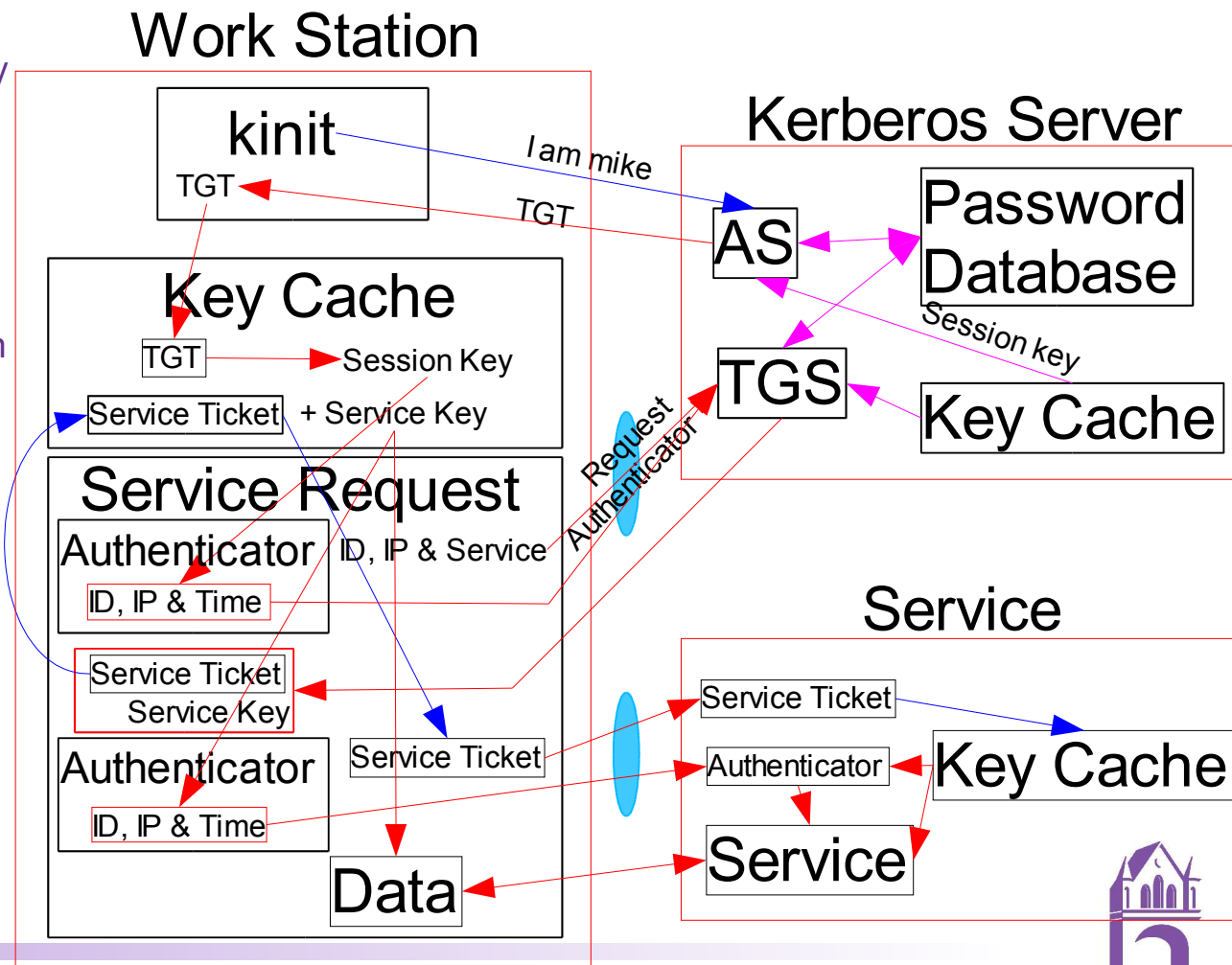
<http://a2z.mimas.ac.uk>



- **Base Authority File**
 - local overarching rules
- **Distributed Use Conditions**
 - x509 rules
 - attribute rules
 - system rules
 - external rules
- **Attribute Certificates**
 - User owned – push model
 - Publicly available – pull model

Kerberos (Authorisation)

- Request to kerberos to use service
 - returns service's-kerberos-key encrypted Service Ticket and temporary service communication key
- Present Service Ticket
 - If Service can decrypt it will get a temporary key
- Communicate with temporary key
 - If Service can decrypt communication is from authorised entity.
- Service Request
 - Authenticator: encrypted [ID, IP, Time Stamp and (short) Time to Live]
 - Service Ticket: [Service Key, ID, IP, Time Stamp & Lifetime] encrypted with service password



- Based on Privilege Management Infrastructure
- PMI certificates mirror PKI certificates except they map an identity to an attribute.

