



## Deep Dive - IBM MQ V8 Latest Security Features

Lyn Elkins – [elkinsc@us.ibm.com](mailto:elkinsc@us.ibm.com)  
Mitch Johnson – [mitchj@us.ibm.com](mailto:mitchj@us.ibm.com)



© 2014 IBM Corporation

### Agenda

- Channel Authentication with Hostnames
- MQ Enhancements for Digital Certificates
- Connection Authentication Enhancements

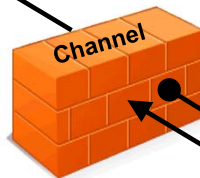
## Channel Authentication – V7.1 feature review

- Set rules to control how inbound connections are treated
  - Inbound Clients
  - Inbound QMgr to QMgr channels
  - Other rogue connections causing FDCs
- Rules can be set to
  - Allow a connection
  - Allow a connection and assign an MCAUSER
  - Block a connection
  - Ban privileged access
  - Provide multiple positive or negative SSL Peer Name matching
- Rules can use any of the following identifying characteristics of the inbound connection
  - IP Address
  - SSL/TLS Subject's Distinguished Name
  - Client asserted user ID
  - Remote queue manager name
  - **And with MQv8..... hostname**

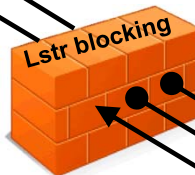
## Channel Access Blocking Points



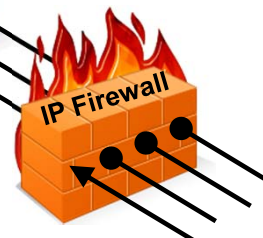
ACLs/SAF



Channel



Lstr blocking



IP Firewall

- Listener Blocking
  - NOT A REPLACEMENT FOR AN IP FIREWALL!!
  - Blocked before any data read from the socket
  - Simplistic avoidance of DoS attack
    - Really the place of the IP firewall
  - Network Pingers if blocked don't raise an alert

- Channel Blocking/Mapping
  - Rules to block channels
  - Rules to map channels to MCAUSER
  - Rules to allow channels as they are
  - Runs before security exit
  - Final check for user ID before allowing through
    - After Security Exit has run and final MCAUSER is assigned
    - Ban privileged users with '\*MQADMIN'

IBM

## Channel Authentication Rules using IP Addresses

- Initial Listener blocking list
  - Should be used sparingly
  - List of IP addresses/range/pattern
  - Not replacing IP firewall
- Channel based *blocking* of IP addresses
  - Single IP address/range/pattern
- Channel *allowed in*, based on IP addresses
  - Single IP address/range/pattern
- Further qualified rule including IP address on another rule type
  - Works with SSLPEER, QMNAME and CLNTUSER

```
SET CHLAUTH(*) TYPE(BLOCKADDR)
ADDRLIST('9.20.*', '192.168.2.10')
```

```
SET CHLAUTH('APPL1.*')
TYPE(ADDRESSMAP)
ADDRESS('9.20.*') USERSRC(NOACCESS)
```

```
SET CHLAUTH('*.SVRCONN')
TYPE(ADDRESSMAP)
ADDRESS('9.20-21.*') MCAUSER(HUSER)
```

```
SET CHLAUTH(*) TYPE(SSLPEERMAP)
SSLPEER('CN="Mitch Johnson"')
ADDRESS('9.20.*') MCAUSER(MITCHJ)
```

5
zGrowth Team – Washington
© 2014 IBM Corporation

IBM

## Channel Authentication Rules using Hostnames (new in V8)

- Initial Listener blocking list
  - Hostnames not allowed
- Channel based blocking of Hostnames
  - Single IP address/range/pattern or hostname/pattern
- Channel allowed in, based on Hostnames
  - Single IP address/range/pattern or hostname/pattern
- Further qualified rule including hostname on another rule type
  - Works with SSLPEER, QMNAME and CLNTUSER

~~```
SET CHLAUTH(*) TYPE(BLOCKADDR)
ADDRLIST( )
```~~

```
SET CHLAUTH('APPL1.*')
TYPE(ADDRESSMAP)
ADDRESS('*.isis.org')
USERSRC(NOACCESS)
```

```
SET CHLAUTH('*.SVRCONN')
TYPE(ADDRESSMAP)
ADDRESS('mach123.ibm.com') MCAUSER(HUSER)
```

```
SET CHLAUTH(*) TYPE(SSLPEERMAP)
SSLPEER('CN="Joe User"')
ADDRESS('s*.ibm.*') MCAUSER(JUSER)
```

6
zGrowth Team – Washington
© 2014 IBM Corporation

## Precedence Order

```

DISPLAY CHLAUTH(APPL1.*)
returns ==>
CHLAUTH(APPL1.*)
TYPE(SSLPEERMAP)
SSLPEER('O="IBM UK"'') MCAUSER(UKUSER)

CHLAUTH(APPL1.*)
TYPE(USERMAP)
CLNTUSER('mhughson') MCAUSER(HUGHSON)

CHLAUTH(APPL1.*)
TYPE(ADDRESSMAP)
ADDRESS('9.180.165.163') MCAUSER(MORAG)

CHLAUTH(APPL1.*)
TYPE(ADDRESSMAP)
ADDRESS('*.ibm.com') MCAUSER(IBMUSER)
    
```

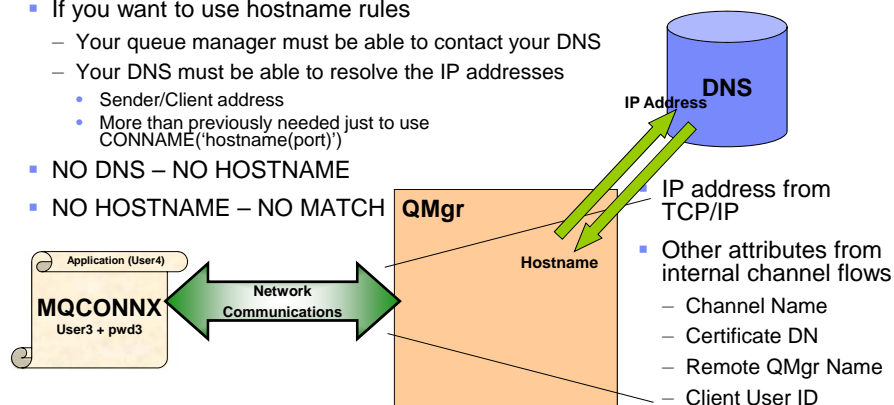
| Order | Identity mechanism      | Notes                                                                          |
|-------|-------------------------|--------------------------------------------------------------------------------|
| 0     | Channel Name            |                                                                                |
| 1     | SSL Distinguished Name  |                                                                                |
| 2=    | Client asserted User ID | Clearly several different user IDs can be running on the same IP address.      |
| 2=    | Queue Manager Name      | Clearly several different queue managers can be running on the same IP address |
| 4     | IP address              |                                                                                |
| 5     | Hostname                | One IP address can have multiple hostnames                                     |



**Chl: APPL1.SVRCONN**  
**DN: CN=M Hughson.O=IBM UK**  
**UID: mhughson**  
**IP: 9.180.165.163**

## Obtaining a hostname

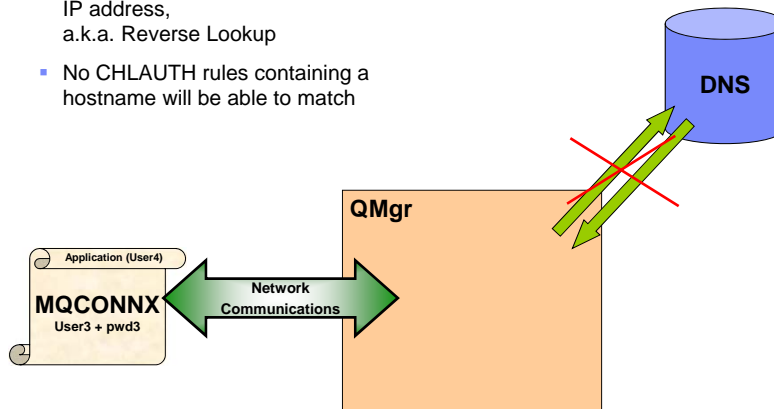
- Hostname is not 'sent' from the other end of the channel
- IP address is obtained from TCP/IP socket
- We must ask the Domain Name Server what the hostname is, a.k.a. Reverse Lookup
- If you want to use hostname rules
  - Your queue manager must be able to contact your DNS
  - Your DNS must be able to resolve the IP addresses
    - Sender/Client address
    - More than previously needed just to use CONNAME('hostname(port)')
- NO DNS – NO HOSTNAME
- NO HOSTNAME – NO MATCH



## Avoiding obtaining a hostname

- To stop the Queue Manager asking the Domain Name Server for hostnames that go with IP address, a.k.a. Reverse Lookup
- No CHLAUTH rules containing a hostname will be able to match

ALTER QMGR  
REVDNS(DISABLED)



## Diagnosing hostname look-up failures

- WebSphere MQ V7.1

AMQ9777: Channel was blocked

### EXPLANATION:

The inbound channel 'SYSTEM.DEF.SVRCONN' was blocked from address '9.180.165.163' because the active values of the channel matched a record configured with USERSRC(NOACCESS). The active values of the channel were 'CLNTUSER(hughson)'.

- WebSphere MQ V8

AMQ9777: Channel was blocked

### EXPLANATION:

The inbound channel 'SYSTEM.DEF.SVRCONN' was blocked from address 'mhughson.ibm.com(9.180.165.163)' because the active values of the channel matched a record configured with USERSRC(NOACCESS). The active values of the channel were 'CLNTUSER(hughson) ADDRESS(mhughson.ibm.com, morag.hursley.ibm.com)'.

## Using MATCH(RUNCHECK) with hostnames

```
DISPLAY CHLAUTH(SYSTEM.ADMIN.SVRCONN) MATCH(RUNCHECK)
SSLPEER('CN="Morag Hughson", O="IBM UK"')
CLNTUSER('mhughson') ADDRESS('9.180.165.163')
returns ==>
CHLAUTH(SYSTEM.ADMIN.SVRCONN)
TYPE(ADDRESSMAP)
ADDRESS('*.ibm.com') MCAUSER(HUGHSON)
```

- Just as before, MATCH(RUNCHECK) mandates an IP address is provided
- Then the queue manager will employ DNS to find the hostname
- MATCH(RUNCHECK) thus also tests whether your DNS is correctly set up.

**Chl: SYSTEM.ADMIN.SVRCONN**  
**DN: CN=Morag Hughson.O=IBM UK**  
**UID: mhughson**  
**IP: 9.180.165.163**

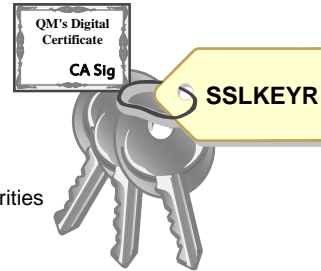


## Agenda

- Channel Authentication with Hostnames
- MQ Enhancements for Digital Certificates
- Connection Authentication Enhancements

## Key Repository (nothing new here!)

- Contains Entity's own Digital Certificate
  - z/OS Queue Manager
    - ibmWebSphereMQ<QMgr Name> (mixed case) label
  - Distributed Queue Manager
    - ibmwebsphermq<qmgr name> (lower case) label
  - Client
    - ibmwebsphermq<logon userid> (lower case) label
  - Digital Certificates from various Certification Authorities
- On z/OS Queue Managers
  - Keyring name
- On Unix®, Windows®, iSeries® QMgrs
  - Key database path
- Clients: mqclient.ini file
  - SSL Stanza – SSLKeyRepository
- MQCONN (MQSCO structure)
  - SSLKeyRepository
- Environment variable
  - export MQSSLKEYR=/var/mqm/ssl/key



**ALTER QMGR SSLKEYR(CSQ1RING)** (z/OS)

**ALTER QMGR SSLKEYR('/var/mqm/qmgrs/QM1/ssl/key')** (Distrib)

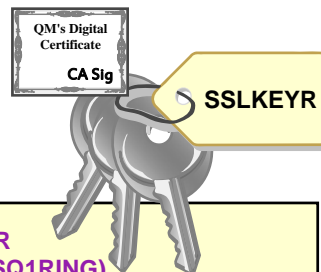
**mqclient.ini** (MQ Client)  
**SSL:**  
**SSLKeyRepository=C:\key**

## Single Queue Manager Certificate

- Name Queue Manager Certificate
  - Using CERTLABL attribute
- Name Client Certificate
  - mqclient.ini file SSL Stanza
    - CertificateLabel
  - MQCONN (MQSCO structure)
    - CertificateLabel
- Environment variable
  - export MQCERTLABL=MyCert

```
MQCNO cno = {MQCNO_DEFAULT};
MQSCO sco = {MQSCO_DEFAULT};

cno.Version = MQCNO_VERSION_4;
sco.Version = MQSCO_VERSION_5;
memcpy(sco.KeyRepository, ...);
memcpy(sco.CertificateLabel,...);
cno.SSLConfigPtr = &sco;
MQCONN (QMName,
        &cno,
        &hConn,
        &CompCode,
        &Reason);
```



**ALTER QMGR SSLKEYR(CSQ1RING) CERTLABL('CSQ1Certificate')** (z/OS)  
**CERTQSSL('SharedCert')**

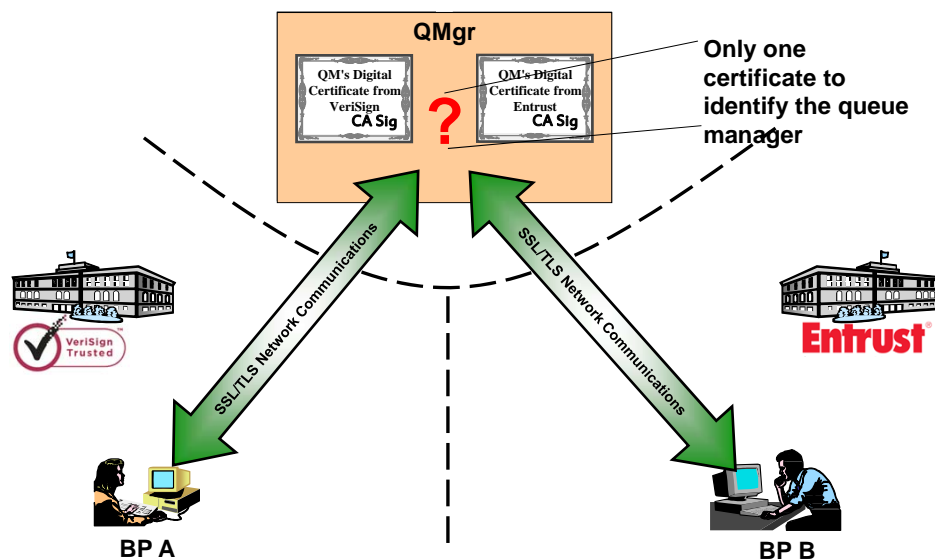
**ALTER QMGR SSLKEYR('var/mqm/qmgrs/QM1/ssl/key') CERTLABL('QM1Certificate')** (Distrib)

**mqclient.ini** (MQ Client)  
**SSL:**  
**SSLKeyRepository=C:\key**  
**CertificateLabel=MyCert**

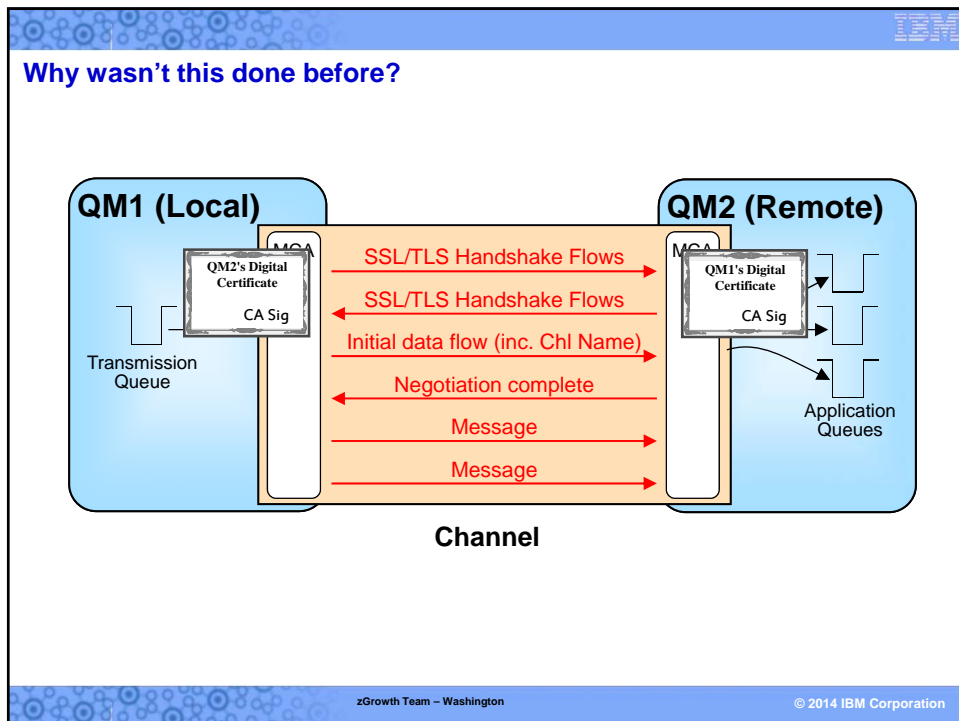
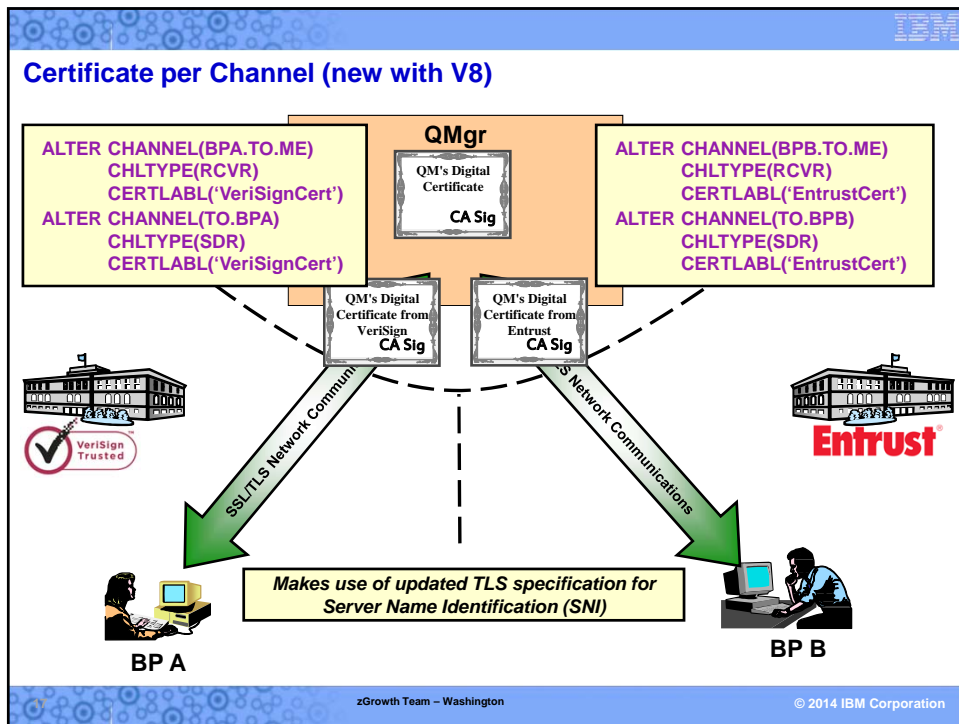
## Use Cases

- Following company policy on certificate labelling
- Using the same certificate for more than one queue manager
  - Not that this would condoned!
- Migrating over to a new certificate when main certificate is ready to expire
  - Used to have to issue GSKit/RACF commands to rename certificate
    - ibmwebsphermqm1 -> ibmwebsphermqm1old
    - ibmwebsphermqm1new -> ibmwebsphermqm1
    - REFRESH SECURITY TYPE(SSL)
  - Now just MQ commands when the time comes
    - Current label is 'QM1 Cert 2013'
    - ALTER QMGR CERTLABL('QM1 Cert 2014')
    - REFRESH SECURITY TYPE(SSL)

## Business Partners with different CA requirements (pre V8)







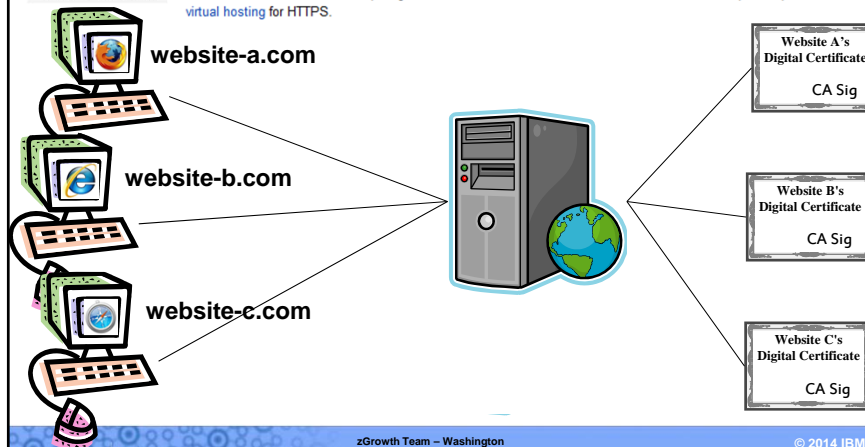
## Server Name Indication



### Server Name Indication

From Wikipedia, the free encyclopedia

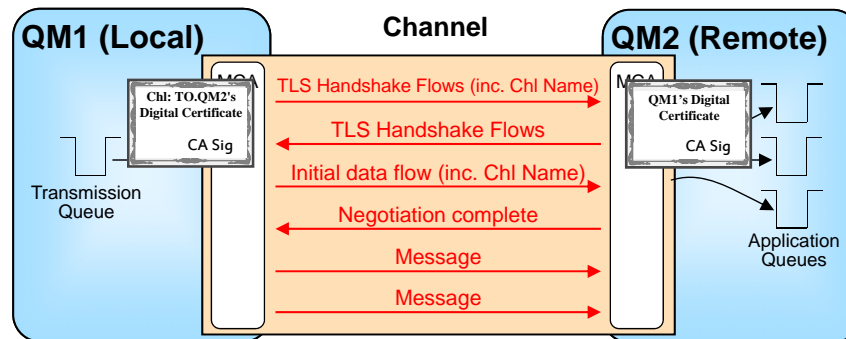
**Server Name Indication (SNI)** is an extension to the [TLS protocol](#)<sup>[1]</sup> that indicates what hostname the client is attempting to connect to at the start of the handshaking process. This allows a server to present multiple certificates on the same IP address and port number and hence allows multiple secure ([HTTPS](#)) websites (or any other [Service over TLS](#)) to be served off the same IP address without requiring all those sites to use the same certificate. It is the conceptual equivalent to [HTTP/1.1 virtual hosting](#) for HTTPS.



zGrowth Team – Washington

© 2014 IBM Corporation

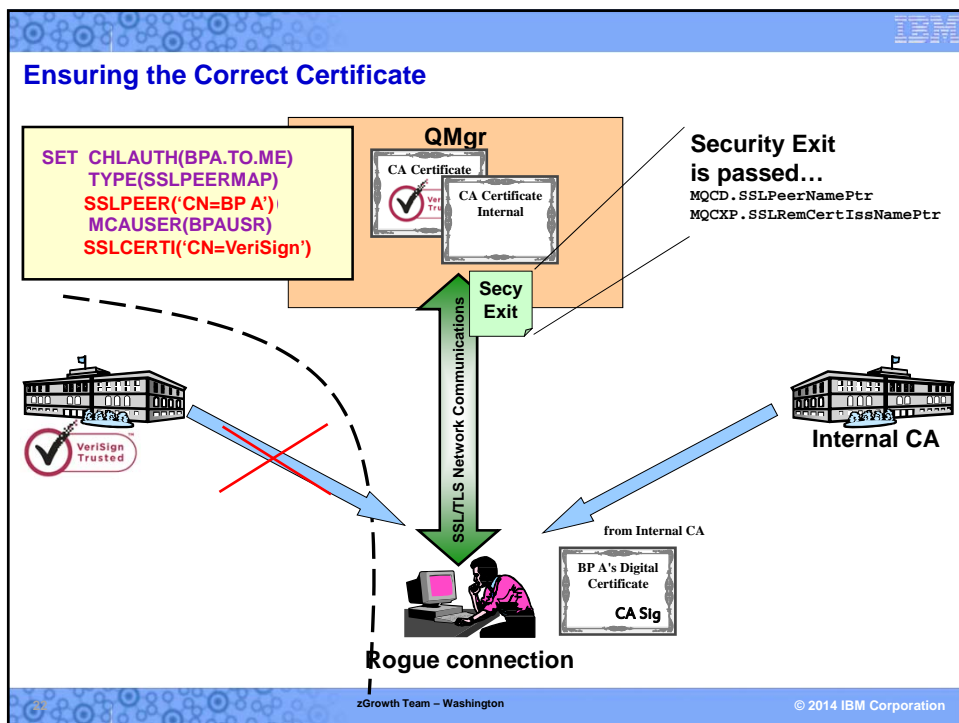
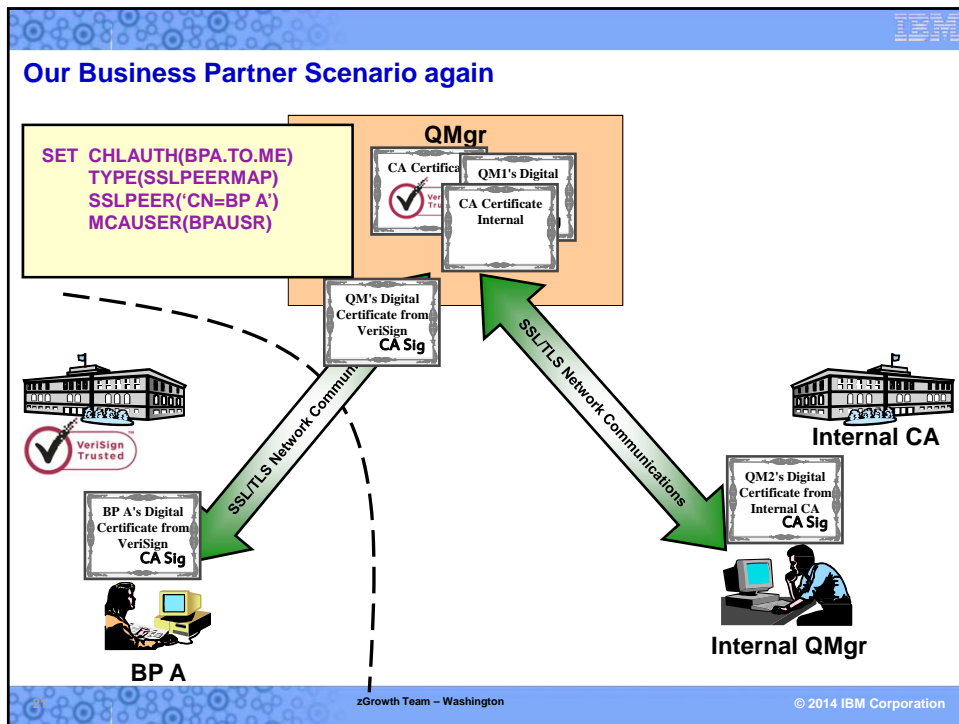
## Using Server Name Indication (SNI) with a channel name



- **Both ends of the channel must be at the new release**
- Only TLS can be used, no SSL
  - Only certain cipherspecs will be able to supply this behaviour
- JSSE doesn't yet support SNI
  - So Java client can't make use of it
- If old sender/client used, we'd only detect that we needed to supply a different certificate after completion of the handshake and will fail the connection, if it hasn't already failed due to using the wrong certificate!

zGrowth Team – Washington

© 2014 IBM Corporation



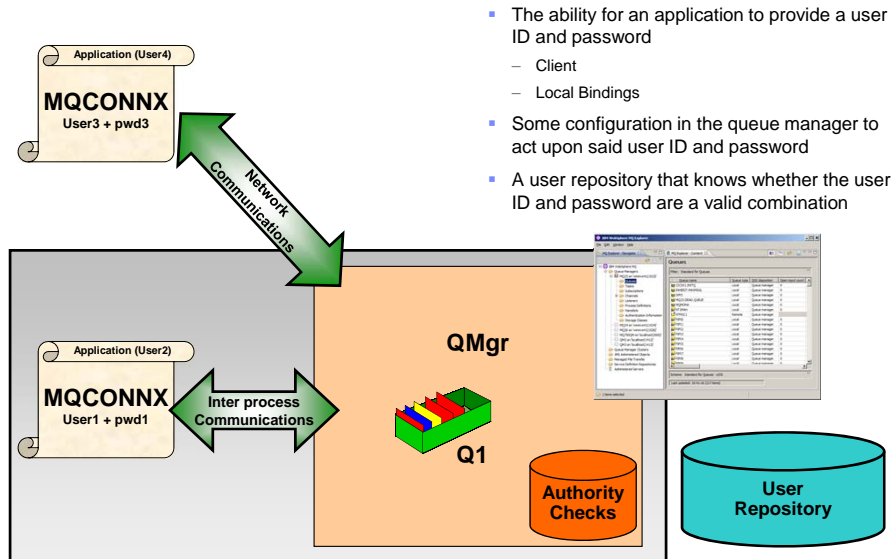
## Summary

- Changes for Channels using SSL/TLS Certificates
  - Single Queue Manager Certificate
    - `ALTER QMGR CERTLABL('My certificate name')`
  - Per Channel Certificate
    - `ALTER CHANNEL ... CERTLABL('This channel certificate')`
  - Certificate Matching
    - `SET CHLAUTH('*')`  
`TYPE(SSLPEERMAP)`  
`SSLPEER('CN=Morag Hughson')`  
`SSLCERTI('CN=IBM CA')`  
`MCAUSER('hughson')`

## Agenda

- Channel Authentication with Hostnames
- MQ Enhancements for Digital Certificates
- Connection Authentication Enhancements

## Connection Authentication – What is it?

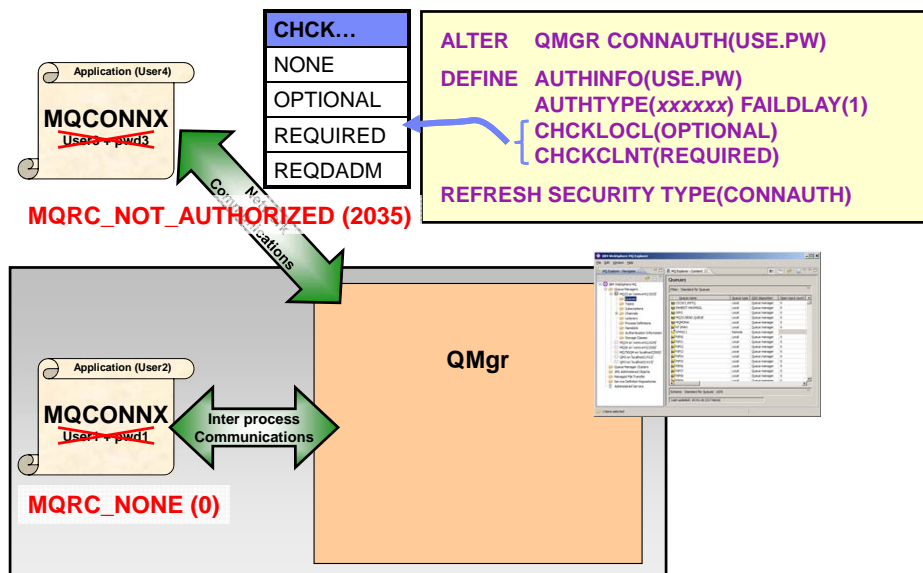


25

zGrowth Team – Washington

© 2014 IBM Corporation

## Connection Authentication – Configuration

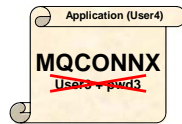


26

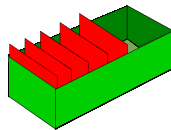
zGrowth Team – Washington

© 2014 IBM Corporation

## Connection Authentication – Error notification



**MQRC\_NOT\_AUTHORIZED (2035)**

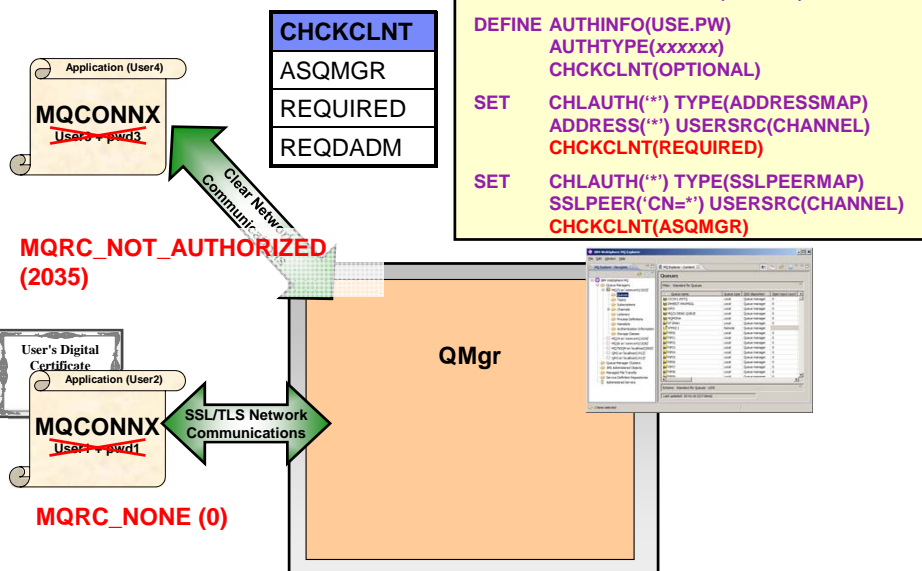


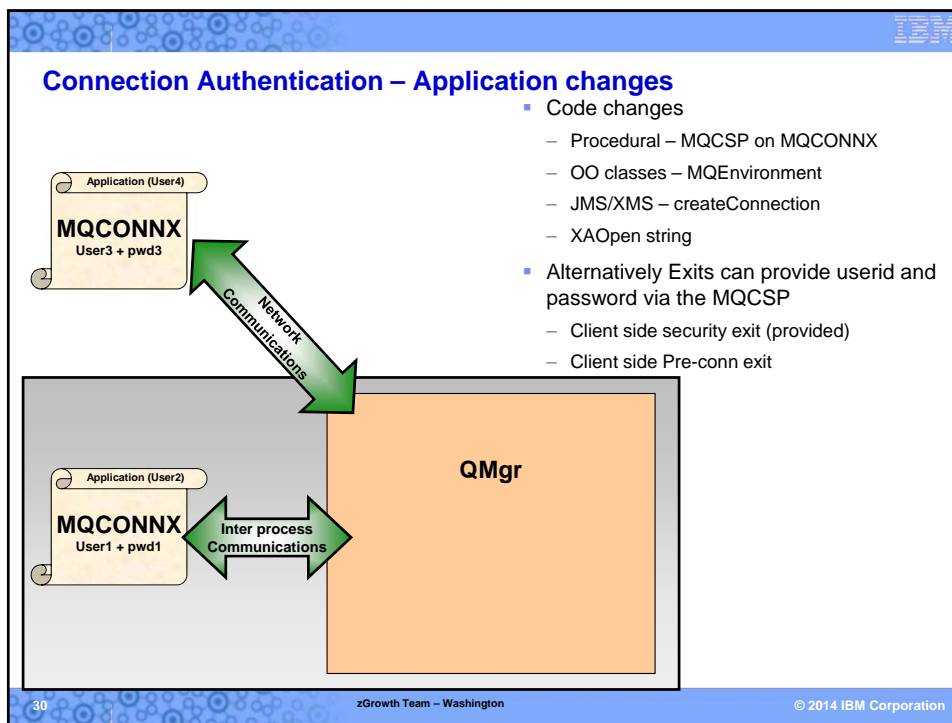
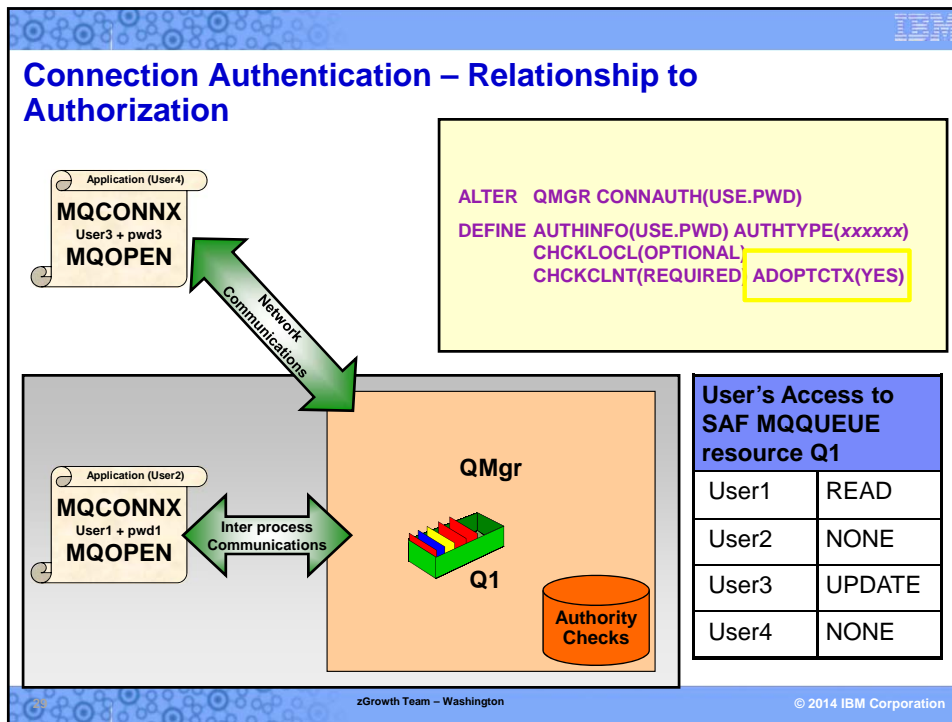
SYSTEM.ADMIN.QMGR.EVENT

**ALTER QMGR AUTHOREV(ENABLED)**

- Application
  - MQRC\_NOT\_AUTHORIZED (2035)
- Administrator
  - Error message
- Monitoring Tool
  - Not Authorized Event message (Type 1 – Connect)
  - MQRQ\_CONN\_NOT\_AUTHORIZED (existing)
    - Connection not authorized.
  - MQRQ\_CSP\_NOT\_AUTHORIZED (new)
    - User ID and password not authorized.
  - Additional field to existing connect event
    - MQCACF\_CSP\_USER\_IDENTIFIER

## Connection Authentication – Configuration Granularity





## Procedural MQI changes

- MQCSP structure
  - Connection Security Parameters
  - User ID and password
- MQCNO structure
  - Connection Options
- WebSphere MQ V6
  - Passed to OAM (Dist only)
  - Also passed to Security Exit
    - Both z/OS and Distributed
    - MQXR\_SEC\_PARMS
- WebSphere MQ V8
  - Acted upon by the queue manager (all platforms)

```
MQCNO cno = {MQCNO_DEFAULT};  
  
cno.Version = MQCNO_VERSION_5;  
  
cno.SecurityParmsPtr = &csp;  
  
MQCONN(XQName,  
        &cno,  
        &hConn,  
        &CompCode,  
        &Reason);
```

```
MQCSP csp = {MQCSP_DEFAULT};  
  
csp.AuthenticationType = MQCSP_AUTH_USER_ID_AND_PWD;  
csp.CSPUserIdPtr       = "johnson";  
csp.CSPUserIdLength    = 7;          /* Max: MQ_CLIENT_USER_ID_LENGTH */  
csp.CSPPasswordPtr     = "passwd";  
csp.CSPPasswordLength  = 8;          /* Max: MQ_CSP_PASSWORD_LENGTH   */
```

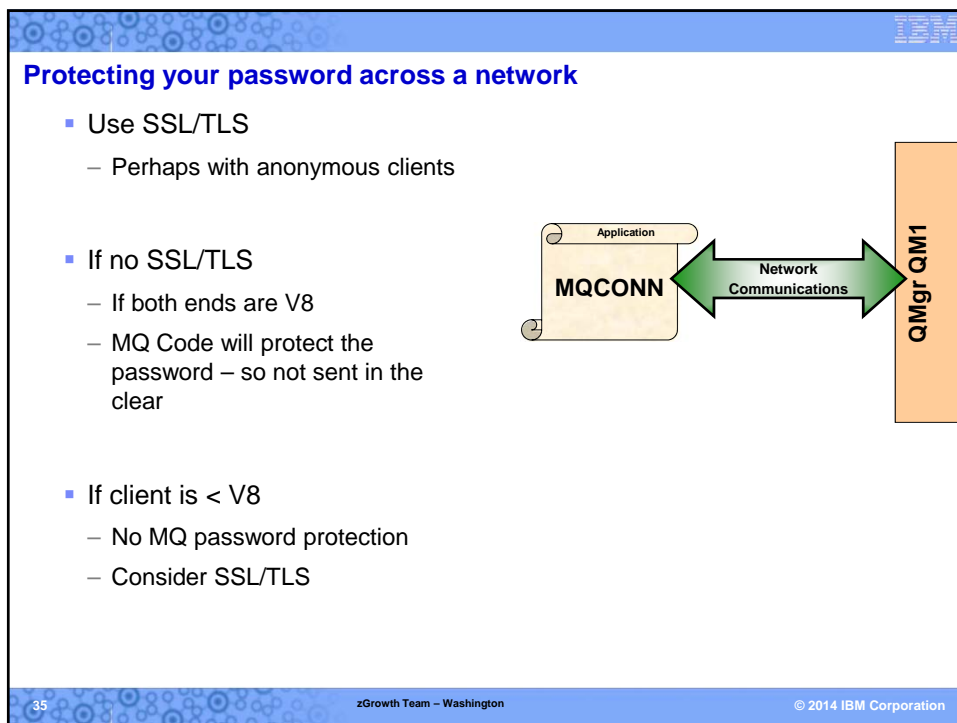
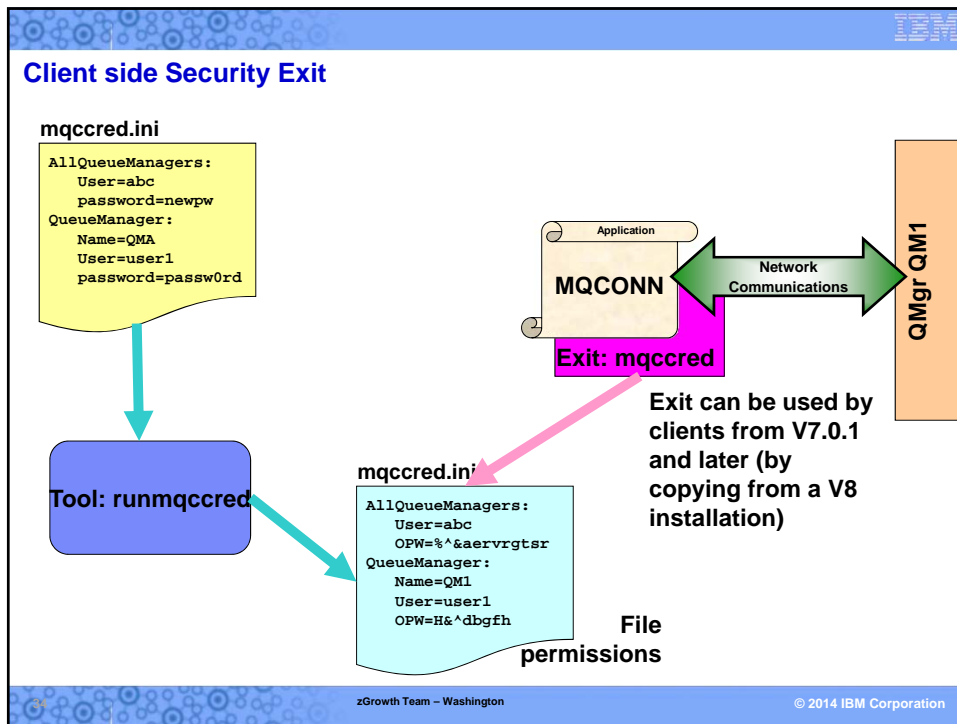
## Object Oriented MQ classes changes

```
MQEnvironment.properties = new Hashtable();  
MQEnvironment.userID = "hughson";  
MQEnvironment.password = "passwd";  
  
System.out.println("Connecting to queue manager");  
MQQueueManager qMgr = new MQQueueManager(QMName);
```

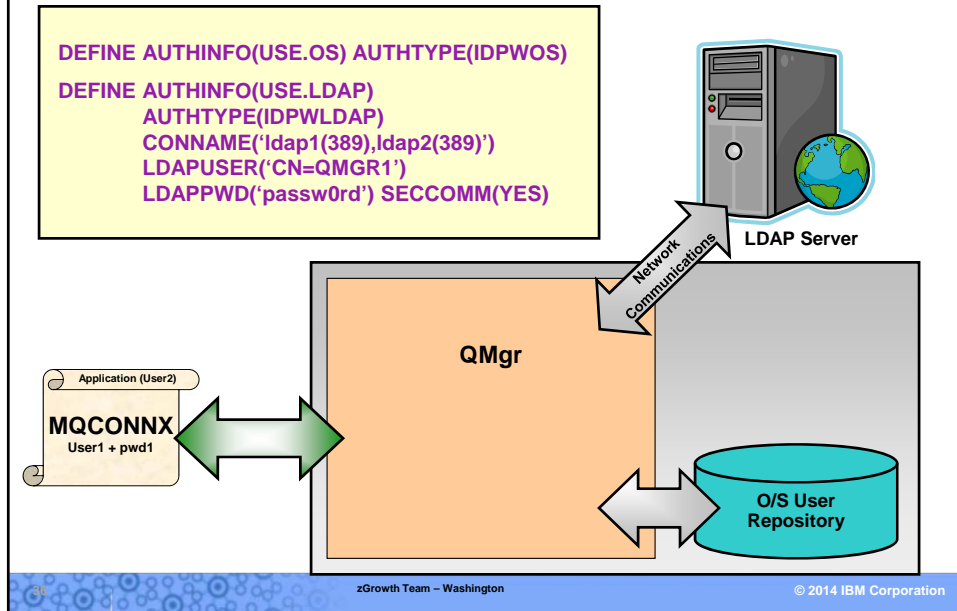
## JMS/XMS classes changes

```
cf = getCF();  
  
System.out.println("Creating the Connection with UID and Password");  
Connection conn = cf.createConnection("hughson", "passwd");
```

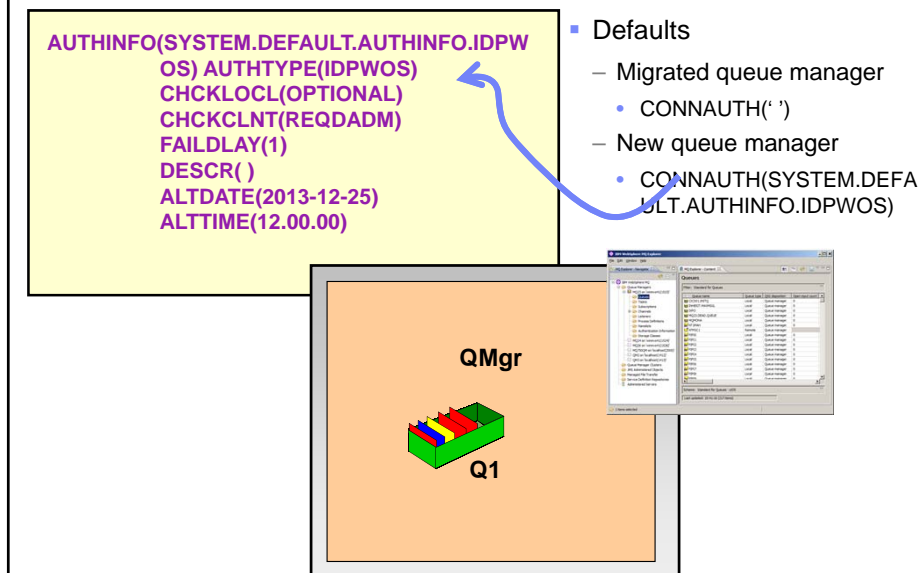




## Connection Authentication – User Repositories



## Migration / Defaults



## Summary

- Channel Authorization (CHAUTH) Rules now support hostnames and domains
  - `SET CHLAUTH('APPL1.*') TYPE(ADDRESSMAP) ADDRESS('*.matahari.com') USERSRC(NOACCESS)`
- Queue Managers now support digital certificates labels at the queue manager and channel levels
- Connection Authentication enhanced to support userid and password checking with the local OS
  - User exits (e.g. CSQ4BCX3) may no longer be required

## Questions?