Better access control and security using a single portal

Todd Rob Wyatt – IoPT Consulting

@trodrob
http://ioptconsulting.com
http://t-rob.net

Peter D’Agosta – Avada Software

@AvadaSoftware

http://www.avadasoftware.com/

Version control

Security related documents have a limited shelf life. Find the latest version of this document or more in this series at the source:

http://t-rob.net/links

This document last updated:

Date	Who	Change
15MAY2013	TRW	Added change control, source links

Middleware Access Control

1. Highlights of MQ 7.1 & MQ 7.5 security
2. Establish default levels of security
   - create MQ security Policies
3. Logical separation of production & non-prod MQ environments
4. Allow an administrative gateway to provide centralized MQ administration, monitoring, and auditing
5. Extend the model to Middleware stack
   - WAS, MQ, WMB, Perimeter

Further Reference on this Topic

- Find the latest version of this deck at http://t-rob.net/links On the same page you can also subscribe to receive update notifications via email when new versions are posted.
  - Please see the "What's New in WebSphere MQ v7.1 Security" slides as prepared and presented by Morag Hughson at the 2011 WSTC conference in Berlin and available for download at http://t-rob.net/links.
  - Although there is some overlap, this presentation is intended to cover different ground. Either presentation will stand alone but consider reviewing both for more comprehensive coverage.
- Because there is so much material to cover, expect to see more new content with a "What's new in WebSphere MQ v7.1 Security" theme. It may show up as presentations, in articles, video or other formats but all of it will be indexed at t-rob.net where you can subscribe using RSS or via email list.

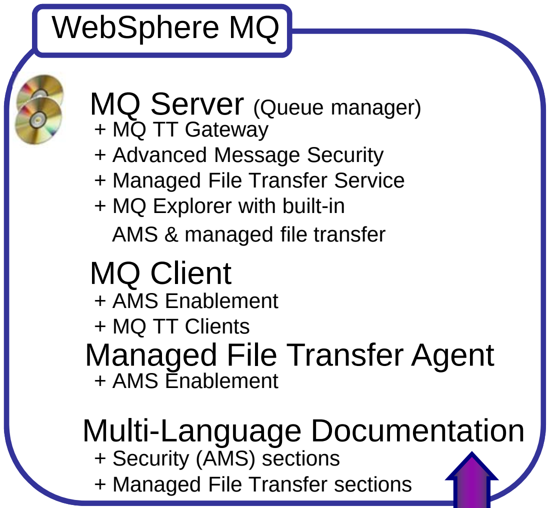
MQ 7.1 security highlights

New Feature	Benefits	*Details*
IP address filtering	Allow or deny connections based on IP address	Blocks IP addresses at the listener and provides allow/deny functionality expressed as per-channel rules
User ID Mapping	Fine grained mapping of connection details to MCAUSER values	Allow or deny connections based on the ID presented in the connection request and map the ID to an MCAUSER
User ID blocking	Controls which user IDs can use which channels	After all exits and mapping are completed, a final check is made of the derived MCAUSER against these rules
Per-channel DLQ settings	Simplifies B2B and other cross-border security	New USEDLQ channel attribute controls which channels will use the DLQ for undeliverable messages.
Infrared360 QmgrBackup Service	Supported method to back up object configurations and security settings	Saves Qmgr and Security configuration for v7.1 QMgrs.

WebSphere MQ V7.5

- Integrated Messaging Offering
  - Single install, packaging & tooling for all Messaging options
  - Reduce time to value, simplify usage

- What' s being delivered?
  - Integration of MQ with MQ FTE, MQ AMS and MQ Telemetry
  - Single install, common integrated tooling and management, simplified licensing and entitlements
  - More complete, easy to use messaging infrastructure, enabling you to gain full range of messaging, swiftly & easily



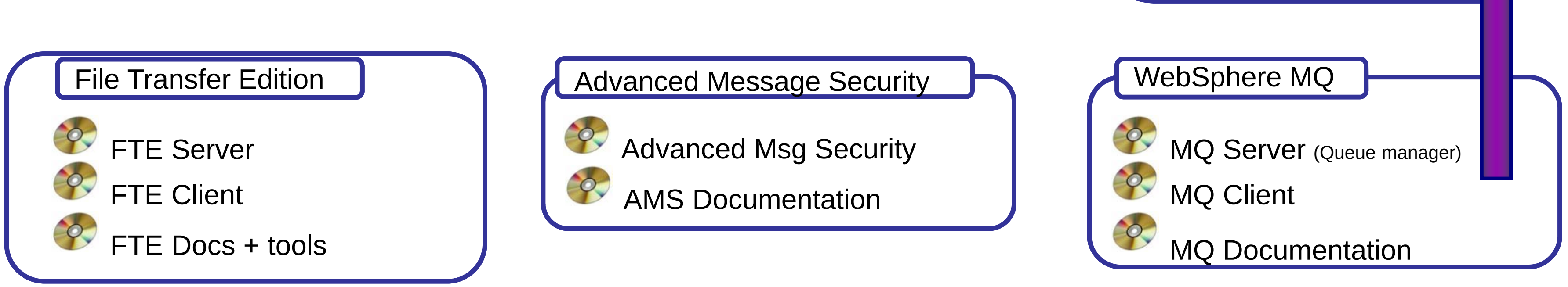
WebSphere MQ

MQ Server (Queue manager)
+ MQ TT Gateway
+ Advanced Message Security
+ Managed File Transfer Service
+ MQ Explorer with built-in
AMS & managed file transfer

MQ Client
+ AMS Enablement
+ MQ TT Clients

Managed File Transfer Agent
+ AMS Enablement

Multi-Language Documentation
+ Security (AMS) sections
+ Managed File Transfer sections



File Transfer Edition

FTE Server

FTE Client

FTE Docs + tools

Advanced Message Security

Advanced Msg Security

AMS Documentation

WebSphere MQ

MQ Server (Queue manager)

MQ Client

MQ Documentation

Security: Channel Access Control

- Simplifying configuration for channel access
  - Clients and queue managers
- SET CHLAUTH definitions control who can use channels
  - Name mapping & Access blocking
- Easy to test rules that you define
  - DISPLAY CHLAUTH can “execute” rules

  - Display via Infrared360 CHLAUTH screen

Security: Channel Access Control

- **MIGRATION NOTE**:
  - Standard rules block clients on new queue managers
  - Secure by default
  - Migrated queue managers behave as *before* until you enable the rules
  - Queue manager attribute CHLAUTH(ENABLED|DISABLED) provides overall control

  - Display via Infrared360 Connections Screen

Middleware Access Control

1. Highlights of MQ 7.1 & MQ 7.5 security
2. Establish default levels of security
   - create MQ security Policies
3. Logical separation of production & non-prod MQ environments
4. Allow an administrative gateway to provide centralized MQ administration, monitoring, and auditing
5. Extend the model to Middleware stack
   - WAS, MQ, WMB, Perimeter

Establish MQ Security Policies
Channel Access Policy (1)

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

“Make sure our system is completely locked down”

Establish MQ Security Policies
Channel Access Policy (2)

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

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Shetland') MCAUSER(BANK123)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Orkney') MCAUSER(BANK456)
```

““Business Partners must connect using SSL.

Map their access from the certificate DNs”

Establish MQ Security Policies
Channel Access Policy (3)

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

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Shetland') MCAUSER(BANK123)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Orkney') MCAUSER(BANK456)

SET CHLAUTH(SYSTEM.ADMIN.SVRCONN) TYPE(ADDRESSMAP)
ADDRESS(‘192.210.1.68') MCAUSER(ADMUSER)
```

“Administrators connect using Infrared360, but don't use SSL.
Map their access to “1” IP Address of the Infrared360 Server”

Establish MQ Security Policies
Channel Access Policy (4)

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

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Shetland') MCAUSER(BANK123)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Orkney') MCAUSER(BANK456)

SET CHLAUTH(SYSTEM.ADMIN.SVRCONN) TYPE(ADDRESSMAP)
ADDRESS(‘192.210.30.68’) MCAUSER(ADMUSER)

SET CHLAUTH(CLUS.*) TYPE(QMGRMAP)
QMNAME(CLUSQM*) MCAUSER(CLUSUSR) ADDRESS(‘192.210.30.*')
```

Our internal cluster doesn't use SSL, but we must ensure only the correct queue managers can connect into the cluster

Establish Default Levels of Security: Blocking at the Listener

- Single list of IP address patterns
- NOT A REPLACEMENT FOR AN IP FIREWALL
  - Temporary blocking
  - Blocking until IP firewall updated
  - Shouldn't be many entries in the list
- Blocked before any data read from the socket
  - i.e. before SSL Handshake
  - Before channel name or userid is known
- Avoiding DoS attack
  - Really the place of the IP firewall
  - Simplistic 'hold' of inbound connection to avoid reconnect busy loop
- Network Pingers if blocked don't raise an alert
  - Immediate close of socket with no data not considered a threat

```
SET CHLAUTH(*) TYPE(BLOCKADDR) ADDRLIST('9.20.*', '192.168.30.10')
```

Middleware Access Control

1. Highlights of MQ 7.1 & MQ 7.5 security
2. Establish default levels of security
   - create MQ security Policies
3. Logical separation of production & non-prod MQ environments
4. Allow an administrative gateway to provide centralized MQ administration, monitoring, and auditing
5. Extend the model to Middleware stack
   - WAS, MQ, WMB, Perimeter

Isolating Production & Non-Production Data

- Mixing Production & Non-Production Servers is more dangerous (career-wise) than mixing *Matter & Anti-Matter*
- Sometimes companies use a non-production machine

  as the high availability failover for a production machine.
  - What's the best way to *prevent an accident* where

    a non-production queue manager (or client),

    connects to a production queue manager?
  - Some companies rely on firewall rules
    - However, that is hard to do when a non-prod

      machine doubles as a production back-up.

Security Features to Separate Prod & Non-Prod Data

New Feature	*Benefits*	*Details*
IP address filtering	Allow or deny connections based on IP address	Blocks IP addresses at the listener and provides allow/deny functionality expressed as per-channel rules
User ID Mapping	Fine grained mapping of connection details to MCAUSER values	Allow or deny connections based on the ID presented in the connection request and map the ID to an MCAUSER
Remote Queue Manager Name	User Mappings can also use the name of the remote queue manager	Allow or Deny Channel Connections based on the name of the remote queue manager.
Certificate DN mapping	Finer granularity for matching certificates	Extends SSLPEER functionality to lists of DNs and with expanded regex- type pattern matching and allow/deny capability
User ID blocking	Controls which user IDs can use which channels	After all exits and mapping are completed, a final check is made of the derived MCAUSER against these rules

Isolate production & non-production with MQ 7.1

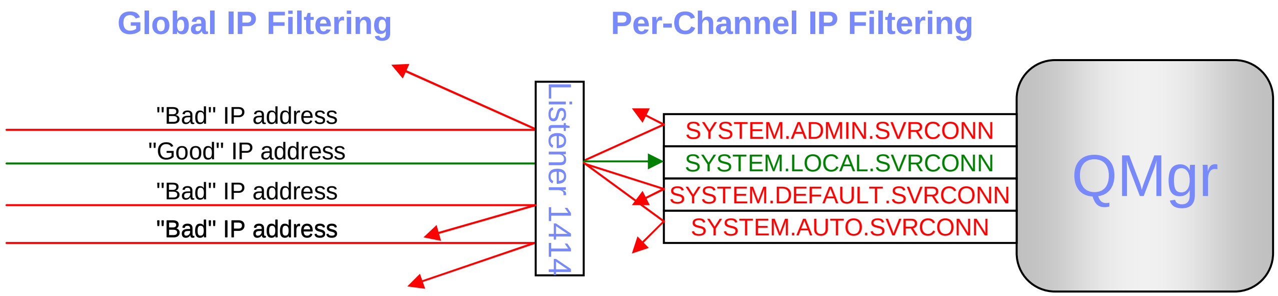
- Use IP Address filtering for isolation
  - A. Create a generic wildcard rule that blocks all IP addresses by default
  - B. A specific rule can then permit (whitelist) specific IP addresses / subnets
  - – This is essentially a mini-firewall under the control of MQ administration
  - – Note: Standard firewalls should also be used.
  - – Note: Try to avoid production & non-production in the same sub-network

Isolate production & non-production with MQ 7.1

- Use Queue Manager Names for isolation
  - A key rule for a successful infrastructure is “Keep it Simple”
  - Simple MQ naming conventions help avoid ambiguity
  - Consider a new Queue Manager naming convention:
    - The first letter of a queue manager should identify the environment
      - P for Prod, D for Dev, T for Test, S for Staging …
    - A generic rule can then allow access to other queue managers from matching environments
  - Even better a specific receiver channel (FROMQMGR.TOQMGR) can be configured to only accept connections from QM1 from specific IP address(es)

New feature: IP address filtering

- Prior to v7.1, the ability to validate connection requests based on IP address was only possible using security exits.
- This functionality is now included natively to filter connection requests based on the IP address of the requestor.
- Two types: Per-channel rules and global blocking rule.
- Global blocking rules occur at the listener before the channel name is known and therefore take precedence over per-channel rules.
- Per-channel rules match from least-specific to most-specific, similar to generic OAM profiles.



```
SET CHLAUTH('*') TYPE(BLOCKADDR) +
      ADDRLIST('192.168.1.*')
```

```
SET CHLAUTH('SYSTEM.*') TYPE(ADDRESSMAP) +
      ADDRESS('*') USERSRC(NOACCESS)

SET CHLAUTH('SYSTEM.LOCAL.SVRCONN') TYPE(ADDRESSMAP) +
      ADDRESS('127.0.0.*')  USERSRC(CHANNEL)
```

IP address filtering guidelines

- Keep in mind: *permitting specific IPs is better than forbidding specific IPs*. A permitted list (also known as whitelisting) says "here is an enumerated list of authorized requestors." A restricted list says "out of the practically infinite number of possible requestors, here are a few 'bad' ones." It is impractical to attempt to list all bad addresses.
- CHLAUTH TYPE(BLOCKADDR) is implemented as a blacklist because it is not intended to be the primary means of connection filtering. Use BLOCKADDR to temporarily deal with transient problems such as a runaway client, or to deal with specific issues such as port scanners causing MQ to cut FDC files.
- CHLAUTH TYPE(ADDRESSMAP) rules are hierarchical. With *all other parameters being equal*, the most specific matching profile name takes precedence. This allows you to establish a deny-all policy followed by specific whitelist rules as shown on the previous slide.
- When other parameters are not equal, multiple rules may apply according to a precedence order.

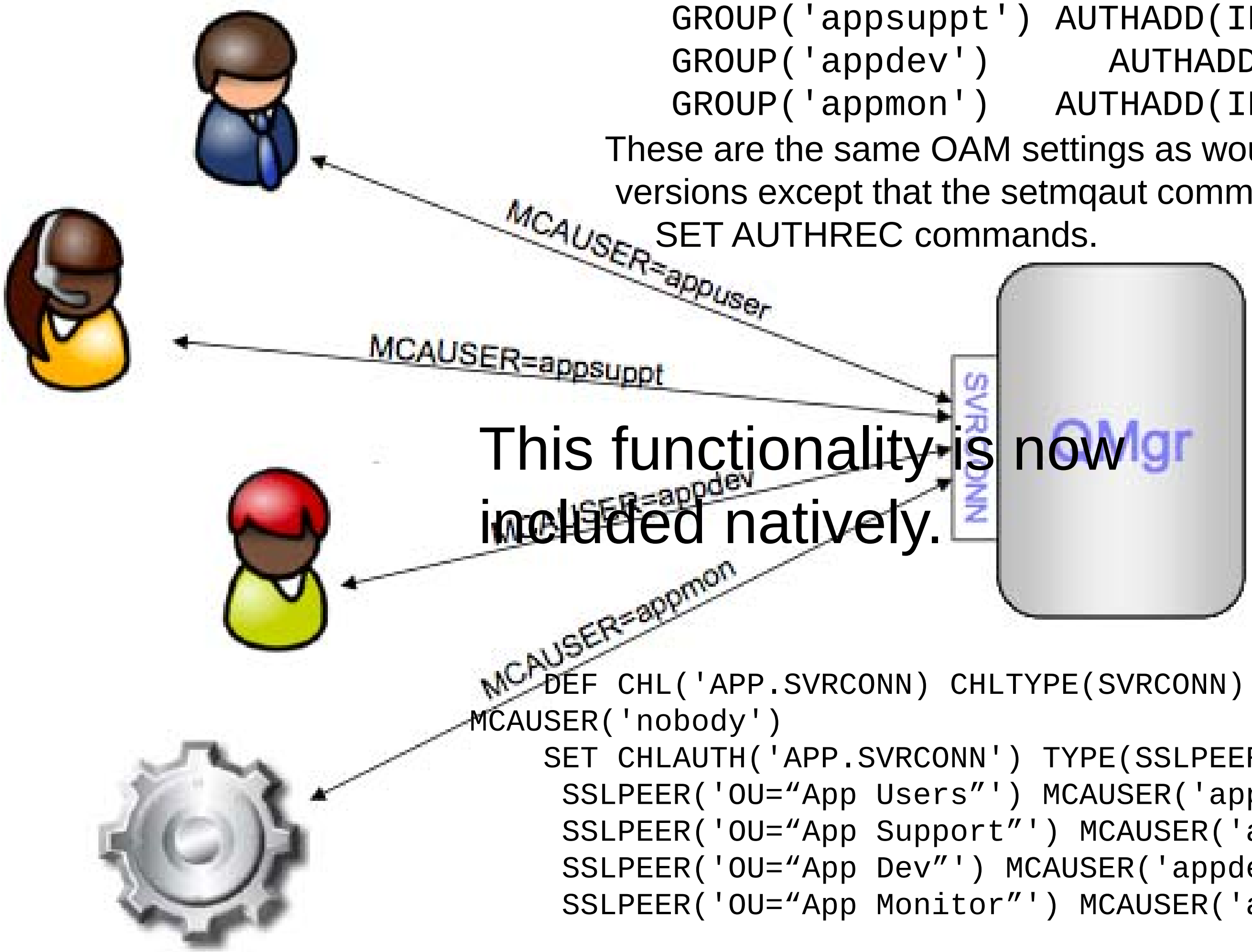
Dynamic mapping of MCAUSER

- The channel's MCAUSER determines the ID used for authorization, the same as in previous versions of MQ.
- Authorization of remote entities is only as granular as the number of MCAUSER values.
- Prior to v7.1, administrators had a choice of either hard coding the value in the channel definition and defining many channels, or implementing a security exit and configuring the validation criteria outside of MQ.
- New in v7.1 is the ability to configure dynamic mapping of the MCAUSER based on a variety of validation criteria. This allows the use of fewer channels to support the same or even finer granularity of authorization than was available in prior versions, and with all configuration details managed natively using standard WebSphere MQ tools.
- Set CHLAUTH: http://bit.ly/sc83Ol
- Channel Authentication records: http://bit.ly/veN5C7

Example of dynamic MCAUSER mapping

```
SET AUTHREC PROFILE(APP.QUEUE) OBJTYPE(QUEUE) +
    GROUP('appuser')  AUTHADD(INQ, DSP, BROWSE, PUT, GET)
    GROUP('appsuppt') AUTHADD(INQ, DSP, BROWSE)
    GROUP('appdev')     AUTHADD(INQ, DSP)
    GROUP('appmon')   AUTHADD(INQ)
```

These are the same OAM settings as would have been required in prior versions except that the setmqaut commands have been migrated to v7.1 SET AUTHREC commands.



This functionality is now included natively.

```
    DEF CHL('APP.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
MCAUSER('nobody')
    SET CHLAUTH('APP.SVRCONN') TYPE(SSLPEERMAP) +
     SSLPEER('OU=“App Users”') MCAUSER('appuser')
     SSLPEER('OU=“App Support”') MCAUSER('appsuppt')
     SSLPEER('OU=“App Dev”') MCAUSER('appdev')
     SSLPEER('OU=“App Monitor”') MCAUSER('appmon')
```

Any connection requests not matching one of the 4 CHLAUTH records is refused. The result is that a single channel definition serves four different security roles for the same application. The DEF CHL, SET CHLAUTH and SET AUTHREC definitions are all managed using standard MQ admin tools, possibly all in the same MQSC script.

Middleware Access Control

1. Highlights of MQ 7.1 & MQ 7.5 security
2. Establish default levels of security
   - create MQ security Policies
3. Logical separation of production & non-prod MQ environments
4. Allow an administrative gateway to provide centralized MQ administration, monitoring, and auditing
5. Extend the model to Middleware stack
   - WAS, MQ, WMB, Perimeter

Centralized MQ Administration with Administrative Gateway Queue Managers

- You started the roll-out at 12:01 AM Sunday morning & requested the mqm password to make the change. The password has expired & everyone waits.
- Alternatively, imagine that all administrative changes can only be done via a secure administrative gateway queue manager.
  - The administrative gateway queue manager must be the most secure and well monitored system in the data center. No access without SSL Certificates, VPN,…
    - See next slide for details
  - The other queue managers (The ones that are being administered).
    - Provide a separate listener and channel that can only be accessed from the secure administrative gateway queue manager.
- The Queue managers that are being administered will publish all of their changes. (This feature has been available since 7.0.1).
- Centralized “Read Only” Queue Managers, can allow queue and channel statistical analysis, capacity analysis & error log analysis & a change log to be accessed, *without* allowing any access to the production queue manager

Administrative Gateway Queue Managers
Blocking mqm IDs except via the Admin gateway QM

- The CHLAUTH BLOCKUSER rules take effect after all other rules and exits have been processed & the final value for MCAUSER has been determined.
- A special value *MQADMIN represents administrative users as defined for the local platform. This makes locking down admin access much easier
- Blank user IDs resolve to the ID of the MCA which matches *MQADMIN.
- This is a blacklist-only setting. However, it is possible to implement a limited deny/allow policy by altering the list of blocked names at different levels. E.g.
- SET CHLAUTH(*) TYPE(BLOCKUSER) USERLIST('nobody, *MQADMIN') SET CHLAUTH(QM.ADMIN.*) TYPE(BLOCKUSER) USERLIST('nobody')
- The first rule blocks administrative users and the MCAUSER 'nobody' (which prevents creating a user ID 'nobody' and putting it into an authorized group. Rules are hierarchical and the most specific one matches.
- The second rule provides a reduced blacklist for QM.ADMIN channels that allows administrators to use these. It is assumed here that some other CHLAUTH rule (e.g. SSLPEERMAP) has validated the QM.ADMIN channel

Administrative Gateway Queue Managers Depend on the Principal of Layered Defense

- First imagine a fort.
- Now add...
  - Perimeter buffer zone
  - Redundant fence
  - Razor wire
  - Flood lights
  - Spot lights
  - Moat
  - More!

- The idea is to place as many barriers as possible between your business assets and an attacker.

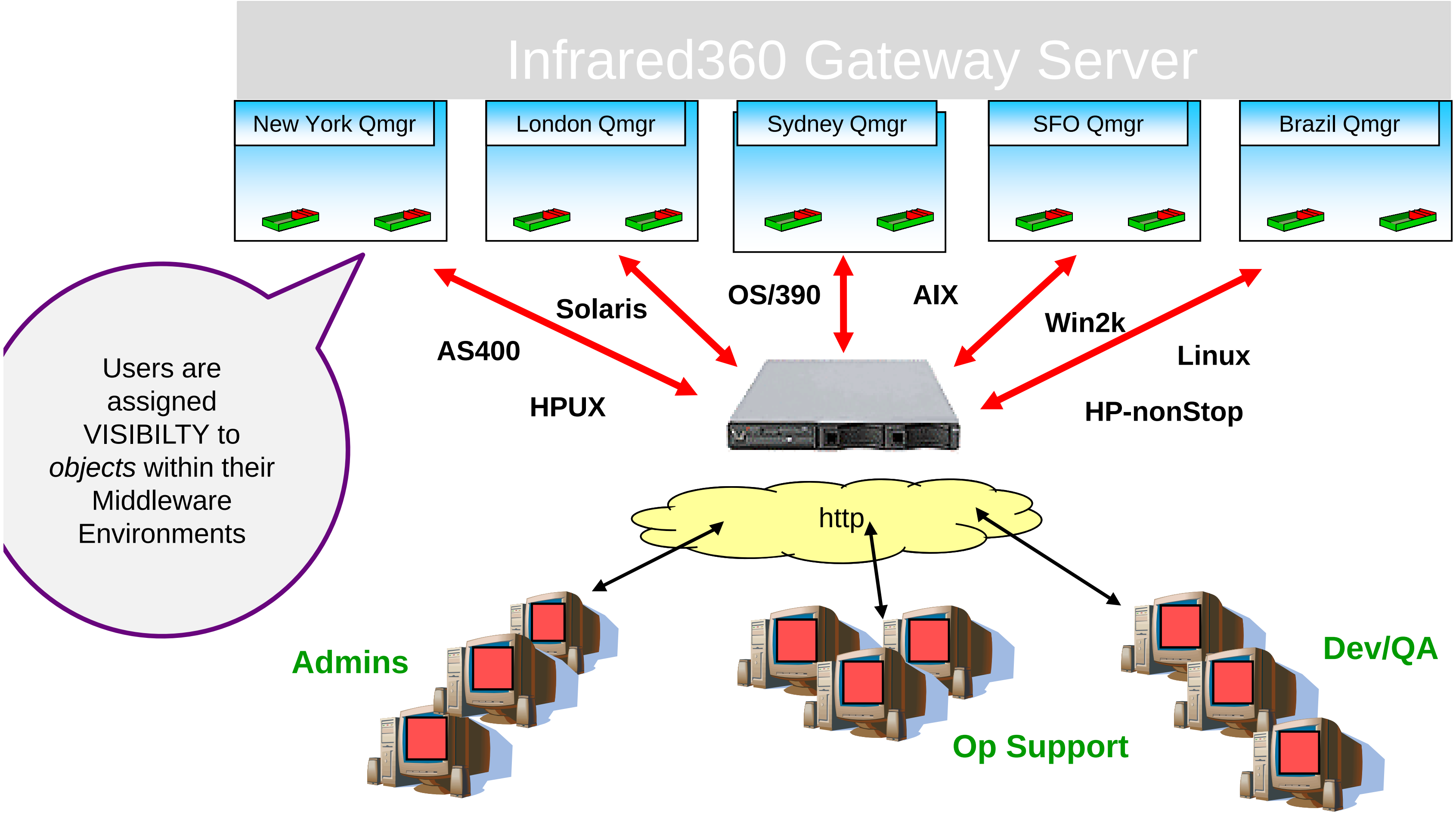
Administrative Gateway Queue Managers
Layered defense – WMQ style!

- MQ 7.1 Security policies should be used to ensure
  1. *Only* the gateway administrative server has privileged access to the other queue managers
  2. That SSL is required for the channel between the administrative gateway and target queue manager.

- On the Administrative gateway server
  - No application traffic is allowed via the administrative gateway

INFRARED 360
Avada Software

Example : The InfraRed360 Environment



Infrared360 Gateway Server

New York Qmgr

London Qmgr

Sydney Qmgr

SFO Qmgr

Brazil Qmgr

AS400

Solaris

OS/390

AIX

Win2k

Linux

HPUX

HP-nonStop

http

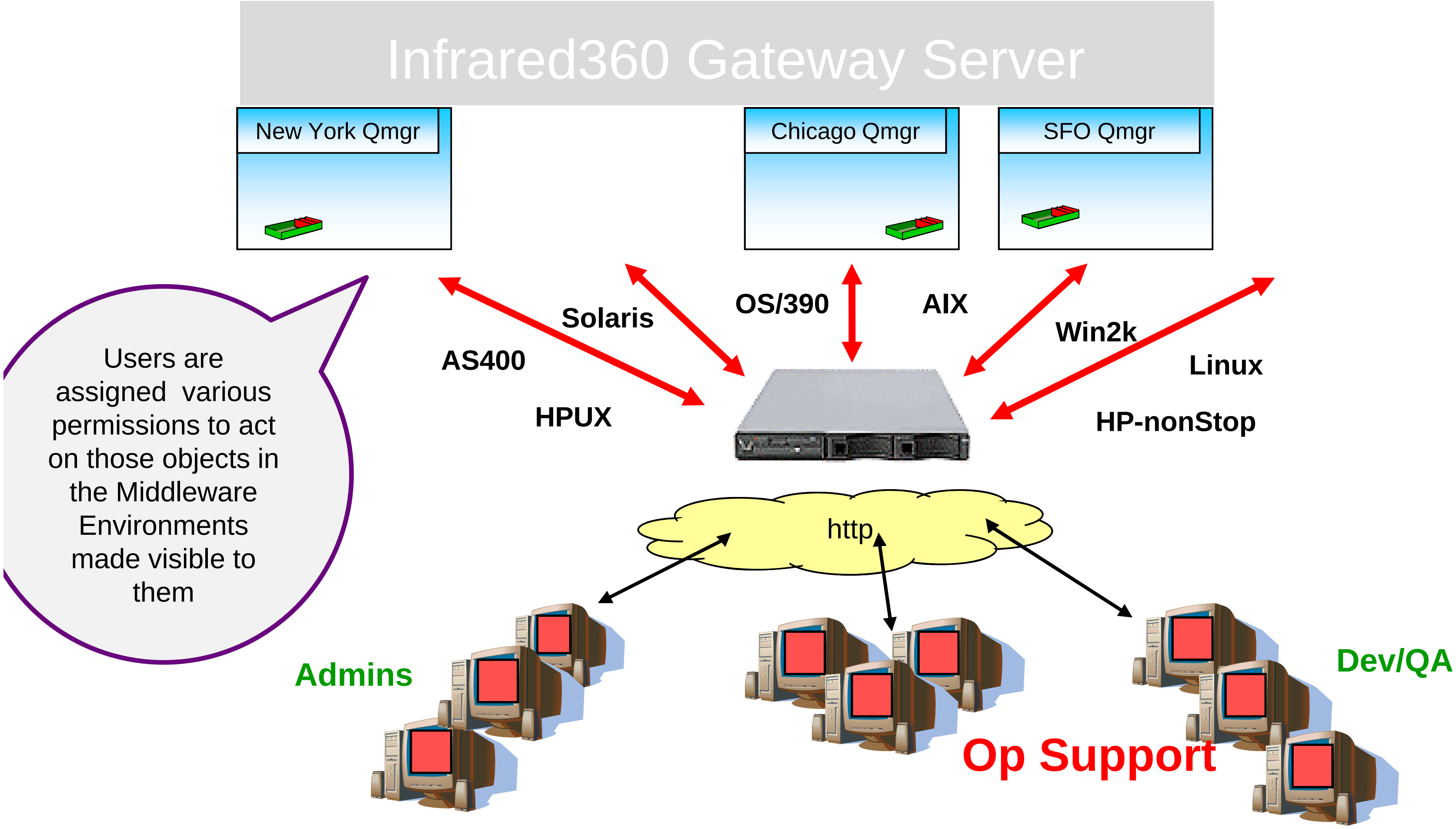
Admins

Op Support

Dev/QA

Users are assigned VISIBILTY to *objects* within their Middleware Environments

Example : The InfraRed360 Environment



Infrared360 Gateway Server

New York Qmgr

Chicago Qmgr

SFO Qmgr

AS400

Solaris

HPUX

OS/390

AIX

Win2k

Linux

HP-nonStop

http

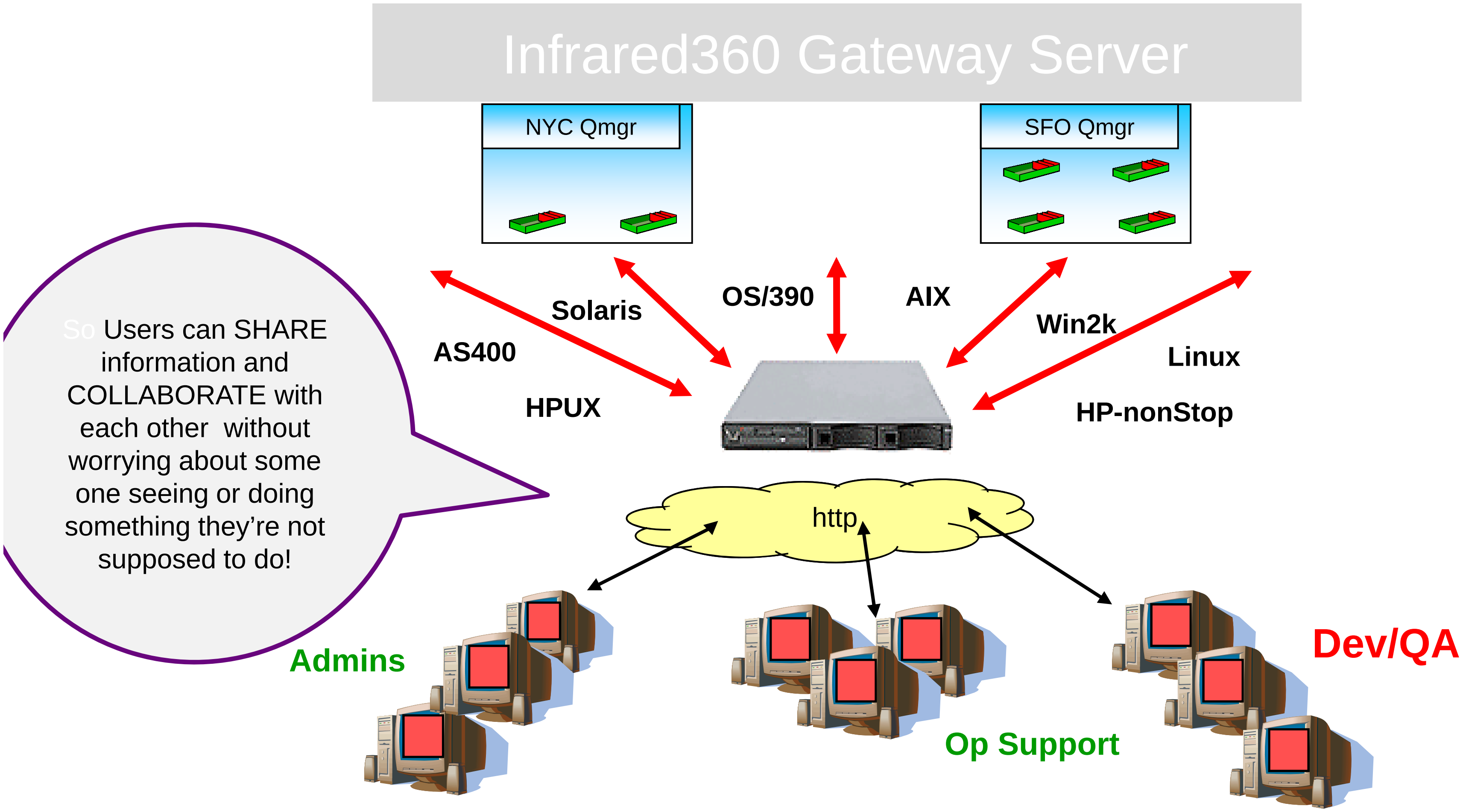
Admins

Op Support

Dev/QA

Users are assigned various permissions to act on those objects in the Middleware Environments made visible to them

Example : The InfraRed360 Environment

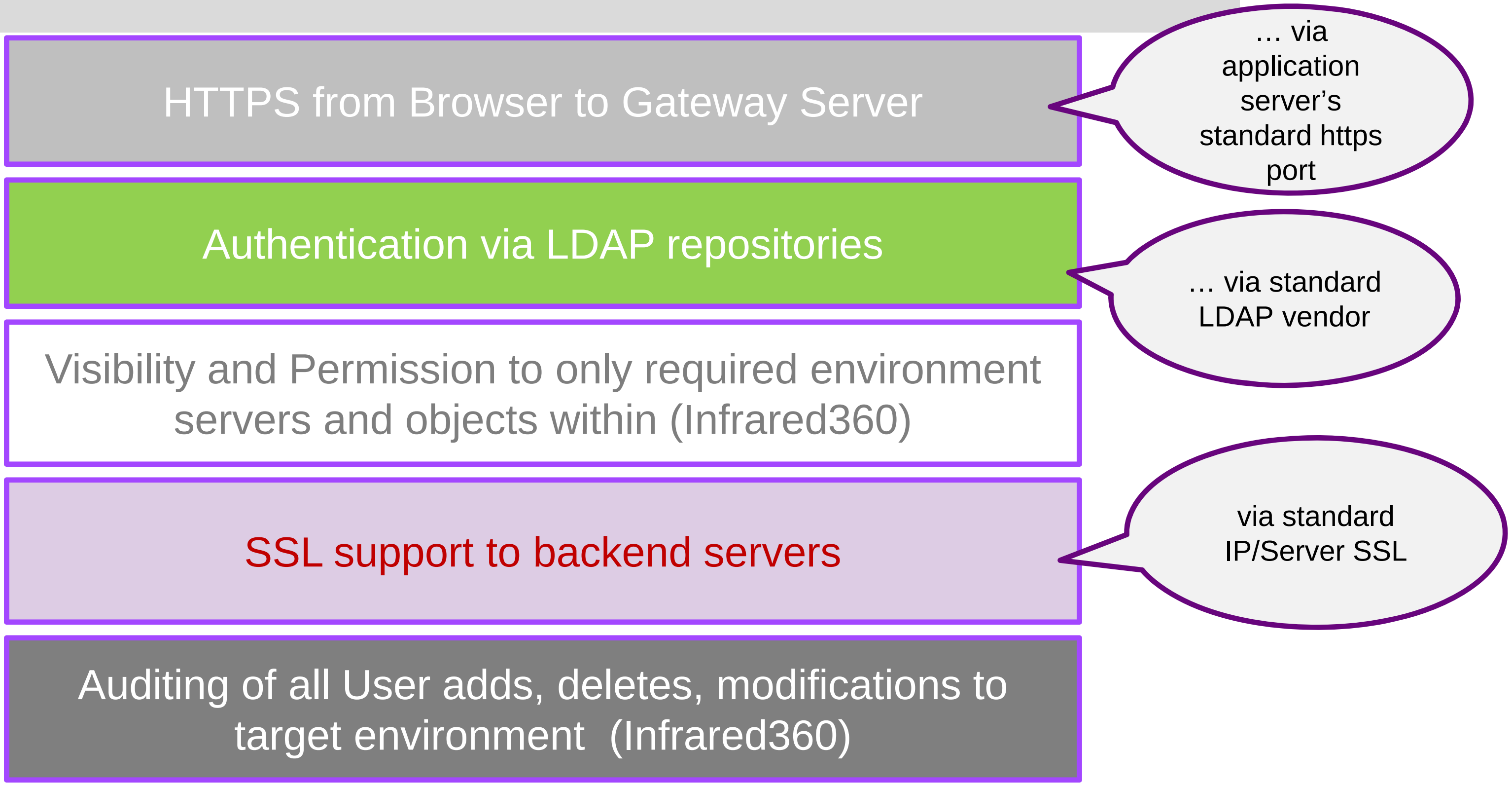


So Users can SHARE information and COLLABORATE with each other without worrying about some one seeing or doing something they're not supposed to do!

Administrative Gateway Queue Managers
Layered defense – WMQ style!

- Display of Logical Environments via
  - Admin access
  - Biz Unit Access
  - Consultant SME access
  - Guest – dev/opp access

Infrared360 Security Compliance!



HTTPS from Browser to Gateway Server

… via application server's standard https port

Authentication via LDAP repositories

… via standard LDAP vendor

Visibility and Permission to only required environment servers and objects within (Infrared360)

SSL support to backend servers

via standard IP/Server SSL

Auditing of all User adds, deletes, modifications to target environment (Infrared360)

MQ Security Practices *That Stand the Test of Time*

- Time is a key element of security strategies
  - Find/Fix vulnerabilities before:
    - an attacker can exploit them
    - an accident happens
  - Find vulnerabilities faster with centralized security monitoring
  - Deny administrative access with a gateway queue manager
    - Combining gateway QMs (layered defense) and centralied security monitoring increase the odds that an attack will be detected before it succeeds
  - Always separate Production (secure) & non-production (less secure)

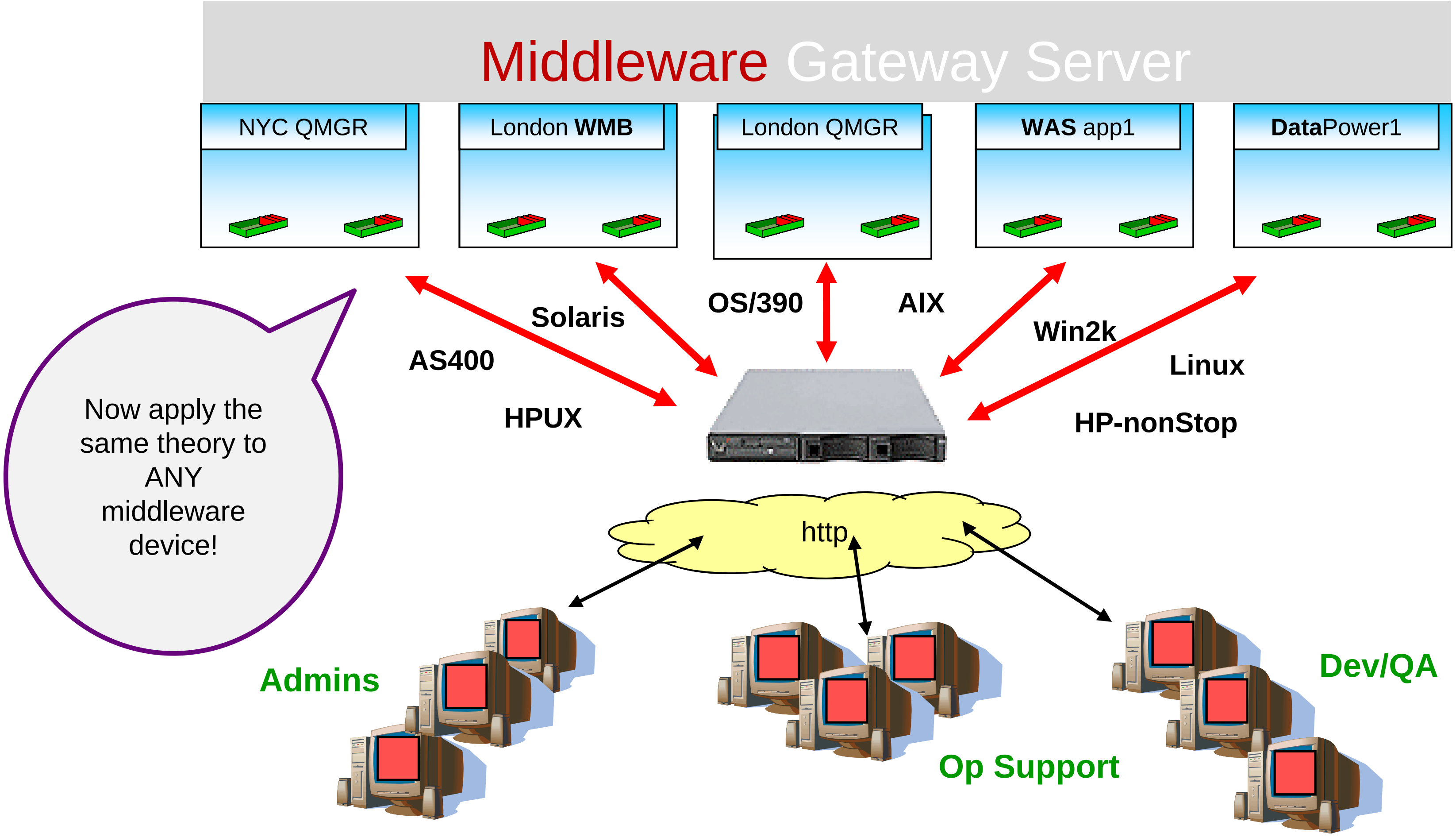
Keep defenses up to date with an MQ Risk Review Group

Middleware Access Control

1. Highlights of MQ 7.1 & MQ 7.5 security
2. Establish default levels of security
   - create MQ security Policies
3. Logical separation of production & non-prod MQ environments
4. Allow an administrative gateway to provide centralized MQ administration, monitoring, and auditing
5. Extend the model to Middleware stack
   - WAS, MQ, WMB, Perimeter

INFRARED 360
Avada Software

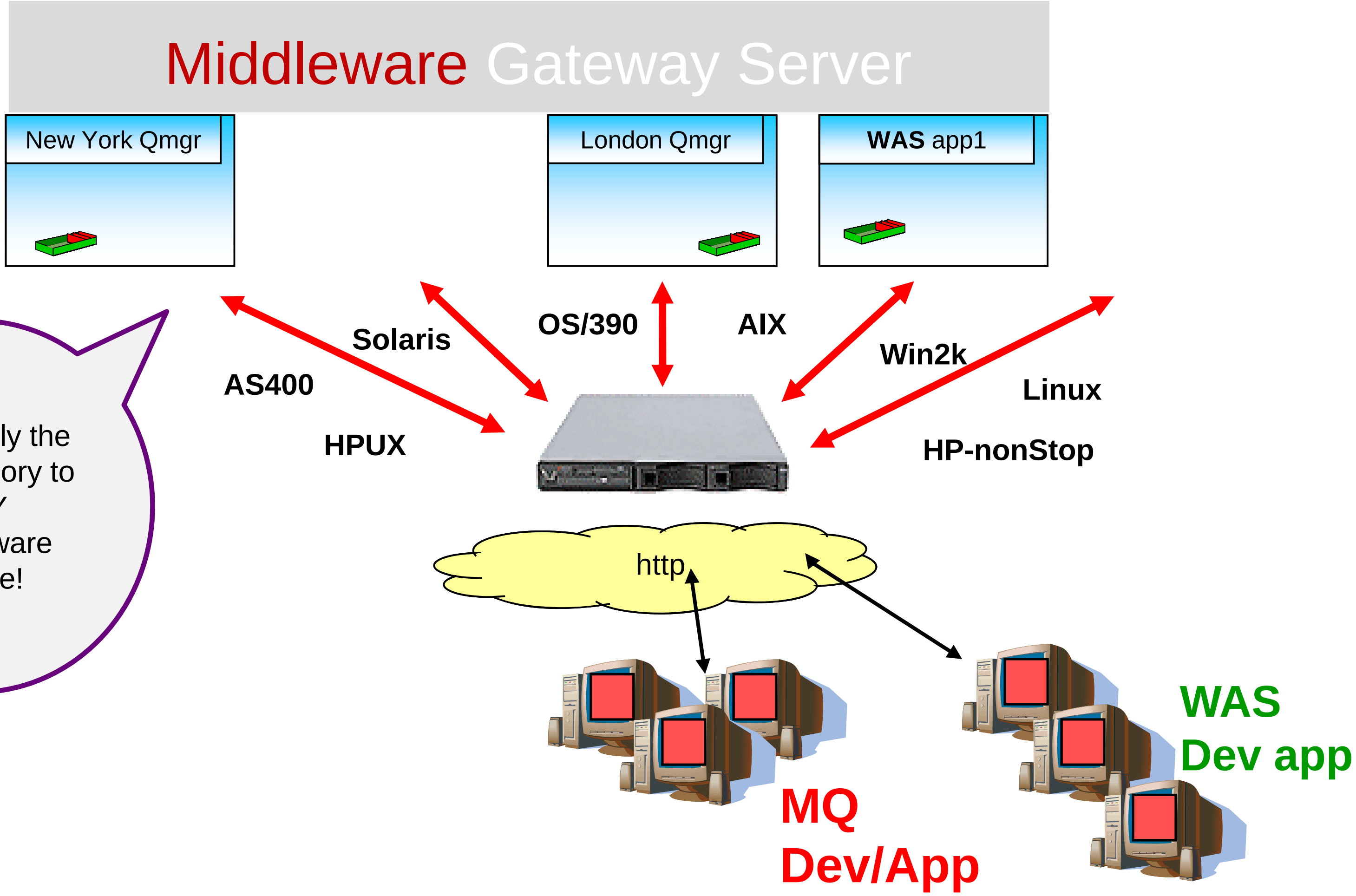
Example : The InfraRed360 Environment



Now apply the same theory to ANY middleware device!

Example : The InfraRed360 Environment

INFRARED 360
Avada Software



Middleware Gateway Server

New York Qmgr

London Qmgr

WAS app1

Solaris

OS/390

AIX

AS400

Win2k

Linux

HPUX

HP-nonStop

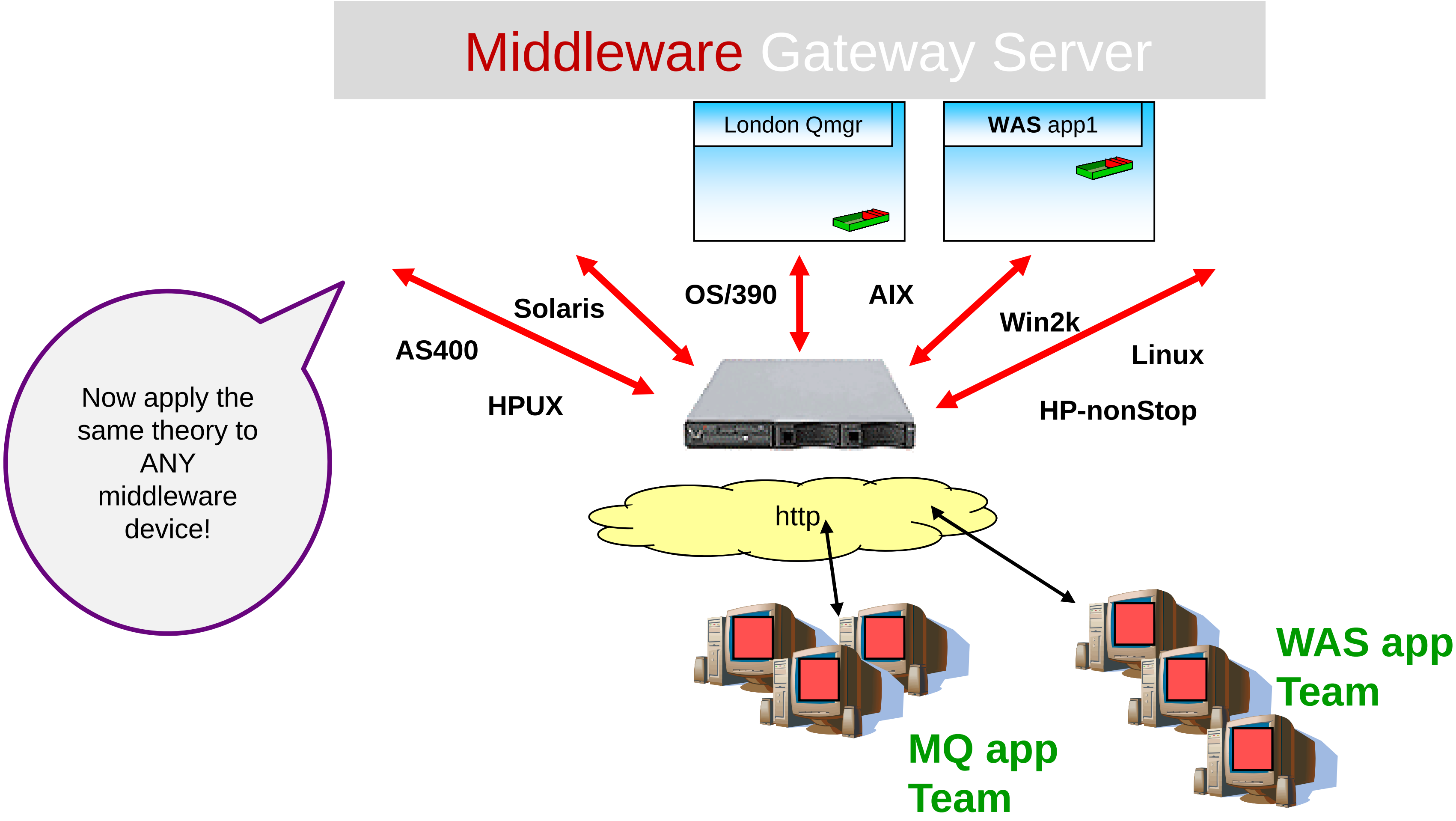
http

MQ
Dev/App

WAS
Dev app

Now apply the same theory to ANY middleware device!

Example : The InfraRed360 Environment



Middleware Gateway Server

London Qmgr

WAS app1

AS400

Solaris

HPUX

OS/390

AIX

Win2k

Linux

HP-nonStop

http

MQ app Team

WAS app Team

Now apply the same theory to ANY middleware device!

Administrative Gateway Access Managers
Layered defense – Middleware style!

- Display of Logical Environments via
  - Biz unit access to WAS
  - Biz unit access to Broker
  - Biz unit access to MQ
  - Biz unit access to Remote App not on IBM stack

Thank You for your time and interest!

Todd Rob Wyatt – IoPT Consulting

@trodrob
http://ioptconsulting.com
http://t-rob.net

Peter D'Agosta – Avada Software

@AvadaSoftware

http://www.avadasoftware.com/