



Nouveautés IBM MQ version 9.1.3

et rappels MQ 9.1.x

Une production : Demey Consulting



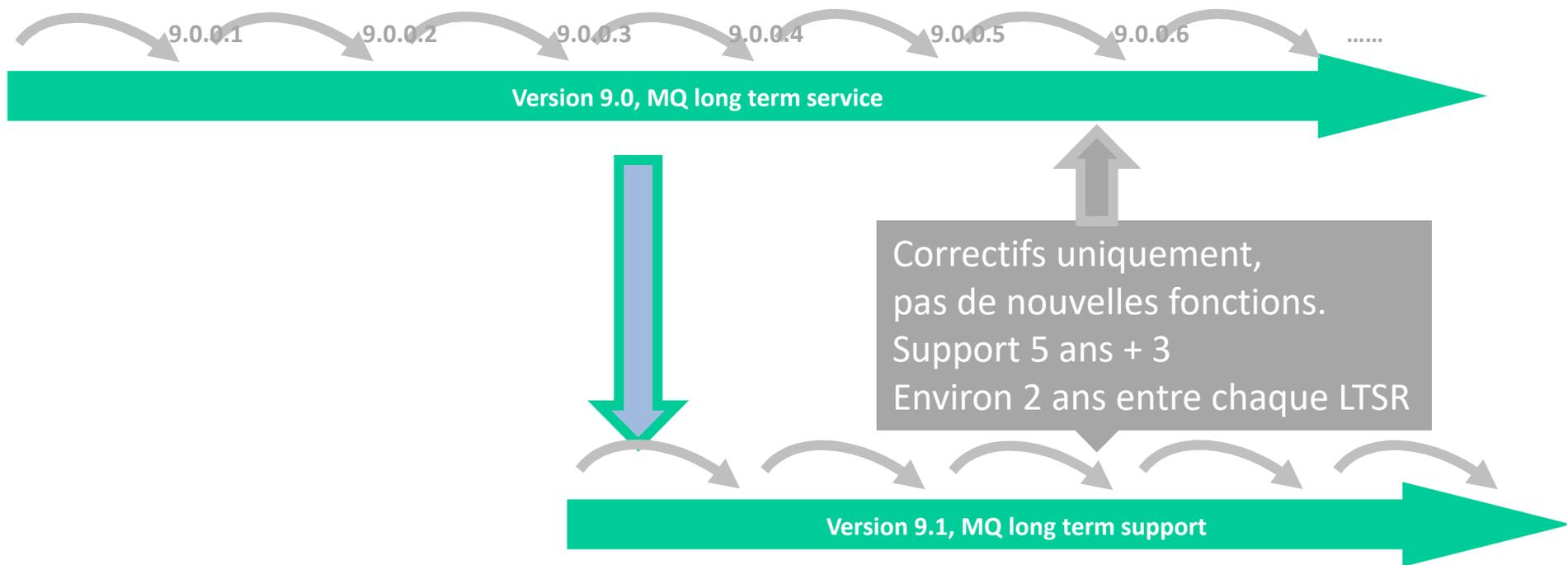
Nouveau modèle de distribution

- LTSR (Long Term Support Release)
 - Propose un socle stable sans ajout de nouvelles fonctions au cours de la vie du produit (comme actuellement si on ne change pas le CMDLEVEL)
 - Les binaires 9.x LTSR seront supportés au minimum 5 ans, avec au delà avec un support étendu (et payant) de 3 ans minimum.
- Continuous Delivery (CD - livraison incrémentale)
 - Permet de disposer rapidement des nouvelles fonctions au fur et à mesure de leur disponibilité (typiquement 3 mises à jour / an)
 - Support 1 an minimum

Technote : IBM MQ FAQ for Long Term Support and Continuous Delivery releases
<http://www-01.ibm.com/support/docview.wss?uid=swg27047919>

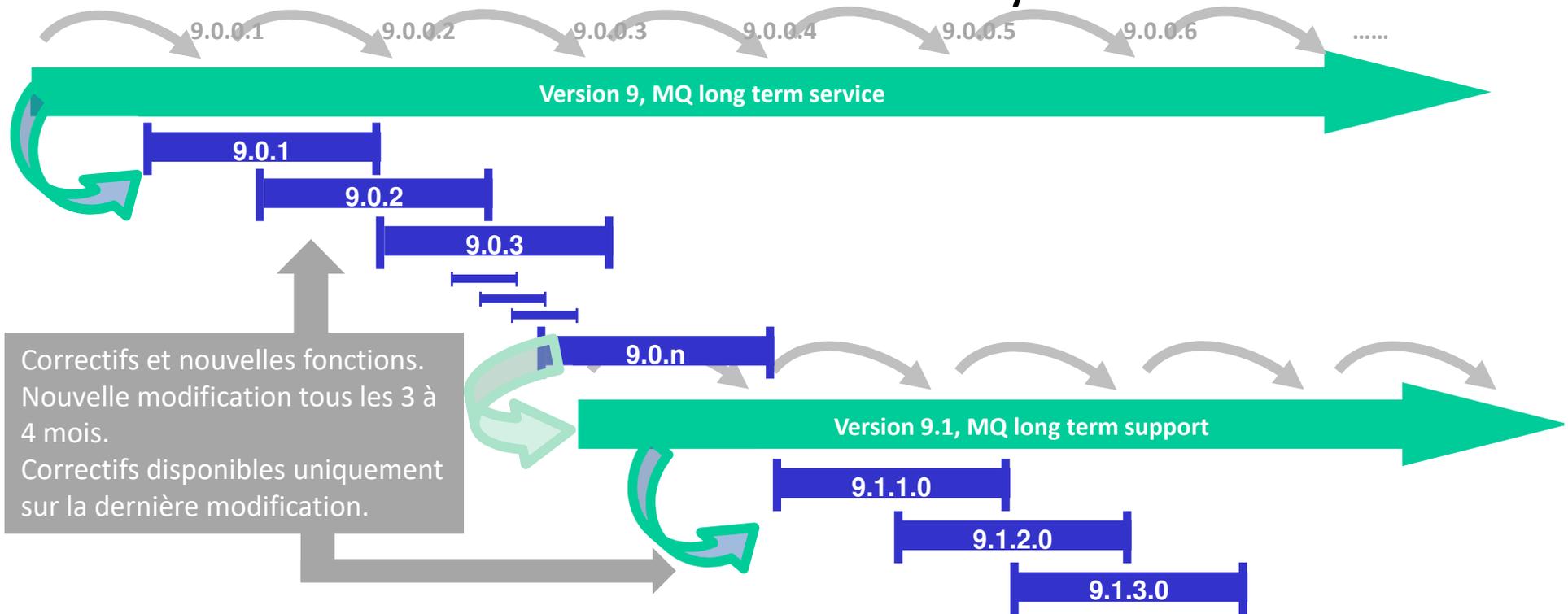


LTSR (Long Term Support Release)





Continuous Delivery





IBM MQ Version 9.1 LTS (06/2018)

- MQ 9.1 = MQ 9.0 + somme(9.0.1:9.0.5) + bonus
- TLS 1.0 déprécié par défaut
 - ré activable dans le qm.ini
- Installation de MQ sur UBUNTU avec les package Debian
 - et non plus avec les packages rpm



IBM MQ version 9.1.1 CD (11/2018)

- Changement de la période de support pour les nouvelles versions CD : 1 an minimum
- Nouveau cipherspec : ANY_TLS12
- Liste des cipherspecs autorisés dans le qm.ini :
`AllowedCipherSpecs=TLS_RSA_WITH_AES_256_CBC_SHA256,..`
- Idempotent runmqsc
 - Nouveau paramètre `IGNSTATE(YES)` pour les commandes START / STOP (distribué).



IBM MQ version 9.1.2 CD (02/2019)

- Uniform Cluster
- CCDT Format JSON
- Support du nom d'application (APPLNAME)
- Connection pools pour REST Messaging
- .NET core support pour environnements Linux
- Améliorations MQ MFT REST API
- Améliorations des Salesforce & Blockchain bridges

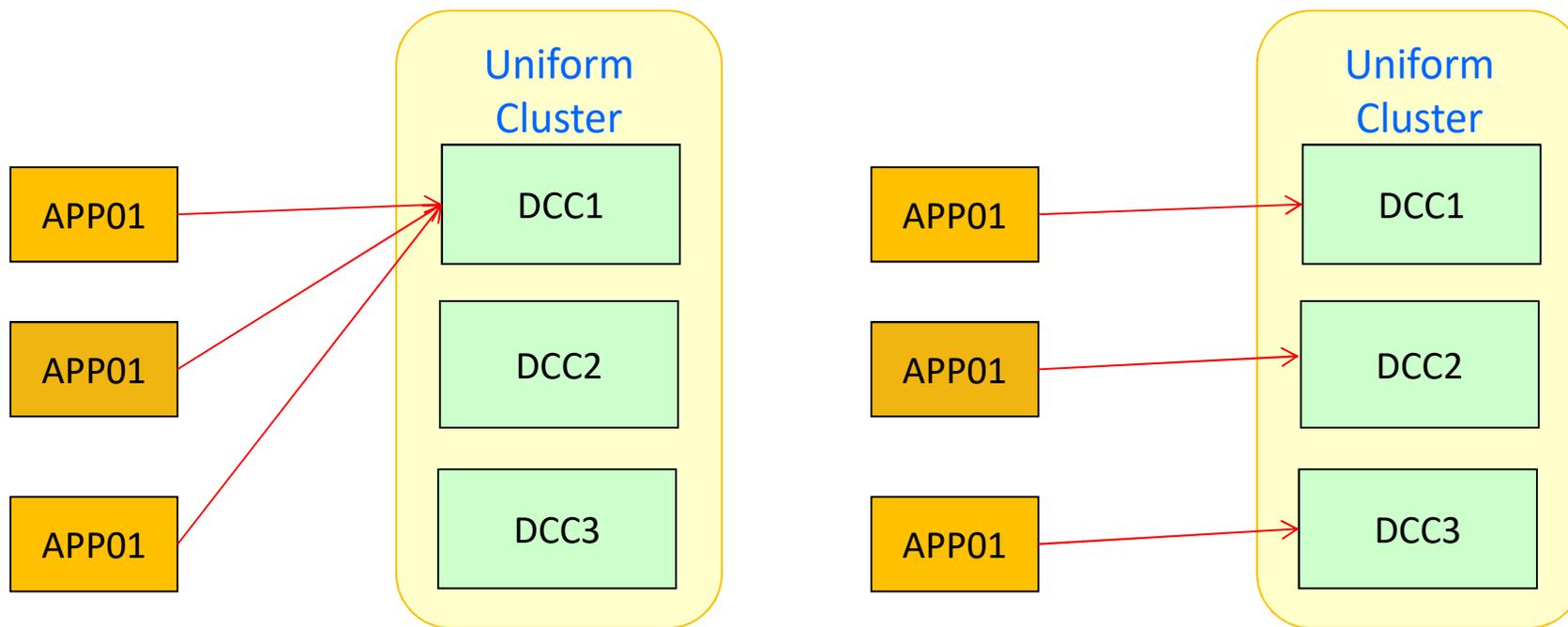


Uniform Cluster

- Extension du cluster MQ
- Permet à des clients MQ de bénéficier des fonctions du cluster MQ :
 - Load balancing dynamique vers plusieurs QM
 - En fonction du nom de l'application
 - Ou du nom du binaire
 - Fail over automatique
- Brique supplémentaire dans les solutions HA



Uniform Cluster





IBM MQ Version 9.1.3

Annonce : 07/2019



API REST Admin

- Nouveau type de commande "runCommandJSON" qui permet d'utiliser le format JSON
- Exemple :

plain text MQSC	runCommandJSON
<pre>{ "type": "runCommand", "parameters": { "command": "DEFINE CHANNEL(ADMIN) CHLTYPE(SVRCONN)" } }</pre>	<pre>{ "type": "runCommandJSON", "command": "define", "qualifier": "channel", "name": "ADMIN", "parameters": { "chltype": "svrconn" } }</pre>

- La réponse est également retournée en JSON.



API REST Messaging

- Liste des messages dans une file
 - Avec sélecteurs
 - Headers uniquement

</messaging/qmgr/{qmgrName}/queue/{queueName}/messagelist>

- Lecture (browse) du message suivant dans une file

</messaging/qmgr/{qmgrName}/queue/{queueName}/message>



Encapsulation de MQ Console dans une IFrame

- Non activé par défaut
- Les pages autorisées à encapsuler la console doivent être déclarées via le paramètre `mqConsoleFrameAncestors`
- Pour autoriser :
 - `http://*.dc-labs.fr/viewMQ`
 - `http://mq1414:9445/consoles`

`setmqweb properties -k mqConsoleFrameAncestors -v`

`http://*.dc-labs.fr/viewMQ, http://mq1414:9445/consoles`

- Possibilité de construire des pages incluant une série de sessions MQ Console
 - Vue globale de l'environnement MQ



Uniform cluster

- Le support du "client re-balancing" est maintenant disponible également pour les clients JMS.
- Commande MQSC : `DISPLAY APSTATUS`
 - et PCF : `MQCMD_INQUIRE_APPL_STATUS`
- Permet de connaître
 - l'état des applications connectées à un Uniform Cluster
 - si les différentes instances sont uniformément réparties
- Exemple :

```
DIS APSTATUS(DC*) type(APPL)
AMQ8932I: Display application status details.
  APPLNAME(DC_APP)      CLUSTER(DCC)
  COUNT(4)              MOVCOUNT(4)
  BALANCED(NO)
```



EphemeralPrefix

- Pour les environnements Container
 - Red Hat OpenShift
- Permet de placer les "domain sockets" (/var/mqm/sockets/<QM>) ailleurs que dans un file system monté.
- Lorsque cet attribut est renseigné, les domain sockets seront écrits en /<EphemeralPrefix>/sockets/<QM>.
- Spécifié :
 - Via une variable d'environnement à la création du QM
 - `MQ_EPHEMERAL_PREFIX = xxx`
 - Via une entrée dans le mqs.ini de l'installation MQ
 - `AllQueueManagers / DefaultEphemeralPrefix= xxx`
- Modifié :
 - Via l'entrée `EphemeralPrefix` du mqs.ini



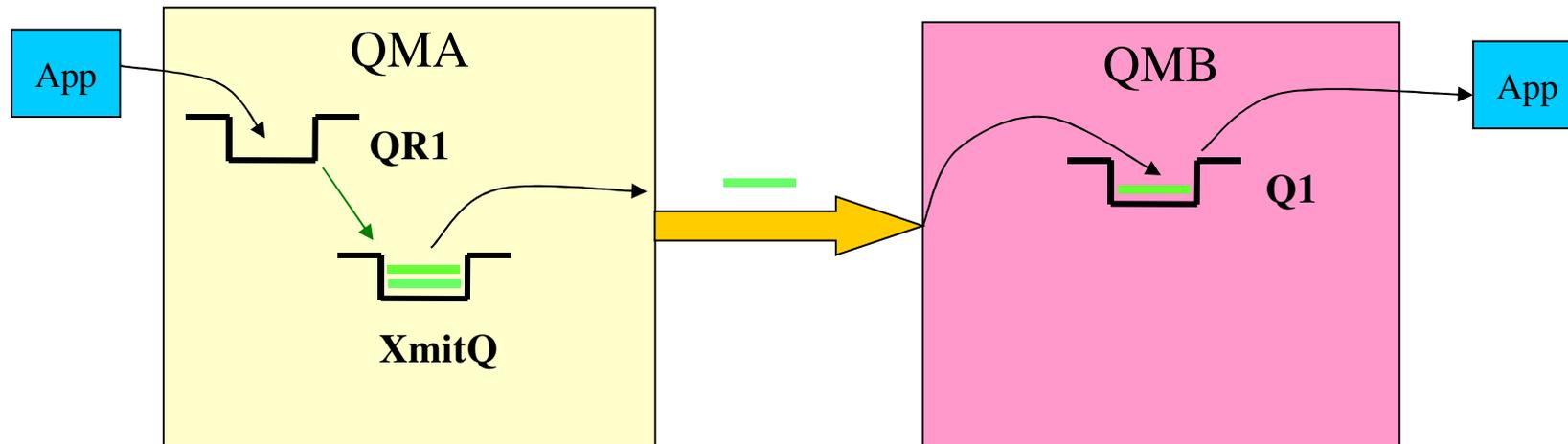
Advanced Message Security

MCA interception sur les canaux DQM

- Nouvelle fonction, disponible uniquement sur z/OS pour l'instant
- Permet d'utiliser la protection AMS des messages dans un environnement où certains Queue Managers n'ont pas la fonction AMS.
- Le fonctionnement est le même que celui qui existait depuis MQ version 8 avec les canaux SVRCONN ("MCA Interception")
- IBM a annoncé son intention de fournir également cette fonction sur les systèmes distribués.



DQM sans SSL/TLS

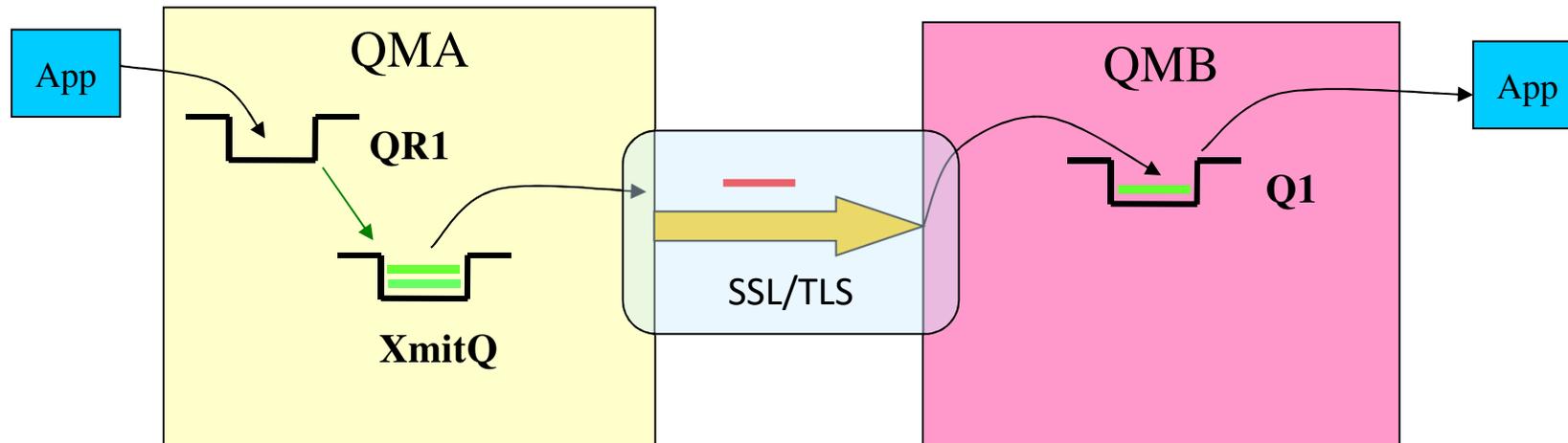


→ Le message n'est pas protégé pendant le transfert, et en clair dans les files

— Message en clair
— Message chiffré



DQM avec SSL/TLS

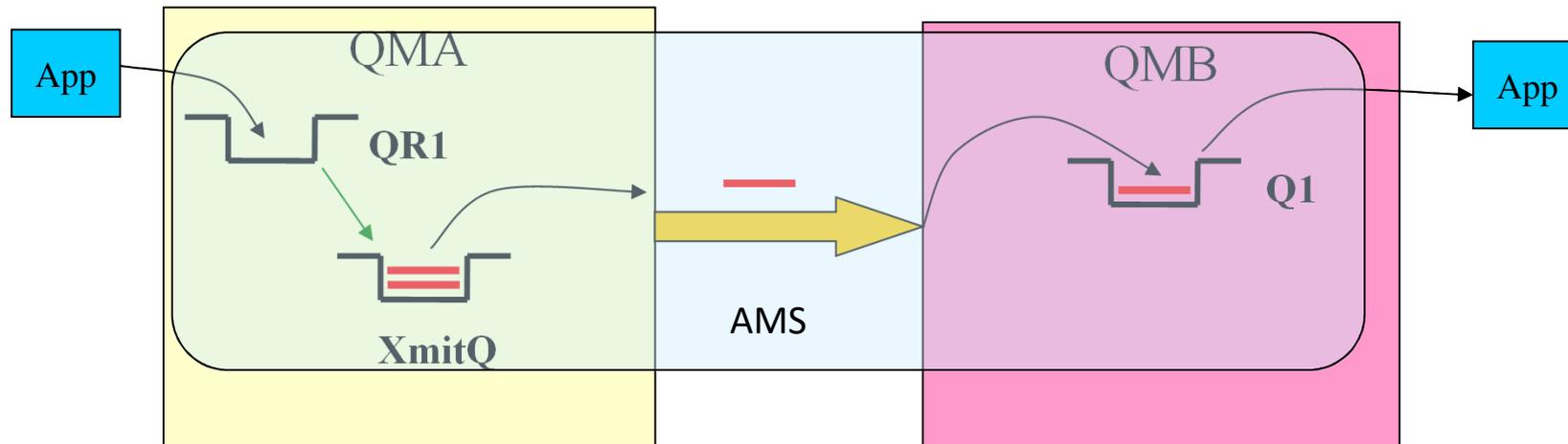


→ Le message est protégé pendant son transfert dans le canal, et en clair dans les files

— Message en clair
— Message chiffré



DQM avec AMS

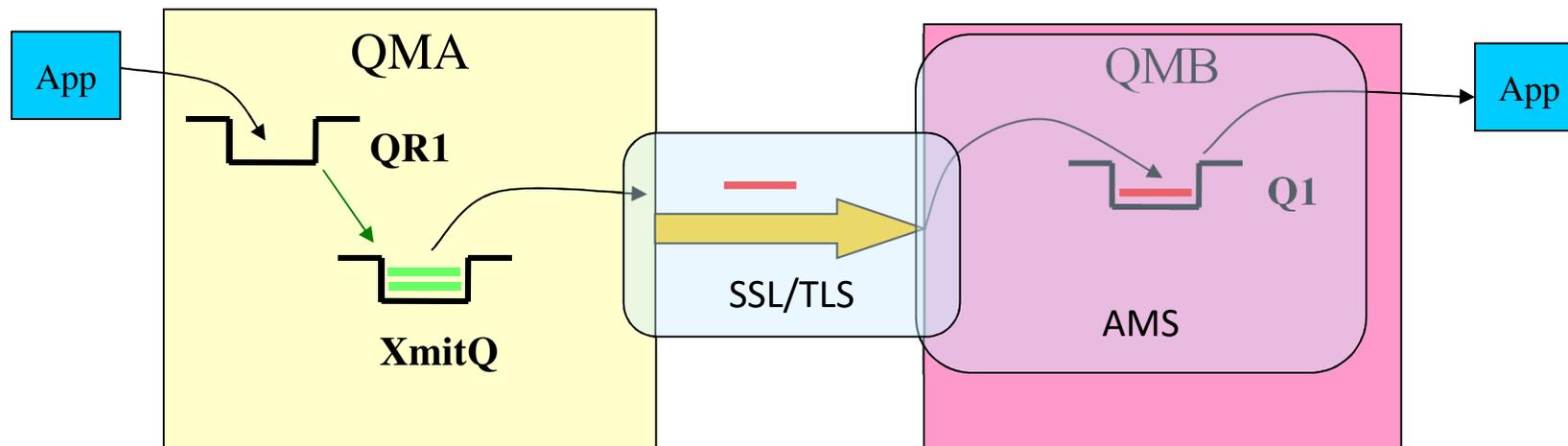


- Le message est protégé pendant son transfert et dans les files
- Contrainte : les deux partenaires doivent utiliser AMS

— Message en clair
— Message chiffré



MCA interception en MQ 913 (z/OS)



→ MCA Interception : Le message est protégé pendant son transfert dans le canal (SSL/TLS) et dans les files du destinataire (AMS).

— Message en clair
— Message chiffré



Autres points

- MQ Explorer (MS0T) mis à jour avec chaque version CD
- Nouvelles commandes MQ Managed File Transfer :
 - `fteClearMonitorHistory`
 - `immediateShutdownTimeout`
 - Support du FTP Protocol Bridge sur IBM i
- Nouvelles notifications pour MQ Appliance
 - Si changement d'état HA, DR ou HA Group
- Conversion de données entre les CCSID 37 et 500
- Support de .NET Core applications sur MacOS



Questions ?