

MQ & MQ/MFT

How They Work in Tandem

Presentation Contents

- **Brief Introduction to MFT**
- **MFT Architecture**
- **MFT Queue Managers**
- **MFT Queue Manager Communications**
- **MFT Client Connectivity**
- **MFT Queue Usage**
- **MFT & MQ Topic Usage**
- **MFT & MQ Security**
- **Summary**

MQ and MQ/MFT

Brief Introduction to MFT

MFT Introduction

■ A “Better” FTP solution

- ▶ File Transport over a reliable network (MQ) instead of an unreliable (TCP) network
- ▶ Asynchronous processing over MQ vs. synchronous processing over FTP
- ▶ Increased endpoint selection
 - IBM MQ Queue Managers
 - FTP/FTPS/SFTP
 - Connect:Direct

■ Built on top of IBM MQ

- ▶ From MQ’s perspective, just another application
- ▶ Inherits all of MQ’s strengths (i.e. Assured delivery)

■ Delivered as part of the MQ software distribution

- ▶ Installs as a component of MQ (distributed)
- ▶ z/OS installation as an independent component
- ▶ Licensing separate from “Base” MQ”
- ▶ License included in MQ “Advanced”

MQ and MQ/MFT

MFT Architecture

MFT Components - 1

■ Queue Managers

- ▶ One or more Queue Managers
- ▶ A Queue Manager may perform more than one “role”

■ Queue Manager Roles

- ▶ Coordination Queue Manager
- ▶ Agent Queue Manager(s)
- ▶ Command Queue Manager(s)
- ▶ Logger Queue Manager(s)

■ Transfer Agent(s)

- ▶ **Java** Program
- ▶ Connects to an Agent Queue Manager
- ▶ Handle File Transfers

■ Logger(s)

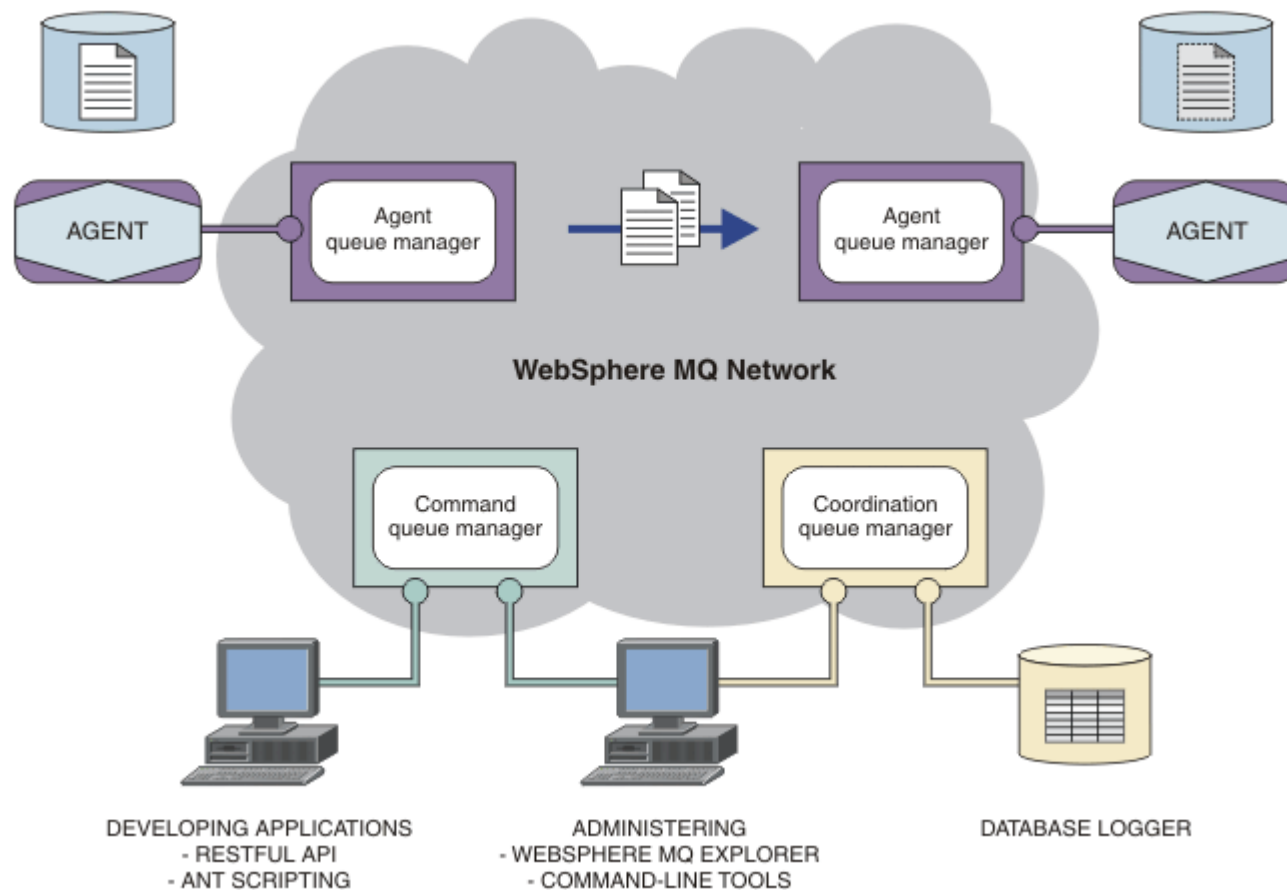
- ▶ **Java** Program
- ▶ Connects to a Logger Queue Manager
- ▶ Log system activity

MFT Components - 2

■ **Command Server(s)**

- ▶ OS Level executable programs
- ▶ Commands connect to a Command Queue Manager
- ▶ Execute control commands delivered to other components

MFT File Transfer Architecture



MQ and MQ/MFT

MFT Queue Managers

MFT Queue Manager Roles

- **Queue Manager Roles**
 - ▶ **Coordination**
 - Hosts MFT Topic
 - ▶ **Agent**
 - Provides MQ Services to the Java Agent
 - Hosts Agent queues
 - Publishes Agent log messages
 - ▶ **Command**
 - Provides MQ Services for the Command libraries
 - ▶ **Logger**
 - Provides MQ Services to the Java Logger
 - Handles Logger subscription to Pub/Sub Topic
- **An individual Queue Manager can perform any or all roles**
- **Queue Managers simply provide standard MQ services**

MFT Queue Manager hosted Objects - 1

■ Agent Queue Manager

- ▶ Each individual Agent has a number of “operational” queues
 - SYSTEM.FTE.COMMAND.*agentName*
 - SYSTEM.FTE.DATA.*agentName*
 - SYSTEM.FTE.EVENT.*agentName*
 - SYSTEM.FTE.REPLY.*agentName*
 - SYSTEM.FTE.STATE.*agentName*
- ▶ Each individual Agent has a number of “security” queues
 - SYSTEM.FTE.AUTHADM1.*agentName*
 - SYSTEM.FTE.AUTHAGT1.*agentName*
 - SYSTEM.FTE.AUTHMON1.*agentName*
 - SYSTEM.FTE.AUTHOPS1.*agentName*
 - SYSTEM.FTE.AUTHSCH1.*agentName*
 - SYSTEM.FTE.AUTHTRN1.*agentName*

MFT Queue Manager hosted Objects - 2

■ Coordination Queue Manager

- ▶ Queue: SYSTEM.FTE
- ▶ Topic: SYSTEM.FTE
- ▶ Namelist: SYSTEM.QPUBSUB.QUEUE.NAMELIST

■ Logger Queue Manager

- ▶ Queue: SYSTEM.FTE.LOG.CMD.*loggerName*
- ▶ Queue: SYSTEM.FTE.LOG.RJCT.*loggerName*

MQ and MQ/MFT

MFT Queue Manager Communications

MFT Queue Manager Communications

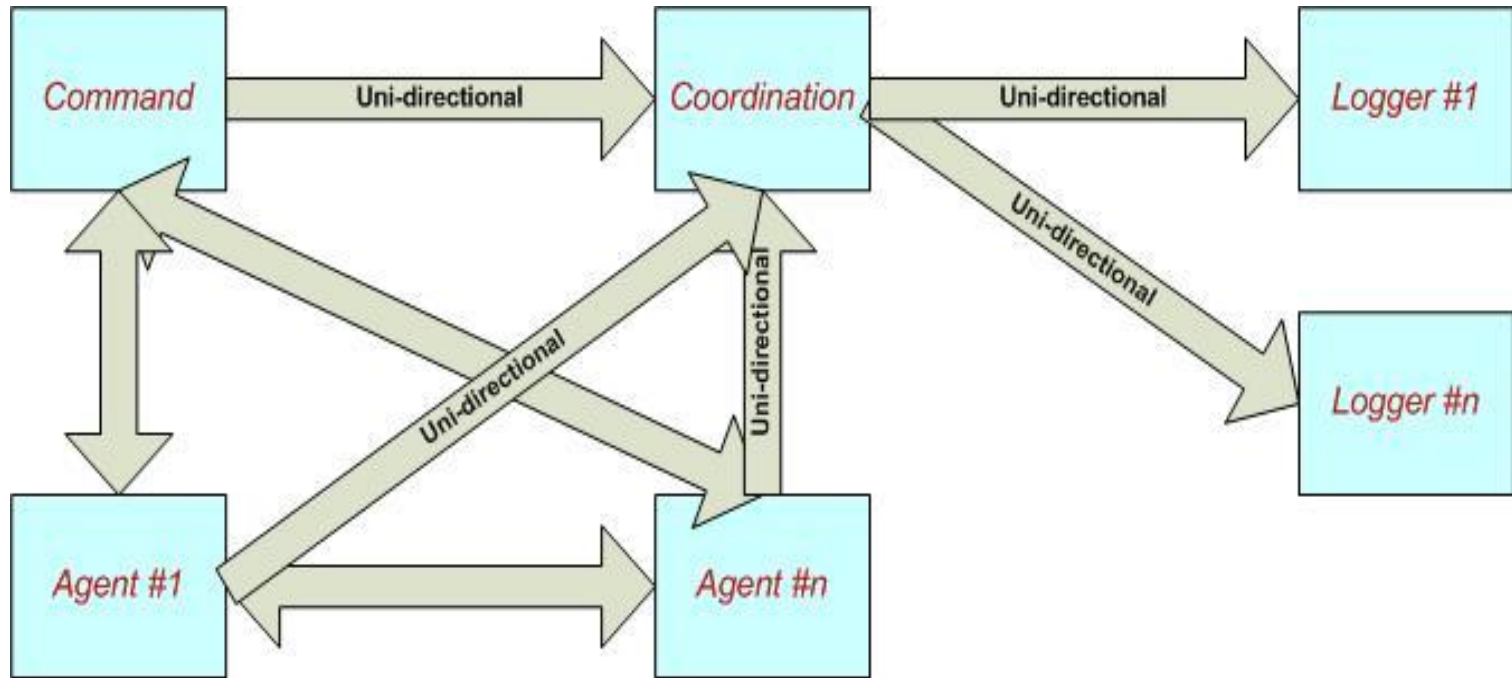
- **Coordination Queue Manager**
 - ▶ Receives commands from Command Queue Manager
 - ▶ Receives subscriptions from Command & Logger Queue Managers
 - ▶ Publishes Topic messages
 - Configuration
 - Logging

- **Agent Queue Manager**
 - ▶ Receives commands from Coordination Queue Manager

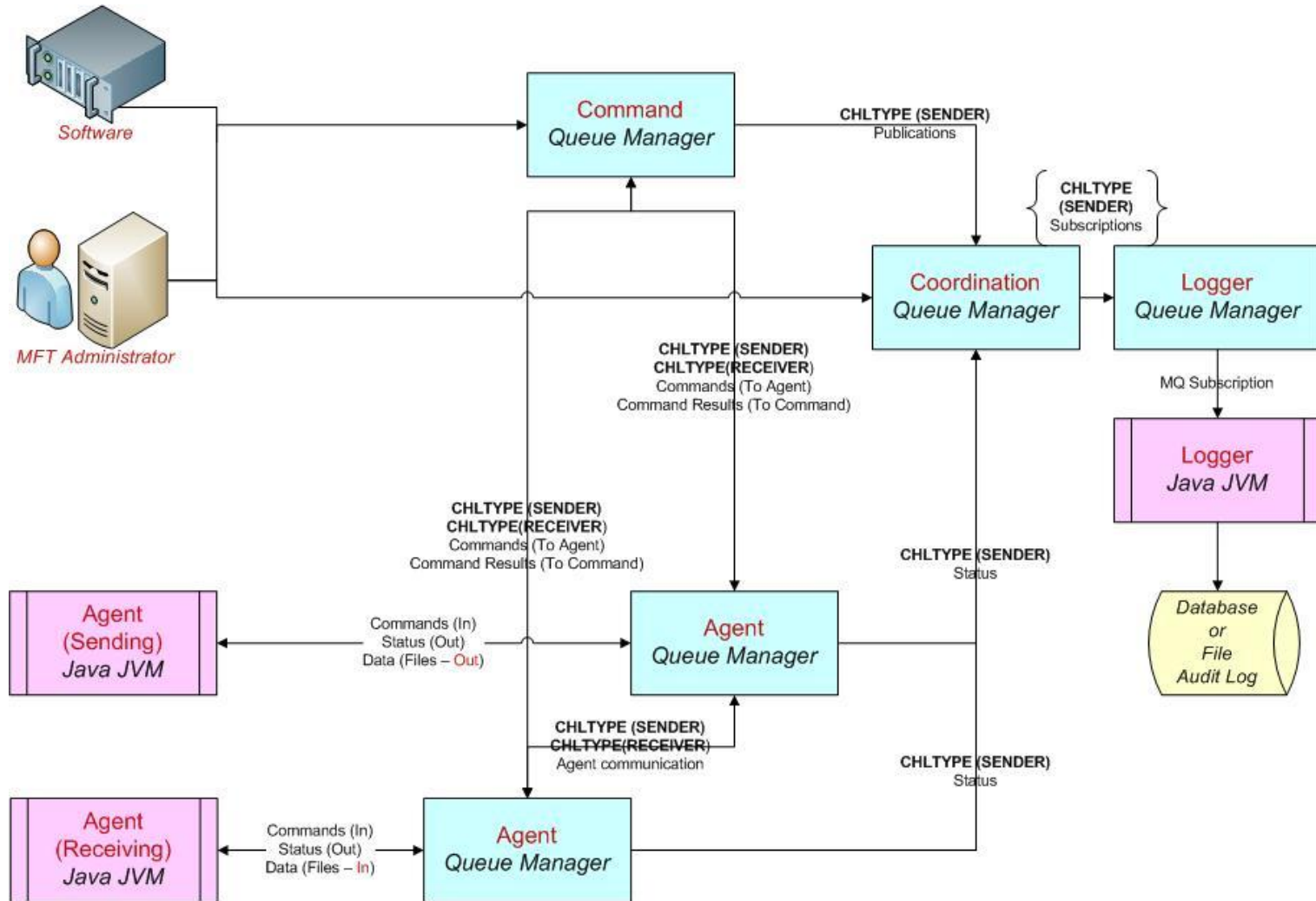
- **Command Queue Manager**
 - ▶ Registers subscription with Coordination Queue Manager
 - ▶ Delivers published “Configuration” messages to Logger
 - ▶ Sends commands to the Agent Queue Managers

- **Logger Queue Manager**
 - ▶ Registers subscription with Coordination Queue Manager
 - ▶ Delivers published “Logging” messages to Logger

MFT Queue Manager Communication Flows



MFT Queue Manager Channels



MFT Communication Options

■ **Point to Point**

- ▶ Sender/Receiver Channel pairs
 - Agent to Agent (bi-directional)
 - Agent to Command (bi-directional)
 - Agent to Coordination (uni-directional)
 - Command to Coordination (uni-directional)
 - Coordination to Logger (uni-directional)
- ▶ Only practical if a small number of Queue Managers

■ **MQ Cluster**

- ▶ Requires Full Repository (and backup) Queue Managers
- ▶ Recommended to be dedicated repository Queue Managers
- ▶ Simplifies Queue Manager connections
- ▶ Scalable solution

Point-to-Point versus Cluster Channels

■ Point to Point Channels

- ▶ 1 Channel * # Logger Queue Managers (Coordination → Logger)
- ▶ 1 Channel * # Command Queue Managers (Command → Coordination)
- ▶ 1 Channel * # Agent Queue Managers (Agent → Coordination)
- ▶ 2 Channels * # Agent Qmgrs * # Command Qmgrs (Agent ↔ Cmd)
- ▶ 2 Channels * # Agent Qmgrs * (# Agent Qmgrs - 1) (Agent ↔ Agent)
- ▶ Each channel requires two definitions; one on each Queue Manager
- ▶ Total MQSC channel definitions:
 - Logger Qmgrs * 2 +
 - Command Qmgrs * 2 +
 - (Agent Qmgrs * 2) +
 - (Agent Qmgrs * 4) * Command Qmgrs +
 - #Agent Qmgrs $C_2 = \text{Agent Qmgrs} * (\text{Agent Qmgrs} - 1)$

■ Cluster Channels

- ▶ Two channel definitions per Queue Manager (CLUSDR & CLUSRCVR)

MQ and MQ/MFT

MFT Client Connectivity

Java Component MQ Connections

■ Do Not Use:

- ▶ *MQSERVER* Environment Variable
- ▶ Client Channel Definition Table (CCDT)
- ▶ “mqclient.ini” file
- ▶ “Pre Connect” exits

■ Do Use:

- ▶ MQCONNX MQI call
- ▶ Properties files to store connection information
- ▶ Property files contain:
 - Queue Manager name
 - Queue Manager hostname
 - Queue Manager port
 - Queue Manager channel (SVRCONN)
- ▶ Note: The logger.properties file only contains the Queue Manager name!
 - The Logger must connect to a local Queue Manager (Server Bindings)

MFT Commands – Agent Server - 1

■ Software installer

- ▶ Install “*Managed File Transfer Base*” (UNIX only)
- ▶ Install “*Managed File Transfer Agent*” (MQ and/or Connect:Direct)
- ▶ Install “*Managed File Transfer Service*” (FTP / FTPS / SFTP)
- ▶ Creates “*installation.properties*” file

■ ftsetupcoordination

- ▶ **First step**; required before an Agent can be created
- ▶ Creates “*coordination.properties*” file
- ▶ Specifies location of the Coordination Queue Manager
- ▶ Populates properties file with Coordination Qmgr connection information
 - Queue Manager name
 - Server Connection channel
 - Hostname & Port
- ▶ Generates MQSC commands to define/delete Topic objects

MFT Commands – Agent Server - 2

■ **ftesetupcommand**

- ▶ **Second step**; required before an Agent can be created
- ▶ Creates “*command.properties*” file
- ▶ Specifies location of the Command Queue Manager
- ▶ Populates properties file with Command Qmgr connection information
 - Queue Manager name
 - Server Connection channel
 - Hostname & Port

■ **ftesetupagent**

- ▶ **Third step**; Creates “*agent.properties*” file
- ▶ Specifies location of the Agent Queue Manager
- ▶ Populates properties file with Agent Queue Manager connection information
 - Queue Manager name
 - Server Connection channel
 - Hostname & Port
- ▶ Generates MQSC commands to define/delete Agent Queues

MFT Commands – Agent Server - 3

■ **ftestartagent**

- ▶ The MFT Agent is a Java program
- ▶ A JVM must be launched in order for the Agent to run.
- ▶ The “start” command launches the Agent JVM

■ **ftestopagent.**

- ▶ The “stop” command terminates the Agent JVM

MFT Commands – Logger Server

■ Software installer

- ▶ Install “*Managed File Transfer Base*” (UNIX only)
- ▶ Install “*Managed File Transfer Logger*”
- ▶ Creates “*installation.properties*” file

■ `ftcreatelogger`

- ▶ **First step**; Creates “*logger.properties*” file
- ▶ Specifies the name of the Logger Queue Manager
- ▶ Populates properties file with Logger Queue Manager connection information
 - Queue Manager name (only supports Server bindings)
- ▶ Generates MQSC commands to define/delete Logger Queues
- ▶ Note: There can be multiple Loggers defined!

■ `ftstartlogger`

■ `ftstoplogger`

MFT Commands – Command Server - 1

■ Software installer

- ▶ Install “*Managed File Transfer Base*” (UNIX only)
- ▶ Install “*Managed File Transfer Tools*”
- ▶ Creates “*installation.properties*” file

■ ftesetupcoordination

- ▶ **First step**; required for commands needing “SYSTEM.FTE” topic data
- ▶ Creates “*coordination.properties*” file
- ▶ Specifies location of the Coordination Queue Manager
- ▶ Populates properties file with Coordination Qmgr connection information
 - Queue Manager name
 - Server Connection channel
 - Hostname & Port
- ▶ Generates MQSC commands to define/delete Topic objects

MFT Commands – Command Server - 2

■ ftesetupcommand

- ▶ **Second step;** Creates “*command.properties*” file
- ▶ Specifies location of the Command Queue Manager
- ▶ Populates properties file with Command Queue Manager connection information
 - Queue Manager name
 - Server Connection channel
 - Hostname & Port

MFT Runtime File System - 1

■ mqDataPath/mqft/

▶ config/

○ coordinationQmgrName/

- **command.properties**
- **coordination.properties**
- **coordinationQmgrName.mqsc**
- agents/

• agentName/

- » **agent.properties**
- » **agentNamecreate.mqsc**
- » **agentNamedelete.mqsc**
- » **ProtocolBridgeCredentials.xml**
- » **ProtocolBridgeProperties.xml**
- » **ConnectDirectCredentials.xml**
- » **ConnectDirectNodeProperties.xml**
- » **ConnectDirectProcessDefinitions.xml**
- » **UserSandboxes.xml** (*Filesystem restrictions*)

MFT Runtime File System - 2

- *mqDataPath/mqft/* ... continued
 - *loggers/*
 - *loggerName/*
 - » **logger.properties**
 - » *loggerNamecreate.mqsc*
 - » *loggerNamedelete.mqsc*
 - ▶ *installations/*
 - *installationName/*
 - **installation.properties**
 - ▶ *logs/*
 - *coordinationQmgrName/*
 - *agents/*
 - *agentName/*
 - » **logs**

MQ and MQ/MFT

MFT Queue Usage

Agent Queue Manager Queues

■ Operational Queues (SYSTEM.FTE. ...)

- ▶ **COMMAND.*agentName*** (*Commands to Agent (XML); e.g. Transfer file*)
- ▶ **DATA.*agentName*** (*File data being transferred*)
- ▶ **EVENT.*agentName*** (*Monitor messages: (1) Definition & (2) State*)
- ▶ **REPLY.*agentName*** (*“Reply” status messages from other Agents*)
- ▶ **STATE.*agentName*** (*Agent’s internal log; used for recovery*)

■ Security Queues (SYSTEM.FTE. ...)

- ▶ **AUTHADM1.*agentName*** (*Shut down agent / Enable Trace on agent*)
- ▶ **AUTHAGT1.*agentName*** (*Receive/Send transfer from/to agent*)
- ▶ **AUTHMON1.*agentName*** (*Create/Delete Resource Monitor*)
- ▶ **AUTHOPS1.*agentName*** (*Delete/Schedule Resource Monitor/Transfer*)
- ▶ **AUTHSCH1.*agentName*** (*Schedule a Transfer / Delete own Schedule*)
- ▶ **AUTHTRN1.*agentName*** (*Start a Transfer / Cancel own Transfer*)

Logger Queue Manager Queues

■ Operational Queues (SYSTEM.FTE. ...)

- ▶ LOG.CMD.*loggerName* (*Commands to Logger(XML); e.g. Transfer file*)
- ▶ LOG.RJCT.*loggerName* (*Messages that the logger is unable to process*)

MQ and MQ/MFT

MFT Topic Usage

MFT Pub/Sub Usage

- **One Topic Tree (“SYSTEM.FTE”)**
 - ▶ Topic hosted on the Coordination Queue Manager
- **Topic used to permanently store configuration data**
 - ▶ Agent configuration and status data (for each Agent)
 - SYSTEM.FTE/**Agents**/*agentName*
- **Topic used to publish Agent log data**
 - ▶ SYSTEM.FTE/**Log**/*agentName/scheduled*
 - ▶ SYSTEM.FTE/**Log**/*agentName/transferId*
- **Topic used to store Transfer status information**
 - ▶ SYSTEM.FTE/**Scheduler**/*agentName*
 - ▶ SYSTEM.FTE/**Transfers**/*agentName/transferId*

MFT Topic Tree

■ “SYSTEM.FTE” Topic Structure

- ▶ SYSTEM.FTE/**Agents**/*agentName*
- ▶ SYSTEM.FTE/**Monitors**/*agentName*
- ▶ SYSTEM.FTE/**Scheduler**/*agentName*
- ▶ SYSTEM.FTE/**Templates**/*templateId*
- ▶ SYSTEM.FTE/**Transfers**/*agentName/transferId*
- ▶ SYSTEM.FTE/**Log**/*agentName/Monitors*
- ▶ SYSTEM.FTE/**Log**/*agentName/scheduleId*
- ▶ SYSTEM.FTE/**Log**/*agentName/transferId*

MFT Pub/Sub Notes

■ Agent Server(s)

- ▶ Agents publish status to the “SYSTEM.FTE” topic
- ▶ Servers have client connection information for the Coordination Queue Manager

■ Command Server(s)

- ▶ Commands can subscribe to the “SYSTEM.FTE” topic
 - e.g. **ftelistagents** information supplied from topic
- ▶ Servers have client connection information for the Coordination Queue Manager

■ Logger Server(s)

- ▶ Logger default is to subscribe to the “SYSTEM.FTE/Log/#” topic
- ▶ Loggers may use a custom Subscription object
- ▶ Loggers may connect to a separate “Logger” Queue Manager
 - Additional MQ configuration required
 - Subscription object on Coordination Queue Manager
 - Local Queue on Logger Queue Manager

MQ and MQ/MFT

MFT Security

MFT Security

■ Queue Manager Connection

- ▶ Queue Manager attribute “CONNAUTH” = IDPWOS IDPWLDAP
- ▶ MQMFTCredentials.xml file must contain a “<qmgr>” tag with:
 - Queue Manager name
 - User ID
 - Password
- ▶ File may be “obfuscated” to prevent simple reading of credentials
 - **fteObfuscate** -credentialsFile *fileName*
- ▶ File Security still essential

■ Channel Security

- ▶ Standard MQ Channel Authorization security in place for Clients

■ User Authority Management

- ▶ agent.properties → authorityChecking = true
- ▶ setmqaut -m *qmgr* -n *securityQueue* -g *group* +*access*

MFT User Authority Management Access

User Action	MFT Authority	Queue	MQ Access
Stop Agent	Administration	AUTHADM1	Browse
Create/Delete Resource Monitor	Monitor	AUTHMON1	Browse
Cancel a Transfer	Transfer Operations	AUTHOPS1	Browse
Delete any Resource Monitor	Monitor Operations	AUTHOPS1	Set
Delete any Schedule	Schedule Operations	AUTHOPS1	Put
Create/Delete a Schedule	Schedule	AUTHSCH1	Browse
Start/Cancel Local Transfer	Transfer Source	AUTHTRN1	Browse
Start/Cancel Remote Transfer	Transfer Destination	AUTHTRN1	Put

MFT Security is MQ Security

MQ – SSL/TLS: x.509 Certificate usage & encryption for inbound & outbound TCP/IP connections

MQ – Channel Authorization: Security rules to restrict inbound TCP/IP connections.

MQ – Connection Authentication: Require Password to authenticate User ID (Local & Remote Connection Settings)

MQ - Object Authority Manager (OAM): Access to MQ objects based upon Group membership

MQ - Advanced Message Security (AMS): x.509 based encryption for messages at rest (e.g. in Queues)

Operating System: User ID & Password Authentication, Group memberships

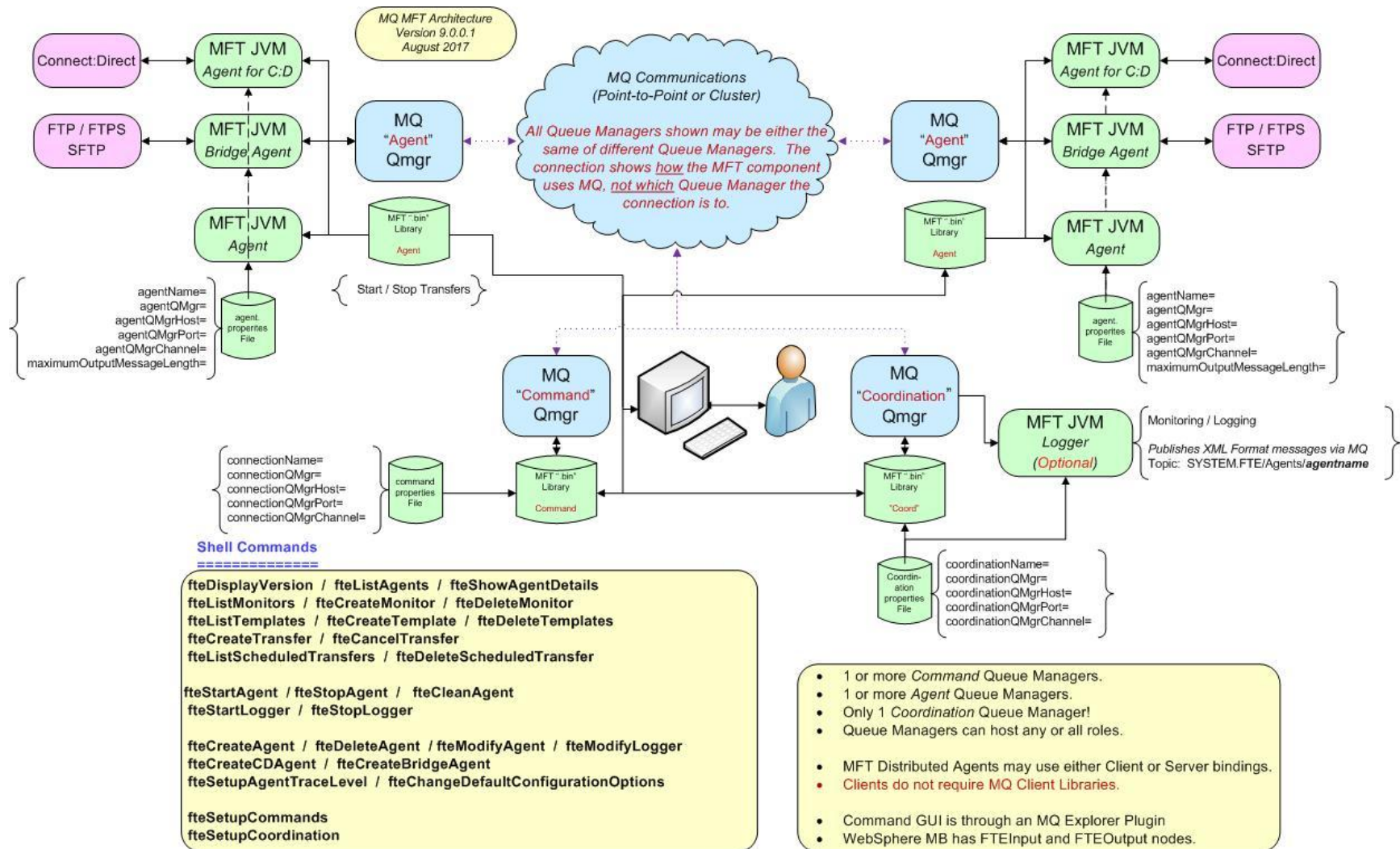
External Security Manager: User ID & Password Authentication, Group memberships

Operating System: File System security (Access to commands and configuration files)

MQ and MQ/MFT

Putting it All Together

MFT Component Architecture



File Transfer Processing - 1

■ Sending Agent starts a File Transfer Process

- ▶ Initiated by Schedule
- ▶ Initiated by Resource Monitor
- ▶ Initiated by Command

■ Receiving Agent participates in the File Transfer Process

- ▶ Initiated by Remote partner (MFT, FTP/FTPS/SFTP, or Connect:Direct)

■ Agents communicate directly with their peer “Agent”

- ▶ MFT
 - SYSTEM.FTE. REPLY.*agentName*
 - Positive acknowledge (“Ack”) of received non-persistent messages
- ▶ FTP/FTPS/SFTP
 - Use appropriate TCP/FTP protocol
- ▶ Connect:Direct
 - Use Connect:Direct TCP protocol

File Transfer Processing - 2

- **Agents publish status directly to the Coordination Queue Manager**
 - ▶ Coordination Queue Manager connection information
 - Connection: *coordination.properties*
 - Topic: `SYSTEM.FTE/Agents/agentName`
 - ▶ If Coordination Queue Manager is unavailable messages store locally
 - `SYSTEM.FTE.EVENT.agentName`
 - MQ routes message to the appropriate Agent Queue Manager
- **File transmitted as one or more MQ messages**
 - ▶ Destination is the `SYSTEM.FTE.DATA.agentName` queue
 - ▶ MQ routes messages to the appropriate Agent Queue Manager
 - ▶ Messages are non-persistent to minimize MQ log impact
 - ▶ Agents inter-communicate to manage the impact to the MQ infrastructure
 - MQ MCA Agents
 - Transmission queues

File Transfer Processing - 3

- **File Transfer notes:**

- ▶ Only the Sending Agent & Queue Manager must be available to start transfer
- ▶ Receiving Agent & Queue Manager needed to complete transfer
- ▶ Coordination, Command, and Logger Queue Managers not required for transfer

Command Processing - 1

- Over 40 *fte...* commands
- Commands distributed across 4 installation components
 - ▶ Agent (18 commands)
 - ▶ Logger (16 commands)
 - ▶ Service (19 commands)
 - ▶ Tools (21 commands)
- Many commands included in multiple components
- Only two types of MFT processes accept commands
 - Agents
 - Loggers
- Commands delivered to “Command” queues
 - SYSTEM.FTE.COMMAND.*agentName*
 - SYSTEM.FTE.LOG.CMD.*loggerName*

Command Processing - 2

- **Commands issued where installed**
 - ▶ Install “*Managed File Transfer Tools*“
 - ▶ Install “*Managed File Transfer Agent*“
 - ▶ Install “*Managed File Transfer Service*“
 - ▶ Install “*Managed File Transfer Logger*“
- **Commands routed to remote Agent or Logger Queue Manager**
 - ▶ Commands specify destination
 - Queue Manager name
 - Agent or Logger name
 - ▶ Command determines remote Command queue name
 - ▶ MQ infrastructure provides the necessary routing
- **Commands, if required, subscribe to “SYSTEM.FTE” Topic**
 - ▶ Servers provide Coordination Queue Manager client connection information
 - ▶ For example:
 - **ftelistagents** subscribes to the “SYSTEM.FTE” topic to obtain list & status

Logger Processing - 1

- **Connect to Logger Queue Manager**
 - ▶ *Connection to local Queue Manager uses Server Bindings*
- **Read commands from Logger Command queue**
 - ▶ Remote commands sent to the Coordination Queue Manager
 - ▶ Commands specify the Logger Queue Manager (if required)
 - ▶ Command messages routed to Logger Queue Manager by MQ (if required)
 - ▶ SYSTEM.FTE.LOG.CMD.*loggerName*
- **Read Subscription messages**
 - ▶ Subscription defined in the *logger.properties* file
 - **wmqfte.source.message.type**
 - automatic subscription (default; logger creates a durable subscription)
 - administrative subscription (Administrator created Subscription object)
 - Queue (Administrator defined queue for a Subscription object)
 - **wmqfte.source.message.name**
 - Subscription or Queue name (if required)

Logger Processing - 2

- Write any invalid messages to the Logger Reject queue
 - ▶ SYSEM.FTE.LOG.RJCT.*loggerName*
 - ▶ Reasons
 - Invalid message type
 - Inconsistent data (e.g. transfer status but no record of transfer)
 - Invalid data (e.g. required fields missing)
 - Data incompatible with target database (e.g. field too large)
- Write valid MFT messages to the Log
 - ▶ File
 - ▶ Database
- Write records to the Logger “log” files
 - ▶ *mqInstallPath*/mqft/logs/*coordinationQmgr*/loggers/*loggerName*/logs (Transfers)
 - ▶ *mqInstallPath*/mqft/logs/*coordinationQmgr*/loggers/*loggerName* (Errors)

MQ and MQ/MFT

Summary

MFT Take Aways

- **MQ has no “special” MFT code**
- **MFT uses standard MQ features**
 - ▶ MQ Channels
 - ▶ Queues
 - ▶ Topics
 - ▶ Queue Security
- **MFT runtime components are all Java or Command Line**
 - ▶ Agents (Java)
 - ▶ Loggers (Java)
 - ▶ Commands (“Shell” commands)
- **Runtime component property files define MQ connections**

Questions & Answers



Presenter

- Glen Brumbaugh
 - Glen.Brumbaugh@TxMQ.com
- Computer Science Background
 - Lecturer in Computer Science, University of California, Berkeley
 - Professorial Lecturer in Information Systems, Golden Gate University, San Francisco
- WebSphere MQ Background (25 years plus)
 - IBM Business Enterprise Solutions Team (BEST)
 - Initial support for MQSeries v1.0
 - Trained and mentored by Hursley MQSeries staff
 - IBM U.S. Messaging Solutions Lead, GTS
 - Platforms Supported
 - MVS aka z/OS
 - UNIX (AIX, Linux, Sun OS, Sun Solaris, HP-UX)
 - Windows
 - iSeries (i5OS)
 - Programming Languages
 - C, COBOL, Java (JNI, WMQ for Java, WMQ for JMS)

Thank
You

The text "Thank You" is rendered in a large, bold, sans-serif font. Each letter is filled with a different portrait of a person. The "T" shows a man in a suit and tie. The "h" shows a woman in a green top. The "a" shows a man with a green face. The "n" shows a woman in a blue top. The "k" shows a man with glasses. The "Y" shows a man in a white lab coat. The "o" shows a man in an orange shirt. The "u" shows a woman in a green top. The letters have a slight drop shadow, giving them a 3D appearance.