U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES

Office of Information Services
Chief Information Officer

CMS Enterprise Messaging Infrastructure

(Including Architecture, Standards, and Implementation Requirements)

December 2003

CMS
CENTERS for MEDICARE & MEDICAID SERVICES

Document Number:
CMS-CIO-STD-INT02

TABLE OF CONTENTS

1. FOREWORD

This document provides an overview of the Centers for Medicare and Medicaid Services (CMS) Enterprise Messaging Infrastructure (EMI), including architecture, standards, implementation, and requirements. An EMI is a key architectural component to enable software systems, make data widely available, reduce redundancy, and provide flexibility to incorporate new technology.

The Office of Information Services' (OIS) Deputy Director for Technology/Chief Technology Officer lead the development of this architecture, including the overall Internet architecture, with support from all elements of OIS as well as input from elements of CMS. It is applicable and serves as the blue print for the implementation of EMI for both in-house and contractor systems that support CMS business operations.

/s/ 12/28/03
Timothy P. Love *Date*
Director, Office of Information Services
Chief Information Officer

/s/ 12/22/03
Wallace K. Fung *Date*
Deputy Director for Technology and
Chief Technology Officer

2. EXECUTIVE SUMMARY

The Centers for Medicare and Medicaid Services (CMS) support a wide variety of applications across disparate platforms. As the agency charged with administering the Medicare, Medicaid, and State Children's Health Insurance Programs, CMS collects, generates and stores financial, health care, and other sensitive information. Most of this information relates to the health care provided to the nation's Medicare and Medicaid beneficiaries, and as such, has access restrictions required under legislative and regulatory directives. In order to provide enhanced services to the end user population by leveraging new and existing technology, CMS has engineered a set of common infrastructure standards that are flexible and robust enough to support the current requirements as well as all future requirements.

This document will outline the components of the messaging infrastructure and provide an architectural blueprint for the design, development, deployment, and support of the messaging infrastructure.

3. BACKGROUND

It is one of CMS's strategic goals to streamline software systems, make data widely available, reduce redundancy, and incorporate new technology. In striving to achieve this goal, CMS has initiated a project to engineer an enterprise infrastructure capable of providing common services to each application and the business unit they support. The resulting 3-tiered architecture includes:

- Presentation Zone – External/Internal interface for access to CMS function and data.
- Application Zone – Business logic repository
- Data Zone – Business data repository

One component of the architecture is the Enterprise Messaging Infrastructure (EMI). The messaging infrastructure will enable ubiquitous data transport services across the enterprise, as well as to any external CMS clients.

In order to support ubiquitous messaging across the enterprise the EMI must be capable of satisfying three core business requirements:

Web Enabled Applications: Future application development will leverage web-enabling technology to satisfy business requirements. The application architecture will support both Intranet and Internet access and will allow for the web enablement of legacy systems. The proposed 3-Tiered architecture is comprised of HTTP servers, Application servers, and Database servers that may be running on disparate hardware and software platforms. The EMI will provide the messaging and routing capabilities to enable the flow of data between applications and data servers.

File Transfer: CMS currently supports file transfer between disparate systems through the use of Connect Direct. Connect Direct is used to satisfy the data transport requirements of both internal and external users. The EMI will provide the same file transfer capabilities in order to enable the migration away from Connect Direct.

Disaster Recovery: A disaster recovery plan is vital to ensure that data integrity and processing capabilities are maintained in the event of an emergency. CMS utilizes multiple data centers to provide the backup and recovery capabilities required by the enterprise. The EMI will satisfy the data transport requirements needed to provide for the synchronization of data between processing facilities.

4. ARCHITECTURE AND STANDARDS

The EMI architecture must consider the following four domains:

- Application Interface
- Queue Manager Topology
- System Management
- Security

The figure below briefly describes these concepts and shows how they overlap.

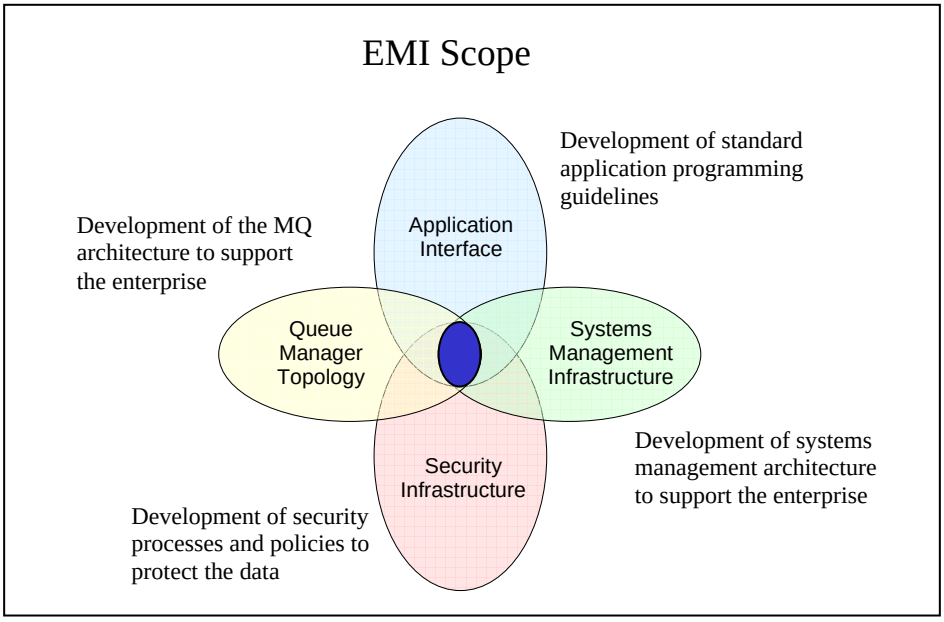


Figure 1 – EMI Scope

4.A. Application Interface

The EMI must be established to provide the necessary message queuing resources to support the requirements of CMS business applications across the enterprise. It also must be flexible enough to support the incremental rollout of integrated applications and the associated infrastructure changes inherent in change. In addition, the infrastructure must be robust enough to support operation of mission critical applications through both planned and unplanned system outages (assuming instances of the applications are still available, or will become available to execute).

Initially achieving and subsequently maintaining this managed infrastructure will require the following, in addition to the requisite capacity planning:

- Enterprise-wide definition and management of naming standards for message queuing objects (for MQ this includes queue managers, queues, channels, and available aliases for these objects);
- Installation and integration of message queuing system management tools to manage the message queuing objects in the CMS enterprise; and
- Integration of the message queuing system management tools with the facilities of EMI to provide automated response and/or alert operations support staff to resolve business application related events.

EMI provides a managed middleware facility for the interchange of business information between business applications. This architecture, depicted in Figure 2, consists of:

- CMS business applications,
- Synchronous/asynchronous message queuing product environment,
- Off-the-shelf and custom-built tools necessary to manage the working messaging environment,
- Off-the-shelf and custom-built tools necessary to secure both the messaging environment and the application data using CMS' own security infrastructure,
- System Management processes and procedures to manage the infrastructure within CMS' existing infrastructure, and
- Development environment to support business application integration and ongoing maintenance of the software supporting the architecture.

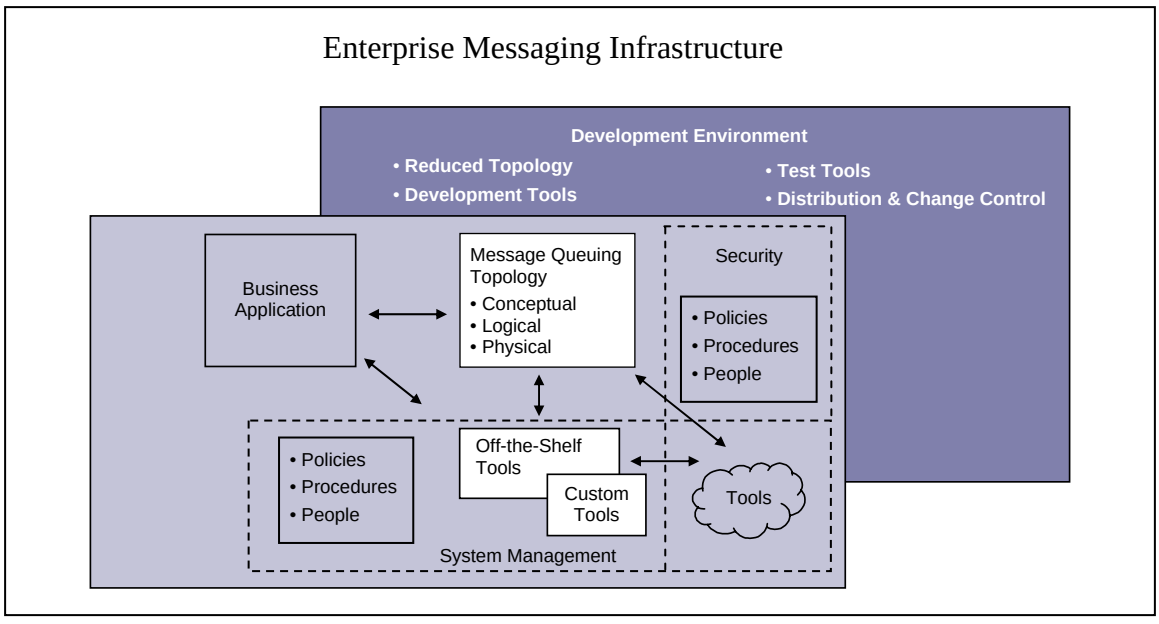


Figure 2 – Enterprise Messaging Infrastructure

4.A.1 Engineering Principles

Listed below are generic engineering principles for developing, deploying, and supporting the new middleware layer:

4.A.1.1 EMI will minimize number of queue managers

Rationale:

- Reduces software licensing costs,
- Simplifies System Management,
- Simplifies Problem Determination,
- Minimizes use of resources,
- Reduces complexity, and
- Reduces software upgrade expense and effort.

Implications:

- Must be balanced against performance requirements,
- Must be balanced against availability requirements, and
- Must resolve organizational ownership issues.

4.A.1.2 EMI will minimize number of physical messaging paths used by applications

Rationale:

- Reduces complexity by minimizing point to point connectivity,
- Establishes common data exchange paths, and
- Provides for efficient use of enterprise network resources.

Implications:

- Network bandwidth and infrastructure must be sufficient to support application requirements and
- Sufficient paths must exist to support required response time and throughput.

4.A.1.3 EMI will segregate interactive and batch messaging paths

Rationale:

- Interactive and batch data flows have substantially different and contradictory response time and throughput requirements.

Implications:

- Requires redundant messaging data paths and
- Increases complexity.

4.A.1.4 EMI will restrict high availability paths to applications with service level agreement

Rationale:

- Redundant paths would be an unnecessary use of resources for applications without service level agreement and
- The most cost effective means since only mission critical data paths need to be duplicated.

Implications:

- Duplicate data paths will increase network complexity and
- Duplicate data paths may increase cost.

4.A.1.5 EMI allows dedicated system management control paths

Rationale:

- System Management Tools must operate even when messaging communications paths are not available and
- Topology must integrate with and support system management.

Implications:

- Messaging and non-messaging paths may be required (for example, SNMP).

4.B. Queue Manager Topology

In providing the EMI architecture the following initial assumptions were made:

MQ Software: All dedicated middleware servers will utilize MQ Server code. The server code provides an application with the ability to continue processing in the event of a network outage. The MQ Client will be utilized by database and application servers located within the same facility as the messaging servers. Database and Application servers using MQ client must include robust error handling to account for the possibility of no connection to the MQ server.

- Data: All data utilizing the EMI will be persistent. While persistence incurs the overhead of logging, data will never be lost. Non-persistent messages will be considered upon review of individual business requirements.

- MQ Configuration: Each MQ server will run in a non-clustered mode. While clustering technology helps to reduce system administration overhead in a highly complex environment, as well as provide enhanced availability and performance, there are some issues that need to be addressed concerning network addressing. The initial rollout of MQ will be simplified at CMS by using non-clustered servers. Subsequent rollouts of MQ may utilize clustered servers as more experience is gained with the product.

- Middleware Architecture: The EMI architecture will utilize both the many-to-many and the hub and spoke processing models.

Each assumption is subject to change based upon the business requirements of the internal and external stakeholders.

4.B.1 Internal Architecture

The internal architecture provides the capability to transfer data between disparate servers within the CMS enterprise. This architecture will provide the functionality to satisfy the file transfer and DB conversations requirements of the enterprise. The many-to-many processing model was chosen for the following reasons:

- Internal connections are static in nature,
- A great deal of functionality beyond providing the assured data delivery is not required,
- The Broker (Hub) does not provide any added value,
- There is one less point of failure,
- There is one less hop for data transfer, and
- Maintenance is simple.

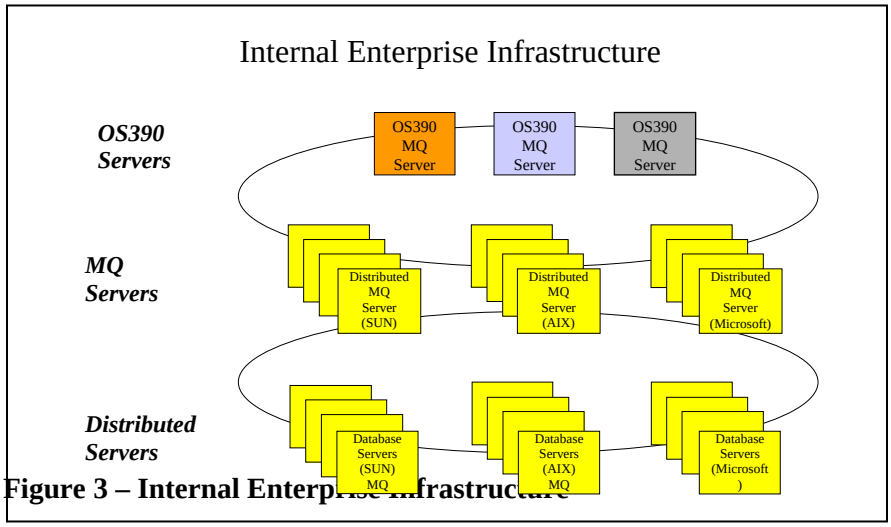


Figure 3 – Internal Enterprise Infrastructure

4.B.2 External Architecture

The external architecture supports the transfer of data between CMS and external entities. This architecture must provide greater flexibility and enhanced functionality in order to satisfy the many requirements of external users.

The hub and spoke processing model was selected for the following reasons:

- Limits connections to external parties,
- Supports data routing at the message broker,
- Supports data transformation at the message broker, and
- Support message segmentation at the message broker.

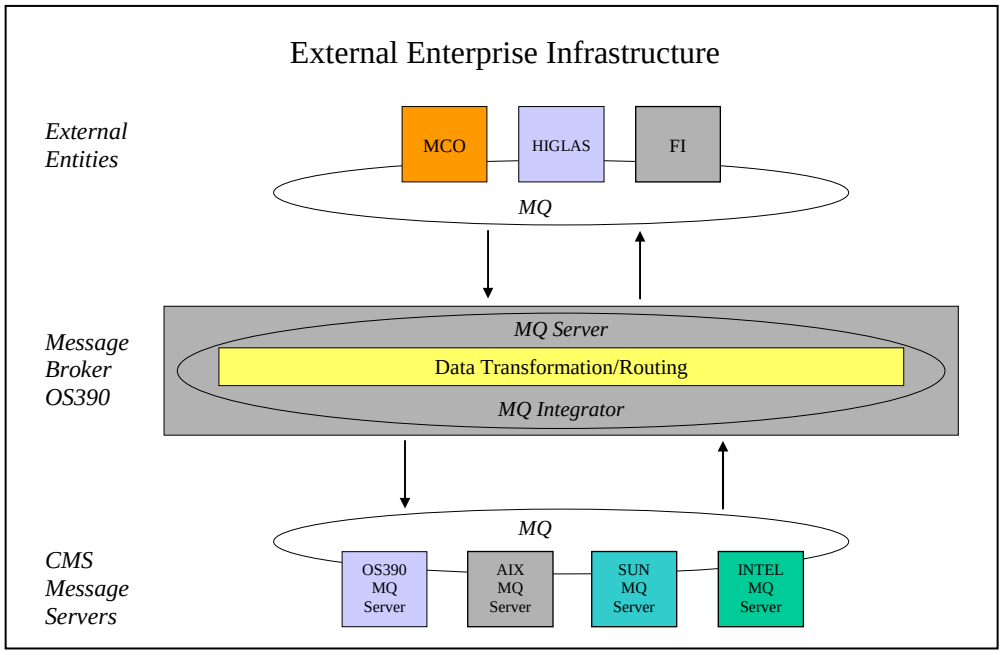


Figure 4 – External Enterprise Infrastructure

4.B.3 Internet/Intranet

The Internet/Intranet architecture represents the future of CMS application development. Within this architecture MQ represents one of the new components that will be used to connect the application servers to the database servers running on disparate platforms. The hub and spoke processing model should be deployed in order to provide the following benefits:

- Limits the number of channel connections between the application servers and the database servers,
- Supports data routing at the message broker,
- Supports data transformation at the message broker, and
- Supports message segmentation at the message broker.

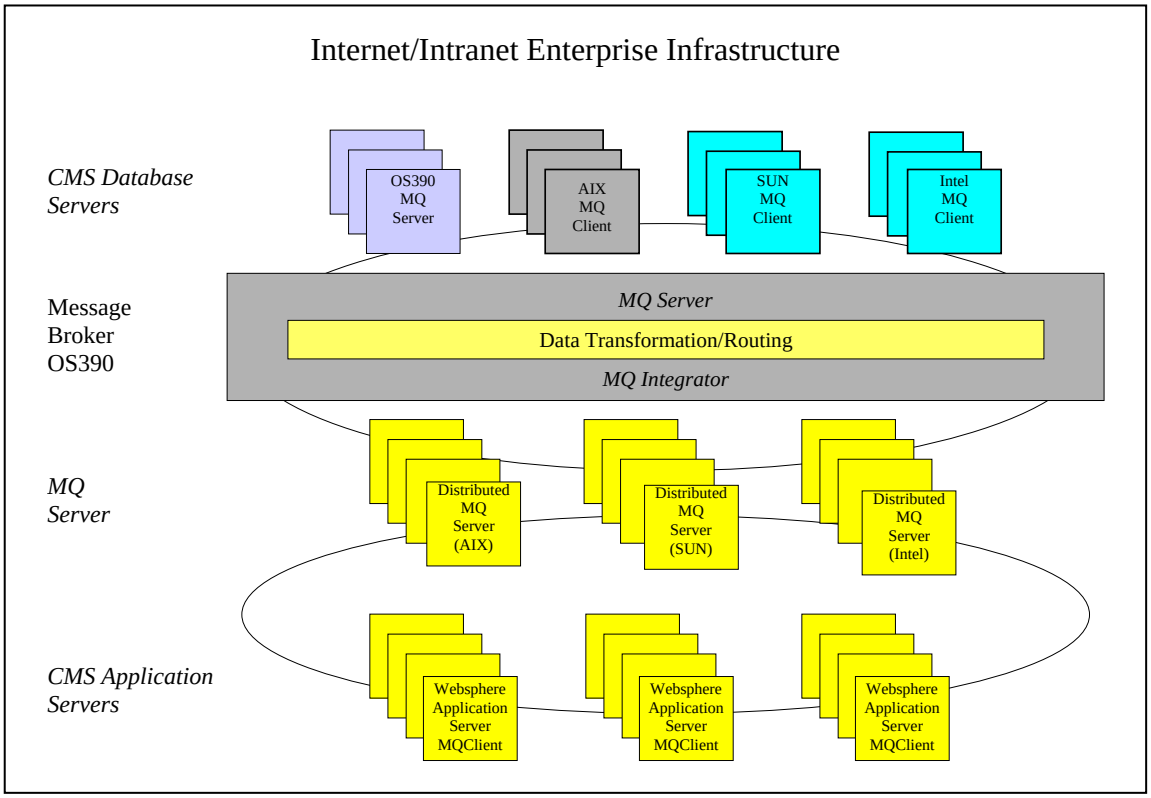


Figure 5 – Internet/Intra Enterprise Infrastructure

4.C. Systems Management Architecture

A comprehensive Systems Management Architecture capable of spanning geographic boundaries and disparate computing platforms will ensure the highest possible availability of the EMI. Effective systems management architecture will provide a central point of control, which enables the proactive management of local and remote messaging resources contained within the EMI. The centralized architecture is depicted in Figure 6.

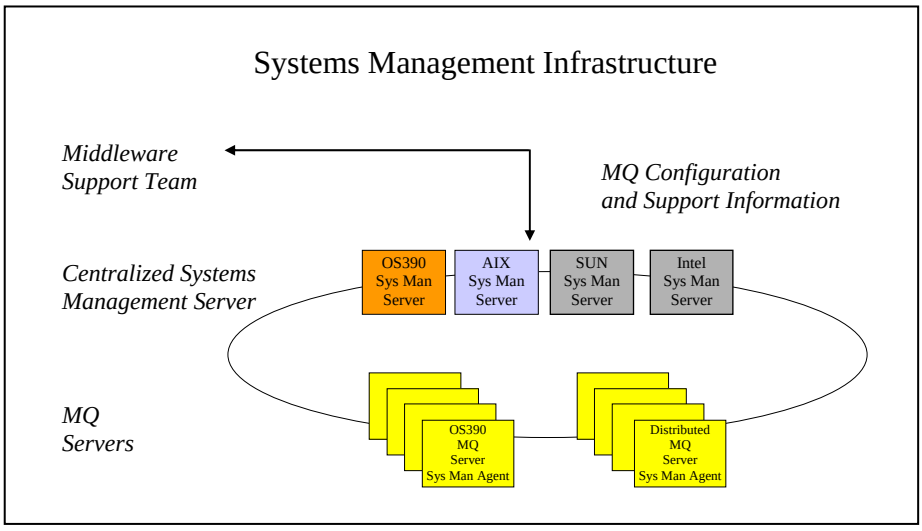


Figure 6 – Systems Management Infrastructure

Listed below is a sample of some of the situations to monitor for:

- Queue Depths
- Queue Service Interval
- Channel Connection
- Channel throughput
- Dead Letter Queue Utilization

The systems management architecture must also allow for the administration and configuration of MQ. There are five basic types of MQ objects that can be acted upon by four general commands. These objects and commands are shown in the following table:

Object	Actions
▪ Queues ▪ Queue Managers ▪ Channels ▪ Processes ▪ Page Set	▪ Alter – changes an existing object definition ▪ Define – defines the attributes of an object and creates the object ▪ Display – displays the attributes of an object ▪ Delete – deletes an object definition

4.C.1 Systems Management Principles

The following are the principles for the systems management architecture.

4.C.1.1 Message Queuing will be centrally managed from both the administrative and operational aspects

Rationale:

- Ensures a standardized approach and provides for enforcement of the standards;
- Facilitates tracking through a centralized audit trail;
- Supports requirements for change management, problem determination, and global monitoring;
- Minimizes the need for MQ training; and
- Optimizes the efficiency of MQ definition.

Implications:

- Changes that have a broad effect will need to be managed,
- Scheduling of changes will be complex,
- Information will need to be available across control boundaries,
- Well-defined interfaces among processes will be required, and
- Organization responsibilities will need to be developed.

4.C.1.2 MQ will be monitored centrally

Rationale:

- Cost of coverage is minimized,
- 7 by 24 coverage is possible, and
- System access capability can best be managed centrally.

Implications:

- Monitoring tools and data collection must provide information to a central command center and
- MQ training for help desk and operations is required.

4.C.1.3 MQ standards will be established and automatically enforced whenever possible

Rationale:

- Standards reduce complexity for the enterprise,
- Standards improve the consistency of the services delivered,
- Standards improve problem resolution by focusing the implementation approaches, and
- Standards improve application development / maintenance productivity.

Implications:

- Documentation and Training packages must be available.

4.C.1.4 All message queuing problems will have root cause analysis and fix performed

Rationale:

- EMI will become a mission critical component;
- EMI will become pervasive and, therefore, the impact of an unsolved problem could grow exponentially; and
- Service expectations will require root cause resolution.

Implications:

- EMI training is essential,
- There may be additional workload for technical support,
- Clear ownership for each problem will be required,
- New processes need to be defined, and
- Techniques such as First Failure Data Capture must be implemented.

4.C.1.5 EMI Management will be proactive rather than reactive wherever possible

Rationale:

- Promotes highest possible service levels,
- Improves reaction times to situations,
- Allows IT reaction before user involvement in many cases,
- Provides for consistent actions to known situations, and
- Provides an opportunity to circumvent a total breakage situation.

Implications:

- Automatic bypass procedures will need to be developed and maintained,
- Currently there are a limited number of tools available and they typically provide limited services,
- Staff will be required to implement and maintain the tools,
- Centralized error handling (command center) will be required, and
- Component Failure Impact Analysis and recovery actions must be defined.

4.C.1.6 EMI problems centrally managed from a technical Help Desk

Rationale:

- Provides 24 by 7 availability,
- Provides early bypass/fix,
- Provides a common point for early determination of the problem impact,
- Provides a central point for problem management,
- Provides a vehicle for problem coordination across platforms,
- Centralizes the problem reassignment knowledge, and
- Allows for enterprise-wide metrics and trend analysis.

Implications:

- Requires automation of by-pass procedures and problem record logging,
- Requires coordination for on call support at all EMI processing centers, for on site problem resolution.
- Represents an additional workload for the technical help desk, and
- Requires performance measures to be established.

4.C.1.7 Message Queuing management tools should take advantage of existing system management and tracking tools and processes and interface to them (like Tivoli or Unicenter)

Rationale:

- Minimizes additional infrastructure complexity,
- Facilitates consolidation of problem information into a common problem database, and
- Uses CMS established procedures, tools and knowledge base.

Implications:

- May require systems development and maintenance efforts.

4.C.1.8 EMI System Management will be automated with commercially available products wherever possible

Rationale:

- Speed of implementation,
- Cost reduction,
- Consistency,
- Robustness, and
- Reliability.

Implications:

- Must define gaps in automation and then be able to fill them,
- Dependent on vendor release cycle for added functionality,

- Tool investment, and
- Increased skill requirements.

4.C.1.9 System Management Tools must be able monitor the health and well being of the EMI infrastructure.

Rationale:

- Allows Help Desk to be proactive,
- Provides historical data to support improvements in process, and
- Provides information that can be used for capacity planning.

Implications:

- Measuring performance may degrade and peformance
- May require specialized tool development to show message traffic flow and performance statistics.

4.D. Security Architecture

The architecture shall ensure the strictest adherence to existing CMS security policies and procedures. This will increase the security of business applications and processes that use the EMI infrastructure. Cooperation between system designers, security designers, and application designers will enable the creation of a viable solution to the business needs.

The security responsibilities and the facilities provided by EMI must be consistent with its role as a reliable message delivery agent for applications and as a user of the underlying transport network. As a managed service, EMI must secure the definition, alteration, and access of its managed resources. The EMI security architecture relies heavily on the EMI Messaging Infrastructure to assure the delivery of messages, in an uncorrupted state, between processes at the end points. Any security mechanisms put in place in EMI must provide for end-to-end security that is shared between the application processes sending and receiving messages and the EMI delivery facilities.

The sections that follow identify those key security requirements that the EMI security architecture addresses.

4.D.1 Identification and Authentication

Identification and Authorization (I&A) services are used to positively identify clients; e.g., end users, applications, server devices, to the enterprise. Although EMI is predominately a set of messaging services, the validation of clients is vital to the success of the EMI security architecture. I&A is the first level of validation that usually occurs when users wish to access the secured resources of the enterprise. All other downstream security measures have some reliance on this initial security check. EMI security interfaces to these end user devices, either directly or indirectly, cascades and strengthens these trust levels by applying additional security on an as needed basis.

4.D.2 Message Integrity

Message integrity is the assurance that the message data has not been modified or otherwise corrupted. EMI will use various encryption-based security services (encryption, digital signature, etc.) to support this need.

4.D.3 Authorization of Application Processes

EMI will provide an option that will limit access and use of a queue to one or more applications specifically granted access and use privileges. This service is usually provided by the existing external security manager and is used by the message based queue manager product.

4.D.4 Authorization of System Processes

EMI will limit access and use of a queue to one or more system processes specifically granted access and use privileges. Processes that issue commands to be executed by EMI infrastructure components must be authorized to do so. Processes that introduce resource definition data to be operated on by EMI infrastructure components must be authorized to do so. This service is usually provided by the existing external security manager and is used by the message based queue manager product.

4.D.5 Security Principles

The following are the principles for the security architecture.

4.D.5.1 Security infrastructure shall support the following fundamental security principles: accountability, authenticity, accessibility, confidentiality, integrity, need to know, privacy, secrecy, and separation of duties.

Rationale:

- The accuracy and timeliness of the business data is vital to the success of the business.

Implications:

- Need to fully understand the security concerns and requirements of CMS business and
- Need to fully understand the possible attacks on the system (these must be identified, categorized, and assigned risk).

4.D.5.2 Security infrastructure must support multiple levels of security

Rationale:

- Level of security shall be based upon the specific needs of the business and the enterprise and
- Security of information tends to vary across the enterprise.

Implications:

- Balance of security, performance, and availability needs to be maintained,
- A minimum level of security must be maintained throughout enterprise, and
- Increased complexity, including:
  - Increased number and types of security products,
  - Increased opportunity for inconsistencies of protection,
  - Increased opportunity for incompatibility of protection,
  - Increased maintenance of protection.

4.D.5.3 EMI security infrastructure shall cooperate with the existing and evolving CMS security infrastructure

Rationale:

- Satisfies CMS legal and compliance requirements and
- Provides consistent security solution for EMI and non-EMI applications that need to communicate.

Implications:

- May require changes to EMI infrastructure as CMS overall infrastructure grows,
- Need to understand the implications to legacy applications that need to maintain a legacy communication and an EMI communication,
- Systems management must support the management and monitoring of such a diverse security infrastructure.

4.D.5.4 Security infrastructure shall be flexible

Rationale:

- Supports a changing and growing business environment

Implications:

- Changes to the security infrastructure must have known and verifiable results,
- Increased complexity including:
  - Incorporation of numerous security mechanisms and
  - Verification of security closure.

4.D.5.5 Security infrastructure shall limit the impact on business applications and processes

Rationale:

- Allows business applications and processes to focus on the business, not on how they will implement security and
- Allows applications to incorporate specific security needs only when directly applicable to the business process.

Implications:

- Complexity and diversity of system security maintenance and monitoring needs to remain as manageable as possible and
- Cooperation and coordination between Systems designers, Security designers, and Business Process designers is required.

4.E. EMI Software Components

The EMI is composed of multiple software components. These components, when integrated, will provide a messaging infrastructure capable of supporting mission critical processing across disparate platforms. The EMI is made up of the following components:

- Websphere MQ
- Websphere MQ Integrator
- Systems Management Tool

4.E.1 Websphere MQ

IBM's Websphere MQ is the core transport of the EMI. MQ provides application programming services that enable application programs to communicate with each other using messages and queues. This form of communication is referred to as commercial messaging. It provides assured, once-only delivery of messages. Using MQ means that you can separate application programs so that the program sending a message can continue processing without having to wait for a reply from the receiver (asynchronous). If the receiver, or the communication channel to it, is temporarily unavailable, the message can be forwarded at a later time. MQ also supports request/reply (synchronous) processing, where the sending application waits for a response from the receiving application.

The programs that comprise a MQ application can be running on different computers, on different operating systems, and at different locations. The applications are written using a common programming interface known as the Message Queue Interface (MQI**)**, so that applications developed on one platform can be transferred to another.

Some of MQ features are listed below:

- Heterogeneous any-to-any connectivity from desktop to mainframe,
- Integrated disparate islands of information,
- Common application programming API,
- Network independent,
- Platform independent (over 35 platforms supported),
- Time independent processing,

- Assured one time delivery of data,
- Message prioritization.

4.E.2 Websphere MQ Integrator

Websphere MQ Integrator is the core of the message/routing hub. MQ Integrator enhances the capabilities of the messaging infrastructure by managing the real-time flow of data based upon rules and policies loaded into the Integrator engine.

MQ Integrator provides the following features:

- Transforms, augments and applies rules to message-based data and routes and distributes it between systems,
- Integrates both existing and new applications with business data using dynamic content and topic-based publish/subscribe functions,
- Displays the application flow through a graphical development environment,
- Allows message formats to be defined through a variety of dictionaries, either those supplied with the product or from a third party, and
- Allows DB2, Oracle, and Sybase updates under transactional control.

4.E.3 Systems Management

In order to support the EMI successfully, a systems management tool must be selected and implemented. MQ, by nature of its architecture, will continue to accept messages from an application even if the network or the destination queue manager/application is no longer functioning. MQ objects must be monitored in order to ensure the continuous flow of data across the enterprise. A systems management tool will provide a middleware team with the ability to monitor and manage EMI resources throughout the enterprise

4.E.4 Standards and Conventions Source Document

The CMS MQ Standards and Conventions are based on the published IBM Standards and Conventions Source Document, *MQ – Standards and Conventions, Version 1.1*, dated October 9, 2000. To review this document, please reference the IBM web site, http://www-3.ibm.com/software/ts/mqseries/txppacs/md01.html.

Any CMS Standards and Conventions that differ from the IBM document are listed and described in the next section.

4.E.5 Modifications to Standards and Conventions

4.E.5.1 Queue Manager

Queue manager names are case-sensitive; use all CAPS.

The format for the Queue Manager name is: XZZDDDDN

X	- Defines environment (d = development, t = test, q = QA, a = acceptance, p = production)
ZZ	- Defines the platform (nt-windows, ax-aix, mv-mvs)
DDDD	- Server name (CONAP01, S0HA, etc.)
N	- Queue manager numeric identifier

Example: DMVS0HA1

4.E.5.2 QUEUES

There may be multiple message queues created for a given application, therefore the queue naming convention will be built around Application and Function. Queue manager names are case-sensitive, use all CAPS. The accepted format is:

The format for the Queue name is: Application.Function.Suffix

Application	- Defines the project (up to 8 chars.)
Function	- Defines the functions being performed (up t o 8 chars)
Suffix	- Clarifies uniqueness when multiple versions of function exists, (ie. V2, test). May consist of multiple (up to 8 character) nodes. The suffix nodes are optional and only used as needed.

Examples: MBD.TESTELIG.TEST
MBD.TESTELIG.QA.V20

4.F. Implementation Requirements

4.F.1 Event Management

The objective of the EMI Event Management Process is to capture and respond to all MQ related events in the CMS environment. These processes will provide a foundation for defining and measuring attainment of service level targets for the EMI infrastructure. EMI Event Management will manage a subset of events related to MQ infrastructure (queues, queue managers, channels, and processes). In some cases, such as message queuing hubs, Event Management will be expanded to include all hardware and software.

When an event occurs the queue manager puts an event message on the appropriate event queue, if defined. The event message contains information about the event and is processed by the systems management software product for MQ.

Queue manager: These events are related to the definitions of resources within queue managers. For example, an application attempts to put a message to a queue that does not exist. These events are related to the definitions of resources within queue managers. For example, if an application attempts to update a resource but the associated user ID is not authorized to perform that operation, a queue manager event is generated.

Performance: These events are notifications that a threshold condition has been reached by a resource. For example, a queue depth limit has been reached. These events are notifications that a threshold condition has been reached by a resource. For example, a queue depth limit has been reached or following an MQGET request a queue has not been serviced within a predefined period of time. Performance events are related to conditions that can affect the performance of applications that use a specified queue. There are two types of performance event:

- Queue depth events, related to the number of messages on a queue and
- Queue service interval events, related to whether messages are processed within a user-specified time interval.

Channel: Channels, as a result of conditions detected during their operation, report these events. For example, when a channel instance is stopped. Channel events are generated:

- By a command to start or stop a channel,
- When a channel instance starts or stops, and
- When a channel receives a conversion error warning when getting a message.

4.F.2 EMI Performance Management

The purpose of this process is to anticipate and manage the performance requirements of the EMI processing environment from both a real-time and historical perspective. The importance of this process is that it provides a means of ensuring that service level agreements are met or exceeded.

This process focuses on the measurement, analysis, monitoring and periodic fine-tuning of components as necessary to meet those requirements.

This process must have the capability of identifying potential constraints, the ability to measure activity at critical points and the analytical ability to determine what action should be taken and when. Support from the Application Development, Operations, and Capacity Planning is essential to design and deployment of performance management metrics.

The goals of performance management are:

- To ensure effective and efficient use of resources while achieving service level commitments and
- To identify resource requirements from performance and/or capacity needs in a timely fashion.

4.F.2.1 EMI Performance Management Data

In order to ensure the performance and availability of the messaging infrastructure a number of statistical categories across all supported platforms must be monitored and recorded for real time and historical analysis including:

- Queue Manager Status – Current status (started/stopped) of each queue manager defined within the enterprise
- Event Information – Current and historical MQ event information
- Channel Performance – Channel performance as recorded by message rate, number of bytes per second, and number of batches per second
- Queue Performance – Gets/Puts per second, Gets/Puts latency, Average Msg Size, Gets/Puts Response times, etc.
- Queue Display – Status of each queue as related by queue depth and number of active processes

In addition to the above information, there are MQ statistics unique to the OS390 environment:

- Buffer Pool Performance - Buffer pools are areas of memory used for messages and queues and are critical to the performance of the queue manager. Monitoring and adjusting the number of pools and buffers in use is vital to the performance of the queue managers.
- Log Manager Performance - In order to support assured delivery, recoverability, and persistence MQ must log all appropriate messages. Log manager performance is monitored through log buffer availability waits and the percentage of log requests satisfied from active versus archive logs.
- Page Set Statistics - Current page set allocation and usage statistics.
- Message Manager Performance - Queue manager message throughput (i.e. MQOPEN/GETS per second).
- Application Statistics - Application and MQ statistics relevant to a specific application including:
  - Application MQI calls,
  - Message throughput by queue,

- Message size reports by application and queue,
- Application and MQ response times,
- API return codes, and
- Application queue locks.

4.F.3 Configuration Management

The configuration of the EMI components will become more critical as this environment grows. Since the middleware components of the EMI will reside on both dedicated and shared servers it will be crucial to plan the configurations so that there is as much commonality between these configurations as possible. This will allow for quicker resolution to problems that will arise as well as allow for the ease of implementing changes to existing configurations. Central management for control of these configurations as well as a central repository to store the configuration information will be maintained.

For the shared server model the only configuration controls that the EMI team will have is specific to the EMI components that are installed on these servers. Since the rest of the shared server's configuration will be outside the control of the EMI team, close attention must be paid to how these servers are configured, i.e., operating system levels, other products installed and hardware. It will also be imperative that a tight coupling be formed between the change management process for these shared servers and the configuration management process so that configuration changes that may impact the availability of EMI are communicated and then documented by the EMI team.

The dedicated EMI server model will have additional configuration considerations beyond just the EMI components. The EMI team will be responsible for creating and maintaining the configurations for the entire dedicated server system.

To support the EMI specific components in both the shared and dedicated server models, a systems management configuration agent will be used to directly configure MQ queue managers from a central point of control. The configuration agent will be installed on every node that has the MQ server product installed. This will allow the middleware team to centrally perform the following function:

- Design a prototype MQ environment configuration for the entire network and implement that design from the systems management hub,
- Re-configure all or part of this environment by modifying the defined prototype,
- Group queue managers so that the same actions can be applied to every queue manager within this group by one single action,
- Group queues, channels and processes so that the same actions can be applied to these resources within this group by one single action,
- Copy or move portions of the MQ environment or delete them,

- Define channels and transmission queues between queue managers,
- Maintain a central configuration database that retains all information about the MQ environment, and
- Detect and resolve any discrepancies between the actual MQ configurations and the information contained within the central configuration database.

Either an application or EMI requirement will bring about changes to the MQ objects. These changes will filter through the EMI change management process and then be directed to the appropriate team.

The purpose of this process is to capture and disseminate to the provider of information technology the technical information regarding the interconnection of the IT resources — and the attributes of each of those resources — of the entire enterprise. The importance of this process is that, by keeping track of what is connected to what (either physically or logically), it ensures current, accurate information so that whenever a change is introduced, the impact of that change can be fully understood and the change can be properly planned.

This process identifies configuration information to be maintained, establishes the information management system, collects the configuration information, populates the database, and makes the information available to other processes. This process recognizes that configuration information can be relatively static or highly dynamic.

The goals of configuration management are:

- To identify, capture, and organize key configuration information,
- To provide accurate information to multiple processes, and
- To maintain the data so that it remains current.

4.F.4 Change Management

The purpose of the Manage Changes process is to introduce changes into the IT environment in such a way as to minimize disruption to that environment. The importance of this process is that disruption of service has serious consequences and, without the proper management and control that it provides, any change has the potential to create chaos.

The execution of the tasks involved in this management and control — whether completed manually or via automation — should be applied to all changes, including software, hardware, configurations, environments, data bases, business applications, etc., for changes large or small.

The tasks associated with the Execution-level parts of the Manage Changes process address many aspects of change activity — including assignments, scheduling, approval, distribution, synchronization, installation, and activation.

The primary focus for the EMI change management process will be to address the notification of change to the EMI support organization as well as the application development organization in a shared server environment.

Goals of Change Management include:

- To introduce changes into the environment with minimal or no disruption to information technology and its users and
- To ensure consideration of appropriate security, contractual constraints and quality controls.

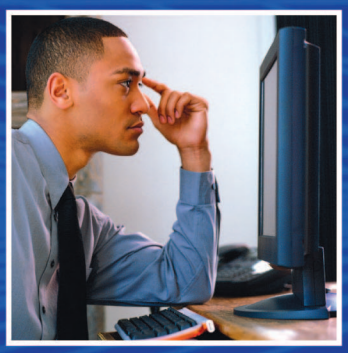
4.F.5 Problem Management

The purpose of this process is to capture information about problems and ensure their resolution. The importance of this process is that it serves to minimize repeat failure and disruption of service.

This process identifies, documents, analyzes, tracks, and resolves all problems within EMI's sphere of operation. It should be noted here that there is no universally accepted or agreed upon definition of what constitutes a "problem." In general, a problem may be defined as any issue raised by customers or providers of IT that is inhibiting their success. For the purpose of the EMI project, a "problem" is identified as a condition that warrants the assistance of the EMI support organization.

Goals of Problem Management

- To meet service level attainment by ensuring problems are resolved quickly,
- To satisfy customers with fast, accurate responses,
- To reduce the number of problems by prevention and permanent resolution, and
- To minimize problem impact and cost.



OIS Office of Information Services

Centers for Medicare & Medicaid Services
7500 Security Boulevard
Baltimore, MD 21244-1850

www.cms.hhs.gov
www.medicare.gov