



**F E D E R A L
S T U D E N T A I D**

We Help Put America Through School

Enterprise Application Integration (EAI) Overview Workshop

FSA Project Sponsor: Ganesh Reddy

Modernization Partner Lead: Bruce Kingsley

April 9, 2003, 1:00 p.m. – 3:00 p.m.

Conf Room: 820 UCP 221 A/B

Dial In: (877) 714-4281 #4622

accenture



EAI Workshops

Workshop 1 – April 9, 2003

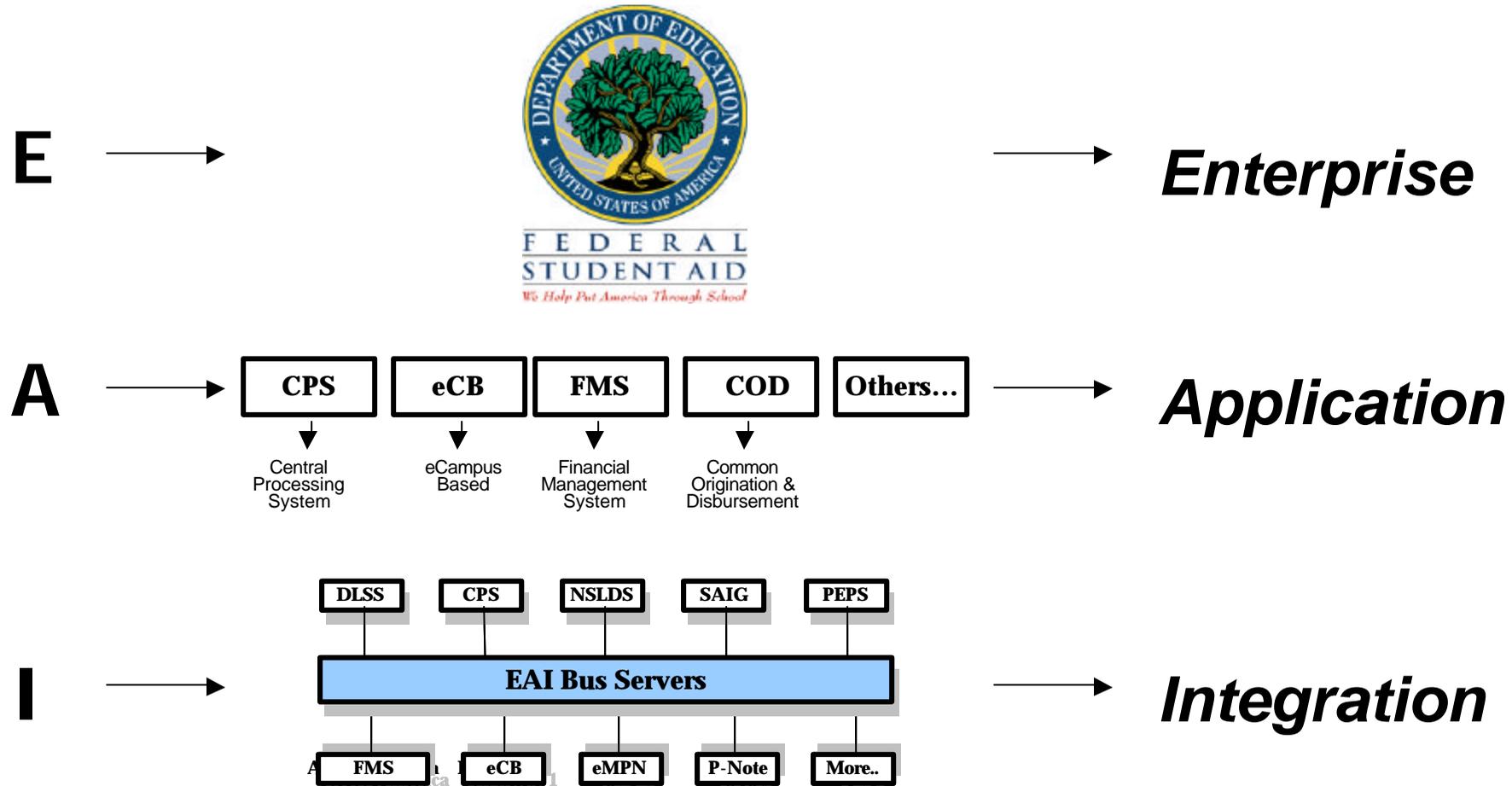
Introduction	Bruce Kingsley
FSA EAI Architecture	Bruce Kingsley
WebSphere MQ	Scott Van Velsor
Application Messaging Interface (AMI)	Bruce Kingsley
Data Integrator	Brian Whisnant
Working with EAI	Eric Suzuki
EAI Operations Support	Patrick Volpe

Workshop 2 – May 7, 2003

Introduction	Bruce Kingsley
FSA EAI Architecture	Bruce Kingsley
COD Transformation	Theresa Pak
LOWeb Interfaces	Jeff Goldhirsch
FAFSA Interfaces	Jeff Goldhirsch
EAI Performance	Eric Suzuki

1:00 – 3:00 p.m.
820 UCP 221 A/B/C

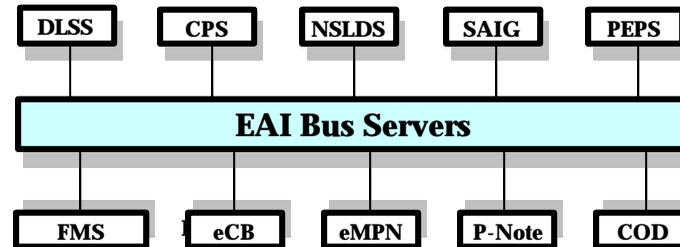
What is EAI?



...EAI means integration of applications across the enterprise.



EAI Summary



Enterprise Application Integration (EAI) provides a messaging infrastructure and integration capabilities to build standardized interfaces to new and legacy systems in support of FSA modernization objectives.

“EAI Bus”: Consists of 2 clustered servers and IBM products on target systems and applications.





EAI Architecture is Fundamental to the FSA Modernization and Integration Strategy

System Integration

Provides messaging infrastructure for integration of existing legacy systems, modernization initiatives, COTS applications, data warehouses, and web-based solutions.

Incremental Modernization

Individual components can be modernized one at a time and still function properly within the overall business context.

- Reduced risk
 - More rapid benefit realization
 - “Build a Little, Test a Little” approach
-

Data Consistency

Provides a capability that can be used to coordinate data across systems.

- Improve consistency
 - Reduce redundancy
-

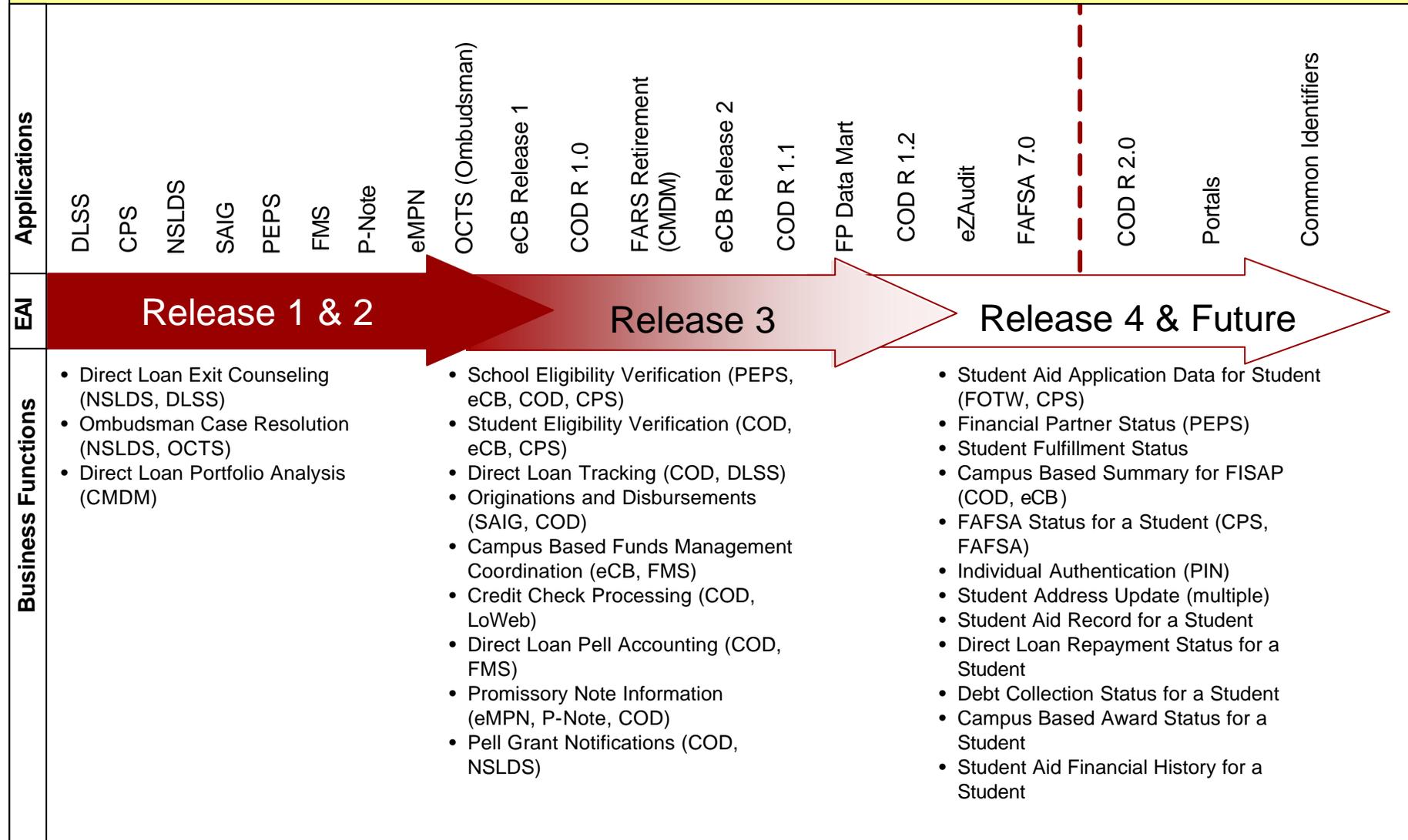
Cost Efficiency

The cost savings associated with this integrated approach are accrued to the modernization initiatives that utilize the EAI infrastructure.



Enterprise Application Integration (EAI) Overall Status at FSA

Integrated Architecture: EAI enables modernization to progress and realize benefits incrementally without requiring completion of an enormous enterprise-wide re-engineering project. Individual components can be modernized one at a time and still function properly within the overall business context.





F E D E R A L
S T U D E N T A I D

We Help Put America Through School

FSA EAI Architecture

accenture

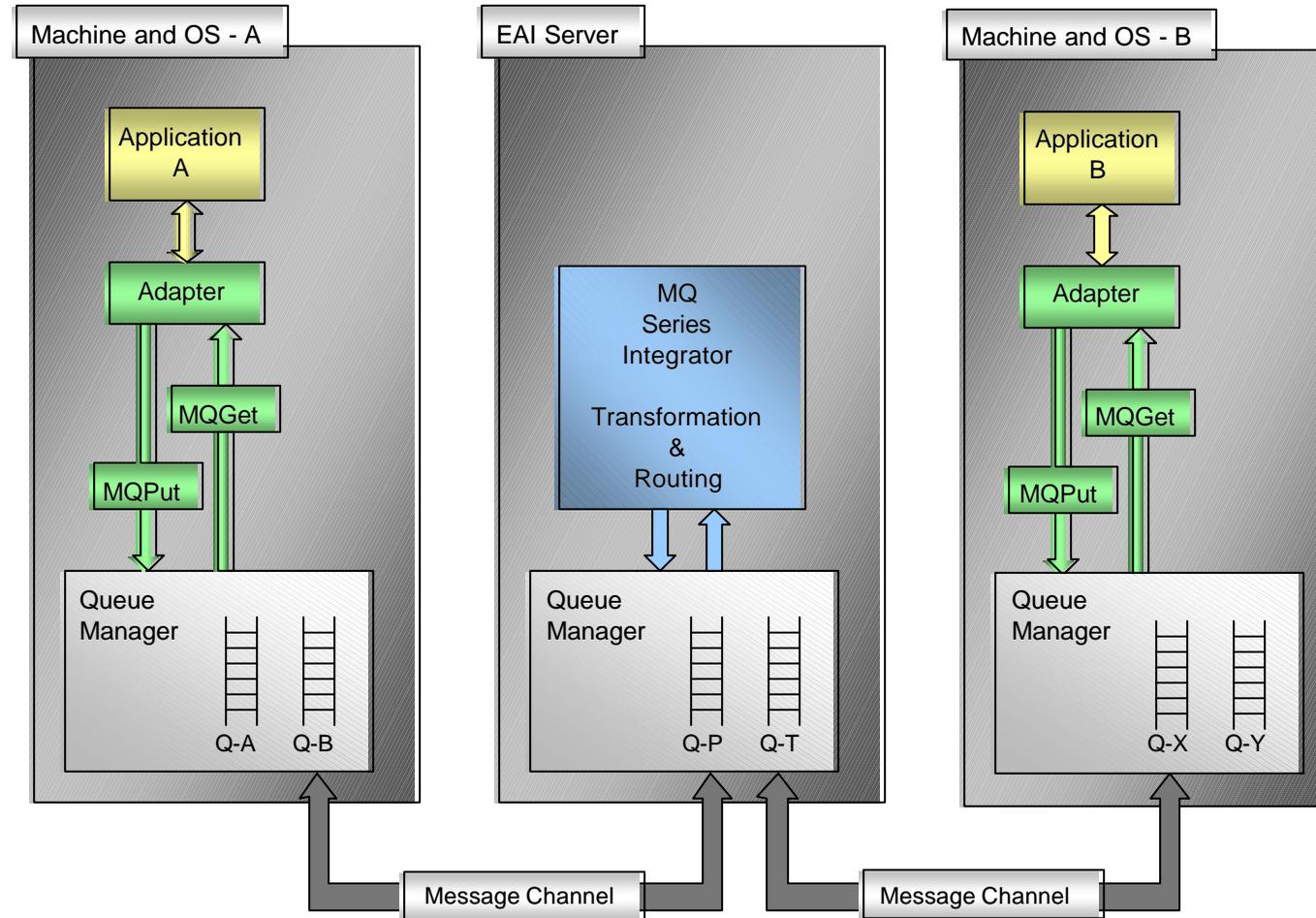


EAI Products

Product/Tool	Vendor	Description
WebSphere MQ (MQSeries)	IBM	Transport of messages between systems.
AMI	Open Applications Group (OAG)	Simplified messaging API for business application programs.
MQ System Integrator (MQSI)	IBM	Transformation, routing, and formatting of messages between systems.
Data Integrator	Commerce Quest	Large volume data transportation. Moves files larger than 100 Megabytes.

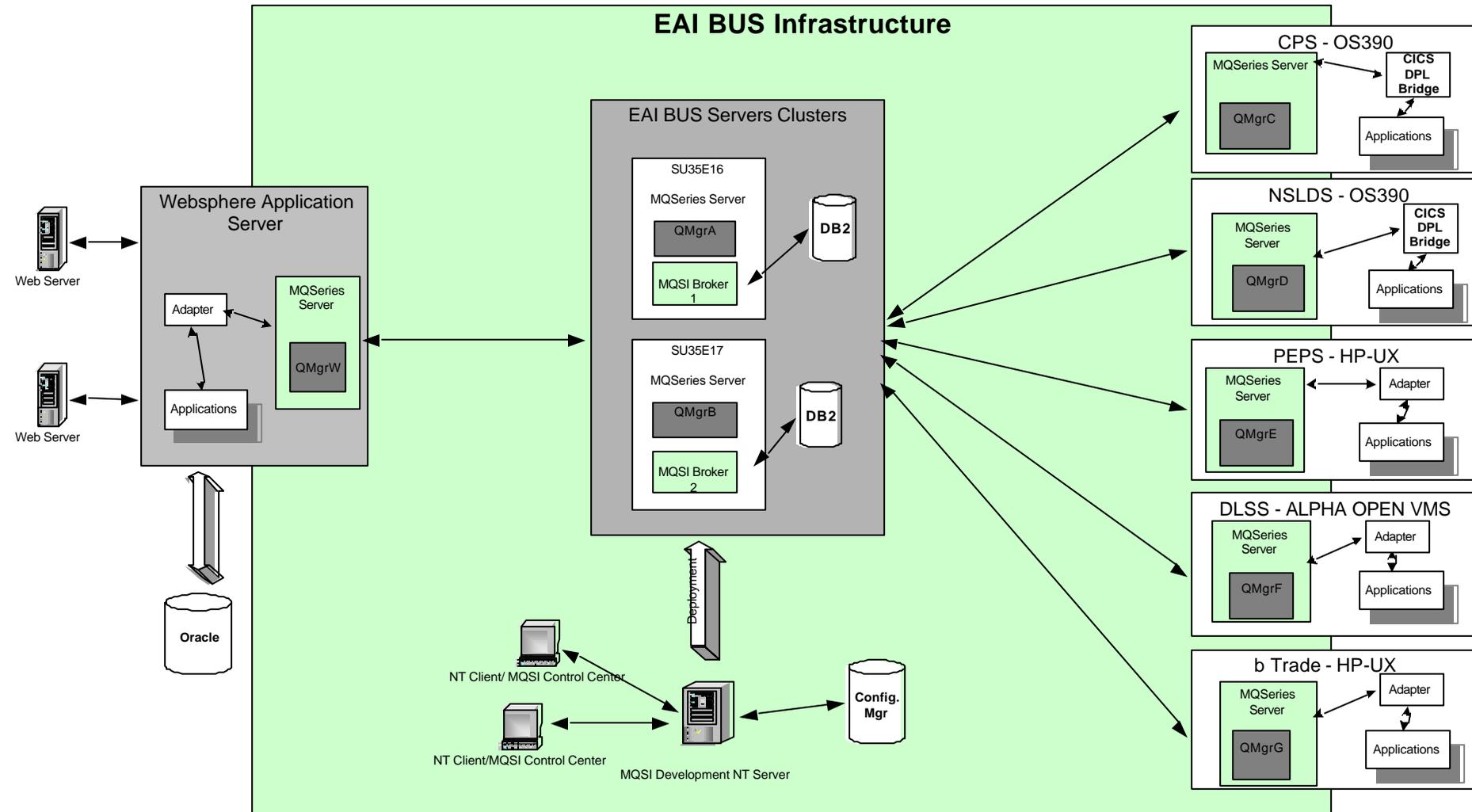


Queue Architecture Hub and Spoke



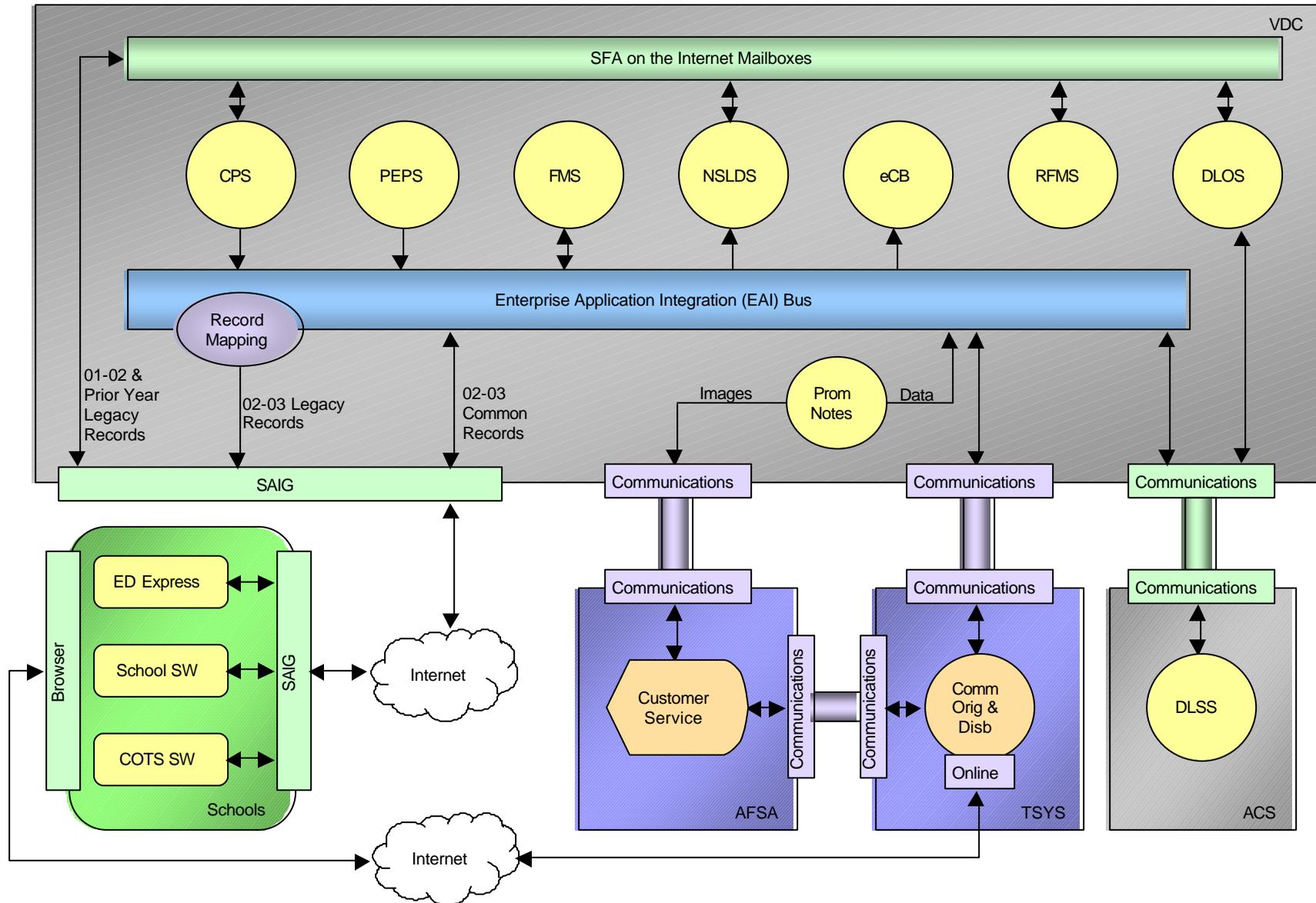


Development Environment - EAI





COD Architecture





F E D E R A L
S T U D E N T A I D

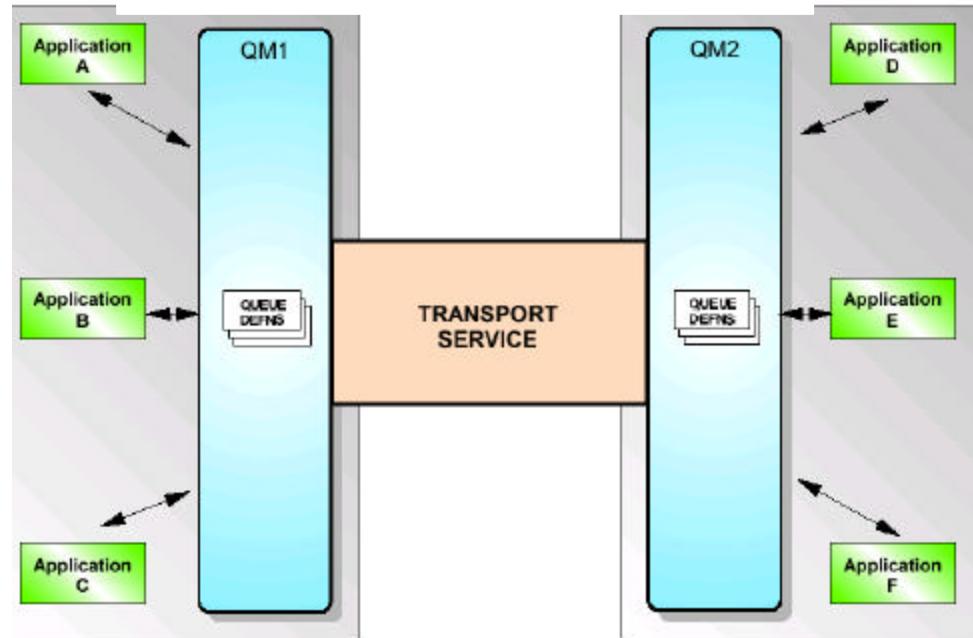
We Help Put America Through School

WebSphere MQ (MQ Series)

accenture



What is MQ Series?



- A single, multi-platform API
- Provides assured message delivery
- Enables applications to be loosely-coupled

MQ Series Messaging architectural goals:

- Time independence
- Location Transparency
- Protocol independence
- Platform Independence
- Many-to-Many Connectivity
- Application unaware



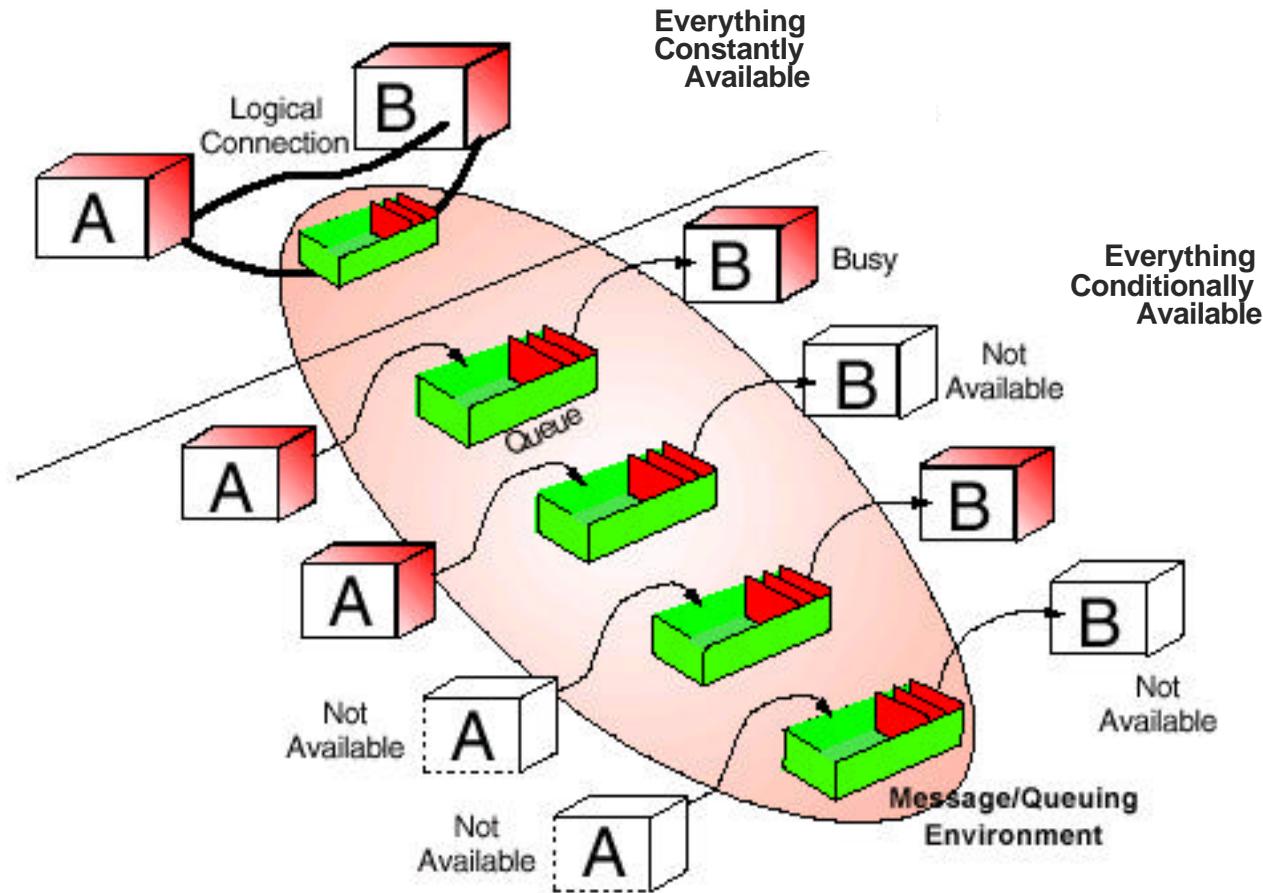
MQ Series is *NOT* a Substitute for:

- Well written applications
- Robust network
- Good operational procedures
- Well managed system





Scenarios for Loosely Coupled Applications:



The Message/Queuing environment is independent of Application A and Application B's availability.



What's a Message?

Message = Header + User Data



A Series of Message Attributes Understood and augmented by the Queue Manager

- Unique Message Id
- Correlation Id
- Routing information
- Reply routing information
- Message priority
- Message codepage/encoding
- Message format
-etc.

Any sequence of bytes

- Private to the sending and receiving programs
- Not meaningful to the Queue Manager

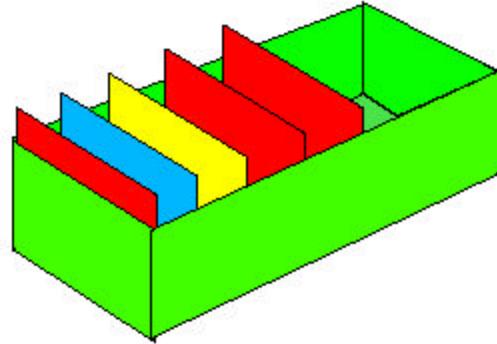
Message Types

- Persistent ... recoverable
- Non Persistent

Up to 100MB message length



Queues and Queue Managers



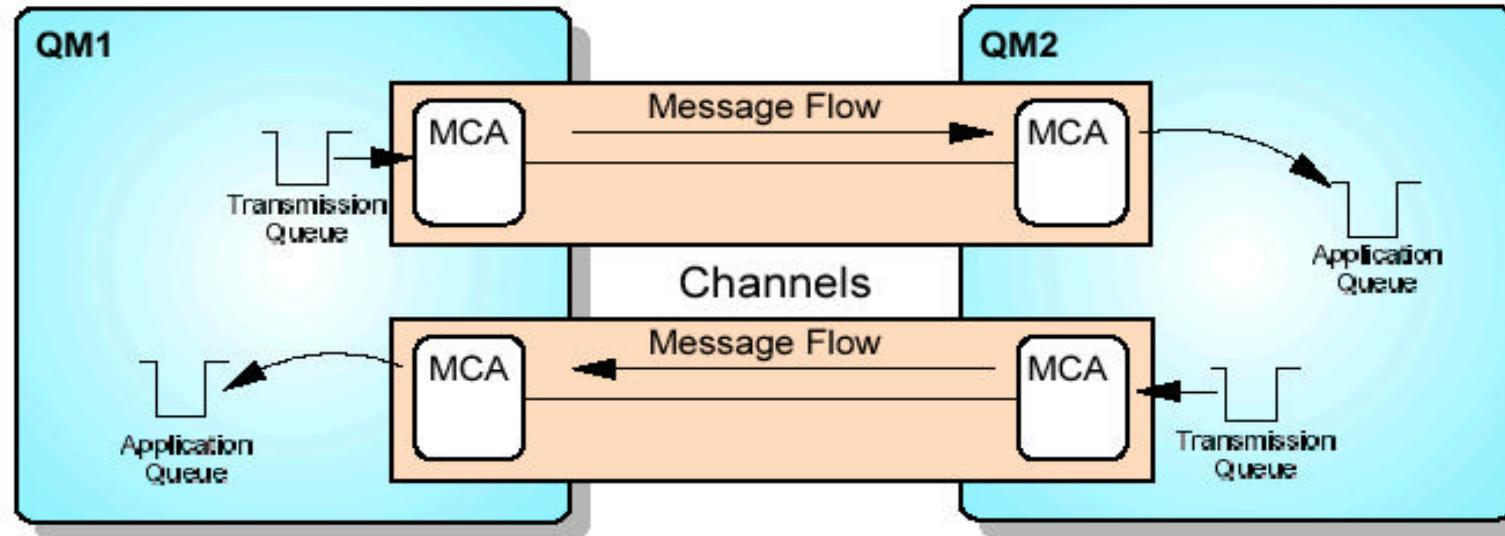
A queue is a place to hold messages.

A queue manager:

- Interacts with operating systems
- Interacts with queues
- Communicates with other queue managers



Channel Concepts



Channels are uni-directional
Channels provide for (application) session concentration
Two-way communication requires two channels



MQ Series Messaging Platforms

Servers:

- OS/390
- AIX
- Windows NT, 2000
- Solaris: Intel & SPARC
- HP-UX
- OS/400
- OS/2
- Compaq OpenVMS
- Compaq NSK
- Compaq Tru64 UNIX
- VSE/ESA
- Windows: 3.1,95,98
- Digital UNIX
- SunOS
- SCO: Openserver, UnixWare
- IRIX
- Dynix/ptx
- NCR
- TPF
- DC/OSx
- Sinix
- Linux
- PalmOS (MQe)
- EPOC (MQe)
- Java (MQe)
- Unisys 2200
- Hitachi

Clients only:

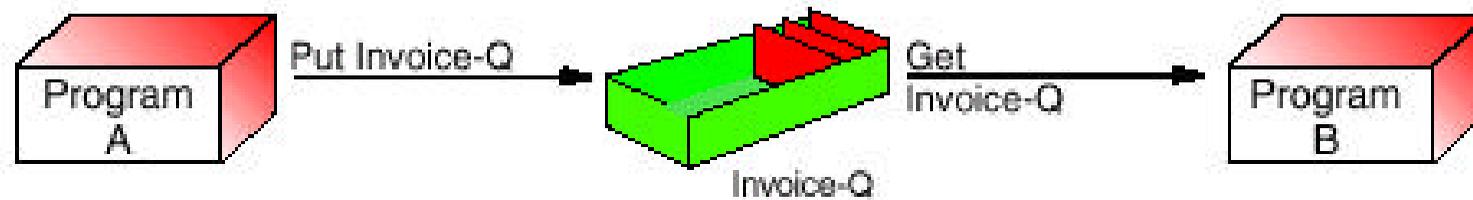
- DG/UX
- HP3000 MPE/iX
- Java
- Windows: 3.1,95,98
- DOS
- VM
- Apple MacOS
- Stratus VOS
- 4690
- Unisys A

39 Platforms

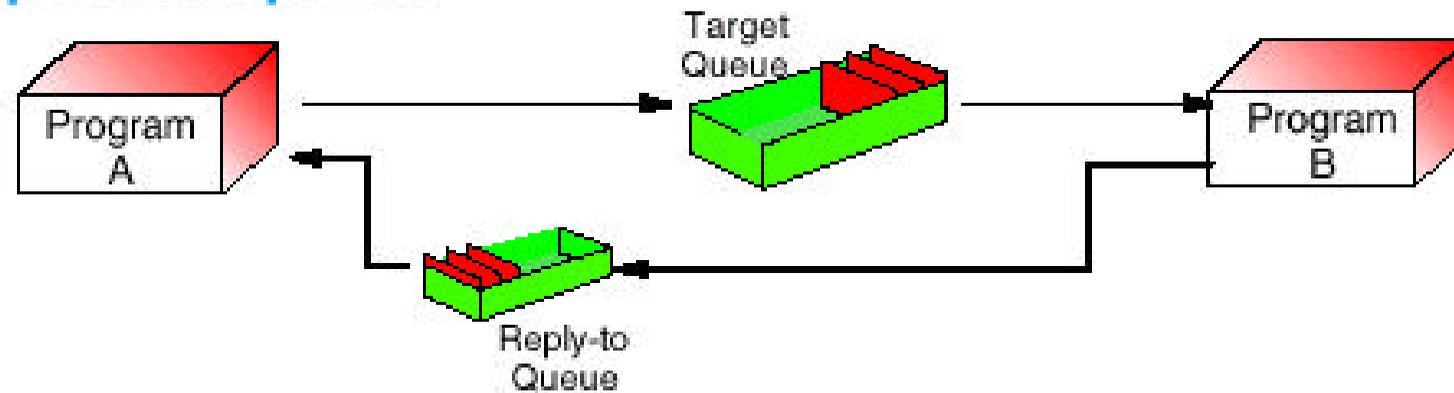


MQ Series Examples I

'Fire and Forget'



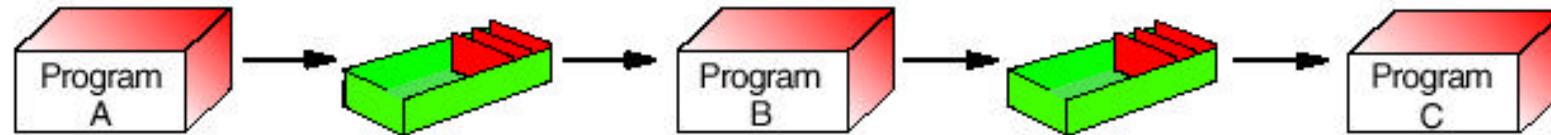
Request/Response



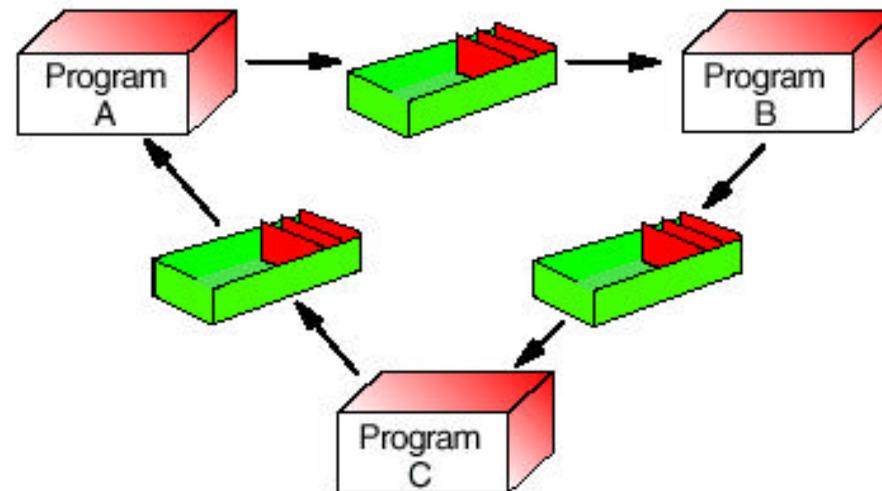


MQ Series Examples II

Chain



Loop





F E D E R A L
S T U D E N T A I D

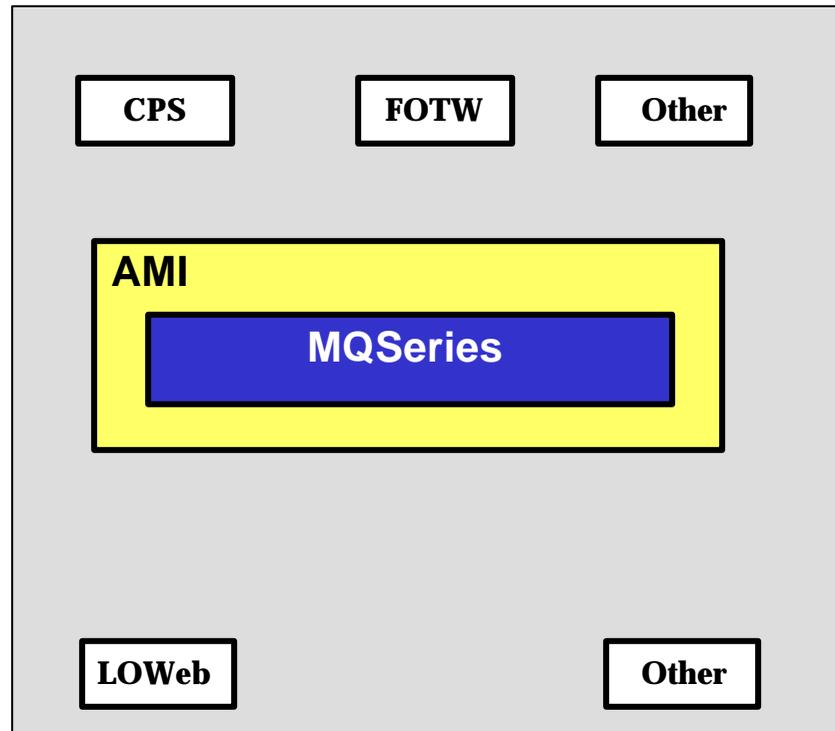
We Help Put America Through School

AMI (Application Messaging Interface)

accenture



What is AMI?



- Message transport API
- High level of abstraction
- Defines messaging “how” and “where”
- AMI supports C, C++, Java, COBOL

Application Messaging Interface (AMI) is a simplified messaging API for business application programs.



- Simplified interface
- Fewer structures, more verbs with single function
- Message handling/transport behavior options moved from program to administration domain (policies)
- Separate calls for different application styles
- Natural style for each language, C, C++, Java

Send and Forget

Request/Response

Request/Reply

Publish and Subscribe

File Transfer



“Where”, “How”, and “What” – Three concepts used by AMI to interface to MQSeries:

```
amSend(ServiceName,PolicyName,Message,...)
```

- **Service - The "Where"**
 - ▶ Defined in repository
 - ▶ Abstraction of an MQSeries Queue or collection of Queues
- **Policy - The "How"**
 - ▶ Defined in repository
 - ▶ Defines quality of service e.g. priority, persistence, confirmation level etc.
 - ▶ Defines how to handle message e.g. error handling, retries, expiry handling,...
- **Message - The "What"**
 - ▶ Message data and attributes (format, correlid,...)



AMI Procedure Style Calls

- **Setup**

- ▶ `amInitialize()`
- ▶ `amTerminate()`

- **Datagram**

- ▶ `amSendMsg()`
- ▶ `amReceiveMsg()`

- **Request/Reply**

- ▶ `amSendRequest()`
- ▶ `amReceiveMsg()`

- **Request/Response**

- ▶ `amReceiveRequest()`
- ▶ `amSendResponse()`

- **Publish/Subscribe**

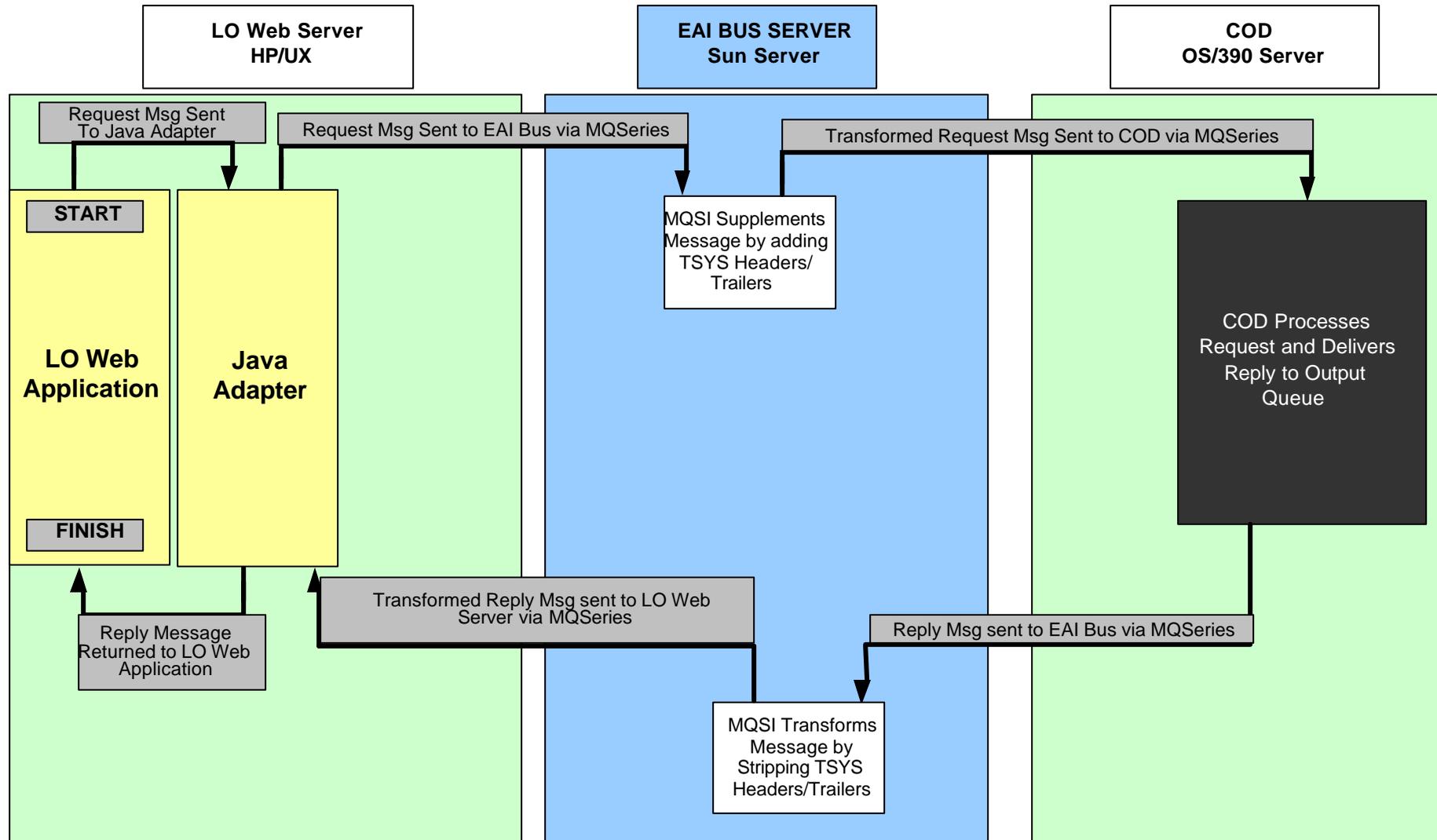
- ▶ `amPublish()`
- ▶ `amReceivePublication()`
- ▶ `amSubscribe()`
- ▶ `amUnsubscribe()`

- **File Transfer**

- ▶ `amSendFile()`
- ▶ `amReceiveFile()`

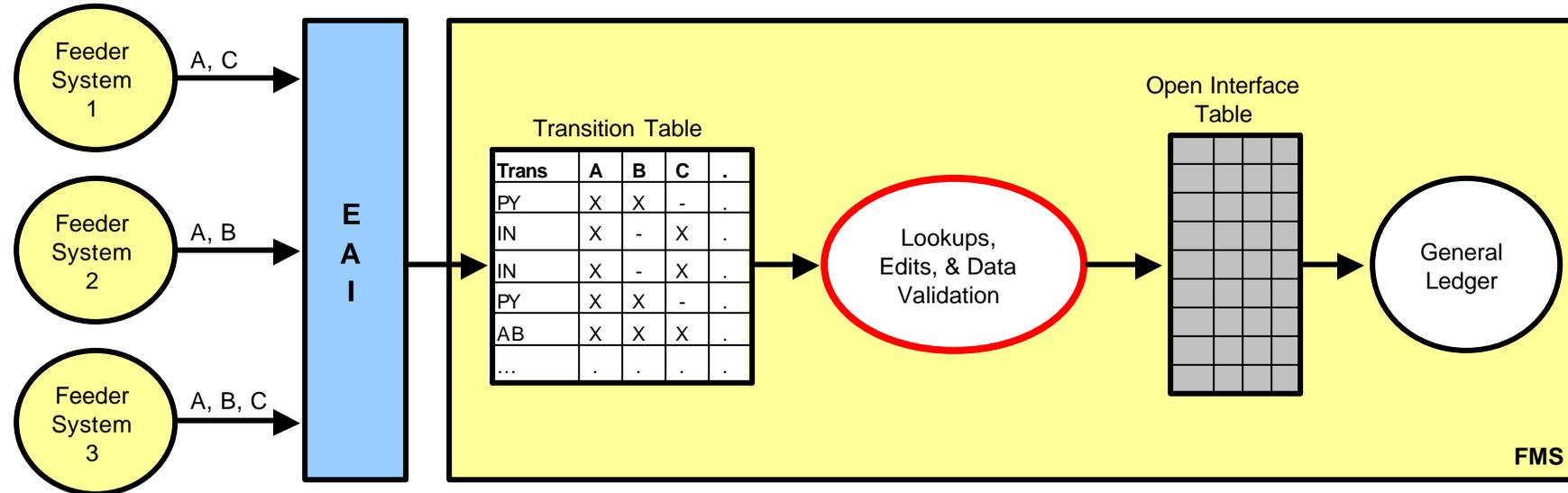


Example Transactional Interface





FMS Utilizes a Transition Table to Map Feeder System Data Elements Into FMS



Feeder Systems	EAI	Transition Table	Data Loading	OI Table	FMS Module
<ul style="list-style-type: none"> • Provide data according to own format • Different systems may provide different data elements 	<ul style="list-style-type: none"> • Routes data to appropriate Transition Table and Column • Converts data formats as necessary • Validates basic data formats 	<ul style="list-style-type: none"> • Contains columns for all possible data elements • Source system is implied by the data 	<ul style="list-style-type: none"> • Performs feeder specific processing • Performs table lookups • Performs data validation and edits 	<ul style="list-style-type: none"> • Presents input data for Module processing • One Open Interface Table for each FMS Module 	<ul style="list-style-type: none"> • One FMS Module each for General Ledger, Accounts Payable, Accounts Receivable



F E D E R A L
S T U D E N T A I D

We Help Put America Through School

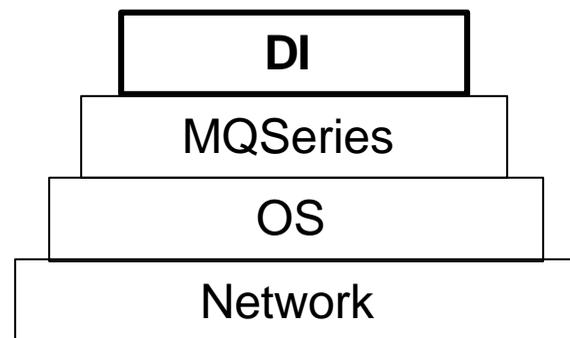
Data Integrator

accenture

What is Data Integrator?

Data Integrator, a CommerceQuest product, provides the means to exchange information between dissimilar:

- Networks
- Operating Systems
- Databases
- Applications



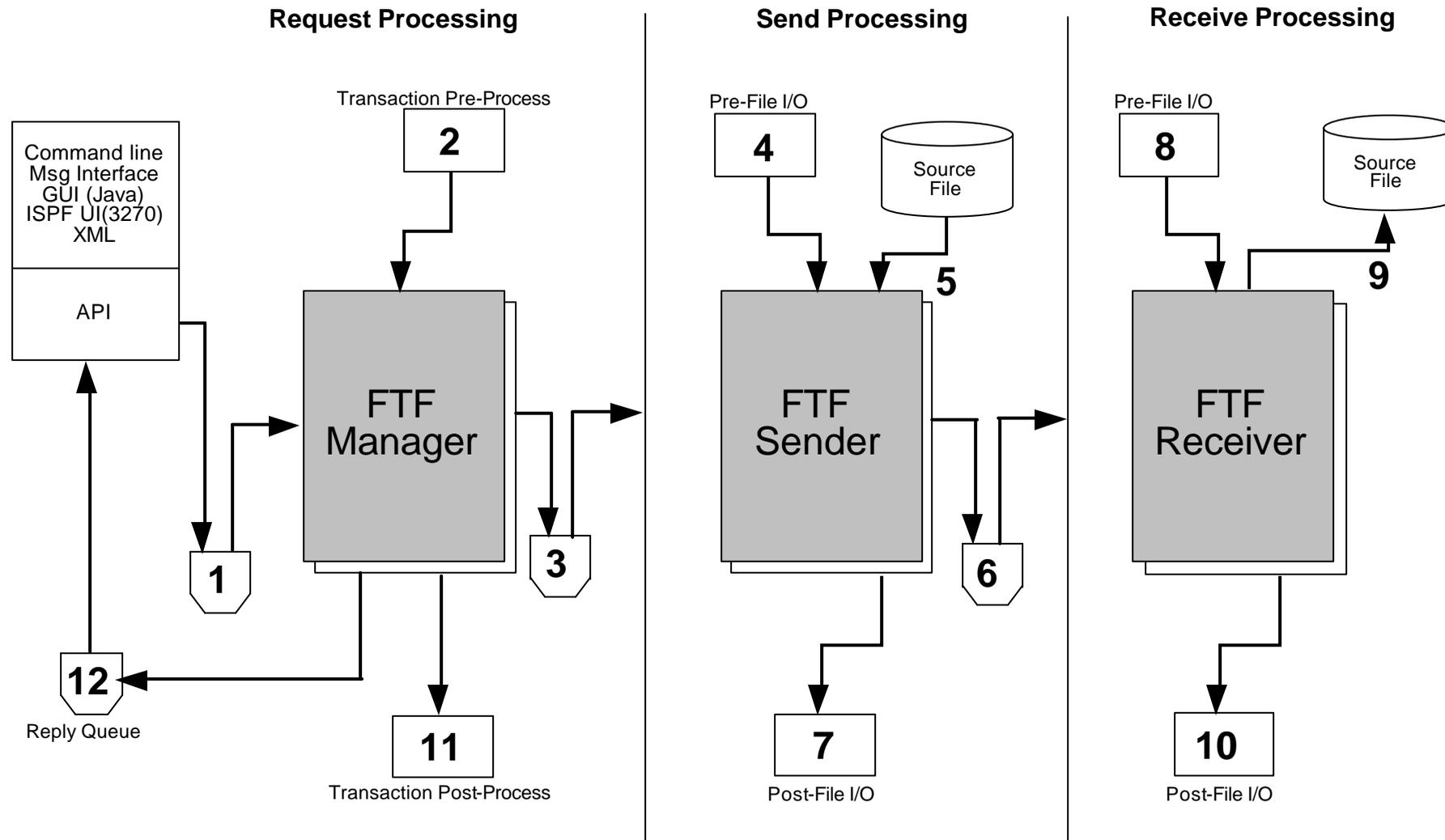


Why Use Data Integrator?

- A simple and reliable way to handle non-realtime file transfers
- DI commands can easily be included in scripts and stored commands, initiating transfers at scheduled times or when called by another process.
- File transfers with a “once and only once” guarantee
- Many existing interfaces use Data Integrator to handle bulk file transfers
 - DI is currently used by CPS, COD, DLSS, FMS, NSLDS, PEPS, SAIG
 - Several thousand school → COD records are transferred via DI every day.



Data Integrator: Flow



10. The Data Integrator is a complex system that integrates data from various sources and processes it. It consists of several components, including the FTF Manager, FTF Sender, and FTF Receiver. The FTF Manager is responsible for managing the flow of data between the user interface and the FTF Sender. The FTF Sender is responsible for sending data to the FTF Receiver. The FTF Receiver is responsible for receiving data from the FTF Sender and processing it. The Data Integrator also includes a Reply Queue and a Source File. The Reply Queue is used to store data that is being processed. The Source File is used to store data that is being processed.



DI: Flow

1. A request is submitted to Data Integrator by one of the supported interfaces. The request is passed to the Data Integrator Manager's input queue.
2. After the Data Integrator Manager accepts the request, but before processing it, the Data Integrator Manager transaction preprocess exit can be called.
3. The Data Integrator Manager submits the request to the Data Integrator Sender via the Sender's input queue.
4. After the Data Integrator Sender accepts the request, but before processing the data being transferred, the Sender can call the sender pre-process exit to perform application-specific processing.
5. The Data Integrator Sender reads and transforms the data into MQSeries messages.
6. The messages that make up the data are submitted to the Data Integrator Receiver via the Receiver's input queue and data queues.
7. After processing the data, the Data Integrator Sender can call the post-process exit to perform application-specific processing.
8. After the Data Integrator Receiver accepts the data, but before processing it, the Data Integrator Receiver can call the receiver pre-process exit to perform application-specific processing.
9. The Data Integrator Receiver retrieves the data messages and processes the data accordingly.
10. After the Data Integrator Receiver processes the data, it can call the Data Integrator Receiver post-process exit to perform application-specific processing.
11. The Data Integrator Manager receives all responses and ends the logical unit of work (LUW). Before ending the LUW, the Data Integrator Manager can call the manager post-processing exit.
12. An optional response is delivered to the appropriate end-user interface indicating that the data transfer has completed.



Manager

- ❖ Manages status of all transfers
 - reads its input queue
 - creates log entries
 - submits stat msgs to queues

- ❖ Starts and stops all transfer units of work

- ❖ Correlates all operational replies and reports final status of the transaction (not to be confused with status msgs)
 - Request completed successfully
 - Request failed
 - Request expired
 - Request canceled



Sender

- ❖ Transforms the data into MQSeries messages
 - reads its input queue
 - creates log entries
 - submits status messages
- ❖ The sender is always where the source data resides
- ❖ Updates the DI Manager with operational replies



Receiver

- ❖ Receives incoming data from MQSeries.
 - reads its input queue
 - creates log entries
 - submits status messages

- ❖ The receiver is always the destination for the data

- ❖ Updates the DI Manager with operational replies

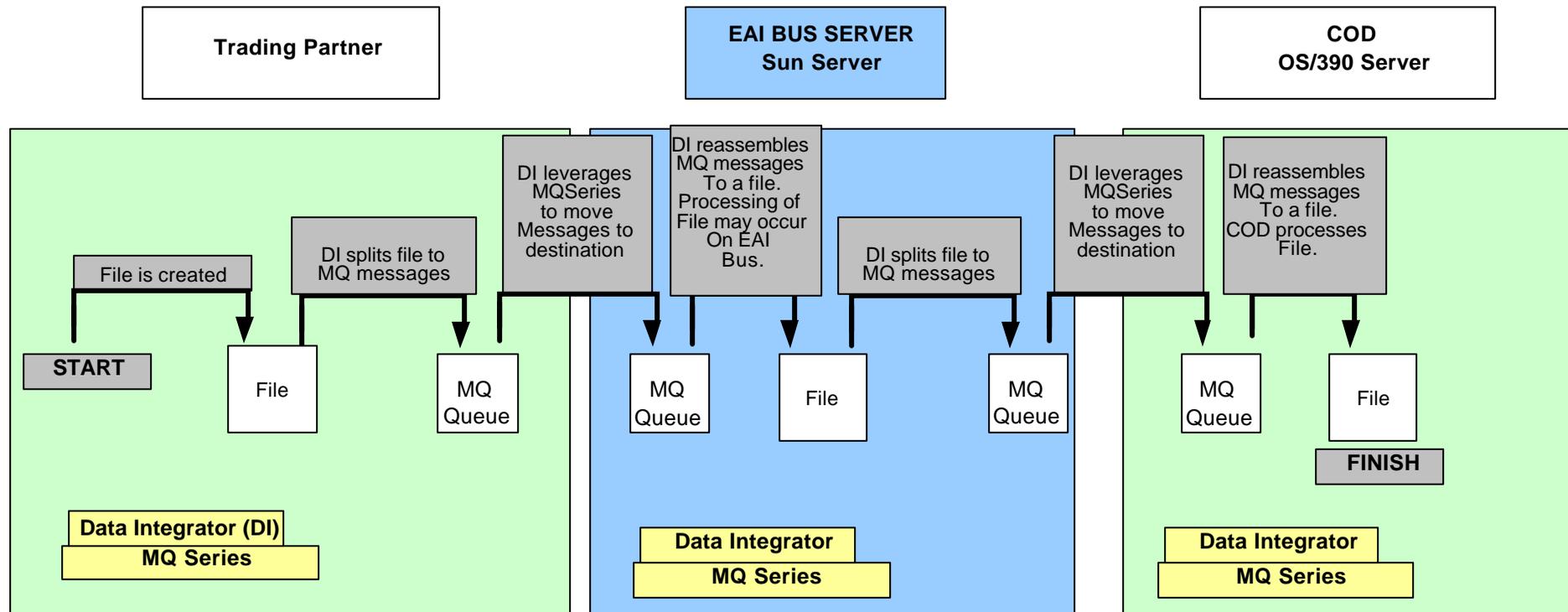


Status

- ❖ Status messages are received by each of the DI components and are not necessary for internal DI processing
- ❖ Status messages provide a reporting system that report on current and past status of data-transfer requests
- ❖ Status messages are MQSeries messages destined for the queue or list of queues defined in the DI config file

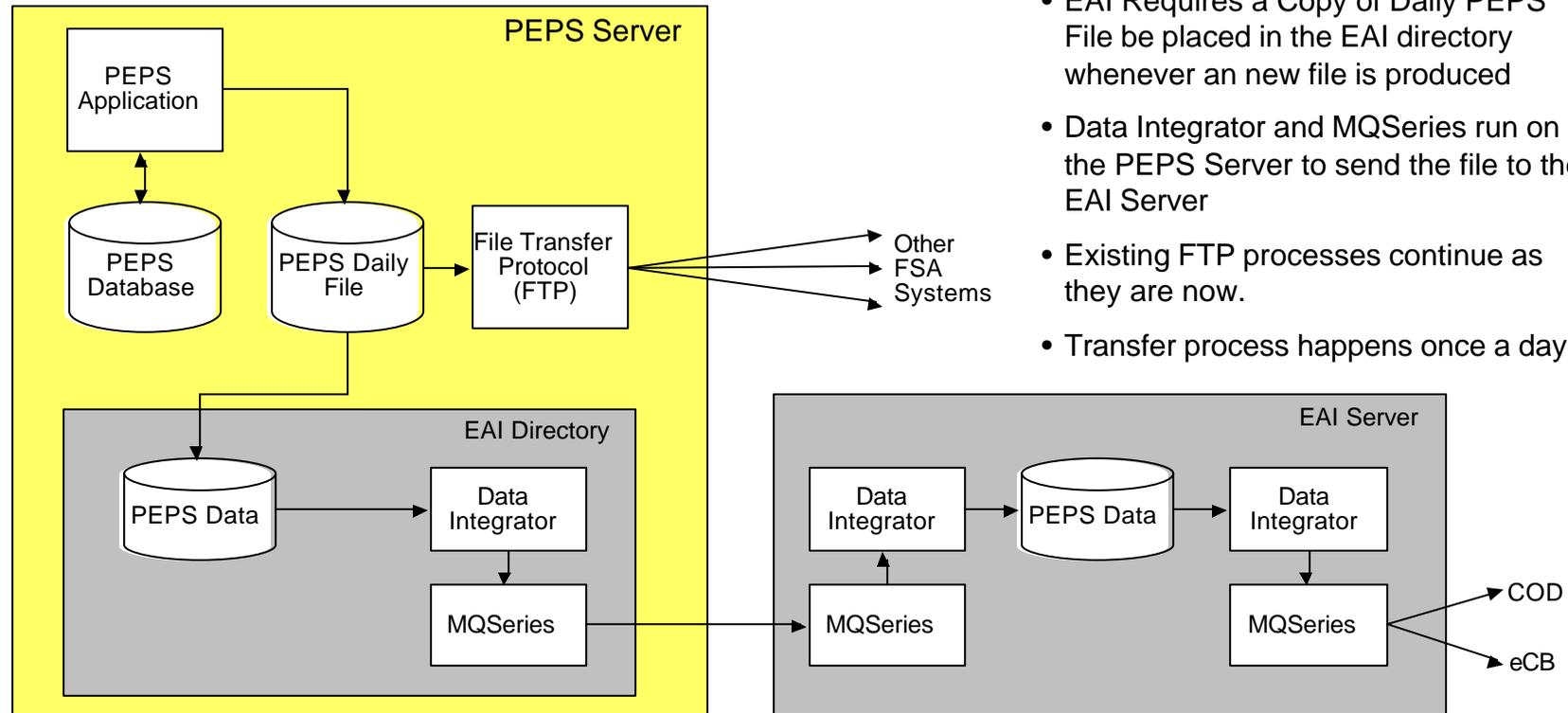


Example Bulk Data Transfer





The PEPS EAI Interface Minimizes the Impact on PEPS Resources



- EAI Requires a Directory and Workspace on PEPS Server
- EAI Requires a Copy of Daily PEPS File be placed in the EAI directory whenever a new file is produced
- Data Integrator and MQSeries run on the PEPS Server to send the file to the EAI Server
- Existing FTP processes continue as they are now.
- Transfer process happens once a day.



F E D E R A L
S T U D E N T A I D

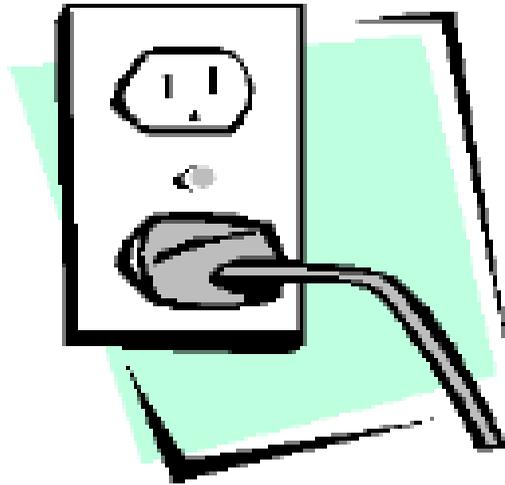
We Help Put America Through School

Working with the EAI Team

accenture



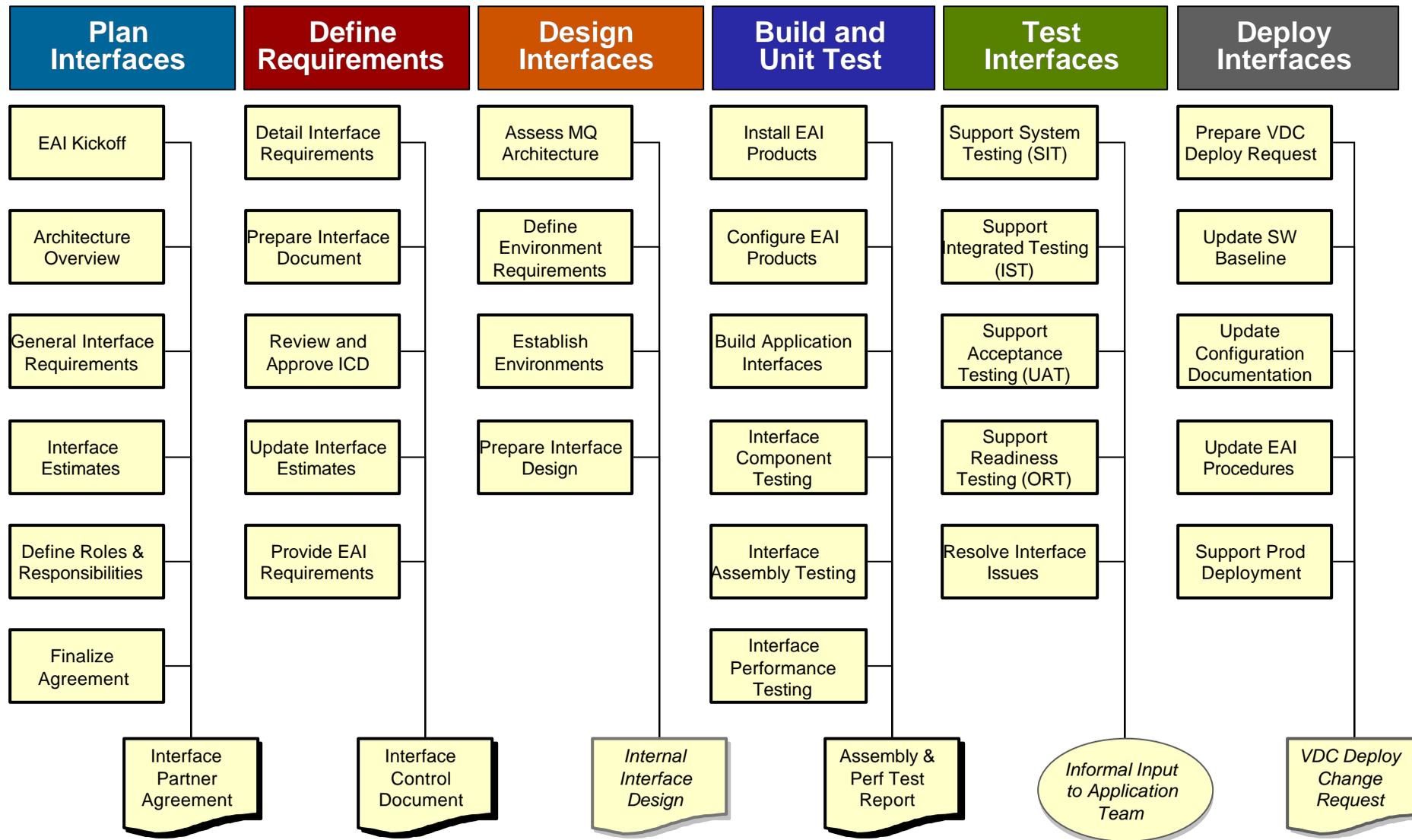
Building EAI Interfaces



- **EAI Architecture provides services to be utilized by application teams.**
- **The EAI Core Architecture implements the EAI products on the legacy system platforms and validates the ability to process legacy system functionality.**
- **Application development teams “plug” into these services and develop application interfaces.**

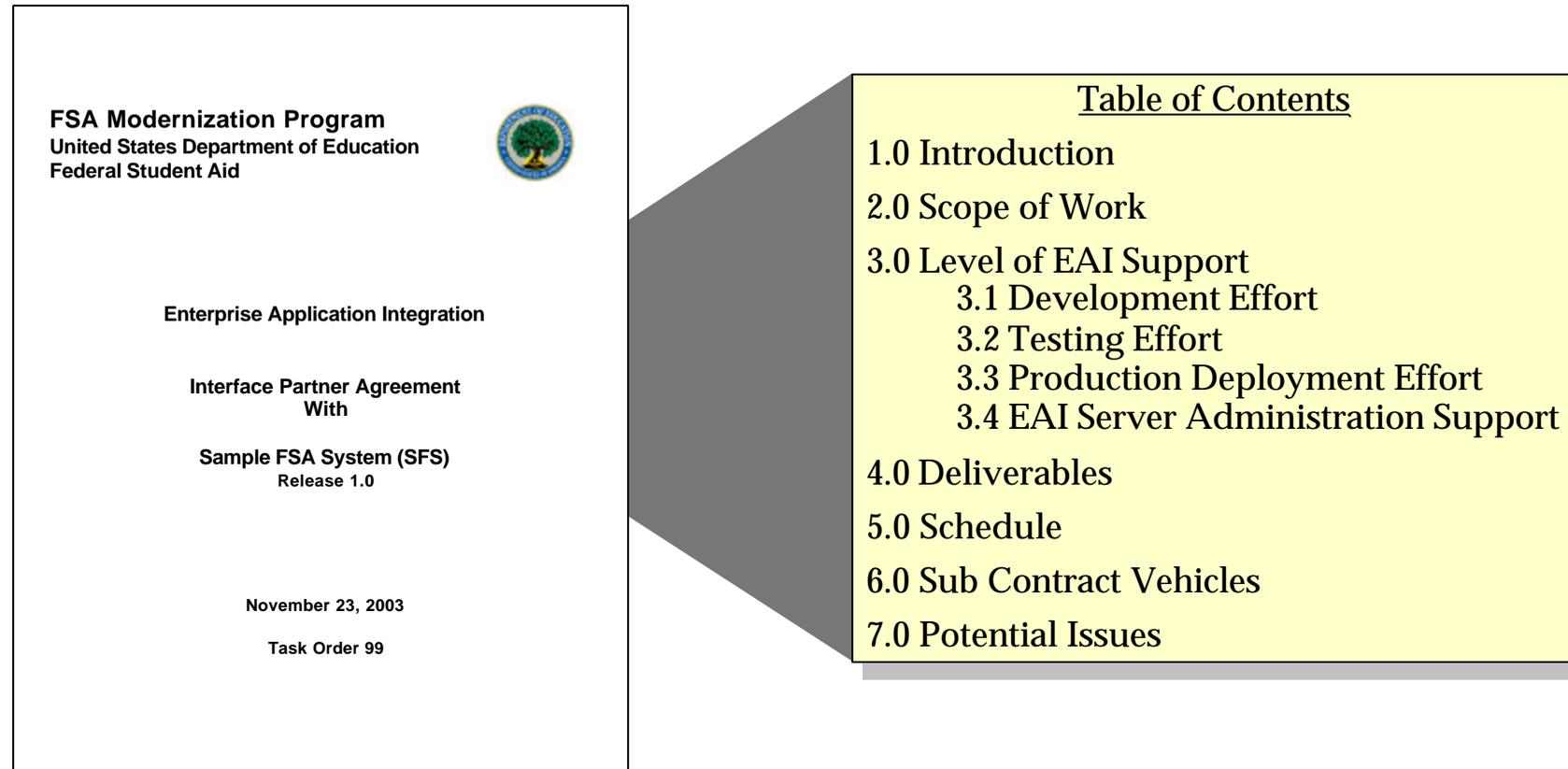


EAI Team Interface Development Tasks





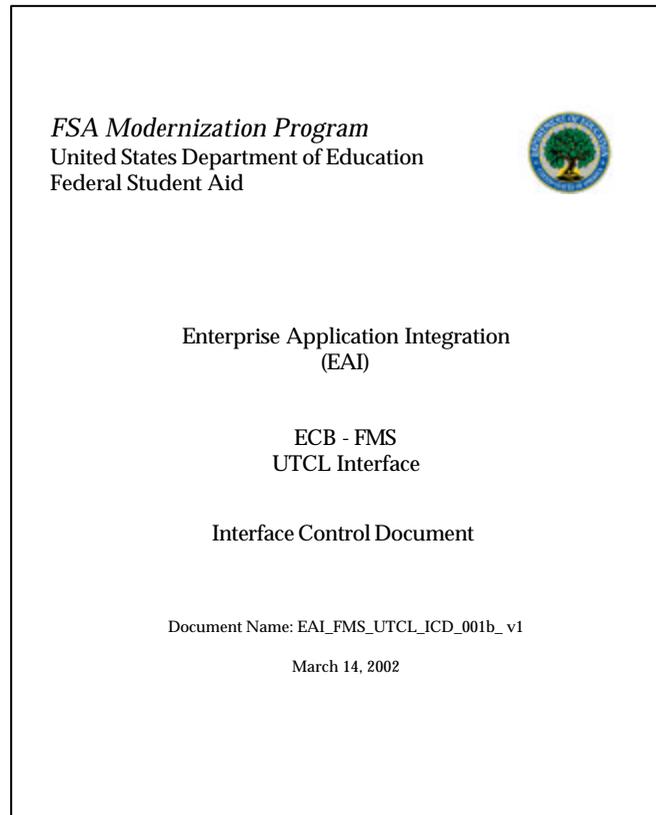
Interface Partner Agreement



The IPA documents the agreement between the Application Team and the EAI Team.



Interface Control Document



- Interface Control Specification
- Interface Overview
- Flow Diagram
- Usage Scenarios
- System Hardware/Software Components
- System Additional Information
- Source System Responsibilities
- Destination System Responsibilities
- Interface Assumptions
- Interface Dependencies
- Table Layout
- Error Messages
- Issues and Considerations

The ICD provides the specifications and requirements for the interface.



Test Documentation

Test Conditions

Condition Number	Interface Component	Test Condition Description	Req X-Ref	Expected Results	Cycle

- **Assembly Testing** verifies the correct operation of the interface
- **Performance Testing** ensures interface performance is adequate

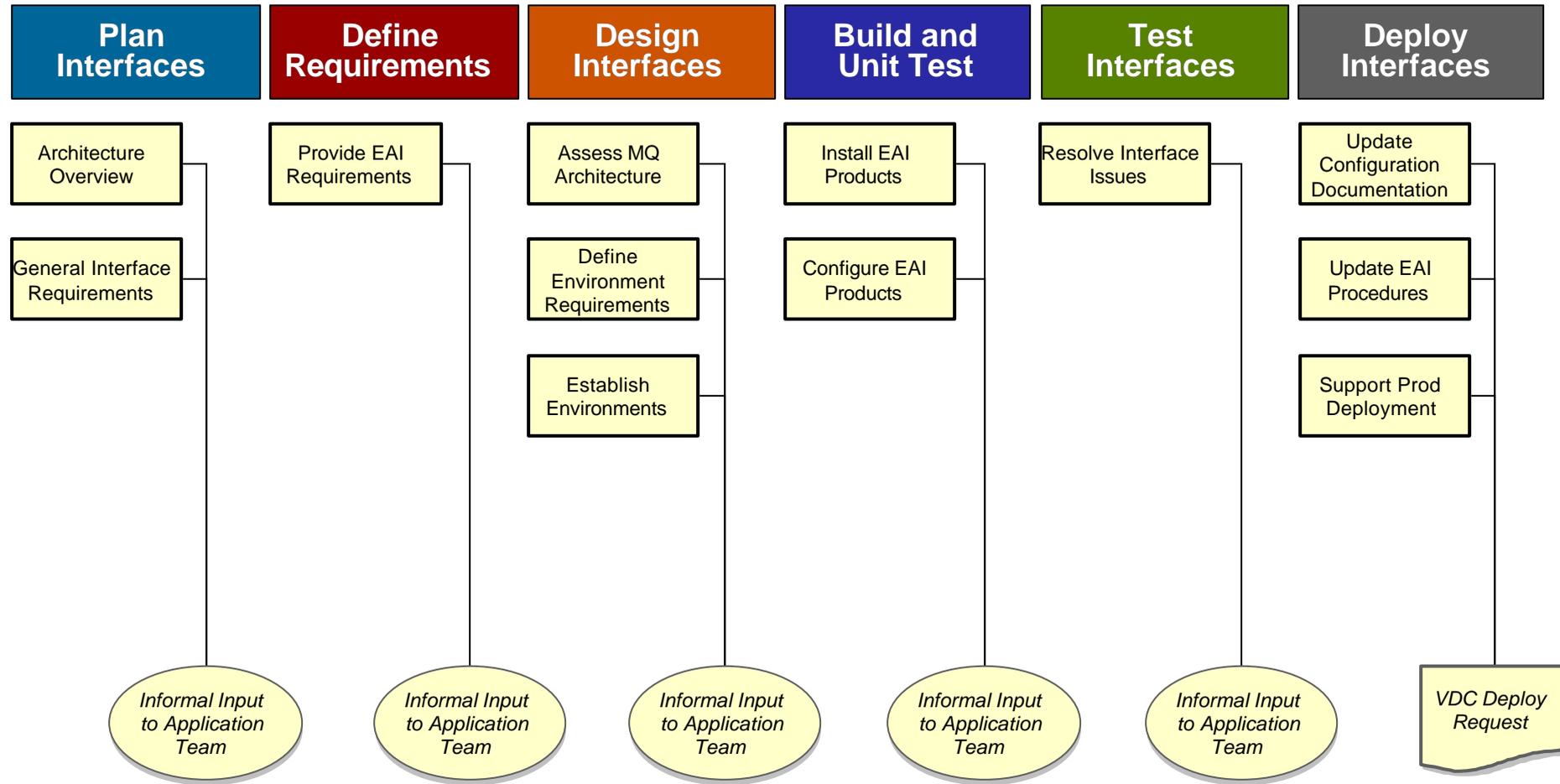
Test Script

Step Number	Interface Component	Condition Number	Action	Input Data	Expected Results	Pass/Fail	Issue Number	Comments

Test Conditions and Test Scripts document the Testing and the Results



EAI Team Interface Support Tasks



An abbreviated process can support Application Teams that develop interfaces on their own.



Costs Include Development and Operations & Maintenance

	Development		Operations & Maintenance	
What is it for?	Core MQ Capability	Application Interface	System Operations and Maintenance	EAI Maintenance and Support
Who will perform the work?	EAI Team	EAI Team & Application Team	VDC	EAI Team
Who will bear the cost?	CIO	Dev App Team	Business Channel (allocated by usage)	Business Channel



Working with the EAI Team: Estimates

First we evaluate the complexity of tasks by category. For example...

Category	Complexity		
	Simple	Medium	Complex
Install & Configure	<ul style="list-style-type: none"> - Basic identification and assessment of the target machine at the VDC - Installation and configuration of MQ Software. <p>General MQ tasks include MQ Build, Create Q Manager, Set up Ports, Define 'Non-Standard' Queues, Start/Stop Scripts,</p>	<ul style="list-style-type: none"> - Basic identification and assessment of a target location at a location outside of the normal data centers - Installation and configuration of MQ Software. <p>General MQ tasks include MQ Build, Create Q Manager, Set up Ports, Define 'Non-Standard' Queues, Start/Stop Scripts,</p>	<ul style="list-style-type: none"> - Identification and assessment of the target machine is more difficult as it is located in a distant location. - Questions about installation and configuration of MQ Software. <p>General MQ tasks include MQ Build, Create Q Manager, Set up Ports, Define 'Non-Standard' Queues, Start/Stop Scripts'</p>



Working with the EAI Team: Estimates

Then we estimate the effort by task, category and quantity...

Detail Task Description	Category	Quantity	Complexity	Requirements/ Analysis	Total Detailed Design	Build & Test	Assembl y Test
Establish network connectivity between systems, enable hardware based encryption	TechArch	1	S	2	20	16	8
Install and Configure MQ Series	Install & Configure	1	S	4	0	40	0
Build Core Adapter (Basic communication channels and flow)	Core Adapters	1	S	4	41	25	12
Create Interface (Detailed interactions, breakdown of data from files to records, evaluation of header/trailer info, pub/sub config etc...)	Interface	1	S	2	20	21	10
Creation of custom program to supplement transformation or interface	Custom Dev	1	S	4	41	41	20
Accommodate transformation of data (MQSI)	Transformations	1	S	3	31	10	7

...but the estimates are highly dependent on the detailed information we receive from Application Teams.



Working with the EAI Team: Estimates

And finally the EAI Team reviews/refines the estimates with the the Application Team. Estimates can vary greatly depending on Application needs. For example...

Task	Effort (Hrs)
	Simple - Complex
Identify and Access Target Machine	8 - 40
Install MQ Software (MQ Series, Data Integrator, MQMon)	24 - 40
Configure MQ Software and Verify (May include building an interface for test purposes)	24 - 64
Identify Application Interface Functions (Define functional requirements, formats, layouts, Prepare Interface Control Document)	40 - 320
Build Application Interfaces	24 - 185
Test Interfaces (functional and performance)	40 - 120
Migrate Into Production (including Ops Readiness Test)	20 - 90



FEDERAL
STUDENT AID

We Help Put America Through School

EAI Operations Support

accenture



EAI Support Overview

Support for the EAI Architecture consists of the following:

- 24 x 7 x 365 Production support
- Tier II, Architecture level support
- EAI Architecture Subject Matter Expert (SME) knowledge
- EAI Application Subject Matter Expert (SME) knowledge
- EAI Application Vendor escalation support
- Monthly Support scheduled published to VDC

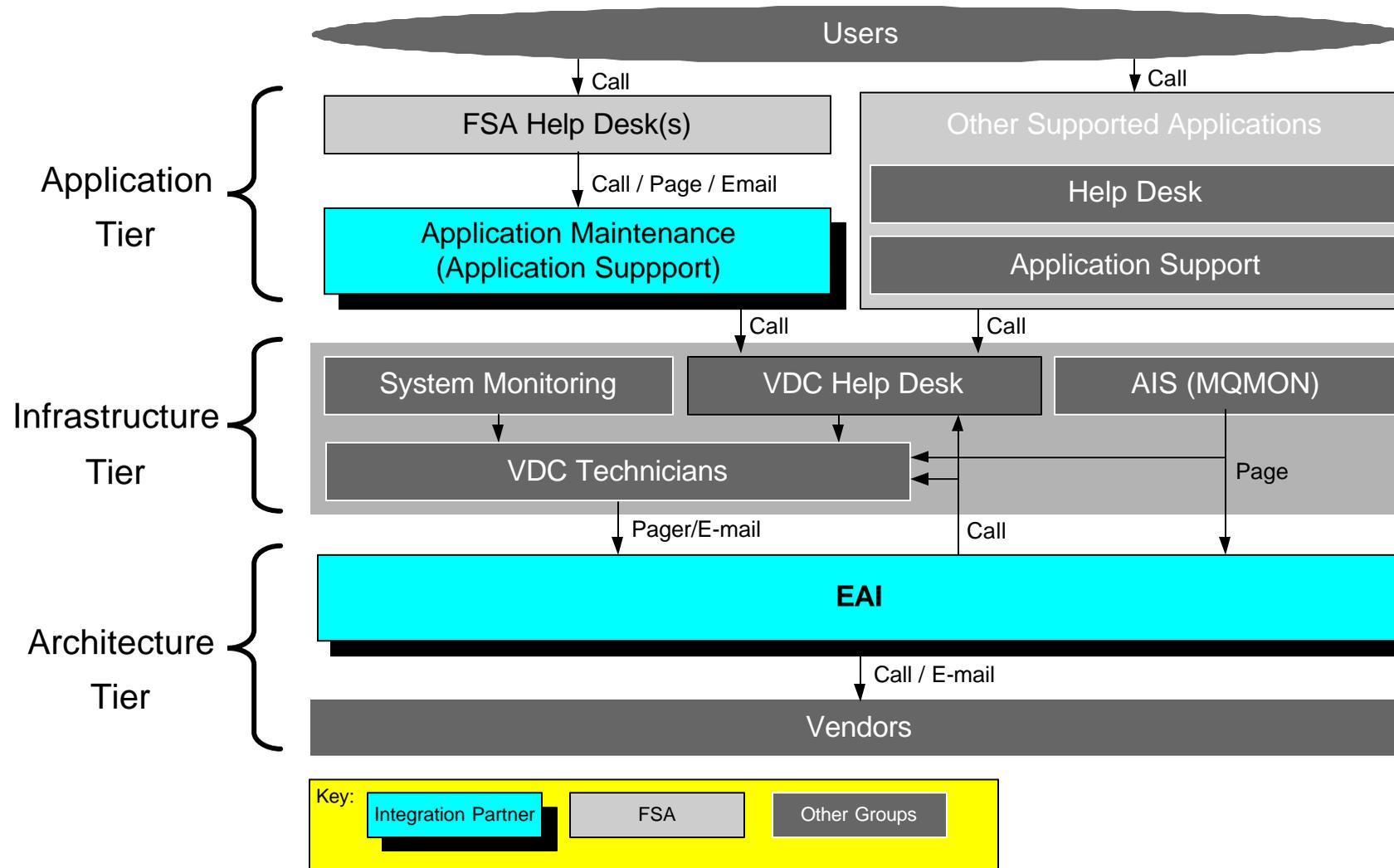


Support Responsibility By Layer

Layer	Example	Development	Maintenance	Monitoring	Operations
Application Software supporting a business function	COD, CPS, DLSS, DLCS, NSLDS, PEPS, FAFSA, Students.gov, Schools Portal	Individual Support Contractors	Individual Support Contractors	Individual Support Contractors	Individual Support Contractors
Infrastructure Platforms for Architecture and Applications	Servers, O/S, Database, Network, Routers, Firewall	VDC Operations (CSC)	VDC Operations (CSC)	VDC Operations (CSC)	VDC Operations (CSC)
Architecture Common Capabilities used by Applications	IBM MQSeries, MQSI, DI, Web Server, Application Server, Search Engine	Integration Partner	Integration Partner	VDC Operations (CSC)	VDC Operations (CSC) / Integration Partner



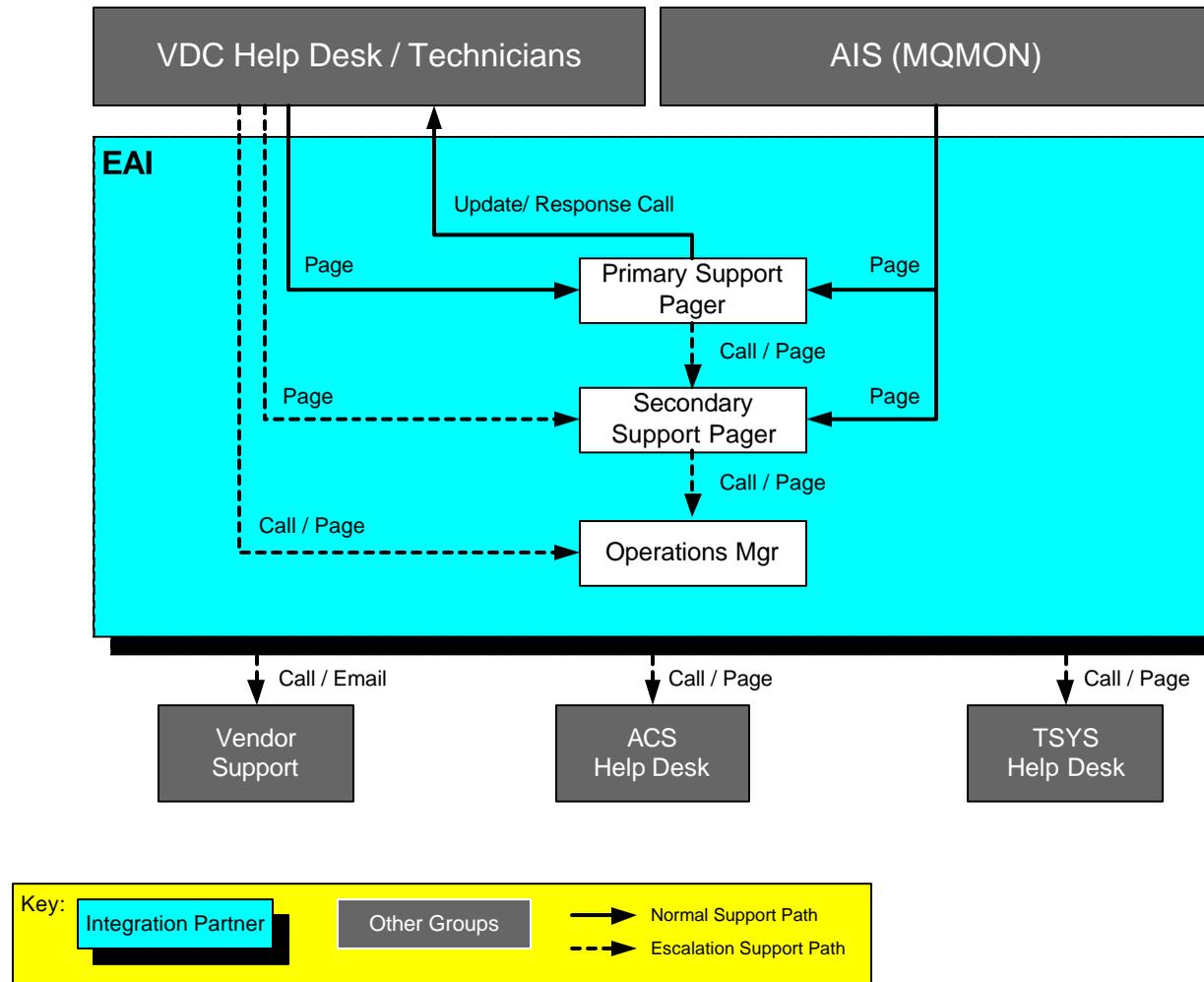
Operations Support Model





Operations Support Model

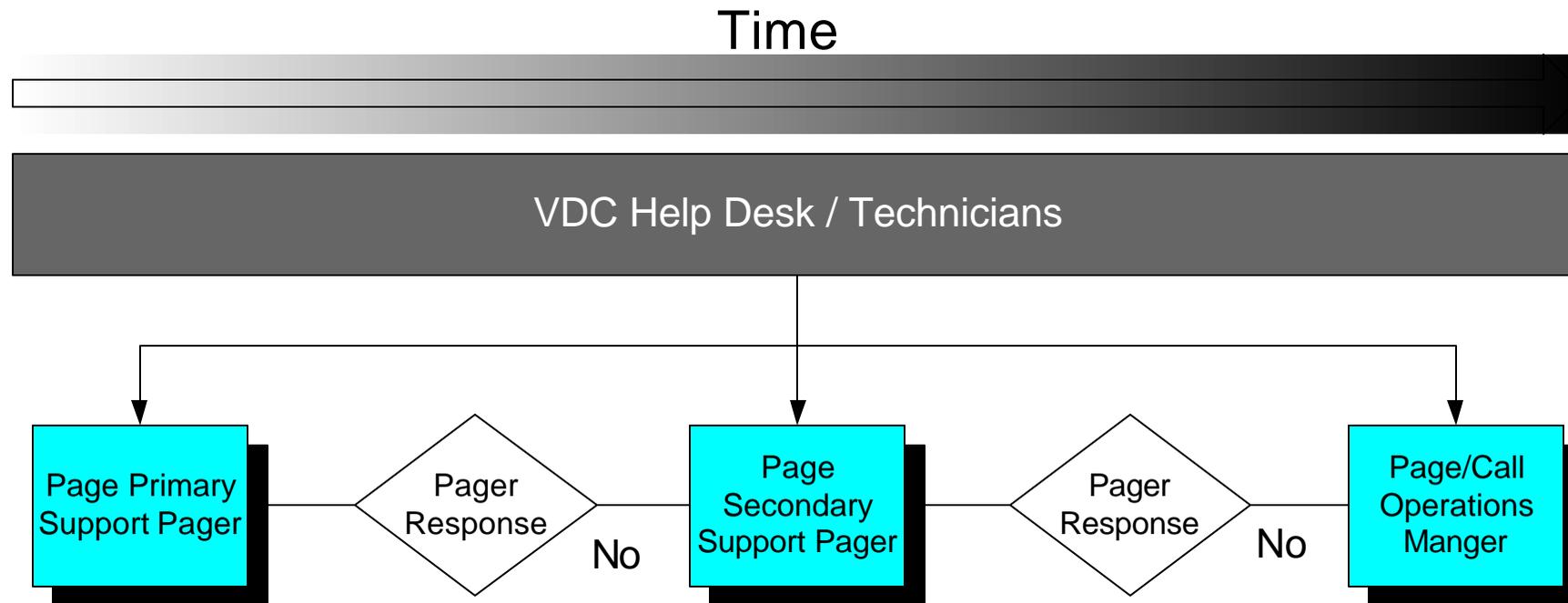
The Primary support person is single point of contact for Production issue resolution. Escalation to next levels if required. solution.





Support Escalation Procedures

VDC technicians have defined escalation procedures to follow.





How To Contact EAI Support

Call VDC Help Desk at

877-366-3338, Option 3 (FSA)

- Provide name of application (server name, if known)
- Specify if issue is Production impacting
- Request to have EAI Support paged
- Provide your contact information

If you have support related questions or issues, contact

Patrick Volpe, EAI Operations Manager

Work #: 202-962-0743

Pager#: 877-578-8795

Email: patrick.e.volpe@accenture.com