FEDERAL STUDENT AID

We Help Put America Through School

Enterprise Application Integration (EAI) Transactional Interfaces

FSA Project Sponsor: Ganesh Reddy
Modernization Partner Lead: Bruce Kingsley

May 7, 2003, 1:00 p.m. – 3:00 p.m.
Conf Room: 820 UCP 221 A/B/C
Dial In: (877) 714-4281 #0876

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
Transactional Interfaces	Jeff Goldhirsch
EAI Performance	Eric Suzuki & Scott Gray

1:00 – 3:00 p.m.

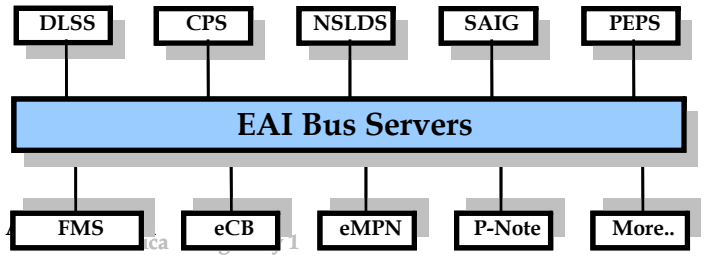
820 UCP 221 A/B/C

What is EAI?

E → FEDERAL STUDENT AID — We Help Put America Through School → ***Enterprise***

A → CPS | eCB | FMS | COD | Others… → ***Application***

- CPS: Central Processing System
- eCB: eCampus Based
- FMS: Financial Management System
- COD: Common Origination & Disbursement

I → 

→ ***Integration***

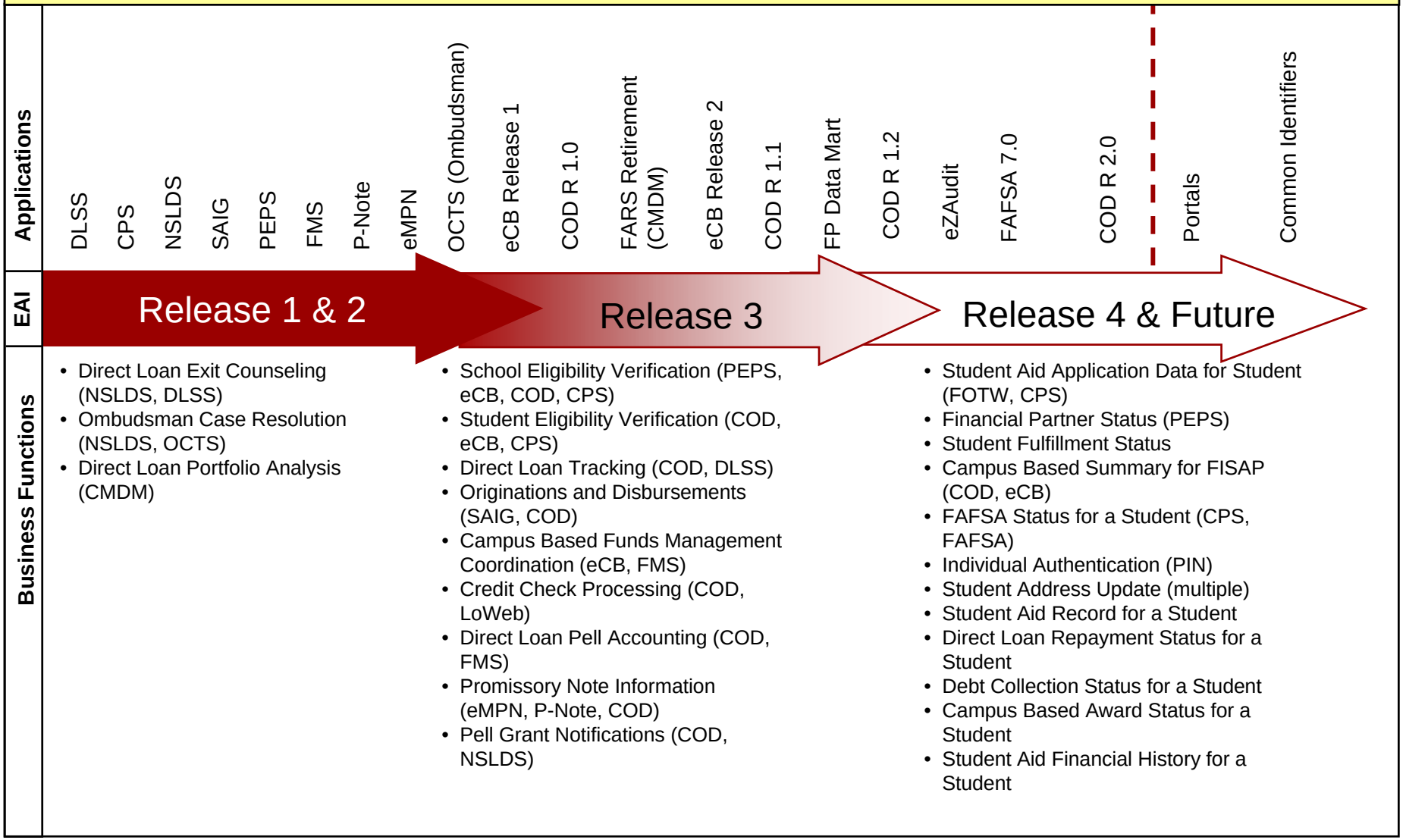
…EAI means integration of applications across the enterprise.

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.



FEDERAL STUDENT AID

We Help Put America Through School

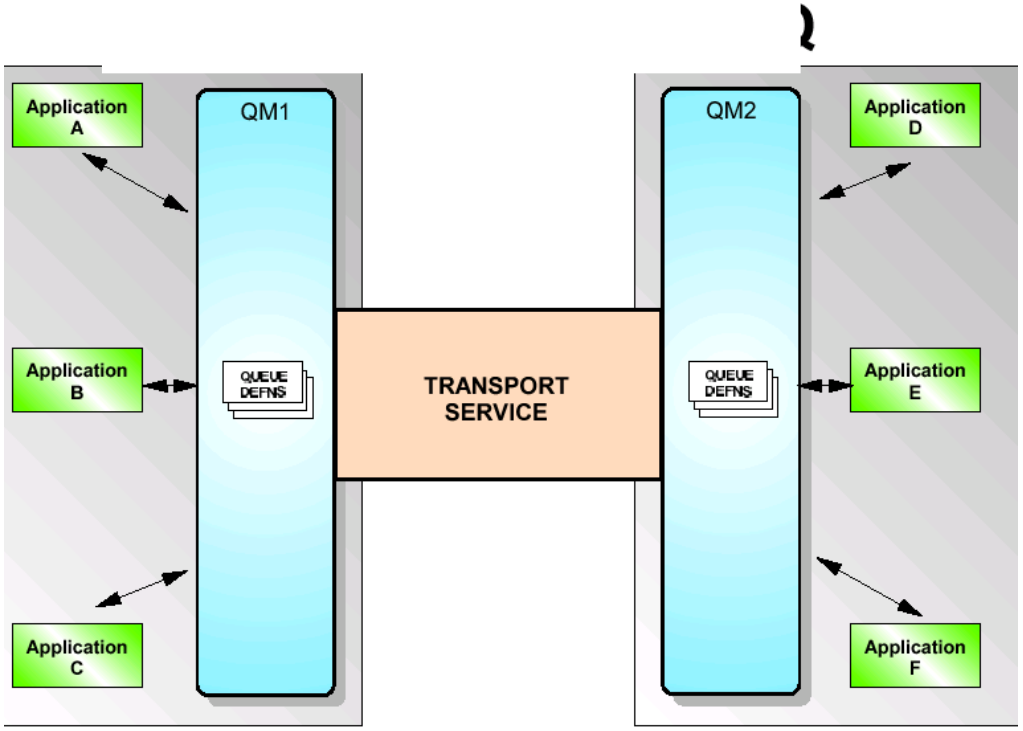
FSA EAI Architecture

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.

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

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.

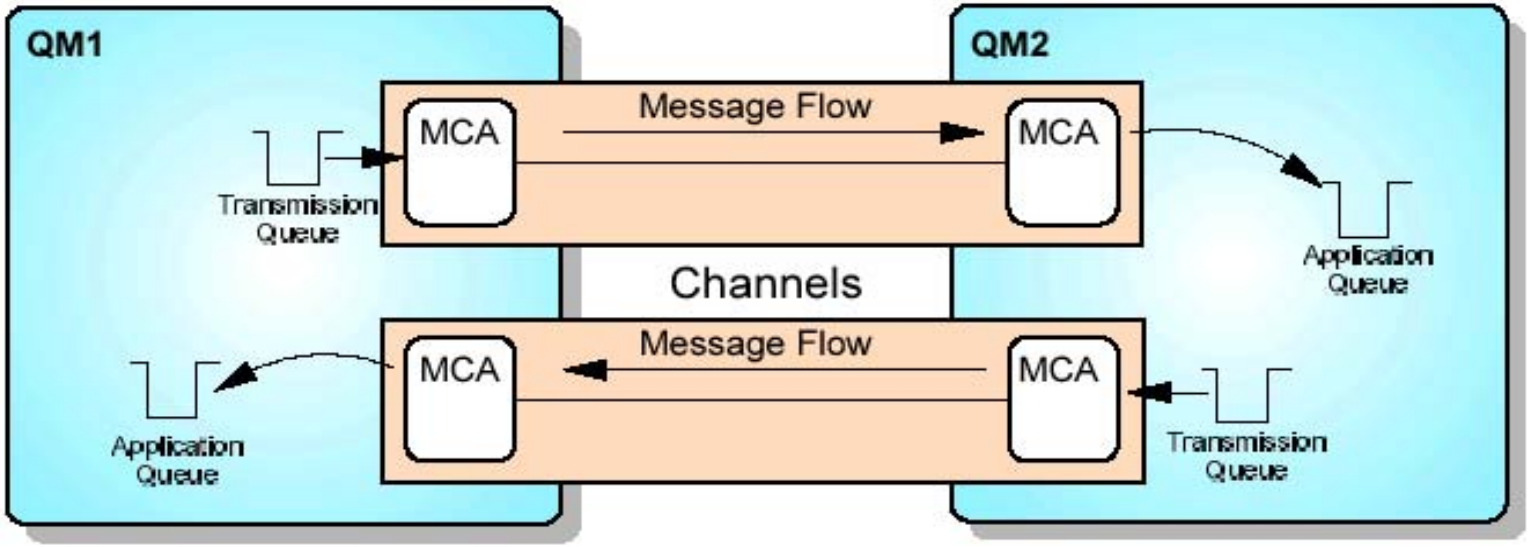
Any sequence of bytes
- Private to the sending and receiving programs
- Not meaningful to the Queue Manager

Message Types

- Persistent ... recoverable
- Non Persistent

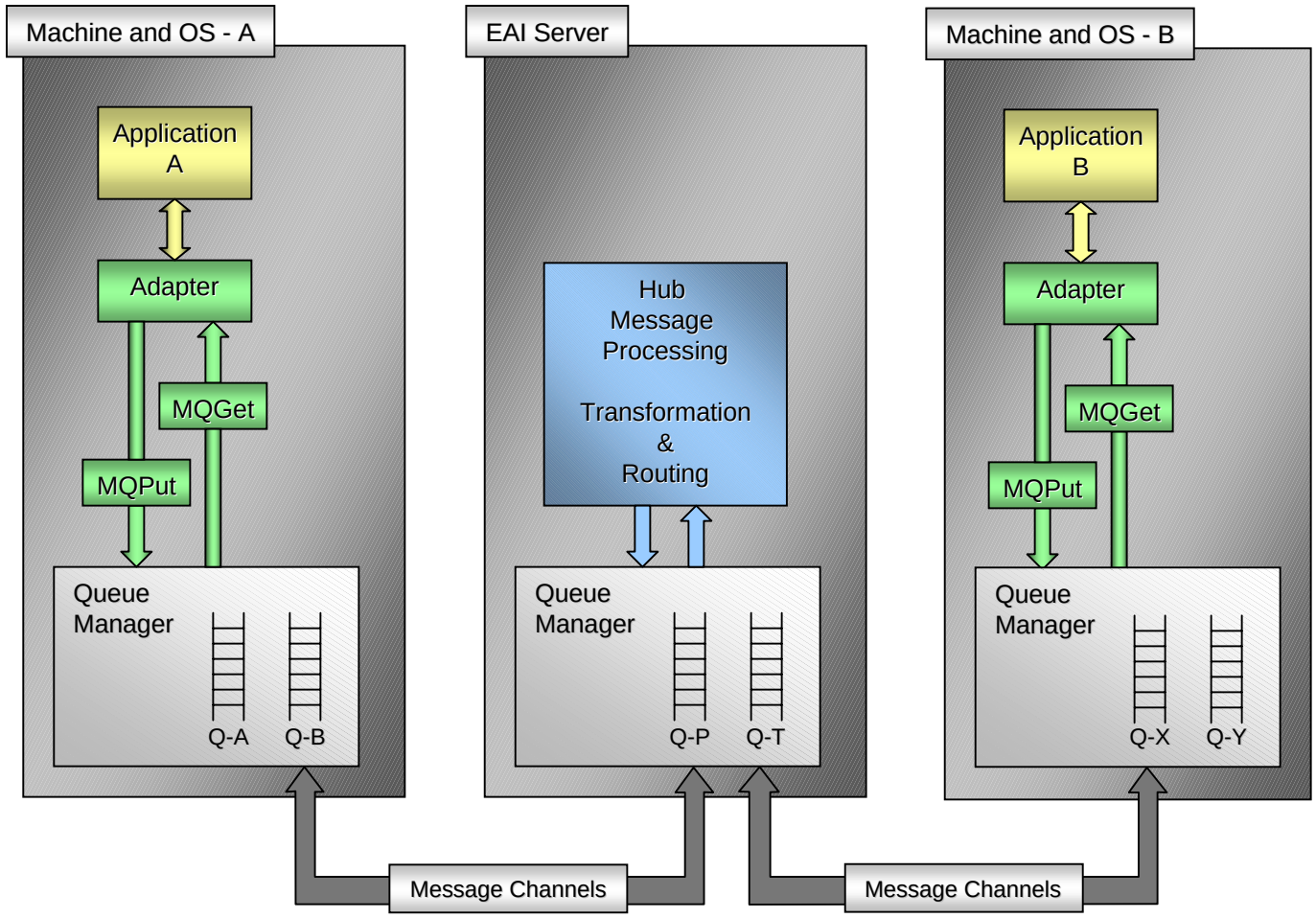
Up to 100MB message length

Channel Concepts

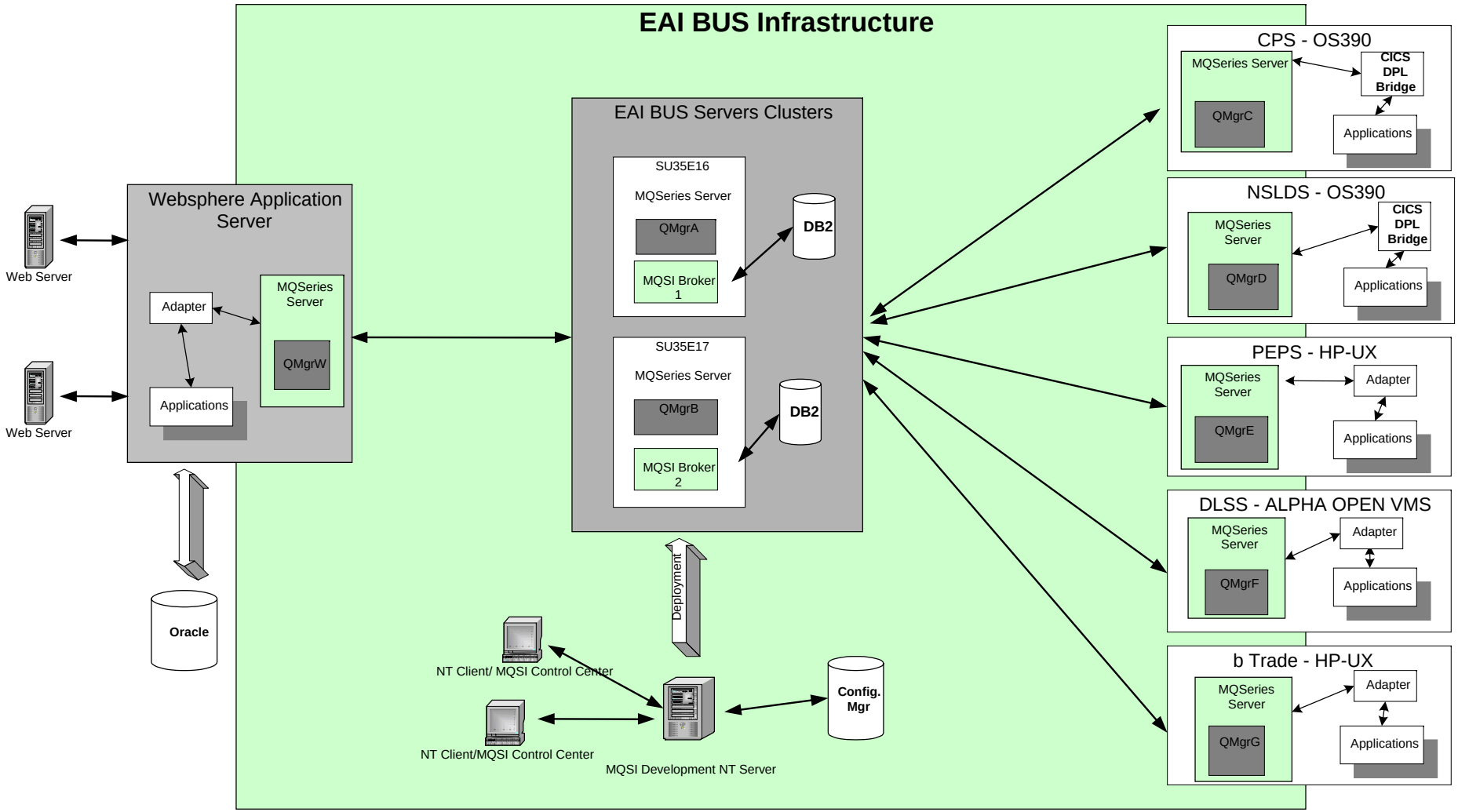


Channels are uni-directional
Channels provide for (application) session concentration
Two-way communication requires two channels

Queue Architecture: Hub and Spoke



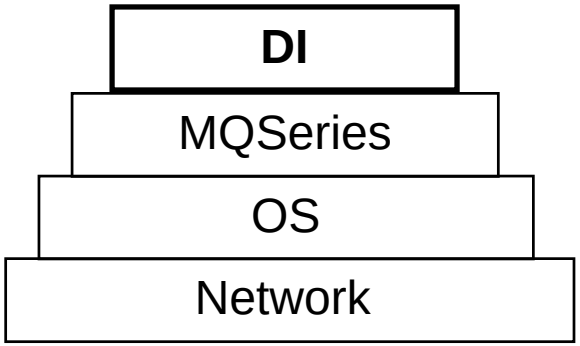
Development Environment - EAI



What is Data Integrator?

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

- Networks
- Operating Systems
- Databases
- Applications

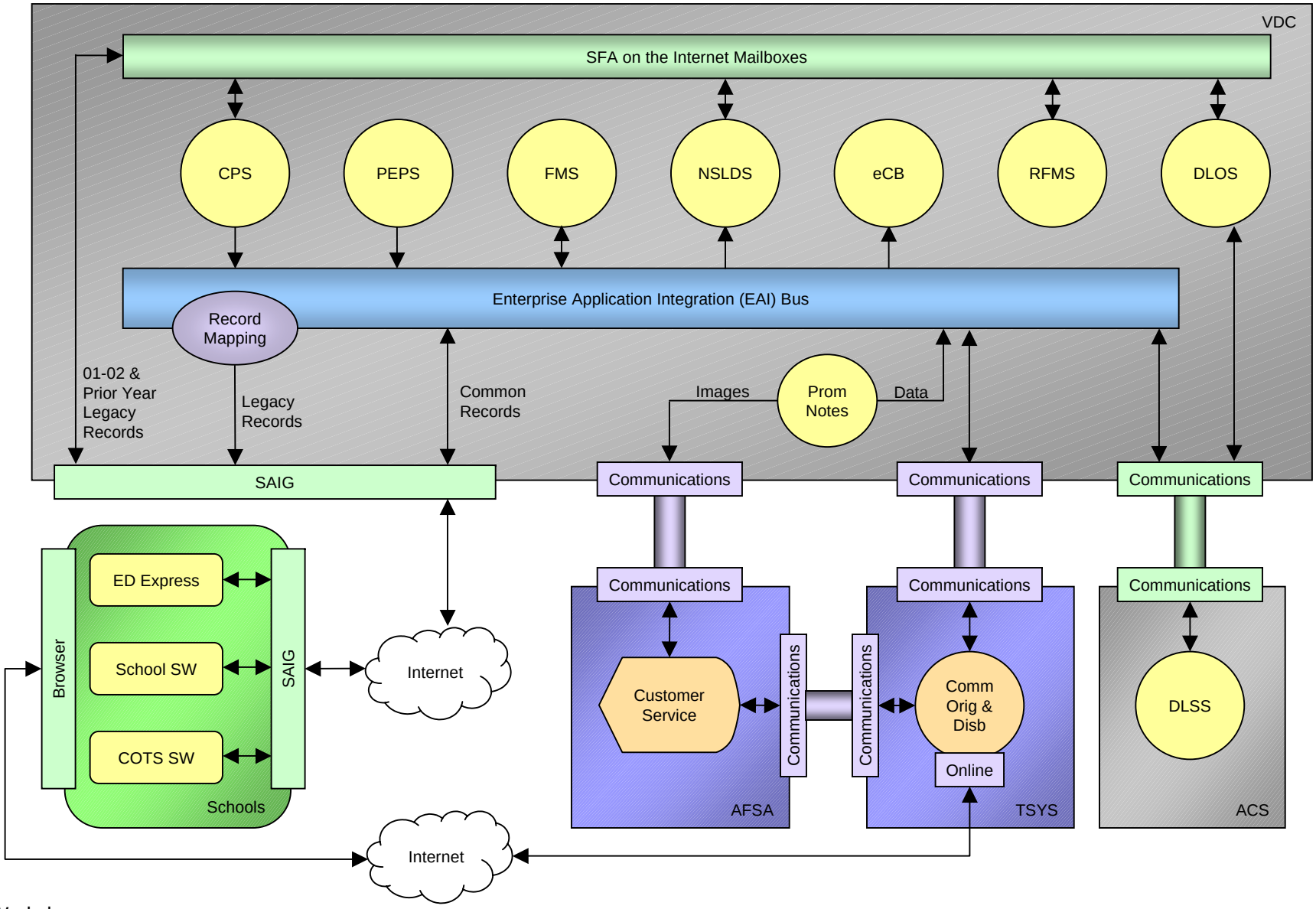


FEDERAL STUDENT AID
We Help Put America Through School

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.

COD Architecture



FEDERAL STUDENT AID

We Help Put America Through School

Common Record Transformation for COD

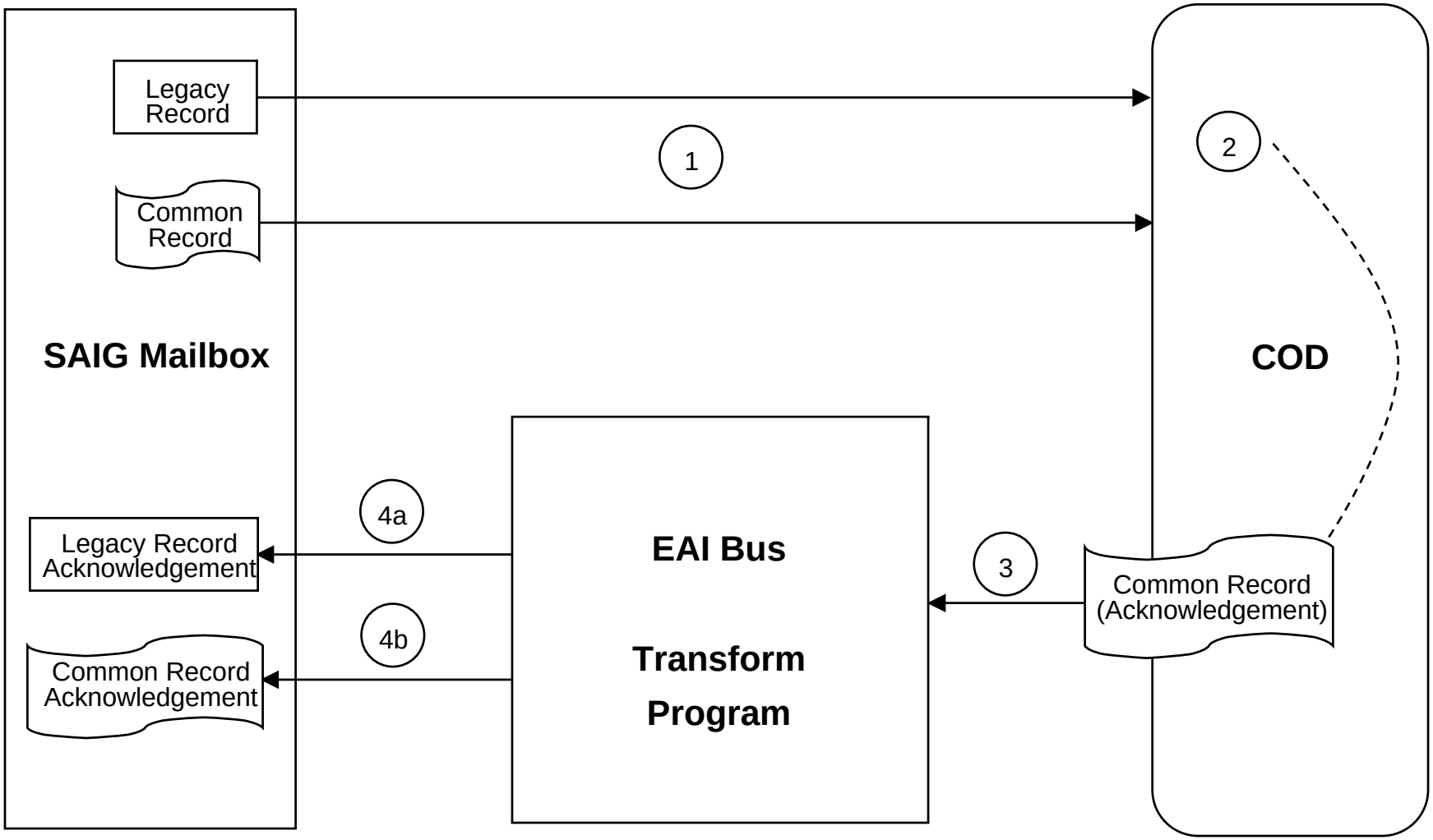
Theresa Pak

accenture

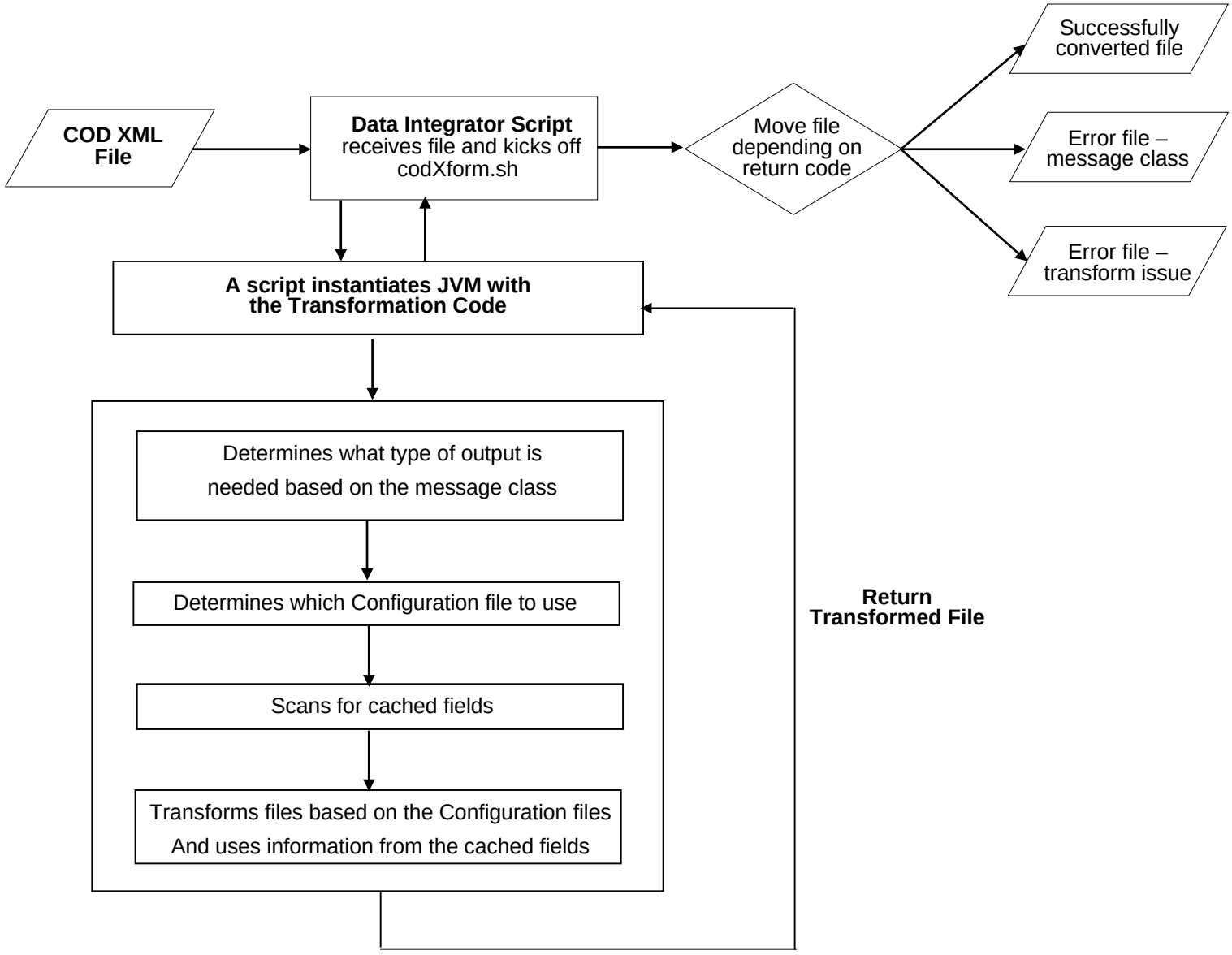
COD Transformation Context

- COD receives input files from schools.
  - School Types
    - Full Participant Schools (Common Record/XML)
    - Phase-In Participant Schools (Legacy Record)
- COD generates a Common Record format Acknowledgement for each input received
- A Response file will also be sent to the school when a Common Record input file is sent.
- The Acknowledgements must be transformed in order for the schools to receive them properly.
  - Common Record format
  - Legacy Record format
- COD System Generated Acknowledgements are also Transformed and sent to schools, however there are no associated input files from schools.

EAI Bus Transformation



EAI Bus Transformation



Example XML Input (Acknowledgement)

```
<?xml version="1.0" encoding="UTF-8"?>
<CommonRecord xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ……
        <DeliveryInfo>
                <SAIGHeadBatchId>12345678901234567890123456</SAIGHeadBatchId>
                ……
        </DeliveryInfo>
        <LegacyInfo>
                <HeadBatchId>#D3X99999CCYYMMDDHHMMSS</HeadBatchId>
                ……
        </LegacyInfo>
        <ReportingSchool EntityID="12345678">
                <AttendedSchool EntityID="23456789">
                        <Student SSN="123456789" BirthDate="1960-01-01" LastName="Tester">
                                <DLSubsidized>
                                        ……
                                </DLSubsidized>
                                <Response>
                                        ……
                                </Response>
                        </Student>
                </AttendedSchool>
                <Response>
                        ……
                </Response>
        </ReportingSchool>
        <Response>
                ……
        </Response>
</CommonRecord>
```

Example from a Configuration File

```
<?xml version="1.0"?>
<ResponseMapping name="DLSubUnsubOrigAck">
        <CachedFields>
                <CachedGroup name="DocumentCache">
                        <RefreshOn>CommonRecord</RefreshOn>
                        <CachedField refName="SAIGHeadBatchId" src="DeliveryInfo.SAIGHeadBatchId"/>
                        ……….
                </CachedGroup>
        </CachedFields>
        ……….
        <FlatFileLayout>
                <RowTemplate name="TWBatchHeader" repeating="no" eor="95">
                        <Field>
                                <Name>TIVWAN Batch Header Id Number</Name>
                                <Start>1</Start>
                                <End>5</End>
                                <Width>5</Width>
                                <Justify>none</Justify>
                                <Pad>none</Pad>
                                <Type>static</Type>
                                <Source>O*N05</Source>
                                <FormatCd/>
                        </Field>
                        ……….
                </RowTemplate>
                ……….
        </FlatFileLayout>
</ResponseMapping >
```

Example Legacy Format Output

Layout Type	Example lines in legacy files
SAIG Header	O*N05TG12345 ,CLS=DISF03OP,XXX,BAT=12345678901234567890123456,ABCD12
Direct Loan Header	DL HEADER 0095DISF03OP#D3X99999CCYYMMDDHHMMSSCCYYMMDDHHMMSS
Detail Record(s)	20010120#D3X99999CCYYMMDDHHMMSS999999999S03X99999999A AE Y
Direct Loan Trailer	DL TRAILER00950001050010000005000000
SAIG Trailer	O*N95TG12345 ,CLS=DISF03OP,XXX,BAT=12345678901234567890123456,ABCD1

R2.0 Message Classes

AY	Production MsgCls	Testing MsgCls	Batch Type	Data Description
02-03	DISF03OP	DTSF03OP	#D	Direct Loan Subsidized/Unsubsidized Origination Acknowledgement
	DIPF03OP	DTPF03OP	PF	Direct Loan PLUS Origination Acknowledgement
	DIOC03OP	DTOC03OP	#E	Direct Loan Change File Acknowledgement
	DIPC03OP	DTPC03OP	#C	Direct Loan PLUS Credit Decision Acknowledgement
	DIPA03OP	DTPA03OP	#A/#D/PF	Direct Loan Promissory Note Acknowledgement -MPN/PLUS
	DIOD03OP	DTOD03OP	#H	Direct Loan Subsidized/Unsubsidized Disbursement Acknowledgement
	DIOD03OP	DTOD03OP	#B	Direct Loan Booking Notification
	DIOD03OP	DTOD03OP	SP	Direct Loan Payment to Servicing
	PGOA03OP	PTOA03OP	#O	Pell Origination Acknowledgement
	PGOP03OP	PTOW03OP	#P	Web-Generated Pell Origination Acknowledgement
	PGDW03OP	PTDW03OP	#W	Web-Generated Pell Disbursement Acknowledgement
	PGDA03OP	PTDA03OP	#D	Pell Disbursement Acknowledgement
	PGDA03OP	PTDA03OP	#G	Pell System-Generated Disbursements Acknowledgement
03-04	DISF04OP	DTSF04OP	#D	Direct Loan Subsidized/Unsubsidized Origination Acknowledgement
	DIPF04OP	DTPF04OP	PF	Direct Loan PLUS Origination Acknowledgement
	DIOC04OP	DTOC04OP	#E	Direct Loan Origination Change
	DIPC04OP	DTPC04OP	#C	Direct Loan PLUS Credit Decision Acknowledgement
	DIPA04OP	DTPA04OP	#A/#D/PF	Direct Loan Promissory Note MPN Acknowledgement
	DIOD04OP	DTOD04OP	#H	Direct Loan Disbursement Acknowledgement
	DIOD04OP	DTOD04OP	#B	Direct Loan Booking Notification
	DIOD04OP	DTOD04OP	SP	Direct Loan Payment to Servicing
	PGOA04OP	PTOA04OP	#O	Pell Origination Acknowledgement
	PGDA04OP	PTDA04OP	#D	Pell Disbursement Acknowledgement
	PGDA04OP	PTDA04OP	#G	System-Generated Pell Disbursements
	PGOP04OP	PTOW04OP	#P	Web-Generated Pell Origination Acknowledgement
	PGDW04OP	PTDW04OP	#W	Web-Generated Pell Disbursement Acknowledgement

FEDERAL STUDENT AID

We Help Put America Through School

Transactional Interfaces
LOWeb and FOTW

Jeff Goldhirsch

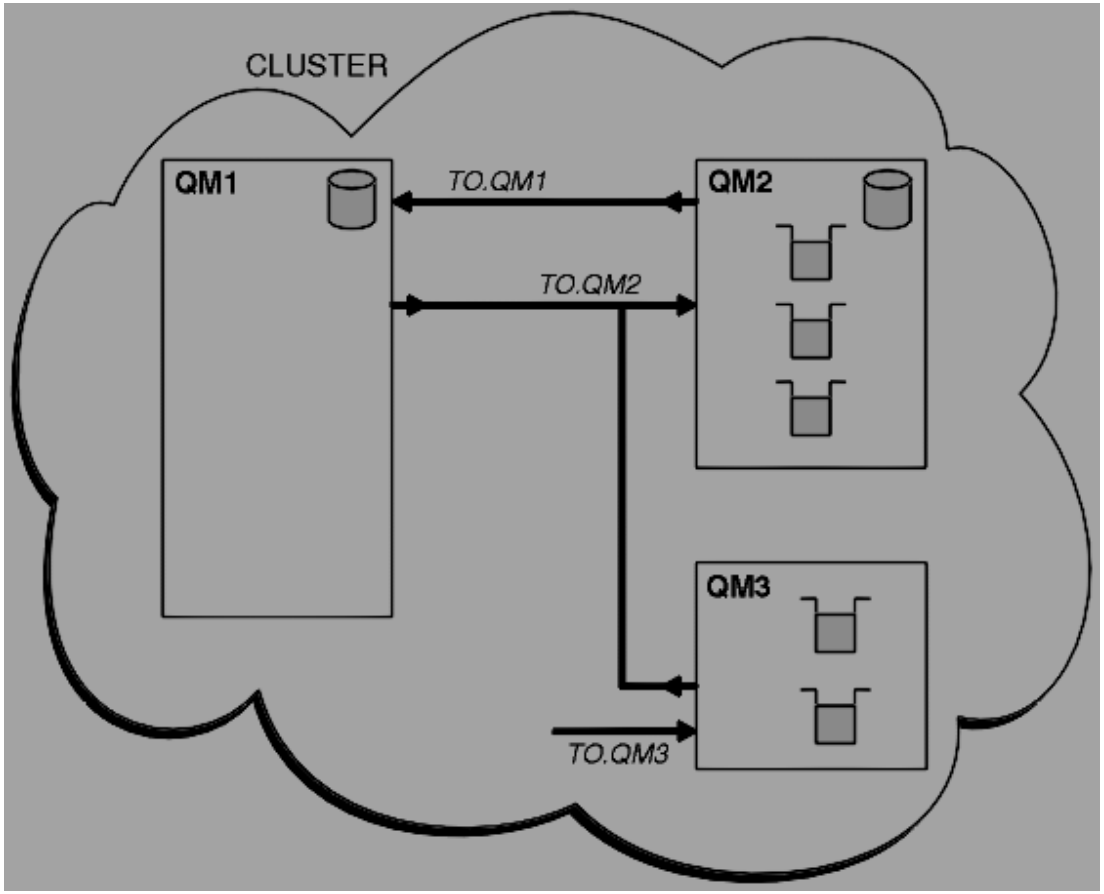
accenture

WebSphere MQ: Clusters

- A Cluster is a network of queue managers that are logically associated.
- Clustered queue managers advertise clustered objects to repository queue managers.
- When compared to distributed queuing, clustering will reduce system administration.
- Clustered Queue Managers can increase availability and allow for workload balancing.
- Queue managers in a cluster may be physically remote.

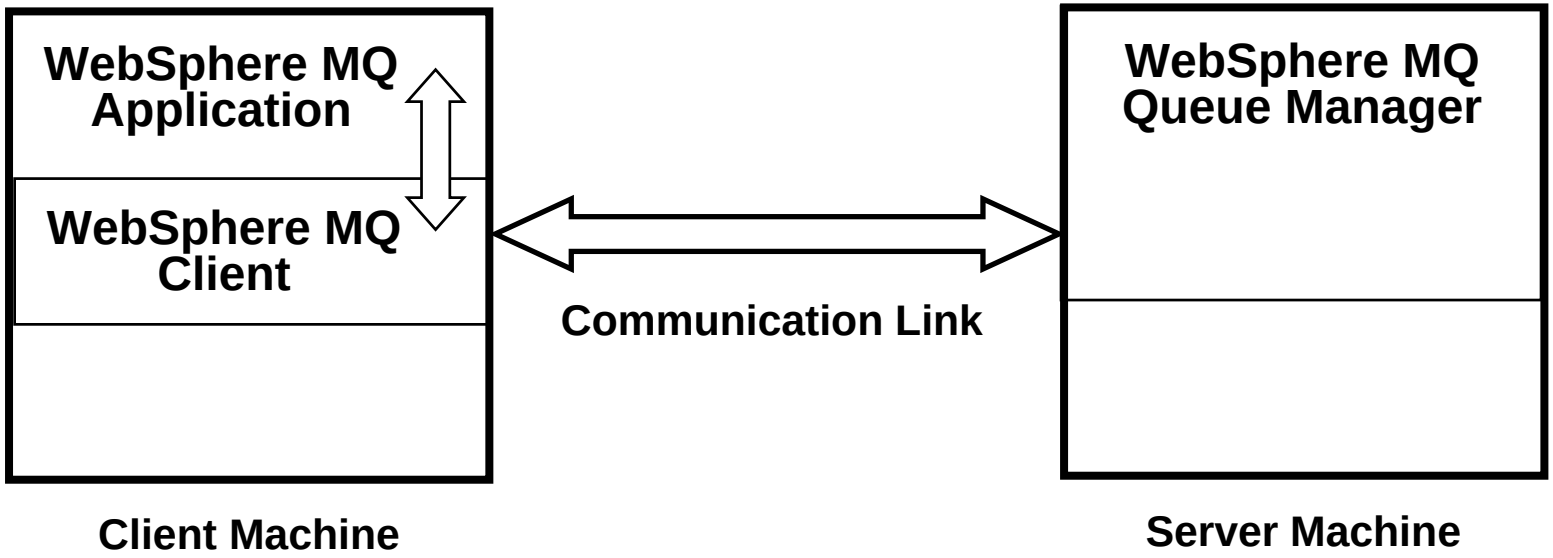
Some Operating Systems include a concept of Clusters or Clustering. Websphere MQ Clustering is totally separate and distinct from any such constructs.

WebSphere MQ: Cluster Objects



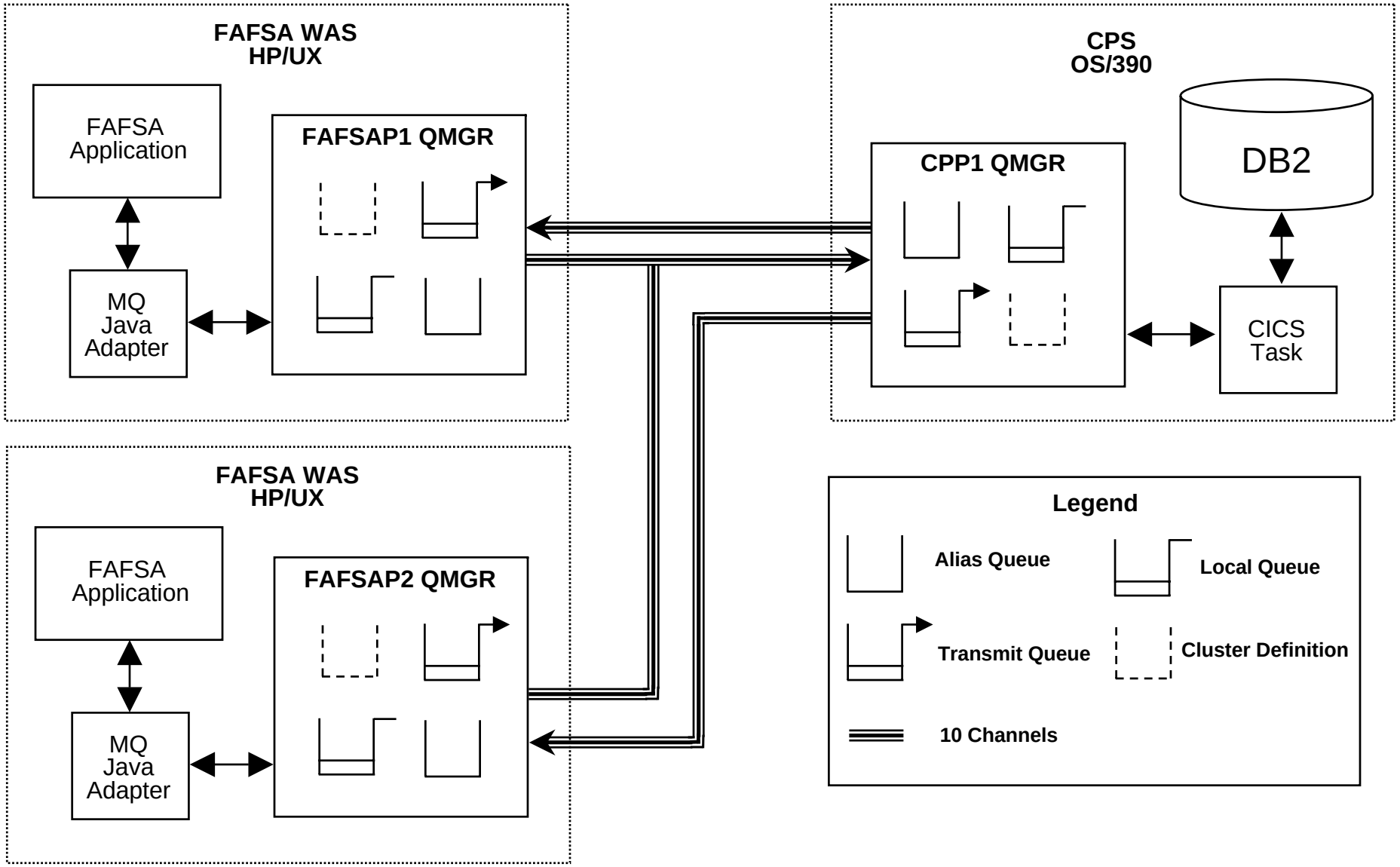
- Cluster
- Cluster Queue Manager
- Cluster Queue
- Repository
- Cluster-Receiver Channel
- Cluster-Sender Channel
- Cluster Transmission Queue

WebSphere MQ: Client Connections



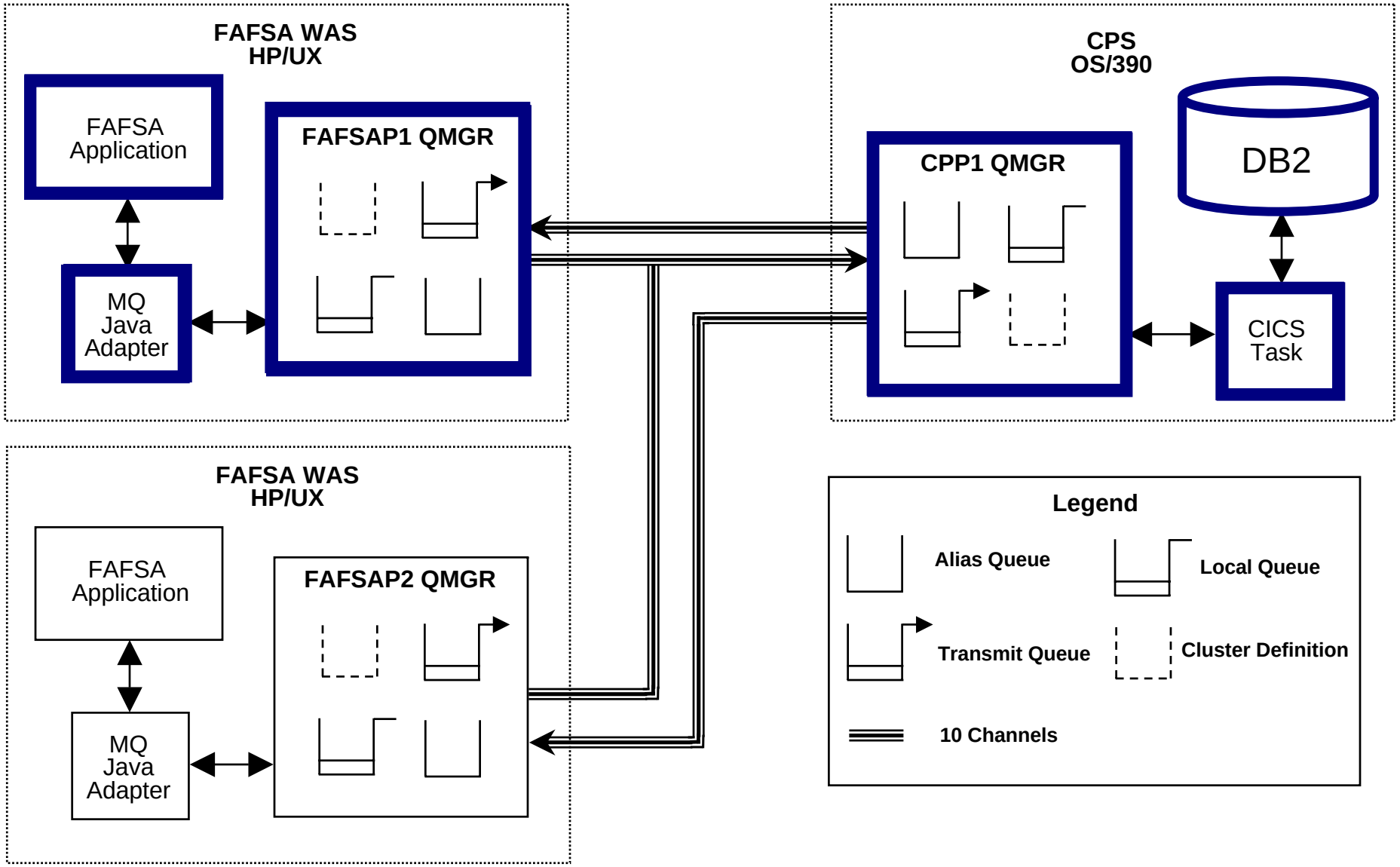
- Part of the WebSphere MQ Product that can be installed outside of base product
- Allows WebSphere MQ applications to connect to a Queue Manager
- MQ Clients connect to Queue Managers using Bi-Directional channels for MQI Calls and for Responses (other channels are Uni-Directional)
- EAI Custom Code manages Client Connections to Queue Managers

FOTW: Transaction Flow



FEDERAL STUDENT AID
We Help Put America Through School

FOTW: Transaction Flow



FEDERAL STUDENT AID
We Help Put America Through School

FOTW: Transaction Flow

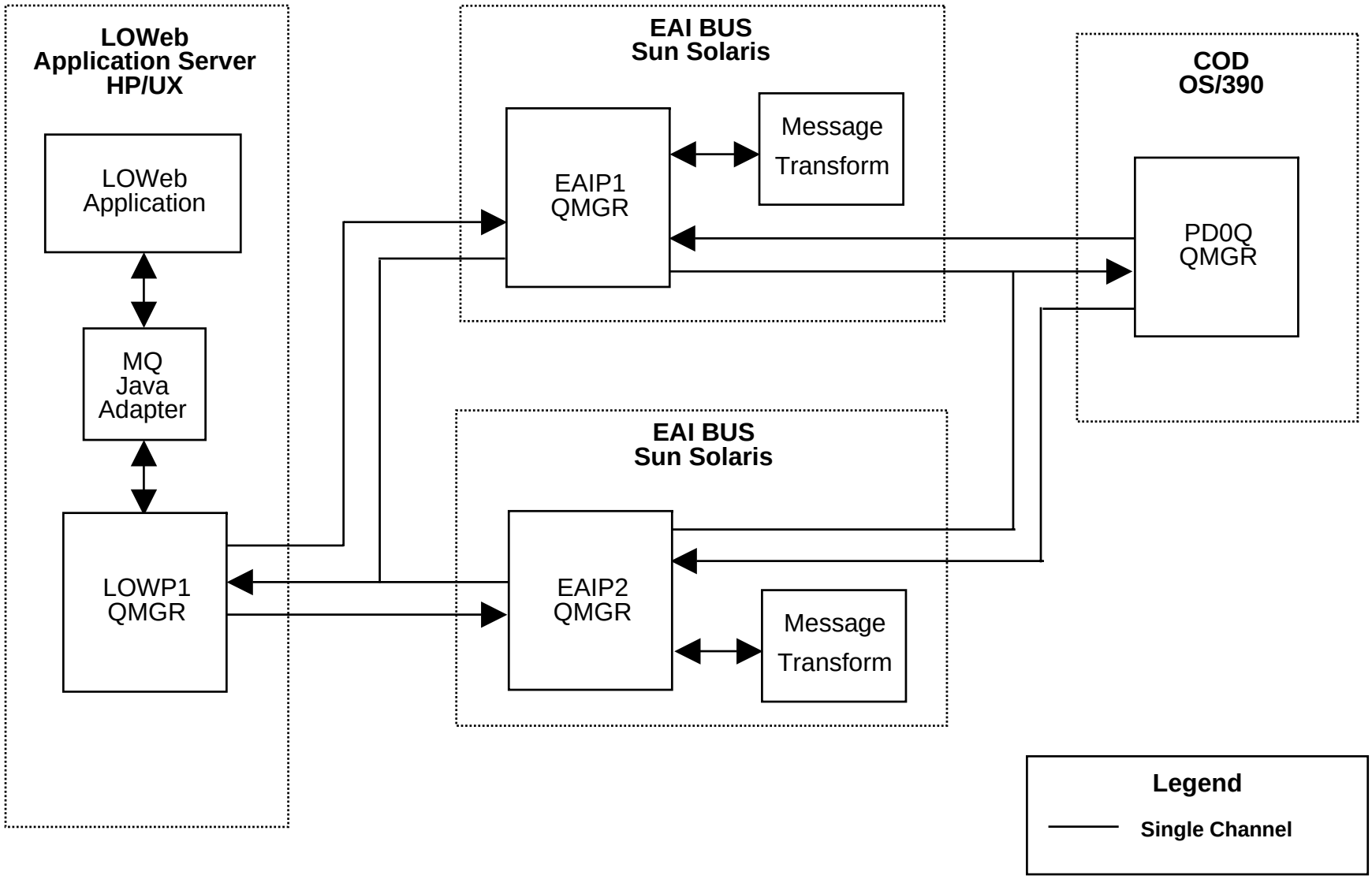
1. A FOTW user generates a request with data for CPS. This FOTW Application calls the MQ Java Adapter.
2. The MQ Java Adapter, first attempts to find an available Client Connection to a Queue Manager. The Adapter also builds the MQ Message and puts the message to the appropriate Queue.
3. In this case, the appropriate queue is located on CPS, so the message is put on a Transmit Queue. Once on the transmit queue, the first available channel will service the message and send it to CPS.
4. The message will then reside on the CPS Queue Manager.
5. A CICS transaction will pull the message off the CPS Queue Manager, process the backend edits, and perform any database calls.
6. The database, based on transaction type, will either execute an insert statement, or a lookup statement.
7. The CICS transaction will will then build a response MQ Message based on the results that are returned from the Database. The Response message is then put to the appropriate queue.
8. In this case, the appropriate queue is located on one of the FAFSA Queue Managers, so the message is put on a transmit queue. Once on the transmit queue, the first available channel will service the message, and send it back to FAFSA
9. The message will then reside on the FAFSA Queue Manager.
10. The MQ Java Adapter returns the response data to the FOTW Application.
11. The FOTW Application returns the reply data to the user.

LOWeb: Transactions

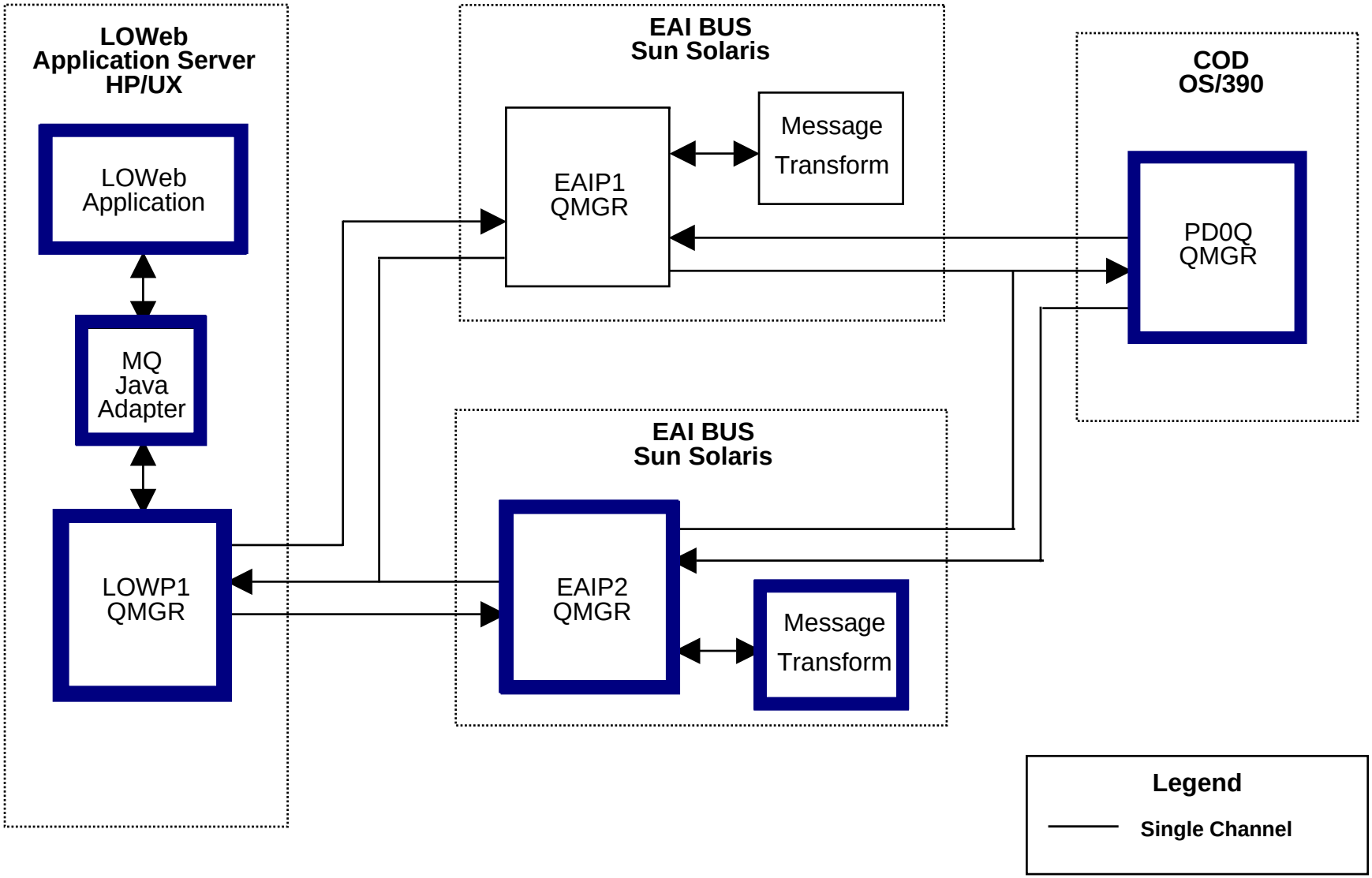
- **Reply Request Interfaces**
  - **Borrower Validation** – LOWeb transaction to verify a borrower has an award on file at COD
  - **Credit Check** – LOWeb transaction that allows COD to perform a Credit Check on a DL PLUS Loan Borrower/Endorser

- **Datagram Interfaces**
  - **Endorser** – Meta-Data about Endorsers are sent to COD
  - **P-Note** – Meta-Data from Scanned Promissory Notes and Electronic Promissory Notes are sent to COD

LOWeb: Transaction Flow



LOWeb: Transaction Flow



FEDERAL STUDENT AID
We Help Put America Through School

LOWeb: Transaction Flow

1. A LOWeb user generates a request with data for COD. This LOWeb Application calls the MQ Java Adapter.
2. The Adapter builds the MQ Message and puts the message to the appropriate Queue.
3. In this case, the appropriate queue is located on the EAI BUS, so the message is put on a Transmit Queue. Once on the transmit queue, the first available channel will service the message and send it to the EAI Bus.
4. The message will then reside on the EAI BUS Queue Manager.
5. The Message is then transformed to match the layout the COD application is expecting.
6. The message is put on a transmit queue for COD. Once on the transmit queue, the channel will service the message, and send it back to COD.
7. COD will process the message and send a reply message back to the EAI Bus.
8. The message will then reside on the EAI Bus Queue Manager.
9. The Message is then transformed to match the layout the LOWeb application is expecting.
10. The message is put on a transmit queue for the LOWeb Queue Manager. Once on the transmit queue, the channel will service the message, and send it back to the LOWeb Queue Manager.
11. The message will then reside on the LOWeb Queue Manager
12. The MQ Java Adapter returns the response data to the LOWeb Application.
13. The LOWeb Application returns the reply data to the user.

FEDERAL STUDENT AID

We Help Put America Through School

EAI Interface Performance

Eric Suzuki

accenture

FEDERAL STUDENT AID
We Help Put America Through School

EAI Performance Test Results

Interface	Objective	Test	Results
FMS - COD	Measure the rate at which the EAI Bus handles FMS - COD Transactions	370 transactions were monitored from their availability on the Bus until they were committed to the FMS database.	Transactions were available to FMS in 166 seconds. (2 messages per second)
		278 transactions were monitored from the time they are available in the FMS database until they land on a queue destined for COD	Transactions were available to be sent to COD in 84 seconds. (3 messages per second)
eCB - FMS	Measure the rate at which the interface processes eCB UTCL file	A UTCL file with 1945 detailed records was monitored from it's availability on the Bus until it was committed to the FMS database.	File was available to FMS in 22 minutes. (1.5 records per second)
		A UTCL file with 325 detailed records was monitored from it's availability on the Bus until it was committed to the FMS database.	File was available to FMS in 3 minutes (1.8 records per second)
PEPS - EAI	Measure time to transfer School File (127 MB) from PEPS to EAI Bus	Transfer file using FTP	Transfer took 2 hrs, 21 min, 28 sec
		Transfer file using EAI	Transfer took 6 min, 10 sec

FEDERAL STUDENT AID
We Help Put America Through School

EAI Performance Test Results

Interface	Objective	Test	Results
DLSS - CMDM	Measure the transfer rate of very large monthly demographic file from DLSS to CMDM	Transfer the file using FTP	Transfer took 13 hours
		Transfer the file using EAI	Transfer took 2.5 hours
P-Note Data from LOWEB	Measure the response time for a request for P-Note Data	Several thousand requests for P-Note data were made from the WebSphere server, through the EAI Bus Server, to the LOWEB Server.	Average response time was 1.2 sec
SAIG - COD	Assess SAIG ability to process 6,000 sends within one hour	Simulated 6,000 sends in a very short period of time	All transactions were processed within 30 minutes.
SAIG[1]	Determine if the configuration with one FTP server will handle the increase in file sizes expected over FY03.	Performance test areas include a wide range of items such as user response times, throughput, CPU and memory utilization, and number of concurrent users etc.	It is our recommendation that the proposed SAIG production environment configuration will sustain the load anticipated for SAIG peak from February 2003 through April 2003.

1 - SAIG Performance Test Report_v3.0.doc (CSC, EAI, SAIG)

COD - XML File Conversion Performance

Test files of 20MB, 160MB and 500MB were processed to obtain an understanding of the performance characteristics of the processor as it operated on the BUS (Sun E3500, 4 4OO MHz processors, 4 GB RAM). Testing showed that there was no performance degradation of the processor as the file size grew.

The processor was able to convert a 20 MB file (4300-6500 transactions) in 60-90 seconds, (depending on file type). Results of the tests have been compared to average and max daily volume statistics to obtain an estimate of total processing time.

File Type	Average # of daily transactions	Average day processing time	Maximum # of daily transactions	Maximum day processing time
Output Common Record	200MB	10 min.	600MB	60 min
Legacy Pell Origination Ack.	30,000	7.5 min.	125,000	31.5 min.
Legacy Pell Disbursement. Ack.	45,000	10 min	175,000	38 min.
Legacy DL Origination Ack.	10,000	3 min.	50,000	15 min.
Legacy DL Disbursement Ack.	14,000	4 min.	115,000	33 min.
System Generated Responses	40,000	8 min.	175,000	32 min.

COD - File Transfer Performance

The tests were conducted from the host system of the trading partner, over MQ/DI, through the BUS. An average and a maximum file size is used, to obtain data on operation under average and heavy loads.

File Type	Frequency	Avg. File Size	Average Time	Max. File Size	Max Time
CPS Abbreviated Applicant File	1 time/day	16MB	10 sec.	50MB	30 sec.
CPS Pell Institution Universe	1 time/day	2.75MB	<5 sec.	2.75MB	<5 sec.
CPS Recipient Record	4 times/year	275MB	3 min.	1.165GB	12 min.
DLSS Daily Batch - Submit	1 time/day	7MB	<5 sec.	50MB	30 sec.
DLSS Daily Batch - Response	1 time/day	100MB	1 min.	200MB	2 min.
DLSS Disbursement Confirmation	1 time/week	24MB	15 sec.	50MB	30 sec.
NSLDS Pell Recipient Info	1 time/day	13.2MB	10 sec.	34MB	15 sec.
NSLDS Pell Recipient Info Response	1 time/day	0.23MB	5 sec.	0.62MB	5 sec.
PEPS Full Refresh Feed	As needed	70MB	40 sec.	100MB	1 min.
PEPS Processed Delta Feed	1 time/day	700KB	<5 sec.	100MB	1 min.
SAIG School Destination Info	1 time/day	10MB	<5 sec.	20MB	10 sec.
SAIG Reports	TBD	111MB	1 min.	2000MB	20 min.

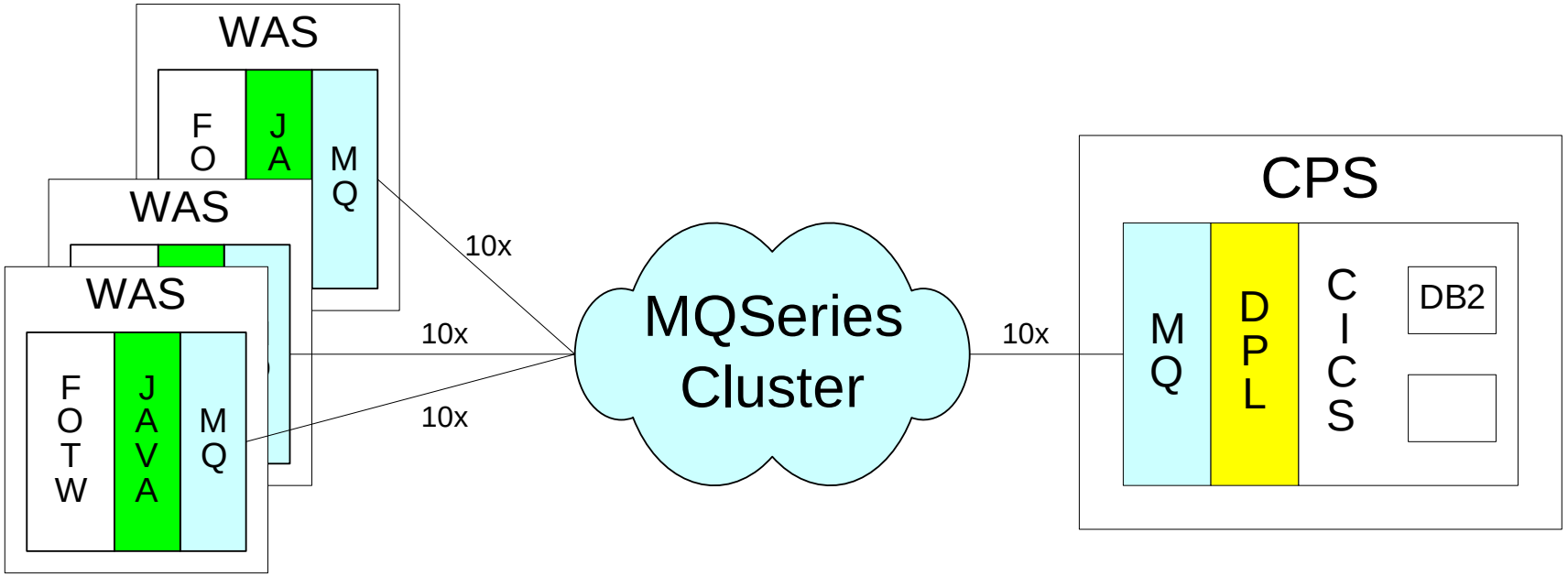
FAFSA 7.0 Performance Testing

Background

- FAFSA 7.0 was the first release of FAFSA where EAI (MQSeries) was used instead of Shadow Direct for FOTW interfaces to CPS.
- EAI, CSC, and Pearson worked together to exercise the backend and identify performance problems
- Goals:
  - 125 Transactions/Second
  - 99.9% Uptime
  - Low impact to application servers

FAFSA 7.0 Performance Testing

Architecture



KEY

JAVA	EAI Java Adapter (Interface to MQSeries)
DPL	IBM Adapter between MQSeries and CICS

FAFSA 7.0 Performance Testing

ITA WAS Driven Testing

(Loadrunner - Maximum of 3k users –> 15-20 transactions/second)

- Stress front end application (ITA)
- Validate no negative EAI effects (EAI, CSC, ITA)

  Issues:

  - Problems with EAI code related to AMI
    - Resulted in code rewrite, replacing AMI with MQI (AMI will no longer be used within FSA applications)

WAS Driven Testing was insufficient to validate CPS ability to handle load

EAI Driven Testing

(EAI created a custom testing stub able to generate all transaction types in any combination up to several hundred/second)

- Stress back end interfaces and application (EAI, CSC, Pearson)

  Issues:

  - MQ / DPL Bridge Issues (40 txns./sec. @ 100% CPU)
    - Patches needed for high volume to address high CPU utilization and stability issues (>125 txns./sec @25-30% CPU)
  - CICS
    - IBM recommended extensive configuration and tuning changes
  - DB2
    - IBM recommended extensive configuration and tuning changes

FAFSA 7.0 Performance Testing

Results

- FAFSA Benefits
  - Testing resulted in complete application being able to support more than the targeted 125 transactions per second.
  - Uptime increased significantly over prior FAFSA application years
- Overall FSA Benefit
  - The highly optimized generic EAI interfaces developed for FAFSA are now in use by other applications (e.g. LOWeb)

FEDERAL STUDENT AID

We Help Put America Through School

Appendix

accenture

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