

**Title:**            **Migrating HP e3000  
Applications**



**Presentation: 25**

*Authors:*        Matt Street & Donna Faudree

*Company:*      Orion Group Software Engineers  
5770 Nimtz Parkway  
South Bend, IN 46628

*Phone:*         (574) 233-3401

*E-mail:*         mstreet@ogse.com  
dfaudree@ogse.com



# What We'll Cover Today

- Introduction
- EasyStep Approach
- N-Tier Architecture Design
- Data Access Approach
- Transactionalizing Entity Data
- Case Study
- Questions



# Migration vs. Re-engineering

- Migration – the translation and rehosting of source code and data to run on Unix or other platforms
- Re-engineering – “modernizing” of applications using current technology, including the client-server model, graphical user interface (GUI), web-based, and relational databases



# Migration vs. Re-engineering

## Migration:

- COBOL reports and batch processes

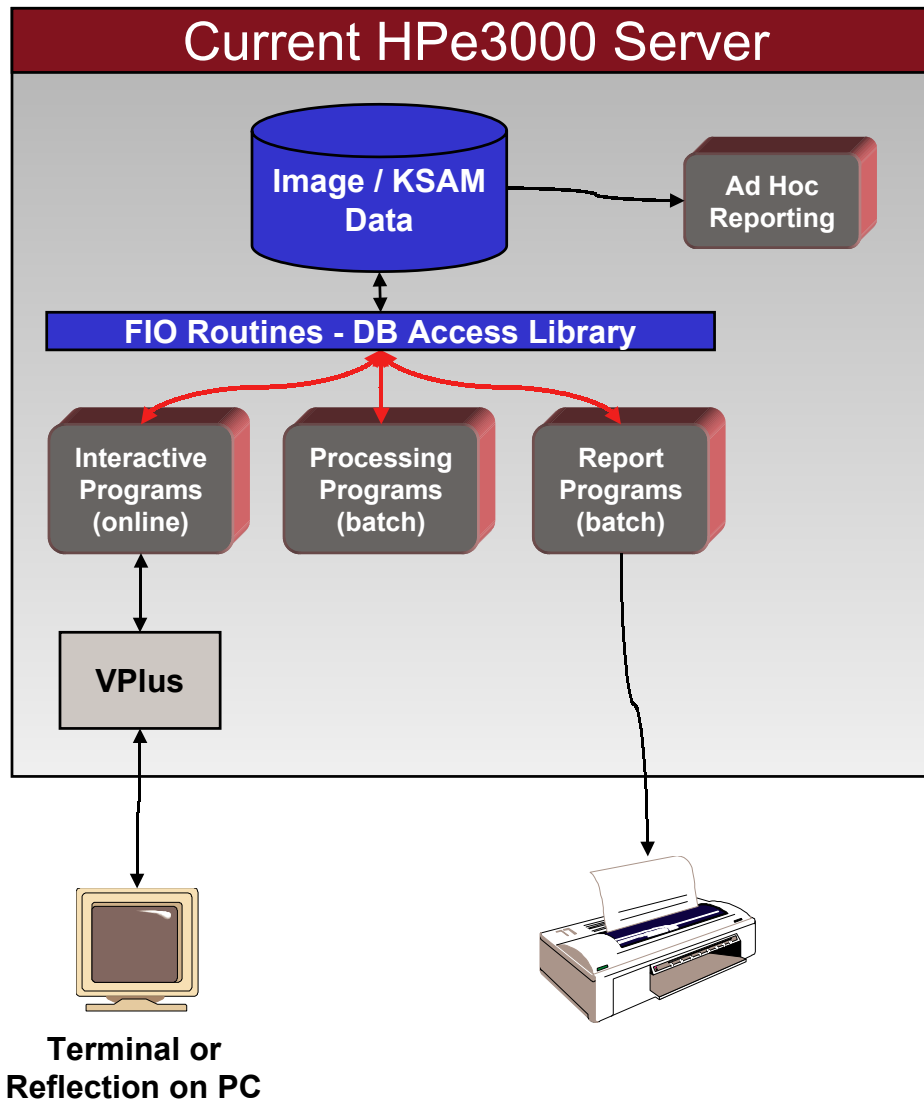
## Re-engineering:

- VPLUS screens/applications
- Image Databases



# EasyStep Approach – Step 1

## Current Environment



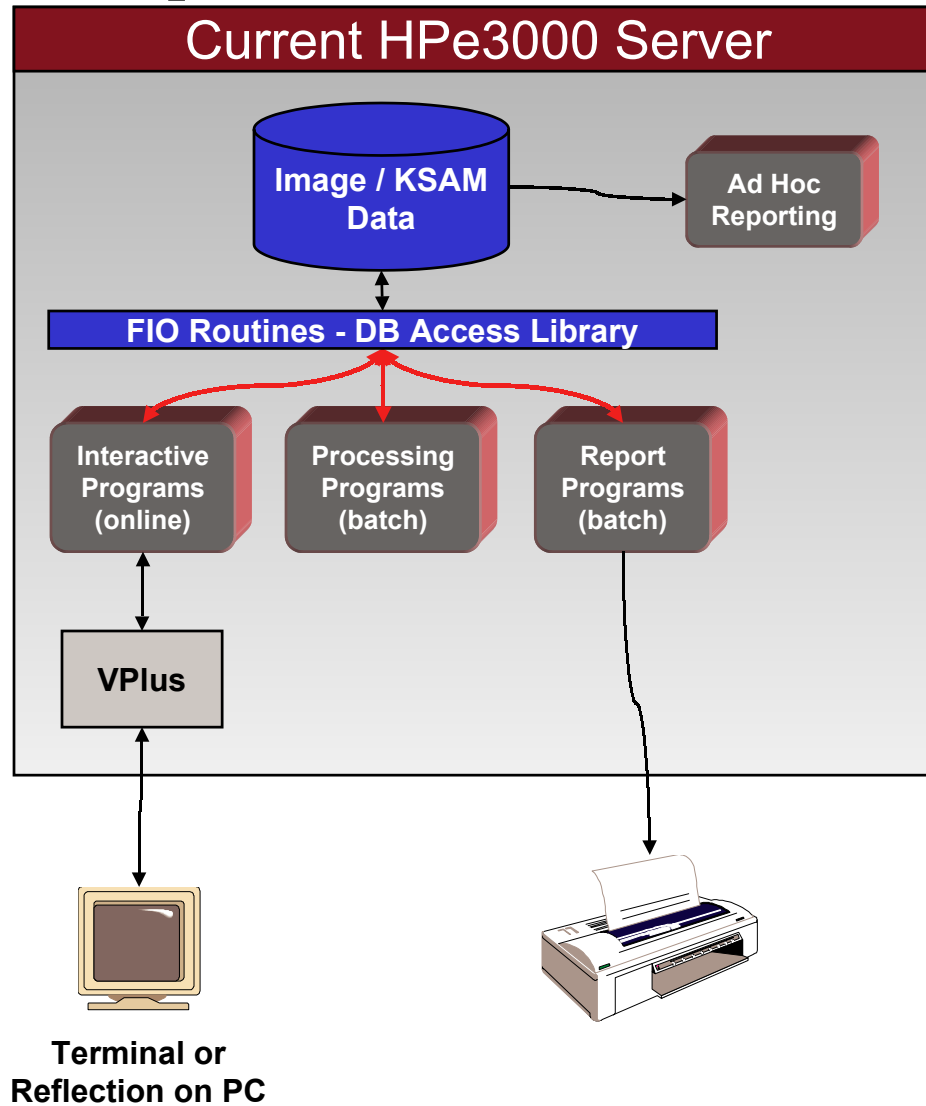
### May Consists of:

- Interactive programs
- Background processes
- Report programs
- Image databases
- KSAM files
- VPLUS screens
- COBOL programs.



# EasyStep Approach – Step 2

## Develop Relational Model



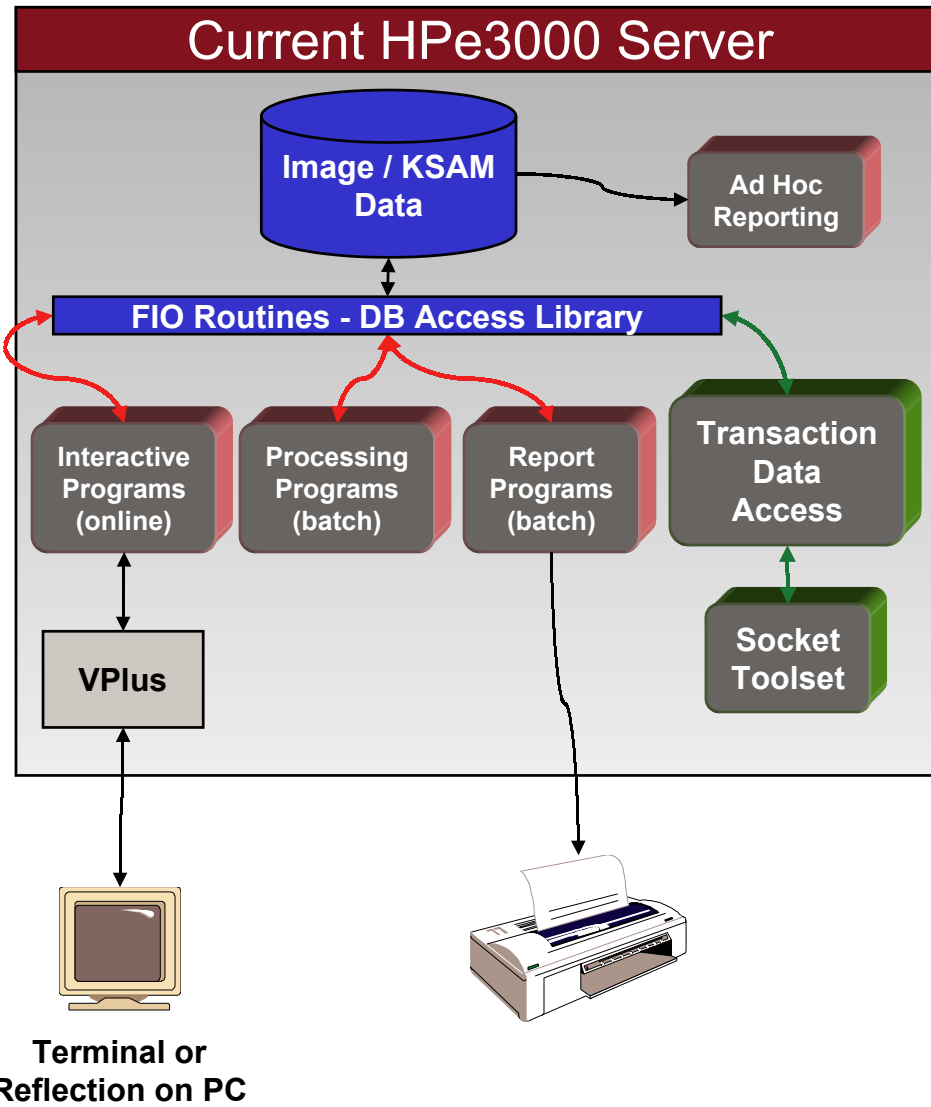
Relational Model (Schema)

This phase involves mapping the data structures on the HPe3000 into a relational database model. For the most part, this should be straight forward for well designed Image databases. Care should be taken to determine unique table keys, proper usage of data types, and to construct efficient relationships between tables. All data which the application uses should be modeled at this time. It is not necessary to convert the data at this time, or choose the relational database / server platform the data will reside in.



# EasyStep Approach – Step 3

## Transactionalize HP3000 Data



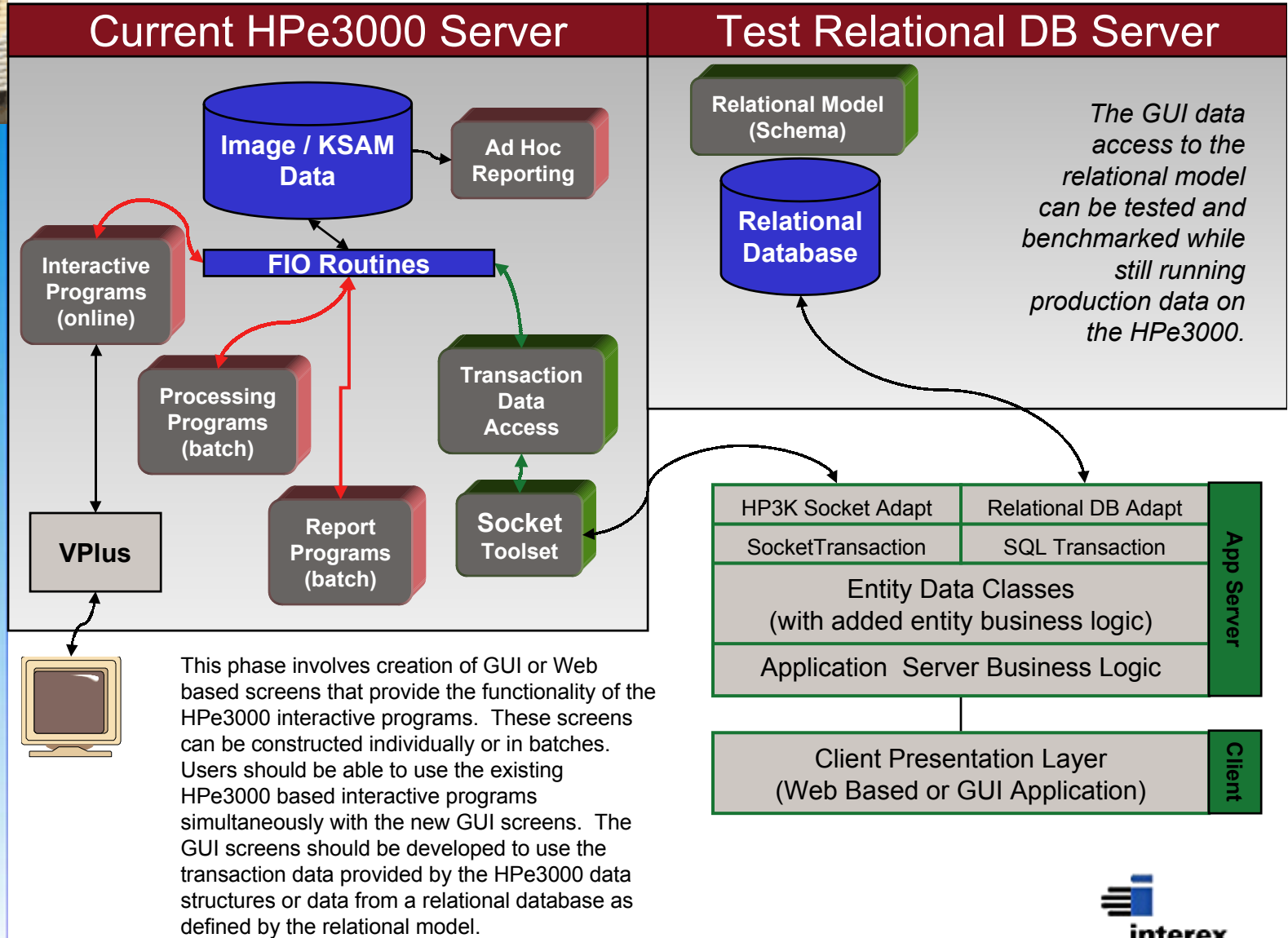
Relational Model (Schema)

This phase involves creation of Transactions to access data on the HPe3000. These transactions will map to the entities (tables) defined in the relational model. Future GUI development and data migration (conversion) programs will utilize the data provided by these transactions.



# EasyStep Approach – Step 4

## GUI Development

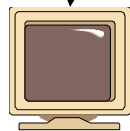
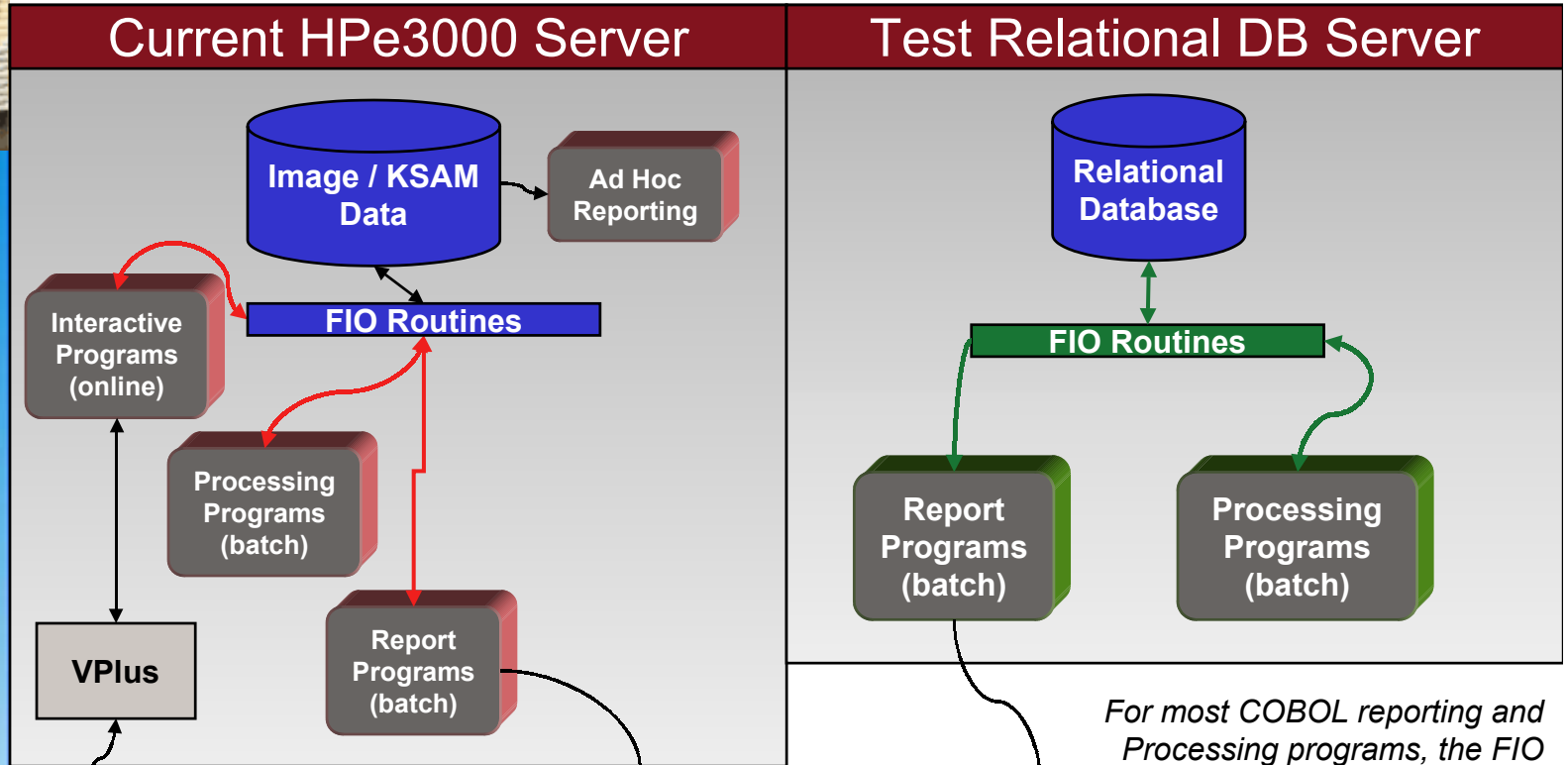




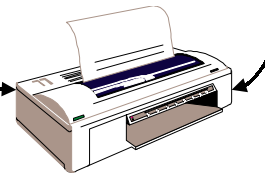


# EasyStep Approach – Step 5

## Batch Program Development



This phase involves the migration of batch programs which provide the reporting, posting or background transaction processing functions of the system. This phase can be done in conjunction with or separate from GUI development.

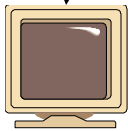
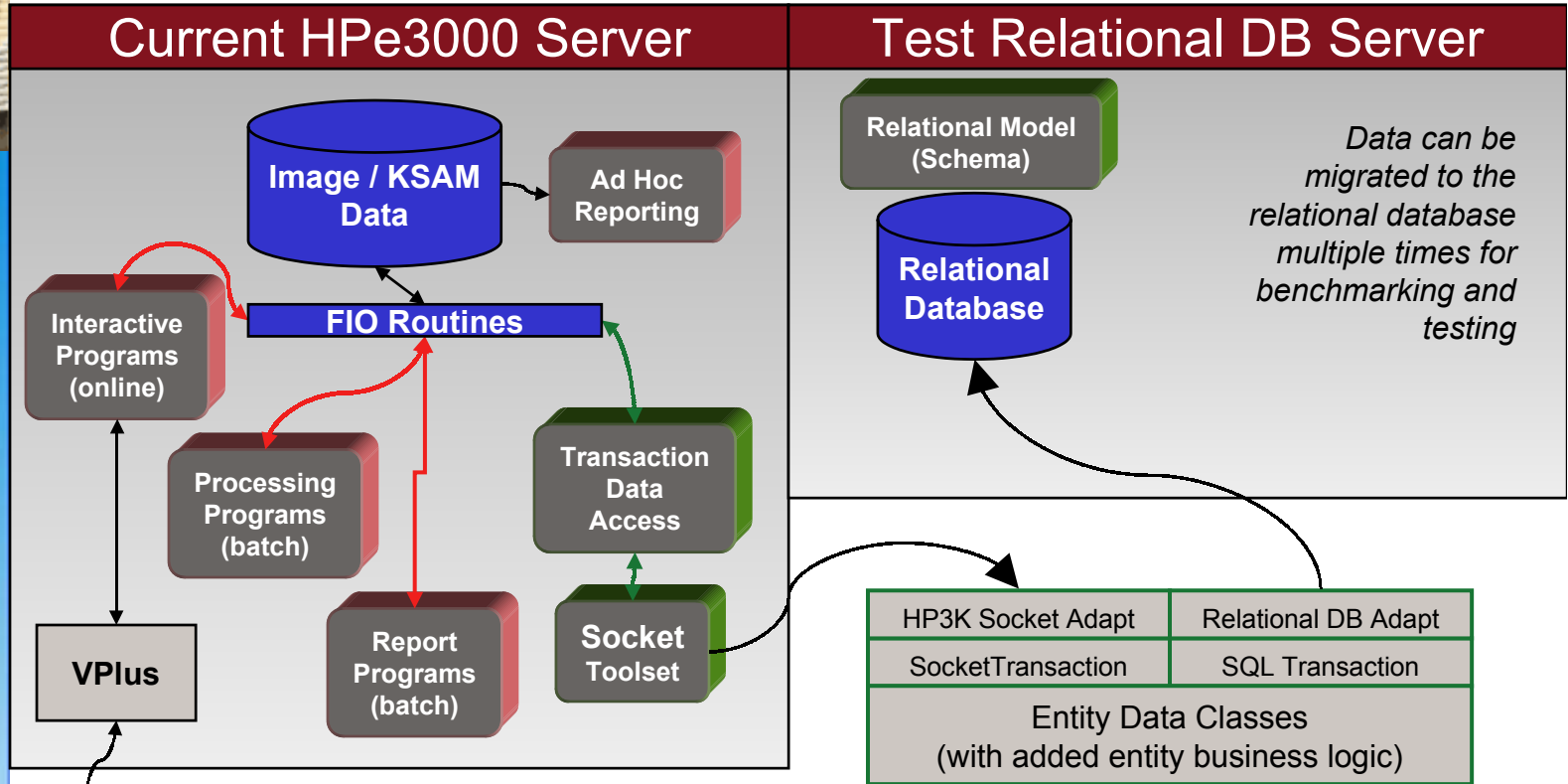


*For most COBOL reporting and Processing programs, the FIO routines can be changed to access the new database and the COBOL migrated to a new COBOL compiler supported by the intended Database Server.*



# EasyStep Approach – Step 6

## Data Migration (Conversion)

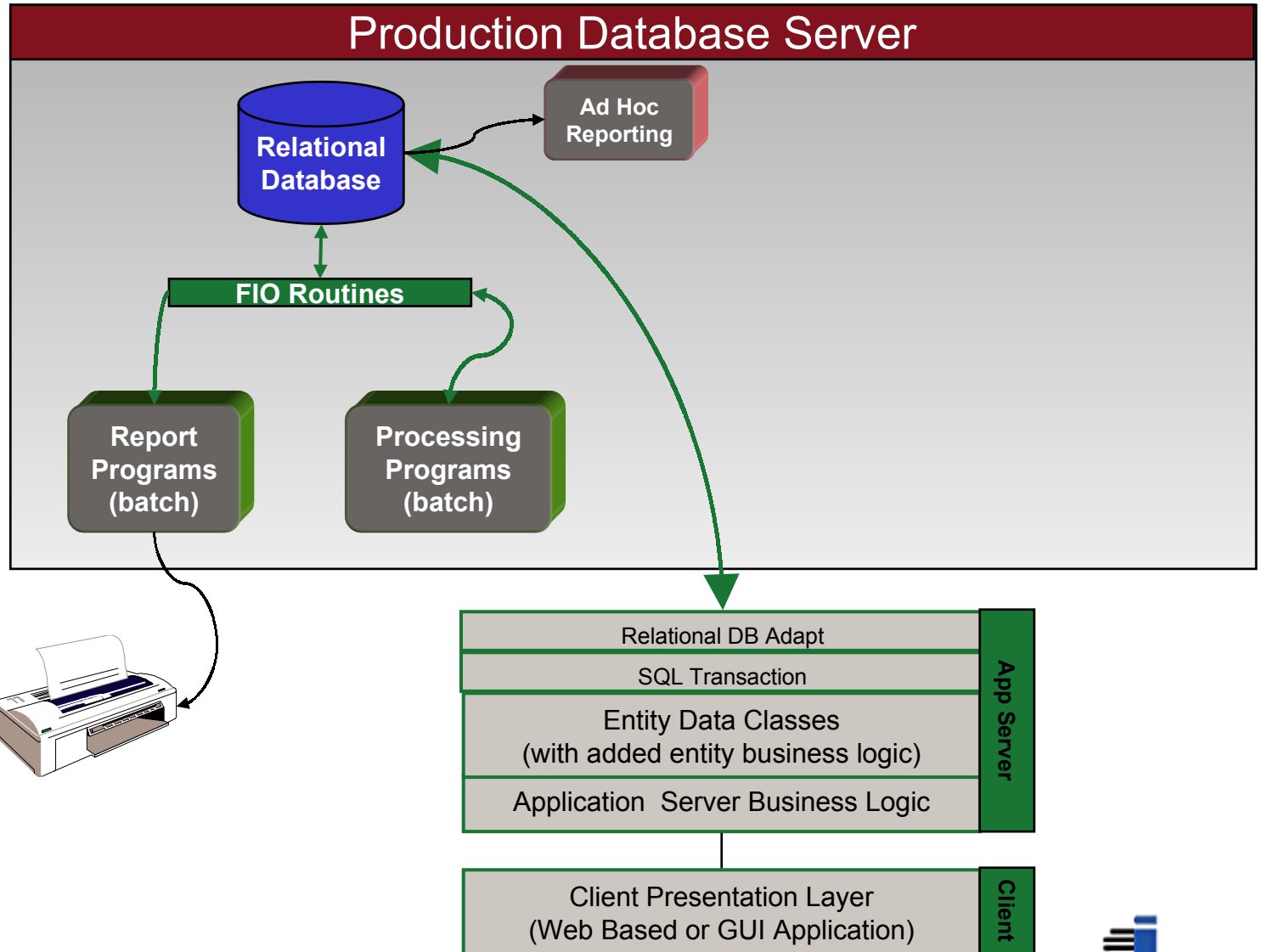


This phase involves the movement of data from the data structures on the HPe3000 to the relational database. This migration can occur for specific entities multiple times during the testing process and form the basis of the final conversion of data to the new database.



# EasyStep Approach – Step 7

## Final Solution





# Benefits and Advantages with EasyStep Approach

- Gradual approach is possible
- New GUI screens can be used with current HP e3000 data
- Same screens can access Image data on HP e3000 and new platform (e.g., Oracle data on HP 9000)
- Parallel testing can take place



# N-tier Architecture Design

- “n” represents any number of physical or logical layers
- Each layer is made up of software components that interact with other components within and across layers
- Consists of at least three layers: presentation, business, and data



# N-tier Architecture Design (Cont.)

- Presentation Layer: provides software interface that end user sees
- Business Layer: provides the foundation for components that provide the processing, business rules, validation, data manipulation, data retrieval, error handling, and other core logic that makes the system “smart”
- Data Layer: Provide the storage platform



# Benefits and Advantages with N-tier Architecture

- Scalability – Each component can reside on a separate machine to increase performance
- Flexibility – Components are interchangeable



# Data Access Approach





# Types of Data Processing

## Online Transaction Processing (OLTP)

- High transaction throughput
- Add/change/delete data
- Predefined transactions
- Response times critical



# Types of Data Processing (Cont.)

## Information on Demand - Ad Hoc Reporting

- Inquire and Report data only
- Undefined transactions - different each time
- Response times not as critical



# ODBC Defined

- Open Database Connectivity (ODBC)
- Used for standard access to multiple types of databases (Image, Oracle, etc)
- Allows PC access from applications like Excel to HP e3000 TurboImage data
- Converts SQL/ODBC calls to TurboImage calls
- Good for Ad Hoc Reporting and extracting data to other applications



# Sockets Defined

- Sockets are a method of establishing a connection between different machines and/or operating systems
- Socket ports are similar to phone numbers for a machine
- Uses low level TurboImage native database access routines that are already developed
- Good for high volume OnLine Transaction Processing (OLTP) performance



# ODBC Benchmarks

- Higher HP e3000 CPU usage
- Low concurrency of users, slower response times under load
- Database locking problems with other applications, not controlled by the application



# Socket Technology Benchmarks

- Low HP e3000 CPU usage (similar to current native HP e3000 applications)
- High concurrency of users
- Database locking controlled by the server program on the HP e3000



# Socket Technology Application Goals

- Fastest response on HP3000-based data
- Read and Update ability from the PC application to the HP e3000
- No performance hit to the production HP e3000 environment



# Transactionalize Data





# Socket Communications

- Socket communications are a way to connect different machines without understanding the different network protocols
- All connectivity between machines uses sockets at the low level
- Berkley Software Distribution Interprocess Communications (BSD IPC) is a standard available on almost all machines



## Socket Communications (Cont.)

- Network Interprocess Communications (NetIPC) is similar and compatible to BSD IPC, but has additional functionality on the HP e3000
- Perform system calls to establish connections and transfer data between machines



# Socket Communications (Cont.)

Processes required for socket connections:

- Listener Process (Waits for new connection requests)
- Server Process (Handles requests once connected)
- Client Process (Asks for connection, sends requests, accepts returned data)



# Listener Process

- A background job on the HP e3000 waits for “calls” on a specified port from clients
- Define the port in the services.net.sys file with a port over 20,000
- Once a request is received, create a separate socket connection for the client to use when sending and receiving information across the socket and start an individual server process



# Listener Process Example

MAIN.

PERFORM OPEN-LISTENER.

PERFORM LISTEN-FOR-CONNECTION UNTIL DONE.

OPEN-LISTENER.

CALL INTRINSIC "IPCCREATE" USING SOCKETKIND,  
PROTOCOL, FLAGS, OPT, CALLEDESC, RESULT.

LISTEN-FOR-CONNECTION.

CALL INTRINSIC "IPCRCVCN" USING CALLEDESC,  
VCDESC, FLAGS, \\\, RESULT.

CALL "SERVER".



# Server Process – HP e3000 Data Layer

- A unique child process is started for each client connection
- All socket communications are handled by the server process
- The server process handles transaction requests and enforces business rules
- Calls sub-routines to handle individual transaction requests



# Server Process Example

MAIN.

```
CALL INTRINSIC "IPCRCV" USING  
IPC-VCDESC, WS-SOCKET-IN,  
IPC-DLEN, IPC-FLAGS, \\  
IPC-RESULT.
```

```
CALL "CUSTOMER".
```

```
CALL INTRINSIC "IPCSEND" USING  
IPC-VCDESC, LS-SOCKET-OUT,  
IPC-DLEN, IPC-FLAGS, \\  
IPC-RESULT.
```



# Customer Transaction Example

CUSTOMER.

MOVE CUST-KEY-IN TO SEARCH-KEY.

PERFORM READ-CUSTOMER.

IF NOT CUSTOMER-FOUND

PERFORM SET-READ-ERROR

ELSE

PERFORM LOAD-SOCKET-BUFFER.





# Presentation Layer

- Microsoft Excel & Visual Basic
- Microsoft Visual FoxPro
- Web Access



# Client Process

- Use BSD IPC to connect to the HP e3000 on the predefined socket port
- Client initiates connection, then is moved to a separate unique socket connection for further transactions
- Each window on a client can have a unique socket connection or share one for the machine



# Microsoft Visual FoxPro

- True Object Oriented Programming
- Using F1 Technologies' Visual Fox Express framework
- Uses C++ DLL for low level socket control



# Visual FoxPro Example

**F1** OG Socket Cursor Class

Step 4 - Define Cursor Behavior

Default Alias: Customer

Buffer Mode Override: 3 - Optimistic Row Buffering

Trans Code Length	8	Update Trans. Code	CUSTU
Add Trans. Code	CUSTA	List Trans. Code	CUSTL
Delete Trans. Code	CUSTD	List Next Trans. Code	CUSTN
Inquire Trans. Code	CUSTI	Previous Trans. Code	CUSTP

Navigation buttons: << < > >>

Finish... Cancel



# Visual FoxPro Example (Cont.)

**Customer Maintenance**

Customer No. 100

Name Sunnyside Furniture

Address1 10736 Solar Blvd.

Address2 po box 123

City Los Angeles State AZ Zip Code 99999-9999

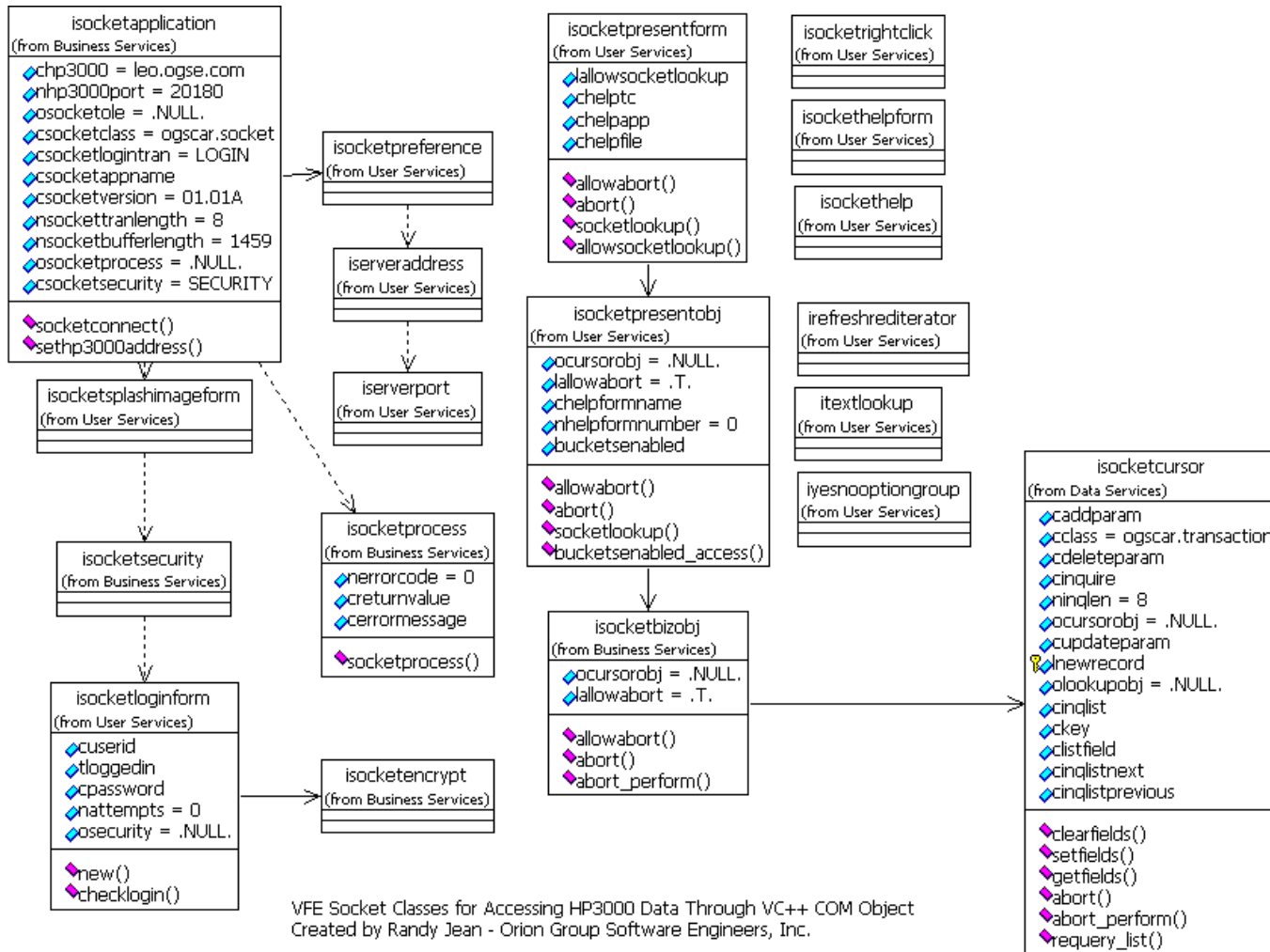
Country U.S.A.

Control Information   Contact Information   Financial Summary

Bal Method	Open Item Accounting	Credit Limit	10,000
Stmt Freq	Quarterly	Credit Rate	BBB
Location	AT Atlanta, GA	Terms Code	N Cash with order
Territory	XX	Ship Via	F Federal Express
Frt Pmt Cd	Prepaid	Tax Code	NY NEW YORK STATE SALES TAX
Partial Ship?	<input checked="" type="radio"/> Yes <input type="radio"/> No	Sales Rep No	500 Norm Z. Goldstein
Finance Chg?	<input checked="" type="radio"/> Yes <input type="radio"/> No	A/R Account	FURNITUR-00100060-01000000 Accounts Receivable - West
Discount %	0.00	Comment	GOOD CUSTOMER
Credit Status	Good Credit		



# Visual FoxPro Example (Cont.)





## Web Access

- The transactions created on the HP e3000 can also be used to create web applications
- Works with HTML, Active Server Pages, Web Objects, etc.
- Orion Group has created socket frameworks in Java and C++ to aide in application development



# Web Access Example

**Account Information - Netscape**

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Location: ebObjects/ECDemo.woa/32/wo/sl54DUDsZlqI2C8ExTc6e5kpFIX/1.6.21.0

Staff Locator Instant Message SmartUpdate

**Cass Furniture Distribution**

Products Search Quick Entry View Order Account Info Order History

### Account Information

Address	Purchases	Payments	
Sunnyside Furniture (100)	Period to Date	Last Payment	\$50.79
10736 Solar Blvd.	Year to Date	Payment Date	07/31/00
po box 123	Last Year	Current balance	\$344.56
Los Angeles, AZ 99999-9999		Other	
Phone number 213-657-4321		Credit Limit	\$10,000.00

Document: Done





**Case Study Will Be Presented**



**Questions?**

**Title:**            **Migrating HP e3000  
Applications**



**Presentation: 25**

*Authors:*        Matt Street & Donna Faudree

*Company:*      Orion Group Software Engineers  
5770 Nimtz Parkway  
South Bend, IN 46628

*Phone:*         (574) 233-3401

*E-mail:*         mstreet@ogse.com  
dfaudree@ogse.com