

Title: **The Easy Step Approach to Migration**

Presentation: 005

Authors: Matt Street & Donna Faudree

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

Phone: (574) 233-3401

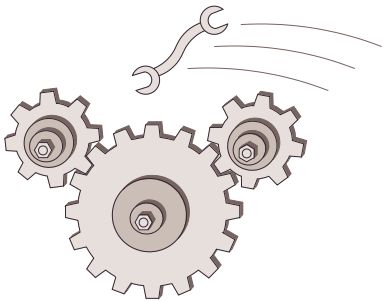
E-mail: mstreet@ogse.com
dfaudree@ogse.com

What We'll Cover Today

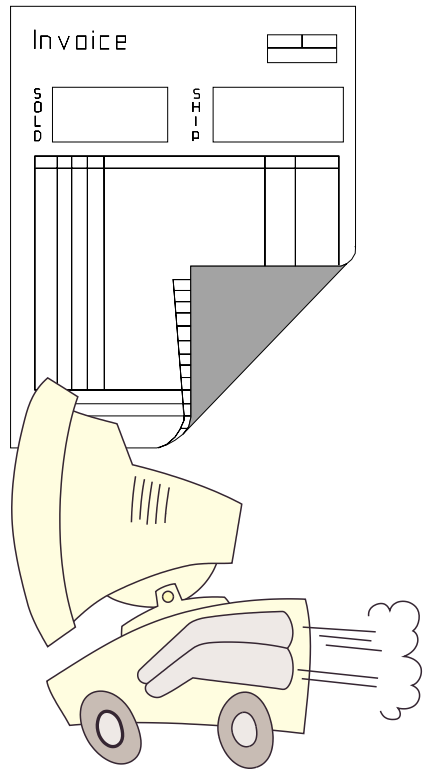
- Introduction
- EasyStep Approach
- N-Tier Architecture Design
- Data Access Approach
- Transactionalizing Entity Data
- Case Study
- Questions

Migration or Re-engineer

- Migration – the translation and re-hosting 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



A Happy Combination

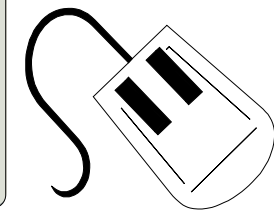
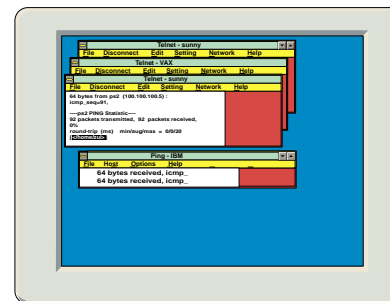


Migration:

- Report Programs
- Processing Programs (Batch processes)
- Command Files
- Image data

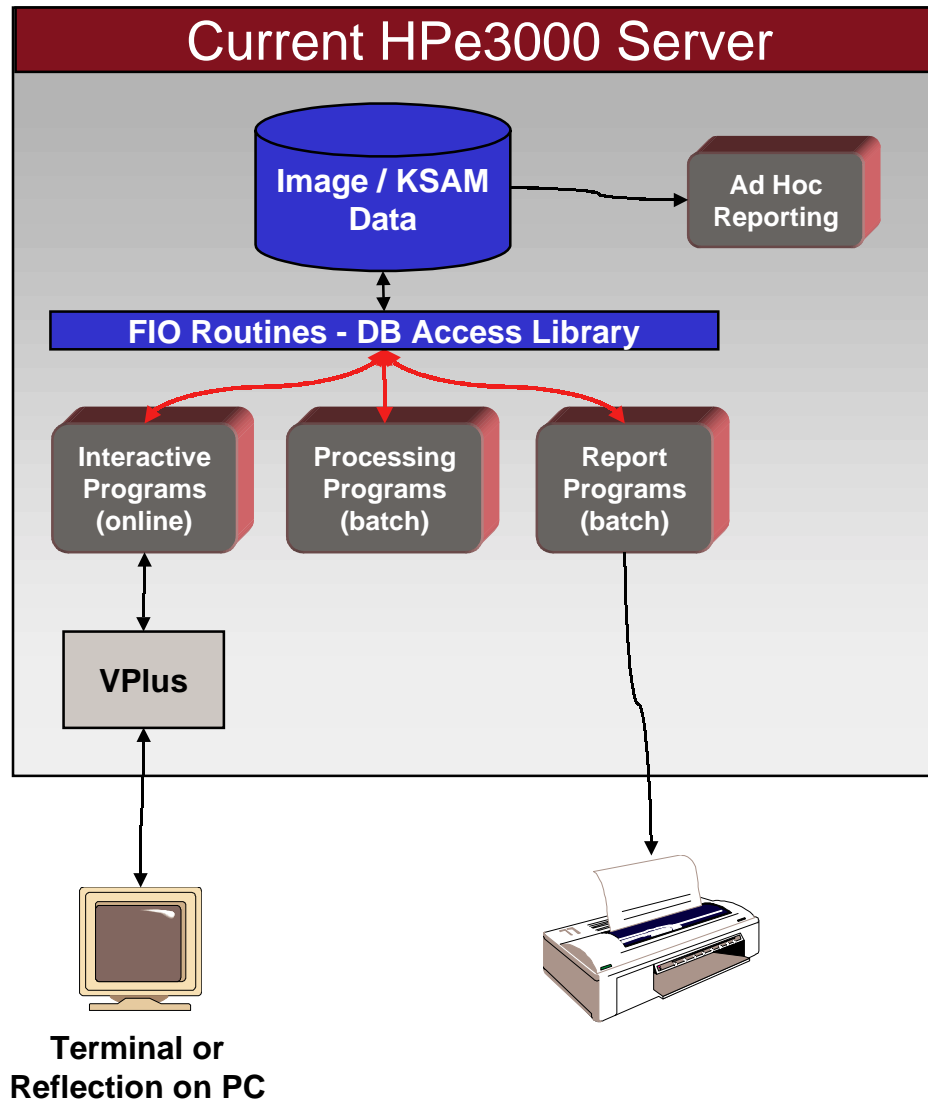
Re-engineer:

- User Interface



EasyStep Approach – Step 1

Current Environment

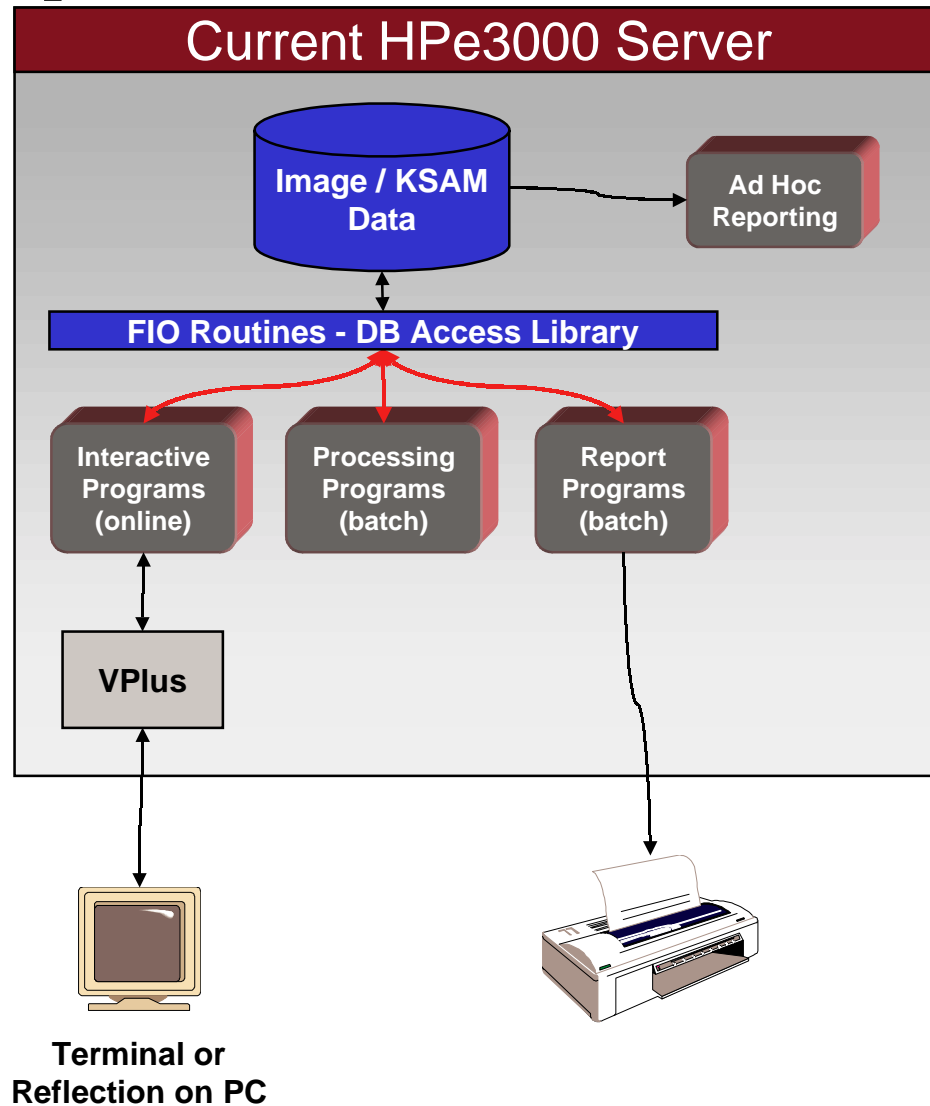


May Consists of:

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

EasyStep Approach – Step 2

Develop Relational Model

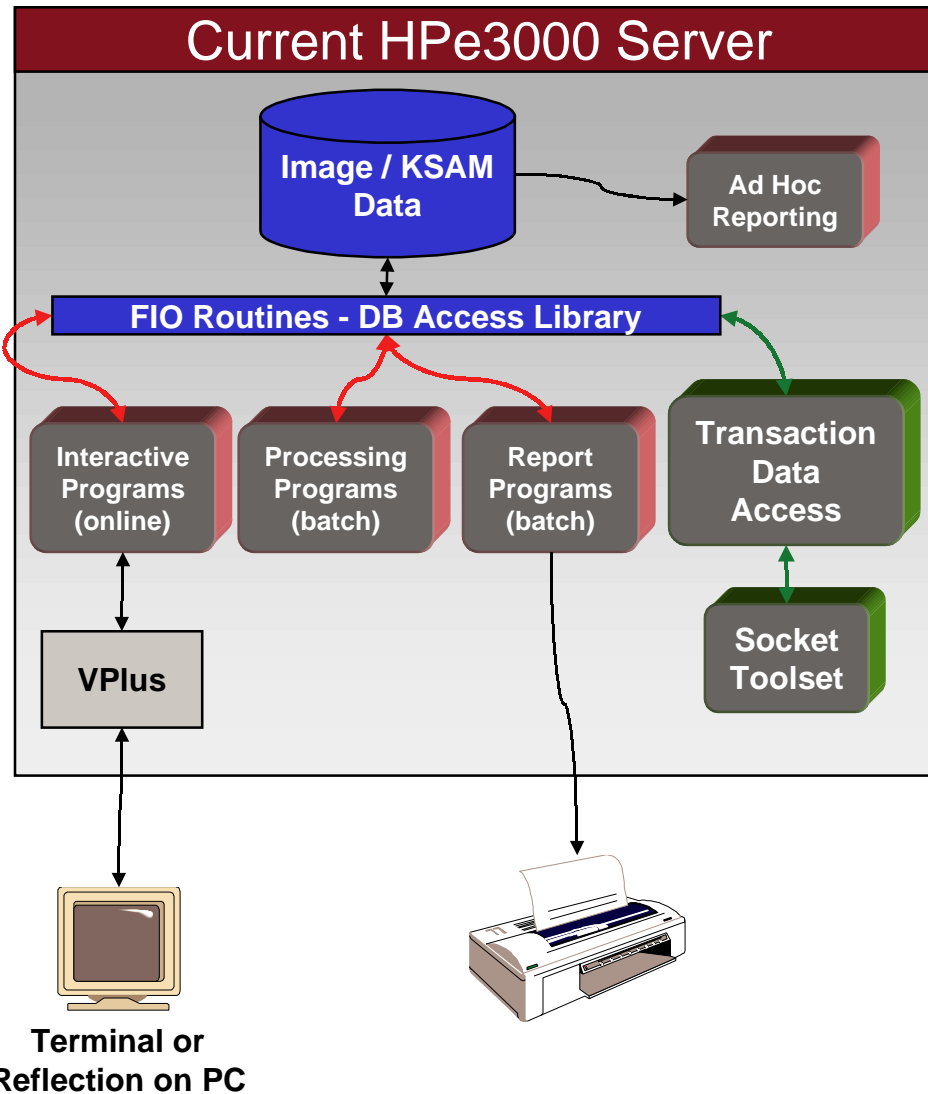


Relational Model (Schema)

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 HPe3000 Data

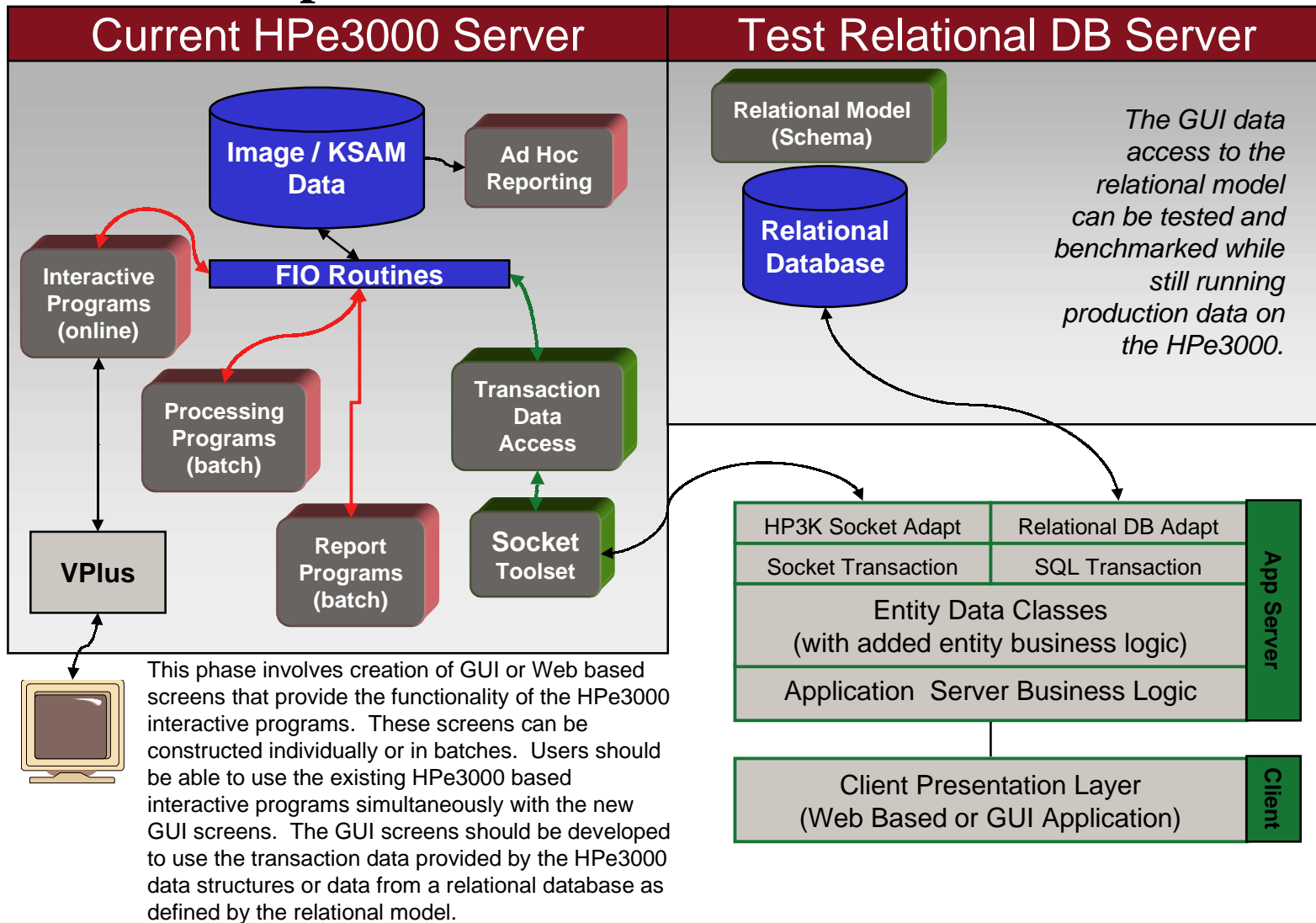


Relational Model (Schema)

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.

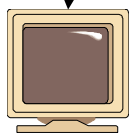
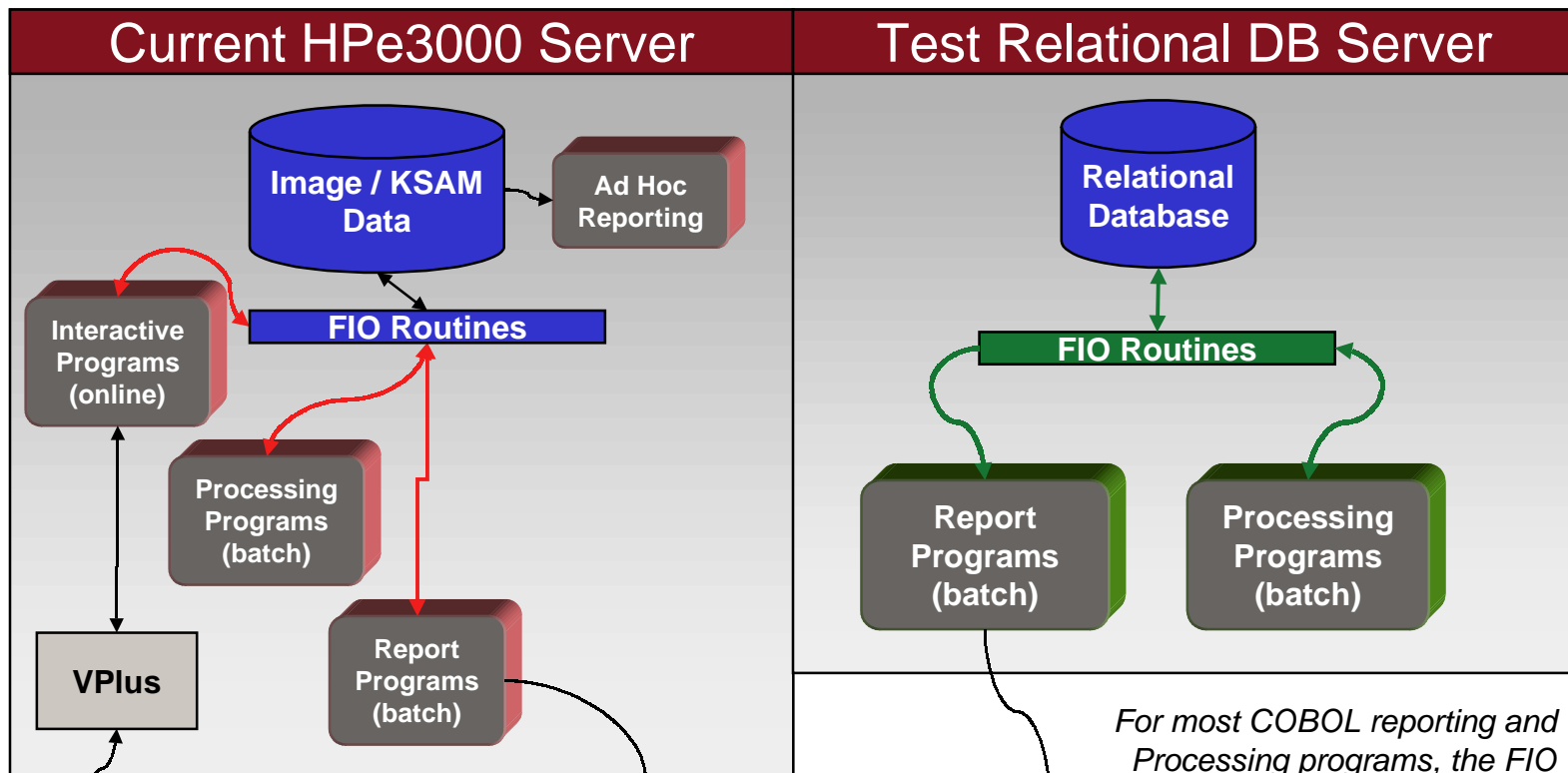
EasyStep Approach – Step 4

GUI Development

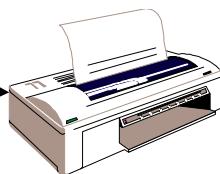


EasyStep Approach – Step 5

Batch Program Development



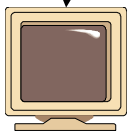
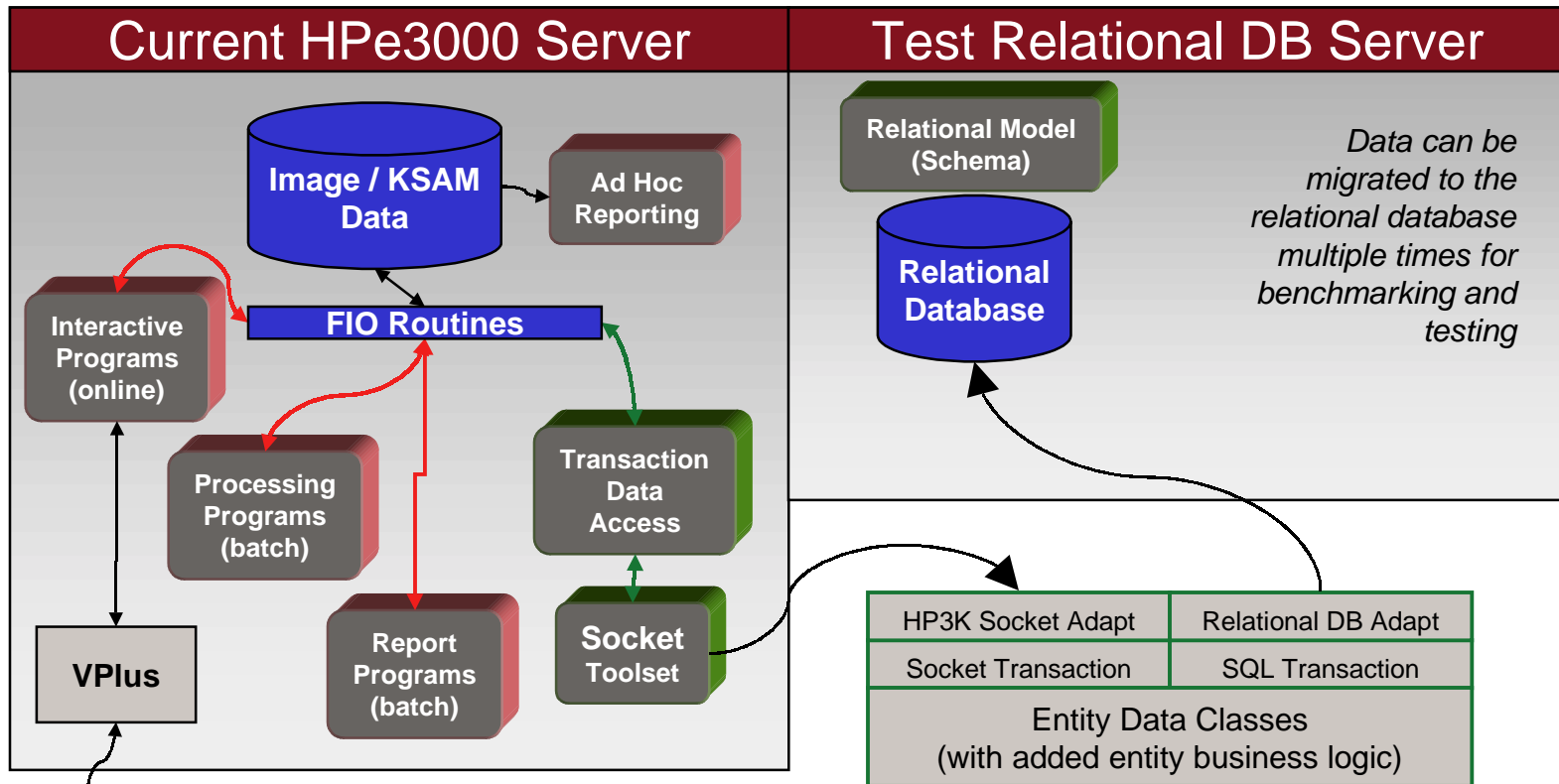
This phase involves the migration of batch programs which provide the reporting, posting or background transaction processing functions of the system. This phase can be done in conjunction with or separate from GUI development.



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

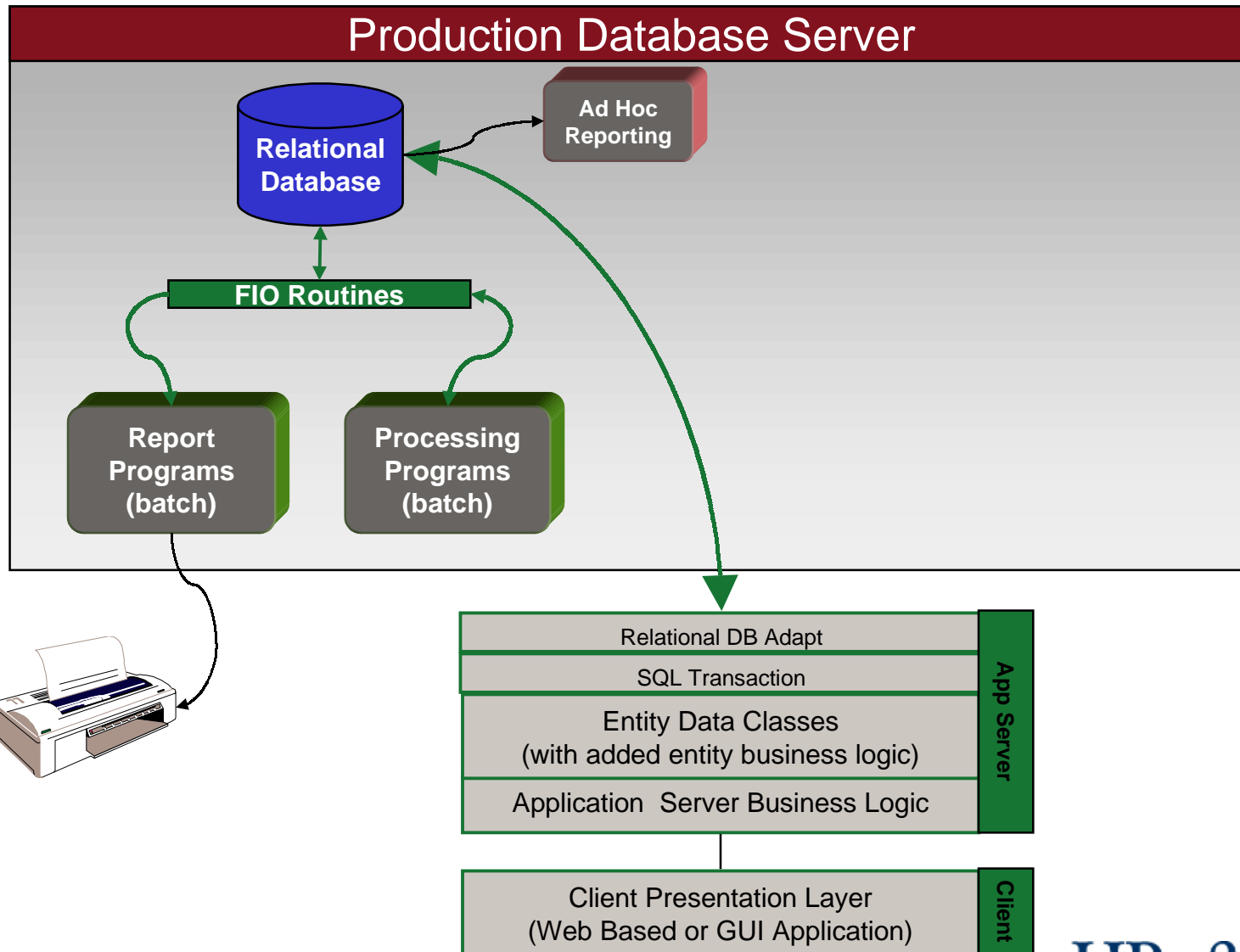
Data Migration (Conversion)



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.

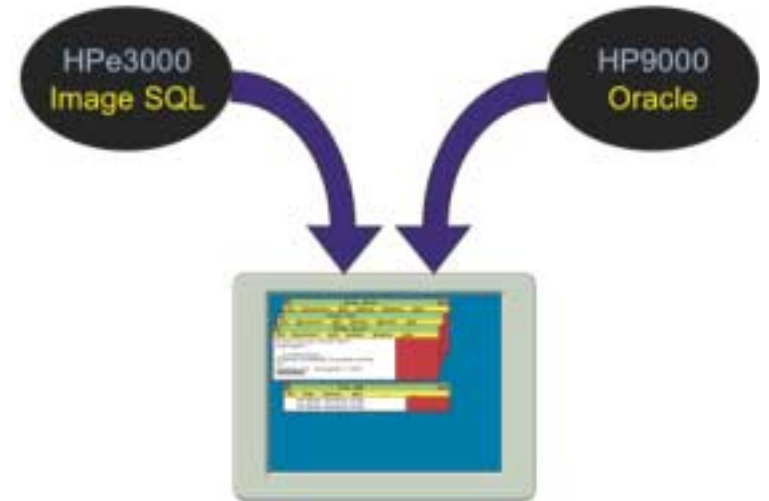
EasyStep Approach – Step 7

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



PARALLEL TESTING
PARALLEL TESTING



HP e3000
SOLUTIONS SYMPOSIUM

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

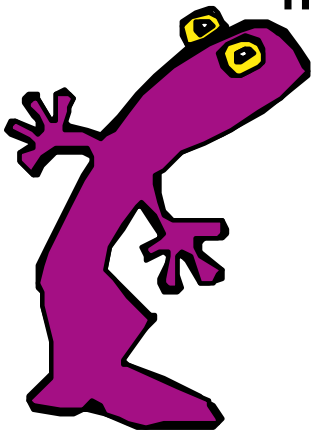


N-tier Architecture Design (Cont.)

- 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

Online Transaction Processing (OLTP)

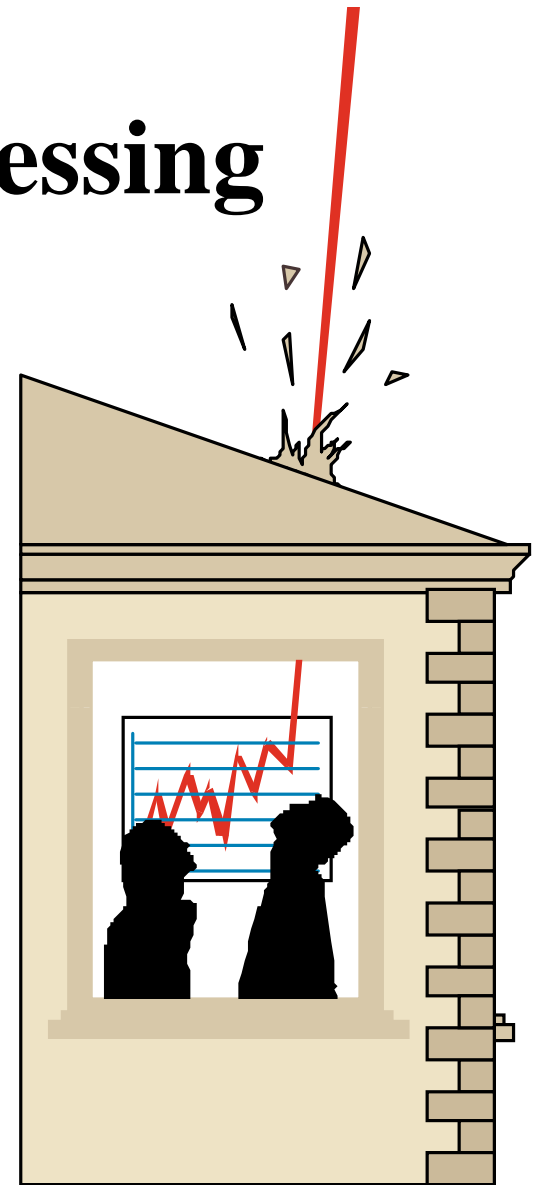
VS.

Information on Demand - Ad Hoc Reporting

Online Transaction Processing (OLTP)

Characteristics:

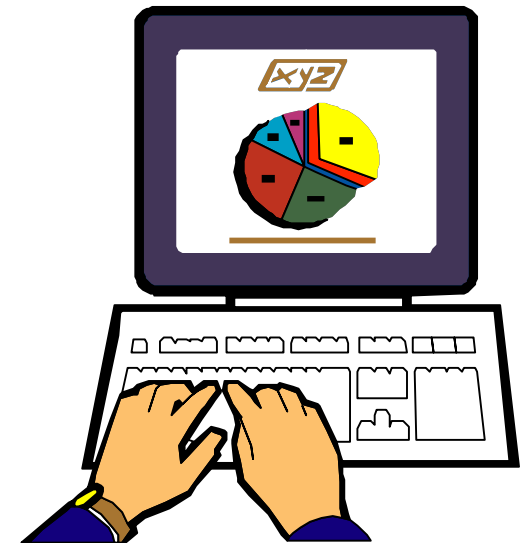
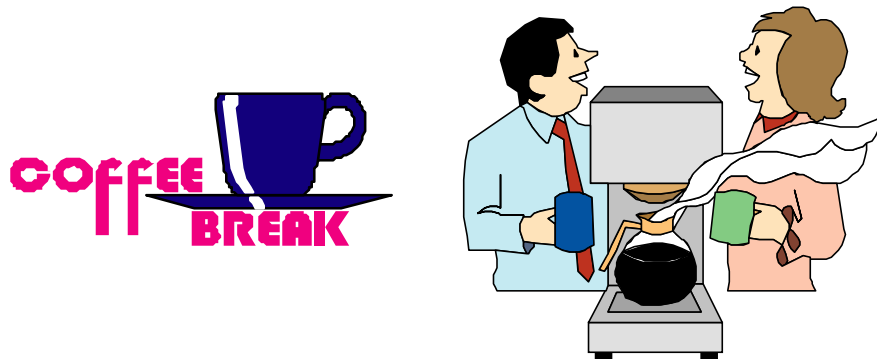
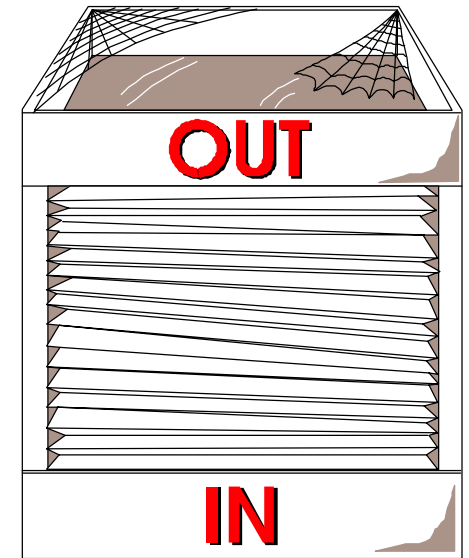
- High transaction throughput
- Add/change/delete data
- Predefined transactions
- Response times critical



Information on Demand / Ad Hoc Reporting

Characteristics:

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



Transactionalize HP e3000 Data

Defined:

Retrieve data from the HP e3000

And prepare it in such a way that it

Looks to the client GUI logic as if it

Was retrieved from a relational database

ODBC Approach

- 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

Socket Approach

- 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 when using with HPe3000

- 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 with HP e3000

- 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 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.)

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 Example

MAIN.

PERFORM OPEN-LISTENER.

PERFORM LISTEN-FOR-CONNECTION UNTIL DONE.

OPEN-LISTENER.

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

LISTEN-FOR-CONNECTION.

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

<<Create and Activate Server Process>>

CALL INTRINSIC "IPCGIVE" USING VCDESC,
SOCKETNAME, NLEN, FLAGS, RESULT.

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.
```

```
IF TRAN-IN OF WS-SOCKET-IN = "CUSTI "  
CALL "CUSTOMER".
```

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

Customer Transaction Example

CUSTOMER.

MOVE CUST-KEY-IN OF WS-SOCKET-IN
TO SEARCH-KEY.

PERFORM READ-CUSTOMER.

IF NOT CUSTOMER-FOUND

PERFORM SET-READ-ERROR

ELSE

PERFORM LOAD-WS-SOCKET-OUT.

Presentation Layer

- Microsoft Excel & Visual Basic
- Microsoft Visual FoxPro
- Java Client (three tier model)
- 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

The screenshot shows the 'Socket Cursor Class' dialog box in Visual FoxPro, specifically the '4 - Define Cursor Behavior' step. The dialog has a title bar with the text 'Socket Cursor Class' and a close button. Below the title bar is a 'Step' dropdown menu set to '4 - Define Cursor Behavior'. The main area contains several fields and buttons:

- Default Alias:** A text box containing 'Customer'.
- Buffer Mode Override:** A dropdown menu set to '3 - Optimistic Row Buffering'.
- Trans Code Length:** A text box containing '8'.
- Add Trans. Code:** A text box containing 'CUSTA'.
- Delete Trans. Code:** A text box containing 'CUSTD'.
- Inquire Trans. Code:** A text box containing 'CUSTI'.
- Update Trans. Code:** A text box containing 'CUSTU'.
- List Trans. Code:** A text box containing 'CUSTL'.
- List Next Trans. Code:** A text box containing 'CUSTN'.
- Previous Trans. Code:** A text box containing 'CUSTP'.

At the bottom of the dialog, there are four navigation buttons: a double left arrow, a single left arrow, a single right arrow, and a double right arrow. Below these are two buttons: 'Finish...' and 'Cancel'.

Visual FoxPro Example (Cont.)

Customer Maintenance

Customer No. 100

Name Sunnyside Furniture

Address1 10736 Solar Blvd.

Address2 po box 123

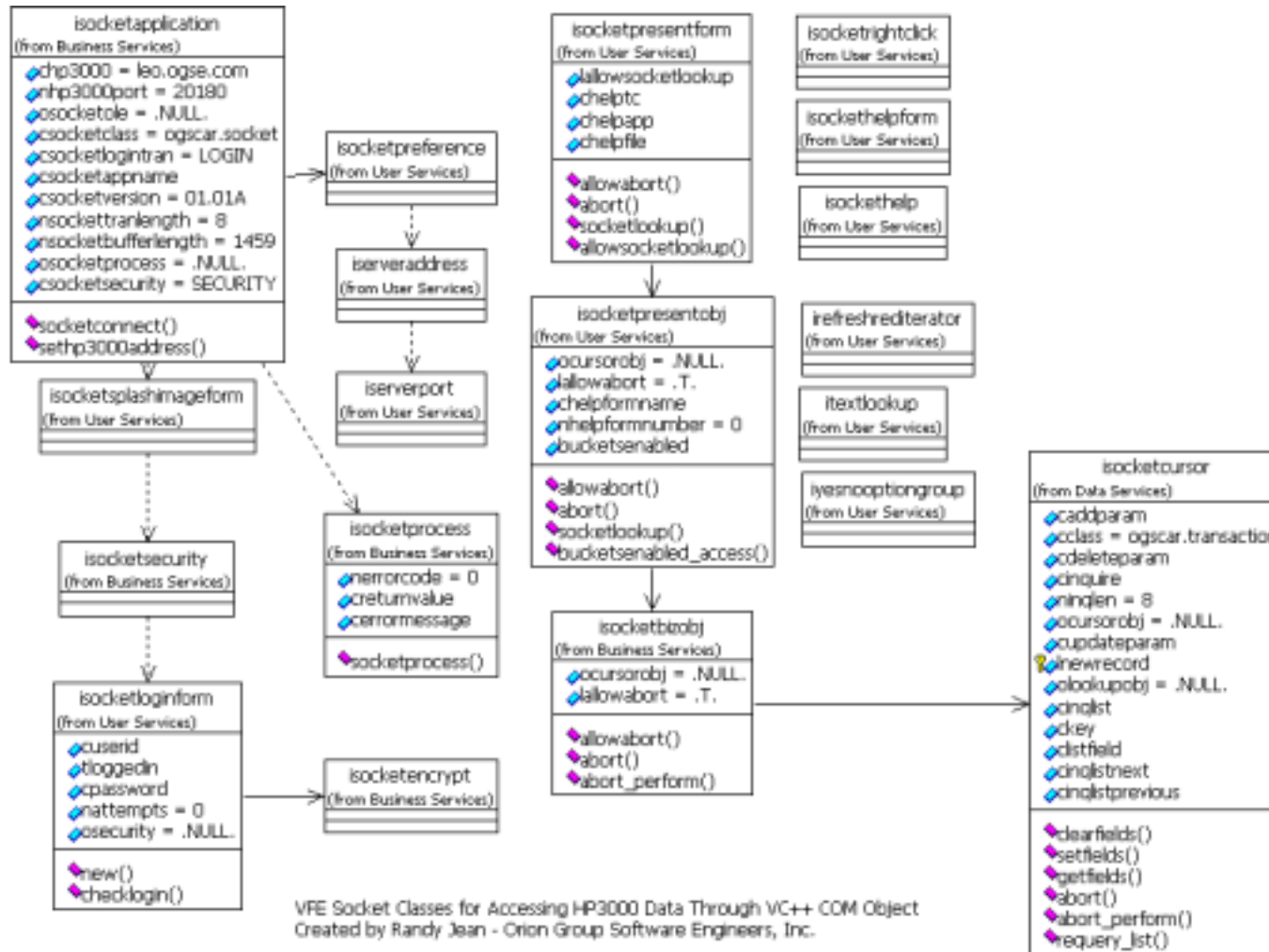
City Los Angeles State AZ Zip Code 99999-9999

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.)



VFE Socket Classes for Accessing HP3000 Data Through VC++ COM Object
Created by Randy Jean - Orion Group Software Engineers, Inc.

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
- Resulting Object on Client is same as if the data was retrieved from a Relational Database

Web Access Example

The screenshot shows a Netscape browser window titled "Account Information - Netscape". The address bar contains the URL: `ebObjects/ECDemo.woa/32/wo/sI54DUDsZlqI2C8ExTc6e5kpFIX/1.6.21.0`. The browser interface includes a menu bar (File, Edit, View, Go, Communicator, Help), a toolbar with icons for Back, Forward, Reload, Home, Search, Netscape, Print, Security, Shop, and Stop, and a status bar at the bottom showing "Document: Done".

The main content area displays the "Carr Furniture Distribution" logo and a navigation menu with buttons for Products, Search, Quick Entry, View Order, Account Info, and Order History. Below the menu, the "Account Information" section is visible, containing the following data:

Address	Purchases	Payments
Sunnyside Furniture (100)	Period to Date	Last Payment
10736 Solar Blvd.	Year to Date	Payment Date
po box 123	Last Year	Current balance
Los Angeles, AZ 99999-9999		
 		Other
Phone number 213-657-4321		Credit Limit
	\$104.56	\$50.79
	\$2,000.90	07/31/00
	\$344.45	\$344.56
		\$10,000.00

Case Study

Proof-Of-Concept Project

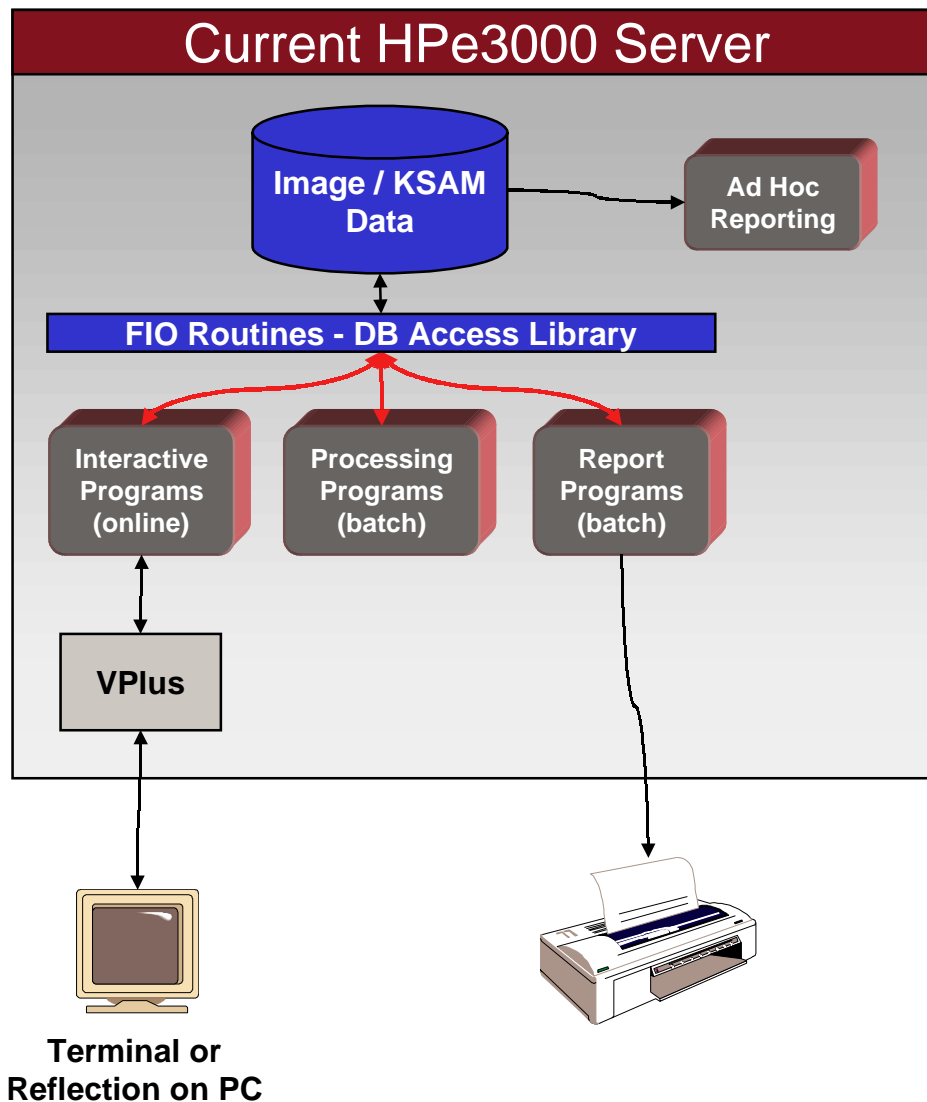
Scope:

- Migrate a subset of Image datasets to Oracle tables
- Migrate one COBOL batch program
- Re-engineer one or more screens
[create a GUI and a Web application]

Proof-Of-Concept Project Goals:

- Determine technical issues related to migration of data and COBOL programs
- Show different client technologies for re-engineered screens (Visual FoxPro or WebObjects)
- Develop “framework” for migration projects to increase efficiency and productivity while decreasing cost

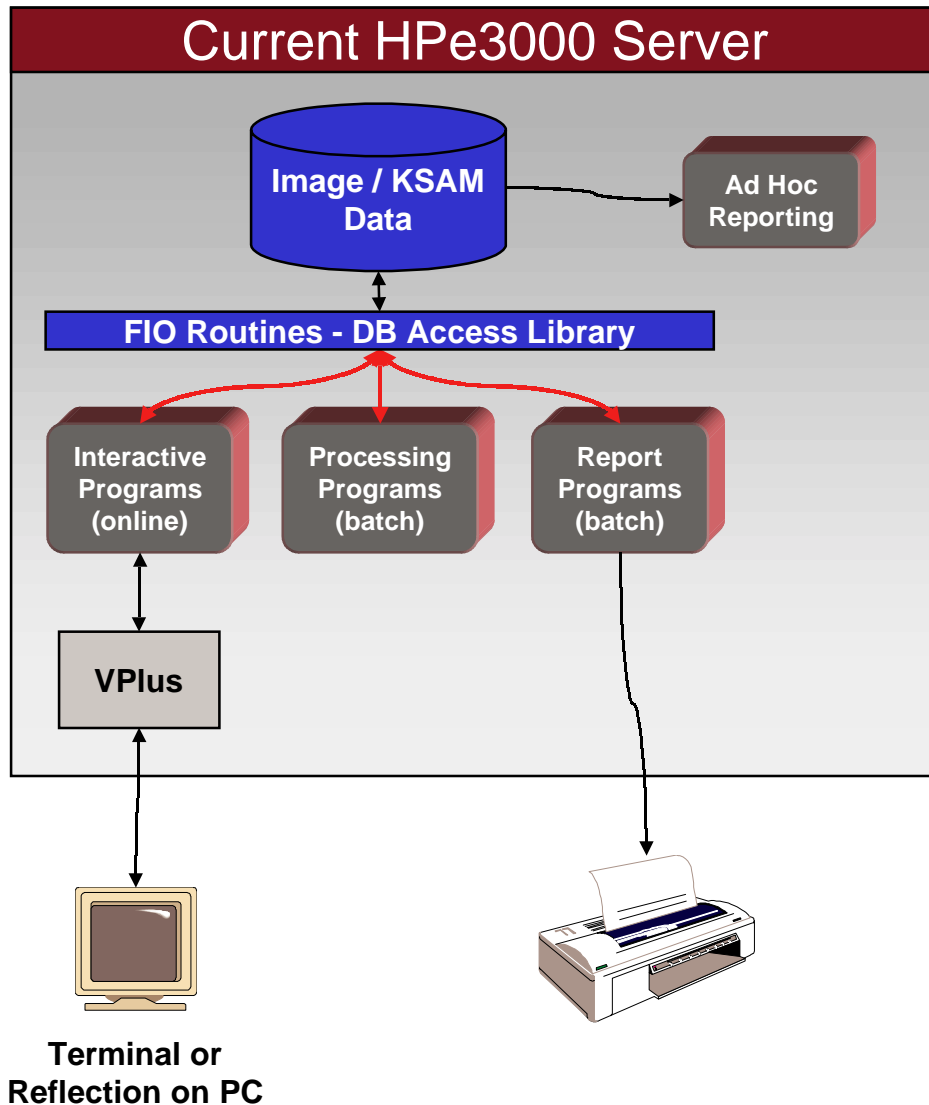
Case Study: Current Environment



May Consists of:

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

Case Study: Develop Relational Model

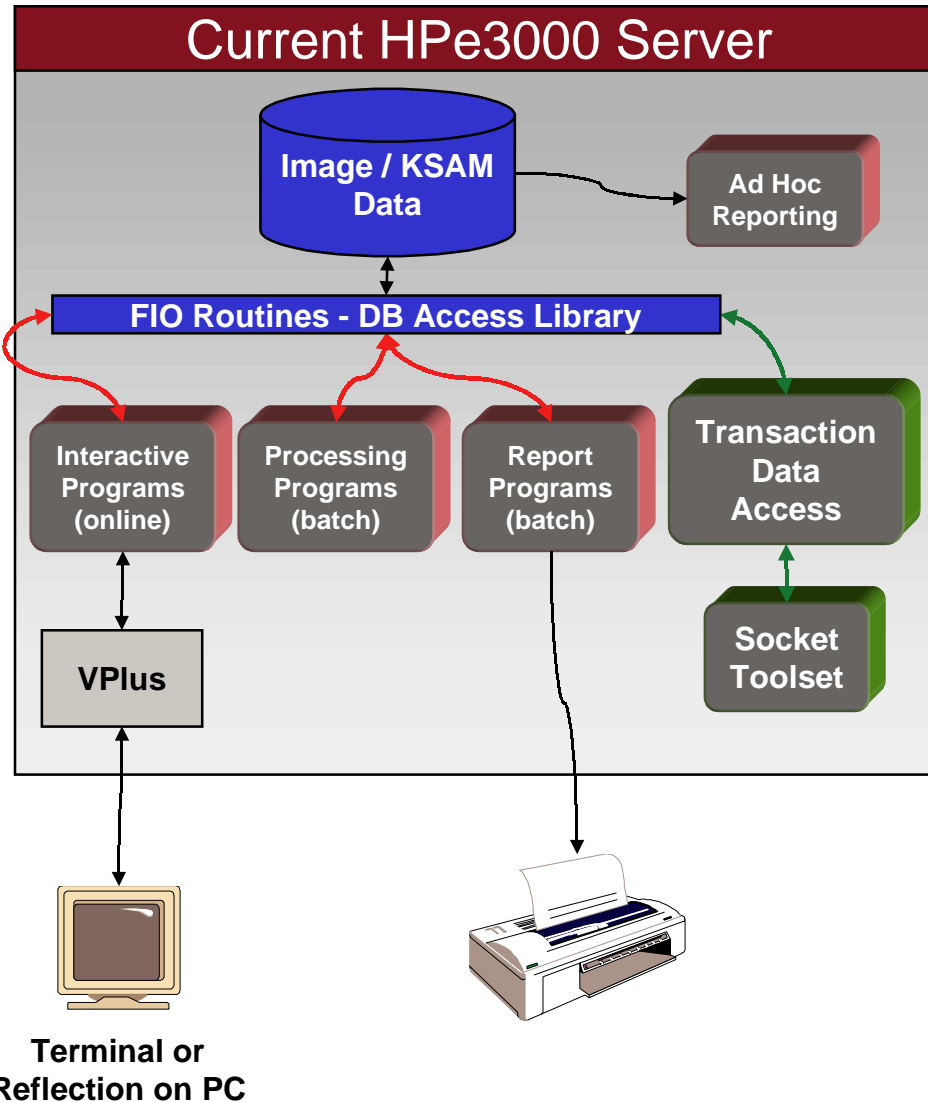


Relational Model (Schema)

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.

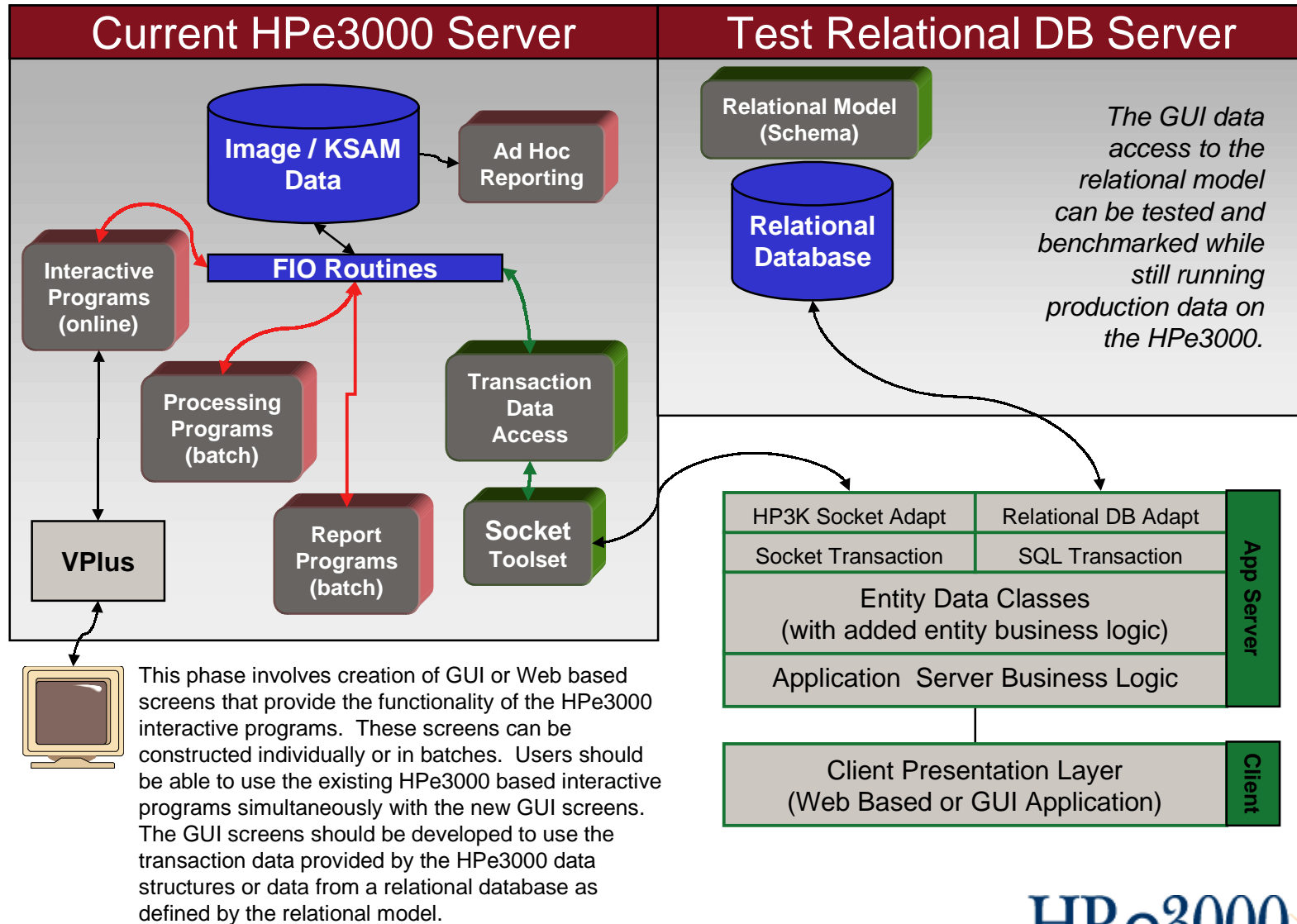
Case Study: Transactionalize HPe3000 Data



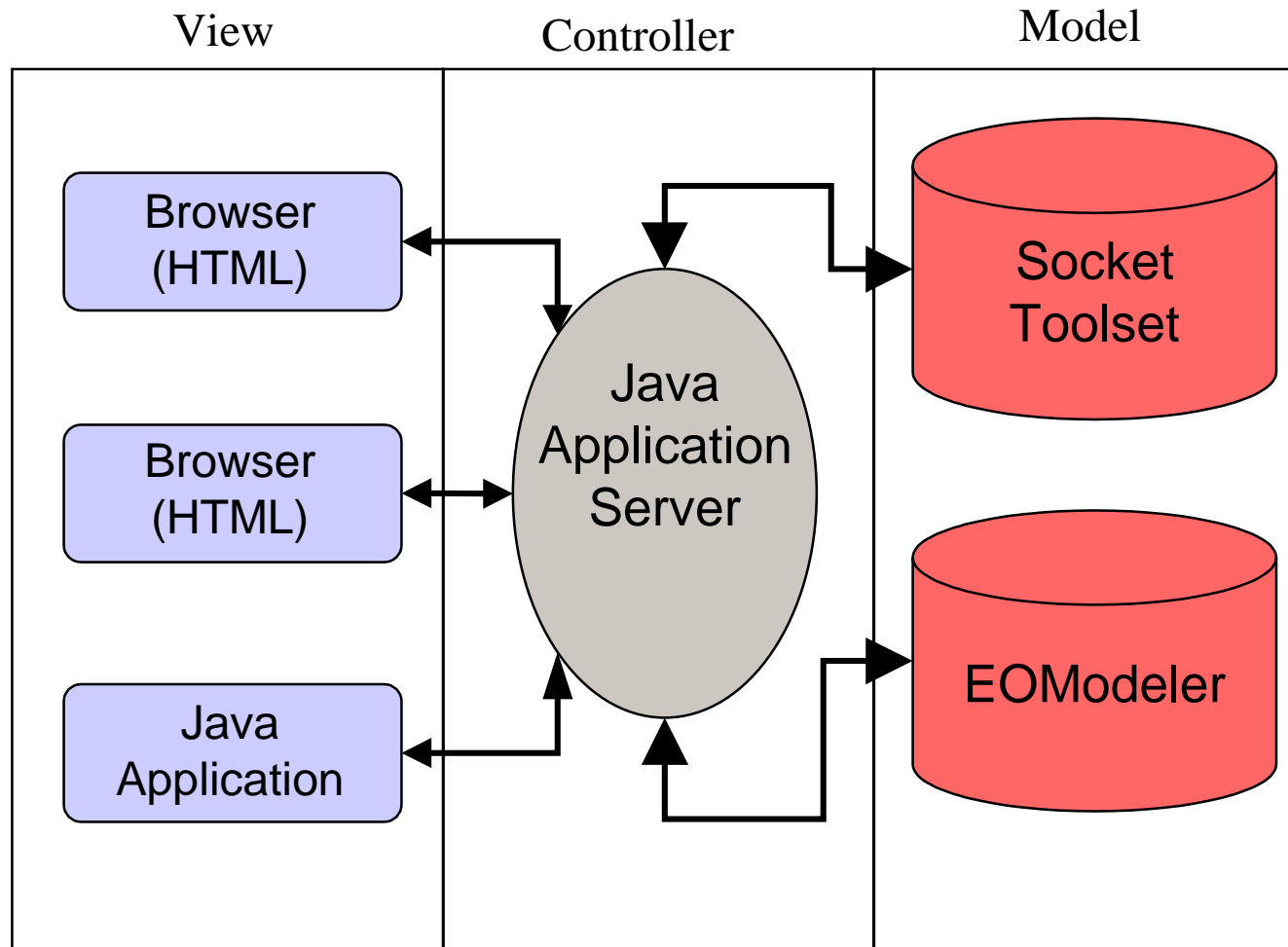
Relational Model
(Schema)

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.

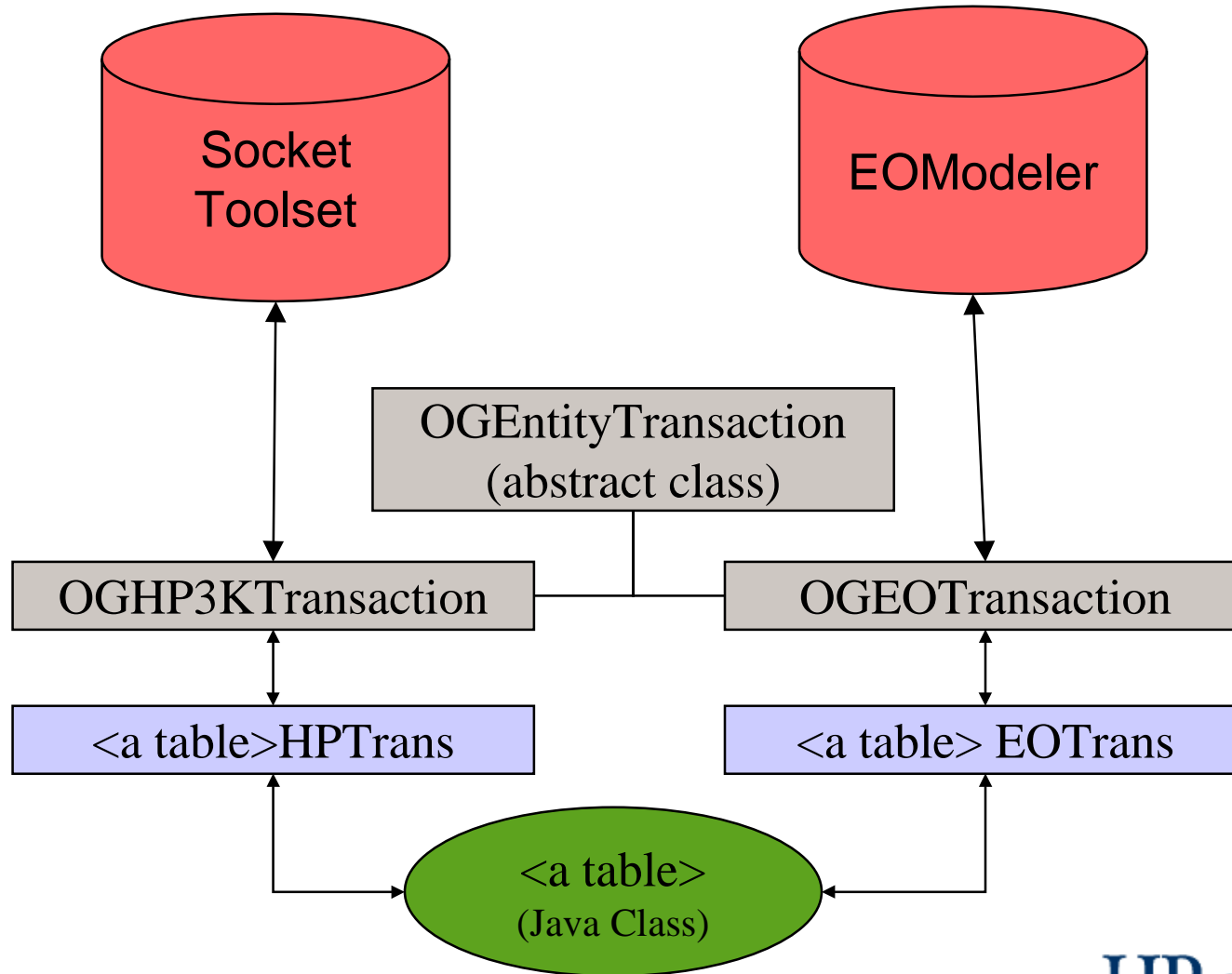
Case Study: GUI Development



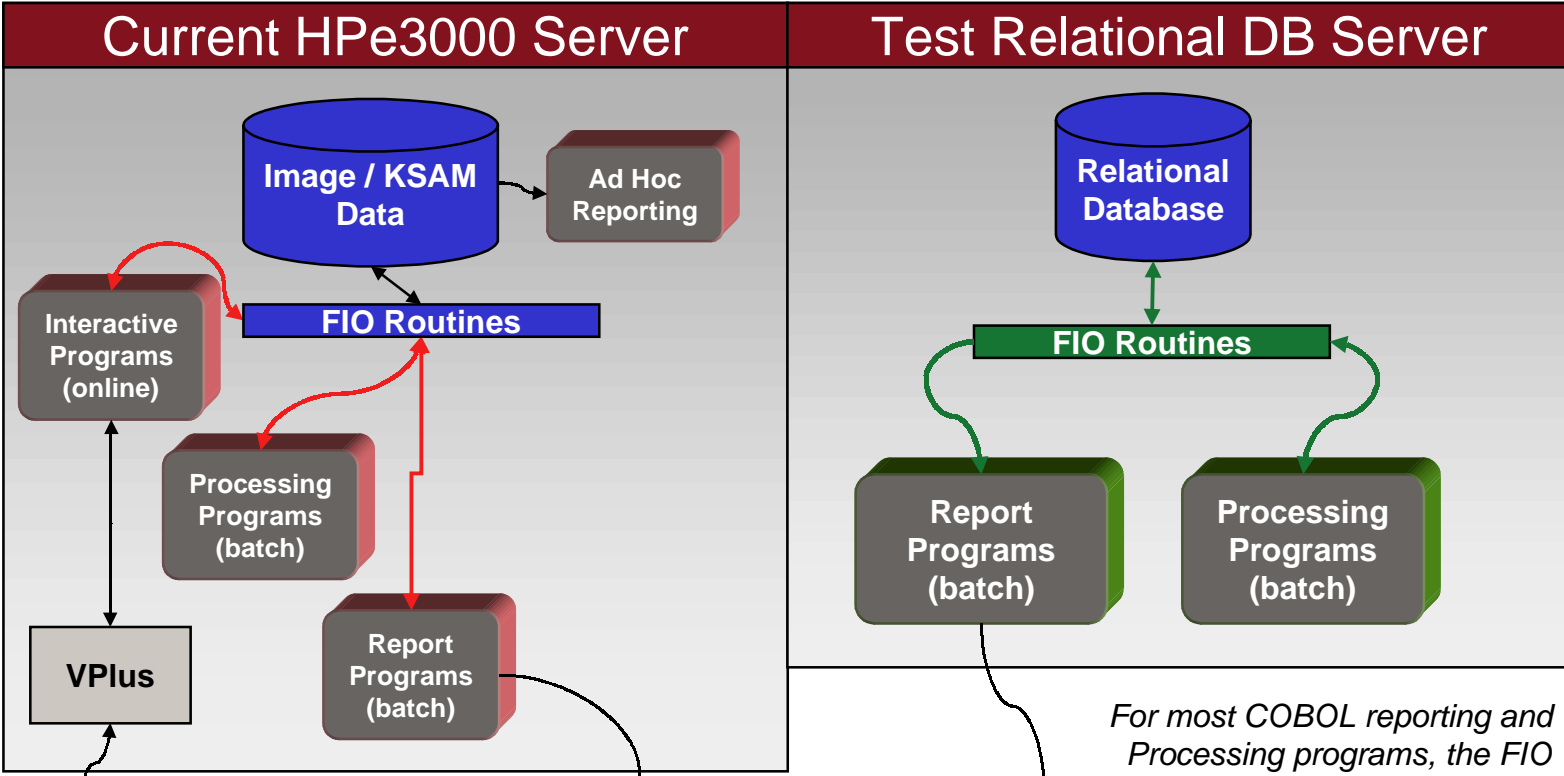
Case Study: GUI Development - WebObjects



Case Study: GUI Development - Java setup



Case Study: Batch Program Development

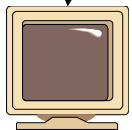
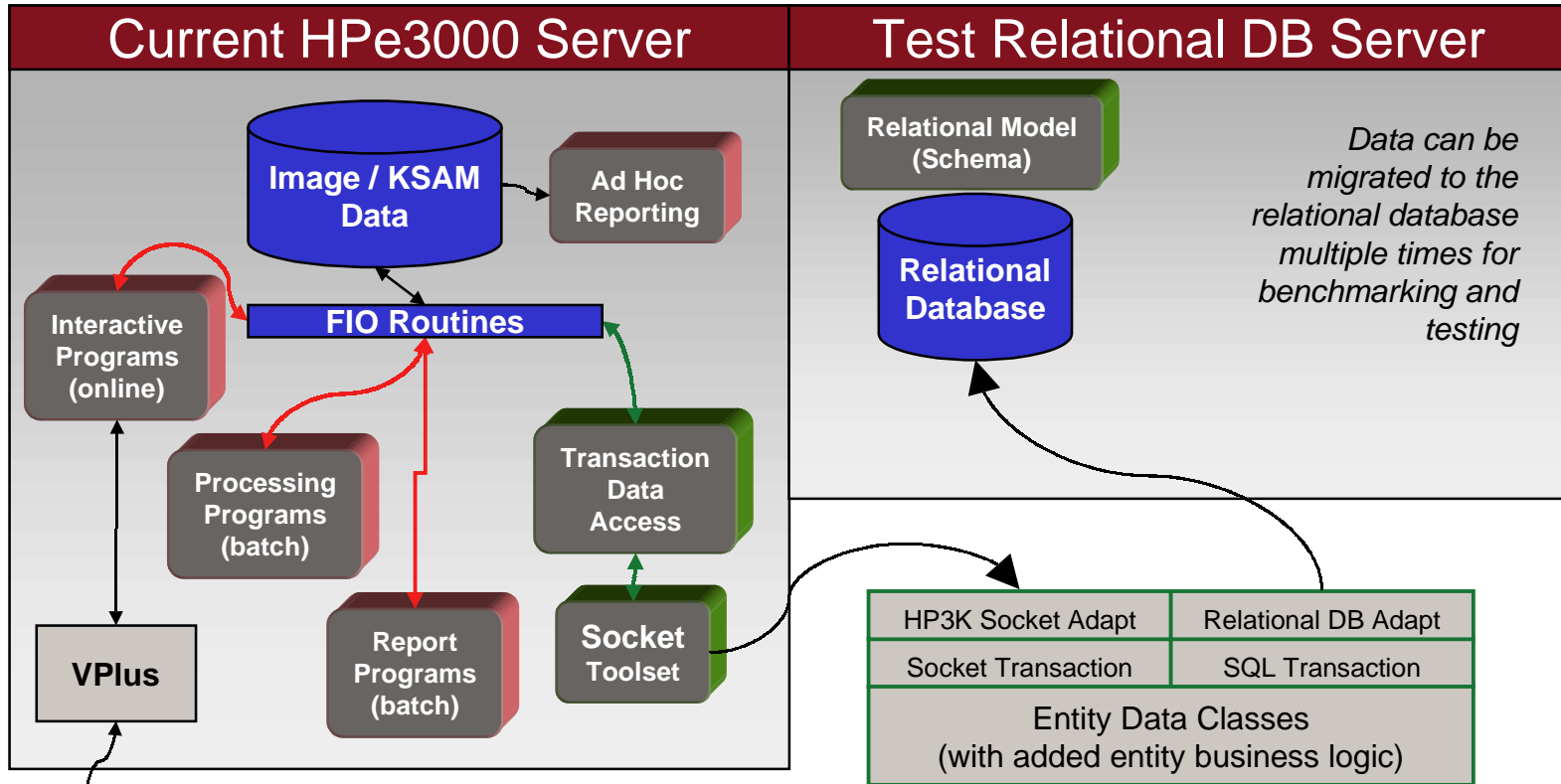


This phase involves the migration of batch programs which provide the reporting, posting or background transaction processing functions of the system. This phase can be done in conjunction with or separate from GUI development.



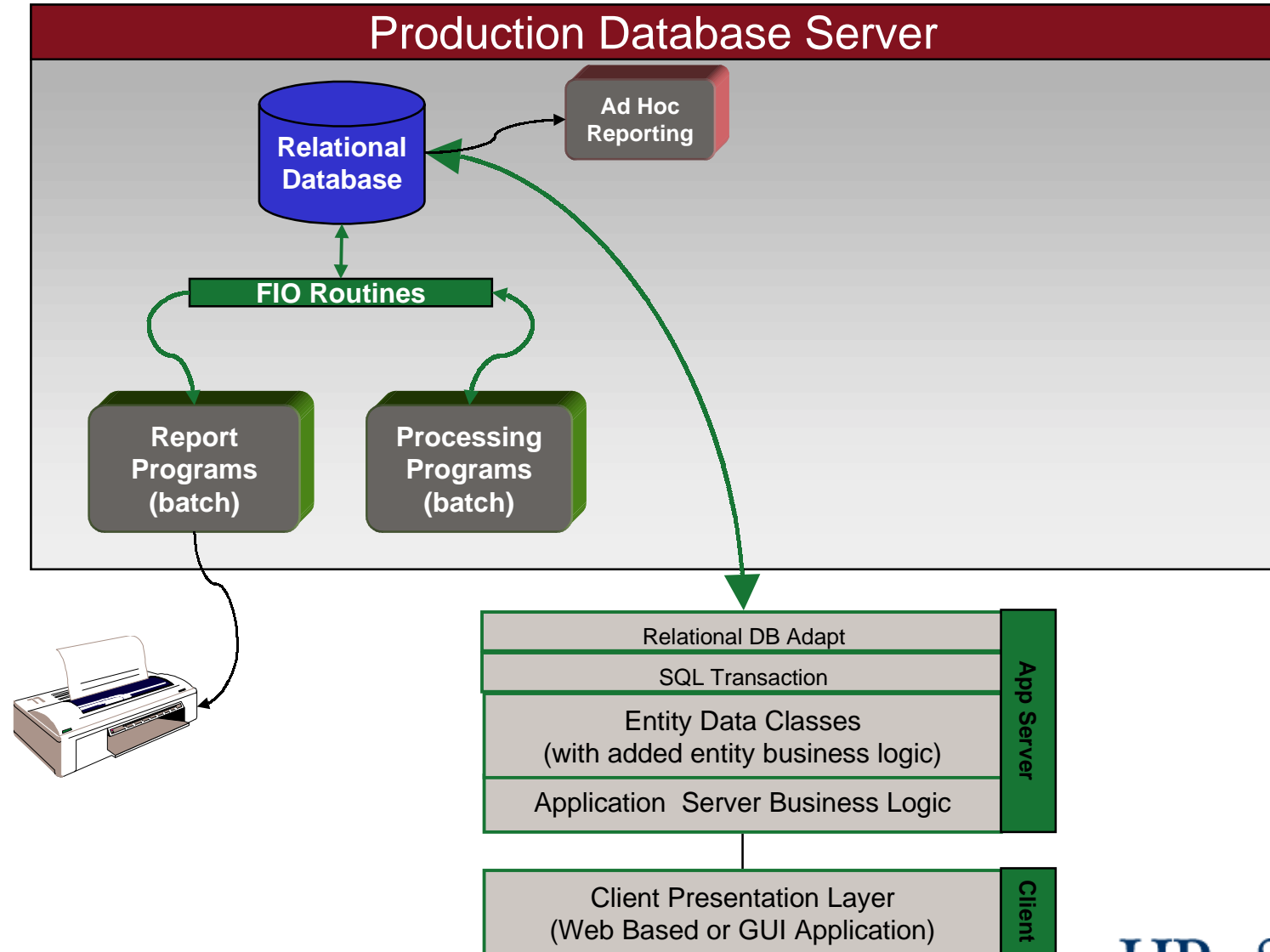
For most COBOL reporting and Processing programs, the FIO routines can be changed to access the new database and the COBOL migrated to a new COBOL compiler supported by the intended Database Server.

Case Study: 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.

Case Study: Final Solution



Questions?

Title: **The Easy Step Approach to Migration**

Presentation: 005

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