

# Legacy System Interaction with Composer Applications

Session 340

Richard Allen  
Texas Instruments



## Overview

- Overview the need for Legacy System Integration
- Define the Legacy Wrapper
- Describe techniques for using Legacy Wrappers
- Discuss how legacy system integration and CBD co-exist



## What is a Legacy System?

- An existing system that performs business functions
- The system can be:
  - Existing system in need of replacement
  - Existing system adequately satisfying business needs
  - Package
- Legacy system can be built with Composer



## Need for Legacy System Integration

### ***Cost***

- Most organizations have portfolio of applications that support the business
- These applications represent the core business of an organization
- These systems represent significant investments
- Cost of replacing these systems would be prohibitive



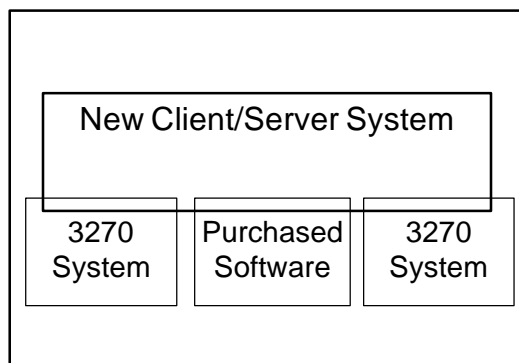
# Need for Legacy System Integration

## *Complexity*

- First generation of client/server systems can be characterized as pilot applications
- The integration with legacy applications is ignored or minimized
- The next generation of client/server applications will address large-scale business problems
- Integrating with legacy applications is a requirement



## Typical Client/Server Architecture



- To be successful, the new application must deal with the overlap of these applications
- Options are:
  - Duplicate the functionality/data
  - Integrate with the applications



## What is the Answer?

- Many alternatives exist:
  - Build a data warehouse
  - Distribute data
  - Develop a new system
- Leverage emerging software/techniques to integrate with legacy applications



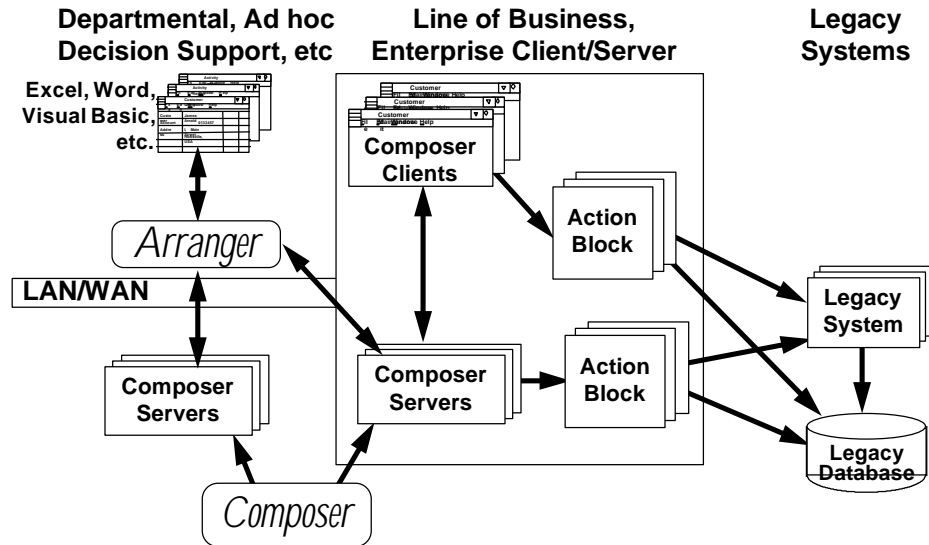
## Legacy Wrapping



- Mechanism of surrounding the legacy application
- Allows for a clean interface so that new systems can access information in the legacy application
- Acts as an API which accesses the legacy system
- Can run on a variety of platforms



## Uses For Legacy Wrappers



© Texas Instruments 1996

9



## Types of Legacy Wrappers

- There are two types of Legacy Wrappers:
  - Data Wrapper
  - Application Wrapper

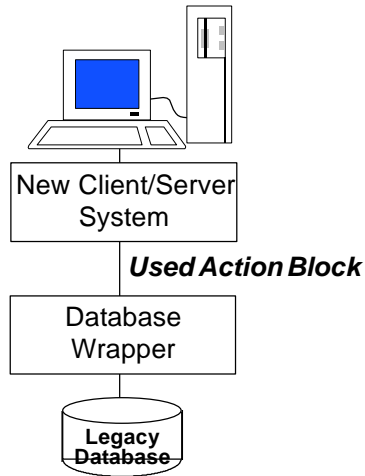
© Texas Instruments 1996

10



# Data Wrappers

## Database Wrapping



- Provides access to the data structures directly
- No legacy code is executed
- The wrapper hides the structure of the legacy data

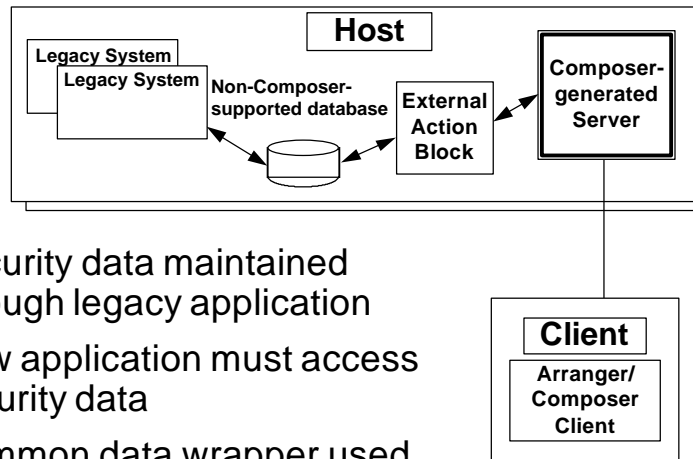


# Data Wrappers

- Data wrappers have three layers:
  - New client/server application
  - Data wrapper
  - Legacy database
- Wrapper can access relational or non-relational data structures
- Wrapper action block can either be internal or external



## Example of a Data Wrapper

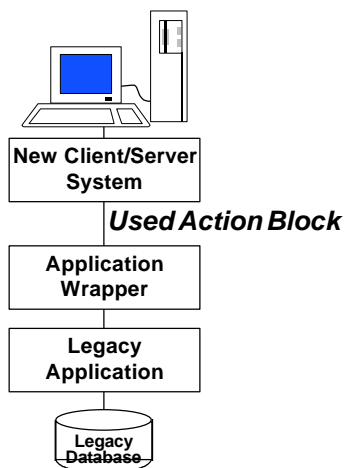


- Security data maintained through legacy application
- New application must access security data
- Common data wrapper used by all applications



## Application Wrappers

### Application Wrapping



- Access the legacy application's logic
- Re-use the logic and business rules of the legacy application
- Insulate the new system from the complexity of the legacy business rules

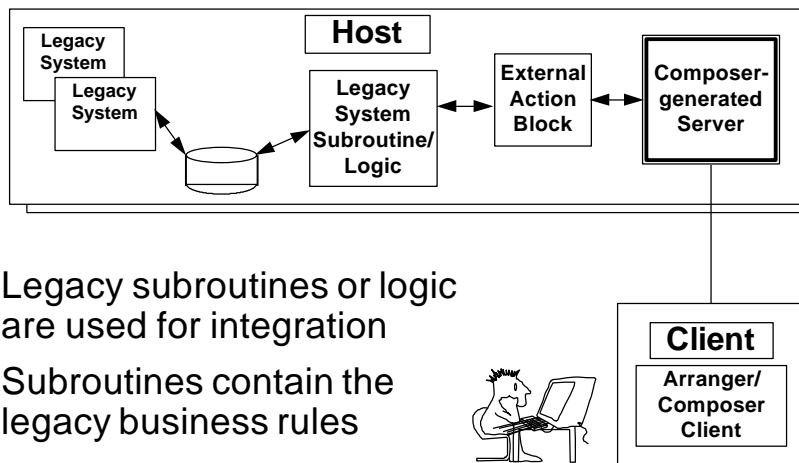


# Application Wrappers

- Application consists of four layers:
  - New client/server application
  - Application wrapper
  - Legacy system
  - Legacy database
- Application logic is used to access the data
- Wrapper process does not need to consider the data format
- Application logic can be accessed through screens, gateways, and subroutines



## Application Wrappers Examples



- Legacy subroutines or logic are used for integration
- Subroutines contain the legacy business rules
- External action blocks call the routines





## Defining Legacy Wrappers in Composer

- Legacy Wrappers are defined as action blocks
- These action blocks can be either internal or external action blocks
- The internal action blocks can only be used for wrapping supported relational databases
- Legacy wrapper can exist at the client or server depending on the technology



## Defining Legacy Wrappers

- One action block should be created for each discrete interface
- Enables the wrappers to be reused throughout a variety of systems
- Facilitates the maintenance of the wrapper action blocks



## Defining Wrappers Example

<i>Legacy Application Functions</i>	<i>Wrappers Defined</i>
Display Customer List Customer Update Customer Create Customer	Display Customer Wrapper List Customer Wrapper Update Customer Wrapper Create Customer Wrapper



## Defining Wrappers—Views

- Import and Export views represent the message to and from the legacy system
- The views can be created using either entity types or worksets
- Entity Views must be implemented in the Technical Design Diagram
- Worksets are recommended for wrappers
  - Do not have entity type requirements
  - Provide for isolation of data



## Steps

- Identify the interface requirements
- Define wrappers to support the interface
- Create a workset that supports the wrapper
- Create an external action block stub in the model
- Generate the EAB stub
- Write the custom code to access the legacy system
- Use the wrapper in the new application



## Approaches for Creating Wrappers

- Defining a standard set of action blocks
  - Higher reuse enabled
  - Used when significant interfacing required
- Creating the action blocks on an as-needed basis
  - Duplicate interfaces may be developed
  - Used when less integration is required
- Combination of both
  - Major interfaces are built
  - Others built as needed



## Legacy Wrappers and CBD

- Legacy Wrappers share common characteristics with CBD operations
  - Provide access to business logic
  - Use abstraction to isolate data
  - Enable reuse
- Legacy Wrappers are defined at a lower level
  - Wrappers are API calls
  - Several wrappers can make up an operation



## Techniques For Legacy Wrapping

- Accessing Composer-supported databases
- Accessing non-Composer-supported databases
- Accessing applications through an Application Programming Interface (API)
- Accessing applications through the screens
- Accessing applications using gateways products

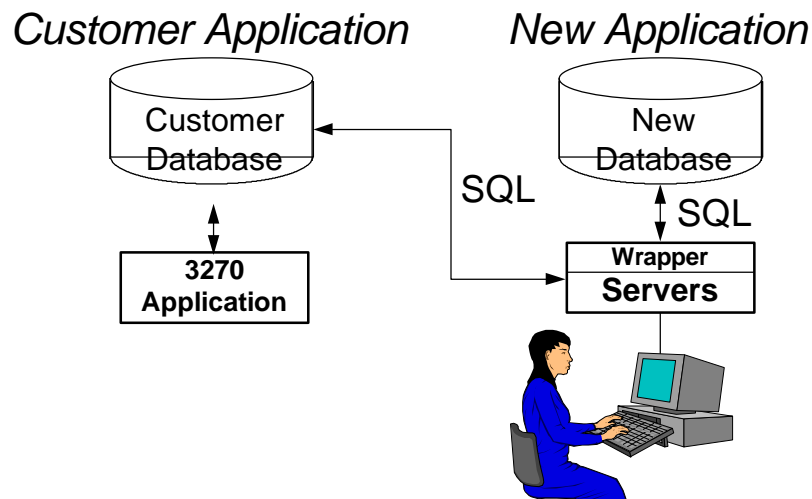


## Technique One: Accessing Supported Databases

- Composer-supported databases are accessed directly
- Example of a data wrapper
- Composer action diagramming statements are used to manipulate the legacy database
- An easy option if the legacy application is built with Composer
- Databases created outside Composer can also be supported with this option



## Technique One: Accessing Supported Databases



## **Technique One: Accessing Supported Databases**

- Provides easy access to the legacy database by the new application
- Legacy database must be modeled
- Provides for forward engineering



## **Special Considerations**

- Model the legacy application
  - Composer-built database - Migrate
  - Non-Composer database - Reverse Engineer
  - Ensure that the Composer definition matches the physical structures

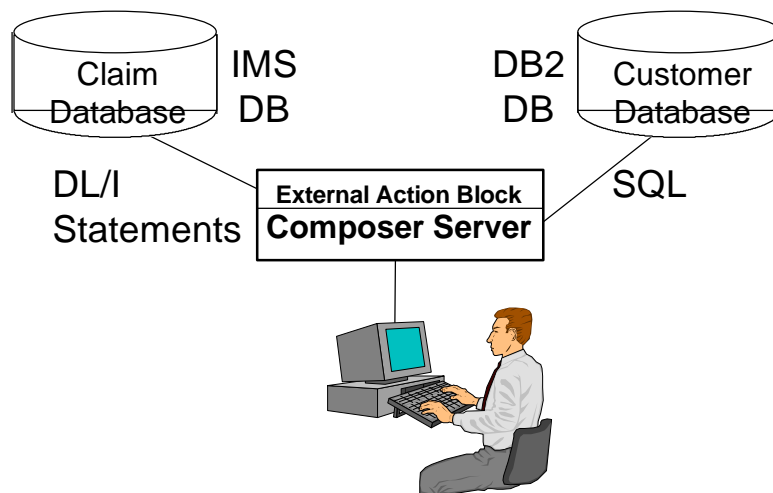


## Technique Two: Accessing Non-Supported Databases

- Non-supported databases are accessed directly
- Example of a data wrapper
- External action block is used to manipulate the legacy database
- Enables access to data structures like:
  - IMS
  - VSAM
  - Flat Files



## Technique Two: Accessing Non-Supported Databases



## **Technique Two: Accessing Non-Supported Databases**

- Provides easy access to the legacy database
- Enables high-performance interfaces to be built
- Similar process to accessing supported databases
- Technique can access data regardless of how it is stored



## **Technique Three: Accessing Applications through API's**

- Legacy application is accessed through the logic of the legacy application
- The legacy application logic can be:
  - I/O routines
  - Sub-Programs
  - API's from vendor packages
- Example of application wrapping



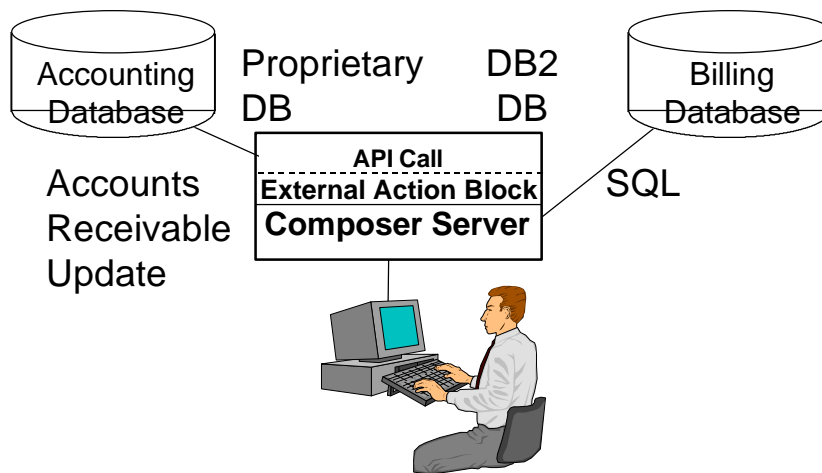


## Technique Three: Accessing Applications through API's

- Enables the wrapping of business logic and data access logic
- Hides the complexity of the legacy business logic and data structures
- Application needs only to understand the interface to the wrapper



## Technique Three: Accessing Applications through API's



## **Technique Four: Accessing Applications through Screens**

- Most organizations have a large population of screen-based applications
- These applications execute under TP Monitors such as CICS or IMS
- Screens typically represent the only knowledge of the existing application



## **Technique Four: Accessing Applications through Screens**

- “Screen Scraper” software is available from a variety of software vendors including:
  - IBM
  - Logitech
  - Platinum
  - And many more...
- These packages typically allow screen definitions to be imported
- These definitions can be manipulated in programs

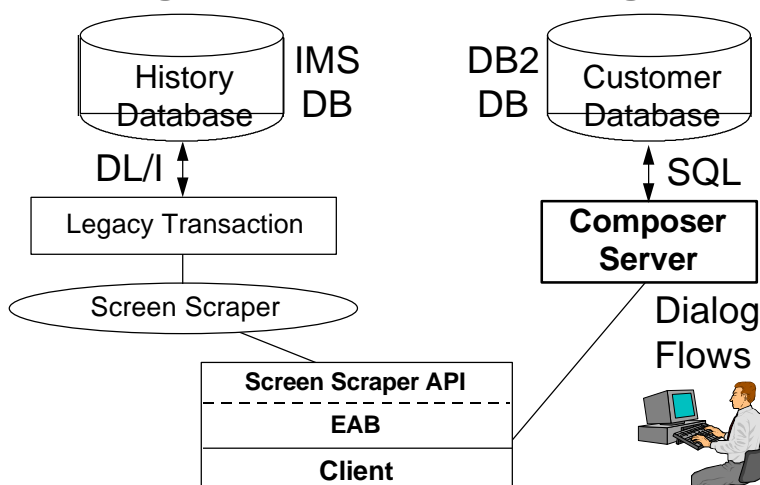


## Technique Four: Accessing Applications through Screens

- Enables wrapping of application logic without modification
- Example of an application wrapper
- All functionality of the application can be accessed
- Developer does not need to know the internals of the application
- Integration through “screen scraping” not as robust as other options



## Technique Four: Accessing Applications through Screens

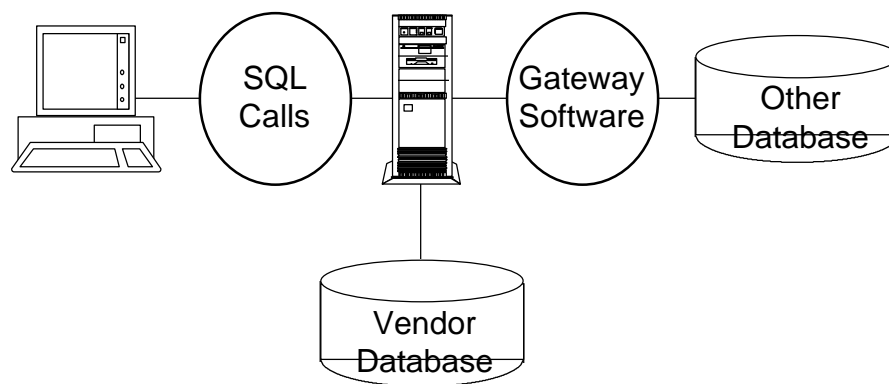


## Technique Five: Accessing Applications through Gateways

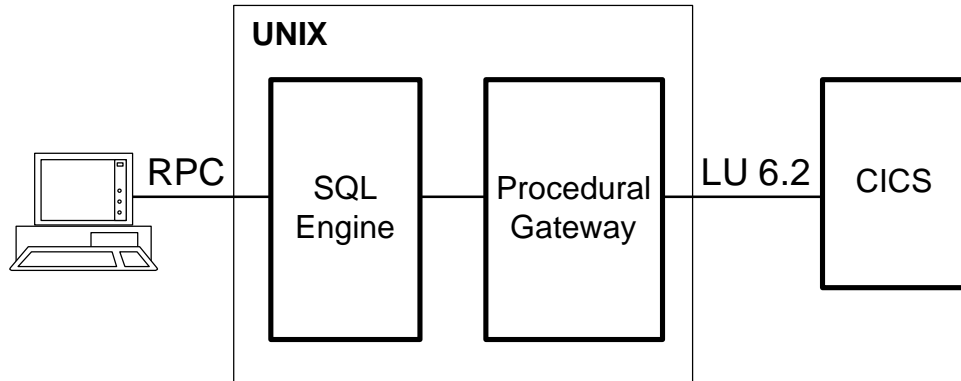
- Gateways are an emerging technology for accessing legacy applications
- Technology provided primarily by the database vendors
- There are two types of gateway products:
  - Data gateway
  - Procedural gateway
- Gateways developed to allow developers common syntax for a variety of legacy applications



### Data Gateways



## Procedural Gateways

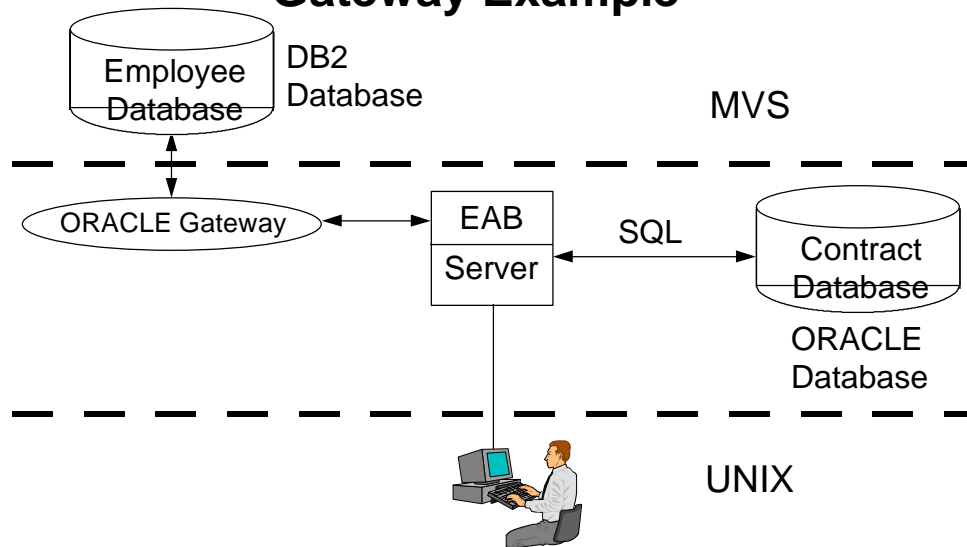


## Gateway Techniques

- Gateway products provide the capability to produce both data and application wrappers
- For data gateways, follow the steps for the non-supported database technique
- For procedural gateways, follow the steps for the API technique
- The gateway products will require setup



## Gateway Example



## Summary



- Mechanism of surrounding the legacy application
- Allows for a clean interface between old and new systems
- Acts as an API which accesses the legacy system
- Documented in TI Methods guide Legacy Application Interworking (Part # 2641080-0001)



## Summary

Many techniques are available:

- Accessing Composer-supported databases
- Accessing non-Composer-supported databases
- Accessing applications through an Application Programming Interface (API)
- Accessing applications through the screens
- Accessing applications using gateway products



## Legacy System Interaction with Composer Applications

Session 340

Richard Allen  
Texas Instruments

