



IBM Software Group

# 2004 WDI / WBIC Customer Conference

*Global Business Transformation*

## WDI Performance and Productivity

Dave Schwartz

**WebSphere.** software



 e-business software

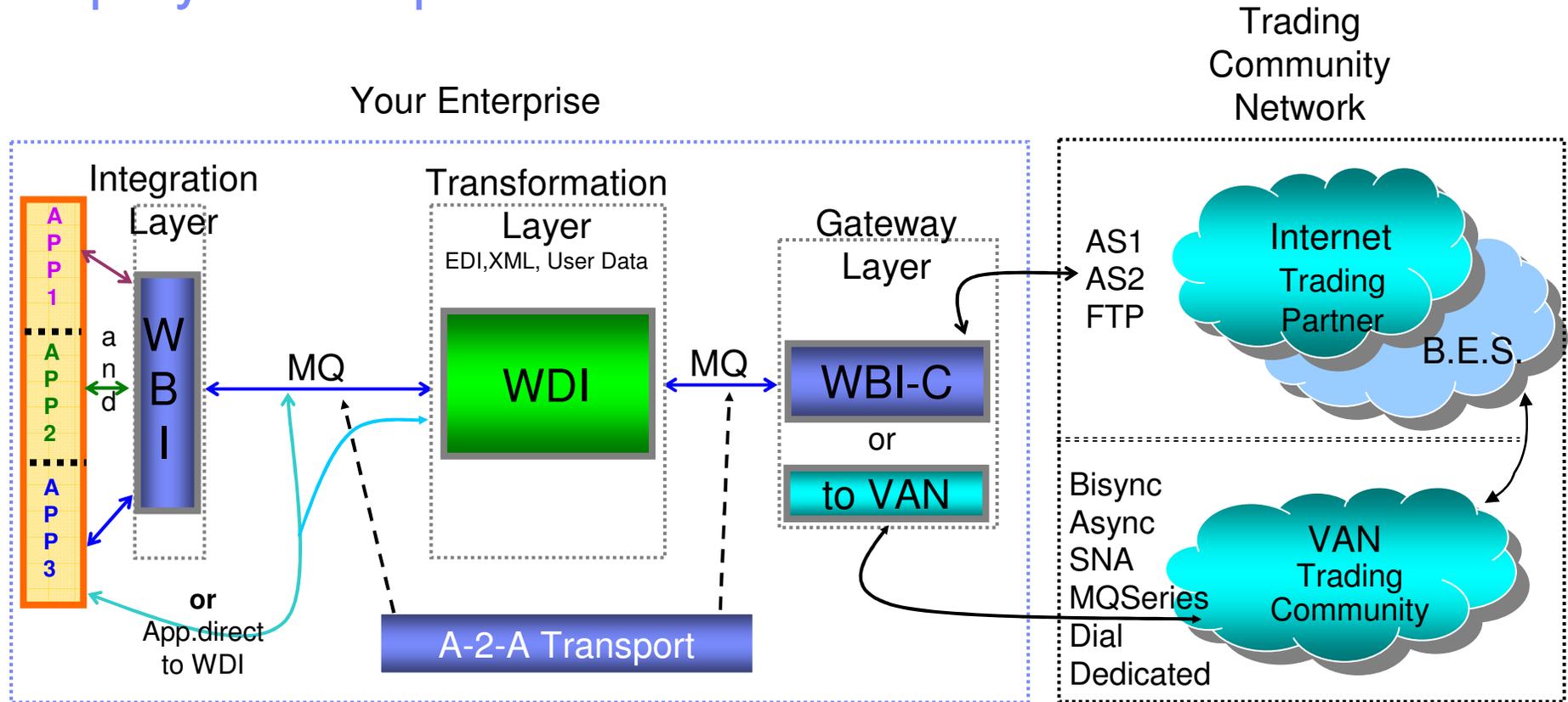
© 2004 IBM Corporation

# Objectives

- Architecture Overview
  
- Performance Considerations
  - Client – Installation considerations
  - WDI Case Study
  - Server – Performance considerations



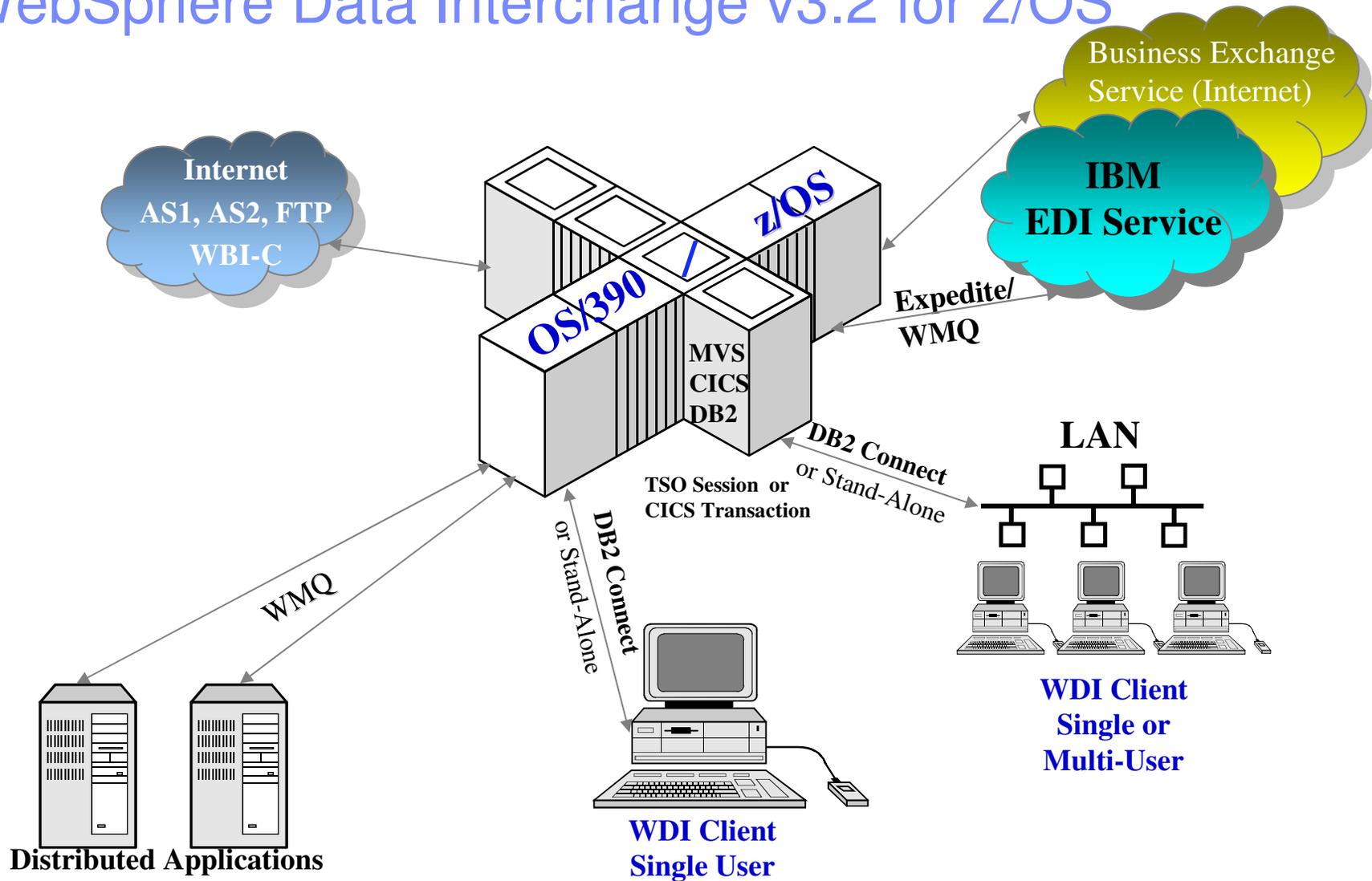
# Deployment Options



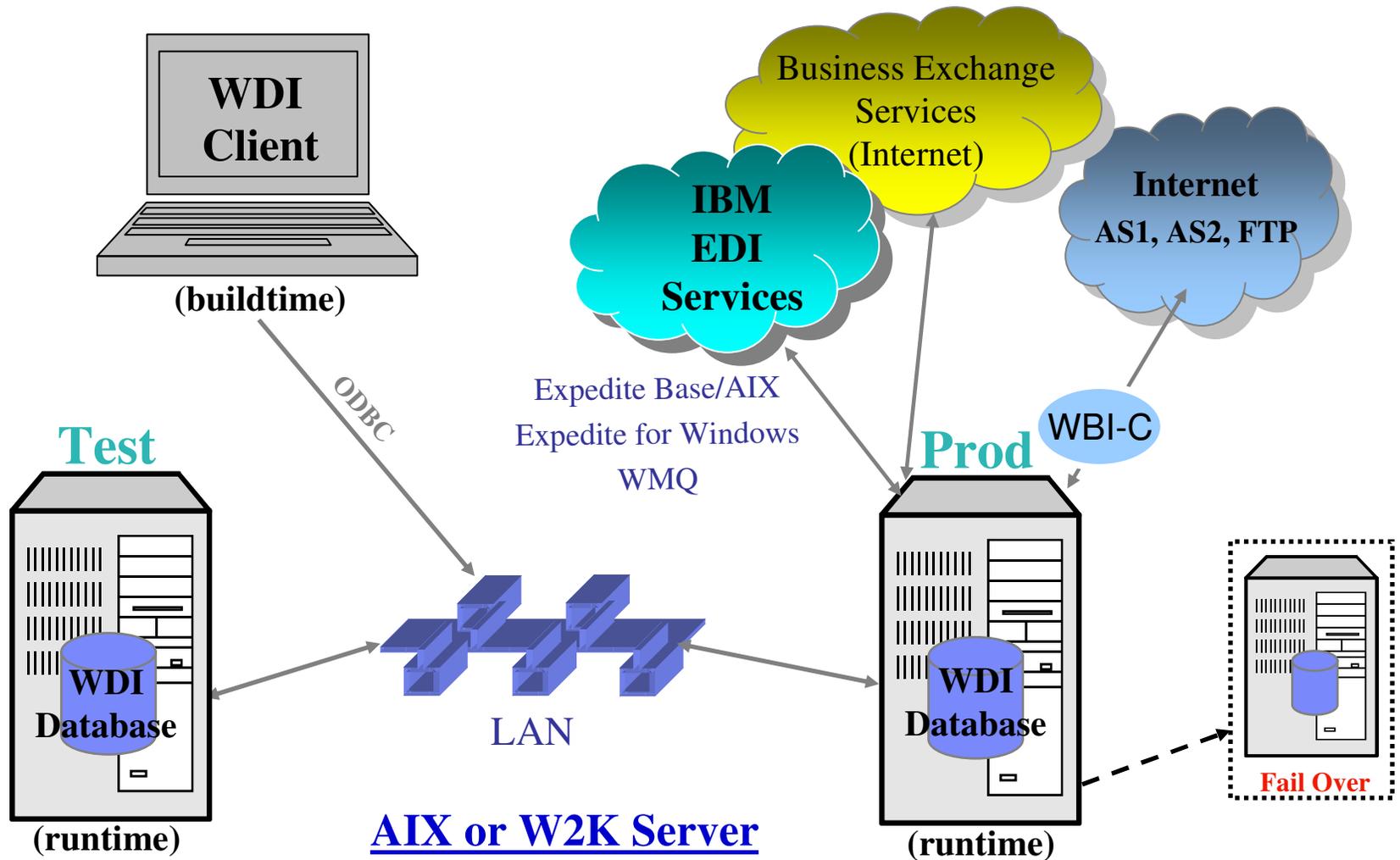
WBI = WebSphere Business Integrator  
 WDI = WebSphere Data Interchange  
 WBI-C = WebSphere Business Integration-Connect  
 VAN = Value Added Network (I.e. IBM EDI Services)  
 BES = IBM Business Exchange Service

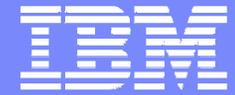


# WebSphere Data Interchange v3.2 for z/OS



# WebSphere Data Interchange for Multiplatforms v3.2





IBM Software Group



WebSphere. software

## WDI Client Infrastructure

**Richard Bennett**

**WDI Client Architect**



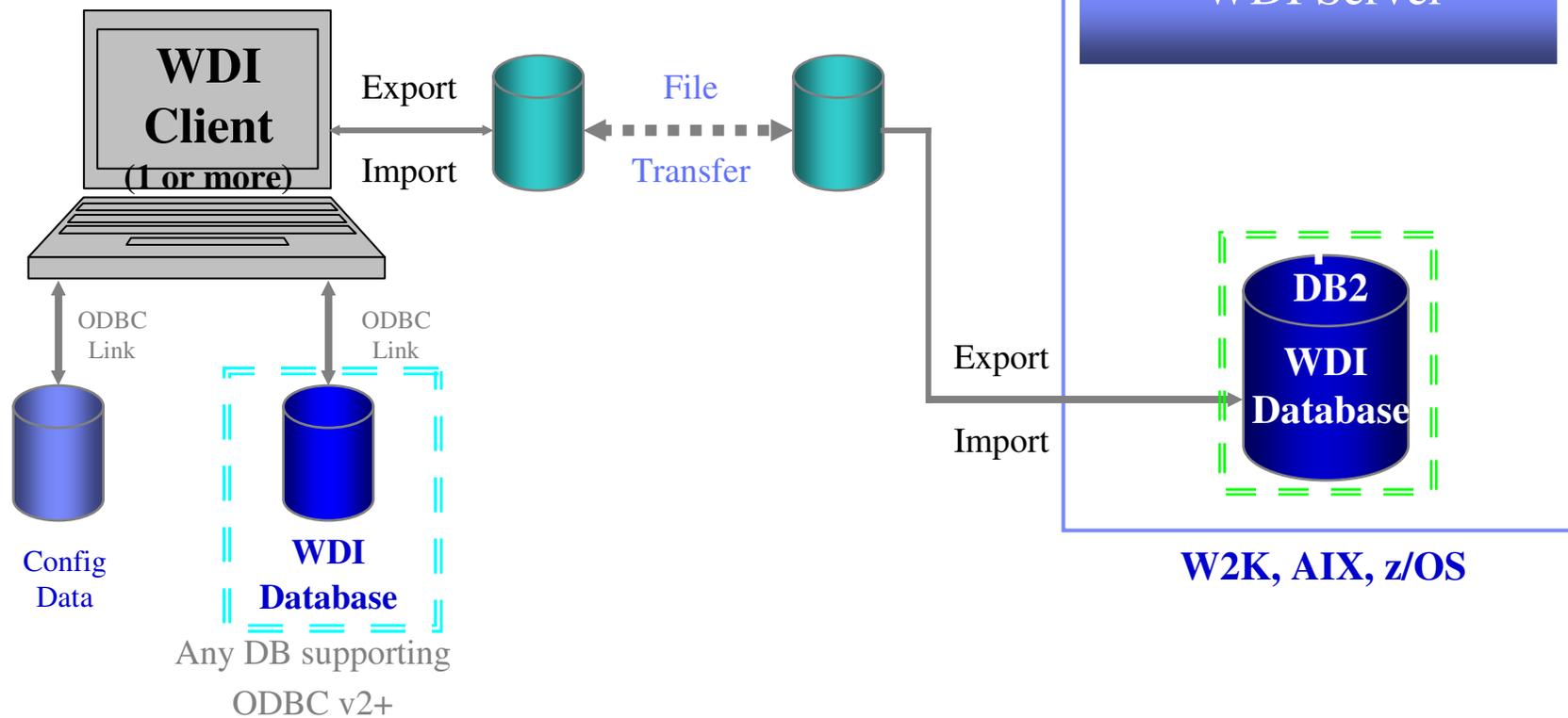
 e-business software

© 2004 IBM Corporation

# WebSphere Data Interchange: WDI Client

Test / Production

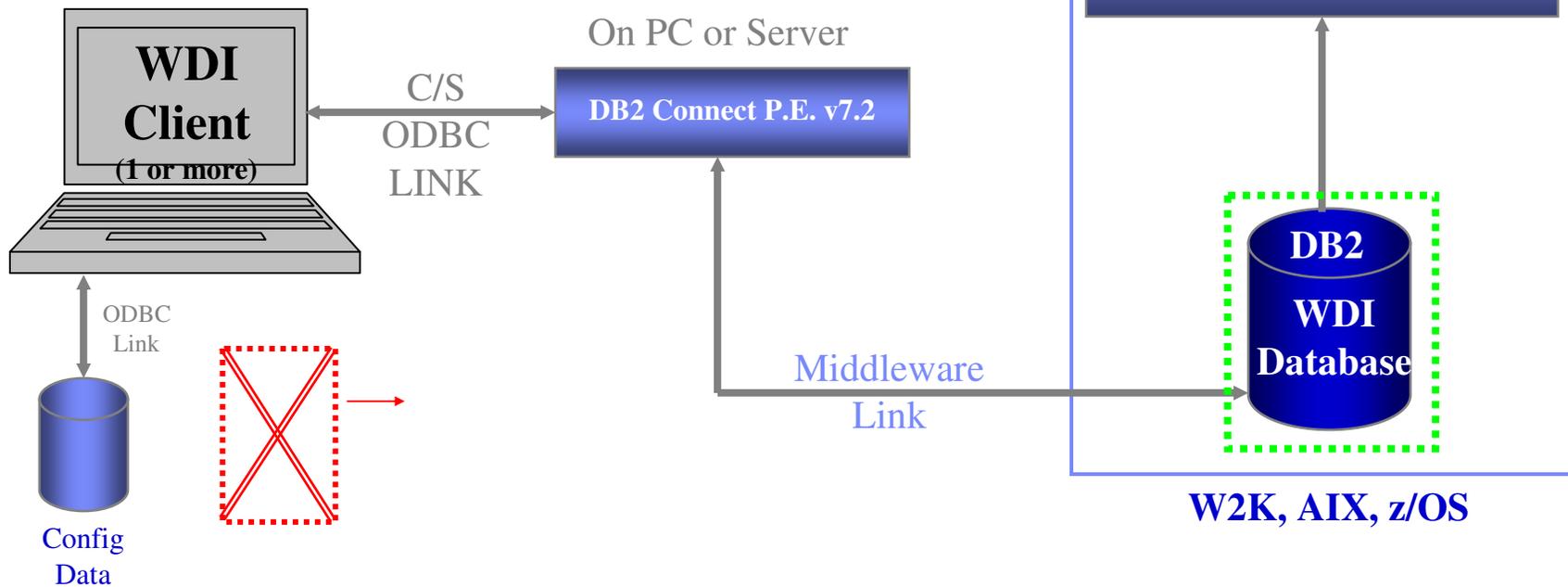
## Stand-Alone mode



# WebSphere Data Interchange: WDI Client

Test / Production

## Client / Server mode

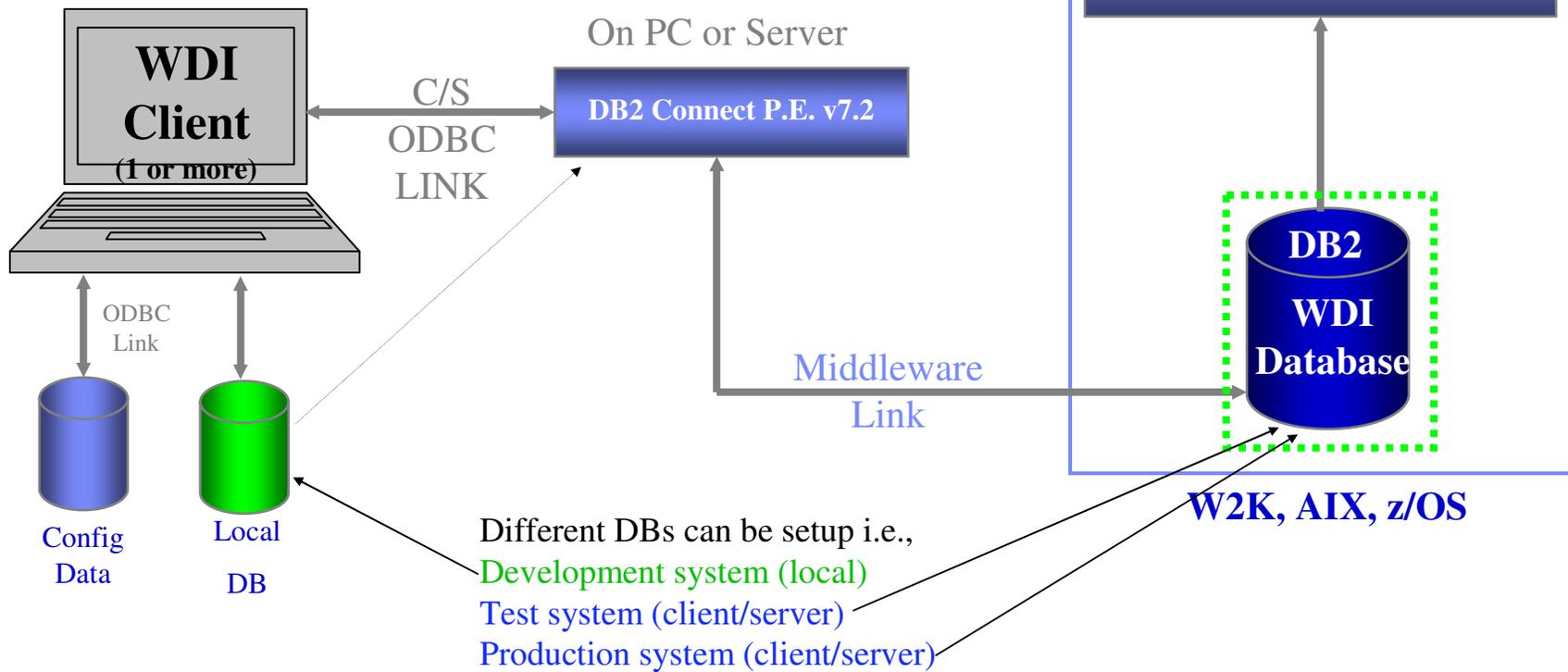


# WebSphere Data Interchange: WDI Client

## Client / Server mode

*& Local DB*

Test / Production



## Additional WDI Client Performance Considerations

- **What else can you do?**

- Improve the performance of the connection to the database
- Create queries that obtain only the information you need to see in a list window
- Do your work involving Maps, Standards, and Data Formats to your local database. If you use Code Lists often in maps, bring those to your local system also





IBM Software Group

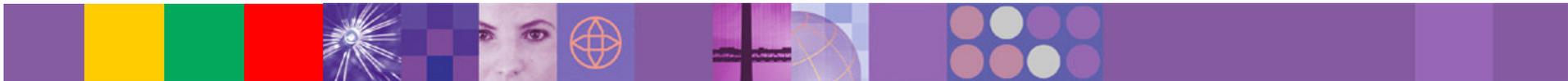


**WebSphere.** software

## WDI Case Study

**Angela Winters-Hill**  
**David Shannon**

**WDI Development**

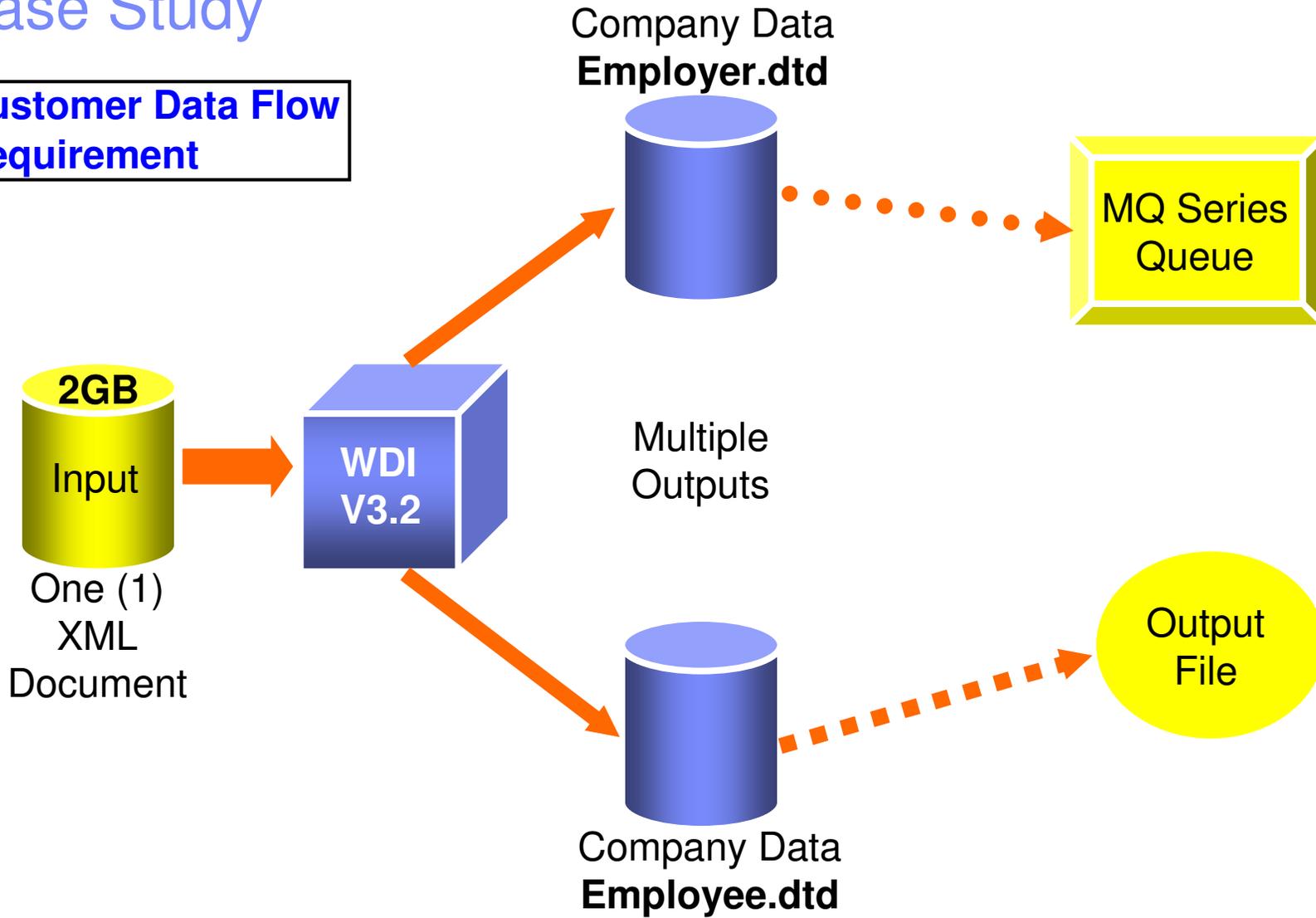


 e-business software

© 2004 IBM Corporation

# Case Study

## Customer Data Flow Requirement



## Case Study

### ■ Customer Data Flow Description

- The Customer receives a single XML transaction with one iteration of a loop for each employee in an organization. Some organizations, like Social Security and Retirement Systems have several million employees on which they report in this transaction. The XML transaction, defined in a Schema, can be 2 GB or more, with an expected growth to 5 GB.
- The Customer required that the single XML transaction be translated into one XML transaction for each employee, and a second, summary XML transaction be created for organization totals.
- The solution will be executed on an AIX machine, using file input. The employee XML transactions will be output to a file, and the summary organization XML transaction is to be sent to a MQ Series Queue.
- The expected time for execution of a 2 GB transaction can be no more than 6 hours.



## Case Study

### ■ Initial Solution

- The initial attempt at a WDI solution was to develop 3 maps. Map 1 (XML->ADF) would create an ADF record for each iteration of the employee loop in the XML transaction. Map 2 (ADF->XML) would treat each ADF record as a separate transaction and create the employee XML transactions. Map2 would MAPCHAIN to Map 3 to output the company XML transaction.
- Memory usage became an immediate concern. The largest input WDI could handle was 1 GB before exhausting a 1.75 GB region on the AIX machine. The 1 GB size was only enabled because a change was made to write the input to a Virtual Array, rather than keep it in memory. Without the change only a 50 MB transaction could be processed without exhausting the 1.5 GB region. The 2 GB limit on Virtual Arrays defined this as a temporary solution.
- The length of time to process the 1 GB file was 15+ hours.



# Case Study

## ■ Solution New Approach

- The second approach revolved around creating a new feature in WDI which had been asked for by several existing customers. That is, the ability to split a single XML transaction into many XML transactions.
- With this feature, an XML->XML map to handle the employee translations was created, and a second map is MAPCHAINED to produce the organization (summary) XML transaction.
- Preliminary executions with this feature revealed unexpected increases in memory usage and very slow processing times.
- The MVS STROBE, Windows IBM Rational Purify tool, and Quantify tool were used to identify memory leaks and intensively used functions.
- When the major memory leaks were corrected, WDI still showed a large amount of memory used, but not a growing amount; runtimes were in the 15-18 hour time range for a 1 GB transaction.



# Server Performance Considerations

- **Performance Issues addressed during case study**
  - Add an index to the EDIDTDHDR table by ROOTELEM; this reduced the CPU time used by the SQL statements accessing the EDIDTDHDR table
  - Add a caching subsystem for WDI objects such as Control Strings, Rules, and Trading Partners. This subsystem allowed subsequent calls to DB2 to be eliminated as long as the same object was being requested. Up to 5 instances of an object can be cached in a PERFORM cycle



## Server Performance Considerations

- On z/OS, eliminate the DB Connection request in each node, this reduced the calls to the STEPLIB and DSNLOAD from 222,000 to 46. Each DB Connect accessed the Call Attachment Facility module (CAF). Up to 5 connects were being done for each
- Add a Node memory pool, so that the AMM would not have to issue "new" and "free" instructions, but could reuse existing, acquired storage.
- Allow reuse of Parser instances; instead of creating a new instance of a parser for each time a parser is used, use a previously initialized, loaded parser



## Server Performance Considerations

- Avoid repeatedly loading DLL's. This was done by not destroying the Message Flow until the message broker is terminated.
- Avoid repeated allocation of output buffer by changing XML\_Writer to keep output buffer intact between calls.
- Avoid repeated allocation of output buffer by changing XML\_Writer to keep output buffer intact between calls.



# Server Performance Considerations

## ■ Case Study RESULTS

- The results of the executions after all the changes:

**on RATSO (AIX), 43p machine, 1.26 GHz**

13 MB took 1 minute 20 seconds

300 MB took 46 minutes

2 GB took 7 hours

**on Windows, 1.6 GHz Pentium processor**

13 MB took 1 minute

300 MB took 22 minutes

2 GB took 2 hours

- Final Customer results on AIX with 2 GB input file:

**?? 1 hour 20 minutes ??**



## Case Study

- **What does this mean to YOU?**

- New feature to split a single XML document into multiple output documents.
- New Utility functions added to filter warning and error messages.
- WDI Server Performance enhancements



## Case Study

- **What does this mean to YOU?**

- New feature to split XML document into multiple output messages.

A single XML document will be split based on a defined XML compound element and reconstructed before the document enters the data transformation message flow. Server CSD14, Client Fix Pack 11

- For more information contact WDI Developer Angela Winters Hill.



## Case Study

- **What does this mean to YOU?**

- New Utility functions added to filter warning and error messages.

Suppress repetitive messages from the print file (RU0003, UT0008, FF0007); suppression is based on the Utility PERFORM keywords:

FILTERMSGs (UTxxxx, Upxxxx, etc). Up to 11

IGNOREINFO(Y/N) - default N

IGNOREWARN(Y/N) - default N



# Questions / Discussion

