Meeting IT Challenges By Transitioning To Service Oriented Architecture & Web Services

Objectives:

- The changing requirements of IT
- Creating business value from Legacy systems
- Reducing cost of change through the successful implementation of web services

Who is ADP?

- **Who is ADP ?**
  - ADP is the largest provider of Human Resources outsourcing services in North America with over 500,000 clients and paying 1 in 5 workers in the private sector.
  - In Canada ADP pays 1 in 4 workers.
  - 44,000 employees worldwide
  - NYSE:ADP > $9B in revenue

The changing requirements of IT

- The traditional view of IT as service organization “Tell me what you need”
- Focus on managing and execution

Changes to...

- Understand and improve the productivity and performance of the organization on a continual basis
- anticipate future business needs and build a long term strategy

The changing requirements of IT

- **SOA can improve the productivity and performance of the organization by:**
  - Creating a re-use culture
  - Improving collaboration
  - Eliminate duplicate spending
- **SOA can help anticipate future business needs and build a long term strategy by:**
  - Creating a single business platform
  - Bringing together heterogeneous systems

Creating a re-use culture

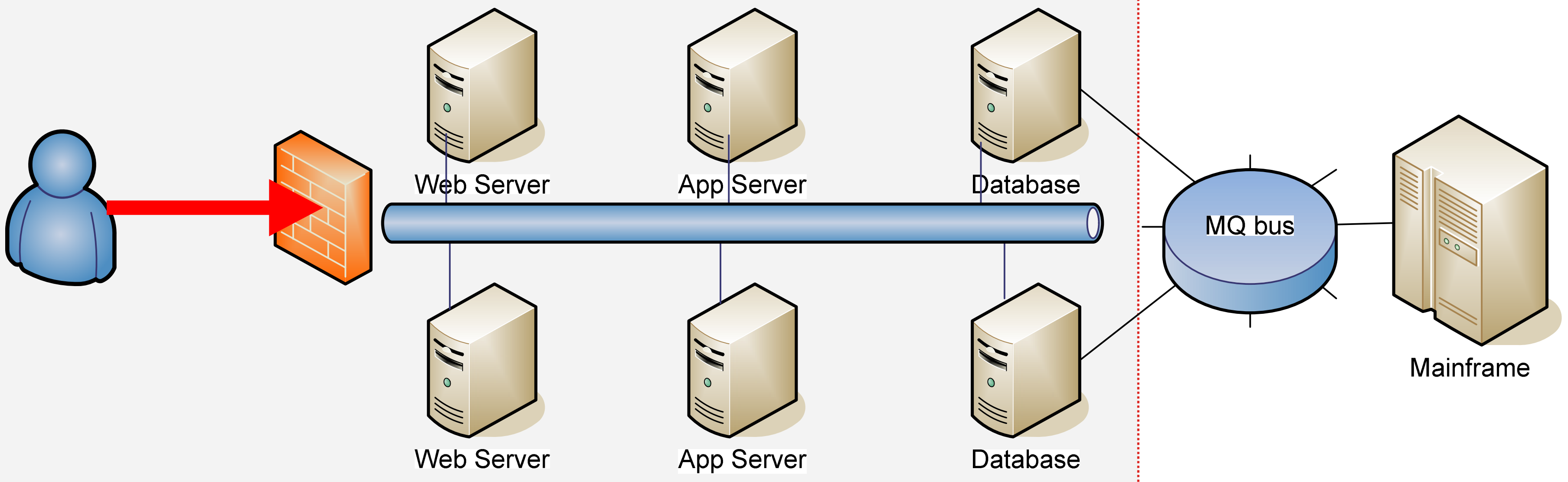
- Typical IT departments are unable to reuse significant past investments
- SOA forces teams to define services first and then interfaces
- Re-use is not guaranteed, but is can bean implicit quality of loosely coupled systems
- Re-use requires an architecture driven approach to development vs. an analysis driven approach

How does ADP use SOA?

- **Use Case 1:**
  - ADP’s core processing technology is a COBOL/MVS application running on a Z/OS mainframe.
  - ADP’s clients want on demand services from web applications
- **Use Case 2:**
  - ADP’s various payroll engines communicate in different ways with the mainframe
  - Cumbersome offline interface for input of customer changes to payroll engine directly.

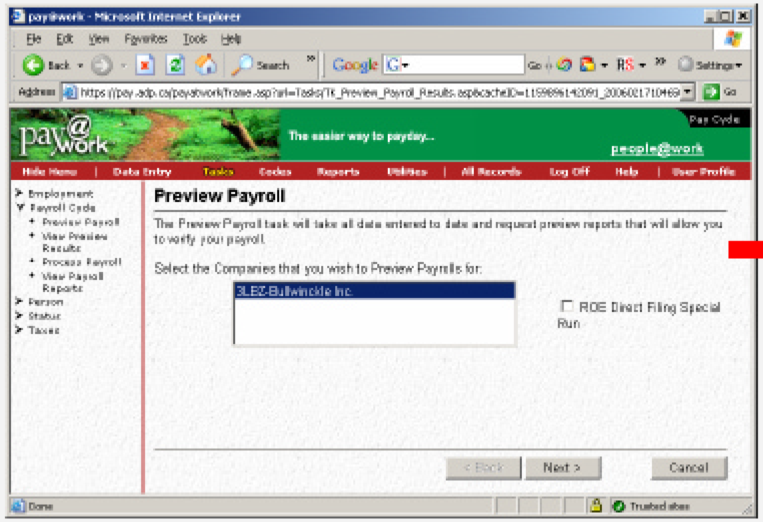
ADP's Core processing

✍ **ADP's core processing architecture links a series of web servers to the mainframe via IBM MQ**



Use Case 1: The Business Problem

- **Client's want payroll previews on demand**
  - SLA in 2005 was 90% in 15 minutes
  - Clients began to call helpdesk after 3 mins
  - Mainframe driven process too slow.



Preview Payroll

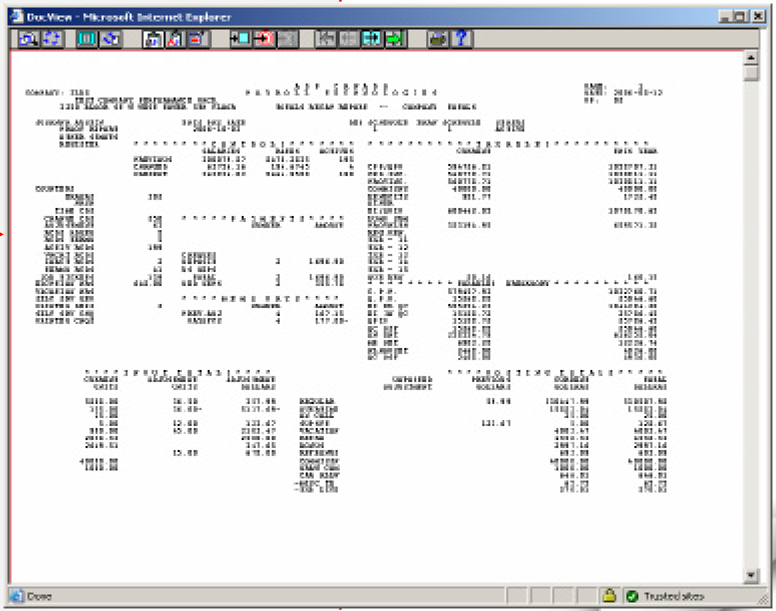
The Preview Payroll task will take all data entered to date and request preview reports that will allow you to verify your payroll.

Select the Companies that you wish to Preview Payrolls for:

3LEZ-Bullwinckle Inc.

☐ ROE Direct Filing Special Run

pay@work - Microsoft Internet Explorer
https://pay.adp.ca/payatwork/frame.asp?url=Tasks/TK_Preview_Payroll_Results.asp&cacheID=1159896142091_2006021710469
The easier way to payday...
people@work
Pay Cycle
Hide Menu | Data Entry | Tasks | Codes | Reports | Utilities | All Records | Log Off | Help | User Profile
Employment
Payroll Cycle
Preview Payroll
View Preview Results
Process Payroll
View Payroll Reports
Person
Status
Taxes
< Back
Next >
Cancel
Done
Trusted sites



UC1: Two possible approaches

- ✍ **Option1: Replicate mainframe calculation engine in web environment**
- ✍ **Option 2: Build real-time interface to mainframe**

Option 1: Rebuild

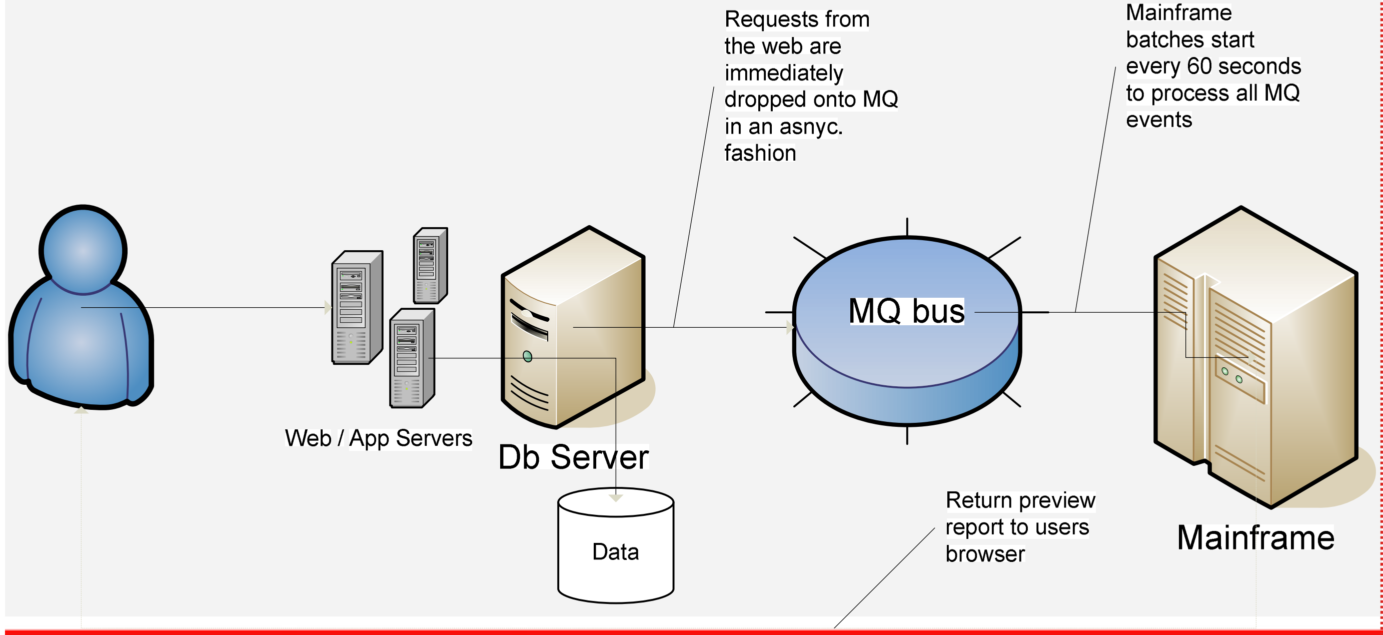
- **Challenges:**
  - Mainframe payroll calculation engine was first developed in 1973
  - Basic calcs were easy, but many clients (3000) had custom calcs created at time of implementation
  - Client setups incorporated many variations inherent in how companies manage payroll for their employees.
  - Risk of discrepancies between web calculations and mainframe
  - Scope of change
- **Opportunities:**
  - Modernize legacy codebase to address deficiencies
  - Potentially lower development cost

Option 2: Leverage

- **Challenges:**
  - Current process not optimized for real-time requests
  - Throttling of process required to manage mainframe MIP usage
  - Higher development cost on Mainframe
- **Opportunities:**
  - Ability to extend mainframe functionality to web
  - Reduce duplication of effort
  - Streamline customer process

Solution

- ✍ **Leverage mainframe assets to extend existing mainframe calc engine to web**
- ✍ **Re-write existing preview process on mainframe to support 1 minute batches**
- ✍ **Re-write MQ connector in Web environment to immediate deliver files to MQ.**


Requests from the web are immediately dropped onto MQ in an asnyc. fashion
Mainframe batches start every 60 seconds to process all MQ events
MQ bus
Web / App Servers
Db Server
Data
Return preview report to users browser
Mainframe

The question is always going to be where is the most efficient and cost effective place to introduce change and the answer isn't always going to be the newer technologies

Other benefits

- ✍ **Leveraging legacy code means using each environment to the best of its capability...**
- ✍ **Web: real-time – don’t build artificial constraints**
- ✍ **MQ: Allows for async implementations.**
- ✍ **Mainframe: Performs best when working on batches of data not single requests**

Delivering business value

- ✍ Overall timeframe for end to end reduced from 90% in 15 minutes to 99.99% in 5 minutes and average of 1.5 minutes
- ✍ Reduced capacity requirement on mainframe for previews by 300% by changing from 1 at a time to batches
- ✍ Total cost of solution much lower by leveraging legacy assets

Use Case 2: The Business Problem

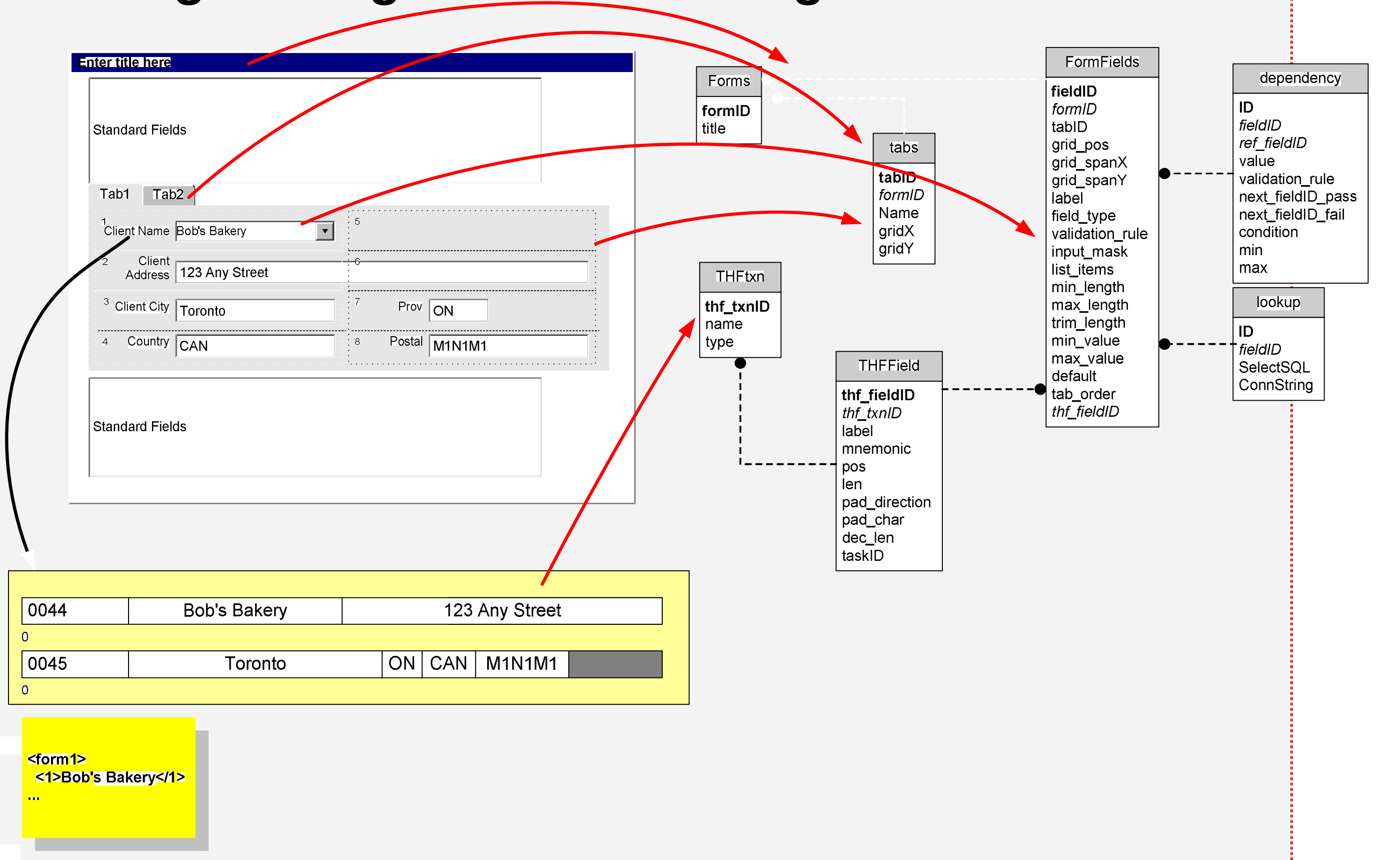
- ✍ Legacy interface to payroll engine was offline batch mode approach to creating 80-byte records
- ✍ Lack of internal controls for authorization and workflow
- ✍ Need for better visibility of pending changes
- ✍ Need for single , common solution to access mainframe

UC2: Approach

- **Create new web-site for generating, storing and maintaining transactions**
- **Turn 80-byte mainframe interface to re-usable service oriented interface**
  - Treat as a service protocol
- **Leverage existing MQ transport channels**

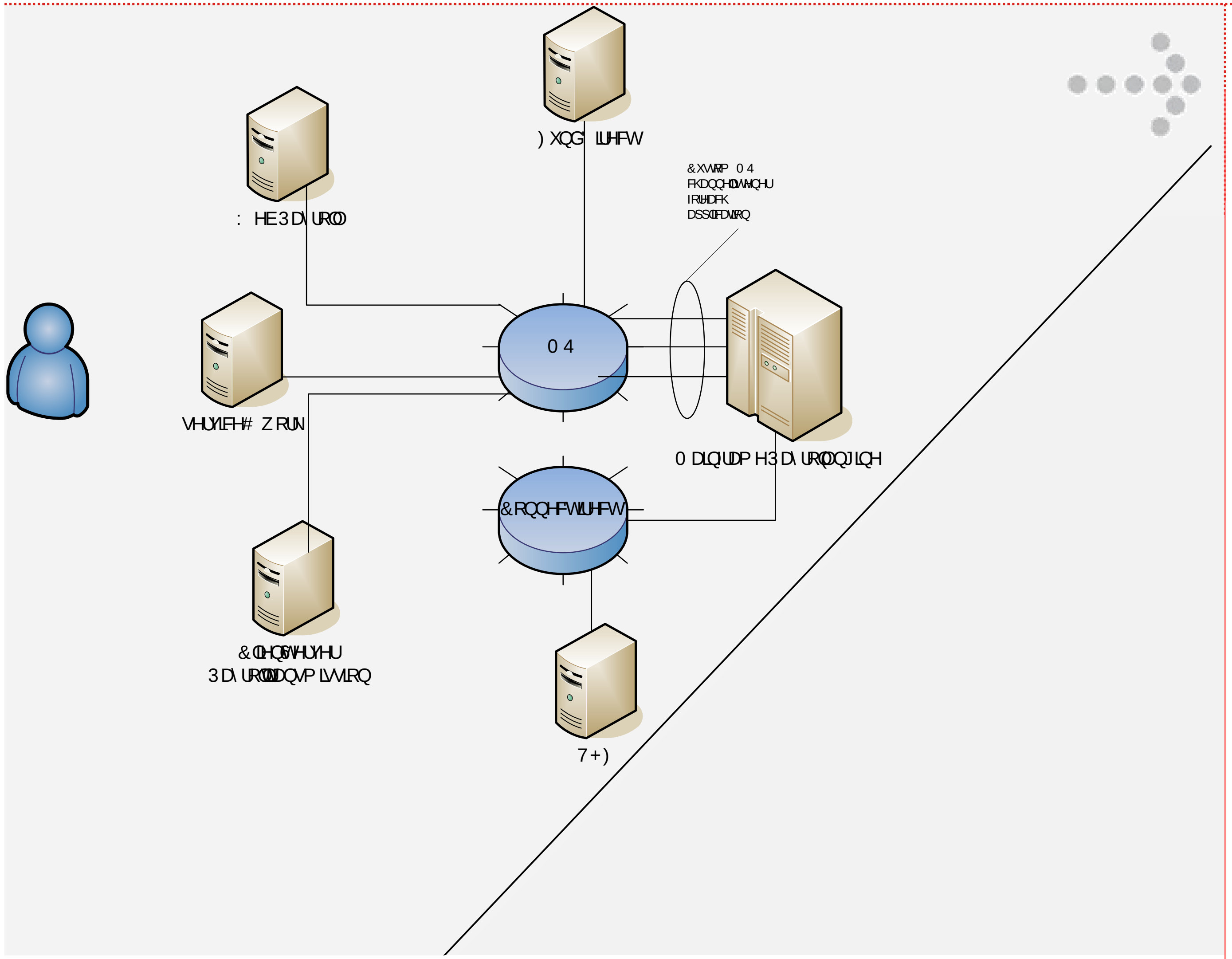
Solution

✍ **Reuse of the 80 byte mainframe record format allowed us to develop a meta-data approach to building storing and maintaining transactions**



```
<form1>
  <1>Bob's Bakery</1>
...
```

Simplified Architecture



Other benefits

- ✍ This approach doesn't ignore the value hidden in legacy mainframe architectures
- ✍ Using a meta-data approach future proofs the applications and allows us to re-purpose the transactions as web forms or web-services
- ✍ A single interface to the mainframe breaks down application boundaries and allows for a single feature development across multiple product sets.

Conclusions

- ✍ **MQ vs. HTTP for accessing legacy systems adds guaranteed delivery and ease of implementation**
- ✍ **Enabling legacy codebase through Web Services increases value of investments**
- ✍ **Creating re-usable real-time entry points to legacy applications reduces cost of change by eliminating tight coupling.**

Questions