Improving the performance, efficiency and TCO of web-enablement for IMS applications

A Presentation to Virtual IMS Connection
By SysperTec
June 8, 2010
Agenda

- Status of IMS Applications
- Web-Enabling Technology
- IMS Web-Enabling Architectures
- VIRTEL Product Overview
- Improving IMS Applications’ Web Access
- Improving IMS Applications’ User Interface
- Improving IMS Applications’ Web Integration
Many IMS applications are in “containment” mode:

- Focus has shifted from legacy/host to web technology
- Legacy application code is seen as old, obscure, complex, inflexible, and brittle → not worth new investments
- Legacy application experts are aging and in diminishing numbers, while younger web application developers are in growing numbers
- Host budgets and staff are frozen or even reduced
- Minimal developments or investments are allowed on host: mostly maintenance
- New developments or investments are with server-based web applications
Leveraging Challenge

Yet IMS applications are here to stay, and IMS data keeps growing ➔ How to:

- Get more out of IMS applications with minimal investment, effort, change, and risk?
- Seamlessly integrate legacy IMS and web technologies
- Maximize the value and ROI of IMS application modernization + get earlier ROI?

- Leave IMS applications unchanged
- Fully reuse existing business logic and 3270 interface
- Integrate web-enabling technology ➔ Make IMS applications look & behave like web applications
Web-Enabling Technology

Host/Web Integration

Create interactive, bi-directional connections between IMS and web applications → web services that consume IMS transactions through open-standard languages like XML or HTML.

Web Access
Extend IMS applications “as-is” to web browsers

Web Services
SOAP

Targeted Automation
HTML + XML

User Interface Modernization
HTML + JavaScript

Reusable Services
XML

Develop modern Web 2.0 user interface that combines IMS’ 3270 data flows with new AJAX functionality and JavaScript widgets to make IMS applications look and behave like Windows applications.
IMS Pre-Web Architecture

Leverages VTAM strength, performance, efficiency, scalability:

- Simple, strong, fast, efficient (low CPU usage), scalable

Leverages legacy business logic (BL) embedded within 3270 (IMS/DC + MFS) presentation layer:

- Proven, stable, amortized
IBM IMS/Web Architecture

No use of VTAM except for 3270 emulation

Multi-layer multi-component architecture:

- Complex, potential for performance, CPU consumption, scalability issues

No use of legacy BL embedded within 3270 (IMS/DC and MFS) presentation layer

Need to develop new (redundant?) BL using CPU-intensive Java

- Time, cost, risk
VIRTEL IMS/Web Architecture

1. Leverages VTAM strength, performance, efficiency, scalability:
   - Simple, strong, fast, efficient (low CPU usage), scalable

2. Leverages legacy BL embedded within 3270 (IMS/DC and MFS) presentation layer:
   - No new BL development = simple, fast, low cost, low risk
VIRTELE - Internal Architecture

DB2 VSAM Access
PDF Generation
RACF Logon Authentication

Function Calls

LE languages: COBOL, etc
XML + JavaScript + HTML + etc
VIRTELE-supplied functions
HTML + JavaScript + VIRTELE tags
Assembler code: not user-accessible

VTAM
Native VTAM Relay
Process Management Scenarios
Protocol Conversion Engine
HTTP/S or SMTP Server
Presentation Management

3270 Data Streams
Installation Parameters
HTML + JavaScript Presentation Templates

VTAM
LU Pool

TCP/IP
Web
HTML / JavaScript Web Pages

VIRTELE Screen Redesigner

© SysperTec Communication
One web-enabling product/technology covering the entire spectrum of web-enabling technology:

**IMS User Interface**
- VTAM Interface
- HTTP

**IMS Transactions**
- IMS Transactions
- HTTP + SMTP Server

**IMS Data**
- IMS Data
- HTTP + SMTP Server

**z/OS Platform**

**WEB CLIENTS**
- Web-to-Host Access
- UI Modernization

**HTML, XML, WEB SERVICES...**

**SMTP**
- THIN-CLIENTS: standard JavaScript capable browsers (no applets)
- EMAIL, SMS...

**HTTP**
- XML, HTML, WEB SERVICES...

**HTTP**
- WEB APPLICATIONS
- WEB SERVERS
- WEBSHERE

© SysperTec Communication
Protocol Conversion Engine

• Host-based dynamic protocol converter that combines native VTAM interface with HTTP and SMTP servers:
  – VTAM Relay ➔ VTAM connection between VIRTELE and 3270 applications in synchronous connected mode
  – HTTP and SMTP Servers ➔ TCP/IP HTTP connection between Web browsers and VIRTEL in asynchronous disconnected mode

• Developed in Assembler (No CPU-intensive Java)

• Runs as started task (or in batch) on mainframe

• Uses compiled presentation templates (HTML + JavaScript) and VIRTEL scenarios to guide the protocol conversion between 3270 and web

=> Shortest possible distance (instruction “path”) between IMS applications and the web
• Extend IMS applications “as-is” to web browsers:

  - Serve IMS 3270 screens as web pages to web browsers
  - No change to IMS 3270 screens: same look and feel
  - No change to IMS business logic, application code, or TM

• Quick fix that can evolve into UI modernization
Extend IMS applications “as-is” to web browsers

3270 screen appearance and ergonomics in a web browser

3270 application codes

Navigation by PFK
3270 Emulation

- Most pervasive web access solution

- Attachmate EXTRA, Zephyr PASSPORT, Hummingbird, IBM PCOMM, BlueZone Emulation, MicroFocus RUMBA, Century tinyterm, OpenConnect WebConnect, …

- Issues and limitations:
  - Requires fat client and/or server or middleware
  - Can’t evolve into UI modernization or integration
  - Can’t support custom features like DBCS conversion, language handling, selective encryption, etc
  - Performance and efficiency (host memory consumption) therefore scalability limitations
**Improved Architecture**

• Typical 3270 emulation architecture:
  - Requires opening special port in host firewall
  - Fat Clients:
    - Applications
    - Heavy browser (Plugin, Applets)

• Need to install-support-upgrade SW/HW outside mainframe
• Support limited to specific client OS/browsers
• VPN required

• VIRTEL thin-client web access architecture:
  - Uses standard TCPIP port
  - Thin Clients:
    - Standard JavaScript compatible browser
    - Windows 7, 64-bits, Mac, …

• Nothing to install-support-upgrade outside mainframe
• Universal client OS/browsers support: Windows 7, Mac, etc
• VPN not required
Improved Deployment

• Nothing to install, support, or upgrade on clients

• No servers between host and clients
  ➤ Simple and instant (2-hour) deployment
  ➤ Universal client support: Windows 7, 64 bits, Mac, etc
  ➤ Low support requirement-effort-cost
Improved Security

• [Standard IP security: SSL encryption (AT-TLS), SSO, PROXY, etc]

• Asynchronous/disconnected mode: closed when inactive

• Uses standard TCP/IP port (TN3270 requires opening a special port in the host firewall)

• VPN not required

• Can reuse RACF—… rules for web-to-host access control

• Support terminal control

• Support selective encryption
Improved Performance & Efficiency

• All-Assembler no-Java no-layer single-component ➔
  Shorter instruction path between IMS applications and
  web ➔ High performance

• VTAM sessions pool is shared between many web users
  ➔ Lowers host memory usage

• Smaller host “footprint” (Lower host memory and CPU
  cycles usage) ➔ Reduced host costs

• Improved scalability (performance x efficiency) for very
  large deployment
Global Deployment Support

- Automated DBCS/EBCDIC conversion
- Automated language handling
- Global deployment in countries like China, Japan, or Russia that use DBCS workstations
User-Friendliness & Flexibility

• User Friendliness:
  – Single Sign On and/or integration with RACF-ACF2-TSS → eliminate redundant username/password management
  – Integration with existing web portal → consistency

• Flexibility:
  – Selective encryption capability
  – Terminal control
  – Evolve from web access to user interface modernization
Web Access - Case Study

Global, high-volume, Web-to-Host gateway to PSA
Peugeot Citroën’s mainframe application network:

- IMS 3270 application on private SNA network had to be accessed globally from standard web browsers
- Presentation or navigation had to remain unchanged to avoid user retraining → simply automate the 3270-HTML conversion
- Benchmarked up to 16,000 simultaneous client connections using a single VIRTEL instance; in production since 2003, serving an average of 3,000 simultaneous customer connections, 150 countries, 24x7 operation
- VIRTEL used to automate character set conversion between local language input and mainframe application. (i.e. DBCS character conversion for Chinese and Russian markets to French UTF8/EBCDIC application)
User Interface Modernization

- Develop modern Web 2.0 user interfaces that make IMS applications look and behave like Windows applications (Screen-based modernization):

<table>
<thead>
<tr>
<th>Combine existing IMS 3270 data flows with:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• AJAX technology (Add new asynchronous functionality)</td>
</tr>
<tr>
<td>• JavaScript widgets (Accelerated development and universal browser portability)</td>
</tr>
</tbody>
</table>

- No change to IMS business logic, application code, 3270 data flows, or TS

- Short or mid term (temporary?) “tactical” solution
No reloading of static page components

**BOOKING**

Journey case validation

<table>
<thead>
<tr>
<th>Case</th>
<th>Case status</th>
<th>Journey type</th>
</tr>
</thead>
<tbody>
<tr>
<td>155155</td>
<td>Opening</td>
<td>CRUISE</td>
</tr>
</tbody>
</table>

Transport: ✔️

### Outward

<table>
<thead>
<tr>
<th>Country</th>
<th>Departure</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>NEW YORK-JFK</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 2007</td>
<td></td>
</tr>
</tbody>
</table>

### Return

<table>
<thead>
<tr>
<th>Departure</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>US</td>
</tr>
<tr>
<td>Male-Hulule</td>
<td>JFK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Voyage: NALDIVES CRUISE - ATOLLS FOREST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum deposit: 750</td>
</tr>
</tbody>
</table>

**CLIENT DETAILS**

Sales assistant: 007

<table>
<thead>
<tr>
<th>Amount</th>
<th>Qty</th>
<th>Total (incl.tax)</th>
<th>Nettotal</th>
<th>Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1300</td>
<td>4</td>
<td>6264</td>
<td>9218</td>
<td>28</td>
</tr>
</tbody>
</table>

**Option buttons**

- Package tour
- Flight only

**Graphical calendar**

**Calculated information**

**Push buttons**

- Validate
- Disconnect
PFK navigation replaced by tabs
Input area with AJAX suggest
Check-box with mask-able area
Input area with AJAX auto-complete

HTML table built with mainframe data

Check-box with mask-able area

Automatic generation of PDF, CSV, XML document

Downloaded or sent by email
### UI Modernization Alternatives

- 10 times less risk, time, and cost to modernize than to redevelop, replace, or rehost

<table>
<thead>
<tr>
<th></th>
<th>Redevelop</th>
<th>Replace</th>
<th>Front-End</th>
<th>Rehost</th>
<th>UI Modernization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cost</strong></td>
<td>$ Millions</td>
<td>$ Millions</td>
<td>$ 100Ks</td>
<td>$ 100Ks</td>
<td>&lt; $100K</td>
</tr>
<tr>
<td><strong>Timeframe</strong></td>
<td>Years</td>
<td>Years</td>
<td>Months-Years</td>
<td>Months</td>
<td>Weeks-Months</td>
</tr>
<tr>
<td><strong>Risk</strong></td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
</tr>
</tbody>
</table>
UI Modernization Benefits

• Address growing users’ dissatisfaction with outdated 3270 “green screens” and demand for web applications

• Make IMS applications easier to market/sell to new users

• Allow access to legacy application by untrained users: esoteric application codes → plain-English selections

• Modernize UI of licensed IMS product or application: no access to source code, or source code modification would cancel product support

• Modernize UI of “frozen” application: source code is lost or so complex/brittle that it is too risky to alter
UI Modernization Benefits (Cont.)

• Use AJAX technology to add new functionality through Web 2.0 user interface without changing legacy IMS application code

• Provide Windows-like UIs when no time and budget to develop new web applications

• Can’t wait new web application, which is years and $ millions away, to offer Web 2.0 experience to users:
  – UI modernization only few weeks-months and $10Ks away
  – Temporary solution until web application is launched
UI Modernization – Case Study

- Large EU bank:  
  ![CA]  

- Develop new Web 2.0 user interface in parallel with DB2 migration:
  
  - In compliance with Corporate web presentation standards
  
  - Without changing any CICS application code
  
  - Without disrupting or interrupting end-user service or DB2 migration
  
  - With equal or better performances than native and emulated 3270
Host/Web Integration

• Create interactive bidirectional connections between IMS and web applications, for example web services that consume IMS transactions through open-standard languages like XML or html:

  • Consume IMS transactions through existing IMS 3270 interface and without developing redundant business logic
  • No change to IMS business logic, application code, 3270 data flows, or TM

• Long term “strategic” solution
Leverages VTAM strength, performance, efficiency, scalability:

- Simple, strong, fast, efficient (low CPU usage), scalable

Leverages legacy BL embedded within 3270 (IMS/DC and MFS) presentation layer:

- No new BL development = simple, fast, low cost, low risk
Performance & Efficiency

• Leverage from VTAM simplicity, strength, performance, efficiency, and efficiency

• All Assembler no-Java no-layer single-component architecture

• Improve performance and/or throughput of IMS/web connections

• Reduce CPU cycles consumption → Reduce host costs
Reduced Development

• Reuse existing IMS/MFS application interface (3270 data streams) including embedded BL to build web services

• Avoid developing new BL that is redundant with BL already in 3270 UI

• Solution’s implementation:
  – Simpler
  – Easier to debug
  – Lower effort and cost
  – Faster implementation
• Same product/technology covers all web-enabling needs:
  – Web access → UI modernization → host/web integration

• Same product/technology covers all integration needs:
  – AJAX, Exchange of large data blocks (Over 32K), Automated conversion between COMMAREA and XML (or HTML, CSV, JSON, etc.), Web Services, SOAP server, PHP server, MQ server, automated text/picture conversion for DBMS GUI, host/WebSphere® connector, etc
  – Can address “special” integration requirements that IBM or other integration products can’t address
Web-Enablement Continuum

Where does screen-based modernization end, and where does web integration start?

Web Access
Access IMS “green screens” from web

Targeted Automation
HTML + XML

Web Services
SOAP

User Interface Modernization
HTML + JavaScript

Reusable Services
XML

Add new functionality using AJAX technology

Mashups

- Consolidate multiple 3270 screens/transactions onto one webpage
- Dynamically generated dropdown lists
- Suggest, auto-complete, etc
- Calls to custom or existing functions: access host data via SQL, generate PDF document, etc
- Send emails
- Activate host transactions (IMS, CICS, etc) or web services
- Integrate with web applications → Mashups
Questions?