



Modern Software Development Tools on OpenVMS

Meg Watson
Principal Software Engineer



- Modern Development Tools for OpenVMS
 - NetBeans/Distributed NetBeans
- Modernizing Existing Applications
 - Web Service Integration Toolkit
- Questions

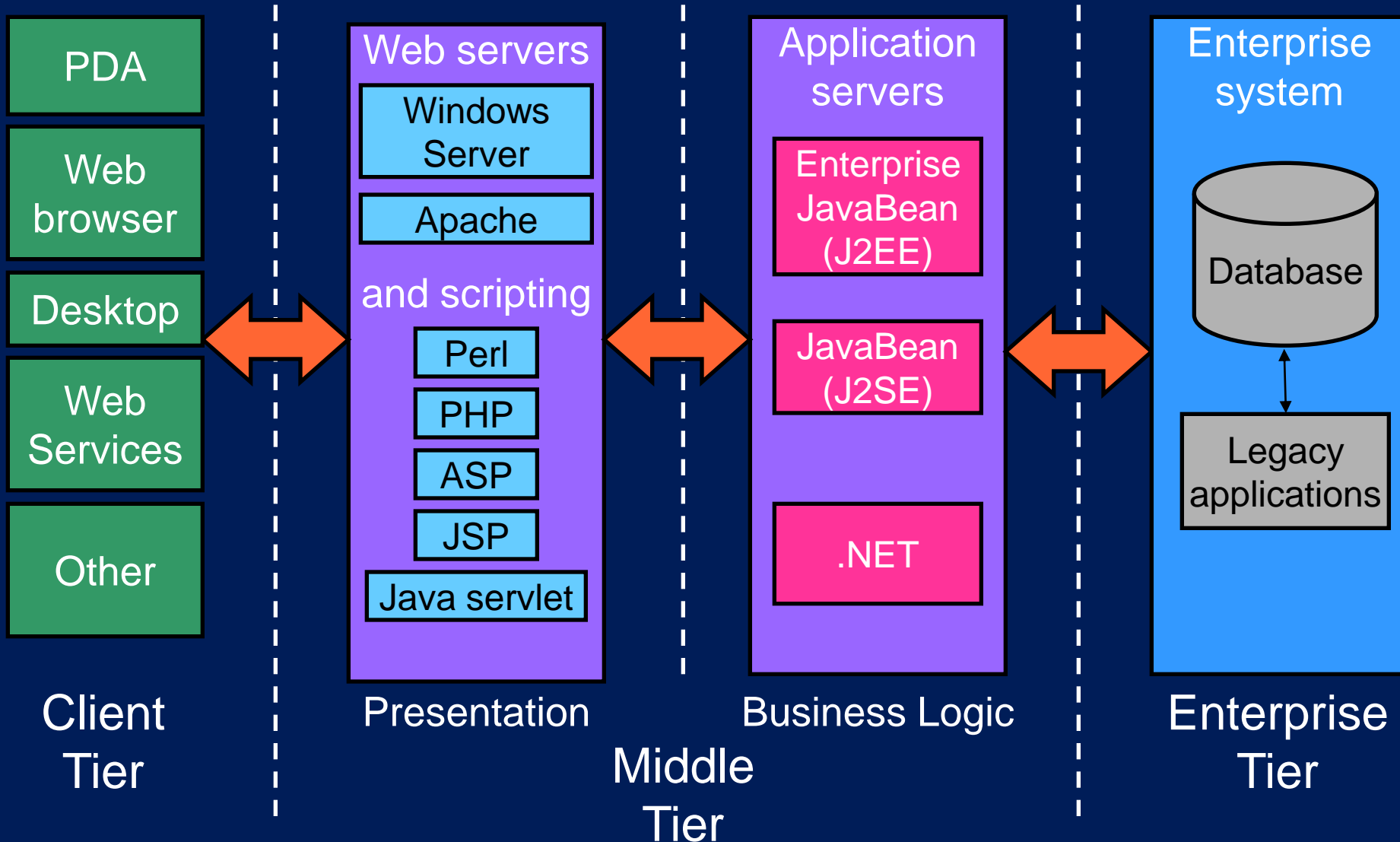
What is Modern Software Development?



- Object-Oriented Languages
 - Java, C++, C#, etc
- Current Popular Software Designs
 - Web-based
 - Application server based
 - JSPs, Servlets, EJBs, Web Services
 - Service oriented architecture
- Distributed Applications
 - Heterogeneous execution environments
- Modern development methodology
 - Agile methods
 - Work well with object-oriented languages

Goal: Use the same tools to develop all the pieces!

Today's Environment



What is NetBeans?



- Sun-Sponsored Open-Source Integrated Development Environment
- 100% Java – runs anywhere there's a JVM
- Current version is 5.5.1, with 6.0 in beta
- Feature-rich: drag-n-drop GUI creation, excellent editing, JSPs, Web services, excellent debugging, profiling, etc.
- Extensible via plug-ins
- Positioned as platform and IDE
- Competes with Eclipse...the leap frog effect

Why Not Use Eclipse?



- We joined the NetBeans open-source project in 2001, long before Eclipse was open-sourced
- NetBeans has the most advanced GUI building tools available in any open-source Java IDE
- NetBeans comes with many features built-in, Eclipse is more spartan by default
- Eclipse certainly has more available plug-ins and bigger corporate backing, but there is lots of disagreement over which is “better”
- NetBeans just won the 2007 “Bossie” for the Best IDE

NetBeans Major Features



- Advanced source code editor
- Drag-n-drop Swing GUI editor
- Web application development - JSPs, JSF, Struts, JSTL, plus debugging
- Enterprise Development – easily create EJB3 and JAX-WS web services
- Built-in application servers – Tomcat, Glassfish
- Complete Ant support, including debugging
- Version control support - CVS, Subversion, VSS and others
- Wizards, code generation and management tools
- Lots more...



Menus

Toolbar

NetBeans IDE 5.5 - Switcher

File Edit View Navigate Source Refactor Build Run CVS Tools Window Help

Projects Files Runtime

- AnagramGame
- AntLab
- HelloWorld
- Switcher

Files Tab

Navigator - SwitchUI

Members View

- SwitchUI()
- initComponents()
- jButton1ActionPerformed(ActionEvent evt)
- jButton2ActionPerformed(ActionEvent evt)
- main(String args)
- jButton1 JButton
- jButton2 JButton

Filters: [Icons]

SwitchUI.java [Unknown*][Unknown*] x meg.xml [Unknown][Unknown] x build.xml [AntLab] x Welcome x

Source Design [Icons]

```
/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
Generated Code

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
Color fg = jButton1.getForeground();
Color bg = jButton1.getBackground();
jButton1.setForeground(bg);
jButton1.setBackground(fg);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
System.exit(0);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
```

Editor

Output

Output window

Integrated Source Code Editor



Code completion

Automatic error checking

Syntax coloring

Abbreviations

Code formatting

Code folding

Shortcuts for formatting, commenting, uncommenting

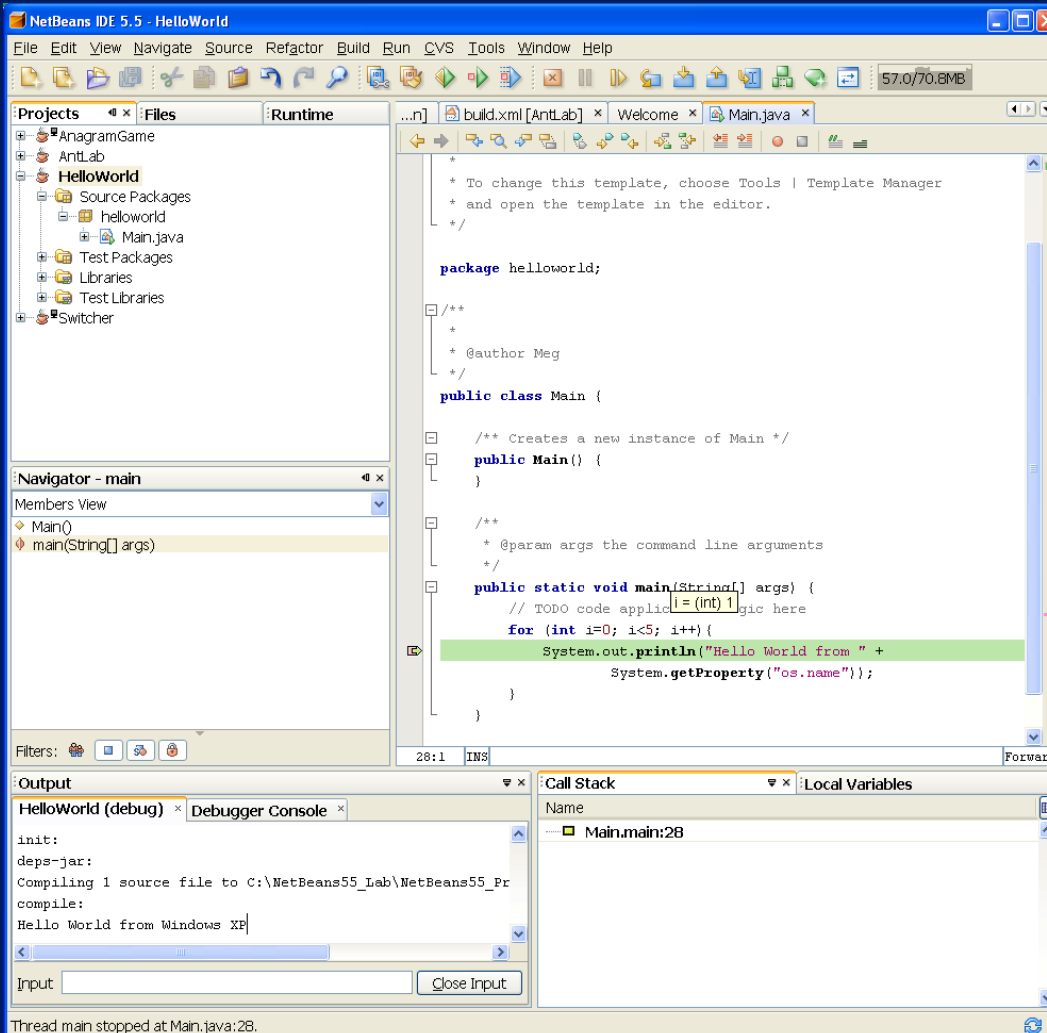
Multiple buffers

Jump list, Bookmarks

The screenshot shows an IDE window with several tabs: SwitchUI.java, meg.xml, build.xml, and Welcome. The main editor displays Java code with syntax highlighting. A yellow highlight is on the first comment block. A 'Generated Code' section is visible. The code includes methods for button actions and a main method. The status bar at the bottom shows '23:4' and 'INS'.

```
/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
Generated Code
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
Color fg = jButton1.getForeground();
Color bg = jButton1.getBackground();
jButton1.setForeground(bg);
jButton1.setBackground(fg);
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
System.exit(0);
}
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
```

Integrated Debugging



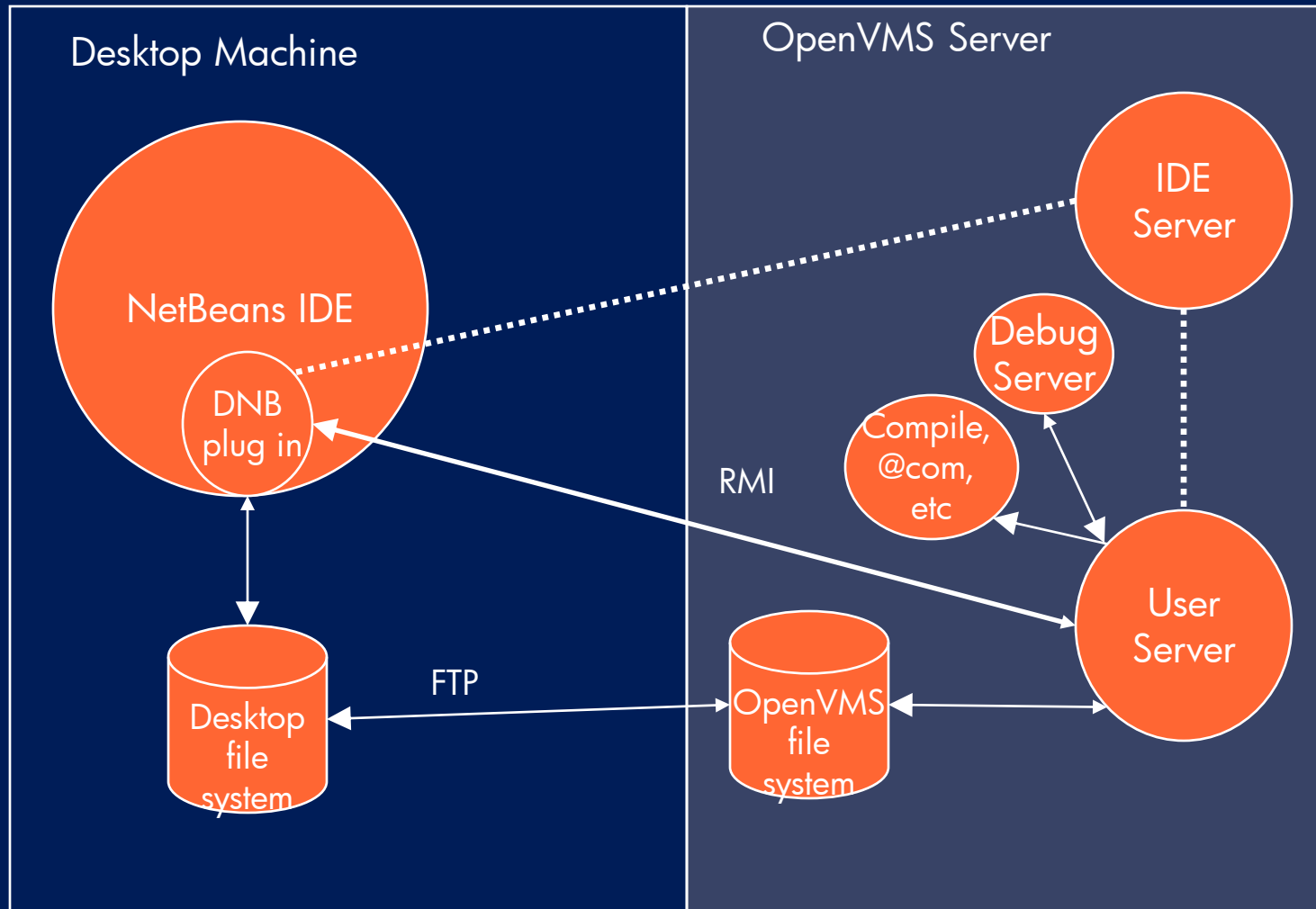
- GUI debugger, integrated with editor
- Debug Java, JSPs, Ant scripts
- Watches for variables, class instances, and expressions
- Breakpoints
- Connection to remote JVM via JPDA
- Multiple thread support
- Tracing and stepping
- Variable evaluation using mouse-over

Distributed NetBeans

- Allows any desktop (Windows, Linux, HP-UX, etc.) to be used to do remote OpenVMS development
- NetBeans runs on the desktop, with our plug-in installed (Windows, MACOS, Linux, HP-UX...)
- Uses FTP or SMB* to access the files on your OpenVMS machine
- Syntax highlighting and formatting, remote compilation, error navigation, remote execution, and remote debugging* for Java, C/C++, Cobol, Fortran, Pascal, and Basic
- EDT keypad!
- Remote execution for DCL command procedures, .EXE files, Ant scripts, Bash shell scripts, MMS files

* In upcoming FT3

The Anatomy of Distributed NetBeans



Distributed NetBeans Version Info



- Distributed NetBeans V5 FT2 available now
- FT3 is coming soon with
 - 3GL debug support
 - SMB support
 - Remote execute for .EXE files
 - Foreign command support
 - Extended password protections
 - Bug fixes
 - more

Modernizing Existing Applications: The Legacy Application ~~Problem~~

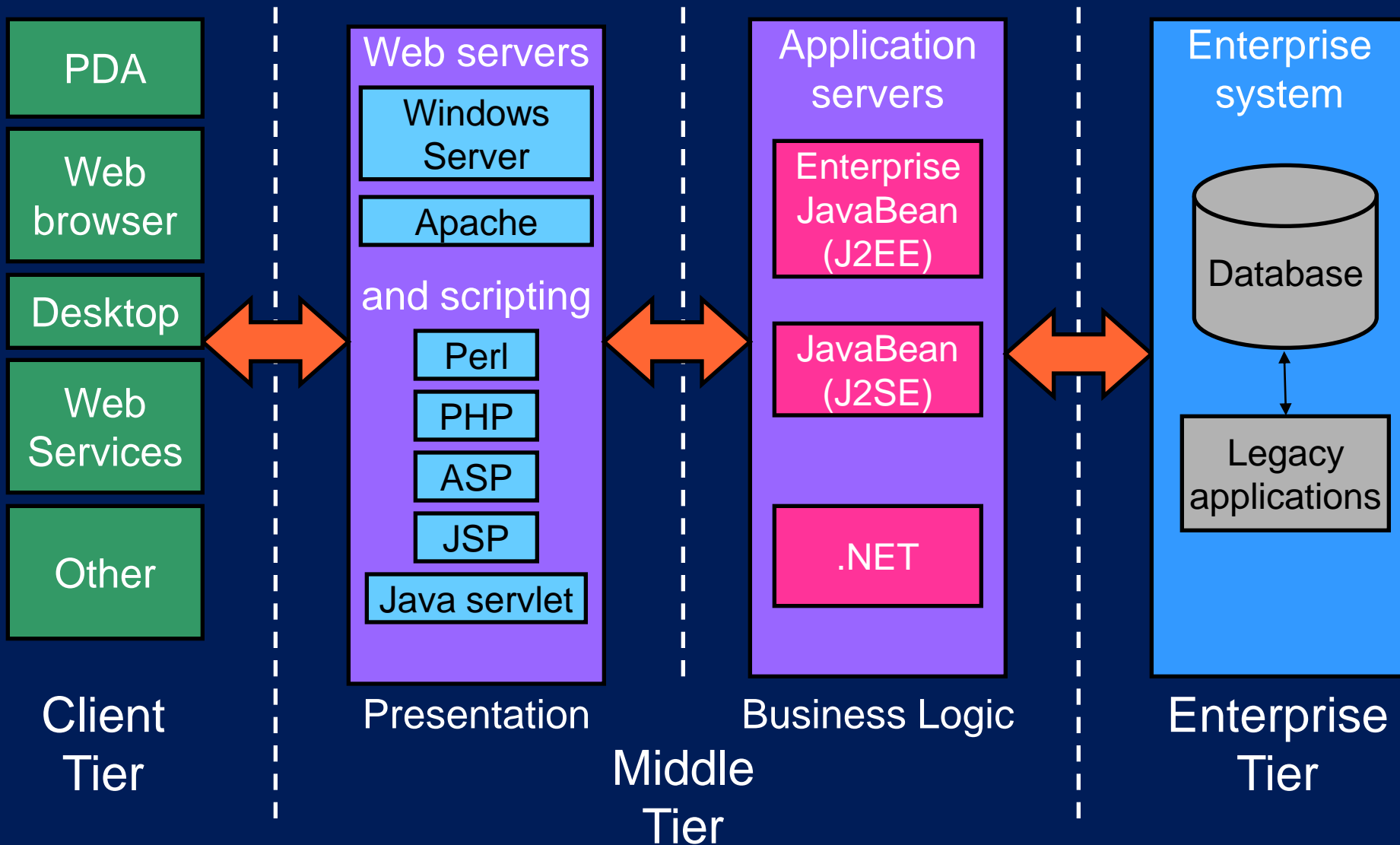


- Legacy Applications continue to be the backbone of today's enterprises.
 - Most business logic still embedded in these applications.
 - Proven reliability
 - Developed familiarity
- New Technologies are continually introduced.
 - As time marches on, new standards & technologies emerge.
- Pressure to integrate existing applications.
 - Leverage existing code & stability in new ways.

Web Services and Integration

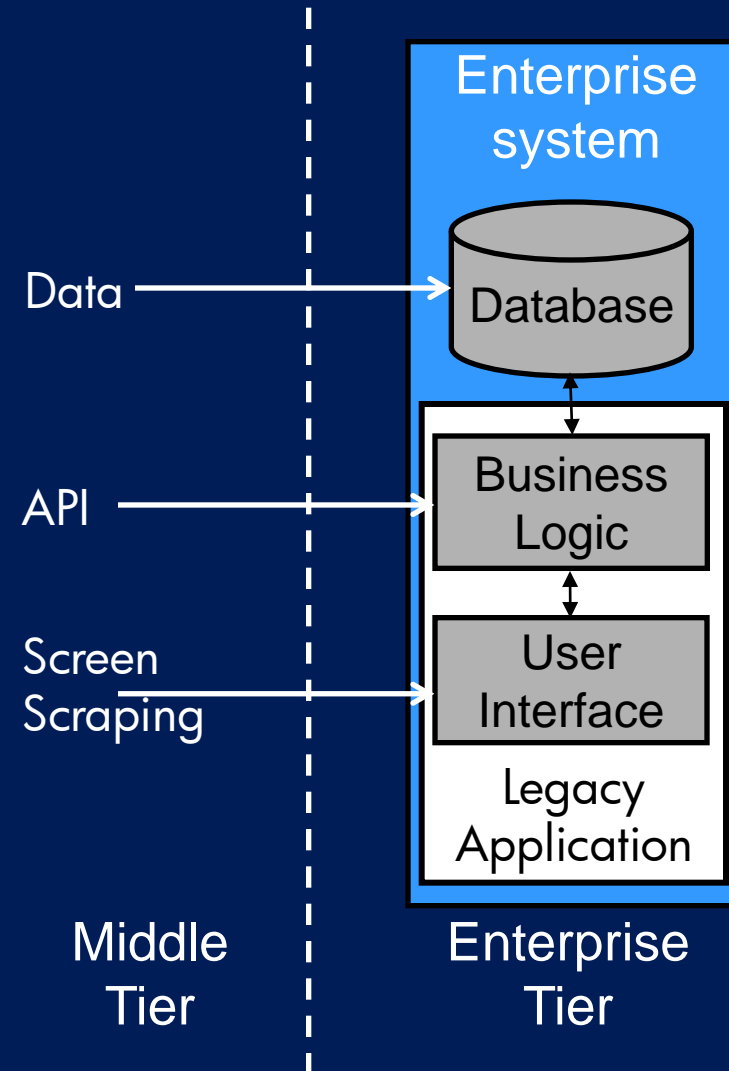
- Web Services is fundamental, enabling technology for integration solutions
 - Vendor, platform, and language independent (Industry Std)
 - The way to integrate with Microsoft .NET
 - An easy way to integrate with J2EE
- Think of Web Services as “middleware for seamless integration”
- Dynamic computing environment for applications

Today's Environment



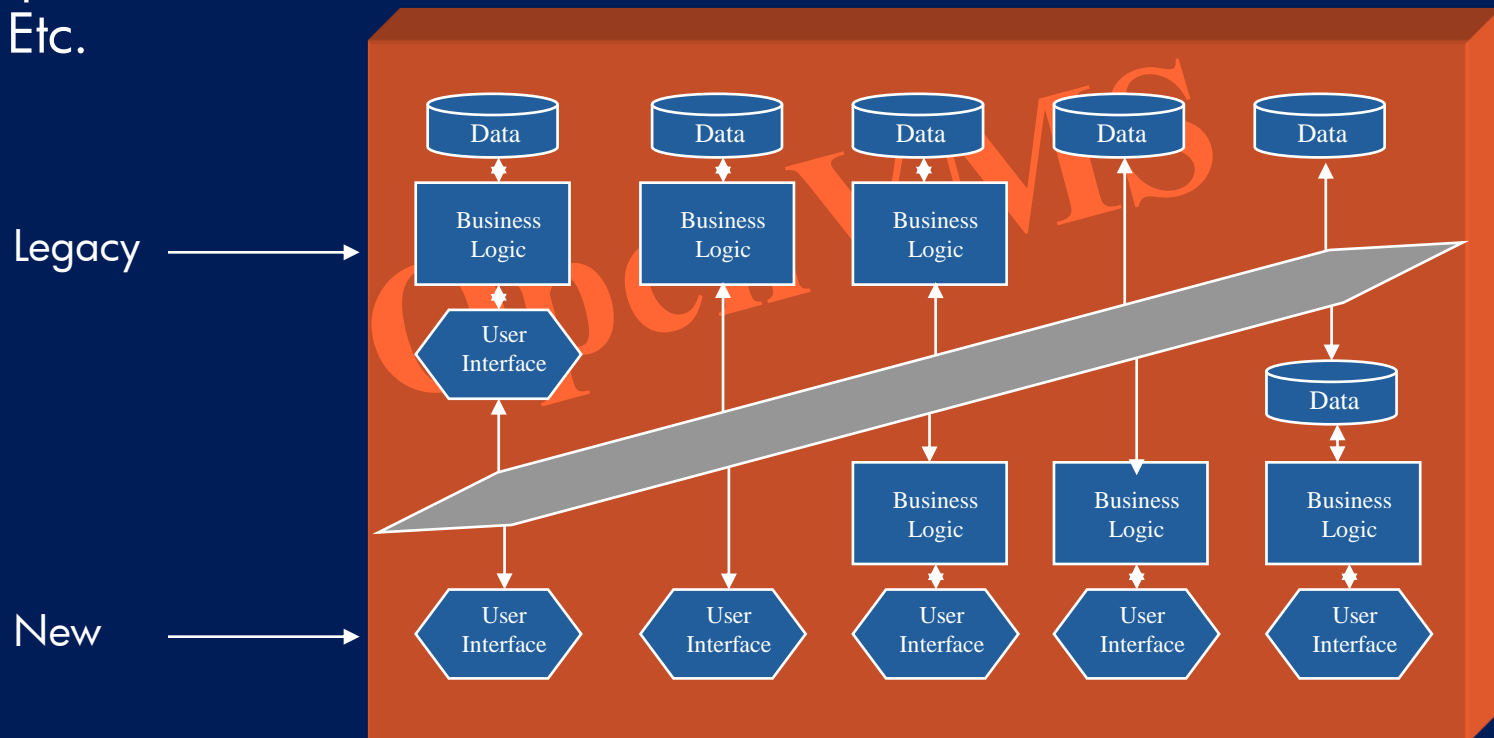
Many integration possibilities

- Drivers to access the enterprise data directly.
- API wrappers to access exposed business logic.
- Screen Scrapers to convert green screens into method calls.



Which one is right for you?

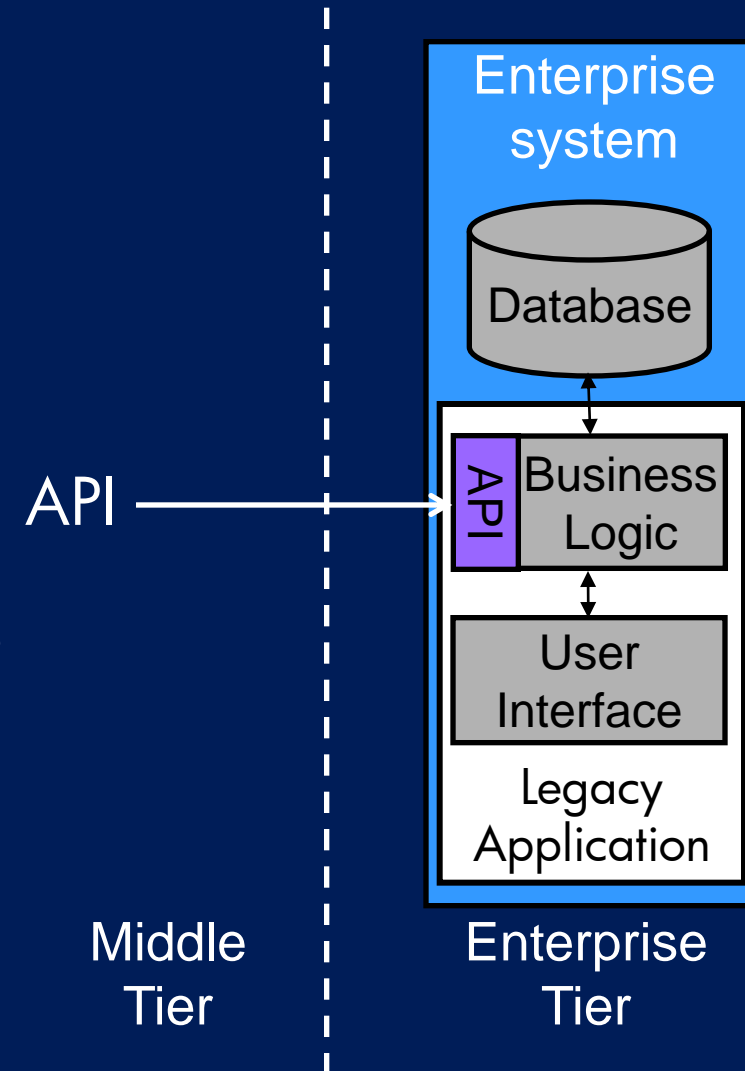
- How open is the application to being encapsulated?
- How much do you want to rewrite?
- How important is performance to you?
- Does the application have natural boundaries that can be exploited?
- Etc.



API Integration



- Expose APIs in the existing application as objects
 - Many integration possibilities, APIs to access data, objects, business logic, specific functions, etc.
 - Requires an API, either existing or newly created, in the application
- Use when:
 - There are significant time, cost, performance, and risk advantages to reuse existing application
- Benefits
 - Uses “natural boundaries” in existing applications
- Enabling technology
 - Specialized toolkits to wrap APIs



WSIT Overview

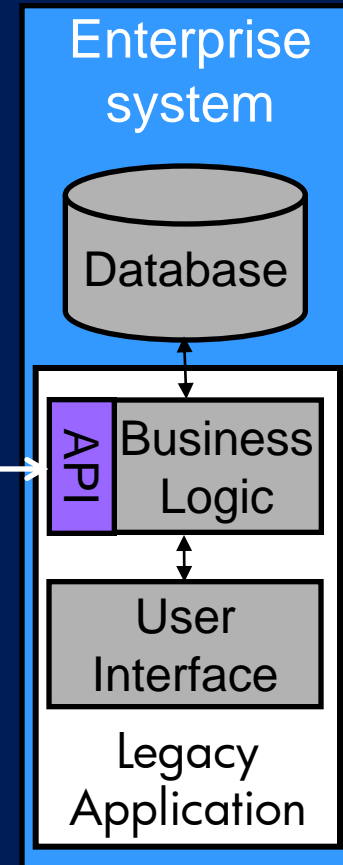


- The going forward API level integration capability for OpenVMS Alpha and I64
- A set of tools that work together to expose application business logic as java objects.
- Target application
 - Integration goals and constraints understood
 - APIs (to be exposed) exist, which might have required modifications to the application
- WSIT user
 - Architect, developer, or consultant, who understands target application

API



Middle Tier



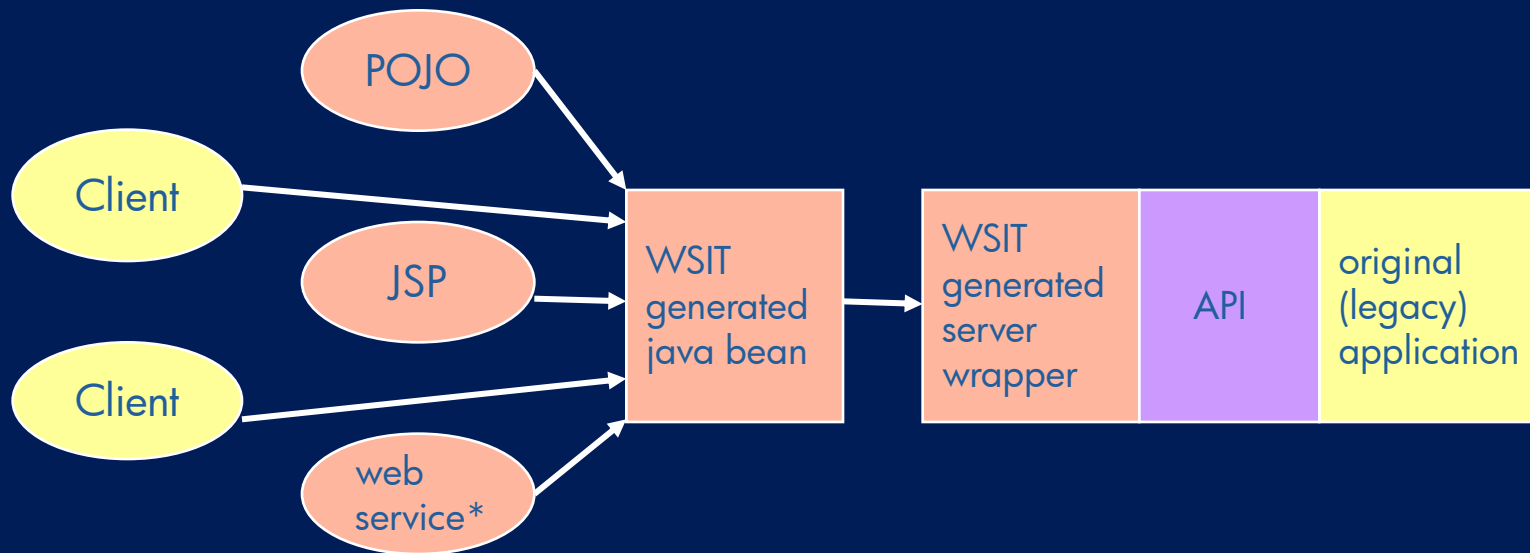
Enterprise Tier

- Tools that can be used individually or together, including:
 - OBJ2IDL
 - Given an I64 object file (with debug records), generates WSIT IDL
 - ACMS2IDL
 - Given an ACMS STDL file, generates WSIT IDL
 - IDL2CODE
 - Given WSIT IDL, generates all wrapper & interface code
 - Uses user modifiable Apache Velocity based templates for generation
 - XMLValidate
 - Validates WSIT IDL against specified XML schema
 - ANT command environment to integrate the development process
 - A scalable runtime environment.

WSIT - Development Steps



- Develop clients for the new application
 - Use the clients that WSIT optionally creates as a starter
 - Write your own using WS, java, J2EE, JSPs, Servlets, etc.



*Sample Web Server client is new for WSIT V2.0

WSIT Features - OpenVMS



- Understands OpenVMS based languages & environments.
 - Knowledge of 3GLs, such as C, BASIC, Fortran, COBOL, etc.
 - ACMS support
 - Support for most OpenVMS datatypes, structures, and arrays
 - Support for OpenVMS standard passing mechanisms
- Understands concepts, such as processes, threading, and inter-process communication within OpenVMS.
 - Built-in support for threaded & non-threaded applications.
 - Built-in support for process/server pooling for quicker access.
 - Built-in support for ACME services-based authentication.

Web Services and Integration - Summary



- Legacy Applications contain gold that needs to be mined.
 - You need to determine the best way to get at this gold
- API level Integration into Legacy Applications provide:
 - Code reuse (reliability, quicker development)
 - Better performance in most cases
- WSIT is an OpenVMS based API Integration Technology
 - Provides a set of tools that work together
 - Developed specifically for OpenVMS
 - Makes development quick & easy
 - Generates a WS friendly java class as the new interface into the application

Web Services Integration Toolkit (cont)



- **Supports Alpha and Integrity**
- **OpenVMS Technical Journal article by David Sullivan**
<http://h71000.www7.hp.com/openvms/journal/v7/>
- **Current released version is 1.2, available for download on website**
- **Current FT version is T2.0 (release in October)**
 - **Support for binary large objects (BLOBs)**
 - **Built-in tracing for all wrapped routines and parameters.**
 - **Tighter checking of Boolean values in IDL.**
 - **Support for POJO clients with zero parameters**
 - **Bug fixes**
- **Coming in V3.0**
 - **Generate a sample AXIS2 web service caller for the encapsulated app**
 - **A monitor tool to manage WSIT application**
 - **More! Plus, we're taking requests!**

Questions?



Contacts:

Jim Lanciani
e-Business Engineering Manager
OpenVMS Systems Group
Jim.Lanciani@hp.com
603-884-2719

Meg Watson
Principal Software Engineer
OpenVMS Systems Group
Meg.watson@hp.com
603-885-2066

For more info:

<http://h71000.www7.hp.com/ebusiness/technology.html>

Slides not used

