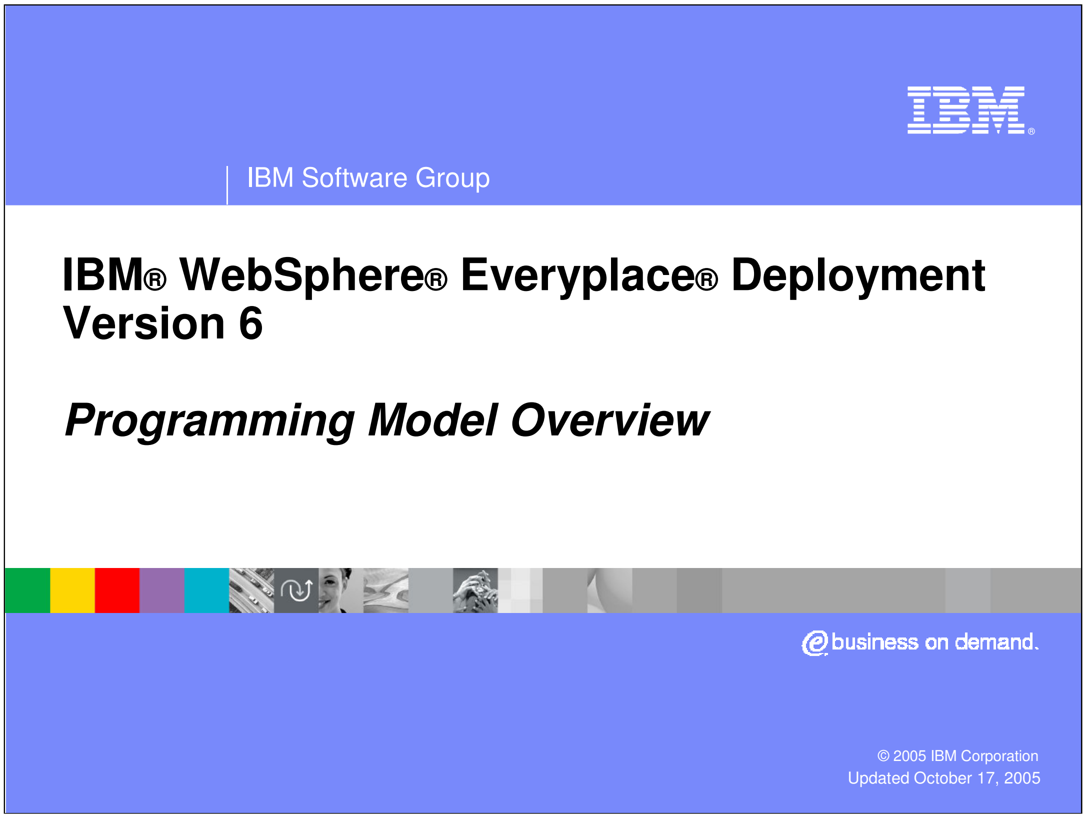


IBM Software Group

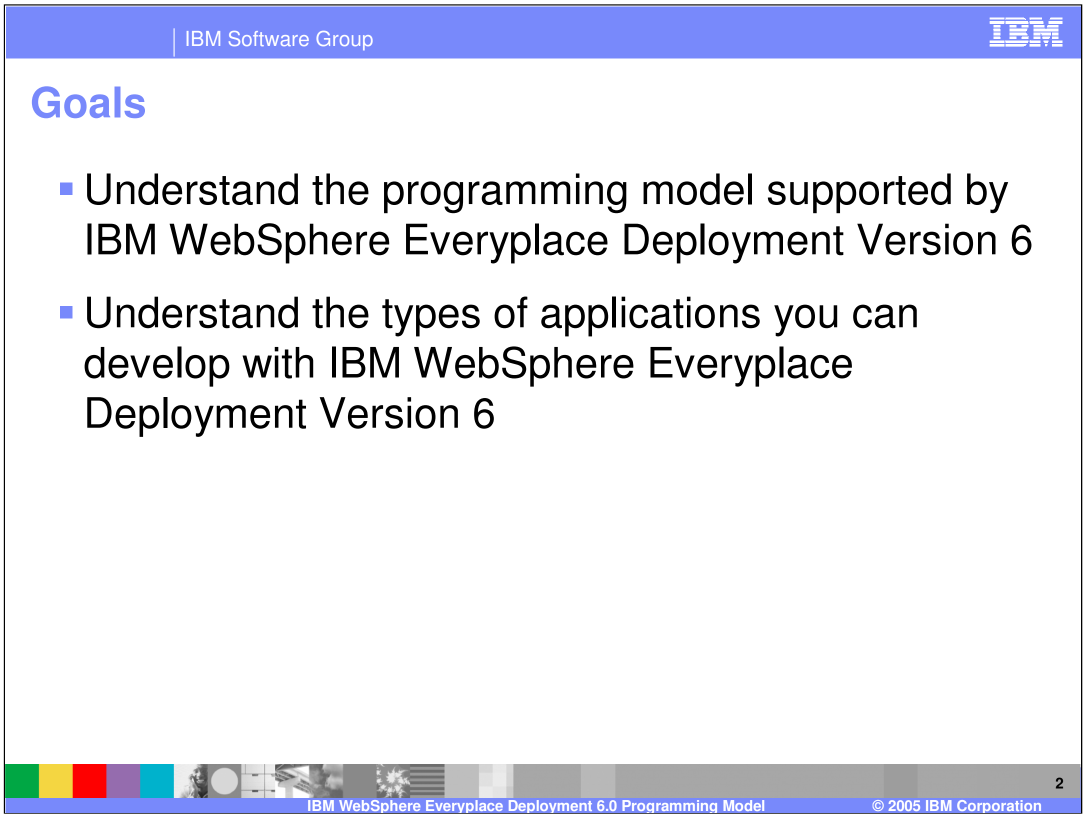
IBM® WebSphere® Everyplace® Deployment Version 6

Programming Model Overview

business on demand.

Updated October 17, 2005

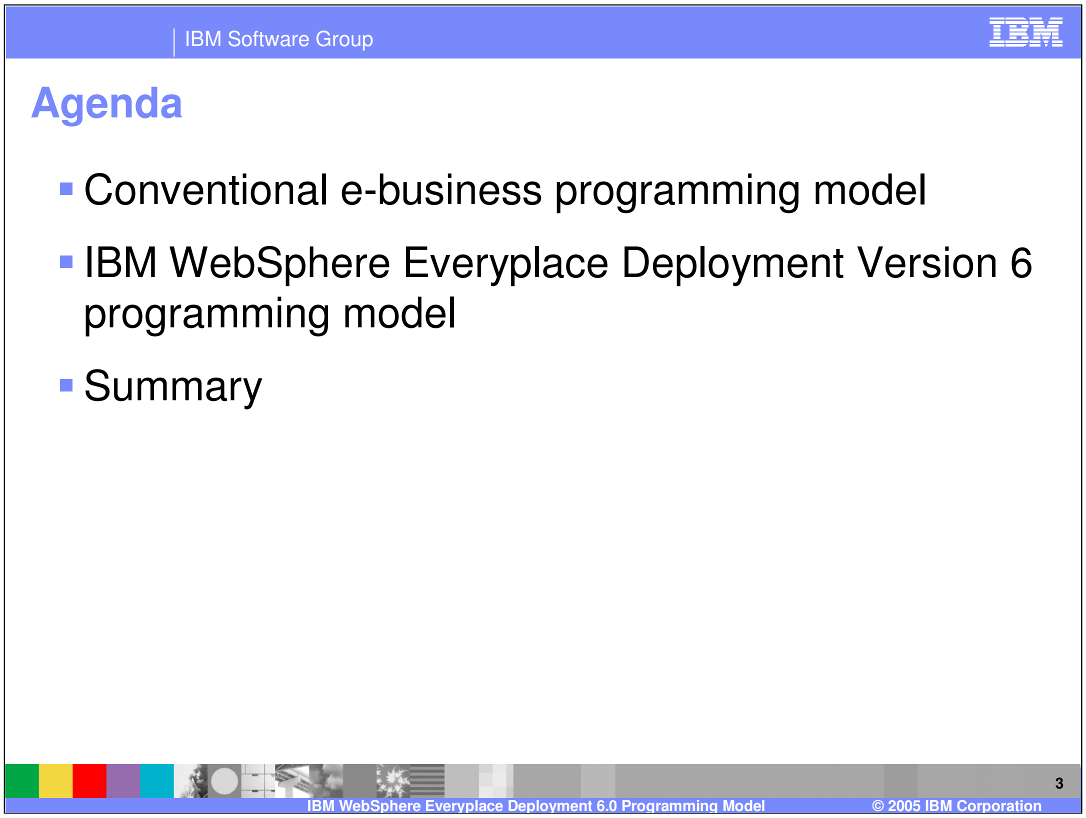
This presentation will provide an overview of the programming model supported by IBM WebSphere Everyplace Deployment Version 6.



Goals

- Understand the programming model supported by IBM WebSphere Everyplace Deployment Version 6
- Understand the types of applications you can develop with IBM WebSphere Everyplace Deployment Version 6

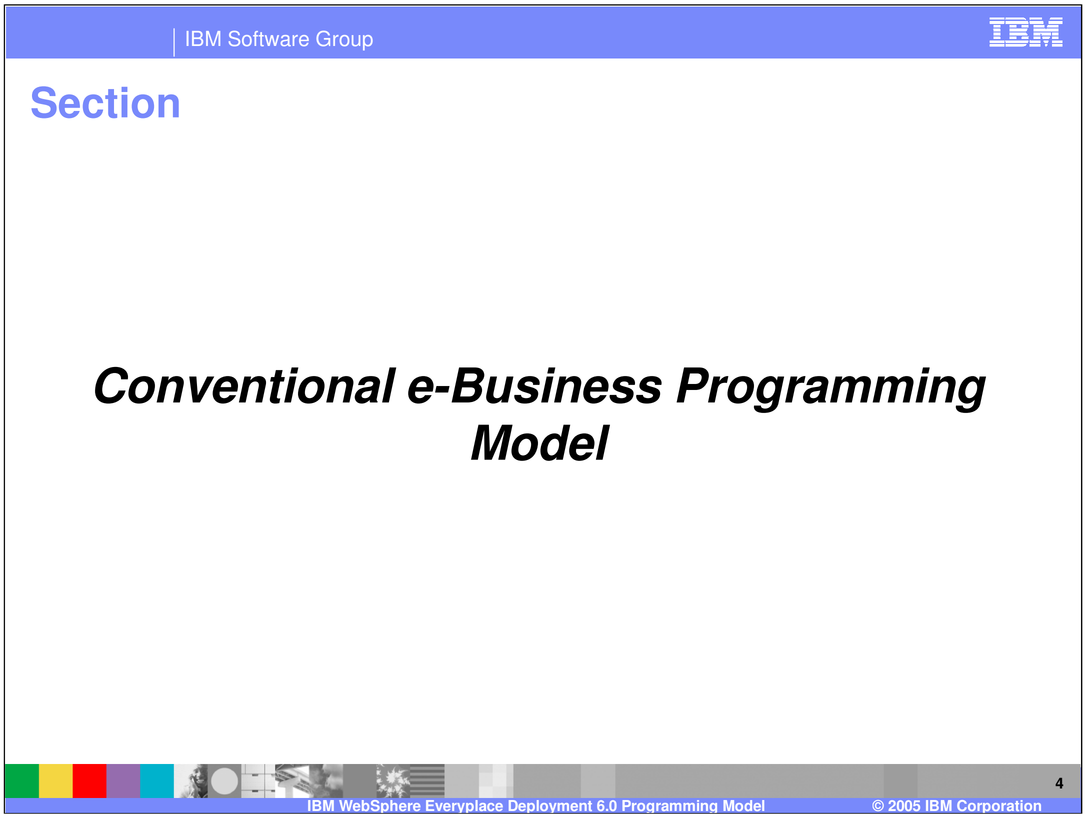
The goals of this presentation are to understand the programming model supported by IBM WebSphere Everyplace Deployment Version 6 and to understand the types of applications you can develop with IBM WebSphere Everyplace Deployment Version 6 as the result of this programming model.



Agenda

- Conventional e-business programming model
- IBM WebSphere Everyplace Deployment Version 6 programming model
- Summary

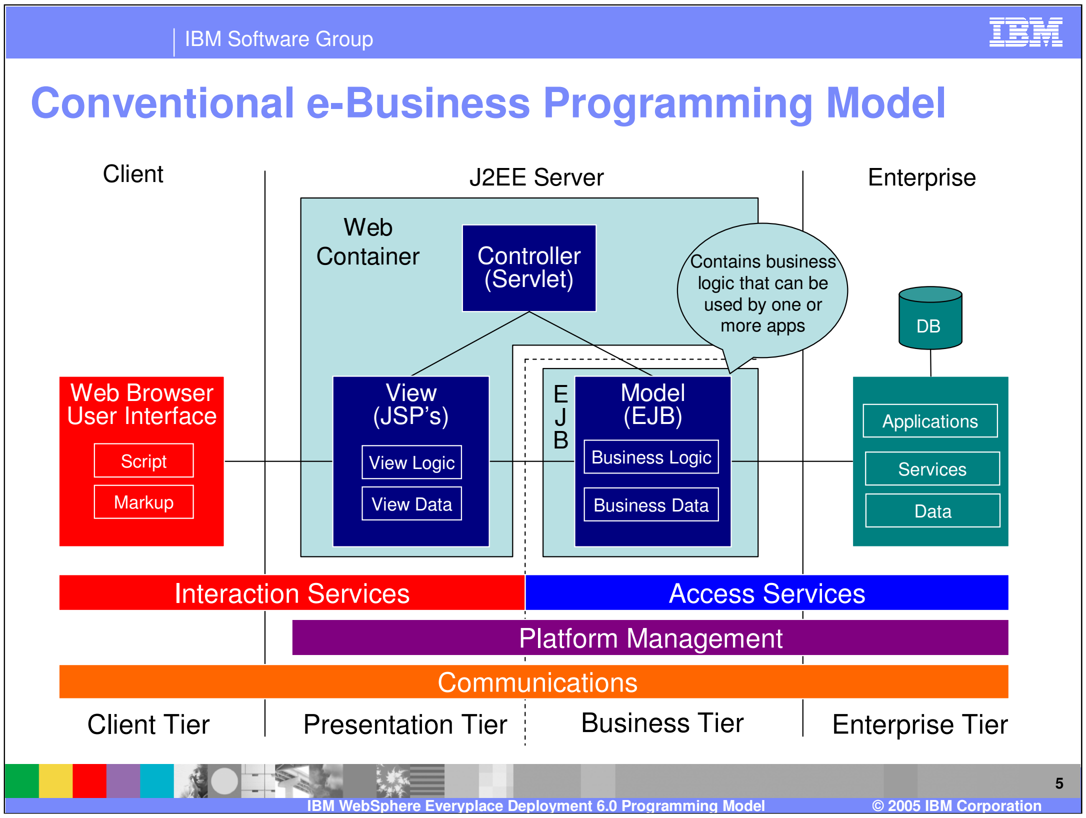
The agenda of this presentation is to review the conventional e-business programming model, to explain how IBM WebSphere Everyplace Deployment Version 6 extends this programming model to the client platform and the types of applications you can develop as a result, and to summarize the benefits of this programming model.



Section

Conventional e-Business Programming Model

Let’s start with a review the conventional programming model used to develop e-business applications.



IBM Software Group

Conventional e-Business Programming Model

5

IBM WebSphere Everyplace Deployment 6.0 Programming Model © 2005 IBM Corporation

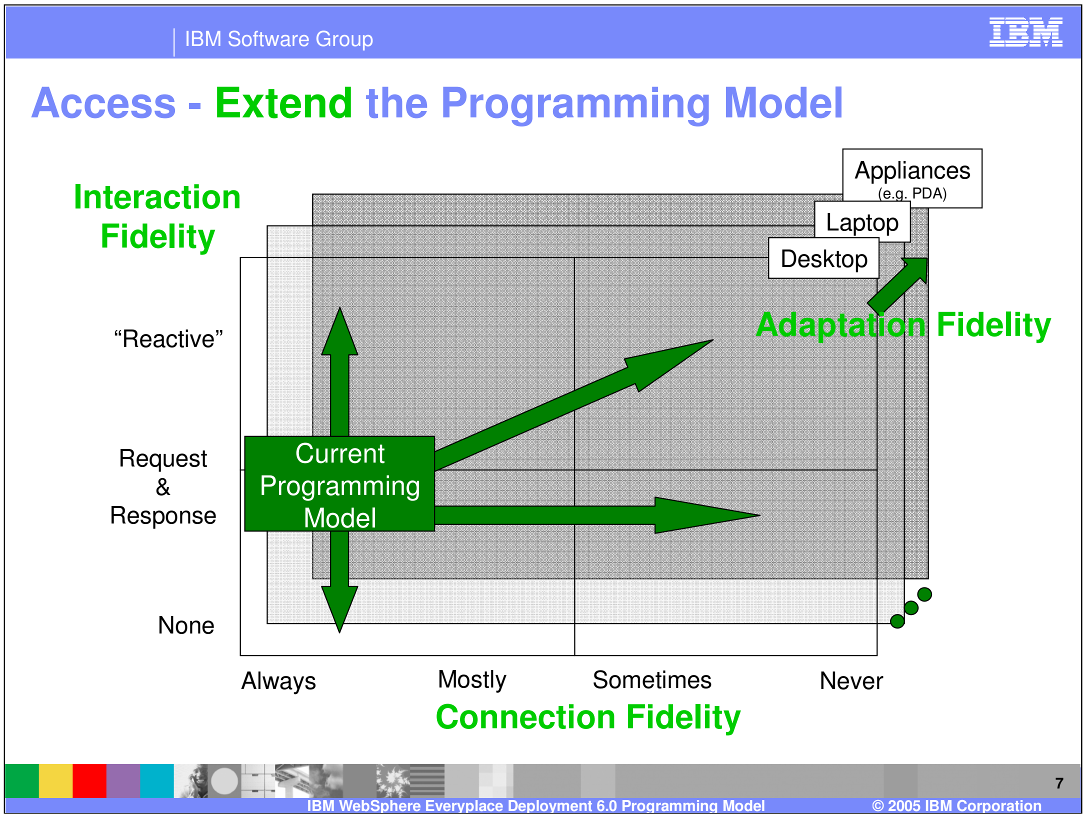
In the conventional e-business end-to-end programming model, also known as the J2EE Web programming model, an end-to-end application consists of components distributed across multiple tiers and nodes, where each node is a physical platform such as a device or server. These components can be logically classified by using the popular model-view-controller (MVC) pattern. In MVC, a *model* represents business data and the business logic that manipulates that data, a *view* interacts with the model on behalf of a user or other external interaction, and a *controller* coordinates the flow of interaction between the view and the model.

The components of the application run in *containers* located on *tiers* as defined by the J2EE architecture. Web applications run in a *Web Container* on the *Presentation Tier* and Enterprise JavaBeans (EJBs) run in an *EJB container* on the *Business Tier*. Containers can provide *Interaction Services* that enable J2EE components to interact with users and *Access Services* that enable J2EE components to access Enterprise applications, services, and data.

Section

IBM WebSphere Everyplace Deployment Version 6 Programming Model

Next, let’s discuss how IBM WebSphere Everyplace Deployment Version 6 extends the conventional e-business programming model to the client.



IBM Software Group

Access - Extend the Programming Model

Interaction Fidelity

"Reactive"

Request & Response

None

Current Programming Model

Appliances (e.g. PDA)

Laptop

Desktop

Adaptation Fidelity

Always

Mostly

Sometimes

Never

Connection Fidelity

7

IBM WebSphere Everyplace Deployment 6.0 Programming Model

© 2005 IBM Corporation

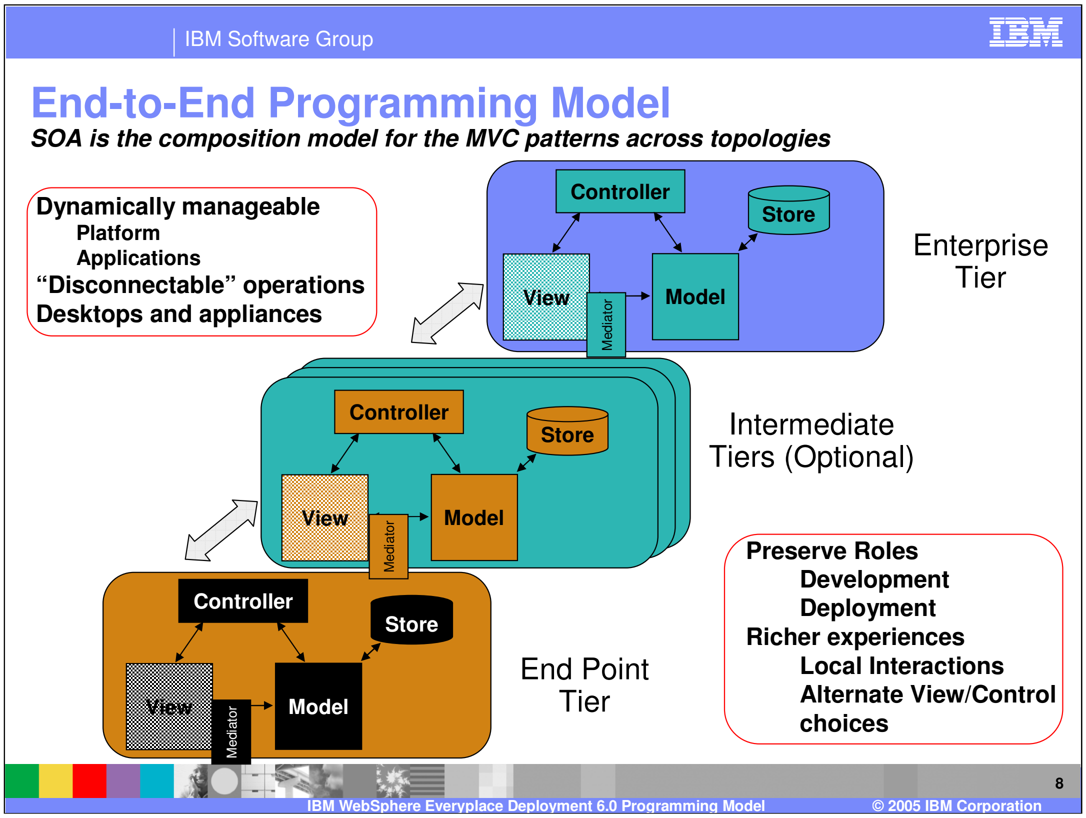
The conventional J2EE programming model works fine in an environment where users interact with a Web browser on a highly functional desktop (or equivalent) device to send requests to and receive responses from Web applications over a relatively reliable, high bandwidth network. However, what about users who work in a much more diverse environment in which these fundamental assertions do not hold true?

IBM's Server Managed Client technology provides a solution by extending the programming model to client devices so that J2EE developers can extend their existing applications to, and create new applications for, a wide range of devices, connection states, and user experiences while allowing them to use their existing skills.
Extending the programming model leads to unique requirements in the following three areas:

•**Connection Fidelity** – The programming model must enable applications to operate across states that are always, mostly, occasionally, and never connected. Network characteristics determine the active connection state. These characteristics range from fully connected, high bandwidth connections such as hardwired Ethernet to very low bandwidth connections such as today's 2G cellular networks. Network characteristics also include latency, geographic coverage, and even billing models.

•**Interaction Fidelity** – The programming model must continue to support the dominant Web browser-based request and response user experience. However, the programming model must also support other user experiences that are increasingly becoming important. Rich user experiences include rich graphical user interfaces (GUIs) that directly utilize the native widgets of an operating system and multimodal browsers that combine visual and voice interactions in a single user experience. For embedded devices, no user experience is available; however, the programming model must enable applications to run on those devices.

•**Adaptation Fidelity** – The programming model must address a wide range of client devices covering desktops, laptops, tablets, and appliances, which include embedded controllers, smart phones and PDAs.



IBM Software Group

End-to-End Programming Model

SOA is the composition model for the MVC patterns across topologies

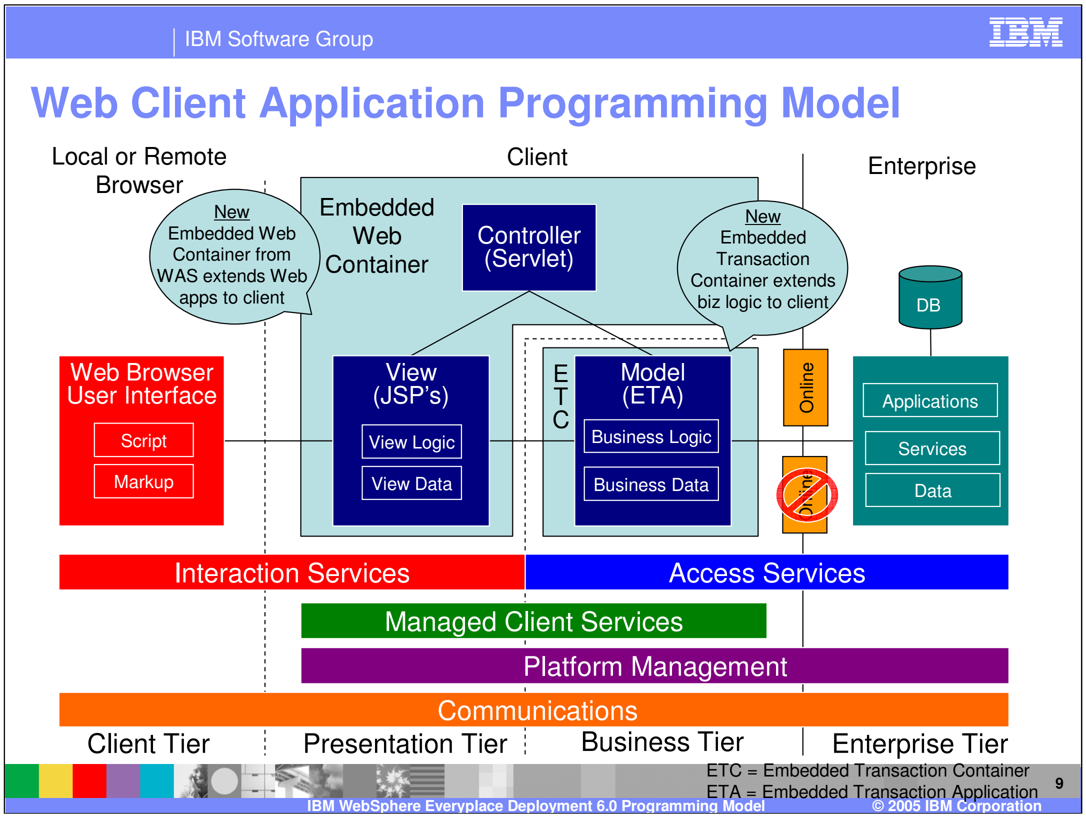
Dynamically manageable
Platform
Applications
“Disconnectable” operations
Desktops and appliances

Preserve Roles
Development
Deployment
Richer experiences
Local Interactions
Alternate View/Control choices

8

IBM WebSphere Everyplace Deployment 6.0 Programming Model © 2005 IBM Corporation

IBM’s Server Managed Client technology defines a programming model that addresses these requirements through a set of containers and services that enable applications to operate in this environment. This includes the capability to distribute key components of applications to the client, and possibly intermediate tiers, so users can continue to perform selected business operations with these applications even when their device is offline. Moving application components to the client can also result in improved performance through local interactions between the end-user and their application, and alternate view/control choices, such as a graphical user interface. By preserving a selected set of J2EE APIs and roles, developers can reuse their skills and experience to develop client applications. Furthermore, IBM WebSphere Everyplace Deployment Version 6 enables you to compose end-to-end applications through its support for a Service-Oriented Architecture (SOA). For more information on how IBM WebSphere Everyplace Deployment Version 6 supports end-to-end applications, please read “End-to-End Applications”.



IBM Software Group

IBM

Web Client Application Programming Model

Local or Remote Browser

Client

Enterprise

New Embedded Web Container from WAS extends Web apps to client

Embedded Web Container

Controller (Servlet)

New Embedded Transaction Container extends biz logic to client

DB

Web Browser User Interface

Script

Markup

View (JSP's)

View Logic

View Data

E T C

Model (ETA)

Business Logic

Business Data

Online

Online

Applications

Services

Data

Interaction Services

Access Services

Managed Client Services

Platform Management

Communications

Client Tier

Presentation Tier

Business Tier

Enterprise Tier

ETC = Embedded Transaction Container

ETA = Embedded Transaction Application

9

IBM WebSphere Everyplace Deployment 6.0 Programming Model

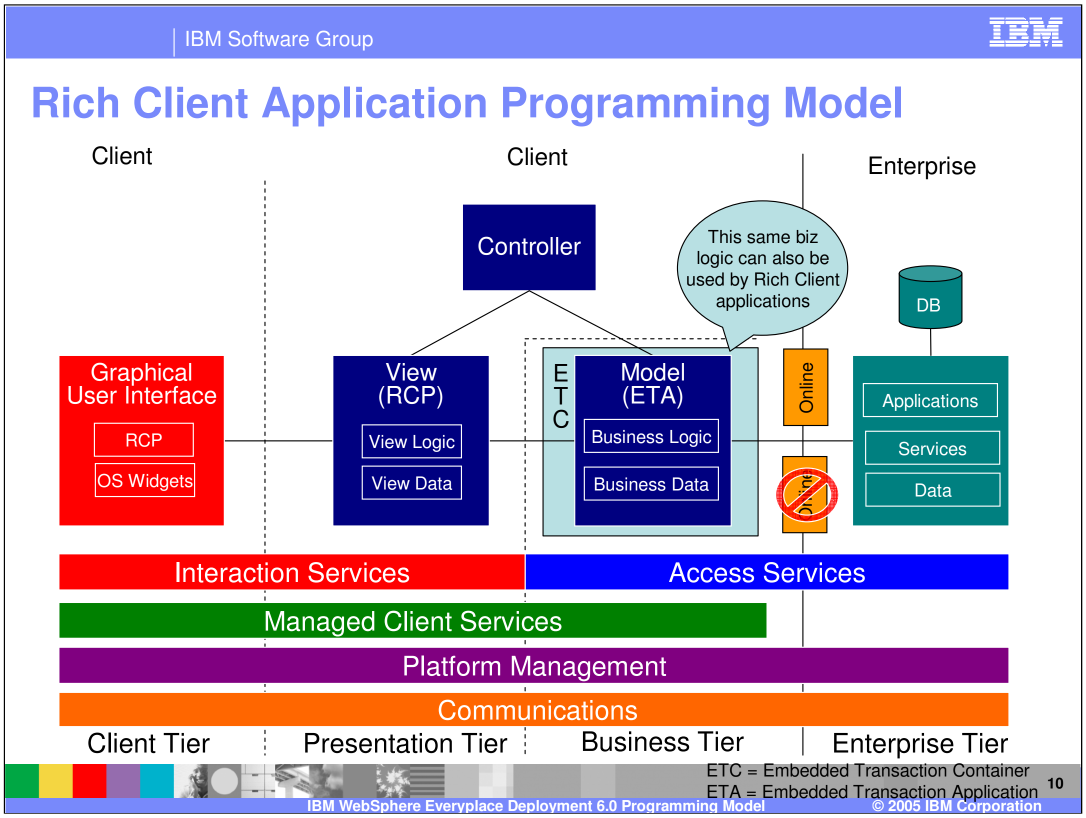
© 2005 IBM Corporation

IBM WebSphere Everyplace Deployment for Windows and Linux Version 6 implements this client programming model by providing a new *Embedded Web Container* from WebSphere Application Server (WAS) and a new *Embedded Transaction Container (ETC)* with services that enable Web applications and Embedded Transaction Applications respectively to run on clients. These containers and services support selected APIs from the J2EE programming model to facilitate reuse of components and skills, while also supporting online and offline operations. The Embedded Web Container extends the J2EE and WebSphere programming model to the client by supporting standard Web applications that comply with the Servlet 2.3 and JSP 1.2 specifications as well as the Servlet 2.4 and JSP 2.0 specifications. The new Embedded Transaction Container further extends the symmetry of the client programming model with the J2EE and WebSphere programming model by supporting a subset of the EJB 2.0 specification. As a testament to the power of this programming model, Blue Martini Software had their Web application running on the client platform and supporting mobile users with laptops in a matter of weeks.

IBM Software Group

IBM

Rich Client Application Programming Model



Client

Client

Enterprise

Controller

This same biz logic can also be used by Rich Client applications

DB

Graphical User Interface

RCP

OS Widgets

View (RCP)

View Logic

View Data

ETC

Model (ETA)

Business Logic

Business Data

Online

Online

Applications

Services

Data

Interaction Services

Access Services

Managed Client Services

Platform Management

Communications

Client Tier

Presentation Tier

Business Tier

Enterprise Tier

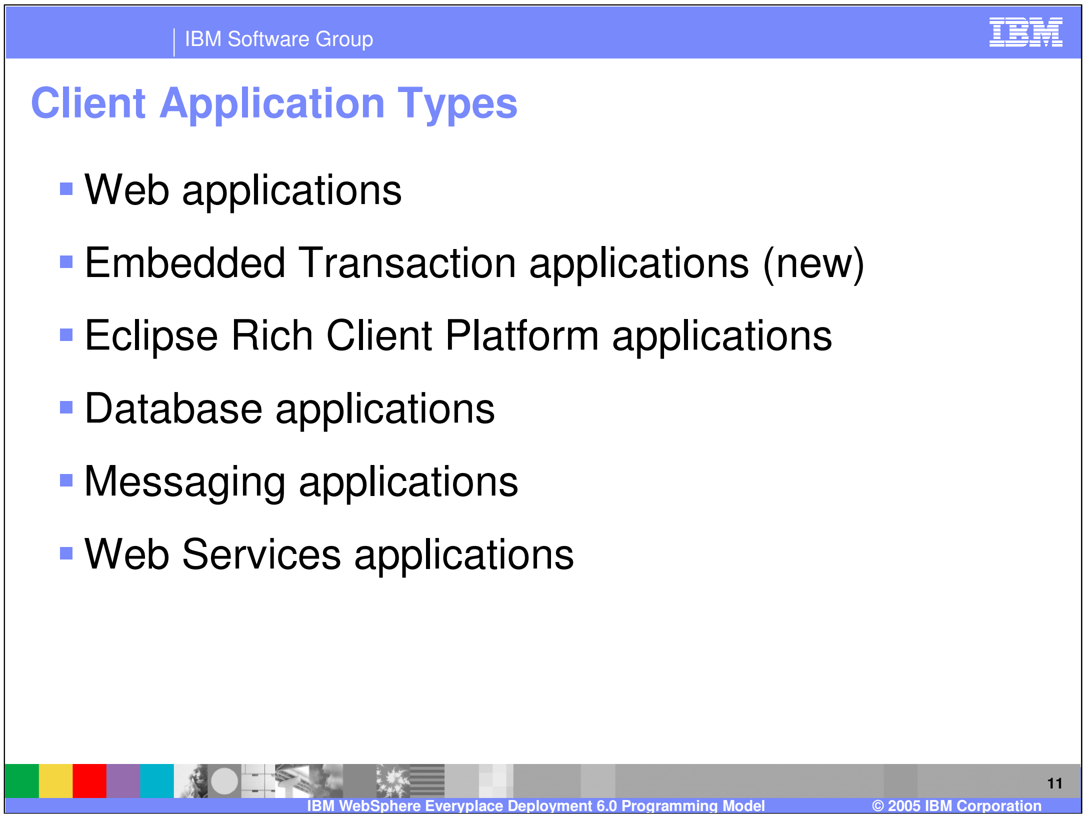
ETC = Embedded Transaction Container

ETA = Embedded Transaction Application

10

IBM WebSphere Everyplace Deployment 6.0 Programming Model

The client platform also enables you to deliver client applications that provide a rich graphical user interface through support of the Eclipse Rich Client Platform. You can continue to use business logic components, such as Embedded Transaction Applications, with your rich client applications.



Client Application Types

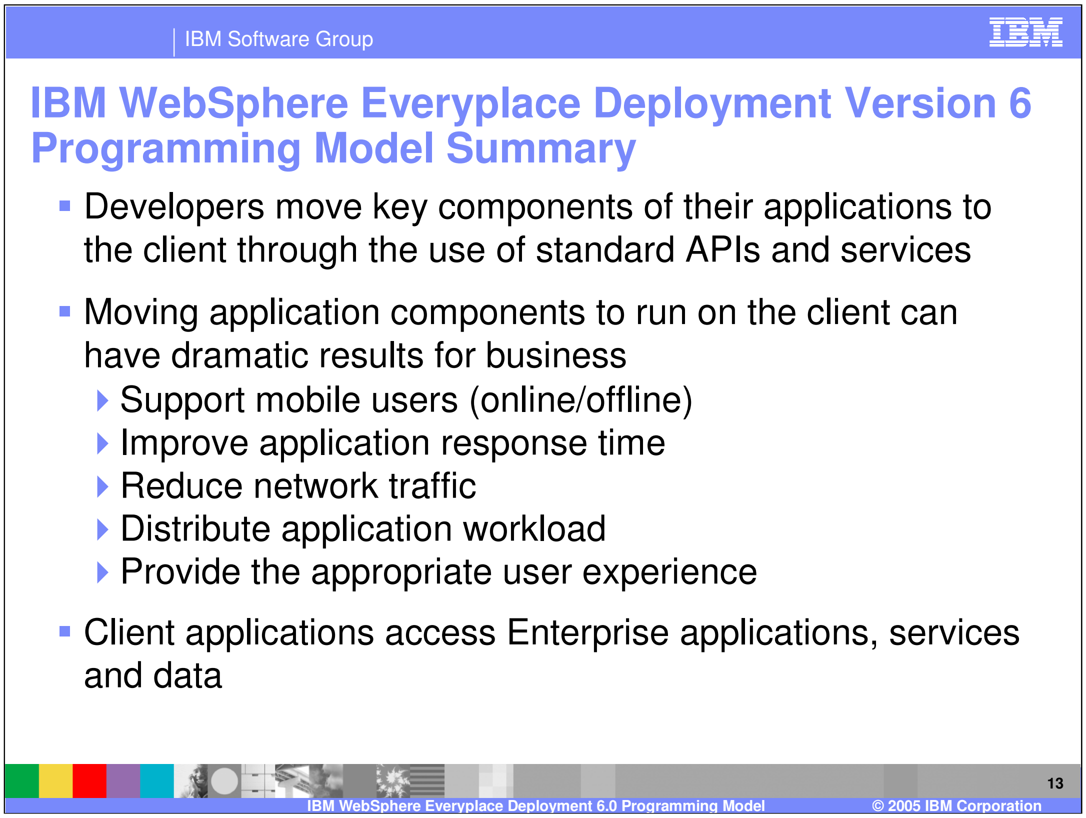
- Web applications
- Embedded Transaction applications (new)
- Eclipse Rich Client Platform applications
- Database applications
- Messaging applications
- Web Services applications

As a result of the Client Services and programming models supported by the client platform, you can develop a variety of standards-based client applications including Web applications that provide a traditional Web browser user experience, new Embedded Transaction applications for business logic, Eclipse Rich Client Platform applications with a rich graphical user interface, database applications to access and synchronize relational data, messaging applications to perform secure transactions, and Web Services applications to access consumer and business data.

Section

IBM WebSphere Everyplace Deployment Version 6 Programming Model Summary

Now we will summarize the programming model for IBM WebSphere Everyplace Deployment Version 6.

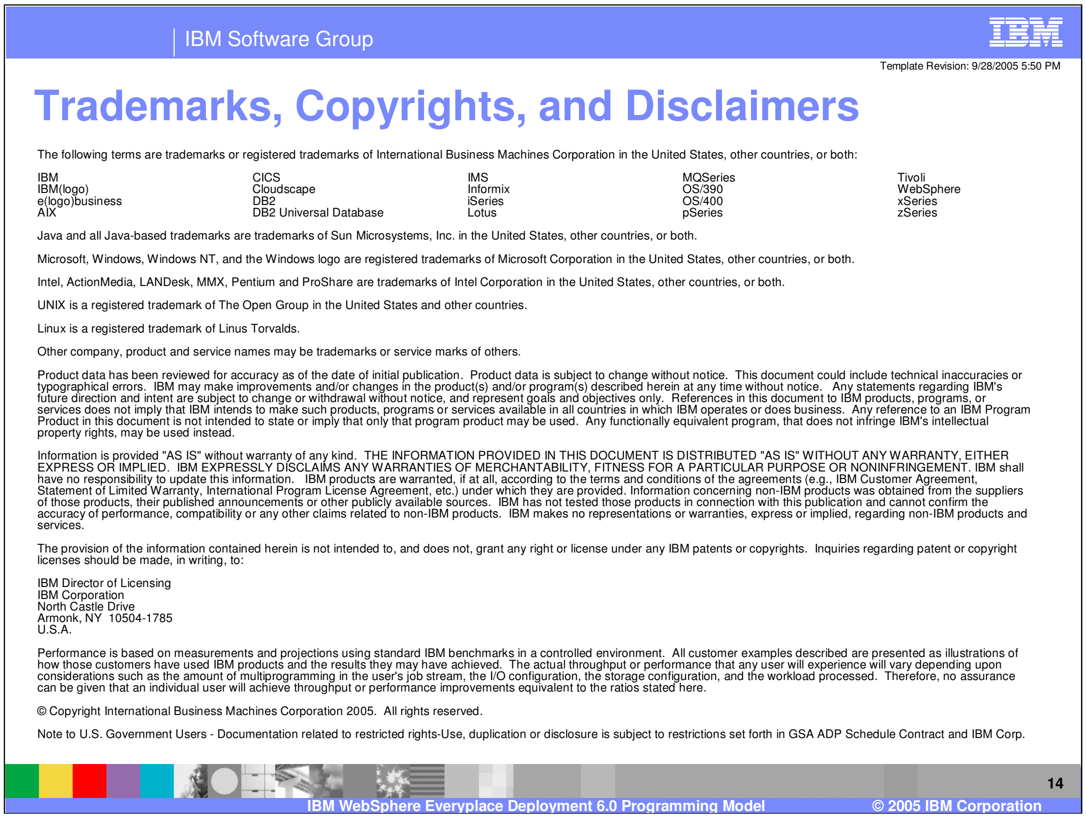


IBM WebSphere Everyplace Deployment Version 6 Programming Model Summary

- Developers move key components of their applications to the client through the use of standard APIs and services
- Moving application components to run on the client can have dramatic results for business
  - Support mobile users (online/offline)
  - Improve application response time
  - Reduce network traffic
  - Distribute application workload
  - Provide the appropriate user experience
- Client applications access Enterprise applications, services and data

Moving application components to run on a client can have dramatic results for business. End-users benefit from improved application response time because applications perform business operations locally on the client. As a result, there is a reduction in network traffic between clients and servers, and in server workload. Furthermore, mobile end-users can continue to productively use their applications from their clients even when they are at a location that does not have network connectivity, such as a customer site. You can also utilize the local graphical user interface (GUI) capabilities of the client devices to deliver a richer user experience than can be supported by a Web browser. Through support of Access Services, client applications can access mission-critical applications, services and data in the Enterprise.

In summary, the powerful IBM WebSphere Everyplace Deployment Version 6 client platform, toolkit, and server platform enable you to develop compelling applications that run on desktops and laptops, and securely access e-business applications, services, and data. You can use programming skills you have already acquired to develop these applications.



Template Revision: 9/28/2005 5:50 PM

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
IBM(logo)
e(logo)business
AIX
CICS
Cloudscape
DB2
DB2 Universal Database
IMS
Informix
iSeries
Lotus
MQSeries
OS/390
OS/400
pSeries
Tivoli
WebSphere
xSeries
zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.