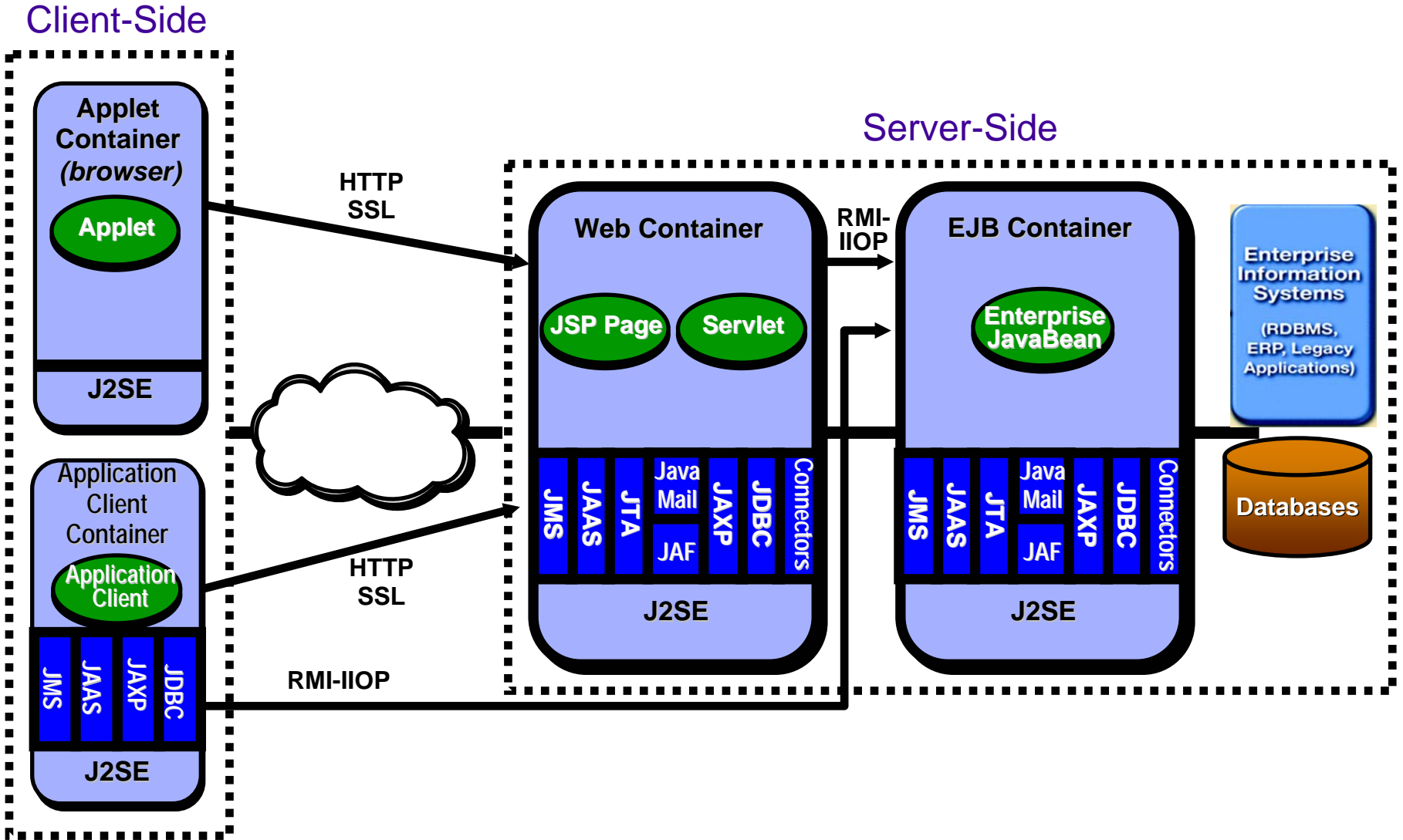
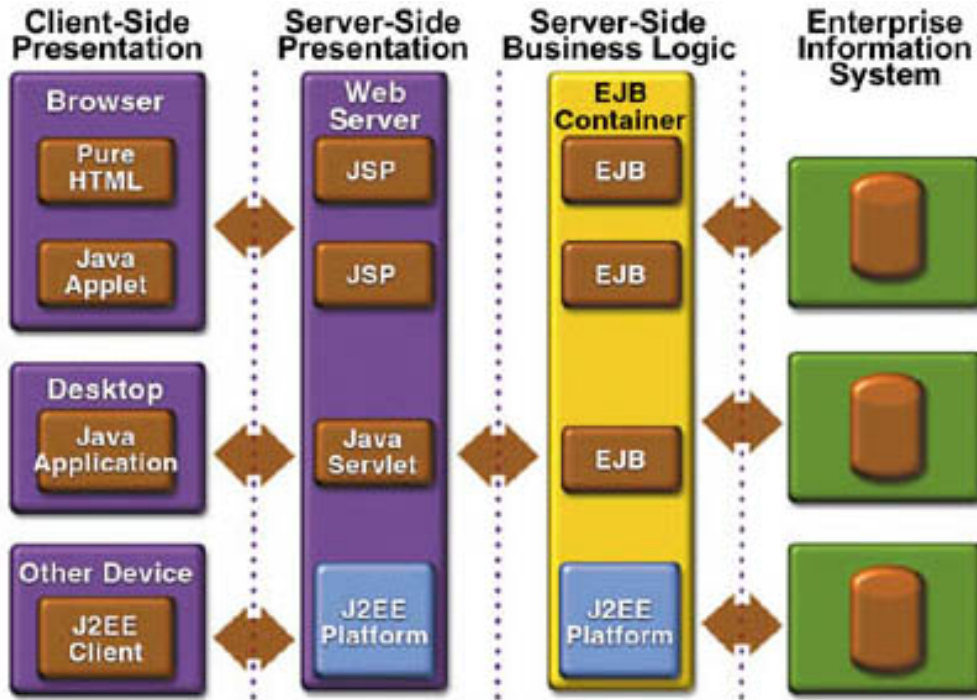


J2EE Platform



J2EE Platform Overview

From "J2EE Overview" on <http://java.sun.com/j2ee/overview3.html>

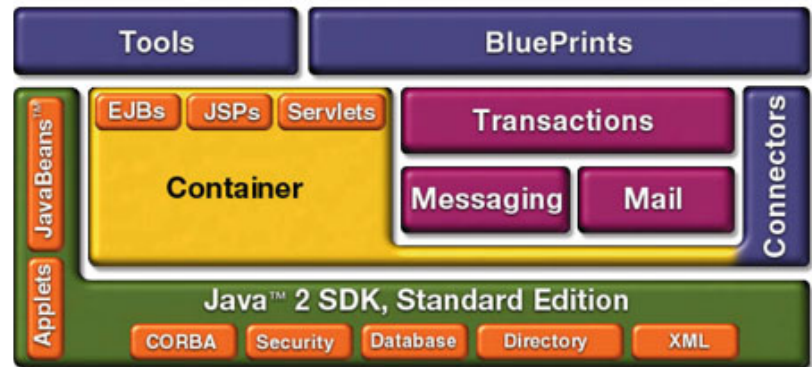


- Qualities of Service (QOS) Built into the Infrastructure
 - Transaction management
 - Life-cycle management
 - Resource pooling
- Application developers focus on business logic, not low-level api coding
- Cross-vendor portability and off-the-shelf components

Components, Containers and Connectors:

- Components are the key focus of application developers
- System vendors implement containers and connectors
 - Hide complexity
 - Enhance portability.

- Client Interface is Thin or Thick
 - Web Pages (HTML/XML/JSP) or Java Clients/Applets
 - Communicates HTML, XML, HTTP, SOAP, SSL, RMI, IIOP, etc.
- Business logic in Enterprise JavaBeans components
 - Distributed, Portable, Transactional, Persistent, Secure
- System vendors implement containers and connectors
- Bottom line benefits:
 - Increased programmer productivity, quality and maintenance
 - Better strategic use of cross-platform computing resources, including portability
 - Greater return on an organization's technology investments.



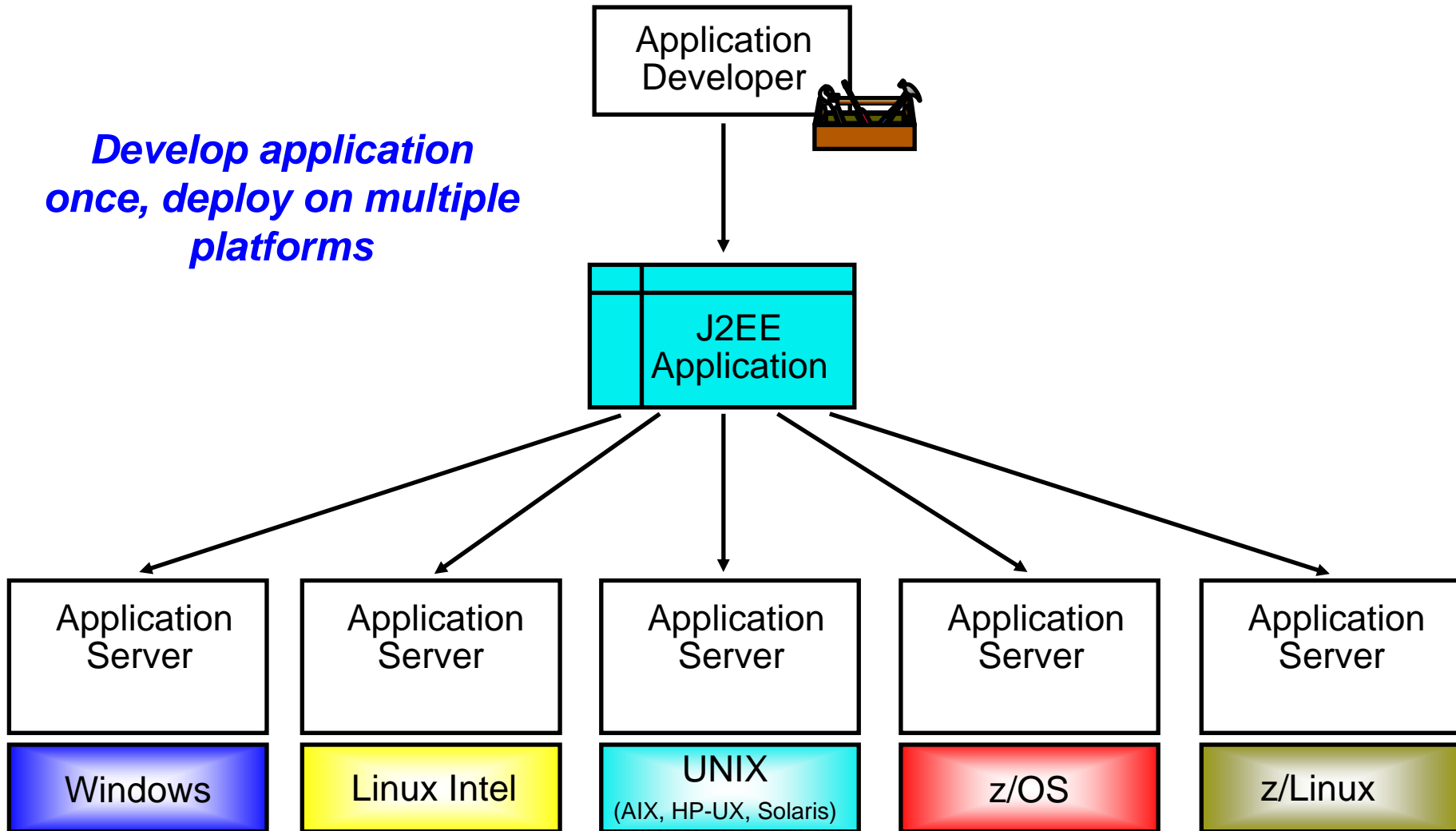
J2EE Standards

	J2EE Feature/API	Description	v1.2	v1.3	v1.4
<i>Application Components</i>	Applet	Java client delivered via browser	■	■	■
	Application Client	Java client run on native OS	■	■	■
	JSP	Server-side presentation logic	■	■	■
	Servlet	Control business logic execution	■	■	■
	EJB	Business logic, data encapsulation	■	■	■
<i>Directory</i>	JNDI	Naming and directory services	■	■	■
<i>Object Access</i>	RMI-IIOP	Remote object access	■	■	■
<i>Transactions</i>	JTA	Transaction services	■	■	■
<i>E-Mail Access</i>	JavaMail	Create/retrieve e-mail messages	■	■	■
	JAF	E-mail MIME type support	■	■	■
<i>Data and Application Access</i>	JDBC	Connect to relational databases	■	■	■
	JMS	Appl.-to-appl. messaging	■	■	■
	J2EE CA	Connect to external resources		■	■
	JAX-P	Java-XML parsing/processing		■	■
<i>Security</i>	JAAS	Authentication, authorization		■	■
	JACC	Connect to external authorization			■
<i>Web Services</i>	WebServices for J2EE	Web services deployment, execution			■
	JAX-RPC	Client access to Web services			■
	SAAJ	Manipulate SOAP attachments			■
	JAX-R	Access to XML registries (UDDI)			■
<i>Management</i>	Management API	Management tools/runtime interface			■
	JMX	Distributed management			■
	Deployment API	Deployment tools/runtime interface			■

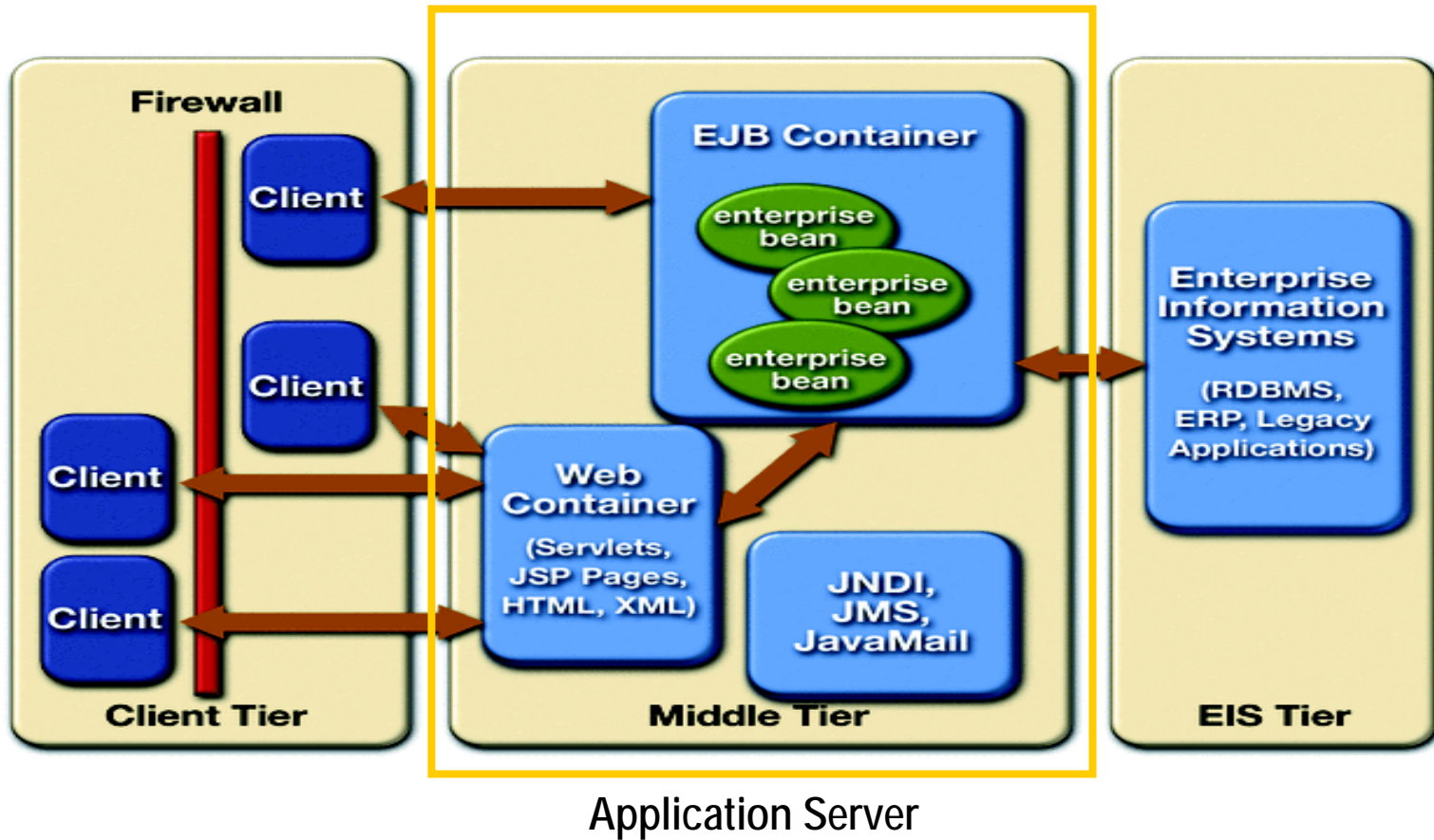
WebSphere Application Server and Rational Application Developer support J2EE V1.4, V1.3 and V1.2

J2EE Platform Portability and Scalability

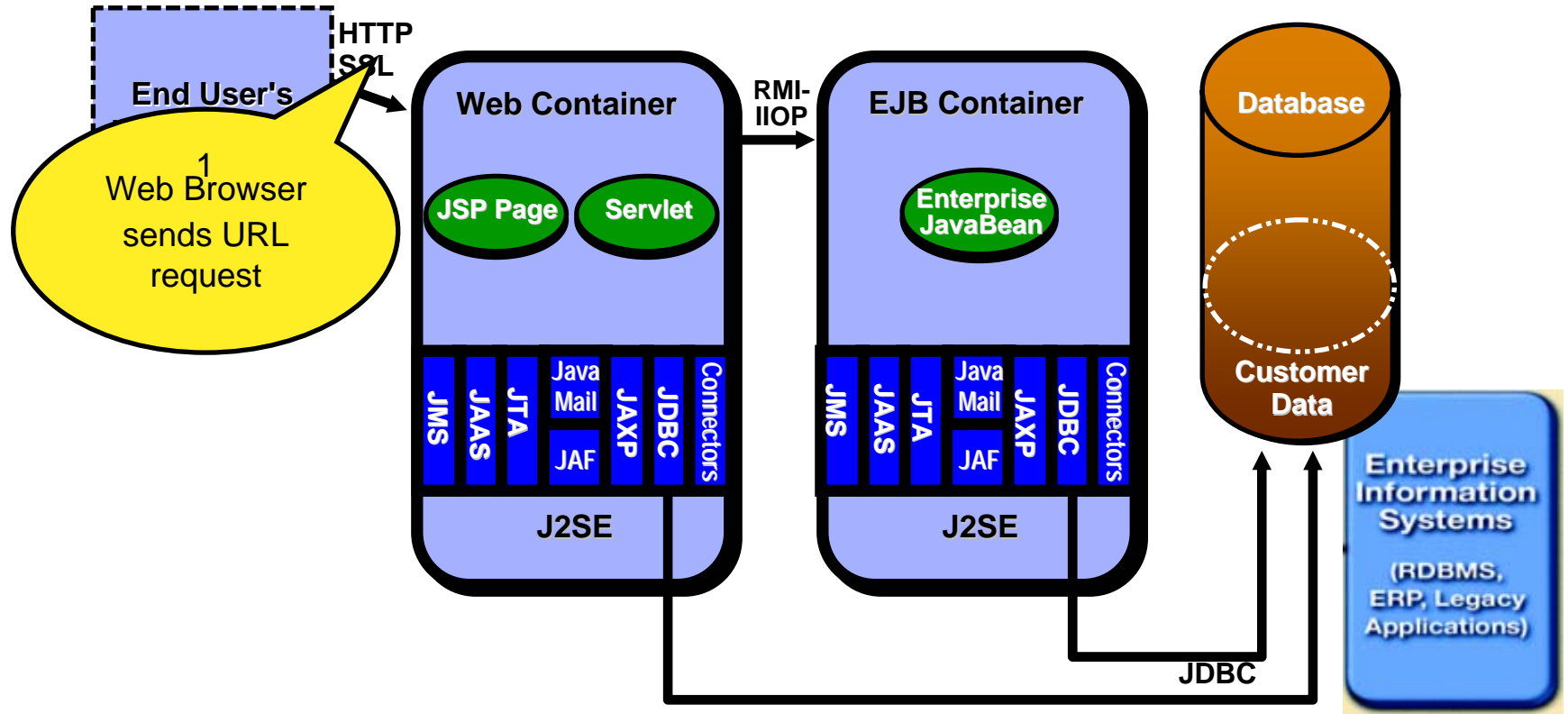
Develop application once, deploy on multiple platforms



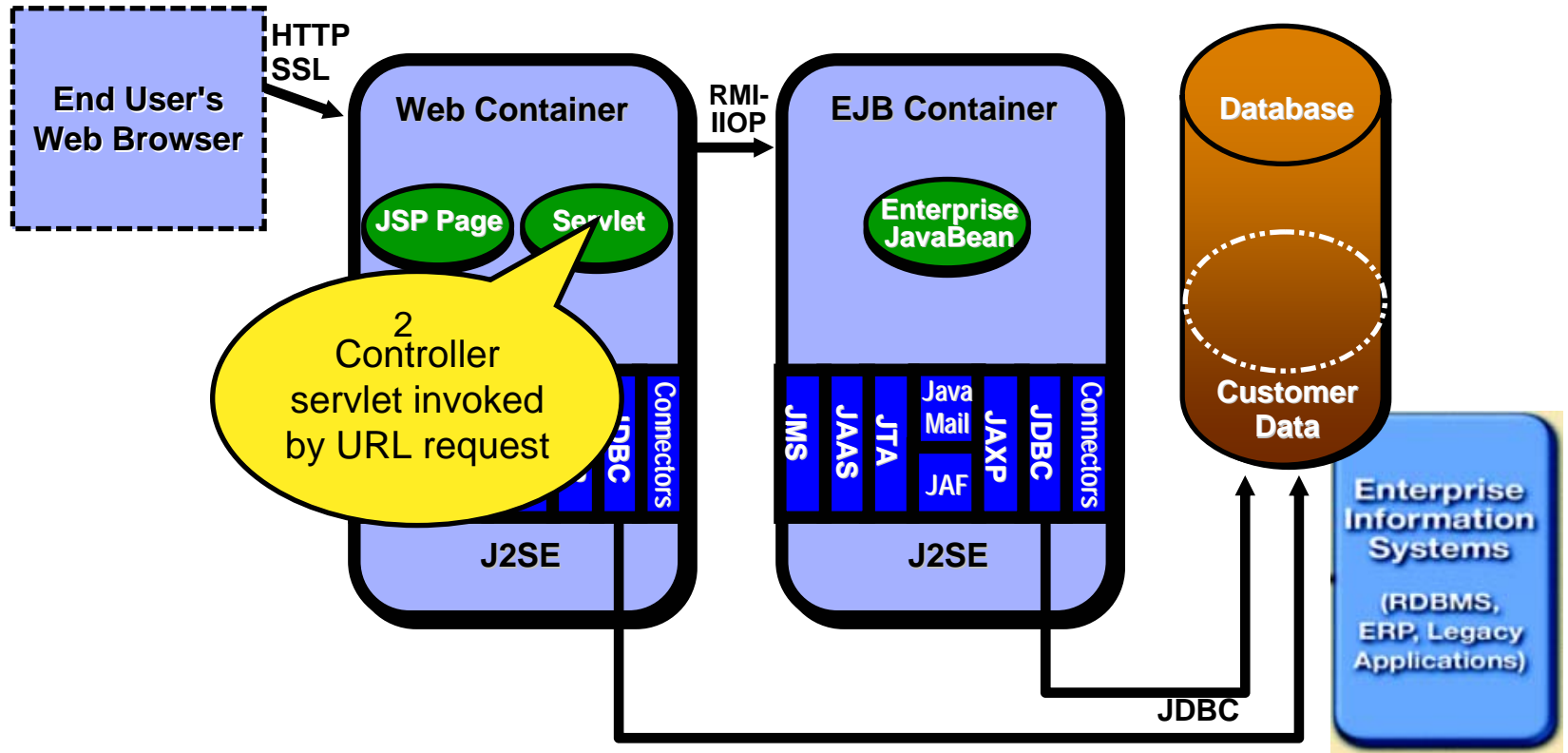
Where does an Application Server fit?



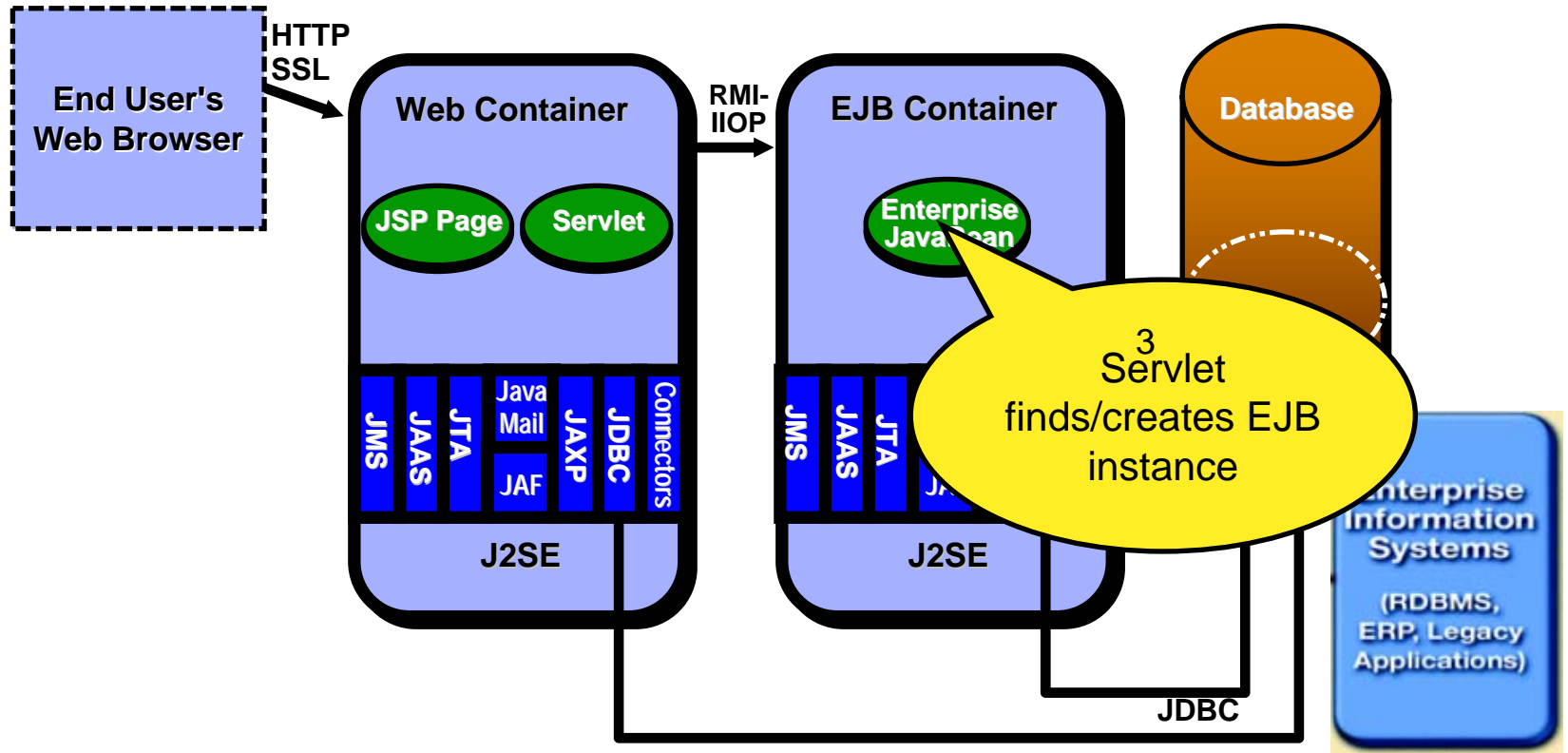
How the parts work together...



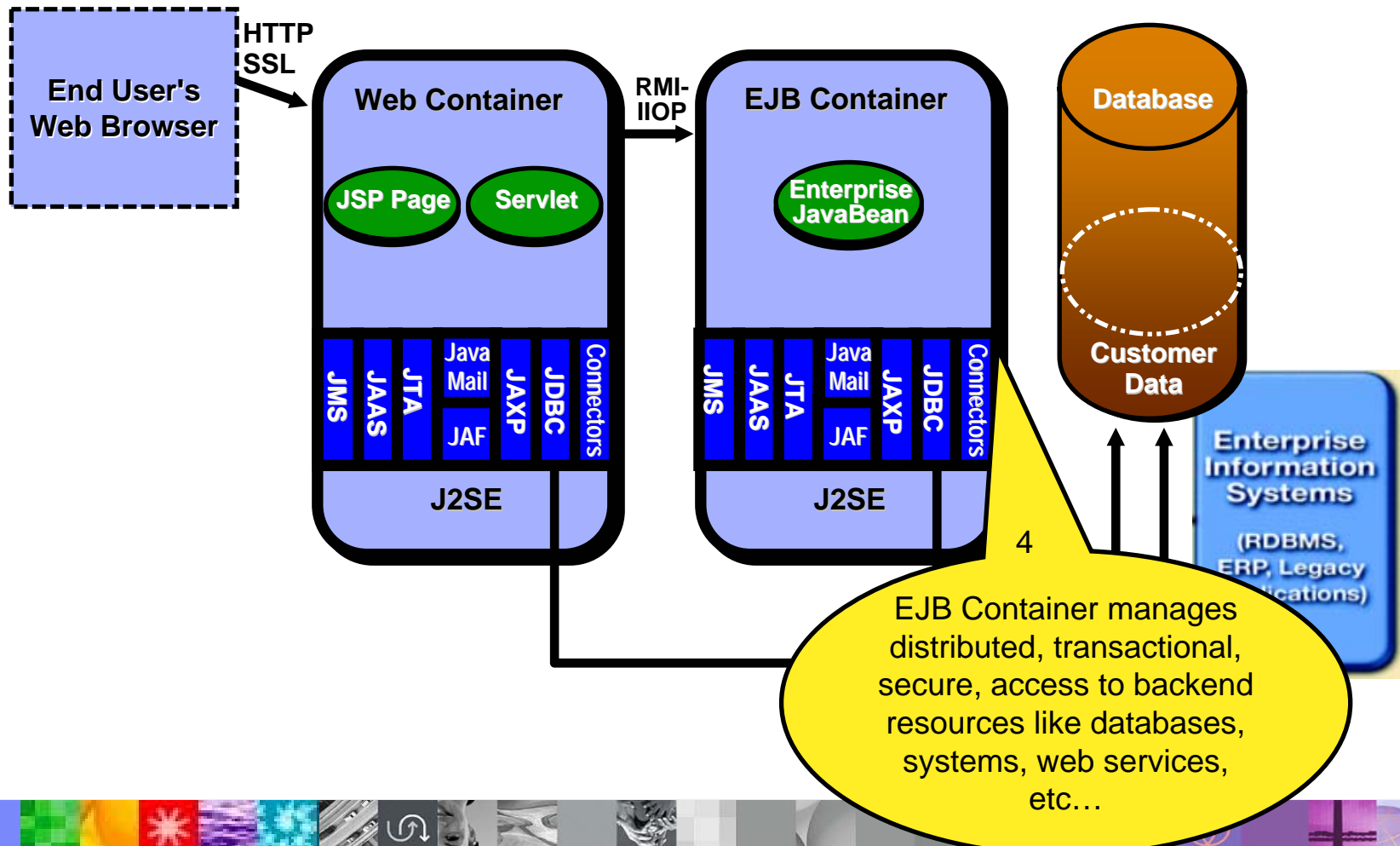
How the parts work together...



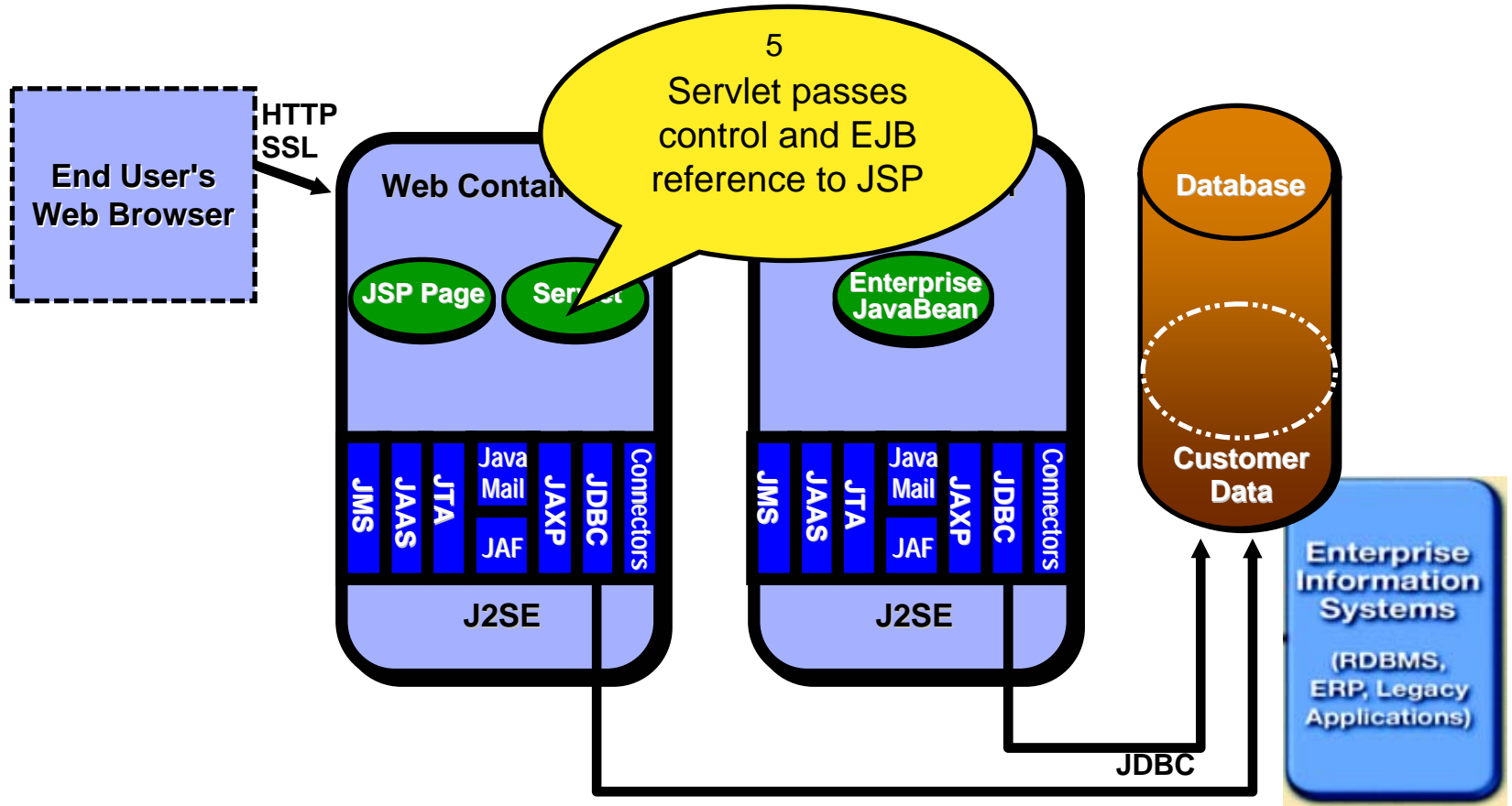
How the parts work together...



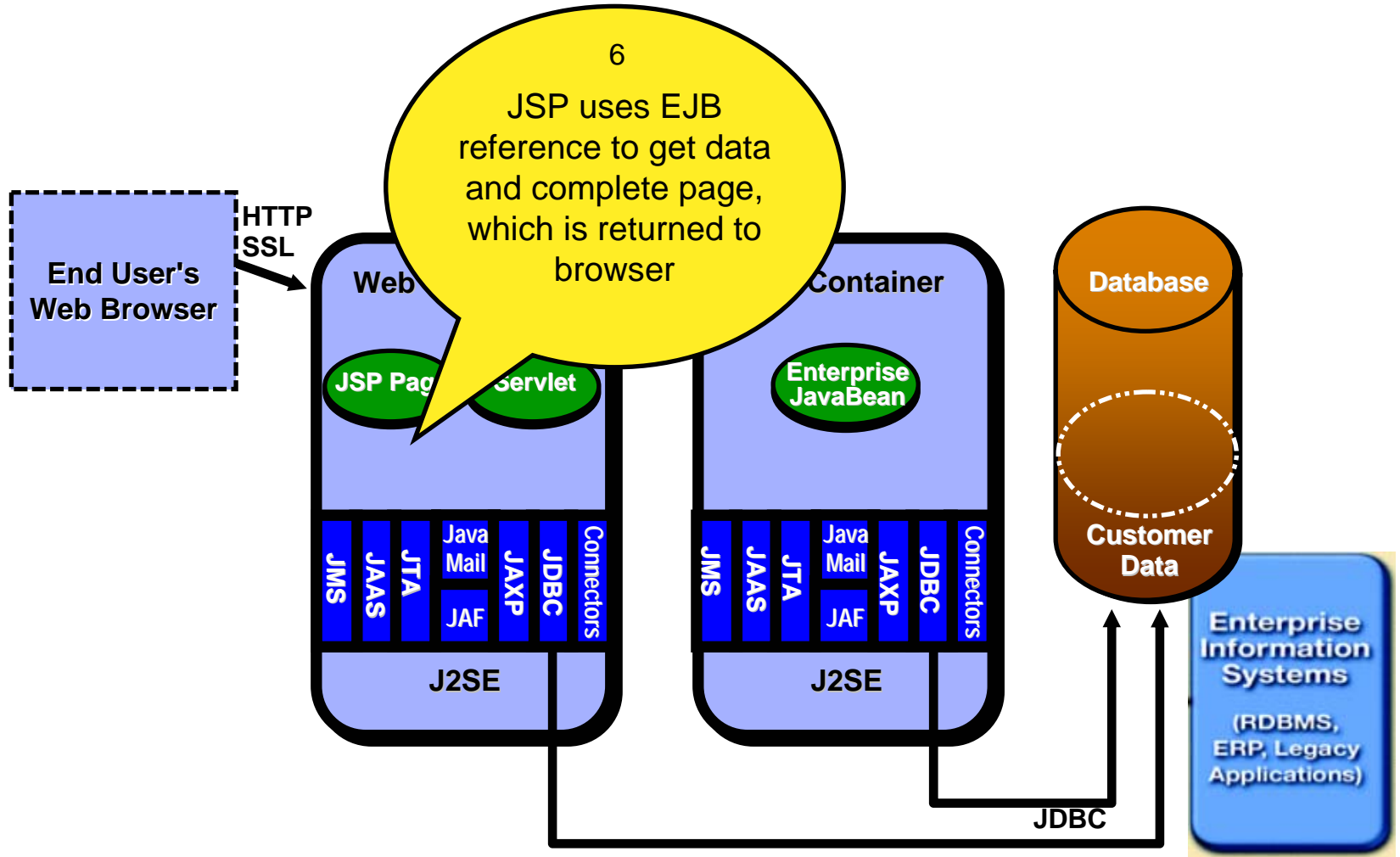
How the parts work together...



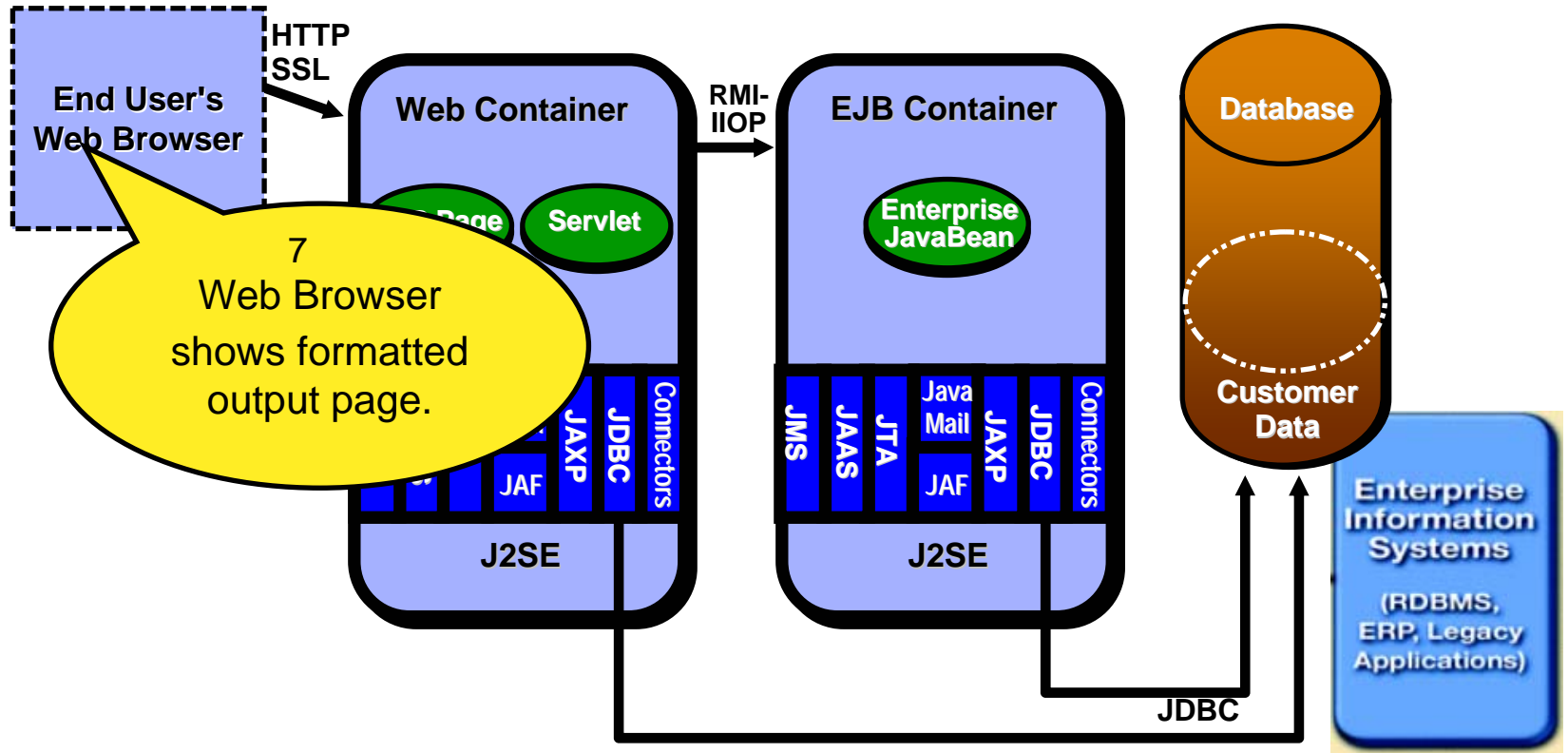
How the parts work together...



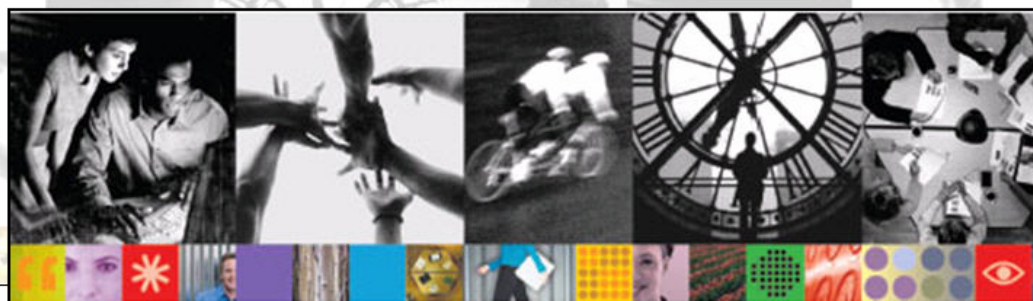
How the parts work together...



How the parts work together...



New JavaDevTools.com Tutorial Demos and more...



JavaDevTools.com

IBM Links

About Bill Hahn

JavaDevTools.com

"Navigating WebSphere, Rational, Workplace and other IBM Eclipse Based Tools..."

Homepage for: RationalCentral.com, WebSphereCentral.com, WorkplaceDev.com, JavaSOA.com, Bill-Hahn.com

Disclaimer: This is a private web site intended for IBM Eclipse Tools enthusiasts. The materials on this site are offered as-is, no support, no warranties. This is not an official IBM website. To achieve the official IBM experience, please visit <http://ibm.com>.

Welcome! Rational Application Developer (RAD) is the foundation of the IBM Software Development Platform (SDP). The **links**, **demos** and **presentations** below can speed your navigation of IBM development tools. **Executives**, **managers**, **architects** and **developers** can use this site to learn more about Rational, WebSphere, Lotus and other IBM Eclipse-based development tools.

Featured Demo

- o [RAD V6 Introduction and Tour Demo](#)

This demonstration showcases RAD in action and concisely explains the breadth and depth of RAD V6. Executives, managers, architects and developers can use this overview and the referenced resources to quickly learn about the power and value of RAD V6.

Most Popular Links

- o [Presentations and Seminar Demos \(Downloads\)](#)
- o [Eclipse - Platform, Evolution, Ecosystem, and beyond \(Article\)](#)
- o [Rational - IBM Software Development Platform \(SDP\) Overview \(Whitepaper\)](#)
- o [WebSphere - IBM Enterprise Service Bus + WBI \(Whitepaper\)](#)
- o [Workplace - IBM Workplace Overview and Resources \(website\)](#)
- o [Portal DevZone - V5.0 Whitepaper](#) and [Portal V5.1 Enhancements](#)

2005/03/24 Updates

- o [RAD V6 Tutorial Demo Series has arrived!](#)

- o [RAD V6 Tutorial Demo Series has arrived!](#)

(Getting Started Guide, Introduction and Tour)

Four scenarios with > 75 demos

- 1) End-2-End (Soup-to-nuts J2EE Build Scenarios)
- 2) Quick Success (Short Ad-hoc Build Scenarios, < 10 min each)
- 3) Banking and Finance (Web App/Site Scenario)
- 4) Financial Advisor Dashboard (Portal/Portlet Scenario)

WebSphereCentral.com
RationalCentral.com
JavaDevTools.com
JavaSOA.com



RAD V6 Tutorial Demo Series

Executives and Managers

- *Watch and Learn:* Simply view the "Tutorial Demo" movies to see ROI in action...in just minutes!

Architects and Developers

- *Watch and Do:* Use the "Tutorial Demo" movies to build the reference applications from scratch!

#1 End-2-End Application

End-2-End Page (End2EndPage.jsp and .java)
Call All

Type of Calls Input->Submit->Results

Diagrams

Web Services:
 o Temperature (www.smthods.net)
 o FastWeather (ServiceObjects.com)

Temperature Web Service FastWeather Web Service

Zip Code: 11101 Cond: Clear
 City: Long Island City
 License Key: WS14-PBJ1-00Y3 State: New York
 Temp°F: 28

XML:
 o Create XML Doc
 o Receive XML Doc

JSF:
 Client-side data:
 o Table
 o Graphs

Counter Usage

COUNTER	COUNT
CounterA	111
CounterB	200

#2 Quick Success Apps

Navigation - QuickSuccess Web Browser X

http://localhost:9060/QuickSuccess/faces/QuickHome.jsp

Welcome to the Quick Success Gallery

Home
 Weather
 Java
 EJB
 EJB Web Service
 SDO CMP Facade
 DB List
 JSF Page Data
 JSF Client Data

#3 Banking & Finance

Financial Advisor Dashboard

Bankers portfolio

Financial Advisor's Dashboard

Customers

Customer #	First Name	Last Name	City	State	Zip	Work Phone	Invested	Projected	% Rate of Return
1	Arthur	Platinum	Long Island	NY	11101	773-123-1111	\$500,000.00	\$527,751.13	5.55%
2	Betty	Golden	Arlington	VA	22202	773-123-2222	\$500,000.00	\$527,568.73	5.51%
3	Charlie	Silver	Fort Lauderdale	FL	33303	773-123-3333	\$500,000.00	\$526,688.73	5.33%
4	Dennis	Copper	Burghil	OH	44404	773-123-4444	\$500,000.00	\$510,625.00	2.12%

Customer Portfolios Performance

Bankers portfolio

Banking > Finance > Portfolio Management

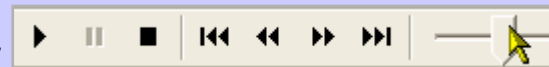
Customer # 1

Portfolio details for Arthur Platinum appear below:
 Invested: \$500,000.00 Projected: \$527,751.13 Rate of return: 5.55%

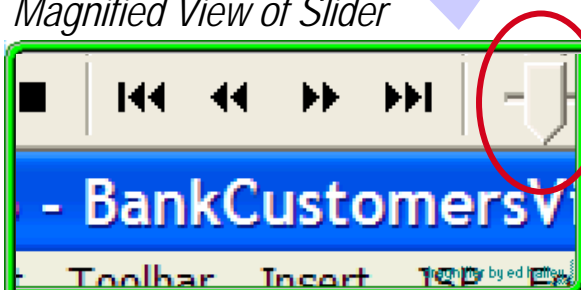
Fund Act #	Fund Name	% of Folio	Invested	Risk	% Return	Projected
101	Blended - Income Plus Life Strategy	5%	\$25,000.00	1	2.124%	\$25,531.00
102	Blended - Conservative Life Strategy	4%	\$20,000.00	2	2.249%	\$20,499.00
103	Blended - Moderate Life Strategy	6%	\$30,000.00	3	6.374%	\$31,912.00
104	Blended - Aggressive Life Strategy	7%	\$35,000.00	4	8.424%	\$37,948.40
105	Honey Market	3%	\$15,000.00	1	2.124%	\$15,318.00
106	Bond - Stable Value	3%	\$15,000.00	1	2.123%	\$15,318.75
107	Bond - Total Market	5%	\$25,000.00	2	6.375%	\$26,592.75

RAD V6 Tutorial Demo Series – How They Work...

- The embedded movie player has playback controls like a DVD player.
- The SLIDER allows accelerated forward/reverse playback control.



Magnified View of Slider



BankCustomers portlet

Bank Customers

Customers Portfolios Dashboard

Cust #	First Name	Last Name	Address	City	State	Zip	e-mail	Work Phone
1	Arthur	Platinum	123 Easy Street	Long Island	NY	11101	Cash.A.Plenty@mail.com	773-123-1111
2	Betsy	Golden	45 Plush Circle	Arlington	VA	22202	Gold.P.Lated@mail.com	773-123-2222
3	Charlie	Silver	67 Humble Avenue	Fort Lauderdale	FL	33303	Simply.E.Nuff@mail.com	773-123-3333
4	Denise	Copper	89 Trailer Park Drive	Burghill	OH	44404	Always.N.Debt@mail.com	773-123-4444

BankPortfolios portlet

Bank Portfolio Management

Customer #

Analyze Customer Portfolio

Portfolio Allocation Graphics Market Simulation

Fund Acct #	Fund Name	% of Folio	\$ Invested	Risk	% Return	\$ Projected
301	Blended - Income Plus Life Strategy	5%	\$25,000.00	1	2.125%	\$25,531.25
302	Blended - Conservative Life Strategy	5%	\$25,000.00	2	4.25%	\$26,062.50
303	Blended - Moderate Life Strategy	5%	\$25,000.00	3	6.375%	\$26,593.75
304	Blended - Aggressive Life Strategy	5%	\$25,000.00	4	8.425%	\$27,106.25
305	Money Market	5%	\$25,000.00	1	2.125%	\$25,531.25
306	Bond - Stable Value	10%	\$50,000.00	1	2.125%	\$51,062.50

Paused 00:59/01:06

End-2-End Application

Demo

"Accelerate development...
increase productivity...
maximize ROI !\$!"

Iterative Development Demos

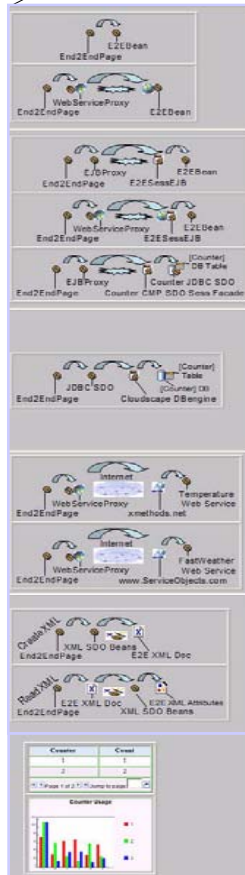
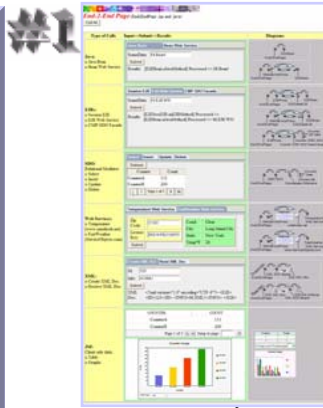
➔ Model, Build, Test, Debug, Deploy

- UML Model Visualizations
- Dynamic Development
- Automated Deployment

J2EE+ Builder Demos

➔ Includes JSF, SDO, Web Services

- Java
- Databases
- Web Services
- Drag-n-Drop GUI



Quick Success Applications

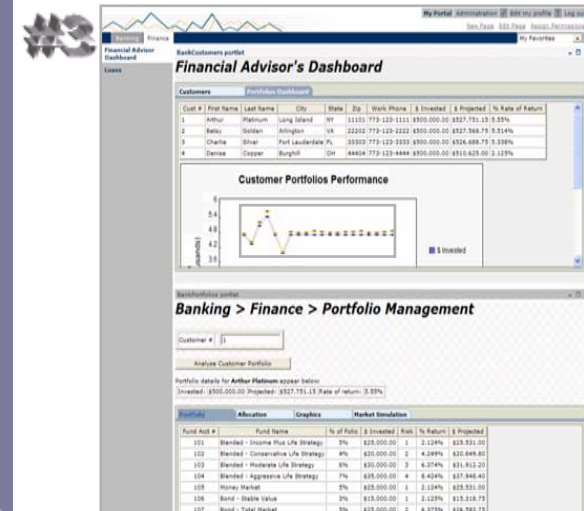
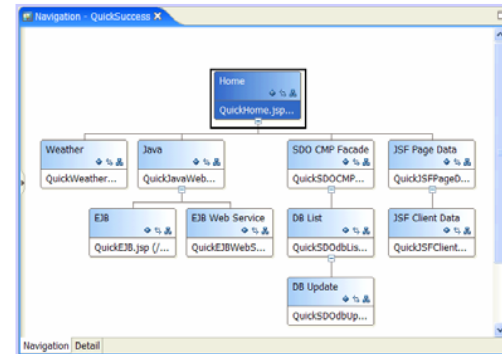
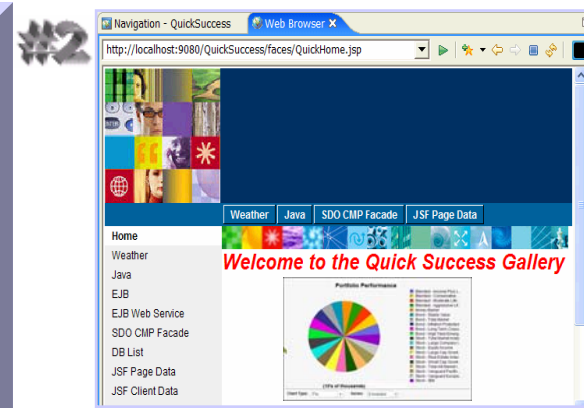
Demo

- Easy build scenarios
- You watch or build
- < 10 minutes each

Banking & Finance Web/Portal Applications

Demo

- Banking Accounts
- Portfolio Management
- Financial Advisor Dashboard





IBM Software Group

J2EE In Action!

→ Quick Success Build Scenarios ←



Name

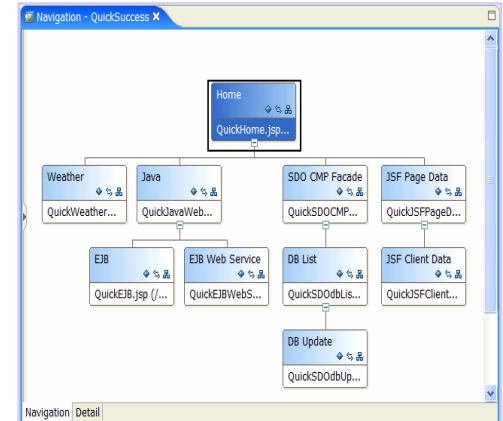
Title

Company

Quick Success Gallery Applications

- Easy J2EE build scenarios
- Internet Weather Web Service
- JavaBean Web Service
- EJB Web Service
- Service Data Objects (SDO)
 - Database Mediator: Select, Insert, Update, Delete
- Java Server Faces (JSF)
 - Rich UI Controls, Charts/Graphs, drag-n-drop
 - Custom coding, Event Driven, MVC Framework

#2



Banking & Finance Web/Portal Applications

- Transfer Funds
- Transaction History
- Account Management
- Portfolio Management
- Asset Allocation
- Graphical portfolio analysis
- Stock Market Simulation
- Mutual Funds, Stocks, Bonds, Money Mkt

#3

Financial Advisor's Dashboard

Cust #	First Name	Last Name	City	State	Zip	Work Phone	\$ Invested	\$ Projected	% Rate of Return
1	Arthur	Platinum	Long Island	NY	11101	773-123-1111	\$500,000.00	\$527,751.15	5.55%
2	Betsy	Golden	Arlington	VA	22202	773-123-2222	\$500,000.00	\$527,568.75	5.514%
3	Charlie	Silver	Fort Lauderdale	FL	33303	773-123-3333	\$500,000.00	\$526,688.75	5.338%
4	Denise	Copper	Burghill	OH	44404	773-123-4444	\$500,000.00	\$510,625.00	2.125%

Customer Portfolios Performance

Line chart showing performance in thousands of dollars (\$k) over time. The Y-axis ranges from 3.6 to 6.0. The chart shows a fluctuating line with a legend for '\$ Invested'.

Banking > Finance > Portfolio Management

Customer #

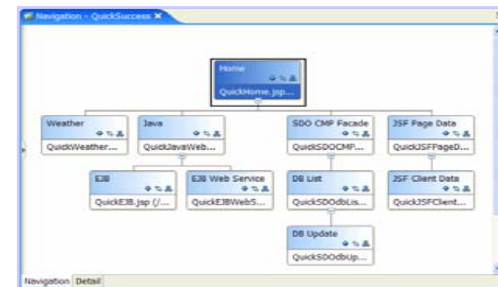
Analyze Customer Portfolio

Portfolio details for **Arthur Platinum** appear below:
 Invested: \$500,000.00 | Projected: \$527,751.15 | Rate of return: 5.55%

Fund Acct #	Fund Name	% of Folio	\$ Invested	Risk	% Return	\$ Projected
101	Blended - Income Plus Life Strategy	5%	\$25,000.00	1	2.124%	\$25,531.00
102	Blended - Conservative Life Strategy	4%	\$20,000.00	2	4.249%	\$20,849.80
103	Blended - Moderate Life Strategy	6%	\$30,000.00	3	6.374%	\$31,912.20
104	Blended - Aggressive Life Strategy	7%	\$35,000.00	4	8.424%	\$37,948.40
105	Money Market	5%	\$25,000.00	1	2.124%	\$25,531.00
106	Bond - Stable Value	3%	\$15,000.00	1	2.125%	\$15,318.75
107	Bond - Total Market	5%	\$25,000.00	2	6.375%	\$26,593.75

Quick Success Scenarios – Ad Hoc Build Tasks

- Weather Web Service – Consuming an internet based Web Service
- Java Web Service – Creating and consuming a JavaBean Web Service
- Session EJB - Creating, coding, testing, and web service enablement
- CMP EJBs – Mapping to a relational database
- CMP SDO Facade – Creating an SDO façade for a CMP EJB
- SDO CRUD – Create, retrieve, update and delete relational data
- JSF Client Data – Enhanced Browser cached JSF UI Components
- Web Site Designer – Constructing and configuring web site navigation



Quick Success Scenarios – Ad Hoc Build Tasks

- Weather Web Service – Consuming an internet based Web Service
- Java Web Service – Creating and consuming a JavaBean Web Service
- Session EJB - Creating, coding, testing, and web service enablement
- CMP EJBs – Mapping to a relational database
- CMP SDO Facade – Creating an SDO façade for a CMP EJB
- SDO CRUD – Create, retrieve, update and delete relational data
- JSF Client Data – Enhanced Browser cached JSF UI Components

Web Service Discovery Dialog

Web Services from a UDDI Registry

Select or enter a registry's inquiry URL followed by a partial or complete service name or key and click Go. You may use '%' wildcards when searching by the service name.

UDDI Registry:

Search text:

Search by: Service name Service key

UDDI Services:

Name	Description	WSDL URL
Weather - Temperature	Endpoint for service	http://www.xmethods.net/sd/2001/Ten
Weather Fetcher	Endpoint for service	http://qlkev.webs.innerhost.com/qlkev
Weather Services from EJSE, Inc.	Endpoint for service	http://www.ejse.com/WeatherService/

New: Scripting Variable, Web Service, EJB Session bean, Relational Record List, Relational Record, JavaBean

QuickWeatherWebService.jsp

http://localhost:9080/QuickSuccess/faces/QuickWeatherWebService.

Quick Weather Web Service

ZipCode:

Submit

Location: Fort Lauderdale, FL

Forecast: Mostly Cloudy

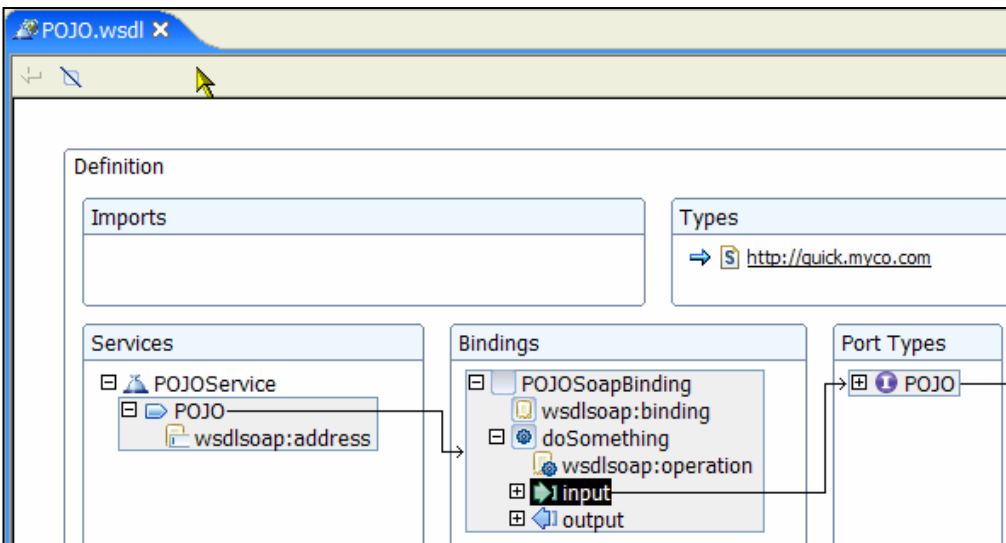
Temperature: 75°F

FeelsLike: 75°F

Humidity: 90%

Quick Success Scenarios – Ad Hoc Build Tasks

- Weather Web Service – Consuming an internet based Web Service
- Java Web Service – Creating and consuming a JavaBean Web Service
- Session EJB - Creating, coding, testing, and web service enablement
- CMP EJBs – Mapping to a relational database
- CMP SDO Facade – Creating an SDO façade for a CMP EJB
- SDO CRUD – Create, retrieve, update and delete relational data
- JSF Client Data – Enhanced Browser cached JSF UI Components



Quick Success Scenarios – Ad Hoc Build Tasks

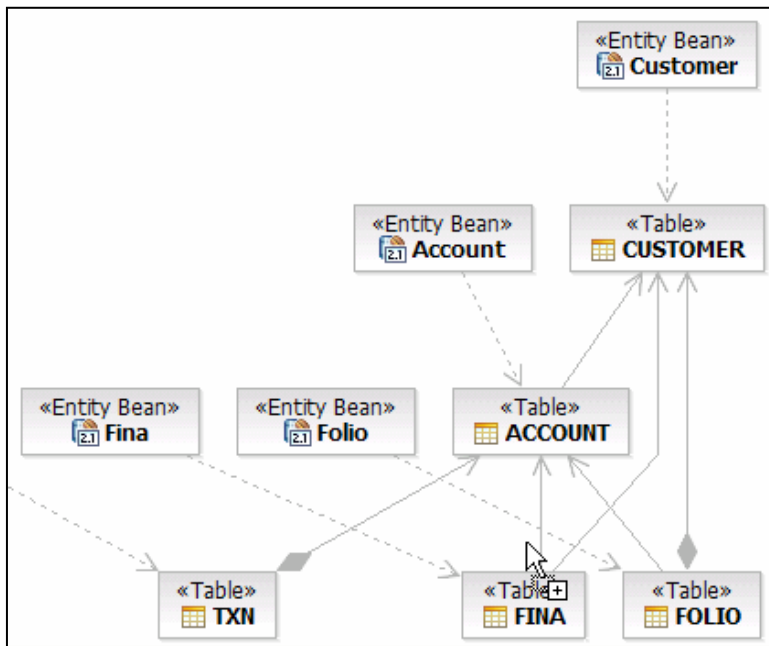
- Weather Web Service – Consuming an internet based Web Service
- Java Web Service – Creating and consuming a JavaBean Web Service
- **Session EJB** - Creating, coding, testing, and web service enablement
- CMP EJBs – Mapping to a relational database
- CMP SDO Facade – Creating an SDO façade for a CMP EJB
- SDO CRUD – Create, retrieve, update and delete relational data
- JSF Client Data – Enhanced Browser cached JSF UI Components

The screenshot shows the Universal Test Client interface. On the left, the JNDI Explorer shows the EJB Beans tree with `SessEJB` selected. The main area displays the `doSomething(String)` method with a parameter `String: Hello 123 !!!`. An `Invoke` button is highlighted. Below it, the results section shows `EJB Processed: Hello 123 !!! (java.lang.String)`. At the bottom, a list of classes is visible, including `java.util.Comparator CASE_INSENSITIVE_ORDER`, `java.io.ObjectStreamField[] serialPersistentFields`, `long serialVersionUID`, `char[] radixTenUnits`, `char[] radixTenTenths`, and `char[] nullArray`.

The screenshot shows a web browser window displaying the `QuickSuccessEJBPage.jsp` page. The URL is `http://localhost:9080/QuickSuccess/faces/QuickSuccessEJBPage.jsp`. The page features a header with the text `Quick EJB` and a form with a text input field containing `SomeData: Creating and calling anc` and a `Submit` button. Below the form, the `EJB Results:` section displays `EJB Processed: Creating and calling and EJB is easy!!!!`.

Quick Success Scenarios – Ad Hoc Build Tasks

- Weather Web Service – Consuming an internet based Web Service
- Java Web Service – Creating and consuming a JavaBean Web Service
- Session EJB - Creating, coding, testing, and web service enablement
- **CMP EJBs** – Mapping to a relational database
- CMP SDO Facade – Creating an SDO façade for a CMP EJB
- SDO CRUD – Create, retrieve, update and delete relational data
- JSF Client Data – Enhanced Browser cached JSF UI Components



Test Client x

Universal Test Client

Welcome

JNDI Explorer

EJB Beans

- CustomerLocal
- CustomerLocalHome
- CustomerLocal (com.myco.quick)
- Method Visibility
- Collection getAccount()
- String getAddr()
- String getCellph()
- String getCity()
- String getEmail()
- Collection getFina()

Parameters

java.lang.String getFirstnm()

Invoke

Results from

com.myco.quick.ejb.CustomerLocal.getFirstnm()

Arthur (java.lang.String)

Work with Object

Quick Success Scenarios – Ad Hoc Build Tasks

- Weather Web Service – Consuming an internet based Web Service
- Java Web Service – Creating and consuming a JavaBean Web Service
- Session EJB - Creating, coding, testing, and web service enablement
- CMP EJBs – Mapping to a relational database
- **CMP SDO Facade** – Creating an SDO façade for a CMP EJB
- SDO CRUD – Create, retrieve, update and delete relational data
- JSF Client Data – Enhanced Browser cached JSF UI Components

Create Session Bean Facade

Session Bean Facade
Create a session bean facade for CMP entity beans.

Name

Package Name

Source Location

Available CMP beans:

- Customer
- Account
- Fina
- Folio
- Txn

QuickSDOCMPFacadePage.jsp Web Browser

http://localhost:9080/QuickSuccess/faces/QuickSDOCMPFacadePage.

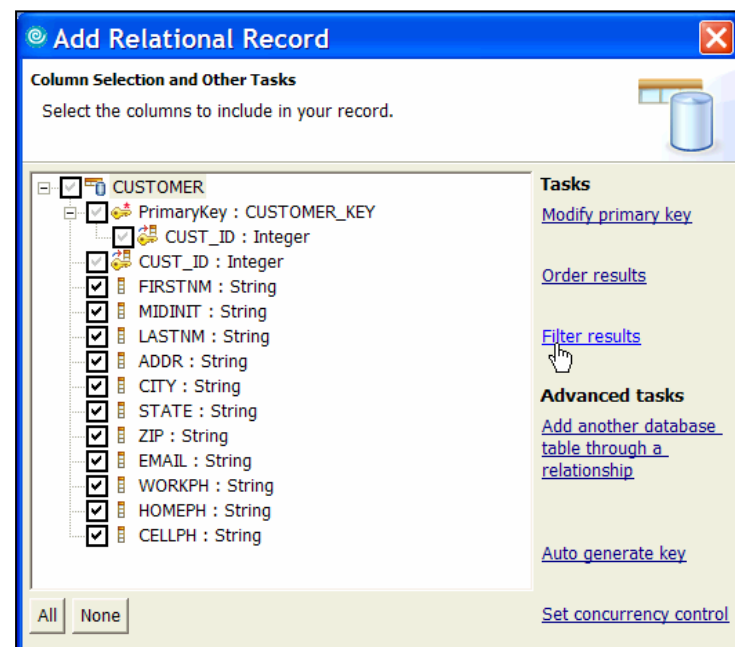
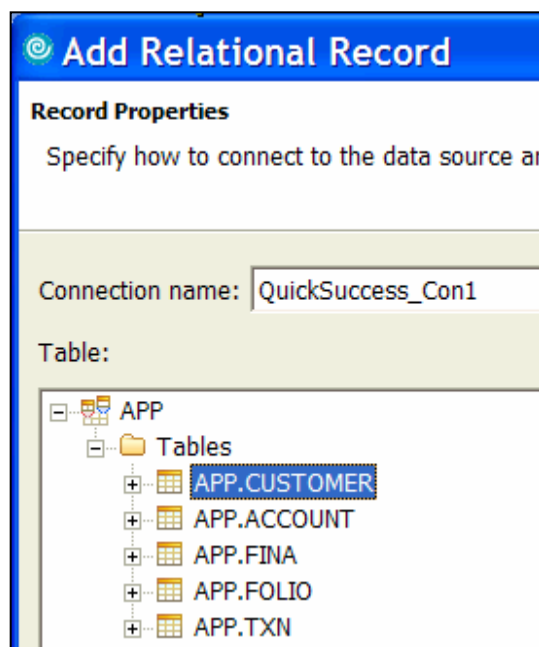
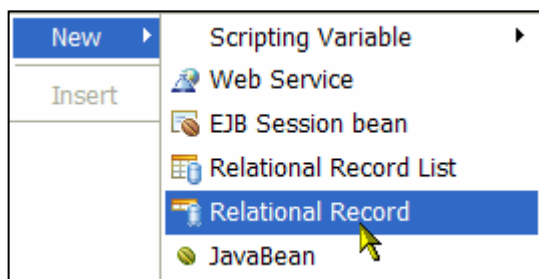
Quick SDO CMP EJB Session Bean Facade

Submit

Cust_id	Firstnm	Lastnm	Addr	City	State	Zip
1	Arthur	Platinum	123 Easy Street	Long Island	NY	11101
2	Betsy	Golden	45 Plush Circle	Arlington	VA	22202
3	Charlie	Silver	67 Humble Avenue	Fort Lauderdale	FL	33303
4	Denise	Copper	89 Trailer Park Drive	Burghill	OH	44404

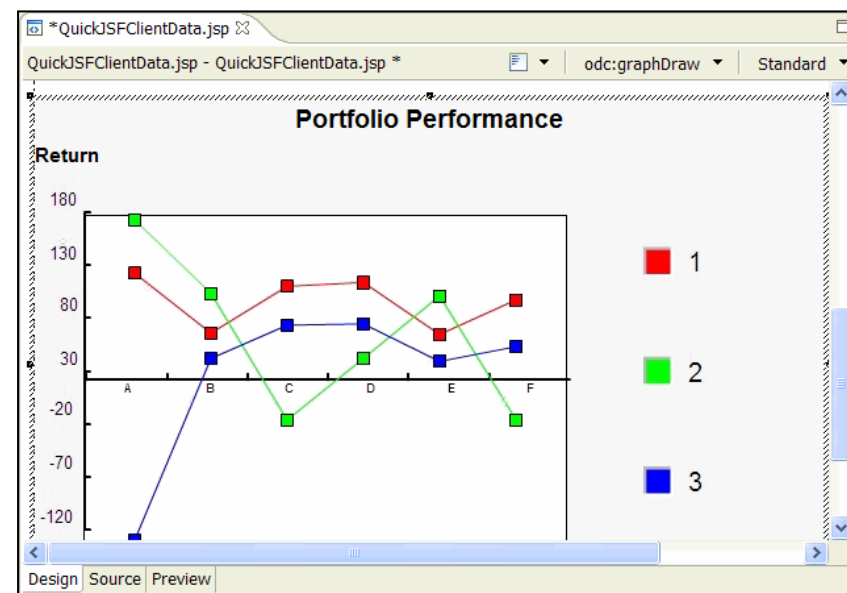
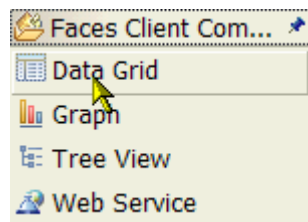
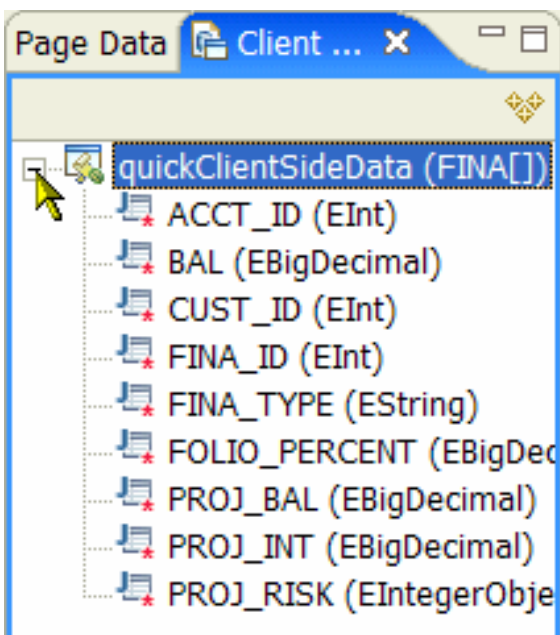
Quick Success Scenarios – Ad Hoc Build Tasks

- Weather Web Service – Consuming an internet based Web Service
- Java Web Service – Creating and consuming a JavaBean Web Service
- Session EJB - Creating, coding, testing, and web service enablement
- CMP EJBs – Mapping to a relational database
- CMP SDO Facade – Creating an SDO façade for a CMP EJB
- **SDO CRUD** – Create, retrieve, update and delete relational data
- JSF Client Data – Enhanced Browser cached JSF UI Components



Quick Success Scenarios – Ad Hoc Build Tasks

- Weather Web Service – Consuming an internet based Web Service
- Java Web Service – Creating and consuming a JavaBean Web Service
- Session EJB - Creating, coding, testing, and web service enablement
- CMP EJBs – Mapping to a relational database
- CMP SDO Facade – Creating an SDO façade for a CMP EJB
- SDO CRUD – Create, retrieve, update and delete relational data
- JSF Client Data – Enhanced Browser cached JSF UI Components





IBM Software Group

J2EE In Action!

→ *End-2-End Build Scenarios* ←



Name

Title

Company

End-2-End Application

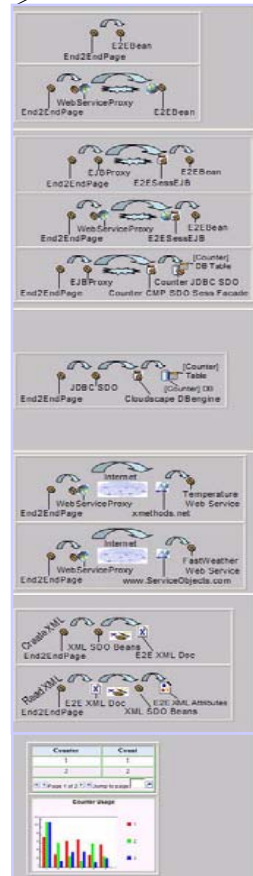
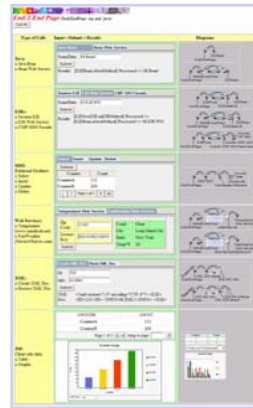
“Accelerate development...
increase productivity...
maximize ROI !\$!”

Iterative Development Demos

- ➔ Model, Build, Test, Debug, Deploy
- UML Visualizations
- Dynamic Development
- On-the-fly code changes
- Incremental build engine
- Testing for Java, EJBs, Web Services, etc...
- Dynamic Interactive Debugger
- Automated Application Packaging and Deployment

J2EE+ Builder Demos

- ➔ Includes JSF, SDO, Web Services
- Java and EJBs (Session, CMP)
- Service Data Objects (SDO)
- SDO CMP Session Facades
- XML and XSD Schemas
- XML SDO JavaBean Bindings
- JavaServer Faces (JSF)
- Database and SQL
- Web Services Creation and Consumption



End-2-End Page (End2EndPage.jsp and java)

Call All

Type of Calls	Input>>Submit>>Results	Diagrams						
Java: o Java Bean o Bean Web Service	Java Bean Bean Web Service SomeData: Hi Bean! Submit Results: [E2EBean.aJavaMethod] Processed >> Hi Bean!							
EJBs: o Session EJB o EJB Web Service o CMP SDO Facade	Session EJB EJB Web Service CMP SDO Facade SomeData: Hi EJB WS! Submit Results [E2ESessEJB.aEJBMethod] Processed >> [E2EBean.aJavaMethod] Processed >> Hi EJB WS!							
SDO: Relational Mediator: o Select o Insert o Update o Delete	Select Insert Update Delete Refresh <table border="1"> <thead> <tr> <th>Counter</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>CounterA</td> <td>111</td> </tr> <tr> <td>CounterB</td> <td>200</td> </tr> </tbody> </table> Page 1 of 2	Counter	Count	CounterA	111	CounterB	200	
Counter	Count							
CounterA	111							
CounterB	200							
Web Services: o Temperature (www.xmethods.net) o FastWeather (ServiceObjects.com)	Temperature Web Service FastWeather Web Service Zip Code: 11101 Cond: Clear City: Long Island City License Key: WS14-PBJ1-KXY3 State: New York Temp*F 28 Submit							
XML: o Create XML Doc o Recieve XML Doc	Create XML Doc Read XML Doc Id: 123 Info: Hi XML! Submit XML <?xml version="1.0" encoding="UTF-8"?><E2E> Doc: <ID>123</ID> <INFO>Hi XML!</INFO> </E2E>							
JSF: Client-side data: o Table o Graphs	<table border="1"> <thead> <tr> <th>COUNTER</th> <th>COUNT</th> </tr> </thead> <tbody> <tr> <td>CounterA</td> <td>111</td> </tr> <tr> <td>CounterB</td> <td>200</td> </tr> </tbody> </table> Page 1 of 2 Jump to page: 	COUNTER	COUNT	CounterA	111	CounterB	200	
COUNTER	COUNT							
CounterA	111							
CounterB	200							

End-2-End Scenario – Soup-2-Nuts J2EE Build Tasks

- Solution Overview – Single JSP captures all build and run scenarios
- Java Beans – Create, code, test, debug and JSF integration
- Dynamic Development – Changing code on-the-fly and debugging
- UML Visualization – Exploring application artifacts
- Performance Profiling – Analyzing performance with graphics and reports

The screenshot displays the 'End-2-End Page' interface for 'End2EndPage.jsp and .java'. It is divided into several functional areas:

- Web Services:** A section for 'Temperature Web Service' (FastWeather Web Service) with input fields for Zip Code (11101), License Key (WS14-PBJ1-KXY3), and a Submit button. Output fields show Cond: Clear, City: Long Island City, State: New York, and Temp°F: 28.
- XML:** A section for 'Create XML Doc' and 'Read XML Doc'. It shows an XML document with fields for Id (123), Info (Hi XML), and XML content: <?xml version="1.0" encoding="UTF-8"?> <E2E> <ID>123</ID> <INFO>Hi XML</INFO> </E2E>
- JSF:** A section for 'Client-side data' showing a table and a bar chart. The table lists CounterA (111) and CounterB (200). The bar chart, titled 'Counter Usage', shows usage for CounterA, CounterB, CounterC, and CounterD.
- Diagrams:** UML diagrams showing relationships between components like End2EndPage, WebServiceProxy, Internet, Temperature Web Service, and FastWeather Web Service.
- Performance:** A small table and bar chart showing counter usage data.

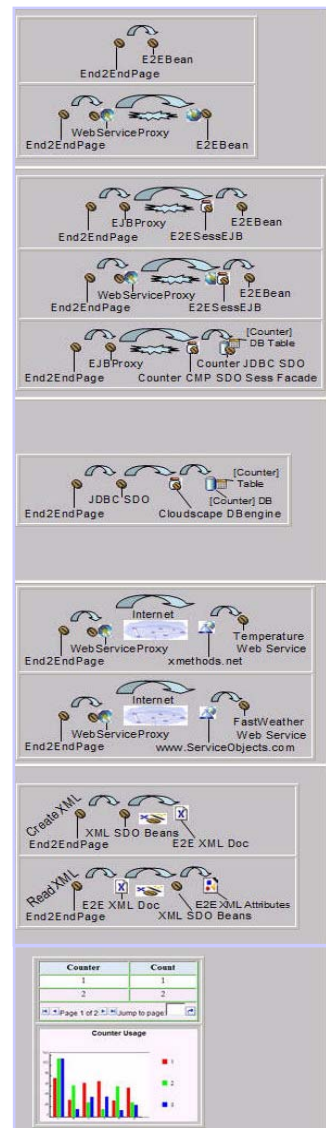
End-2-End Scenario – Soup-2-Nuts J2EE Build Tasks

- Solution Overview – Single JSP captures all build and run scenarios
- Java Beans – Create, code, test, debug and JSF integration
- Dynamic Development – Changing code on-the-fly and debugging
- UML Visualization – Exploring application artifacts
- Performance Profiling – Analyzing performance with graphics and reports

End-2-End Page (End2EndPage.jsp and .java)

Call All

Type of Calls	Input->Submit->Results	Diagrams												
Web Services: o Temperature (www.xmethods.net) o FastWeather (ServiceObjects.com)	<table border="1"> <tr> <td>Temperature Web Service</td> <td>FastWeather Web Service</td> </tr> <tr> <td>Zip Code: 11101</td> <td>Cond: Clear</td> </tr> <tr> <td>License Key: WS14-PBJ1-KCY3</td> <td>City: Long Island City</td> </tr> <tr> <td>Submit</td> <td>State: New York</td> </tr> <tr> <td></td> <td>Temp*F: 28</td> </tr> </table>	Temperature Web Service	FastWeather Web Service	Zip Code: 11101	Cond: Clear	License Key: WS14-PBJ1-KCY3	City: Long Island City	Submit	State: New York		Temp*F: 28			
Temperature Web Service	FastWeather Web Service													
Zip Code: 11101	Cond: Clear													
License Key: WS14-PBJ1-KCY3	City: Long Island City													
Submit	State: New York													
	Temp*F: 28													
XML: o Create XML Doc o Recieve XML Doc	<table border="1"> <tr> <td>Create XML Doc</td> <td>Read XML Doc</td> </tr> <tr> <td>Id: 123</td> <td></td> </tr> <tr> <td>Info: Hi XML!</td> <td></td> </tr> <tr> <td>Submit</td> <td></td> </tr> <tr> <td>XML: <?xml version="1.0" encoding="UTF-8"?><E2E></td> <td></td> </tr> <tr> <td>Doc: <ID>123</ID><INFO>Hi XML!</INFO></E2E></td> <td></td> </tr> </table>	Create XML Doc	Read XML Doc	Id: 123		Info: Hi XML!		Submit		XML: <?xml version="1.0" encoding="UTF-8"?><E2E>		Doc: <ID>123</ID><INFO>Hi XML!</INFO></E2E>		
Create XML Doc	Read XML Doc													
Id: 123														
Info: Hi XML!														
Submit														
XML: <?xml version="1.0" encoding="UTF-8"?><E2E>														
Doc: <ID>123</ID><INFO>Hi XML!</INFO></E2E>														
JSF: Client-side data: o Table o Graphs	<table border="1"> <thead> <tr> <th>COUNTER</th> <th>COUNT</th> </tr> </thead> <tbody> <tr> <td>CounterA</td> <td>111</td> </tr> <tr> <td>CounterB</td> <td>200</td> </tr> </tbody> </table> <p>Page 1 of 2 Jump to page: []</p>	COUNTER	COUNT	CounterA	111	CounterB	200							
COUNTER	COUNT													
CounterA	111													
CounterB	200													



End-2-End Scenario – Soup-2-Nuts J2EE Build Tasks

- Solution Overview – Single JSP captures all build and run scenarios
- Java Beans – Create, code, test, debug and JSF integration
- Dynamic Development – Changing code on-the-fly and debugging
- UML Visualization – Exploring application artifacts
- Performance Profiling – Analyzing performance with graphics and reports

The screenshot shows the IBM Rational Software Development Platform interface during a debug session. The main editor displays the source code of `E2EBean.java`, which includes a package declaration, a class declaration, and a method `aJavaMethod`. A context menu is open over the `someData` variable in the method, with the `Change Value...` option selected. The Variables view on the right shows the current state of the variables: `this = E2EBean (id=48)` and `someData = "Hi"`. The Package Explorer on the bottom right shows the project structure, including the `com.myc0.e2e.java` package and the `E2EBean` class.

```
/*
 * Created on Jan 30, 2005
 */
package com.myc0.e2e.java;

/**
 * @author Bill Hahn
 */
public class E2EBean {
    public String aJavaMethod(String someData) {
        return "[E2EBean.aJavaMethod] Processed >>" + someData;
    }
}
```

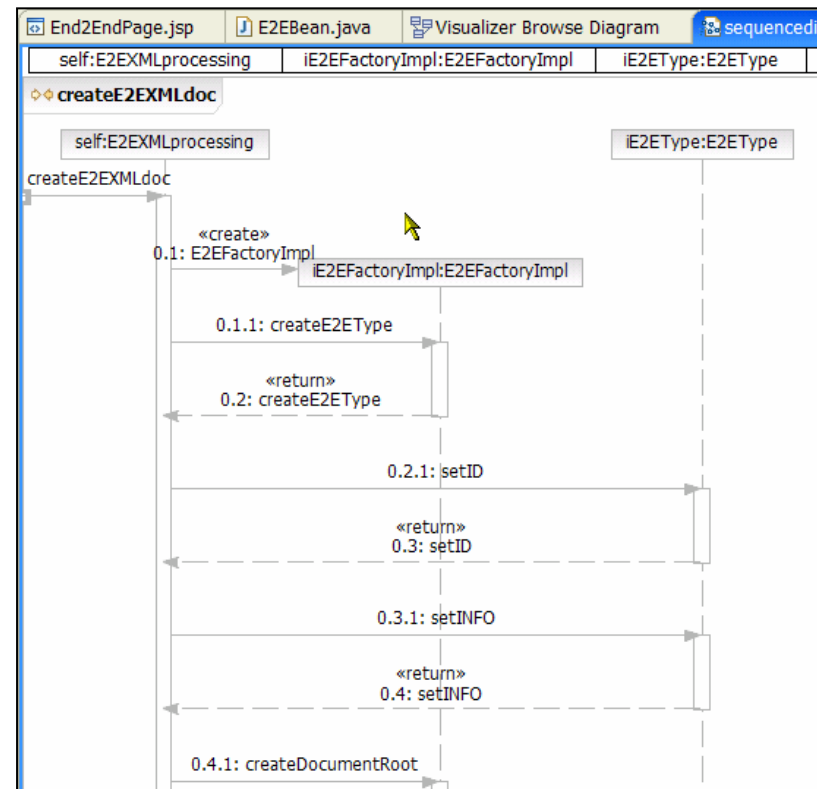
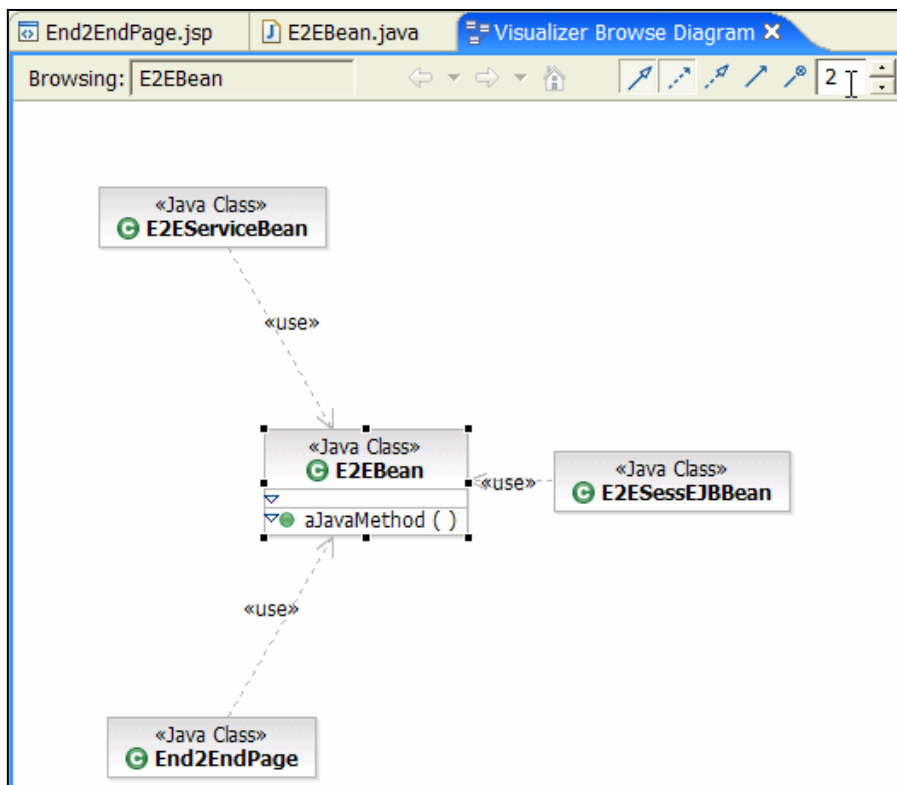
End-2-End Scenario – Soup-2-Nuts J2EE Build Tasks

- Solution Overview – Single JSP captures all build and run scenarios
- Java Beans – Create, code, test, debug and JSF integration
- Dynamic Development – Changing code on-the-fly and debugging
- UML Visualization – Exploring application artifacts
- Performance Profiling – Analyzing performance with graphics and reports

Type of Calls	Input>>Submit>>Results	Diagrams
Java: o Java Bean o Bean Web Service	<div style="border: 1px solid gray; padding: 5px;"> <div style="border-bottom: 1px solid gray; display: flex; justify-content: space-between;"> Java Bean Bean Web Service </div> <div style="padding: 5px;"> SomeData: <input type="text"/> <input type="button" value="Submit"/> </div> <div style="padding: 5px;"> Results: [E2EServiceBean.aWSJavaMethod] Processed >> [E2EBean.aJavaMethod] Processed >> [BP #2] </div> </div>	<div style="border: 1px solid gray; padding: 5px;"> <pre> graph TD End2EndPage --> E2EBean </pre> </div> <div style="border: 1px solid gray; padding: 5px;"> <pre> graph TD End2EndPage --> WebServiceProxy WebServiceProxy --> E2EBean </pre> </div>
EJBs: o Session EJB o EJB Web Service o CMP SDO Facade	<div style="border: 1px solid gray; padding: 5px;"> <div style="border-bottom: 1px solid gray; display: flex; justify-content: space-between;"> Session EJB EJB Web Service CMP SDO Facade </div> <div style="padding: 5px;"> SomeData: <input type="text"/> <input type="button" value="Submit"/> </div> <div style="padding: 5px;"> Results: [E2ESessEJB.anEJBMethod] Processed >> [E2EBean.aJavaMethod] Processed >> [BP #3] </div> </div>	<div style="border: 1px solid gray; padding: 5px;"> <pre> graph TD End2EndPage --> EJBProxy EJBProxy --> E2ESessEJB </pre> </div> <div style="border: 1px solid gray; padding: 5px;"> <pre> graph TD End2EndPage --> WebServiceProxy WebServiceProxy --> E2ESessEJB </pre> </div>

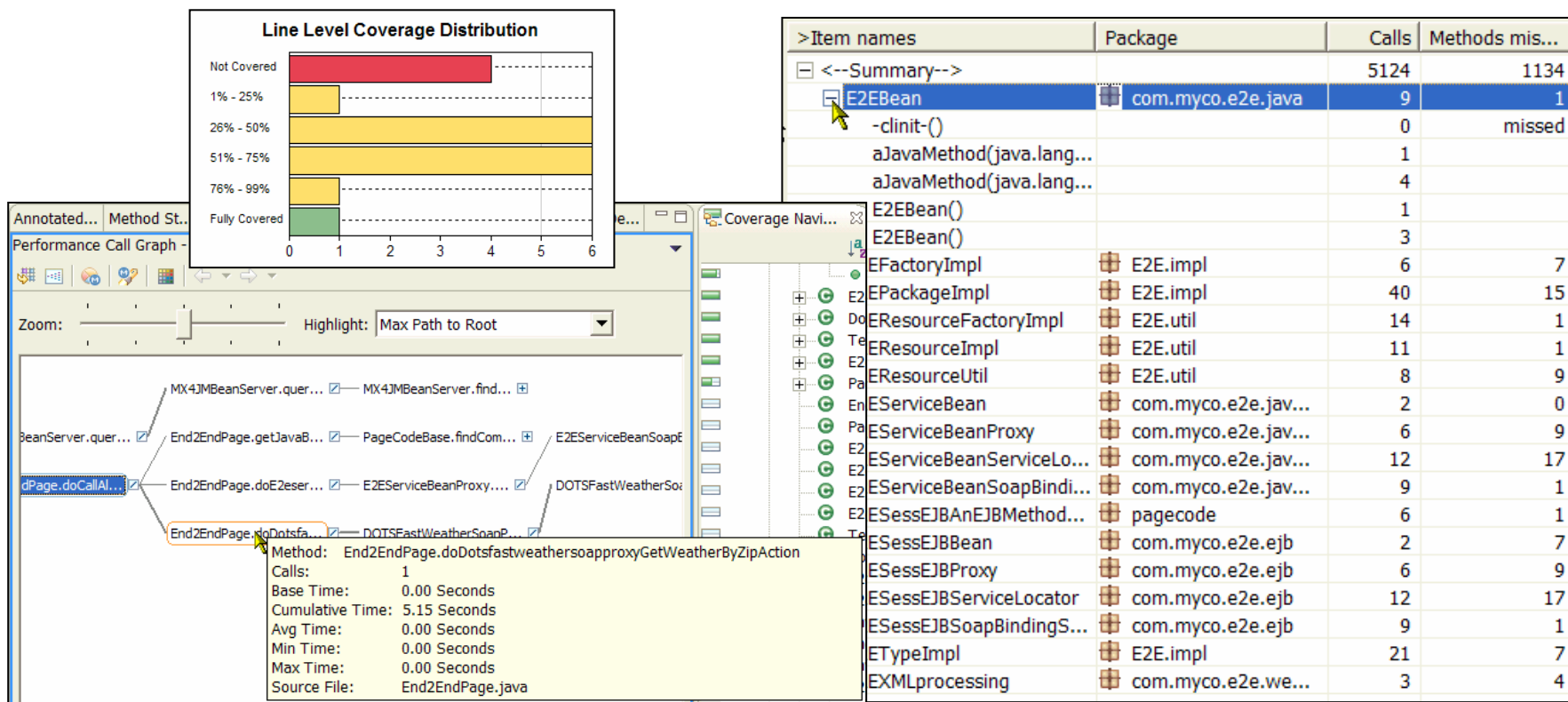
End-2-End Scenario – Soup-2-Nuts J2EE Build Tasks

- Solution Overview – Single JSP captures all build and run scenarios
- Java Beans – Create, code, test, debug and JSF integration
- Dynamic Development – Changing code on-the-fly and debugging
- **UML Visualization** – Exploring application artifacts
- Performance Profiling – Analyzing performance with graphics and reports




End-2-End Scenario – Soup-2-Nuts J2EE Build Tasks

- Solution Overview – Single JSP captures all build and run scenarios
- Java Beans – Create, code, test, debug and JSF integration
- Dynamic Development – Changing code on-the-fly and debugging
- UML Visualization – Exploring application artifacts
- **Performance Profiling** – Analyzing performance with graphics and reports



Resources

- <http://ibm.com/developer> 

(IBM developerWorks for technologies and/or products)
 - Java, J2EE, Web Services and other technology zones are on the left hand side
 - WebSphere Zone: <http://ibm.com/developer/websphere>
 - Rational Zone: <http://ibm.com/developer/rational>
 - Free Education Portal: <http://ibm.com/developer/training>
- <http://www.redbooks.ibm.com>

(How-To Step-by-step Practical Implementation Books)
- Sun Java/J2SE/J2EE Tutorials: [Java Tutorial Online](#) , [J2EE Tutorial Online](#) , [J2EE Tutorial BOOK](#)
- <http://JavaDevTools.com> (a.k.a. JavaSOA.com, RationalCentral.com, WebSphereCentral.com)

(Tutorials, Links, Presentations, White Papers, Articles, etc.)

developerWorks
DB2 ▶
eServer ▶
Lotus ▶
Rational ▶
Tivoli ▶
WebSphere ▶
Autonomic computing ▶
Grid computing ▶
Java™ technology ▶
Linux ▶
Open source ▶
Power Architecture™ ▶
SOA and Web services ▶
Web architecture ▶
Wireless technology ▶
XML ▶
Feedback
Related links
• ISV resources
• alphaWorks (emerging technologies)
• Academic program
• IBM Redbooks
• IBM Press books

