Building Java Applications with JavaFX 8 and Java EE 7 [CON2150]

Being a smart Java developer on both client and server-side

Bruno Borges Principal Product Manager – Java Evangelist Oracle Latin America September, 2014



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



Speaker



Bruno Borges

Principal Product Manager, Java Evangelist Oracle Latin America

@brunoborges
bruno.borges@oracle.com



Copyright © 2014, Oracle and/or its affiliates. All rights reserved. |

Program Agenda

- JavaFX Desktop Applications Today
- ² Exposing Server-side Data with Java EE 7
- ³ Java EE 7 APIs for Client-side
- Putting all together





JavaFX Desktop Applications



Copyright © 2014, Oracle and/or its affiliates. All rights reserved. |

Usual Desktop Architecture

Presentation tier

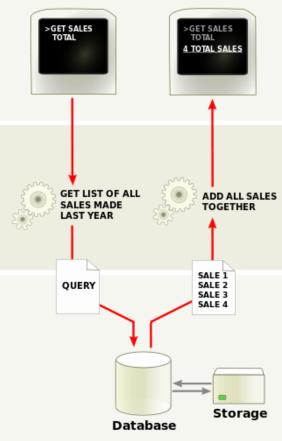
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



Source: Wikipedia



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

JavaFX Frameworks







Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Java EE 7

Not just for Web Development



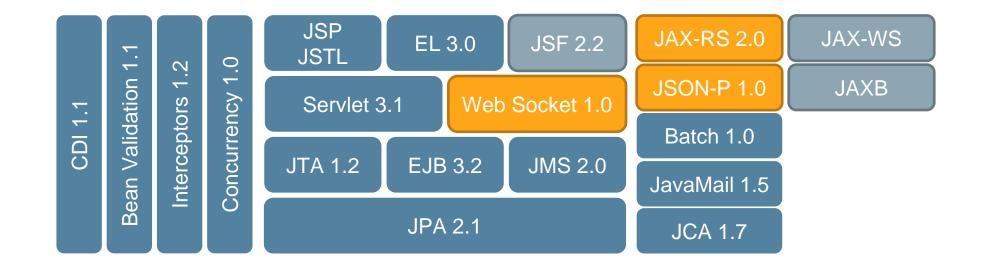
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Java EE 7 APIs





Java EE 7 APIs for Exposing Data





JSR 356: Java API for WebSocket 1.0

Annotated ServerEndpoint Example

@ServerEndpoint("/echo")

public class EchoServer {

@OnMessage

public void onMessage(Session session, String message) {

session.getOpenSessions()
 .forEach(s ->s.getAsyncRemote().sendText(message));



```
JSR 339 – JAX-RS 2.0
```

Annotated REST Service

@Path("/chat")

public class ChatServer {

@GET @Produces("plain/text")

@Consumes("plain/text")

public String echo(String message) {

return "Echoing via REST the message: "+message;

. . .



JSR 372 – JavaServer Faces 2.3

- Build webpages and embed them in JavaFX applications with WebView
- Inject real Java objects by manipulating the DOM tree in WebView
 - Do this instead of running Applets in webpages loaded on browsers
 - Consider this for applications that require fast layout changes/evolution
- Part of your application is dynamically loaded from Web
 - Or even the whole screen is loaded as web page?



Java EE 7 APIs for Client-side





Java EE 7 APIs To Be Used on Client-side



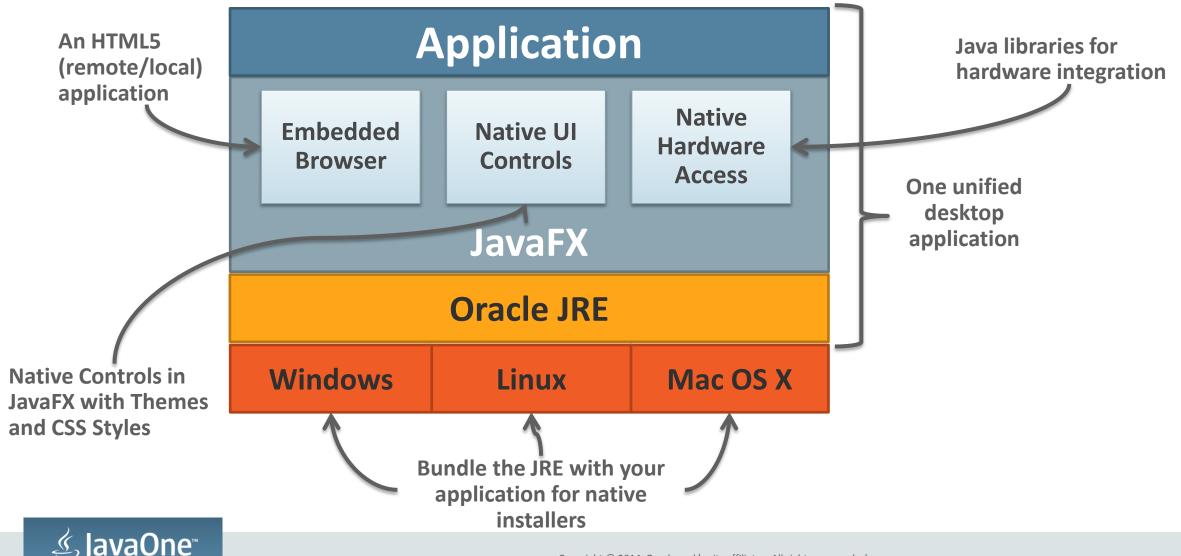


Java EE 7 APIs You Can Use but Likely Won't...



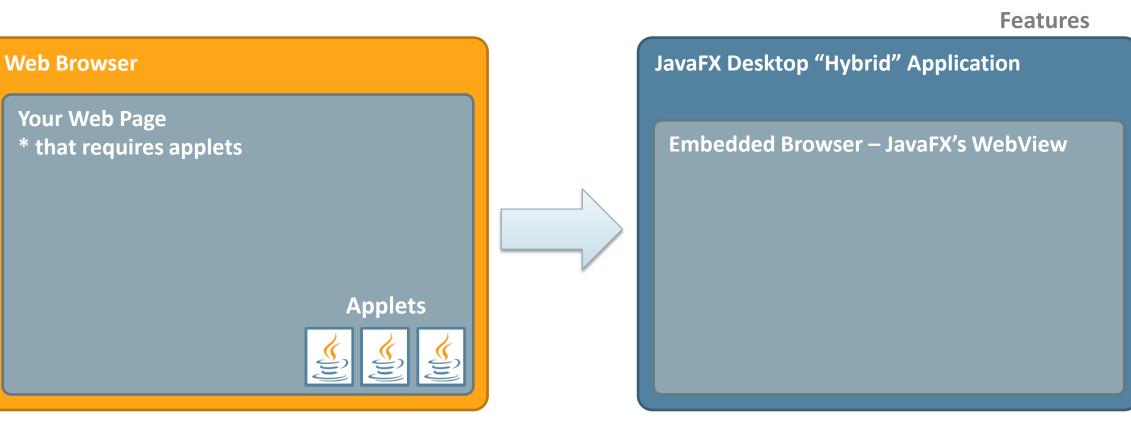


Hybrid JavaFX Desktop Application – Architecture



JavaFX Hybrid Apps

Building Desktop Apps instead of Web Apps with Applets



Native



Copyright © 2014, Oracle and/or its affiliates. All rights reserved. |

Demo



Copyright © 2014, Oracle and/or its affiliates. All rights reserved. |

Adopt-A-JSR

- Want Java EE APIs to be aligned with Desktop/JavaFX Applications?
- Want a Java EE "Client Profile"?

glassfish.org/adoptajsr



Oracle Confidential –

JavaOne™ ORACLE