

**ORACLE®**

ORACLE®

# HTML5 Application Development with Java

Peter Doschkinow  
Senior Java Architect

MAKE THE  
FUTURE  
JAVA

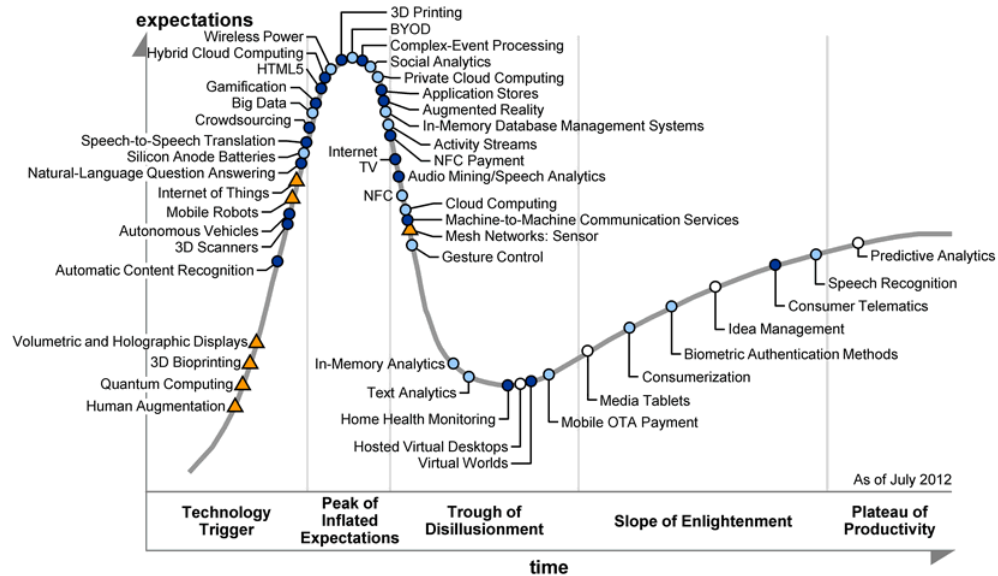
ORACLE®

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.

# Agenda

- Motivation
- HTML5 Overview
  - Related Java Technologies
- Thin Server Architecture
- Demo

# Motivation



Plateau will be reached in:  
 O less than 2 years   ● 2 to 5 years   ● 5 to 10 years   ▲ more than 10 years   ⊗ obsolete before plateau

Gartner's 2012 Emerging Technologies Hype Cycle

- Need for clarification
  - What is behind the hype
- Architectural consequences of new trends
- What offers the Java platform to meet the new challenges
- Building of common understanding

# Web Technology History

- 1991 HTML
- 1994 HTML2
- 1996 CSS1
- 1997 HTML4
- 1998 CSS2
- 2000 XHTML1
- 2002 Tableless Web Design
- 2005 AJAX
- 2009 HTML5: as of Dec 2012 W3C CR
- 1995 JavaScript @ Netscape
- 1996 ECMAScript 1.0, 1.1
- 1997 ECMAScript 1.2
- 1998 ECMAScript 1.3
- 2000 ECMAScript 3
- 2010 ECMAScript 5
- Next: ECMAScript 6 Harmony

# HTML5 Features

W3C / Web Hypertext Application Technology Working Group(WHATWG)

- Markup
  - Semantic markup replacing common usages of generic `<span>`, `<div>`
    - `<nav>`, `<footer>`, `<audio>`, `<video>`, ...
- API
  - Canvas 2D (for immediate mode 2D drawing), Timed media playback
  - Offline Web Applications, Local Storage and Filesystem, Web Storage
  - Geolocation, Web Storage, IndexedDB
  - File API, Drag-and-Drop, Browser History
  - ...

# HTML5 Features

Offloaded to other specs, originally part of HTML5

- **WebSocket API, Server-Sent Events(SSE)**, Web Messaging, Web Workers, Web Storage (Web Apps WG )
- **WebSocket Protocol** (IETF HyBi WG)
- WebRTC (WebRTC WG )
- Canvas 2D (HTML WG)
- ...



# HTML5 Standards Association

Device



Geolocation  
Device orientation and motion  
Multimedia

Data



Web storage, Offline Web Applications  
File System, Indexed database  
**Web socket**  
**Server-sent events**

Logic



Web workers  
Touch events

+



UI



Elements  
Canvas  
Svg, webgl

+



# HTML5 Related Technologies at Oracle

- ADF Mobile and JavaFX
  - Contain WebView component, that uses open source browser engine WebKit
- JAX-RS, WebSocket, JSON
  - Part of Java EE 7, implemented in GlassFish 4.0, TBD in WebLogic
- Server-Sent Events
  - Implemented in GlassFish 4.0, TBD in WebLogic
- Partially supported in JSF 2.2, part of Java EE 7
- HTML5 support in NetBeans

# HTML5 Browser Support and Demos

- Browser test and support
  - <http://acid3.acidtests.org>
  - <http://caniuse.com>

- Amazing presentation of HTML5 features

- <http://slides.html5rocks.com>

- HTML5 Canvas 3D (WebGL)

- [http://oos.moxiecode.com/js\\_webgl/fish/index.html](http://oos.moxiecode.com/js_webgl/fish/index.html)

- [http://oos.moxiecode.com/js\\_webgl/world/index.html](http://oos.moxiecode.com/js_webgl/world/index.html)

# Web Sockets - Working Draft		Usage stats: Global							
<i>Bidirectional communication technology for web apps</i>		Support:	57.1%						
		Partial support:	4.64%						
		Total:	61.74%						
Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
								2.1	2.1
								2.2	2.2
								2.3	2.3
	7.0	16.0				3.2		3.0	3.0
	8.0	17.0	23.0			4.2-4.3		4.0	4.0
	9.0	18.0	24.0	5.1		5.0-5.1		4.1	4.1
Current	10.0	19.0	25.0	6.0	12.1	6.0	5.0-7.0	4.2	7.0
Near future		20.0	26.0		12.5				10.0
Farther future		21.0	27.0						

Notes Known issues (0) Resources (4) Feedback Edit on GitHub

# Modern Web Development

## Exciting Industry Trend



- It's difficult and potentially costly to build modern web applications
  - Web? Native? Flash? Build for many? Build for one? Form factor?
  - Expertise, development cost, testing and support across platforms
- HTML5 is designed to address the cross-platform jungle
  - Attempts to codify best-practices that have emerged
  - Well suited for mobile devices

# HTML5 Architectural Implications

## The Browser Is the Platform

- HTML5 is the new UI across devices
  - Multimedia, Graphics, Offline, Real-time Communication, Device Access, File access, Semantic markup, CSS3
  - Applications == HTML5 + JavaScript + CSS3 + Server Resources
- Requires a different programming approach
  - Servers no longer generating markup language
  - Clients responsible for presentation logic and execution
  - JavaScript is part of the domain model, JSON is the payload
  - Event-Driven
  - No need for browser plugin

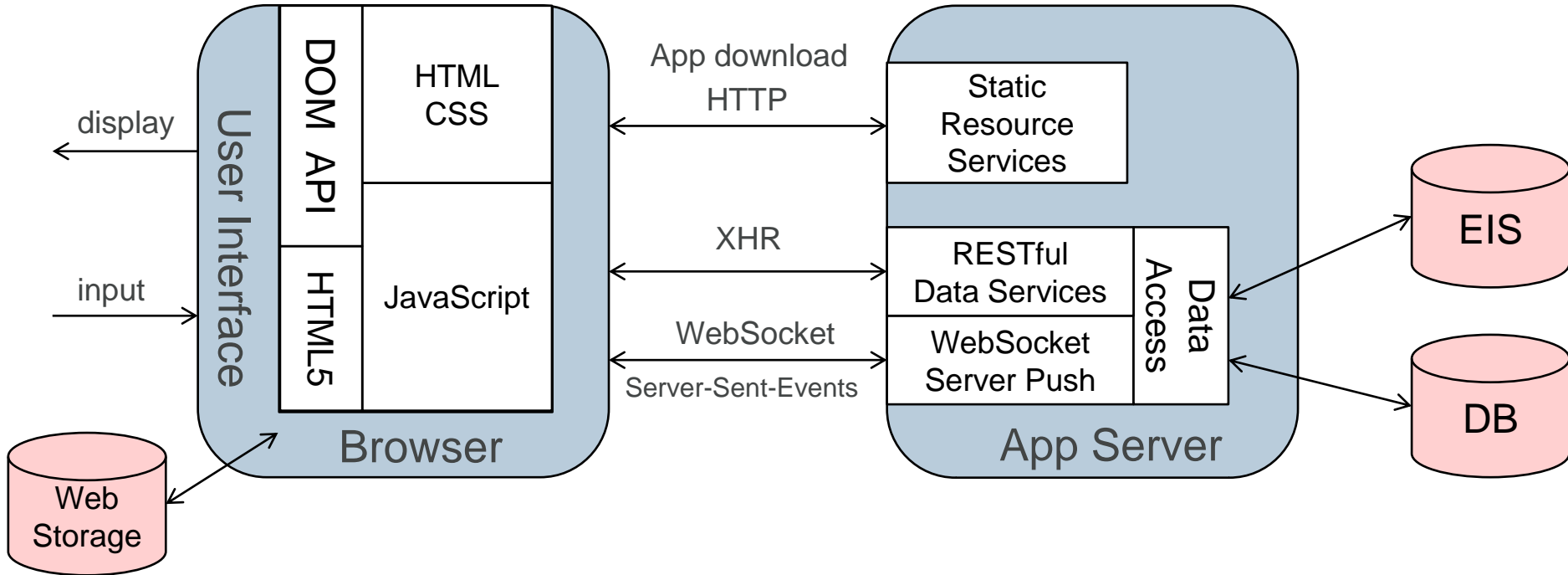
# Thin Server Architecture (TSA)

## Background

- Main idea: move the presentation layer to the client. The server is responsible for providing access to the application data and for serving the static resources that implement the presentation layer.
- Similar architectures
  - SOFEA: Service-Oriented Front-End Architecture
  - RIA: Rich Internet Application (Flash, Silverlight, JavaFX)
  - SPA: Single Page Application
    - AJAX, browser plugins (for Flash, Silverlight, JavaFX)
- [www.thinserverarchitecture.com](http://www.thinserverarchitecture.com) (2008)

# Thin Server Architecture Diagram

Runtime application presentation



# Thin Server Architecture

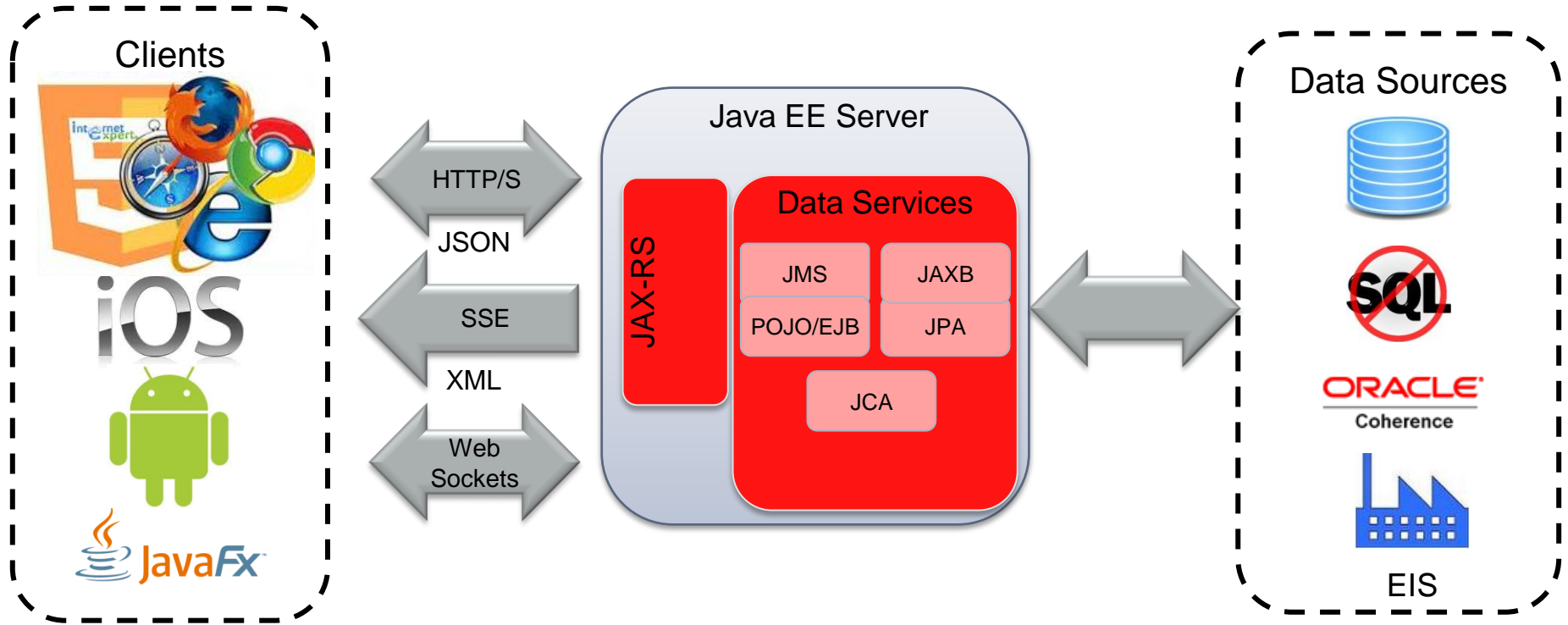
## Advantages

- Improved performance
  - Caching, no presentation data transmitted again and again
- Scalability
  - Less data to transfer, session state is on the client
- Reduced complexity
  - UI control is not split between client and server, UI events stay on client
- Improved user experience
- Offline support only possible with TSA



# Thin Server Architecture

With Java EE



# Thin Server Architecture

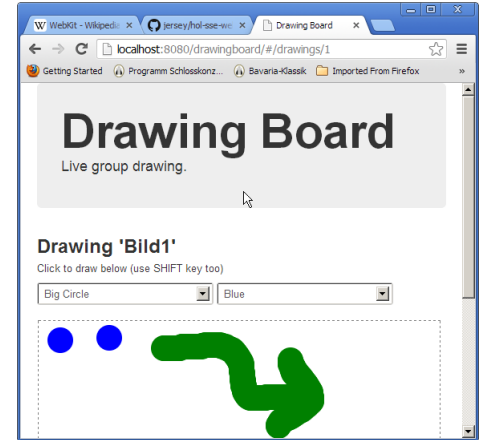
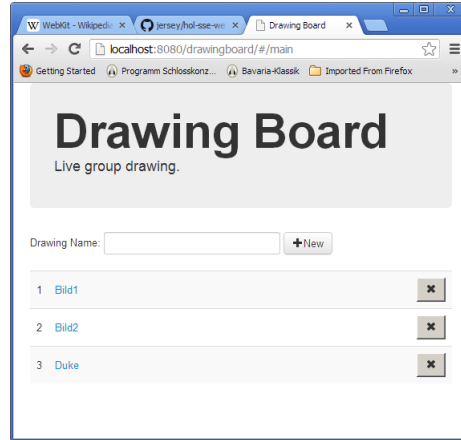
## Related Oracle Projects

- Avatar
  - Ent-to-end TSA framework based on HTML5 and JavaScript (also server-side)
- Easel
  - JavaScript tooling support
- Nashorn
  - JavaScript implementation on the JVM
- EclipseLink/TopLink data services
  - Enable REST access to RDBMS and NoSQL data using JSON or XML
  - Live Data Notifications over WebSockets or Server Sent Events
- PaaS for FMW

# Drawing Board Demo

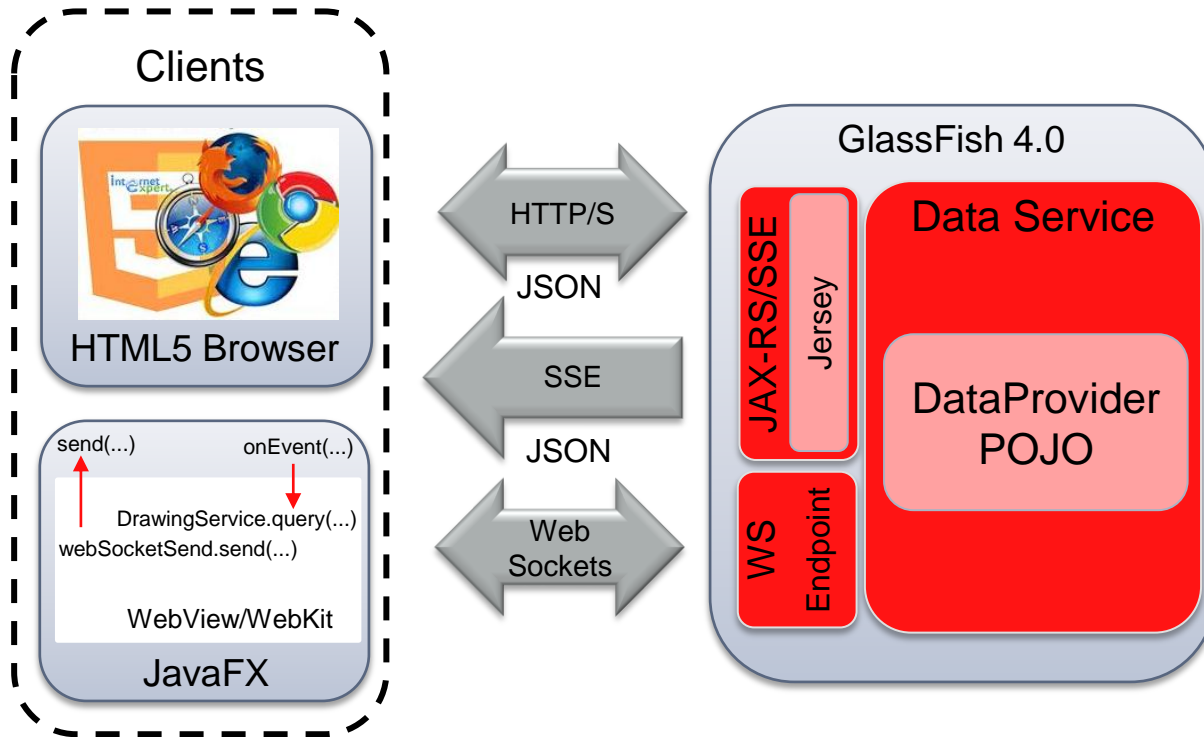
<http://github.com/jersey/hol-sse-websocket>

- Collaborative drawing
- Two-page application
  - List of drawings
  - Drawing
- Demonstrating
  - Server-side: JAX-RS, JSON, WebSocket, SSE Java API
  - Client-side: JAX-RS, WebSocket, SSE Java and JavaScript API
  - JavaFX **hybrid** Java/HTML5 application



# Drawing Board Demo

## TSA - Architecture



# Drawing Board Demo

## Technology usage

- JAX-RS: CRUD for drawings
- SSE: distributing the list of drawings to all connected clients
- WebSocket: distributing the updates of a drawing to all connected clients
- JSON: implementing of encoder/decoder of the WebSocket server endpoint
- Java – JavaScript bridge(WebEngine): modifying the AngularJS client by replacing the WebSocket/SSE JavaScript client communication with a Java implementation in the JavaFX client

# Links

## ■ HTML5

- <http://www.w3.org/TR/html5/>
- <http://www.whatwg.org/specs/web-apps/current-work/multipage/>
- <http://en.wikipedia.org/wiki/HTML5>

## ■ Thin Server Architecture

- <http://www.thinserverarchitecture.com>
- <http://review.us.oracle.com/review2/Review.html#reviewId=130188>

## ■ JAX-RS

- <http://jax-rs-spec.java.net>
- <http://jersey.java.net>

## ■ JSON

- <http://json-processing-spec.java.net>
- <http://jsonp.java.net>

## ■ WebSocket

- <http://websocket-spec.java.net>
- <http://tyrus.java.net>

## ■ Server-Sent Events

- <http://jersey.java.net>

## ■ JavaFX

- <http://www.oracle.com/technetwork/java/javafx/overview/index.html>
- <http://docs.oracle.com/javafx/2/api/javafx/scene/web/WebEngine.html>

**Hardware and Software**

**ORACLE®**

**Engineered to Work Together**

**ORACLE®**