

# AVATAR: JAVASCRIPT-BINDING FÜR JAVA-EE KOMPONENTEN

NIKO KÖBLER  
HEIKO SPINDLER

QUALITECTS GROUP

# Agenda



- Das Web als Plattform
- Avatar Grundlagen
- Avatar Features
- Avatar Demo

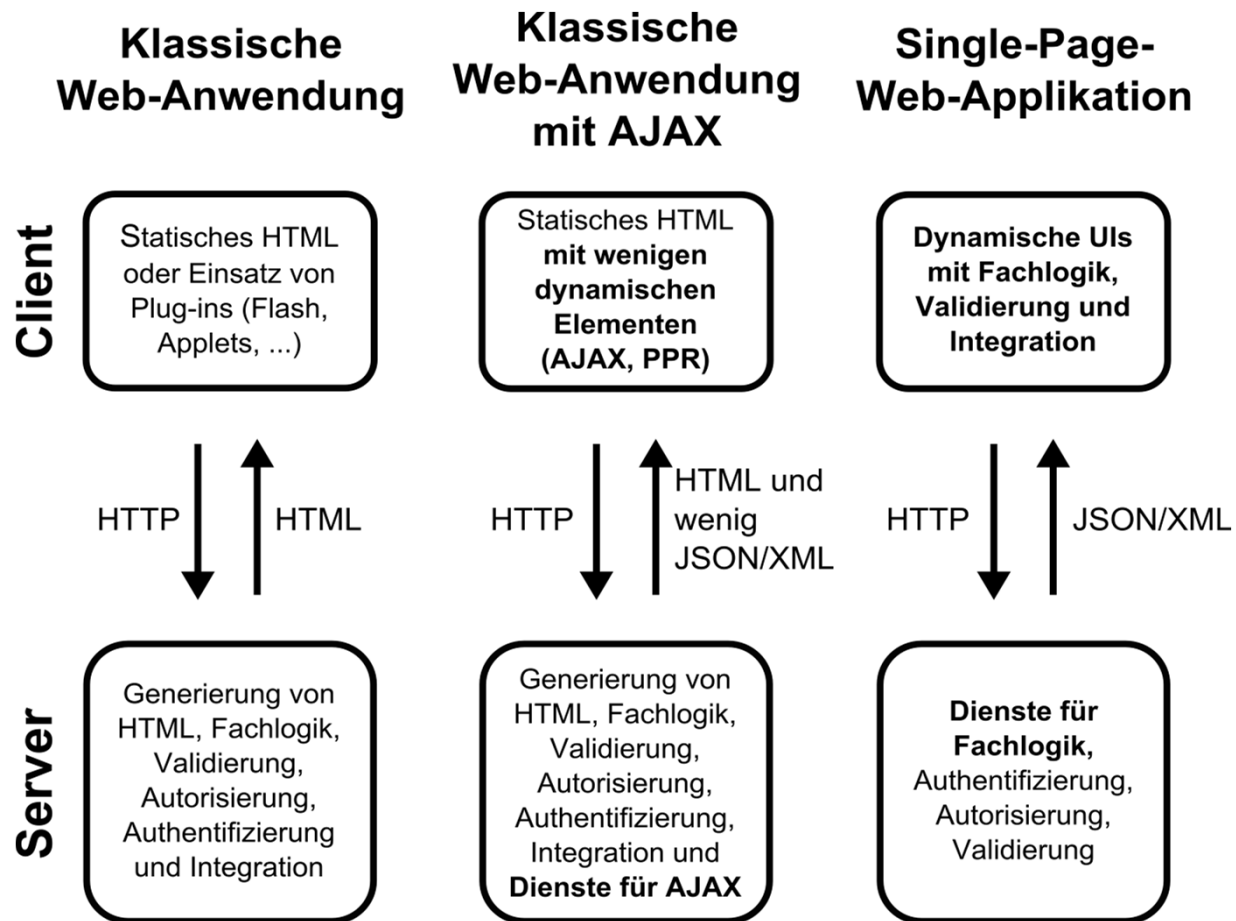
# „JavaScript is the new black“

## Das Web als Plattform

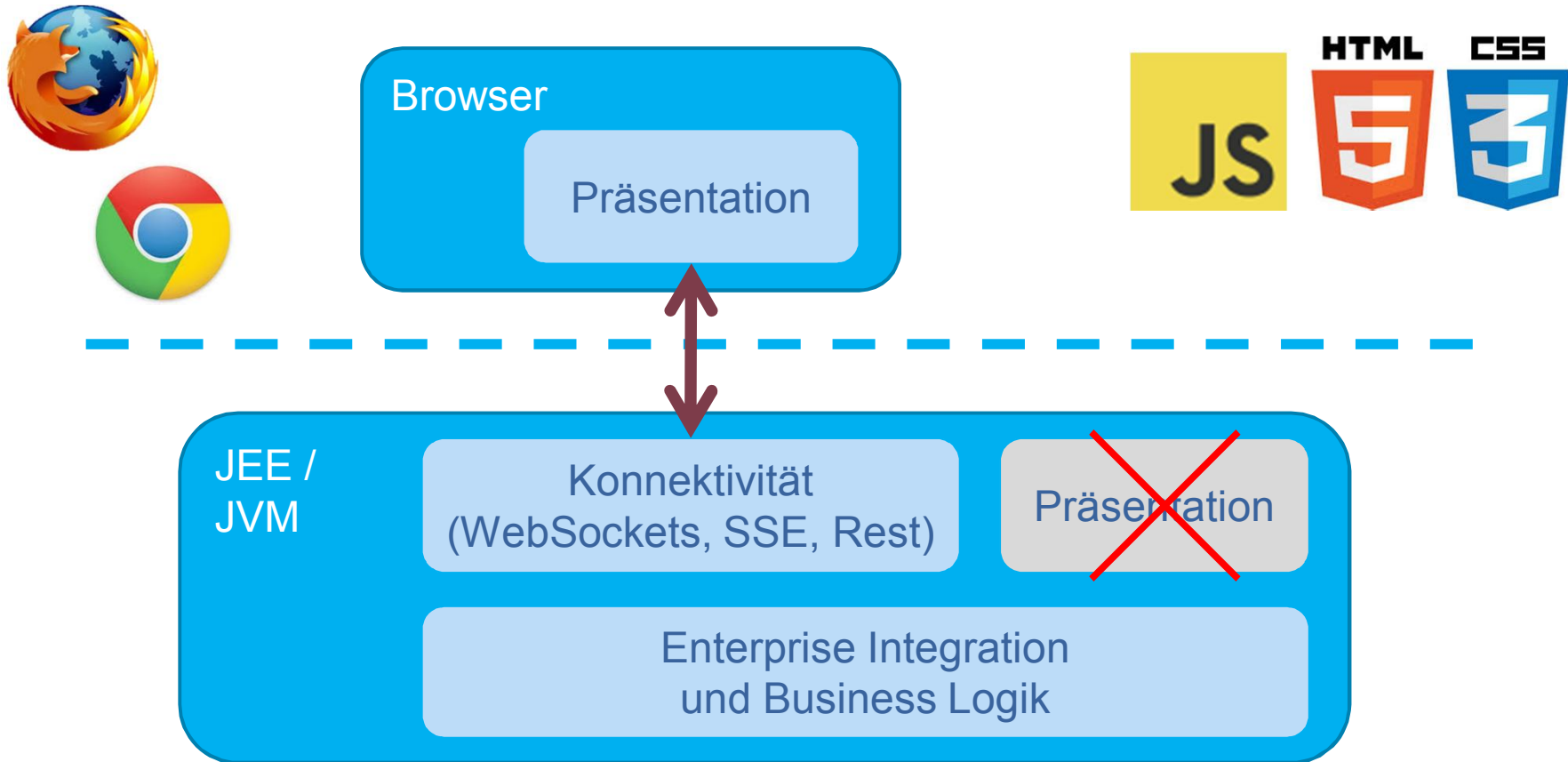
- Das Web wird zur Plattform
  - ▣ Einheitliche Standards
- JavaScript wird zur allgemeinen Programmiersprache für eine Vielzahl von Geräten
  - ▣ JavaScript ist schnell genug
  - ▣ Unterschiede der Browser reduzieren sich (Frameworks, Standards)
  - ▣ Best-Practices und Know-How sind vorhanden
  - ▣ QS-Maßnahmen und Werkzeuge entstehen



# Web-Anwendung im Wandel



# Modern Web Application Architecture



# Avatar Basics



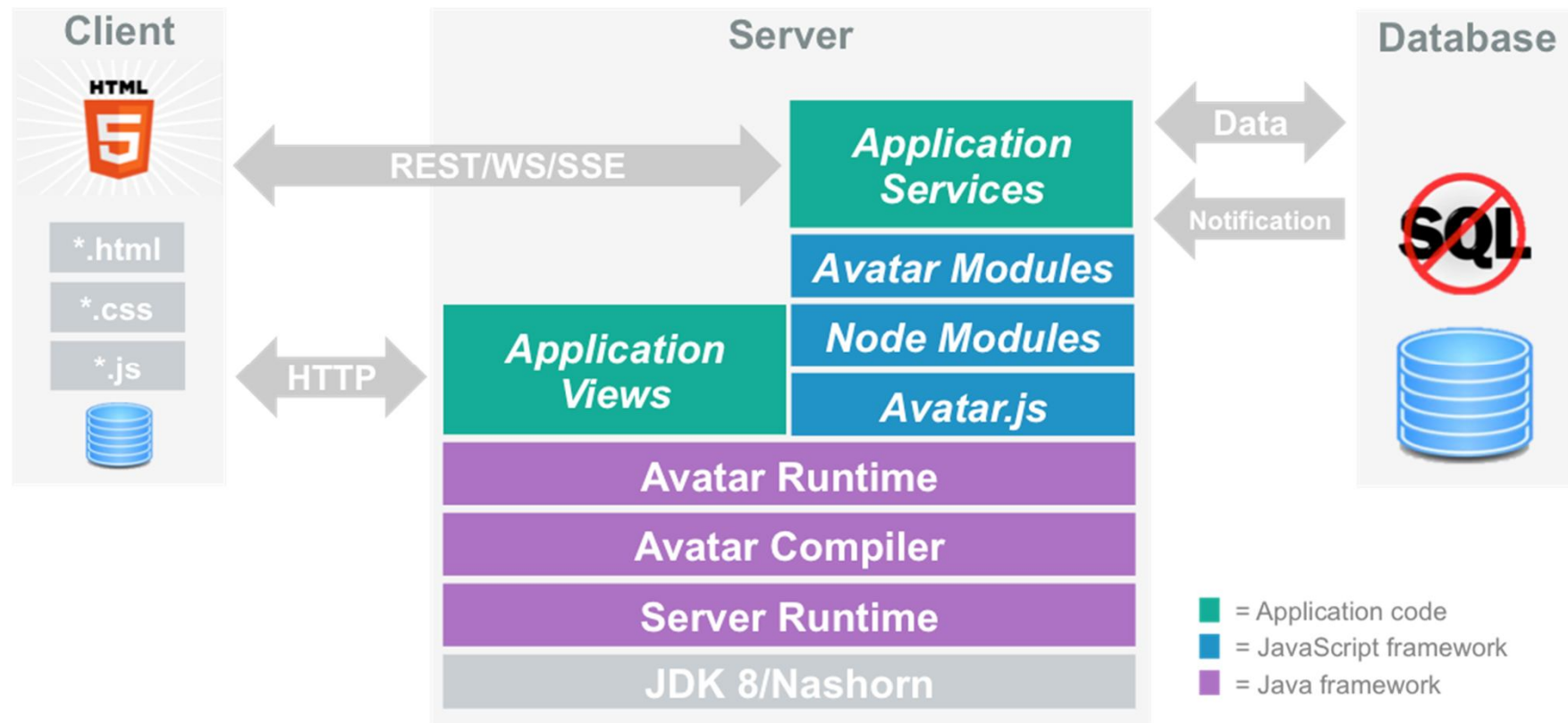
- Plattform für server-seitiges JavaScript auf der JVM
- Basiert auf Nashorn (JDK 8)
- Open-source Projekt auf [java.net](http://java.net) gestartet von Oracle
- 95% Node.js-API kompatibel
  - Code in JavaScript
  - Single threaded event loop
  - Viele npm-Packages funktionieren (Express, Async, etc)
  - Benötigt die selben Voraussetzungen wie Node.js
  - Keine Chrome V8 nativen APIs

# Avatar Features



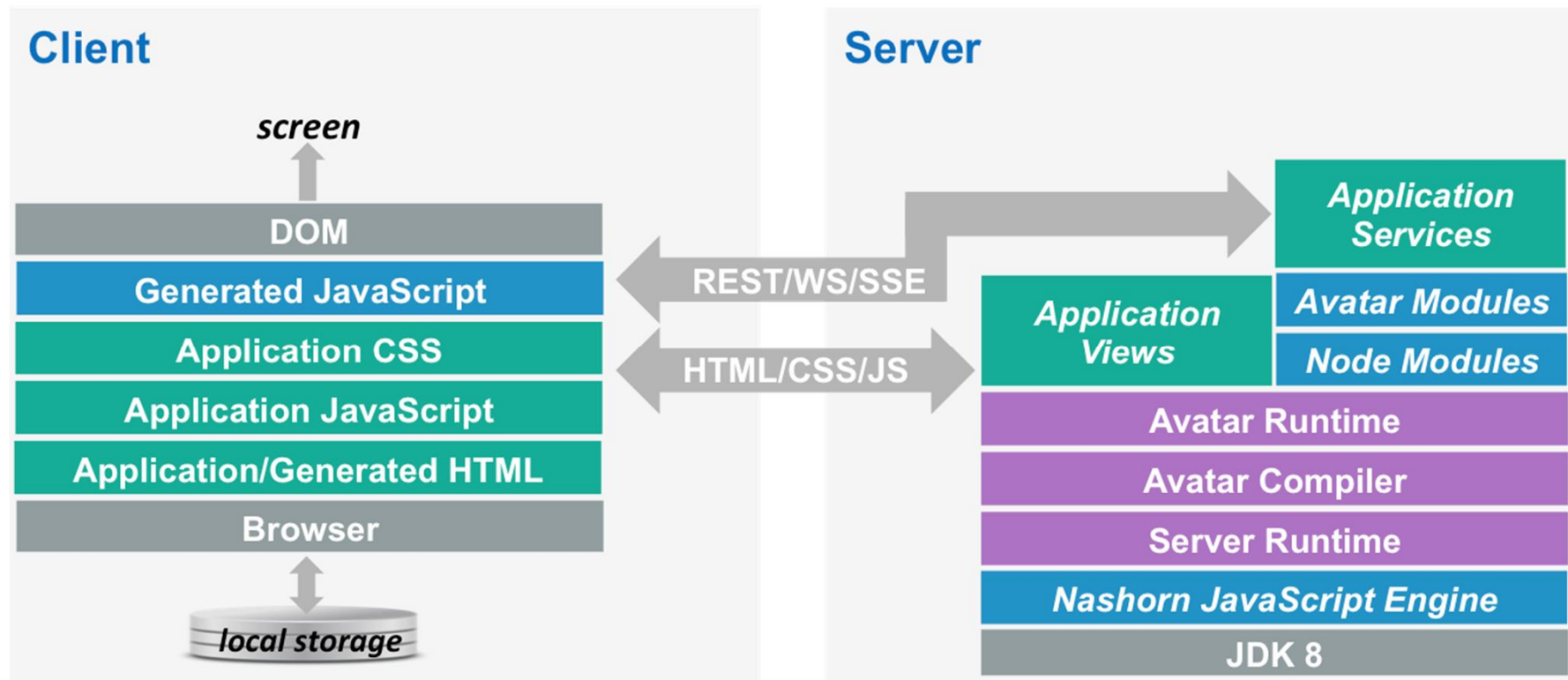
- Zugriff auf Java Frameworks, Bibliotheken und Werkzeuge
  - Durch JavaScript Wrapper
  - Durch asynchrone Adapter
- Eingebettet in den Security Manager
- Besteht aus einer Server- und Client-Komponente

# Avatar Architektur





# Zusammenspiel Client- und Server Komponenten



# Definition des Modells

```
<script data-model="local"
data-instance="add">
  var AddMathModel = function() {
    this.left = 0; this.right = 0;
    this.reset = function() {
      this.left = this.right = 0;
    };
  };
</script>
```

Operand 1:

Operand 2:

The result is  $4 + 6 = 10$

Reset

Next

# 2-way Binding zwischen Modell und UI-Elementen

```
<input id="lfa" type="text" data-value="#{add.left}"/>
```

```
<input id="rta" type="text" data-value="#{add.right}"/>
```

```
<span id="output1">The result is #{add.left} +  
#{add.right} = #{add.left + add.right}</span>
```

```
<button onclick="#{add.reset()}"  
id="reset">Reset</button>
```

# Avatar Provider



```
var avatar = require("org/glassfish/avatar");

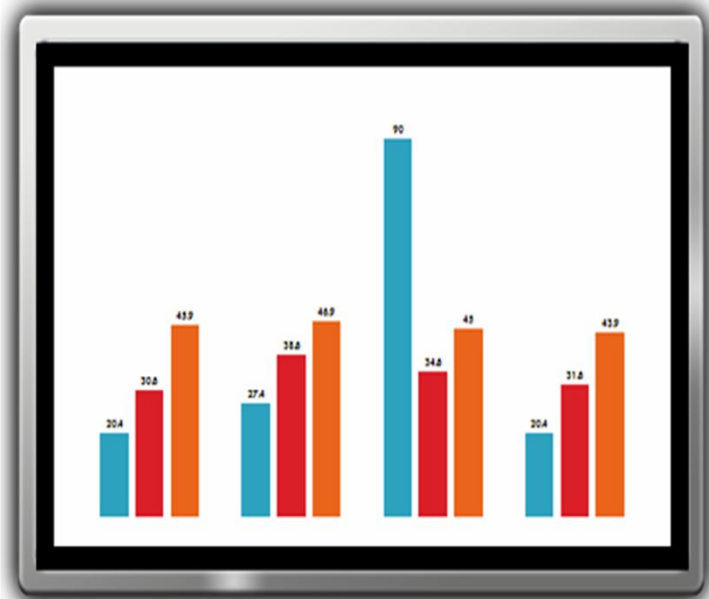
var itemsFileProvider =
  new avatar.FileDataProvider(
    { filename: "rest-sample.txt", key: "key" } );

var itemsJpaProvider =
  new avatar.JPADataProvider(
    { persistenceUnit:"rest", createTables:"true",
      entityType:"Item" }
  );
```

# Avatar RestService

```
avatar.registerRestService({url:"data/items/{item}"},
function() {
  this.onGet = function(request, response) {
    itemsFileProvider.get( this.item,
      function(error, itemValue){
        if (!itemValue) { itemValue = {};}
      }
    response.send(itemValue).done();
  });
};
}
```

# Demo Time



# Timeline (unverbindlich)



Zeitraum	Meilensteine
JavaOne 2013	Projekt Avatar Launch GlassFish Runtime
Q2/2014	WebLogic Runtime (WLS 12.1.3)
Mitte 2015	WebLogic 12.1.4 Avatar Commercial Support

# Fazit



- Kommerzieller Support von Oracle (in Zukunft)
- Leichte Integration in die JEE-Welt und den Unternehmenskontext
  - JMS, JPA, Rest, Java-APIs
- Harmonisches Zusammenspiel zwischen Client und Server
- Einsatz der Client-Komponente ist kein MUSS
  - Einsatz der Server-Komponente aber auch nicht



Q & A

AVATAR

# Links



- Project Avatar: <https://avatar.java.net/>
- Qualitects Group: [www.qualitects-group.com/](http://www.qualitects-group.com/)