

ORACLE®



JavaOne™

ORACLE®

JSF 2.3: The Community Takes Over

CON7981

Ed Burns

Servlet Specification Co-leads

Java EE Platform Group

Kito Mann

JSF Expert Group Member and Independent Consultant

September 2016

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 Bios

- Ed Burns



- Servlet and JSF co-spec lead since 1.0
- Frequent international conference speaker
- Author of four Enterprise Java related books from McGraw-Hill, including JSF 2.0 the Complete Reference

- Kito Mann



- JSF EG member since 1.2
- Founder of JSFCentral.com
- Host of Enterprise Java Newscast
- Author of JSF in Action
- Independent trainer and consultant

Our Plan for Your Time Investment

- 1 ➤ Big Picture
- 2 ➤ JSF Historical Perspective
- 3 ➤ Big Ticket New Features
- 4 ➤ Java EE Platform Integration
- 5 ➤ Summary

PrimeFaces Wins Duke's Choice Award 2016

<https://www.oracle.com/javaone/dukes-choice-award.html>

- Category: Enterprise Java



Java EE 8 (Revised Proposal, 2016)

Batch	Dependency Injection	JACC	JAXR	JSTL	Management
Bean Validation	Deployment	JASPICTM	JMS	JTA	Servlet
CDI	EJB	JAX-RPC	JSF	Java Persistence	Web Services
Common Annotations	EL	JAX-RS	JSON-B	JavaMail	Web Services Metadata
Concurrency EE	Interceptors	JAX-WS	JSON-P	Managed Beans	WebSocket
Connector	JSP Debugging	JAXB	JSP	Security	
		Configuration	Health Check		

Java EE 8 (Revised Proposal, 2016)



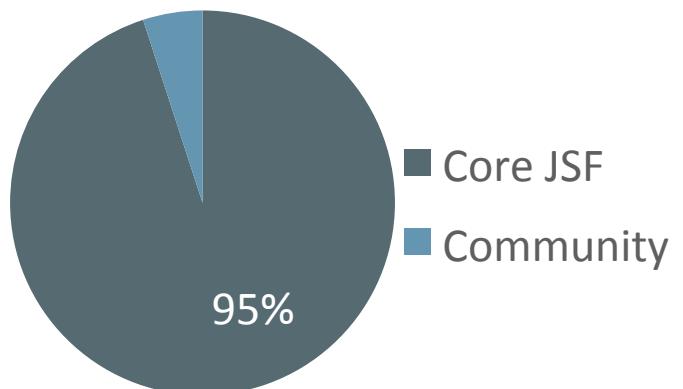
JSF and the Cloud

- Java EE 6 is already in Oracle's public cloud
- It is often a part of an early stage of “breaking up the monolith” efforts
 - Start by replacing direct in-process invocation of CDI beans with REST calls to remote services
 - Check out JoinFaces for a way to quickly run JSF in Spring Boot



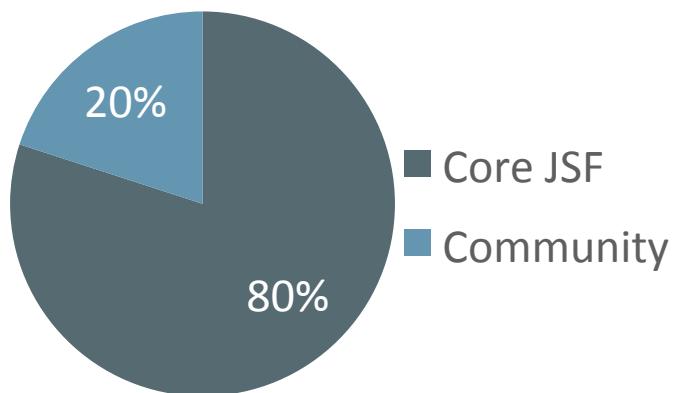
JSF Historical Perspective

2004



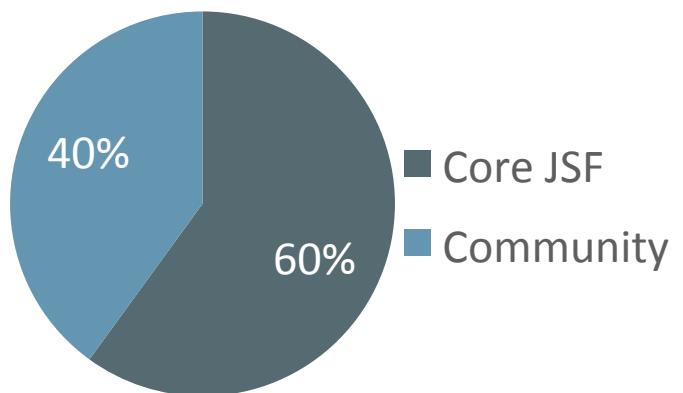
JSF Historical Perspective

2008



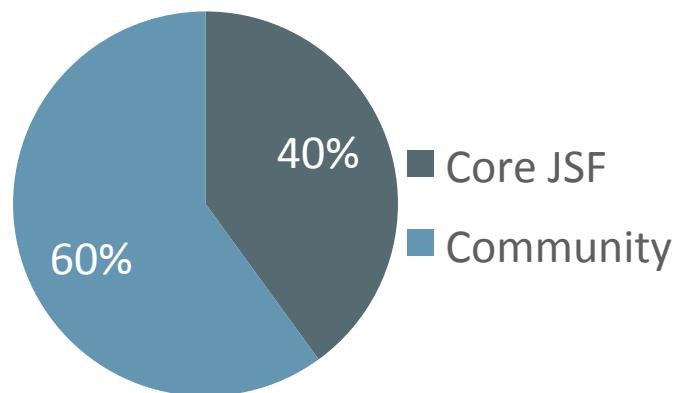
JSF Historical Perspective

2009



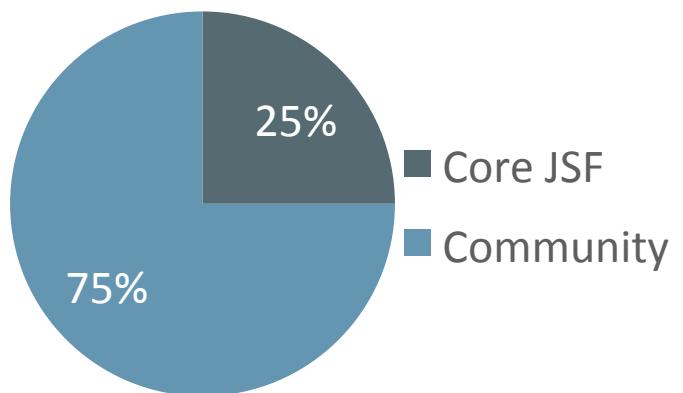
JSF Historical Perspective

2010



JSF Historical Perspective

2013

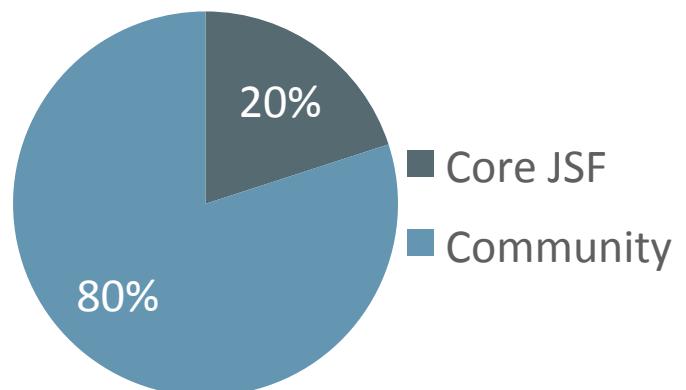


JSF Historical Perspective

<http://bit.ly/JsfFtw2015>

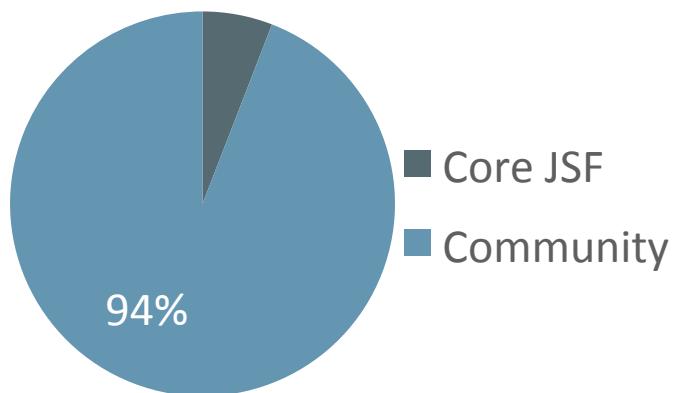
Poll: JSF and Spring MVC tie

2014



JSF Historical Perspective

2016



JSF Historical Perspective

- JSR 127
 - JSF 1.0 11 March 2004
 - JSF 1.1 27 May 2004
- JSR 252
 - JSF 1.2 11 May 2006
 - JSF 1.2 Maintenance Release 1 19 December 2006
 - JSF 1.2 Maintenance Release 2 13 June 2008
 - JSF 1.2 Maintenance Release 3 25 August 2008
- JSR 314
 - JSF 2.0 1 July 2009
 - JSF 2.1 16 July 2010
 - JSF 2.1 Maintenance Release 2 22 November 2010
- JSR 344
 - JSF 2.2 Started 14 April 2011
 - Final June 2013
 - <http://bit.ly/JavaEE7WrapsUp>
- JSR 372
 - JSF 2.3
 - Started 22 September 2014
 - Early Draft Review October 2015
 - Proposed Final Draft July 2017



JSF 2.3 Big Ticket New Features

<http://arjan-tijms.omnifaces.org/p/jsf-23.html>

- Better CDI Integration
 - Way more things are injectable
 - Finally marking legacy managed beans as deprecated
- Websocket Integration
- Ajax Method Invocation
- Class Level Bean Validation
- Java Time support
- UIData and UIRepeat improvements

JSF 2.3 Big Ticket New Features

<http://arjan-tijms.omnifaces.org/p/jsf-23.html>

Artefact	EL name	Qualifier	Type
Application	#{}{application}	-	java.lang.Object (javax.servlet.ServletContext)
ApplicationMap	#{}{applicationScope}	@ApplicationMap	java.util.Map<String, Object>
CompositeComponent	#{}{cc}	(Not injectable)	javax.faces.component.UIComponent
Component	#{}{component}	(Not injectable)	javax.faces.component.UIComponent
RequestCookieMap	#{}{cookie}	@RequestCookieMap	java.util.Map<String, Object>
FacesContext	#{}{facesContext}	-	javax.faces.context.FacesContext
Flash	#{}{flash}	-	javax.faces.context.Flash
FlowMap	#{}{flowScope}	@FlowMap	java.util.Map<Object, Object>
HeaderMap	#{}{header}	@HeaderMap	java.util.Map<String, String>
HeaderValuesMap	#{}{headerValues}	@HeaderValuesMap	java.util.Map<String, String[]>
InitParameterMap	#{}{initParam}	@InitParameterMap	java.util.Map<String, String>
RequestParameterMap	#{}{param}	@RequestParameterMap	java.util.Map<String, String>

JSF 2.3 Big Ticket New Features

<http://arjan-tijms.omnifaces.org/p/jsf-23.html>

Artefact	EL name	Qualifier	Type
Request	#{}{request}	(Not injectable)	java.lang.Object (javax.servlet.http.HttpServletRequest)
RequestMap	#{}{requestScope}	@RequestMap	java.util.Map<String, Object>
ResourceHandler	#{}{"resource"}	-	javax.faces.application.ResourceHandler
Session	#{}{session}	(Not injectable)	java.lang.Object (javax.servlet.http.HttpSession)
SessionMap	#{}{sessionScope}	@SessionMap	java.util.Map<String, Object>
View	#{}{view}	-	javax.faces.component.UIViewRoot
ViewMap	#{}{viewScope}	@ViewMap	java.util.Map<String, Object>
ExternalContext	#{}{externalContext} (new)	-	javax.faces.context.ExternalContext

JSF 2.3 Big Ticket New Features

WebSocket Integration

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml" xmlns:f="http://xmlns.jcp.org/jsf/core"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:body>
    <h:form id="form">
      <h:commandButton id="button" value="push" action="#{pushBean.send}">
        <f:ajax />
      </h:commandButton>
    </h:form>

    <f:websocket channel="push" onmessage="function(message
          {document.getElementById('form:button').value=message;}" />
    <div id="user" />
    <f:websocket channel="user" user="user" onmessage="function(message)
          {document.getElementById('user').innerHTML=message;}" />
    <div id="view" />
    <f:websocket channel="view" scope="view" onmessage="function(message)
{document.getElementById('view').innerHTML=message;}" /
  </h:body></html>
```

JSF 2.3 Big Ticket New Features

WebSocket Integration

```
@Model
public class PushBean {
    @Inject @Push
    private PushContext push;
    @Inject @Push
    private PushContext user;

    @Inject @Push
    private PushContext view;

    public void send(){
        push.send("pushed!");
        user.send("pushed!", "user");
        view.send("pushed!");
    }
}
```

```
import javax.websocket.Endpoint;
import javax.websocket.EndpointConfig;
import javax.websocket.Session;
public class FakeEndpoint extends Endpoint {
    @Override
    public void onOpen(Session session, EndpointConfig
config) {// https://java.net/jira/browse/
WEBSOCKET_SPEC-240
    }
}
```

JSF 2.3 Big Ticket New Features

Ajax Method Invocation

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml" xmlns:f="http://xmlns.jcp.org/jsf/core"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
    <h:body>

        <h:form>
            <h:commandScript name="foo" autorun="true" action="#{spec613.setResult('foo')}"
render=":result" />
            <h:commandScript name="bar" action="#{spec613.setResult('bar')}" render=":result" />
        </h:form>
        <h:outputText id="result" value="#{spec613.result}" />
    </h:body>
</html>
```

JSF 2.3 Big Ticket New Features

GetEvalScripts

@Named

@RequestScoped

```
public class SomeBean {
```

```
    @Inject
```

```
    private FacesContext context;
```

```
    public void eval() {
```

```
        context.getPartialViewContext()
```

```
            .getEvalScripts()
```

```
            .add("alert('After response')");
```

```
}
```



JSF 2.3 Big Ticket New Features

Class Level Bean Validation

```
<h:form prependId="false" id="form">
  <h:panelGrid columns="2">
    <h:outputText value="Password" />
    <h:inputSecret id="password1" value="#{backingBean.password1}" />
    <h:outputText value="Password again" />
    <h:inputSecret id="password2" value="#{backingBean.password2}" />
    <h:commandButton id="submit" value="Submit"/>
  </h:panelGrid>
  <h:messages id="err"/>
  <f:validateWholeBean value="#{backingBean}" copierType="SerializationCopier"
validationGroups="com.sun.faces.test.javaee8.validateWholeBean.PasswordValidationGroup" />
</h:form>
```

JSF 2.3 Big Ticket New Features

Class Level Bean Validation

```
@Named  
@RequestScoped  
@Password(groups = PasswordValidationGroup.class)  
public class BackingBean implements Serializable {  
    private String password1;  
    private String password2;  
  
    // ... }  
  
@Constraint(validatedBy = {PasswordValidator.class})  
@Documented  
@Target(TYPE)  
@Retention(RUNTIME)  
public @interface Password {  
    String message() default "Password fields must match";  
    Class<?>[] groups() default {};  
    Class<? extends Payload>[] payload() default {};  
}
```



JSF 2.3 Big Ticket New Features

Class Level Bean Validation

```
public class PasswordValidator implements ConstraintValidator<Password,  
BackingBean> {  
  
    @Override  
    public void initialize(Password constraintAnnotation) {}  
    @Override  
    public boolean isValid(BackingBean value, ConstraintValidatorContext context)  
    {  
        boolean result;  
        result = value.getPassword1().equals(value.getPassword2());  
        return result;  
    }  
}
```

JSF 2.3 Big Ticket New Features

Java DateTime Support

```
-----> <h:inputText id="localDateTimeValue1" value="#{backingBean.localDateTime1}">
    <f:convertDateTime type="localDateTime"/>
</h:inputText>

<h:outputText id="localDateTimeValue2" value="#{backingBean.localDateTime2}">
    <f:convertDateTime type="localDateTime" timeStyle="short"/>
</h:outputText>

<h:outputText value="localTime1" />

<h:inputText id="localTime1" value="#{backingBean.localTime1}">
    <f:convertDateTime type="localTime" />
</h:inputText>

<h:outputText id="localTimeValue1" value="#{backingBean.localTime1}">
    <f:convertDateTime type="localTime" />
</h:outputText>

<h:outputText value="localTime2" />
```

JSF 2.3 Big Ticket New Features

UIData and UIRepeat Improvements

- Supports Iterable

A woman with long dark hair and glasses, wearing a denim jacket over a black top, is sitting at a desk and working on a computer. She is looking down at the screen. The background is a blurred office environment.

Multi-Tenancy in JSF 2.2

Resource Library Contracts

Faces Flows

Modularity and Multi-tenant capability

- Allow composing a JSF app as a collection of modules
 - Faces Flows modularize behavior
 - Resource Library Contracts modularize appearance
- Well defined contract for each



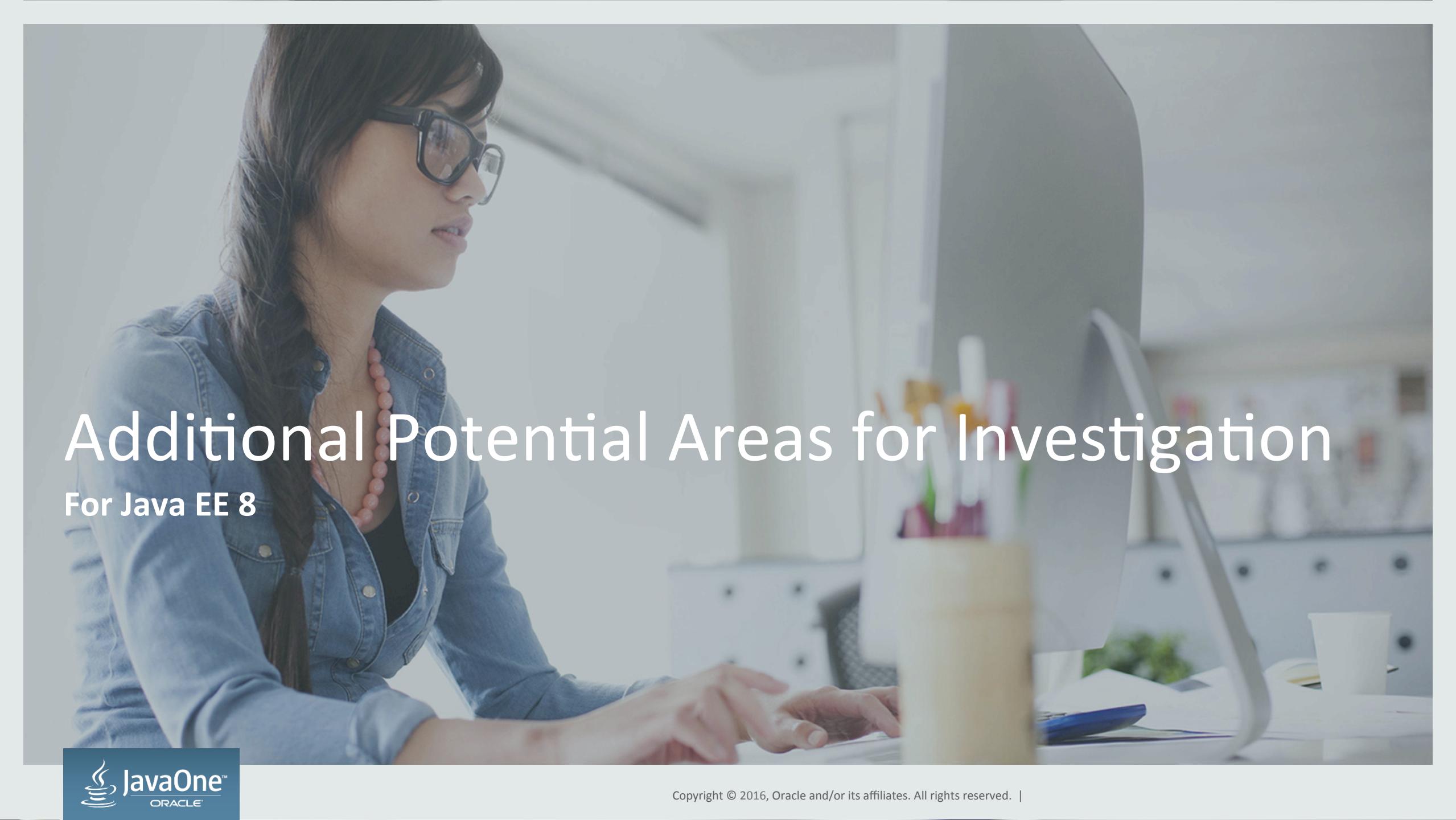
Resource Library Contracts and Faces Flows

What's going on here?

- Two new concepts in JSF 2.2

- Resource Library Contract
 - Faces Flows

 Builds on facelets concepts
 Builds on navigation concepts

A woman with long dark hair and glasses, wearing a denim jacket over a black top, is sitting at a desk and working on a computer. She is looking down at the screen. The background is a blurred office environment.

Additional Potential Areas for Investigation

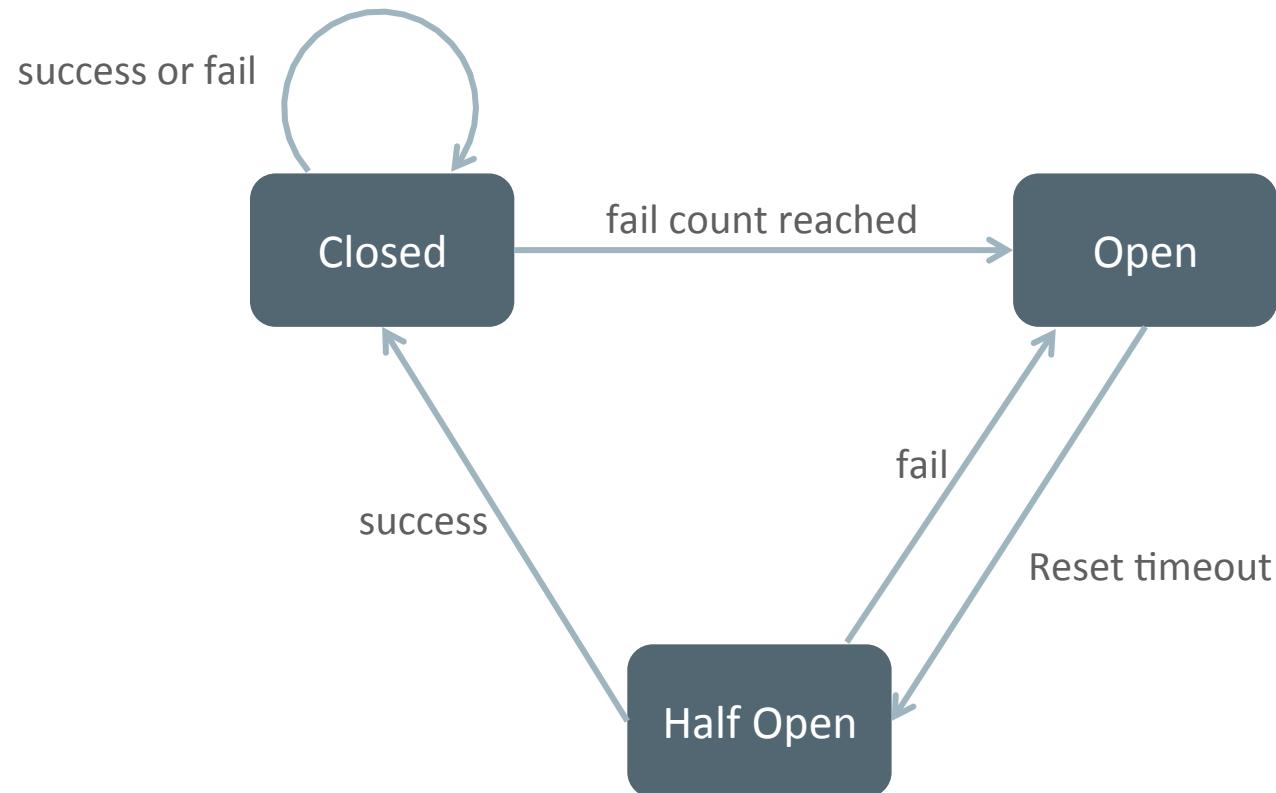
For Java EE 8

Additional Potential Java EE 8 Focus Areas

- Config
 - Externalized config
 - faces-config.xml
 - web.xml
- HealthCheck
 - Ensure server is up and can contact downstream services
- Security
 - Doing OAuth and Time-Based One-Time Password are currently too hard. It would be nice to make this easier

Additional Potential Java EE 8 Focus Areas

Circuit Breaker



Summary and Current Status

- JSR-372 formed on 22 September 2014
- Early Draft Review JavaOne 2015
- Working toward Public Review

Other Sessions of Interest

- Java EE Applications on Oracle Java Cloud Service [CON3069]
Thursday 22 September, 1:00, Parc 55 Powell I/II
- Breaking Apart the Monolith with Microservices [CON3127]
Thursday 22 September, 2:30, Parc 55 Cyril Magnin I
- Configure Once, Run Everywhere with Apache Tamaya [CON1141]
Thursday 22 September, 16:00 Parc 55 Mission

Next Steps

Give us your feedback

- Take the survey
 - <http://glassfish.org/survey>
- Send technical comments to
 - users@javaserverfaces-spec-public.java.net
- Join the JCP – come to Hackergarden in Java Hub
 - https://jcp.org/en/participation/membership_drive
- Join or track the JSRs as they progress
 - <https://java.net/projects/javaee-spec/pages/Specifications>
- Adopt-a-JSR
 - <https://community.oracle.com/community/java/jcp/adopt-a-jsr>

Where to Learn More at JavaOne

Session Number	Session Title	Day / Time
CON1558	What's New in the Java API for JSON Binding	Monday 5:30 p.m.
BOF7984	Java EE for the Cloud	Monday 7:00 p.m.
CON4022	CDI 2.0 Is Coming	Tuesday 11:00 a.m.
CON7983	JAX-RS 2.1 for Java EE 8	Tuesday 12:30 p.m.
CON8292	Portable Cloud Applications with Java EE	Tuesday 2:30 p.m.
CON7980	Servlet 4.0: Status Update and HTTP/2	Tuesday 4:00 p.m.
CON7978	Security for Java EE 8 and the Cloud	Tuesday 5:30 p.m.
CON7979	Configuration for Java EE 8 and the Cloud	Wednesday 11:30 a.m.
CON7977	Java EE Next – HTTP/2 and REST	Wednesday 1:00 p.m.
CON6077	The Illusion of Statelessness	Wednesday 4:30 p.m.
CON 7981	JSF 2.3	Thursday 11:30 a.m.

Safe Harbor Statement

The preceding 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.

Integrated Cloud Applications & Platform Services



JavaOne™

ORACLE®