

MVC 1.0 - by Example

Ivar Grimstad

*Principal Consultant, Cybercom Sweden
JCP Expert Group Member (JSRs 368, 371, 375)*



@ivar_grimstad



<https://github.com/ivargrimstad>



<https://www.linkedin.com/in/ivargrimstad>



<http://lanyrd.com/profile/ivargrimstad/>

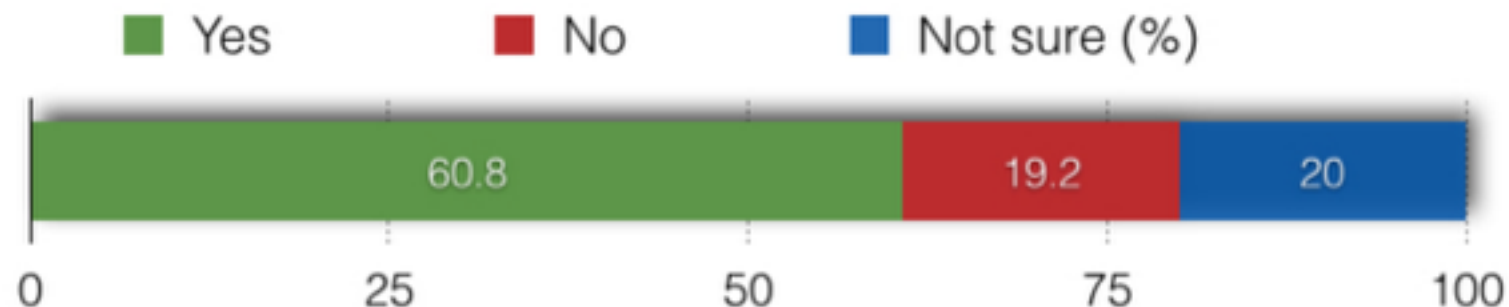


MVC

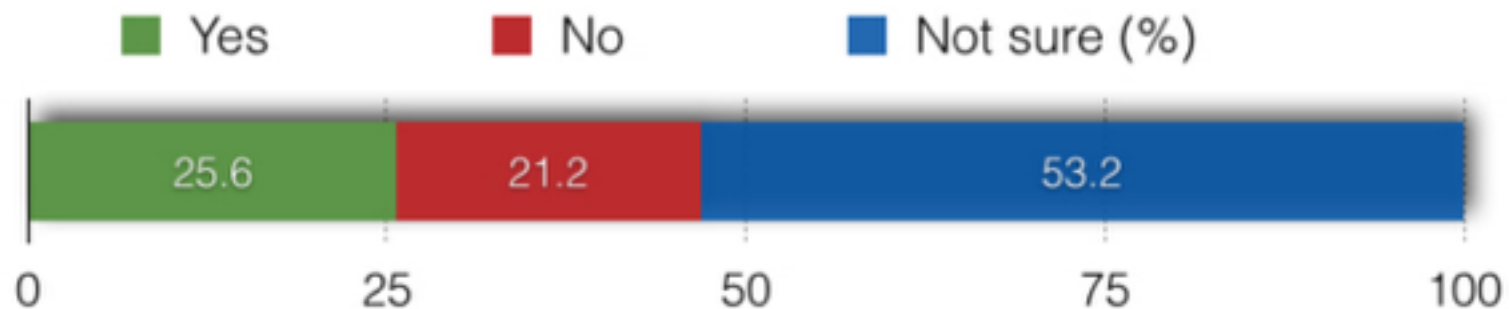
Why?

MVC Support

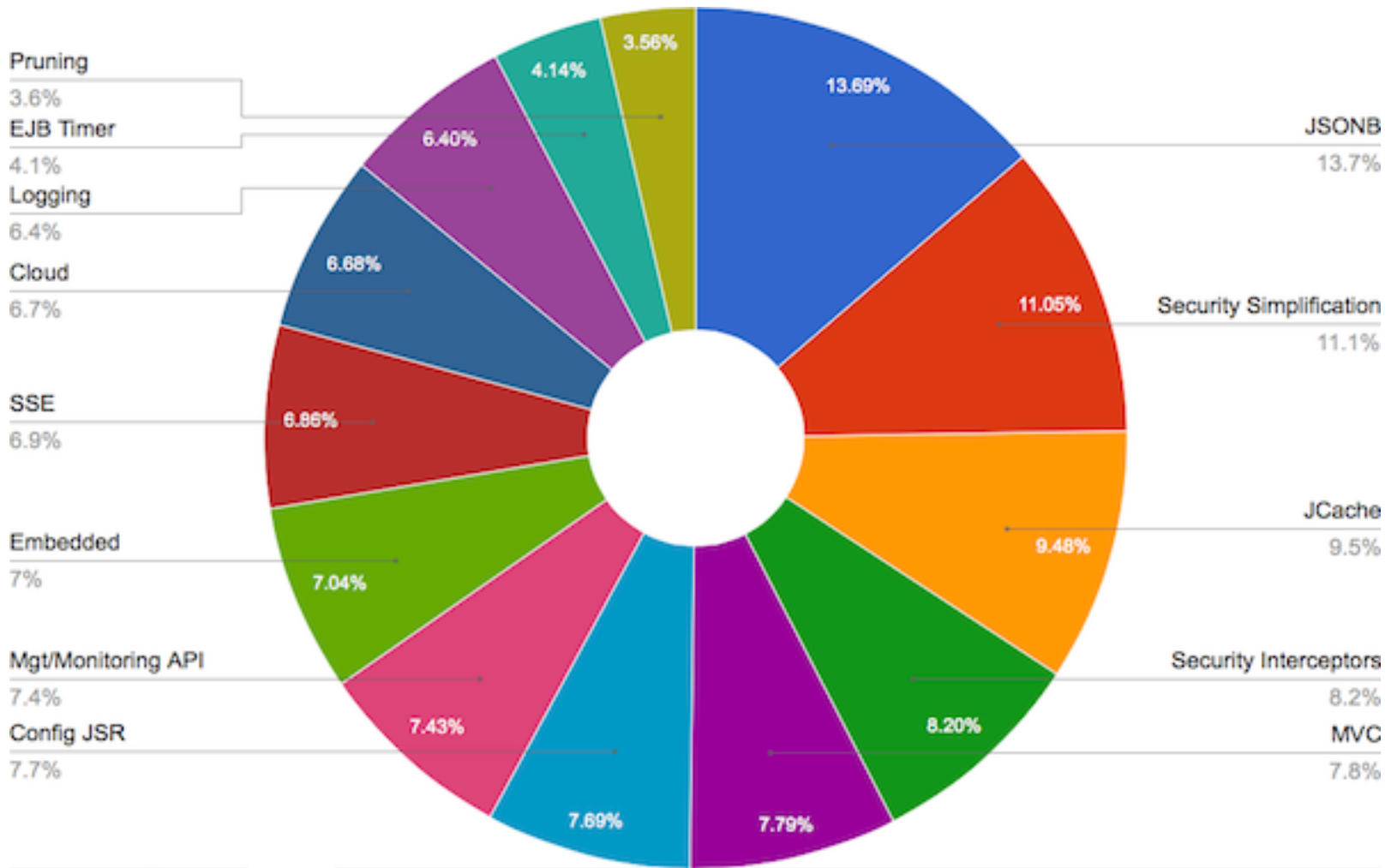
Should Java EE provide support for MVC, alongside JSF?



Is there any one de-facto standard technology in this space to which we should look for inspiration?



https://java.net/downloads/javaee-spec/JavaEE8_Community_Survey_Results.pdf



<http://glassfish.org/survey>

JSR 371 - MVC 1.0

JSR 371 Expert Group

Specification Leads



Santiago Pericas-Geertsen

Oracle

Manfred Riem

Oracle

Expert Group

Mathieu Ancelin

Caelum
: Rodrigo Turini

Frank Caputo

Ivar Grimstad

IBM
: Paul Nicolucci

innoQ Deutschland GmbH
: Stefan Tilkov

Christian Kaltepoth

Liferay, Inc
: Neil Griffin

Mann, Kito D.

Oracle
: Santiago Pericas-Geertsen

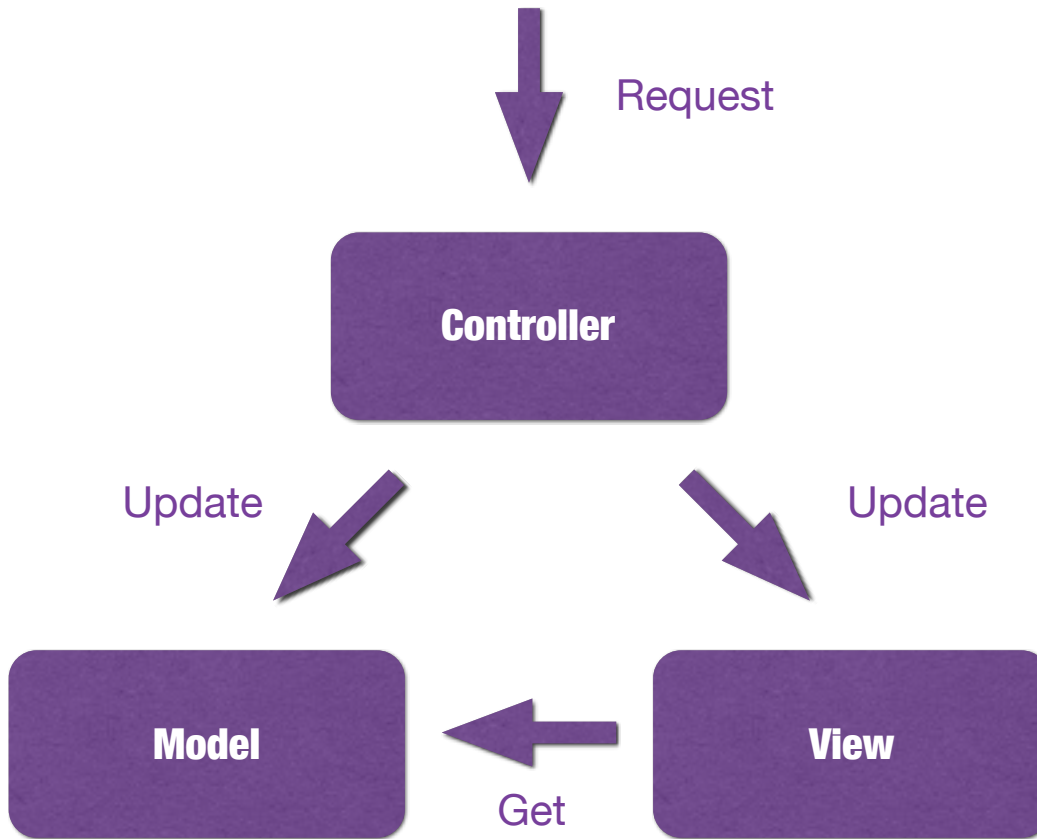
Oracle
: Manfred Riem

RedHat
: Joshua Wilson

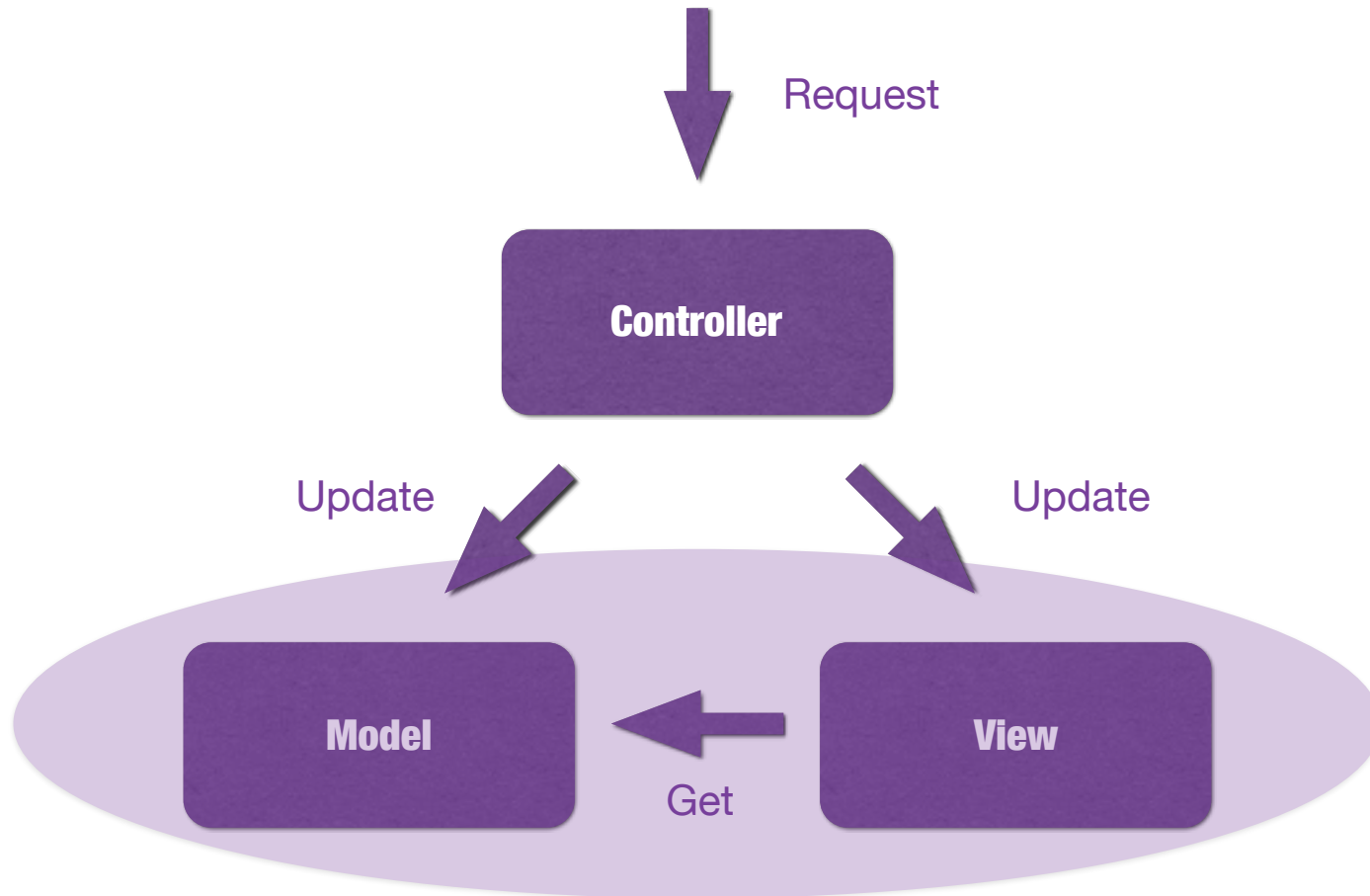
Guilherme de Azevedo Silveira

TmaxSoft, Inc.
: Woong-ki Lee

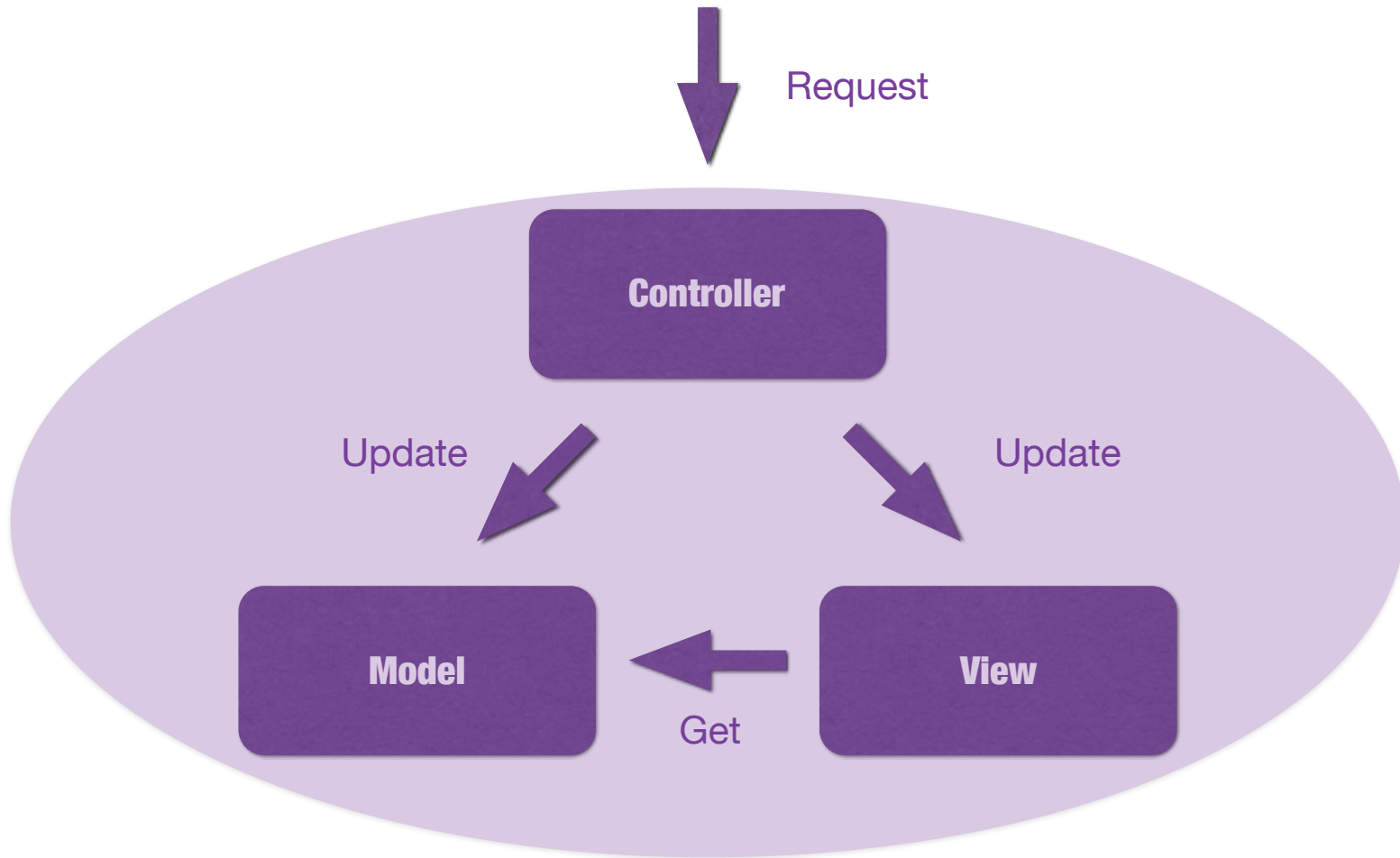
Action-based MVC



Component-based MVC



Action-based MVC



Comparable Frameworks

Spring MVC

around since 2005

“Real World” since 2008 (Spring 2.5 onward)

today’s leader in Action-Land

Struts 2

around since 2004 (WebWork 2)

“Real World” since 2007 (Struts 2.0.9 onward)

still widely adopted

Existing Java EE Technologies

Key Decisions

Key Decision

Build MVC 1.0 on top of JAX-RS

Controllers

Controller

```
@Controller
@Path("hello")
public class HelloController {

}
```

Views

View

```
@Controller
@Path("hello")
public class HelloController {

}
```

View

```
@Controller  
@Path("hello")  
public class HelloController {
```

```
    @GET  
    public String hello() {  
        return "hello.jsp";  
    }  
}
```


View

```
@Controller
@Path("hello")
public class HelloController {

    @GET
    public Viewable hello() {
        return new Viewable("hello.jsp");
    }
}
```

View

```
@Controller
@Path("hello")
public class HelloController {

    @GET
    public Response hello() {
        return Response.status(OK).entity("hello.jsp").build();
    }
}
```

View

```
@Controller
@Path("hello")
public class HelloController {

    @View("hello.jsp")
    @GET
    public void hello() {

    }
}
```

View

```
@View("hello.jsp")
@Controller
@Path("hello")
public class HelloController {

    @GET
    public void hello() {

    }
}
```

Models

Model

```
@Named("greeting")
@RequestScoped
public class Greeting {

    private String message;

    public void setMessage(String message) {
        this.message = message;
    }

    public void getMessage() { return message; }
}
```

Model

```
@View("hello.jsp")
@Controller
@Path("hello")
public class HelloController {

    @GET
    public void hello() {

    }
}
```

Model

```
@View("hello.jsp")
@Controller
@Path("hello")
public class HelloController {

    @Inject
    private Greeting greeting;

    @GET
    public void hello() {
        greeting.setMessage("Hello Bengaluru!");
    }
}
```


Model

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>MVC 1.0 Hello Demo</title>
  </head>
  <body>
    <h1>Hello ${greeting.message}</h1>
  </body>
</html>
```

Model

```
@View("hello.jsp")
@Controller
@Path("hello")
public class HelloController {

    @GET
    public void hello() {

    }
}
```

Model

```
@View("hello.jsp")
@Controller
@Path("hello")
public class HelloController {

    @Inject
    private Models model;

    @GET
    public void hello() {
        model.put("message", "Hello Bengaluru!");
    }
}
```

Model

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>MVC 1.0 Hello Demo</title>
  </head>
  <body>
    <h1>Hello ${message}</h1>
  </body>
</html>
```

Validation

Validation

```
@Controller
@Path("form")
public class FormController {

    @POST
    public Response formPost(@Valid @BeanParam FormDataBean f) {

        return Response.status(OK).entity("data.json", "application/json");
    }
}
```

ConstraintValidationException

@Min, @NotNull
etc.

Validation

```
public class FormViolationMapper implements  
    ExceptionMapper<ConstraintViolationException>  
{  
  
    public Response toResponse(ConstraintViolationException e) {  
  
        // process violations ...  
  
        return Response.status(BAD_REQUEST)  
            .entity("error.jsp").build();  
    }  
}
```

Validation

Allows for MVC Error Handling

```
@Controller
@Path("form")
public class FormController {

    @Inject private BindingResult br;

    @POST
    public Response formPost(@Valid @BeanParam FormDataBean f) {
        if (br.isFailed()) {
            return Response.status(BAD_REQUEST)
                .entity("error.jsp").build();
        }
        return Response.status(OK).entity("data.jsp").build();
    }
}
```


Security

Cross Site Request Forgery

```
@Controller
@Path("/csrf")
public class HelloController {

    @CsrfValid
    @POST
    public Response post(@FormParam("greeting") String greet {

    }
}
```

Cross Site Request Forgery

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>MVC 1.0 Hello Demo</title>
  </head>
  <body>
    <form action="csrf" method="post" accept-
charset="utf-8">
      <input type="submit" value="Click here"/>
      <input type="hidden" name="{mvc.csrf.name}"
value="{mvc.csrf.token}"/>
    </form>
  </body>
</html>
```

Scopes

@RedirectScoped

```
@Named("greeting")
@RedirectScoped
public class Greeting {

    private String message;

    public void setMessage(String message) {
        this.message = message;
    }

    public void getMessage() { return message; }
}
```

@RedirectScoped

@RequestScoped

@Controller

```
public class HelloController {
```

@Inject

```
private Greeting greeting;
```

@RedirectScoped

@POST

```
@Path("from")
```

```
public String hello() {  
    greeting.setMessage("Hello JavaLand!");  
    return "redirect:to";  
}
```

```
}
```

```
}
```

@RedirectScoped

```
@Controller
public class HelloController {

    @Inject
    private Greeting greeting;

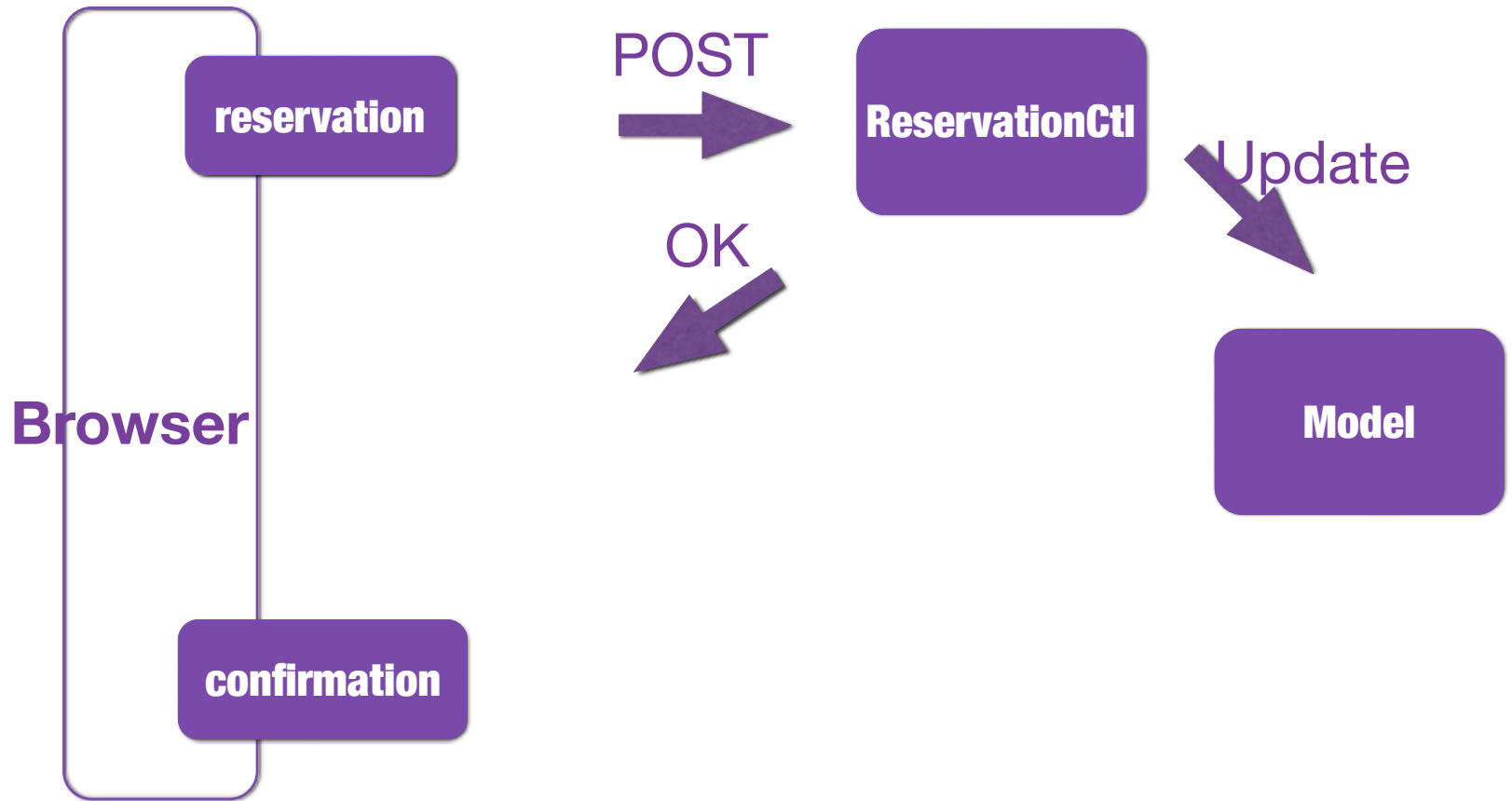
    @POST
    @Path("to")
    public String hello() {

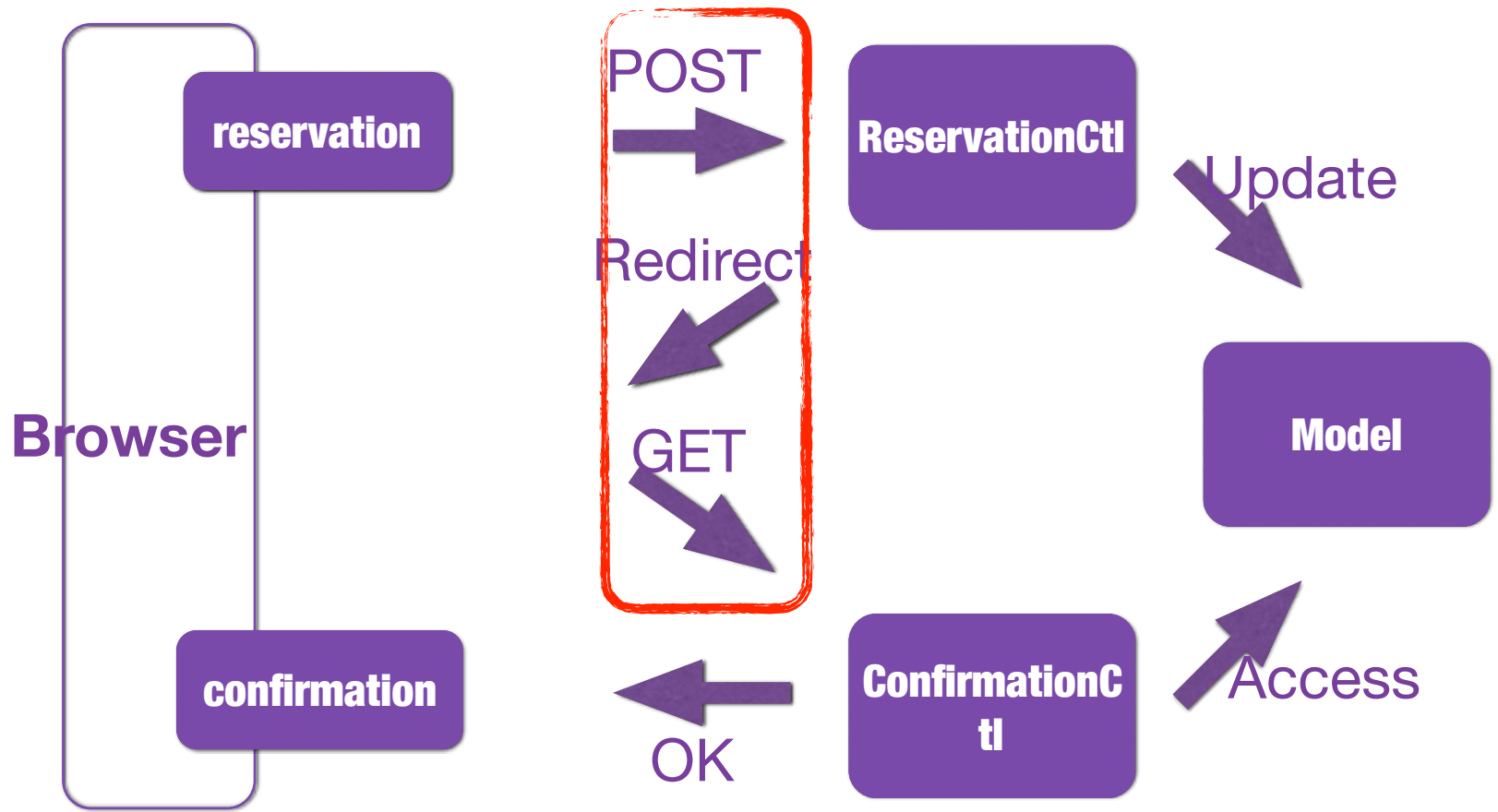
        return "here.jsp";
    }
}
```

Same Bean

Tired of slides?

**Show me the *real*
CODE !**





View Engines

Bring Your Own View Engine

ViewEngine

```
public interface ViewEngine {  
  
    boolean supports(String view);  
  
    void processView(ViewEngineContext context)  
        throws ViewEngineException;  
}
```

ViewEngine

```
@Priority(OUTRAGEOSLY_HIGH)
@ApplicationScoped
public class FreemarkerViewEngine extends ViewEngineBase {
    ...
}
```

Events

BeforeControllerEvent

AfterControllerEvent

BeforeProcessViewEvent

AfterProcessViewEvent

ControllerRedirectEvent

Events Example

```
@ApplicationScoped
public class EventObserver {

    void onBeforeController(@Observes BeforeControllerEvent e) {

    }

    void onAfterController(@Observes AfterControllerEvent e) {

    }

}
```

Tool Support

JPA Modeler

<http://jpamodeler.github.io/>



Gaurav Gupta
@jGauravGupta

Summary

Action-based MVC

Existing Java EE Technologies

Build MVC 1.0 on top of JAX-RS

Project Page

<https://java.net/projects/mvc-spec>

GitHub

<https://github.com/mvc-spec>

Reference Implementation

<https://ozark.java.net>

Samples

<https://github.com/ivargrimstad/mvc-samples>

Blog

<http://www.agilejava.eu/>

cybercom.com