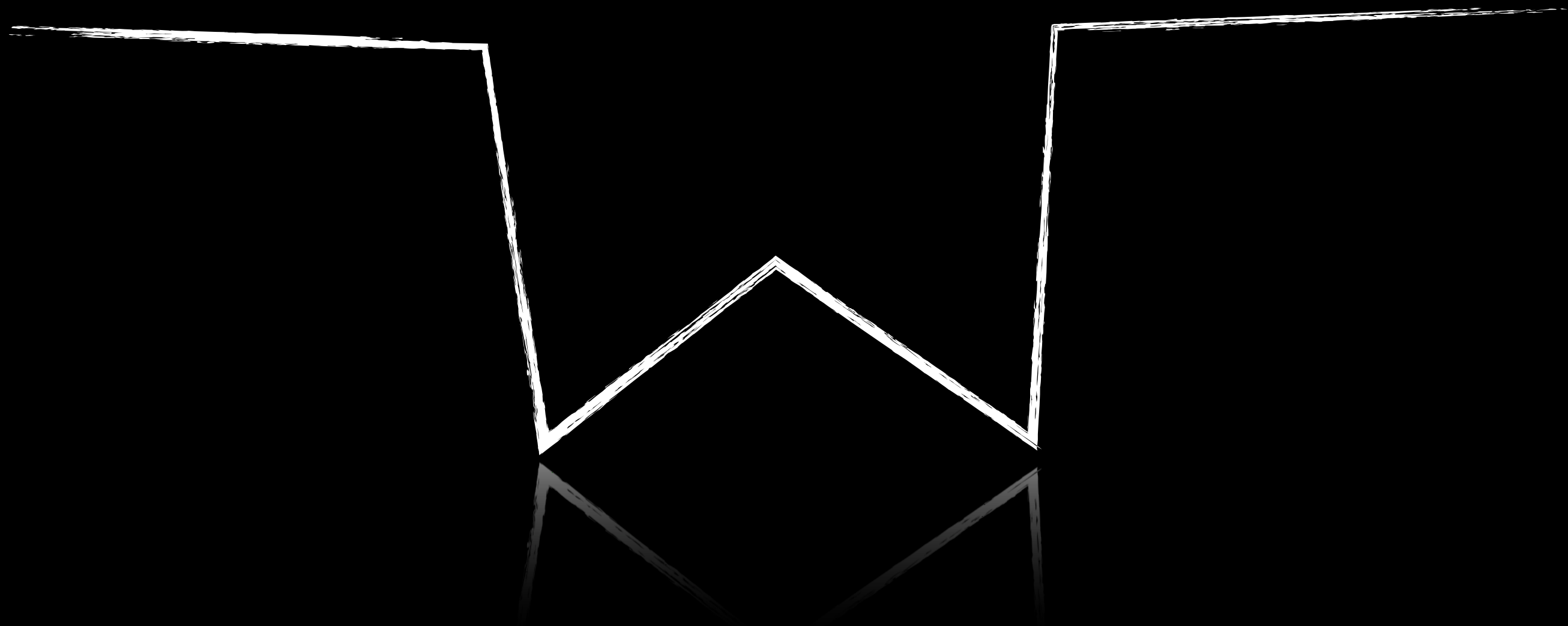


WICKET



Apache Wicket

Michael Plöd / Olaf Siefert
Senacor Technologies AG

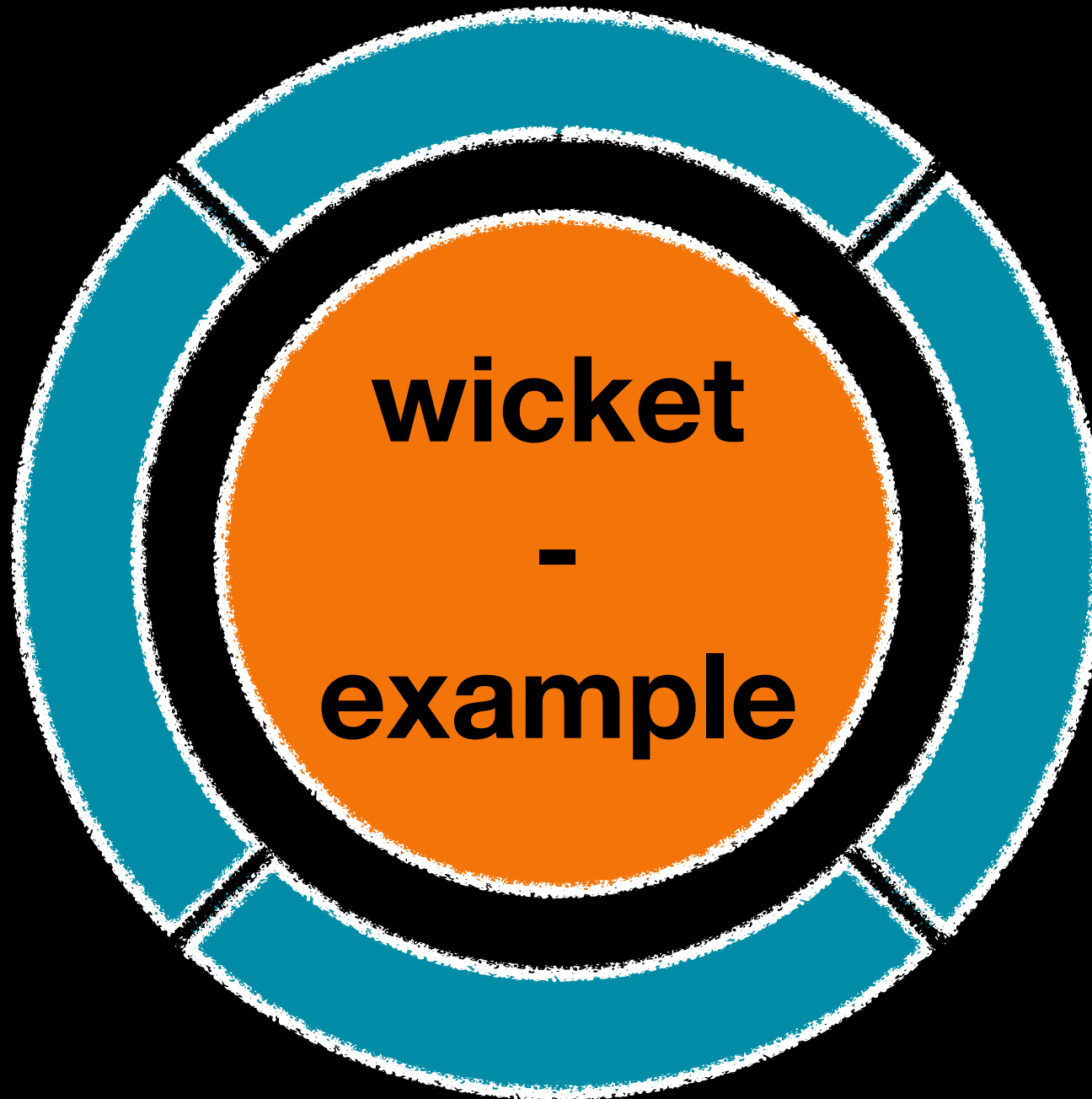


Quickstart

```
mvn archetype:create
```

- DarchetypeGroupId=org.apache.wicket
- DarchetypeArtifactId=
wicket-archetype-quickstart
- DarchetypeVersion=1.4.5
- DgroupId=com.senacor
- DartifactId=wicket-example

WebApplication.java



HomePage.java

HomePage.html

web.xml

web.xml

configures

Application



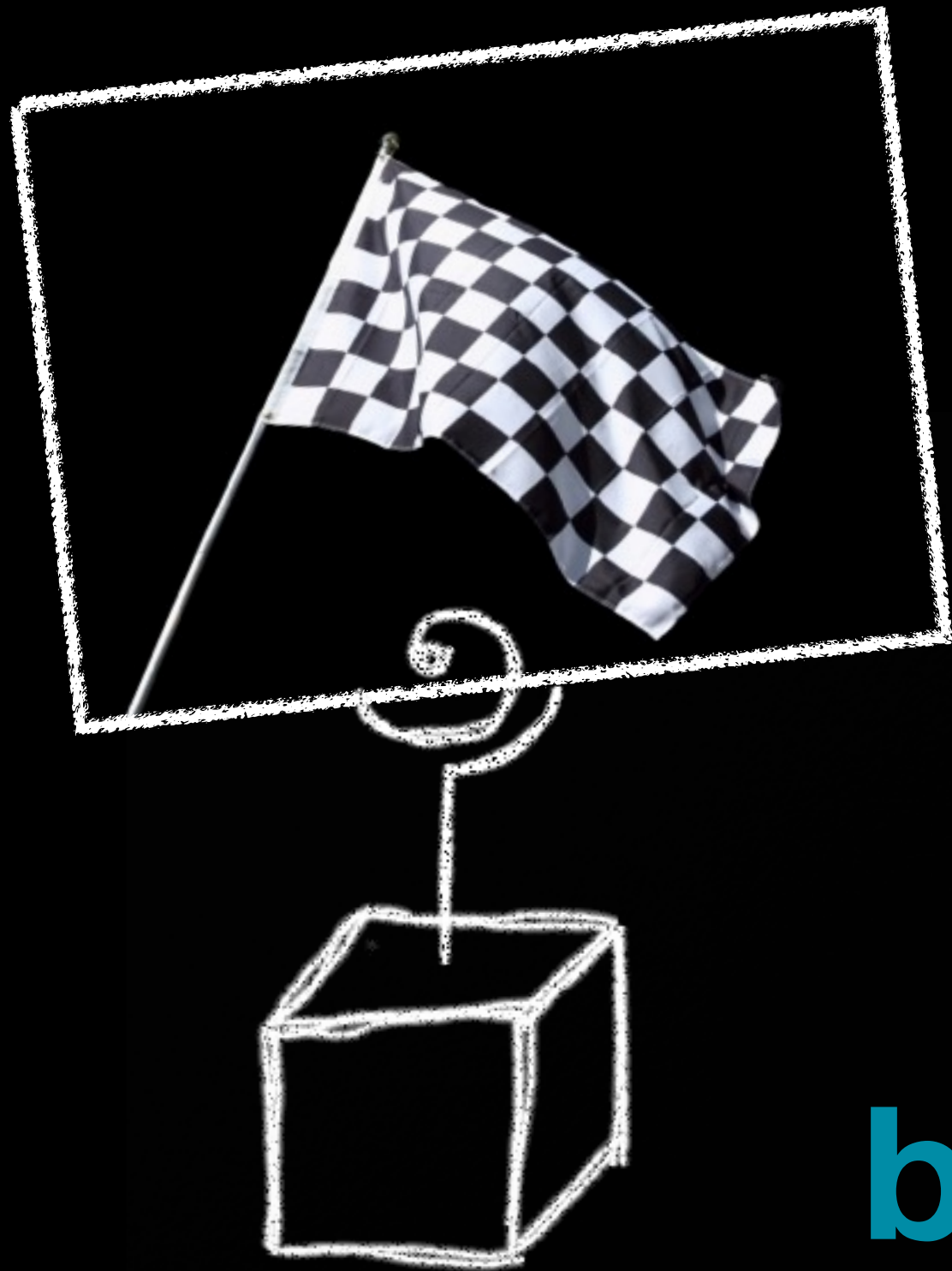
66

```
<web-app ... >
  <display-name>wicket-example</display-name>

  <filter>
    <filter-name>wicket.wicket-example</filter-name>
    <filter-class>org.apache.wicket.protocol.http.WicketFilter</filter-class>
    <init-param>
      <param-name>applicationClassName</param-name>
      <param-value>com.senacor.WicketApplication</param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>wicket.wicket-example</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
</web-app>
```

99

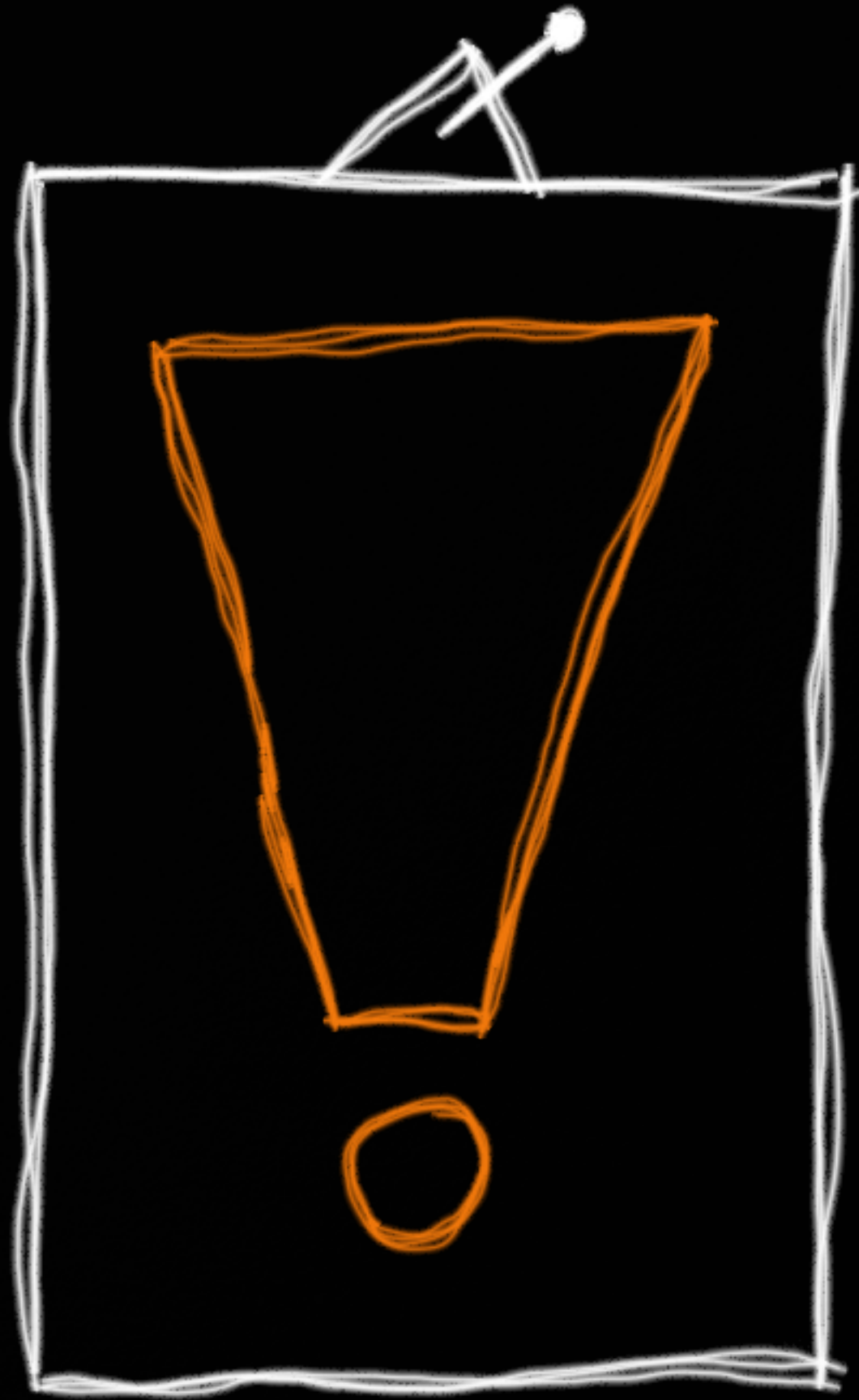


Application
is responsible for
bootstrapping

66

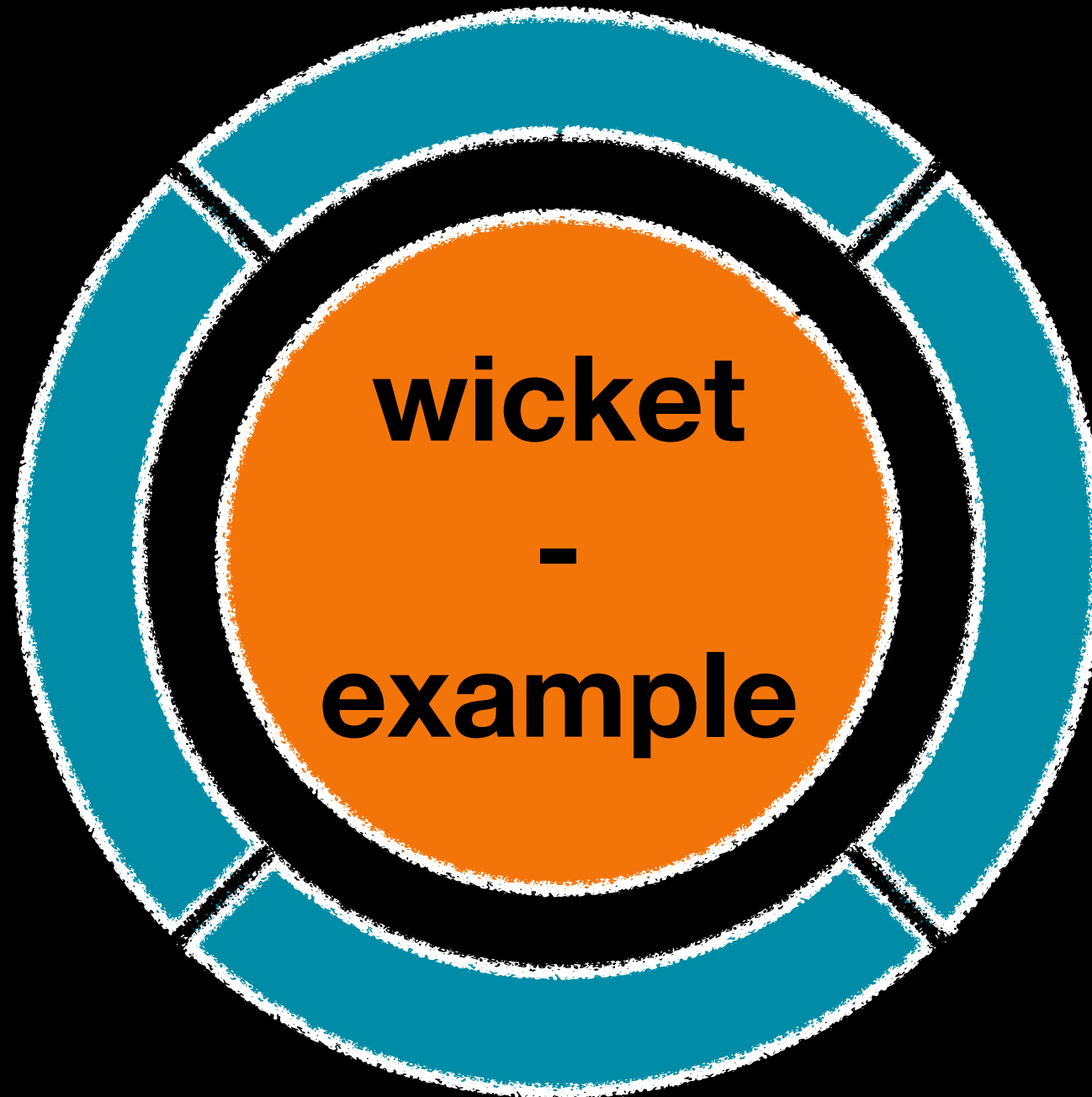
```
package com.senacor;  
  
import org.apache.wicket.protocol.http.WebApplication;  
  
public class WicketApplication extends WebApplication  
{  
    public WicketApplication() {}  
  
    public Class<HomePage> getHomePage()  
    {  
        return HomePage.class;  
    }  
}
```

99



no additional
configuration

WebApplication.java

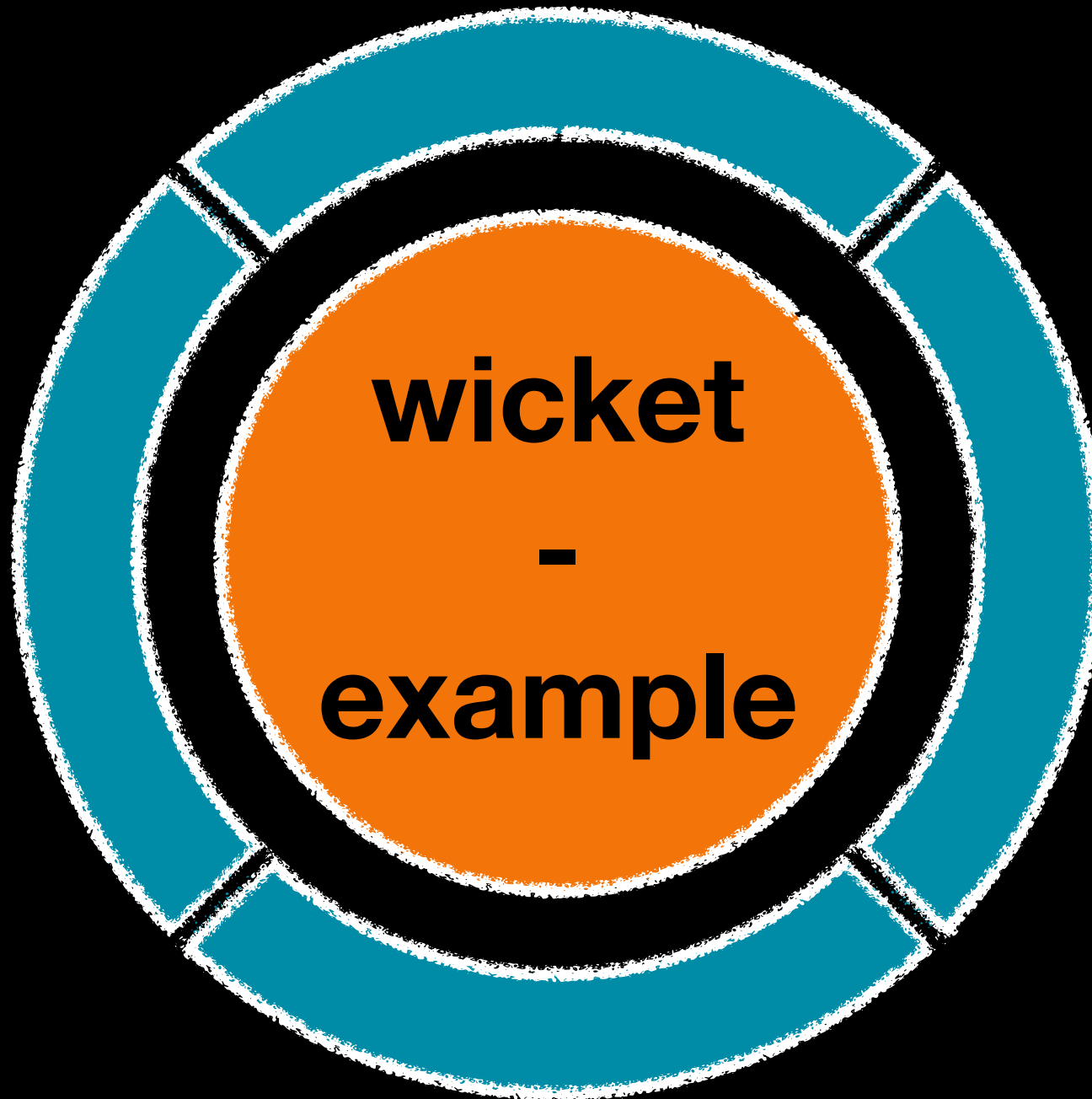


HomePage.java

HomePage.html

web.xml

WebApplication.java



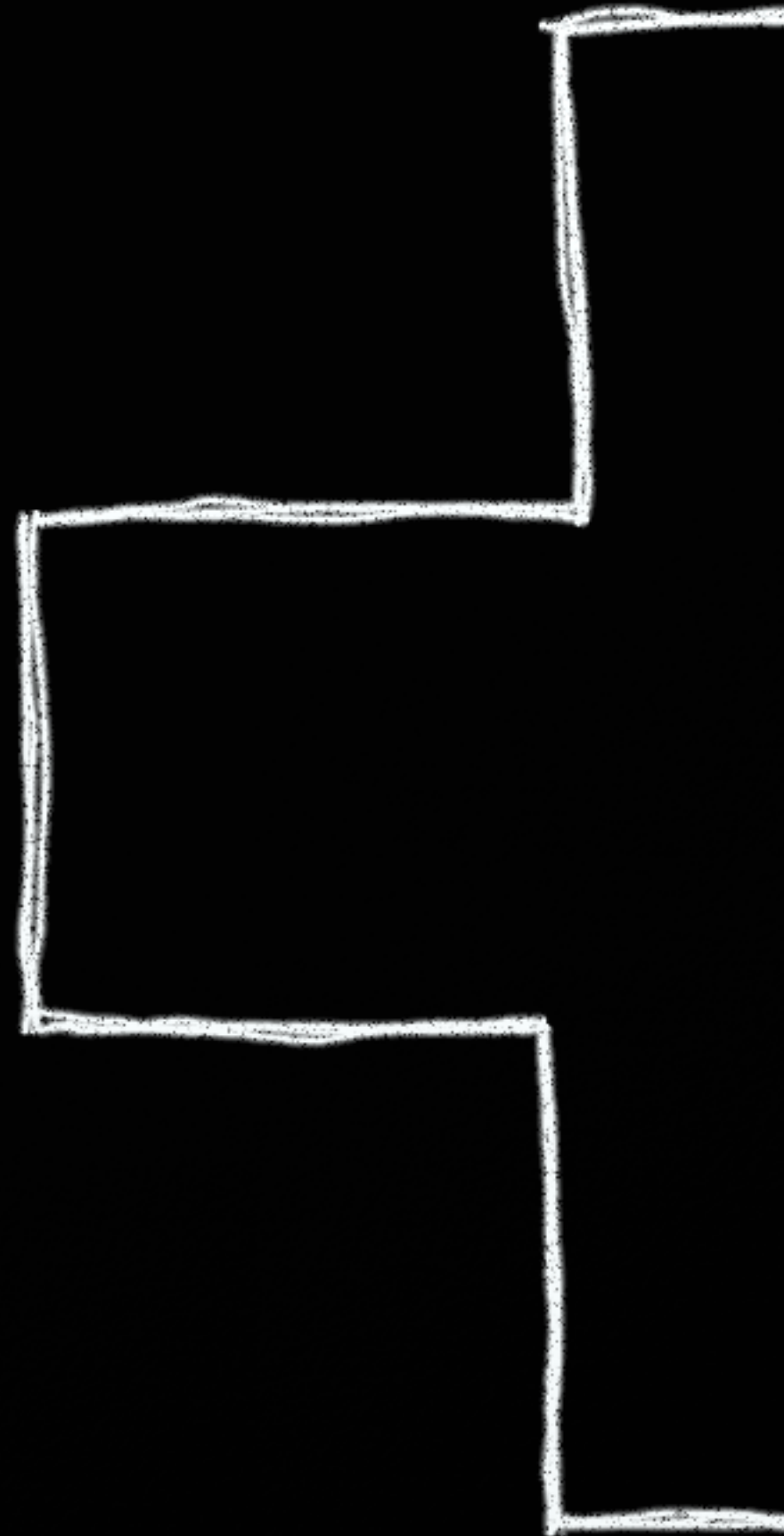
HomePage.java

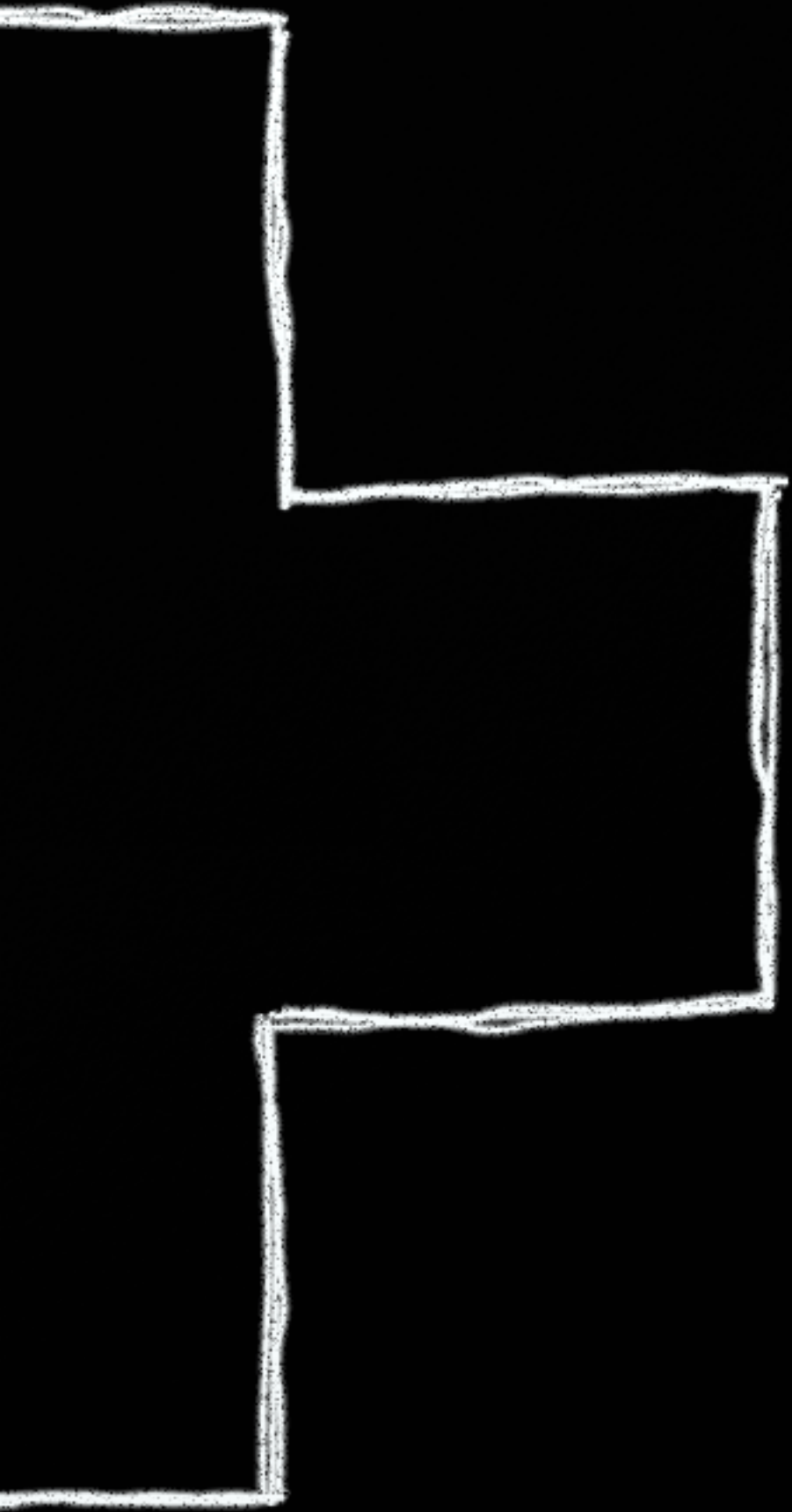
HomePage.html

web.xml

Page

defines the Java part





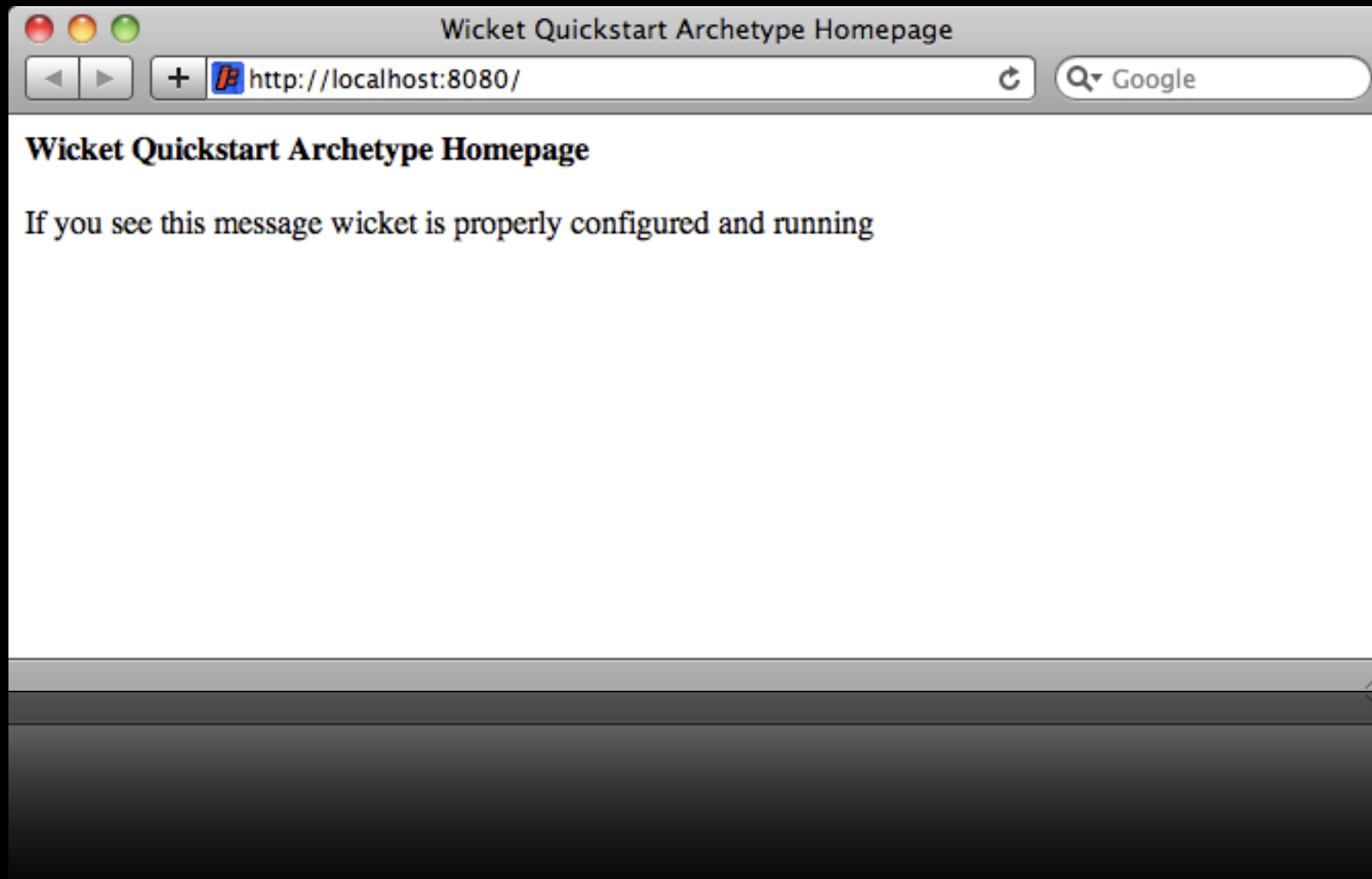
HTML

defines the markup part

66

```
public class HomePage extends WebPage {  
    public HomePage(final PageParameters parameters) {  
        add(new Label("message",  
            "If you see this message wicket is  
            properly configured and running"));  
    }  
}
```

```
<html xmlns:wicket=  
    "http://wicket.apache.org/dtds.data/wicket-xhtml1.4-strict.dtd" >  
    <head>  
        <title>Wicket Quickstart Archetype Homepage</title>  
    </head>  
    <body>  
        <strong>Wicket Quickstart Archetype Homepage</strong>  
        <br/><br/>  
        <span wicket:id="message">message will be here</span>  
    </body>  
</html>
```



`<span wicket:id="message">message will be here</span>`

```
add(new Label("message",  
              "If you see this message wicket is  
              properly configured and running"));
```



Let's dive in



Allschools Network

Latest News

DOWN TO NOTHING - "Number One"

DOWN TO NOTHING haben mit "Number One" einen neuen Song online gestellt. Checkt ihn [Mehr lesen](#)

CHUCK RAGAN - Tourupdate

Hier die Dates: 11.03.2010 Düsseldorf (Ger), Zakk 12.03.2010 Münster (Ger), Sputnikhalle 13.03.2010 Bremen (Ger), ... [Mehr lesen](#)

ALESANA auf Tour mit A SKYLIT DRIVE, EMAROSA, BURY TOMORROW

ALESANA kommen mit A SKYLIT DRIVE, EMAROSA & BURY TOMORROW auf Tour. Hier die Dates: "THE ARTERY FOUNDATION TOUR" 24.04.2010 Leipzig ... [Mehr lesen](#)

DROPKICK MURPHYS - Restkarten verfügbar

Das Konzert am Samstag von den DROPKICK MURPHYS in Offenbach war restlos ausverkauft. Für die drei letzten Termine diese Woche sind noch ... [Mehr lesen](#)

BLAKFISH - Tourstart (w/BIFFY CLYRO)

Die vier Jungs von BLAKFISH werden als Support von BIFFY CLYRO auf den anstehenden Daten spielen werden. Ab Mittwoch werden sie bei fünf ... [Mehr lesen](#)

Verlosung

Hoch den Rock Vol. 4 Verlosung



ALLSCHOOLS PRESENTS: HOCH DEN ROCK VOL. 4

Jeden 2. Freitag im Monat ist das Stahlwerk wieder mit einer amtlichen Rockveranstaltung am Start und mit einer amtlichen Rockveranstaltung am Start und mit einer amtlichen Rockveranstaltung am Start und

Jeden 2. Freitag im Monat ist das Stahlwerk wieder mit einer amtlichen Rockveranstaltung am Start und mit einer amtlichen Rockveranstaltung am Start und

ALLSCHOOLS PRESENTS: HOCH DEN ROCK VOL. 4

Verlosung

Hoch den Rock Vol. 4 Verlosung

Alle gewonnenen Daten können werden, ab Mittwoch werden sie bei fünf ... [Mehr lesen](#)



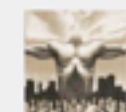
Benutzername:

Passwort:

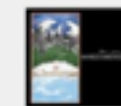
[LOGIN](#)

[Passwort vergessen?](#)

Recommended CDs



Another Breath
The God Complex



Make Do And Mend
Bodies Of Water



Bodies Of Water
Make Do And Mend



The God Complex
Another Breath

Recommended CDs

WHAT'S ON
THIS WEEK



UNDEROATH & ARCHITECTS
Tour 2010



UNDEROATH & ARCHITECTS
Tour 2010

Latest News

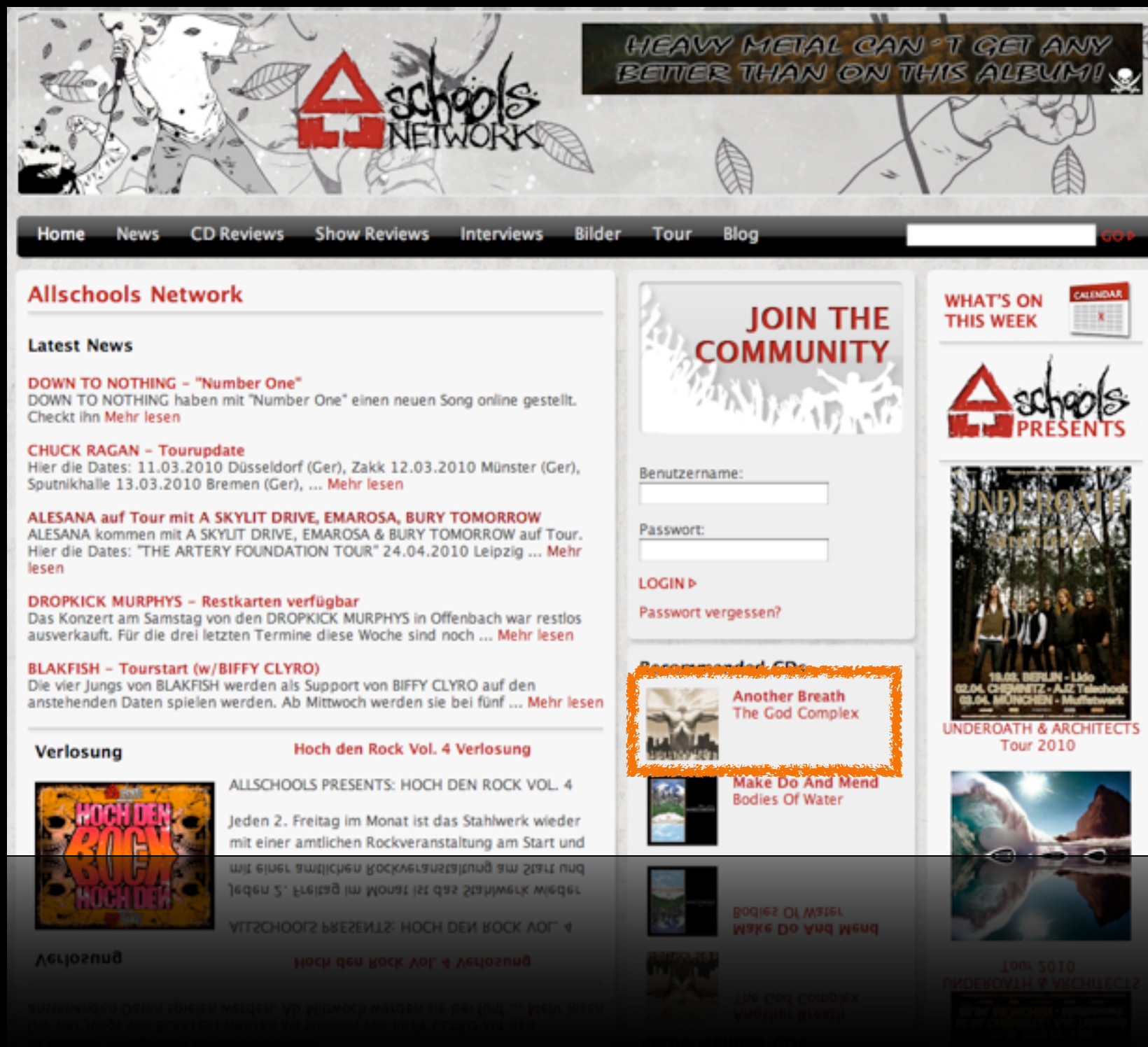
DOWN TO NOTHING haben mit "Number One" einen neuen Song online gestellt. Checkt ihn [Mehr lesen](#)

Hier die Dates: 11.03.2010 Düsseldorf (Ger), Zakk 12.03.2010 Münster (Ger), Sputnikhalle 13.03.2010 Bremen (Ger), ... [Mehr lesen](#)

Das Konzert am Samstag von den DROPKICK MURPHYS in Offenbach war restlos ausverkauft. Für die drei letzten Termine diese Woche sind noch ... [Mehr lesen](#)

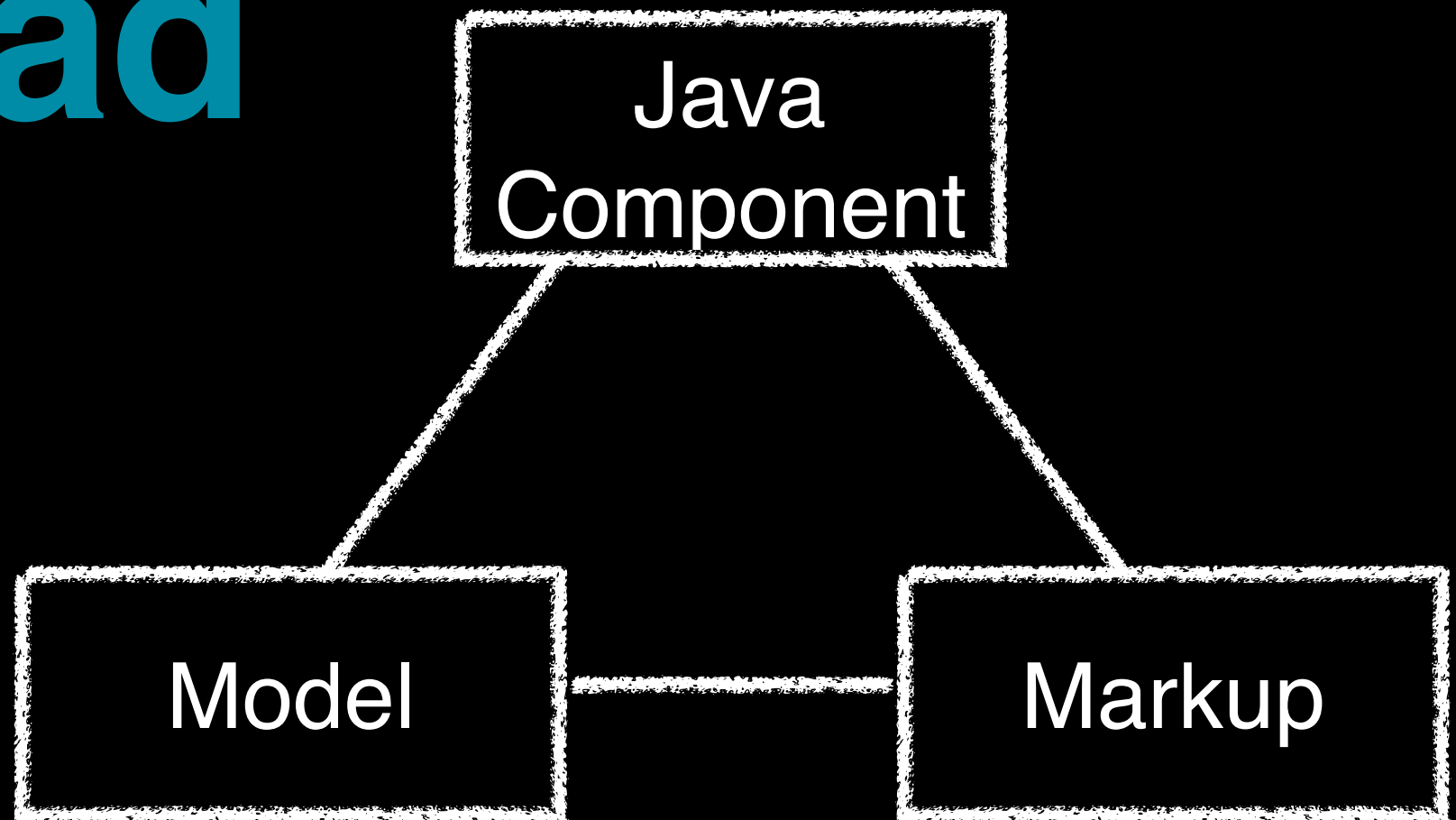
Die vier Jungs von BLAKFISH werden als Support von BIFFY CLYRO auf den anstehenden Daten spielen werden. Ab Mittwoch werden sie bei fünf ... [Mehr lesen](#)

Hoch den Rock Vol. 4 Verlosung



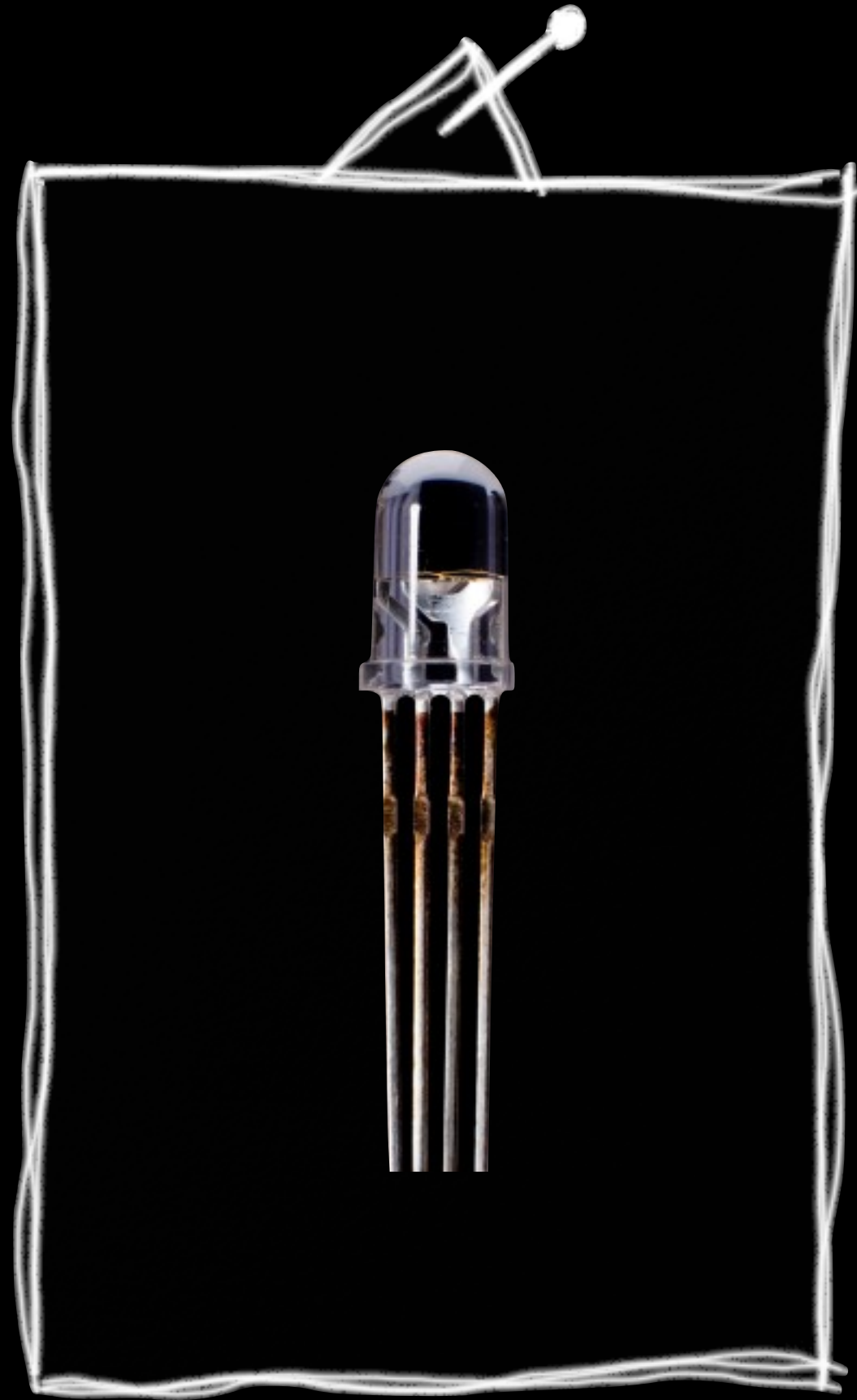
Custom
Component

Wicket Component Triad

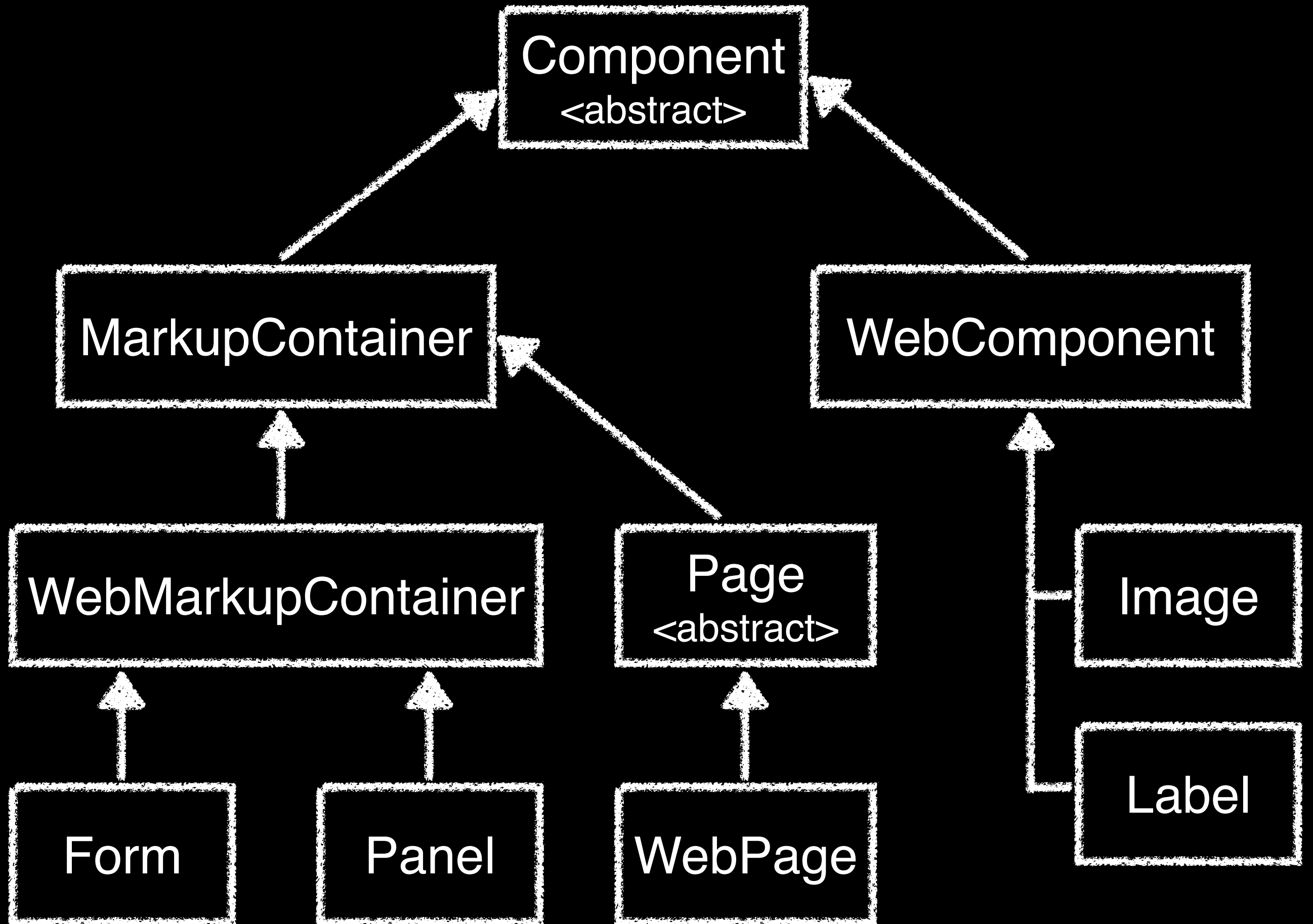




**Java components provide
behaviour**

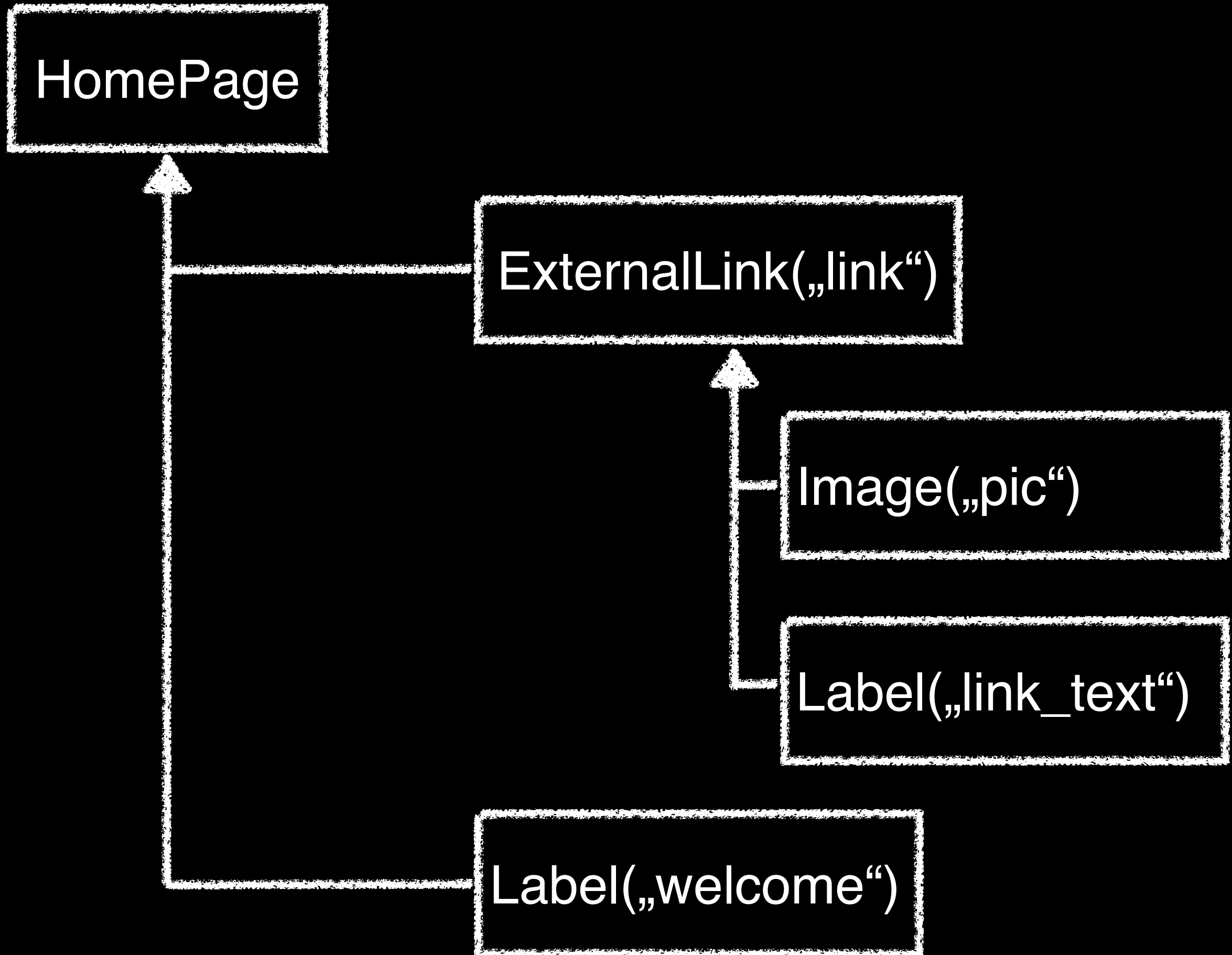


On the Java side
Everything
is a
Component





A Page
is the head of a
Component
Tree



66

```
ExternalLink link = new ExternalLink("link", "http://www.twitter.com");  
link.add(new Label("link_text", "Twitter"));  
link.add(new Image("pic", "twitter.png"));  
add(link);  
add(new Label("welcome", "My link collection"));
```

```
<body>  
  <span wicket:id= "welcome"/>  
  <a wicket:id="link">  
    <img wicket:id="pic"/>  
    <span wicket:id="link_text"/>  
  </a>  
</body>
```

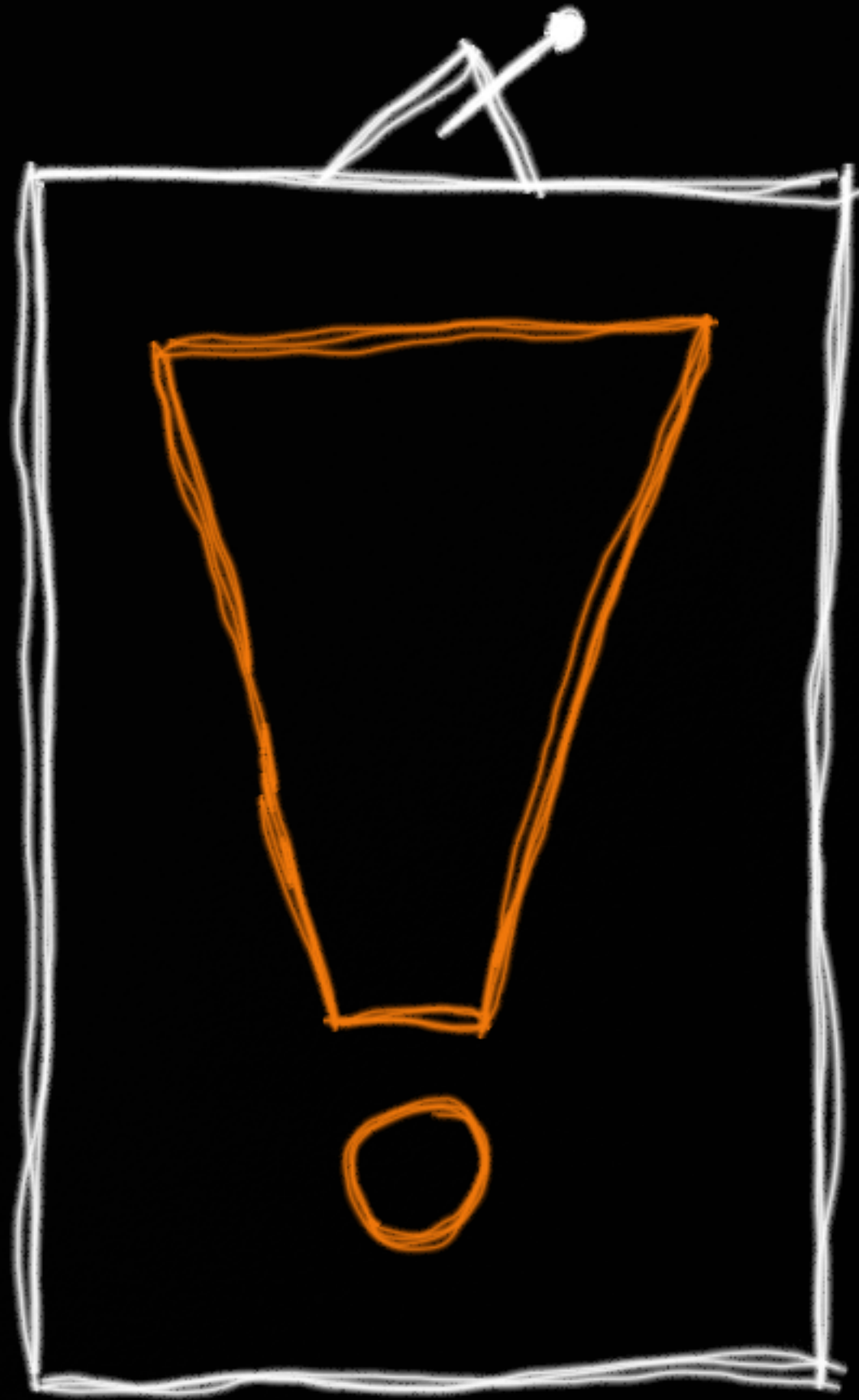
99



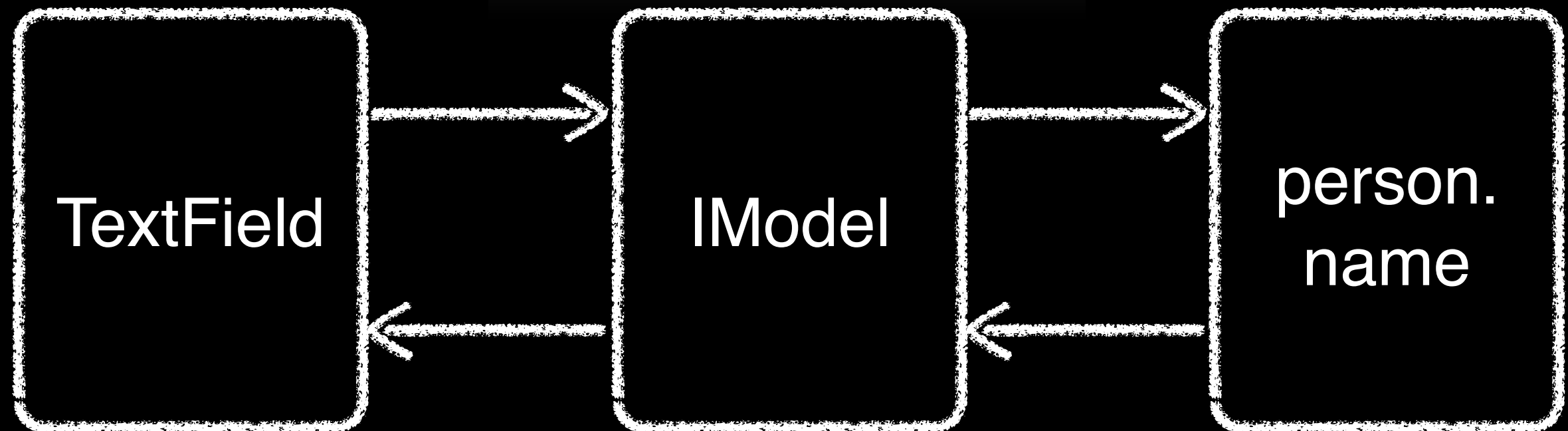
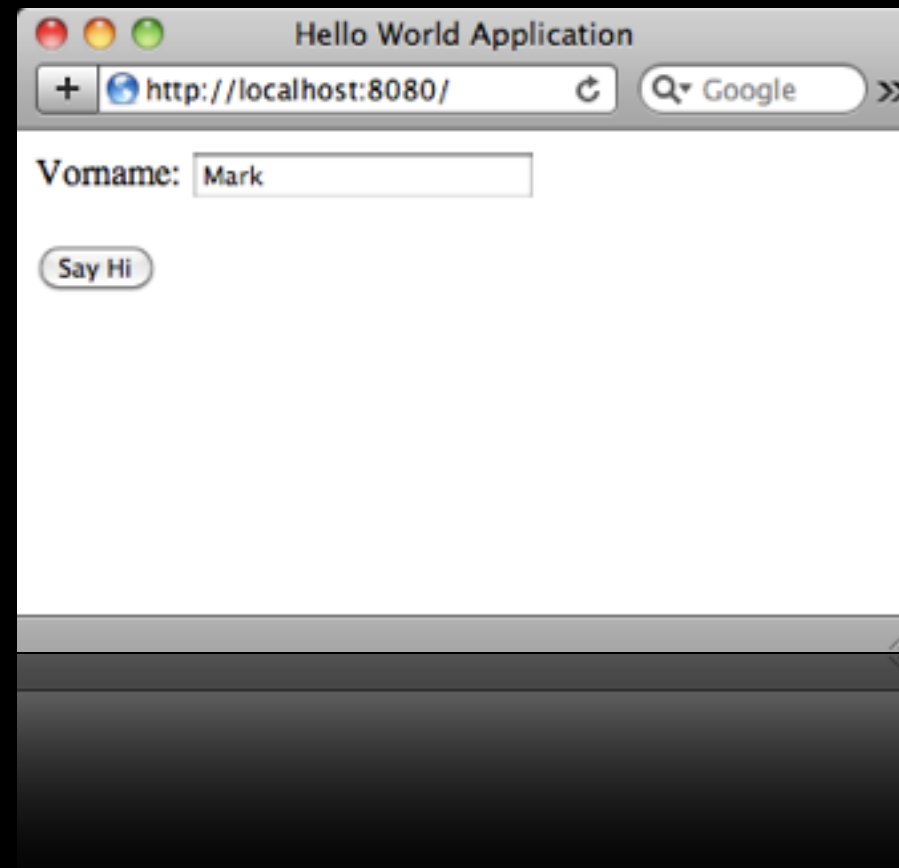
Models



Models
provide data



Understanding
Models is crucial



66

Model

```
new Label<String>("lastname", "Plöd");
```

```
new Label<String>("lastname", person.getLastname());
```

```
new Label<Date>("birthday",  
                new Model<Date>(person.getBirthday()));
```

99

66

```
final SimpleDateFormat df = ...
```

```
String time = df.format(new Date());  
Model clock = new Model(time);  
add(new Label("static_clock", clock));
```

```
Model<String> dynClock = new Model<String>() {  
    @Override  
    public String getObject() {  
        return df.format(new Date());  
    }  
};  
add(new Label("dynamic_clock", dynClock));
```

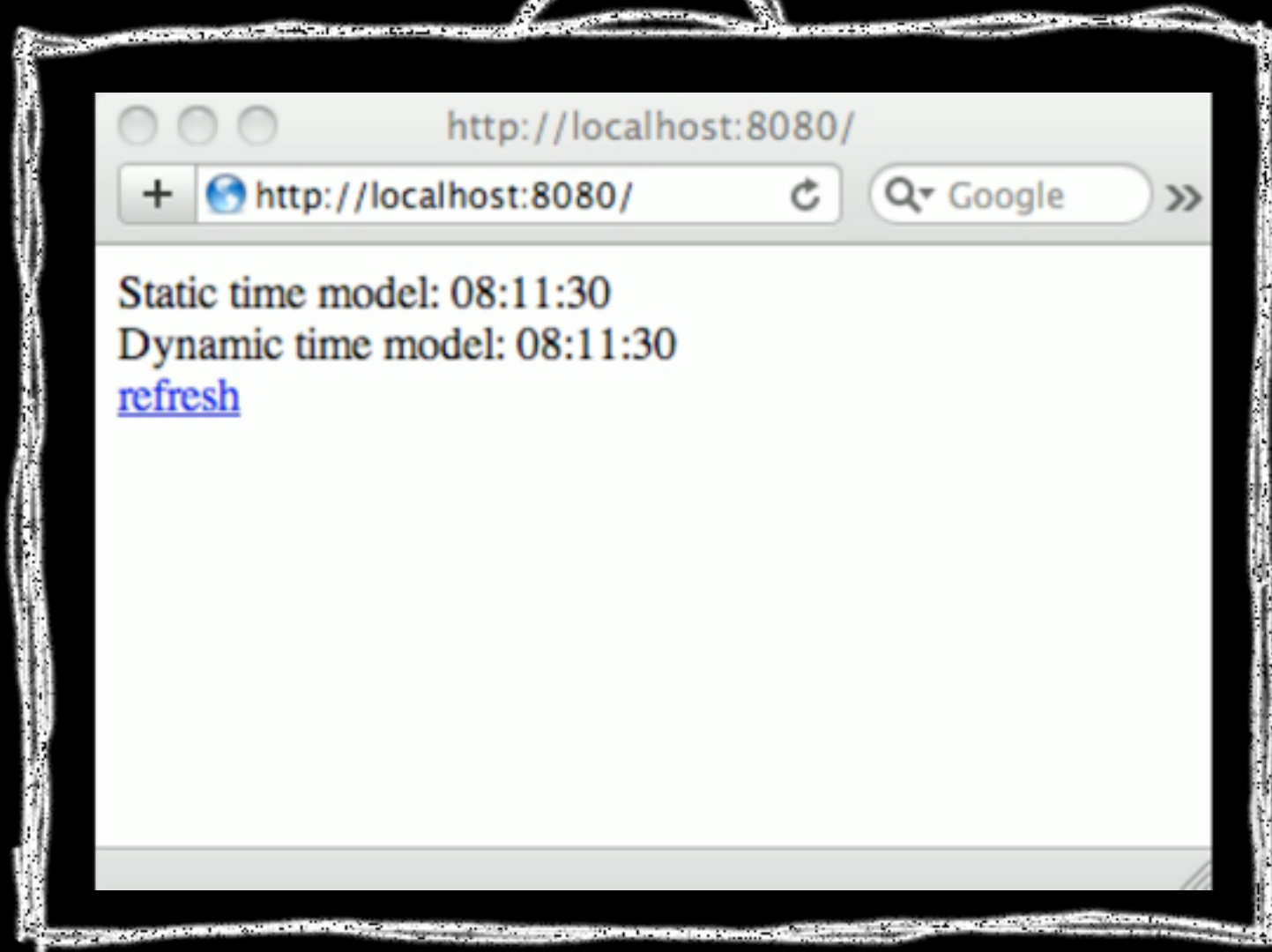
```
add(new Link("refresh"){ public void onClick() {}});
```

99

Static

vs

Dynamic Models



66

```
final SimpleDateFormat df = ...
```

```
String time = df.format(new Date());
```

```
Model clock = new Model(time);
```

```
add(new Label("static_clock", clock));
```

```
Model<String> dynClock = new Model<String>() {
```

```
    @Override
```

```
    public String getObject() {
```

```
        return df.format(new Date());
```

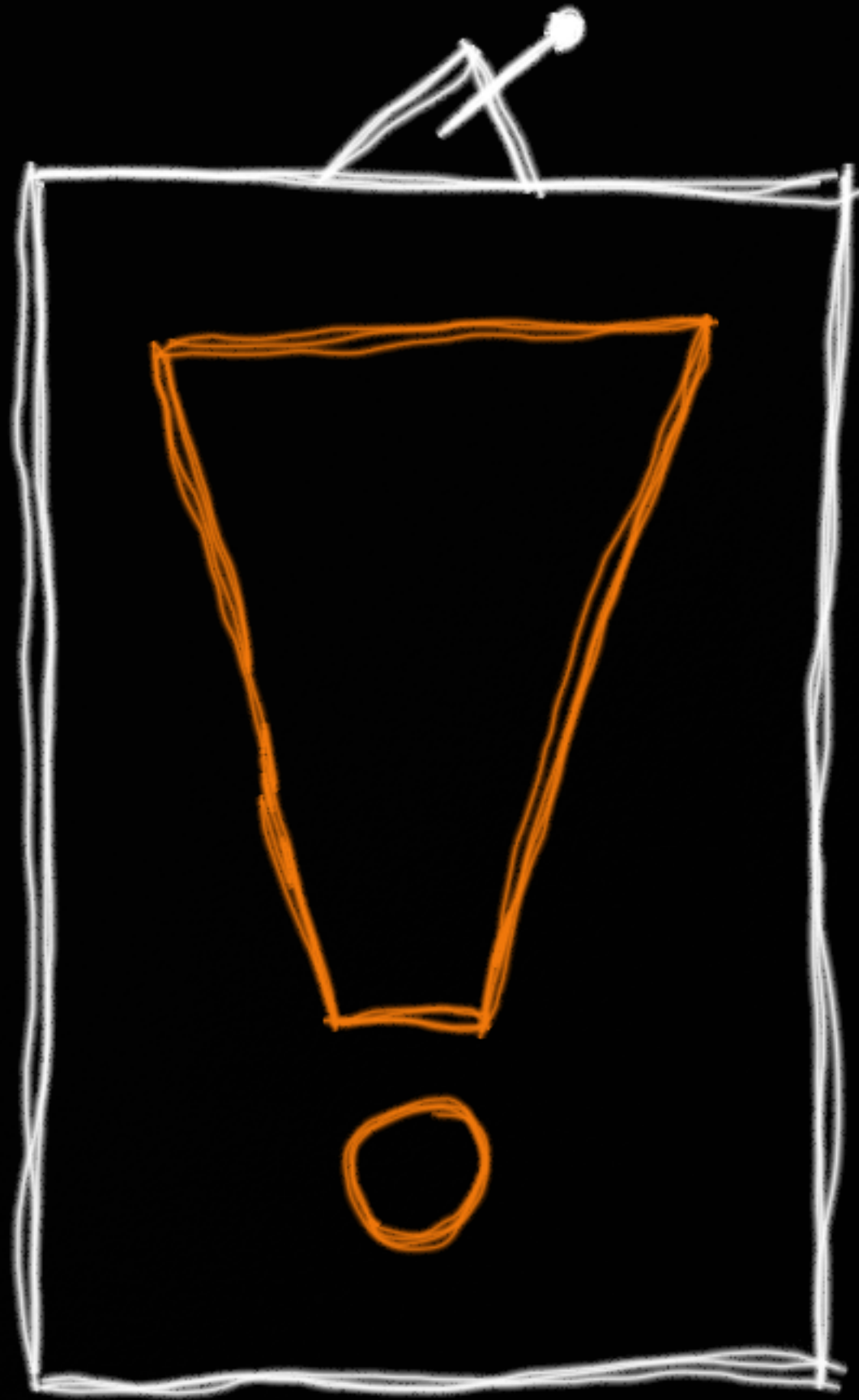
```
    }
```

```
};
```

```
add(new Label("dynamic_clock", dynClock));
```

```
add(new Link("refresh"){ public void onClick() {}});
```

99



Model is **not**
null-safe

66

PropertyModels

```
Person p = ....;  
new Label<Date>("birthday",  
               new PropertyModel<Date>(p, "birthday"));  
new Label<String>("firstname",  
                 new PropertyModel<String>(p, "firstname"));  
  
setModel(new CompoundPropertyModel(p));  
new Label<Date>("birthday");  
new Label<String>("firstname");
```

99

Each component tree

gets

Serialized

to a

PageMap



Detachable Models

reduce memory footprint



66

```
private class PersonModel extends LoadableDetachableModel {  
    private long id;  
    public PersonModel(long id) {  
        this.id = id;  
    }  
    @Override  
    protected Object load() {  
        return personService.getPersonById(this.id);  
    }  
}
```

```
setModel(new CompoundPropertyModel(new PersonModel(123));  
new Label<Date>("birthday");  
new Label<String>("firstname");
```

99

RULES

- ☐ **Self-Contained**
- ☐ **Plain Java**
- ☐ **Reusable**

Form Processing

Successfully completed necessary work placement
Developing Graduate Training Scheme
Engineering Project Team.

Education

Postgraduate College 1991
Institute of Technology

2.1 BA (Hons) in Software Engineering
Jobsville University

2 S levels grade I and II, 1987
5 A Levels grade A, 1987
Jobsville High School

References

John Doe
Chief Technical Officer
Software Company
Computer House
Jobsville
01234 123456

To obtain employment as a software developer in a dynamic computer

Highlights

- 9 years IT development experience
- Friendly, flexible, and pro-active manner
- Successful at customer facing work
- Flair for creativity and design
- Extensive Internet and Intranet development experience

Employment History

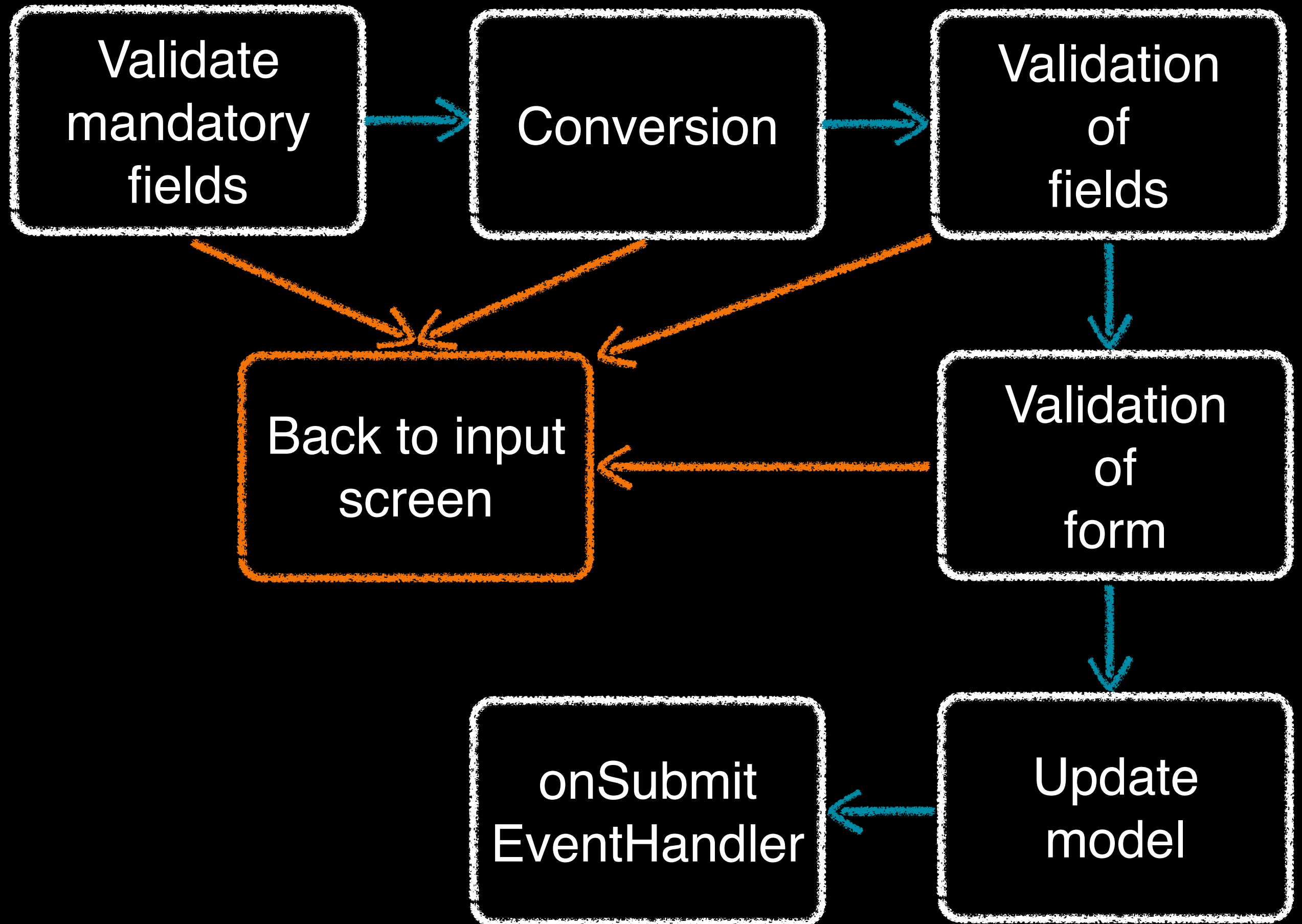
Software Developer, Computer Company, February 2001 – present

I am a key member in the software team responsible for the development of cutting edge internet search technology. I have developed several major products and bespoke web applications covering all stages of the software development cycle. Additionally I am responsible for project management and customer support on a number of high profile projects.

Major Accomplishments

- Produced state of the art image recognition software for automated logo detection
- Developed several web based applications using advanced Artificial Intelligence techniques
- Successfully project managed the development of a high profile project from design to installation
- Delivered bespoke solutions to meet specific customer requirements
- Ported software to various platforms
- Developed of intranet
- Full development of new company website

Software Engineer



66

```
Form<Person>form = new Form<Person>("form") {  
    @Override  
    protected void onSubmit() {  
        Person p = getModelObject();  
        // do something  
    }  
};
```

```
form.setModel(  
    new CompoundPropertyModel<Person>(new Person()));  
form.add(new TextField("firstname"));  
form.add(new TextField("lastname"));  
form.add(new Button("save"));  
add(form);
```

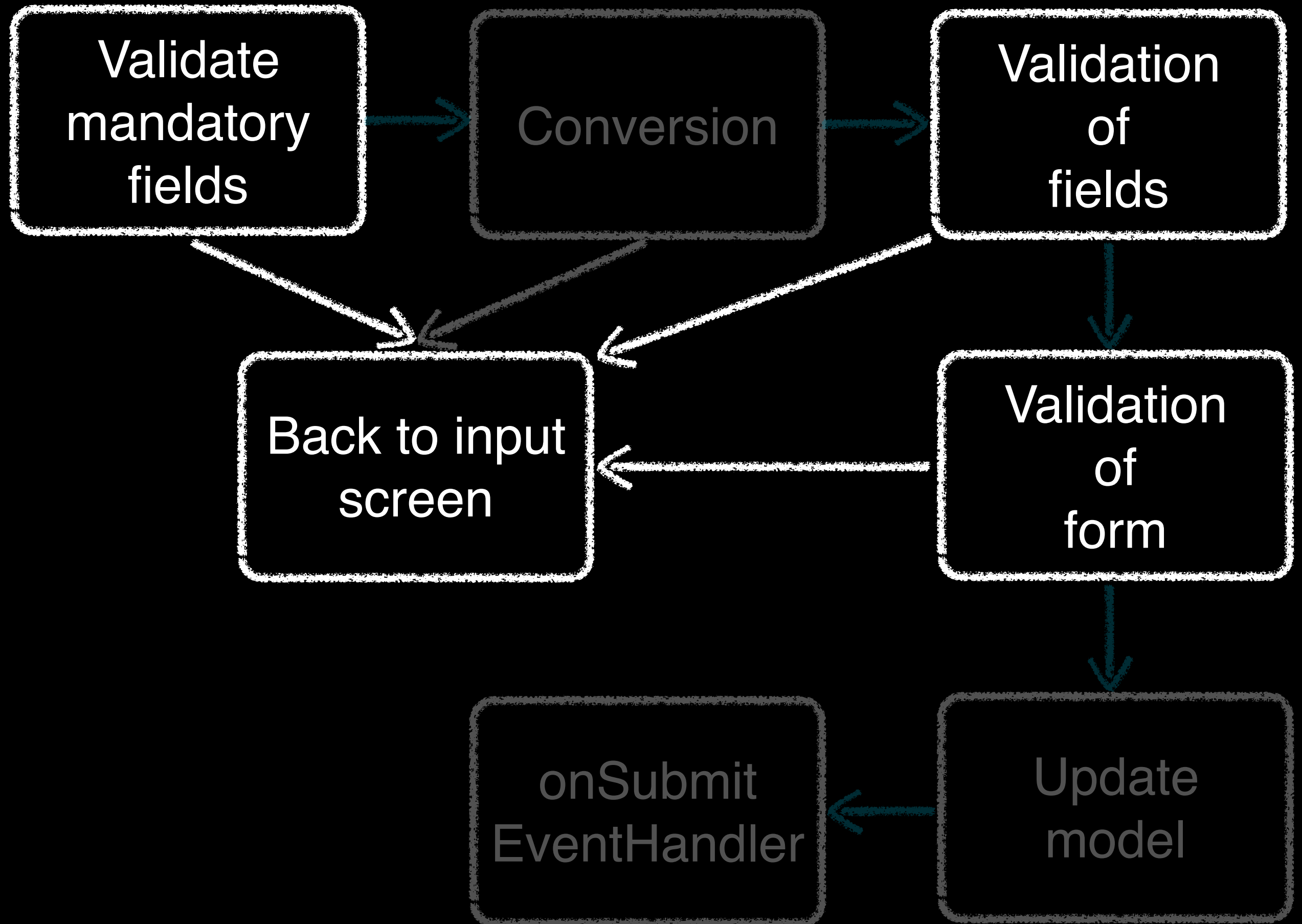
99

„Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the universe trying to build bigger and better idiots.

So far, the universe is winning.“

-- Robert Cringely --





66

```
form.add(new TextField("firstname").setRequired(true));
```

```
form.add(new TextField("email")  
        .setRequired(true)  
        .add(EmailValidator.getInstance()));
```

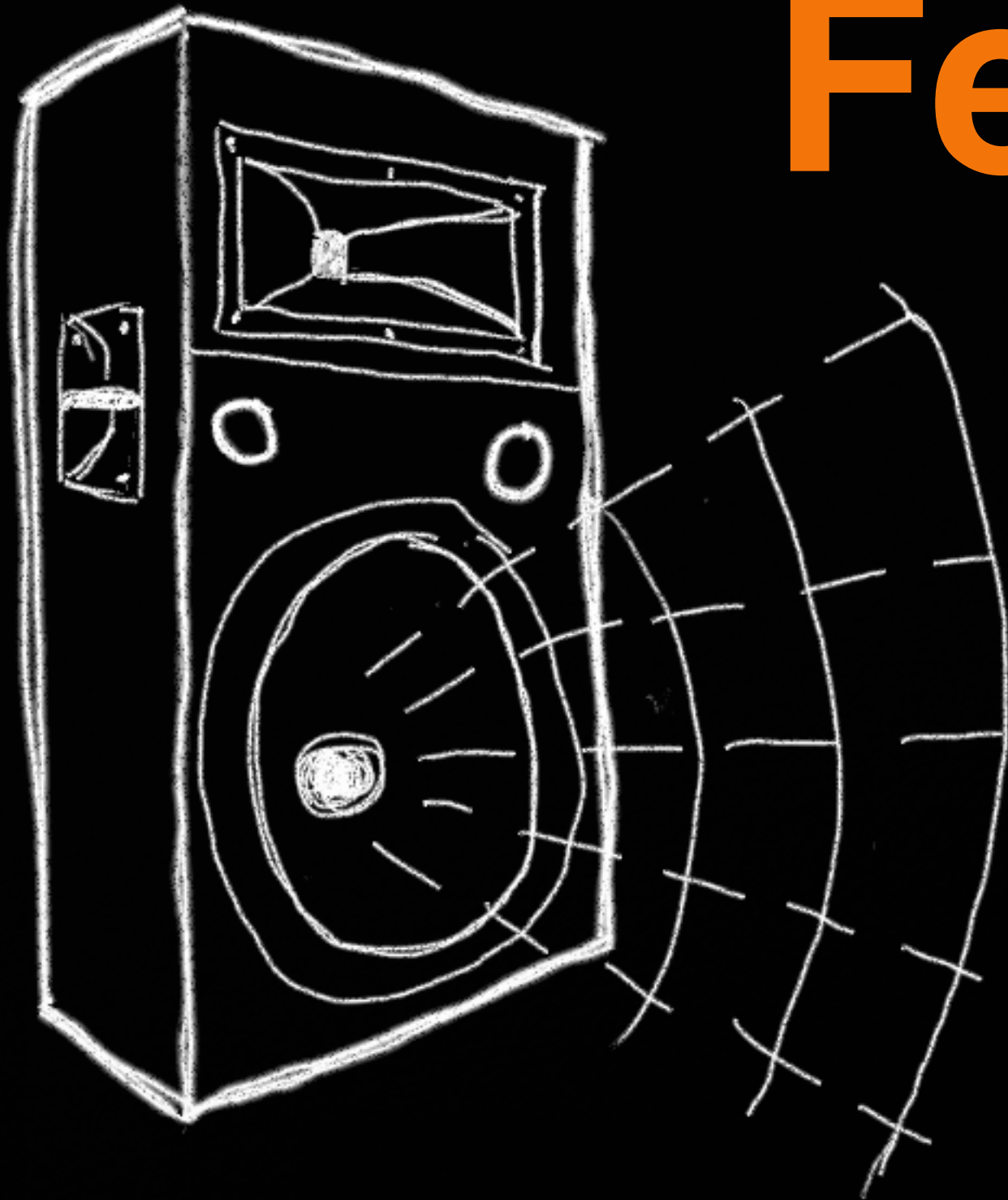
```
form.add(new TextField("age")  
        .add(NumberValidator.minimum(18)));
```

```
form.add(new TextField("username")  
        .add(StringValidator.lengthBetween(6, 10)));
```

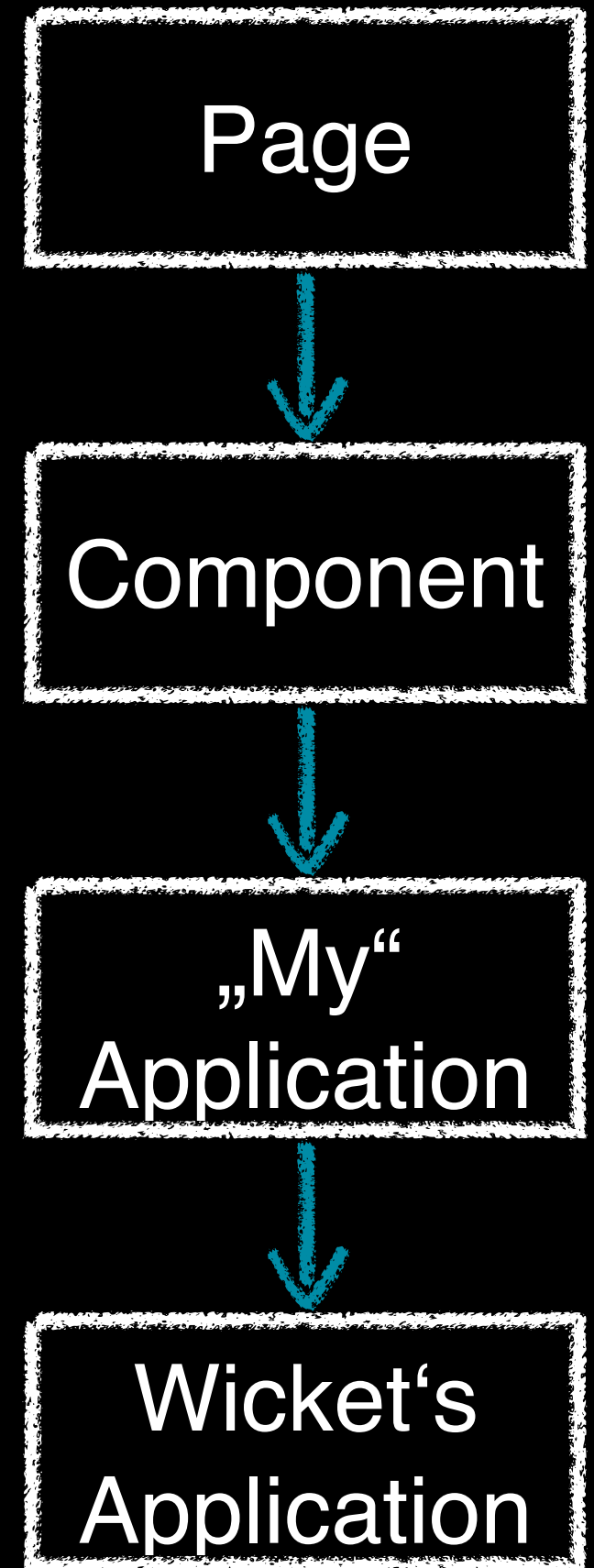
```
form.add(new TextField("username")  
        .add(new PatternValidator("[a-zA-Z0-9]*"))  
        .add(StringValidator.lengthBetween(6, 10)));
```

99

Providing Feedback



Hierarchy of Feedback Messages



66 **#MyPage.java**

```
Form form = new Form("myform");  
form.add(new TextField("email")  
        .setRequired(true)  
        .add(EmailAddressValidator.getInstance()));  
form.add(new TextField("age")  
        .setRequired(true)  
        .add(NumberValidator.minimum(18)));  
form.add(new FeedbackPanel("feedback"));
```

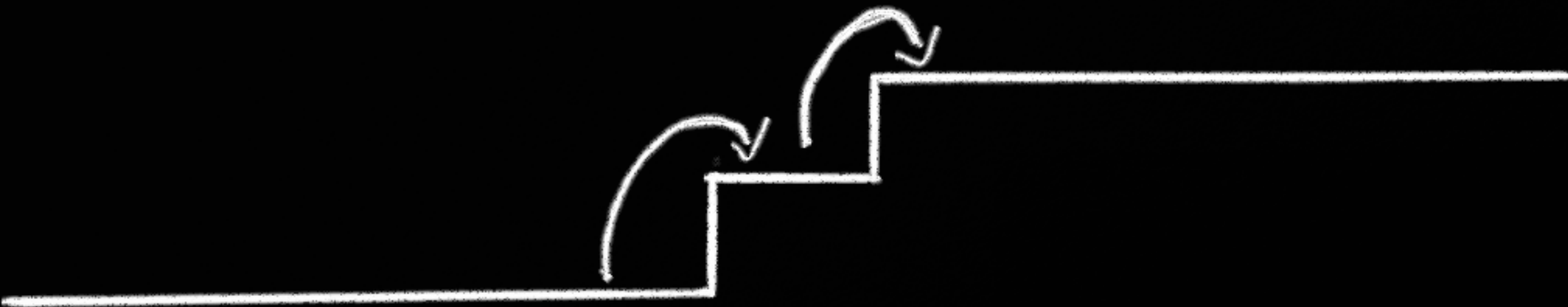
#MyPage.java

```
Form form = new Form("myform");  
form.add(new TextField("email")  
        .setRequired(true)  
        .add(EmailAddressValidator.getInstance()));  
form.add(new TextField("age")  
        .setRequired(true)  
        .add(NumberValidator.minimum(18)));  
form.add(new FeedbackPanel("feedback"));
```

#MyPage.properties

```
Required=Please enter a ${label}  
myform.email.Required=Please provide an email address  
myform.email.EmailAddressValidator=Email address invalid  
NumberValidator.minimum=Not enough
```

Let's take the



next steps

AJAX

is easy



66

```
final SimpleDateFormat df = ...
```

```
String time = df.format(new Date());  
Model clock = new Model(time);  
add(new Label("static_clock", clock));
```

```
Model<String> dynClock = new Model<String>() {  
    @Override  
    public String getObject() {  
        return df.format(new Date());  
    }  
};
```

```
add(new Label("dynamic_clock", dynClock));
```

```
add(new Link("refresh"){ public void onClick() {} });
```

```
Model<String> dynClock = new Model<String>() {  
    ...  
};  
final Label label = new Label("dynamic_clock", dynClock);  
label.setOutputMarkupId(true);  
add(label);  
  
add(new AjaxLink("refresh") {  
    @Override  
    public void onClick(AjaxRequestTarget target) {  
        target.addComponent(label);  
    }  
});
```

YES

that's all

66

```
final Form<Person>form = new Form<Person>("form");
form.setOutputMarkupId(true);
form.setModel(...);
form.add(new TextField("id").setActive(false));
form.add(new TextField("firstname"));
form.add(new TextField("lastname"));
form.add(new AjaxSubmitLink("save") {
    @Override
    public void onClick(AjaxRequestTarget target, Form<Person> f) {
        Person p = f.getModelObject();
        Person savedPerson = save(p);
        form.setModel(...); //with saved Person
        target.addComponent(form);
    }
});
```

99

AJAX Components

AjaxLink

AjaxSubmitLink

AutoCompleteTextField

AjaxEditableLabel

There is no difference

between

AJAX-

and

usual-

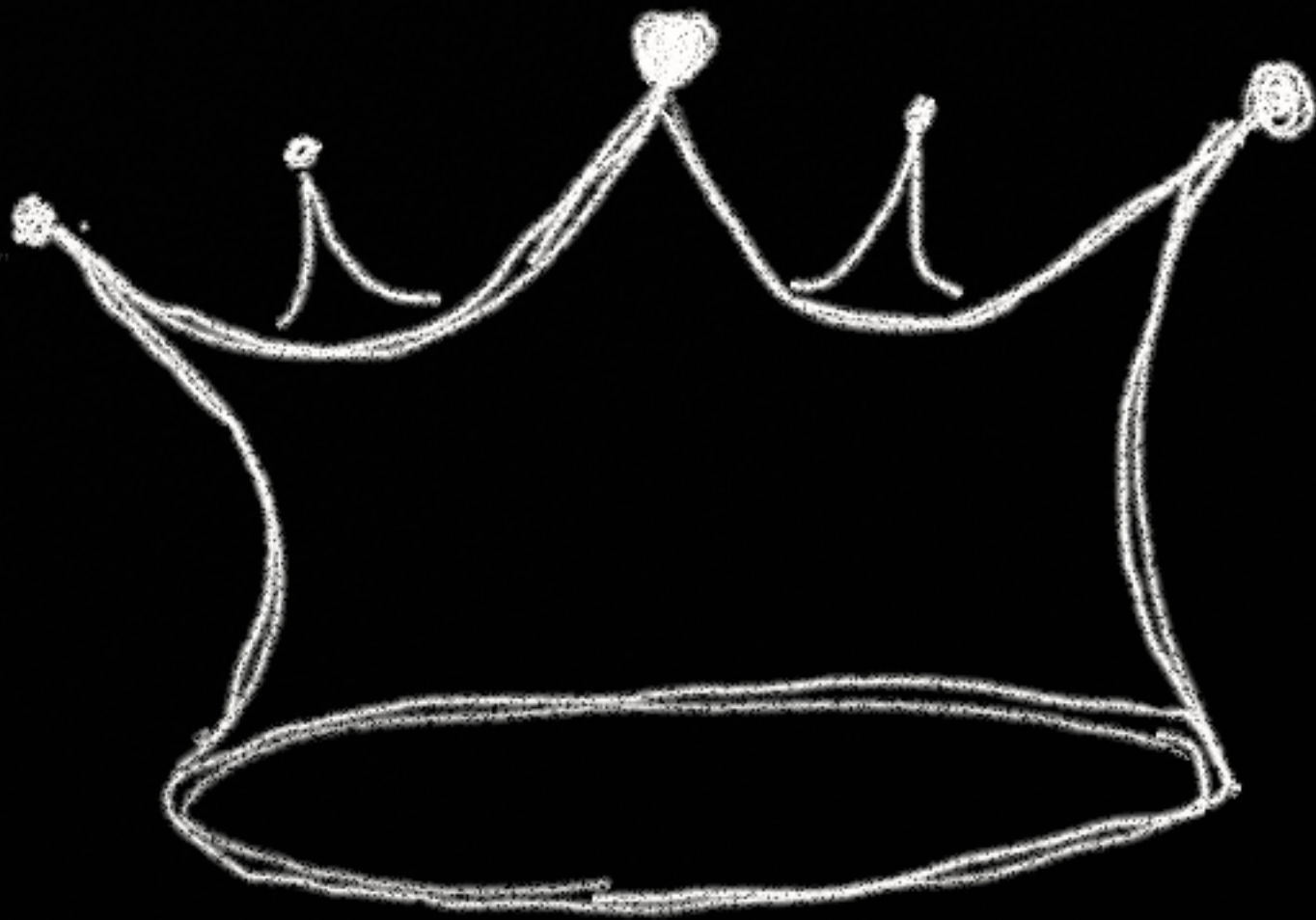
components





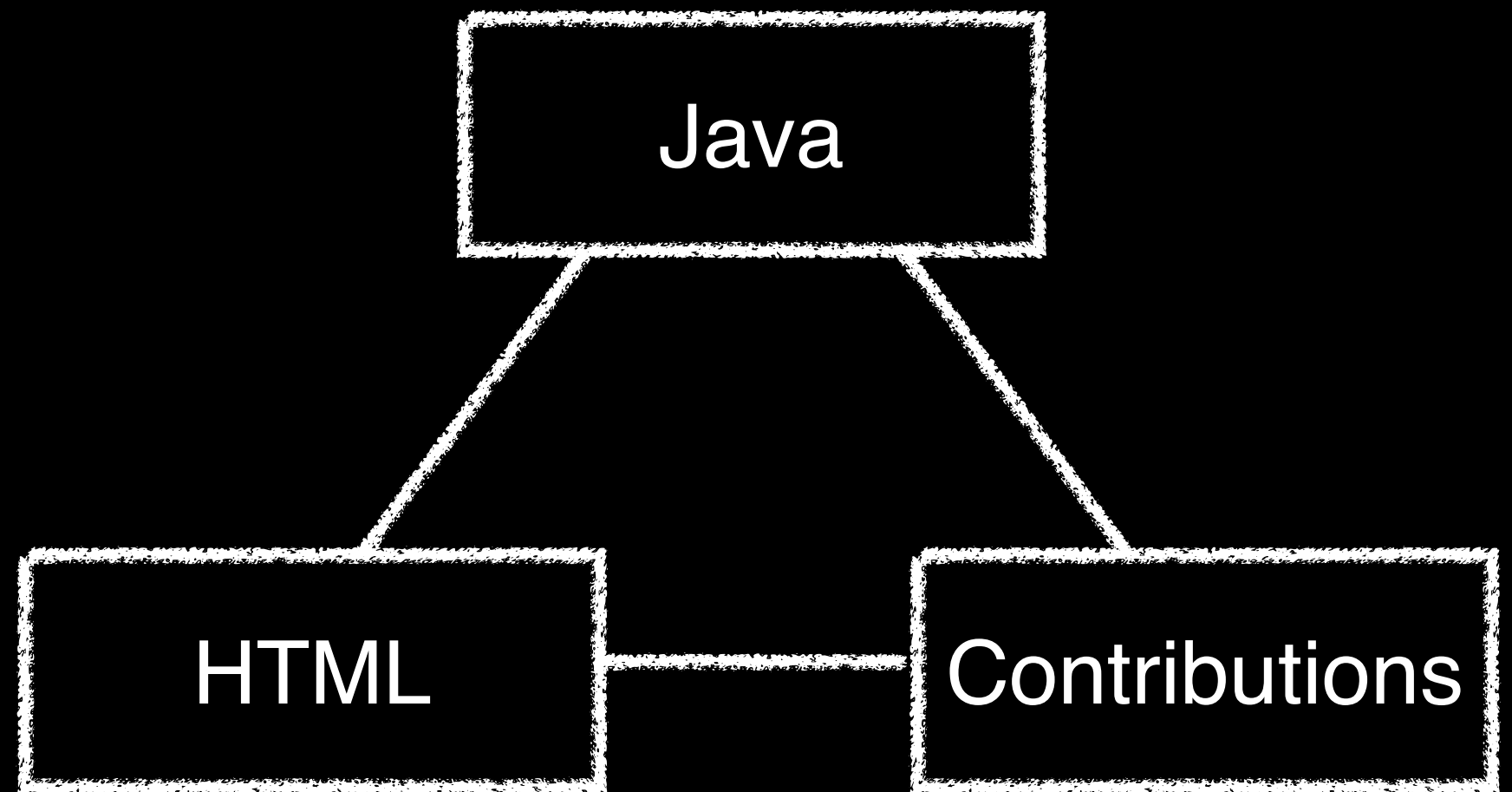
Creating
**Custom
Components**

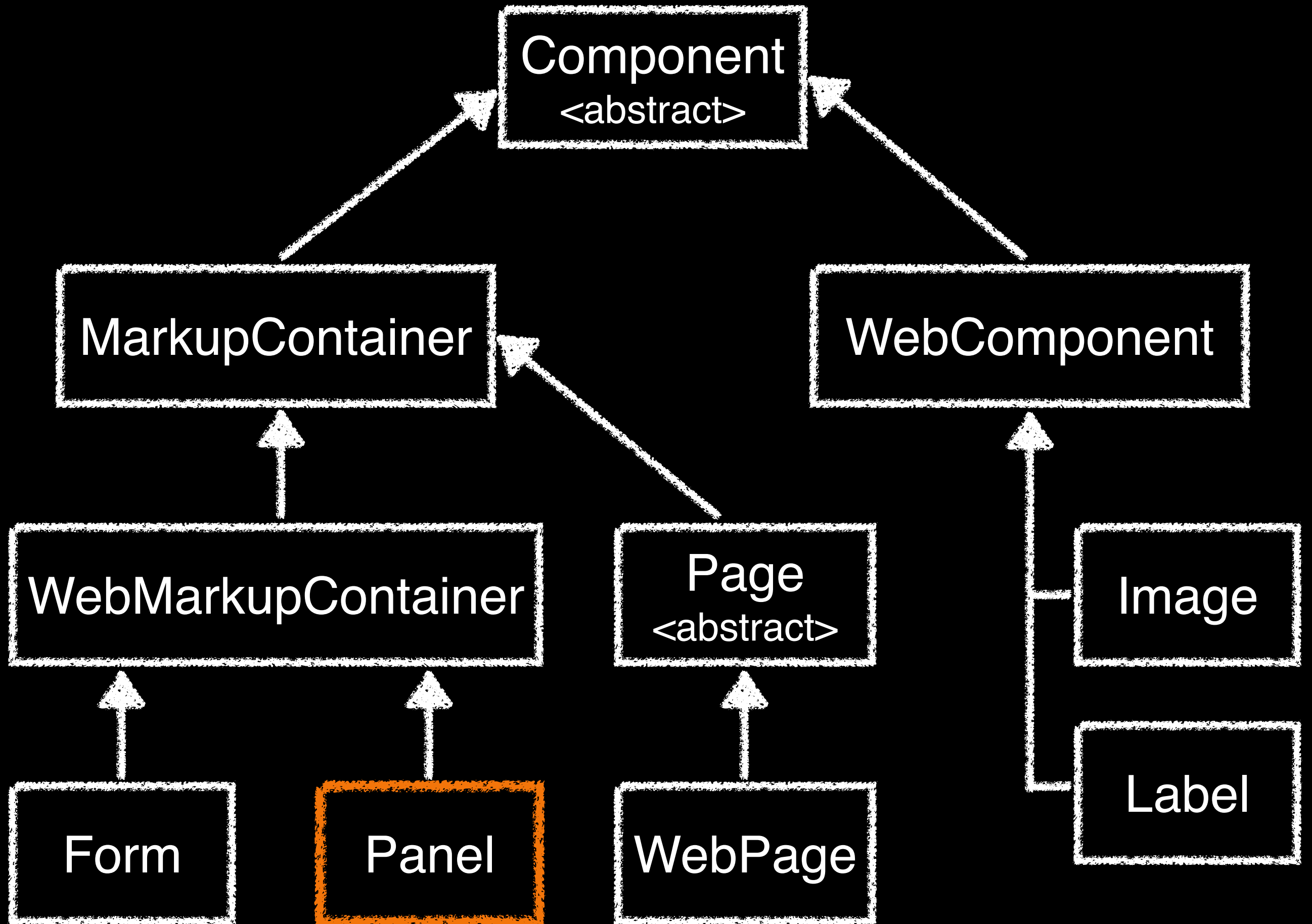
Object Orientation



is
KING

Components



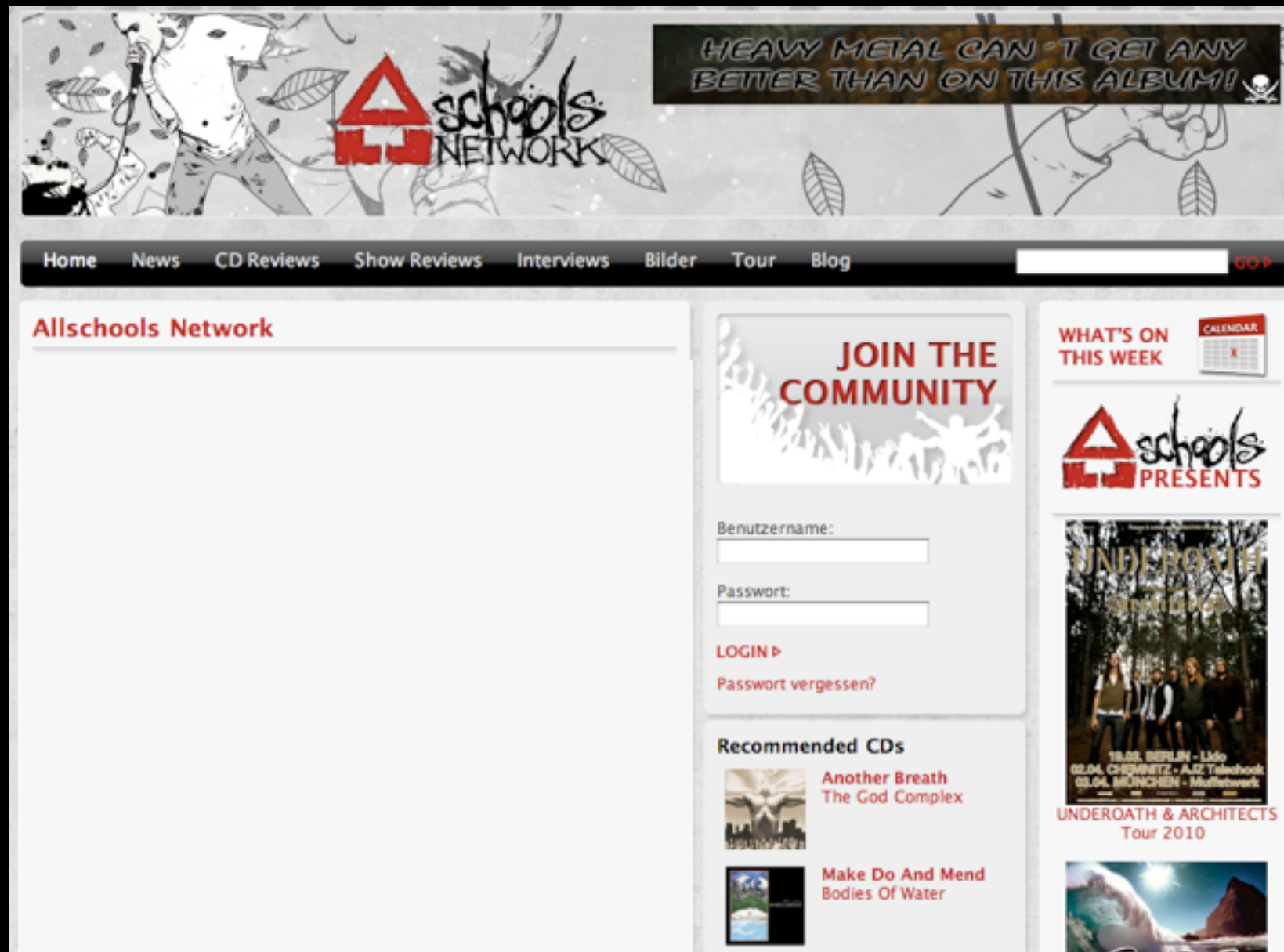


66

```
public class LabelDropDownChoice extends Panel {  
    public LabelDropDownChoice(String id) {  
        super(id);  
        CompoundPropertyModel labels =  
            new CompoundPropertyModel(new LoadableDetachableModel() {  
                @Override  
                protected Object load() {  
                    return labelService.retrieveAll();  
                }  
            });  
        add(new DropDownChoice("label", labels, new IChoiceRenderer()  
        {  
            public Object getDisplayValue(Object object) {return ((Label)object).getName();}  
  
            public String getIdValue(Object object, int index) {  
                return Long.toString(((Label)object).getId());  
            }  
        }  
        ));  
    }  
}
```

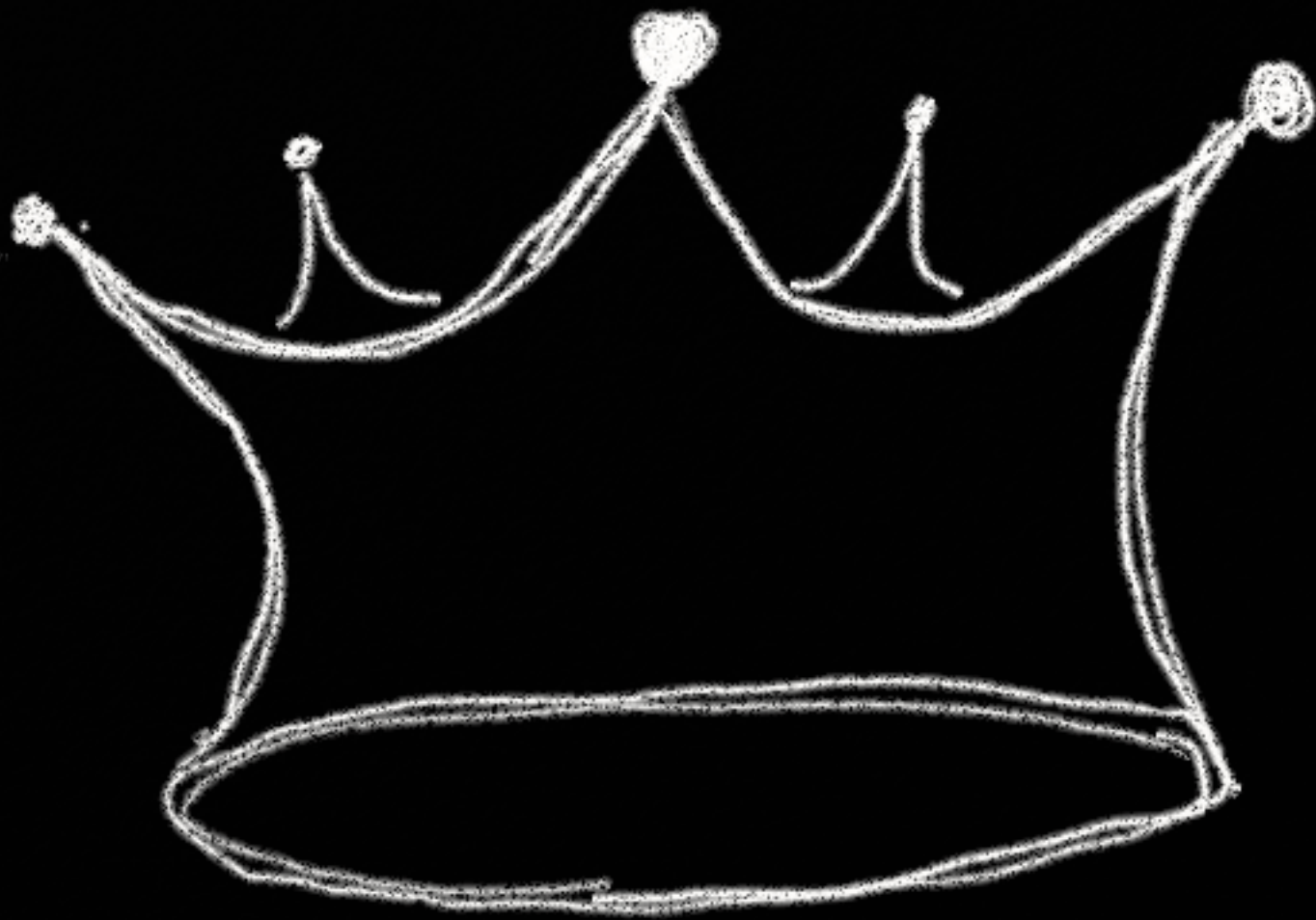
```
<wicket:panel>  
  <select wicket:id="label">  
    <option>site 1</option>  
    <option>site 2</option>  
  </select>  
</wicket:panel>
```

Consistent Layouts



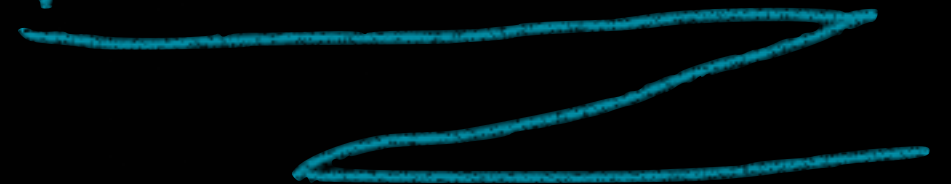
Object Orientation

is

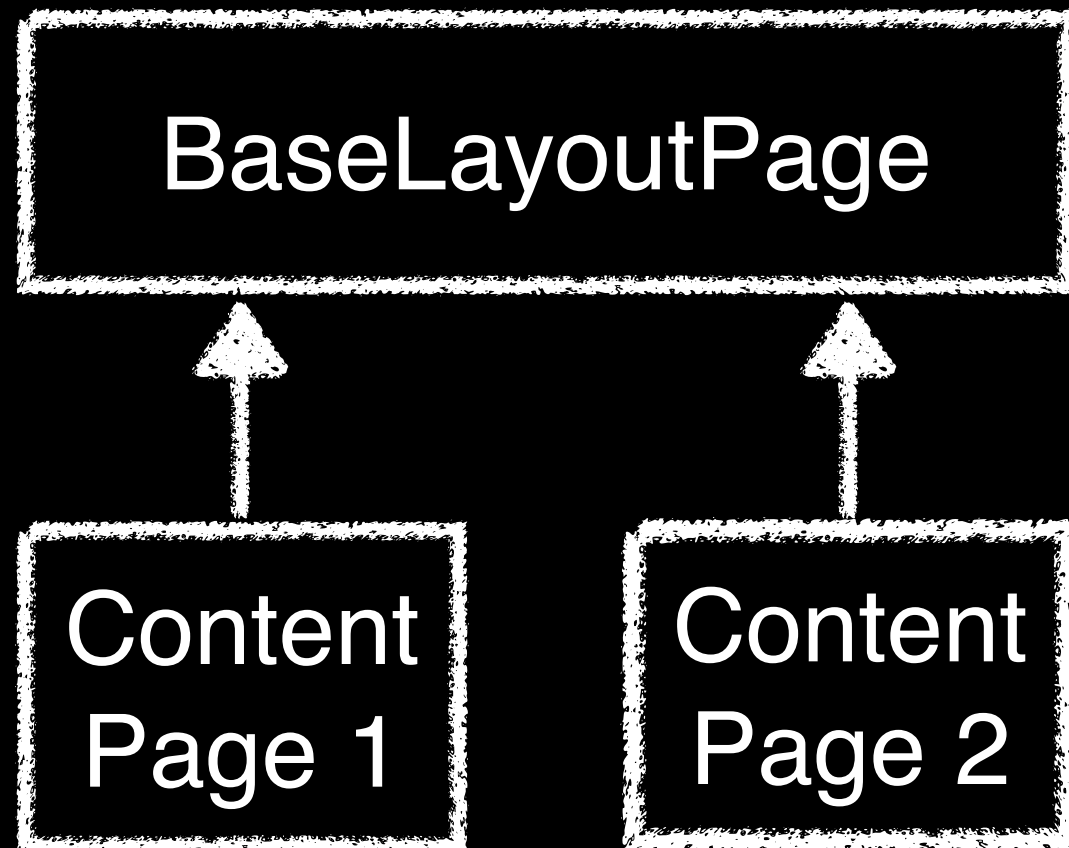


KING

AGAIN

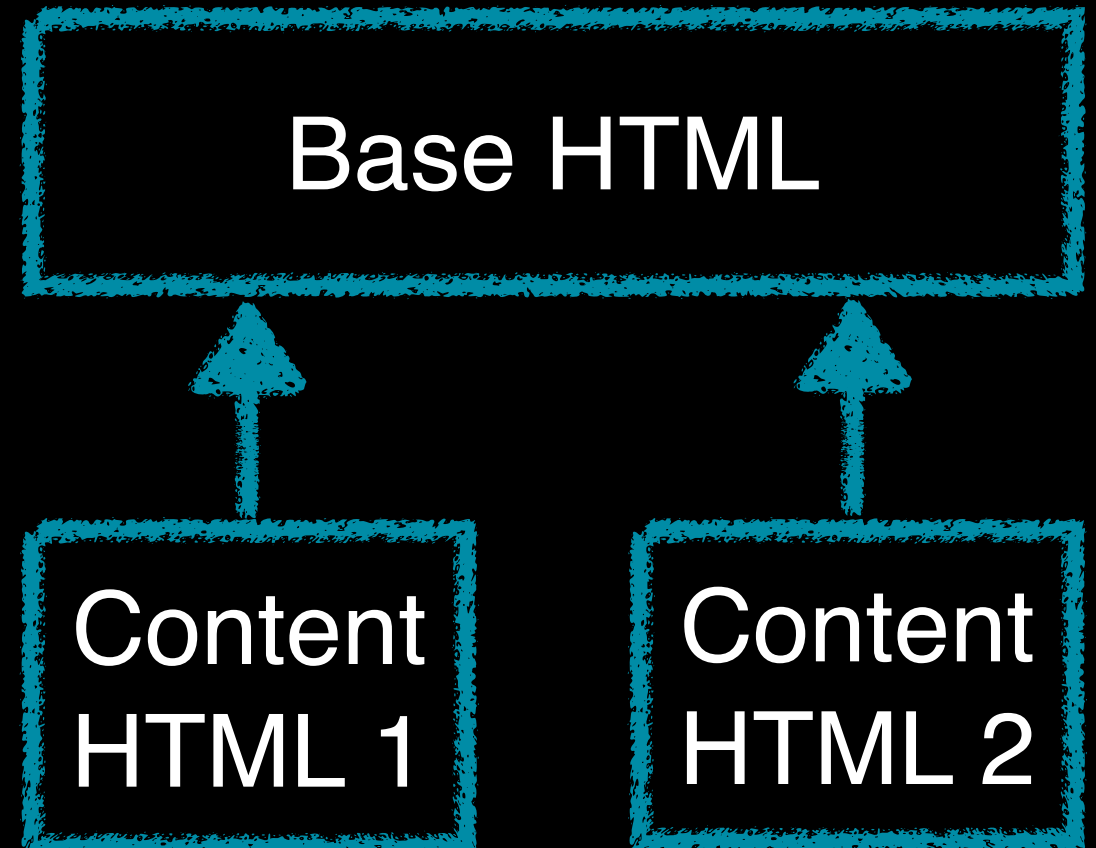


```
public abstract class
BaseLayoutPage extends
WebPage {
...
}
```

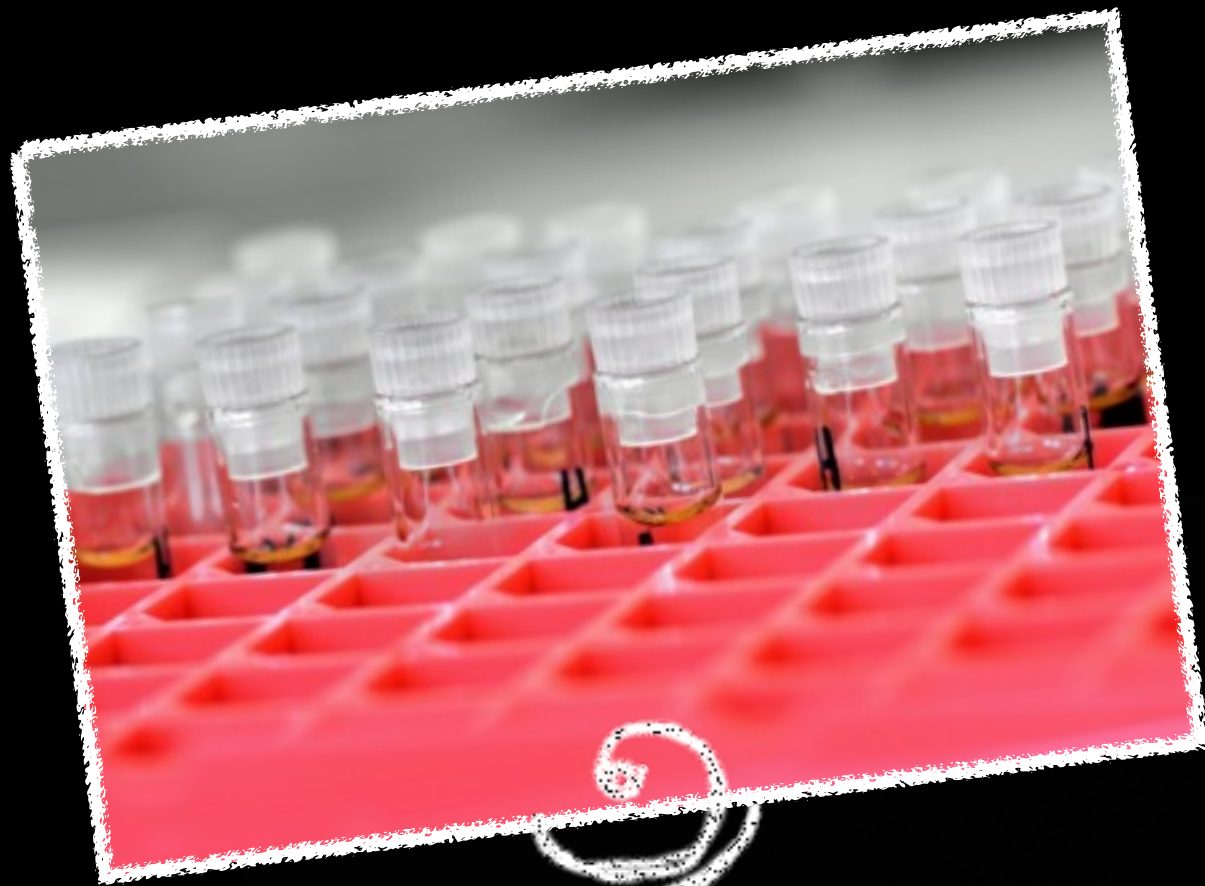


```
public class ContentPage
extends BaseLayoutPage {
...
}
```

```
<html>
  <head>...</head>
  <div id="main">
    <wicket:child/>
  </div>
</html>
```



```
<wicket:extend>
...
</wicket:extend>
```



Testing Wicket Applications

66

```
public class HomePage extends WebPage {
    public HomePage() {
        final SimpleDateFormat df = new SimpleDateFormat("hh:mm:ss");
        add(new Label("static_clock", new Model(df.format(new Date()))));
        Model<String> dynClock = new Model<String>() {
            @Override
            public String getObject() {
                return df.format(new Date());
            }
        };
        final Label label = new Label("dynamic_clock", dynClock);
        label.setOutputMarkupId(true);
        add(label);
        add(new AjaxLink("refresh") {
            @Override
            public void onClick(AjaxRequestTarget target) {
                target.addComponent(label);
            }
        });
    }
}
```

```
public class TestHomePage extends TestCase {  
    private WicketTester tester;  
    @Override  
    public void setUp() {  
        tester = new WicketTester(new WicketApplication());  
    }  
  
    public void testRenderMyPage() throws Exception {  
        tester.startPage(HomePage.class);  
        tester.assertRenderedPage(HomePage.class);  
        final SimpleDateFormat df = new SimpleDateFormat("hh:mm:ss");  
        Date before = new Date();  
        tester.assertLabel("static_clock", df.format(before));  
        tester.assertLabel("dynamic_clock", df.format(before));  
        Thread.sleep(2000);  
        tester.clickLink("refresh", true);  
        Date after = new Date();  
        tester.assertLabel("static_clock", df.format(before));  
        tester.assertLabel("dynamic_clock", df.format(after));  
    }  
}
```

There is much more

Security

Error Pages

RESTful URLs

Spring

...





Pro & Contra



Further Reading



Wicket In Action

Martijn Dashorst, Eelco Hillenius
Manning



Wicket

Roland Förther, Olaf Siefert, Carl-Eric Menzel
dpunkt Verlag

Questions?



Michael Plödd

Olaf Siefert

Senacor Technologies AG