Wicket in Action

Martijn Dashorst

ApacheCon EU 2009
Amsterdam

- Employee @ Topicus
- VP Apache Wicket
- Committer

Martijn Das

- Apache Member
- Author Wicket in Action

4 Parts

I. Getting started with Wicket

II. Ingredients for your Wicket applications

III. Preparing for the real world

IV. The future of Wicket

Part I

Getting started with Wicket

What is Wicket?

http://wicket.apache.org/

Google

Apache Wicket - Home | Manning: Wicket in Action

APACHE WICKET

Wicket

Home
Introduction
Vision
Planet Wicket
Community
News

Getting Started

Examples
Getting Wicket
Components
QuickStart
More...

Documentation

Wiki
Blogs
Reference
Books
JavaDocs

Releases

Wicket 1.4
Wicket 1.3
Wicket 1.2
Wicket 1.1
Wicket 1.0

Welcome to Apache Wicket

With proper mark-up/logic separation, a POJO data model, and a refreshing lack of XML, Apache Wicket makes developing web-apps simple and enjoyable again. Swap the boilerplate, complex debugging and brittle code for powerful, reusable components written with plain Java and HTML.

- Check the feature list
- Read some Wicket buzz, some Wicket blogs or Planet Wicket
- Find out why you should use Wicket
- Check out some selected examples in detail or see them and many more in live action
- Take a look at our live component reference
- Go and download Wicket.
- See what extras are available

Wicket is released under the Apache License, Version 2.0.

Apache Wicket 1.3.5 is released!

October 18, 2008

The Apache Wicket team is proud to announce the availability of the fifth maintenance release: Apache Wicket 1.3.5. A lot of bugs have been squashed and several improvements implemented. It is recommended you update to Wicket 1.3.5 at your earliest convenience.

Eager people click here to download the distribution, others can read further:

- Download 1.3.5 now!
- Release notes

We thank you for your patience and support.

We thank you for your patience and support.

- The Wicket Team

Apache Wicket 1.4-m3 with improved generics

What is Wicket?

Apache Wicket is a **component oriented** open source **web application** framework using just **Java** and **HTML**.

(**XML** not included)

What does that **mean**?

"Writing a Wicket app is rather **more** like writing an event-based **desktop application** than a **web application**."

- Michael Porter (LShift)

6535 - Milou van Duijvenvoorde HA4A

zoekresultaten

Leerlingkaart | Personalia | Registratie | Plaatsing | Vakken | Absentie | Resultaten | Examen | BRON | Begeleiding

Resultaten - voortgangsdossier

1F | 2F | HA3B | HA4A | Examendossier

Opleiding HAVO-3

Peildatum (schooljaar) 31-07-2007 (2006/2007)

Sorteren op Links uitlijnen

Toon historische vakkeuzes (0)

Status: OK

	1	Periode								2	3	4
Vak	R							P	r	R	R	R
aardrijkskunde	5.7	7.3	5.4					6	-	6.2	5.7	7.6
culturele en kunstzinnige vorming								-	-			6.3
Duits	5.6	5.6	8.0	5.0	4.8			6	-	5.5	5.5	5.9
economie	6.8	5.7	4.3					5	-	5.0	7.2	7.1
Engels	7.0	5.5	7.4	5.6	7.2			6	-	6.3	7.9	6.2
Frans	6.3	6.4	6.3	5.9				6	-	6.3	3.1	5.1
geschiedenis	6.9	6.0	6.3	7.8	6.8	8.0		7	-	7.1	6.4	6.4
handvaardigheid I		8.6	5.0					6	-			
informatica		5.4						5	-	5.4		3.6
lichamelijke oefening	7.0	7.0						7	-	7.0	7.0	7.0
mentorles								-	-			
muziek								-	-			7.9
natuurkunde	5.2	7.1						7	-	7.1	5.3	6.4
Nederlands	7.8		8.1	1.0	5.6			6	-	6.0	5.6	5.6
scheikunde	6.1	8.0	6.4					7	-	6.8	7.4	6.4

PDF | Exporteren | Overgangsadvies | Importeer resultaten

Top 5 features

1. Just Java and HTML™
2. No XML configuration
3. Object Oriented Programming for the Web
4. Easy custom component creation
5. Community

Just Java™

- Components are objects (just use **extends**)
- No annotation hunting
- Not managed by the framework (just use **new**)

Just Java™

```
public class EditPersonLink extends Link {
    private final Person person;
    public EditPersonLink(String id, Person person) {
        super(id);
        this.person = person;
    }
    public void onClick() {
        setResponsePage(new EditPersonPage(person));
    }
}
```

Just Java™

```
public class EditPersonLink extends Link {
    private final Person person;
    public EditPersonLink(String id, Person person) {
        super(id);
        this.person = person;
    }
    public void onClick() {
        setResponsePage(new EditPersonPage(person));
    }
}
```

Just Java™

```
public class EditPersonLink extends Link {
    private final Person person;
    public EditPersonLink(String id, Person person) {
        super(id);
        this.person = person;
    }
    public void onClick() {
        setResponsePage(new EditPersonPage(person));
    }
}
```

Just HTML™

- Designer friendly
- No new language to learn
- No serverside scripting in markup
- Developers won't $#$#% up your design
- Designers won't $*&#$# up your code

Just HTML™

```
<table>
  <tr>
    <c:forEach var="item" items="${sessionScope.list}">
      <td>
        <c:out value="${item.name}" />
      </td>
    </c:forEach>
  </tr>
</table>
```

Just HTML™

```
<h:dataTable value="#{list}" var="item">
  <h:column>
    <h:outputText va      "#{ite  name}"/>
  </h:column>
</h:dataTable>
```

Just HTML™

```
<table>
  <tr wicket:id="list">
    <td wicket:id="firstname">John</td>
    <td wicket:id="lastname">Doe</td>
  </tr>
</table>
```

designers  developers


XML?

No XML

- **Page navigation is done in Java:**

```
setResponsePage(new MyPage());
```

- **Configuration is done in Java:**

```
protected void init() {
    getDebugSettings()
        .setAjaxDebugModeEnabled(true);
}
```

Object Oriented Programming **for the web**

“Wicket is the first truly modern framework [...] to **fully realize** the promise of **object oriented development**.”

—James McLaughlin

Embrace encapsulation

```
public class PersonLink extends Link {
    private Person person;
    public PersonLink(String id, Person p) {
        this.person = p;
    }
    public void onClick() {
        person.someOperation();
    }
}
```

Embrace encapsulation

```
public class PersonLink extends Link {
    private Person person;
    public PersonLink(String id, Person p) {
        this.person = p;
    }
    public void onClick() {
        person.someOperation();
    }
}
```

Use the right abstractions

```
public abstract class PersonEditPanel extends Panel {
    public PersonEditPanel(String id, Person p) {
        add(new Button("save") {
            public void onSubmit() {
                onSave(person);
            }
        });
        add(new Button("delete") { ... });
    }
    protected abstract void onSave(Person p);
    protected abstract void onDelete(Person p);
}
```

Use the right abstractions

```
public abstract class PersonEditPanel extends Panel {
    public PersonEditPanel(String id, Person p) {
        add(new Button("save") {
            public void onSubmit() {
                onSave(person);
            }
        });
        add(new Button("delete") { ... });
    }
    protected abstract void onSave(Person p);
    protected abstract void onDelete(Person p);
}
```

Easy custom component creation

public class PersonLink **extends** Link { ... }

just use **extends**

public class PersonEditPanel **extends** Panel { ... }

public class NationalitySelect **extends**
DropDownChoice { ... }

Community

> “The Wicket community is by far the **most vibrant** and **helpful** open source **community** that I have been a part of.”
>
> —Jeremy Thomerson

Community

- Part of The Apache Software Foundation

"**not** simply a group of projects sharing a server, **but** rather a **community** of developers and users."

Community

- Part of The Apache Software Foundation
- User list: >800 subscribers, 59 messages/day
- Meetups around the world
- ##wicket on irc.freenode.net

Community Projects

- **Wicket Stuff**
  jquery, dojo, scriptaculous, google maps, hibernate, minis, prototip, etc.
- databinder.net
- Wicket Web Beans
- Tons of other projects at googlecode, sourceforge, etc.

http://cafepress.com/apachewicket

Some Examples

Hello, World!

Hello, World!

<h1>Hello, World!</h1>

Hello, World!

<h1>[replaced text]</h1>

Hello, World!

<h1 **wicket:id="msg"**>[replaced text]</h1>

Hello, World!

```
<h1 wicket:id="msg">[replaced text]</h1>
```

+

```
add(new Label("msg", "Hello, World!"))
```

Hello, World!

<h1 wicket:id="msg">[replaced text]</h1>

+

add(new Label("msg", "Hello, World!"))

=

<h1>Hello, World!</h1>

Hello, World!

```
<html>
    <head>
        <title>Home page</title>
    </head>
    <body>
        <h1 wicket:id="msg">Gets replaced</h1>
    </body>
</html>
```

Hello, World!

```
package com.example;

import org.apache.wicket.markup.html.WebPage;
import org.apache.wicket.markup.html.basic.Label;

public class HomePage extends WebPage {
    public HomePage() {
        add(new Label("msg", "Hello, World!"));
    }
}
```

Click counter

Click counter

This link has been clicked 123 times.

Click counter

This link has been clicked 123 times.

This ****link**** has been clicked 123 times.

Click counter

This link has been clicked 123 times.

This link has been clicked ****123**** times.

Click counter

This link has been clicked 123 times.

```
This <a wicket:id="link" href="#">link</a>
has been clicked <span>123</span> times.
```

Click counter

This link has been clicked 123 times.

```
This <a wicket:id="link" href="#">link</a> has
been clicked <span wicket:id="clicks">123</
span> times.
```

Click counter

```
public class ClickCounter extends WebPage {
}
```

Click counter

```
public class ClickCounter extends WebPage {
    public ClickCounter() {
    }
}
```

Click counter

```
public class ClickCounter extends WebPage {
    private int clicks = 0;
    public ClickCounter() {
        }
    }
```

Click counter

```
public class ClickCounter extends WebPage {
    private int clicks = 0;
    public ClickCounter() {
        add(new Link("link") {
            protected void onClick() {
                clicks++;
            }
        });
    }
}
```

Click counter

```
public class ClickCounter extends WebPage {
    private int clicks = 0;
    public ClickCounter() {
        add(new Link("link") { ... });
        add(new Label("clicks",
            new PropertyModel(this, "clicks")));
    }
}
```

Ajax Click counter

```
public class ClickCounter extends WebPage {
    private int clicks = 0;
    public ClickCounter() {
        add(new Link("link") { ... });
        add(new Label("clicks",
                      new PropertyModel(this, "clicks")));
    }
}
```

Ajax Click counter

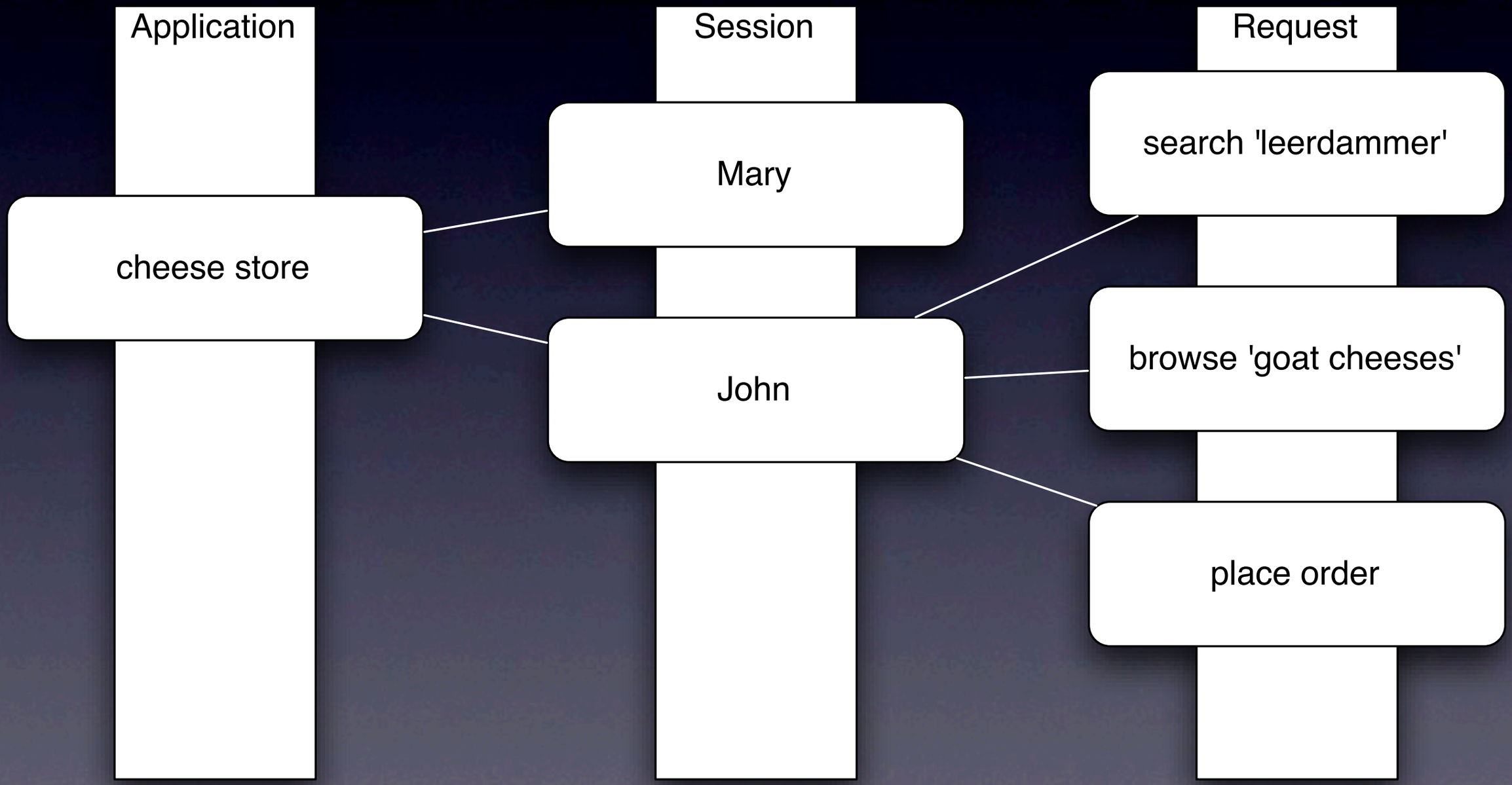
```
public ClickCounter() {
    final Label label = new Label(...);
    add(label);
    label.setOutputMarkupId(true)
    add(new Link("link") {
        public void onClick() {
            clicks++;
        }
    });
}
```

Ajax Click counter

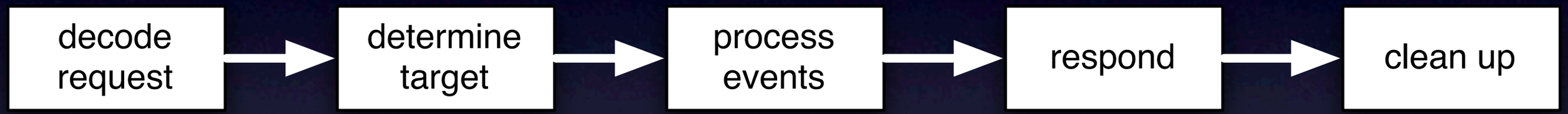
```
public ClickCounter() {
    final Label label = new Label(...);
    add(label);
    label.setOutputMarkupId(true)
    add(new AjaxLink("link") {
        public void onClick(AjaxRequestTarget t) {
            clicks++;
            t.addComponent(label);
        }
    });
}
```

The architecture of Wicket

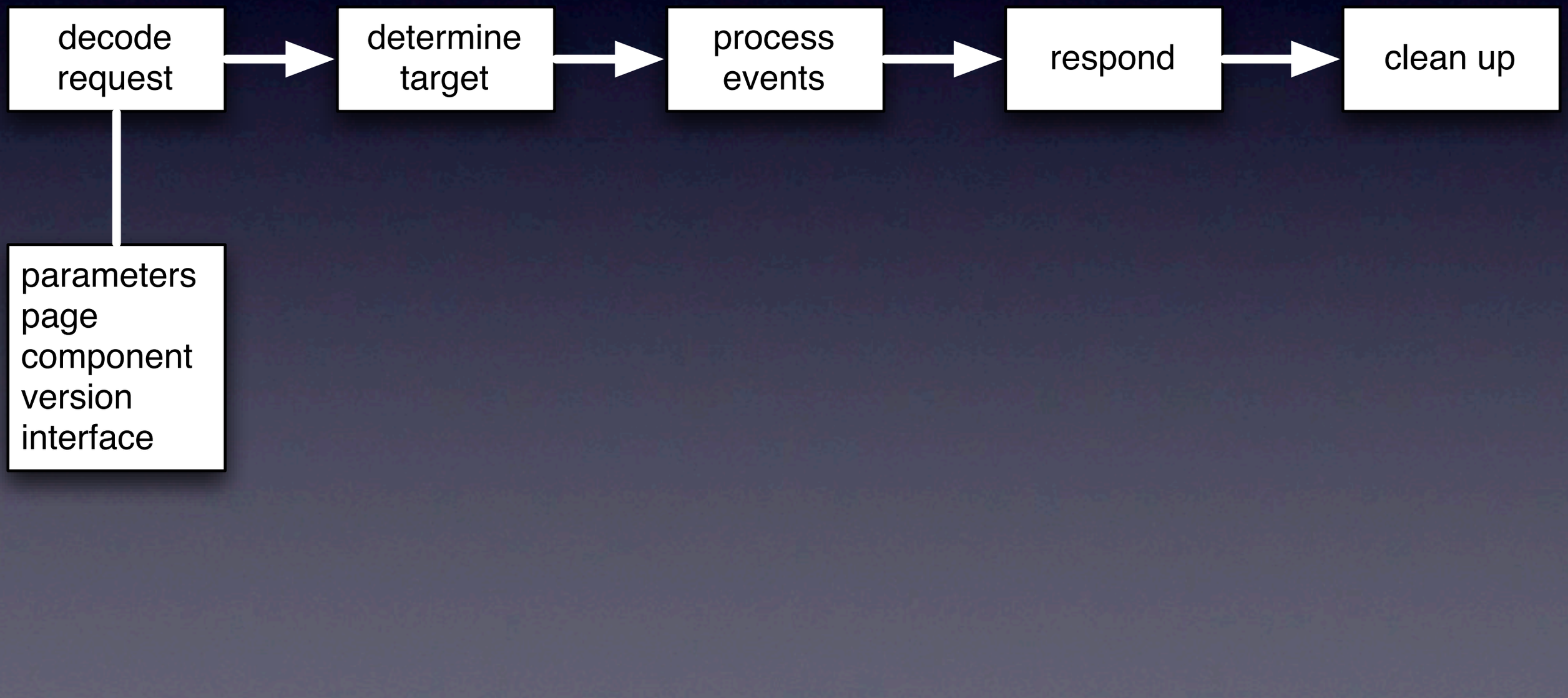
Handling requests



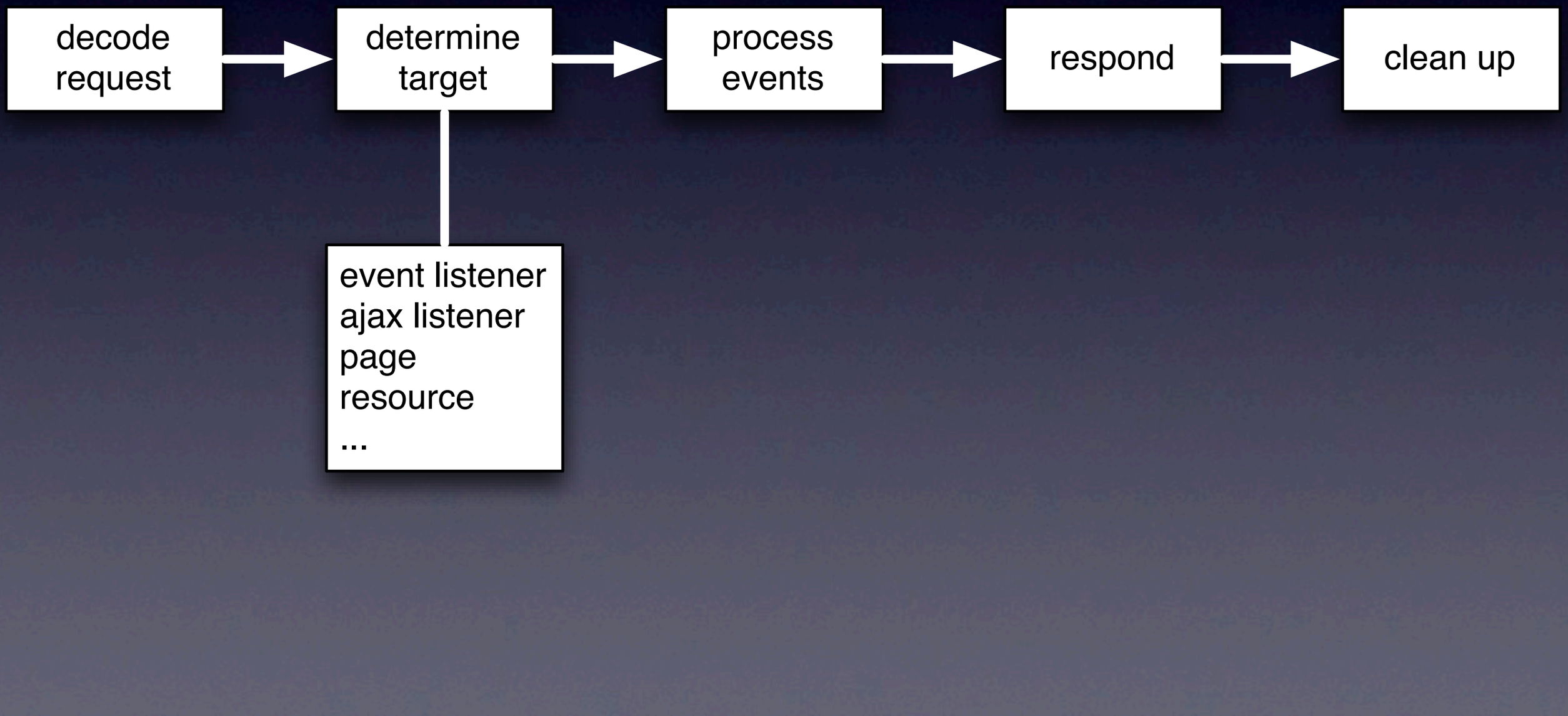
Handling requests



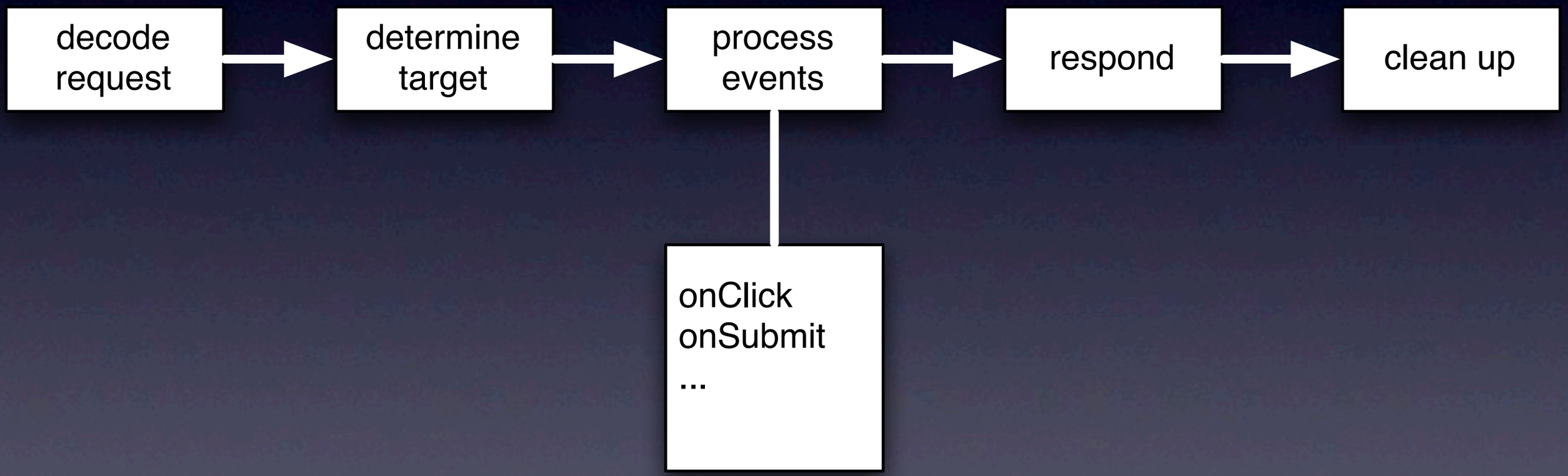
Handling requests



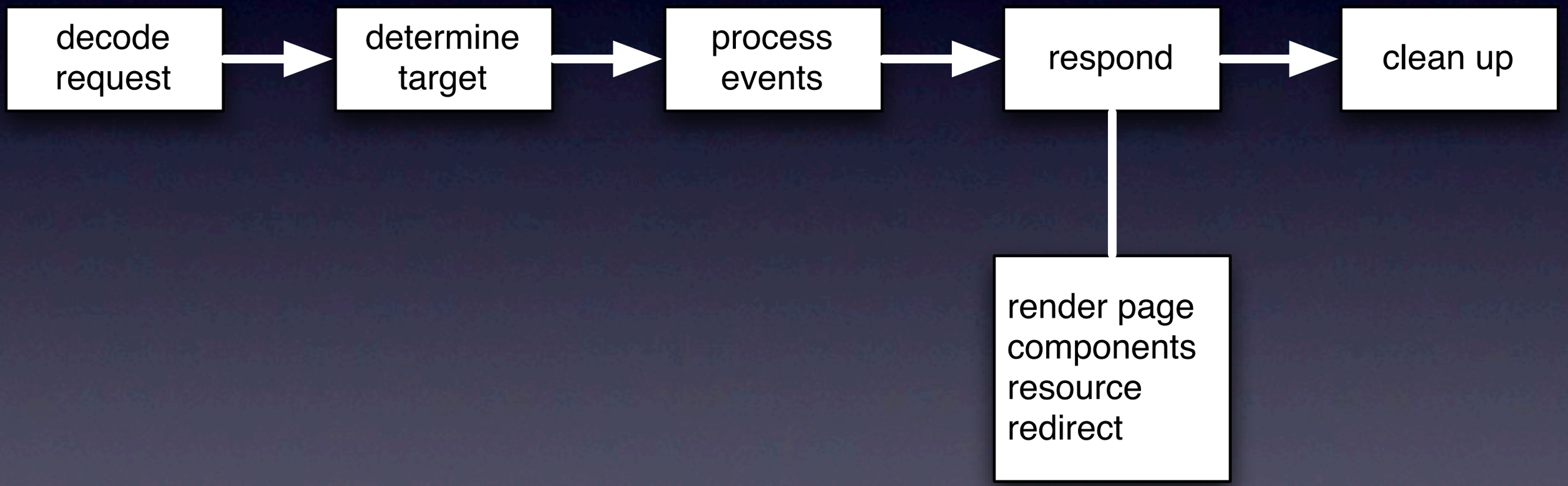
Handling requests



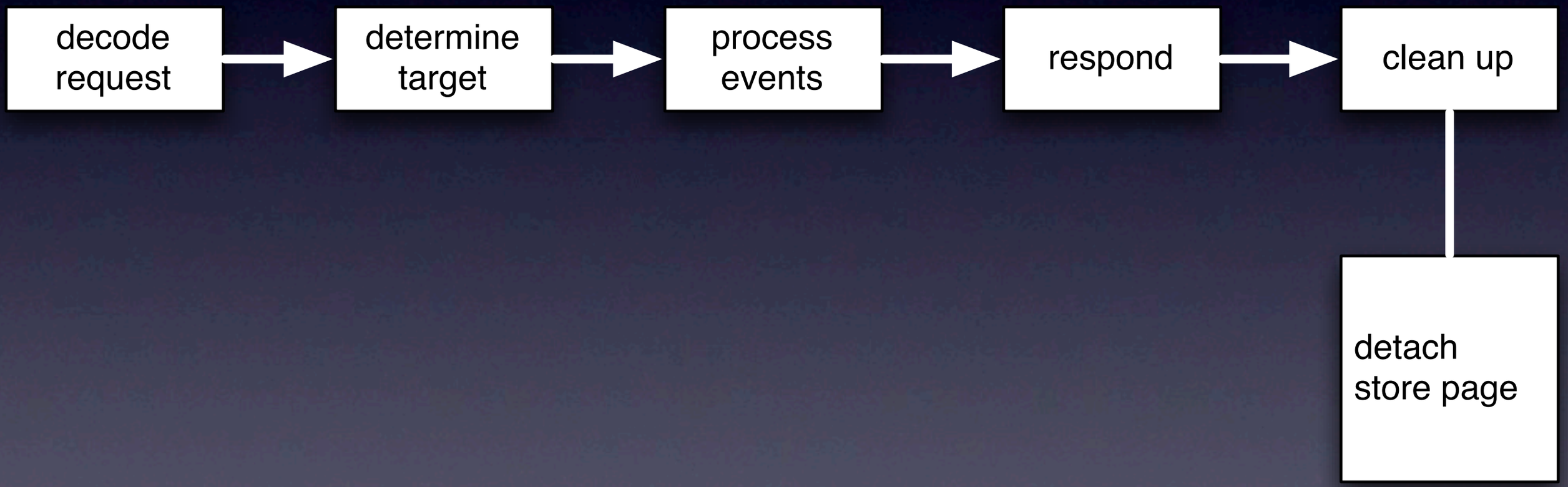
Handling requests



Handling requests



Handling requests



Handling requests

- Access to page is single threaded
- No multithreading issues in UI code
- Session is **not** single threaded

Part II

Ingredients for your Wicket application

4 ingredients

- Components
- Behaviors
- Models
- Page Composition

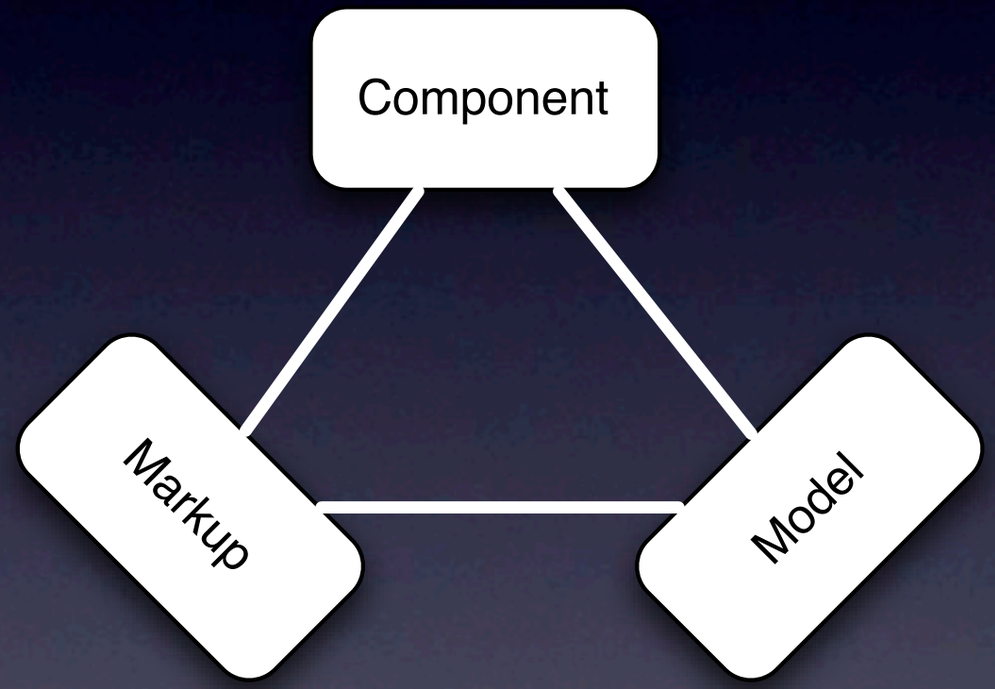
Wicket Components

Components

- Label
- Link
- AjaxLink
- ListView
- Form
- TextField
- DropDownChoice
- Tree
- DataTable
- DatePicker
- CustomerLink
- NationalityDropDownChoice
- ShoppingCartPanel

Component

- Can handle events (onClick, onSubmit, ...)
- Writes its markup to the response
- May or may not have a Model



Component

- Component has identifier (**wicket:id**)
  new Label(**"msg"**, "Hello, World!");

- Markup has same identifier (**wicket:id**)
  <h1 **wicket:id="msg"**>Gets replaced</h1>

Component

- Components can be nested inside one another

```
<a href="#" wicket:id="link">
    <span wicket:id="label">replaced</span>
</a>

Link link = new Link("link") {...};
add(link);
link.add(new Label("label", "Some text"));
```

Component

- Components can be nested inside one another

```
<a href="#" wicket:id="link">
    <span wicket:id="label">replaced</span>
</a>

Link link = new Link("link") {...};
add(link);
link.add(new Label("label", "Some text"));
```

Component

- Components can be nested inside one another

```
<a href="#" wicket:id="link">
    <span wicket:id="label">replaced</span>
</a>

Link link = new Link("link") {...};
add(link);
link.add(new Label("label", "Some text"));
```

Component

- Components can receive events

```
public class EditPersonForm extends Form {
    public EditPersonForm(String id, Person p) {
        super(id, new Model(p));
    }
    public void onSubmit() {
        person.save();
        setResponsePage(new ListPage());
    }
}
```

component: bookmarkable page link

- Links to a page with an URL that is bookmarkable
- Emulate this with setResponsePage(Page.class)
- Use PageParameters to construct URL arguments
- Restful URL's with mounting

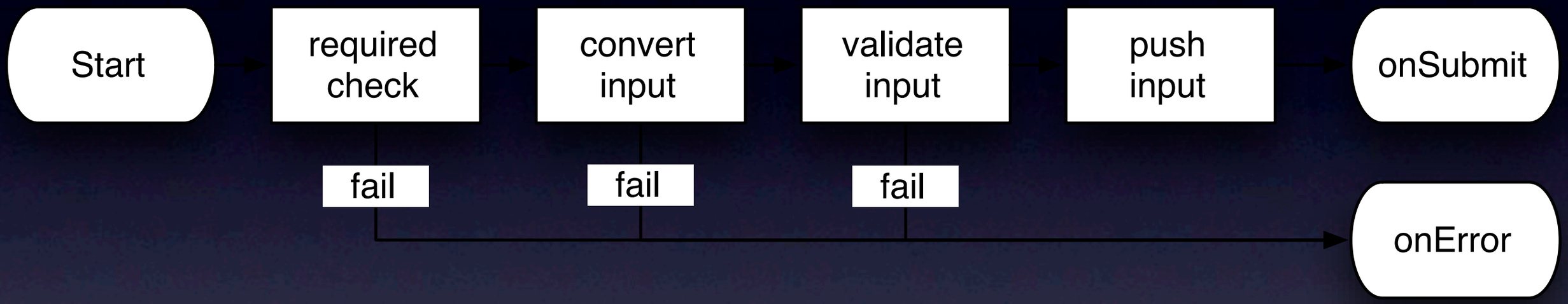
component: repeaters

- RepeatingView, RefreshingView, ListView, PropertyListView
- Used to repeat markup

```
<tr wicket:id=”cheeses”>
    <td wicket:id=”name”></td><td
wicket:id=”price”></td>
</tr>
new PropertyListView(“cheeses”, cheeses) {
    protected void onPopulateItem(ListItem item) {
        item.add(new Label(“name”));
    }
}
```

component: form

- Encapsulates HTML <form>
- May be nested (as opposed to HTML)
- Has onSubmit and onError handlers


Start
required check
convert input
validate input
push input
onSubmit
fail
fail
fail
onError

component: textfield

- Encapsulates <input type=”text”>
- setRequired(true)
- setPersistent(true)
- add(new PatternValidator(“\d\d\d\d[a-z][a-z]”));

component: Page

- Use WebPage as your base class
- Has own markup file (next to Java file)
- Can use markup inheritance with <wicket:child /> and <wicket:extend></wicket:extend>
- Can have different markup using localization, styles and variations

component: panel

- Base class for reusable components
- Has own markup file next to Java file <wicket:panel>....</wicket:panel>
- Works with i18n and i10n, styles and variations
- Can contribute to head section with <wicket:head></wicket:head>

Behaviors

Behaviors are **decorators**
for components

Behaviors

- Manipulate component tags
- Add JavaScript to components
- Add Ajax behavior to components

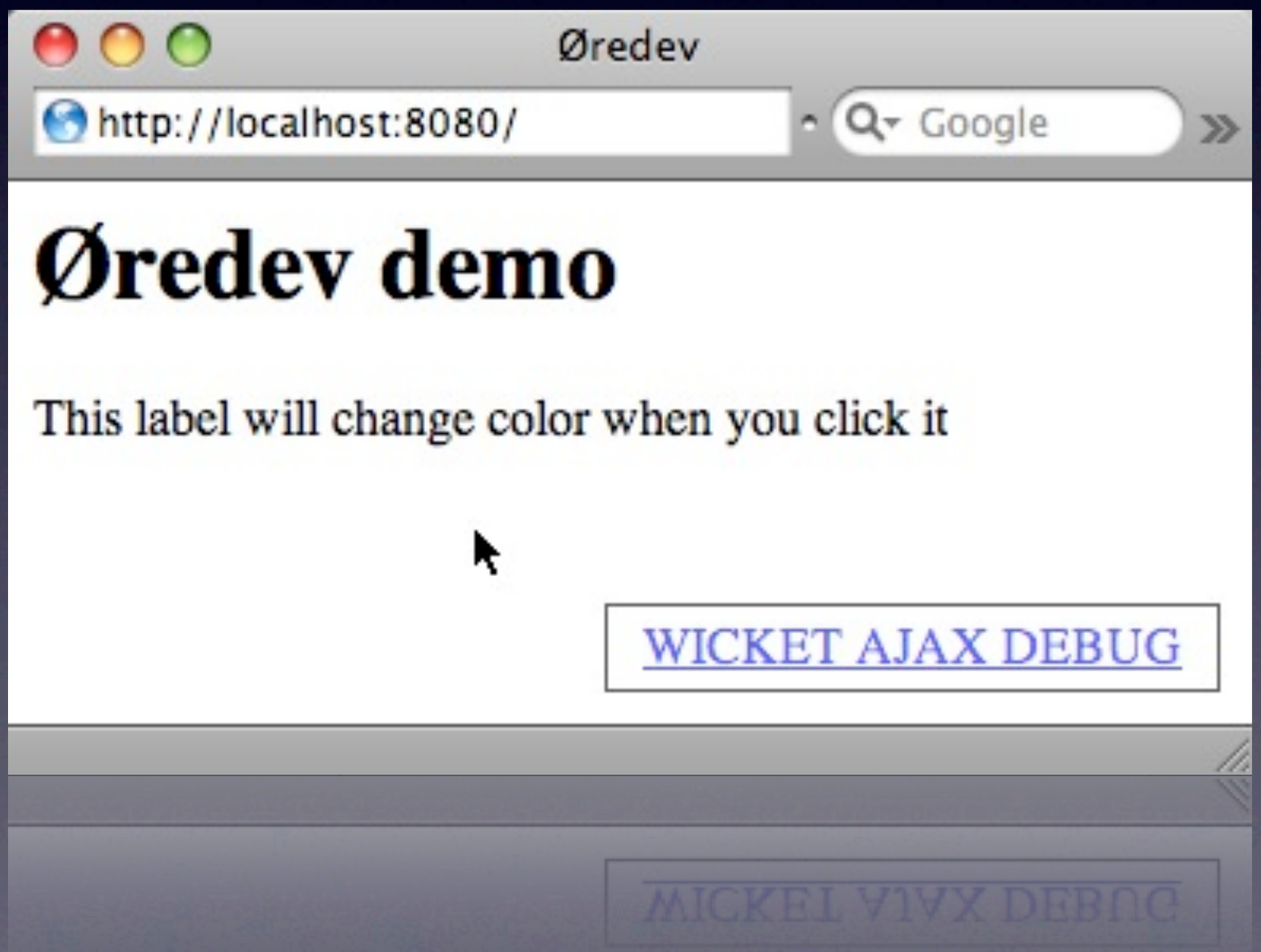
Adding JavaScript to a component

```
link.add(new AbstractBehavior() {
    public void onComponentTag(Component component,
                               ComponentTag tag) {
        tag.put("onclick",
                "return confirm('Are you sure?');");
    }
});
```

```
<a href="..." onclick="return confirm('Are you sure?');">
    Play thermonuclear war
</a>
```

Ajax behaviors

Create a Label that changes **color with a click** using Ajax



```
Label label = new Label("label", "Click me");
```

```
Label label = new Label("label", "Click me");
label.setOutputMarkupId(true);
```

```
Label label = new Label("label", "Click me");
label.setOutputMarkupId(true);
label.add(new AjaxEventBehavior("onclick") {
});
```

```
Label label = new Label("label", "Click me");
label.setOutputMarkupId(true);
label.add(new AjaxEventBehavior("onclick") {
    private boolean isRed = false;

    protected void onComponentTag(...) {
        super.onComponentTag(component, tag);
        if(isRed) { tag.put("style", "color:red"); }
    }
});
```

```
Label label = new Label("label", "Click me");
label.setOutputMarkupId(true);
label.add(new AjaxEventBehavior("onclick") {
    private boolean isRed = false;
    protected void onEvent(AjaxRequestTarget target) {
        isRed = !isRed;

    }
    protected void onComponentTag(...) {
        super.onComponentTag(component, tag);
        if(isRed) { tag.put("style", "color:red"); }
    }
});
```

```
Label label = new Label("label", "Click me");
label.setOutputMarkupId(true);
label.add(new AjaxEventBehavior("onclick") {
    private boolean isRed = false;
    protected void onEvent(AjaxRequestTarget target) {
        isRed = !isRed;
        target.addComponent(getComponent());
    }
    protected void onComponentTag(...) {
        super.onComponentTag(component, tag);
        if(isRed) { tag.put("style", "color:red"); }
    }
});
```

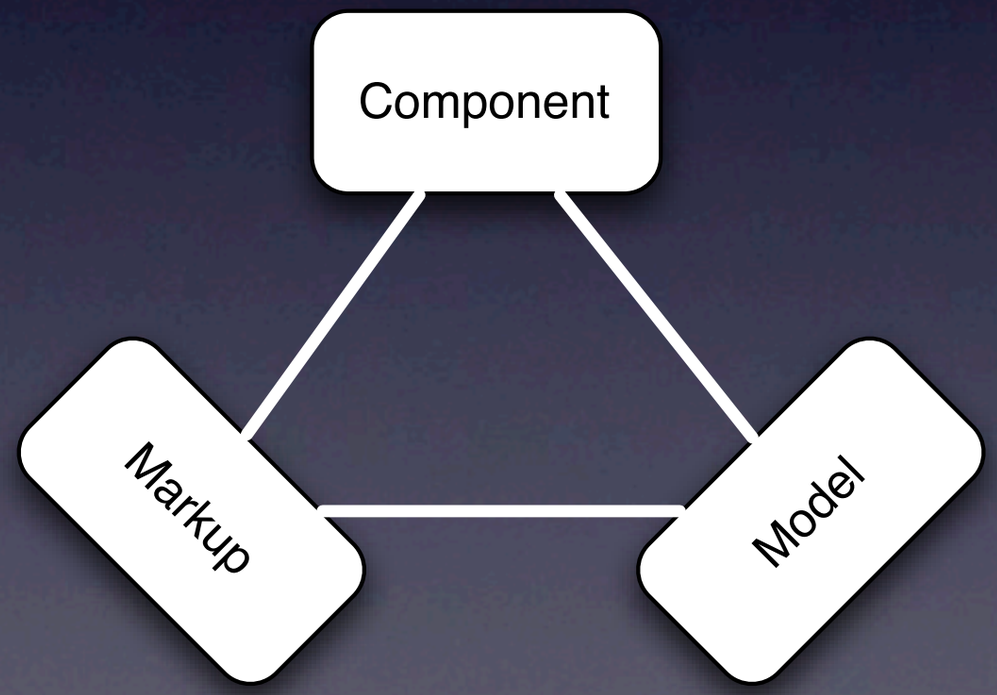
Ajax behaviors

- **Can re-render components**
  target.addComponent(...);
- **Can execute JavaScript**
  target.appendJavaScript("$('foo').explode();");
- **Can throttle the client side events**
  behavior.setThrottleDelay(Duration.seconds(10));
- **Can add new header related resources**
  JavaScript, stylesheets, etc.

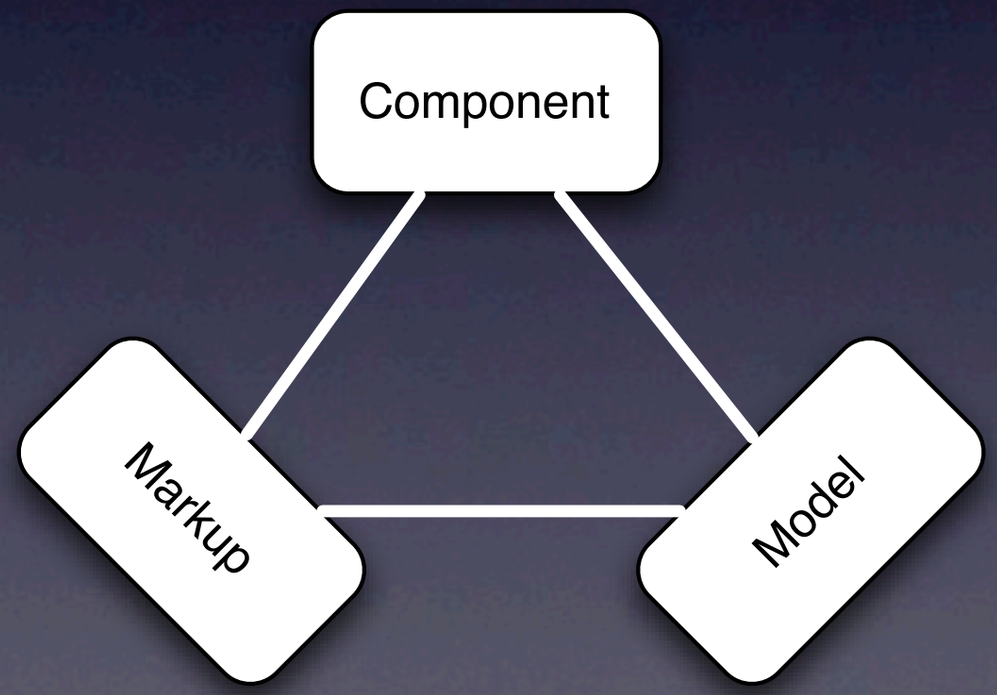
Ajax Components

- Links, Forms, Buttons, CheckBoxes
- Ajax fallback components
- Indicating components
- Tree, DataTable, PagingNavigator

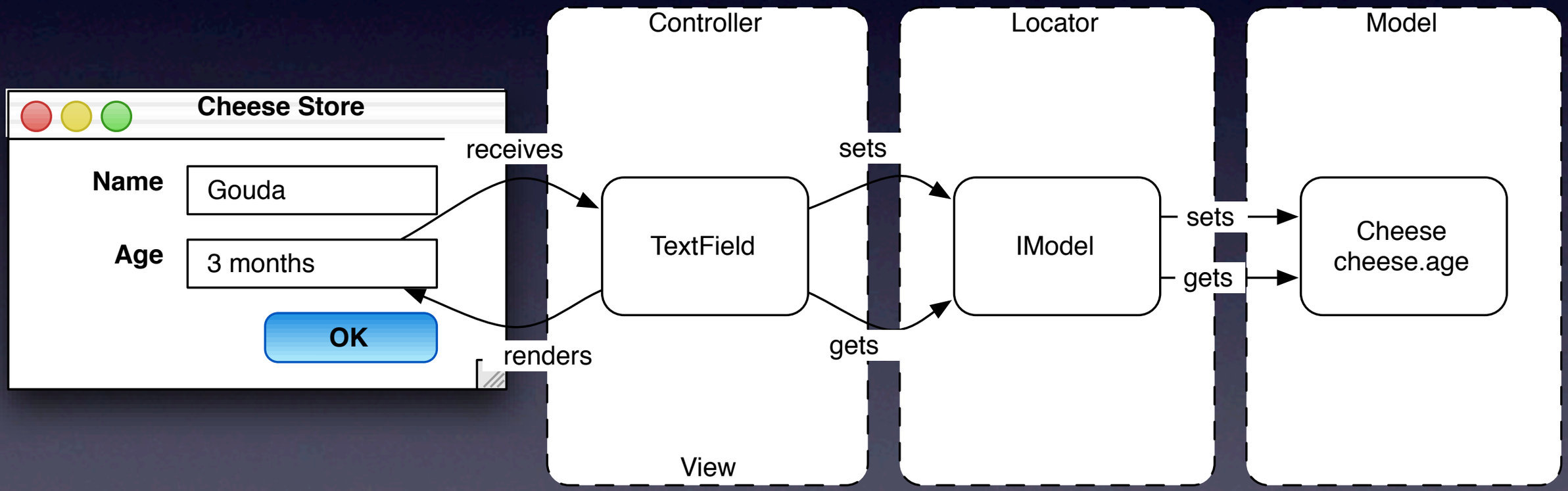
Understanding models



Wicket **models** allow components to **retrieve** and **store data**.



Models


Controller
Locator
Model
Cheese Store
Name
Gouda
Age
3 months
OK
receives
sets
TextField
IModel
sets
gets
Cheese
cheese.age
renders
gets
View

Simple Model usage

```
new Label("name", customer.getName())

new Label("street",
          customer.getAddress().getStreet())

new TextField("username", new Model("<username>"))
```

What's wrong with this?

```
new Label("street",
                customer.getAddress().getStreet())
```

- Label isn't notified of street/address/ customer changes
- NullPointerException: customer, address could be null

PropertyModel usage

```
new Label("street",
        new PropertyModel(customer, "address.street"))
```

```
setModel(new CompoundPropertyModel(customer));
add(new Label("address.street"));
add(new Label("address.zipcode"));
add(new Label("address.city"));
```

PropertyModel

+ null safe
+ dynamic
— not rename safe

Page composition:

creating a **consistent layout**

YUI Base Page

file:///Workspaces/apachecon/yahoo/index.html

Google

Header

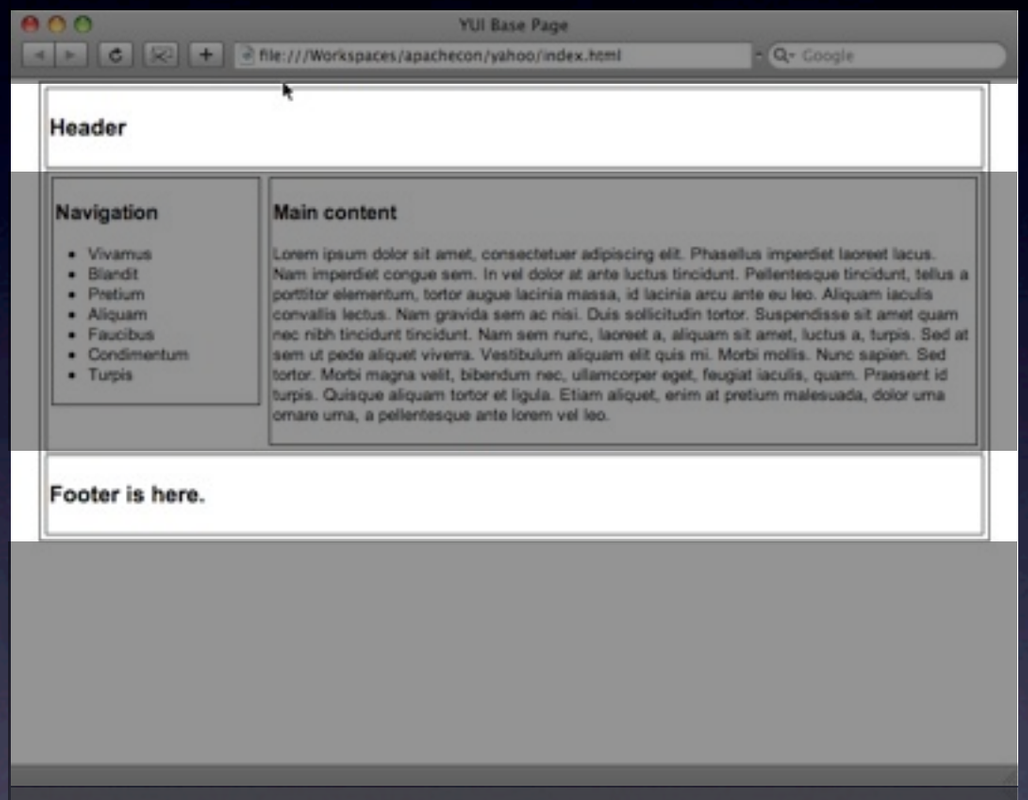
Navigation

- Vivamus
- Blandit
- Pretium
- Aliquam
- Faucibus
- Condimentum
- Turpis

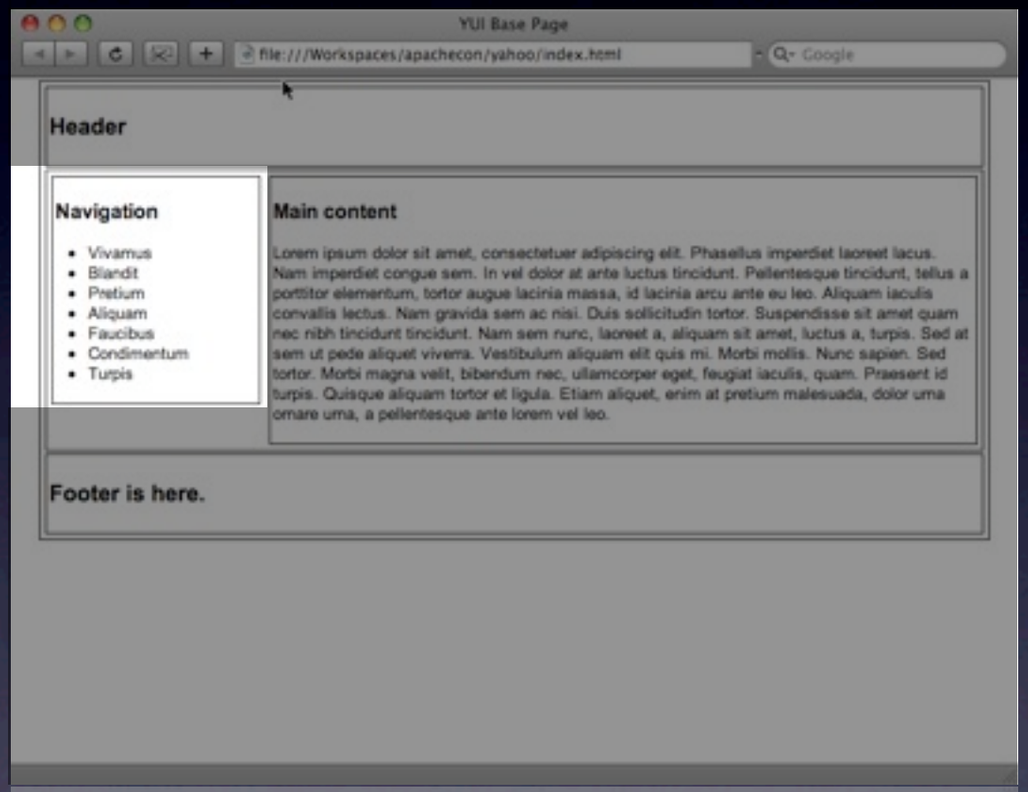
Main content

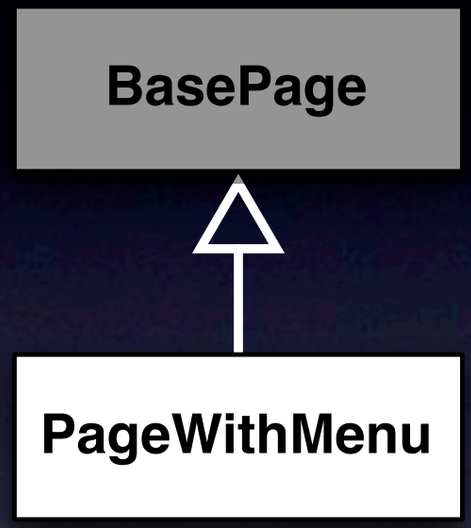
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus imperdiet laoreet lacus. Nam imperdiet congue sem. In vel dolor at ante luctus tincidunt. Pellentesque tincidunt, tellus a porttitor elementum, tortor augue lacinia massa, id lacinia arcu ante eu leo. Aliquam iaculis convallis lectus. Nam gravida sem ac nisi. Duis sollicitudin tortor. Suspendisse sit amet quam nec nibh tincidunt tincidunt. Nam sem nunc, laoreet a, aliquam sit amet, luctus a, turpis. Sed at sem ut pede aliquet viverra. Vestibulum aliquam elit quis mi. Morbi mollis. Nunc sapien. Sed tortor. Morbi magna velit, bibendum nec, ullamcorper eget, feugiat iaculis, quam. Praesent id turpis. Quisque aliquam tortor et ligula. Etiam aliquet, enim at pretium malesuada, dolor urna ornare urna, a pellentesque ante lorem vel leo.

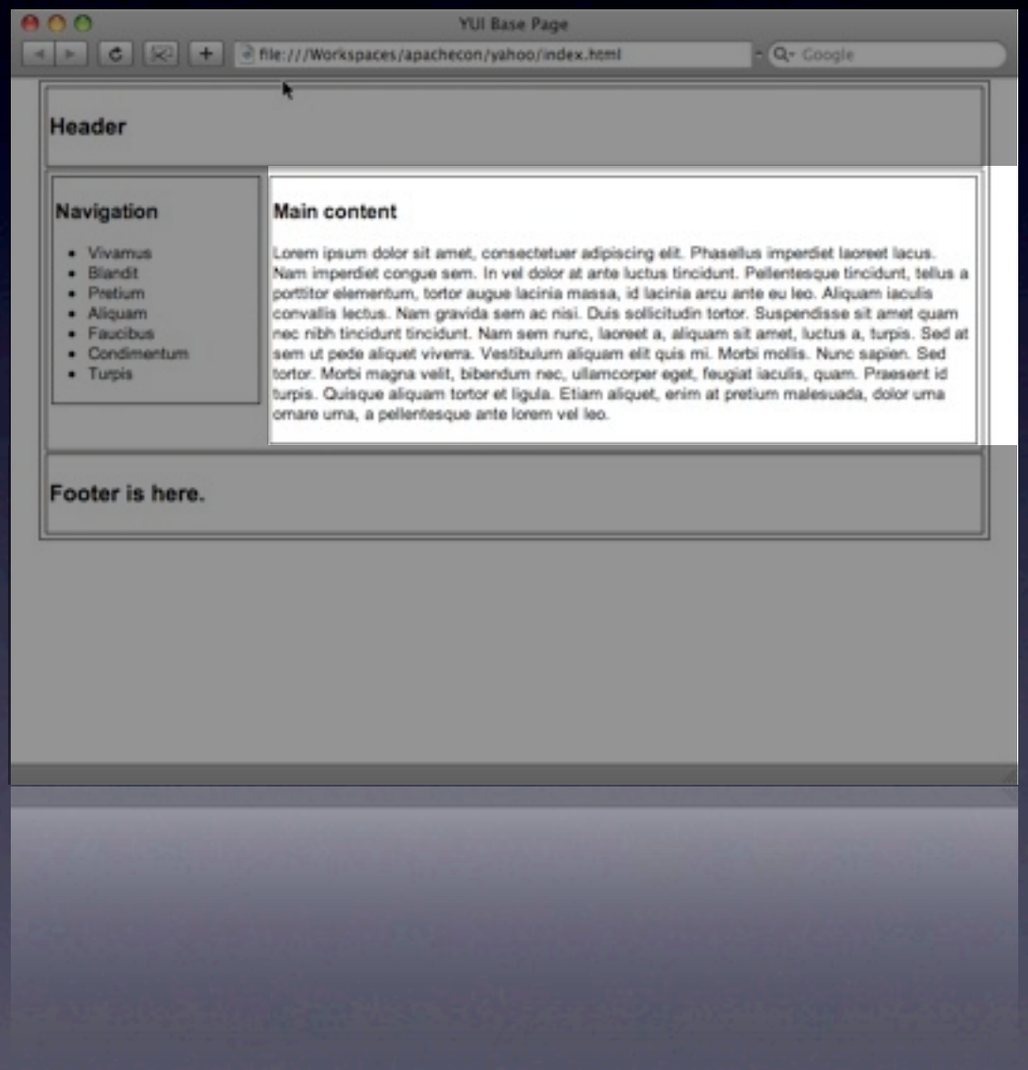
Footer is here.

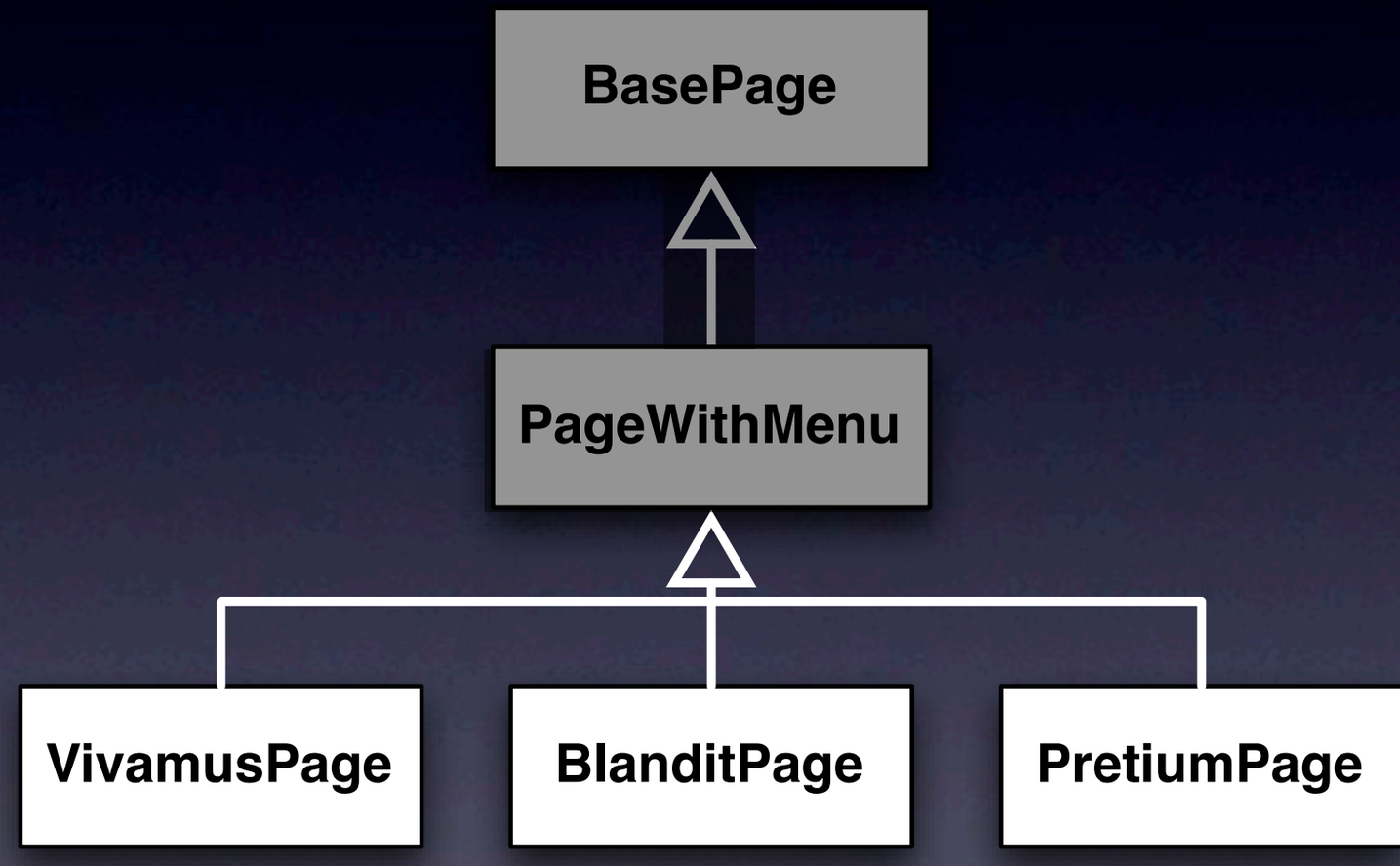

YUI Base Page
file:///Workspaces/apachecon/yahoo/index.html
Google
Header
Navigation
Vivamus
Blandit
Pretium
Aliquam
Faucibus
Condimentum
Turpis
Main content
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus imperdiet laoreet lacus. Nam imperdiet congue sem. In vel dolor at ante luctus tincidunt. Pellentesque tincidunt, tellus a porttitor elementum, tortor augue lacinia massa, id lacinia arcu ante eu leo. Aliquam iaculis convallis lectus. Nam gravida sem ac nisi. Duis sollicitudin tortor. Suspendisse sit amet quam nec nibh tincidunt tincidunt. Nam sem nunc, laoreet a, aliquam sit amet, luctus a, turpis. Sed at sem ut pede aliquet viverra. Vestibulum aliquam elit quis mi. Morbi mollis. Nunc sapien. Sed tortor. Morbi magna velit, bibendum nec, ullamcorper eget, feugiat iaculis, quam. Praesent id turpis. Quisque aliquam tortor et ligula. Etiam aliquet, enim at pretium malesuada, dolor urna ornare urna, a pellentesque ante lorem vel leo.
Footer is here.

BasePage


YUI Base Page
file:///Workspaces/apachecon/yahoo/index.html
Google
Header
Navigation
Vivamus
Blandit
Pretium
Aliquam
Faucibus
Condimentum
Turpis
Main content
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus imperdiet laoreet lacus. Nam imperdiet congue sem. In vel dolor at ante luctus tincidunt. Pellentesque tincidunt, tellus a porttitor elementum, tortor augue lacinia massa, id lacinia arcu ante eu leo. Aliquam iaculis convallis lectus. Nam gravida sem ac nisi. Duis sollicitudin tortor. Suspendisse sit amet quam nec nibh tincidunt tincidunt. Nam sem nunc, laoreet a, aliquam sit amet, luctus a, turpis. Sed at sem ut pede aliquet viverra. Vestibulum aliquam elit quis mi. Morbi mollis. Nunc sapien. Sed tortor. Morbi magna velit, bibendum nec, ullamcorper eget, feugiat iaculis, quam. Praesent id turpis. Quisque aliquam tortor et ligula. Etiam aliquet, enim at pretium malesuada, dolor urna ornare urna, a pellentesque ante lorem vel leo.
Footer is here.


BasePage
PageWithMenu


YUI Base Page
file:///Workspaces/apachecon/yahoo/index.html
Google
Header
Navigation
Vivamus
Blandit
Pretium
Aliquam
Faucibus
Condimentum
Turpis
Main content
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus imperdiet laoreet lacus. Nam imperdiet congue sem. In vel dolor at ante luctus tincidunt. Pellentesque tincidunt, tellus a porttitor elementum, tortor augue lacinia massa, id lacinia arcu ante eu leo. Aliquam iaculis convallis lectus. Nam gravida sem ac nisi. Duis sollicitudin tortor. Suspendisse sit amet quam nec nibh tincidunt tincidunt. Nam sem nunc, laoreet a, aliquam sit amet, luctus a, turpis. Sed at sem ut pede aliquet viverra. Vestibulum aliquam elit quis mi. Morbi mollis. Nunc sapien. Sed tortor. Morbi magna velit, bibendum nec, ullamcorper eget, feugiat iaculis, quam. Praesent id turpis. Quisque aliquam tortor et ligula. Etiam aliquet, enim at pretium malesuada, dolor urna ornare urna, a pellentesque ante lorem vel leo.
Footer is here.


BasePage
PageWithMenu
VivamusPage
BlanditPage
PretiumPage

Java inheritance

```
public class BasePage extends WebPage {
    public BasePage() {
        add(new HeaderPanel("header"));
        add(new FooterPanel("footer"));
    }
}
```

Java inheritance

```
public class PageWithMenu extends BasePage {
    public PageWithMenu() {
        add(new Menu("menu"));
    }
}
```

Java inheritance

```
public class VivamusPage extends PageWithMenu {
    public VivamusPage() {
        add(new Label("vivamus", "..."));
        add(new Link("link"){...});
    }
}
```

Markup inheritance

```
<html>
<body>
    <div id="bd">
        <div wicket:id="header"></div>
        <wicket:child />
        <div wicket:id="footer"></div>
    </div>
</body>
</html>
```

Markup inheritance

```
<html>
<body>
    <div id="bd">
        <div wicket:id="header"></div>
        <wicket:child />
        <div wicket:id="footer"></div>
    </div>
</body>
</html>
```

Markup inheritance

```
<wicket:extend>
    <div wicket:id="menu"></div>
    <div id="main">
        <wicket:child />
    </div>
</wicket:extend>
```

Markup inheritance

```
<wicket:extend>
    <div wicket:id="menu"></div>
    <div id="main">
        <wicket:child />
    </div>
</wicket:extend>
```

Markup inheritance

```
<wicket:extend>
    <table>
        <tr>
            <td wicket:id="name"></td>
            <.....>
        </tr>
    </table
</wicket:extend>
```

Markup inheritance

- Inherit markup from child pages
- Includes <wicket:head> sections
- Works for Panels too
- Very easy to create consistent layout

Part III

Preparing for the real world

Integrate with Spring & Guice

Getting at Spring

```
public class CheesesPage extends WebPage {
    private CheesesDao dao;
    public CheesesPage() {
        List<Cheese> = dao.getCheeses();
        ...
    }
}
```

Getting at Spring

```
public class CheesesPage extends WebPage {
    @SpringBean
    private CheesesDao dao;
    public CheesesPage() {
        List<Cheese> = dao.getCheeses();
        ...
    }
}
```

Make Application Spring aware

```
protected void init() {
    addComponentInstantiationListener(
            new SpringComponentInjector(this));
}
```

Make Wicket Guice aware

```
protected void init() {
    addComponentInstantiationListener(
        new GuiceComponentInjector(this,
                                   getModule()));
}
private Module getModule() {
    return new Module() {
        public void configure(Binder binder) {
            binder.bind(CheesesDao.class)
                  .to(CheesesDaoImpl.class);
    }};
}
```

Getting at Guice

```
public class CheesesPage extends WebPage {
    private CheesesDao dao;
    public CheesesPage() {
        List<Cheese> = dao.getCheeses();
        ...
    }
}
```

Getting at Guice

```
public class CheesesPage extends WebPage {
    @Inject
    private CheesesDao dao;
    public CheesesPage() {
        List<Cheese> = dao.getCheeses();
        ...
    }
}
```

Putting your application into production

Test your Wicket application

WicketTester

- Test components directly, or their markup
- Runs tests without starting server
- Ajax testing (server side)
- Runs in IDE, ant, maven builds
- Achieves high code coverage

HelloWorld test

```
@Test
public void labelContainsHelloWorld() {
}
```

HelloWorld test

```
@Test
public void labelContainsHelloWorld() {
    WicketTester tester =
                        new WicketTester();
}
```

HelloWorld test

```
@Test
public void labelContainsHelloWorld() {
    WicketTester tester = new WicketTester();
    tester.startPage(HelloWorld.class);
}
```

HelloWorld test

```
@Test
public void labelContainsHelloWorld() {
    WicketTester tester = new WicketTester();
    tester.startPage(HelloWorld.class);
    tester.assertLabel("message",
                       "Hello, World!");
}
```

Link test

```
@Test
public void countingLinkClickTest() {
}
```

Link test

```
@Test
public void countingLinkClickTest() {
    WicketTester tester =
                        new WicketTester();
}
```

Link test

```
@Test
public void countingLinkClickTest() {
    WicketTester tester = new WicketTester();
    tester.startPage(LinkCounter.class);
}
```

Link test

```
@Test
public void countingLinkClickTest() {
    WicketTester tester = new WicketTester();
    tester.startPage(LinkCounter.class);
    tester.assertModelValue("label", 0);
}
```

Link test

```
@Test
public void countingLinkClickTest() {
    WicketTester tester = new WicketTester();
    tester.startPage(LinkCounter.class);
    tester.assertModelValue("label", 0);
    tester.clickLink("link");
}
```

Link test

```
@Test
public void countingLinkClickTest() {
    WicketTester tester = new WicketTester();
    tester.startPage(LinkCounter.class);
    tester.assertModelValue("label", 0);
    tester.clickLink("link");
    tester.assertModelValue("label", 1);
}
```

Navigation test

```
@Test
public void navigateToSecondPage() {
    WicketTester tester = new WicketTester();
    tester.startPage(new FirstPage());
    tester.clickLink("link");
    tester.assertRenderedPage(SecondPage.class);
}
```

Navigation test

```
@Test
public void navigateToSecondPage() {
    WicketTester tester = new WicketTester();
    tester.startPage(new FirstPage());
    tester.clickLink("link");
    tester.assertRenderedPage(
                                SecondPage.class);
}
```

URL Makeover

UGLY url

http://cheesr.com/shop?wicket:bookmarkablePage=

%3Acom.cheesr.shop.CheeseDetailsPage&cheese=edam

Prettier urls

http://cheesr.com/cheeses?cheese=edam

http://cheesr.com/cheeses/cheese/edam

http://cheesr.com/cheeses/edam

Mount pages for prettiness

http://cheesr.com/cheeses/edam

```
mount(new IndexedParamUrlCodingStrategy("cheeses",
                                        CheeseDetailsPage.class));
```

Configure for production

2 configuration modes

1. Development
   maximize developer happiness
2. Deployment
   maximize end-user happiness

Development mode

- exceptional error pages
- dynamic markup reloading
- no caching
- no javascript/css optimizations
- discover mistakes early (serialization, missing components, ...)
- Wicket debugger visible

Production mode

- Cache markup resources
- No checks
- Don’t display stack traces to users
- Minimize/compress JavaScript
- Don’t generate wicket tags
- Wicket debugger not visible

Default setting

```
INFO - WebApplication - [WicketInActionApplication] Started Wicket
                                                    version 1.3.0 in
development mode
*******************************************************************
*** WARNING: Wicket is running in DEVELOPMENT mode.             ***
***                                  ^^^^^^^^^^^                ***
*** Do NOT deploy to your live server(s) without changing this.  ***
*** See Application#getConfigurationType() for more information. ***
*******************************************************************
```

Go to production

- System property:
  java -Dwicket.configuration=deployment
- Servlet/filter initialization parameter (web.xml)
- Context initialization parameter (web.xml)

```
<init-param>
    <param-name>configuration</param-name>
    <param-value>deployment</param-value>
</init-param>
```

Part IV

The future of Wicket

Wicket 1.4

- **Java 5 all the way**
  CustomerLink(String id, IModel model)

Wicket 1.4

- Java 5 all the way
  CustomerLink(String id, IModel<Customer>)
- Mostly code compatible with Wicket 1.3
- Improved OSGi awareness

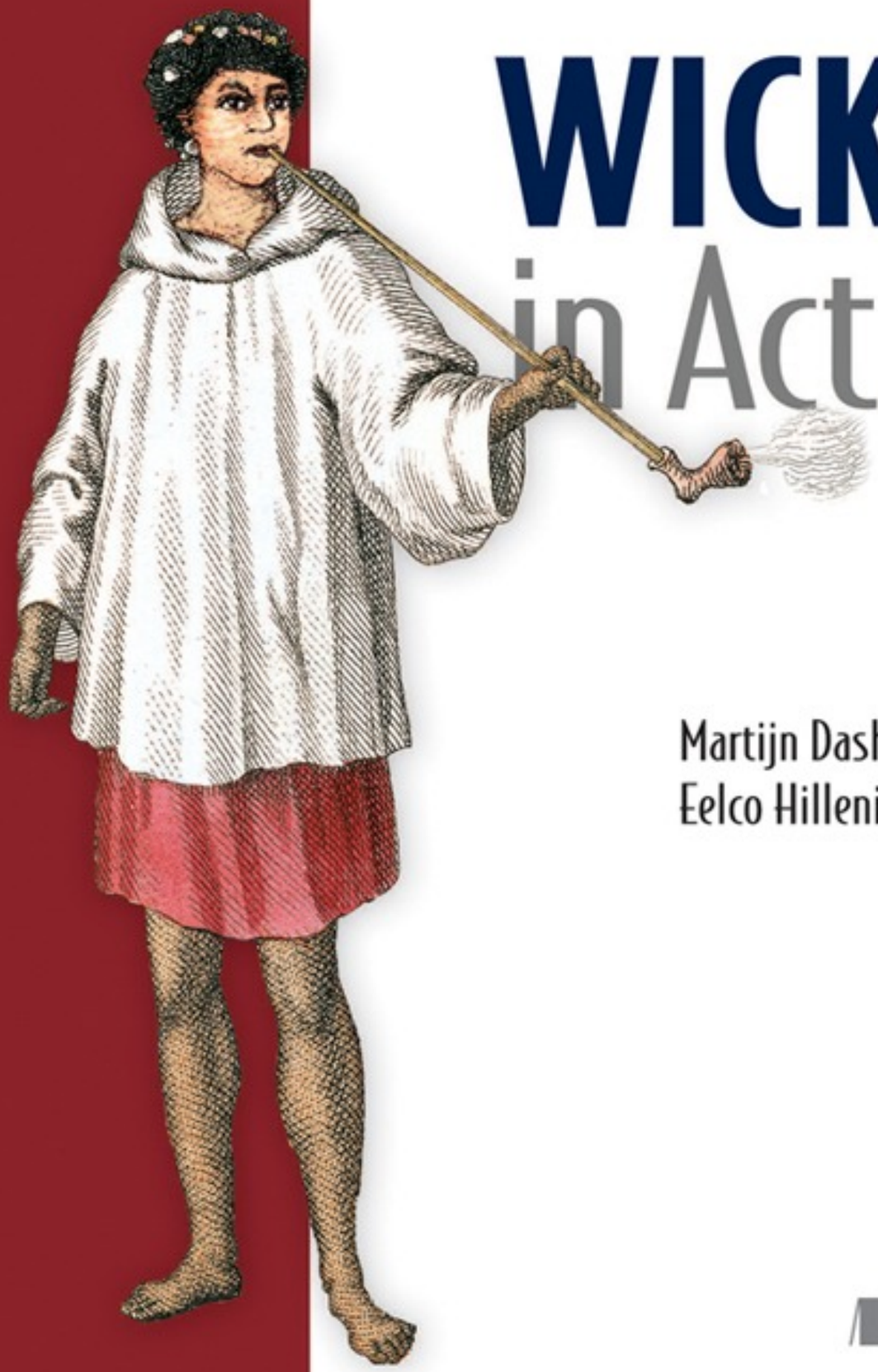
Wicket 1.5

- New Ajax implementation
- Improved unit test support
- Simplified resources
- Improved multi-window support
- and much more...

More information

- http://wicket.apache.org
  - blog: http://wicketinaction.com
    - irc: **##wicket** @ irc.freenode.net
      - User list: users@wicket.apache.org

One more thing...



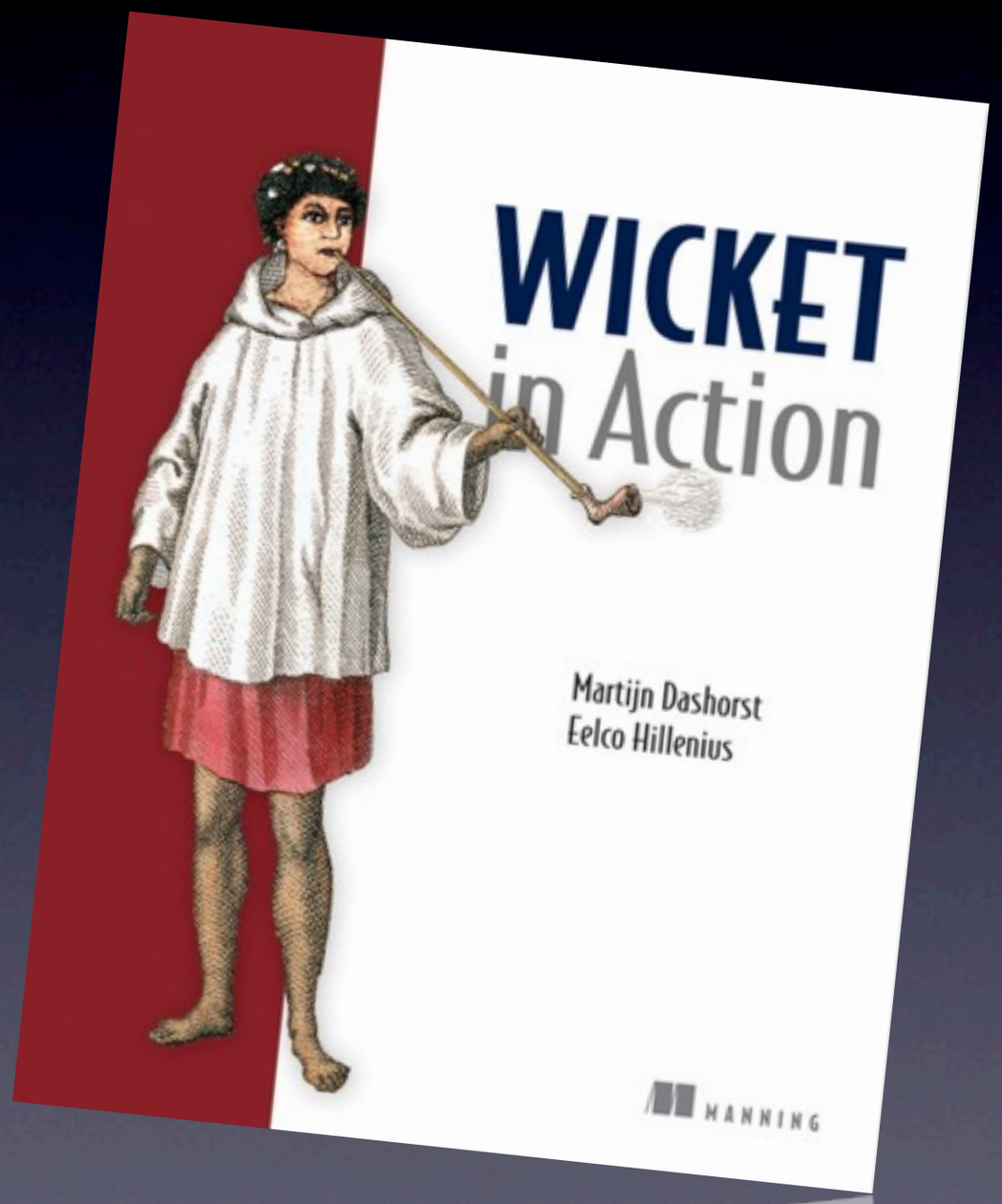
WICKET in Action

Martijn Dashorst
Eelco Hillenius

MANNING

Wicket in Action

- Martijn Dashorst and Eelco Hillenius
- August, 2008 | 392 pages
- ISBN: 1932394982
- $44.99 Softbound print book - (includes free ebook)
- $27.50 PDF ebook



the first **5** persons asking a question will receive a **free** E-Book of **Wicket in Action**

Questions?

Ask!

Thank you

for your attention...