

Rich Web Applications in Server-side Java without Plug-ins or JavaScript

Joonas Lehtinen, PhD

Vaadin Ltd - CEO

vaadin.com/vaadin

[@joonaslehtinen](https://twitter.com/joonaslehtinen)

vaadin } >

Rich Web Applications in Server-side Java without Plug-ins or JavaScript

Joonas Lehtinen, PhD

Vaadin Ltd - CEO

vaadin.com/vaadin

[@joonaslehtinen](https://twitter.com/joonaslehtinen)

vaadin } >



vaadin }>

V

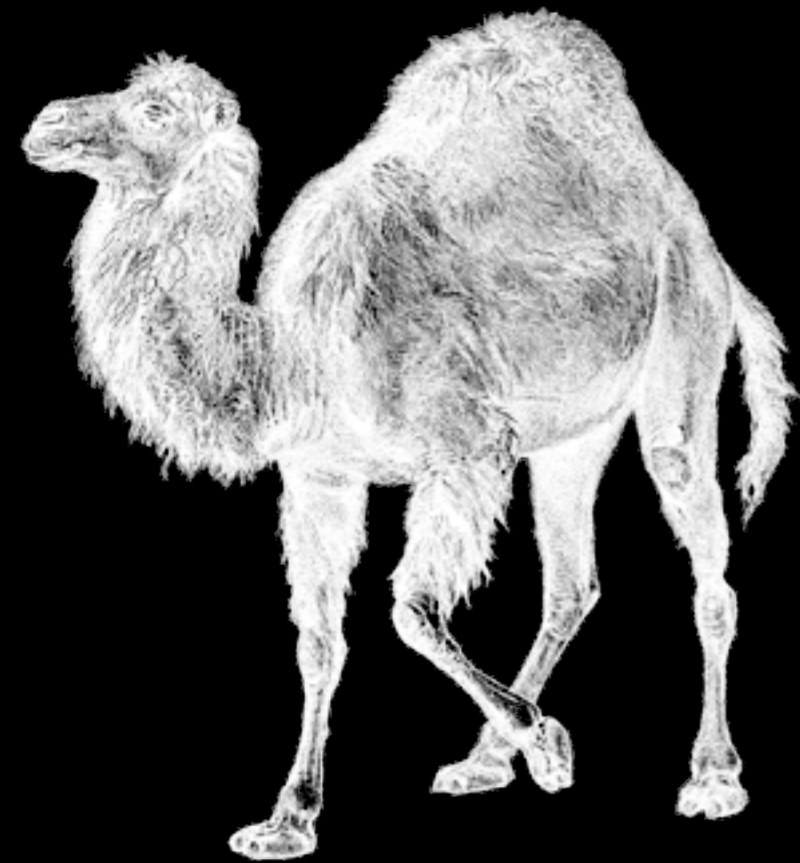


**Vaadin is a
UI framework
for rich web
applications**

java } html >

A white mountain peak graphic on a black background. The peak is a simple triangle with a sharp apex, centered horizontally. The text "History behind Vaadin" is written in white, sans-serif font, centered below the peak.

History behind Vaadin



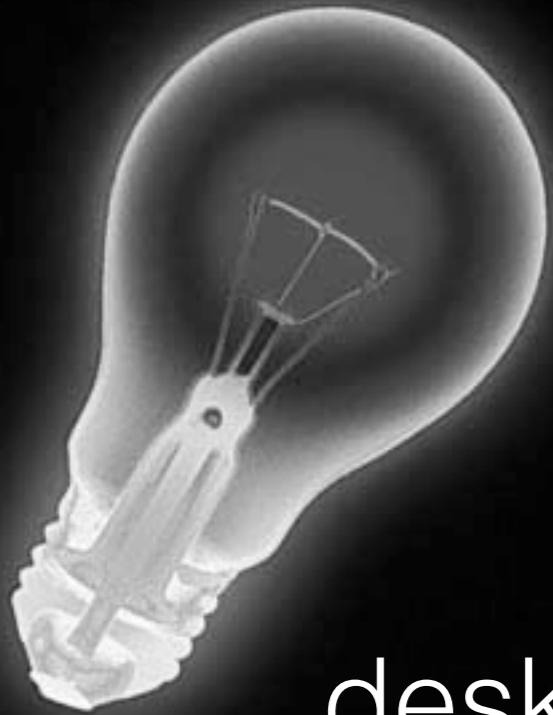
healthcare portal, >100 kloc of perl,

1998

web 1.0, netscape, ie5, ie6, ...



thinking of
object oriented design, desktop, Java, U and I ...



desktop programming paradigm for web!

found

2000



IT MILL

millstone

2002

ajax

2005

google web toolkit

2008

re-released as

2009

vaadin }>



plug-ins, mainstream

2010

2011

Used by >50.000
developers in >150
countries

Building apps for
fortune 500, startups,
governments, ...
across all industries

Collaborating by
contributing > 200
plug-ins and helping
each other on a forum
with >1500 posts / m

Backed by 50+ person
full-time team

see our transparent vaadin.com/roadmap

2012



**Apache
License**

Contents

Server-side RIA

What is it? Pros & cons?

Discussion



In practice

Coding a Vaadin
application step-by-step

Vaadin

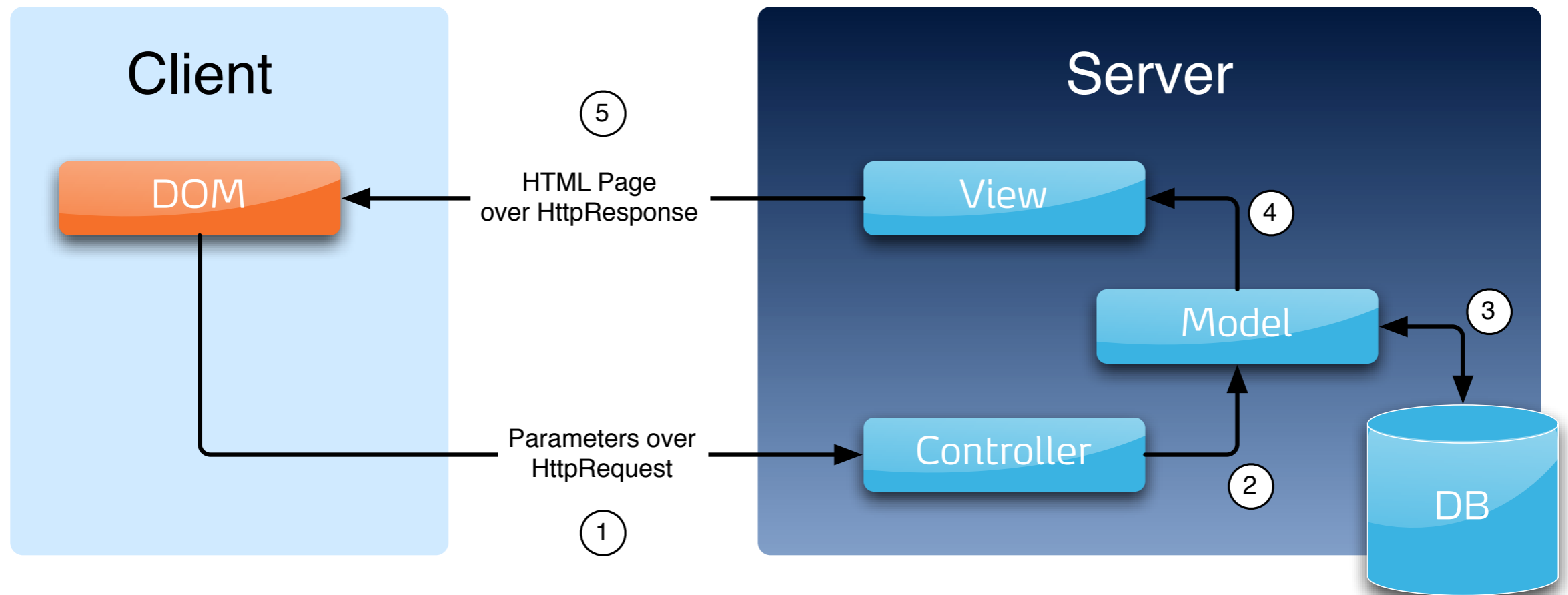
Technology, Features, Getting started



Server-side RIA



“Web 1.0”

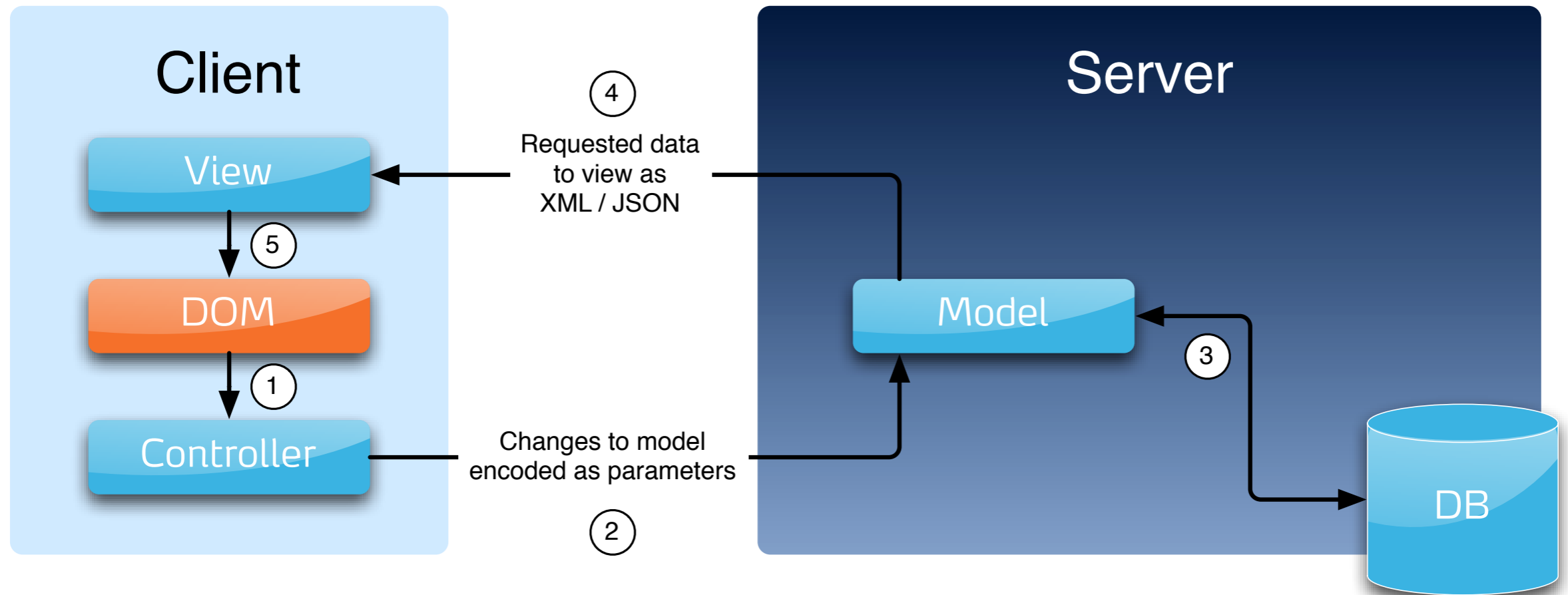




client-side ria

fat client

Client-side RIA



~~Refactoring?~~

~~Automated
API docs?~~

weak typing

~~Compile
time
checking?~~

~~Code
completion?~~

JavaScript
facilitated
spaghetti

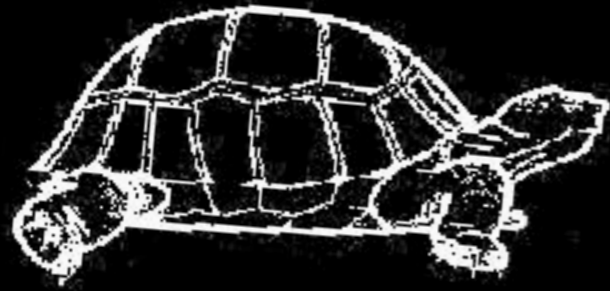
> 1 developers

+

weak typing

+

> 10k SLOC



performance

browsers
are
different



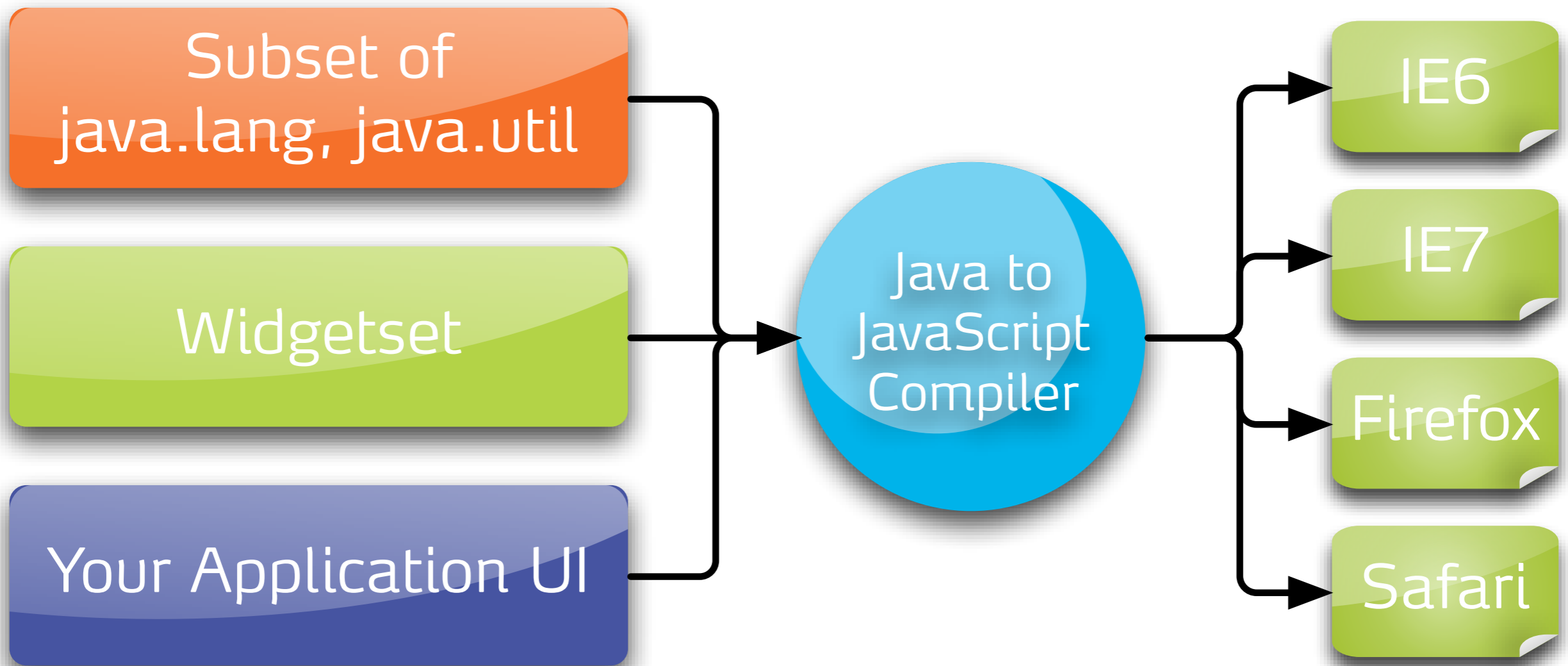
features



bugs



Google Web Toolkit



simpler

- 100% Java
- Static typing
- Object oriented
- Excellent tooling

less bugs

- Stop debugging
JavaScript spaghetti
- Ignore most
browser differences

client-side

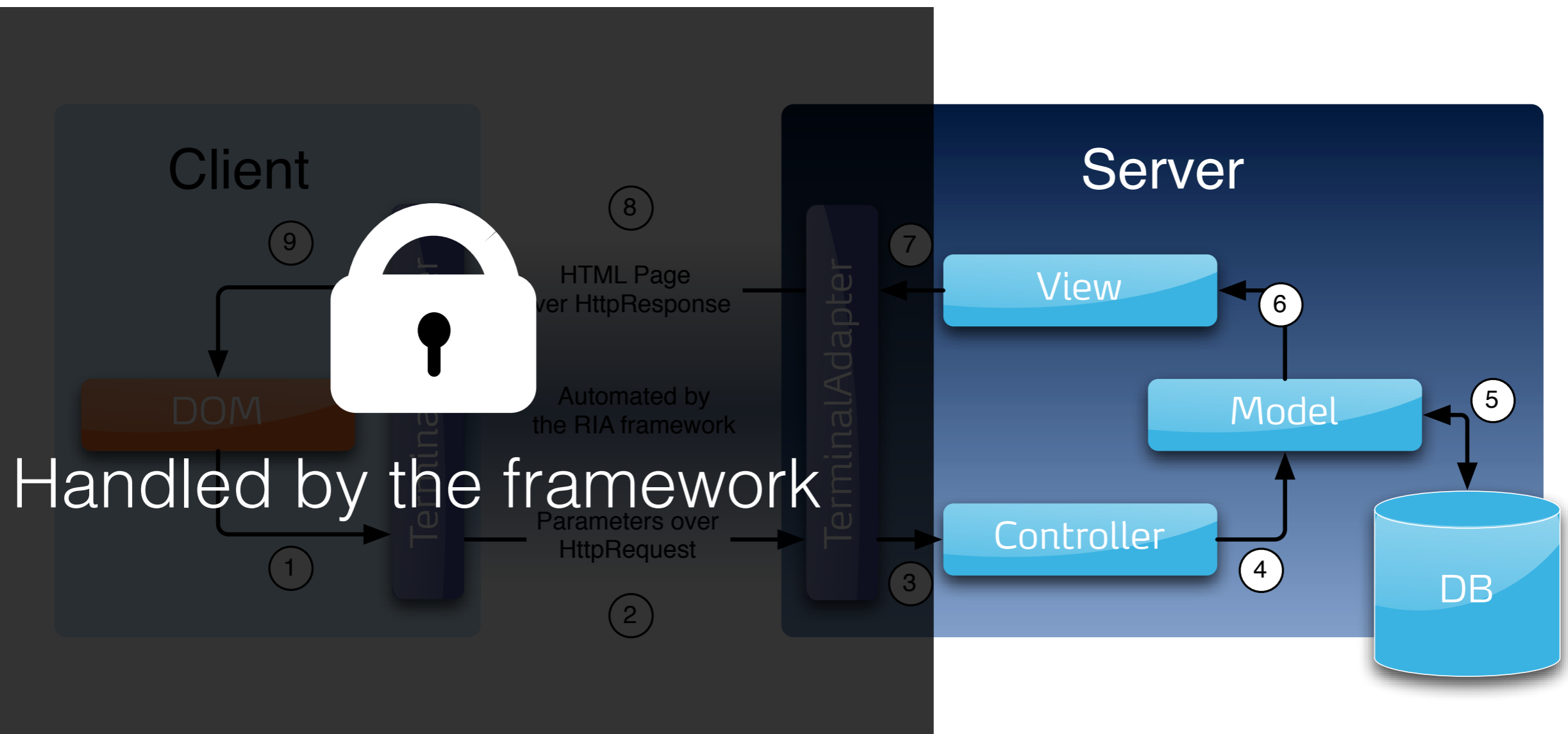
- UI in client
- Asynchronous RPC
- Services (for UI)

Building wonderful
apps doesn't
require writing **fat**
web clients.

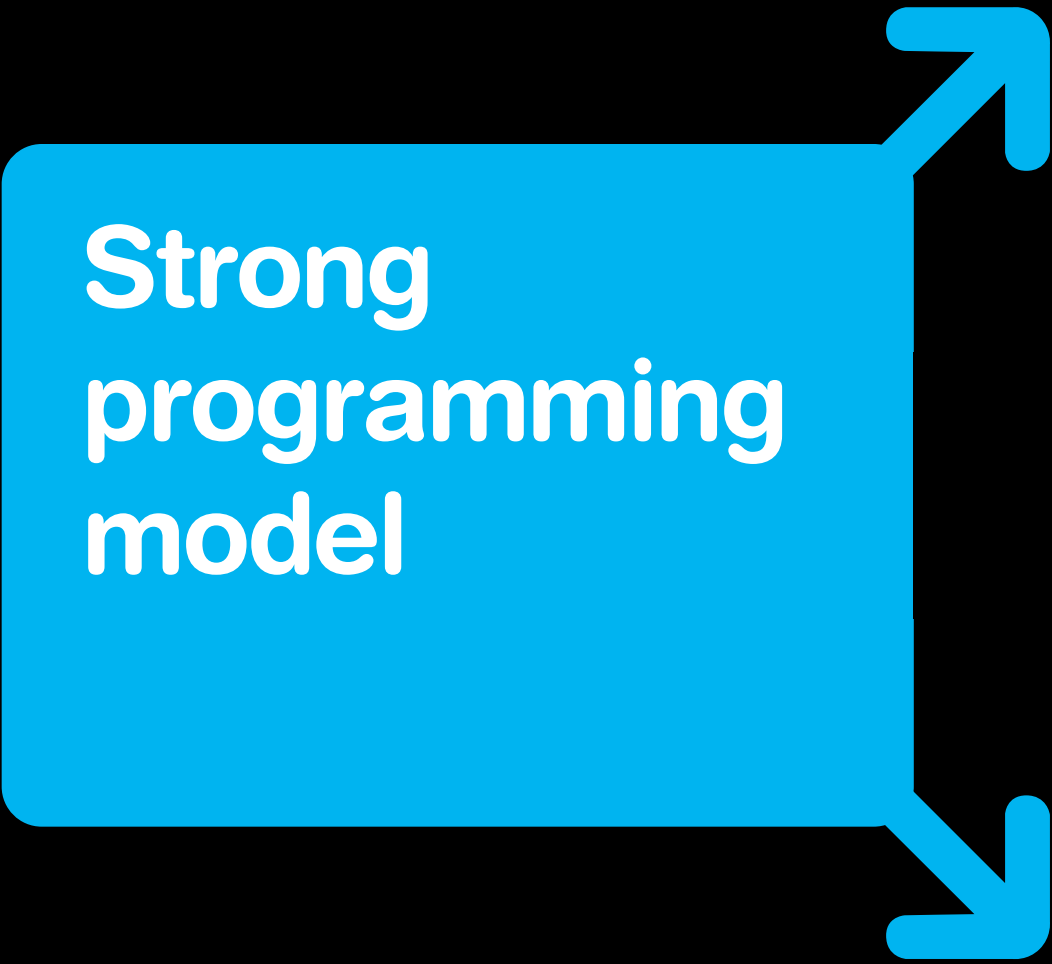
server-side ria
thin client



Server-side RIA



High level



**Strong
programming
model**

simpler

- 100% Java
- Strong typing
- Object oriented
- Excellent tooling

less bugs

- No JavaScript debugging
- Ignore browsers

server environment

- Access any library
- Use any language
- Runtime is not limited by browser
- Trusted
- Well defined

simplified communications

- No RPC layer
- No services (for UI)
- Synchronous
- Direct access to server API

secure

- Code stays in server
- No services (for UI)
- Predictable runtime

not as scalable

UI state is stored in
the server memory

Measured **12.000**
active concurrent
users per server
for a ticketing app

* Amazon EC2-large; 20.622 Ajax requests /
minute peak; limited by storage layer

no offline mode













connection to server
is always required

#1 benefit
development
is really fast



Seeing is believing

**What we can do
in 20 minutes?**

NAME	LAST MODIFIED	SIZE
 .DS_Store	Fri Nov 11 08:56:34 CET 2011	21508
 .localized	Mon Oct 26 18:15:21 CET 2009	0
▼  docs	Wed Nov 09 16:22:26 CET 2011	0
 AbsoluteLayout.ComponentPosition.html	Tue Nov 24 12:36:47 CET 2009	7878
 AbsoluteLayout.html	Tue Nov 24 12:36:47 CET 2009	5724
 AbstractComponent.ComponentErrorEvent.html	Tue Nov 24 12:36:47 CET 2009	10237
 AbstractComponent.ComponentErrorHandler.htm	Tue Nov 24 12:36:47 CET 2009	9045
 AbstractComponent.html	Thu Nov 10 19:11:42 CET 2011	27020
 AbstractComponentContainer.html	Tue Nov 24 12:36:47 CET 2009	15433
 AbstractField.ReadOnlyStatusChangeEvent.html	Tue Nov 24 12:36:47 CET 2009	5974
 AbstractField.html	Tue Nov 24 14:36:02 CET 2009	16614
 AbstractLayout.html	Tue Nov 24 12:36:47 CET 2009	11879

[Overview](#) [Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV NEXT

[FRAMES](#) [NO FRAMES](#)

Uses of Interface com.vaadin.ui.AbstractComponent.ComponentErrorHandler

Packages that use [AbstractComponent.ComponentErrorHandler](#)

[com.vaadin.ui](#)

Uses of [AbstractComponent.ComponentErrorHandler](#) in [com.vaadin.ui](#)

Methods in [com.vaadin.ui](#) that return [AbstractComponent.ComponentErrorHandler](#)

AbstractComponent.ComponentErrorHandler	AbstractComponent.getErrorHandler () Gets the error handler for the component.
---	---

Methods in [com.vaadin.ui](#) with parameters of type [AbstractComponent.ComponentErrorHandler](#)

AbstractComponent	AbstractComponent.setErrorHandler (AbstractComponent.ComponentErrorHandler errorHandler)
-----------------------------------	--

NAME	LAST MODIFIED	SIZE
.DS_Store	Fri Nov 11 08:56:34 CET 2011	21508
.localized		
docs		
AbsoluteLa		
AbsoluteLa		
AbstractCo		
AbstractCo		
AbstractCo		
AbstractCo		
AbstractFie		
AbstractFie		
AbstractLay		

Editor [X]

Document HTML

B **I** **U** X₂ X² [List Icons] [Link] [Image] [Unlink]

Background [v] Foreground [v] Font [v] Size [v]

[Overview](#) [Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#) [FRAMES](#) [NO FRAMES](#)

Uses of Interface

[com.vaadin.ui.AbstractComponent.ComponentErrorHandler](#)

Packages that use [AbstractComponent.ComponentErrorHandler](#)

<u>com.vaadin.ui</u>

Uses of [AbstractComponent.ComponentErrorHandler](#) in [com.vaadin.ui](#)

Methods in [com.vaadin.ui](#) that return [AbstractComponent.ComponentErrorHandler](#)

<u>AbstractComponent.ComponentErrorHandler</u>	<u>AbstractComponent.getErrorHandler()</u> Gets the error handler for the component.
--	--

Methods in [com.vaadin.ui](#) with parameters of type [AbstractComponent.ComponentErrorHandler](#)

void	<u>AbstractComponent.setErrorHandler(AbstractComponent.ComponentErrorHandler errorHandler)</u> Sets the error handler for the component.
------	--

Methods in [com.vaadin.ui](#) with parameters of type [AbstractComponent.ComponentErrorHandler](#)

`AbstractComponent.setErrorHandler(AbstractComponent.ComponentErrorHandler errorHandler)`

NAME	LAST MODIFIED	SIZE
.DS_Store	Fri Nov 11 08:56:34 CET 2011	21508
.localized		
docs		
AbsoluteLa		
AbsoluteLa		
AbstractCo		
AbstractCo		
AbstractCo		
AbstractCo		
AbstractFie		
AbstractFie		
AbstractLay		

```

Editor
Document HTML
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<!--NewPage-->
<HTML>
<HEAD>
<!-- Generated by javadoc (build 1.5.0_20) on Thu Nov 19 17:05:17 EET 2009 -->
<TITLE>
Uses of Interface com.vaadin.ui.AbstractComponent.ComponentErrorHandler (Vaadin)
</TITLE>

<LINK REL = "stylesheet" TYPE = "text/css" HREF = ".././././stylesheet.css" TITLE = "Style">

<SCRIPT type = "text/javascript">
function windowTitle()
{
parent.document.title = "Uses of Interface com.vaadin.ui.AbstractComponent.ComponentErrorHandler (Vaadin)";
}
</SCRIPT>
<NOSCRIPT>
</NOSCRIPT>

</HEAD>

<BODY BGCOLOR = "white" onload = "windowTitle();" >

<!-- ===== START OF TOP NAVBAR ===== -->
<A NAME = "navbar_top"><!-- --></A>
<A HREF = "#skip-navbar_top" title = "Skip navigation links"></A>
<TABLE BORDER = "0" WIDTH = "100%" CELLPADDING = "1" CELLSPACING = "0" SUMMARY = "">
<TR>
<TD COLSPAN = 2 BGCOLOR = "#EEEEFF" CLASS = "NavBarCell1">
<A NAME = "navbar_top_firstrow"><!-- --></A>
<TABLE BORDER = "0" CELLPADDING = "0" CELLSPACING = "3" SUMMARY = "">
<TR ALIGN = "center" VALIGN = "top">
<TD BGCOLOR = "#EEEEFF" CLASS = "NavBarCell1"> <A HREF = ".././././overview-summary.html"><FONT CLASS = "NavBarFont1">
<B>Overview</B></FONT></A> &nbsp;  </TD>
<TD BGCOLOR = "#EEEEFF" CLASS = "NavBarCell1"> <A HREF = "../package-summary.html"><FONT CLASS = "NavBarFont1">
<B>Package</B></FONT></A> &nbsp;  </TD>

```

Overview Package Cla

PREV NEXT

Packages th

[com.vaadin.ui](#)

Uses of Abs

Methods in [com.vaadin](#)

[AbstractComponent](#)

Methods in [com.vaadin.ui](#) with parameters of type [AbstractComponent.ComponentErrorHandler](#)

Edit `AbstractComponent.setErrorHandler(AbstractComponent.ComponentErrorHandler errorHandler)`

2 Java class files
50 LOC + imports
no HTML
no CSS
no JavaScript
no Cut-n-Paste

Vaadin Framework

goals

Developer
experience

User
experience

Scalability

ideas



**amazing
components**

User Inteface

Data Source

Theme

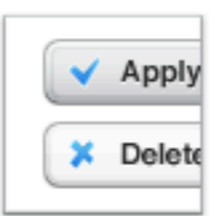
- All Samples
- UI Basics
 - Tooltips
 - Icons
 - Runo theme icons **NEW**
 - Error indicator
 - Progress indication **NEW**
 - JavaScript API **NEW**
- Buttons
 - Push button
 - Link button
 - Checkbox
- Links
 - Link
 - Link, configure window
 - Link, sized window
- Texts
 - Label, plain text
 - Label, preformatted
 - Label, rich text
- Embedding
 - Image **NEW**
 - Flash **NEW**
 - Web content **NEW**
- Value Input Components
 - Dates
 - Pop-up date selection

UI Basics

18 SAMPLES



Tooltips



Icons



Runo theme icons



Error indicator



Progress indication



JavaScript API



Push button



Link button



Checkbox



Link



Link, configure window



Link, sized window



Label, plain text



Label, preformatted



Label, rich text



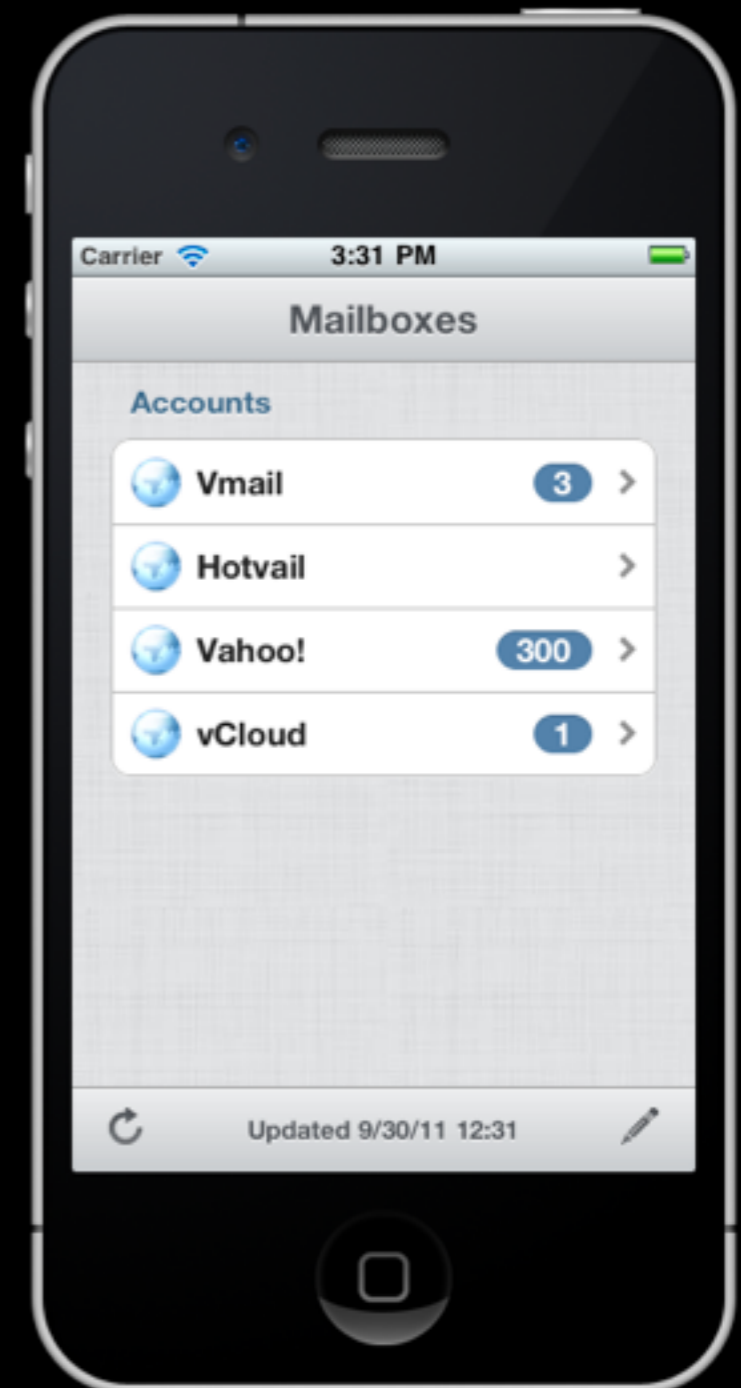
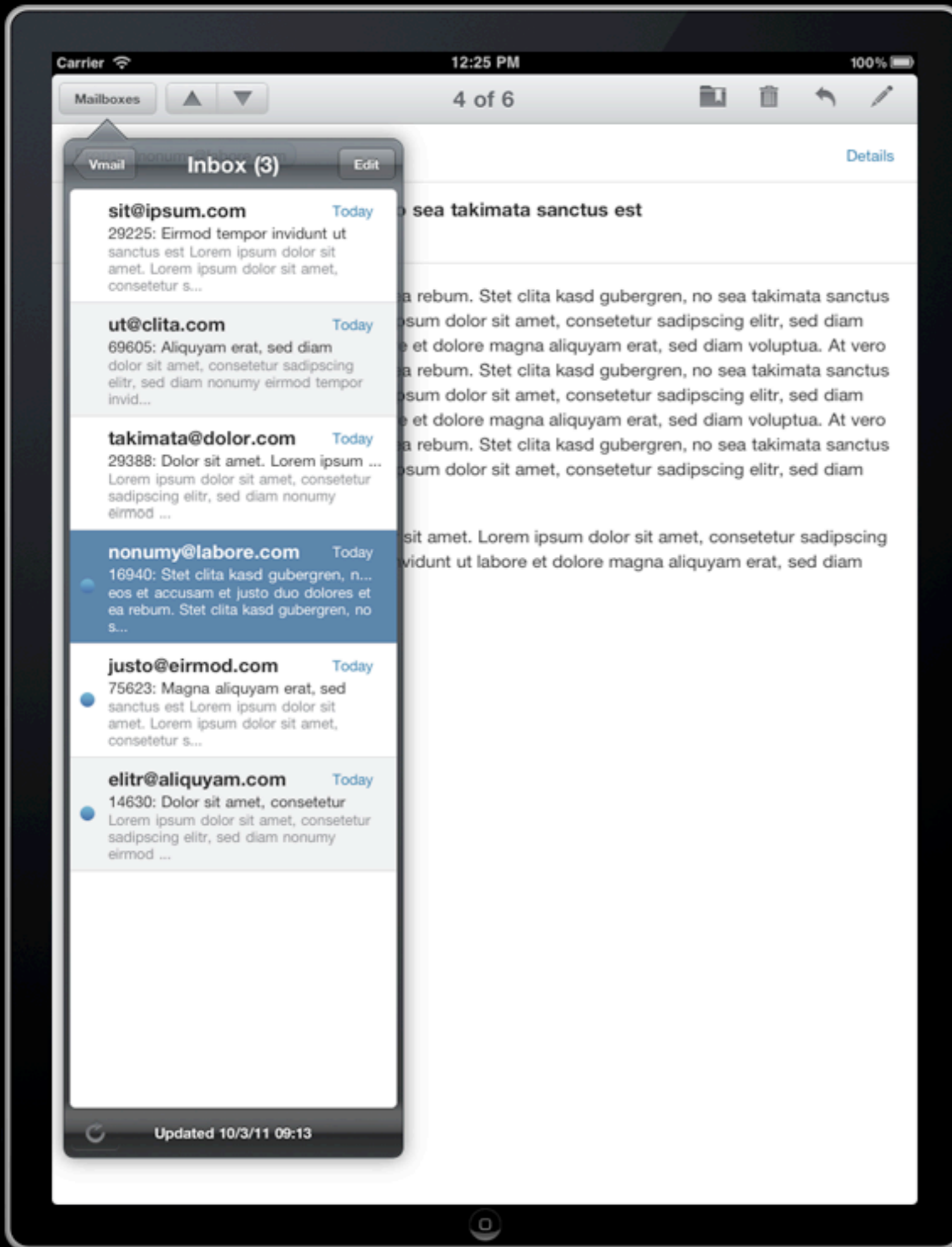
Image



Flash



Web content



User Interface
Data Source
Theme

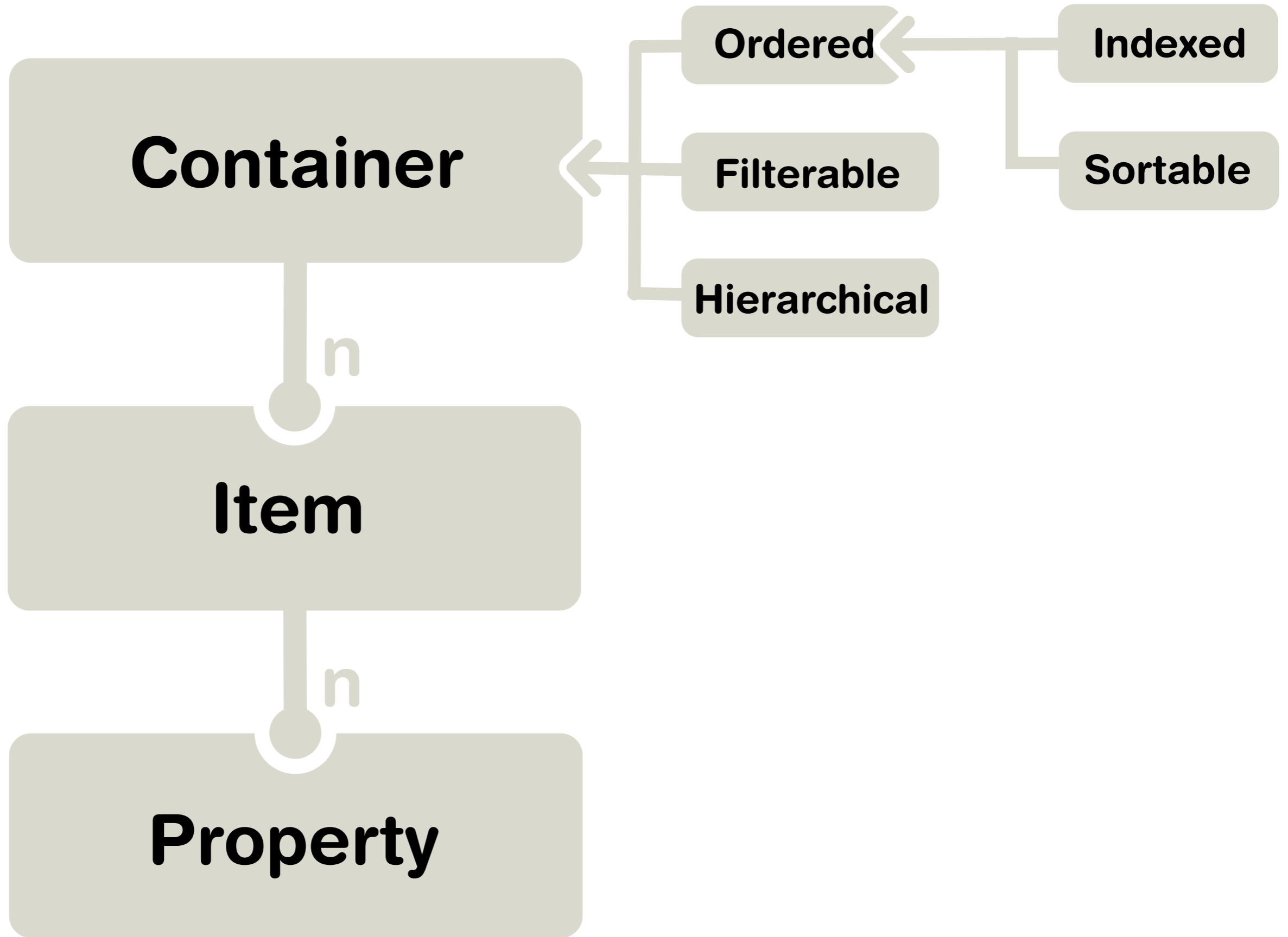
Reindeer

Runo

Chameleon

Custom

User Interface
Data Source
Theme



InMemory, Bean, Method,
Collection, JDBC, JPA,
Hibernate, TextFile,
FileSystem, Properties,
EclipseLink, Lucene,
Mockups, GAE, ... **Your's**

example

SQLContainer

Table

or any other UI component

SQLContainer

FreeformQuery

JDBC

SELECT * FROM ...

```
connectionPool =  
    new SimpleJDBCConnectionPool(  
        "org.hsqldb.jdbc.JDBCDriver",  
        "jdbc:hsqldb:mem:sqlcontainer",  
        "password", "", 2, 2);  
SQLContainer container =  
    new SQLContainer(  
        new FreeformQuery(  
            "SELECT * FROM people",  
            Arrays.asList("ID"),  
            connectionPool));  
Table table = new Table(null, container);
```

Table

or any other UI component

SQLContainer



TableQuery

JDBC



Database Table

- **lazy loading**
- **advanced filtering**
- **sorting**
- **connection pooling**
- **transactions**
- **optimistic locking**
- **autocommit mode**


- **HSQLDB**
- **MySQL**
- **PostgreSQL**
- **Oracle**
- **MS SQL Server**



combine power of

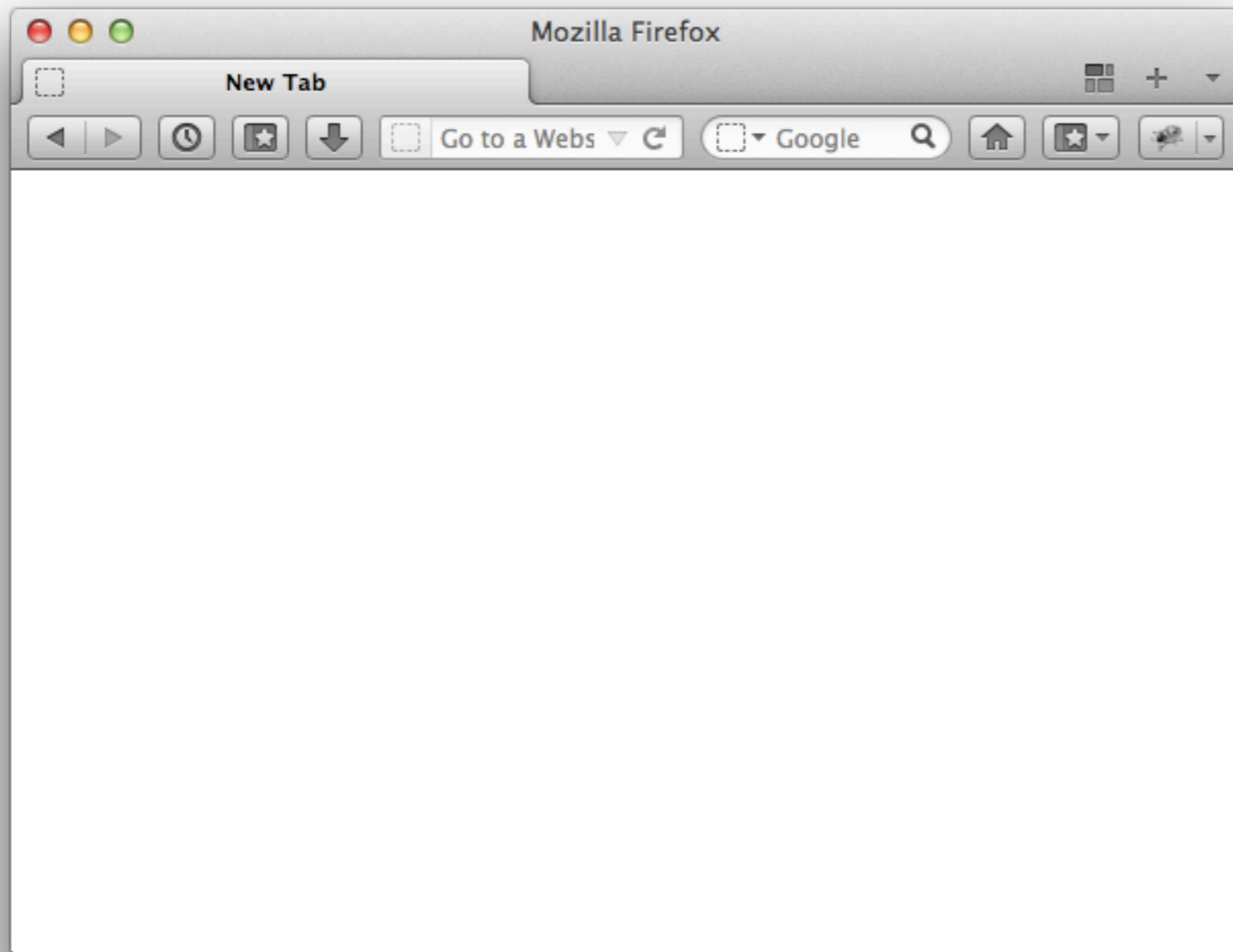
- Server-side RIA
- Google Web Toolkit

Layers of abstraction

	backend server	frontend server	RPC	browser	browser
	any language	any language	json / xml	java 	javascript
Vaadin	required	required	<i>optional</i>	<i>optional</i>	<i>optional</i>
GWT	required	required	required	required	<i>optional</i>
ExtJS	required	required	required	X	required

**How does it
work, really?**

```
name = new TextField("Name");  
greetButton = new Button("Greet");  
  
greetButton.addListener(new ClickListener() {  
    public void buttonClick(ClickEvent event) {  
        mainWindow.showNotification("Hi " + name);  
    }  
});
```



- Initial HTML
- CSS (theme)
- Images
- JavaScript

830k total



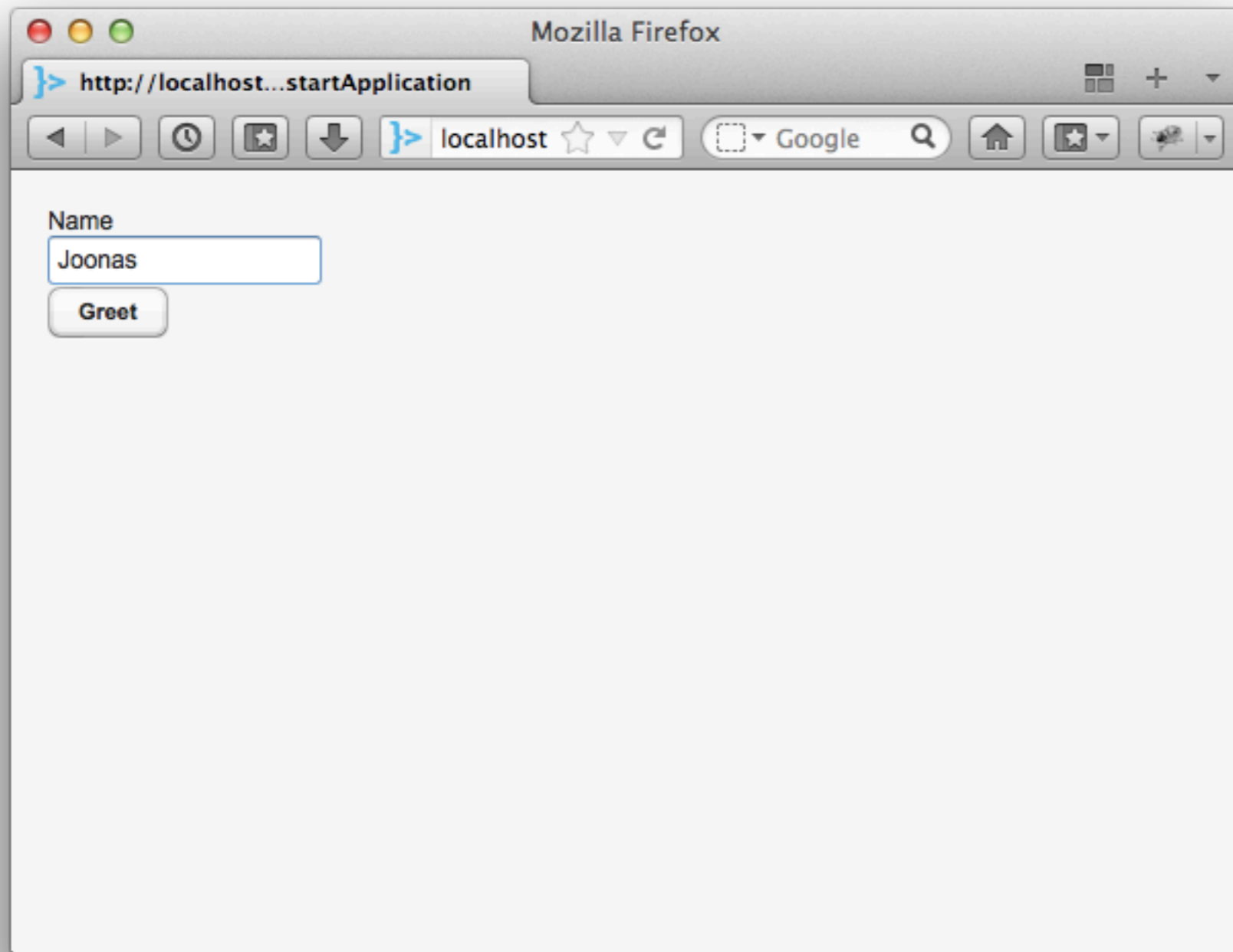
compress

250k



**reduced
widgetset**

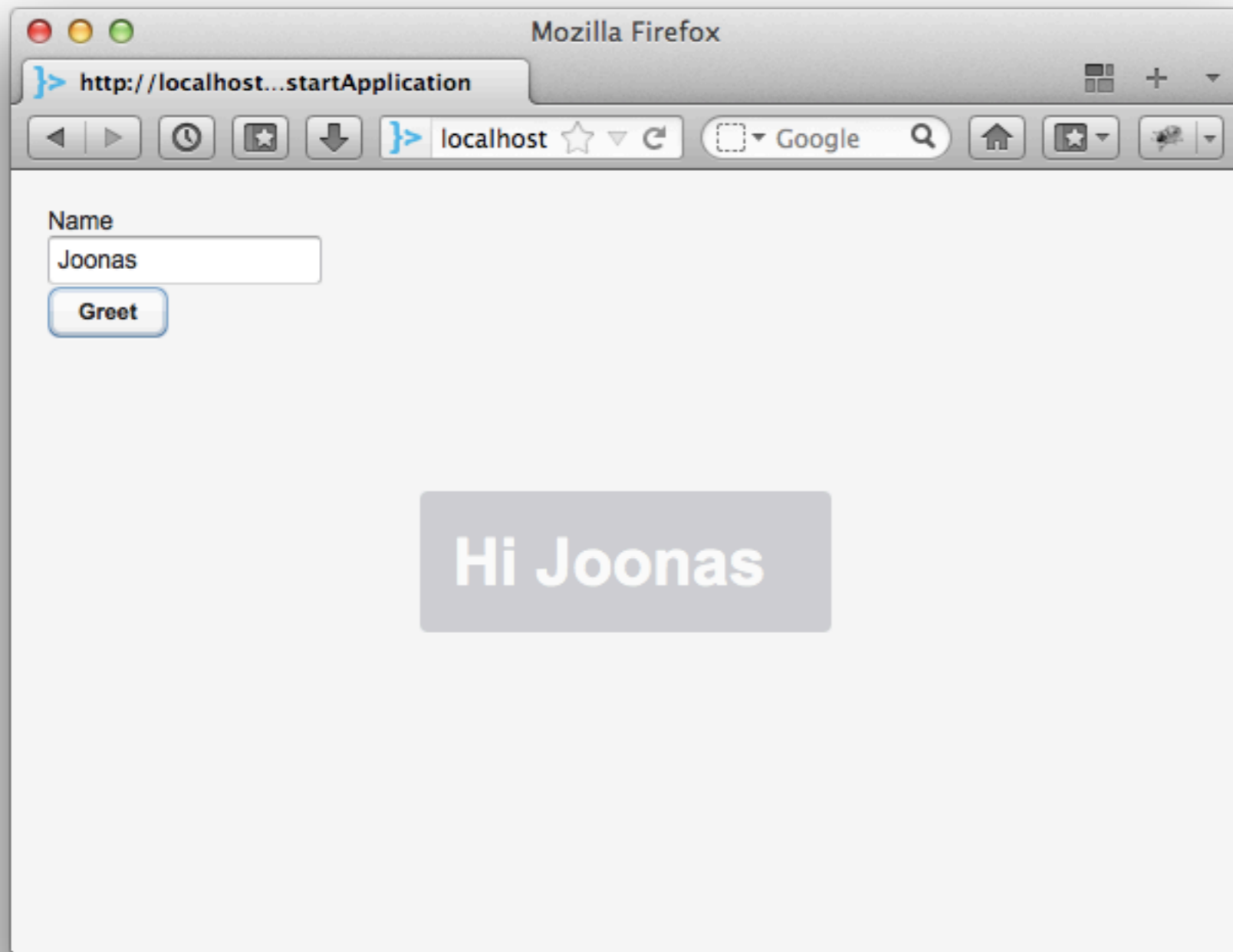
120k



- name="Joonas"
- button clicked

150 bytes

```
public void buttonClick(ClickEvent event) {  
    mainWindow.showNotification("Hi " + name);  
}
```

- name="Joonas"
- button clicked

150 bytes



- Add notification

466 bytes

Vaadin UI component architecture

“UI Component”

- Button, Table, Tree, ...
- Server-side data
- Full Java API

HTTP(S)

“Widget”

- Client-side peer for the component
- Runs on JavaScript

Java

- Compiled with JDK

Java

- Google Web Toolkit

Creating new UI
components is
really easy

New Vaadin Widget

New Component wizard

This wizard creates a new Vaadin widget.

Source folder:

Package:

Name:

Superclass:

Template:

Simple client-side and server-side component with client-server communication

Implement two classes

Server-side

“UI Component”

- Define **API**
- Receive client events
- Send UI updates back

Automatic

Client-side


“Widget”

- **Render** to DOM
- Collect user events

Vaadin Add-on Package Export

Vaadin Add-on Package Export
Define which resources should be exported into the Vaadin add-on package.

Select the resources to export.

<input checked="" type="checkbox"/> ▶  test	
--	--

Manifest:

Implementation title:
Name of the add-on. Used in Vaadin Directory.

Implementation version:
Version of the add-on. A "major.minor.revision" format is suggested.

Widgetsets:
Comma separated list of widgetsets included in the add-on. Refers to the GWT xml files (.gwt.xml).

Select the export destination:

JAR file:

Options:

Overwrite existing files without warning

Upload New Add-on

Select a category to post your new add-on to.

Note, that if you're updating a previous add-on, that is done by editing the add-on from the list above.



UI Components



Server-side and/or client-side
UI components



Themes



Themes for Vaadin applications



Data Components



Components related to the
Vaadin data model, e.g.
Container or Validator
implementations



Tools



Tools for Vaadin developers



Miscellaneous



Other Vaadin add-ons

Upload Add-on Package

Directory

Browse

All

UI Components

Data Components

Themes

Tools

Miscellaneous

Guest

[Authoring](#)

[Subscribe RSS](#)

[Help](#)

[FAQ](#)

[Feedback](#)

Most Recent Highest Rated Top Downloads

Showing **CERTIFIED STABLE BETA EXPERIMENTAL**

« Previous Next » 1 2 3 4 5 6 7 8 9 10 11 12 13 14 204 Results

EasyUploads

In [UI Components](#) by [Matti Tahvonen](#)

Use file uploads as fields in Form, upload multiple files at once - easily!

Version 0.4.2 BETA

★★★★★ 2

↓ 304

OpenLayers Wrapper

In [UI Components](#) by [Matti Tahvonen](#)

Vaadin server side components that wrap essential OpenLayers objects

Version 0.4.0 EXPERIMENTAL

★★★★★ 2

↓ 142

I18N4Vaadin

In [Miscellaneous](#) by [Petter Holmström](#)

A small add-on for creating localized applications

Version 0.9.0 BETA

★★★★★ 1

↓ 10

Navigator

In [UI Components](#) by [Joonas Lehtinen](#)

Navigator is an easy to use view manager that supports lazy initialization, bookmarking and multiple browser windows.

Version 0.3 EXPERIMENTAL

★★★★★ 3

↓ 234

ConfirmDialog

In [UI Components](#) by [Run Uilder](#)

A versatile confirm dialog for Vaadin

Version 1.1.0 BETA

★★★★★ 6

↓ 676

CustomField

In [UI Components](#) by [Henri Sara](#)

A form field whose presentation and logic can be customized

Version 0.8.2 BETA

★★★★★ 8

↓ 1075

Drawer

In [UI Components](#) by [Henrik Paul](#)

An animated component to hide or show information

Transactional Container

In [Data Components](#) by [Tommi Laukkanen](#)

Transactional Container offers same base features as Lazy Query Container but reads and writes all data

Directory

Search Add-ons

Browse

All

- UI Components
- Data Components
- Themes
- Tools
- Miscellaneous

Guest
Authoring

- Subscribe RSS
- Help
- FAQ
- Feedback

PaperStack

UI Components by Tom Midd... 4.4 5 404

Report this add-on

PaperStack - Screenshot 1

PaperStack

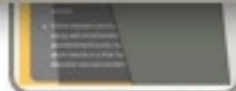
- Current view is the first subcomponent added to this switch to the next subcomponent either by dragging mouse or by clicking on it once.
- PaperStack is a ComponentContainer. Any Vaadin Some Field components in a FormLayout below...



+ More...



Code Example



Screenshot 2



Screenshot 1

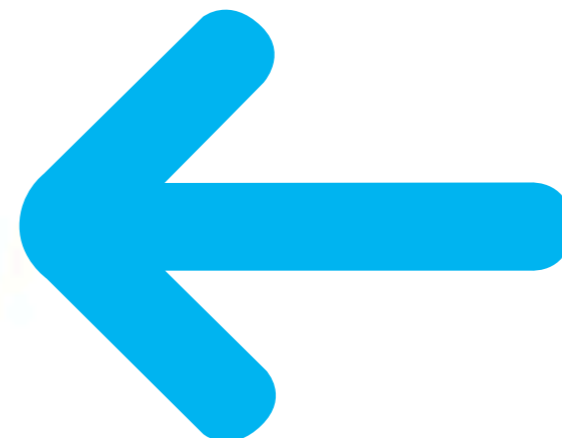
Release notes

0.0.1:

on:
n/paperstack

Project Explorer X Outline


- ▼ PostitNotes
 - ▶ 24 Deployment Descriptor: PostitNotes
 - ▼ Java Resources: src
 - ▼ com.example.postitnotes
 - ▶ PostitnotesApplication.java
 - ▶ com.example.postitnotes.widgetset
 - ▶ Libraries
 - ▶ JavaScript Resources
 - ▶ build
 - ▼ WebContent
 - ▶ META-INF
 - ▶ VAADIN
 - ▼ WEB-INF
 - ▼ lib
 - ▶ paperstack-0.8.1.jar
 - ▶ vaadin-6.5.1.jar
 - ▶ web.xml



Project Explorer X Outline

- PostitNotes
 - Deployment Descriptor: PostitNotes
 - Java Resources: src
 - com.example.postitnotes

Compile widgetset

 Your client side code in the project PostitNotes might need a recompilation. Compile widgetset now?

Suspend automatic recompilation

No Yes

- META-INF
- VAADIN
- WEB-INF
 - lib
 - paperstack-0.8.1.jar
 - vaadin-6.5.1.jar
 - web.xml

```
PaperStack notes = new PaperStack();
```

```
@Override
```

```
public void init() {
```

```
    // === Layout =====
```

```
    HorizontalLayout lo = new HorizontalLayout();
```

```
    Window mainWindow = new Window("Postitnotes Application", lo);
```

```
    lo.setSizeFull();
```

```
    lo.addComponent(notes);
```

```
    lo.setComponentAlignment(notes, Alignment.MIDDLE_CENTER);
```

```
    setMainWindow(mainWindow);
```

```
    notes.setWidth("350px");
```

```
    notes.setHeight("350px");
```

```
    // === Note 1 =====
```

```
    notes.addComponent(new Label("<h1>TODO / Today</h1><div style='font-size: 24px'>" +  
        "<p>Enjoy the conference...</p></div>", Label.CONTENT_XHTML), "#fef49c");
```

```
    // === Note 2 =====
```

```
    notes.addComponent(new Label("<h1>TODO / Tomorrow</h1><div style='font-size: 24px'>" +  
        "<p>Learn Vaadin!</p></div>", Label.CONTENT_XHTML), "#b2ffa1");
```

```
    // === Note 3 =====
```

```
    notes.addComponent(new Label("<div style='font-size: 60px'><center><br/><br/><br/>" +  
        "<DOUBLE<br/><br/><br/>SPEED</center></div>", Label.CONTENT_XHTML), "#b2ffa1");
```

```
    // === Note 4 =====
```

```
    com.vaadin.ui.RichTextArea rta = new RichTextArea();
```

```
    rta.setSizeFull();
```

```
    notes.addComponent(rta);
```

```
    rta.setValue("<span style='font-size: 35pt; color: green;'>You can use any " +  
        "<Vaadin components here...</span>");
```

```
}
```



**embrace
java**

**Any JVM
Language**

Scala **with Scaladin add-on**

```
val layout =  
  new VerticalLayout(width = 100 pct, height = 100 pct) {  
    add(new Label(content = "Persons"))  
    add(new Table(width = 100 pct, height = 100 pct),  
        ratio = 1)  
    add(new HorizontalLayout(spacing = true) {  
      add(new Button("Edit selected", _ => editClicked()))  
      add(new Button("Add new", _ => addNewClicked()))  
    })  
  }  
getMainWindow.setContent(layout)
```

Internet Explorer
Chrome
Firefox
Safari
Opera
iOS
Android

6..

**No
browser
plugins**

**Nothing to
install**

**Servlet
Portlet
(most) clouds**

Apache Tomcat, version 4.1 or later

Oracle WebLogic Server, version 9.2 or later

Oracle WebLogic Portal, version 10gR3

IBM WebSphere Application Server, version 6.1 or later

IBM WebSphere Portal, version 6.1 and 7.0

JBoss Application Server, version 3.2.8 or later

Jetty, version 5 or later

Glassfish, version 2 or later

Liferay Portal 5.2 or later

GateIn Portal 3.1

eXo Platform 3

Google App Engine

Vaadin supports Java Servlet API 2.3 and JSR-168 and JSR-286 Portlet Specifications and should work with any Java application server that conforms to these standards.



Eclipse
IntelliJ IDEA
Netbeans
Maven
Ant
Spring Roo
...

Editor.java - [vt] - [~/IdeaProjects/vt]

vt src Editor

Project Editor

Project

vt (~/.IdeaProjects/vt)

- .idea
- lib
- src
 - Editor
 - vt
 - web
 - WEB-INF
 - web.xml
 - index.jsp
 - vt.iml
- External Libraries

Editor.java

100% x 100%

Text VaadinDesigner

Components

Search

AbsoluteLayout

Hierarchy

- absoluteLayout_1
 - tabSheet_1

Properties (com.vaadin.ui.Absolute...)

Layout Properties

Width: 100.0% x Height: 100.0% x Visible: Margins:

Run tom

Server Tomcat Log

Deployment

- vt:war exploded

Output

```
Mar 7, 2012 7:07:30 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/19 config=null
Mar 7, 2012 7:07:30 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 384 ms
Connected to server
Mar 7, 2012 7:07:31 PM com.vaadin.terminal.gwt.server.AbstractApplicationServlet checkProductionMode
WARNING:
=====
Vaadin is running in DEBUG MODE.
Add productionMode=true to web.xml to disable debug features.
To show debug window, add ?debug to your application URL.
=====
```

4: Run 5: TODO

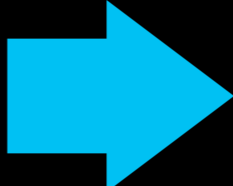
Event Log

Compilation completed successfully (22 minutes ago)

179M of 791M

Maven

```
mvn archetype:generate  
-DarchetypeGroupId=com.vaadin  
-DarchetypeArtifactId=  
  vaadin-archetype-clean
```

```
mvn package  yourproject-1.0.war
```

PizzaForm.java

Name Base Price

Toppings

- Item 0
- Item 1
- Item 2
- Item 3
- Item 4
- Item 5
- Item 6
- Item 7
- Item 8
- Item 9

Components

Search

12/03/10

PopupDateField

AbsoluteLayout

VerticalSplitPanel

Width: auto x Height: auto Visible Margin


Delete Cancel Save

Source Design



New Roo Project

Create a new Roo Project



Project name:

Top level package name:

Project type:

Description:

Roo Installation

Use default Roo installation (currently 'Roo 1.1.1.RELEASE')

Use project specific Roo installation:

Install: [Configure Roo Installations...](#)

Maven Support

Provider:

Contents

Use default location

Use external location

Location:

Working sets

persistence setup --provider HIBERNATE
--database HYPERSONIC_IN_MEMORY

entity --class ~.domain.**Topping**

field string --fieldName **name** --notNull

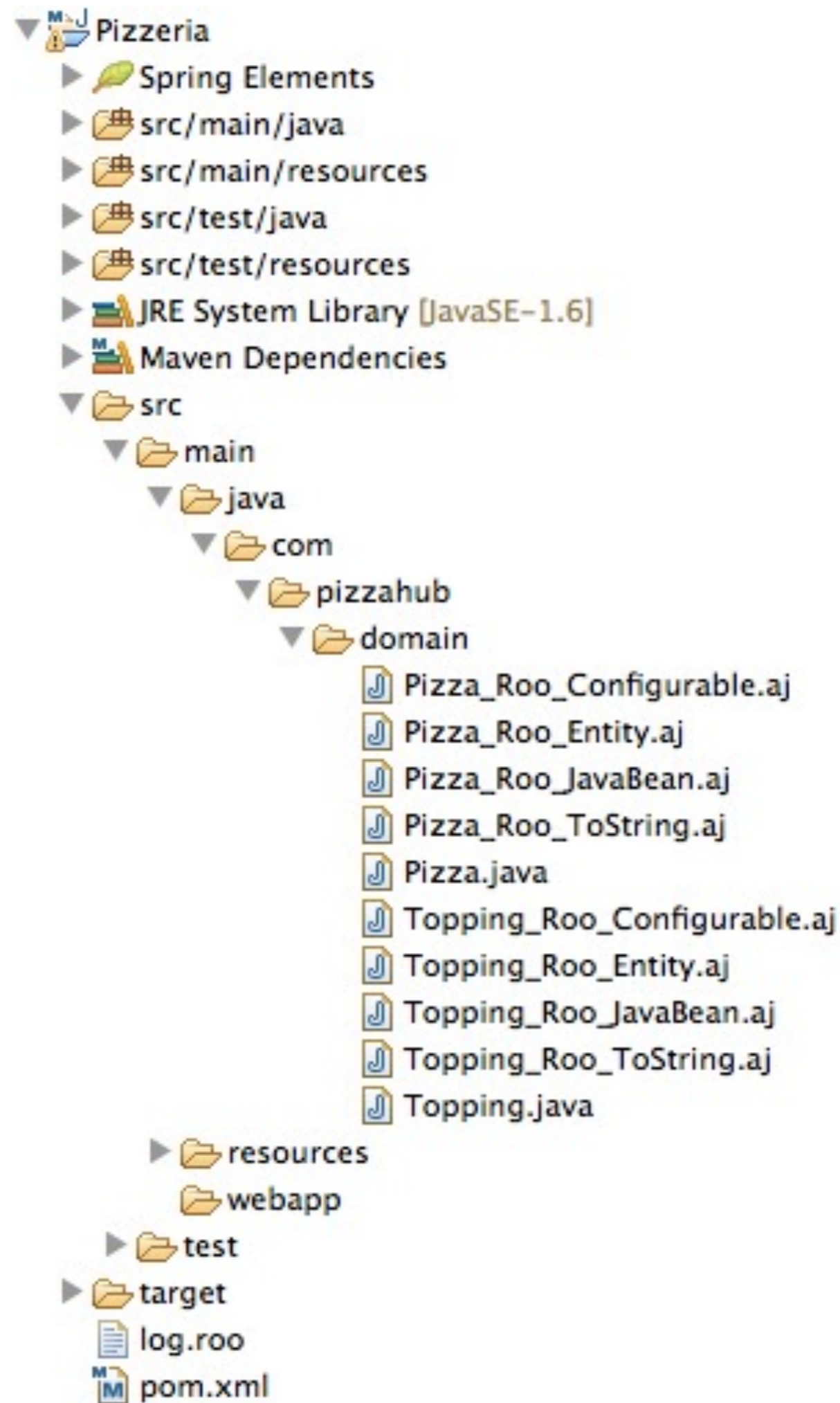
entity --class ~.domain.**Pizza**

field number --fieldName **price**

--type java.lang.Float

field set --fieldName **toppings**

--type ~.domain.Topping



```
package com.pizzahub.domain;
```

```
+ import org.springframework.roo.addon.entity.RooEntity;
```

```
@RooJavaBean
```

```
@RooToString
```

```
@RooEntity
```

```
public class Pizza {
```

```
    private Float price;
```

```
    @ManyToMany(cascade = CascadeType.ALL)
```

```
    private Set<Topping> toppings = new HashSet<Topping>();
```

```
}
```

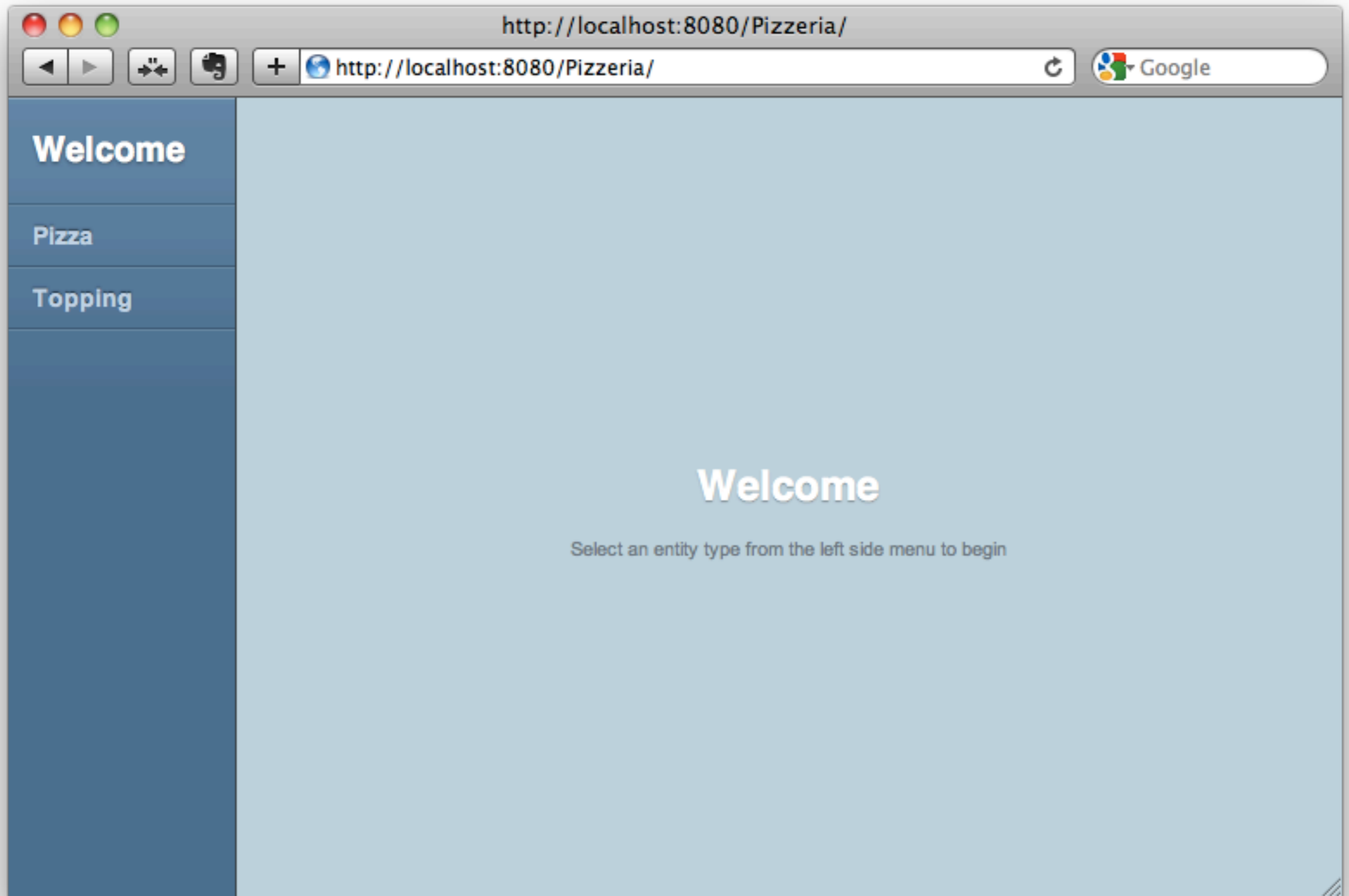
vaadin setup

```
--applicationPackage ~.web  
--baseName PizzaShop  
--themeName pizza  
--useJpaContainer false
```

vaadin generate all

```
--package ~.web.ui  
--visuallyComposable true
```


- ▼ web
 - AbstractEntityView_Roo_AbstractEntityView.aj
 - AbstractEntityView.java
 - AutomaticEntityForm.java
 - EntityEditor.java
 - EntityFieldWrapper.java
 - EntitySetFieldWrapper.java
 - EntityTableColumnGenerator.java
 - PizzaHubApplication.java
 - PizzaHubEntityManagerView_Roo_VaadinEntityManagerView.aj
 - PizzaHubEntityManagerView.java
 - PizzaHubWindow.java
- ▼ ui
 - PizzaForm_Roo_VaadinVisuallyComposableEntityForm.aj
 - PizzaForm.java
 - PizzaView_Roo_VaadinEntityView.aj
 - PizzaView.java
 - ToppingForm_Roo_VaadinVisuallyComposableEntityForm.aj
 - ToppingForm.java
 - ToppingView_Roo_VaadinEntityView.aj
 - ToppingView.java



http://localhost:8080/Pizzeria/

http://localhost:8080/Pizzeria/ Google

ID	PRICE	TOPPINGS
1	10.0	Onion, Olive, Bacon, Pepper
2	8.0	Onion, Pepper, Tomato, Cheese, Mushroom

Welcome

Pizza + New

Topping

Price
6.0

Toppings

Onion
Bacon
Olive
Pepper
Salami
Mushroom

>> <<

**Cheese
Tomato**

Save Cancel [Delete](#)

PizzaForm.java

Price

Toppings

Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9

Save Cancel Delete

Components

Search

Input ▾

Button

ComboBox Button

Hierarchy

absoluteLayout_1

scrollPanel

scrollContent

fieldLayout

priceField

toppingsField

Selected: absoluteLayout_1

Layout Properties

Width auto × Height auto

Visible Margin

The image shows the Vaadin IDE interface for editing a form. The main window displays a design view of a form titled "PizzaForm.java". The form contains a "Price" input field, a "Toppings" list with items "Item 0" through "Item 9", and navigation buttons ">>" and "<<". At the bottom of the design view are "Delete", "Cancel", and "Save" buttons. The right-hand side of the IDE features a component inspector with three panels: "Components", "Hierarchy", and "Selected: scrollContent".

Components Panel: Includes a search bar and two categories: "Input" (containing a "ComboBox") and "Button" (containing a "Button").

Hierarchy Panel: Shows the component tree structure:

- absoluteLayout_1
 - scrollPanel
 - scrollContent (selected)
 - fieldLayout
 - priceField
 - toppingsField

Selected: scrollContent Panel: Shows the properties for the selected component:

- Layout: Properties
- Width: 100.0%
- Height: 100.0%
- Visible:
- Margin:

field string --class ~.domain.Pizza
--notNull --fieldName name --sizeMin 3

The screenshot displays the Vaadin IDE interface for PizzaForm.java in Design mode. The main design area shows a form with two input fields: "Name of the Pizza" and "Price". Below these is a list of items from "Item 0" to "Item 9" with navigation buttons ">>" and "<<". At the bottom are "Delete", "Cancel", and "Save" buttons. The right-hand side features a Component Hierarchy tree where "nameField" is selected, and a Properties panel showing "Width" as "100.0%" and "Height" as "auto".



Welcome

Pizza [+ New](#)

Topping

ID	NAME	PRICE	TOPPINGS
1	Vaadin Special	8.0	mustard, onion, Olives, Tomato

Name of the Pizza *

Vaadin Special

Price

8.0

Toppings

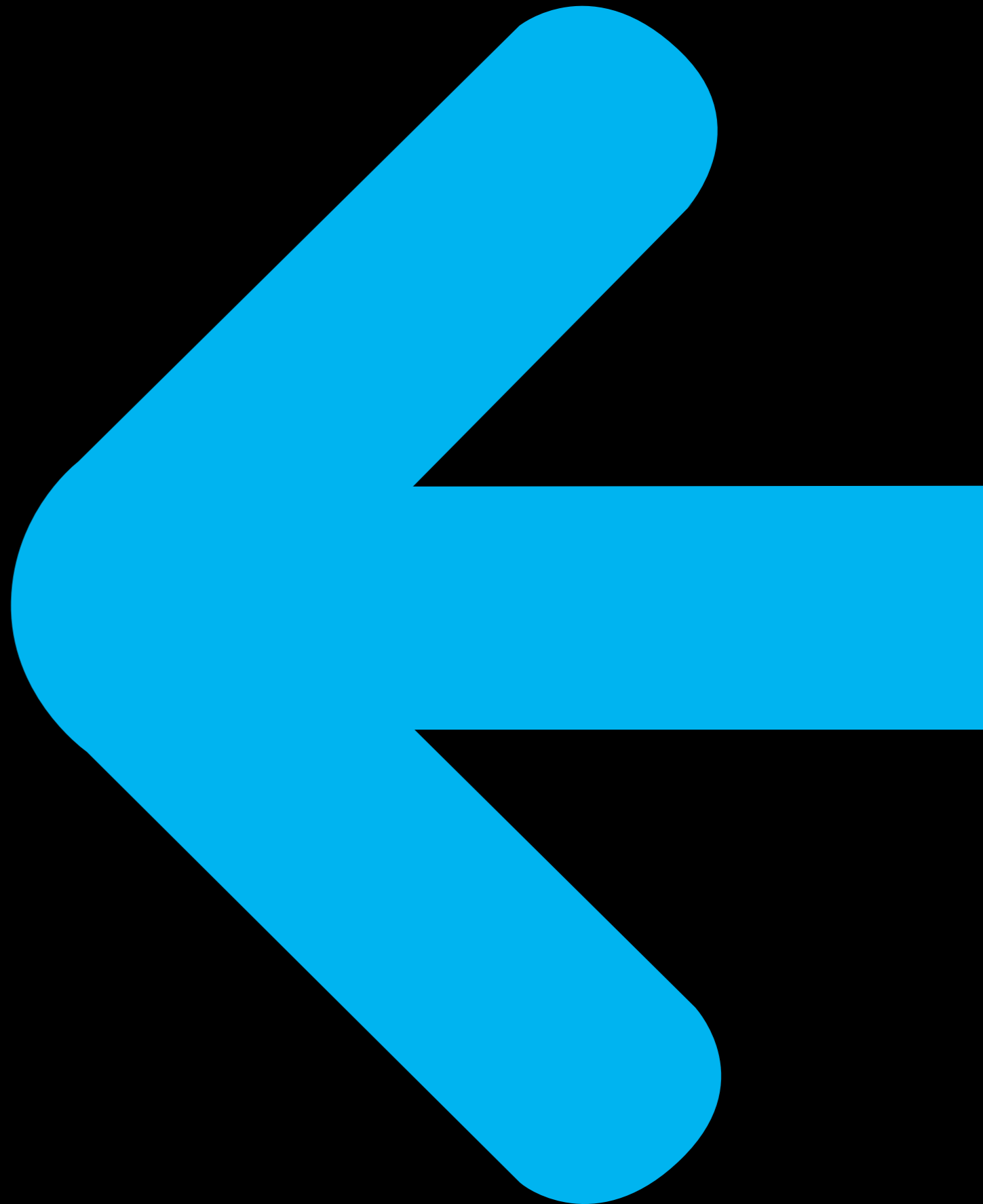
cheese
Bacon
Pepperoni
Pepper



onion
mustard
Olives
Tomato

[Delete](#) [Cancel](#) [Save](#)

getting
started



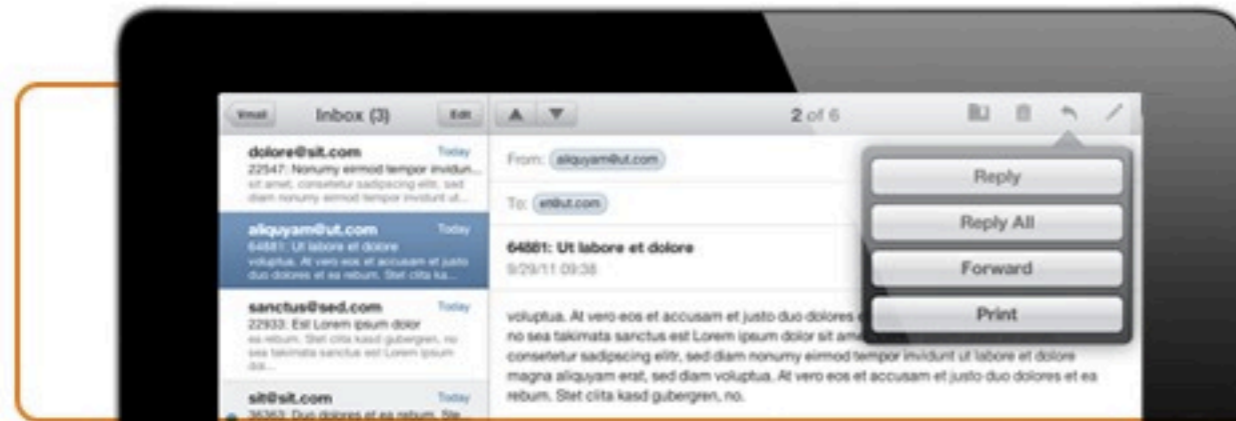
thinking of U and I

Vaadin is a Java framework for building **modern web applications** that look great, perform well and make you and your users happy. [Learn More »](#)

→ [Demo](#)

→ [Download](#) Version **6.7.0**
Apache License

→ [Learn](#)



Special offer for limited time

Vaadin TouchKit 2.0

Mobilize your applications.

[Learn More »](#)



Blog

Subscribe
 Show all »

Vaadin 6.7.0 'Geneva' is out

10/04/11 09:40 by Sami Ekblad

Vaadin TouchKit 2.0

10/03/11 16:48 by Sami Kaksonen

Version Shift

08/29/11 11:46 by Henrik Paul

Download

Latest Release

Download Vaadin 6.7.6 

Jar-only for all platforms, 4.7MB

Release Notes
for Vaadin 6.7.6

Other Packages

Sampler demo application

Looking for the Sampler WAR? [Download Sampler](#)

What's Next?

Eclipse Quick Start »

Using Eclipse, you can get up and running with Vaadin in minutes. [Set up Eclipse for Vaadin »](#)

Step-by-step tutorial »

Follow this tutorial to quickly get grasps with Vaadin development. [Read the tutorial »](#)

Discuss with other developers »

Our discussion forum offers expert help regarding all your problems. [Start a discussion »](#)

vaadin.com/download

Comparison

- Vaadin 6.7
 Wicket 1.5.0
 GWT 2.4.0
 Ext GWT 2.2.5
 Smart GWT 2.5
 Flex 4.5.1
 JQuery 1.6.4
 ZK 5.0.8
 RichFaces 4.0.0
 JavaFX 2.0
 Swing Java 7

 Print

FEATURE	VAADIN 6.7	GWT 2.4.0	WICKET 1.5.0
Core UI widgets	***	**	*
No browser plug-in required	●	●	●
No JavaScript programming required	●	●	●
Framework extensions are done in Java	●	●	
No HTML authoring required	●	●	
No XML configuration required	●	●	●
UI programming on client-side	Java, Javascript	Java, Javascript	Javascript
UI programming on server-side	Java, JVM languages	-	Java, JVM languages
Page request oriented			●
Application oriented	●	●	
Backed up by a corporation	●	●	
Commercial support & guarantees available	●		
Vendor offers experts services to complement t	●		
Free for commercial use	●	●	●
License	Apache 2.0	Apache 2.0	Apache 2.0
Portlet support	●		*
Google App Engine support	●	●	●
Requires a Java Servlet container	●		●
Inherent search engine support			●
Documentation	***	**	**
Eclipse IDE support	***	***	**
		**	**
		**	**
		●	●

vaadin.com/comparison

Download for Free

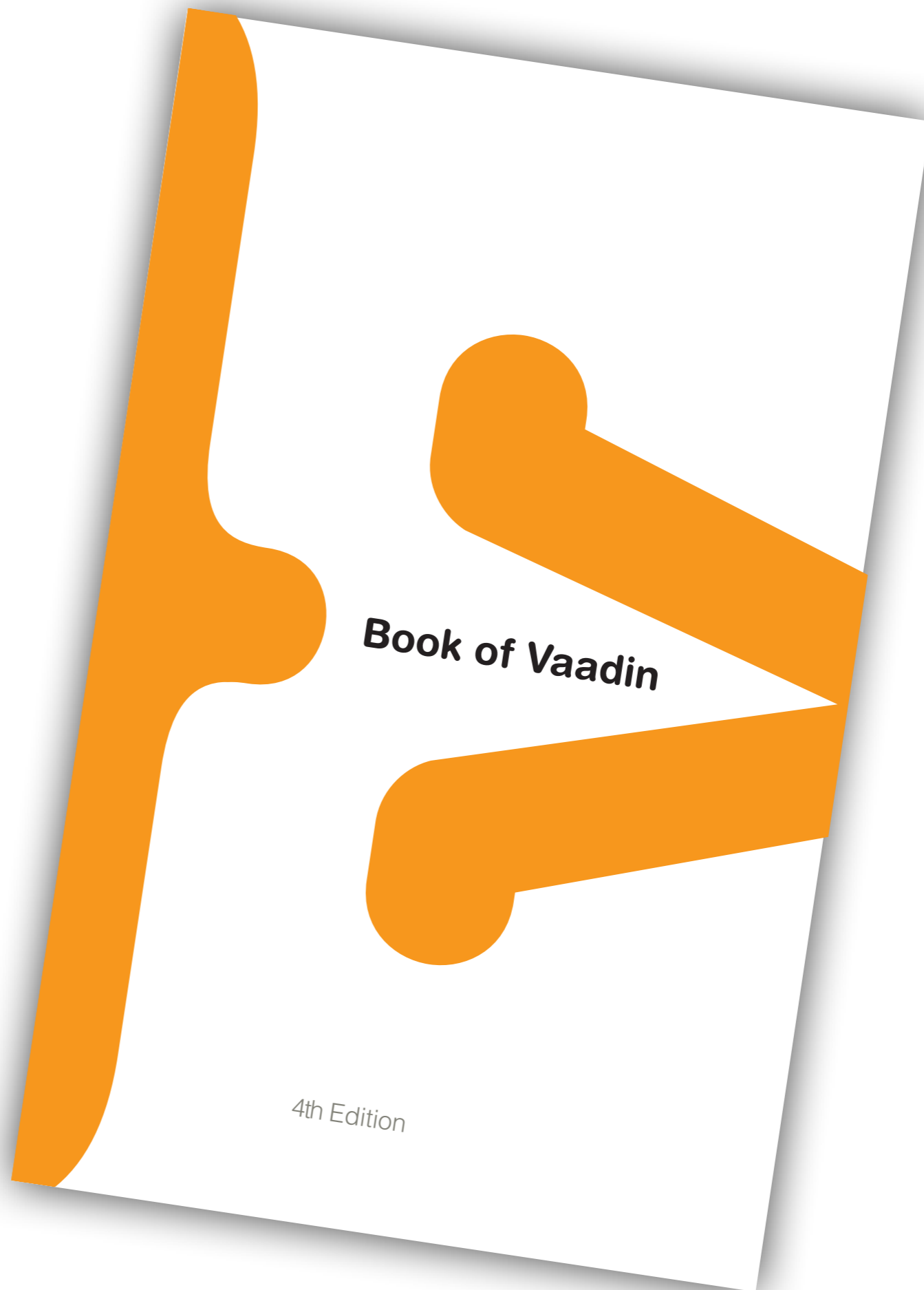
vaadin.com/book



eclipse



674 pages





Cover image by Grand-Duc, Wikipedia

Community Experience Distilled

Learning Vaadin

Master the full range of web development features powered by Vaadin-built RIAs

Foreword by Dr. Joonas Lehtinen
Vaadin, CEO and Founder

Nicolas Fränkel

[PACKT] open source*
PUBLISHING community experience distilled

amazon.com®

\$43



CONTENTS INCLUDE:

- About Vaadin
- Creating An Application
- Components
- Layout Components
- Themes
- Data Binding and more...

Getting Started with Vaadin

By Marko Grönroos

ABOUT VAADIN

Vaadin is a server-side Ajax web application development framework that allows you to build web applications just like with traditional desktop frameworks, such as AWT or Swing. An application is built from user interface components contained hierarchically in layout components.

In the server-driven model, the application code runs on a server, while the actual user interaction is handled by a client-side engine running in the browser. The client-server communications and any client-side technologies, such as HTML and JavaScript, are invisible to the developer. As the client-side engine runs as JavaScript in the browser, there is no need to install plug-ins. Vaadin is released under the Apache License 2.0.

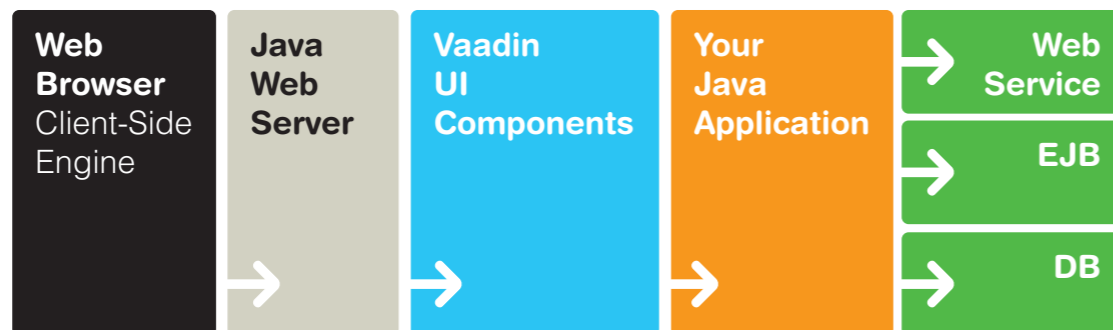


Figure 1: Vaadin Client-Server Architecture

If the built-in selection of components is not enough, you can look for alternatives with the Google Web Toolkit (GWT)

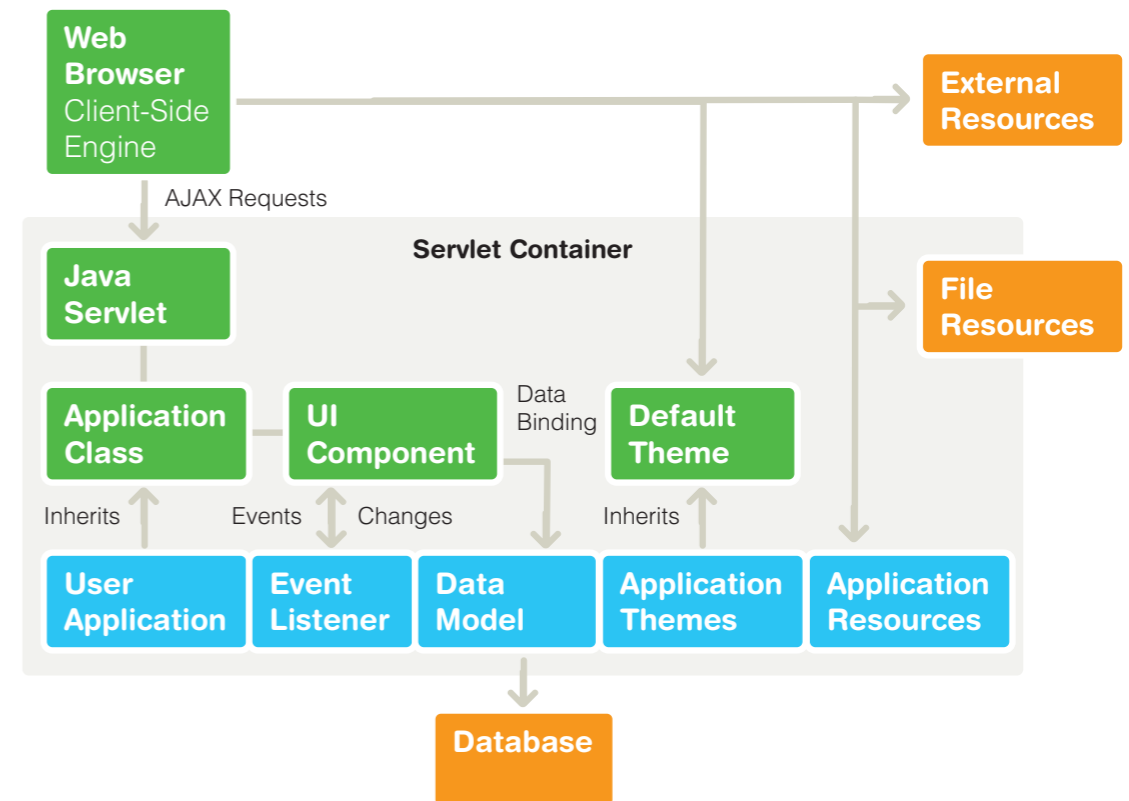


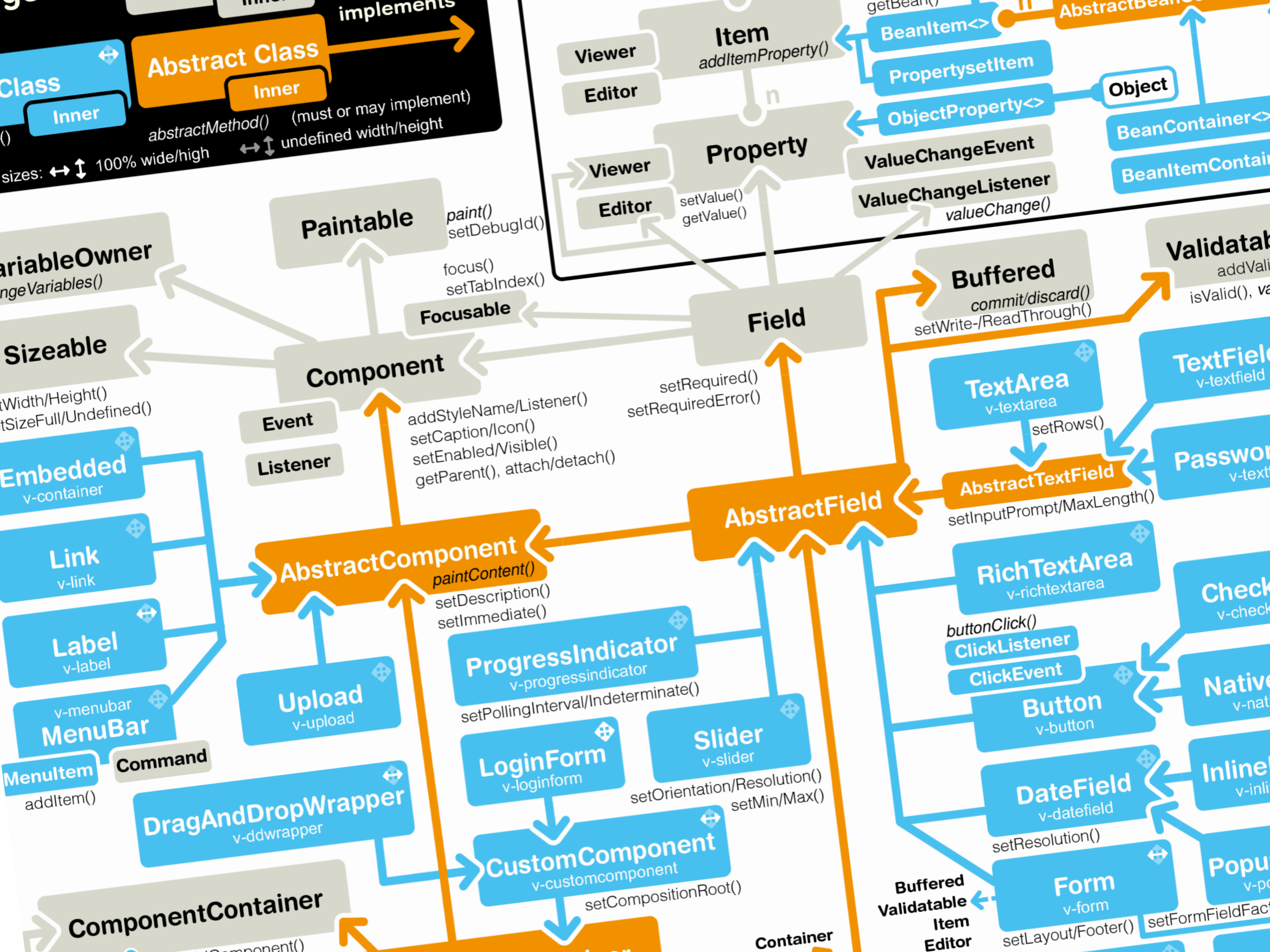
Figure 2: Architecture for Vaadin Applications



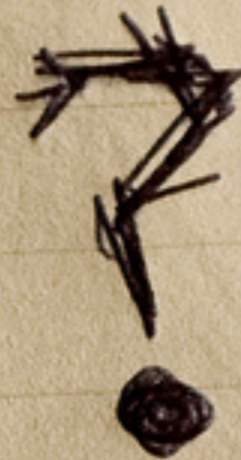
You can get a reference to the application object from any component attached to the application with `getApplication()`

Event Listeners

In the event-driven model, user interaction with user interface components triggers server-side events, which you can handle



Questions Comments



vaadin.com/joonas
[@joonaslehtinen](https://twitter.com/joonaslehtinen)
[#vaadin](https://hashtagger.com/hashtag/vaadin)

