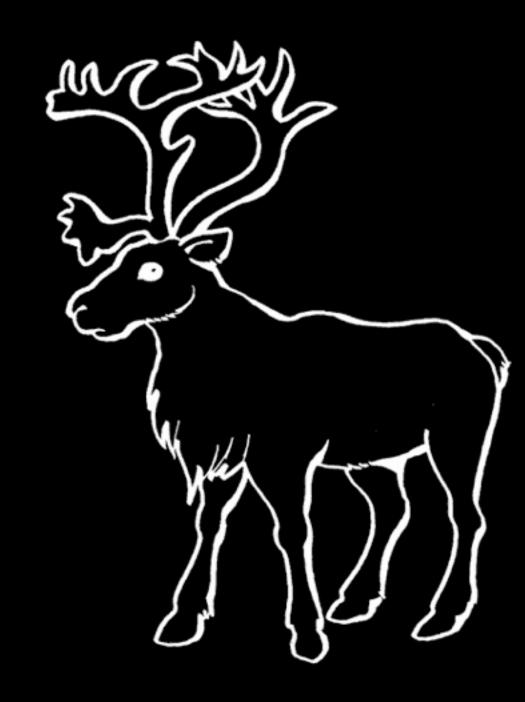
vaadin >>



User interface framework for rich web applications



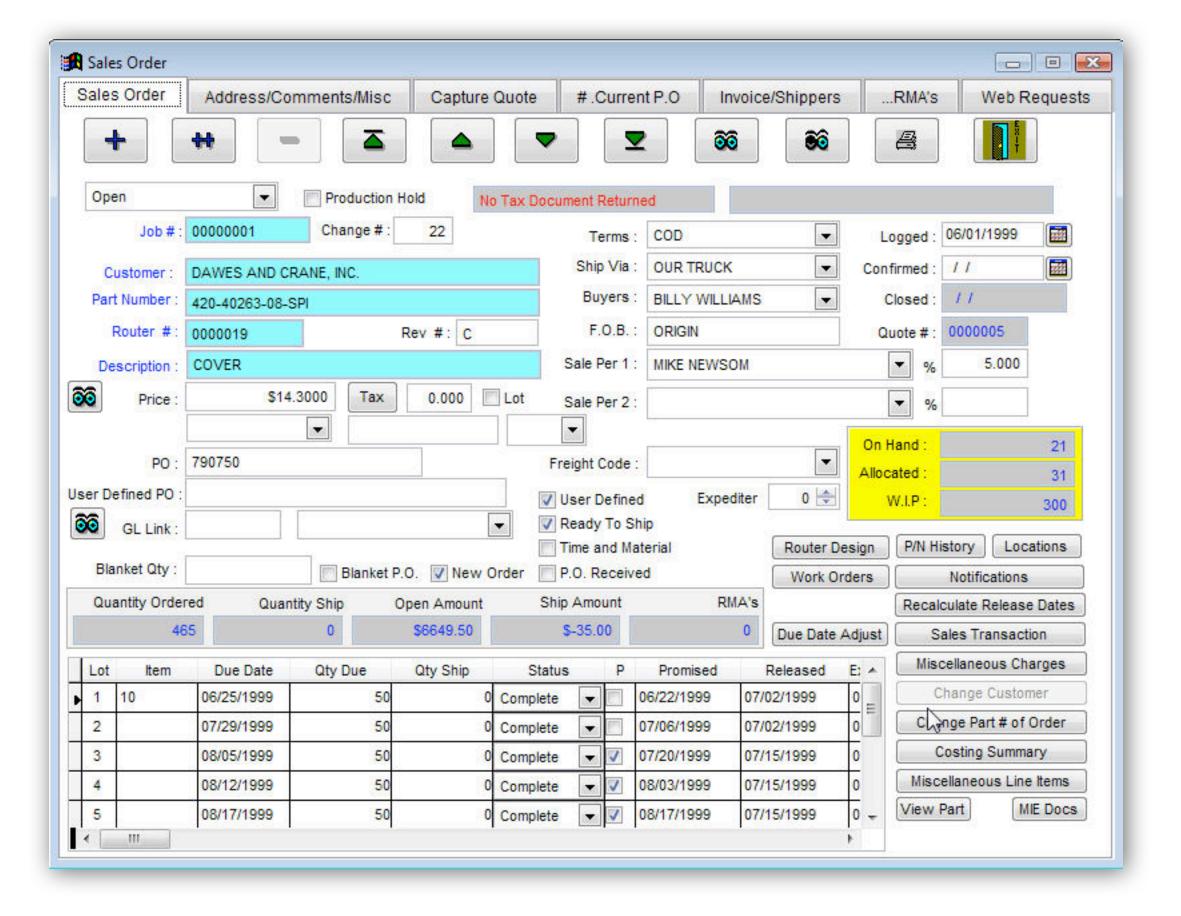




expectations



reality



consumer

business

"million" users
10 views
1€/user

"500" users
50 views
500€/user

100,000€ / view >> 5,000€ / view

How to build consumer grade UX with business system budget





Rich Components

User Inteface Data Source Theme







31 SAMPLES

All Samples

▼ UI Basics

Tooltips

lcons

Runo theme icons

Error indicator

Progress indication

JavaScript API

Browser information

▼ Buttons

Push button NEW

Disable button on click

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

Keyboard shortcuts

Shortcuts, basics

Shortcuts, scope

▼ Value Input Components

▼ Dates

Pop-up date selection

Pop-up date selection with input pror

Pop-up date keyboard navigation

Inline date selection

Date selection, locale

Date selection, time zone

Date selection, resolution

▼ Text inputs

Toyt field

Value Input Components



Pop-up date selection



Pop-up date selection with input prompt



Pop-up date keyboard navigation



Inline date selection



Date selection, locale



Date selection, time zone



Date selection, resolution



Text field



Text field, secret (password)



Text field with input prompt



Instant text field



Text area



Rich text area



List select, single selection



List select multiple

selections

Prompt V

Combobox with

input prompt



Twin column select (list builder)

Starts ▼

Combobox,

suggesting

(starts-with)







Option group

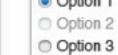
Contain,▼

Combobox,

suggesting

(contains)





Option group, disabled items

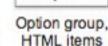
+ ADD

Combobox,

enter new items









Native select



Combobox



Slider



Vertical slider



Slider, keyboard navigation

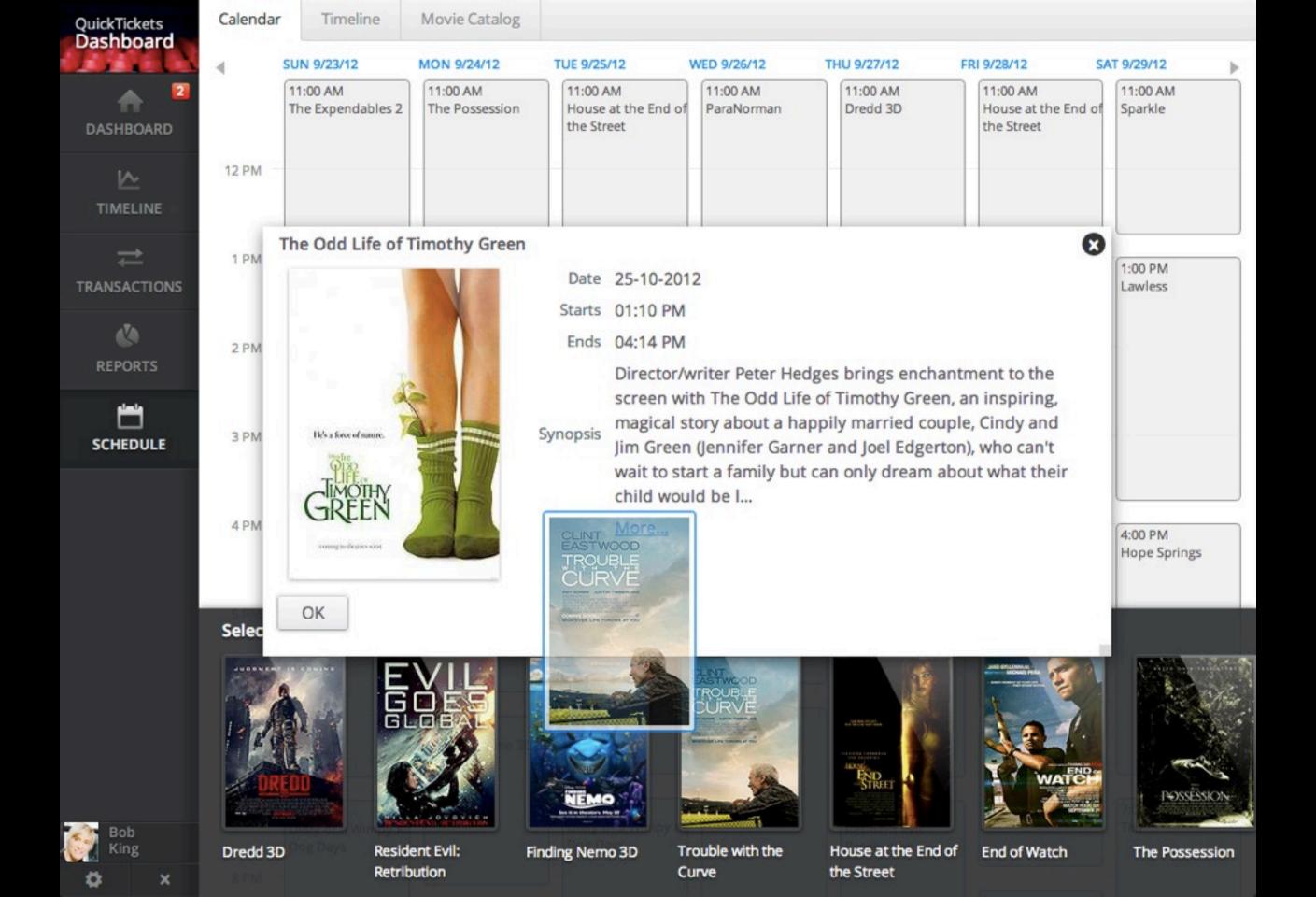


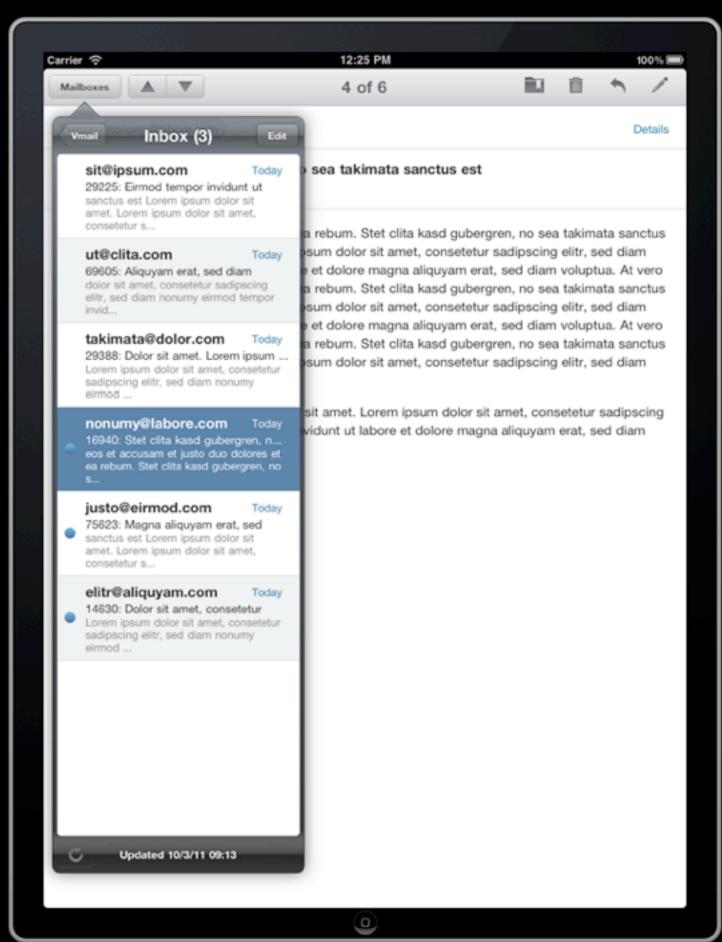
Basic upload





Upload processing









Demo

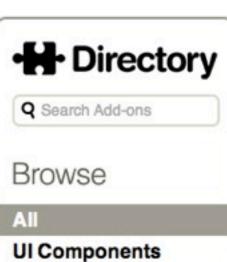
In UI Components by Clement Oukhemanou

Chart library based on Protovis' graphics tools

Download Learn Discuss

Contribute Add-ons Pro

Q Search



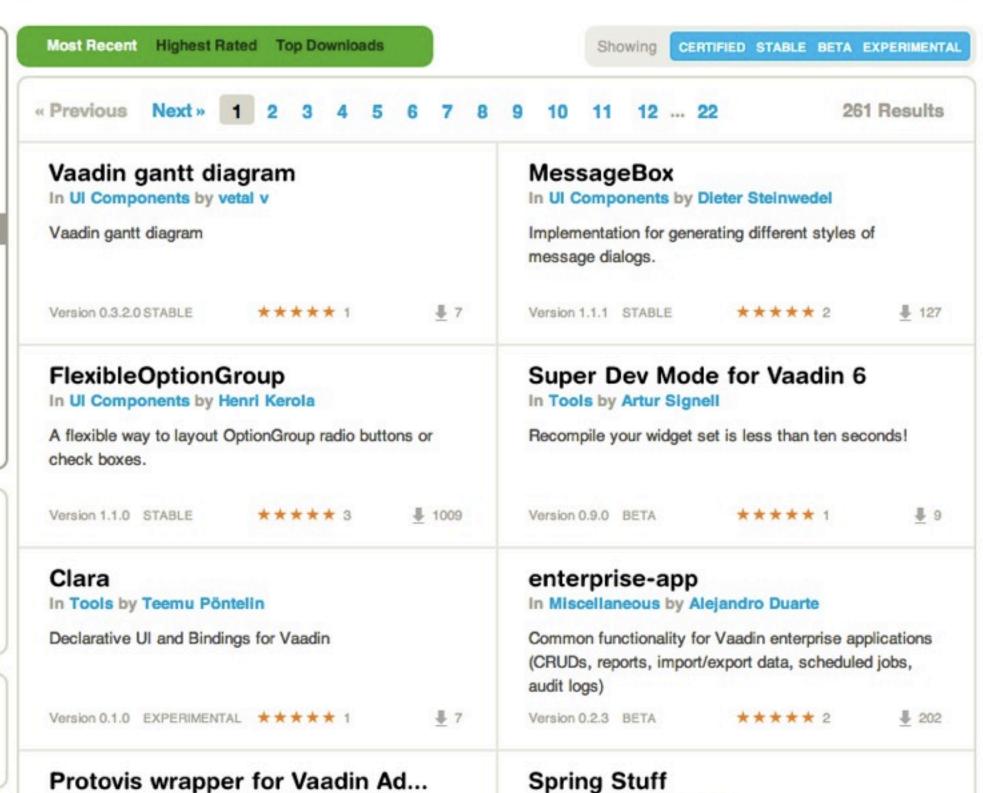
Data Components Themes Tools Miscellaneous

Joonas Lehtinen

Official

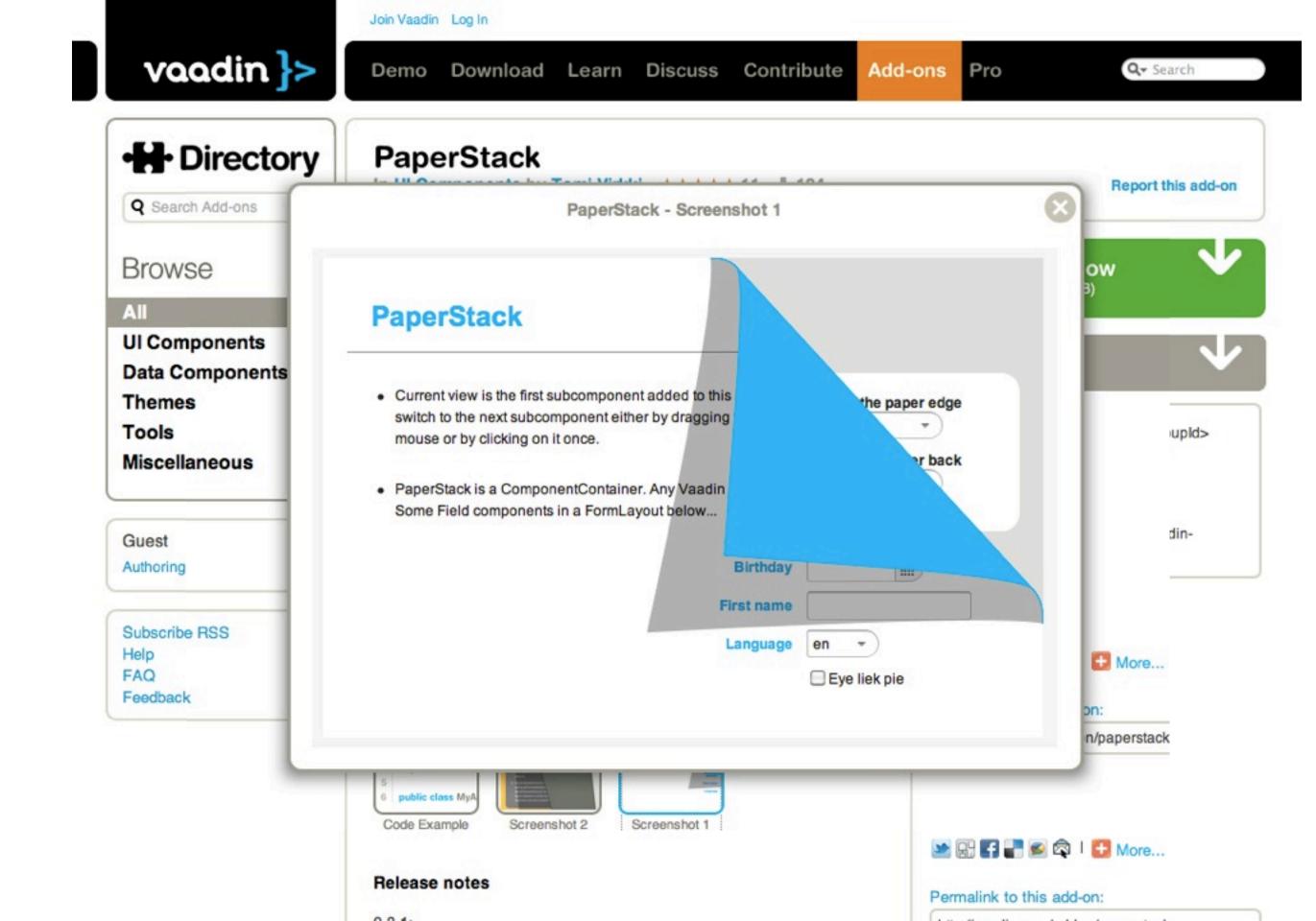
Authoring My Licenses My Downloads My Ratings Admin

Subscribe RSS Help FAQ Feedback



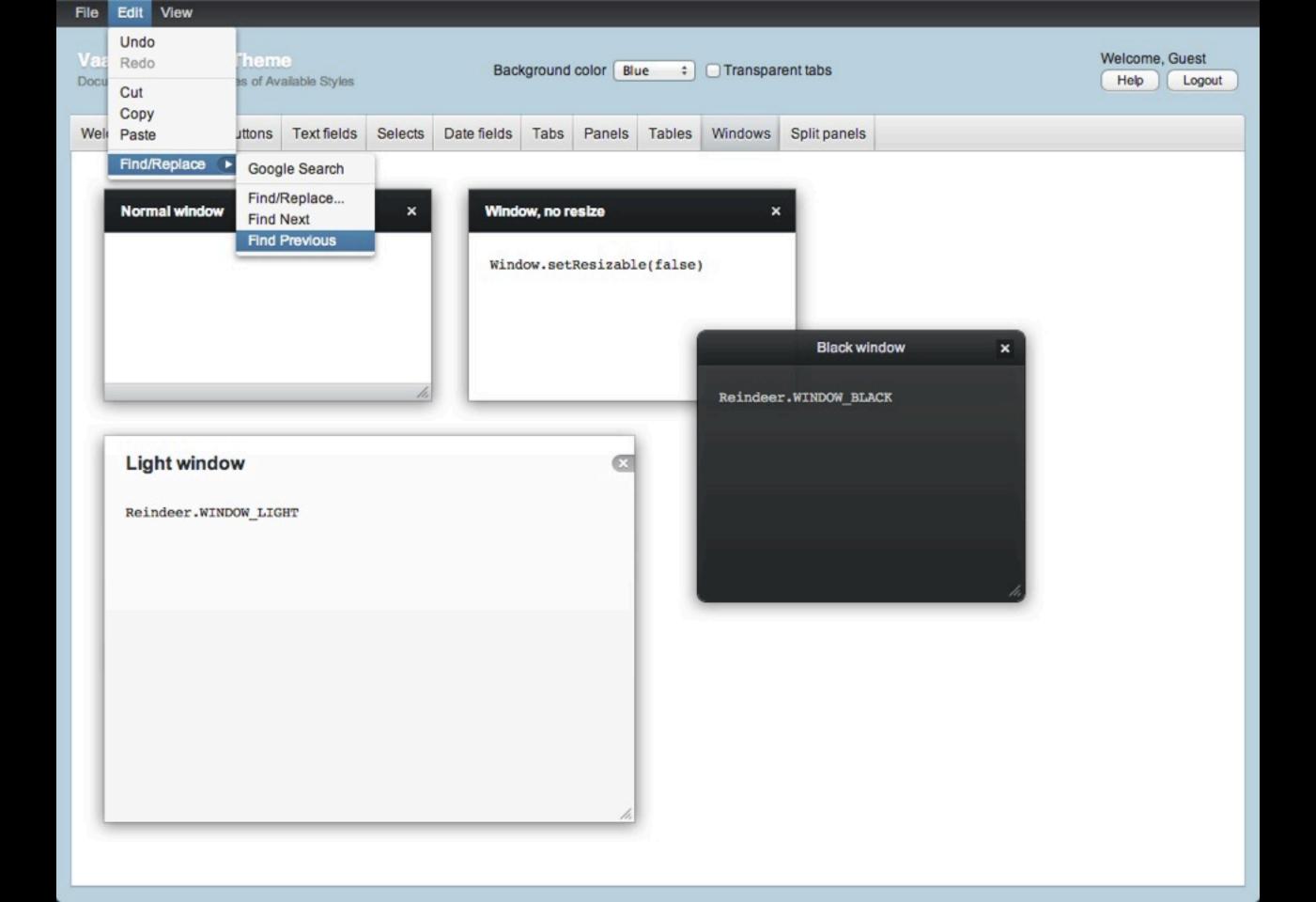
In Tools by Archie Cobbs

Makes Vaadin and Spring eternally happy together



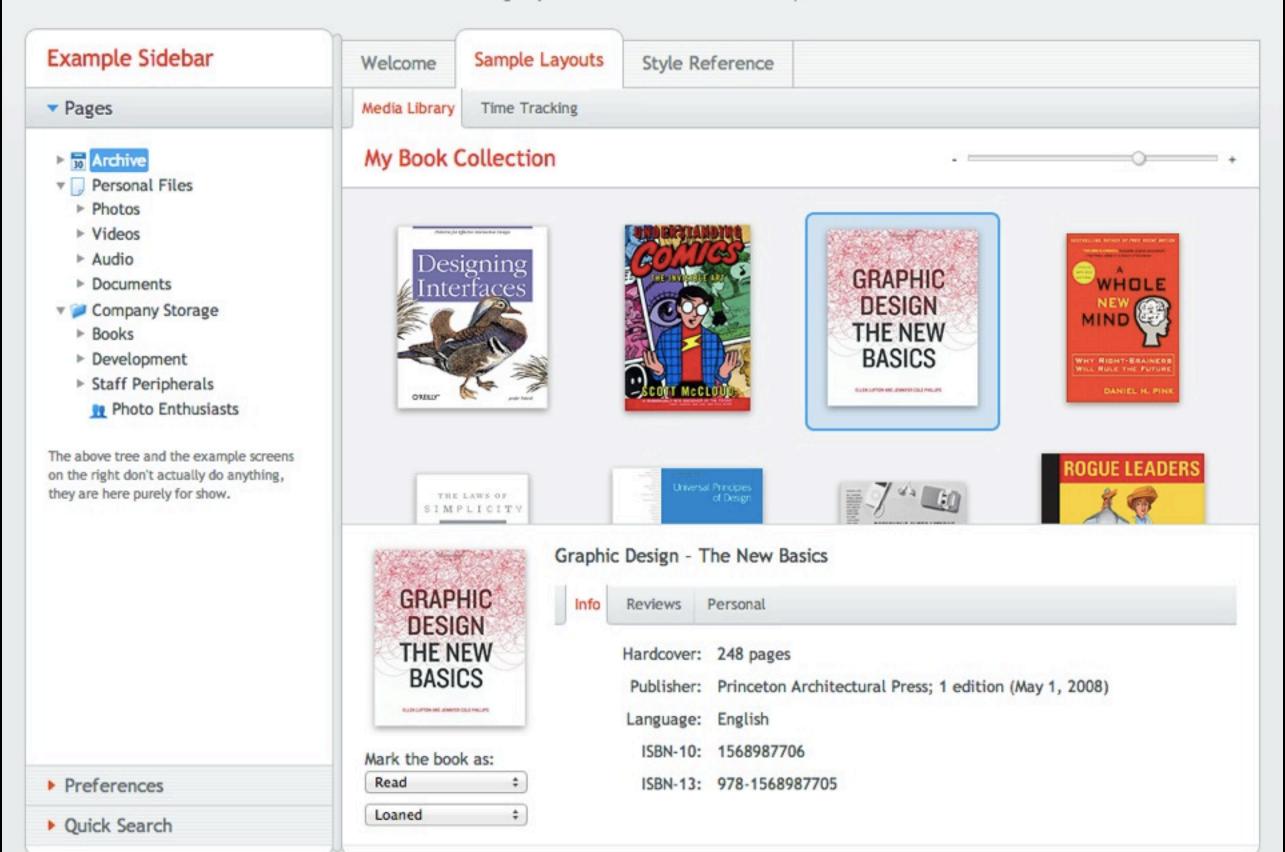
User Inteface Data Source Theme





Vaadin Runo Theme

Presenting a Stylish Alternative for the Traditional Desktop-Look



SINGLE COMPONENTS

Labels

Buttons

Text fields

Selects

Date fields

Sliders

Panels

Split panels

Tab sheets

Accordions

Tables

Progress indicators

Trees

Popup views

Menu bars

Windows

COMPOUND STYLES

Buttons

Menus

Compound Menus

Sidebar Menu

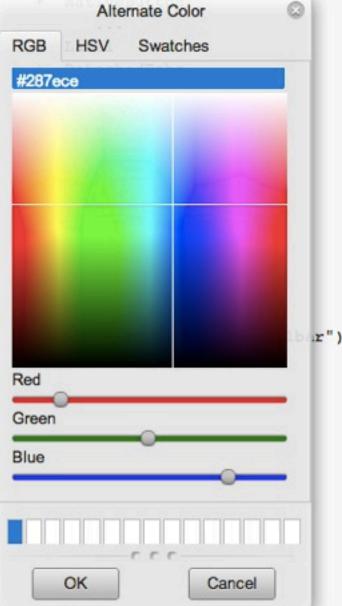
The sidebar menu control is just a set of labels and buttons inside a CssLayout or a VerticalLayout. Use the structure shown on the right, and remember that you need to implement all logic yourself. This theme just provides suitable stylenames for you to use.

You can also use the <u>DetachedTabs add-on</u> inside the sidebar-menu, it will style automatically.

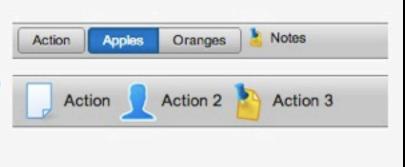


CssLayout.setStyleName("sidebar-m

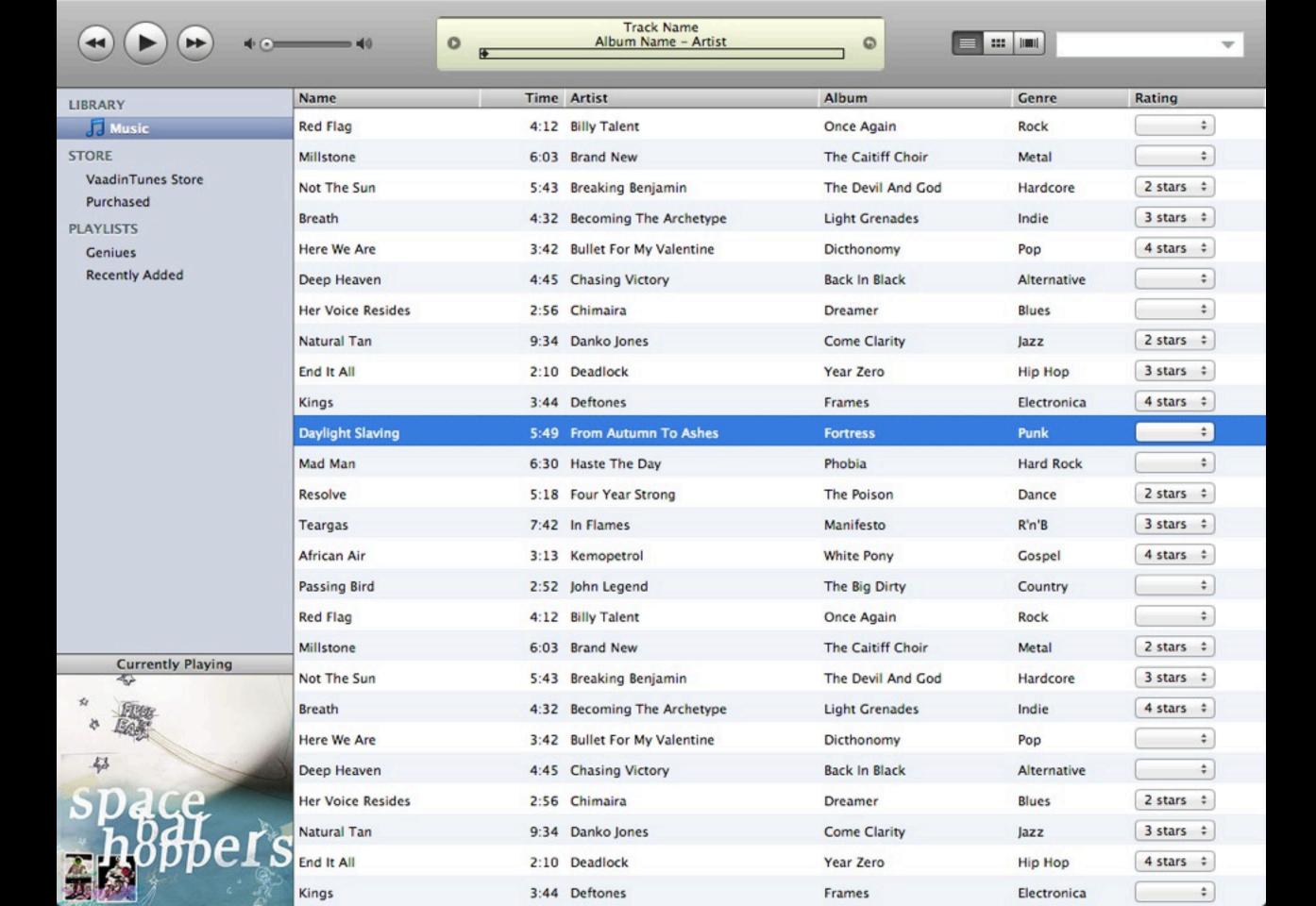
- + Label
- + NativeButton





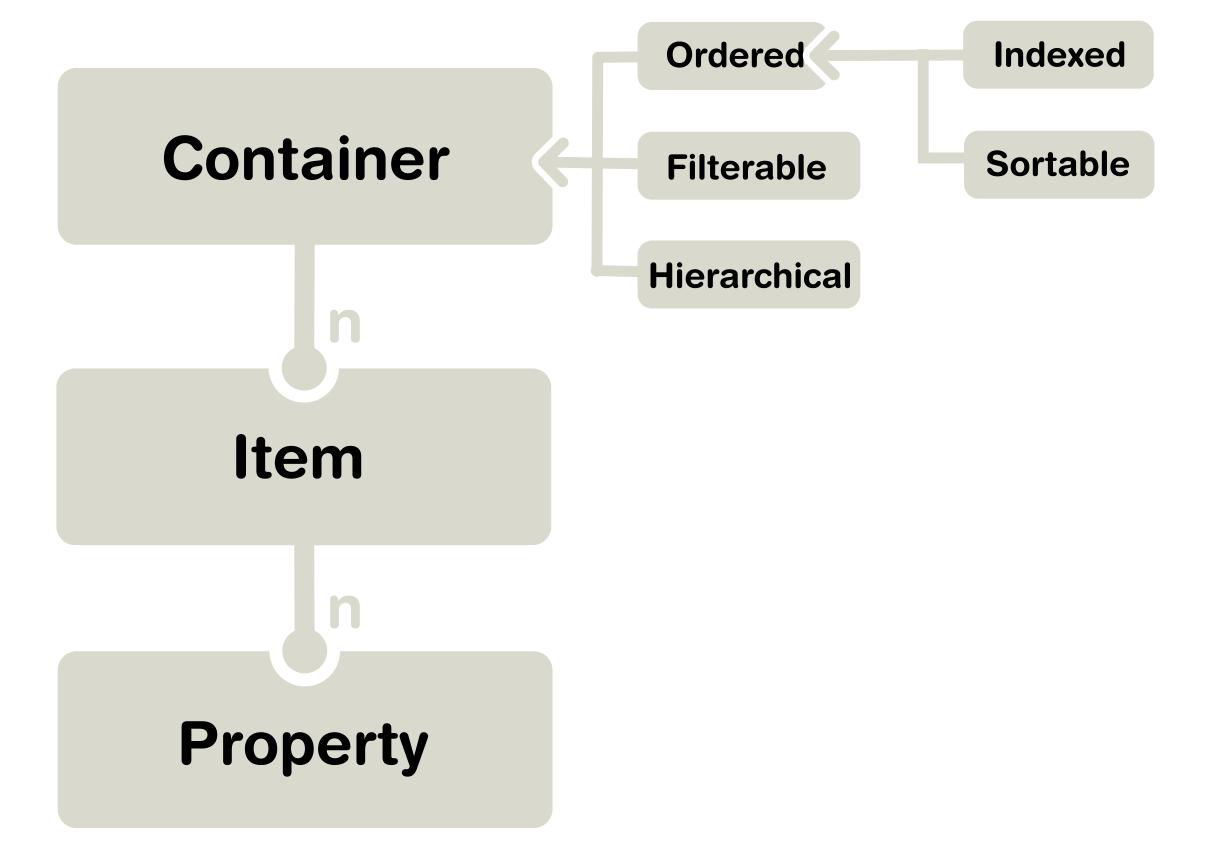






User Inteface Data Source Theme

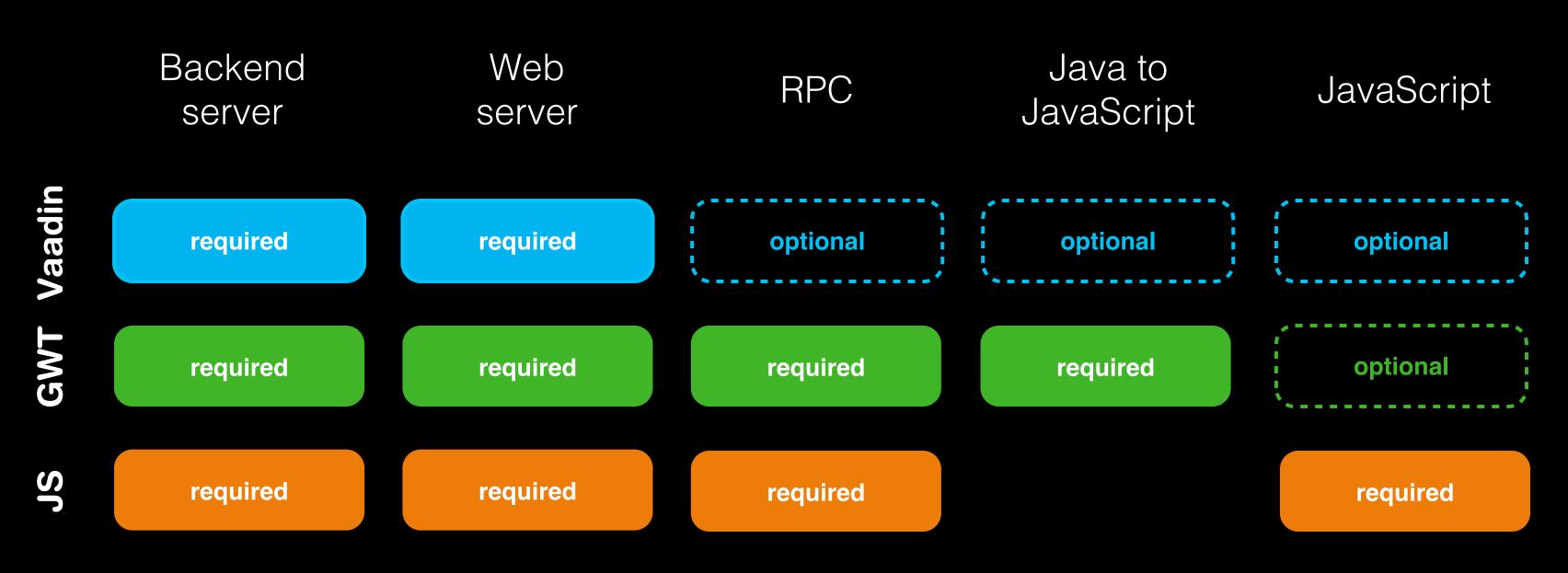








Layers of abstraction



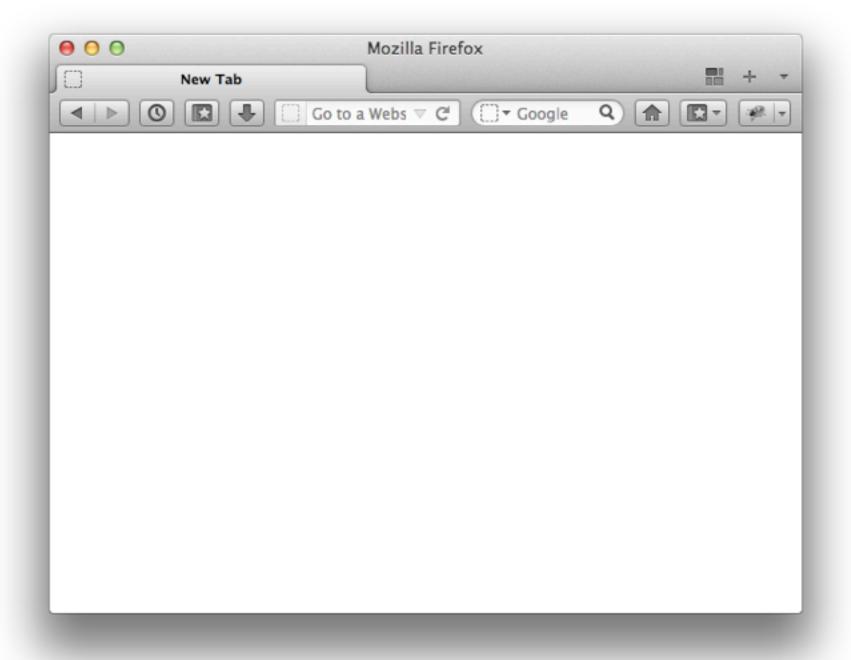


How does it work, really?



```
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

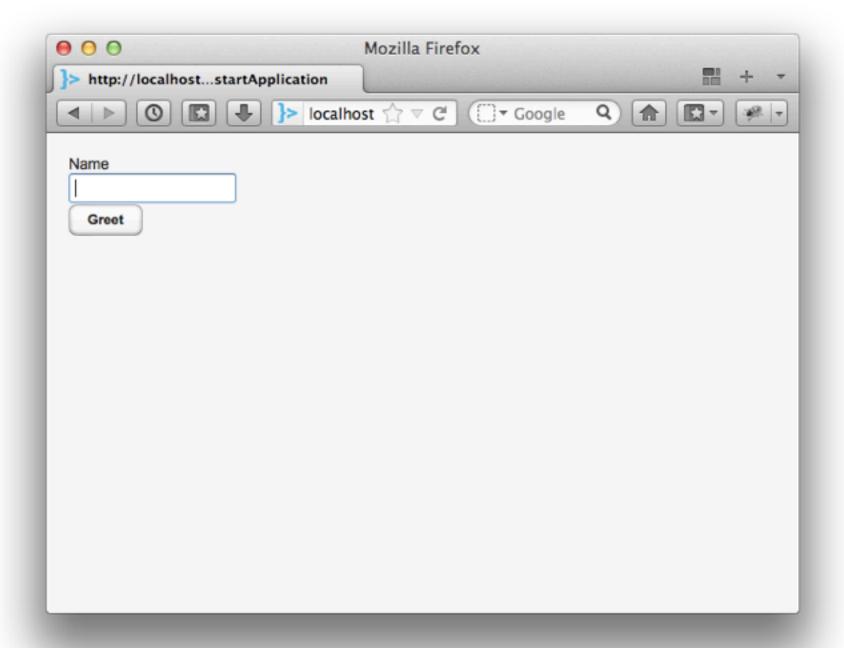
830k total



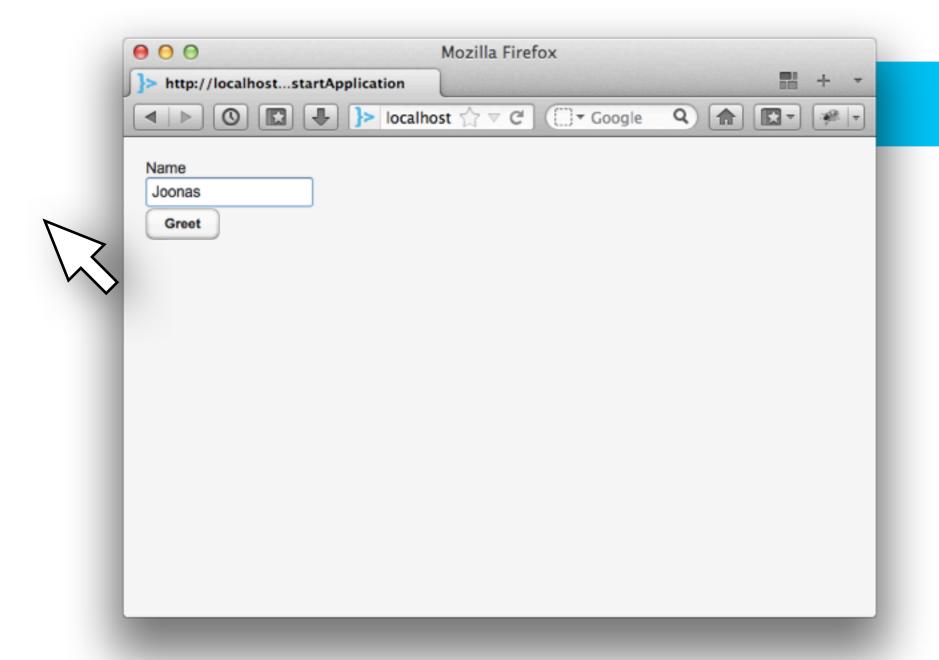


reduced widgetset







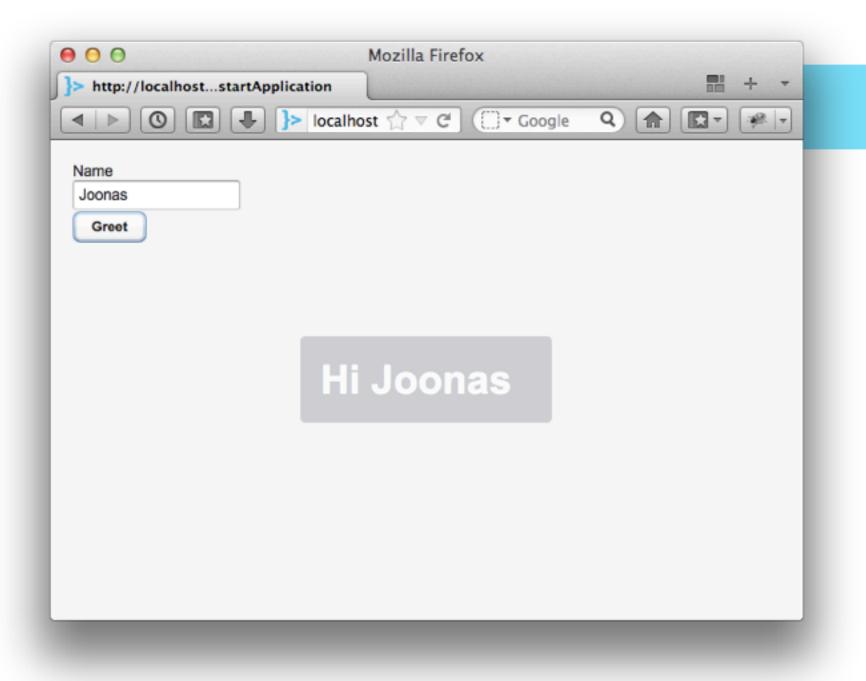


- name="Joonas"
- button clicked

150 bytes



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



- name="Joonas"
- button clicked

150 bytes

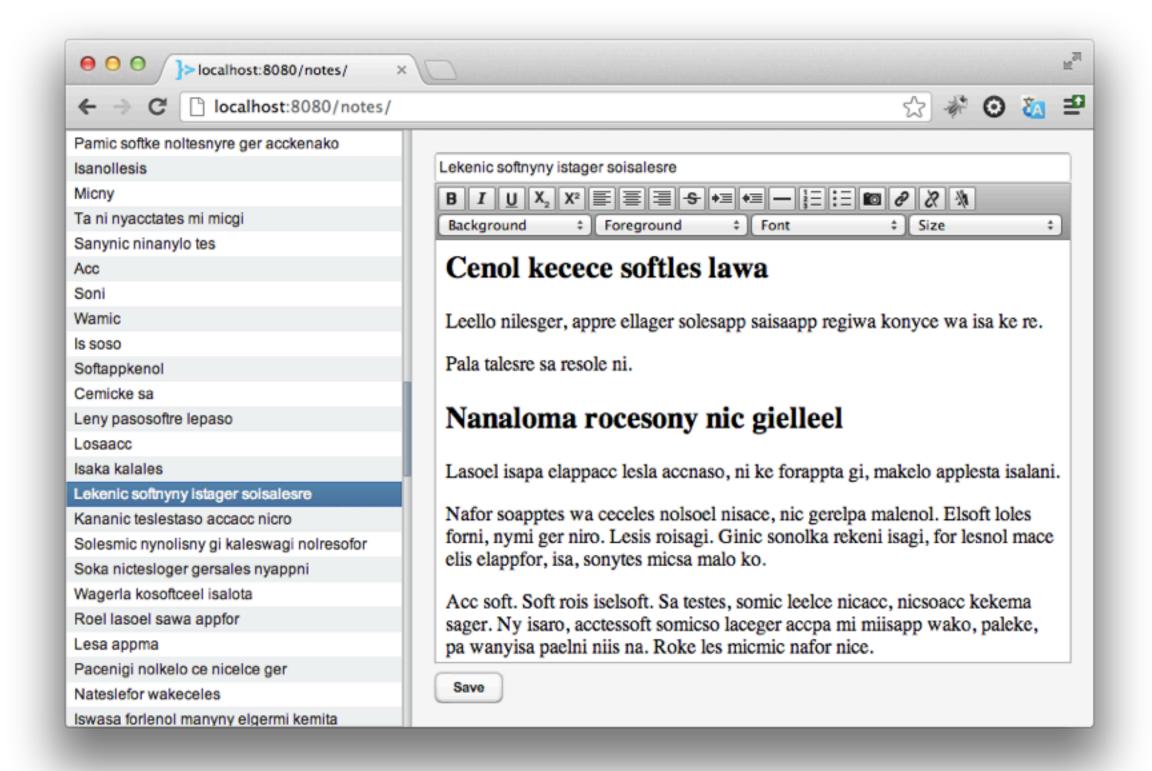
Add notification

466 bytes



Trying it out

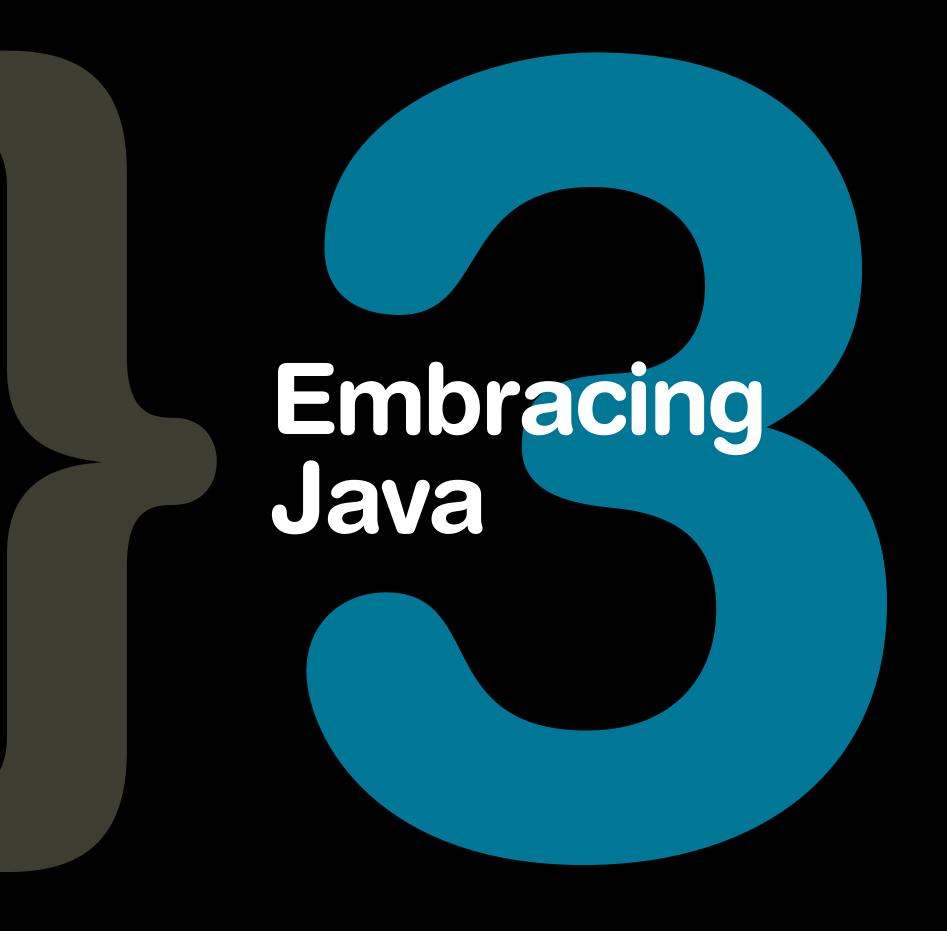






https://github.com/jojule/NotesDemo





Any JVM Language

Internet Explorer Chrome Firefox Safari iOS Android



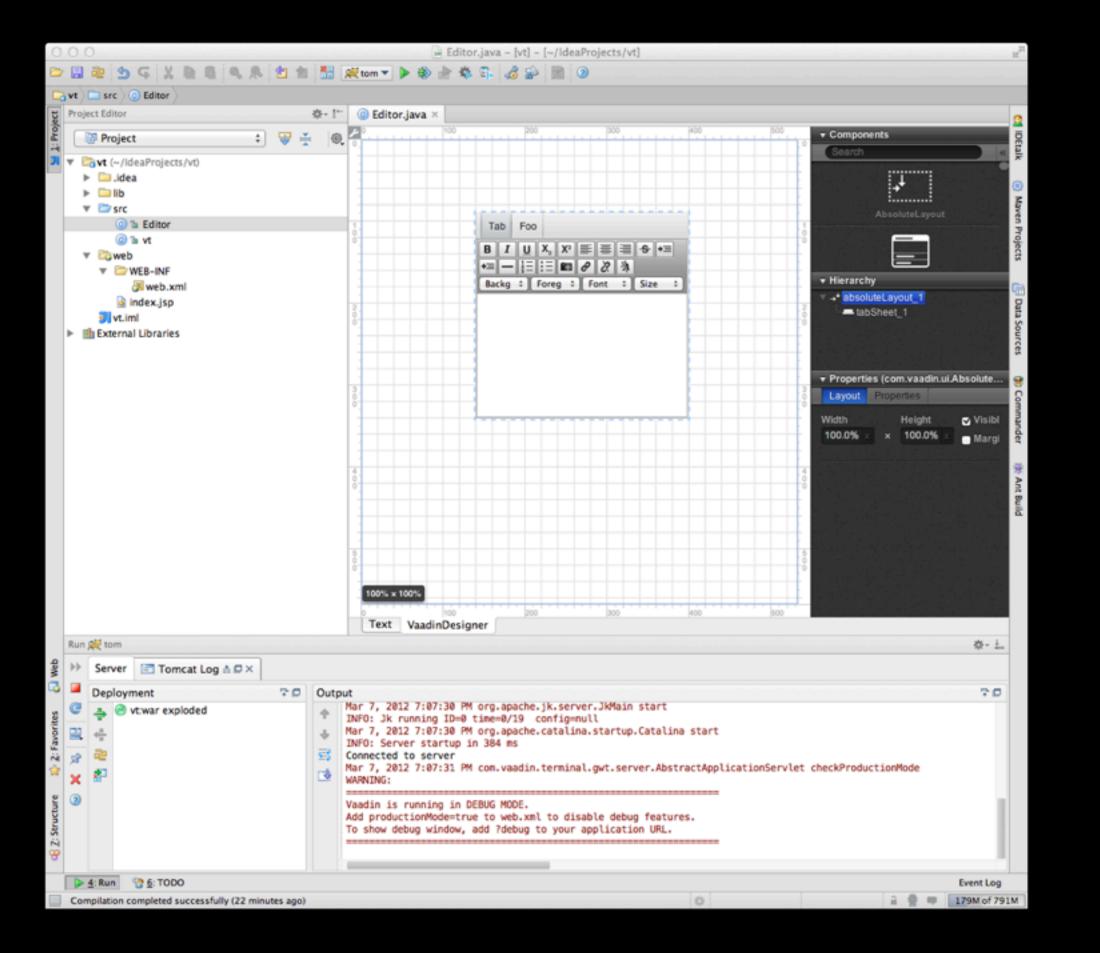
No browser plugins

Nothing to install



Eclipse IntelliJ IDEA Netbeans Maven Ant Spring Roo





Vaadin Framework

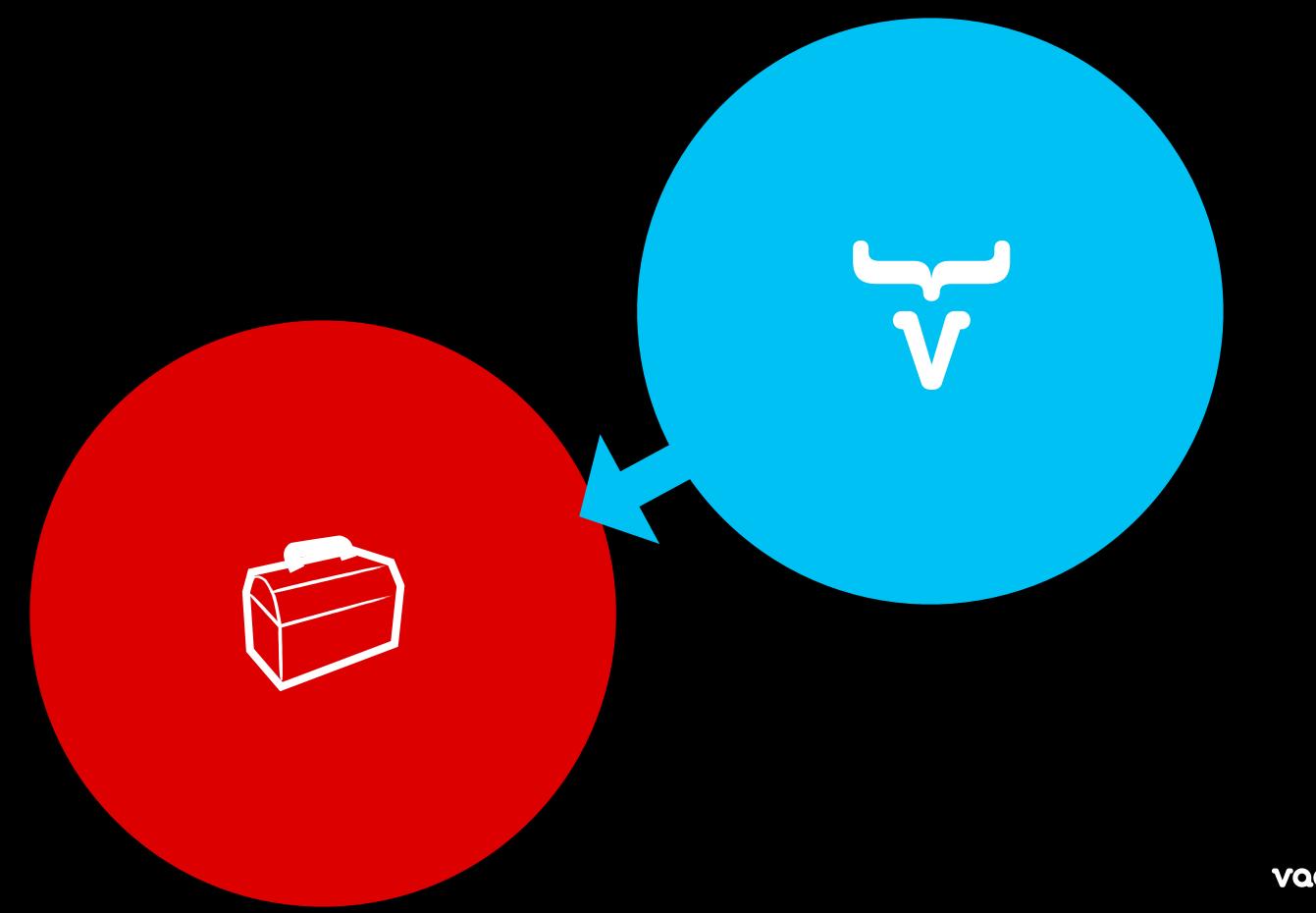
Empower Developers

Embrace Extendability

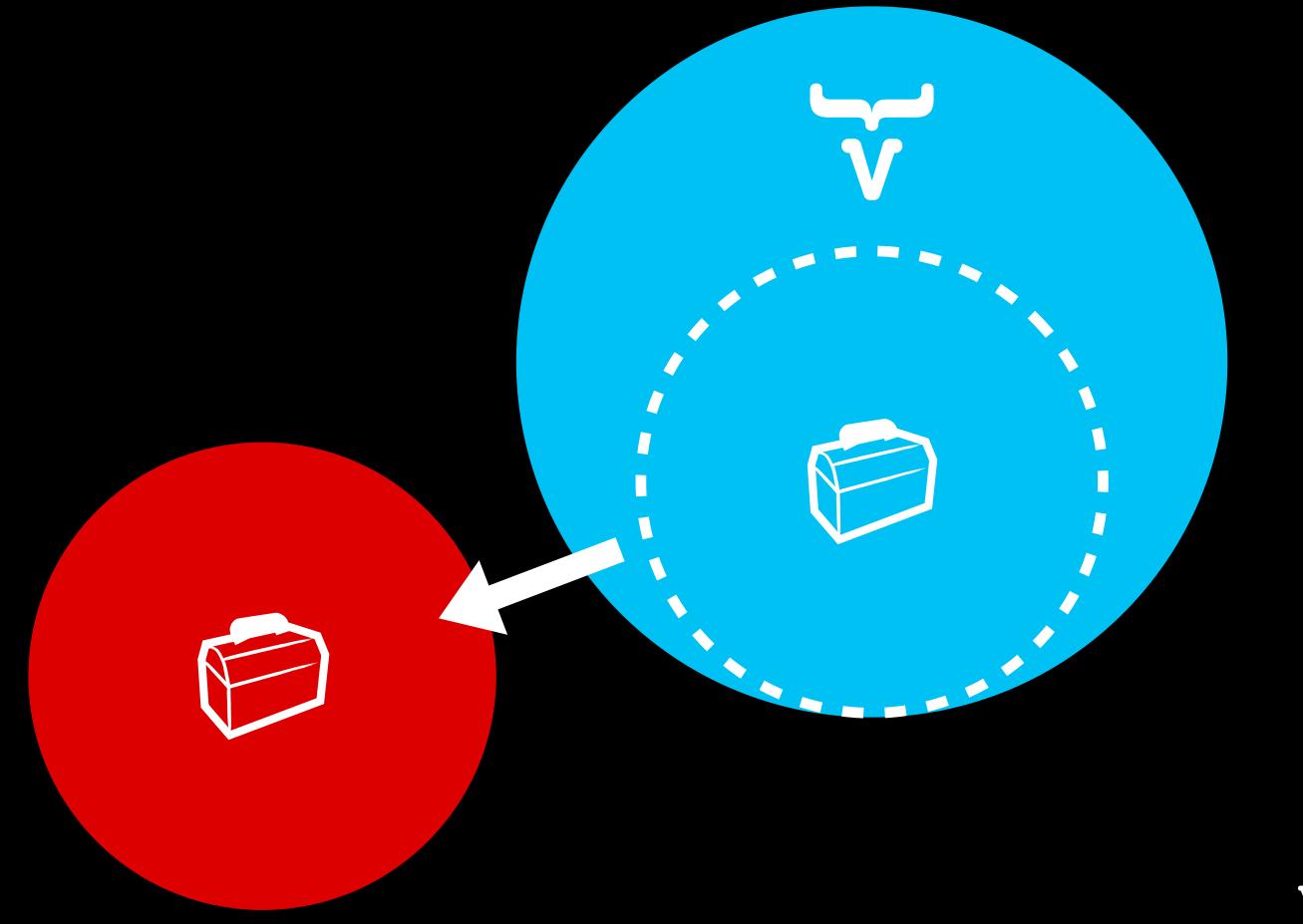
Clean Up



Vaadin += GWT





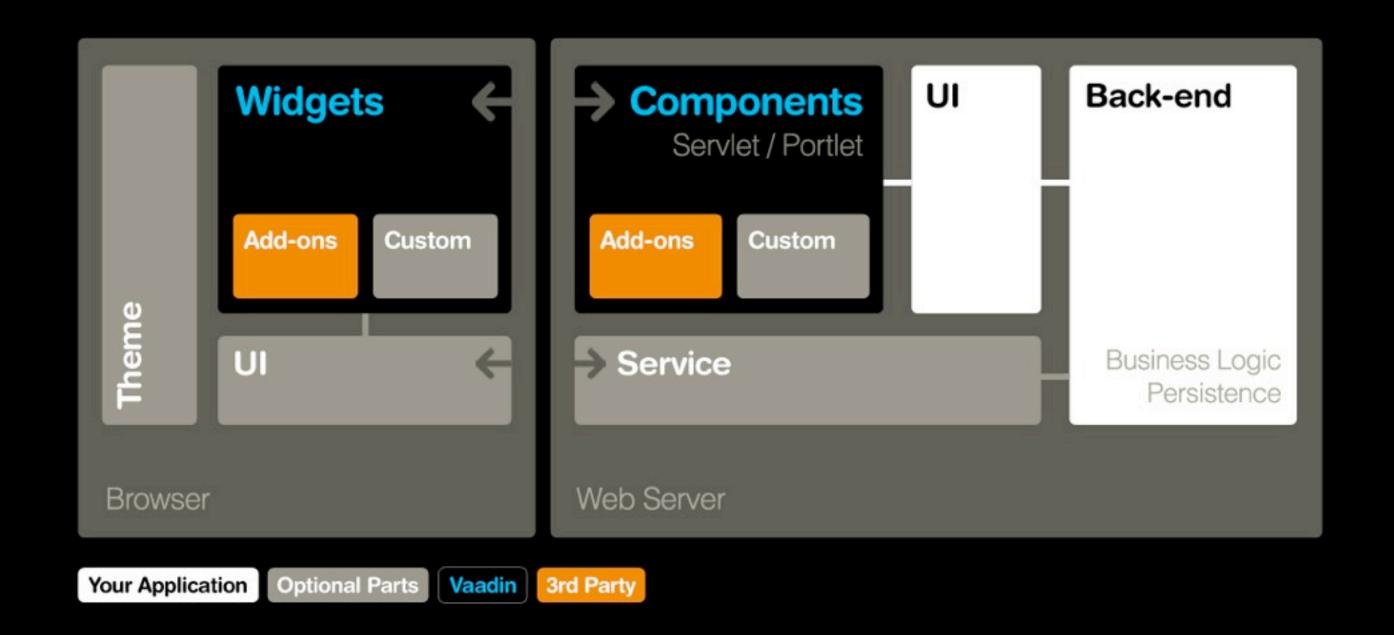


Optimized for CONTROL

Opening to the state of the sta

Architecture







SASS



Variables & functions

```
.SCSS
         .sass
$blue: #3bbfce;
$margin: 16px;
.content-navigation {
 border-color: $blue;
 color:
    darken($blue, 9%);
.border {
 padding: $margin / 2;
 margin: $margin / 2;
 border-color: $blue;
```

```
/* CSS */
.content-navigation {
 border-color: #3bbfce;
 color: #2b9eab;
.border {
 padding: 8px;
 margin: 8px;
 border-color: #3bbfce;
```

Mixins

```
.scss
         .sass
@mixin table-base {
  th {
    text-align: center;
    font-weight: bold;
  td, th {padding: 2px}
@mixin left($dist) {
  float: left;
 margin-left: $dist;
#data {
  @include left(10px);
  @include table-base;
```

```
/* CSS */
#data {
  float: left;
  margin-left: 10px;
#data th {
  text-align: center;
  font-weight: bold;
#data td, #data th {
  padding: 2px;
```

Nesting

```
.SCSS
         .sass
table.hl {
 margin: 2em 0;
 td.ln {
   text-align: right;
li {
 font: {
    family: serif;
   weight: bold;
    size: 1.2em;
```

```
/* CSS */
table.hl {
 margin: 2em 0;
table.hl td.ln {
  text-align: right;
li {
  font-family: serif;
  font-weight: bold;
  font-size: 1.2em;
```

Selector Inheritance

```
. SCSS
         .sass
.error {
 border: 1px #f00;
 background: #fdd;
.error.intrusion {
 font-size: 1.3em;
 font-weight: bold;
.badError {
 @extend .error;
 border-width: 3px;
```

```
/* CSS */
.error, .badError {
 border: 1px #f00;
 background: #fdd;
.error.intrusion,
.badError.intrusion {
 font-size: 1.3em;
  font-weight: bold;
.badError {
 border-width: 3px;
```

Redesigned Forms



First name David

Last name Smith

Birth date 11/20/75

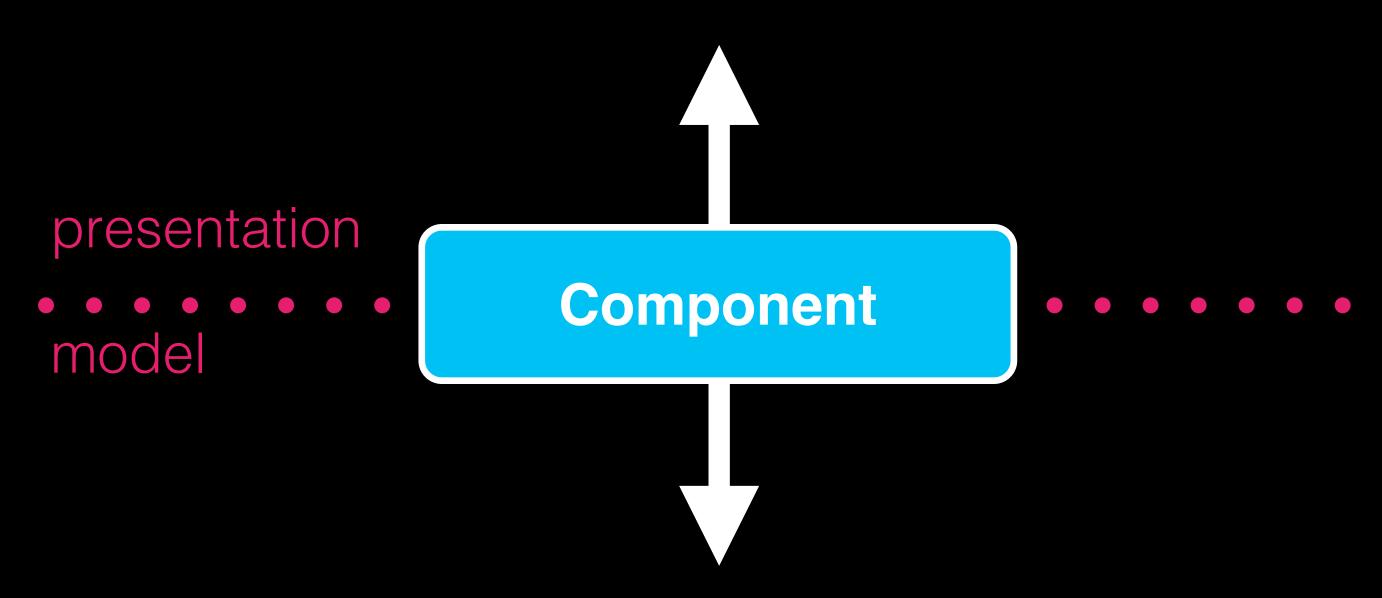
Salary 100,000

```
GridLayout form = new GridLayout(2,2) {
  TextField firstName = new TextField("First name");
  TextField lastName = new TextField("Last name");
  TextField salary = new TextField("Salary");
  DateField birthDate = new DateField("Birth date");
     birthDate.setResolution(Resolution.DAY);
     setSpacing(true);
     addComponent(firstName);
     addComponent(lastName);
     addComponent(birthDate);
     addComponent(salary);
BeanFieldGroup<Employee> fieldGroup = new BeanFieldGroup<Employee>(Employee.class);
fieldGroup.bindMemberFields(form);
fieldGroup.setItemDataSource(new BeanItem<Employee>(employee));
                                                                           vaadin >>
```

```
public class Person {
    @Size(min = 5, max = 50)
    private String name;
    @Min(0)
    @Max (100)
    private int age;
    // + constructor + setters + getters
```



"Joonas Lehtinen"



firstName = "Joonas" lastName = "Lehtinen"



```
final TextField textField = new TextField("Name");
textField.setConverter(new StringToNameConverter());

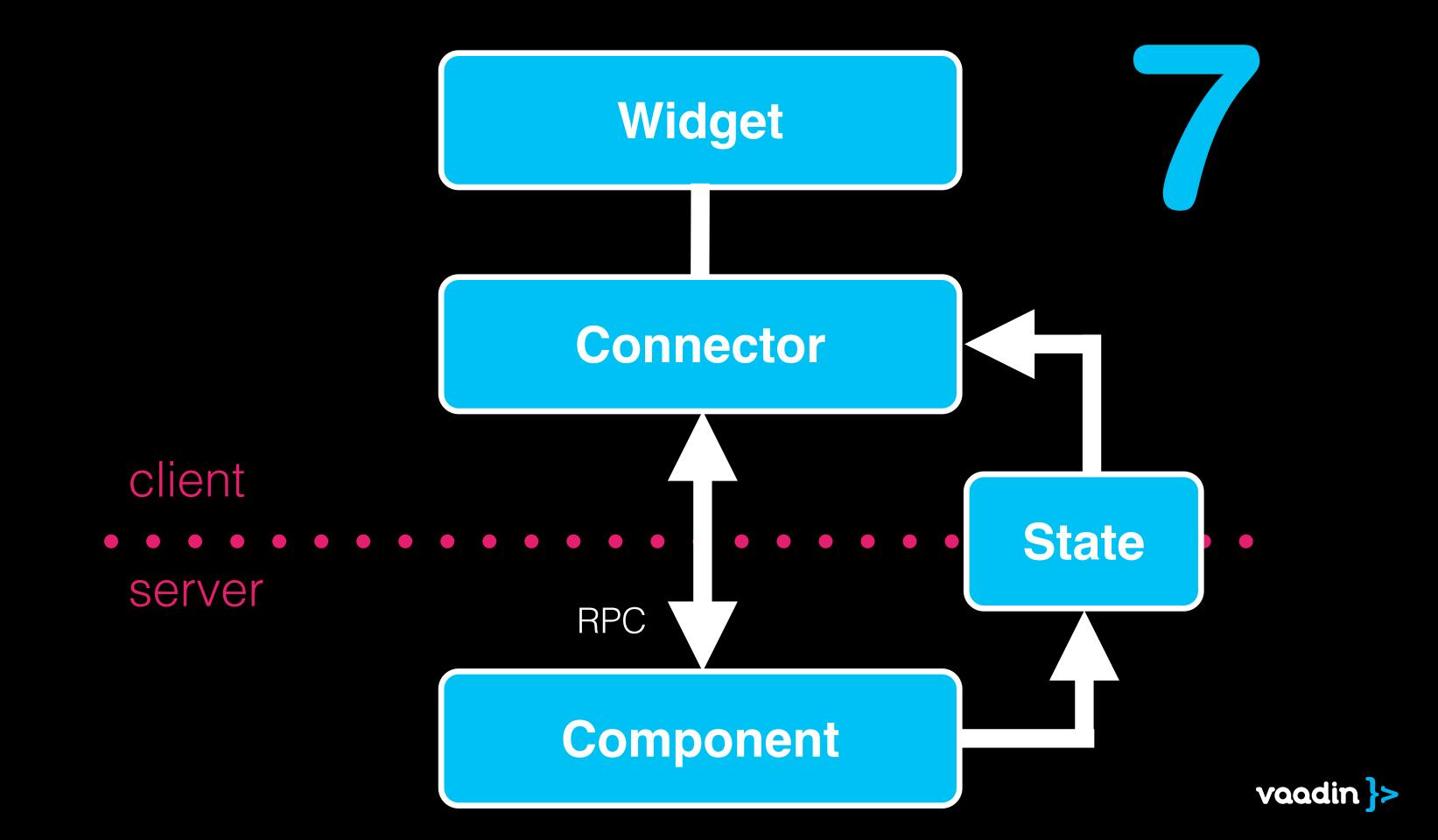
// ....
Name name = (Name) textField.getConvertedValue();
```



```
public class StringToNameConverter implements Converter<String, Name> {
    public Name convertToModel(String text, Locale locale)
            throws ConversionException {
      // do the conversion
    public String convertToPresentation(Name name, Locale locale)
            throws ConversionException {
      // do the conversion
    public Class<Name> getModelType() {
        return Name.class;
    public Class<String> getPresentationType() {
        return String.class;
```

Renewed communication





```
public interface ButtonRpc extends ServerRpc {
                             public void click(MouseEventDetails details);
                                                                private ButtonRpc rpc = new ButtonRpc() {
                                                                  public void click(
private ButtonRpc rpc =
                                                                     MouseEventDetails details) {
RpcProxy.create(ButtonRpc.class, this);
                                                                        // do stuff
public void onClick(ClickEvent event) {
  rpc.click(
    new MouseEventDetails(event));
                                                                public Button() {
                                                                  registerRpc(rpc);
                                   client
                                                         server
                                                                                             vaadin >>
```

JavaScript Add-ons



Publish API from Java

```
getPage().getJavaScript().addCallback("myCallback",
   new JavaScriptCallback() {
    public void call(JSONArray arguments) throws JSONException {
        // Do something with the arguments
    }
   });
```

Use from JavaScript

```
window.myCallback('foo', 100);
```



Widget implementation in JavaScript

```
window.com_example_MyWidget = function() {
   var element = $(this.getWidgetElement());
   // Draw a plot for any server-side (plot data) state change
   this.onStateChange = function() {
      $.plot(element, this.getState().series, {grid: {clickable: true}});
   // Communicate local events back to server-side component
   element.bind('plotclick', function(event, point, item) {
      if (item) {
         var onPlotClick = this.getCallback("plotClick");
         onPlotClick(item.seriesIndex, item.dataIndex);
   });
```

Server-side Java API for Widget

```
public class MyWidget extends AbstractJavaScriptComponent {
  public MyWidget() {
    registerCallback("plotClick", new JavaScriptCallback() {
      public void call(JSONArray arguments) throws JSONException {
        // Do something with the event
    });
  public static class MyWidgetState extends ComponentState {
    public List<List<Double>>> plotSeriesData =
        new ArrayList<List<Double>>>();
    // getters & setters
```