

Vaadin
today

@joonaslehtinen



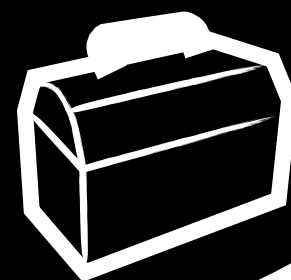
**Intro to
Vaadin**



```
new Label("Hello world")
```

Who is [using]
Vaadin?

New in
7





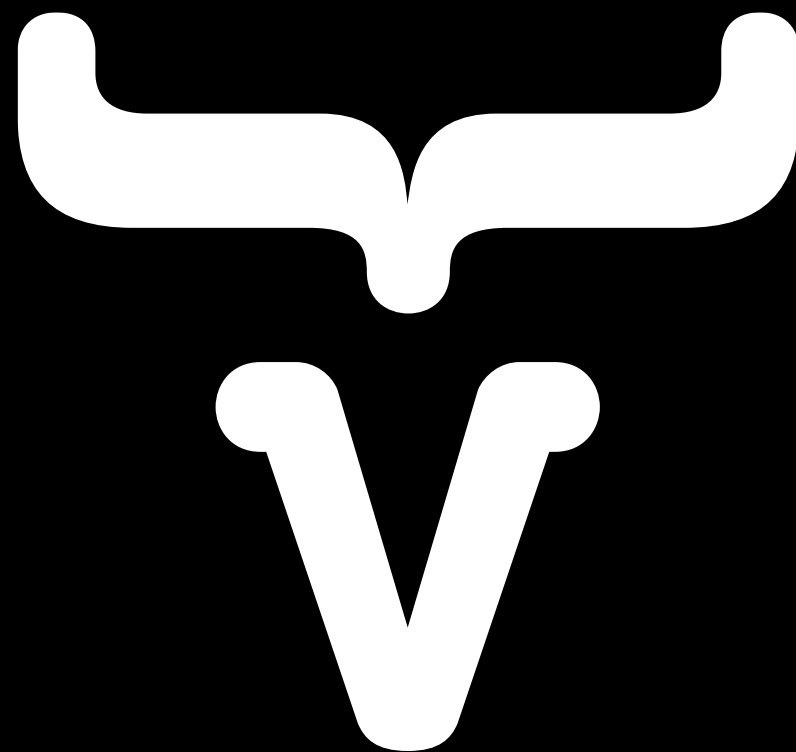
Getting
started

QA

vaadin }>



User interface framework for rich web applications



java } html >

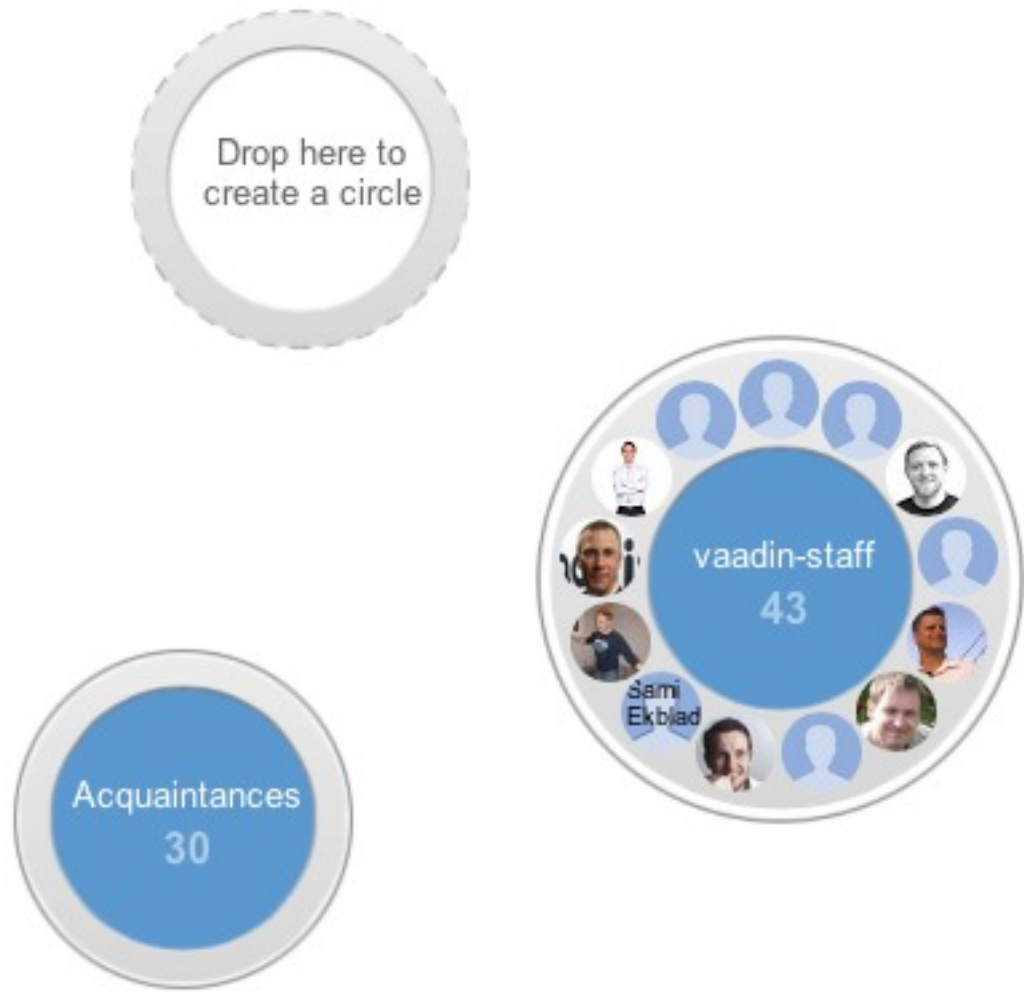
Why on earth?

enterpriseconsumers



Drop here to
create a circle





expectations

reality

Sales Order

Sales Order Address/Comments/Misc Capture Quote # .Current P.O Invoice/Shippers ...RMA's Web Requests

Open ☐ Production Hold No Tax Document Returned

Job #: 00000001 Change #: 22

Customer: DAWES AND CRANE, INC.

Part Number: 420-40263-08-SPI

Router #: 0000019 Rev #: C

Description: COVER

Price: \$14.3000 Tax 0.000 Lot

PO: 790750

User Defined PO:

GL Link:

Blanket Qty: Blanket P.O. ☒ New Order ☐ P.O. Received

Terms: COD

Ship Via: OUR TRUCK

Buyers: BILLY WILLIAMS

F.O.B.: ORIGIN

Sale Per 1: MIKE NEWSOM % 5.000

Sale Per 2: %

Logged: 06/01/1999

Confirmed: / /

Closed: / /

Quote #: 0000005

On Hand: 21

Allocated: 31

W.I.P.: 300

Router Design P/N History Locations

Work Orders Notifications

Recalculate Release Dates

Sales Transaction

Miscellaneous Charges

Change Customer

Change Part # of Order

Costing Summary

Miscellaneous Line Items

View Part MIE Docs

Quantity Ordered	Quantity Ship	Open Amount	Ship Amount	RMA's
465	0	\$8649.50	\$-35.00	0

Lot	Item	Due Date	Qty Due	Qty Ship	Status	P	Promised	Released	E
1	10	06/25/1999	50	0	Complete	<input type="checkbox"/>	06/22/1999	07/02/1999	0
2		07/29/1999	50	0	Complete	<input type="checkbox"/>	07/06/1999	07/02/1999	0
3		08/05/1999	50	0	Complete	<input checked="" type="checkbox"/>	07/20/1999	07/15/1999	0
4		08/12/1999	50	0	Complete	<input checked="" type="checkbox"/>	08/03/1999	07/15/1999	0
5		08/17/1999	50	0	Complete	<input checked="" type="checkbox"/>	08/17/1999	07/15/1999	0

consumer

“million” users

business

“500” users

100,000€ / view >> 5,000€ / view

Problem

How to build consumer
grade UX with business
system budget

How?

123

Key Ideas



Rich Components

User Interface

Data Source

Theme

[All Samples](#)

▼ UI Basics

Tooltips

Icons

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 prompt

Pop-up date keyboard navigation

Inline date selection

Date selection, locale

Date selection, time zone

Date selection, resolution

▼ Text inputs

Text field

Value Input Components

31 SAMPLES



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



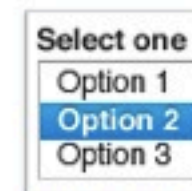
Instant text field



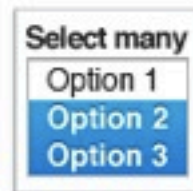
Text area



Rich text area



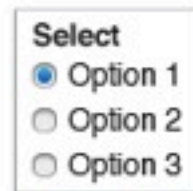
List select, single selection



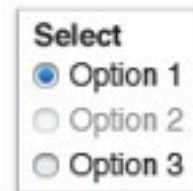
List select, multiple selections



Twin column select (list builder)



Option group



Option group, disabled items



Option group, HTML items



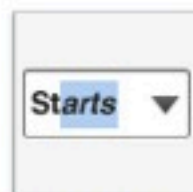
Native select



Combobox



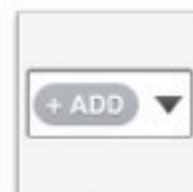
Combobox with input prompt



Combobox, suggesting (starts-with)



Combobox, suggesting (contains)



Combobox, enter new items



Slider



Vertical slider



Slider, keyboard navigation



Basic upload



Single-click upload



Upload processing





Edit Sample

Caption	
Description	
Enabled	<input checked="" type="checkbox"/>
Id	
Immediate	<input checked="" type="checkbox"/>
Read Only	<input type="checkbox"/>
Required	<input type="checkbox"/>
Style Name	
Tab Index	0
Visible	<input checked="" type="checkbox"/>
Value	The quick brown fox jumps
Icon	(none) ▾
Size	100% ▾ 100% ▾

Description

The RichTextArea allows 'rich' formatting of the input. Click the *Edit* button to edit the label content with the




RichTextArea.

Related samples

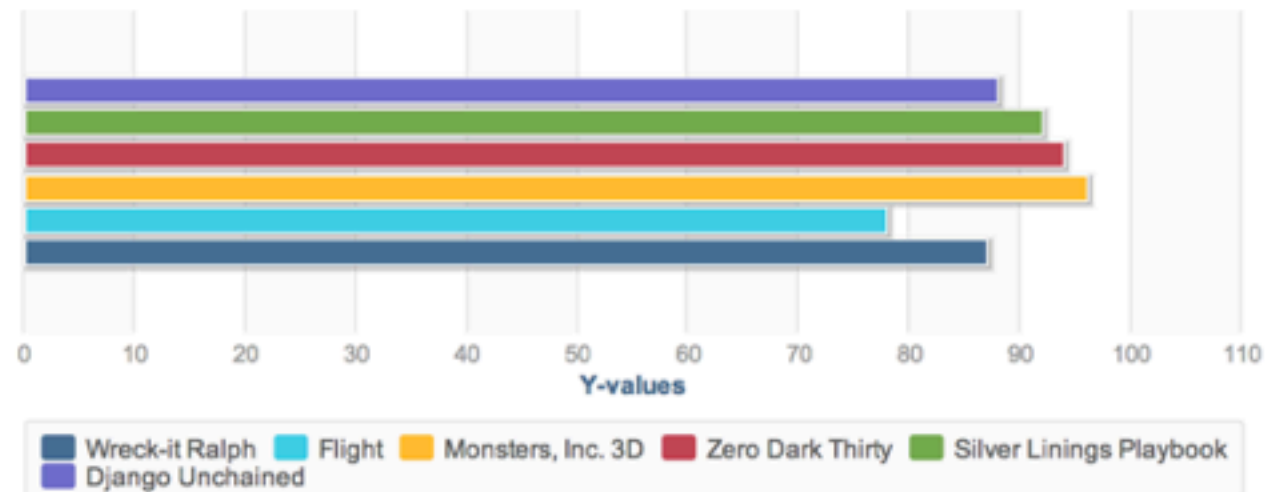
[Text area](#)

API

- [RichTextArea](#)
- [AbstractField](#)
- [AbstractComponent](#)
- [AbstractClientConnector](#)

**DASHBOARD**²SALESTRANSACTIONSREPORTSSCHEDULE

TOP GROSSING MOVIES



Vaadin Charts

NOTES

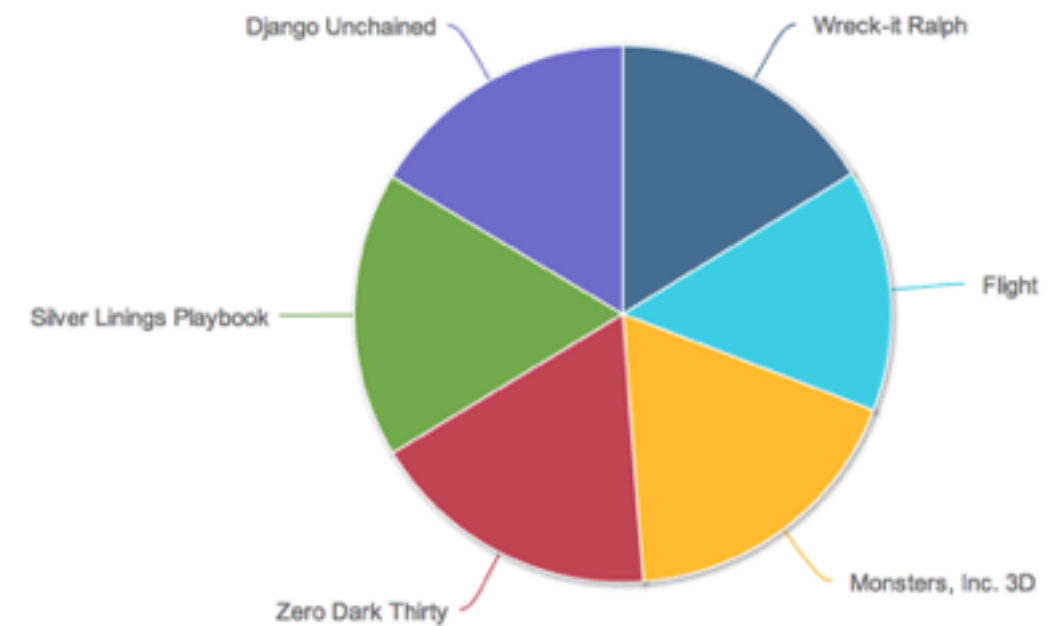
Remember to:

- Zoom in and out in the Sales view
- Filter the transactions and drag a set of them to the Reports tab
- Create a new report
- Change the schedule of the movie theater

TOP 10 TITLES BY REVENUE

	Title	Revenue
1	Skyfall	\$61266.4
2	Wreck-it Ralph	\$57261.37
3	Silver Linings Playbook	\$54930.16
4	Lincoln	\$54785.75
5	Life of Pi	\$49687.61
6	Flight	\$46322.71
7	Monsters, Inc. 3D	\$41018.66
8	Django Unchained	\$35166.89
9	Rise of the Guardians	\$34929.45
10	The Impossible	\$29322.2

POPULAR MOVIES



Vaadin Charts



SUN 2/3/13



12 PM

1 PM

2 PM



3 PM

4 PM

5 PM

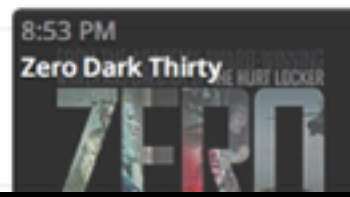
6 PM



7 PM

8 PM

9 PM



10 PM

MON 2/4/13



Les Miserables



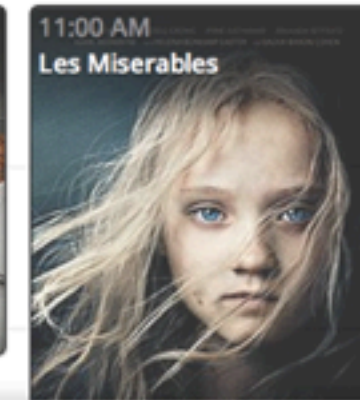
TUE 2/5/13



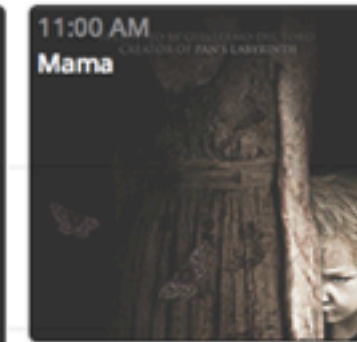
WED 2/6/13



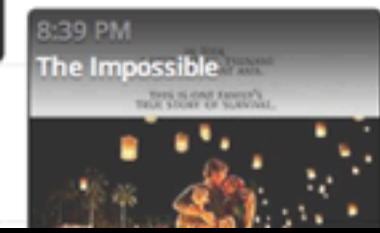
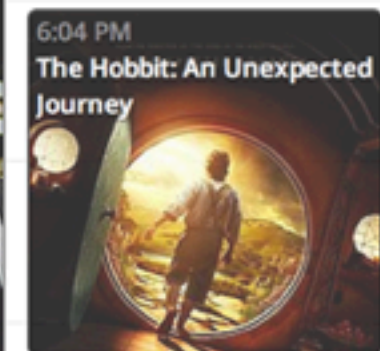
THU 2/7/13



FRI 2/8/13



SAT 2/9/13



Date 07-00-2013

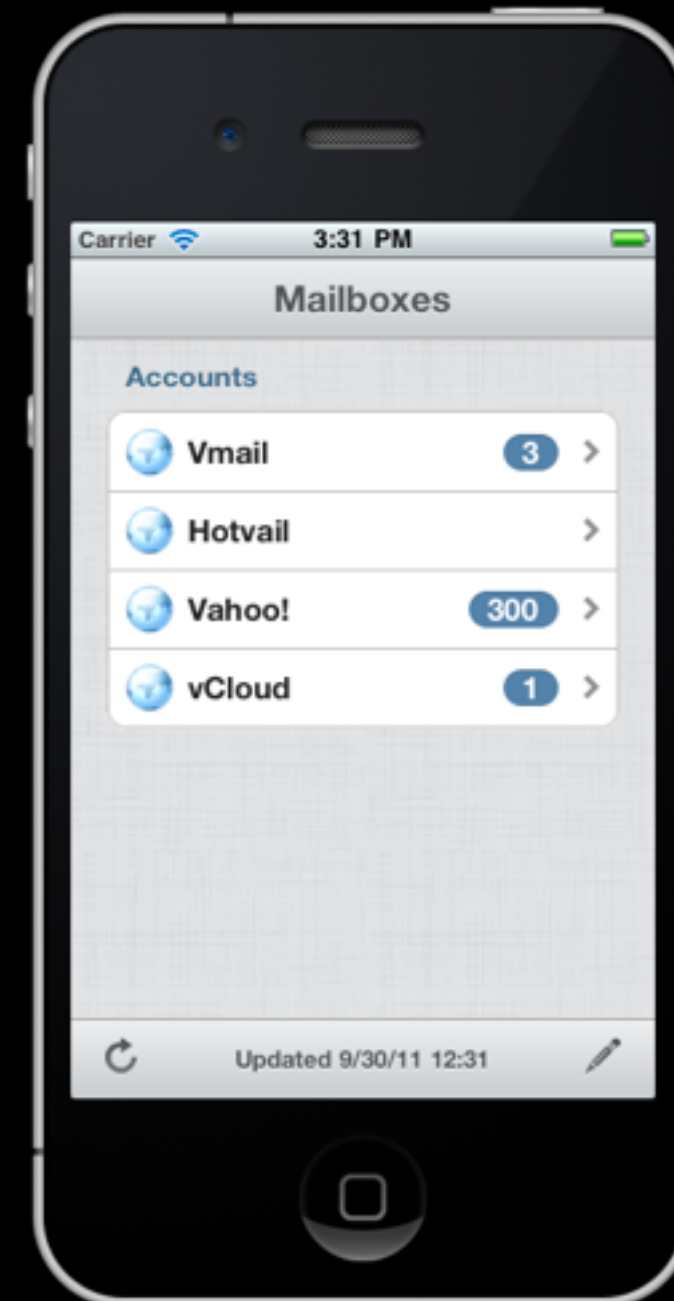
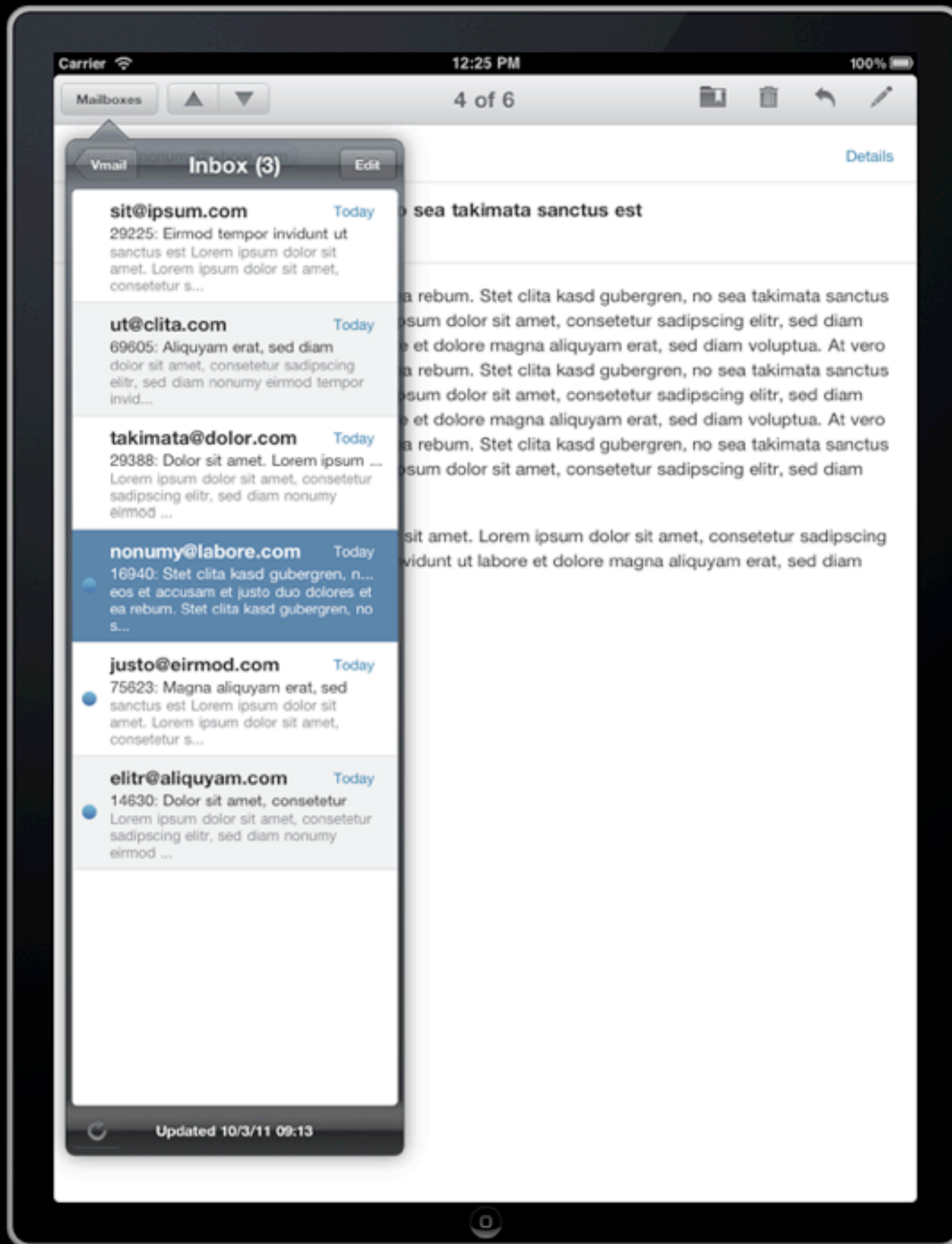
Starts 11:00 AM

Ends 01:50 PM

Duration 170 minutes

Synopsis Set against the backdrop of 19th-century France, Les Miserables tells an enthralling story of broken dreams and unrequited love, passion, sacrifice and redemption-a timeless testament to the survival of the human spirit. Jackman plays ex-prisoner Jean Valjean, hunted for decades by the ruthless policeman Javert (Crowe) after he breaks parole. When Valjean agrees to care for factory worker Fantine's (Hathaway) young daughter, Cosette, their lives change forever. In December 2012, the world's longest-running musical brings its power to the big screen in Tom Hooper's sweeping and spectacular interpretation of Victor Hugo's epic tale. -- (C) Universal

Close





Sign In [Register](#)

[Demo](#) [Learn](#) [Download](#) [Community](#) [Services](#) [Company](#)

[Releases](#) [Add-ons](#) [Tooling](#) [Roadmap](#)

Directory

All versions

Browse

All

UI Components

Data Components

Themes

Tools

Miscellaneous

Official

[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](#) ... [27](#)

316 Results

CSVValidation

In [UI Components](#) by [Marko Grönroos](#)

Client-side validation of text fields

Version 0.5.1 BETA

★★★★★ 3

↓ 1220

SimpleCalendar

In [UI Components](#) by [John Ahlroos](#)

Date selection made easy.

Version 0.1 EXPERIMENTAL No ratings yet

↓ 2

PopupButton

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

Button with popup

Version 2.2.0 BETA

★★★★★ 19

↓ 8546

Number Field

In [UI Components](#) by [Francesco Portus](#)

Numeric text field that allows insert only a valid numbers with keyboard and also allows spin value up/down.

Version 1.0.2 BETA

★★★★★ 2

↓ 99

Directory

All versions ▼

Browse

All

UI Components

Data Components

Themes

Tools

Miscellaneous

Official

Guest

[Authoring](#)

[Subscribe RSS](#)

[Help](#)

[FAQ](#)

[Feedback](#)

PaperStack

In **UI Components** by **Tomi Virkki** ★★★★★ 14 ⬇ 1119

[Report this add-on](#)

Version

2.0.0 ▼

Maturity

EXPERIMENTAL

License

[Apache License 2.0](#)

Vaadin

7.0+

[Available for 6](#)

Browser Compatibility



9

Overview

PaperStack is a component container whose subcomponents are presented sequentially, one subcomponent at a time. User can switch between the subcomponents by mouse dragging the upper right corner of a view revealing the underlying subcomponent simultaneously. The transition effect simulates leafing through a stack of papers.

Highlights



Screenshot 1



Screenshot 2



Code Example

Release notes

Download Now

Login required



Maven POM



Related Links

→ [Source Code \(0.8\)](#)

→ [Discussion Forum](#)

→ [Online Demo](#)

→ [Source Code \(2.0\)](#)

[Permalink to this add-on:](#)

<http://vaadin.com/addon/paperstack>

Directory

Search Add-ons

All versions

Browse

All

UI Components

Data Components

Themes

Tools

Miscellaneous

Official

Guest

Authoring

Subscribe RSS

Help

FAQ

Feedback

PaperStack

In [UI Components](#) by [Tomi Virkki](#) ★★★★★ 14 ⬇ 1119

[Report this add-on](#)

Version

2.0.0

Download Now



PaperStack - Screenshot 2



PaperStack

Clone elements

- The impressive clones of the first element.
- The resulting save using the background.
- Switch on element the clone-elements around.
- Clone element count using setCloneElement(cloneElementCount). Inc count results in a finer box naturally reduces rendering.

[Restart the demo](#)

Release notes

Directory

Search Add-ons

All versions

Browse

All

UI Components

Data Components

Themes

Tools

Miscellaneous

Official

Guest

[Authoring](#)

[Subscribe RSS](#)

[Help](#)

[FAQ](#)

[Feedback](#)

PaperStack

In [UI Components](#) by [Tomi Virkki](#) ★★★★★ 14 ⬇ 1119

[Report this add-on](#)

Version

2.0.0

Maturity

EXPERIMENTAL

License

[Apache License 2.0](#)

Vaadin

7.0+

[Available for 6](#)

Browser Compatibility



9

Download Now

Login required

Maven POM

Related Links

[Source Code \(0.8\)](#)

Overview

PaperStack is a
presented sequence
between the sub
a view revealing
transition effect

Highlights



Screenshot 1



Screenshot 2



Code Example

Release notes

PaperStack - Code Example

```
PaperStack paperStack = new PaperStack();

paperStack.addComponent(new Label("Hello!"));

InlineDateField inlineDateField = new InlineDateField();
inlineDateField.setResolution(DateField.RESOLUTION_DAY);
paperStack.addComponent(inlineDateField, "#999");

mainWindow.addComponent(paperStack);
```

paperstack

Directory

All versions

Browse

All

UI Components

Data Components

Themes

Tools

Miscellaneous

Official

Joonas Lehtinen

[Authoring](#)

[My Licenses](#)

[My Downloads](#)

[My Ratings](#)

[Admin](#)

[Subscribe RSS](#)

[Help](#)

[FAQ](#)

[Feedback](#)

PaperStack

In [UI Components](#) by [Tomi Virkki](#) ★★★★★ 14 ⬇ 1119

[Report this add-on](#)

Version

2.0.0

Maturity

EXPERIMENTAL

License

[Apache License 2.0](#)

Vaadin

7.0+

[Available for 6](#)

Browser Compatibility



9

Overview

PaperStack is a component container whose subcomponents are presented sequentially, one subcomponent at a time. User can switch between the subcomponents by mouse dragging the upper right corner of a view revealing the underlying subcomponent simultaneously. The transition effect simulates leafing through a stack of papers.

Highlights



Screenshot 1



Screenshot 2



Code Example

Release notes

Download Now

Version 2.0.0 (45 kB)



Maven POM



```
<dependency>
  <groupId>org.vaadin.virkki</groupId>
  <artifactId>paperstack</artifactId>
  <version>2.0.0</version>
</dependency>
```

```
<repository>
  <id>vaadin-addons</id>
  <url>http://maven.vaadin.com/vaadin-
  addons</url>
</repository>
```

Related Links

➔ [Source Code \(0.8\)](#)

➔ [Discussion Forum](#)

➔ [Online Demo](#)

➔ [Source Code \(2.0\)](#)

My Rating Rated Oct 13, 2010 7:17 PM



This add-on is like a wow effect in a can. Open and drop to your app for an instant coolness.

Remove Rating

User Interface
Data Source
Theme

Normal window

x

Window, no resize

x

```
Window.setResizable(false)
```

Black window

x

```
Reindeer.WINDOW_BLACK
```

Light window

x





```
Reindeer.WINDOW_LIGHT
```


Vaadin Runo Theme

Presenting a Stylish Alternative for the Traditional Desktop-Look

Example Sidebar

▼ Pages

- ▶  Archive
- ▼  Personal Files
 - ▶ Photos
 - ▶ Videos
 - ▶ Audio
 - ▶ Documents
- ▼  Company Storage
 - ▶ Books
 - ▶ Development
 - ▶ Staff Peripherals
 - ▶  Photo Enthusiasts

The above tree and the example screens on the right don't actually do anything, they are here purely for show.

▶ Preferences

▶ Quick Search

Welcome

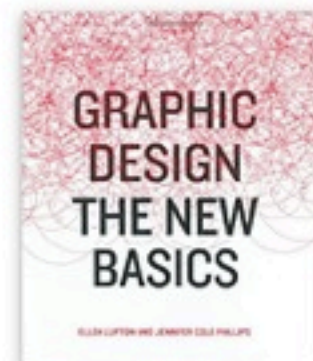
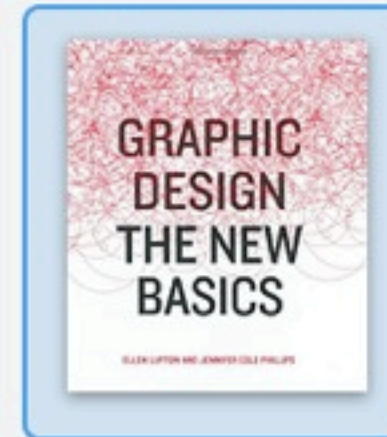
Sample Layouts

Style Reference

Media Library

Time Tracking

My Book Collection



Graphic Design - The New Basics

Info

Reviews

Personal

Hardcover: 248 pages

Publisher: Princeton Architectural Press; 1 edition (May 1, 2008)

Language: English

ISBN-10: 1568987706

ISBN-13: 978-1568987705

Mark the book as:

Read

Loaned

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 [DetachedTabs add-on](#) inside the sidebar-menu, it will style automatically.

```
CssLayout.setStyleName("sidebar-menu")
+ Label
+ NativeButton
```

Edit Colors

Note: only native buttons are styled inside the menu, normal buttons are left untouched.

Preset

Blue

Background Color

All

Base Font

13px

Base Color

Alternate Color

Toolbar is a simple layout with a styleName. It provides a background and a little padding for its contents. Normally you will want to put buttons inside it, but segment controls fit in nicely as well.

Alternate Color

RGB HSV Swatches

#287ece

Red

Green

Blue

OK Cancel

FRUITS

Apples

Oranges

Bananas

Grapes

VEGETABLES

Tomatoes

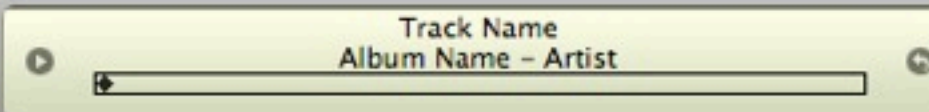
Cabbages

Potatoes

Carrots

Action Apples Oranges Notes

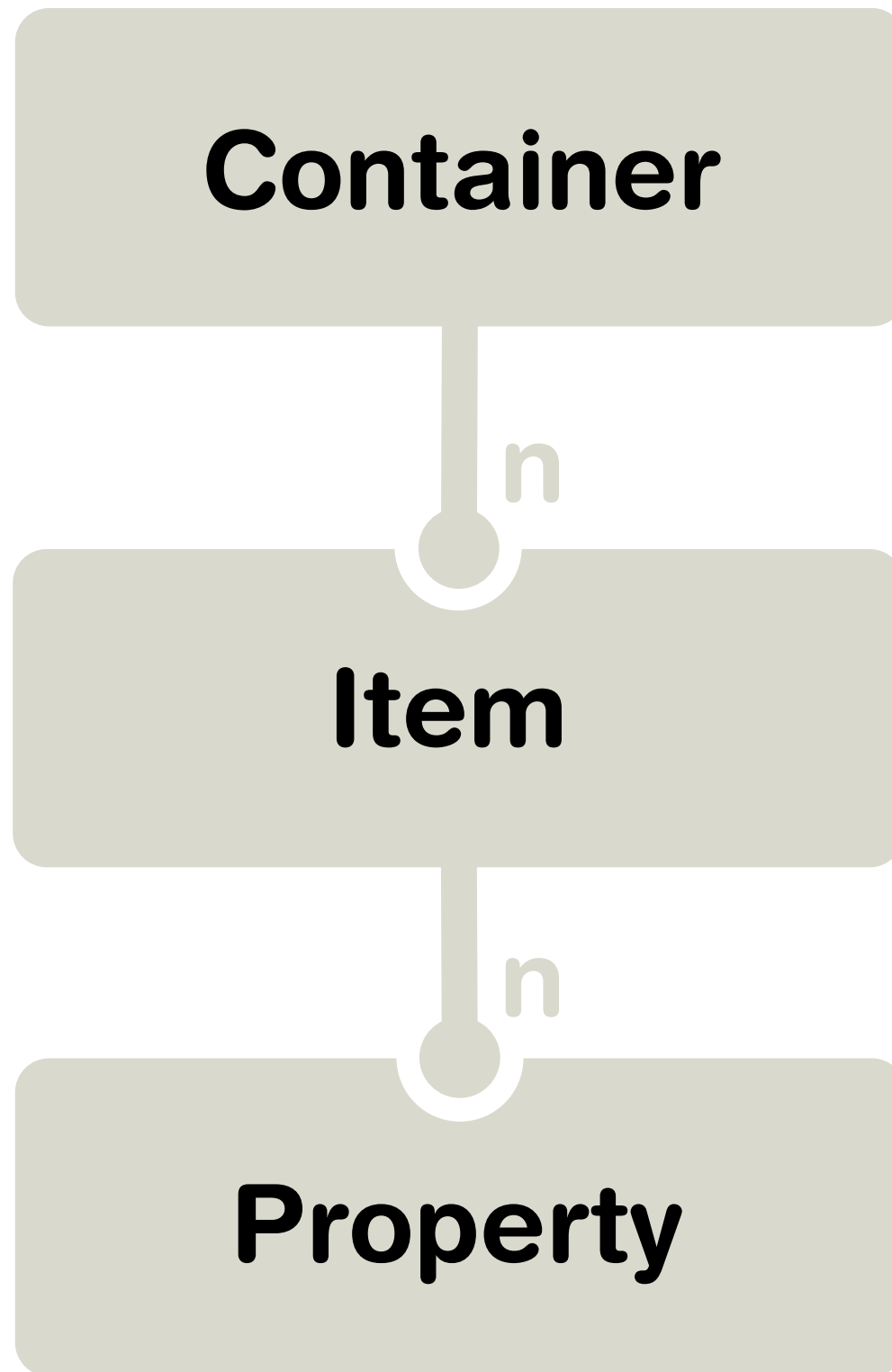
Action Action 2 Action 3

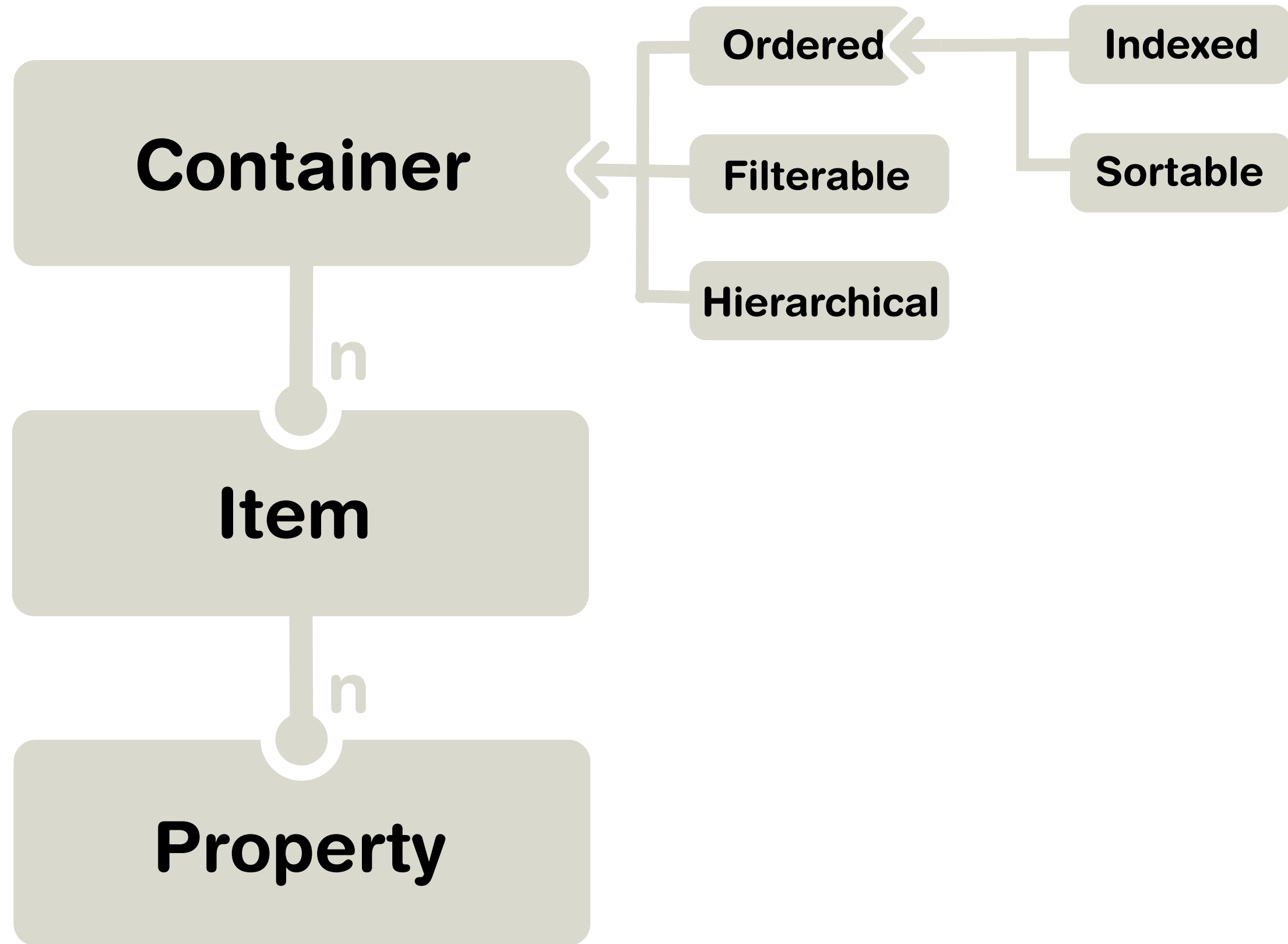


LIBRARY	Name	Time	Artist	Album	Genre	Rating
Music	Red Flag	4:12	Billy Talent	Once Again	Rock	
STORE	Millstone	6:03	Brand New	The Caitiff Choir	Metal	
VaadinTunes Store	Not The Sun	5:43	Breaking Benjamin	The Devil And God	Hardcore	2 stars
Purchased	Breath	4:32	Becoming The Archetype	Light Grenades	Indie	3 stars
PLAYLISTS	Here We Are	3:42	Bullet For My Valentine	Dichthony	Pop	4 stars
Geniues	Deep Heaven	4:45	Chasing Victory	Back In Black	Alternative	
Recently Added	Her Voice Resides	2:56	Chimaira	Dreamer	Blues	
	Natural Tan	9:34	Danko Jones	Come Clarity	Jazz	2 stars
	End It All	2:10	Deadlock	Year Zero	Hip Hop	3 stars
	Kings	3:44	Deftones	Frames	Electronica	4 stars
	Daylight Slaving	5:49	From Autumn To Ashes	Fortress	Punk	
	Mad Man	6:30	Haste The Day	Phobia	Hard Rock	
	Resolve	5:18	Four Year Strong	The Poison	Dance	2 stars
	Teargas	7:42	In Flames	Manifesto	R'n'B	3 stars
	African Air	3:13	Kemopetrol	White Pony	Gospel	4 stars
	Passing Bird	2:52	John Legend	The Big Dirty	Country	
	Red Flag	4:12	Billy Talent	Once Again	Rock	
	Millstone	6:03	Brand New	The Caitiff Choir	Metal	2 stars
	Not The Sun	5:43	Breaking Benjamin	The Devil And God	Hardcore	3 stars
	Breath	4:32	Becoming The Archetype	Light Grenades	Indie	4 stars
	Here We Are	3:42	Bullet For My Valentine	Dichthony	Pop	
	Deep Heaven	4:45	Chasing Victory	Back In Black	Alternative	
	Her Voice Resides	2:56	Chimaira	Dreamer	Blues	2 stars
	Natural Tan	9:34	Danko Jones	Come Clarity	Jazz	3 stars
	End It All	2:10	Deadlock	Year Zero	Hip Hop	4 stars
	Kings	3:44	Deftones	Frames	Electronica	



User Interface
Data Source
Theme





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



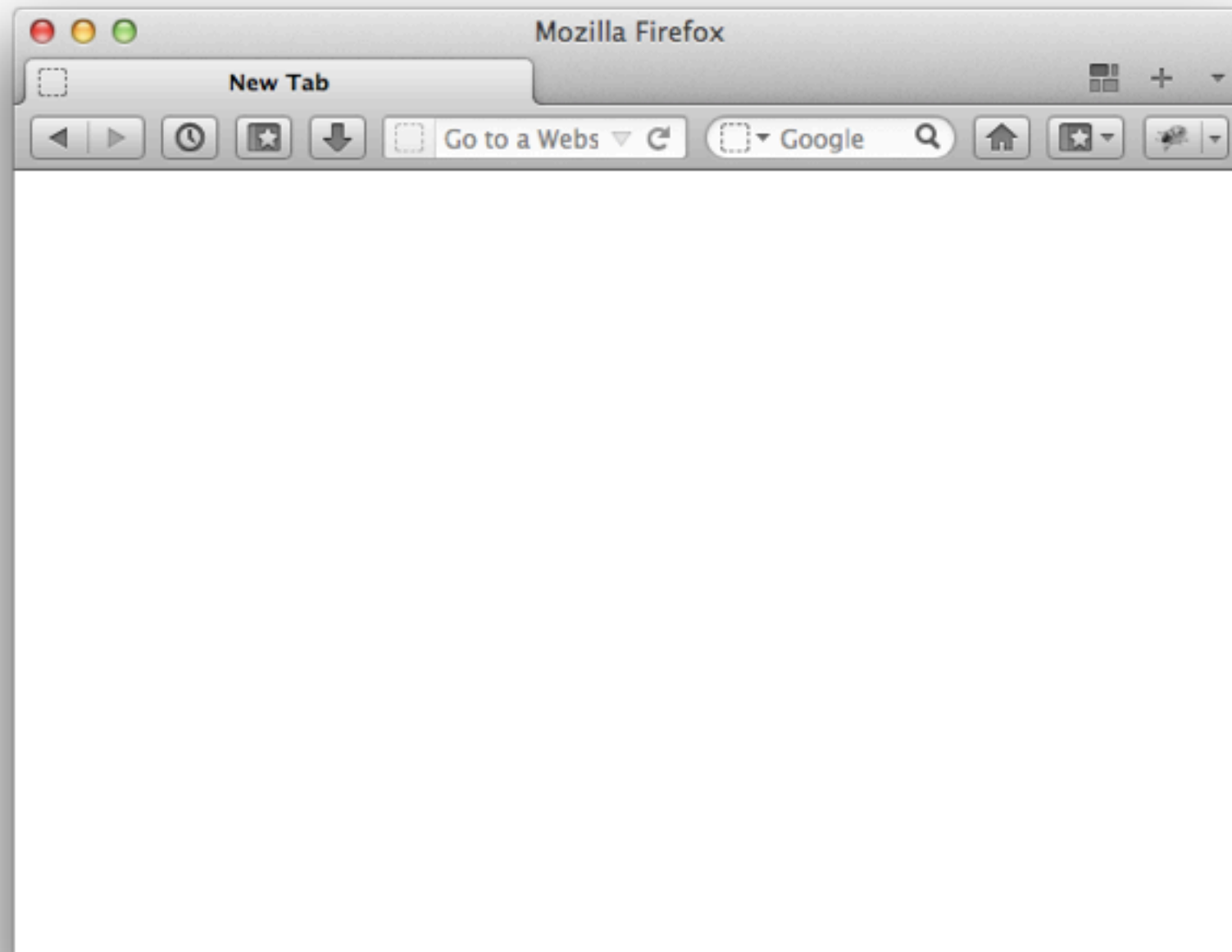
Server + Client

Web application layers

	Backend server	Web server	RPC	Java to JavaScript	JavaScript
Vaadin	required	required	optional	optional	optional
GWT	required	required	required	required	optional
JS	required	required	required		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



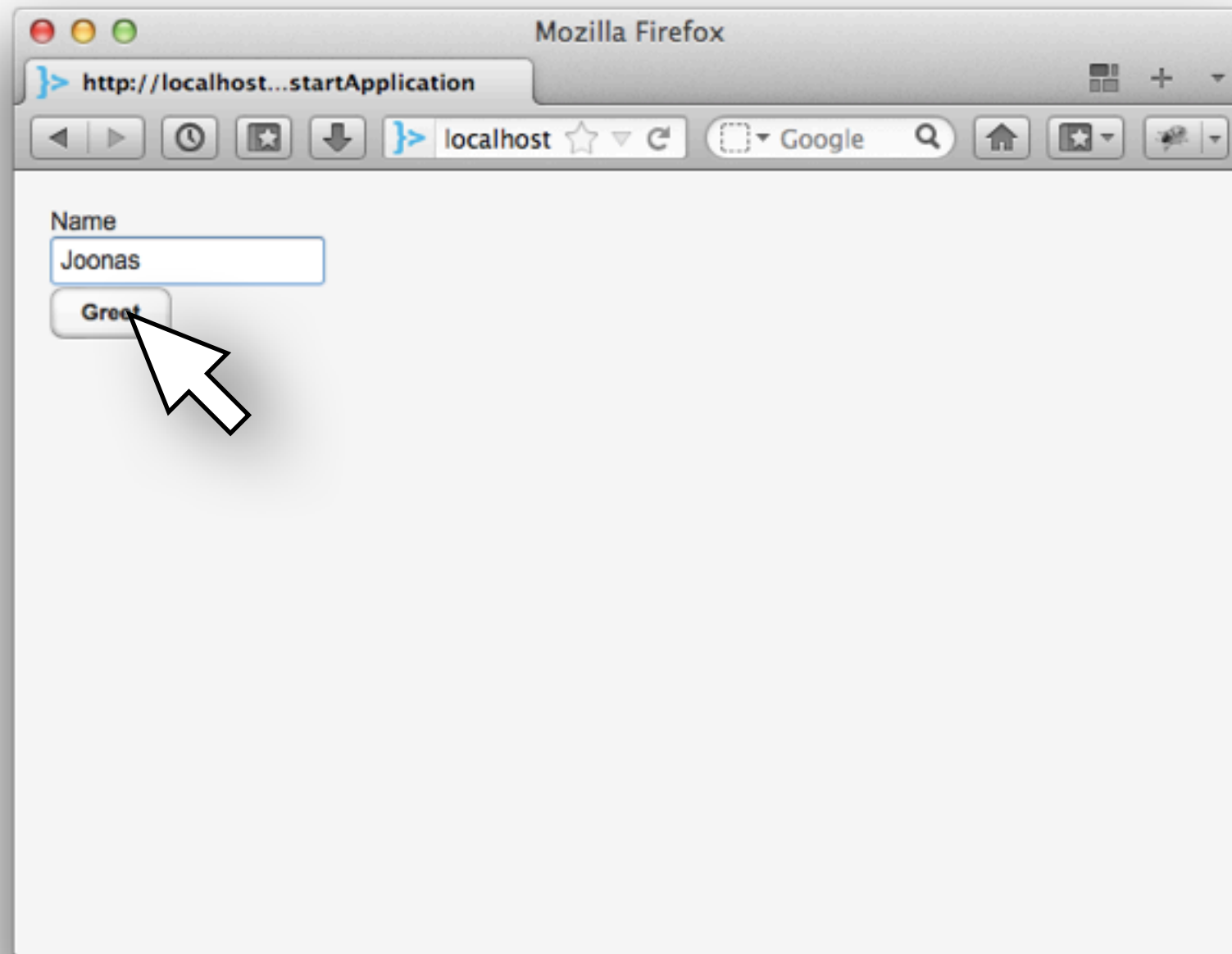
compress

250k



**reduced
widgetset**

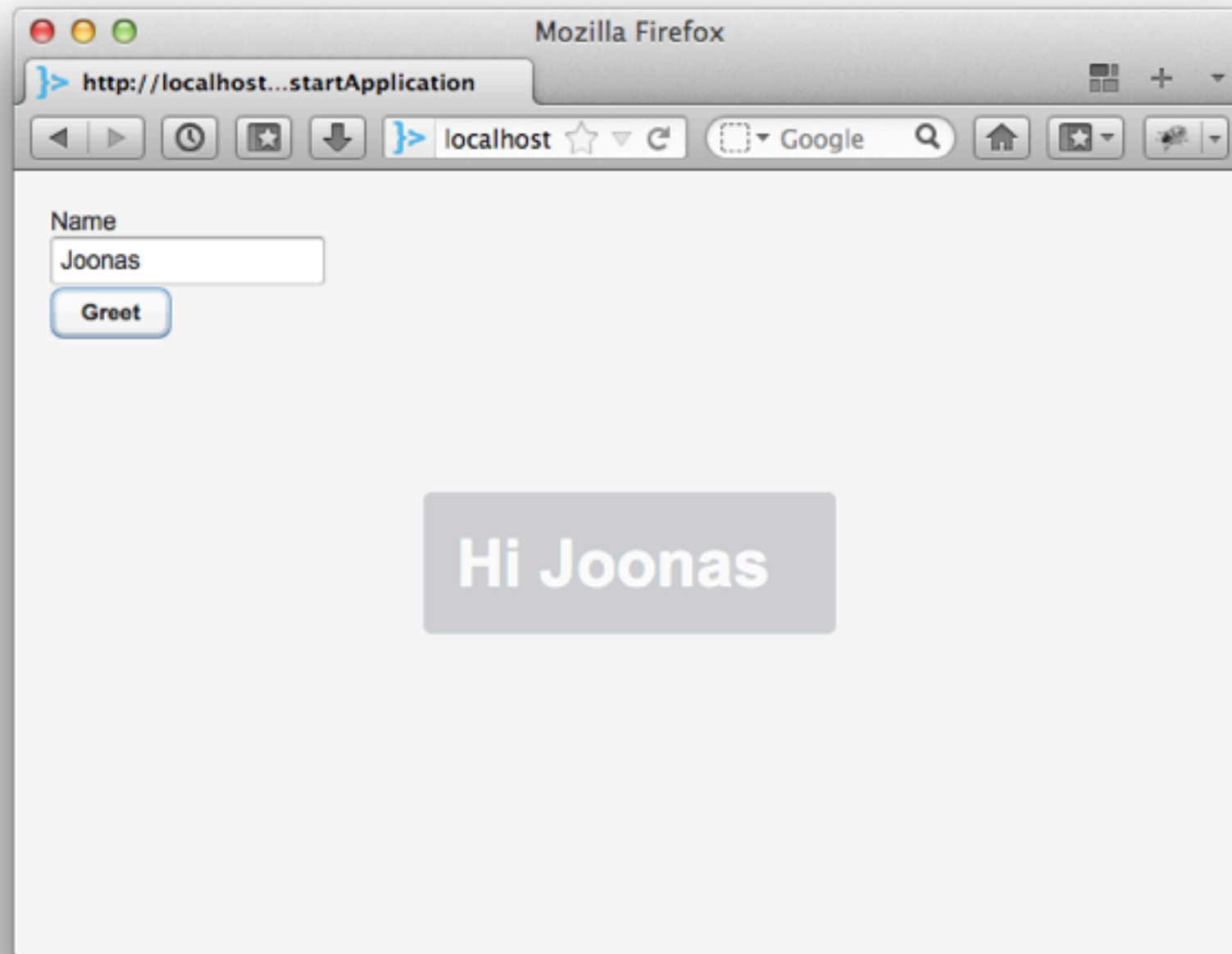
120k




- 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

Trying it out

<https://github.com/jojule/NotesDemo>



Embracing Java



**Any JVM
Language**

Internet Explorer
Chrome
Firefox
Safari
Opera
iOS
Android



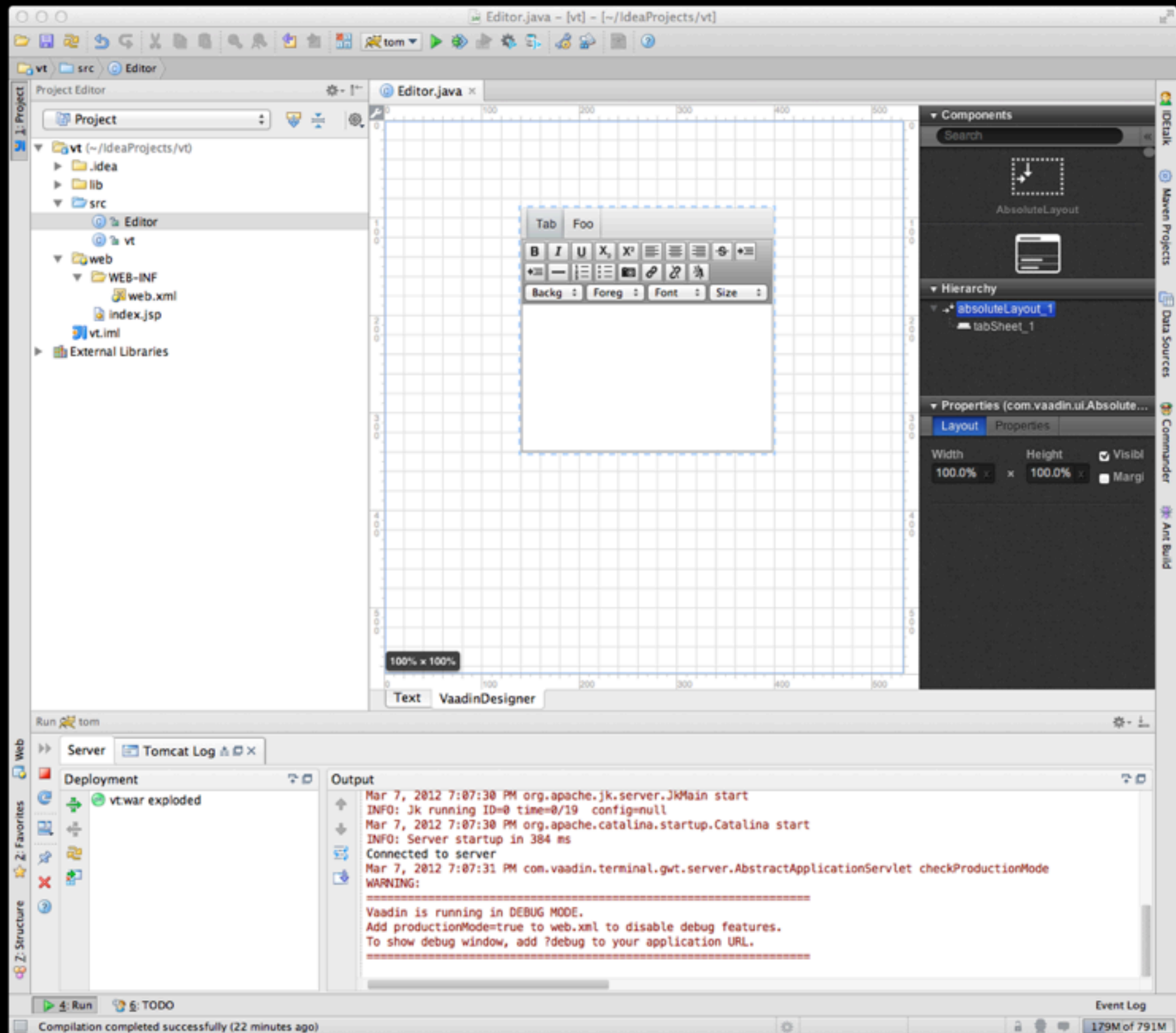
**No
browser
plugins**

**Nothing to
install**

**Servlet
Portlet**
(most) clouds



Eclipse
IntelliJ IDEA
Netbeans
Maven
Ant
Spring Roo
...



Vaadin
today

0.0



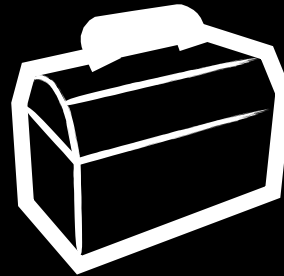
v0.1
2001

v3
2002

Open
Source

v4
2006

Ajax



v5
2007

v

7

v6
2009

today

Renewed
from Inside

Sass

JS

HTML5

+=
GWT

UI

RPC
State

Field

Complete
stack

New in Vaadin 7

Google Web Toolkit built-it

Whole stack is built in one product.

Development of client-side applications is supported to enable offline mode.

Includes all features of GWT 2.5 like SuperDevMode, Elemental and UIBinder.

New windowing API

Renewed windowing API that gives easier access to the web platform.

Built in support for multiple browser tabs.

Flexible URI fragment based Navigation API.

Redesigned forms

Field groups can bind fields on any layout to a data source.

Declarative validation with JSR-303 validators.

Generics and explicit converters make data formatting easily customizable.

Excellent support for JavaScript

Publishing of JavaScript callbacks directly from server.

Use any JavaScript widgets with simple generic wrapper without GWT.

Sass-based themes

Supports most important features of the leading Sass language.

Automatic compilation of SCSS to CSS on the fly.

Makes theme CSS structured and eases re-use.

Simplified widget development

Statically typed 2-way Java-based RPC.

Automated component state synchronization.

HTML5

Redesigned layout use browsers' native layouting engine and minimizes reflows.

Full control with CSS to all layout properties.

Now uses HTML5 doctype.

More extension points

Any widget can be customized with client-side extensions.

Any part of the generated start-up page can be changed with extension.

Supports client-side extensions that have no visible parts.

Heartbeat

- Optionally keeps session alive
- Reliable browser window close monitoring

Redesigned layouts

- Minimal or no layout calculations to maximize layout speed
- Full control of layouts with CSS including borders and margins
- Redesigned lighter DOM for vertical, horizontal and css layout

Modular Jars

- Split to seven jars to allow deploying only what you need

Hierarchical error handling

- Delegate component errors to parent components

Adding multiple components at once

- All component containers now support `addComponents(Component...)`
- Support adding multiple components in constructors

UI replaces Application

- New architecture make writing Vaadin applications simpler
- Supports multiple browser tabs by default

Closing UI programmatically both from background thread and in ui event handlers

- Reliable clean of UI when browser is closed

Multiple concurrent themes

- Support showing components with different themes on one page at once
- Theme names are included in CSS selectors

Browser and request details available in `init`

- Direct access to request and session on UI `init`
- Access detected browser details on UI `init`

RPC

- Call server-side methods from clients-side in Java
- Call client-side methods from server
- Static typing allows compile time checking
- Supports
 - Primitive types
 - Boxed primitive types
 - String
 - Java beans as long as all property types are also supported
 - Connector
 - enum
 - Array
 - URLReference (used to transmit Resource)
 - List<SupportedType>
 - Map<SupportedType, <SupportedType>>
 - Set<SupportedType>

Delayed RPC

- Allows piggybacking the next XHR call instead of calling to server immediately
- Supports optional folding of RPC calls together preserving only last value

Connectors

- Connectors provide a flexible communication channel between client and server
- Separating communication code from widgets promotes reusability

Dynamic state generation

- Allow calculating component state on the fly just before state is sent to client

Differential state synchronization

- Only parts of the state that are modified are sent over the wire
- Only parts that differ from default values are sent in initial state synchronization
- Nothing is sent for unchanged components

Server-client shared state

- Java objects can be shared between client and server for easy component development
- State is automatically mirrored from server to client
- Support both public fields and bean properties

Supports all types supported by the new RPC API

High level view navigation

- Support for URI fragment based view management
- Support for registering both views instances as well as view classes
- Programmatic navigation with `navigateTo()`
- Supports saving bookmarks to views
- Supports parameterized views
- Views can block navigation

ColorPicker component

- Easy to use interface with clickable color gradients
- RGB, HSV and swatches color modes
- Color history
- Color preview
- CSS color code representation and handling

Fine-grained listeners

- For any component and event type, write `addFooListener()` instead of generic `addListener(FooEvent)`
- Supports code completion in IDE:s better
- Makes it possible to use Java 8 Lambdas

Automated repaints

- No more need to call `requestRepaint()` in your components

Renewed Vaadin Maven Plugin

- Now includes features from GWT Maven Plugin

Renewed Eclipse Plugin

- Adds Apache Ivy based dependency management

Page bootstrapping renewed

- Simpler inclusion of Vaadin UIs to custom web pages

UIProvider

- Default UI provider chooses UI based on servlet parameters
- Allows providing a different UI based on request parameters, URI or browser

VaadinSession

- Full control over session lifecycle
- Abstract away from servlets and portlets

VaadinService

- Access deployment information and HTTP requests easily
- Abstract away from servlets and portlets

Generics in data model

- Property `getValue()` returns expected type

Two phase commit support

- Transactional properties support commit/rollback

Component extension API

- Allow adding functionality and customizations to any component
- Modify DOM and hook event listeners

JavaScript callbacks

- Declare client-side JavaScript API from server
- Eases integration with parts of the page not controlled with Vaadin

Relative paths on host page

- More flexible deployment
- Adds support to Apache ProxyPass and other similar proxies

Application archetype

- New Maven archetype eases creation of Vaadin 7 applications
- Supports widgetset compilation and precompiled widgetsets

HTML5

- Vaadin 7 uses HTML5 doctype
- Use any parts of HTML5 in your application

State change listeners

- Client-side can listen to shared state changes to update just the changed parts of the widget

Page

- Listen to page resizes
- Control navigation

Loading custom JavaScript

- Annotate server-side classes with @JavaScript to request loading of JavaScript files
- Automated control of loading order and ensuring that files are loaded only once

Deprecated API removed

- Cleaned by removing API deprecated in Vaadin 6 or before

Changed constants to enums

- Provide enums everywhere where possible instead of constants

Image component

- Specialized UI component for showing images
- Replaces Embedded

BrowserFrame component

- Specialized UI component for embedding web pages with iframes
- Replaces Embedded

Flash component

- Specialized UI component for embedding flash
- Replaces Embedded

Support for Firefox ESR

- Adds support for the latest Firefox extended support release in addition to latest stable Firefox release

Sass Compiler

- Allows modularization of themes for better reuse and easier maintenance
- Support the most important features of SCSS
- Implemented in Java, no dependency to Ruby
- Supports all of CSS
- On the fly conversion of SCSS to CSS in servlet
- Built in themes are now based on Sass
- Can be used in client-side projects as well

State change delegation

- State can be annotated to automatically delegate state changes to corresponding widgets
- Removed need to implement state change listeners and calling widget methods

Google Web Toolkit included

- A full copy of GWT is included in Vaadin Framework
- Vaadin team maintains a branch of GWT that brings newest features and bug fixes early on
- All functionality of GWT is included to enable writing of client side UI:s, stateless applications, offline functionality and custom widgets
- Included Elemental library gives direct access to all cutting edge browser features
- Both browser plugin based dev mode debugging as well as super dev mode are supported

ThreadLocal access helpers

- Direct access to VaadinService, VaadinRequest, VaadinResponse, VaadinSession and UI from anywhere during HTTP request
- Service and UI are accessible also from outside HTTP request

Native support for percent sizes

- Letting browser do the percent to pixel calculation speeds up rendering

Bootstrap page API

- Let add-ons modify the bootstrap page to remove need for complicated custom servlets

Component ID

- Replaces debug ids to allow wider use possibilities for identifying corresponding widget elements in DOM

Custom class loader

- Allow specifying custom class loaders to better support Java EE, CDI and Spring
- Supports both servlets and portlets

Bean Validation

- Annotate beans with JSR-303 standard annotations to automatically create validators for the fields

Range retrieval for indexed containers

- API for retrieving a range of Item ids in Container.Indexed to optimize performance

JavaScript Connector

- Wrap around any existing JavaScript based widget to adapt it for use in Vaadin
- No widgetset compilation needed
- Write client-side connectors in JavaScript
- Support for shared state and rpc

Field group

- Allow data binding of multiple fields together to item data source
- Supports buffering
- Supports two phase commit
- Annotation based and field name based property mapping

Explicit data model converters

- All fields support explicit conversion from presentation format to data source format
- Conversions are bidirectional
- Allow setting converters to components directly in components or for the whole application at once

Built-in default converters

- Automated conversions between String, Boolean, Long, Date, Double, Float, Integer and Number
- Conversion direction reversion
- Built in converters support internationalization

Custom field component

- Build new fields as composition of existing components

Simplified validation API

- No need to implement isValid() in validators any more

Unsupported browser detection

- Really old browsers are detected and customizable instructions given on updating the browser

Nested bean properties

- BeanItem supports nested properties to allow flattening complex datatypes

Preserving UI state on refresh

- Default is now reinitializing UI from page reload
- UI can be configured to preserve state over page reloads

Vaadin 6 compatibility layer

- Set of helper classes are provided to ease migration from Vaadin 6

Explicit layouts for Window and Panel

- Unification of API for component parents
- Window and Panel components now require setting layout explicitly

Customized Table column formats

- Converters can be set per Table column to customize column formatting

Layout manager

- Allows building custom layout calculations for widgets when browser based layouts are not powerful enough
- Optimizes number of reflows by batching layout calculations from multiple widgets together

Client-side component hierarchy

- Access the components hierarchy enables optimizations
- List connector hierarchy in debug console

Automatic injection of CSS files

- Inject custom CSS with @StyleSheet annotation

Popup blocker support

- Open popups and start downloads in a way not stopped by popup blockers

Client driven client-server mapping

- Mapping between server-side and client-side components is defined in client-side annotations to avoid server-side classpath issues

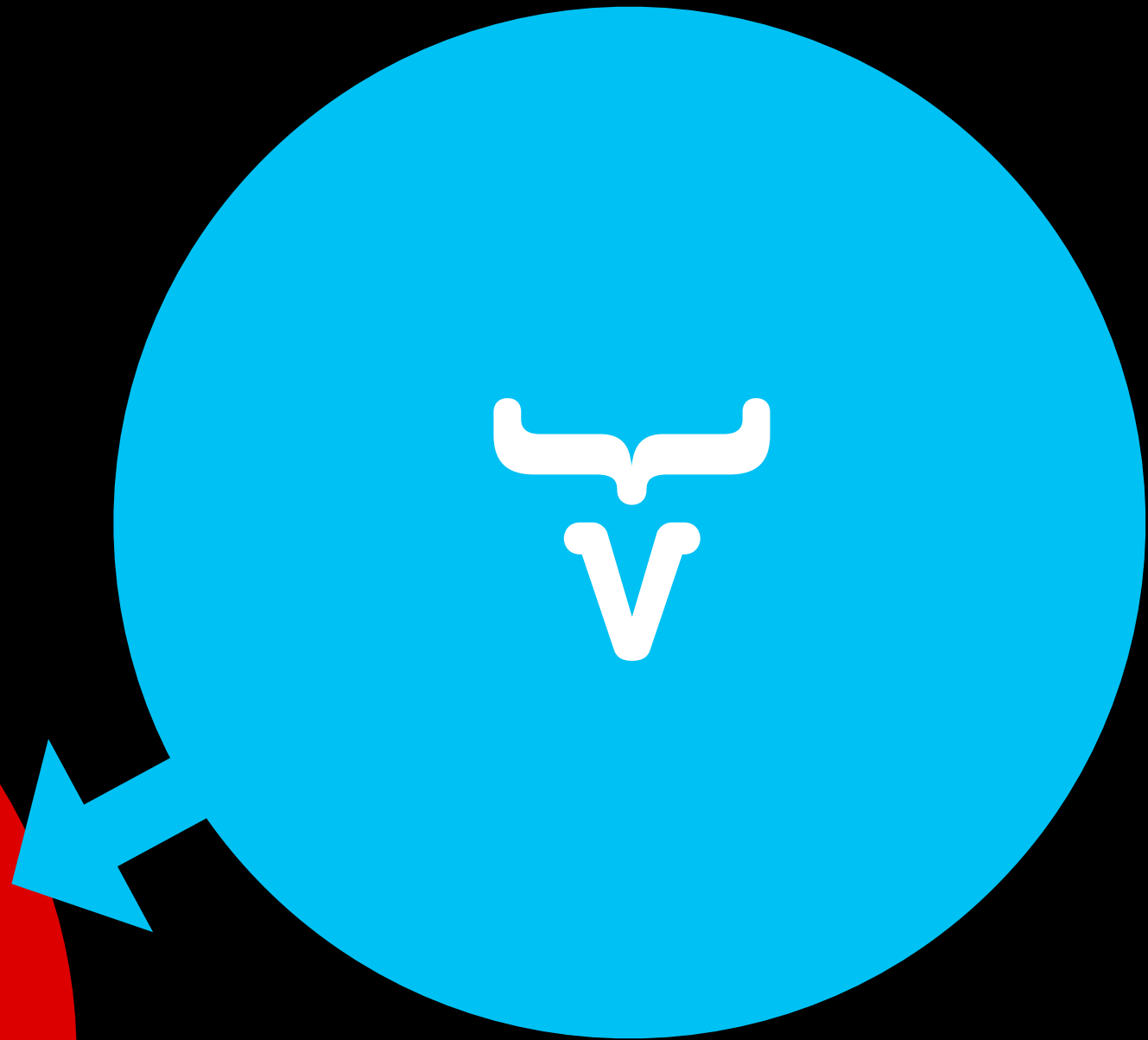
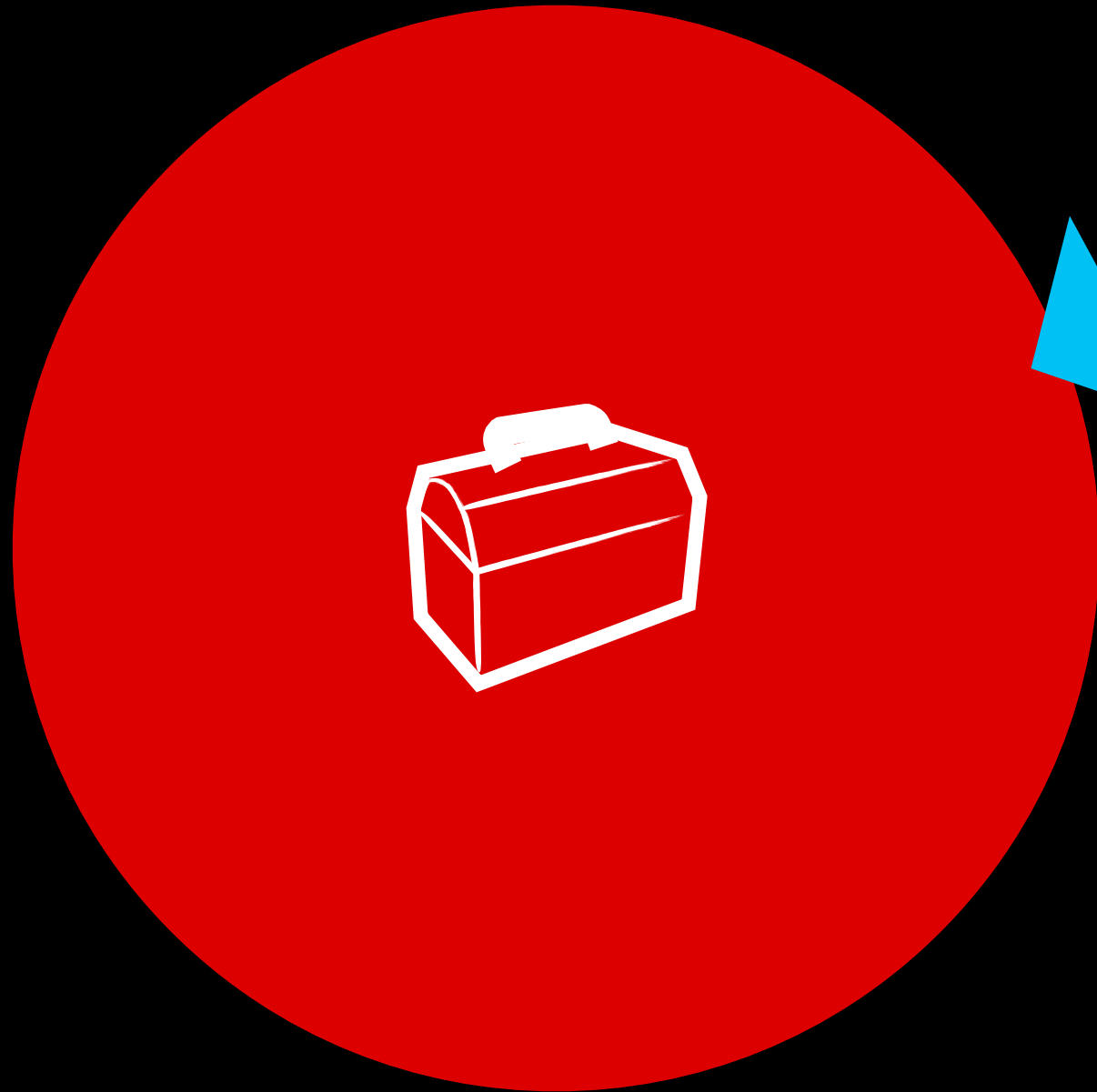
ComputedStyle API

- Access computed CSS styles on the client-side

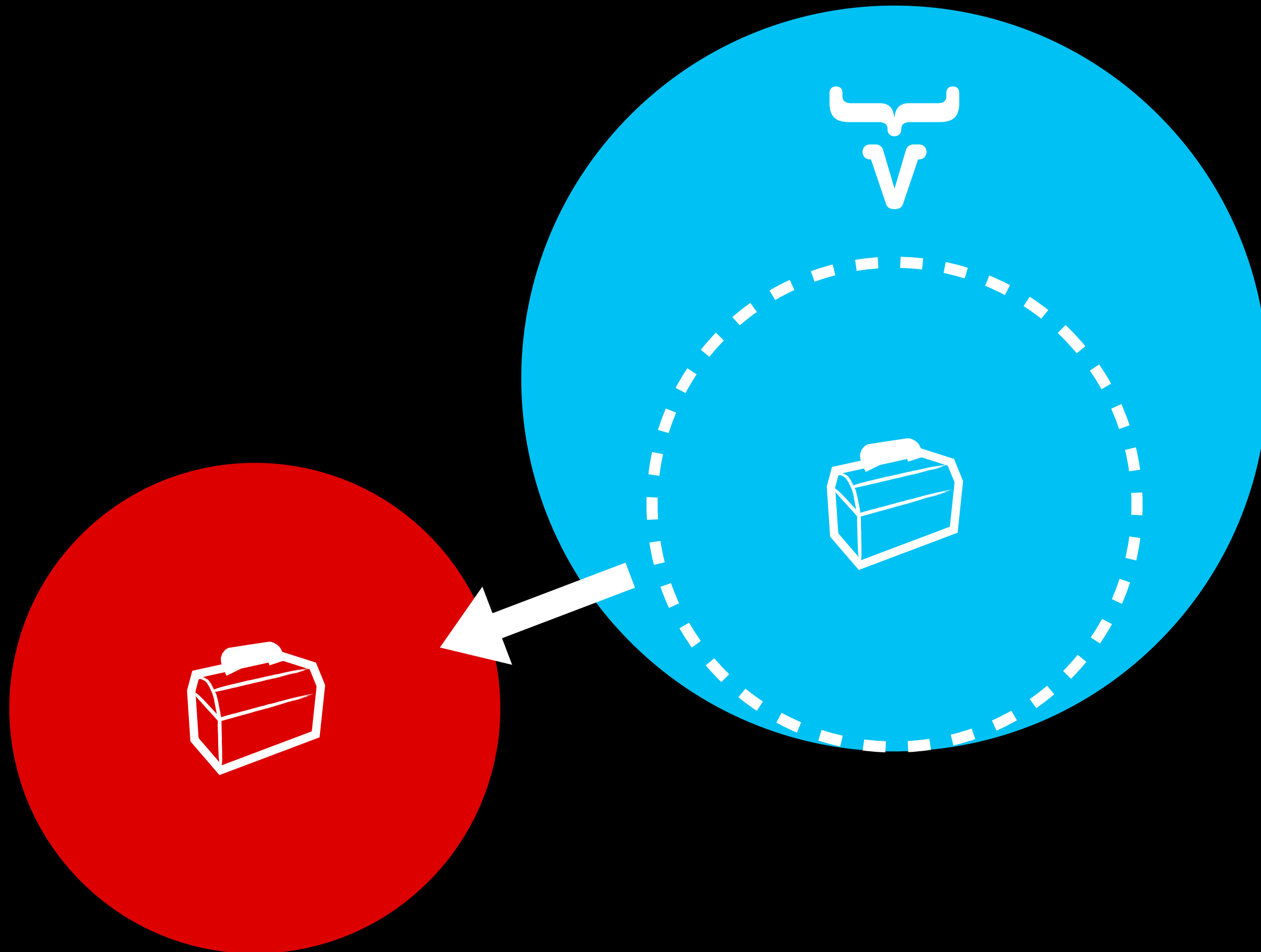
Favorite picks



Vaadin += GWT







GWT
Compatible

Client-side
Optimized for
Control

Server-side
Optimized for
Productivity

Client-side vs Server-side

Optimized for
Productivity

Optimized for
Control

Architecture



Your Application Optional Parts Vaadin 3rd Party

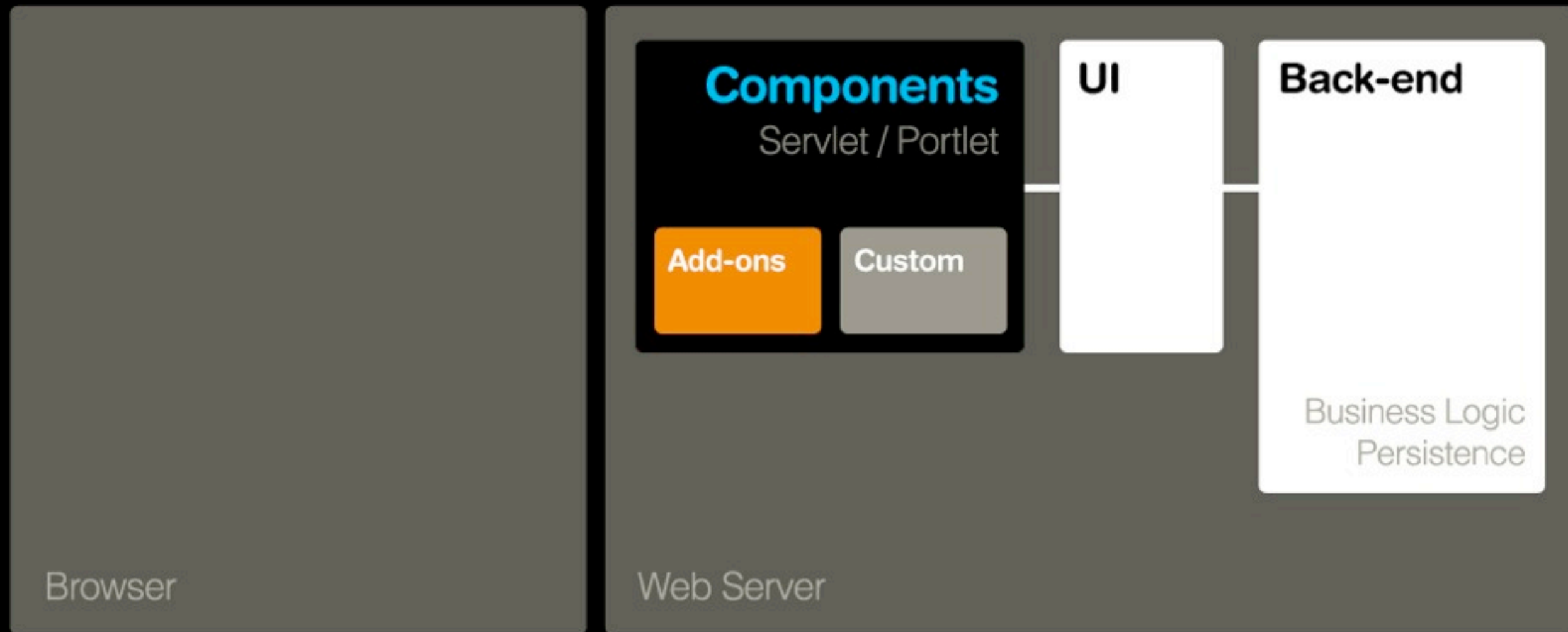


Your Application

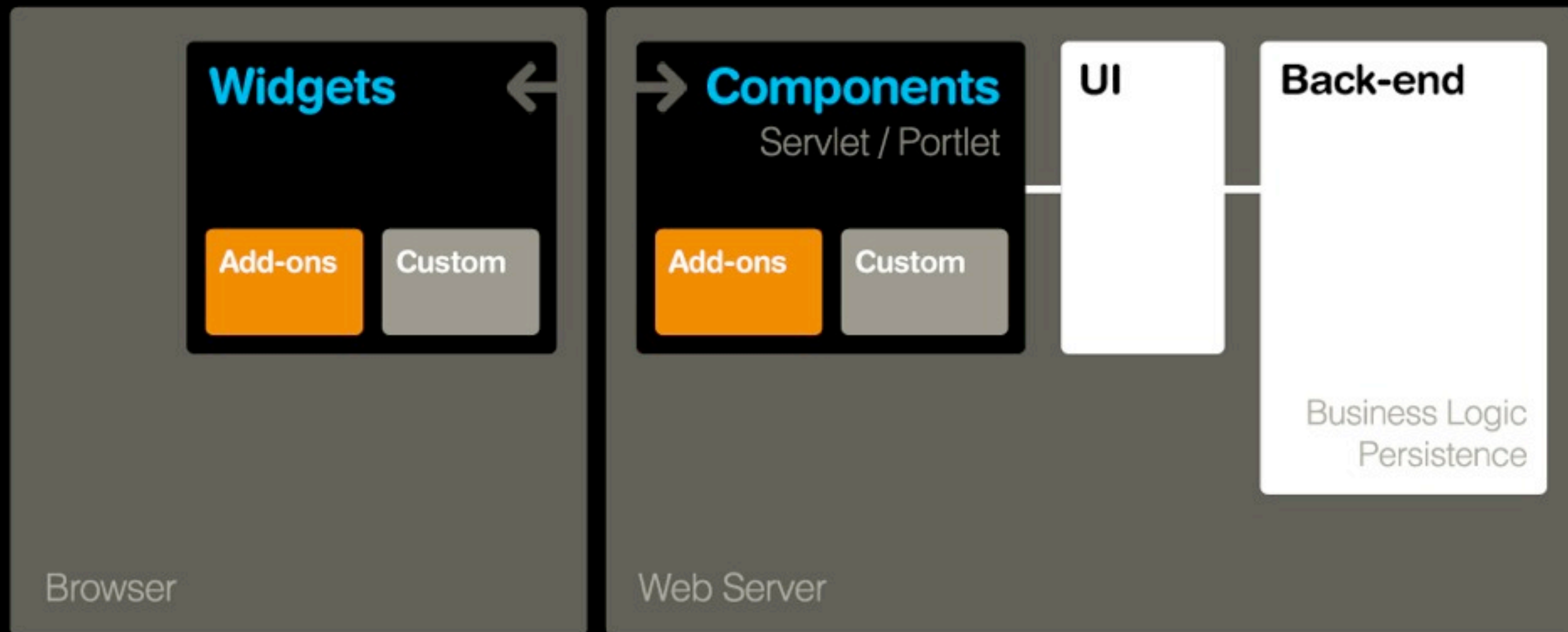
Optional Parts

Vaadin

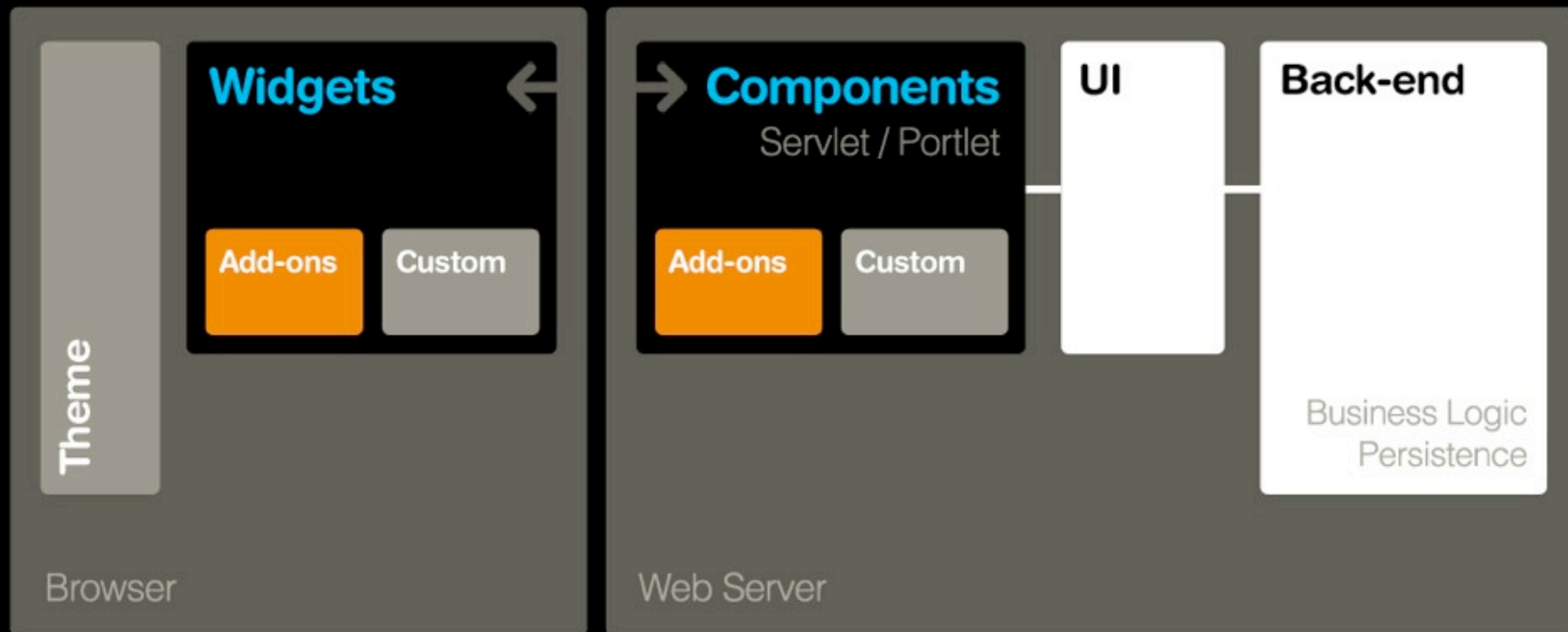
3rd Party



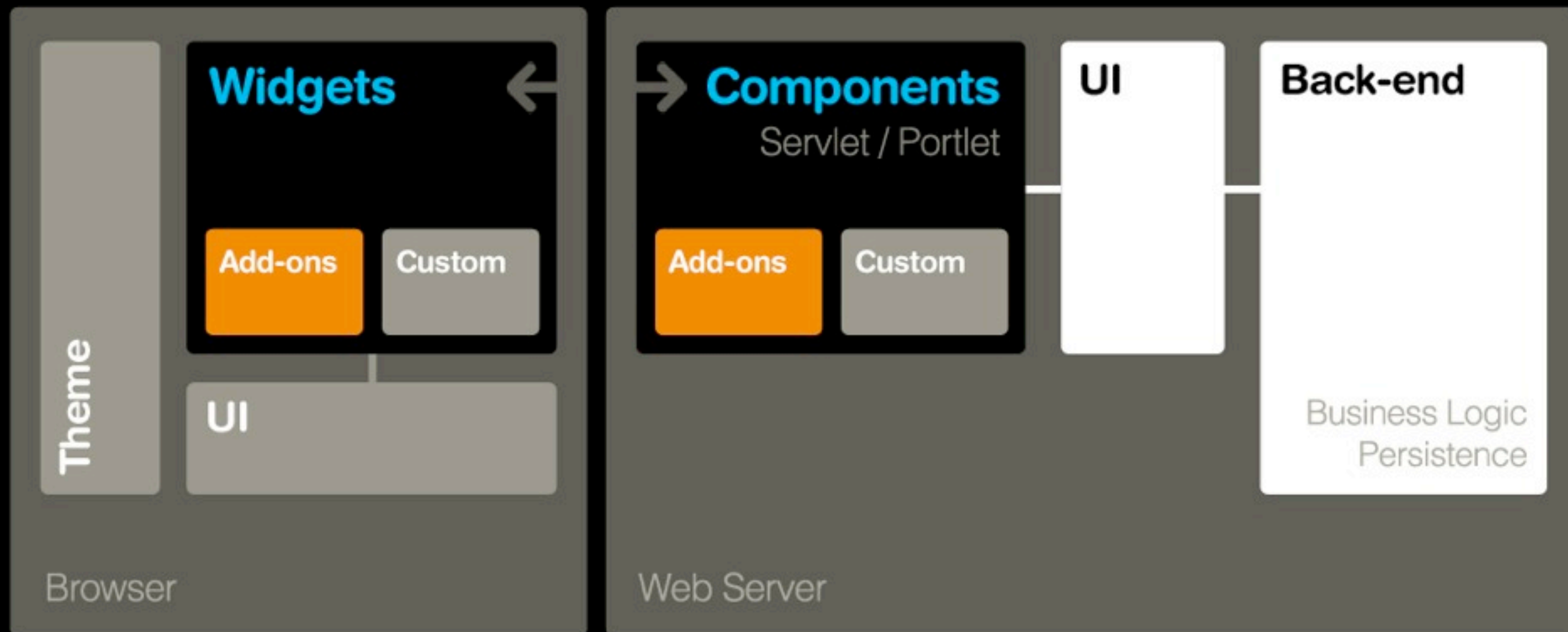
Your Application Optional Parts Vaadin 3rd Party



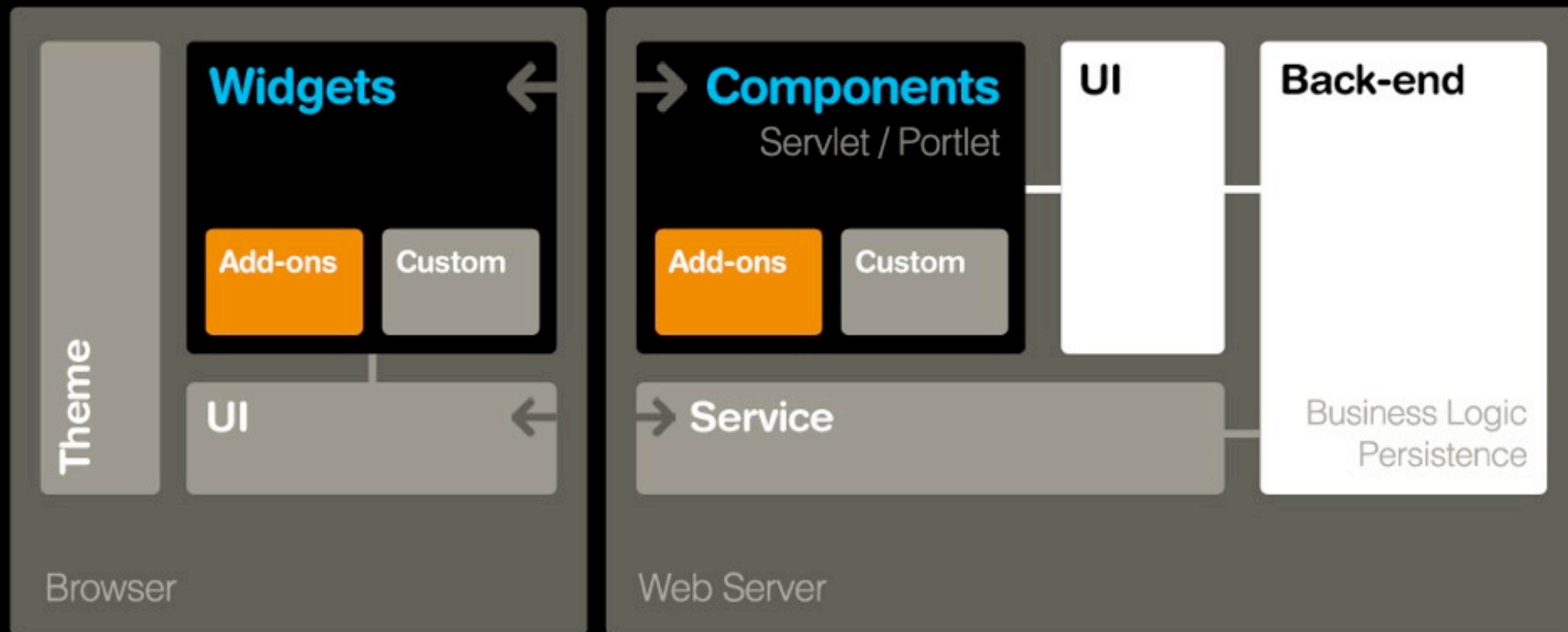
Your Application Optional Parts **Vaadin** 3rd Party



Your Application Optional Parts Vaadin 3rd Party

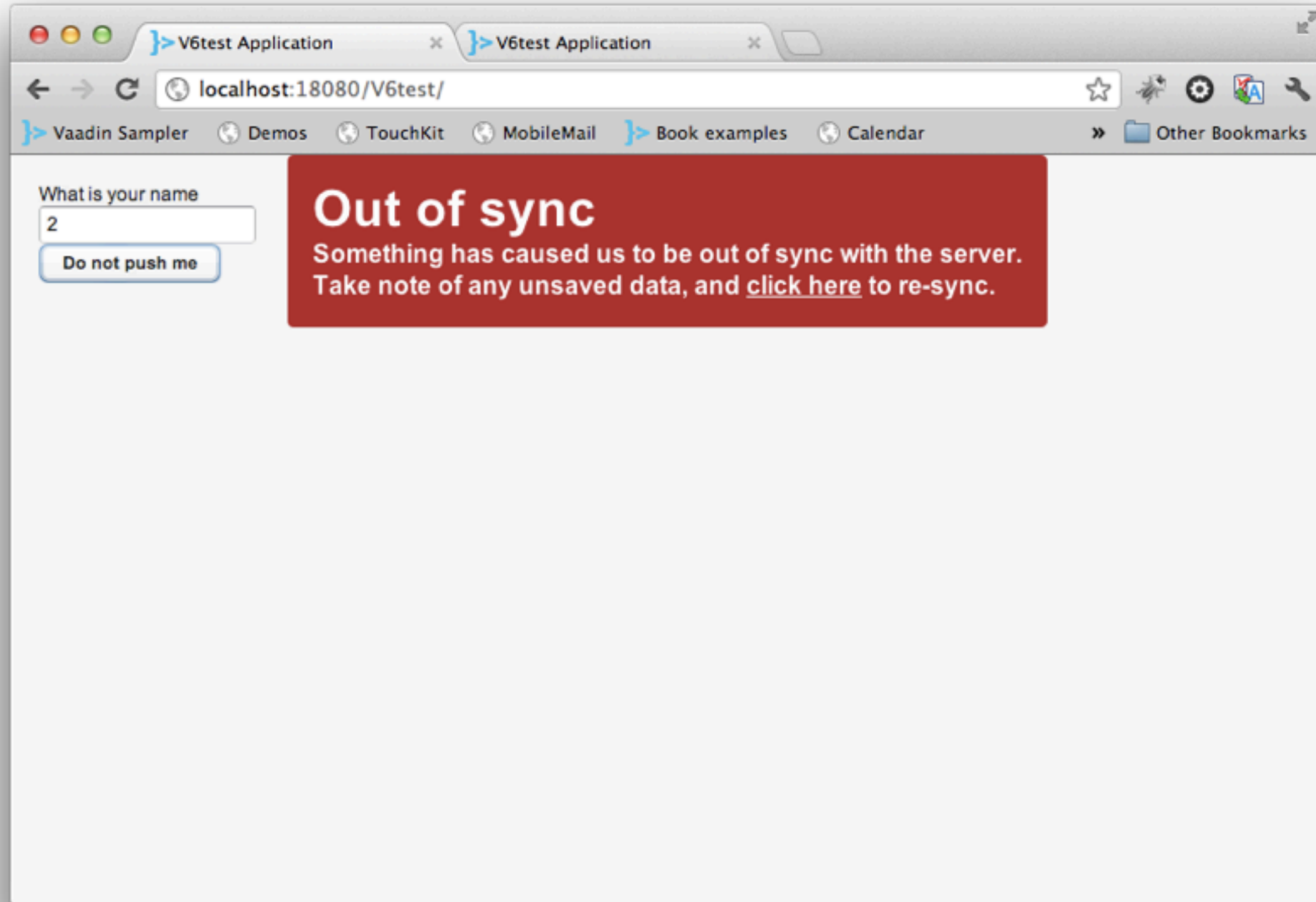


Your Application Optional Parts Vaadin 3rd Party



Your Application Optional Parts Vaadin 3rd Party

New Windowing API



Built-in high level

View management

```
public class Vaadin6App extends Application {

    public void init() {
        setMainWindow(createWindow());
    }

    public Window getWindow(String name) {
        Window window = super.getWindow(name);
        if (window == null) {
            window = createWindow();
            window.setName(name);
            addWindow(window);
        }
        return window;
    }

    private Window createWindow() {
        Window window = new Window("Vaadin 6 Application");
        window.addComponent(new TextField("What is your name"));
        window.addComponent(new Button("Do not push me"));
        return window;
    }

}
```

```
@Title("Vaadin 7 Application")
public class Vaadin7uiUI extends UI {

    public void init(VaadinRequest request) {
        addComponent(new TextField("What is your name"));
        addComponent(new Button("Do not push me"));
    }

}
```

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;
}
```

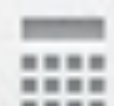
Redesigned Forms


```
public class Employee {  
    String firstName;  
    String lastName;  
    double salary;  
    Date birthDate;  
    // Getters, setters, ...  
}
```

```
Form form = new Form();  
form.setItemDataSource(  
    new BeanItem<Employee>(employee));
```

Birth Date

11/20/75



First Name

David

Last Name

Smith

Salary

100000.0

First name

David

Last name

Smith

Birth date

11/20/75



Salary

100,000

```
form.setFormFieldFactory(new FormFieldFactory() {  
    public Field createField(Item item, Object propertyId,  
        Component uiContext) {  
  
        if ("birthDate".equals(propertyId)) {  
            DateField df = new DateField();  
            df.setResolution(DateField.RESOLUTION_DAY);  
            return df;  
        }  
  
        // ..  
  
        return DefaultFieldFactory.createFieldByPropertyType(item  
            .getItemProperty(propertyId).getType());  
    }  
});
```

```
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));
```

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

“Joonas Lehtinen”

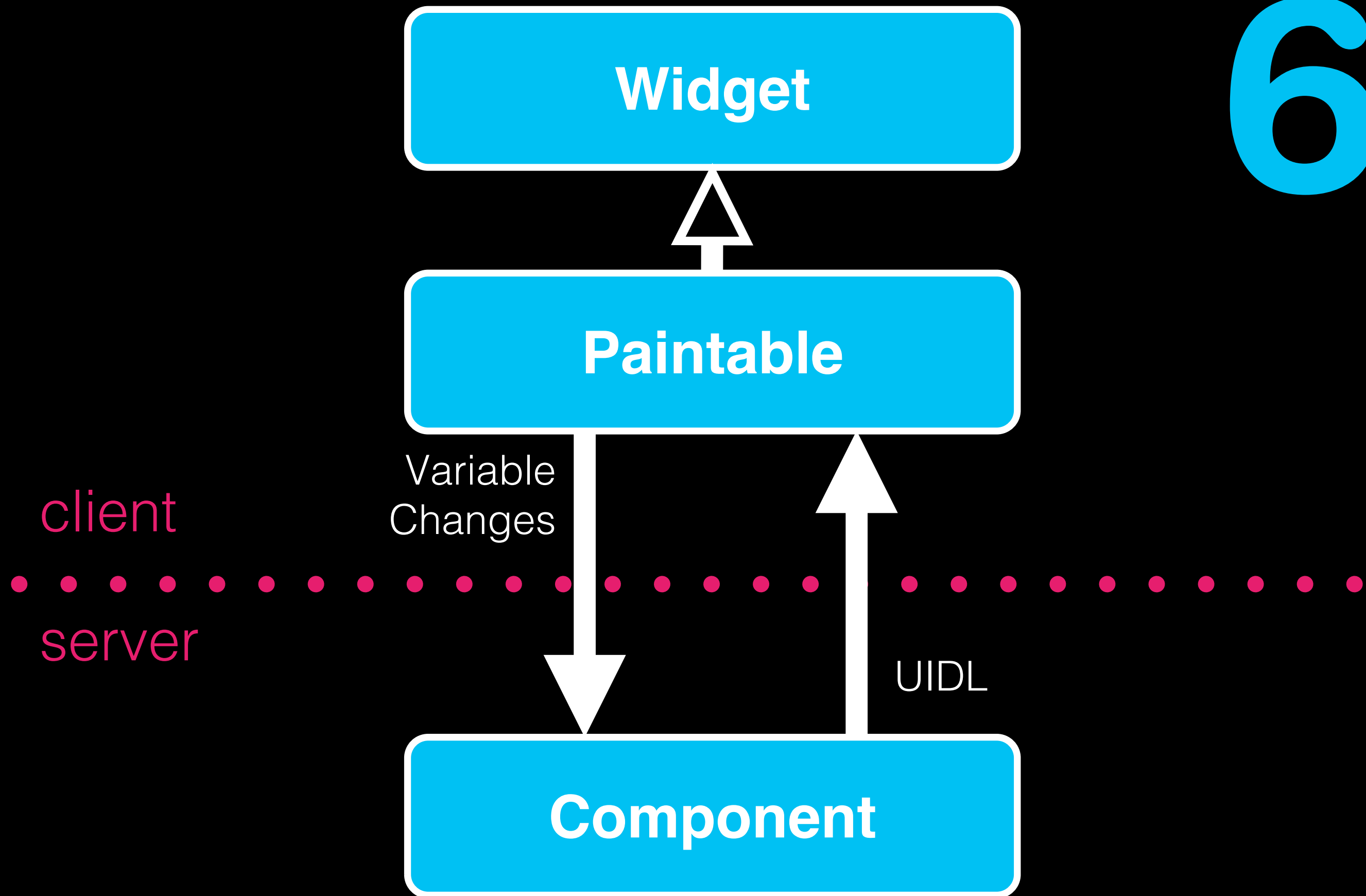
presentation
.....
model



**firstName = “Joonas”
lastName = “Lehtinen”**

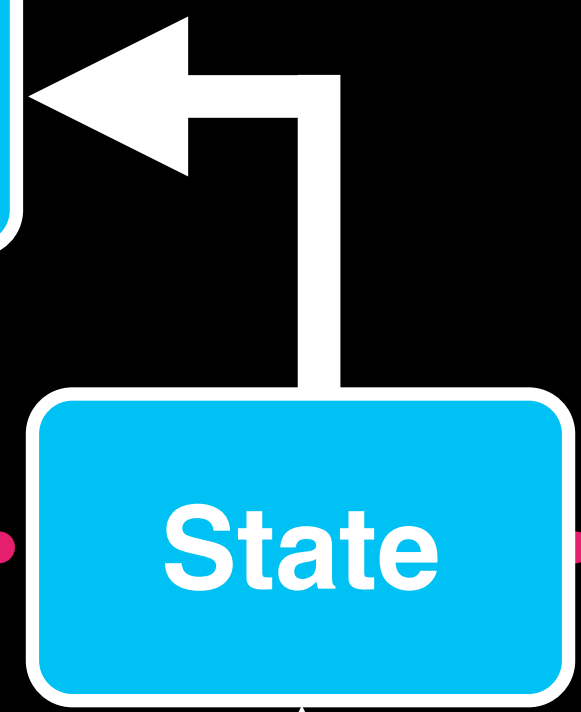
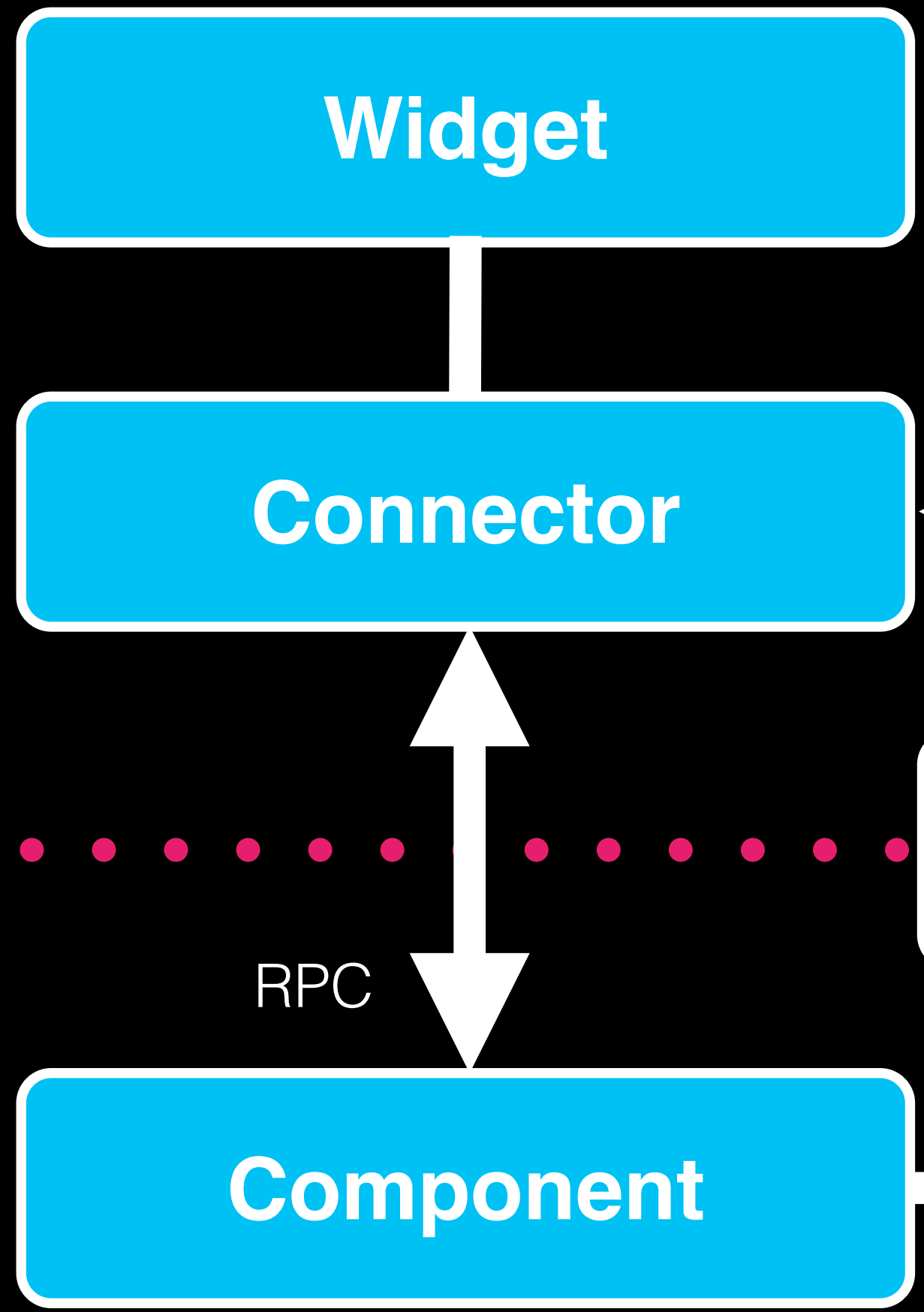
RPC State

6



7

client
.....
server



```
public interface ButtonRpc extends ServerRpc {  
    public void click(MouseEventDetails details);  
}
```

```
private ButtonRpc rpc =  
    RpcProxy.create(ButtonRpc.class, this);  
  
public void onClick(ClickEvent event) {  
    rpc.click(  
        new MouseEventDetails(event));  
}
```

client

```
private ButtonRpc rpc = new ButtonRpc() {  
    public void click(  
        MouseEventDetails details) {  
        // do stuff  
    }  
};  
  
public Button() {  
    registerRpc(rpc);  
}
```

server

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

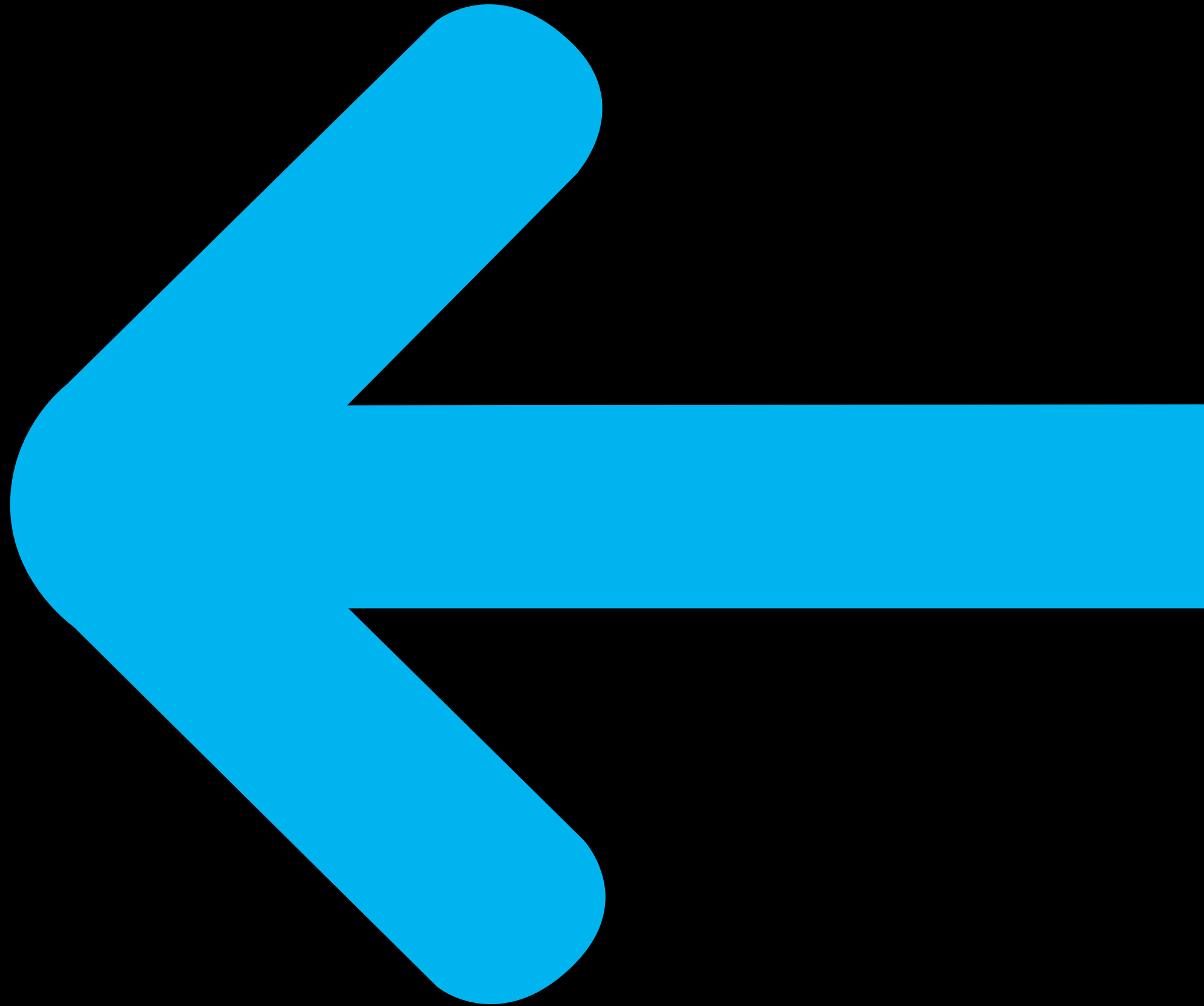
```
public class MyWidget extends AbstractJavaScriptComponent {

    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<List<Double>>> plotSeriesData =
            new ArrayList<List<List<Double>>>();
        // getters & setters
    }

}
```

getting
started



Tutorial

This tutorial teaches you the basics of Vaadin 7 with a simple one class addressbook application. It is designed to be a fast read, instead of trying to follow good design practices.

Run the application

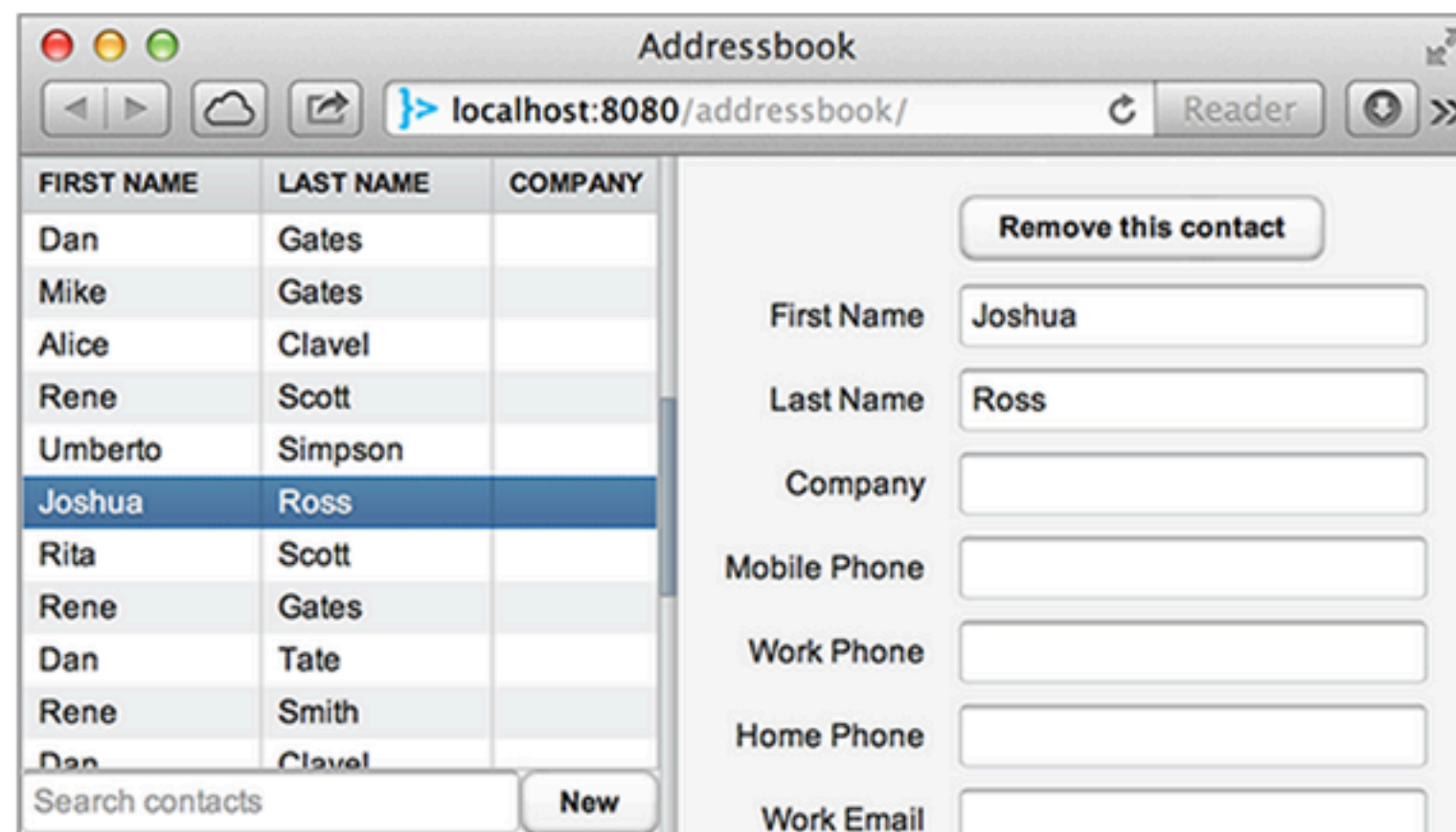
Running locally

Get the [source from Github](#) and run it:

- `git clone git://github.com/vaadin/addressbook.git`
- `cd addressbook`
- `mvn jetty:run`

Learn more

Read the [Book of Vaadin](#) - it is a comprehensive, up-to-date and free guide on everything about Vaadin.



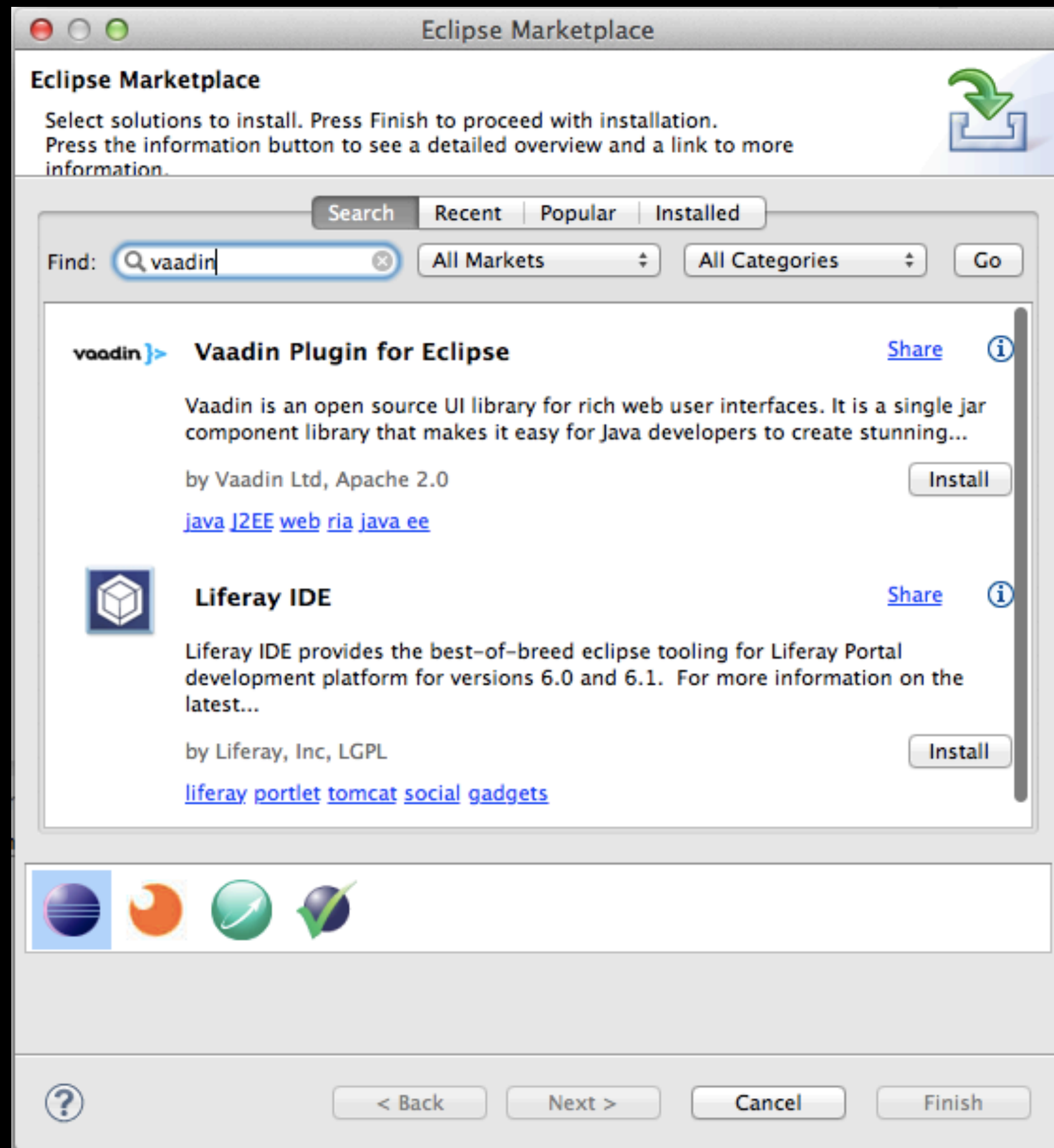
⊕ `package com.vaadin.tutorial.addressbook;`

```
@Title("Addressbook")
public class AddressbookUI extends UI {
```

User Interface

UI class defines what is shown on

Eclipse



Maven

```
mvn archetype:generate  
-DarchetypeGroupId=com.vaadin  
-DarchetypeArtifactId=  
    vaadin-archetype-application  
-DarchetypeVersion=7.0.0
```

Download for Free
vaadin.com/book

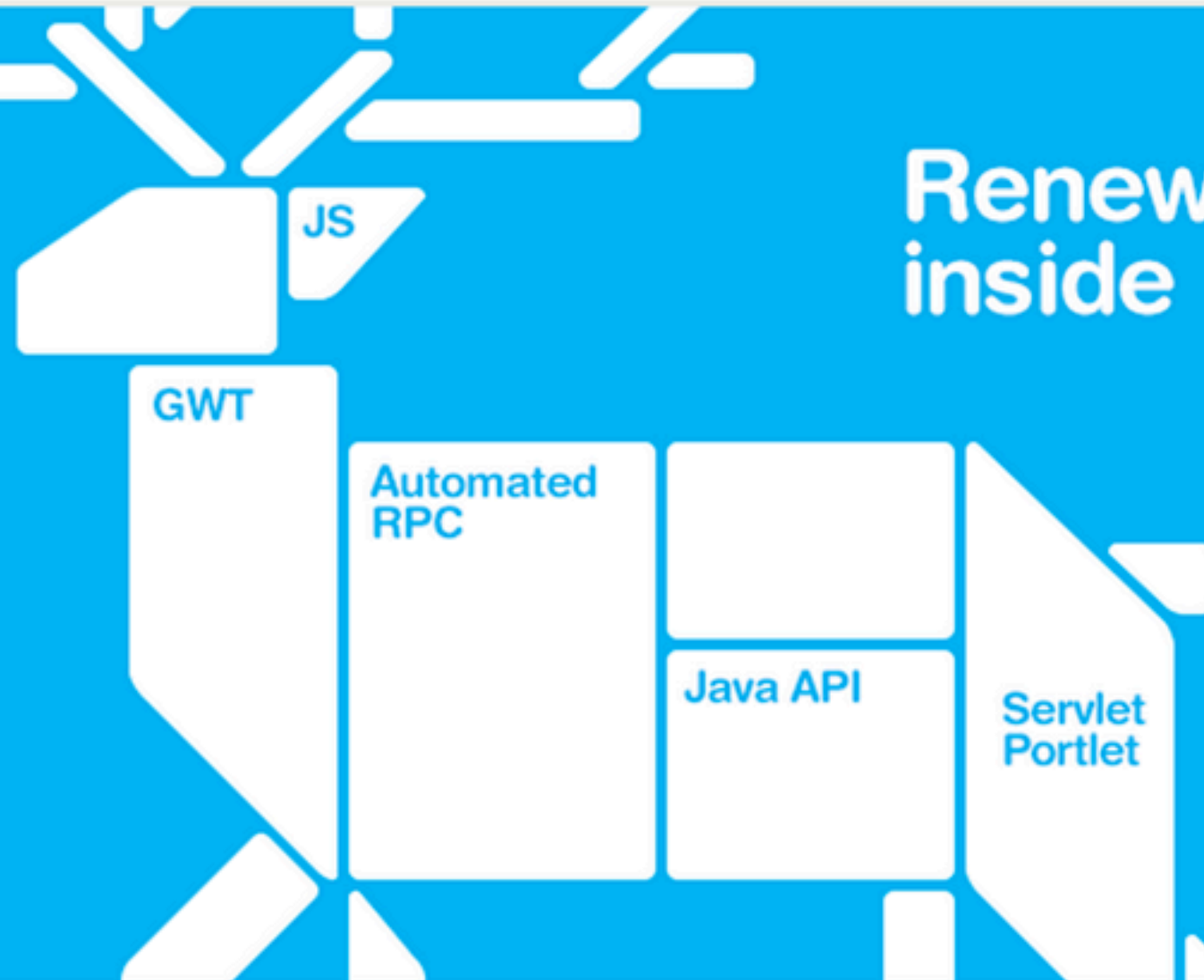


Book of Vaadin

~700 pages



Complete
stack



Renewed from
inside



joonas@vaadin.com
vaadin.com/joonas
@joonaslehtinen