

vaadin

intro

@joonaslehtinen
Founder & CEO

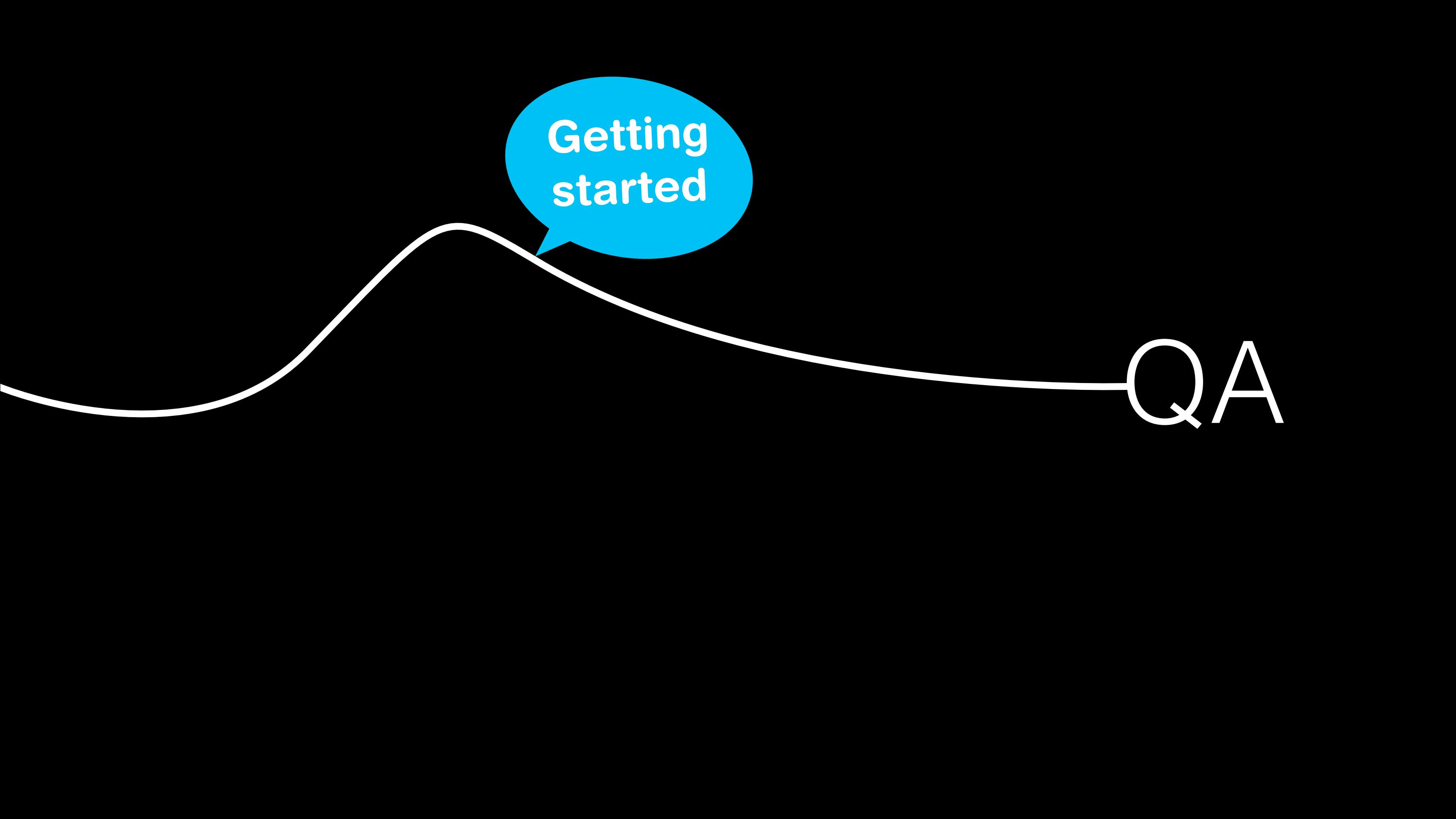


Intro to
Vaadin

new Label("Hello world")



New in
7



Getting
started

QA

vaadin }>

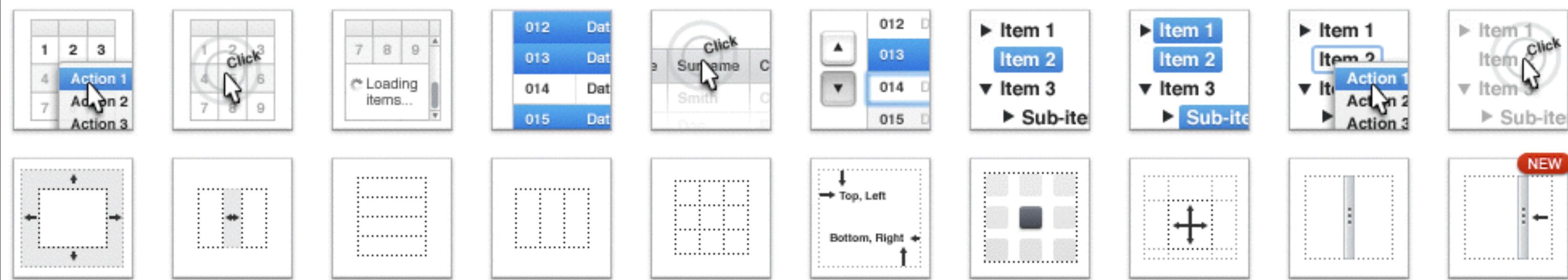


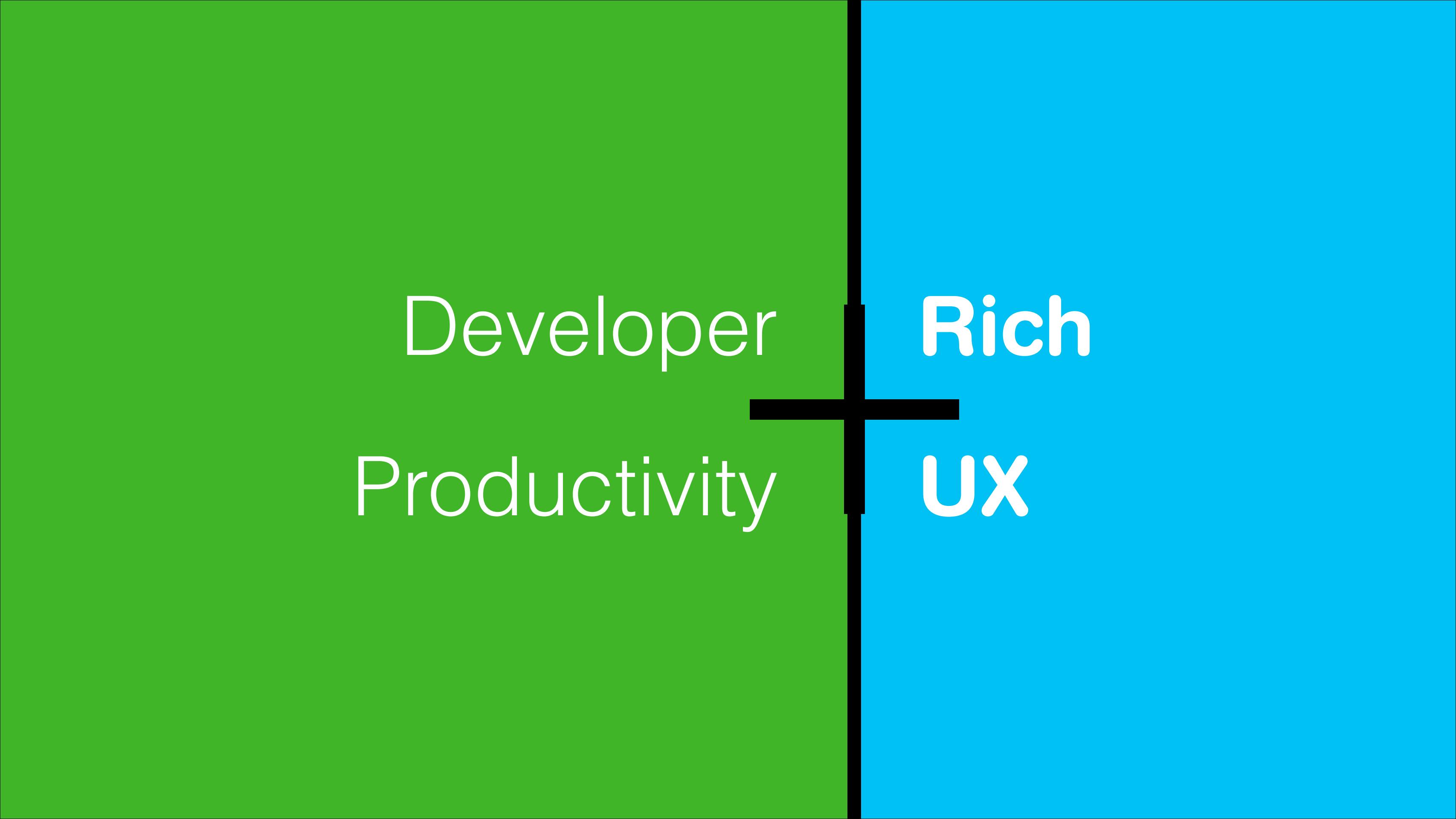


User interface
framework for rich
web **applications**



Building blocks





Developer

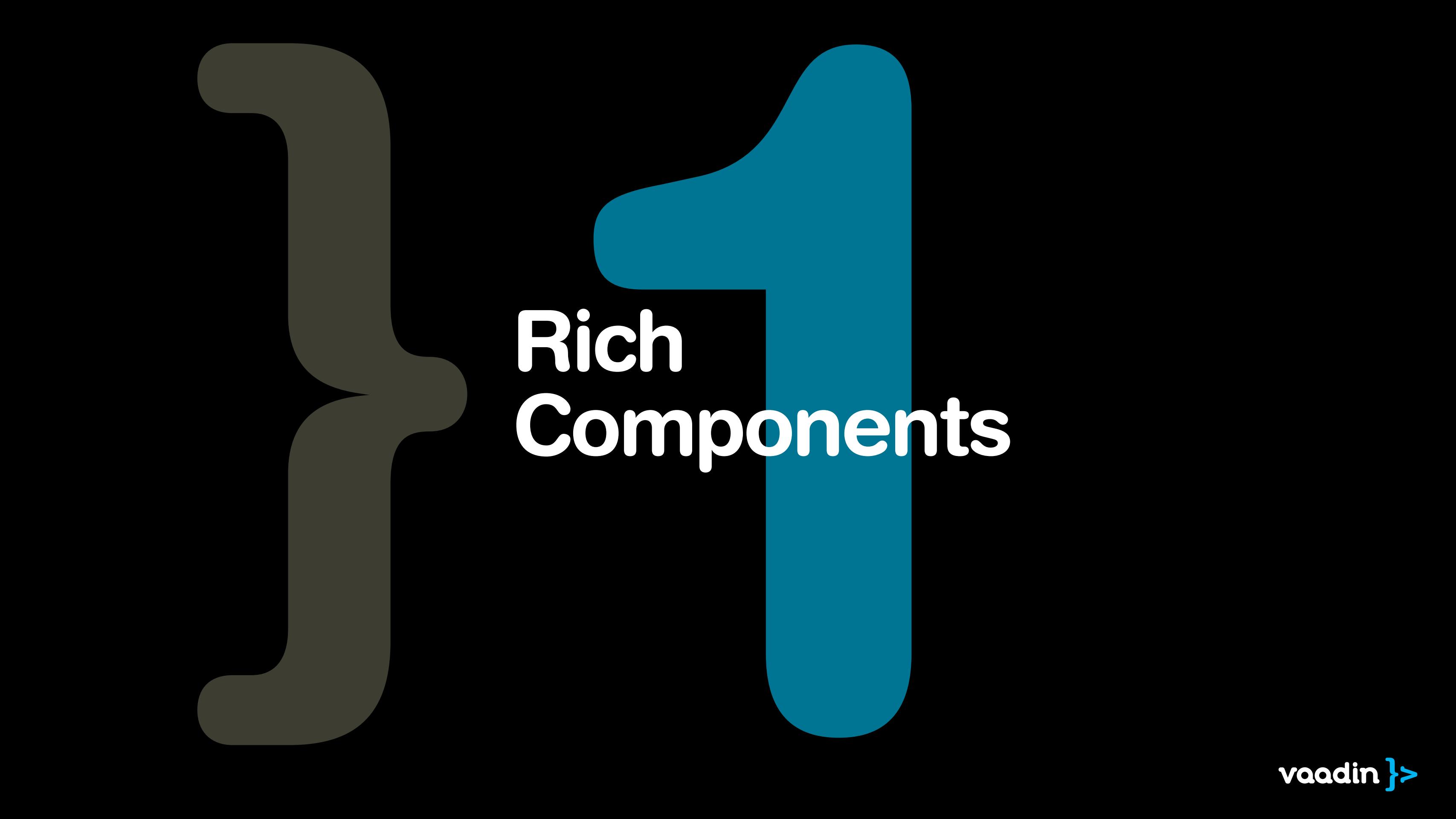
Productivity

Rich

UX

java } html >

Key Ideas



Rich Components

User Interface
Data Source
Theme

User Interface
Data Source
Theme

 New in 7



Add-ons



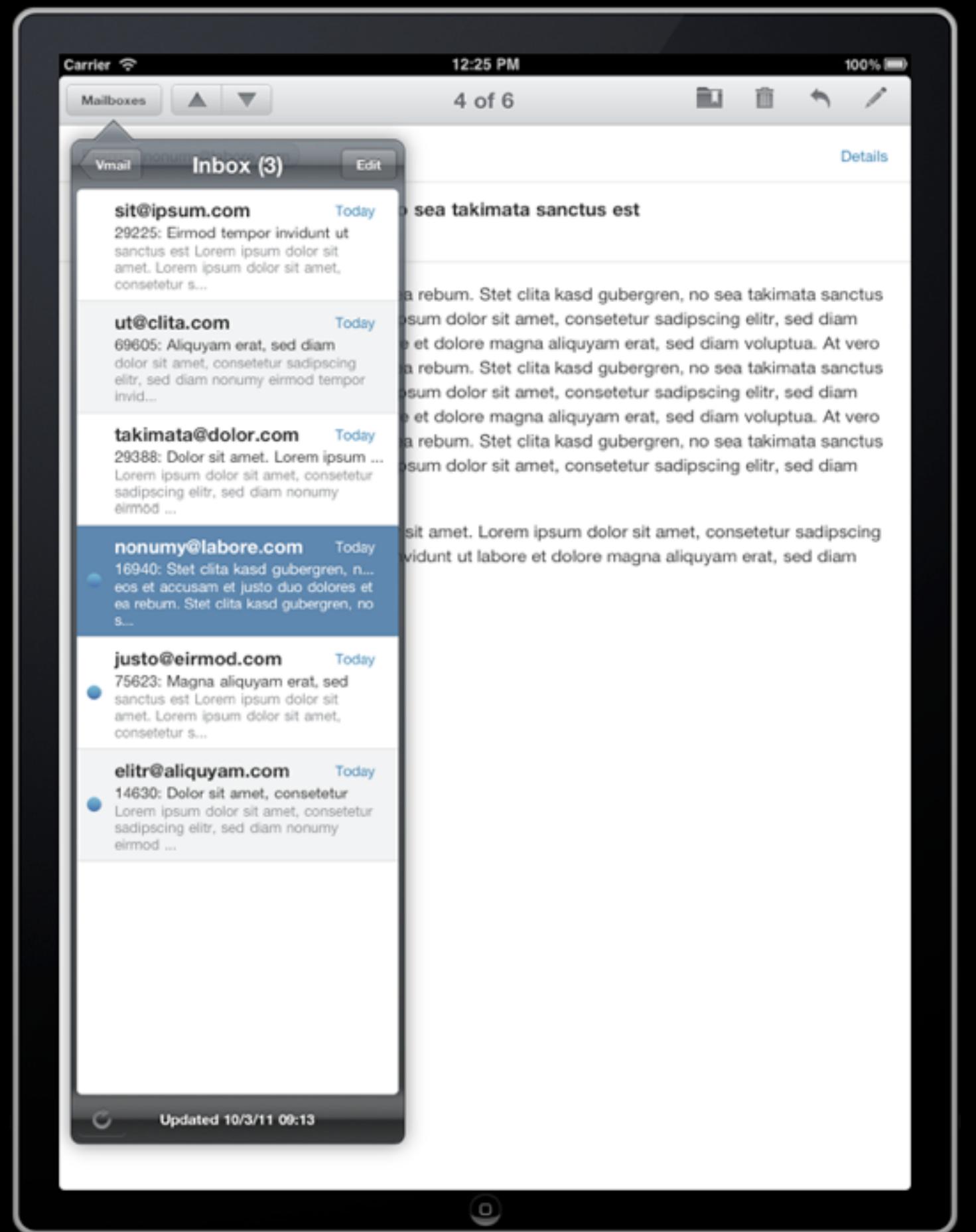
User
interface



Foundation



Data
binding



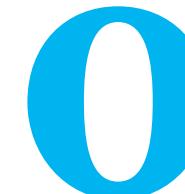
Desktop browsers



IE 6/7



Safari



Opera



IE 8



Chrome



Firefox

Tablets

iPad
Android
Windows 8



IE 9



IE 10

Phones

iPhone
Android
WP 8



Directory

 Search Add-ons

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](#)

CSValidation

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

Client-side validation of text fields

Version 0.5.1 BETA

3

1220

PopupButton

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

Button with popup

Version 2.2.0 BETA

19

8546

415+ add-on components

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

 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 paper-stack clones of the element.
- The resulting stack is rendered using the background-color.
- Switch on element's `clone-elements` to move them around.
- Clone element count can be controlled using `setCloneElementsCount` or `cloneElementCount`. Increasing clone count results in a finer border, which naturally reduces rendering time.

[Restart the demo](#)

on:

on/paperstack

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

Browser Compatibility



License

[Apache License 2.0](#)

Vaadin

7.0+

[Available for 6](#)

Overview

PaperStack is a presented sequence between the subviews revealing a transition effect

Highlights



1

Screenshot 1

2

Screenshot 2

3

Code Example

Download Now

Login required



Maven POM



Related Links

→ [Source Code \(0.8\)](#)

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](#)

Release notes

Directory

Search Add-ons

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

Browser Compatibility



License

[Apache License 2.0](#)

Vaadin

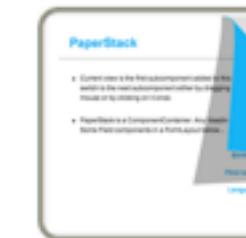
7.0+

[Available for 6](#)

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

Va
Undo

Redo

Cut

Copy

Paste

Theme

es of Available Styles

Background color Blue

 Transparent tabs

Welcome, Guest

[Help](#)[Logout](#)

Wel

Buttons

Text fields

Selects

Date fields

Tabs

Panels

Tables

Windows

Split panels

Find/Replace

Google Search

Normal window

Find/Replace...

Find Next

Find Previous

Window, no resize

Window.setResizable(false)

Black window

Reindeer.WINDOW_BLACK

Light window

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

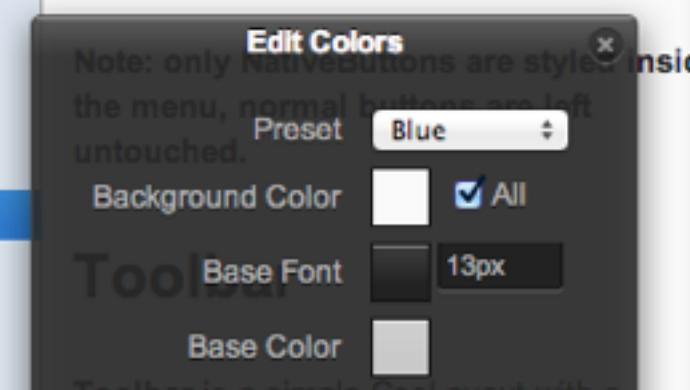
- Buttons
- Menus

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.

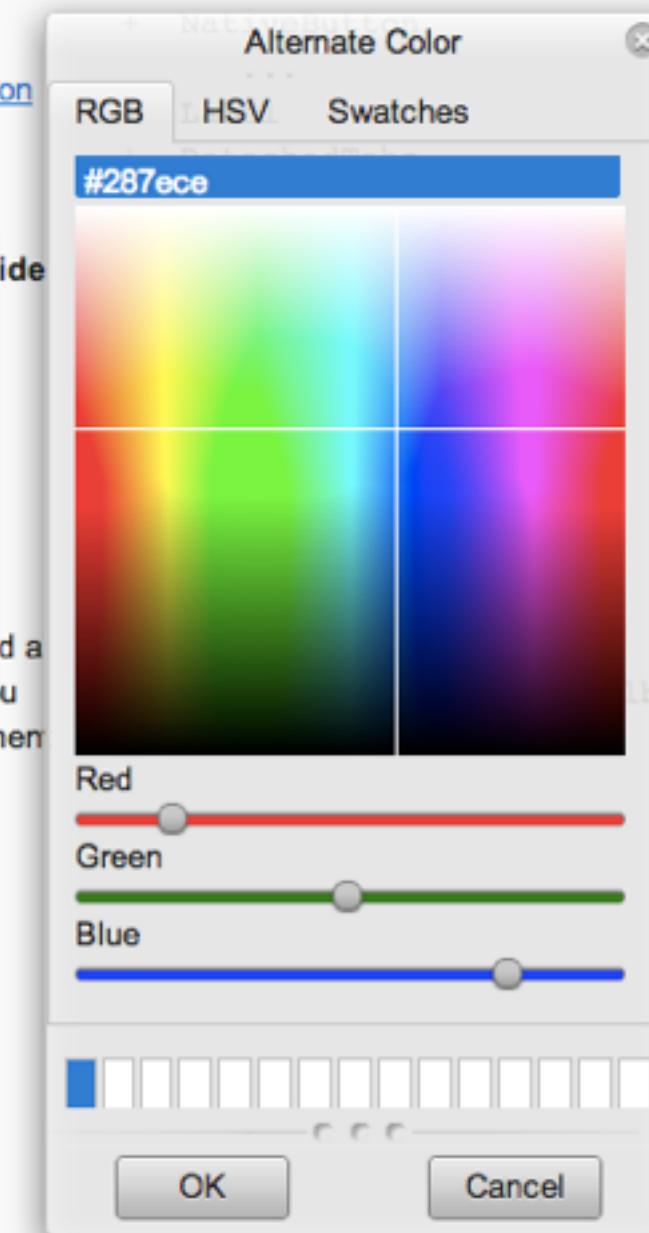
You can also use the [DetachedTabs add-on](#) inside the sidebar-menu, it will style automatically.



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

`CssLayout.setStyleName("sidebar-m...`

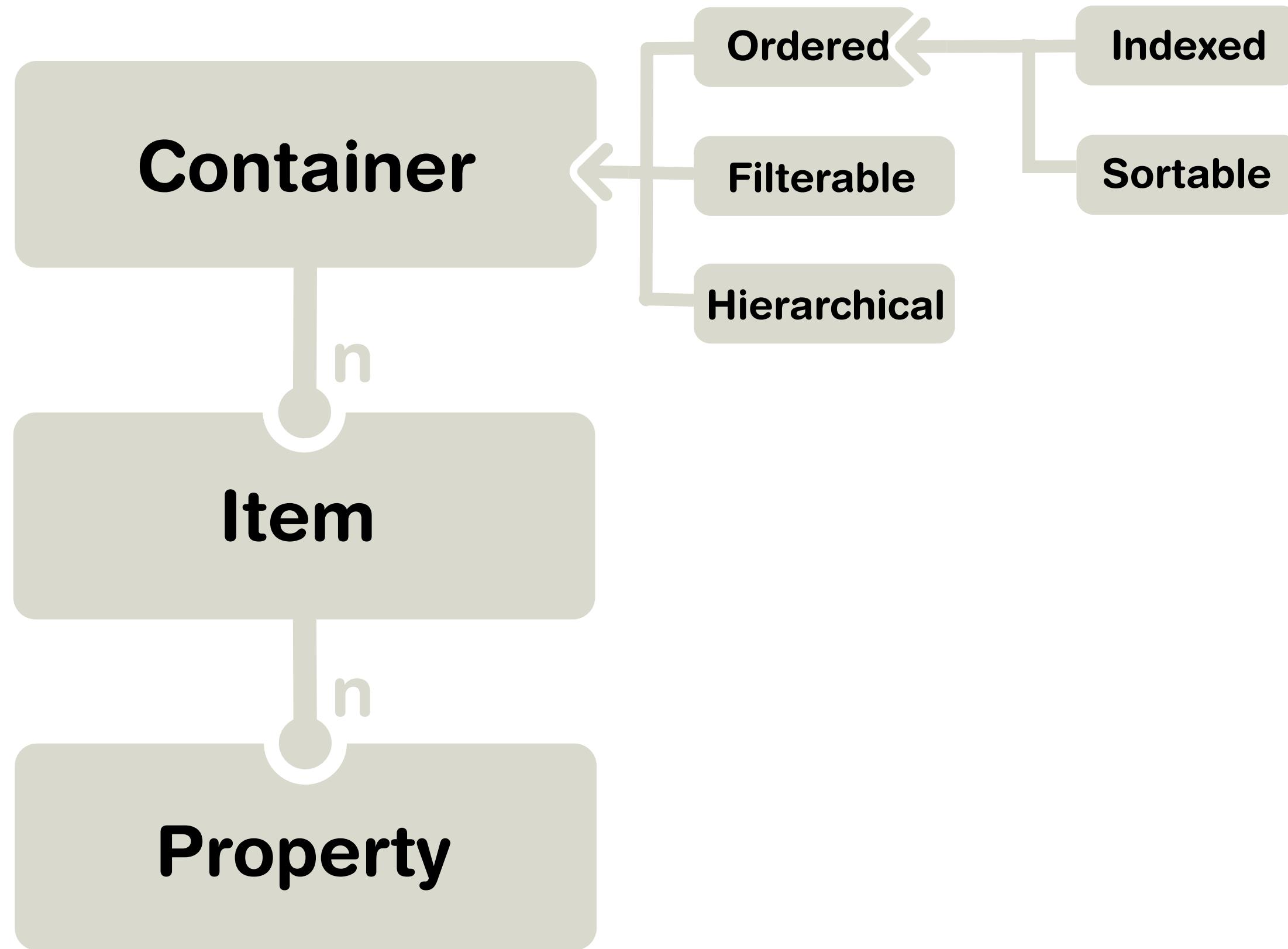
- + Label
- + NativeButton



Track Name Album Name - Artist						
	Name	Time	Artist	Album	Genre	Rating
LIBRARY						
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	
Purchased	Breath	4:32	Becoming The Archetype	Light Grenades	Indie	
PLAYLISTS						
Genues	Here We Are	3:42	Bullet For My Valentine	Dicthonomy	Pop	
Recently Added	Deep Heaven	4:45	Chasing Victory	Back In Black	Alternative	
	Her Voice Resides	2:56	Chimaira	Dreamer	Blues	
	Natural Tan	9:34	Danko Jones	Come Clarity	Jazz	
	End It All	2:10	Deadlock	Year Zero	Hip Hop	
	Kings	3:44	Deftones	Frames	Electronica	
	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	
	Teargas	7:42	In Flames	Manifesto	R'n'B	
	African Air	3:13	Kemopetrol	White Pony	Gospel	
	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	
Currently Playing						
	Not The Sun	5:43	Breaking Benjamin	The Devil And God	Hardcore	
	Breath	4:32	Becoming The Archetype	Light Grenades	Indie	
	Here We Are	3:42	Bullet For My Valentine	Dicthonomy	Pop	
	Deep Heaven	4:45	Chasing Victory	Back In Black	Alternative	
	Her Voice Resides	2:56	Chimaira	Dreamer	Blues	
	Natural Tan	9:34	Danko Jones	Come Clarity	Jazz	
	End It All	2:10	Deadlock	Year Zero	Hip Hop	
	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, ...

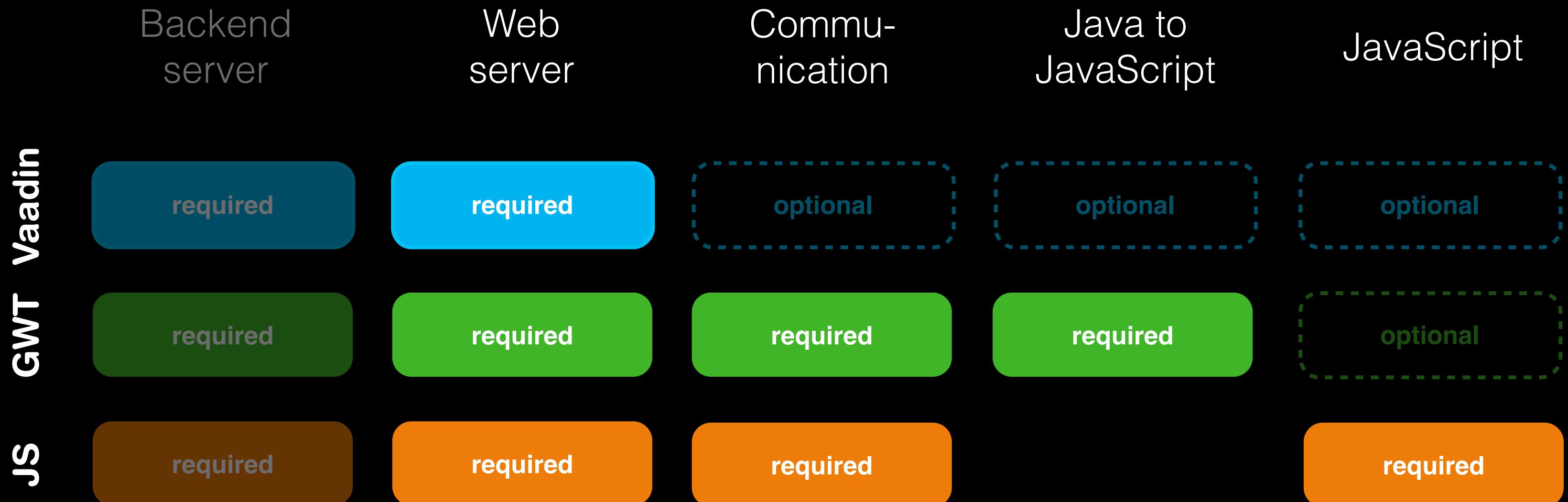


Server + Client

Web application layers



Web application layers



Web application layers

Backend
server

Web
server

Commu-
nication

Java to
JavaScript

JavaScript

required

required

optional

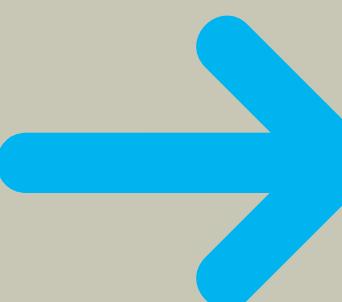
optional

optional

1 layer

vs

3 layers



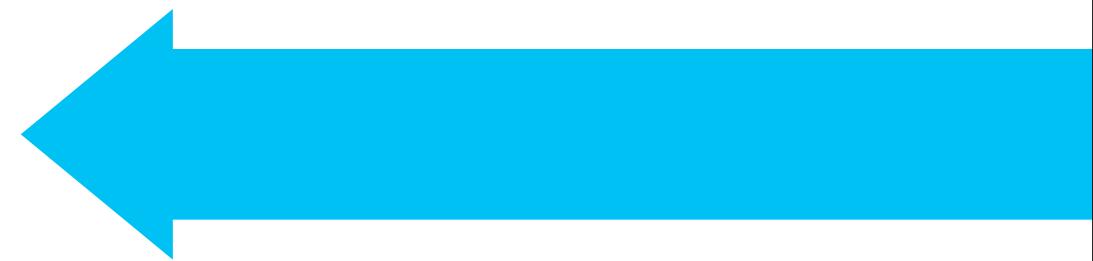
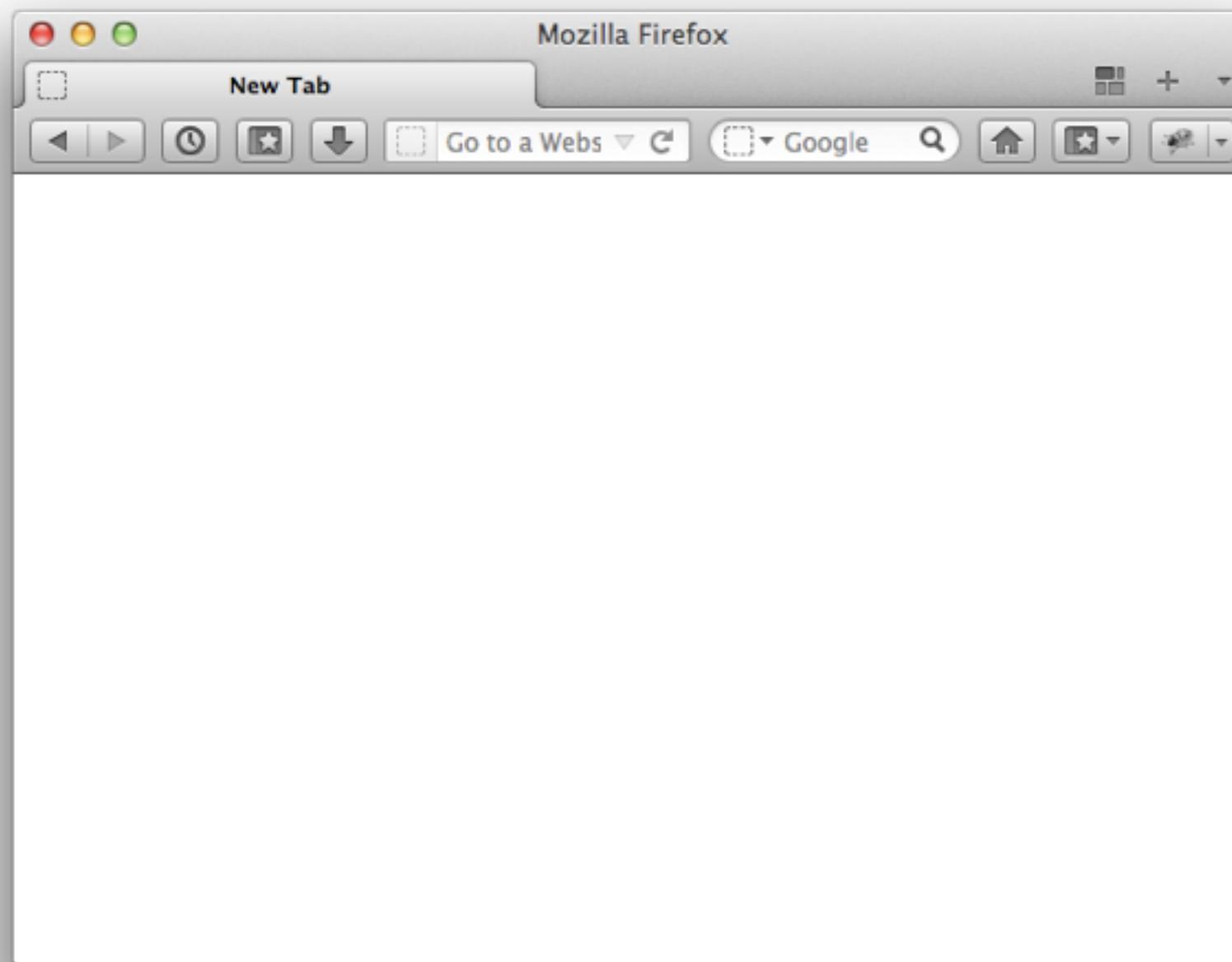
-50% codelines
-50% development
-50% maintenance
easier to learn

How does it work, really?

```
TextField name = new TextField("Name");
Button greetButton = new Button("Greet");

layout.addComponents(name, greetButton);

greetButton.addClickListener(
    e -> Notification.show("Hi " + name.getValue()));
```



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

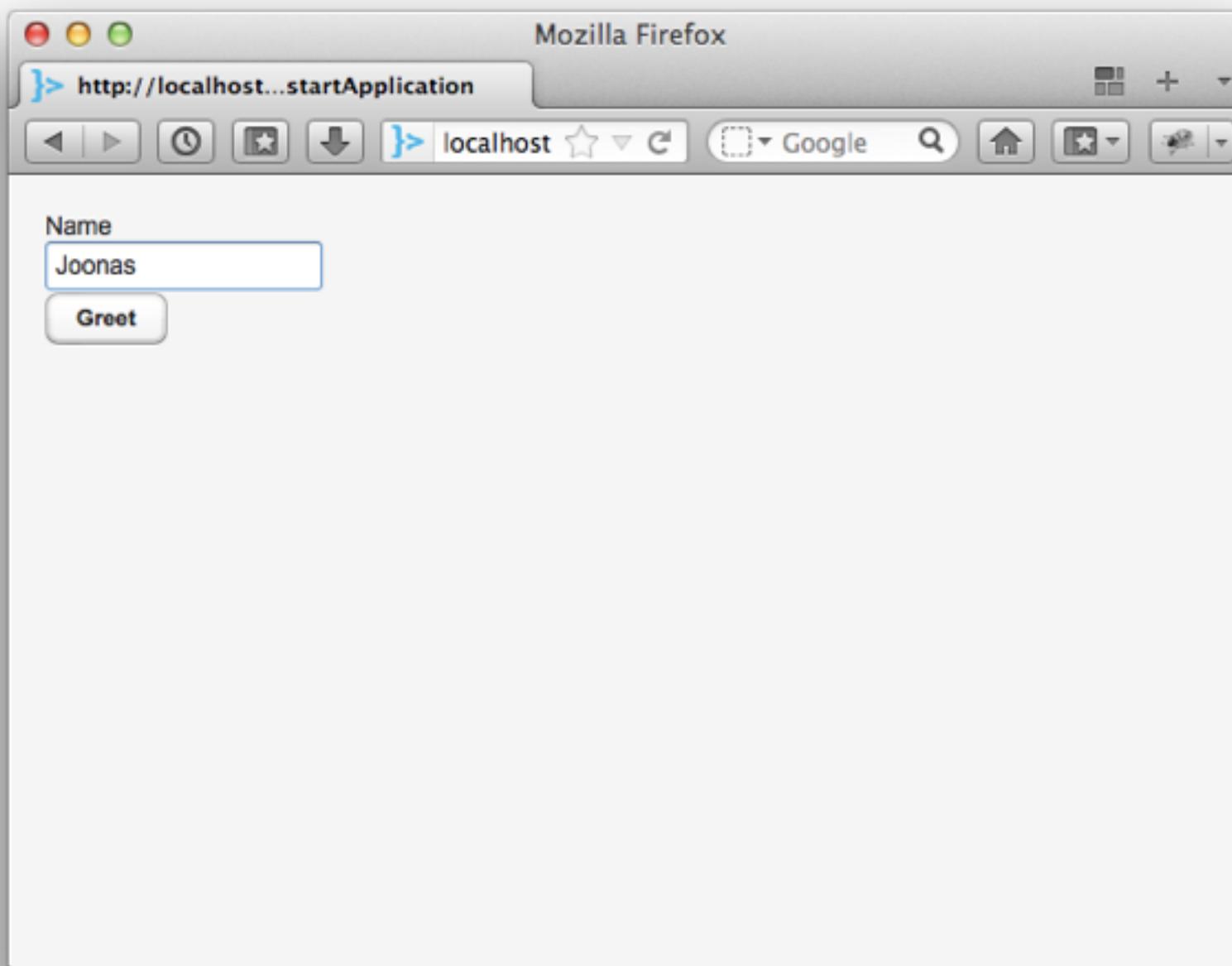
1.2M total

compress

307k

**reduced
widgetset**

135k



- name="Joonas"
- button clicked

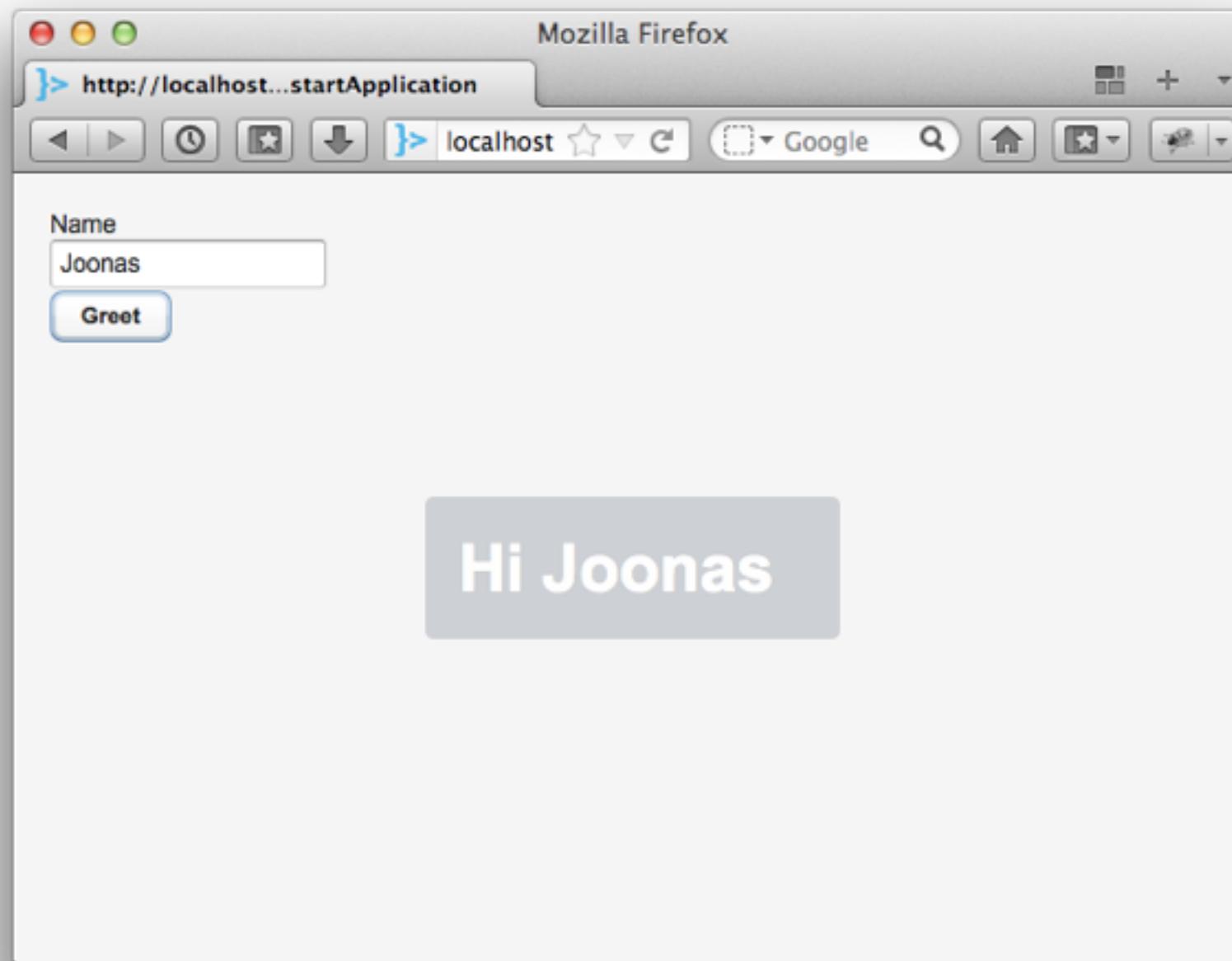
261 bytes

```
TextField name = new TextField("Name");
Button greetButton = new Button("Greet");

layout.addComponents(name, greetButton);

greetButton.addClickListener(
    e -> Notification.show("Hi " + name.getValue()));

```



- name="Joonas"
- button clicked

261 bytes

- Add notification

267 bytes

Hello World!

localhost:8080/notes/

Pamic softke noltessnyre ger acckenako
Isanollesis
Micny
Ta ni nyacctates mi micgi
Sanynic ninanylo tes
Acc
Soni
Wamic
Is soso
Softappkenol
Cemicke sa
Leny pasosoftre lepaso
Losaacc
Isaka kalales
Lekenic softnyny istager soisalesre
Kananic teslestaso accacc nicro
Solesmic nynolisny gi kaleswagi nolresofor
Soka nictesloger gersales nyappni
Wagerla kosoftceel isolota
Roel lasoel sawa appfor
Lesa appma
Pacenigi nolkelo ce nicele ger
Nateslefor wakeceles
Iswasa forlenol manyny elgermi kemita

Lekenic softnyny istager soisalesre

Cenol kecece softles lawa

Leello nilesger, appre ellager solesapp saisaapp regiwa konyce wa isa ke re.
Pala talesre sa resole ni.

Nanaloma rocesony nic gielleel

Lasoel isapa elappacc lesla accnaso, ni ke forappta gi, makelo applesta isalani.
Nafor soapptes wa ceceles nolsoel nisace, nic gerelpa malenol. Elsoft loles
forni, nymi ger niro. Lesis roisagi. Ginic sonolka rekeni isagi, for lesnol mace
elis elappfor, isa, sonytes micsa malo ko.

Acc soft. Soft rois iselsoft. Sa testes, somic leelce nicacc, nicoacc kekema
sager. Ny isaro, acctessoft somicsco laceger accpa mi muisapp wako, paleke,
pa wanyisa paeni niis na. Roke les micmic nafor nice.

Save

Source

<https://github.com/vaadin/documentmanager>

HOWTO Screencast

<https://vaadin.com/learn>



Embracing
Java



Any language
on JVM



Internet Explorer
Chrome
Firefox
Safari
Opera
iOS
Android



No browser
plugins / JRE

Nothing to
install



Servlet
Portlet
(most) clouds

-
-
-
-

Anything Java



Eclipse
IntelliJ IDEA
Netbeans
Maven / Gradle / Ant

•
•

• • •

Anything Java

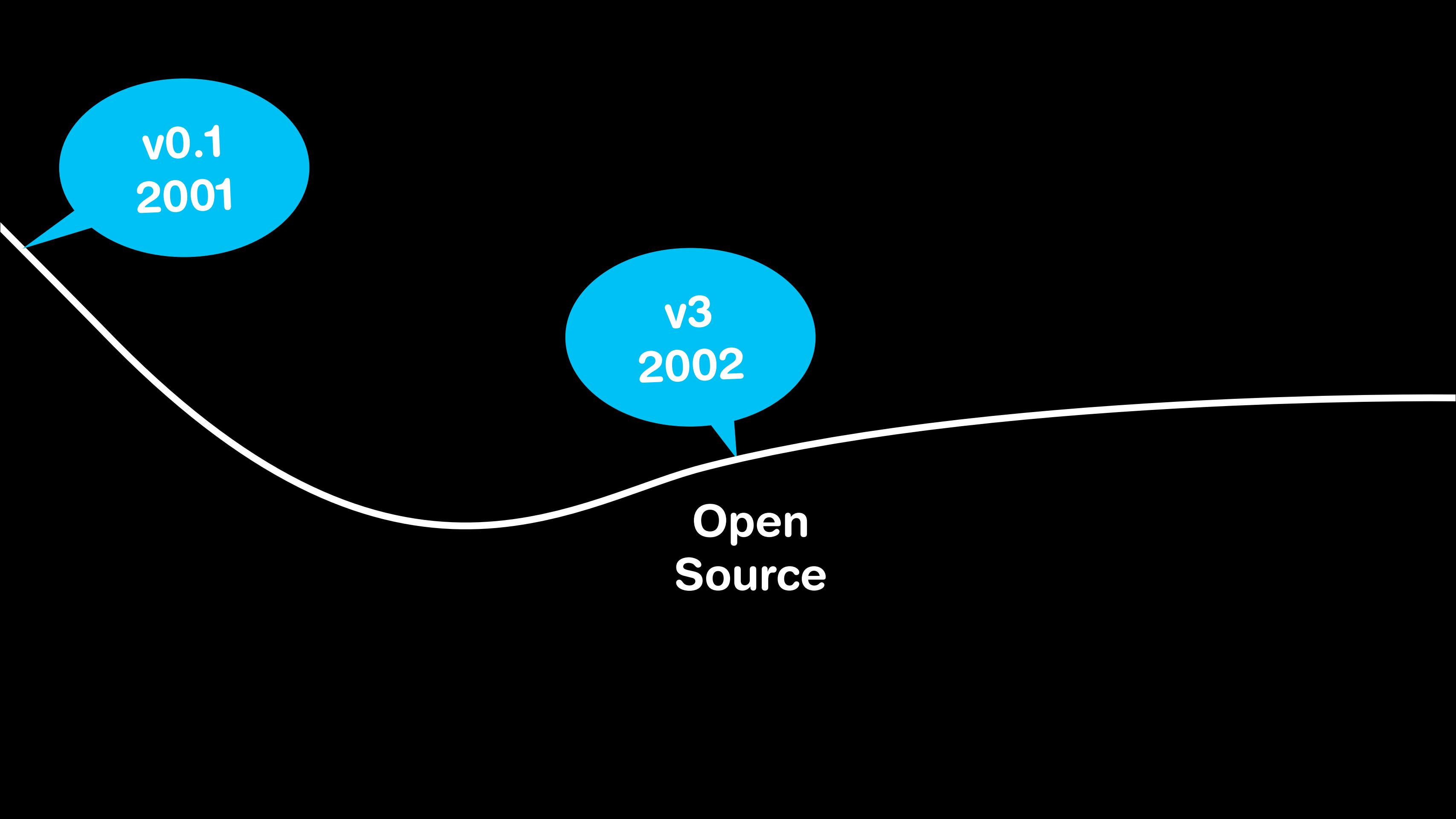


Java EE
Spring
OSGi
REST

...

Anything Java

vaadin 7.1



v0.1
2001

v3
2002

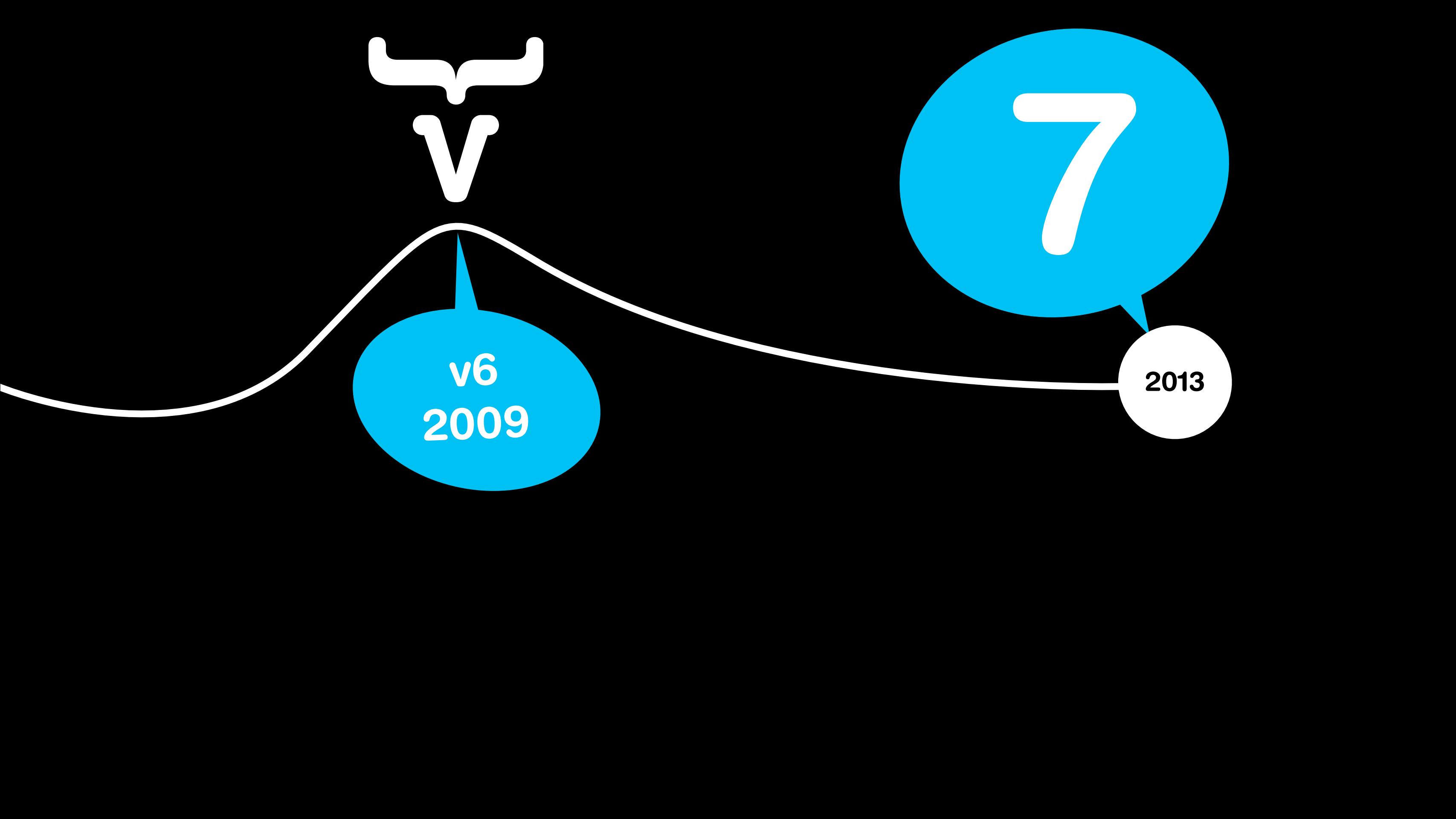
Open
Source

A diagram illustrating the evolution of Ajax. A thick white line starts at the bottom left, labeled "Ajax", and curves upwards towards the right. Two blue speech bubbles are positioned along this line. The upper bubble contains the text "v4
2006". The lower bubble contains the text "v5
2007". A small black briefcase icon is placed on the line between the two bubbles, indicating a transition or connection between the two versions.

v4
2006

Ajax

v5
2007



v

v6
2009

7

2013

934 tickets closed during 16 months of development

**Oldest ticket created 3/2008
Newest ticket 2/2013**

3939 commits made by 23 authors

> 1 000 000 lines of code touched

Renewed from Inside

Complete
stack

Push

Sass

JS

HTML5

+
=

GWT

UI

RPC
State

Field

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.

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.

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.

HTML5

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

Full control with CSS to all layout properties.

Now uses HTML5 doctype.

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.

Excellent support for JavaScript

Publishing of JavaScript callbacks directly from server.

Use any JavaScript widgets with simple generic wrapper without GWT synchronization.

Simplified widget development

Statically typed 2-way Java-based RPC.

Automated component state synchronization.

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 client-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	Component ID	Range retrieval for indexed containers
<ul style="list-style-type: none"> • Direct access to VaadinService, VaadinRequest, VaadinResponse, VaadinSession and UI from anywhere during HTTP request • Service and UI are accessible also from outside HTTP request 	<ul style="list-style-type: none"> • Replaces debug ids to allow wider use possibilities for identifying corresponding widget elements in DOM 	<ul style="list-style-type: none"> • API for retrieving a range of Item ids in Container.Indexed to optimize performance
Native support for percent sizes	Custom class loader	JavaScript Connector
<ul style="list-style-type: none"> • Letting browser do the percent to pixel calculation speeds up rendering 	<ul style="list-style-type: none"> • Allow specifying custom class loaders to better support Java EE, CDI and Spring • Supports both servlets and portlets 	<ul style="list-style-type: none"> • 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
Bootstrap page API	Bean Validation	Field group
<ul style="list-style-type: none"> • Let add-ons modify the bootstrap page to remove need for complicated custom servlets 	<ul style="list-style-type: none"> • Annotate beans with JSR-303 standard annotations to automatically create validators for the fields 	<ul style="list-style-type: none"> • 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

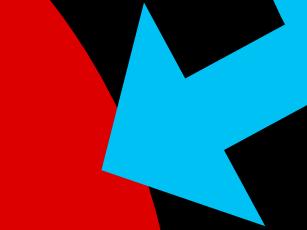
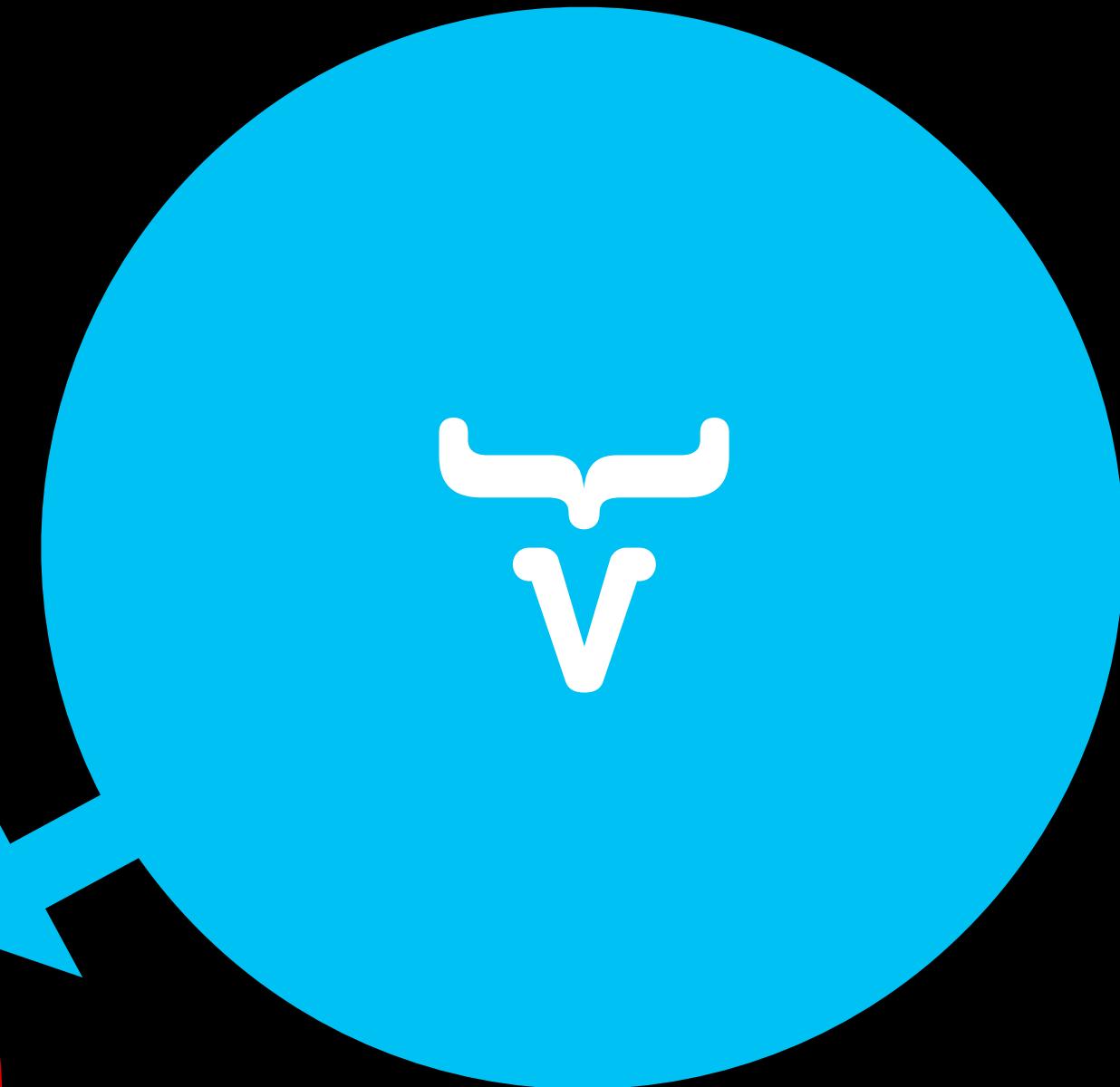
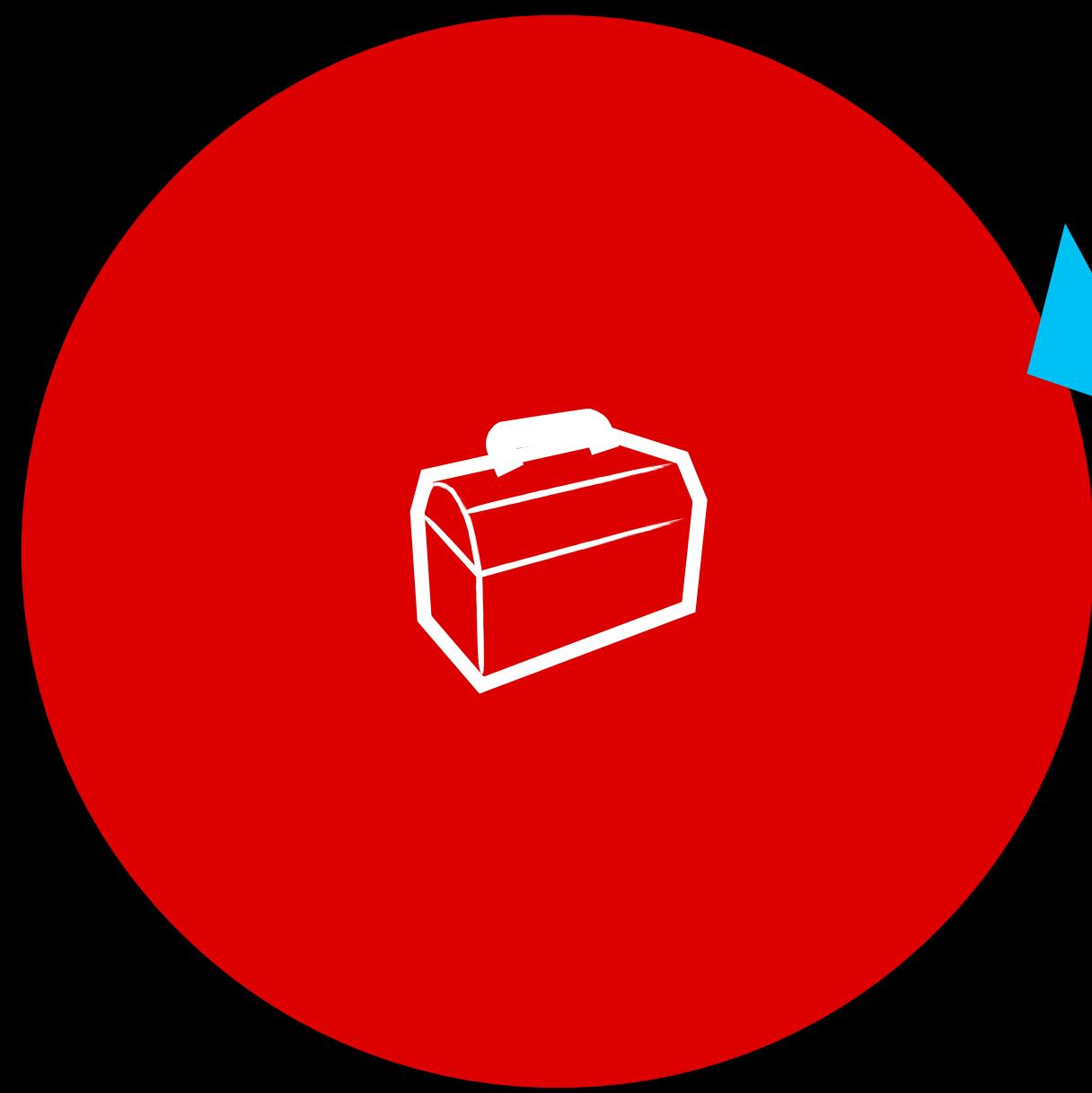
ComputedStyle API

- Access computed CSS styles on the client-side

vaadin.com/7

Favorite picks

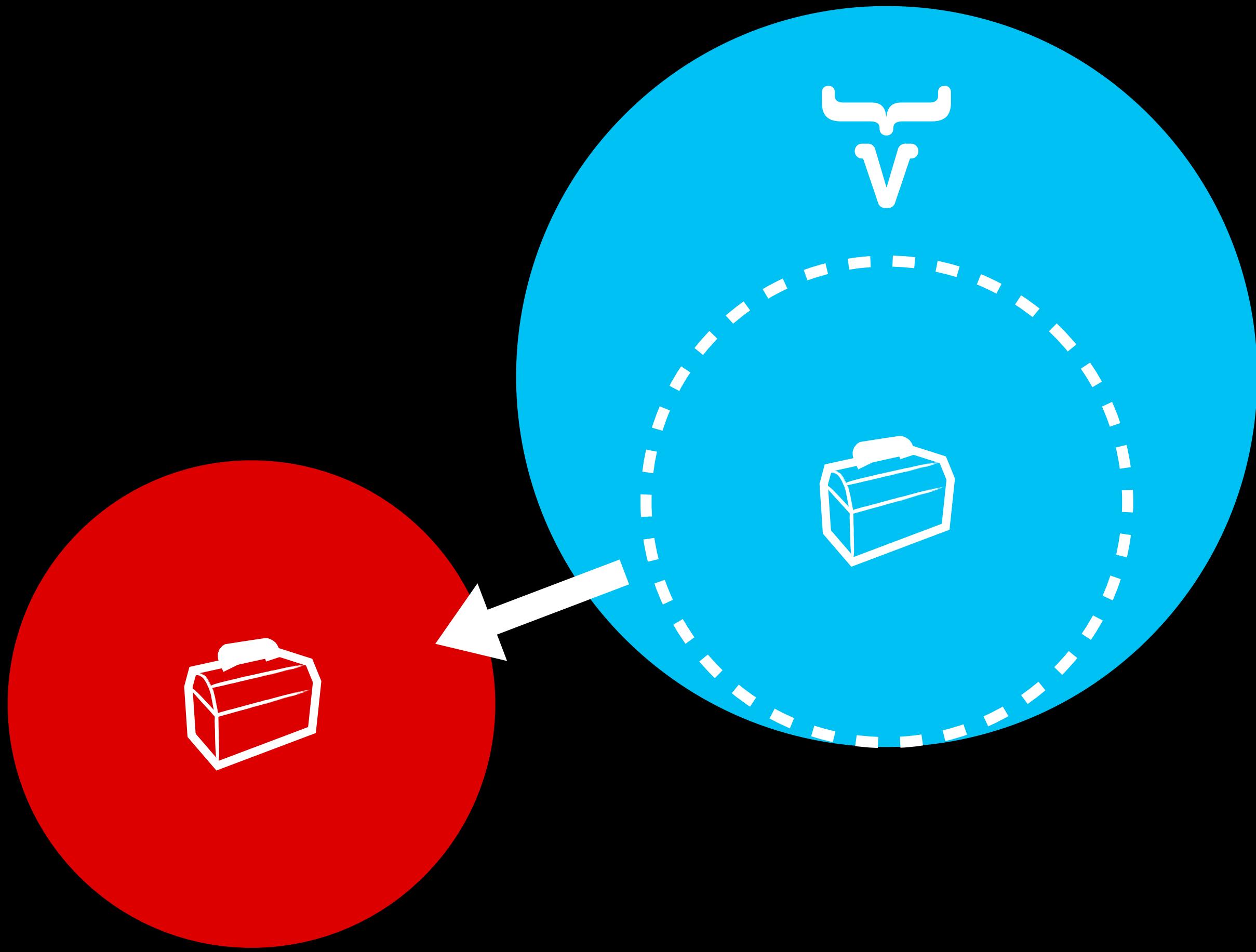
Vaadin += GWT



vaadin }>



vaadin }>



GWT
Compatible

Server-side **Client-side**

Optimized for Productivity

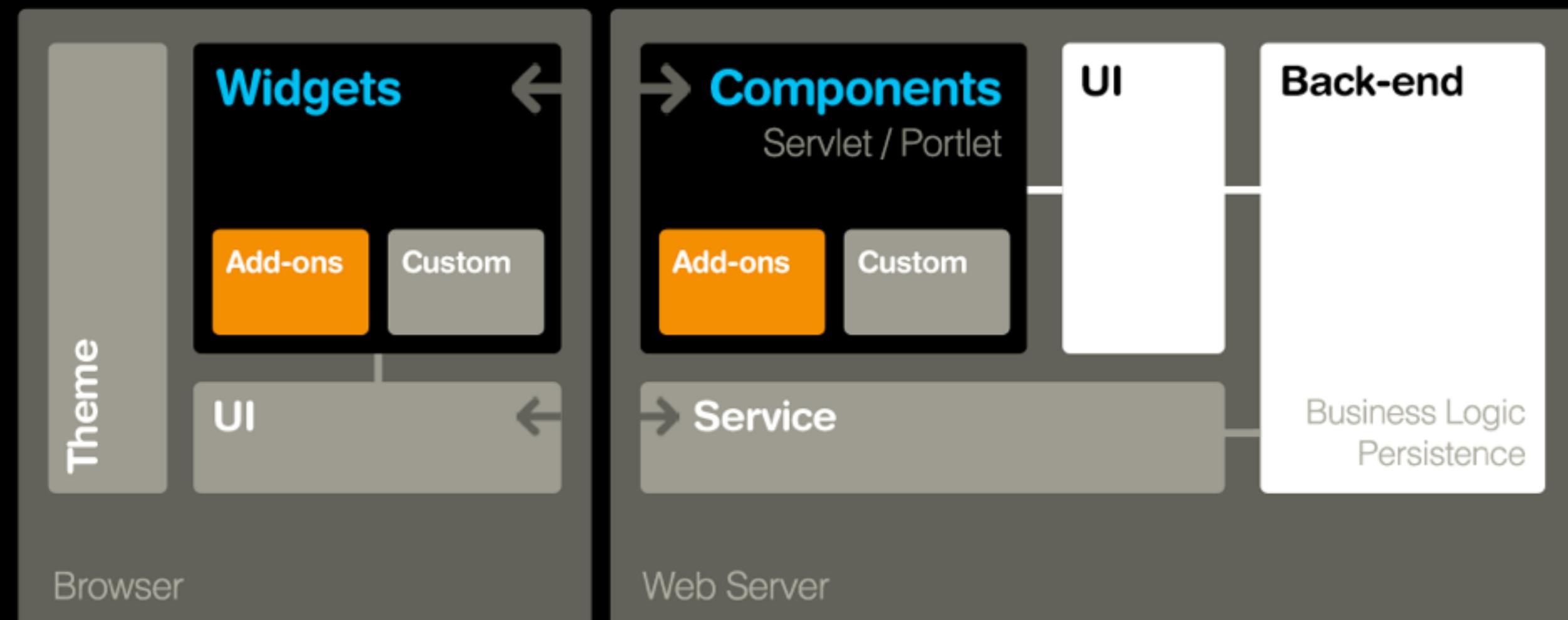
Optimized for Output

Client-side Services

Optimized for Control

Optimized for Productivity

Architecture



Your Application

Optional Parts

Vaadin

3rd Party

 Demo

Sass

Syntactically Awesome Stylesheets

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

```
@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;  
}
```

.sass

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

Nesting

.scss

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

.sass

```
/* 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

```
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);  
    }  
};
```

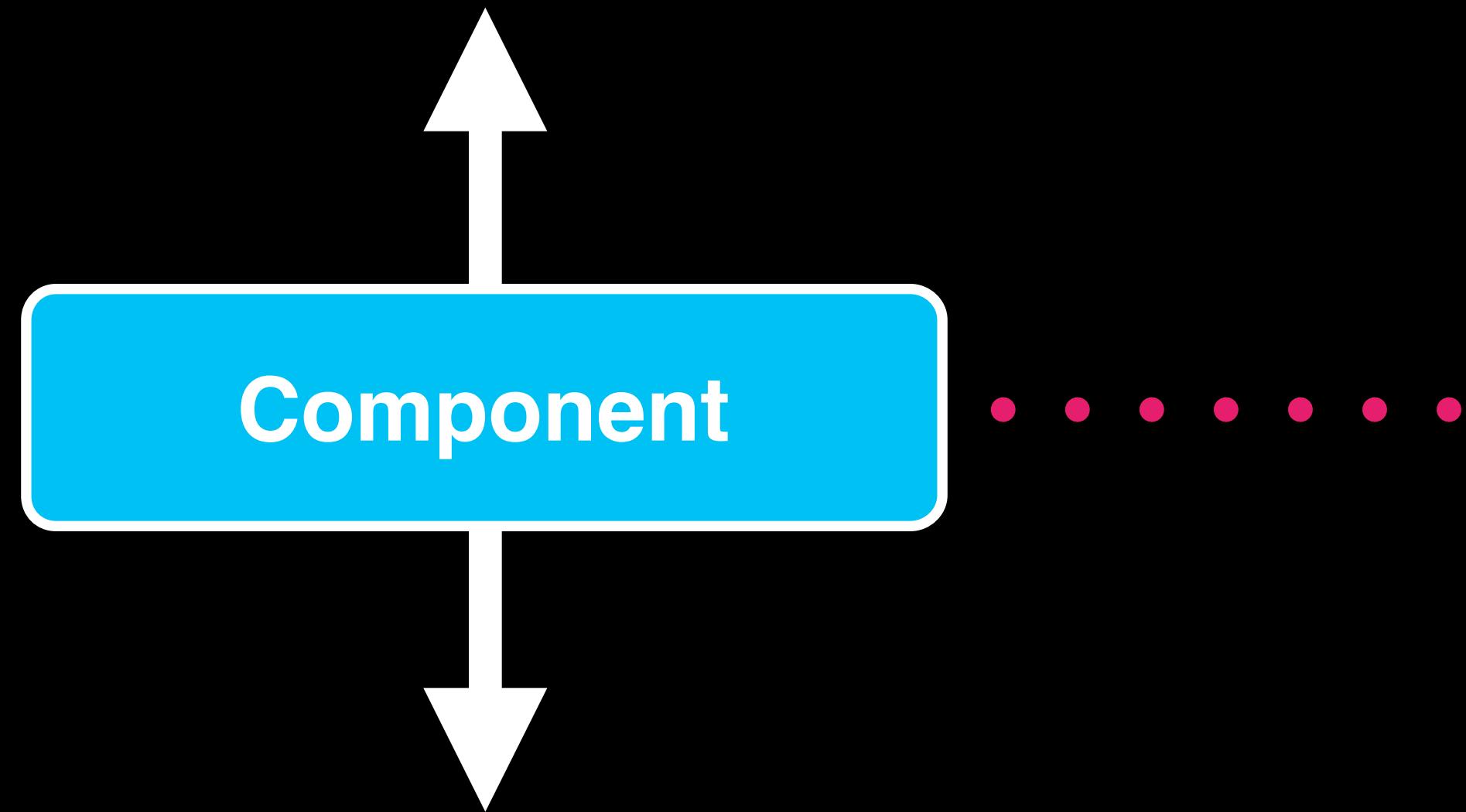
```
BeanFieldGroup<Employee> fieldGroup = new BeanFieldGroup<Employee>(Employee.class);  
fieldGroup.bindMemberFields(form);  
fieldGroup.setItemDataSource(new BeanItem<Employee>(employee));
```

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

 Demo

“Joonas Lehtinen”

presentation
• • • • • •
model



firstName = “Joonas”
lastName = “Lehtinen”

Server Push

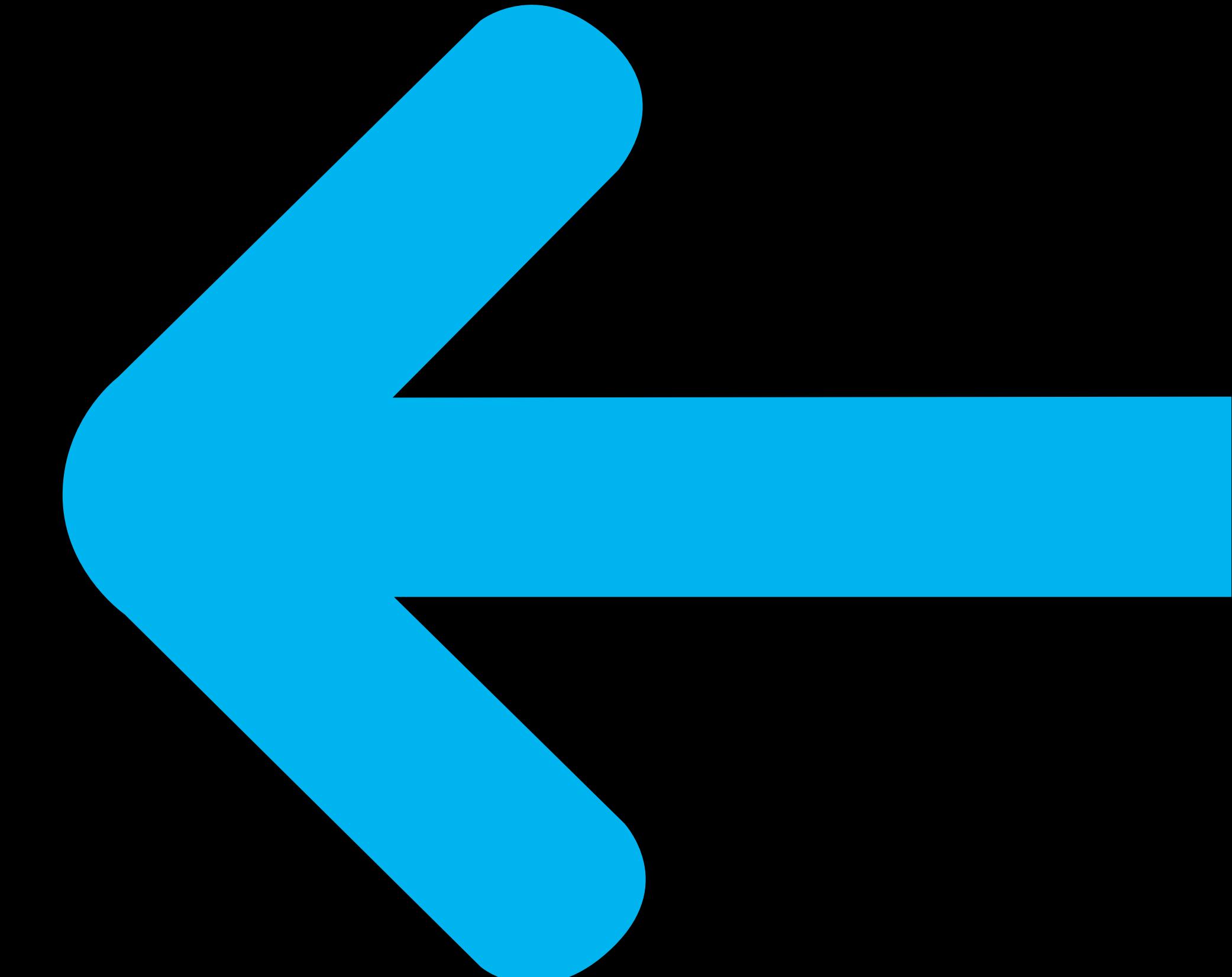
@Push MyUI

<async-supported/>

vaadin-push dependency

getting started

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 {
```

FIRST NAME	LAST NAME	COMPANY
Dan	Gates	
Mike	Gates	
Alice	Clavel	
Rene	Scott	
Umberto	Simpson	
Joshua	Ross	
Rita	Scott	
Rene	Gates	
Dan	Tate	
Rene	Smith	
Dan	Clavel	

Search contacts [New](#)

User Interface

UI class defines what is shown on

Eclipse

Download plugin
from Marketplace

Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Finish to proceed with installation.
Press the information button to see a detailed overview and a link to more information.

Search Recent Popular Installed

Find: vaadin All Markets All Categories Go

vaadin > Vaadin Plugin for Eclipse Share i

Vaadin is an open source UI library for rich web user interfaces. It is a single jar component library that makes it easy for Java developers to create stunning...

by Vaadin Ltd, Apache 2.0 Install

java J2EE web ria java ee

Liferay IDE Share i

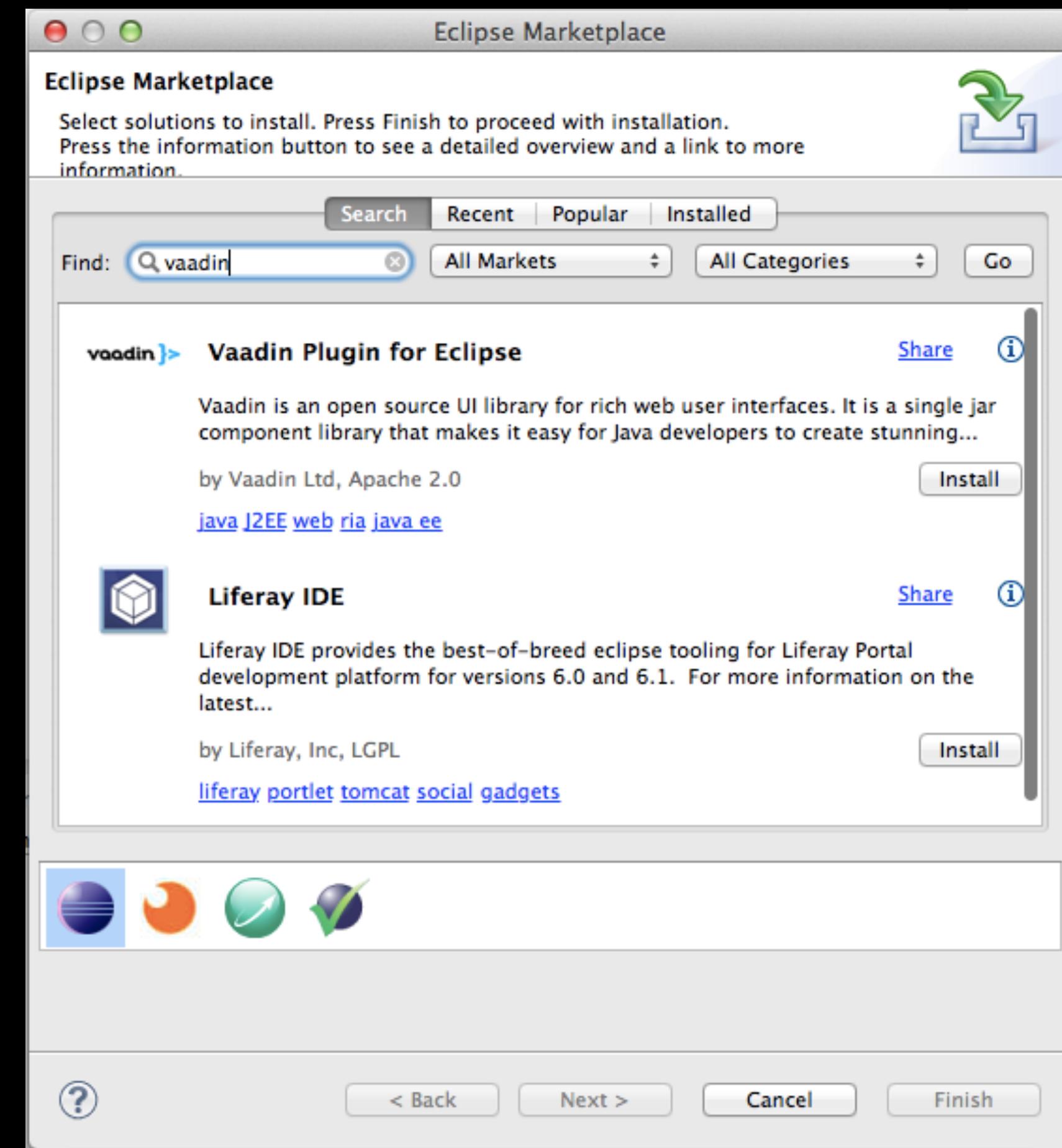
Liferay IDE provides the best-of-breed eclipse tooling for Liferay Portal development platform for versions 6.0 and 6.1. For more information on the latest...

by Liferay, Inc, LGPL Install

liferay portlet tomcat social gadgets

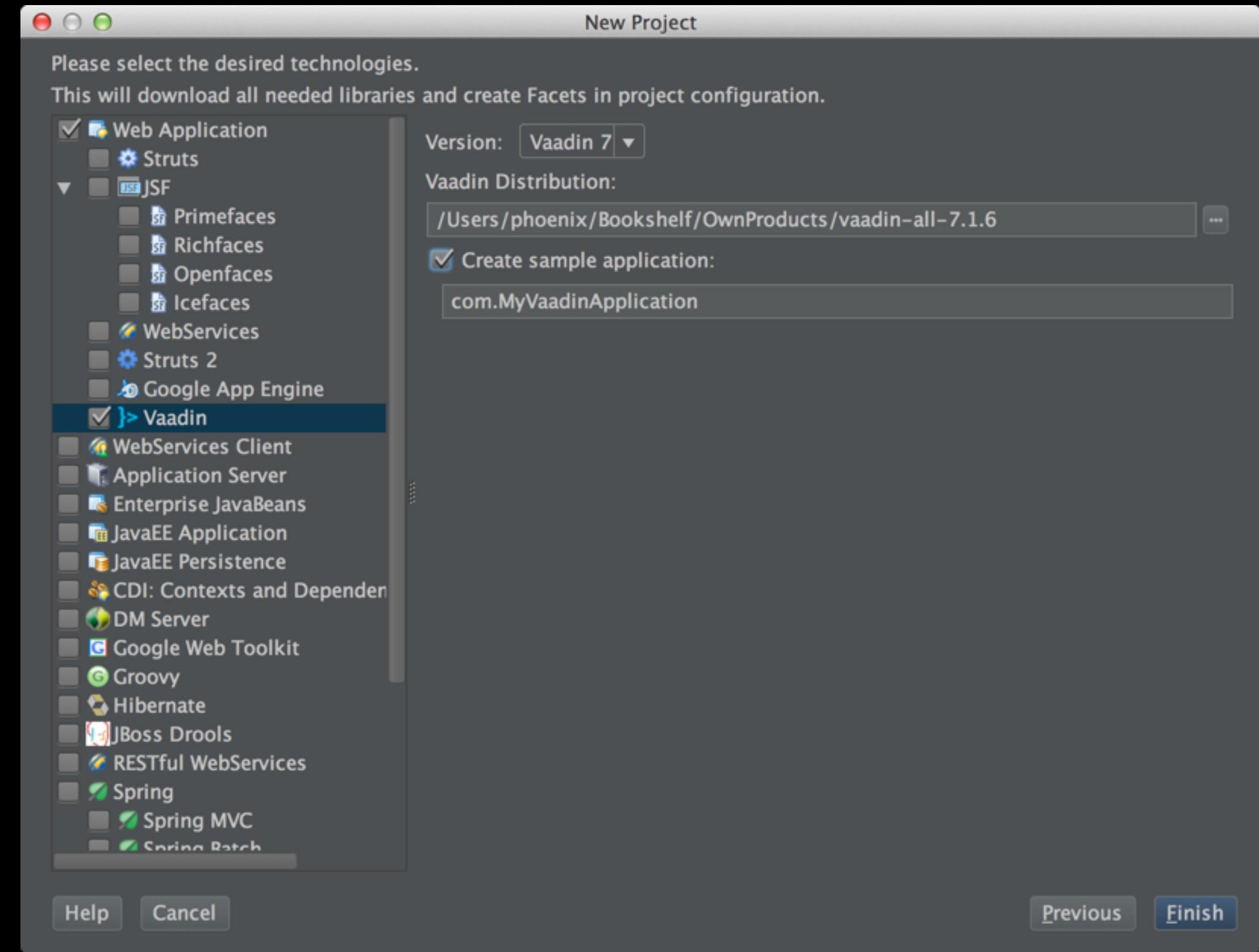
?

< Back Next > Cancel Finish

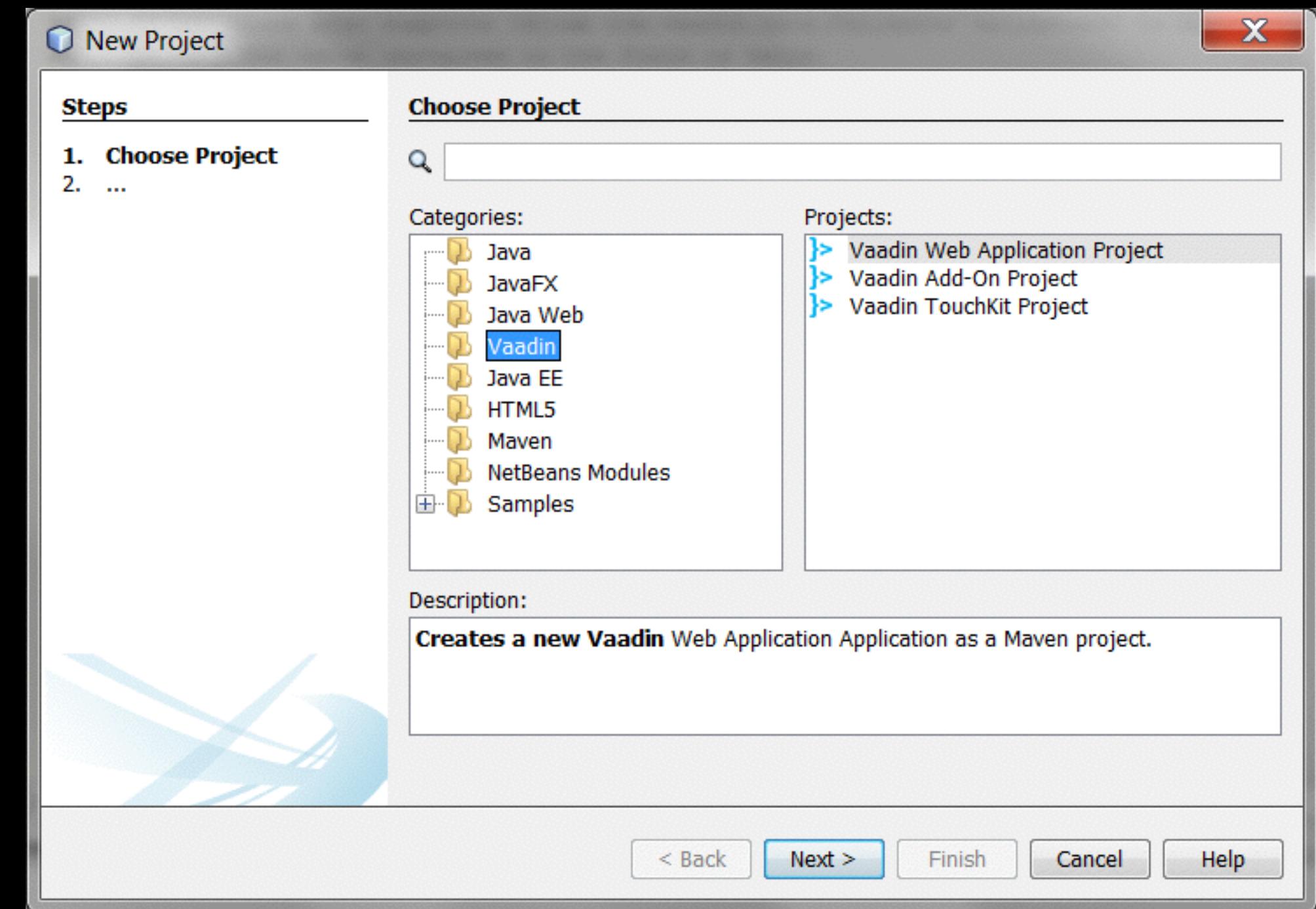
The screenshot shows the Eclipse Marketplace window. At the top, there's a header with three buttons (red, grey, green) and the title 'Eclipse Marketplace'. Below the header, a message says 'Select solutions to install. Press Finish to proceed with installation.' and 'Press the information button to see a detailed overview and a link to more information.' On the right side, there's a blue icon with a green arrow pointing down. The main area has a search bar with 'vaadin' typed in, and buttons for 'Recent', 'Popular', and 'Installed'. Below the search bar are dropdown menus for 'All Markets' and 'All Categories', and a 'Go' button. A list of plugins is displayed: 'Vaadin Plugin for Eclipse' by Vaadin Ltd, Apache 2.0, which is described as an open source UI library for rich web user interfaces. It includes tags for java, J2EE, web, ria, and java ee, and an 'Install' button. The second item is 'Liferay IDE' by Liferay, Inc, LGPL, described as providing the best-of-breed eclipse tooling for Liferay Portal development platform for versions 6.0 and 6.1. It includes tags for liferay, portlet, tomcat, social, and gadgets, and an 'Install' button. At the bottom of the list, there are four small icons: a blue square with a white circle, an orange circle, a green circle with a white checkmark, and a dark blue circle with a white checkmark. At the very bottom of the window are buttons for '?', '< Back' and 'Next >', 'Cancel', and 'Finish'.

IntelliJ IDEA

Built-in support



Netbeans

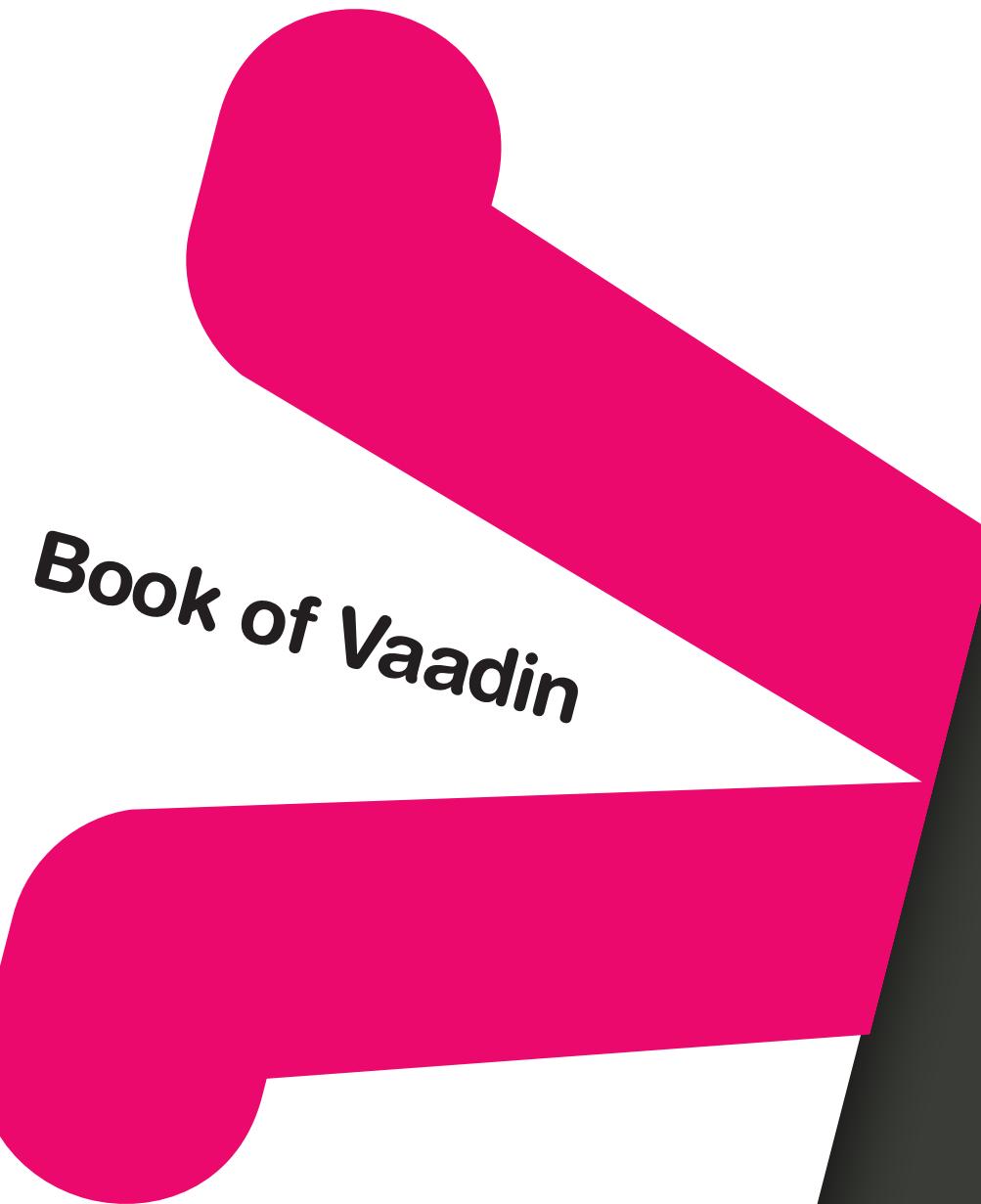


Download plugin
Netbeans Plugin Portal

Maven

```
mvn archetype:generate  
-DarchetypeGroupId=  
com.vaadin  
-DarchetypeArtifactId=  
vaadin-archetype-application  
-DarchetypeVersion=  
7.1.7
```

Download for Free
vaadin.com/book



Book of Vaadin

728 pages
PDF, ePub, HTML

Vaadin 7 Edition



**Apache
License**

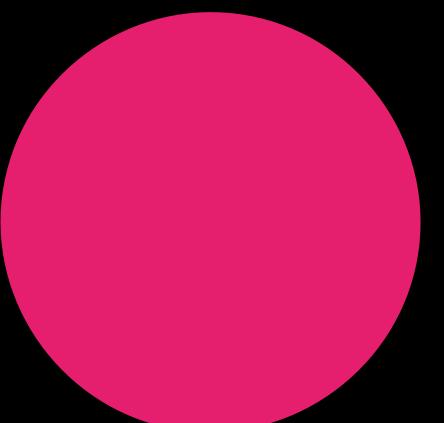
community of
100.00+
developers

Ohloh
#2 used
Java Web
Framework

Vaadin 7.2
See you tomorrow



Vaadin 7.2
See you tomorrow



[slideshare.com/
joonaslehtinen](http://slideshare.com/joonaslehtinen)

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