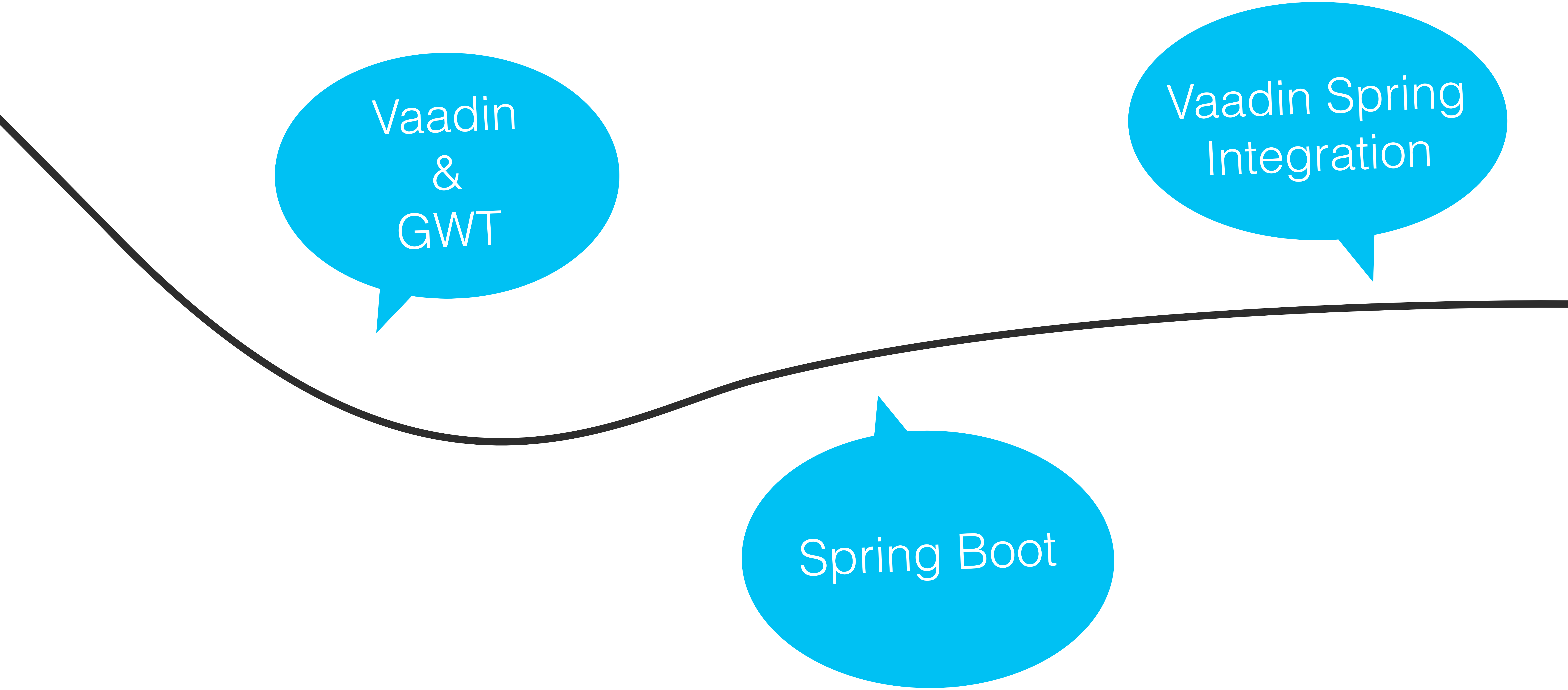
JavaCro

Remote controlling Parrot AR Drone with Spring Boot & Vaadin

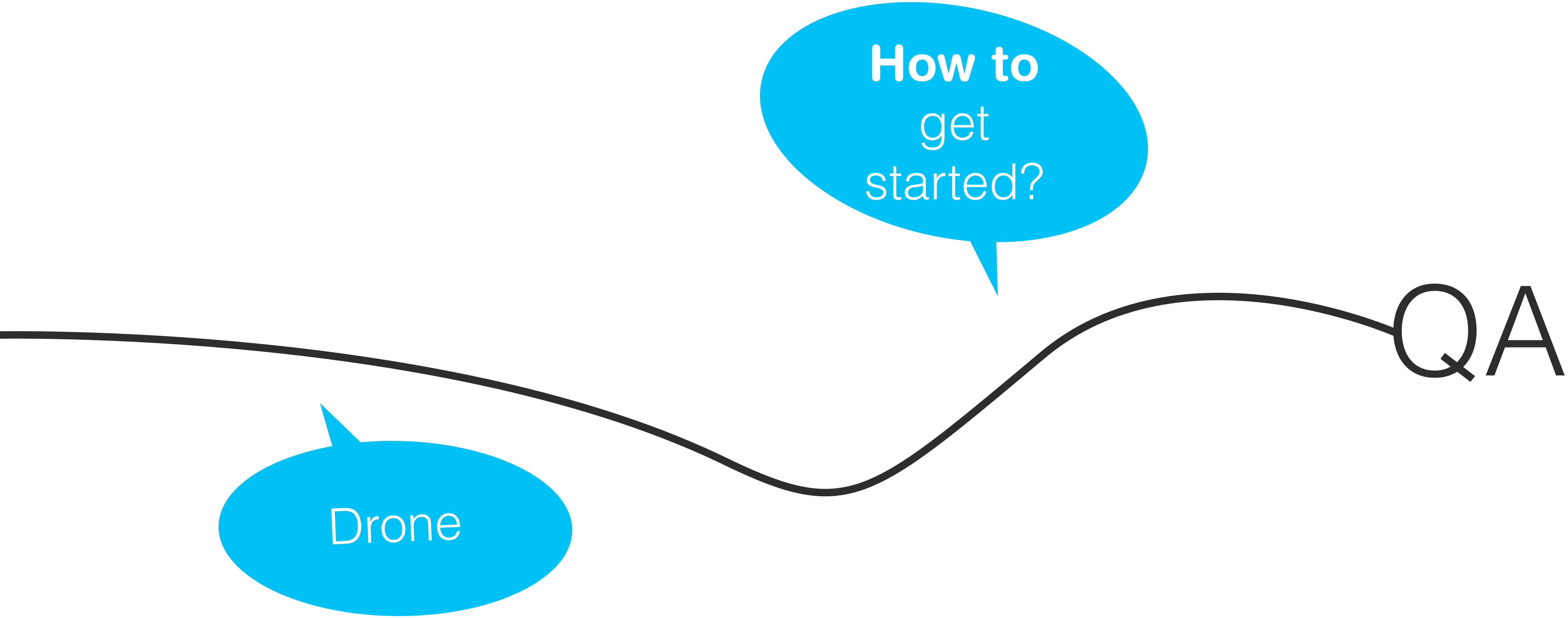
Peter Lehto
@peter_lehto
expert & trainer
vaadin }>



Vaadin
&
GWT

Spring Boot

Vaadin Spring Integration



How to get started?

Drone

QA

vaadin }>

vaadin }>

Server driven UI framework

Developer
Productivity

+

Rich
UX

```
> var foo = [0];
> foo == !foo;
> var foo = [0];
> foo == !foo;
> true
> var foo = [0];
> foo == !foo;
> true

> [] + [];
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>
```

```
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> [object Object]
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> [object Object]

> {} + [];
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> [object Object]

> {} + [];
> 0
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> [object Object]

> {} + [];
> 0
```

```
> {} + {};
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> [object Object]

> {} + [];
> 0
```

```
> {} + {};
> NaN
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> [object Object]

> {} + [];
> 0
```

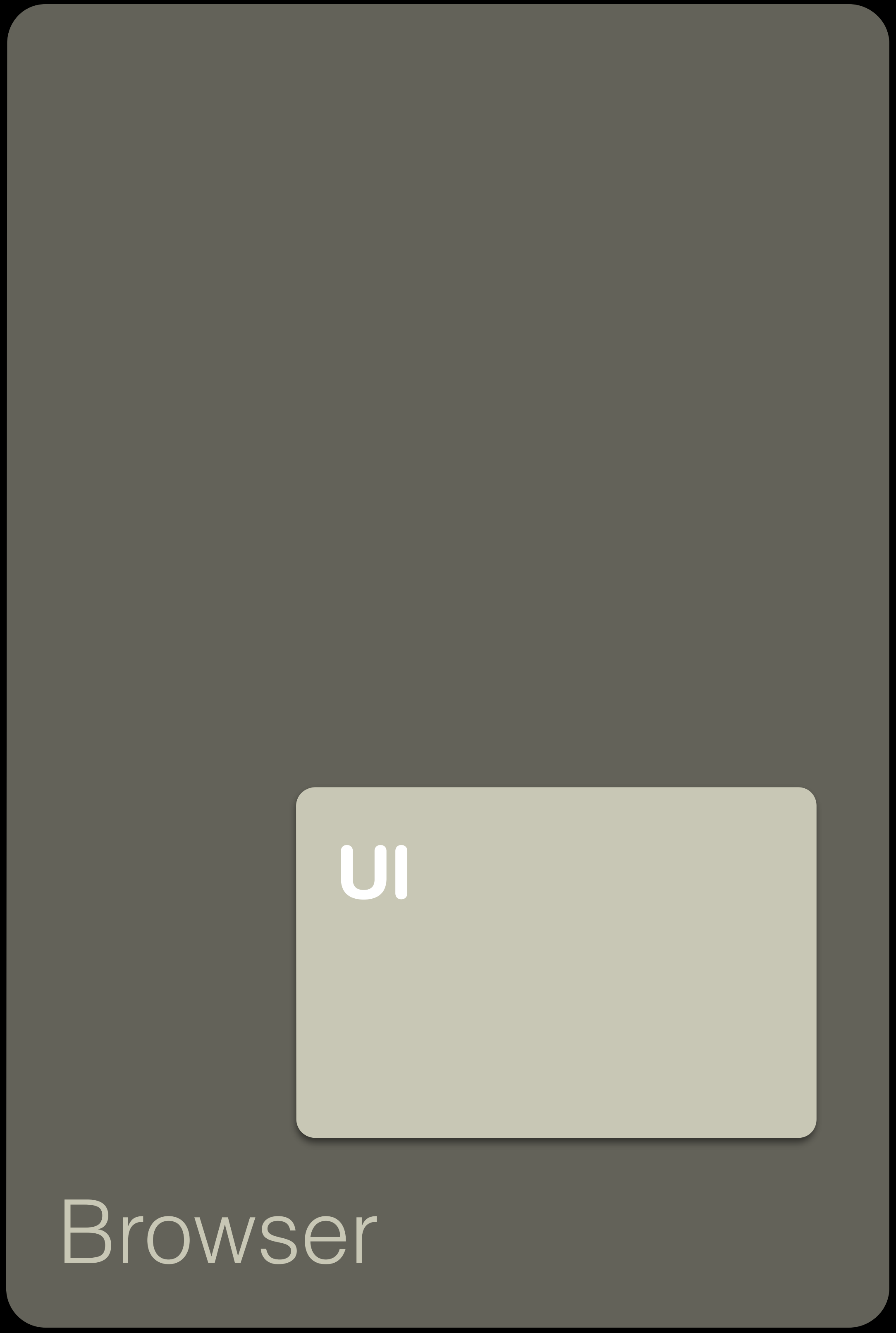
```
> {} + {};
> NaN

> NaN == NaN;
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> [object Object]

> {} + [];
> 0
```

```
> {} + {};
> NaN

> NaN == NaN;
> false
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> [object Object]

> {} + [];
> 0
```

```
> {} + {};
> NaN

> NaN == NaN;
> false

> typeof NaN;
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> [object Object]

> {} + [];
> 0
```

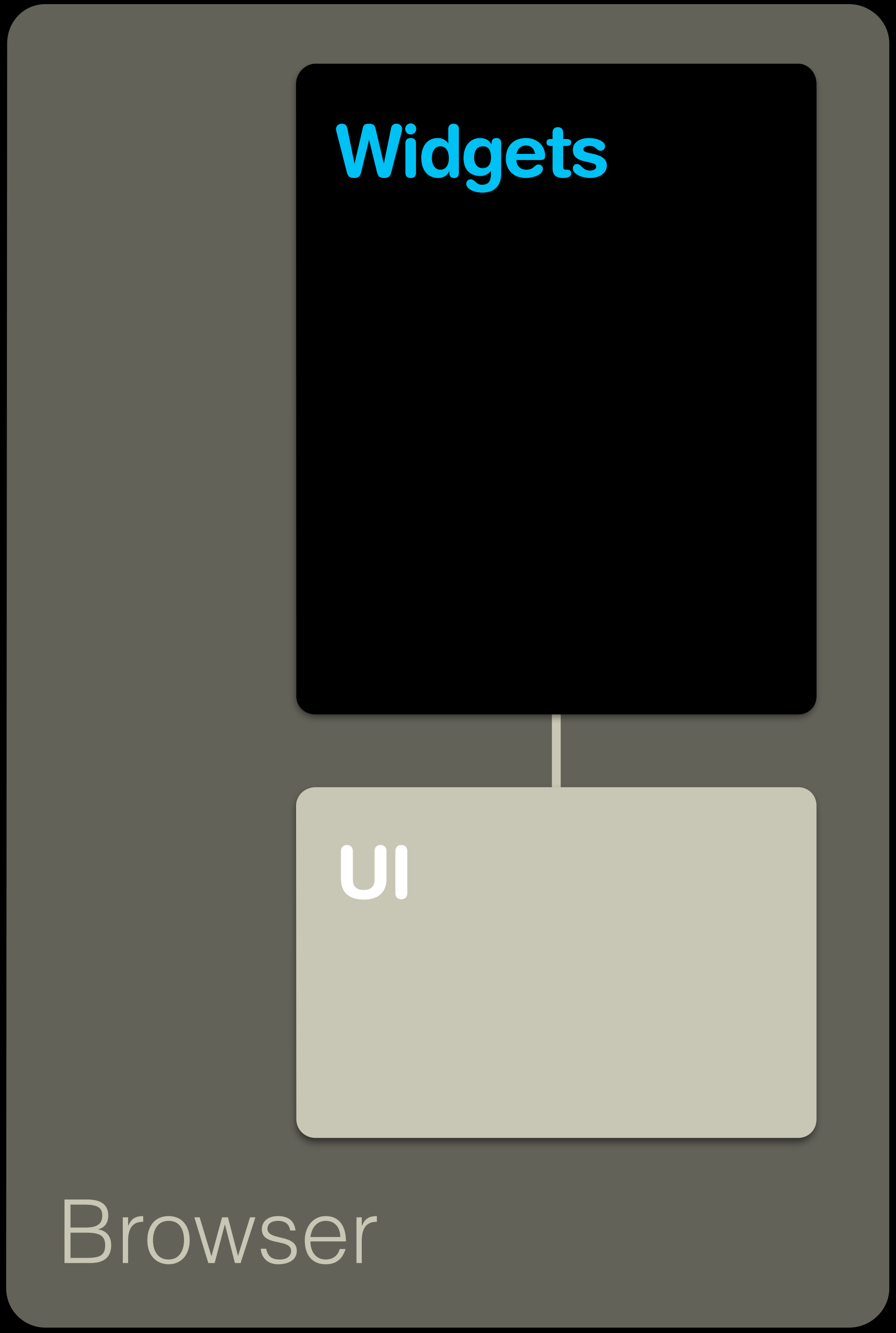
```
> {} + {};
> NaN

> NaN == NaN;
> false

> typeof NaN;
> number
> var foo = [0];
> foo == !foo;
> true

> [] + [];
>

> [] + {};
> [object Object]

> {} + [];
> 0
```

```
> {} + {};
> NaN

> NaN == NaN;
> false

> typeof NaN;
> number
```



UI

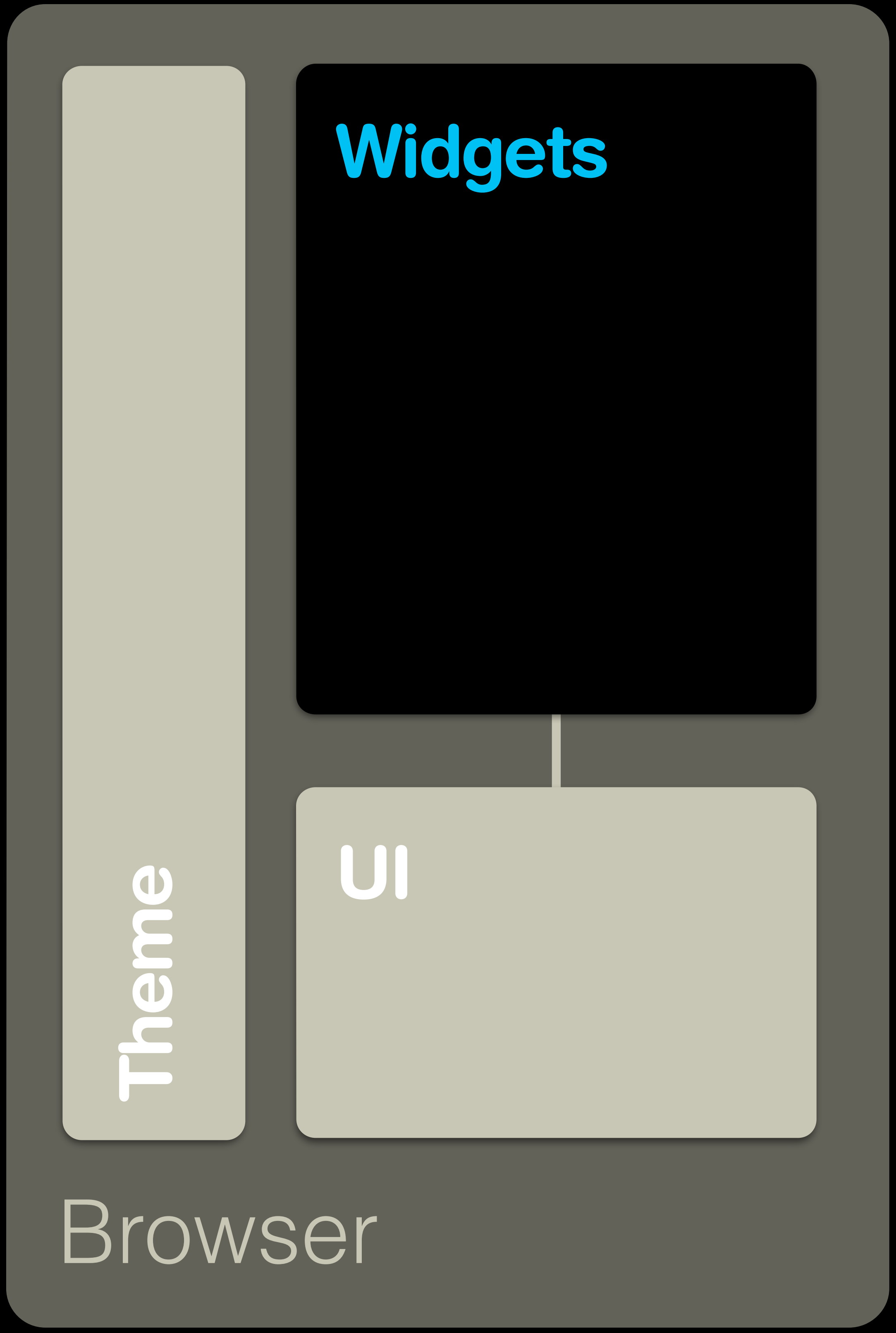
Browser

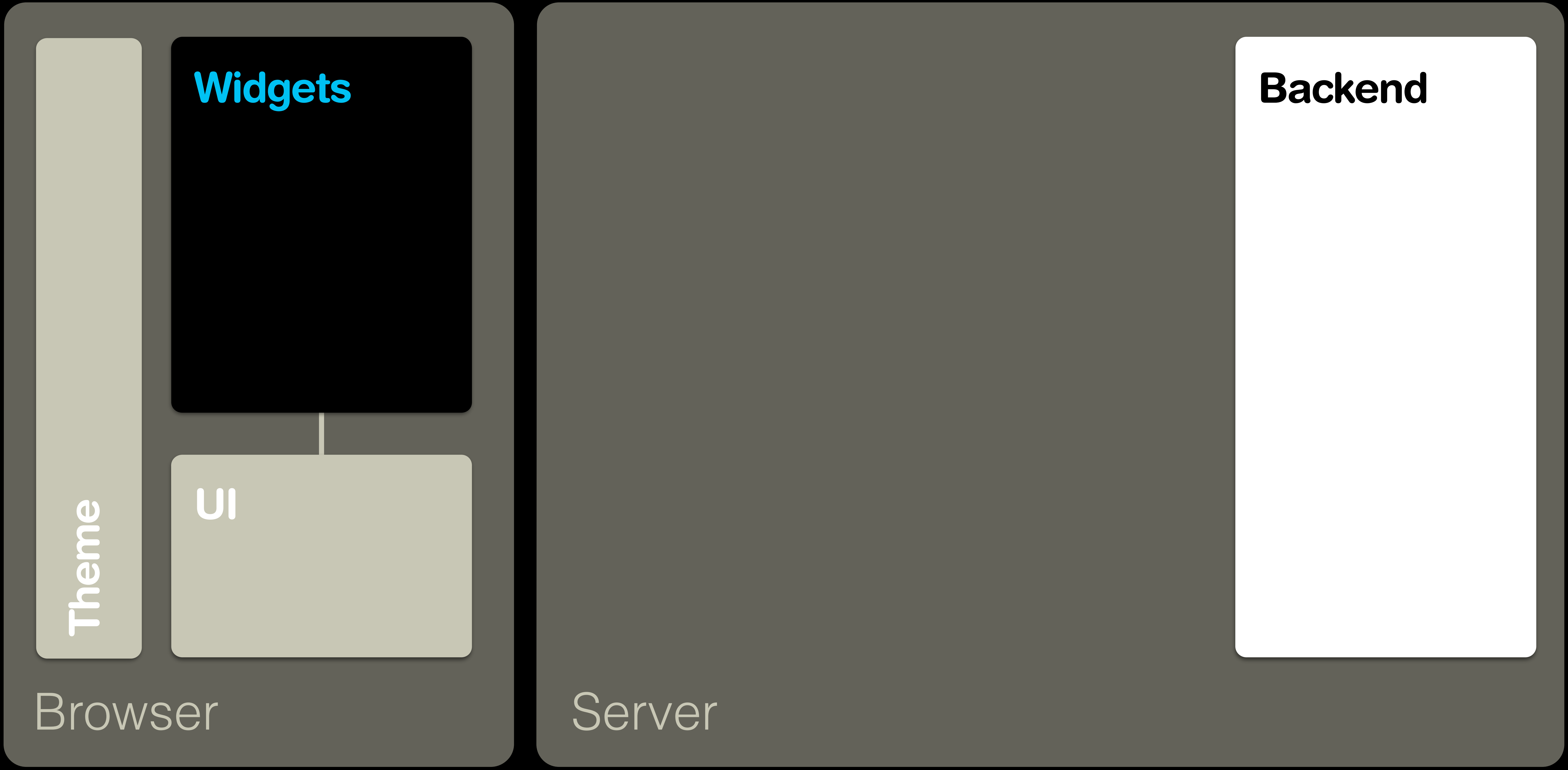


Widgets

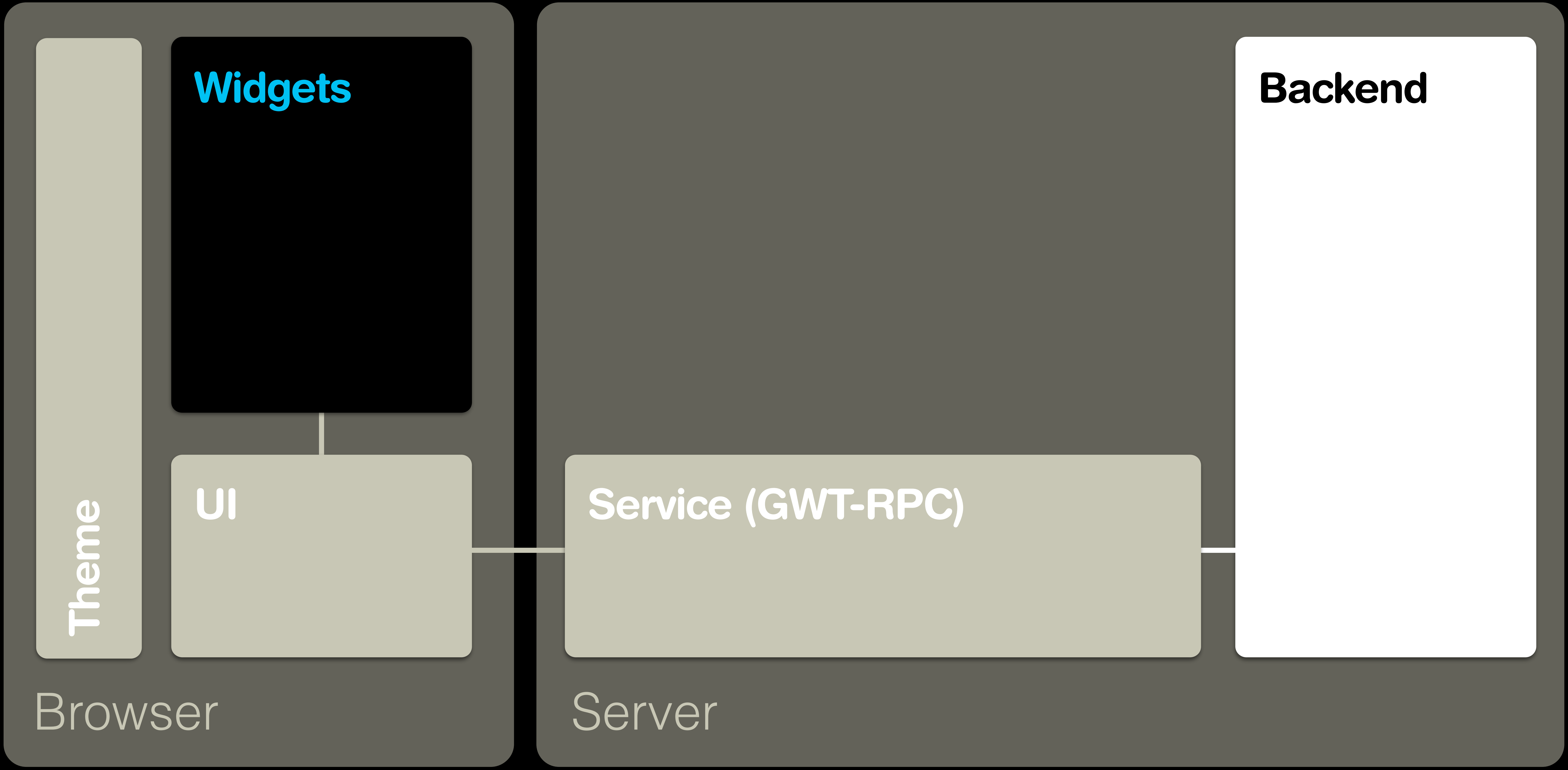
UI

Browser


Widgets
UI
Theme
Browser



vaadin }>



Widgets

Backend

Theme

UI

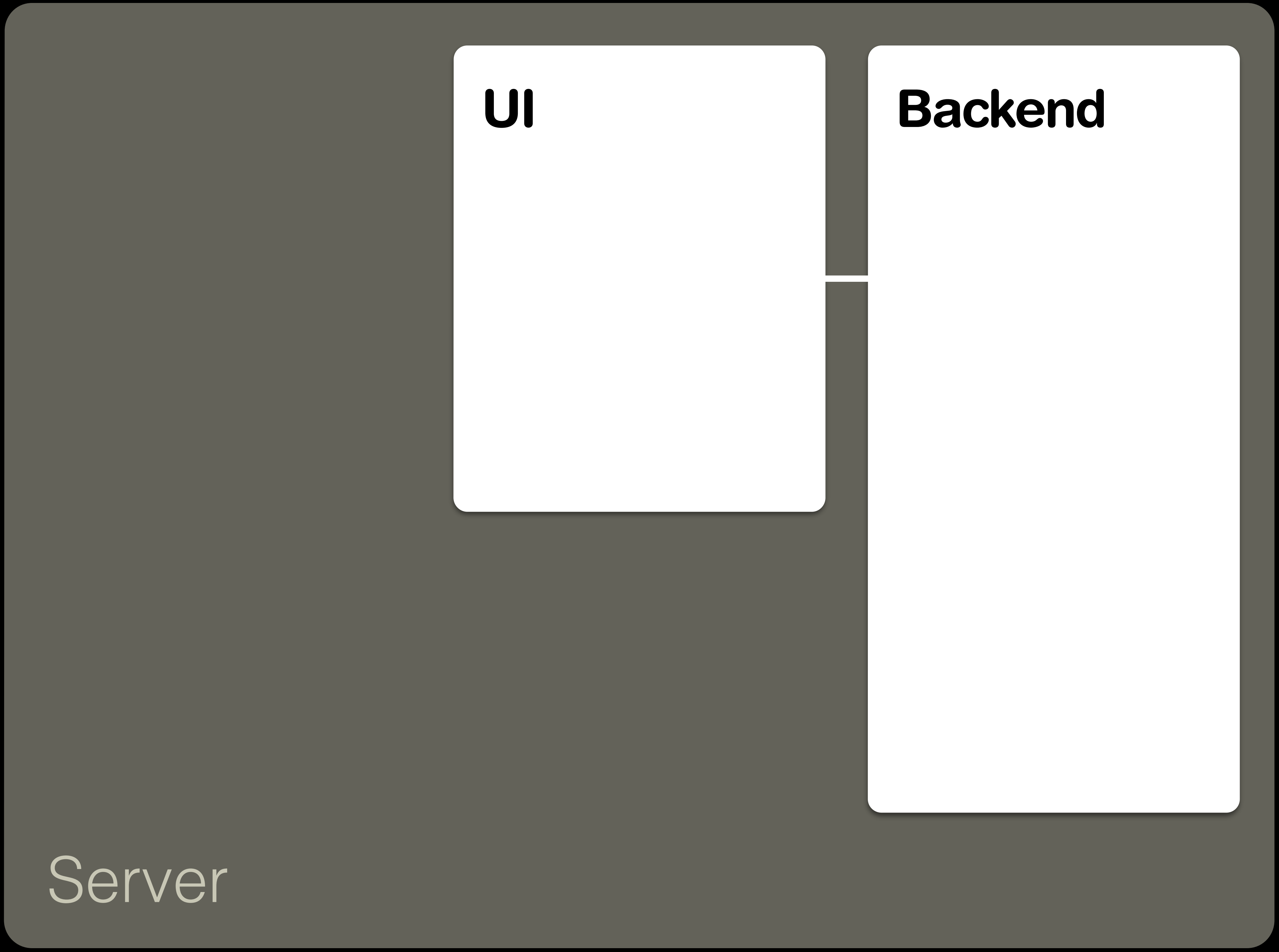
Service (GWT-RPC)

Browser

Server

Backend

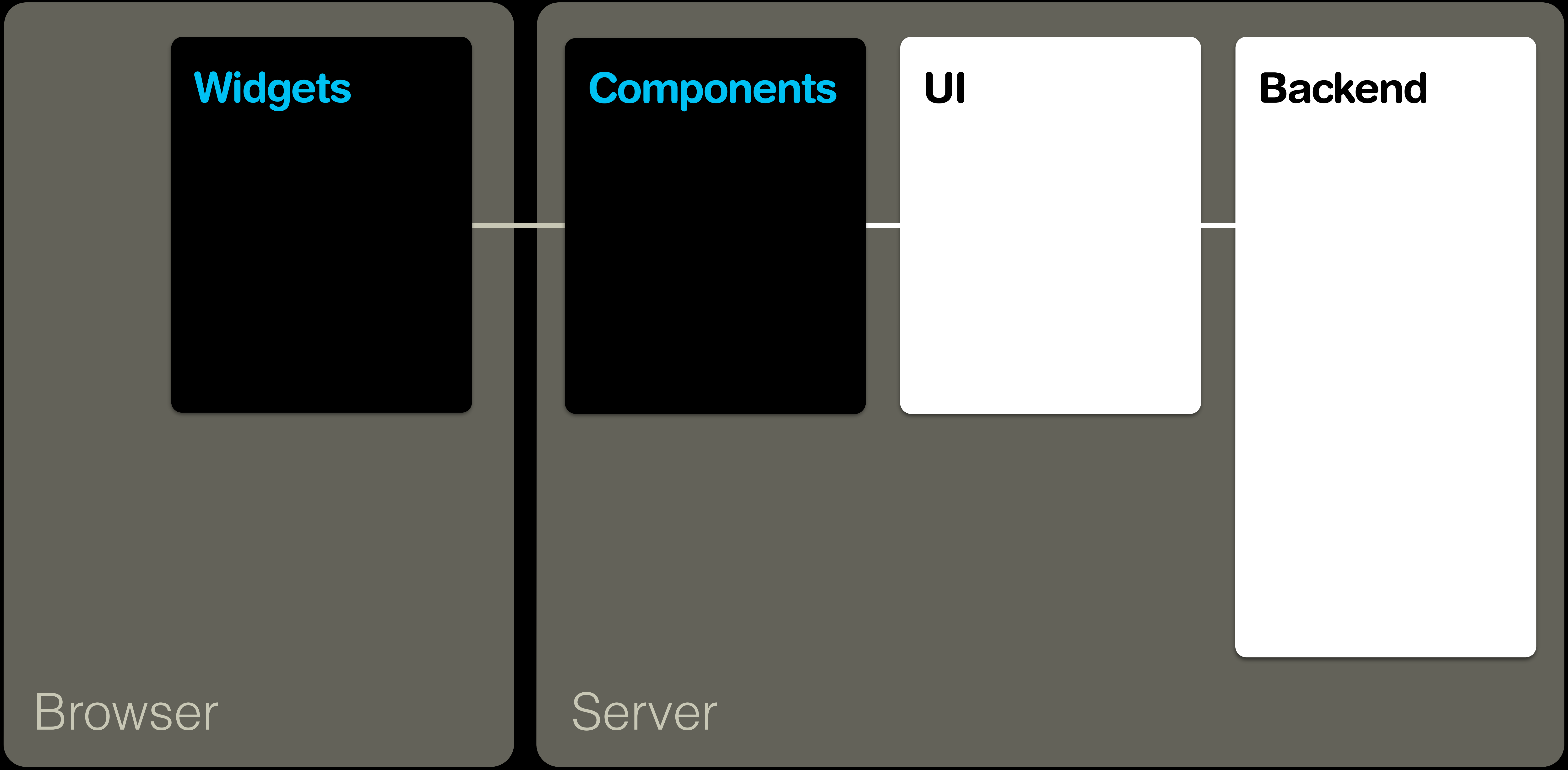
Server

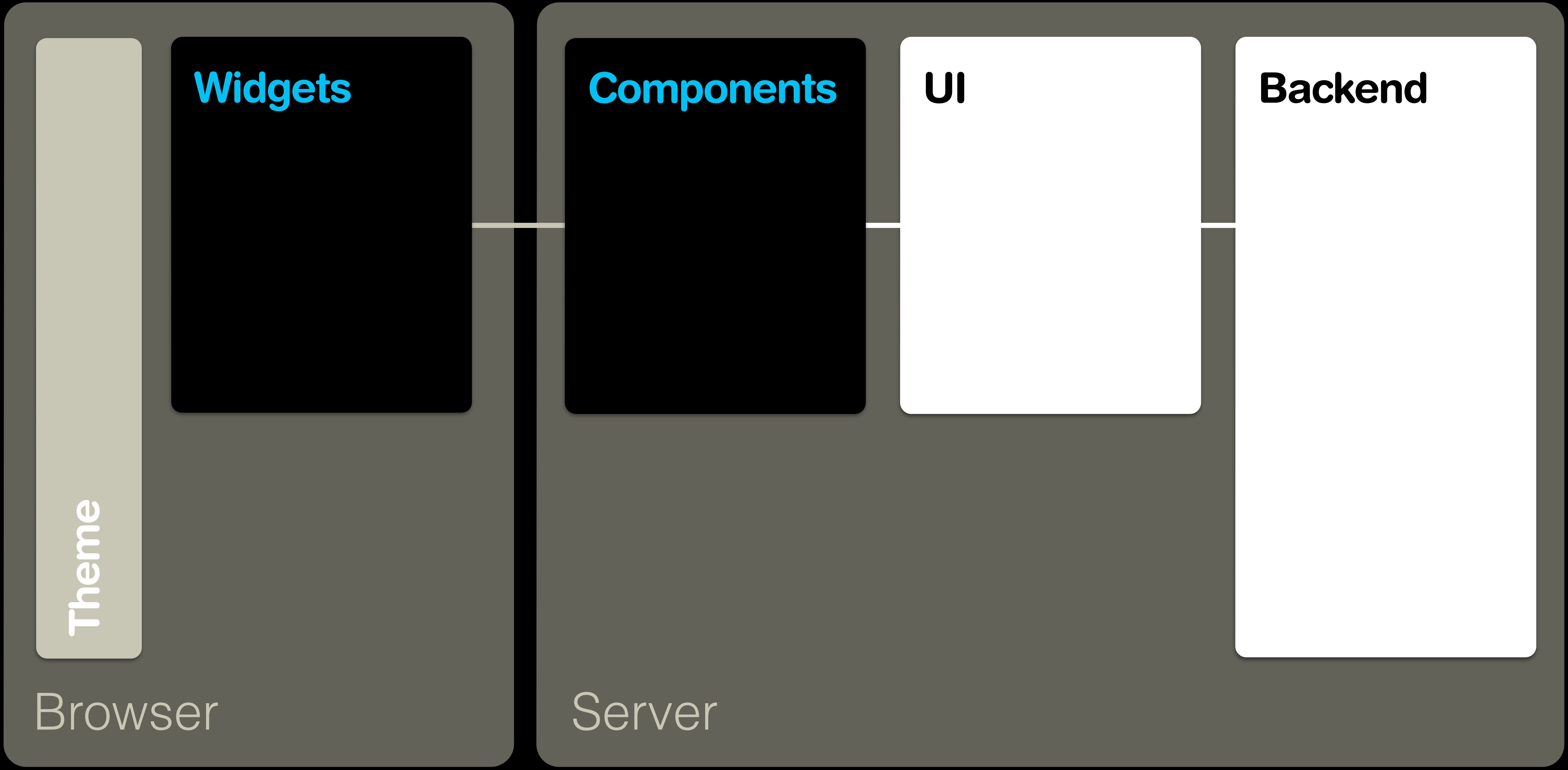


UI

Backend

Server


Widgets
Components
UI
Backend
Browser
Server



Theme

Widgets

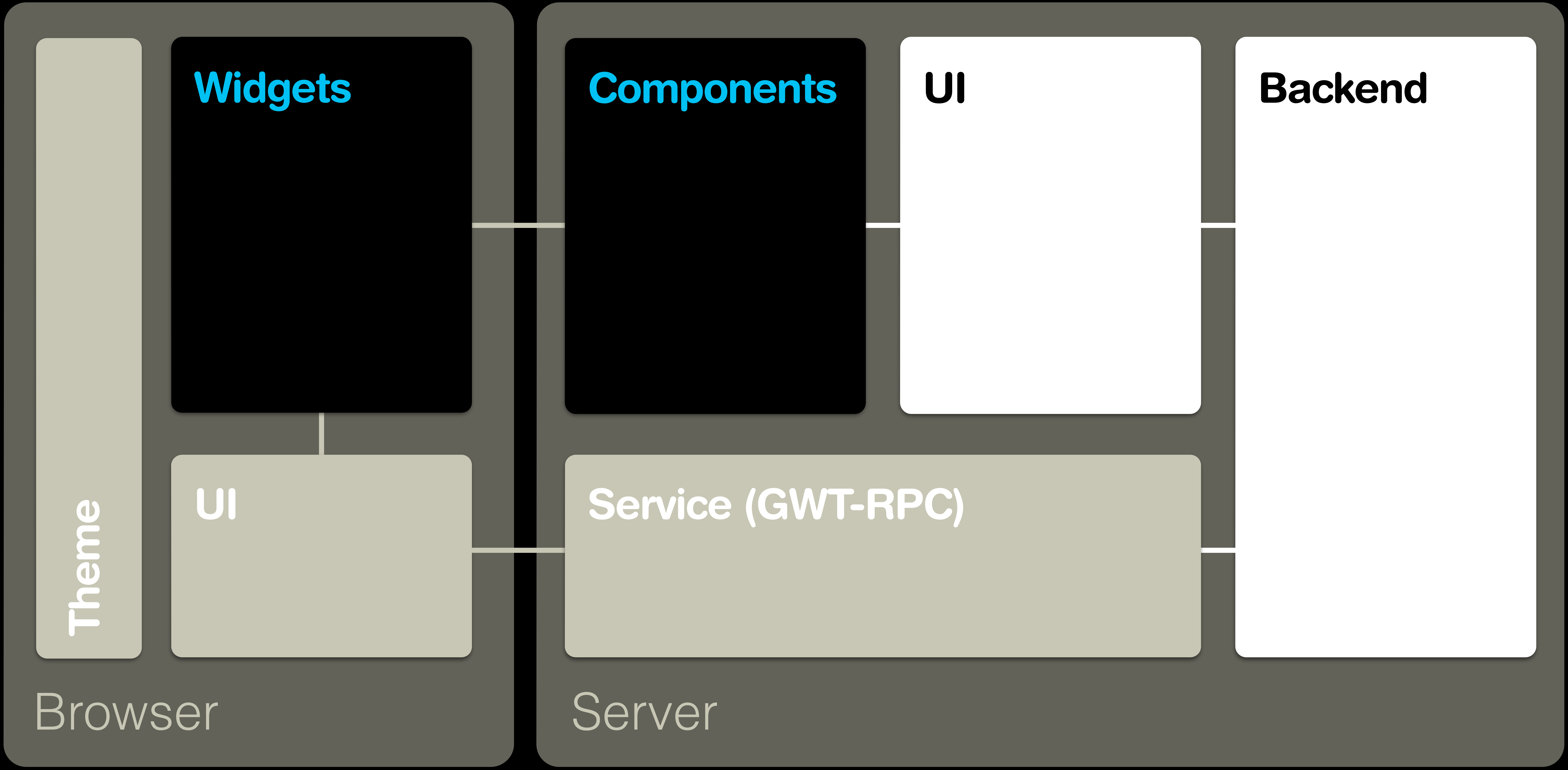
Browser

Components

UI

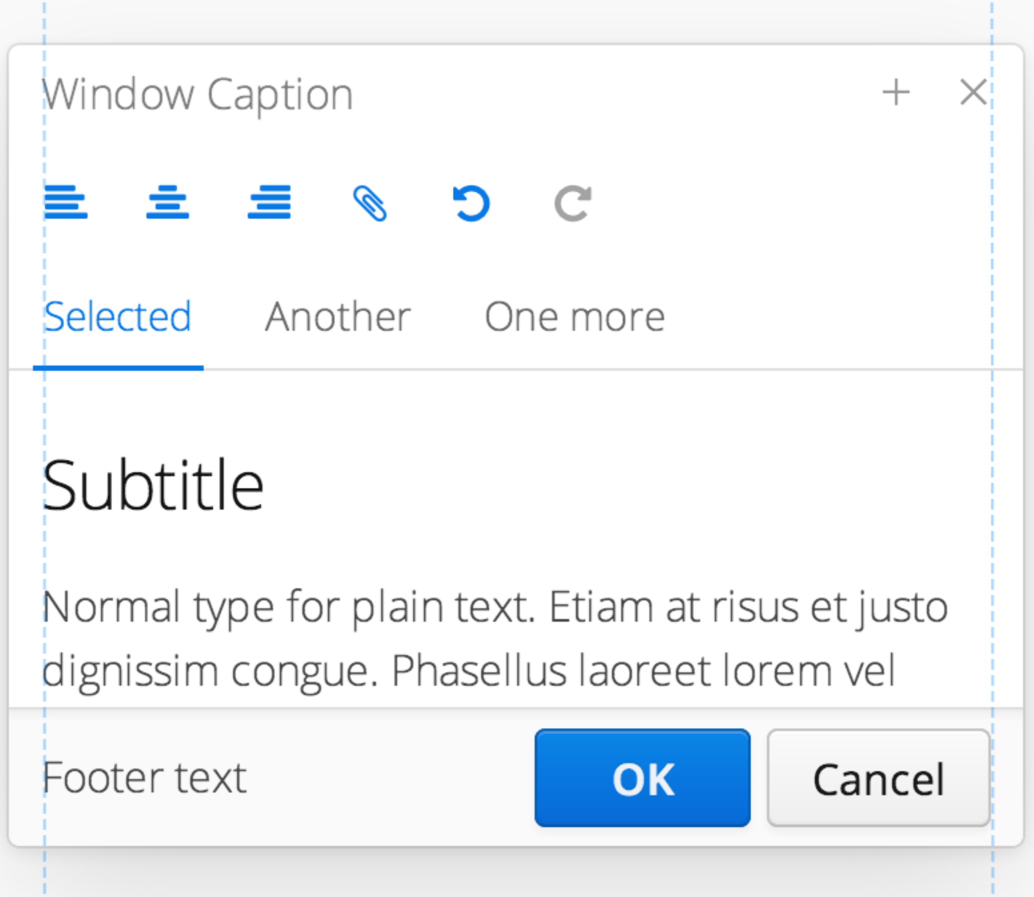
Backend

Server


Theme
Widgets
UI
Browser
Components
UI
Backend
Service (GWT-RPC)
Server

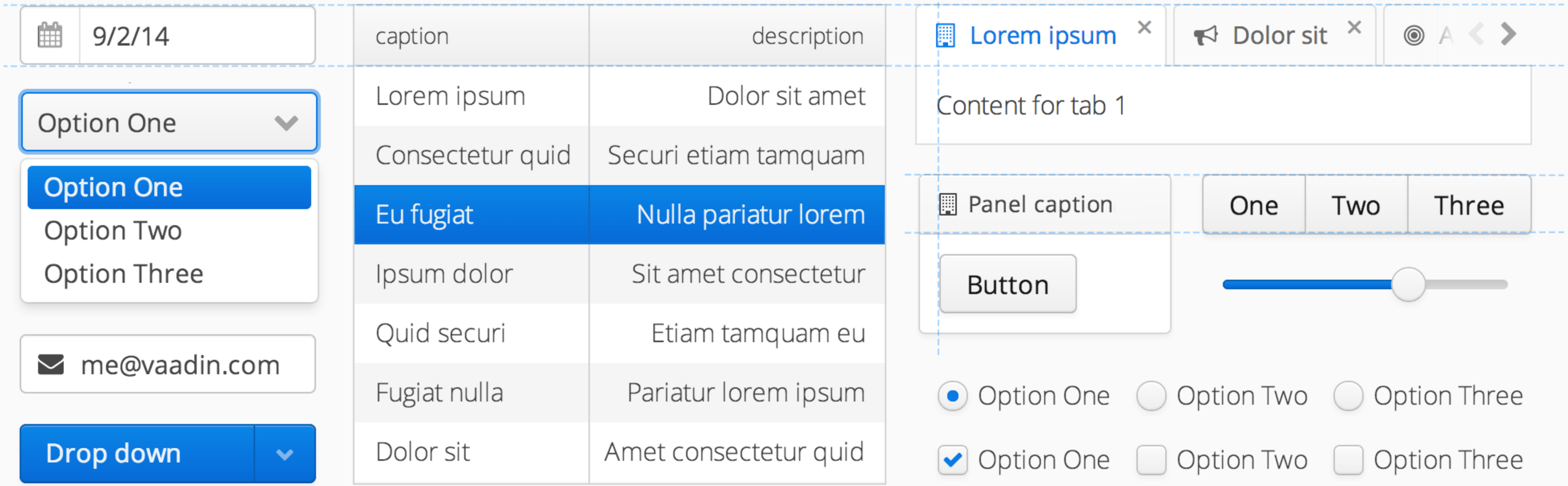
vaadin }>

User Interface **Components**



Subtitle

Normal type for plain text. Etiam at risus et justo dignissim congue. Phasellus laoreet lorem vel


9/2/14

me@vaadin.com

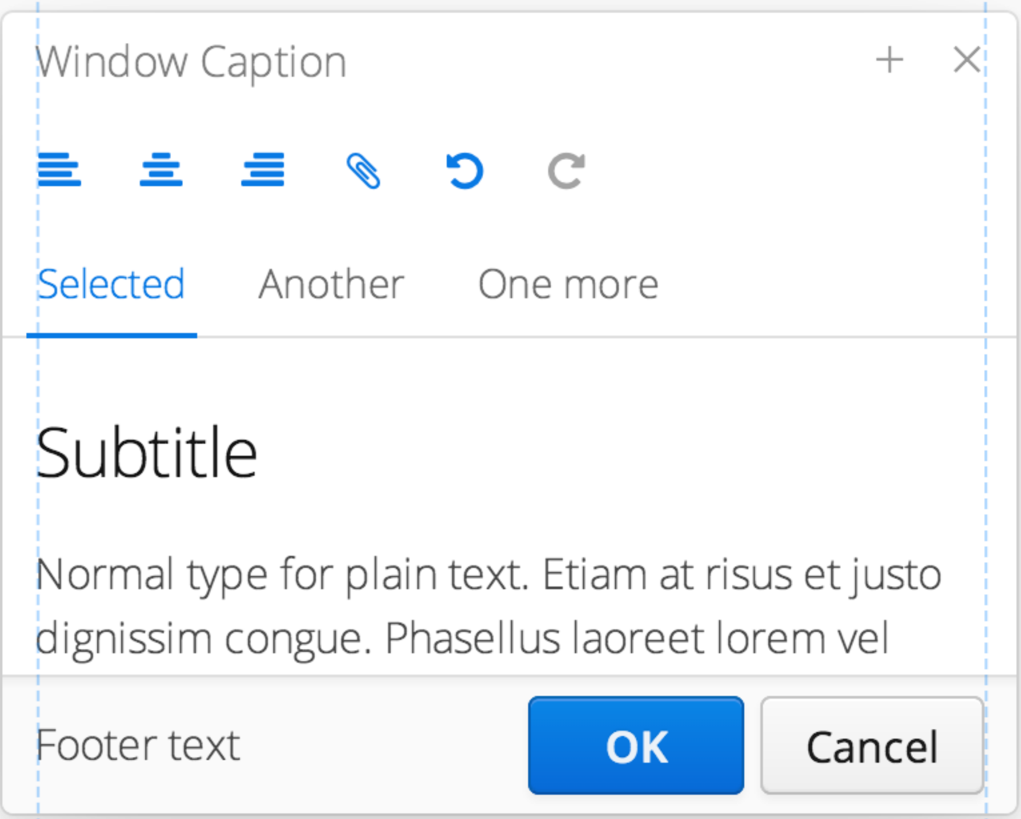
caption	description
Lorem ipsum	Dolor sit amet
Consectetur quid	Securi etiam tamquam
Eu fugiat	Nulla pariatur lorem
Ipsum dolor	Sit amet consectetur
Quid securi	Etiam tamquam eu
Fugiat nulla	Pariatur lorem ipsum
Dolor sit	Amet consectetur quid

Content for tab 1

● Option One ○ Option Two ○ Option Three

☑ Option One ☐ Option Two ☐ Option Three

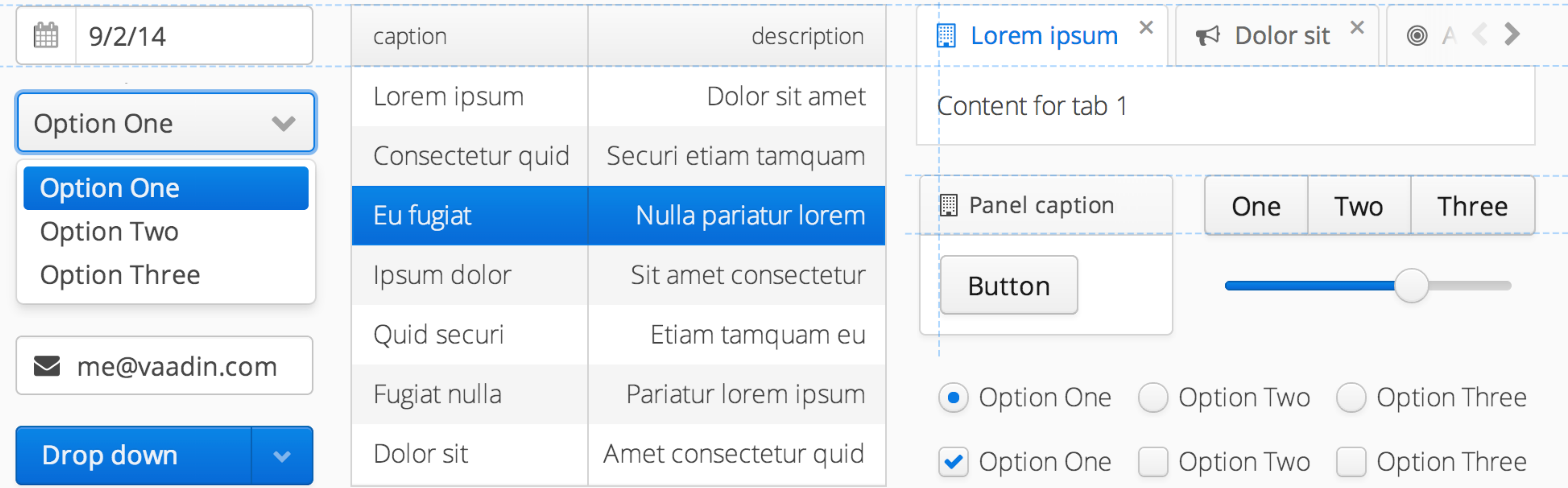
User Interface **Components**


Window Caption

Subtitle

Normal type for plain text. Etiam at risus et justo dignissim congue. Phasellus laoreet lorem vel

Footer text


9/2/14

Option One

- Option One
- Option Two
- Option Three

me@vaadin.com

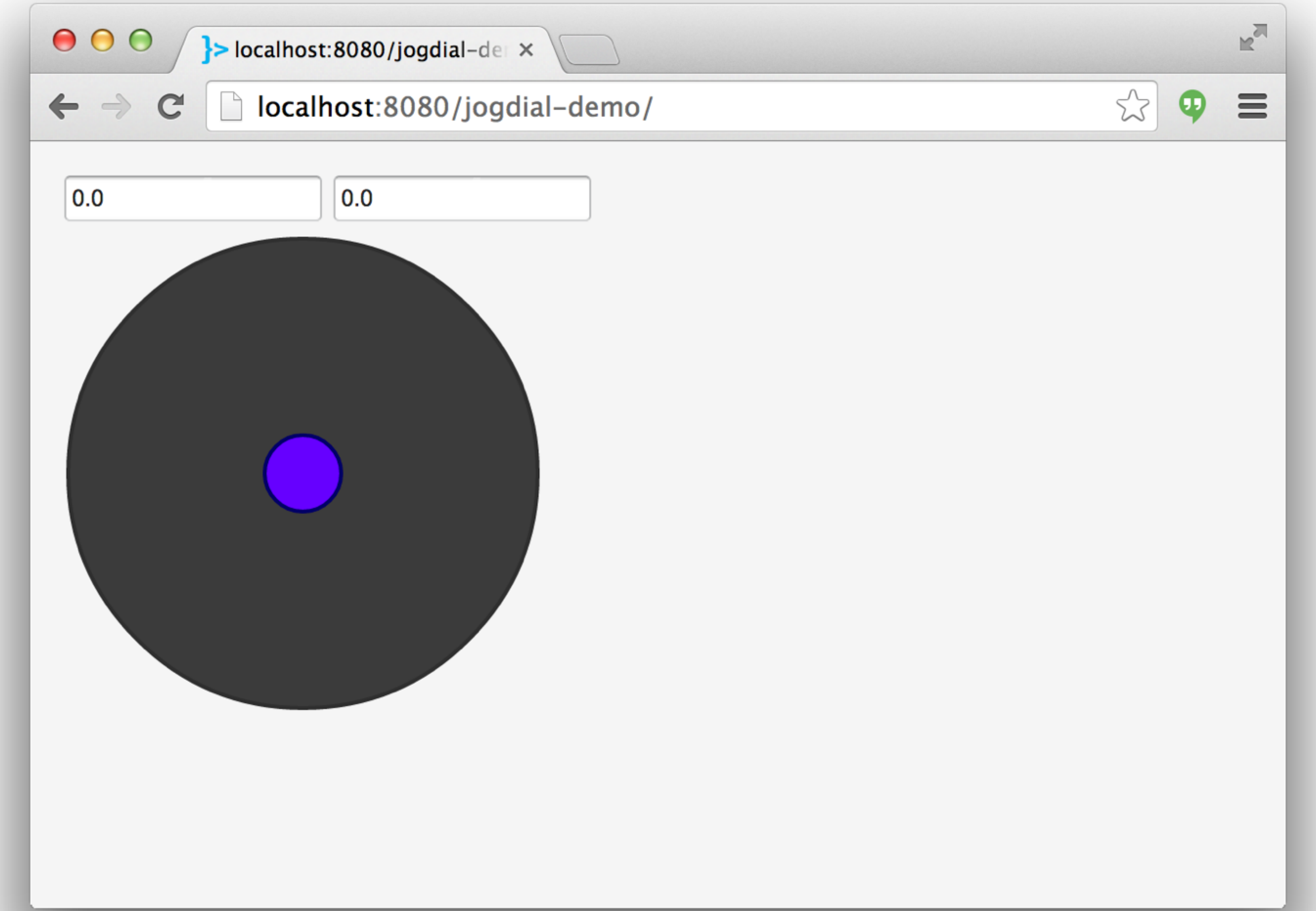
caption	description
Lorem ipsum	Dolor sit amet
Consectetur quid	Securi etiam tamquam
Eu fugiat	Nulla pariatur lorem
Ipsum dolor	Sit amet consectetur
Quid securi	Etiam tamquam eu
Fugiat nulla	Pariatur lorem ipsum
Dolor sit	Amet consectetur quid

Content for tab 1

● Option One ○ Option Two ○ Option Three

☑ Option One □ Option Two □ Option Three

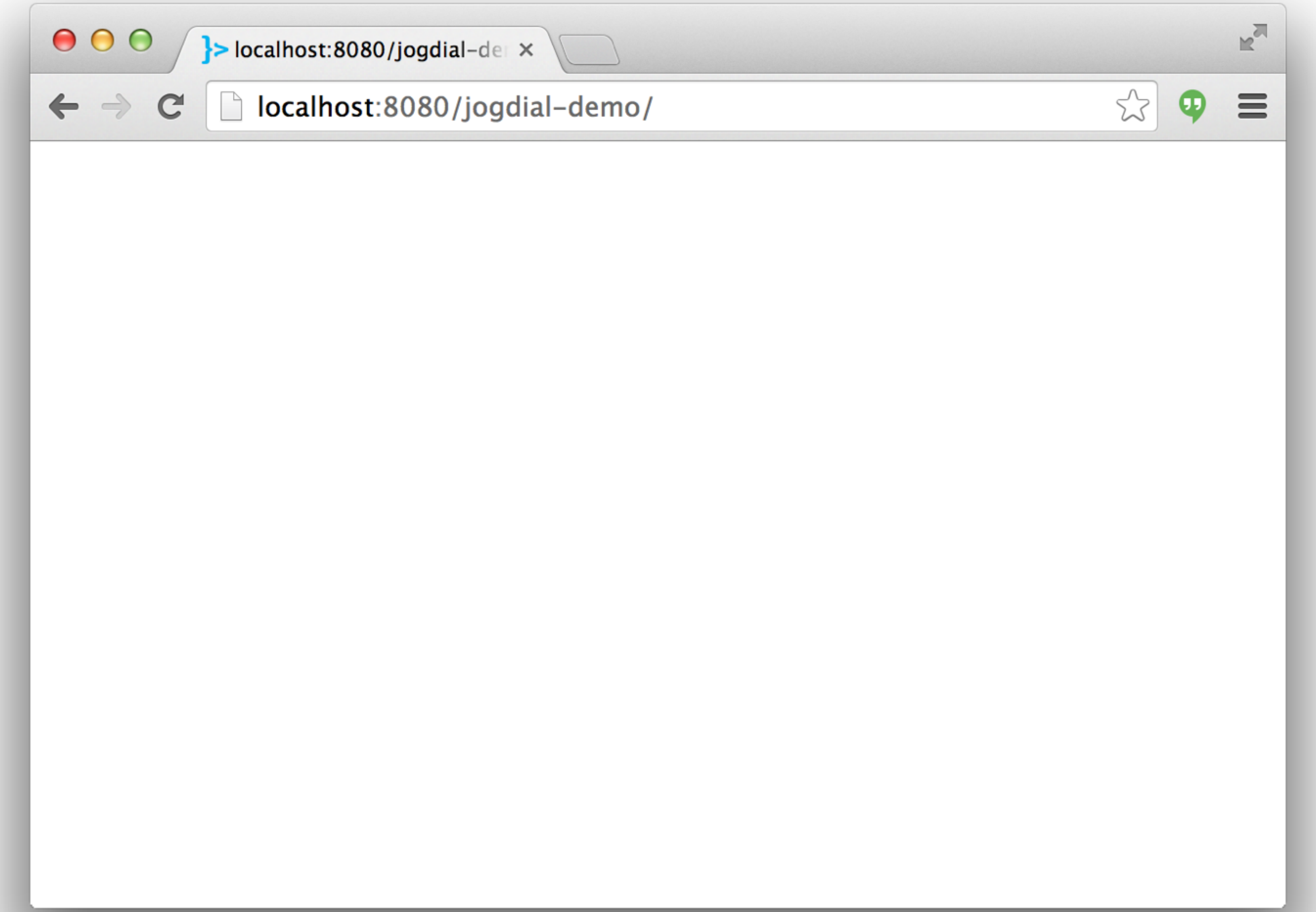
How does Vaadin work, really?


localhost:8080/jogdial-de
localhost:8080/jogdial-demo/
0.0
0.0

```java
JogDial jogDial = new JogDial(Size.MEDIUM);

TextField x = new TextField();
TextField y = new TextField();
```

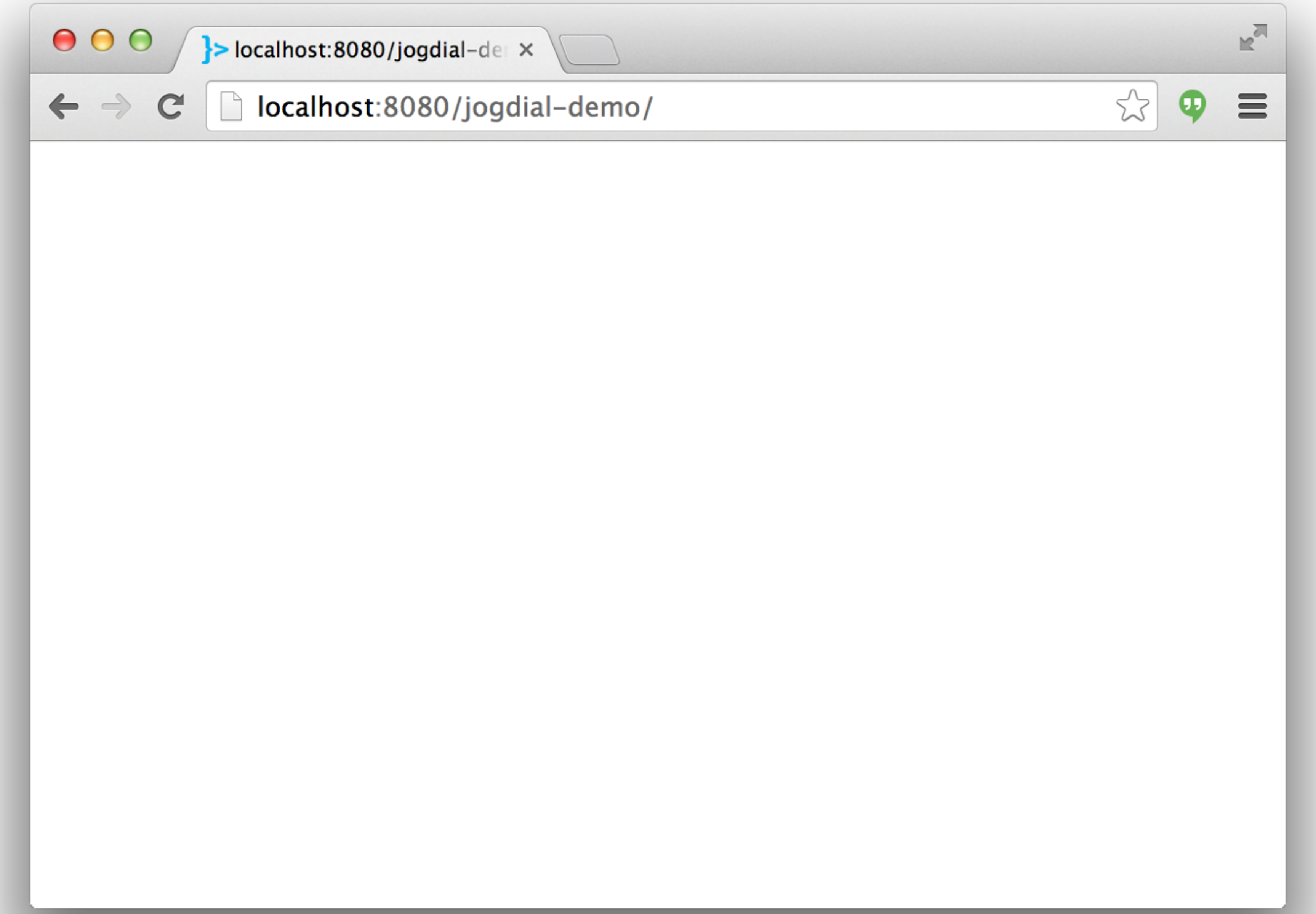
```
JogDial jogDial = new JogDial(Size.MEDIUM);

TextField x = new TextField();
TextField y = new TextField();

jogDial.addMoveListener(e -> {
   x.setValue(e.getX());
   y.setValue(e.getY());
});
```

```java
JogDial jogDial = new JogDial(Size.MEDIUM);

TextField x = new TextField();
TextField y = new TextField();

jogDial.addMoveListener(e -> {
   x.setValue(e.getX());
   y.setValue(e.getY());
});

layout.addComponents(x, y, jogDial);
```



localhost:8080/jogdial-demo/

- Loader page
- CSS Theme
- Images
- JavaScript

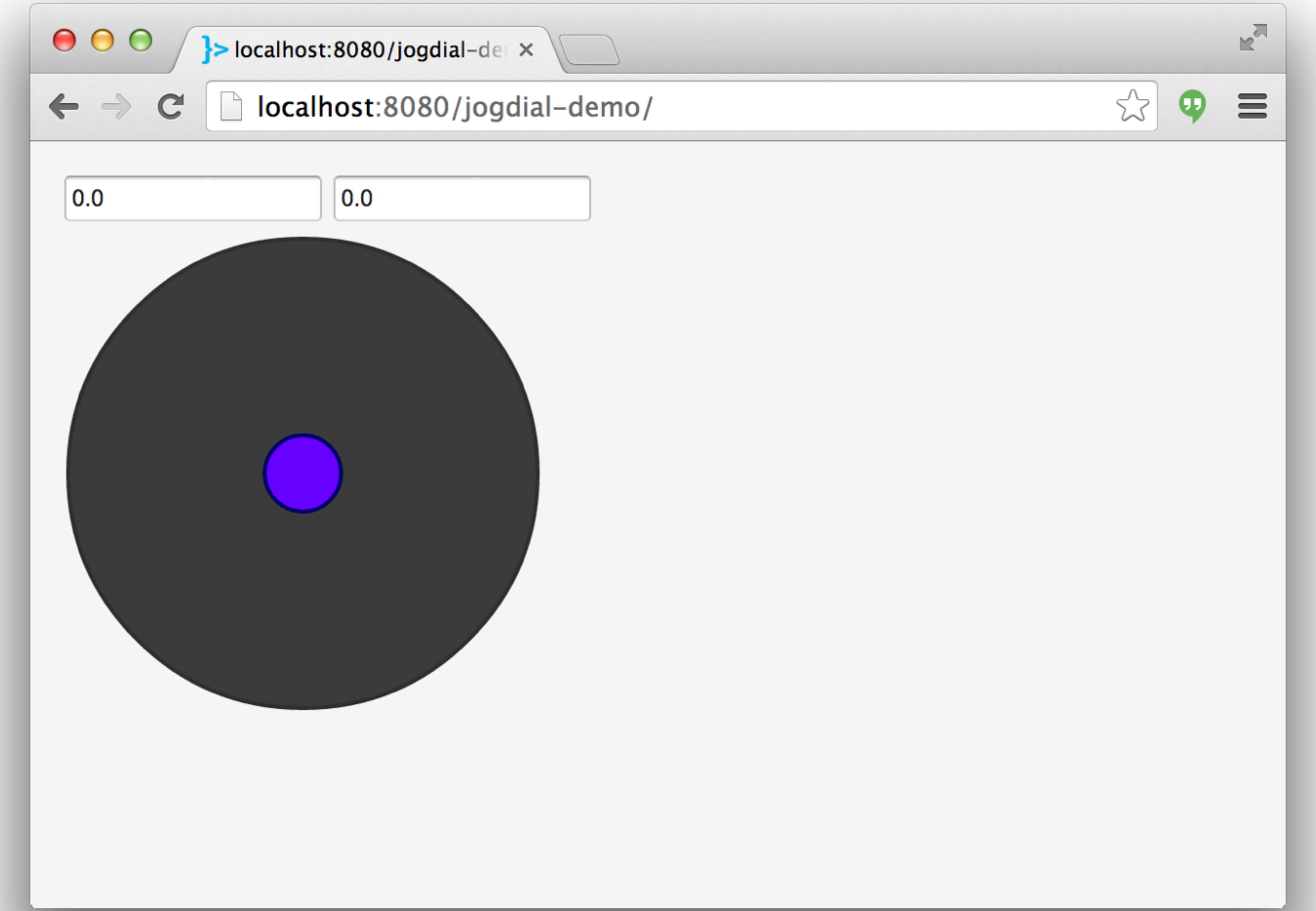


localhost:8080/jogdial-demo/

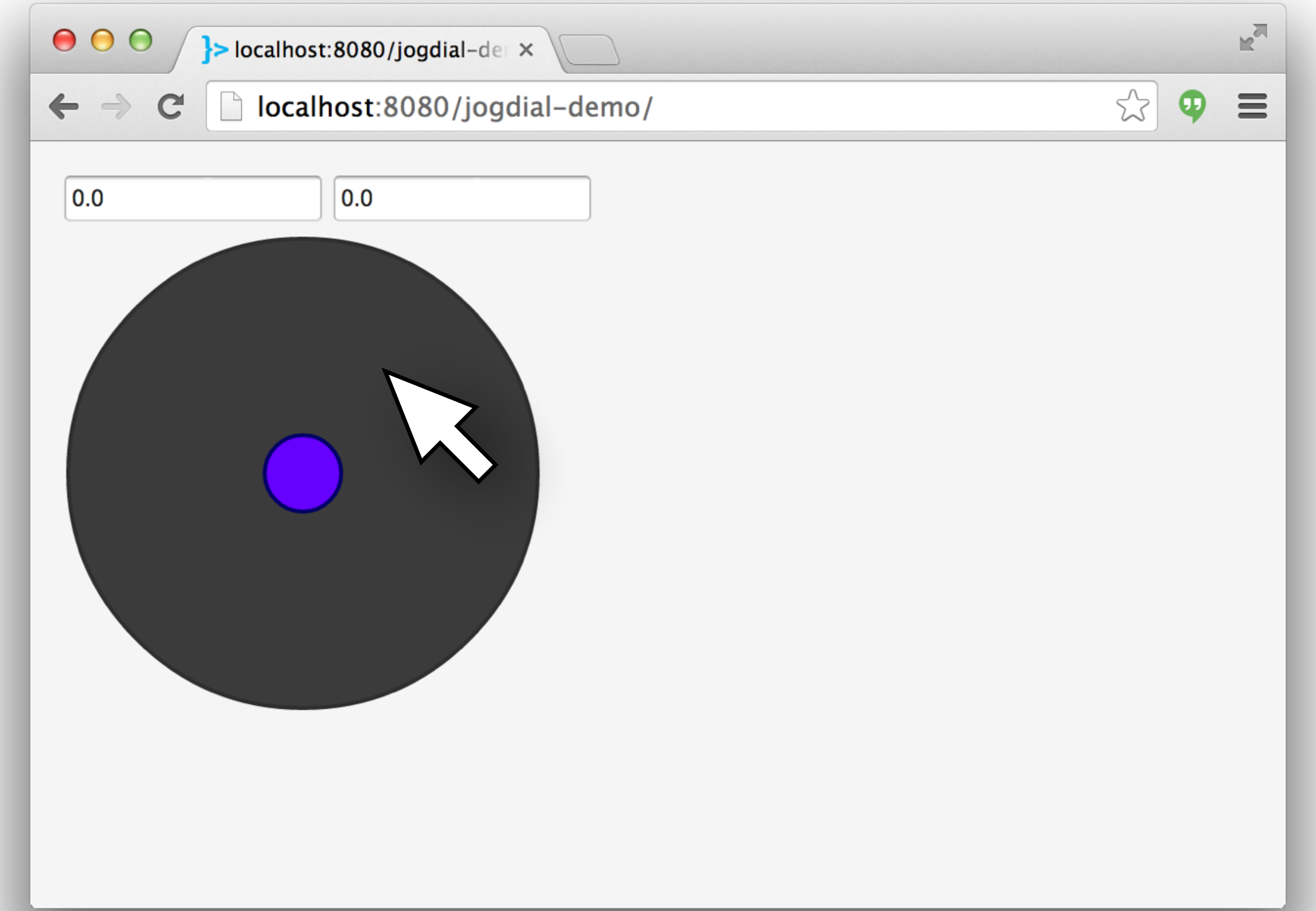
- Loader page
- CSS Theme
- Images
- JavaScript

Compressed & reduced Thin client

135k

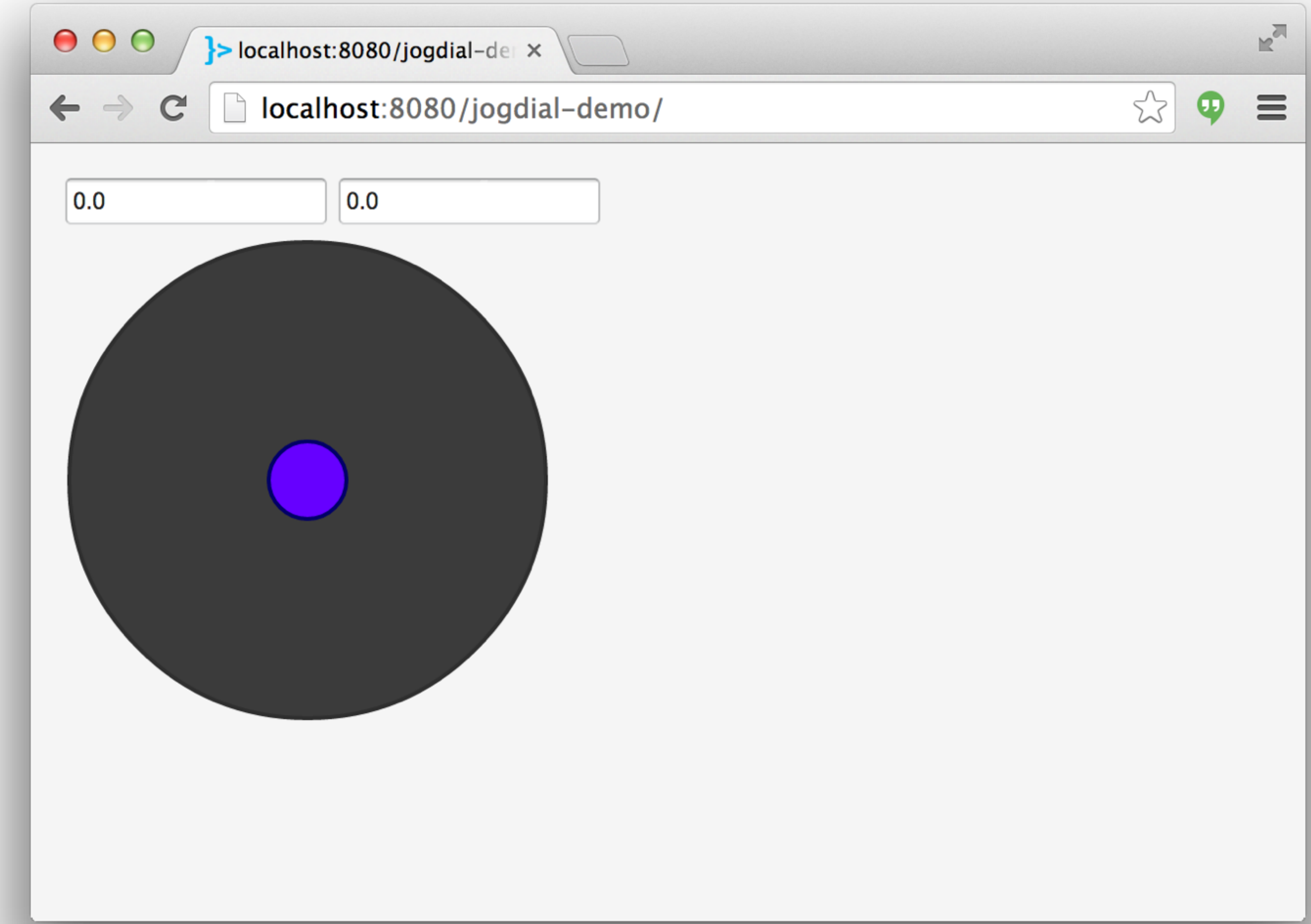


0.0 0.0



0.0

0.0

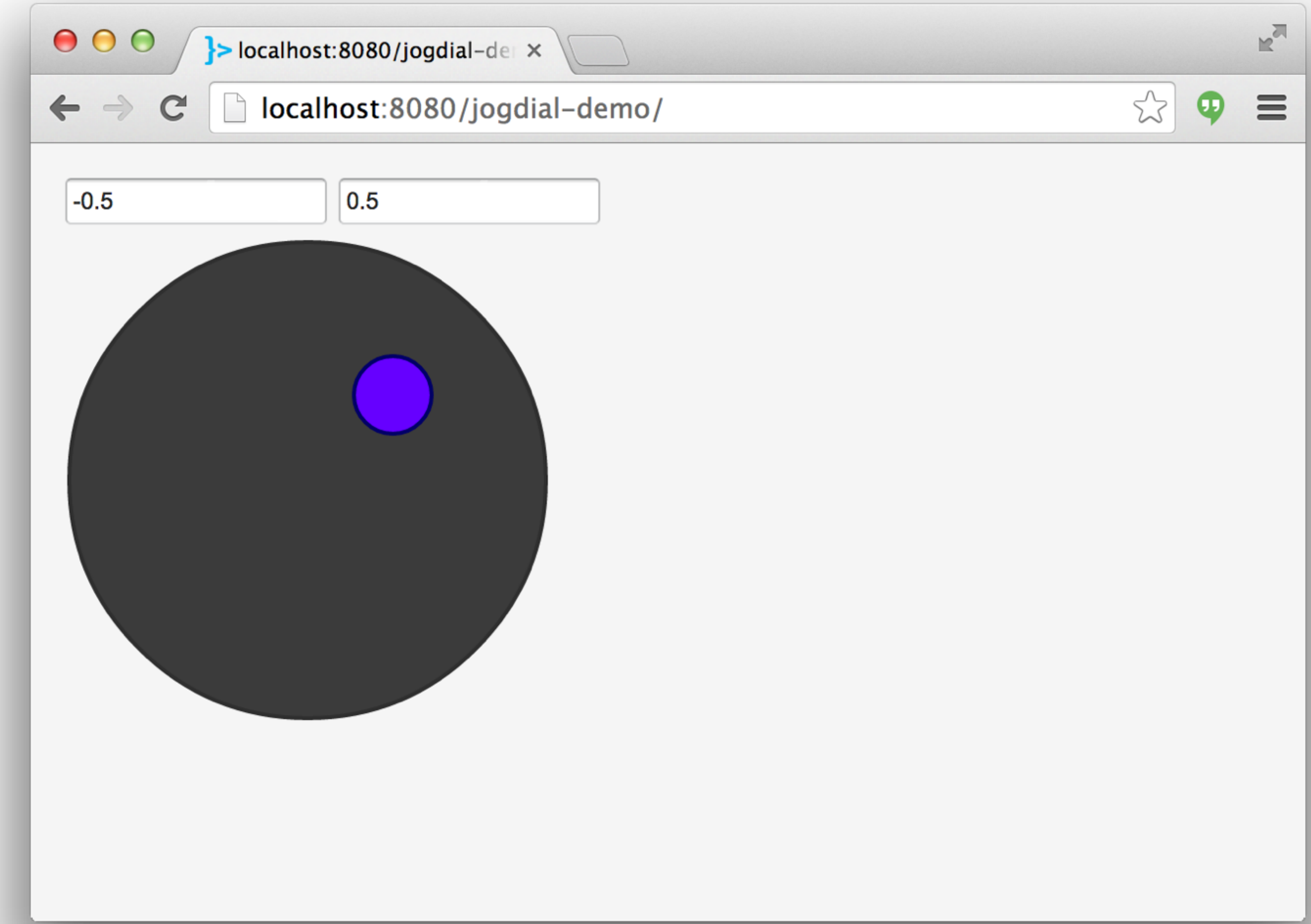


0.0 0.0

- jogDialMoved(-0.5,0.5)

314 bytes

```
JogDial jogDial = new JogDial(Size.MEDIUM);

TextField x = new TextField();
TextField y = new TextField();

jogDial.addMoveListener(e -> {
   x.setValue(e.getX());
   y.setValue(e.getY());
});

layout.addComponents(x, y, jogDial);
```



- jogDialMoved(-0.5,0.5)

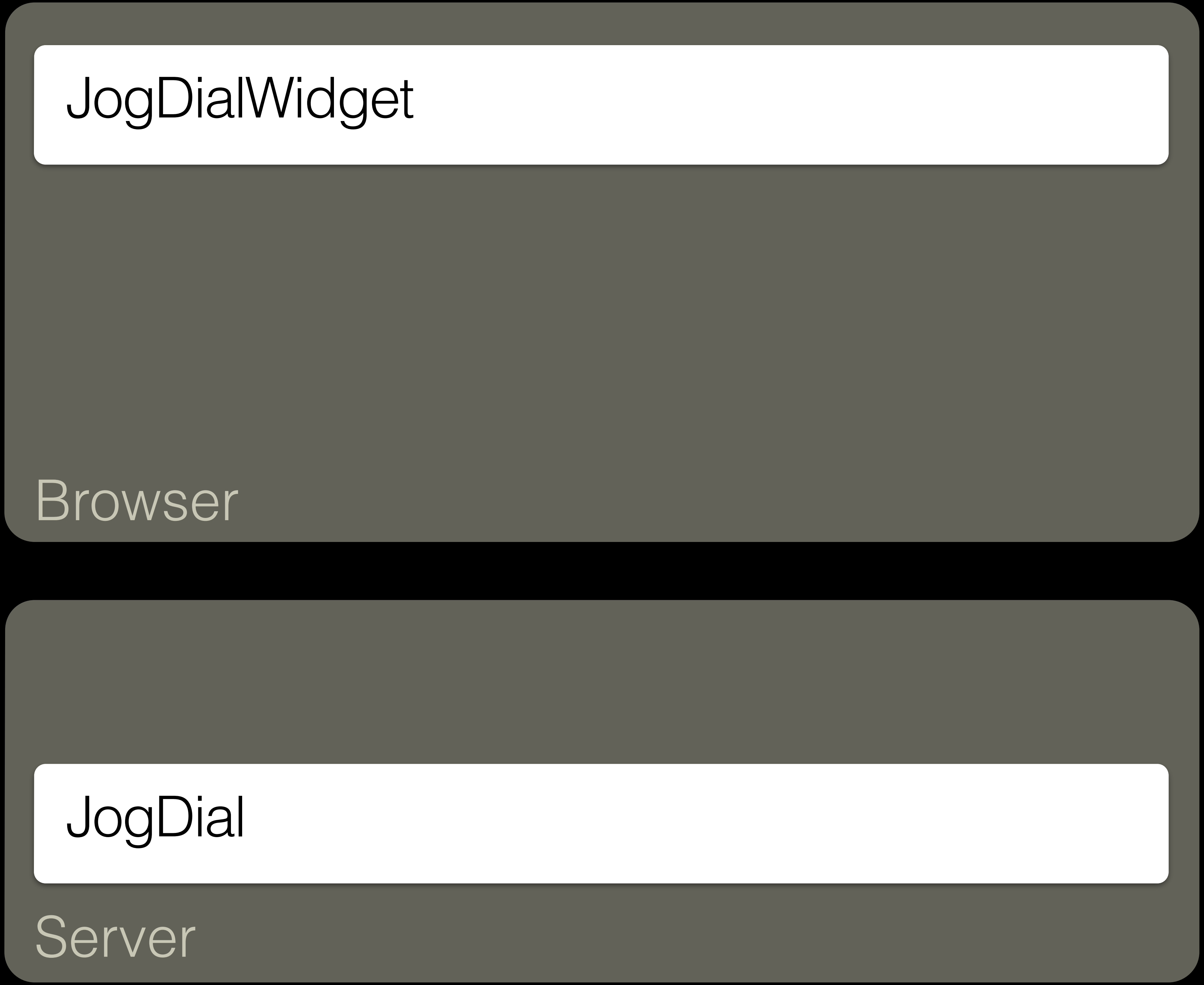
314

- moveCap(-0.5,0.5)
- x.value=-0.5
  y.value=0.5

351 bytes

JogDial

Server

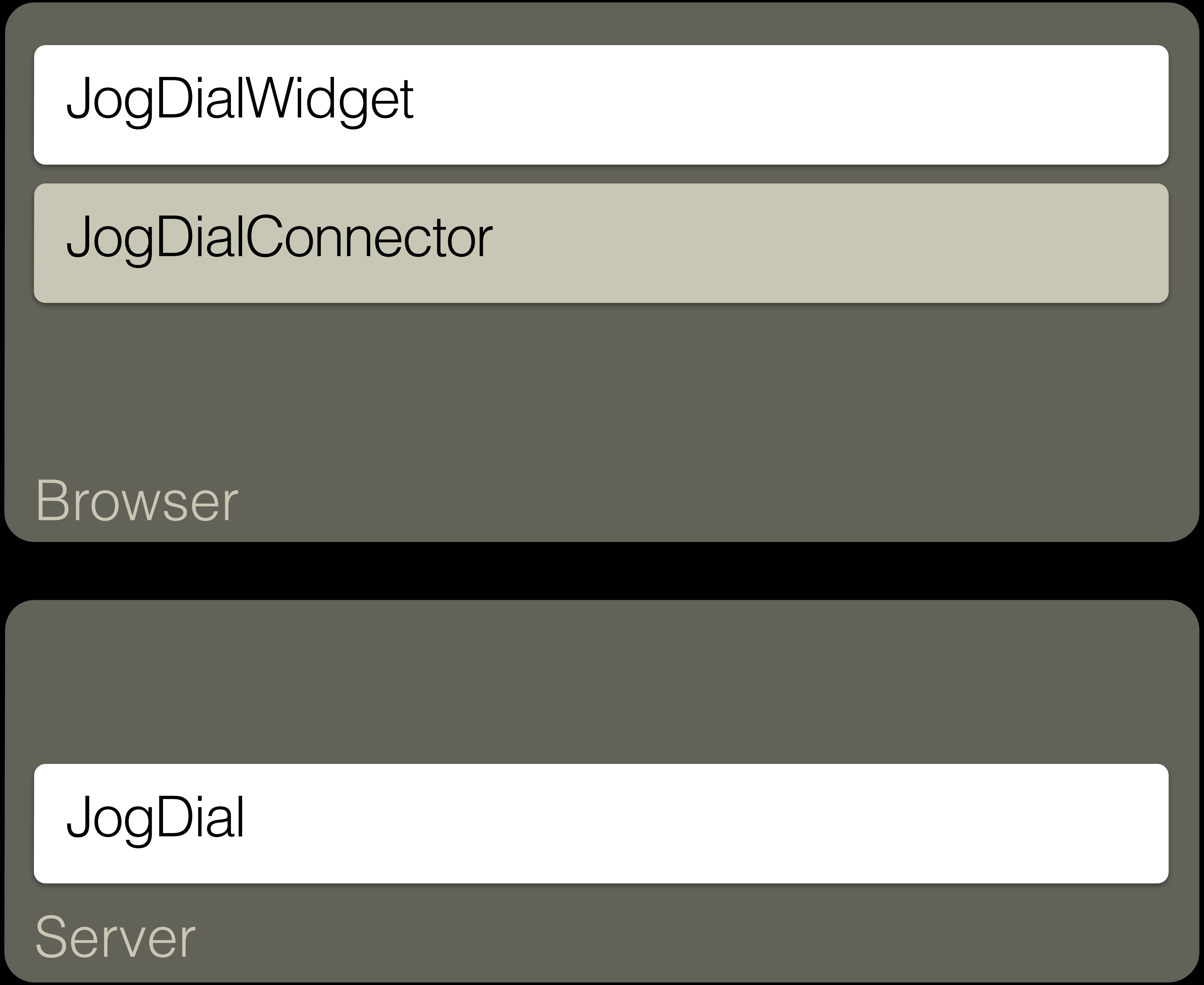


JogDialWidget

Browser

JogDial

Server



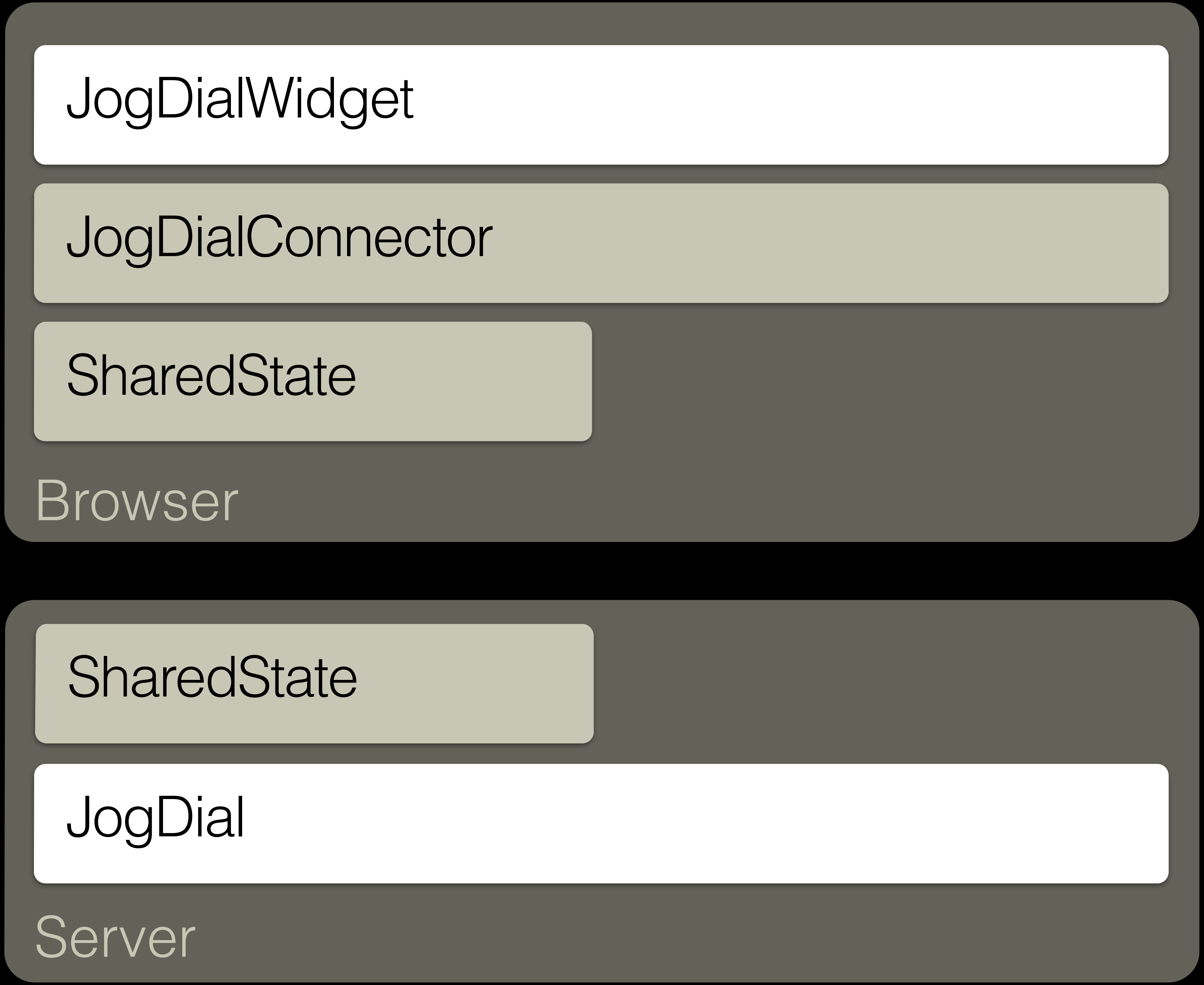
JogDialWidget

JogDialConnector

Browser

JogDial

Server



JogDialWidget

JogDialConnector

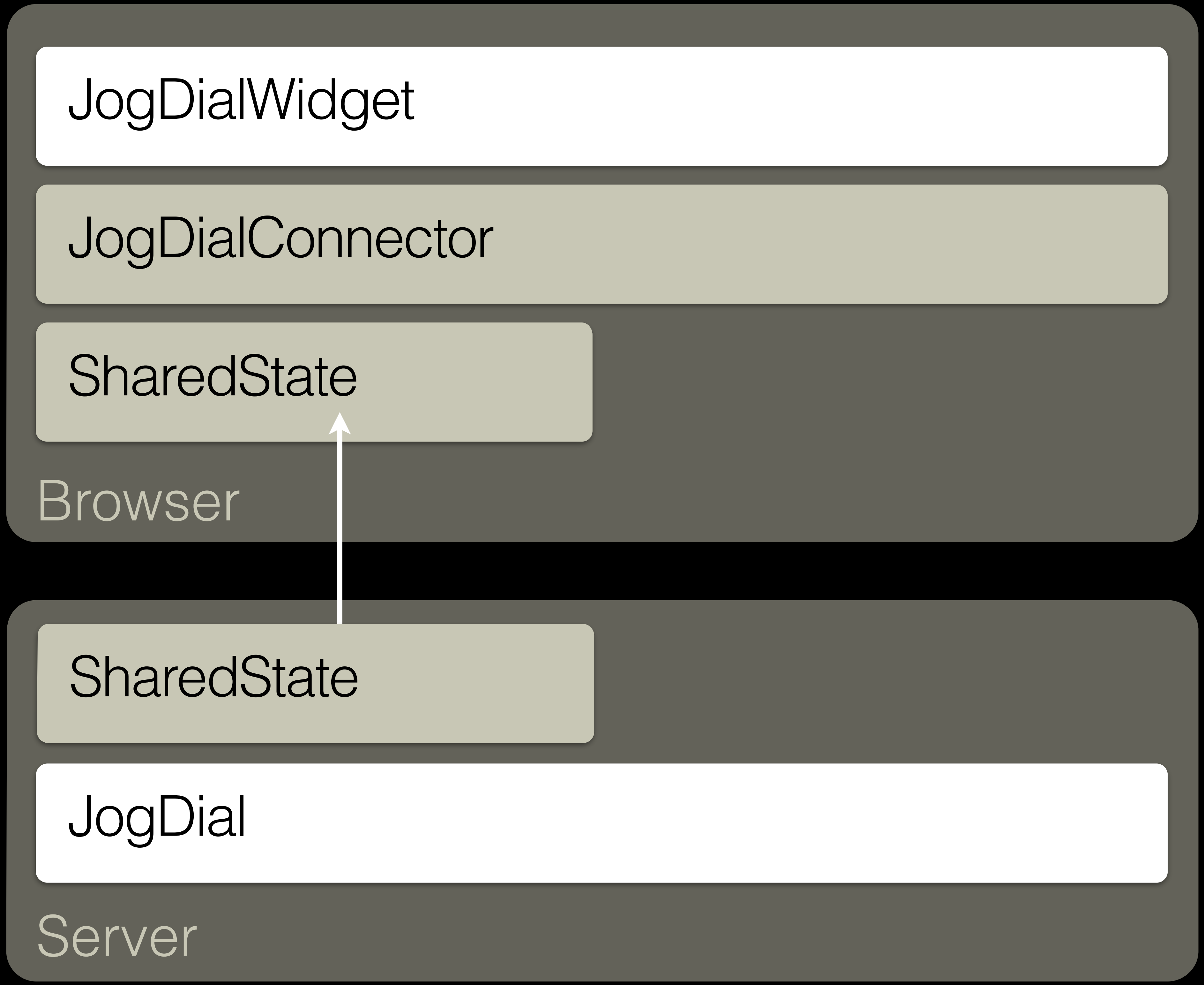
SharedState

Browser

SharedState

JogDial

Server



JogDialWidget

JogDialConnector

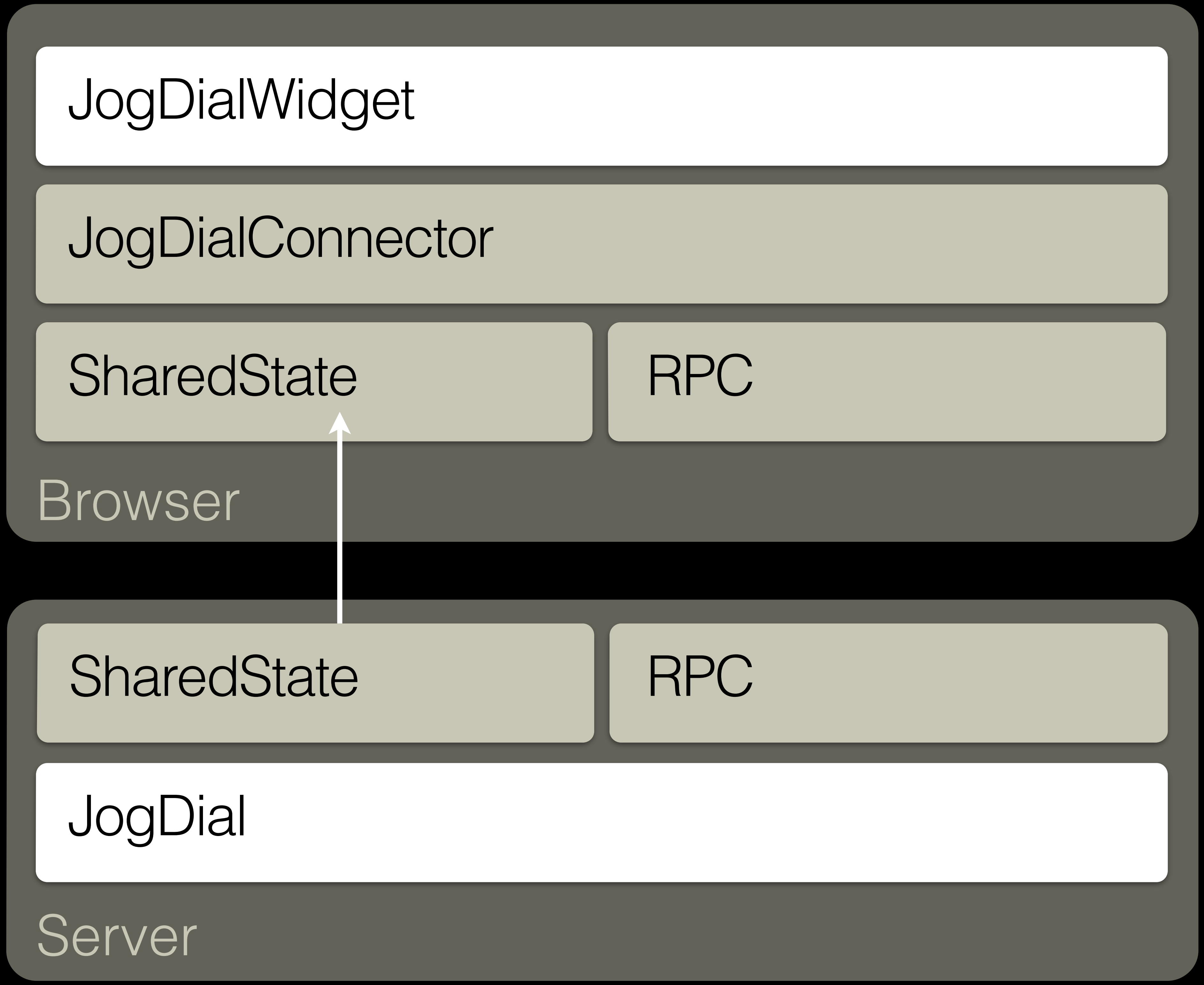
SharedState

Browser

SharedState

JogDial

Server



JogDialWidget

JogDialConnector

SharedState

RPC

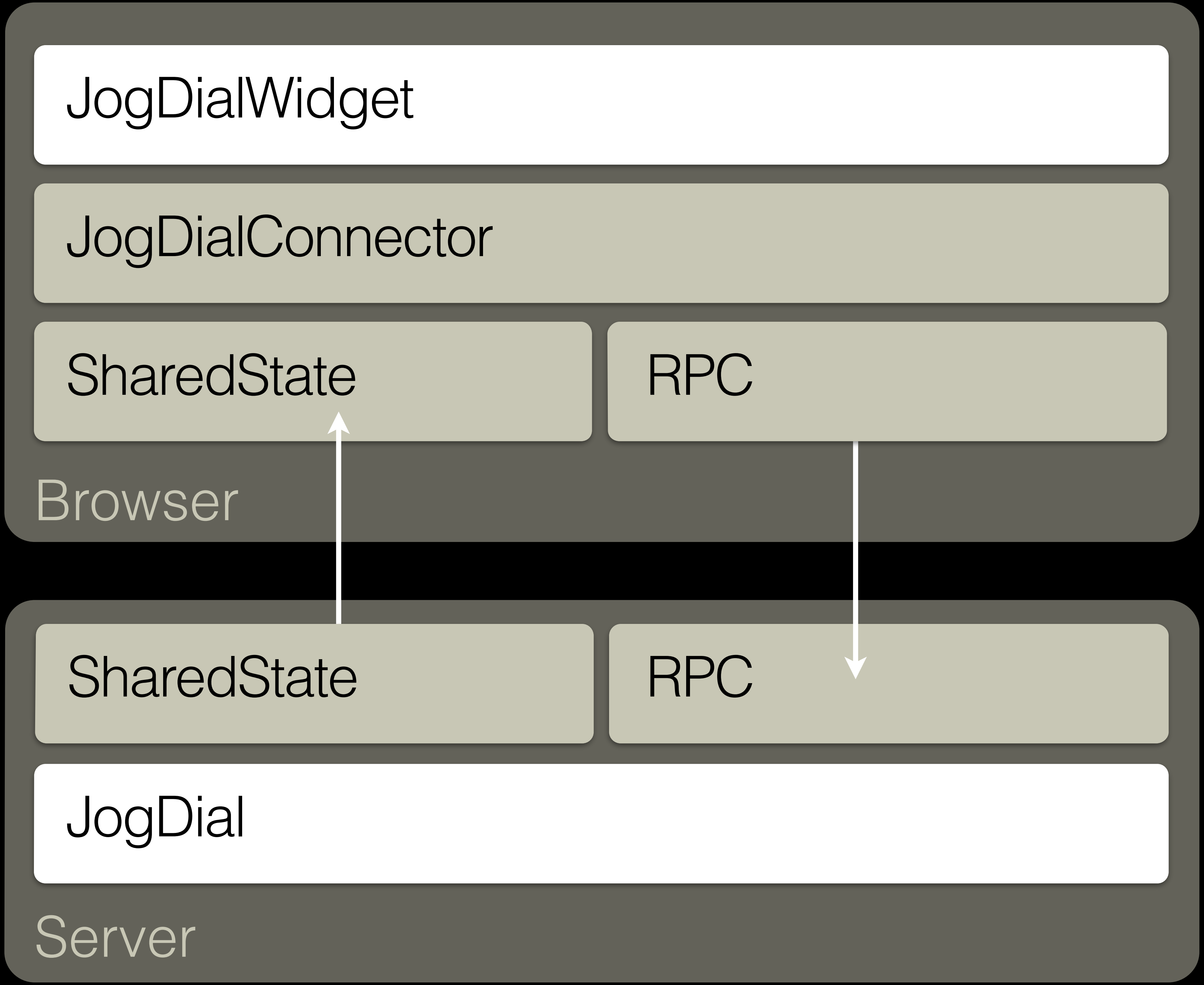
Browser

SharedState

RPC

JogDial

Server



JogDialWidget

JogDialConnector

SharedState

RPC

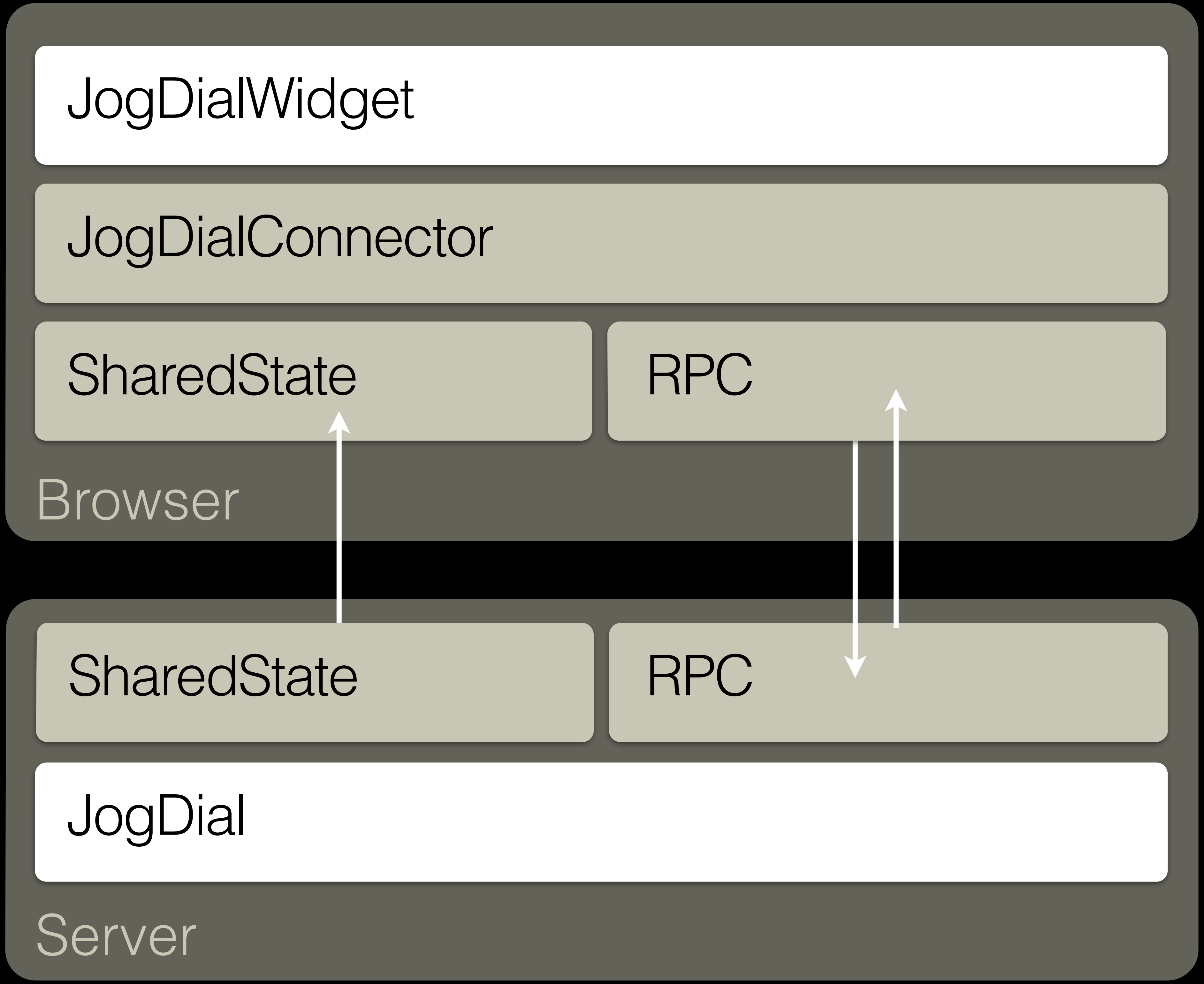
Browser

SharedState

RPC

JogDial

Server



JogDialWidget

JogDialConnector

SharedState

RPC

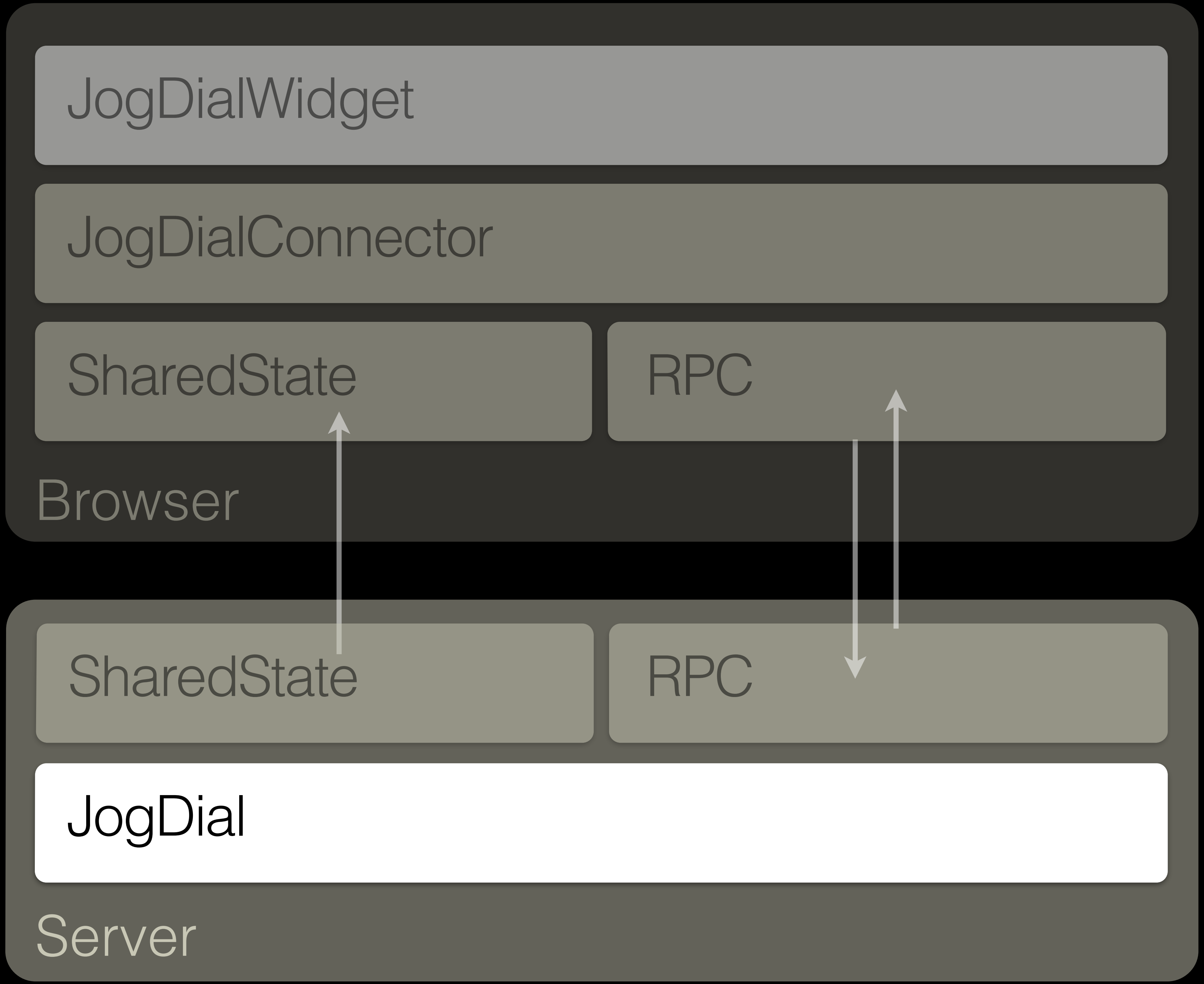
Browser

SharedState

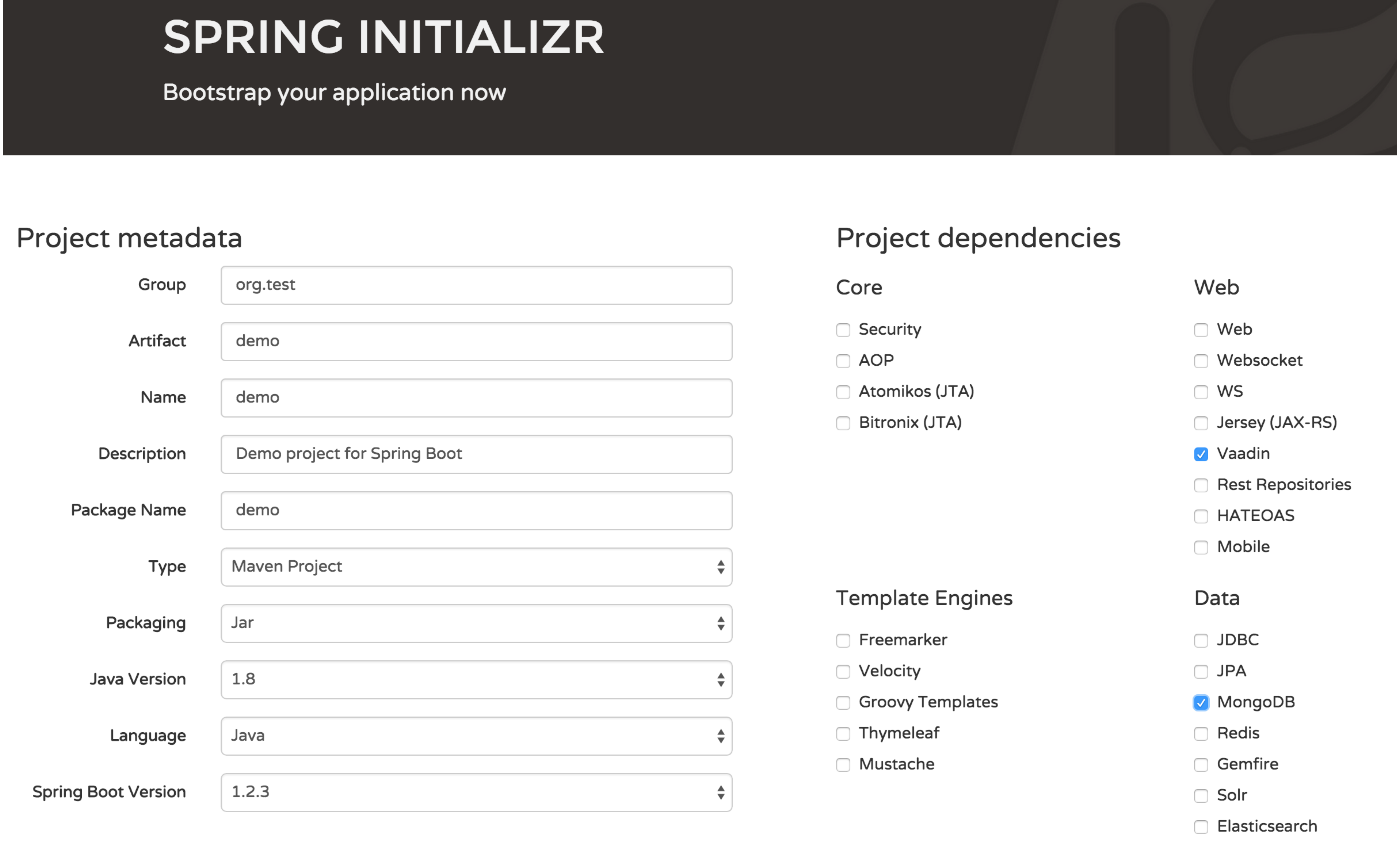
RPC

JogDial

Server


JogDialWidget
JogDialConnector
SharedState
RPC
Browser
SharedState
RPC
JogDial
Server

spring



SPRING INITIALIZR

Bootstrap your application now

Project metadata

Group	org.test
Artifact	demo
Name	demo
Description	Demo project for Spring Boot
Package Name	demo
Type	Maven Project
Packaging	Jar
Java Version	1.8
Language	Java
Spring Boot Version	1.2.3

Project dependencies

Core

□ Security
□ AOP
□ Atomikos (JTA)
□ Bitronix (JTA)

Web

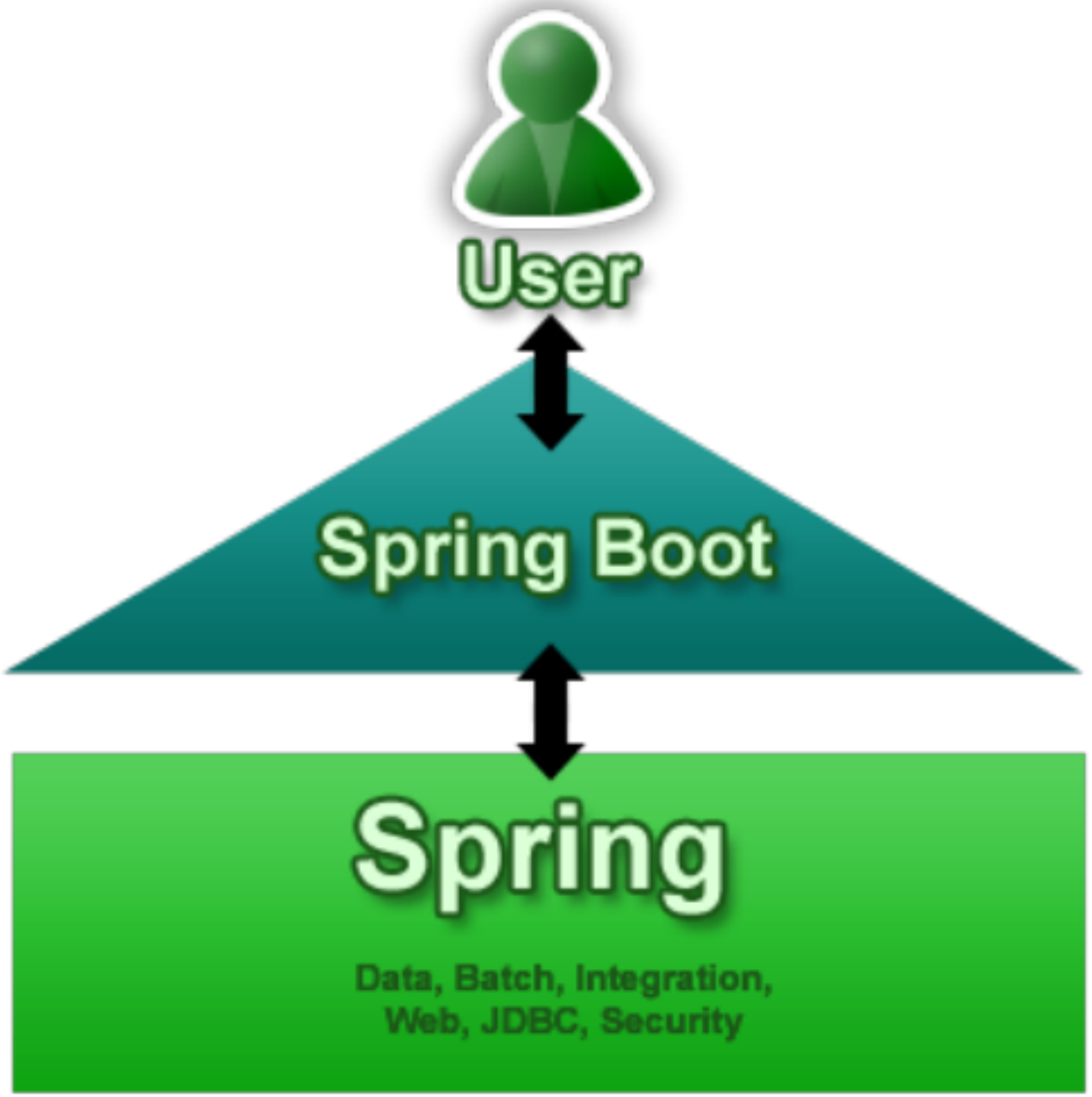
□ Web
□ Websocket
□ WS
□ Jersey (JAX-RS)
☑ Vaadin
□ Rest Repositories
□ HATEOAS
□ Mobile

Template Engines

□ Freemarker
□ Velocity
□ Groovy Templates
□ Thymeleaf
□ Mustache

Data

□ JDBC
□ JPA
☑ MongoDB
□ Redis
□ Gemfire
□ Solr
□ Elasticsearch


User
Spring Boot
Spring
Data, Batch, Integration,
Web, JDBC, Security

spring

→ Single point of focus

spring

→ Single point of focus

→ A tool for getting started quickly

spring

- Single point of focus
- A tool for getting started quickly
- Non-functional requirements

spring

→ Single point of focus

→ A tool for getting started quickly

→ Non-functional requirements

→ Easily customizable

spring

spring

→ Maven POM hierarchy

spring

→ Maven POM hierarchy

→ Annotation based configuration

- Maven POM hierarchy
- Annotation based configuration
- Dependencies as @Bean

spring

- → Maven POM hierarchy
- → Annotation based configuration
- → Dependencies as @Bean
- → Embedded web server

Spring Demo

Approx 5 minutes.

Vaadin & Spring integration

```
<parent>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-parent</artifactId>
   <version>1.2.3.RELEASE</version>
</parent>

<dependencies>
   <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
   </dependency>
   <dependency>
      <groupId>org.vaadin.spring</groupId>
      <artifactId>spring-boot-vaadin</artifactId>
   </dependency>
</dependencies>
```

```
<parent>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-parent</artifactId>
   <version>1.2.3.RELEASE</version>
</parent>

<dependencies>
   <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
   </dependency>
   <dependency>
      <groupId>org.vaadin.spring</groupId>
      <artifactId>spring-boot-vaadin</artifactId>
   </dependency>
</dependencies>
```

```
<parent>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-parent</artifactId>
   <version>1.2.3.RELEASE</version>
</parent>

<dependencies>
   <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
   </dependency>
   <dependency>
      <groupId>org.vaadin.spring</groupId>
      <artifactId>spring-boot-vaadin</artifactId>
   </dependency>
</dependencies>
```

Vaadin Spring Integration Demo

Approx 10 minutes.

spring + vaadin }>

thinking of U and I

spring + vaadin }>

thinking of U and I

→ @SpringUI / @SpringView

spring + vaadin

thinking of U and I

- @SpringUI / @SpringView
- @UIScope

spring + vaadin }>

thinking of U and I

- @SpringUI / @SpringView
- @UIScope
- Spring managed beans

spring + vaadin }>

thinking of U and I

- @SpringUI / @SpringView
- @UIScope
- Spring managed beans
- EventBus for loose coupling

```
@SpringBootApplication
@EnableTouchKitServlet
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args).start();
    }

    @Bean
    DroneTemplate provideTemplate(TaskExecutor taskExecutor,
            DroneStateChangeCallback[] callbacks) throws UnknownHostException {
        return new DroneTemplate(taskExecutor, callbacks);
    }

    @Bean
    Drone provideDrone() {
        return new Drone();
    }
}
```

```
@SpringBootApplication
@EnableTouchKitServlet
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args).start();
    }

    @Bean
    DroneTemplate provideTemplate(TaskExecutor taskExecutor,
            DroneStateChangeCallback[] callbacks) throws UnknownHostException {
        return new DroneTemplate(taskExecutor, callbacks);
    }

    @Bean
    Drone provideDrone() {
        return new Drone();
    }
}
```

```
@SpringBootApplication
@EnableTouchKitServlet
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args).start();
    }

    @Bean
    DroneTemplate provideTemplate(TaskExecutor taskExecutor,
            DroneStateChangeCallback[] callbacks) throws UnknownHostException {
        return new DroneTemplate(taskExecutor, callbacks);
    }

    @Bean
    Drone provideDrone() {
        return new Drone();
    }
}
```

```
@SpringBootApplication
@EnableTouchKitServlet
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args).start();
    }

    @Bean
    DroneTemplate provideTemplate(TaskExecutor taskExecutor,
            DroneStateChangeCallback[] callbacks) throws UnknownHostException {
        return new DroneTemplate(taskExecutor, callbacks);
    }

    @Bean
    Drone provideDrone() {
        return new Drone();
    }
}
```

```java
@TouchKitUI
@Theme("drone")
public class DroneUI extends UI implements InitializingBean, DisposableBean {

    @Autowired
    private DroneTemplate service;

    @Autowired
    private ControlPanel controlPanel;

    @Autowired
    private EventBus eventBus;

    @Autowired
    private DroneEmergencyDialog emergencyDialog;
```

```
@TouchKitUI
@Theme("drone")
public class DroneUI extends UI implements InitializingBean, DisposableBean {

    @Autowired
    private DroneTemplate service;

    @Autowired
    private ControlPanel controlPanel;

    @Autowired
    private EventBus eventBus;

    @Autowired
    private DroneEmergencyDialog emergencyDialog;
```

```
@TouchKitUI
@Theme("drone")
public class DroneUI extends UI implements InitializingBean, DisposableBean {

    @Autowired
    private DroneTemplate service;

    @Autowired
    private ControlPanel controlPanel;

    @Autowired
    private EventBus eventBus;

    @Autowired
    private DroneEmergencyDialog emergencyDialog;
```

```java
@TouchKitUI
@Theme("drone")
public class DroneUI extends UI implements InitializingBean, DisposableBean {

    @Autowired
    private DroneTemplate service;

    @Autowired
    private ControlPanel controlPanel;

    @Autowired
    private EventBus eventBus;

    @Autowired
    private DroneEmergencyDialog emergencyDialog;
    @Override
    public void afterPropertiesSet() throws Exception {
        eventBus.subscribe(this);
    }

    @EventBusListenerMethod
    protected void onEmergencyEvent(DroneEmergencyEvent event) {
       this.access(() -> emergencyDialog.show(event.getEmergencyType(), this));
    }

    @EventBusListenerMethod
    protected void onLowBatteryEvent(DroneLowBatteryEvent event) {
       this.access(() -> emergencyDialog.show(Emergency.BATTERY, this));
    }

    @Override
    public void destroy() throws Exception {
        eventBus.unsubscribe(this);
    }
}
```

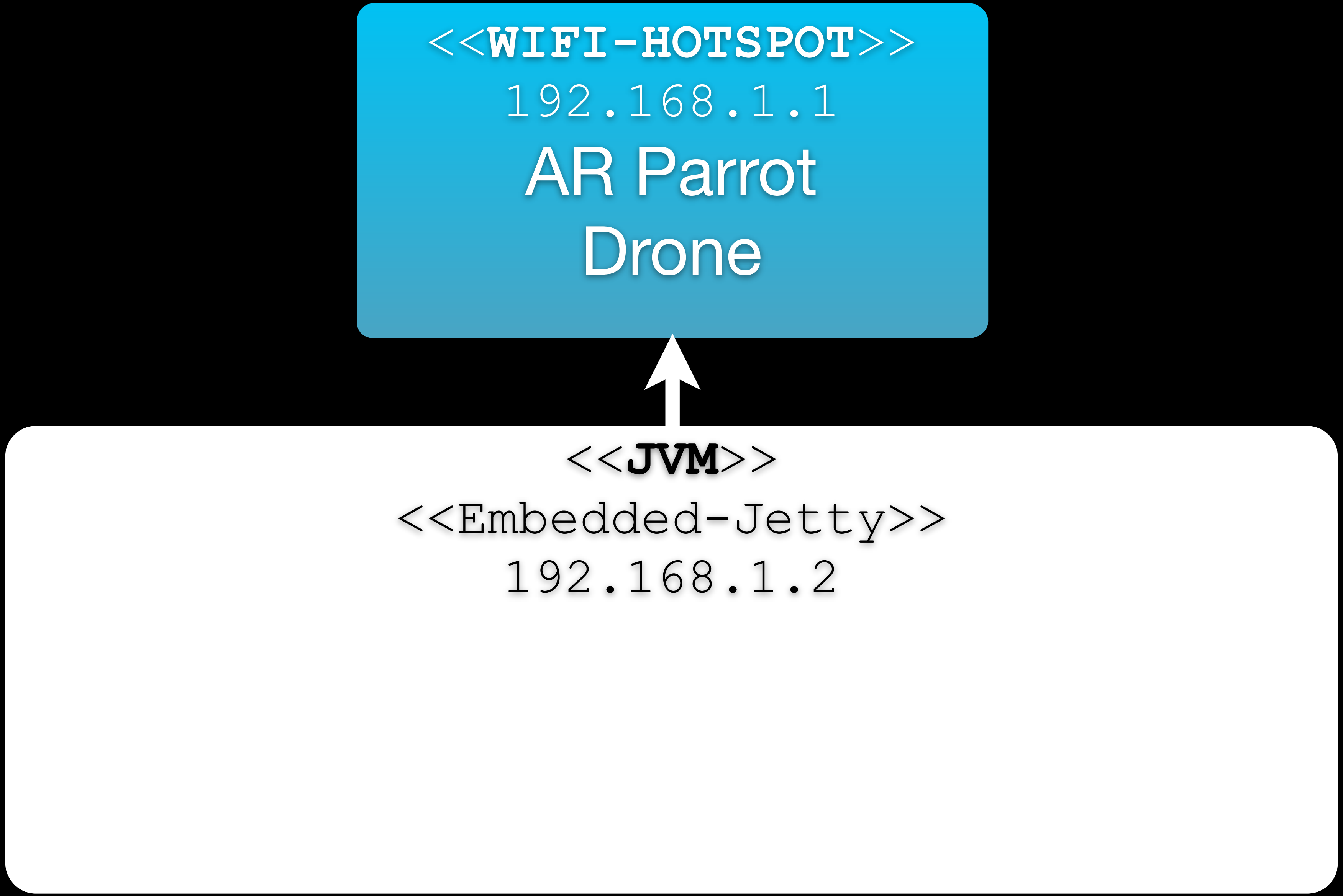
```
    @Override
    public void afterPropertiesSet() throws Exception {
        eventBus.subscribe(this);
    }

    @EventBusListenerMethod
    protected void onEmergencyEvent(DroneEmergencyEvent event) {
       this.access(() -> emergencyDialog.show(event.getEmergencyType(), this));
    }

    @EventBusListenerMethod
    protected void onLowBatteryEvent(DroneLowBatteryEvent event) {
       this.access(() -> emergencyDialog.show(Emergency.BATTERY, this));
    }

    @Override
    public void destroy() throws Exception {
        eventBus.unsubscribe(this);
    }
}
```

Parrot

<<**WIFI-HOTSPOT**>>

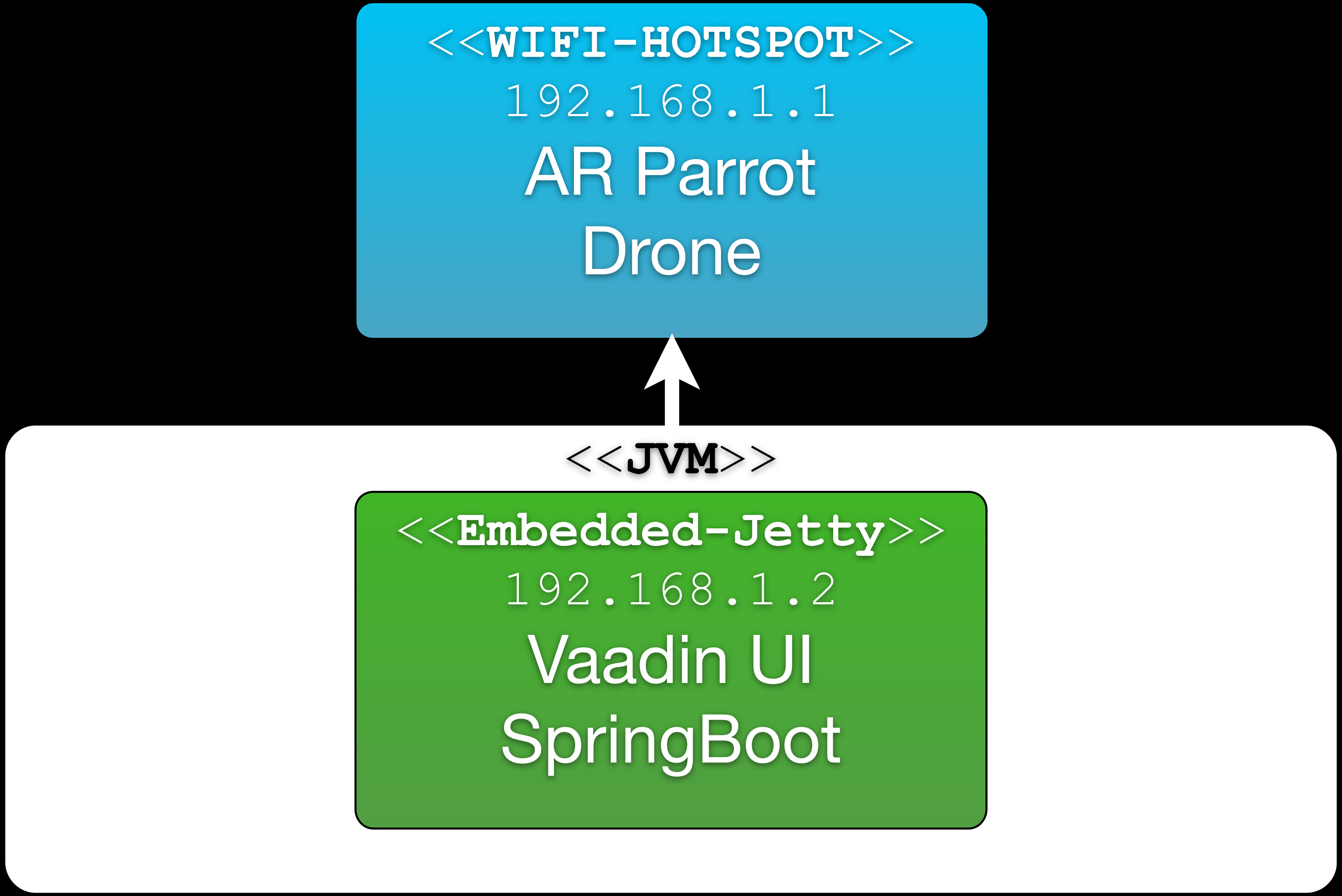
192.168.1.1

AR Parrot
Drone



<<**WIFI-HOTSPOT**>>
192.168.1.1
AR Parrot
Drone

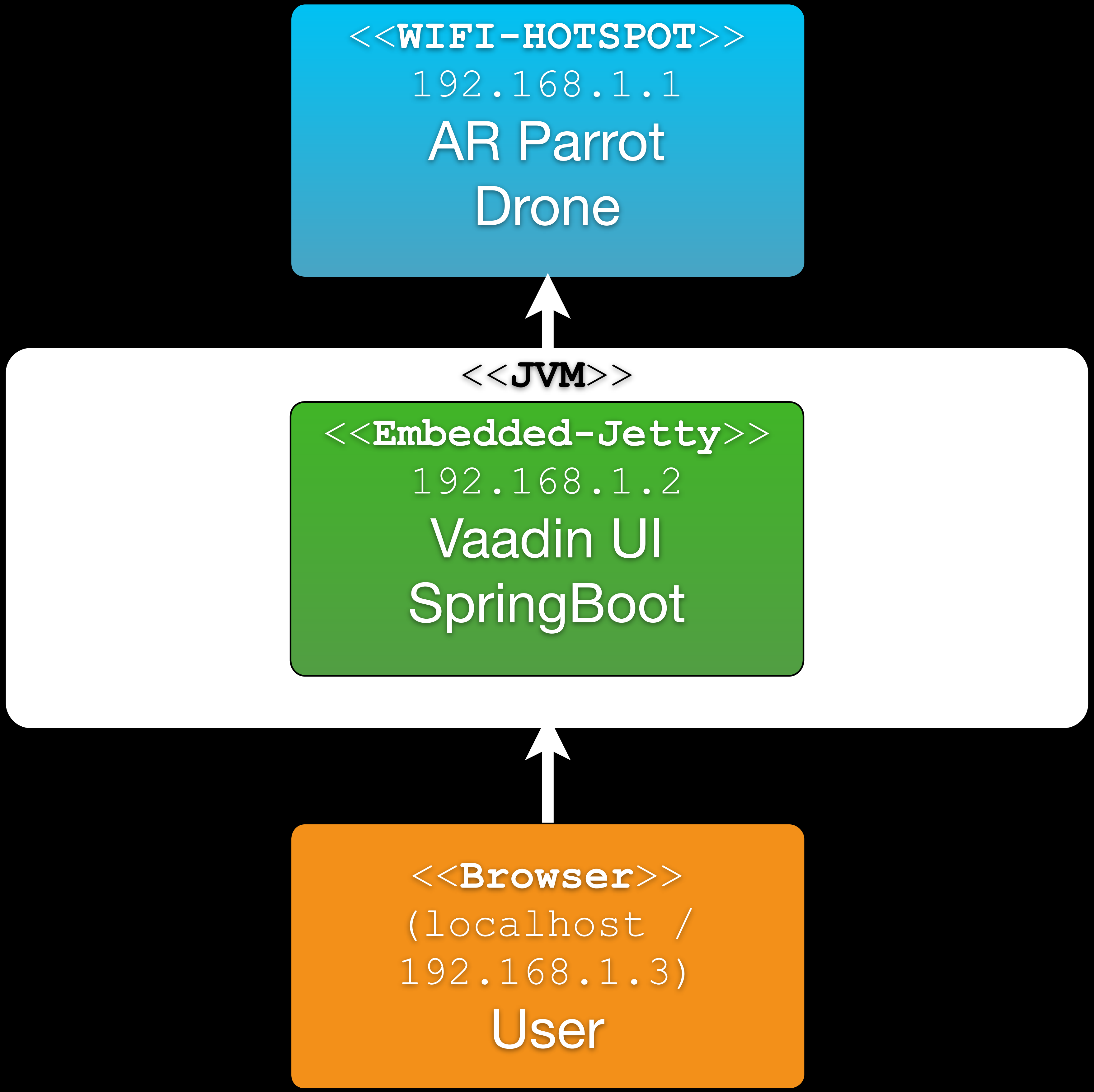
<<**JVM**>>
<<Embedded-Jetty>>
192.168.1.2



<<WIFI-HOTSPOT>>
192.168.1.1
AR Parrot
Drone

<<JVM>>

<<Embedded-Jetty>>
192.168.1.2
Vaadin UI
SpringBoot



<<WIFI-HOTSPOT>>
192.168.1.1
AR Parrot
Drone

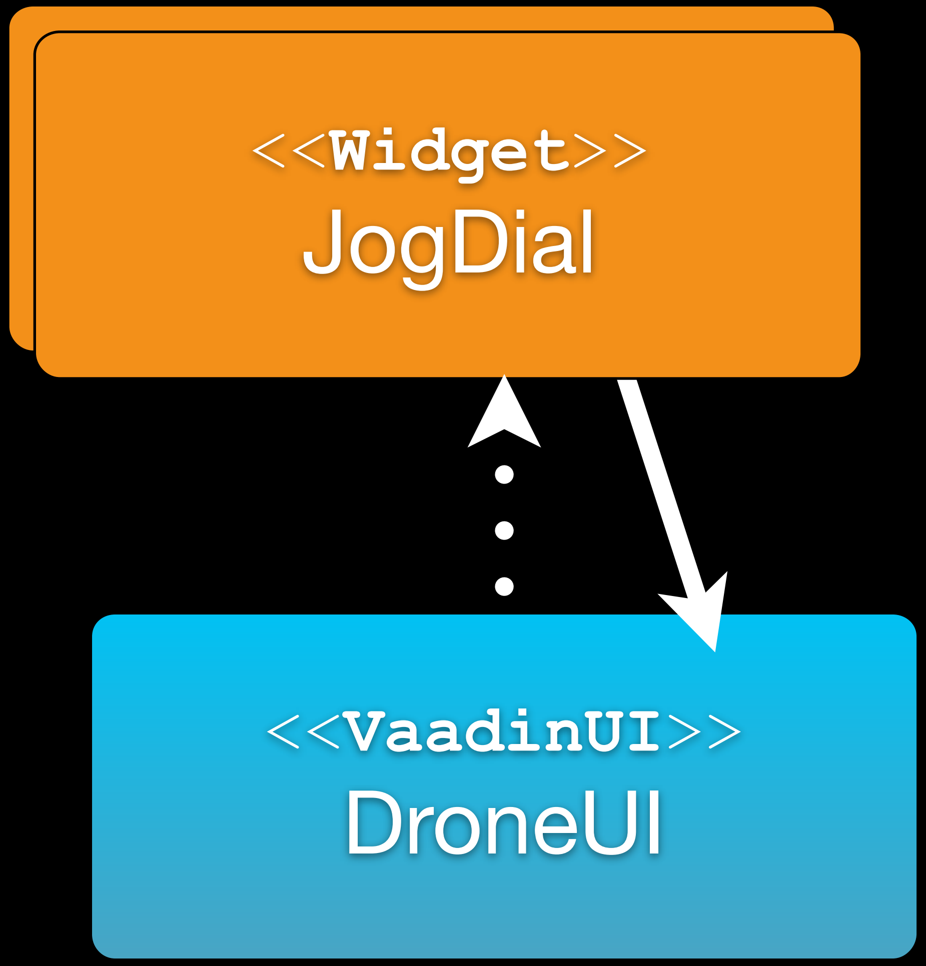
<<JVM>>

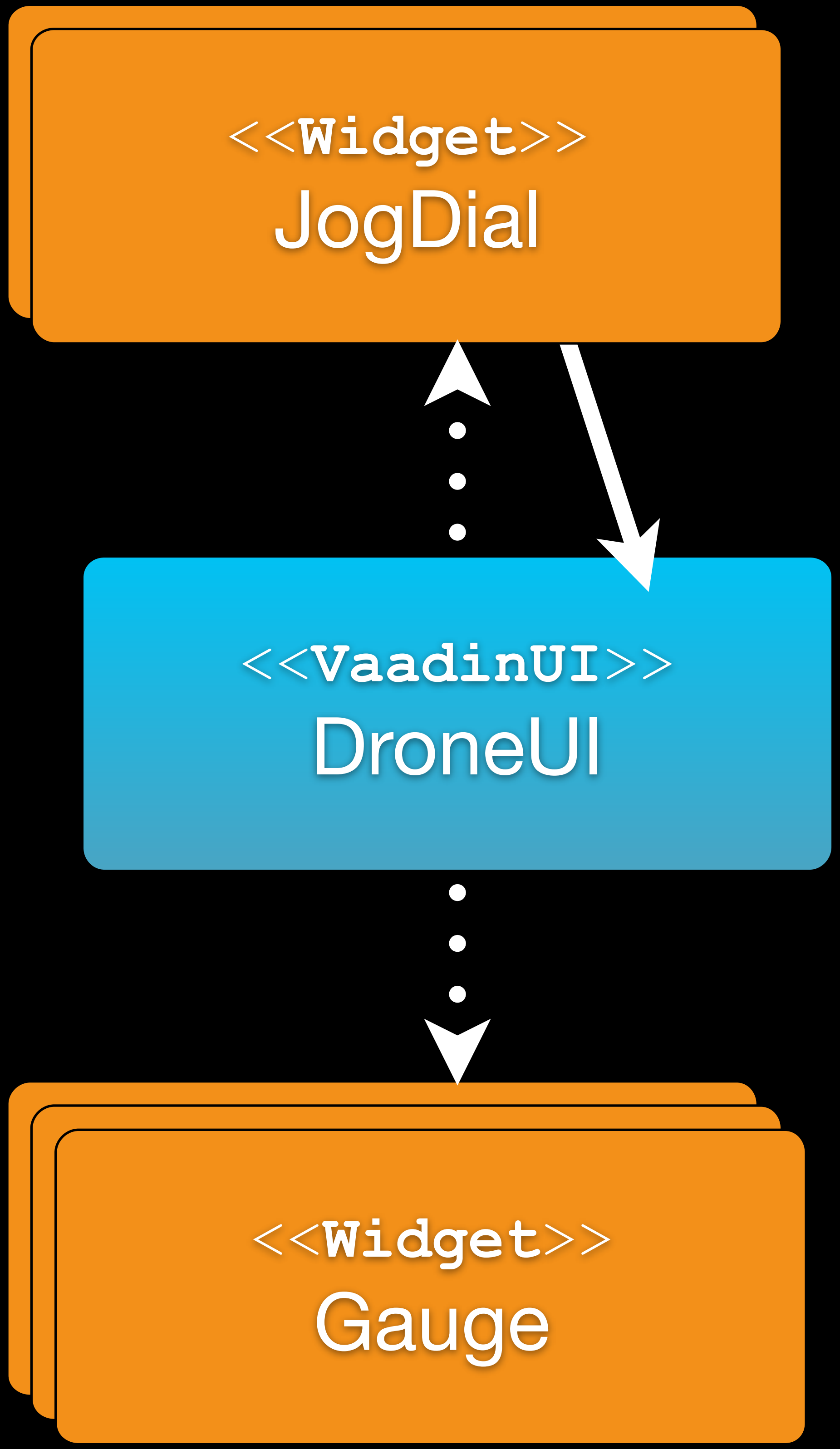
<<Embedded-Jetty>>
192.168.1.2
Vaadin UI
SpringBoot

<<Browser>>
(localhost /
192.168.1.3)
User

<<VaadinUI>>

DroneUI

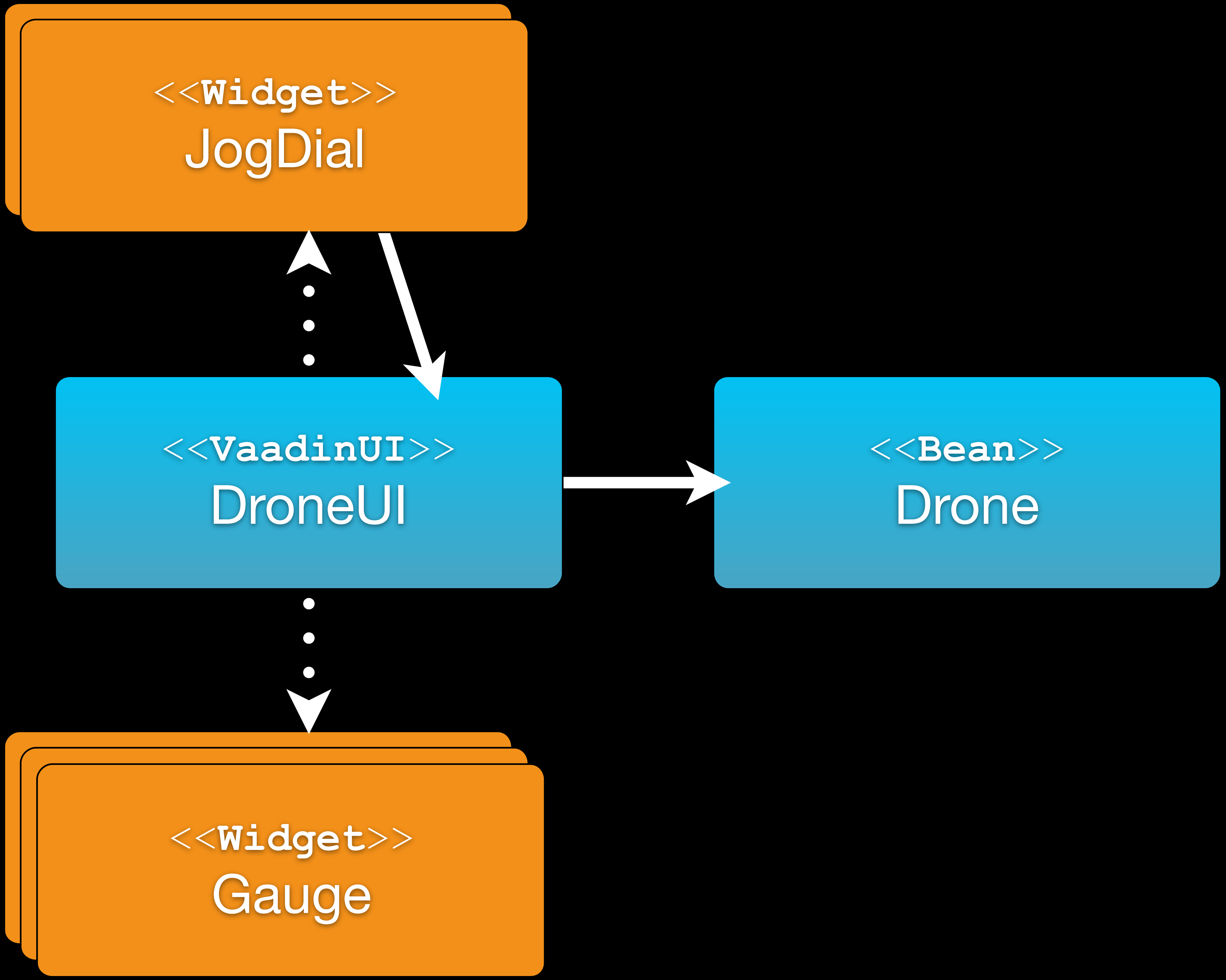

<<Widget>>
JogDial
<<VaadinUI>>
DroneUI

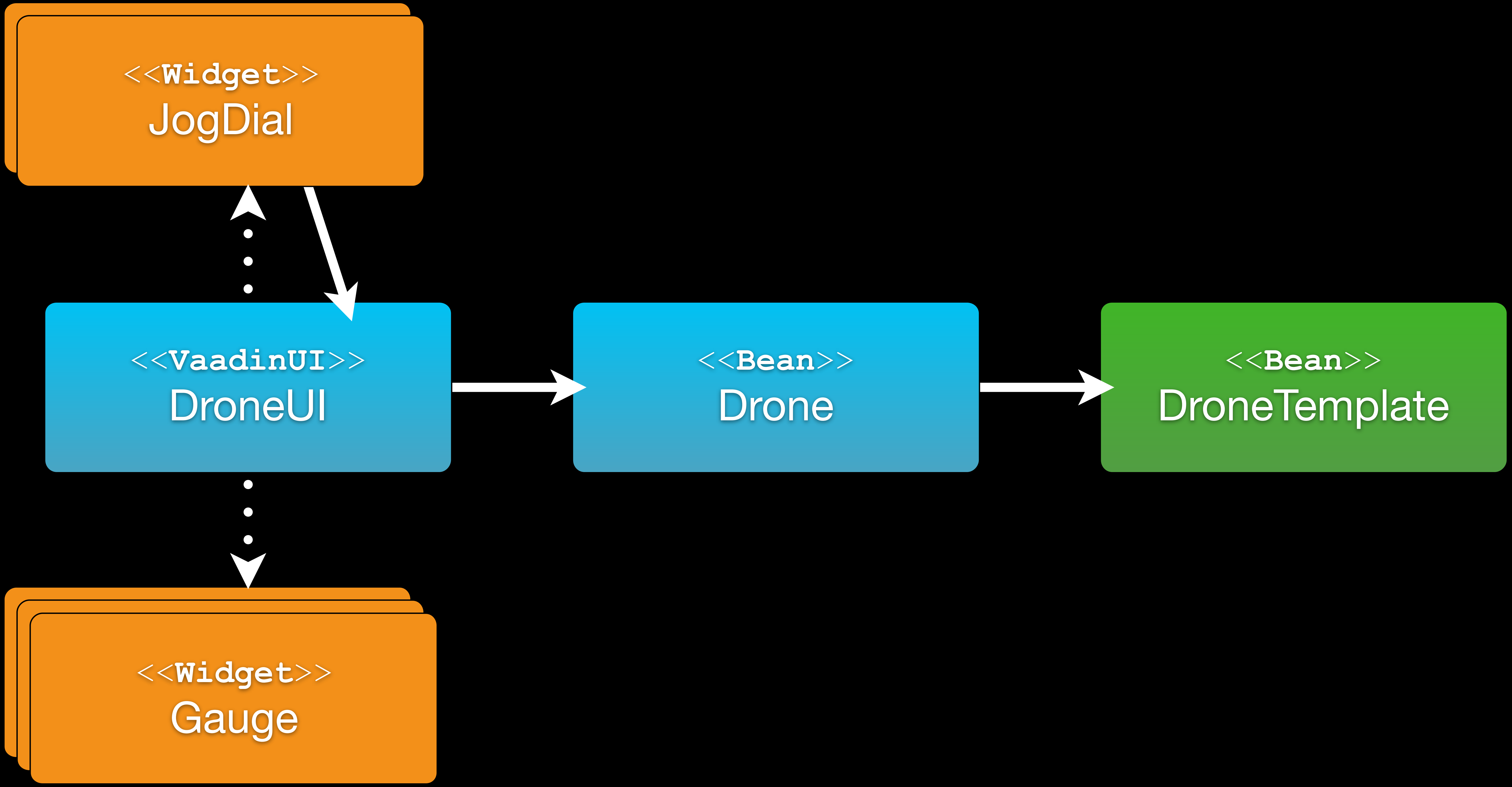


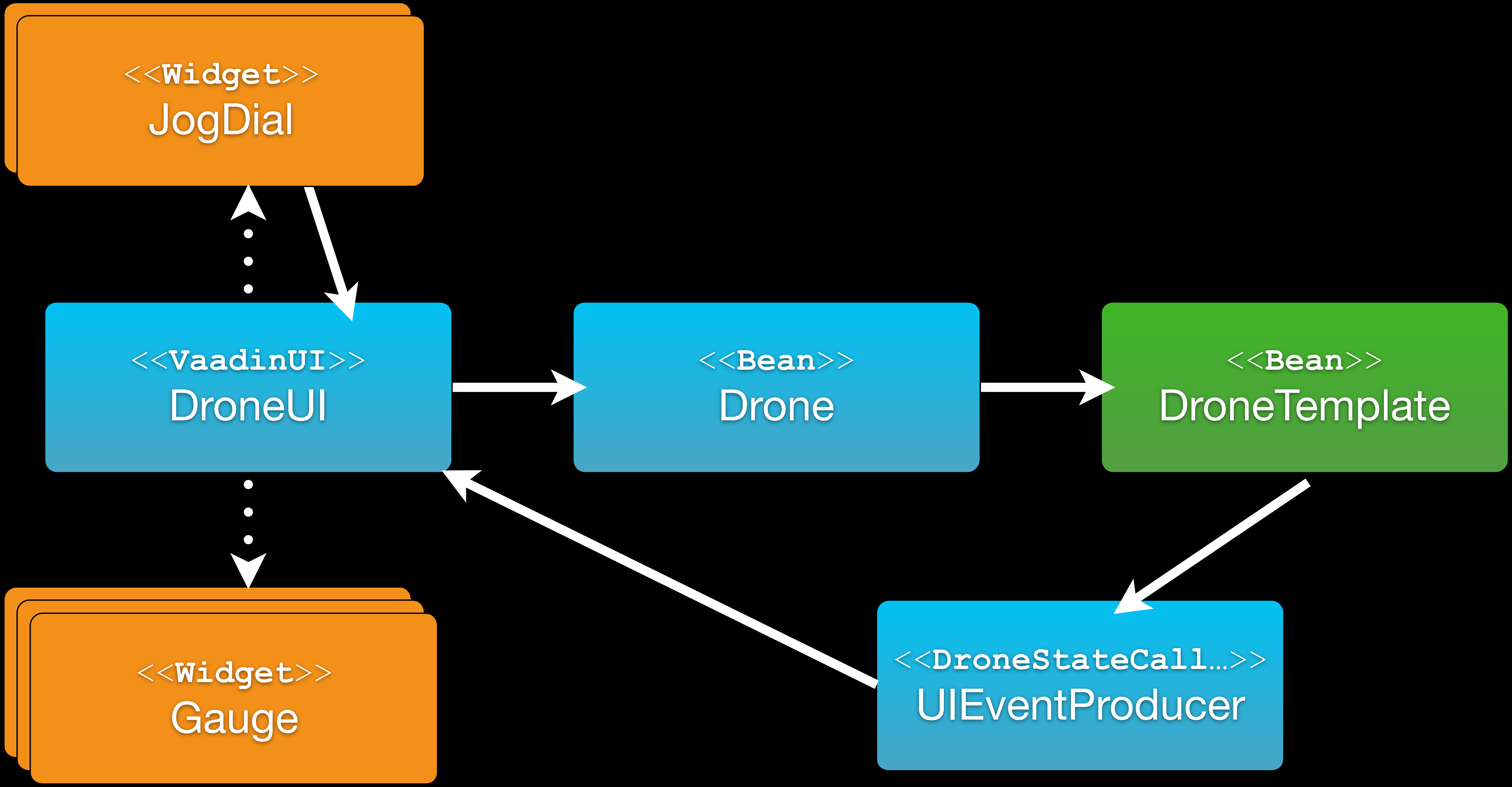
<<Widget>>
JogDial

<<VaadinUI>>
DroneUI

<<Widget>>
Gauge


<<Widget>>
JogDial
<<VaadinUI>>
DroneUI
<<Bean>>
Drone
<<Widget>>
Gauge


<<Widget>>
JogDial
<<VaadinUI>>
DroneUI
<<Bean>>
Drone
<<Bean>>
DroneTemplate
<<Widget>>
Gauge



<<Widget>>
JogDial

<<VaadinUI>>
DroneUI

<<Bean>>
Drone

<<Bean>>
DroneTemplate

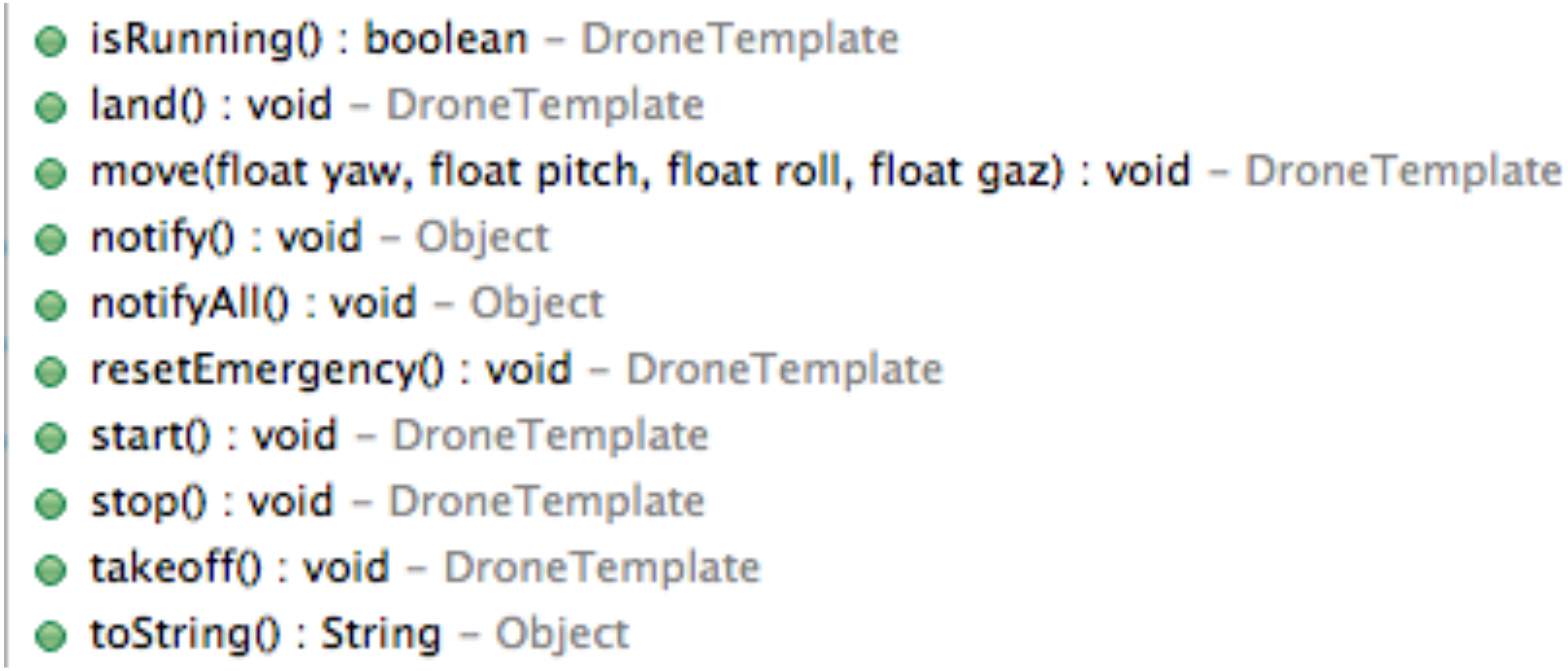
<<Widget>>
Gauge

<<DroneStateCall…>>
UIEventProducer

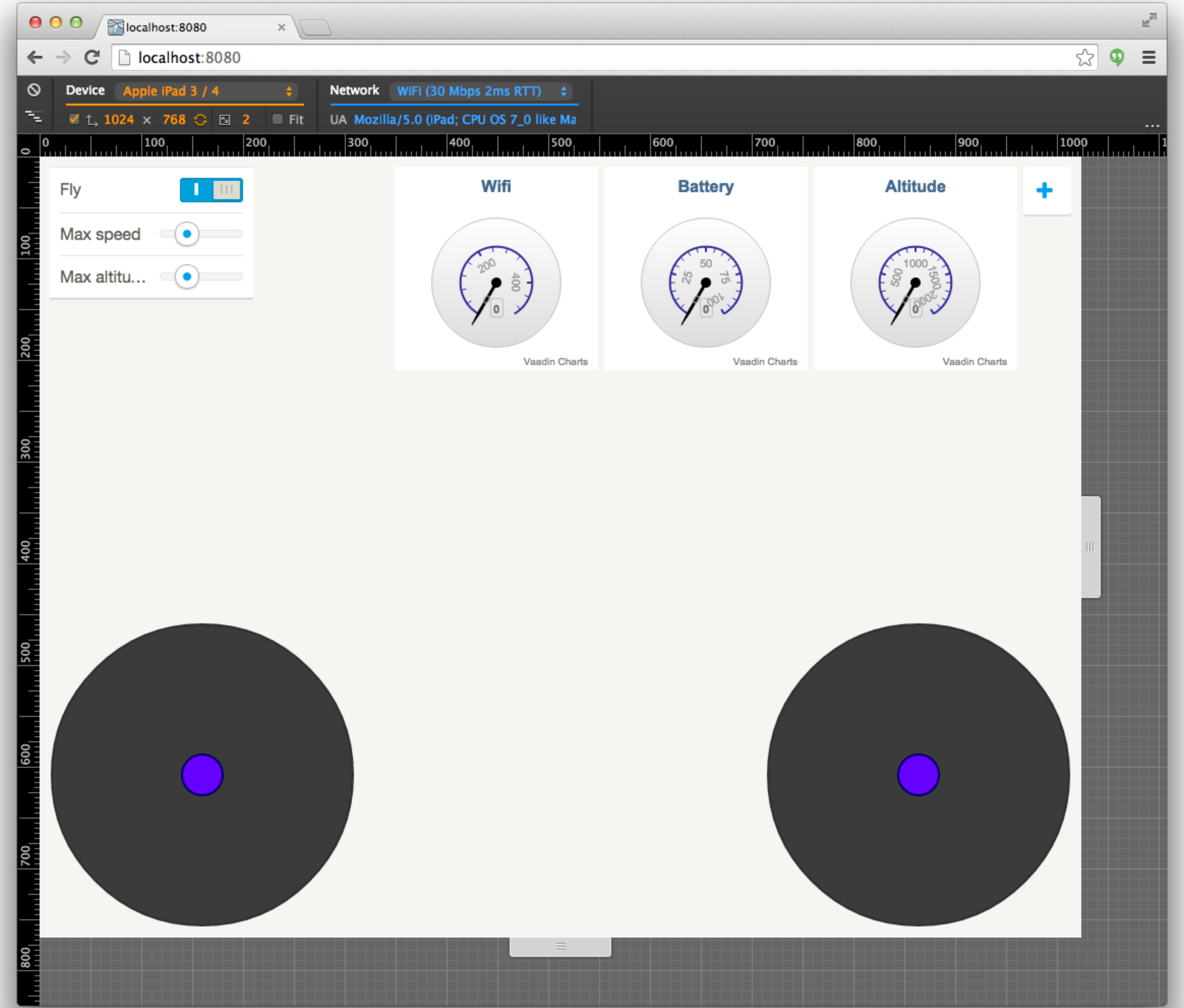
```java
@Component
class UIEventProducer implements DroneStateCallback {

    @Autowired
    @EventBusScope(EventScope.APPLICATION)
    private EventBus eventBus;

    @Override
    public void onDroneStateChanged(DroneState latestState) {
        if (latestState.isEmergency()) {
            eventBus.publish(this, new DroneEmergencyEvent());
        }
        if (latestState.isBatteryTooLow()) {
            eventBus.publish(this, new DroneLowBatteryEvent());
        }

        eventBus.publish(this,
                new DroneBatteryEvent(this, latestState.getBattery()));

        eventBus.publish(this,
                new DroneThetaEvent(this, latestState.getTheta()));

        eventBus.publish(this,
                new DroneAltitudeEvent(this, latestState.getAltitude()));
    }
}
```

```java
@Component
class UIEventProducer implements DroneStateCallback {

    @Autowired
    @EventBusScope(EventScope.APPLICATION)
    private EventBus eventBus;

    @Override
    public void onDroneStateChanged(DroneState latestState) {
        if (latestState.isEmergency()) {
            eventBus.publish(this, new DroneEmergencyEvent());
        }
        if (latestState.isBatteryTooLow()) {
            eventBus.publish(this, new DroneLowBatteryEvent());
        }

        eventBus.publish(this,
                new DroneBatteryEvent(this, latestState.getBattery()));

        eventBus.publish(this,
                new DroneThetaEvent(this, latestState.getTheta()));

        eventBus.publish(this,
                new DroneAltitudeEvent(this, latestState.getAltitude()));
    }
}
```

```
@Component
class UIEventProducer implements DroneStateCallback {

    @Autowired
    @EventBusScope(EventScope.APPLICATION)
    private EventBus eventBus;

    @Override
    public void onDroneStateChanged(DroneState latestState) {
        if (latestState.isEmergency()) {
            eventBus.publish(this, new DroneEmergencyEvent());
        }
        if (latestState.isBatteryTooLow()) {
            eventBus.publish(this, new DroneLowBatteryEvent());
        }

        eventBus.publish(this,
                new DroneBatteryEvent(this, latestState.getBattery()));

        eventBus.publish(this,
                new DroneThetaEvent(this, latestState.getTheta()));

        eventBus.publish(this,
                new DroneAltitudeEvent(this, latestState.getAltitude()));
    }
}
```



- isRunning() : boolean – DroneTemplate
- land() : void – DroneTemplate
- move(float yaw, float pitch, float roll, float gaz) : void – DroneTemplate
- notify() : void – Object
- notifyAll() : void – Object
- resetEmergency() : void – DroneTemplate
- start() : void – DroneTemplate
- stop() : void – DroneTemplate
- takeoff() : void – DroneTemplate
- toString() : String – Object



localhost:8080

Device Apple iPad 3 / 4

1024 × 768 2 □ Fit

Network WiFi (30 Mbps 2ms RTT)

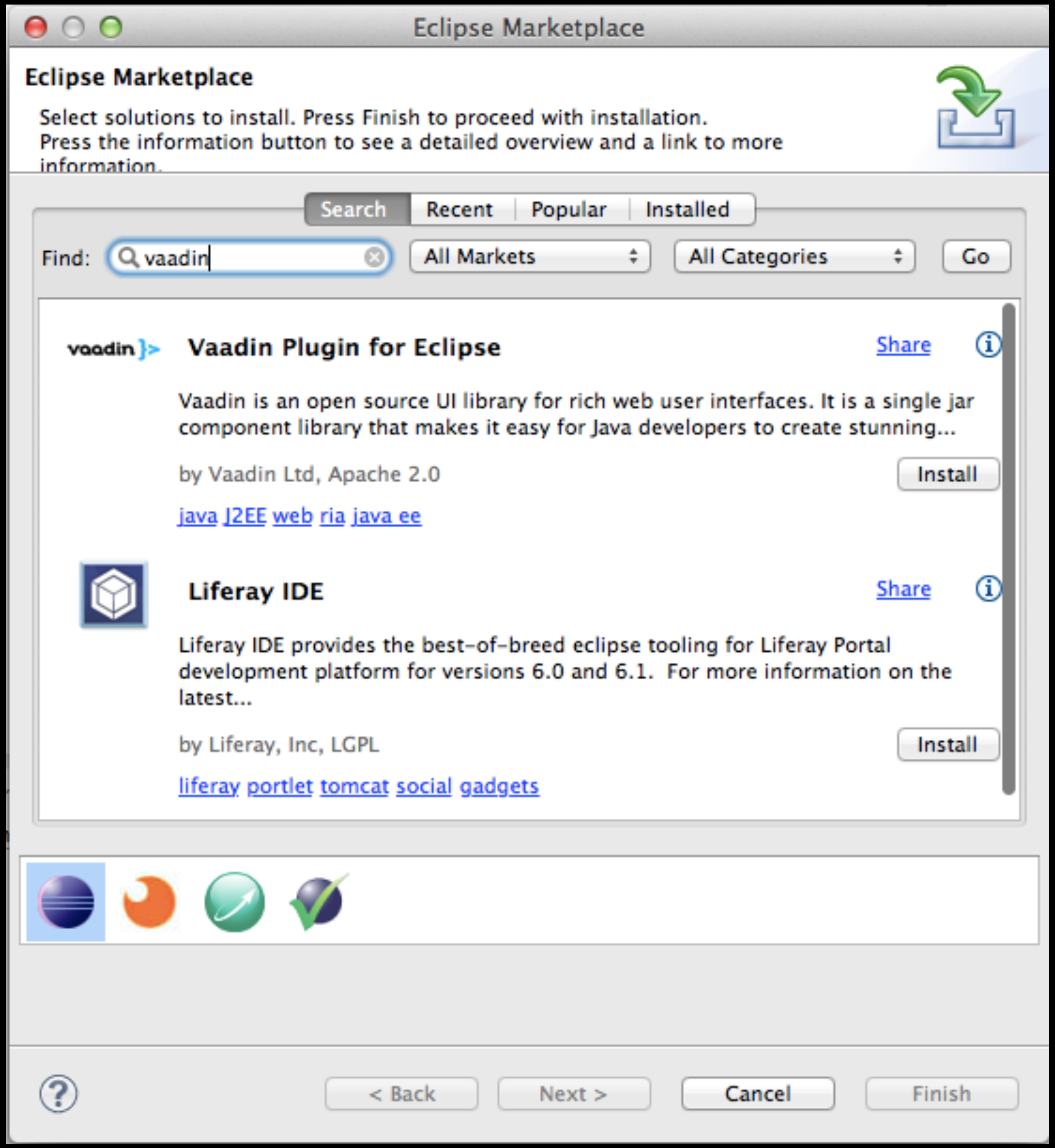
UA Mozilla/5.0 (iPad; CPU OS 7_0 like Ma

Getting started

Getting started

Eclipse

Download Vaadin 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 Plugin for Eclipse Share

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

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

Maven

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

Download for Free

vaadin.com/book

PDF, ePub, HTML

Vol 1
288 pages

Vol 2
434 pages

Lessons learned today

Lessons learned today

1. Drones are super cool!

Lessons learned today

1. Drones are super cool!
2. Vaadin is good for web apps, brings you great productivity

Lessons learned today

1. Drones are super cool!
2. Vaadin is good for web apps, brings you great productivity
3. Boot kickstarts development and helps you all the way

Lessons learned today

1. Drones are super cool!
2. Vaadin is good for web apps, brings you great productivity
3. Boot kickstarts development and helps you all the way
4. Official Vaadin Spring integration is already in Beta

JavaCro

Thank you!

Peter Lehto
@peter_lehto
expert & trainer
vaadin }>