

• GENERAL •

JavaFx

• OVERVIEW •



GERRIT

GRUNWALD

canoo Engineering AG

TWITTER: @HANSOLO_

BLOG: HARMONIC-CODE.ORG

Agenda

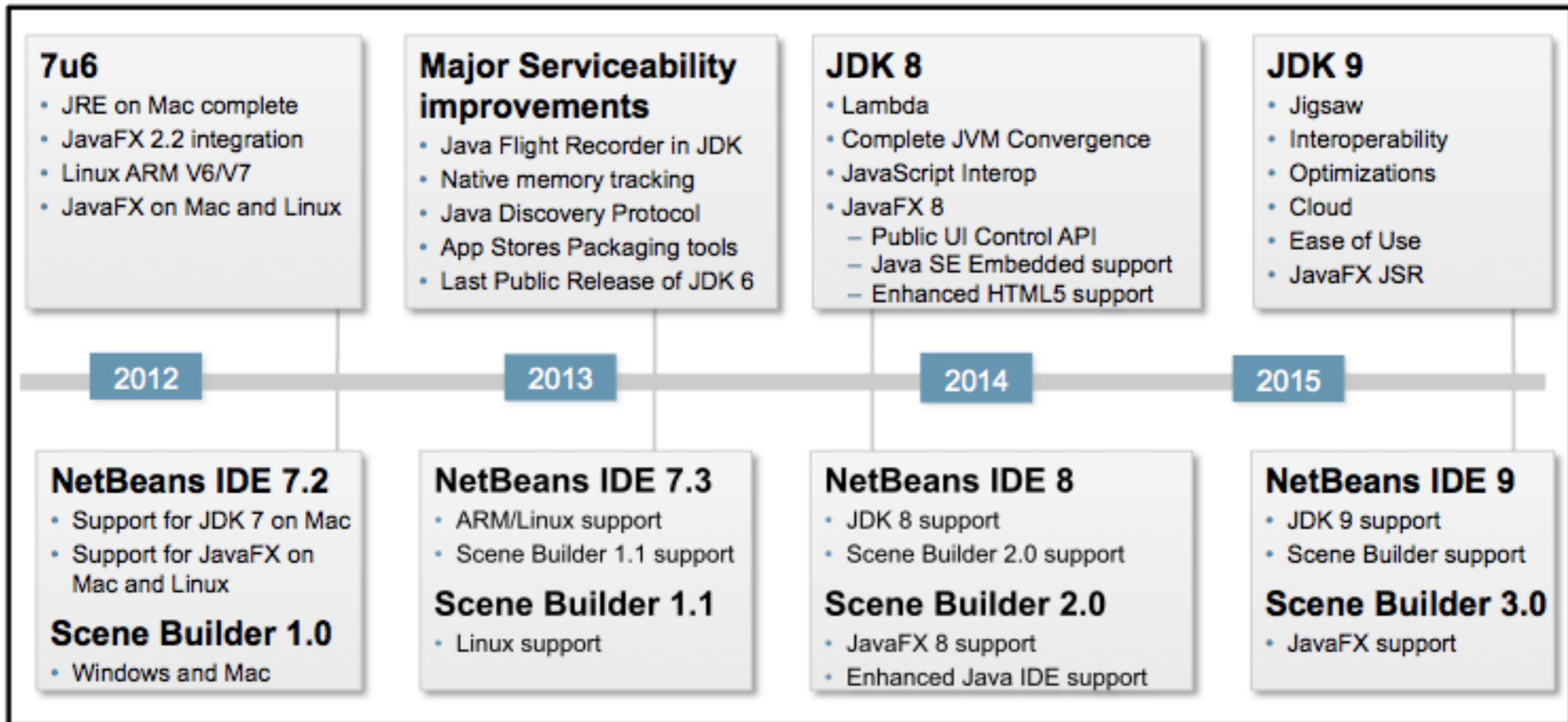
- * **HISTORY**
- * **NEW PLUGIN**
- * **SCENE GRAPH**
- * **JAVA API**
- * **PROPERTIES**
- * **BINDINGS**
- * **CONTROLS**
- * **CSS**
- * **WEBVIEW**
- * **JFXPANEL**
- * **CHARTS**
- * **FXML**

Some

HISTORY

11/2006	F3
05/2007	JAVA FX 1.0
02/2009	JAVA FX 1.1
06/2009	JAVA FX 1.2
04/2010	JAVA FX 1.3
08/2010	JAVA FX 1.3.1
10/2011	JAVA FX 2.0
04/2012	JAVA FX 2.1
08/2012	JAVA FX 2.2

Roadmap



What

JAVA FX

really is...

It is the successor to

JAVA SWING

and it's still not

FINISHED

Available for

- * **WINDOWS**
- * **MAC OS X**
- * **LINUX**
- * **ARM (PREVIEW)**

Versions

* **JAVAFX 2.2 BUNDLED WITH JDK**

> JAVA 7U6

* **STANDALONE FOR JAVA6**

The architecture

JavaFX Public API's and Scene Graph

Quantum Toolkit

Prism

Glass
Windowing
Toolkit

Media Engine

Web Engine

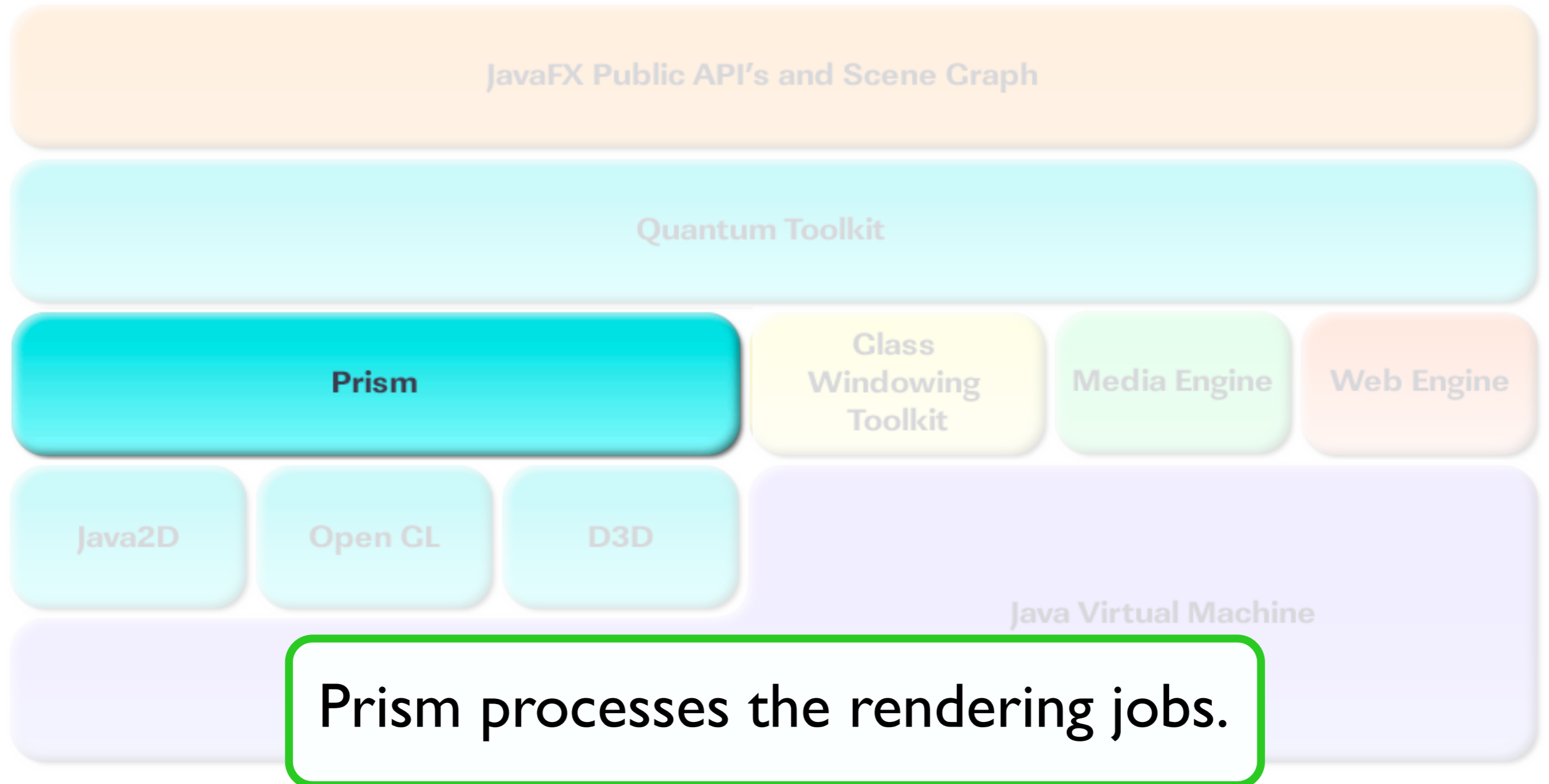
Java2D

Open GL

D3D

Java Virtual Machine

The architecture



The architecture

JavaFX Public API's and Scene Graph

Quantum Toolkit

Prism

Glass
Windowing
Toolkit

Media Engine

Web Engine

Java2D

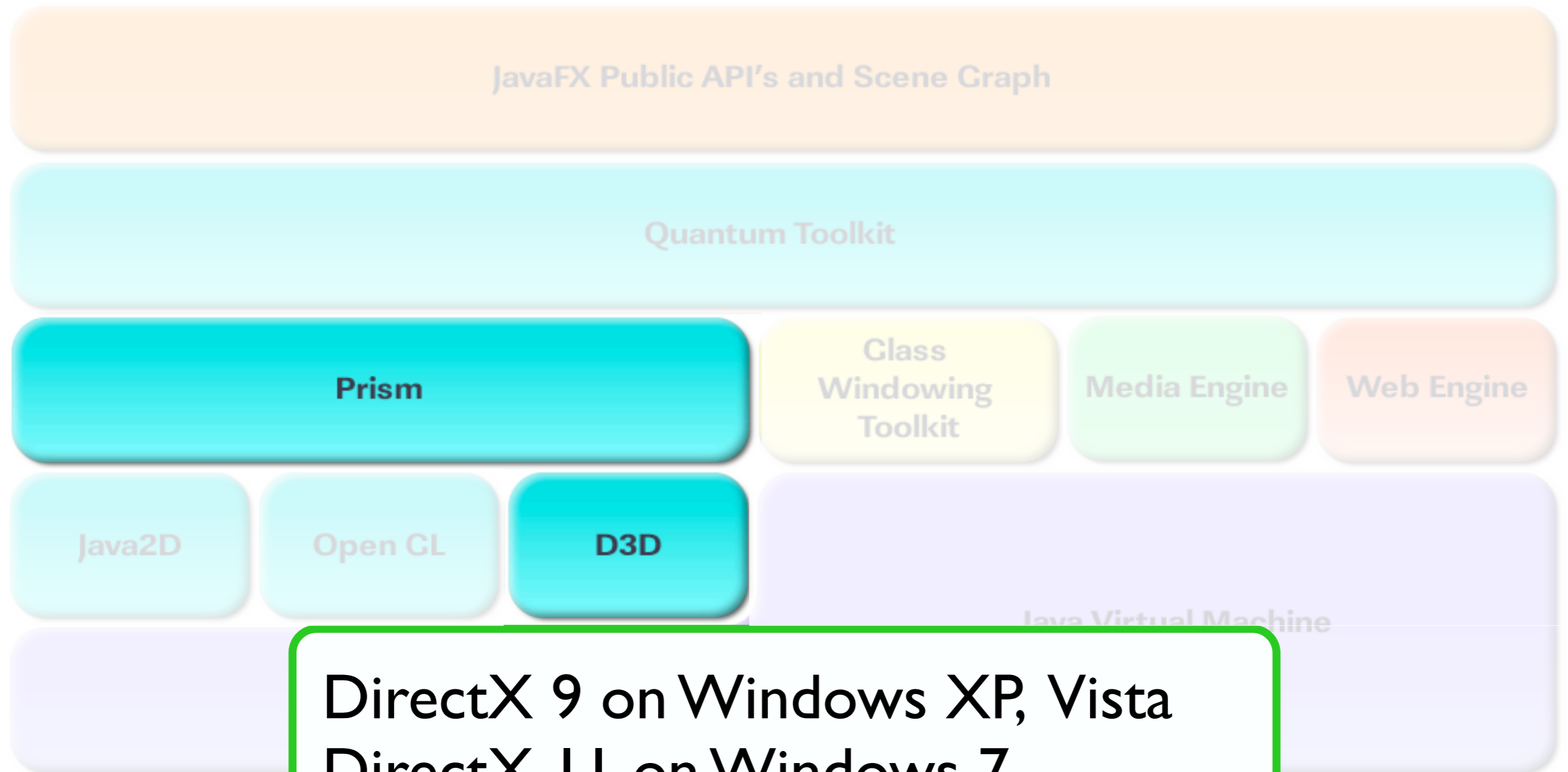
Open GL

D3D

Java Virtual Machine

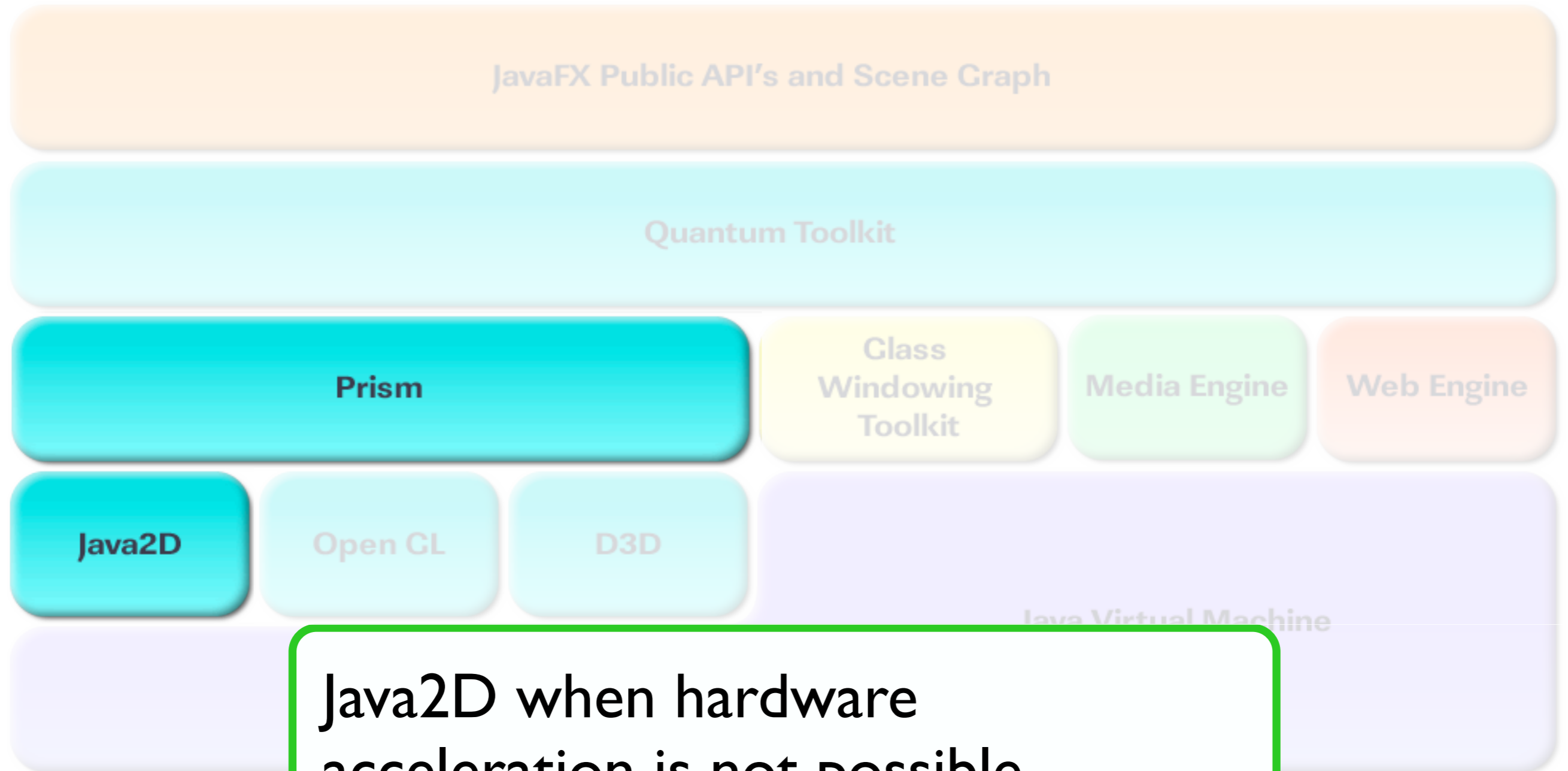
OpenGL on Mac, Linux, Embedded

The architecture



DirectX 9 on Windows XP, Vista
DirectX 11 on Windows 7

The architecture



Java2D when hardware acceleration is not possible

The architecture

JavaFX Public API's and Scene Graph

Quantum Toolkit

Prism

Glass
Windowing
Toolkit

Media Engine

Web Engine

Java2D

Open GL

D3D

Provides low level native operating system services

The architecture

JavaFX Public API's and Scene Graph

Quantum Toolkit

Prism

Glass
Windowing
Toolkit

Media Engine

Web Engine

Java2D

Ties Prism and Glass Windowing Toolkit together and makes them available to the JavaFX layer above

The architecture

JavaFX Public API's and Scene Graph

Quantum Toolkit

Prism

Glass
Windowing
Toolkit

Media Engine

Web Engine

Java2D

Open GL

D3D

Java Virtual Machine

Open Source

* **JAVAFX SOURCE CODE IS PART
OF THE OPEN JFX PROJECT**

[HTTP://OPENJDK.JAVA.NET/PROJECTS/OPENJFX/](http://openjdk.java.net/projects/openjfx/)

completely open
source around 02/2013

Again a new

PLUGIN

Browser Plugin

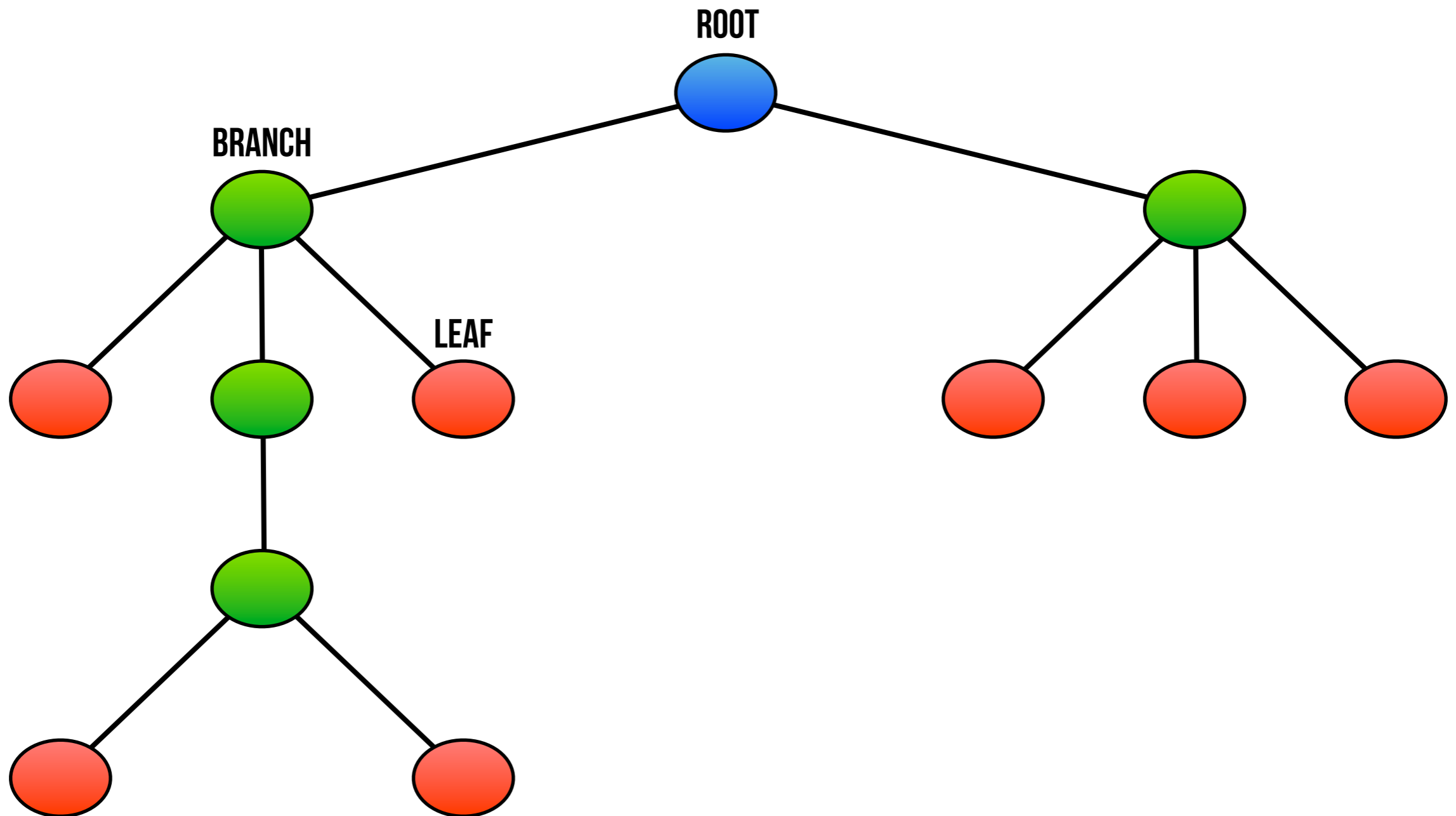
* **FASTER LOADING OF JAVAFX
WEB APPS BASED ON PRISM**

* **PRE-LOADER FOR IMPROVED
USER EXPERIENCE**

The

SCENE GRAPH

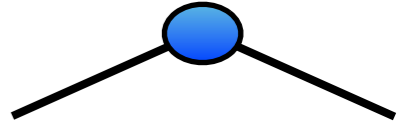
Scene Graph



Scene Graph

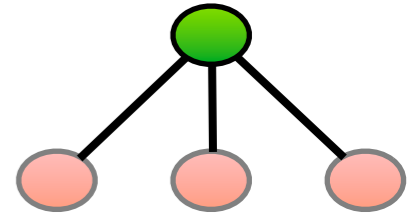
- * **HANDLES THE UI**
- * **TREE STRUCTURE**
- * **ONE ROOT NODE**
- * **BRANCH + LEAF NODES**

Root Node



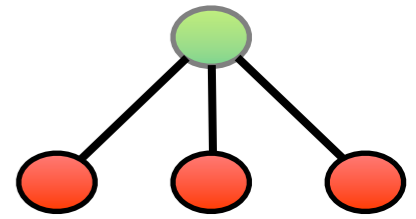
- * **THE ONLY NODE WITHOUT A PARENT NODE**

Branch Nodes



- * **ARE DERIVED FROM**
`javafx.scene.Parent`
- * **CAN CONTAIN OTHER NODES**

Leaf Nodes



* **SHAPES**

* **MEDIA**

* **IMAGES**

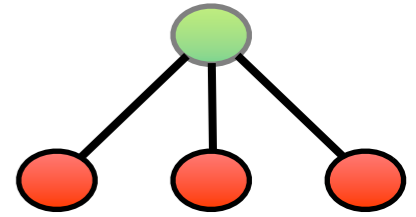
* **CONTROLS**

* **TEXT**

* **CHARTS**

* **WEBVIEW**

Leaf Nodes



* **HAVE NO**
getChildren()

Scene Graph

- * **ROOT NODE IS A STACKPANE**
- * **STAGE IS CONTAINER FOR ROOT**
- * **ALIVE...NO DEAD BITMAPS**

A typical app

```
public class SceneGraphStructure extends Application {  
    @Override public void start(Stage stage) {  
        stage.setTitle("Hello World");  
        Button button = new Button("Say 'Hello World'");  
        button.setOnAction(new EventHandler<ActionEvent>() {  
            @Override public void handle(ActionEvent evt) {  
                System.out.println("Hello World");  
            }  
        });  
        StackPane root = new StackPane();  
        root.getChildren().add(button);  
        stage.setScene(new Scene(root, 300, 250));  
        stage.show();  
    }  
}
```

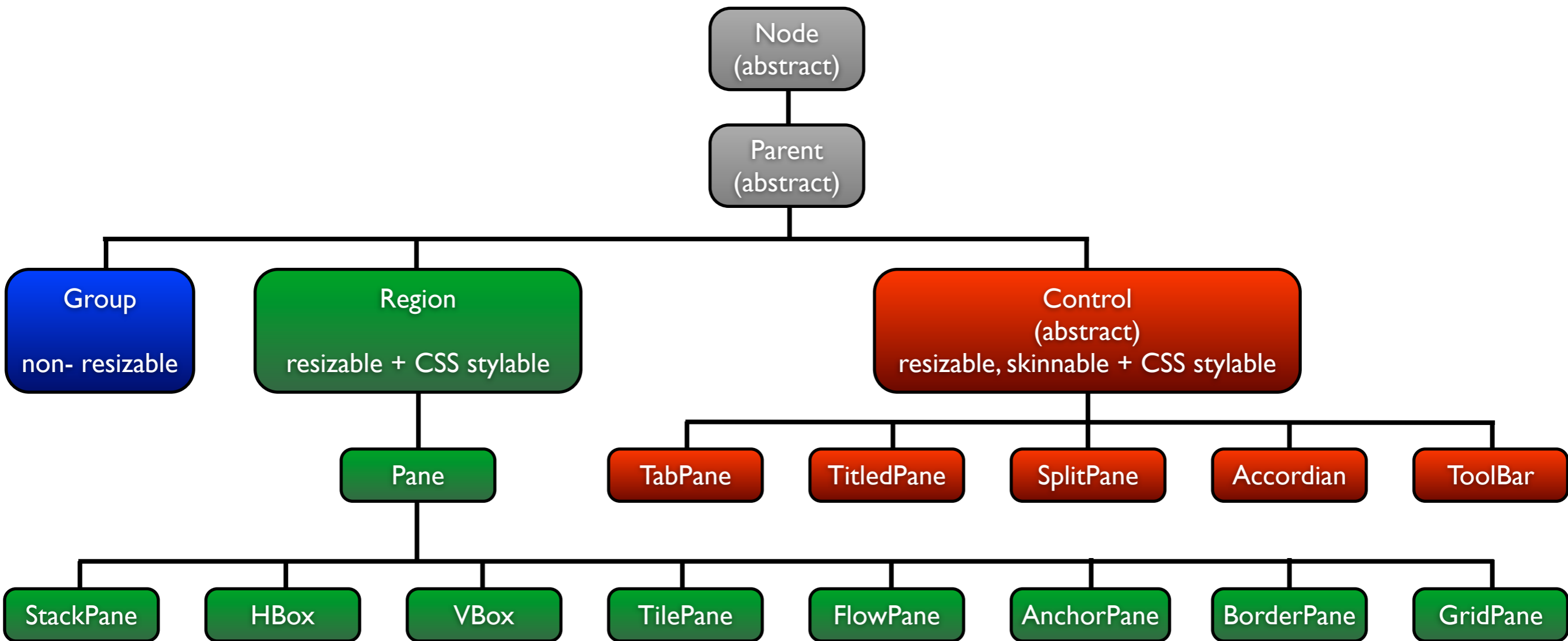
Scene Graph

```
public static void main(String[] args) {  
    launch(args);  
}
```

start JavaFx application

```
}
```

Layout classes



The

JAVA

Api

JavaFx Script is

NODEAD

It lives on as

VISAGE

Now we have

PURE JAVA

Some

EXAMPLES

Code examples

```
// Java FX 1.x
public def timer = Timeline {
  repeatCount: Timeline.INDEFINITE
  keyframes: KeyFrame {
    time: 1s
    action: function() {...}
  }
}
```

```
// Java FX 2.x
private Timeline timer =
TimelineBuilder.create()
  .cycleCount(Timeline.INDEFINITE)
  .keyFrames(
    new KeyFrame(Duration.seconds(1),
      new EventHandler() {...}
    )
  )
  .build();
```

Code examples

```
// Java FX 1.x
view = ImageView {
    image:image
    translateX:bind x + (view.scaleX - 1)
    translateY:bind y + (view.scaleY - 1)
};
```

```
// Java FX 2.x
view = new ImageView(image);
view.translateXProperty().bind(
    x.add(view.getScaleX() - 1));
view.translateYProperty().bind(
    y.add(view.getScaleY() - 1));
```

Properties and

BINDINGS

Properties

```
// Property string
private static final String VALUE_PROPERTY = "value";

// A double property
double value;

// The getter method
public double getValue() {
    return value;
}

// The setter method
public void setValue(double newValue) {
    double oldValue = value;
    value = newValue;
    firePropertyChange(VALUE_PROPERTY, oldValue, value);
}
```

Java Swing

Properties

```
// A double property
DoubleProperty value;

// The getter method
public double getValue() {
    return value.get();
}

// The setter method
public void setValue(double newValue) {
    value.set(newValue);
}

// The property method
public DoubleProperty valueProperty() {
    return value;
}
```

JavaFx

Bindings

- * **HIGH-LEVEL BINDING**
- * **FLUENT API**
- * **BINDINGS CLASS**
- * **LOW-LEVEL BINDING**

Bindings

* **UNIDIRECTIONAL BINDING**

bind();

* **BIDIRECTIONAL BINDING**

bindBidirectional();

High-Level

```
IntegerProperty number1 = new SimpleIntegerProperty(1);  
IntegerProperty number2 = new SimpleIntegerProperty(2);  
DoubleProperty  number3 = new SimpleDoubleProperty(0.5);
```

```
// High-Level Binding (Fluent API)
```

```
NumberBinding sum      = number1.add(number2);
```

```
NumberBinding result = number1.add(number2).multiply(number3);
```

```
// High-Level Binding (Binding class)
```

```
NumberBinding result = Bindings.add(number1, number2);
```

```
NumberBinding result = Bindings.add(number1, multiply(number2, number3));
```

High-Level

- * **FLUENT API IS SELFEXPLAINING**
- * **MORE READABLE CODE**
- * **MIGHT BE A BIT SLOWER**

Low-Level

```
IntegerProperty number1 = new SimpleIntegerProperty(1);
IntegerProperty number2 = new SimpleIntegerProperty(2);
DoubleProperty  number3 = new SimpleDoubleProperty(0.5);

// Low-Level Binding
DoubleBinding db = new DoubleBinding() {
    {
        super.bind(number1, number2, number3);
    }

    @Override protected double computeValue() {
        return (number1.get() + number2.get() * number3.get());
    }
}
```

Low-Level

- * **OVERRIDES A BINDING CLASS**
- * **IS MORE FLEXIBLE**
- * **COULD BE FASTER**

JavaFx

CONTROLS

Some

EXAMPLES

- ▶ Node 1
- ▼ Node 2
- String
- ▶ Node 3

Dog

Color Color Color Color Color Color Color

- Hello
- Bye
- Disabled

Red Orange Yellow Green Blue Indigo Violet

Cat Dog Horse

Button 1 Button 2

Button 1 Button 2

Button 1 Button 2

- ▼ Root node
- Child Node 1
- Child Node 2
- ▼ Child Node 3
- Child Node 4
- Child Node 5
- Child Node 6
- Child Node 7
- Child Node 8

100,00 €
-12,34 €
33,01 €
71,00 €
23.000,00 €
-6,00 €
0,00 €
42.223,00 €
-12,05 €



Text

Label styled as a bar



A simple label with a graphic on the left.



[Hyperlink with Image](#)



button

Left Button Center Button Right Button

◀ 1 2 3 4 5 6 7 ▶

Tab 1 Tab 2 Tab 3 Tab 4

- Make a choice... ▼
- Option 1
 - Option 2
 - Option 3
 - Option 4
 - Option 5
 - Option 6
 - Longer ComboBox item
 - Option 7

Edit or Choose... ▼

◀ ● ● ● ● ● ● ▶

2/7

3/7



- Simple checkbox
- Three state checkbox
- Disabled

Invited	First	Last	Email
<input checked="" type="checkbox"/>	Jacob	Smith	jacob.smith@example.com
<input type="checkbox"/>	Isabella	Johnson	isabella.johnson@example.com
<input checked="" type="checkbox"/>	Ethan	Williams	ethan.williams@example.com
<input checked="" type="checkbox"/>	Emma	Jones	emma.jones@example.com
<input type="checkbox"/>	Michael	Brown	michael.brown@example.com



Control structure

* **CONTROL**

* **SKIN**

* **BEHAVIOR**

* **CSS**

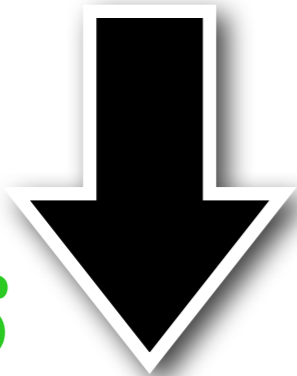
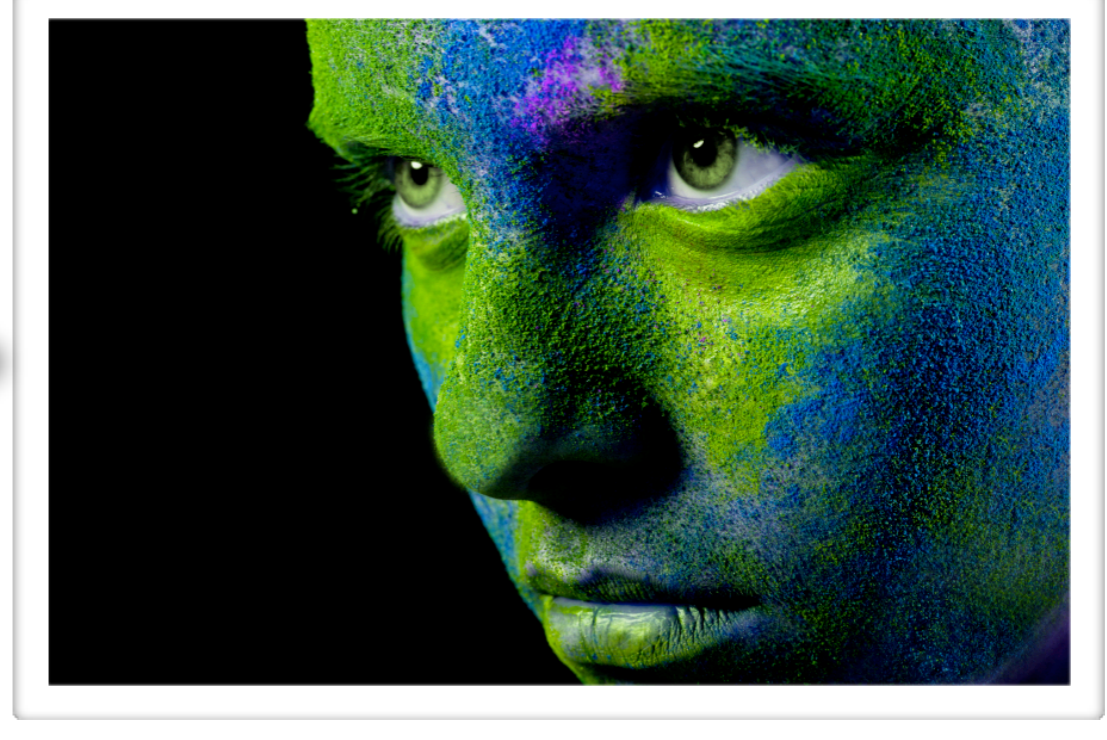
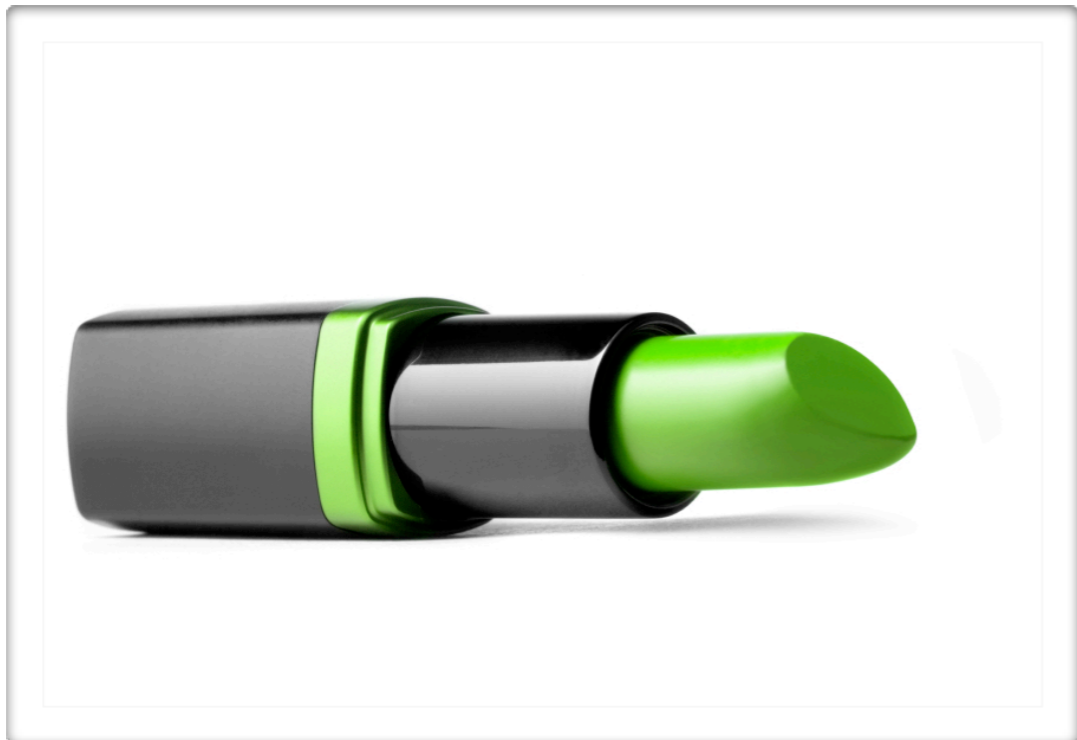


Control

Behavior

CSS

Skin



Styling with

css

Remember

LOOK + FEELS

in Swing ?

Forget them...

Using CSS

- * **ONE DEFAULT CSS CASPIAN.CSS
FOR ROOT AND CONTROLS**
- * **JAVAFX CSS IS BASED ON W3C
CSS 2.1 + SOME ADDITIONS**

Using CSS

- * **EITHER OVERRIDE THE DEFAULTS TO STYLE YOUR APP**
- * **OR APPLY YOUR OWN CSS FILE**

The Caspian.css

```
.root {  
  -fx-base           : #d0d0d0;  
  -fx-background     : #f4f4f4;  
  -fx-color          : -fx-base;  
  -fx-hover-base    : ladder(-fx-base,  
                             derive(-fx-base, 20%) 20%,  
                             derive(-fx-base, 30%) 35%,  
                             derive(-fx-base, 40%) 50%);  
  -fx-pressed-base  : derive(-fx-base, -20%);  
  -fx-focused-base  : -fx-base;  
  -fx-body-color    : linear-gradient(to bottom,  
                                       derive(-fx-color, 34%) 0%,  
                                       derive(-fx-color, -18%) 100%);  
  ...  
}
```

The default CSS

```
.button {  
  -fx-skin                : "com.sun.javafx.scene.control.skin.ButtonSkin";  
  -fx-background-color    : -fx-shadow-highlight-color, -fx-outer-border,  
                           -fx-inner-border, -fx-body-color;  
  -fx-background-insets  : 0 0 -1 0, 0, 1, 2;  
  -fx-background-radius  : 5, 5, 4, 3;  
  -fx-padding             : 0.166667em 0.833333em 0.25em 0.833333em;  
  -fx-text-fill          : -fx-text-base-color;  
  -fx-alignment          : CENTER;  
  -fx-content-display     : LEFT;  
}
```



Standard

The custom CSS

```
.root {  
  -fx-base: #252525; /* scene.getRoot().setStyle("-fx-base: #252525"); */  
}  
  
.button {  
  -fx-font-family      : "Verdana";  
  -fx-font-size        : 16px;  
  -fx-background-radius: 9, 9, 8, 7;  
  -fx-padding          : 9px 16px 9px 16px;  
}
```



Custom

A simple app



Simple Application

Name

Lastname

Adress

Caspian Styler

FX Experience Tools

Caspian Styler

Animation Spline Editor

Derived Color Calculator

Label: ToggleBu...
Button: ToggleBu...
Hyperlink: Menu... Choice...
CheckBox:
RadioButton:
ComboBox: Editable ComboBox
Slider:
TextField: TextArea

Text
Font: System Default
Font Size: 15

Sizes
Padding: 5
Border Width: 1
Radius: 5

Styling
Simple Advanced

Colors
Base: #D0D0D0
Text on Base: #000000 Auto
Background: #F4F4F4
Text on Background: #000000 Auto
Field Background: #FFFFFF
Field Text: #000000 Auto
Focus: #0093FF

Style
Top Highlight:
Bottom Highlight:
Gradient: Default

Copy Stylesheet Save Stylesheet

Apply some CSS

```
Scene scene = new Scene(pane, Color.rgb(75, 75, 75));  
scene.getStylesheets().add("file:///customstylesheet.css");
```



```
.root {
  -fx-font-family      : "Verdana";
  -fx-font-size        : 13.0px;
  -fx-base              : #363636;
  -fx-background       : #5C5C5C;
  -fx-focus-color      : #FF001B;
  -fx-control-inner-background : #DCDCDC;
  -fx-inner-border     : linear-gradient(to bottom, derive(-fx-color, 90.23825953613186%) 0%,
                                           derive(-fx-color, 17.136566353587632%) 100%);
  -fx-body-color       : linear-gradient(to bottom, derive(-fx-color, 45.81081081081081%) 0%,
                                           derive(-fx-color, -9.603603603603602%) 100%);
}
.button {
  -fx-background-radius : 30, 30, 29, 28;
  -fx-padding           : 7px 14px 7px 14px;
}
.label {
  -fx-padding          : 7px 22px 7px 14px;
}
.label {
  -fx-padding          : 7px 8px 7px 10px;
}
.text-field {
  -fx-padding          : 7px 10px 7px 10px;
}
.label {
  -fx-text-fill        : -fx-text-background-color;
}
.button {
  -fx-background-insets : 0 0 -1 0, 0, 3, 4;
}
.button:focus {
  -fx-background-insets : -1.4, 0, 3, 4;
}
.separator:horizontal .line {
  -fx-border-color      : derive(-fx-background, -80%) transparent transparent transparent;
}
}
```

A simple app



Simple Application

Name

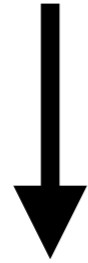
Lastname

Adress

WebView and

WEBENGINE

SCENE



GROUP



NODE



NODE

SCENE



WEBVIEW

WEBENGINE

Web View

- * **EXTENSION OF NODE**
- * **ENCAPSULATES WEBENGINE**
- * **INCORPORATES HTML INTO
THE SCENE**

WebEngine

- * **PROVIDES WEBPAGE FUNCTION**
- * **SUPPORTS USER INTERACTION**
- * **ENABLES DOM ACCESS AND JS**

WebView

```
stage.setTitle("WebView");
```

```
WebView browser = new WebView();  
WebEngine engine = browser.getEngine();  
engine.load("http://harmonic-code.org");
```

```
StackPane pane = new StackPane();  
pane.getChildren().add(browser);  
stage.setScene(new Scene(pane, 980, 720));  
stage.show();
```


Harmonic Code

The life of a geek that loves to code JavaFX, Swing and HTML5...

MONDAY, SEPTEMBER 3, 2012

SteelSeries 3.9.30 released and moved to github

This is just a short info on the SteelSeries Java Swing library:

I moved the SteelSeries Swing library from project Kenai to github to have all projects in one place. Because I was working on it anyway I also created another major release which mainly contains some bugfixes (nothing special). In addition I have added the possibility to switch off the lcd background and the possibility to blink the lcd text (both features have been requested by users).

So if you would like to get the latest source code you should from now on take the code from the [github repo](#) and also issues should be filed on github instead of project Kenai.

Cheers and keep coding...

Eingestellt von Han.Solo um 12:00 AM 4 Kommentare



Recommend this on Google

[Links zu diesem Post](#)

Labels: [steelseries](#), [swing](#)



FOLLOW ME ON [twitter](#)

[Linked in](#) profile



Migrating with

JFXPANEL

How

DOEZIT

work ?

- * *Behaves like JPanel*
- * *Hosts a JavaFx Scene*
- * *Forwards events*
- * *Should be accessed
from the Edt*

Creation

```
// Add a JFXPanel to a Swing JFrame
JFrame frame = new JFrame(„JFXPanel“);
JFXPanel fxPanel = new JFXPanel();
frame.add(fxPanel);
```

```
Platform.runLater(new Runnable() {
    @Override public void run() {
        initFX(fxPanel);
    }
});
```

Initialization

```
// Initialize the JFXPanel
void initFX(JFXPanel fxPanel) {
    // Code to create a JavaFX scene
    ...

    fxPanel.setScene(scene);
}
```

So you could use

JAWAFX

in Swing...

...means also

HTML5

in Swing



But

KEEP

in mind

2

You have

UI-THREADS

It's up to you to

SYNCRONIZE

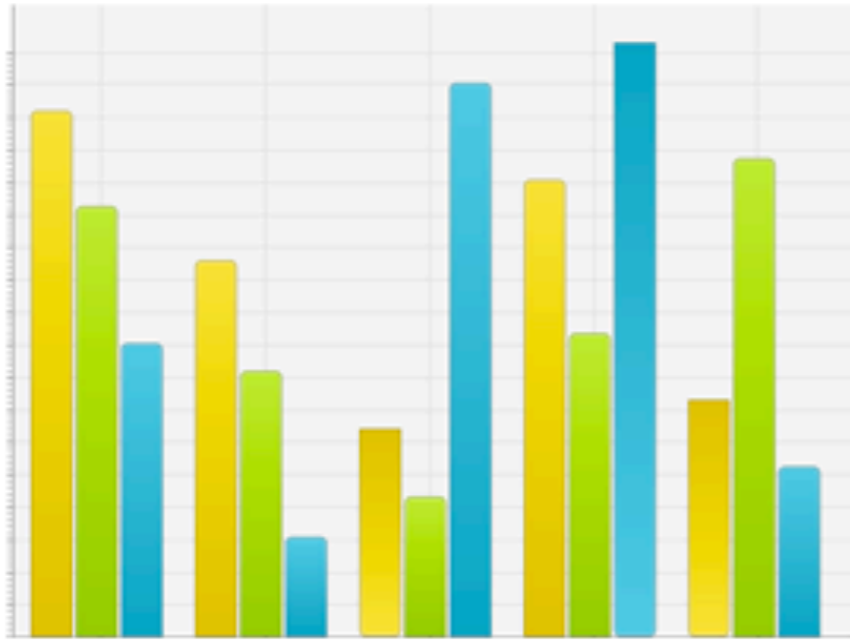
them manually



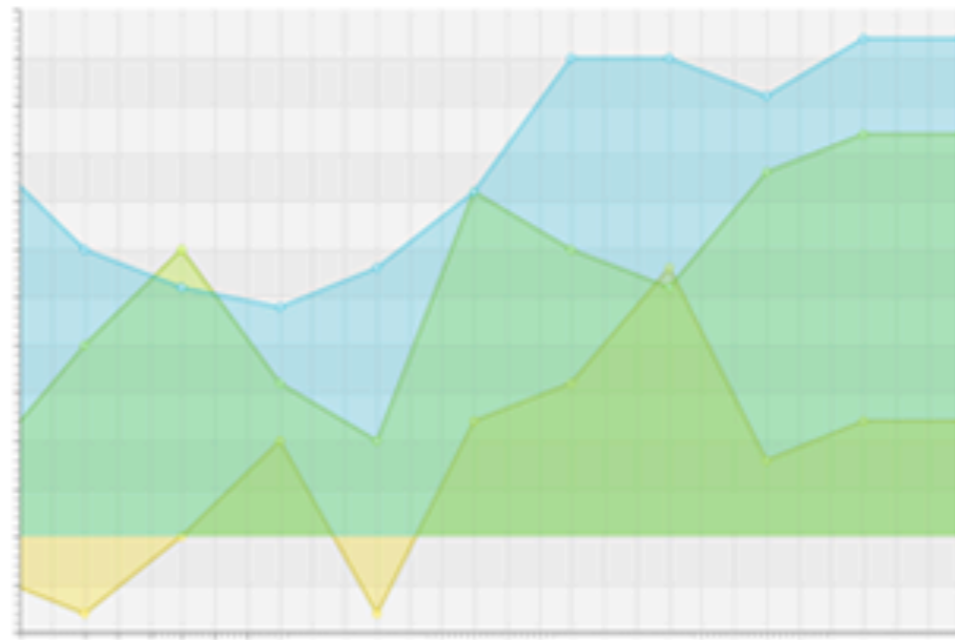
JavaFx

CHARTS

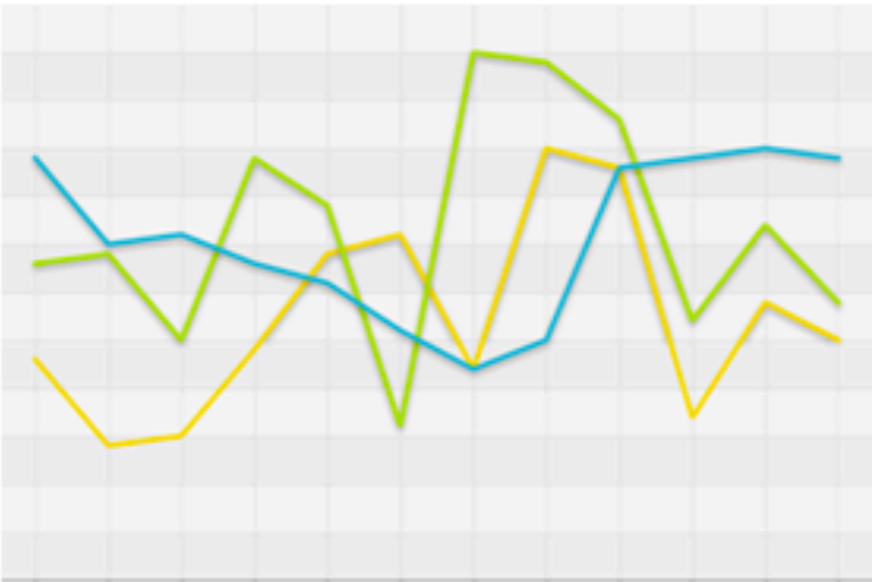
BAR



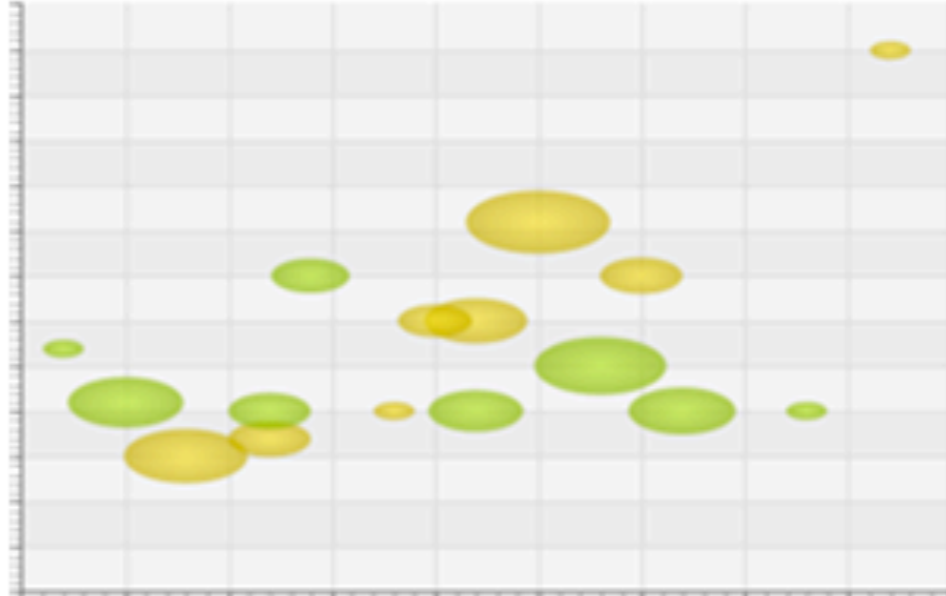
AREA



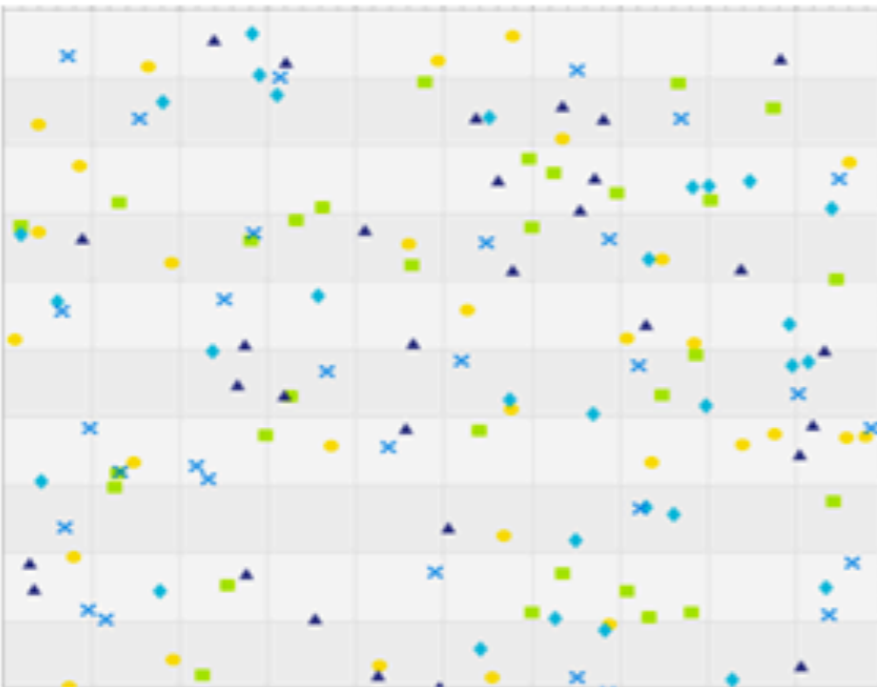
LINE



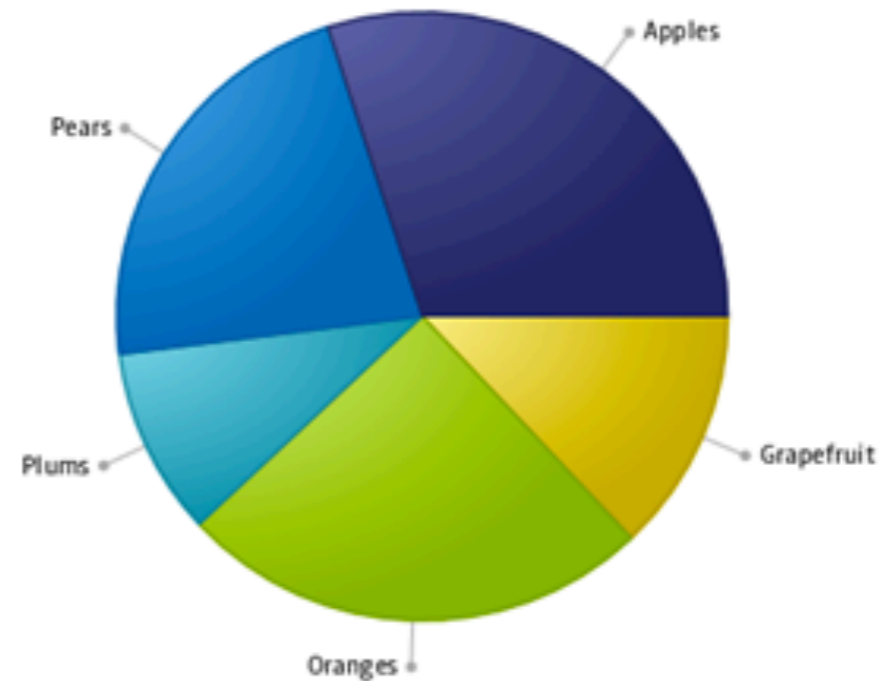
BUBBLE



SCATTER



PIE



JavaFx Charts

- * **CAN BE ANIMATED**
- * **CAN BE STYLED USING CSS**
- * **CAN BE CUSTOMIZED**

Creating a Piechart

```
@Override public void start(Stage stage) {  
    Scene scene = new Scene(new Group(), 500, 500);  
    stage.setTitle("Imported fruits");  
  
    ObservableList<PieChart.Data> pieChartData =  
        FXCollections.observableArrayList(  
            new PieChart.Data("Grapefruit", 13),  
            new PieChart.Data("Oranges", 25),  
            new PieChart.Data("Plums", 10),  
            new PieChart.Data("Pears", 22),  
            new PieChart.Data("Apples", 30));  
    final PieChart chart = new PieChart(pieChartData);  
    chart.setTitle("Imported Fruits");  
  
    ((Group) scene.getRoot()).getChildren().add(chart);  
    stage.setScene(scene);  
    stage.show();  
}
```


Need more

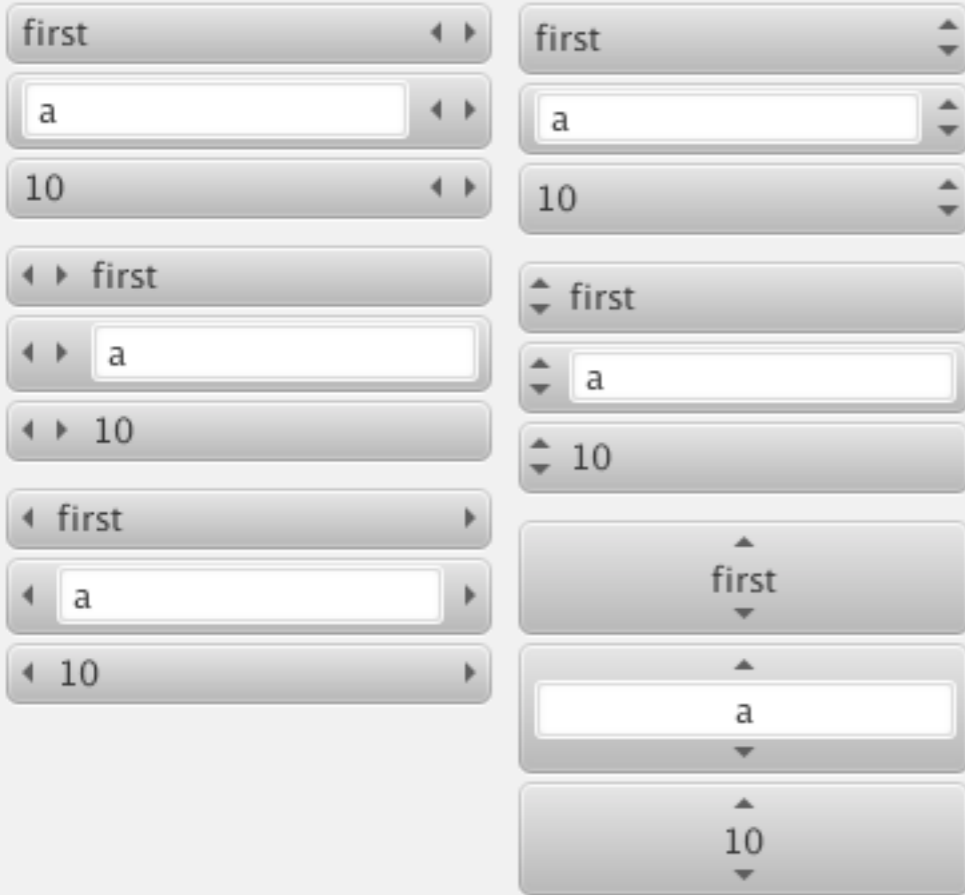
CONTROLS ?


here you go

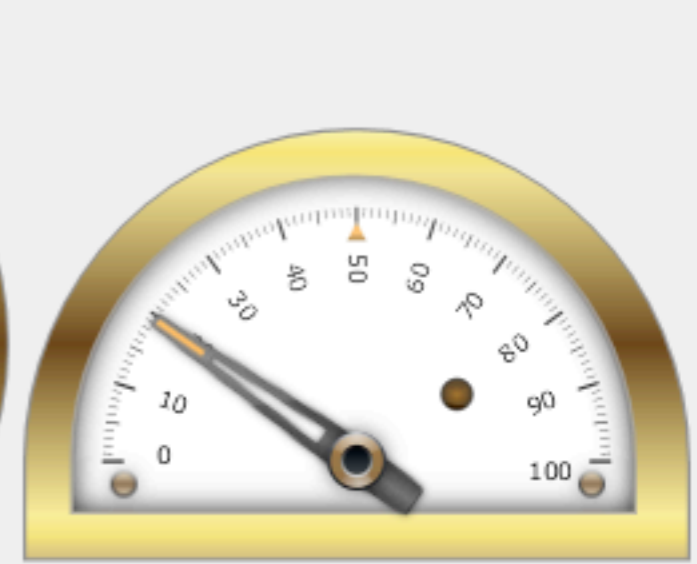
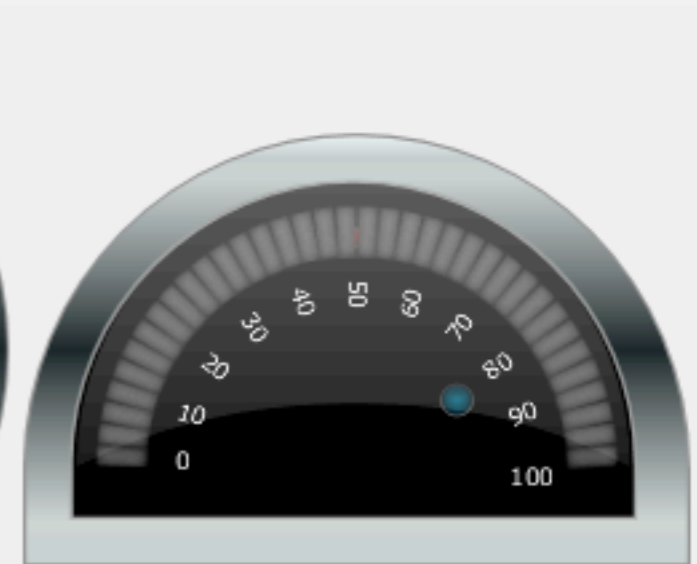
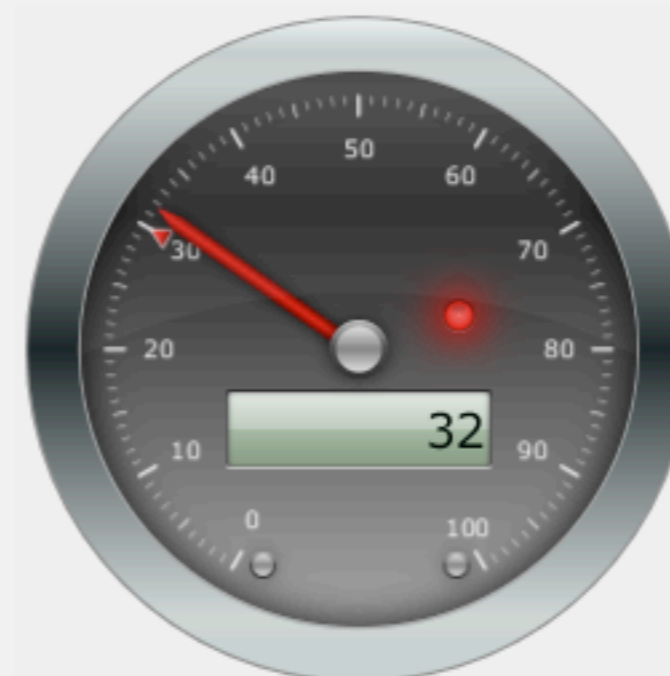
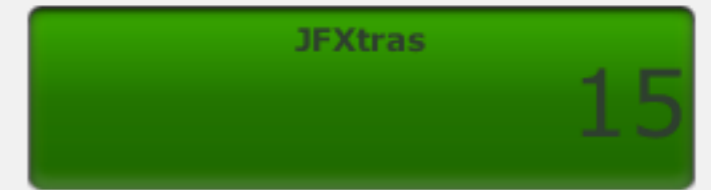
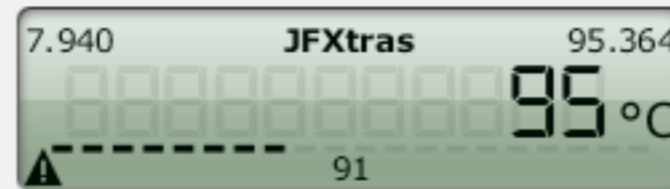
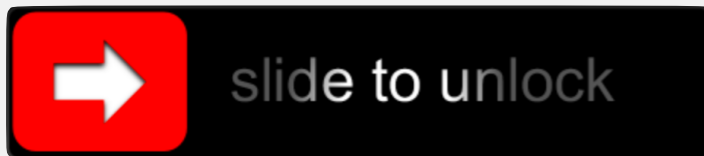
JFXTRAS

Some

EXAMPLES



01.03.2011 





**YOU WANNA BE PART OF
THE PARTY ?**

WE WANT YOU AT
JFEXTRAS



What's new in

JAVA FX 8

JavaFx 8

- * **SUPPORT FOR EMBEDDED**
- * **3D SUPPORT**
- * **SWING-NODE**
- * **PUBLIC API FOR CONTROLS**
- * **PERFORMANCE++**
- * **NO PLUGIN ANYMORE**

Keep coding...

