



JavaFX

Philipp Auinger

Was ist JavaFX?

- Framework zur Erstellung plattformübergreifender Java-Applikationen
- „rich client applications“
- Teil der **Java Platform, Standard Edition** oder kurz **Java SE**
- Ab der für Herbst 2018 geplanten Version 11 ist das für die Entwicklung von Desktopanwendungen und Rich Internet Applications vorgesehene Framework nicht mehr Bestandteil des JDK.



Quelle: <https://www.heise.de/developer/meldung/Oracle-gliedert-JavaFX-aus-dem-JDK-aus-3988711.html>

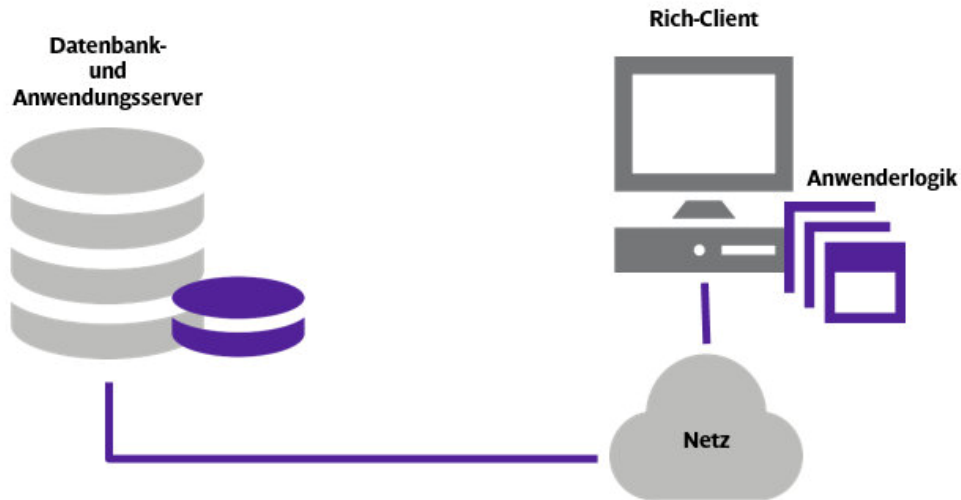
- JavaFX liegt im OpenJFX-Projekt
 - Auf Github verfügbar: <https://github.com/openjdk/jfx>
 - Gluon Website: <https://gluonhq.com/products/javafx/>
 - „Wir wollen es möglich machen das mehr und mehr Entwickler sich an der weiteren Entwicklung von JavaFX beteiligen können“



Quellen: <https://de.wikipedia.org/wiki/JavaFX>, <https://docs.oracle.com/javafx/>, <https://jaxenter.de/openjfx-13-javafx-13-lemmermann-86883>

Rich Client oder Webapplikation

Rich Client (Fat Client)

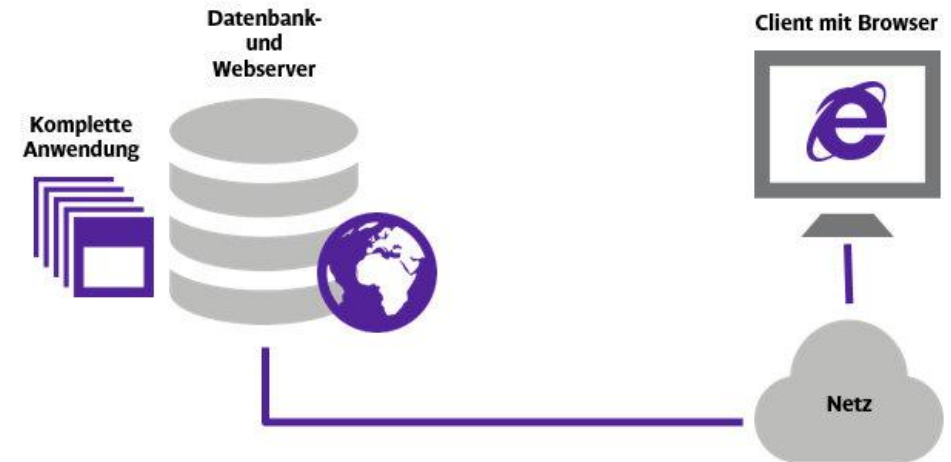


Bestmögliche Performance

Offline - Fähigkeit

Nützliche Komponenten
sind bereits implementiert

Webapplikation



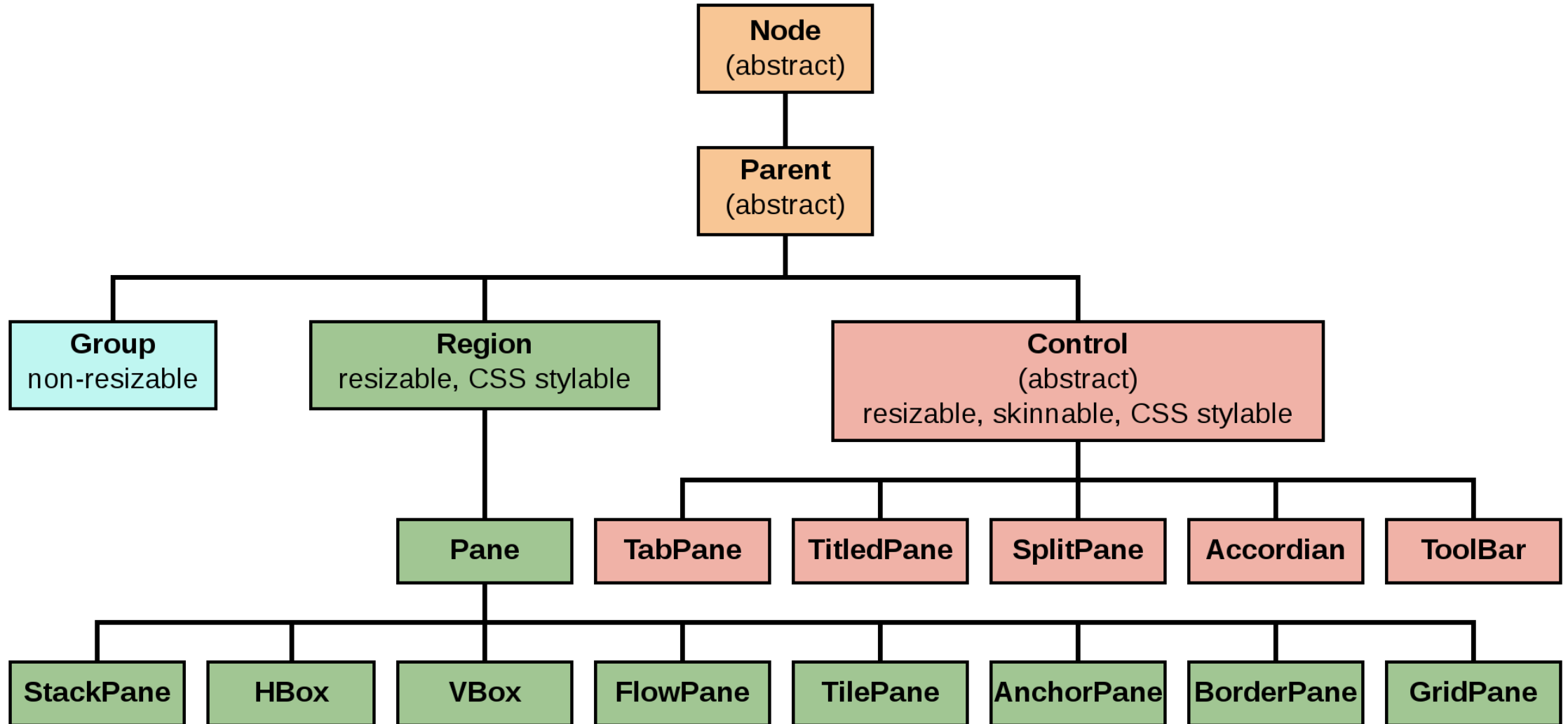
Plattformunabhängig

Geringe Wartungskosten

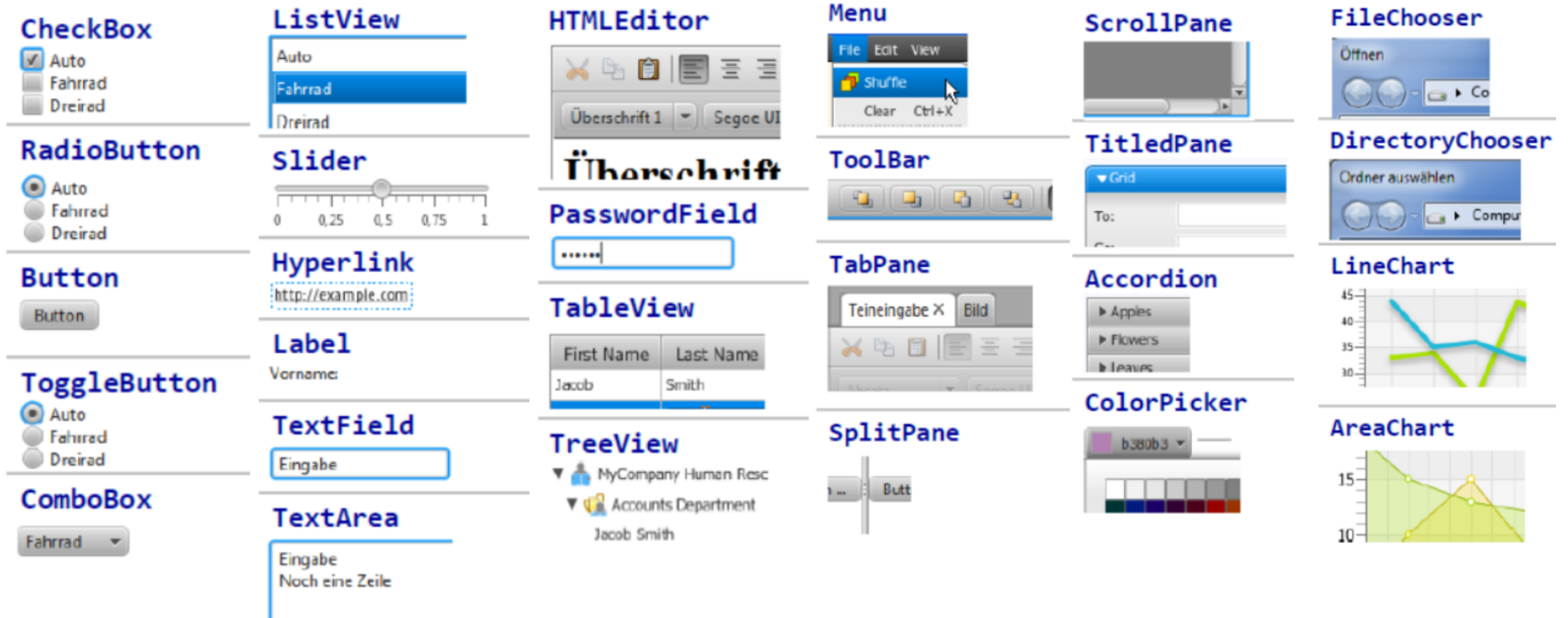
Kosten und Zeitersparnis

JavaFX Architektur

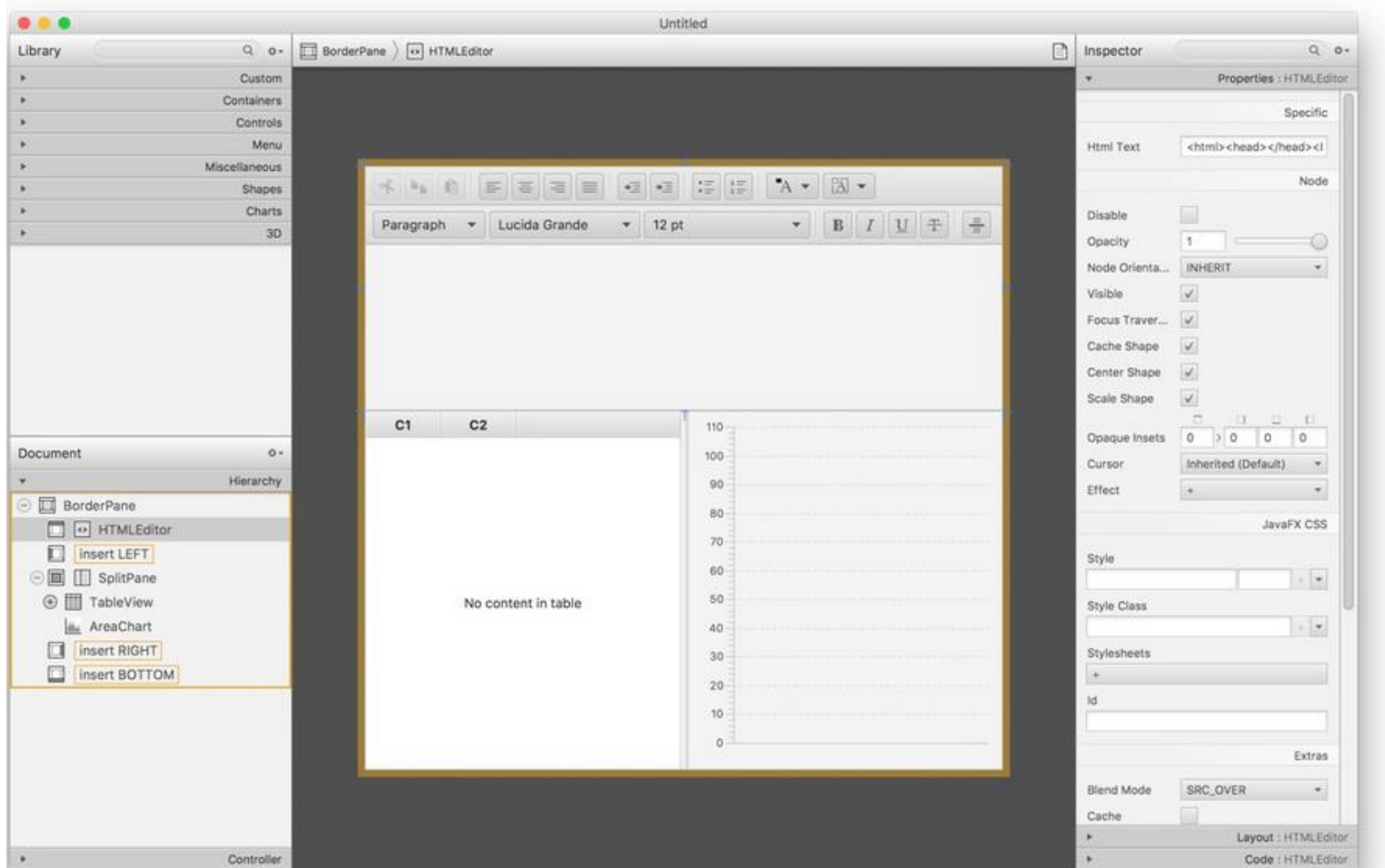
Hierarchie der Container-Elemente



Beispiele für UI-Controls in JavaFX



Scenebuilder

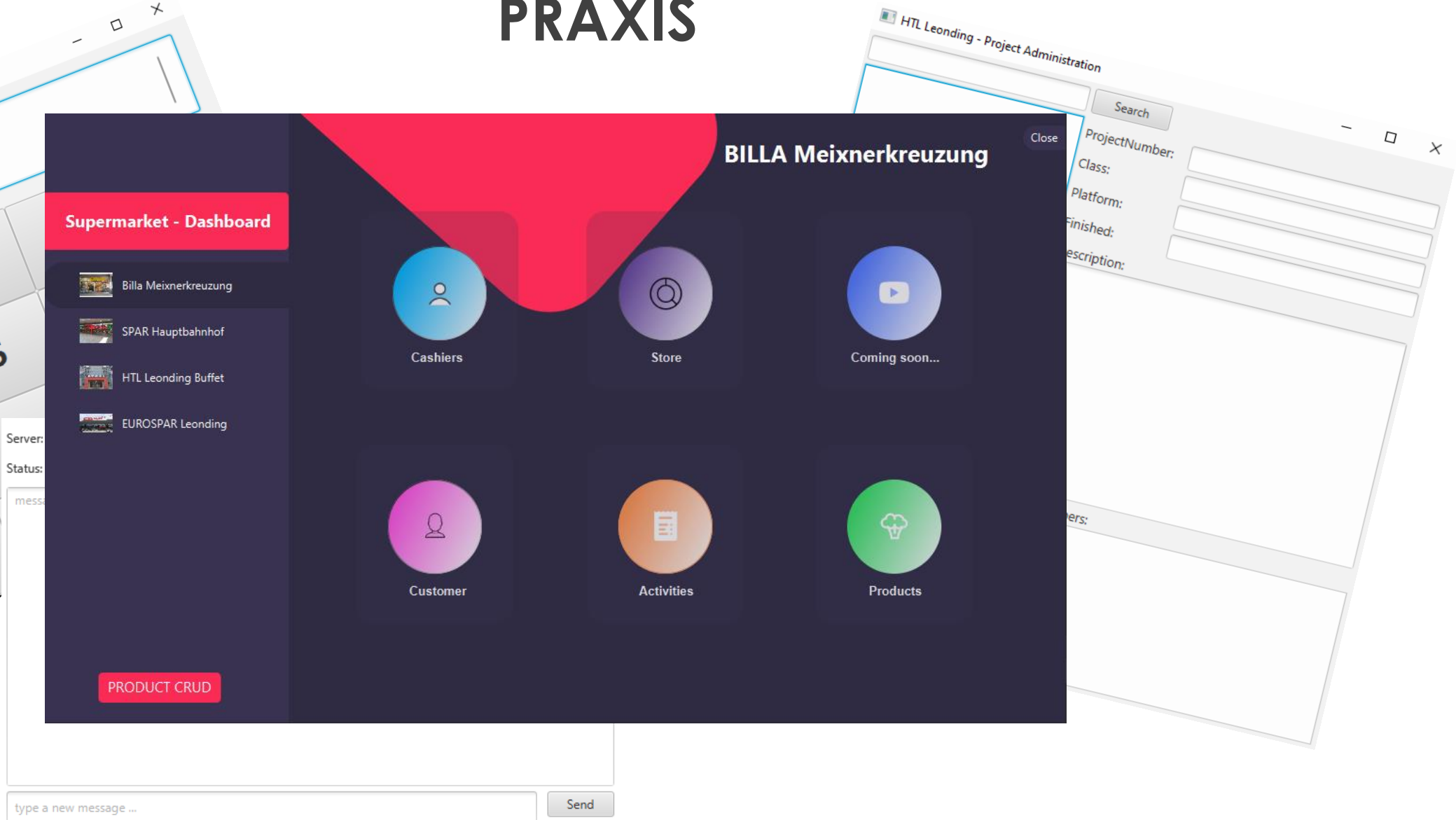


Drag & Drop,
Rapid Application
Development.

Download Now

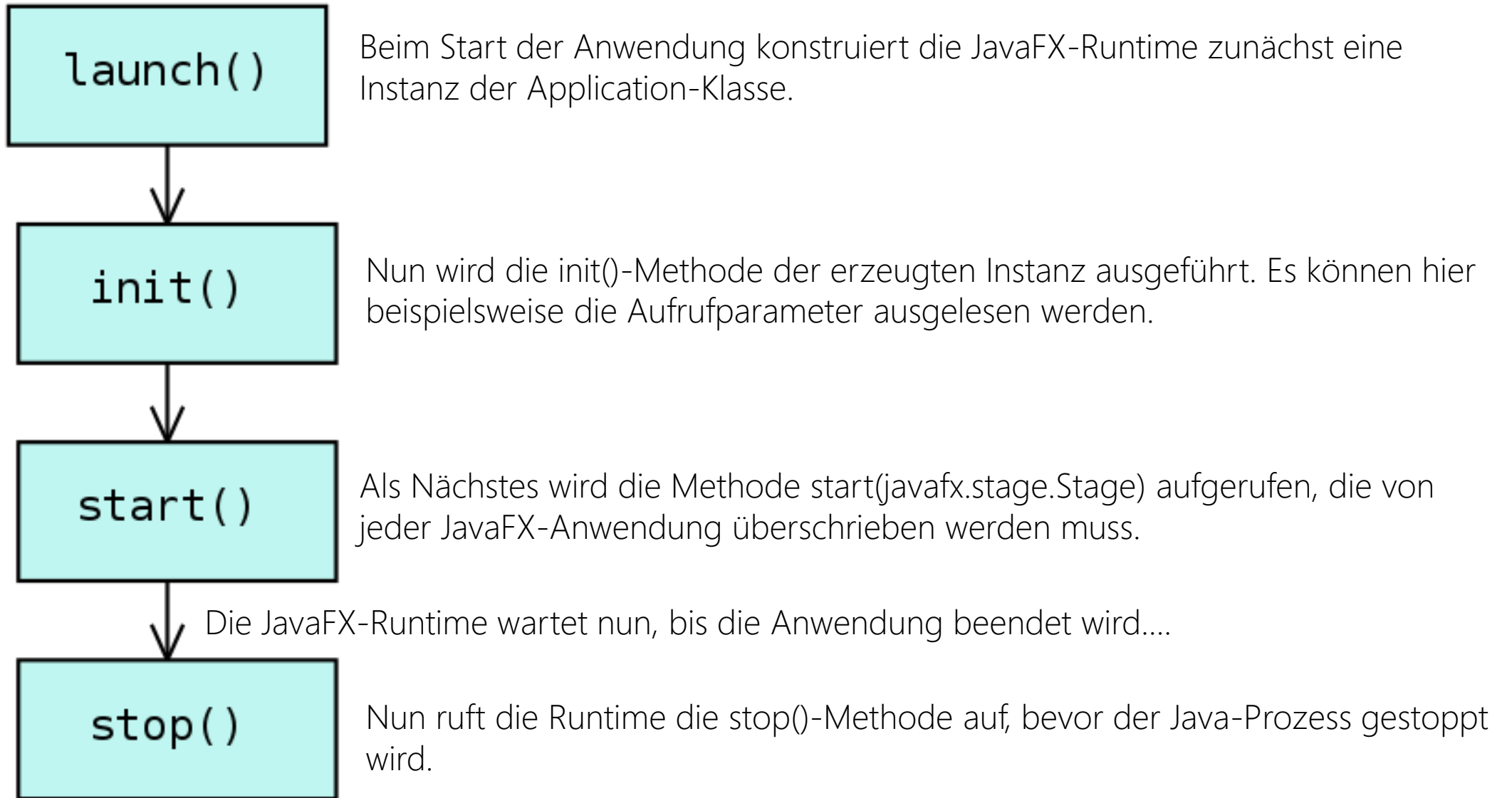
Quelle: <https://gluonhq.com/products/scene-builder/>

JavaFX PRAXIS

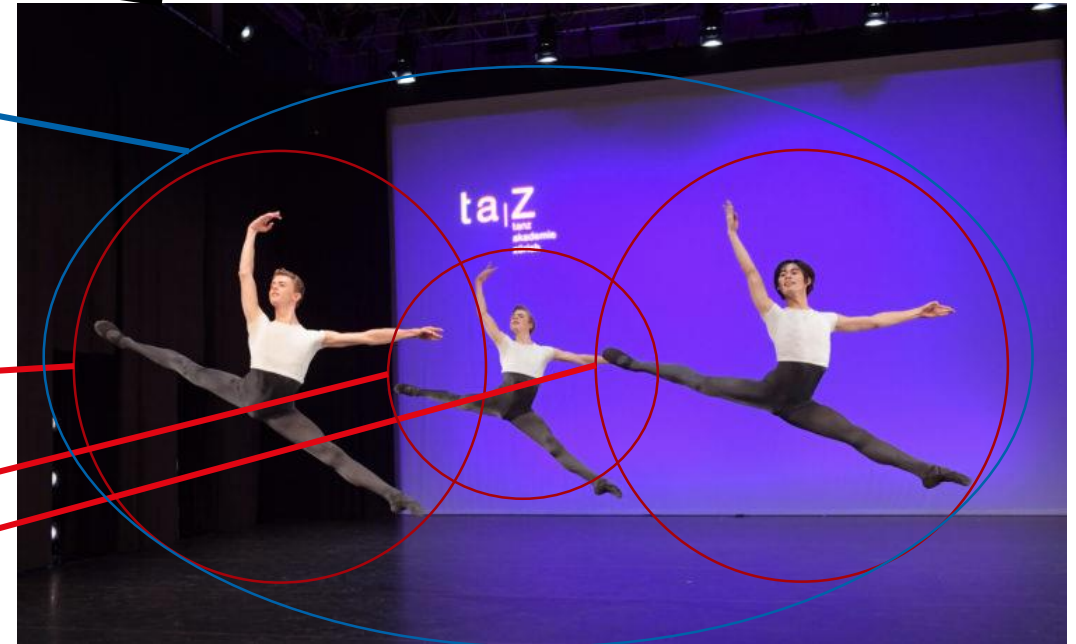
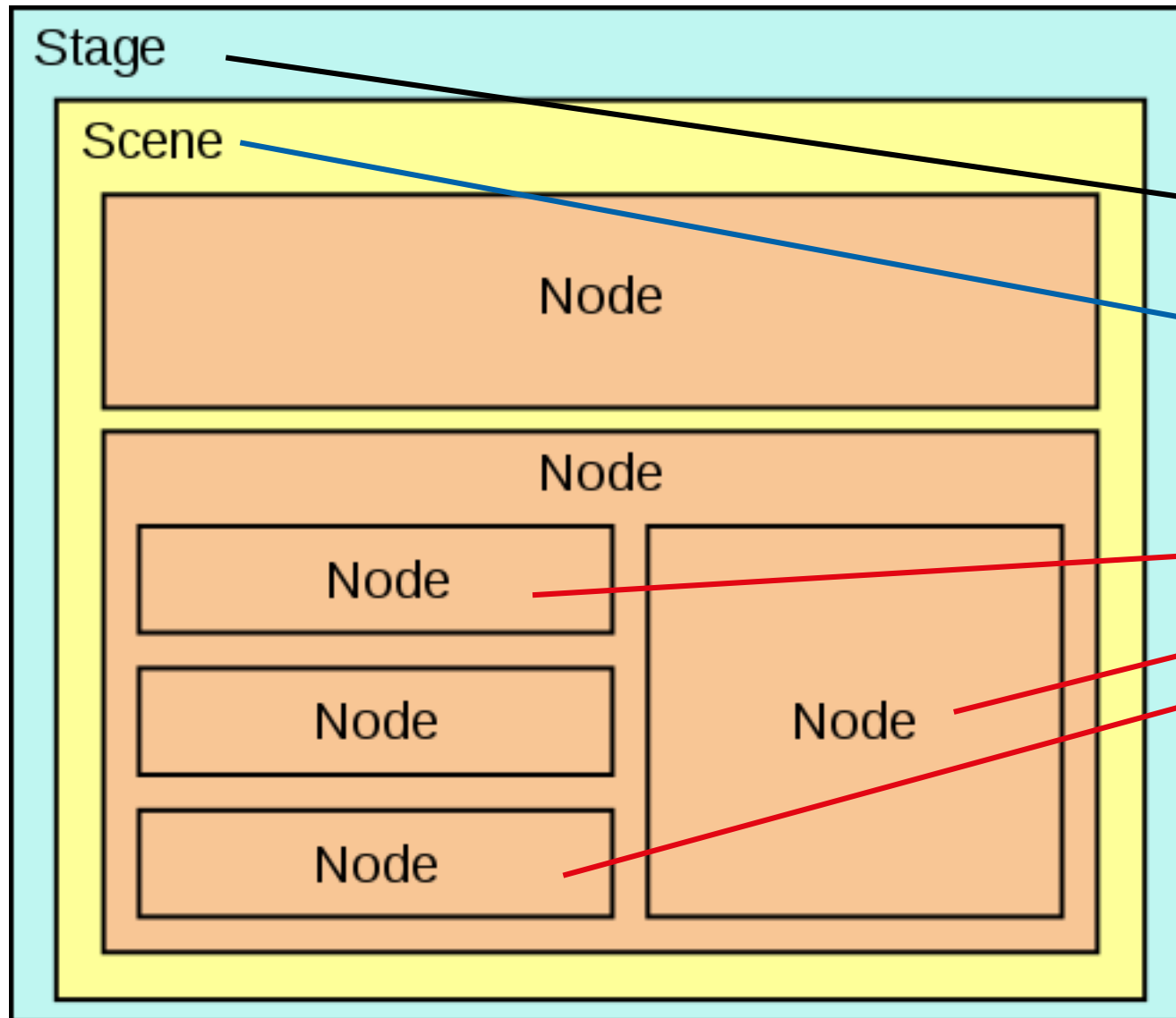


Quelle: <https://www.youtube.com/watch?v=AoKoCv6VtR0>,
<https://github.com/k33ptoo/javafx-sample-dashboard>

• JavaFX Lebenszyklus •



JavaFX Architektur



Bühne – Aufführung - Tänzer

JavaFX MVC

View

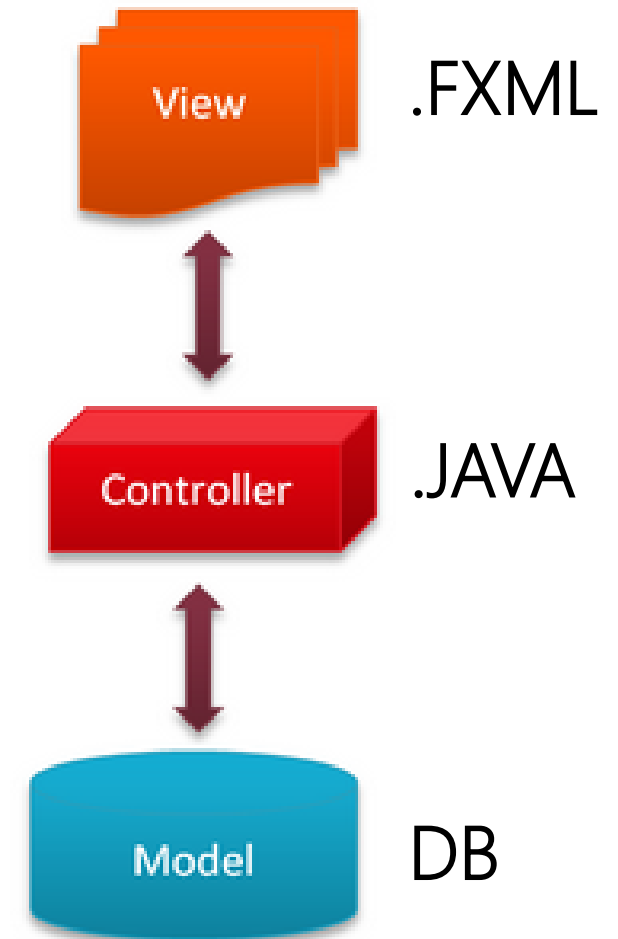
- ist für die Präsentation von Daten zuständig
- initialisiert, organisiert und speichert alle Oberflächen Elemente
- ist für Style und Layout zuständig

Controller

- verbindet View und Model
- enthält Steuerungslogik
- delegierte Geschäftslogik, enthält sie aber nicht!

Model

- speichert Daten
- besitzt nur Getter- und Setter-Funktionen



JavaFX Events

onAction im FXML setzen
Funktion im Controller
implementieren

```
<Button onAction="#onCloseBtn"/>
```

```
@FXML
private void onCloseBtn(ActionEvent event){
    Node source = (Node) event.getSource();
    Stage stage = (Stage)
source.getScene().getWindow();
    stage.close();
}
```

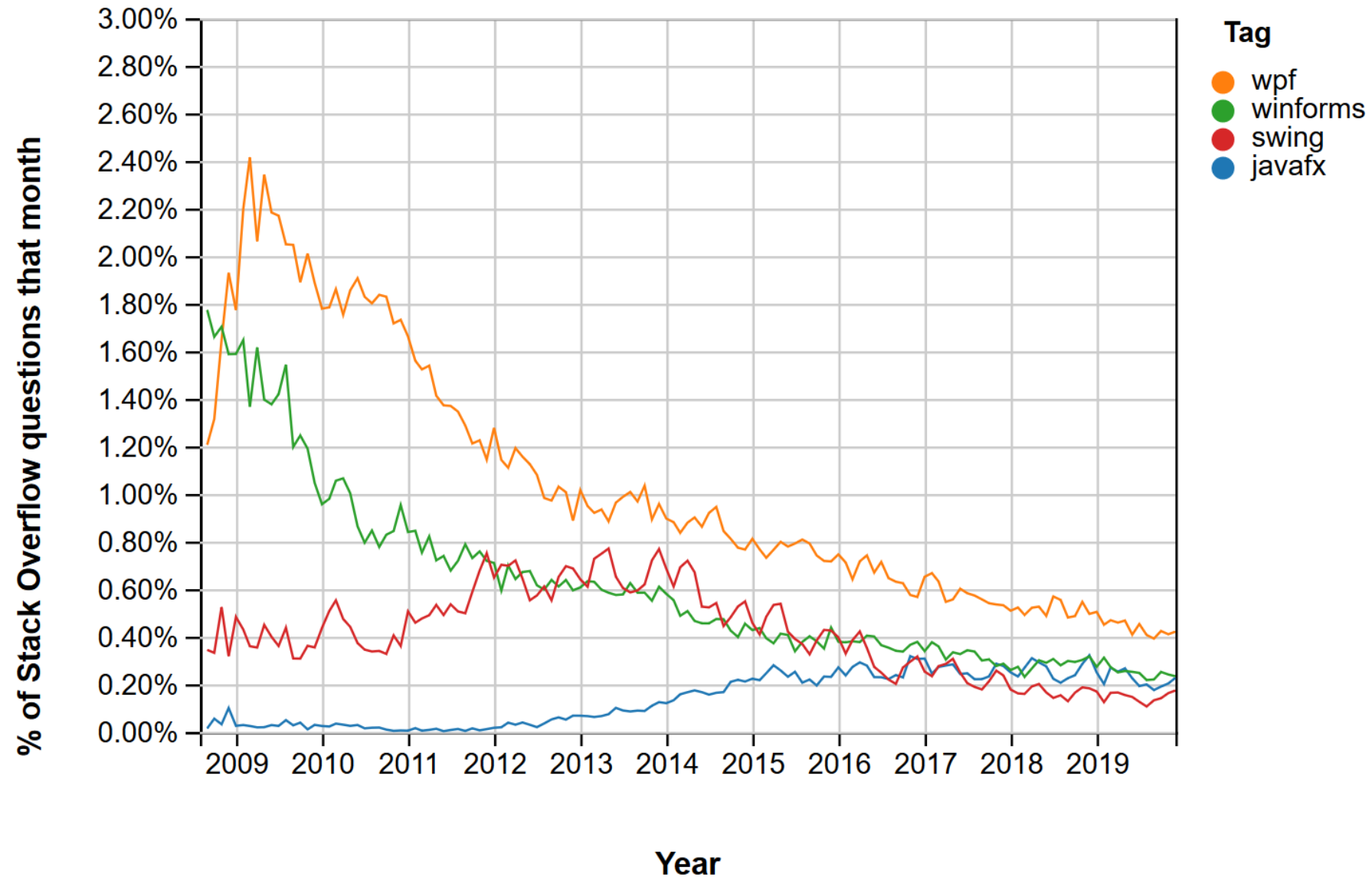
fx:id im FXML setzen
Dann mit Controller
darauf zugreifen

```
<TextField fx:id="id" />
```

```
@FXML
private TextField id;
```

```
id.textProperty().addListener((obs, oldText, newText) -> {
    // event Listener with regex
    if (!newText.matches("\\d*")) {
        id.setText(newText.replaceAll("[^\\d]", ""));
    }
});
```

Statistik



Quelle: <https://insights.stackoverflow.com/trends?tags=java%2Cwpf%2Cwinforms%2Cswing>

Kritik

Der ehemalige Sun-Mitarbeiter Shai Almog bescheinigt JavaFX eine wenig rosige Zukunft.

- Er teilt JavaFX-Nutzer in drei Kategorien:
 - Unternehmen mit großen Investitionen in Swing
 - Studenten
 - eingefleischte Fans.

Der Jobmarkt sehe düster aus: Auf der Karriereseite *dice.com* findet man nur 28 Stellen, die JavaFX-Kenntnisse verlangten, im Gegensatz zu 198 Stellenangebote für Swing, 2333 für Android und 16752 für Java EE (Stand 22. November 2015). Quelle: <https://de.wikipedia.org/wiki/JavaFX#Kritik>

dice.com 2020 (US)

JavaFX: 3

Swing: 24

Android: 750

Java EE: 25

indeed.com 2020 (Worldwide)

JavaFX: 58

Swing: 9773

Android: 15996

Java EE: 1391

karriere.at 2020 (AT)

JavaFX: 1

Swing: 14

Android: 111

Java EE: 72



**Vielen Dank für
Ihre Aufmerksamkeit**

