



**Projektarbeit**

# **Benutzeroberfläche für Spracherkennung mit JavaFX**

**Von Luis San Martin Martinez**

**aicas GmbH -2021**

# Inhalt

- Aktuelles Szenario
- Werkzeuge
  - CMUSphinx
  - JavaFX
- Entwicklung
  - Phasen
  - Demos
- Fazit

# Aktuelles Szenario

- **Beschreibung**
  - Erstellung einer grafischen Oberfläche
- **Anwendungsfälle und Demos**
  - Spracherkennungssteuerung für Automobile und die 4 Demos
- **Umfang und Ziel**
  - Integration und Steuerung von Multiprozessen durch Sprachbefehle.







# Werkzeuge

- **CMUSphinx**

- Was ist das und wofür wird es verwendet?
- Struktur
- Code

- **JavaFX**

- Was ist das und wofür wird es verwendet?
- Struktur
- Code





Powered By  
CMUSphinx

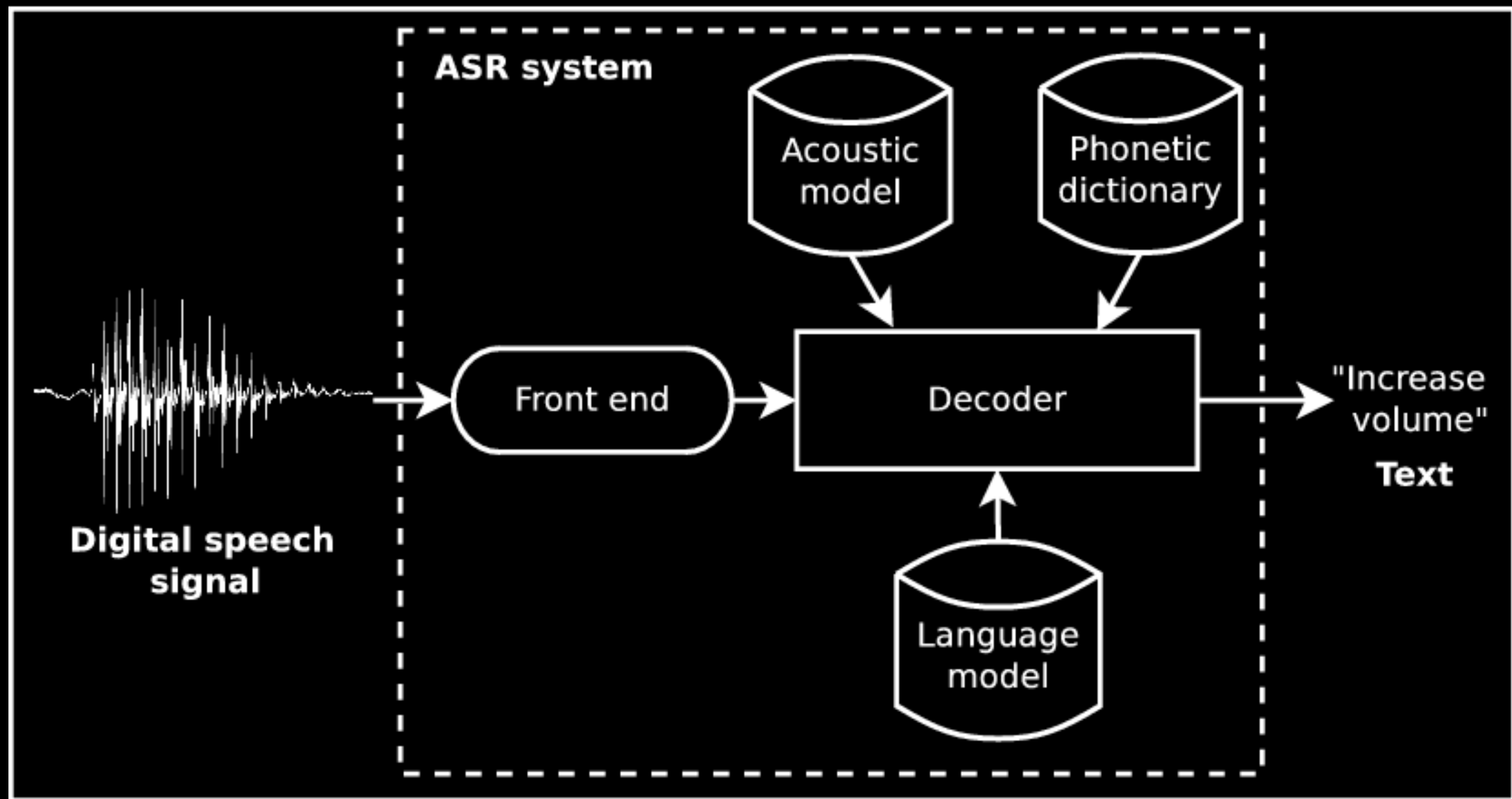




# Was ist CMUSphinx 4 ?

Es ist eine externe OpenSource-Bibliothek für die Spracherkennung, die von der CMU erstellt wurde

- Spracherkennung (Demos)
- Transkription von Audio in Text
- Modelltraining



# Digital Speech Signal

```
LiveSpeechRecognizer recognizer = new LiveSpeechRecognizer(configuration);
```

```
    recognizer.startRecognition(true);  
    SpeechResult result = recognizer.getResult();  
    startRecognition(false).  
    recognizer.stopRecognition();
```

- LiveSpeechRecognizer :Voice Signal RealTime
- StreamSpeechRecognizer: Voice Signal Local



# Automatic Speech Recognition

*Configuration configuration = new Configuration();*

- Acoustic model
- Dictionary
- Grammar/Language model

# Text Output

```
//Creating SpeechResult object  
SpeechResult result;
```

```
//Check if recognizer recognized the speech  
while ((result = recognizer.getResult()) != null) {
```

```
    //Get the recognized speech  
    String voiceCommand = result.getHypothesis();
```

```
    System.out.println(result.getHypothesis());
```







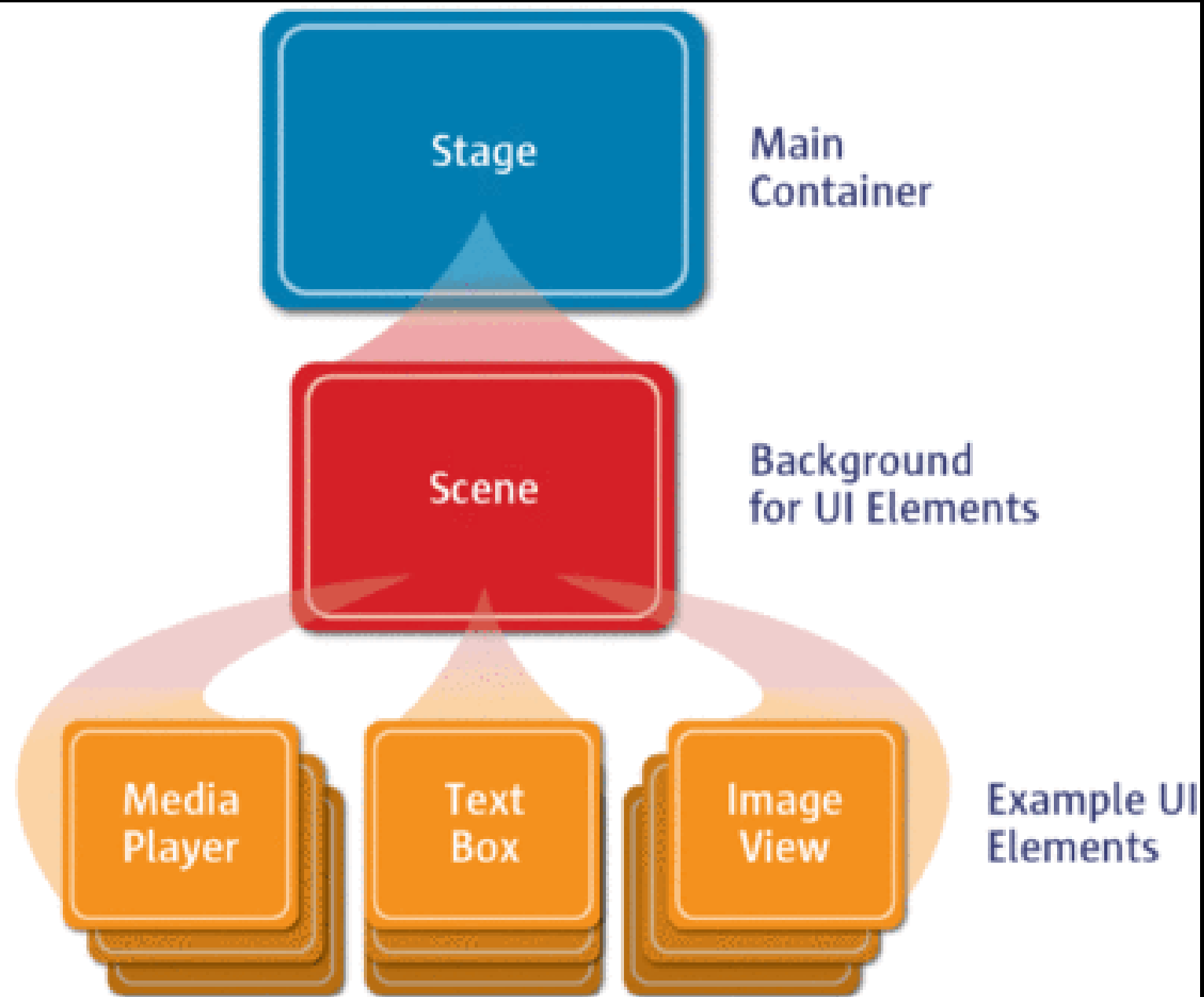


# Was ist JavaFX ?

JavaFX ist eine Softwaretechnologie, die es in Kombination mit Java ermöglicht, hochmoderne Anwendungen mit fortschrittlichen Inhalten, Audio und Video zu erstellen und bereitzustellen.

- Multiplattform-Entwicklung
- Integration mit Java
- CSS-Code-Integration





# Stage

*@Override*

*public void start(Stage primaryStage) throws  
IOException, Exception {*

*primaryStage\_ = primaryStage;*

*//Start Container*

*primaryStage.setTitle("Start the App")*



# Scene

```
VBox layoutGui = new VBox(20);  
    layoutGui.getChildren().addAll(labelGui, btnTest);  
    layoutGui.setAlignment(Pos.BASELINE_CENTER);  
    sceneStart = new Scene(layoutGui, 300, 300);
```

# Button

```
btnTest = new Button("Demos");
```

```
btnTest.setOnAction(e -> test(primaryStage));
```

# Anwendung

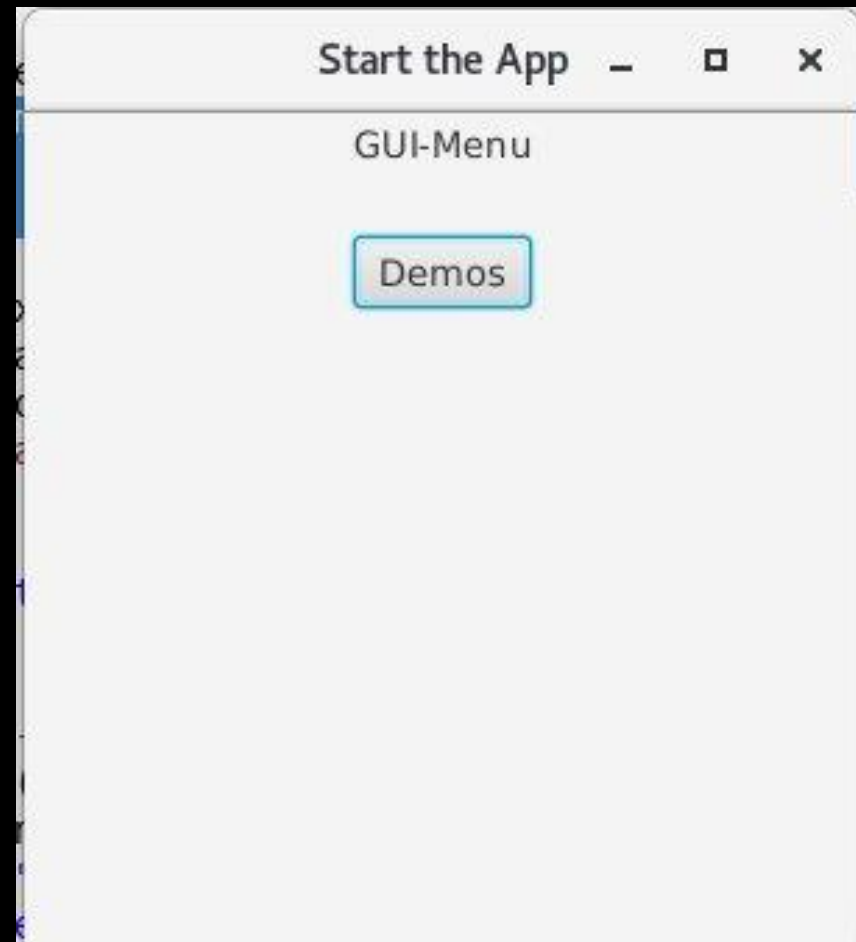
*Der Entwicklungsprozess dieses Projekts gliedert sich in drei Phasen*

- Vorbereitung und Konfiguration
- Implementierung CMUSphinx (Demos)
- GUI-Erstellung mit JavaFX (mit Thread und Platform.runLater())



# StartMenu

funktioniert mit dem Wakeup-Befehl  
Active=false



# DemosMenu

Active=false



Start the App — □ ×

Main Menu

Transcriber

Assistant

SpeakerId

Aligner

Back

# IndividuellMenu

*Active=true*

Voice Recognition - □ ×

AssistantButton

AssistantDemo

Back



# Anwendungen in der Zukunft

- Automotive Anwendung mit Voice Controlling und CrossOver mit OpenCV als Upgrade-Option
- Intelligenter Sprachassistent in Rasp-PI mit Aufgabenautomatisierungsfunktionen
- Training CMUSphinx mit Machine und DeepLearning

## Fazit

- Erfahrung im Umgang mit dieser Technologie
- Das Hauptziel wurde erreicht, nämlich eine GUI zu erstellen
- Auch das sekundäre Ziel wurde erreicht, nämlich die Steuerung der GUI





**Vielen Dank für Ihre  
Aufmerksamkeit und Ihr  
Feedback!!!**