

# .NET Speech API



Seminar *Objektorientiertes Programmieren mit .NET und C#*

Alexander Schubert

Institut für Informatik

Software & Systems Engineering



24.01.2013

# Agenda

- Überblick über die Microsoft Speech APIs
- Microsoft Speech API: Voraussetzungen, Zugriff und Namespaces
- Text-zu-Sprache
  - Codebeispiel
  - Speech Synthesis Markup Language (SSML) 1.0
  - Weitere Funktionen
  - Asynchrone Sprachausgabe & Events
- Spracherkennung
  - Modi der Spracherkennung
  - Grammatiken
  - Semantische Werte
  - Speech Recognition Grammar Specification 1.0 (SRGS)
- Demo

# Überblick über die Microsoft Speech APIs

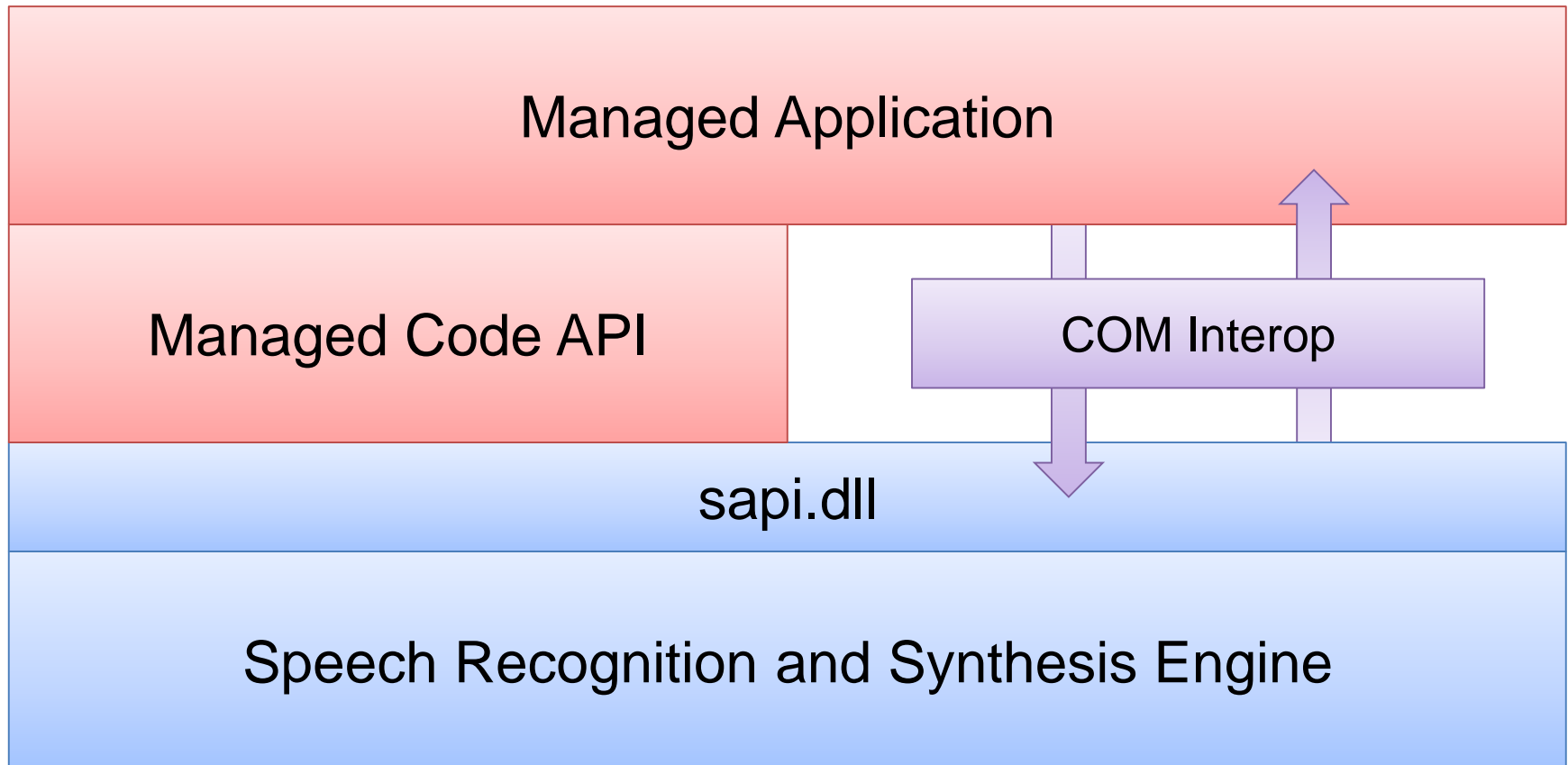
- **Microsoft Speech API (SAPI):** System.Speech.\*
  - Desktop
- **Windows Phone API:** Windows.Phone.Speech.\*
  - Windows Phone 8
- **Microsoft Speech Platform:** Microsoft.Speech.\*
  - Client-Server
  - Kinect
  - ...



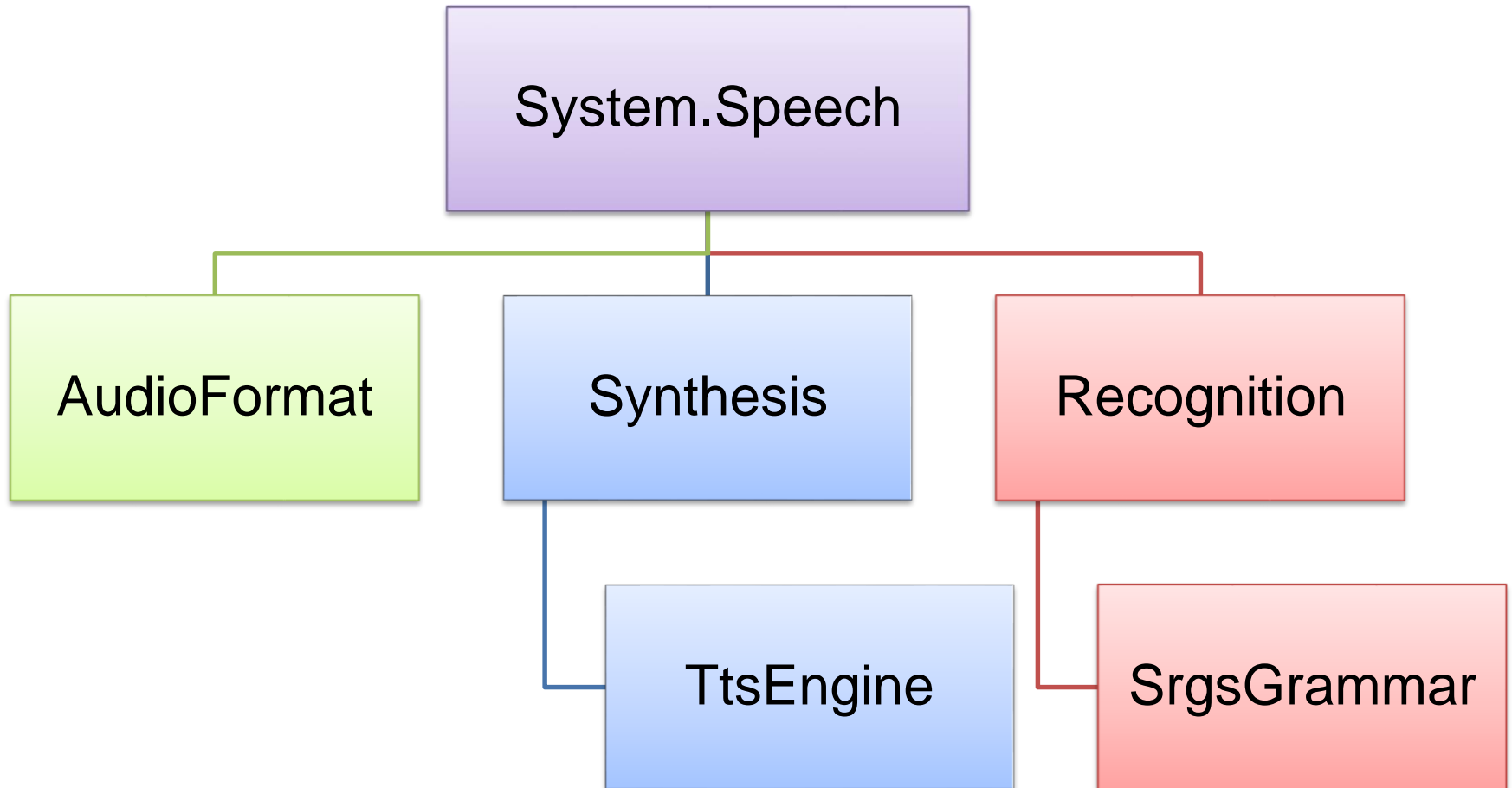
## Voraussetzungen für die Microsoft Speech API (SAPI)

- mindestens **.NET Framework 3.0** (Windows XP, besser Vista)
- **Sprachausgabe**: Installierte **SAPI 5** kompatible **Stimmen/Engine**
  - **kostenfreie** Stimmen aus dem Internet → **geringe Qualität**
  - **engl. MS Stimmen** in Windows **integriert** → **mittlere Qualität**
  - verschiedene **kostenpflichtige** Stimmen → **gute Qualität**
  - ⇒ Qualität der Sprachausgabe abhängig von der Engine
- **Spracherkennung**: Engine aus **Windows Sprachpaketen**

## Zugriff auf die Speech API aus .NET



# Speech API Namespaces



# Text-zu-Sprache



## Text-zu-Sprache – Codebeispiel

```
using System.Speech.Synthesis;
// ...
SpeechSynthesizer tts = new SpeechSynthesizer();

PromptBuilder prompt = new PromptBuilder(new CultureInfo("en-US"));
prompt.AppendText("It is now");
prompt.AppendTextWithHint(DateTime.Now.ToLongTimeString(), SayAs.Time);

tts.SpeakAsync(prompt);
```

// Möglichkeiten für SayAs: Buchstabieren, Zahlen, Uhrzeit, Datum, Telefonnummern



# Speech Synthesis Markup Language (SSML) 1.0

- World Wide Web Consortium (W3C) Recommendation (2004)

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<speak version="1.0"
```

```
  xmlns="http://www.w3.org/2001/10/synthesis"
```

```
  xml:lang="en-US">
```

```
  It is now <say-as interpret-as="time">11:05</say-as>.
```

```
</speak>
```

## Text-zu-Sprache – Weitere Funktionen

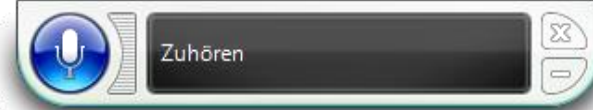
- Auswahl von **Stimmen** nach *Sprache, Geschlecht, Alter* und *Name*
  - `tts.SelectVoiceByHints(...)`
  - `prompt.StartVoice(...); ...; prompt.EndVoice(...)`
- Variation von Sprache: **Betonungen, Geschwindigkeit, Lautstärke**
  - `tts.Rate, tts.Volume`
  - `prompt.StartStyle(...); ...; prompt.EndStyle(...)`
- **Pausen** im Text
  - `prompt.AppendBreak(...)`
- Angabe von **phonetischer Aussprache**
  - `tts.AddLexicon(...)`
  - `prompt.AppendTextWithPronunciation(...)`
- Einbinden von **Audiodateien**
  - `prompt.AppendAudio(...)`

## Text-zu-Sprache – Asynchrone Sprachausgabe & Events

Auslöser	Event
<b>Start</b> und <b>Ende</b> eines Prompt	SpeakStarted, SpeakCompleted
<b>Status:</b> Ready, Speaking, Paused	StateChanged
<b>Markierung</b>	BookmarkReached
<b>Wort</b>	SpeakProgress
<b>Laut</b>	PhonemeReached
<b>Mundbild</b>	VisemeReached

# Spracherkennung





## Spracherkennung – Eingebaute Fähigkeiten



Die **Windows-Spracherkennung** übernimmt...

- Interaktion mit **Windows Forms** oder **WPF Controls** (Button, Checkbox, ...)
- **Diktat** von Text

Klick mich

Klick mich

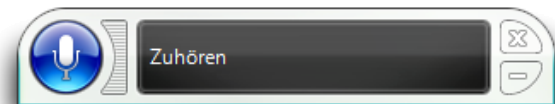
Der **Programmierer** schreibt Code für...

- Interaktion über **weitere Sprachbefehle**
- **speziellere Anwendungen** der Spracherkennung

# Modi der Spracherkennung

## gemeinsam genutzt: SpeechRecognizer

- globale Konfiguration (Systemsteuerung)
- globales Ein-/Ausschalten
- Integration in den Benutzungsablauf



## prozessintern: SpeechRecognitionEngine

- ohne Spracherkennungsleiste
- Auswahl von Audioquellen
- programmgesteuertes Ein-/Ausschalten

# Grammatiken für die Spracherkennung

- **Diktatgrammatik:** einfache Grammatik für beliebige Worte
  - DictationGrammar
  - DictationGrammar("grammar:dictation#spelling")
- Erzeugung zur **Laufzeit**
  - GrammarBuilder, Choices
  - SrgsDocument
- **XML:** W3C Speech Recognition Grammar Specification 1.0 (SRGS)

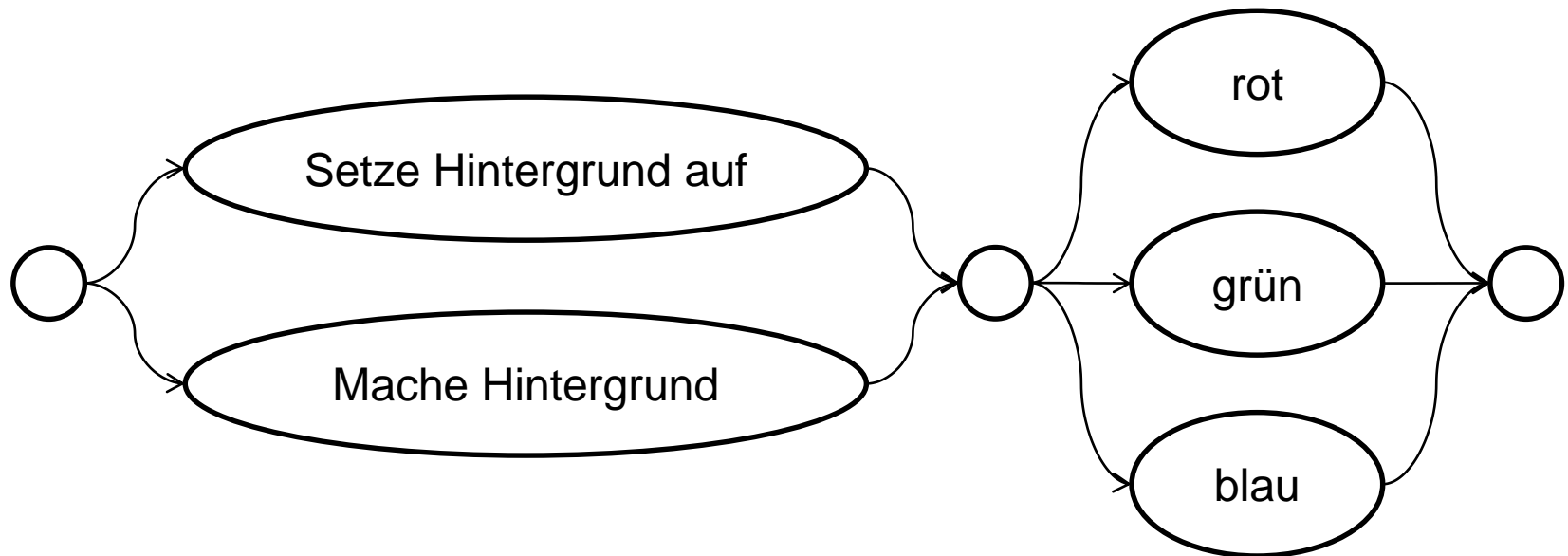
## Spracherkennung – Ein Codebeispiel

```
using System.Speech.Recognition;
// ...
var recognizer = new SpeechRecognitionEngine();
recognizer.SetInputToDefaultAudioDevice();
recognizer.LoadGrammar(new DictationGrammar());

recognizer.SpeechRecognized +=
    (sender, e) => { Console.WriteLine(e.Result.Text); };
recognizer.RecognizeAsync(RecognizeMode.Multiple);
```



# Eine einfache Grammatik



# Eine einfache Grammatik

```
GrammarBuilder gb = new GrammarBuilder();
```

```
Choices back = new Choices();
```

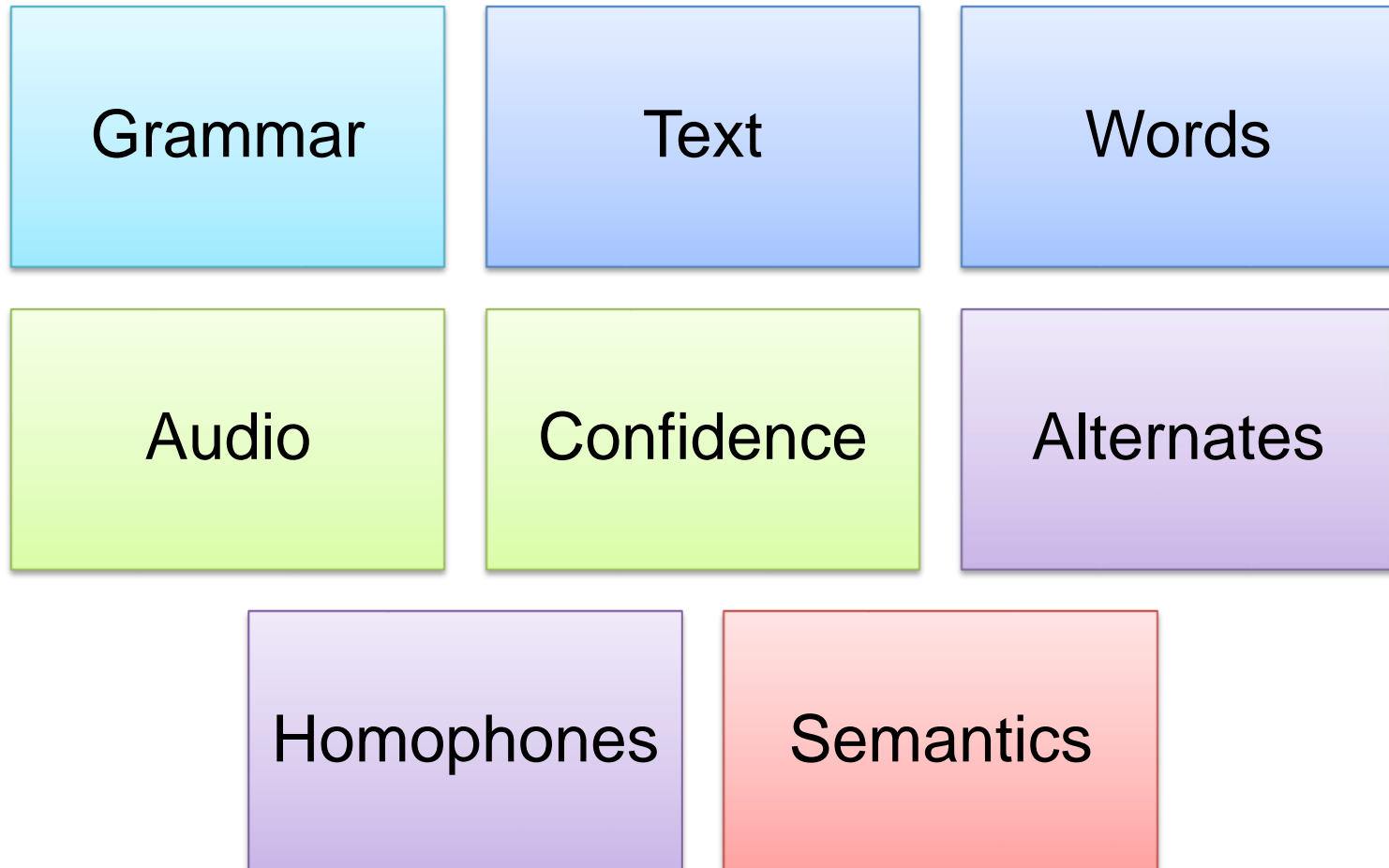
```
back.Add("Setze Hintergrund auf"); back.Add("Mache Hintergrund");  
gb.Append(back);
```

```
Choices colors = new Choices();
```

```
colors.Add("rot");  
colors.Add("grün");  
colors.Add("blau");  
gb.Append(colors);
```

```
Grammar grammar = new Grammar(gb);
```

# Ausgabe der Spracherkennung: `RecognitionResult`



# Eine einfache Grammatik

```
GrammarBuilder gb = new GrammarBuilder();
```

```
Choices back = new Choices();
```

```
back.Add("Setze Hintergrund auf"); back.Add("Mache Hintergrund");  
gb.Append(back);
```

```
Choices colors = new Choices();
```

```
colors.Add("rot");  
colors.Add("grün");  
colors.Add("blau");  
gb.Append(colors);
```

```
Grammar grammar = new Grammar(gb);
```

# Eine einfache Grammatik mit semantischen Werten

```
GrammarBuilder gb = new GrammarBuilder();
```

```
Choices back = new Choices();
```

```
back.Add("Setze Hintergrund auf"); back.Add("Mache Hintergrund");  
gb.Append(back);
```

```
Choices colors = new Choices();
```

```
colors.Add(new SemanticResultValue("rot", 1));  
colors.Add(new SemanticResultValue("grün", 2));  
colors.Add(new SemanticResultValue("blau", 3));  
gb.Append(new SemanticResultKey("color", colors));
```

```
Grammar grammar = new Grammar(gb);
```

```
// Zugriff mit e.Result.Semantics["color"].Value  
// gibt 1, 2 oder 3
```

# Speech Recognition Grammar Specification 1.0 (SRGS)

```
<?xml version="1.0" encoding="utf-8" ?>
<grammar version="1.0" xml:lang="de-DE" root="root" tag-format="semantics/1.0"
  xmlns="http://www.w3.org/2001/06/grammar">
  <rule id="root">
    <one-of>
      <item> Setze Hintergrund auf </item>
      <item> Mache Hintergrund </item>
    </one-of>
    <ruleref uri="#color" />
  </rule>
  <rule id="color">
    <one-of>
      <item> rot <tag> out.color = "1"; </tag> </item>
      <item> grün <tag> out.color = "2"; </tag> </item>
      <item> blau <tag> out.color = "3"; </tag> </item>
    </one-of>
  </rule>
</grammar>
```

Frage: Wie oft hat der Vortragende das Wort „**genau**“ benutzt?

oder

**Demo: Sprachanalyse des Vortrags**

**Noch Fragen?**